Integrating External Credit Scores in Internal Credit Risk Models

Fredrik Karlsson, Simon Malmberg

Master's thesis
2026:E60

Lund Institute of Technology
Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

Centrum Scientiarum Mathematicarum

Master's Theses in Mathematical Sciences 2026:E60
ISSN 1404-6342

LUTFMS-3557-2026

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lu.se/

Abstract

From both a legal and business perspective, it is important for credit institutes to accurately assess the risk of default for potential borrowers. This is traditionally done using statistical methods, such as logistic regression or decision trees, utilising sociodemographic and credit history information. This thesis investigates whether a bureau score, provided by a credit bureau, should be incorporated as an internal variable within the credit risk model, referred to as a model-in-model approach, or used as a standalone model and combined with an internal model through ensemble techniques. The two approaches are evaluated using logistic regression and XGBoost on a dataset of real loan applications from a specific market. The logistic regression models utilise Weight of Evidence transformed variables, and the XGBoost models use non-transformed features. Model performance is evaluated using the discriminatory power measures Gini and KS on 100 bootstrap samples. Both Gini and KS are consistently higher for the model-in-model approach across the bootstrap samples. However, the evidence is less conclusive as to whether this result generalises to future datasets.

Keywords: Credit Risk, Bureau Score, Scorecard, Default, Ensemble Model, Model-in-Model, Logistic Regression, XGBoost, Weight of Evidence, Feature Engineering, Feature Selection, Gini, KS

Acknowledgments

In the first place, we want to thank our supervisors, Jerry & Gustav[1], at the bank. Jerry for sharing his extensive knowledge of consumer credit risk and for his involvement in helping us find a thesis topic. Gustav for his assistance in academic writing; without him, the report would not be nearly as refined. Of course, we also want to thank all the other members of the credit risk team for their willingness to help and for our interesting coffee chats.

We also want to thank our academic supervisor Magnus Wiktorsson for his guidance, and for all our intriguing conversations. Ranging from the answer to the universe, to the name of King Carl XVI Gustafs grandchildren.

Lund, May 2026
Fredrik Karlsson & Simon Malmberg

[1]These are not their real names...

Contents

Dictionary

AUC	–	Area Under Curve
Bad	–	Customer who has defaulted within 12 months from approval
BsM	–	Bureau Score Model
CDF	–	Cumulative Distribution Function
EM	–	Ensemble Model
Good	–	Customer who has not defaulted within 12 months from approval
HP	–	Hyperparameter
IM	–	Internal Model
IV	–	Information Value
KGB	–	Known-good-bad
KS	–	Kolmogorov-Smirnov
LR	–	Logistic Regression
MiM	–	Model-in-Model
MIV	–	Marginal Information Value
OOS	–	Out of Sample
OR	–	Odds Ratio
Override	–	Manually approved customer
PD	–	Probability of Default
ROC	–	Receiver Operating Characteristic
WoE	–	Weight of Evidence

Chapter 1

Background

1.1 Introduction

Banking has long been a cornerstone of society. Many view medieval Florence as the cradle of modern banking. The Medici family rose to become one of the most powerful dynasties to ever walk this earth by meeting a timeless and universal demand among companies and individuals: credit. Regardless of whether you are a medieval farmer or a Fortune 500 company, your operations are facilitated by a (fair) credit institution. The credit system is essential to help illiquid individuals and companies in need of capital.

However, supplying credit induces risk on both the creditor and the debtor. If the debtor does not repay their credit, the creditor may suffer a loss. Inducing debt on individuals has also been shown to have a negative effect on society. Studies show that having a debt registered with the Swedish Enforcement Authority (Kronofogden) more than doubles the risk of suicide attempt (Rojas, 2023). Thus, the process of evaluating potential customers is critical from both a business and a societal perspective.

In light of the 2008 financial crisis, when major financial institutions like Lehman Brothers failed, the G20 leaders[1] and the Basel Committee on Banking Supervision urged the development of a new method to account for credit losses, IFRS9[2]. According to IFRS9, banks are required to estimate their expected credit loss (ECL). To calculate ECL it is essential to estimate the probability of default (PD). An account has entered default when a debtor is more than 90 days late with their repayment. Under IFRS9, PD can be estimated using three different approaches: a through-the-cycle method, a point-in-time method, or a hybrid of the two (see Figure 1.1). This is to account for variability in PD due to economic cycles (Frykström and Li, 2018).

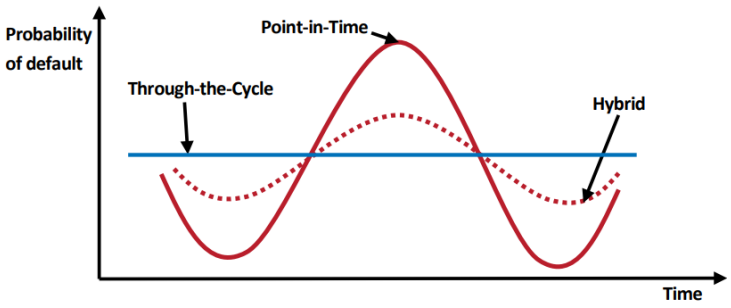


Figure 1.1: Illustration of through-the-cycle, point-in-time and hybrid approach to estimate PD.

[1] Intergovernmental forum including the largest economies.
[2] International Financial Reporting Standard.

This implies that credit institutions are required to collect credit-related data from their customers in order to assess risk parameters such as PD. Credit bureaus, for instance UC (Upplysningscentralen) or FICO (Fair, Isaac & Company), provide this service. In addition to credit history information, they provide a score based on their own models, a *bureau score*. This score is typically a three digit integer value, with lower values indicating lower creditworthiness. This score can be included in models that predict risk components. The following thesis investigates how this score is best integrated within PD models, an ensemble approach or model-in-model approach?

1.2 Problem formulation

There is an ongoing debate on whether credit institutions should adopt a model-in-model (MiM) or ensemble model (EM) approach when developing scorecards (cf. Figure 1.2 & 1.3). The currently prevailing approach among practitioners is MiM. This is reinforced in the academic setting, where authors commonly adopt this method (see for instance Berg et al. (2020)). This approach has the advantage of being easier to implement. In contrast, the EM approach is operationally advantageous due to its modularity. If the bureau score provider updates their scores, the recalibration is simple. However, this thesis looks at the problem mainly from a mathematical point of view. Yielding the final problem formulation:

- Does the Model-in-Model approach outperform the Ensemble Model approach for PD estimation, or vice versa?

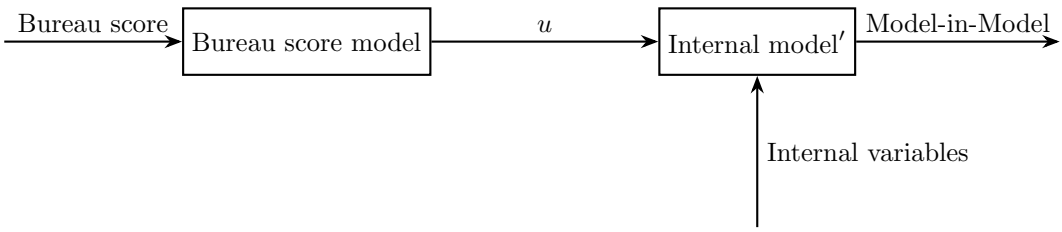


Figure 1.2: Block diagram of the Model-in-Model approach.

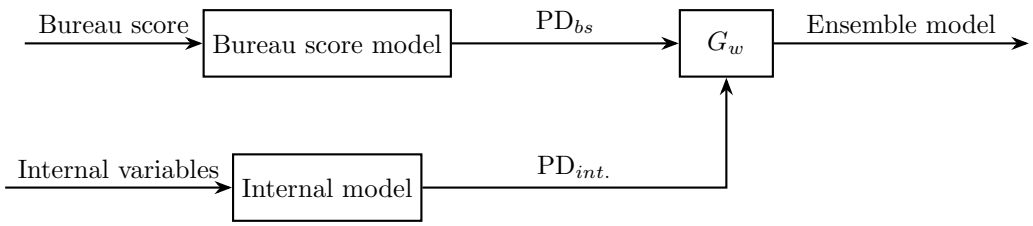


Figure 1.3: Block diagram of the Ensemble model approach.

1.2.1 Methodology

In order to answer the question of which approach is preferable, we conducted an empirical study on real customer data from a bank. The analysis relies on two models commonly applied in credit risk modelling: logistic regression (LR) and extreme gradient boosting (XGBoost).

1.3 Credit Scoring

1.3.1 History of Credit Scoring

Throughout the centuries, the way loan applicants are being evaluated have grown more sophisticated. The modern method for evaluating potential customers is *Credit Scoring*. Credit scoring refers to the statistical method of evaluating the grade of credit risk associated with a certain customer or applicant (Siddiqi, 2017, p. 9). The method originates from the early twentieth century, when applications were manually scored on a number of metrics on a piece of paper, a *Scorecard*. The key metrics commonly used were sociodemographic data, e.g., age, marital status, and salary.

Before the turn of the millennium, most financial institutions outsourced their credit scoring to credit bureaus such as FICO. In later years, the trend has changed and most banks conduct an in-house approach. There are several key drivers for this change. In particular profit optimization and need for transparency (Siddiqi, 2017, pp. 52-54).

With the advancement of computing, the statistical methods used in credit scoring have evolved. Modern credit scoring began with the use of linear discriminant analysis (Roberts, 2019, p. 1). Currently, banks extensively use logistic regression. In later years, advanced machine learning methods, such as support vector machines and neural networks, have increased in popularity (Siddiqi, 2017, p. 17). However, the demand for transparent credit scoring hinders the use of such models in several credit scoring applications.

1.3.2 Transparent Credit Scoring

National law and EU-regulation (e.g., KKrL 13 § & GDPR)[3] stipulates that applicants have the right to know why they are denied a credit. This is the reason for the extensive use of logistic regression and tree based models in consumer credit risk. These models are interpretable and easy to explain to customers.

1.3.3 Application & Behavioural Scorecards

A previous customer and a new customer are generally handled separately. New customers are typically evaluated using an *application scorecard*. Similarly, existing customers are evaluated using a *behavioural scorecard*, where known customer behaviour is leveraged (Siddiqi, 2017, p. 14).

1.4 Motivation

1.4.1 Consumer Credit Institute

This thesis was performed in collaboration with a Swedish consumer bank. Due to confidentiality considerations, the bank has chosen to remain unnamed. We will refer to them by the consumer credit institute or just the bank. In their current scorecard development process, they use a MiM approach. This study was conducted in order to find which approach is preferable.

1.4.2 Related Work

Ensemble methods are well documented in the academic literature. Chi and Hsu (2012) showed that a dual scoring approach, combining a credit bureau score and a behavioural score in a matrix like fashion, outperformed the individual classifiers. Zhu et al. (2001) studied the combination of two consumer credit scores, a bureau score and an application credit score. Using a logistic regression to combine the scores, the performance of the ensemble model exceeded that of the individual scores. In a paper investigating the use of digital footprints in credit scoring, it was shown that the credit bureau score is an influential feature in a logistic regression model (MiM)

[3]Konsumentkreditlag (2010: 1846) and General Data Protection Regulation

(Berg et al., 2020). Although the field is thoroughly traversed, the literature fails to answer our explicit problem formulation.

Final Motivation

The gap in academic literature, together with our relevant data source and empirical methodology, makes us confident that we can contribute to the field.

1.5 Scope

This study is conducted on KGB customers (known-good-bad) from a confidential market. Thus, the models are built solely on customers who have been accepted, which may result in a sample bias. As customer behaviour may differ between markets and time periods, it is not certain that the results are generalizable across future datasets. Moreover, this project concerns application scorecards.

1.6 Thesis Outline

In Chapter 2 we will present the required theory behind the models and statistical tests. Chapter 3 describes the dataset and how it was acquired. In Chapter 4 the modelling workflows are reviewed. The following Chapter 5 exhibits the resulting models, their performance, and comparisons. Chapter 6 discusses the previously found results. The final Chapter 7 presents the concluding ideas.

Chapter 2

Theory

2.1 Models

2.1.1 Logistic Regression

In logistic regression, the outcome variable is p_i, which is defined as the probability that the event of interest occurs for observation i: $\mathbb{P}(Y_i = 1)$. Therefore, it is natural to view each observation as a Bernoulli distributed random variable: $Y_i \in \text{Bin}(1, p_i)$. Since $p_i \in (0, 1)$, ordinary linear regression may not suffice. Conveniently, one can instead predict the log-odds with a linear regression and transform back to a probability. In (2.1) $\boldsymbol{x}_i$ and $\boldsymbol{\beta}$ are the input variables and their coefficients, respectively (Rawlings et al., 1998, pp. 509–510):

$$\text{logodds}_i = \ln \frac{p_i}{1 - p_i} = \boldsymbol{x}_i \boldsymbol{\beta} \implies p_i = \frac{e^{\boldsymbol{x}_i \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i \boldsymbol{\beta}}}. \tag{2.1}$$

The outcomes are binary which means that the log-odds cannot be calculated. Consequently, it is not possible to perform a least squares estimate of the coefficients. Instead, the maximum likelihood method is used. This problem can only be solved numerically (e.g. with the Newton-Rhapson method).

Wald based confidence interval

To quantify the uncertainty in the estimated coefficients, we use the fact that the maximum likelihood estimator is asymptotically normal:

$$\hat{\boldsymbol{\beta}} \sim \text{N}(\boldsymbol{\beta}, (X^T W X)^{-1}) \qquad n \to \infty,$$

where X is the design matrix whose rows contain the covariates of each observation, and W is a diagonal matrix with $w_{ii} = p_i(1 - p_i)$. From this an asymptotic confidence interval of β_j can be constructed as:

$$I_{\beta_j} = (\hat{\beta}_j \pm \lambda_{\alpha/2} \cdot d(\hat{\beta}_j)), \tag{2.2}$$

where $d(\hat{\beta}_j) = \sqrt{(X^T W X)^{-1}}_{jj}$ and $\lambda_{\alpha/2}$ is the quantile $(1-\alpha/2)$ of the standard normal distribution (Agresti, 2007, pp. 106–110).

Odds Ratio

The Odds Ratio (OR) is an interpretation of the coefficients β_j for $j \in 1, \dots n$. The OR is the factor with which the odds change when increasing x_i with one unit:

$$\text{OR}_i = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_i (x_i + 1) + \dots \beta_n x_n}}{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_i x_i + \dots \beta_n x_n}} = e^{\beta_i}. \tag{2.3}$$

2.1.2 Decision Trees

Decision trees are a way to classify or score observations without assuming an underlying distribution for the population (i.e., non-parametric). A decision tree consists of decision nodes, leaf nodes, and one root node. The root node and decision nodes split and group the population according to certain characteristics. The leaf nodes are the final layer in a tree and either classify or score a certain characteristic. A visualisation makes the interpretation of the nodes clear (see Figure 2.1). In the figure a simple non-prejudicial credit scoring model is presented. In the root node, the tree separates men and women. If the applicant is a female, we give them a positive score, if the applicant is a male, we examine their age. If the applicant is a young man, we give them a negative score, otherwise a neutral score.

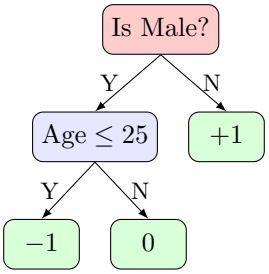


Figure 2.1: Visualization of a simple credit scoring decision tree. Leaf nodes are green. Decision node is purple. Root node is red.

2.1.3 Ensemble Trees & Boosting

Many machine learning methods are rooted in decision trees, so called ensemble tree methods. Rather than relying on a single complex tree, ensemble methods combine many shallow trees. The resulting score is therefore the sum of the individual score of all trees. Suppose we have Harry, a 24 year old man making $\$60k$ annually as a wizard. If we feed him through the ensemble tree in Figure 2.2, he will score -1 in the first tree and $+1$ in the second, i.e., $f(\text{Harry}) = -1 + 1 = 0$.

(a) Tree 1 **(b)** Tree 2

Figure 2.2: Visualization of a simple ensemble tree method.

Now the question arises, how are these trees built? The standard method for constructing trees is boosting. In short, boosting is a sequential approach to build trees that correct the errors of previous trees. The most common boosting algorithm is AdaBoost, short for adaptive boosting (Coadou, 2022). With boosting introduced, we will pivot to XGBoost - Extreme Gradient Boosting.

2.1.4 XGBoost

We will introduce the key aspects of XGBoost. The interested reader can find the algorithm in detail in the original XGBoost paper (Chen and Guestrin, 2016). Training a machine learning algorithm requires an optimisation problem. In tree boosting, the problem is to minimise the loss function found in (2.4). In this equation $\phi(\cdot)$ is the model, $l(\cdot)$ is a convex and differentiable function that penalises the distance between the predictions $\hat{y}_i$ and the target y_i, $\Omega(\cdot)$ penalises

the complexity of the model. In the penalty function $\Omega(\cdot)$, T and w are the number of leaves and leaf weights, respectively, λ and γ are hyperparameters. Moreover, N and K are the number of observations and trees, respectively:

$$\mathcal{L}(\phi) = \sum_{i=1}^{N} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k), \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||_2^2. \tag{2.4}$$

The XGBoost framework builds on the idea that we can perform a second order expansion of the next tree function (2.5). Thus, this means that g_i and h_i are the first and second partial derivatives of $l(y_i, \hat{y}^{(t-1)})$ w.r.t $\hat{y}^{(t-1)}$. Consequently, the problem can be interpreted as, at each iteration t, identifying the new tree f_t that produces the greatest reduction in the loss function:

$$\begin{aligned} \mathcal{L}^{(t)} &= \sum_{i=1}^{N} l\big(y_i, \hat{y}^{(t-1)} + f_t(\mathbf{x}_i)\big) + \Omega(f_t) \\ &\simeq \sum_{i=1}^{N} \left[l\big(y_i, \hat{y}^{(t-1)}\big) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \Omega(f_t). \end{aligned} \tag{2.5}$$

Without being concerned about the precise formulation, these partial differentials can be used to score splits. There are multiple strategies for selecting an optimal split, such as the *exact greedy algorithm* or an *approximate algorithm*. The exact method enumerates *all* possible splits, while the approximate method bins features.

Finally, we introduce two more hyperparameters, shrinkage (commonly referred to as η) and column subsampling. These are two hyperparameters that help reduce overfitting. The shrinkage scales new weights with a factor η after each iteration of the tree boosting. Column subsampling limits the number of features considered when evaluating splits. In later concerned models, we subsample once per tree, but it is also possible to subsample once per node or tree level.

2.1.5 Ensemble Methods

In this context, ensemble methods refer to the act of combining probabilities (cf. G_w in Figure 1.3). There are numerous ways to combine probabilities or scores. In this report, we want to combine the probability estimate from the bureau score $p_{_{BsM}}$ and the internal model $p_{_{IM}}$. The easiest approach is to use a convex combination of the two probabilities and assign weights (Satopää et al., 2014):

$$\hat{p} = w \cdot p_{_{BsM}} + (1 - w) \cdot p_{_{IM}}, \; w \in (0, 1). \tag{2.6}$$

Another approach is to compute the bureau score logodds_{BsM} and the internal model logodds_{IM} and then form weights according to a logistic regression:

$$\text{logodds}_{\text{EM}} = \beta_0 + \beta_1 \text{logodds}_{BsM} + \beta_2 \text{logodds}_{IM}. \tag{2.7}$$

2.2 Model Building

2.2.1 WoE

In consumer credit risk modelling, continuous variables are rarely used. Instead, the characteristics are grouped, and each group is assigned a Weight of Evidence (WoE). This makes it possible to capture non-linear dynamics in a linear model. More specifically, WoE is a measure of the predictive power of each attribute:

$$\text{WoE}_i = \log\left(\frac{\%\text{Good}_i}{\%\text{Bad}_i}\right). \tag{2.8}$$

In this equation, the numerator is the number of goods in category i in relation to all goods[1]. Analogously, the denominator is the number of bads in category i relative to all bads (Siddiqi, 2017, p. 184). This means that if the bad rate is high in one category, then the WoE will be negative. In contrast, the WoE will be positive if the bad rate is low.

2.2.2 IV & Binning

The previous section begs for the question of how to choose attributes or *bins*. Typically, industry knowledge or binning algorithms are used. For example, younger customers are generally considered less creditworthy than older ones. Hence, it may be suitable to group the age variable in three; $18-24$, $25-31$ and $32+$. One can then compute the WoE for each category and substitute these values in place of the original continuous age variable. Another approach is binning algorithms. It is standard practice to form bins by optimising Information Value (IV). IV is a way to measure predictive power of characteristics, it is defined as:

$$\text{IV} = \Sigma_{i=1}^{n}(\%\text{Good}_i - \%\text{Bad}_i) \cdot \log \frac{\%\text{Good}_i}{\%\text{Bad}_i}. \tag{2.9}$$

Generally, attributes with an IV below 0.1 are considered to have weak predictive power, those between 0.1 and 0.3 moderate, and those above 0.3 strong (Siddiqi, 2017, p. 184). With this introduced, it is natural to explain MIV.

2.2.3 Marginal Information Value

The marginal information value (MIV) is a measurement that is used to select features. The method is commonly used by practitioners and originates in a presentation by industry professional Gerard Scallan (Scallan, 2013). MIV is defined as:

$$\text{MIV} = \sum_i (\%\text{Good}_i - \%\text{Bad}_i)(\text{WoE}_i - \text{WoE}_{i,\text{exp}}), \tag{2.10}$$

where $\text{WoE}_{i,\text{exp}}$ is defined as:

$$\text{WoE}_{i,\text{exp}} = \log(\frac{\%\text{Good}_{i,\text{exp}}}{\%\text{Bad}_{i,\text{exp}}}). \tag{2.11}$$

The numerator is the number of goods expected in category i according to the existing model in relation to the total expected number of goods. Analogously, the denominator is the number of expected bads in category i according to the existing model in relation to the expected number of bads in total. Accordingly, the IV formula in (2.9) is a special case of the MIV formula in (2.10), where the model is the naive model predicting all observations equally. The MIV modelling procedure is continued until the remaining variables are deemed no longer significant. This is defined as the remaining MIV being less than 0.02 or the marginal χ^2 p-value being greater than 5% (Scallan, 2013, p. 11).

2.2.4 Hyperparameter Tuning

Observing (2.4 & 2.5) we notice that the loss function $\mathcal{L}$ has hyperparameters that need to be specified. These parameters are found in Table 2.1. The techniques used in this project for tuning hyperparameters are *Bayesian Optimization* and *Randomized Grid Search*.

Randomized Grid Search

Randomized Grid Search is a way to find a suitable set of hyperparameters without testing all possible combinations on the grid. Hyperparameters are sampled without replacement and evaluated using cross validation. This is run for a prespecified n-iterations and the highest scoring model is returned. In this project, the scoring method used is binary cross entropy:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} -y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i). \tag{2.12}$$

[1]See Dictionary for definitions of good & bad.

Table 2.1: Hyperparameters tuned for XGBoost.

Parameter	Definitions
`K`	Number of trees
η	Shrinkage
`max_depth`	Maximum depth of one tree
γ	Penalty parameter for number of leaves
λ	L_2 regularization term
`colsample_bytree`	Subsampling rate per tree (features)
`subsample`	Subsampling rate per tree (observations)

Bayesian Optimization

Bayesian optimization is an algorithm widely used in hyperparameter tuning for machine learning models. The method is used when the objective function f is expensive to evaluate. This is a complex algorithm, and it would be beyond the scope of this report to explain it in detail. The main idea can be understood by observing Algorithm 1.

Algorithm 1: Basic pseudo-code for Bayesian optimization (Frazier, 2018)

Place a Gaussian process prior on f
Observe f at n_0 points according to an initial space-filling experimental design Set $n = n_0$
while $n \leq N$ **do**
    Update the posterior probability distribution on f using all available data
    Let x_n be a maximizer of the acquisition function over x, where the acquisition function is computed using the current posterior distribution
    Observe $y_n = f(x_n)$
    Increment n
end
return Either the point evaluated with the largest $f(x)$, or the point with the largest posterior mean

In the later implementation of the optimization schema, f is the complete mapping of hyperparameters and data to a final model score. In our case, the final model score is the binary cross entropy (2.12). To sample new hyperparameters, the acquisition function is optimized. There are several types of acquisition functions, but in general they are based on the posterior distribution of f. The acquisition function used in this project is the *expected improvement* (2.13). In this equation, f_n^* is the score of the best hyperparameters yet found, and the expectation is under the posterior distribution, i.e., $\mathbb{E}_n[\cdot] = \mathbb{E}[\cdot|x_{1:n}, y_{1:n}]$:

$$\mathrm{EI}_n(x) := \mathbb{E}_n[[f(x) - f_n^*]^+]. \tag{2.13}$$

K-fold Cross Validation

Cross Validation is a widely applied technique for reliably assessing model performance. Both Bayesian optimization and randomized grid search leverages cross validation. The idea is to split the data in k-folds, utilize $k-1$ folds to train the model and then evaluate on the holdout (the 1 remaining fold). This is repeated until all k folds have been used as holdout (i.e. k iterations). The results across the k-folds are then averaged out (James et al., 2021, pp. 203–205). A visualization can be found in Figure 2.3.

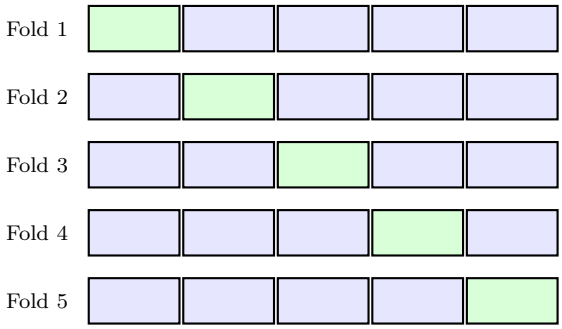


Figure 2.3: k-Fold Cross Validation ($k = 5$).

2.2.5 Feature Importance

Tree boosting models are in general difficult to interpret. They can easily involve thousands of splits, which makes it troublesome to determine which features are significant. This is the use case of *feature importance*. For XGBoost models there are two main methods to evaluate feature importance, split based and gain based. The split based method is a measure that counts the occurrences a feature is utilized in splits. More splits, higher importance. The gain based approach measures the average information gain of a feature split. Higher average gain, higher importance.

In this thesis all feature importance plots measure the information gain. Recalling g_i and h_i from (2.5), we obtain that the gain associated with a given split is defined as (Chen and Guestrin, 2016):

$$\mathcal{L}_{\text{split}} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma. \tag{2.14}$$

In this equation I_L and I_R are the instance set of nodes left and right to the split and $I = I_L \cup I_R$. Notably, this expression is dependent on the choice of hyperparameters.

2.2.6 Ad hoc Elbow Plot

The elbow method is a popular method used to determine the K in K-means clustering. Plotting the percentage of variance explained against the number of clusters we pick the point that gives best trade-off between fit and complexity. If we consider the resulting curve as an arm, this is the point that resembles the elbow (Bholowalia and Kumar, 2014). Inspired by this, we constructed our own method that plots the number of features in an XGBoost model against performance on test data (AUC-score) (cf. Figure 5.6). Without having to stretch our imagination too far, this curve resembles an arm. However, instead of looking at the elbow, the point of interest is where performance begins to worsen. This is the point where the curve goes from flat to trending downward.

2.3 Model Validation

"All models are wrong, but some are useful" $\sim$ George Box is a cliche (but alas so true) aphorism commonly mentioned in statistics. To determine whether models really are useful we need to validate them. This section introduces which techniques are used when validating our models.

2.3.1 ROC & AUC

A popular tool to evaluate classification performance is the receiver operating characteristics curve (ROC-curve). On the x-axis is the specificity reversed and on the y-axis is the sensitivity. The specificity is also known as the true negative rate and the sensitivity as the true positive rate. More specifically:

$$\begin{aligned} \text{Specificity} &= \mathbb{P}(\hat{Y}_i = 0 | Y_i = 0) \\ \text{Sensitivity} &= \mathbb{P}(\hat{Y}_i = 1 | Y_i = 1). \end{aligned} \tag{2.15}$$

These metrics depend on a threshold probability. If we map the specificity and sensitivity together with multiple thresholds we get a curve, a ROC-curve (Agresti, 2007, pp. 142–144) (cf. Figure 5.11). A useful model will have both high sensitivity and specificity. Thus, we want the curve to stay close to the top left corner.

We are now ready to define AUC, AUC is an acronym for Area Under the Curve and is a measure of how well the model can order observations. Accordingly, a model perfectly classifying observations will have AUC equal to 1 and a model no better than guessing will have AUC equal to 1/2. In consumer credit risk modelling, it is common to look at the Gini coefficient. The Gini coefficient maps uniquely to AUC according to the formula:

$$\text{Gini} = 2 \cdot \text{AUC} - 1. \tag{2.16}$$

2.3.2 Kolmogorov-Smirnov

The Kolmogorov-Smirnov statistic is a non-parametric method used to test if two distributions differ. In our case, we want to determine whether the distribution of ordered goods and bads are different. It is defined as follows:

$$D = \max_x |F_{\text{bad}}(x) - F_{\text{good}}(x)|. \tag{2.17}$$

Begin with sorting the observations with respect to PD, from high to low. Walking through the whole dataset we calculate the cumulative percent of goods and bads, $F_{\text{good}}(x)$ and $F_{\text{bad}}(x)$ (Massey, 1951). As we expect the model to correctly classify bads, initially $F_{\text{bad}}(x)$ should be high and $F_{\text{good}}(x)$ low. Thus a high statistic indicates that the distributions differ and that the model captures characteristics well.

2.3.3 Bootstrap

Bootstrap is a statistical method used to estimate the uncertainty associated with estimated statistics. Traditionally, it is defined as (James et al., 2021, pp. 209–212): Let Z denote the original dataset with N observations. Let α be the statistic of interest and B the number of bootstrap samples. We generate B new datasets, each containing N observations, by sampling with replacement from Z. We denote these datasets by $Z_1, Z_2, \ldots, Z_B$. For each Z_i we create an estimate α_i. Finally the average of the estimates is calculated:

$$\hat{\alpha} = \frac{1}{B} \sum_{i=1}^{B} \alpha_i. \tag{2.18}$$

The uncertainty of the estimate can be quantified using the bootstrap estimate of the standard error:

$$\mathrm{SE}_B(\alpha) = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}(\hat{\alpha} - \alpha_i)^2}. \tag{2.19}$$

2.4 Model Comparison

A hypothesis test can be performed to determine if the performance of the models differ significantly. If we denote the result from one model, R_1 and the result from another model, R_2. We define the null and alternative hypothesis as:

$$\begin{aligned} H_0 &: R_1 = R_2 \\ H_a &: R_1 \neq R_2. \end{aligned} \tag{2.20}$$

There are various tests one can use to evaluate this hypothesis. In this section we present two of them, the paired Student's t-test and Wilcoxon signed rank test.

2.4.1 Paired Student's t-test

The paired Student's t-test is used to determine if the mean of two populations differ. We have samples from two populations R_1 and R_2, where the samples are paired, R_1^i and R_2^i. In our case, they originate from the same sampled dataset. We denote the expected difference $\mu_d = E[R_1 - R_2]$ and want to test the hypothesis:

$$\begin{aligned} H_0 &: \mu_d = 0 \\ H_a &: \mu_d \neq 0. \end{aligned} \tag{2.21}$$

We can then form the difference $z_i = R_1^i - R_2^i$ for each sample, calculate $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$, and the sample standard deviation $s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{n}(z_i - \bar{z})^2}$. The test statistic is defined as $t = \frac{\bar{z}}{s/\sqrt{n}}$. If H_0 is true, t should be Student's t-distributed (using normal approximation). Normal approximation generally requires that the the different z_i are independent (Thukral et al., 2023).

2.4.2 Wilcoxon Signed Rank Test

The Wilcoxon signed-rank test uses the same paired samples as the previous test, but tests if the median differs from zero. This is done in the following manner: We calculate the difference $D_i = R_1^i - R_2^i$ for all pairs and assume $D_i \neq 0$. We assume D_i comes from an arbitrary distribution F with a median m, and test the hypothesis:

$$\begin{aligned} H_0 &: m = 0 \\ H_a &: m \neq 0. \end{aligned}$$

We then calculate the absolute value $|D_i|$, sort them and assign ranks r_i to all observations, with r_i=1 for the observation with the smallest $|D_i|$, $r_i = 2$ for the second smallest, and so forth until $r_i = n$ for the largest $|D_i|$. The signed rank test statistic becomes:

$$S = \sum_{i=1}^{n} r_i I(D_i > 0) - \frac{n(n+1)}{4}, \tag{2.22}$$

where $I(D_i > 0) = 1$ if $D_i > 0$ and $I(D_i > 0) = 0$ if $D_i < 0$. The variance of S is given by:

$$V = \frac{1}{24}n(n+1)(2n+1) - \frac{1}{48}\sum_{j}^{m}(t_j(t_j+1)(t_j-1), \tag{2.23}$$

where t_j is the number of tied absolute values of D_j and m is the number of unique ranks. If $|D_i| \neq |D_j|$ for all $i \neq j$ then, the variance in (2.23) can be simplified:

$$V = \frac{1}{24}n(n+1)(2n+1). \tag{2.24}$$

Then normal approximation yields the test statistic:

$$Z = \frac{S}{\sqrt{V}},$$

since $E[S] = 0$ if the null hypothesis is true (Harris and Hardin, 2013).

Chapter 3

Data

3.1 Collection Process

Before describing the data used in this study, the data collection process is explained. Credit scoring models are developed using historical customer application data combined with performance data to predict risk components such as PD. The collection process is as follows:

1. Credit application process: A potential customer applies for credit. During the application process, the customer provides information related to sociodemographic and financial characteristics. This information is uploaded to the bank's internal application database.
2. The application data is extracted from the bank's internal application database. The dataset contains information available at the time of application.
3. Application data is further augmented with credit bureau information. Credit bureau information includes individuals credit behaviour. E.g., existing loans, repayment history, delinquencies and credit utilization.
4. Application and bureau data are linked with performance data, i.e., observed repayment outcomes after credit/loan pay out to examine whether an account have entered default.

3.2 Description

The given dataset contains approximately 17,000 applications for credit from a confidential market during the time period $2023 - 2025$. In the given dataset, all observations are accepted customers (see Figure 3.1). This means that all observations have either entered default or not. The observation is registered as good if the customer have not entered default within 12 months, and bad otherwise. The default definition can also be considered on a longer time horizon, e.g., 18 months. See Appendix B for a complete overview of the variables used in this thesis.

3.3 Random Downsampling

The proportion of approved applicants who enter default in consumer lending portfolios represents a small share of the population. In order to build accurate models, the bad rate in the modelling data must be sufficiently high. To address this issue, goods are randomly downsampled. This was not performed by the authors, the provided dataset had a sufficient bad rate of approximately 15%.

3.4 Raw Information Values

The information values for all binned variables in the original training dataset are shown in Figure 3.2. As expected, the bureau score looks like a super-variable. Moreover, 15 features are considered to have moderate predictive power, and 27 to have weak predictive power.

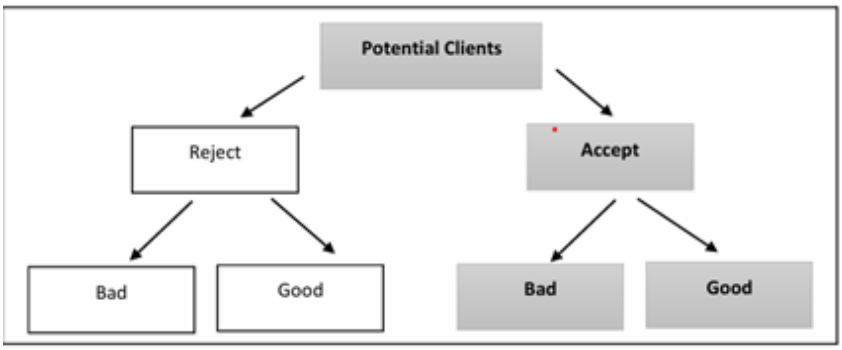


Figure 3.1: Illustration of development data used in this thesis.

3.5 Time Dependancy

During the modelling, it was found that the variable that explained whether an applicant was on social security or not had strong predictive power. What was strange about this characteristic was not that the classes 'Yes' or 'No' had predictive power. The predictive power was boiled down to whether the variable was 'Missing' or 'Not Missing'. In October of 2023, there was a policy shift that raised the acceptance threshold. This is clearly visualised in Figure 3.3. Note that the customer who breaks this pattern in March of 2024 is likely an override (a manually approved customer). Consequently, the proportion of bads was significantly lower after this policy change. Close in time to this policy shift, the bank stopped collecting this variable. The variable acted as a proxy for the policy shift. Hence the variable would not be useful to predict future defaults and was not used in further modelling. Moreover, the policy shift had an influence on the distribution of other features, in Table 3.1 we can see the mean for two critical features and the dependent variable before and after October 2023.

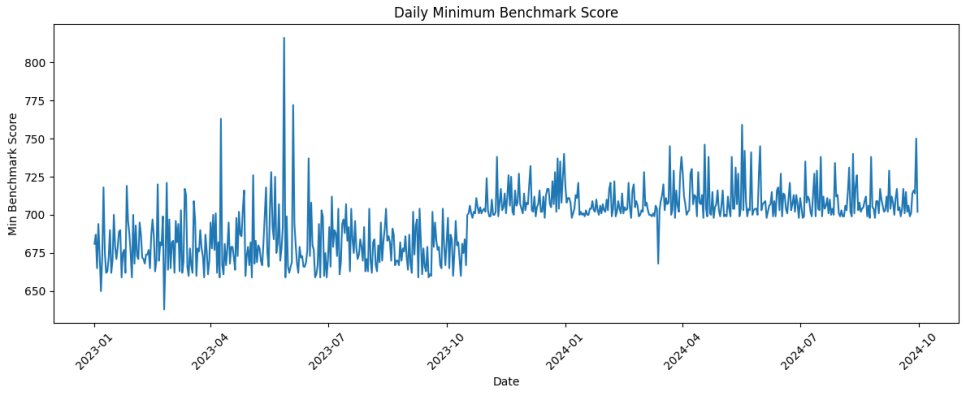


Figure 3.3: Plot of daily minimum benchmark score over the full time period.

Table 3.1: Mean for different variables before and after policy shift.

Variable	Mean-Before	Mean-After
Bad	0.1731	0.1318
AGE	38.00	39.16
Income_to_limit	0.8374	0.9682

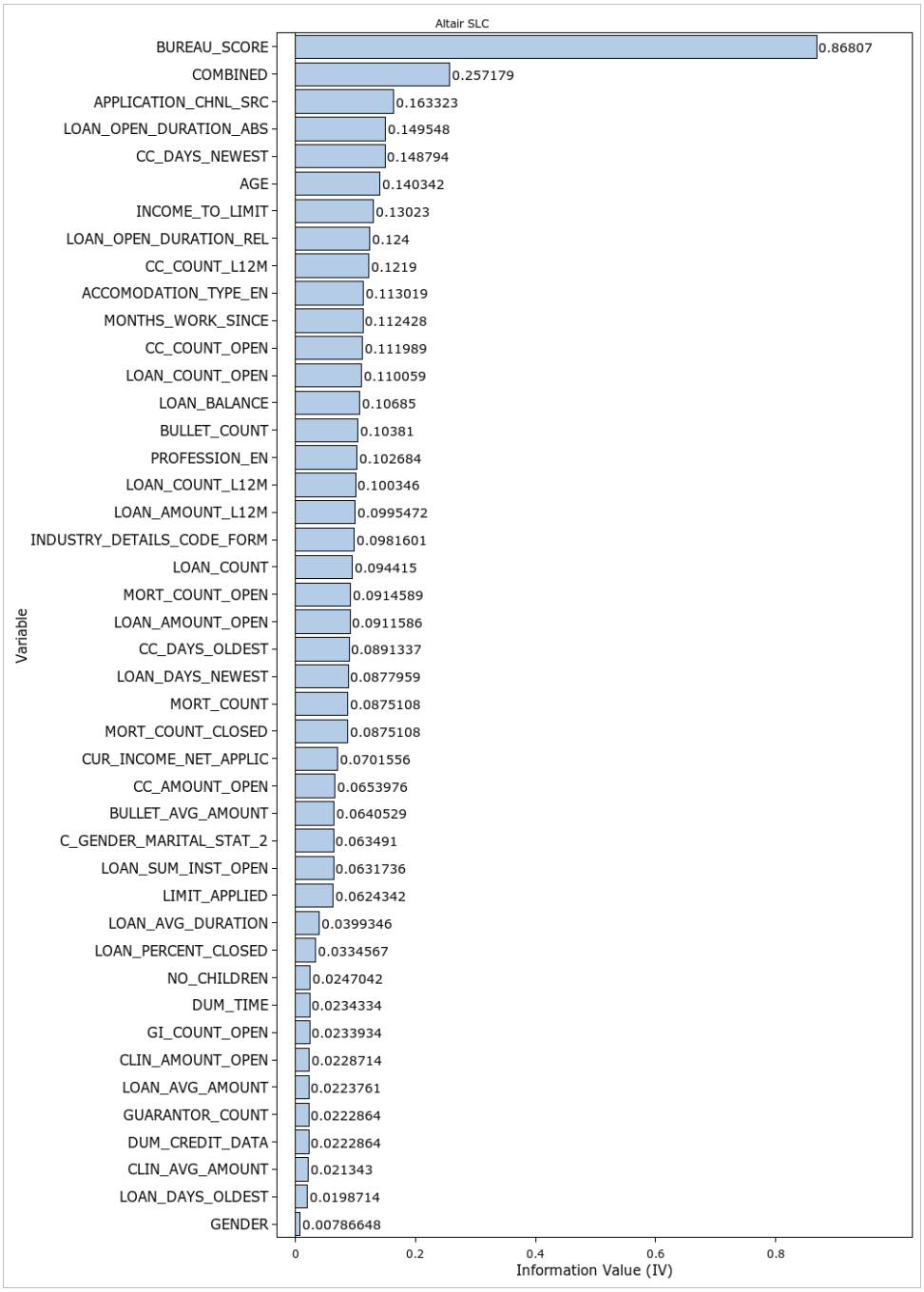


Figure 3.2: Plot of IV for all variables in the dataset.

Chapter 4

Method

4.1 Data Method

Similar to most projects involving real data, significant time was spent on pre-processing. As the dataset contained 76 variables and roughly 17,000 observations some strange things where bound to happen.

4.1.1 Zeros & Missing

Initially some variables could be removed due to high missing rates. Some variables had suspicious values, e.g., the woman with 64 children or the lady who have worked for -12,000 months. Characteristics like this was handled the same as having no information at all, i.e., missing. Other variables, generally those revolving credit history, had missing values that needs to be interpreted carefully. For example number of open credit cards. If this value is empty the reasonable interpretation is that the customer has never had a credit card, i.e., should be zero. But this could also mean that the value is missing. In order to determine the correct interpretation, the value was put into context with other variables. However, a zero in this column means that the customer have had a credit card but not currently. These are two different characteristics and must be handled differently.

Flags

To allow for different interpretations, we introduced different flags. The first flag was a true missing value, like the mother of 64 children. This was flagged differently between features to bin them appropriately. The other flag was a true zero, e.g., a customer who has never owned a credit card. This was denoted as -998.

WoE Binning

As discussed previously, we flag true missing values differently. In the WoE binning process, these observations were binned together with the characteristic with the most similar bad rate.

NaN Interpretation

Similarly, when the XGBoost models were built, the missing values were converted to `NaN`. This is a data type supported by XGBoost.

4.1.2 Feature Engineering

Part of the processing stage revolves around creating new features to better explain the PD. Using domain knowledge and heuristic approaches to combine raw features, we can improve predictive power (Guyon and Elisseeff, 2003). In this project, we created two new combinations of features that had substantial predictive power.

Income to Applied Credit Limit

The first feature created was based on the idea that you should not be able to use more credit than your income. Using that, we constructed a feature that describes the ratio between the applicants net income and applied credit limit. This feature has moderate predictive ability (cf. INCOME_TO_LIMIT in Figure 3.2).

Age & No. of Credit cards

The second feature originates in the thought that having many credit cards is worse credit-wise if you are young rather than old. This combination was shown to hold moderate predictive power (cf. COMBINED in Figure 3.2).

Other Combinations

In addition to these two new features, there were also some combination features that were provided to us. These are recognized in Figure 3.2 by having the prefix C_. Some of these were shown to have weak predictive capability. A more detailed description of these variables can be found in Appendix A.

Time Dependancy

To handle the time dependancy described in Section 3.5, we introduced a dummy variable that explains whether or not the customer applied before or after the policy shift:

$$\texttt{dum\_time} = \begin{cases} 1 & \text{if appl. date after date of policy shift,} \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

Credit History Void

Some observations had missing values across all credit history data. We suspect that this may be a certain characteristic. Thus, we created a dummy variable that explains if the credit related data is voided:

$$\texttt{dum\_credit\_data} = \begin{cases} 1 & \text{if credit history is non-empty,} \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

4.1.3 Train-Test split

In the modelling procedures, we chose to split the data $70 - 30$ and stratify based on default, i.e., assign 70% of observations to the training set and 30% to the test set (with an equal bad rate between the two groups). The split was chosen heuristically since we have a sufficiently large dataset.

4.1.4 WoE-Transformations

Using the binning techniques introduced in Section 2.2.2, we yield the WoE-transformed variables that were used in the logistic regression models. A detailed description of the bins and their corresponding WoE can be found in Appendix B.

4.2 Logistic Regression Method

The logistic regression modelling procedure in this project consisted of three main parts (cf. Figure 4.1). Firstly, we processed the data (Section 4.1). Following that, we sampled a train-test split (Section 4.1.3). After processing was completed, the variables were transformed into WoE categories (based only on training data). To perform these splits, both heuristic and numerical methods were used. For example, the binning procedure optimised IV while constraining the number and size of the bins. An example of how the bins look can be seen in Table 4.1.

Figure 4.1: Workflow for logistic regression modelling.

Table 4.1: Example WoE for the variable Income_To_Limit.

Income_To_Limit	**Total**	**Total%**	**Bads**	**Badrate**	**WOE**$_i$	**IV**$_i$
$(-\infty, 0.5055]$	4083	32.66	852	20.87	-0.41244	0.063919
(0.5055,0.836333]	2402	27.21	499	14.67	0.015492	0.0000065
(0.836333, ∞)	5016	40.12	507	10.11	0.439918	0.066333

The subsequent stage in the modelling process involved determining which features would be included in the final model. The procedure was similar in both the MiM and EM cases. We started with a null model (intercept for EM and only bureau score for MiM) and added the variable with the highest IV. Following that we sequentially added variables with the highest MIV (2.10). This process was repeated until no further variables were deemed significant.

Model-in-Model

In the MiM case the model is complete when no more variables are significant according to MIV.

Ensemble Model

Following the MIV procedure in the EM case, the internal model (IM) needs to be combined with the bureau score model (BsM). To clarify, the bureau score model is a logistic regression with *only* bureau score as input. Both approaches described in Section 2.1.5 were tried out. The approach utilizing a regression (2.7) was found to perform better. After these regressions have been estimated, the EM model is complete.

4.3 XGBoost Method

The XGBoost modelling procedure was made up of four main parts (see Figure 4.2). Preprocessing, finding initial hyperparameters, selecting features and tuning the hyperparameters. Since we had already developed two logistic regression models, most of the data preprocessing had already been done.

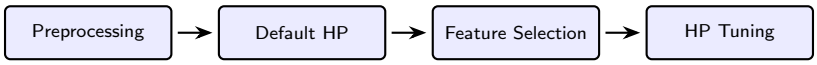


Figure 4.2: Workflow for XGBoost modelling.

Following the preprocessing, a simple randomized grid was performed to find parameters used going in to the feature selection stage. For this we used the implementation `RandomizedSearchCV` from the open source Python library `scikit-learn`. The feature selection was performed according to Algorithm 2 and the ad hoc elbow method described in Section 2.2.6. In short, when removing features starts to deteriorate the AUC-score noticeably we stop eliminating features. After the features were selected, it was of interest to tune the hyperparameters. This was performed using an implementation of the *Bayesian Optimization* algorithm (`BayesSearchCV`, also from `scikit-learn`).

Ensemble Model

Combining the XGBoost model with the bureau score required two decisions. First, we had to select a model for the bureau score itself. Then, we had to choose a model to ensemble the resulting scores. The initial idea was to again use a XGBoost approach. However, the BsM has one variable and the EM has two variables (the model scores), a tree boosting approach for this few variables is therefore not suitable. Hence, we chose to once again use logistic regression for the BsM and EM.

Algorithm 2: Backward Feature Elimination

Input: Default model and p features
Output: AUC and eliminated variables
for $i = 1$ **to** p **do**
    Remove least influential feature in the model from previous iteration (or input model if $i = 1$)
    Retrain model with default hyperparameters on the remaining $p - i$ features
    Evaluate model on test, store AUC_i & eliminated feature
end

What differs from the LR models is that we used bureau score as a continuous variable. This is to keep the same interpretation of bureau score as in the XGBoost model.

4.4 Ad hoc Bootstrap

The traditional bootstrap, described in Section 2.3.3 will be the inspiration for our evaluation method. We use the Gini coefficient from (2.16) and the KS from (2.17) as performance measures. We denote the full dataset as Z with N observations. Next, we generate $n = 100$ new samples from Z, stratified by default. Each sample contains one train set and one test set. Which gives us $Z^1_{train}, \dots Z^n_{train}$ all containing $0.7N$ unique observations and $Z^1_{test}, \dots, Z^n_{test}$ all containing $0.3N$ unique observations. We also have that $Z^i_{train} \cap Z^i_{test} = \emptyset$.

We then train the MiM and EM on Z^i_{train} and evaluate them on their respective test data Z^i_{test}. For each model and evaluation, we obtain a test statistic α^i_{MiM} and α^i_{EM}. We then treat them as paired samples and define the difference $\Delta^i = \alpha^i_{MiM} - \alpha^i_{EM}$. To determine the expected value of Δ, one can employ the natural estimator:

$$\Delta^* = \frac{1}{n} \sum_{i=1}^{n} \Delta^i. \tag{4.3}$$

Where the sample standard error is:

$$\text{s}(\Delta) = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (\Delta_i - \Delta^*)^2}. \tag{4.4}$$

Using normal approximation a confidence interval for Δ can be constructed:

$$I_\Delta = \left(\Delta^* - \lambda_{\alpha/2}\, d,\ \Delta^* + \lambda_{\alpha/2}\, d\right) \tag{4.5}$$

where $d = \frac{\text{s}(\Delta)}{\sqrt{n}}$, is the standard error and $\lambda_{\alpha/2}$ is the quantile corresponding to the confidence level $1 - \alpha$. (Blom et al., 2017, pp. 208–300)

4.5 The Answer to Life, the Universe and Everything

In this project there were several endeavours that required a random seed, e.g. stratified sampling on default. The seed chosen was the answer to life, the universe and everything $(9 \cdot 6)_{13}$[1] (Adams, 1979).

[1] 42

4.6 Benchmark Model

To ensure our models were sufficiently accurate, they were compared against a benchmark model provided to us by the bank. The benchmark score is a three digit integer value. Analogously to the bureau score, a higher value corresponds to a lower PD. The benchmark scores, in the provided dataset, ranges between 638 and 993. The model structure behind the scores is unknown, i.e., there is no unique mapping back to a PD. However, we can still view the ordering capabilities via Gini. It is important to note that performance of a model decays with time. When we train a model on up-to-date data, the new model will almost always outperform the old one. This means that we are not satisfied solely because we outperform the benchmark.

Chapter 5

Results

In this chapter the resulting models and comparisons between them are presented. Confidence intervals and tests presented are always on the 95% significance level unless explicitly said otherwise. In Table 5.1 the performance of the different models are shown. Note that the test and train split is not the same for LR as XGBoost. This is due to the fact that the LR models were built in SAS while the XGBoost models were built in Python. Moreover, it was not of interest to compare the LR models to the XGBoost models in detail.

Table 5.1: Model performance on modelling test-train split.

	Benchmark	MiM LR	EM LR	MiM XGBoost	EM XGBoost
Gini Train	–	0.6104	0.6077	0.8196	0.7709
KS Train	–	0.4702	0.4575	0.6577	0.6068
Gini Test	0.5034	0.6055	0.5994	0.6454	0.6406
KS Test	0.3775	0.4577	0.4577	0.4999	0.4900

5.1 Benchmark Results

The performance of the benchmark model will be presented in form of a Gini and KS score. The statistics are based on all observations who had a non-missing benchmark score. The Gini and KS test statistic can be seen in Table 5.1. Recall that we do not know the structure of the model, we only have a score for each observation. Hence, the test statistics shown are the statistics on all available data. The ROC-curve of the benchmark will be included in the presentation of later models.

5.2 Logistic Regression Results

The first two models built were the logistic regression models, one MiM and one EM. We start of with presenting results for the two different approaches.

5.2.1 Model-in-Model

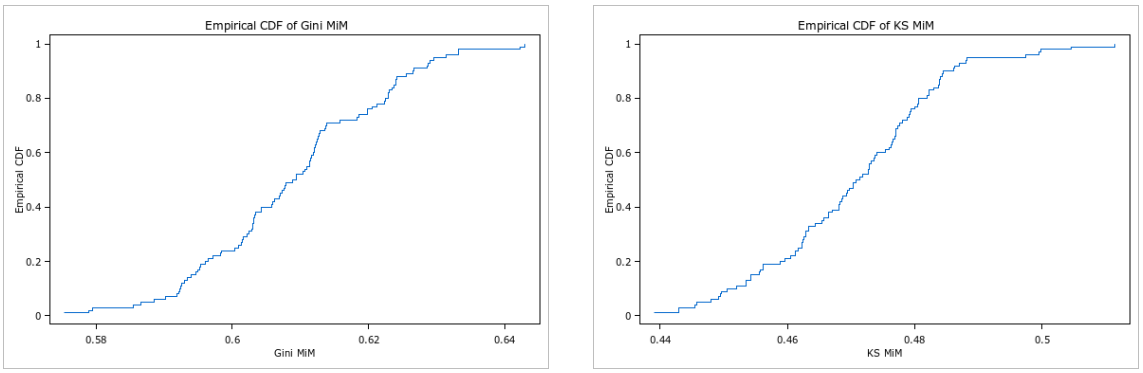
The first regression presented uses the bureau score as a feature within the model, the LR MiM. This model was acquired by following the methodology described in Section 4.2. In Table 5.2 the odds ratios and their Wald confidence intervals are presented. The resulting Gini and KS on train and test data can be found in Table 5.1. To ensure stability over sample splits, the bootstrap described in Section 4.4 was carried out. In Appendix C.1 the bootstrap results for the coefficients are presented. The resulting ROC-curve on the test data is presented in Figure 5.3. In Figure 5.1 the empirical bootstrap distributions for MiM Gini and KS are shown.

Table 5.2: Odds ratios with Wald confidence intervals for LR MiM coefficients.

Effect	Lower	Point Estimate	Upper
WOE_APPLICATION_CHNL_SRC	0.473	0.544	0.627
WOE_Industry_details_code_form	0.368	0.442	0.531
WOE_ACCOMODATION_TYPE_EN	0.525	0.621	0.735
WOE_PROFESSION_EN	0.356	0.426	0.510
WOE_BULLET_COUNT	0.408	0.486	0.580
WOE_LOAN_BALANCE	0.434	0.518	0.619
WOE_C_GENDER_MARITAL_STAt_2	0.433	0.536	0.663
WOE_combined	0.588	0.659	0.739
WOE_CUR_INCOME_NET_APPLIC	0.391	0.497	0.632
WOE_LIMIT_APPLIED	0.269	0.374	0.521
WOE_Income_To_Limit	0.558	0.698	0.873
WOE_bureau_score	0.417	0.446	0.476
dum_credit_data	2.117	2.944	4.093
Intercept	0.044	0.061	0.084

Table 5.3: Odds ratios with Wald confidence intervals for LR IM.

Effect	Lower	Point Estimate	Upper
WOE_Income_To_Limit	0.582	0.725	0.903
WOE_APPLICATION_CHNL_SRC	0.394	0.453	0.520
WOE_Industry_details_code_form	0.408	0.489	0.585
WOE_MONTHS_WORK_SINCE	0.497	0.596	0.715
WOE_BULLET_COUNT	0.388	0.461	0.549
WOE_PROFESSION_EN	0.335	0.401	0.480
WOE_CC_COUNT_L12M	0.430	0.506	0.596
WOE_C_GENDER_MARITAL_STAt_2	0.406	0.500	0.616
WOE_CC_DAYS_OLDEST	0.365	0.458	0.576
WOE_ACCOMODATION_TYPE_EN	0.539	0.641	0.761
WOE_LOAN_BALANCE	0.425	0.534	0.671
WOE_LOAN_OPEN_DURATION_ABS	0.339	0.411	0.498
WOE_CUR_INCOME_NET_APPLIC	0.385	0.487	0.617
WOE_LIMIT_APPLIED	0.237	0.328	0.454
WOE_MORT_COUNT	0.409	0.508	0.633
WOE_combined	0.479	0.537	0.603
dum_credit_data	9.510	14.015	20.654
Intercept	0.00913	0.0134	0.0197



(a) Empirical CDF of Gini for MiM.

(b) Empirical CDF of KS for MiM.

Figure 5.1: Empirical CDF of Gini and KS for LR MiM.

5.2.2 Ensemble Model

The EM consists of three models: an internal model, a bureau score model, and an ensemble model. The internal model was built analogously to the MiM. The other two models are simple regressions with one and two variables, respectively. Following the methodology in Section 4.2 we acquire the three final models. The variables used, their estimates, and Wald confidence intervals are shown in Table 5.3, 5.4 and 5.5, respectively. Observing the coefficients of the EM, we notice that the OR corresponding to the IM is higher than the BsM. This suggests that the internal model is more influential.

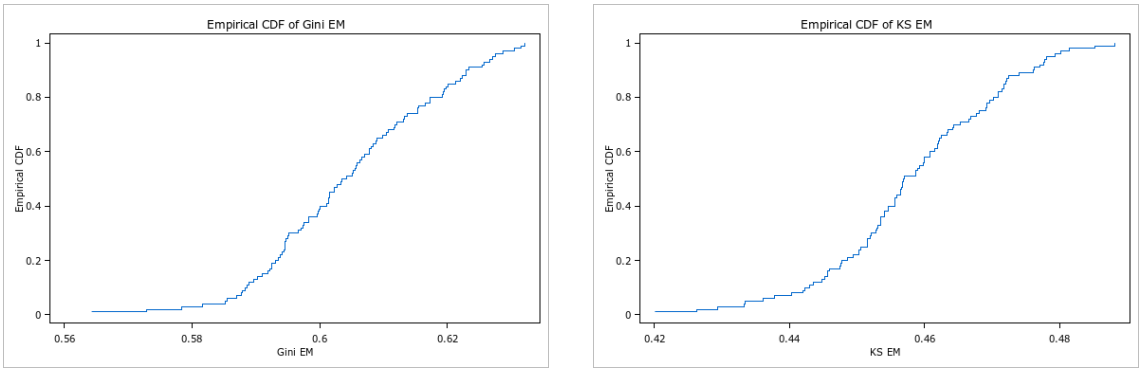
Table 5.4: Odds ratios with Wald confidence intervals for LR BsM.

Effect	Lower	Point Estimate	Upper
WOE_bureau_score	0.346	0.368	0.391
Intercept	0.165	0.175	0.184

Table 5.5: Odds ratios with Wald confidence intervals for LR EM.

Effect	Lower	Point Estimate	Upper
logodds_model	2.048	2.171	2.302
logodds_bureau	1.712	1.834	1.965
Intercept	1.703	1.918	2.161

Analogously to the previous section, bootstrap was performed as described in Section 4.4. The bootstrap statistics for the coefficients are presented in Appendix C.2, C.3 and C.4. In Figure 5.2 the empirical bootstrap distributions of Gini and KS are presented. The resulting ROC-curve on the test data is presented in Figure 5.3.



(a) Empirical CDF of Gini for EM.

(b) Empirical CDF of KS for EM.

Figure 5.2: Empirical CDF of Gini and KS statistics for LR EM.

5.2.3 Comparison

The ROC-curve on the test data, shown in Figure 5.3, indicates that the difference between models is negligible. In Table 5.1, we can see that MiM has a small advantage over EM in terms of Gini, but both models perform equal in terms of KS. To investigate the matter further we need to observe the bootstrap results.

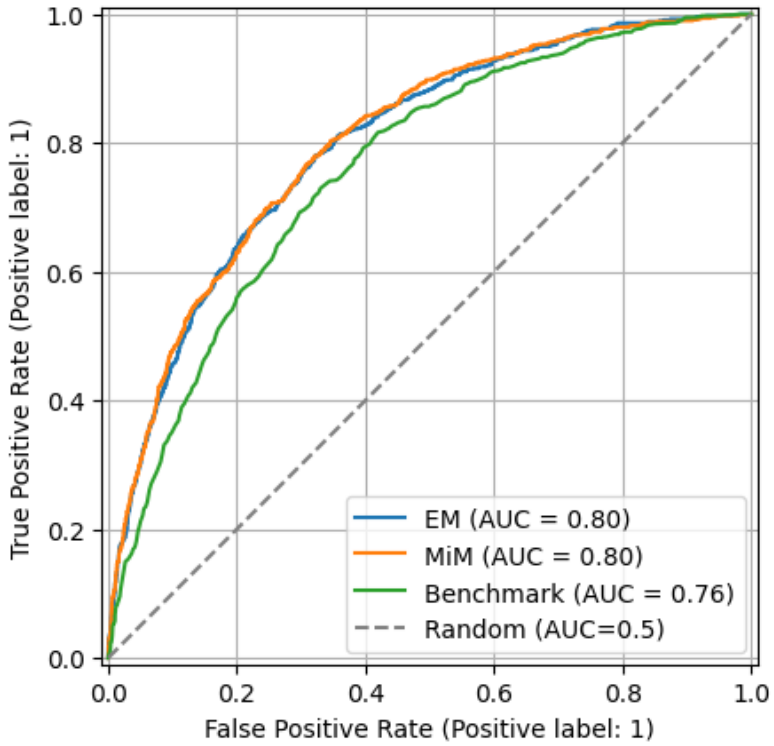


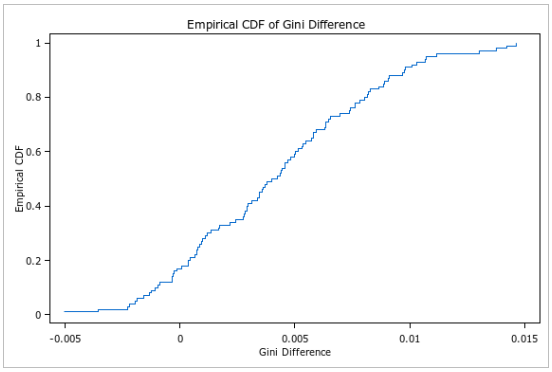
Figure 5.3: ROC-curves for LR models on modelling test data.

To compare the models over the bootstrap splits, the differences in Gini and KS are calculated:

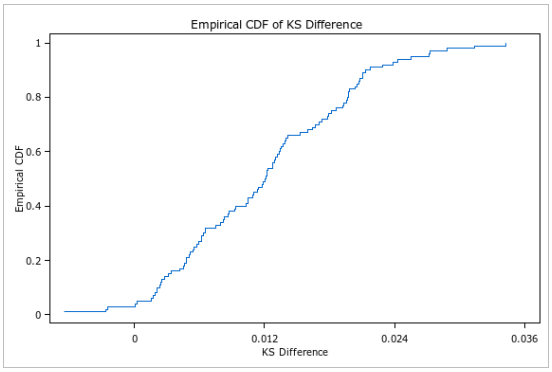
$$\Delta_i^{\text{Gini}} = \text{Gini}_i^{\text{MiM}} - \text{Gini}_i^{\text{EM}}. \tag{5.1}$$

$$\Delta_i^{\text{KS}} = \text{KS}_i^{\text{MiM}} - \text{KS}_i^{\text{EM}}. \tag{5.2}$$

The empirical distributions of the differences are plotted in Figure 5.4. The body of the distributions are positive, which implies that the MiM generally has higher Gini and KS. The box plots for the two deltas are shown in Figure 5.5. In both the Δ^{Gini} and Δ^{KS} plots, zero is within two standard deviations.

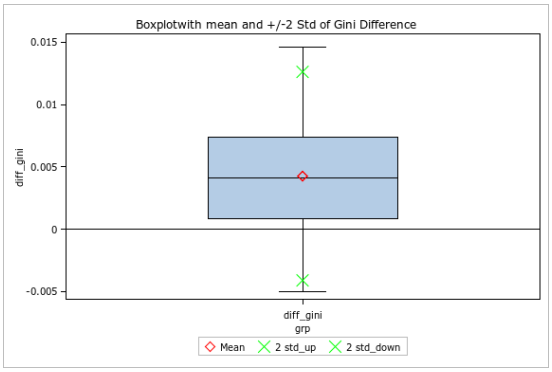


(a) Empirical CDF of Δ^{Gini}.

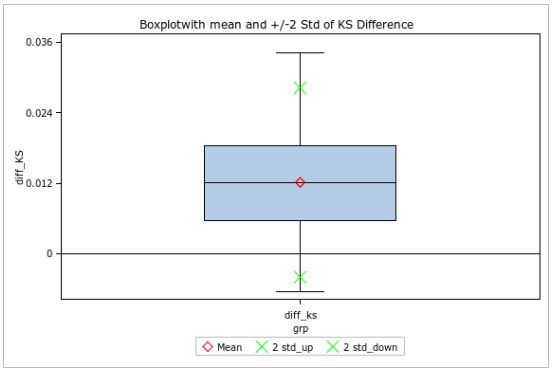


(b) Empirical CDF of Δ^{KS}.

Figure 5.4: Empirical CDF of Gini- and KS differences for LR.



(a) Boxplot of Δ^{Gini}.



(b) Boxplot of Δ^{KS}.

Figure 5.5: Boxplots of Gini- and KS differences for LR.

To perform the tests described in Section 2.4 we use Δ^{Gini} and Δ^{KS}. If we can reject the hypothesis that $\Delta^{\cdot} = 0$, the MiM statistic is significantly higher. The results of these tests are shown in Tables 5.6 and 5.7. We can see that MiM significantly outperforms EM on the Student's t test and the Wilcoxon signed-rank test. However, it is worth noting that the samples are not independent, as they share observations to a great extent.

Table 5.6: Test results for Gini differences.

Test	Symbol	Value	Two-sided p-value
Student's t test	t	10.17	< 0.0001
Wilcoxon signed-rank test	S	2162	< 0.0001

Table 5.7: Test results for KS differences.

Test	Symbol	Value	Two-sided p-value
Student's t test	t	15.05	< 0.0001
Wilcoxon signed-rank test	S	2473	< 0.0001

5.3 XGBoost results

The second part of this thesis resulted in two XGBoost models, a MiM and an EM. We continue this chapter by presenting the results for the two different approaches.

5.3.1 Model-in-Model

For the XGBoost approach we will present the features used, their importance, the hyperparameters, and finally the performance on train and test data. The ad hoc elbow plot is shown in Figure 5.6. We observe that the AUC starts to deteriorate when there are approximately 20 features left. The final 23 features and their importance are shown in Figure 5.7. It follows that the XGBoost MiM shares features with the LR MiM to a great extent. The final hyperparameters are presented in Appendix D. None of the hyperparameters have hit the boundary of their optimization constraints. The ROC curve can be found in Figure 5.11. The performance on the train and test data is observed in Table 5.1. Notably, the performance is massively higher on train vs test. In Figure 5.8 the empirical bootstrap distributions as described in Section 4.4 are presented.

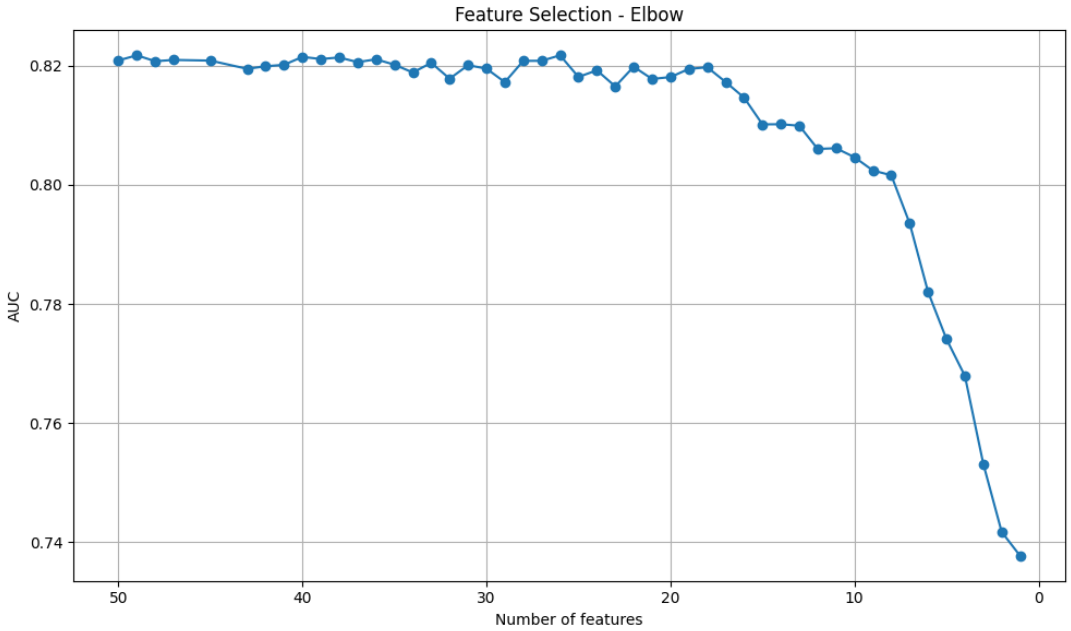


Figure 5.6: Number of features against AUC for XGBoost MiM (backward selection).

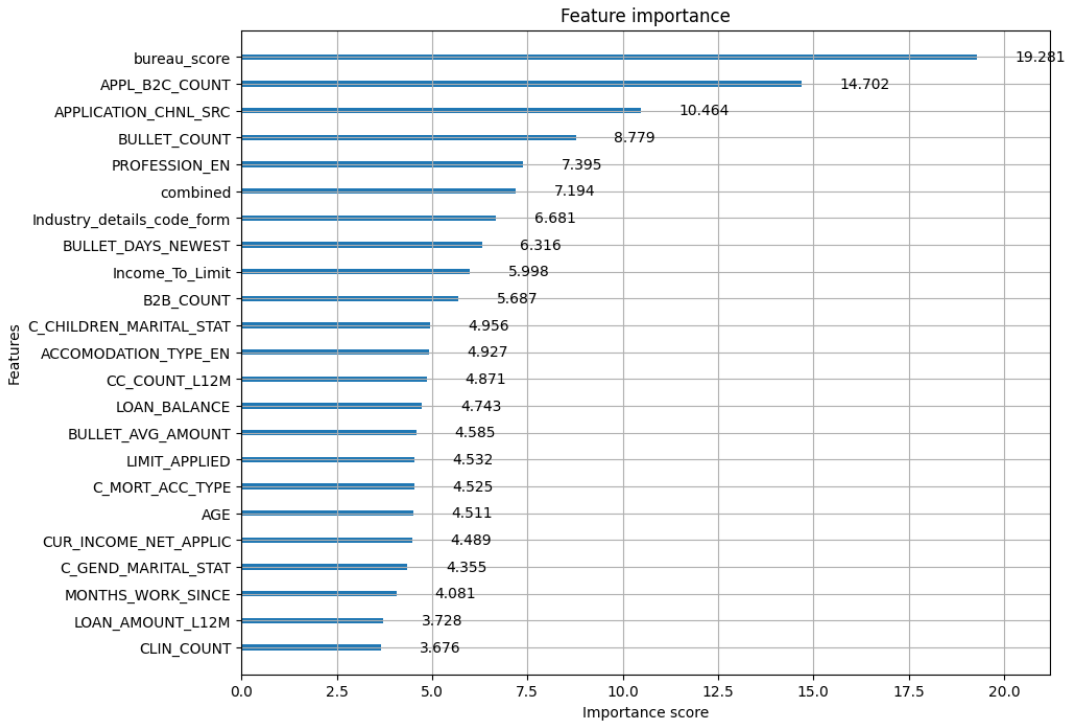
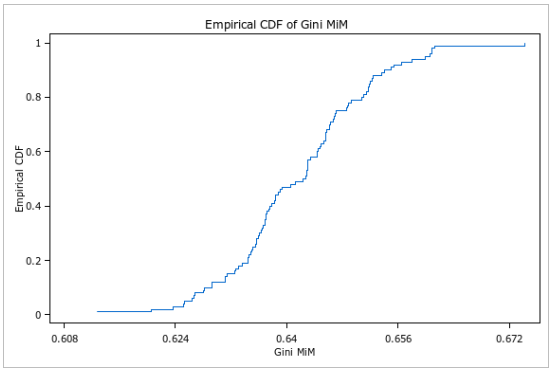
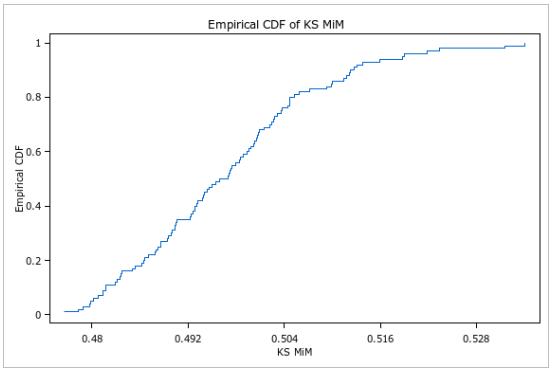


Figure 5.7: Feature gain importance plot for final XGBoost MiM.



(a) Empirical CDF of Gini for MiM.



(b) Empirical CDF of KS for MiM.

Figure 5.8: Empirical CDF of Gini and KS for XGBoost MiM.

5.3.2 Ensemble Model

Analogous to the LR EM, the XGBoost EM consists of three models, an internal model, a bureau score model and an ensemble model.

Internal Model

For the IM we will present which features were selected, their importance, and the hyperparameters. The ad hoc elbow plot is shown in Figure 5.9. Observing this, we notice that AUC starts to deteriorate after approximately 25 left out features. The 26 features selected in the final model together with their importance are shown in Figure 5.10. Note that this model includes four more features than the MiM (if we exclude the bureau score). The hyperparameters are presented in Appendix D.

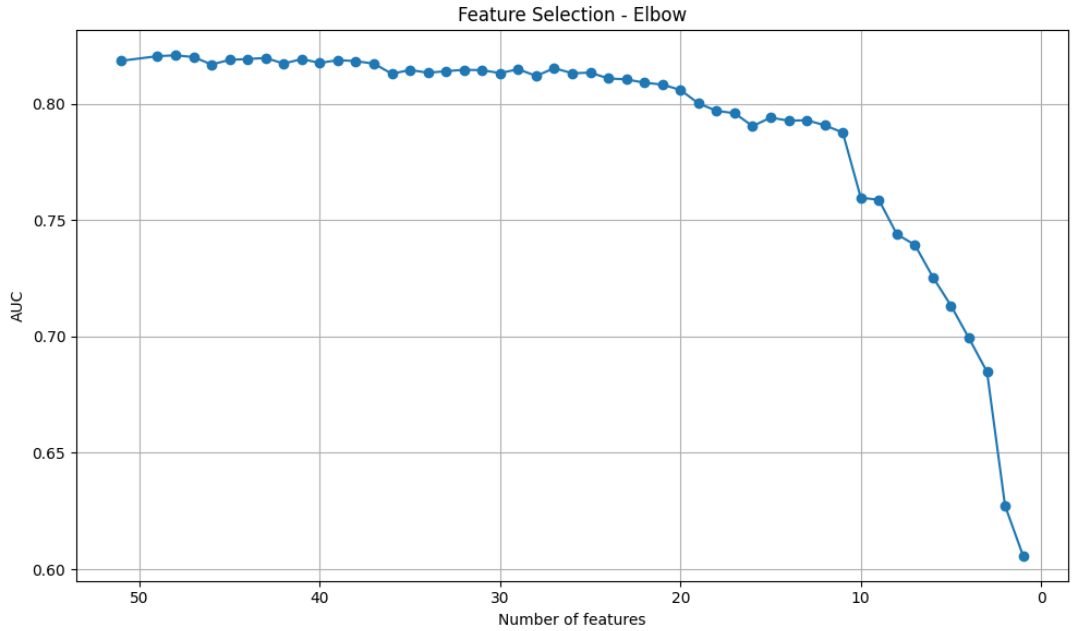


Figure 5.9: Number of features against AUC for XGBoost IM (backward selection).

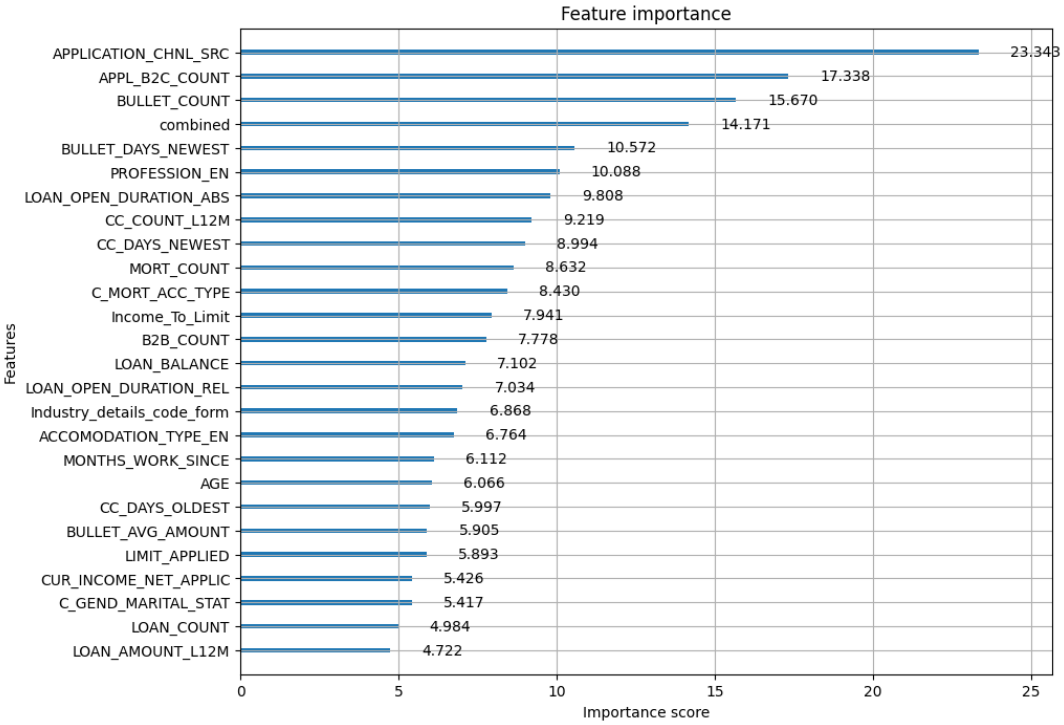


Figure 5.10: Feature gain importance plot for final XGBoost IM.

Bureau score Model

For the BsM, we will present the regression parameters, and their confidence intervals. The odds ratios are presented in Table 5.8. The OR corresponding to bureau score is intuitive, increasing the bureau score by one unit decreases the odds of default with a factor of 0.9984. The bootstrap estimates are found in Appendix C.5.

Table 5.8: Odds ratios with Wald confidence intervals for XGBoost BsM.

Effect	Lower	Point Estimate	Upper
bureau_score	0.9983	0.9984	0.9985
Intercept	205506	499449	1213829

Ensemble Model

Analogously to the BsM, we present the parameters of the EM regression. The parameters and their Wald confidence intervals are presented in Table 5.9. In particular, the estimate for the IM is higher, implying that the IM holds more information. The bootstrap estimates are found in Appendix C.6.

Table 5.9: Odds ratios with Wald confidence intervals for XGBoost EM.

Effect	Lower	Point Estimate	Upper
logodds_model	3.850	4.120	4.409
logodds_bureau	1.240	1.337	1.444
Intercept	2.359	2.712	3.117

Performance

The performance of the EM is presented in Table 5.1. The difference in performance between test and train is smaller than that of the XGBoost MiM. The ROC curve can be observed in Figure 5.11. The EM significantly outperforms the benchmark and performs slightly worse than the MiM.

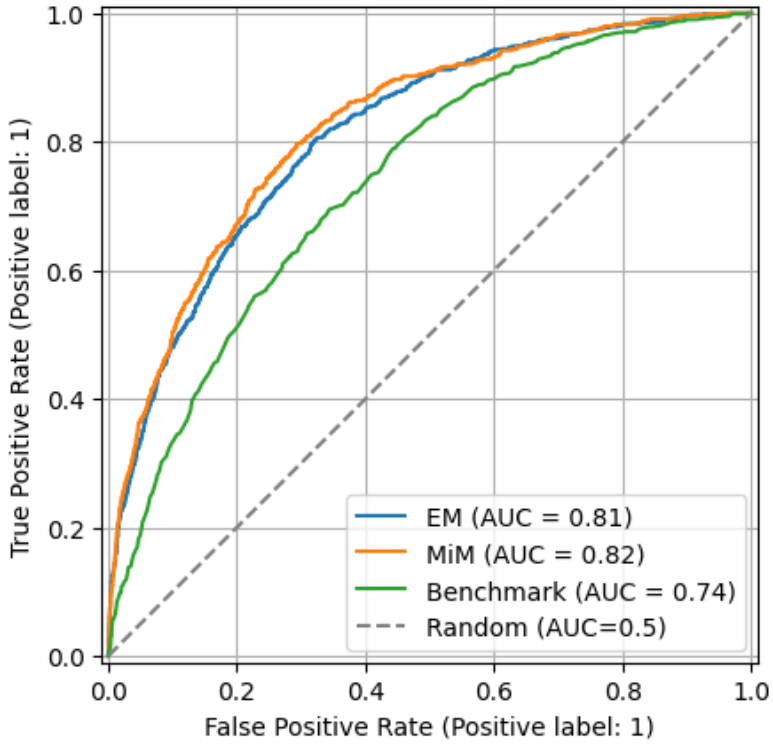
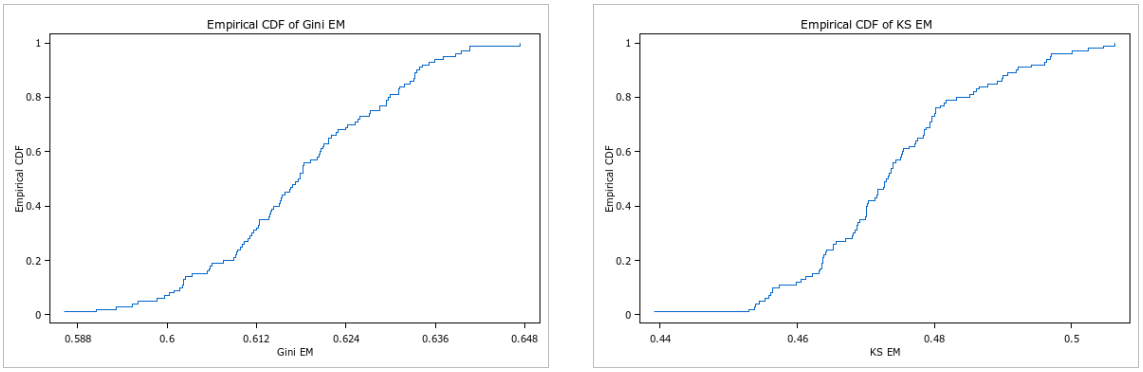


Figure 5.11: ROC-curves for XGBoost models on modelling test data..

In Figure 5.12 the empirical bootstrap distributions are presented, as described in Section 4.4.



(a) Empirical CDF of Gini for EM.

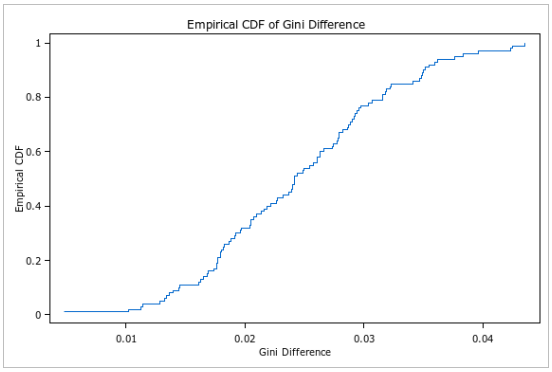
(b) Empirical CDF of KS for EM.

Figure 5.12: Empirical CDF of XGBoost EM Gini and KS.

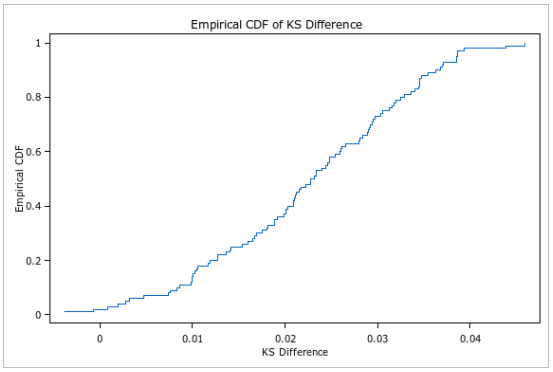
5.3.3 Comparison

To compare the models in detail, the same procedure as described in Section 5.2.3 was followed. In Figure 5.13 the empirical distributions of the differences Δ^{Gini} and Δ^{KS} are shown. In both cases, the body of the distribution is clearly positive, which implies that the MiM performs better. In

Figure 5.14 the box plots of the differences are shown. The differences are clearly skewed to the advantage of the MiM.

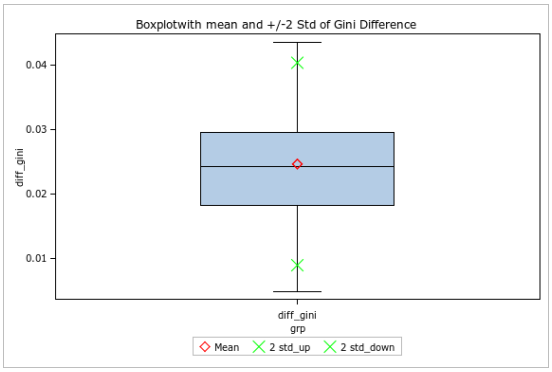


(a) Empirical CDF of Δ^{Gini}.

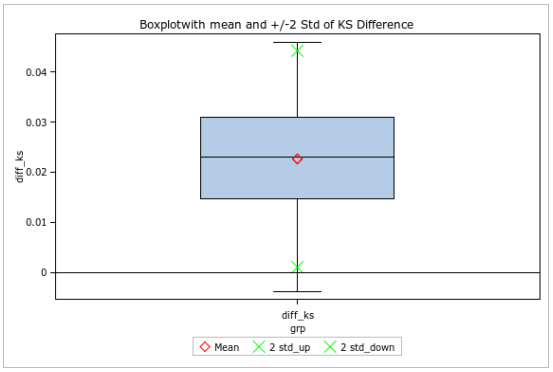


(b) Empirical CDF of Δ^{KS}.

Figure 5.13: Empirical CDF of XGBoost Gini- and KS differences.



(a) Boxplot of Δ^{Gini}.



(b) Boxplot of Δ^{KS}.

Figure 5.14: Boxplots of XGBoost Gini- and KS differences.

Running the tests and observing the results in Table 5.10 and 5.11 we can see that the XGBoost MiM outperforms the XGBoost EM on the Student's t test and Wilcoxon signed-rank test. Recall that the samples are not independent.

Table 5.10: Test results for XGBoost Gini differences.

Test	Symbol	Value	Two-sided p-value
Student's t test	t	31.39	< 0.0001
Wilcoxon signed-rank test	S	2525	< 0.0001

Table 5.11: Test results for XGBoost KS differences.

Test	Symbol	Value	Two-sided p-value
Student's t test	t	20.95	< 0.0001
Wilcoxon signed-rank test	S	2518	< 0.0001

Chapter 6

Discussion

6.1 Model Architecture

6.1.1 Logistic Regression

The two LR models show similar performance, though the MiM has a slight edge. Over the bootstrap samples, the MiM consistently beats the EM. This makes us confident that the MiM approach is favourable on the existing dataset. A sanity check that the original train-test split is representative is that the odds ratios presented in the previous section are generally within the empirical 90% confidence bounds presented in Appendix C. The modest discrepancy in Gini & KS between test and train reinforces this notion.

Examining the features selected in the two models, we observe that both models include the three features with the highest IV in Figure 3.2 and `Income_to_Limit`. However, beyond these variables, the selection of features is more dispersed. This is primarily due to correlation. Consequently, even if a variable contains substantial predictive information on its own, it may contribute marginal additional information if correlated features are already included.

The dummy variable `dum_credit_data` enters both models, the estimated odds ratio is greater than one, implying a positive coefficient β. This indicates that the variable has the opposite effect compared to its marginal relationship with default. A likely explanation is that several other variables in the models already capture information related to `dum_credit_data`. Consequently, this characteristic can be effectively accounted for multiple times, and the coefficient of `dum_credit_data` acts to partially neutralise this overlapping information.

6.1.2 XGBoost

The results of the XGBoost models are clear. The MiM is consistently stronger in terms of bootstrap Gini & KS despite having a modest performance improvement on the original test data. The hyperparameters and the regression coefficients in EM are stable w.r.t. bounds and bootstrap variance. This suggests that the models found are consistent across our sample. However, our models are prone to overfit, as found by observing the discrepancy in performance between test and train. Perhaps better regularisation can be found by further tweaking the hyperparameters. However, the XGBoost test performances are strong compared to that of the LR models. Moreover, there is no goal in itself to perform poorly on train data.

6.1.3 Ensemble Weights

In both EM cases, the ensemble weights are higher for the IM than for the BsM. This suggests that our IM overrides some of the information contained in the bureau score. Thus, the information inside the bureau score may not be fully captured. This could be an explanation for the poorer EM

performance.

6.1.4 Complexity

A non-negligible factor in practice is model complexity. If one model is more expensive to develop than another, this must be compensated for by performance. In both the LR and XGBoost cases, the complexity of EM is higher. Each EM consists of three models, while MiM consists of one. This reinforces the idea that MiM is the superior model architecture.

6.1.5 Collinearity

We suspect that the poorer performance of the EM is due to collinearity. Without knowing the complete structure of the bureau score, the bureau score is likely a mapping of the credit history features. This means that when we build our IM we model information already contained within the bureau score. When we combine the models, we use overlapping information, resulting in poor synergy effects. This is reinforced by the fact that the IM contains more features than the MiM.

6.2 Limitations

6.2.1 Generalization

The models are built upon data from a subset of all customers. If we are to definitely conclude that the MiM approach is advantageous, the set used must be representative of the population, or the results have to be more definite. Consider the LR; the box plots in Figure 5.5 suggest that the EM approach is advantageous on some subsets. Thus, it is not possible to definitely generalise the results. Although this is the case, we have a strong inclination towards the MiM approach in the general case.

6.2.2 Missing Data

As presented in Section 4.1.1 we handle true missing values differently to correctly interpret them in our models. However, initially, we used a numeric interpretation of this missing value (-999). This gave strange interpretations in the WoE bins and XGBoost trees. E.g., grouping -999 and 0 because they are next to one another in the ordered set, despite having drastically different interpretations. Interestingly, the incorrect approach yielded marginally better OOS Gini & KS statistics.

6.2.3 Dependant Bootstrap Samples

The bootstrap Gini & KS statistics are not independent. As bootstrap samples are drawn with replacement from the dataset, there is a significant overlap between the test and train splits used. This means that the bootstrap statistics are not independent. As the Student's t-test and Wilcoxon signed rank test require independent observations, the results need to be interpreted with a grain of salt.

6.2.4 Time Dependancy

As mentioned in Section 3.5, there was a policy shift in the end of 2023. This altered the frequency of defaults and the distribution of the dependent variable. However, it should be noted that after building the existing models, there was no significant information left in the variable `dum_time`. This indicates that the features have the same effect on default, both before and after the lower risk threshold.

6.2.5 WoE

As mentioned in Section 2.2.1, it is industry standard to bin variables using WoE. However, this condenses the information contained in the variables. In a study on German credit registry data, the WoE approach in LR was shown to not improve performance, inducing loss of information

(Seitshiro and Govender, 2024). The binning techniques used in our models resulted in few bins. This means that we heavily condensed the information within variables. It is possible that this loss of information influenced one approach more than the other, affecting the model comparison.

6.2.6 XGBoost Validity

The XGBoost models are high dimensional. We have not rigorously examined the implications of this; e.g., some features may be heavily correlated. The main point of interest was not to build perfect models, but to examine the preferable architecture. This is the motivation for why we did not further investigate alternative XGBoost feature selection methods.

Chapter 7

Conclusion

7.1 Answering formulated problem

We can conclude that the MiM approach, on the dataset concerned, is advantageous to the EM approach as a PD estimator. Performance on the original test split is slightly better, and this pattern is consistently confirmed across the bootstrap splits. However, there are some limitations suggesting that further research may be warranted. E.g., non-representative data and variability in WoE bins.

7.2 Future Research

7.2.1 Reject Inference

Initially, the idea was to further evaluate our models using reject inference. In our model, we have only used accepted customers; this is clearly a biased set of customers (cf. Figure 3.1). To counteract this, one typically uses information from rejected customers to improve robustness (Siddiqi, 2017, p. 215). Unfortunately, this data was not delivered on time by an outside function at the credit institute. This would have been the easiest way to increase the validity of our models.

7.2.2 Alternative WoE approach

In this paper, we use the raw WoE variables in the LR models, estimating one coefficient per feature. Another common approach, commonly used in consumer credit risk, is to use a series of dummy variables, one for each bin. Since this approach is commonly employed, it would clearly be of interest to examine it in more detail.

7.2.3 Behavioural scorecards

As introduced in Section 1.3.3, there are two types of scorecards. Both types are used to an equal extent in industry. Since our study is limited to application scorecards, it would be valuable to investigate the same issue in the context of behavioural scorecards as well.

Appendix A

Variable Description

Variable	Type	Description
APPLICATION_DATE	Date	The date the application was submitted
APPLICATION_CHNL_SRC	String	The source which application was sent through (e.g. online or in store)
AGE	Integer	Age of applicant
GENDER	String	Gender of applicant
LIMIT_APPLIED	Float	The credit limit which the customer has applied for
LIMIT_APPROVED	Float	The credit limit which customer got approved for
bad12	Binary	1 if customer defaults within a year, 0 otherwise
RESIDENT_IN_COUNTRY	Date	Date from which customer have resided in the country where application is concerned
Industry_details_code_form	String	Code depending on industry customer works in
Social_security	String	Flag depending whether or not customer have social security benefits
ACCOMODATION_TYPE	String	Type of living situation, e.g. rented apartment
MARITAL_STATUS	String	Marital status, e.g. single or married
PROFESSION	String	Profession, e.g. manual worker
benchmark_score	Integer	Score from previously deployed model
NO_CHILDREN	Integer	Number of children
MONTHS_WORK_SINCE	Integer	Number of months customer have been employed at current work
CUR_INCOME_NET_APPLIC	Float	Monthly net income
bureau_score	Integer	External score from credit-bureau
B2B_COUNT	Integer	Number of business accounts (giro accounts, investment loans, leasing agreements, credit lines, credit cards, mortgage loans)
GUARANTOR_COUNT	Integer	Number of loans where the customer acts as surety
APPLC_B2C_COUNT	Integer	Number of applications for consumer loans, leasing or credit cards (within last 10 days)
CC_COUNT_OPEN	Integer	Number of open credit cards (revolving cards and charge cards)
CC_COUNT_L12M	Integer	Number of credit cards opened within last 12 months
CC_AMOUNT_OPEN	Float	Total limit of open revolving credit cards (not including charge cards as there is no amount stored at Bureau)
CC_DAYS_OLDEST	Integer	Age (in days) of oldest credit card (at date of latest Bureau request)

Variable	Type	Description
CC_DAYS_NEWEST	Integer	Age (in days) of newest credit card (at date of latest Bureau request)
GI_COUNT_OPEN	Integer	Number of open giro accounts
MORT_COUNT	Integer	Number of mortgage loans
MORT_COUNT_OPEN	Integer	Number of open mortgage loans
MORT_COUNT_CLOSED	Integer	Number of closed mortgage loans
CLIN_COUNT	Integer	Number of overdrafts and credit lines
CLIN_COUNT_OPEN	Integer	Number of open overdrafts and credit lines
CLIN_AVG_AMOUNT	Float	Average amount of overdrafts and credit lines
CLIN_AMOUNT_OPEN	Float	Total limit of open overdrafts and credit lines
LOAN_COUNT	Integer	Number of consumer loans and leasing agreements
LOAN_COUNT_OPEN	Float	Number of open consumer loans and leasing agreements
LOAN_COUNT_CLOSED	Float	Number of closed consumer loans and leasing agreements
LOAN_PERCENT_CLOSED	Percentage	Percentage of consumer loans and leasing agreements that are closed
LOAN_COUNT_L12M	Integer	Number of consumer loans and leasing agreements opened within last 12 months
LOAN_AMOUNT_OPEN	Float	Total amount of open consumer loans and leasing agreements
LOAN_AMOUNT_L12M	Float	Total amount of consumer loans and leasing agreements opened within last 12 months
LOAN_AVG_AMOUNT	Float	Average amount of open consumer loans and leasing agreements
LOAN_AVG_DURATION	Float	Average duration of open consumer loans and leasing agreements
LOAN_SUM_INST_OPEN	Float	Estimated sum of installments of open consumer loans and leasing agreements (estimated installment = amount / duration)
LOAN_BALANCE	Float	Estimated balance of open consumer loans and leasing agreements (= estimated installment * rest duration)
LOAN_DAYS_OLDEST	Integer	Age (in days) of oldest consumer loan or leasing agreement
LOAN_DAYS_NEWEST	Integer	Age (in days) of newest consumer loan or leasing agreement
LOAN_OPEN_DURATION_ABS	Float	Rest duration (in months) of latest consumer loan or leasing agreement
LOAN_OPEN_DURATION_REL	Percentage	Rest duration (in % of total duration) of latest consumer loan or leasing agreement
BULLET_COUNT	Integer	Number of bullet loans (including microloans with 30 to 90 days term)
BULLET_AVG_AMOUNT	Float	Average amount of bullet loans
BULLET_DAYS_NEWEST	Integer	Number of days since/to latest maturity date of bullet loan
C_MORT_ACC_TYPE	String	Combination between mortgage status (do or do not have) and living situation. E.g. no mortgage and rental property.
C_GENDER_MARITAL_STAT_2	String	Gender combined with partner situation e.g. male and single
C_CHILDREN_MARITALL_STAT_2	String	Number of children combined with partner situation e.g. 0 children and single

Appendix B

WoE Details

Variable	Low Bound	High Bound	WoE
BUREAU_SCORE	-999.00	8662.00	-1.483037
	8663.00	9231.00	-0.767626
	9232.00	9455.00	-0.184925
	9456.00	9640.00	0.1917535
	9641.00	9751.00	0.66458
	9752.00	9862.00	1.3354388
	9863.00	9981.00	2.0554828
COMBINED	18-24_Few, 18-24_Many, 18-24_No, 25-31_Ma, 32+_Many		-0.906409
	25-31_Fe, 25-31_No		-0.167107
	32+_Few, 32+_No		0.3908554
APPLICATION_CHNL_SRC	Online		-0.316665
	Store		0.5231507
LOAN_OPEN_DURATION_ABS	-998.00	45.00	0.2917013
	46.00	73.00	-0.257755
	74.00	999999.00	-0.61709
CC_DAYS_NEWEST	-998.00	1480.00	-0.166758
	1481.00	3236.00	0.6675314
	3237.00	13362.00	1.2394014
AGE	18.00	24.00	-0.909363
	25.00	31.00	-0.217084
	32.00	111.00	0.237714
INCOME_TO_LIMIT	-999.00	0.51	-0.412441
	0.51	0.84	0.0154921
	0.84	12.50	0.4399179
LOAN_OPEN_DURATION_REL	-998.00	0.67	0.3158905
	0.67	0.85	-0.098224
	0.85	999999.00	-0.453279
CC_COUNT_L12M	-998.00	-998.00	-0.00355
	0.00	0.00	0.3322461
	1.00	9999999.00	-0.708781
ACCOMODATION_TYPE_EN	Living with parents		-0.750388
	Rented house		-0.410672
	Owner-occupied apartment, Rented apartment		0.0058199
	Owner-occupied house		0.5944606
MONTHS_WORK_SINCE	-999.00	57.00	-0.238589
	58.00	117.00	0.1907

Variable	Low Bound	High Bound	WoE
	118.00	655.00	0.6277436
CC_COUNT_OPEN	-998.00	-998.00	-0.00355
	0.00	2.00	0.2324055
	3.00	9999999.00	-0.934851
LOAN_COUNT_OPEN	-998.00	-998.00	0.1411361
	0.00	0.00	0.5012077
	1.00	1.00	0.1246023
	2.00	4.00	-0.238821
	5.00	9999999.00	-0.81873
LOAN_BALANCE	-998.00	25258.61	0.1796476
	25259.48	9999999.00	-0.600592
BULLET_COUNT	-998.00	-998.00	0.1600352
	1.00	9999999.00	-0.654936
PROFESSION_EN	Apprentice/Trainee, Federal voluntarice service, Homemaker, Self-employed/Freelancer, Student		-0.615745
	Manual worker		-0.188427
	Retired, Salaried employee (non-manual)		0.1912075
	Civil servant		0.9184423
LOAN_COUNT_L12M	-998.00	0.00	0.258746
	1.00	1.00	-0.054655
	2.00	9999999.00	-0.51712
LOAN_AMOUNT_L12M	-998.00	1527.00	0.2158995
	1528.00	42844.00	-0.245826
	42864.00	9999999.00	-0.776145
INDUSTRY_DETAILS_CODE_FORM	A, B, E, F, G, H, I, T		-0.355181
	D, K, L, N, Q, R, S, Z		0.0029889
	C, J, M, O, P, U		0.5248951
LOAN_COUNT	-998.00	3.00	0.1919203
	4.00	7.00	-0.212032
	8.00	9999999.00	-0.683047
MORT_COUNT_OPEN	-999.00	0.00	-0.120249
	1.00	13.00	0.7677176
LOAN_AMOUNT_OPEN	-998.00	39876.00	0.151645
	39880.00	9999999.00	-0.606293
CC_DAYS_OLDEST	-998.00	351.00	-0.033615
	352.00	999.00	-0.440317
	1000.00	1001.00	-0.809088
	1002.00	4624.00	0.1855156
	4625.00	15053.00	0.9273091
CUR_INCOME_NET_APPLIC	-999.00	2096.00	-0.328902
	2097.00	66000.00	0.2146649
LIMIT_APPLIED	800.00	2430.00	0.3841914
	2444.00	7600.00	-0.047145
	7700.00	15000.00	-0.396586
GENDER	M		-0.090014
	F		0.0874904

Appendix C

Ad Hoc Bootstrap

The bootstrap estimates are shown to reinforce the results by confirming stability across sample splits.

Table C.1: 5th percentile, median, and 95th percentile of $e^{\boldsymbol{\beta}}$ for LR MiM.

Variable	5th percentile	Median	95th percentile
WOE_BULLET_COUNT	0.45781	0.49474	0.53450
WOE_CUR_INCOME_NET_APPLIC	0.44950	0.50880	0.56353
WOE_ACCOMODATION_TYPE_EN	0.54767	0.59368	0.65222
WOE_bureau_score	0.43266	0.44612	0.45657
WOE_APPLICATION_CHNL_SRC	0.52581	0.55595	0.58118
WOE_Industry_details_code_form	0.41914	0.45673	0.49284
WOE_PROFESSION_EN	0.40030	0.42800	0.46291
WOE_LOAN_BALANCE	0.46533	0.49854	0.54461
WOE_C_GENDER_MARITAL_STAt_2	0.45567	0.50600	0.56034
WOE_LIMIT_APPLIED	0.37191	0.43771	0.51456
WOE_Income_To_Limit	0.59856	0.67088	0.73676
WOE_combined	0.64080	0.66833	0.70997
dum_credit_data	2.49036	2.95907	3.46024
Intercept	0.05217	0.06120	0.07163

Table C.2: 5th percentile, median, and 95th percentile of $e^{\boldsymbol{\beta}}$ for LR IM.

Variable	5th percentile	Median	95th percentile
WOE_Income_To_Limit	0.62730	0.70345	0.77309
WOE_combined	0.51993	0.54292	0.57515
WOE_APPLICATION_CHNL_SRC	0.43465	0.45710	0.47939
WOE_Industry_details_code_form	0.45310	0.49382	0.53558
WOE_MONTHS_WORK_SINCE	0.55960	0.60088	0.65414
WOE_BULLET_COUNT	0.43738	0.46841	0.50791
WOE_PROFESSION_EN	0.37278	0.40069	0.43680
WOE_CC_COUNT_L12M	0.48494	0.51462	0.55521
WOE_C_GENDER_MARITAL_STAt_2	0.43327	0.48306	0.52953
WOE_CC_DAYS_OLDEST	0.44290	0.49060	0.54172
WOE_ACCOMODATION_TYPE_EN	0.56571	0.60951	0.67091
WOE_MORT_COUNT	0.45995	0.50900	0.55452
WOE_LOAN_BALANCE	0.46843	0.51464	0.57904
WOE_LOAN_OPEN_DURATION_ABS	0.38230	0.41489	0.45248
WOE_CUR_INCOME_NET_APPLIC	0.44794	0.50594	0.56276
WOE_LIMIT_APPLIED	0.33252	0.38356	0.44561
dum_credit_data	11.21720	13.16060	16.19140
Intercept	0.01183	0.01439	0.01686

Table C.3: 5th percentile, median, and 95th percentile of $e^{\boldsymbol{\beta}}$ for LR BsM.

Variable	5th percentile	Median	95th percentile
Intercept	0.17274	0.17471	0.17652
WOE_bureau_score	0.35768	0.36753	0.37562

Table C.4: 5th percentile, median, and 95th percentile of $e^{\boldsymbol{\beta}}$ for LR EM.

Variable	5th percentile	Median	95th percentile
Intercept	1.89704	1.93227	1.96099
log_odds_im	2.13486	2.16265	2.18656
log_odds_bs	1.82211	1.85065	1.87898

Table C.5: 5th percentile, median, and 95th percentile of $e^{\boldsymbol{\beta}}$ for XGBoost BsM.

Variable	5th percentile	Median	95th percentile
Intercept	249026	383359	530280
bureau_score	0.99839	0.99843	0.99848

Table C.6: 5th percentile, median, and 95th percentile of $e^{\boldsymbol{\beta}}$ for XGBoost EM.

Variable	5th percentile	Median	95th percentile
Intercept	2.55291	2.62613	2.74701
log_odds_im	4.06481	4.17752	4.29233
log_odds_bs	1.27902	1.28984	1.32615

Appendix D

Hyperparameters

The hyperparameters presented in Table D.1 are only the parameters that were optimized (cf. Table 2.1). There are several other hyperparameters that were kept to the default settings. For the exact values we refer to the `Scikit-learn` documentation.

Table D.1: Hyperparameters found for XGBoost models.

Parameter	**Internal Model**	**Model-in-Model**
`K`	469	448
η	0.036	0.034
`max_depth`	4	5
γ	1.738	3.311
λ	9.670	5.678
`colsample_bytree`	0.7	0.650
`subsample`	0.730	0.539

Bibliography

Adams, D. (1979). *The Hitchhiker's Guide to the Galaxy*. Heinemann, London.

Agresti, A. (2007). *Introduction to Categorical Data Analysis*. Wiley, Hoboken, NJ, USA, 2nd edition. [Online]. Available: https://ebookcentral.proquest.com/lib/lund/detail.action?docID=290465.

Berg, T., Burg, V., Gombovi, A., and Puri, M. (2020). On the rise of fintechs: Credit scoring using digital footprints. *The Review of Financial Studies*, 33(7):2845–2897.

Bholowalia, P. and Kumar, A. (2014). Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9):17–24.

Blom, G., Enger, J., Englund, G., Grandell, J., and Holst, L. (2017). *Sannolikhetsteori och statistikteori med tillämpningar*. Studentlitteratur AB, Lund.

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, New York, NY, USA. ACM.

Chi, B.-W. and Hsu, C.-C. (2012). A hybrid approach to integrate genetic algorithm into dual scoring model in enhancing the performance of credit scoring model. *Expert Systems with Applications*, 39(3):2650–2661.

Coadou, Y. (2022). Boosted decision trees. In *Artificial Intelligence for High Energy Physics*, chapter 2, pages 9–58. World Scientific.

Frazier, P. I. (2018). A tutorial on Bayesian optimization.

Frykström, N. and Li, J. (2018). IFRS 9 – the new accounting standard for credit loss recognition. Economic Commentary 3, Sveriges Riksbank, Stockholm.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Harris, T. and Hardin, J. W. (2013). Exact Wilcoxon signed-rank and Wilcoxon Mann–Whitney ranksum tests. *The Stata Journal*, 13(2):337–343.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer US.

Massey, Frank J., J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78.

Rawlings, J. O., Pantula, S. G., and Dickey, D. A. (1998). Springer, New York, NY, USA, 2nd edition.

Roberts, T. (2019). Credit scoring approaches guidelines. Technical report, International Committee on Credit Reporting and World Bank Group, Washington, DC.

Rojas, Y. (2023). Status of debtor registration at an enforcement authority and risk of nonfatal suicide attempt. *Crisis*, 44(3):209–215. PMID: 35138185.

Satopää, V. A., Baron, J., Foster, D. P., Mellers, B. A., Tetlock, P. E., and Ungar, L. H. (2014). Combining multiple probability predictions using a simple logit model. *International Journal of Forecasting*, 30(2):344–356.

Scallan, G. (2013). Marginal Kolmogorov-Smirnov analysis: Measuring lack of fit in logistic regression. Presented at the Edinburgh Credit Scoring Conf. [Online]. Available: `https://www.scoreplus.info/assets/files/Marginal-KS-analysis-Measuring-lack-of-fit-in-logistic-regression-Edinburgh-conference-Aug-2013.pdf`.

Seitshiro, M. B. and Govender, S. (2024). Credit risk prediction with and without weights of evidence using quantitative learning models. *Cogent Economics & Finance*, 12(1):2338971.

Siddiqi, N. (2017). John Wiley & Sons, Hoboken, New Jersey, 2nd edition.

Thukral, S., Kovac, S., and Paturu, M. (2023). Chapter 29 - t-test. In Eltorai, A. E., Liu, T., Chand, R., and Kalva, S. P., editors, *Translational Interventional Radiology*, Handbook for Designing and Conducting Clinical, pages 139–143. Academic Press.

Zhu, H., Beling, P. A., and Overstreet, G. A. (2001). A study in the combination of two consumer credit scores. *The Journal of the Operational Research Society*, 52(9):974–980.