Interpretable machine learning for imbalanced credit scoring datasets

Article *in* European Journal of Operational Research · June 2023

DOI: 10.1016/j.ejor.2023.06.036

Yujia Chen
The University of Edinburgh

Raffaella Calabrese
The University of Edinburgh

Interpretable machine learning for imbalanced credit scoring datasets

Yujia Chen*[1], Raffaella Calabrese[1], and Belen Martin-Barragan[1]

[1] *Business School, University of Edinburgh, 29 Buccleuch Place, Edinburgh EH8 9JS, UK*

Abstract

The class imbalance problem is common in the credit scoring domain, as the number of defaulters is usually much less than the number of non-defaulters. To date, research on investigating the class imbalance problem has mainly focused on indicating and reducing the adverse effect of the class imbalance on the predictive accuracy of machine learning techniques, while the impact of that on machine learning interpretability has never been studied in the literature. This paper fills this gap by analysing how the stability of Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP), two popular interpretation methods, are affected by class imbalance. Our experiments use 2016-2020 UK residential mortgage data collected from European Datawarehouse. We evaluate the stability of LIME and SHAP on datasets of progressively increased class imbalance. The results show that interpretations generated from LIME and SHAP are less stable as the class imbalance increases, which indicates that the class imbalance does have an adverse effect on machine learning interpretability. To check the robustness of our outcomes, we also analyse two open-source credit scoring datasets and we obtain similar results.

Keywords: OR in banking; Interpretability; Stability; Credit scoring; Machine learning

*Corresponding author. Email address: Yujia.Chen@ed.ac.uk

1. Introduction

Financial institutions rely on credit scoring models to estimate the default probability of borrowers and decide whether or not to approve loan applications. With the boosted enthusiasm in the machine learning-based predictive techniques adopted in finance, applications such as credit scoring have gained substantial interest from both academia and industry (Bank of England, 2019; Brown & Mues, 2012; Chang et al., 2018; Lessmann et al., 2015). However, machine learning techniques such as Neural Networks and Extreme Gradient Boosting (XGBoost) are regarded as "black-box" methods since they are too complex to explain and validate their predictions.

In academia, there has been a debate about the trade-off between the gain in accuracy and the loss in interpretability obtained with advanced credit scoring models (e.g., Bücker et al., 2021; Dumitrescu et al., 2022; Gunnarsson et al., 2021). Moreover, regulators have been committed to revealing new risks brought by machine learning techniques and emphasising the need for modelling transparency and interpretability in the lending sector. For example, in the United States, the Equal Credit Opportunity Act (ECOA) requires creditors to provide statements of specific reasons to applicants against whom adverse action is taken. Therefore the public can be protected from the risk of using "black-box" credit scoring models following this regulation (Consumer Financial Protection Bureau, 2022). Similar regulation is also included in the General Data Protection Regulation (GDPR) in the European Union (Voigt & von dem Bussche, 2017). The European Union also proposed the Artificial Intelligence (AI) Act to identify risk categories for AI applications, and credit scoring is classified as a high-risk AI application (European Commission, 2021). The European Banking Authority (EBA) recognises the necessity for financial institutions to take interpretability into account in their financial decisions (European Banking Authority, 2020). It also insists on using machine learning interpretability techniques when building Internal ratings-based models (European Banking Authority, 2021). Similarly, in a report on the governance of AI in Finance (Laurent Dupont et al., 2020), the French Prudential Supervision and Resolution Authority (ACPR) discusses the requirements of interpretability and potential interpretability methods that could be used with "black-box" models in credit scoring. As mentioned by EBA and ACPR (European Banking Authority, 2020; Laurent Dupont et al., 2020), model-agnostic interpretation methods such as Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016) and SHapley Additive exPlanations (SHAP) (Lundberg & Lee, 2017) could be used at a second stage to interpret the prediction results generated by the "black-box" credit scoring models. These interpretation methods can keep the high predictive accuracy of the machine learning models and make the prediction interpretable.

Class imbalance is also a common issue in the credit scoring domain, as bad customers only account for a very small proportion of all customers. The proportion of defaults varies with financial products. For example, the default rate in mortgage portfolios is typically less than 0.5% (Thomas et al., 2017), while loans to Small and Medium Enterprises (SMEs) have a higher percentage (around 5%) (Andreeva et al., 2016; Gramegna & Giudici, 2021). In the context of the class imbalance problem, current research has mainly focused on analysing the effects of class imbalance on the predictive ability of machine learning techniques. It can be concluded that the predictive performance decreases as the class imbalance level increases and the researchers have proposed various methods to improve it (e.g. Calabrese and Osmetti, 2015; Chawla et al., 2011; Krawczyk, 2016; Li et al., 2019). However, based on our knowledge, the impact of class imbalance on the performance of interpretation methods has never been studied in the literature. Considering the growing popularity of using interpretation methods for high-stakes decision-making such as credit scoring, this omission is a significant gap in the research.

Since regulations like ECOA (Consumer Financial Protection Bureau, 2022) and GDPR (Voigt & von dem Bussche, 2017) provide the right for individuals to receive an explanation of a decision made by "black-box" systems, customers could expect some guidance provided by interpretation methods on how to act to observe a desired outcome, such as an approval of a loan (ICO and The Alan Turing Institute, 2020; Singh et al., 2021). However, suppose the interpretive performance could be disturbed due to the class imbalance problem - customers may put effort into improving a specific feature (credit index) that may not be taken into account in their next loan application, hence getting disappointed and losing opportunities due to the misleading information. Moreover, financial institutions who provide unstable or misleading interpretations for similar situations seem to violate regulations (e.g., ECOA and GDPR) and could thus put themselves at risk. Customers will also lose faith in those financial institutions which may lead to a crisis of confidence and consequently bring substantial financial loss to the institutions.

We make three methodological and three empirical contributions to the literature. From the methodological perspective, first, we propose an experimental framework including a controlled sampling process to investigate the stability of LIME and SHAP considering the effects of class imbalance, which has never been studied in the literature. Second, we apply two novel indexes, named Sequential Rank Agreement (SRA) (Ekstrøm et al., 2019) and Coefficient of Variation (CV), to evaluate the interpretation stability in two ways. For both LIME and SHAP, key information sought by users is the importance ranking of the relevant features. Specifically, a ranking list of features is determined by the magnitude of the absolute SHAP values and the magnitude of the absolute LIME coefficient values. Therefore, in this paper, we measure the stability of LIME and SHAP based on their feature ranking lists and the corresponding feature importance values. We use SRA to measure the feature ranking stability, defined as the similarity of feature ranking lists generated to interpret the prediction results for a specific target at the same class imbalance level. For the feature importance value stability, we use CV to measure the similarity of the absolute SHAP values and absolute LIME coefficients corresponding to the same feature when interpreting the prediction results for a specific target at the same class imbalance level. Third, we extend the work of Visani et al. (2021) to measure and compare the "internal" and "external" stability of LIME by checking the coefficients' confidence intervals and the similarity of features during the feature selection process in LIME, using the Coefficients Stability Index (CSI) and the Variables Stability Index (VSI), respectively.

From the empirical point of view, our first contribution is to measure the stability of LIME and SHAP when the level of class imbalance incrementally increases in the credit scoring context for the first time. To make our experiments as close to bank practice as possible, we use a dataset on residential mortgage defaults obtained from the European Datawarehouse, a centralised securitisation repository implemented by the European Central Bank that collects, validates and distributes standardised loan-level data for several European countries. We also use two additional open source credit scoring datasets, which are South German Credit Dataset [1] and Taiwan Credit Card Dataset [2], to enable reproducibility and verify the robustness of our experiments. Second, we use both XGBoost and Random Forest[3] as the "black-box" machine learning models to make predictions, and we evaluate the interpretation stability based on them. We choose XGBoost and RF since they have become increasingly prevalent in the credit scoring domain over recent years for their superior predictive performance (Barbaglia et al., 2021; Gunnarsson et al., 2021; Xia et al., 2017). Moreover, they perform relatively better with class imbalance than other "black-box" machine learning models (Brown & Mues, 2012; Fitzpatrick & Mues, 2016). Third, we show that the stability of LIME and SHAP will be affected by the class imbalance, especially at extreme class imbalance levels (lower than 5% default rate). Specifically, the feature importance ranking becomes less stable as the class distribution becomes less balance. The variation of feature importance value deepens with the increase of the class imbalance level, and the "internal" stability of LIME does suffer from extremely imbalanced data.

The rest of this paper is structured as follows. Section 2 reviews the work that has used the interpretation methods in the credit scoring domain, and we focus on how they deal with the class imbalance problem. A brief introduction to XGBoost, LIME and SHAP can be found in Section 3. Section 4 thoroughly explains the proposed experimental framework from two aspects: data pre-processing and sampling procedure. Section 5 explains the stability indexes used in this paper. The results of the empirical study are then presented and discussed in Section 6. Eventually, Section 7 gives the conclusions drawn from the study and discusses the possible directions of future research.

2. Literature review

Besides applying interpretable predictive models (e.g. logistic regression), using model-agnostic interpretation methods (Molnar, 2021) to explain the prediction results separately after applying the machine learning "black-box" predictive models, has attracted more attention from both academia and industry. The great advantage of the model-agnostic interpretation methods over interpretable predictive models is their flexibility, including model flexibility, explanation flexibility and representation flexibility (See Ribeiro et al., 2016 for more details).

Researchers have developed various model-agnostic interpretation methods focusing on different interpretation aspects. Some methods such as Partial Dependence Plots (Friedman, 2001) and Accumulated Local Effects Plots (Apley & Zhu, 2020) interpret the "black-box" models by analysing the relationships among features or between features and dependent variables. Other methods try to interpret the prediction results of "black-box" models in various ways, including through identifying the features' contributions to the predictions like SHAP or through

[1] https://archive.ics.uci.edu/ml/datasets/South+German+Credit

[2] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[3] The results of XGBoost and Random Forest (RF) are very similar. Due to the page limits, we only present the results of XGBoost in the main text. The results of RF can be found in Supplementary Materials (Part H).

analysing the variance in feature values when disturbing the prediction results like Counterfactual Explanations (Wachter et al., 2017). There are also methods like LIME and Local Rule-Based Explanation (Guidotti et al., 2018) using interpretable models such as logistic regression and decision trees to understand the prediction results or extract the IF-THEN rules directly.

Table 1 summarises the existing literature on model-agnostic interpretation methods in the credit scoring domain. Martens et al. (2007) and Szwabe and Misiorek (2018) used Support Vector Machine (SVM) and Tree-based ensembles, respectively, to predict the credit scoring behaviours. Both papers applied decision trees such as C4.5 and CART to approximate predictions provided by first stage "black-box" models, finally generating decision rules to interpret the prediction results. Bracke et al. (2019) at the Bank of England applied Shapley values and clustering algorithms to explain how much each feature of a scoring model contributes to the final prediction provided by a Gradient Tree Boosting (GB) model. The authors concluded that the important features found by Shapley values, loan-to-value ratio and current interest rate, are in line with the relevant literature. Barbaglia et al. (2021) from the European Commission similarly used GB classifiers to predict loan defaults. Using ALE plots, they observed a non-linear relationship between the current LTV and the probability of default. Ariza-Garzon et al. (2020) and Moscato et al. (2021) used the same loan default data from a P2P platform. They applied various interpretation methods such as LIME and SHAP to explain some classification models such as RF, XGBoost and Neural Networks (NN). Gramegna and Giudici (2021) also used LIME and SHAP to generate feature weights after applying XGBoost to estimate the default probability of SMEs. To evaluate LIME and SHAP's ability to define distinct groups of observations, the authors further employed the feature weights generated by LIME and SHAP as input space for a K-means clustering and an RF model. The results showed that SHAP seems to have a clear advantage in terms of discriminative power compared with that of LIME. Bücker et al. (2021) tried to build the explanation framework for credit scoring by using different model-agnostic interpretation methods to provide either global-level or local-level interpretations at various stages of the credit scoring process to satisfy the requirements of stakeholders' interests. It can be seen from Table 1 that LIME and SHAP are the two most popular model-agnostic interpretation methods used in the credit scoring domain, and therefore we use both of them in this paper.

We report in Table 1 the percentage of defaults in each work. Except Bücker et al. (2021), all other datasets have different degrees of class imbalance, which is regarded as a common characteristic in the credit scoring domain (Thomas et al., 2017).

The class imbalance has an adverse effect on detecting rare events (e.g., the defaulters in credit scoring datasets) in the classification problem. To achieve better classification performance, several imbalanced learning techniques have been proposed. For a review of these methods, see Haixiang et al. (2017), He and Garcia (2009) and Kaur et al. (2019). Focusing on credit risk, Brown and Mues (2012) and Fitzpatrick and Mues (2016) performed a comparison of different classifiers on few credit scoring datasets, and concluded that the Tree-based ensembles such as GB and RF performed relatively well with pronounced class imbalance. Other research also indicated that balancing sample distribution such as using under-sampling or over-sampling techniques can

Table 1: *Summary of papers using model-agnostic interpretation methods in the credit scoring domain*

Authors (year)	ML models	Interpretation methods	Dataset	Class distribution	Method for class imbalance
Ariza-Garzon et al. (2020)	XGBoost, RF	SHAP	P2P lending	20%	Hybrid-sampling
Barbaglia et al. (2021)	XGBoost, GB	ALE plots	Residential mortgages	0.3%-15% (7 countries)	Under-sampling, Over-sampling
Bracke et al. (2019)	GB	Shapley values	Residential mortgages	2.5%	None
Bücker et al. (2021)	SVM, GB	SHAP, LIME, iBreakDown, PDPs	Home equity line of credit	52%	None
Bussmann et al. (2021)	XGBoost	SHAP	P2P lending for SMEs	11%	None
Gramegna and Giudici (2021)	XGBoost	SHAP, LIME	SMEs Loans	1%	Under-sampling
Liu et al. (2022)	XGBoost, Variants of GB, RF, SVM, NN	SHAP	Residential mortgages, Bank loans, P2P lending	6.7%-24% (4 datasets)	None
Martens et al. (2007)	SVM	Rule extraction techniques	Bankruptcy of firms	18%	None
Moscato et al. (2021)	RF, NN	SHAP, LIME, LORE, BEEF, Anchors	P2P lending	23%	Under-sampling, Over-sampling, Hybrid-sampling
Óskarsdóttir and Bravo (2021)	XGBoost	SHAP	Agricultural lending	12%	None
Patil et al. (2020)	XGBoost, NN	SHAP, LIME, LRP	Fraud detection	0.1%	Over-sampling
Szwabe and Misiorek (2018)	GB, RF	Rule extraction techniques	Credit cards	6.7%-30% (4 datasets)	None

improve classification accuracy (Crone & Finlay, 2012; Marqués et al., 2013; Moscato et al., 2021; Namvar et al., 2018). The main analyses in credit risk summarised in Table 1 are in line with the findings of the above literature that they either used Tree-based ensembles as classification models or employed sampling techniques to achieve better predictive power.

Besides analysing the adverse effect of class imbalance on classification performance, some researchers also investigated its impact on the interpretations of the explanatory variables for interpretable models, such as Logistic Regression. King and Zeng (2001a, 2001b) theoretically and empirically showed that the estimation bias of coefficients in Logistic Regression could be greatly magnified by class imbalance. Owen (2007) also suggested that, in the case of extreme class imbalance, the minority class only contributes to the Logistic Regression estimation via its sample mean vector, and this issue cannot be solved by using penalisation or likelihood weighting (Li et al., 2019).

However, to the best of our knowledge, no research has considered the potential effect of class imbalance on the model-agnostic interpretation methods. For example, Patil et al. (2020) obtained a balanced dataset by oversampling, and they applied LIME and SHAP to the balanced dataset to pick out the important features. They concluded that oversampling does not alter the feature correlation since the important features for predictions of valid and fraud observations are consistent. Still, they did not compare the performance of LIME and SHAP on the imbalanced dataset with the balanced one. Bussmann et al. (2021) only mentioned in the conclusion and future research section that it would be interesting to analyse the effects of imbalanced datasets and sampling techniques on their proposed Shapley value context.

Some researchers have already questioned the general robustness of the model-agnostic interpretation methods, such as LIME and SHAP, especially when interpreting the out-of-distribution samples, due to the permutation mechanism and the sampling procedure included in these methods (Alvarez-Melis & Jaakkola, 2018; Shaikhina et al., 2021; Sundararajan & Najmi, 2020; Visani et al., 2021). Some possible remedies to solve this problem were proposed. For example, both Slack et al. (2021) and Zhao et al. (2021) exploited prior knowledge in Bayesian framework and developed Bayesian versions of LIME and SHAP to capture the uncertainty and improve the consistency in interpretations. Li et al. (2020) utilised a fixed reference distribution (e.g., training set) to control the uncertainty brought by permutation in repeated interpretations of SHAP. A similar method is also used by Shankaranarayana and Runje (2019) and Zafar and Khan (2019) for LIME. Although the above research evaluated the robustness of LIME and SHAP, none of them take into account the class imbalance problem. The intuitions could be that the class imbalance may be of no effect on the stability of the interpretations or even mitigate the uncertainty since the interpretations may come from the majority class, which may have more stable distributions, or it could have an adverse effect on the performance of interpretation methods since the rare events may be out-of-distribution and therefore it could be hard to interpret them. Therefore, this paper contributes to filling this gap by designing a novel framework based on feature ranking and value stability indexes to compare the stability of the two most used interpretation methods – LIME and SHAP - under different levels of class imbalance to see how the class imbalance affects the interpretations.

3. Overview of XGBoost, LIME and SHAP

In this section, we present an overview of XGBoost, LIME and SHAP. We start by setting the notation. Let $D = \left\{(\mathbf{x}^i, y^i)\right\}_{i=1}^{N}$ be a dataset with N loans where $\mathbf{x}^i$ is the feature vector, and y^i be a binary response variable with $y = 1$ if default occurs and $y = 0$ otherwise. Let x_j $(j = 1, 2, ..., P)$ be the value of jth feature of $\mathbf{x}^i$. In this section, we briefly introduce XGBoost, the "black-box" machine learning technique chosen to make predictions, and LIME and SHAP, the interpretation methods chosen to explain the predictions provided by XGBoost.

3.1. XGBoost

3.1.1 Brief overview

XGBoost proposed by Chen and Guestrin (2016) is an advanced gradient tree boosting model in the machine learning literature. It shares the concept of gradient boosting algorithm which applies an additive form of weak base learners to minimise the loss function to measure how well the model fits the current data. The general gradient boosting runs a series of iterations m $(m = 1, 2, ..., M)$, where at each iteration m, the base learner f_m is sought by minimising the objective function expressed as follows:

$$min \sum_{i=1}^{N} L(y^i, F_{m-1}(\mathbf{x}^i) + f_m(\mathbf{x}^i)) \tag{1}$$

where $L(\cdot)$ is the loss function, $f_m(\mathbf{x}^i)$ is the base learner and $F_M(\mathbf{x}^i) = \sum_{m=1}^{M} f_m(\mathbf{x}^i)$ is the additive boosted model that represents the prediction on the mth iteration. Specifically, XGBoost uses CART decision tree algorithm as the base learner:

$$f_m(\mathbf{x}^i) = T(\mathbf{x}^i, \Theta) = \sum_{k=1}^{K} w_k I(\mathbf{x}^i \in R_k) \tag{2}$$

where $\{R_k\}_{k=1}^{K}$ denotes K disjoint regions of the feature space (K leaves in a CART), $I(\cdot)$ is an indicator function, with $I = 1$ if and only if $\mathbf{x}^i \in R_k$ and $I = 0$ otherwise, w_k represents the weights on the kth leaf and $\Theta = (w_k, R_k)_{k=1}^{K}$ is a set of unknown parameters that needs to be optimised.

In XGBoost, a regularisation term is added to Equation 1 to avoid overfitting:

$$min \sum_{i=1}^{N} L(y^i, F_{m-1}(\mathbf{x}^i) + f_m(\mathbf{x}^i)) + (\alpha K + \frac{1}{2}\lambda \sum_{k=1}^{K} {w_k}^2) \tag{3}$$

where $\alpha > 0$ is a l_1-penalty on the number of leaves in a CART as shown in Equation 2, $\lambda > 0$ is a l_2-penalty on the leaf nodes weights w_k. Compared to the general gradient boosting algorithm, XGBoost quickly approximates Equation 3 with a second-order Taylor expansion and it further speeds the convergence during the model training by applying an approximate greedy algorithm for finding the optimal tree structure. For more details about XGBoost see Chen and Guestrin (2016) and Xia et al. (2017).

Table 2: *XGBoost hyper-parameters tuning grid*

Hyper-parameters	**Grid setting**
Maximum tree depth	1, 3, 5
Sample-based subsampling rate	0.6, 0.8, 1
Feature-based subsampling rate	0.6, 0.8
Minimum child weight	1, 3, 5
No. of iterations	100, 110, 120,130,..., 500

3.1.2 Parameter tuning

Implementation of XGBoost requires setting several hyper-parameters. For example, the learning rate and the number of iterations are two hyper-parameters, which control the model convergence to make the boosting process more robust to over-fitting. A lower learning rate generally requires a larger number of iterations to ensure a sufficient convergence. Other hyper-parameters such as the penalty terms α and λ introduced in Equation 3, maximum tree depth, feature-based and sample-based subsampling rates, enable XGBoost to control the tree complexity, thereby avoiding overfitting and accelerating learning. This paper uses the Python package *xgboost* (Chen & Guestrin, 2016) to conduct the learning process.

Considering a trade-off between the model performance and the computational cost, we employ a stepwise parameter tuning process, which is also used in Xia et al. (2017). First, we follow the learning rate 0.1 as suggested by Friedman (2001) and Xia et al. (2017), and we fix the number of iterations at 200. The rest of the hyper-parameters are tuned using a grid search method. Afterwards, we keep the learning rate at 0.1, and tune the number of iterations with the the rest of the hyper-parameters fixed at the values obtained from the first step. Table 2 reports the parameter tuning grid and other parameters that are not involved will be set to the default values in the *xgboost* Python package. The parameter values selected for the tuning grid are based on the preliminary exploration and similar research using gradient boosting algorithms in the credit scoring domain (Barbaglia et al., 2021; Chang et al., 2018; Fitzpatrick & Mues, 2016; Gunnarsson et al., 2021; Xia et al., 2017). In each step, we use five-fold cross-validation to find the best combination of the hyper-parameters with the highest H-measure obtained on the validation data.

We use H-measure in Python package *hmeasure* to evaluate the predictive performance of XGBoost since it is a preferred metric for the imbalanced dataset (Calabrese et al., 2016; Fitzpatrick & Mues, 2016; Lessmann et al., 2015). As a coherent alternative to the Area Under the Curve (AUC), H-measure allows one to specify a severity ratio, measured by the cost of misclassifying a class 0 data point to the cost of misclassifying a class 1 data point (Hand, 2009, 2010). This paper uses the default setting of the severity ratio which equals to the reciprocal of relative class frequency, so that misclassifying the rare class (class 1 data points) is considered a more severe mistake, which exactly suits the credit scoring applications.

3.2. LIME

LIME proposed by Ribeiro et al. (2016) aims to interpret the machine learning model prediction of a specific target $\mathbf{x}^i$ by appropriating the "black-box" machine learning model ($f : \mathbb{R}^d \rightarrow \mathbb{R}$) with a local interpretable model $g \in G$, where G is a family of possible interpretable models. To fit a local surrogate centred on $\mathbf{x}^i$, LIME generates a new dataset (neighbourhood around $\mathbf{x}^i$) by randomly perturbing features from the target $\mathbf{x}^i$ and obtaining the corresponding predictions from the "black-box" model. The interpretable model g is then trained on the new dataset $\mathcal{Z}$, which is weighted by the distances from the perturbed samples to the target $\mathbf{x}^i$. Therefore, the learned interpretable model g ensures local fidelity, which means it should be a good approximation of the "black-box" model predictions locally (focusing on the target $\mathbf{x}^i$) but does not guarantee a good global approximation. Mathematically, the interpretation for the target $\mathbf{x}^i$ can be obtained by optimising the following objective function:

$$argminL(f, g, \pi_{\mathbf{x}^i}) + \Omega(g)$$

where $\pi_{\mathbf{x}^i}$ represents a proximity measure between a sample to the target $\mathbf{x}^i$, so we need to define the neighborhood around $\mathbf{x}^i$. $\Omega\,(g)$ is a regulation term which measures the complexity of the interpretable model g, $L(\cdot)$ is a loss function which is minimised to get an interpretable model g most similar to f in the neighbourhood defined by $\pi_{\mathbf{x}^i}$, while the model complexity $\Omega(g)$ is kept low:

$$L(f, g, \pi_{\mathbf{x}^i}) = \sum_{\mathbf{z},\mathbf{z}' \in \mathcal{Z}} \pi_{\mathbf{x}^i}(\mathbf{z})(f(\mathbf{z}) - g(\mathbf{z}'))^2$$

where $\mathbf{z}'$ denotes the simplified inputs and $\mathbf{z}' \in \{0,1\}^{P'}$[4]. In this paper, we follow the default setting in the Python package *lime* for applying LIME to tabular data. Specifically, Ridge Regression is used as the interpretable model g. $\pi_{\mathbf{x}^i}(\mathbf{z}) = exp(-D(\mathbf{x}^i, \mathbf{z})^2/\sigma^2)$ is an exponential kernel, which smoothly assigns higher weights to samples closer to the target of interest. Here we use the default setting in the *lime* package, where D represents the Euclidean distance function and σ is the kernel width $0.75 \times \sqrt{P}$. To control the model complexity $\Omega(g)$, a feature selection step is performed initially to select 10 most important features to be used in the Ridge regression.

The Ridge regression estimates a linear relationship between the selected features and the approximated predictions, which provides an interpretation of the prediction through its coefficients: the larger the absolute coefficient values, the larger variation in the value of the response variable when the feature is changed. Therefore, the absolute coefficient values of features can represent the feature importance value. By sorting the absolute coefficient values in a decreasing order, we can get a feature ranking list, which will be used for the LIME stability measurement.

[4] Interpretation methods often use simplified inputs $\mathbf{x}' \in \{0,1\}^{P'}$ that map to the original inputs through a mapping function $\mathbf{x} = h_x(\mathbf{x}')$. Local methods try to ensure $g(\mathbf{z}') \approx f(h_x(\mathbf{z}'))$ whenever $\mathbf{z}' \approx \mathbf{x}'$.

3.3. SHAP

SHAP (Lundberg et al., 2020; Lundberg & Lee, 2017) is based on the Shapley value (Shapley, 1953), a concept from game theory that assigns fair payout to a player depending on its contribution to the total gain when coalitions are taken into account. In the machine learning field, a player is a certain feature value, coalitions are possible feature subsets S, and the fair payout represents the contribution of a certain feature value to the prediction. Thus, in our context, the Shapley value ϕ_j of a feature value x_j in the target $\mathbf{x}^i$ can be calculated by averaging the prediction differences (contribution) generated between the model with and without j over all possible feature subsets S and considering all feature orderings:

$$\phi_j = \sum_{S \subseteq \{1,\ldots,P\}} \frac{|S|!(P-|S|-1)!}{P!}[f_S(x_S) - f_{S\setminus j}(x_{S\setminus j})] \tag{4}$$

where x_S denotes the values of the input features in the set S of target $\mathbf{x}^i$, $f_S(x_S)$ denotes the model trained with x_j present in x_S, and $f_{S\setminus j}(x_{S\setminus j})$ denotes the model trained without x_j in x_S. It is proved that Shapley values is the only explanation method in the broad class of additive feature attribution methods that could simultaneously satisfy three properties — local accuracy, missingness and consistency (Lundberg & Lee, 2017).

Unfortunately, as the number of the features increases, averaging over all possible feature subsets will be an intractable problem, hence sampling-based approximation methods are always applied to solve Equation 4 (Lundberg & Lee, 2017; Štrumbelj & Kononenko, 2014). However, those approximation methods reply on post hoc modelling of an arbitrary function and thus can still be slow and also suffer from sampling variability.

Lundberg et al. (2020) therefore proposed Tree SHAP that could provide a fast and exact computation of Shapley values by leveraging the internal structure of tree-based models such as XGBoost. Shapley values require a summation of prediction differences over all possible feature subsets. Tree SHAP collapses this summation into a set of calculations specific to each leaf in a tree in the tree-based models, hence reducing the complexity of exact Shapley value computation from exponential to polynomial time.

Specifically, to compute the impact of a specific feature subset ($f_S(x_S)$ in Equation 4) during the Shapley value calculation, Tree SHAP uses interventional expectations over a user-supplied background dataset:

$$f_S(x_S) = E[f(X) \mid do(X_S = x_S)] \tag{5}$$

where X represents a sample in the background dataset, and the do-notation formulation emphasizes an intervention on X when we manually set the feature values in X_S to the same values in x_S of the target $\mathbf{x}^i$ ($X_S = x_S$). The interventional Tree SHAP enforces independence between the set S and the set of remaining features based on the laws of causality (Janzing et al., 2019). It should be noted that since the background dataset is fixed, Tree SHAP calculates Equation 5 by iterating over each sample in the background dataset, hence there is no estimation variability in Tree SHAP like other sampling-based approximation methods (see Lundberg et al. (2020) for more algorithm details).

In this paper, we use the training set as the background dataset and we use the Python package *shap* to apply Tree SHAP (abbreviated as SHAP in this paper) in our experiments. For each target $\mathbf{x}^i$, the absolute SHAP value of each feature measures the magnitude of contribution to the prediction and therefore can be regarded as the feature importance value. Features with larger absolute SHAP values contribute more to the prediction and therefore are more important. By sorting the absolute SHAP values in a decreasing order, we could get a ranking list of features, which will be used for SHAP stability measurement.

3.4. LIME and SHAP comparison

Both LIME and SHAP measure feature contribution at the observation level (local explanation). Benefiting from the solid theoretical foundation in game theory, SHAP ensures that the prediction is fairly distributed among the features, and therefore could further provide global model interpretations such as global feature importance for a whole dataset, which are consistent with the local explanations (Lundberg & Lee, 2017). Based on the solid theory, SHAP also guarantees contractive explanations by comparing the prediction with the average prediction, which is not feasible for LIME. Therefore, SHAP is more suitable and compliant when people need full interpretations locally and globally based on a solid theoretical foundation.

While LIME lacks the theoretical foundation as SHAP and tends to be "internal" unstable (this will be discussed in Section 5.3.3), LIME is time efficient and can make human-friendly explanations compared to SHAP. Tree SHAP is relatively faster compared to other types of SHAP by leveraging the internal structure of tree-based models. Machine learning models other than tree-based models can only rely on other SHAP algorithms, such as Kernel SHAP, which is slow and impractical to use when computing Shapley values for many observations (Molnar, 2021). LIME provides short and clear explanations using the feature selection step. An interpretable model, such as Ridge regression in LIME, also enables LIME to make statements about changes in prediction for changes in the input, which is not applicable in SHAP. Besides, LIME is capable of providing interpretations for a new observation only based on properties of the training set (mean and standard deviation) and the machine learning prediction function, while SHAP needs to access the data (training set or other background datasets) to compute the Shapley values (Lundberg & Lee, 2017; Ribeiro et al., 2016). Therefore, LIME is more appropriate in applications when the recipient of interpretations is a lay person or with a limitation of time, or there is a restriction of access to data.

4. Experimental setup

To analyse the effects of the class imbalance on LIME and SHAP, this paper conducts an empirical study, which compares the performance of LIME and SHAP when explaining the predictions of XGBoost on credit scoring datasets with different default rates. The proposed experimental framework is provided in Figure 1. In this section, we use the European Datawarehouse data as an example to explain the experimental framework.

In **Step 1**, loan data for residential property purchased in the UK are collected and pre-processed for the following analysis. The detailed data preparation process will be introduced in Section 4.1. Since LIME and SHAP interpret the predictions on individuals, to make the stability of the interpretive performance across varied

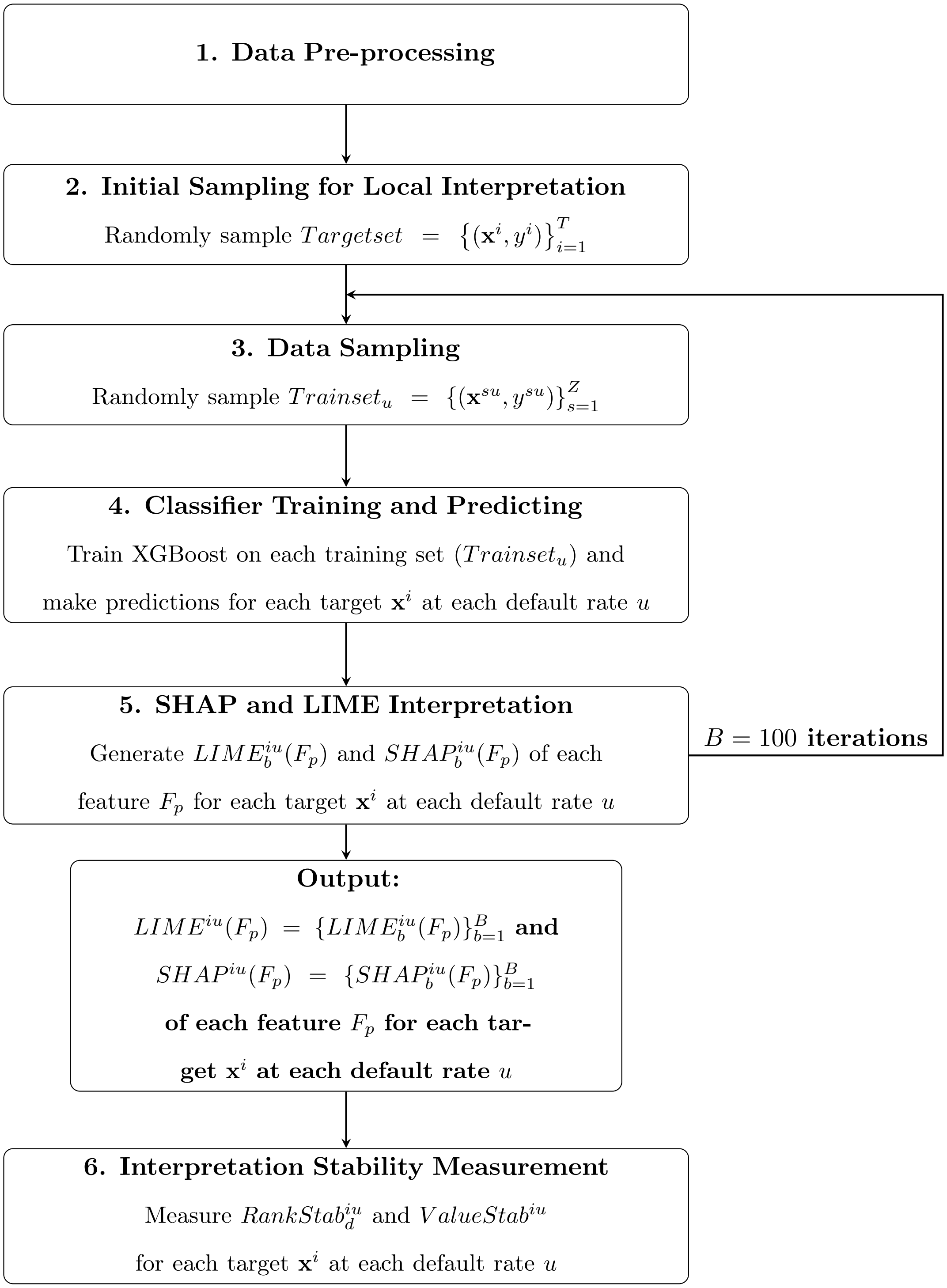

1. Data Pre-processing
2. Initial Sampling for Local Interpretation
Randomly sample $Targetset = \{(\mathbf{x}^i, y^i)\}_{i=1}^{T}$
3. Data Sampling
Randomly sample $Trainset_u = \{(\mathbf{x}^{su}, y^{su})\}_{s=1}^{Z}$
4. Classifier Training and Predicting
Train XGBoost on each training set ($Trainset_u$) and make predictions for each target $\mathbf{x}^i$ at each default rate u
5. SHAP and LIME Interpretation
Generate $LIME_b^{iu}(F_p)$ and $SHAP_b^{iu}(F_p)$ of each feature F_p for each target $\mathbf{x}^i$ at each default rate u
$B = 100$ iterations
Output:
$LIME^{iu}(F_p) = \{LIME_b^{iu}(F_p)\}_{b=1}^{B}$ and
$SHAP^{iu}(F_p) = \{SHAP_b^{iu}(F_p)\}_{b=1}^{B}$
of each feature F_p for each target $\mathbf{x}^i$ at each default rate u
6. Interpretation Stability Measurement
Measure $RankStab_d^{iu}$ and $ValueStab^{iu}$ for each target $\mathbf{x}^i$ at each default rate u

Figure 1: *Proposed experimental framework*

class imbalance comparable, it is necessary to initially sample the target individuals of which the predictions from the "black-box" model will be interpreted. Therefore, we randomly select 100 defaults[5] and 100 non-defaults

[5] The reason why we choose 100 defaults is based on the number of defaults in the sample, which is 3,229. When

to be used as targets ($Targetset = \left\{(\mathbf{x}^i, y^i)\right\}_{i=1}^{T}, T = 200$) for local interpretation in **Step 2**. After the initial sampling, for the rest of the data, we build 12 training sets ($Trainset_u = \{(\mathbf{x}^{su}, y^{su})\}_{s=1}^{Z}$) with the same sample size Z but different default rates u ($u = 1\%, 2.5\%, 5\%, 10\%, ..., 45\%, 50\%$) in **Step 3**. By doing so, it is possible to identify whether the interpretations of machine learning predictions generated by LIME and SHAP are adversely affected when there is a substantially lower number of observations in one of the classes. The sampling procedure will be explained in Section 4.2. In **Step 4**, XGBoost is trained on the training sets generated from Step 3 with the best parameters selected using the grid search method introduced in Section 3.1.2. The trained XGBoost with parameters selected with the highest H-measure value is then applied to get the predictions on each target $\mathbf{x}^i$ obtained in Step 2. In **Step 5**, the feature importance value, referring to the absolute LIME coefficient value $LIME_b^{iu}(F_p)$ or the absolute SHAP value $SHAP_b^{iu}(F_p)$ of each feature F_p in the feature set $F = \{F_1, ..., F_P\}$, is generated for each target $\mathbf{x}^i$ at each default rate u. We repeat the process from Step 3 to Step 5 $B = 100$ times. After iterations, for each target $\mathbf{x}^i$ at each default rate u, we obtain 100 feature importance values for each feature F_p, denoted as $LIME^{iu}(F_p) = \{LIME_b^{iu}(F_p)\}_{b=1}^{B}$ or $SHAP^{iu}(F_p) = \{SHAP_b^{iu}(F_p)\}_{b=1}^{B}$. Then in **Step 6**, for each target $\mathbf{x}^i$ at each default rate u, we first measure the ranking and value stability for each feature F_p, based on its 100 feature importance values[6], noted as $RankStab^{iu}(F_p)$ and $ValueStab^{iu}(F_p)$. Then we calculate the final ranking stability for each target $\mathbf{x}^i$ at each default rate u, denoted as $RankStab_d^{iu}$, by aggregating ranking stability indexes of features at each list depth d of feature ranking lists. Simultaneously, the final value stability for each target $\mathbf{x}^i$ at each default rate u, denoted as $ValueStab^{iu}$, is measured by the average of value stability indexes of all features in the feature set $F = \{F_1, ..., F_P\}$. The details of stability indexes will be introduced in Section 5. Note that we repeat the whole experiment 5 times to obtain the stability measurements for 5 sets of targets (5×200 targets) to avoid any potential bias results when only depending on 200 targets. The stability measurements results for 5 sets are reported in Section 6.

4.1. Data pre-processing

In this empirical study, we use UK residential mortgage data between January 2016 and December 2020, collected from European Datawarehouse. Here a mortgage default is defined as being in arrears for three months or more with mortgage payments. The loan-level data provides loan characteristics, borrower information, property information and loan performance for each loan. In particular, the loan characteristics include some static information such as loan original balance, and some dynamic information such as the current interest rate. The

initially sampling 100 defaults, we have a sample size Z equals to $(3,229 - 100) \times 2 = 6,258$, with a reasonable number of defaults in the dataset with 1% default rate, which equals to 63 ($6,258 \times 1\%$). A larger initially sampled number of defaults will lead to fewer defaults in the dataset with 1% default rates. Please refer to Section 4.2 for more details of the sampling process.

[6] We present some examples in the Supplementary Materials (Part B) that show how the distribution of absolute SHAP values for specific features over 100 iterations could change when the default rate increases from 1% to 50% for a specific target. This further demonstrates the importance of studying the effects of class imbalance on the interpretations

borrower information contains the employment status, age, annual income, etc., of the loan borrower collected at the origination of the loan. The underlying assets information provides the type, original value, current value (dynamic), location, etc., of the property. Loan performance information provides the status of the loan, whether it is performing or in arrears, and for how many months it has been in arrears, which is also dynamic. Note that the dynamic information is updated at least quarterly.

Table 3: *Explanatory variables*

Explanatory Variable	**Type**	**Description**
Loan Characteristics		
Loan Seniority	Numeric	Number of months since the loan granted
Current Loan Balance	Numeric	Unpaid principal balance
Interest Rate Type	Categorical	Floating rate, Discount rate, Fixed rate or fixed rate with a compulsory future switch to floating
Interest Rate	Numeric	Current interest rate
Re-mortgage	Categorical	If is a re-mortgage loan - Yes, No
Repayment Method	Categorical	Interest only, Repayment, Mixed principal and interest
DTI	Numeric	Original debt-to-income ratio
Borrower Information		
Age of Borrower	Numeric	(Primary) borrower age
Gross Income	Numeric	Sum of primary income and secondary income (if reported)
Employment Type	Categorical	Employed, Self-employed, Other
Single Borrower	Categorical	If is single borrower - Yes, No
First Time Buyer	Categorical	If is first time buyer - Yes, No
Property Information		
Property Type	Categorical	Bungalow, Terraced house, Flat/Apartment, House (detached)
LTV	Numeric	Original loan-to-value
CLTV	Numeric	Current loan-to-value
Region	Categorical	10 English regions, Wales, Scotland, Northern Ireland, Not known

To select the explanatory variables as close as possible to a real scenario, we refer to several published papers that used the same or similar database. Barbaglia et al. (2021) is a paper from the European Commission, which used the residential mortgage dataset from the European Datawarehouse as we did. Bracke et al. (2019) is a paper from the Bank of England, which similarly conducted a UK residential mortgage default analysis. Li et al. (2019) and Sirignano et al. (2018) also predicted the mortgage loan default with similar features as we have, although they used mortgage data from the U.S.

Table 4: *Data Collection Time*

Collect explanatory variable	2016Q1	2016Q2	2016Q3	2016Q4	2017Q1	2017Q2	2017Q3	2017Q4
Collect response variable	2017Q1	2017Q2	2017Q3	2017Q4	2018Q1	2018Q2	2018Q3	2018Q4
Collect explanatory variable	2018Q1	2018Q2	2018Q3	2018Q4	2019Q1	2019Q2	2019Q3	2019Q4
Collect response variable	2019Q1	2019Q2	2019Q3	2019Q4	2020Q1	2020Q2	2020Q3	2020Q4

As Barbaglia et al. (2021) indicated, European Datawarehouse data is rich but unexplored. It comes with some flaws that need to be addressed before using it. The main flaws include: inactive or matured loans still exist in the database; some loans are inconsistent across time period analysed; and some loans have unexpected attributes such as very high interest rate (e.g., over 20%). In order to address these flaws, we implemented an intensive data cleaning process based on our understanding of the data, combined with the data cleaning steps mainly introduced in Barbaglia et al. (2021) and Bracke et al. (2019). The data cleaning details can be found in Supplementary Materials (Part A). Table 3 reports the explanatory variables we used in the loan default predictive model.

Since the database is updated at a quarter level and we need to predict the loan status (1 = default, 0 = non-default) one year ahead, here we identify the loan status (i.e. response variable) quarterly and collected the explanatory variables one year in advance to build our dataset. The detailed data collection time is shown in Table 4. Note that we first exclude those loans that were already defaulted in 2016 Q1 – Q4 from the whole dataset since we do not have data in 2015 to collect their explanatory variables. After this data collection process, we have a dataset in which for each loan there are loan records for a series of quarters. Based on this dataset, when a loan is first found to be in default in a certain quarter in the period 2017Q1 to 2020Q4, we remove its records in other quarters to ensure that there is no duplicated defaulted loan in the dataset and avoid the possibility that the defaulted loan is also shown as a non-defaulted one in a different period. Since we do not consider the impact of the time change on the prediction and interpretation results in this study, we also remove the duplicated non-defaulted loans, in other words, we randomly select one record for each non-defaulted loan in the dataset (Calabrese et al., 2016). As a result of this selection, we obtain a dataset with all distinct loans, including 3,229 defaulted loans. The default rate is 0.6%, which indicates the highly unbalanced nature of the mortgage default variable as stated in Thomas et al. (2017).

4.2. Sampling procedure

After initially sampling the targets, which will also be used as a test set to evaluate accuracy, we perform data resampling to create the training sets with various loan default rates ranging from 5% to 50% in increments of 5%. We also include two more extreme loan default rates which are 2.5% and 1%. To exclude the effect of sample size on the predictive and interpretive performance, we fix the sample size to $Z = 6,258$, which is the size of a balanced dataset (loan default rate = 50%) with all 3,129 defaults included and 3,129 randomly selected non-defaults. Note that we do not use any over-sampling method such as SMOTE (Harald et al., 2016) to create more records of defaults and to artificially increase the sample size since this method involves randomness that might generate unseen features with a stochastic approach that would add an additional level of complexity to the problem (Bueff et al., 2022).

Table 5: *Datasets Structure*

Default Rate	Number of Defaults	Number of Non-defaults	Total
1%	63	6,195	6,258
2.5%	156	6,102	6,258
5%	313	5,945	6,258
10%	626	5,632	6,258
15%	939	5,319	6,258
20%	1,252	5,006	6,258
25%	1,565	4,693	6,258
30%	1,877	4,381	6,258
35%	2,190	4,068	6,258
40%	2,503	3,755	6,258
45%	2,816	3,442	6,258
50%	3,129	3,129	6,258

Therefore, with the sample size fixed at 6,258, we under-sample the defaults as well as randomly select more non-defaults to obtain an increasing level of class imbalance. The number of the defaults and non-defaults for each default rate are shown in Table 5.

5. Stability measurement

In this paper, we use Sequential Rank Agreement (SRA) proposed by Ekstrøm et al. (2019) to measure the ranking stability of feature lists generated by LIME and SHAP for each target $\mathbf{x}^i$ at each default rate u ($RankStab^{iu}$ in Figure 1). Based on our knowledge, this paper is the first research that uses this method to compare the LIME and SHAP feature ranking lists. The details of SRA can be found in Section 5.1. Besides evaluating the feature ranking stability, we also measure the feature importance value stability generated by LIME and SHAP for each target $\mathbf{x}^i$ at each default rate u ($ValueStab^{iu}$ in Figure 1) using the Coefficient of Variation (CV), as explained in Section 5.2.

For the sake of completeness, we also include another two stability measures, namely Variables Stability Index (VSI) and Coefficients Stability Index (CSI), that have been used before in the literature only for LIME (Visani et al., 2021). The VSI is proposed to check whether the selected features are the same or not among the repeated LIME interpretations. The CSI measures the LIME stability through the similarity of coefficients among the repeated LIME interpretations. The details of VSI and CSI are introduced in Section 5.3.

Both SRA and VSI are based on feature variation, but SRA takes into account the feature's position in the ranking lists. SRA can be used for both SHAP and LIME but VSI can only apply to LIME since VSI focuses on feature selection step which is not included in SHAP. Both CV and CSI focus on feature importance value stability. CV considers the feature coefficient value itself whereas CSI only checks whether the confidence intervals of coefficients for the same feature overlap or not in different LIME interpretations. Note that Visani et al. (2021) proposed VSI and CSI originally to check the "internal" stability of LIME, which refers to the stability of explanations derived from repeated LIME calls under the same conditions and the same distribution (law). As mentioned in Section 3.3, there should be no "internal" instability (estimation variability) in Tree SHAP like LIME since the background dataset is fixed. Therefore, in this paper, we consider VSI and CSI specifically for LIME to examine the class imbalance effects on the "internal" stability and compare them with the "external" stability for LIME. More details are explained in Section 5.3.3.

We remind that b ($b = 1, ..., B$) represents the iteration with $B = 100$ and $F = \{F_1, ..., F_P\}$ the feature set in this section.

5.1. Sequential Rank Agreement

The SRA value could provide the level of ranking agreement using a function of the depth in the lists. Following our experimental framework (Figure 1), for each target $\mathbf{x}^i$ at each default rate u, we obtain 100 feature importance values for each feature after repeating Step 3 to Step 5 100 times. In other words, we obtain 100 sets of feature importance values of all features for each target $\mathbf{x}^i$ at each default rate u. Therefore, by sorting the feature importance values in each set in decreasing order, the corresponding ordered feature names could form 100 feature ranking lists. Therefore, in what follows, we describe the process of obtaining SRA values based on 100 feature ranking lists for one target $\mathbf{x}^i$ at a specific default rate u.

We denote the feature importance value of a feature as $\Omega_b^{iu}(F_p)$[7]. Therefore, a set of feature importance values of all features can be denoted as $\Omega_b^{iu} = \left\{\Omega_b^{iu}(F_1), ..., \Omega_b^{iu}(F_P)\right\}$. By sorting the feature importance values in Ω_b^{iu} in decreasing order, we obtain the feature ranking list L_b^{iu}. Every feature ranking list L_b^{iu} contains the same number of features. We point out that in the SHAP feature ranking lists, the number of features equals to that used in the "black-box" machine learning model since SHAP uses all features involved in the predictive model to interpret the prediction result. While in LIME feature ranking lists, the number of features equals to

[7] In our experimental framework (Figure 1), $\Omega_b^{iu}(F_p)$ would be either $LIME_b^{iu}(F_p)$ or $SHAP_b^{iu}(F_p)$.

Table 6: *Example set of ranking lists shows: (a) three sets of feature importance values (Ω_1, Ω_2, Ω_3) with the CV values of features ($ValueStab(F_p)$), (b) three feature ranking lists (L_1, L_2, L_3) based on Panel (a), (c) ranking $R(F_p)$ obtained by each feature in each of three lists with the ranking agreement values of features ($RankStab(F_p)$), and (d) the cumulative set of features up to a given depth in the three ranking lists (i.e., a feature is added to $S(d)$ whenever it appears in at least one list) with the SRA value $RankStab_d$ of each list depth d.*

(a)

Feature	Ω_1	Ω_2	Ω_3	$ValueStab(F_p)$
A	5	6	5	0.11
B	4	4	7	0.35
C	3	5	6	0.33
D	2	3	3	0.22
E	1	2	4	0.65

(b)

Ranking	L_1	L_2	L_3
1	A	A	B
2	B	C	C
3	C	B	A
4	D	D	E
5	E	E	D

(c)

Feature	R_1	R_2	R_3	$RankStab(F_p)$
A	1	1	3	1.33
B	2	3	1	1
C	3	2	2	0.33
D	4	4	5	0.33
E	5	5	4	0.33

(d)

Depth (d)	S_d	$RankStab_d$
1	$\{A, B\}$	2.33
2	$\{A, B, C\}$	2.66
3	$\{A, B, C\}$	2.66
4	$\{A, B, C, D, E\}$	3.32
5	$\{A, B, C, D, E\}$	3.32

the number of unique features selected during the feature selection step in all 100 LIME interpretable models. We further denote a ranking function of a feature ranking list L_b^{iu} as $R_b^{iu} : \{F_1, ..., F_P\} \to \{1, ..., P\}$, such that $R_b^{iu}(F_p)$ is the ranking of a feature F_p in a feature ranking list L_b^{iu}, as illustrated in Table 6. For example, Ω_1 in Panel (a) is a set of feature importance values of 5 features, L_1 in Panel (b) is the feature ranking list based on Ω_1, and the value of R_1 in Panel (c) is the corresponding ranking of each feature in L_1. Thus, for each feature F_p, the ranking agreement value $RankStab^{iu}(F_p)$ across B feature ranking lists can be calculated as follows:

$$RankStab^{iu}(F_p) = \frac{1}{B-1} \sum_{b=1}^{B} \left(R_b^{iu}(F_p) - \overline{R}^{iu}(F_p) \right)^2 \quad (6)$$

where $\overline{R}^{iu}(F_p) = \frac{1}{B} \sum_{b=1}^{B} R_b^{iu}(F_p)$ is the expected ranking of the feature F_p over B feature ranking lists. Therefore, the ranking agreement value of features A, B, C, D and E in Panel (c) of Table 6 is 1.33, 1, 0.33, 0.33 and 0.33 respectively. $RankStab^{iu}(F_p)$ can be interpreted as the expected Euclidean distance of the individual rankings from the expected ranking over all the lists for each feature F_p.

As shown in Panel (d) of Table 6, we define the list depth d as an integer, with $1 \leq d \leq P$, then S_d^{iu} denotes the set of unique features ranked less than or equal to the list depth d in all the feature ranking lists in the set L^{iu}. Hence, after obtaining the ranking agreement values $RankStab^{iu}(F_p)$ for all features in the set F, we could obtain the SRA value $RankStab_d^{iu}$ of each list depth d by calculating the weighted expected ranking agreement values of all features in the set S_d^{iu}:

$$RankStab_d^{iu} = \frac{\sum_{F_p \in S_d^{iu}} (B-1) RankStab^{iu}(F_p)}{(B-1)|S_d^{iu}|} \tag{7}$$

where $|S_d^{iu}|$ is the cardinality of the set S_d^{iu}. Therefore, the SRA value of list depth 1 to 5 in Panel (d) of Table 6 is 2.33, 2.66, 2.66, 3.32 and 3.32 respectively. The SRA value $RankStab_d^{iu}$ in each list depth d is equivalent to the pooled variance of features found in S_d^{iu}.

When comparing SRA values for one target under the same list depth among different class imbalance levels, a smaller SRA value suggests the feature ranking lists agree more on the rankings, which means the feature rankings are more stable. For example, the $RankStab_d^{iu}$ of every list depth will be 0 when the ranking lists are identical.

5.2. Coefficient of Variation

The CV is a statistical measure of relative variability, defined as the ratio of the standard deviation to the mean among a set of data points. The CV is particularly suitable for our study as it is dimensionless and thus, comparable among different sets of data points with various means or various units.

We continue using the notations in Section 5.1. The CV value for a feature F_p ($ValueStab^{iu}(F_p)$) across B sets of feature importance values can then be calculated as follows:

$$ValueStab^{iu}(F_p) = \frac{\sqrt{\frac{\sum_{b=1}^{B} (\Omega_b^{iu}(F_p) - \overline{\Omega}^{iu}(F_p))^2}{B-1}}}{\overline{\Omega}^{iu}(F_p)} \tag{8}$$

where the numerator is the sample standard deviation of the feature importance values for the feature F_p over B sets of feature importance values, and the denominator $\overline{\Omega}^{iu}(F_p) = \frac{1}{B}\sum_{b=1}^{B} \Omega_b^{iu}(F_p)$, is the expected feature importance value of the feature F_p over B sets of feature importance values. For example, the CV values of 5 features in Table 6 are shown in Panel (a).

After getting the CV value $ValueStab^{iu}(F_p)$ for each feature F_p, the CV value for a target $\mathbf{x}^i$ at a default rate u, which is defined as $ValueStab^{iu}$, can be obtained by calculating the average of CV values $ValueStab^{iu}(F_p)$ of all the features whose importance values presenting at least 2 times[8] in the B sets of feature importance values,

[8] When calculating the CV value for each target $\mathbf{x}^i$ at a default rate u, we do not consider the feature which only presents once in all B explanation models since it will have only one feature importance value and the CV value $ValueStabFeat^{iu}(F_p)$ of this feature will be 0, which cannot truly represent a meaningful degree of variability. Therefore, P^* in Equation 9 will be less than or equal to the number of unique features in B interpretable models.

which can be expressed as follows:

$$ValueStab^{iu} = \frac{1}{P^*} \sum_{p=1}^{P^*} ValueStab^{iu}(F_p). \tag{9}$$

Since the CV value $ValueStab^{iu}$ measures the degree of variability in the feature importance values, when comparing CV values for one target among different class imbalance levels, a larger CV value indicates less stable interpretations of the prediction made for the chosen target.

5.3. Variables Stability Index & Coefficients Stability Index

In this section, we introduce the VSI and CSI proposed by Visani et al. (2021) to measure the stability of LIME. We start by setting the notation. We consider $g_1^{iu}, ..., g_M^{iu}$ as M interpretable models (Ridge regressions) generated by LIME for a target $\mathbf{x}^i$ at a default rate u.

5.3.1 VSI

The VSI is proposed to check the stability of the feature selection step included in LIME — whether the selected features are same among M interpretable models. Let $C^{iu} = \{C_1^{iu}, ..., C_K^{iu}\}$ be the set of all possible combinations of the M interpretable models, two by two. The generic element of the set C^{iu} is the pair of interpretable models $C_k^{iu} = (g_\alpha^{iu}, g_\beta^{iu})$ and the number of pairs K in the set C^{iu} equals to $\frac{M!}{2!(M-2)!}$. For each pair C_k^{iu}, we count the number of the same features used by both interpretable models, denoted by $SAME(C_k^{iu})$. Note that $SAME(C_k^{iu})$ is an integer and $0 \leq SAME(C_k^{iu}) \leq 10$[9]. Hence the VSI value vsi^{iu} for a target $\mathbf{x}^i$ at a default rate u can be calculated as follows:

$$vsi^{iu} = \frac{\frac{1}{K}\sum_{k=1}^{K} SAME(C_k^{iu})}{10} \tag{10}$$

where the numerator calculates the average number of same features used by all pairs of interpretable models in the set C^{iu}. We further normalise dividing by the number of selected features 10, and obtain the VSI value vsi^{iu} for one target ranging from 0 to 1. The more it approaches 1, the more features found in M interpretable models are the same.

5.3.2 CSI

The CSI measures the LIME stability through the similarity of coefficients generated from M interpretable models for a target $\mathbf{x}^i$ at a default rate u. Specifically, we calculate 95% confidence intervals of each coefficient in M interpretable models (see Visani et al. (2021) for more mathematical explanation), and consider the coefficients for a feature to be unstable when the calculated confidence intervals for this feature from different interpretable

[9] The maximum value of $SAME(C_k^{iu})$ is the number of features selected in the feature selection step, which equals to 10.

models are not overlapped at all. Instead, we consider the coefficients of a feature to be stable whenever the confidence intervals overlap to some extent. In what follows, we use equations to explain the CSI.

The comparison among confidence intervals is carried out separately for each feature F_p. Therefore, let $CI^{iu}(F_p) = \{CI_1^{iu}(F_p), ..., CI_D^{iu}(F_p)\}$ be the set of all 95% confidence intervals of the coefficients for a certain feature F_p presented in the M interpretable models. Let $A^{iu}(F_p) = \{A_1^{iu}(F_p), ..., A_T^{iu}(F_p)\}$ be the set of all possible combinations of the D confidence intervals of the coefficients for the feature F_p, two by two. The generic element of the set $A^{iu}(F_p)$ is the pair of confidence intervals $A_t^{iu}(F_p) = (A_\alpha^{iu}(F_p), A_\beta^{iu}(F_p))$ and the number of pairs T in the set $A^{iu}(F_p)$ equals to $\frac{D!}{2!(D-2)!}$. Hence for each pair $A_t^{iu}(F_p)$, we consider a binary variable $OVERLAP(A_t^{iu}(F_p))$, which equals to 1 if the pair of confidence intervals is overlapped and 0 otherwise:

$$OVERLAP(A_t^{iu}(F_p)) = \begin{cases} 0 & \text{If } A_\alpha^{iu}(F_p) \cup A_\beta^{iu}(F_p) = \emptyset \\ 1 & \text{Otherwise.} \end{cases}$$

We calculate $OVERLAP(A_t^{iu}(F_p))$ for all pairs of confidence intervals in the set $A^{iu}(F_p)$ and add them up. The outcome is a count variable, which we normalise dividing by the number of pairs T, to obtain a Partial Index $PI^{iu}(F_p)$ for the feature F_p considered:

$$PI^{iu}(F_p) = \frac{1}{T}\sum_{t=1}^{T} OVERLAP(A_t^{iu}(F_p)).$$

To obtain the CSI value csi^{iu} for a target $\mathbf{x}^i$ at a default rate u, we average the Partial Indices of all the features presenting at least 2 times[10] in the M interpretable models, which can be expressed as follows:

$$csi^{iu} = \frac{1}{P^*}\sum_{p=1}^{P^*} PI^{iu}(F_p). \tag{11}$$

Similar to the VSI value vsi^{iu}, the CSI value csi^{iu} for one target also ranges from 0 to 1, and the more it approaches to 1, the more the LIME coefficients for the same feature in the M interpretable models can be considered stable for the chosen target.

5.3.3 Internal stability vs. External stability

Consider performing LIME to explain a prediction made for a certain target for several times when the predictive model and the underlying training set stay unchanged. Hence LIME follows the same distribution (law) to generate new data points to build the interpretable models. However, due to the random nature of the sampling, LIME could still generate different data points among repeated calls and build different interpretable models,

[10] Similar to the CV value, When calculating the CSI csi^{iu} for a target $\mathbf{x}^i$ at a default rate u, we do not consider the features which only presents once in all M interpretable models since there is no pair of confidence intervals can be compared. Therefore, P^* in Equation (11) will be less than or equal to the number of unique features P in M interpretable models.

thus providing unstable interpretations for the chosen target. The internal stability described here is different from the stability measured by following our experimental framework introduced in Figure 1. In our experiments, we measure the ranking and value stability of LIME under the same class imbalance level using different training sets and therefore different predictive models, which means LIME generates data points and build interpretable models based on different training sets and therefore different distribution (law) to some extent. Therefore, we name the stability measured in our experiments the "external" stability.

It is important to note that Tree SHAP used in our experiments does not include a sampling procedure, hence there is no need to check the internal stability for SHAP. To check the internal stability of LIME, we perform an experiment in which we use the same 200 initially sampled targets and adjust Step 5 of experimental framework introduced in Figure 1 to fit the measurement of internal stability of LIME. We specifically repeat the Step 5 for 30 times, so that we could get 30 LIME interpretable models (Ridge regressions) based on the same conditions (generating the neighborhood from the same training set/distribution). Moreover, other than repeating the process from Step 3 to Step 5 for 100 times, here we only repeat for 10 times. Hence we could get 30×10 LIME interpretable models for each target at each default rate, in which every 30 LIME interpretation models are based on the same conditions. We then calculate the VSI value vsi^{iu} using Equation 10 and the CSI value csi^{iu} using Equation 11 for every 30 LIME interpretable models based on the same conditions. Therefore, at every default rate u, we could obtain 10 VSI values and 10 CSI values for each target $\mathbf{x}^i$ and we calculate the average of them respectively to get the average VSI $vsi^{iu}_{internal}$ and the average CSI $csi^{iu}_{internal}$ for each target $\mathbf{x}$.

In this study, we also measure the external stability of LIME using CSI and VSI. Specifically, we follow our original experimental framework (Figure 1) to calculate the VSI value $vsi^{iu}_{external}$ using Equation 10 and the CSI value $csi^{iu}_{external}$ using Equation 11 for 100 interpretable models (100 sets of feature importance values) based on different conditions (generating the neighborhood from different training set/distribution) for each target $\mathbf{x}^i$ at the same default rate u.

By doing this, we could compare $vsi^{iu}_{internal}$ and $csi^{iu}_{internal}$ with $vsi^{iu}_{external}$ and $csi^{iu}_{external}$ respectively, to gain a greater insight into the effects of class imbalance on the interpretation stability of LIME.

6. Experimental Results

Following the experimental framework described at the beginning of Section 4, for LIME and SHAP respectively, we repeat Step 3 to Step 5 100[11] times to obtain 100 sets of feature importance values, which can be transformed into 100 feature ranking lists, for each target $\mathbf{x}^i$ at every class imbalance level u. We then calculate the SRA value $RankStab^{iu}_d$ of each list depth d introduced in Section 5.1 to measure the feature ranking stability of LIME and SHAP for each target $\mathbf{x}^i$ at every class imbalance level u, and the results are discussed in Section 6.1. Similarly,

[11] We take 100 iterations since we need to make sure the stability indexes converge for more accurate measurements. Based on our experiments, the results of measurements converge after around 60 iterations (See Supplementary Materials (Part E) for more details).

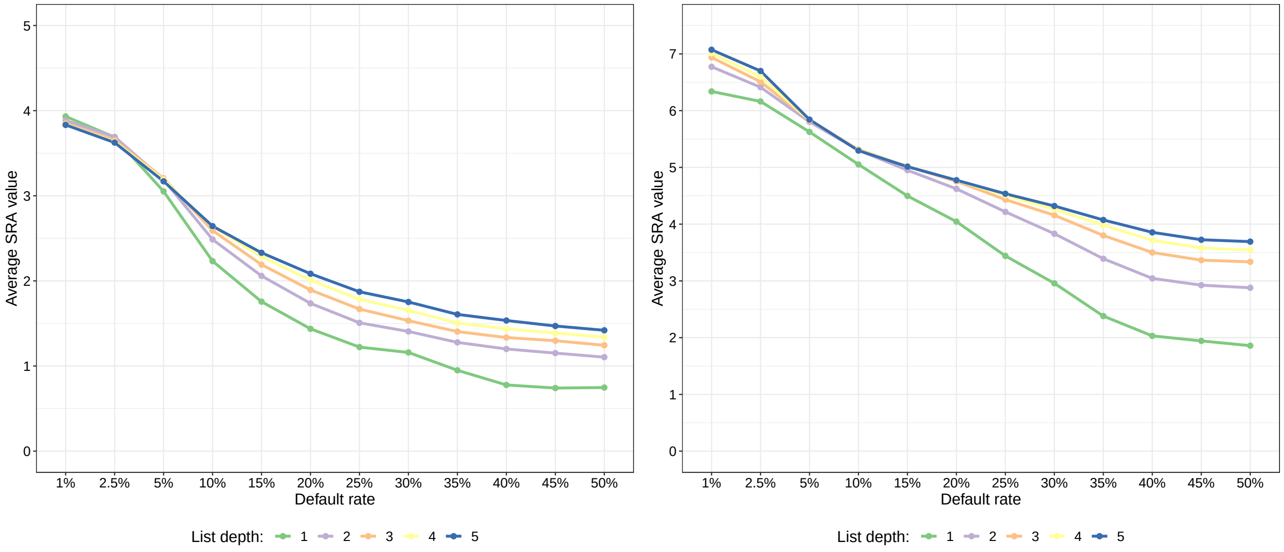
the CV value $ValueStab^{iu}$ introduced in Section 5.2 are calculated to measure the feature importance value stability of LIME and SHAP for each target $\mathbf{x}^i$ at every class imbalance level u, and the results are discussed in Section 6.2. Moreover, for each target $\mathbf{x}^i$, we obtain the internal VSI value $vsi^{iu}_{internal}$ and CSI value $csi^{iu}_{internal}$, as well as the external VSI value $vsi^{iu}_{external}$ and CSI values $csi^{iu}_{external}$ following the description in Section 5.3 to further evaluate the stability of LIME, and the results are discussed in Section 6.3. Note that our results are consistent across all three datasets. In Section 6.1, Section 6.2 and Section 6.3, we describe the results for the European Datawarehouse data. In Section 6.4, we summarise the results of two open-source credit scoring datasets, and the details of the results can be found in the Supplementary Materials (Parts F and G). The prediction results (H-measure) can also be found in Supplementary Materials (Part C). Note that the relative spread of H-measure values (dispersion levels of H-measure values) for fixed targets (regarded as the test set) over different levels of class imbalance is basically stable, which confirms that the prediction performance will not affect the stability of the interpretations generated by LIME and SHAP.

6.1. SRA results

Recall that we repeat the whole experiment framework 5 times and therefore we have stability measurement results for 1000 (5 × 200) targets. To achieve a more general result, at each class imbalance level u, we calculate the average of the SRA value $RankStab^{iu}_{d}$ of all 1000 initially sampled targets for each list depth d. Panel (a) and Panel (b) of Figure 2 shows the average SRA value for LIME and SHAP respectively. Here we use a line chart to show, for a particular list depth d, the trend of the average SRA value as the class imbalance level decreases. Specifically, on the x-axis is the class imbalance level, represented by the default rate, ranging from 1% (extreme imbalanced) to 50% (balanced), and on the y-axis is the average SRA value. Please note that the range of the average SRA value on the y-axis for LIME and SHAP is different in order to make the line chart more visible. Each line represents a list depth and therefore each point corresponding to a default rate on the line is the average SRA value at a certain class imbalance level. Note that for clarity of presentation, here we only show the average SRA value for the list depth from 1 to 5, which represents the ranking stability of the top most important 5 features, and the average SRA values for the complete list depths can be found in Supplementary Materials (Part D), which leads to similar conclusions.

Every line in Figure 2 show a distinct downward trend with slight fluctuations, which means the average SRA value of each list depth continues to decrease as the default rate gradually increases. Recall that the smaller the SRA value, the better the agreement achieves among the ranking lists. The results indicate that the feature ranking stability increases with the decrease of the class imbalance, thereby confirming the class imbalance does have an adverse effect on the interpretive performance of both LIME and SHAP.

Other than comparing the average SRA value based on the variation of the class imbalance level, we could also observe a regular tendency when comparing within each class imbalance level (default rate). With the exception of 1% and 2.5% default rates in the line chart of LIME, in each default rates for LIME and SHAP, the



(a) *LIME*

(b) *SHAP*

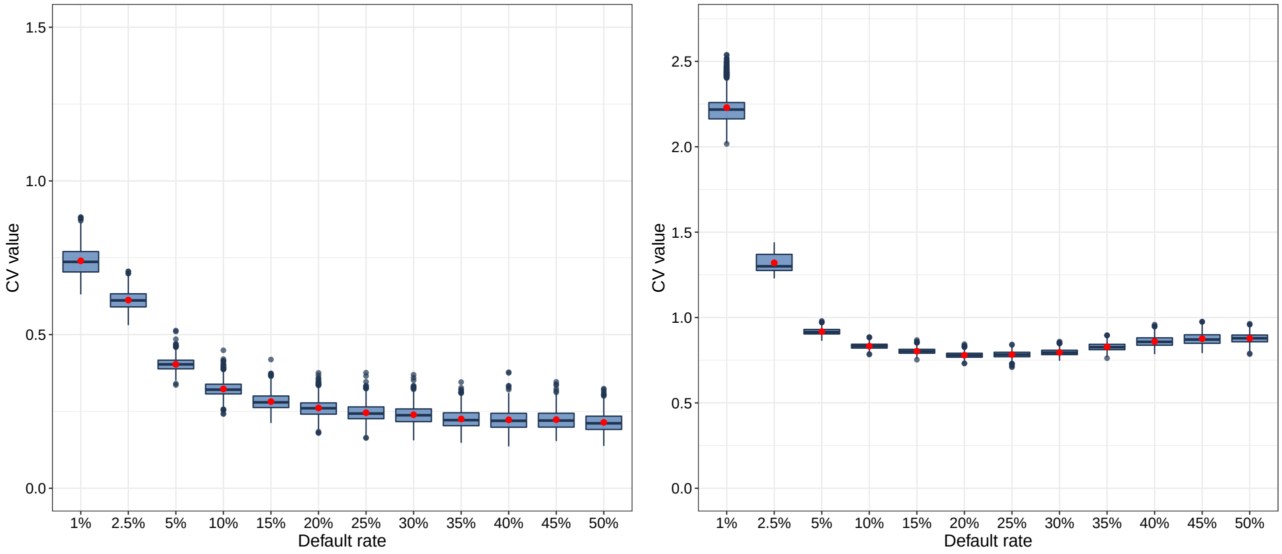
Figure 2: *Line charts of averaged SRA values for LIME in Panel (a) and SHAP in Panel (b)*

average SRA value presents the minimum when the list depth equals to 1, indicating the best agreement, and then increases as the the list depth increases. It indicates that the feature ranking lists generated by LIME and SHAP agree more on higher rankings and can achieve the most stable ranking for the most important feature, but the variability still exists as the average SRA value is not equal to 0. For the average SRA value within 1% and 2.5% default rates in the line chart of LIME, although not distinct, there is an opposite trend that they show higher disagreement (larger average SRA value) in the top of the lists followed by a decrease as the list depth increases. The reason behind this is rather subtle. Looking at the absolute value of the Ridge regression coefficients in LIME, we see that most of them are very close to zero and have almost equal absolute value at these two extreme class imbalance levels (1% and 2.5%). It implies that when features are ranked according to the magnitude of their absolute value of the coefficients, their order becomes more uncertain and more close to a random permutation. Hence the feature ranked in the top of the lists may obtain a larger ranking agreement value $RankStabFeat^{iu}(F_p)$ according to Equation 6, which results in a larger average SRA value.

When comparing the average SRA value between LIME and SHAP, it can be seen that for every default rate, the average SRA value of all list depths for SHAP are larger than those for LIME. This is reasonable since LIME performs a preliminary feature selection step and only use 10 selected features but not all 37 features used by SHAP to generate interpretations, which leads to a much smaller ranking range for LIME, and hence a smaller average SRA value.

6.2. CV results

At each class imbalance level u, we obtain a total of 1000 CV values $ValueStab^{iu}$ for all 1000 initially sampled targets. For better illustration, for LIME and SHAP respectively, we draw a box-and-whisker plot (boxplot) for every set of 1000 CV values at each class imbalance level, as shown in Panel (a) and Panel (b) in Figure 3.



(a) *LIME*

(b) *SHAP*

Figure 3: *Boxplots of CV values for LIME in Panel (a) and SHAP in Panel (b)*

The x-axis represents the class imbalance level (default rate), and the y-axis represents the CV value. The range of the CV value on the y-axis for LIME and SHAP is different in order to make the box plot more visible. Each box in Figure 3 extends from the first quartile (Q1) to the third quartile (Q3) of the set of CV values, with a line at the median (Q2). The whiskers extend no more than $1.5 \times IQR(IQR = Q3 - Q1)$ and end at the farthest data point within this interval from the edges of box, to show the range of the set of CV values. Outliers are plotted as separate blue dots. The red dot on each box shows the mean of the set of CV values.

As demonstrated in Panel (a) of Figure 3, CV values for LIME under 1% and 2.5% default rates are evidently larger compared with the other sets of CV values. While there is a decreasing trend after 5%, there is no significant difference in the sets of CV values when the default rate is greater than 15%. Recall that the larger the CV value, the greater the (average) variability among the feature importance values. It proves that the absolute LIME coefficients generated are much less stable in the case of the extreme class imbalance. The same conclusion can also applies to SHAP based on the results presented in Panel (b) of Figure 3, with the distinction between the CV values at the 1%, 2.5% default rates and others are relatively more obvious.

When comparing the CV values between LIME and SHAP, we can see that the CV values for SHAP at each default rate are larger than those for LIME. One possible reason behind this could be that the CV values of SHAP are calculated based on all 37 features, which may result in more variation among feature importance values especially for those relatively unimportant features.

6.3. VSI and CSI results

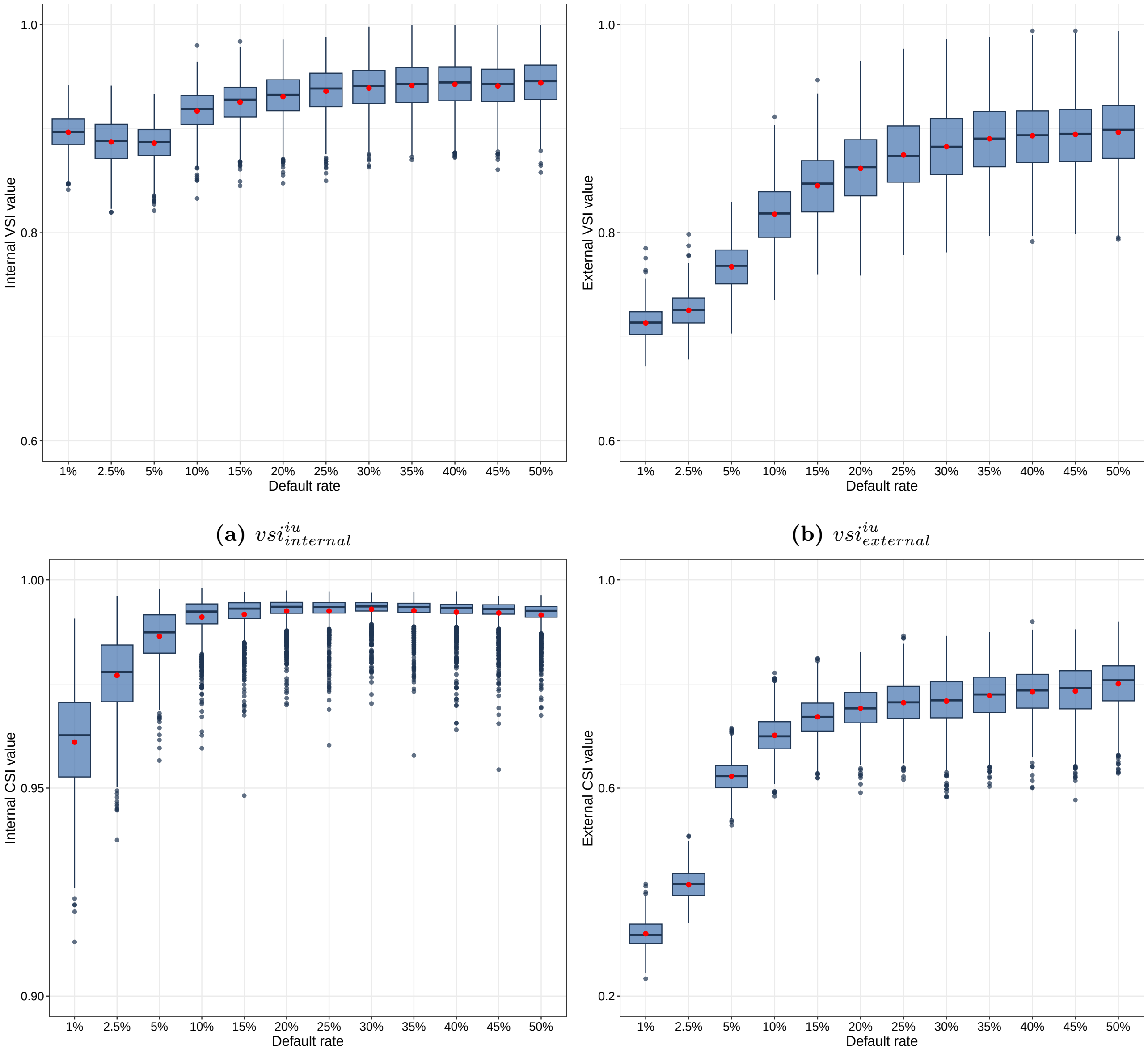
We now focus on the feature selection stability and the feature coefficients stability of LIME, which are measured by VSI and CSI respectively. Figure 4 shows the boxplots of internal VSI values $vsi^{iu}_{internal}$, external VSI values $vsi^{iu}_{external}$, internal CSI values $csi^{iu}_{internal}$ and external CSI values $csi^{iu}_{external}$ in Panel (a), (b), (c) and (d) respectively. For all four plots, on the x-axis is the default rate, which represents the class imbalance level, and on the y-axis is the VSI or CSI value. The range of the internal VSI value in Panel (a) and the external VSI value in Panel (b) on the y-axis is the same, starting from 0.6 to 1. For clarity of presentation, in Panel (c) we zoom in on the internal CSI value with a range from 0.9 to 1 on the y-axis, and we zoom out on the external CSI value in Panel (d) with a range from 0.2 to 1 on the y-axis. Each boxplot in Figure 4 represents a set of 1000 VSI or CSI values at a certain class imbalance level for all 1000 initially sampled targets. The setting of the boxplot is the same as that described for Figure 3 in Section 6.2.

Recall that for the chosen target, the more the VSI value approaches 1, the more the features found in different LIME interpretable models are the same. Similarly, the more the CSI value approaches 1, the more the LIME coefficient values for the same feature in different LIME interpretable models may be considered stable.

As we can see in Panels (a) and (b) of Figure 4, internal and external VSI values show different behaviours. As shown in Panel (a), all the internal VSI values are above 0.8, most 0.9, which means that the selected features in LIME interpretable models generated from the same conditions are almost the same. Even for the extreme class imbalanced cases (1%, 2.5% and 5% default rates), internal VSI values are slightly lower than for the more balanced cases, but with an average still around 0.9. However, we can more clearly see the adverse effect of class imbalance on the stability of LIME interpretations from the boxplots of external VSI values in Panel (b). Specifically, the mean of each set of external VSI values starts from 0.7 at 1% default rate and has a distinct growing tendency with the increase of the default rate, which proves that the similarity of the selected features in LIME interpretable models increases as the class distribution becomes more balanced.

When looking into the internal and external CSI values in Panels (c) and (d) of Figure 4, we can see both of them share the effect of class imbalance on stability, although for the external CSI values such an effect is more extreme. Similar to the internal VSI values, all the internal CSI values in Panels (c) are above 0.9, but the range of the set of internal CSI values for the extreme class imbalanced cases (1%, 2.5% and 5% default rates) are obviously larger than for the more balanced cases. The mean of the set of internal CSI values also increases from 1% default rate and remains basically unchanged after the default rate reaches 10%. It confirms that the coefficients are less stable in the case of an extreme class imbalance (1%, 2.5% and 5% default rates). As shown in Panel (d), the mean of each set of external CSI values starts from only around 0.3 at 1% default rate, then goes up with the default rate and shows a dramatic increase between 2.5% default rate and 5% default rate. It confirms that LIME generates more stable coefficients of the same feature based on more balanced datasets, and the concordance of coefficients to the same feature tends to be seriously affected at the extreme class imbalance level (1% and 2.5% default rates).

When comparing the internal stability with the external stability of LIME, for both VSI and CSI values, the internal ones show higher values than the external ones at each class imbalance level. It indicates that although there are still inconsistencies, repeated calls of LIME based on the same conditions tend to yield more stable interpretation results than those based on different conditions. This is reasonable since the neighbourhoods



(a) $vsi^{iu}_{internal}$ **(b)** $vsi^{iu}_{external}$

(c) $csi^{iu}_{internal}$ **(d)** $csi^{iu}_{external}$

Figure 4: *Boxplots of internal VSI values in Panel (a), external VSI values in Panel (b), internal CSI values in Panel (c) and external CSI values in Panel (d) for LIME*

sampled from different training sets on which the interpretable models are built can increase the instability of the interpretations for a certain target. Moreover, the external stability of LIME is more affected by class imbalance than its internal stability. One possible reason behind this is that when we repeat the sampling procedure, due to the lack of information for defaulters in the (extreme) imbalanced training sets, we can only build the interpretable models for a certain target based on incomplete and distinct information and therefore increase the variability among the interpretation results.

6.4. Robustness check using two additional datasets

We apply the experiments above described also to two additional datasets to check the robustness of the stability measurement results. We use the South German Credit Dataset, which contains 1,000 observations and 21 predictors with personal credit risk information; and the Taiwan Credit Card Dataset, which contains 30,000 observations and 24 predictors with customers' credit card transaction information. Both datasets are open-sourced, differ with respect to the number of observations and predictors, and are used by many papers in the credit scoring literature (e.g., Gunnarsson et al., 2021; Lessmann et al., 2015), which complement the European Datawarehouse mortgage data in terms of feature and sample size variety.

The details of the results are presented in Supplementary Materials (Parts F and G). Overall, the SRA, CV, CSI and VSI results using the two open-source datasets are consistent with those using European Datawarehouse data. This confirms the robustness of our results that the stability of interpretability results could be adversely affected by class imbalance. For feature importance ranking stability, the only difference is that for the open source datasets, the SRA values when the list depth is equal to 1 significantly differ from the SRA values at other list depths. This may be because the features ranked first are relatively stable, but there is greater randomness in the subsequent rankings. The feature importance values generated by LIME and SHAP are also less stable at extreme class imbalance based on open source datasets, while the difference between CV values at 1% and 2.5% default rates and CV values at larger default rates are less obvious for SHAP compared to that on the European Datawarehouse data. Similarly, the results of VSI and CSI for LIME on the open source data also agree with those on the European Datawarehouse data.

7. Conclusions and future research

In this paper, we consider two popular model-agnostic interpretation methods — LIME and SHAP, and study their interpretative performance over various class imbalance levels. We achieve this by proposing a controlled sampling process to produce a series of datasets with different default rates but the same sample size. We use residential mortgage data provided by the European Datawarehouse and two more open source credit scoring datasets to verify the robustness of our results. XGBoost and Random Forest are selected as the "black-box" machine learning models to generate prediction since they are widely used in the credit scoring literature for their excellent performance (Barbaglia et al., 2021; Gunnarsson et al., 2021; Xia et al., 2017). We focus on the feature ranking lists and the corresponding feature importance values generated by LIME and SHAP. Sequential Rank Agreement (SRA) and Coefficient of Variation (CV) are then applied to measure the feature ranking stability and the feature importance value stability respectively. We further evaluate the "internal" stability and the "external" stability of LIME by using Variables Stability Index (VSI) and Coefficients Stability Index (CSI).

The results of our experiments show that the class imbalance does have an adverse effect on the interpretive performance of both LIME and SHAP. Firstly, the feature importance rankings generated by LIME and SHAP are more stable as the class imbalance level decreases (from 1% default rate to 50% default rate). Secondly, there is greater variability among the absolute SHAP values corresponding to the same feature and also the absolute LIME coefficients corresponding to the same feature in the case of an extreme class imbalance (1%, 2.5% and 5% default rates). Finally, in LIME, the consistency of the selected features, as well as the similarity of the coefficients for the same feature, does increase as the class distribution becomes more balanced (from 1% default rate to 50% default rate). Even when we measure the "internal" stability of LIME, for which we perform repeated calls of LIME based on the same predictive model and the same training set, it appears that LIME does generate less stable interpretations at extreme class imbalance levels (1%, 2.5% and 5% default rates).

To the best of our knowledge, this is the first study that measures the stability of LIME and SHAP in terms of class imbalance, which fills a key research gap in the literature. Although we focus on credit scoring in this paper, the proposed experimental framework can be used in other operational research applications that also suffer from the class imbalance problem, such as the medical industry. Our research has important implications for financial institutions and other adopters who have already or are willing to use the model-agnostic interpretation methods to interpret the "black-box" machine learning models, to measure the stability of the selected interpretation methods. Although the interpretation methods are flexible and easy to adopt, practitioners should be very careful when applying them to imbalanced datasets and making any decisions based on them, as the class imbalance can obviously exacerbate the instability of interpretations, especially at extreme class imbalance levels (under 5% for the proportion of the rare events).

As mentioned in the Introduction, various imbalanced learning techniques have been introduced to improve classification performance (e.g. Calabrese and Osmetti, 2015; Chawla et al., 2011; Krawczyk, 2016). Similarly, the potential effects of these imbalanced learning techniques on the performance of interpretation methods could be investigated in future research. It is worth noting that although the resampling methods could be used to tackle imbalanced data challenges, they could cause problems such as overfitting (over-sampling methods) or information loss (under-sampling methods) (Fernández et al., 2018; Haixiang et al., 2017; Kaur et al., 2019; Li et al., 2019). More importantly, resampling methods could increase randomness and add noise for the original input data, which is not desirable in financial applications and limits their prevalent in the corporate landscape (Gunnarsson et al., 2021; Lessmann et al., 2015; Sanz et al., 2015). Therefore, it would be preferable to apply other imbalanced learning techniques to credit scoring, such as cost-sensitive or algorithm-based methods (e.g., Paleologo et al., 2010; Zhang et al., 2014) to help in generating unbiased interpretation results. More fundamentally, it would be valuable to investigate the effects of class imbalance on the stability of interpretation methods theoretically. For example, we could start using Logistic Regression as the predictive model and establishing the theoretical results of interpretations generated by SHAP or LIME in the context of class imbalance. This could provide further guidance for analysing the stability of interpretation methods when using more complicated "black-box" machine learning models.

Besides the directions of future research mentioned above, another interesting extension would be to explore how sample size changes may affect the stability of the interpretation methods while the class imbalance remains the same. Finally, the novel interpretation method could be investigated to consider the class imbalance issue and the potential stability measurements could be embedded to provide insights of the interpretation stability.

References

Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the Robustness of Interpretability Methods. *2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018).*

Andreeva, G., Calabrese, R., & Osmetti, S. A. (2016). A comparative analysis of the UK and Italian small businesses using Generalised Extreme Value models. *European Journal of Operational Research*, *249*(2), 506–516.

Apley, D. W., & Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *82*(4), 1059–1086.

Ariza-Garzon, M. J., Arroyo, J., Caparrini, A., & Segovia-Vargas, M. J. (2020). Explainability of a Machine Learning Granting Scoring Model in Peer-to-Peer Lending. *IEEE Access*, *8*, 64873–64890.

Bank of England. (2019). *Machine learning in UK financial services* (tech. rep.).

Barbaglia, L., Manzan, S., & Tosetti, E. (2021). Forecasting Loan Default in Europe with Machine Learning. *Journal of Financial Econometrics.*

Bracke, P., Datta, A., Jung, C., & Sen, S. (2019). Machine Learning Explainability in Finance: An Application to Default Risk Analysis. *SSRN Electronic Journal*, (816).

Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, *39*(3), 3446–3453.

Bücker, M., Szepannek, G., Gosiewska, A., & Biecek, P. (2021). Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society.*

Bueff, A. C., Cytryński, M., Calabrese, R., Jones, M., Roberts, J., Moore, J., & Brown, I. (2022). Machine learning interpretability for a stress scenario generation in credit scoring based on counterfactuals. *Expert Systems with Applications*, *202*, 117271.

Bussmann, N., Giudici, P., Marinelli, D., & Papenbrock, J. (2021). Explainable Machine Learning in Credit Risk Management. *Computational Economics*, *57*(1), 203–216.

Calabrese, R., Marra, G., & Osmetti, S. A. (2016). Bankruptcy prediction of small and medium enterprises using a flexible binary generalized extreme value model. *Journal of the Operational Research Society*, *67*(4), 604–615.

Calabrese, R., & Osmetti, S. A. (2015). Improving forecast of binary rare events data: A gam-based approach. *Journal of Forecasting*, *34*(3), 230–239.

Chang, Y. C., Chang, K. H., & Wu, G. J. (2018). Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. *Applied Soft Computing Journal*, *73*, 914–920.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2011). SMOTE: Synthetic Minority Over-sampling Technique. *Journal Of Artificial Intelligence Research*, *16*, 321–357.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Consumer Financial Protection Bureau. (2022). *Consumer Financial Protection Circular 2022-03: Adverse action notification requirements in connection with credit decisions based on complex algorithms* (tech. rep.).

Crone, S. F., & Finlay, S. (2012). Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting*, *28*(1), 224–238.

Dumitrescu, E., Hué, S., Hurlin, C., & Tokpavi, S. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*, *297*(3), 1178–1192.

Ekstrøm, C. T., Gerds, T. A., & Jensen, A. K. (2019). Sequential rank agreement methods for comparison of ranked lists. *Biostatistics*, *20*(4), 582–598.

European Banking Authority. (2020). *EBA report on big data and advanced analytics* (tech. rep.).

European Banking Authority. (2021). *EBA discussion paper on machine learning for IRB models* (tech. rep.).

European Commission. (2021). Proposal for a regulation of the European Parliament and the Councils laying down harmonised rules on Artificial Intelligence.

Fernández, A., García, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research*, *61*, 863–905.

Fitzpatrick, T., & Mues, C. (2016). An empirical comparison of classification algorithms for mortgage default prediction: Evidence from a distressed mortgage market. *European Journal of Operational Research*, *249*(2), 427–439.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232.

Gramegna, A., & Giudici, P. (2021). SHAP and LIME: An Evaluation of Discriminative Power in Credit Risk. *Frontiers in Artificial Intelligence*, *4*, 140.

Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., & Giannotti, F. (2018). Local Rule-Based Explanations of Black Box Decision Systems.

Gunnarsson, B. R., vanden Broucke, S., Baesens, B., Óskarsdóttir, M., & Lemahieu, W. (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research*, *295*(1), 292–305.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220–239.

Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning*, *77*(1), 103–123.

Hand, D. J. (2010). Evaluating diagnostic tests: The area under the ROC curve and the balance of errors. *Statistics in medicine*, *29*(14), 1502–1510.

Harald, S., Bart, B., & Roesch, D. (2016). *Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS (Wiley and SAS Business Series)*. Wiley.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284.

ICO and The Alan Turing Institute. (2020). Explaining decisions made with AI.

Janzing, D., Minorics, L., & Blöbaum, P. (2019). Feature relevance quantification in explainable AI: A causal problem.

Kaur, H., Pannu, H. S., & Malhi, A. K. (2019). A Systematic Review on Imbalanced Data Challenges in Machine Learning. *ACM Computing Surveys (CSUR)*, *52*(4).

King, G., & Zeng, L. (2001a). Explaining rare events in international relations. *International Organization*, *55*(3), 693–715.

King, G., & Zeng, L. (2001b). Logistic regression in rare events data. *Political Analysis*, *9*(2), 137–163.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, *5*(4), 221–232.

Laurent Dupont, Fliche, O., & Yang, S. (2020). *Governance of Artificial Intelligence in Finance* (tech. rep.). ACPR.

Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, *247*(1), 124–136.

Li, X., Zhou, Y., Dvornek, N. C., Gu, Y., Ventola, P., & Duncan, J. S. (2020). Efficient shapley explanation for features importance estimation under uncertainty. *Medical Image Computing and Computer Assisted Intervention - MICCAI 2020 Lecture Notes in Computer Science*, 792–801.

Li, Y., Bellotti, T., & Adams, N. (2019). Issues using logistic regression with class imbalance, with a case study from credit risk modelling. *Foundations of Data Science*, *1*(4), 389–417.

Liu, W., Fan, H., & Xia, M. (2022). Credit scoring based on tree-enhanced gradient boosting decision trees. *Expert Systems with Applications*, *189*.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, *2*(1), 56–67.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4768–4777.

Marqués, A. I., García, V., & Sánchez, J. S. (2013). On the suitability of resampling techniques for the class imbalance problem in credit scoring. *Journal of the Operational Research Society*, *64*(7), 1060–1070.

Martens, D., Baesens, B., Van Gestel, T., & Vanthienen, J. (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research*, *183*(3), 1466–1476.

Molnar, C. (2021). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.*

Moscato, V., Picariello, A., & Sperlí, G. (2021). A benchmark of machine learning approaches for credit score prediction. *Expert Systems with Applications*, *165*(May 2020), 113986.

Namvar, A., Siami, M., Rabhi, F., & Naderpour, M. (2018). Credit risk prediction in an imbalanced social lending environment. *International Journal of Computational Intelligence Systems*, *11*(1), 925–935.

Óskarsdóttir, M., & Bravo, C. (2021). Multilayer network analysis for improved credit risk prediction. *Omega*, *105*.

Owen, A. B. (2007). Infinitely imbalanced logistic regression. *Journal of Machine Learning Research*, *8*, 761–773.

Paleologo, G., Elisseeff, A., & Antonini, G. (2010). Subagging for credit scoring models. *European Journal of Operational Research*, *201*(2), 490–499.

Patil, A., Framewala, A., & Kazi, F. (2020). Explainability of SMOTE based oversampling for imbalanced dataset problems. *Proceedings - 3rd International Conference on Information and Computer Technologies, ICICT 2020*, 41–45.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 1135–1144.

Sanz, J. A., Bernardo, D., Herrera, F., Bustince, H., & Hagras, H. (2015). A Compact Evolutionary Interval-Valued Fuzzy Rule-Based Classification System for the Modeling and Prediction of Real-World Financial Applications with Imbalanced Data. *IEEE Transactions on Fuzzy Systems*, *23*(4), 973–990.

Shaikhina, T., Bhatt, U., Zhang QuantumBlack Konstantinos Georgatzis QuantumBlack Alice Xiang, R., & Weller, A. (2021). Effects of Uncertainty on the Quality of Feature Importance Explanations. *AAAI Workshop on Explainable Agency in Artificial Intelligence.*

Shankaranarayana, S. M., & Runje, D. (2019). ALIME: Autoencoder Based Approach for Local Interpretability. *Intelligent Data Engineering and Automated Learning – IDEAL 2019*, *11871 LNCS*, 454–463.

Shapley, L. S. (1953). A Value for n-Person Games. In *Contributions to the theory of games* (pp. 307–318). Princeton University Press.

Singh, R., Dourish, P., Howe, P., Miller, T., Sonenberg, L., Velloso, E., & Vetere, F. (2021). Directive Explanations for Actionable Explainability in Machine Learning Applications.

Sirignano, J., Sadhwani, A., & Giesecke, K. (2018). Deep Learning for Mortgage Risk. *SSRN Electronic Journal*, 1–83.

Slack, D., Hilgard, S., Singh, S., & Lakkaraju, H. (2021). Reliable Post hoc Explanations: Modeling Uncertainty in Explainability. *35th Conference on Neural Information Processing Systems (NeurIPS 2021).*

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, *41*(3), 647–665.

Sundararajan, M., & Najmi, A. (2020). The many shapley values for model explanation. *37th International Conference on Machine Learning (ICML 2020)*, 9210–9220.

Szwabe, A., & Misiorek, P. (2018). Decision Trees as Interpretable Bank Credit Scoring Models. In S. Kozielski, D. Mrozek, P. Kasprowski, B. Małysiak-Mrozek, & D. Kostrzewa (Eds.), *Beyond databases, architectures and structures. facing the challenges of data proliferation and growing variety* (pp. 207–219). Springer, Cham.

Thomas, L., Crook, J., & Edelman, D. (2017). *Credit Scoring and Its Applications.* SIAM.

Visani, G., Bagli, E., Chesani, F., Poluzzi, A., & Capuzzo, D. (2021). Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models. *Journal of the Operational Research Society.*

Voigt, P., & von dem Bussche, A. (2017). *The EU General Data Protection Regulation (GDPR): A Practical Guide* (1st ed.). Springer, Cham.

Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *SSRN Electronic Journal.*

Xia, Y., Liu, C., Li, Y. Y., & Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, *78*, 225–241.

Zafar, M. R., & Khan, N. M. (2019). DLIME: A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems. *Proceedings of Anchorage '19: ACM SIGKDD Workshop on Explainable AI/ML (XAI) for Accountability, Fairness, and Transparency.*

Zhang, Z., Gao, G., & Shi, Y. (2014). Credit risk evaluation using multi-criteria optimization classifier with kernel, fuzzification and penalty factors. *European Journal of Operational Research*, *237*(1), 335–348.

Zhao, X., Huang, W., Huang, X., Robu, V., & Flynn, D. (2021). BayLIME: Bayesian Local Interpretable Model-Agnostic Explanations. *37th Conference on Uncertainty in Artificial Intelligence 2021*, 887–896.