AALBORG UNIVERSITY

STUDENT REPORT

Master Thesis
Mathematics-economics, Financial Engineering

Predicting HFE C282Y Homozygosity from Blood Biomarkers

A Cost-Efficient Machine Learning Approach

21st May 2026

Dept. of Mathematical Sciences
Thomas Manns vej 23
DK-9220 Aalborg Ø
http://math.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Predicting HFE C282Y Homozygosity from Blood Biomarkers

Theme:
A Cost-Efficient Machine Learning Approach

Project Period:
Spring Semester 2026

Project Group:
MO10-06

Author(s):
Aron Reinholdt Eg
Magnus Bredvig

Supervisor(s):
Christophe Biscio & Søren Byg Vilsen

Copies: 1

Numbered Pages: 52

Date of Completion:
21st May 2026

Abstract:

This study develops a framework to assess pre-screening for HFE C282Y homozygosity using blood-based biomarkers. Due to temporary inaccessibility of the UK Biobank (UKB), analyses were conducted on synthetic data with 63 features generated to reflect UKB distributions, with missing data handled using multiple imputation by chained equations.
The modelling pipeline combined imputation, bagging to address class imbalance, and predictive modelling, using logistic regression as a baseline and XGBoost as the primary method, evaluated on validation and test sets. A cost-sensitive framework, including heuristic optimization and a mixed-integer linear programming formulation, was applied to determine classification thresholds under sensitivity and cost constraints.
Model performance was near random, as the simulated data preserve marginal distributions but do not capture predictive relationships between biomarkers and genotype.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the authors.

Preface

This master thesis is written in the period from 1st of February 2026 to 21st May 2026 by MO10-06, attending 10th semester of Mathematics-economics, Financial Engineering at Aalborg University. The main theme of this report is to develop a machine learning model that predicts the risk of mutation in the HFE gene, specifically C282Y homozygous. Furthermore, the economic cost for using the model as a pre-screening tool is evaluated.

References in the project are in the Vancouver referencing style and are therefore written as

[source number, page number].

The master thesis has used generative AI for language improvement, including grammar. Furthermore, AI has been used for debugging, during the programming phase.
The group would like to thank our supervisors Christophe Biscio & Søren Byg Vilsen for great help and guidance.

Aron Reinholdt Eg

Magnus Bredvig

Contents

List of Figures **v**

List of Tables **v**

List of Algorithms **vi**

1 Introduction **1**

2 Gradient Boosting Models **3**
2.1 Gradient Boosting for Classification 3
2.2 Extreme Gradient Boosting 7

3 Construction of Cost Function **12**
3.1 Cost-Sensitive Decision Making 14
3.2 Heuristics 15

4 Linear Programming **17**
4.1 Linear programming 17
4.2 Mixed Integer Linear Programming 19

5 Missing data **24**
5.1 Multiple Imputation by Chained Equations 24

6 Application **27**
6.1 Data 27
6.2 Modelling 31
6.3 Results 36

7 Discussion **49**

8 Conclusion **52**

Appendices **53**

A Linear programming **55**
A.1 Simplex Example 55

B Application **59**
B.1 Modelling 59

C Mice imputations **61**
C.1 Feature combination - Full 61
C.2 Feature combination - Biochemistry Markers 62
C.3 Feature combination - Haematological Assays 63

References **64**

List of Figures

1.1 Prevalence of haemochromatosis among ethnicities. 2

2.1 Flowchart for gradient boosting algorithm. 5

4.1 Illustration of the branch-and-bound process. 21
4.2 Illustration of the integer programming problem. 22
4.3 Complete branching tree for the ILP. 23

6.1 Map of the UK showing UKB baseline assessment centres. 29
6.2 Percentage of `C282Y` homozygous within cohort and stratified by `Sex`. . . . 32
6.3 Overview of application approach. 33
6.4 Comparison of imputed features `Full` vs `Bio`. 37
6.5 Comparison of imputed features `Full` vs. `Haema` 37
6.6 AUC curves for logistic regression on feature combination `Full`. 40
6.7 AUC curves for XGBoost on feature combination `Full`. 41
6.8 AUC curves for logistic regression on feature combination `Bio`. 43
6.9 AUC curves for XGBoost on feature combination `Bio`. 44
6.10 AUC curves for logistic regression on feature combination `Haema`. 46
6.11 AUC curves for XGBoost on feature combination `Haema`. 47

List of Tables

3.1 Equal cost matrix per person. 12
3.2 Cost matrix per person. 14

6.1 Ethnic group distribution. 28
6.2 Feature overview. 30
6.3 Optimal XGBoost hyperparameters across imputations and feature sets. . . 38
6.4 Wilcoxon Rank Sum Test on feature combination `Full`. 39
6.5 Validation set results with optimal threshold `Full`. 40
6.6 Test set results with optimal threshold `Full`. 41
6.7 Test set results with cost-optimal threshold `Full`. 42
6.8 Wilcoxon Rank Sum Test on feature combination `Bio`. 42
6.9 Validation set results with optimal threshold `Bio`. 43
6.10 Test set results with optimal threshold `Bio`. 44
6.11 Test set results with cost-optimal threshold `Bio`. 45
6.12 Wilcoxon Rank Sum Test on feature combination `Haema`. 45
6.13 Validation set results with optimal threshold `Haema`. 46
6.14 Test set results with optimal threshold `Haema`. 47
6.15 Test set results with cost-optimal threshold `Haema`. 48

List of Algorithms

1 Gradient Boosting. 4
2 Exact Greedy algorithm for Split Finding. 9
3 Approximate Alghorithm for Split Finding. 9
4 Sparsity-Aware Split Finding. 11
5 Simplex Method. 18
6 Branch-and-Bound for MILP 20
7 Multiple Imputation by Chained Equations 26

1 | Introduction

Hereditary HFE haemochromatosis (HH) is one of the most common inherited metabolic disorders in populations of Northern European ancestry, as illustrated in Figure 1.1. It is characterised by increased intestinal absorption of dietary iron, which may lead to progressive iron accumulation in the body. The most prevalent genetic cause is the C282Y variant in the HFE gene, although other variants such as H63D and S65C also occur. Individuals may carry these variants as homozygotes, heterozygotes, or compound heterozygotes.

It is important to distinguish between genotype, iron overload, and clinical disease. C282Y homozygosity represents a genetic risk state, but not all individuals with this genotype develop clinically significant iron overload or complications. Clinical haemochromatosis is defined by the presence of iron overload with evidence of tissue involvement, rather than genotype alone. In clinical practice, iron overload is typically assessed using transferrin saturation (TS) and serum ferritin (SF), where elevated levels indicate increased iron stores. Genetic testing is used to confirm HFE variants when there is biochemical or clinical suspicion [31].

TS and SF are not included in standard blood panels and are typically measured only in specific clinical contexts where iron overload is suspected. As a result, early stages of iron accumulation or genetic susceptibility often remain undetected.

If untreated, iron accumulation can affect multiple organs, including the liver, pancreas, heart, and joints, potentially leading to liver disease, diabetes, cardiomyopathy, and arthropathy. However, the clinical penetrance of hereditary HFE haemochromatosis is incomplete and varies by sex and age. Symptoms typically appear after the age of 30 in men and after menopause in women. Common early manifestations include fatigue, joint pain, and nonspecific abdominal symptoms.

Epidemiological studies demonstrate that C282Y homozygotes have an increased risk of several morbidities, but the proportion who develop clinically significant disease remains limited. For example, data from the UK Biobank, a large general population study, show that C282Y homozygous individuals, particularly men, have higher lifetime risks of liver disease, diabetes, arthritis, and frailty compared to non-carriers [27, 33]. However, many individuals with this genotype do not develop clinically relevant iron overload.

In Denmark, approximately 0.4% of the population (1 in 250 individuals) are C282Y homozygotes, corresponding to around 20,000 individuals. Additionally, more than 13% carry at least one HFE variant. Similar prevalences are observed in other Northern European populations, including in UK Biobank data [24, 27]. Despite this relatively high genetic prevalence, only a subset of genetically predisposed individuals develop clinically significant disease.

Due to the variable penetrance and risk of overdiagnosis, population-level genetic screening for HH is not currently recommended in countries such as Denmark and the United Kingdom [37, 32]. Current practice instead relies on case finding through symptoms, abnormal iron biomarkers, or cascade testing among relatives.

This creates a diagnostic gap, i.e., individuals with elevated genetic risk or early biochemical changes are often not identified until later stages of disease. At the same time, many individuals with C282Y homozygosity may never require treatment. Treatment, typically therapeutic phlebotomy, is only indicated in individuals with clinically significant iron overload.

These challenges highlight the need for scalable pre-screening strategies that can identify individuals at increased genetic risk, while limiting unnecessary follow-up and overtreatment.

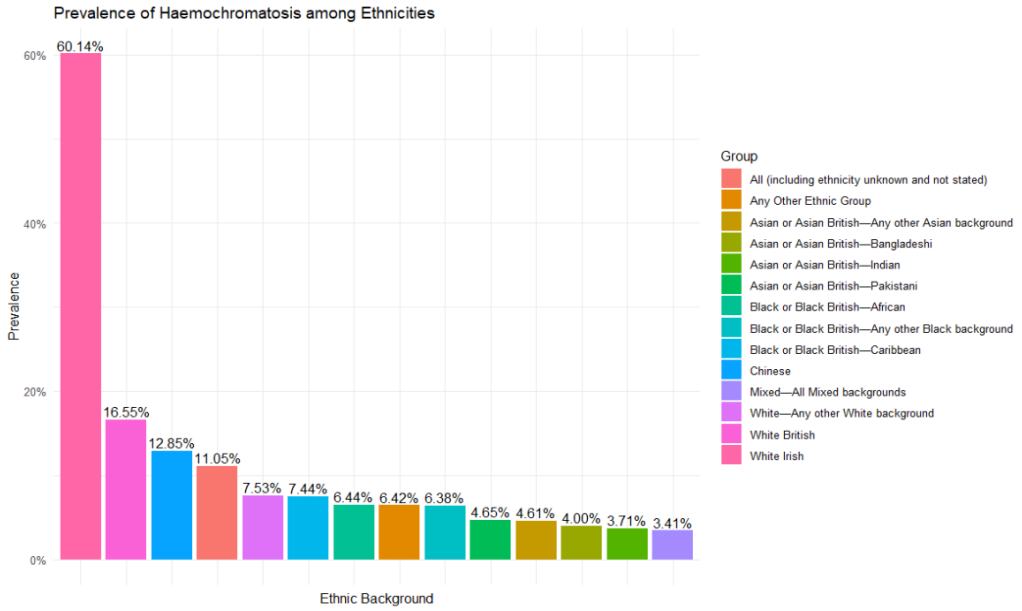


Figure 1.1: Prevalence of haemochromatosis among ethnicities, based on [17, Table 2].

In this context, machine learning (ML) offers a promising approach.

Rather than diagnosing clinical haemochromatosis, an ML model can be framed as a risk stratification tool for C282Y homozygosity. Using demographic variables and measured blood markers, such models may identify individuals with an increased likelihood of carrying the C282Y homozygous genotype. These individuals could then be referred for confirmatory HFE genotyping and appropriate biochemical evaluation, e.g. TS and SF.

Previous studies have explored prediction models for haemochromatosis, often including specialised biomarkers such as TS and SF or genetic data [23, 3]. While these models achieve high discriminatory performance, they do not reflect a pre-screening setting, as the predictors themselves are typically only measured when disease is already suspected.

In contrast, ML models based on measured blood markers have demonstrated promising results in other monogenic blood disorders, where subtle physiological patterns in laboratory tests can indicate underlying genetic variants [7, 8]. This suggests that similar approaches may be applicable to HFE gene mutations.

From a health-economic perspective, early identification of high-risk individuals could reduce long-term complications and healthcare costs, while avoiding the inefficiencies of population-wide genetic testing. Importantly, a positive prediction in a pre-screening model should lead to relatively low-cost confirmatory testing rather than immediate treatment, whereas the implications of missed cases depend on the probability of progression to clinically significant disease.

This thesis therefore aims to develop and evaluate a machine learning model for risk stratification of C282Y homozygosity using demographic data and blood parameters. Furthermore, it investigates the potential of such a model to function as a cost-effective pre-screening tool within a clinical context.

2 | Gradient Boosting Models

In this chapter, the machine learning model used in the analysis is introduced. While a wide variety of algorithms exist, our focus is on methods, which combine multiple models to improve predictive performance. In particular, boosting methods have proven effective in handling complex datasets by sequentially correcting errors of prior models.

Firstly, an overview of *gradient boosting*, which provides the theoretical foundation for the *eXtreme Gradient Boost* (XGBoost) algorithm, is stated. Gradient boosting is based on the core idea of building models iteratively to reduce prediction errors, while XGBoost introduces some enhancements, such as regularization, efficient computation, and improved handling of sparse data.

2.1 Gradient Boosting for Classification

Consider a dataset of input-output pairs $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where each $x_i \in \mathbb{R}^d$ is a vector of explanatory variables and $y_i \in \{e_1, \ldots, e_K\}$ is a one-hot vector of the associated class. Let $L(y, F)$ be a differentiable loss function that quantifies the deviation between the observed class y and the model prediction $F(x)$.

The goal is to construct a function F that gives good predictive performance. Rather than fitting a single complex model at once, gradient boosting represents F as an additive expansion

$$F_M(x) = \sum_{m=0}^{M} f_m(x),$$

where $f_0(x)$ is an initial guess and the remaining terms, f_m, are *base learners*, typically regression trees. These components are introduced sequentially, with each new term chosen to improve upon the model obtained so far.

The central idea is to view the problem as one of iterative function approximation. At iteration m, the current model F_{m-1} produces predictions $F_{m-1}(x_i)$, and the loss depends on these values. The direction in which the loss decreases most rapidly is given by the negative gradient

$$-\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}.$$

These gradient values act as *pseudo-residuals*, i.e. the residuals in the m'th iteration, indicating how the model should change locally to better match the data.

Since the model is built additively, the next component f_m is chosen to approximate this negative gradient. A regression tree offers a flexible piecewise-constant approximation, enabling the algorithm to follow the steepest descent direction in function space. With each iteration, the model adjusts incrementally toward a function that better fits the observed data.

In this way, gradient boosting performs a stage-wise form of functional gradient descent, where each added term f_m provides a small step in the direction that reduces the loss the most, as seen in Algorithm 1, which is based on [14, Alg. 1].

Algorithm 1 Gradient Boosting.

Input: Data $\mathcal{D}$, differentiable loss $L(y, F)$, number of iterations M, learning rate $\nu \in (0, 1]$.

Step 1: Initialization

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma).$$

Step 2: Boosting iterations

for $m = 1, \ldots, M$ **do**

(A) Compute pseudo-residuals

$$r_{im} = - \left. \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right|_{F=F_{m-1}}, \qquad i = 1, \ldots, n.$$

(B) Fit regression tree

Fit a regression tree to $\{r_{im}\}_{i=1}^{n}$ and obtain terminal regions

$$R_{1m}, \ldots, R_{J_m m}.$$

(C) Region-wise optimization

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma), \qquad j = 1, \ldots, J_m.$$

(D) Update

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} \mathbb{1}[x \in R_{jm}] .$$

end for

Output: $F_M(x)$.

The learning rate ν is a hyperparameter that controls the contribution of each newly added tree to the overall model. As seen in the update step, the term ν scales the update from the fitted tree before it is added to the current model $F_{m-1}(x)$. Consequently, the learning rate determines how strongly the model adjusts to the errors identified at each boosting iteration. A smaller learning rate leads to more gradual updates and typically requires a larger number of iterations M to achieve good predictive performance, but it can improve generalization by reducing the risk of overfitting. Conversely, a larger learning rate allows the model to learn faster but may increase the risk of fitting noise in the data.

At each iteration, the pseudo-residuals represent the negative functional gradient of the empirical risk,

$$r_{im} = - \left. \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right|_{F=F_{m-1}},$$

and the regression tree in Step (B) provides a piecewise-constant approximation to this functional gradient.

To compute the region-wise updates in Step (C), let $F_{i,m-1} = F_{m-1}(x_i)$ for observations

in a fixed terminal region R_{jm} and define

$$\Phi(\gamma) = \sum_{x_i \in R_{jm}} L(y_i, F_{i,m-1} + \gamma).$$

Let the gradient and the Hessian at the current iteration be given as

$$g_i = \frac{\partial L(y_i, F)}{\partial F}\bigg|_{F=F_{i,m-1}}, \qquad h_i = \frac{\partial^2 L(y_i, F)}{\partial F^2}\bigg|_{F=F_{i,m-1}}.$$

A second-order Taylor expansion around $\gamma = 0$ yields the quadratic approximation

$$\Phi(\gamma) \approx \text{const} + \Big(\sum_{x_i \in R_{jm}} g_i\Big)\gamma + \frac{1}{2}\Big(\sum_{x_i \in R_{jm}} h_i\Big)\gamma^2,$$

whose minimizer, assuming $h_i > 0$ for at least one entry in R_{jm} and $h_i \geq 0$ for the remaining, is

$$\gamma_{jm} = -\frac{\sum_{x_i \in R_{jm}} g_i}{\sum_{x_i \in R_{jm}} h_i}. \tag{2.1}$$

The general flow of the gradient boost algorithm can be seen in Figure 2.1. The flowchart outlines the iterative procedure of gradient boosting. At each stage, pseudo-residuals are computed from the current model, a regression tree is fitted to these residuals, and the resulting update is added to the model. This process is repeated for M iterations, yielding the final boosted predictor F_M.

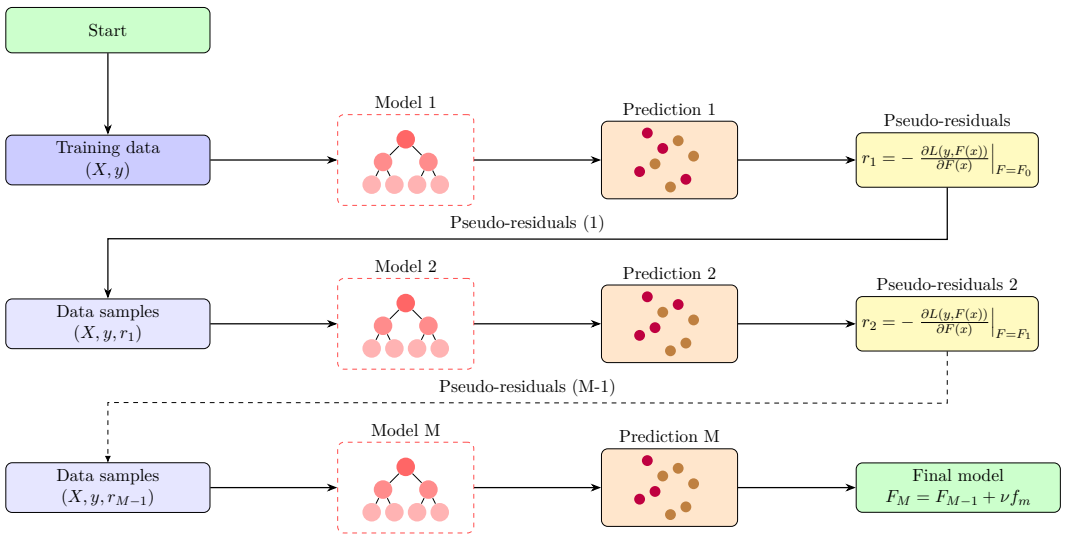


Figure 2.1: Flowchart for gradient boosting algorithm.

In the context of the present study, this framework will be applied to the binary classification problem of identifying individuals carrying the C282Y variant. Consequently, the binary logistic loss is introduced, which forms the basis for computing the pseudo-residuals used throughout the boosting process.

2.1.1 Binary Logistic Regression

For binary classification with $y_i \in \{0, 1\}$, define the log-odds

$$F(x) = \log\left(\frac{p(x)}{1-p(x)}\right), \qquad p(x) = \frac{1}{1+\exp(-F(x))}.$$

Using the negative Bernoulli log-likelihood, the loss function is

$$L(y, F(x)) = -\sum_{i=1}^{n} y_i \log\left(p(x_i)\right) + (1-y_i)\log\left(1-p(x_i)\right). \tag{2.2}$$

Now, in order to transform Eq. (2.2) to a function of the log-odds, $F(x)$, instead of the predicted probability, p, the summation is dropped and the expression is simplified as

$$\begin{aligned} L(y_i, F(x_i)) &= -\Big(y_i \log\left(p(x_i)\right) + (1-y_i)\log\left(1-p(x_i)\right)\Big) \\ &= -y_i \log\left(p(x_i)\right) - (1-y_i)\log\left(1-p(x_i)\right) \\ &= -y_i\big(\log\left(p(x_i)\right) - \log\left(1-p(x_i)\right)\big) - \log\left(1-p(x_i)\right) \\ &= -y_i F(x_i) - \log\left(1-p(x_i)\right), \end{aligned} \tag{2.3}$$

where the log-odds were used in the last line. The remaining term in Eq. (2.3) can be expressed w.r.t. the log-odds as follows

$$\begin{aligned} \log\left(1-p(x_i)\right) &= \log\left(1 - \frac{\exp(F(x_i))}{1+\exp(F(x_i))}\right) \\ &= \log\left(\frac{1+\exp(F(x_i))}{1+\exp(F(x_i))} - \frac{\exp(F(x_i))}{1+\exp(F(x_i))}\right) \\ &= \log\left(\frac{1}{1+\exp(F(x_i))}\right) \\ &= \log(1) - \log\left(1+\exp(F(x_i))\right). \end{aligned}$$

Substituting this into Eq. (2.3) yields the logistic loss expressed by log-odds

$$L(y_i, F(x_i)) = -y_i F(x_i) + \log\left(1+\exp(F(x_i))\right).$$

The first and second order derivatives are

$$\begin{aligned} g_i &= \frac{\partial L}{\partial F(x_i)} = -y_i + \frac{\exp(F(x_i))}{1+\exp(F(x_i))} = -y_i + p(x_i), \\ h_i &= \frac{\partial^2 L}{\partial F(x_i)^2} = \frac{\exp(F(x_i))}{\left(1+\exp(F(x_i))\right)^2} = p(x_i)(1-p(x_i)). \end{aligned}$$

At iteration m, evaluate these at $F_{i,m-1} = F_{m-1}(x_i)$ and write

$$p_{im} = \frac{1}{1+\exp(-F_{m-1}(x_i))}, \qquad r_{im} = y_i - p_{im}, \quad \text{and} \quad w_{im} = p_{im}(1-p_{im}).$$

Substituting $g_i = p_{im} - y_i = -r_{im}$ and $h_i = w_{im}$ into Eq. (2.1) yields

$$\gamma_{jm} = \frac{\sum_{x_i \in R_{jm}} r_{im}}{\sum_{x_i \in R_{jm}} w_{im}}.$$

This can then be used to update the log-odds using Step (D) in Algorithm 1, and subsequently the class probabilities in the m'th iteration.

Once the boosting procedure returns the final log-odds estimate $F_M(x)$, the predicted class probabilities follow from the logistic inversion

$$\widehat{p}_M(x) = \frac{1}{1 + \exp(-F_M(x))}.$$

The preceding derivations describe gradient boosting under the binary logistic loss, which is suitable when distinguishing between two outcome categories. In the present context, this framework allows us to model the probability of being a C282Y carrier versus non-carrier based on clinical or genetic features.

2.2 Extreme Gradient Boosting

Let the data have the form $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$. The model prediction for an observation x_i is obtained by aggregating the outputs of M base learners. As in standard gradient boosting, these base learners are chosen to be regression trees, whose predictions are combined additively to form the final model. Formally, the boosted predictor can be written as

$$F_M(x_i) = \sum_{m=0}^{M} f_m(x_i) \quad f_k \in \mathcal{F},$$
$$\mathcal{F} = \{f(x) = \gamma_{q(x)}\}(q : \mathbb{R}^d \to J, \gamma \in \mathbb{R}^J),$$

where $\mathcal{F}$ is the space of regression trees. Each tree, f_m, has its own structure q and leaf weights γ. The number of leaves in the tree is given as J. To train the M trees for the XGBoost model, the aim is to minimize the following regularized loss function

$$L(y, F(x)) = \sum_{i=1}^{n} L_{GB}(y_i, F(x_i)) + \sum_{m=0}^{M} \Omega(f_m), \tag{2.4}$$
$$\Omega(f_m) = \zeta J_m + \frac{1}{2}\lambda\|\gamma\|^2,$$

where L_{GB} is the differentiable loss function from gradient boosting, with gradient g and Hessian h. The penalization term Ω consist of hyper parameters λ and ζ. Where λ penalizes large leaf weights, thereby controlling model sensitivity and preventing overfitting and ζ penalizes the complexity of the model.

Since the model in Eq. (2.4), has functions as it parameters, regular optimization cannot be used. Instead the model is trained iteratively, as in gradient boosting. Let F_{m-1} describe the prediction for the first $m-1$ base learners. Hence at each new iteration the aim is to minimize the following

$$L(y, F_m(x)) = \sum_{i=1}^{n} L_{GB}(y_i, F_{m-1}(x_i) + f_m(x_i)) + \Omega(f_m).$$

In order to optimize the function in the current iteration, second order Taylor approximation around $f_m(x) = 0$ is used, which yields

$$L(y, F_m(x)) \approx \sum_{i=1}^{n} \Big(L_{GB}(y_i, F_{m-1}(x_i)) + g_i f_m(x_i) + h_i f_m^2(x_i)\Big) + \Omega(f_m).$$

The constant term can be ignored, since it does not depend on $f_m(x_i)$, which yields

$$L(y, F_m(x)) \approx \sum_{i=1}^{n} \Big(g_i f_m(x_i) + h_i f_m^2(x_i) \Big) + \Omega(f_m). \tag{2.5}$$

For a tree f_m with terminal regions $1, \ldots, J_m$, define $R_{jm} = \{i \mid q(x_i) = j\}$ as the set of instances that fall into region j, where q maps x_i to the regional index. Then Eq. (2.5) is rewritten such that

$$L(y, F_m(x)) \approx \sum_{i=1}^{n} \Big(g_i f_m(x_i) + h_i f_m^2(x_i) \Big) + \zeta J_m + \frac{1}{2}\lambda \sum_{j=1}^{J_m} \gamma_j^2.$$

When considering the instances of R_{jm} and that $f_m(x_i)$ is a representation of the whole tree construction, it becomes the weight γ_j when considering the j'th region. This yields

$$L(y, F_m(x)) \approx \sum_{j=1}^{J_m} \left(\sum_{i \in R_{jm}} g(x_i)\gamma_j + \frac{1}{2}\Big(\sum_{i \in R_{jm}} h(x_i) + \lambda \Big)\gamma_j^2 \right) + \zeta J_m.$$

The optimal weight γ_j that contributes to minimize the loss function is

$$\gamma_j = -\frac{\sum_{i \in R_{jm}} g_i}{\sum_{i \in R_{jm}} h_i + \lambda}.$$

Hence the corresponding optimal value can be calculated by

$$L(y, F_m(x), q) = -\frac{1}{2} \sum_{j=1}^{J_m} \frac{(\sum_{i \in R_{jm}} g_i)^2}{\sum_{i \in R_{jm}} h_i + \lambda} + \zeta J_m, \tag{2.6}$$

where $\frac{(\sum_{i \in R_{jm}} g_i)^2}{\sum_{i \in R_{jm}} h_i + \lambda}$ is a similarity score. The optimal value in Eq. (2.6) can be used to measure the quality of the tree structure q. Where small values of the Hessian h would indicate good prediction accuracy for the tree, hence the ratio will be bigger and the loss will hence be minimized.

It would be infeasible to calculate the similarity score for all tree structures. Hence a greedy algorithm that determines the best possible split for each node is used instead. It starts with a single leaf and keeps on adding nodes to the tree.

Assume all observations are in one leaf, I, and, for some threshold, split it into two new leafs I_L and I_R. Letting $I = I_L \cup I_R$, then the reduction in the loss function is given as

$$L_{split} = \frac{1}{2} \left(\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right) - \zeta,$$

which is called the gain. It gives an indication of how the candidate split contributes to minimising the loss function. Below in Algorithm 2, it can be seen how the greedy algorithm works. The exact greedy algorithm evaluates every possible split point for every feature. At each node, it sorts the feature values and checks every unique value to find the split that maximizes gain. However, this requires all data is available in memory so it can be sorted and scanned repeatedly. When the dataset is very large, this becomes computationally expensive or even infeasible.

To overcome this limitation, XGBoost introduces an approximate, histogram-based, algorithm. Instead of evaluating every unique split point, it first groups feature values into

Algorithm 2 Exact Greedy algorithm for Split Finding, [6, Alg. 1].

Input: Instance set of current node I, feature dimension d.
Initialization:

$$Gain = 0, \qquad G = \sum_{i \in I} g_i, \qquad H = \sum_{i \in I} h_i, \qquad score = 0.$$

for $k = 1, \ldots .d$ **do**

$$G_L = 0, \qquad H_L = 0.$$

for $j \in$ sorted(I, by x_{jk}) **do**

$$\begin{aligned} G_L &= G_L + g_j, \qquad H_L = H_L + h_j \\ G_R &= G - G_L, \qquad H_R = H - H_L \\ score &= \max\left(score, \frac{G_L^2}{H_R^2 + \lambda} + \frac{G_R^2}{H_R^2 + \lambda} - \frac{G^2}{H^2 + \lambda}\right). \end{aligned}$$

end for
end for
Output: Split with highest score.

a fixed number of bins, often based on percentiles. The algorithm then computes gradient and Hessian per bin and evaluates splits only between bins. In other words, instead of checking every possible split, it checks a subset of representative splits. This reduces memory usage and computation time, while typically achieving near same accuracy as the exact method.

Algorithm 3 Approximate Alghorithm for Split Finding, [6, Alg. 2].

for $k = 1, \ldots, d$ **do**
Propose $S_k = \{s_{k,1}, s_{k,2}, \cdots, s_{k,l}\}$ by percentiles on feature k
Proposal can be done per tree (global) or per split (local)
end for
for $k = 1, \ldots, d$ **do**

$$\begin{aligned} G_{kv} &= \sum_{j \in \{j | s_{k,v} \geq x_{jk} > s_{k,v-1}\}} g_j, \\ H_{kv} &= \sum_{j \in \{j | s_{k,v} \geq x_{jk} > s_{k,v-1}\}} h_j. \end{aligned}$$

end for
Follow same steps as in previous section to find highest score only among proposed splits.

One important thing in the Algorithm 3, is to suggest candidate splits, which is typically done using percentile of the specific feature x_k. This ensures that the amount of data in each quantile are evenly distributed across the data. The weighted quantile sketch extends the idea of quantile-based candidate splits by weighting observations using their Hessians. Consider the data for the k'th feature $\mathcal{D}_k = \{(x_{1k}, h_{1k}), (x_{2k}, h_{2k}), \ldots, (x_{nk}, h_{nk})\}$, where h is the corresponding Hessian for the data point. A rank function $\rho_k : \mathbb{R} \to [0, \infty]$ can be

defined and represents the proportion of Hessian, for a feature value that are smaller than z

$$\rho_k(z) = \frac{\sum_{(x,h)\in\mathcal{D}_k, x<z} h}{\sum_{(x,h)\in\mathcal{D}_k} h}.$$

The main idea is to find candidate splits $S_k = \{s_{k,1}, s_{k,2}, \cdots, s_{k,l}\}$, such that

$$|\rho_k(s_{k,j}) - \rho_k(s_{k,j+1})| < \varepsilon, \quad s_{k,1} = \min_i x_{ki}, \quad s_{k,l} = \max_i x_{ki},$$

where ε is an approximation factor and $1/\varepsilon$ roughly describes the number of candidate points. In this case each data points is weighted by their Hessian h. This can be seen by rewriting Eq. (2.5) and obtain the following

$$L(y, F_m(x)) \approx \sum_{i=1}^{n} \frac{1}{2} h_i \left(f_m(x_i) - \frac{g_i}{h_i} \right)^2 + \Omega(f_m) + \text{const.}$$

This shows that the second-order approximation in XGBoost can be interpreted as a weighted squared loss, with labels g_i/h_i and weights h_i.

In case of sparse data, which can arise from missing values or frequent zeros due to feature engineering, XGBoost uses the Sparsity-Aware Split Finding algorithm, see Algorithm 4, to efficiently handle sparsity patterns. For missing values, a default direction is added to each tree node. The algorithm computes the gain assuming all missing values go left, then assuming they all go right. The optimal default direction is chosen based on which option yields the highest gain.

The algorithms in Algorithm 2 and Algorithm 4 both aim to determine the optimal split point by maximizing the gain function derived from the second-order approximation of the objective in XGBoost. However, they differ in their treatment of missing feature values. The exact greedy algorithm assumes fully observed feature values and does not explicitly incorporate missingness, whereas the sparsity-aware split finding algorithm is specifically designed to handle missing or sparse data by learning the optimal direction for missing values during split construction.

Thus, the sparsity-aware algorithm extends the exact greedy method by restricting computations to non-missing values and by learning the optimal routing of missing observations, improving efficiency on sparse datasets.

Algorithm 4 Sparsity-Aware Split Finding, [6, Alg. 3].

Input: Instance set of current node I, feature dimension d.
Input: $I_k = \{i \in I \mid x_{ik} \neq \text{missing}\}$ for each feature k.
Initialization:

$$Gain = 0, \qquad G = \sum_{i \in I} g_i, \qquad H = \sum_{i \in I} h_i, \qquad score = 0.$$

for $k = 1, \ldots, d$ **do**
(A) Missing values assigned to the right child

$$G_L = 0, \qquad H_L = 0.$$

for $j \in$ sorted(I_k, in ascending order by x_{jk}) **do**

$$\begin{aligned} G_L &= G_L + g_j, \qquad H_L = H_L + h_j, \\ G_R &= G - G_L, \qquad H_R = H - H_L, \\ score &= \max\left(score, \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G^2}{H + \lambda}\right). \end{aligned}$$

end for
(B) Missing values assigned to the left child

$$G_R = 0, \qquad H_R = 0.$$

for $j \in$ sorted(I_k, in descending order by x_{jk}) **do**

$$\begin{aligned} G_R &= G_R + g_j, \qquad H_R = H_R + h_j, \\ G_L &= G - G_R, \qquad H_L = H - H_R, \\ score &= \max\left(score, \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G^2}{H + \lambda}\right). \end{aligned}$$

end for
end for
Output: The split and default direction for missing values-left or right-corresponding to the maximum gain obtained by evaluating all candidate splits in Step (A) and Step (B).

3 | Construction of Cost Function

The aim of this chapter is to motivate the use of cost-based evaluation measures rather than traditional error-based metrics when developing classification models. In standard classification settings, model performance is typically evaluated using the error rate, implicitly assuming equal misclassification costs, i.e. 0/1 loss, as illustrated in Table 3.1, which is a common assumption across many models [11].

Table 3.1: Equal cost matrix per person.

		Actual	
		Negative	Positive
Predicted	Negative	$C_{TN} = 0$	$C_{FN} = 1$
	Positive	$C_{FP} = 1$	$C_{TP} = 0$

The most common metrics for evaluating classification is using either the misclassification rate (MCR) or the accuracy (acc),

$$MCR = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[y_i \neq \widehat{y}_i], \qquad acc = 1 - MCR,$$

where $\widehat{y}_i$ is the predicted class for observation i and y_i is the true label. These metrics assumes equal cost of misclassification. In some applications, metrics like F_β or the Matthews correlation coefficient (MCC) are used. F_β allows prioritizing recall or precision when certain errors are more important, while MCC provides a balanced measure even for imbalanced datasets. Neither metric directly incorporates a true cost matrix

$$\text{Recall} = \frac{TP}{TP + FN}, \qquad \text{Precision} = \frac{TP}{TP + FP},$$

$$F_\beta = (1 + \beta^2)\frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}},$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}.$$

For $\beta = 1$, the regular F_1 score is obtained, and it weights precision and recall equally. For $\beta > 1$ it weights recall higher than precision, and vice versa for $\beta < 1$ [13].

F_β does not directly incorporate a true misclassification cost matrix, it only adjusts the relative importance of recall versus precision according to β. Therefore, it does not reflect real-world costs, only a heuristic weighting.

MCC is a measure of classification performance that takes all four elements of the confusion matrix into account. It can be interpreted as a correlation coefficient between the observed and predicted classifications, with values ranging from -1, which is total disagreement, to 1, which is perfect classification and 0 indicating performance no better than random guessing. Unlike accuracy and F_β, MCC provides a balanced evaluation even in the presence of class imbalance, as it does not favor the majority class. However, similar to F_β, MCC does not explicitly incorporate a misclassification cost matrix and therefore does not directly reflect application-specific costs.

Since the objective is to minimize the total economic cost associated with implementing the model, it is necessary to work with an explicit cost matrix. In many applications, cost matrices are defined deterministically, where the cost of each outcome is fixed. However,

in a clinical setting such as HH, this approach may not be appropriate. In particular, the model predicts C282Y homozygosity, which represents a genetic risk rather than the presence of clinically manifest disease. Not all individuals with this genotype develop clinically significant iron overload or require treatment. Consequently, assigning fixed treatment costs to all true positives, or fixed disease costs to all false negatives, would overestimate the true economic impact. To address this, costs are defined in terms of expected values, incorporating the clinical penetrance of the disease. This ensures that the cost structure reflects the real clinical pathway, where a positive prediction leads to confirmatory testing and only a subset of individuals subsequently require treatment. Let p_{disease} denote clinical penetrance. Empirical studies indicate that penetrance varies substantially by sex, 28.4% for men and 1.2% for women, however, a population-average value of $p_{\text{disease}} = 14.8\%$ is used. This simplifies the cost model, while acknowledging that the true risk of disease progression is heterogeneous across individuals [1].

The cost terms used are based on a technical report, which investigated the economic impact of screening for HH in Northern Ireland [21]. The cost terms are

- C_{TP} represents the expected cost of correctly identifying a C282Y homozygous individual. This includes:
  - Cost of confirmatory HFE genotyping, estimated at 56£,
  - Expected treatment cost, weighted by the probability of developing clinically significant iron overload, estimated at 1188£.

  Thus,

$$C_{TP} = 56£ + 0.148 \cdot 1188£ = 232£.$$

- Since correctly identifying a non-carrier results in no further clinical action, the cost is

$$C_{TN} = 0£.$$

- C_{FP} represents the cost of unnecessary follow-up, which in this setting corresponds to confirmatory genotyping

$$C_{FP} = 56£.$$

- C_{FN} represents the expected cost of missing a C282Y homozygous individual. This depends on the probability of developing clinically significant disease and the associated downstream costs

$$C_{FN} = 0.148 \cdot 1675£ = 248£.$$

- $C_A = n \cdot 56£$ denotes the cost of genotype testing the entire cohort.
- C_S denotes the infrastructure cost associated with running the model, estimated as 8,932£ [30].

The resulting penetrance-adjusted cost matrix is given in Table 3.2.

Table 3.2: Cost matrix per person.

Predicted \ Actual	Negative	Positive
Negative	$C_{TN} = 0$ £	$C_{FN} = 248$£
Positive	$C_{FP} = 56$ £	$C_{TP} = 232$ £

Compared to a deterministic cost formulation, this approach results in lower expected costs for both true positives and false negatives. This reflects that only a fraction of genetically at-risk individuals develop clinically significant disease, and therefore that costs should be interpreted as averages rather than certain outcomes.

Note that the cost can also be considered as random, since the final cost usually is unknown until the model is used in practice [18].

3.1 Cost-Sensitive Decision Making

A way of accounting for the severity of the error is by using cost-sensitive learning. One method is expected cost, which takes into account the probability of being in the true class j, and the cost of predicting it as class k. Expected cost minimization is a generalization of the Bayes optimal classifier,

$$\widehat{y} = \arg\max_{y} \mathbb{P}(y \mid x),$$

i.e., the class with the highest probability conditioned on the features x, is chosen [16]. When cost differs, the expected cost is utilized instead

$$\arg\min_{k \in \{1,\dots,K\}} \sum_{j=1}^{K} \mathbb{P}(j \mid x) C_{k,j}, \tag{3.1}$$

where $K \in \mathbb{N}$ is the number of classes, $\mathbb{P}(j \mid x)$ is the probability of being in class j given features x and $C_{k,j}(x)$ is the cost of classifying an observation from class j as class k. In the binary case, Eq. (3.1) reduces to

$$\arg\min_{k \in \{0,1\}} \sum_{j=0}^{1} \mathbb{P}(j \mid x) C_{k,j}, \tag{3.2}$$

where $C_{00} = C_{TN}, C_{01} = C_{FN}, C_{10} = C_{FP}$ and $C_{11} = C_{TP}$. Hence, when predicting k as class 0, Eq. (3.2) becomes

$$(1 - \mathbb{P}(j = 1 \mid x))C_{TN} + \mathbb{P}(j = 1 \mid x)C_{FN},$$

and when predicting k as class 1, it becomes

$$(1 - \mathbb{P}(j = 1 \mid x))C_{FP} + \mathbb{P}(j = 1 \mid x)C_{TP}.$$

In order to determine if an observation belongs to class 1, the following inequality is considered

$$\begin{aligned}(1 - \mathbb{P}(j = 1 \mid x))C_{FP} + \mathbb{P}(j = 1 \mid x)C_{TP} \\ < (1 - \mathbb{P}(j = 1 \mid x))C_{TN} + \mathbb{P}(j = 1 \mid x)C_{FN}.\end{aligned} \tag{3.3}$$

The inequality in Eq. (3.3) can be rewritten as a classification rule for class 1;

$$\mathbb{P}(j = 1 \mid x) > \frac{C_{FP} - C_{TN}}{(C_{FP} - C_{TN}) + (C_{FN} - C_{TP})}, \tag{3.4}$$

where the right hand side, under certain conditions, is called the cost-optimal decision threshold t^* [11]. For this threshold to be optimal, the cost matrix must satisfy several requirements, namely all costs must be fixed, independent of the covariates x, and stationary, meaning that it does not change during training, and it must adhere to the reasonable assumptions that $C_{FP} > C_{TN}$ and $C_{FN} > C_{TP}$.

Because optimal cost-sensitive decisions rely on posterior class probability estimates, the accuracy of these probabilities are important. A model may correctly rank observations while still producing poorly *calibrated* probabilities. In such cases, the resulting decisions may not be cost-optimal, since cost-sensitive decision rules depend on the actual values of the predicted posterior probabilities rather than their relative ordering, and miscalibration can therefore lead to incorrect decisions when applying cost-based thresholds. Ranking refers to the ability of a model to order observations according to their predicted probability of belonging to the positive class, regardless of whether the predicted probabilities themselves are numerically accurate [11].

3.2 Heuristics

In practice, the true posterior probabilities and the analytically optimal decision threshold are typically unknown. To address this limitation, empirical cost functions can be defined using the predicted probabilities generated by the model and the available dataset. These functions make it possible to compute and minimize the total expected cost driven by data, thereby yielding practical thresholds for cost-sensitive decision-making. In this subsection, the cost function developed for this project is presented.

A practical approach is to define a cost function using threshold-dependent counts of TP, TN, FP, and FN. These counts are defined as

$$\begin{aligned}
q_{TP}(t) &= \sum_{i=1}^{n} \mathbb{1}[\widehat{y}_i(t) = 1, y_i = 1], \\
q_{TN}(t) &= \sum_{i=1}^{n} \mathbb{1}[\widehat{y}_i(t) = 0, y_i = 0], \\
q_{FP}(t) &= \sum_{i=1}^{n} \mathbb{1}[\widehat{y}_i(t) = 1, y_i = 0], \\
q_{FN}(t) &= \sum_{i=1}^{n} \mathbb{1}[\widehat{y}_i(t) = 0, y_i = 1],
\end{aligned}$$

where

$$\widehat{y}_i = \mathbb{1}[\widehat{p}_i > t].$$

Using these threshold-dependent counts, the following empirical cost function is proposed

$$\mathcal{C}(t) = q_{TP}(t)C_{TP} + q_{TN}(t)C_{TN} + q_{FP}(t)C_{FP} + q_{FN}(t)C_{FN}. \tag{3.5}$$

The expression in Eq. (3.5) represents the total cost incurred by the model over the dataset as a function of the decision threshold t. Minimizing $\mathcal{C}(t)$ yields a data-driven estimate of the optimal threshold for cost-sensitive decision-making.

Since this project is clinical in nature, the threshold must satisfy a minimum sensitivity of 0.95. Moreover, for the predictive system to be economically viable, its total cost must be lower than the cost of screening the cohort, C_A, minus the cost associated with server power, C_S. Incorporating these considerations, the resulting optimization problem is

$$\begin{aligned} \underset{t\in[0,1]}{\text{minimize}} \;& \mathcal{C}(t) \\ \text{subject to} \;& \mathcal{C}(t) - (C_A - C_S) \leq 0, \\ & 1 - \text{sensitivity}(t) \leq 0.05, \\ & n \geq 0, \quad C_{k,j} \geq 0. \end{aligned} \tag{3.6}$$

4 | Linear Programming

Having established the cost structure of misclassification in Eq. (3.5), the next step is to formulate an optimization framework capable of identifying the optimal decision threshold in Eq. (3.6). To support this, the concepts of *Linear Programming* (LP) and, subsequently, *Mixed Integer Linear Programming* (MILP) are introduced [39, 40].

4.1 Linear programming

LP is a mathematical framework for optimizing linear objective functions, subject to a set of linear constraints. An LP problem can be formulated on the form

$$\begin{aligned} &\text{minimize } c^\top x \\ &\text{subject to } Ax \leq b, \\ &\qquad x \geq 0, \end{aligned} \tag{4.1}$$

where $x \in \mathbb{R}^n$ is the decision vector, $c \in \mathbb{R}^n$ is a vector of objective coefficients, and $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ encode the linear inequality constraints. The structure of LP ensures that both the objective function and the constraints spanning the feasible regions are linear, which implies that any optimal solution, if one exists, is attained at an extreme point of the feasible *polyhedron*.

The feasible set defined by Eq. (4.1) is a polyhedron, that is, the intersection of finitely many closed half-spaces of the form

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}.$$

Such sets are convex and have a piecewise-linear geometry consisting of faces, edges, and extreme points. Extreme points are of particular importance in LP theory, as they represent candidates for optimality, i.e. an extreme point cannot be written as a convex combination of two distinct feasible points, and, due to the convexity and linearity of the LP, at least one optimal solution, should it exist, is always found at one of these points.

A widely used algorithm for solving LPs is the *Simplex* method. Instead of searching the entire feasible region, the Simplex algorithm exploits the polyhedral structure. Starting from an initial feasible extreme point, the method progresses along edges of the polyhedron to adjacent extreme points that improve the objective value. Each move corresponds to a change in the basis of the constraint matrix, known as a pivot. The procedure continues until no adjacent extreme point yields further improvement, at which stage optimality is certified [10]. The Simplex method is summarized in Algorithm 5.

The basis chosen in Step 1 gives a basic feasible solution, which corresponds to a vertex of the feasible region. After choosing the entering variable e, a search direction is computed, which corresponds to moving along an edge of the feasible region. Then the Minimum Ratio Test is performed to determine how far it is possible to move along this edge and which variable leaves the basis. When the basis is updated, a new basic feasible solution is obtained, corresponding to a neighbouring vertex of the feasible region.

Algorithm 5 Simplex Method.

Input: LP in standard form $\min\{c^\top x \ : \ Ax = b,\, x \geq 0\}$, with a known initial basic feasible solution.

Step 1: Initialization

Choose an initial basis B with corresponding basic variables x_B

$$x_B = A_B^{-1} b \geq 0.$$

Let N denote the complementary index set.

Step 2: Compute reduced costs

Compute dual variables and reduced costs:

$$y^\top = c_B^\top A_B^{-1}, \qquad r = c - A^\top y.$$

Select the entering variable index

$$e = \arg\min_{j \in N} \{r_j\}.$$

Step 3: Optimality test

if $r_e \geq 0$ **then**

Stop: current basic feasible solution is optimal.

end if

Step 4: Compute search direction

$$d = A_B^{-1} A_{\star e}.$$

Step 5: Unboundedness test

if $d_i \leq 0 \ \ \forall i$ **then**

Terminate: the objective function is unbounded.

end if

Step 6: Determine the leaving variable with Minimum Ratio Test

$$\theta = \min_{i:d_i>0} \frac{(x_B)_i}{d_i}, \qquad o = \arg\min_{i:d_i>0} \frac{(x_B)_i}{d_i}.$$

Step 7: Update the basis

Pivot: replace B_o with N_e in the basis and update

$$B \leftarrow (B \setminus B_o) \cup N_e, \qquad x_B \leftarrow x_B - \theta d, \qquad (x_B)_o \leftarrow \theta.$$

Return to Step 2.

Output: Optimal solution x^* and objective value $c^\top x^*$.

Building on these concepts, it is important to note that classical LP formulations are limited to continuous decision variables. However, some decision problems require discrete choices that cannot be represented adequately within a purely linear and continuous framework. For instance, selecting whether an observation is assigned to one side of a threshold or another introduces a binary structure. To model such decisions explicitly, the LP framework is extended to include integer-valued variables. This leads to the class of *Integer Linear Programs* (ILPs), and, more generally, MILPs, where only a subset of

the decision variables are restricted to take integer values. These formulations keep the objective and constraints linear, but give more flexibility in what can be modelled. In particular, they enable decision problems involving discrete choices or logical relationships to be represented. In the next section, these concepts are introduced in more detail, and it is illustrated how they can be used to formulate and solve the optimization problem of interest.

4.2 Mixed Integer Linear Programming

As mentioned in Section 4.1, a MILP problem is an optimization problem of the form

$$\begin{aligned} &\text{minimize } c^\top x \\ &\text{subject to } Ax \leq b, \\ &\qquad x_i \in \mathbb{Z} \quad \text{for } i \in \mathcal{I}, \\ &\qquad x_j \geq 0 \quad \text{for } j \notin \mathcal{I}, \end{aligned} \tag{4.2}$$

where $\mathcal{I}$ denotes the set of indices corresponding to variables that take integer values. The linear structure remains unchanged, but the introduction of integrality constraints, Eq. (4.2), makes the problem different from classical LP. This means the feasible region is no longer a convex polyhedron but a collection of discrete points or unions of polyhedral sets. Hence, another method is needed, in order to solve MILP problems. The main idea used to solve such problems is to consider the problem without the integrality condition, i.e., to solve the *relaxation* of the problem.

A way to solve MILP problems is to use a tree-search strategy known as *branch-and-bound* (BnB). The procedure begins by ignoring the integrality constraints and solving the LP-relaxation of the original problem with the simplex method, Algorithm 5, referred to as the root node and denoted ILP_0. This relaxation provides a *candidate solution*, x^*, which may violate integrality, and a lower bound on the objective value that any integer-feasible solution can achieve within the corresponding region of the search space. If the initial relaxation happens to return an integer-feasible point, the algorithm terminates immediately, since optimality is guaranteed by the relaxation's lower-bounding property. Otherwise, in case where one or more integer-restricted variables take fractional values, the algorithm proceeds to the branching step. Here, a variable x_h^* that violates integrality is selected, and the current problem is split into two new subproblems by imposing mutually exclusive constraints

$$x_h \leq \lfloor x_h^* \rfloor, \quad \text{and} \quad x_h \geq \lceil x_h^* \rceil.$$

These constraints ensure that any feasible solution in either subproblem moves closer to integrality for that variable. The resulting pair of subproblems forms two child nodes in the BnB tree, which is illustrated on the left-hand side of Figure 4.1. Each child node represents a restricted version of the original problem and is solved again using LP-relaxation.

As the tree expands, BnB relies on *fathoming* to avoid an exhaustive enumeration of all possible integer assignments. Fathoming is driven by a global record of the best integer-feasible solution found so far, called the *incumbent*, with objective value z_{opt}. Each time an LP-relaxation is solved at a node, its objective value provides a lower bound on any integer-feasible solution obtainable from that node's subtree. If this lower bound is no better than z_{opt}, or if the relaxation is infeasible, then the entire subtree can be discarded, since there is no possibility of improvement. Similarly, if the LP-relaxation at a node yields an integer-feasible solution, the incumbent is updated if the new solution is better, and

the node is fathomed, since no further branching is needed. The right-hand illustration in Figure 4.1 shows how fathoming reduces the size of the search tree in practice.

Furthermore, a practical consideration is the order in which nodes are selected for exploration. A commonly used strategy is *depth-first search* (DFS), where the algorithm follows one branch of the tree as far as possible before backtracking. An alternative is *breadth-first search* (BFS), which explores all nodes at a given depth before moving deeper into the tree. The BnB method is summarized in Algorithm 6.

Algorithm 6 Branch-and-Bound for MILP

Input: MILP problem $\min\{c^\top x : Ax = b,\ x_j \geq 0,\ j \notin \mathcal{I},\ x_i \in \mathbb{Z},\ i \in \mathcal{I}\}$ and search method.
Initialization:
  Solve LP-relaxation of the original problem (ILP_0), with z_{LP} as its objective value and x^* its solution.
  Create a node list $\mathcal{N} \leftarrow \{\text{ILP}_0\}$.
  Set the incumbent $z_{\text{opt}} \leftarrow \infty$ and solution $x_{\text{opt}} \leftarrow \emptyset$.
while $\mathcal{N} \neq \emptyset$ **do**
  Select a node P from $\mathcal{N}$ and remove it.
  Solve LP-relaxation of P and obtain solution x^* with value z_{LP}.
  if P is infeasible **then**
    Fathom.
    continue
  end if
  if $z_{\text{LP}} \geq z_{\text{opt}}$ **then**
    Fathom: cannot improve incumbent.
    continue
  end if
  if x^* satisfies all integrality constraints **then**
    $z_{\text{opt}} \leftarrow z_{\text{LP}}$.
    $x_{\text{opt}} \leftarrow x^*$.
    Fathom.
    continue
  end if
  Identify a fractional integer variable x_h^*, $h \in \mathcal{I}$.
  Create two subproblems:
    Left branch: add constraint $x_h \leq \lfloor x_h^* \rfloor$.
    Right branch: add constraint $x_h \geq \lceil x_h^* \rceil$.
  Add both child nodes to the list: $\mathcal{N} \leftarrow \mathcal{N} \cup \{\text{Left branch}, \text{Right branch}\}$.
end while
Output: x_{opt} and z_{opt}.

To connect the general BnB principles to a concrete setting, Example 4.1 demonstrates how the method operates on an explicit ILP.

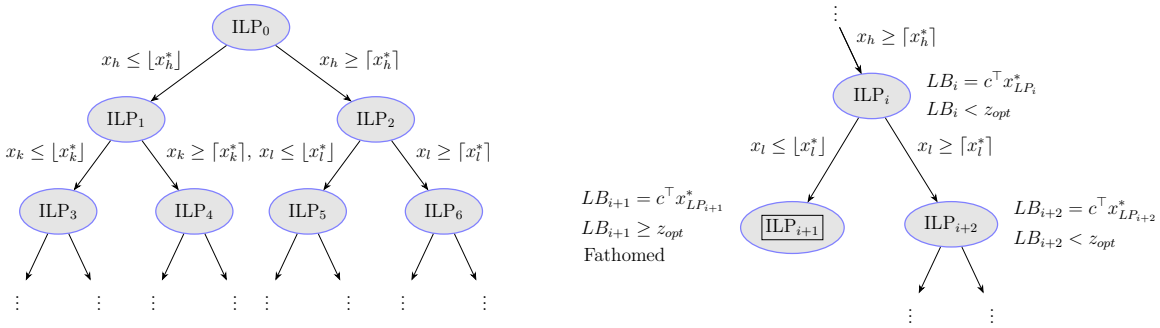


Figure 4.1: Illustration of the branch-and-bound process. Left: Basic branching structure. Right: Example with fathomed nodes, where the box around ILP_{i+1} indicates that it is a leaf in the tree.

Example 4.1
Consider the following ILP on the form

$$\begin{aligned} \text{minimize } & -17x_1 - 12x_2 \\ \text{subject to } & 10x_1 + 7x_2 \leq 40, \\ & x_1 + x_2 \leq 5, \\ & x_1, x_2 \geq 0, \\ & x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

The left-hand plot in Figure 4.2 shows the integer lattice points together with the feasible region of the LP-relaxation. Removing the integrality constraints yields the LP-relaxed problem, which is solved using Algorithm 5. The algorithm on the first LP-relaxation is covered in Appendix A. This yields the optimal solution

$$x^* = \left(\tfrac{5}{3}, \tfrac{10}{3}\right), \qquad z_{\text{LP}} = -17 \cdot \tfrac{5}{3} - 12 \cdot \tfrac{10}{3} = \tfrac{205}{3} \approx -68.33.$$

This value provides an lower bound on the optimal ILP objective. Since the solution is fractional in both variables, it cannot be accepted as feasible for the ILP, and the BnB procedure must branch. The first branching step is performed on the fractional variable $x_1 = 5/3$. This produces two subproblems

$$x_1 \leq \left\lfloor \tfrac{5}{3} \right\rfloor = 1, \qquad x_1 \geq \left\lceil \tfrac{5}{3} \right\rceil = 2.$$

The right-hand side of Figure 4.2 illustrates the corresponding feasible subregions, ILP_1 and ILP_2.

Then the tree is explored using depth-first search (DFS), meaning that branch $x_1 \leq 1$ is examined first. Solving this LP-relaxed subproblem, again using Algorithm 5, yields the point

$$(x_1, x_2) = (1, 4),$$

which is both feasible and integer. Its objective value is

$$z = -17 \cdot 1 - 12 \cdot 4 = -65.$$

Because this is the first integer-feasible point encountered, it becomes the incumbent, i.e. $x_{\text{opt}} = (1, 4)$ and $z_{\text{opt}} = -65$, and the node is immediately fathomed, since no further branching from this point can improve its own value.

The algorithm then backtracks to explore the other branch $x_1 \geq 2$. Along this branch, additional nodes are created whenever fractional solutions appear, while nodes are fathomed either because their LP-relaxations are infeasible, e.g. when $x_1 \geq 1$ and $x_2 \geq 3$, or because their LP bounds are worse than the current incumbent.

Eventually, the search identifies the integer-feasible point,

$$(x_1, x_2) = (4, 0), \qquad z = -17 \cdot 4 - 12 \cdot 0 = -68,$$

which improves upon the incumbent. Thus the incumbent is updated to $x_{\text{opt}} = (4, 0)$ with objective value $z_{\text{opt}} = -68$. All remaining nodes are ultimately fathomed, since none of them can yield a better solution than this updated incumbent.

The full branching tree is shown in Figure 4.3. Since all branches have been fathomed, the optimal solution to the ILP is

$$(x_1, x_2) = (4, 0), \qquad z_{\text{opt}} = -68.$$

△

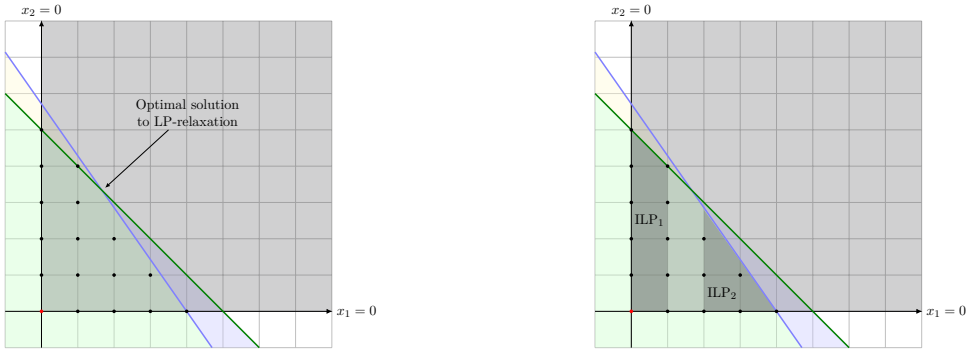


Figure 4.2: Left: Illustration of the integer programming problem. The dots represent feasible integer points, while the shaded region below the constraint lines and restricted to the first quadrant represents the feasible region of the LP-relaxation. Right: The feasible subregions, ILP_1 and ILP_2, formed after the first branching step, shown in dark grey.

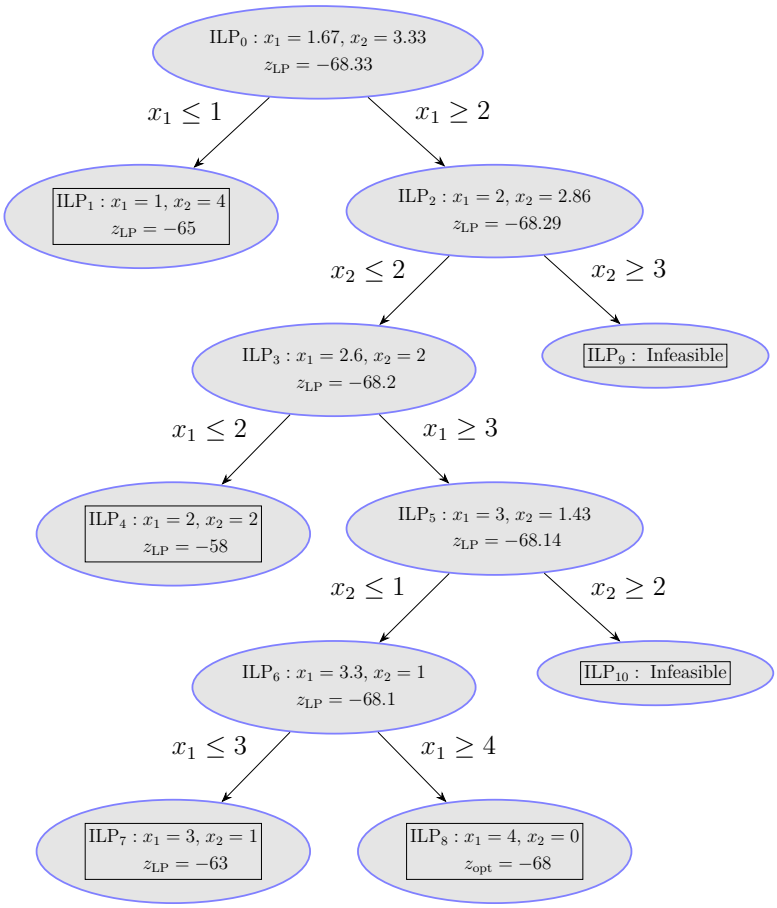


Figure 4.3: Complete branching tree for the ILP.

5 | Missing data

Missing data are common in large general population studies, including the UK Biobank (UKB). Missingness occurs when values of one or more features are unavailable for some participants. In UKB, missing data may arise from several sources, including non-response to questionnaire items, non-attendance at follow-up assessments, pre-analytical or laboratory-related failures affecting biological samples, or exclusion of measurements during predefined quality-control procedures [2, 4].

Let Z denote a $n \times 1$ data vector containing the features of interest, and let R be a corresponding binary missingness indicator, where $R_i = \mathbb{1}[Z_i \text{ is observed}]$ for $i = 1, \ldots, n$. For a given realization r of R, the observed and unobserved components of Z are denoted by $Z^r = (Z_i)_{r_i=1}$ and $Z^{\neg r} = (Z_i)_{r_i=0}$, respectively. Furthermore, let $z = (z^r, z^{\neg r})$ denote a realization of Z. The interpretation and handling of missing data depend on assumptions about the missingness mechanism. Following Rubin's framework in [28], missingness can be classified by the conditional distribution

$$\mathbb{P}(R = r \mid Z = z).$$

Data are said to be *missing completely at random* (MCAR) if

$$\mathbb{P}(R = r \mid Z = z) = \mathbb{P}(R = r),$$

i.e., if the probability of missingness is independent of the data vector Z. Under MCAR, the observed data constitute a random subsample of the full cohort. MCAR could arise when, a data file becomes corrupted due to a system failure, affecting a random subset of records. Data are said to be *missing at random* (MAR) if

$$\mathbb{P}(R = r \mid Z = z) = \mathbb{P}(R = r \mid Z^r = z_r),$$

meaning that, conditioned on the observed data, the probability of missingness does not depend on the unobserved values $Z^{\neg r}$. An example of MAR is that young people are less likely to report their income in surveys, but the missingness depends on age, which is observed, rather than the actual income.

If its neither MCAR nor MAR, then data are *missing not at random* (MNAR), in which case $P(R = r \mid Z = z)$ depends directly on the unobserved values. An example of data that are MNAR is income, in which more wealthy subjects are less likely to report their income in surveys than less wealthy subjects are. Distinguishing MAR from MNAR is generally not possible based on the observed data alone and must therefore rely on study design, documentation of data-collection procedures, and domain knowledge.

The choice of an appropriate method for handling missing data depends on the assumed missingness mechanism. When data are plausibly MAR, inference can proceed without explicitly modelling the missingness process. Under this assumption, likelihood-based and imputation-based approaches provide valid statistical inference provided that the analysis models are correctly specified.

5.1 Multiple Imputation by Chained Equations

Multiple imputation differs from conventional complete-data analysis, by explicitly acknowledging uncertainty about unobserved values. Rather than filling in missing data once, multiple imputation replaces each missing value by several plausible draws, creating multiple completed versions of the dataset. These completed datasets are analysed separately

using a model of interest, and the resulting estimates are subsequently combined to reflect both sampling variability and uncertainty due to missingness. This modular structure, i.e. imputation, analysis, and pooling, forms the conceptual foundation of multiple imputation [38].

Let $Z = (Z_1, \ldots, Z_p)$ denote a collection of p features subject to missingness. For each feature Z_j, let Z_j^r and $Z_j^{\neg r}$ denote its observed and missing components, respectively. Let

$$Z^r = (Z_1^r, \ldots, Z_p^r), \quad Z^{\neg r} = (Z_1^{\neg r}, \ldots, Z_p^{\neg r})$$

denote the observed and unobserved portions of the data. The goal of multiple imputation is to generate draws from the predictive distribution

$$\mathbb{P}(Z^{\neg r} \mid Z^r),$$

which, under a MAR assumption and correct specification of the imputation model, is sufficient for valid inference, without explicit modelling of the missingness mechanism. Multiple imputation generates m completed datasets $Z^{(1)}, \ldots, Z^{(m)}$ by drawing

$$Z_{\neg r}^{(h)} \sim \mathbb{P}(Z^{\neg r} \mid Z^r), \quad h = 1, \ldots, m$$

and combining each draw with the observed data Z^r.

5.1.1 Fully Conditional Specification and the MICE Algorithm

In many applications, specifying a joint model for the full multivariate distribution of Z is impractical, particularly when features differ in scale, distribution, or measurement type. *Multiple Imputation by Chained Equations* (MICE) addresses this by specifying a sequence of conditional models rather than a single joint model, a strategy known as *fully conditional specification*.

Let $Z_{-j} = (Z_1, \ldots, Z_{j-1}, Z_{j+1}, \ldots, Z_p)$ denote all features except Z_j. The MICE framework specifies, for each feature Z_j, a conditional model of the form

$$\mathbb{P}(Z_j \mid Z_{-j}, \theta_j),$$

where θ_j denotes parameters specific to the conditional distribution of Z_j. These parameters are local to the corresponding conditional model and are not required to correspond to a factorisation of any underlying joint distribution of Z. This type of sampling, is known as the Gibbs sampler.

Missing values are imputed iteratively by cycling through the features $j = 1, \ldots, p$, conditioning on the most recent imputations of the remaining features. At iteration t, the algorithm successively draws

$$\theta_j^{[t]} \sim \mathbb{P}\left(\theta_j \mid Z_j^r, Z_{-j}^{[t-1]}\right),$$
$$Z_j^{\neg r,[t]} \sim \mathbb{P}\left(Z_j \mid Z_j^r, Z_{-j}^{[t-1]}, \theta_j^{[t]}\right),$$

where $Z_{-j}^{[t-1]}$ denotes the most recent imputations of the remaining features. By construction, previous imputations of Z_j influence current updates only indirectly through their relationship with the other features. Once the iterative procedure has stabilised, one complete dataset is obtained. Repeating the entire procedure generates multiple completed datasets, enabling statistical inference by combining results across imputations.

This procedure is summarised in Algorithm 7.

Algorithm 7 Multiple Imputation by Chained Equations (MICE).

Input: Incomplete dataset $Z = (Z_1, \ldots, Z_p)$ with missingness pattern r.
Initialization: Initialize $Z_j^{\neg r,[0]}$ for all j.
for h=1, ..., m **do**
  for $t = 1, \ldots, T$ **do**
    for $j = 1, \ldots, p$ **do**
      Draw $\theta_j^{[t]} \sim \mathbb{P}(\theta_j \mid Z_j^r, Z_{-j}^{[t-1]})$
      Draw $Z_j^{\neg r,[t]} \sim \mathbb{P}(Z_j \mid Z_j^r, Z_{-j}^{[t-1]}, \theta_j^{[t]})$
      Update $Z_j^{[t]} \leftarrow (Z_j^r, Z_j^{\neg r,[t]})$
    end for
  end for
end for
Output: m completed dataset $Z_{imp} = \{Z^{(1),[T]}, \ldots, Z^{(m),[T]}\}$.

The specification of the imputation model is an important step in multiple imputation and often represents a methodological challenge. Implementing the MICE algorithm requires a series of modelling choices that determine how missing values are imputed and how uncertainty is propagated through the analysis. In the MICE framework, an imputation model is specified separately for each feature with missing values. For each incomplete feature, a univariate imputation model must be defined. The choice of imputation model depends on several considerations, such as the scale and distributional characteristics of the feature to be imputed. At each iteration of Algorithm 7, the univariate imputation procedure takes as input a set of predictors that are treated as complete at that stage and returns a single imputed value for each missing entry in the target feature. In practice, the specification of an imputation model involves several decisions. Notably seven choices should be considered when implementing MICE:

1. Whether the MAR assumption is plausible.
2. The form of imputation model.
3. The set of features to include as predictors into the imputation model.
4. Whether features, that are functions of other (incomplete) features, should be imputed.
5. The order in which features should be imputed.
6. The setup of the starting imputations, e.g. mean or median, and the number of iterations, T.
7. The number of imputed datasets, m.

With respect to the form of the imputation model in Item 2, MICE supports a range of univariate imputation techniques tailored to different feature types and measurement scales.

6 | Application

In this chapter, the theoretical framework presented in the preceding chapters is applied in practice. First, the dataset used in this study is introduced, followed by a description of the data preprocessing and cleaning procedures undertaken to prepare the data for imputation. Next, the modelling approach is outlined. Finally, the results of the analysis are presented and discussed.

The UK Biobank (UKB) RAP was temporarily closed on the 23rd of April and, at the time of writing, access has not yet been restored. Consequently, it was not possible to conduct analyses using the original dataset. Instead, all downstream analyses were performed using simulated data. These simulations were informed by summary statistics and percentile information available through the UKB Showcase [36]. As a result, the majority of the subsequent figures and tables are based either on these simulated datasets or on findings reported in previous studies investigating C282Y homozygosity within the UKB cohort.

A detailed description of the UKB dataset is provided in Section 6.1, while the methodology used for generating the simulated data is presented in Subsection 6.1.1. The modelling section remains unchanged despite the use of simulated data. Subsection 6.2.2 primarily discusses the applicability of multiple imputation by chained equations (MICE) to the UKB cohort. Section 6.3 presents the results of the models on the simulated data.

6.1 Data

The UKB is a large general population study established to investigate genetic, environmental, and lifestyle determinants of a wide range of health outcomes. Between 2006 and 2010, the study recruited 503,317 participants across England, Scotland, and Wales. Eligible individuals were 40-69 years of age at the time of recruitment and lived within approximately 40 km of one of 22 dedicated assessment centres, where all baseline assessments were conducted. A map of baseline assessment centres is shown in Figure 6.1. Approximately 9.2 million individuals were invited, of whom 5.5% enrolled [35]. Baseline assessments included structured questionnaires, standardised clinical measurements, and biological sample collection. Participants provided blood, urine, and saliva samples, which form the basis of UKB's large-scale biochemistry, proteomics, metabolomics, and infectious disease marker datasets. In addition, genome-wide genotyping was performed for the full cohort, supplemented by *whole-exome* (WES) and *whole-genome sequencing* (WGS) initiatives that, as of 2025, cover nearly all participants [34].

Although the cohort includes individuals across a wide range of educational and occupational backgrounds, UKB participants differ from the general UK population. The cohort demonstrates a *healthy volunteer bias*, with participants being healthier and less socio-economically deprived than the general population, at the time of recruitment. This is reflected in multiple comparative indicators, e.g., participants were less likely to be obese, to smoke, or to consume alcohol daily, and reported fewer self-reported health conditions than nationally representative samples. Long-term follow-up has shown substantially lower health event rates among participants compared with national averages, for example, at ages 70-74, all-cause mortality was 46.2% lower in men and 55.5% lower in women, while total cancer incidence was 11.8% lower in men and 18.1% lower in women than in the general population [15]. Demographically, the cohort contains 54% females and 46% males and the ethnic composition is broadly comparable to that of the UK population at the time of baseline recruitment. As shown in Table 6.1, 94.6% of UKB participants self-identified as

White, closely matching the 94.5% reported in the 2001 UK Census for adults aged 40-69 years, and only modestly higher than the 91.3% reported in the 2011 Census for the same age range. The 2021 Census shows a continued gradual diversification of the population in this age group, but the UKB cohort remains broadly aligned with the demographic profile observed at baseline recruitment.

For the present study, analyses were restricted to participants with successfully completed WES, reducing the cohort to 468,878 individuals. From the WES data, variants in the HFE gene were extracted to classify participants according to carrier status for the `C282Y`, `H63D`, and `S65C` mutations, which serves as response variables in the downstream modelling analyses.

Table 6.1: Ethnic group distribution for individuals aged 40-69 years in UKB and the 2001, 2011 and 2021 UK Censuses.

Ethnic group	**UKB**[a]		**2001 Census**[a]		**2011 Census**[a]		**2021 Census**[b]	
	No.	**%**	**No.**	**%**	**No.**	**%**	**No.**	**%**
Total	499,877	—	20,198,307	—	23,146,612	—	22,671,424	—
White	472,837	94.6	19,085,322	94.5	21,133,317	91.3	20,473,139	90.3
Black or Black British	8,066	1.6	302,073	1.5	565,777	2.4	598,864	2.6
Mixed	2,958	0.6	82,389	0.4	191,085	0.8	183,824	0.8
Indian	5,951	1.2	325,651	1.6	442,338	1.9	483,579	2.1
Pakistani	1,837	0.4	147,695	0.7	239,166	1.0	256,541	1.1
Bangladeshi	236	0.0	46,220	0.2	75,919	0.3	86,383	0.4
Chinese	1,574	0.3	70,572	0.3	109,412	0.5	105,385	0.5
Other Asian	1,858	0.4	73,917	0.4	240,324	1.0	212,297	0.9
Other ethnic group	4,560	0.9	64,468	0.3	149,274	0.6	271,412	1.2

[a] UKB ethnicity counts and Census 2001 and 2011 values are taken from [15, Table 1].
[b] 2021 Census ethnicity values are taken from dataset RM032 in [26].

The feature `Ethnic Background` is excluded from the modelling, due to concerns that it could exert disproportionate influence on the predictions. Moreover, it is a self-reported variable, meaning that it depends on how individuals identify themselves, which may introduce inconsistencies, for example, grouping individuals across different generations of immigrants under the same category. The dataset comprises 63 features and three response variables, as summarized in Table 6.2. Each response variable represents one of four groups, with identical factor levels across groups, namely non-carrier, homozygous, heterozygous or compound heterozygous.

The features can be categorized into three subgroups: demographics, biochemistry markers, and haematological assays. The biochemistry biomarkers represent a set of clinical measures collected at baseline, for all participants. These biomarkers were selected for their established role as disease risk factors, their diagnostic utility, or their ability to capture physiological traits not otherwise well characterized, while remaining feasible for large-scale assessment [36, ID: 17518]. The haematological assays include measurements obtained from whole blood prior to processing, capturing characteristics of blood cell composition and morphology [36, ID: 100081].

All features listed in Table 6.2 are derived from blood tests. Restricting the input variables to such measurements ensures that the pre-screening model remains applicable in standard clinical and public health settings.

Different combinations of feature subsets will be evaluated, in order to investigate whether blood tests can be restricted to a single type. The different feature combinations considered are

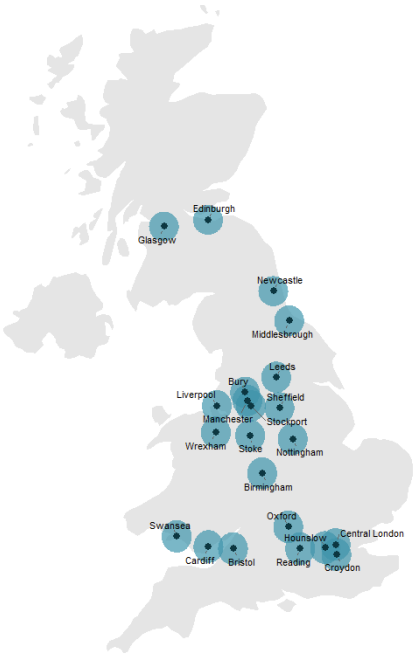


Figure 6.1: Map of the UK showing UKB baseline assessment centres.

- Demographics, Biochemistry markers and Haematological assays.
- Demographics and Biochemistry markers.
- Demographics and Haematological assays.

These features, grouped and composed in different ways, form the input variables for the pre-screening models evaluated. Demographics are included in all combinations, because there are well-documented differences in biomarkers and haematological measures between men and women [20], some of these differences are presented in Table B.1 in Appendix B. The different feature combinations are denoted `Full`, `Bio` and `Haema` in the following model analysis.

To address missing values in the dataset, several preprocessing steps were undertaken. First, all observations with missing values in the response variable `C282Y` were identified and removed. Subsequently, the distribution of missingness across all features was examined and visualized. For an overview, see Table B.2 in Appendix B. As the variables `Rheumatoid factor` and `Oestradiol` exhibited substantial proportions of missing values, namely 91.60% and 84.43%, respectively, these features were excluded from further analysis [2]. Imputing such highly incomplete variables would likely introduce bias due to insufficient information in the remaining data.

6.1.1 Simulation of Data

Due to the lack of access to the original UKB data, a synthetic dataset was generated to approximate the structure and distributional properties of the true cohort. The simulation strategy was based on summary statistics, specifically reported quantiles obtained from the UKB Showcase [36].

For continuous variables, reported quantiles were used to construct smooth quantile functions via monotone spline interpolation. Synthetic observations were then generated

Table 6.2: Feature overview.

Response variables		
C282Y	H63D	S65C
Features		
Demographics		
Age of recruitment	Sex	Ethnic Background
Biochemistry markers[a]		
Alanine aminotransferase	Albumin	Alkaline phosphatase
Apolipoprotein A	Apolipoprotein B	Aspartate aminotransferase
C-reactive protein	Calcium	Cholesterol
Creatinine	Cystatin C	Direct bilirubin
Gamma glutamyltransferase	Glucose	HbA1c
HDL cholesterol	IGF-1	LDL direct
Lipoprotein A	Oestradiol	Phosphate
Rheumatoid factor	SHBG	Testosterone
Total bilirubin	Total protein	Triglycerides
Urate	Urea	Vitamin D
Haematological assays[b]		
Basophill count	Basophill percentage	Eosinophill count
Eosinophill percentage	Haematocrit percentage	Haemoglobin concentration
High light scatter reticulocyte count	High light scatter reticulocyte percentage	Immature reticulocyte fraction
Lymphocyte count	Lymphocyte percentage	Mean corpuscular haemoglobin
Mean corpuscular haemoglobin concentration	Mean corpuscular volume	Mean platelet (thrombocyte) volume
Mean reticulocyte volume	Mean sphered cell volume	Monocyte count
Monocyte percentage	Neutrophill count	Neutrophill percentage
Nucleated red blood cell count	Nucleated red blood cell percentage	Platelet count
Platelet crit	Platelet distribution width	Red blood cell (erythrocyte) count
Red blood cell (erythrocyte) distribution width	Reticulocyte count	Reticulocyte percentage
White blood cell (leukocyte) count		

[a] An overview can be seen on UKB Showcase with ID 17518 [36].
[b] An overview can be seen on UKB Showcase with ID 100081 [36].

by sampling from a uniform distribution on [0,1] and mapping these values through the corresponding quantile functions. Demographic variables were simulated separately. Age was generated using the same quantile-based interpolation approach, while sex was simulated as a binary variable according to the observed proportion in UKB.

Genetic variants were simulated under the assumption of *Hardy-Weinberg equilibrium* (HWE), a principle in population genetics describing the relationship between allele and genotype frequencies in a large randomly mating population. Under HWE, and in the absence of evolutionary forces such as selection, mutation, migration, or genetic drift, genotype frequencies are determined directly from the allele frequency. For a locus with minor allele frequency p, the genotype probabilities are given by

$$\begin{aligned}\Pr(G = 0) &= (1-p)^2,\\ \Pr(G = 1) &= 2p(1-p),\\ \Pr(G = 2) &= p^2,\end{aligned}$$

where G denotes the number of minor alleles. These probabilities sum to one and define

a categorical distribution over the three possible genotypes [25]. The allele frequency for the HFE variant `C282Y`, $p = 0.0775$, was used to parametrise the genotype simulation. Genotypes were assigned independently by sampling from the corresponding distribution.

Biochemical and haematological measurements were simulated independently using the quantile-based approach described above. For variables known to differ substantially by sex, such as testosterone, sex-specific distributions were simulated separately. Furthermore, the variables `Rheumatoid factor` and `Oestradiol` were excluded from the simulated data, due to the high proportion of missingness. `Nucleated red blood cell count` and `Nucleated red blood cell percentage` were also not simulated because of the lack of available quantiles.

The final simulated dataset consists of $n = 501{,}938$ individuals and includes demographic, genetic, biochemical, and haematological variables generated independently from approximated marginal distributions of the UKB cohort. While this approach approximately reproduces the univariate distributions, it does not preserve the multivariate dependence structure and correlations present in the original data, which should be considered when interpreting the results.

Missing values were introduced by randomly masking observations according to the variable-specific missingness proportions reported in Table B.2 in Appendix B. This was done to mimic the missing-data structure of the original cohort and to enable evaluation of the MICE imputation procedure on the simulated dataset.

6.2 Modelling

This section describes the modelling workflow. The complete procedure is summarized in the flowchart presented in Figure 6.3.

Prior to imputing missing values, the feature combination was specified, and the dataset was partitioned into training (70%), validation (15%), and test (15%) subsets. The split was performed prior to imputation in order to prevent data leakage, ensuring that information from the validation and test sets does not influence the imputation model. This resulted in 351,357 observations in the training set and 75,291 observations in both the validation and test sets.

Missing values were handled using the `mice` package in `R`. Imputation is performed on the training data to estimate the conditional imputation models, generating $m = 5$ imputed datasets. Subsequently, the fitted imputation model was applied to the validation and test sets using the `mice.mids` function.

To address the class imbalance in the `C282Y` homozygous mutation, bootstrap aggregation (bagging) was employed. The class imbalance in the entire cohort can be seen in Figure 6.2. Specifically, multiple balanced training sets were constructed by combining all observations from the minority class with subsamples of the majority class, such that each resampled dataset contained an equal number of observations from each class. In total, 25 subsamples of the majority class were generated, resulting in 25 distinct downsampled training sets. A separate model was trained on each of these datasets, and predictions for the validation and test sets were obtained by averaging the predicted probabilities across all models. This approach preserves the information contained in the majority class while reducing the risk of overfitting.

The bagging procedure was applied independently to each of the $m = 5$ imputed datasets, resulting in a total of 125 training datasets, to reflect the uncertainty of missing data and obtain more robust predictions.

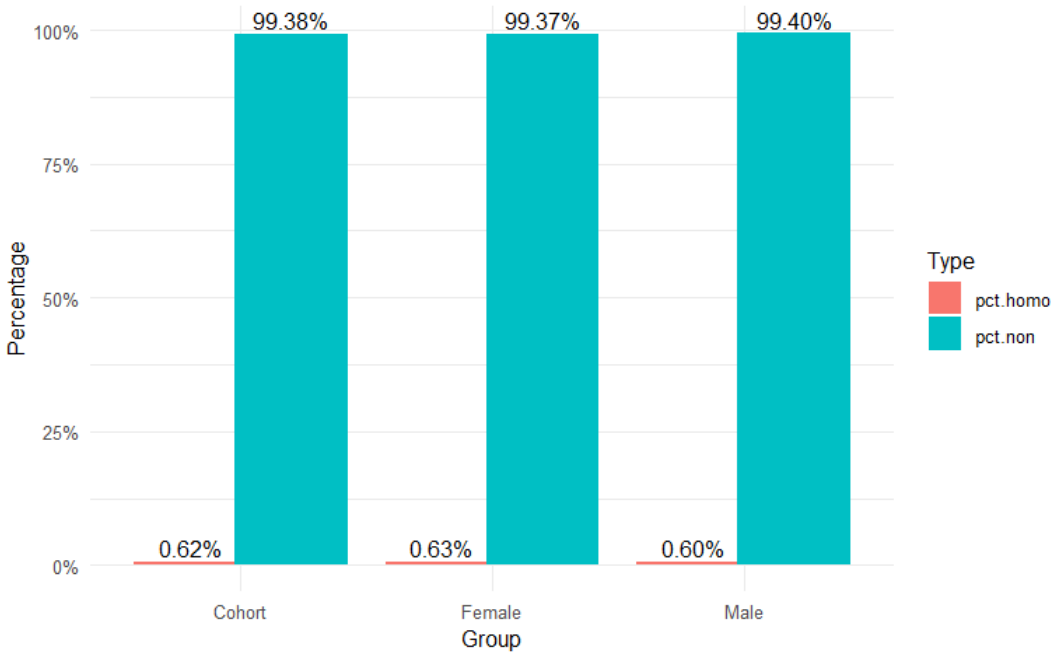


Figure 6.2: Percentage of `C282Y` homozygous within cohort and stratified by `Sex`, based on [22, Table 1].

As a baseline model, logistic regression was employed. The model was trained using the bagging and imputation framework as described above, for a direct comparison with the XGBoost models introduced below. An interaction term on the feature `Sex` was included, in order to determine whether there is an inherent underlying difference between sexes.

For the predictive modelling task, XGBoost was employed using a logistic objective. To account for class imbalance, the training procedure was combined with the bagging strategy described above, such that each model was fitted on a balanced subsample of the data.

Hyperparameter tuning was performed using grid expansion combined with 5-fold cross-validation on each downsampled training set. The hyperparameters considered were the learning rate (η), maximum tree depth, subsampling ratio, column sampling ratio, and the regularization parameter λ. The search space consisted of 162 distinct hyperparameter combinations, defined as

$$\begin{aligned} \eta &\in \{0.01, 0.05, 0.1\}, \\ \lambda &\in \{1, 2.5, 5\}, \\ \texttt{max_depth} &\in \{6, 8\}, \\ \texttt{subsample} &\in \{0.7, 0.8, 1.0\}, \\ \texttt{colsample_bytree} &\in \{0.7, 0.8, 1.0\}. \end{aligned}$$

For each combination, model performance was evaluated using the cross-validated log-loss, and the optimal number of boosting iterations was determined using early stopping, where the maximum number of iterations was set to 500.

To account for variability induced by the downsampling procedure, the tuning process was repeated 25 times with different subsamples of the training data. The final hyperparameter combination for each imputed dataset was obtained by aggregating the best-performing combination across repetitions by selecting most frequent combinations.

Using the selected hyperparameters, models were subsequently trained on each of the 25 downsampled datasets per imputation. Predictions for the validation sets were obtained for each model, and the final predicted probabilities were computed by averaging across all models within each imputed dataset. This procedure was repeated for all five imputed datasets, ensuring that both sampling variability and imputation uncertainty were incorporated into the predictions.

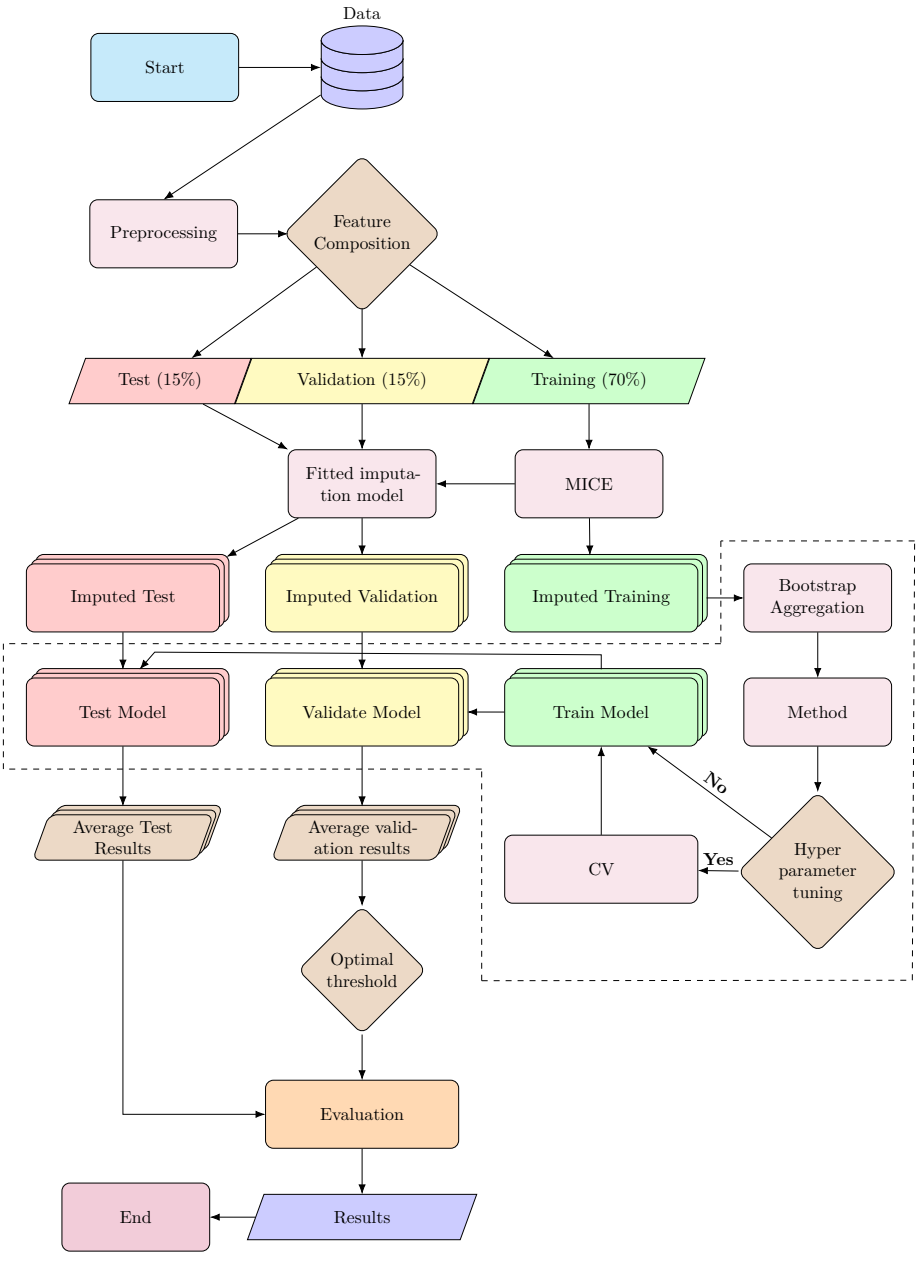


Figure 6.3: Flowchart of application. Dashed lines indicates workflow for several iterations.

The averaged predicted probabilities were then used to determine the optimal classification threshold. For the cost function, $\mathcal{C}$, introduced in Section 3.2, threshold selection was performed using a ranked grid search over the predicted probabilities. In addition the threshold optimization problem was formulated as an MILP, allowing the incorporation of explicit constraints on performance measures. Lastly, a comparison of model performance

between optimal classification threshold and the cost-optimal threshold, t^*, introduced in Eq. (3.4), is made. The cost-optimal threshold used is based on the cost matrix in Table 3.2, which yields

$$t^* = \frac{C_{FP} - C_{TN}}{(C_{FP} - C_{TN}) + (C_{FN} - C_{TP})} = \frac{56 - 0}{(56 - 0) + (248 - 232)} = 0.78.$$

6.2.1 MILP Formulation

As described in Section 3.2, the cost function is

$$\mathcal{C}(t) = q_{TP}(t)C_{TP} + q_{TN}(t)C_{TN} + q_{FP}(t)C_{FP} + q_{FN}(t)C_{FN}.$$

The objective is to determine the threshold $t \in [0,1]$ that minimizes the total cost while satisfying the constraints in Eq. (3.6).

To obtain an MILP formulation, the binary decision variables $\widehat{y}_i \in \{0,1\}$ are introduced to represent the predicted class labels induced by the threshold, applied to the predicted probabilities $\widehat{p}_i$. Through this construction, the empirical counts in $\mathcal{C}(t)$ are represented directly in terms of the decision variables.

The relationship between the threshold and the predicted labels is enforced using a Big-M formulation

$$\begin{aligned} &\widehat{p}_i - t \geq -M(1 - \widehat{y}_i), \\ &\widehat{p}_i - t \leq M\widehat{y}_i, \qquad i = 1, \ldots, n, \end{aligned}$$

which ensures consistency between $\widehat{y}_i$ and the threshold-based decision rule. Since $\widehat{p}_i, t \in [0, 1]$, the choice $M = 1$ is sufficient.

The sensitivity constraint imposed in Eq. (3.6) is written in terms of the decision variables using

$$q_{TP}(t) = \sum_{i=1}^{n} \widehat{y}_i y_i,$$

leading to the linear constraint

$$\sum_{i=1}^{n} \widehat{y}_i y_i \geq 0.95 \sum_{i=1}^{n} y_i.$$

Similarly, the total cost $\mathcal{C}(t)$ can be expressed as a linear function of $\widehat{y}_i$, such that

$$\begin{aligned} \mathcal{C}(t) &= \widehat{y}_i y_i C_{TP} + (1 - \widehat{y}_i)(1 - y_i)C_{TN} + \widehat{y}_i(1 - y_i)C_{FP} + (1 - \widehat{y}_i)y_i C_{FN} \\ &= \widehat{y}_i y_i \alpha + \widehat{y}_i \beta + \underbrace{y_i(C_{FN} - C_{TN}) + C_{TN}}_{\text{constant}}, \end{aligned}$$

where

$$\begin{aligned} \alpha &= C_{TP} + C_{TN} - C_{FP} - C_{FN}, \\ \beta &= C_{FP} - C_{TN}. \end{aligned}$$

Since the final term is independent of the decision variables, it can be incorporated into the right-hand side of the cost constraint in Eq. (3.6). Let C_S represent the fixed operational

cost of maintaining the deployed prediction system, including the server infrastructure required for implementation. In this setting, the infrastructure cost is estimated at

$$C_S = 8{,}932£, \tag{6.1}$$

which reflects the cost of hosting and maintaining the decision support system. Let C_A denote the total cost associated with screening the entire cohort without preselection. Given a cohort size of the validation and test set is 75,291 individuals and a per-person genotyping cost of 56£, the total cost of universal screening amounts to

$$C_A = 75{,}291 \cdot 56£ = 4{,}216{,}296£. \tag{6.2}$$

The cost constraint ensures that the model-based screening strategy does not exceed the cost of universal screening after accounting for implementation costs. The resulting MILP formulation is

$$\begin{aligned}
\underset{t,\,\widehat{y}_i}{\text{minimize}} \quad & \sum_{i=1}^{n} \left[\widehat{y}_i(y_i\alpha + \beta)\right] \\
\text{subject to} \quad & \sum_{i=1}^{n} \widehat{y}_i y_i \geq 0.95 \sum_{i=1}^{n} y_i, \\
& \sum_{i=1}^{n} \left[\widehat{y}_i(y_i\alpha + \beta)\right] \leq C_A - C_S - \sum_{i=1}^{n} \left[y_i(C_{FN} - C_{TN}) + C_{TN}\right], \\
& \widehat{p}_i - t \geq -(1 - \widehat{y}_i), \quad i = 1, \ldots, n, \\
& \widehat{p}_i - t \leq \widehat{y}_i, \quad i = 1, \ldots, n, \\
& t \in [0,1], \qquad \widehat{y}_i \in \{0,1\}.
\end{aligned}$$

This formulation provides a direct implementation of the cost minimization problem in Eq. (3.6), allowing the inclusion of both clinical and economic constraints, by using sensitivity and cost, respectively.

6.2.2 MICE Assumptions and Implementation

In order to justify the use of MICE for handling missing data, arguments for Items 1-7 on Page 26 must be made. These arguments are grounded in the structure of the UKB data, the preprocessing steps undertaken, and the intended downstream modelling workflow.

Firstly, regarding the MAR assumption, blood count and blood biochemistry measurements in UKB were collected with the intention of universal assessment, and missingness primarily arises from pre-analytical, logistical, and laboratory-related factors such as insufficient sample volume, assay failure, or quality-control exclusion [2]. These missingness mechanisms are related to study procedures and measurement processes rather than to participants' underlying biomarker concentrations. Consequently, it is less plausible that missingness depends directly on the unobserved biomarker value itself. On this basis, the MAR assumption is adopted as a reasonable working assumption for the present analysis.

Secondly, the specification of the univariate imputation models, the choice of predictors, and the treatment of derived features were determined jointly as part of a staged preprocessing and modelling workflow. Initially, the distribution of missingness across all features was examined, leading to the exclusion of `Rheumatoid factor` and `Oestradiol`, which exhibited missingness levels exceeding 80%. Imputing these features would have relied almost entirely on model-based assumptions with limited empirical support and therefore posed a risk of bias. Excluding these features reduced model complexity while preserving

features with sufficient observed information. For the entire overview of missingness per feature, see Table B.2 in Appendix B.

For the remaining features, *predictive mean matching* (PMM) was selected as the univariate imputation model, as the retained predictors consisted primarily of continuous numerical measurements derived from routine blood tests. PMM preserves the empirical distribution of observed values by imputing missing entries using observed neighbouring values with similar predicted means, thereby avoiding implausible parametric extrapolation. This is appropriate in biomedical data, where departures from normality and extreme values may be clinically meaningful. The only categorical feature retained in the dataset was `Sex`, which contained no missing values and therefore required no imputation.

To mitigate multicollinearity and preserve known relationships, features representing "percentages" in Table 6.2, were excluded from the imputation step. The features were excluded since they are deterministic functions of other features, and were therefore reconstructed after imputation using the imputed features. For an overview of the formulas used to calculate the "percentage" relations, see UKB Showcase under ID 100081 [36].

A custom imputation order was applied such that features with the lowest proportion of missing values were imputed first, rather than relying on the default left-to-right cycling. Initial imputed values were generated using the default initialization procedures in `mice`, which draw on observed marginal distributions. The number of iterations was set to $T = 40$, to ensure stability of the imputations while maintaining computational efficiency, which is appropriate given the limited degree of missingness after preprocessing. The number of datasets was set to $m = 5$, which is consistent with common practice in applied multiple imputation settings [29]. Lastly, imputation models were estimated exclusively on the training data to prevent information leakage. The fitted imputation object was subsequently applied to the validation and test sets using the `mice.mids` function, ensuring that imputations for these datasets were based solely on the structure learned from the training data.

6.3 Results

As noted previously, the simulated nature of the data used in the application limits the reliability and validity of the resulting estimates. Consequently, the numerical results should not be interpreted as representative or conclusive. Nevertheless, the primary purpose of the application is to illustrate and evaluate the proposed methodology and the manner in which the results would be expected to align under an appropriate data-generating setting.

As illustrated in Figure 6.3, the imputation procedure using `mice` is initiated after the preprocessing stage and the selection of the feature combination. As described in Subsection 5.1.1, a completed dataset is obtained once the iterative imputation procedure has converged to a stable state. To assess convergence and stability, the mean and standard deviation of some feature coefficients are visualized in Figure 6.4 and Figure 6.5.

The iterative behaviour of the `Full` feature set is compared with the reduced feature sets to assess differences in convergence behaviour.

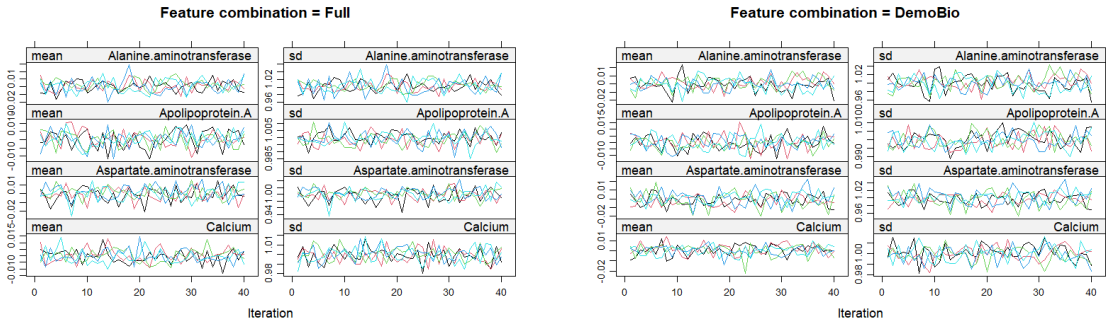


Figure 6.4: Comparison of `Full` vs `Bio`. The figure presents iterative plots from the `mice` procedure for different feature coefficients. The left panel illustrates the procedure under the `Full` feature combination, while the right panel corresponds to the `Bio` feature combination.

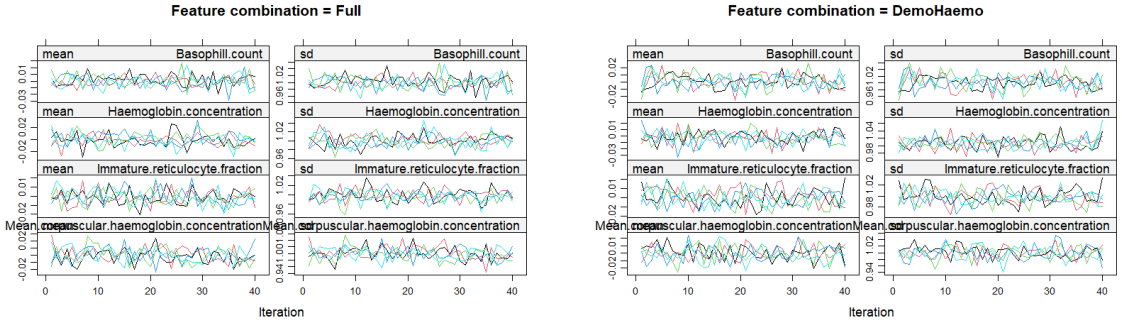


Figure 6.5: Comparison of imputed features `Full` vs. `Haema`. The figure presents iterative plots from the `mice` procedure for different feature coefficients. The left panel illustrates the procedure under the `Full` feature combination, while the right panel corresponds to the `Haema` feature combination.

Based on the iterative trajectories shown in Figure 6.4 and Figure 6.5, broadly consistent patterns emerge regarding the convergence behaviour of the `mice` imputation procedure across feature sets. In all cases, the chains appear to stabilise after a relatively small number of iterations, with no clear systematic trends, suggesting that convergence is achieved.

When comparing the `Full` feature combination to the reduced `Haema` and `Bio` subsets, the differences in convergence behaviour are limited. While the `Full` specification may exhibit slightly smoother trajectories and somewhat reduced variability in certain features, there is no strong visual evidence that convergence is consistently achieved more rapidly. Instead, all feature combinations appear to reach a stable region within a similar number of iterations.

This suggests that incorporating a larger set of features may contribute to marginally improved stability of the imputation process, but does not lead to a substantial improvement in convergence speed. The overall behaviour is therefore consistent with the expectation that richer conditioning information can stabilise the conditional models in `mice`, while the rate of convergence remains broadly comparable across specifications.

The remaining plots for the other features can be found in Appendix C.

In addition to the assessment of the imputation procedure, the results of the hyperparameter tuning provide insight into how the predictive models adapt to different feature sets. The selected XGBoost combinations across imputations and feature combinations are presented in Table 6.3. These results reflect the aggregation of optimal combinations obtained from repeated cross-validation and subsampling, as described in the modelling section.

Table 6.3: Optimal XGBoost hyperparameters across imputations and feature sets.

$\texttt{Feature_set}_{\text{imp}}$	n_iter	max_depth	η	λ	subsample	colsample_bytree
$\texttt{Full}_1$	24	8	0.05	5.0	0.8	0.8
$\texttt{Full}_2$	9	6	0.10	1.0	0.8	0.8
$\texttt{Full}_3$	6	8	0.05	5.0	0.8	0.8
$\texttt{Full}_4$	9	8	0.05	1.0	0.8	0.8
$\texttt{Full}_5$	16	8	0.05	1.0	0.8	0.8
$\texttt{Bio}_1$	34	6	0.05	1.0	0.8	0.8
$\texttt{Bio}_2$	24	6	0.05	5.0	0.8	0.8
$\texttt{Bio}_3$	17	6	0.05	1.0	0.8	0.8
$\texttt{Bio}_4$	51	6	0.01	1.0	0.8	0.8
$\texttt{Bio}_5$	37	8	0.01	1.0	0.8	0.8
$\texttt{Haema}_1$	9	6	0.10	5.0	1.0	0.7
$\texttt{Haema}_2$	14	6	0.10	2.5	0.7	0.8
$\texttt{Haema}_3$	11	8	0.10	2.5	0.8	0.8
$\texttt{Haema}_4$	11	8	0.05	1.0	0.8	0.8
$\texttt{Haema}_5$	18	8	0.10	1.0	0.8	0.8

Across all feature sets, the subsampling parameters `subsample` and `colsample_bytree` exhibit a high degree of consistency, with values close to 0.8 dominating. This suggests that moderate stochasticity in both the observation and feature dimensions is consistently favoured across feature combinations.

More distinct differences are observed in the structural and optimisation-related hyperparameters. For the `Full` feature set, deeper trees are consistently selected for the majority of imputations. This suggests that deeper interaction structures may be beneficial when the full feature set is available. The learning rate is typically moderate with $\eta = 0.05$, while the regularisation parameter varies between $\lambda = 1$ and $\lambda = 5$, indicating that some degree of shrinkage is necessary in more complex combinations.

In contrast, the `Bio` feature set generally favours more shallow trees and smaller learning rates with η ranging from 0.01 to 0.05. This could indicate a more conservative learning process, which may be required due to increased noise or weaker signal strength in the biological markers. The regularisation parameter is predominantly $\lambda = 1$, with occasional increases, suggesting that overfitting remains a concern in this setting.

The `Haema` feature set displays intermediate behaviour. Here both tree depths, 6 and 8, are represented, and the learning rate is more frequently higher, with $\eta = 0.10$, implying that faster optimisation is feasible without compromising generalisation. Additionally, intermediate regularisation values $\lambda = 2.5$ appear in multiple cases, reflecting a balance between model flexibility and stability. Slight deviations in subsampling like, e.g. values of 0.7 or 1.0, further indicate a modest sensitivity to sampling variation in this feature set.

Overall, these results demonstrate that although certain hyperparameters remain stable across all combinations, the optimal balance between model complexity, learning rate, and

regularisation is dependent on the underlying feature set. This behaviour could stem from differences in information content and noise structure between the `Full`, `Bio`, and `Haema` feature combinations.

6.3.1 Results - Full

For each modelling approach, logistic regression and XGBoost, each observation yields five distinct predicted probabilities due to the multiple imputations generated by the MICE procedure. To assess whether the imputations lead to systematically different predictions, the Wilcoxon rank-sum test is applied to all pairwise combinations of imputation-specific predicted probabilities. This non-parametric test is used to evaluate whether the distributions of predicted probabilities differ significantly across imputations without imposing parametric assumptions, such as normality. This provides a method of investigating potential systematic variation introduced by the imputation process. In addition, such variation may reflect the influence of the underlying missing-data structure, including observations with a high degree of missingness, which may induce differences in the imputed values and, consequently, in the resulting predictions.

Table 6.4: Wilcoxon Rank Sum Test on feature combination `Full`.

Group 1	Group 2	p-val. LogReg	p-val. XGB
1	2	0.00	0.00
1	3	0.00	0.00
1	4	0.00	0.00
1	5	0.00	0.04
2	3	0.00	0.54
2	4	0.01	0.00
2	5	0.50	0.00
3	4	0.00	0.01
3	5	0.00	0.00
4	5	0.20	0.00

For the generated data, the Wilcoxon rank-sum tests indicate that the predicted probability distributions differ significantly across most pairs of imputations for both logistic regression and XGBoost, as seen in Table 6.4. In particular, the majority of pairwise comparisons yield p-values below the 0.05 significance level, suggesting that the imputation-specific predictions are not drawn from identical distributions. This pattern is consistent across both modelling approaches. Nevertheless, the overall results suggest that the multiple imputation procedure introduces variability in the resulting predictions across imputations.

In Table 6.5, the results obtained from the validation set are presented for the `Full` feature combination. Note that Model_i specifies whether the model is logistic regression or XGBoost, where $i = 1, \ldots, 5$ denotes the imputation number.

Across all models, sensitivity is consistently high at approximately 0.95, while specificity is very low, ranging between 0.04 and 0.07. This indicates that the models predominantly classify observations as positive, resulting in a large number of false positives. Despite this, precision is effectively zero in all cases, meaning that almost none of the predicted positives are correct. Consequently, the F_1 and F_2 scores remain close to zero, reflecting the poor balance between correctly identified positives and the large number of incorrect

positive predictions.

The threshold-independent metrics support this conclusion. Both logistic regression and XGBoost achieve ROC-AUC values close to 0.50 and PR-AUC values around 0.004, indicating performance comparable to random classification under severe class imbalance. Similarly, the Matthews correlation coefficient (MCC) is consistently equal to zero across all imputations and modelling approaches, suggesting no meaningful discriminatory power when jointly considering both classes.

Overall, neither logistic regression nor XGBoost is able to extract a meaningful predictive signal from the data under the `Full` feature specification. Although both models achieve high sensitivity, this is not accompanied by meaningful improvements in specificity or precision, resulting in poor overall classification performance.

The results in Table 6.5 indicate that the MILP-based thresholding approach does not lead to meaningful differences in predictive performance across models or imputations. The selected thresholds are similar across all combinations, approximately 0.36-0.37 for logistic regression and 0.46-0.49 for XGBoost, resulting in nearly identical performance metrics. Consequently, the additional computational complexity introduced by the MILP framework appears unjustified, as it does not translate into improved classification outcomes.

Table 6.5: Results obtained from the validation set, using feature combination `Full`.

Model	MILP	Threshold	Sensitivity	Specificity	Precision	F_1	F_2	MCC	ROC-AUC	PR-AUC	$\mathcal{C}$	$\mathcal{C} < C_A - C_S$
LogReg_1	**0.37**	**0.37**	0.95	0.07	0.00	0.01	0.02	0.00	0.495	0.004	**3,992,720**	TRUE
LogReg_2	0.37	0.37	0.95	0.06	0.00	0.01	0.02	0.00	0.503	0.004	4,030,632	TRUE
LogReg_3	0.36	0.36	0.95	0.05	0.00	0.01	0.02	0.00	0.501	0.004	4,036,792	TRUE
LogReg_4	0.37	0.37	0.95	0.06	0.00	0.01	0.02	0.00	0.495	0.004	3,996,920	TRUE
LogReg_5	0.37	0.37	0.95	0.06	0.00	0.01	0.02	0.00	0.497	0.004	4,015,232	TRUE
XGB_1	0.46	0.46	0.95	0.05	0.00	0.01	0.02	0.00	0.508	0.004	4,036,064	TRUE
XGB_2	**0.47**	**0.47**	0.95	0.06	0.00	0.01	0.02	0.00	0.514	0.004	**4,008,344**	TRUE
XGB_3	0.49	0.49	0.95	0.04	0.00	0.01	0.02	0.00	0.503	0.004	4,095,984	TRUE
XGB_4	0.48	0.48	0.95	0.05	0.00	0.01	0.02	0.00	0.514	0.004	4,040,824	TRUE
XGB_5	0.47	0.47	0.95	0.05	0.00	0.01	0.02	0.00	0.501	0.004	4,049,280	TRUE

The ROC and PR curves in Figure 6.7 and Figure 6.6 visually confirm the limited discriminatory ability observed in Table Table 6.5.

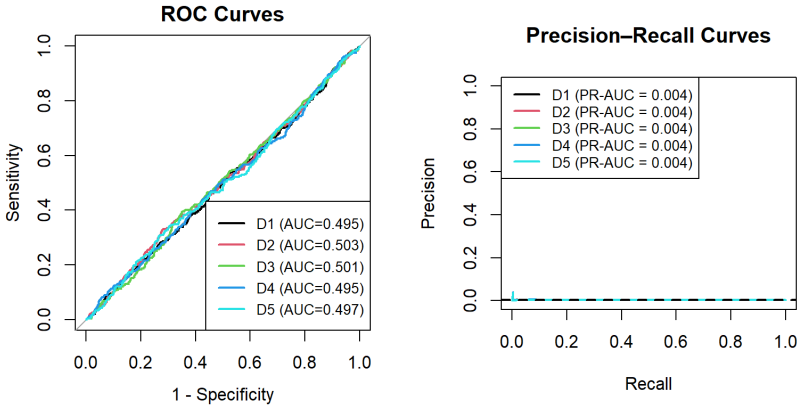


Figure 6.6: ROC-AUC and PR-AUC for Logistic regression models on validation sets with feature combination `Full`.

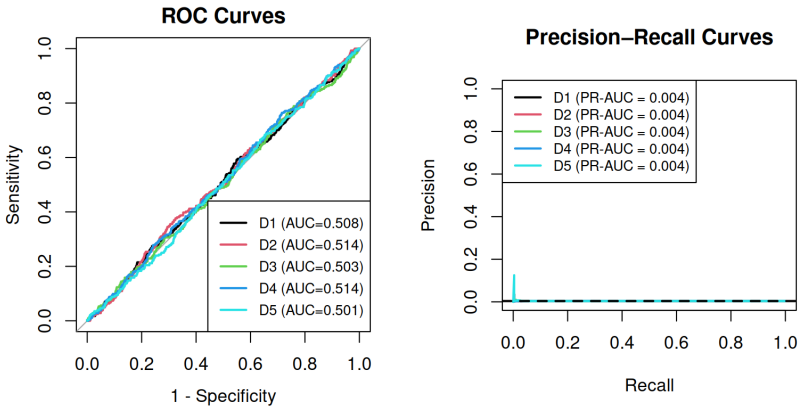


Figure 6.7: ROC-AUC and PR-AUC for XGBoost models on validation sets with feature combination `Full`.

In Table 6.6, the results obtained on the test set using the optimal threshold for the `Full` feature combination are presented. The thresholds selected on the validation set are $t_{\text{LogReg}} = 0.37$ and $t_{\text{XGB}} = 0.47$, corresponding to the solutions with the lowest cost in Table 6.5.

The test-set results largely mirror the validation performance, with consistently high sensitivity but negligible precision and discriminatory ability across imputations. Although some variation in specificity and total cost is observed, the overall predictive performance remains weak for both modelling approaches.

From an economic perspective, the total cost of screening the entire cohort is given in Eq. (6.2) and amounts to approximately 4.22 million £. The fixed infrastructure cost given in Eq. (6.1), is comparatively small at 8,932£. Together, these define an effective cost ceiling of approximately 4.21 million £, corresponding to the benchmark for universal screening adjusted for implementation costs. As shown in Table 6.6, most model combinations yield total costs in the range of approximately 3.87 to 4.03 million £, thereby remaining below this benchmark. Although modest cost reductions are observed relative to universal screening, these savings arise primarily from the imposed thresholding structure rather than meaningful predictive discrimination. It is worth noting that XGB_3 and XGB_4 do not satisfy the cost constraint.

Table 6.6: Results obtained from the test set, using optimal threshold, using feature combination `Full`.

Model	Sensitivity	Specificity	Precision	F_1	F_2	MCC	$\mathcal{C}$	$\mathcal{C} < C_A - C_S$
LogReg_1	0.94	0.06	0.00	0.01	0.02	0.00	3,998,976	TRUE
LogReg_2	0.94	0.05	0.00	0.01	0.02	0.00	4,039,280	TRUE
LogReg_3	0.95	0.06	0.00	0.01	0.02	0.00	4,022,968	TRUE
LogReg_4	0.95	0.06	0.00	0.01	0.02	0.00	4,030,584	TRUE
LogReg_5	0.93	0.05	0.00	0.01	0.02	0.00	4,037,872	TRUE
XGB_1	0.89	0.09	0.00	0.01	0.02	0.00	3,870,384	TRUE
XGB_2	0.94	0.06	0.00	0.01	0.02	0.00	4,005,584	TRUE
XGB_3	1.00	0.00	0.00	0.01	0.02	0.00	4,265,992	FALSE
XGB_4	0.99	0.01	0.00	0.01	0.02	0.00	4,224,696	FALSE
XGB_5	0.93	0.07	0.00	0.01	0.02	0.00	3,969,304	TRUE

Lastly, the results obtained using the cost-optimal threshold, $t^* = 0.78$, are presented

in Table 6.7. The high threshold leads to an extremely conservative classification rule, where few observations are predicted as positive. As a consequence, sensitivity collapses while specificity approaches one across all models.

Overall, the cost-optimal threshold substantially reduces screening costs, but at the expense of effectively eliminating predictive utility. As a consequence of the high threshold, no observations are predicted as positive for all XGBoost models. Hence sensitivity and all performance metrics are not computable.

Table 6.7: Results obtained from the test set, using cost-optimal threshold $t^* = 0.78$, on feature combination `Full`.

Model	Sensitivity	Specificity	Precision	F_1	F_2	MCC	$\mathcal{C}$	$\mathcal{C} < C_A - C_S$
LogReg_1	0.01	1.00	0.01	0.01	0.01	0.01	78,888	TRUE
LogReg_2	0.00	1.00	0.01	0.00	0.00	0.00	77,336	TRUE
LogReg_3	0.00	1.00	0.01	0.00	0.00	0.00	77,000	TRUE
LogReg_4	0.00	1.00	0.01	0.00	0.00	0.00	76,832	TRUE
LogReg_5	0.00	1.00	0.01	0.00	0.00	0.00	77,168	TRUE
XGB_1	NA	1.00	NA	NA	NA	NA	0	TRUE
XGB_2	NA	1.00	NA	NA	NA	NA	0	TRUE
XGB_3	NA	1.00	NA	NA	NA	NA	0	TRUE
XGB_4	NA	1.00	NA	NA	NA	NA	0	TRUE
XGB_5	NA	1.00	NA	NA	NA	NA	0	TRUE

6.3.2 Results - Biochemistry Markers

As in the `Full` feature setting, each observation yields five predicted probabilities due to the multiple imputations generated by the MICE procedure. To assess whether these imputations lead to systematically different predictions, the Wilcoxon rank-sum test is applied to all pairwise combinations of imputed predictions.

The results in Table 6.8 indicate that predicted probability distributions differ significantly across most pairs of imputations, particularly for logistic regression, where all comparisons yield statistically significant results. For XGBoost, a similar pattern is observed, although a small number of comparisons are not statistically significant. Overall, the imputation procedure introduces variability in the predicted probabilities, though the effect appears slightly less consistent than in the `Full` setting.

Table 6.8: Wilcoxon Rank Sum Test on feature combination `Bio`.

Group 1	Group 2	p-val. LogReg	p-val. XGB
1	2	0.00	0.00
1	3	0.00	0.00
1	4	0.00	0.21
1	5	0.00	0.00
2	3	0.00	0.00
2	4	0.00	0.00
2	5	0.03	0.01
3	4	0.00	0.00
3	5	0.00	0.36
4	5	0.00	0.00

The validation results in Table 6.9 largely mirror the patterns observed in the `Full` setting. Across all imputations and both modelling approaches, the models favour positive classifications, reflected in consistently high sensitivity and very low specificity. As a consequence, precision remains effectively zero, and all combined performance measures indicate negligible predictive utility. Threshold-independent metrics further corroborate the lack of discrimination, with ROC-AUC values fluctuating around 0.5 and PR-AUC remaining extremely low, as depicted in Figure 6.8 and Figure 6.9.

The MILP-based threshold selection again yields very similar thresholds across imputations, approximately 0.39-0.40 for logistic regression and 0.44-0.48 for XGBoost, resulting in identical performance. Hence, consistent with the `Full` setting, the additional optimisation step does not translate into meaningful improvements in predictive performance.

Table 6.9: Results obtained from the validation set, using feature combination `Bio`.

Model	MILP	Threshold	Sensitivity	Specificity	Precision	F_1	F_2	MCC	ROC-AUC	PR-AUC	$\mathcal{C}$	$\mathcal{C} < C_A - C_S$
$LogReg_1$	0.40	0.40	0.95	0.04	0.00	0.01	0.02	0.00	0.505	0.004	4,107,184	TRUE
$LogReg_2$	**0.40**	**0.40**	0.95	0.05	0.00	0.01	0.02	0.00	0.482	0.004	**4,074,536**	TRUE
$LogReg_3$	0.40	0.40	0.95	0.03	0.00	0.01	0.02	-0.01	0.488	0.004	4,149,632	TRUE
$LogReg_4$	0.39	0.39	0.95	0.03	0.00	0.01	0.02	-0.01	0.471	0.003	4,160,384	TRUE
$LogReg_5$	0.39	0.39	0.95	0.03	0.00	0.01	0.02	0.00	0.476	0.004	4,120,288	TRUE
XGB_1	0.44	0.44	0.95	0.06	0.00	0.01	0.02	0.00	0.504	0.004	4,002,968	TRUE
XGB_2	0.46	0.46	0.95	0.06	0.00	0.01	0.02	0.00	0.482	0.004	4,004,424	TRUE
XGB_3	**0.47**	**0.47**	0.95	0.07	0.00	0.01	0.02	0.00	0.483	0.004	**3,991,488**	TRUE
XGB_4	0.48	0.48	0.95	0.06	0.00	0.01	0.02	0.00	0.509	0.004	3,994,904	TRUE
XGB_5	0.48	0.48	0.95	0.05	0.00	0.01	0.02	0.00	0.492	0.004	4,044,240	TRUE

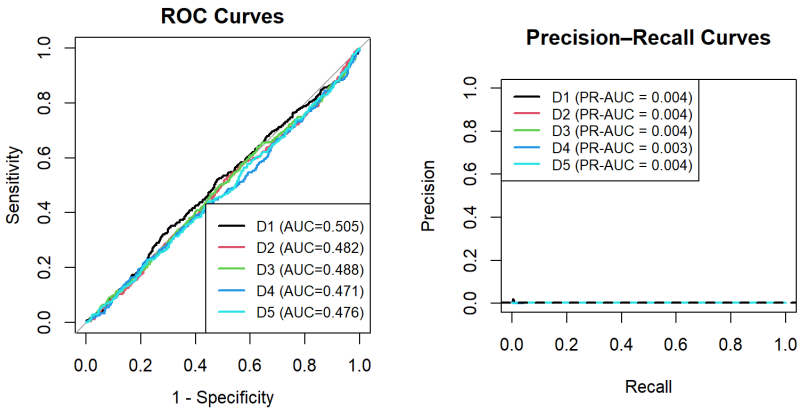


Figure 6.8: ROC-AUC and PR-AUC for Logistic regression models on validation sets with feature combination `Bio`.

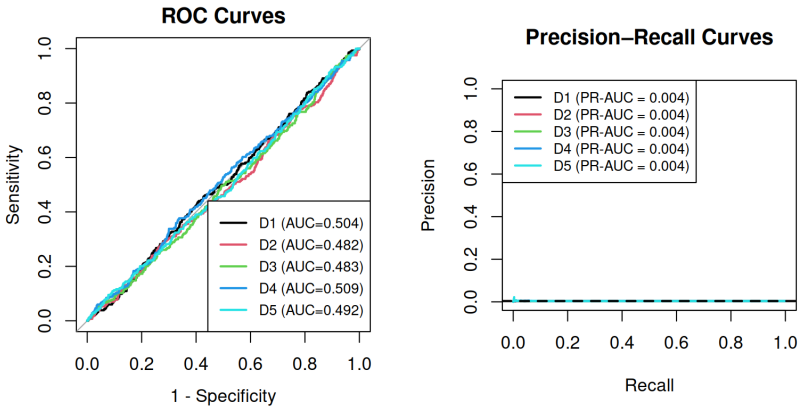


Figure 6.9: ROC-AUC and PR-AUC for XGBoost models on validation sets with feature combination `Bio`.

On the test set, the results remain broadly unchanged, as seen in Table 6.10, compared to the results for the `Full` feature combination. Logistic regression continues to produce predictions with very high sensitivity but almost no specificity, while XGBoost exhibits greater variability across imputations. In particular, XGB_1 and XGB_2 achieve relatively higher specificity at the cost of reduced sensitivity, leading to lower total costs. Nevertheless, this variation reflects threshold-driven trade-offs rather than genuine predictive signal, as precision remains negligible across all combinations.

From an economic perspective, most model combinations satisfies the cost constraint, indicating modest cost savings. These reductions are again driven primarily by thresholding rather than improved classification performance. Notably, XGBoost_5 exceeds the cost constraint, reflecting its near-complete classification of observations as positive.

Table 6.10: Results obtained from the test set, using optimal threshold, using feature combination `Bio`.

Model	Sensitivity	Specificity	Precision	F_1	F_2	MCC	$\mathcal{C}$	$\mathcal{C} < C_A - C_S$
LogReg_1	0.97	0.04	0.00	0.01	0.02	0.00	4,083,968	TRUE
LogReg_2	0.96	0.04	0.00	0.01	0.02	0.00	4,082,096	TRUE
LogReg_3	0.99	0.03	0.00	0.01	0.02	0.01	4,155,496	TRUE
LogReg_4	0.97	0.04	0.00	0.01	0.02	0.00	4,097,072	TRUE
LogReg_5	0.95	0.05	0.00	0.01	0.02	0.00	4,071,576	TRUE
XGB_1	0.82	0.19	0.00	0.01	0.02	0.00	3,468,680	TRUE
XGB_2	0.90	0.10	0.00	0.01	0.02	0.00	3,837,200	TRUE
XGB_3	0.94	0.06	0.00	0.01	0.02	0.00	3,996,440	TRUE
XGB_4	0.98	0.02	0.00	0.01	0.02	0.00	4,199,688	TRUE
XGB_5	0.99	0.01	0.00	0.01	0.02	0.00	4,241,776	FALSE

Finally, the results obtained using the cost-optimal threshold $t^* = 0.78$ in Table 6.11 exhibit the same behaviour as in the `Full` setting. The high threshold leads to a conservative classification rule, effectively eliminating positive predictions. As a result, sensitivity collapses to zero for logistic regression, while all performance metrics become undefined for XGBoost due to the absence of predicted positives. Although this drastically reduces total costs, it renders the models unusable.

Table 6.11: Results obtained from the test set, using cost-optimal threshold $t^* = 0.78$, on feature combination `Bio`.

Model	Sensitivity	Specificity	Precision	F_1	F_2	MCC	$\mathcal{C}$	$\mathcal{C} < C_A - C_S$
LogReg_1	0.00	1.00	0.00	0.00	0.00	0.00	70,352	TRUE
LogReg_2	0.00	1.00	0.00	0.00	0.00	0.00	70,240	TRUE
LogReg_3	0.00	1.00	0.00	0.00	0.00	0.00	70,184	TRUE
LogReg_4	0.00	1.00	0.00	0.00	0.00	0.00	70,296	TRUE
LogReg_5	0.00	1.00	0.00	0.00	0.00	0.00	70,632	TRUE
XGB_1	NA	1.00	NA	NA		NA	0	TRUE
XGB_2	NA	1.00	NA	NA		NA	0	TRUE
XGB_3	NA	1.00	NA	NA		NA	0	TRUE
XGB_4	NA	1.00	NA	NA		NA	0	TRUE
XGB_5	NA	1.00	NA	NA		NA	0	TRUE

Overall, restricting the feature space to biochemistry markers does not improve predictive performance relative to the `Full` specification. While minor differences in variability and cost are observed, both modelling approaches continue to exhibit near-random classification behaviour, suggesting that biochemistry markers alone do not contain sufficient signal for the prediction task.

6.3.3 Results - Haematological Assays

As in the previous feature settings, each observation yields five predicted probabilities due to the multiple imputations generated by MICE. The Wilcoxon rank-sum test is therefore applied to all pairwise combinations of imputed predictions to assess whether the imputations introduce systematic variation.

The results in Table 6.12 indicate a more heterogeneous pattern compared to both the `Full` and `Bio` settings. For logistic regression, most comparisons are statistically significant, although a notable exception is the comparison between imputations 1 and 3, suggesting similar prediction distributions in this case. For XGBoost, several comparisons are not statistically significant, indicating that the variability introduced by imputation is less consistent.

Table 6.12: Wilcoxon Rank Sum Test on feature combination `Haema`.

Group 1	Group 2	p-val. LogReg	p-val. XGB
1	2	0.00	0.00
1	3	0.82	0.25
1	4	0.00	0.71
1	5	0.00	0.00
2	3	0.00	0.00
2	4	0.00	0.00
2	5	0.01	0.00
3	4	0.00	0.13
3	5	0.00	0.00
4	5	0.00	0.00

The validation results in Table 6.13 are broadly consistent with earlier findings. Both modelling approaches continue to favour positive classifications, leading to high sensitivity

and low specificity across all imputations. As a result, precision and composite performance metrics remain close to zero, and there is no indication of meaningful predictive discrimination.

Threshold-independent metrics show a slight numerical improvement compared to previous feature sets, particularly for XGBoost, where ROC-AUC values reach up to approximately 0.53, as depicted in Figure 6.10 and Figure 6.11. However, these values remain close to random classification and do not translate into improved classification outcomes. The MILP-based threshold optimisation yields identical threshold to grid search, which, again, gives similar performance across imputations.

Table 6.13: Results obtained from the validation set, using feature combination `Haema`.

Model	MILP	Threshold	Sensitivity	Specificity	Precision	F_1	F_2	MCC	ROC-AUC	PR-AUC	$\mathcal{C}$	$\mathcal{C} < C_A - C_S$
$LogReg_1$	0.42	0.42	0.95	0.06	0.00	0.01	0.02	0.00	0.511	0.004	3,995,128	TRUE
$LogReg_2$	0.41	0.41	0.95	0.06	0.00	0.01	0.02	0.00	0.508	0.004	4,004,816	TRUE
$LogReg_3$	0.42	0.42	0.95	0.07	0.00	0.01	0.02	0.00	0.509	0.004	3,985,272	TRUE
$LogReg_4$	**0.42**	**0.42**	0.95	0.07	0.00	0.01	0.02	0.01	0.513	0.004	**3,969,872**	TRUE
$LogReg_5$	0.41	0.41	0.95	0.06	0.00	0.01	0.02	0.00	0.511	0.004	3,996,304	TRUE
XGB_1	**0.47**	**0.47**	0.95	0.07	0.00	0.01	0.02	0.01	0.515	0.004	**3,963,880**	TRUE
XGB_2	0.46	0.46	0.95	0.07	0.00	0.01	0.02	0.00	0.527	0.004	3,979,896	TRUE
XGB_3	0.46	0.46	0.95	0.06	0.00	0.01	0.02	0.00	0.526	0.004	4,004,760	TRUE
XGB_4	0.47	0.47	0.95	0.06	0.00	0.01	0.02	0.00	0.533	0.004	4,024,192	TRUE
XGB_5	0.44	0.44	0.95	0.07	0.00	0.01	0.02	0.00	0.510	0.004	3,972,280	TRUE

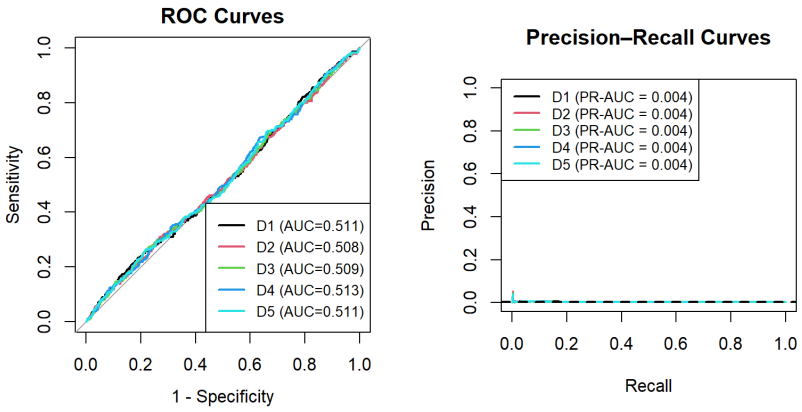


Figure 6.10: ROC-AUC and PR-AUC for Logistic regression models on validation sets with feature combination `Haema`

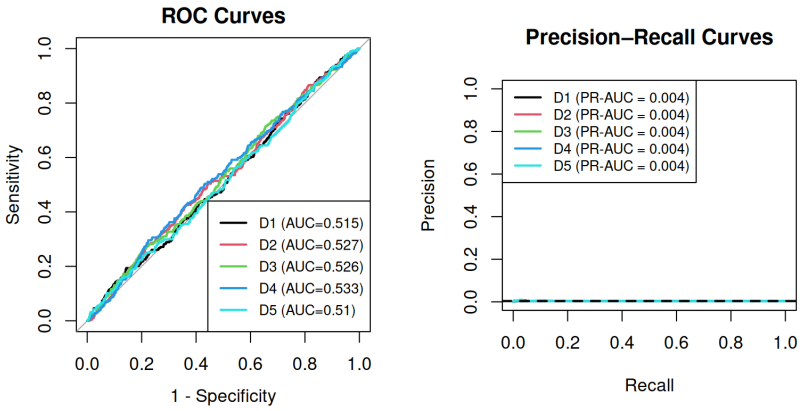


Figure 6.11: ROC-AUC and PR-AUC for XGBoost models on validation sets with feature combination `Haema`

On the test results in Table 6.14, logistic regression yields highly consistent results across imputations. XGBoost, however, exhibits greater variation, particularly in the trade-off between sensitivity and specificity. For instance, XGB_5 achieves higher specificity at the cost of reduced sensitivity, resulting in a lower total cost. Conversely, XGB_4 prioritises sensitivity, leading to increased cost and reduced specificity.

Despite these variations, the overall predictive performance remains weak. The differences between imputations and models are primarily driven by threshold-induced trade-offs rather than any underlying predictive signal, as reflected by the precision and MCC values.

From an economic perspective, all combinations satisfies the cost constraint, with XGB_5 achieving the lowest cost among all models. However, as in previous settings, these cost reductions stem from altered classification behaviour rather than improved risk stratification.

Table 6.14: Results obtained from the test set, using optimal threshold, using feature combination `Haema`.

Model	Sensitivity	Specificity	Precision	F_1	F_2	MCC	$\mathcal{C}$	$\mathcal{C} < C_A - C_S$
$LogReg_1$	0.94	0.07	0.00	0.01	0.02	0.00	3,990,840	TRUE
$LogReg_2$	0.95	0.07	0.00	0.01	0.02	0.00	3,989,928	TRUE
$LogReg_3$	0.95	0.07	0.00	0.01	0.02	0.01	3,982,728	TRUE
$LogReg_4$	0.94	0.07	0.00	0.01	0.02	0.00	3,977,176	TRUE
$LogReg_5$	0.95	0.07	0.00	0.01	0.02	0.01	3,963,352	TRUE
XGB_1	0.92	0.07	0.00	0.01	0.02	0.00	3,963,344	TRUE
XGB_2	0.84	0.16	0.00	0.01	0.02	0.00	3,589,112	TRUE
XGB_3	0.83	0.17	0.00	0.01	0.02	0.00	3,560,056	TRUE
XGB_4	0.96	0.05	0.00	0.01	0.02	0.00	4,066,512	TRUE
XGB_5	0.73	0.26	0.00	0.01	0.02	0.00	3,164,216	TRUE

Finally, the results obtained using the cost-optimal threshold $t^* = 0.78$ in Table 6.15 follow the same pattern as previously observed. The high threshold leads to no positive predictions for XGBoost and zero sensitivity for logistic regression. While this drastically reduces total costs, it eliminates any practical utility of the models.

Table 6.15: Results obtained from the test set, using cost-optimal threshold $t^* = 0.78$, on feature combination `Haema`.

Model	Sensitivity	Specificity	Precision	F_1	F_2	MCC	$\mathcal{C}$	$\mathcal{C} < C_A - C_S$
LogReg_1	0.00	1.00	0.01	0.01	0.00	0.00	75,320	TRUE
LogReg_2	0.00	1.00	0.01	0.01	0.00	0.00	75,712	TRUE
LogReg_3	0.00	1.00	0.01	0.01	0.00	0.00	75,656	TRUE
LogReg_4	0.00	1.00	0.01	0.00	0.00	0.00	76,944	TRUE
LogReg_5	0.00	1.00	0.01	0.01	0.00	0.00	75,040	TRUE
XGB_1	NA	1.00	NA	NA	NA	NA	0	TRUE
XGB_2	NA	1.00	NA	NA	NA	NA	0	TRUE
XGB_3	NA	1.00	NA	NA	NA	NA	0	TRUE
XGB_4	NA	1.00	NA	NA	NA	NA	0	TRUE
XGB_5	NA	1.00	NA	NA	NA	NA	0	TRUE

Overall, the `Haema` feature set exhibits slightly greater variability across imputations and models compared to the `Bio` setting, particularly for XGBoost. Nevertheless, this does not translate into improved predictive performance, and both modelling approaches continue to perform close to random classification.

7 | Discussion

This chapter discusses the methodological choices and limitations of the project, including the handling of missing data, class imbalance, and cost-sensitive decision-making. Furthermore, alternative modelling strategies are considered and the implications of uncertainty in both data and cost estimates are examined. Finally, the potential clinical and socioeconomic impact of the proposed screening framework, as well as limitations related to the representativeness of the data, are discussed.

In Algorithm 4, a specific XGBoost imputation method for handling missing data was introduced. However, the approach utilized in this project was Multiple Imputations by Chained Equations (MICE), as it provides results that are comparable across a wider range of models, whereas Algorithm 4 is specifically designed for XGBoost. Using MICE therefore ensured a more model-agnostic preprocessing approach and improved the comparability of results throughout the project. Furthermore, [12] examined the use of different imputation methods in the prediction of major adverse cardiovascular events. The study found that MICE achieved the highest imputation accuracy among the methods considered. Consequently, the findings suggest that MICE is an appropriate choice for the imputation procedure used in this project, as it has been shown to provide accurate and reliable imputations in similar clinical prediction settings.

While MICE was selected based on methodological and data-driven arguments, its validity ultimately depends on the assumption that data are missing at random. Although this assumption was motivated by the structure of the UK Biobank data, where missingness largely arises from measurement and laboratory processes, it cannot be formally verified. If missingness in certain biomarkers is related to unobserved patient characteristics, the imputed values may be biased. Furthermore, the use of predictive mean matching ensures realistic imputations by preserving the empirical distribution of observed values, but its performance depends on the predictive quality of the imputation models.

The approach for minimizing cost in this project is based on first estimating class probabilities and subsequently identifying the optimal decision threshold by minimizing the empirical cost function. This yields a two-stage framework separating probability estimation from decision-making. A limitation of this approach is that the optimal threshold is sensitive to the specified cost parameters. In practice, these costs are often uncertain or context-dependent, which may lead to suboptimal decisions if misclassification costs are misspecified. An alternative approach is to incorporate asymmetric misclassification costs directly into the training objective through a cost-weighted loss function

$$L(y, F(x)) = -\sum_{i=1}^{n} \Big[C_{FN}\, y_i \log(\widehat{p}_i) + C_{FP}\,(1 - y_i) \log(1 - \widehat{p}_i) \Big].$$

This integrates cost considerations directly into the estimation procedure and may be preferable in settings where misclassification costs are well-defined and fixed. For XGBoost, cost sensitivity could be introduced through observation-level weights based on the true class, which scale the gradient and Hessian of the logistic loss

$$\begin{aligned} w_i &= \begin{cases} C_{FN}, & \text{if } y_i = 1, \\ C_{FP}, & \text{if } y_i = 0, \end{cases} \\ g_i &= w_i(p_i - y_i), \\ h_i &= w_i p_i (1 - p_i). \end{aligned}$$

However, incorporating asymmetric costs directly into the loss function alters the optimization objective and may distort probability calibration, as the model is no longer trained to approximate the true posterior probabilities. In contrast, the two-stage formulation preserves the probabilistic interpretation of the estimated class probabilities, while allowing the decision threshold to be adjusted flexibly to different cost structures without retraining the classifier.

One potential advantage of implementing the pre-screening strategy proposed in this project is the earlier detection of e.g. haemochromatosis. In many cases, treatment consists of regular therapeutic phlebotomy, where blood is removed to reduce iron accumulation. If the collected blood meets regulatory safety standards, it may potentially be used for blood donation purposes, thereby increasing healthcare resource utilization. Given the current shortage of blood donors in Denmark [5], such an approach may contribute to reducing the need for recruitment of additional donors while simultaneously supporting the national blood supply. Consequently, the proposed screening strategy may not only provide clinical benefits through earlier diagnosis, but also generate potential socioeconomic advantages within the healthcare system.

In Denmark, all individuals between the ages of 50 and 74 are offered screening for bowel cancer. The objective of the screening program is to identify precancerous lesions or cancer at an early stage, thereby enabling timely treatment and reducing disease progression [19]. As the bowel cancer screening program targets a demographic group comparable to the population considered in this report, the associated screening expenditures could be incorporated as an additional budget constraint within the cost function, since the true screening cost per person is established. Furthermore, these expenditures could serve as a benchmark for assessing whether the proposed screening strategy is economically reasonable and practically implementable within a healthcare setting.

The data considered in this project exhibit a substantial class imbalance, where the minority class constitutes approximately 0.6% of all observations. To address this issue, bagging combined with iterative random undersampling was employed. By repeatedly generating undersampled training sets, this approach mitigates the loss of information associated with a single random undersampling step, while remaining computationally feasible given the large size of the dataset. An alternative approach for handling class imbalance is Near-Miss undersampling. This method retains observations from the majority class that are closest to the minority class observations [9]. The underlying rationale is that observations located near the decision boundary are the most informative for classification, potentially improving the model's ability to distinguish between classes. Nevertheless, bagging with iterative random undersampling was preferred in this project, as it preserves a larger degree of variability from the majority class across iterations and reduces the risk of overemphasising noisy or unrepresentative boundary observations.

During the search for the optimal threshold satisfying the imposed constraints, the misclassification costs are treated as fixed. The costs used in Chapter 3 are derived from external reports and should be interpreted as estimates of the true underlying costs. Consequently, the actual costs remain uncertain until the model is deployed in a real-world setting. An alternative approach would therefore be to model the costs as uncertain during the training and validation phases, and only assume them to be known at deployment, when no further adjustments to the model are possible.

As discussed in Section 6.1, the data used in this study are affected by a healthy volunteer bias, which introduces a form of selection bias. This may limit the generalizability and transferability of the developed models, as the study cohort differs from the broader target population. Consequently, model performance observed within the cohort may not

accurately reflect performance in the general population.

8 | Conclusion

The thesis aimed to develop and evaluate a machine learning model for risk stratification of C282Y homozygosity using demographic variables and available blood parameters, and to assess its potential as a cost-effective pre-screening tool in a clinical setting.

Due to the loss of access to the UK Biobank dataset, it was not possible to train, validate, or evaluate the proposed models. Consequently, no conclusions can be drawn regarding predictive performance, generalizability, or clinical utility. This also implies that any assessment of cost-effectiveness remains purely theoretical, as such analysis depends on reliable estimates of model sensitivity and specificity.

Despite this limitation, a complete modelling framework was developed, including data preprocessing strategies, model specifications, and an outline for both predictive evaluation and economic assessment. The proposed methodology remains applicable.

A key direction for future work is to re-implement and run the full modelling pipeline once access to the UK Biobank dataset, or a comparable dataset, becomes available. This would allow for a complete empirical evaluation of the proposed approach, including model training, validation, and assessment of predictive performance. Only with such results would it be possible to properly evaluate the model's clinical applicability and its potential role as a cost-effective pre-screening tool.

Appendices

A | Linear programming

A.1 Simplex Example

In this appendix, the Simplex procedure applied to the linear programming relaxation in Example 4.1 is introduced. Relaxing the integrality constraints yields a standard linear program whose optimal solution provides both an upper bound on the objective value and insight into the geometry of the feasible region. To illustrate this process explicitly, consider the relaxed problem

$$\begin{aligned} \text{minimize } & -17x_1 - 12x_2 \\ \text{subject to } & 10x_1 + 7x_2 \leq 40, \\ & x_1 + x_2 \leq 5, \\ & x_1, x_2 \geq 0. \end{aligned}$$

The canonical form is converted to standard form, by adding slack variables and substituting the inequalities with equalities. Hence, the input becomes

$$\begin{aligned} &\text{Standard form :} \\ &\quad \begin{aligned} \text{minimize } & -17x_1 - 12x_2 \\ \text{subject to } & 10x_1 + 7x_2 + s_1 = 40, \\ & x_1 + x_2 + s_2 = 5, \\ & x_1, x_2, s_1, s_2 \geq 0, \end{aligned} \end{aligned}$$

where s_1 and s_2 are slack variables and the initial basic feasible solution is $x_I = (0,0)^\top$.

Step 1: x_B is calculated to ensure the feasibility in the relaxation problem. The initial basis is $B = \{s_1, s_2\}$ with complimentary set $N = \{x_1, x_2\}$, which gives

$$\begin{aligned} A_B = A_B^{-1} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\ b &= (40,5)^\top, \\ x_B = A_B^{-1} b &= \begin{bmatrix} 40 \\ 5 \end{bmatrix}. \end{aligned}$$

Step 2: Firstly, $y^\top$ is calculated to check if the initial basis satisfies the constraints in the relaxation problem

$$\begin{aligned} c_B^\top &= \begin{bmatrix} 0 & 0 \end{bmatrix}, \\ y^\top = c_B^\top A_B^{-1} &= \begin{bmatrix} 0 & 0 \end{bmatrix}. \end{aligned}$$

Secondly, the reduced cost is computed

$$\begin{aligned} c = \begin{bmatrix} -17 \\ -12 \\ 0 \\ 0 \end{bmatrix} \quad A^\top &= \begin{bmatrix} 10 & 1 \\ 7 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \\ r = c - A^\top y = \begin{bmatrix} -17 \\ -12 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 10 & 1 \\ 7 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} &= \begin{bmatrix} -17 \\ -12 \\ 0 \\ 0 \end{bmatrix}, \end{aligned}$$

where it is obvious that $e = 1$, since its the one providing biggest change in the objective.

Step 3: Optimality is checked by

$$r_1 = -17 < 0,$$

and since the current basic feasible solution is not optimal, the algorithm proceeds to the next step.

Step 4: Next, the search direction for $e = 1$ is calculated using

$$A_{\star 1} = \begin{bmatrix} 10 \\ 1 \end{bmatrix},$$

such that

$$d = A_B^{-1} A_{\star 1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 1 \end{bmatrix}.$$

Step 5: Check if the unbounded condition is satisfied;

$$d_1 = 10 > 0 \quad d_2 = 1 > 0.$$

Since both are greater than 0, the algorithm proceeds to the next step.

Step 6: Use the Minimum Ratio Test, to determine which basic variable should be left out;

$$\theta = \min \left\{ \frac{40}{10}, \frac{5}{1} \right\} = \min\{4, 5\} = 4,$$

which yields the outgoing variable $o = 1$.

Step 7: Update the basis and the basic feasible solution by excluding the outgoing variable from the basis, $B_1 = \{s_1\}$, and replace it with the entering variable from the complimentary set, $N_1 = \{x_1\}$, such that

$$B = (\{s_1, s_2\} \setminus \{s_1\}) \cup \{x_1\} = \{x_1, s_2\}.$$

Then the basic variables are updated

$$\begin{aligned} x_B &= \begin{bmatrix} 40 \\ 5 \end{bmatrix} - 4 \begin{bmatrix} 10 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ (x_B)_1 &= 4. \end{aligned}$$

The updated values for the initial basis and the feasible solution becomes

$$A_B = \begin{bmatrix} 10 & 0 \\ 1 & 1 \end{bmatrix}, \quad x_B = \begin{bmatrix} 4 \\ 1 \end{bmatrix}.$$

Return to *step 2*: Starting the second iteration with the new basic variable, it is checked that the constraints are satisfied;

$$\begin{aligned} c_B^\top &= \begin{bmatrix} -17 & 0 \end{bmatrix}, \\ A_B^{-1} &= \begin{bmatrix} \frac{1}{10} & 0 \\ -\frac{1}{10} & 1 \end{bmatrix}, \\ y^\top &= c_B^\top A_B^{-1} = \begin{bmatrix} -17 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{10} & 0 \\ -\frac{1}{10} & 1 \end{bmatrix} = \begin{bmatrix} -\frac{17}{10} & 0 \end{bmatrix}. \end{aligned}$$

The new reduced cost is computed as

$$r = c - A^\top y = \begin{bmatrix} -17 \\ -12 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 10 & 1 \\ 7 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -\frac{17}{10} \\ 0 \end{bmatrix} = \begin{bmatrix} -17 \\ -12 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -17 \\ -\frac{119}{10} \\ -\frac{17}{10} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{10} \\ \frac{17}{10} \\ 0 \end{bmatrix},$$

where $e = 2$.

Step 3: Optimality is checked by

$$-\frac{1}{10} < 0,$$

and since the feasible solution is not optimal, the algorithm continues.

Step 4: The search direction for $e = 2$ is calculated using

$$A_{\star 2} = \begin{bmatrix} 7 \\ 1 \end{bmatrix},$$

such that

$$d = A_B^{-1} A_{\star 1} = \begin{bmatrix} \frac{1}{10} & 0 \\ -\frac{1}{10} & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{7}{10} \\ \frac{3}{10} \end{bmatrix}.$$

Step 5 Check if the unbounded condition is satisfied;

$$d_1 = \frac{7}{10} > 0,$$
$$d_2 = \frac{3}{10} > 0.$$

Since both entries are positive, the objective is still bounded.
Step 6: Use the Minimum Ratio Test, to determine which basic variable should be left out;

$$\theta = \min\left\{\frac{40}{7}, \frac{10}{3}\right\} = \frac{10}{3}.$$

which yields the outgoing variable $o = 2$.

Step 7: Update the basis and the basic feasible solution by excluding the outgoing variable from the basis, $B_2 = \{s_2\}$, and replace it with the entering variable from the complimentary set, $N_2 = \{x_2\}$, such that

$$B = (\{x_1, s_2\} \setminus \{s_2\}) \cup x_2 = \{x_1, x_2\}.$$

Then the basic variables are updated

$$x_B = \begin{bmatrix} 4 \\ 1 \end{bmatrix} - \frac{10}{3} \begin{bmatrix} \frac{7}{10} \\ \frac{3}{10} \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} - \begin{bmatrix} \frac{7}{3} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{5}{3} \\ 0 \end{bmatrix},$$
$$(x_B)_2 = \frac{10}{3}.$$

The updated values for the basis and the feasible solution becomes

$$A_B = \begin{bmatrix} 10 & 7 \\ 1 & 1 \end{bmatrix}, \quad x_B = \begin{bmatrix} \frac{5}{3} \\ \frac{10}{3} \end{bmatrix}.$$

Return to *Step 2*: Starting the third iteration with the new basic variable, it is checked that the constraints are satisfied;

$$\begin{aligned} c_B^\top &= \begin{bmatrix} -17 & -12 \end{bmatrix}, \\ A_B^{-1} &= \begin{bmatrix} \frac{1}{3} & -\frac{7}{2} \\ -\frac{1}{3} & \frac{10}{3} \end{bmatrix}, \\ y^\top &= \begin{bmatrix} -17 & -12 \end{bmatrix} \begin{bmatrix} \frac{1}{3} & -\frac{7}{3} \\ -\frac{1}{3} & \frac{10}{3} \end{bmatrix} = \begin{bmatrix} -\frac{5}{3} & -\frac{1}{3} \end{bmatrix}. \end{aligned}$$

The new reduced cost is computed as

$$r = c - A^\top y = \begin{bmatrix} -17 \\ -12 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 10 & 1 \\ 7 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -\frac{5}{3} \\ -\frac{1}{3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{5}{3} \\ \frac{1}{3} \end{bmatrix},$$

where $e = \{1, 2\}$.

Step 3: Optimality is checked by

$$r_1 = 0 \geq 0. \quad r_2 = 0 \geq 0,$$

and since the condition is satisfied, the optimal solution has been reached.

This means, a change in the variables x_1, x_2 would not provide further reduction in the objective, therefore the current basic variables, $x_B = \begin{bmatrix} \frac{5}{3} & \frac{10}{3} \end{bmatrix}$, must be the optimal solution for the minimization problem, with the optimal value

$$-17\left(\frac{5}{3}\right) - 12\left(\frac{10}{3}\right) = -\frac{205}{3} = -68.33.$$

B | Application

B.1 Modelling

Table B.1: Median values of numeric features by sex and their absolute difference.

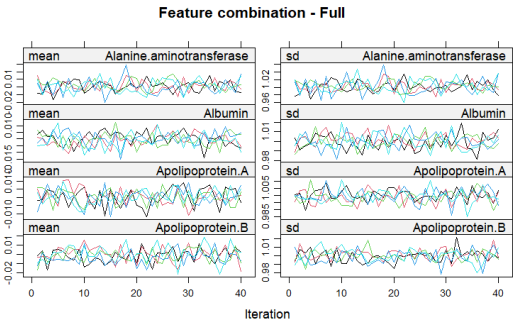
Feature	Female	Male	Abs Diff	Feature	Female	Male	Abs Diff
Oestradiol	396.40	204.10	192.30	Platelet distribution width	16.37	16.50	0.13
Urate	264.10	349.90	85.80	LDL direct	3.57	3.46	0.11
Platelet count	261.00	233.70	27.30	Lymphocyte count	1.91	1.80	0.11
SHBG	56.39	36.93	19.46	Monocyte count	0.40	0.50	0.10
Creatinine	63.10	80.00	16.90	C-reactive protein	1.38	1.28	0.10
Gamma glutamyltransferase	21.40	33.10	11.70	Glycated haemoglobin (HbA1c)	35.20	35.30	0.10
Testosterone	1.01	11.62	10.60	Mean platelet (thrombocyte) volume	9.25	9.17	0.08
Alanine aminotransferase	17.49	23.88	6.39	Phosphate	1.20	1.12	0.08
Haematocrit percentage	39.27	43.34	4.07	Neutrophill count	4.00	4.07	0.07
Aspartate aminotransferase	23.00	26.20	3.20	White blood cell (leukocyte) count	6.63	6.70	0.07
Alkaline phosphatase	81.70	79.00	2.70	Cystatin C	0.86	0.92	0.06
Lipoprotein A	22.20	19.78	2.42	Reticulocyte percentage	1.23	1.29	0.06
Lymphocyte percentage	29.50	27.50	2.00	Glucose	4.91	4.95	0.04
Total bilirubin	7.26	9.13	1.87	Eosinophill count	0.12	0.16	0.04
Haemoglobin concentration	13.50	15.00	1.50	Red blood cell (erythrocyte) distribution width	13.36	13.32	0.04
IGF-1	20.74	21.78	1.04	Basophill percentage	0.44	0.41	0.03
Monocyte percentage	6.40	7.42	1.02	Platelet crit	0.24	0.21	0.03
Age at recruitment	57.00	58.00	1.00	High light scatter reticulocyte percentage	0.35	0.37	0.02
Mean reticulocyte volume	105.60	106.23	0.63	Calcium	2.38	2.37	0.01
Neutrophill percentage	60.90	61.50	0.60	Reticulocyte count	0.05	0.06	0.01
Albumin	44.92	45.51	0.59	Apolipoprotein B	1.02	1.01	0.01
Red blood cell (erythrocyte) count	4.32	4.75	0.43	High light scatter reticulocyte count	0.01	0.02	0.01
Cholesterol	5.82	5.45	0.36	Basophill count	0.02	0.02	0.00
Urea	5.10	5.46	0.36	Immature reticulocyte fraction	0.29	0.29	0.00
Triglycerides	1.33	1.70	0.36	Nucleated red blood cell count	0.00	0.00	0.00
Direct bilirubin	1.46	1.79	0.33	Nucleated red blood cell percentage	0.00	0.00	0.00
Mean sphered cell volume	82.82	82.49	0.33				

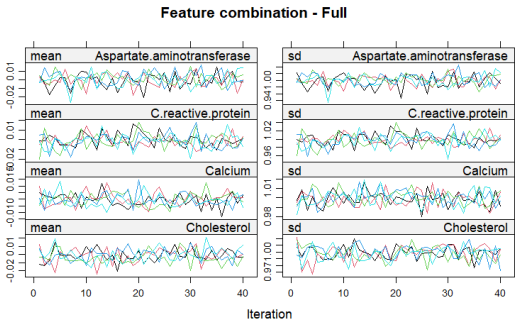
Table B.2: Distribtution for percentage of missing values per feature after removing observations with missing entries in C282Y.

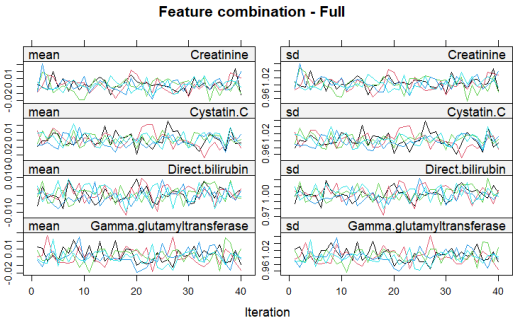
Feature	% NA	Feature	% NA
Rheumatoid factor	91.60	Calcium	12.66
Oestradiol	84.43	Albumin	12.63
Lipoprotein A	23.68	Vitamin D	8.91
Direct bilirubin	19.00	IGF-1	5.12
Testosterone	13.60	Apolipoprotein B	5.09
SHBG	13.47	Total bilirubin	5.01
Apolipoprotein A	13.15	Aspartate aminotransferase	4.96
Phosphate	12.80	Glycated haemoglobin (HbA1c)	4.85
Glucose	12.73	C-reactive protein	4.81
Total protein	12.73	LDL direct	4.78
HDL cholesterol	12.67	Urate	4.72
Creatinine	4.65	Triglycerides	4.68
Gamma glutamyltransferase	4.65	Urea	4.67
Alanine aminotransferase	4.64	Alkaline phosphatase	4.60
Cystatin C	4.61	Immature reticulocyte fraction	4.60
Cholesterol	4.60	Mean reticulocyte volume	4.60
High light scatter reticulocyte count	4.60	Reticulocyte count	4.60
Reticulocyte percentage	4.60	High light scatter reticulocyte percentage	4.60
Mean sphered cell volume	4.60	Nucleated red blood cell percentage	3.14
Nucleated red blood cell count	3.14	Basophill count	3.13
Eosinophill count	3.13	Lymphocyte count	3.13
Monocyte count	3.13	Neutrophill count	3.13
Basophill percentage	3.13	Eosinophill percentage	3.13
Lymphocyte percentage	3.13	Monocyte percentage	3.13
Neutrophill percentage	3.13	Mean platelet (thrombocyte) volume	2.95
Platelet distribution width	2.95	Mean corpuscular haemoglobin concentration	2.95
Platelet crit	2.95	White blood cell (leukocyte) count	2.95
Mean corpuscular haemoglobin	2.95	Platelet count	2.95
Mean corpuscular volume	2.95	Red blood cell (erythrocyte) distribution width	2.95
Haematocrit percentage	2.95	Haemoglobin concentration	2.95
Red blood cell (erythrocyte) count	2.95	Ethnic Background	0.11
Sex	0.00	Age at recruitment	0.00
H63D	0.00	S65C	0.00
C282Y	0.00		

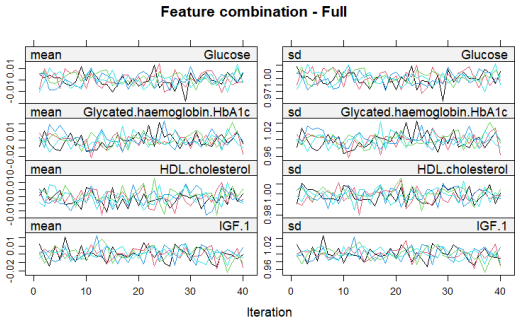
C | Mice imputations

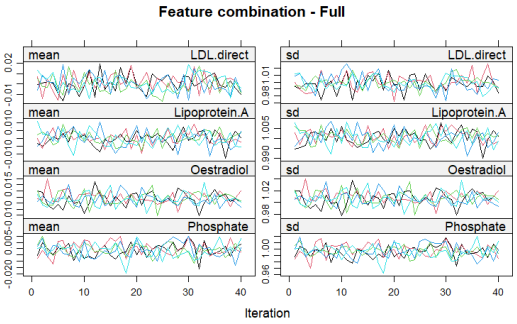
C.1 Feature combination - Full

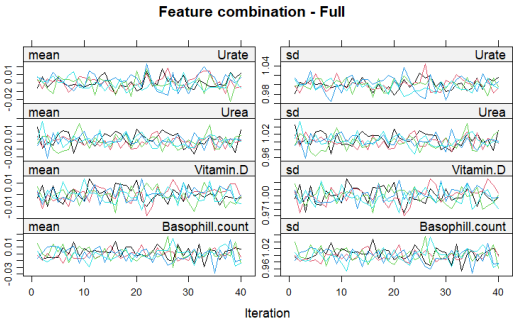


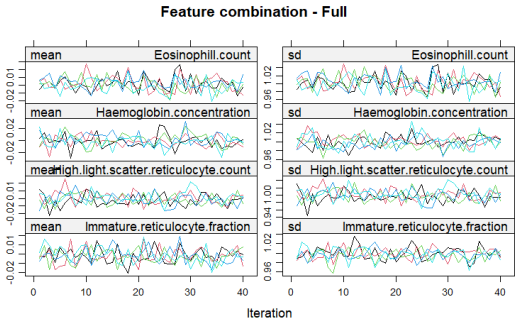


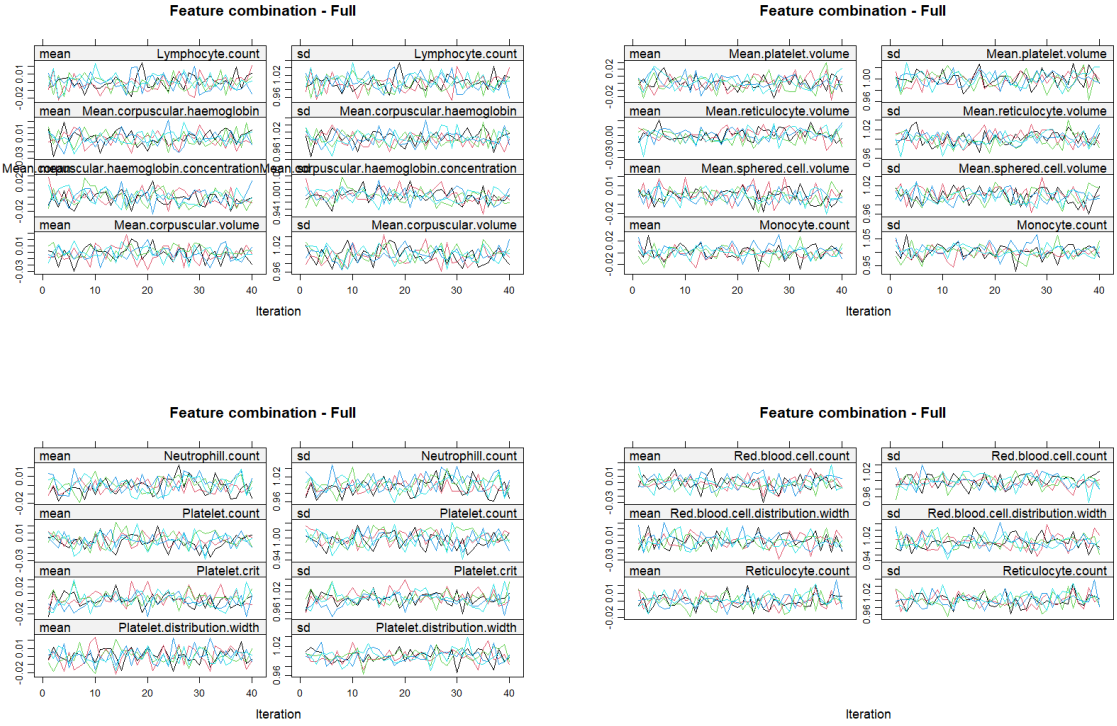




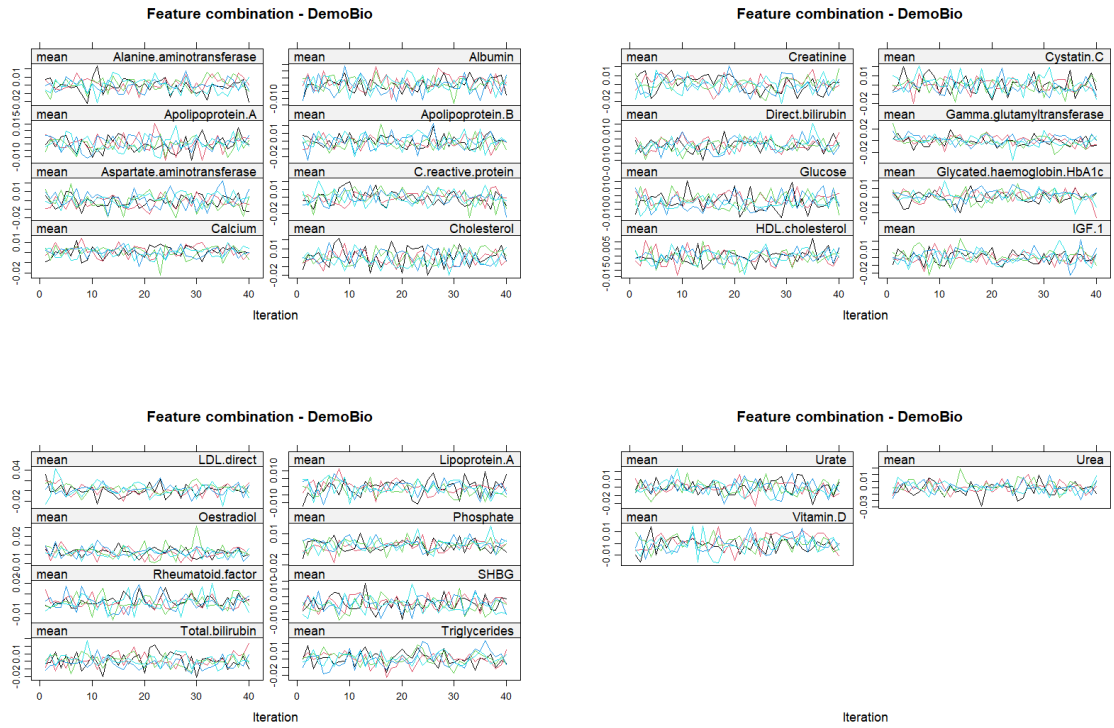




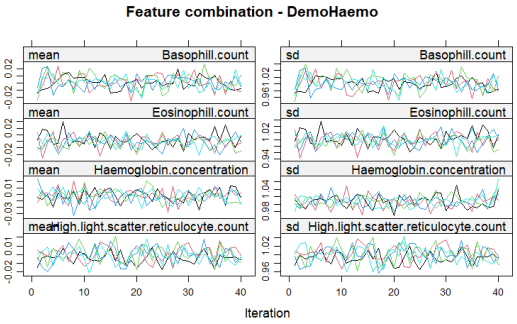


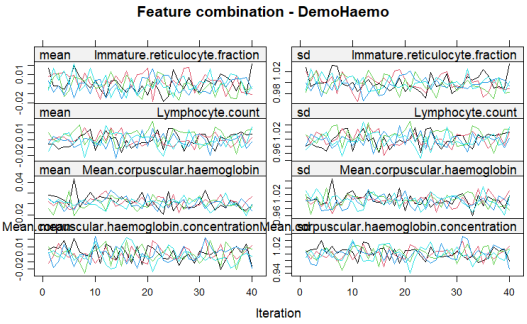


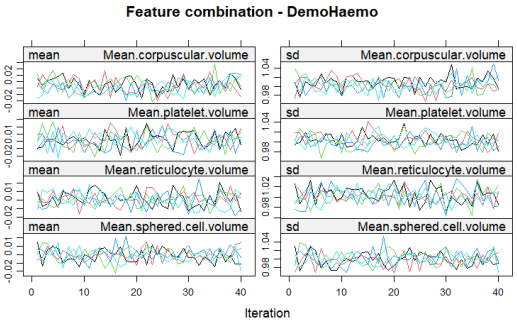
C.2 Feature combination - Biochemistry Markers

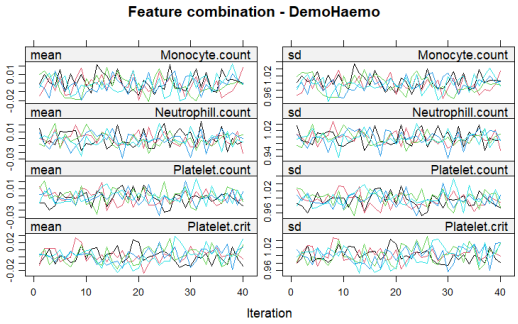


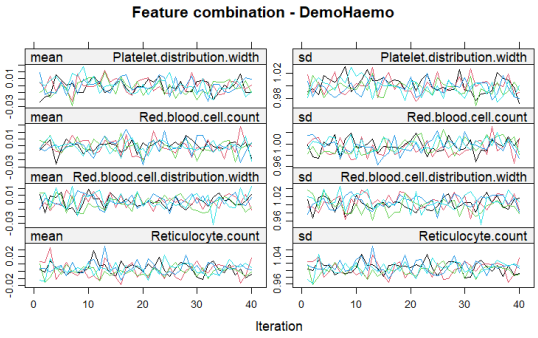
C.3 Feature combination - Haematological Assays











References

[1] Allen, K. J., Gurrin, L. C., Constantine, C. C., Osborne, N. J., Delatycki, M. B., Nicoll, A. J., McLaren, C. E., et al. (2008). Iron-overload-related disease in hfe hereditary hemochromatosis. *New England Journal of Medicine*, 358(3):221–230.

[2] Allen, N. E., Arnold, M., Parish, S., et al. (2021). Approaches to minimising the epidemiological impact of sources of systematic and random variation that may affect biochemistry assay data in uk biobank. *Wellcome Open Research*, 5:222. Version 2; peer review: 2 approved.

[3] Andrew Lim, Mark Speechley, P. C. A. (2025). Predicting c282y homozygote genotype for hemochromatosis using serum ferritin and transferrin saturation values from 44,809 participants of the heirs study. *Canadian Journal Gastroenterology and hepatology*, 6(6):e70170.

[4] Austin, P. C., White, I. R., Lee, D. S., and van Buuren, S. (2021). Missing data in clinical research: A tutorial on multiple imputation. *Canadian Journal of Cardiology*, 37(9):1322–1331.

[5] Bloddonorerne Danmark (2025). 5 grunde til at blive bloddonor. Accessed: 2026-05-07.

[6] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

[7] Christensen, F., Kılıç, D. K., Fuglkjær, A. D., Petersen, J., El-Galaly, T. C., Glenthøj, A., Helby, J., and Nielsen, I. E. (2025a). Machine learning-based detection of hbs and hbc carriers in the uk general population. *EJHaem*, 6(6):e70170.

[8] Christensen, F., Kılıç, D. K., Nielsen, I. E., El-Galaly, T. C., Glenthøj, A., Helby, J., Frederiksen, H., Möller, S., and Fuglkjær, A. D. (2025b). Classification of alpha-thalassemia data using machine learning models. *Computer Methods and Programs in Biomedicine*, 260:108581.

[9] Comet (2021). Resampling to properly handle imbalanced datasets in machine learning. `https://www.comet.com/site/blog/resampling-to-properly-handle-imbalanced-datasets-in-machine-learning/`. Accessed: 2026-05-21.

[10] Dantzig, G. B. (1990). *Origins of the simplex method*, pages 141–151. Association for Computing Machinery.

[11] Dedene, S. V. G. (2005). Cost-sensitive learning and decision making revisited. *EUROPEAN JOURNAL OF OPERATION RESEARCH*, 166(1):212–220.

[12] Faquih, T., van Smeden, M., Luo, J., Le Cessie, S., Kastenmüller, G., Krumsiek, J., Noordam, R., van Heemst, D., Rosendaal, F. R., van Hylckama Vlieg, A., Willems van Dijk, K., and Mook-Kanamori, D. O. (2020). A workflow for missing values imputation of untargeted metabolomics data. *Metabolites*, 10(12):486.

[13] Ferrer, L. (2025). No need for ad-hoc substitutes: The expected cost is a principled all-purpose classification metric. *Transactions on Machine Learning Research*, 1(1):1–27.

[14] Friedman, J. H. (1999). Greedy function approximation: A gradient boosting machine. Technical report ucb/stat/trebst, Stanford University, Department of Statistics.

[15] Fry, A., Littlejohns, T. J., Sudlow, C., Doherty, N., Adamska, L., Sprosen, T., Collins, R., and Allen, N. E. (2017). Comparison of sociodemographic and health-related characteristics of uk biobank participants with those of the general population. *American Journal of Epidemiology*, 186(9):1026–1034.

[16] Geek for geeks (2025). Bayes theorem in machine learning. Website: Geekforgeeks. Last updated 31 Dec, 2025.

[17] Kerr, S. M., Fletcher, B. S., Tzoneva, G., Shuldiner, A. R., Gilbert, E., and Wilson, J. F. (2025). The landscape of hereditary haemochromatosis risk and diagnosis across the british isles and ireland. *PubMed Central*. PMCID: PMC12868708.

[18] Komisarenko, V. and Kull, M. (2025). Cost-sensitive classification with cost uncertainty: do we need surrogate losses? *Machine Learning*.

[19] Kræftens Bekæmpelse (2025). Screening for tarmkræft. Accessed: 2026-05-07.

[20] Lew, J., Sanghavi, M., Ayers, C. R., McGuire, D. K., Omland, T., Atzler, D., Gore, M. O., Neeland, I., Berry, J. D., Khera, A., Rohatgi, A., and de Lemos, J. A. (2017). Sex-based differences in cardiometabolic biomarkers. *Circulation*, 135(6):544–555.

[21] LLP, F. C. and UK, H. (2025). The economic impact of screening for genetic haemochromatosis in northern ireland. Technical report, Haemochromatosis UK. 104-page report commissioned by Haemochromatosis UK; accessed 2026-02-06.

[22] Lucas, M. R., Delgado, J., Beaumont, R. N., Hawkes, G., Wood, A. R., Wright, C. F., Shearman, J. D., Atkins, J. L., and Pilling, L. C. (2026). Genetic and lifestyle modifiers of haemochromatosis-related clinical outcomes in hfe c282y homozygotes. *JHEP Reports*, 8(4):101781.

[23] Martins Conde, P., T., S., and Nguyen, T. (2020). An efficient machine learning-based approach for screening individuals at risk of hereditary haemochromatosis. *Sci Rep 10*, 387:e079147.

[24] Milman, N., Koefoed, P., Pedersen, P., Nielsen, F. C., and Eiberg, H. (2003). Frequency of the HFE C282Y and H63D mutations in danish patients with clinical haemochromatosis initially diagnosed by phenotypic methods. *European Journal of Haematology*, 71(6):403–407.

[25] National Institute of Justice (2023). Hardy-weinberg principle.

[26] Office for National Statistics (2021). Dataset selection. Accessed: 2026-03-31.

[27] Pilling, L. C., Tamosauskaitė, J., Jones, G., et al. (2019). Common conditions associated with hereditary haemochromatosis genetic variants: Cohort study in UK biobank. *BMJ*, 364:k5222.

[28] Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

[29] Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association*, 91(434):473–489.

[30] Statistics Denmark (n.d.). Hosted server - analysis platform. Accessed: 2026-04-29.

[31] Sundhed.dk (2024). Hæmokromatose. Patienthåndbogen på sundhed.dk. Accessed: 2026-02-06.

[32] Sundhed.dk - Lægehåndbogen (2024). Hæmokromatose. Sundhed.dk: Sundhedsfaglig Lægehåndbogen. Updated 23 February 2024; Accessed 6 February 2026.

[33] Tamosauskaitė, J., Atkins, J. L., Pilling, L. C., et al. (2019). Hereditary hemochromatosis associations with frailty, sarcopenia and chronic pain: Evidence from 200,975 older UK biobank participants. *The Journals of Gerontology: Series A*, 74(3):337–342.

[34] UK Biobank (2025). Whole exome sequencing data. Accessed: 2026-03-30.

[35] UK Biobank (2026a). Our participants. Accessed: 2026-03-30.

[36] UK Biobank (2026b). Uk biobank data showcase search. `https://biobank.ndph.ox.ac.uk/showcase/search.cgi`. Accessed: 2026-03-31.

[37] UK National Screening Committee (2021). Haemochromatosis screening recommendation. GOV.UK: Health Screening Recommendations. Screening not recommended for haemochromatosis; accessed 2026-02-06.

[38] van Buuren, S. and Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67.

[39] Vanderbei, R. J. (2020). *Linear Programming; Foundations and Extensions*. Springer, 5th ed. edition.

[40] Wolsey, L. A. and Nemhauser, G. L. (2014). *Integer and Combinatorial Optimization*. Wiley, 1st ed. edition.