Applying Machine Learning to Credit Decision Modelling: A Case Study in Sub-Saharan Africa

Bachelor's Thesis

to obtain the degree of Bachelor of Science (B.Sc.) in the field of Economics at the Faculty of Business and Economics,
Freie Universität Berlin

Berlin, August 27, 2023

Submitted by
Adrian Schmieg

Supervised by
Prof. Dr. Dr. Andreas Löffler
Department of Finance, Accounting & Taxation

Abstract

This paper investigates the potential advantages of using machine learning algorithms to predict decisions on microloan applications at a bank that operates in a least-developed country. A challenger model was constructed using gradient boosting and the results were compared to the current scorecard. The performance and interpretability of both models were evaluated, revealing that the challenger model boasts higher precision, leading to a decreased rate of incorrect rejections. However, the overall performance difference is marginal, and the test sample is too small to draw definitive conclusions regarding precision. Instead, it is proposed that automating approvals rather than rejections may be a more efficient and beneficial solution for both the bank and its clients. The study concludes that machine learning holds significant potential in financial applications beyond credit decision modelling, particularly in less developed countries.

Contents

List of Figures

List of Tables

1 Introduction

This thesis investigates the possible advantages of using machine learning algorithms in modelling the credit decisions of a microfinance bank in a least-developed country in sub-Saharan Africa (hereafter referred to simply as *the bank* or similar).

The assessment of credit risk is an important task for the financial outcome of banks worldwide. At the same time, it has an enormous impact on people's lives and economic opportunities and thus implies not only economic but also ethical questions. Improving risk assessment in terms of efficiency and fairness is thus a worthwhile effort. This project is motivated by the interest in achieving these goals with regards to the credit risk assessment process of a specific bank and loan product. Secondarily, it aims to broaden the understanding of the suitability of machine learning (ML) for such predictions in least-developed countries.

The thesis is structured as follows: section 1 elaborates the motivation for this research project and defines its objectives. Section 2 discusses the choice of algorithms and the overall methodological setup, before section 3 presents the results of the chosen process. We discuss the findings in section 4 and outline some additional considerations. Finally, section 5 concludes the results and learnings.

1.1 Motivation

The bank operates in a market where microloans are among the most popular financial products for individuals and small businesses. Consequently, the bank's portfolio consists predominantly of microloans, the applications for which are collected by field agents and then handled by a team of underwriters. The underwriters are responsible for assessing the credit risk of each loan application and deciding whether to approve or reject it. Even though they are experienced in both back-office and fieldwork, the process is time-consuming, and the bank is looking for ways to increase efficiency. The bank, which wishes to remain unnamed and unlocalized, has kindly provided a dataset of all loan applications. The dataset contains information about the loan application, the applicant, and the decision taken by the underwriter. The features of the dataset are described in more detail in section 2.

Least-developed countries (LDCs) have very different economic challenges from industrialized countries, often featuring a large informal sector while being politically and economically volatile. Economic growth is impeded by weak institutions, low administrative capacities and a lack of infrastructure. In addition, the availability, duration and quality of education are often very low. All of this can result in a higher volatility of the economy as a whole, but also partially in higher resilience of everyday economic activities to economic shocks. This bank in particular has observed a surprising degree of decoupling between macroeconomic impacts and the performance of their clients' businesses. This may be due to the nature of these micro-enterprises often being in everyday goods and services. Nonetheless, economic shocks are more frequent and can cause political instability, which in turn does have a major impact on business activity.

While financial institutions in industrialized countries have access to a wide array of data sources of usually high quality, the same cannot be said for LDCs. Especially microfinancial institutions (MFIs) are affected by this as their clients tend to be among the lower-income population and often do not have a formal credit history. Many MFIs in LDCs therefore have to rely purely on their own data collection efforts. This can result in a lack of data, which is associated with low data quality regarding the information that is relevant for a bank.

Large financial institutions have recently explored machine learning for credit risk assessment and some information on this subject is available, but very little research can be found on the topic for LDCs. This work aims to shed some light on this topic. The regulatory framework in these countries is often more permissive than in industrialized ones, which can make it easier to implement new types of models. At the same time, the low quality and availability of data can make it harder to estimate reliable models. In the case at hand, decisions are often made based on experience and impressions from conversations, photographs of the business, etc. This makes it difficult to formalize the information, rendering large portions of previous research inapplicable. Yet, machine learning can potentially help identify patterns in the data that correspond with indicators that are subconsciously perceived by humans but not typically considered in a model. This is what this study investigates to fill the gap in previous research.

1.2 Theoretical Background

Banks have both an incentive and a regulatory requirement to assess the credit risk of their loan portfolios. The International Financial Reporting Standards (IFRS) have been agreed upon across the industry and contain standards concerning loan losses. The IFRS 9 (cf. Zülch and Hendler 2019) is the current standard for the accounting of financial instruments. It requires banks to estimate the expected credit losses of their loan portfolios to provision them adequately. While the requirement is only to *quantify* the cost of risk, there is an obvious incentive for banks to *minimize* it. According to the standard, the measurement of expected credit losses shall be

- unbiased and probability-weighted, considering several possible outcomes,
- sensitive to the time value of money,
- based on information about past events, current conditions, and forecasts of future economic conditions.

However, banks are not required to factor in every possible outcome. The minimum requirement is to consider the event of default and the event of non-default. A common way to denote this (cf. Pfeiffer 2022; Bowman 2020) is to formulate the expected loss from credit as

$$EL = EAD * PD * LGD, \tag{1}$$

where EL denotes the expected loss, EAD the exposure at default, PD the probability of default, and LGD the loss given default. The expected loss is defined as the expected monetary value lost by a bank on an individual loan i. The EAD is the amount of money that is still owed by the borrower at the time of default. Out of that amount, the LGD denotes the share that is expected to be irrecoverable within a period of 90 days, after which a loan is classified as defaulted. This formula (1) leads to the basic optimization problem

$$\min_{\text{EAD, PD, LGD}} \sum_{i=1}^{n} EL_i = \sum_{i=1}^{n} EAD_i * PD_i * LGD_i, \text{ where n is the number of loans.} \tag{2}$$

Because EAD tends to be proportional to the loan amount, minimizing it would work against the profit maximization goal of a business. Therefore, the aim is to minimize the PD and LGD. The latter can be lowered by implementing recovery measures, such as guarantors and collateral. Meanwhile, the average PD of all loans i in the portfolio can be reduced by only serving customers with a low probability of default. This is leveraged by using a scorecard to determine which clients qualify for a loan.

The bank has found it to be difficult to reliably estimate the PD of new clients because only basic cashflow data and demographic information are available about them. For repeat customers, it has proven useful to use transaction data, but this is not available for new clients. For this reason, their applications have long been manually assessed by underwriters. Typically, a scoring model *(scorecard)* would assess the PD of a loan application, serving as a basis for the decision to approve or reject it. The scorecard is built on a regression model that is trained on historical data. In the case of this bank, the loan product offered via this digital channel is too new to reliably estimate the PD, which is why the bank has explored the use of a scorecard that models the underwriters' decisions as a proxy. It aims to increase the efficiency of the assessment process. This model is currently being tested and will be used as a baseline in this study.

Potentially, the cost of serving a credit client could be lowered by accelerating the application process. Since EAD is relatively low, the bank is willing to accept a higher PD in order to increase efficiency. This could be achieved by automating part of the assessment. However, this would require a model that can predict the PD or the decisions of the analysts with sufficient accuracy. The latter is the objective here. We will establish the precise research objective in section 1.3.

The bank operates under the assumption that the underwriters' decisions reduce the PD when compared to accepting all clients. It is, however, virtually impossible to verify this assumption because the counterfactual cannot be quantified; while we know the default rate among borrowers that have been accepted, we cannot extrapolate the hypothetical default rate among those that have been rejected. We will therefore adopt the assumption for this research project, as it seems reasonable to assume that trained industry professionals are in fact able to identify at least a share of the riskier clients. Under this assumption, a better model of underwriter decisions would lead to a lower PD and thus a lower expected loss, and/or an efficiency gain in the credit assessment process and consequently a lower cost of serving a credit client.

1.3 Objectives

This thesis will evaluate the performance of the scorecard in comparison to a machine learning model. The objective is to develop a model that maintains a basic level of interpretability while enabling an efficiency gain in the loan assessment process. The model should be interpretable in the sense that the decision can be traced back to the input variables. This is important not just to understand whether the model uses reasonable predictors, but also to ensure that it is consistent with the business strategy and non-discriminatory.

We will investigate whether improvements can be achieved while building the model using the same data source and a similar timespan. A suitable algorithm will be determined to build a challenger model for the existing regression-based scorecard. The comparison will reveal the relative performance and each models strengths and weaknesses. At the same time, it will show whether the scorecard could be improved by using other variables or algorithms (as suggested by Folpmers et al. 2022; Bowman 2020).

2 Methodology

This section details the methods used in building the model. We will first describe the data, then the abstract modelling problem, and penultimately the choice of algorithm. Finally, we will evaluate possible evaluation metrics and determine the ones to use in this application. The goal of this chapter is to provide context for the choices made and explain how they were motivated.

2.1 Dataset

The bank kindly provided a dataset of microloans that were handled through their digital loan assessment between February 2022 and June 2023. This channel makes up for the vast majority of microloans, which in turn are the major component of the bank's portfolio. The dataset contains 92,331 rows and 51 columns, which are listed in Table 3. After dropping non-assigned values in the columns *AmountApplied, MaturityApplied, Sector*, and *decision*, there are 61,079 observations left. These columns are the bare minimum that should be filled for all loans that are processed; therefore the remaining ones must have been gathered or stored incorrectly. We will also impose the constraint that the case must not have been rejected while the application was filed, as this usually means that there was a formal error in the application (meaning the variable *RejectOnBot* must be 0). Additionally, the loan amount must be at least 4000 local currency units (LCU), as any amount below is decidedly below the bank's minimum loan amount, and the sector must not be *Consommation*. This requirement has been set by the bank for operational reasons. Furthermore, we will only consider data from July 7, 2022, onward, as the bank changed the application process at that time, and we want to avoid any bias that may be introduced by this. After further data cleaning (see appendix), 23,505 observations remain, which is critically low, but should still be a large enough sample to proceed.

The models were trained on a subsample of the available data, while the remaining observations were used for testing. The data were split based on the application date, with the training sample being all observations from March 15th, 2023, and earlier, while the test sample contains all later applications. This date was chosen to be briefly after the creation of the scorecard that is being challenged. Based on this split, the test sample makes up for 25.5% of the data, therefore matching typical split ratios.

Before moving on to the operational setup, we will briefly establish the optimization problem at hand. Going forward, we will use the variable *rejection*, which is the inversion of the variable *decision* in the dataset. This is the dependent variable used by the scorecard; hence, it will be used here as well. This is the matter of interest for each given application. The dependent variable *rejection* is a random variable that follows a binomial distribution of the form $Y \sim \text{Bin}(n, p)$. The goal is to estimate the decision $y_i \in \{0, 1\}$ for every application i, where 0 is coded as approval and 1 as rejection, based on a vector of independent variables X_i. The independent variables are a mix of cardinal and nominal variables, where the latter were dummy-encoded to produce a vector of discrete variables. Based on the sample of size $n = 23505$, the estimated probability of rejection $P(Y = 1)$ for the binomial distribution is the same as a maximum likelihood estimation (Mitchell 2017), where $I(\cdot)$ is the indicator function:

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{n} I(y_i = 1) = \frac{4157}{23505} \approx 0.1769 \tag{3}$$

2.2 Methods for Feature Selection

To optimize out-of-sample performance, we used a subset of the available variables for our model. There were several options on how to select which features to use. Before all else, we eliminated some unqualified features based on missing values and challenging data types. The characteristics of the variables and the process of dimensionality reduction are described in detail in the appendix.

First, we analyzed the correlation between the independent variables and the target using Pearson's correlation coefficient. The results of the correlation analysis are displayed in the appendix in Table 5 and Figure 6. Most of the selected variables have a correlation coefficient $|r|$ between 0.05 and 0.075, which is unexpectedly low. This may be caused by one of the following reasons:

- It is possible that the variables are indeed not relevant to the decision-making process. This would result in a non-predictive model. However, since a model was successfully built previously, we consider this highly unlikely.
- The interaction between the variables and the target is in fact non-linear.

We suspect non-linearities in the data in two manifestations. First, some financial variables and other large-scale continuous variables are better depicted as an exponential or logarithmic scaling (cf. Binsbergen, Han, and Lopez-Lira 2023; Friedman 2001). Second, due to the data mostly being verbally inquired from the clients, we expect to observe somewhat inaccurate values. Rounded values are much more prevalent than a hypothetical underlying distribution would suggest, and some clients might be inclined to manipulate information to skew the decision in their favor. To this end, we determine that tree-based models may be better-suited to capture such non-linearities.

Due to the lack of results from the correlation analysis, we further conducted a univariate regression analysis, the results of which are also shown in the appendix in Figure 7, Figure 8, and Figure 9. The results did not yield a clear result either but granted some insight into the relationship between the variables and the target nonetheless. Many of them in fact seem to have non-linear relationships with the target. This kind of dependency can barely be captured by correlation coefficients but can be recognized by decision trees. This is sometimes done in the form of tree-based feature selection. That is not required here, as the later model will be tree-based anyway. [1]

[1] Another popular method is Principal Component Analysis (PCA), which is a dimensionality reduction technique that transforms the data into a new set of variables that are combinations of the original variables. The new variables are ordered by the amount of variance they explain in the data, then the first few variables are selected as the model features. However, it is difficult to reconstruct how a change in a particular variable influences the prediction, therefore making the model hard to interpret. There are ways to interpret PCA models, and other methods do exist, such as Recursive Feature Elimination, used by Granström and Abrahamsson (2019). These are, however, highly complex to implement and interpret, and are therefore outside the scope of this paper.

2.3 Machine Learning

Machine learning, in the sense of this thesis, describes computer-run models that are estimated based on a dataset and can then be used to predict a response variable for new observations. However, the term can be used in a much broader sense and is now often used interchangeably with Artificial Intelligence (AI). To clarify the scope of this thesis, we will define the term machine learning and its domains as follows.

Definition Machine learning describes ”automatic computing procedures [...] that learn a task from a series of examples” (Michie, Spiegelhalter, and Taylor 1999). According to the authors, ”attention has focussed on decision-tree approaches, in which classification results from a sequence of logical steps”. This will be covered in section 2.5. The author further distinguishes between *supervised* and *unsupervised* learning. Combinations and melange procedures also exist, such as *reinforcement learning*. In supervised learning, the algorithm is given a set of examples, including predictors and a response variable, and is tasked with modelling the response variable (also called *label* in machine learning).

Some models can be trained either as supervised or unsupervised models, e.g., Artificial Neural Networks (ANNs). ANNs process data points using several different methods, and the number of so-called *nodes*, i.e. decision points, is very high. This can make such models highly predictive, but the downside is that it renders the model non-interpretable and can lead to overfitting. For this reason, some fields, such as finance, refrain from using these kinds of algorithms, as their choices usually have to be explainable in retrospect (Folpmers et al. 2022). Hence, we will focus on supervised learning models in this thesis.

Class Imbalance When datasets contain an uneven number of observations from the different classes of the target variable, they can be rebalanced by either undersampling the majority class, oversampling the minority class, or a combination of both. Synthetic Minority Oversampling Technique (SMOTE) is a popular oversampling algorithm that creates synthetic observations by interpolating between existing observations using a nearest-neighbor approach (Chawla et al. 2002). SMOTE has been found to enable performance gains in many applications, yet it has also been identified to be useless in others (Granström and Abrahamsson 2019). With the minority class of our dataset making up 18% of observations, rebalancing may or may not be beneficial. We will therefore compare the performance of models trained on the original dataset with models trained on a rebalanced dataset using SMOTE.

2.4 Similar Applications

Due to machine learning being a rather recent development, research is still catching up on its risks and benefits in various fields. For this reason, banks have been skeptical of its adaptation. Some publications on the application of machine learning in the field of credit risk assessment do exist, but they mostly cover basic, high-quality datasets with a common structure, originating from US and European banks. Presumably, internal research in large financial institutions is more advanced than this, but this is rarely available to the public.

Applications of machine learning to PD models have previously been investigated, among others in undergraduate and graduate theses in the fields of statistics and mathematics. As such, Granström and Abrahamsson (2019) compare a wide array of algorithms, each with and without class rebalancing and using different specifications for feature engineering. They find that eXtreme Gradient Boosting (XGBoost) tends to outperform other algorithms, including logistic regression and decision trees. Machado and Holmer (2022) compared XGBoost to the newly developed Categorical Boosting Algorithm (CatBoost) and found that the latter offers even greater performance at lower computational cost, increasing machine learning algorithms' edge over logistic regression. However, opposite results were observed on datasets without categorical variables.

Digression: Choice Modelling
The field of choice modelling focuses on modelling the choices of individuals, in most cases consumers. While the fields of machine learning and choice modelling have much in common, they have developed largely independently from each other. Machine learning originates mostly in the field of computer science, while choice modelling has its roots in economics and statistics. According to Cranenburgh et al. (2022), this results in these issues: since there has been little interaction between the two fields for a long time, the fields have each developed their own terminology and software, making it more difficult for researchers from one field to follow publications from the other. Furthermore, there would also be misconceptions about machine learning in the choice modelling community, resulting in a lack of recognition for the advantages of the approach. In turn, it can be assumed that the machine learning community has not yet incorporated recent advantages in statistical modelling.

2.5 Choice of Algorithm

The choice of algorithm is a crucial step in the process of building a machine learning model, as vastly different results and structures can be achieved. The algorithm for this application should be performant, but interpretable, to enable adequate supervision.

While statisticians and computer scientists have developed numerous different algorithms, a few have established themselves as quasi-standards. Prominently, the open-source package *Scikit-learn* bundles some of the most used machine learning algorithms in an easy-to-use and harmonized fashion. It was introduced by Pedregosa et al. (2011). We will use *Scikit-learn* because it contains most of the algorithms that were used in similar applications and is straightforward to use.

Bussmann et al. (2021) consider regulatory and ethical boundaries for the application of AI in the field of credit risk assessment. They address three criteria for explainable artificial intelligence that were proposed by the European Commission's *High-Level Expert Group on AI* in 2019. Even though these criteria aim toward highly complex, autonomous systems, they provide a useful perspective for simpler machine learning tasks as well. Hence, we try to adhere to them in our choice of algorithm. The criteria are described by Bussmann et al. as follows:

1. **Human agency and oversight**
   Decisions must be informed, and there must be human-in-the-loop oversight.

2. **Transparency**
   AI systems and their decisions should be explained in a manner adapted to the concerned stakeholder. Humans need to be aware that they are interacting with an AI system.

3. **Accountability**
   AI systems should invoke mechanisms to ensure their accountability and auditability, especially in terms of their algorithms, data, and design processes.

None of these criteria contradict the use of machine learning and AI per se, but it is often suggested to choose models that balance accuracy with explainability over more predictive ones, e.g. by Murdoch et al. (as cited in Bussmann et al. 2021: p.207). The following paragraphs will discuss the choice of algorithm in the context of these criteria.

Decision Trees Arguably one of the simplest machine learning algorithms is the decision tree algorithm. It is a nonparametric algorithm, meaning that it does not make any assumptions about the underlying distribution of the data. Instead, it builds a tree-like structure, where each node represents a binary decision based on one variable. The tree is built iteratively, with each step splitting the data into two subsets based on the value of the respective variable. The process is repeated until the tree reaches the specified depth or until there are not enough observations to split the data any further. The final nodes contain the predictions of the target variable. Figure 1 shows an example of a decision tree for the data at hand.

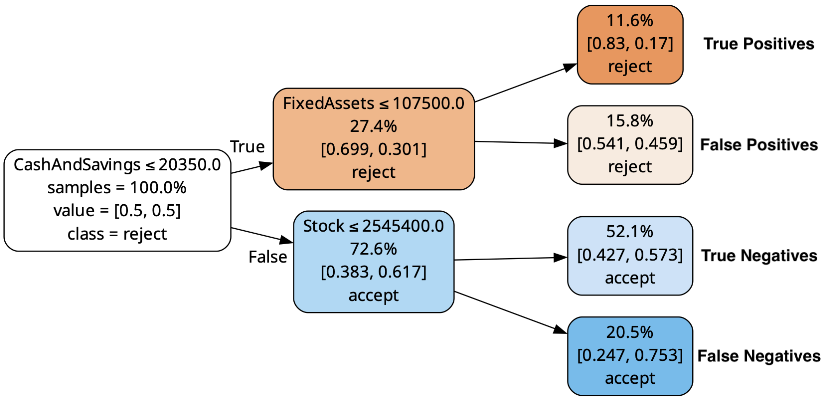

Figure 1: Exemplary Decision Tree.

The first split is based on the variable *CashAndSavings*, the second split then depends on the leaf (i.e., the decision at the first split) and is either based on *FixedAssets* or *Stock*. The *value* arrays contain information about how many observations of each class were assigned to each node.

The mathematical workings of the decision tree algorithm are as follows (Scikit-learn contributors 2023). The criterion that determines whether a split is made is called *impurity*. The default loss function is the Gini impurity, which will be used here as well. Let our dataset be represented by $\mathcal{D}$, each potential split at a node be θ, and the loss function be $L(\cdot)$.

Then the impurity of a node m is defined as

$$G(\mathcal{D}_m, \theta) = \frac{n_m^{true}}{n_m} L\big(\mathcal{D}_m^{true}(\theta)\big) + \frac{n_m^{false}}{n_m} L\big(\mathcal{D}_m^{false}(\theta)\big), \tag{4}$$

where $\mathcal{D}_m^{true}$ and $\mathcal{D}_m^{false}$ are the subsets of $\mathcal{D}_m$ in response to the respective split criterion θ. The impurity of the split is minimized to solve for the optimal θ:

$$\theta^* = \underset{\theta}{\arg\min}\; G(\mathcal{D}_m, \theta), \tag{5}$$

and the process is repeated for each node m until a stopping criterion is reached.

Unfortunately, decision trees can be prone to underfitting and high variance, as they cannot always capture the complexity of the causality behind a dataset (Friedman 2001). This can be mitigated by using ensemble methods, one of which will be covered in the following paragraph. Ensemble models are built upon classical machine learning models and combine multiple *weak* classifiers or regressors into one estimator. This is done by either averaging the results of several estimators, e.g. with random forests, or by *boosting* one estimator with another, e.g. with gradient boosting. Friedman suggest boosting as a means to overcome this issue. This will be discussed in the following paragraph.

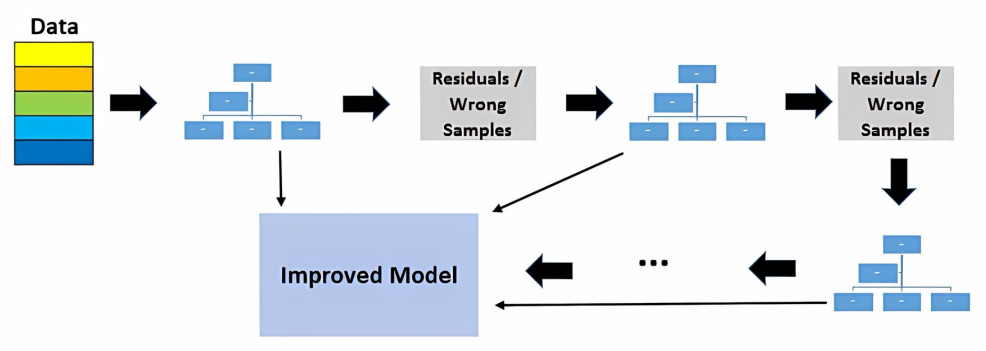

Figure 2: Boosting Process for Decision Trees.
Source: Yıldırım (2020)

Gradient Boosting Gradient boosting is a supervised learning technique that can be used for regression and classification tasks and builds upon the concept of decision trees. The term *boosting* describes using an ensemble of weak prediction models, in this case decision trees, and iteratively improving them. This is done by adding a new tree to the ensemble that is fitted to the residuals of the previous iteration. This process is visualized in Figure 2. Depending on the data, this sometimes allows gradient boosting to achieve severe performance gains over decision trees.

Gradient Boosting Classifier (GBC) is a variant of the *TreeBoost* algorithm introduced by Friedman (2001). Another popular implementation building upon this is *XGBoost*. Frank, Gao, and Yang (2023) found that these two newer implementations yield similar results while using a dataset that is structurally similar to ours. For these reasons, we chose to use *GBC* from *Scikit-learn* (Pedregosa et al. 2011) for the application at hand.

In formal terms, the objective function of the Gradient Boosting Classifier is defined as follows, where h_k depicts the individual trees in the model, and K is the number of trees:

$$\hat{y}_i = F_K(x_i) = \sum_{k=1}^{K} h_k(x_i) \tag{6}$$

In the case of boosting, in contrast to bagging, each tree h_k is fitted to the residuals of the previous tree h_{k-1} at the learning rate γ (see Friedman 2001):

$$h_k = \arg\min_h \sum_{i=1}^{n} L\Big(y_i, F_{k-1}(x_i) + \gamma h(x_i)\Big) \tag{7}$$

Finally, in the default case of log-loss, the regressor is mapped to a binary prediction using the sigmoid function:

$$p(y_i = 1) = \sigma\big(F_M(x_i)\big) \tag{8}$$

Intuition behind the choice of algorithm We chose to pursue tree-based algorithms because of the following hypothesis. A tree-based model should be able to reflect the decision-making process of the analysts more accurately than a parametric regression model, as its mechanics are closer to the human thought process. A human will likely not look at all variables at once and assign a weight to each one of them but rather iterate over the variables, taking microdecisions at each indicator, influencing what they will look for next. This is similar to the way a decision tree is built. Furthermore, PD models and the like are required to be interpretable, which is a strength of tree-based models in contrast to other machine learning algorithms. To this end, partial dependence plots and variable importance measures can be computed and provide a fairly intuitive way to understand the model's workings (Friedman 2001).

2.6 Hyperparameters

Hyperparameters define a model's structure, while the model parameters are learned from the data. Therefore, they influence how a model behaves during training (or in statistical terms, *estimation*) and how it performs on unseen data. For example, the depth of a decision tree is a hyperparameter, while the split criteria θ are model parameters. A lower predefined depth will lead to a smaller tree, which in turn will have fewer parameters and likely lead to lower in-sample accuracy. On the other hand, a too large tree will likely overfit to the data and lead to lower out-of-sample accuracy. The optimal values depend on the dataset and cannot be universally determined. However, some algorithms are less sensitive to changing hyperparameters than others.

A very popular approach to find the optimal hyperparameters is to use grid search. This is done by defining a set of possible values for each hyperparameter and then iterating over all possible combinations. The optimal combination is then selected based on a predefined metric. This is a very time and resource-intensive process, as the model has to be retrained for each combination. In the case of a monotonous relationship between hyperparameters and performance, we felt that a more efficient approach could be a Gaussian interval-halving optimization process. As an implementation of this, we found *HalvingGridSearchCV* in *Scikit-learn*. However, as it is an experimental release as of now, it did not work for this dataset. We proceeded with *Scikit-learn*'s *GridSearchCV* instead, using a set of fairly standard values. These are listed in the appendix under Hyperparameter Tuning.

As an extra step to reduce the risk of overfitting, we use cross-validation (CV). This means that the data are split into k subsets (*folds*); in this case, $k = 3$. Then, the model is trained on $k - 1$ folds and tested on the remaining one. This is repeated k times, each time using a different fold for testing. The average performance across all folds is then used as the performance metric for hyperparameter selection. k-folding is a vastly computationally expensive process, as the model has to be trained k times multiplied by the number of hyperparameter combinations. Therefore, we deviated from the standard value of $k = 5$ and used only 3 instead. This should still help to reduce overfitting but is less computationally expensive.

2.7 Metrics

Depending on which metric is chosen to judge model performance, different models may be preferred. This is particularly true for imbalanced datasets such as the one at hand, where one class is more frequent than the other. Popular metrics in the area of machine learning include:

> **Accuracy**: The ratio of correctly predicted observations, known as True Positives (TPs), to the total number of observations. This metric is not suitable for imbalanced datasets, as it will be high even if the model always predicts the majority class.
>
> **Precision**: The ratio of correctly predicted positive observations to the total number of predicted positive observations. This metric is partially suitable for imbalanced datasets, as it will be low if the model only predicts the majority class. However, it neglects how many of the negative observations are predicted correctly.
>
> **Sensitivity**: The ratio of correctly predicted positive observations to the total number of actual positive observations. This metric is also known as the true positive rate (TPR) and is the same as recall. Its counterpart is the false positive rate (FPR) or type I error rate.
>
> **Specificity**: The share of correct negative predictions to the total number of actual negative observations. It is also called true negative rate (TNR), opposing to the false negative rate (FNR) or type II error rate.

Evidently, a model that is optimized for accuracy, for example, may not be the best choice in terms of precision and vice versa. Hence, the following metrics are also used, and provide a more holistic view of the model's performance:

> **Area under the curve (AUC)**: The share of observations that are in the area covered by the Receiver Operating Characteristic (ROC) curve. The ROC curve is a plot of the TP rate against the False Positive (FP) rate over different thresholds. It is popular for binary classification problems, as it indicates the performance independently of the threshold. It provides a good overview of the model's performance but neglects inhomogeneity across the ROC curve, i.e. its shape.
>
> **Mean Squared Error**: The average of the squared differences between the predicted and actual values. This metric is very popular and has the advantage that during the regression process of gradient boosting, squaring the errors heavily penalizes predictions of the wrong class. The downfall is that a large number of

correct predictions of the majority class may outweigh incorrect predictions of the minority class nonetheless. A model that does not represent the minority class adequately may still be preferred in this case.

While accuracy and precision are easily interpretable, they are not suitable as the sole criteria because of their respective drawbacks. Internally, GBC uses the mean squared error (MSE) as the loss function by default, and we saw no reason to deviate from this. It is also used for regression and classification tasks in some applications, e.g. by Frank, Gao, and Yang (2023). We will compute the AUC in addition and incorporate accuracy, precision, sensitivity (TPR), and specificity (TNR) for model selection and evaluation. Notably, the challenger model's performance will also be judged in comparison to the existing scorecard. In the best-case scenario of increased correct predictions across both classes, the challenger model would be preferable by all metrics.

3 Results

We have estimated several models according to the procedure established in the previous chapter. The results are presented in this section. We will use the metrics determined in the previous section for evaluation. We first estimated a pair of models using all variables determined in section 2. The model was estimated using both the original dataset and a rebalanced dataset using SMOTE. They are labeled GBC_{full} and $\text{GBC}_{\text{full}}^{\text{SMOTE}}$ respectively. The hyperparameters for each model were tuned independently. Based on the feature importances shown in the appendix, we decided to re-estimate the models while leaving out the less influential variables. The metrics for the resulting four models are displayed in Table 1.

Table 1: Comparison of Metrics between Gradient Boosting Models.

	GBC_{full}	$\text{GBC}_{\text{full}}^{\text{SMOTE}}$	$\text{GBC}_{\text{reduced}}$	$\text{GBC}_{\text{reduced}}^{\text{SMOTE}}$
ROC AUC	0.5888	0.5903	0.5832	**0.5992**
MSE	**0.2525**	0.2555	0.2559	0.2571
Accuracy	**0.7475**	0.7445	0.7441	0.7429
Precision	**0.9565**	0.8624	0.9558	0.7620
Sensitivity	0.1813	0.1943	0.1699	**0.2300**
Specificity	0.9964	0.9864	**0.9965**	0.9684

Bold values indicate the best performance by each metric, respectively.

According to the AUC, both class-rebalanced models perform slightly better than the models trained on imbalanced data. The difference in sensitivity is even more pronounced. However, the advantages end there. Specificity is slightly lower for the SMOTE-rebalanced models, and precision is drastically worse. The latter is particularly relevant, as we will elaborate later. While it could be assumed that the models are, in fact, slightly more predictive overall, there is reason to believe that the metrics are deceptive. The relative feature importances indicate that the SMOTE-rebalanced models assign far more weight especially to the variable *DaysOfSale*, and far less to *SalesMin*. However, the partial dependence plots (shown in Figure 3) suggest that the SMOTE-rebalanced model captures amplified random noise rather than a meaningful relationship.

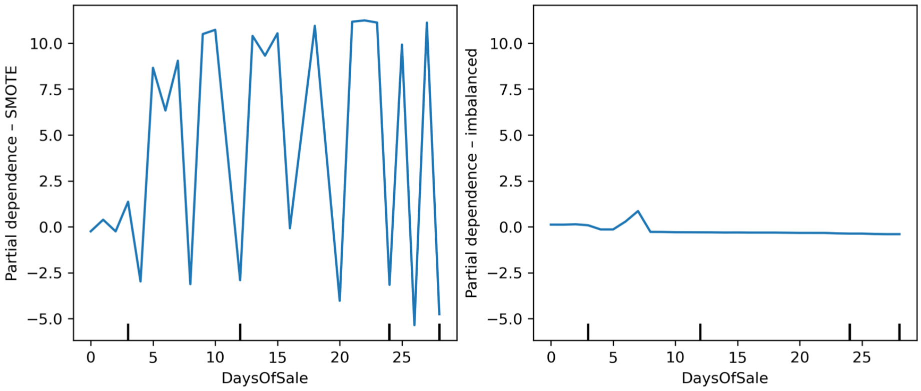

Figure 3: Partial dependence plots for the variable *DaysOfSale*.

Left: partial dependence of the rebalanced reduced model, $\text{GBC}^{\text{SMOTE}}_{\text{reduced}}$.
Right: partial dependence of the $\text{GBC}_{\text{reduced}}$ model.

The results of the hyperparameter grid search revealed that the optimal maximum depth for the individual trees in the rebalanced models is 20, while it is only 5 for the imbalanced models. This leads to the suspicion that these models are overfitted and therefore unsuitable for new data, even though they were constructed using cross-validation. This suspicion is supported by the first trees of the models (see Figure 10 in the appendix). A visual inspection yields that they are tremendously more complex than their imbalanced counterparts. To confirm that the models are indeed overspecified, further testing on a holdout sample or further investigation of the performance on the training and testing data would be helpful. We will disregard the models here, both for the above-mentioned reason and because they are possibly less suitable for the application regardless (see section 4).

Among the imbalanced models, the respective full model performs slightly better than the reduced model by every metric. The difference is most pronounced in precision and sensitivity; however, the differences seem unsubstantial overall. We therefore chose the more parsimonious $\text{GBC}_{\text{reduced}}$ model as the challenger because it is easier to interpret and less likely to be overfitted. Table 2 shows the metrics of the reduced model and the scorecard. Comparing the two, it is obvious that the scorecard outperforms the reduced model on sensitivity and, by a narrow margin, on AUC. While the full model is on par with the scorecard in terms of AUC, it is still behind regarding sensitivity. Machado and Holmer (2022) highlight the need to evaluate all groups of predictions, i.e. the confusion matrix of a model. It is displayed for the scorecard and the reduced model in Figure 4.

Table 2: Comparison of Metrics between Scorecard and Gradient Boosting Models.

	Scorecard	$\text{GBC}_{\text{reduced}}$	GBC_{full}
ROC AUC	**0.5916**	0.5832	0.5888
MSE	0.2531	0.2559	**0.2525**
Accuracy	0.7469	0.7441	**0.7475**
Precision	0.8993	0.9558	**0.9565**
Sensitivity	**0.1927**	0.1699	0.1813
Specificity	0.9905	**0.9965**	0.9964

Bold values indicate the best performance by each metric, respectively.

Although the overall performance is similar, the scorecard is superior in predicting the minority class, which is of interest here. Considering that the scorecard's purpose is to predict rejections, the second most important goal must be to avoid type I errors, i.e., wrong rejections. The challenger model more than halves these cases (4.4% as opposed to 10% of predictions), which means that far fewer clients would be rejected by mistake. While this is a massive improvement, it is important to note that the number of observations in this *false positive* class is very low at 55 for the scorecard and 20 for the reduced GBC model.

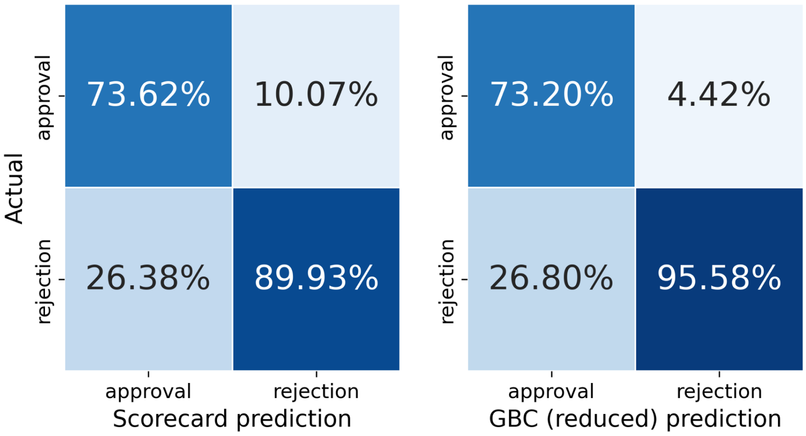

Figure 4: Confusion matrices for the scorecard and challenger model.

Percentages reflect the share of the predicted class.

4 Discussion

4.1 Non-Linearity

We pursued the use of machine learning to reflect non-linear behavior of decision-makers in response to the independent variables. Based on the partial dependence plots shown in Figure 5, the model seems to capture these non-linearities well, seemingly without major overspecifications. Values that may be unrealistic have been adequately captured. For example, the peak in the partial dependence plot for *Stock* shows that applications with *Stock* = 10 000 000 would be more likely to be rejected. This is plausible, as the underwriter may assume that these figures are not based on accounting, but rather a spontaneous estimation. The same may be true for the variable *DaysOfSale*. Having exactly seven days of sale is uncommon in this market, which is why it stands to reason that this particular value is often indicative of unprecise statements in the application. Furthermore, the overall trend of higher values having a lower probability of rejection is very plausible.

Meanwhile, the partial dependence on *SalesMin* is less intuitive to interpret. This is problematic because it is the most important variable in the model. While the overall trend is plausible, the fluctuations appear to be random. This may, however, also be due to codependence of the variable with other variables, which is not captured by the partial dependence plots. A further investigation of this would be beneficial but is beyond the scope of this thesis. Too detailed fluctuations of the partial dependence could be remedied by increasing the minimum number of samples per leaf and decreasing the maximum depth of the trees. It remains to be seen how this would influence the performance of the model.

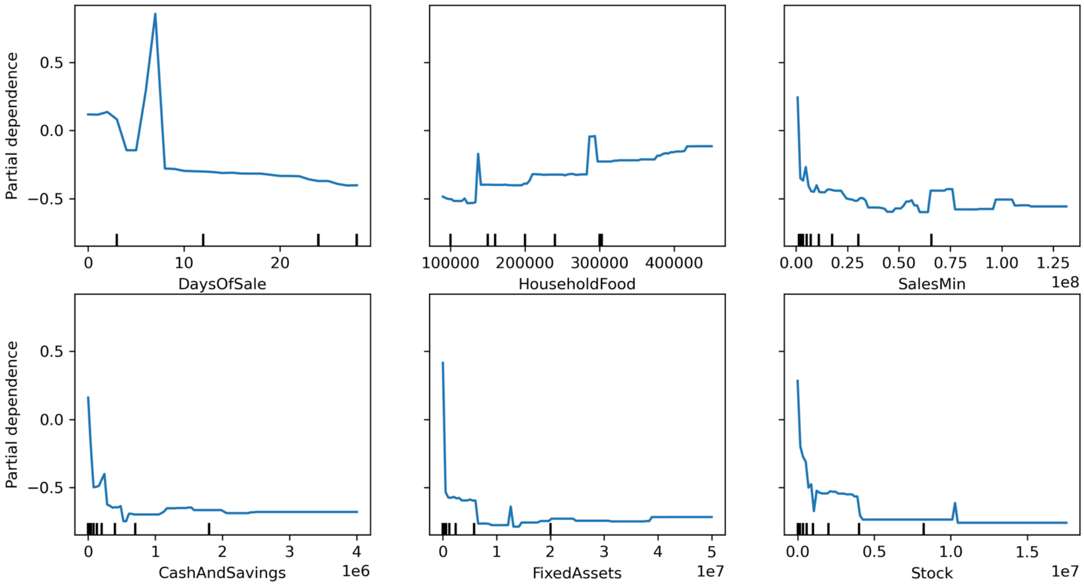

Figure 5: Partial dependence plots for the challenger model $\text{GBC}_{\text{reduced}}$.

4.2 Model Performance

The metrics of the challenger model yield mixed results, yet a conclusion can be drawn from an application-oriented point of view: while gradient boosting does not necessarily result in a more predictive model overall, the improvement achieved in lowering false positives is promising. However, the number of observations in the group of false rejections ended up being very low, which is why the results should be interpreted with caution. While it is clear that both the scorecard and the gradient boosting model perform reasonably well, the subsample of false positives is too small to reject the null hypothesis that the challenger model is no more performant than the scorecard.[2]

This raises some questions:

1. **Are the wrong rejections actually desirable clients?** With this model, we have been targeting the rejection decision of the underwriters. However, while still assuming that the decisions are adequate overall, it is possible that the decision labels are misleading. The clients that fall into the group of FPs could have been approved due to faulty data. It has been observed by the bank that applications sometimes feature inaccurate or misleading data that are recognized by the underwriter. For procedural reasons, they are not always corrected on the application. Hence, the application data do not always reflect the reality and the consequent logical approval/rejection decision. Therefore, some of the FP clients may not actually be desirable by the decision criteria, even when the corresponding decision label indicates otherwise. This is not easily clarified, as it would require a change in the data collection process. Yet, it would allow for much better insights.

2. **Could a model of this kind be implemented to enhance loan application processing?** While there certainly are drawbacks to using machine learning for this task (as highlighted in section 2.5), we have also seen a significant benefit. Presumably, a model that is targeted toward a specific task and carefully trained only on applications that are eligible for automated processing could perform well in a production environment. In the country of operation of this particular bank, the legislation does not impose restrictions relating to this, although neither the practice of the regulatory authority nor their past decisions are known to us.

[2] Testing the results for significance would allow for more robust conclusions. One possibility to determine whether the difference in predictive power is significant would be to compute the metrics for each models predictions for m randomly drawn test samples. This would yield a distribution of each metric. Then, using a Kolmogorov-Smirnov test or a t-test, the difference in means between the alternate models could be tested for significance. However, this would require a larger sample size than the one available to us and is therefore not feasible. The Kolmogorov-Smirnov-test would also require the assumption that the distributions are continuous, which is not the case for the metrics in question. However, Walsh (1963) showed that the test results present a conservative lower bound for the significance level even when the distributions are discrete, as in this case. This is beneficial to future research on this topic.

3. **Is the efficiency gain worth the cost of implementing a new model?** If a model were to perform similarly to these results in a production environment, the performance gain could justify the cost of implementation. However, models often perform better during testing than in production, and the cost of implementation is not negligible. Therefore, such a model would have to be tested more extensively on unfiltered data and optimized to further minimize the risk of overfitting in order to be considered for implementation. Because of the time-based split that was used, the performance would likely hold up in such testing.

Interestingly, the gradient boosting model results in a similar predicted rejection rate as the scorecard, even though this is much lower than the actual rejection rate. The scorecard was originally calibrated to have a rejection rate of 10% and achieved 6.5% on the test sample. The gradient boosting model was trained on a sample with a rejection rate of 10.6% and predicts an even lower 5.4% as rejections. The actual rejection rate in the test sample is much higher than in the train sample at $\hat{p} = 30.5$%. It is known that the rejection rate has been increasing in a near-linear manner for the past six months, the reasons for which are not entirely clear and will not be covered here. It may, however, indicate that there is a time trend in the independent variable vector. This could correspond to a shift in the clients' attributes that underwriters are responding to but the models are not. Nonetheless, such a trend is not known as of now. The decreasing rejection rate of the models could be remedied modelling-wise by including a time variable; however, that would require assumptions about the future behavior of the time series.

The vast discrepancy between the rejection rate in the training and test sample furthermore causes the model to perform worse than if the data split were random. Nonetheless, this would have skewed the results in favor of the newly estimated model, which is why we have chosen to use a time-based split. As an alternative to changing the targeted metric, the model's robustness to a change in the rejection rate could potentially also be increased by reincorporating class-rebalancing to bring the minority class up to, for example, 20%. Because this would be much lower than in the SMOTE models that were tested, it would likely have less strong adverse effects on the respective metrics. Additionally, using the F1 score as a performance metric to combat class imbalance may have been beneficial; although ultimately the best results would be achieved by choosing a metric in correspondence with the business objective, in this case the negative predictive value (NPV).

The challenger model was able to outperform the scorecard in terms of precision and specificity, while lacking some of the scorecard's sensitivity. This is minor, and remedied by its lower rate of wrong rejections. However, even though the wrong rejections (FPR) were halved, the overall difference in performance is rather marginal. The lower FPR may still present a significant benefit over the scorecard, but could not be sufficiently validated based on the available data.

4.3 Feature Selection & Interpretability

Using the feature importance measures and partial dependence plots that were computed pursuant to Friedman (2001), we did find the model to be fairly interpretable. The main advantage of regression models over this approach remains that the coefficients are directly interpretable as the marginal effect of a feature on the target variable, remaining the same for any given value of the feature (save for the transformation of log-odds to probability). The ease of interpretability with regard to one given set of values could be improved for the machine learning model by increasing the degree of freedom, i.e. by reducing the number of trees.

During data analysis we found correlations to the target variable to be low. In retrospect, it is questionable whether Pearson's correlation coefficient was a good choice for sets of two discrete variables. Since dummies were built for some of the independent variables, these may have caused low correlation. Apart from this, even for the continuous independent variables, the dependent variable is discrete, potentially causing the same problem. Marcus (2022) teaches the use of Pearson's correlation coefficient for discrete variables, which is what was used in this work. Opposingly, Granström and Abrahamsson (2019) use Kendall's τ instead. Fortunately, the model itself is not affected by this choice, and even the feature selection would likely have been the same. Nonetheless, a correlation analysis following, for example, Kendall, could possibility have yielded additional insights.

4.4 Ethical Implications

In the process of building the challenger model, we have seen that gradient boosting models are capable of adjusting significantly depending on the targeted metric. We observe a high TNR, or specificity, for the challenger model; however, it comes at the cost of a high rate of false negatives (i.e. type II errors) – just like the scorecard. This is not an issue for this application, because only rejections are of interest. However, an alternative approach would be to target automating approval decisions instead. This would have the advantage that the bank could automate an even larger portion of loan application decisions. Even if a model were to perform worse on this target variable, the increase in efficiency could potentially offset the higher cost of risk from false predictions. This could be an ethically superior outcome too, as the rejection of clients would be decided by humans, which is often considered preferable to automating the rejection of a client. One option to achieve this goal is to optimize the hyperparameters to maximize the NPV. Based on the efficiency increase achieved in this study, this could be a promising approach.

When applying a model in a context as sensitive as this one, it is important to consider the potential for biases in the data and the model. This is especially relevant when using machine learning, as the model may reproduce biases in a less recognizable manner than parametric models. The partial dependence plots were carefully investigated and found to be in line with business intuition for most variables. In order to implement such a model in a production environment, it would be necessary to investigate the model's behavior in more detail, especially with regards to codependence of variables. Additionally, it should always be considered how a model performs when the underlying data shifts. For example, the confusion matrices in this study could look vastly different if there had been higher inflation during the time period of the data. To remedy this, it may be helpful to use a regressor instead of a classifier, as it would allow for an easier threshold adjustment by creating risk groups. This would make it easier to observe the share of clients in each group. Furthermore, the degree of assessment could be adjusted to the risk group, integrating the model into the decision-making process of the responsible underwriters. Another possibility to make the model more robust to changes like inflation is to use ratios. Vidovic and Yue (2020) compile a list of industry-standard ratios and observe extremely good model performance by using them. Unfortunately, most of the variables and ratios could not be computed for our dataset, as the relevant columns contain far too many missing values.

4.5 Limitations & Future Research

The dataset that was available for this analysis had systematic data quality issues. Out of the initial 92 000 observations only 34 000 met basic quality requirements (rounded). Some had to be excluded because of a change in questions in the application process, but most were either rejected right away for not being filed correctly, or were missing key information. This is a major issue for the bank, as it is not possible to build a reliable model without sufficient data. The bank should therefore consider investing in a more robust data collection process, as it would allow for a more reliable model as well as more extensive business insights. Unfortunately, these issues also imply that the results of this study may not generalize to the entirety of borrowers. While the filtering criteria were carefully selected to ensure that the sample is as representative of the population as possible, we cannot be certain that no bias has been introduced.

Further research could clarify whether the observed ability of Gradient Boosting to identify the minority class holds up on a larger scale. It is also highly relevant if there are time trends in specific variables, and what nature they are of. This could encompass an analysis of the political and macroeconomic factors that influence microcredit borrowers in this region. Furthermore, in order to determine the potential for the implementation of machine learning models for decision-making, research is needed on the legal and ethical implications of such models. Specifically, the regulatory practices may differ from country to country and even between different regulatory authorities within a country.

Another promising application where machine learning could be beneficial in this field is natural language processing. For example, the purpose of the loan – *LoanPurpose* in our dataset – and even free text inputs from applicants could be used to extract additional information based on the content and wording, albeit the ethical implications of such a usecase would have to be considered very carefully. Thus, instead of using such a source to improve a rejection model, it could be leveraged to pre-process the data and reduce the amount of manual work required by the underwriters.

5 Conclusion

The goal of this study was to establish the potential of machine learning in the context of credit decision modelling in a least-developed country. We developed a challenger model using gradient boosting, which is a machine learning technique that has been shown to be effective in similar applications. The challenger model was able to drastically lower the share of faulty rejections, even though the overall share of correct predictions was similar to the scorecard's. This is due to the specificity of the model being higher at the cost of a marginally lower sensitivity. As discussed in section 2.7, this minor difference outweighs the higher precision by some metrics, because it is amplified by the class sizes. Because the model targets rejections, this is irrelevant in this application, as the approved cases are forwarded to the underwriters for final approval. In this case, it is more important to minimize the number of wrong rejections than the number of wrong approvals. These results are very desirable and would be highly beneficial in practice; however, the sample size was not large enough to test their significance.

The second objective was for the model to be interpretable. The results in section 3 demonstrate that the model can be interpreted and analyzed thoroughly, albeit less intuitively than a regression model like the scorecard. Although the scorecard held up well in comparison, it does not reflect some non-linear patterns that were discovered during the development of the challenger model, like the ones mentioned in section 4. These features could be incorporated more adequately by transforming the variables appropriately when using them in a parametric model. Overall, these results confirm that gradient boosting is a suitable algorithm for this application.

We suggested ways to invert the model target in order to automate approvals instead of rejections in section 4.4. While the scorecard was designed for rejections because approvals were more difficult to identify, such a model would have a clear ethical advantage; the rejection of a loan application can have substantial consequences for the applicant and should therefore be well-considered. The flexibility of gradient boosting models may enable this inversion without a major loss in performance. In addition, automating approvals could massively enhance the processing time for a loan application, as the time it takes for a human underwriter to review the application could be reduced or even eliminated. This would give the bank a competitive advantage over other MFIs in the region, as the processing time is a major factor for the popularity of microcredit products. This could potentially lead to an increase in the number of applications, which in turn could be leveraged to offset the possibly higher cost of risk.

To summarize, there is great potential for the application of machine learning in the finance sector in LDCs. Its use for credit decision modelling is promising and demands for further research, but other applications are feasible as well. Research on machine learning in finance is still scarce, and we expect to see a much broader scope of applications within the next years.

References

Walsh, John E. (Dec. 1963). "Bounded probability properties of Kolmogorov-Smirnov and similar statistics for discrete data". In: *Annals of the Institute of Statistical Mathematics* 15(1), pp. 153–158. ISSN: 1572-9052. DOI: 10.1007/BF02865912.

Michie, Donald, D. J. Spiegelhalter, and Charles Taylor (Jan. 1999). "Machine Learning, Neural and Statistical Classification". In: *Technometrics* 37. DOI: 10.2307/1269742.

Friedman, Jerome H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine". In: *The Annals of statistics* 29(5), pp. 1189–1232. ISSN: 0090-5364.

Chawla, Nitesh V. et al. (June 2002). "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16, pp. 321–357. ISSN: 1076-9757. DOI: 10.1613/jair.953.

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Mitchell, Tom M. (2017). *Machine Learning*. McGraw Hill series in computer science. McGraw Hill. ISBN: 978-1-259-09695-2. URL: https://books.google.de/books?id=ifdcswEACAAJ.

Granström, Daria and Johan Abrahamsson (2019). *Loan Default Prediction using Supervised Machine Learning Algorithms*. OCLC: 1235232666. Stockholm. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-252312 (visited on 2023-07-26).

Zülch, Henning and Matthias Hendler (2019). *International financial reporting standards (IFRS) 2019: deutsch-englische Textausgabe der von der EU gebilligten Standards und Interpretationen*. 13. Ausgabe. Wiley Text. Wiley: Weinheim. ISBN: 978-3-52750982-9.

Bowman, Zach (2020). *Credit risk modeling during the COVID-19 pandemic: Why models malfunctioned and the need for challenger models*. URL: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/audit/us-audit-cecl-credit-risk-modeling.pdf.

Vidovic, Luka and Lei Yue (Nov. 2020). *Machine Learning and Credit Risk Modelling*. S&P Global. URL: https://www.spglobal.com/marketintelligence/en/news-insights/blog/machine-learning-and-credit-risk-modelling (visited on 2023-07-26).

Yıldırım, Soner (Feb. 2020). *Gradient Boosted Decision Trees-Explained*. URL: https://towardsdatascience.com/gradient-boosted-decision-trees-explained-9259bd8205af (visited on 2023-07-27).

Bussmann, Niklas et al. (Jan. 2021). "Explainable Machine Learning in Credit Risk Management". In: *Computational Economics* 57(1), pp. 203–216. ISSN: 0927-7099, 1572-9974. DOI: 10.1007/s10614-020-10042-0.

Cranenburgh, Sander van et al. (2022). "Choice modelling in the age of machine learning - Discussion paper". In: *Journal of choice modelling* 42. ISSN: 1755-5345.

Folpmers, Marco et al. (2022). *The application of machine learning and challenger models in IRB Credit Risk modelling: The use in risk driver selection*. URL: https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/risk/deloitte-nl-risk-challenger-models-v01-risk-driver-selection.pdf (visited on 2023-06-27).

Machado, Linnéa and David Holmer (2022). *Credit risk modelling and prediction: Logistic regression versus machine learning boosting algorithms*.

Marcus, Jan (2022). *Schließende Statistik*. Lecture Series Fall 2022. Berlin, Germany.

Pfeiffer, Christian (June 2022). *Implementation of the expected credit loss model for receivables*. URL: https://kpmg.com/de/en/home/insights/2018/06/expected-credit-loss-receivables.html (visited on 2023-06-27).

Binsbergen, Jules H van, Xiao Han, and Alejandro Lopez-Lira (2023). "Man versus Machine Learning: The Term Structure of Earnings Expectations and Conditional Biases". In: *The Review of financial studies* 36(6), pp. 2361–2396. ISSN: 0893-9454.

Frank, Murray Z, Jing Gao, and Keer Yang (2023). "Behavioral Machine Learning? Computer Predictions of Corporate Earnings also Overreact". In: *arXiv.org*. ISSN: 2331-8422.

Scikit-learn contributors (2023). *User Guide*. URL: https://scikit-learn/stable/user_guide.html (visited on 2023-07-31).

Appendix

Feature Elimination

In the simplest case, we would include all available variables in a model. However, this could easily lead to overfitting, as the model would be able to learn the noise in the data, which would lead to poor performance on unseen data. Regression models are often selected using the Akaike or Schwarz criterion, which penalize variables that do not improve the model's performance enough to justify their inclusion. Such criteria do not appear to be common in the machine learning space. Another approach from regression modelling is to conduct a univariate regression analysis for each feature and the target variable. The features with the highest explanatory power are then selected for the model. However, this approach does not take into account the interaction between variables, which is a major advantage of machine learning models.

We conducted a correlation analysis, which is visualized in Figure 6. It indicates a noticeable correlation to the target variable only for *SecondActivity* and *DaysOfSale* (both have an absolute value of $r = 0.12$ with *decision*). The following variables have a correlation coefficient $|r|$ between 0.05 and 0.075: *SalesLastWeek, AmountApplied, HouseholdFood, Children, HousingCondition Propriétaire, CashAndSavings, Stock*, and *Adults*.

In addition to the variables that qualify based on the correlation and univariate regression analysis, the scorecard that is used in production further contains CashAndSavings, FixedAssets and Stock, yet it does not contain the variables Adults, Children, AmountApplied, SupplierAdvance, and BusinessCondition. This thesis used a combination of both selections, resulting in the set of variables listed below. However, AmountApplied was dropped because it has major implications for the lending policy and would make it difficult to compare the resulting challenger model to the scorecard. Furthermore, we decided to drop variables with a high number of missing values, as we cannot be certain whether they are missing at random or following a trend. Therefore, they could introduce bias. The variable MaritalStatus was removed to ensure a non-discriminatory lending policy. The following variables were dropped due to missing or zero-values: SalesLastWeek at 20409 non-assigned values and SupplierAdvance at 22663 zero-values. The remaining variables are:
`Adults, BusinessCondition, Children, DaysOfSale, HouseholdFood, SalesMin, HousingCondition, CashAndSavings, FixedAssets, Stock, Sector, SecondActivity.`

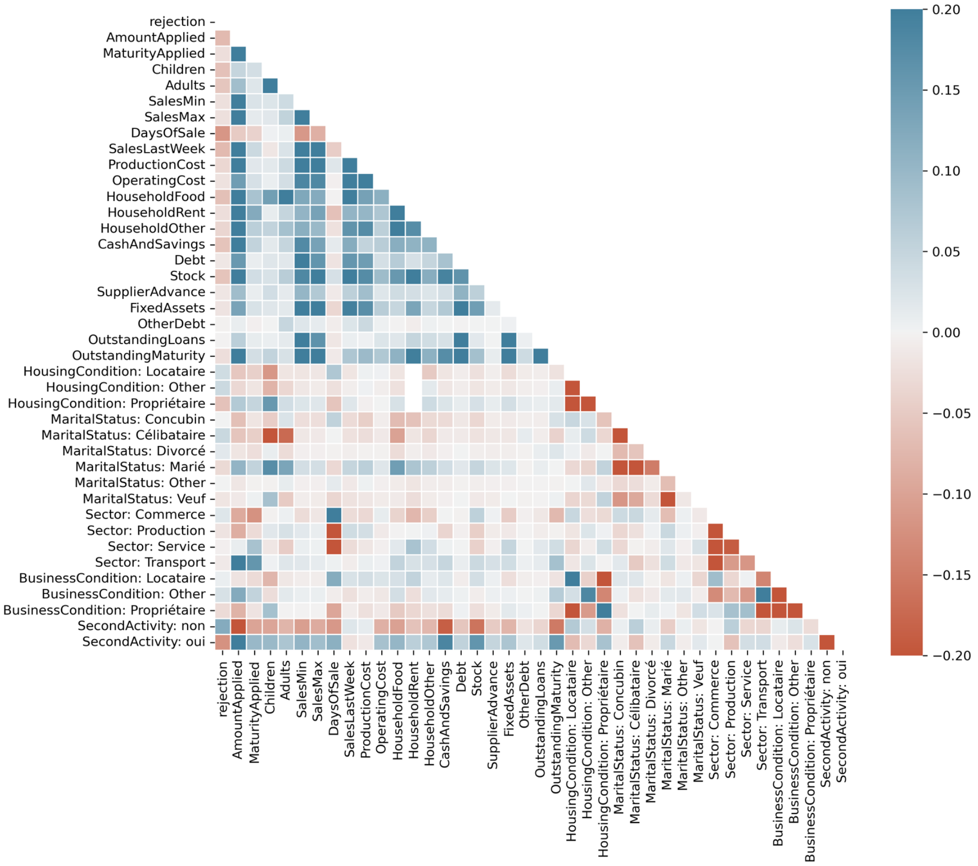

Figure 6: Correlation matrix of the relevant features of the dataset.

The first column depicts the correlation to the target variable. Note that the autocorrelation of each variable is excluded.

For some of these variables, the majority of the explainable difference (approximately 80% to 95% approval) corresponds to a very narrow range slightly above zero for the concerned variable. This is presumably a result of the analysts' decision-making process if they find that these ranges allow for a better separation between clients. Given this non-linearity, it may be beneficial to not only use a non-linear model but also scale or compute new variables, e.g. logged variables or linear combinations of variables. This is considered scaling in statistics or feature engineering in machine learning. However, it tends to be very time and resource intensive, and was hence not explored in this paper.

List of Variables

Table 3: Variables in the raw dataset.

	Variable	Type		Variable	Type
1	AssessmentId	cat	26	ProductionCost	#
2	NationalIdNumber	cat	27	OperatingCost	#
3	AnalyzedByUnderwriter	int	28	FixedAssets	#
4	RejectOnBot	int	29	Purchases	#
5	RejectReason	cat	30	SupplierAdvance	#
6	RejectComment	cat	31	CurrentAssets	#
7	AmountApplied	#	32	HousingCondition	cat
8	AmountApproved	#	33	MaritalStatus	cat
9	MaturityApplied	int	34	Children	int
10	MaturityApproved	int	35	Adults	int
11	LoanPurpose	cat	36	HouseholdFood	#
12	ProfessionalExperience	int	37	HouseholdRent	#
13	Sector	cat	38	HouseholdOther	#
14	SubSector	cat	39	CashAndSavings	#
15	SecondActivity	cat	40	Debt	#
16	SecondSector	cat	41	OtherDebt	#
17	SecondSubSector	cat	42	ElectricityExpenses	#
18	BusinessCondition	cat	43	TransportExpenses	#
19	SalesMin	#	44	ClothingExpenses	#
20	SalesMax	#	45	SchoolingExpenses	#
21	SalesAvg	#	46	MargeValue	#
22	SalesLastWeek	#	47	FinishedTime	cat
23	DaysOfSale	int	48	OutstandingLoans	#
24	Stock	#	49	OutstandingMaturity	int
25	OtherRevenue	#	50	dateApproved	cat
			51	decision	int

The variable names have homogenized and ordered. Type refers to the type of variable: date, categorical (denoted *cat*), floating-point number (#) or integer number (*int*).

Feature Importance in the Gradient Boosting Models

Table 4: Gini-Based Relative Feature Importances.

	GBC_{full}	$\text{GBC}_{\text{full}}^{\text{SMOTE}}$	$\text{GBC}_{\text{reduced}}$	$\text{GBC}_{\text{reduced}}^{\text{SMOTE}}$
Adults	1.37%	5.23%		
Children	1.29%	6.81%		
DaysOfSale	2.39%	30.47%	3.03%	36.02%
HouseholdFood	5.90%	5.38%	6.45%	8.67%
SalesMin	47.23%	8.68%	51.95%	13.49%
CashAndSavings	7.72%	5.64%	7.49%	7.99%
FixedAssets	14.43%	17.32%	16.40%	19.97%
Stock	14.60%	11.36%	14.68%	13.85%
BusinessCondition: Other	0.09%	2.31%		
BusinessCondition: Propriétaire	0.31%	2.19%		
HousingCondition: Other	0.13%	0.68%		
HousingCondition: Propriétaire	0.62%	0.73%		
SecondActivity: oui	0.27%	0.79%		
Sector: Production	0.65%	1.06%		
Sector: Service	2.81%	1.34%		
Sector: Transport	0.17%	0.02%		

Univariate Regression Analysis

The univariate regression analysis was conducted visually using logistic regression plots. The plots for the categorical variables are shown in Figure 7 and those for numberical variables in Figures 8 and 9. The numerical variables with a notable regression slope (i.e. beta factor) are listed below. For categorical variables, not only the slope, but also the difference in levels between classes is relevant.

1. Adults: medium positive trend
2. AmountApplied: medium positive trend
3. Children: medium positive trend
4. DaysOfSale: medium positive trend
5. HouseholdFood: medium-strong non-linear positive trend
6. SalesLastWeek: medium-weak non-linear positive trend
7. SalesMin: medium-weak non-linear positive trend
8. SupplierAdvance: weak non-linear positive trend

Additionally, there is a visible difference in levels between the two classes for these categorical variables:

1. BusinessCondition: weak negative trend for *Locataire* and *Other* compared to *Proprietaire* (English: tenant, other compared to owner)

2. HousingCondition: Similar to BusinessCondition, but with a stronger trend

3. MaritalStatus: medium-weak linear trend for *Celibataire* and *Divorce* compared to *Marié* and *Veuf* (English: single, divorced compared to married, widowed). No trend for *Concubin* (English: domestic partnership) and *Other*.

4. SecondActivity: medium-weak positive trend for *oui* (English: yes) compared to *non* (English: no).

5. Sector: weak negative trend for Commerce when compared to the other sectors.

Hyperparameter Tuning

Listed in the form:
Hyperparameter as scikit-learn argument: values to try, separated by comma

```
max_depth: 5, 10, 20, None
min_samples_split: 2, 5, 10, 25
learning_rate: 0.05, 0.1, 0.25
n_estimators: 5, 10, 50, 100
```

Figure 7: Regression plots for categorical features.

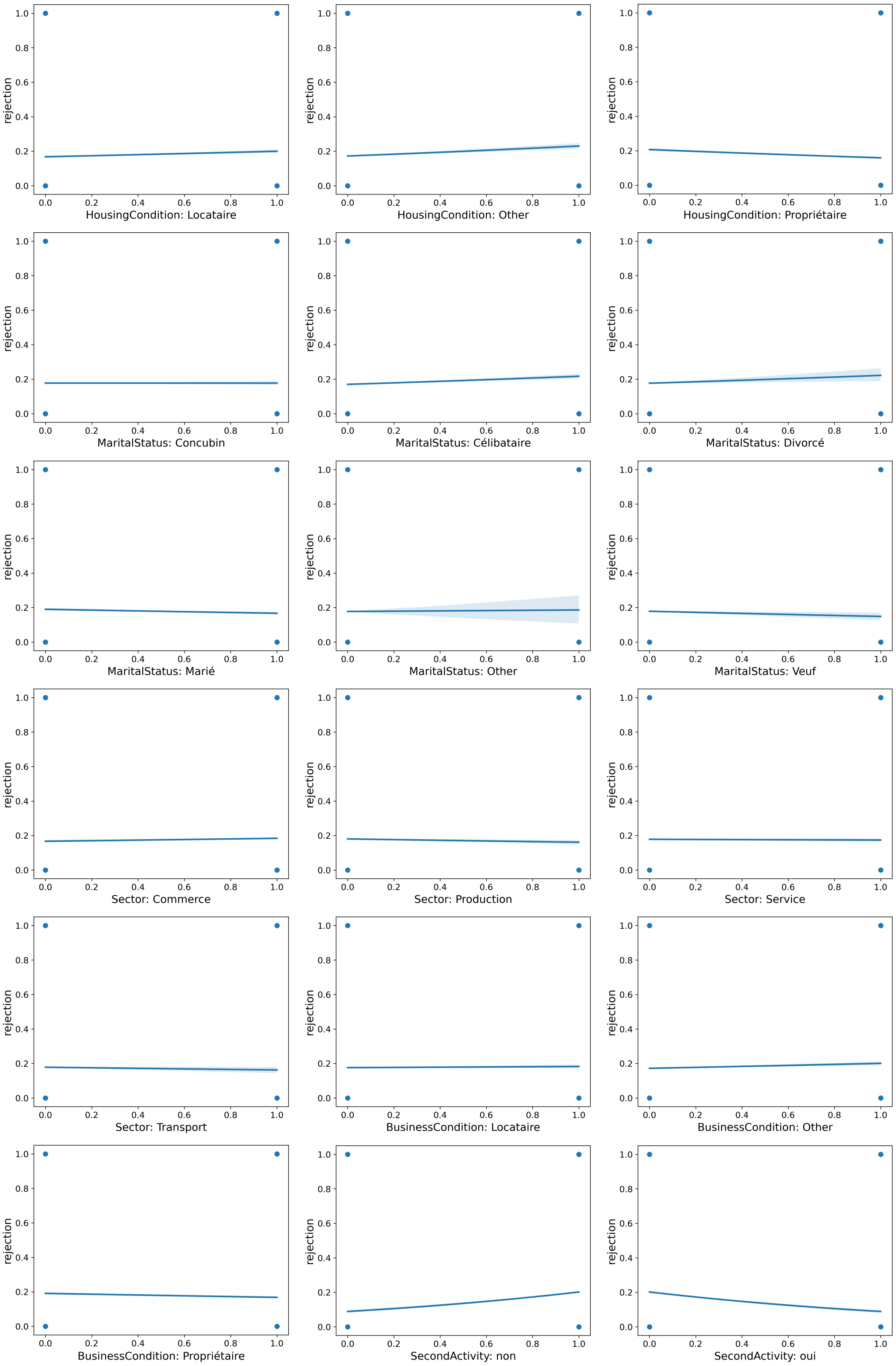

Figure 8: Regression plots for numerical features.

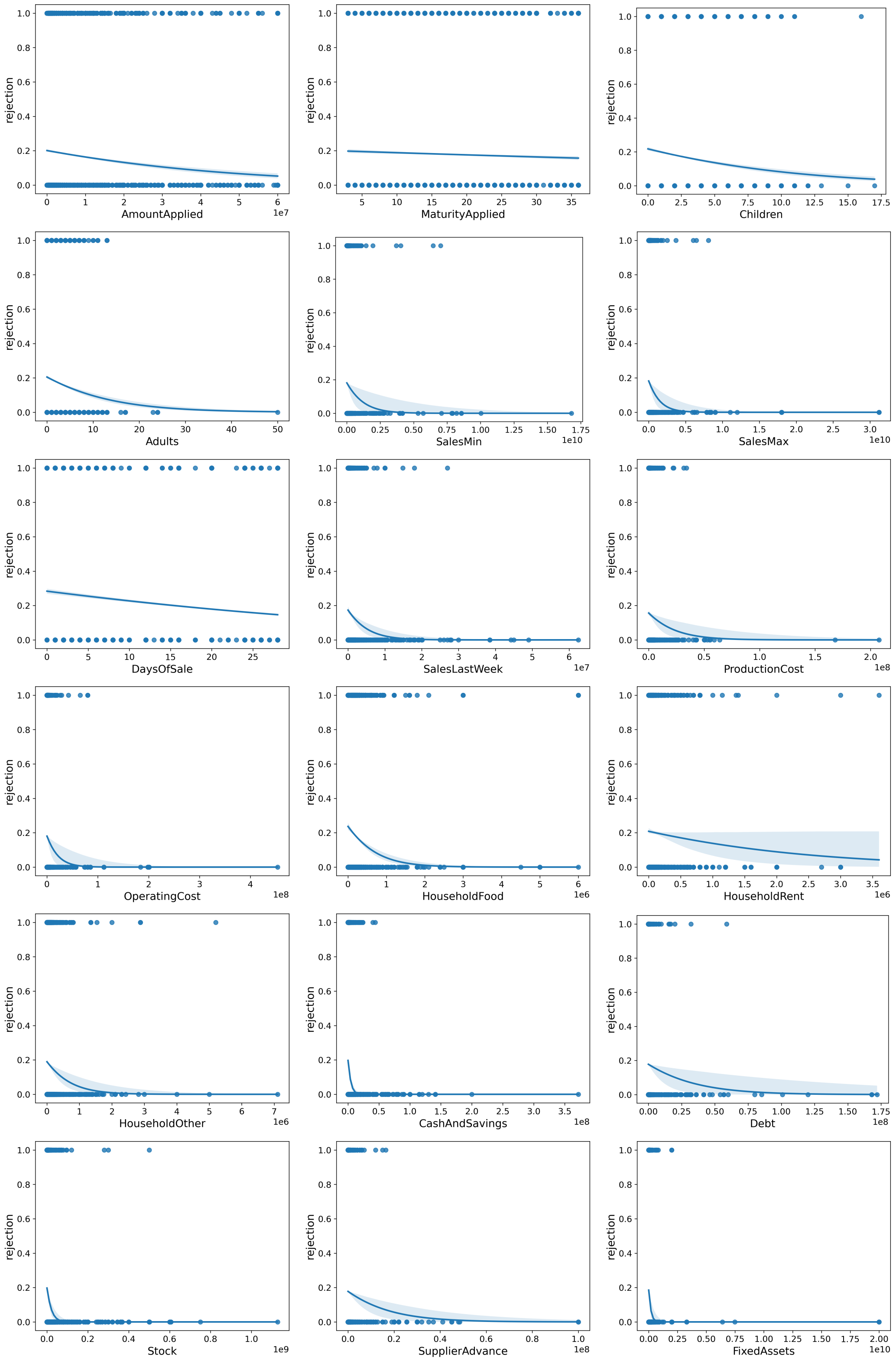

Figure 9: Regression plots for numerical variables – continued.

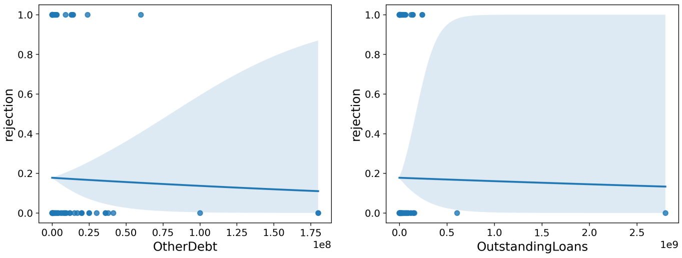

Table 5: Correlation coefficients of independent variables to target variable.

	Correlation to target
SecondActivity: non	0.1217
SecondActivity: oui	-0.1217
DaysOfSale	-0.1168
SalesLastWeek	-0.0711
AmountApplied	-0.0705
HouseholdFood	-0.0638
Children	-0.0624
HousingCondition: Propriétaire	-0.0608
CashAndSavings	-0.0590
Stock	-0.0580
Adults	-0.0561
MaritalStatus: Célibataire	0.0445
HousingCondition: Other	0.0428
HousingCondition: Locataire	0.0383
HouseholdOther	-0.0357
ProductionCost	-0.0337
MaritalStatus: Marié	-0.0299
BusinessCondition: Propriétaire	-0.0295
BusinessCondition: Other	0.0295
MaturityApplied	-0.0247
HouseholdRent	-0.0246
OutstandingMaturity	-0.0236
SalesMin	-0.0233
SalesMax	-0.0231
Sector: Commerce	0.0217
OperatingCost	-0.0191
Sector: Production	-0.0181
FixedAssets	-0.0173
MaritalStatus: Veuf	-0.0164
MaritalStatus: Divorcé	0.0156
SupplierAdvance	-0.0146
Debt	-0.0132
Sector: Transport	-0.0101
BusinessCondition: Locataire	0.0073
Sector: Service	-0.0042
OtherDebt	-0.0019
MaritalStatus: Other	0.0014
OutstandingLoans	-0.0008
MaritalStatus: Concubin	0.0000

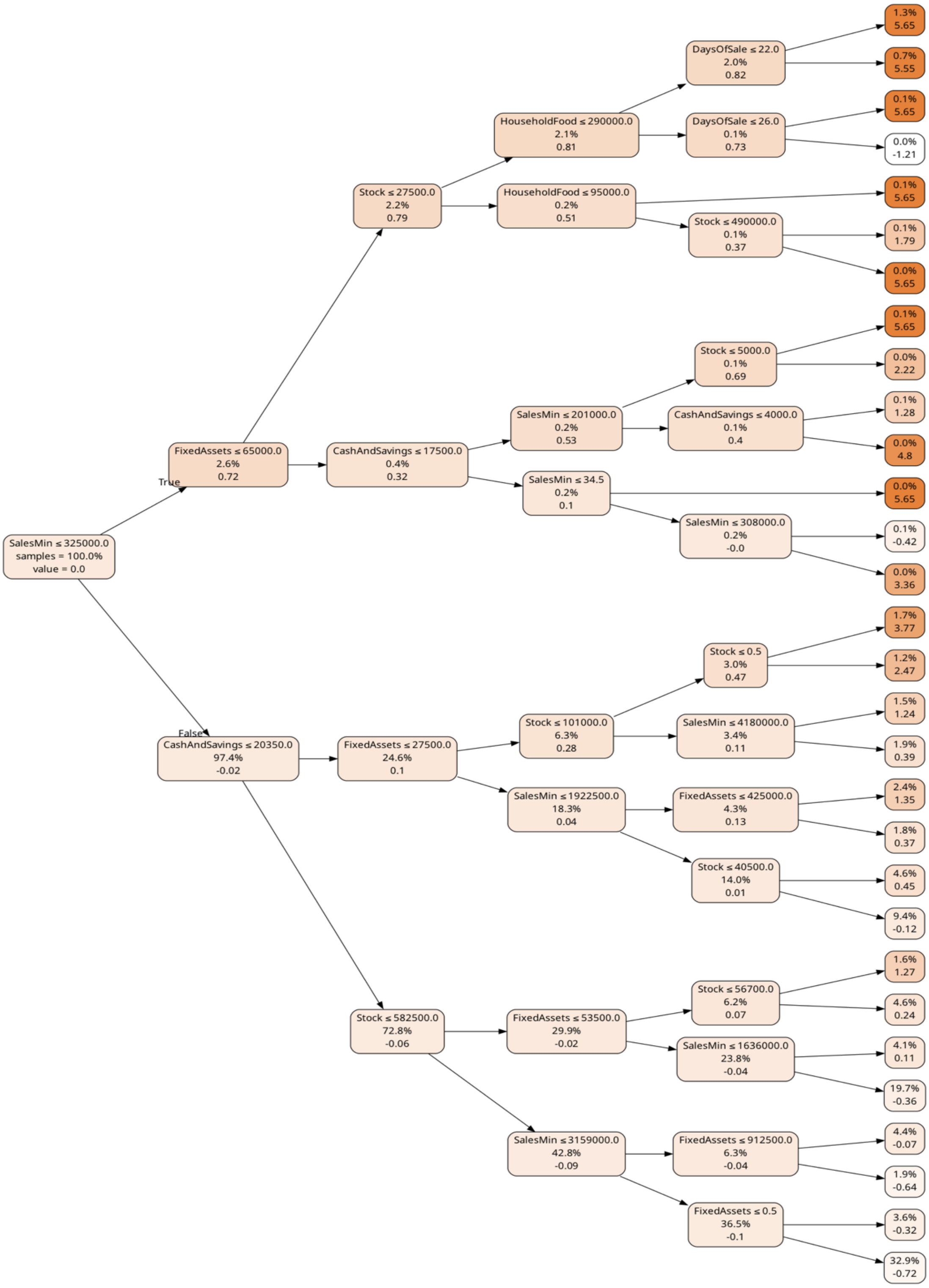

Figure 10: First decision tree of the reduced model GBC$_{\text{reduced}}$.

The decision trees of the SMOTE-rebalanced model could not be included in this document due to their large dimensions. They are included in the accompagning digital files.