Explainable AI for Credit Scoring in Banks
Appendices

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 α is the intercept and β_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.

Gradient Boosting Decision Trees

In this section we provide more details on the GBDT models briefly discussed in section 3.1 of the paper.

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

$$L = \underset{F}{argmin} \sum_{i=1}^{M} l(y_i, F(x_i)) \tag{A.6}$$

Here, the strong learner F can be represented as a sum of T weak learners f_w (e.g., decision trees), such that $F(x_i = \sum_{w=1}^{T} f_w(x_i))$. 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:

$$L_w = \sum_{i=1}^{M} l(y_i, F_{w-1}(x_i) + f_w(x_i)) \tag{A.7}$$

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

$$L_w = \sum_{i=1}^{M} \left[l(y_i, F_{w-1}(x_i) + g_i f_w(x_i) + \frac{h_i}{2} f_w^2(x_i)) \right] \tag{A.8}$$

Where $g_i = \frac{\partial l(y_i, F_{w-1}(x_i))}{\partial F_{w-1}(x_i)}$ and $h_i = \frac{\partial^2 l(y_i, F_{w-1}(x_i))}{\partial^2 F_{w-1}(x_i)}$ 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).

Shapley values

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

In (A.9) E 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:

$$\sum_{j=1}^{N} \theta_j(\hat{f}) = \hat{f}(x) - E\hat{f}(X) \tag{A.9}$$

Shapley values (Shapley, 1953) were initially used for calculating a *fair* payout, i.e., finding payouts to players reflecting their contribution to the total payout. 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 $S \subseteq M = \{1, \dots, M\}$ 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:

$$\emptyset_j = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|!\,(M - |S| - 1)!}{M!} \big(v(S \cup \{j\}) - v(s)\big), j = 1, \dots, M \qquad \text{(A.10)}$$

An interpretation of Equation A.10 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:

$$\emptyset_j = \frac{1}{\#players} \sum_{coalitions\ excluding\ j} \frac{marginal\ contribution\ of\ j\ to\ coalition}{number\ of\ coalitions\ excluding\ j\ of\ this\ size} \qquad \text{(A.11)}$$

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*. 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).

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

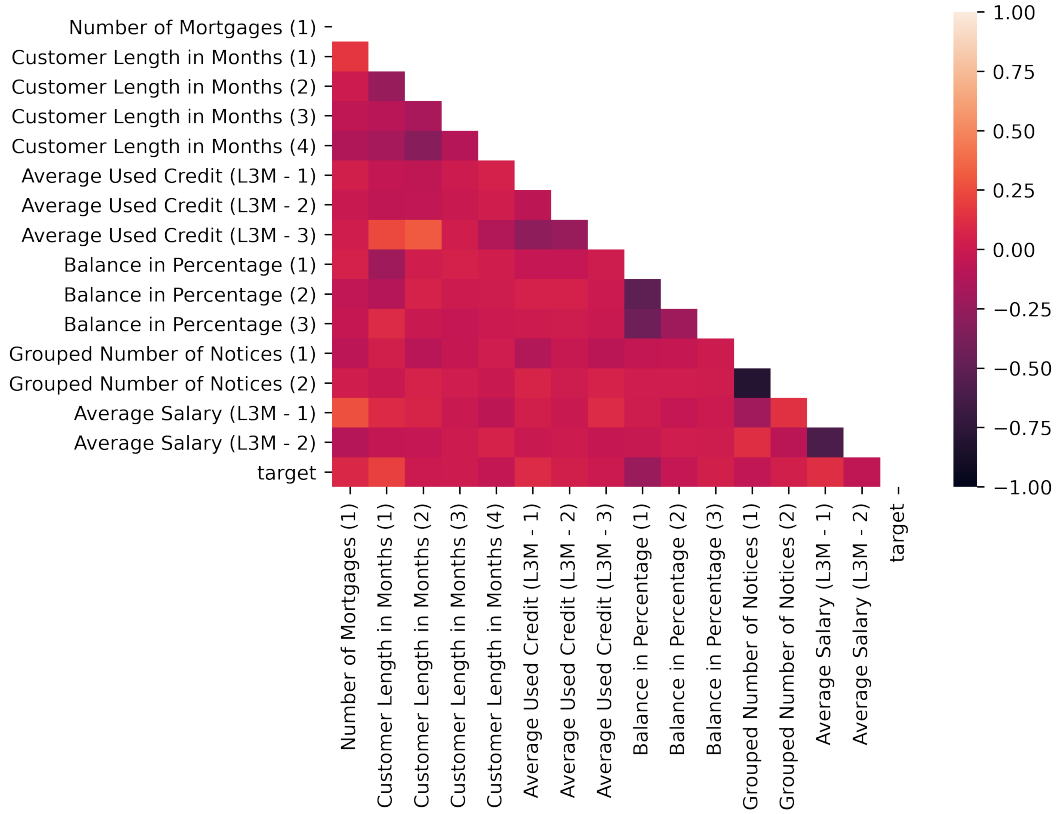


FIGURE C.1: Correlation heatmap for the features used in the LR model. The feature combinations are color coded by correlation, explained by the color scale to the right.

C.2 Principal component analysis of LR features

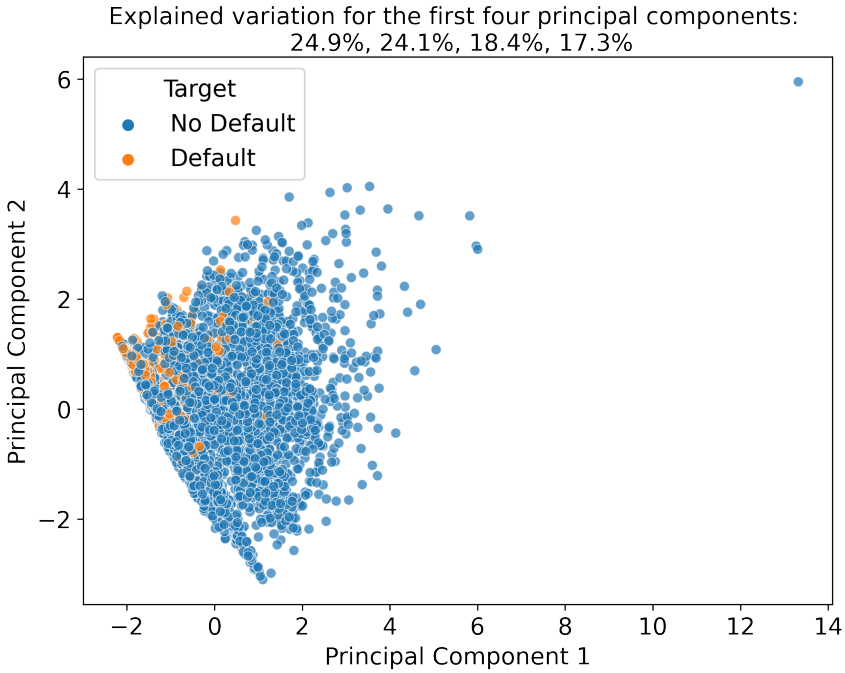


FIGURE C.2: Principal component analysis (PCA) conducted on the Logistic Regression dataset.

C.3 Violin plot of LR features

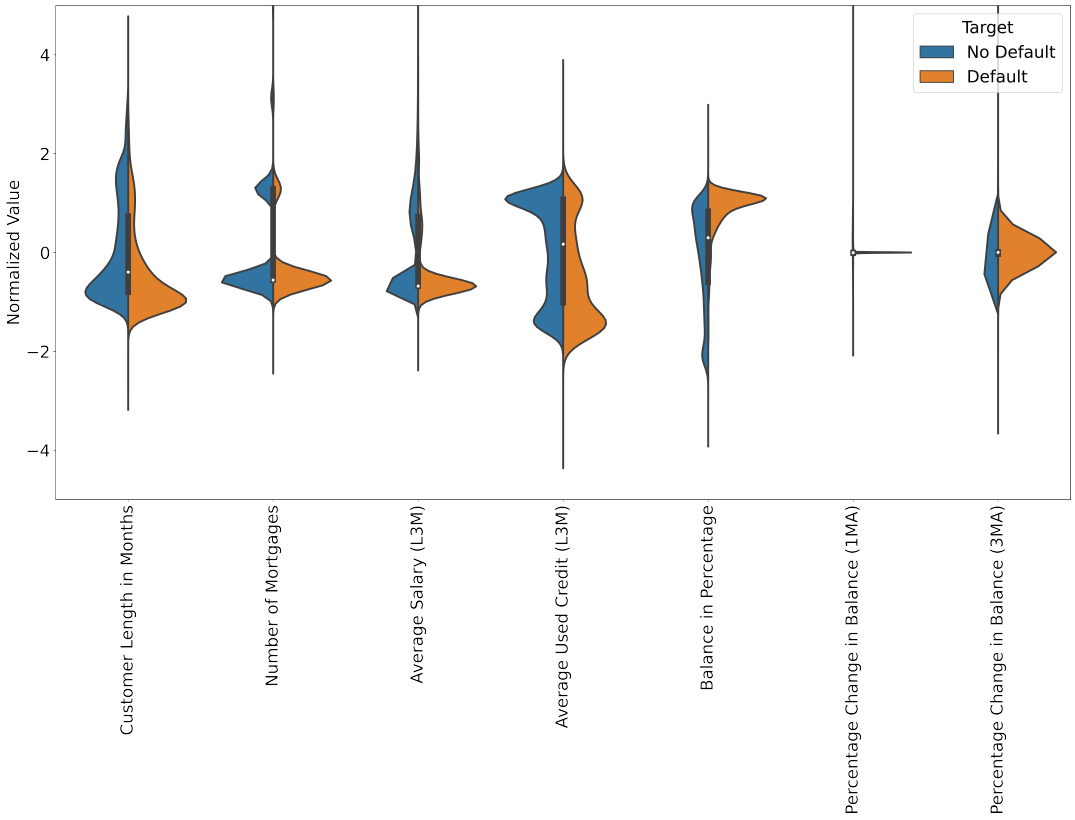


FIGURE C.3: Kernel density estimation on the Logistic Regression dataset visualized through violin plots.

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:

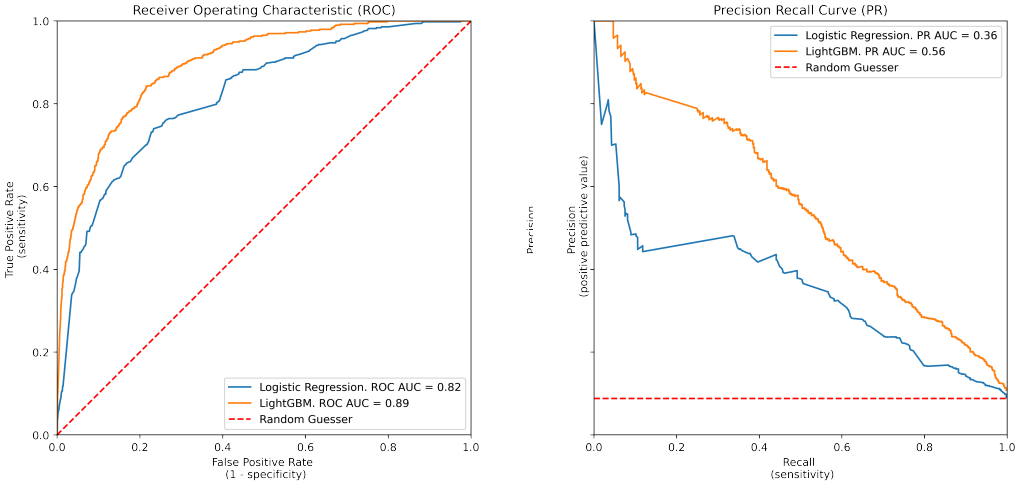
$$PPV = \frac{TP}{TP + FP} \tag{E.3}$$

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

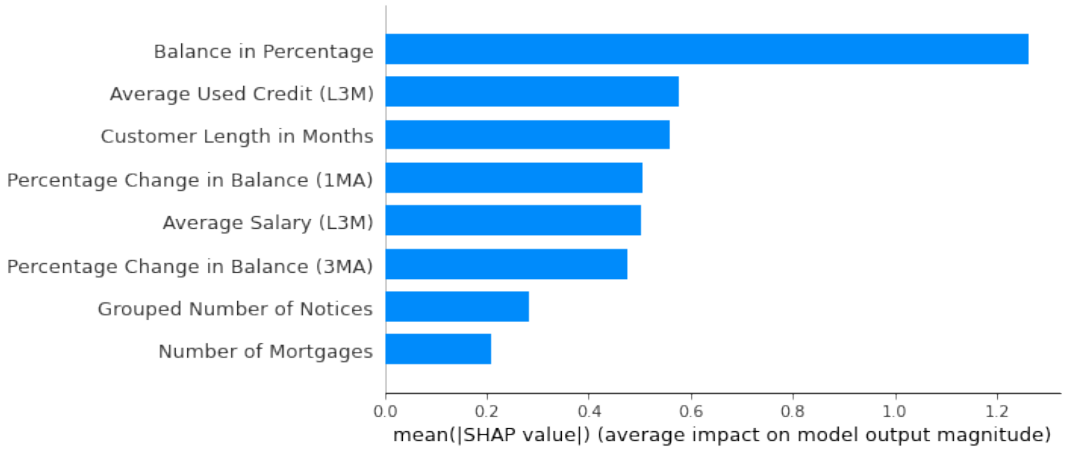


FIGURE F.2: Simplified SHAP variable importance plot for the LightGBM (LR) model ranked by importance. Note that SHAP values are in absolute *log-odds*.

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

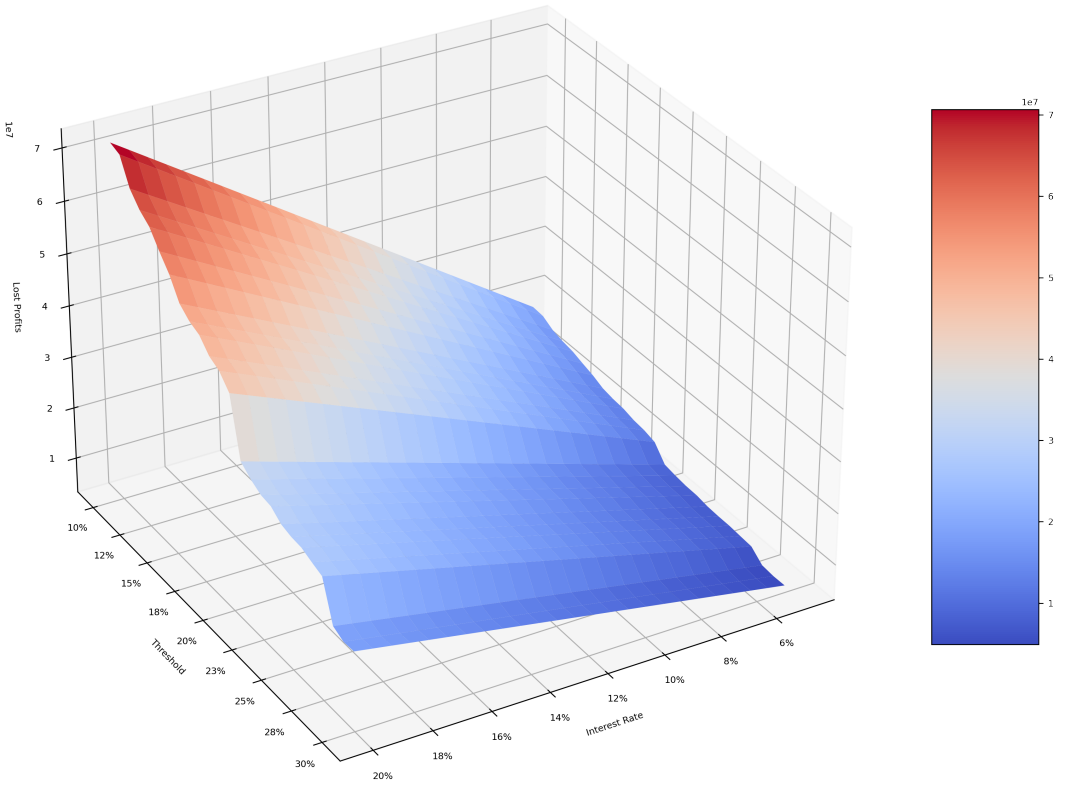


FIGURE G.1: 3D plot of differences in lost profits between Logistic Regression and LightGBM, for various levels of interest rates and thresholds for default. The graph approximates the current yearly loss of not using LightGBM as credit scoring model. Note that the LightGBM model is calibrated.

Appendix H

Calibration of LightGBM model

H.1 Uncalibrated LightGBM vs calibrated LightGBM

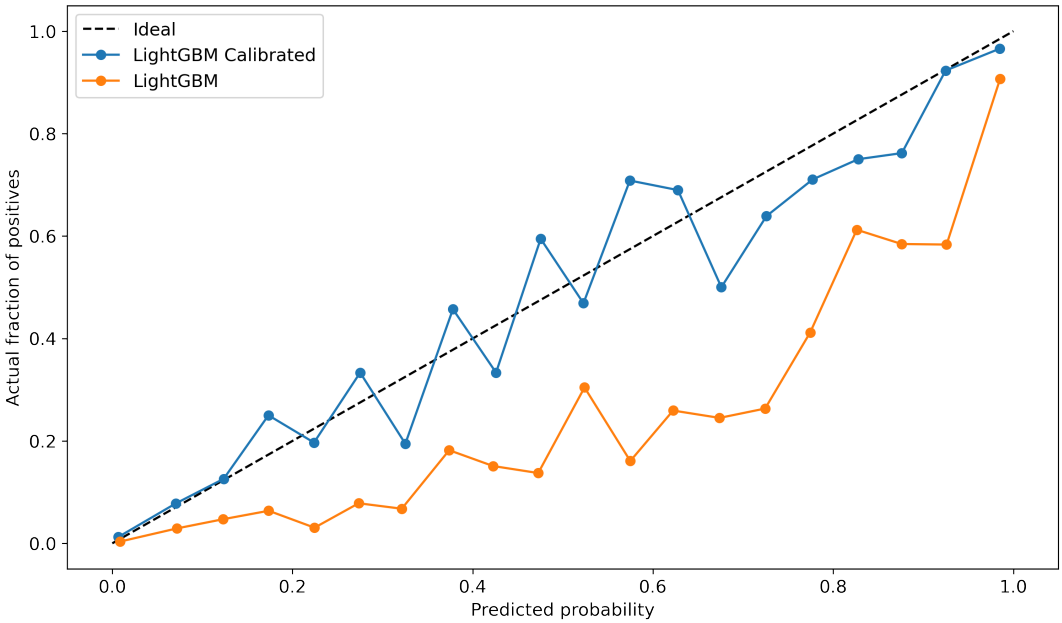


FIGURE H.1: Predicted probabilities against true fraction of probabilities. The black dotted line represents a perfectly calibrated classifier. The blue and orange lines represents the uncalibrated and calibrated LightGBM models, respectively. Note how the uncalibrated LightGBM model (yellow) overestimates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

Appendix I

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.

I .1 LightGBM dataset

Principal Component Analysis

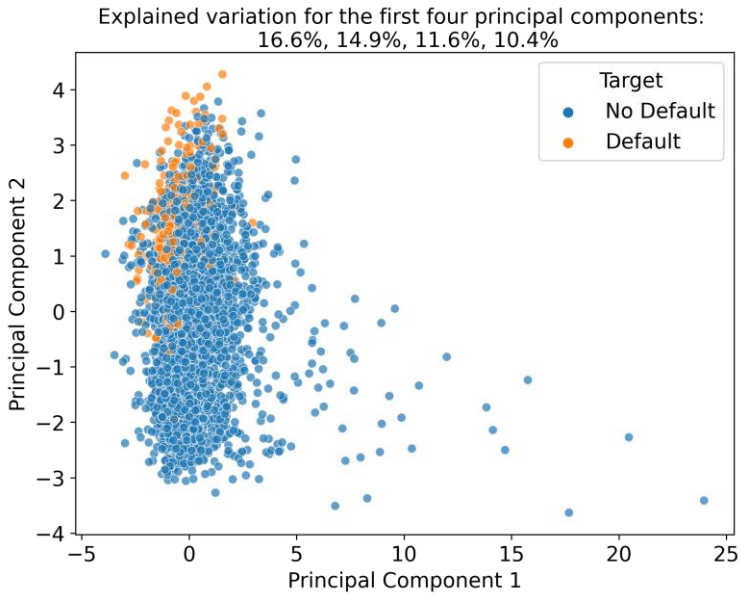


FIGURE I.1: Principal Component Analysis on the dataset used for the LightGBM model. Each instance is normalized and projected onto the space spanned out by the most dominant eigenvectors. Each instance is color-coded based on the target class.

Figure I.1 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.

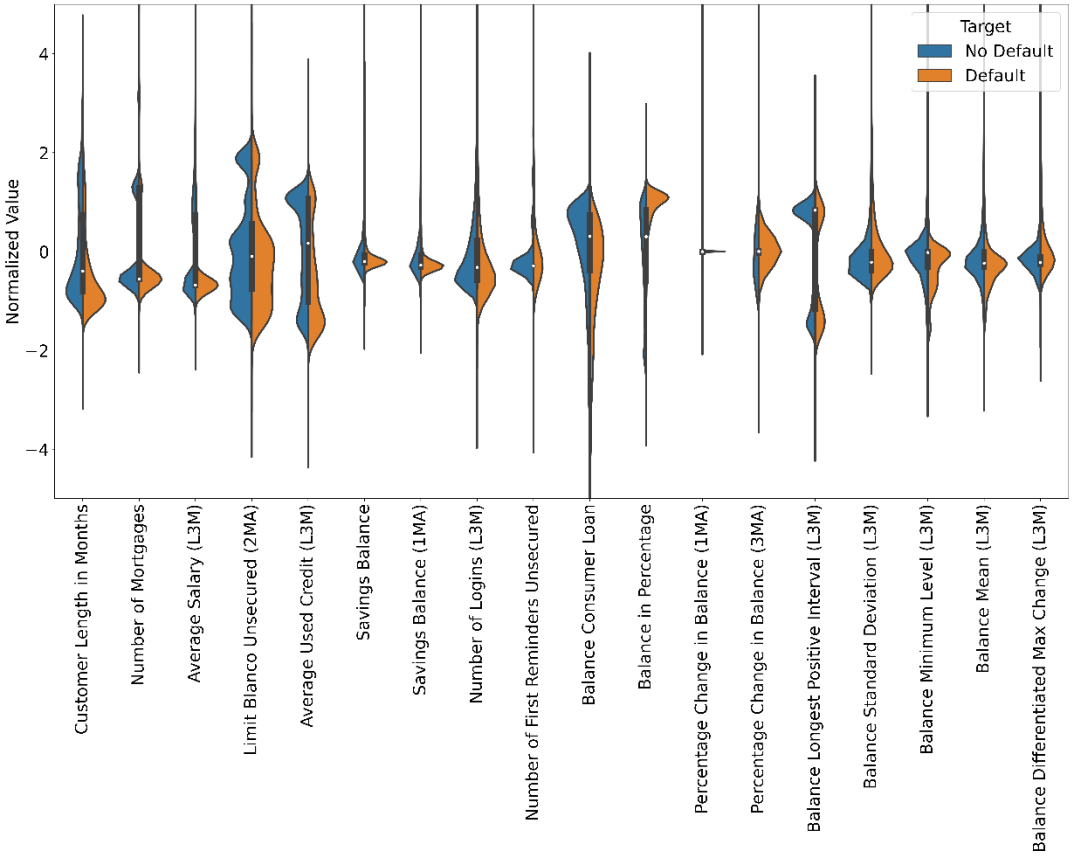


FIGURE I.2: Violin plot of the data set used for the LightGBM model, with normalized values. Each violin indicates a feature distribution and is colored based on the target class. White dots indicate the feature median, and the black bar indicates the interquartile range. Outliers with an absolute standard deviation larger than 5 are removed for visualization purposes.

Figure I.2 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 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

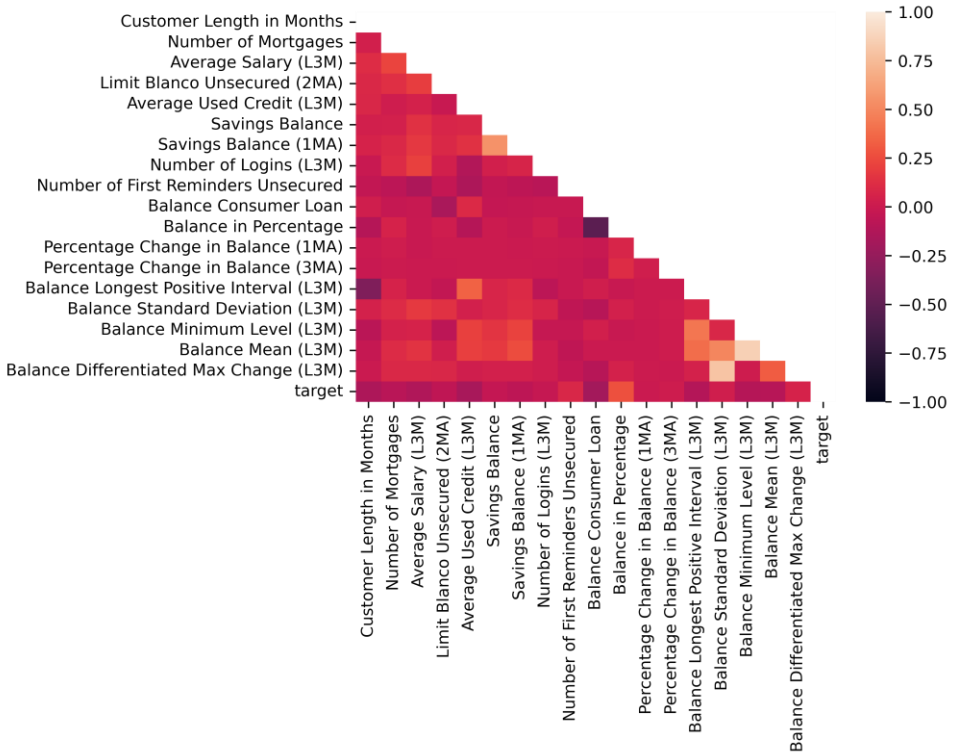


FIGURE I.3: Correlation heatmap of the dataset used by the LightGBM model. The feature combinations are color-coded based on correlation, explained by the color scale to the right. Light colors indicate positive correlations.

Figure I.3 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.