Journal of Risk **25**(2), 1–25
DOI: 10.21314/JOR.2022.046

Research Paper

Explainable artificial intelligence for credit scoring in banking

Borger Melsom,[1] Christian Bakke Vennerød,[1] Petter E. de Lange,[2] Lars Ole Hjelkrem[2] and Sjur Westgaard[1]

[1]Department of Industrial Economics and Technology Management, Faculty of Economics and Management, Norwegian University of Science and Technology, Alfred Getz vei 3, Trondheim, Norway; emails: borger.melsom@gmail.com, christian.b.vennerod@outlook.com, sjur.westgaard@ntnu.no

[2]Department of International Business, Faculty of Economics and Management, Norwegian University of Science and Technology, Larsgadrvegen 2, Alesund 6025, Norway; emails: petter.e.delange@ntnu.no, lars.o.hjelkrem@ntnu.no

(Received April 3, 2022; revised June 23, 2022; accepted August 22, 2022)

ABSTRACT

This paper proposes an explainable machine learning model for predicting credit default using a real-world data set provided by a Norwegian bank. We combine a Light Gradient Boosting Machine (LightGBM) model with Shapley additive explanations (SHAP), an explainable artificial intelligence (XAI) framework that enables the interpretation of explanatory variables affecting the predictions. Using the LightGBM model, we achieve a 1% increase in the area under the receiver operating characteristic curve and a 19% increase in the area under the precision recall curve compared with an industry-standard logistic regression model. An empirical analysis using SHAP on the explanatory variables is also conducted. Our main contribution is the exploration of how the implementation of XAI methods can be applied in banking to improve the interpretability and reliability of state-of-the-art machine learning models. We specifically find that LightGBM models may outperform logis-

Corresponding author: P. E. de Lange

Print ISSN 1465-1211 | Online ISSN 1755-2842

tic regression models for credit scoring in terms of both predictive performance and explainability.

Keywords: credit default modeling; explainable artificial intelligence (XAI); Shapley additive explanations (SHAP); machine learning; Light Gradient Boosting Machine (LightGBM); risk management.

1 INTRODUCTION

The most significant risk that a bank faces is credit risk (Apostolik 2009). Technological progress has made it possible for banks to reduce loan losses by enabling a better understanding of the risk profiles of their customers. Linear and nonlinear regression (logit and probit) models have been the industry standard for these analyses. Over the past 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). European banks have to abide by strict data 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* 2022). The explainability deficit of the best-performing ML models means that banks have to sacrifice predictive power in order to abide by the regulations. This paper argues that explainable artificial intelligence (XAI) techniques allow the best of both worlds.

Credit risk is usually synonymous with default risk (Apostolik 2009), and a report by Financial Supervisory Authority Norway (2021) found that the default rate on consumer loans in Norwegian banks increased significantly during the three years prior to its publication. Finding a way to leverage state-of-the-art ML techniques to improve the accuracy of credit scoring models is a promising strategy to reverse this ominous development. In this study, we apply XAI to ML models to provide these models with a sufficient level of explainability. The study is conducted on a data set obtained from Sparebanken Møre's retail market lending database. Sparebanken Møre is a listed Norwegian savings-and-loan (trustee savings) bank, which is actively exploring the opportunities created by the technological shift (Sparebanken Møre 2020).

The literature typically identifies three main elements of risk in credit risk modeling: probability of default (PD), exposure at default (EAD) and loss given default (LGD) (Doumpos *et al* 2019). Sparebanken Møre uses these same three elements in their risk classification of customers (Sparebanken Møre 2020). In this study, we will focus on PD and classify customers into two categories; defaulting and nondefaulting. We will then compare a logistic regression (LR) model with a state-of-the-art

ML model including XAI, in terms of predictive performance and level of explainability. The LR model, including all explanatory variables, was received from the bank and thus acts as an industry standard model. One aim of our study is to demonstrate to the bank management that they can in fact improve their default rate on consumer loans by employing ML models. Consequently, we believe that the LR model actually in use at the bank is the correct benchmark model against which to compare the performance of our ML models. Though XAI can also be applied to regression models, we will use the regression coefficients to explain the logit model to better contrast the benefits of ML in combination with XAI.

The main contribution of this paper is the implementation of XAI methods on a real-world data set, exploring how these methods can be applied to improve the interpretability and reliability of state-of-the-art ML models. We specifically find that Light Gradient Boosting Machine (LightGBM) models may outperform LR models for credit scoring in terms of both predictive performance and explicability.

This paper is organized as follows. Section 2 offers an introduction to credit risk, credit scoring in banking, the challenges banks face in explaining credit scoring models and a discussion regarding AI for credit scoring. We rely on key findings in the literature to select the AI and XAI methods for this study. Section 3 outlines the models employed in our study. Section 4 introduces and explains the data set. Finally, Section 5 provides an assessment of the models' performance and an in-depth analysis of the explainability of the models' output. Section 6 states our conclusions. Model details and data can be found in an online appendix.

2 LITERATURE REVIEW

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 local interpretable model-agnostic explanations (LIME) framework by Ribeiro *et al* (2016) and Shapley additive explanations (SHAP) values by Lundberg and Lee (2017). These models were created to help users understand the reasons behind the predictions of complex models.

The literature focusing on using XAI for credit scoring in finance is very limited. Nevertheless, there are some highly relevant previous works. These include integrating XAI on credit scoring models for peer-to-peer (P2P) lending data sets (Ariza-Garzón *et al* 2020; Bussmann *et al* 2020); an empirical study comparing XAI with a scorecard model for credit scoring on a publicly available credit bureau data set (Bücker *et al* 2022); 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 the Standard

& Poor's 500 (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 abovementioned previous applications of XAI to credit scoring were promising.

Gramegna and Giudici (2021) find that SHAP outperforms LIME in discriminating observations in their credit scoring model and that SHAP values appear to better assign values to the dynamics of the Gradient Boosting model than the LIME weights. Bussmann *et al* (2020) focus on one specific explainable model for fintech risk management, using extreme gradient boosting (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 of Bücker *et al* (2022), which show that ML techniques can achieve a level of interpretability comparable with the traditional scorecard method while preserving its computational edge. According to Ariza-Garzón *et al* (2020), applying XAI on nonlinear models such as XGBoost may even improve the explainability compared with statistical approaches (eg, LR). Such advanced models enable an understanding of complex, nonlinear aspects of the relationships between variables that classic models are unable to discover. This includes aspects such as "curved relationships, structural breaks, heteroscedasticity and outlying behavior" (Ariza-Garzón *et al* 2020).

Based on the results from Gramegna and Giudici (2021), we find sufficient evidence for utilizing SHAP in this paper.

From the discussion above, it is clear 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 (ie, the data set is highly imbalanced). Bayesian analysis of risk data was successfully applied to overcome such disparities in the early 2000s (Giudici 2001; Giudici and Bilotta 2004). In recent years, certain ML techniques, such as the gradient boosting decision tree (GBDT), have shown similar traits. Correspondingly, GBDT has frequently been used for credit scoring in the literature because it provides reasonable accuracy for imbalanced classification problems (Benhamou *et al* 2021; Brown and Mues 2012). 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) performs better globally than all the other methods in their study of credit scoring models in P2P lending. They also show that this improved 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.

3 METHODOLOGY

In this section we provide a brief outline of GBDTs, whereafter we present the essential features of LightGBM and LR. A description of Shapley values is also provided. Lastly, we outline the essential properties of SHAP.

3.1 GBDT and LightGBM

Ensemble methods combine several learners to obtain better predictive performance than a single constituent learning algorithm. The ensemble method used in this paper is the boosting algorithm LightGBM, where learners are trained on misclassified instances from the previous learners. We refer the reader to Freund and Schapire (1995) for a more detailed explanation of ensemble learners.

GBDTs use the boosting technique by sequentially training decision trees based on the residuals from the previous trees, with the objective of minimizing the loss of the strong learner F. Referring to Zhang *et al* (2017), the strong learner F can be represented as a sum of T weak learners f_w (eg, decision trees) such that $F(\boldsymbol{x}_i) = \sum_{w=1}^{T} f_w(\boldsymbol{x}_i)$. At the wth step, the previous $w-1$ weak learners are fixed when constructing the wth weak learner. Thus, when constructing the wth learner, GBDT minimizes the loss

$$L_w = \sum_{i=1}^{M} l(y_i, F_{w-1}(\boldsymbol{x}_i) + f_w(\boldsymbol{x}_i)). \tag{3.1}$$

Here, $\boldsymbol{x}_i$ and y_i are the feature vectors and corresponding targets, respectively, and $F_{w-1}(\boldsymbol{x}) = \sum_{k=1}^{w-1} f_k(\boldsymbol{x})$. 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 a given loss function with respect to the previous predictions, F_{w-1}, instead of actual labels Y.

One of the limitations of traditional GBDT methods such as XGBoost (Chen and Guestrin 2016) and AdaBoost (Freund and Schapire 1999) 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). LightGBM is a variant of the GBDT designed to be significantly faster than conventional GBDT techniques without sacrificing accuracy. This is done by implementing gradient-based one-side sampling and exclusive feature bundling. We refer the reader to the seminal paper by Ke *et al* (2017) for further technical details.

3.2 Logistic regression

To evaluate the LightGBM model's relative performance, we develop an LR baseline model. LR is commonly used to predict categorical values (Lever *et al* 2016) and is a common method for credit scoring in banks (Nguyen 2015). We built the LR model by leveraging our knowledge of how Sparebanken Møre applies LR for their credit scoring 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 the dependent variable can be split 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 any real number, whereas the probability on the left-hand side must lie between 0 and 1. This trick is performed by the logit function. The logit model we employed in this study is

$$P(Y_n = 1 \mid x_{1n}, \ldots, x_{Tn}) = \frac{1}{1 + \exp\{-(\alpha + \sum_{t=1}^{T} \beta_t x_{nt})\}}. \tag{3.2}$$

3.3 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, ie,

$$\theta_j(\hat{f}) = \beta_j x_j - \mathbb{E}(\beta_j X_j). \tag{3.3}$$

Here, $\mathbb{E}(\beta_j X_j)$ 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 minus the average predicted value:

$$\sum_{j=1}^{N} \theta_j(\hat{f}) = \hat{f}(x) - \mathbb{E}(\hat{f}(X)). \tag{3.4}$$

For more complex nonlinear models, such as LightGBM, finding these feature contributions is more complicated due to the inherent complexity of the model. Despite being nonlinear 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 by the feature value, similarly to linear models. In a nonlinear model, however, the effect of a feature can also depend on other features' effects, making it much harder to estimate feature effects. Shapley values leverage ideas from cooperative game theory to tackle this problem (Shapley 1953). Shapley values were originally used for calculating a fair payout (ie, finding payouts to players that reflect their contribution to the total payout). Since the sum of all individual payouts equals the total payout to the coalition, Štrumbelj and Kononenko (2013)

found that Shapley values can be applied in order to explain 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 (ie, 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:

$$\phi_j = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|!(M - |S| - 1)!}{M!}(v(S \cup \{j\}) - v(S)), \quad j = 1, \ldots, M. \quad (3.5)$$

An interpretation of (3.5) 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

$$\phi_j = \frac{1}{\#\text{ players}} \sum_{\text{coalitions excluding } i} \frac{\text{marginal contribution of } i \text{ to coalition}}{\text{number of coalitions excluding } i \text{ of this size}}. \quad (3.6)$$

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 explanatory technique not based on Shapley values will violate local accuracy or consistency (Molnar 2019, Section 9.6).

3.4 SHAP

Using (3.5) directly would yield exact Shapley values, but it would require the 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 (3.5), calculating Shapley values exactly would thus be challenging and computationally expensive. One solution to this problem is to use weighted linear regression (KernelSHAP) (Lundberg and Lee 2017). Another approach (the one employed in this study), TreeSHAP (Lundberg

et al 2018), is optimized for tree-based ML models such as LightGBM. TreeSHAP uses the conditional expectation $\mathbb{E}_{X_S|X_C}(\hat{f}(x) \mid x_S)$ as the contribution function v in (3.5) to estimate feature attributions. Here, X_S 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, Section 9.6). Moreover, because of their additive property (Shapley 1953), the Shapley values of the ensemble tree model are equal to the weighted average Shapley values of the individual trees.

4 DATA

The models outlined in Section 2 were implemented on a proprietary data set provided by Sparebanken Møre. The data is completely anonymized, has 50 000 rows and 16 features and was normalized upon receipt. Descriptions of the features and detailed statistics of the data are provided in Appendix A online. Four new features were generated for the LR model. These are further explained in Appendix A.4 online. The data set is substantially imbalanced, with a minority class of 2%. It contains historical customer data captured in Norway in past years, where each row represents one month for one customer. The target variable indicating default is determined by actual defaults within the 12 months following the date of recording the data. The data set contains mostly mortgages, but other loans are also present, such as car loans, credit-card loans and sole-proprietor loans. Note that there is no feature in the data set to identify the type of loan. Data visualization plots (heat maps) and a description of the features of the data are provided in Appendix A online.

4.1 Data transformation for logistic regression

The LR model used in this paper was generously provided by Sparebanken Møre. It was developed through thorough feature selection procedures and multiple iterations of different data transformations. The best-performing LR model was created using the following data transformation of the variables (Table 1 shows the transformed features): "Deposits", "arrears", "loan_value" and "wealth" were truncated at kr200, kr30, kr10 000, kr2 000 and exponentiated with $\frac{1}{5}, \frac{1}{2}, \frac{1}{5}, \frac{1}{5}$, respectively; "Collateral_savings" was created by performing a Boolean OR operation on the features "collateral", "savings_bank" and "savings_fund"; and "Households" is a binary variable based on certain codes in "sector". A variable inflation factors (VIF) method was used to verify the low degree of multicollinearity in the data. The Box–Tidwell test was performed to verify that the resulting data set used for LR is linear in log-

TABLE 1 Transformed features for the logistic regression model.

Old feature(s)	Transformed features	Transformation
Deposits	deposits_tr2_r5	$\text{pmin}(\text{deposits}, 200\,000)^{1/5}$
Arrears	arrears_tr3_r2	$\text{pmin}(\text{arrears}, 30\,000)^{1/2}$
Wealth	wealth_tr20_r5	$\text{pmin}(\text{wealth}, 2000\,000)^{1/5}$
Loan_value	loan_value_tr10_r5	$\text{pmin}(\text{loan\_value}, 10\,000\,000)^{1/5}$
Collateral	collateral_savings	pmin(collateral
Savings_bank	collateral_savings	+ savings_bank
Savings_fund	collateral_savings	+ savings_fund, 1)
Sector	households	sector 8500 or 9800

odds. A correlation heatmap of the transformed features are provided in Figure 2 of Appendix A online.

5 RESULTS

The data was split into a training set and a test set using stratified sampling, ensuring that the training set and test set both had approximately the same percentage of target classes as the original data set. Due to the relatively scarce amount of data, we used a test set containing 20% of the original data, corresponding to 10 000 rows. The test set was used to mimic out-of-sample instances for post-training model evaluation.

Stratified k-fold cross-validation was then used to optimize training on the training set. With k-fold cross-validation, the training data is partitioned into k folds, where for each fold a model is trained on the remaining data and evaluated using the held-out fold. In this study, $k = 10$ was found to be the optimal parameter. A more accurate estimate of the model-predicting performance is obtained by averaging the predictions over all 10 folds. Stratified k-fold cross-validation was implemented to ensure that each fold had approximately the same proportion of the target class.

The performance of the models was measured using receiver operating characteristic (ROC) and PR plots 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, AUC ROC and AUC PR give 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.

TABLE 2 Out-of-sample results for our LightGBM data set compared with a benchmark logistic regression model.

	LightGBM	LR
Training time (s)	142	1
Inference time (s)	17	0.002
ROC AUC	0.936	0.927
PR AUC	0.243	0.198

5.1 Model performance

Table 2 shows the out-of-sample results from the LightGBM model compared with a benchmark LR model. Notably, the optimal LightGBM model uses significantly more features than the optimal LR model provided by the bank, as it is able to use more of the information in the underlying data set. The capability of exploiting information from a larger number of features than the LR model is of course a key reason for the former's improved performance. It is clear that LightGBM performs better than the LR measured in both ROC AUC and PR AUC. However, LR is substantially faster when it comes to training the model and inferring on the holdout data set. In our opinion, the training and inference time of the LightGBM model can be considered manageable and should not cause any practical issues.

Table 3 shows the confusion matrices of our LightGBM model compared with the LR model. Note that the table includes the thresholds 0.02 and 0.05. Using a PD threshold of 0.02, any accounts with PD higher than 0.02 are classified as defaulting, and any accounts with lower or equal 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 $= 0.02$) our LightGBM model is able to capture more accounts subject to default, although at the expense of lower precision (more false positives). For the benchmark LR model, we can see that its precision is better for both thresholds, but its predictive performance in catching true positives is significantly reduced for the less strict threshold (0.05). These metrics are summarized in Table 5 of the online appendix.

Figure 1 provides the ROC and PR curves for the LR and LightGBM models. Both models perform well as measured by the ROC AUC, with scores above 0.9. They clearly outperform random guesses, indicating strong predictive capabilities. Figure 1(a) shows that, surprisingly, the LightGBM model performs only slightly better than the LR model, with an area under the LightGBM curve (orange) of 0.936 compared with 0.927 for the LR model (blue).

In terms of the PR metric, the difference is more prominent. Figure 1(b) reveals a LightGBM AUC of 0.243, which is significantly larger than the LR model's score

TABLE 3 Confusion matrix for different cutoff limits for the LightGBM and logistic regression models.

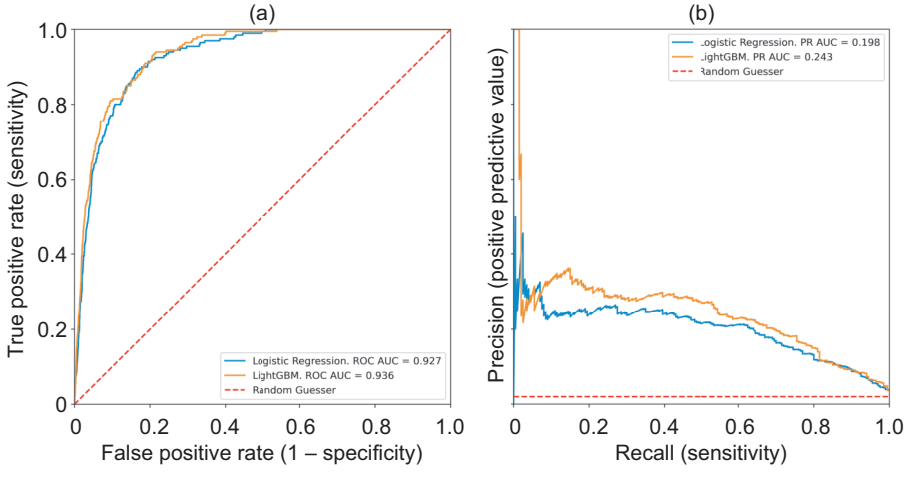
(a) Threshold = 0.02

	LightGBM		LR	
	Actual positive	**Actual negative**	**Actual positive**	**Actual negative**
Predicted positive	188	2372	174	1510
Predicted negative	12	7428	26	8290

(b) Threshold = 0.05

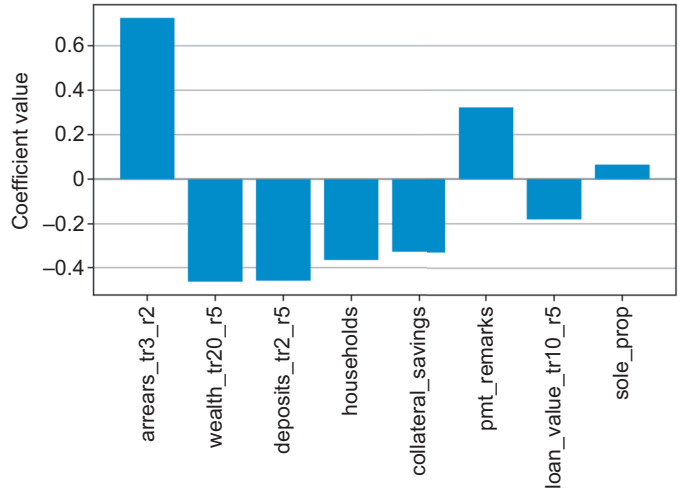
	LightGBM		LR	
	Actual positive	**Actual negative**	**Actual positive**	**Actual negative**
Predicted positive	170	1457	139	661
Predicted negative	30	8343	61	9139

FIGURE 1 Evaluation curves for the LightGBM and LR models.



(a) ROC plot. (b) PR plot.

FIGURE 2 Feature importances derived from the values of the coefficients of the LR model.



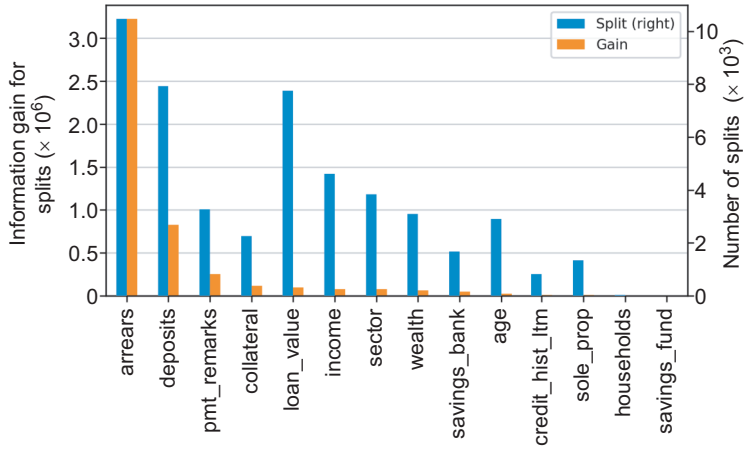
Features are ranked by absolute values.

of 0.198. The ROC and PR AUC performance of the two models shows that the LightGBM model outperforms the LR model in predicting default, thus indicating that LightGBM is the superior model for credit scoring in banks. This is our main result in terms of model performance.

5.2 Logistic regression explainability

Figure 2 shows the "feature importances" in the LR model. Since the data is normalized and has independent variables, the magnitude of the coefficient values can be used to indicate the importance of the model's predictions. The feature "arrears_tr3_r2" is the most important feature, with approximately twice the coefficient value of any other feature. With a value of approximately 0.7, a one-unit increase in arrears, ceteris paribus, would cause the odds of default to double ($\exp(0.7) = 2$). Intuitively, it makes sense that the default probability is higher for customers with more arrears. The next two features in terms of importance are "wealth_tr2_r5" and "deposits_tr2_r5". Both features have negative coefficient values, meaning that high levels of wealth and deposits reduce the PD in the LR model. It makes sense that wealthier customers are less likely to default. The same applies to "households" and "collateral_savings", as it is intuitive that customers with these traits are less likely to default. On the other hand, "pmt_remarks" has a positive

FIGURE 3 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 is shown on the right-hand axis, and the corresponding information gain for splits on the left-hand axis.

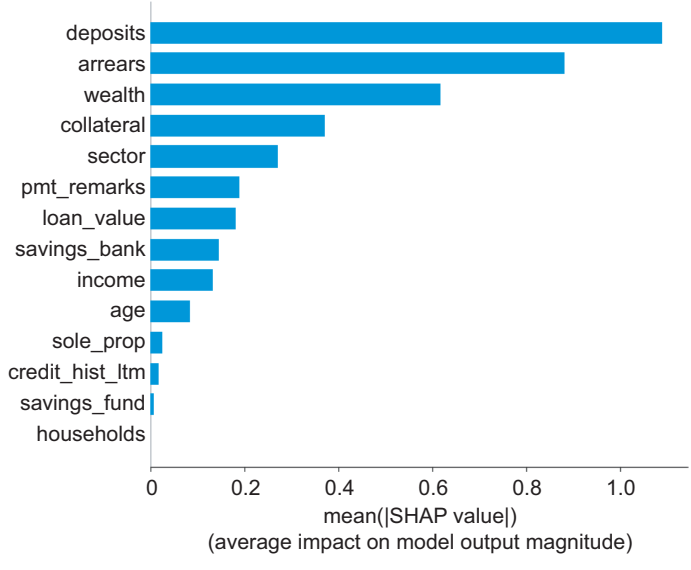
coefficient, indicating that persons with a turbulent credit history are more likely to default.

The coefficient values can thus explain which features contribute most to the model output and in what direction the features affect the predictions. However, it is difficult to understand dependencies between the features in the model from the coefficients, though partial dependence plots, as presented by Friedman (2001), could provide some insights. Further, due to the transformations outlined in online Appendix A.4 (eg, deposits being truncated at kr200 and exponentiated with 0.2), interpreting the features and corresponding coefficients becomes nontrivial. Thus, LR models are not automatically easy to interpret despite being relatively simple and linear in log-odds.

5.3 LightGBM explainability

Figure 3 shows the feature importances in the LightGBM model. Note that in order to obtain feature importances for the entire LightGBM model we averaged the feature importances across the 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, we can see that "arrears", "deposits" and "loan_value" are the features most frequently used as nodes in the model, whereas "arrears" and "deposits" are the features with the

FIGURE 4 Simplified SHAP variable importance plot for the LightGBM model ranked by importance.



Note that SHAP values are in absolute log-odds.

highest total information gain corresponding to the splits. We observe clear similarity with the most important features found by the LR model in Figure 2, as both models have identified "arrears" and "deposits" among the most important features. Unlike the LR model, the LightGBM uses untransformed features. A shortcoming of LightGBM feature importances compared with LR is that the latter are nondirectional, suggesting why XAI is needed to bolster the explainability of LightGBM.

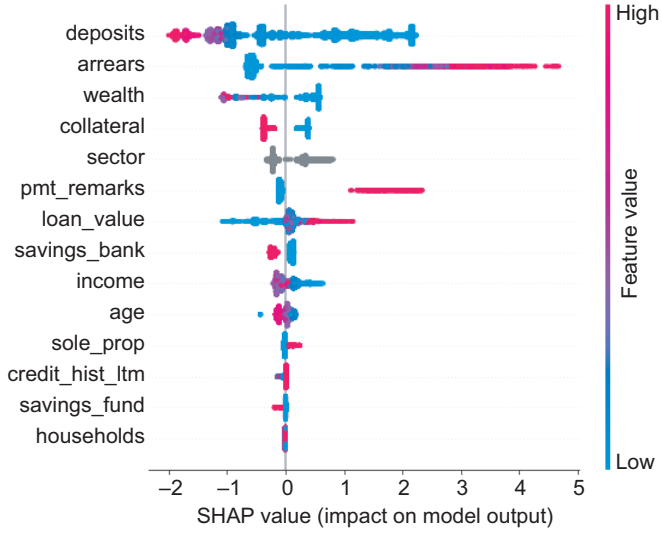
5.4 SHAP explanations

We apply SHAP values to provide further explanations of the workings of the LightGBM model. However, the tenfold cross-validation of the LightGBM model complicates the application of SHAP, as SHAP expects a single model as input. In order to overcome this issue, we averaged the SHAP values of the 10 individual models, in line with the recommendations of the creator of SHAP (Lundberg 2018).

5.4.1 Global explanations

Figure 4 shows the magnitude of the contribution by each feature, measured in absolute log-odds values. We observe that "deposits", "arrears" and "wealth" are the three

FIGURE 5 SHAP variable importance plot for the LightGBM model.

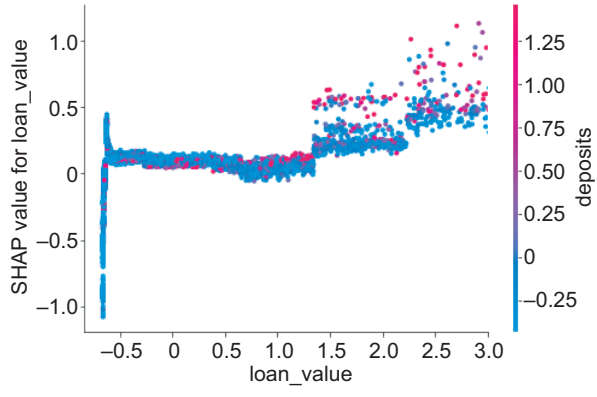


Positive SHAP values are associated with an increase in default probability, and feature values are color coded according to the scale on the right-hand side. For example, a high level of "arrears", shown in red, is associated with an increase in default probability, whereas a low level of "collateral", shown in blue, is associated with a decrease in the PD. SHAP values are measured in log-odds.

features with the highest impact on the model. The plot also shows that "households" has no impact on the output from the models. This corresponds well with the LightGBM feature importance plot (Figure 3), but is in sharp contrast with the LR plot in Figure 2, where "households" is found to be an important feature. Figure 5 shows a more detailed summary of the LightGBM model, where the effects of different feature values on the resulting prediction are visualized. High and low relative feature values in the plot are color-coded as red and blue, respectively. Categorical features are colored gray. The horizontal axis shows SHAP values measured in log-odds. High SHAP values are associated with an increase in the predicted PD, whereas low SHAP values correspond to a reduction in the predicted PD. The features are ranked by importance, with the most important features for the prediction at the top.

5.4.1.1 Dependencies between variables. Unlike LR, LightGBM can use complex cross-feature relations in its black-box calculation. Though partial dependence plots can display dependencies between variables, they cannot show how the importance of a variable can vary for different feature values and instances. SHAP is able to visualize this by plotting all instances in a scatter plot, with the feature value on the

FIGURE 6 SHAP dependence plot showing SHAP values for "loan_value" values, color-coded based on the value of the "deposits" feature.



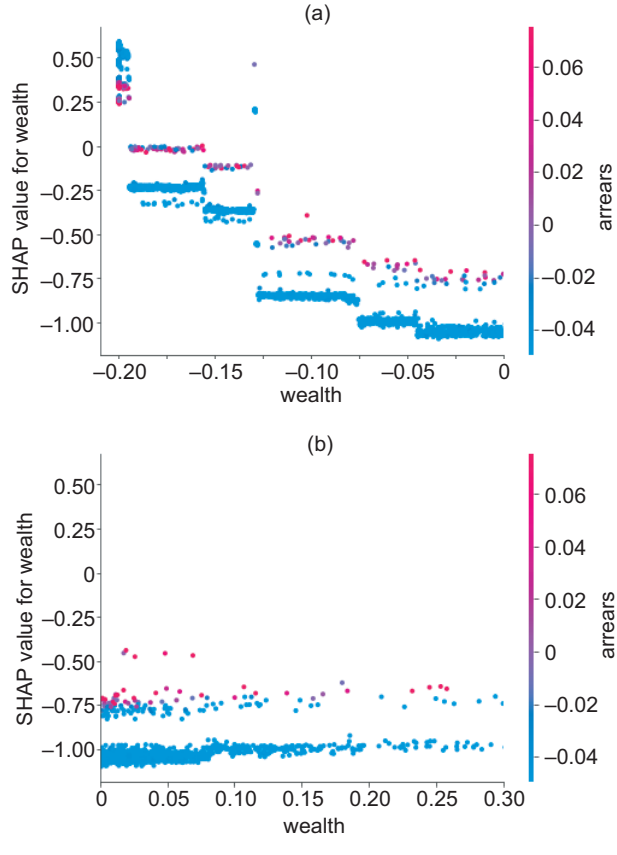
Each dot represents a customer, and positive SHAP values are associated with an increase in default probability.

horizontal axis and the importance measured in SHAP values on the vertical axis. By color-coding the values of a second feature, the plot can then display dependencies between variables and how they affect the model.

Figure 6 shows how the SHAP values for "loan_value" vary with the feature value. The plot clearly suggests a trend, where the smallest loan values are associated with negative SHAP values, loan values between approximately -0.5 and 1.4 contribute minimally to the SHAP values before there is a jump in SHAP values at around 1.4. Above this threshold, the importance of the "loan_value" feature as an indicator of default increases. At the same threshold, we can observe that the vertical spread increases, indicating that other features are interacting with "loan_value" in this region. The color-coding suggests that "deposits" is one of the variables that interact with "loan_value" above 1.4, as red dots dominate among the outliers. The concentration of red dots in the upper right corner can be interpreted as follows: the effect of high loan values on predicting default is greater for customers with large amounts of deposits. Note that this does not imply that large deposit values are associated with an increased PD; it merely suggests that loan value is a more important default indicator for customers with large deposit values.

Without knowing the details of the loan portfolio in the bank and working only with normalized data, it is difficult to interpret the underlying reason for the sudden rise at loan_value ≈ 1.4. Using two-sided Z-values and the normalized property of the data set, it is clear that the region with increased SHAP values contains the 8% largest loan values. The LR model is not able to capture the complexity of the inter-

FIGURE 7 SHAP dependence plots showing SHAP values for "wealth' values, color-coded based on the value of the "arrears" feature.



(a) All instances below average wealth. (b) All instances up to three standard deviations above the mean.

action between "loan_value" and "deposits". Considering that both variables have significant feature importance in the LightGBM model (displayed in Figure 3), it is likely that these findings contribute to the superiority of the LightGBM model.

Another example of the complex relations that the LightGBM model captures is displayed in Figure 7 by plotting two of the most important features measured in SHAP variable importance: "wealth" and "arrears". Figure 7 visualizes the SHAP values for "wealth", with below-average values plotted in part (a) and above-average values in part (b). The trends in these two plots are very different, meaning that the

impact of wealth as a predictor of default differs between customers with large and small amounts of wealth. This is important information for the bank.

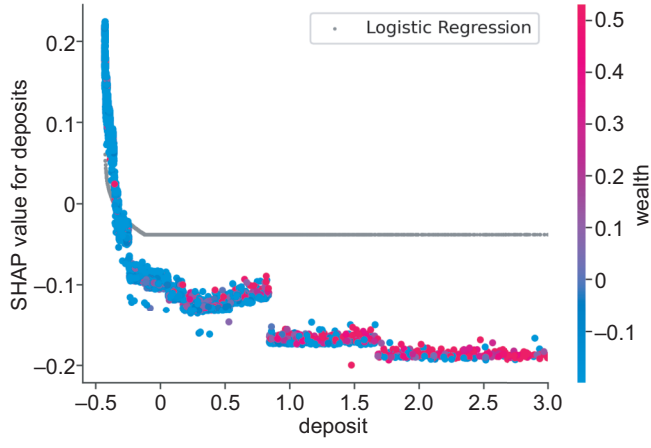
For customers with lower than average wealth, we can observe in Figure 7(a) that the SHAP values decrease as wealth increases. Only customers with a minimal amount of wealth have a positive SHAP value, suggesting that only a complete lack of wealth will increase a customer's PD. As the wealth of customers increases toward the mean value, the SHAP values gradually move in the direction of nondefault. For customers with above-average wealth, however, the SHAP plot in Figure 7(b) is almost horizontal, meaning that an increase in wealth above the mean value does not further impact the PD.

Looking at the color coding for the arrears feature in Figure 7, we can observe that the impact on the vertical axis is more significant for the blue dots than for the red dots (ie, the effect of "wealth" on the default prediction is greater for customers with a small amount of arrears). It can appear counterintuitive that the SHAP values are higher for the group of customers with the lowest amount of wealth (upper-left of Figure 7(a)) but low arrears. We believe that this counterintuitive relation might be an indicator of loan type and stem from instances of defaulting credit card customers. For the other instances, since the SHAP summary plot in Figure 5 shows a positive correlation between arrears and default probability, the finding suggests that "wealth" is a more important indicator of default for customers with low arrears than high arrears.

Figure 8 combines a SHAP dependency plot with an LR dependency plot. The LR dependencies were derived by multiplying the coefficient of the "deposits" feature, $\beta = -0.458$, by the transformed feature value for each customer, "deposits_tr2_r5". These constitute the y-values. The transformed feature values were then mapped back to original values in order to use the same x-axis. The result is visualized with gray dots in the figure. Note that the gray line is horizontal for "deposits" greater than -0.1 due to the truncation applied in the transformation of the "deposits" feature. The vertical distribution on the y-axis stops at about $y = 0.8$. The SHAP dependency plot is color-coded based on wealth values.

As we can see from the figure, the LR model is able to capture some of the feature effects of the "deposits" variable on the resulting prediction. However, it appears to underestimate the negative effect of higher deposits and strongly underestimates the positive effect of lower deposits on the target variable. We argue that this inability to sufficiently fit the curve of the SHAP values stems from the relative simplicity of the LR model and the requisite of being linear in log-odds.

FIGURE 8 SHAP dependence plot showing SHAP values for "deposits", color-coded based on the value of the "wealth" feature.



Logistic regression feature effects for deposits are displayed in gray and scaled to log-odds, corresponding to the SHAP values on the vertical axis. The transformed deposit values are mapped back to original values to fit the horizontal axis.

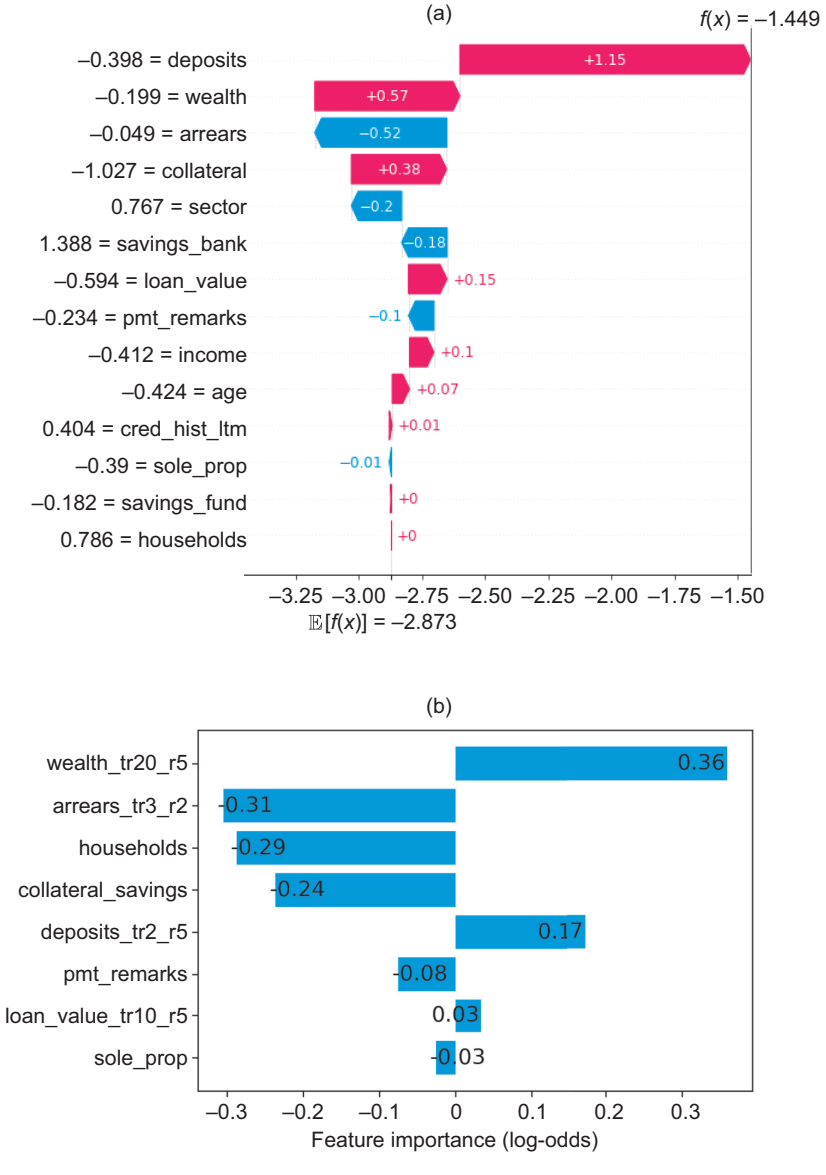
5.4.2 Local explanations

In this section we offer a novel way of comparing local explanations from SHAP with the best available local explanations from the LR model. This approach can show where individual predictions differ between the models and provide insights into to why one model outperforms the other.

We illustrate the differences using an instance where the LR model falsely predicted nondefault while the LightGBM model correctly predicted default. The models are compared using a SHAP waterfall plot for the LightGBM model and an equivalent plot for feature importances in the LR model, shown in Figure 9. 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-hand side. In part (a), the prediction by the model is displayed as $f(x)$ and the predicted expected PD (positive bias) from SHAP is expressed as $\mathbb{E}[f(x)]$, both measured in log odds. For the LR feature effects plots, since the LR model has almost independent variables, the feature values are multiplied by the LR model coefficients, yielding the feature effects. Since the data is normalized, higher absolute values indicate a greater contribution to the resulting prediction.

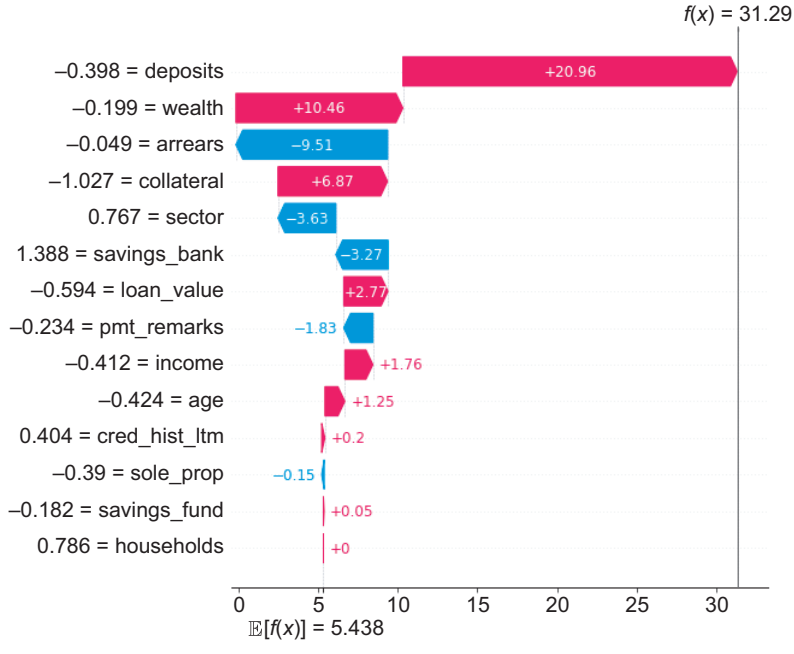
The LightGBM model correctly predicts default with a probability of 31.3%,

FIGURE 9 Local explanation of the predicted feature contributions to a customer that defaulted.



All values are in log-odds. (a) PD prediction = 31.3%. (b) PD prediction = 0.33%.

FIGURE 10 SHAP waterfall plot with values in probabilities.



whereas the LR predicts nondefault with a PD score of 0.33% (below the 2% threshold). "Deposits" and "wealth" are the two features that contribute most toward a default prediction in both models, but the magnitude of the contributions differs significantly between the models. While the below-average deposit value of −0.398 yields the largest importance value in the SHAP plot, it has a much weaker effect in the LR model. The same can be observed for the wealth value of −0.199, which yields the second-biggest contribution with SHAP but contributes significantly less than "arrears" and "households" in LR. We believe that the LR model underestimates the impact of both these feature values and that the inaccuracy can be partly explained by LR's inability to interpret the dependencies between "wealth" and "deposits" and how both features interact with the "arrears" feature (−0.049, below average for this specific instance). In the discussion surrounding Figure 7(a) we found that wealth is a more important predictor of default for customers with small amounts of arrears. The LR model does not capture the elevated probability from this dependency, resulting in a smaller contribution by the "wealth" feature. The difference in importance between the models is even greater for the "deposit" feature. As shown in Figure 8, the LR feature importances severely undershoot the SHAP values for the

lowest "deposit" feature values. This underestimation compared with SHAP explains the relatively small feature effect of "deposits" in the LR model. The differences in importance for these two feature values can thus explain why the models produced opposite predictions.

We believe that this type of model comparison, combining our visualization in Figure 9 with insights from global SHAP dependence plots, can contribute to increased trust in both GBDT models and SHAP plots.

Due to the additive property of SHAP values, we can further approximate the feature effects as probability measures. By normalizing the SHAP values for a given instance x_i and multiplying the normalized SHAP values by the probability $f(x_i) - \mathbb{E}[f(x_i)]$, we can convert the SHAP values from log-odds to probabilities while retaining the additive property. The result is shown in Figure 10, where we see that the predicted PD, $f(x) = 31.29\%$, equals the sum of the SHAP values in probabilities and the expected predicted PD, $E(f(x)) = 5.3\%$.

Comparing Figure 9(b) with Figure 10, we believe that the latter method of explaining the prediction is much more intuitive and descriptive. Since a loan applicant has the right to receive a detailed justification for any rejection of a loan application, we argue that LightGBM and SHAP can be used as supportive tools for both loan applicants and loan officers in a bank. As the variables used in LightGBM are not transformed, and SHAP presents the effects through probabilities, interpreting the underlying mechanisms causing a default prediction becomes trivial. The improvements of the local and global explanations compared with LR credit models shows that SHAP has the potential to advance credit scoring in terms of both predictive performance and explainability.

6 CONCLUSION

This paper has shown that LightGBM models may outperform LR models for credit scoring in terms of both predictive performance and explainability.

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 underlying mechanisms explaining the credit models. Finally, increased explainability can improve customers' trust in the credit scoring systems by providing detailed reasoning for customers whose loan applications are rejected.

DECLARATION OF INTEREST

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper.

REFERENCES

Apostolik, R. (2009). *Foundations of Banking Risk: An Overview of Banking, Banking Risks, and Risk-Based Banking Regulation*. Wiley Finance Series. Wiley (https://doi.org/10.1002/9780470555996).

Ariza-Garzón, M. J., Arroyo, J., Caparrini, A., and Segovia-Vargas, M. (2020). Explainability of a machine learning granting scoring model in peer-to-peer lending. *IEEE Access* **8**, 64 873–64 890 (https://doi.org/10.1109/ACCESS.2020.2984412).

Benhamou, E., Ohanna, J. J., Saltiel, D., and Guez, B. (2021). Explainable AI (XAI) models applied to planning in financial markets. Research Paper 3862437, Université Paris-Dauphine (https://doi.org/10.2139/ssrn.3862437).

Brown, I., and Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications* **39**(3), 3446–3453 (https://doi.org/10.1016/j.eswa.2011.09.033).

Bücker, M., Szepannek, G., Gosiewska, A., and Biecek, P. (2022). Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society* **73**(1), 70–90 (https://doi.org/10.1080/01605682.2021.1922098).

Bussmann, N., Giudici, P., Marinelli, D., and Papenbrock, J. (2020). Explainable AI in fintech risk management. *Frontiers in Artificial Intelligence* **3**, Paper 26 (https://doi.org/10.3389/frai.2020.00026).

Chen, T., and Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery And Data Mining*, pp. 785–794. ACM, New York (https://doi.org/10.1145/2939672.2939785).

Doumpos, M., Lemonakis, C., Niklis, D., and Zopounidis, C. (2019). *Analytical Techniques in the Assessment of Credit Risk: An Overview of Methodologies and Applications*. EURO Advanced Tutorials on Operational Research. Springer (https://doi.org/10.1007/978-3-319-99411-6).

Financial Supervisory Authority Norway (2021). Resultatrapport for finansforetak, 1 halvår 2021. Finanstilsynet, Oslo. URL: https://bit.ly/3sT0R9w. (In Norwegian.)

Freund, Y., and Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, Vitányi, P. (ed), pp. 23–37. Lecture Notes in Computer Science, Volume 904. Springer (https://doi.org/10.1007/3-540-59119-2_166).

Freund, Y., and Schapire, R. E. (1999). A short introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 1401–1406. Morgan Kaufmann.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics* **29**(5), 1189–1232 (https://doi.org/10.1214/aos/1013203451).

Giudici, P. (2001). Bayesian data mining, with application to benchmarking and credit scoring. *Applied Stochastic Models in Business and Industry* **15**(1), 69–81 (https://doi.org/10.1002/asmb.425).

Giudici, P., and Bilotta, A. (2004). Modelling operational losses: a Bayesian approach. *Quality and Reliability Engineering International* **20**(5), 407–417 (https://doi.org/10.1002/qre.655).

Gramegna, A., and Giudici, P. (2021). SHAP and LIME: an evaluation of discriminative power in credit risk. *Frontiers in Artificial Intelligence* **4**, Paper 140 (https://doi.org/10.3389/frai.2021.752558).

Hess, A. S., and Hess, J. R. (2019). Logistic regression. *Transfusion* **59**(7), 2197–2198 (https://doi.org/10.1111/trf.15406).

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. (2017). *LightGBM: a highly efficient Gradient Boosting Decision Tree*. Advances in Neural Information Processing Systems, Volume 30. Curran Associates, Red Hook, NY. URL: https://bit.ly/3GpZY0r.

Lever, J., Krzywinski, M., and Altman, N. (2016). Logistic regression. *Nature Methods* **13**(7), 541–542 (https://doi.org/10.1038/nmeth.3904).

Lundberg, S. (2018). How to get SHAP values of the model averaged by folds? GitHub Depository. URL: https://github.com/slundberg/shap/issues/337.

Lundberg, S., and Lee, S. (2017). *A Unified Approach to Interpreting Model Predictions*. Advances in Neural Information Processing Systems, Volume 30. Curran Associates, Red Hook, NY (https://doi.org/10.48550/arXiv.1705.07874).

Lundberg, S., Erion, G., and Lee, S. (2018). Consistent individualized feature attribution for tree ensembles. Preprint (arXiv:1705.07874v2 [cs.AI]) (https://doi.org/10.48550/arXiv.1802.03888).

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

Nguyen, H. (2015). Default predictors in credit scoring: evidence from france's retail banking institution. *The Journal of Credit Risk* **11**(2), 41–66 (https://doi.org/10.21314/JCR.2015.191).

Quinto, B. (2020). *Next-Generation Machine Learning with Spark: Covers XGBoost, LightGBM, Spark NLP, Distributed Deep Learning with Keras, and More*. Apress, Berkeley, CA (https://doi.org/10.1007/978-1-4842-5669-5).

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?": explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1135–1144. ACM, New York (https://doi.org/10.1145/2939672.2939778).

Shapley, L. S. (1953). Stochastic games. *Proceedings of the National Academy of Sciences* **39**(10), 1095–1100 (https://doi.org/10.1073/pnas.39.10.1095).

Shrikumar, A., Greenside, P., and Kundaje, A. (2019). Learning important features through propagating activation differences. Preprint (arXiv:1704.02685v2) (https://doi.org/10.48550/arXiv.1704.02685).

Sparebanken Møre (2020). Annual report 2020. Report, Sparebanken Møre. URL: https://rapporter.sbm.no/upload_images/625CCE55505F474186F5A2B6204632B1.pdf.

Štrumbelj, E., and Kononenko, I. (2013). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems* **41**(3), 647–665 (https://doi.org/10.1007/s10115-013-0679-x).

Young, H. P. (1985). Monotonic solutions of cooperative games. *International Journal of Game Theory* **14**, 65–72 (https://doi.org/10.1007/BF01769885).

Zhang, H., Si, S., and Hsieh, C. (2017). GPU-acceleration for Large-scale Tree Boosting. Preprint (arXiv:1706.08359 [stat.ML]) (https://doi.org/10.48550/arXiv.1706.08359).