Improving the understanding of cancer and cancer care by applying data science and machine learning methods to electronic patient records

Andres Tamm

Wolfson College
University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Michaelmas 2023

Abstract

Electronic health records (EHR) hold great potential for improving the understanding of cancer care by containing high-resolution real-world data for large numbers of patients. This dissertation explores the application of data science and machine learning (ML) methods to EHRs for the purposes of translational colorectal cancer (CRC) research.

I first explore the challenges in using EHRs throughout the data life cycle. I present a lightweight information extraction pipeline that retrieves TNM staging scores—common descriptors of cancer severity—from free text clinical reports with high sensitivity and precision, and also retrieves information about the presence and recurrence of CRC. These data items are essential to CRC research, for identifying cases, studying treatment variation, and comparing treatment outcomes. The pipeline was developed using data from Oxford University Hospitals (OUH) and Royal Marsden (RMH) NHS Foundation Trusts (FT), and supported the establishment of the National Institute for Health Research (NIHR) Health Informatics Collaborative (HIC) CRC database.

I then focus on a specific application: combining the faecal immunochemical test (FIT) results with routinely collected data to predict CRC in symptomatic patients. The current practice is to refer patients with FIT above 10 µg/g for invasive endoscopic investigations, but only one in six investigated have CRC, motivating prediction model development. I demonstrate that an externally-derived model does not outperform FIT in the Oxford University Hospitals FIT dataset (OUH-FIT), and highlight the importance of clinically-relevant performance measures. I then show that employing more predictors, a spectrum of ML models, and novel training methods, was not sufficient to outperform FIT on OUH-FIT data. Finally, I build on and incorporate an existing sequence analysis method into an interactive app that allows to explore and cluster thousands of medical event sequences, such as visualising treatment patterns of CRC patients.

The principal contributions are: a holistic discussion of EHR data quality; a staging extraction algorithm that facilitates further research/audits; a comprehensive pipeline for developing/evaluating FIT-based CRC prediction models; and a fast medical sequence exploration app that can help check data quality and identify treatment variations. There is considerable potential to use these tools on larger datasets to understand if FIT-based models are bound to fail (or if they may work on subgroups with more severe disease); and to contrast different treatment patterns employed for subgroups of CRC patients with complex disease, such as those with liver metastases.

Improving the understanding of cancer and cancer care by applying data science and machine learning methods to electronic patient records

Andres Tamm

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2023

Acknowledgements

I am very grateful to my parents Marje and Tiit for their continuous love and support; to Micon my partner and sunshine; to my friends who met me with warmth and kindness, including Annie and Christian; and to Irem Luqman and Thomas Page for organising dance classes that kept me happy and sane throughout this time.

I am very lucky to have had Jim Davies, Brian Nicholson and Eva Morris as my supervisors. Their encouragement and belief in me helped me to believe in myself and to be kind to myself, especially during the difficult COVID period. They were very supportive and responsive throughout my DPhil years, and the depth of their expertise and experience helped steer this work in a rigorous and clinically relevant direction.

I am also very thankful to my examiners Professors Max Van Kleek and Helen Coleman for their effort in going through this thesis page by page (!) and offering their insights, suggestions for improvement, and valuable points of discussion. Feedback from Prof. Max van Kleek and Dr Marion Mafham was essential during my Transfer and Confirmation to help me think critically and plan the work well.

The work on electronic health records is inevitably collaborative. I am indebted to Chris Cunningham, Steve Harris, Helen Jones, Stephanie Little, Theresa Noble, Will Perry, Gail Roadknight, Hizni Salih, Brian Shine, Kinga Várnai, Kerrie Woods, Tingyan (Tina) Wang, Jaimie Withers, and Diana Withrow (in alphabetical order), and to other members of the NIHR HIC colorectal cancer collaborative.

I also gratefully acknowledge the support from the Engineering and Physical Sciences Research Council (EPSRC) that provided me with a doctoral studentship as part of the Centre for Doctoral Training in Health Data Science.

This thesis was created using the OxThesis LaTeX template by John McManigle, Sam Evans and Keith A. Gillow [1].

Abstract

Electronic health records (EHR) hold great potential for improving the understanding of cancer care by containing high-resolution real-world data for large numbers of patients. This dissertation explores the application of data science and machine learning (ML) methods to EHRs for the purposes of translational colorectal cancer (CRC) research.

I first explore the challenges in using EHRs throughout the data life cycle. I present a lightweight information extraction pipeline that retrieves TNM staging scores—common descriptors of cancer severity—from free text clinical reports with high sensitivity and precision, and also retrieves information about the presence and recurrence of CRC. These data items are essential to CRC research, for identifying cases, studying treatment variation, and comparing treatment outcomes. The pipeline was developed using data from Oxford University Hospitals (OUH) and Royal Marsden (RMH) NHS Foundation Trusts (FT), and supported the establishment of the National Institute for Health Research (NIHR) Health Informatics Collaborative (HIC) CRC database.

I then focus on a specific application: combining the faecal immunochemical test (FIT) results with routinely collected data to predict CRC in symptomatic patients. The current practice is to refer patients with FIT above 10 µg/g for invasive endoscopic investigations, but only one in six investigated have CRC, motivating prediction model development. I demonstrate that an externally-derived model does not outperform FIT in the Oxford University Hospitals FIT dataset (OUH-FIT), and highlight the importance of clinically-relevant performance measures. I then show that employing more predictors, a spectrum of ML models, and novel training methods, was not sufficient to outperform FIT on OUH-FIT data. Finally, I build on and incorporate an existing sequence analysis method into an interactive app that allows to explore and cluster thousands of medical event sequences, such as visualising treatment patterns of CRC patients.

The principal contributions are: a holistic discussion of EHR data quality; a staging extraction algorithm that facilitates further research/audits; a comprehensive pipeline for developing/evaluating FIT-based CRC prediction models; and a fast medical sequence exploration app that can help check data quality and identify treatment variations. There is considerable potential to use these tools on larger

datasets to understand if FIT-based models are bound to fail (or if they may work on subgroups with more severe disease); and to contrast different treatment patterns employed for subgroups of CRC patients with complex disease, such as those with liver metastases.

Contents

List of Figures

List of Abbreviations

BERT . . . . . Bidirectional encoder representations from transformers

BRC . . . . . Biomedical research centre

CRC . . . . . Colorectal cancer

CV . . . . . Cross-validation

EHR . . . . . Electronic Health Records

G . . . . . Gilbert, the most adorable little furry ball of love

Hb . . . . . Haemoglobin

FIT . . . . . Faecal immunochemical test

LoRA . . . . . Low Rank Adaptation of large language models (LLMs)

ML . . . . . Machine learning

MCV . . . . . Mean cell volume

NAM . . . . . Neural additive model

NICE . . . . . National Institute for Health and Care Excellence

NIHR HIC . . National Institute for Health and Care Research - Health Informatics Collaborative

NLP . . . . . Natural language processing

NPV . . . . . Negative predictive value

OUH . . . . . Oxford University Hospitals

PPV . . . . . Positive predictive value, also known as precision

PR curve . . . Precision-recall curve

RWD . . . . . Real-world data

SGT . . . . . Sequence graph transform

Heading on this journey with both excitement and self-doubt, holding both gently.

1 Introduction

Contents

1.1 Aims and context

This dissertation is aimed at applying machine learning methods (ML) to electronic health records (EHRs) to improve the understanding of cancer and cancer care, with a special focus on translational research into colorectal cancer (CRC). Colorectal cancer is the fourth most common cancer in the UK and the third most common in the world, with 44,100 new cases diagnosed every year in the UK [2, 3]. CRC is also the second most common cause of cancer deaths both in the UK and in the world, accounting for 10% of all cancer deaths between 2017 and 2019 in the

UK and about 9.2% of new deaths in 2020 in the world [2, 3].

I have worked closely with the National Institute for Health and Care Research (NIHR) Health Informatics Collaborative (HIC) CRC theme at Oxford, to help establish a multi-centre CRC database [4]. I was initially hoping that the database would grow faster, so that it would be large and complete enough for significant ML analyses, such as finding predictive factors of cancer recurrence, but this has not yet been the case. Nevertheless, this work has been essential for gaining a better understanding of EHRs and developing information extraction tools that are more broadly useful for cancer research, and it can also yield other outputs if the database continues to grow.

I have also worked with the Oxford University Hospitals (OUH) clinical biochemistry laboratory and researchers from the Primary Care Health Sciences department, to see if the faecal immunochemical test (FIT) use can be optimised to better detect CRC in symptomatic patients by combining it with other routine data. These efforts led to the development of the OUH-FIT dataset, which I subsequently used for exploring the validity of an externally-derived prediction model and for evaluating whether locally-derived models could beat current clinical practice.

Throughout the dissertation I have collaborated with the Oxford NIHR Biomedical Research Centre (BRC) clinical informatics team that has been developing pipelines to effectively extract data from EHRs for research use. I have thus had the valuable opportunity to learn more about the information extraction pipelines that are a key component of EHR based research. The work of the BRC informatics team has also been essential for contributing Oxford data to the NIHR HIC CRC database, for establishing the OUH-FIT dataset, and for setting up the secure data environments needed to effectively work with these datasets.

Importantly, this dissertation has been conducted in an interdisciplinary setting, receiving much needed input from clinicians, including Dr Helen Jones, Dr Will Perry, Dr Brian Nicholson, and Dr Neel Doshi.

Overall, this work illustrates how EHR data can be applied from the ground-up in service of improving healthcare, and points out challenges that need to be

considered on the way. The next section provides a brief overview of each of the chapters included in the thesis.

1.2 Outline and key research questions

The dissertation starts with groundwork: Chapter 2 discusses data quality and ethical issues that pertain to using electronic health records (EHRs) for research, and Chapter 3 develops a pipeline for extracting key information about colorectal cancer from free text. The next two chapters focus on a specific application of combining the faecal immunochemical test (FIT) with routine data to improve colorectal cancer detection: Chapter 4 attempts to externally validate an existing risk prediction model, and Chapter 5 locally develops machine learning models for the same purpose. The last substantial Chapter 6 develops a data visualisation tool that can help better understand data quality and whether the data may contain temporal patterns of events that are predictive of an outcome.

These chapters correspond to five key research questions (RQs):

1. What are the key data quality issues and ethical aspects to consider when using electronic health records for research? (Chapter 2)

2. Is it possible to develop a lightweight information extraction pipeline that can detect whether a free text histopathology or imaging report discusses current primary colorectal cancer or current recurrent colorectal cancer, and that can extract cancer TNM staging scores from anywhere in the report? Lightweight means that the algorithms depend on a small number of packages which can be relatively easily installed. (Chapter 3)

3. Can the Nottingham colorectal cancer risk prediction models outperform the faecal immunochemical test (FIT) for colorectal cancer detection on Oxford University Hospitals (OUH) data? The models would outperform the FIT, if they reduce the number of patients referred to subsequent investigations based on their FIT result while capturing the same number of cancers as FIT. (Chapter 4)

4. Is it possible to find a machine learning model—from a set of models with varying degrees of interpretability and flexibility—that would outperform the FIT test for colorectal cancer detection on routinely collected Oxford hospital data? 'Outperforming' has the same meaning as in RQ 3. (Chapter 5)

5. Is it possible to develop a lightweight software program that can automatically group patients with similar medical event sequences, and display the grouped sequences, so that a user can have a quick overview of how the different medical events follow each other over time and how different patterns of medical events may be associated with different clinical outcomes? 'Lightweight' has the same meaning as in RQ 2. (Chapter 6)

1.3 Publications

Publications and manuscripts associated with this dissertation are listed in Table 1.1. One paper has been published that describes the establishment of a multi-centre colorectal cancer research database, outlines a pipeline of data quality checks, and showcases the potential of the data [4]; it overlaps with the content of Chapter 2. The information extraction algorithms described in Chapter 3 were used to support a pilot analysis described in that paper, but the development of these algorithms was not described there. The algorithms that extract the affirmation status and TNM staging of colorectal cancer from free text are defined and validated in an unpublished manuscript that will be submitted to BMJ Health and Care Informatics. Analyses described in Chapters 4 and 5 are currently in the manuscript stage as well.

Table 1.1: Publications and manuscripts associated with the dissertation

Chapter	Publication or manuscript	Status
2	Andres Tamm et al. "Establishing a colorectal cancer research database from routinely collected health data: the process and potential from a pilot study". In: *BMJ Health & Care Informatics* 29.1 (2022).	Published.
3	Andres Tamm, Helen Jones, Neel Doshi, ..., Eva Morris. "Supporting cancer research on real-world data: Extracting colorectal cancer status and explicitly written TNM stages from free-text imaging and histopathology reports". Manuscript to be submitted to *BMJ Health & Care Informatics*.	Approved by H Jones, waiting approval and feedback from other authors.
4	Andres Tamm, Brian Shine, Tim James, ... Eva Morris, Jim Davies, Brian D. Nicholson. "External validation of the COLOFIT colorectal cancer risk prediction model in the Oxford-FIT dataset: the importance of population characteristics and clinically relevant evaluation metrics". Manuscript to be submitted to the *Gut*.	Under review by senior author Brian Nicholson.
5	Andres Tamm, Brian Shine, Tim James, ... Eva Morris, Jim Davies, Brian D. Nicholson. "It is hard to beat FIT: Machine learning models that combine the faecal immunochemical test (FIT) with routinely collected data did not outperform FIT for colorectal cancer detection in 51,477 patients. Manuscript that will likely be submitted to the *Lancet eClinicalMedicine*.	In the process of being updated given a new batch of Oxford data and the changes in patient population over time.

1.4 Additional information on colorectal cancer

This dissertation relies on the data of individuals with diagnosed or suspected colorectal cancer (CRC). Here, additional background on CRC is provided for reference.

1.4.1 Incidence

CRC is the fourth most common cancer in the UK and the third most common in the world [2, 3]. In the UK, 44,063 new cases were diagnosed every year between 2017 and 2019 [2]. In the world, approximately 1.88 million new cases were diagnosed in 2020, corresponding to 9.8% of all new cancer cases [3] and amounting to an age-standardised incidence rate of about 19.6 cases per 100,000 person-years [5].

1.4.2 Mortality and survival

CRC is the second most common cause of cancer deaths both in the UK and in the world, accounting for 10% of all cancer deaths between 2017 and 2019 in the UK and about 9.2% of new deaths in 2020 in the world [2, 3]. For CRCs diagnosed between 2015 and 2019 in England, age-standardised 1-year survival was 77.3% for females and 79.7% for males, and age-standardised 5-year survival was 58.9% for females and 58.6% for males [6].

1.4.3 Diagnostic pathways

In the UK, the standard diagnostic pathway recommended by the National Institute for Health and Care Excellence (NICE) is to triage most symptomatic patients in primary care with the FIT test, before referral to a suspected cancer pathway [7], although patients are also diagnosed in other ways. According to 2020 data from England, colorectal cancers were most commonly diagnosed via urgent suspected cancer GP referrals corresponding to NICE guidance (37%), followed by emergency presentations (24%), GP referrals (16%), screening (11%), and other routes (12%) [8].

1.4.4 Risk factors

World Cancer Research Fund and the American Institute for Cancer Research reviewed 99 studies covering more than 29 million adults and reported strong evidence for the following protective and risk factors [9]. Protective factors include physical activity, and consumption of wholegrains, dietary fibre, dairy, and calcium supplements. Risk factors include smoking, inflammatory bowel disease, consumption of processed or red meat, consumption of two or more alcoholic drinks per day, being overweight or obese, and being tall. There is also evidence that early life exposures may have a significant role [10].

1.4.5 Pathogenesis

Colorectal cancer develops when cells in colonic crypts accumulate genetic and epigenetic alterations, first developing into a neoplastic polyp and later into a malignant lesion [11]. There are at least three molecular pathways that can result in the development of CRC: the conventional pathway leading to chromosomal instability, the microsatellite instability pathway, and the serrated pathway [12], each associated with a set of genetic mutations. Schmitt and Greten note that patient survival and treatment response cannot be accurately predicted from the presence of these mutations alone, and the interaction of tumour cells with the tumour microenvironment and associated inflammatory processes are important for pathogenesis [13].

1.4.6 Treatment

Common treatment options for CRC include endoscopic treatment, surgery, radiotherapy and different classes of drugs for metastatic cancer; multiple treatment regimes are often given to patients with metastatic cancer [11].

The first step in effective data analysis is to imagine the data generating process.

2

Understanding and visualising electronic health records

Contents

The data that supports this thesis comes entirely from electronic health records (EHR), which raise unique challenges for research. In this chapter, I review the research use of EHRs, describe the processes I applied when working with and checking the quality of patient records, and conclude with a discussion of some ethical themes that I was aware of when applying ML models to EHRs.

2.1 The research use of electronic health records

Digitisation of the healthcare system and health records is a relatively recent phenomenon. In UK, initial attempts to digitally transform the NHS between 2002 and 2011 were mostly unsuccessful [14], although significant but insufficient progress had been made by February 2023 [15]. An essential part of the digital transformation strategies has been the adoption of EHRs, with a target that 90% of NHS trusts use EHRs by December 2023—which was subsequently met [16]—and that all trusts use EHRs by March 2025 [15].

EHRs are a type of real-world data (RWD). Data from registries, claims, patient-reported outcomes, and wearables are some of the other forms of RWD [17]. The National Institute for Health and Care Excellence (NICE) has made the routine use of RWD as one of its strategic priorities [18], and notes that RWD can be used to "characterise health conditions, interventions, care pathways and patient outcomes and experiences; design, populate and validate economic models [...]; develop or validate digital health technologies [...]; identify, characterise and address health inequalities; understand the safety of medical technologies including medicines, devices and interventional procedures; assess the impact of interventions (including tests) on service delivery and decisions about care; [and] assess the applicability of clinical trials to patients in the NHS" [19]. EHRs in particular can contribute to all of these use cases and thus hold great potential for improving care and advancing the understanding of health and illness. And even though Randomised Controlled Trials (RCTs) are thought to provide the highest quality evidence for the safety and efficacy of treatments—due to carefully designed study protocols and randomisation that enables causal inference—non-randomised studies based on EHRs can provide complementary and essential information, especially when RCTs are not feasible or available or do not have enough external validity (NICE have listed several examples in their RWD guidance [19]). In general, evidence derived from EHRs could help better answer the question 'Will this intervention work in the population?' [20].

Despite their potential, a major limitation is that EHRs have originally not been created for research purposes [21] and can have various data quality issues [22].

Beyond the specifics of data quality (see the next section), Sauer et al discuss general pitfalls in the usage of EHRs for research, especially in the context of developing ML models [21]. These include but are not limited to – *sample selection bias* that can arise when researchers use too broad or too narrow operational definitions when selecting the cohort of patients they are interested in; *imprecise variable definitions* for outcome variables and other characteristics that are studied; failure to distinguish that *timestamped results may not have been available to clinicians at the time of their measurement* and hence did not contribute to care decisions at these time points; *failure to understand the context* in which data was generated, such as when only billable procedure codes are recorded or when diagnosis codes are assigned retrospectively with variable timestamps; *data leakage* between model development and validation sets, especially when multiple records per patient are available and can independently be used to train the model; the *relationship between disease severity and data availability*, such that individuals who are more severely ill have more complete or more frequent observations; *unobserved variables* that influence the data generation process, for example treatment allocation decisions that are dependent on the personal style of the physician and on other less documented characteristics of the patient and their circumstances; *failure to use clinically relevant performance measures* when developing a prediction model, for example, using aggregate measures such as the *c*-statistic when the model is meant to be used only at high levels of sensitivity or specificity.

In the long-term, the clinical workflows and software systems that EHRs depend on could also be improved, so that they provide more high quality data by design and support the vision of a learning health system (LHS): a system where routinely collected data is used continuously in collaboration with patients, clinicians and other stakeholders to improve care [23].

2.2 Data quality and data transformations

It is paramount to assess the quality of data that has been extracted from EHRs before using it in research or service evaluation studies. According to a recent review

[22], there are seven dimensions of EHR data quality that can be distinguished: *completeness* of the record for a given data item; *correctness* of the observed values; *concordance* of the observed values with other data items and data sources; *currency* or how up to date the data is; *plausibility* of the observed value or its distribution; *conformance* of the data item to a data model; and *bias* which refers to missingness not at random. For example, a blood test record is not *complete* if it is not available for all patients who meet the study eligibility criteria; it is not *concordant* if the test result is above the normal range in a laboratory database, but another clinical report claims that the result was 'normal'; the record is not *plausible* if it is reported as "80,000" but the observed maximum value in a large number of other patients has been "800"; and the record is not *conformant* with a data model if the model requires numeric results but results are instead reported as 'positive' or 'negative'. The authors also note that while the majority of data quality assessments (DQA) occur after data has been extracted from the EHR, there are earlier stages in the data life cycle where DQA is important, including data entry, data transfer to a warehouse, and data extraction for a specific research project. At the data entry stage, it can also be important to understand which factors of the user interface support data quality [24], especially when designing or updating the data entry system. Furthermore, data quality is context dependent: data can be of sufficient quality for some research questions but not for others [25].

As part of my DPhil, I have helped to establish a multi-centre NIHR HIC colorectal cancer (CRC) database [4] by helping to collate and check the quality of data submitted by the participating research centres. I created a Python-based data quality workflow, partly based on previous data quality reports created by Tingyan (Tina) Wang at Oxford University Hospitals NHS Foundation Trust, and by the NIHR HIC Cardiovascular COVID-19 theme at Imperial College Healthcare NHS Trust. The workflow reads in data from a relational database or .csv files, detects basic quality issues, points out the locations of potentially problematic columns, provides a quick overview of the submitted tables and columns, and saves the results into a data quality report in Excel (the formatting of which I later manually

adjusted to make it neater). Figure 2.1 illustrates what the summary sheet of the data quality report looks like. The main quality issues that I was detecting were *completeness* (the proportion of missing records, number of patients with missing records, availability of units for blood test results); *correctness* (records that contain unusual values such as non-word characters); *plausibility* (dates that are out of plausible range); *concordance* (do patient identifiers in all tables match with the patient identifiers in the cohort-defining demographics table); and *conformance* (were all data items specified in the data model submitted). These were minimal feasible checks; additional quality checks are probably necessary when using the data for a specific research question. Initially, I also reported min/mean/median/max and percentile statistics, which are useful for checking plausibility. Considering earlier stages of the data life cycle, the NIHR HIC CRC database collated data that had already been entered in the respective NHS trusts by healthcare workers, and the data entry process varied depending on the data item due to different protocols and information systems used. For example, in OUH, laboratory test results are usually automatically entered into the laboratory information system once a barcoded sample is processed by the corresponding machine, whereas histopathology reports are generally generated by a reporting pathologist who fills out a form that is then stored in a cellular pathology database. If any questions arise about the data entry process during research, it is possible to contact the contributing centres for more information. Once the data had been submitted by the centres, it was further processed by a reproducible code script that ensured the data of different centres was in the same format (for example, that dates are always in the same format, and that the same column has similar possible values across centres).

Ideally, each data item stored in the EHR would also contain *metadata* about how it was generated and what it means. This can be essential for effectively using the data and understanding its quality, but in the EHR datasets I have worked with this was often not available. Discussing the data items with clinicians who were familiar with the data collection process helped to resolve ambiguities. For example, when working with the OUH-FIT dataset, it was not clear if missing values for the FIT test

results meant that the faecal sample had not been returned by the patient, or that it had been returned but the biochemical assay failed (for example, due to insufficient quantity of the sample). Consultation with the clinical biochemistry team revealed that it often meant the sample was returned, and this distinction was especially important for one of the future studies that are to be conducted with that dataset.

Metadata should also include any *data transformations* that were applied to raw data to generate the final data product provided to researchers. Ideally, data transformations would be coded in scripts to ensure reproducibility, in which case metadata could point to the code that was run to generate the tables or columns. Despite having a common data model, the data submitted by each research centre to the NIHR HIC CRC database was at times in a slightly different format. To combine the data, I used a semi-manual approach where I first inspected the data items to see which transformations need to be applied, and then specified the transformations in a reproducible python script (such as specifying the names of columns that need to be changed, or specifying the date formats that need to be applied to each date column). I designed the script such that the transformations were specified as a dictionary, making it easier to inspect what was done, especially as the number of tables was large.

Temporality is another essential aspect of EHR data quality [26]. This is because the processes used to collect the data, and the patient population itself, can change over time and systematically influence the recorded values. Quan et al [26] give examples of how this can lead to false conclusions when comparing the number of hospital admissions and infection severity over time. They recommend inspecting the time series of key variables to detect abrupt changes, and also provide a software tool to help accomplish this [27]. In my work, it was necessary to better understand the quality of OUH-FIT data, and by recommendation of Prof Gary Abel, I visualised the proportion of patients who had a positive test result over time, which revealed a trend of increasing positivity (see Appendix C.6). Some of this change was probably due to the adoption of a different sample collection strategy (there was a small increase in recorded test positivity after a new collection

device was introduced), and some of the change was probably due to changes in patient population (the increase in the proportion of positive tests was positively correlated to the increase in the number of patients being tested). This can have implications for how well ML models designed to predict the risk of cancer work on that data (if the population continues to change to a higher-risk population, for example, then ML models could eventually work better on that dataset).

NIHR HIC Colorectal Cancer: Data Quality Report

Centre ...
Date 13/07/2021

Summary

No major quality issues were found, as such there is no need to change the submitted data now (we only had questions about a few columns, but these are not a priority). Please focus on providing the remaining essential data items as the next step. In the order of priority, we would ask:

...

Which potential issues are present?

(Note: these are automatically flagged potential issues and are not necessarily problematic.)

potential_issue	issue_present
Data not submitted	yes
Subject identifiers do not match	no
More than 90% of rows are empty	yes
Column available for less than 10% of patients	yes
Column is empty	yes
Column contains unusual values	yes
Column contains same values in multiple letter cases	yes
Date other than birth is before 1900 or after 2022	no
Birth date is before 1900 or after 2022	no
Some episodes have no diagnoses or procedures recorded	yes
Some patients have multiple colorectal sites recorded	yes
Potentially unusual blood test result	no
Missing blood test unit	yes
Missing blood test reference range	yes

Where are the potential issues present, and do they require any actions?

level	item	priority	table	column	potential_issues	action	priority_action
table	asa	essential			Data not submitted;	data could be submitted when ready	2
table	imaging	essential			Data not submitted;	data could be submitted when ready	1
column	smokingstatus	essential	NIHRHIC_(	SmokingStatus	Column contains same values in multiple letter cases;	no action needed; can be fixed at analysis stage	
column	smokingstatus	essential	NIHRHIC_(	PackYears	More than 90% of rows are empty;Column available for less than 10% of patients;	please check if more data could be provided	3

Figure 2.1: First sheet of a data quality report. The first sheet provides a summary, lists which of the potential issues were detected, and points out locations of these issues along with their priority.

2.3 The life cycle of EHR data

As noted in the previous section, data contained in EHRs passes through a life cycle before it is provided to researchers for a specific project. This includes the original data entry; any subsequent modifications to data; possible transfer of data to a data warehouse; extraction of relevant data items from the data warehouse (or directly from EHRs) for a specific study; and any deidentification and post-processing

pipelines that are applied, such as generation of anonymised patient identifiers, redaction of dates and names from free text clinical reports, and shifting of dates in structured data. At each stage, issues and challenges can occur, and when researchers finally analyse the extracted data, failure to consider the previous data generating processes can lead to misleading inferences (Figure 2.2). Many of the data quality issues that can arise were discussed in the previous section. However, considering the EHR data life cycle can provide a more holistic perspective about the issues that can occur, and help to proactively find potential issues by reflecting on how each stage is locally implemented. For example, extracting required data items from the data warehouse can be a complex procedure, because the desired data items are often spread out over multiple databases, some of which can be complex. In my work with the NIHR HIC CRC data, it was not always clear if extracted chemotherapy treatments represented planned or administered treatments, and further examination of the chemotherapy database schema helped to clarify this. As another example, the deidentification processes that are applied to extracted data can impact the performance of any natural language processing algorithms that are subsequently applied to the data, especially if deidentification unintentionally removes relevant information. When initially working with NIHR HIC CRC data, I noticed that the existing deidentification algorithm unintentionally removed certain cancer TNM stage values as it mistook these for postcodes (this was subsequently fixed).

It is also worth considering how the structure of the healthcare system may affect the health data life cycle. In England, 42 integrated care systems (ICS) are responsible for most NHS services and collaborate with different service providers, including but not limited to primary care providers (covering general practitioners, dentists, optometrists, pharmacists) and NHS Trusts and Foundation Trusts (providing most hospital, community, mental health and specialist care) [28]. Most NHS trusts that share patients do not use the same electronic health record system which poses a challenge to sharing data between hospitals [29] and may also affect all stages of the health data life cycle. For example, data entry protocols may differ between systems; data extraction for research may require setting up separate

pipelines for each system to extract data about the same medical entities; and interpretation of the data during analysis requires care that data obtained from different systems has the same meaning. There may also be differences in practice between trusts that are not related to electronic health record systems and that nevertheless affect how data is collected and modified.

Furthermore, the research use of EHRs also holds the potential to improve data quality via feedback. For example, if researchers discover that some data items are entered in an unstandardised or incomplete way, this could be fed back to clinicians who enter the data, to initiate a discussion on whether data entry protocols and front-end systems could be improved.

Overall, the successful use of EHRs for research requires careful consideration of all stages in the data life cycle, and it is likely that clinicians or other people familiar with the local data collection processes need to be involved to make effective use of the data and avoid false inferences [21].

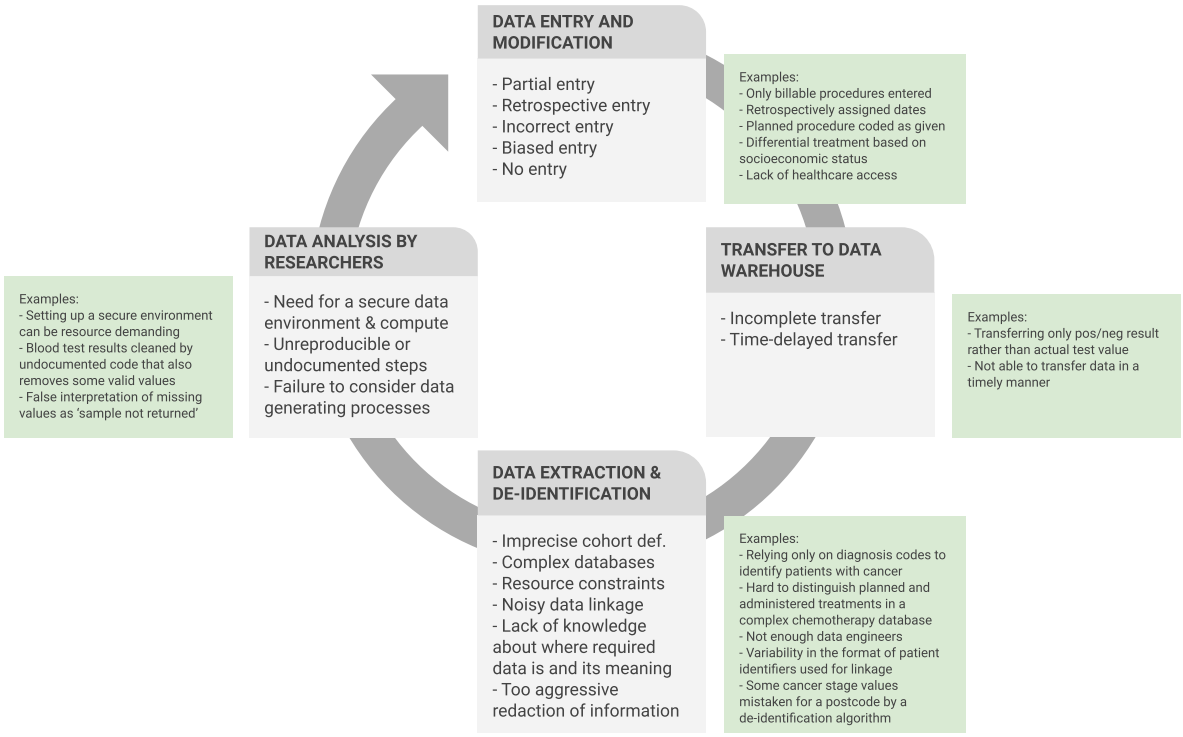


Figure 2.2: Life cycle of EHR data when used for research, and issues that can arise at each stage. ***During original data entry*** – clinicians may enter data partially, retrospectively, or incorrectly, such as entering procedure codes for planned not given procedures [21]; patients from some socioeconomic statuses may be more likely to be misdiagnosed [30] leading to biased records; some demographic groups may lack healthcare access and never contribute to EHRs [30]. ***During the transfer of data to warehouse*** – data may be transferred partially such that only summary results like 'positive' or 'negative' are transferred; and data may not be transferred in a timely manner due to resource constraints. ***During data extraction and de-identification*** for a specific research study – patient cohort may be derived too strictly using only diagnosis codes [21] while additional patients could be identified by analysing free text medical reports; the data items required for a study can be contained in multiple, complex databases, and it can take effort to identify these; the clinical informatics team may be under strain leading to project delays; de-identification algorithms applied to extracted data can unintentionally remove relevant information. ***During analysis*** – a research project may not be feasible due to lack of resources for setting up a secure data environment; unreproducible or undocumented processing steps can lead to misunderstandings when multiple researchers work on the same data; failure to consider the previous data generating processes can lead to biased results [21]. Note: the life cycle does not show ethical and regulatory stages.

2.4 Beyond descriptive statistics: visualising patient event traces

An essential aspect of understanding the quality of EHR data is the ability to describe and examine it. The data quality pipeline I used also provided descriptive summaries for all tables and columns included in the dataset. Although it is valuable to summarise the data one item at a time, it can also be valuable to examine how the data of each patient "looks like" as a whole. One way to accomplish this is to plot a timeline of events for each patient (see Figure 2.3). In such plots, each patient is represented by a horizontal line that indicates time, and events of interest are marked on it using different symbol-color combinations. The events can be different treatments, investigations, and outcomes; and event times can be given relative to a reference event (such as diagnosis), or relative to some absolute time. These timeline plots can help understand the data and to check its quality. For example, the timelines of some patients in the NIHR HIC CRC database consisted mostly of radiotherapy events. Follow-up investigations showed that some of these patients were coming to receive radiotherapy in a NIHR HIC centre, but their main patient records were probably stored elsewhere, which means that the data for these patients can be too incomplete to be reliably used. The timeline plots can also help interpret subgroups that arise when clustering EHRs, especially when clustering clinical event sequences (see Chapter 6). The clusters can then be interpreted by examining the clinical event timelines for a randomly selected subset of patients that belong to each cluster.

Several tools already exist for creating visual summaries of temporal patient records, some of which are reviewed in [31]. For example, the LifeLines2 software [32] displays different classes of events on separate lines (rather than on one line) and similarly aligns events to a reference event. The advantage of the method reported here is its simplicity: it requires minimal software installation and could easily be deployed in the trusted research environment that stores the data.

A limitation of the timeline plots is that they are meaningful only if the number of unique events that are plotted is relatively small: otherwise, the graph can easily

become saturated. I nevertheless hope it is a valuable tool for inspecting and understanding EHRs.

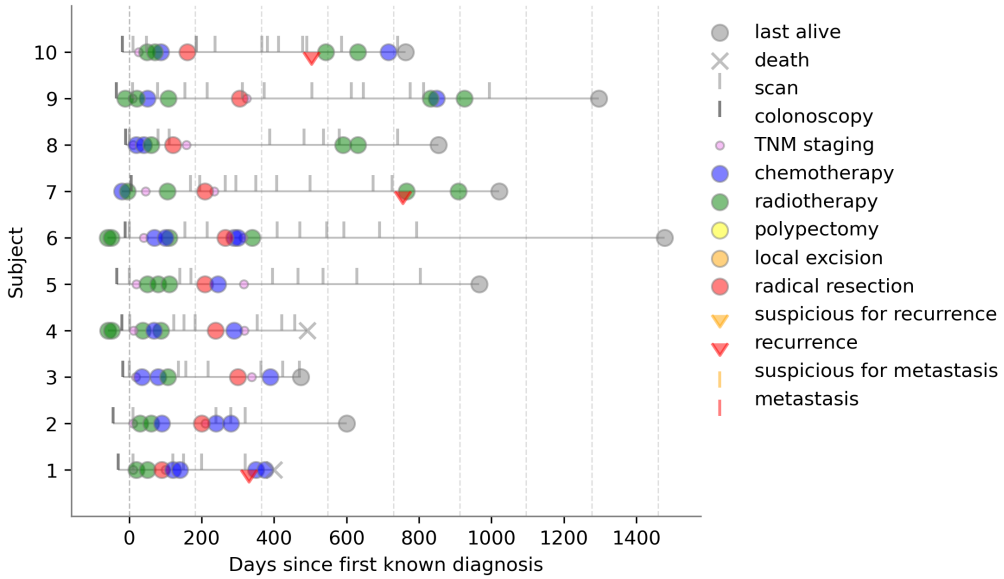


Figure 2.3: Example of patient timeline plots that show hypothetical treatment patterns. In these plots, each patient is represented by a horizontal line, and events of interest are marked on the line with specific symbol-color combinations based on the time when they occurred. Here, all timelines contain the pattern "diagnosis, scan, chemoradiotherapy, radical resection, chemo(radio)therapy, scan". This figure is based on another graph created by the author and subsequently published in [4].

2.5 Ethical aspects

There are a variety of ethical issues that pertain to the use of EHRs in medical research, such as the justification for research use, the effect of data management and data quality on the validity of research and its benefits, the impact on healthcare system operation and patient engagement, potential risks to patients and other parties, approaches that should be taken to mitigate risks, how and how broadly should consent be obtained, rights to autonomy *vs* duty to support public good, the fair sharing of any research benefits, among others [33]. The use of ML models in health care raises partly similar and partly unique ethical issues that, according to a fairly recent review in 2021, fall under the themes of data privacy and security, trust, accountability for errors, and bias in algorithms and data [34]. The issue that health datasets can be inaccurate or unrepresentative, and thus lead to biased

algorithms that differentially benefit some members of the population [30, 34], is perhaps worth of special mention, because data is a key ingredient for developing any ML algorithm. Furthermore, it is not enough for individuals from diverse backgrounds to be represented proportionally or equally in health datasets, as there can still be biases in how the data is recorded (e.g. a particular demographic group may be more likely to be misdiagnosed and thus have less accurate health records) [30]. Below, I will discuss some of these issues in relation to my work.

Data diversity. The datasets I have worked with are not diverse in terms of age or ethnicity. For example, in the OUH-FIT dataset that I used for comparing current clinical practice to CRC risk prediction models, at least 72% individuals were white and 78% were at least 50 years old. The dataset contains records for 31,964 patients who were offered the FIT test, which is usually done when there is suspicion of CRC, and the risk of CRC significantly increases with age [35]. It is therefore understandable that there is age bias, although it also means that any insight derived from that data is less likely applicable to younger age groups. In addition, imbalance between the number of cases and non-cases for an outcome variable such as cancer, can determine whether there is enough data to evaluate an ML model in that subgroup. Even though the number of younger patients in the OUH-FIT dataset may seem relatively large at face value (2,971 individuals were less than 40 years old), the absolute number of cancers in that age group was small (14). Therefore, any estimate for how well a cancer risk prediction model performs in that age group is not reliable, because it is conditional on these 14 patients being representative enough of the cancers seen in this age band. Similarly, as the OUH-FIT dataset contains only seven cases of cancer across the 1,450 patients who reported non-white ethnicities, it is not possible to study whether the risk prediction models developed on that data would generalise to non-white ethnic groups. In my analyses, I was not able to develop a cancer risk prediction model that would work better than current clinical practice, and hence there was no need to perform such follow-up analyses. However, if for some reason these models would

work better for non-white ethnic groups, then the lack of data diversity means that this pattern could not have been discovered.

Non-maleficience. The duty to do no harm is a common principle of medical ethics [36]. I have thought about it in the context of externally validating a Nottingham CRC risk prediction model in the Oxford dataset. The model uses FIT test results in conjunction with demographics and blood tests to make predictions (see Chapter 4). Validating a model implies there is hope to deploy it at one point for the benefit of patients. However, the real-life deployment of any risk prediction model can be very risky: the model may not generalise beyond the population it was developed on (e.g. it did not generalise to Oxford population in the current analysis); it may cease to work if the patient population changes in time (e.g. if the FIT test starts to be offered more broadly to patients such that those with even lower risk symptoms are included); and it can also fail if the recording and measurement of blood test results that the model depends on changes (e.g. if the hospital opts for a new biochemical assay with different analytic performance). Furthermore, models that are developed and validated on *retrospective* data (as in this case) can underperform in *prospective* real life usage, even if they are shown to 'work' in the retrospectively collated dataset. This can happen, for example, when data recorded in retrospective datasets is influenced by the health status of the patient, so that the blood tests that the model depends on were not ordered for patients who were assessed to be healthier by clinicians, but who would still qualify to be tested by the model in prospective usage [21]. Any deployment thus needs to be carefully considered to avoid harm.

Truthfulness, autonomy and trust. A major issue in applying ML models to health data is also *predictive uncertainty* [37]. Risk prediction models, such as the Nottingham-derived CRC risk prediction model that I validated, usually output point estimates of risk, as in 'John Smith has 10% risk of cancer'. Clinical researchers commonly try to ensure that these risk estimates are valid by evaluating model calibration in the target population [38], which shows that predicted risks correspond to true risks *on average*. However, even if a model is calibrated on

average, there can still be considerable uncertainty in the individual point estimates of risk[1], which implies that simply reporting point estimates may not be truthful (and could be misleading). For example, if an uncertainty interval was included, the Bayesian estimate for John Smith might read 'there is 95% probability that JS's risk of cancer lies between 0% and 30%'. There is a question then about how this should be explained to a patient so that they can make an informed choice about whether to participate in any further testing and thus exercise their autonomy. It can also be reasonable to abstain from using risk predictions when the uncertainty is too high, and instead collect more information about the patient in these cases [37]. Having valid estimates of predictive uncertainty can also help foster trust in the ML system [37], which can be essential for the system to be deployed effectively and beneficially [34].

Finally, even though the use of real world health data in conjunction with the powerful tools of data science and machine learning holds great potential for improving health care, then as Kerasidou and Kerasidou have noted, it is not clear if data-intensive health technologies are the best way to promote health compared to other possible social interventions, because health is inevitably linked to the conditions in which people are 'born, grow, live, work and age' [40]. So while it is good to keep responsibly exploring and developing our health data science capabilities, we must not forget that caring for our health is a much, much wider enterprise.

2.6 Protocols for accessing and analysing data

All empirical chapters of this thesis rely on data obtained from the Oxford University Hospitals (OUH). To access the data, I obtained an honorary contract with the OUH

[1]As reviewed by Kompa et al [37], uncertainty can come from observation noise in the variables that were measured and uncertainty in model parameters [39], and also from dataset shift. Consider an example of uncertainty in model parameters: if one has fitted a logistic regression model and obtained the regression coefficients that are applied to each data point to make a risk prediction, it is clear upon reflection that these coefficients are not final. The coefficients would change if one developed the model again on a new sample of data from the same population, and thus the predicted risk for that particular patient would also change. This uncertainty in model parameters should in some way be incorporated into an uncertainty interval around the risk estimate.

and completed several trainings including information governance. Pseudonymised data was analysed in a remotely accessed trusted research environment. Research results were exported using an "airlock" mechanism: a request form was filled, detailing what files were exported and why, and a member of the OUH clinical data team then reviewed the request and files. No patient-level data was exported.

For analyses involving the FIT test (Chapters 4 and 5), the use and collation of data were described in detail in a Data Protection Impact and Assessment (DPIA) form that OUH Information Governance approved and the studies were registered in OUH as a service evaluation (first CSS-BIO-3-4730, later updated as 9076).

For analyses involving the NIHR HIC data (Chapters 3 and 6), study protocols were approved by the NIHR HIC colorectal cancer theme team. All participating centres had local information governance approvals for sharing the data with OUH. The protocol for the collection and management of NIHR HIC CRC data has been reviewed and approved by the East Midlands - Derby Research Ethics Committee (REF Number: 21/EM/0028).

2.7 Other sources and repositories of cancer data

This dissertation relied entirely on data derived from EHRs of patients with diagnosed or suspected CRC. Here, other sources of cancer data that are potentially relevant to the research questions are described and discussed.

2.7.1 Data from cancer registries and related sources

Cancer registries systematically collect data for all cancer cases in a defined population with the aim of improving care [41].

In England, the National Disease Registration Service (NDRS) collects data for all individuals with cancer [42, 43]. The Cancer Outcomes and Services Dataset (COSD) is used as the national standard for cancer reporting [44]. Other national datasets can be linked to the registry data: the Radiotherapy Data Set (RTDS) [45, 46], the Systemic Anti-Cancer Therapy Dataset (SACT) that encompasses chemotherapy data [47], and hospital episode statistics (HES) that covers hospital

diagnoses and procedures [48]. The COloRECTal cancer data repository (CORECT-R) aims to link datasets relevant to CRC [49] (`https://www.ndph.ox.ac.uk/corectr/corect-r`). A key resource in CORECT-R is the National Colorectal Cancer Dataset, which contains information on all CRCs diagnosed in England between 1997 and 2018. It collates information from the cancer registry, RTDS, SACT, HES, and patient experience surveys.

National cancer data is also collected by cancer registries in Northern Ireland [50], Scotland [51], and Wales [52].

2.7.2 Cancer registry data linked to primary care data

English cancer registry data can also be linked to primary care data contained in the Clinical Practice Research Datalink (CPRD) [53] (`https://www.cprd.com/cprd-linked-data`), the QResearch (`https://www.qresearch.org/`), and OpenSAFELY [54] (`https://www.opensafely.org`) databases. The number of patient records covered by these databases is in the order of millions. The SAIL databank in Wales collates various data sources for the Welsh population, including but not limited to cancer registry data, primary care data, and hospital episodes, and may also serve as a source of registry data linked to primary care data [55] (`https://saildatabank.com/data/explore-the-data/`).

2.7.3 The relevance of EHRs over other data sources

A substantial part of this dissertation (Chapters 4 and 5) focussed on whether the faecal immunochemical test (FIT) can be combined with demographics, blood tests and other routine data to improve on the accuracy of FIT alone in symptomatic primary care patients. National cancer data alone does not encompass the target population (primary care patients with suspected CRC) and was not used. It is possible that primary care databases that link to registry data, such as the CPRD, would cover the target population and necessary data items, and this can be explored later. The Oxford University Hospitals data was well suited for the research questions involving FIT, as it covers the target population, allows checking data quality with

local clinicians, and contains histopathology reports and hospital records that can be used to identify cancer cases in a transparent and reproducible manner.

The information extraction pipelines described in Chapter 2 were designed to extract information about the presence and recurrence of CRC from free text to support research into CRC. The establishment of the NIHR HIC CRC database that collates EHRs from multiple NHS trusts [4] motivated the development of these pipelines, as data submitted to cancer registries by the participating trusts was not available at the time. Even if registry data could be linked to the NIHR HIC CRC database, it is still desirable to have algorithms that extract cancer information from the rawer source data, because this enables additional data quality checks and supports transparency and reproducibility. Furthermore, the NIHR HIC CRC database does not duplicate the work of cancer registries, as it encompasses a richer set of variables than those that can feasibly be collected by registries (such as blood test results from secondary care), and the automated data extraction pipelines used during the collation of the database can potentially make timely data available faster than in registries. Cancer registry data on the other hand provides a much larger sample size and geographical coverage and could have potentially been used for treatment pattern clustering analyses in Chapter 6.

There is nothing quite so useless, as doing with great efficiency, something that should not be done at all.

— Peter Drucker

3

Extracting information about the presence, stage, and recurrence of colorectal cancer from free text clinical reports

Contents

3.1 Introduction

3.1.1 Motivation

There is great potential to use routinely collected NHS data to improve colorectal cancer care: if this data could be automatically collated across multiple NHS trusts, it would allow studying specific questions about colorectal cancer treatments and outcomes that require high resolution real-world data, complementing the lower-resolution and larger national cancer datasets [4, 49]. However, routinely collected data is often unstructured, such that key points of information are only contained in free text format [56].

Our aim was to develop a lightweight software tool that can extract essential information about colorectal cancer from free text histopathology and radiology reports, to support the creation of a NIHR HIC colorectal cancer database [4]. We initially focused on three pieces of information: (1) detecting if primary colorectal cancer is present, (2) characterising the severity of cancer by extracting the TNM staging values, and (3) identifying cancer recurrence and metastasis. Not all biomedical research centres participating in the NIHR HIC colorectal cancer theme were able to send their anonymised clinical reports to be included in the database: we therefore aimed to create a pipeline that could easily be run in another hospital without requiring complex installation or significant computational resources (such as a GPU). The main pipeline thus uses regular expressions (regex) and a variation of the ConText algorithm [57]. Initial evaluation showed that it worked well for tasks (1) and (2), and somewhat less for (3) - we thus also included a transformer-based machine learning model as it may be better at processing contextual information for detecting the site of recurrence/metastasis.

Before proceeding to describe the information extraction pipeline, we first review the essential concepts of cancer TNM staging and recurrence, and methods that have already been employed to detect these.

3.1.2 TNM staging summarises the severity of cancer

The stage of cancer (its size and spread) is essential for planning and evaluating treatments, estimating prognosis, auditing the quality of care, tracking cancers at the population level, and conducting both observational and clinical research into cancer [58]. It is recorded using the TNM classification which has three main components [58]: T for describing the size and local spread of primary tumour; N for indicating if cancer has invaded the regional lymph nodes; and M for indicating if cancer has spread to another region of the body. Each component is recorded using a small set of values, often reported in a sequence. For example, 'T1 N0 M0' approximately means 'relatively small cancer that has not spread to regional lymph nodes or other parts of the body'. The TNM classification also includes other categories, such as lymphatic invasion (L), venous invasion (V), perineural invasion (Pn), residual tumour status (R) and tumour grade (G), that are similarly reported with letters and numbers.

Despite being an essential descriptor of cancer, the TNM categories are not necessarily recorded in a structured format in electronic patient records. Instead, they can appear in free text in variable ways (Table 3.1). Furthermore, clinical reports can contain alphanumeric strings that are like TNM categories but have a different meaning. For example, 'T1' may also refer to the first thoracic vertebrae or to a T1-weighted magnetic resonance image.

We therefore designed the TNM detection component of our pipeline to capture TNM staging in a variety of formats, and to distinguish TNM staging from similar-looking phrases. Note that we focused on extracting explicitly given TNM staging values, but some medical reports that do not report TNM staging in customary letters and numbers may still contain enough information to infer the staging – this

is a more difficult task that we did not currently attempt (difficult because the TNM categories are not assigned and need to be deduced from natural language).

Table 3.1: Variability in reported TNM values

Example	Comment
T1/2/3 N0 V0	Multiple values given for T category
T1 NO MO	Zero mis-spelt as O
T1 n1	Some letters written in lower case
T1N0M0	No gap between letters
T 1 N0M0	Gap between letter and value
pT1 *vs* ypT1 *vs* ymrT1	Variability in how prefixes are written
T1a (solitary tumour) N0	TNM values separated by text such as comments
R0 pT3 L0 V0 N0 Mx *vs* pT2 N1 Mx L0 V0 R1	Multiple TNM values given in variable order
Staged as T2	Only a single TNM value is given

Existing algorithms for TNM stage extraction

Several attempts have been made to extract explicitly given TNM staging values from free text (Table 3.2). It is hard to evaluate how thorough and flexible these methods are as only two studies have publicly available source code [59, 60] and one provides extraction rules in their publication [61].

Considering methods where source code is available, Abedian et al [59, 62] used a regex-based tool to extract TNM staging from pathology reports for four cancer subtypes, achieving overall accuracies of 89%, 90% and 97% for the T, N and M stages respectively. However, their regex assumes the reports to follow a specific structure (staging is assumed to be given after phrases such as 'pathological staging') and allows for less variation in how the staging is written (does not allow gaps, mis-spelling of 0 as O, or repeated values). This tool can also be harder to set up as it requires the installation of multiple software packages. Odisho et al [60, 63] also used regex, but required specific anchoring characters to precede the staging phrase ('pt', 'yp' and 'p') and seemed to require the staging to be given in a very specific format 'pt<number>n<number>m<number>'. D'Avolio et al [61] allowed staging to be written more flexibly (gaps, 0 mis-spelt as O), but did not allow repeated values, text comments between TNM values, and TNM subcategories ('1a' instead of '1').

Ansoborlo et al [64] and Khor et al [65] note in their publication that they tried to account for false positive matches (such as 'T2' referring to a vertebra), and Khor et al [65] specifically used exclusion keywords such as 'scan' to filter out these false positives, but it is not clear how flexible their extraction rules were otherwise. The other publications did not provide enough detail about the extraction rules [66–70].

Overall, we are not aware of any freely available algorithm that can extract TNM stages written in a sufficient variety of ways from anywhere in the free text report.

Table 3.2: Studies where explicitly given TNM staging has been extracted from text

Author	Year	TNM categories	Source code	Cancer type	Data sources	Region	Method	Flexibility
Ladas et al [66]	2023	preT, T, N, M, L, V, R, G, Pn	No	Unknown	Pathology reports	Germany	regex	Unknown
Ansoborlo et al [64]	2023	T, N, M	No	Tracheo-bronchial	MDT reports	France	regex, naive bayes classifier	Accounts for false positive patterns (e.g. T2 vertebra)
Huang et al [67]	2023	T, N	No	Urological	Pathology reports	Singapore	regex-based	Unknown
Abedian et al [59]	2021	preT, T, N, M	Yes [62]	Breast, colon, prostate, other	Pathology reports	USA	regex applied to a window around anchoring term	Limited: no gap, no misspelling, no repetition, strict prefix, requires anchoring phrase (such as "primary tumour")
Odisho et al [60]	2020	T, N, M, Pn	Yes [63]	Prostate	Pathology reports	USA		Limited: requires anchoring characters ("pt", "yp", "p"), and format pt<number>n<number>m<number>
Khor et al [65]	2019	T, N, M	No	Bladder, kidney, prostate, testes	Pathology reports, radiology reports and clinical notes	Australia	regex	Accounts for false positive patterns (e.g. 'T1 image')
AAlAbdulsalam et al [68]	2018	T, N, M	No	Colon, lung, prostate	Abstract records and pathology reports	USA	regex	Unknown
Kim et al [69]	2014	T, N, M	No	Prostate	Pathology reports	USA	regex	Unknown
Ashis et al [70]	2014	T, N, M	No	Breast, lung, prostate	Pathology reports	USA	regex	Unknown
D'Avolio et al [61]	2008	T, N, M	Regex in paper [61]	Prostate	Pathology reports	USA	regex	Allows for gap, 0 mis-spelt as O (in N and M). Does not allow repeated values or comments between TNM values

3.1.3 Recurrence and metastasis are essential cancer outcome variables

Cancer can re-emerge after treatment from residual tumour cells, either in a similar anatomical location as the original tumour (locoregional recurrence) or in a site distant from it (systemic, or distant recurrence)[71, 72]. Recurrence is an essential outcome variable in colorectal cancer studies that assess the quality of care, compare the efficacy of treatments or find prognostic markers [73]; for example, six of the seven research questions that were listed by clinicians during the development of the NIHR HIC CRC database required identifying cancer recurrence [4]. Metastasis is a concept related to recurrence, indicating that cancer has spread to another part of the body (this could be detected before or during treatment, in which case it is often called synchronous metastasis, or after, in which case it may be called metachronous [74] and may represent distant recurrence). However, similarly to TNM staging, information about recurrence and metastasis was not recorded in a structured format in any of the hospitals that submitted data to NIHR HIC CRC database, and the publications reviewed below indicate that it may not be routinely recorded in general.

Existing methods of recurrence detection

At least two strategies have been used to retrospectively ascertain cancer recurrence in electronic health records: the first uses diagnosis and treatment patterns to identify likely instances of recurrence, and the second uses natural language processing (NLP) to extract information from clinical reports.

The first approach is exemplified by Danish studies that use clinical codes to identify a disease-free period after treatment and to look for evidence of recurrence thereafter, such as the presence of ICD-10 codes for metastasis without a record of new primary cancer diagnosis [73, 75, 76]. It is also illustrated by a recent study that looked for specific patterns of chemo- and radiotherapy, surgical procedures, and causes of death after initial primary treatment, to infer breast cancer recurrence [77].

A variety of NLP methods have also been used for the detection of cancer recurrence, reviewed by Sangariyavanich et al [78]. For example, Strauss et al used

a rule-based algorithm that detected medical concepts, assigned assertion statuses (such as negation and uncertainty), and applied additional rules on that data to infer recurrence [79]. Carrell et al mapped clinical text to standardised medical concepts and devised rules that assess the presence and absence of certain concepts to infer recurrence [80]. Zeng et al also mapped text to standardised medical concepts, filtered out negated and uncertain concepts using rules, and used the remaining concepts as inputs to a support vector machine classifier [81]. Liu et al used versions of the BERT transformer model [82] to classify entire discharge summary reports for metastasis [83]. Batch et al used convolutional and recurrent neural networks to classify entire computed tomography (CT) reports for metastasis, including the current report and all previous reports at each time step; they also used a baseline model that converted text to vectors with the 'term frequency inverse document frequency' (TF-IDF) method and passed it through an ensemble of classical machine learning models [84]. Sangariyavanich et al [78] list additional studies, though for our purposes it is relevant to note that a variety of methods have been used (rule-based algorithms, classical ML models, deep ML models) that operate on different levels of the text (sections of the report, entire report, sequence of reports), and that use different representations of text (including but not limited to bag of words, TF-IDF, and learnable numerical embeddings). Sangariyavanich et al also note that transformer-based deep learning models performed the best [78], although any performance comparisons of models across studies are limited because there is likely a large heterogeneity in the medical texts that were used to develop the models.

We additionally note that several of the mentioned studies use more general algorithms as part of their pipeline, such as the NegEx method for detecting if extracted concepts were negated [85]. Some of these methods are similar to the ConText algorithm, a more general rule-based method of assigning assertion statuses (such as presence, possibility, negation) to extracted concepts [57]). More recently, van Aken et al compared classical and deep learning methods for assigning assertion statuses ('present', 'possible', 'absent') to clinical concepts, and found that transformer models pretrained on biomedical and clinical data performed the best [86].

While it could be valuable to use diagnosis and treatment patterns to identify recurrence in the NIHR HIC CRC data, we decided to pursue the text extraction strategy first, because histopathology and radiology reports contain descriptions of the actual tumour and are thus closer to the ground truth data than the clinical codes that are subsequently assigned.

3.2 Methods

3.2.1 Ethics approval

This analysis uses data from the NIHR HIC CRC database [4]. The protocol for the collection and management of that data has been reviewed and approved by the East Midlands - Derby Research Ethics Committee (REF Number: 21/EM/0028). The project proposal for extracting information from text was approved by the NIHR HIC CRC coordinating team.

3.2.2 The information extraction pipeline at a glance

There are three main components to the information extraction pipeline:

1. Identify imaging and histopathology reports that describe current primary colorectal cancer (a regex-based algorithm).

2. Extract TNM staging categories, either from all clinical reports or those that correspond to primary colorectal cancer (a regex-based algorithm).

3. Detect if clinical reports discuss cancer recurrence and metastasis and identify its anatomical site (implemented as a regex-based algorithm, and also accomplished by transformer-encoder models).

Each component is described in subsequent sections.

3.2.3 Identifying reports that discuss colorectal cancer

To detect whether a clinical report discusses current primary colorectal cancer (CRC), we processed it in three stages:

1. Identify keywords referring to colorectal tumours.

   (a) Extract matches for words and word pairs that directly represent colorectal tumours, such as "crc", "colo-rectal carcinoma", and others.

   (b) Extract matches for general tumour keywords such as "cancer" and retain the ones that are preceded or followed within a 100-character distance by a colorectal site keyword such as "splenic flexure" or "sigmoid". (100 characters performed well in semi-manual validation, when extracting matches and inspecting outputs.)

2. Exclude tumour keywords that have one of the following statuses: negated, general, historic, possible (i.e. not affirmative), metastatic, or treatment response (see Table 3.3 for examples). The remaining keywords are assumed to represent tumours in present and affirmative sense.

3. Decide that a clinical report mentions current CRC, if it has at at least one tumour keyword that was not marked for exclusion in the previous step.

We used an extensive vocabulary of keywords for tumours and anatomical sites created in collaboration with Dr Neel Doshi (see Appendix A.1). In Step 2, the statuses of tumour keywords were identified using a ConText-like algorithm [57]: we looked for certain keywords on the left and right sides of the tumour keywords (such as "no" and "negative" for the "negated" status, see Appendix A.2 for more examples), and accepted these only if certain termination keywords did not occur between the status keyword and tumour keywords (such as "although", "apart"). Diverging from the original ConText algorithm, we did not use the same termination keywords for all statuses; we considered the status keywords only if they occurred within a maximum allowed distance from the tumour keywords; and we did not

apply pseudo-trigger terms used in the original algorithm. Constraining the distance was useful for detecting specific patterns while reducing false positives (e.g. we used a short distance for "no" on the right side to capture phrases "tumour: No", because "no" does not usually refer to the tumour keyword when it occurs farther away on the right side). Keywords for detecting the assertion statuses (such as negation) were partly taken from the python implementation of the ConText algorithm [87], and partly created by examining the clinical reports.

Table 3.3: Examples of assertion statuses assigned to colorectal tumour keywords

Assertion status	Example
Negated	no evidence of colorectal cancer
General	in patients with colorectal cancer
Historic	history of colorectal cancer
Possible	suspicious for colorectal cancer
Metastatic	recurrent colorectal cancer
Treatment response	reduction in size of colorectal tumour

3.2.4 Extracting TNM stage values

TNM staging is commonly reported using a limited set of letters and numbers (Table 3.4), so it should be extractable by pattern matching with regular expressions. This was accomplished in four stages:

1. Extract phrases that contain TNM staging
   (a) Extract phrases that contain a sequence of TNM values, such as "pT1 (text) N0 (3/10) M0 R0 V0 L0".
   (b) Extract phrases that contain a single TNM value, such as "stage: pT1".
2. Filter and clean the phrases
   (a) Split phrases that contain multiple T values at the T value. For example, "pT1 N0 M0 text pT2 N1 MX" would be split into "pT1 N0 M0 text" and "pT2 N1 MX".
   (b) Identify historical phrases and mark these for exclusion.

(c) Identify unusual phrases such as multiple-choice prompts for T staging ("T 1 / 2 / 3 / 4"), and mark these for exclusion.

(d) Clean the phrases, retaining only TNM values. For example, "pT1 a & b (text) N0" would be replaced with "pT1a/1b N0".

3. Extract TNM values from the cleaned phrases. For example, if the cleaned phrase is “pT1a/1b N0”, then “1a” and “1b” will be the extracted T values and 0 will be N value. A total of 11 TNM categories were extracted, including T, N and M (Table 3.4). Values were only extracted from unique phrases to reduce running time.

4. For each clinical report, report the maximum and minimum values for each TNM category (as some reports may contain multiple TNM phrases).

Extracting TNM values in stages also allows for quality checks: the TNM phrases extracted at Step 1 can be examined along with their surrounding text, and manually excluded from subsequent steps if needed.

The regular expressions that extract TNM phrases were designed to reduce false positives. Firstly, the building-block patterns that match individual TNM values were constrained, so that each TNM value must be immediately preceded and/or followed by another TNM value (as in "t0n0m0") or be surrounded by nonword characters such as empty spaces (as in "summary: t0 n0 m0."). Secondly, when extracting a sequence of TNM values, only certain common sequences were allowed, such as "T ... N ... M", "T ... N ... R", "T ... N", among others, where "..." can contain text or other TNM values. This is because in valid TNM phrases, the values are usually given in a certain order. Thirdly, when extracting TNM values that appear alone, a dictionary of keywords was used to retain or reject them. For example, "T1" was retained if preceded by the word "staging" (as in "staging: pT1") but rejected if closely followed by the word "no" (as in "tumour perforation (pT4): No"). On the other hand, the patterns were made flexible enough to avoid false negatives – they allowed for repetitions in TNM

values, mis-spelling of 0 as O (only in sequences of values), gaps between letters and values, and comments between values. These steps together should ensure that valid TNM phrases are extracted from anywhere in the free text report, without having to detect the sections of the report first.

During the cleaning stage, historical TNM-phrases were identified using a method similar to the ConText algorithm [57]. Phrases were marked as historical if they were preceded or followed by certain words, and if other termination patterns did not occur in between. For example, if a TNM phrase was preceded by the word "prior", and words like "although", "apart", and "however" did not occur between "prior" and the TNM phrase, it was marked as historical. However, as most reports did not contain historical phrases, the performance of detecting these was not evaluated in this paper.

Table 3.4: TNM categories extracted by the algorithm

Shorthand	TNM category	Possible values
Tpre	Prefix of primary tumour, e.g. "yp"	Combination of the letters a, c, m, p, r, y. For example, "yp", "ymr", "p"**.
T	Tumour extent	0, 1, 1a-1d, 2, 2a-2d, 3, 3a-3d, 4, 4a-4d, X, is
N	Nodal invasion	0, 1, 1a-1c, 2, 2a-2c, 3, 3a-3c, X
M	Distant metastasis	0, 1, 1a-1c, X
V	Venous invasion	0, 1, 2, X
R	Residual tumour status	0, 1, 2, X
L	Lymphatic invasion	0, 1, X
Pn	Perineural invasion	0, 1, X
G	Grade of differentiation	1, 2, 3, 4, X
SM	Kikuchi level*	1, 2, 3
H	Haggitt level*	0, 1, 2, 3, 4, I, II, III, IV

Notes. *The Kikuchi and Haggitt level are not part of the TNM classification but were extracted because they are reported similarly and were of interest for colorectal cancer clinicians. **All combinations of these letters may not occur in practice.

3.2.5 Extracting information about recurrence and metastasis

The general approach

To identify whether a medical report discusses recurrent or metastastic colorectal cancer, we processed it in the following steps.

1. Extract all keywords that refer to recurrence or metastasis with 300 characters of text to the left and right. The keywords include, for example, 'recurrence', 'regrowth', 'mets', 'metatastic'. On inspection, 300 characters seemed to generally contain the information needed for classifying the keyword.

2. Identify the affirmation status of each keyword: classify it as 'present', 'possible', 'not present', 'historic', or 'other'. For example, if the keyword with surrounding text is 'there is no evidence of *recurrence* in this image', the status of the recurrence keyword is 'not present'.

3. Identify the anatomical sites associated with each recurrence/metastasis keyword. We first aimed to detect the broad anatomical site, classifying the keyword as 'distant', 'locoregional', or 'other'. The 'distant' category included sites of distant metastasis, such as liver, lung, bone, and skin. The 'locoregional' category encompassed sites of local colorectal cancer recurrence, such as the large intestine, and regional recurrence such as the peritoneum and omentum. The 'other' category was added for cases when anatomical site was unclear. We additionally attempted to detect whether the anatomical site was liver ('yes', 'no') and lung ('yes', 'no'), as these represented common sites of metastasis. We did not create models for other specific anatomical sites, because the number of examples in the labelled data was likely too small for obtaining reliable results.

As the set of keywords that refer to recurrence/metastasis is small, the first step was accomplished using pattern matching with regular expressions. This also allows for manual quality control of results: all extracted keywords with surrounding text can be examined, and information extracted from each text extract can be manually corrected if needed. Steps two and three were executed with two different methods: a regex-based algorithm, and fine-tuned bi-directional transformer encoder models. We decided to include the transformer models, because initial evaluation indicated that performance of the regex-based algorithm was not as good for anatomical sites.

Regex-based algorithm

The regex-based algorithm for detecting the assertion status of each recurrence/metastasis keyword was very similar to the colorectal cancer detection algorithm described in Section 3.2.3), and was again based on a modified ConText [57] algorithm. We initially detected whether each recurrence/metastasis keyword was associated with one or more of the following statuses: present, possible, negated, historical, not assessed, general, false positive (see Table 3.5 for examples). Next, all recurrence/metastasis keywords that were associated with the negated, historic, not assessed, general and false positive statuses were excluded. Finally, the assertion status was assigned to be 'possible' if the keyword was associated with the 'possible' status, and 'present' if it was not associated with the 'possible' status but did match for the 'present' status.

Table 3.5: Examples of assertion statuses assigned to recurrence and metastasis keywords

Assertion status	Example
Present	there is an interval increase in liver metastases
Possible	probable recurrence
Negated	the site is clear of recurrence
Historic	clinical history: recurrent rectal cancer
Not assessed	recurrence cannot be assessed in this image due to artefacts
General	<drug name> is recommended for treating liver metastases
False positive	recurrent collection in the pelvis

To detect the anatomical site of recurrence, we used regular expressions that matched for anatomical sites associated with colorectal cancer, including both locoregional sites (such as large intestine) and distant sites (such as lung and liver). The dictionary was created with the help of Dr Helen Jones and Dr Neel Doshi, and included synonyms for each anatomical site (see Appendix A.1). Matches for anatomical sites were retained if they occurred within 100-character distance from the recurrence/metastasis keyword and were not preceded or followed by certain termination keywords (such as 'from' in 'recurrence *from* colon cancer'. Finally, each recurrence keyword was categorized as 'locoregional' if it was not associated with any of the anatomical sites from the 'distant' category and 'distant' otherwise;

each metastasis keyword was classified 'locoregional' if it was only associated with sites from the 'locoregional' category and 'distant' otherwise.

Bidirectional transformer encoder

Transformer is an artificial neural network that uses positional encoding and self-attention mechanisms to produce contextualised numerical representations ('embeddings') for each element in a sequence while allowing to process the elements in parallel [88]. The BERT model ('Bidirectional Encoder Representations from Transformers') is a variant of the transformer that is trained to predict randomly masked tokens in a sentence by integrating information from both the left and right sides of each masked token [82]. A BERT-style architecture is appealing for our task, because we attempt to classify specific keywords (referring to recurrence or metastasis) in a text extract, and the information to achieve that task may be contained on both the left and right sides. We used two versions of BERT as our 'base models': the DistilBERT which is a smaller but similarly performing version of the original BERT [89], and BioClinicalBERT which is a variant of the original BERT that was fine-tuned on biomedical and electronic health record data [90, 91]. DistilBERT was included because it was designed to be lightweight, and BioClinicalBERT was employed as a medical domain-specific BERT model; BioClinicalBERT was also one of the best-performing transformer models for clinical assertion detection according to van Aken et al [86].

We then fine-tuned each base model separately for the four different classification tasks: (1) Predict affirmation status of each recurrence/metastasis keyword ('present', 'possible', 'not present', 'historical', 'other'); (2) predict the broad anatomical site associated with each recurrence/metastasis keyword ('distant', 'locoregional', 'other'); (3) predict if anatomical site is liver ('yes', 'no'), and (4) predict if anatomical site is lung ('yes', 'no').

The architecture of each task-specific model consisted of the base model layers, plus a linear classification layer with softmax added on top of the last encoder block. The classification task was formulated as token classification (the added

layer made predictions for each token in the input sequence), but only predictions corresponding to the first token of the target keyword (e.g. 'recurrence') in each text extract were used to train the model. This is because the location of each target keyword is already known, and the transformer output corresponding to that token integrates information from the left and right sides of that token. Please see Figure 3.1 for a basic structure of the DistilBERT classifier, and Appendix D for more information on transformer models.

We also explored two different strategies of fine-tuning: (1) updating all parameters ('weights') of the model; or (2) representing the updates to each weight matrix as a product of two-smaller weight matrices (the Low Rank Adaption method, or LoRA [93]). The LoRA method is desirable in our case, because it does not require a separate copy of the fine-tuned base model to be saved for each of the four tasks; instead, only the update matrices need to be saved, which are much smaller due to greatly reduced number of parameters (the number of trainable parameters was less than 2% of the number of base model parameters). When the model is later used for inference, the update matrices can simply be merged with the base model. LoRA can also be used as part of a 'differentially private' model training pipeline that makes it less likely that any specific text extract can be reconstructed from the model weights [94], although we did not pursue this currently.

To finetune the models, we first created an annotated dataset of 1,826 report extracts that describe recurrence and metastasis, covering a variety of cases (see Section 3.2.8). We then split the entire reports (to which the extracts belonged to) into training, validation, and test sets (50%, 25%, 25%), stratifying based on the biomedical research centre (Oxford, Royal Marsden) and type of keyword (recurrence, metastasis); the extracts were split at the report rather than report extract level to reduce the likelihood of data leakage between data splits. When tuning all parameters of the base models, we fixed batch size to 16 and tuned the learning rate in [1e-5, 2e-5, 3e-5, 4e-5, 5e-5]. When using LoRA, we fixed batch size to 16, set the rank of update matrices to 8, and tuned the learning rate in [1e-4, 2e-4, 3e-4, 4e-4, 5e-4], the scaling parameter alpha in [8, 16], and also tuned the

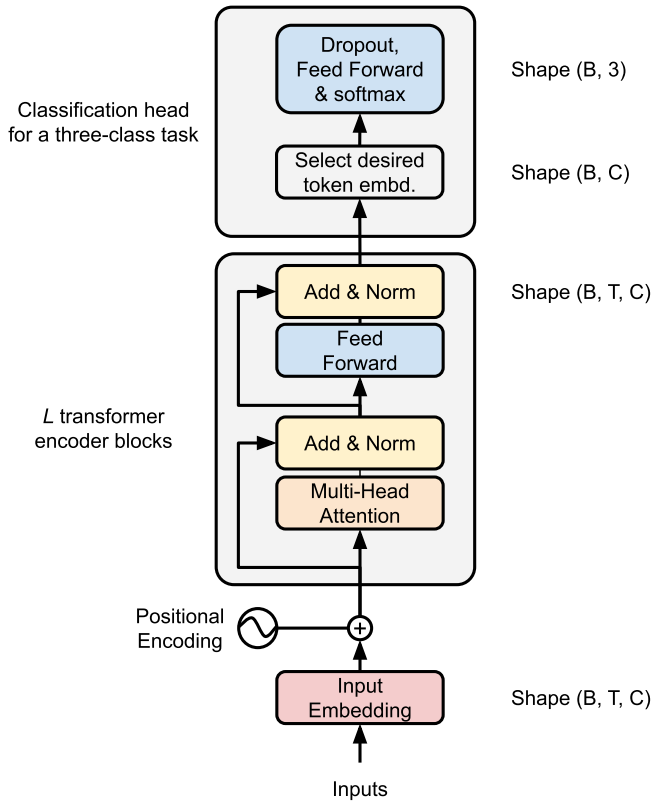


Figure 3.1: Diagram of the DistilBERT model for three-class token classification, based on Figure 1 in Vaswani et al [88] and Sanh et al [89]. Originally, the input data is of shape (B, T, C) where B is the number of sequences in a batch, T is the number of tokens in a sequence, and C is the dimension of the token embeddings. Positional encoding vectors are added to input data, and positionally encoded input data is passed through L transformer encoder blocks to produce contextualised embeddings. To use the transformer encoder output for token classification, the embedding of the desired token in a sequence is selected, yielding shape (B, C). This is passed through a feedforward neural network that linearly transforms the C input values into num_classes output values (in this example, three), and the softmax operation is applied to get class confidence scores. Dropout is additionally used during training. The classification head follows the default settings used by the Hugging Face implementation of the DistilBERT classifier [92]. Note that in Hugging Face implementation, the selection of the desired token was done implicitly by assigning a label "-100" to all tokens expect the target token, and the softmax operation was not separately applied during training because the cross-entropy loss in PyTorch package uses softmax implicitly. This model was used for classifying whether a text extract described recurrence or metastasis (target word) in a negated, present or possible sense. The data was processed such that the target word always occurred in the middle of the sentence. The first token corresponding to the target word was used for classification as it integrates information from both the left and right context of the target.

set of layers which are updated (we considered only updating the query and value matrices of the attention layers, or updating all weight matrices of the model). We trained for at most 10 epochs (with early stopping), using the AdamW optimizer and linear learning rate scheduler, and chose the model that performed best on the validation set. We then re-trained the base model on the combined training and validation sets using the selected hyperparameters and the number of training steps corresponding to the best performing number of epochs.

3.2.6 Iterative algorithm development on a multi-centre dataset

The regex-based algorithms were iteratively developed on imaging and histopathology reports of colorectal cancer patients as part of a NIHR HIC research programme to establish a colorectal cancer database [4]. It was primarily developed using 49,340 imaging and 15,435 pathology reports from the Oxford University Hospitals (OUH) NHS Foundation Trust (FT), and 49,340 imaging and 7,609 pathology reports from the Royal Marsden (RMH) NHS FT. It was also run on 4,252 imaging and 1,379 pathology reports from the data of Imperial College Healthcare NHS FT, and on reports from the Christie NHS FT. The data from Oxford and Royal Marsden NHS FTs was used iteratively by generating patterns to extract the desired information (primary colorectal cancer status, TNM staging, and recurrence/metastasis), then examining the unique values of extracted matches, and analysing errors in collaboration with clinicians. Data from Imperial and Christie NHS FTs was used once, and results fed back to the development team.

The transformer models for recurrence detection, unlike the regex-based algorithms, were developed on 1,826 text extracts from histopathology and imaging reports from the OUH and RMH NHS FTs; each text extract contained a keyword that represented recurrence or metastasis and 300 characters of text to the left and right (see Section 3.2.8). The number of text extracts for developing transformers was much smaller than the number of reports used to develop the regex-based algorithms, because initial development of the transformer models required annotated data,

whereas regex-based algorithms could be developed by applying them to the entire un-annotated dataset and examining the retrieved matches.

3.2.7 Evaluation of the CRC and TNM detection algorithms

The CRC and TNM stage algorithms were evaluated at the report level: we judged whether a clinical report was correctly marked as describing current CRC, and whether the maximum TNM stage values were correctly extracted from the report (we focused on maximum values, as most reports only contained a single values for the TNM categories, although some contained multiple).

Selection of clinical reports

To assess CRC detection, we randomly sampled a hundred reports from each of the four categories: imaging reports predicted to contain CRC according to the regex-based algorithm, imaging reports not predicted contain CRC, histopathology reports predicted to contain CRC, and histopathology reports not predicted to contain CRC. We repeated this process separately for OUH clinical reports used to develop the algorithm (selecting 400 reports), and on a newer sample of OUH clinical reports unseen during development (selecting an additional 400 reports among these 'future reports'), yielding a total of 800 reports.

To evaluate TNM stage extraction, we similarly selected a hundred reports from each of the following categories: (1) imaging reports where a T, N or M value was detected; (2) imaging reports where no T/N/M values were detected; (3) histopathology reports where a T, N, or M value was detected; and (4) histopathology reports where no T/N/M values were detected. We again applied this process both to OUH reports used in algorithm development and to future OUH clinical reports, selecting a total of 400 + 400 = 800 reports. Performance on future OUH reports will not yet be reported, as these have taken much longer than expected for clinicians to evaluate, but will be included in a published version of this chapter.

The stratified random selection based on CRC and T/N/M detection helped ensure that enough positive samples were available to estimate the PPV and NPV

of the algorithms, because many reports did not contain CRC or TNM staging. We also used a separate sample of reports to evaluate CRC detection, and a separate sample to evaluate TNM stage extraction, to provide better estimates of the PPV and NPV of each algorithm on the dataset it was developed on (we also did not want to bias the TNM algorithm evaluation dataset to only contain clinical reports that were predicted to contain CRC by the CRC detection algorithm, as that would exclude any false negative reports).

When sampling imaging reports, we only selected from reports that contained imaging types commonly used for investigating CRC and its metastases (for example, we included MRI scans of the pelvis but excluded x-rays of the foot). These reports were identified using a list of imaging codes compiled by Dr Helen Jones. The histopathology reports were not further filtered due to absence of metadata.

The decision to select 100 reports from each category was based on feasibility (reviewing the reports is effortful and time of clinicians is limited), and on a simplified power analysis using Gaussian approximation to binomial distribution. According to the Gaussian approximation, when the true positive predictive value of the algorithm is assumed to be 90%, selecting 100 reports ensures that there is only a 10% probability that the sample-based estimator differs from the true value by more than 5%[1].

We did not use clinical reports from the Royal Marsden (RMH) and Imperial College Healthcare (ICH) NHS Foundation Trusts to evaluate the algorithm at this stage. This is because a test set was not set aside for the RMH data when developing the regex-based algorithms (a mistake of the author), and it was not possible to obtain newer clinical reports from their team due to resource constraints (we initially expected that to be easier). The information extraction pipeline was run on ICH clinical reports, but due to a change in their informatics team, and information governance policies, they did not have enough resources to locally

[1]The difference e between the true population proportion p and a sample proportion $\hat{p}$ is approximately Gaussian distributed with $e \sim \mathcal{N}(0, p(1-p)/n)$. The probability that the absolute value of this difference is greater than 0.05 is then given by $\mathrm{P}(|e| > 0.05) = (1 - \Phi(0.05/\sqrt{p(1-p)/n})) \cdot 2$, where n is the sample size and Φ is the CDF of the standard normal distribution.

evaluate the performance of the algorithm, or to send information to Oxford in enough detail. However, initial evaluation conducted on the RMH development set showed good performance, and examination of the extracted TNM phrases from Imperial (without being able to view the left and right side context) looked valid. In a future evaluation, we hope to include a new batch of clinical reports that are expected to be submitted by one of the new NIHR HIC participating sites. Depending on resources, we may also additionally repeat the evaluation process for RMH reports. However, we expect the results to be publishable, especially for the TNM stage detection algorithm, only based on OUH reports.

Annotation of selected reports

The reports selected for TNM stage evaluation were reviewed by Dr Neel Doshi, and reports selected for CRC detection were reviewed the author of this thesis in collaboration with Dr Helen Jones (the author reviewed all CRC reports, accepted obvious cases such as 'biopsy - rectal adenocarcinoma' based on a clinician-approved dictionary of tumour keywords, and flagged harder cases for review by Dr Jones). The algorithms were run on the selected reports, and reviewed by displaying both the extracted values and reports in a Shiny app [95] (Figure 3.2) and in LibreOffice. Correcting extracted values was significantly more time effective (especially due to the numerous TNM categories and relatively large number of reports) than annotating reports from scratch; this could induce positive bias if algorithm's mistakes are overlooked. However, a subset of the TNM staging reports was double checked by the author, and error analysis of all reports showed that mismatches between algorithms' predictions and ground truth values were definitely picked up.

Performance metrics

We evaluated CRC and TNM stage extraction with the metrics of PPV, NPV, sensitivity, and specificity, because these are familiar to clinical researchers. The outputs of the CRC detection algorithms were binary (a clinical report was predicted to discuss primary CRC or not), and hence the metrics could be computed in the usual way. However, as each TNM category can take multiple values (Table 3.4),

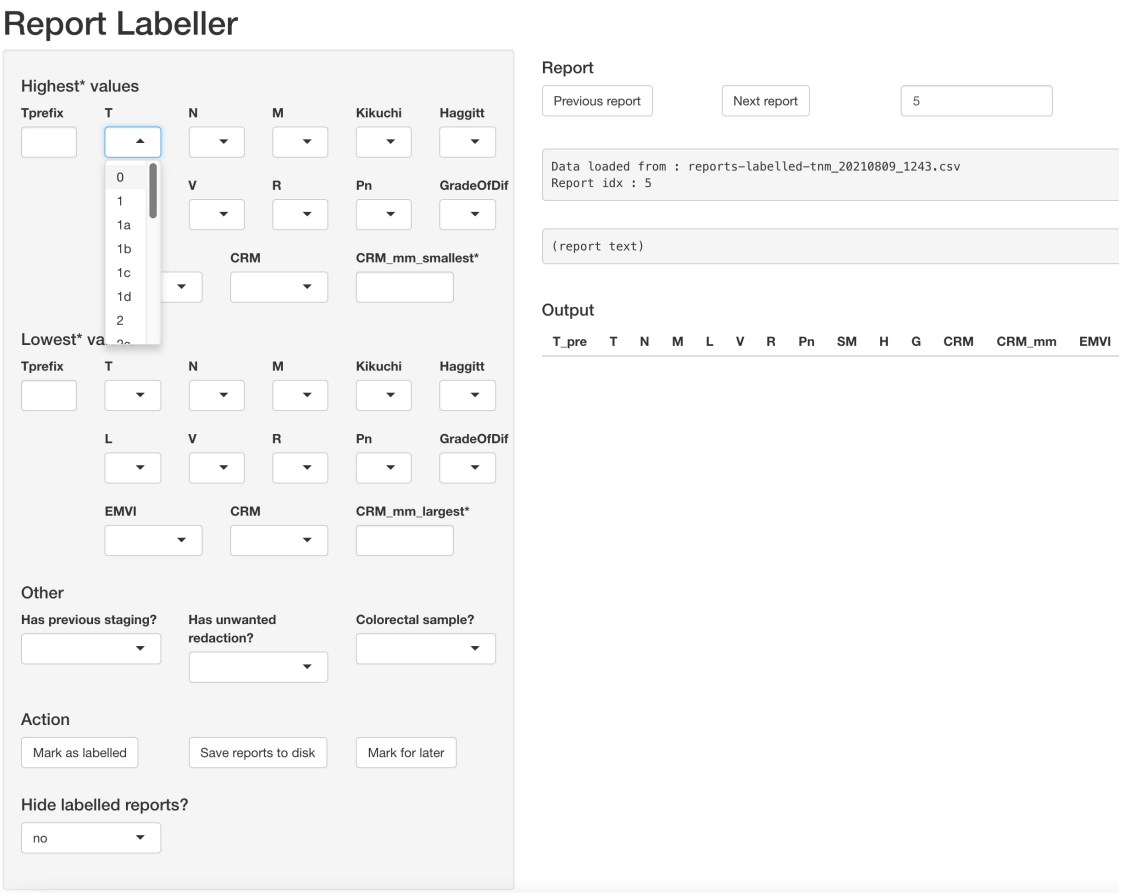


Figure 3.2: Shiny app for annotating imaging and pathology reports for TNM staging. The app displays reports and allows users to enter the required variables, or to correct information already extracted. It was originally developed for evaluating TNM stage extraction, but can also be used for evaluating other types of extracted information.

we computed these metrics relative to a 'null' value. For example, in the case of T category, PPV is the proportion of clinical reports where the maximum T value was correctly detected among all reports for which a T value was returned by the algorithm (i.e. where a non-null value was returned), NPV represents the proportion of reports that were correctly marked as not containing a T value among all reports where no T values were detected by the algorithm, sensitivity is the proportion of reports where the maximum T value was correctly detected among all reports that contained a T value, and specificity is the proportion of reports that were correctly marked as not containing a T value among all reports that did not contain a T value. PPV and sensitivity, when computed this way, are also known as micro-averages in the machine learning literature [96].

For evaluating TNM staging, we additionally computed PPV, NPV, sensitivity, and specificity separately for each of the main values under each TNM category. For example, the main T values are '0', '1', '2', '3', '4', 'X', and 'is'; and we computed PPV for value 0 (proportion of reports where the maximum value is zero among all reports where the maximum value returned by the algorithm was zero), PPV for value 1 etc. This was to double check that the algorithm could detect all values well, even those that occurred less frequently, but this analysis was limited due to a small sample size (for a similar reason, we did not use macro averages to evaluate TNM staging as some TNM categories were rarely present and their performance estimators highly variable).

We also note that the estimates of sensitivity and specificity may not accurately represent the true sensitivity and specificity of the algorithm if it was run on all clinical reports available to us, because we used stratified random selection when sampling reports; however, they still indicate the proportion of positive and negative examples that were correctly detected in the random sample. The estimates of PPV and NPV, however, should reflect well the PPV and NPV of the algorithm on all of our clinical reports, given sufficient sample size.

3.2.8 Evaluation of the recurrence detection algorithms

The recurrence detection algorithms were evaluated at the report extract level: we tested whether each report extract that mentioned recurrence or metastasis was correctly classified for its assertion status and anatomical site(s). We used the 470 held-out report extracts that were not used during the development of the transformer models (see below). It would later be useful to evaluate the algorithms at the report rather than report extract level - for example, if an anatomical site is not detected in a more difficult report extract, it could still be correctly detected in another extract from the same clinical report (see Section 3.4.2 for discussion).

Selection and annotation of report extracts

We first extracted keywords referring to recurrence and metastasis, along with 300 characters of surrounding text, from all OUH and RMH imaging and histopathology reports. The patterns that matched the keywords were developed in collaboration with clinicians and covered synonyms and different word forms for the concepts of recurrence and metastasis (e.g. 'recurrence', 'recurrent', 'regrowth', etc).

We then randomly sampled 90 text extracts from 16 different subsets of the data to ensure that the model development dataset contained a variety of examples ([research centre: OUH, RMH] x [report type: imaging, pathology] x [concept: recurrence, metastasis] x [negation status: negated, not negated] = 16 categories). The negation status was inferred using a previously developed regex-based algorithm (Section 3.2.5). When sampling from the negation status categories, we ensured that 2/3 of reports were predicted to be non-negated and 1/3 negated, as negated phrases are probably easier to detect (for example, many were of the form 'no liver metastasis'), and the non-negated category covers a broader set of categories (such as recurrence/metastasis being discussed in a present, possible, historical or general sense). After randomly sampling initial 1,440 text extracts (90 extracts * 16 categories), we additionally added all extracts that appeared in the same clinical reports as the sampled extracts to ensure that the text extracts covered entire reports, and removed duplicates, yielding 1,826 extracts in total. The text extracts were then annotated in collaboration with Dr Helen Jones (obvious cases such as 'recurrent rectal cancer' were accepted by the author and ambiguous or unclear cases were flagged for review by Dr Jones).

During the development of transformer models for recurrence detection, the 1,826 extracts were split into training, validation and held-out test sets (see Section 3.2.5). The held-out set included 470 report extracts (188 pathology and 282 imaging report extracts), and was used for evaluating model performance.

Performance metrics for recurrence ascertainment

We computed the standard metrics of PPV, NPV, sensitivity, and specificity. We evaluated the models (and the regex-based algorithm) on six classification tasks that are of particular interest:

- Status: present. Did the report extract indicate that recurrence/metastasis was currently present?
- Status: possible. Did the report extract indicate that recurrence/metastasis was suspected to be present?
- Site: abdominopelvic. Did the report extract associate recurrence/metastasis with an abdominopelvic anatomical site (such as large intestine or omentum)?
- Site: distant. Did the report extract describe recurrence/metastasis in a distant anatomical site (such as liver, lung, bone)?
- Site: liver. Did the report extract describe recurrence/metastasis in the liver?
- Site: lung. Did the report extract describe recurrence/metastasis in the lung?

Note that the transformer models fine-tuned for the assertion detection task initially predicted it to be one of ['present', 'possible', 'negated', 'historic', 'other'], and the models fine-tuned for detecting the broad anatomical site predicted it to be one of ['abdominopelvic', 'distant', 'other']. For assertion status, detecting the 'present' and 'possible' categories is of most interest and hence the performance for detecting these was evaluated separately; for anatomical sites, researchers are likely to be interested in specifically selecting patients with either abdominopelvic or distant recurrence, and hence it is reasonable to evaluate these separately as well. When the performance of models is evaluated on predicting a single class (such as the anatomical site being distant), it is possible to tune the classification threshold to achieve a specific level of sensitivity; for simplicity, this was currently not pursued - instead, we chose the prediction of each model to be the class with highest score (which is the same as highest probability *if* the models are calibrated).

In addition, we did not evaluate performance separately for recurrence keywords (such as 'recurrence', 'regrowth') and metastasis keywords (such as 'metastatic'), as the number of clinical report extracts was relatively small.

3.2.9 Software

The regex-based algorithms for CRC detection, TNM stage extraction and recurrence ascertainment were implemented in python 3.9, primarily relying on *numpy* (v1.23.5)[97], *pandas* (v1.4.3)[98] and *regex* (v2020.10.15)[99] libraries. The transformer model for recurrence ascertainment relied mainly on *pytorch* (v2.1.1)[100], *transformers* (v4.36.2)[101] and *peft* (v0.7.0)[102] libraries, with the *transformers* and *peft* being part of the HuggingFace ML platform [103]. The code will be made freely available at `https://github.com/tammandres/textmining` after the work has been published (although the transformer weights will not be published for privacy reasons).

3.3 Results

3.3.1 Primary colorectal cancer

The regex-based algorithm achieved approximately 95% sensitivity and at least 90% PPV for detecting primary CRC in imaging and pathology reports that had been used to derive the rules ('training data' in Table 3.6; Section 3.2.6). Note that in this case, the rules were derived by iterating over a much larger sample of reports, some of which were randomly selected for evaluation, so this performance estimate is less likely to be overly optimistic. Performance on a future set of OUH reports was lower: on future pathology reports, the sensitivity had dropped from about 95% to 84%, and on future imaging reports, the PPV had dropped from about 90% to 78%.

Error analysis showed that reduction in sensitivity on future OUH pathology reports was mainly due to supplementary reports that described genetic testing for mismatch repair proteins (MMR) without explicitly mentioning CRC: clinicians noted that these tests were done only when CRC was confirmed and hence they implied its existence (so the reports were labelled as containing CRC), but keywords for MMR had not been included in the regex-based algorithm. As these were usually supplementary reports, it was not crucial to correctly classify them; when these 11 reports were ignored (labelled as not containing CRC), the sensitivity on future OUH pathology reports increased to 94.5%. The drop in PPV on imaging reports was due to patterns not picked up by the algorithm: this included a specific negation pattern such as 'tumour is ... no longer ... seen/visible', general statements such as or 'CRC MDT' (colorectal cancer multi-disciplinary meeting), or where the report discussed complete tumour response to treatment (in which case the tumour was no longer present).

3.3.2 TNM staging

The TNM staging algorithm had at least 97% PPV and 83% sensitivity for detecting the maximum values of all TNM staging categories in the OUH histopathology and imaging reports (Table 3.7). For most TNM categories, sensitivity was higher

Table 3.6: Performance of a regex-based algorithm for detecting primary colorectal cancer in histopathology and imaging reports of OUH colorectal cancer patients

Report type	N_{report}	N_{crc}	PPV	NPV	Sensitivity	Specificity
Training data						
Pathology	200	99	94.0 (87.5, 97.2)	95.0 (88.8, 97.8)	94.9 (88.7, 97.8)	94.1 (87.6, 97.2)
Imaging	200	95	90.0 (82.6, 94.5)	95.0 (88.8, 97.8)	94.7 (88.3, 97.7)	90.5 (83.4, 94.7)
Future test data						
Pathology	200	109	92.0 (85.0, 95.9)	83.0 (74.5, 89.1)	84.4 (76.4, 90.0)	91.2 (83.6, 95.5)
Imaging	200	85	78.0 (68.9, 85.0)	93.0 (86.3, 96.6)	91.8 (84.0, 96.0)	80.9 (72.7, 87.0)

Notes. N_{crc} - number of reports describing primary colorectal cancer, PPV - positive predictive value, NPV - negative predictive value. OUH - Oxford University Hospitals. 95% Wilson confidence intervals are shown in brackets.

than 95%, with the exception of T-stages in pathology reports (90.0%) and M stages in imaging reports (83.7%).

The lower sensitivity for T staging in pathology reports (90.0%) was due to cases where the absence of tumour could be inferred (e.g. phrases such as 'colonic biopsy - within normal limits'), but where T stage was not explicitly reported as 'T0' (ignoring these 10 cases would yield a 100% sensitivity for pathological T stage). Similarly, reduced sensitivity for M stage in imaging reports (83.7%) was mainly due to cases where the absence of metastasis could be inferred but was not written as 'M0' (ignoring these 6 cases would yield a sensitivity of 97.7%). Some other errors were due to inconsistency: pathology reports sometimes contained sentences that reported a single T-stage value, such as 'tumour perforation (T4b): yes', but the final T stage that was reported in report summary was sometimes different (e.g. 4a or 4); in these cases, the value of the summary stage was preferred as the ground truth maximum value. A few false positives were also due to some keywords not being included in the regex-based filter for single T-stage values (e.g. 'T1 weighing' was not included in the filter but 'T1 weighted' was). There were too few examples of Kikuchi (8), Haggit (0) and tumour grade (2) to reliably evaluate the performance on these categories.

The main values under each TNM category were generally detected equally well, with a few exceptions discussed above (e.g. lower sensitivity for T0 and M0; Appendix A.1).

Overall, this shows that the code performed very well for extracting the main numerically reported TNM staging values (the task it was designed for), even though TNM stage values are missed when these are not explicitly reported but can be inferred from text.

The main TNM staging algorithm was somewhat slow, possibly due to the variable-length look-behind regular expressions that helped constrain the context of TNM staging values (it ran approximately 1.42 minutes per 400 clinical reports). We also explored the performance of a simpler algorithm that does not initially constrain the context of values that are extracted (but does it later when extracting values from the retrieved TNM phrases). The simpler code ran roughly 4x faster (approximately 0.35 minutes per 400 reports), and still achieved high PPV (>92%) and sensitivity (>86%, Table 3.8), although it was not as precise as the main algorithm.

Note that the algorithm was designed by iterating over a large number of clinical reports (section 3.2.6), so it is unlikely to be specific to the random sample that was selected for evaluation, and hence unlikely to be overly optimistic. An additional evaluation on future OUH reports that are completely unseen will be included in a published version of this chapter.

Table 3.7: Performance of a TNM stage algorithm for detecting the maximum value of each TNM category in the clinical reports of OUH colorectal cancer patients

TNM category	N_{report}	N_{value}	$\text{PPV}_{\text{micro}}$	NPV	$\text{Sensitivity}_{\text{micro}}$	Specificity
Pathology reports - training data						
Tpre	200	99	100.0 (96.3, 100.0)	100.0 (96.3, 100.0)	100.0 (96.3, 100.0)	100.0 (96.3, 100.0)
T	200	102	99.0 (94.6, 99.8)	90.0 (82.6, 94.5)	90.0 (83.0, 94.3)	100.0 (95.9, 100.0)
N	200	88	100.0 (95.8, 100.0)	100.0 (96.7, 100.0)	100.0 (95.8, 100.0)	100.0 (96.7, 100.0)
M	200	72	100.0 (94.9, 100.0)	100.0 (97.1, 100.0)	100.0 (94.9, 100.0)	100.0 (97.1, 100.0)
V	200	91	100.0 (95.9, 100.0)	100.0 (96.6, 100.0)	100.0 (95.9, 100.0)	100.0 (96.6, 100.0)
R	200	91	98.9 (94.0, 99.8)	100.0 (98.9, 100.0)	98.9 (94.0, 99.8)	100.0 (96.6, 100.0)
L	200	91	100.0 (95.9, 100.0)	100.0 (96.6, 100.0)	100.0 (95.9, 100.0)	100.0 (96.6, 100.0)
Pn	200	74	100.0 (94.9, 100.0)	98.4 (94.5, 100.0)	97.3 (90.7, 99.3)	100.0 (97.0, 100.0)
Kikuchi	200	8	100.0 (67.6, 100.0)	100.0 (98.0, 100.0)	100.0 (67.6, 100.0)	100.0 (98.0, 100.0)
Haggitt	200	0	-	100 (98.1, 100.0)	-	100.0 (98.1, 100.0)
G	200	2	100.0 (32.2, 100.0)	100.0 (98.1, 100.0)	100.0 (32.2, 100.0)	100.0 (98.1, 100.0)
Imaging reports - training data						
Tpre	200	4	100.0 (51.0, 100.0)	100.0 (98.1, 100.0)	100.0 (51.0, 100.0)	100.0 (98.1, 100.0)
T	200	100	98.0 (93.0, 99.4)	95.0 (88.8, 97.8)	95.1 (89.1, 97.9)	97.9 (92.8, 99.4)
N	200	90	97.7 (92.1, 99.4)	97.3 (92.4, 99.1)	95.6 (89.1, 98.3)	99.1 (95.0, 99.8)
M	200	43	100.0 (90.4, 100.0)	95.7 (91.5, 97.9)	83.7 (70.0, 91.9)	100.0 (97.6, 100.0)
V	200	27	100.0 (87.5, 100.0)	100.0 (97.8, 100.0)	100.0 (87.5, 100.0)	100.0 (97.8, 100.0)

Continued on next page

Table 3.7 – continued from previous page

TNM category	N_{report}	N_{value}	$\text{PPV}_{\text{micro}}$	NPV	$\text{Sensitivity}_{\text{micro}}$	Specificity

Notes. N_{value} is the number of reports that contains information about a TNM category (for example, a T staging value for the T category). $\text{PPV}_{\text{micro}}$ is the micro-average of positive predictive values for detecting each TNM value when it was present; NPV is the negative predictive value for correctly detecting that no TNM values were present; $\text{Sensitivity}_{\text{micro}}$ is the micro-average of sensitivities for detecting each TNM value when it was present. 95% Wilson confidence intervals are shown in brackets.

Table 3.8: Performance of a **simplified** TNM stage algorithm for detecting the maximum value of each TNM category in the clinical reports of OUH colorectal cancer patients

TNM category	N_{report}	N_{value}	$\text{PPV}_{\text{micro}}$	NPV	$\text{Sensitivity}_{\text{micro}}$	Specificity
Pathology reports - training data						
Tpre	200	99	100.0 (96.2, 100.0)	99.0 (94.7, 99.8)	99.0 (94.5, 99.8)	100.0 (96.3, 100.0)
T	200	102	97.0 (91.6, 99.0)	90.9 (83.6, 95.1)	89.1 (81.9, 93.6)	100.0 (95.9, 100.0)
N	200	88	100.0 (95.8, 100.0)	100.0 (96.7, 100.0)	100.0 (95.8, 100.0)	100.0 (96.7, 100.0)
M	200	72	92.0 (83.6, 96.3)	100.0 (97.0, 100.0)	95.8 (88.5, 98.6)	97.7 (93.3, 99.2)
V	200	91	98.9 (94.0, 99.8)	100.0 (96.6, 100.0)	98.9 (94.0, 99.8)	100.0 (96.6, 100.0)
R	200	91	95.7 (89.3, 98.3)	100.0 (96.6, 100.0)	96.7 (90.8, 98.9)	99.1 (95.0, 99.8)
L	200	91	100.0 (95.9, 100.0)	100.0 (96.6, 100.0)	100.0 (95.9, 100.0)	100.0 (96.6, 100.0)
Pn	200	74	98.6 (92.6, 99.8)	98.4 (94.4, 99.6)	97.3 (90.7, 99.3)	99.2 (95.6, 99.9)
Kikuchi	200	8	100.0 (64.6, 100.0)	99.5 (97.1, 99.9)	87.5 (52.9, 97.8)	100.0 (98.0, 100.0)
Haggitt	200	0	-	100 (98.0, 100.0)	-	96.5 (93.0, 98.3)
G	200	2	33.3 (9.7, 70.0)	100.0 (98.1, 100.0)	100.0 (34.2, 100.0)	98.0 (94.4, 99.2)
Imaging reports - training data						
Tpre	200	4	40.0 (16.8, 68.7)	100.0 (98.0, 100.0)	100.0 (51.0, 100.0)	96.9 (93.5, 98.6)
T	200	100	95.1 (89.0, 97.9)	94.9 (88.6, 97.8)	94.2 (87.9, 97.3)	95.9 (89.9, 98.4)
N	200	90	97.7 (92.1, 99.4)	97.3 (92.4, 99.1)	95.6 (89.1, 98.3)	99.1 (95.0, 99.8)
M	200	43	97.4 (86.5, 99.5)	96.3 (92.2, 98.3)	86.0 (72.7, 93.4)	99.4 (96.5, 99.9)
V	200	27	100.0 (87.5, 100.0)	100.0 (97.8, 100.0)	100.0 (87.5, 100.0)	100.0 (97.8, 100.0)

Notes. N_{value} is the number of reports that contains information about a TNM category (for example, a T staging value for the T category). $\text{PPV}_{\text{micro}}$ is the micro-average of positive predictive values for detecting each TNM value when it was present; NPV is the negative predictive value for correctly detecting that no TNM values were present; $\text{Sensitivity}_{\text{micro}}$ is the micro-average of sensitivities for detecting each TNM value when it was present. 95% Wilson confidence intervals are shown in brackets.

3.3.3 Recurrence and metastasis

The performance of transformers and a regex-based model for detecting the presence of recurrence/metastasis (present, possible) and its anatomical site (abdominopelvic, distant, liver, lung) is given in Table 3.11. Confusion matrices for one of the best parameter-efficient transformer models (BioClinicalBERT-lora) and a regex-based model are given in Table 3.9 (for presence of recurrence/metastasis), and in Table 3.10 (for anatomical site). The models were evaluated on short extracts of imaging and histopathology reports that had been set aside as a test-set and not used during the development of the transformer models (188 pathology and 282 imaging report extracts); each extract contained a keyword that described

recurrence/metastasis together with surrounding text (see Section 3.2.8). The performance for detecting anatomical site was evaluated only on extracts that described recurrence/metastasis in a present or possible sense.

Note that this is a more conservative way of evaluating performance: a model does not have to do well on all report extracts to correctly detect that a clinical report (that consists of multiple extracts) mentions recurrence in an affirmative sense and in a certain anatomical site. In subsequent work, it could be useful to evaluate the models at the report (rather than report extract) level, for which the current held-out report extract dataset was too small.

Detecting the presence of recurrence

When evaluated on histopathology report extracts, the regex-based model had 84.8% sensitivity and 86.4% PPV for detecting the presence of recurrence/metastasis. The BioClinicalBERT-lora, DistilBERT-full, and DistilBERT-lora models performed similarly to the regex-based model (sensitivities ranged from 86.7% to 87.6% and PPVs from 84.3% to 86.8%, Table 3.11). The full BioClinicalBERT model had the highest sensitivity (94.3%), but somewhat lower PPV (81.1%).

On imaging reports, the regex-based model had lower sensitivity (81.6%) and PPV (82.4%) than the transformer models (their sensitivities ranged from 89.3% to 95.1%, and PPVs from 83.1% to 89.7%, Table 3.11). The BioClinicalBERT-lora model had one of the highest point estimate for sensitivity (93.2%), and highest PPV (89.7%).

There were only six extracts from histopathology reports that described recurrence/metastasis in a possible rather than affirmative sense, and hence there was not enough data to evaluate the performance for identifying that category. In imaging reports, the regex-based model had 63.6% sensitivity and 67.7% PPV, while the transformer models had sensitivities in the range of 33.3% to 63.6% and PPVs in 72.0% to 88.9%. The best performing model can be considered the BioClinicalBERT-lora, as it had one of the highest sensitivities (63.6%) and PPVs (77.8%). This comparison is again limited due to there being only 33 imaging report

extracts that expressed suspicion about recurrence/metastasis, and hence there was more uncertainty in performance estimates (wide confidence intervals).

A confusion matrix for one of the best performing, parameter-efficient transformer models (BioClinicalBERT-lora) showed that on histopathology reports, it mostly confused the 'present' category with 'other' (3.9. Further error analysis showed that when it incorrectly predicted recurrence to be 'present', then recurrence was often described in a general sense (e.g. 'management of patients with metastasis'). The regex-based model also mostly confused 'present' with 'other', but in this case the false positive report extracts often discussed recurrence/metastasis in a historical sense (e.g. 'scan in <month> <year> demonstrated <condition> and ... metastases'). In imaging reports, the transformer model more often confused the 'present' category with 'possible', such that in 8 out of 12 times when it falsely predicted recurrence to be 'present', it was actually discussed in a 'possible' sense. On the other hand, when the regex-based model falsely predicted recurrence to be present, it more often belonged to the 'other' category.

Overall, the regex-based model achieved at least 84% sensitivity and PPV on pathology report extracts, and at least 81% sensitivity and PPV on imaging report extracts. In pathology reports, the regex-based algorithm was competitive with transformers, whereas in imaging reports its performance estimates were lower - there, 3 out of 4 transformer models had sensitivities greater than 93% with PPVs at least as good or higher as the regex-based algorithm.

Detecting the anatomical site of recurrence

In pathology report extracts, the regex-based algorithm had 73.1% sensitivity and 71.7% PPV for detecting that recurrence/metastasis occurred in an abdominopelvic anatomical site, 64.2% sensitivity and 72.3% PPV for distant sites, and 52.1% sensitivity and 96.2% PPV for liver (Table 3.11). (There were not enough pathology report extracts that discussed recurrence/metastasis in lungs to evaluate performance for that site.). All transformer models had higher sensitivities and PPVs for the abdominopelvic and distant sites, and for liver; in most cases the point estimate

Table 3.9: Confusion matrices of the BioClinicalBERT-lora model and a regex-based model for detecting the assertion status of recurrence/metastasis in clinical report extracts

		Predicted status		
Model	True status	Present	Possible	Other
Held-out* pathology report extracts				
BioClinicalBERT-lora	Present	91	3	11
BioClinicalBERT-lora	Possible	2	4	0
BioClinicalBERT-lora	Other	15	4	58
Regex-based	Present	89	5	11
Regex-based	Possible	1	5	0
Regex-based	Other	13	2	62
Held-out imaging report extracts				
BioClinicalBERT-lora	Present	96	4	3
BioClinicalBERT-lora	Possible	8	21	4
BioClinicalBERT-lora	Other	3	2	141
Regex-based	Present	84	4	15
Regex-based	Possible	7	21	5
Regex-based	Other	11	6	129

Notes. *There may be some optimism in the performance of the regex-based model, because it was developed on nearly all clinical reports available to us (except the newer OUH reports), some of which were included in the dataset that served as the held-out test set for the BioClinicalBERT-lora model; however the regex-based model was not specifically tuned to perform well on that subset of reports and so its optimism is likely small. The BioClinicalBERT-lora model predicts values in ['present', 'possible', 'negated', 'historic', 'other'], but to make comparisons to the regex-based model easier, we combined 'negated', 'historic', and 'other' under the 'other' label.

for sensitivity was at least 10 percentage points higher, and PPV was similar or higher. For example, the BioClinicalBERT-lora model had 88.5% sensitivity and PPV for detecting the abdominopelvic site; 83.0% sensitivity and 78.6% PPV for distant site; and 75.0% sensitivity and 100.0% PPV for liver (Table 3.11).

In imaging report extracts, the regex-based algorithm had 76.1% sensitivity and 92.1% PPV for abdominopelvic sites, 81.1% sensitivity and 93.6% PPV for distant sites, 76.3% sensitivity and 100% PPV for liver, and 69.7% sensitivity and 95.8% PPV for lung (Table 3.11). All transformer models had higher estimates of sensitivity for distant site (>85.6%), liver (>84.7%), and lung (>78.8%), but not consistently higher estimates for the abdominopelvic site where sensitivities ranged from 69.6% to 80.4%. The PPVs of transformers ranged from 88.5%-91.9%

for distant site, 81.0%-87.9% for liver, 89.7%-96.% for lung, and 84.2% to 90.2% for abdominopelvic site (Table 3.11).

Confusion matrices showed that the regex-based model tended to confuse abdominopelvic and distant sites in pathology report extracts (Table 3.10. Some of these errors were due to pathology reports discussing both the site of distant metastasis and the colorectal (abdominopelvic) tumour it originated from in the same report (e.g. a report may discuss the metastasis site earlier in the report and later say 'metastatic colorectal cancer'); it was harder to capture cases such as this in the regex-based algorithm.

In summary, the regex-based model had somewhat low sensitivities on pathology reports (52.1% - 73.1%), especially for detecting distant sites in general and liver in particular. The BioClinicalBERT models had sensitivities at least 10 percentage points higher with similar or better PPVs, potentially because they were better at disambiguating the locoregional and distant sites that can both be mentioned in a pathology report. In imaging reports, the rule based models had somewhat higher sensitivities (69.7% - 81.1%), although the transformers still had higher estimates of sensitivity, especially for distant sites (although coupled with lower PPV).

Comparison of full and parameter-efficient transformer models

The base transformer models (BioClinicalBERT and DistilBERT) were fine-tuned for recurrence detection using two methods: tuning all parameters of the model ('-full' in Table 3.11), or representing the updates to each parameter matrix as a product of two smaller matrices (the low rank adaptation method or LoRA [93], '-lora' in Table 3.11). LoRA greatly reduces the number of parameters that need to be tuned and allows for more efficient use of memory when creating multiple versions of each base model (Section 3.2.5).

There was no obvious pattern for whether the full version of each base model performed better than its LoRA counterpart. For example, the BioClinicalBERT-full had higher point estimates of sensitivity than BioClinicalBERT-lora for all classification tasks in pathology reports (the estimates were about 1.9 to 7.6

Table 3.10: Confusion matrices of the BioClinicalBERT-lora model and a regex-based model for detecting the broad anatomical site of recurrence/metastasis in report extracts

			Predicted site	
Model	True site	Abdominopelvic	Distant	Other
Held-out* pathology report extracts				
BioClinicalBERT-lora	Abdominopelvic	46	2	4
BioClinicalBERT-lora	Distant	3	43	7
BioClinicalBERT-lora	Other	3	3	0
Regex-based	Abdominopelvic	38	10	4
Regex-based	Distant	12	34	7
Regex-based	Other	3	3	0
Held-out imaging report extracts				
BioClinicalBERT-lora	Abdominopelvic	34	9	3
BioClinicalBERT-lora	Distant	6	80	4
BioClinicalBERT-lora	Other	0	0	0
Regex-based	Abdominopelvic	35	5	6
Regex-based	Distant	3	73	14
Regex-based	Other	0	0	0

Notes. *There may be some optimism in the performance of the regex-based model, because it was developed on nearly all clinical reports available to us (except the newer OUH reports), some of which were included in the dataset that served as the held-out test set for the BioClinicalBERT-lora model; however the regex-based model was not specifically tuned to perform well on that subset of reports and so its optimism is likely small. The BioClinicalBERT-lora model predicts values in ['present', 'possible', 'negated', 'historic', 'other'], but to make comparisons to the regex-based model easier, we combined 'negated', 'historic', and 'other' under the 'other' label. BioClinicalBERT-lora is the BioClinicalBERT model fine-tuned using the Low Rank Adaptation method.

percentage points higher), but estimates of its PPV were also lower in 3 out of 4 tasks, and the pattern of higher sensitivities was not evident in imaging reports (Table 3.11). In general, one would expect the full models to do better or equally well as LoRA, hoping that the reduced number of parameters in LoRA would not deteriorate performance too much; however, the lack of this pattern could also be because our model development dataset was relatively small (1,356 examples), so the full models may have been more likely to overfit, and there may not have been enough statistical power to detect a performance difference on the held-out dataset (470 examples).

Table 3.11: Performance of fine-tuned transformer models and a regex-based algorithm for classifying clinical report extracts that describe recurrence or metastasis

Task	Model	N_{phrase}	N_{positive}	PPV	NPV	Sensitivity	Specificity
Held-out* pathology report extracts							
Status: present	BioClinicalBERT-full	188	105	81.1 (73.3, 87.1)	90.9 (81.6, 95.8)	94.3 (88.1, 97.4)	72.3 (61.8, 80.8)
Status: present	BioClinicalBERT-lora	188	105	84.3 (76.2, 89.9)	82.5 (72.7, 89.3)	86.7 (78.9, 91.9)	79.5 (69.6, 86.8)
Status: present	DistilBERT-full	188	105	86.8 (79.0, 92.0)	84.1 (74.7, 90.5)	87.6 (80.0, 92.6)	83.1 (73.7, 89.7)
Status: present	DistilBERT-lora	188	105	84.3 (76.2, 89.9)	82.5 (72.7, 89.3)	86.7 (78.9, 91.9)	79.5 (69.6, 86.8)
Status: present	Regex-based	188	105	86.4 (78.5, 91.7)	81.2 (71.6, 88.1)	84.8 (76.7, 90.4)	83.1 (73.7, 89.7)
Status: possible	-	188	6	-	-	-	-
Site: abdominopelvic	BioClinicalBERT-full	111	52	85.7 (74.3, 92.6)	92.7 (82.7, 97.1)	92.3 (81.8, 97.0)	86.4 (75.5, 93.0)
Site: abdominopelvic	BioClinicalBERT-lora	111	52	88.5 (77.0, 94.6)	89.8 (79.5, 95.3)	88.5 (77.0, 94.6)	89.8 (79.5, 95.3)
Site: abdominopelvic	DistilBERT-full	111	52	91.8 (80.8, 96.8)	88.7 (78.5, 94.4)	86.5 (74.7, 93.3)	93.2 (83.8, 97.3)
Site: abdominopelvic	DistilBERT-lora	111	52	88.9 (76.5, 95.2)	81.8 (70.9, 89.3)	76.9 (63.9, 86.3)	91.5 (81.6, 96.3)
Site: abdominopelvic	Regex-based	111	52	71.7 (58.4, 82.0)	75.9 (63.5, 85.0)	73.1 (59.7, 83.2)	74.6 (62.2, 83.9)
Site: distant	BioClinicalBERT-full	111	53	91.7 (80.4, 96.7)	85.7 (75.0, 92.3)	83.0 (70.8, 90.8)	93.1 (83.6, 97.3)
Site: distant	BioClinicalBERT-lora	111	53	89.6 (77.8, 95.5)	84.1 (73.2, 91.1)	81.1 (68.6, 89.4)	91.4 (81.4, 96.3)
Site: distant	DistilBERT-full	111	53	82.7 (70.3, 90.6)	83.1 (71.5, 90.5)	81.1 (68.6, 89.4)	84.5 (73.1, 91.6)
Site: distant	DistilBERT-lora	111	53	78.6 (66.2, 87.3)	83.6 (71.7, 91.1)	83.0 (70.8, 90.8)	79.3 (67.2, 87.7)
Site: distant	Regex-based	111	53	72.3 (58.2, 83.1)	70.3 (58.2, 80.1)	64.2 (50.7, 75.7)	77.6 (65.3, 86.4)
Site: liver	BioClinicalBERT-full	111	48	97.6 (87.4, 99.6)	88.6 (79.0, 94.1)	83.3 (70.4, 91.3)	98.4 (91.5, 99.7)
Site: liver	BioClinicalBERT-lora	111	48	100.0 (90.4, 100.0)	84.0 (74.1, 90.6)	75.0 (61.2, 85.1)	100.0 (94.3, 100.0)
Site: liver	DistilBERT-full	111	48	94.7 (82.7, 98.5)	83.6 (73.4, 90.3)	75.0 (61.2, 85.1)	96.8 (89.1, 99.1)
Site: liver	DistilBERT-lora	111	48	95.5 (84.9, 98.7)	91.0 (81.8, 95.8)	87.5 (75.3, 94.1)	96.8 (89.1, 99.1)
Site: liver	Regex-based	111	48	96.2 (81.1, 99.3)	72.9 (62.7, 81.2)	52.1 (38.3, 65.5)	98.4 (91.5, 99.7)
Site: lung	-	111	1	-	-	-	-
Held-out imaging report extracts							
Status: present	BioClinicalBERT-full	282	103	83.1 (75.3, 88.8)	97.0 (93.1, 98.7)	95.1 (89.1, 97.9)	88.8 (83.4, 92.6)
Status: present	BioClinicalBERT-lora	282	103	89.7 (82.5, 94.2)	96.0 (92.0, 98.0)	93.2 (86.6, 96.7)	93.9 (89.3, 96.5)
Status: present	DistilBERT-full	282	103	86.0 (78.2, 91.3)	93.7 (89.1, 96.5)	89.3 (81.9, 93.9)	91.6 (86.6, 94.9)
Status: present	DistilBERT-lora	282	103	83.1 (75.3, 88.8)	97.0 (93.1, 98.7)	95.1 (89.1, 97.9)	88.8 (83.4, 92.6)
Status: present	Regex-based	282	103	82.4 (73.8, 88.5)	89.4 (84.1, 93.1)	81.6 (73.0, 87.9)	89.9 (84.7, 93.5)
Status: possible	BioClinicalBERT-full	282	33	88.9 (67.2, 96.9)	93.6 (89.9, 95.9)	48.5 (32.5, 64.8)	99.2 (97.1, 99.8)
Status: possible	BioClinicalBERT-lora	282	33	77.8 (59.2, 89.4)	95.3 (92.0, 97.3)	63.6 (46.6, 77.8)	97.6 (94.8, 98.9)

Continued on next page

Table 3.11 – continued from previous page

Task	Model	N_{phrase}	N_{positive}	PPV	NPV	Sensitivity	Specificity
Status: possible	DistilBERT-full	282	33	72.0 (52.4, 85.7)	94.2 (90.6, 96.4)	54.5 (38.0, 70.2)	97.2 (94.3, 98.6)
Status: possible	DistilBERT-lora	282	33	78.6 (52.4, 92.4)	91.8 (87.9, 94.5)	33.3 (19.8, 50.4)	98.8 (96.5, 99.6)
Status: possible	Regex-based	282	33	67.7 (50.1, 81.4)	95.2 (91.8, 97.2)	63.6 (46.6, 77.8)	96.0 (92.8, 97.8)
Site: abdominopelvic	BioClinicalBERT-full	136	46	84.6 (70.3, 92.8)	86.6 (78.4, 92.0)	71.7 (57.5, 82.7)	93.3 (86.2, 96.9)
Site: abdominopelvic	BioClinicalBERT-lora	136	46	85.0 (70.9, 92.9)	87.5 (79.4, 92.7)	73.9 (59.7, 84.4)	93.3 (86.2, 96.9)
Site: abdominopelvic	DistilBERT-full	136	46	90.2 (77.5, 96.1)	90.5 (83.0, 94.9)	80.4 (66.8, 89.3)	95.6 (89.1, 98.3)
Site: abdominopelvic	DistilBERT-lora	136	46	84.2 (69.6, 92.6)	85.7 (77.4, 91.3)	69.6 (55.2, 80.9)	93.3 (86.2, 96.9)
Site: abdominopelvic	Regex-based	136	46	92.1 (79.2, 97.3)	88.8 (81.0, 93.6)	76.1 (62.1, 86.1)	96.7 (90.7, 98.9)
Site: distant	BioClinicalBERT-full	136	90	88.8 (80.5, 93.8)	76.6 (62.8, 86.4)	87.8 (79.4, 93.0)	78.3 (64.4, 87.7)
Site: distant	BioClinicalBERT-lora	136	90	89.9 (81.9, 94.6)	78.7 (65.1, 88.0)	88.9 (80.7, 93.9)	80.4 (66.8, 89.3)
Site: distant	DistilBERT-full	136	90	91.9 (84.1, 96.0)	78.0 (64.8, 87.2)	87.8 (79.4, 93.0)	84.8 (71.8, 92.4)
Site: distant	DistilBERT-lora	136	90	88.5 (80.1, 93.6)	73.5 (59.7, 83.8)	85.6 (76.8, 91.4)	78.3 (64.4, 87.7)
Site: distant	Regex-based	136	90	93.6 (85.9, 97.2)	70.7 (58.0, 80.8)	81.1 (71.8, 87.9)	89.1 (77.0, 95.3)
Site: liver	BioClinicalBERT-full	136	59	87.9 (77.1, 94.0)	89.7 (81.0, 94.7)	86.4 (75.5, 93.0)	90.9 (82.4, 95.5)
Site: liver	BioClinicalBERT-lora	136	59	81.0 (69.6, 88.8)	89.0 (79.8, 94.3)	86.4 (75.5, 93.0)	84.4 (74.7, 90.9)
Site: liver	DistilBERT-full	136	59	87.7 (76.8, 93.9)	88.6 (79.7, 93.9)	84.7 (73.5, 91.8)	90.9 (82.4, 95.5)
Site: liver	DistilBERT-lora	136	59	87.9 (77.1, 94.0)	89.7 (81.0, 94.7)	86.4 (75.5, 93.0)	90.9 (82.4, 95.5)
Site: liver	Regex-based	136	59	100.0 (92.1, 100.0)	84.6 (75.8, 90.6)	76.3 (64.0, 85.3)	100.0 (95.2, 100.0)
Site: lung	BioClinicalBERT-full	136	33	96.3 (81.7, 99.3)	93.6 (87.3, 96.9)	78.8 (62.2, 89.3)	99.0 (94.7, 99.8)
Site: lung	BioClinicalBERT-lora	136	33	96.7 (83.3, 99.4)	96.2 (90.7, 98.5)	87.9 (72.7, 95.2)	99.0 (94.7, 99.8)
Site: lung	DistilBERT-full	136	33	89.7 (73.6, 96.4)	93.5 (87.1, 96.8)	78.8 (62.2, 89.3)	97.1 (91.8, 99.0)
Site: lung	DistilBERT-lora	136	33	93.1 (78.0, 98.1)	94.4 (88.3, 97.4)	81.8 (65.6, 91.4)	98.1 (93.2, 99.5)
Site: lung	Regex-based	136	33	95.8 (79.8, 99.3)	91.1 (84.3, 95.1)	69.7 (52.7, 82.6)	99.0 (94.7, 99.8)

Notes. *The clinical report extracts were randomly divided into a model development set (75%) and held-out test set (25%); the BioClinicalBERT and DistilBERT transformer models were developed on the model development set and their performance is evaluated on the held-out test set. There may be some optimism in the performance of the regex-based model, because it was developed on nearly all available clinical reports (except newer OUH reports), some of which were included in the dataset that served as the held-out test set for the transformer models. However the regex-based model was not specifically tuned to perform well on that subset of reports and hence its optimism is probably small. In names of transformer models, '-full' means that all parameters of the base model were fine-tuned; '-lora' means that the Low Rank Adaptation method was used for fine-tuning the parameters. N_{phrase} is the number of phrases that contain recurrence or metastasis keywords, N_{positive} is the number of phrases that were positive examples for a given task. PPV - positive predictive value; NPV - negative predictive value. 95% Wilson confidence intervals are shown in brackets.

3.4 Discussion

3.4.1 Main findings

We developed a lightweight pipeline that extracts key information about colorectal cancer (CRC) from imaging and histopathology reports contained in electronic health records and thus facilitates translational research. The CRC detection component helps identify clinical reports that discuss current primary CRC, which could be used for TNM stage identification and for identifying date of diagnosis. The TNM stage extraction algorithm is essential for describing the severity of CRC and allows to select patients with a similar disease profile for subsequent analyses, such as for comparative treatment effectiveness studies. Finally, the recurrence and metastasis detection models provide crucial outcome variables for many studies that could be done on high-resolution real-world CRC data, and facilitate research into metastatic CRC in general.

The CRC detection module performed well, especially on pathology reports (sensitivity near 94%, PPV near 92%), but the results may need to be double checked on new samples of reports, and especially on imaging reports where performance was lower. The pipeline facilitates this by outputting all extracted tumour keywords with surrounding text, which can then be accepted or rejected.

The TNM stage detection algorithm had excellent performance: for the main T/N/M categories, PPV was >97% in both imaging and pathology reports; sensitivity was near 100% in pathology reports and at least 95% in imaging reports for numerically reported T/N/M values (i.e. when excluding reports where T/N/M categories could be inferred from text, otherwise it dropped to 90% for T categories in pathology reports and to 84% for M categories in imaging reports). The TNM stage detection code is probably the most useful contribution of this chapter. This is because TNM staging captures essential information about the status of cancer and is reported in specific ways using letters and numbers, making it likely that the current algorithm will generalise to clinical reports from other medical centres. However, its usefulness depends on two key factors: (1) do the clinical reports

usually report TNM staging in their customary letters and numbers (the current algorithm cannot infer TNM staging if it is not explicitly given), and (2) do the reports usually give more than one TNM staging value in a sequence (e.g. 'T0 N0') as the performance for detecting single TNM values is dependent on a regex-based algorithm for disambiguating its context and may be less generalisable. Note that the TNM algorithm differs from CanStaging+ [104], a staging tool used by cancer registries: the algorithm automatically extracts staging from free text if it is given in letters and numbers, whereas CanStaging+ requires user to input information about the tumour to receive a staging score.

The recurrence detection algorithms, including the regex-based model, correctly detected the presence of recurrence/metastasis most of the time: PPV and sensitivity were at least greater than 80% in report extracts that mentioned recurrence/metastasis. The performance of the regex-based algorithm was more variable for detecting the broad anatomical site (abdominopelvic/distant): in imaging reports, its sensitivity was at least 75% and PPV at least 92%, in pathology reports the sensitivity was at least 64% and PPV 71%. However, these algorithms would likely perform better when evaluated on entire reports (rather than report extracts), as a report could still be correctly classified if only one of its extracts is correctly classified. Furthermore, we showed that it is possible to create transformer models that are as good or better than the regex-based algorithm even with a relatively small dataset, which paves the way for improving these models with active learning (see below). We also note that the performance of our transformer models for detecting the assertion status of recurrence/metastasis was lower than of the assertion status models developed by van Aken et al [86]. However, van Aken et al fine-tuned their models to distinguish only the present/possible/absent categories, so it is likely that their models would make errors in the current dataset that also contains general statements (e.g. 'in patients with metastasis') and historical statements ('metastasis diagnosed in <year>'). Indeed, when such statements are entered into their demo application [105], they are marked as 'present' for metastasis, which would count as an error in our study.

This chapter presented a partial evaluation of these algorithms: to publish the TNM stage extraction results, these should additionally be evaluated on a future sample of OUH reports (the reports have already been selected for annotation, but the work has been much slower than anticipated). The recurrence algorithm could also be further evaluated at the report level (by using an additional set of annotated reports which are also already selected). However, the recurrence models are less likely generalisable to a variety of clinical reports, and are probably most useful for research that would utilise the NIHR HIC CRC database. As the development of that database has slowed down, it would be rational to revisit these algorithms if there is progress on new data being included.

3.4.2 Limitations

There are several limitations of our information extraction pipeline that could be addressed in future work.

1. *No spell checking.* If the clinical reports contain a significant number of spelling errors for tumour-related keywords, anatomical sites, and recurrence/metastasis keywords, then the algorithms will perform worse. A potential initial solution is to check for spelling mistakes semi-manually by first identifying all edits of the relevant keywords that exist in the clinical reports by using Peter Norvig's spell checking functions [106], and then correcting the edits that indeed represent spelling errors.

2. *Regex-based algorithms may not generalise to different medical reports.* The CRC detection algorithm, the regex-based model for detecting recurrence/metastasis, and the single TNM value detection module in the TNM stage extraction algorithm all use specific patterns on the left and right side of each extracted keyword to identify its context. These patterns may not generalise equally well to medical reports from other centres if these use language differently (e.g. abbreviations, report structure, commonly used expressions). On the other hand, if the code is run on a new set of reports, all keywords extracted by the

regex-based algorithms can be examined together with their surrounding text to have an initial sense of performance. Furthermore, the method of detecting TNM sequences should generalise better, as it relies only on TNM-like values occurring near each other to correctly identify these as TNM values.

3. *Only explicitly given TNM stages were extracted.* If the clinical reports do not report TNM staging in customary letters and numbers, but nevertheless contain enough information about the tumour to infer the TNM stage, then the current code will not work for these reports.

4. *The TNM stage algorithm was derived on clinical reports of individuals with CRC.* It is therefore not clear if it would work equally well for other cancer types. However, as TNM stages are generally written in a similar way with letters and numbers, and as the algorithm accommodates a variety of ways these can be reported, it is likely that it can generalise well, especially if the new reports contain sequences of TNM values rather than single values (that are harder to disambiguate).

5. *The recurrence/metastasis detection algorithms were only evaluated at the report extract level.* It is desirable to know how well the algorithms would do when evaluated on entire reports, as a report may contain multiple extracts, some of which are easier to classify. Initial evaluations indicated that the regex-based algorithm performed better on entire reports (data not shown); however rigorous evaluation would require annotating additional reports. The additional evaluation could also be used to study whether there is any difference in performance for recurrence-related keywords and metastasis keywords. On the other hand, it was encouraging to see that the transformer-based models performed similarly or better than the regex-based algorithm, because it could be easier to improve their accuracy using active learning (see below), compared to manually reviewing errors and adjusting the rules of the regex-based algorithm.

6. *The recurrence detection algorithms only identify clinical reports that describe recurrence.* For detecting recurrences more comprehensively, other approaches can be used alongside, such as identifying a disease-free period after treatment and looking for evidence of recurrence thereafter using diagnosis and treatment records [73, 75, 76]. Data from all events that indicate recurrence can then be integrated, for example, by choosing the date of recurrence as the date of the earliest diagnosis code, procedure code or clinical report that indicated colorectal cancer after a disease-free period.

7. *Algorithms were mostly evaluated on OUH data, and did not always include newer, future OUH reports for additional validation.* The CRC and TNM algorithms were evaluated only on OUH data, even though they were developed using more data sources, such as clinical reports from RMH and Imperial. We focussed on OUH data, because we could obtain a newer sample of reports for further validating the models, which was currently not possible for RMH and Imperial data (see Section 3.2.7). The NIHR HIC CRC project is also waiting to receive data from new participating research centres - if this happens during the first months of 2024, then these could be used to additionally evaluate the TNM staging algorithm to make for a stronger publication; but if that does not happen soon, then the OUH data itself should also make for a sufficiently good publication.

8. *CRC detection algorithm may have excluded synchronous metastases* by excluding detected tumour keywords when the word 'metastatic' is nearby. This should be further examined.

3.4.3 Future directions

There are immediate, feasible ways to improve the algorithms:

- *The transformer models can be iteratively improved using confidence-based active learning, that helps to select the most informative examples for updating the model*[107]. Firstly, an existing model would be used to make predictions

for clinical report extracts that have not yet been annotated; a subset of report extracts whose predictions have low confidence scores (based on a pre-specified threshold) can then be selected for review, and incorrectly predicted examples can be used to fine-tune or retrain the model. Furthermore, new examples with low confidence scores should be selected from the different classes that are predicted by the model, to broaden the scope of the training dataset. In addition, attempts could be made to recalibrate the model before using its confidence scores, to ensure that these are more meaningful.

- *Adding support for parallel processing.* The TNM stage extraction algorithm, and other regex-based algorithms, could be wrapped in an external function that executes them in parallel on chunks of clinical reports, potentially greatly reducing their running time.

- *Evaluating the algorithms on additional data sources.* In particular, TNM staging and recurrence algorithms could be evaluated on newer OUH clinical reports. This would provide additional reassurance that TNM staging code works on a newer sample of data from the same medical centre, and allow to evaluate the recurrence algorithm on entire reports (not just report extracts).

It would also be possible to explore other information extraction strategies and to expand the types of data that are extracted, but this would require more effort:

- *Expanding the pipeline to cover the Royal College of Pathologists' (RCP) minimum dataset requirements*[108]. Ideally, the information extraction pipeline would detect as many data items as possible from the RCP's data specification, such as site of tumour, maximum tumour diameter, differentiation etc. This should be relatively straightforward for pathology reports that are structured according to RCP's format, but can be much harder for reports that do not follow that structure or are given as narratives. In the latter case, rules or synonyms would need to be formulated for each category of information; or a sufficient number of reports covering the variations of each category need to be annotated to train an ML model.

- *Relation extraction methods could be explored for better detection of anatomical sites of recurrence/metastasis.* Currently, the anatomical site was inferred with a simple regex-based algorithm, or a bidirectional transformer model that processed the context of each recurrence/metastasis keyword. However, there are methods that directly attempt to detect a relation between entities [109]. These may work better as the classifier is given information about both the recurrence/metastasis keyword and the anatomical site, although they would be more complex to implement.

- *Large Language Models (LLMs) could be used with zero- or few-shot prompting*[110] *to extract the same information as in our pipeline, as well as additional data items.* The author of this thesis experimented a little bit with the open-source model Mistral-7b-instruct [111], but did not observe consistent performance for detecting the assertion status of recurrence. Larger models such as GPT-4[112] could do better, especially as these have been shown to perform well on medical tasks despite not being explicitly developed for the medical domain [113]. However, the potential gains offered by LLMs may not outweigh the financial cost (GPT-4 is not free), and a potential environmental cost for using the graphical processing units that the models rely on. Furthermore, the current pipeline was meant to be easily installed and used in other hospitals (requiring the installation of only a few python packages); any pipeline involving LLMs would also be harder to set up, especially as the usage of non-open source LLMs requires guarantees that medical data is securely processed. Overall, it could be more economical to fine tune the BERT-based transformer models locally, as these may achieve desired accuracy with well-conducted additional training. LLMs could be valuable for more complex tasks instead (such as producing summaries of clinical reports) that cannot be achieved well enough with other methods.

- *The algorithms developed in this Chapter can be used alongside cancer registry data.* Firstly, the transformer-based models could be expanded to detect

TNM staging from free text when TNM staging is not given in letters and numbers. This could be fruitfully combined with the CanStaging+ [104] tool used by cancer registries: the model would automatically extract the pieces of information that the CanStaging+ tool needs to assign a TNM stage (such as level of tumour invasion). Secondly, the TNM staging scores extracted by the algorithm can be compared to the staging submitted to the cancer registry, and discrepancies can be examined to improve the algorithm and further check registry data quality. Thirdly, the TNM stage extraction tool can be used to extract staging from reports where it is already given, so that individuals who collate registry data can focus on manually processing reports where the staging is not given but could be inferred.

3.4.4 Conclusion

This chapter started with a quote from Peter Drucker, "There is nothing quite so useless, as doing with great efficiency, something that should not be done at all". Even though natural language processing pipelines are essential for converting the unstructured information in electronic health records to structured, usable data, it would be even better if the essential data items were recorded in a structured format during the original healthcare interactions. This would allow the data to be used more effectively and at scale for understanding health and improving care, although it would need to be carefully and collaboratively implemented to address the various reasons why clinicians may resort to unstructured reporting [56], and to ensure that the user interface supports the collection of high quality data [24]. This is also consistent with a growing understanding that the data stored for a single patient is not only meant to support the care of that patient, but—with appropriate anonymisation, data quality checks, opt-out policies and safeguards in place—provide insight into how the healthcare system functions and how it can be improved [114].

Everything should be made as simple as possible, but not simpler

— unknown author, derived from Albert Einstein

4

External validation of Nottingham colorectal cancer risk prediction models on the Oxford University Hospitals FIT dataset: The importance of simple baselines and appropriate evaluation metrics

Contents

4.1 Introduction

The Faecal Immunochemical Test (FIT) has high sensitivity and specificity for detecting colorectal cancer (CRC) in patients with unexplained symptoms indicative of CRC [115]. It measures the amount of blood in stool, and results of at least 10 µg of haemoglobin per gram of faeces are usually considered positive. However, approximately 5 in 6 patients with a positive FIT result do not have cancer [116]. This has prompted researchers to build prediction models that combine FIT with other routinely collected data, such as demographics, clinical symptoms, and blood test results, to reduce the number of false positives. The existing models are reviewed more thoroughly in Chapter 5, with the conclusion that none of the models has performed clearly better than the FIT test. Recently, a new set of logistic and Cox risk prediction models was developed in Nottingham as part of the NIHR-funded COLOFIT programme [117], utilising the largest dataset to date (34,231 patients). This chapter is focussed on externally validating the new Nottingham models on an Oxford dataset and discussing the evaluation of FIT-test based prediction models in more detail, as commonly used performance metrics may not be informative enough in this case. We conclude there is not enough evidence that Nottingham models do better than the FIT test in the clinically meaningful range of sensitivities, at least on the Oxford dataset. We additionally make the model evaluation code freely available, facilitating the evaluation of FIT-test based models in the future.

4.1.1 The Nottingham models for predicting colorectal cancer

The Nottingham colorectal cancer risk prediction models are logistic regression and Cox proportional hazards models that use the FIT test result, sex, age, and two blood tests—platelets and mean cell volume (MCV)—as predictor variables [118] . FIT test results and platelets are transformed using log and power transformations.

The candidate predictor variables used during model development additionally included the haemoglobin blood test, which was dropped by the variable selection algorithm. All predictor variables, except sex, were continuous. The haemoglobin, platelets, and MCV were available for at least 90% of patients in the Nottingham data. The choice of additional variables beyond FIT was informed by systematic reviews conducted as part of the wider COLOFIT programme, the clinical experience of the COLOFIT investigator team, and the proportion of missingness. The effect of each variable on the linear predictor of the models is visualised in Appendix B.1.

The models were derived by generating 10 datasets using multiple imputation and applying the multivariate fractional polynomial algorithm to select the predictor variables to be included in models and data transformations (if any) to be used for the included variables. In addition, the derivation process was repeated multiple times and model parameters averaged to obtain the bootstrap-averaged versions of the models. This resulted in four models that use the full set of predictor variables:

- Full logistic model (referred to as 'Nottingham-lr' in text),
- Full logistic model with bootstrap model averaging ('Nottingham-lr-boot'),
- Full Cox model ('Nottingham-cox'),
- Full Cox model with bootstrap model averaging ('Nottingham-cox-boot').

Three additional models were derived in Nottingham that include less predictor variables: a FIT-only model, FIT and age model, and FIT-age-sex model. These were developed using a similar process, but bootstrap-averaged versions were not

created. The simpler models help assess how much gain in predictive performance is obtained by including additional predictor variables in models.

- FIT-age-sex logistic model ('Nottingham-fit-age-sex')
- FIT-age logistic model ('Nottingham-fit-age')
- FIT-only logistic model ('Nottingham-fit')

The full Nottingham logistic and Cox models had good overall performance on the Nottingham dataset, with *c*-statistics ranging from 0.934 to 0.937; reliability diagrams indicated that the models were well calibrated on their development dataset [119]. Importantly, the most recent data reported by the Nottingham team indicates that the Cox model achieved a 39% reduction in colonoscopies at the same level of sensitivity as FIT at threshold in their internal validation data, although the reduction was less than 5% in derivation data $\geq$10[118].

4.1.2 Relevance of the Oxford dataset for external validation

The Nottingham models were developed on the data of adult patients who returned a FIT sample as ordered by their GP. The Oxford University Hospitals FIT dataset (OUH-FIT) used in this analysis represents a similar population of patients with GP-requested FITs, and thus is a relevant target population. The Nottingham and Oxford datasets are compared in more detail in the Results Table 4.3.

4.1.3 Evaluation of FIT-test based prediction models

There are at least two considerations when evaluating the diagnostic performance of prediction models that utilise the FIT test: which performance metrics to use, and how to compare the models against FIT.

Clinical prediction models are commonly evaluated by analysing discrimination, calibration and net benefit statistics [120], and these standard metrics are also included in our model evaluation pipeline for completeness. However, commonly used discrimination metrics, the *c*-statistic and ROC-curve, may not be informative

enough when the proportion of patients with a clinical outcome is small (indeed, in Oxford and Nottingham datasets the prevalence of cancer was less than 2%). ROC curves and thus *c*-statistics of two models can be very similar in this case, but one model can have a much higher positive predictive value at some classification thresholds [121]. We therefore include the precision-recall (PR) curve, which graphs the positive predictive value (PPV) against sensitivity, and the average precision metric which estimates the area under the PR-curve [122]. It is important to note, however, that *c*-statistic and average precision score summarise model performance over all possible classification thresholds, but most thresholds are less relevant. This is because the FIT test, as used in clinical practice, will capture most cancers (its sensitivity was estimated to be 89% in a meta-analysis [115]), so the classification thresholds associated with low cancer detection rate are not of equal interest. In other words, only the upper area of the ROC curve and the rightmost area of the PR curve where sensitivity is high are most relevant. In addition to reporting global performance metrics, we therefore always present the ROC and PR-curves, and tabulate metrics such as PPV and specificity at high levels of sensitivity. The discrimination metrics should also be assessed first, because a model that cannot distinguish cancers from non-cancers is useless even if it has perfect calibration.

A prediction model that utilises the FIT test should not only perform well according to performance metrics, but also perform better than the FIT test. Otherwise, there is no reason to use the model over a simpler, already implemented strategy. We propose it is important to compare models against FIT from three points of view.

1. Firstly, the goal of FIT-test based prediction models is arguably to reduce the number of false positives while capturing a similar number of cancers as FIT test thresholded at 10 µg/g, the standard testing strategy recommended by NICE [7]. The comparison can be made by evaluating the PPV of the model at the same level of sensitivity as FIT ≥ 10. If the model has higher PPV, it means that the same number of cancers are detected, but the number of false positives is smaller. In addition to comparing PPVs, one could compare

other quantities directly related to PPV: percent reduction in false positives $((1 - \text{ppv}_{model})/(1 - \text{ppv}_{fit}) * \text{ppv}_{fit}/\text{ppv}_{model}) \cdot 100)$, and percent reduction in the total number of colonoscopies $((1 - \text{ppv}_{fit}/\text{ppv}_{model}) \cdot 100)$, which were not computed in this analysis.

2. Secondly, models should be compared against FIT over all classification thresholds for a broader overview of the relative gain that models can offer. This can be done by producing ROC and PR-curves. It can also be useful to produce curves that directly show the gain in PPV relative to FIT test at each level of sensitivity (and possibly the reduction in false positives or reduction in total number of colonoscopies). This can be achieved by interpolating the PR-curves of models and FIT to the same grid of sensitivity values, using for example a method of Davis and Goadrich [121].

3. Thirdly, if decision curve analysis (DCA)[123] is performed, it is important to include a FIT-only model, not just the standard strategy of FIT thresholded at 10 µg/g. Otherwise the models are evaluated over multiple classification thresholds while FIT is evaluated at one, which can make the net benefit of models appear higher at some levels of predicted risk, but it may not be higher if FIT test was simply used at a different threshold. If a FIT-only model is included, it must be calibrated. An alternative is to include FIT test at multiple thresholds in the analysis (e.g. at 2, 5, 10, 50, 100). In other words, even if a model has higher net benefit than FIT at threshold 10, the model is unlikely to be worthwhile if it has the same net benefit as FIT test at another threshold.

   The external validation pipeline described below includes all three types of comparisons.

4.2 Methods

4.2.1 Oxford University Hospitals FIT dataset (OUH-FIT)

The OUH-FIT dataset was curated as an extension of the NIHR HIC Colorectal Cancer research database [4] in collaboration between clinical academics and data scientists based at the Oxford NIHR Health Informatics Collaborative at the Big Data Institute and the Nuffield Department of Primary Care Health Sciences at the University of Oxford. It contains data for patients who had FIT test results or colorectal cancer recorded between January 2017 and March 2022. Stool samples collected for FIT testing were analysed using the HM-JACKarc analyser (Hitachi Chemical Diagnostics Systems Co., Ltd). In addition to FIT test results, the dataset includes demographics, routine blood tests, inpatient and outpatient diagnosis and procedure codes, histopathology and imaging reports, chemo- and radiotherapy records, and other sources of information.

The use and collation of the OUH-FIT dataset was described in detail in the Data Protection Impact Assessment (DPIA) form that was approved by the OUH Information Governance team. The study was registered in OUH as a service evaluation (under CSS-BIO-3-4730, later updated as 9076).

4.2.2 Preprocessing

FIT tests requested by GPs in Oxfordshire were identified by comparing locations associated with the FIT test result to a list of known GP practice locations. Non-numeric FIT values (such as 'positive') were discarded. Non-numeric characters in numeric FIT values were removed (for example, '>450' was replaced with 450). The first FIT value was selected for each patient. If multiple values occurred at the same date, the maximum was chosen with priority given to GP FITs. FIT values were further processed using known limits of detection (LoD, 2 µg/g) and quantification (LoQ, 4 µg/g)[124]: values below 2 µg/g were replaced with 0, values between 2 and 4 µg/g were replaced with 4.

'Core' blood tests [125] were identified in the OUH laboratory data for patients who had a FIT value retrieved: haemoglobin, platelets, white cells, MCV, mean cell haemoglobin, serum ferritin, and C-reactive protein. Blood test data was cleaned: non-numeric values were removed, and incorrect numeric values such as dates appearing in the result field were discarded. Blood test results within [-365 days, +14 days] from the earliest FIT were retained, and results that occurred after colorectal cancer diagnosis were excluded. The blood test result closest to FIT date was then selected for each patient.

To characterise the patient cohort, additional data was extracted. Maximum body mass index value within [-365, 0] days of the earliest FIT date was computed for each patient. GP reported symptoms, such as abdominal pain, were extracted from clinical details associated with the earliest FIT test using an extensive set of keywords developed by Withrow et al [125].Common treatments and procedures (polypectomy, local excision, radical resection, chemo- and radiotherapy) were extracted from inpatient and outpatient data using OPCS-4 procedure codes.

4.2.3 Identifying cases of colorectal cancer

Individuals were considered to have colorectal cancer if they had an inpatient or outpatient ICD-10 diagnosis code (C18-C20), or if they had a pathology report that described current colorectal cancer. Colorectal cancer was identified from the reports using an extensive set of keywords developed in collaboration with Dr Neel Doshi. Irrelevant references to colorectal cancer were excluded using a ConText [57]-like regex-based algorithm (for example, general references such as 'in patients with colorectal cancer...' were excluded). Results were validated by examining both the included and excluded matches along with surrounding text. Five erroneous matches were excluded manually.

The date of colorectal cancer was chosen to be the earliest date among inpatient diagnosis codes, outpatient diagnosis codes, and the date at which the pathology report was received (date of biopsy was not available). If a colorectal cancer treatment (radiotherapy, chemotherapy, polypectomy, colonic stent, local excision,

radical resection) occurred within 180 days before the diagnosis code or pathology report, the treatment date was chosen as the cancer date. It was not possible to determine if these treatments were given for colorectal cancer; however, temporal proximity to cancer diagnosis and the nature of treatments made this highly likely.

T-stages for colorectal cancer were extracted from imaging and pathology reports using a rule-based algorithm that first detected all TNM phrases (such as 'pT1/2 N0 M0') and then extracted values; false positive matches were excluded using a ConText-like algorithm [57].

After the study inclusion criteria were applied (see below), the cancers that remained can be assumed to be newly diagnosed rather than prevalent, because the cancers were always preceded by a GP-requested FIT test that is offered to symptomatic patients suspected of colorectal cancer, patients with cancer before FIT were excluded, and the time interval between the FIT test and the first evidence of cancer was not more than 365 days.

4.2.4 Inclusion criteria

Patients were included if (1) their earliest FIT test was requested by the GP, (2) they were at least 18 years old at the time of FIT, (3) they had at least 365 days of follow-up, and (4) they had no records of colorectal cancer before the FIT test. In addition, patients whose earliest record of cancer occurred after the follow-up date were considered not to be cancer cases, but still included. The second inclusion criterion also excludes patients who did not have 365 days of follow-up, but who had cancer before 365 days had passed. This ensures that FITs followed by cancer and FITs not followed by cancer were taken in the same period. Otherwise, the cancer group would contain FIT tests with newer dates than the non-cancer group, and many potential non-cancer cases would be excluded, which can induce bias.

4.2.5 Imputation of missing values

A complete case analysis was conducted as the primary analysis because the variables required for Nottingham models were missing for about 5% of individuals in the

OUH-FIT dataset, less so for individuals with colorectal cancer. Results from imputed data were reported as a sensitivity analysis. For sensitivity analyses, data was imputed using MICE (multiple imputation with chained equations), with random forest and predictive mean matching, using the *miceforest* Python package [126]. Random forest can be more suitable for imputing blood test values, as bloods are unlikely to be linearly related.

4.2.6 The external validation pipeline

Using the OUH-FIT dataset, we applied the Nottingham FIT models and calculated discrimination and calibration metrics as outlined below, followed by a net benefit analysis. To double check that the Nottingham model formulas were implemented correctly, predictions were generated for three common sets of FIT, age, platelet and MCV values, and checked against the predictions of the local Nottingham models.

Baselines: what the Nottingham models are compared to

The Nottingham models were compared to FIT at threshold ≥10 µg Hb/g, the recommended threshold for CRC investigation by NICE and BSG [127, 128]. This was done by computing PPV and other diagnostic metrics for the models at the same level of sensitivity, and by including FIT ≥ 10 in net benefit analysis.

To have a broader overview of the performance of models against FIT over multiple classification thresholds, we present ROC and PR curves for the models and FIT, compute gain in PPV for the models relative to FIT at multiple levels of sensitivity, and compute global performance metrics such as the *c*-statistic) for models and FIT.

In net benefit analysis, the models were also compared to a Nottingham FIT-only model and an Oxford FIT-only spline model. The spline model was derived using Oxford data and predicts the probability of cancer for each FIT value. It used quadratic splines applied to the natural logarithm of FIT values, with 2 knots placed at 10 µg/g and 100 µg/g. Even though the model was derived on Oxford data, its discrimination on Oxford data was not optimistic: it was a monotonic

transformation of the FIT test, so it performed exactly as the FIT test itself. Its calibration metrics may have some optimism, but it is likely to be small compared to the Nottingham models, because the Nottingham models were also recalibrated using the Oxford data for the net benefit analyses. In addition, as the model was a smooth monotonic transformation (no wiggly turns or sharp jumps), any absolute optimism it has is probably small. This model is illustrated in Appendix B.3.

Discrimination metrics

The following metrics were computed for analysing overall discrimination:

- *c*-statistic, an estimator for area under the ROC curve. It estimates the probability that a randomly selected individual with cancer scores higher than an individual without. *c*-statistic was computed for all patients, and additionally in age groups of (18,50], (50, 70], (70, 80], (80, 101].

- Average precision (AP), an estimator for area under the precision-recall (PR) curve [122]. A model will have high AP if it has high positive predictive value across different classification thresholds. AP is better than *c*-statistic for comparing models when the number of positive cases is much smaller than the number of negatives [121]. AP was also computed for all patients, and in the same age groups as the *c*-statistic.

To study discrimination in more detail, we computed common diagnostic metrics at multiple sensitivities and risk thresholds:

- Positive predictive value (PPV), negative predictive value (NPV), and specificity at predefined levels of sensitivity. Sensitivity was chosen to be the same as for FIT test at threshold $\geq$10 µg/g (83.45%), and additional sensitivities were selected (99, 90, 80, 45, 25). If one model has better PPV at the same level of sensitivity, it implies that a smaller number of individuals test positive to capture the same number of cancers.

- PPV, NPV, sensitivity, and specificity at predefined levels of risk: 1%, 2%, 2.5%, 3%, 4%, 5%, 10%. Here, risk is the predicted probability of CRC according to models. These were included for comparison with the original Nottingham report [119].

We also produced the following figures that show discrimination in detail:

- Receiver-operating characteristic (ROC) curve. Plots false positive rate against sensitivity.

- Precision-recall (PR) curve. Plots sensitivity against PPV. If a model has higher PPV than FIT at the same level of sensitivity, it leads to less referrals while detecting the same number of cancers.

- PPV gain relative to FIT. Plots gain in positive predictive value relative to FIT test at each level of sensitivity. This is based on the precision-recall curve, and directly displays the performance of the model relative to the FIT test.

Calibration metrics

We firstly computed minimum required metrics of calibration [120]:

- The observed-expected (O/E) ratio, a ratio between the observed proportion of cancer cases and the average predicted risk of cancer. If the ratio is 1, the model does not over- or underestimate risk in general [38, 120].

- Intercept and slope of a logistic model that predicts the occurrence of cancer from the logits of predicted probabilities. Observed-expected ratio of 1 together with a slope of 1 should indicate that the model does not over- or underestimate risk on average [120].

We also generated calibration curves for exploring calibration over a range of predicted risks (predicted probabilities of CRC):

- Binned calibration curve (reliability diagram). Individuals were divided into 10 bins using equal intervals of predicted probabilities (for example, if the maximum predicted probability is 100%, then individuals with probabilities in the range [0, 10%) are in the first bin)[129]. Curves with decile bins were generated for comparison, but they can be uninformative when cancer prevalence is low. To further explore calibration at lower risk levels, 10 equal-width bins were created within the range 0-20% of risk.

- Smooth calibration curve. Observed cancer events were regressed against predicted probabilities of cancer using locally weighted scatterplot smoothing (LOWESS). Recommended in the literature [38, 120, 130].

Recalibration

In our first round of analysis we found that Nottingham models were not calibrated in Oxford data (see Results 4.3.4). As all Nottingham models performed similarly, we attempted recalibration for the logistic models only, resulting in three additional models:

- 'Nottingham-lr-3.5': multiplication of FIT values by a constant. This recalibration method examines whether miscalibration can be corrected by a simple, externally derived conversion factor. Note that the faecal samples collected for FIT testing were analysed using different sensors in Nottingham and Oxford (Oxford - HM-Jack sensor, Nottingham - OC sensor). In a separate Nottingham dataset, the OC sensor results were 3.5 times higher than HM-Jack results (ignoring zeros), and linear regression predicting OC from HM-Jack results (including zeros) had a slope of 1.3. FIT values were thus multiplied by 3.5 or 1.3, to see if this could sufficiently recalibrate the models.

- 'Nottingham-lr-quant': quantile transformation of FIT values. This method indicates whether the differences in FIT test distributions across datasets are the likely cause of miscalibration. FIT values in Oxford data that fell

within each quantile bin were be replaced with FIT values from the model development data that fell within the same bin. Bins were generated using percentiles from 0 to 95 with 1% step size, and from 95 to 100 with 0.01% step size. If the same Oxford FIT value extended over multiple bins, it was replaced with the median of external FIT values over the same bins.

- 'Nottingham-lr-platt': logistic recalibration (Platt scaling) This method examined whether a simple rescaling of predicted risks could correct miscalibration. The probabilities returned by the model were transformed, using previously estimated logistic intercept and slope, such that the predicted risks were calibrated on average, i.e. the observed-expected ratio was approximately 1.

Net benefit analysis

A decision curve [123] was generated for all Nottingham models, for an Oxford FIT-only spline model, and for FIT at threshold $ge10$ µg/g. The strategies of 'test all' and 'test none' were additionally included for comparison. Net benefit was also summarised in a table that shows number of colonoscopies, detected cancers, missed cancers, negative colonoscopies, net benefit, and net colonoscopies avoided, per 100,000 referrals, as in the Nottingham model validation report [119].

Transformation of FIT values

For the original Nottingham models, and the recalibrated Nottingham-lr-3.5 and Nottingham-lr-platt models, FIT values lower than 4 µg/g were replaced with 4, to match the minimum FIT value in model development dataset. The models apply logarithmic transformations to FIT values, and not replacing zeros with a positive number would have made the models unusable as the logarithms would have produced values of negative infinity. Further replacing all values below four with four was beneficial for model calibration, as then the minimum contribution of FIT test values to the risk score was the same in Oxford and Nottingham datasets.

Confidence intervals

Bootstrap confidence intervals were computed using the percentile method over 1000 samples and checked by examining the bootstrap distributions of statistics. To compute confidence intervals for ROC curves, data of each bootstrap sample was linearly interpolated to the same grid of values using a 1% increment in false positive rate. For PR curves, data was interpolated using the method of Davis and Goadrich [121]. Bootstrap samples were processed in parallel using 16 cores. Bootstrap was stratified so that the percentage of cancers was the same across samples (otherwise, unrealistic variation of prevalence would be observed between samples, including no cancers in some).

In analyses that require imputation, multiple imputation was conducted 5 times within each bootstrap sample, and the average value of the statistic over 5 imputations was computed for each sample. The method of using multiple imputation after bootstrap was shown to provide confidence intervals with intended coverage [131].

To explore how binned calibration curves vary over bootstrap samples, curves derived from 100 randomly selected samples were plotted on the same figure together with the curve from the original sample. If multiple imputation was used, curves from the first imputation of each bootstrap sample were shown.

4.2.7 Software

Analysis was performed in Python 3.9. Discrimination metrics, discrimination curves, and binned calibration curves were computed using the scikit-learn library [132]. Smoothed calibration curve was computed using *statsmodels*[133]. Data was imputed with *miceforest*[126]. Decision curves were generated using *dcurves*[134].

Additional sensitivity analyses were conducted in R: *mice*[135] was used for sensitivity analysis for imputation; common discrimination and calibration metrics were computed using R's *PRROC*[136] and *rms*[137] libraries; decision curves were created with the *dcurves*[138] package.

Code for the external validation analysis (without data) will be made available on `https://github.com/tammandres/fitval` after a suitable period of time has passed and there is permission from the wider team.

4.3 Results

4.3.1 The patient cohort

Inclusion criteria

FIT test result was available for 39,119 patients (770 colorectal cancers). After applying the inclusion criteria (Figure 4.1), 20,627 patients were retained (287 colorectal cancers). Among these, 19,541 patients (284 colorectal cancers) had complete records for sex, age, MCV, and platelets – the variables used in full Nottingham prediction models.

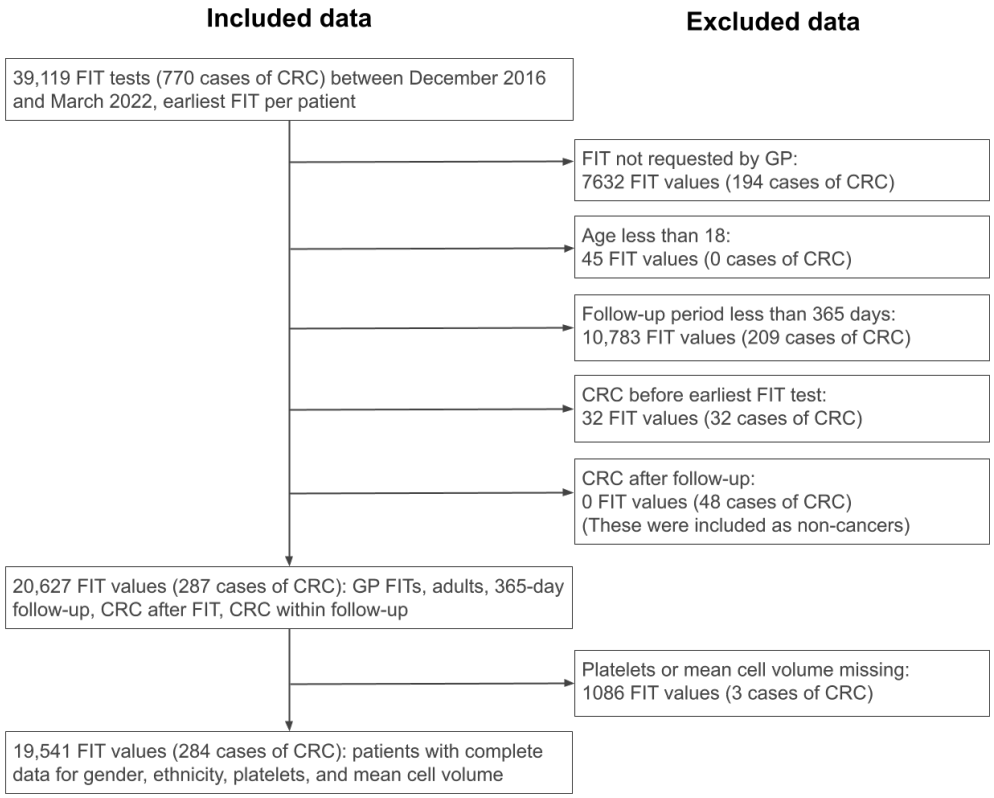


Figure 4.1: Flow diagram showing the steps to building the OUH-FIT external validation cohort. Table 4.1 outlines the characteristics of the patients included in the complete case analysis

Descriptive statistics for included patients

An overview of included individuals in the OUH-FIT dataset is given in Table 4.1. Prevalence of colorectal cancer was 1.45%. Compared to individuals without cancer, patients with cancer had higher median age (73.5 vs 62.3), were more likely to be male (57.4% vs 41.8%) and had higher proportion of FIT test results over 10 µg/g (83.5% vs 7.9%).

Table 4.1: Summary of the OUH-FIT complete case dataset

	No colorectal cancer	Colorectal cancer
Number of patients	19257	284
Age		
18-39.9	1638 (8.5%)	13 (4.6%)
40-49.9	2547 (13.2%)	23 (8.1%)
50-59.9	4679 (24.3%)	39 (13.7%)
60-69.9	3405 (17.7%)	45 (15.8%)
70-79.9	3972 (20.6%)	79 (27.8%)
≥ 3016 (15.7%)	85 (29.9%)	
Median (25th, 75th percentile)	62.3 (51.5, 75.4)	73.5 (59.4, 81.4)
Min and max	18.2, 100.9	31.2, 92.1
Gender		
F	11215 (58.2%)	121 (42.6%)
M	8042 (41.8%)	163 (57.4%)
Ethnicity		
Asian	448 (2.3%)	2 (0.7%)
Black	149 (0.8%)	2 (0.7%)
Mixed	124 (0.6%)	-
Other Ethnic Groups	163 (0.8%)	1 (0.4%)
White	14359 (74.6%)	208 (73.2%)
Not stated	3577 (18.6%)	63 (22.2%)
Not known	437 (2.3%)	8 (2.8%)
Multiple deprivation index		
Median (25th, 75th percentile)	8.0 (7.0, 10.0)	8.0 (7.0, 10.0)
Min, max	1.0, 10.0	1.0, 10.0
Not known	1650 (8.6%)	12 (4.2%)
FIT (µg Hb/g)		
0-1.9	16175 (84.0%)	29 (10.2%)
2-9.9	1554 (8.1%)	18 (6.3%)
10-99.9	1087 (5.6%)	90 (31.7%)
≥100	441 (2.3%)	147 (51.8%)
Median (25th, 75th percentile)	0.2 (0.0, 0.7)	120.4 (25.2, 450.0)
Min, max	0.0, 811.9	0.0, 794.4
Symptoms - GP reported		
Abdominal mass	22 (0.1%)	-
Abdominal pain	2306 (12.0%)	32 (11.3%)
Anaemia	3341 (17.3%)	45 (15.8%)
Bloating	561 (2.9%)	8 (2.8%)
Blood in stool	1853 (9.6%)	26 (9.2%)
Change in bowel habit	6514 (33.8%)	107 (37.7%)
Constipation	650 (3.4%)	8 (2.8%)
Diarrhoea	2097 (10.9%)	35 (12.3%)
Family history of colorectal cancer	199 (1.0%)	2 (0.7%)
Fatigue	220 (1.1%)	6 (2.1%)
Inflammation	215 (1.1%)	5 (1.8%)
Iron deficiency anaemia	1257 (6.5%)	13 (4.6%)
Melaena	224 (1.2%)	2 (0.7%)
Rectal pain	130 (0.7%)	1 (0.4%)
Thrombocytosis	161 (0.8%)	1 (0.4%)

Continued on next page

Table 4.1 – continued from previous page

	No colorectal cancer	Colorectal cancer
Weight loss	1286 (6.7%)	23 (8.1%)
Not known	4913 (25.5%)	67 (23.6%)
Not known	5189 (25.5%)	68 (23.7%)
CRC-relevant treatments		
No treatments recorded	18597 (96.6%)	69 (24.3%)
chemotherapy	421 (2.2%)	101 (35.6%)
local excision	14 (0.1%)	11 (3.9%)
radical resection	77 (0.4%)	160 (56.3%)
radiotherapy	220 (1.1%)	16 (5.6%)
T stage		
1	-	35 (12.3%)
2	-	29 (10.2%)
3	-	82 (28.9%)
4	-	39 (13.7%)
Not known	-	99 (34.9%)

Notes. *T-stage was extracted from radiology and pathology reports using a pattern-matching algorithm. **CRC-relevant treatments are procedures used for treating colorectal cancer (CRC), but they may also be given for other conditions. Data is shown for patients who had complete records for platelets and mean cell volume (in addition to FIT values, sex and age that were available for all patients).

Descriptive statistics for core blood tests

Summaries of core blood tests are given in Table 4.2. Patients with colorectal cancer were more likely to have low haemoglobin (52.5% vs 34.8%), elevated platelets (22.5% vs 10.1%), high white cell counts (15.8% vs 12.4%), low mean cell haemoglobin concentration (33.5% vs 17.6%), and low mean cell volume (19.4% vs 7.1%), compared to patients without cancer. They were also more likely to have high serum ferritin and C reactive protein. Note that these are point estimates; statistical tests for the difference in proportions were not included as this was not the objective of the analysis.

4.3.2 Comparison of Nottingham and Oxford datasets

General characteristics of Nottingham and Oxford FIT datasets are given in Table 4.3. In both cases, the setting of FIT testing was primary care, patients were followed up for 12 months to detect cancer in the secondary care healthcare record, and the rate of cancer was similar (1.56% in Nottingham, 1.39% in the entire Oxford dataset, and 1.45% in Oxford complete cases). Table 4.3 includes all Oxford data, not complete cases.

Table 4.2: Summary of core bloods in the OUH-FIT complete case dataset

	No colorectal cancer	Colorectal cancer (CRC)
Number of patients	19257	284
Haemoglobin		
Median (25th, 75th percentile)	133.0 (121.0, 144.0)	125.0 (109.8, 140.0)
Min, max	50.0, 226.0	55.0, 187.0
Low haemoglobin	6705 (33.0%)	149 (51.9%)
Normal haemoglobin	15320 (75.3%)	170 (59.2%)
Not known	1078 (5.3%)	3 (1.0%)
Platelets		
Median (25th, 75th percentile)	263.0 (221.0, 313.0)	298.0 (242.8, 372.5)
Min, max	9.0, 1288.0	103.0, 920.0
High platelets	1939 (9.5%)	64 (22.3%)
Normal platelets	18485 (90.9%)	250 (87.1%)
Not known	1083 (5.3%)	3 (1.0%)
White cells		
Median (25th, 75th percentile)	6.6 (5.5, 8.1)	7.4 (6.3, 9.2)
Min, max	1.6, 237.5	3.3, 17.8
High white cells	2388 (11.7%)	45 (15.7%)
Normal white cells	18807 (92.5%)	270 (94.1%)
Not known	1079 (5.3%)	3 (1.0%)
Mean cell haemoglobin (MCH)		
Median (25th, 75th percentile)	29.9 (28.4, 31.1)	28.6 (26.2, 30.3)
Min, max	13.8, 49.0	15.4, 37.3
Low MCH	3382 (16.6%)	95 (33.1%)
Normal MCH	16943 (83.3%)	216 (75.3%)
Not known	1079 (5.3%)	3 (1.0%)
Mean cell volume (MCV)		
Median (25th, 75th percentile)	91.0 (87.5, 94.3)	89.0 (84.4, 93.0)
Min, max	53.1, 134.7	61.8, 107.5
Low MCV	1373 (6.8%)	55 (19.2%)
Normal MCV	18450 (90.7%)	250 (87.1%)
Not known	1079 (5.3%)	3 (1.0%)
Serum ferritin (CFER)		
Median (25th, 75th percentile)	68.7 (24.9, 143.8)	33.2 (13.7, 112.2)
Min, max	1.0, 4572.5	1.0, 931.0
High CFER	671 (3.3%)	10 (3.5%)
Low CFER	2527 (12.4%)	67 (23.3%)
Normal CFER	10760 (52.9%)	172 (59.9%)
Not known	9580 (47.1%)	115 (40.1%)
C-reactive protein (CRP)		
Median (25th, 75th percentile)	2.0 (0.8, 5.4)	5.0 (1.8, 21.1)
Min, max	0.2, 358.7	0.2, 236.4
High CRP	3494 (17.2%)	85 (29.6%)
Normal CRP	13719 (67.4%)	159 (55.4%)
Not known	5164 (25.4%)	65 (22.6%)

Notes. Normal, high, and low values for these bloods were defined as in Withrow et al [125]. Low HGB: < 130 g/L for males, < 120 g/L for females. High PLT: > 400 * 109/L. High WBC: > 11 * 109/L. Low MCH: < 27.4 pg/cell. Low MCV: < 80 fl. Low CFER: < 20 µg/L. High CFER: ≥ 350 µg/L. High CRP: > 10 mg/L. Data is shown for patients who had complete records for platelets and mean cell volume (in addition to FIT values, sex and age that were available for all patients).

Table 4.3: Description of Nottingham and Oxford FIT datasets

	Nottingham	Oxford (OUH-FIT)
Population	34,231	20,627
Setting of FIT testing	Primary Care	Primary Care
Eligibility	Patients deemed at risk of colorectal cancer by the GP	Patients deemed at risk of colorectal cancer by the GP
Study period	11/2017 to 11/2021	01/01/2017 to 31/3/2022
Age range (years)	Not known	18-101
Age, median (25th, 75th)	66 (54, 77)	62.18 (51.40, 75.36)
Sample collection device	Primarily buffer device	Stool pot until June 2021, then transition to buffer device
FIT analytical method	OC-Sensor PLEDIA	HM-JACKarc
FIT reported range, µg Hb/g faeces	4-69,800	2-400
Timing between FIT and colorectal cancer	up to 12 months	up to 12 months
FIT values, median (25th, 75th)	4 (4, 8)	0.20 (0, 0.80)
Colorectal cancer cases (%)	533 (1.56%)	287 (1.39%)
Haemoglobin, median (25th, 75th)	131 (118, 143)	133 (121, 144)
Mean cell volume, median (25th, 75th)	92 (88, 96)	91 (87.5, 94.20)
Platelets, median (25th, 75th)	268 (219, 324)	264 (221, 313)
Ethnicity		
White	70.8%	74.5%
Asian	4.3%	2.3%
Black	2.5%	0.77%
Other	1.9%	0.84%
Not recorded	20.6%	20.9%

4.3.3 Discrimination: distinguishing cancers from non-cancers

This section presents the overall discrimination metrics (*c*-statistic and average precision), and common threshold-based diagnostic metrics such as positive predictive value. The threshold-based metrics are reported at selected levels of sensitivity, but not at selected levels of predicted probabilities of cancer because the models were not calibrated (see section 4.3.4).

Overall discrimination

All Nottingham models discriminated well between cancers and non-cancers: *c*-statistics ranged from 90.6% to 92.7%, and average precisions (AP) from 21.6% to 31.3% (Table 4.4). The AP of a model that predicts randomly is the same as the proportion of cancers in the dataset (1.45%), so the AP of all models was better than random guessing.

The FIT test had a similar *c*-statistic (91.5%), but lower AP (21.8%) than the full Nottingham logistic and Cox models that incorporate FIT, age, sex, and bloods – these full models had an AP between 30.8-31.3, and *c*-statistic between 92.5-92.7 (Table 4.4). The Nottingham FIT-age and FIT-age-sex models had higher point estimates of performance than the FIT test alone, but lower point estimates than the logistic and Cox models (Table 4.4). The Nottingham FIT-only model had slightly lower *c*-statistic (90.6) and AP (21.6) than the FIT test, because all FIT values less than 4 were replaced with 4 to match the development dataset value range.

Overall discrimination by age group

In the younger age groups (18-40 and 40-50 years), the *c*-statistics of Nottingham FIT-age, FIT-age-sex, and full models ranged from (93.9% to 96.6%), whereas in older age groups they ranged from 89.0% to 93.2% (Table 4.5). However, it is not clear if the gain in *c*-statistic observed in younger age-groups is statistically significant, as the confidence intervals were overlapping. In addition, the number of cancers in younger age groups was very small (13 in the 18-40 group and 23 in the 40-50 group), so the point estimates were very imprecise (wide confidence intervals).

Table 4.4: Average precision and *c*-statistic for Nottingham prediction models and the FIT test

Model	Average precision (%)	*c*-statistic (%)
FIT test	21.8 (19.0, 25.6)	91.5 (89.5, 93.4)
FIT-spline	21.8 (19.0, 25.6)	91.5 (89.5, 93.4)
Simpler Nottingham models		
Nottingham-fit	21.6 (18.8, 25.5)	90.6 (88.4, 92.5)
Nottingham-fit-age	25.2 (21.6, 30.4)	92.1 (90.3, 93.8)
Nottingham-fit-age-sex	25.8 (22.4, 31.3)	92.2 (90.4, 93.9)
Full Nottingham models		
Nottingham-lr	31.2 (26.6, 36.7)	92.6 (90.7, 94.4)
Nottingham-lr-boot	30.8 (26.2, 36.4)	92.5 (90.5, 94.3)
Nottingham-cox	31.3 (26.7, 36.7)	92.7 (90.9, 94.4)
Nottingham-cox-boot	31.2 (26.5, 36.7)	92.7 (90.8, 94.4)

Notes. FIT-spline is an Oxford-derived FIT-only model. Nottingham-lr and Nottingham-cox are full logistic and Cox models that contain FIT, age, sex and blood tests as predictors. Nottingham-lr-boot and Nottingham-cox-boot are variants of the full models developed using bootstrap averaging. Nottingham-fit, Nottingham-fit-age, and Nottingham-fit-age-sex are logistic models with less predictor variables: FIT, FIT and age, and FIT-age-sex, respectively.

The average precision of the Nottingham FIT-age, FIT-age-sex, and full models did not show a simple trend (higher or lower) between younger and older age groups (Table 4.5).

Even if the models may perform better in younger age groups than older age groups, it is not clear if they would perform better than the FIT test alone within these age groups. For example, in the youngest (18-40) age group, the *c*-statistic of FIT alone was 96.6% whereas the *c*-statistic observed for the Nottingham models ranged from 93.2 to 96.6. The 40-50 years age group had a somewhat lower *c*-statistic for FIT alone than for the models (91.8 vs 91.8 to 95.6), but the estimates were imprecise.

In conclusion, a larger dataset is required to know if any differences observed between age groups are reliable.

Discrimination by sensitivity

At higher risk thresholds, where less than 45% of all cancers would be detected (sensitivities <45%), the precision-recall curve showed that the full Nottingham

models had higher positive predictive value (PPV) than the FIT test (Figure 4.2). For example, at a threshold where sensitivity was about 25%, the PPV of the models was about 14% higher than that of FIT (Table 4.6). At sensitivities below 45%, confidence intervals for the gain in PPV compared to FIT did not usually overlap with zero (Figure 4.2), so the gain was statistically significant.

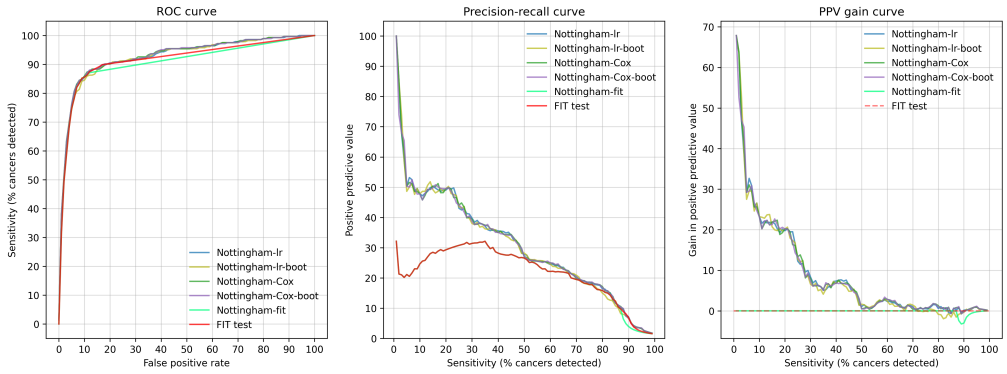
At lower risk thresholds, where more than 55% of all cancers would be detected (sensitivities >55%), the PPVs of all Nottingham models and FIT were similar. This was true for the full Nottingham models (Figure 4.2) and the FIT-age and FIT-age-sex models (Figure 4.3). For example, at a threshold where 90% of all cancers were detected (90% sensitivity), the PPV of FIT was 7.0%, whereas the PPV of full Nottingham models ranged from 6.6 to 7.1% (Table 4.6), with confidence intervals overlapping (Figure 4.2). The Nottingham FIT-only model also performed similarly to the more complex Nottingham models in this higher-sensitivity region (Figure 4.3). (An exception to this pattern is that the Nottingham FIT-only, FIT-age and FIT-age-sex models had somewhat lower point estimates of PPV at 90% sensitivity than the full models, but this was probably because all FIT values less than 4 were replaced with 4 to match the range of development data, which meant that there was some loss of information at these lower thresholds.)

The FIT test at threshold ≥10 µg/g had sensitivity of 83.5% and PPV of 13.7%. At the same level of sensitivity, the PPVs of all Nottingham models ranged from 12.6% to 14.5% (Table 4.6), and confidence intervals for the gain in PPV relative to FIT overlapped with zero (Figures 4.2 and 4.3).

Running the analysis on a multiply imputed dataset, rather than complete cases, led to similar results (Appended Figure B.7).

The risk thresholds given in Table 4.6 should not be interpreted as probabilities because the original Nottingham models were not calibrated. However, a Nottingham logistic regression model that was well calibrated after quantile transformation had sensitivities less than 45% for predicted risks greater than 15.6%. The higher thresholds where Nottingham models performed better thus corresponded to risks roughly greater than 15%.

(a)



(b)

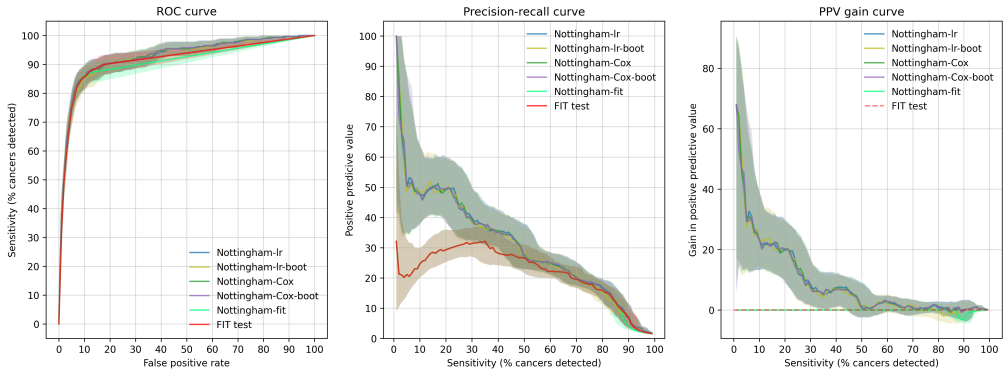


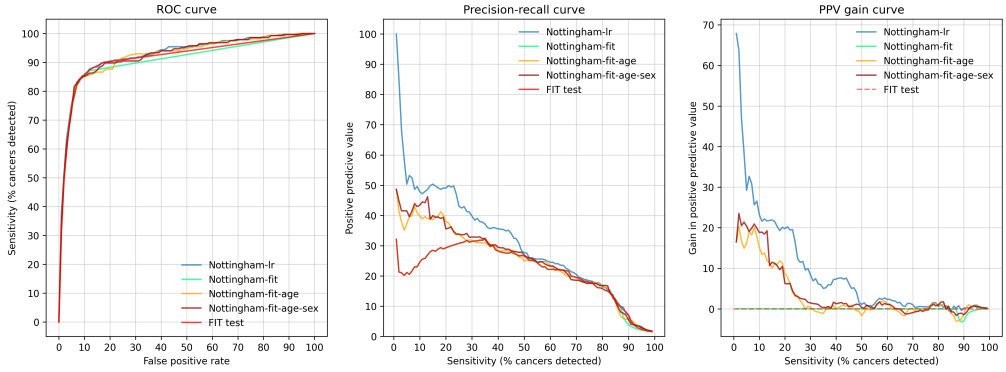
Figure 4.2: ROC curve, precision-recall curve, and gain in positive predictive value relative to FIT for the full Nottingham logistic and Cox models, and for the FIT-only model. The full models include FIT, age, sex, mean cell volume and platelets as predictors. Curves show the full logistic model ('Nottingham-lr'), bootstrap-averaged logistic model ('Nottingham-lr-boot'), full Cox model ('Nottingham-cox'), bootstrap averaged Cox model ('Nottingham-cox-boot'), the FIT-only model ('Nottingham-fit'), and the FIT test. Confidence intervals are not included in the top panel (a); bottom panel displays 95% bootstrap percentile confidence intervals (b). Curves of the FIT test and Nottingham FIT-only model are almost completely overlapping because the FIT-only model is a monotonic transformation of FIT values.

Table 4.5: Average precision and *c*-statistic by age group for the Oxford FIT test and Nottingham models

Model	Average precision	*c*-statistic
Age (18, 40) - 1651 patients, 13 CRC		
FIT	14.1 (8.7, 31.9)	96.6 (94.6, 98.1)
FIT-spline	14.1 (8.7, 31.9)	96.6 (94.6, 98.1)
Nottingham-fit	13.6 (7.6, 31.8)	93.2 (84.8, 98.1)
Nottingham-fit-age	17.7 (10.4, 37.1)	96.6 (93.8, 98.6)
Nottingham-fit-age-sex	14.4 (8.9, 28.8)	95.0 (89.4, 98.4)
Nottingham-lr	29.5 (11.3, 52.9)	93.9 (85.8, 98.5)
Nottingham-lr-boot	29.3 (11.4, 52.7)	93.8 (85.7, 98.5)
Nottingham-cox	29.3 (11.3, 52.7)	93.9 (85.8, 98.5)
Nottingham-cox-boot	26.5 (11.2, 51.1)	93.9 (85.9, 98.5)
Age [40, 50) - 2570 patients, 23 CRC		
FIT	22.4 (15.1, 34.9)	91.8 (82.3, 98.4)
FIT-spline	22.4 (15.1, 34.9)	91.8 (82.3, 98.4)
Nottingham-fit	22.3 (15.0, 34.8)	91.8 (83.0, 98.4)
Nottingham-fit-age	22.8 (16.0, 36.8)	95.6 (90.5, 98.7)
Nottingham-fit-age-sex	22.4 (15.7, 34.7)	94.4 (87.4, 98.6)
Nottingham-lr	36.8 (22.1, 57.0)	94.8 (88.8, 98.7)
Nottingham-lr-boot	32.5 (20.5, 52.8)	95.0 (89.2, 98.8)
Nottingham-cox	36.2 (21.4, 55.8)	95.0 (89.1, 98.7)
Nottingham-cox-boot	35.8 (21.3, 55.0)	94.7 (88.2, 98.7)
Age [50, 60) - 4718 patients, 39 CRC		
FIT	22.6 (16.3, 36.1)	94.3 (89.7, 98.1)
FIT-spline	22.6 (16.3, 36.1)	94.3 (89.7, 98.1)
Nottingham-fit	22.5 (16.1, 35.8)	93.0 (87.7, 97.4)
Nottingham-fit-age	22.7 (15.7, 36.8)	92.5 (85.9, 97.6)
Nottingham-fit-age-sex	25.9 (17.7, 40.8)	91.3 (83.5, 97.2)
Nottingham-lr	38.4 (26.6, 55.3)	90.5 (82.8, 97.0)
Nottingham-lr-boot	37.6 (26.1, 55.0)	90.2 (82.4, 96.9)
Nottingham-cox	37.7 (26.2, 55.1)	90.6 (83.2, 97.0)
Nottingham-cox-boot	37.0 (25.5, 54.2)	90.5 (83.0, 97.0)
Age [60, 70) - 3450 patients, 45 CRC		
FIT	25.2 (17.5, 36.4)	91.9 (86.4, 96.4)
FIT-spline	25.2 (17.5, 36.4)	91.9 (86.4, 96.4)
Nottingham-fit	24.8 (16.9, 36.0)	89.8 (83.7, 94.9)
Nottingham-fit-age	28.6 (19.8, 41.4)	90.1 (82.3, 96.3)
Nottingham-fit-age-sex	30.0 (20.5, 43.5)	90.3 (83.7, 95.5)
Nottingham-lr	32.0 (21.6, 46.2)	93.2 (88.2, 97.4)
Nottingham-lr-boot	30.8 (20.8, 45.0)	92.9 (88.1, 97.0)
Nottingham-cox	32.0 (21.9, 46.3)	93.2 (88.2, 97.3)
Nottingham-cox-boot	32.7 (22.0, 46.8)	93.2 (88.3, 97.3)
Age [70, 80) - 4051 patients, 79 CRC		
FIT	30.8 (23.4, 42.0)	90.1 (85.2, 94.3)
FIT-spline	30.8 (23.4, 42.0)	90.1 (85.2, 94.3)
Nottingham-fit	30.3 (22.9, 41.8)	89.3 (84.5, 93.4)
Nottingham-fit-age	31.5 (24.1, 43.1)	90.6 (86.1, 94.1)
Nottingham-fit-age-sex	32.9 (25.4, 45.2)	90.3 (85.5, 94.2)
Nottingham-lr	39.6 (30.4, 51.3)	89.0 (84.1, 93.3)
Nottingham-lr-boot	40.2 (30.8, 51.8)	89.1 (84.2, 93.4)
Nottingham-cox	39.8 (30.6, 51.3)	89.0 (84.2, 93.3)
Nottingham-cox-boot	40.0 (30.9, 51.3)	89.1 (84.3, 93.4)
Age [80, 101) - 3101 patients, 85 CRC		
FIT	23.3 (17.7, 31.6)	89.7 (85.6, 92.9)
FIT-spline	23.3 (17.7, 31.6)	89.7 (85.6, 92.9)
Nottingham-fit	22.9 (17.1, 31.4)	88.2 (84.1, 92.0)
Nottingham-fit-age	23.6 (18.1, 32.0)	89.7 (86.0, 92.9)
Nottingham-fit-age-sex	23.5 (18.4, 32.0)	90.3 (86.8, 93.3)
Nottingham-lr	27.3 (21.1, 36.6)	90.9 (87.3, 93.8)
Nottingham-lr-boot	26.7 (20.8, 36.3)	90.6 (86.9, 93.6)
Nottingham-cox	27.0 (20.9, 36.5)	90.9 (87.4, 93.8)
Nottingham-cox-boot	27.1 (20.9, 36.7)	91.0 (87.4, 93.8)

Notes. FIT-spline is an Oxford-derived FIT-only model. Nottingham-lr and Nottingham-cox are full logistic and Cox models that contain FIT, age, sex and blood tests as predictors. Nottingham-lr-boot and Nottingham-cox-boot are variants of the full models developed using bootstrap averaging. Nottingham-fit, Nottingham-fit-age, and Nottingham-fit-age-sex are logistic models with less predictor variables: FIT, FIT and age, and FIT-age-sex, respectively.

(a)



(b)

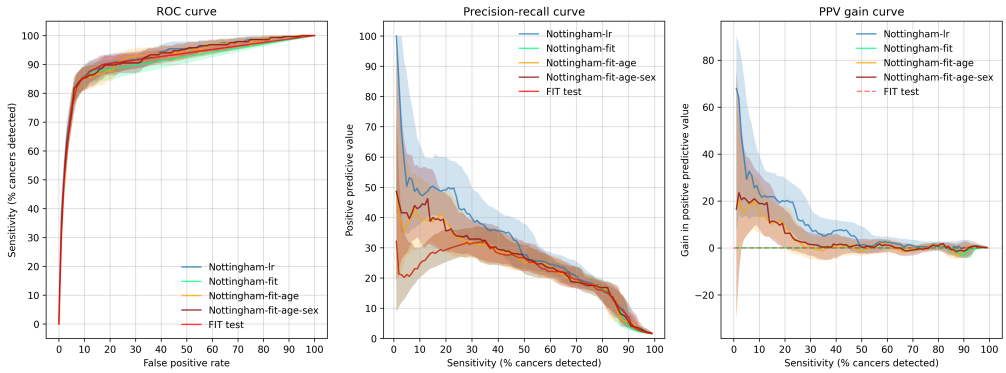


Figure 4.3: ROC curve, precision-recall curve, and gain in positive predictive value relative to FIT for the Nottingham logistic models and the FIT test. Curves represent the Nottingham FIT-only model ('Nottingham-fit'), the Nottingham FIT and age model ('Nottingham-fit-age'), the Nottingham FIT-age-sex model ('Nottingham-fit-age-sex'), and the full Nottingham logistic model that additionally includes platelets and mean cell volume ('Nottingham-lr'). The upper panel (a) does not show confidence intervals, the lower panel (b) displays 95% bootstrap percentile confidence intervals. The curves of the FIT test and the Nottingham FIT-only model are almost completely overlapping because the FIT-only model is a monotonic transformation of FIT values.

Table 4.6: Specificity, positive predictive value and negative predictive value at selected levels of sensitivity for the FIT test, for the FIT-spline model, and the Nottingham models

Model	Specificity	PPV	NPV	Threshold (approx.)*
Sensitivity 99%				
FIT	7.6 (6.0, 10.6)	1.6 (1.5, 1.6)	99.8 (99.8, 99.9)	0.36
FIT-spline	7.6 (6.0, 10.6)	1.6 (1.5, 1.6)	99.8 (99.8, 99.9)	0.24
Nottingham-fit	6.3 (4.9, 8.8)	1.5 (1.5, 1.6)	99.8 (99.7, 99.8)	0.16
Nottingham-fit-age	18.3 (7.1, 29.9)	1.8 (1.5, 2.0)	99.9 (99.8, 100.0)	0.08
Nottingham-fit-age-sex	17.1 (5.5, 30.5)	1.7 (1.5, 2.1)	99.9 (99.7, 100.0)	0.08
Nottingham-lr	18.1 (5.9, 31.6)	1.8 (1.5, 2.1)	99.9 (99.8, 100.0)	0.06
Nottingham-lr-boot	18.4 (5.4, 30.4)	1.8 (1.5, 2.1)	99.9 (99.7, 100.0)	0.11
Nottingham-cox	18.3 (5.7, 31.0)	1.8 (1.5, 2.1)	99.9 (99.7, 100.0)	0.07
Nottingham-cox-boot	16.4 (5.6, 30.3)	1.7 (1.5, 2.1)	99.9 (99.7, 100.0)	0.07
Sensitivity 90%				
FIT	76.4 (59.7, 89.1)	7.0 (3.3, 10.8)	99.8 (99.8, 99.8)	3.64
FIT-spline	76.4 (59.7, 89.1)	7.0 (3.3, 10.8)	99.8 (99.8, 99.8)	0.98
Nottingham-fit	63.0 (49.0, 89.1)	3.9 (2.7, 10.8)	99.8 (99.7, 99.8)	0.16
Nottingham-fit-age	77.6 (69.9, 86.7)	5.6 (4.2, 9.1)	99.8 (99.8, 99.8)	0.26
Nottingham-fit-age-sex	77.3 (64.8, 86.5)	5.5 (3.6, 8.9)	99.8 (99.8, 99.8)	0.27
Nottingham-lr	82.7 (62.9, 90.0)	7.1 (3.5, 11.7)	99.8 (99.8, 99.8)	0.29
Nottingham-lr-boot	81.1 (64.2, 88.8)	6.6 (3.6, 10.6)	99.8 (99.8, 99.8)	0.48
Nottingham-cox	81.9 (65.6, 89.8)	6.8 (3.7, 11.5)	99.8 (99.8, 99.8)	0.26
Nottingham-cox-boot	81.7 (64.9, 89.7)	6.8 (3.6, 11.5)	99.8 (99.8, 99.8)	0.29
Sensitivity 83.45%**				
FIT	92.3 (87.2, 94.1)	13.7 (9.1, 17.1)	99.7 (99.7, 99.7)	10.45
FIT-spline	92.3 (87.2, 94.1)	13.7 (9.1, 17.1)	99.7 (99.7, 99.7)	2.78
Nottingham-fit	92.3 (83.3, 94.1)	13.7 (8.3, 17.1)	99.7 (99.7, 99.7)	0.66
Nottingham-fit-age	92.6 (82.1, 94.5)	14.2 (6.4, 18.3)	99.7 (99.7, 99.7)	0.71
Nottingham-fit-age-sex	92.4 (85.4, 94.5)	13.9 (7.8, 18.3)	99.7 (99.7, 99.7)	0.68
Nottingham-lr	92.6 (88.2, 94.5)	14.3 (9.4, 18.4)	99.7 (99.7, 99.7)	0.74
Nottingham-lr-boot	91.5 (85.3, 94.0)	12.6 (7.7, 17.1)	99.7 (99.7, 99.7)	0.87
Nottingham-cox	92.5 (89.0, 94.7)	14.1 (10.1, 18.7)	99.7 (99.7, 99.7)	0.7
Nottingham-cox-boot	92.7 (88.8, 94.6)	14.5 (9.9, 18.6)	99.7 (99.7, 99.7)	0.76
Sensitivity 80%				
FIT	93.7 (91.5, 95.0)	15.8 (12.2, 19.1)	99.7 (99.7, 99.7)	16.05
FIT-spline	93.7 (91.5, 95.0)	15.8 (12.2, 19.1)	99.7 (99.7, 99.7)	4.28
Nottingham-fit	93.7 (91.5, 95.0)	15.8 (12.2, 19.1)	99.7 (99.7, 99.7)	1.18
Nottingham-fit-age	94.2 (91.5, 95.0)	16.8 (12.2, 19.2)	99.7 (99.7, 99.7)	1.27
Nottingham-fit-age-sex	94.2 (91.8, 94.9)	16.8 (12.6, 18.8)	99.7 (99.7, 99.7)	1.27
Nottingham-lr	94.0 (91.1, 95.1)	16.4 (11.7, 19.5)	99.7 (99.7, 99.7)	1.11
Nottingham-lr-boot	93.3 (90.6, 94.9)	15.0 (11.2, 18.9)	99.7 (99.7, 99.7)	1.12
Nottingham-cox	94.3 (92.0, 95.2)	17.1 (12.8, 19.7)	99.7 (99.7, 99.7)	1.28
Nottingham-cox-boot	94.2 (91.9, 95.2)	16.8 (12.7, 19.8)	99.7 (99.7, 99.7)	1.2
Sensitivity 45%				
FIT	98.3 (97.9, 98.6)	27.8 (23.6, 32.0)	99.2 (99.2, 99.2)	161.75
FIT-spline	98.3 (97.9, 98.6)	27.8 (23.6, 32.0)	99.2 (99.2, 99.2)	18.57
Nottingham-fit	98.3 (97.9, 98.6)	27.8 (23.6, 32.0)	99.2 (99.2, 99.2)	10.43
Nottingham-fit-age	98.3 (97.8, 98.7)	28.5 (22.8, 33.3)	99.2 (99.2, 99.2)	8.28
Nottingham-fit-age-sex	98.3 (97.9, 98.7)	28.3 (24.0, 33.8)	99.2 (99.2, 99.2)	7.67
Nottingham-lr	98.7 (98.0, 99.0)	34.1 (24.9, 40.2)	99.2 (99.2, 99.2)	9.81
Nottingham-lr-boot	98.6 (98.0, 99.0)	32.6 (24.9, 40.0)	99.2 (99.2, 99.2)	7.48
Nottingham-cox	98.7 (97.9, 99.0)	34.4 (24.4, 40.1)	99.2 (99.2, 99.2)	9.93
Nottingham-cox-boot	98.7 (97.9, 99.0)	34.2 (24.4, 40.0)	99.2 (99.2, 99.2)	9.53
Sensitivity 25%				
FIT	99.2 (99.0, 99.3)	30.8 (26.8, 35.9)	98.9 (98.9, 98.9)	450.06
FIT-spline	99.2 (99.0, 99.3)	30.8 (26.8, 35.9)	98.9 (98.9, 98.9)	27.65
Nottingham-fit	99.2 (99.0, 99.3)	30.8 (26.8, 35.9)	98.9 (98.9, 98.9)	17.51
Nottingham-fit-age	99.3 (99.0, 99.5)	33.8 (27.2, 43.6)	98.9 (98.9, 98.9)	15.81
Nottingham-fit-age-sex	99.3 (99.1, 99.5)	33.8 (28.1, 41.5)	98.9 (98.9, 98.9)	15.95
Nottingham-lr	99.5 (99.3, 99.7)	43.0 (33.8, 57.3)	98.9 (98.9, 98.9)	18.7
Nottingham-lr-boot	99.5 (99.2, 99.7)	43.6 (32.9, 56.4)	98.9 (98.9, 98.9)	15.16
Nottingham-cox	99.5 (99.3, 99.7)	44.1 (33.5, 56.8)	98.9 (98.9, 98.9)	18.56
Nottingham-cox-boot	99.5 (99.3, 99.7)	44.1 (33.0, 56.8)	98.9 (98.9, 98.9)	18.16

Notes. *The threshold for FIT test (FIT) is given in micrograms Hb / g, and for FIT-spline and Nottingham models it is given as probability of cancer in percentage (e.g. 10.43 is 10.43% probability of cancer). The threshold is marked as approximate, because specificity, PPV and NPV were interpolated to estimate these quantities at exact levels of sensitivity, and threshold was thus interpolated too. **The sensitivity of 83.45% is also the sensitivity of FIT test at threshold 10. The Nottingham models include the full logistic model ('Nottingham-lr'), bootstrap-averaged full logistic model ('Nottingham-lr-boot'), full Cox model ('Nottingham-cox'), bootstrap averaged full Cox model ('Nottingham-cox-boot'), logistic model with FIT-age-sex as predictors ('Nottingham-fit-age-sex'), logistic model with FIT and age as predictors ('Nottingham-fit-age'), and a FIT-only logistic model ('Nottingham-fit'). FIT-spline is an Oxford-derived FIT-only model.

4.3.4 Calibration: comparing predicted and actual probabilities of cancer

Calibration of original Nottingham models

The observed-expected (o/e) ratio was approximately 2 for all original Nottingham models, suggesting that the models predicted too low risk on average (Table 4.7). Smooth calibration curves showed that the Nottingham models predicted lower than observed risks over the full range of predicted risks in the OUH-FIT dataset (Figures 4.4 and 4.5). Calibration curves based on dividing predicted risks into bins provided the same conclusion (Appended Figures B.5 and B.6).

For risks less than 20%, which are more likely to be used as decision thresholds in clinical practice, the predicted probability of cancer was approximately half the true probability (Figure 4.4 and 4.5). For example, a predicted risk of 5% corresponded approximately to a true risk of 10%.

The calibration of all Nottingham logistic models, the Cox model, and the bootstrap-averaged Cox model was almost identical (Figures 4.4 and 4.5), whereas the logistic bootstrap-averaged model had slightly better calibration at risk thresholds less than 2%, and worse calibration at thresholds greater than 2% (Figure 4.4.

Running the analysis on a multiply imputed dataset, rather than complete cases, led to similar results (Appended Figure B.8).

Recalibration

Recalibration attempts were made for the full logistic regression model ('Nottingham-lr') and for the FIT-only, FIT-age, and FIT-age-sex models. The full Cox and bootstrap-averaged models were not recalibrated as they performed almost identically to the full logistic model (see Figure 4.2).

- Quantile transformation of Oxford FIT values to Nottingham FIT values approximately recalibrated all models, yielding an o/e ratio of 1.0 for the FIT-only model, 1.1 for the other models, and producing calibration curves that were close to the ideal calibration line (see models with '-quant' suffix in Table 4.7 and Figure 4.6).

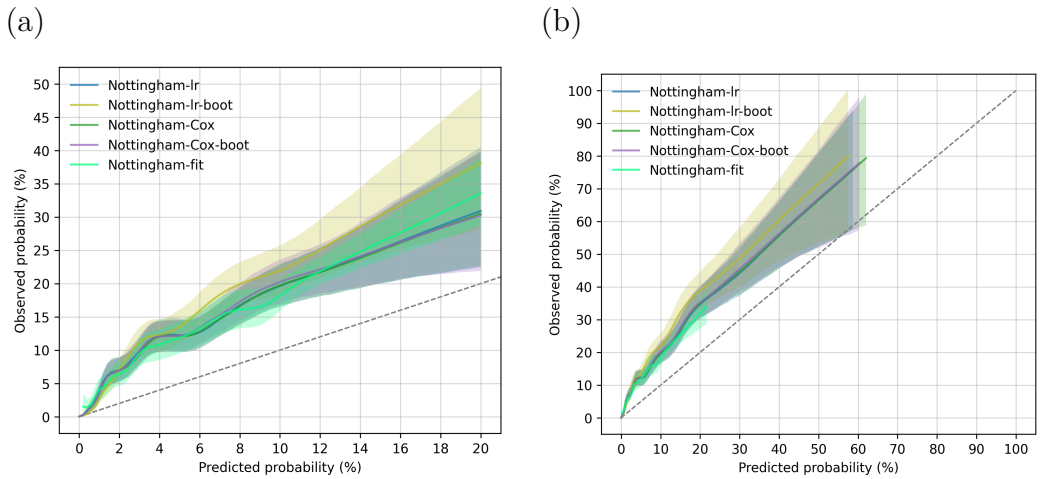


Figure 4.4: Smooth calibration curves for the full Nottingham logistic and Cox models, and for the FIT-only model. The full models include FIT, age, sex, MCV and platelets as predictors. Curves show the full logistic model ('Nottingham-lr'), bootstrap-averaged logistic model ('Nottingham-lr-boot'), full Cox model ('Nottingham-cox'), bootstrap averaged Cox model ('Nottingham-cox-boot'), and the FIT-only logistic model ('Nottingham-fit'). Left panel shows calibration in the clinically meaningful range of risks (<20%, a), and right panel over the full range of risks (b). Curves were created by applying LOWESS-smoothing to predicted probabilities and cancer events. Shaded areas show 95% bootstrap percentile confidence intervals.

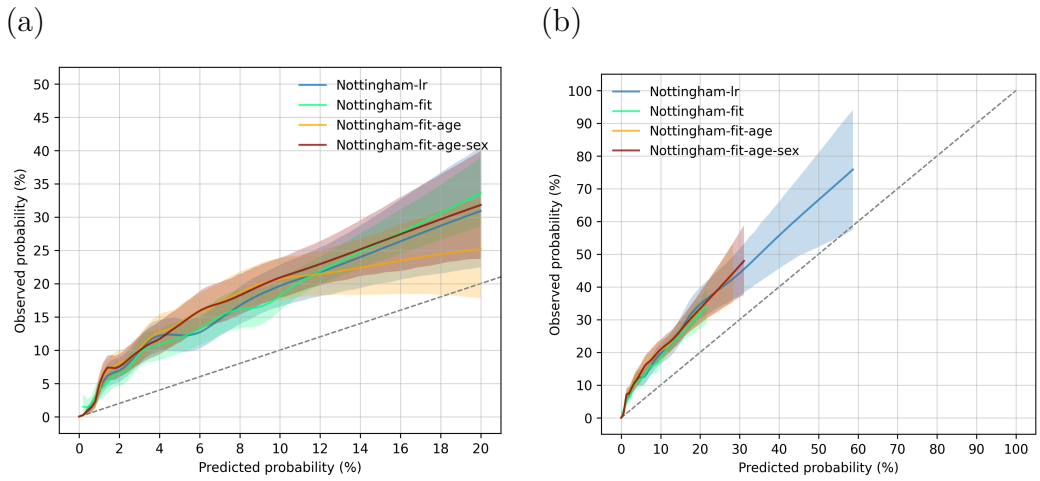


Figure 4.5: Smooth calibration curves for the full and simpler Nottingham logistic models. Curves display the full logistic model ('Nottingham-lr'), and the FIT-only, FIT-age, and FIT-age-sex models ('Nottingham-fit', 'Nottingham-fit-age', 'Nottingham-fit-age-sex'). Left panel shows calibration in the clinically meaningful range of risks (<20%, a), and right panel over the full range of risks (b). Curves were created by applying LOWESS-smoothing to predicted probabilities and cancer events. Shaded areas show 95% bootstrap percentile confidence intervals.

- Constant multiplication of all FIT values by 3.5 before applying the models also recalibrated the models, yielding very similar results to quantile transformation (see models with '-3.5' suffix in Table 4.7 and Figure 4.6).

- Logistic recalibration improved calibration, but less than quantile transform or constant multiplication: predicted risks were somewhat low on the lower risk range (3% predicted risk corresponded to roughly 6% actual risk), and high in the higher range (20% corresponded to roughly 15%) ('Nottingham-lr-platt' in Table 4.7 and Figure 4.6). o/e ratio was 1 by definition.

The success of quantile transformation and constant multiplication for recalibrating the models is consistent with the observation that Oxford FIT values were lower than in Nottingham (Appendix B.4).

Note that quantile transformation and logistic recalibration are not shown in Figure 4.6 for the FIT-age and FIT-age-sex models to make the Figure easier to read. As these methods performed similarly to the simplest method (conversion factor), only the conversion approach is shown for these models.

Table 4.7: Overall calibration metrics for the original and recalibrated Nottingham models

Model	Event rate (%)	Mean risk (%)	o/e ratio	Log intercept	Log slope
FIT-spline	1.453	1.408 (1.347, 1.471)	1.032 (0.988, 1.079)	0.034 (-0.156, 0.231)	0.998 (0.928, 1.074)
Original models					
Nottingham-fit	1.453	0.76 (0.724, 0.798)	1.911 (1.82, 2.007)	0.845 (0.632, 1.076)	1.028 (0.968, 1.092)
Nottingham-fit-age	1.453	0.694 (0.658, 0.729)	2.094 (1.992, 2.208)	1.034 (0.8, 1.274)	1.049 (0.991, 1.115)
Nottingham-fit-age-sex	1.453	0.697 (0.662, 0.733)	2.085 (1.983, 2.197)	1.032 (0.796, 1.266)	1.049 (0.99, 1.117)
Nottingham-lr	1.453	0.732 (0.693, 0.772)	1.986 (1.882, 2.098)	0.9 (0.675, 1.138)	1.018 (0.959, 1.083)
Nottingham-lr-boot	1.453	0.74 (0.707, 0.774)	1.963 (1.878, 2.054)	1.452 (1.179, 1.735)	1.184 (1.116, 1.259)
Nottingham-cox	1.453	0.73 (0.692, 0.77)	1.99 (1.886, 2.101)	0.917 (0.691, 1.159)	1.023 (0.965, 1.088)
Nottingham-cox-boot	1.453	0.729 (0.691, 0.768)	1.993 (1.892, 2.103)	0.992 (0.762, 1.24)	1.046 (0.986, 1.112)
Recalibrated models					
Nottingham-fit-platt	1.453	1.453 (1.387, 1.523)	1.0 (0.954, 1.048)	-0.0 (-0.164, 0.178)	1.0 (0.942, 1.062)
Nottingham-fit-quant	1.453	1.413 (1.413, 1.423)	1.029 (1.022, 1.029)	0.169 (-0.025, 0.39)	1.054 (0.979, 1.149)
Nottingham-fit-3.5	1.453	1.388 (1.329, 1.448)	1.047 (1.003, 1.093)	0.074 (-0.124, 0.273)	1.008 (0.935, 1.088)
Nottingham-fit-age-platt	1.453	1.453 (1.381, 1.525)	1.0 (0.953, 1.052)	-0.0 (-0.176, 0.179)	1.0 (0.944, 1.063)
Nottingham-fit-age-quant	1.453	1.3 (1.279, 1.328)	1.118 (1.095, 1.136)	0.236 (0.049, 0.438)	1.039 (0.972, 1.121)
Nottingham-fit-age-3.5	1.453	1.299 (1.238, 1.358)	1.119 (1.07, 1.174)	0.12 (-0.068, 0.313)	0.994 (0.931, 1.066)
Nottingham-fit-age-sex-platt	1.453	1.453 (1.382, 1.526)	1.0 (0.952, 1.052)	0.0 (-0.176, 0.18)	1.0 (0.944, 1.064)
Nottingham-fit-age-sex-quant	1.453	1.297 (1.277, 1.326)	1.12 (1.096, 1.138)	0.244 (0.055, 0.448)	1.041 (0.974, 1.124)
Nottingham-fit-age-sex-3.5	1.453	1.296 (1.236, 1.357)	1.121 (1.071, 1.175)	0.128 (-0.058, 0.327)	0.996 (0.933, 1.071)
Nottingham-lr-platt	1.453	1.453 (1.383, 1.524)	1.0 (0.953, 1.051)	-0.0 (-0.177, 0.183)	1.0 (0.943, 1.064)
Nottingham-lr-quant	1.453	1.313 (1.283, 1.354)	1.107 (1.073, 1.133)	0.167 (-0.015, 0.37)	1.017 (0.95, 1.096)
Nottingham-lr-3.5	1.453	1.291 (1.229, 1.353)	1.126 (1.074, 1.182)	0.099 (-0.088, 0.295)	0.981 (0.915, 1.053)

Notes. Log intercept and log slope are the intercept and slope of a logistic regression model that predicts cancer events from the logits of predicted probabilities of cancer. The calibration metrics were computed for several Nottingham models, and for an Oxford-derived FIT-only model that is used as a comparator in net benefit curves ('FIT-spline'). The Nottingham models include the full logistic model ('Nottingham-lr'), bootstrap-averaged full logistic model ('Nottingham-lr-boot'), full Cox model ('Nottingham-cox'), bootstrap averaged full Cox model ('Nottingham-cox-boot'), logistic model with FIT-age-sex as predictors ('Nottingham-fit-age-sex'), logistic model with FIT and age as predictors ('Nottingham-fit-age'), and a FIT-only logistic model ('Nottingham-fit'). The suffixes '**platt**', '**quant**' and '**3.5**' denote different recalibration methods applied to the Nottingham models (logistic recalibration, quantile transformation and constant multiplication, respectively - see Section 4.2.6).

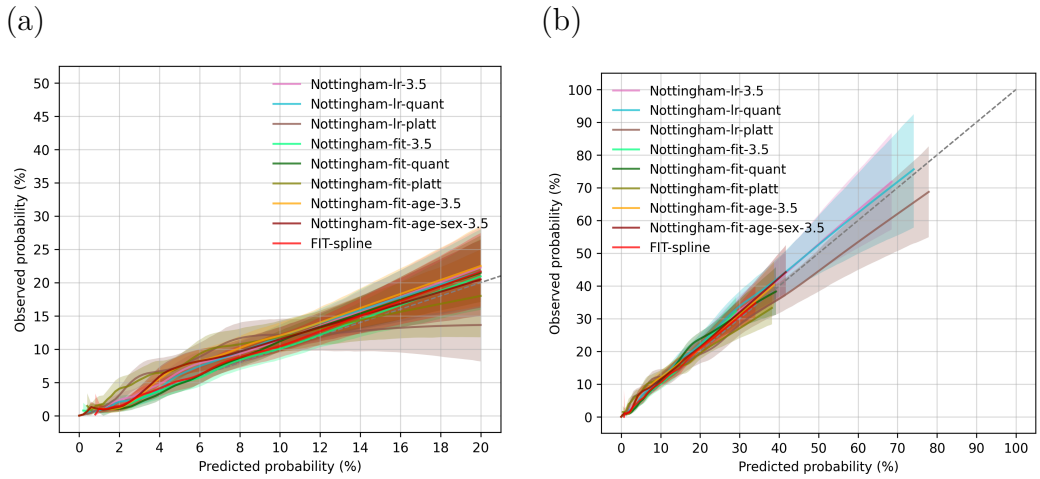


Figure 4.6: Smooth calibration curves for recalibrated Nottingham logistic models. Left panel shows calibration in the clinically meaningful range of risks (<20%, a), and right panel over the full range of risks (b). Curves were created by applying LOWESS-smoothing to predicted probabilities and cancer events. Shaded areas show 95% bootstrap percentile confidence intervals. FIT-spline is an Oxford FIT-only model. The suffixes '**platt**', '**quant**' and '**3.5**' denote different recalibration methods applied to Nottingham models (logistic recalibration, quantile transformation and constant multiplication, respectively - see Section 4.2.6). The models contain a different number of predictors: '**fit**' (FIT test only), '**fit-age**' (FIT and age), '**fit-age-sex**' (FIT, age and sex), and '**lr**' (full model with FIT, age, sex and blood tests).

4.3.5 Discrimination and net benefit by risk threshold

In this section, we present common diagnostic metrics and net benefit at selected levels of colorectal cancer risk (at selected predicted probabilities of colorectal cancer).

Common diagnostic metrics by cancer risk

Sensitivity, specificity, PPV and NPV at selected probabilities of colorectal cancer for the recalibrated Nottingham models and the Oxford FIT spline model are given in Table 4.8. The 2.5% risk threshold roughly corresponded to FIT test at threshold $\geq$ 10 µg Hb/g. The FIT test at threshold $\geq$ 10 had 83.5% sensitivity and 13.4% positive predictive value. At the equivalent 2.5% risk threshold, the sensitivities of Nottingham models ranged from 79.9% to 85.6% and positive predictive values ranged from 11.3 to 16.4%. Results for original Nottingham models are not included here, because the models were not calibrated and their predicted risks cannot be interpreted as probabilities of cancer.

NB. Some variability in the reported diagnostic metrics between models at the same risk threshold is due to miscalibration because it was not possible to perfectly recalibrate the Nottingham models. Therefore, when comparing the models at a particular risk threshold, it would thus be good to check that the compared models are similarly calibrated at that threshold, using figure 4.4.

Net benefit by cancer risk

Net benefit curves were created for predicted probabilities of colorectal cancer ranging from 0 to 10%, as this is the clinically meaningful range of risk (for example, a colorectal cancer risk of 3% may already be sufficient to trigger further investigation).

The recalibrated Nottingham models yielded higher net benefit than FIT test at threshold 10, for predicted risks ranging from about 5% to 10% (Figure 4.7). However, confidence intervals were overlapping in that range of risk.

The recalibrated Nottingham FIT-age, FIT-age-sex, and full models did not yield clearly higher net benefit than the Nottingham FIT-only model or the Oxford FIT-spline model in the clinically meaningful risk range from 0 to 10% (Figure 4.7).

Table 4.8: Sensitivity, specificity, positive predictive value, and negative predictive value at selected risk thresholds for recalibrated Nottingham models and the Oxford FIT-spline model

Model	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)
Risk 1%				
FIT-spline	89.8 (86.3, 93.0)	84.0 (83.5, 84.6)	7.6 (7.3, 8.0)	99.8 (99.8, 99.9)
Nottingham-fit-platt	84.5 (80.3, 88.4)	91.5 (91.1, 92.0)	12.8 (12.1, 13.6)	99.8 (99.7, 99.8)
Nottingham-fit-quant	89.8 (86.3, 93.0)	84.0 (83.5, 84.6)	7.6 (7.3, 8.0)	99.8 (99.8, 99.9)
Nottingham-fit-3.5	87.0 (83.1, 90.5)	88.2 (87.7, 88.6)	9.8 (9.3, 10.3)	99.8 (99.7, 99.8)
Nottingham-fit-age-platt	84.5 (80.3, 88.4)	91.3 (90.9, 91.7)	12.5 (11.8, 13.3)	99.8 (99.7, 99.8)
Nottingham-fit-age-quant	88.7 (84.9, 91.9)	85.7 (85.3, 86.0)	8.4 (7.9, 8.7)	99.8 (99.7, 99.9)
Nottingham-fit-age-3.5	85.9 (81.7, 89.8)	88.7 (88.2, 89.1)	10.0 (9.5, 10.6)	99.8 (99.7, 99.8)
Nottingham-fit-age-sex-platt	84.9 (80.6, 88.7)	91.4 (91.0, 91.8)	12.7 (11.9, 13.5)	99.8 (99.7, 99.8)
Nottingham-fit-age-sex-quant	88.4 (84.9, 91.9)	85.8 (85.4, 86.2)	8.4 (8.0, 8.8)	99.8 (99.7, 99.9)
Nottingham-fit-age-sex-3.5	85.9 (81.7, 89.8)	88.7 (88.2, 89.1)	10.1 (9.5, 10.6)	99.8 (99.7, 99.8)
Nottingham-lr-quant	88.7 (85.2, 91.9)	86.5 (86.1, 86.7)	8.8 (8.4, 9.2)	99.8 (99.7, 99.9)
Nottingham-lr-3.5	87.0 (83.1, 90.5)	88.9 (88.5, 89.4)	10.3 (9.8, 11.0)	99.8 (99.7, 99.8)
Nottingham-lr-platt	86.6 (82.7, 90.5)	89.6 (89.2, 90.0)	10.9 (10.3, 11.6)	99.8 (99.7, 99.8)
Risk 2%				
FIT-spline	85.2 (81.0, 89.1)	91.0 (90.6, 91.5)	12.3 (11.6, 13.1)	99.8 (99.7, 99.8)
Nottingham-fit-platt	81.3 (76.8, 85.6)	93.3 (92.9, 93.6)	15.1 (14.2, 16.2)	99.7 (99.6, 99.8)
Nottingham-fit-quant	86.3 (82.4, 90.1)	89.1 (88.0, 89.1)	10.4 (9.4, 10.8)	99.8 (99.7, 99.8)
Nottingham-fit-3.5	85.2 (81.0, 89.1)	90.6 (90.2, 91.1)	11.8 (11.2, 12.6)	99.8 (99.7, 99.8)
Nottingham-fit-age-platt	82.7 (78.2, 87.0)	93.3 (93.0, 93.7)	15.5 (14.4, 16.4)	99.7 (99.7, 99.8)
Nottingham-fit-age-quant	84.5 (81.0, 90.1)	90.4 (89.0, 90.5)	11.5 (10.2, 11.8)	99.7 (99.7, 99.8)
Nottingham-fit-age-3.5	84.2 (79.9, 88.0)	91.4 (91.0, 91.8)	12.6 (11.8, 13.3)	99.7 (99.7, 99.8)
Nottingham-fit-age-sex-platt	82.4 (77.8, 86.3)	93.3 (92.9, 93.6)	15.3 (14.2, 16.3)	99.7 (99.7, 99.8)
Nottingham-fit-age-sex-quant	85.9 (82.0, 89.8)	89.5 (89.2, 89.7)	10.8 (10.2, 11.2)	99.8 (99.7, 99.8)
Nottingham-fit-age-sex-3.5	84.9 (80.6, 88.7)	91.5 (91.1, 91.9)	12.8 (12.0, 13.5)	99.8 (99.7, 99.8)
Nottingham-lr-quant	85.6 (81.3, 89.4)	90.0 (89.6, 90.1)	11.2 (10.5, 11.6)	99.8 (99.7, 99.8)
Nottingham-lr-3.5	84.9 (80.6, 88.7)	91.6 (91.2, 92.0)	13.0 (12.2, 13.8)	99.8 (99.7, 99.8)
Nottingham-lr-platt	81.7 (77.1, 85.9)	93.4 (93.0, 93.7)	15.4 (14.4, 16.4)	99.7 (99.6, 99.8)
Risk 2.5%				
FIT $\geq$ 10	83.5 (78.9, 87.7)	92.1 (91.7, 92.5)	13.4 (12.6, 14.3)	99.7 (99.7, 99.8)
FIT-spline	83.5 (78.9, 87.7)	91.9 (91.5, 92.3)	13.2 (12.4, 14.0)	99.7 (99.7, 99.8)
Nottingham-fit-platt	79.6 (75.0, 84.2)	93.8 (93.5, 94.2)	15.9 (14.8, 17.1)	99.7 (99.6, 99.8)
Nottingham-fit-quant	85.6 (81.3, 89.4)	90.1 (90.0, 90.1)	11.3 (10.7, 11.8)	99.8 (99.7, 99.8)
Nottingham-fit-3.5	84.5 (80.3, 88.4)	91.6 (91.2, 92.0)	12.9 (12.2, 13.7)	99.8 (99.7, 99.8)
Nottingham-fit-age-platt	80.3 (75.7, 84.9)	93.9 (93.5, 94.3)	16.2 (15.1, 17.3)	99.7 (99.6, 99.8)
Nottingham-fit-age-quant	84.5 (80.3, 88.7)	91.0 (90.6, 91.2)	12.2 (11.4, 12.7)	99.7 (99.7, 99.8)
Nottingham-fit-age-3.5	83.8 (79.6, 87.7)	92.1 (91.8, 92.6)	13.6 (12.8, 14.4)	99.7 (99.7, 99.8)
Nottingham-fit-age-sex-platt	82.0 (77.5, 86.3)	93.8 (93.5, 94.2)	16.4 (15.2, 17.5)	99.7 (99.6, 99.8)
Nottingham-fit-age-sex-quant	84.5 (80.3, 88.4)	91.1 (90.7, 91.3)	12.3 (11.5, 12.8)	99.7 (99.7, 99.8)
Nottingham-fit-age-sex-3.5	83.8 (79.6, 87.7)	92.3 (91.9, 92.7)	13.8 (13.0, 14.7)	99.7 (99.7, 99.8)
Nottingham-lr-quant	84.2 (80.3, 88.0)	91.3 (90.9, 91.4)	12.4 (11.7, 13.0)	99.7 (99.7, 99.8)
Nottingham-lr-3.5	82.7 (78.2, 86.6)	92.4 (92.0, 92.8)	13.8 (13.0, 14.8)	99.7 (99.7, 99.8)
Nottingham-lr-platt	79.9 (75.0, 84.2)	94.0 (93.7, 94.4)	16.4 (15.3, 17.6)	99.7 (99.6, 99.8)
Risk 3%				
FIT-spline	83.1 (78.5, 87.0)	92.5 (92.1, 92.9)	14.0 (13.2, 14.9)	99.7 (99.7, 99.8)
Nottingham-fit-platt	77.5 (72.5, 82.0)	94.2 (93.9, 94.5)	16.5 (15.3, 17.7)	99.6 (99.6, 99.7)
Nottingham-fit-quant	85.2 (81.0, 89.1)	91.0 (91.0, 91.1)	12.3 (11.7, 12.9)	99.8 (99.7, 99.8)
Nottingham-fit-3.5	83.5 (78.9, 87.7)	92.2 (91.8, 92.6)	13.6 (12.8, 14.5)	99.7 (99.7, 99.8)
Nottingham-fit-age-platt	79.2 (74.3, 83.5)	94.3 (94.0, 94.7)	17.1 (15.9, 18.3)	99.7 (99.6, 99.7)
Nottingham-fit-age-quant	83.8 (79.2, 87.7)	92.0 (91.8, 92.1)	13.4 (12.6, 14.0)	99.7 (99.7, 99.8)
Nottingham-fit-age-3.5	83.1 (78.5, 87.3)	92.8 (92.5, 93.2)	14.6 (13.7, 15.6)	99.7 (99.7, 99.8)
Nottingham-fit-age-sex-platt	78.9 (73.9, 83.5)	94.4 (94.0, 94.7)	17.1 (15.9, 18.3)	99.7 (99.6, 99.7)
Nottingham-fit-age-sex-quant	84.2 (79.6, 88.0)	91.9 (91.8, 92.1)	13.3 (12.6, 14.0)	99.7 (99.7, 99.8)
Nottingham-fit-age-sex-3.5	82.7 (78.2, 87.0)	92.8 (92.5, 93.2)	14.6 (13.6, 15.5)	99.7 (99.7, 99.8)
Nottingham-lr-quant	81.0 (76.4, 85.6)	92.2 (91.9, 92.4)	13.3 (12.4, 14.0)	99.7 (99.6, 99.8)
Nottingham-lr-3.5	81.0 (76.4, 85.2)	93.1 (92.7, 93.4)	14.7 (13.8, 15.7)	99.7 (99.6, 99.8)
Nottingham-lr-platt	79.2 (74.3, 83.5)	94.5 (94.2, 94.9)	17.6 (16.3, 18.8)	99.7 (99.6, 99.7)
Risk 4%				
FIT-spline	80.3 (75.7, 84.5)	93.5 (93.2, 93.9)	15.5 (14.4, 16.5)	99.7 (99.6, 99.8)
Nottingham-fit-platt	75.7 (71.1, 80.3)	94.8 (94.5, 95.2)	17.8 (16.5, 19.2)	99.6 (99.6, 99.7)
Nottingham-fit-quant	83.5 (78.9, 87.3)	92.1 (92.0, 92.1)	13.4 (12.7, 14.1)	99.7 (99.7, 99.8)
Nottingham-fit-3.5	81.7 (77.1, 85.9)	93.2 (92.8, 93.5)	15.0 (14.0, 16.0)	99.7 (99.6, 99.8)
Nottingham-fit-age-platt	72.9 (67.9, 77.8)	95.1 (94.8, 95.4)	18.0 (16.5, 19.4)	99.6 (99.5, 99.7)
Nottingham-fit-age-quant	82.0 (77.1, 85.9)	93.4 (93.2, 93.5)	15.4 (14.4, 16.2)	99.7 (99.6, 99.8)
Nottingham-fit-age-3.5	80.3 (75.7, 84.5)	93.9 (93.5, 94.2)	16.2 (15.1, 17.3)	99.7 (99.6, 99.8)
Nottingham-fit-age-sex-platt	72.9 (68.0, 78.2)	95.1 (94.8, 95.4)	18.0 (16.7, 19.5)	99.6 (99.5, 99.7)
Nottingham-fit-age-sex-quant	80.3 (75.4, 84.5)	93.4 (93.2, 93.6)	15.2 (14.3, 16.0)	99.7 (99.6, 99.8)
Nottingham-fit-age-sex-3.5	80.3 (75.7, 84.5)	93.8 (93.5, 94.2)	16.1 (15.0, 17.2)	99.7 (99.6, 99.8)
Nottingham-lr-quant	79.2 (73.9, 83.5)	93.7 (93.4, 93.8)	15.5 (14.5, 16.4)	99.7 (99.6, 99.7)
Nottingham-lr-3.5	78.5 (73.6, 83.1)	94.2 (93.8, 94.5)	16.6 (15.4, 17.7)	99.7 (99.6, 99.7)
Nottingham-lr-platt	74.3 (69.4, 79.2)	95.2 (94.9, 95.5)	18.7 (17.3, 20.1)	99.6 (99.5, 99.7)
Risk 5%				
FIT-spline	77.5 (72.5, 82.0)	94.2 (93.8, 94.5)	16.4 (15.3, 17.6)	99.6 (99.6, 99.7)
Nottingham-fit-platt	72.2 (67.3, 77.1)	95.4 (95.1, 95.7)	18.8 (17.3, 20.4)	99.6 (99.5, 99.6)
Nottingham-fit-quant	77.8 (73.6, 82.7)	94.0 (94.0, 94.1)	16.2 (15.2, 17.2)	99.7 (99.6, 99.7)
Nottingham-fit-3.5	78.5 (73.9, 83.1)	93.9 (93.6, 94.3)	16.1 (15.0, 17.2)	99.7 (99.6, 99.7)
Nottingham-fit-age-platt	70.4 (65.5, 75.4)	95.6 (95.3, 95.9)	19.0 (17.5, 20.6)	99.5 (99.5, 99.6)
Nottingham-fit-age-quant	76.1 (70.8, 81.0)	94.7 (94.5, 94.8)	17.4 (16.2, 18.6)	99.6 (99.5, 99.7)
Nottingham-fit-age-3.5	76.8 (71.8, 81.3)	94.7 (94.4, 95.0)	17.6 (16.3, 19.0)	99.6 (99.6, 99.7)
Nottingham-fit-age-sex-platt	68.3 (63.4, 73.6)	95.6 (95.3, 95.9)	18.8 (17.2, 20.5)	99.5 (99.4, 99.6)
Nottingham-fit-age-sex-quant	74.3 (69.4, 79.6)	94.7 (94.5, 94.8)	17.1 (16.0, 18.2)	99.6 (99.5, 99.7)
Nottingham-fit-age-sex-3.5	76.4 (71.5, 81.0)	94.7 (94.4, 95.0)	17.5 (16.3, 18.9)	99.6 (99.6, 99.7)
Nottingham-lr-quant	77.1 (72.2, 81.7)	94.7 (94.5, 94.9)	17.7 (16.5, 18.7)	99.6 (99.6, 99.7)
Nottingham-lr-3.5	75.7 (70.8, 80.6)	94.9 (94.6, 95.2)	17.9 (16.6, 19.3)	99.6 (99.5, 99.7)
Nottingham-lr-platt	70.4 (65.5, 75.4)	95.8 (95.5, 96.1)	19.7 (18.1, 21.4)	99.5 (99.5, 99.6)
Risk 10%				
FIT-spline	65.8 (60.6, 71.1)	96.5 (96.3, 96.8)	21.8 (20.0, 23.7)	99.5 (99.4, 99.6)
Nottingham-fit-platt	58.8 (53.2, 64.4)	97.0 (96.7, 97.2)	22.4 (20.2, 24.5)	99.4 (99.3, 99.5)
Nottingham-fit-quant	63.4 (58.1, 69.0)	96.7 (96.6, 96.7)	21.9 (20.1, 23.9)	99.4 (99.4, 99.5)
Nottingham-fit-3.5	66.9 (61.6, 72.2)	96.4 (96.1, 96.7)	21.6 (19.8, 23.4)	99.5 (99.4, 99.6)
Nottingham-fit-age-platt	58.5 (52.5, 64.1)	97.4 (97.1, 97.6)	24.6 (22.2, 27.0)	99.4 (99.3, 99.5)
Nottingham-fit-age-quant	56.3 (50.7, 62.0)	97.3 (97.2, 97.5)	23.8 (21.5, 26.0)	99.3 (99.3, 99.4)
Nottingham-fit-age-3.5	58.1 (52.5, 64.1)	97.1 (96.8, 97.3)	22.7 (20.4, 25.0)	99.4 (99.3, 99.5)
Nottingham-fit-age-sex-platt	57.4 (52.1, 62.7)	97.4 (97.1, 97.6)	24.4 (22.0, 26.8)	99.4 (99.3, 99.4)
Nottingham-fit-age-sex-quant	56.3 (50.7, 61.6)	97.4 (97.3, 97.5)	24.2 (21.8, 26.4)	99.3 (99.3, 99.4)
Nottingham-fit-age-sex-3.5	58.1 (52.8, 63.7)	97.1 (96.8, 97.3)	22.6 (20.5, 25.1)	99.4 (99.3, 99.5)
Nottingham-lr-quant	57.0 (51.8, 62.7)	97.4 (97.2, 97.5)	24.3 (21.9, 26.8)	99.4 (99.3, 99.4)
Nottingham-lr-3.5	59.5 (53.9, 65.1)	97.1 (96.9, 97.4)	23.4 (21.2, 26.0)	99.4 (99.3, 99.5)
Nottingham-lr-platt	59.2 (53.5, 64.8)	97.4 (97.1, 97.6)	24.8 (22.4, 27.5)	99.4 (99.3, 99.5)

Notes. The diagnostic metrics were computed for several Nottingham logistic models, and for an Oxford-derived FIT-only model for comparison. The suffixes '**platt**', '**quant**' and '**3.5**' denote different recalibration methods applied to Nottingham models (logistic recalibration, quantile transformation and constant multiplication, respectively - see Section 4.2.6). The models contain a different number of predictors: '**fit**' (FIT test only), '**fit-age**' (FIT and age), '**fit-age-sex**' (FIT, age and sex), and '**lr**' (full logistic regression model with FIT, age, sex and blood tests). PPV - positive predictive value; NPV - negative predictive value.

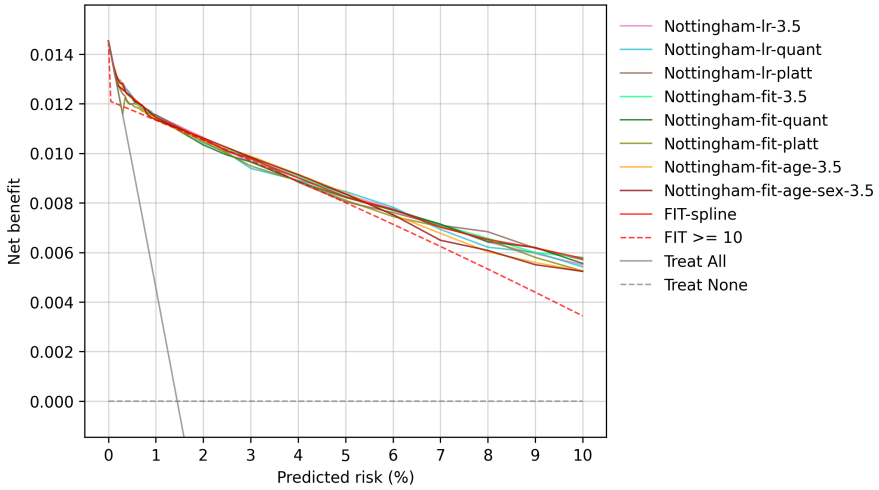
Net benefit and net colonoscopies avoided for 100,000 tests at a 2.5% risk threshold (corresponding to FIT at threshold 10) are shown in Table 4.9. Net benefit at various thresholds is shown in Table 4.10. Net benefit in these tables is obtained by multiplying the net benefit shown in the figures by 100,000.

Table 4.9: Net benefit per 100,000 tests for recalibrated Nottingham logistic models at a 2.5% risk threshold (the risk of cancer corresponding to FIT $\geq$ 10 µg/g), and for FIT $\geq$ 10 µg/g, "test all" and "test none" strategies

Model	Colono-scopies	Cancers detected	Cancers missed	Negative colono-scopies	Net cancers detected	Net colono-scopies avoided
Test all	100000.0	1453.4	0.0	98546.6	-1073.5	0.0
Test none	0.0	0.0	1453.4	0.0	0.0	41865.8
FIT $\geq$ 10	9032.3	1212.8	240.5	7819.5	1012.3	81346.9
FIT-only models						
Nottingham-fit-platt	7256.5	1156.5	296.8	6100.0	1000.1	80871.0
Nottingham-fit-quant	11017.9	1243.5	209.8	9774.3	992.9	80589.5
Nottingham-fit-3.5	9518.4	1228.2	225.2	8290.3	1015.6	81474.8
FIT and age models						
Nottingham-fit-age-platt	7195.1	1166.8	286.6	6028.4	1012.2	81341.8
Nottingham-fit-age-quant	10096.7	1228.2	225.2	8868.5	1000.8	80896.6
Nottingham-fit-age-3.5	8960.6	1218.0	235.4	7742.7	1019.4	81623.3
FIT, age and sex models						
Nottingham-fit-age-sex-platt	7287.2	1192.4	261.0	6094.9	1036.1	82273.2
Nottingham-fit-age-sex-quant	9994.4	1228.2	225.2	8766.2	1003.4	80998.9
Nottingham-fit-age-sex-3.5	8807.1	1218.0	235.4	7589.2	1023.4	81776.8
Full models						
Nottingham-lr-quant	9830.6	1223.1	230.3	8607.5	1002.4	80958.0
Nottingham-lr-3.5	8684.3	1202.6	250.8	7481.7	1010.8	81285.5
Nottingham-lr-platt	7067.2	1161.7	291.7	5905.5	1010.2	81265.0

Notes. Different strategies of referring 100,000 patients with suspected colorectal cancer to further investigations (whole colon examinations) are compared in this table: '**Test all**' - refer all patients; '**Test none**' - refer no one; '**FIT $\geq$ 10**' - refer patients who have a FIT value of at least 10 µg/g; '**Nottingham-...**' - refer patients if they have at least 2.5% probability of colorectal cancer according to one of several Nottingham models. All models are recalibrated versions of the original Nottingham models: the suffixes '**platt**', '**quant**' and '**3.5**' denote different recalibration methods (logistic recalibration, quantile transformation and constant multiplication, respectively - see Section 4.2.6). The models contain a different number of predictors: '**fit**' (FIT test only), '**fit-age**' (FIT and age), '**fit-age-sex**' (FIT, age and sex), and '**lr**' (full logistic regression model with FIT, age, sex and blood tests). A decision threshold of 2.5% was used for the models because it corresponds to the predicted probability of colorectal cancer for the FIT test at threshold 10 µg/g, allowing to evaluate the models at a similar level of risk as the standard clinical practice. '**Colonoscopies**' is the total number of patients that test positive under each testing strategy and would thus be referred. '**Cancers detected**' are true positives and '**Cancers missed**' are false negatives. '**Negative colonoscopies**' are false positives. Net benefit ('**Net cancers detected**') is computed as the number of true positives, minus the odds of cancer times false positives. '**Net colonoscopies avoided**' is computed as the number of true negatives, minus the odds of not having cancer times false negatives. The odds of cancer at a 2.5% predicted probability of cancer are 0.0256. Net cancers detected in this table corresponds to net benefit in Figure 4.7 at a risk threshold of 2.5%, multiplied by 100,000.

(a)



(b)

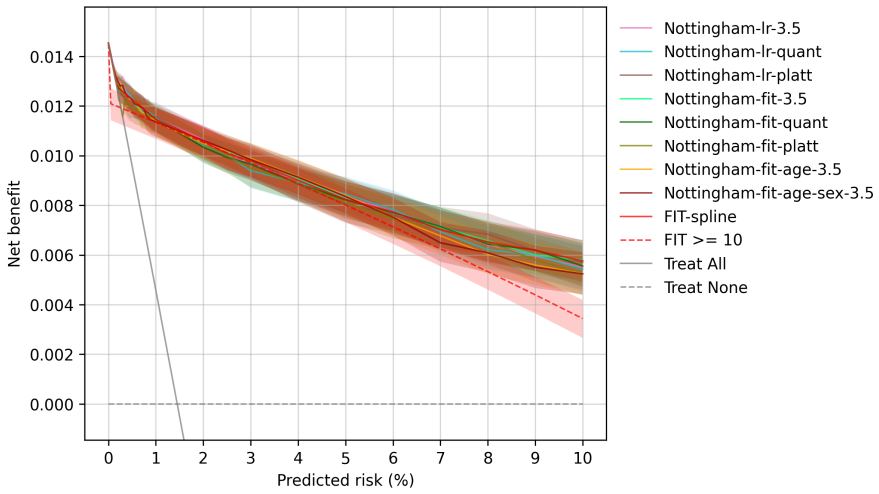


Figure 4.7: Net benefit curves for recalibrated Nottingham models and for FIT test at threshold ≥10. An individual risk of 0.15-0.5% (depending on the model) roughly corresponded to a positive predictive value of 3%. Curves are shown without (a) and with 95% bootstrap percentile confidence intervals (b). The FIT-spline is an Oxford FIT-only model that illustrates the net benefit gained from varying the FIT threshold, rather than keeping the threshold fixed at 10. Three different versions of the full Nottingham logistic model are shown, corresponding to different recalibration methods ('Nottingham-lr-3.5', 'Nottingham-lr-quant', 'Nottingham-lr-platt'). The three recalibration methods are similarly shown for the Nottingham logistic FIT-only model ('Nottingham-fit-3.5', 'Nottingham-fit-quant', 'Nottingham-fit-platt'). However, to simplify the figure, only the conversion factor approach to recalibration is shown for Nottingham logistic FIT-age and FIT-age-sex models ('Nottingham-fit-age-3.5' and 'Nottingham-fit-age-sex-3.5').

Table 4.10: Net benefit per 100,000 tests for recalibrated Nottingham logistic models and for FIT $\geq$ 10 µg/g, "test all" and "test none" strategies at various risk thresholds

Model	Net benefit	Net benefit (95% CI)	NI avoided	NI avoided (95% CI)
Risk 1%				
Test all	457.9	(457.9, 457.9)	0	(0.0, 0.0)
Test none	0	(0.0, 0.0)	-45335.4	(-45335.4, -45335.4)
FIT $\geq$ 10	1133.9	(1067.4, 1193.8)	66915.7	(60341.2, 72855.2)
FIT-spline	1145.6	(1093.9, 1194.8)	68082.5	(62963.4, 72950.6)
Nottingham-fit-platt	1143.8	(1089.5, 1202.6)	67903.4	(61632.8, 73717.8)
Nottingham-fit-quant	1145.6	(1093.9, 1194.8)	68082.5	(62963.4, 72950.6)
Nottingham-fit-3.5	1146.2	(1090.7, 1198.3)	68133.7	(62645.0, 73298.4)
Nottingham-fit-age-platt	1141.7	(1078.9, 1197.6)	67688.4	(61477.4, 73226.0)
Nottingham-fit-age-quant	1147.1	(1088.9, 1194.5)	68225.8	(62468.1, 72919.1)
Nottingham-fit-age-3.5	1135.7	(1076.0, 1191.9)	67099.9	(61187.0, 72664.3)
Nottingham-fit-age-sex-platt	1147.7	(1084.6, 1205.2)	68282.1	(62043.5, 73978.6)
Nottingham-fit-age-sex-quant	1143.3	(1089.1, 1193.0)	67852.2	(62488.9, 72775.4)
Nottingham-fit-age-sex-3.5	1135.8	(1075.6, 1192.1)	67110.2	(61146.2, 72678.6)
Nottingham-lr-quant	1154.8	(1100.9, 1203.8)	68993.4	(63649.5, 73841.0)
Nottingham-lr-3.5	1153.4	(1095.2, 1206.8)	68850.1	(63087.6, 74141.5)
Nottingham-lr-platt	1155.4	(1096.1, 1214.1)	69049.7	(63178.1, 74864.6)
Risk 2%				
Test all	-557.8	(-557.8, -557.8)	0	(0.0, 0.0)
Test none	0	(0.0, 0.0)	27332.3	(27332.3, 27332.3)
FIT $\geq$ 10	1053.3	(987.0, 1114.4)	78941.7	(75693.5, 81935.7)
FIT-spline	1058.4	(998.5, 1115.8)	79192.5	(76258.5, 82007.4)
Nottingham-fit-platt	1047	(976.6, 1110.4)	78634.7	(75185.0, 81741.0)
Nottingham-fit-quant	1033.9	(971.0, 1085.7)	77995	(74909.0, 80533.4)
Nottingham-fit-3.5	1050.3	(989.6, 1107.3)	78798.4	(75824.5, 81589.0)
Nottingham-fit-age-platt	1068.3	(1001.7, 1129.8)	79678.6	(76415.9, 82694.7)
Nottingham-fit-age-quant	1034.9	(976.1, 1095.5)	78041	(75162.5, 81010.3)
Nottingham-fit-age-3.5	1049.5	(985.8, 1106.7)	78757.5	(75634.6, 81562.5)
Nottingham-fit-age-sex-platt	1061.8	(994.9, 1122.0)	79361.3	(76080.3, 82309.4)
Nottingham-fit-age-sex-quant	1037.6	(978.3, 1095.3)	78174.1	(75267.1, 81004.2)
Nottingham-fit-age-sex-3.5	1061.4	(998.8, 1119.7)	79340.9	(76275.4, 82196.5)
Nottingham-lr-quant	1042.3	(980.1, 1096.3)	78404.4	(75359.0, 81050.4)
Nottingham-lr-3.5	1064.2	(1001.0, 1122.9)	79479	(76382.9, 82355.9)
Nottingham-lr-platt	1053.8	(987.0, 1116.3)	78967.3	(75696.5, 82033.2)
Risk 2.5%				
Test all	-1073.5	(-1073.5, -1073.5)	0	(0.0, 0.0)
Test none	0	(0.0, 0.0)	41865.8	(41865.8, 41865.8)
FIT $\geq$ 10	1012.3	(945.6, 1074.3)	81346.9	(78745.1, 83762.5)
FIT-spline	1008	(940.8, 1070.1)	81178	(78557.0, 83598.7)
Nottingham-fit-platt	1000.1	(929.4, 1066.9)	80871	(78112.7, 83475.8)
Nottingham-fit-quant	992.9	(930.3, 1050.8)	80589.5	(78148.4, 82846.5)
Nottingham-fit-3.5	1015.6	(951.5, 1076.0)	81474.8	(78972.4, 83828.9)
Nottingham-fit-age-platt	1012.2	(943.8, 1078.1)	81341.8	(78673.9, 83911.0)
Nottingham-fit-age-quant	1000.8	(938.2, 1057.6)	80896.6	(78454.8, 83113.2)
Nottingham-fit-age-3.5	1019.4	(955.9, 1078.5)	81623.3	(79146.3, 83926.5)
Nottingham-fit-age-sex-platt	1036.1	(968.5, 1098.0)	82273.2	(79637.6, 84688.7)
Nottingham-fit-age-sex-quant	1003.4	(939.0, 1059.1)	80998.9	(78486.0, 83169.0)
Nottingham-fit-age-sex-3.5	1023.4	(958.5, 1083.1)	81776.8	(79248.8, 84105.3)
Nottingham-lr-quant	1002.4	(938.5, 1059.6)	80958	(78465.7, 83189.4)
Nottingham-lr-3.5	1010.8	(944.9, 1070.7)	81285.5	(78716.0, 83624.8)
Nottingham-lr-platt	1010.2	(940.2, 1071.3)	81265	(78531.7, 83645.4)
Risk 3%				
Test all	-1594.5	(-1594.5, -1594.5)	0	(0.0, 0.0)
Test none	0	(0.0, 0.0)	51554.9	(51554.9, 51554.9)
FIT $\geq$ 10	971	(903.2, 1033.1)	82950.4	(80759.6, 84957.4)
FIT-spline	978.5	(912.8, 1040.5)	83194.3	(81068.4, 85198.9)
Nottingham-fit-platt	949	(873.6, 1017.9)	82240.8	(79801.2, 84466.9)
Nottingham-fit-quant	965.6	(902.1, 1022.8)	82774.7	(80722.6, 84626.5)
Nottingham-fit-3.5	975.3	(907.9, 1038.2)	83088.5	(80909.8, 85123.6)
Nottingham-fit-age-platt	978.4	(910.0, 1043.7)	83190.9	(80977.3, 85299.6)
Nottingham-fit-age-quant	973.7	(904.0, 1029.5)	83039.1	(80785.7, 84842.2)
Nottingham-fit-age-3.5	989	(922.0, 1051.6)	83532.1	(81367.1, 85555.2)
Nottingham-fit-age-sex-platt	974.6	(905.7, 1041.8)	83066.4	(80837.8, 85238.3)
Nottingham-fit-age-sex-quant	977	(910.0, 1034.9)	83143.1	(80977.8, 85016.7)
Nottingham-fit-age-sex-3.5	984.2	(915.2, 1047.4)	83376.8	(81145.2, 85419.2)
Nottingham-lr-quant	939.3	(869.6, 1003.3)	81925.2	(79673.2, 83995.5)
Nottingham-lr-3.5	965.9	(899.5, 1029.6)	82784.9	(80638.4, 84844.2)
Nottingham-lr-platt	984.6	(912.4, 1048.7)	83390.5	(81054.4, 85461.4)
Risk 4%				
Test all	-2652.8	(-2652.8, -2652.8)	0	(0.0, 0.0)
Test none	0	(0.0, 0.0)	63666.1	(63666.1, 63666.1)
FIT $\geq$ 10	887	(819.0, 951.0)	84954.7	(83321.9, 86490.1)
FIT-spline	900.9	(830.7, 967.4)	85287.3	(83603.4, 86884.2)
Nottingham-fit-platt	888.7	(816.4, 961.7)	84995.7	(83260.2, 86745.8)
Nottingham-fit-quant	887	(818.6, 947.0)	84954.7	(83311.9, 86395.1)
Nottingham-fit-3.5	906.2	(836.9, 970.9)	85415.3	(83752.0, 86967.0)
Nottingham-fit-age-platt	857.6	(784.5, 931.6)	84248.5	(82493.2, 86024.5)
Nottingham-fit-age-quant	919.4	(843.7, 976.8)	85732.6	(83915.4, 87109.3)
Nottingham-fit-age-3.5	915.4	(847.1, 982.8)	85635.3	(83997.2, 87252.8)
Nottingham-fit-age-sex-platt	858.7	(785.7, 936.9)	84274.1	(82523.0, 86152.2)
Nottingham-fit-age-sex-quant	894.9	(821.5, 958.0)	85144.1	(83383.3, 86659.3)
Nottingham-fit-age-sex-3.5	913.5	(844.4, 980.2)	85589.3	(83931.1, 87191.4)
Nottingham-lr-quant	890.9	(815.3, 953.3)	85046.8	(83234.4, 86546.4)
Nottingham-lr-3.5	901.5	(827.5, 968.7)	85302.7	(83526.7, 86914.8)
Nottingham-lr-platt	883.6	(811.3, 956.6)	84872.8	(83137.9, 86623.5)
Risk 5%				
Test all	-3733.3	(-3733.3, -3733.3)	0	(0.0, 0.0)
Test none	0	(0.0, 0.0)	70932.9	(70932.9, 70932.9)
FIT $\geq$ 10	801.3	(732.1, 867.3)	86157.3	(84841.9, 87411.2)
FIT-spline	823.9	(747.1, 895.6)	86587.2	(85127.9, 87948.8)
Nottingham-fit-platt	810.4	(734.8, 885.9)	86331.3	(84893.2, 87764.3)
Nottingham-fit-quant	822	(755.9, 897.2)	86551.4	(85295.7, 87979.2)
Nottingham-fit-3.5	827.1	(753.3, 898.5)	86648.6	(85246.1, 88004.8)
Nottingham-fit-age-platt	794.6	(721.0, 873.5)	86029.4	(84632.2, 87528.8)
Nottingham-fit-age-quant	828.8	(750.1, 902.9)	86679.3	(85184.2, 88087.1)
Nottingham-fit-age-3.5	840.6	(771.1, 910.4)	86904.5	(85583.9, 88230.1)
Nottingham-fit-age-sex-platt	766.5	(690.6, 846.8)	85497.2	(84053.8, 87022.9)
Nottingham-fit-age-sex-quant	803.7	(731.8, 883.4)	86203.4	(84836.8, 87718.3)
Nottingham-fit-age-sex-3.5	835.8	(766.8, 907.7)	86812.3	(85502.0, 88179.0)
Nottingham-lr-quant	845.7	(774.9, 914.2)	87001.7	(85655.5, 88302.0)
Nottingham-lr-3.5	835.2	(761.4, 908.8)	86802.1	(85399.4, 88199.3)
Nottingham-lr-platt	804.2	(728.8, 883.2)	86213.6	(84780.6, 87713.3)
Risk 10%				
Test all	-9496.3	(-9496.3, -9496.3)	0	(0.0, 0.0)
Test none	0	(0.0, 0.0)	85466.5	(85466.5, 85466.5)
FIT $\geq$ 10	344	(266.1, 419.1)	88562.5	(87861.4, 89238.1)
FIT-spline	576	(491.8, 659.6)	90650.4	(89893.0, 91402.7)
Nottingham-fit-platt	524.8	(438.9, 611.3)	90189.9	(89416.5, 90967.7)
Nottingham-fit-quant	555.5	(471.4, 647.6)	90466.2	(89708.8, 91295.2)
Nottingham-fit-3.5	580	(499.8, 660.8)	90686.2	(89964.4, 91413.8)
Nottingham-fit-age-platt	560.1	(474.2, 644.3)	90507.1	(89734.3, 91264.8)
Nottingham-fit-age-quant	527.7	(437.3, 614.7)	90215.4	(89401.8, 90998.5)
Nottingham-fit-age-3.5	524.8	(437.8, 613.0)	90189.9	(89406.6, 90983.1)
Nottingham-fit-age-sex-platt	546.4	(461.7, 631.2)	90384.3	(89621.7, 91147.1)
Nottingham-fit-age-sex-quant	533.9	(443.5, 614.7)	90271.7	(89457.9, 90998.7)
Nottingham-fit-age-sex-3.5	523.7	(441.8, 610.2)	90179.6	(89442.6, 90957.9)
Nottingham-lr-quant	541.9	(454.9, 634.0)	90343.4	(89560.3, 91172.4)
Nottingham-lr-3.5	550.4	(461.7, 640.3)	90420.1	(89621.4, 91229.1)
Nottingham-lr-platt	570.3	(480.5, 658.5)	90599.3	(89790.6, 91392.7)

Notes. Different strategies of referring 100,000 patients with suspected colorectal cancer to further investigations (colonoscopy) are compared in this table: '**Test all**' - refer all patients; '**Test none**' - refer no one; '**FIT $\geq$ 10**' - refer patients who have a FIT test value of at least 10 µg/g; '**Nottingham-...**' - refer patients if they have a prespecified probability of colorectal cancer (1%, 2%, ..., 10%) according to one of several Nottingham models. All Nottingham models are recalibrated versions of the original models: the suffixes '**platt**', '**quant**' and '**3.5**' denote different recalibration methods (logistic recalibration, quantile transformation and constant multiplication, respectively - see Section 4.2.6). The models contain a different number of predictors: '**fit**' (FIT test only), '**fit-age**' (FIT and age), '**fit-age-sex**' (FIT, age and sex), and '**lr**' (full logistic regression model with FIT, age, sex and blood tests). FIT-spline is an Oxford FIT-only spline model. **Net benefit** is computed as the number of true positives, minus the odds of cancer times false positives. '**NI**' (Net intervention avoided) is computed as the number of true negatives, minus the odds of not having cancer times false negatives. 95% CI refers to 95% bootstrap percentile confidence interval. Net benefit in this table corresponds to net benefit in Figure 1.7 multiplied by 100,000.

4.4 Discussion

4.4.1 Main findings

The purpose of this analysis was to evaluate the diagnostic performance of Nottingham colorectal cancer (CRC) risk prediction models in the Oxford University Hospitals FIT (OUH-FIT) dataset. The models were developed on patients who were offered FIT testing by their GP, and thus represent patients with symptoms indicative of CRC. The OUH-FIT dataset represents the same broad population of patients, and is therefore an intended setting of use. Nottingham models were assessed from three perspectives: discrimination, calibration, and net benefit.

Discrimination

Prediction models should distinguish cancer cases from non-cancer cases. In practice, a threshold is chosen to decide if someone tests positive. When the threshold is high, less cancers are captured, but patients who test positive are more likely to be cancer cases. The lower the threshold the more cancers are captured, but there are usually also more false positives. Examining the performance over a range of thresholds, we found that the overall discrimination of the Nottingham models in the OUH-FIT dataset was high (c-statistics ranged from 90.6 to 92.7%), although similar to the c-statistic of the FIT test alone (91.5%).

The positive predictive value of the Nottingham models, as compared to the FIT test alone, varied depending on the risk threshold. At lower risk thresholds (risks $< 15\%$) that would capture 45% to 100% of all cancers, the models had similar PPV as the FIT test. If the purpose of the model is to reduce the number of false positives while detecting a high number of cancers (the standard practice of using FIT at threshold ≥ 10 µg Hb/g faeces led to the detection of 84% of all cancers in Oxford data), there does not seem to be benefit of using the models instead of the FIT test. At higher risk thresholds (risks $> 15\%$) where less than 45% of cancers would be captured, the models had a higher positive predictive value than FIT. This is probably because these risk thresholds correspond to FIT values much higher than the commonly used threshold of 10 µg/g, so at these thresholds

there are more patients with a negative FIT result and cancer, and presumably the additional variables included in models help to distinguish cancers among this subgroup. This finding is unlikely to be clinically useful, however, as the purpose of testing symptomatic patients in primary care is to identify most cancers.

Calibration

A model also needs to be calibrated to be useful in practice as an individual's risk of cancer predicted by the model needs to correspond to the actual risk cancer. For example, if the model predicts that someone's risk of cancer is 3%, but the actual risk is 6%, the model may lead to inappropriate decisions not to investigate or refer. We found that:

- All Nottingham models were poorly calibrated in the OUH-FIT data: predicted risks were approximately half of the actual risk in the clinically meaningful risk range (0-20%). For example, a predicted colorectal cancer probability of 5% corresponded to an actual probability of 10% in Oxford patients.

- It was possible to recalibrate the Nottingham models by making the Oxford FIT test distribution similar to Nottingham FIT distribution using two approaches: (i) quantile transformation; and (ii) by applying a conversion factor of 3.5. This indicates that miscalibration was likely caused by differences in the FIT test distributions.

- The difference in FIT test distributions was probably caused by the different FIT sensors used in Nottingham (OC-Sensor PLEDIA) and Oxford (HM-JACKarc) as a within patient dual sensor analysis performed in Nottingham also showed that the OC-sensor FIT values were on average 3.5 times higher than HM-JACKarc values (REF).

- The difference in FIT test distributions was probably not caused by differences in cancer risk, as prevalence was similar in Oxford and Nottingham (about 1.39% in all of Oxford data, 1.45% in the Oxford complete case analysis, and 1.56% in Nottingham). The discrepancy in FIT distributions was also

probably not caused by differences in FIT reporting ranges [Nottingham OC-Sensor PLEDIA <4-50,000 µg/g, Oxford HM-JACKarc <1-400 µg/g], because miscalibration remained when lower FIT values (0-200 µg/g) were used in analysis.

Net benefit

Net benefit at a particular risk threshold is proportional to the number of true positives (detected cancers), minus the number of false positives weighed by the odds of cancer at that threshold. The (recalibrated) Nottingham models yielded a higher net benefit in the OUH-FIT dataset than FIT ≥10 µg Hb/g faeces for risks ranging from 5% to 10%. However, at risk thresholds under 5% there was no clear difference in net benefit. The full Nottingham models, and FIT-age and FIT-age-sex models, did not yield a clearly higher net benefit than the Nottingham FIT-only model or the Oxford FIT-spline model in the clinically meaningful risk range from 0 to 10%.

Comparison of Nottingham logistic and Cox models

The Nottingham logistic and Cox models had highly similar discrimination: they predicted almost identical risks of cancer for all patients. Calibration was also very similar; only the bootstrap-averaged logistic model had slightly worse calibration than other models. It is possible that Cox models lose their unique advantage over logistic regression when they are used for binary classification: the goal was to predict if cancer occurs within a fixed period of time from the first FIT test (365 days), so the censored individuals who have less than 365 days of data might not contain any useful information for estimating that quantity, even though the Cox models use the data of these censored individuals to estimate survival probabilities when less than 365 days have passed.

Comparison to simpler Nottingham models

The full Nottingham models performed similarly to the simpler FIT-age and FIT-age-sex models at sensitivities greater than 50%. At lower level of sensitivity, the

full model that additionally includes platelets and MCV had higher point estimates of PPV than the FIT-age and FIT-age-sex models.

4.4.2 Limitations due to sample size

This external validation study was conducted on a relatively large sample (19,541 patients, 284 cases of cancer), and indeed the confidence interval (CI) widths observed for the overall discrimination and calibration metrics were reasonably narrow. For example, CI widths were less than 4% for the *c*-statistic on percentage scale, <11% for average precision on percentage scale, <0.3 for o/e ratio, and <0.2 for calibration slope. This study is also likely to meet Riley et al's minimum sample size requirements for external validation, as in their worked example, sample size was less than 10,000 for all statistics they considered when the rate of positive events was 1.8%, which is similar to the rate of cancer observed in this study [120].

The most important parameter in our study is arguably the positive predictive value at the level of sensitivity corresponding to FIT test at threshold 10 µg/g, as it determines if the model can lead to a reduced number of false positives compared to standard clinical practice. FIT test at threshold 10 µg/g had positive predictive (PPV) value of 13.4%, and Nottingham models had PPVs between 12.6-14.5% (at most about a 1% gain in PPV over FIT), with confidence intervals of the difference overlapping with 0. Note that at the 13.4% level of sensitivity, a 1% gain in PPV leads to about 9% reduction in false positives, and a 5% gain to 30% reduction. The bootstrap confidence interval width for the gain in PPV relative to FIT at the sensitivity of FIT ≥10 µg/g was approximately 5% for the full Nottingham logistic model. This implies that the sample size was probably large enough to detect larger reductions in false positives, but probably insufficient to capture smaller gains.

4.4.3 Lack of discrimination in Oxford data

In the Oxford dataset, the Nottingham models did not lead to a significantly reduced number of positive tests (a proxy for colonoscopies), compared to the standard practice of referring patients with a FIT result of at least 10 µg Hb/g. However, the

Nottingham team reported a 39% reduction in colonoscopies relative FIT $\geq$ 10 µg Hb/g when they evaluated their model on a more recent sample of Nottingham data that was not used for developing the models [118] . How to reconcile this difference?

The Oxford and Nottingham datasets were similar in several aspects: the setting of FIT testing (primary care), the prevalence of colorectal cancer (1.45% in Oxford, 1.56% in Nottingham); the sex distribution; and the quartiles of age, platelets and MCV (Table 4.3). The distribution of ethnic categories was also broadly similar (70.3% white, 8.7% non-white, 20.6% not recorded in Nottingham; 74.5% white, 4.5% non-white, 20.9% not recorded in Oxford; Table 4.3). The slightly higher percentage of known non-white individuals in Nottingham (4.2% higher) is unlikely to explain why the model performed worse in Oxford data because it is a small difference and because in Nottingham data the *c*-statistic of the model was similar for different ethnic categories (0.94 for white, 0.9-0.93 for other groups [119]). The relationship between each predictor variable used in Nottingham models and the risk of CRC in Oxford data was also broadly similar to the contribution that these variables make to the linear predictor of Nottingham models (Appendix B.1). Even though the scale of FIT values was very different in Nottingham and Oxford due to the use of different faecal analysers, when the Oxford FIT distribution was made more similar to the Nottingham distribution by quantile transformation, or by multiplying FIT values with a constant, then these recalibrated Nottingham models did not still have higher PPV at same level of sensitivity as FIT $\geq$ 10 µg/g.

However, despite having a similar prevalence of colorectal cancer (CRC), the percentage of patients with a positive FIT result was about twice as low in Oxford (9%) than in Nottingham (22%). Assuming that data quality is high in both centres (e.g. that most cases of CRC were correctly identified), this difference in FIT positivity combined with a similar rate of cancer implies that FIT was less precise for detecting CRC in Nottingham than in Oxford. Indeed, the positive predictive value of FIT $\geq$ 10 in Nottingham was 6.1% , while in Oxford it was 13.7%, and the estimated probability of cancer for a FIT value of 10 µg Hb/g was 0.6% in Nottingham and 2.5% in Oxford. It is therefore possible that the

Nottingham dataset has a higher proportion of serious bowel diseases other than CRC that lead to an elevated FIT result. If these other diseases are associated with a different platelet count and MCV than colorectal cancer, then the inclusion of platelets and MCV in Nottingham models could make the models more useful in Nottingham than in Oxford data. For example, non-cancerous polyps can be associated with an elevated FIT result [139], and perhaps some of these polyps would also be associated with a lower platelet count or higher MCV than CRC. This can be further investigated by examining if platelets and MCV are more useful for detecting CRC in FIT positive patients in Nottingham than in Oxford, and studying the prevalence of other diseases among the FIT positive patients who do not have CRC.

The hypothesised larger proportion of non-cancerous bowel diseases in Nottingham could be due to differences in socioeconomic status: 9.3% of Oxford population was income-deprived compared to 19.9% in Nottingham in 2019 [140]. However, it could also be due to the local practices of how FIT test is used: perhaps Nottingham GPs offer FITs to patients with higher-risk symptoms (who in Oxford would have been referred without requiring FIT). In Oxford, FIT was offered to lower-risk patients up until the COVID-19 pandemic; then the local guidance changed and it started to be also offered to patients with higher-risk symptoms such as rectal bleeding who would have otherwise been directly referred [141].

4.4.4 Statistical significance versus predictive power

The Nottingham models were originally developed using the fractional polynomials algorithm, which automatically selects variables to be included in the model and transformations for each variable, based on statistically significant improvements in model fit [142]. However, as the size of datasets used for developing prediction models increases (e.g. there were more than 30,000 patients in Nottingham data), the statistical power for detecting small effects also increases. Variable selection algorithms that assess statistical significance of model fit can therefore select variables whose (real) association with colorectal cancer is so small that their contribution to discriminating cancer cases from non-cancer cases is very small or

negligible. Even though identifying and describing these small effects can be of scientific interest, explanation is not the same as predictive power [143]. It can be argued that a clinical prediction model that contains less variables is potentially more useful than a more complex model that performs only a little bit better, because a complex model can be more costly to implement and more likely to fail if the setting of its use changes (e.g. if a hospital changes the measurement device of a blood test that is included in the model). When developing a prediction model on large datasets, it could therefore be useful to assess alternative versions of the model that include less variables, to see if the sparser models perform similarly. One way to achieve this is to use lasso regression, where the penalty strength parameter can be used to increase sparsity, although this will likely require a form of data splitting (such as cross-validation) to choose the appropriate level of sparsity.

4.4.5 External validity of FIT-test based prediction models

It is generally recommended that prediction models be evaluated in datasets that were not used for developing the models, to demonstrate 'external validity'[144]. However, models can be valid only in the context of a specific target population or setting, and it is not meaningful to talk about models being 'valid' in general [145]. Sperrin et al therefore propose the concept of 'targeted validation' which means evaluating model performance in the intended population or setting [145].

The Nottingham colorectal cancer risk prediction models are intended to be used for patients who display symptoms indicative of CRC in primary care, so the patients included in the OUH-FIT dataset are clearly a relevant target population. However, this external validation exercise has indicated that there can be important differences between patients within this group: the type of FIT analyser used, and potentially the prevalence of diseases other than CRC that are associated with a positive FIT result. These factors—and also others such as the prevalence of CRC—could strongly affect the calibration and discrimination of any FIT-test based prediction model in a new region or hospital. In a future study, it could be valuable to analyse the Nottingham and Oxford datasets more thoroughly, to understand

why a prediction model works in one but not in the other. Knowing "how, when, and why" a model works is arguably even more useful than demonstrating its validity or invalidity within a target population [146]. If characteristics that affect the performance of FIT-based models are identified, it would also be useful to know how these vary between primary care populations to understand if a prediction model developed in a single site could be applied at multiple sites. However, even if a model does not generalise geographically, it could still be clinically useful locally [146].

On this path effort never goes to waste

—Bhagavad Gita, translated by Eknath Easwaran

5

Combining the faecal immunochemical test with routinely collected data to predict the risk of colorectal cancer: A machine learning approach

Contents

5.1 Introduction

5.1.1 Motivation

The faecal immunochemical test (FIT) is recommended by the National Institute for Health and Care Excellence (NICE) for triaging patients with symptoms of colorectal cancer for further investigations [7]. These investigations commonly include colonoscopy, an invasive endoscopic examination of the lower gastrointestinal tract. FIT test measures the amount of haemoglobin (Hb) in stool, and results at least 10 µg of Hb per gram are considered positive. At the 10 µg Hb/g threshold, FIT has high sensitivity (89%) and specificity (81%) for detecting colorectal cancer [115]. However, only about 1 in 6 individuals who test positive have cancer [116], and the colonoscopy services suffer from a backlog of cases due to the COVID-19 pandemic. It is therefore desirable to improve the positive predictive value of the test, to both reduce the number of potentially unnecessary invasive investigations, and to ensure that cases of cancer are detected as early as possible. A potential way of improving on the FIT test is to build a prediction model that combines FIT with other routinely collected data (such as age, sex, and blood tests), and several such models have already been created. However, the existing models are limited because they were developed on patients that were already referred,

they were not thoroughly evaluated against the FIT test alone, or when evaluated against FIT they did not perform better.

In this work, we explore whether prediction models can detect cancer better than the FIT test alone by (1) including a more diverse set of clinical variables for predicting the risk of cancer, (2) employing machine learning models with different degrees of interpretability and flexibility, and (3) using novel ways of fitting these models to data.

Using a diverse set of predictor variables—such as blood tests, diagnosis codes, and procedure codes—could help discover previously unconsidered relationships. Employing machine learning algorithms other than logistic regression can lead to better performing models because the relationship between these clinical variables and risk of cancer could well be nonlinear (for example, a blood test result could have a U-shaped relationship with cancer, being informative only when its value is too high or low). It is also of interest to explore machine learning models with different degrees of interpretability and flexibility: the most flexible (and least interpretable) models help assess how much information the predictor variables together hold about the risk of cancer, whereas interpretable models help evaluate how much of the performance can be retained when some of the flexibility is sacrificed in return for intelligibility. An interpretable model can also be clinically more desirable than a black-box model, even if it performs worse. Finally, all models need to be fitted to the data, and in the case of a binary classification problem like ours this is commonly done by maximizing the logarithm of predicted probabilities. However, when the number of positive examples is much lower than the number negative examples in the data, then directly maximizing the area under the receiver-operating characteristic (ROC) curve or area under the precision-recall (PR) curve using novel methods could yield better results.

5.1.2 Existing colorectal cancer risk prediction models

Models for predicting the risk of colorectal cancer have been previously developed both on symptomatic patients and on other, usually broader populations (Table

5.1). The first category of models is directly relevant to this study, while the second category provides information about which additional predictor variables may be useful.

Models developed on symptomatic patients

The simplest model developed on symptomatic patients is the FIT, age, and sex score (FAST), derived from a logistic model [147]. Subsequent studies indicated it could lead to 9-21% reduction in colonoscopy demand when used on referred patients [148, 149]. However, these studies did not recommend the model, partly because it had a high false positive rate when applied to all symptomatic patients to guide referral (rather than further triaging patients already referred). Furthermore, even if the model would lead to reduced colonoscopy demand when applied to referred patients, it might not do better than simply using the FIT test again at a higher threshold for the same patients. Indeed, the ROC curve of the FAST model closely followed the ROC curve of the FIT test on the same dataset that the original model was developed on [150], and external validation of another FIT-age-sex model developed on Nottingham data showed that its ROC and precision-recall curves were almost identical to those of the FIT test (see Chapter 4). The original FAST model was developed on referred symptomatic patients, which could explain why it had high false positive rate on all symptomatic patients.

The other models developed on referred symptomatic patients are the COLONPREDICT [150] and COLONOFIT [151] scores, also derived from logistic models. COLONPREDICT uses FIT, age, sex, clinical symptoms, treatment with aspirin, and history of colonoscopy as predictor variables; while COLONOFIT uses FIT, age, sex, smoking status, and history of colonoscopy (Table 5.1). These models were developed on referred patients, so their thresholds for what counts as a positive test need to be set low to yield very high levels of sensitivity (>99%), to not miss cancers that would be detected by existing referrals. COLONPREDICT achieved approximately 7.3% higher positive predictive value than the FAST score near the 100% level of sensitivity [150], but this result can be optimistic because the model

was evaluated on the same dataset it was developed on. COLONOFIT score had a higher *c*-statistic than the FAST score, but at high levels of sensitivity where more than 99% of all cancers would be detected, the ROC curves were similar [151].

The FAST, COLONPREDICT, and COLONOFIT scores were developed on relatively small samples (867-1572 patients) which can limit their generalisability. Furthermore, the models were developed on patients already referred, which can restrict the range and reduce the predictive potential of input variables. In contrast, Withrow et al [125] and Crooks et al [119] fitted models to data of nearly all patients who were offered FIT testing by their GP (16,604 and 34,231 individuals, respectively). Utilising an Oxford dataset, Withrow et al found that including age, sex, and routinely collected blood tests in models did not lead to significantly higher positive predictive value (PPV) at the same level of sensitivity as the FIT test at threshold 10. The most recent results of Crooks et al on a Nottingham dataset indicated a 5% reduction in colonoscopies at the same level of sensitivity as FIT at threshold 10 on model derivation data, and a 39% reduction on an internal validation data [118]. However, the model developed on Nottingham data did not perform significantly better than the FIT test on Oxford data (see Chapter 4). The lack of performance in Oxford data could be due to differences in patient populations: the Nottingham dataset had more patients with a positive FIT result but no CRC, and the blood tests may have been useful for discriminating CRC within this subgroup and hence led to an overall improved performance over FIT (see Discussion 4.4.3).

In summary, there is currently not enough evidence that existing prediction models developed for guiding the referral of symptomatic patients consistently perform better than the FIT test. The model architectures used have also been limited to logistic and Cox models, which restricts the type of predictive patterns that can be learned from data.

Models developed on other populations

Models developed on a broader population can be useful for earlier detection of colorectal cancer even before the symptoms appear. The COLONFLAG model is a

decision tree ensemble that uses age, sex, complete blood count, and time trends in complete blood count to predict the risk of CRC; it achieved a *c*-statistic of 0.81 in UK primary care data when the most recent full blood count was 3-6 months before diagnosis [152]. In a subsequent evaluation, it was found that the *c*-statistic was 0.78 for a 18-24 month interval (which would allow for more time to intervene if CRC is detected), and most of the performance was due to the inclusion of age variable [153]. Virdee et al developed a simpler joint mixed-effects and Cox model that used age and time trends in haemoglobin, mean cell volume and platelets [154]; their model achieved a *c*-statistic of 0.75-0.76 for predicting CRC 21-27 months after the baseline full blood count and performed as well as COLONFLAG on the same dataset. Hippisley-Cox et al developed a Cox proportional hazards model on primary care data that incorporated alcohol consumption, history of gastrointestinal cancer, low haemoglobin, and clinical symptoms into its risk score, achieving an AUC of 0.89-0.91 for predicting CRC within a 2-year period from baseline. That model may be less useful as the previous two for early detection of cancer, as it requires individuals to already have some symptoms indicative of CRC [154].

Other models were derived on a bowel screening population. Cooper et al [155] trained logistic and neural network models to detect CRC or advanced adenoma within 28 days from the FIT test in patients referred to colonoscopy from a bowel screening programme. The models incorporated FIT test results, age, sex, deprivation and screening history, and achieved *c*-statistics of 0.66 and 0.69, respectively. Cooper et al [156] also developed a Cox model that employed demographics, screening data, blood test results, comorbidities and lifestyle information, predicting colorectal cancer within 2 years from the faecal occult blood test (FOBT) result with a *c*-statistic of 0.86.

There are several additional machine learning models that utilise numerous predictor variables, reviewed by Burnett et al [157]. Overall, these studies indicate that both routinely collected bloods and clinical symptoms can be useful for predicting the risk of colorectal cancer, although it is not clear if the additional

variables remain relevant when applied to patients with symptoms of colorectal for whom the FIT test result—a strong predictor of cancer—is also available.

Table 5.1: Existing colorectal cancer risk prediction models

Model	Year	Predictor variables	Type	Population	Reference
Models developed on symptomatic patients					
FAST	2017	FIT, age, sex	logistic	symptomatic patients referred to colonoscopy	[147–149]
COLONPREDICT	2016	FIT, age, sex, haemoglobin, aspirin, history of colonoscopy, symptoms (rectal mass, rectal bleeding, change in bowel habit), benign anorectal lesion	logistic	symptomatic patients referred to colonoscopy	[150, 158]
COLONOFIT	2019	FIT, age, smoking status, history of colonoscopy	logistic	symptomatic patients referred to colonoscopy	[151]
Withrow et al	2022	Model 1: FIT, age, sex, ferritin, platelets, c-reactive protein Model 2: FIT, sex, mean cell volume	logistic	patients offered FIT by their GP	[125]
Crooks et al	2022	FIT, age, sex, platelets, mean cell volume	logistic, cox	patients offered FIT by their GP	[119]
Models developed on other populations					
COLONFLAG	2017	Age, sex, complete blood count, trends in complete blood count over 18 and 36 months	decision tree ensemble	individuals aged >40 with full blood count	[152, 153]
Virdee et al	2022	Age; trends in haemoglobin, mean cell volume and platelets	joint model of a linear mixed-effects model and a Cox ph model	individuals aged >40 with at least one record for haemoglobin, mean cell volume, and platelets	[154]
Hippisley-Cox et al	2012	Alcohol consumption, history of gastrointestinal cancer, haemoglobin, symptoms (rectal bleeding, abdominal pain, appetite loss, weight loss, change in bowel habit)	Cox ph	individuals aged >30 without gastrointestinal symptoms in last 12 months	[159]
Cooper et al	2018	FIT, age, sex, deprivation, colorectal cancer screening history	logistic; artificial neural network	Individuals with FIT >20 µg/g referred to colonoscopy from bowel screening	[155]
Cooper et al	2020	Faecal occult blood test (FOBT) result, previous FOBT results, haemoglobin, platelets, mean cell volume, ferritin, diabetes, inflammatory bowel syndrome, age, sex, smoking status, alcohol consumption, BMI, deprivation, family history of gastrointestinal cancer	Cox ph	Individuals invited to bowel screening	[156]

5.2 Methods

5.2.1 Ethics

This work is based on the Oxford University Hospitals (OUH) FIT dataset (OUH-FIT), the use and collation of which was described in detail in the Data Protection Impact Assessment (DPIA) form that was approved by the OUH Information Governance team. The study was registered in OUH as a service evaluation (under CSS-BIO-3-4730, later updated as 9076).

5.2.2 Extracting clinical data

Clinical data was extracted from the Oxford University Hospitals (OUH) NHS Foundation Trust (FT) system for patients who had a colorectal cancer diagnosis code (C18-C20) or a FIT test request, as part of the FIT Bowel Screening project. Patients were excluded if they did not have a quantitative FIT test result, their FIT was not requested by GP, they were less than 18 years old, they did not have a 180-day follow-up period after the first FIT, they had colorectal cancer before the earliest FIT, or they did not have records for certain common blood tests (haemoglobin, mean cell volume, platelets, white cells).

Identifying cases of colorectal cancer

Individuals were considered to have colorectal cancer if they had an inpatient or outpatient ICD-10 diagnosis code (C18-C20), or if they had a pathology report that described current colorectal cancer. The date of colorectal cancer was chosen to be the earliest date among inpatient diagnosis codes, outpatient diagnosis codes, dates of colorectal cancer treatments, and the date at which the pathology report was received. T-stages for colorectal cancer were extracted from imaging and pathology reports using a rule-based algorithm that first detected all TNM phrases (such as 'pT1/2 N0 M0') and then extracted individual values. (See Section 4.2.3 in Chapter 4 for more detail.)

Extracting clinical symptoms

Clinical symptoms relevant for colorectal cancer (such as abdominal pain) were extracted from free text using regular expressions based on a comprehensive set of keywords developed by Withrow et al [125]. Irrelevant symptom keywords (such as negated keywords like 'no abdominal pain') were filtered out using a ConText-like algorithm [57]. Relevant existing spelling variations (including spelling mistakes) for symptom keywords were detected and incorporated using a spell-checker based on Peter Norvig's code [106]. Results were validated by examining the included and excluded matches for symptom keywords along with surrounding text.

5.2.3 Machine learning analysis

In the machine learning analysis, almost all potentially relevant predictor variables were included in models. Even if the number of variables is larger than the number of cancer cases, this does not necessarily lead to overfitting (optimistic) models, because all models were regularised. In addition, models were always evaluated on data they were not trained on, to ensure that performance estimates were not optimistic even if the model would overfit the training dataset.

Outcome variable

Outcome predicted by the models was the presence of colorectal cancer (yes/no) detected within 180 days from the earliest FIT result date. An additional sensitivity analysis for 365-day follow-up was conducted.

Predictor variables

We firstly assessed the predictive potential of more commonly collected data. We thus included FIT test results, age, sex, clinical symptoms reported in the FIT test request, and all blood tests available for at least 20% of patients (6300 patients). Blood test results were included if they occurred within 60 days before or 30 days after the earliest FIT test (but always before or at the first instance of colorectal cancer). Categorical variables were transformed to indicators (for example, an

indicator was created for the presence of each clinical symptom), and the first category was dropped. After removal of highly correlated variables (see below), this yielded 49 predictor variables.

We also assessed the predictive potential of more rarely and less reliably recorded data: we additionally included body mass index, multiple deprivation index, ethnicity, all blood tests available for at least 100 patients, and all diagnosis (ICD-10) codes, procedure codes (OPCS-4), and prescription codes reported for at least 100 patients. Blood tests were again included if they occurred within 60 days before or 30 days after the earliest FIT test; diagnosis codes, procedure codes, prescriptions, BMI and deprivation measurements were included if occurring within 365 days before the earliest FIT. Only first three characters were retained for diagnosis and procedure codes (letter and two digits), to avoid capturing too much detail. This yielded 582 predictors.

We additionally fitted some machine learning models using only 3 predictor variables: FIT test results, age, and sex. This helps to better assess the gain in performance from including the common and more rarely collected data.

Removal of highly correlated predictor variables

Blood tests that had a monotonic (Spearman) correlation of at least 0.7 to another blood test, and that were not among the "core" bloods, were removed. This was to facilitate multiple imputation and to simplify interpretation. In each pair of correlated blood tests, the one with more missing values was removed. The "core" blood tests were specifically of interest and included haemoglobin, platelets, white cell count, mean cell volume, serum ferritin, and C-reactive protein. In addition, all indicator variables that were identical to at least one other indicator were removed.

Data transformations for predictor variables

Transforming the input data is essential for creating optimal machine learning models, as the gradient-based optimisation methods that are used for training these models are sensitive to the distribution of the input data. Two transformations were considered for continuous variables: (1) log-transformation followed by robust

scaling, and (2) quantile-transformation to Gaussian distribution. In the first method, all non-negative continuous variables were first log-transformed, and then standardised by subtracting the median and dividing by interquartile range. This was helpful, as the distribution of many blood tests was highly skewed, and visual checks showed it performed well in most cases. The second method replaces the values of each variable, such that the number of observations that falls into each quantile bin corresponds to the Gaussian distribution; it is more radical as it only preserves the rank-order of original values. The second method was considered because developers of one of the machine learning models we used (NODE-GAM) found it helpful. The type of transformation was selected during hyperparameter tuning. Data transformation was not applied in the case of decision tree ensembles, and the EBM model, as it was unnecessary.

Machine learning models

A set of machine learning models with varying degrees of flexibility and interpretability were included in the analysis (Table 5.2).

Penalised logistic regression (PLR) with L1 penalty (lasso regression) served as the simplest interpretable baseline model: if this performs as well as more complex models, then the latter are not necessary. PLR makes predictions like ordinary logistic regression: a linear combination of predictor variables plus an intercept is passed through a sigmoid function to yield a prediction between 0 and 1: $p_i = \sigma(w_0 + w_1 x_{1i} + ... + w_k x_{ki})$, where p_i is the predicted probability of cancer for patient i, $x_{1i}...x_{ki}$ are values of k predictor variables for patient i, $w_1...w_k$ are regression coefficients associated with the variables, and w_0 is the intercept. During optimisation, a penalty is applied that encourages the coefficients to be small. PLR was used over regular logistic regression because the penalty protects against overfitting when the number of input variables is large, and L1 penalty was used in particular because it can shrink some coefficients to zero and serves as a variable selection method [160, pg. 219]. An unpenalised logistic model was also used for comparison (denoted as "LR (basic)").

Generalised additive models (GAM) learn an arbitrarily shaped decision function for each variable that describes how the log odds of cancer (in the case of binary outcome) change as the value of the variable increases. Predictions are made by adding the outputs of these decision functions for each variable. In GAMs, the linear predictor of logistic regression is thus replaced with an additive predictor $w_0 + f_1(x_1) + ... + f_p(x_p)$, where $f_1...f_p$ are univariate functions that describe how the predictor changes as the value of the corresponding predictor variable increases [161, pg. 96-97]. GAMs are more flexible than logistic regression (logistic regression is a special case of GAM where each decision function is linear) while still being interpretable because it is possible to visualise the decision function of each variable. For example, GAMs would be able to learn a U-shaped relationship between the value of a blood test and the risk of cancer while ordinary logistic regression cannot. Three implementations of GAMs were considered: sparse neural additive model (SNAM)[162], explainable boosting machine (EBM)[163], and neural oblivious decision tree ensemble GAM (NODE-GAM)[164]. In SNAM, each input variable is passed through its own neural network which allows learning a decision function with arbitrary shape, and a grouped lasso penalty is applied to the networks which can nullify some of them and serves as a variable selection method. SNAM was used because it is implemented in neural network software which allows to conveniently use novel optimisation methods that maximise areas under the curve (see below), allows training these models faster on large datasets using mini-batch gradient descent and graphical processing units, and makes it straightforward to apply the grouped lasso penalty to protect against overfitting. EBM builds on the basic structure of GAMs by allowing some decision functions to act on pairs of variables for learning pairwise interactions. It implements an algorithm that automatically ranks and detects a prespecified number of interactions, and learns the decision functions using decision trees. EBM was employed due to automatic interaction detection and because it has performed well on clinical data [165]. NODE-GAM also uses decision trees, can detect two-way interactions, and has performed equally or better than EBM and other GAMs in tests [164]. NODE-GAM is also implemented in neural

network software, which provides the same benefits as SNAM. Unlike SNAM, it uses a variable selection mechanism that does not require each variable to have its own neural network (smaller computational cost), and it can detect two-way interactions.

Decision tree ensembles make predictions by aggregating the outputs of multiple decision trees, where each tree consists of one or more splits on the predictor variables. Decision tree ensembles can learn more complicated relationships than GAMs but still retain a degree of interpretability because it is possible to compute relevance scores for input variables, such as the average information gain for each variable that results from including that variable in decision trees. Random forest (RF) fits complex decision trees to bootstrap samples of the data using a subset of randomly selected variables and aggregates the predictions of individual trees [166]. Gradient-boosted decision tree (GBDT) learns simple decision trees sequentially, such that the next tree improves on the predictions of the previous tree [167, 168]. Decision tree ensembles were used because they have been known to perform well across diverse classification tasks, and in machine learning competitions [168]. While single decision trees would be more interpretable than decision tree ensembles, these were not employed as they are known to not generalise well [160, pg. 316].

A feedforward artificial neural network, also known as a multilayer perceptron (MLP) consists of multiple layers stacked one after the other, such that the outputs of each layer are the inputs to the next layer [169]. Each layer linearly transforms its input values to a pre-specified number of outputs and applies a non-linear activation function. MLP is potentially the most flexible model because due to the universal approximation theorem of neural networks, it can learn almost any kind of input-output relationship given enough hidden units [169, pg. 194]. However, the contributions of individual variables are harder to interpret. Both flat and deep networks were considered (at most with three hidden layers), and the architecture that performed best on held-out validation data was selected.

Please see Appendix D for more information on the ML models.

Table 5.2: Machine learning models used in the data driven analysis

Model	Interpretability	Flexibility	Interactions
Linear models			
Logistic regression (LR)	Full	Low	None*
Penalised logistic regression (PLR)	Full	Low	None*
Generalised additive models (GAM)			
Sparse neural additive model (SNAM)	Full	Intermediate	None*
Neural oblivious decision tree ensemble GAM (NODE-GAM)	Full	Intermediate	2-way
Explainable boosting machine (EBM)	Full	Intermediate	2-way
Decision tree ensembles			
Random forest (RF)	Intermediate	High	2-way and higher order
Gradient-boosted decision tree (GBDT)	Intermediate	High	2-way and higher order
Artificial neural networks			
Multilayer perceptron (MLP)	Low	High	2-way and higher order

Notes. *Even though interaction terms can be included in logistic regression and neural additive models, this was not pursued in the data driven analysis because it is not obvious which of the numerous possible variable combinations should be added. The other models allow detecting relevant interactions automatically.

Performance metrics

The global performance of models across classification thresholds was assessed using area under the receiver-operating characteristic (ROC) curve and average precision (AP). Area under the ROC curve, while commonly reported, is undesirable for comparing models in highly imbalanced datasets where the number of patients with cancer is much lower than the number of patients without cancer. This is because a model with slightly higher area under the curve could have a much higher positive predictive value at certain classification thresholds, and this important difference would remain undetected [121]. We thus include average precision (AP), which estimates the area under the precision-recall curve [122]. AP is high when both sensitivity is high (most cases of cancer are detected) and positive predictive value is high (when most people who test positive have cancer), a required property for a model that could potentially be used to prioritise patients for hospital investigations. We also report three additional metrics: positive predictive value, negative predictive value, and specificity at 80% sensitivity and at the sensitivity of FIT test at threshold 10 µg/g.

The alignment of predicted and observed probabilities (calibration) was not assessed in this analysis, as the goal was not the development of a prediction model. If any of the models, however, have excellent discrimination, their calibration can be analysed and corrected if needed.

Novel ways of fitting models to data

Prediction models with a binary (yes/no) outcome variable are commonly fitted by minimizing the logistic loss (also called binary cross-entropy, which is equivalent to maximizing the logarithm of predicted probabilities). During each optimisation iteration, an average loss is computed across all samples (patients). However, if the number of patients with cancer is small compared to patients without cancer, the model could achieve a low average loss if it predicts low probabilities of cancer for all patients without discrimination. One potential way to address this problem is to optimise the model directly for area under the ROC curve (AUROC), because high AUROC implies that the predictions for each patient with cancer tend to be higher than predictions for all patients without cancer [170]. AUROC cannot easily be optimised as it is a non-differentiable function of the data, however, novel methods have been published and successfully applied for that purpose in medical image analysis [170, 171]. We specifically use the compositional AUROC maximization method, where a standard logistic loss and a novel AUC loss are iteratively optimised, as it can be more effective than training a model with novel loss function from scratch [171]. In addition to maximizing AUROC, we also optimise the models for area under the precision recall curve (AUPRC)[172]. Please see Appendix D for more information.

Hyperparameter selection

Most machine learning models (except basic logistic regression) have many settings (or hyperparameters) to choose from with no obviously good choice. Hyperparameters include, for example, penalty strength in penalised logistic regression and number of layers in an artificial neural network. Hyperparameters were selected independently from the held-out test set (see 'Evaluating model performance with cross-validation'), first using 20 random draws from the possible values of all parameters, and then making 20 further sequential draws using a Bayesian optimization algorithm with tree-parzen estimators [173]. Hyperparameter values that achieved the best average precision on a separate model validation set were

selected. This process is illustrated for the SNAM model in Figure 5.1. Making random draws from possible parameter combinations does not guarantee that a good combination is found when the number of draws is low; Bayesian optimisation ameliorates this by making intelligent guesses for what parameter combinations to try based on the performance of combinations already evaluated. Asynchronous hyperband scheduler was additionally used to terminate less promising parameter configurations early to save computational time.

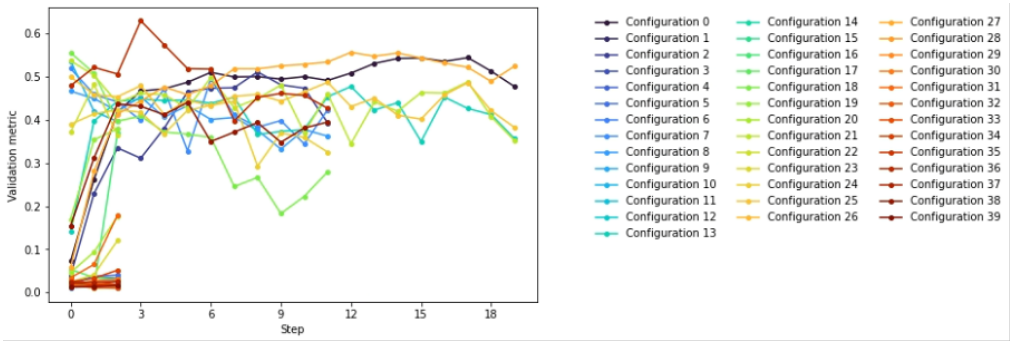


Figure 5.1: Illustration of the hyperparameter selection process for the sparse neural additive model. 40 configurations of hyperparameters were considered; first 20 were random draws from the parameter space, last 20 were sequentially drawn using Bayesian optimization with tree-parzen estimators. Validation metric (average precision score) was evaluated at each optimisation step. Less promising trials were terminated early using the asynchronous hyperband scheduler. Some trials were also terminated early because the validation metric stopped improving. If the later draws tend to have better scores than earlier ones, this indicates that the Bayesian optimization is performing effectively.

Model training

During hyperparameter tuning (see above), artificial neural network models (SNAM, NODE-GAM, MLP) and penalised logistic regression (PLR) were trained using early stopping [174] – when the average precision score in the validation set stopped improving, the training was stopped. Logistic regression models were trained for not more than 50 epochs, other models were trained for at most 150 epochs. Batch size was 128 or 1024 and was selected during tuning. The model chosen in the tuning phase was retrained on the entire training and validation set for the number of epochs selected with early stopping. Exponential learning rate decay was applied,

such that it decayed to 30% of the initial value if the maximum number of training epochs was achieved (50 or 150, depending on the model).

Evaluating model performance with cross-validation

Five-fold cross-validation procedure with a train-validation-test split was used to select hyperparameters for models and to estimate the performance of models (Figure 5.2). First, the data was randomly split into five folds, such that the proportion of cancer cases was similar in each fold. One of the folds was then set aside as the held-out test set, and the other four folds served as the model selection set. The model selection set was further randomly split into training (75%) and validation (25%) sets, again preserving the proportion of cancer cases. Different versions of the model (with different hyperparameters) were fitted to the training set and evaluated on the validation set. The hyperparameter configuration that achieved the best average precision on the validation set was chosen. The model with chosen hyperparameters was then refitted to the entire model selection set and evaluated on the held-out test set. These steps were repeated for five times, until each of the five folds had served as the held-out test set. All data processing parameters (such as parameters of data imputation models) were learned on the model selection set only, to keep the held-out test set independent.

Imputation of missing values

In the main analysis, where we utilised more commonly reported routinely collected data, missing values were imputed using multiple imputation with chained equations (MICE), along with predictive mean matching and a random forest imputation model. Imputation quality was checked by examining the distributions of original and imputed data for each variable. While it is generally recommended to use the outcome variable (presence of cancer) to impute predictor variables, this was not done to make the analysis conservative. If the imputation model is not correct (an assumption that cannot fully be checked), then imputing using the outcome variable can inflate the performance of subsequently fitted models. For example, if the relationship between outcome and predictors is stronger in observed data than

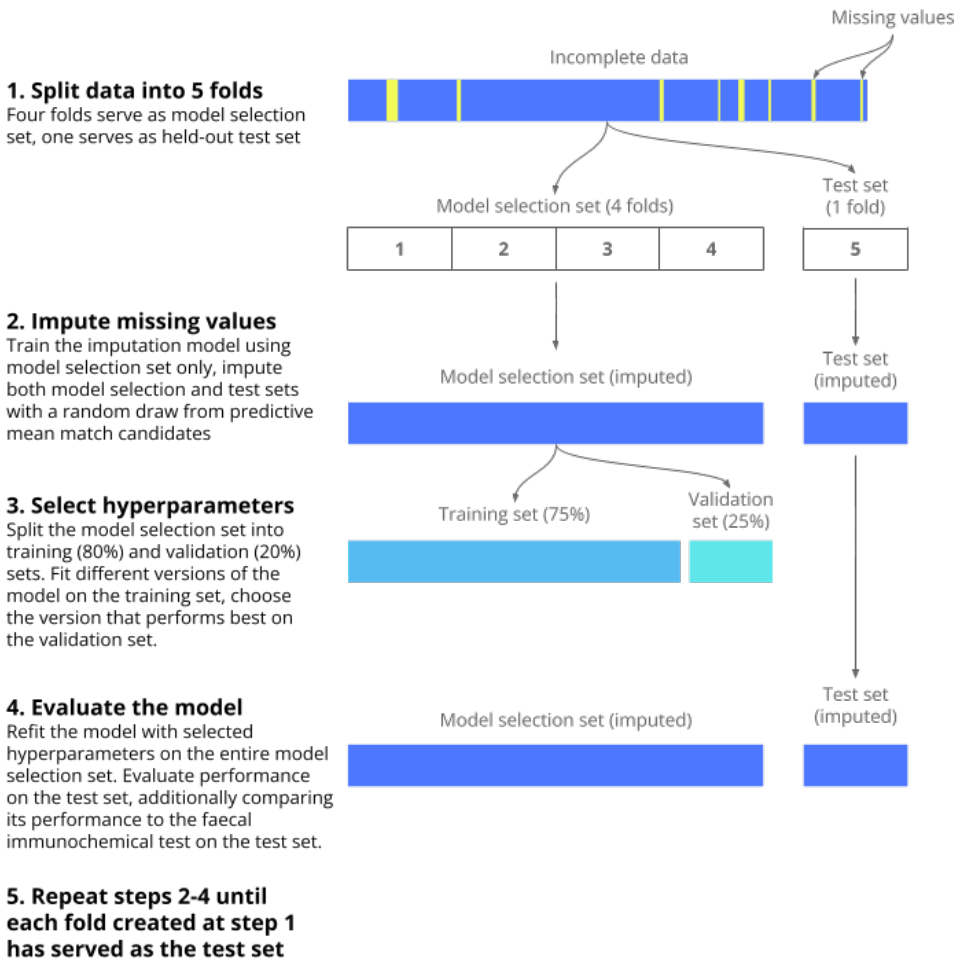


Figure 5.2: Evaluating model performance with five-fold cross-validation and train-validation-test split. The data is first split into five folds; one serves as the held-out test set while the remaining serve as the model selection set. The model selection set is further split into training (75%) and validation (25%) sets, which are used to select the best hyperparameters for a given model. The model with selected parameters is evaluated on the held-out test set. This is repeated five times until each fold has served as the held-out test set. Note that the held-out test set is only used to evaluate the performance of a selected model; it is not used to select hyperparameters for a model or for training the model.

in missing data, and this difference is not accounted for by other observed variables, then imputing using the outcome would inflate subsequent estimates. On the other hand, not using the outcome variable for imputation can deflate the performance of subsequent prediction models, as the relationship between outcome and predictors in observed data is not leveraged for imputation. For a sensitive application such as cancer risk prediction, it seems better to be conservative. However, imputing

data with the outcome variable included was run as a sensitivity analysis. To save computational time, data was imputed once within each cross-validation fold (see below). However, each fold was imputed with a different random draw, which amounts to multiple imputation across folds and allows to incorporate uncertainty in missing values in the results.

In the analysis that incorporated more rarely collected data, multiple imputation was only used for more commonly collected blood tests (available for approximately 20% of patients); other bloods were imputed with median. Median imputation was used so that the imputed values are plausible (many blood test distributions are skewed), and to simplify the analysis. However, uncertainty in missing values is not properly incorporated in this case – if a variable imputed with median is found to be an important predictor of cancer in this dataset with incomplete values, its effect may be different in a new dataset if one decided to measure it for all patients there. Furthermore, imputing with median weakens the correlation structure that may exist between variables. In a later analysis it could be interesting to generate different sub-models for observed and missing values within a same prediction model to better leverage any relationships that may be present in observed data, but this was not pursued.

Software and hardware

Analyses were performed in Python (v3.9). Data was processed with pandas (v1.4.3)[98]. Random forests were fitted with scikit-learn (v1.1.1)[132], gradient-boosted decision trees were run using xgboost (v1.6.1)[168], and the explainable boosting machine was trained with interpret (v0.3.0)[175]. The architecture of the NODE-GAM model was obtained from the nodegam (v0.3.0) package [176]. Penalised logistic regression, SNAM, NODE-GAM, and MLP models were trained with PyTorch (v1.12.1)[100]. Novel loss functions and their optimizers were obtained from libauc (v1.3.0)[177]. Hyperparameter optimisation was achieved with hyperopt (v0.2.7)[173] and the tune library from ray (v2.2.0)[178]. Data was imputed using miceforest (v5.6.1)[126]. Natural language processing for identifying pathology

reports that describe colorectal cancer was performed with custom code based on the regex (v2020.10.15) package. SNAM, NODE-GAM and MLP were trained using an NVIDIA Tesla V-100 GPU with 16GB of RAM, all other models were trained on the CPU.

5.2.4 Reproducibility

Code for the analyses will be made available at `https://github.com/tammandres/fitml`. All sources of randomness were seeded to ensure reproducibility. However, the results for logistic regression models are not completely reproducible as hyperparameter tuning was performed in parallel to speed up computation, although the results were very similar across multiple runs. Clinical datasets used in the analyses cannot be shared but are securely stored in the Oxford University Hospitals (OUH) Trust system.

5.3 Results

5.3.1 Patient cohort and the distribution of predictor variables

There were 51,903 patients with a FIT test result recorded between December 2016 and February 2023, 1,020 had colorectal cancer. After applying the study inclusion criteria, 31,964 individuals were retained, among them 453 cases of colorectal cancer (Figure 5.3). 3,354 (10.5%) of the included patients had a positive FIT test result for their first FIT test, and 387 of the 453 individuals with colorectal cancer were among the FIT positives (FIT test had 85.4% sensitivity for CRC).

Please note that compared to Chapter 4, this analysis uses an updated dataset with roughly 12 months of extra data. The number of included patients is significantly larger than previously (31,964 vs 19,541) because the monthly number of FITs done in Oxford has greatly increased over time. For example, the average monthly number of FITs between March 2017 and February 2022 was 473, and between March 2022 and February 2023 it was 1175. The testing volume has

increased probably because FIT started to be used to also triage higher-risk patients after the first COVID wave [141].

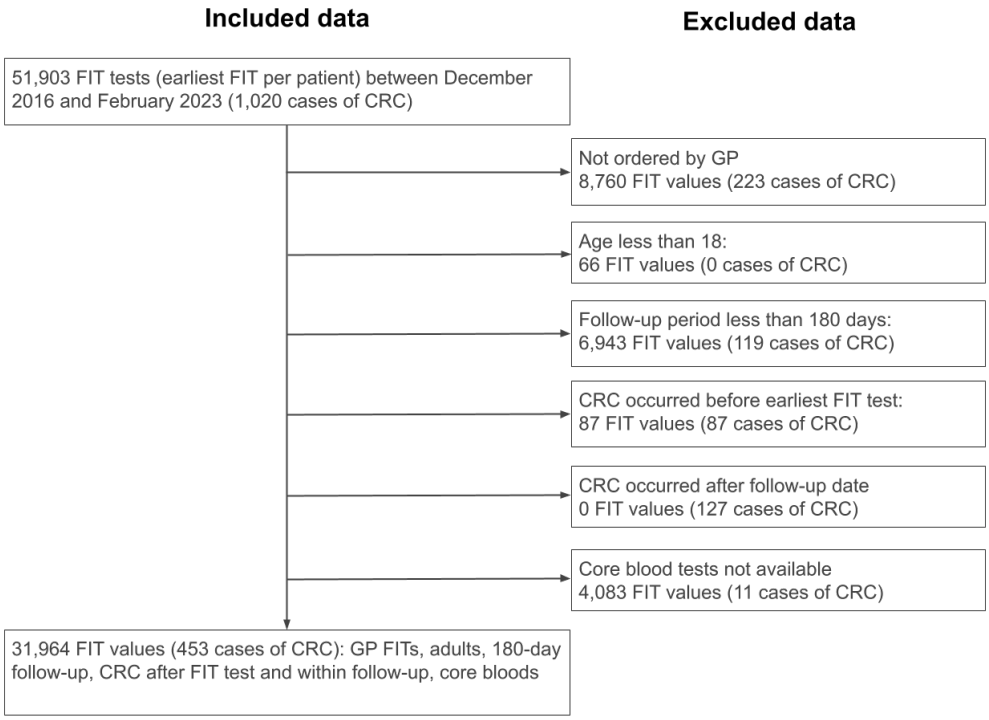


Figure 5.3: Number of patients and colorectal cancer cases at each step of applying the study inclusion criteria. The earliest FIT test result was used for each patient.

Summary statistics for patients with and without colorectal cancer (CRC) are given in Table 5.3. Cancer patients had higher point estimates for median age (73.7 CRC, 63.5 non-CRC), a higher proportion of males (53.6% CRC, 41.6% non-CRC), and higher median FIT test result (27 CRC, 0 non-CRC). The proportion of patients with different GP reported symptoms was similar, except for anaemia (30.9% CRC, 23.9% non-CRC). 51.7% of CRC patients had records for T-stage 3 or 4 within 6-months of the earliest diagnosis date.

Table 5.3: Descriptive statistics for the patient cohort

	No colorectal cancer	Colorectal cancer
Number of patients	31511	453
Age		
18-39.9	2957 (9.4%)	14 (3.1%)
40-49.9	4054 (12.9%)	31 (6.8%)

Continued on next page

Table 5.3 – continued from previous page

	No colorectal cancer	Colorectal cancer
50-59.9	6889 (21.9%)	61 (13.5%)
60-69.9	5533 (17.6%)	79 (17.4%)
70-79.9	6830 (21.7%)	137 (30.2%)
≥80	5248 (16.7%)	131 (28.9%)
Median (25th, 75th)	63.5 (51.3, 76.2)	73.7 (60.4, 81.0)
Min, max	18.1, 102.7	27.0, 102.2
Gender		
F	18406 (58.4%)	210 (46.4%)
M	13105 (41.6%)	243 (53.6%)
Ethnicity		
Asian	739 (2.3%)	2 (0.4%)
Black	229 (0.7%)	1 (0.2%)
Mixed	184 (0.6%)	-
Other Ethnic Groups	291 (0.9%)	4 (0.9%)
White	22900 (72.7%)	327 (72.2%)
Not stated	6146 (19.5%)	105 (23.2%)
Not known	1022 (3.2%)	14 (3.1%)
IMDD		
Median (25th, 75th)	8.0 (7.0, 10.0)	8.0 (7.0, 10.0)
Min, max	1.0, 10.0	1.0, 10.0
Not known	2803 (8.9%)	17 (3.8%)
FIT (µg Hb/g)		
0-1.9	25780 (81.8%)	37 (8.2%)
2-9.9	2764 (8.8%)	29 (6.4%)
10-99.9	2065 (6.6%)	142 (31.3%)
≥8100	902 (2.9%)	245 (54.1%)
Median (25th, 75th)	0.0 (0.0, 0.0)	142.5 (27.0, 400.0)
Min, max	0.0, 400.0	0.0, 400.0
Symptoms - GP reported		
Abdominal mass	30 (0.1%)	2 (0.4%)
Abdominal pain	4475 (14.2%)	53 (11.7%)
Anaemia	7544 (23.9%)	140 (30.9%)
Bloating	1023 (3.2%)	5 (1.1%)
Blood in stool	2831 (9.0%)	73 (16.1%)
Change in bowel habit	11968 (38.0%)	162 (35.8%)
Constipation	1107 (3.5%)	9 (2.0%)
Diarrhoea	3723 (11.8%)	43 (9.5%)
Family history of CRC	281 (0.9%)	2 (0.4%)
Fatigue	457 (1.5%)	2 (0.4%)
Inflammation	388 (1.2%)	4 (0.9%)
Iron deficiency anaemia	2981 (9.5%)	44 (9.7%)
Melaena	355 (1.1%)	1 (0.2%)
Rectal pain	198 (0.6%)	2 (0.4%)
Thrombocytosis	366 (1.2%)	5 (1.1%)
Weight loss	2410 (7.6%)	41 (9.1%)
Not known	4103 (13.0%)	52 (11.5%)
Body mass index		
Median (25th, 75th)	27.4 (24.1, 31.8)	28.4 (25.4, 32.8)
Min, max	11.3, 281.2	19.8, 45.6
Not known	27898 (88.5%)	414 (91.4%)
Number of unique blood test codes		
Median (25th, 75th)	34.0 (30.0, 37.0)	35.0 (31.0, 38.0)

Continued on next page

Table 5.3 – continued from previous page

	No colorectal cancer	Colorectal cancer
Min, max	16.0, 45.0	18.0, 45.0
Number of unique diagnosis codes		
Median (25th, 75th)	6.0 (2.0, 10.0)	4.5 (1.0, 8.0)
Min, max	1.0, 47.0	1.0, 18.0
Number of unique procedure codes		
Median (25th, 75th)	2.0 (1.0, 4.0)	2.0 (1.0, 3.0)
Min, max	1.0, 17.0	1.0, 9.0
Number of unique prescription codes		
Median (25th, 75th)	7.0 (2.0, 15.0)	6.0 (2.0, 13.8)
Min, max	1.0, 57.0	1.0, 33.0
CRC-relevant treatments*		
No treatments recorded	30316 (96.2%)	93 (20.5%)
Chemotherapy	797 (2.5%)	192 (42.4%)
Local excision	31 (0.1%)	11 (2.4%)
Radical resection	146 (0.5%)	264 (58.3%)
Radiotherapy	341 (1.1%)	15 (3.3%)
T-stage (extracted)**		
1	-	45 (9.9%)
2	-	44 (9.7%)
3	-	152 (33.6%)
4	-	82 (18.1%)
Not known	-	130 (28.7%)

Notes. *CRC-relevant treatments are procedures used for treating colorectal cancer (CRC), but they may also be given for other conditions. **T-stage was extracted from radiology and pathology reports using a pattern-matching algorithm.

Based on point estimates for proportions, cancer patients were also more likely to have low haemoglobin, elevated platelets, elevated white cells, low mean cell haemoglobin, low mean cell volume, low ferritin, and high C-reactive protein at or before the first record of cancer (Table 5.4). The distributions of these blood tests, and of age, were also visibly different between cancer and non-cancer patients, although these differences were not large (Figure 5.4). The difference in the distribution of FIT test values was the most marked.

5.3.2 Machine learning models performed similarly to the FIT test in the clinically meaningful range of sensitivities, but outperformed FIT at lower sensitivities

We first report results for machine learning (ML) models that were trained using FIT, age, sex, common blood tests, and clinical symptoms as predictor variables.

Table 5.4: Summary of selected blood tests for the patient cohort

	No colorectal cancer	Colorectal cancer
Haemoglobin		
Median (25th, 75th)	133.0 (121.0, 144.0)	124.0 (108.0, 139.0)
Min, max	34.0, 226.0	53.0, 168.0
Low haemoglobin	10116 (32.1%)	243 (53.6%)
Normal haemoglobin	23114 (73.4%)	223 (49.2%)
Platelets		
Median (25th, 75th)	265.0 (223.0, 316.0)	306.0 (250.0, 370.0)
Min, max	9.0, 1288.0	103.0, 920.0
High platelets	2804 (8.9%)	98 (21.6%)
Normal platelets	29785 (94.5%)	384 (84.8%)
White cells		
Median (25th, 75th)	6.7 (5.6, 8.1)	7.5 (6.2, 9.2)
Min, max	1.3, 237.5	3.6, 20.3
High white cells	2471 (7.8%)	62 (13.7%)
Normal white cells	30416 (96.5%)	421 (92.9%)
Mean cell haemoglobin		
Median (25th, 75th)	30.1 (28.6, 31.3)	28.6 (25.9, 30.3)
Min, max	13.8, 49.4	12.5, 35.2
Low mean cell haemoglobin	4862 (15.4%)	164 (36.2%)
Normal mean cell haemoglobin	27166 (86.2%)	302 (66.7%)
Mean cell volume		
Median (25th, 75th)	91.3 (87.7, 94.7)	88.8 (83.0, 93.0)
Min, max	53.1, 134.7	55.0, 105.0
Low mean cell volume	1990 (6.3%)	89 (19.6%)
Normal mean cell volumne	29946 (95.0%)	379 (83.7%)
Serum ferritin		
Median (25th, 75th)	71.5 (26.3, 150.9)	31.9 (12.6, 105.4)
Min, max	1.0, 6727.8	1.0, 789.6
High serum ferritin	1111 (3.5%)	14 (3.1%)
Low serum ferritin	3669 (11.6%)	109 (24.1%)
Normal serum ferritin	17212 (54.6%)	281 (62.0%)
Not known	14299 (45.4%)	172 (38.0%)
C-reactive protein		
Median (25th, 75th)	2.0 (0.8, 5.3)	5.6 (1.8, 23.7)
Min, max	0.2, 396.8	0.2, 236.4
High C-reactive protein	4075 (12.9%)	128 (28.3%)
Normal C-reactive protein	19727 (62.6%)	208 (45.9%)
Not known	9083 (28.8%)	133 (29.4%)

Notes. Normal, high, and low values for these bloods were defined as in Withrow et al [125]. Low HGB: < 130 g/L for males, < 120 g/L for females. High PLT: $>$ 400 * 109/L. High WBC: > 11 * 109/L. Low MCH: < 27.4 pg/cell. Low MCV: $<$ 80 fl. Low CFER: < 20 µg/L. High CFER: ≥ 350 µg/L. High CRP: > 10 mg/L.

Machine learning models performed well according to metrics that summarise performance over all classification thresholds: all models had higher average precision than the FIT test in all cross-validation folds, and some models also had higher *c*-statistic in all folds (PLR, SNAM, EBM, GBDT) (Figure 5.5, Tables 5.5 and 5.6). NODE-GAM model had the highest gain over FIT test in average precision

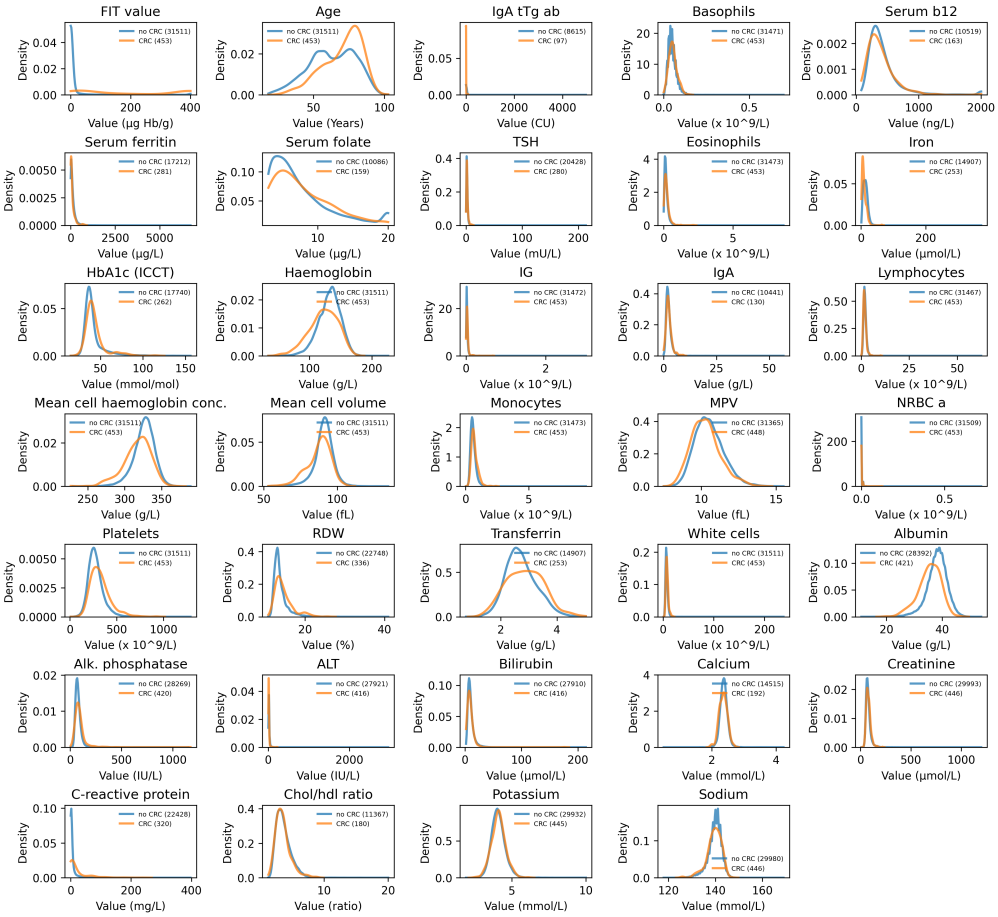


Figure 5.4: Distributions of continuous variables used in the main analysis (FIT test, age, and common blood tests). Figures show the kernel density estimate.

(13.3 points on average), followed by penalised logistic regression (11.9 points on average). However, there was no model that had consistently better positive predictive value than the FIT test at 80% level of sensitivity, or at the level of sensitivity corresponding to FIT at threshold 10. (The sensitivity FIT at threshold 10 was in the range of 83.5-89.0%, depending on the held-out cross-validation fold.)

The same pattern can be seen when examining precision-recall curves, gain in positive predictive value compared to FIT, and reduction in false positive rate compared to FIT at each level of sensitivity (Figure 5.6, panels B-D). All ML models have higher point estimates for PPV than the FIT test for sensitivities approximately less than 60%. At higher sensitivities, positive predictive values of the models were similar to those of the FIT test.

The simple logistic regression model, indicated as 'LR (basic)' in figures and

tables, had lower point estimates of average precision than all other models except the MLP (Figure 5.5), and performed worse than the FIT test for sensitivities higher than approximately 50%. The simple logistic regression model did not utilise any penalty, and only robust scaling was used as a transformation for continuous predictor variables (the variables were centered at median and scaled by interquartile range). The drop in performance was due to no other data transformation being applied to continuous variables (penalised regression with robust scaling had similar drop in performance – data not shown). In other models, the continuous variables were either log-transformed and then robust-scaled or transformed to Gaussian distribution using quantile transformation.

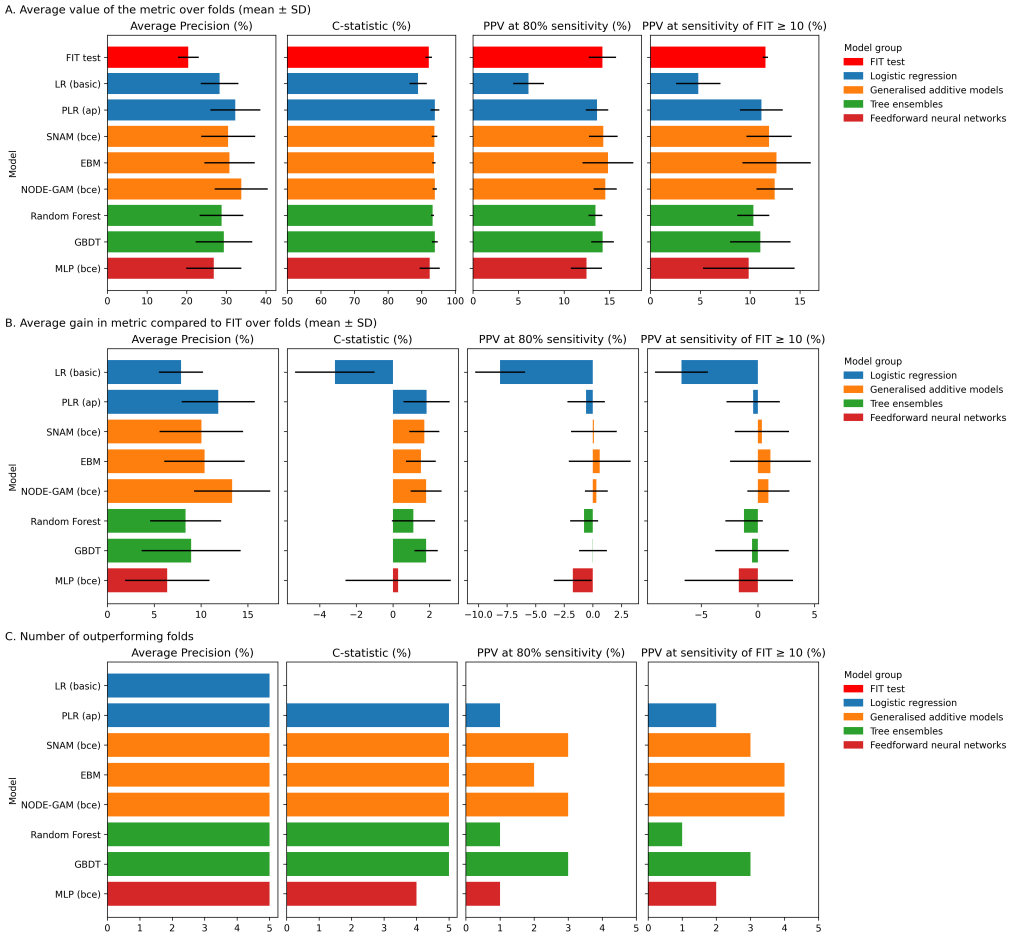


Figure 5.5: Performance of machine learning models over cross-validation folds on the held-out data. A: Mean and standard deviation (std) of each performance metric over the 5 folds. B: Mean and std of the difference in performance of each model and FIT test over the 5 folds. C: Number of cross-validation folds in which the model had higher score than the FIT test. PPV – positive predictive value. Note that the sensitivity of FIT test at threshold 10 varied somewhat between the held-out cross-validation folds (83.5-89.0%).

Table 5.5: Performance of machine learning models over cross-validation folds

Model	Loss function	Metric, mean (std)							
		ap	*c*-statistic	npv-80	ppv-80	spec-80	npv-fit10	ppv-fit10	spec-fit10
Performance on model selection sets (can be optimistic)									
FIT test	-	20.3 (0.7)	92.1 (0.2)	99.7 (0.0)	13.8 (0.3)	92.8 (0.2)	99.8 (0.0)	11.5 (0.1)	90.6 (0.0)
Logistic regression									
LR (basic)	-	30.3 (2.2)	90.5 (0.4)	99.7 (0.0)	7.2 (1.1)	84.8 (2.4)	99.7 (0.0)	5.2 (0.8)	77.2 (3.4)
PLR	ap	36.3 (1.9)	94.0 (0.5)	99.7 (0.0)	14.6 (1.1)	93.2 (0.6)	99.8 (0.0)	11.5 (1.0)	90.5 (1.0)
Generalised additive models									
EBM	-	49.0 (10.5)	96.3 (1.0)	99.7 (0.0)	21.3 (4.1)	95.6 (1.1)	99.8 (0.0)	16.9 (3.4)	93.7 (1.5)
SNAM	bce	36.0 (5.1)	94.5 (0.5)	99.7 (0.0)	16.3 (1.5)	94.0 (0.6)	99.8 (0.0)	12.6 (1.5)	91.4 (1.2)
NODE-GAM	bce	38.7 (2.7)	95.3 (0.2)	99.7 (0.0)	17.1 (0.9)	94.4 (0.3)	99.8 (0.0)	13.3 (0.4)	92.0 (0.3)
Tree ensembles									
Random Forest	-	79.0 (10.2)	99.1 (0.8)	99.7 (0.0)	59.6 (22.8)	99.0 (0.9)	99.8 (0.0)	52.0 (24.4)	98.4 (1.2)
GBDT	-	74.5 (18.3)	99.3 (0.7)	99.7 (0.0)	61.7 (28.9)	98.8 (1.2)	99.8 (0.0)	58.4 (32.1)	98.2 (2.3)
Artificial neural networks									
MLP	bce	52.3 (22.3)	96.9 (2.0)	99.7 (0.0)	32.2 (27.1)	96.0 (2.5)	99.8 (0.0)	27.8 (26.2)	94.6 (3.2)
Performance on held-out test sets									
FIT test	-	20.4 (2.6)	92.1 (1.0)	99.7 (0.0)	14.2 (1.5)	93.0 (0.8)	99.8 (0.0)	11.5 (0.3)	90.6 (0.1)
Logistic regression									
LR (basic)	-	28.3 (4.8)	89.0 (2.6)	99.6 (0.0)	6.1 (1.7)	81.3 (4.5)	99.7 (0.0)	4.8 (2.2)	71.3 (13.4)
PLR	ap	32.3 (6.3)	94.0 (1.3)	99.7 (0.0)	13.6 (1.2)	92.6 (0.8)	99.8 (0.0)	11.1 (2.1)	89.8 (2.8)
Generalised additive models									
EBM	-	30.8 (6.3)	93.6 (0.5)	99.7 (0.0)	14.8 (2.8)	93.2 (1.2)	99.8 (0.0)	12.6 (3.4)	90.8 (3.4)
SNAM	bce	30.5 (6.8)	93.8 (0.9)	99.7 (0.0)	14.3 (1.6)	93.0 (0.9)	99.8 (0.0)	11.9 (2.3)	90.6 (2.4)
NODE-GAM	bce	33.7 (6.7)	93.9 (0.6)	99.7 (0.0)	14.5 (1.3)	93.2 (0.7)	99.8 (0.0)	12.5 (1.8)	91.2 (1.7)
Tree ensembles									
Random Forest	-	28.8 (5.5)	93.2 (0.4)	99.7 (0.0)	13.5 (0.8)	92.6 (0.5)	99.8 (0.0)	10.3 (1.6)	89.0 (2.4)
GBDT	-	29.4 (7.1)	93.9 (0.8)	99.7 (0.0)	14.2 (1.2)	93.0 (0.7)	99.8 (0.0)	11.0 (3.0)	89.1 (4.7)
Artificial neural networks									
MLP	bce	26.8 (6.9)	92.4 (3.0)	99.7 (0.0)	12.5 (1.7)	91.8 (1.1)	99.8 (0.0)	9.9 (4.6)	84.1 (13.7)

Notes. **Loss functions**: bce – binary cross-entropy, also known as logistic loss; ap – average precision loss. **Metrics**: ap - average precision, *c*-statistic - area under ROC curve; npv-80, ppv-80, spec-80 - negative predictive value (NPV), positive predictive value (PPV) and specificity at 80% sensitivity; npv-fit10, ppv-fit10, spec-fit10 - NPV, PPV and specificity at the sensitivity of FIT $\geq$10 µg/g. **Data splits**: Data was split into the model selection set and test sets, and the model selection set was further split into training and validation sets – see Figure 5.2. This table shows the best model from each class of models (e.g. the PLR model was trained with multiple loss functions, and the version with ap loss had highest validation set score.

Table 5.6: Performance of machine learning models relative to FIT test over cross-validation folds

Model	Loss function	Metric, mean (std)					
		ap	*c*-statistic	ppv-80	spec-80	ppv-fit10	spec-fit10
		Gain on model selection sets (can be optimistic)					
Logistic regression							
LR (basic)	-	10.1 (1.5)	-1.6 (0.3)	-6.6 (0.9)	-8.0 (2.3)	-6.3 (0.8)	-13.4 (3.4)
PLR	ap	16.1 (1.4)	1.9 (0.4)	0.7 (0.9)	0.4 (0.5)	-0.0 (1.0)	-0.1 (0.9)
Generalised additive models							
EBM	-	28.7 (10.1)	4.2 (0.9)	7.5 (4.0)	2.7 (1.1)	5.4 (3.5)	3.2 (1.4)
SNAM	bce	15.8 (4.5)	2.4 (0.4)	2.4 (1.3)	1.2 (0.6)	1.1 (1.6)	0.8 (1.1)
NODE-GAM	bce	18.4 (2.2)	3.2 (0.2)	3.3 (1.0)	1.6 (0.4)	1.8 (0.4)	1.4 (0.3)
Tree ensembles							
Random Forest	-	58.7 (10.4)	7.0 (0.7)	45.8 (22.9)	6.2 (1.0)	40.4 (24.3)	7.8 (1.2)
GBDT	-	54.3 (18.6)	7.2 (0.9)	47.9 (28.9)	6.0 (1.2)	46.8 (32.2)	7.6 (2.3)
Artificial neural networks							
MLP	bce	32.0 (21.7)	4.7 (2.0)	18.4 (26.9)	3.2 (2.4)	16.2 (26.2)	4.0 (3.1)
		Gain on held-out test sets					
Logistic regression							
LR (basic)	-	7.9 (2.4)	-3.2 (2.2)	-8.1 (2.2)	-11.7 (4.5)	-6.7 (2.3)	-19.3 (13.3)
PLR	ap	11.8 (3.9)	1.8 (1.3)	-0.6 (1.6)	-0.3 (1.0)	-0.4 (2.4)	-0.8 (2.8)
Generalised additive models							
EBM	-	10.4 (4.3)	1.5 (0.8)	0.6 (2.7)	0.2 (1.1)	1.1 (3.6)	0.2 (3.3)
SNAM	bce	10.0 (4.5)	1.7 (0.8)	0.1 (2.0)	0.0 (1.1)	0.4 (2.4)	-0.0 (2.3)
NODE-GAM	bce	13.3 (4.1)	1.8 (0.8)	0.3 (1.0)	0.2 (0.5)	0.9 (1.9)	0.6 (1.6)
Tree ensembles							
Random Forest	-	8.4 (3.8)	1.1 (1.2)	-0.8 (1.2)	-0.4 (0.7)	-1.2 (1.7)	-1.5 (2.3)
GBDT	-	9.0 (5.3)	1.8 (0.6)	0.0 (1.2)	0.0 (0.7)	-0.5 (3.2)	-1.5 (4.6)
Artificial neural networks							
MLP	bce	6.4 (4.5)	0.3 (2.9)	-1.8 (1.7)	-1.2 (1.0)	-1.7 (4.8)	-6.5 (13.6)

Notes. **Loss functions**: bce – binary cross-entropy, also known as logistic loss; ap – average precision loss. **Metrics**: ap - average precision, *c*-statistic - area under ROC curve, ppv-80 - positive predictive value (PPV) at 80% sensitivity, spec-80 - specificity at 80% sensitivity; ppv-fit10, spec-fit10 - PPV and specificity at the sensitivity of FIT $\geq$10 µg/g. **Data splits**: Data was split into the model selection set and test sets, and the model selection set was further split into training and validation sets – see Figure 5.2. This table shows the best model from each class of models (e.g. the PLR model was trained with multiple loss functions, and the version with ap loss had highest validation set score.

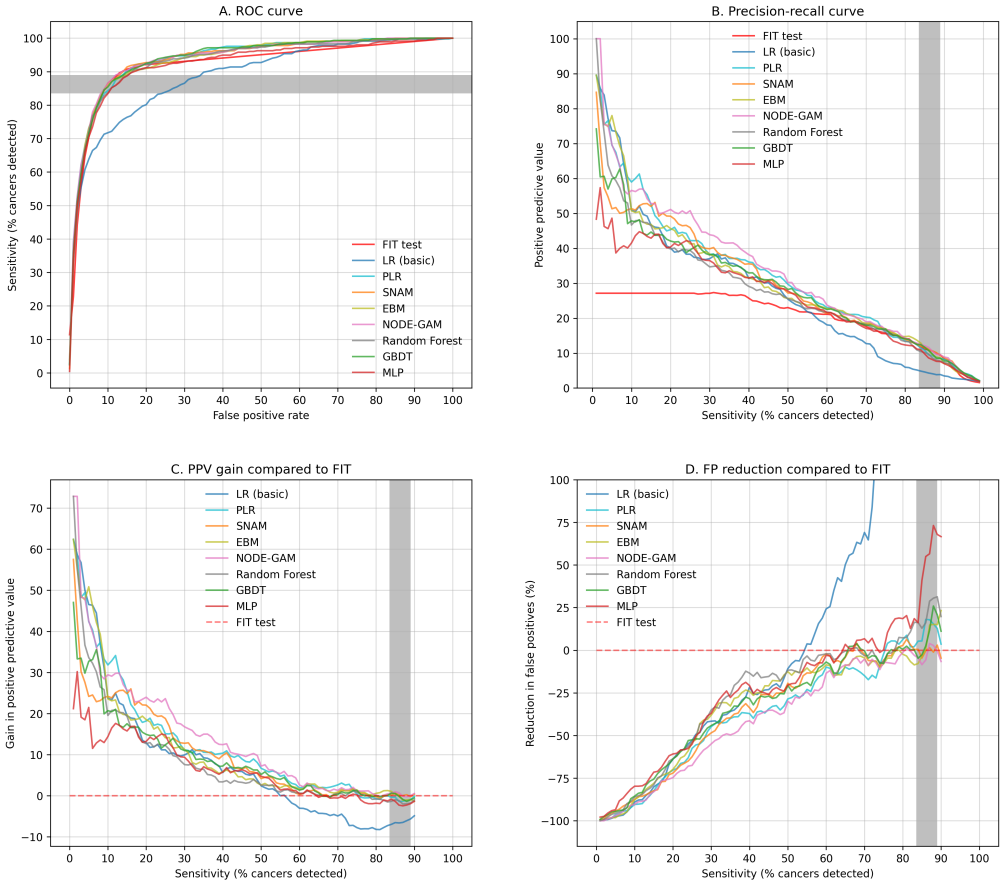


Figure 5.6: Performance of machine learning models and the faecal immunochemical test (FIT) for detection of colorectal cancer. Panels display the ROC-curve (A), precision-recall curve (B), gain in positive predictive value compared to the FIT test (C), and percent reduction in false positives compared to FIT (D). Curves show the mean value of each quantity over 5 cross-validation folds. The vertical grey area is shaded between the minimum and maximum sensitivities of FIT test greater than or equal to 10 µg/g on the held-out folds. All curves were interpolated to a fixed grid of values at 1-unit increments. Data for sensitivities greater than 90% is not shown for panels C and D because the interpolation for FIT test is inaccurate in that range. LR – basic logistic regression with linear data transformation, PLR – penalised logistic regression with log or quantile data transformation, SNAM – sparse neural additive model, EBM - explainable boosting machine, NODE-GAM - neural oblivious decision tree ensemble GAM, GBDT – gradient-boosted decision tree, MLP - multilayer perceptron.

5.3.3 Including additional prediction variables, such as rare blood tests and clinical codes, did not improve the performance of machine learning models

To explore the effect of including more predictor variables, we trained each type of machine learning model (except EBM) using three different sets of variables:

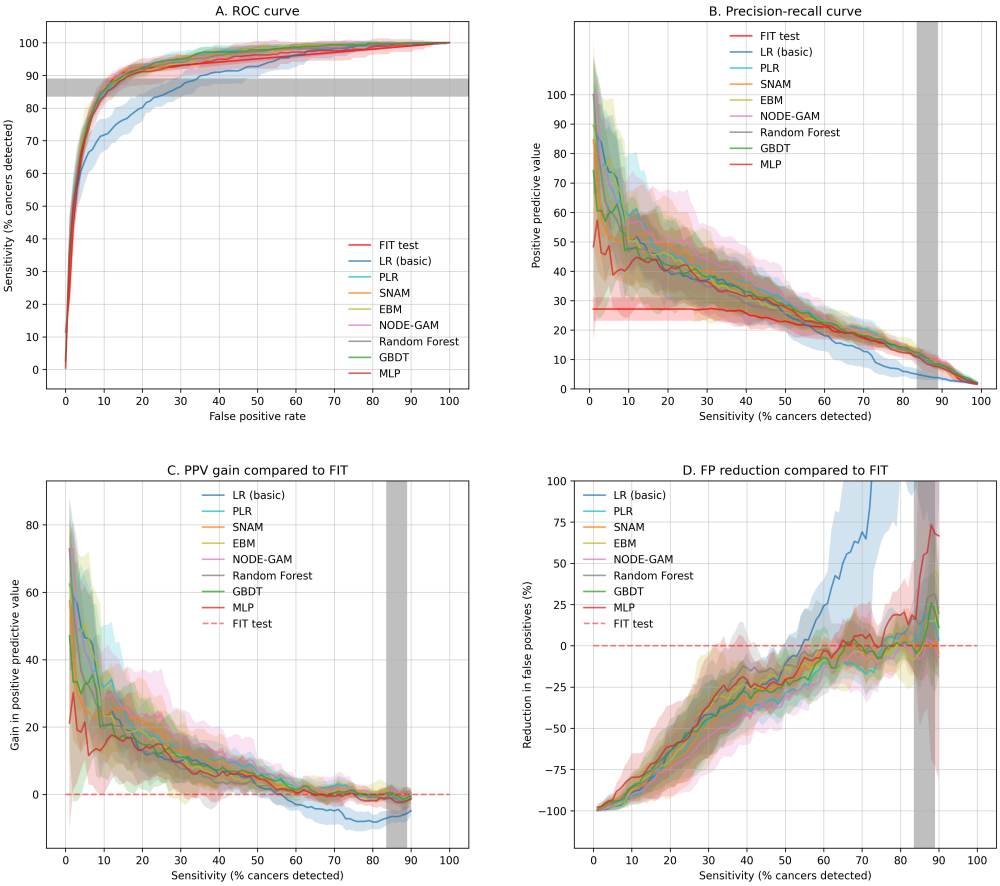


Figure 5.7: Performance of machine learning models and the faecal immunochemical test (FIT) for detection of colorectal cancer, showing the variability between cross-validation folds. For each quantity, the shaded area shows the mean plus-minus standard deviation over five cross-validation folds. Please see Figure 5.6 for more information.

(1) FIT, age, and sex (FAST); (2) FIT, age, sex, common blood test results, and clinical symptoms ("common" set of variables); (3) FAST, common bloods, clinical symptoms, rare bloods, diagnosis codes, procedure codes, prescription codes, ethnicity, and body-mass index ("extended" set of variables).

The precision-recall curves show that at high levels of sensitivity ($> 80\%$), the positive predictive values (PPV) of all models and the FIT test were similar, regardless of whether age, sex, blood tests and other predictor variables were included (Figure 5.8). At lower levels of sensitivity, the FIT-age-sex model also had a similar PPV to the FIT test, except below the 30% sensitivity threshold where it showed gain, possibly because the FIT test had achieved its maximum threshold of 400 µg Hb/g and could not provide additional information to discriminate

cancers. The models that included common and extended sets of predictors had higher point estimates of PPV than both the FIT test and the FAST model at sensitivities approximately less than 60% but had similar estimates of PPV to each other. Note that the Explainable Boosting Machine (EBM) model was not included in this analysis as tuning and training it on the extended set of predictor variables was time consuming, although other generalised additive models (SNAM and NODE-GAM) were included.

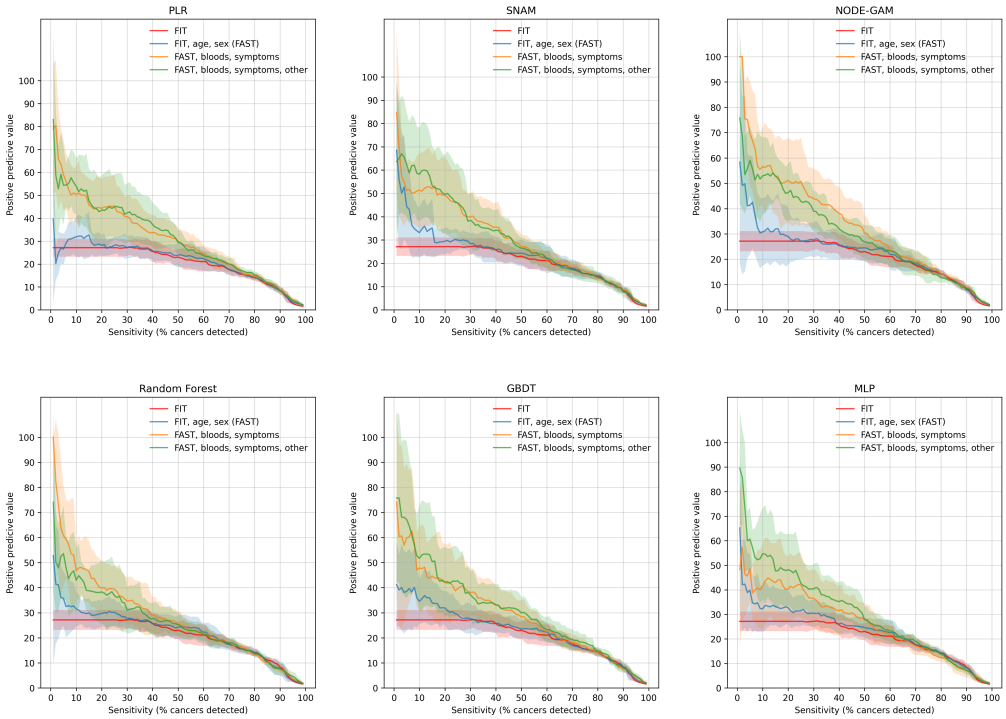


Figure 5.8: Precision-recall curves for each machine learning model trained with three sets of predictor variables. The three sets of variables were (1) FIT, age, and sex (FAST), (2) FIT, age, sex, common bloods, and clinical symptoms; and (3) the same as (2) plus rare bloods, diagnosis/procedure/prescription codes, ethnicity, and BMI. Each machine learning model is shown in a different pane: PLR – penalised logistic regression, SNAM – sparse neural additive model, NODE-GAM – neural oblivious decision tree ensemble GAM, random forest, GBDT – gradient boosted decision tree, MLP – multilayer perceptron (a feedforward neural network). The curves display the mean of positive predictive value (PPV) over 5 cross-validation folds, shaded areas give mean plus-minus standard deviation.

5.3.4 The variables most predictive of colorectal cancer were the FIT test, age, sex, certain bloods, and clinical symptoms

We also examined two of the highest performing models (NODE-GAM and penalised logistic regression) to discover variables that were most predictive of colorectal cancer. This analysis should be considered as indicative of the effects of each variable, but not statistically robust, because computing confidence intervals for the effect of each variable over cross-validation folds is more difficult due to non-independence of the folds and was not the aim of the study.

Relevant variables in the logistic model

For penalised logistic regression (PLR), we inspected the regression coefficients over cross-validation folds. The mean and standard deviation for each coefficient over the folds is shown in Figure 5.9 (exponentiated to aid interpretation), and the coefficients within each fold are displayed in Figure 5.10. FIT test, sex, iron deficiency anaemia as a GP-recorded symptom, change in bowel habit, age, abdominal pain, melaena, diarrhoea, mean cell volume, platelets, and weight loss were on average associated with at least a 20% change in the odds of cancer. Some other blood tests and symptoms were associated with at least a 10% change, including mean cell haemoglobin concentration, creatinine, albumin, ferritin, serum B12 and immunoglobulin G (Figure 5.9). In the logistic model, all continuous variables had been log-transformed, then centered at median and rescaled using interquartile range, so the coefficients indicate the effects of increasing the value of each continuous variable by interquartile range on the log-scale.

However, it is important to note that the regression coefficient of each predictor variable in the PLR model represents the partial effect of the variable when controlling for other variables in the model, which can make the interpretation of these effects more difficult. For example, the haemoglobin blood test had a positive regression coefficient in all cross-validation folds, implying that increased haemoglobin levels were associated with increased risk of colorectal cancer. This

seems unexpected, as low rather than high levels of haemoglobin are a potential symptom of CRC. However, the haemoglobin levels were correlated with several other bloods, including having positive associations with mean cell haemoglobin concentration (MCHC) and iron levels. Both MCHC and iron had negative regression coefficients in all cross-validation folds, which were on average stronger in magnitude than for HGB. Therefore, a decrease in haemoglobin levels would have been associated with a decrease in both MCHC and iron, and the overall effect would have still been increased risk of cancer, as expected.

We also made the penalised regression model sparser and thus more interpretable by manually increasing penalty strength while accepting at most a 2% absolute reduction in performance on the validation set based on the average precision metric (e.g. if AP of the original PLR model was 32%, then in the sparse model it was at least 30%). In the sparse model, only the FIT test, age, iron levels, mean cell volume, platelets, and mean haemoglobin concentration remained relevant (being on average associated with at least a 10% change in odds of cancer, and the standard-deviation error bar not intersecting with zero). The sparse model performed similarly to the original model on the test set but had some loss in performance especially at lower levels of sensitivity (Figure 5.11).

Important variables in the generalised additive model (NODE-GAM)

In the NODE-GAM model, the top 12 most important variables were the FIT test, age, mean cell volume, mean cell haemoglobin concentration, albumin, platelets, gender, alanine transaminase (ALT), C-reactive protein, iron deficiency anaemia (symptom), change in bowel habit (symptom) and serum B12 (Figure 9), based on the average variable relevance score over the cross-validation folds. The effects of these variables varied somewhat between the folds, but generally had a similar shape, except for the ALT blood test where the effect did not seem consistent (Figure 5.12). The effects were in the expected direction for most variables: FIT value, age, platelets, male gender, C-reactive protein and change in bowel habit were positively associated with the risk of cancer, whereas the increase in mean cell

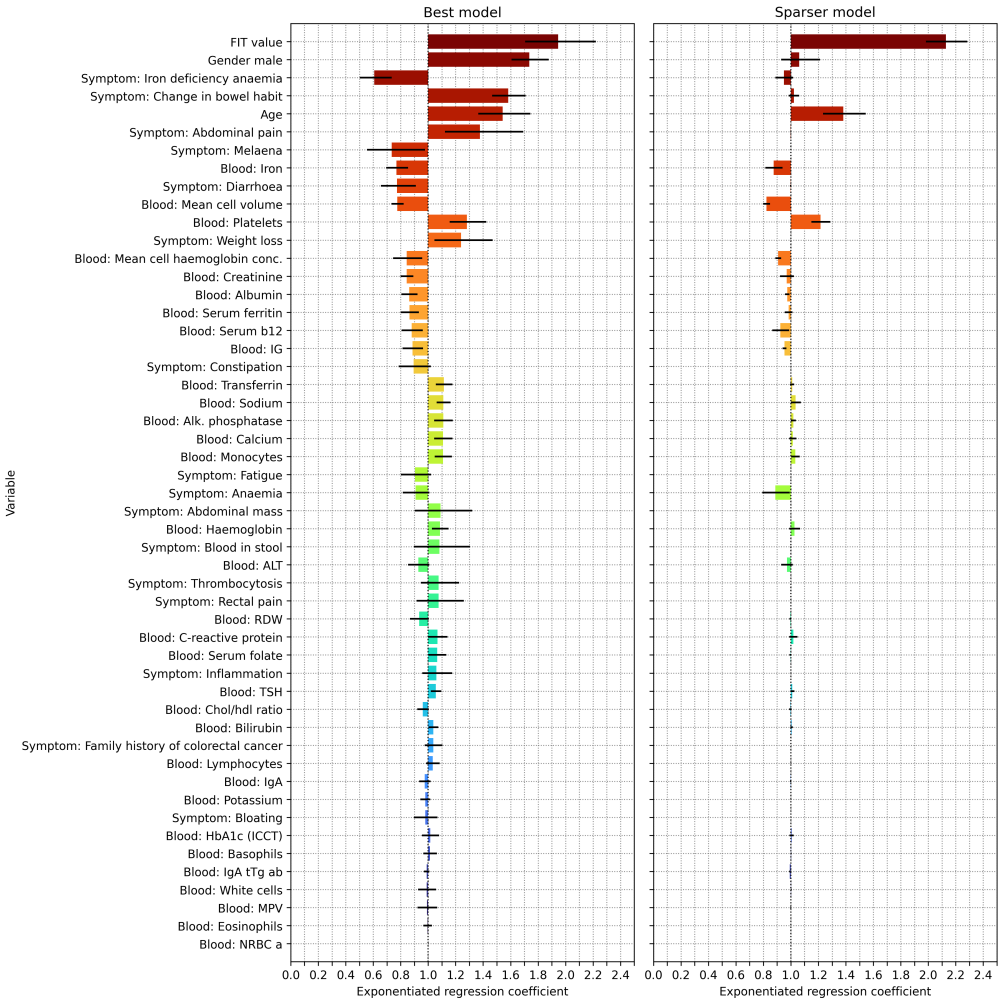


Figure 5.9: Summary of regression coefficients of the penalised logistic regression (PLR) model over cross-validation folds. Mean and standard deviation over the folds was computed for the coefficient of each variable, then exponentiated to aid interpretation. Left: The full model is the PLR model where the penalty parameter was chosen using hyperparameter tuning with Bayesian optimisation. Right: In the sparse model, the penalty of the full model was manually increased so that there was no more than a 2% absolute decrease in the average precision metric. The increase in penalty however led to a more interpretable model, as more coefficients were zero (or close to zero).

volume and mean cell haemoglobin concentration was associated with decreased risk. The increase in albumin and B12, and the GP-recorded symptom of iron deficiency anaemia were associated with decreased risk. However, these effects are again partial (effects of each variable controlling for the levels of other variables) and should be interpreted with care. For example, the presence of the iron deficiency anaemia symptom was associated with decreased cancer risk. However, when the symptom was recorded, patients often had low haemoglobin (and low MCV

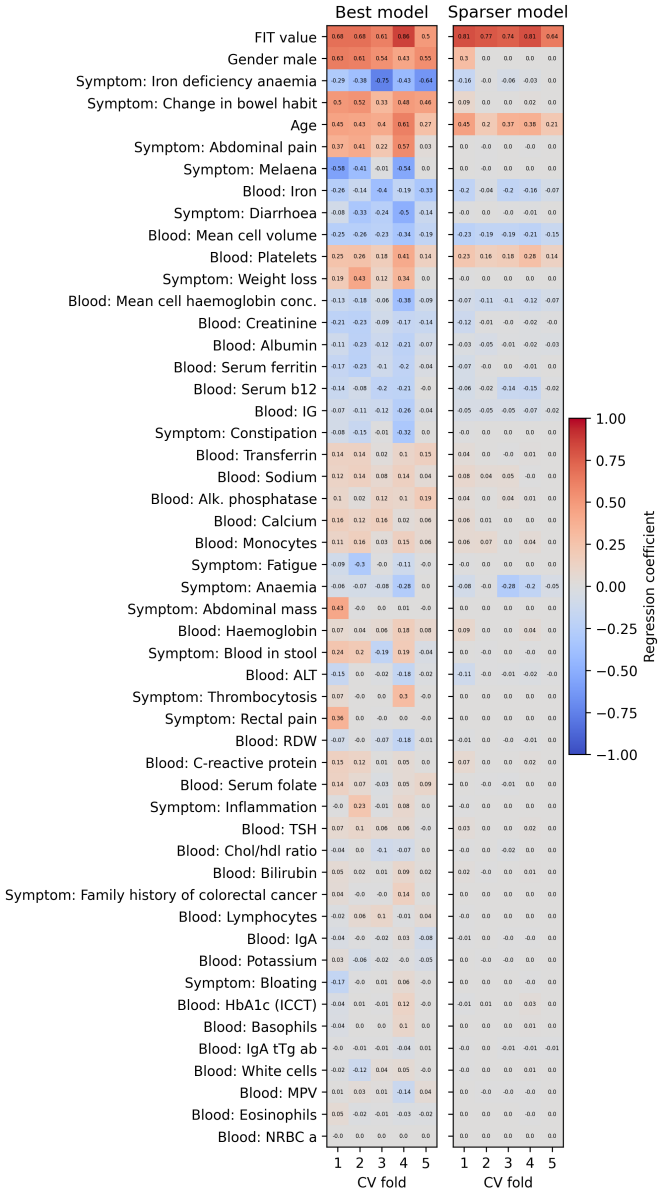


Figure 5.10: Regression coefficients of the penalised logistic regression (PLR) model in each cross-validation fold. Left: The full model is the PLR model where the penalty parameter was chosen using hyperparameter tuning with Bayesian optimisation. Right: In the sparse model, the penalty of the full model was manually increased so that there was no more than a 2% absolute decrease in the average precision metric.

and MCHC), so the overall effect could have still been increased risk of cancer. The NODE-GAM model exhibited some but not strong overfitting to the training dataset: its mean average precision over-cross validation folds was 38.7 on the model selection set (training data), and 33.7 on the held-out test set, so it is plausible that the learned relationships are close to the real underlying relationships and

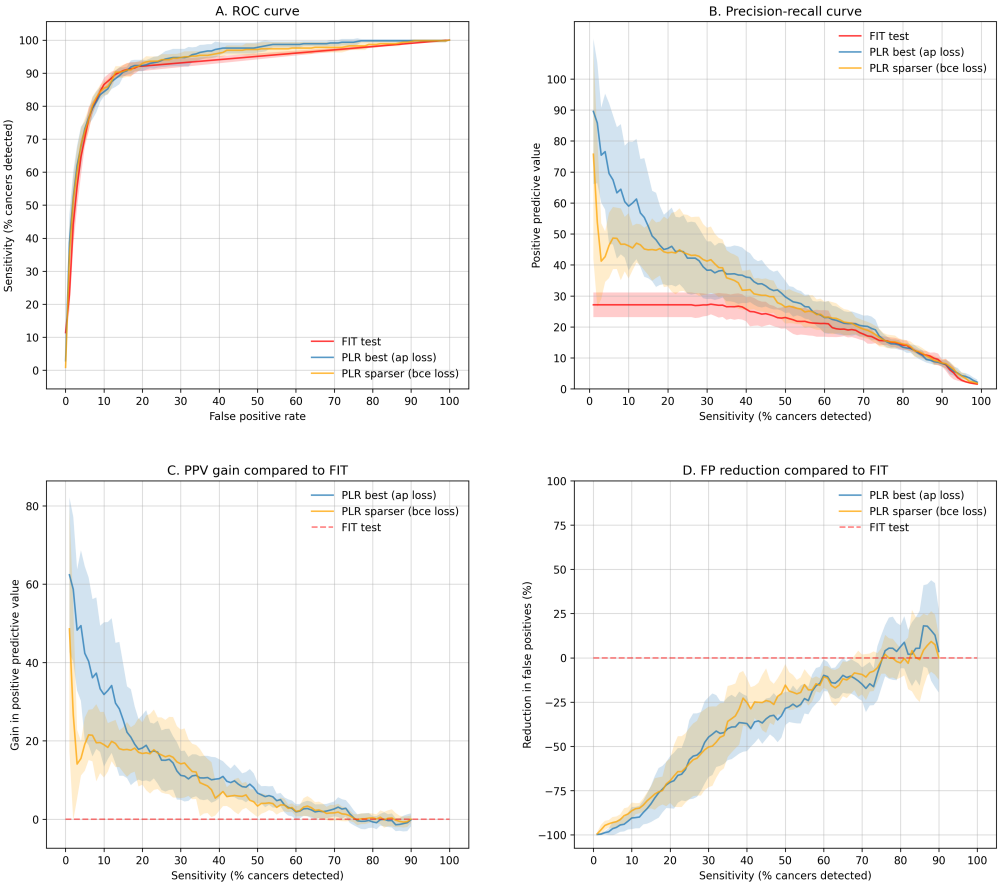


Figure 5.11: Performance of the penalised logistic regression model compared to a sparser version of the model. Panels display the receiver-operating characteristic (ROC) curve (A), the precision-recall curve (B), gain in positive predictive value for each model compared to the FIT test (C), and percent reduction in false positives for each model compared to the FIT test (D). Curves show the mean value of each quantity over 5 cross-validation folds. For curves C and D, data is not shown for sensitivities higher than 90%, because it was not possible to accurately interpolate the positive predictive value and false positive rate for the FIT test within this range of values.

do not simply represent the model fitting to noise.

Contributions of variables were not further discussed with clinical experts because the models did not outperform FIT alone. Even if the contributions were statistically significant, they were not practically significant. It was nevertheless reassuring that some effects were consistent with the literature, such as CRC being associated with a higher platelet count and lower mean cell volume [119, 154].

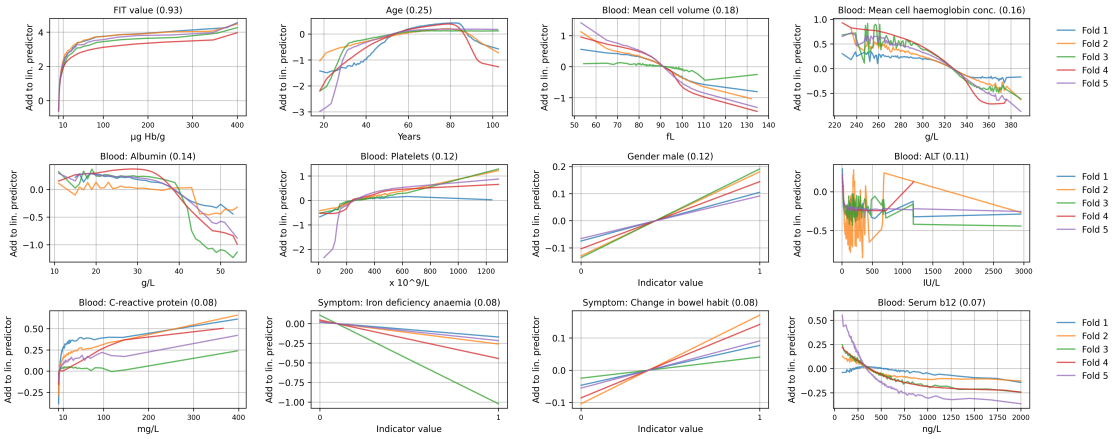


Figure 5.12: Effects of the most relevant variables selected by the NODE-GAM model. The model makes predictions by adding up the effects of each variable (which can be read from the graphs), adding an intercept, and applying the sigmoid transformation.

5.3.5 Optimising models for high area under the curve did not generally lead to better performing models

We fitted some machine learning models to data using three different methods: (1) minimizing the binary cross-entropy loss function, which is the standard method of training models ("bce" in figures and tables); (2) minimizing the average precision loss which optimises models for high area under the precision-recall curve ("ap" in figures and tables); and (3) minimizing the compositional AUC margin loss which maximises area under the ROC curve ("caucm" in figures and tables). This was done using the libauc python library and was therefore only applicable to models implemented in pytorch: penalised logistic regression (PLR), sparse neural additive model (SNAM), NODE-GAM, and the multilayer perceptron (MLP). All metrics reported for the losses below refer to performance on held-out cross-validation folds that were not used for training the models. Results are summarised in Figure 5.13 and Table 5.7.

For the PLR model, employing the ap loss led on average to a 1.4% improvement in average precision over the bce loss, with improvement seen in 4 out of 5 folds. For MLP, the ap loss was associated with a 4.9% improvement over bce loss, outperforming the bce loss in all folds, although the MLP model still performed worse than the PLR model. AP loss did not lead to improved performance in other models.

The compositional AUC loss improved the *c*-statistic of the MLP model on average by 0.97% over the bce loss, and the average precision score on average by 2.3% over the bce loss, with improvement seen in 4 out of 5 folds. The compositional AUC loss was not associated with improved performance in other models.

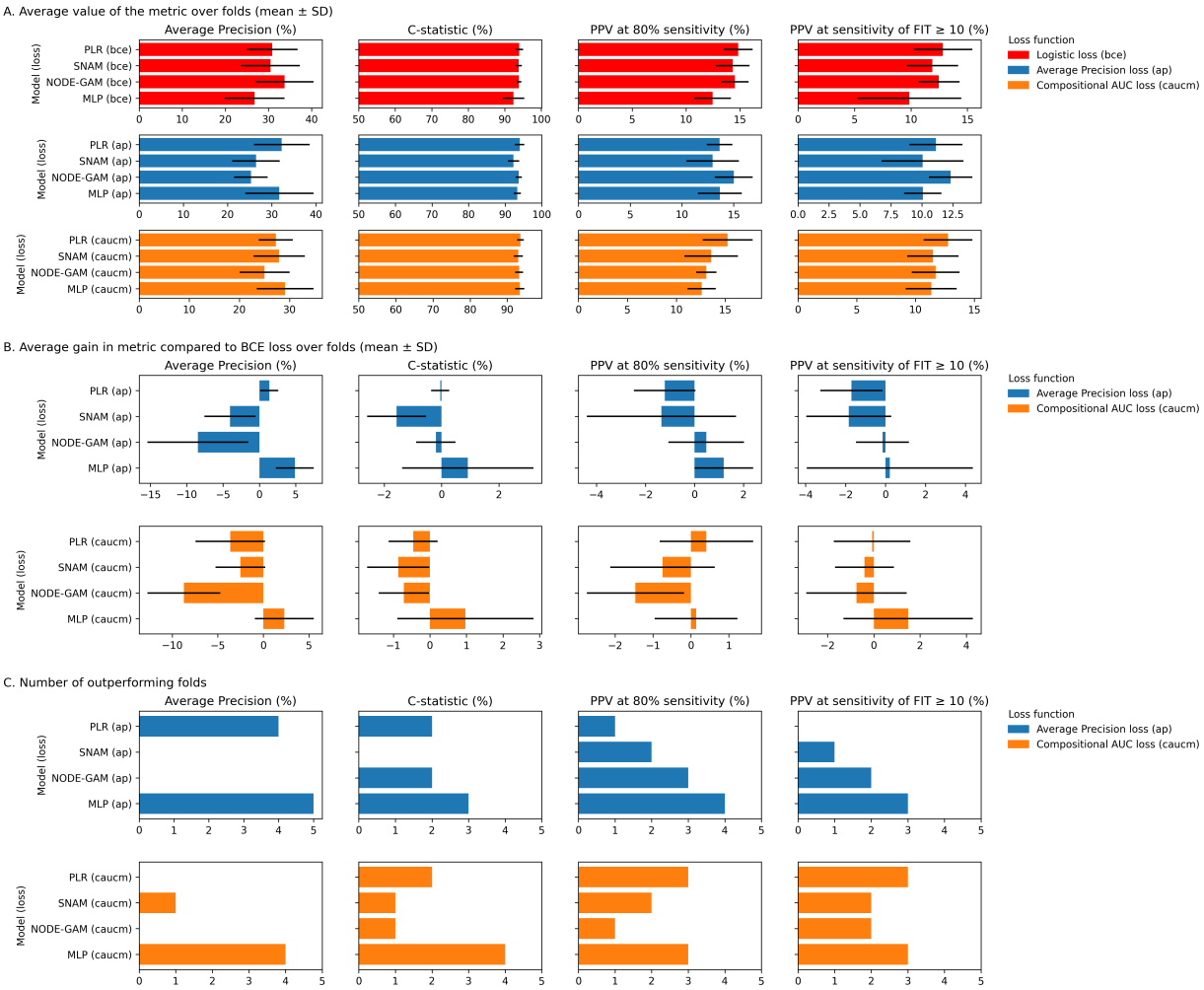


Figure 5.13: Performance of machine learning models trained with different loss functions over the cross-validation folds. A: Mean and standard deviation (std) for each model-loss combination over the 5 held-out cross-validation folds. B: Mean and std of the difference in performance of each model trained with average precision (ap) loss and compositional AUC loss (caucm), when compared to the same model trained with the standard binary cross-entropy (bce) loss, over the 5 folds. C: Number of cross-validation folds in which the model trained with an alternative loss (ap, caucm) performed better than the same type of model trained with the standard loss (bce). PPV – positive predictive value.

Table 5.7: Performance of machine learning models trained with different loss functions

Model	Metric, mean (std)							
	ap	*c*-statistic	npv-80	ppv-80	spec-80	npv-fit10	ppv-fit10	spec-fit10
Performance on model selection sets (can be optimistic)								
bce loss								
PLR	30.9 (5.8)	94.0 (1.0)	99.7 (0.0)	14.8 (1.3)	93.3 (0.7)	99.8 (0.0)	12.8 (2.6)	91.3 (2.4)
SNAM	30.5 (6.8)	93.8 (0.9)	99.7 (0.0)	14.3 (1.6)	93.0 (0.9)	99.8 (0.0)	11.9 (2.3)	90.6 (2.4)
NODE-GAM	33.7 (6.7)	93.9 (0.6)	99.7 (0.0)	14.5 (1.3)	93.2 (0.7)	99.8 (0.0)	12.5 (1.8)	91.2 (1.7)
MLP	26.8 (6.9)	92.4 (3.0)	99.7 (0.0)	12.5 (1.7)	91.8 (1.1)	99.8 (0.0)	9.9 (4.6)	84.1 (13.7)
ap loss								
PLR	32.3 (6.3)	94.0 (1.3)	99.7 (0.0)	13.6 (1.2)	92.6 (0.8)	99.8 (0.0)	11.1 (2.1)	89.8 (2.8)
SNAM	26.4 (5.4)	92.3 (1.6)	99.7 (0.0)	13.0 (2.5)	92.0 (1.9)	99.8 (0.0)	10.1 (3.3)	86.9 (8.1)
NODE-GAM	25.3 (3.8)	93.7 (0.9)	99.7 (0.0)	15.0 (1.8)	93.4 (0.9)	99.8 (0.0)	12.3 (1.8)	91.1 (1.8)
MLP	31.7 (7.7)	93.3 (1.0)	99.7 (0.0)	13.7 (2.1)	92.5 (1.5)	99.8 (0.0)	10.1 (1.5)	88.8 (2.0)
caucm loss								
PLR	27.3 (3.4)	93.5 (0.9)	99.7 (0.0)	15.3 (2.6)	93.4 (1.2)	99.8 (0.0)	12.8 (2.1)	91.3 (2.2)
SNAM	27.9 (5.1)	93.0 (1.2)	99.7 (0.0)	13.6 (2.7)	92.3 (2.2)	99.8 (0.0)	11.5 (2.2)	90.2 (2.2)
NODE-GAM	25.0 (5.0)	93.2 (1.1)	99.7 (0.0)	13.1 (1.0)	92.3 (0.8)	99.8 (0.0)	11.7 (2.0)	90.4 (2.3)
MLP	29.1 (5.7)	93.4 (1.2)	99.7 (0.0)	12.6 (1.4)	91.9 (1.2)	99.8 (0.0)	11.3 (2.2)	90.0 (2.4)
Performance on held-out test sets								
bce loss								
PLR	34.4 (4.3)	94.5 (0.5)	99.7 (0.0)	15.8 (0.7)	93.9 (0.4)	99.8 (0.0)	12.6 (0.6)	91.5 (0.5)
SNAM	36.0 (5.1)	94.5 (0.5)	99.7 (0.0)	16.3 (1.5)	94.0 (0.6)	99.8 (0.0)	12.6 (1.5)	91.4 (1.2)
NODE-GAM	38.7 (2.7)	95.3 (0.2)	99.7 (0.0)	17.1 (0.9)	94.4 (0.3)	99.8 (0.0)	13.3 (0.4)	92.0 (0.3)
MLP	52.3 (22.3)	96.9 (2.0)	99.7 (0.0)	32.2 (27.1)	96.0 (2.5)	99.8 (0.0)	27.8 (26.2)	94.6 (3.2)
ap loss								
PLR	36.3 (1.9)	94.0 (0.5)	99.7 (0.0)	14.6 (1.1)	93.2 (0.6)	99.8 (0.0)	11.5 (1.0)	90.5 (1.0)
SNAM	30.5 (7.4)	92.6 (1.5)	99.7 (0.0)	14.7 (2.8)	93.1 (1.5)	99.8 (0.0)	11.7 (2.3)	90.5 (1.9)
NODE-GAM	29.3 (6.0)	93.7 (0.5)	99.7 (0.0)	15.3 (1.1)	93.6 (0.5)	99.8 (0.0)	12.7 (0.7)	91.5 (0.6)
MLP	43.6 (4.7)	95.2 (1.6)	99.7 (0.0)	22.7 (7.7)	95.7 (1.4)	99.8 (0.0)	19.5 (6.6)	94.5 (1.7)
caucm loss								
PLR	29.2 (5.7)	93.8 (1.4)	99.7 (0.0)	15.0 (1.6)	93.4 (0.8)	99.8 (0.0)	12.4 (1.2)	91.3 (0.9)
SNAM	29.4 (6.5)	93.5 (1.4)	99.7 (0.0)	14.4 (1.8)	93.0 (1.0)	99.8 (0.0)	11.8 (1.9)	90.6 (1.8)
NODE-GAM	26.5 (3.1)	93.2 (0.6)	99.7 (0.0)	12.8 (2.0)	91.9 (1.5)	99.8 (0.0)	11.1 (1.6)	90.0 (1.7)
MLP	32.3 (4.6)	94.6 (1.9)	99.7 (0.0)	16.7 (3.3)	94.0 (1.4)	99.8 (0.0)	13.7 (2.7)	92.0 (1.7)

Notes. **Loss functions**: bce – binary cross-entropy, also known as logistic loss; ap – average precision loss. **Metrics**: ap - average precision, *c*-statistic - area under ROC curve; npv-80, ppv-80, spec-80 - negative predictive value (NPV), positive predictive value (PPV) and specificity at 80% sensitivity; npv-fit10, ppv-fit10, spec-fit10 - NPV, PPV and specificity at the sensitivity of FIT $\geq$10 µg/g. **Data splits**: Data was split into the model selection set and test sets, and the model selection set was further split into training and validation sets – see Figure 5.2.

5.3.6 Sensitivity analyses

We also ran several sensitivity analyses to check the robustness of the results reported in the main analysis (Appendix C): we included outcome variable in the missing data imputation models (C.1), explored another imputation method that is implicitly included in the GBDT prediction model (C.2), increased the length of follow-up to 365 days (C.3), ran the analysis with a different random seed (C.4), used models pretrained with the regular loss function before applying the novel loss functions (C.5), and checked the performance on a newer subset of the data where most faecal samples were processed differently using a sample picker (C.6). Results were similar to the main analysis.

5.4 Discussion

5.4.1 Main findings

It is hard to outperform the FIT test: we did not find a way to significantly reduce the number of false positives while capturing as many cancers as FIT test alone would. This is despite using a relatively large dataset (31,964 patients), considering both simple and flexible machine learning models (logistic regression, generalised additive models, decision tree ensembles, feedforward neural networks), including a variety of predictor variables (demographics, common blood tests, rare blood tests, clinical codes), and optimising the models to have high area under the ROC and precision-recall curves.

The prediction models did outperform FIT at lower levels of sensitivity where about less than 60% of all cancers would be detected. In this lower sensitivity region, only patients with FIT values well above the clinically recommended threshold of 10 µg Hb/g test positive, so a significant number of patients with a negative FIT result will have cancer. It is possible that in this case, the additional predictor variables included in models are useful, because they help decide which of the FIT-negative patients do have cancer.

Most of the predictor variables we considered were not helpful for predicting the risk of cancer when included in a model alongside the FIT test, at least at the current sample size of the study. This was true for all rare blood tests, clinical codes, ethnicity, body mass index, and for many of the common bloods and clinical symptoms. On the other hand, a smaller number of predictor variables—including age, mean cell volume, platelets, and some others—were probably predictive of colorectal cancer, but they were still not useful enough when the FIT test result was also available.

5.4.2 Sample size did not prohibit the use of machine learning models, but their full potential was unlikely to be realised

Some of the most flexible machine learning models used in this analysis—artificial feedforward neural networks—can learn almost any kind of (continuous) relationship between input and output variables, including linear, U-shaped, 'wiggly' etc [179]. The decision tree ensembles are also very flexible, as they can learn non-linear relationships and interactions. However, the ability to learn flexible relationships is potentially useful only if the true input-output relationship is non-linear, and if there is enough data to accurately capture it; even linear relationships can be hard to detect when the effect size is small.

Most blood tests included in the 'extended set' of variables had about 400-3500 observed values (>90% values missing, Appendix C.1), and the number of patients with records for the various diagnosis/procedure/prescription codes was less than 500 for approximately 75% of codes. As the rate of colorectal cancer is small (1.42% in this study), the number of cancer cases with observed values for each of these variables was even smaller. If these variables have a weak relationship with the risk of cancer, which they presumably do, the number of values is probably too small to extract useful information from them. However, we believe it was still good to include these variables in the analysis to exclude the possibility of overlooking any strong effects.

In the main analysis, most imputed blood tests had at least 14,000 observations (which included at least 192 cases of cancer, Appendix C.1)), which is intuitively sufficient to detect some relationships between these bloods and cancer. However, depending on effect size, some of these relationships can still be too weak at the current sample size, and the sample size may also be insufficient for detecting two-way or higher order interactions.

However, as the codebase for this analysis is established, it can be used to run the same type of modelling quite fast on a much larger regional FIT dataset, which may become acessible to Oxford researchers in the future.

5.4.3 Routinely collected data *vs* cancer-specific biological assays

In our main analysis, we combined FIT test results with routinely collected data available in electronic patient records (age, sex, common blood tests, and clinical symptoms). If a model using routine data would outperform the FIT test alone, it could be readily evaluated in prospective studies and clinical trials (or even deployed), because it is easy to obtain the information required for applying the model. A potential drawback, however, is that routinely collected data is not specific for detecting colorectal cancer, so there is likely an upper limit of performance that can be achieved by combining FIT results with routine data, and in this analysis no significant performance gain was observed. It could be worthwhile to explore whether more specific biological assays for colorectal cancer could significantly improve the performance of the FIT test (for example, by helping decide which of the FIT positive patients have cancer, to boost its positive predictive value). These could include tests that look for cancer specific methylation patterns in cell-free DNA [180], such as the Multi-Cancer Early Detection (MCED) test [181].

5.4.4 Model evaluation with cross-validation was computationally efficient, but it was not clear how to obtain statistical confidence intervals

A drawback of cross-validation for evaluating model performance is that there is no straightforward way of obtaining statistical confidence intervals. Reporting the standard error of performance estimates across folds may not be valid, because performance estimates at each fold are not independent [182] (hence a standard deviation rather than standard error was reported). However, it is still possible to sense how robust the results are to different ways of randomly splitting and imputing the data: each cross-validation fold (when used as the held-out test set) represents a different split of the data and is associated with a different random draw from the data imputation model. The uncertainty due to data splits and imputation is therefore incorporated in the results, although not as rigorously as in a statistical confidence interval. Repeated cross-validation may have been even better for incorporating uncertainty due to data splitting, but we did not pursue this to save computational time. A sensitivity analysis, where we ran cross-validation again with a different random seed, did not yield substantially different results (Appendix C.4), and each held-out fold had a substantial number of samples (approximately 6392 patients and 90 cases of CRC).

Machine learning models could have also been evaluated using bootstrap. The bootstrap bias-correction method, which is recommended for clinical prediction models, involves first fitting the model on the original data, and then correcting its performance estimate for optimism. Optimistic bias is estimated by fitting a new model on a bootstrap sample of the original data, computing a difference between its performance on the bootstrap sample and the original data, repeating this process B times, and taking the average of these differences. Computational cost aside, this method is unlikely to be suitable for machine learning models because training and test data are not separate. Consider an extreme case where a model completely overfits the training data and does not learn any generalisable patterns. In this case, the model makes a perfect prediction for all patients in the training

data but performs randomly on the data of new patients it has not seen. After bias correction, the performance estimate of this model should be random, but the bootstrap procedure suggests a better-than random (optimistic) estimate. This is because about 63% of patients in the original data are in each bootstrap sample, and thus the difference in performance between each bootstrap sample (perfect) and original data (not perfect but better than random) is not large enough to capture the full optimistic bias. It should be possible to show that whenever there is a large enough degree of overfitting, the estimated bias is less than the true bias. This is partly also because having repeated observations in each bootstrap sample does not sufficiently increase the performance of the model on that bootstrap sample, because a highly overfitting model performs well for almost any observation it encounters.

Other bootstrap methods (including out-of-bag bootstrap, the .632 and .632+ estimators) involve training the model on bootstrap samples of the original data and evaluating it on the observations that were left out of the bootstrap samples, with additional bias correction applied in the .632 methods. While these have a proper train-test split, the out-of-bag bootstrap is biased because only about 63% of data is used for training the models. The bias correction in the .632 methods involves computing a weighted average of performances on the training and test data, which rests on assumptions that may need to be checked. In addition, bootstrap approaches are computationally much more expensive than cross-validation, requiring each model to be fitted at least about 100 times. In practice models need to be fitted many more times than that, unless simplifying assumptions are made [183], because usually multiple versions of each machine learning model need to be trained on each data split to choose the best combination of hyperparameters. This would have been considerably more expensive in our analysis, as with 5-fold cross-validation and 40 hyperparameter tuning trials per model, it took about 48 hours to run. Like cross-validation, bootstrap methods do not provide a straightforward way of getting statistical confidence intervals due to non-independence of performance estimates across samples.

5.4.5 The general additive models differed in their scalability, and lacked control over their smoothness

We employed three types of generalised additive models (GAMs) in this analysis: the explainable boosting machine (EBM)[163], the sparse neural additive model (SNAM)[162], and the neural oblivious decision tree ensemble GAM (NODE-GAM)[164]. To double check that the models were indeed able to learn 'feature functions' that describe the effect of each variable—functions with potentially different shapes—we trained these models on a dummy binary classification dataset that was simulated from a generalised additive model with known feature functions (linear, U-shaped, sinusoidal, piecewise linear). The models were indeed able to recover shapes of the input feature functions (data available upon request). This quality check was important especially for the SNAM and NODE-GAM models, because models based on neural networks can 'fail silently'[184]. Below we list a few interesting observations made when applying the models; these statements are not precise as the purpose of this analysis was not a detailed evaluation of GAM architectures.

GAMs differed in their scalability. In the extended analysis, where 582 predictor variables were used, the hyperparameter tuning and cross-validation process for the EBM model used more than approximately 24 hours of computing time and thus EBM was excluded from that analysis; this is possibly because feature functions that describe the effect of each variable are learned by iterating through all variables a large number of times [163]. The SNAM and NODE-GAM models could be trained faster in the 582-predictor case, partly because they utilised parallel computation via the GPU. The NODE-GAM also benefitted from automatic feature selection, as it did not require that a separate neural network be specified for each variable as in SNAM. There are also other GAM architectures that can accommodate an even larger number of input variables with a lower computational cost [185, 186].

A limitation of all GAM architectures we employed is that it is not possible to control the smoothness of the feature functions that are learned. If the dataset is not large enough, the model can overfit the data, and potentially learn functions

that at least partially represent noise and not the true effects of each variable (for example, it is plausible that some sharp fluctuations and jumps in the functions learned by the NODE-GAM model do not represent real effects; Figure 5.12). It could be desirable to control the smoothness of these functions, so that with a smaller dataset more of the feature functions are smoother and some even close to linear. As the number of predictor variables increases, it may not be feasible to have a separate smoothing parameter for each variable; however, if the learned feature functions were a linear combinations of a smaller set of basis functions as proposed by Radenovic et al [186], it would perhaps be possible to control the smoothness of these bases to obtain differential control over the smoothness of the learned functions (so that some are more smooth than others).

5.4.6 Novel loss functions did not lead to clearly better performance

We used novel techniques for optimising our prediction models for high area under the ROC curve (and high area under the precision-recall curve), with the hope that this could lead to better performance in the context of an imbalanced dataset. Even though we observed a small gain in performance for the logistic regression model, we did not observe a general improvement among the different model classes we tested. Perhaps the performance gain offered by these techniques was too small to be reliably detected (the publication for one of the methods, the maximisation of area under PR curve, reports relatively small gains [172]). On the other hand, it could be more promising to explore novel methods that maximize *partial* area under the ROC and PR curves, as this is arguably of more clinical interest than the entire area.

5.4.7 The future of colorectal cancer risk prediction models

We attempted to externally validate new Nottingham colorectal cancer risk prediction models in the Oxford dataset (Chapter 4), reviewed several other published models, and employed a variety of machine learning techniques to build a model.

In the light of these findings, what could the future of CRC risk prediction in symptomatic patients look like? Some possibilities are as follows.

FIT test combined with a few routinely collected variables outperforms FIT test alone. This would require the routinely collected variables to be strong enough predictors of cancer risk, so that they would lead to a significantly smaller number of false positives than FIT alone, which is probably not true in general because routinely collected data is not specific to cancer. However, this could still occur when patients with a positive FIT test result in the absence of CRC have other diseases which can be readily distinguished from CRC on the basis of common bloods. Perhaps this is why the Nottingham team was able to report a gain in performance over FIT alone, while their model and other locally derived models did not outperform FIT in the Oxford dataset. The performance of such prediction models may thus be strongly dependent on the prevalence of other colorectal diseases, in which case these prediction models would be useful only in some areas of the UK. This possibility should be explored further by analysing the Oxford and Nottingham datasets in more detail to clarify why a prediction model works in one but not in the other.

FIT combined with many routinely collected variables outperfoms FIT alone. Even if routinely collected variables are weakly predictive of CRC, combining many weak (independent) predictors can lead to good predictions overall. To further test this possibility, a dataset that is many times larger than the current one is needed, so that there is enough power to detect weak effects and interactions between variables. Such a model may not be as clinically useful, because it can require too many variables to be observed, could be less trusted by clinicians, and it can be harder to understand when and why it works.

FIT combined with a few cancer-specific tests outperforms FIT alone. In this case, patients who test positive for FIT could additionally be offered biological assays that have good diagnostic performance for the detection of colorectal cancer, such as the Multi-Cancer Early Detection test that had 70% sensitivity and 94% PPV for CRC in patients referred to urgent investigation for possible lower gastrointestinal cancer (see pg. 16 in the supplementary appendix of Nicholson et al [181]) . This

is more promising than combining FIT with routinely collected data, as it could lead to prediction models that are valid over a wider population of symptomatic patients and more reliable. However, the applicability of these models would also require the assays to be easily available and cost-effective.

6

A bird's eye view on patterns of care: clustering patient event logs using event-pair distances

Contents

6.1 Introduction

6.1.1 Motivation

Clinical researchers who are determined to harness the potential of electronic health records (EHR) are often faced with large multi-dimensional datasets, where the data of each patient is spread over multiple tables. For example, the NIHR HIC colorectal cancer (CRC) database contains data for 12,903 patients [4] in 30 tables, and the freely-available MIMIC-III critical care database encompasses 38,596 patients and 26 tables [187]. Although it is straightforward to summarise the patient cohort one data item at a time, it is more difficult to understand how the data of each patient 'looks like' as a whole. After all, the data of each patient consists of a sequence of events in time, and the pattern of these events is potentially more informative than the presence or absence of individual events. As described in Chapter 2, the data of each patient can be visualised using timeline plots, where each patient is represented by a horizontal line, and each event of interest is marked on that line with a different symbol-color combination (see Figure 2.3). While this allows to examine the patterns of events for a small number of patients, *is it possible to automatically quantify the similarity between medical event sequences, so that patients with similar patterns of events can be grouped together and visualized together*?

This chapters introduces a fast sequence characterisation and clustering pipeline, together with an interactive app, that allows clinical researchers to quickly review thousands of event sequences, in order to discover patterns that may exist in the data (such as diagnosis, treatment, and surveillance patterns). The aim of the pipeline is to provide a novel clinical data visualisation tool that could help a researcher better understand their data, both its quality and whether the data may contain temporal

patterns of events that are of interest to their research question (the discovered patterns could then be studied further with more formal methods). It builds on an existing sequence embedding method (the sequence graph transform, or SGT [188]), so that it incorporates information about event times. This is because the timing of events is important, and could sometimes be crucial, for determining whether two sequences of medical events are similar. SGT was chosen, as it is fast, allows to capture both short-term and long-term dependencies between events in the sequence, and—as stated previously—can be modified to include information about event times. I first briefly review other existing methods, and then illustrate the pipeline on a multi-centre dataset of colorectal cancer patients [4], exploring treatment patterns for patients with rectal cancer.

6.1.2 Methods of characterising medical event sequences

Let the data of each patient consist of a sequence of events (such as ['scan', 'diagnosis', 'scan', 'radiotherapy']) and associated relative event times (such as [-10 days, - days, 15 days, 64 days])[1]; in this example, the relative event times are given with respect to a reference event ('diagnosis'). The aim is to discover patients with similar event traces, but it is not possible to directly compute the similarity between any two traces because initially there are no common variables that describe them. One way to compare traces is to compute a predefined set of features, for example "Feature 1: Does the trace contain major surgery?", "Feature 2: Does the trace contain major surgery followed by chemotherapy within 180 days?" etc. Such manual 'feature engineering' would be appropriate for answering a specific research question, but it could be effortful to apply in general. We therefore consider methods of automatically characterising event traces, that associate each trace with a fixed-size numerical vector (where each element is a feature that characterises the traces), also known in the machine learning literature as an 'embedding'. Traces can then be compared by computing a similarity score between their embedding

[1]For simplicity, we assume that the events of interest are already defined at the desired level of granularity, as each medical event usually belongs to a hierarchy of concepts (e.g. as in the SNOMED CT medical vocabulary [189]). It is also possible to include multiple versions of the same event in a sequence, each expressed at a different level of hierarchy [190].

vectors, and the dimension of the embeddings can also be reduced to visualise and explore the sequences in two-dimensional space. We also briefly discuss alternative methods that attempt to more directly quantify a similarity between two sequences using edit-distance. However, having an 'embedding' or a 'feature vector' for each sequence is potentially more desirable, as it is then possible to display the sequences on a scatter plot based on their similarity, which can facilitate the exploration and understanding of data (see below).

Note that if one only considers the events and ignores event times, then each event sequence can be thought of as a sequence of words, and methods commonly used in natural language processing can be applied to extract features from that sequence, provided that they internally map the input sequence elements to an embedding (for example, feedforward neural networks, recurrent neural networks, and transformers trained to predict the next element in a sequence [191]). However, we are especially interested in methods that also incorporate event times (or can be modified to do so), because time is important in clinical context. For example, the sequence ['surgery', 'radiotherapy'] has a very different meaning if radiotherapy occurs close to surgery (in which case it is probably a planned follow-up treatment), or a year after surgery (in which it case it may be an initially unplanned treatment of a subsequent recurrence of cancer). Diagnosing and treating patients within a specific timeframe is also part of the NHS cancer standards [192], so methods that are used to cluster and detect treatment patterns should account for time to at least potentially distinguish treatment patterns that comply with standards from those that do not.

Below, I review some of the strategies of characterising and comparing medical event sequences.

Indicators and event counts

Possibly the simplest way of automatically extracting features from a sequence of events is to create indicator variables for the presence of each event (1: present, 0: absent). For example, the set of all possible events is {a, b, c}, then a sequence [a, b, b, a, b] would be converted to a vector [1, 1, 0]. A disadvantage is that this

ignores information about event order. To capture information about event order, indicators could be created for all consecutive event pairs, triples, etc. For example, when the sequences consist of at most three events from the set {a, b, c}, then there are nine possible pairs of consecutive events: (a, a), (a, b), ..., (c, b), (c, c). Such subsequences of two events are also called bigrams, and subsequences of n events n-grams [193]. Indicators for single events and bigrams were some of the methods used by Song et al in an earlier paper on clustering event logs [194].

Instead of creating event indicators, one could count the occurrence of each event in a sequence (known as a 'bag of words' representation in NLP), or to normalise the counts for the total number of events in the sequence and down-weigh events occurring in most sequences (the 'term frequency inverse document frequency' or TF-IDF method in NLP). Both methods, in conjuction with dimension reduction, were used as baselines for representing medical event sequences by Landi et al [195].

Event indicators, bigrams, event counts and TF-IDF do not contain information about the relative timing between events. Although bigrams could be modified such that they take a value 1 only if the timing between the two events is less than a certain threshold, such as '6 months'.

Sequence embeddings from artificial neural networks

Event traces can be processed by artificial neural networks that extract features from the sequence, which can then be used for comparing the sequences. There are various approaches that differ on (1) how the input sequence is represented before it is fed into the neural network, (2) the neural network architecture that processes the sequence representation, and (3) the task the network is trained to perform. The inputs can be represented as embeddings, where each sequence element is mapped to a fixed-size randomly initialised vector [195–197]; they could also be represented as matrices where each row is an event and each column is a time-bin [190]; or as graphs [198]. To process the input representations, researchers have used feedforward neural networks (NN)[190], convolutional NNs [195], recurrent NNs [190], and transformers [196–198]. The networks have been trained using supervised learning (e.g. to predict

subsequent adverse events [190, 198]), self-supervised learning (e.g. to predict the next medical event in the sequence [197]), or semi-supervised and unsupervised learning (e.g. reconstructing the original event sequence or some properties of it from a compressed representation of the sequence [195, 196]). There is variability to degree in which time information was included in the mentioned studies (included in [190, 196]), although all of the architectures can be modified to incorporate event times. We illustrate these studies below, noting that this is not an exhaustive review, but an illustration of the different methods that are possible. Furthermore, there has been a review on modelling temporal EHR data with neural networks [199], which does not directly focus on embedding medical event sequences, but the model architectures reported could likely be used for that purpose.

Landi et al [195] modelled sequences of medical concepts. After preprocessing, each patient sequence was divided into subsequences of length L (padded if needed). Each concept in the sequence was then mapped to an embedding of size N, so the subsequence was represented as an L by N matrix. This learnable embedding matrix was passed through convolution layers with multiple output channels, each sliding a window over the subsequence to extract features from it. The extracted features were then passed through a feedforward neural network that was trained to reconstruct the original subsequence, making it an autoencoder model. Finally, outputs of an internal ('hidden') layer of the network were used as the sequence representation. Landi et al showed that these representations distinguished patients with different complex diseases, as well as different disease subtypes, despite the model not being trained to specifically recognize these. Their method, however, did not incorporate event times, although it could be modified to do so by using an L by T matrix for the input instead where T denotes the number of time bins.

De Oliveira et al [190] converted each event sequence to a matrix, where each row describes an event that can occur in the sequence, each column is a time bin, and each cell represents the number of occurrences of the event in the time bin. They then processed the sequence-matrices using feedforward neural networks, with or without applying a recurrent neural network over the time bins, for predicting

mortality after cardiac implant surgery. They additionally used autoencoder neural networks to reconstruct the original sequence-matrix, but trained these such that the compressed latent representation of the event sequence would distinguish patients at risk of mortality. Overall, these models extract features from patient event sequences and the outputs of their internal ('hidden') layers could be used for comparing the sequences, although this was not the focus of the authors. As they represented the input sequences using time bins, their method incorporates information about event times.

Kraljevic et al [197] used the MedCAT natural language processing pipeline [200] to extract medical concepts (disorders, medications, findings, procedures) from the clinical documents of more than 800,000 patients. The data of each patient was then represented as a sequence of medical concepts (ordered by the date of the document from which it was extracted). They then trained a transformer model [88] to predict the next concept in the sequence, and provided a web app that allows to predict the next disorder/medication/finding/procedure for any input sequence of concepts. Although the paper does not discuss how event times were processed, it seems likely that the relative timing of medical concepts was not inputted to the model (transformers by default do not incorporate time information). Nevertheless, outputs of the transformer-encoder block corresponding to the last element in the sequence could be used as representations of the entire sequence (as they integrate information over all sequence elements), and could be used for comparing event sequences.

Zeng et al [196] used a transformer model on sequences of medical visits: each visit was associated with multiple characteristics (such as date, diagnoses, procedures, and resource utilisation type) which were mapped to their own embedding vectors, and pooled to obtain a single embedding per visit. An additional embedding for patient demographics was prepended to the sequence of these medical visit embeddings, which were all passed through transformer-encoder layers. Transformer-decoder layers were then used to reconstruct the medical events and expenditure from the encoder output for the patient embedding. When the patient embeddings were visualised after model training, they indeed distinguished patients with different

types of disease and resource utilisation patterns. Importantly, their sequence modelling pipeline incorporated event times, which were mapped to an embedding using sine and cosine functions with different periods, similarly to the positional encoding in the original transformer paper [88].

Xu et al [198] represented medical visits as hypergraphs, to better model the interactions between procedures, diagnoses, medications and services recorded at each visit. They applied a transformer model to the nodes and edges of the graph to learn relationships within and between visits, and use these for downstream risk prediction. Even though in their study, they did not extract embeddings for medical event sequences to quantify the similarity of sequences, we expect this to be possible given their model architecture, and we highlight their study as a different example of how medical event sequences can be represented for modelling.

Fast feature extraction from graphs

An event sequence can be represented as a directed graph, containing a node for each event that appeared in the sequence, and an edge between the nodes if the two events follow each other. For example, the sequence [a, b, b] has two nodes [a, b], and two edges (a to b), (b to b). There are a variety of methods for mapping graphs to vectors [201], including node2vec [202] which attempts to preserve neighborhoods of nodes in the graph. Here, we only review the Sequence Graph Transform [188], as it is fast, can extract short- and long-term dependencies from the sequence, and can be modified to include information about event timings (as we show later).

The sequence graph transform (SGT) is a method of computing embeddings for event sequences that considers all pairs of events that can follow each other [188]. For example, if each event sequence contains one or more events from the set {A, B, C}, the sequence graph transform considers the pairs (A, A), (A, B), (A, C), (B, A), (B, B), (B, C), (C, A), (C, B), (C, C), where the notation (a, b) means "event a followed by event b". The SGT feature for each event pair is computed as $\phi_{a,b} = (1/n \cdot \sum_{i=1}^{n} \exp(-k \cdot |\text{position}_{b,i} - \text{position}_{a,i}|))^{1/k}$, where $|\text{position}_{b,i} - \text{position}_{a,i}|$ is the absolute distance between A and B in the ith instance

of the pair (A, B) in the sequence, n is the number of (A, B) instances in the sequence, and $k > 0$ is a tuning parameter. For example, the sequence "ABAB" contains three instances of event A followed by B, shown in bold: "**AB**AB", and "**A**BA**B**", and "AB**AB**", with distances $(2-1) = 1$, $(4-1) = 3$ and $(4-3) = 1$. Assuming that $k = 1$, the SGT feature for the pair (A, B) in the sequence "ABAB" is thus $\phi_{a,b} = 1/3 \cdot [\exp(-1) + \exp(-3) + \exp(-1)] \approx 0.26$. The SGT feature could optionally be scaled by sequence length to make it length-sensitive. The tuning parameter k only has an effect when there are multiple instances of an event pair (a, b), in which case it controls the degree to which long-term dependencies are extracted: increasing k increases the contribution of the (a,b) pair with the smallest distance between a and b (thus giving less importance to pairs where a and b are farther away)[2]. Ranjan et al have shown that the SGT feature between events A and B is sensitive to both short-term and long-term dependencies between A and B in the sequence; they also showed that SVM classifiers used on top of SGT features performed similarly to or better than feedforward and recurrent neural networks for classifying protein sequences and computer network intrusions (although they did not give enough detail on how the classification labels were constructed for protein data).

Other approaches

Previously discussed methods associated each event sequence with a feature vector, which allows to quantify the similarity of sequences by computing a distance metric between the vectors. A different approach to comparing the similarity of event sequences is to consider the number of edits that are needed to convert one sequence to another: if more edits are required, the sequences are more dissimilar. For example, Aspland et al [203] use a modified Needle-Wunsch algorithm that allows to assign different penalties to the edits, and includes prior knowledge by specifying which medical events can be swapped, defining similar groups of events, and assigning importance weights to the events. However, methods based on edit-distance do not incorporate information about event times, and can thus be

[2]This can be seen from $\lim_{k\to\infty}(1/n \cdot \sum_{i=1}^{n} \exp(-k \cdot d_i))^{1/k} = \lim_{k\to\infty}(1/n)^{1/k} \cdot \exp(-k \cdot d_1)^{1/k} \cdot (1 + \exp(-k \cdot (d_2 - d_1)) + ... + \exp(-k \cdot (d_n - d_1)))^{1/k} = \exp(d_1)$ if $d_1 < d_2 < ... < d_n$ and $k > 0$.

less optimal for medical event sequences. Although it may be possible to modify their algorithm, such that the timing between events can also be edited and would contribute to the distance metric.

Ranjan et al [188] briefly review string kernel methods in which the similarity of two sequences depends on the degree to which they contain similar substrings. However, they note that these methods usually extract local sequence features and may not detect longer-term patterns in the sequence. These methods also do not incorporate event times by default and it is not clear if they could effectively be modified to do so.

6.2 Method

6.2.1 Ethics approval

This analysis uses data from the NIHR HIC CRC database [4]. The protocol for the collection and management of that data has been reviewed and approved by the East Midlands - Derby Research Ethics Committee (REF Number: 21/EM/0028).

6.2.2 Extracting sequences of treatment events

We used data from the Oxford University Hospitals (OUH) NHS FT, Imperial College Healthcare (ICH) NHS FT, and Royal Marsden (RMH) NHS FT, that had been submitted to the NIHR HIC CRC database [4]. We first extracted data items containing demographics, inpatient and outpatient diagnoses and procedures, histopathology and imaging reports, and chemo- and radiotherapy treatment records. We processed the data items to ensure they were comparable across the centres, such as removing non-word characters from ICD-10 procedure codes and ensuring that dates were in a consistent format. We then identified the approximate CRC diagnosis date as the earliest date among inpatient and outpatient ICD-10 codes for CRC (codes starting with C18-C20), and the cancer site from the first three characters of the diagnosis code (C18: colon, C19: rectosigmoid, C20: rectum). For descriptive purposes, we also used a previously developed rule-based algorithm (Chapter 3) to extract T-stages from imaging and pathology reports. We identified

the maximum T-stage within [0, 180] days from diagnosis, giving preference to pathological staging if that was available.

We then used the Office of Population Censuses and Surveys Classification of Intervention and Procedures (OPCS-4) codes to identify local excision surgeries (OPCS-4 codes starting with H402, H412 and H34), radical resection surgeries (OPCS-4 codes starting with H04–H11, H29, H33, X14), and polypectomies (OPCS-4 codes starting with H20 and H23). We also extracted all radiotherapy and chemotherapy events with timestamps, incorporating only the first cycle of each chemotherapy course to simplify the resulting event sequences. We organised the data into an event log format, where each row contained an event from the set {'diagnosis', 'polypectomy', 'local excision', 'radical resection', 'chemotherapy', 'radiotherapy'}, along with its start date. For each event, we also computed a relative time in weeks from the date of diagnosis.

In subsequent analyses, and as recommended by Dr Helen Jones, we only included patients with rectal cancer to see if it is possible to distinguish different patterns of treatments given to a relatively similar group of patients.

6.2.3 Embedding event traces

After preprocessing, the data of each patient consisted of a sequence of events (such as ['diagnosis', 'scan', 'radiotherapy']) and associated relative event times (such as [0 weeks, 1 week, 16 weeks]). We then transformed the data of each patient into a feature vector, such that both the sequence and timing of events contributes to the feature values.

Time-sensitive sequence graph transform (t-SGT)

We extracted features from medical event sequences using a modified SGT algorithm (see Section 6.1.2 for the description of SGT). In the original method, a score is computed for each pair of consecutive events based on the distance d between the events, such that the score is lower when the distance is larger ($\exp(-d)$). The distance is originally computed based on relative event positions, but we instead

used relative event times to make the algorithm time-sensitive. For example, if event B occurred 10 weeks after diagnosis, and event A occurred 3 weeks after, their distance is 7 weeks. We similarly used an exponential decay function, but added a rate parameter r that controls how fast the feature value decays as the time between events increases. Figure 6.1 illustrates the decay function with $r = 0.5$, in which case the score drops from a maximum value of 1 to about 0.3 as the distance in time increases to 6 months. As in the original SGT, we used a parameter k that controls the degree to which long-term dependencies between events are extracted, so that when there are multiple instances of event pair (A, B), the SGT feature is computed as $\phi_{a,b} = (1/n \cdot \sum_{i=1}^{n} \exp(-r \cdot k \cdot d_i))^{1/k}$. In this formula, increasing k will increase the contribution of the event pair instance with the shortest distance. In SGT, feature values are computed for all possible pairs of consecutive events: if the sequences consist of at most v unique events, then v^2 features are extracted from each sequence. We implemented the modified SGT based on Algorithm 1 from the SGT paper [188], and checked that it yields the same values as the original SGT implementation [204] when distances are computed from event positions instead of times. We call the modified SGT algorithm time-sensitive SGT (t-SGT).

We used SGT for three main reasons. Firstly, it can be modified to incorporate event times. Secondly, it is relatively fast: with 4,064 sequences of median length 3, and 36 event pairs, it ran in less than 3 seconds even without processing the sequences in parallel. Thirdly, it allows to control the degree to which short and long-term dependencies are extracted from the sequences: with larger k, only the nearest connections from event A to B contribute to the feature value. We also note that patterns of more than two events can be inferred from pairwise features: the sequence [a, b, c] is distinguished from [a, c, b] by having a feature value for (b, c) but not (c, b); although this statement may not always hold for more complex sequences.

6.2.4 Dimension reduction followed by clustering

Having associated each sequence of medical events with a vector of features, it is possible to compute the similarity between sequences and to cluster them. However,

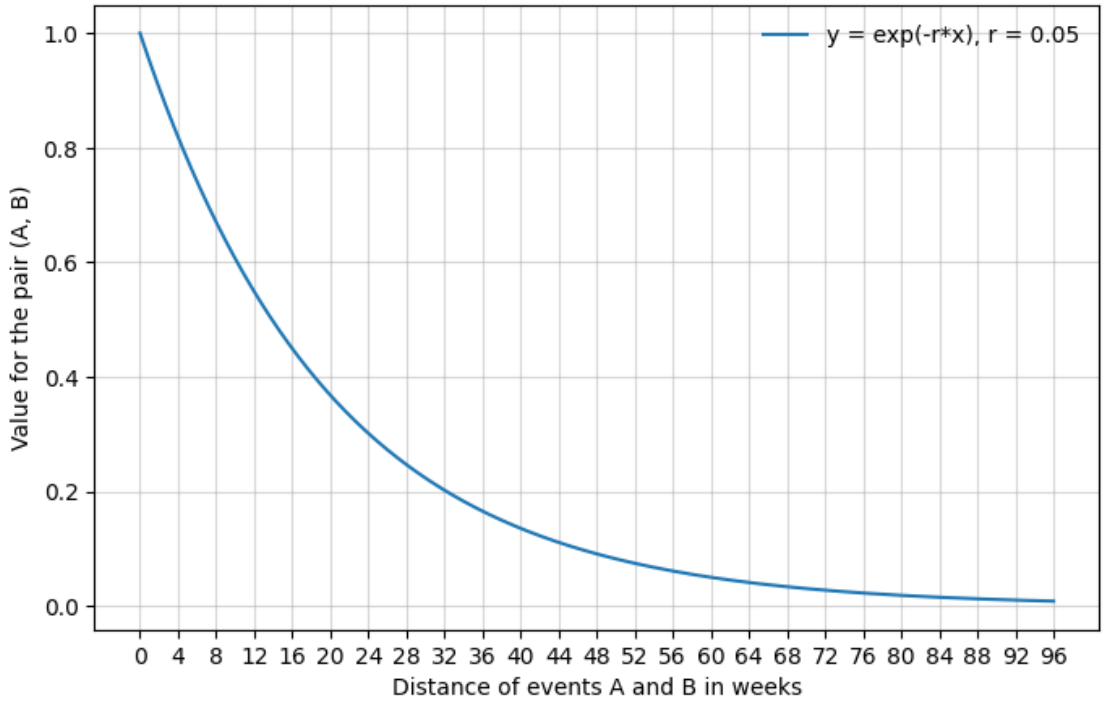


Figure 6.1: How relative time between two events influences their SGT feature score. In the time-sensitive SGT method, a score is computed for each pair of consecutive events, depending on how far the events occur in time. The score is derived using exponential decay with rate parameter r.

it can be desirable to first transform the feature vectors into two-dimensions, so that the similariy of the sequences can be visualized on a scatter plot and interactively explored. We used the PaCMAP method for dimension reduction, because it has been shown to preserve both local and global structure of the input data [205]. Initial exploration also showed that principal components analysis, a commonly employed method, did not provide a good separation of treatment patterns: the event sequence embeddings were almost evenly spread out over the two-dimensional space.

After dimension reduction, we clustered the two-dimensional representations of event sequences using HDBSCAN [206], a hierarchical density-based clustering algorithm. HDBSCAN allows for clusters with variable shapes and densities, and does not require the number of clusters to be specified in advance (instead, minimum acceptable cluster size is defined).

6.2.5 Interactive visualisation of patient timeline clusters

We implemented the previously described dimension reduction and clustering pipeline in an interactive Dash [207] app. The app displays all event sequences on a scatter plot, using their dimension-reduced sequence embeddings (panel B in Figure 6.3). It also colors the event sequences according to their cluster membership, and additionally displays average SGT feature values for each cluster (panel A in Figure 6.3; also see Results). An essential feature of the app is the ability to visualise a random sample of patient treatment timelines for any selected region of the event sequence scatter plot (panel C in Figure 6.3). The app also has a collapsible settings panel that allows to select a subset of patients for analysis, to adjust the parameters of the clustering pipeline, and to adjust the graph that displays patient timelines (Figure 6.2). Importantly, all graphs were implemented in *plotly*[208] and are thus interactive, allowing the user to zoom in, to select subregions, and adjust axis limits. (For clarity, Dash is a general tool for producing interactive dashboards. In this case, the event sequences were processed under the hood by the modified SGT feature extraction algorithm that I implemented, and by the PaCMAP dimension reduction algorithm, and the resulting data was then interactively displayed via features of Dash and *plotly*.)

6.2.6 Software

The time-sensitive SGT algorithm was implemented in Python 3.9 using *numpy* (v1.23.1)[97] and *pandas* (v1.4.3)[98]. We also used the original SGT code from the *sgt* package (v2.0.3)[204] to check our implementation. Dimension reduction was performed with PaCMAP (v0.7.0)[205], and clustering with HDBSCAN (v0.8.28)[206]. The pipeline was implemented as an interactive app in Dash (v2.11.1)[207], with interactive figures created using *plotly* (v5.6.0)[208].

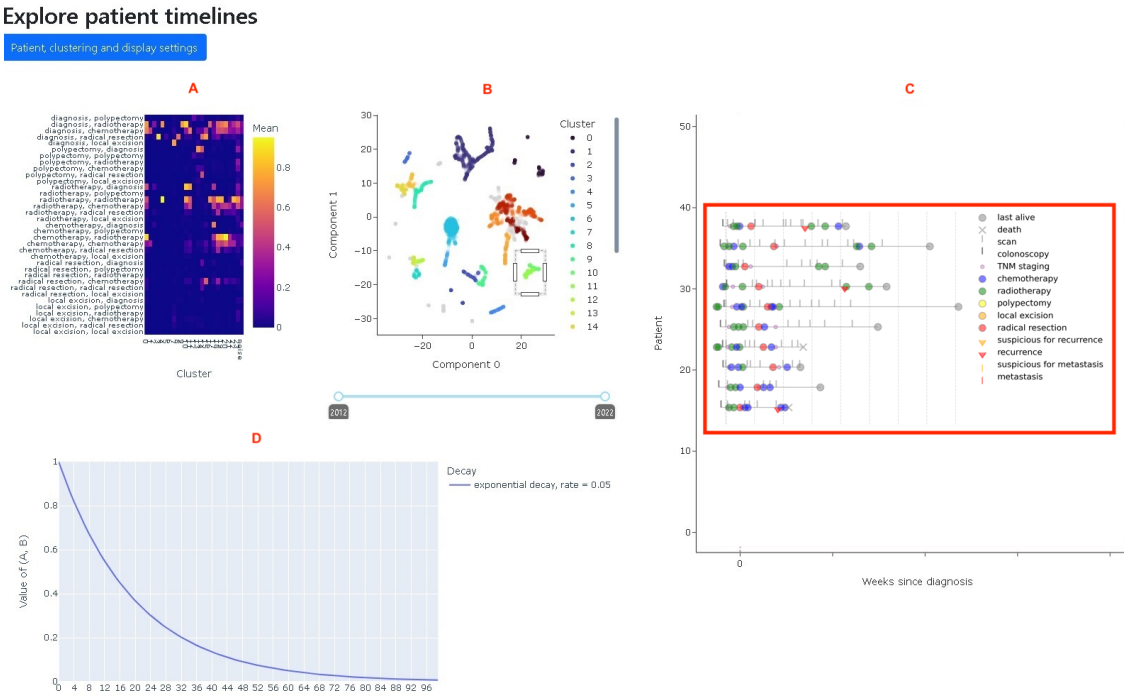


Figure 6.2: Display panel of the Dash app for exploring and clustering patient event sequences. (A) Average values for SGT features in each cluster are shown on the left, which can help to interpret the clusters. (B) Patient event sequence embeddings are displayed on a scatter plot and colored according to their cluster. (C) A random sample of patient timelines is visualised using patient timeline plots, where each patient is represented as a line and each event as a different symbol on the line; the actual timelines are not shown in this screenshot to avoid displaying patient-level data and have been replaced with hypothetical timelines taken from a previous figure published by the author [4]. If the user selects a region on the scatter plot in B, a sample of timelines from this region is displayed. (D) The exponential decay function used for computing SGT features is shown to help choose the value of the rate parameter. The function describes how fast the score for each consecutive event pairs decreases as their distance in time increases.

6.3 Results

6.3.1 Descriptives

There were 4,064 patients with diagnosis codes for rectal cancer across the three biomedical research centres (Imperial, Oxford and Royal Marsden NHS FTs); demographics are summarised in Table 6.1. Note that for anonymity, we do not provide patient numbers separately for the BRCs, and we represent each BRC with a randomly assigned letter. As our aim was to discover broad treatment patterns, we extracted all instances of chemotherapy, radiotherapy, local excision, radical resection, and polypectomy events for all patients (see Methods). The

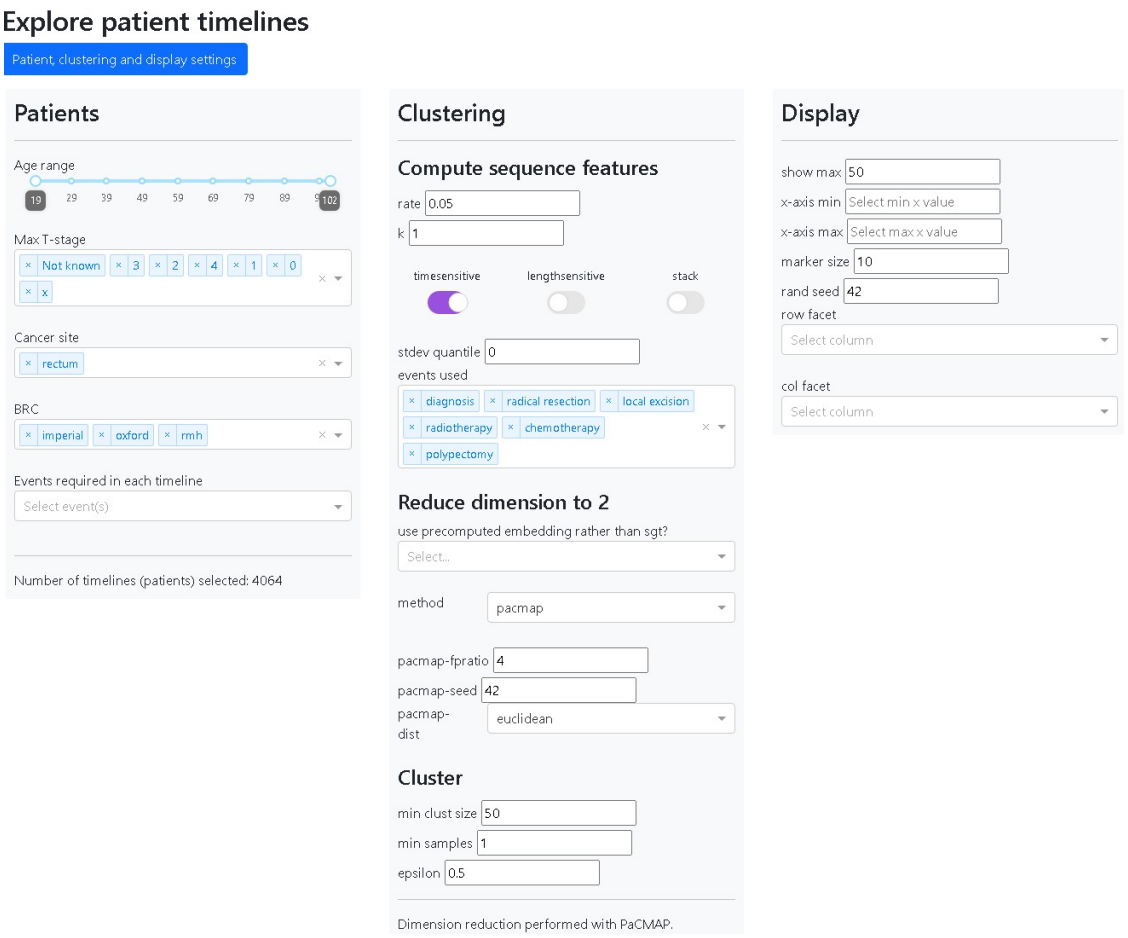


Figure 6.3: Settings panel of the Dash app for exploring and clustering patient event sequences. The left subpanel allows to select a subgroup of patients for analysis: for example, based on their age or site of cancer. The middle subpanel allows to adjust settings of the feature extraction, dimension reduction and clustering algorithms. The right subpanel controls how timelines of patient event sequences are displayed, such as the maximum number of timelines that are displayed.

extracted treatment event sequences were relatively short, with half of the sequences containing more than 3 events, and 25% containing more than 5 (Table 6.1).

Table 6.1: Patient cohort for the clustering analysis

Characteristic	Value, n (%)
Number of patients	4064
Gender	
F	1452 (35.7%)
M	2612 (64.3%)
Age	
18-39.9	229 (5.6%)
40-49.9	323 (7.9%)
50-59.9	814 (20.0%)

Continued on next page

Table 6.1 – continued from previous page

Characteristic	Value
60-69.9	1162 (28.6%)
70-79.9	959 (23.6%)
≥80	577 (14.2%)
T-stage (extracted)*	
0	12 (0.3%)
1	128 (3.1%)
2	232 (5.7%)
3	376 (9.3%)
4	58 (1.4%)
Not known	3257 (80.1%)
CRC-relevant treatments	
No treatments recorded	1075 (26.5%)
Chemotherapy	2005 (49.3%)
Radiotherapy	1797 (44.2%)
Local excision	179 (4.4%)
Radical resection	874 (21.5%)
Outcomes**	
Death	1513 (37.2%)
Length of treatment event sequences***	
Median (25th, 75th)	3 (2, 5)
Min, max	1, 20

Notes. *T-stage was extracted from radiology and pathology reports using a pattern-matching algorithm. **Other relevant outcomes, such as recurrence and metastasis, were not included as they were not available for one of the BRCs. ***Each sequence contained one or more events from the set {'diagnosis', 'local excision', 'radical resection', 'radiotherapy', 'chemotherapy', 'polypectomy'}.

6.3.2 The interactive clustering app enabled to discover sequences with different treatment patterns, but some differences were probably not clinically meaningful

We tuned parameters of the clustering pipeline using the interactive Dash app (see Section 6.2.5), until we found a clustering where distinct treatment patterns appeared in each cluster, when visualising a random sample of patient timelines within the clusters. We used rate = 0.05 and k = 1 for computing SGT features, FP-ratio of 4 for dimension reduction, and a minimum cluster size of 50.

The two-dimensional representations of patient event sequences are displayed in Figure 6.4. Visual inspection showed that the event sequences could have been grouped into approximately 10 clusters. However, partitioning the larger clusters into smaller ones seemed to provide more meaningful groupings, as it better distinguished

certain treatment patterns (such as local excision surgeries from less invasive polyp removals). We therefore applied the HDBSCAN clustering algorithm with a minimum cluster size of 50, which yielded 24 clusters plus a noise cluster (Figure 6.4).

An essential feature of the clustering app is the ability to visualise a random sample of patient treatment timelines for any selected region of the two-dimensional event sequence scatter plot shown in Figure 6.4. While it would be helpful to show a sample of these treatment timelines per cluster, we do not display these for extra precaution as they contain patient level data. We instead report one treatment pattern motif per cluster, derived by visually examining the treatment timelines within the cluster (Figure 6.5). Some of the motifs were simple, as many treatment sequences were short. For example, clusters 3, 5 and 8 mainly contained the radical resection surgery (their sequences were usually of the form ['diagnosis', 'radical resection']), distinguished by how close the resection occurred to the inferred date of diagnosis. Clusters 0, 10 and 11, on the other hand, contained variations of radiotherapy (r) and chemotherapy (c) sequences occurring close to diagnosis date, such as 'r-r-r-c-r' (cluster 0), 'r-r-r-c' (cluster 10), 'r-r-c' (cluster 11). In this case, it was not clear if these patterns should be considered clinically different enough, although the feature extraction algorithm was able to distinguish these (see Discussion).

The clusters can additionally be interpreted by visualising the average value of each SGT-feature per cluster (Figure 6.6). An SGT feature is computed for each consecutive pair of clinical events (allowing for other events to occur in-between), and the value is larger if the two events occur closer in time. This shows aspects of the previously described patterns: for example, in cluster 0, the SGT feature for radiotherapy followed by chemotherapy is visible, whereas in clusters 10 and 11 its average value is close to zero. Similarly, the SGT feature for 'diagnosis $\rightarrow$ radical resection' is prominent for clusters 3, 5, 8; and its value is stronger when resection is closer to diagnosis. While helpful, these graphs are not as informative as the timeline graphs because it is harder to map characteristics of event pairs to patterns involving more than two events.

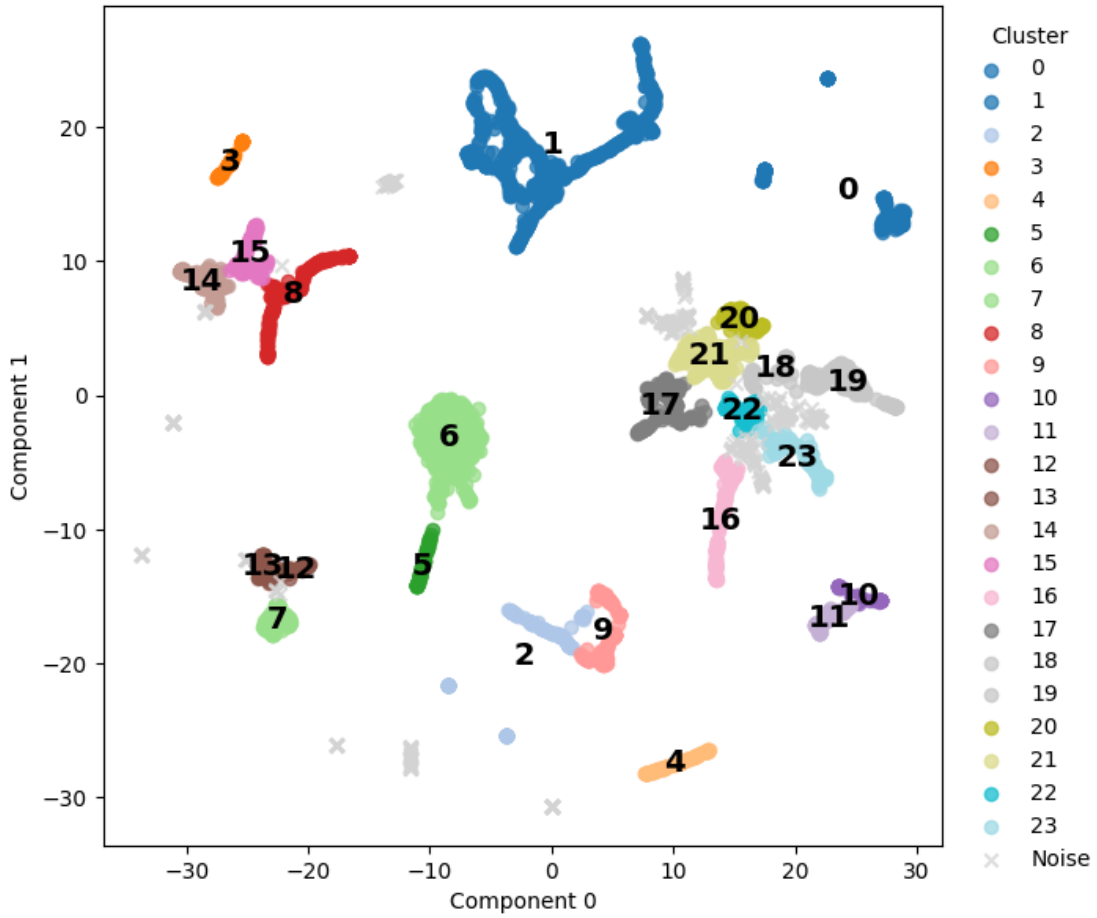


Figure 6.4: Two-dimensional representation and clustering of treatment sequences for rectal cancer patients. Sequences of treatment events and event times were extracted for each patient with rectal cancer from the NIHR HIC CRC database. The treatment events were defined using OPCS-4 procedure codes and grouped into broad categories of 'local excision', 'radical resection', 'radiotherapy', 'chemotherapy', and 'polypectomy'. The event sequences, along with relative event times, were then transformed to fixed-width feature vectors with time-sensitive Sequence Graph Transform (SGT), a modification of the original SGT algorithm [188]; reduced to two dimensions with PaCMAP [205]; and clustered using HDBSCAN [206]. Each point in the figure represents a treatment sequence for one patient (e.g. 'diagnosis, chemotherapy, local excision').

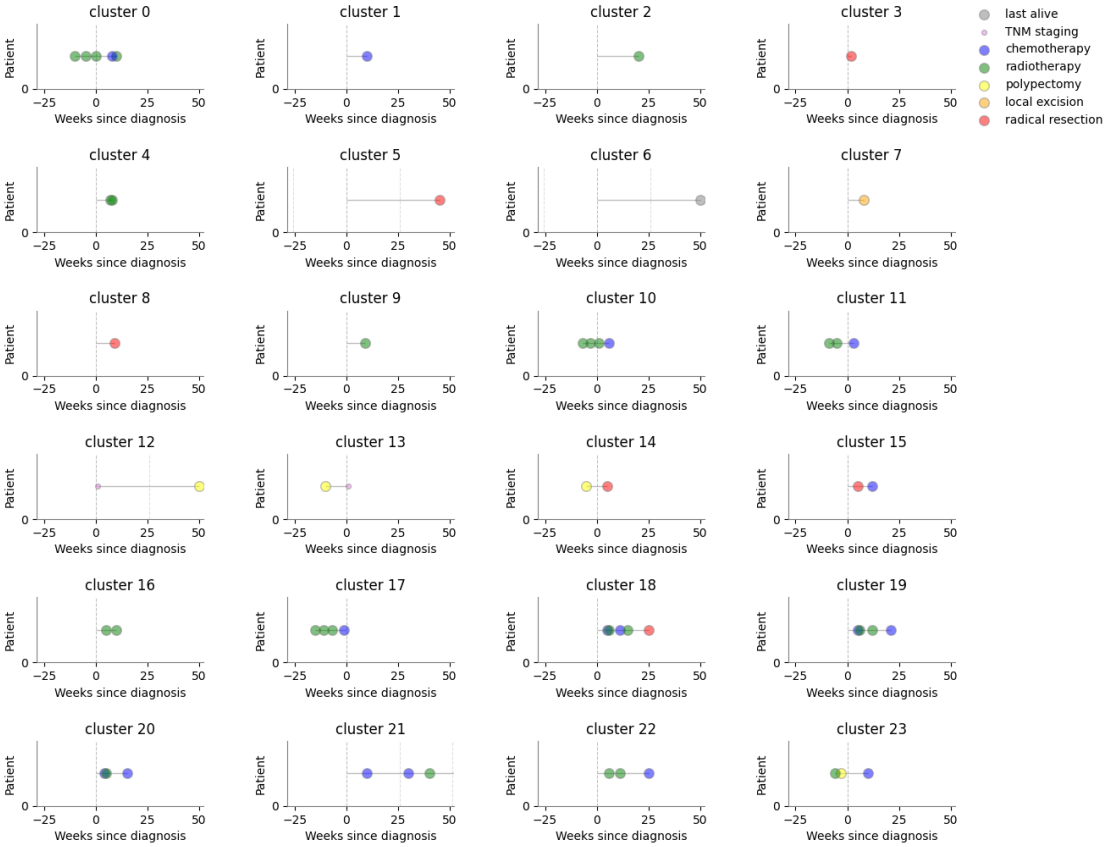


Figure 6.5: Motifs of treatment patterns discovered through interactive clustering of treatment sequences of rectal cancer patients. Sequences of treatment events and event times were extracted for each patient with rectal cancer from the NIHR HIC CRC database. The event-sequences were transformed to feature vectors using a time-sensitive modification of the SGT [188] algorithm, reduced to two dimensions with PaCMAP [205], and clustered with HDBSCAN [206]. Random samples of treatment sequences were visualised for each cluster, and outstanding motifs were manually extracted for each cluster. The motifs are shown instead of the actual treatment timelines, because it was not clear if the treatment timeline plots were anonymous enough.

6.3.3 The clusters can be examined using descriptive statistics and survival curves, but these would be more useful if stratified by initial disease profile

Once the clusters of treatment event sequences were discovered, it was possible to compute descriptive statistics (Table 6.2), examine survival profiles for mortality (Figure 6.8), and to study whether some research centres that contributed data were more represented in some of the clusters (Figure 6.7). The descriptive statistics additionally confirmed that some treatment patterns were more dominant in some clusters (e.g. nearly all patients in cluster 3 had major surgeries), and some clusters

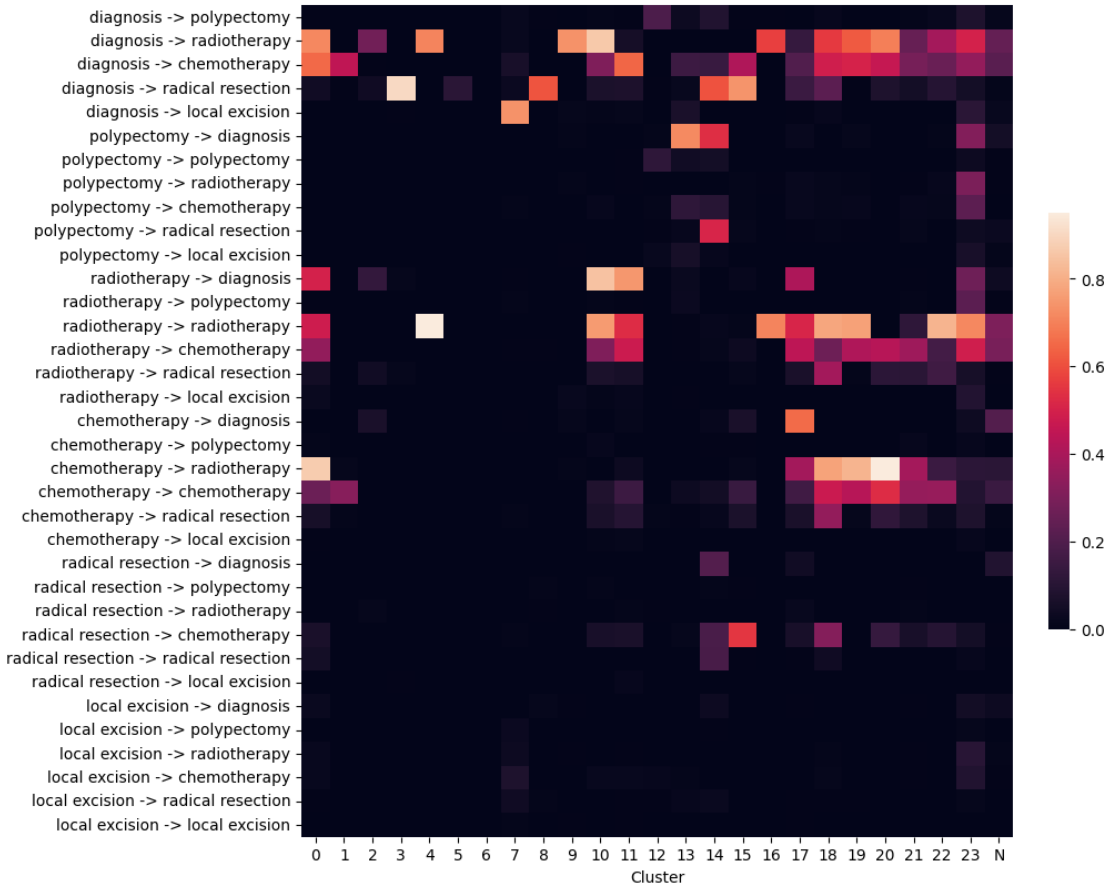


Figure 6.6: Average SGT feature values for each pair of consecutive treatment events within each cluster. Sequences of treatment events and event times were extracted for each patient with rectal cancer from the NIHR HIC CRC database. The event-sequences were transformed to feature vectors using a time-sensitive modification of the SGT [188] algorithm, reduced to two dimensions, and clustered. The SGT algorithm computes a score for each pair of consecutive events, with the value being higher if the two events occur closer in time. All patient treatment event sequences contained one or more events from the set {'diagnosis', 'local excision', 'radical resection', 'radiotherapy', 'chemotherapy', 'polypectomy'}.

tended to have older patients (e.g. median age in cluster 4 was 82). The survival graphs indicated that there were probable differences in mortality between the clusters, and the research centre graph showed that some research centres were much more prevalent in some clusters (e.g. centre B in cluster 14).

Our collaborating clinician Dr Helen Jones reviewed the descriptive statistics and graphs in detail, examining the clusters by procedure type (e.g. she grouped clusters 2, 4, 9, 16, as these patients only had records for radiotherapy). While she found the analysis interesting, she noted that it was not clear if it provided useful insight

into how patients with rectal cancer were treated. This is partly because differences in observed treatment patterns and outcomes were probably confounded by patient characteristics at diagnosis – for example, cancer T-stage was not known for most patients (80%), and so it was not possible to group patients by similar disease severity. On the other hand, if we could first select a subpopulation of patients that have similar disease severity and comorbidities at the time of diagnosis, we could better examine whether patients with similar conditions are treated differently across hospitals (variation in practice), and whether some treatment patterns are more likely to yield better outcomes (see Discussion).

In addition, it can be hard to distinguish differences in treatment patterns from differences in data recording practices across the research centres. For example, a prominent treatment pattern in cluster 14 was polypectomy followed by radical resection surgery, which was more prevalent in research centre B (approximately 15% of patients) than in A and C (less than 5% of patients). However, this could simply mean that polypectomy was not recorded using OPCS-4 procedure codes in centres A and C (procedure codes were used in this analysis), but perhaps instead recorded in endoscopy reports that describe polypectomies.

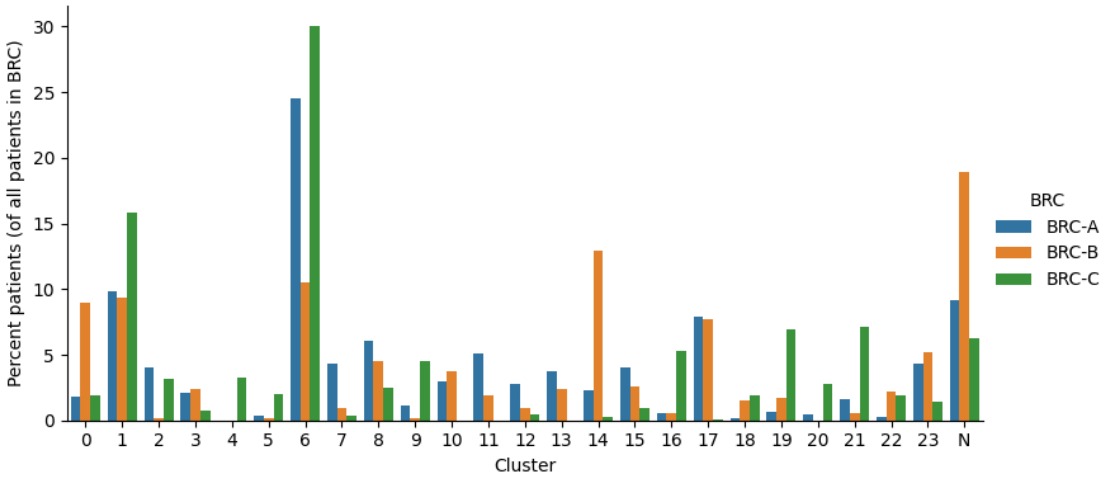


Figure 6.7: Distribution of patients from each research centre over the identified event sequence clusters. For each participating research centre, we computed the percentage of their patients falling to each cluster (the centres were represented anonymously as A, B, C). This allows to judge whether some research centres are more prominently represented in some clusters, controlling for the different number of patients within the centres.

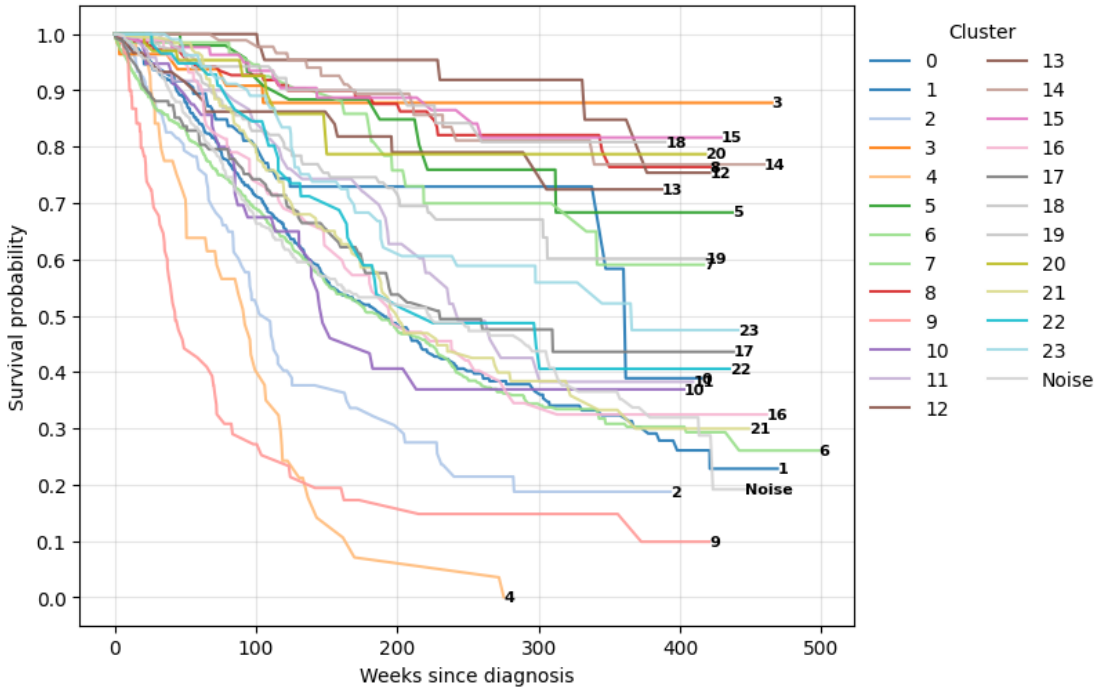


Figure 6.8: Kaplan-Meier survival curves for patients within each event sequence cluster. The outcome was mortality, and time until the outcome was computed from the inferred date of cancer diagnosis. Confidence intervals are not shown for clarity.

Table 6.2: Descriptive statistics for patient event sequence clusters

Cluster	Size	Gender		Age	T-stage						Treatments					Outcomes
	N	F	M	Med (25, 75)	-	1	2	3	4	NA	CT	RT	PP	LE	RR	Death
0	115	45 (39.1)	70 (60.9)	69 (58, 76)	-	3 (2.6)	3 (2.6)	3 (2.6)	2 (1.7)	104 (90.4)	115 (100.0)	115 (100.0)	6 (5.2)	7 (6.1)	22 (19.1)	26 (22.6)
1	533	190 (35.6)	343 (64.4)	64 (55, 71)	1 (0.2)	6 (1.1)	29 (5.4)	80 (15.0)	19 (3.6)	398 (74.7)	533 (100.0)	42 (7.9)	17 (3.2)	3 (0.6)	20 (3.8)	239 (44.8)
2	123	49 (39.8)	74 (60.2)	72 (64, 82)	-	3 (2.4)	7 (5.7)	4 (3.3)	-	109 (88.6)	22 (17.9)	104 (84.6)	3 (2.4)	1 (0.8)	14 (11.4)	68 (55.3)
3	57	17 (29.8)	40 (70.2)	67 (60, 77)	1 (1.8)	6 (10.5)	11 (19.3)	15 (26.3)	2 (3.5)	22 (38.6)	-	2 (3.5)	8 (14.0)	2 (3.5)	57 (100.0)	5 (8.8)
4	73	24 (32.9)	49 (67.1)	82 (72, 88)	-	1 (1.4)	1 (1.4)	-	1 (1.4)	69 (94.5)	-	73 (100.0)	-	-	-	39 (53.4)
5	51	13 (25.5)	38 (74.5)	59 (50, 65)	-	-	-	1 (2.0)	-	50 (98.0)	1 (2.0)	-	8 (15.7)	1 (2.0)	51 (100.0)	9 (17.6)
6	1047	374 (35.7)	673 (64.3)	64 (55, 75)	1 (0.1)	32 (3.1)	33 (3.2)	69 (6.6)	8 (0.8)	904 (86.3)	17 (1.6)	30 (2.9)	21 (2.0)	2 (0.2)	15 (1.4)	454 (43.4)
7	69	28 (40.6)	41 (59.4)	68 (63, 79)	-	22 (31.9)	16 (23.2)	1 (1.4)	-	30 (43.5)	12 (17.4)	5 (7.2)	17 (24.6)	69 (100.0)	8 (11.6)	15 (21.7)
8	156	53 (34.0)	103 (66.0)	66 (54, 73)	1 (0.6)	7 (4.5)	50 (32.1)	44 (28.2)	3 (1.9)	51 (32.7)	9 (5.8)	3 (1.9)	25 (16.0)	2 (1.3)	156 (100.0)	21 (13.5)
9	118	47 (39.8)	71 (60.2)	80 (68, 86)	-	2 (1.7)	1 (0.8)	3 (2.5)	-	112 (94.9)	5 (4.2)	118 (100.0)	2 (1.7)	5 (4.2)	1 (0.8)	60 (50.8)
10	58	25 (43.1)	33 (56.9)	70 (62, 79)	-	-	2 (3.4)	1 (1.7)	-	55 (94.8)	40 (69.0)	58 (100.0)	8 (13.8)	5 (8.6)	21 (36.2)	29 (50.0)
11	74	25 (33.8)	49 (66.2)	65 (56, 74)	-	1 (1.4)	2 (2.7)	6 (8.1)	1 (1.4)	64 (86.5)	74 (100.0)	74 (100.0)	9 (12.2)	5 (6.8)	25 (33.8)	30 (40.5)
12	50	18 (36.0)	32 (64.0)	68 (61, 76)	-	12 (24.0)	7 (14.0)	8 (16.0)	-	23 (46.0)	2 (4.0)	1 (2.0)	50 (100.0)	4 (8.0)	3 (6.0)	5 (10.0)
13	60	20 (33.3)	40 (66.7)	66 (58, 72)	1 (1.7)	9 (15.0)	11 (18.3)	11 (18.3)	-	28 (46.7)	19 (31.7)	3 (5.0)	60 (100.0)	6 (10.0)	4 (6.7)	12 (20.0)
14	105	33 (31.4)	72 (68.6)	67 (60, 73)	1 (1.0)	4 (3.8)	8 (7.6)	16 (15.2)	2 (1.9)	74 (70.5)	43 (41.0)	4 (3.8)	80 (76.2)	5 (4.8)	105 (100.0)	13 (12.4)
15	87	38 (43.7)	49 (56.3)	62 (51, 68)	-	-	14 (16.1)	37 (42.5)	11 (12.6)	25 (28.7)	87 (100.0)	12 (13.8)	13 (14.9)	1 (1.1)	87 (100.0)	11 (12.6)
16	130	41 (31.5)	89 (68.5)	68 (59, 78)	-	-	2 (1.5)	2 (1.5)	-	126 (96.9)	3 (2.3)	130 (100.0)	3 (2.3)	-	3 (2.3)	50 (38.5)
17	143	52 (36.4)	91 (63.6)	64 (54, 74)	1 (0.7)	1 (0.7)	6 (4.2)	17 (11.9)	3 (2.1)	115 (80.4)	141 (98.6)	143 (100.0)	12 (8.4)	4 (2.8)	40 (28.0)	60 (42.0)

Continued on next page

Cluster	Size	Gender		Age	T-stage					Treatments					Outcomes
	N	F	M	Med (25, 75)	-	1	2	3	4	NA	CT	RT	PP	LE	RR	Death
18	54	23	31	60	-	2	2	7	-	43	54	54	8	1	47	9
		(42.6)	(57.4)	(50, 69)		(3.7)	(3.7)	(13.0)		(79.6)	(100.0)	(100.0)	(14.8)	(1.9)	(87.0)	(16.7)
19	175	61	114	63	-	-	-	8	1	166	175	175	14	1	16	47
		(34.9)	(65.1)	(55, 71)				(4.6)	(0.6)	(94.9)	(100.0)	(100.0)	(8.0)	(0.6)	(9.1)	(26.9)
20	69	28	41	60	1	1	2	4	-	61	69	69	2	-	23	8
		(40.6)	(59.4)	(50, 70)	(1.4)	(1.4)	(2.9)	(5.8)		(88.4)	(100.0)	(100.0)	(2.9)		(33.3)	(11.6)
21	185	73	112	62	1	1	3	17	2	161	185	185	14	2	52	87
		(39.5)	(60.5)	(51, 70)	(0.5)	(0.5)	(1.6)	(9.2)	(1.1)	(87.0)	(100.0)	(100.0)	(7.6)	(1.1)	(28.1)	(47.0)
22	59	21	38	62	-	-	-	6	2	51	52	59	4	-	21	26
		(35.6)	(64.4)	(55, 69)				(10.2)	(3.4)	(86.4)	(88.1)	(100.0)	(6.8)		(35.6)	(44.1)
23	116	29	87	67	2	7	8	5	-	94	107	116	65	27	34	37
		(25.0)	(75.0)	(58, 75)	(1.7)	(6.0)	(6.9)	(4.3)		(81.0)	(92.2)	(100.0)	(56.0)	(23.3)	(29.3)	(31.9)
Noise	357	125	232	68	1	8	14	11	1	322	240	222	41	26	49	153
		(35.0)	(65.0)	(59, 78)	(0.3)	(2.2)	(3.9)	(3.1)	(0.3)	(90.2)	(67.2)	(62.2)	(11.5)	(7.3)	(13.7)	(42.9)

Notes. For age, median and 25th and 75th percentiles are given. Treatments: CT - chemotherapy, RT - radiotherapy, PP - polypectomy, LE - local excision, RR - radical resection.

6.3.4 Incorporating event times in the analysis was useful for distinguishing planned and unplanned treatment patterns in at least one instance

We reran the clustering pipeline with the original SGT feature extraction method that does not incorporate information about event times (see Section 6.1.2). In this case, the distance between any pair of events is computed based on the position (but not timing) of the events. We used the same parameter for k (1), and a rate parameter of 0.3 for comparability (so that rate multiplied by median time in the time-sensitive analysis would be equal to rate multiplied by 1 unit in the time-insensitive analysis; the original SGT method does not include a rate parameter [188] and ignoring it by setting it to 1 did not make a difference to the results).

We first noticed that sequence embeddings clustered into smaller, more granular points when they were visualised in two-dimensions (Figure 6.9). This may have been because when event times are ignored, there are less differences between sequences (e.g. all sequences of the form 'a-b-c' have equal feature vectors regardless of the time between the events); or perhaps because not considering event times can lead to more patterns being extracted (e.g. if a and b are close but b and c are far in 'a-b-c', the time-sensitive algorithm will only extract the pattern a-b as the SGT feature decays exponentially with time, but the time-insensitive algorithm will likely extract 'a-b-c' as the distance between b and c is only 1 unit).

On visual inspection, many clusters contained treatment patterns that were also similar in terms of the timing between events, but there were also clusters with strong differences between event times. For example, Cluster 2 contained sequences were chemotherapy was administered close to surgery (< 10 weeks), or a long time after (> 60 weeks). If time is not considered, these sequences have the same pattern ('surgery, chemotherapy'). However, they should not be clustered together, because the first group probably represents planned treatment sequences, whereas the second probably represents unplanned treatments for subsequent cancer recurrence.

This showed that not considering time *can* lead to undesirable clustering, although it is not a comprehensive comparison between the time-sensitive and the original SGT method.

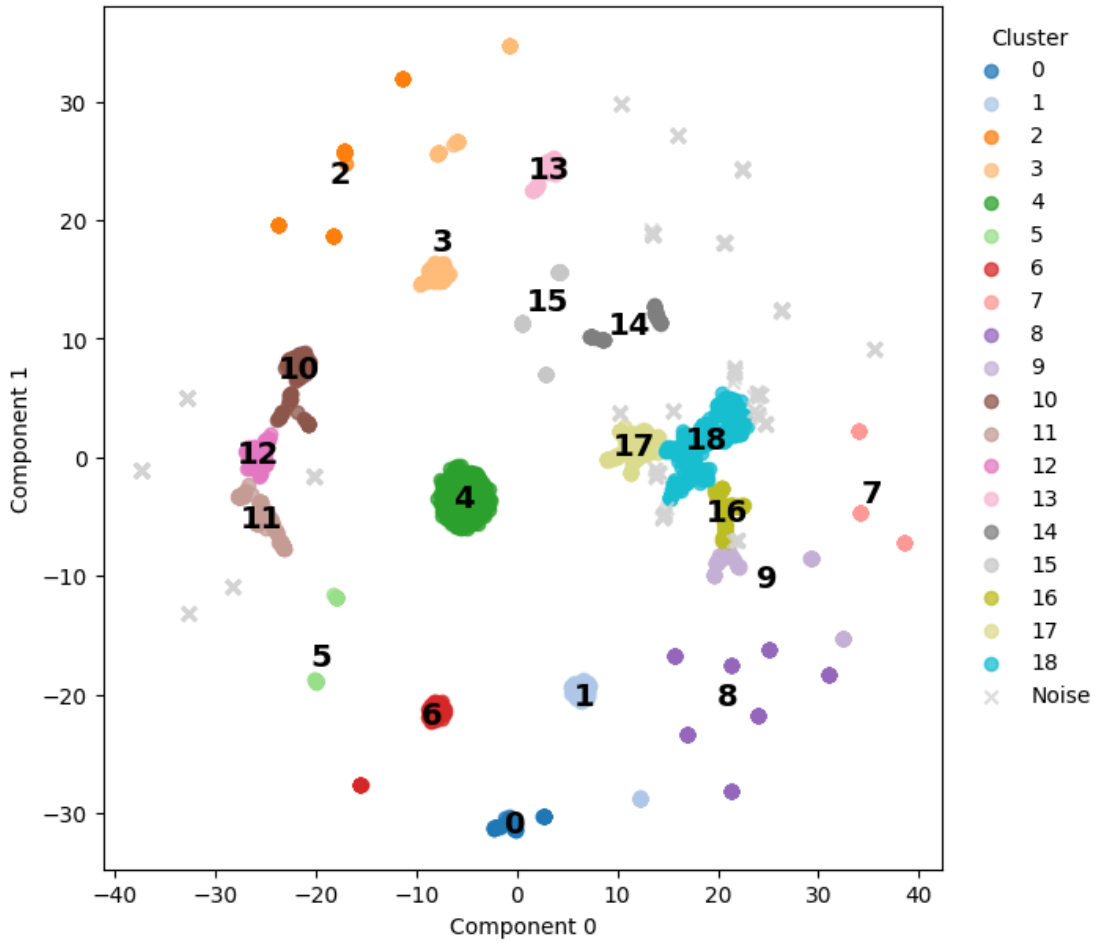


Figure 6.9: Two-dimensional representation and clustering of treatment sequences for rectal cancer patients, based on the original SGT method that is not time-sensitive. Sequences of treatment events and event times were extracted for each patient with rectal cancer from the NIHR HIC CRC database. The treatment events were defined using OPCS-4 procedure codes and grouped into broad categories of 'local excision', 'radical resection', 'radiotherapy', 'chemotherapy', and 'polypectomy'. The event sequences, along with relative event times, were then transformed to fixed-width feature vectors with the Sequence Graph Transform (SGT)[188]; reduced to two dimensions and clustered. Each point in the figure represents a treatment sequence for one patient (e.g. 'diagnosis, chemotherapy, local excision').

6.3.5 Dimension reduction facilitated the exploration of event sequences

After extracting features from each medical event sequence, we reduced the features into two dimensions and displayed them on a scatter plot, so that one can interactively explore the 'space' of event sequences. In this two-dimensional scatter plot (Figure 6.4), each point represents an event sequence, and sequences that occur closer together should have more similar treatment patterns. Indeed, when zooming into different regions of the scatter plot using the interactive app, and visualising the event sequences within these regions, it was possible to see regular changes in treatment patterns. For example, Figure 6.10 shows a close-up view of clusters 8, 14 and 15 (corresponding to the upper left corner in Figure 6.4). The cluster 8 (shown in light green in the rightmost half of the figure) contains event sequences where diagnosis is followed by radical resection surgery, and as one moves from the bottom part of the cluster towards the top part, the distance in time between these events increases. Cluster 15 (shown in orange in the middle) mostly contains sequences with pattern "diagnosis, radical resection, chemotherapy", whereas cluster 14 (shown in yellow on the left) often contains the pattern "polypectomy, radical resection". The "polypectomy, resection pattern" is especially clear in the top left side of cluster 14, and starts to more often contain the pattern "polypectomy, resection, chemotherapy" as one moves down the cluster. In other words, interactively exploring the two-dimensional space of event sequences helps to study the variation of treatment patterns within and between clusters, and to judge whether clustering based on treatment patterns is appropriate at face value (i.e. whether different clusters indeed contain event sequences with similar patterns).

6.4 Discussion

We developed an interactive sequence visualisation and clustering pipeline that allows to quickly discover patterns from medical event sequences. We illustrated it by clustering sequences of treatments given to rectal cancer patients across three NHS research centres that contributed data to the NIHR HIC CRC database.

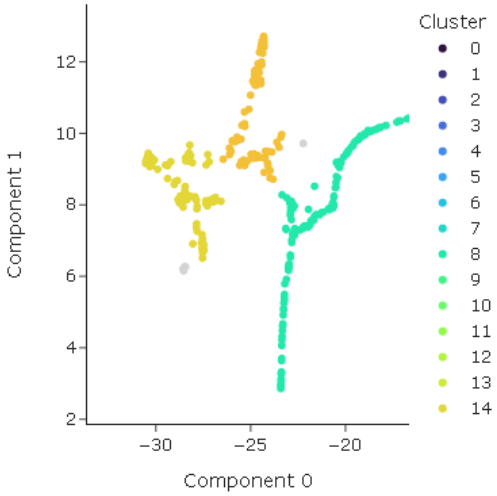


Figure 6.10: A close-up view of treatment sequence clusters within the interactive app. clusters 8, 14 and 15. Note that this is a screenshot from the interactive clustering app, and the color scheme is not the same as in Figure 6.4.

Importantly, the algorithm that extracts features from event sequences takes into account the relative timing between events, which is important for characterising medical event sequences: as illustrated in Results, the same event sequence is more likely to reflect planned treatments if the events occur close in time, than when they are far apart. The pipeline also allows to interactively explore the event sequences in two-dimensional space, which helps to judge whether the obtained clusters were appropriate, and helps to discover treatment patterns that may exist in the data. In addition, the pipeline does not require large computational resources, and should be easily deployable within a clinical information system.

Unfortunately, we could not show that applying the clustering pipeline yielded meaningful insight into the treatment patterns offered to rectal cancer patients within the NIHR HIC CRC database. The method is more likely to yield useful results, however, if it would be applied to a more specific subpopulation of patients that have a similar disease profile at diagnosis, which would require a larger dataset. It is also not clear if the feature extraction algorithm satisfies all properties required for characterising medical event sequences. These and other limitations, along with

future directions, are discussed in more detail below.

6.4.1 Limitations

Clustering metrics. We did not provide a metric for quantitatively analysing the quality of clustering, which is important for rigorously comparing the performance of our clustering pipeline under different settings and to different clustering methods. Clustering quality is usually evaluated using ground-truth labels or the separability of clusters [209]. Methods that rely on ground truth are not immediately useful in our case, because our pipeline is for discovering initially unknown patterns in medical event sequences. However, these methods could be used post-hoc, if a clinician would review the discovered patterns, and assign a randomly selected subset into distinct groups. Methods that evaluate cluster separability are also insufficient: while distinct clusters are desirable, they are meaningless if they do not also distinguish the medical event patterns that exist in the data. On the other hand, as our pipeline is meant for data exploration, it is not crucial to have a quantitative metric. By experimenting with different settings and interactively visualising random samples of event sequences assigned to each cluster, it is possible to discover different sequence patterns as illustrated in the Results.

Lack of prior medical knowledge. Another limitation is that the pipeline does not incorporate prior medical knowledge other than the medical events that are included in patient event sequences. The method treats all included events equally, but some patterns may be more meaningful. It may be possible to modify the method, such that some event pairs are assigned higher weights, or assigned a different exponential decay rate that determines how fast the feature value decreases as the time between events increases. Alternatively, one could possibly develop a clustering method with clinician-in-the-loop: (1) a clinician could be shown a random sample of event sequences divided into clusters using the current method; (2) they could then regroup the sequences into new clusters in an interactive app (e.g. they could reorganize the sequences on an interactive scatter plot, while viewing the actively selected sequences on a patient timeline plot); (3) these new

cluster labels could be used to train a transformer model for sequence classification which would also provide an embedding for each event sequence; (4) finally, a new random sample of event sequences could be provided to the clinician, iterating the process several times. However, such interactive clustering process would likely be more computationally demanding and requires a large sample size, compared to the current lightweight method.

Potentially sparse feature vectors. If each sequence contains a small number of unique elements, as was the case in this analysis, the feature vectors computed by SGT can be sparse (in 75% sequences, more than 30 of the 35 features were zero). If the commonly used Euclidean distance metric is applied for computing similarities between sequences for clustering, then sequences with less elements (such as short sequences) can have more similar scores due the many zeros their vectors share. On the other hand, the cosine similarity metric that only consider nonzero elements may not be appropriate, as it can lead to vectors with very different event times, but a similar relative event spacing, to be considered similar. In our pipeline, we used the PaCMAP method to transform the feature vectors first into two dimensions, which should be less affected by sparsity as it does not attempt to directly preserve Euclidean distances between sequence vectors, but instead uses near, mid-near and further pairs of vectors. However, these pairs are still selected by default using scaled Euclidean distances, and it is not currently clear if this could have undesirable effects for even sparser vectors.

Understanding of theoretical properties. Furthermore, the time-sensitive SGT method may not always work well for extracting local patterns from long sequences. For example, the sequences 'a-b-c' and 'a-b——b-c——a–c' can have very similar values for the SGT features (a,b), (a,c) and (b,c), especially if the tuning parameter k is set > 1 to focus on short-term dependencies. The values for other features, including (b,b) and (c,a), are zero for the first sequence as these subsequences do not occur, and very close to zero for the second sequence because the events occur far apart in time. Based on these SGT features, it is therefore not possible to deduce that the triple 'a-b-c' was present in the first sequence, but not in

the second. In general, more work could be done to better understand (1) what properties should a feature extraction algorithm for medical event sequences have, and (2) are all these properties present for the time-sensitive SGT algorithm. Perhaps a sequence processing algorithm that uses convolutional neural network layers would be better for extracting local sequence features (as it would use a sliding window). On the other hand, the SGT algorithm can also be applied to subsequences of the original sequence (e.g. by splitting the original sequence into chunks), to better focus on local features.

Comparison to more powerful methods. It could be useful to contrast the simpler SGT feature extraction method with a transformer model [88] that can extract complex dependencies from the input sequence due to its multi-headed self-attention mechanism. In fact, we did construct a simple time-sensitive transformer-encoder, using sinusoidal positional encoding based on relative event times, to predict the next event in each medical event sequence. However, the sequence embeddings obtained from the model were not clearly better than the SGT embeddings for distinguishing treatment patterns at face value, and the dataset may have been too small for effectively training the model (< 9000 event sequences for all CRC patients). The transformer model could also be improved using learnable positional embeddings: this could help the algorithm assign different importance to different time points in the patient pathway.

6.4.2 Potential to discover meaningful variations in care

Ultimately, the sequence clustering pipeline can be used and/or further developed to discover meaningful variations in care, such as exploring which treatment and surveillance patterns are associated with better outcomes. This would require several modifications to the analysis.

Firstly, a larger and more complete dataset is needed. It would then be possible to select a more specific group of patients for analysis that have similar disease characteristics and comorbidities at diagnosis, while still retaining a sufficiently large sample size. If different patterns of treatment are discovered through clustering,

this would help ensure that differences in outcomes between the clusters are more likely due to the differences in treatments that were assigned, rather than patient characteristics such as age or disease severity (although there can be other confounding characteristics).

Secondly, it could be useful to focus on a subgroup of patients with complex disease, where it is known that variable treatment strategies are given, and where it is uncertain which strategy is optimal. These could be, for example, CRC patients with synchronous liver metastases (studying variations in treatment for these patients was listed as an important research question by the NIHR HIC CRC theme).

Thirdly, it could be useful to pursue a semi-supervised clustering strategy: the clustering algorithm could be guided to both find patients with similar treatment patterns, and to distinguish patients who have different treatment outcomes (such as differences in rates of cancer recurrence). This may be achieved with neural networks that simultaneously cluster and classify outcomes (e.g. [210]). In this case, it is possible to apply both more complex neural network models as well as simpler models that are built on top of the existing SGT feature extraction algorithm.

This analysis could be conducted on a colorectal cancer dataset for the Thames Valley Region that is likely to become available soon (personal communication with prof. Jim Davies). The clustering pipeline described in this chapter is likely to be useful for exploring more complex temporal treatment patterns (such as "treatment A followed by B and by at least two surveillance scans is associated with better outcomes in patient population C"), and the Thames Valley dataset would allow defining treatment events and patient characteristics with enough detail and over time. Databases that link national cancer datasets with hospital episode statistics and chemoradiotherapy data, such as the CORECT-R repository [211], could also be used for studying treatment variations with the clustering pipeline. National datasets would provide a larger sample and permit studying geographical variation.

6.4.3 Other clinically relevant uses

Other than exploring variations in treatment, the clustering pipeline could also help study associations between temporal healthcare usage patterns and outcomes. For example, do patients who did not return their FIT test have less contact with the healthcare system than those who returned it? Or do cancer patients who are diagnosed as emergency presentations have different prediagnostic healthcare usage patterns compared to those diagnosed via other routes? (I thank Marta Berglund for discussions about how the method could be used for studying healthcare usage patterns.)

The ways to improve are endless, and so are the ways to appreciate what is already here.

7 Conclusion

Contents

The final Chapter summarises the main findings across the thesis, discusses potential contributions beyond research findings, then proceeds with a more personal reflection on learnings, and concludes by discussing future directions.

7.1 Summary of chapters and contributions

This section summarises the main findings for each chapter. The key research questions, findings, and computer code are also briefly outlined in Table 7.1.

7.1.1 Groundwork

This dissertation draws on data from relatively raw EHRs, which needed to be cleaned and understood before being used. In addition, some essential information about cancer was only available in free text format, so it was necessary to develop an information extraction pipeline. These efforts are summarised in Chapters 1 and 2, which together lay the groundwork for subsequent parts of this thesis.

Understanding and visualising electronic health records

In Chapter 2, "Understanding and visualising electronic health records", I review the general issues of using EHRs for health research throughout the data life cycle, discuss common pitfalls and data quality issues, and report on creating minimal data quality and data collation pipelines to support the creation of a NIHR HIC CRC database [4]. I also discuss how visualising the timelines of treatments and other clinical events can help evaluate data quality and gain a more holistic understanding of the data. I conclude by discussing ethical issues, especially those that arise from a lack of data diversity, and reflect on how risk prediction models developed on retrospective EHR datasets can fail when deployed prospectively.

Extracting information about colorectal cancer from free text

In Chapter 3, "Extracting information about the presence, stage, and recurrence of colorectal cancer from free text clinical reports", I report on the development of a lightweight information extraction pipeline. I designed the pipeline to support the creation of a multi-centre NIHR HIC CRC database [4], so that essential information about CRC can be retrieved from free text histopathology and imaging reports. My motivation was to create a software tool that can easily be run in another

hospital system, because not all NIHR HIC participating centres could contribute their anonymised free text reports to a common database.

The information extraction pipeline thus consists of regex-based algorithms that (1) detect if a clinical report discusses current CRC; (2) extract TNM staging scores that describe the severity of cancer; and (3) extract information about the presence and broad anatomical site of cancer recurrence and metastasis. The pipeline additionally includes fine-tuned bidirectional transformer encoder (BERT) models for task (3), as the regex-based model performed less well for anatomical sites. The regex-based algorithms were mainly and iteratively developed on more 64,000 clinical reports from Oxford University Hospitals (OUH) NHS FT, and on more than 56,000 reports from Royal Marsden NHS FT, by creating patterns and examining matches. The transformer models were fine-tuned on a stratified random sample of 1,826 report extracts. An additional sample of unseen future OUH reports was used for evaluating the CRC detection algorithm, but not others due to resource constraints (this work is ongoing for publishing the results).

The CRC detection module had at least 90% sensitivity and at least 84% positive predictive value (PPV) on random samples of pathology reports from the OUH training data and from unseen future OUH data; PPV was lower on imaging reports.

The TNM stage detection algorithm identified the main T/N/M categories with at least 97% PPV and at least 83% sensitivity in both imaging and pathology reports on a random sample from the OUH training data; some loss in sensitivity was due to clinical reports that did not contain numerically reported TNM staging values but still contained enough information to infer staging (excluding these would have yielded $> 95\%$ sensitivity) . It was not possible to evaluate the TNM algorithm in a timely manner on future OUH reports, and this is underway for publishing the results. However, it is unlikely that the training data estimate is too optimistic, because the algorithm was not tailored to perform well on the relatively small random sample of training data that was used for evaluation.

The recurrence detection algorithms correctly detected the presence of recurrence/metastasis with at least 80% sensitivity and PPV, in report extracts that

mentioned recurrence/metastasis (some extracts mentioned recurrence/metastasis in the 'present' sense whereas others were general, negated or historic statements). The performance was more variable for detecting the broad anatomical site of recurrence/metastasis, although the transformer models tended to perform better than the regex-based models, which indicates that additional fine-tuning of these models with active learning could yield even better performance. The performance of recurrence algorithms is likely to be higher on full clinical reports rather than report extracts, and this additional evaluation is also necessary for publishing the results.

The primary contribution to the clinical community is the TNM stage detection algorithm. This is because TNM staging is a widely used system for describing the extent and spread of cancer, and it is recorded in a limited number of formats which makes it likely that the algorithm can generalise to data from other hospitals.

7.1.2 Predicting the risk of colorectal cancer

Many individuals see their general practitioner (GP) with unexplained symptoms indicative of CRC. The Faecal Immunochemical Test (FIT) is commonly used to decide who should be referred to further investigations, which usually includes colonoscopy, an invasive endoscopic investigation of the colon. While FIT can safely rule out majority of individuals from further investigations, it suffers from a less-than-ideal positive predictive value: only about 1 in 6 individuals who test positive have cancer [116]. Researchers have therefore attempted to develop prediction models that combine FIT results with routine data to reduce unnecessary referrals.

External validation of Nottingham CRC risk prediction models

In Chapter 4, "External validation of Nottingham colorectal cancer risk prediction models on the Oxford University Hospitals FIT dataset", I evaluate whether Nottingham-derived conventional CRC risk prediction models perform better than the current clinical practice of referring patients with a FIT result $\geq$ 10 µg Hb/g faeces, on the Oxford University Hospitals FIT (OUH-FIT) dataset. I first discuss the importance of clinically relevant performance metrics: to perform better than

the current clinical practice, the model should have a higher PPV at the same level of sensitivity as FIT $\geq$ 10. If this is the case, the model would detect as many cancers as captured by the current clinical practice, but would lead to less patients being referred to invasive follow-up investigations, which is arguably the motivation for developing the model. I then showed that the Nottingham models did not perform better than the standard clinical practice: FIT $\geq$ 10 had a sensitivity of 84% and PPV of 14%; the Nottingham prediction models had a similar PPV as the FIT test at 84% level of sensitivity, and also similar PPVs at sensitivities greater than 45%. I compared the Nottingham and Oxford populations to discuss the potential reasons why the model may not have validated, also noting that the Nottingham model did not perform better than the FIT test in their own large model development set but did so in their own smaller internal validation set (which is an unusual pattern, and which can provide insight into the factors that determine model performance). I also reported a comprehensive set of performance measures, including calibration and net benefit statistics, to additionally evaluate the Nottingham models, but these are less important than comparing the PPVs of models and FIT at the sensitivity of FIT $\geq$ 10.

Combining the FIT test with routine data to predict colorectal cancer: A machine learning approach

In Chapter 5, "Combining the FIT test with routine data to predict colorectal cancer: A machine learning approach", I comprehensively explore whether routinely collected data can be combined with FIT test results to develop a model that outperforms current clinical practice of referring patients with a FIT result $\geq$ 10 µg Hb/g faeces. I explored this by (1) including a more diverse set of clinical variables for predicting the risk of cancer, (2) employing machine learning models with different degrees of interpretability and flexibility, and (3) using novel ways of fitting models to data. I included 582 predictor variables that represented routinely collected blood tests, as well as procedure/diagnosis/prescription codes, hoping that these variables may better capture the underlying health-status of the patient. I employed simple and interpretable models (logistic regression); complex and

nearly uninterpretable models (artificial neural networks); and models in between (general additive models and decision tree ensembles). Having a spectrum of models helped ensure that complex relationships between predictor variables are more likely to be detected if they exist in the data, and that a clinically explainable high-performing model can also be found in this were the case. I also explored novel ways of fitting models to data by maximizing area under the receiver-operating characteristic (ROC) curve and area under the precision-recall (PR) curve, which can yield better results in imbalanced datasets where the proportion of patients with an outcome is much smaller than without. I employed data for 31,964 patients (453 cancer cases) in the OUH-FIT dataset, and ensured that all models were regularised to reduce overfitting. Five-fold cross-validated results showed that none of the models, predictor variable combinations, or training regimes outperformed the current clinical practice of FIT ≥ 10. I concluded by discussing the limitations of the analysis (which mainly stem from data quality and sample size), and speculated on the future of FIT-test based prediction models.

The principal contributions are the development of a comprehensive model development and evaluation pipeline; highlighting the importance of using clinically relevant performance measures; affirming the need to understand why a model works and not simply showing that it does (in the case of Nottingham internal results); and demonstrating that it is hard to beat the standard clinical practice of FIT ≥ 10.

7.1.3 Working with sequences of clinical events

Risk prediction models described in previous chapters operated on a collection of variables, such as blood test results, without considering the relative position of these events in time. However, electronic patient records are by their nature event sequences, consisting of procedures, diagnoses, scans, blood tests, and more.

In Chapter 6, "A bird's eye view on patterns of care: clustering patient event logs", I present a fast sequence clustering pipeline embedded in an interactive app, that can be used to quickly review thousands of clinical event sequences to discover patterns. I illustrated the clustering pipeline by exploring the treatment patterns

of rectal cancer patients in the multi-centre NIHR HIC CRC database. While it was possible to see that different treatment patterns clustered at face value, that some hospitals were represented more in some clusters, and that some clusters were associated with worse survival outcomes, it was nevertheless not possible to discover anything obviously meaningful about the treatments given to rectal cancer patients, despite our collaborating clinician Dr Helen Jones extensively reviewing the results. It would be more fruitful to apply the clustering pipeline to a more narrowly defined patient population (such as patients with rectal cancer *and* similar disease severity *and* synchronous liver metastases), in which case any differences observed in treatment patterns between hospitals are more likely to reflect variations in clinical practice, and any outcomes observed between the clusters are more likely to be due to treatment pattern differences (even though causality cannot be proven and additional confounders may need to be included). However, this would require a larger and more complete dataset than I currently had access to. Nevertheless, it was possible to show that the method could identify clusters of treatment patterns, and could be used to interactively explore the data. Furthermore, the clustering method and the interactive app are fast and should be easily deployable in hospital systems to facilitate the exploration of EHRs by researchers and data scientists. The clustering method also takes into account the relative timing of events, and is therefore more likely to be suited to clinical event sequences than clustering methods that do not. Furthermore, the sequence processing method embedded in the pipeline could be used to extract patterns from blood test sequences of patients who have had a FIT test result, and thus be used to explore if the inclusion of such time-trend information could improve the performance of FIT-test based prediction models; in that case, inclusion of other sequence processing methods like transformers, and other time-trend extraction methods, would be informative and necessary as well.

Table 7.1: Key research questions, findings and contributions

Chapter	Key research question (RQ)	Main findings	Software
2	What are the key data quality (DQ) issues and ethical aspects to consider when using electronic health records (EHRs) for research?	General pitfalls and data quality quality issues for using EHRs for research were discussed based on recent reviews by other authors and personal experience. A python pipeline for checking basic data quality issues, and "patient timeline plots" for visualising the data (both developed by the author) were briefly described. Data quality issues were discussed through the lens of "EHR data lifecycle", which can be a helpful model for identifying and ameliorating DQ issues. Ethical issues pertaining to research use of EHRs were discussed with specific focus on data diversity, non-maleficience and truthfulness.	Python code was used for generating data quality reports for the NIHR HIC CRC database, but it will not currently be published, as it is likely too specific to the data at hand and may not generalise well to other medical datasets. Code for visualising patient timelines will be published as part of the codebase for Chapter 6.
3	Is it possible to develop a lightweight information extraction pipeline that can detect whether a free text histopathology or imaging report discusses current primary colorectal cancer (CRC) or current recurrent CRC, and that can extract cancer TNM staging scores from anywhere in the report? Lightweight means that the algorithms depend on a small number of packages which can be relatively easily installed.	The CRC detection algorithm performed well on pathology reports, yielding at least 84% sensitivity on all reports, at least 94% sensitivity on non-supplementary reports, and at least 92% PPV; performance on imaging reports was lower. The outputs can be reviewed to increase sensitivity and PPV. The TNM stage detection algorithm identified the main T, N and M categories with at least 97% PPV and 83% sensitivity in imaging and pathology reports; sensitivity was at least 95% when only including reports where TNM staging was explicitly reported. The recurrence detection algorithms correctly detected the presence of recurrence/metastasis with at least 80% sensitivity and PPV, in report extracts that mentioned recurrence/metastasis. Performance was variable for detecting the anatomical site, although transformer models tended to perform better than regex-based models, indicating there is potential to improve the transformers with active learning.	Python code for detecting current primary CRC and recurrent CRC, and extracting TNM staging scores from free text clinical reports will be accessible at `https://github.com/tammandres/textmining` after a paper has been published.
4	Can the Nottingham colorectal cancer risk prediction models outperform the faecal immunochemical test (FIT) for colorectal cancer detection on Oxford University Hospitals (OUH) data? The models would outperform the FIT, if they reduce the number of patients referred to subsequent investigations based on their FIT result while capturing the same number of cancers as FIT.	The Nottingham colorectal cancer risk prediction models did not have higher positive predictive value at the same level of sensitivity as the current clinical practice of applying the FIT test at threshold ≥ 10 µg/g, in the Oxford FIT dataset. This implies that the models were not able to simultaneously capture the same number of cancers as the FIT test and reduce the number of patients referred to subsequent investigations.	Python code for evaluating FIT-based risk prediction models will be freely accessible at `https://github.com/tammandres/fitval` after a paper has been published. The code computes a comprehensive set of performance metrics, including metrics that directly evaluate whether the model outperforms the FIT test for reducing the number of referrals.
5	Is it possible to find a machine learning model—from a set of models with varying degrees of interpretability and flexibility—that would outperform the FIT test for colorectal cancer detection on routinely collected Oxford FIT data? 'Outperforming' has the same meaning as in the third RQ.	Up to 582 predictor variables were included, machine learning (ML) models with varying degrees of interpretability and flexibility were applied, and novel ways of fitting the ML models that maximise areas under the ROC and precision-recall curves were employed. Five-fold cross-validated results showed that none of the ML models, predictor variable combinations, or model training regimes outperformed the FIT test, in terms of having a higher PPV than FIT ≥ 10 µg/g.	Python code for fitting and evaluating an array of machine learning models relative to the FIT test with k-fold 'train-validation-test' cross-validation will be accessible at `https://github.com/tammandres/fitml` after a paper has been published.
6	Is it possible to develop a lightweight software program that can automatically group patients with similar medical event sequences, and display the grouped sequences, so that a user can have a quick overview of how the different medical events follow each other over time and how different patterns of medical events may be associated with different clinical outcomes? 'Lightweight' has the same meaning as in the second RQ.	Data from the NIHR HIC colorectal cancer database was used. Treatment patterns of rectal cancer patients (different sequences of surgeries and chemo-radiotherapy) were clustered and explored to showcase the method. It was seen that different treatment patterns grouped at face value, that some hospitals were represented more in some clusters, and that some clusters were associated with worse survival outcomes. However, it was not possible to yet draw clinically useful inferences from this exploratory analysis. It would be more fruitful to apply the clustering method to a more narrowly defined patient population with similar disease severity, in which case observed differences in treatment patterns between hospitals are more likely to reflect variations in clinical practice, and observed differences in outcomes between the clusters are more likely to be due to treatment pattern differences (even though causality cannot be proven). The clustering method and the interactive app also worked relatively fast and should be easily deployable to other hospital systems to facilitate the exploration of EHRs. The strength of the clustering method is also that it takes the relative time between events into account.	Python code for clustering and exploring medical event sequences of patients can be accessed at `https://github.com/tammandres/event-sequence-explorer` after a paper has been published.

7.2 Contributions beyond research findings

Additional contributions beyond research findings are considered here. The result obtained in Chapter 4—that the externally derived COLOFIT risk prediction model did not outperform the FIT test alone in Oxford data—is likely to encourage healthy caution in the implementation of the model (it has influenced discussions of the COLOFIT team with NHS England), and motivate further studies into what factors determine model performance. The machine learning analysis of FIT-based prediction models conducted in Chapter 5 attracted interest from a representative of the NHS Transformation Directorate, who requested it as a case study. Both the COLOFIT validation analysis and the machine learning analysis may also influence future NICE guidance on applying FIT-based models in primary care—if such guidance will be created or incorporated into existing FIT guidance—as NICE has been reviewing the performance of FIT-based models in a recent report (although before my work was conducted) [212]. The analyses reported in this dissertation will also likely lead to research collaborations: for example, a researcher from University College London recently expressed interest in the data visualisation tool described in Chapter 6. The software codes for extracting TNM staging from free text and validating FIT-based colorectal cancer risk prediction models are straightforward to use and will be helpful for future analyses of Oxford FIT data, but could also be employed by other researchers.

Are there any learnings for future digital health infrastructures? When working with EHR data, I noticed there was a lack of well-documented metadata to describe how each data item was collected and what it meant, which can hinder the use of data for service evaluation and improvement. Even though the whole health data life cycle is important (Chapter 2), metadata is a concrete aspect that should be straightforward to improve by retrospectively contacting individuals who enter the data to obtain more detail and by attempting to incorporate metadata entry into the data entry process at source. However, high-quality data on its own will not be useful unless there are motivated people to act on it. Even though data and modelling hold great potential for improving the NHS, there are other

aspects such as the working culture and well-being of all healthcare workers that I assume are at least equally important for a well-functioning service and should not be overlooked in the digital age.

7.3 A personal summary of learnings

7.3.1 Data processing

- Spend more time with clinicians to better understand how the key data items in your dataset were originally generated, and what the missing values mean.
- Be careful with dates. The standard *pandas* python library for processing tables can inconsistently convert date strings to date format (i.e. day-month and month-day). It is better to first check that all dates in a column comply with a specific format (e.g. dd/mm/yyyy), and then convert datestrings to dates supplying that format without relying on automatic inference rules.
- Examine time series of included variables to spot trends and/or abrupt changes[26]. Sometimes, a trend is harder to see when plotting raw values, but is visible when plotting a transformation – for example, plotting the percentage of patients that had a positive test result within each time period.
- Plot random samples of patient timelines that contain events of interests (such as treatments or blood tests or outcomes that are being studied). This can be used to spot data quality issues and to better understand the data.

7.3.2 Machine learning models

- A new fancy machine learning method is unlikely to greatly outperform a strong established method, such as *xgboost*[168], unless the context of the problem strongly suggests otherwise. Having a sufficiently large, diverse and high-quality dataset is probably more important than having a fancy model.
- Resist the temptation to fine-tune regex-based information extraction algorithms. Adding a new rule can increase sensitivity, but decrease precision

in unexpected ways. A better use of time can be to train an ML model on labelled examples. The model can then be iteratively updated using active learning, by feeding it more examples of data that it initially got wrong.

- A machine learning method will probably not work the first time you use it, and if it does it will probably not work the way you thought it would. A significant amount of fine-tuning, experimentation, and understanding of the data is required.

- Models can fail 'silently'[184]. Run tests to ensure that the model works as you expected. This can include creating dummy data with specific patterns and seeing that the model is able to detect these patterns. Similarly, it can be helpful to pass a small sample of data through your neural network, and print the shape and values of the output at each stage. It can be easier to debug errors if this is first done on a CPU rather than a GPU.

- Always set aside a held-out test set before touching the data.

7.3.3 Mental patterns

- The avenues to improve your work are endless. Try to resist the temptation to improve and focus on rigorously completing a smaller chunk of the work, no matter how incomplete it may seem.

- Writing up your research results as a chapter or a paper gives new and more thorough perspectives. This is probably because writing makes you think about the work as a whole, and in attempting to precisely and concisely describe your findings, it also makes you understand the work better.

- Multiply the time you expect to write something by two or three to get a more truthful estimate.

7.4 Broad future directions

7.4.1 Optimising the use of FIT test

There is work ongoing to produce a database of electronic patient records across the Thames Valley. This could be used to explore comorbidities and other characteristics of patients who are offered the FIT test, which may determine whether a colorectal cancer risk prediction model that combines routinely collected data with FIT is successful in reducing the number of false positives compared to FIT alone. The work done in Oxfordshire can also be expanded by collaborating with the team of Dr David Humes in Nottingham to better understand what makes FIT-based risk prediction models work. In addition, it would be interesting to study if FIT can be combined with a cell-free DNA test, which could result in a more generalisable way to improve on the FIT test than models that use routine data.

7.4.2 Blood test trends for cancer prediction

Virdee and colleagues have set out a vision to study blood test trends in primary care data [213]. In particular, the skills I have developed for interactively visualising EHR data could be used to better understand which blood test trends may exist. There is a saying in the ML community that it is always good to 'look' at the data before using it, and I may able to produce interactive visual summaries of random samples of patient timelines to help with this work. In addition to more classical methods that are deployed in this project, I would also be curious to explore transformer-based time series models, especially as the size of the dataset may afford these.

7.4.3 Understanding cancer treatment patterns

The work I have done for clustering patients based on their treatment patterns can be extended to larger regional and/or national datasets, to better explore variations in treatment and outcomes after treatment. For example, if it is possible to collate a much larger regional CRC dataset (as mentioned above), then it may be possible to extract a specific subpopulation of patients with initial disease severity at diagnosis, which was currently not feasible due to sample size and incomplete

records. For example, it may be possible to extract data for CRC patients who had liver metastases and similar TNM staging at diagnosis. Different treatment strategies can then be compared using a conventional statistical analysis (colorectal cancer surgery before liver surgery, liver surgery before colorectal cancer surgery, or both surgeries at the same time). ML methods can additionally be used to explore the treatment patterns in a more granular way, seeing if the patterns of two types of surgeries (and perhaps also chemo- and radiation therapy treatments) cluster into groups based on both the sequence and relative timing of events, and if some of the clusters are associated with better outcomes. This could also be explored with semi-supervised models that attempt to simultaneously cluster patients and predict treatment outcomes (as pure unsupervised clustering may not yield clinically meaningful distinctions). The topic of studying CRC patients with synchronous liver metastases was originally a research theme proposed by Dr Helen Jones and other clinicians as part of the NIHR HIC CRC collaboration), and I am confident that they would be happy to contribute to this research project if a sufficiently large dataset becomes available.

Appendices

A

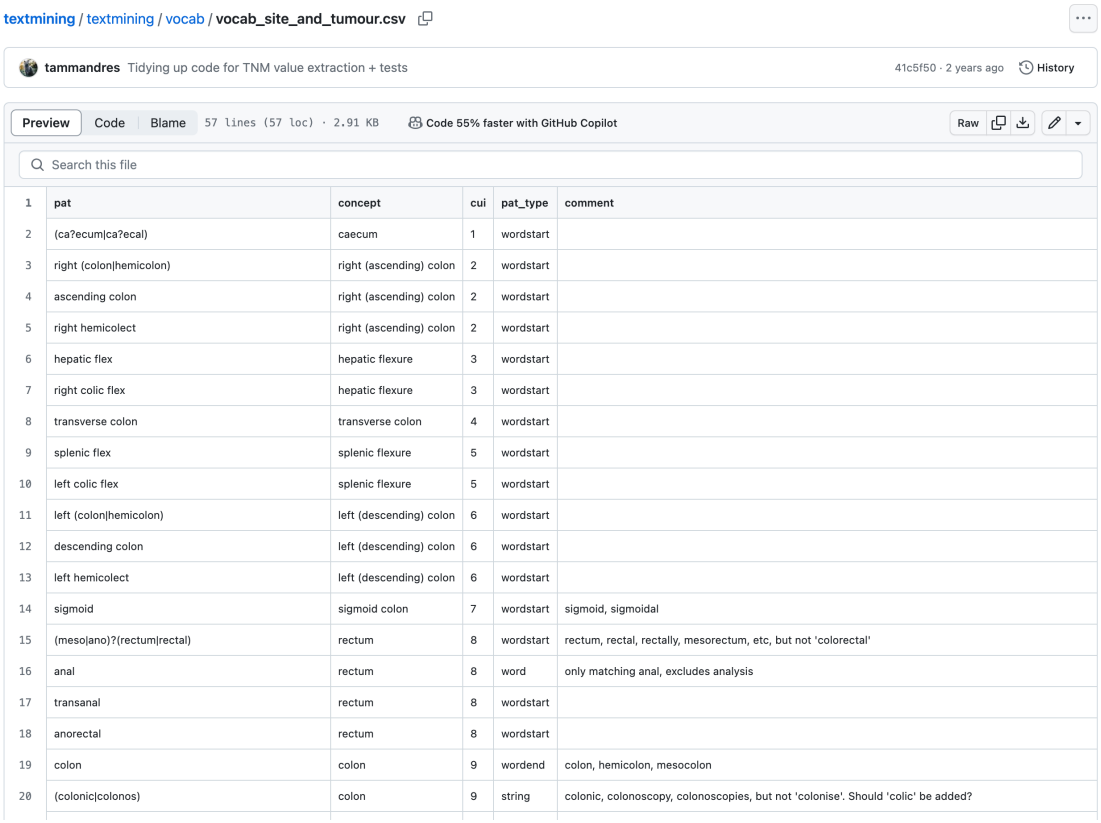
Additional results for extracting information about colorectal cancer from imaging and histopathology reports

Contents

This appendix contains supplementary results for Chapter 3.

A.1 Patterns for tumour keywords and sites

Figure A.1 illustrates the first the 20 patterns (out of 57) used for detecting tumour keywords and anatomical sites.



textmining / textmining / vocab / **vocab_site_and_tumour.csv**

tammandres Tidying up code for TNM value extraction + tests — 41c5f50 · 2 years ago — History

Preview | Code | Blame — 57 lines (57 loc) · 2.91 KB — Code 55% faster with GitHub Copilot — Raw

Search this file

	pat	concept	cui	pat_type	comment
2	(ca?ecum|ca?ecal)	caecum	1	wordstart	
3	right (colon|hemicolon)	right (ascending) colon	2	wordstart	
4	ascending colon	right (ascending) colon	2	wordstart	
5	right hemicolect	right (ascending) colon	2	wordstart	
6	hepatic flex	hepatic flexure	3	wordstart	
7	right colic flex	hepatic flexure	3	wordstart	
8	transverse colon	transverse colon	4	wordstart	
9	splenic flex	splenic flexure	5	wordstart	
10	left colic flex	splenic flexure	5	wordstart	
11	left (colon|hemicolon)	left (descending) colon	6	wordstart	
12	descending colon	left (descending) colon	6	wordstart	
13	left hemicolect	left (descending) colon	6	wordstart	
14	sigmoid	sigmoid colon	7	wordstart	sigmoid, sigmoidal
15	(meso|ano)?(rectum|rectal)	rectum	8	wordstart	rectum, rectal, rectally, mesorectum, etc, but not 'colorectal'
16	anal	rectum	8	word	only matching anal, excludes analysis
17	transanal	rectum	8	wordstart	
18	anorectal	rectum	8	wordstart	
19	colon	colon	9	wordend	colon, hemicolon, mesocolon
20	(colonic|colonos)	colon	9	string	colonic, colonoscopy, colonoscopies, but not 'colonise'. Should 'colic' be added?

Figure A.1: A subset of patterns for detecting tumour keywords and anatomical sites. This screenshot illustrates the first 20 patterns out of 57 that were used to detect concepts related to tumours and anatomical sites. It is an older version: the current version contains a few modifications, such as distinguishing mesorectum as a separate anatomical site from rectum. Each pattern is given in the 'pat' column, as a string or a regular expression. Values in the 'pat_type' column are used to additionally refine the pattern. For example, if 'pat_type' is 'wordstart', then 'pat' is extended such that variable number of word characters can follow. If 'pat' is 'word', it is bracketed by nonword characters.

A.2 Patterns for detecting the context of tumour keywords

Figure A.2 illustrates the first the 20 patterns (out of 119) used for detecting the affirmation status of tumour keywords – whether each keyword was negated, discussed in a general sense, discussed in a historical sense, etc. Note that additional qualifiers were applied to the patterns (such as how far they can occur from the tumour keyword, and which other keywords terminate their scope) – the qualifiers are not currently displayed.

tammandres Tidying up code for TNM value extraction + tests — 41c5f50 · 2 years ago — History

textmining / textmining / vocab / **context.csv** — Top

Preview | Code | Blame | 119 lines (119 loc) · 9.92 KB | Code 55% faster with GitHub Copilot | Raw

	category	pat	pat_type
2	negated	: (no|none|negative)	word
3	negated	(absent|excluded|free|negative|resolved|ruled out)	word
4	negated	not (demonstrated|identified|indicated|known|present|seen|significant|suggested)	word
5	negated	cannot be seen	word
6	negated	(clear|free) (of|from)	word
7	negated	(nil|no|not|without)	word
8	negated	no (evidence|features|indication|sign)	wordstart
9	negated	not (contain|indicat|represent|show|suggest)	wordstart
10	negated	cannot see	word
11	negated	absence of	word
12	negated	(is|are) negative for	word
13	negated	preferred over	word
14	historic	(histor|previous|predates|prior|known|recent|has had)	wordstart
15	historic	clinical (details|history|information)	wordstart
16	historic	(19\d\d|20\d\d)	word
17	historic	(jan|feb|mar|apr|in may|may 19\d\d|may 20\d\d|jun|jul|aug|sep|oct|nov|dec)	word
18	historic	(january|february|march|april|in may|may 19\d\d|may 20\d\d|june|july|august|september|october|november|december)	word
19	historic	(19\d\d|20\d\d)	word
20	historic	(jan|feb|mar|apr|may 19\d\d|in may|may 20\d\d|jun|jul|aug|sep|oct|nov|dec)	word

Figure A.2: A subset of patterns for detecting the context of tumour keywords. First 20 patterns ('pat') that were used for detecting the context are shown. Note that additional qualifiers were applied to the patterns (such as how far they can occur from the tumour keyword, and which other keywords terminate their scope) – these are not currently displayed.

A.3 Performance of the TNM stage extraction algorithm for detecting the main values within each TNM category

Each TNM category can be reported using multiple values (Table 3.4). For example, T category can take values 0, 1, 1a-1d, 2, 2a-2d, 3, 3a-3d, 4, 4a-4d, X, is. Performance of the TNM stage extraction algorithm for detecting each main value of each TNM category is reported in Table A.1. (The main values for T category are 0, 1, 2, 3, 4, X, is; ignoring subcategories a-d and not considering the misclassification of these as an error.) Some TNM values occured rarely, so their performance estimates are uncertain. However, it can be seen that almost all values were detected with high precision and sensitivity. The reduced sensitivities for T0 and M0 were due to clinical reports not reporting the TNM stage in letters and numbers, but where this TNM value could have been inferred from text.

Table A.1: Performance of the TNM stage algorithm for detecting the main values within each TNM category in clinical reports of OUH colorectal cancer patients

TNM cat.	Value	N_{report}	N_{value}	$\text{PPV}_{\text{micro}}$	NPV	$\text{Sensitivity}_{\text{micro}}$	Specificity
Pathology reports - training data							
Tpre	p	200	92	100.0 (96.0, 100.0)	100.0 (96.6, 100.0)	100.0 (96.0, 100.0)	100.0 (96.6, 100.0)
Tpre	yp	200	7	100.0 (64.6, 100.0)	100.0 (98.0, 100.0)	100.0 (64.6, 100.0)	100.0 (98.0, 100.0)
T	0	200	13	100.0 (43.9, 100.0)	94.9 (90.9, 97.2)	23.1 (8.2, 50.3)	100.0 (98.0, 100.0)
T	1	200	17	100.0 (81.6, 100.0)	100.0 (97.9, 100.0)	100.0 (81.6, 100.0)	100.0 (97.9, 100.0)
T	2	200	18	100.0 (82.4, 100.0)	100.0 (97.9, 100.0)	100.0 (82.4, 100.0)	100.0 (97.9, 100.0)
T	3	200	42	100.0 (91.6, 100.0)	100.0 (97.9, 100.0)	100.0 (91.6, 100.0)	100.0 (97.6, 100.0)
T	4	200	20	100.0 (83.9, 100.0)	100.0 (97.9, 100.0)	100.0 (83.9, 100.0)	100.0 (97.9, 100.0)
N	0	200	54	100.0 (93.4, 100.0)	100.0 (97.4, 100.0)	100.0 (93.4, 100.0)	100.0 (97.4, 100.0)
N	1	200	23	100.0 (85.7, 100.0)	100.0 (97.9, 100.0)	100.0 (85.7, 100.0)	100.0 (97.9, 100.0)
N	2	200	8	100.0 (67.6, 100.0)	100.0 (98.0, 100.0)	100.0 (67.6, 100.0)	100.0 (98.0, 100.0)
N	x	200	3	100.0 (43.9, 100.0)	100.0 (98.1, 100.0)	100.0 (43.9, 100.0)	100.0 (98.1, 100.0)
M	0	200	29	100.0 (88.3, 100.0)	100.0 (97.8, 100.0)	100.0 (88.3, 100.0)	100.0 (97.8, 100.0)
M	1	200	3	100.0 (43.9, 100.0)	100.0 (98.1, 100.0)	100.0 (43.9, 100.0)	100.0 (98.1, 100.0)
M	x	200	40	100.0 (91.2, 100.0)	100.0 (97.7, 100.0)	100.0 (91.2, 100.0)	100.0 (97.7, 100.0)
V	0	200	61	100.0 (94.1, 100.0)	100.0 (97.3, 100.0)	100.0 (94.1, 100.0)	100.0 (97.3, 100.0)
V	1	200	30	100.0 (88.6, 100.0)	100.0 (97.8, 100.0)	100.0 (88.6, 100.0)	100.0 (97.8, 100.0)
R	0	200	83	98.8 (93.6, 99.8)	100.0 (96.8, 100.0)	98.9 (95.6, 99.8)	99.1 (95.3, 100.0)
R	1	200	7	100.0 (64.6, 99.8)	100.0 (98.0, 100.0)	98.9 (64.6, 99.8)	100.0 (98.0, 100.0)
R	2	200	1	-	99.5 (97.2, 99.9)	0.0 (0.0, 79.3)	100.0 (98.1, 100.0)
L	0	200	55	100.0 (93.5, 100.0)	100.0 (97.4, 100.0)	100.0 (93.5, 100.0)	100.0 (97.4, 100.0)
L	1	200	36	100.0 (90.4, 100.0)	100.0 (97.7, 100.0)	100.0 (90.4, 100.0)	100.0 (97.7, 100.0)
Pn	0	200	54	100.0 (93.2, 100.0)	99.3 (96.2, 100.0)	98.1 (90.2, 99.3)	100.0 (97.4, 100.0)
Pn	1	200	20	100.0 (83.2, 100.0)	99.4 (96.9, 100.0)	95.0 (76.4, 99.3)	100.0 (97.9, 100.0)
Kikuchi	2	2	8	100.0 (34.2, 100.0)	100.0 (98.1, 100.0)	100.0 (34.2, 100.0)	100.0 (98.1, 100.0)
Kikuchi	3	6	8	100.0 (61.0, 100.0)	100.0 (98.1, 100.0)	100.0 (61.0, 100.0)	100.0 (98.1, 100.0)
G	1	200	1	100.0 (20.7, 100.0)	100.0 (98.1, 100.0)	100.0 (20.7, 100.0)	100.0 (98.1, 100.0)
G	3	200	1	100.0 (20.7, 100.0)	100.0 (98.1, 100.0)	100.0 (20.7, 100.0)	100.0 (98.1, 100.0)
Imaging reports - training data							

Continued on next page

Table A.1 – continued from previous page

TNM cat.	Value	N_{report}	N_{value}	$\text{PPV}_{\text{micro}}$	NPV	$\text{Sensitivity}_{\text{micro}}$	Specificity
Tpre	ym	200	4	100.0 (51.0, 100.0)	100.0 (98.1, 100.0)	100.0 (51.0, 100.0)	100.0 (98.1, 100.0)
T	0	200	5	100.0 (20.7, 100.0)	98.0 (94.9, 99.2)	20.0 (3.6, 62.4)	100.0 (98.1, 100.0)
T	1	200	4	100.0 (51.0, 100.0)	100.0 (98.1, 100.0)	100.0 (51.0, 100.0)	100.0 (98.1, 100.0)
T	2	200	22	95.7 (79.0, 99.2)	100.0 (97.9, 100.0)	100.0 (85.1, 100.0)	100.0 (96.9, 100.0)
T	3	200	44	100.0 (92.0, 100.0)	100.0 (97.6, 100.0)	100.0 (92.0, 100.0)	100.0 (97.6, 100.0)
T	4	200	26	96.2 (81.1, 99.3)	99.4 (96.8, 99.9)	96.2 (81.1, 99.3)	100.0 (96.8, 100.0)
T	x	200	2	100.0 (34.2, 100.0)	100.0 (98.1, 100.0)	100.0 (34.2, 100.0)	100.0 (98.1, 100.0)
N	0	200	50	97.9 (89.1, 99.6)	98.0 (94.4, 99.3)	94.0 (83.8, 97.9)	99.3 (96.3, 99.9)
N	1	200	31	100.0 (88.6, 100.0)	99.4 (96.7, 99.9)	96.8 (83.8, 99.4)	100.0 (97.8, 100.0)
N	2	200	9	90.0 (59.6, 98.2)	100.0 (98.0, 100.0)	100.0 (70.1, 100.0)	99.5 (97.1, 99.9)
M	0	200	27	100.0 (87.1, 100.0)	99.4 (96.8, 99.9)	96.3 (81.7, 99.3)	100.0 (97.8, 100.0)
M	1	200	15	100.0 (70.1, 100.0)	96.9 (93.3, 98.6)	60.0 (35.7, 80.2)	100.0 (98.0, 100.0)
M	x	200	1	100.0 (20.7, 100.0)	100.0 (98.1, 100.0)	100.0 (20.7, 100.0)	100.0 (98.1, 100.0)
V	0	200	19	100.0 (83.2, 100.0)	100.0 (97.9, 100.0)	100.0 (83.2, 100.0)	100.0 (97.9, 100.0)
V	1	200	8	100.0 (67.6, 100.0)	100.0 (98.0, 100.0)	100.0 (67.6, 100.0)	100.0 (98.0, 100.0)

Notes. N_{value} is the number of reports that contains a value. PPV - positive predictive value, NPV - negative predictive value. 95% Wilson confidence intervals are shown in brackets.

B

Additional results for the external validation of Nottingham colorectal cancer risk prediction models

Contents

This appendix contains supplementary results for Chapter 4.

B.1 Predictor-outcome relationships encapsulated in Nottingham colorectal cancer risk prediction models

The Nottingham models are additive models that contain power and log transformations for FIT, age, and platelets, and no transformations for sex and mean cell volume. To better understand the effect of each variable, its contribution to the linear predictor can be visualised. This is shown in the bottom panel of Figure B.1. The relationships between these variables and the risk of cancer were also estimated in Oxford data using binning and LOWESS-smoothing (top panel, Figure B.1). Overall, the relationships encapsulated in Nottingham models resemble the relationships observed in Oxford data (although the top and bottom panel are not on the same scale, as the bottom panel shows contribution to the linear predictor while the top panel shows probability of cancer). Note that the shapes of the relationships learned by the different Nottingham models are very similar, but the curves are shifted relative to each other on the y-axis – this does not mean that the models lead to different risk predictions, because they also differ in their intercepts. Indeed, all Nottingham models performed very similarly (Chapter 4).

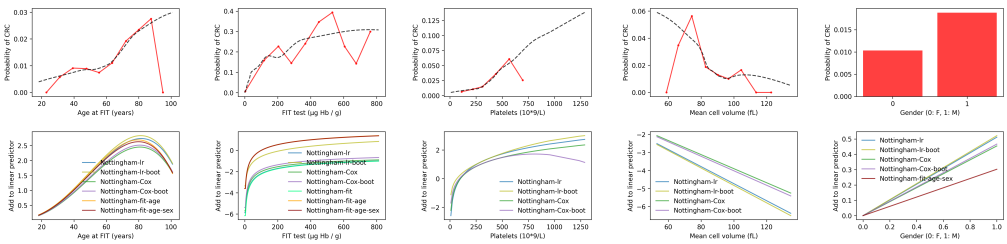


Figure B.1: Predictor-outcome relationships encapsulated in Nottingham models and in the Oxford dataset. Top row: relationships between the value of each variable and the risk of cancer in Oxford data, estimated using binning (red) or LOWESS-smoothing (dashed black line). Bottom row: Contribution of each variable to the linear predictor of Nottingham models.

B.2 Predicted probabilities of cancer according to logistic and Cox models

The Nottingham logistic and Cox models performed very similarly in terms of discrimination, calibration and net benefit, and the relationships between input variables and cancer were also similar (Figure B.1). Consistent with this, the Nottingham logistic and Cox models predicted highly similar probabilities of colorectal cancer for all patients (Figure B.2).

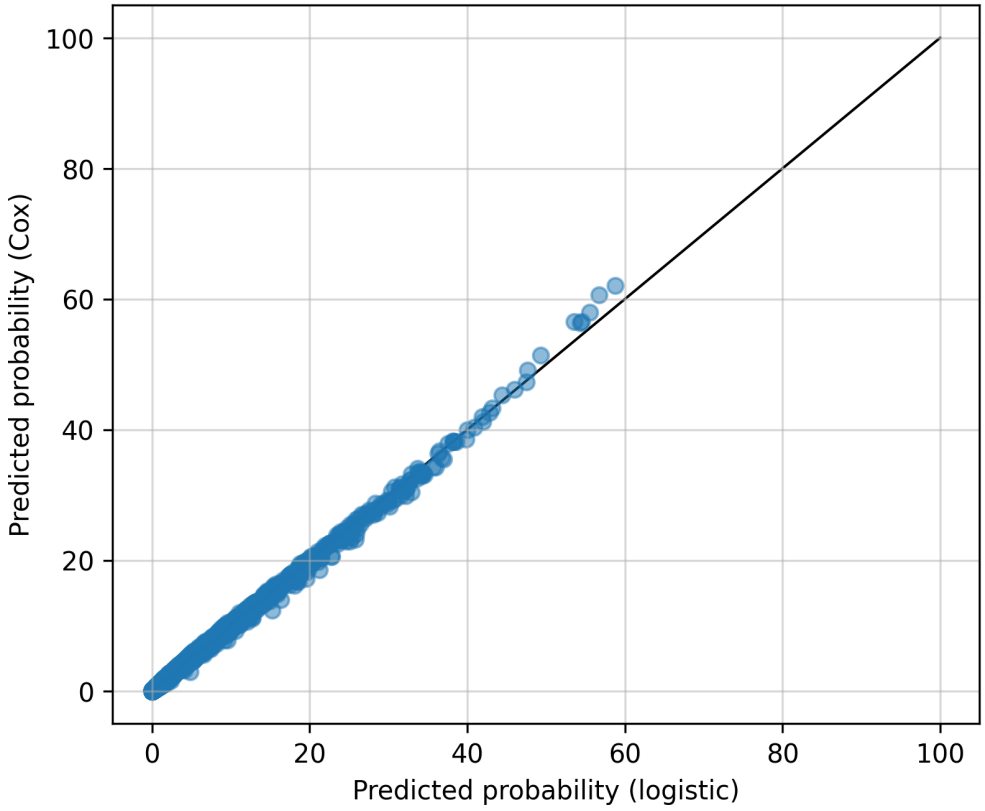


Figure B.2: Predicted probabilities of colorectal cancer according to Nottingham logistic regression and Cox proportional hazard models for each patient

B.3 Relationship between FIT values and risk of cancer in Oxford data

To estimate the risk (probability) of colorectal cancer for patients at each FIT value, the FIT values were transformed using splines and a logistic model was used – the orange line labelled as 'logistic on spline(log(FIT+1))' in Figure B.3. This showed that at FIT value 10, the risk was approximately 2.5%. However, this estimate is likely to be imprecise, as there are not many cancers with FIT values close to 10.

Other methods were also used to explore the relationship between FIT and risk of cancer: dividing FIT values into bins and computing the proportion of colorectal cancers in each bin; monotonic regression; and logistic regression on raw FIT values or log-transformed FIT values (Figure A3). Logistic regression on raw FIT values was unsuitable, as it led to too low predicted risks for lower FIT values and too high predicted risks for higher FIT values compared to LOWESS, monotonic and logistic spline regression. Logistic regression applied to logarithm of FIT values (after adding a constant of 1) yielded better but still less accurate results especially for FIT values less than 20. The model that applied a spline transformation on logarithm of FIT values performed similarly to the nonparametric LOWESS model.

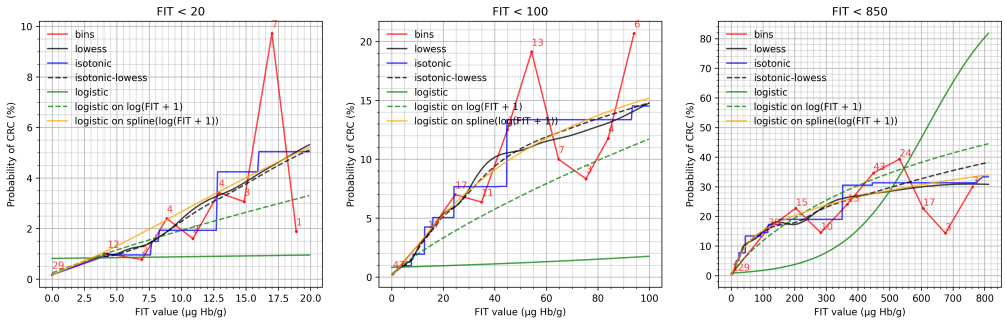


Figure B.3: Relationship between FIT values and probability of colorectal cancer in the Oxford FIT dataset. Red: curve based on dividing FIT values into equal-width bins and computing the proportion of cancers in each bin. Black: curve based on LOWESS-smoothing. Blue: relationship between FIT value and cancer according to isotonic (monotonic) regression. Black-dashed: LOWESS-smoothing applied to the isotonic regression curve. Green: relationship between FIT and cancer according to logistic regression model. Green-dashed: relationship between FIT and cancer according to logistic regression applied to log(FIT + 1). FIT + 1 was used in the logarithm, because some FIT values were zero. Orange: relationship between FIT and cancer according to a logistic model applied to spline(log(FIT + 1)).

B.4 Distribution of FIT values in the Oxford and Nottingham datasets

The FIT values tended to be lower in Oxford (Figure B.4). For example, the 75th percentile of FIT values was 8 in Nottingham and 0.8 in Oxford (Table B.1). Note that these figures and tables are given for the entire Oxford dataset (not for complete case analysis), and limits of detection (LoD) and quantification (LoQ) were applied to Oxford values before entering the data to models such that values less than 2 (LoD) were replaced with 0, and values between 2 and 4 (LoQ) were replaced with 4.

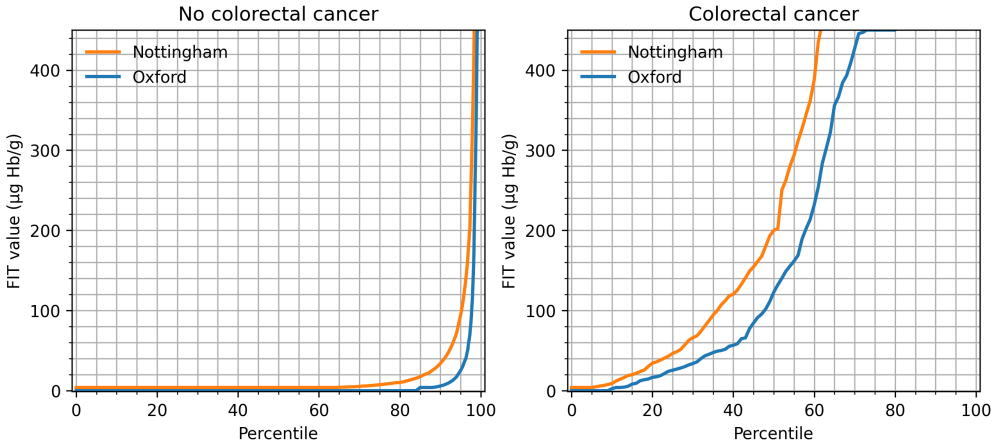


Figure B.4: Distributions of FIT values in Oxford and Nottingham for patients with and without colorectal cancer. The y-axis is capped at 450 for clarity, but the maximum FIT value in Nottingham was near 70,000 µg Hb/g, and in Oxford was near 800 µg Hb/g.

Table B.1: Selected percentiles of FIT values in Nottingham and Oxford

	Nottingham			Oxford		
Percentile	All	No CRC	CRC	All	No CRC	CRC
0	4.0	4.0	4.0	0.0	0.0	0.0
25	4.0	4.0	46.5	0.0	0.0	25.6
50	4.0	4.0	200.0	0.0	0.0	122.9
75	8.0	7.4	1146.4	0.0	0.0	450.0
80	11.2	10.2	1685.4	0.0	0.0	450.0
85	20.0	17.4	2467.0	4.0	4.0	513.8
90	40.2	34.0	4640.4	7.4	5.8	550.0
95	120.1	94.0	8303.4	35.5	25.2	600.6
99	1479.8	940.3	18655.2	450.0	401.6	670.6
100	≈70000	50000	≈70000	≈800	≈800	≈800

B.5 Summary of the OUH-FIT dataset including missing values

Table B.2: Summary of the OUH-FIT dataset with missing values included

	No colorectal cancer	Colorectal cancer (CRC)
Number of patients	20340	287
Age		
18-39.9	1762 (8.7%)	13 (4.5%)
40-49.9	2714 (13.3%)	23 (8.0%)
50-59.9	5004 (24.6%)	39 (13.6%)
60-69.9	3583 (17.6%)	46 (16.0%)
70-79.9	4156 (20.4%)	80 (27.9%)
≥80	3121 (15.3%)	86 (30.0%)
Median (25th, 75th percentile)	62.0 (51.3, 75.2)	73.6 (59.5, 81.4)
Min and max	18.2, 100.9	31.2, 92.1
Gender		
F	11797 (58.0%)	123 (42.9%)
M	8543 (42.0%)	164 (57.1%)
Ethnicity		
Asian	459 (2.3%)	2 (0.7%)
Black	153 (0.8%)	2 (0.7%)
Mixed	128 (0.6%)	-
Other Ethnic Groups	172 (0.8%)	1 (0.3%)
White	15079 (74.1%)	211 (73.5%)
Not stated	3875 (19.1%)	63 (22.0%)
Not known	474 (2.3%)	8 (2.8%)
Multiple deprivation index		
Median (25th, 75th percentile)	8.0 (7.0, 10.0)	8.0 (7.0, 10.0)
Min, max	1.0, 10.0	1.0, 10.0
Not known	1791 (8.8%)	12 (4.2%)
FIT (µg Hb/g)		
0-1.9	17143 (84.3%)	29 (10.1%)
2-9.9	1603 (7.9%)	18 (6.3%)
10-99.9	1131 (5.6%)	91 (31.7%)
≥100	463 (2.3%)	149 (51.9%)
Median (25th, 75th percentile)	0.2 (0.0, 0.7)	122.9 (25.6, 450.0)
Min, max	0.0, 811.9	0.0, 794.4
Symptoms - GP reported		
Abdominal mass	23 (0.1%)	-
Abdominal pain	2433 (12.0%)	34 (11.8%)
Anaemia	3509 (17.3%)	45 (15.7%)
Bloating	589 (2.9%)	8 (2.8%)
Blood in stool	1959 (9.6%)	26 (9.1%)
Change in bowel habit	6900 (33.9%)	108 (37.6%)
Constipation	694 (3.4%)	8 (2.8%)
Diarrhoea	2228 (11.0%)	35 (12.2%)
Family history of CRC	208 (1.0%)	2 (0.7%)
Fatigue	240 (1.2%)	6 (2.1%)
Inflammation	226 (1.1%)	5 (1.7%)
Iron deficiency anaemia	1322 (6.5%)	13 (4.5%)
Melaena	233 (1.1%)	2 (0.7%)
Rectal pain	139 (0.7%)	1 (0.3%)
Thrombocytosis	175 (0.9%)	1 (0.3%)
Weight loss	1360 (6.7%)	23 (8.0%)
Not known	5189 (25.5%)	68 (23.7%)
T stage*		
1	-	35 (12.2%)
2	-	29 (10.1%)
3	-	84 (29.3%)
4	-	40 (13.9%)
Not known	-	99 (34.5%)
CRC-relevant treatments**		
No treatments recorded	19662 (96.7%)	69 (24.0%)
Chemotherapy	425 (2.1%)	102 (35.5%)
Radiotherapy	14 (0.1%)	11 (3.8%)
Local excision	80 (0.4%)	163 (56.8%)
Radicall resection	231 (1.1%)	16 (5.6%)

Notes. *T-stage was extracted from radiology and pathology reports using a pattern-matching algorithm. **CRC-relevant treatments are procedures used for treating colorectal cancer (CRC), but they may also be given for other conditions.

B.6 Common diagnostic metrics for original and recalibrated Nottingham models near the sensitivity of FIT test at threshold ≥10

Table B.3: Specificity, positive predictive value and negative predictive value for original and recalibrated Nottingham models and for the FIT test near the sensitivity of FIT test at threshold ≥10

Model	Specificity	PPV	NPV	Threshold (approx)
Sensitivity 90%				
FIT	76.4 (59.7, 89.1)	7.0 (3.3, 10.8)	99.8 (99.8, 99.8)	3.64
FIT-spline	76.4 (59.7, 89.1)	7.0 (3.3, 10.8)	99.8 (99.8, 99.8)	0.98
Nottingham-fit	63.0 (49.0, 89.1)	3.9 (2.7, 10.8)	99.8 (99.7, 99.8)	0.16
Nottingham-fit-platt	63.0 (49.0, 89.1)	3.9 (2.7, 10.8)	99.8 (99.7, 99.8)	0.32
Nottingham-fit-quant	76.4 (59.7, 88.6)	7.0 (3.3, 10.6)	99.8 (99.8, 99.8)	1.43
Nottingham-fit-3.5	82.1 (61.4, 89.1)	6.9 (3.6, 10.8)	99.8 (99.8, 99.8)	0.25
Nottingham-fit-age	77.6 (69.9, 86.7)	5.6 (4.2, 9.1)	99.8 (99.8, 99.8)	0.26
Nottingham-fit-age-platt	77.6 (69.9, 86.7)	5.6 (4.2, 9.1)	99.8 (99.8, 99.8)	0.55
Nottingham-fit-age-quant	82.5 (72.7, 89.4)	7.1 (4.6, 11.2)	99.8 (99.8, 99.8)	0.29
Nottingham-fit-age-3.5	83.1 (72.1, 88.5)	7.3 (4.5, 10.4)	99.8 (99.8, 99.8)	0.31
Nottingham-fit-age-sex	77.3 (64.8, 86.5)	5.5 (3.6, 8.9)	99.8 (99.8, 99.8)	0.27
Nottingham-fit-age-sex-platt	77.3 (64.8, 86.5)	5.5 (3.6, 8.9)	99.8 (99.8, 99.8)	0.56
Nottingham-fit-age-sex-quant	82.6 (76.9, 89.5)	7.1 (5.4, 11.2)	99.8 (99.8, 99.8)	0.34
Nottingham-fit-age-sex-3.5	83.9 (77.8, 88.2)	7.6 (5.7, 10.1)	99.8 (99.8, 99.8)	0.36
Nottingham-lr	82.7 (62.9, 90.0)	7.1 (3.5, 11.7)	99.8 (99.8, 99.8)	0.29
Nottingham-lr-boot	81.1 (64.2, 88.8)	6.6 (3.6, 10.6)	99.8 (99.8, 99.8)	0.48
Nottingham-cox	81.9 (65.6, 89.8)	6.8 (3.7, 11.5)	99.8 (99.8, 99.8)	0.26
Nottingham-cox-boot	81.7 (64.9, 89.7)	6.8 (3.6, 11.5)	99.8 (99.8, 99.8)	0.29
Nottingham-lr-quant	83.6 (74.6, 89.8)	7.5 (5.0, 11.5)	99.8 (99.8, 99.8)	0.54
Nottingham-lr-3.5	84.4 (74.9, 90.6)	7.8 (5.0, 12.3)	99.8 (99.8, 99.8)	0.46
Nottingham-lr-platt	82.7 (62.9, 90.0)	7.1 (3.5, 11.7)	99.8 (99.8, 99.8)	0.64
Sensitivity 83.45%**				
FIT	92.3 (87.2, 94.1)	13.7 (9.1, 17.1)	99.7 (99.7, 99.7)	10.45
FIT-spline	92.3 (87.2, 94.1)	13.7 (9.1, 17.1)	99.7 (99.7, 99.7)	2.78
Nottingham-fit	92.3 (83.3, 94.1)	13.7 (8.3, 17.1)	99.7 (99.7, 99.7)	0.66
Nottingham-fit-platt	92.3 (83.3, 94.1)	13.7 (8.3, 17.1)	99.7 (99.7, 99.7)	1.33
Nottingham-fit-quant	92.4 (87.1, 93.8)	13.4 (9.0, 16.9)	99.7 (99.7, 99.7)	4.26
Nottingham-fit-3.5	92.3 (87.7, 94.1)	13.7 (9.1, 17.1)	99.7 (99.7, 99.7)	3.07
Nottingham-fit-age	92.6 (82.1, 94.5)	14.2 (6.4, 18.3)	99.7 (99.7, 99.7)	0.71
Nottingham-fit-age-platt	92.6 (82.1, 94.5)	14.2 (6.4, 18.3)	99.7 (99.7, 99.7)	1.54
Nottingham-fit-age-quant	92.1 (88.4, 94.0)	13.4 (9.6, 17.1)	99.7 (99.7, 99.7)	3.08
Nottingham-fit-age-3.5	92.8 (87.0, 94.4)	14.6 (8.6, 18.1)	99.7 (99.7, 99.7)	2.96
Nottingham-fit-age-sex	92.4 (85.4, 94.5)	13.9 (7.8, 18.3)	99.7 (99.7, 99.7)	0.68
Nottingham-fit-age-sex-platt	92.4 (85.4, 94.5)	13.9 (7.8, 18.3)	99.7 (99.7, 99.7)	1.47
Nottingham-fit-age-sex-quant	92.5 (88.3, 93.9)	14.0 (9.5, 16.8)	99.7 (99.7, 99.7)	3.34
Nottingham-fit-age-sex-3.5	92.6 (87.5, 94.4)	14.2 (9.0, 18.1)	99.7 (99.7, 99.7)	2.78
Nottingham-lr	92.6 (88.2, 94.5)	14.3 (9.4, 18.4)	99.7 (99.7, 99.7)	0.74
Nottingham-lr-boot	91.5 (85.3, 94.0)	12.6 (7.7, 17.1)	99.7 (99.7, 99.7)	0.87
Nottingham-cox	92.5 (89.0, 94.7)	14.1 (10.1, 18.7)	99.7 (99.7, 99.7)	0.7
Nottingham-cox-boot	92.7 (88.8, 94.6)	14.5 (9.9, 18.6)	99.7 (99.7, 99.7)	0.76
Nottingham-lr-quant	91.6 (89.0, 93.8)	12.8 (10.1, 16.6)	99.7 (99.7, 99.7)	2.7
Nottingham-lr-3.5	92.2 (88.4, 94.1)	13.6 (9.6, 17.2)	99.7 (99.7, 99.7)	2.37
Nottingham-lr-platt	92.6 (88.2, 94.5)	14.3 (9.4, 18.4)	99.7 (99.7, 99.7)	1.65
Sensitivity 80%				
FIT	93.7 (91.5, 95.0)	15.8 (12.2, 19.1)	99.7 (99.7, 99.7)	16.05
FIT-spline	93.7 (91.5, 95.0)	15.8 (12.2, 19.1)	99.7 (99.7, 99.7)	4.28
Nottingham-fit	93.7 (91.5, 95.0)	15.8 (12.2, 19.1)	99.7 (99.7, 99.7)	1.18
Nottingham-fit-platt	93.7 (91.5, 95.0)	15.8 (12.2, 19.1)	99.7 (99.7, 99.7)	2.39
Nottingham-fit-quant	93.6 (91.5, 94.8)	15.4 (12.2, 18.8)	99.7 (99.7, 99.7)	5.25
Nottingham-fit-3.5	93.7 (91.5, 95.0)	15.8 (12.2, 19.1)	99.7 (99.7, 99.7)	4.67
Nottingham-fit-age	94.2 (91.5, 95.0)	16.8 (12.2, 19.2)	99.7 (99.7, 99.7)	1.27
Nottingham-fit-age-platt	94.2 (91.5, 95.0)	16.8 (12.2, 19.2)	99.7 (99.7, 99.7)	2.83
Nottingham-fit-age-quant	93.8 (91.4, 94.9)	15.9 (12.1, 18.8)	99.7 (99.7, 99.7)	4.19
Nottingham-fit-age-3.5	94.0 (91.2, 95.1)	16.5 (11.8, 19.3)	99.7 (99.7, 99.7)	4.15
Nottingham-fit-age-sex	94.2 (91.8, 94.9)	16.8 (12.6, 18.8)	99.7 (99.7, 99.7)	1.27
Nottingham-fit-age-sex-platt	94.2 (91.8, 94.9)	16.8 (12.6, 18.8)	99.7 (99.7, 99.7)	2.84
Nottingham-fit-age-sex-quant	93.4 (91.5, 94.7)	15.2 (12.2, 18.1)	99.7 (99.7, 99.7)	4.02
Nottingham-fit-age-sex-3.5	93.9 (91.7, 95.0)	16.2 (12.5, 19.0)	99.7 (99.7, 99.7)	4.04
Nottingham-lr	94.0 (91.1, 95.1)	16.4 (11.7, 19.5)	99.7 (99.7, 99.7)	1.11
Nottingham-lr-boot	93.3 (90.6, 94.9)	15.0 (11.2, 18.9)	99.7 (99.7, 99.7)	1.12
Nottingham-cox	94.3 (92.0, 95.2)	17.1 (12.8, 19.7)	99.7 (99.7, 99.7)	1.28
Nottingham-cox-boot	94.2 (91.9, 95.2)	16.8 (12.7, 19.8)	99.7 (99.7, 99.7)	1.2
Nottingham-lr-quant	92.6 (91.1, 95.1)	13.8 (11.6, 19.5)	99.7 (99.7, 99.7)	3.29
Nottingham-lr-3.5	93.6 (91.7, 95.1)	15.5 (12.5, 19.4)	99.7 (99.7, 99.7)	3.42
Nottingham-lr-platt	94.0 (91.1, 95.1)	16.4 (11.7, 19.5)	99.7 (99.7, 99.7)	2.48

Notes. The threshold for FIT test (FIT) is given in micrograms Hb / g, and for FIT-spline and Nottingham models it is given as probability of cancer in percentage (e.g. 10.43 is 10.43% probability of cancer). The threshold is marked as approximate, because specificity, PPV and NPV were interpolated to estimate these quantities at exact levels of sensitivity, and threshold was interpolated too. **The sensitivity of 83.45% is the sensitivity of FIT test at threshold ≥10.

B.7 Binned calibration curves for Nottingham models

Binned calibration curves, also known as reliability diagrams, are shown here for full Nottingham logistic and Cox models (Figure B.5, and for the full and simpler logistic models (Figure B.6).

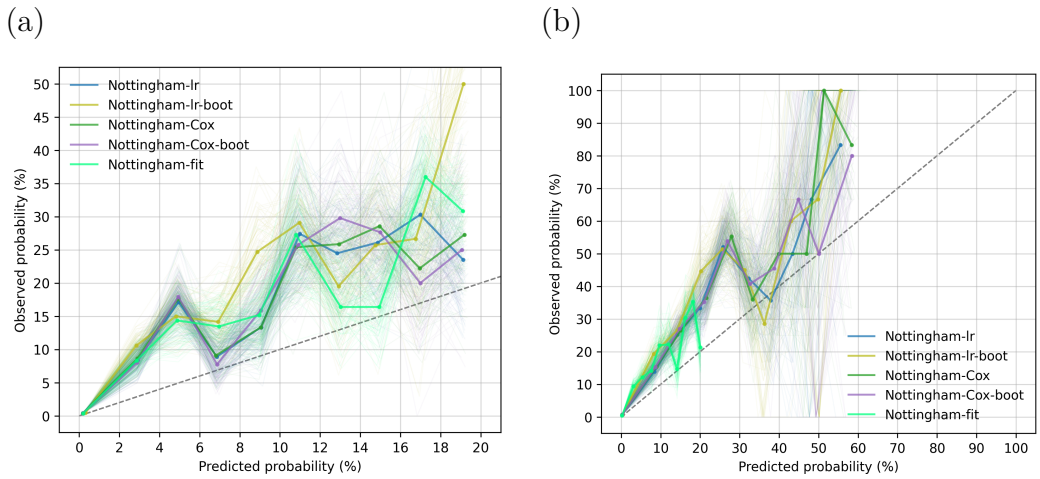


Figure B.5: Binned calibration curves (reliability diagrams) for the full Nottingham logistic and Cox models. Curves are shown for risks less than 20% (a), and over the full range of risk (b). The curves were created by dividing predicted probabilities into 10 bins, using equal-width intervals of probabilities, and computing the proportion of cancers in each bin. The light lines show curves from 100 bootstrap samples

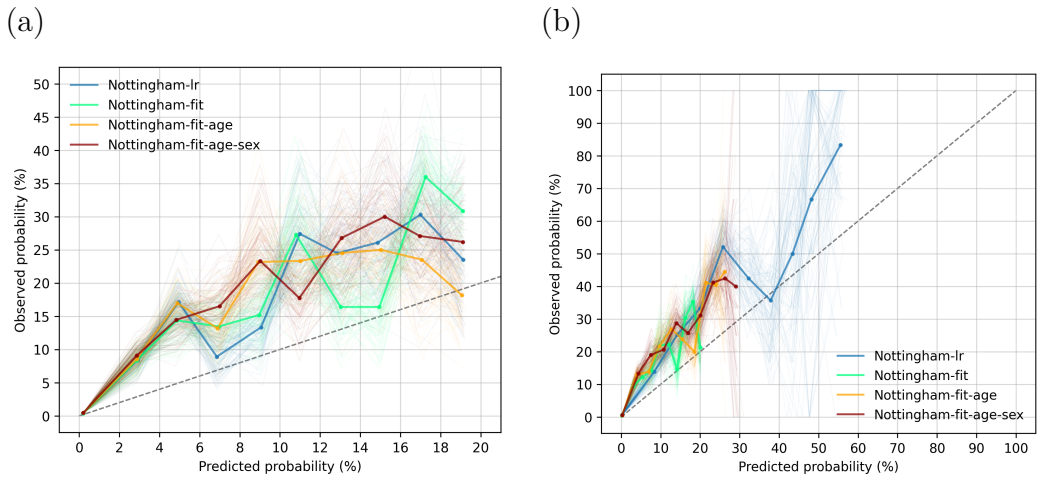


Figure B.6: Binned calibration curves (reliability diagrams) for the full Nottingham logistic model ('Nottingham-lr'), and for Nottingham FIT-only, FIT-age, and FIT-age-sex models ('Nottingham-fit', 'Nottingham-fit-age', 'Nottingham-fit-age-sex'). Curves are shown for risks less than 20% (a), and over the full range of risk (b). The curves were created by dividing predicted probabilities into 10 bins, using equal-width intervals of probabilities, and computing the proportion of cancers in each bin. The light lines show curves from 100 bootstrap samples

B.8 Performance of Nottingham models when missing values are included

Even though the MCV and platelet blood tests used in Nottingham models had missing values only for less than 5% of patients, we still performed a sensitivity analysis using multiple imputation, employing the MICE algorithm with a random forest imputation model via the python's *miceforest* package. Please see the Methods in Chapter 4 for more information about how this was combined with bootstrap for deriving confidence intervals. This appendix shows a slightly older version of the analysis where raw FIT test values were not transformed based on the limits of detection and quantification (e.g. values lower than limit of detection were replaced with 0 in the main analysis) - however, these transformations had a minimal effect on results. ROC and precision-recall curves are displayed in Figure B.7, and smooth calibration curves in Figure B.8, and these curves are similar to the ones observed in the main analysis.

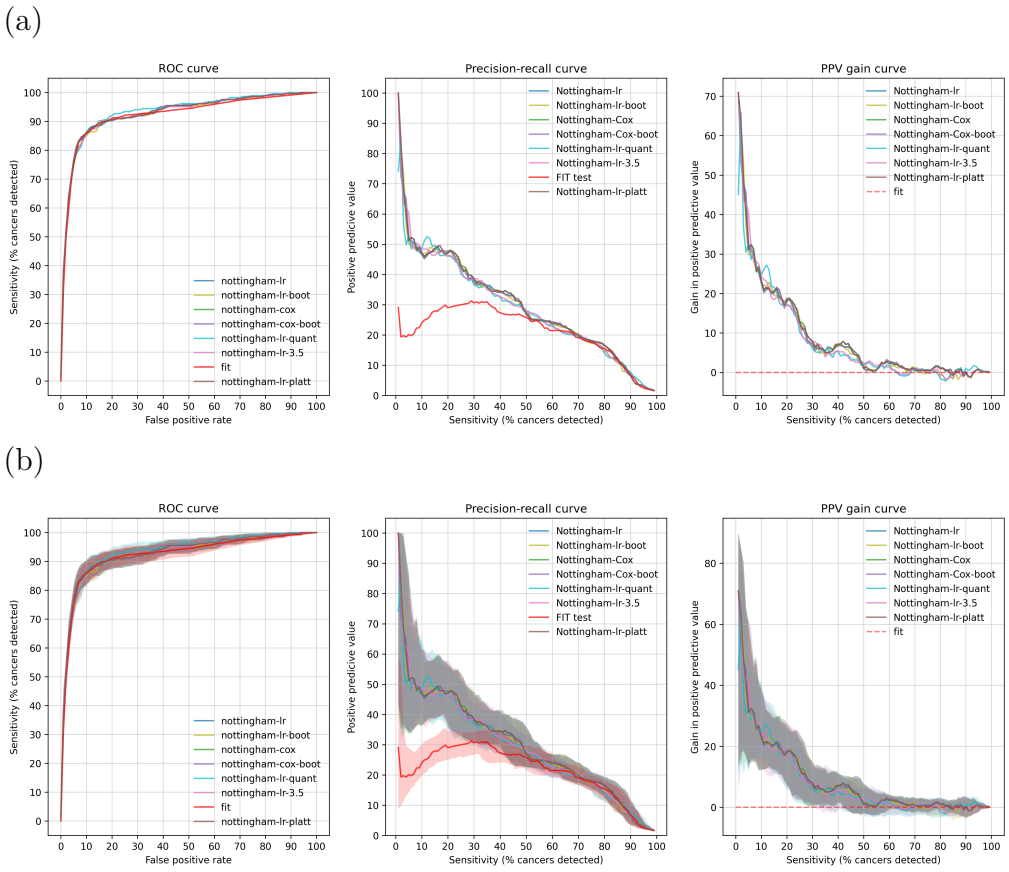


Figure B.7: ROC curve, precision-recall curve, and gain in positive predictive value relative to FIT for the full Nottingham logistic and Cox models on multiply imputed data. The full models include FIT, age, sex, mean cell volume and platelets as predictors. Curves show the full logistic model ('Nottingham-lr'), bootstrap-averaged logistic model ('Nottingham-lr-boot'), full Cox model ('Nottingham-cox'), bootstrap averaged Cox model ('Nottingham-cox-boot'); and full logistic models recalibrated with quantile transformation ('Nottingham-lr-quant'), logistic recalibration ('Nottingham-lr-platt'), and constant multiplication ('Nottingham-lr-3.5). Confidence intervals are not included in the top panel (a); bottom panel displays 95% bootstrap percentile confidence intervals (b). Curves of the FIT test and Nottingham FIT-only model are almost completely overlapping because the FIT-only model is a monotonic transformation of FIT values.

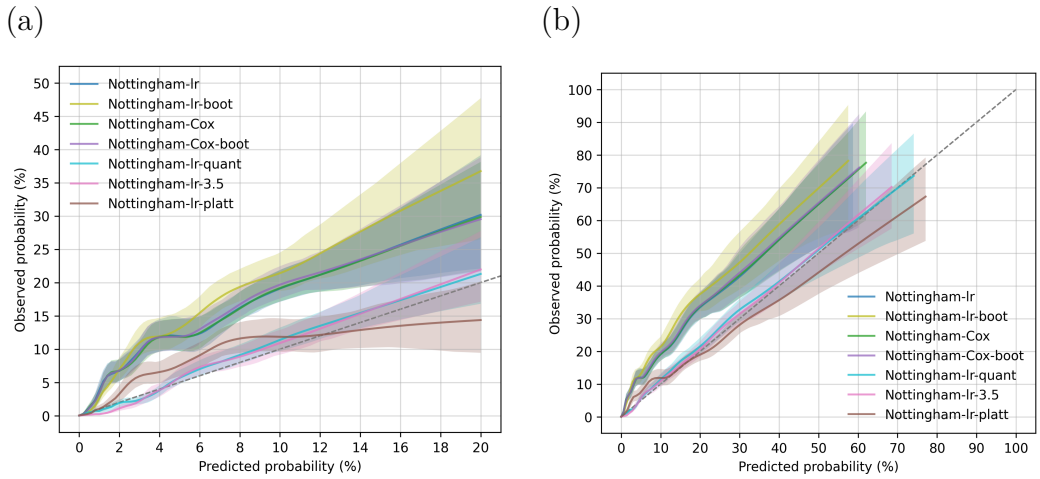


Figure B.8: Smooth calibration curves for the full Nottingham logistic and Cox models on multiply imputed data. The full models include FIT, age, sex, MCV and platelets as predictors. Curves show the full logistic model ('Nottingham-lr'), bootstrap-averaged logistic model ('Nottingham-lr-boot'), full Cox model ('Nottingham-cox'), bootstrap averaged Cox model ('Nottingham-cox-boot'); and full logistic models recalibrated with quantile transformation ('Nottingham-lr-quant'), logistic recalibration ('Nottingham-lr-platt'), and constant multiplication ('Nottingham-lr-3.5). Left panel shows calibration in the clinically meaningful range of risks (<20%, a), and right panel over the full range of risks (b). Curves were created by applying LOWESS-smoothing to predicted probabilities and cancer events. Shaded areas show 95% bootstrap percentile confidence intervals.

C

Sensitivity analyses and additional results for FIT-test based machine learning models

Contents

Here we present sensitivity analyses for FIT-test based machine learning models that were described in Chapter 5, and a few additional results. In all sensitivity analyses, we fitted a selection of both simple and flexible machine learning models, without including every model used in the main analysis to save computational resources. Where relevant, we included at least one model from the class of linear

models, general additive models, and tree ensembles.

C.1 Including the outcome variable in missing data imputation models

In the main analysis, we did not include the outcome variable in data imputation models, as many variables had a large proportion of values missing and it was not possible to check that the relationship between each predictor and outcome in observed data would carry over to missing data (which is necessary for imputations to be valid). However, not including the outcome can dilute the relationship between outcome and predictors if it exists. We therefore imputed data with outcome included and fitted the basic logistic regression, penalised logistic regression (PLR), SNAM, and GBDT models. Results were similar to the primary analysis: all models tended to have higher PPV than the FIT test at lower levels of sensitivity (less than 60%), and similar PPV at higher levels of sensitivity (Figure C.1).

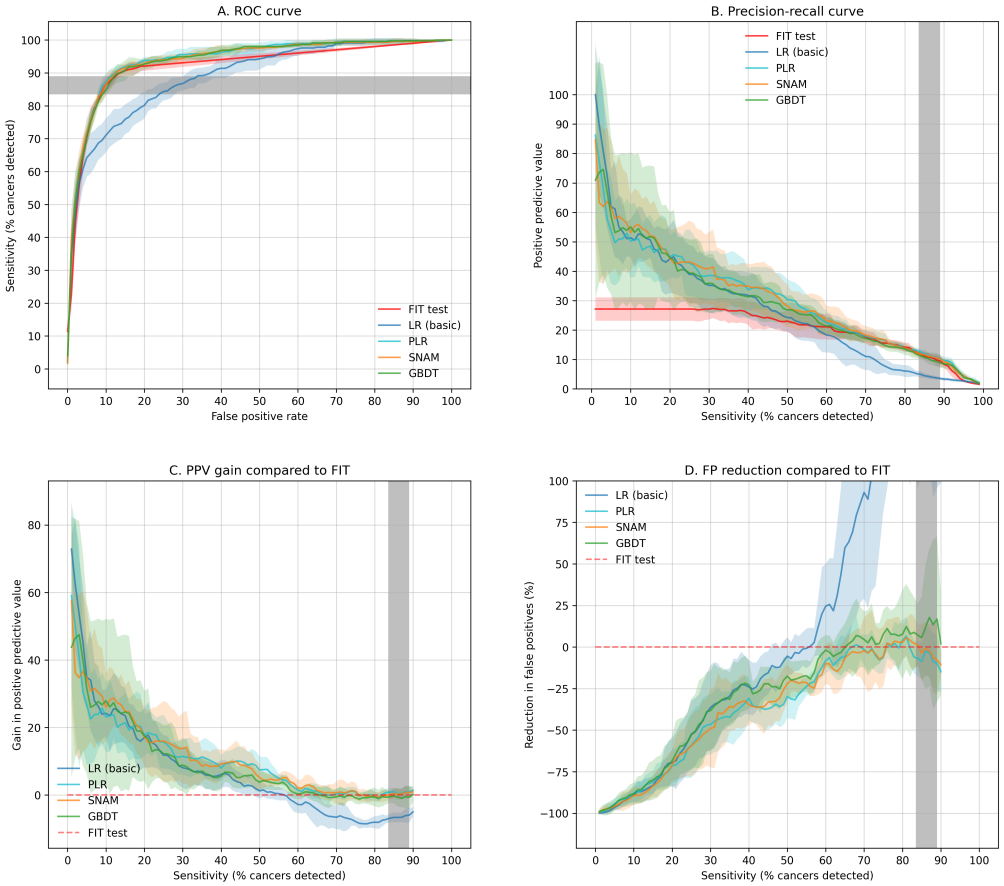


Figure C.1: Performance of machine learning models and the faecal immunochemical test (FIT) for detection of colorectal cancer, when the colorectal cancer outcome is included in missing data imputation models. Panels display the ROC-curve (A), precision-recall curve (B), gain in positive predictive value compared to the FIT test (C), and percent reduction in false positives compared to FIT (D). Curves show the mean value of each quantity over 5 cross-validation folds; shaded area covers the mean plus-minus one standard deviation. The vertical grey area is shaded between the minimum and maximum sensitivities of FIT test greater than or equal to 10 µg/g on the held-out folds. All curves were interpolated to a fixed grid of values at 1-unit increments. Data for sensitivities greater than 90% is not shown for panels C and D because the interpolation for FIT test is inaccurate in that range. LR – basic logistic regression with linear data transformation, PLR – penalised logistic regression with log or quantile data transformation, SNAM – sparse neural additive model, GBDT – gradient-boosted decision tree.

C.2 Handling missing values implicitly in the gradient-boosted decision tree

The gradient-boosted decision tree (GBDT) model, as implemented in the xgboost library, offers another way of handling missing data: default classification directions are learned for each decision tree node of each variable that has missing values, and samples with missing values are classified in the default direction. For example, if a decision tree node includes a split "if the haemoglobin blood test is less than 20, split left; otherwise split right", then missing values are always classified either to the left or right, depending on which default direction was learned for that node. Fitting the GBDT model with imputation of missing values led to a similar pattern of results as observed in the main analysis; the model did not outperform the FIT test in the clinically meaningful range of sensitivities (greater than 80%) (Figure C.2).

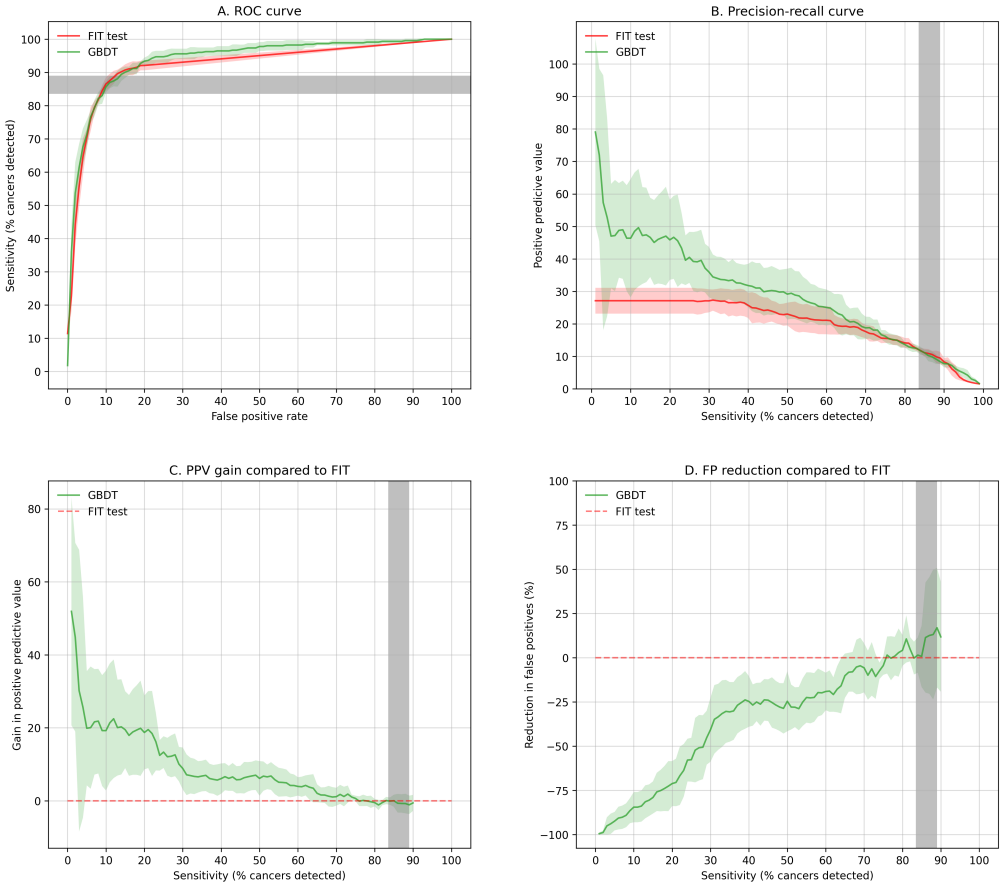


Figure C.2: Performance of the gradient-boosted decision tree (GBDT) model and the faecal immunochemical test (FIT) for detection of colorectal cancer, when missing data is imputed implicitly by the GBDT model. Panels display the ROC-curve (A), precision-recall curve (B), gain in positive predictive value compared to the FIT test (C), and percent reduction in false positives compared to FIT (D). Curves show the mean value of each quantity over 5 cross-validation folds; shaded area covers the mean plus-minus one standard deviation. The vertical grey area is shaded between the minimum and maximum sensitivities of FIT test greater than or equal to 10 µg/g on the held-out folds. All curves were interpolated to a fixed grid of values at 1-unit increments. Data for sensitivities greater than 90% is not shown for panels C and D because the interpolation for FIT test is inaccurate in that range.

C.3 Increasing the length of follow-up to 365 days

In the main analysis, cases of colorectal cancer were identified by looking for records of cancer within 180 days from the first FIT test. We increased the length of follow-up to 365 days to see if this makes a difference. However, longer length of follow-up leads to a smaller sample size as patients are included only if their first FIT occurs at least 365 days before the date of data cut-off, and it could make the task of prediction harder due to a longer time-horizon. 25,856 patients met the study inclusion criteria, and there were 398 cases of colorectal cancer. The pattern of results was similar to the main analysis: none of the models outperformed FIT at clinically meaningful range of sensitivities (greater than 80%), but the models tended to have higher positive predictive value at lower sensitivities (Figure C.3).

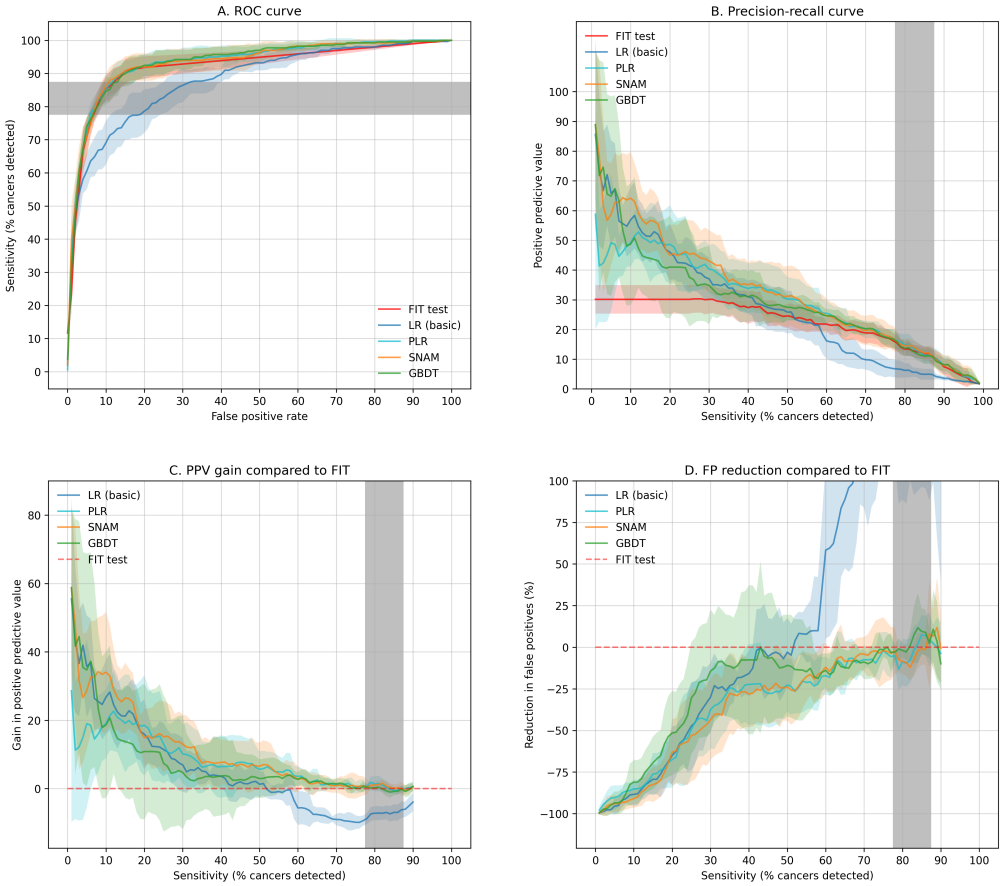


Figure C.3: Performance of machine learning models and the faecal immunochemical test (FIT) for detection of colorectal cancer, when length of follow-up for identification of cancer is set at 365 days. Panels display the ROC-curve (A), precision-recall curve (B), gain in positive predictive value compared to the FIT test (C), and percent reduction in false positives compared to FIT (D). Curves show the mean value of each quantity over 5 cross-validation folds; shaded area covers the mean plus-minus one standard deviation. The vertical grey area is shaded between the minimum and maximum sensitivities of FIT test greater than or equal to 10 µg/g on the held-out folds. All curves were interpolated to a fixed grid of values at 1-unit increments. Data for sensitivities greater than 90% is not shown for panels C and D because the interpolation for FIT test is inaccurate in that range. LR – basic logistic regression with linear data transformation, PLR – penalised logistic regression with log or quantile data transformation, SNAM – sparse neural additive model, GBDT – gradient-boosted decision tree.

C.4 Running the analysis with a different random seed

In the main analysis, models were evaluated using stratified 5-fold cross-validation. As there were 453 cases of cancer, this meant there were about 90 cancers in each fold. To check that the results are robust to how the 453 cancer patients are randomly divided between the five folds, we created the folds using a different random split and fitted the same models as in the main analysis. The pattern of results was again similar to the primary analysis, with the exception that the MLP model performed worse than previously at high levels of sensitivity (Figures C.4 and C.5). The performance of the MLP model could potentially be increased by tuning it more carefully – however, as none of the other models outperformed FIT test at clinically relevant levels of sensitivity (greater than 80%), and the models included flexible tree ensembles, it is highly unlikely that further tuning of MLP—which is also a flexible model—would lead to a model that outperforms the FIT test.

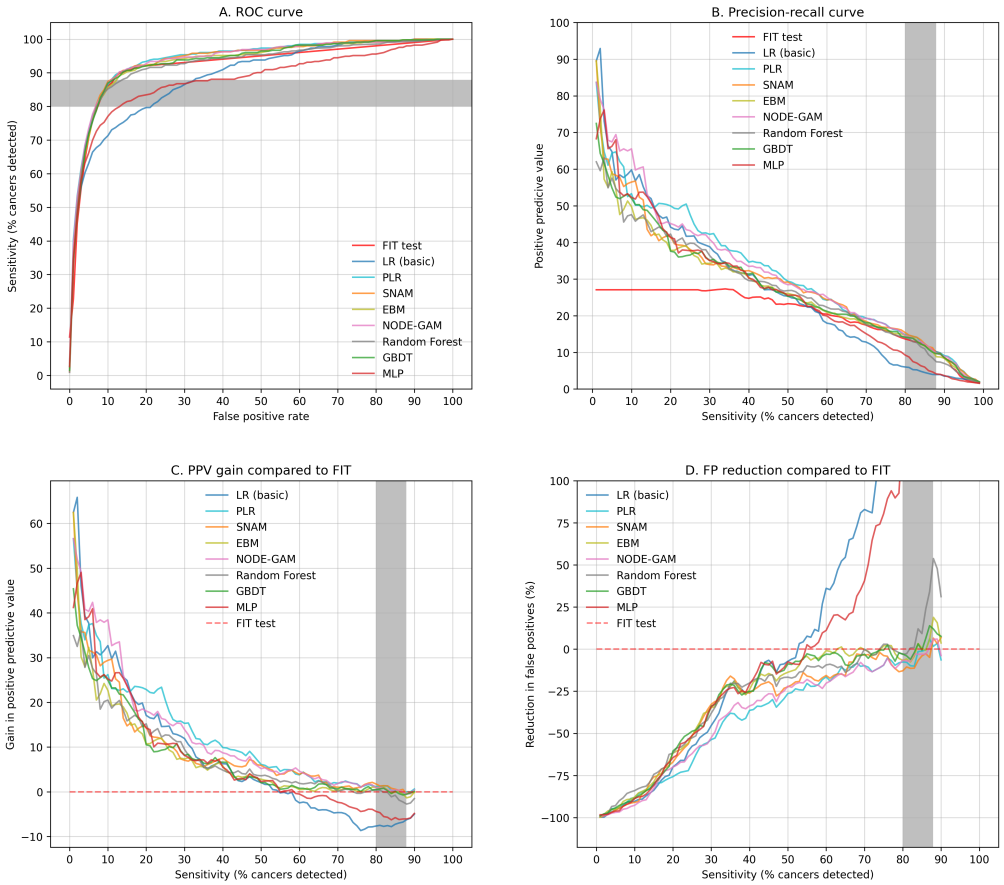


Figure C.4: Performance of machine learning models and the faecal immunochemical test (FIT) for detection of colorectal cancer, when the main analysis is run with a different random seed. Panels display the ROC-curve (A), precision-recall curve (B), gain in positive predictive value compared to the FIT test (C), and percent reduction in false positives compared to FIT (D). Curves show the mean value of each quantity over 5 cross-validation folds, but not displaying the variability between folds for clarity. The vertical grey area is shaded between the minimum and maximum sensitivities of FIT test greater than or equal to 10 µg/g on the held-out folds. All curves were interpolated to a fixed grid of values at 1-unit increments. Data for sensitivities greater than 90% is not shown for panels C and D because the interpolation for FIT test is inaccurate in that range. LR – basic logistic regression with linear data transformation, PLR – penalised logistic regression with log or quantile data transformation, SNAM – sparse neural additive model, GBDT – gradient-boosted decision tree, EBM – explainable boosting machine, NODE-GAM – neural oblivious decision tree ensemble GAM, MLP – multilayer perceptron.

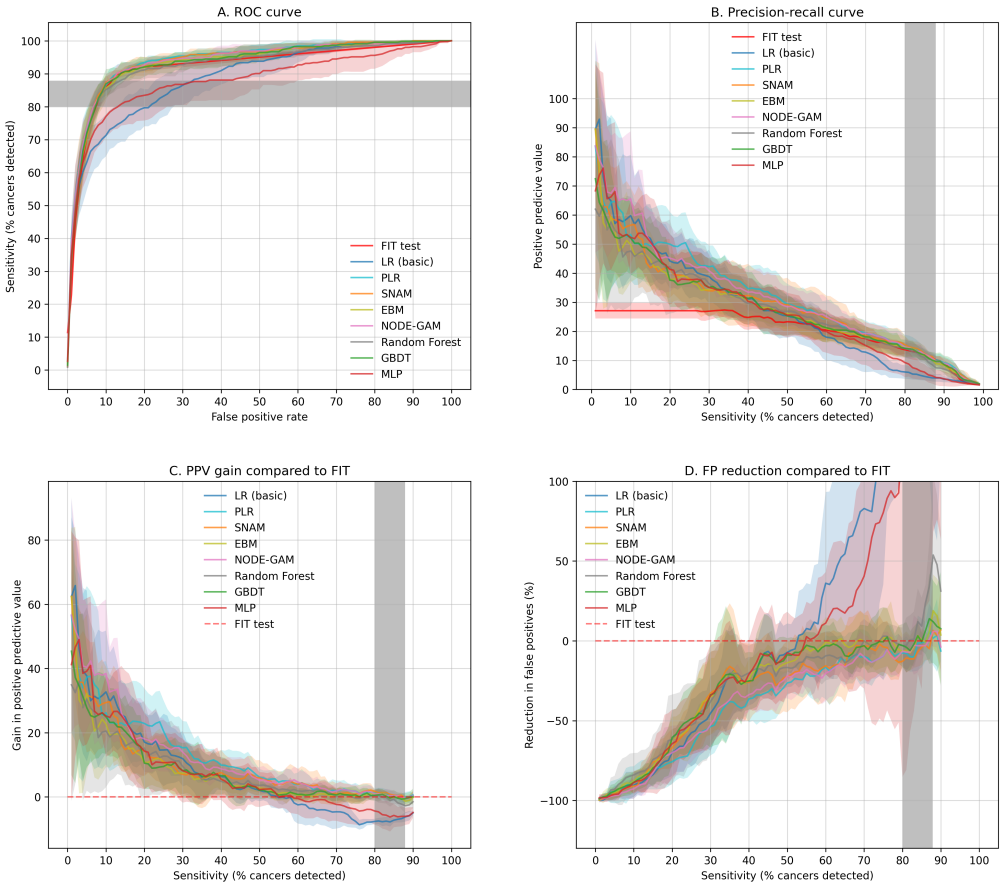


Figure C.5: Performance of machine learning models and the faecal immunochemical test (FIT) for detection of colorectal cancer, when the main analysis is run with a different random seed. For each quantity in panels A-D, the shaded areas show mean plus-minus one standard deviation over cross-validation folds. Please refer to Figure C.4 for more information.

C.5 Using pretrained models before training them with the novel loss function

The original publication describing the maximisation of area under the precision recall curve first pretrained their model with the standard cross-entropy loss, and then continued training using the novel average precision (AP) loss function. Here, we followed a similar process: we first tuned a penalised logistic regression model using the binary-cross entropy loss function. We then continued training the model using the AP loss, this time tuning the parameters of the loss function and the regularisation parameters of the pretrained model (because the effect of these is dependent on the scale of the losses returned by the loss function). The resulting model did not outperform the FIT test at clinically relevant levels of sensitivity (sensitivities greater than 80%, Figure C.6), and performed similarly to an AP-loss model trained from scratch (the PLR model in Figure 5.6).

It is possible that this procedure did not have a strong effect because our model was simple, whereas the AP loss publication used a deep residual neural network models that had 18 and 34 layers. Furthermore, when we trained the penalised logistic regression with AP loss from scratch, we found that the model had about 1.4% higher average precision score than the model trained with the standard loss, so we can be quite confident that the AP-loss model was not underperforming. Our pretraining procedure could be improved in the future by loading the pretrained weights of the model corresponding to each training set (rather than model selection set – see Figure C.6).

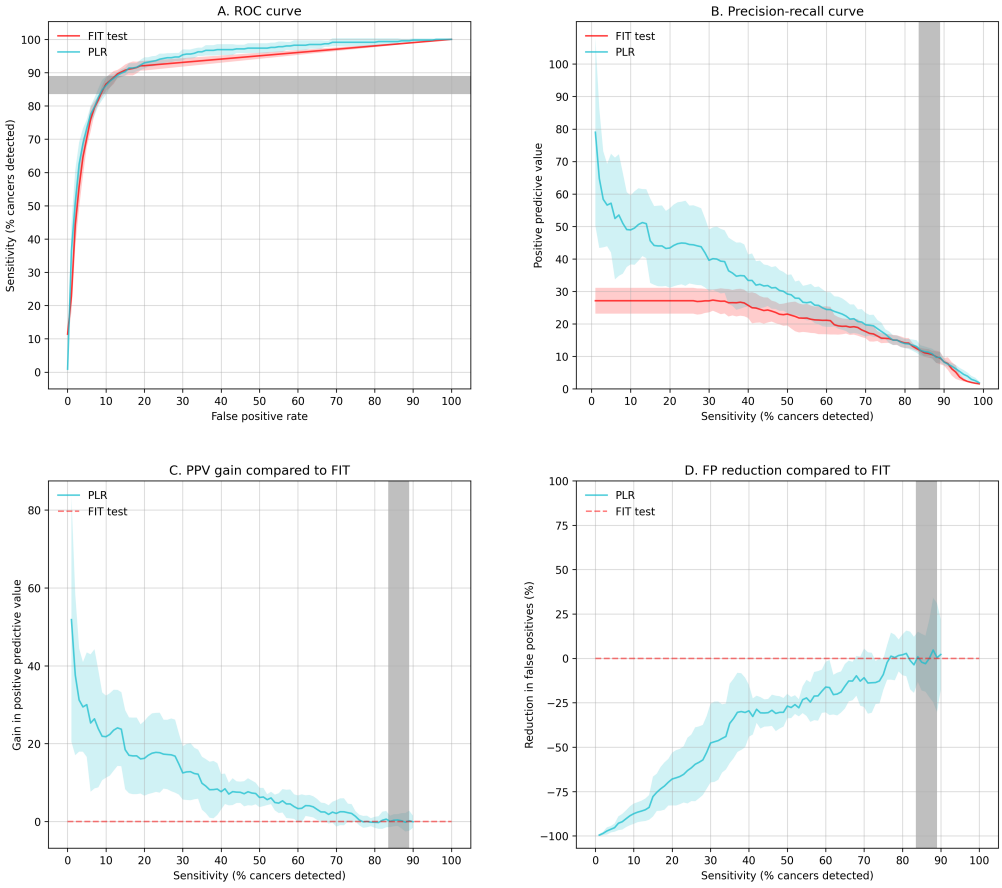


Figure C.6: Performance of penalised logistic regression (PLR) and the faecal immunochemical test (FIT) for the detection of colorectal cancer, when the PLR model was first tuned and trained using the binary cross-entropy loss function, and subsequently tuned and trained with the average precision loss. Panels display the ROC-curve (A), precision-recall curve (B), gain in positive predictive value compared to the FIT test (C), and percent reduction in false positives compared to FIT (D). Curves show the mean value of each quantity over 5 cross-validation folds; shaded area covers the mean plus-minus one standard deviation. The vertical grey area is shaded between the minimum and maximum sensitivities of FIT test greater than or equal to 10 µg/g on the held-out folds. All curves were interpolated to a fixed grid of values at 1-unit increments. Data for sensitivities greater than 90% is not shown for panels C and D because the interpolation for FIT test is inaccurate in that range.

C.6 Robustness to changes in the distribution of FIT values over time

The Oxford clinical biochemistry laboratory changed their FIT sample collection strategy in July 2021: instead of having patients send their faeces to the laboratory in a collection pot which had permitted homogenisation and reduced errors in collection, patients were given sample pickers for collecting faecal samples into a buffer solution that prevented haemoglobin degradation during prolonged transit. Before the sample pickers were introduced, there may have been some degradation of haemoglobin during the time the collection pot was transported to the laboratory, so it is important to understand to what extent the change to buffer kits influenced the distribution of FIT values. In an internal report, we found that the percentage of positive FITs was generally close to 9% before the buffer kits were adopted and increased to 11% in November-December 2021, two months after the kits had been adopted (Figure C.7). FIT positivity increased further from 11% to 16% in the year after adoption of the buffer kits, and this increase was positively correlated to the increasing monthly number of tests. The larger increase in FIT positivity is likely related to a change in the population of patients being tested, as FIT started to be used to additionally triage patients with higher-risk symptoms after the first COVID wave in Thames Valley.

To double check if changes in the distribution of FIT values over time may have affected the results of this analysis, we included data from October 2021 onwards (when the adoption of buffer kits was coming to an end). This yielded 9526 patients, with 137 cases of colorectal cancer. The 5-fold cross-validation procedure used for the main analysis would have left only 27 cases of cancer for each of the five held-out datasets, which could have made the results very sensitive to how the data is randomly split. We therefore performed this analysis with 5-fold cross-validation repeated 10 times, each time with a different random seed. This would have not been feasible for the main analysis, as it would have made the computational cost too high for some machine learning models. However, the penalised logistic regression and gradient boosted decision tree (GBDT) models could be fitted very fast to data,

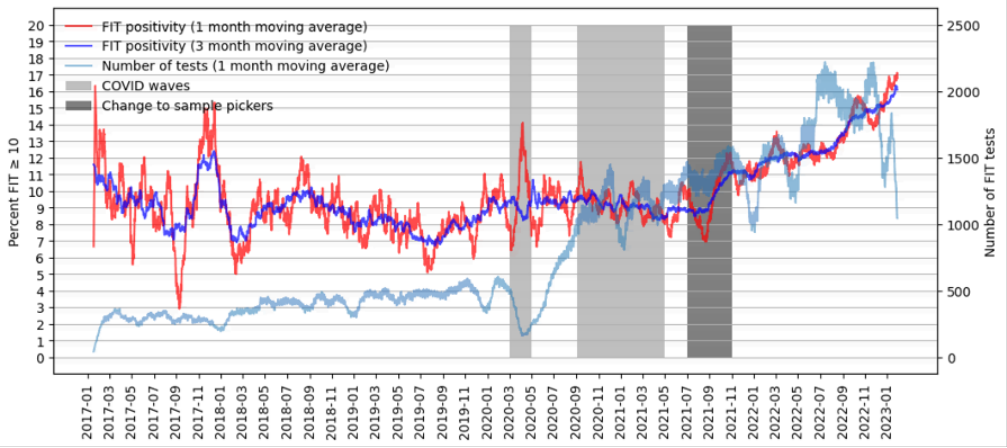


Figure C.7: Trends in FIT positivity according to moving average. Red: percent of positive tests according to 1 month moving average (30.44 day window centered at each observation). Blue: percent of positive tests according to 3 month moving average (91.31 day window centered at each observation). Light blue: number of FIT tests, according to 1 month moving average. Grey: first and second COVID waves. Dark grey: change to sample pickers.

and as these covered the cases of having both a simple and a flexible prediction function, we did not include additional models. The results were again similar to the main analysis: we saw gain over FIT at lower levels of sensitivity where less than 50% of all cancers were detected, and no obvious gain at higher, clinically relevant levels of sensitivity where more than 50% of all cancers were detected (Figure C.8).

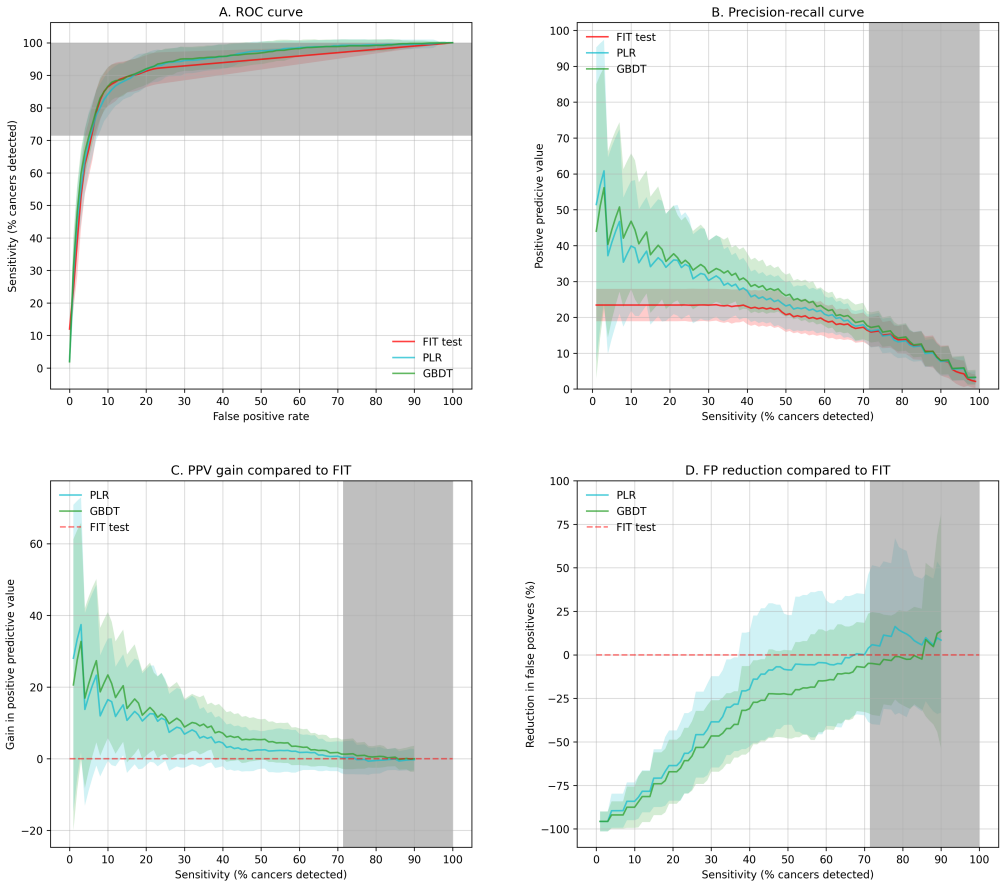


Figure C.8: Performance of the penalised logistic regression (PLR), gradient-boosted decision tree (GBDT), and faecal immunochemical test (FIT) for the detection of colorectal cancer on a subset of the data where most faecal samples were collected using sample pickers. Panels display the ROC-curve (A), precision-recall curve (B), gain in positive predictive value compared to the FIT test (C), and percent reduction in false positives compared to FIT (D). Curves show the mean value of each quantity over 50 cross-validation folds (5-fold cross-validation repeated 10 times); shaded area covers the mean plus-minus one standard deviation. The vertical grey area is shaded between the minimum and maximum sensitivities of FIT test greater than or equal to 10 µg/g on the held-out folds. All curves were interpolated to a fixed grid of values at 1-unit increments. Data for sensitivities greater than 90% is not shown for panels C and D because the interpolation for FIT test is inaccurate in that range.

C.7 Number of missing values in predictor variables

Table C.1: Number of observed and missing values for continuous variables used in the machine learning analysis

	Colorectal cancer			No colorectal cancer		
Variable	N_{obs}	N_{mis}	p_{mis} (%)	N_{obs}	N_{mis}	p_{mis} (%)
Haemoglobin	453	0	0.0	31511	0	0.0
Mean cell haemoglobin conc.	453	0	0.0	31511	0	0.0
Mean cell volume	453	0	0.0	31511	0	0.0
Platelets	453	0	0.0	31511	0	0.0
White cells	453	0	0.0	31511	0	0.0
NRBC a	453	0	0.0	31509	2	0.0
Eosinophils	453	0	0.0	31473	38	0.1
Monocytes	453	0	0.0	31473	38	0.1
IG	453	0	0.0	31472	39	0.1
Basophils	453	0	0.0	31471	40	0.1
Lymphocytes	453	0	0.0	31467	44	0.1
MPV	448	5	1.1	31365	146	0.5
Creatinine	446	7	1.5	29993	1518	4.8
Sodium	446	7	1.5	29980	1531	4.9
Potassium	445	8	1.8	29932	1579	5.0
imdd (max)	436	17	3.8	28708	2803	8.9
Albumin	421	32	7.1	28392	3119	9.9
Alk. phosphatase	420	33	7.3	28269	3242	10.3
ALT	416	37	8.2	27921	3590	11.4
Bilirubin	416	37	8.2	27910	3601	11.4
RDW	336	117	25.8	22748	8763	27.8
C-reactive protein	320	133	29.4	22428	9083	28.8
Serum ferritin	281	172	38.0	17212	14299	45.4
TSH	280	173	38.2	20428	11083	35.2
HbA1c (DCCT)	262	191	42.2	17740	13771	43.7
Iron	253	200	44.2	14907	16604	52.7
Transferrin	253	200	44.2	14907	16604	52.7
Calcium	192	261	57.6	14515	16996	53.9
Chol/hdl ratio	180	273	60.3	11367	20144	63.9
Serum b12	163	290	64.0	10519	20992	66.6
Serum folate	159	294	64.9	10086	21425	68.0
IgA	130	323	71.3	10441	21070	66.9
IgA tTg ab	97	356	78.6	8615	22896	72.7
Urea	93	360	79.5	5856	25655	81.4
Triglyceride	81	372	82.1	5674	25837	82.0
Pros spec ag	67	386	85.2	3566	27945	88.7
CA-125	63	390	86.1	5473	26038	82.6
Prothromb. time	55	398	87.9	3179	28332	89.9
Urine creat.	51	402	88.7	2776	28735	91.2
aPTT	50	403	89.0	3082	28429	90.2
Urine albumin	50	403	89.0	2562	28949	91.9
ESR	45	408	90.1	3775	27736	88.0
IgG	44	409	90.3	2719	28792	91.4
IgM	44	409	90.3	2719	28792	91.4

Continued on next page

Table C.1 – continued from previous page

	Colorectal cancer			No colorectal cancer		
Variable	N_{obs}	N_{mis}	p_{mis} (%)	N_{obs}	N_{mis}	p_{mis} (%)
BMI (max)	39	414	91.4	3613	27898	88.5
Amylase	39	414	91.4	1973	29538	93.7
25-OH vitamin d	38	415	91.6	3212	28299	89.8
Glucose	38	415	91.6	2932	28579	90.7
POCT PCO2	32	421	92.9	2013	29498	93.6
POCT CCL	32	421	92.9	2012	29499	93.6
PNA+	32	421	92.9	2012	29499	93.6
POCT MHb	32	421	92.9	2011	29500	93.6
POCT FO2Hb	32	421	92.9	2011	29500	93.6
POCT FCOHb	32	421	92.9	2010	29501	93.6
PCA++	32	421	92.9	2007	29504	93.6
PK+	32	421	92.9	2006	29505	93.6
POCT co3(p,st)c	32	421	92.9	2003	29508	93.6
POCT cLac	32	421	92.9	1995	29516	93.7
POCT P5OC	32	421	92.9	1985	29526	93.7
Free thyroxine	32	421	92.9	1862	29649	94.1
POCT cGlu	31	422	93.2	2005	29506	93.6
POCT temp	31	422	93.2	1900	29611	94.0
GGT	30	423	93.4	1933	29578	93.9
POCT FiO2	30	423	93.4	1808	29703	94.3
CEA	29	424	93.6	564	30947	98.2
Phosphate	26	427	94.3	2042	29469	93.5
Free kappa	24	429	94.7	1404	30107	95.5
Free lambda	24	429	94.7	1404	30107	95.5
k/l ratio	24	429	94.7	1404	30107	95.5
NT-proBNP	18	435	96.0	949	30562	97.0
CA-199	17	436	96.2	504	31007	98.4
PXP glucose	16	437	96.5	912	30599	97.1
Magnesium	13	440	97.1	1020	30491	96.8
POCT bilirubin	11	442	97.6	805	30706	97.4
LDH	10	443	97.8	509	31002	98.4
Uric acid	9	444	98.0	685	30826	97.8
Prothromb INR	9	444	98.0	588	30923	98.1
dose	9	444	98.0	580	30931	98.2
IS potassium	9	444	98.0	505	31006	98.4
IS sodium	9	444	98.0	505	31006	98.4
IS glucose	9	444	98.0	501	31010	98.4
IS anion gap	9	444	98.0	498	31013	98.4
IS ionised Ca	9	444	98.0	454	31057	98.6
IS chloride	9	444	98.0	453	31058	98.6
Time in range	8	445	98.2	541	30970	98.3
D-dimer	7	446	98.5	598	30913	98.1
POCT ur urobilinog	5	448	98.9	534	30977	98.3
Rheumat factor	4	449	99.1	408	31103	98.7
FSH	1	452	99.8	568	30943	98.2

Notes. Continuous variables above the double horizontal line were included in the main analysis and represent the more commonly available variables; continuous variables below the line were additionally included in the extended analysis. Not all blood tests were included included in models, as only one blood test was retained among highly correlated bloods.

C.8 Multiple imputation traces for variables used in the main analysis

All continuous variables used in the main machine learning analysis were imputed with the MICE (multiple imputation via chained equations) algorithm and a random forest imputation model. The data was first split into five folds, each serving as the held-out test set while the remaining four folds served as the model selection set. Imputation models were trained on each of the model selection sets, then applied to the model selection set and held-out test set. Traces of mean values for each variable over iterations of the MICE algorithm are shown in Figure C.9. Note that traces of each variable over the five model selection sets were usually fluctuating around each other as the maximum number of iterations was reached: this roughly indicates that the MICE algorithm converged. However, these traces are not expected to converge as fully as in a conventional application of MICE, because imputation models were trained on overlapping sets of data, while conventionally the models are trained on the same set of data. The traces of variables may therefore not converge fully, especially when the number of missing values is smaller (in which case the different data splits could contain a different subset of missing values which could have a different mean).

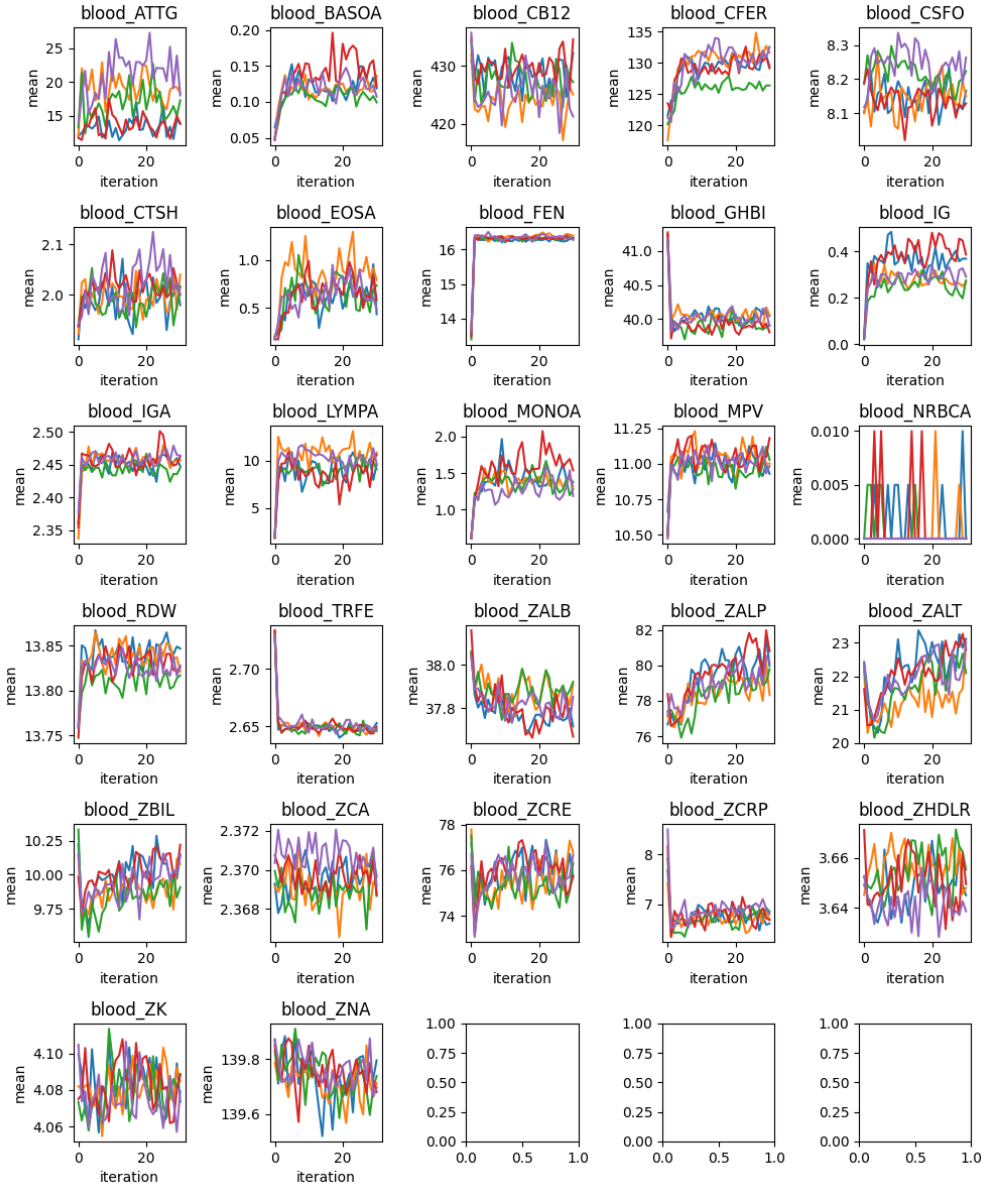


Figure C.9: Traces of mean values of imputed variables over the iterations of the MICE algorithm. The data was split into five folds, each serving as the held-out test set while the remaining four folds served as the model selection set. Imputation models were trained on each of the five model selection sets, and this figure displays the traces of mean values for each imputed variable over the five sets. For each variable, trace shows the mean of imputed values, not of all values. Blood test names are abbreviated - more 'readable' names are available upon request.

D

Machine learning models in more detail

Contents

Several machine learning (ML) models were used in this dissertation. Section D.1 describes the models employed in Chapter 5 for cancer risk prediction, and Section D.2 describes the conventional and novel loss functions that were used for training these models. Section D.3 describes the transformer models applied for text classification in Chapter 3. Finally, section D.4 lists the hyperparameters.

D.1 ML models used for colorectal cancer risk prediction

D.1.1 Linear models

Penalised logistic regression (PLR)

The penalised logistic regression (PLR) model makes predictions like an ordinary logistic model: a linear combination of predictor variables plus an intercept is passed through a sigmoid function to yield a prediction between 0 and 1. In the context of predicting colorectal cancer, the prediction is a risk score that can be interpreted as probability of cancer if the model is calibrated. Let y_i be the outcome for patient i, such that $y_i = 1$ if the patient had cancer and $y_i = 0$ otherwise, and let p_i be their prediction in $\in [0, 1]$. Also let $\mathbf{x}_i$ be a p-dimensional column vector of predictor variables for patient i and $\mathbf{w}$ be a column vector of regression coefficients (weights) associated with each variable, and let w_0 be the intercept term. The prediction is then given by [214]:

$$p_i = \sigma(w_0 + \mathbf{x}_i^{\mathrm{T}}\mathbf{w}) = \frac{1}{1 + \exp(-w_0 - \mathbf{x}_i^{\mathrm{T}}\mathbf{w})}$$

During training, an objective is minimised that consists of negative log-likelihood for n patients plus a regularisation term for the p regression coefficients [214]:

$$\mathcal{L} = -\sum_{i=1}^{n}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] + \lambda \sum_{j=1}^{p}((1 - \alpha)w_j^2/2 + \alpha|w_j|)$$

The regularisation term reduces overfitting (optimism) by shrinking the weights towards zero. Hyperparameter α controls the type of penalty applied: L1 penalty, also known as lasso regression ($\alpha = 1$); L2 penalty, also known as ridge regression ($\alpha = 0$); or a linear combination of the two known as 'elastic net' ($0 < \alpha < 1$) [214]. Hyperparameter λ controls the strength of the penalty. If λ is sufficiently large, lasso penalty forces some regression coefficients to be exactly zero and performs variable selection which generally yields a more interpretable model; ridge penalty on the other hand will keep all variables in the model, and this may yield a better predictor if many variables have a similarly strong relationship with the

outcome [160, pg. 223-224]. During optimisation, values are found for the regression coefficients that minimise the objective, while the hyperparameters α and λ are fixed (hyperparameter values can be selected through cross-validation). There are also methods to efficiently evaluate the coefficients over many values of λ [214].

PLR is an interpretable model: each exponentiated regression coefficient represents a multiplicative increase in the odds of cancer for a unit increase in the value of the variable. This can be seen from the following (the patient index i is dropped for readability):

$$\frac{p}{1-p} = e^{w_0+w_1x_1+...+w_px_p}$$

and

$$e^{w_0+w_1(x_1+1)+...+w_px_p} = e^{w_1} \cdot e^{w_0+w_1x_1+...+w_px_p} = e^{w_1} \cdot \frac{p}{1-p}$$

Logistic regression (including PLR) is a type of *generalised linear model* with a logit link function:

$$\text{logit}(p_i) = \log\frac{p_i}{1-p_i} = w_0 + \mathbf{x}_i^{\mathrm{T}}\mathbf{w}$$

D.1.2 Generalised additive models (GAM)

Generalised additive models (GAM) are a generalisation of linear models, where the linear predictor of the form $w_0+w_1x_1+...+w_px_p$ is replaced with an additive predictor $w_0 + f_1(x_1) + ... + f_p(x_p)$; when predicting a binary outcome, the additive predictor is again passed through a sigmoid function to constrain it in $[0, 1]$ [161, pg. 96-97]:

$$p = \sigma(w_0 + f_1(x_1) + ... + f_p(x_p)) = \frac{1}{1 + \exp(-(w_0 + f_1(x_1) + ... + f_p(x_p)))}$$

In the notation above, w_0 is the intercept term, $w_1...w_p$ are the regression coefficients associated with each of the p predictor variables $x_1...x_p$ in logistic regression, and $f_1...f_p$ are univariate functions that describe how the additive predictor changes as the value of the corresponding predictor variable increases in GAMs.

GAMs are more flexible than logistic regression, because the contribution of each predictor variable x_i to the linear predictor is not constrained to be a line as

in logistic regression, but they are still interpretable because due to additivity the effects of each predictor can be separately examined and visualised [161, pg. 86-87]. Traditionally, GAMs have been implemented using smoothers such as splines, and trained with the backfitting algorithm [161, pg. 90-91].

Sparse neural additive model (SNAM)

Neural additive models (NAM) are a type of GAM where the functions $f_1 ... f_p$ that describe the contributions of p predictor variables to the linear predictor each take the form of a feedforward neural network (FNN) [215] (see Section D.1.4 for more information on FNNs). NAM is thus a collection of FNNs, where each network processes only one input variable (or "feature"), and the outputs of these feature networks are aggregated (Figure D.1). The feature networks should be able to learn almost any kind of relationship between the values of the input variable and its contribution to the linear predictor (U-shape, sigmoidal, linear, etc) given enough hidden units, due to the universal approximation theorem of neural networks [169, pg. 194]. NAMs were used in Chapter 5 because they can be trained efficiently on large datasets using graphical processing units (GPU) on mini-batches of data [215], and it was straightforward to combine NAMs with novel loss functions that maximize areas under the ROC and precision-recall curves (see Chapter 5).

NAMs can be trained like other artificial neural networks, for example using adaptive learning rate optimisers with mini-batch training [215]. The contribution of individual variables can be interpreted by visualising the learned functions $f_1 ... f_p$.

The sparse NAM (SNAM) is a variation of NAM, where during model training, a grouped lasso penalty is applied to the feature networks, which can nullify some of the feature networks and thus serves as a variable selection method [162]. This was desirable in Chapter 5, as I applied SNAM to a relatively large number of predictor variables (582). Without the grouped lasso penalty, the network may have otherwise been more prone to overfitting the data.

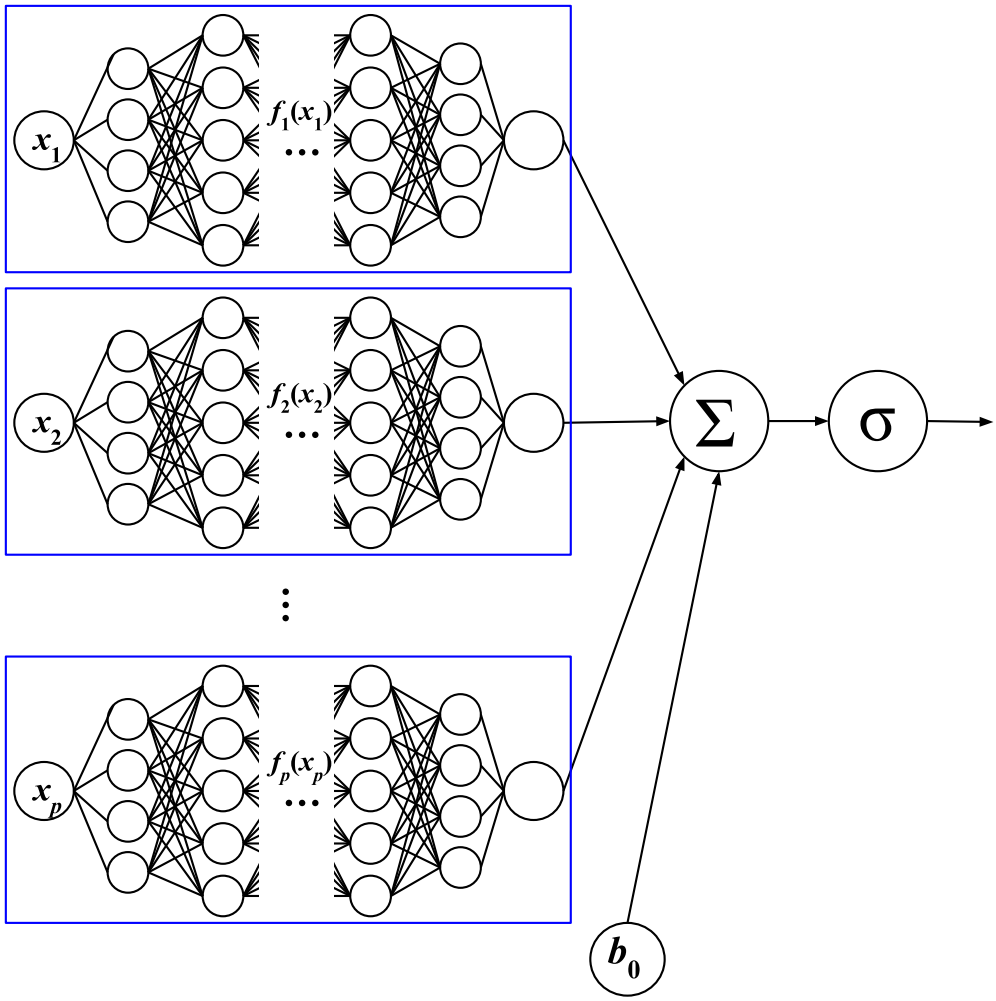


Figure D.1: Structure of the Neural Additive Model (NAM), based on Agarwal et al [215].

Neural oblivious decision tree ensemble GAM (NODE-GAM)

NODE-GAM is an implementation of GAM that uses differentiable oblivious decision trees [164]. An oblivious decision tree (ODT) of depth d compares whether each of the d splitting variables was above a learnable threshold, and it is made differentiable by choosing the splitting variables by a sparse weighted sum of all variables, and by replacing the threshold comparison operation with a function that operates like a sparse sigmoid on the difference between the value of the variable and its threshold [216]. NODE-GAM consists of L layers, where each layer has I differentiable ODTs of the same depth, and the outputs of all previous layers become inputs to the next layer similar to the "NODE" architecture [216]. However, to make it a GAM, the

architecture is constrained such that a single ODT can only operate on one input variable, and ODTs can be connected across layers only if they operate on the same variable. NODE-GAM also allows some trees (and their connections) to operate on the same pair of variables for detecting pairwise interactions. Furthermore, NODE-GAM includes additional attention weights in the feature selection function within each tree to help determine which previous trees to focus on. NODE-GAM computes the final prediction as a weighted average of the outputs of all trees, where the weights are learned by an additional linear layer. Please refer to the NODE-GAM paper for a visualisation of the model architecture [164].

For regularisation, NODE-GAM applies dropout to the outputs of each tree (controlled by the output_dropout parameter), and to the weights in the final linear layer (last_dropout). To increase diversity of trees, each tree operates on a random subset of variables (colsample_bytree). L2 penalty is also applied to the weights in the last linear layer.

There are several advantages to this architecture: it is based on differentiable ODTs which permits parallel computation and allows the model to be efficiently trained using mini-batches and GPUs like other neural network models; unlike NAMs, NODE-GAM does not require each input variable to have its own neural network which can reduce the number of parameters and computational cost of the model; NODE-GAM architecture automatically selects single variables and pairwise interactions among the input variables; and due to the use of ODTs, NODE-GAM may better learn feature functions that have quick non-linear jumps [164]. Finally, since NODE-GAM is implemented in neural network software, it is possible to relatively easily apply novel loss functions for training the model (see Chapter 5). Like other GAMs, the contribution of input variables can be interpreted by visualising the feature functions of each variable (see Figure 5.12 in Chapter 5 for an example). When the model also detects pairwise interactions, it is called NODE-GA^{2}M.

Explainable boosting machine (EBM)

Explainable boosting machine (EBM) aims to improve on the basic GAM structure by including pairwise interactions between variables to make a more powerful prediction model [163]. The additive predictor thus becomes

$$w_0 + \sum f_i(x_i) + \sum f_{ij}(x_i, x_j)$$

and Lou et al call this *generalised additive models plus interactions*, or GA^{2}Ms [163].

To achieve this, Lou et al propose an algorithm called FAST that efficiently measures and ranks the contribution of all possible pairs of variables for interaction detection. To learn individual feature functions, they use gradient boosting with shallow decision trees, where in each cycle of boosting they sequentially cycle through all predictor variables [217]. Boosting [167] can be intuitively understood as a process of slow learning, where weak learners such as shallow decision trees are fitted sequentially to data, such that the next learner improves on the predictions of the previous learner [160, pg. 321]. EBM was used in Chapter 5 due to its automatic interaction detection, and because it has been applied successfully to clinical data in terms of model accuracy and ability to interpret the effects of individual predictor variables [165].

D.1.3 Decision tree ensembles

Tree-based methods partition the space of predictor variables into simple regions, and in the case of classification (e.g. "cancer" vs "no cancer"), often use the most common class in a region to make a prediction for all observations that fall into the region [160, pg. 303, 311]. However, single decision trees often have a smaller prediction accuracy than other methods, and their structure can be very sensitive to the particular sample of data they are trained on; practitioners usually employ ensembles of multiple decision trees to improve on single trees [160, pg. 316]. Two types of commonly used decision tree ensembles are described here: gradient boosted decision tree (GBDT) and random forests (RF). Both models are collections of

decision trees: the predictions of individual trees are aggregated to obtain an overall prediction. GBDT and RF differ in how the trees are learned from data.

Gradient boosted decision tree (GBDT)

In Chapter 5, Chen and Guestrin's implementation of GBDT known as XGBoost was used, as it is known to have performed well in machine learning competitions [168]. Let f_k be the k-th decision tree and let $\mathbf{x}_i$ be a p-dimensional column vector of predictor variables for patient i. The real-valued prediction from K trees is given by

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i)$$

Each tree has T leaves, a T-dimensional vector of weights $\mathbf{w}$ associated with the leaves, and a decision rule or structure q that assigns a vector of input variables to one of the leaves. By default, the complexity of each tree is given by (tree-specific indices are dropped):

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

which is used as a regularisation term when training the model.

The model is trained by adding one tree at a time. At each round, a new tree is added to improve the prediction at the previous round. The objective to be optimised is the loss l at round t plus a regularisation term for the added tree f_t:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

This approach of sequential training is known as boosting [160]. To quickly optimise the objective, a second-order Taylor expansion of the loss at point $\hat{y}^{(t-1)}$ is used:

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$. After constant terms are removed, this yields:

$$\tilde{\mathcal{L}}^{(t)} \simeq \sum_{i=1}^{n} [g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

After further derivation, Chen and Guestrin show that the optimal loss value for a fixed tree structure q is given by:

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i\in I_j} g_i)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T$$

where I_j is the set of patients (instances, examples) that fall on leaf j. This equation is used as a measure of tree structure quality when attempting to find an optimal structure. As it is usually not feasible to evaluate all possible structures, a greedy algorithm is used that iteratively adds branches to a tree. In analyses described in Chapter 5, the task was colorectal cancer risk prediction, so the loss function was the logistic loss used in logistic regression.

The importance of individual predictor variables can be quantified by computing the average gain in the tree structure quality when that variable is used to split a leaf [218]. However, compared to GAMs, it is still harder to understand the contribution of each variable, because a single variable can appear in multiple trees, each of which may also incorporate other variables.

Random forests (RF)

In random forests (RF) [166], a number of decision trees are built on bootstrap samples of original data (a procedure known as bagging), and for each split in each decision tree a random sample of m predictor variables out of p predictors are considered as split candidates, typically $\sqrt{p}$ [160, pg. 319]. The motivation of bagging is to reduce the variance of the prediction model (individual decision trees have high variance in the sense that if a single tree would be trained on two different samples from the same population the individual trees may be quite different), and the motivation of using a random subset of predictors as split candidates is to decorrelate the trees to improve the variance reduction in bagging (if all variables were used as split candidates for all trees and some variables are stronger predictors, then the resulting trees may be similar because the stronger predictors are likely to appear in many trees) [160, pg. 319-320]. Similarly to boosted trees, the importance of individual variables can be quantified using mean decrease in node impurity

(measured by statistics such as gini index) [160, pg. 312, 320]. RF was used as a prediction model in Chapter 5, as it is known to perform similarly to boosted trees in many cases but can be simpler to tune [219, pg. 587].

D.1.4 Feedforward neural networks

A feedforward neural network, also known as a multilayer perceptron (MLP), usually consists of multiple layers stacked one after the other, such that the outputs of each layer are the inputs to the next layer [169]. Each layer linearly transforms its input values to a pre-specified number of output values, adds a bias term, and applies a non-linear activation function. For example, the first two layers in a simple feedforward network are given by [169]

$$h^{(1)} = g^{(1)}(W^{(1)\mathrm{T}}x + b^{(1)})$$
$$h^{(2)} = g^{(2)}(W^{(2)\mathrm{T}}h^{(1)} + b^{(2)})$$

where $W^{(1)}$ is a p by q matrix with p denoting the number of input values and q denoting the number of output values in the first layer, $b^{(1)}$ is the bias for the first layer (a scalar), and $g^{(1)}$ is the activation function for the first layer. $W^{(1)\mathrm{T}}x$ thus computes q linear combinations of the input variables x. The activation functions used in input and hidden layers (all layers before the last) are often rectified linear units (ReLU) that compute $g(z) = max(0, z)$ [169]. In Chapter 5 of this dissertation, the task of the neural network was prediction of colorectal cancer risk, so the output layer returned only one value and the sigmoid activation function was used to transform it into [0, 1] interval like in logistic regression. Each layer can also be visually represented as consisting of a number of nodes equal to the number of output values the layer computes (Figure D.2).

Feedforward neural networks (and artificial neural networks in general) are commonly trained on minibatches of data using stochastic gradient descent (SGD) and related methods that modify SGD to improve on it [220, pg. 275-276, 307]. Training can be regularised by applying L1 and L2 penalties to the weights, and by randomly dropping out some hidden nodes during training ("dropout") [174,

221]. Training can also benefit from batch normalisation that transforms the outputs of a layer by subtracting the mean and dividing by the standard deviation within a mini-batch [220, 222]. However, batch normalisation and dropout may not work well together [223].

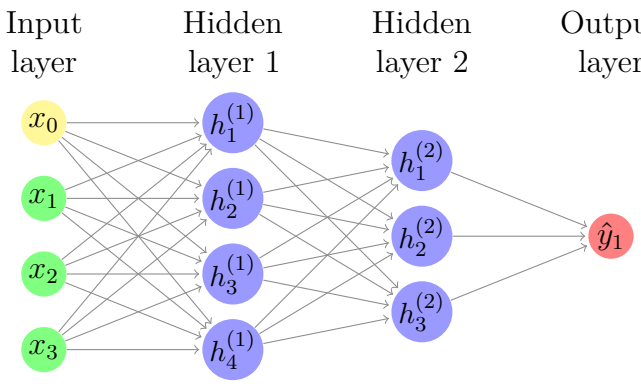


Figure D.2: A feedforward neural network with two hidden layers.

D.2 Loss functions used for colorectal cancer risk prediction

In Chapter 5, ML models were trained for predicting the risk of colorectal cancer using three different loss functions: the conventional binary cross-entropy (BCE), the AUC margin loss [171], and the AP loss [172]. The latter two losses attempt to maximize areas under the ROC and precision-recall curves, respectively, and may work better for imbalanced data. The data used in Chapter 5 was imbalanced, as less than 2% of patients had the outcome (cancer).

D.2.1 Binary cross-entropy

Artificial neural networks can often be considered parametric models that define a probability distribution $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$, so the principle of maximum likelihood is used for parameter estimation, which is equivalent to using cross-entropy as the loss function [169, pg. 174]. Binary cross-entropy (BCE) is often used when the outcome belongs to one of two classes (e.g. 'cancer' or 'no cancer') and is defined as [224]:

$$\ell = -\sum_{i=1}^{n} z_i[y_i \log(p_i) + (1-y_i)\log(1-p_i)]$$

where $p_i \in [0,1]$ is a prediction returned by the model for patient i, $y_i \in \{0,1\}$ indicates the class label (e.g. 0: no cancer, 1: cancer), and z_i is an optional weight. The predicted probabilities p_i are often computed as $p_i = \sigma(h_i) = 1/(1+\exp(-h_i))$ where h_i is the output of the last linear layer of an artificial neural network. Instead of applying the sigmoid operation, the binary cross-entropy loss can be computed directly with the h_i as inputs because this allows to use the log-sum-exp trick for numerical stability [224, 225].

D.2.2 Maximising area under the ROC curve

Receiver-operating characteristic (ROC) curve displays sensitivity (y-axis) against false positive rate (x-axis) for each possible classification threshold [226]. It can be summarised by computing area under the ROC curve, abbreviated here as

AUROC to distinguish it from area under the precision-recall curve, but it is also known as AUC. AUROC is equivalent to the probability that a randomly chosen positive example (e.g. a patient with cancer) will receive a higher risk score than a randomly chosen negative example (e.g. a patient without cancer) [226]. In datasets where the proportion of positive examples is small, a prediction model could be better trained by maximising AUROC, as high AUROC implies that all positives tend to score higher than negatives [170].

AUROC can be defined as [170, 227]:

$$\begin{aligned}\text{AUROC}(\mathbf{w}) &= \Pr(h_{\mathbf{w}}(\mathbf{x}) \geq h_{\mathbf{w}}(\mathbf{x}')|y = 1, y' = -1) \\ &= \mathbb{E}[\mathbb{I}(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}') \geq 0)|y = 1, y' = -1]\end{aligned}$$

where $\mathbf{x}$ is a randomly drawn positive example, $\mathbf{x}'$ is a randomly drawn negative example, $h_{\mathbf{w}}(\mathbf{x})$ is a prediction made by the model h with parameters w, and $y_i \in -1, 1$ denotes the outcome (e.g. -1: no cancer, 1: cancer).

To optimise AUROC with respect to model parameters, the indicator function that prevents differentiation can be replaced with a surrogate loss, and the optimisation problem can be expressed as [170]:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{N_+ N_-} \sum_{\mathbf{x}\in S_+} \sum_{\mathbf{x}'\in S_-} \ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}'))$$

where S_+ and S_- are the sets of positive and negative examples, N_+ and N_- are the numbers of positive and negative examples, and ℓ is the surrogate loss.

Yuan et al note that a squared loss function $\ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}')) = (1 - h_{\mathbf{w}}(\mathbf{x}) + h_{\mathbf{w}}(\mathbf{x}'))^2$ has been a promising surrogate loss as it allows to reformulate the optimisation problem efficiently, but it may underperform on easy and on noisy data [170]. They developed an improved loss function based on the squared loss, called the AUC margin loss, that starts from reformulating the squared loss as a sum of three terms: the first two minimise the variance of prediction scores on positive and negative data, respectively, and the third attempts to increase the distance between mean scores of positive and negative data [170, pg. 3043]. Finally, they modify the third term to be a squared hinge loss, which yields the following AUC margin loss:

$$A_{\mathrm{M}}(\mathbf{w}) = A_1(\mathbf{w}) + A_2(\mathbf{w}) + \max_{\alpha \geq 0} \left(2\alpha \left(m - a(\mathbf{w}) + b(\mathbf{w})\right) - \alpha^2\right)$$

where $A_1(\mathbf{w}) = \mathbb{E}[(h_{\mathbf{w}}(\mathbf{x}) - a(\mathbf{w}))^2 | y = 1]$, $a(\mathbf{w}) = \mathbb{E}[h_{\mathbf{w}}(\mathbf{x}) | y = 1]$, $A_2(\mathbf{w}) = \mathbb{E}[(h_{\mathbf{w}}(\mathbf{x}') - b(\mathbf{w}))^2 | y' = -1]$, $b(\mathbf{w}) = \mathbb{E}[h_{\mathbf{w}}(\mathbf{x}') | y' = -1]$, and m is a margin parameter and α is a non-negativity constraint. Note that the last term uses the relationship $s^2 = \max_{\alpha \geq 0}(2\alpha - \alpha^2)$, which intuitively holds because a maximum of a quadratic in α occurs at the point $\alpha = s$. Yuan et al claim that unlike the squared loss, the AUC margin loss is robust to both easy and noisy data. They also note that maximizing the AUC margin loss directly may not yield the best performance, and instead recommend pretraining a model with a conventional loss function and then continuing training with the AUC margin loss.

In a subsequent publication, Yuan et al outline an AUC maximisation approach where a conventional loss and AUC loss are maximized compositionally [171]. Their objective is

$$\min_{\mathbf{w} \in \mathbb{R}^d} L_{\mathrm{AUC}}(\mathbf{w} - \alpha \nabla L_{\mathrm{AVG}}(\mathbf{w}))$$

where L_{AUC} is the AUC margin loss function, L_{AVG} is a conventional loss function (the BCE loss in my use case), and α is a tuning parameter. Here the outer function is the AUC loss and the inner function is a gradient step for minimising the BCE loss. They propose a specific stochastic optimisation algorithm that alternates between taking an optimisation step for the BCE loss and for the AUC loss, and claim that it outperforms both the conventional cross-entropy loss and the two-stage AUC optimisation approach (first BCE, then AUC) on medical image datasets.

Compositional maximisation of AUC margin loss and BCE loss was used in Chapter 5, because it outperformed both the conventional and two-stage approaches according to Yuan et al [171], and because it was straightforward to apply this method to models implemented in the pytorch language via the libauc python library [177].

D.2.3 Maximising area under the precision-recall curve

Precision-recall (PR) curve displays precision (y-axis) against recall (x-axis) for each possible classification threshold [121] (see Figure 5.6 for an example). Precision and recall are also known as positive predictive value and sensitivity. PR curve may better characterise model performance than ROC curve when the proportion of positive examples such as cancer cases is small [121]. This is because the number of negative examples is much higher than the number of positive examples, and so a large change in the number of false positives can still appear as a small change in false positive rate that is shown in the ROC curve [121].

PR curve can be summarised by computing area under the curve, which is equivalent to summarising precision over all levels of recall. Average precision (AP) is an estimator for area under the precision-recall curve, and can be defined as [172]:

$$\text{AP} = \frac{1}{n_+} \sum_{i=1}^{n} \mathbb{I}(y_i = 1) \frac{\sum_{s=1}^{n} \mathbb{I}(y_s = 1)\mathbb{I}(h_w(x_s) \geq h_w(x_i))}{\sum_{s=1}^{n} \mathbb{I}(h_w(x_s) \geq h_w(x_i))}$$

where n_+ is the number of positive examples, h_w is a prediction model with parameters w, and $h_w(x_i)$ is the model prediction for a single example x_i. For each positive observation, this equation computes the proportion of positive observations among all observations that have the same or higher predicted score than the current positive observation. Average precision can also be expressed as a weighted average of precisions at each threshold of the precision-recall curve, weighted by the increase in recall from the previous threshold [122].

Given that PR-curve can be a good summary of model performance in imbalanced datasets, one may ask whether a risk prediction model can be directly optimised to have a high area under the PR-curve. Qi et al recently developed a method for optimising average precision that they claimed to have outperformed other methods [172], and that can be easily combined with neural network models implemented in the pytorch language via the libauc python library [177]. To optimise AP, Qi et al replace the non-differentiable indicator functions in the numerator and denominator of AP with a surrogate loss function ℓ and obtain the following objective:

$$\min_{\mathbf{w}} P(\mathbf{w}) = \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{D}_+} \frac{-\sum_{s=1}^{n} \mathbb{I}(y_s = 1)\ell(\mathbf{w}; \mathbf{x}_s; \mathbf{x}_i)}{\sum_{s=1}^{n} \ell(\mathbf{w}; \mathbf{x}_s; \mathbf{x}_i)}$$

They use a squared hinge loss with m as the margin parameter:

$$\ell(\mathbf{w}; \mathbf{x}_s; \mathbf{x}_i) = \left(\max\left\{m - \left(h_{\mathbf{w}}(\mathbf{x}_i) - h_{\mathbf{w}}(\mathbf{x}_s)\right), 0\right\}\right)^2$$

Finally, they rewrite the objective as a sum of coupled compositional functions and propose stochastic algorithms named SOAP ("stochastic optimisation of AP"). Due to their complexity, the algorithms are beyond the scope of this appendix.

D.3 Transformers for text classification

The transformer is an artificial neural network architecture introduced by Vaswani et al in the context of translating text between languages [88], and has since been used in well-known generative machine learning models such as the OpenAI's GPT-4 [228] and Google's Gemini [229], in many natural language processing tasks and also in other fields like computer vision [230]. Variants of the BERT transformer encoder model [82] were used in Chapter 3 for classifying text extracts that discuss cancer recurrence and metastasis. This section first describes the multi-head attention mechanism that is core to all transformer architectures, describes the transformer encoder layer that is relevant for text classification, and then discusses BERT and its variants.

D.3.1 Multi-head attention

A core part of the transformer is the 'scaled dot-product attention' that operates on keys, queries and values (all vectors). Each value vector is given a weight depending on the compatibility of its key with a query, and a weighted sum of values is computed. Query, key and value vectors are arranged into matrices Q, K and V, respectively, and the attention is given by [88]

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

where d_k is the dimension of query and key vectors. For example, suppose a sentence is split into T tokens, each associated with a numeric vector (an 'embedding'). The query of the first token is compared to the keys of all other tokens (including the token itself) to compute a weight for each token. A weighted average of the values of all tokens is computed, such that the values of tokens whose key is more similar to the query are given a higher weight. The result is stored at the position of the first token, and the same process is repeated for other tokens. This allows a token to 'communicate' with other tokens [231], so that it takes into account information from other parts of the sentence. A scaling factor d_k is used to avoid the dot products growing large and saturating the softmax. Importantly, the

attention can be computed for all tokens in parallel which allows the transformer to be scaled to large datasets [88].

Instead of using a single attention operation, the scaled dot-product attention is computed h times in parallel, each time using different Q, K and V matrices. This was accomplished by creating h different linear projections from the original embedding at the token position and concatenating the results. This is known as *multi-head attention*. Note that because the key, query and value matrices are projected from the same input embedding, it is also called *self-attention*.

The multi-head attention layer is incorporated into the transformer encoder layer, and usually multiple such layers are stacked (see below). Vaswani et al also used a decoder layer with modified multi-head attention, where keys and values for the attention are taken from the output of the encoder layer, and where a masking operation prevents the current token from communicating with next tokens in a sequence.

D.3.2 The transformer encoder

The original transformer model consisted of encoder and decoder stacks for text translation [88], but text classification can be performed with the encoder alone [82]. Before data is passed to the encoder, the input sequence is tokenized, each token is mapped to a numerical vector (an 'embedding'), and a positional encoding vector is added to each token that contains information about the position of that token in the sequence. The embedded and encoded data is then passed through multiple transformer encoder blocks. Each block applies the multi-head attention operation followed by a feedforward neural network (Figure D.3). Note that different parts of the output embedding of the attention layer correspond to different attention heads, so the FNN layer applied to the embedding presumably helps integrate information across the heads. When embeddings of the input sequence are passed through the transformer encoder layers, the embeddings at the position of each token can start to contain information about other positions in the sequence due

to the attention mechanism. This representation of the sequence can then be used for classifying the sequence.

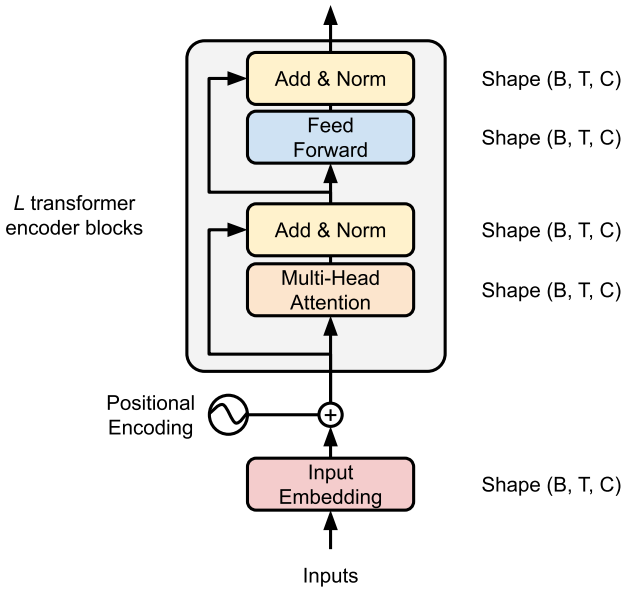


Figure D.3: Transformer encoder, based on Figure 1 in Vaswani et al [88]. Originally, the input data is of shape (B, T, C) where B is the number of sequences in a batch, T is the number of tokens in a sequence, and C is the dimension of the token embeddings. Positional encoding vectors are added to input embeddings, and positionally encoded input data is passed through L transformer encoder blocks to produce contextualised embeddings. Each transformer encoder block applies multi-head attention followed by a feedforward neural network (FNN) that processes each position separately and identically. Each block also includes residual connections followed by layer normalisation in the attention and feedforward sublayers (the input to each sublayer is added to the output of the sublayer, and layer normalisation is applied). In the original implementation, the shape of the input data (B, T, C) is retained throughout the layers.

.

D.3.3 Bidirectional encoder representations from transformers (BERT)

BERT is a transformer encoder model that was pre-trained on general text (BooksCorpus and English Wikipedia) to create deep bidirectional representations of language that can subsequently be fine-tuned at a low computational cost for performing various downstream natural language processing (NLP) tasks [82]. BERT was pre-trained using two unsupervised learning tasks: in masked language modelling, some input tokens were masked at random and predicted by the model; in next sentence

prediction, the model had to distinguish whether sentence B in a pair (A, B) followed A or was randomly selected. BERT contains bi-directional representations of text, because for each token position, the self-attention module in the transformer encoder layer integrates information from all token positions (both left and right). Two main BERT models exist: BERT-base has 12 encoder blocks with a hidden dimension of 768 and 12 attention heads, totalling 110 million parameters; BERT-large has 24 encoder blocks with a hidden dimension of 1024 and 16 attention heads, totalling 340 million parameters. Devlin et al successfully fine-tuned BERT for eleven NLP tasks, outperforming state-of-the-art models [82]. BERT can be fine-tuned for classification by adding a linear classification layer followed by softmax on top of the transformer encoder output and fine-tuning all parameters end to end. For classification, the encoder output corresponding to the first token is usually used, because Devlin et al used a tokenisation scheme where the first token is called 'CLS' and was used for next sentence prediction during pre-training.

D.3.4 Distilled and specialised versions of BERT

In Chapter 3, two variants of BERT were fine-tuned for classifying clinical text extracts that discuss cancer recurrence or metastasis. BERT-based models were used, because fine-tuned BERTs achieved state-of-the-art performance on general language understanding tasks when BERT was released [82], and because the transformer encoder structure in BERT integrates information from both the left and right context in a sentence. Consideration of both left and right context was important, as one of my tasks was to classify whether a text extract that discusses recurrence does so in an affirmative, possible or negative sense, and the information that would allow classification was contained on the left, right or both sides of the recurrence keyword. Specifically, DistilBERT [89] was used as it performed similarly to BERT at a lower computational cost, and Bio_ClinicalBERT [91] was applied because it had been further pre-trained on clinical data.

DistilBERT is 40% smaller than BERT but retains 97% of its performance in general language understanding [89]. DistilBERT was created through the

technique of knowledge distillation [232]. DistilBERT has the same transformer encoder structure as BERT but twice as few layers.

Bio_ClinicalBERT is a variant of BERT that was further pre-trained on biomedical and electronic health record data [90, 91]. Bio_ClinicalBERT was initialised from the BioBERT, which is BERT pre-trained on biomedical text (PubMed abstracts and PMC full-text articles) [233]. Bio_ClinicalBERT was then additionally pre-trained using clinical text from the MIMIC-III critical care database [187].

D.4 Hyperparameters for risk prediction models

To train a machine learning model, the practitioner has to usually choose among several settings (or hyperparameters) that are beyond the original parameters of the model that are learned from data. This section lists the hyperparameters I used for training the ML models for colorectal cancer risk prediction (hyperparameters for BERT models are described in Chapter 3). I generally initialised models with default parameters recommended in their packages or publications, and chose the tuning ranges based on reading about the parameters and experimenting on data (the data did not include the held-out cross-validation sets used for model evaluation). More detail about model parameters is available in python code that will be freely available once the corresponding thesis chapters are published.

For PLR, I used L1 regularisation (lasso) to achieve variable selection, and I tuned the penalty coefficient loguniformly in [-3, 3] (i.e. the coefficient was expressed as 10^a where $a \sim \text{Unif}(-3, 3)$).

For SNAM, I tuned the structure of feature neural networks over the set {'128-128-64', '64-64-32', '1024', '64'}. For example, '128-128-64' means there were three hidden layers with 128, 128 and 64 hidden units. I tuned the group L1 penalty for feature nets loguniformly in [-5, 3], and L1 penalty for linear terms loguniformly in [-5, 3]. Note that some variables were entered to SNAM as indicators (e.g. gender), so there was no need to include a feature neural network for these, and a regular L1 penalty was used instead.

For EBM, I tuned the number of boosting rounds (trees) uniformly in [500, 2500] with step size 100, the learning rate loguniformly in [-5, -1], the number of pairwise interactions in {5, 6, 7, 8, 9, 10}, maximum number of bins in [8, 128] with step size 8, maximum number of interaction bins in [4, 32] with step size 8, number of inner bags in {0, 10, 20}, validation_size uniformly in [0.1, 0.3], and min_samples_leaf in {2, 3, 4}. Please refer to `https://interpret.ml/docs/python/api/ExplainableBoostingClassifier.html` for parameter descriptions.

For NODE-GAM, I tuned the number layers in {2, 3}, number of trees in {5, 10, 50, 100, 200}, tree depth in {2, 3}, output dropout in [0, 0.9], last dropout in [0, 0.9],

colsample_bytree in [0.25, 0.5, 0.75, 1], and the ga2m indicator in {0, 1}. If ga2m indicator is 1, the model can also learn pairwise interactions. I also tuned the L1 and L2 regularisation coefficients applied to the last weight layer loguniformly in [-6, 4].

For XGB, I tuned the number of boosting rounds (trees) in [1, 1001] with step size 100, tree depth in [1, 10] with step size 1, learning rate loguniformly in [-4, -0.25], reg_alpha loguniformly in [-5, 1], reg_lambda loguniformly in [0, 1], gamma uniformly in [0, 100], min_child_weight uniformly in [0, 100], max_delta_step uniformly in [0, 10], subsample uniformly in [0.1, 1], and colsample_bytree uniformly in [0.1, 1]. Please refer to `https://xgboost.readthedocs.io/en/stable/parameter.html` for parameter descriptions.

For RF, I tuned tree depth in [2, 20] with step size 1, max_features uniformly in [0.01, 1], class_weight in {'balanced', None}, min_samples_split in [2, 10] with step size 1, min_samples_leaf in [1, 10] with step size 1. Please refer to `https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html` for parameter descriptions. Note that tuning tree depth is generally not necessary for RF models because bagging protects against overfitting. However, tuning depth seemed to yield better results in the data I used.

For the multilayer perceptron (MLP), I tuned the structure in {'128-128-64', '64-64-32', '1024', '64'}. I also separately tuned models with and without batch normalisation. If batch normalisation was not used, I tuned dropout uniformly in [0, 0.75]. I also tuned the L1 and L2 regularisation coefficients loguniformly in [-4, 4].

For all models implemented in the PyTorch package (PLR, SNAM, NODE-GAM, MLP) I tuned the learning rate loguniformly in [-5, -1], batch size in {128, 1024}, and data preprocessing in {'logrobust', 'norm'} where logrobust means "log1p transformation followed by robust scaling" and "norm" means quantile transformation to Gaussian distribution. During experimentation, I found that the models generally performed better when data transformations were applied. The log-transformation made the distribution of blood tests less skewed, and the robust scaling (subtracting median and dividing by interquartile range) made the scaling more robust to outliers.

Quantile transformation was considered because the authors of the NODE-GAM ML model found it to be useful [164].

References

[1] John McManigle, Sam Evans, and Keith A. Gillow. *OxThesis, a LaTeX template for an Oxford University thesis*. Accessed in January 2024. URL: https://github.com/mcmanigle/OxThesis.

[2] Cancer Research UK. *Bowel Cancer Incidence*. 2024. URL: https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/bowel-cancer (visited on 09/17/2024).

[3] Hyuna Sung et al. "Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries". In: *CA: a cancer journal for clinicians* 71.3 (2021), pp. 209–249.

[4] Andres Tamm et al. "Establishing a colorectal cancer research database from routinely collected health data: the process and potential from a pilot study". In: *BMJ Health & Care Informatics* 29.1 (2022).

[5] Eileen Morgan et al. "Global burden of colorectal cancer in 2020 and 2040: incidence and mortality estimates from GLOBOCAN". In: *Gut* 72.2 (2023), pp. 338–344.

[6] National Cancer Registration and Analysis Service (NCRAS). *Cancer Survival in England: adult, stage at diagnosis, childhood and geographical patterns*. 2024. URL: https://www.cancerdata.nhs.uk/survival/cancersurvivalengland (visited on 10/01/2024).

[7] National Institute for Health and Care Excellence. *Suspected cancer: recognition and referral. NICE Guideline [NG12]*. https://www.nice.org.uk/guidance/ng12. 2015.

[8] National Disease Registration Service (NDRS). *Routes to diagnosis*. 2024. URL: https://nhsd-ndrs.shinyapps.io/routes_to_diagnosis (visited on 10/01/2024).

[9] World Cancer Research Fund/American Institute for Cancer Research. *Continuous Update Project Expert Report 2018. Diet, nutrition, physical activity and colorectal cancer*. 2018. URL: https://www.wcrf.org/wp-content/uploads/2021/02/Colorectal-cancer-report.pdf (visited on 10/01/2024).

[10] Caitlin C Murphy and Timothy A Zaki. "Changing epidemiology of colorectal cancer—birth cohort effects and emerging risk factors". In: *Nature reviews Gastroenterology & hepatology* 21.1 (2024), pp. 25–34.

[11] Evelien Dekker et al. "Colorectal cancer". In: *The Lancet* 394.10207 (2019), pp. 1467–1480.

[12] James E East et al. “British Society of Gastroenterology position statement on serrated polyps in the colon and rectum”. In: *Gut* 66.7 (2017), pp. 1181–1196.

[13] Mark Schmitt and Florian R Greten. “The inflammatory pathogenesis of colorectal cancer”. In: *Nature Reviews Immunology* 21.10 (2021), pp. 653–667.

[14] National Audit Office. *Digital transformation in the NHS*. Tech. rep. Department of Health and Social Care, NHS England & NHS Improvement, NHS Digital, May 2020.

[15] Health and Social Care Committee. *Digital transformation in the NHS, Eighth Report of Session 2022–23*. Tech. rep. House of Commons, June 2023.

[16] NHS Digital. *90% of NHS trusts now have electronic patient records*. Accessed in January 2024. 2023. URL: https://digital.nhs.uk/news/2023/90-of-nhs-trusts-now-have-electronic-patient-records.

[17] Fang Liu and Demosthenes Panagiotakos. “Real-world data: a brief review of the methods, applications, challenges and opportunities”. In: *BMC Medical Research Methodology* 22.1 (2022), p. 287.

[18] National Institute for Health and Care Excellence. *NICE strategy 2021 to 2026: Dynamic, Collaborative, Excellent*. Apr. 2021. URL: https://www.nice.org.uk/about/who-we-are/corporate-publications/the-nice-strategy-2021-to-2026.

[19] National Institute for Health and Care Excellence. *NICE real-world evidence framework. Corporate document [ECD9]*. Accessed in January 2024. 2022. URL: https://www.nice.org.uk/corporate/ecd9.

[20] Alan Leviton and Tobias Loddenkemper. “Design, implementation, and inferential issues associated with clinical trials that rely on data in electronic medical records: a narrative review”. In: *BMC Medical Research Methodology* 23.1 (2023), p. 271.

[21] Christopher M Sauer et al. “Leveraging electronic health records for data science: common pitfalls and how to avoid them”. In: *The Lancet Digital Health* 4.12 (2022), e893–e898.

[22] Abigail E Lewis et al. “Electronic health record data quality assessment and tools: a systematic review”. In: *Journal of the American Medical Informatics Association* 30.10 (2023), pp. 1730–1740.

[23] Joanne Enticott et al. “Leaders’ perspectives on learning health systems: a qualitative study”. In: *BMC Health Services Research* 20 (2020), pp. 1–13.

[24] Olatunde O Madandola et al. “The relationship between electronic health records user interface features and data quality of patient clinical information: an integrative review”. In: *Journal of the American Medical Informatics Association* 31.1 (2024), pp. 240–255.

[25] Neil D Lawrence. “Data readiness levels”. In: *arXiv preprint arXiv:1705.02245* (2017).

[26] T Phuong Quan et al. “Health record hiccups—5,526 real-world time series with change points labelled by crowdsourced visual inspection”. In: *GigaScience* 12 (2023), giad060.

[27] T Phuong Quan. “daiquiri: data quality reporting for temporal datasets”. In: *Journal of Open Source Software* 7.80 (2022).

[28] Tom Powell. *The structure of the NHS in England. House of Commons Library research briefing*. July 10, 2023, pp. 9–12, 42–45. URL: `https://researchbriefings.files.parliament.uk/documents/CBP-7206/CBP-7206.pdf` (visited on 10/20/2024).

[29] Leigh R Warren et al. "Improving data sharing between acute hospitals in England: an overview of health record system distribution and retrospective observational analysis of inter-hospital transitions of care". In: *BMJ open* 9.12 (2019), e031637.

[30] Anmol Arora et al. "The value of standards for health datasets in artificial intelligence-based applications". In: *Nature Medicine* (2023), pp. 1–10.

[31] Vivian L West, David Borland, and W Ed Hammond. "Innovative information visualization of electronic health record data: a systematic review". In: *Journal of the American Medical Informatics Association* 22.2 (2015), pp. 330–339.

[32] Taowei David Wang et al. "Aligning temporal data by sentinel events: discovering patterns in electronic health records". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2008, pp. 457–466.

[33] Jan Piasecki et al. "Ethical issues in biomedical research using electronic health records: a systematic review". In: *Medicine, Health Care and Philosophy* 24.4 (2021), pp. 633–658.

[34] Kathleen Murphy et al. "Artificial intelligence for good health: a scoping review of the ethics literature". In: *BMC medical ethics* 22.1 (2021), pp. 1–17.

[35] Rebecca L Siegel et al. "Colorectal cancer statistics, 2023". In: *CA: a cancer journal for clinicians* 73.3 (2023), pp. 233–254.

[36] Raanan Gillon. "Medical ethics: four principles plus attention to scope". In: *Bmj* 309.6948 (1994), p. 184.

[37] Benjamin Kompa, Jasper Snoek, and Andrew L Beam. "Second opinion needed: communicating uncertainty in medical machine learning". In: *NPJ Digital Medicine* 4.1 (2021), p. 4.

[38] Ben Van Calster et al. "Calibration: the Achilles heel of predictive analytics". In: *BMC medicine* 17.1 (2019), pp. 1–7.

[39] Alex Kendall and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems* 30 (2017).

[40] Angeliki Kerasidou and Charalampia Kerasidou. "Data-driven research and healthcare: public trust, data governance and the NHS". In: *BMC medical ethics* 24.1 (2023), p. 51.

[41] Donald M Parkin. "The evolution of the population-based cancer registry". In: *Nature Reviews Cancer* 6.8 (2006), pp. 603–612.

[42] Katherine E Henson et al. "Data resource profile: national cancer registration dataset in England". In: *International journal of epidemiology* 49.1 (2020), 16–16h.

[43] *National disease registration service (NDRS)*. 2024. URL: `https://digital.nhs.uk/services/national-disease-registration-service` (visited on 10/14/2024).

[44] National Disease Registration Service. *Cancer outcomes and services dataset (COSD)*. Oct. 8, 2024. URL: https://digital.nhs.uk/ndrs/data/data-sets/cosd (visited on 10/14/2024).

[45] Sabrina Sandhu et al. "Cohort profile: radiotherapy dataset (RTDS) in England". In: *BMJ open* 13.6 (2023), e070699.

[46] National Disease Registration Service. *Radiotherapy data set (RTDS)*. Aug. 15, 2024. URL: https://digital.nhs.uk/ndrs/data/data-sets/rtds (visited on 10/14/2024).

[47] National Disease Registration Service. *NRDS Systemic anti-cancer therapy data set (SACT)*. June 11, 2024. URL: https://digital.nhs.uk/services/data-access-request-service-dars/dars-products-and-services/data-set-catalogue/ndrs-systemic-anti-cancer-therapy-data-set-sact (visited on 10/14/2024).

[48] NHS Digital. *Hospital episode statistics (HES)*. Aug. 29, 2024. URL: https://digital.nhs.uk/data-and-information/data-tools-and-services/data-services/hospital-episode-statistics (visited on 10/14/2024).

[49] Amy Downing et al. "Data resource profile: the COloRECTal cancer data repository (CORECT-R)". In: *International Journal of Epidemiology* 50.5 (2021), 1418–1418k.

[50] Public Health Scotland. *Northern Ireland Cancer Registry*. 2024. URL: https://www.qub.ac.uk/research-centres/nicr/AboutUs/Registry/ (visited on 10/14/2024).

[51] Public Health Scotland. *Scottish Cancer Registry (SMR06)*. Sept. 27, 2024. URL: https://web.www.healthdatagateway.org/dataset/fa45c071-1852-4c02-bddc-fda4517bffa8 (visited on 10/14/2024).

[52] Public Health Wales. *Welsh Cancer Intelligence and Surveillance Unit (WCISU)*. 2024. URL: https://phw.nhs.wales/services-and-teams/welsh-cancer-intelligence-and-surveillance-unit-wcisu/ (visited on 10/14/2024).

[53] Emily Herrett et al. "Data resource profile: clinical practice research datalink (CPRD)". In: *International journal of epidemiology* 44.3 (2015), pp. 827–836.

[54] Elizabeth J Williamson et al. "Factors associated with COVID-19-related death using OpenSAFELY". In: *Nature* 584.7821 (2020), pp. 430–436.

[55] Kerina H Jones et al. "A profile of the SAIL databank on the UK secure research platform". In: *International journal of population data science* 4.2 (2019).

[56] Elizabeth Ford et al. "Extracting information from the text of electronic medical records to improve case detection: a systematic review". In: *Journal of the American Medical Informatics Association* 23.5 (2016), pp. 1007–1015.

[57] Henk Harkema et al. "ConText: an algorithm for determining negation, experiencer, and temporal status from clinical reports". In: *Journal of biomedical informatics* 42.5 (2009), pp. 839–851.

[58] James D Brierley, Mary K Gospodarowicz, and Christian Wittekind. *TNM classification of malignant tumours*. John Wiley & Sons, 2017.

[59] Sajjad Abedian et al. “Automated extraction of tumor staging and diagnosis information from surgical pathology reports”. In: *JCO Clinical Cancer Informatics* 5 (2021), pp. 1054–1061.

[60] Anobel Y Odisho et al. “Natural language processing systems for pathology parsing in limited data environments with uncertainty estimation”. In: *JAMIA open* 3.3 (2020), pp. 431–438.

[61] Leonard W D’Avolio et al. “Facilitating clinical outcomes assessment through the automated identification of quality measures for prostate cancer surgery”. In: *Journal of the American Medical Informatics Association* 15.3 (2008), pp. 341–348.

[62] Weill Cornell Medicine Research Informatics. *PathExtractor*. Accessed on June 27, 2023. URL: `https://github.com/wcmc-research-informatics/PathExtractor/blob/master/resources/tnmNumericValue.regex`.

[63] Anobel Y Odisho and Briton Park. *Prostate-pathology-parser*. Accessed on June 27, 2023. URL: `https://github.com/ucsfurology/Prostate-Pathology-Parser`.

[64] Marie Ansoborlo et al. “Prescreening in oncology trials using medical records. Natural language processing applied on lung cancer multidisciplinary team meeting reports”. In: *Health Informatics Journal* 29.1 (2023), p. 14604582221146709.

[65] Richard C Khor et al. “Extracting tumour prognostic factors from a diverse electronic record dataset in genito-urinary oncology”. In: *International journal of medical informatics* 121 (2019), pp. 53–57.

[66] Nektarios Ladas et al. “Programming techniques for improving rule readability for rule-based information extraction natural language processing pipelines of unstructured and semi-structured medical texts”. In: *Health Informatics Journal* 29.2 (2023), p. 14604582231164696.

[67] Honghong Huang et al. “Natural language processing in urology: Automated extraction of clinical information from histopathology reports of uro-oncology procedures”. In: *Heliyon* 9.4 (2023).

[68] Abdulrahman K AAlAbdulsalam et al. “Automated extraction and classification of cancer stage mentions fromunstructured text fields in a central cancer registry”. In: *AMIA Summits on Translational Science Proceedings* 2018 (2018), p. 16.

[69] Brian J Kim et al. “Second prize: A natural language processing program effectively extracts key pathologic findings from radical prostatectomy reports”. In: *Journal of endourology* 28.12 (2014), pp. 1474–1478.

[70] Naveen Ashish, Lisa Dahm, and Charles Boicey. “University of California, Irvine–Pathology Extraction Pipeline: The pathology extraction pipeline for information extraction from pathology reports”. In: *Health informatics journal* 20.4 (2014), pp. 288–305.

[71] Kilian GM Brown and Cherry E Koh. “Surgical management of recurrent colon cancer”. In: *Journal of gastrointestinal oncology* 11.3 (2020), p. 513.

[72] Viet H Le et al. “Metachronous peritoneal metastases following curative resection for colon cancer: Understanding risk factors and patterns of recurrence”. In: *Journal of surgical oncology* 123.2 (2021), pp. 622–629.

[73] Timothy L Lash et al. “A validated algorithm to ascertain colorectal cancer recurrence using registry resources in D enmark”. In: *International journal of cancer* 136.9 (2015), pp. 2210–2215.

[74] Jennie Engstrand et al. “Synchronous and metachronous liver metastases in patients with colorectal cancer—towards a clinically relevant definition”. In: *World Journal of Surgical Oncology* 17.1 (2019), pp. 1–10.

[75] Linda Aagaard Rasmussen et al. “A validated algorithm for register-based identification of patients with recurrence of breast cancer—Based on Danish Breast Cancer Group (DBCG) data”. In: *Cancer epidemiology* 59 (2019), pp. 129–134.

[76] Linda Aagaard Rasmussen et al. “Time from incident primary cancer until recurrence or second primary cancer: risk factors and impact in general practice”. In: *European journal of cancer care* 28.5 (2019), e13123.

[77] Claire MB Holloway et al. “Identifying breast cancer recurrence in administrative data: Algorithm development and validation”. In: *Current Oncology* 29.8 (2022), pp. 5338–5367.

[78] Ekapob Sangariyavanich et al. “Systematic review of natural language processing for recurrent cancer detection from electronic medical records”. In: *Informatics in Medicine Unlocked* (2023), p. 101326.

[79] Justin A Strauss et al. “Identifying primary and recurrent cancers using a SAS-based natural language processing algorithm”. In: *Journal of the American Medical Informatics Association* 20.2 (2013), pp. 349–355.

[80] David S Carrell et al. “Using natural language processing to improve efficiency of manual chart abstraction in research: the case of breast cancer recurrence”. In: *American journal of epidemiology* 179.6 (2014), pp. 749–758.

[81] Zexian Zeng et al. “Using natural language processing and machine learning to identify breast cancer local recurrence”. In: *BMC bioinformatics* 19.17 (2018), pp. 65–74.

[82] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423.

[83] Ke Liu et al. “MetBERT: a generalizable and pre-trained deep learning model for the prediction of metastatic cancer from clinical notes”. In: *AMIA Annual Symposium Proceedings*. Vol. 2022. American Medical Informatics Association. 2022, p. 331.

[84] Karen E Batch et al. “Developing a cancer digital twin: Supervised metastases detection from consecutive structured radiology reports”. In: *Frontiers in artificial intelligence* 5 (2022), p. 826402.

[85] Wendy W Chapman et al. "A simple algorithm for identifying negated findings and diseases in discharge summaries". In: *Journal of biomedical informatics* 34.5 (2001), pp. 301–310.

[86] Betty Van Aken et al. "Assertion detection in clinical notes: Medical language models to the rescue?" In: *Proceedings of the Second Workshop on Natural Language Processing for Medical Conversations*. 2021, pp. 35–40.

[87] Brian Chapman et al. *pyConTextNLP*. Accessed on December 6, 2023. URL: https://github.com/chapmanbe/pyConTextNLP/tree/master/KB.

[88] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[89] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108* (2019).

[90] Emily Alsentzer et al. "Publicly available clinical BERT embeddings". In: *arXiv preprint arXiv:1904.03323* (2019).

[91] Emily Alsentzer. *ClinicalBERT - Bio + Clinical BERT Model*. Accessed on December 6, 2023. URL: https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT.

[92] Hugging Face developers. *DistilBertForTokenClassificationn*. URL: https://huggingface.co/docs/transformers/en/model_doc/distilbert#transformers.DistilBertForTokenClassification (visited on 10/10/2024).

[93] Edward J Hu et al. "Lora: Low-rank adaptation of large language models". In: *arXiv preprint arXiv:2106.09685* (2021).

[94] Da Yu et al. "Differentially private fine-tuning of language models". In: *arXiv preprint arXiv:2110.06500* (2021).

[95] RStudio Team. *Shiny from RStudio*. https://shiny.rstudio.com/. Accessed: 2021-08-04.

[96] Tobias Schlosser et al. "A Consolidated Overview of Evaluation and Performance Metrics for Machine Learning and Computer Vision". In: (2023).

[97] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. URL: https://doi.org/10.1038/s41586-020-2649-2.

[98] The pandas development team. *pandas-dev/pandas: Pandas*. Version 1.4.3. Feb. 2020. URL: https://doi.org/10.5281/zenodo.3509134.

[99] Matthew Barnett and Ma Lin. *regex: Alternative regular expression module, to replace re.* Version 2020.10.15. 2020. URL: https://github.com/mrabarnett/mrab-regex.

[100] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[101] Thomas Wolf et al. "Huggingface's transformers: State-of-the-art natural language processing". In: *arXiv preprint arXiv:1910.03771* (2019).

[102] Sourab Mangrulkar et al. *PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods*. https://github.com/huggingface/peft. 2022.

[103] *HuggingFace*. Accessed in December 2023. URL: https://huggingface.co/.

[104] Isabelle Soerjomataram et al. "CanStaging+: an electronic staging tool for population-based cancer registries". In: *The Lancet Oncology* 22.8 (2021), p. 1069.

[105] *Assertion detection tool*. Accessed in December 2023. 2021. URL: https://ehr-assertion-detection.demo.datexis.com/.

[106] Peter Norvig. *How to write a spelling corrector*. https://norvig.com/spell-correct.html. Accessed: 2023-10-10.

[107] Thomas Searle et al. "MedCATTrainer: a biomedical free text annotation interface with active learning and research use case specific customisation". In: *arXiv preprint arXiv:1907.07322* (2019).

[108] *Standards and datasets for reporting cancers: Dataset for histopathological reporting of colorectal cancer*. Accessed in December 2023. URL: https://www.rcpath.org/resourceLibrary/g049-dataset-for-histopathological-reporting-of-colorectal-cancer.html.

[109] Daniel Jurafsky and James H Martin. "Speech and Language Processing". In: 3rd ed. Online draft. 7 January 2023. Chap. Relation and event extraction. URL: https://web.stanford.edu/~jurafsky/slp3/21.pdf.

[110] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[111] Albert Q Jiang et al. "Mistral 7B". In: *arXiv preprint arXiv:2310.06825* (2023).

[112] OpenAI et al. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].

[113] Harsha Nori et al. "Capabilities of gpt-4 on medical challenge problems". In: *arXiv preprint arXiv:2303.13375* (2023).

[114] Department of Health and Social Care. *Data saves lives: reshaping health and social care with data*. Accessed in December 2023. 2022. URL: https://www.gov.uk/government/publications/data-saves-lives-reshaping-health-and-social-care-with-data/data-saves-lives-reshaping-health-and-social-care-with-data.

[115] Kai Sheng Saw et al. "Faecal immunochemical test to triage patients with possible colorectal cancer symptoms: meta-analysis". In: *British Journal of Surgery* 109.2 (2022), znab411.

[116] Nigel D'Souza et al. "Faecal immunochemical test is superior to symptoms in predicting pathology in patients with suspected colorectal cancer symptoms referred on a 2WW pathway: a diagnostic accuracy study". In: *Gut* 70.6 (2021), pp. 1130–1138.

[117] *COLOFIT: Optimal use of Faecal Immunochemical Testing for patients with symptoms of possible colorectal cancer*. https://fundingawards.nihr.ac.uk/award/NIHR133852. Accessed: 2023-10-10.

[118] SER Bailey et al. "COLOFIT: Development and internal-external validation of models using age, sex, faecal immunochemical and blood tests to optimise diagnosis of colorectal cancer in symptomatic patients". In: *medRxiv* (2024), pp. 2024–03.

[119] Colin Crooks et al. *Validation and performance of models for predicting the risk of colorectal cancer incorporating FIT, age, sex and available blood indices using the Nottingham COLOFIT cohort*. personal communication.

[120] Richard D Riley et al. "Minimum sample size for external validation of a clinical prediction model with a binary outcome". In: *Statistics in Medicine* 40.19 (2021), pp. 4230–4251.

[121] Jesse Davis and Mark Goadrich. "The relationship between Precision-Recall and ROC curves". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 233–240.

[122] *Scikit-learn: Average precision score*. `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score`. Accessed: 2022-01-05.

[123] Andrew J Vickers and Elena B Elkin. "Decision curve analysis: a novel method for evaluating prediction models". In: *Medical Decision Making* 26.6 (2006), pp. 565–574.

[124] Carolyn Piggott et al. "Analytical evaluation of four faecal immunochemistry tests for haemoglobin". In: *Clinical Chemistry and Laboratory Medicine (CCLM)* 59.1 (2021), pp. 173–178.

[125] Diana R Withrow et al. "Combining faecal immunochemical testing with blood test results to identify patients with symptoms at risk of colorectal cancer: a consecutive cohort of 16,604 patients tested in primary care". In: (). Manuscript submitted for publication.

[126] Samuel Von Wilson et al. *miceforest: Fast, Memory Efficient Imputation with LightGBM*. Version 5.6.1. 2022. URL: `https://github.com/AnotherSamWilson/miceforest`.

[127] National Institute for Health and Care Excellence. *Quantitative faecal immunochemical testing to guide colorectal cancer pathway referral in primary care. Diagnostics guidance [DG56]*. `https://www.nice.org.uk/guidance/dg56`. 2023.

[128] Kevin J Monahan et al. "Faecal immunochemical testing (FIT) in patients with signs or symptoms of suspected colorectal cancer (CRC): a joint guideline from the Association of Coloproctology of Great Britain and Ireland (ACPGBI) and the British Society of Gastroenterology (BSG)". In: *Gut* 71.10 (2022), pp. 1939–1962.

[129] Alexandru Niculescu-Mizil and Rich Caruana. "Predicting good probabilities with supervised learning". In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 625–632.

[130] Peter C Austin and Ewout W Steyerberg. "Graphical assessment of internal and external calibration of logistic regression models by using loess smoothers". In: *Statistics in medicine* 33.3 (2014), pp. 517–535.

[131] Jonathan W Bartlett and Rachael A Hughes. "Bootstrap inference for multiple imputation under uncongeniality and misspecification". In: *Statistical methods in medical research* 29.12 (2020), pp. 3533–3546.

[132] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[133] Skipper Seabold and Josef Perktold. "Statsmodels: Econometric and statistical modeling with python". In: *Proceedings of the 9th Python in Science Conference*. Vol. 57. 61. Austin, TX. 2010, pp. 10–25080.

[134] Shaun Porwal and Rohan Singh. *A Python package for Decision Curve Analysis to evaluate prediction models, molecular markers, and diagnostic tests*. Version 1.0.6. 2022. URL: `https://github.com/MSKCC-Epi-Bio/dcurves/`.

[135] Stef van Buuren and Karin Groothuis-Oudshoorn. "mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software* 45.3 (2011), pp. 1–67.

[136] Jan Grau, Ivo Grosse, and Jens Keilwagen. "PRROC: computing and visualizing precision-recall and receiver operating characteristic curves in R". In: *Bioinformatics* 31.15 (2015), pp. 2595–2597.

[137] Frank E Harrell Jr. *rms: Regression Modeling Strategies*. R package version 6.2-0. 2021. URL: `https://CRAN.R-project.org/package=rms`.

[138] Daniel D. Sjoberg. *dcurves: Decision Curve Analysis for Model Evaluation*. https://github.com/ddsjoberg/dcurves, https://www.danieldsjoberg.com/dcurves/. 2022.

[139] Michael F Bath et al. "Faecal immunochemical testing for haemoglobin in detecting bowel polyps in symptomatic patients: multicentre prospective cohort study". In: *BJS open* 7.2 (2023), zrac161.

[140] Office for National Statistics. *Exploring local income deprivation: A detailed picture of disparities within English local authorities to a neighbourhood level*. May 24, 2021. URL: `https://www.ons.gov.uk/visualisations/dvc1371/#/E06000018` (visited on 10/31/2024).

[141] Thames Valley Cancer Alliance (TVCA). *Primary Care Guidance: Lower GI two-week wait pathway during COVID-19*. July 2020.

[142] Patrick Royston and Douglas G Altman. "Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling". In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 43.3 (1994), pp. 429–453.

[143] Tal Yarkoni and Jacob Westfall. "Choosing prediction over explanation in psychology: Lessons from machine learning". In: *Perspectives on Psychological Science* 12.6 (2017), pp. 1100–1122.

[144] Gary S Collins et al. "Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD) the TRIPOD statement". In: *Circulation* 131.2 (2015), pp. 211–219.

[145] Matthew Sperrin et al. "Targeted validation: validating clinical prediction models in their intended population and setting". In: *Diagnostic and Prognostic Research* 6.1 (2022), p. 24.

[146] Joseph Futoma et al. "The myth of generalisability in clinical research and machine learning in health care". In: *The Lancet Digital Health* 2.9 (2020), e489–e492.

[147] Joaquín Cubiella et al. "The fecal hemoglobin concentration, age and sex test score: development and external validation of a simple prediction tool for colorectal cancer detection in symptomatic patients". In: *International journal of cancer* 140.10 (2017), pp. 2201–2211.

[148] Jayne Digby et al. "Appraisal of the faecal haemoglobin, age and sex test (FAST) score in assessment of patients with lower bowel symptoms: an observational study". In: *BMC gastroenterology* 19.1 (2019), pp. 1–7.

[149] Rigers Cama et al. "Evaluation of the FAST score in patients with suspected colorectal cancer in the Herts Valley CCG". In: (2022).

[150] Jesús-Miguel Herrero et al. "Symptom or faecal immunochemical test based referral criteria for colorectal cancer detection in symptomatic patients: a diagnostic tests study". In: *BMC gastroenterology* 18.1 (2018), pp. 1–10.

[151] Fernando Fernandez-Banares et al. "Prediction of advanced colonic neoplasm in symptomatic patients: a scoring system to prioritize colonoscopy (COLONOFIT study)". In: *BMC cancer* 19.1 (2019), pp. 1–12.

[152] Yaron Kinar et al. "Development and validation of a predictive model for detection of colorectal cancer in primary care by analysis of complete blood counts: a binational retrospective study". In: *Journal of the American Medical Informatics Association* 23.5 (2016), pp. 879–890.

[153] Jacqueline Birks et al. "Evaluation of a prediction model for colorectal cancer: retrospective analysis of 2.5 million patient records". In: *Cancer medicine* 6.10 (2017), pp. 2453–2460.

[154] Pradeep S Virdee et al. "Full blood count trends for colorectal cancer detection in primary care: development and validation of a dynamic prediction model". In: *Cancers* 14.19 (2022), p. 4779.

[155] Jennifer Anne Cooper et al. "Risk-adjusted colorectal cancer screening using the FIT and routine screening data: development of a risk prediction model". In: *British journal of cancer* 118.2 (2018), pp. 285–293.

[156] Jennifer Anne Cooper et al. "The use of electronic healthcare records for colorectal cancer screening referral decisions and risk prediction model development". In: *BMC gastroenterology* 20 (2020), pp. 1–16.

[157] Bruce Burnett et al. "Machine Learning in Colorectal Cancer Risk Prediction from Routinely Collected Data: A Review". In: *Diagnostics* 13.2 (2023), p. 301.

[158] Joaquín Cubiella et al. "Development and external validation of a faecal immunochemical test-based prediction model for colorectal cancer detection in symptomatic patients". In: *BMC medicine* 14.1 (2016), pp. 1–13.

[159] Julia Hippisley-Cox and Carol Coupland. "Identifying patients with suspected colorectal cancer in primary care: derivation and validation of an algorithm". In: *British Journal of General Practice* 62.594 (2012), e29–e37.

[160] Gareth James et al. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013.

[161] T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1990. URL: `https://books.google.co.uk/books?id=qa29r1Ze1coC`.

[162] Shiyun Xu et al. “Sparse neural additive model: Interpretable deep learning with feature selection via group sparsity”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2023, pp. 343–359.

[163] Yin Lou et al. “Accurate intelligible models with pairwise interactions”. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013, pp. 623–631.

[164] Chun-Hao Chang, Rich Caruana, and Anna Goldenberg. “NODE-GAM: Neural Generalized Additive Model for Interpretable Deep Learning”. In: *International Conference on Learning Representations*. 2021.

[165] Rich Caruana et al. “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission”. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 1721–1730.

[166] Leo Breiman. “Random forests”. In: *Machine learning* 45 (2001), pp. 5–32.

[167] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[168] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[169] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep feedforward networks*. MIT press, 2016. Chap. 6.

[170] Zhuoning Yuan et al. “Large-scale robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3040–3049.

[171] Zhuoning Yuan et al. “Compositional training for end-to-end deep AUC maximization”. In: *International Conference on Learning Representations*. 2021.

[172] Qi Qi et al. “Stochastic optimization of areas under precision-recall curves with provable convergence”. In: *Advances in neural information processing systems* 34 (2021), pp. 1752–1765.

[173] James Bergstra, Daniel Yamins, and David Cox. “Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures”. In: *International conference on machine learning*. PMLR. 2013, pp. 115–123.

[174] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Regularization for deep learning*. MIT press, 2016. Chap. 7.

[175] Harsha Nori et al. “Interpretml: A unified framework for machine learning interpretability”. In: *arXiv preprint arXiv:1909.09223* (2019).

[176] Chun-Hao Chang. *NODE GAM: Differentiable Generalized Additive Model for Interpretable Deep Learning*. Version 0.3.0. 2022. URL: `https://github.com/zzzace2000/nodegam`.

[177] Zhuoning Yuan et al. “LibAUC: A Deep Learning Library for X-risk Optimization”. In: *29th SIGKDD Conference on Knowledge Discovery and Data Mining*. 2023.

[178] Richard Liaw et al. "Tune: A Research Platform for Distributed Model Selection and Training". In: *arXiv preprint arXiv:1807.05118* (2018).

[179] Michael Nielsen. *visual proof that neural nets can compute any function.* Determination press, 2015. Chap. 4. URL: http://neuralnetworksanddeeplearning.com/chap4.html.

[180] Fuqiang Zhao et al. "Efficacy of cell-free DNA methylation-based blood test for colorectal cancer screening in high-risk population: a prospective cohort study". In: *Molecular Cancer* 22.1 (2023), p. 157.

[181] Brian D Nicholson et al. "Multi-cancer early detection test in symptomatic patients referred for cancer investigation in England and Wales (SYMPLIFY): a large-scale, observational cohort study". In: *The Lancet Oncology* (2023).

[182] Stephen Bates, Trevor Hastie, and Robert Tibshirani. "Cross-validation: what does it estimate and how well does it do it?" In: *Journal of the American Statistical Association* (2023), pp. 1–12.

[183] Xavier Bouthillier et al. "Accounting for variance in machine learning benchmarks". In: *Proceedings of Machine Learning and Systems* 3 (2021), pp. 747–769.

[184] A Karpathy. *A Recipe for Training Neural Networks.* http://karpathy.github.io/2019/04/25/recipe/. Accessed: 2023-10-19. 2019.

[185] Abhimanyu Dubey, Filip Radenovic, and Dhruv Mahajan. "Scalable interpretability via polynomials". In: *Advances in neural information processing systems* 35 (2022), pp. 36748–36761.

[186] Filip Radenovic, Abhimanyu Dubey, and Dhruv Mahajan. "Neural basis models for interpretability". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 8414–8426.

[187] Alistair EW Johnson et al. "MIMIC-III, a freely accessible critical care database". In: *Scientific data* 3.1 (2016), pp. 1–9.

[188] Chitta Ranjan, Samaneh Ebrahimi, and Kamran Paynabar. "Sequence graph transform (SGT): a feature embedding function for sequence data mining". In: *Data Mining and Knowledge Discovery* 36.2 (2022), pp. 668–708.

[189] International Health Terminology Standards Development Organisation. *SNOMED CT starter guide.* Accessed in December 2023. URL: http://snomed.org/sg.

[190] Hugo De Oliveira et al. "Explaining predictive factors in patient pathways using autoencoders". In: *Plos one* 17.11 (2022), e0277135.

[191] Daniel Jurafsky and James H Martin. *Speech and Language Processing.* 3rd ed. Online draft. 7 January 2023. URL: https://web.stanford.edu/~jurafsky/slp3/21.pdf.

[192] NHS England. *Widespread clinical support for reforming NHS cancer standards to speed up diagnosis for patients.* Accessed in January 2024. URL: https://www.england.nhs.uk/2023/08/widespread-clinical-support-for-reforming-nhs-cancer-standards-to-speed-up-diagnosis-for-patients/.

[193] Anni Järvelin, Antti Järvelin, and Kalervo Järvelin. "s-grams: Defining generalized n-grams for information retrieval". In: *Information Processing & Management* 43.4 (2007), pp. 1005–1019.

[194] Minseok Song, Christian W Günther, and Wil MP Van der Aalst. "Trace clustering in process mining". In: *Business Process Management Workshops: BPM 2008 International Workshops, Milano, Italy, September 1-4, 2008. Revised Papers 6*. Springer. 2009, pp. 109–120.

[195] Isotta Landi et al. "Deep representation learning of electronic health records to unlock patient stratification at scale". In: *NPJ digital medicine* 3.1 (2020), p. 96.

[196] Xianlong Zeng, Simon Lin, and Chang Liu. "Transformer-based unsupervised patient representation learning based on medical claims for risk stratification and analysis". In: *Proceedings of the 12th ACM conference on bioinformatics, computational biology, and health informatics*. 2021, pp. 1–9.

[197] Zeljko Kraljevic et al. "Foresight-Generative Pretrained Transformer (GPT) for Modelling of Patient Timelines using EHRs". In: ().

[198] Ran Xu et al. "Hypergraph Transformers for EHR-based Clinical Predictions". In: *AMIA Summits on Translational Science Proceedings* 2023 (2023), p. 582.

[199] Feng Xie et al. "Deep learning for temporal data representation in electronic health records: A systematic review of challenges and methodologies". In: *Journal of biomedical informatics* 126 (2022), p. 103980.

[200] Zeljko Kraljevic et al. "Multi-domain clinical natural language processing with medcat: the medical concept annotation toolkit". In: *Artificial Intelligence in Medicine* 117 (2021), p. 102083.

[201] Martin Grohe. "word2vec, node2vec, graph2vec, x2vec: Towards a theory of vector embeddings of structured data". In: *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 2020, pp. 1–16.

[202] Aditya Grover and Jure Leskovec. "node2vec: Scalable feature learning for networks". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 855–864.

[203] Emma Aspland et al. "Modified Needleman–Wunsch algorithm for clinical pathway clustering". In: *Journal of Biomedical Informatics* 115 (2021), p. 103668.

[204] Chitta Ranjan. *Sequence Graph Transform (SGT) — Sequence Embedding for Clustering, Classification, and Search*. Accessed in December 2023. URL: `https://github.com/cran2367/sgt`.

[205] Yingfan Wang et al. "Understanding how dimension reduction tools work: an empirical approach to deciphering t-SNE, UMAP, TriMAP, and PaCMAP for data visualization". In: *arXiv preprint arXiv:2012.04456* (2020).

[206] Leland McInnes, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering". In: *The Journal of Open Source Software* 2.11 (Mar. 2017).

[207] Dash development team. *Dash*. Accessed in December 2023. URL: `https://github.com/plotly/dash`.

[208] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: `https://plot.ly`.

[209] *scikit-learn 1.3.2: Clustering performance evaluation*. Accessed in December 2023. URL: https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation.

[210] Henrique Aguiar et al. "Learning of cluster-based feature importance for electronic health record time-series". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 161–179.

[211] UK Colorectal Cancer Intelligence Hub. *CORECT-R data catalogue v1.0*. Sept. 2020. URL: https://digital.nhs.uk/data-and-information/data-tools-and-services/data-services/hospital-episode-statistics (visited on 10/14/2024).

[212] National Institute for Health and Care Excellence (NICE). *2022 exceptional surveillance of suspected cancer: recognition and referral (NICE guideline NG12) and quantitative faecal immunochemical tests to guide referral for colorectal cancer in primary care (NICE diagnostics guidance 30)*. June 30, 2022. URL: https://www.nice.org.uk/guidance/ng12/resources/2022-exceptional-surveillance-of-suspected-cancer-recognition-and-referral-nice-guideline-ng12-and-quantitative-faecal-immunochemical-tests-to-guide-referral-for-colorectal-cancer-in-primary-care-nic-11132498701/chapter/Surveillance-decision?tab=evidence.

[213] Pradeep S Virdee et al. "BLOod Test Trend for cancEr Detection (BLOTTED): protocol for an observational and prediction model development study using English primary care electronic health record data". In: *Diagnostic and Prognostic Research* 7.1 (2023), p. 1.

[214] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. "Regularization paths for generalized linear models via coordinate descent". In: *Journal of statistical software* 33.1 (2010), p. 1.

[215] Rishabh Agarwal et al. "Neural additive models: Interpretable machine learning with neural nets". In: *Advances in neural information processing systems* 34 (2021), pp. 4699–4711.

[216] Sergei Popov, Stanislav Morozov, and Artem Babenko. "Neural Oblivious Decision Ensembles for Deep Learning on Tabular Data". In: *International Conference on Learning Representations*.

[217] Yin Lou, Rich Caruana, and Johannes Gehrke. "Intelligible models for classification and regression". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2012, pp. 150–158.

[218] xgboost developers. *XGBoost python package: Python API reference: get_score*. URL: https://xgboost.readthedocs.io/en/stable/python/python_api.html (visited on 10/14/2024).

[219] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009.

[220] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Optimization for training deep models*. MIT press, 2016. Chap. 8.

[221] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[222] Sergey Ioffe. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[223] Xiang Li et al. "Understanding the disharmony between dropout and batch normalization by variance shift". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2682–2690.

[224] PyTorch development team. *BCEWithLogitsLoss*. URL: `https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html` (visited on 10/10/2024).

[225] Gregory Gundersen. *The log-sum-exp trick*. Oct. 9, 2020. URL: `https://gregorygundersen.com/blog/2020/02/09/log-sum-exp` (visited on 10/10/2024).

[226] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern recognition letters* 27.8 (2006), pp. 861–874.

[227] Stéphan Clémençon, Gábor Lugosi, and Nicolas Vayatis. "Ranking and empirical minimization of U-statistics". In: (2008).

[228] Josh Achiam et al. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023).

[229] Gemini Team et al. "Gemini: a family of highly capable multimodal models". In: *arXiv preprint arXiv:2312.11805* (2023).

[230] Saidul Islam et al. "A comprehensive survey on applications of transformers for deep learning tasks". In: *Expert Systems with Applications* (2023), p. 122666.

[231] Andrej Karpathy. *Let's build GPT: from scratch, in code, spelled out*. 2023. URL: `https://www.youtube.com/watch?v=kCc8FmEb1nY` (visited on 10/10/2024).

[232] Geoffrey Hinton. "Distilling the Knowledge in a Neural Network". In: *arXiv preprint arXiv:1503.02531* (2015).

[233] Jinhyuk Lee et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4 (2020), pp. 1234–1240.