Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

Borger Christopher Melsom
Christian Bakke Vennerød

# Explainable AI for Credit Scoring in Banks

Master's thesis in Industrial Economics and Technology Management
Supervisor: Sjur Westgaard
Co-supervisor: Petter Eilif De Lange
June 2022

NTNU
Norwegian University of Science and Technology

Borger Christopher Melsom
Christian Bakke Vennerød

# Explainable AI for Credit Scoring in Banks

Master's thesis in Industrial Economics and Technology Management
Supervisor: Sjur Westgaard
Co-supervisor: Petter Eilif De Lange
June 2022

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU

Kunnskap for en bedre verden

# Preface

This thesis concludes our Master of Science degree in Industrial Economics and Technology Management at the Norwegian University of Science and Technology (NTNU). The thesis embodies independent and original work performed by Borger Christopher Melsom and Christian Bakke Vennerød in the spring of 2022.

We would like to thank our supervisors, Professor Petter Eilif De Lange and Professor Sjur Westgaard, for their constructive feedback and encouragement throughout the entire process. They have our sincerest gratitude. Furthermore, we want to thank our collaborating Norwegian medium-tier bank, and our contact persons there. Although we cannot name them nor the bank we are sincerely grateful for all of their contribution and their deep domain expertise. This thesis could not have happened without them.

Trondheim, June 2022

# Abstract

Credit scoring models applied by banks are required by financial authorities to be sufficiently explainable. Logistic regression has long been the industry standard for these models. Over the last few decades, machine learning (ML) techniques have advanced default predictions further, with increased predictive performance. However, current ML approaches are often perceived as black boxes, meaning that it is hard to understand the inner workings of the models. The explainability deficit of the best-performing ML models means that banks have to sacrifice predictive power in order to abide by the regulations regarding explainability.

This paper proposes an explainable ML model for predicting credit default on a real-world dataset provided by a Norwegian bank. We combine a LightGBM model with SHAP, an explainable AI (XAI) framework, which enables the interpretation of explanatory variables affecting the predictions. The LightGBM model is compared to the bank's actual credit scoring model (Logistic Regression), where we achieve a 17% and 114% increase in ROC AUC and PR AUC, respectively. For comparison reasons, a separate LightGBM model with the same original features as the ones in Logistic Regression is trained, where we achieve a 9% and 56% increase in ROC AUC and PR AUC, respectively.

Our main contribution is the implementation of XAI methods in banking, exploring how these methods can be applied to improve the interpretability and reliability of state-of-the-art ML models. We specifically find that LightGBM models outperform LR models for credit scoring in terms of both predictive performance and explainability, and that the economic value of the predictive improvement can be substantial. For the same reasons, European legislators have acknowledged that ML might play an essential part in banking moving forward. This paper shows that XAI can assist banks in enabling ML by overcoming the obstacles related to explainability.

# Sammendrag

Finansmyndighetene stiller strenge krav til bankers kredittmodeller. De må være nøyaktige, men også tilstrekkelig forklarlige. Logistisk regresjon (LR) har lenge vært industristandarden innen banksektoren, men i løpet av de siste tiårene har maskinlæring (ML) forbedret de prediktive egenskapene til kredittmodeller betydelig. Imidlertid blir dagens ML tilnærminger ofte sett på som "svarte bokser", da det er vanskelig å forstå logikken og utregningene som ligger bak modellene. Gapet i forklarbarhet mellom de beste ML modellene og dagens LR modeller, gjør at bankene må ofre noe av nøyaktigheten i prediksjonene til kredittmodellene sine for å imøtekomme myndighetens krav til forklarbarhet.

I denne oppgaven presenterer vi en forklarbar ML modell som predikerer mislighold blant kunder i en norsk bank. Vi kombinerer LightGBM med SHAP, et forklarbart AI (XAI) rammeverk, som gjør det mulig å tolke hvordan de ulike forklaringsvariablene til modellen påvirker prediksjonene. LightGBM-modellen sammenlignes med bankens kredittmodell (Logistisk Regresjon), som benyttes daglig i praksis, hvor vi oppnår en 17% og 114% forbedring i hhv. ROC AUC og PR AUC. For å kunne sammenligne metodene mer direkte, trente vi deretter opp en egen LightGBM modell basert på variablene i LR-datasettet. Denne modellen gav forbedringer på hhv. 9% og 56% i ROC AUC og PR AUC.

De viktigste bidragene fra denne oppgaven er anvendelsen av XAI innenfor banksektoren, og analysen av hvordan disse metodene kan benyttes for å forbedre forklarbarheten og påliteligheten til moderne ML modeller. Vi viser at LightGBM-modeller er bedre enn dagens LR-modeller både når det gjelder nøyaktigheten i prediksjonene og forklarbarheten til modellen. I tillegg viser vi at den prediktive forbedringen ved å bruke av ML kan ha betydelig økonomisk verdi for bankene. Europeiske lovgivere anerkjenner at ML trolig vil være en helt vesentlig del av bankvirksomheten i årene som kommer. Denne oppgaven viser hvordan XAI kan hjelpe bankene med å overkomme utfordringene relatert til forklarbarheten til slike modeller.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

A recent report by the European Banking Authority (EBA) acknowledged that the standard regression models used in internal ratings-based (IRB) models might no longer be able to utilize all the data available for banks. Thus, EBA states that ML might play an essential part in banking moving forward (European Banking Authority, 2021). Linear and non-linear regression models (logit and probit models) have long been the industry standard for credit modeling. Over the last few decades, machine learning (ML) techniques have advanced default predictions further. However, current ML approaches are often perceived as black boxes, meaning that it is hard to understand the inner workings of the models (Ariza-Garzón et al., 2020; Gramegna and Giudici, 2021). With the implementation of the Basel II agreement (Basel Committee on Banking Supervention, 2006) and the General Data Protection Regulation (GDPR) (European Union, Parliament and Council, 2016), European banks have to abide by strict regulations enforcing a certain level of explainability in all decision-making data-based models. The regulations pose significant obstacles for banks seeking to employ state-of-the-art ML techniques for modeling credit risk (Bücker et al., 2021).

With its latest discussion paper, EBA uncovered three main challenges related to the complexity of ML models; i) The challenge of interpreting the results, ii) the challenge of ensuring that management functions properly understand the models, and iii) the challenge of justifying the results to supervisors (European Banking Authority, 2021). The European Union is also working on AI-specific regulations. In the European Commission's proposal for new AI regulations, AI systems used for credit scoring are defined as "high-risk AI systems." The Commission proposes that such systems should be bound by a much stricter regulatory framework related to transparency, opacity, et cetera, as credit scoring models determine people's access to financial resources (European Commission, 2021a). Overall, the European legislators demand better explainability from credit scoring models than current black-box ML models can provide.

In other words, there is a need to bridge the explainability deficit of current state-of-the-art machine learning models. One promising framework that might achieve this is the SHAP framework, which has been applied successfully in other high-risk areas, such as disease detection (El-Sappagh et al., 2021; Peng et al., 2021) and surgery technique selection (Yoo et al., 2020). In this study, we apply the XAI framework SHAP to an ML model to improve the performance of current credit scoring models while achieving a level of explainability within the guidelines of the European legislators.

The main contribution of this paper is the implementation of XAI methods on a real-world dataset, exploring how these methods can be applied to improve the interpretability and reliability of state-of-the-art ML models. We specifically find that LightGBM models outperform Logistic Regression models for credit scoring both in terms of predictive performance and explainability. The dataset used in this paper is particularly comprehensive, containing over 13 million records of time series data for 13,969 unique customers. The properties of the dataset enabled us to explore a novel way of integrating multivariate time-series analyses in the ML model through the use of daily balance data, significantly improving the performance of the credit scoring model.

This paper is organized as follows. In section 2, relevant literature regarding XAI for credit scoring is reviewed, and we rely on key findings in the literature to select the AI and XAI methods for this study. Section 3 outlines the models we employ. Section 4 introduces and explains the data set. Finally, section 5 provides an assessment of the models' performance, an in-depth analysis of the explainability of the models' output, and an analysis of the potential economic value of an improved credit scoring model.

# Chapter 2

# Literature review

European legislatures have enforced several regulations regarding the explainability of ML models, and these laws are expected to be strengthened in the future. Automated processes will be subject to stricter regulations, as errors and biases in the underlying data can have more significant ramifications in AI decision-making processes than in human (European Commission, 2021b) (as cited in Bibal et al., 2021). As the regulatory scrutiny related to explainability increases, it is essential for financial institutions to evaluate the explainability of their credit models (Yang and Wu, 2021). Bastos and Matos (2022) finds XAI to be a solution, as it enables banks to abide by the regulatory transparency requirements in the Basel agreements without sacrificing predictive accuracy.

Explainable AI (XAI) techniques can be applied to overcome the lack of explainability in black box AI models while preserving their predictive utility (Gramegna and Giudici, 2021). The two widely accepted state-of-the-art XAI frameworks are the LIME framework by Ribeiro et al. (2016) and SHAP values by Lundberg and Lee (2017). These models were created to help users understand the reasons behind predictions of complex models.

The literature focusing on the use of XAI for credit scoring in finance is very limited. Nevertheless, there are some highly relevant previous works. This involves integrating XAI on credit scoring models for P2P lending data sets (Misheva et al., 2021; Bussmann et al., 2020; Ariza-Garzón et al., 2020), applying XAI to explain home equity credit risk models (Davis et al., 2022), an empirical study comparing XAI with a scorecard model for credit scoring on a publicly available credit bureau data set (Bücker et al., 2021), comparing different XAI models' effectiveness on separating data from a set of small and medium-sized enterprises data (Gramegna and Giudici, 2021) and applying XAI to interpret a model for predicting crashes on S&P500 (Benhamou et al., 2021). We are not aware of XAI having been applied to an actual customer database from a bank before. However, the results from the above-mentioned applications of XAI on credit scoring are promising.

Misheva et al. (2021) analyze the effectiveness of LIME and SHAP XAI techniques in the context of credit risk management. Both LIME and SHAP are found to provide "consistent explanations." However, the SHAP values are highlighted as the most robust and effective in explaining the importance of the model's different features. Gramegna and Giudici (2021) also find that SHAP outperforms LIME in discriminating observations in their credit scoring model. Davis et al. (2022) apply both SHAP and LIME to analyze the explainability of the output from credit risk models. Although the authors find LIME to suffer from potential instability issues, they argue that the computation time of KernelSHAP makes it unscalable for datasets with

many features. This paper applies a different method, TreeSHAP, that does not suffer from scalability issues as it is polynomial in runtime. Bussmann et al. (2020) focus on one specific explainable model for fintech risk management, using XGBoost with SHAP. They find that this model clearly outperforms the LR base model in terms of predictive accuracy while also providing a detailed explanation for each prediction. This is in line with the findings in Bücker et al. (2021), which show that ML techniques can achieve a level of interpretability comparable to the traditional scorecard method while preserving its computational edge. According to Ariza-Garzón et al. (2020), applying XAI on non-linear models such as XGBoost may even improve the explainability compared to statistical approaches, e.g., LR. Such advanced models enable an understanding of complex, non-linear aspects of the relationships between variables that classic models are unable to discover. This includes aspects like "curved relationships, structural breaks, heteroscedasticity and outlying behavior" (Ariza-Garzón et al., 2020). Based on the results from Misheva et al. (2021) and Gramegna and Giudici (2021), we find sufficient evidence for utilizing SHAP in this paper.

The discussion above clearly shows that utilizing AI for enhanced predictive performance, in combination with XAI for sufficient explainability, can potentially improve current credit scoring models. However, a challenge with credit scoring as a classification problem is that only a small minority of the customers are usually expected to default, i.e., that the dataset is highly imbalanced. Gradient Boosting Decision Tree (GBDT) is an ML technique that has been frequently used for credit scoring in the literature because it provides good accuracy for such imbalanced classification problems (Brown and Mues, 2012; Benhamou et al., 2021). One example is Bussmann et al. (2020), who show that the GBDT method XGBoost (Chen and Guestrin, 2016) clearly yields better accuracy than the LR base model for predicting default on a P2P data set. This is in line with Ariza-Garzón et al. (2020) who find a GBDT model (XGBoost) to perform better globally than all other methods in their study of credit scoring models in P2P lending. They also show that this increased performance comes from "a better description of the relationships among the variables." The works conducted on P2P lending are closely related to credit scoring in banks, as the classification problem is fundamentally similar. Thus, we find convincing evidence in the literature for applying a GBDT model for credit scoring in this study. As Benhamou et al. (2021) finds LightGBM to be the better GBDT model, with three times the speed of XGBoost and similar predictive performance, this study will employ LightGBM.

This study applies LightGBM and SHAP to a comprehensive customer dataset. It extends the literature in three ways: (i) by implementing the credit scoring model on a real-life dataset from a bank, (ii) through the use of the bank's own LR model to benchmark the ML method, (iii) by utilizing multivariate time-series data to improve predictive performance, and (iv) by analyzing the potential economic gain from using LightGBM versus Logistic Regression.

# Chapter 3

# Methodology

This chapter provides a brief outline of Gradient Boosting Decision Trees, whereafter we present the essential features of LightGBM and Logistic regression. Also, a description of Shapley values is provided, and lastly, we outline the essential properties of SHAP.

## 3.1 Gradient Boosting Decision Trees

Ensemble methods combine several learners to obtain better predictive performance than a single constituent learning algorithm. The ensemble method used in this paper is boosting, where learners are trained on misclassified instances from the previous learners. Thus, several weak learners are combined into one strong learner. With weak learners, we mean models whose performance is slightly better than random chance. The advantages of using weak learners are outlined in Freund and Schapire (1995) and can be summarised as being computationally simple, with the ability to reduce overfitting and bias (Bartlett et al., 1998). Furthermore, a broad range of hyperparameters can be applied to the model in order to force each learner to remain weak, which will be further discussed in chapter 5.

Gradient Boosting Decision Trees (GBDT) utilize the boosting technique by sequentially training decision trees based on the residuals from the previous trees. Building on the works by Zhang et al. (2017), a standard GBDT model can be expressed with the following set of equations.

We are given $MxN$ input data, with $X = \{\mathbf{x_i}\}_{i=1}^{M}$, feature vectors $\mathbf{x_i} = (x_{i1}, \ldots, x_{iN})$, and targets $Y = (y_1, \ldots, y_M)$. Overall, GBDT tries to find a strong learner $F$ by minimizing a loss function $L$:

$$L = \arg\min_{F} \sum_{i=1}^{M} l(y_i, F(\mathbf{x_i})) \tag{3.1}$$

Here, the strong learner $F$ can be represented as a sum of $T$ weak learners $f_w$ (e.g., decision trees), $F(\mathbf{x_i}) = \sum_{w=1}^{T} f_w(\mathbf{x_i})$. At the $w$-th stage, the previous $w-1$ weak learners are fixed when learning the $w$-th weak learner. Thus, when constructing the $w$-th learner, the following loss is minimized by GBDT:

$$L_w = \sum_{i=1}^{M} l(y_i, F_{w-1}(\mathbf{x_i}) + f_w(\mathbf{x_i})) \tag{3.2}$$

Here, $F_{w-1}(\mathbf{x}) = \sum_{k=1}^{w-1} f_k(\mathbf{x})$. This can be further approximated by using first- and second-order Taylor expansions:

$$L_w = \sum_{i=1}^{M} \left[ l(y_i, F_{w-1}(\mathbf{x_i}) + g_i f_w(\mathbf{x_i}) + \frac{h_i}{2} f_w^2(\mathbf{x_i}) \right] \tag{3.3}$$

Where $g_i = \frac{\partial l(y_i, F_{w-1}(\mathbf{x_i}))}{\partial F_{w-1}(\mathbf{x_i})}$, and $h_i = \frac{\partial^2 l(y_i, F_{w-1}(\mathbf{x_i}))}{\partial^2 F_{w-1}(\mathbf{x_i})}$ are the first- and second-order partial derivatives, respectively. Thus, GBDT performs gradient descent in the function space; at each step $w$, GBDT tries to find the function $f_w$ that minimizes $L_w$. Each weak learner $f_w$ trains on the negative gradient of the loss function, with respect to the previous predictions, $F_{w-1}$, instead of actual labels $Y$. The result is a model for reducing bias and variance, and that can be used for both regression and classification on numerous applications (Breiman, 1998).

## 3.2 LightGBM

One of the limitations of traditional GBDT methods, such as AdaBoost (Freund and Schapire, 1999) and XGBoost (Chen and Guestrin, 2016), is the time-consuming process of iterating through all of the data in order to estimate the information gain for all possible splits (Quinto, 2020). Light Gradient Boosting Machine (LightGBM) is a variant of GBDT designed to be significantly faster than conventional GBDT techniques without sacrificing accuracy. This is done by implementing Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) (Ke et al., 2017). GOSS exploits that the information gain for instances with larger gradients (undertrained instances) is higher. By randomly dropping instances with smaller gradients, and to a larger extent keeping instances with larger gradients, the number of instances used for training can be reduced without sacrificing the accuracy in information gain estimation used for feature splitting in GBDT. EFB exploits sparse data by bundling mutually exclusive features into a single feature. These two improvements significantly improve the computational speed and memory consumption of LightGBM, making it state-of-the-art for many applications (Ke et al., 2017).

## 3.3 Logistic Regression

To evaluate the LightGBM model's relative performance, a Logistic Regression (LR) baseline model was used. LR is commonly used to predict categorical values (Lever et al., 2016) and is currently the most popular method for credit scoring in banks. In order to make the baseline as realistic as possible, our cooperating bank generously provided us with their LR model, which is used as of this date to perform credit scoring in practice. The essential property of LR is that a linear combination of independent variables can be mapped to a probability score (Hess and Hess, 2019) and that the dependent variable can be classified into two groups based on the scores (Bussmann et al., 2020). The linear model $\pi = \beta X$ is the simplest, but the term on the right-hand side may take on any real number, whereas the probability on the left side must lie between zero and one. This trick is performed by the logit function. The logit model we have employed in this study is:

$$P(Y_n = 1 | x_{1n}, ..., x_{Tn}) = \frac{1}{1 + e^{-(\alpha + \sum_{t=1}^{T} \beta_t x_{nt})}} \tag{3.4}$$

We refer to Appendix A for a more detailed description of Logistic Regression.

## 3.4 Shapley values

With LR, it is trivial to see how a given feature value $x_j$ contributes to the prediction. The effect of feature $j$ is the difference between the feature value and the average feature value, that is:

$$\theta_j(\hat{f}) = \beta_j x_j - \mathbf{E}(\beta_j X_j) \tag{3.5}$$

Here, $\mathbf{E}(\beta_j X_j)$ is the mean effect estimate for feature $j$. Similarly, we can find the feature contributions of all features for a given instance by taking the predicted value less the average predicted value:

$$\sum_{j=1}^{N} \theta_j(\hat{f}) = \hat{f}(x) - \mathbf{E}(\hat{f}(X)) \tag{3.6}$$

For more complex non-linear models, such as LightGBM, finding these feature contributions is more complicated due to the inherent complexity of the model. Despite being non-linear in probabilities and odds, LR is linear in log-odds. Thus, given that the features are independent, the feature effect in log-odds can be found by multiplying the feature coefficient with the feature value, similar to linear models. In a non-linear model, however, the effect of a feature can also depend on other features' effects, making it much harder to estimate feature effects. Shapley values lever ideas from cooperative game theory to tackle this problem (Shapley, 1953). Shapley values were initially used for calculating a *fair* payout, i.e., finding payouts to players reflecting their contribution to the total payout. Since the sum of all individual payouts equals the total payout to the coalition, Strumbelj and Kononenko (2013) found that Shapley values can be applied for explaining models by viewing features as players and the predictions as payouts. Thus, given a game with $M$ features participating, where the aim is to maximize some objective function, we have the following.

Let $S \subseteq M = \{1, \ldots, M\}$ be a feature group, i.e., a subset consisting of $|S|$ features. In addition, let $v(S)$ be a contribution function that maps feature subsets to real numbers, indicating the contribution of feature group $S$ to the total prediction. Then, the amount that feature $j$ contributes to the final prediction of one instance is the weighted sum of all possible feature group combinations:

$$\phi_j = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|!(M - |S| - 1)!}{M!} (v(S \cup \{j\}) - v(S)), j = 1, \ldots, M \tag{3.7}$$

An interpretation of Equation 3.7 is that Shapley values represent the average expected marginal contribution of a feature on a given prediction after all feature combinations have been checked. Informally, this can be expressed as:

$$\phi_j = \frac{1}{\#\text{ players}} \sum_{\text{coalitions excluding j}} \frac{\text{marginal contribution of j to coalition}}{\text{number of coalitions excluding j of this size}} \tag{3.8}$$

Over the years, several techniques for explaining AI models have been developed, such as LIME (Ribeiro et al., 2016) and DeepLift (Shrikumar et al., 2019). Common to these techniques however, is that they do not necessarily meet the properties of *local accuracy*, *missingness*, and *consistency*. In order to have a unified measure of feature importance, an explanatory model should satisfy the following three requirements. It should match the original model for a single instance (*local accuracy*), attribute zero importance to missing features in a given coalition (*missingness*), and increase any attributions for a given feature if the underlying model changes into giving that feature more impact (*consistency*) (Lundberg and Lee, 2017). Young (1985) found that the only values satisfying these three properties are Shapley values. This implies that any explanation technique not based on Shapley values will violate local accuracy or consistency (Molnar, 2019).

## 3.5 SHAP

Using Equation 3.7 directly would yield exact Shapley values, but it would require retraining of the prediction model on all feature subsets $S \subseteq M$, where M is the set of all features. With the exponential complexity of Equation 3.7, calculating Shapley values exactly would thus be challenging and computationally expensive. One solution to this problem is using weighted linear regression (KernelSHAP) (Lundberg and Lee, 2017). Another approach, and the one employed in this study, TreeSHAP (Lundberg et al., 2019), is optimized for tree-based machine learning models such as LightGBM. TreeSHAP uses the conditional expectation $\mathbf{E}_{X_S|X_C}(\hat{f}(x)|x_S)$ as the contribution function $v$ in Equation 3.7 to estimate feature attributions. Here, $X$ is a matrix of instances, $x$ is a single instance, $X_C$ is coalition data, and $\hat{f}$ is the underlying model.

Thus, given an ensemble tree, by pushing all subsets $S \subseteq M$ down each tree simultaneously and keeping track of each subset's overall weights as well as the number of subsets, Shapley values of each tree can be calculated in polynomial time (Molnar, 2019). Moreover, because of the additive property of Shapley values (Shapley, 1953), the Shapley values of the ensemble tree model equals the weighted average Shapley values of the individual trees.

# Chapter 4

# Data

The models outlined in chapter 3 were implemented on a proprietary dataset, generously provided by a medium-tier bank in Norway. The data set contains time series data for 13,969 unique customers and is split into two different files; a *mainfile* that consists of monthly customer application data and behavioral data with a total of 268,120 records, and a *balancefile* that contains 13,017,635 records of daily account movements. These data sets are linked through unique customer identification numbers and dates. The data contains only unsecured consumer loans and is captured over approximately four years.

The data contains historical customer data captured in Norway over the past years, where each row in the *mainfile* represents one month for one customer, and each row in the *balancefile* represents one day for one customer. It is captured end-of-month and end-of-day, respectively. The data is imbalanced, meaning that the target variable is unevenly distributed. More specifically, the defaulting customers constitute a minority class of 8.8% of the total customers. The target variable indicating default is determined by the customer being in default for at least 90 days within the 12 months following the scoring date. The choice of target is in line with the regulatory definition of default for Norwegian banks and thus the industry standard.

The following sections outline the overall strategy behind creating the final datasets used for the LightGBM and the LR model. First, the data preparation steps conducted for both models are presented. Then we present the exploratory data analysis conducted on the finished processed datasets. The features used in both models are further explained in Appendix B.

## 4.1 Data preparation for LightGBM

Utilizing daily account movements in credit scoring models has rarely been performed in the literature. One of our main hypotheses for how to advance the performance of the credit scoring model was that for some customers, the reasons explaining a default could be found in the historical changes in the daily account data. Therefore, it was relevant to include as much historical data at the highest granularity possible.

Unlike models such as LSTM and ARIMA, LightGBM and Logistic Regression are not designed to handle time-series data directly. Thus, as both datasets were multivariate time-series, several data processing steps were conducted in order to convert the temporal data into static data that can be utilized by the LightGBM and Logistic Regression models. These steps can be summarized as data filtering, feature extraction, and feature selection. Note that these steps were only applied to the

LightGBM model dataset, as the bank predefined the features used in the Logistic Regression model. This is further discussed in section 4.2.

### Data Filtering

As the data contains a large number of observations per customer, it was necessary to filter out noise. Figure 4.1 shows the overall strategy for selecting these observations. In the figure, the customer is said to be in *legal default* after having failed to fulfill its loan obligations for 90 days, shown as the *pink* line. The objective is to predict such legal defaults occurring within the next 12 months, as shown with the stapled *orange* line. For the remainder of the text, we define this as a *default*, unless otherwise specified. Thus, all observations after a default are irrelevant for predicting legal default and, consequently, removed for all defaulting customers.

Based on the logic described above, only the last 90 days leading up to a default are used in the models. The 3-month window was chosen to compromise between including enough historical data and adding too many observations, leading to noise. Hence, since the *mainfile* and the *balancefile* are structured in a monthly and daily format, three and 90 observations prior to a default event were considered, respectively. This is shown on the left-hand side of Figure 4.1. Furthermore, the fact that the target variables were fitted retrospectively to the dataset enabled us to use the last observations in the datasets for the non-defaulting customers.

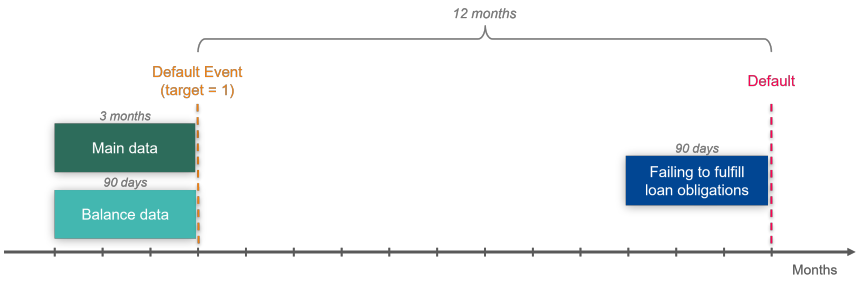



FIGURE 4.1: Timeline illustrating the distinction between default and legal default. 90 days of balance data and 3 months of main data are used for predicting the probability of a legal default within the next 12 months. A legal default occurs if a customer fails to fulfill its obligations over a period of 90 days.

### Feature extraction

Once the data was filtered, several aggregation measures were implemented to extract signals from the data. These aggregation measures differed between the *mainfile* and the *balancefile*.

On the *mainfile*, a pivot transformation was implemented. This transformation is exemplified through Table 4.1, where the original data is converted to a format with one row per customer, i.e., a matrix with $13,969$ rows. The new table is built around the last observation of each customer, following the description above. In the pivot operation, one- and three-month lags were utilized. There are two reasons why a customer might have its last observation at time $T$; either it enters a default at $T+1$, or there are no subsequent observations for the customer in the data. Either way, the final dataset will include features captured at time $T$ and the lagged features from $T-1$ and $T-3$. Note that in the pivot transformation, missing observations are preserved to ensure consistency between the observations. Hence, if the *mainfile*

has a customer with only 1 observation, as is the case for new loan applicants, NaN values are generated for the lagged features.

In order to capture the development leading up to the last observation, several functions were applied to the lagged features, further increasing the feature space. In addition to including the still features, shown in Table 4.1b, both the actual difference in feature values and the percentage changes in the feature values for the lagged features were included in the pivot transformation.

| Date | Id | Features |
|---|---|---|
| 30.9 | A | $x_A^0$ |
| 31.8 | A | $x_A^{-1}$ |
| 31.7 | A | $x_A^{-2}$ |
| 30.6 | A | $x_A^{-3}$ |
| 31.5 | B | $x_B^0$ |
| 31.3 | B | $x_B^{-2}$ |
| 28.2 | B | $x_B^{-3}$ |
| 31.7 | C | $x_C^0$ |

(A) Before pivot transformation.

| Date | Id | Current | Lag 1 | Lag 2 |
|---|---|---|---|---|
| 30.9 | A | $x_A^0$ | $x_A^{-1}$ | $x_A^{-3}$ |
| 31.5 | B | $x_B^0$ | $NaN$ | $x_B^{-3}$ |
| 31.7 | C | $x_C^0$ | $NaN$ | $NaN$ |

(B) After pivot transformation.

TABLE 4.1: Illustration of the pivot transformation used on the *mainfile* dataset. $x_j^i$ represents an observation (array of feature values) for customer $j$ at time $i$, where $i = 0$ indicates the last observation present in the data. Observe how customer $A$ has four consecutive rows of data, and thus no NaN values after the transformation, whereas customer $B$ misses an observation at time $T - 1$ and thus has NaN values for the lags of 1.

For the *balancefile*, a different set of aggregation measures was applied. Based on each customer's balance movements over the last 90 days, five new features were generated, where three of them are visualized in Figure 4.2. The new features represent the standard deviation, maximum value, and minimum value over the entire period. The purpose of adding these features is to obtain deeper insights into the customer's economic situation and financial stability. In addition to the information provided by these three features, we wanted to derive a measurement indicating financial distress. Based on the assumption that distressed customers will struggle to remain balance-positive for long periods, a "distress feature" was designed to capture the longest coherent period with a positive balance, constituting our fourth feature. Finally, the fifth and last feature generated based on the *balancefile* dataset captures irregularly large deposits by measuring the difference between the largest and the second-largest jump in the balance. There are two reasons why this feature is assumed to be relevant. First, abnormally large jumps in the balance may indicate loan disbursements from other banks, meaning that the feature can uncover worrying signs in an otherwise positive balance. Secondly, a measure of the stability of the income might provide additional insights into the customer's financial situation. All five features were ultimately joined with the *mainfile*, creating one extensive dataset with one row per customer. The inclusion of the balance features provides a lot of

additional information to the credit scoring models. Furthermore, it incorporates time series data in a way the bank has never done before.

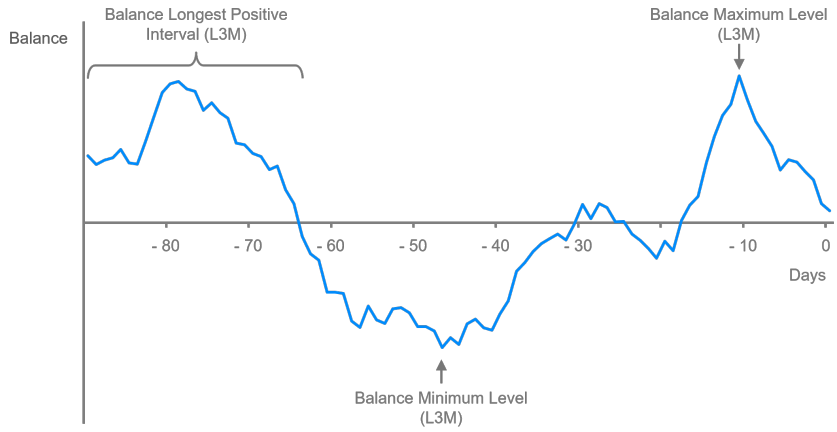



FIGURE 4.2: Illustration of three of the new features generated from the *balancefile*. The last 90 relevant days of account information were used for each customer to create aggregated balance features.

**Feature Selection**

The resulting dataset from the feature extraction procedures contained 13,969 rows, corresponding to one row per customer. Due to the pivot transformations, this dataset entailed more than 100 features. The number of features had to be significantly reduced to make the model more explainable and avoid the curse of dimensionality, i.e., separating the data based on too many features. Features were dropped based on a backward feature selection procedure on a random subset of the data used for training. The procedure is commonly used within the ML literature and can be summarized as follows; Start with all features and iteratively remove features with low SHAP importance on the validation dataset. The resulting dataset that eventually was used for training and evaluating the LightGBM model has 13,969 rows and 18 features. Feature explanations, statistics, and distributions are provided in Appendix B.

## 4.2 Data preparation for Logistic Regression

The bank provided the LR model used in this study, making it an entirely realistic benchmark model. The target variable of the bank's model is the same as the default variable used in ours, i.e., predicting the probability of a legal default occurring within the next 12 months. The bank's LR model utilizes six features, where each feature is split into several bins. Consequently, we created an LR-specific dataset based on a recipe from the bank that was one-hot-encoded to match the categories defined by the bins. To ensure that the LR model and the corresponding dataset complied with the assumptions behind Logistic Regression, we used Variance Inflation Factors (VIF) to verify an acceptable level of multicollinearity among the features and the Box-Tidwell test to check for linearity in log-odds. As the bank has deemed its LR model a trade secret, we are precluded from disclosing further details of its inner workings or details of the performed binning operations.

## 4.3 Data visualization

The following subsections present data visualization techniques on the dataset used for the LightGBM model. First, Principal Component Analysis is conducted to look for any clear linear separation of the dataset. Second, kernel density estimation is performed and visualized through violin plots, to better indicate the feature distributions. Finally, correlation heatmaps are presented to look for any patterns between the features. Data visualizations of the data used for the LR model are found in Appendix C.

### 4.3.1 LightGBM dataset

#### Principal Component Analysis

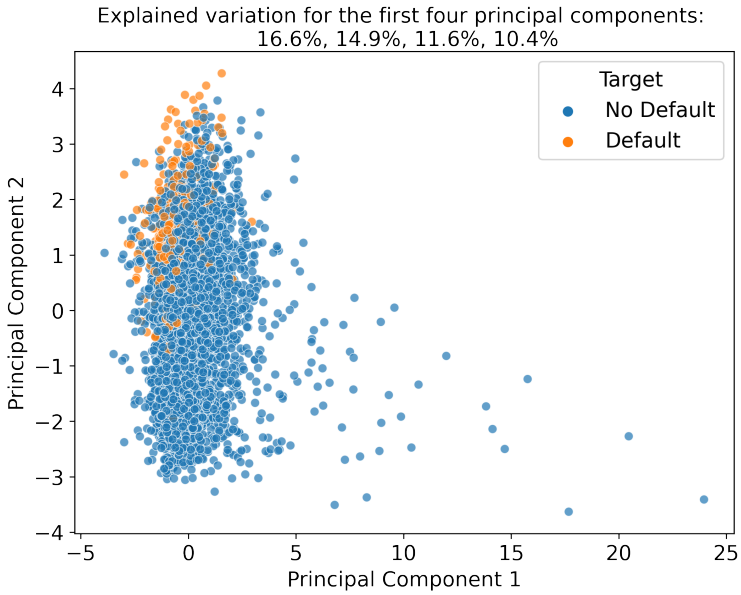



FIGURE 4.3: Principal Component Analysis on the dataset used for the LightGBM model. Each instance is normalized and projected onto the space spanned out by the most dominant eigenvectors. Each instance is color-coded based on the target class.

Figure 4.3 displays the resulting plot after performing Principal Component Analysis (PCA) on the training dataset used by the LightGBM model (Jolliffe, 1986). PCA projects high-dimensional data down to two dimensions using the most dominant eigenvalues and their corresponding eigenvectors. In the plot, each dot represents one instance in the data set and is colored based on the target class. The two axes are the two largest principal components, which represent the two directions in the dataset with the most variance.

It is evident from the figure that no clear separation of the target variable exists, indicating that utilizing a vanilla linear data-separation model without further data transformations would yield poor results. Furthermore, the two largest principal components only explain approximately 31% of the variance in the dataset. This lack of importance, combined with the poor separation, indicates that a model with the flexibility to handle non-linear correlations, such as LightGBM, is preferred.

**Violin Plot**

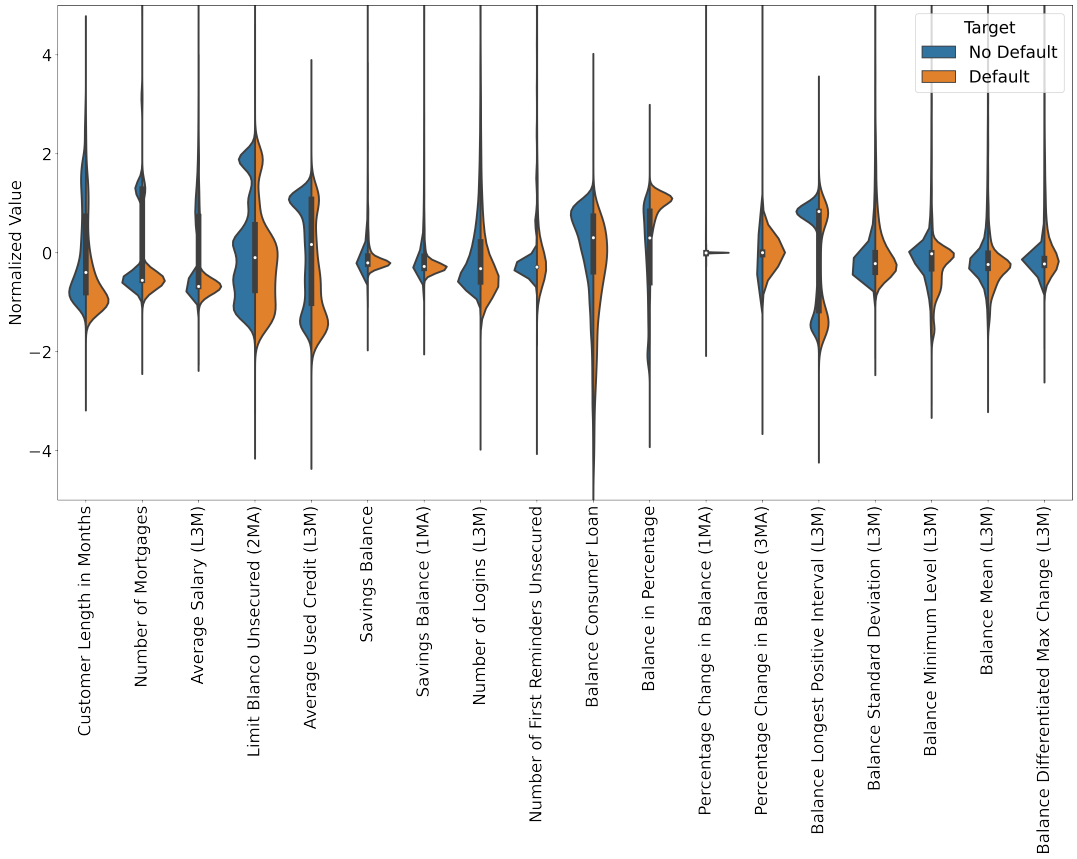



FIGURE 4.4: Violin plot of the data set used for the LightGBM model, with normalized values. Each violin indicates a feature distribution and is colored based on the target class. White dots indicate the feature median, and the black bar indicates the interquartile range. Outliers with an absolute standard deviation larger than 5 are removed for visualization purposes.

Figure 4.4 displays a violin plot of the data set used for training the LightGBM model (Hintze and Nelson, 1998). A violin plot combines box plots and kernel density plots by estimating the underlying distribution of each feature. Thus, violin plots are suitable for displaying feature characteristics efficiently. In the figure, each white dot represents the feature median, whereas the width of each violin indicates the frequency of data points. The black bar of each violin indicates the interquartile range. Each violin is colored based on the target variable.

From the violin plot, it is evident that most of the features are, to some extent, concentrated around their means. Notable differences in the feature distributions for the two target classes are also present, indicating a signal in the data with the potential to separate these classes. For all features, the tails of the feature distributions are thin, represented in the plot as the upper and lower thin lines. The feature *Percentage Change in Balance (1MA)* stands out, with almost all of the data points concentrated around 0.0. The abnormal shape of this violin is caused by a few outliers with significant percentage changes. The reader is referred to Appendix B, where feature statistics are presented through the usage of quantiles.

### Correlation heatmap

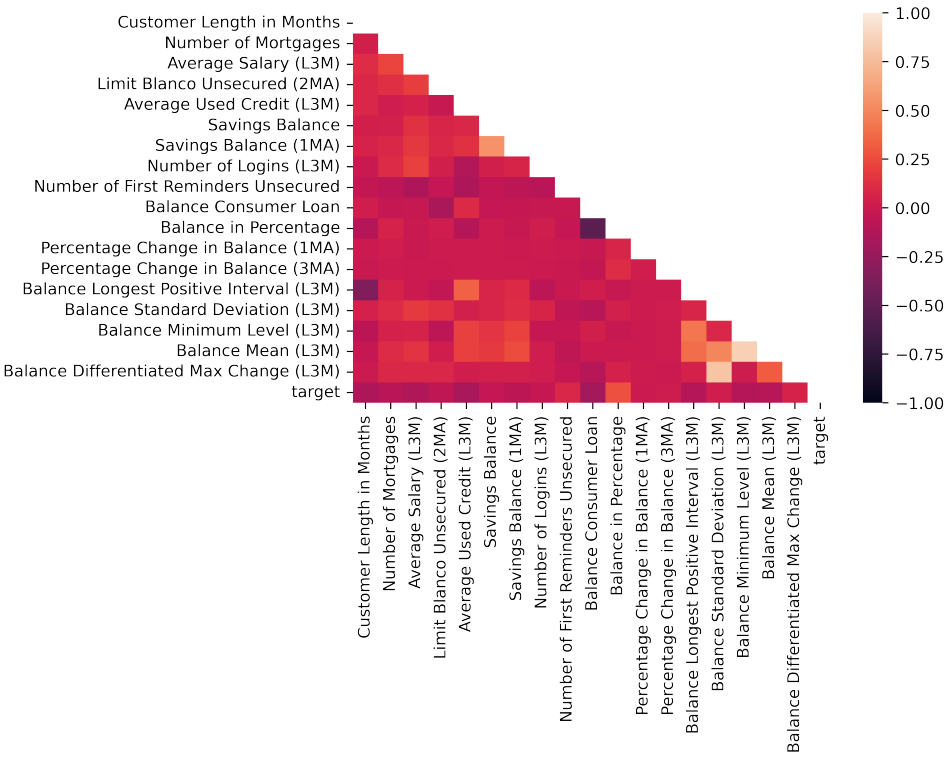



FIGURE 4.5: Correlation heatmap of the dataset used by the LightGBM model. The feature combinations are color-coded based on correlation, explained by the color scale to the right. Light colors indicate positive correlations.

Figure 4.5 shows the linear correlation between all features, including the target variable. The colors shown on the right-hand axis indicate the magnitude of the correlation. From the plot, it is clear that the target variable does not display any significant correlation with the features, and that most of the features are only weakly correlated with themselves. A few stronger correlations exists however, most notably between two pairs of balance-features; *Balance Mean (L3M)* with *Balance Minimum Level (L3M)*, and *Balance Differentiated Max Change (L3M* with *Balance Standard Deviation (L3M)*. It is quite expected that a pair of features related to the balance level and another pair related to the volatility display strong correlations. Over three months, if the average balance is high, the minimum balance level is often high. Conversely, if the balance standard deviation is high, the largest differentiated balance change tends to be high. These pairs of strong correlations could indicate that consumers behave relatively steadily over three months. Since these correlations are so strong, it would be difficult to include all of these features in a Logistic Regression model without violating the assumption of independent variables. However, for LightGBM, correlated features are less of an issue.

# Chapter 5

# Results

In this chapter, we present and discuss the findings from three different areas. First, we evaluate the performance of the different models described in chapter 3. Secondly, we demonstrate how SHAP can be applied to improve the interpretability and reliability of state-of-the-art ML models. Lastly, the economic value of allowing advanced ML models for credit scoring in banks is examined. We focus on how ML models such as LightGBM can advance credit scoring by enabling the models to process more extensive datasets. However, to verify the predictive advantage of LightGBM compared to LR, a second, scaled-down LightGBM model is fitted solely on the features used in the LR model.

Before the models were developed, the data was split into a training set and a test set using stratified sampling. Stratified sampling ensures that the training set and test set have approximately the same proportion of the target class as the original data set. Due to the high amount of signal in the data, and to prevent overfitting, we decided to use a bigger test set than conventional. The test set contained 40% of the original data, corresponding to 5,587 customers. Furthermore, it was completely held out during the development phase as a further measure to prevent overfitting and ensure the validity of the results. Class distributions for the training and test set are found in Appendix B. Stratified k-fold cross-validation was further used to optimize the training on the training set. With this method, the training data is partitioned into k-folds, each fold having approximately the same target distribution. Then, for each fold, a model is trained on the remaining training data and evaluated using the held-out fold. In this study, $k = 10$ folds were found to be the optimal parameter. Thus, during training, 10 sub-models were trained. The resulting final predictions are the mean of the predictions over all ten folds.

For the LightGBM model, the Neptune-Optuna client (Niedzwiedz, 2022) was used to perform hyperparameter searches. Several intervals in the hyperparameter space were defined and narrowed down iteratively on each trial. The resulting hyperparameters from the best performing trial, used in the final model, are found in Appendix D.

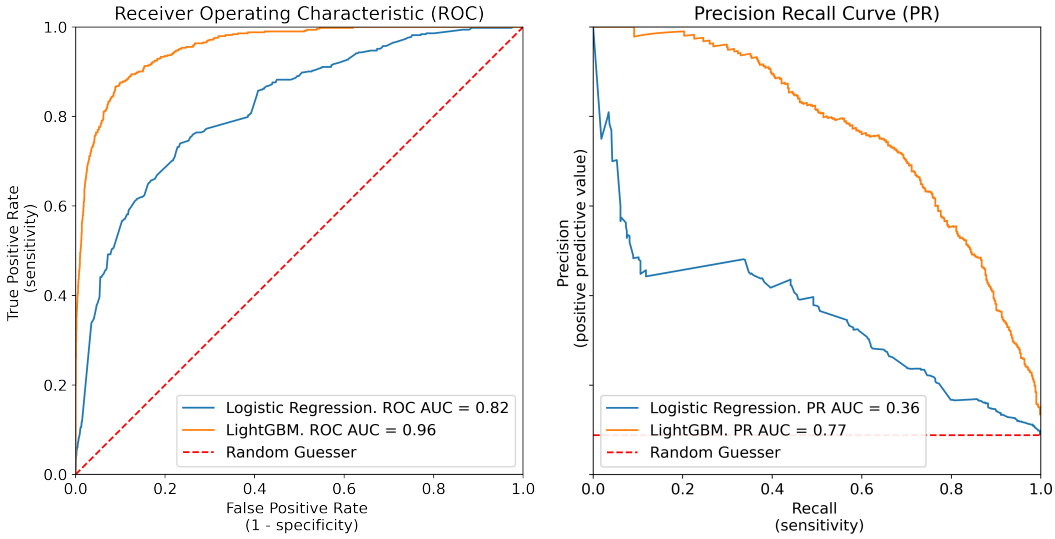
The performance of the models was evaluated using ROC- and PR curves, with their corresponding area under the curve (AUC) values. One of the advantages of using these two evaluation metrics is that they are not constrained to thresholds for classifying default or not default. Hence, ROC AUC and PR AUC provide an aggregated performance measure across all possible classification thresholds. As this study focuses on XAI and explainable credit scoring models, these evaluation metrics were deemed appropriate for assessing the performance of the models. The metrics are further explained in Appendix E.

## 5.1 Model Evaluation

| | | LightGBM | | Logistic Regression | |
|---|---|---|---|---|---|
| | **Actual** | *Positive* | *Negative* | *Positive* | *Negative* |
| | **Predicted** | | | | |
| *Threshold = 10%* | *Positive* | 467 | 1,170 | 464 | 3,255 |
| | *Negative* | 25 | 3,926 | 28 | 1,841 |
| *Threshold = 15%* | *Positive* | 455 | 927 | 438 | 2,524 |
| | *Negative* | 37 | 4,169 | 54 | 2,572 |

TABLE 5.1: Confusion matrix for different thresholds for the LightGBM and Logistic Regression models.

Table 5.1 shows confusion matrices comparing the performance of the LightGBM model and the Logistic Regression model. For the LightGBM model with a *threshold* of 10%, the value of 467 represents the number of true positives, 1,170 is the number of false positives, 25 is the false negatives, whereas 3,926 represents the true negative values. Note that the table includes the thresholds 10% and 15%. Using a probability of default (PD) threshold of 10%, means that any customers with PD higher than 10% are classified as defaulting, and any customers with lower or equal PD are classified as not defaulting. Thus, from a practical perspective, lower thresholds correspond to stricter models, as fewer loans are granted. From the table, we can see that at the strictest level (*threshold = 10%*), the LightGBM model is able to capture more customers subject to default yet still achieves a higher precision (fewer false positives).




(A) ROC plot (B) PR plot

FIGURE 5.1: Evaluation curves. (a) ROC plot and (b) PR plot comparing the performance of the LightGBM and LR model. It is clear that LightGBM outperforms LR with a 17% and a 114% increase in ROC AUC and PR AUC, respectively.

Figure 5.1 provides the ROC and PR curves for both the LR model and the LightGBM model. Both models perform well measured in ROC AUC, with scores above 0.8, indicating strong predictive capabilities. It is evident from both plots, however, that the LightGBM model outperforms the LR model for all thresholds, with an area under the LightGBM curve (*orange*) of 0.96 compared to 0.82 for the LR model (*blue*). The difference constitutes a 17% improvement in ROC AUC for the LightGBM model.

The findings in Table 5.1 and Figure 5.1 clearly show the advantage of the LightGBM model, as it outperforms the benchmark LR model for all thresholds. Further evaluation metrics, confirming the edge of LightGBM, are summarized in Table 5.2.

**LightGBM model based on LR features**

A second, scaled-down LightGBM model was created to confirm the predictive advantage of LightGBM compared to LR. This model used the same six features as the LR model to make the comparison as realistic as possible. Note that these features were not binned as in the LR model but used directly. The scaled-down LightGBM version still outperformed the LR model, achieving a ROC AUC of 0.89, corresponding to a 9% increase. The results of this model is shown in Table 5.2 as LightGBM (LR). Further comparison figures, such as ROC AUC and PR AUC curves, are found in Appendix F.

| | Metric | LightGBM | LightGBM (LR) | Logistic Regression |
|---|---|---|---|---|
| *Threshold = 10%* | *F1-score* | *43.9%* | *26.0%* | *22.0%* |
| | *Recall* | *94.9%* | *97.0%* | *94.3%* |
| | *Precision* | *28.5%* | *15.0%* | *12.5%* |
| | *Accuracy* | *78.6%* | *51.4%* | *41.2%* |
| *Threshold = 15%* | *F1-score* | *48.6%* | *29.1%* | *25.3%* |
| | *Recall* | *92.5%* | *95.1%* | *89.0%* |
| | *Precision* | *32.9%* | *17.2%* | *14.8%* |
| | *Accuracy* | *82.8%* | *59.1%* | *53.9%* |

TABLE 5.2: Metrics for the three models that were trained. LightGBM was trained with 18 features, whereas LightGBM (LR) was used for directly comparing Logistic Regression with LightGBM using the same 6 features.

## 5.2 LightGBM explainability

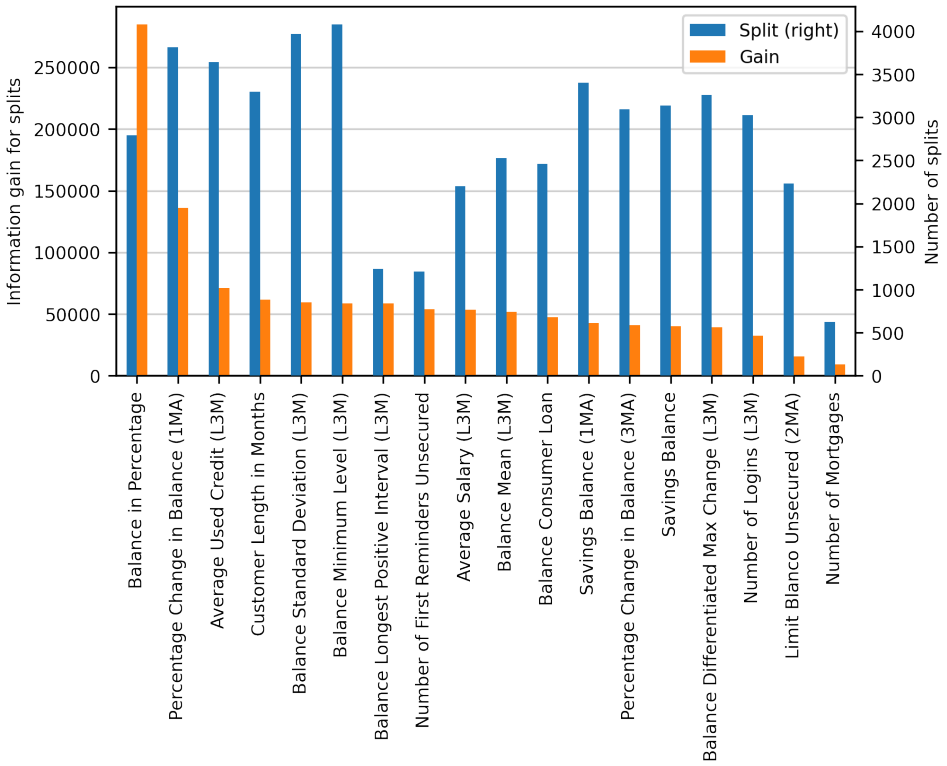



FIGURE 5.2: Feature importance according to the LightGBM model. Average values across the 10 models from the stratified cross-validation, ranked by information gain. The number of splits on the right y-axis, and corresponding information gain for splits on the left y-axis.

Figure 5.2 shows the feature importances in the LightGBM model. Note that in order to obtain the splits and gains for the entire LightGBM model, we averaged the feature importances across the ten individual models resulting from cross-validation. In the plot, blue bars indicate the total number of splits on each feature, whereas orange bars indicate the total information gains of splits that use the feature. From the plot, it is clear that *Balance in Percentage* and *Percentage Change in Balance (1MA)* are the two features associated with the highest information gain. It can also be observed that *Balance Minimum Level (L3M)*, *Balance Standard Deviation (L3M)*, and *Percentage Change in Balance (1MA)* are the features with the largest number of splits in each node. Besides just being a proxy for average feature effects on the dependent variable, a clear disadvantage of LightGBM explainability plots is the lack of directional feature effects. The high information gain of *Balance in Percentage* indicates that it is an important feature for separating the two classes in the dataset. However, it is impossible to interpret to which extent the feature would impact a given prediction. This information is not provided by the number of splits either, as it only measures the number of times each feature is used in the model. There is clearly a need for bolstering the explainability of LightGBM, and the following section shows how SHAP can be used to bridge the explainability gap.

## 5.3 SHAP explanations

In this section, we apply SHAP to provide further explanations of the workings of the LightGBM model. SHAP values correspond to feature effects, and this section shows how such explanations can solve the main challenges uncovered by EBA, related to the complexity of ML models, as discussed in chapter 1. As outlined in section 3.5, SHAP values are calculated using a conditional expectation function derived from the LightGBM model. However, the 10-fold cross-validation of the LightGBM model complicates the application of SHAP, as SHAP expects one single model as input. In order to overcome this issue, we averaged the SHAP values of the ten individual models, in line with the recommendations of the creator of SHAP, (Lundberg, 2018).

### 5.3.1 Global explanations

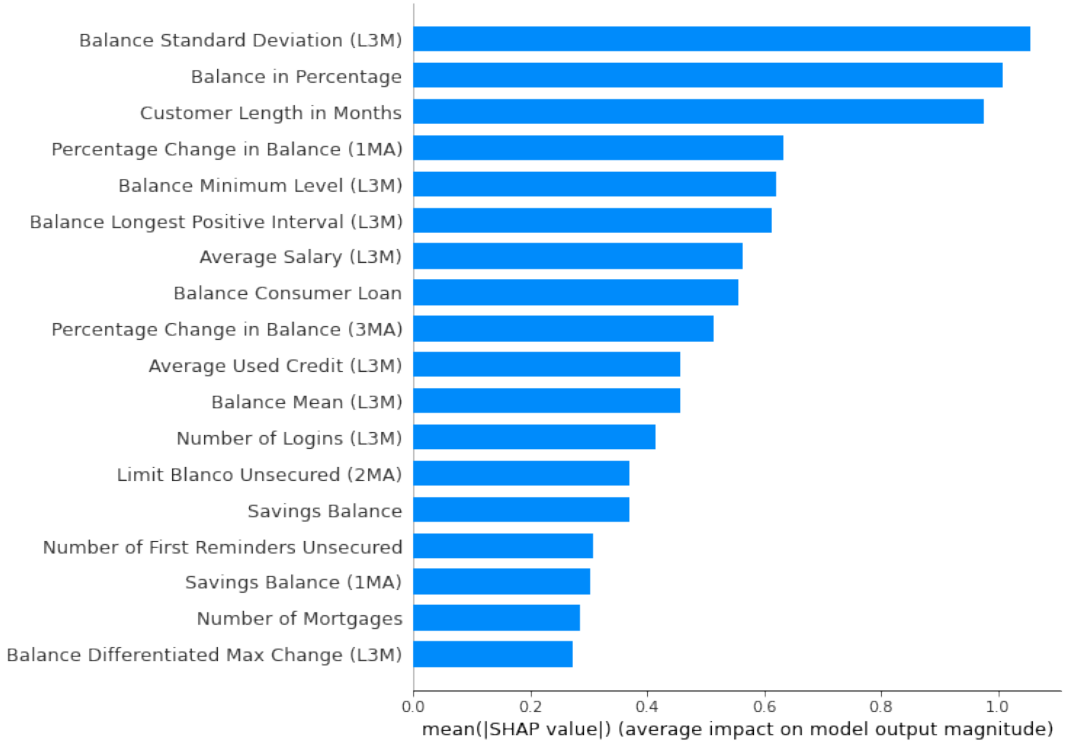



FIGURE 5.3: Simplified SHAP variable importance plot for the LightGBM model, ranked by importance. Note that SHAP values are in absolute *log-odds*.

Figure 5.3 shows the magnitude of the contribution of each feature, measured in absolute log-odds values. We observe that *Balance Standard Deviation (L3M)*, *Balance in Percentage* and *Customer Length in Months* have the highest impact on the model. The feature effects found by SHAP correspond reasonably well with the LightGBM importance plot in Figure 5.2, as many of the same features show significant importance measured in either splits or gain. Furthermore, we can observe that features from the *balance* dataset are of high importance. This clearly indicates that utilizing daily account movements for credit scoring customers gives the model more signal, thus increasing its predictive performance.

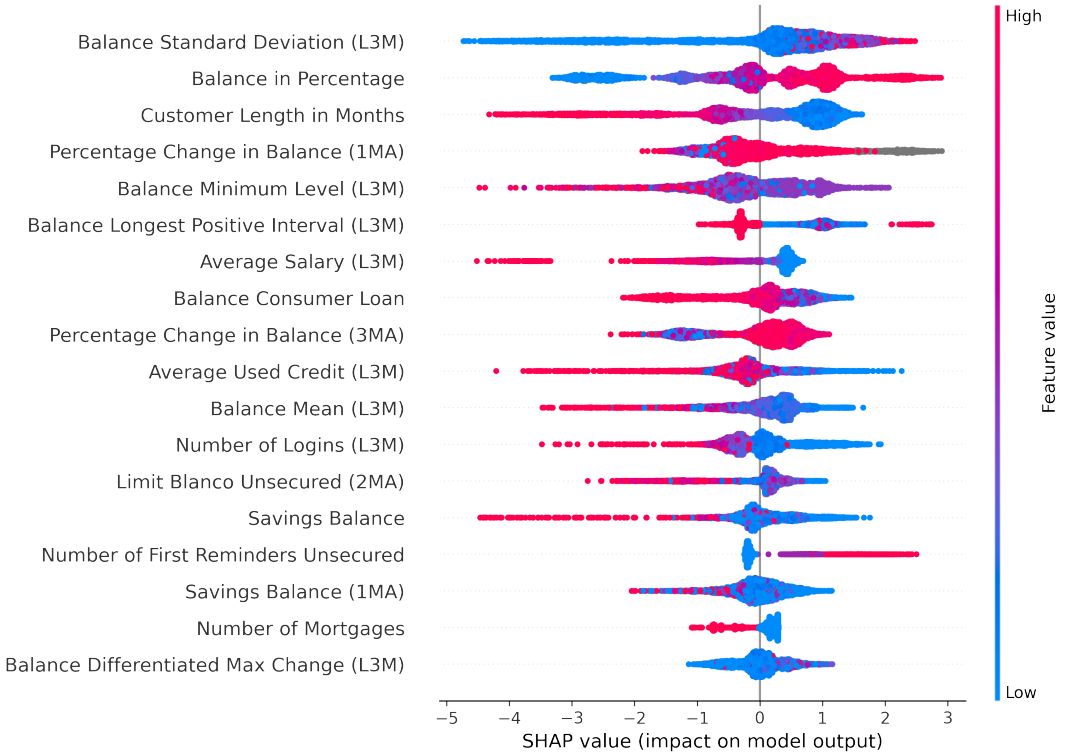



FIGURE 5.4: SHAP variable importance plot for the LightGBM model. Positive SHAP values are associated with an increase in default probability, and feature values are color-coded according to the scale on the right side. E.g., a high level of *Customer Length in Months*, shown in red, is associated with a decrease in default probability, whereas a low feature level, shown in blue, is associated with an increase in the probability of default. Missing values are colored grey, and all SHAP values are measured in log-odds.

Figure 5.4 shows a more detailed summary of the workings of the LightGBM model, where directional feature effects on the resulting predictions are visualized. Each dot on the feature rows represents a single instance in the dataset, distributed on the x-axis according to the SHAP value for that feature value. The high and low relative feature values are color-coded as red and blue, respectively. Missing values are colored grey. High SHAP values are associated with an increase in the predicted probability of default, whereas low SHAP values correspond to a reduction in the predicted probability of default. The features are ranked by importance, with the most important features for the prediction at the top.

Most of the feature rows in Figure 5.4 display a distinct trend in how different feature values affect the model. In other words, one can observe a clear distinction between the red and blue dots, as the high and low feature values contribute in different directions in terms of default probability.

### Dependence plot

Contrary to LR, LightGBM can utilize complex cross-feature relations in its black-box calculation. Though partial dependence plots can display dependencies between a variable and the response, they cannot show how the importance of a variable can vary for specific feature values or display interaction effects between features. SHAP can visualize this by plotting all instances in a scatter plot, with the feature value on the x-axis and the importance measured in SHAP values on the y-axis. By color-coding the values of a second feature, the plot can then display dependencies between variables and how they affect the model. The insights from such dependence plots can provide valuable information for banks and regulators about the inner workings of ML models and thus help evaluate whether the model is in accordance with the regulatory requirements described in chapter 2.

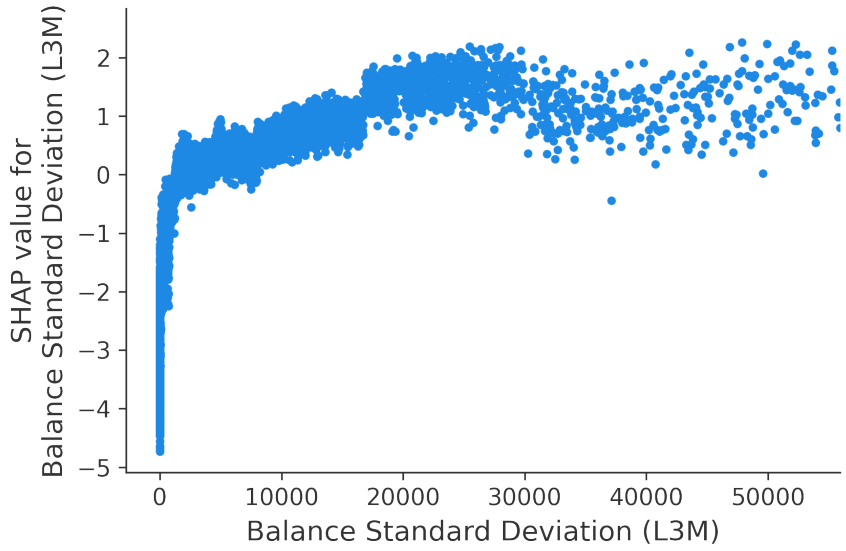



FIGURE 5.5: SHAP dependence plot showing SHAP values for *Balance Standard Deviation (L3M)*, up to the 95th percentile. Each dot represents one customer and positive SHAP values are associated with an increase in default probability.

Figure 5.5 shows how the SHAP values, and thus the feature effects, for *Balance Standard Deviation (L3M)* vary for different feature values. Note that the data is not normalized. The upward trend in the plot indicates that higher volatility in a customer's balance over the last 90 days is associated with an increase in default probability. Standard deviations below approximately 2,000 are associated with negative SHAP values, meaning that these feature values contribute in the direction of non-default. In the range between approximately 2,000 and approximately 30,000, the SHAP values increase steadily, denoting that the importance of the *Balance Standard Deviation (L3M)* feature as an indicator of default increases. Above 30,000, the plot is significantly sparser as few customers experience such high levels of volatility in their balance.

The findings in Figure 5.5 can be interpreted as follows; customers with a stable economic situation, defined by low volatility in their balance, are less likely to default. Conversely, customers with more variation in their balance are far more likely to default on their debt obligations. Seeing that *Balance Standard Deviation (L3M)* is

the feature with the highest average impact on the model, as shown in Figure 5.3, the information in this dependence plot can be important for understanding the predictions of the LGB model.

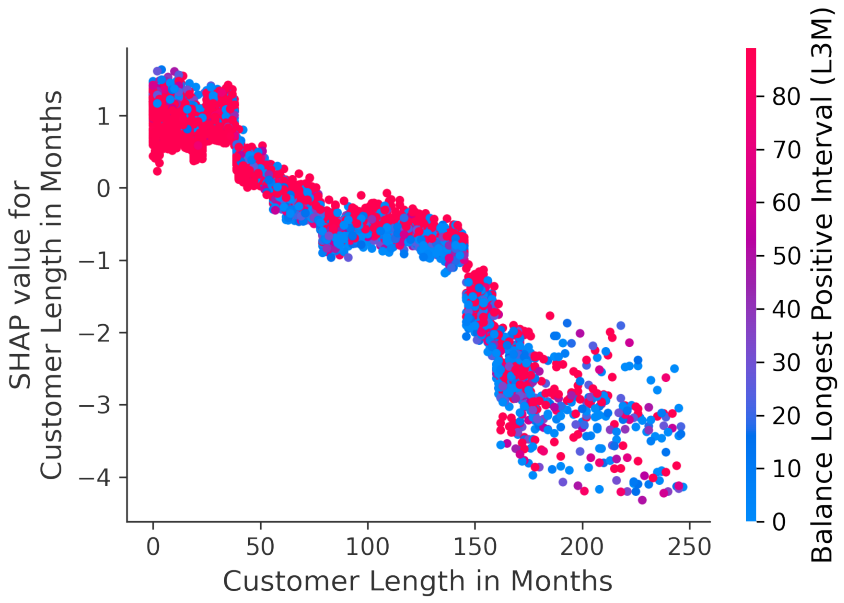



FIGURE 5.6: SHAP dependence plot showing SHAP values for *Customer Length in Months*, color-coded based on the value of the *Balance Longest Positive Interval (L3M)* feature. Each dot represents one customer and positive SHAP values are associated with an increase in default probability.

The SHAP dependence plot can be further extended by color-coding interaction effects between features. This is displayed in Figure 5.6, where the SHAP values of *Customer Length in Months* are displayed and coloured based on the feature values of *Balance Longest Positive Interval (L3M)*. A clear trend is visible, where longer customer relationships with the bank are associated with a lower probability of default. Furthermore, a few larger shifts in the effect of the probability of default are visible. For instance, customer relationships shorter than approximately 4 years (48 months) contribute in the direction of default (positive SHAP values), whereas relationships longer than approximately 12 years (144 months) are very positive in terms of creditworthiness (large negative SHAP values). More mature customers are thus less likely to default on their loans. However, the vertical spread in the plot indicates that other features interact with *Customer Length in Months*. For customer lengths below 150, the vertical separation of the color-coding suggests that *Balance Longest Positive Interval (L3M)* is one among these variables. For customers with longer continuous positive balances (red dots), the feature effect of customer length on default is reduced, as the red dots tend to lie closer to a SHAP value of 0. However, for customers with a shorter continuous positive balance (blue dots), the effect of customer length is more important for default prediction, as the absolute SHAP values are larger. This pattern ends for customer lengths above 150, stipulating that other features have more significant interaction effects with *Customer Length in Months* for these instances.

### Dependence plot with Logistic Regression coefficients

In the following subsection, we offer a novel way of comparing the SHAP feature effects with the feature effects in the LR model. The LR dependencies were derived using the coefficient of the features and mapping the binned values back to the original values to use the same x-axis. The result is visualized with grey dots in the figures. This approach can show where the feature effects differ between the models and provide insights as to why one model outperforms the other.

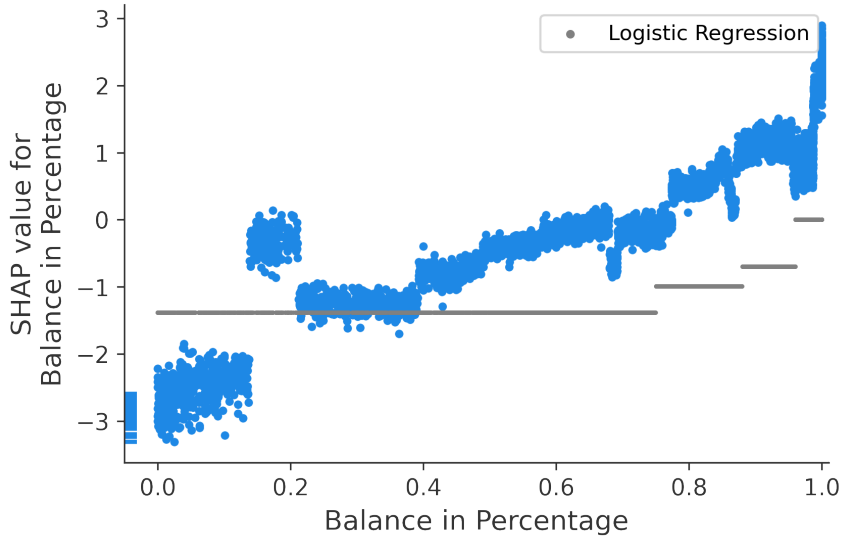



FIGURE 5.7: SHAP dependence plot showing SHAP values for *balance in Percentage*. Logistic Regression feature effects for the balance are displayed in grey and scaled to log-odds, corresponding to the SHAP values on the y-axis. All values from the 4 separate *Balance in Percentage* bins are mapped back to original values to fit the x-axis.

Figure 5.7 combines a SHAP dependency plot with an LR dependency plot for the *Balance in Percentage* feature. This feature measures the percentage share of the issued consumer loan that the customer has outstanding at the time of credit scoring. Specifically, a feature value of 0.0 means the customer has repaid all its debt, while a value of 1.0 implies all of the debt is still outstanding. The leftmost dots in the plot, next to the y-axis, represent missing feature values in the dataset provided by the bank and correspond to the gray dots in Figure 5.4.

Both the SHAP and LR graphs display an upward-sloping trend, where higher feature values are associated with a higher probability of default. Comparing the two graphs plotted in Figure 5.7, one can observe that the magnitude of the SHAP feature effects is greater than the LR model's feature effects on both ends of the x-axis (*further away from* $y = 0$). Thus, the LR model appears to underestimate the effects of the *Balance in Percentage* feature, though it is able to capture the overall trend of the effects. For instance, on the one hand, the LR model assigns the same negative contribution for all feature values below approximately 0.75. The straight grey LR line shows this. On the other hand, the LightGBM model is able to differentiate this group of customers substantially. LightGBM clearly finds segments where the customers with outstanding consumer loans lower than approximately 17% have notable negative SHAP values, indicating significant creditworthiness. The difference between the models is also visible for the largest feature values. For example, for customers with outstanding consumer loans over 95%, LightGBM and SHAP

yield a significantly higher probability of default than the LR model. All in all, the LightGBM model's advantage in its ability to differentiate customers based on the *Balance in Percentage* feature can be a part of the explanation as to why the model significantly outperforms the LR model.

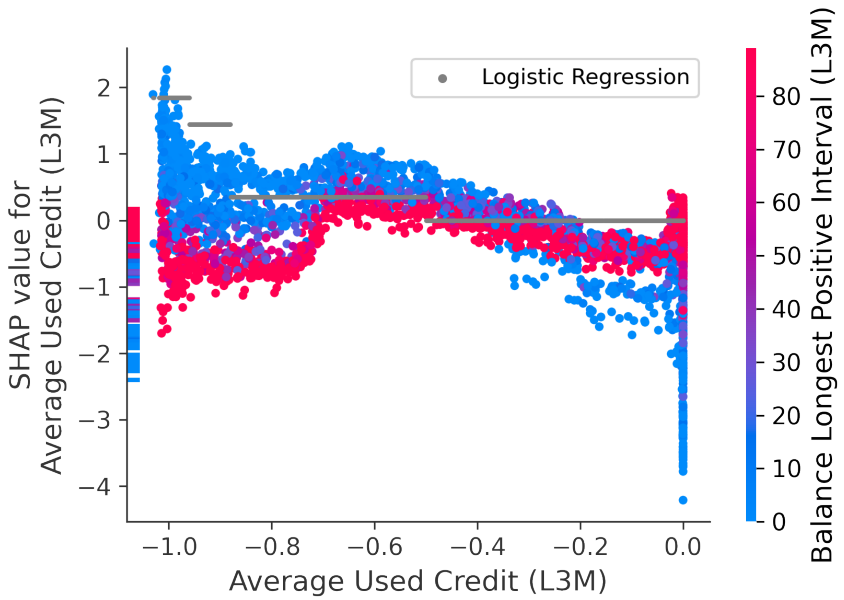



FIGURE 5.8: SHAP dependence plot showing SHAP values for *Average Used Credit (L3M)*, color-coded based on the value of the *Balance Longest Positive Interval (L3M)* feature. Logistic Regression feature effects for average used credit is displayed in grey and scaled to log-odds, corresponding to the SHAP values on the y-axis. All values from the 4 separate *Average Used Credit* bins are mapped back to original values to fit the x-axis.

An example where the feature importances of the LR model coincide better with the SHAP values is shown in Figure 5.8. The figure contains both SHAP and LR dependence plots for *Average Used Credit (L3M)*. The feature measures the average share of the granted credit limit for unsecured products drawn in the last three months. Drawn credit is defined as a negative number and credit limit as a positive number, meaning that feature values of $-1.0$ and $0.0$ indicate that the credit facilities have been fully drawn and remained untouched, respectively.

*Average Used Credit (L3M)* feature values of below $-0.9$ contribute substantially in the direction of default in the LR model. The effects in the LightGBM model are more ambiguous, as the vertical distribution of the SHAP values ranges from $-1$ to $+2$, indicating that other variables might interact with the feature. From the color coding, it is evident that the interaction effect between *Average Used Credit (L3M)* and *Balance Longest Positive Interval (L3M)* can help explain the spread. Among customers that have drawn most of their credit facilities (feature values below approximately $-0.7$), those with a shorter period of continuous positive balance (*blue*) are far more likely to default than those with longer positive stretches (*red*). For the latter, the default probability is actually reduced, meaning that a combination of a low feature value for *Average Used Credit (L3M)* and a high *Balance Longest Positive Interval (L3M)* is an indicator of creditworthiness. The LR model's inability to detect such multidimensional relationships between features makes it fundamentally inferior to advanced ML models, explaining some of the deficit in predictive utility.

### Decision plot

Figure 5.9 can provide further insights into why LightGBM outperforms the LR model. The plot shows the feature effects of 20 instances that go into default. These instances are wrongly classified as non-default by the LR model but correctly classified as default by the LightGBM model. The features on the y-axis are ordered by descending importance, whereas the upper x-axis shows the LightGBM predictions on these instances. The colors of the lines indicate the predicted probability of default; red indicates strong confidence in default, whereas blue indicates lower confidence. Moving from the bottom to the top of the plot, one can observe that each feature's effects on the resulting prediction are added to the intercept. The intercept, represented as the gray vertical line in the plot at 10%, is the chosen cutoff for predicting default versus non-default. The value was selected to be 10%, as it represents the sum of the true proportion in the test class (8.8%), plus a small risk margin.

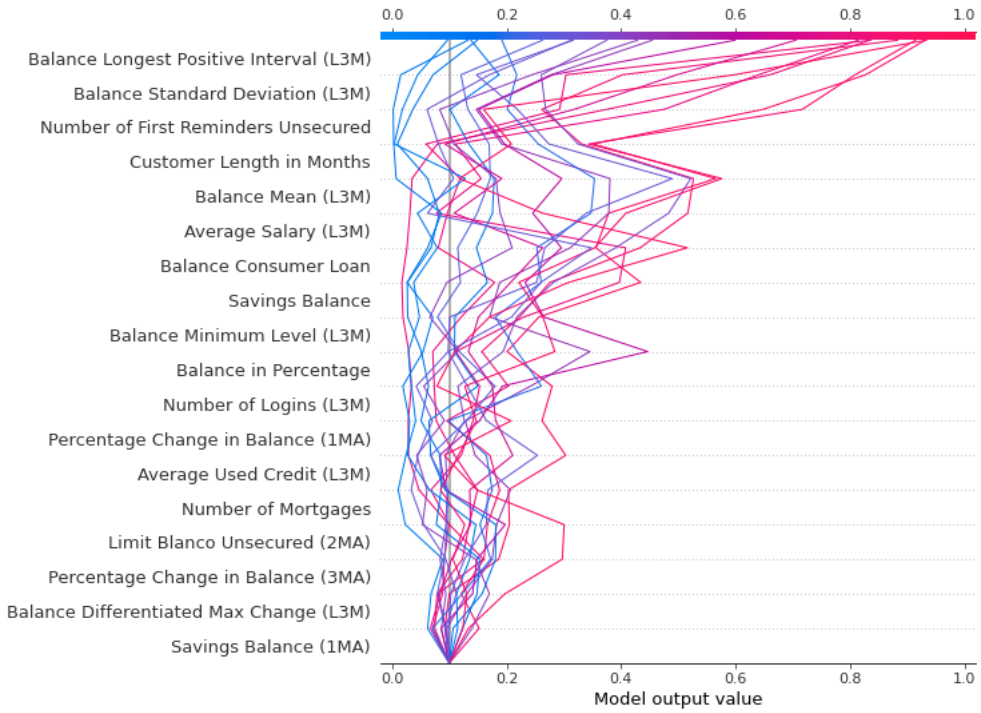



FIGURE 5.9: Decision plot for 20 defaulting observations correctly classified by the LightGBM model but wrongly classified by the LR model. Each line represents one customer. The plot is read bottom-up, starting with the chosen intercept and adding feature effects until each observation's final prediction is reached. Features are ordered based on importance, in descending order.

As we can see from the upper x-axis in Figure 5.9, the model displays a wide range of default probabilities among the 20 customers. The predictions vary from 0.9 to 0.1 and are color-coded accordingly. The SHAP feature effects show that most of the differences in PD are caused by the four most important features, as one can observe a large spike in the predicted probabilities of the red lines for the upper four features. For almost all instances, the balance features *Balance Longest Positive Interval (L3M)*, *Balance Standard Deviation (L3M)* and *Balance Mean* have an elevating effect on the PD. This effect means that these three variables contribute significantly to the LightGBM model's correct default predictions. Since the Logistic Regression model is created without the balance features, this might explain why the LightGBM model correctly classified these instances as default, whereas LR did not.

### 5.3.2 Local explanations

The GDPR demands that all users subject to an automated decision-making process that "significantly affects" them have the right to obtain an explanation for the outcome (European Union, Parliament and Council, 2016). Thus, there is an absolute need for sufficient local explanations of ML credit scoring models. This section demonstrates how SHAP can be used to abide by these regulations.

#### Waterfall plot

SHAP can provide descriptive and intuitive explanations for individual predictions through waterfall plots. The SHAP waterfall plots show the contribution of each feature value to the default prediction, with red and blue bars indicating positive and negative contributions, respectively. The features are ranked by importance, and the actual feature values are displayed on the left side. In Figure 5.10, the prediction by the LightGBM model is displayed as $f(x)$ and the positive bias or intercept from the LightGBM model is expressed below the plot as $E[f(X)]$, both measured in log odds. Note that the difference between the true proportion of the target class and the intercept visualized in the plot is caused by the LightGBM model being slightly underfitted and having a minor bias towards not predicting default. The figure shows a waterfall plot for a customer that LightGBM correctly predicted not to default, while LR falsely predicted to default. By analyzing an instance with conflicting predictions, we can provide insights into the strengths of the LightGBM model and showcase the straightforwardness of the local explanations provided by SHAP. The log-odds $f(x)$ value of $-12.871$, corresponds to a probability of 0.0003%, making the LightGBM model very certain in its non-default prediction.

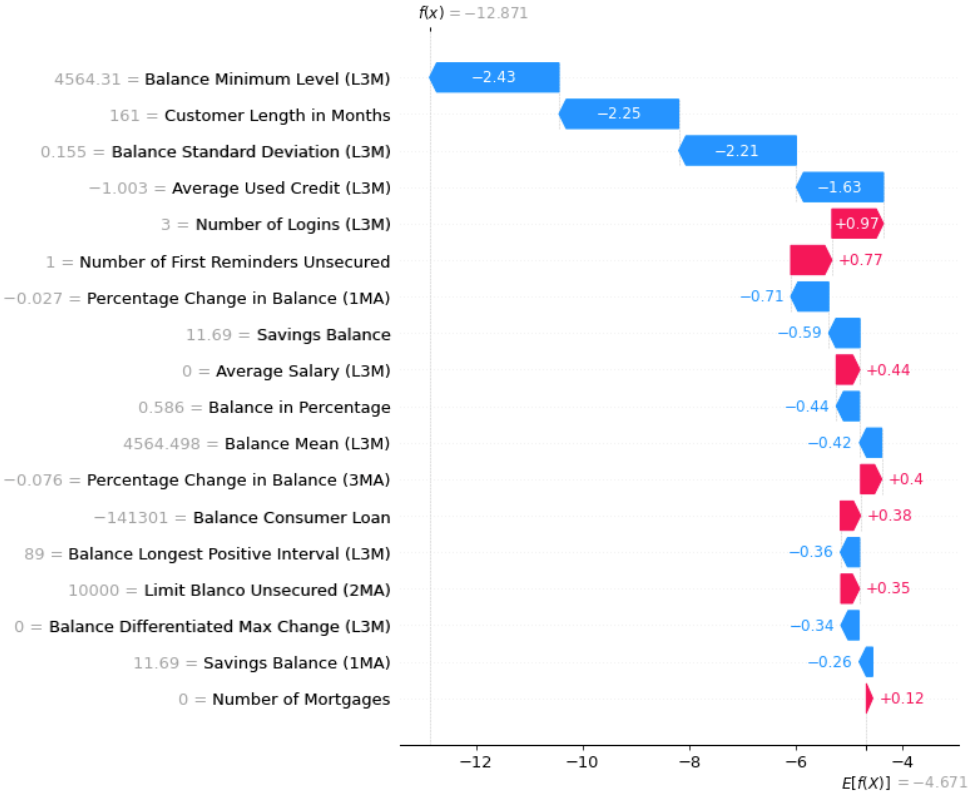



FIGURE 5.10: SHAP waterfall plot visualizing the predicted feature contributions to a non-defaulting customer. All values are in log-odds. The prediction by the model, $f(x)$, corresponds to a probability of 0.0003%. The bias of the LightGBM model is displayed as $E[f(X)]$.

The feature value contributing the most to the non-default prediction is the *Balance Minimum Level (L3M)* of 4,564, showing that constantly having a positive balance is an important indicator of creditworthiness in the LightGBM model. The second most decisive contribution comes from the *Customer Length in Months* value of 161. The negative contribution of −2.25 reflects the findings in Figure 5.6, where customer relationships longer than 144 months displayed a significant reduction in the probability of default. The depicted customer is further a part of the low-risk group with a stable economic situation identified in Figure 5.5, exhibiting a *Balance Standard Deviation (L3M)* as low as 0.155. The last major contributor in the direction of non-default is the *Average Used Credit (L3M)* feature value of −1.003. It is interesting that having over-drawn the credit facilities actually contributes towards a non-default prediction for this customer in the LightGBM model. This differs from the LR model, where Figure 5.8 shows that such low feature values elevate the PD significantly. However, because the customer has a high *Balance Longest Positive Interval (L3M)* value, the LightGBM model considers the combination of these features to have a positive impact on the creditworthiness of the customer. The ability to capture such complex feature interactions showcases one of the important strengths of the LighGBM model.

**Waterfall plot with probabilities**

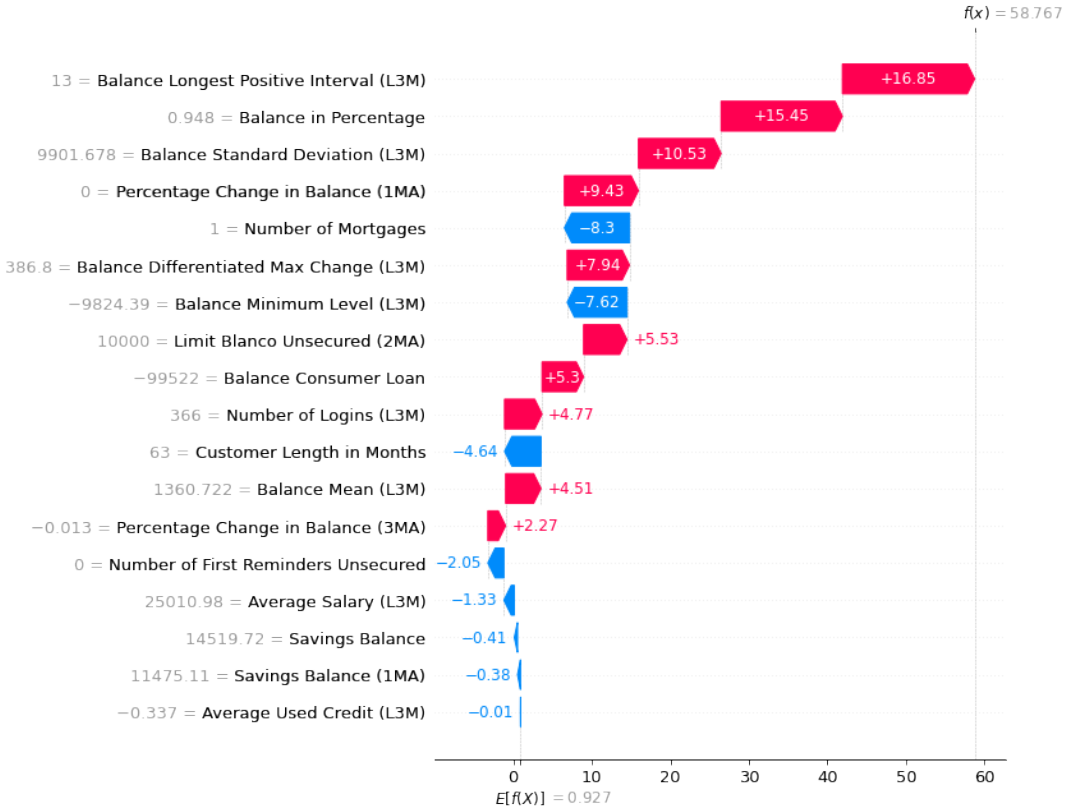



FIGURE 5.11: SHAP waterfall plot with feature effects converted to probabilities. E[f(X)] is the bias of LightGBM, whereas, f(x) is the prediction of LightGBM for this instance. All SHAP values are transformed to probabilities. Feature values shown on the left-hand side.

The explainability of the SHAP waterfall plots can be further improved by converting the feature effects from log-odds to probabilities. The conversion is possible due to the additive property of SHAP values. By normalizing the SHAP values for a

given instance $x_i$ and multiplying the normalized SHAP values with the difference between the prediction and the bias ($f(x_i) - E[f(X)]$), we can convert the SHAP values from log-odds to probabilities while retaining the additive property. However, as outlined in section 3.4, SHAP is just an approximation of exact Shapley values. Thus, the probabilities will not be exact, but the additive property will still be enforced.

An example of SHAP waterfall plots with probabilities is shown in Figure 5.11, where we see that the predicted probability of default, $f(x) = 58.8\%$, equals the sum of the SHAP values in probabilities, and the expected probability of default from the model, $E[f(X)] = 0.9\%$. As mentioned earlier, due to the LightGBM model being slightly underfitted, it exhibits some bias towards predicting non-default, which can explain the difference in the intercept compared to the true target class proportion. However, this difference is distributed over all features, making the individual feature effects relatively accurate.

The instance in Figure 5.11 represents a customer correctly identified as defaulting by the LightGBM model but predicted not to default by the Logistic Regression model. The plot clearly shows that the features from the balance dataset are important; *Balance Longest Positive Interval (L3M)*, *Balance in Percentage* and *Balance Standard Deviation (L3M)* increase the probability of default with approximately 17%, 15%, and 11%, respectively. These features correspond with the most important features found in the decision plot in Figure 5.9. The feature effects of the SHAP waterfall plots with scaled probabilities are intuitive and easy to understand, making them suitable for explaining model outcomes even to non-practitioners.

### 5.3.3 Summary of SHAP explanations

The applications of SHAP that are discussed in this section show that the framework is capable of improving the explainability of LightGBM significantly and providing even more insightful model explanations than the current industry-standard LR models. We argue that the combination of SHAP and LightGBM has the potential to answer the three challenges highlighted by EBA, previously discussed in chapter 1.

- *SHAP can ease the challenge of interpreting results*
  The local explanations provided by waterfall plots show that SHAP provides an intuitive approach for interpreting results. Furthermore, whereas feature effects in LR are always given as the feature values multiplied with the corresponding betas, SHAP displays more flexibility and is more accurate, easing the challenge of interpreting the results.

- *SHAP can facilitate managers' understanding of the credit models.*
  The dependence plots with LR coefficients provide improved comparisons between different credit scoring models, enabling managers to bolster their understanding of the models.

- *SHAP can help to justify a model's results to supervisory authorities*
  Comprehensive global explanations visualizing feature importances, feature dependencies, and interactions between features enable a detailed understanding of the different features' impact on the model output.

Additionally, the local explanations provided by SHAP enable justifications for individual predictions, which is a regulatory requirement imposed by GDPR (European Union, Parliament and Council, 2016).

## 5.4 The economic value of a more accurate model

This section analyzes the potential economic value of the LightGBM model's increased predictive performance compared to the bank's LR model. Credit risk modeling has several areas of use within a bank, where common usage includes evaluating the creditworthiness of new loan applicants and calculating loan loss provisions. We will here focus on the combined gains by improving both. Thus, we find it relevant to analyze the incorrect predictions produced by the two models by identifying false positives and false negatives for both models, using the *held-out test set*.

For this purpose, we created two evaluation metrics - LGD and LP. LGD represents loss-given-default and is the associated loss from customers who received a loan but defaulted (false negatives). This metric was calculated as a proxy by assuming that all remaining balance is lost on default. LP represents lost profits and is the *yearly* alternative cost of not granting loans to non-defaulting customers (false positives). LP was calculated by assuming a flat 10% yearly interest rate on all consumer loans. A separate, 3-dimensional LP plot, that includes changes in interest rates, is provided in Appendix G.

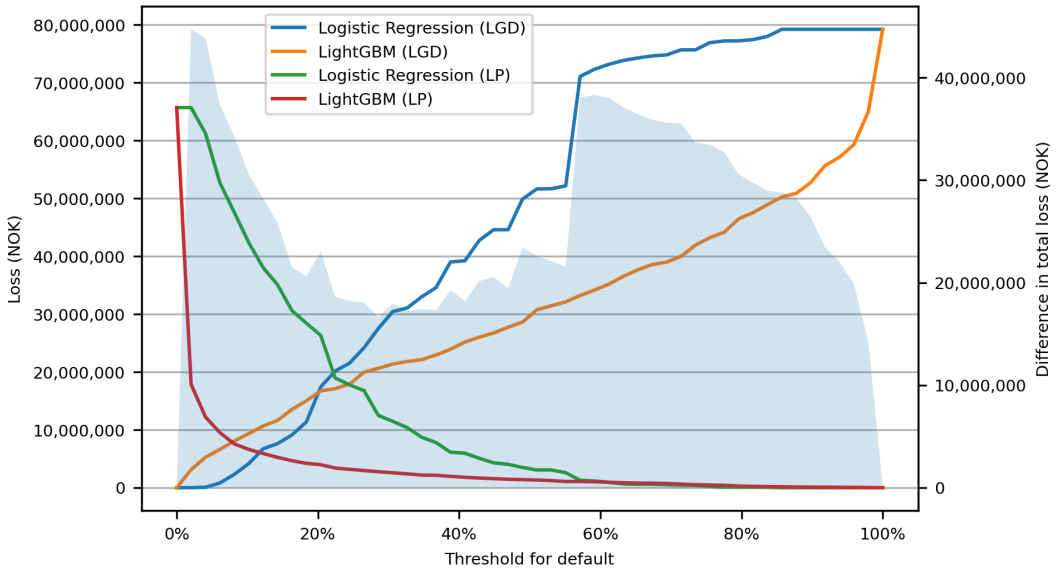



FIGURE 5.12: Approximated costs for imperfect credit scoring models. The X-axis indicates the threshold for default, whereas the left y-axis indicates potential losses. The shaded area displays the difference in total loss between Logistic Regression and LightGBM, and is measured on the right y-axis. *Loss given default* (LGD) and *lost profits* (LP) are plotted, for both LightGBM and Logistic Regression.

Figure 5.12 shows LGD (loss-given-default) and LP (lost-profits) for both models for various probabilities of default thresholds (x-axis). The *left* y-axis indicates the losses for LP and LGD, whereas the *right* y-axis indicates the total loss of having an imperfect model. As we can see from the figure, the losses from the LightGBM model are lower than Logistic Regression for both LP and LGD at almost all thresholds higher than 20%. Below 20%, the losses from the LP of Logistic Regression are significantly larger than those from the LP of LightGBM. Note that this plot assumes that the default threshold is equal for both models. Therefore, it was necessary to calibrate the

LightGBM model by fitting ridge regressors to the one-hot encoded indexes of the leaves produced by the LightGBM model. The calibration was performed solely for this plot in order to keep the comparison between LR and LightGBM as simple as possible. More details about the calibration procedure are presented in Appendix H.

The relative smoothness of the red and orange curves of the LightGBM model compared to the green and blue of the LR model indicates that the potential losses of using the LightGBM model are less sensitive to changes in the threshold for classifying default. The sudden jumps in the LR curves suggest that any minor changes or inaccuracies in the LR model can have significant economic impacts on the bank. Banks typically operate with probability bands for default rather than clear thresholds, and with smoother LP and LGD curves, the model will operate more uniformly within these bands. We argue that the increased smoothness of the LightGBM curves further highlights the advantages of enabling LightGBM for credit scoring.

Despite being an approximated figure, Figure 5.12 indicates that bank losses can be substantially reduced by enabling ML in credit scoring. As the current effective interest rates on consumer loans in Norway typically range between 10% and 25%[1], having an internal threshold of default in the bank somewhere between 10% and 20% seems like a reasonable approximation. At these thresholds, the lost profits of the LightGBM model are significantly lower than for the LR model, with a reduction ranging from about NOK $35,000,000$ at 10% to $22,000,000$ at 20%. This benefit clearly outweighs the slight inferiority of the LightGBM model's LGD for the same thresholds. The total reduced losses for using LightGBM instead of LR, on the *held-out test set*, shown as the shaded area with the y-axis on the right-hand side, are between NOK $30,000,000$ and $20,000,000$ yearly, depending on the threshold for default.

Overall, Figure 5.12 illustrates that the economic value of enabling advanced ML methods for credit scoring in banks can be substantial. As seen in the previous sections, specifically in section 5.1 and section 5.3.1, the difference in predictive ability is likely caused by the inherent ability of ML models to capture non-linear relationships in the data efficiently.

[1] Finansportalen.no - a service by the Norwegian Consumer Council. Loan amount: 100,000 NOK, period of repayment: 1 year (Accessed 01.06.2022)

# Chapter 6

# Conclusion

In this paper, we have shown that LightGBM models outperform LR models for credit scoring in terms of both predictive performance and explainability, and that the economic value of the predictive improvement can be substantial. Three models were utilized and compared in this paper; a full-scale LightGBM model, which utilizes daily multivariate time-series data from the balance accounts of the bank's customers, a Logistic Regression model that was received from the bank and used as a benchmark, as well as a second LightGBM model trained on the same features as the LR model. The full-scale LightGBM model achieved a ROC AUC of 0.96, corresponding to a 17% improvement compared to the benchmark LR model, confirming our hypothesis that using daily balance data improves the predictive performance. Furthermore, the performance of the second LightGBM model, achieving a 9% increase in ROC AUC compared to the LR model, shows that LightGBM is more accurate than LR on the same dataset.

Our main contribution is the application of the explainable AI framework SHAP, where we utilize SHAP values for both global explanations of ML models and local explanations of individual predictions. We have shown how this framework can be applied to improve the interpretability and reliability of state-of-the-art ML models and, specifically, how SHAP can be used to meet the challenges outlined by the European Banking Association concerning AI. Lastly, our work highlights the potential economic value of allowing banks to utilize advanced ML models for credit scoring.

Improving the performance and the explainability of credit scoring models should have positive implications for multiple stakeholders. First, banks would be better equipped to manage their risk and, consequently, reduce their losses. Second, financial authorities would be provided with a more intuitive and detailed tool to interpret the credit models' underlying mechanisms. Finally, increased explainability can improve customers' trust in the credit scoring systems by providing detailed reasoning for customers whose loan applications are rejected.

We identify three potential future improvements to this study that we view as crucial steps on the path toward enabling XAI for credit scoring. First, different tree-based models with XAI should be evaluated on the dataset. Second, the Logistic Regression model used for comparing these models should be open-sourced to enable a more comprehensive comparison with LightGBM. Third, more research on the calibration of LightGBM models should be conducted. Our *uncalibrated* LightGBM models clearly outperform the LR model from the bank, and we believe there is unrealized potential for advancing default predictions further by calibrating the LightGBM model.

# Bibliography


Ariza-Garzón, Miller Janny et al. (2020). "Explainability of a Machine Learning Granting Scoring Model in Peer-to-Peer Lending". In: *IEEE Access* 8, pp. 64873–64890. DOI: 10.1109/ACCESS.2020.2984412.

Bartlett, Peter et al. (1998). "Boosting the margin: a new explanation for the effectiveness of voting methods". In: *The Annals of Statistics* 26.5, pp. 1651 –1686. DOI: 10.1214/aos/1024691352. URL: https://doi.org/10.1214/aos/1024691352.

Basel Committee on Banking Supervention (2006). *International Convergence of Capital Measurement and Capital Standards*. English. URL: https://www.bis.org/publ/bcbs128.pdf.

Bastos, João A and Sara M Matos (2022). "Explainable models of credit losses". eng. In: *European journal of operational research* 301.1, pp. 386–394. ISSN: 0377-2217.

Benhamou, E. et al. (2021). "Explainable AI (XAI) models applied to planning in financial markets". In: DOI: https://dx.doi.org/10.2139/ssrn.3862437.

Bibal, Adrien et al. (2021). "Legal requirements on explainability in machine learning". In: *Artificial Intelligence and Law* 29, pp. 149–169. DOI: 10.1007/s10506-020-09270-4.

Breiman, Leo (1998). "Arcing classifier (with discussion and a rejoinder by the author)". In: *Ann. Statist.* 26.3, pp. 801–849. DOI: https://doi.org/10.1214/aos/1024691079.

Brown, Iain and Christophe Mues (2012). "An experimental comparison of classification algorithms for imbalanced credit scoring data sets". In: *Expert systems with applications* 39.3, pp. 3446–3453. ISSN: 0957-4174.

Bussmann, Niklas et al. (Apr. 2020). "Explainable AI in Fintech Risk Management". In: *Frontiers in Artificial Intelligence* 3. DOI: 10.3389/frai.2020.00026.

Bücker, Michael et al. (2021). "Transparency, auditability, and explainability of machine learning models in credit scoring". In: *Journal of the Operational Research Society* 0.0, pp. 1–21. DOI: 10.1080/01605682.2021.1922098.

Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on knowledge discovery and data mining*. Vol. 13-17-. KDD '16. ACM, pp. 785–794. ISBN: 1450342329.

Connelly, Lynne (2020). "Logistic regression". In: *Medsurg nursing* 29.5, pp. 353–354. ISSN: 1092-0811.

Davis, Randall et al. (Jan. 2022). "Explainable Machine Learning Models of Consumer Credit Risk". In: DOI: 10.2139/ssrn.4006840.

El-Sappagh, Shaker et al. (Jan. 2021). "A multilayer multimodal detection and prediction model based on explainable artificial intelligence for Alzheimer's disease". In: *Scientific Reports* 11 (1). DOI: 10.1038/s41598-021-82098-3.

European Banking Authority (2021). *EBA Discussion paper on machine learning for IRB models*. English. URL: https://www.eba.europa.eu/sites/default/documents/files/document_library/Publications/Discussions/2022/Discussion%20on%20machine%20learning%20for%20IRB%20models/1023883/Discussion%20paper%20on%20machine%20learning%20for%20IRB%20models.pdf.

European Commission (2021a). *Proposal for a regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts*. English. Accessed May 9, 2022 [Online]. URL: https://eur-lex.europa.eu/resource.html?uri=cellar:e0649735-a372-11eb-9585-01aa75ed71a1.0001.02/DOC_1&format=PDF.

— (2021b). *White Paper On Artificial Intelligence - A European approach to excellence and trust*. English. Accessed May 11, 2022 [Online]. URL: https://ec.europa.eu/info/sites/default/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf.

European Union, Parliament and Council (May 4, 2016). *Official Journal of the European Union*. L 119/41. Volume 59. European Union.

Freund, Yoav and Robert E. Schapire (1995). "A desicion-theoretic generalization of on-line learning and an application to boosting". In: *Computational Learning Theory*. Springer Berlin Heidelberg, pp. 23–37. ISBN: 978-3-540-49195-8.

— (1999). "A Short Introduction to Boosting". In: *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, pp. 1401–1406.

Gramegna, Alex and Paolo Giudici (2021). "SHAP and LIME: An Evaluation of Discriminative Power in Credit Risk". In: *Frontiers in Artificial Intelligence* 4, p. 140. ISSN: 2624-8212. DOI: 10.3389/frai.2021.752558. URL: https://www.frontiersin.org/article/10.3389/frai.2021.752558.

Hess, Aaron S and John R Hess (2019). "Logistic regression". In: *Transfusion (Philadelphia, Pa.)* 59.7, pp. 2197–2198. ISSN: 0041-1132.

Hintze, Jerry L. and Ray D. Nelson (1998). "Violin Plots: A Box Plot-Density Trace Synergism". In: *The American Statistician* 52.2, pp. 181–184. DOI: 10.1080/00031305.1998.10480559.

Jolliffe, I. T. (1986). "Principal Component Analysis and Factor Analysis". In: *Principal Component Analysis*. Springer New York. Chap. 5, pp. 115–128. ISBN: 978-1-4757-1904-8. DOI: 10.1007/978-1-4757-1904-8_7. URL: https://doi.org/10.1007/978-1-4757-1904-8_7.

Ke, Guolin et al. (2017). "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Lever, Jake, Martin Krzywinski, and Naomi Altman (2016). "Logistic regression". In: *Nature methods* 13.7, pp. 541–542. ISSN: 1548-7091.

Lundberg, Scott (2018). *github.com/slundberg/shap/issues/337*. Accessed: 2021-11-27. URL: https://github.com/slundberg/shap/issues/337#issuecomment-441710372.

Lundberg, Scott and Su-In Lee (2017). "A unified approach to interpreting model predictions". In: *CoRR* abs/1705.07874. arXiv: 1705.07874. URL: http://arxiv.org/abs/1705.07874.

Lundberg, Scott M., Gabriel G. Erion, and Su-In Lee (2019). *Consistent Individualized Feature Attribution for Tree Ensembles*. arXiv: 1802.03888 [cs.LG].

Misheva, Branka Hadji et al. (Feb. 2021). "Explainable AI in Credit Risk Management". In: arXiv: 2103.00949 [q-fin.RM].

Molnar, Christoph (2019). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. Chap. 9.6. URL: https://christophm.github.io/interpretable-ml-book/shap.html.

Niedzwiedz, Piotr (2022). *Neptune Optuna Hyperparamet Optimization.* URL: https://docs.neptune.ai/integrations-and-supported-tools/hyperparameter-optimization/optuna.

Nixon, Jeremy et al. (2019). *Measuring Calibration in Deep Learning*. DOI: 10.48550/ARXIV.1904.01685. URL: https://arxiv.org/abs/1904.01685.

Peng, Junfeng et al. (Apr. 2021). "A multilayer multimodal detection and prediction model based on explainable artificial intelligence for Alzheimer's disease". In: *Journal of Medical Systems* 45 (5). DOI: 10.1007/s10916-021-01736-5.

Quinto, Butch (2020). *Next-Generation Machine Learning with Spark : Covers XGBoost, LightGBM, Spark NLP, Distributed Deep Learning with Keras, and More*. 1st ed. 2020. Apress : Imprint: Apress. ISBN: 1-4842-5669-7.

Ribeiro, Marco Túlio, Sameer Singh, and Carlos Guestrin (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *CoRR* abs/1602.04938. arXiv: 1602.04938. URL: http://arxiv.org/abs/1602.04938.

Shapley, L. S. (1953). "Stochastic Games". In: *Proceedings of the National Academy of Sciences* 39.10, pp. 1095–1100. ISSN: 0027-8424. DOI: 10.1073/pnas.39.10.1095. eprint: https://www.pnas.org/content/39/10/1095.full.pdf. URL: https://www.pnas.org/content/39/10/1095.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje (2019). *Learning Important Features Through Propagating Activation Differences*. arXiv: 1704.02685 [cs.CV].

Strumbelj, Erik and Igor Kononenko (2013). "Explaining prediction models and individual predictions with feature contributions". In: *Knowledge and Information Systems* 41, pp. 647–665.

Yang, Yimin and Min Wu (2021). "Explainable Machine Learning for Improving Logistic Regression Models". In: *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, pp. 1–6. DOI: 10.1109/INDIN45523.2021.9557392.

Yoo, Tae Keun et al. (Feb. 2020). "Explainable Machine Learning Approach as a Tool to Understand Factors Used to Select the Refractive Surgery Technique on the Expert Level". In: *Translational Vision Science  Technology* 9 (8). DOI: https://doi.org/10.1167/tvst.9.2.8.

Young, H.P. (1985). "Monotonic solutions of cooperative games." In: *Int J Game Theory* 14. DOI: https://doi.org/10.1007/BF01769885.

Zhang, Huan, Si Si, and Cho-Jui Hsieh (2017). *GPU-acceleration for Large-scale Tree Boosting*. arXiv: 1706.08359 [stat.ML].

# Appendix A

# Logistic Regression Theory

For evaluating the relative performance of the LightGBM model, an industry-standard LR model generously provided by a medium-tier bank in Norway. LR is commonly used to predict categorical values (Lever et al., 2016) and is the most popular method for credit scoring in banks. The LR model from the bank was used in order to make the baseline as realistic as possible. The essential property of LR is that a linear combination of independent variables can be mapped to a probability score (Hess and Hess, 2019), and that the dependent variable can be classified into two groups based on the scores (Bussmann et al., 2020). This section outlines how LR works for estimating probability of default (PD):

Let $Y_n$ be the estimated default probability for customer $n$, based on on the feature values $x_{1n}, ..., x_{Tn}$. PD can then be expressed as:

$$P(Y_n = 1|x_{1n}, ..., x_{Tn}) = p_n \tag{A.1}$$

This probability can be further expressed as an odds ratio, which is an indicator of an association between variables (Connelly, 2020). Odds ratio can be defined as the ratio of the probability of an outcome occurring to the probability of it not occurring (Lever et al., 2016):

$$Odds\ ratio = \frac{p_n}{1 - p_n} \tag{A.2}$$

The linear combination of the independent variables can be expressed as the natural logarithm of the odds ratio. This yields *the logistic regression equation* (Hess and Hess, 2019):

$$\ln(\frac{p_n}{1 - p_n}) = \alpha + \sum_{t=1}^{T} \beta_t x_{nt} \tag{A.3}$$

Where $\alpha$ is the intercept and $\beta_t$ is the $t$'th regression coefficient. These parameters are estimated using MLE. Solving Equation A.3 for $p$ gives a probability function that maps the linear function back to probabilities:

$$p_n = \frac{1}{1 + e^{-(\alpha + \sum_{t=1}^{T} \beta_t x_{nt})}} \tag{A.4}$$

This expression is called *the logistic function* and yields a sigmoid curve, which lies between 0 and 1 for all values of the linear predictor (Lever et al., 2016; Hess and Hess, 2019).

Using Equation A.1 and Equation A.4, the PD can thus be expressed as a logistic function:

$$P(Y_n = 1|x_{1n}, ..., x_{Tn}) = \frac{1}{1 + e^{-(\alpha + \sum_{t=1}^{T} \beta_t x_{nt})}} \tag{A.5}$$

The LR model is non-linear in probabilities and odds (Equation A.4), but linear in log-odds (Equation A.3). Any input variable can be transformed, but the logistic regression equation (Equation A.4) will remain linear. The transformation of the input variables applies to all of the data, meaning that some non-linear relationships between features might be overlooked by the model.

# Appendix B

# Data

## B.1 Features used in the LR model

| Feature Name | Feature Explanation | Type |
|---|---|---|
| *Customer Length in Months* | *Number of months since the customer first joined as a client* | *bin* |
| *Number of Mortgages* | *Number of mortgages at the time of scoring* | *bin* |
| *Average Salary (L3M)* | *Average salary of the customer, last three months* | *bin* |
| *Average Used Credit (L3M)* | *Average used credits by the customer, last three months* | *bin* |
| *Balance in Percentage* | *Balance in Percentage* | *bin* |
| *Grouped Number of Notices* | *Grouping of reminder variables* | *bin* |

TABLE B.1: Explanations of the features used in the LR model.

## B.2 Features used in the LightGBM model

| Feature Name | Feature Explanation | Type |
|---|---|---|
| *Customer Length in Months* | *Number of months since the customer first joined as a client* | *float* |
| *Number of Mortgages* | *Number of mortgages at the time of scoring* | *float* |
| *Average Salary (L3M)* | *Average salary of the customer, last three months* | *float* |
| *Limit Blanco Unsecured (2MA)* | *Limit Blanco two months ago* | *float* |
| *Average Used Credit (L3M)* | *Average used credits by the customer, last three months* | *float* |
| *Savings Balance* | *Sum balance at the time of scoring* | *float* |
| *Savings Balance (1MA)* | *Sum balance one month before scoring* | *float* |
| *Number of Logins (L3M)* | *Number of logins, last three months* | *float* |
| *Number of First Reminders Unsecured* | *Number of first reminders on unsecured loans* | *float* |
| *Balance Consumer Loan* | *Balance of the consumer loan at the time of scoring* | *float* |
| *Balance in Percentage* | *Balance in Percentage* | *float* |
| *Percentage Change in Balance (1MA)* | *Percentage change in balance between one month ago and time of scoring* | *float* |
| *Percentage Change in Balance (3MA)* | *Percentage change in balance between three months ago and time of scoring* | *float* |
| *Balance Longest Positive Interval (L3M)* | *Longest continous period of positive balance, over the last three months* | *float* |
| *Balance Standard Deviation (L3M)* | *Standard deviation of balance, last three months* | *float* |
| *Balance Minimum Level (L3M)* | *The lowest balance level, last three months* | *float* |
| *Balance Mean (L3M)* | *Balance mean, last three months* | *float* |
| *Balance Differentiated Max Change (L3M)* | *The differentiated maximum change in balance, last three months* | *float* |

TABLE B.2: Explanations of the features used in the LightGBM model. A subset of these features was used for the LightGBM (LR) model. This model was used for comparing LR and LightGBM more directly.

## B.3 Feature statistics

| Feature name | Mean | Std. Dev. | Min | 25% | 50% | 75% | Max | Count NaN |
|---|---|---|---|---|---|---|---|---|
| *Customer Length in Months* | *64.4* | *59.1* | *0.0* | *17.0* | *41.0* | *108.0* | *247.0* | *0* |
| *Number of Mortgages* | *0.3* | *0.5* | *0.0* | *0.0* | *0.0* | *1.0* | *5.0* | *1* |
| *Average Salary (L3M)* | *14,905.9* | *21,945.6* | *0.0* | *0.0* | *0.0* | *30,921.7* | *505,969.8* | *4* |
| *Limit Blanco Unsecured (2MA)* | *42,913.6* | *30,217.4* | *0.0* | *20,000.0* | *40,000.0* | *60,000.0* | *150,000.0* | *2,835* |
| *Average Used Credit (L3M)* | *-0.4* | *0.4* | *-1.1* | *-0.8* | *-0.4* | *0.0* | *0.0* | *2,886* |
| *Savings Balance* | *35,722.6* | *155,915.8* | *-1,965.0* | *190.7* | *4,087.1* | *24,913.7* | *6,253,344.9* | *0* |
| *Savings Balance (1MA)* | *30,458.1* | *96,637.5* | *-255.4* | *161.9* | *3,769.1* | *23,371.0* | *3,025,371.2* | *0* |
| *Number of Logins (L3M)* | *68.8* | *93.1* | *0.0* | *14.0* | *39.0* | *89.0* | *1,419.0* | *0* |
| *Number of First Reminders Unsecured* | *0.3* | *1.1* | *0.0* | *0.0* | *0.0* | *0.0* | *15.0* | *0* |
| *Balance Consumer Loan* | *-97,884.9* | *94,672.0* | *-500,000.0* | *-134,114.8* | *-69,094.0* | *-27,986.5* | *-0.4* | *219* |
| *Balance in Percentage* | *0.7* | *0.3* | *0.0* | *0.5* | *0.8* | *0.9* | *1.1* | *219* |
| *Percentage Change in Balance (1MA)* | *0.0* | *2.8* | *-1.0* | *-0.1* | *0.0* | *0.0* | *211.8* | *741* |
| *Percentage Change in Balance (3MA)* | *-0.1* | *2.6* | *-1.0* | *-0.2* | *-0.1* | *0.0* | *201.8* | *1,249* |
| *Balance Longest Positive Interval (L3M)* | *57.3* | *38.0* | *0.0* | *13.0* | *89.0* | *89.0* | *89.0* | *0* |
| *Balance Standard Deviation (L3M)* | *15,078.5* | *34,701.4* | *0.0* | *1,462.4* | *7,503.6* | *15,099.0* | *1,047,771.4* | *0* |
| *Balance Minimum Level (L3M)* | *1,201.2* | *63,817.5* | *-102,880.9* | *-19,393.5* | *0.0* | *1,000.5* | *1,192,793.8* | *0* |
| *Balance Mean (L3M)* | *20,442.1* | *80,739.1* | *-100,326.1* | *-4,901.6* | *1,291.1* | *19,573.4* | *1,277,713.6* | *0* |
| *Balance Differentiated Max Change (L3M)* | *26,426.8* | *103,991.6* | *0.0* | *137.7* | *3,099.6* | *14,141.6* | *3,364,772.6* | *0* |

| Feature name | "No Reminders" | "Reminder 1" | "Reminder 2" |
|---|---|---|---|
| *Grouped Number of Notices*** | *7,077* | *980* | *324* |

*** Categorical feature value counts - feature only used in LR model*

TABLE B.3: Feature statistics for the training set consisting of 8,381 instances (60% of the data). Note that the data is not normalized.

## B.4 Class distributions

| | **Training** | **Test** |
|---|---|---|
| *Size* | 60% (8,381) | 40% (5,588) |
| *Minority class* | 8.82% (739) | 8.80% (492) |

TABLE B.4: Class distribution and size of each dataset, used for all models. Stratified sampling was used to split the datasets evenly.

# Appendix C

# Data Visualization for Logistic Regression

## C.1 Correlation heatmap of LR features

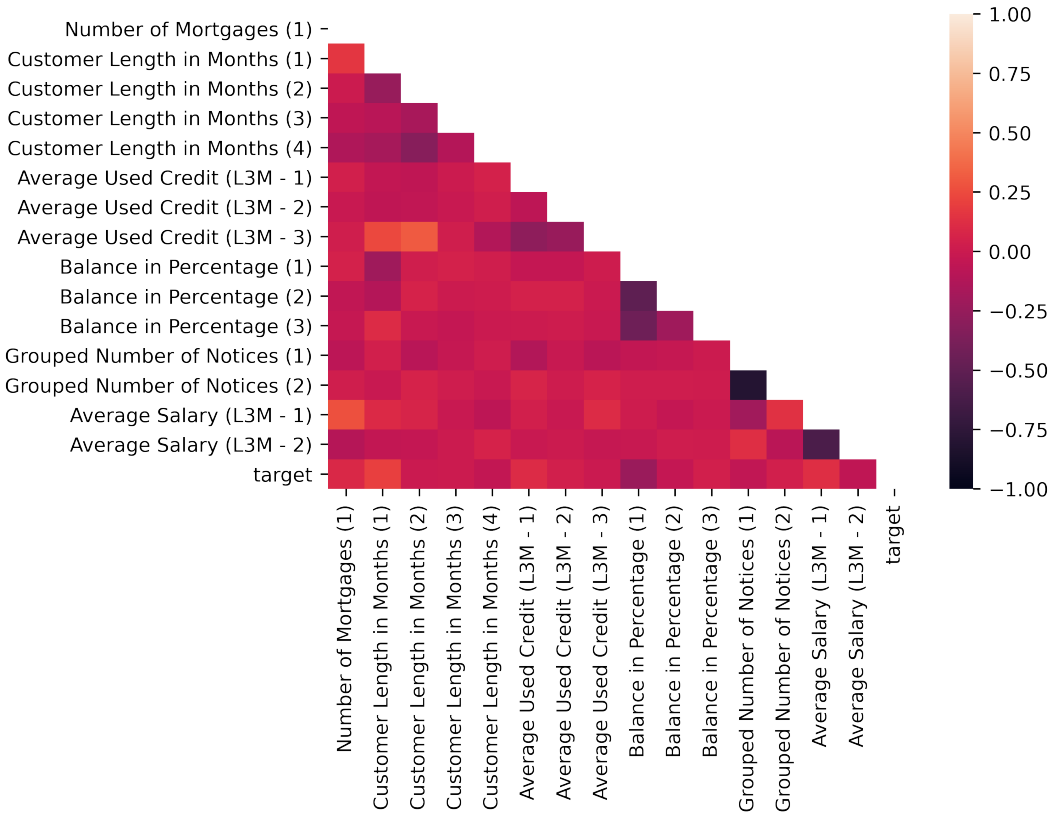



FIGURE C.1: Correlation heatmap for the features used in the LR model. The feature combinations are color coded by correlation, explained by the color scale to the right.

## C.2 Principal component analysis of LR features

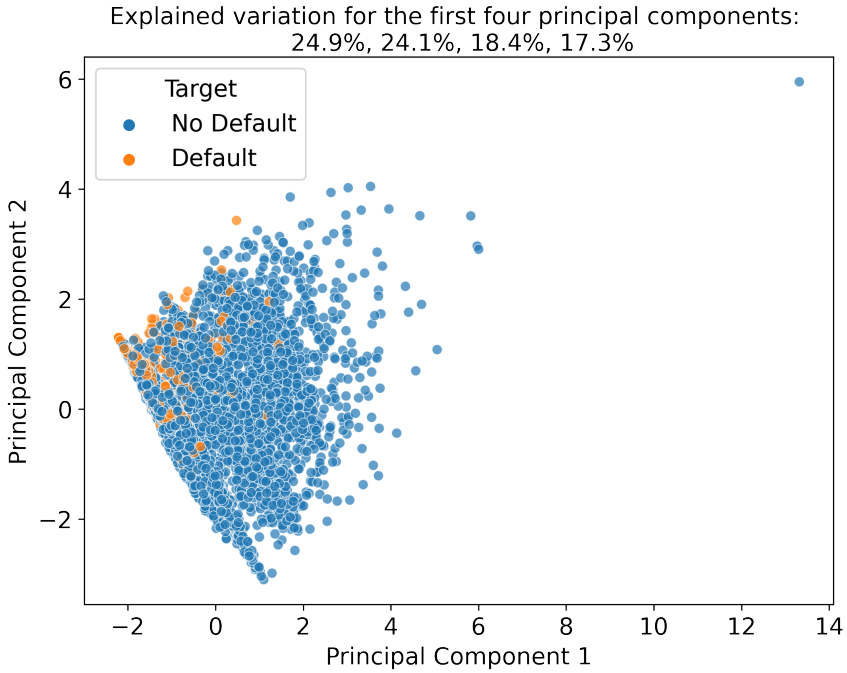



FIGURE C.2: Principal component analysis (PCA) conducted on the Logistic Regression dataset.

## C.3 Violin plot of LR features

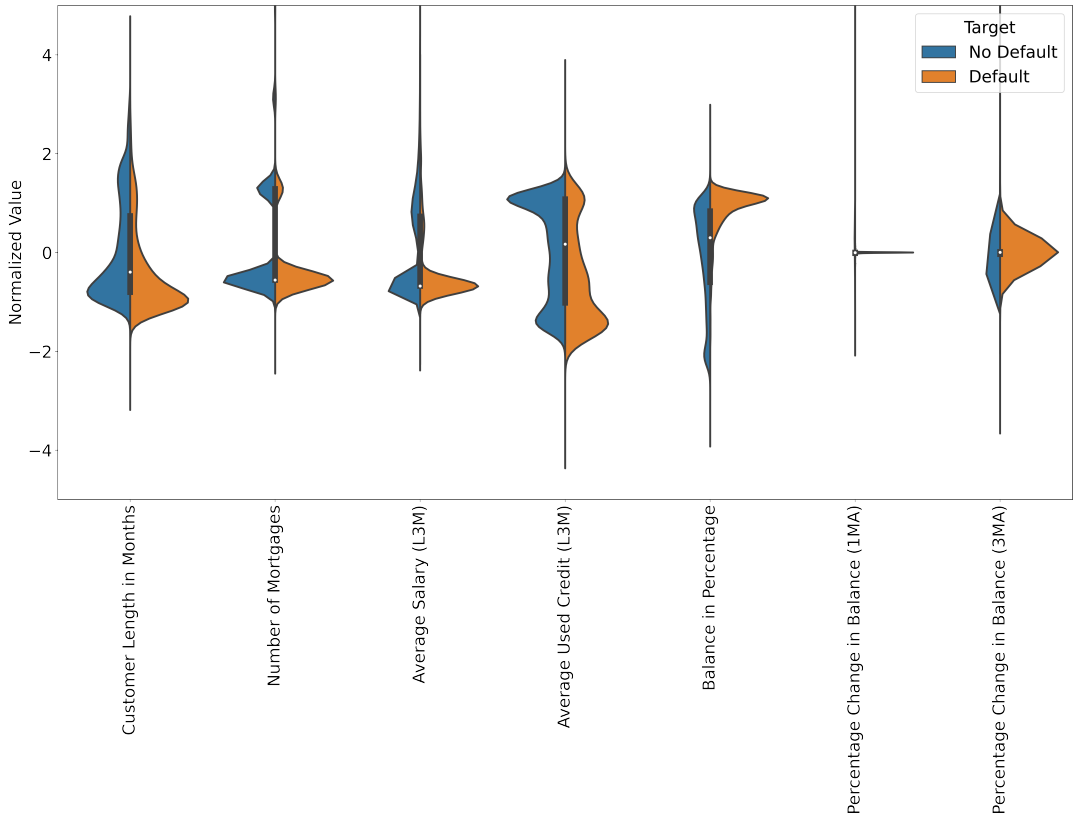



FIGURE C.3: Kernel density estimation on the Logistic Regression dataset visualized through violin plots.

# Appendix D

# Model details

## D.1 Final hyperparameters for LightGBM model

| Hyperparameter | Value |
|---|---|
| *Boosting* | *GBDT* |
| *Metric* | *AUC* |
| *Learning rate* | *0.007* |
| *Scale pos. weight* | *11.5* |
| *Boosting rounds* | *25,000* |
| *Early stopping* | *5,000* |
| *Number of leaves* | *3* |
| *Max bin* | *255* |
| *Min data in leaf* | *1* |
| *Max depth* | *-1* |
| *Number of splits* | *10* |
| *Lambda L1 (Lasso)* | *0.6* |
| *Lambda L2 (Ridge)* | *0.02* |

TABLE D.1: Hyperparameters for the final LightGBM model. The exact same parameters were used on the scaled-down LightGBM (LR) model.

# Appendix E

# ROC and PR evaluation metrics

**Receiver operating characteristic (ROC)** ROC curves plot the true positive rate (TPR), also called recall, on the y-axis against the false positive rate (FPR) on the x-axis for all possible cut-off values:

$$TPR = \frac{TP}{TP + FN} \tag{E.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{E.2}$$

Accurate models are recognized by as high TPR as possible for low FPR values, meaning that a bigger AUC is better.

**Precision recall (PR)** Precision recall curves plot positive predictive value (PPV), also called precision, on the y-axis and recall on the x-axis:

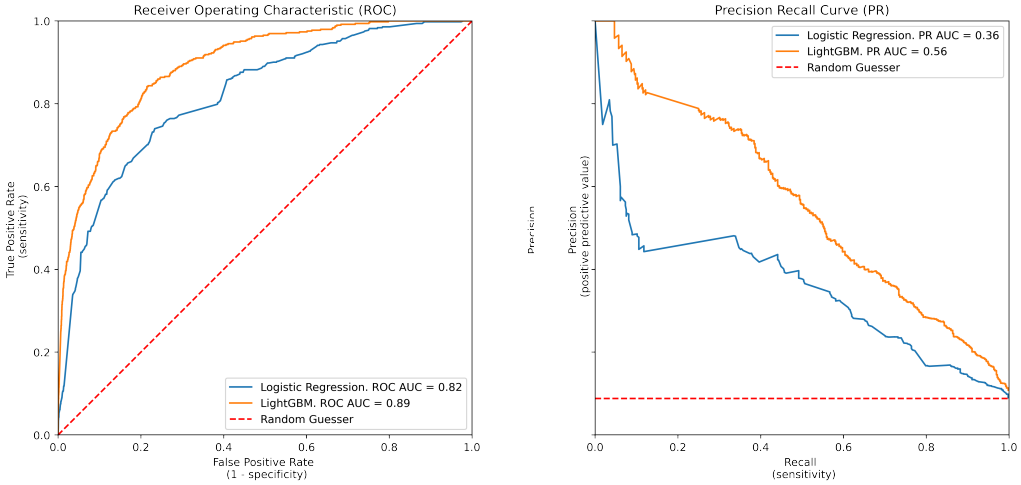
$$PPV = \frac{TP}{TP + FP} \tag{E.3}$$

For imbalanced data sets with smaller positive classes, the most important task of the model is to correctly predict positive cases. The focus on negative predictions are reduced, meaning that the importance of PPV increases. This makes precision recall a valuable measurement for the LightGBM model.

# Appendix F

# Model comparison with same features

## F.1 AUC and PRC curves




(A) ROC plot

(B) PR plot

FIGURE F.1: Evaluation curves. (a) ROC plot and (b) PR plot comparing the performance of the LightGBM and LR models where both models are trained on the same features. Note that the LR variables are binned in order to comply with the LR assumptions, whereas LightGBM are trained on the features directly.

## F.2 SHAP Feature Importance

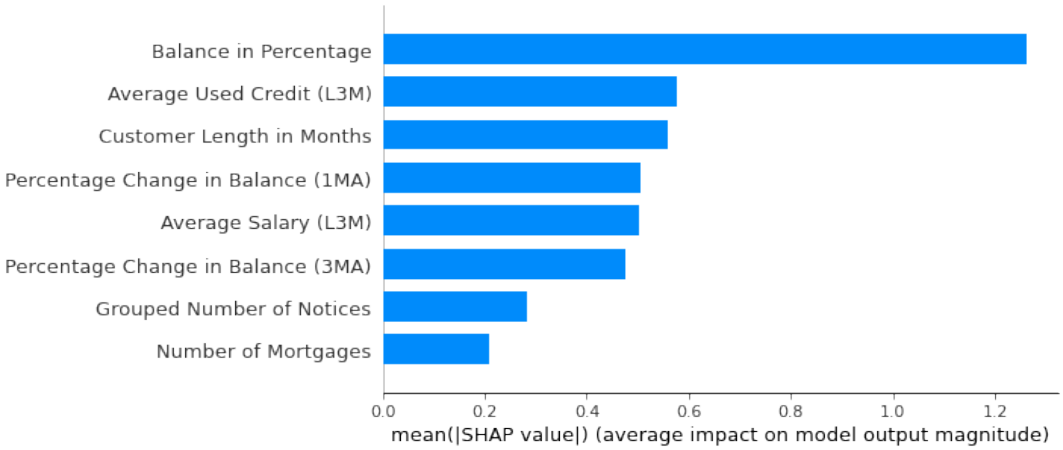



FIGURE F.2: Simplified SHAP variable importance plot for the LightGBM (LR) model ranked by importance. Note that SHAP values are in absolute *log-odds*.

## F.3 Confusion matrices

| | | **LightGBM (LR)** | | **Logistic Regression** | |
|---|---|---|---|---|---|
| | **Actual** | *Positive* | *Negative* | *Positive* | *Negative* |
| | **Predicted** | | | | |
| *Threshold = 10%* | *Positive* | 477 | 2,700 | 464 | 3,255 |
| | *Negative* | 15 | 2,396 | 28 | 1,841 |
| *Threshold = 15%* | *Positive* | 468 | 2,260 | 438 | 2,524 |
| | *Negative* | 24 | 2,836 | 54 | 2,572 |

TABLE F.1: Confusion matrix for different cut-off limits for the LightGBM and Logistic Regression models, where both models are trained on the same features.

# Appendix G

# Difference in approximated lost profits for the two models

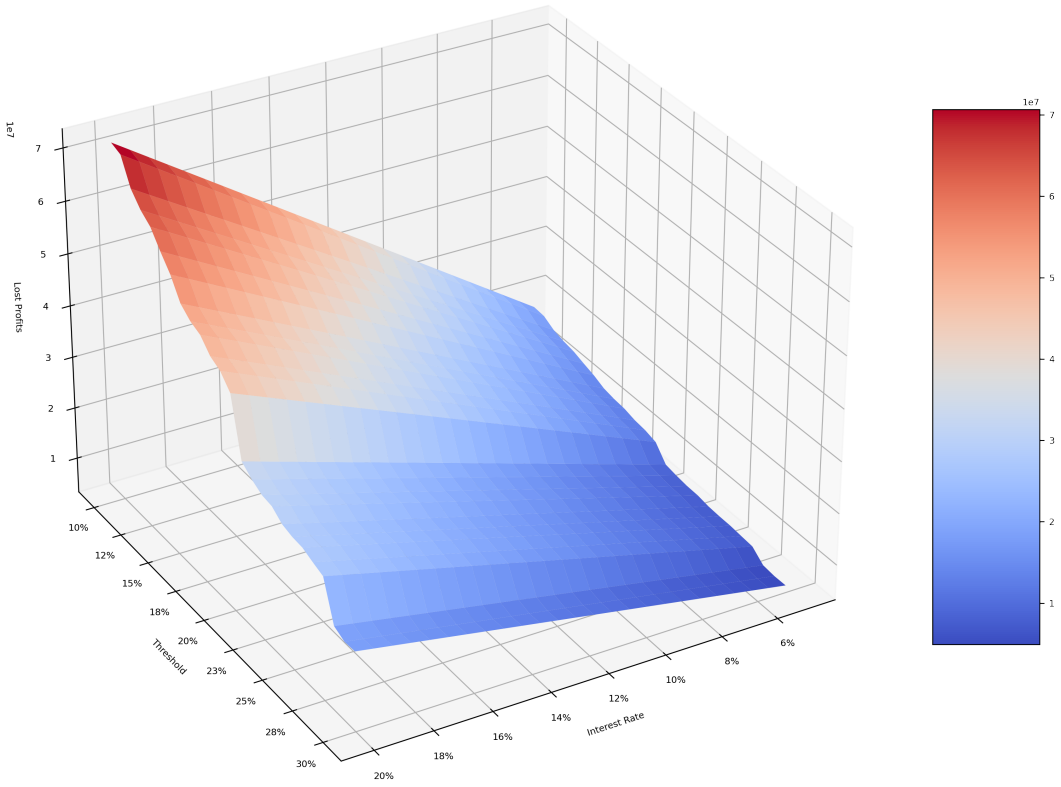



FIGURE G.1: 3D plot of differences in lost profits between Logistic Regression and LightGBM, for various levels of interest rates and thresholds for default. The graph approximates the current yearly loss of not using LightGBM as credit scoring model. Note that the LightGBM model is calibrated.

# Appendix H

# Calibration of LightGBM model

## H.1 Uncalibrated LightGBM vs calibrated LightGBM

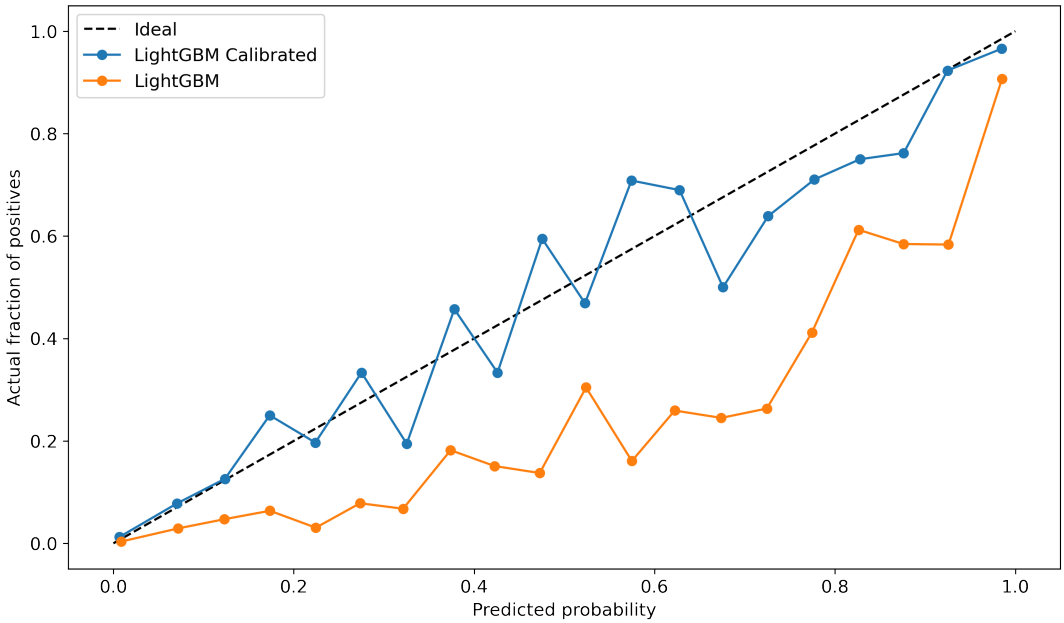



FIGURE H.1: Predicted probabilities against true fraction of probabilities. The black dotted line represents a perfectly calibrated classifier. The blue and orange lines represents the uncalibrated and calibrated LightGBM models, respectively. Note how the uncalibrated LightGBM model (yellow) overestimates the true probabilities.

## H.2 Theory and procedure behind LightGBM calibration

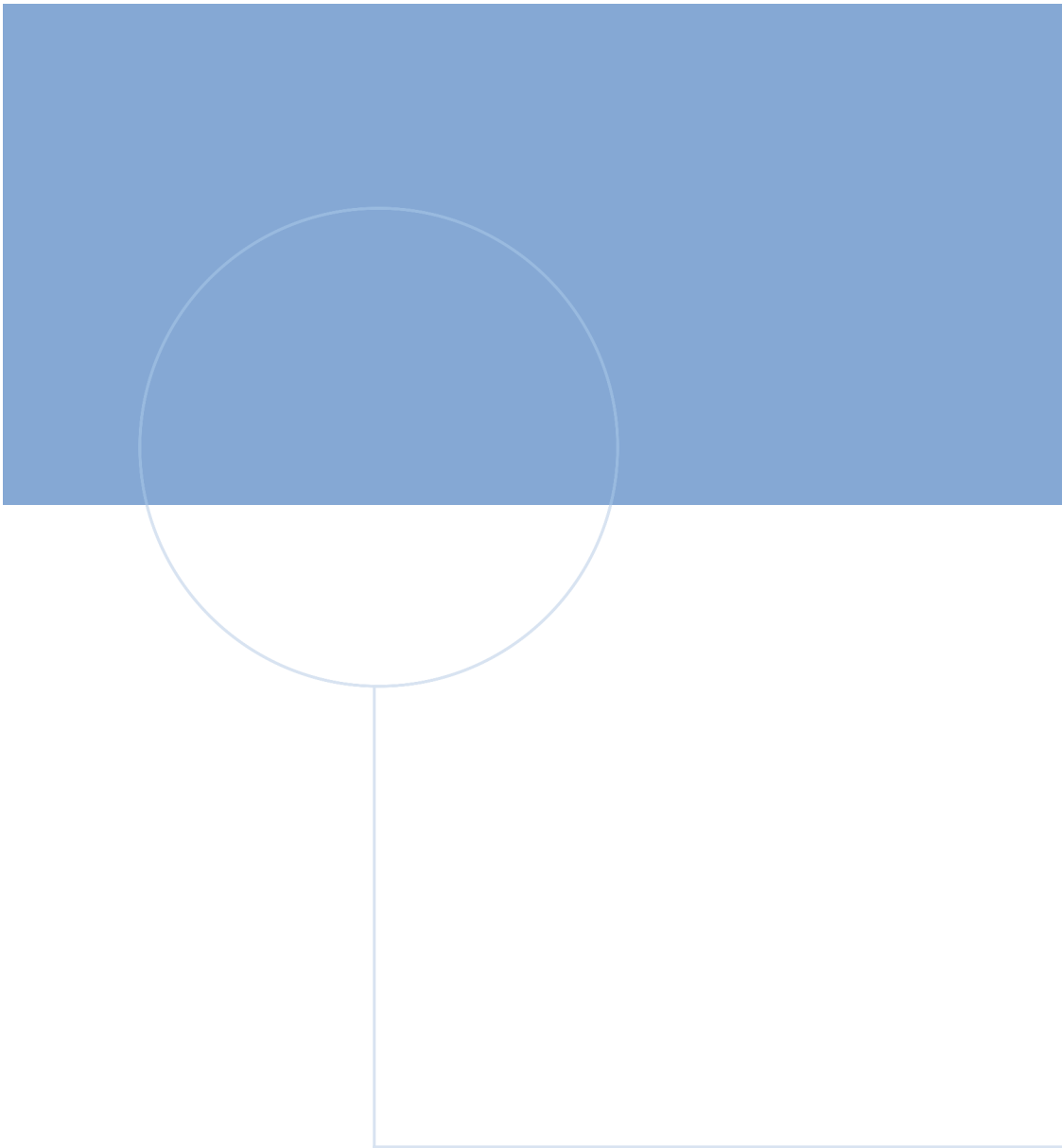
The idea behind calibration within machine learning is that a model's predicted probabilities of outcomes reflect the true probabilities of those outcomes (Nixon et al., 2019). Thus, a classification model is calibrated if the predicted probability $\hat{p}$ is always equal to the true probability, $p$, for a given class $y$. From Figure H.1, it is clear that the *uncalibrated* LightGBM model overestimates the true probabilities. For instance, for a predicted probability of 40%, the true fraction of positives (representing true probabilities) is just below 20%.

Our calibration procedure can be summarized as follows:

1. Instead of using the LightGBM model directly for predicting, we stored the index of the leaf used for the prediction. Thus, since our model had 25000 trees, an array of shape $(Nx25000)$ was stored for predictions on $N$ instances. Each element in the array indicates the leaf of each tree.
2. This array was then one-hot encoded, yielding a $(Nx75000)$ array.
3. Using this array as our input data, $X$, and the actual targets as the labels, $y$, a Linear Regression model $f(X, y)$ was trained.

Thus, this model would function as a regressor mapping the LightGBM classifier output to a calibrated probability between 0 and 1.

Since we used stratified k-fold cross-validation, 10 LightGBM models were trained. Thus, 10 models had to be calibrated. Therefore, the procedure mentioned above was repeated 10 times, yielding 10 Linear Regression models (calibrators). Thus, the final predictions became the mean of the outputs from all 10 calibrators. Note that to reduce overfitting, the calibrators were only trained on the training data, representing 60% of the overall dataset, and only evaluated on the test data. The resulting calibrated LightGBM model is shown in Figure H.1 in blue, where we clearly see that the predicted probabilities are closer to the ideal probabilities.



NTNU

Norwegian University of
Science and Technology