Imbalanced Credit Risk Prediction in Ensemble Learning Classifiers: A Comparative Analysis of SMOTE, ADASYN, SMOTETomek, and Cluster Centroids

Zixue Zhao[1] and Vesarach Aumeboonsuke[2]

National Institute of Development Administration, Thailand

E-mail: nana0000@foxmail.com[1], vesarach.a@nida.ac.th[2]

Received July 6, 2023**; Revised** August 28, 2023**; Accepted** September 15, 2023

Abstract

The improvement of financial institutions' ability to predict customers' credit risk has benefited from the continuous updating of machine learning algorithms. The ensemble algorithms represented by Random Forest and XGBoost carry forward the advantages of decision tree structures and perform well on datasets with complex characteristics, such as credit data. Research methodology is conducted via sampling design; we employ the German Credit dataset, a benchmark dataset comprising 1000 samples and 18 features, for empirical analysis. Measurement Design: various performance metrics such as accuracy, precision, recall, and F1-score are used to assess the efficacy of the balancing techniques. Analysis Design: A comparative analysis is conducted to evaluate the strengths and weaknesses of these balancing techniques in different ensemble learning classifiers. However, in real credit datasets, the number of defaulted samples is usually only a small percentage. Class imbalance greatly weakens the predictive performance of ensemble models. Therefore, this paper employed four different techniques for handling class imbalance problems, namely SMOTE, ADASYN, SMOTETomek, and ClusterCentroid. A comparison of different tree-based models in datasets is demonstrated where balancing techniques are applied. The conclusions show that all models perform much better on the datasets with balancing techniques than without balancing. Balancing the data in advance does improve the predictive ability of the models, and the over-sampling and integrated sampling methods outperform the under-sampling techniques on small and medium-sized datasets.

Keywords: credit risk; imbalance data; ensemble learning; tree–based model; balancing techniques

Introduction

Stigliz and Weiss (1981) laid the groundwork in the area of commercial lending by proposing a model of adverse selection based on the theory of information asymmetry. They argued that the size of commercial loans is often not at the optimal level, leading to an imbalance between the supply of loans from financial institutions and the demand for commercial loans (Dobbie & Skiba, 2013). This phenomenon is not only prevalent in commercial lending but is also commonly observed in the field of personal consumer loans. Adverse selection and moral hazard, arising from information asymmetry, can produce negative effects on the consumer loan market (Stigliz and Weiss, 1981).

In financial activities, the information asymmetry between borrower and lender exacerbates the rise of credit risk. While big data mining technology has alleviated some aspects of information asymmetry, it introduces new challenges, particularly when using ensemble learning algorithms for credit risk prediction. These algorithms, although powerful, often struggle with imbalanced datasets, leading to inaccurate risk assessments and potential financial instability (Auronen, 2003).

The field of credit risk prediction has a rich historical background and has witnessed extensive theoretical advancements, particularly in the realm of credit scoring models. The integration of artificial intelligence and credit risk management, meanwhile, remains a nascent area of study (Witten & Frank, 2002). Machine learning is a significant subfield within the realm of artificial intelligence, characterized by its ability to convert disorganized input into meaningful knowledge. In the context of this transformation process, if our sole emphasis is on indiscriminately imparting data to the learner without making tailored alterations for certain data sets to enhance their efficacy, we will be unable to sufficiently obtain the requisite information (Lessmann et al., 2015). Furthermore, in the context of big data, it is not feasible to apply a single classifier to heterogeneous datasets. Integrated algorithms have been shown to enhance the accuracy of extracting genuine information from data. The assessment and juxtaposition of ensemble approaches hold significant value as points of reference within the field of credit risk prediction.

The utilization of data–balancing strategies is currently lacking in the field. According to Haixiang (2016), whereas boosting for ensemble learning involves data sampling without

replacement, the absence of upfront processing can still result in significant model bias. In a thorough literature analysis conducted by Dastile (2020), a total of 74 relevant studies were examined. The findings of this evaluation revealed that a mere 18% of the comparisons made between boosting family models and effective methods for addressing unbalanced data, primarily undersampling techniques, yielded positive results (Dastile, 2020). Furthermore, the level of depth in the feature engineering is insufficient. The absence of preprocessing techniques for categorical features and the absence of a viable method for addressing the sparsity issue resulting from unique thermal coding are evident.

To address these challenges, this study introduces and evaluates four advanced balancing techniques—SMOTE (Synthetic Minority Over–sampling Technique), ADASYN (Adaptive Synthetic Sampling), SMOTETomek, and ClusterCentroids—in conjunction with ensemble learning classifiers (Baesens et al., 2003; Chawla et al., 2002; He et al., 2008). These techniques aim to correct the class imbalance by either oversampling the minority class or undersampling the majority class. Our empirical analysis demonstrates that incorporating these balancing techniques significantly improves the predictive performance of ensemble models, particularly for the minority class, thereby providing a more reliable basis for credit risk assessment.

By focusing on these advanced balancing techniques, this study not only contributes to the existing literature on credit risk prediction but also offers practical insights for financial institutions aiming to improve their risk assessment models.

Several studies have explored the use of machine learning algorithms in credit risk prediction (Wang et al., 2021; Alessi et al., 2018). However, these studies often overlook the importance of data balancing techniques and feature engineering, particularly the preprocessing of categorical features (Zhou, 2021). In light of these gaps, this article aims to introduce and evaluate advanced balancing techniques such as SMOTE, ADASYN, SMOTETomek, and ClusterCentroids, and provide a detailed account of feature engineering methods employed. Our empirical analysis aims to serve as a comprehensive point of reference within the field of credit risk prediction.

Literature Review

The field of credit risk prediction has a rich historical background and has witnessed extensive theoretical advancements, particularly in the realm of credit scoring models (Thomas, 2000; West, 2000). The integration of artificial intelligence and machine learning into credit risk management is a nascent but rapidly evolving area of study (Witten et al., 2016; Brynjolfsson & McAfee, 2014).

Identifying the Gap

While ensemble methods have shown promise in enhancing the accuracy of credit risk prediction (Zhang & Ma, 2012; Lessmann et al., 2015), there exists a significant gap in the literature concerning the effective handling of imbalanced datasets. A systematic review by Dastile (2020) found that only 18% of studies on boosting family models employed effective means for dealing with unbalanced data, most of which utilized undersampling techniques. This lack of upfront processing may lead to high model bias and ultimately unreliable risk assessments (Haixiang et al., 2016).

Moreover, existing studies often overlook the importance of feature engineering, particularly the preprocessing of categorical features and addressing the sparsity problem arising from unique thermal coding (reference for feature engineering gap).

In this paper, we aim to fill these gaps by introducing and evaluating advanced balancing techniques such as SMOTE, ADASYN, SMOTETomek, and ClusterCentroids, and by providing a detailed account of feature engineering methods employed. Our empirical analysis aims to serve as a comprehensive point of reference within the field of credit risk prediction.

Decision Tree and Tree–based ensemble algorithms

Decision Tree (DT)

DT is a kind of supervised learning. The classical ID3 algorithm runs based on information entropy (the scale of random uncertainty of an event), so that the tree acts as a decision center and analyzes the information entropy value of each input feature, with the higher entropy value as the parent node, and splits downwards until the best decision is reached when the entropy value is the lowest. Suppose the distribution of feature A is

$P(A = 1) = p, P(A = 0) = 1 - p\ (0 \leq p \leq 1)$ then

$nt(p) = -plog_2 p - (1 - p)\log_2(1 - p)$. When $p = 1\ or\ p = 0$, the entropy value equals to zero, which means a certain event.

As an improved algorithm of ID3, C4.5 introduces information gain as a basis for decision making. Essentially, it is a conditional probability entropy. Suppose the contribution of feature B to some set D in the dataset is gain (B, D), and the entropy of set D is Ent(D), then

$$gain(D,B) = E(D) - E(D|B)$$

The ratio of information gain to the entropy of feature A in set D is the information gain ratio. However, the limitation of information gain is that if a feature takes more values, then the information gain will be larger. Therefore, Classification and Regression Tree (CART) introduces the Gini index for feature selection in regression or classification problems. Similar to the information entropy, the smaller the Gini index, the smaller the uncertainty of the sample. If the probability of a sample being classified as class n in a data set is p_n, then the Gini index of the probability distribution is $Gini(p) = \sum_{n=1}^{N} p_n(1 - p_n)$. In order to facilitate the calculation, CART assumes that the decision tree is branched in the form of a binary tree. Each feature is continuously divided into "yes" and "no". Finally, the best branching point is determined by the Gini index. This CART form of tree is one of the most versatile decision tree generation algorithms to date. In an article with a high impact factor, the authors compared the prediction error rates of CART Decision Tree, Neural Network, and k-NN on banking data (Galindo et al., 2000). Recent studies have also proposed methods to solve the non-linear problem by combining Decision trees and lasso Logistic Regression. It not only improves the interpretability of the framework of a decision tree but also has better performance than a Random Forest.

Bagging——Random Forest (RF)

Theoretically, a model with only one decision tree does not perform well in solving classification problems with high-dimensional features (Hastie et al., 2009; Kotsiantis, 2013). To solve more complex classification problems, we can integrate many trees into a forest. This is an ensemble learning technique called Random Forest (RF). It is a parallel integration model in which every two trees are independent of each other, also called Bagging, and finally the final prediction is determined using mean reversion.

Compared with traditional credit scorecard models for default prediction, Random Forest performs more accurately and robustly under all evaluation metrics. It is popular among large banks in various countries. These banks or credit institutions are committed to collecting multidimensional big data, then using machine learning models such as Random Forests, Bayesian networks, and clustering to develop personalized installment products for customers (Kumar et al., 2022); some evaluate the credit background of customers, by applying models like Logistic Regression and Gradient Boosted Decision trees (GBDT) to comprehensively evaluate each dimension of user information and predict their ability to perform. Random Forests have also outperformed logistic regression in building financial early warning systems (EWSs), especially the decision mechanism based on expert voting (Wang et al., 2021). This has been validated with stock exchange data from the United States, Mexico, and other countries (Alessi et al., 2018).

Boosting family

In the previous section, we mentioned parallel ensemble learning bagging, and now we review another serial ensemble learning technique: boosting. Boosting is a method to reach the optimal choice, corresponding to gradient descent. The classifier in Fig. 2.5 is a homogeneous classifier. The most successful use of which is the gradient–boosting decision tree (GBDT) integrated learner with CART as the weak classifier. Several base classifiers can become strong classifiers after iterative combination.

In order to understand boosting, it is important to mention the concept of cost function. Suppose a computer program, after a lot of learning, fits a function to distinguish between good and bad customers, then this function leads to two results: the machine is able to predict with 100% accuracy, or not with 100% accuracy. It is clear that the former is almost impossible to achieve. There may be an error between the prediction result of the program and the real result, and what we want is to make this error as small as possible. Suppose the true result is $y(x)$, and the program predicts the result as $h(x)$, (while x is feature of hypothesized function $h(x)$), then there is a cost function measuring the gap between real results and hypothesized results,

$$cost\ function = \big(h(x) - y(x)\big)^2$$

When x is consisted by all the features of one client and it is already constant, we assume that totally number of x is m, and $x^{(i)}$ is the i^{th} of x. $h(x)$ is determined by the parameters θ of x. θ_j is the j^{th} of θ. So $h(x)$ could be rewritten as $h_\theta(x)$. In order to minimize the cost function, we use $J(\theta)$ to represent cost function, then the basic purpose in our program could be illustrated as follow:

$\min_{\theta_j} J(\theta)$:

$$J(\theta_j) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Optimal θ could be found by different learning algorithms. From calculus, we know that usually this minimization problem can always be solved using letting the partial derivative equal 0. In fact, that is how the programmers did handle it – they came up with the most primitive method called gradient descent, by slowly and gradually decreasing θ to eventually find θ that minimizes the cost function.

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$$

where θ is the first-order Taylor expansion of $\theta - 1$ and $\theta_j = \theta_{j-1} + \Delta\theta_j$. On this basis, gradient boosting borrows the idea of gradient descent and replaces the parameter space with the function space, then a similar iteration can be obtained as follows: $f_t(x) = f_{t-1}(x) + \Delta f_t(x)$, when $\Delta f_t(x) = -\alpha_t[\partial/\partial F(x)]_{F(x)=F_{j-1}(x)}$, and $F(x) = \sum_{t=0}^{T} f_t(x)$.

In short, both gradient descent and gradient boosting are methods to find the parameter or function that gives the lowest cost function. Gradient boosting algorithms combined with the Decision Tree mechanism form a classical ensemble learning classifier: GBDT. Consequently, emerging gradient boosting algorithms such as XGBoost (XGB), LightGBM (LGBM), and CatBoost have been derived. They all allow the use of decision trees as a framework (Chen & Guestrin, 2016), and the differences can be summarized in Table 1.

Table 1 Comparison of XGBoost,LightGBM and CatBoost

	XGBoost	LightGBM	CatBoost
Cost Function	$L(\emptyset) = \sum_i l\left(y_i, \hat{y}_i\right) + \sum_k \Omega(f_k)$, $\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$, $\Omega(f) = \Upsilon T + \frac{1}{2}\lambda\|w\|^2$ (Cost function could be customized by CatBoost)		
Decision Tree Algorithm	Greedy algorithm Approximation algorithm	Gradient based unilateral Sampling (GOSS) Histogram algorithm	Oblivious tree
Growth strategy	Level-wise	Leaf-wise	Binary tree
Category feature support	Not support	Support Gradient encoding	Support One-hot encoding
Characteristic	Accuracy improved by Taylor second order expansion. Leaf division rate fastened by Sparsity-aware algorithm.	Parallel computing. Voting-based strategy.	Good at dealing with Categorical features. Multi-GPU work is supported.

The Boosting family is considered to have better differentiation and resistance to overfitting when dealing with medium-sized datasets after a long period of training and testing, compared to a single classifier and parallel Bagging algorithm. Based on GBDT, Chen proposed the Extreme Gradient Boosting algorithm (XGB) in 2016, which expands the cost function and regularization term through the second-order Taylor expansion to improve computational accuracy. XGB is widely employed in data feature mining. Since then, Microsoft has also released the LightGBM algorithm based on histogram decision trees in January 2017, which uses Gradient-One-Side Sampling (GOSS) to greatly reduce the time and overhead of computing information gain and limit the growth depth (leaf-wise) of "leaves" with lower gain to achieve efficiency.

In April 2017, a new boosting family algorithm, CatBoost, was also proposed by Yandex from Russia (Dorogush, 2018). It is an algorithm in the framework of GBDT based on symmetric decision trees supporting categorical variables and has outstanding performance in dealing with categorical features. Also, CatBoost contains an ordered boosting algorithm that enables it to alleviate the problem of prediction bias during iterations, among others. The author argues that

CatBoost outperforms the previous two models in dealing with binary classification problems and reduces the requirement for data preprocessing, which is in line with Prokhorenkova et al. (2018).

Among the existing studies, the novel XGBoost, LightGBM, and CatBoost have been widely used by researchers in recent years in the field of internet credit risk control due to their excellent model validation performance (Prokhorenkova et al., 2018). Researchers started to widely apply XGBoost to credit scoring within a few years after its rise, and good results have been achieved with Bayesian hyperparametric optimization for tuning parameters and improved model interpretation. Ma (2018) also pioneered the application of LightGBM to the prediction of credit fraud risk within a year after its release. They used it to propose a risk management strategy for P2P platforms. The article argues that the predictive power of LightGBM, although only 0.1% better than the XGBoost–based model, is a huge improvement when scaled to platforms and investors. Subsequently, two years after the release of the Catboost algorithm, Dauod (2019) compared the prediction accuracy of the three algorithms, XGBoost, LightGBM, and Catboost, using a larger credit dataset (about 350,000 samples). The results indicated that the LightGBM–based model was faster and more accurate than the other two in large data sets with multiple features. Gamini (2021), however, found in a recent study that after using three boosting algorithms to predict credit card fraud, Catboost performed best in the validation of the confusion matrix with higher precision and recall than the other two models. Furthermore, they concluded that the three boosting algorithms are not inherently better or worse, despite CatBoost's slightly higher prediction accuracy in 5–fold cross–validation.

Previous studies have suggested that XGBoost, LightGBM, and CatBoost each have their own strengths in credit default prediction, but the latter two are superior to the former in terms of computational speed and processing of category–based features (Chen & Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018). XGBoost is suitable for relatively small datasets due to its slower learning speed and higher explanatory power (Daoud, 2019); whereas LightGBM and CatBoost are dealing with large datasets and have their own advantages in terms of learning speed and prediction accuracy, respectively, but the three models can be largely influenced by feature preprocessing, tuning, dataset segmentation methods, and different scenarios (Zhou, 2021). In realistic financial settings, the choice of credit approval models must take more factors into account, such as the economic budget and company size. A high accuracy rate doesn’t include everything (Baesens et al., 2003).

Class–imbalance data resampling techniques

There are generally significant class imbalances in credit datasets. In credit risk prediction, some scholars argue that bad customers being misclassified as good customers (Type I error) has more serious consequences than misclassifying good customers as bad customers (Caouette et al., 2008), and that many judgment metrics are more biased towards the majority category samples, so oversampling techniques should be used to increase the proportion of minority category samples. However, other voices argue that if only minority categories are emphasized. Then far more good customers will be rejected than bad customers will be accepted, which is also a big loss for financial institutions, so the majority category sample and minority category sample are equally important. Therefore, it is important to ensure that the predictive power of the minority class sample is not weakened while not overemphasizing the majority class sample. Although the hyperparameters of some models can adjust the weights on different categories of data (so that the influence of minority category samples is stronger), due to the different class balancing techniques and effects of different models, in order to facilitate cross–sectional comparisons between models, this paper uses over–sampling, under–sampling, and integrated sampling techniques to enhance the predictive power of minority category samples. The methods are synthetic data method (SMOTE), adaptive integrated oversampling (ADASYN), SMOTE with TomekLink (SMOTETomek), and Cluster Centroid. We will comprehensively compare the effects of these four methods.

SMOTE. Its essence is an oversampling processing technique that borrows the idea of the k–neighborhood algorithm to generate new samples from a few classes of samples to add to the data set, calculated as

$$x_{new} = x + \text{rand}\,(0,1) * |x - xn|$$

where x is the minority class sample and xn denotes the randomly selected nearest neighbor (Chawla, 2003).

Adaptive integrated oversampling (ADASYN). It is also a distance–based proximity sampling technique but focuses more on sampling a few classes of samples at the boundary (Haixiang et al., 2016). The calculation procedure is as follows:

Based on the data imbalance $d = \frac{m_s}{m_l}$, the number of samples to be synthesized is calculated, where β is used to control the parameter $G = (m_l - m_s)\beta$ for the sample balance. The k–neighborhood of each minority class sample x_i is subsequently calculated using the Euclidean distance and the full majority class case around is obtained. $r_i = \frac{\Delta_i}{K}, i = 1, \ldots, m,$ and Δ_i is the number of samples belonging to the majority class in the k–neighborhood, where $\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i$.

The number of minority class samples to be synthesized is calculated for each minority class sample, and 1 minority class sample is selected among the k neighbors of the minority class sample to be synthesized based on repeated synthesis until the number of required syntheses calculated in the previous step is satisfied.

$$g_i = \hat{r}_i \times G$$

$$s_i = x_i + (x_{zi} - x_i) \times \lambda, \quad \lambda \in [0,1]$$

SMOTETomek. It is an integrated sampling method. The principle is to first expand the sample size using the over–sampling method of SMOTE, and subsequently use the under–sampling method of TomekLink to pair the positive samples closest to the negative samples one by one, and then remove these pairs to achieve the effect of strengthening the decision boundary.

ClusterCentroids. It is an under–sampling method based on $K - means$ clustering algorithm. The specific implementation steps are:

- Cluster the dataset using k–means unsupervised clustering algorithm to get k_1 centers.
- Calculate the distance between each center to all majority class samples and remove the closest part of majority class samples.
- Continue to cluster the remaining samples to get k_2 centers and repeat to calculate the distance between k_2 to all majority class samples and remove the closest of some.
- Repeat the above three steps until the number of positive and negative samples is balanced.

Research Methodology

The Research Methodology section of the paper outlines a structured approach to data engineering and model fitting for credit risk assessment. The methodology is divided into four key steps: Data Recoding: The paper specifies that some classifiers, such as CatBoost, do not require one-hot coding, thus eliminating the need for this operation in those cases. Feature Selection: The initial screening of features is based on logistic regression to calculate information value (IV). After fitting the model with a trained Random Forest, the features are further screened based on information gain. This is described as a kind of embedded feature screening. Imbalanced Data Processing: Various resampling techniques like SMOTE, ADAYSN, SMOTETomek, and ClusterCentroid are used to address class imbalance in the data. The performance of classifiers is then compared both with and without the use of these balancing techniques. Model Fitting: Bayesian tuning algorithms based on Tree-structured Parzen Estimator (TPE) are used. This is particularly important for algorithms like XGBoost, which are prone to overfitting and have multiple hyper-parameters to control.

Research Methodology Overview:

Sampling Design: For empirical analysis, we utilize the German Credit dataset, a well-regarded benchmark dataset that comprises 1000 samples and 18 features.

Measurement Design: The efficacy of the balancing techniques is assessed using various performance metrics, including accuracy, precision, recall, and F1-score.

Analysis Design: A comparative analysis is conducted to evaluate the strengths and weaknesses of these balancing techniques when applied to different ensemble learning classifiers.

The methodology aims to provide a comprehensive approach to credit risk modeling, from feature selection to model tuning, while also addressing the challenges posed by imbalanced data.

Data engineering

Data recoding. This operation will not be applied to some classifiers that do not require one-hot coding, such as Catboost.

Feature selection. Initial screening of features based on logistic regression to calculate information value (IV) values (Tsai, 2009); after model fitting with trained Random Forest to generate information gain, continue backward iterative screening of features. Overall, it is a kind of embedded feature screening (Xia et al., 2018; Xia et al., 2017).

Imbalanced data processing. The class imbalance data were resampled using SMOTE, ADAYSN, SMOTETomek, and ClusterCentroid methods, respectively. And the performance of the classifier without using any balancing techniques and after using these three techniques separately are subsequently compared. Figure 1 presents the general technical approach of this paper.

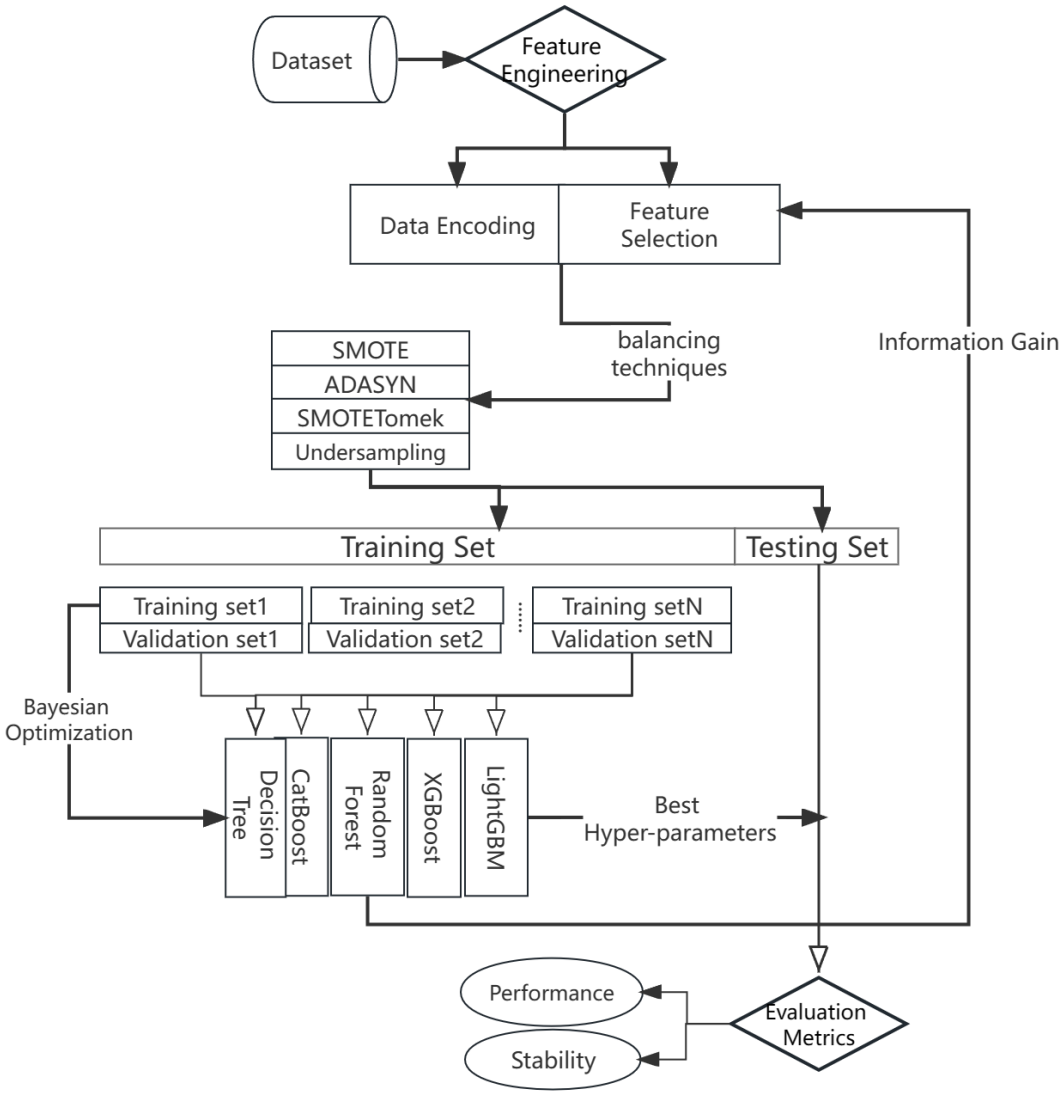

Fig. 1 Technology roadmap. The squares indicate the specific algorithms used in each section; the black arrows indicate the order of the steps; the white arrows indicate the application of the algorithm to the data set.

Model Fitting

This part is divided into two main levels. (a) Using a Bayesian tuning algorithm based on a tree–structured Parzen estimator (TPE) This is due to the fact that XGBoost, etc., as naturally prone to over–fitting tree models, have more hyper–parameters to control in order to find the optimal pruning strategy, and the best hyper–parameter configuration is huge. In contrast, the Bayesian approach of TPE has the advantage of supporting all types of parameter search and taking less time compared to other hyper–parameter optimization (HPO), which has good global search capability but does not easily fall into local optimums (Yang & Abdallah, 2020)

In the ensemble model, we focus on the n_estimators to control the number of "trees" in the integrated model (Figure 2) and the "max depth" parameter to control the growth of the trees. (b) Based on the Bayesian HPO process (Table 2), we compute the combination of hyper–parameters that performs optimally on the validation set. This combination is then used to train the model on the training set.

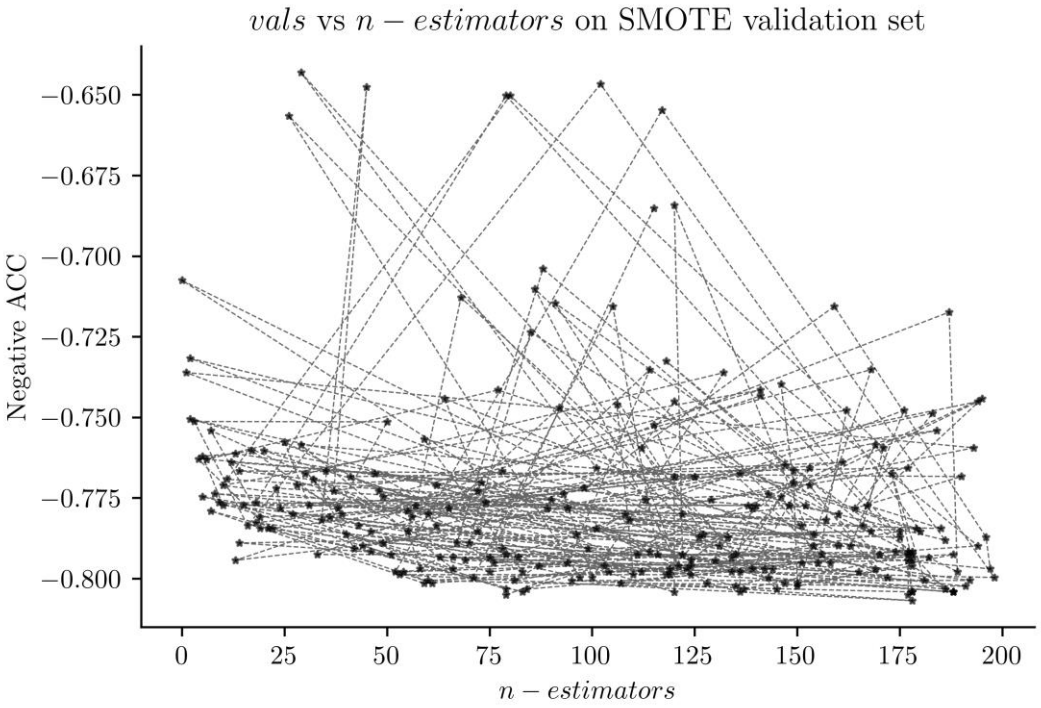

Fig. 2 TPE–Bayesian HPO process of finding Random Forest's best numbers of estimators according to accuracy scores

Table 2. HPO list, type and search configure

Model	Hyper-parameter	Type	Search Configure
DT Classifier	Criterion	categorical	[‘entropy’,’gini’]
	max_depth	discrete	[1,5]
	min_impurity_decrease	continuous	[0,0.5]
	min_samples_leaf	discrete	[3,50]
	splitter	categorical	['best','random']
RF Classifier	n_estimators	discrete	[1,200]
	Criterion	categorical	[‘entropy’,’gini’]
	max_depth	discrete	[1,20]
	min_samples_leaf	discrete	[1,20]
	min_samples_split	discrete	[1,20]
	max_leaf_nodes	discrete	[100,200]
	max_features	discrete	[1,20]
XGB Classifier	n_estimators	discrete	[50,150]
	max_depth	discrete	[1,10]
	learning_rate	continuous	[0,0.5]
	objective	categorical	['binary:logistic','binary:hinge']
	reg_alpha	continuous	[0,3]
	reg_lambda	continuous	[0,3]
	gamma	continuous	[1,10]
	colsample_bytree	continuous	[0,1]
	colsample_bylevel	continuous	[0,1]
LGBM Classifier	n_estimators	discrete	[20,300]
	max_depth	discrete	[1,5]
	learning_rate	continuous	[0,0.2]
	reg_alpha	continuous	[0,5]
	reg_lambda	continuous	[0,60]
	num_leaves	discrete	[5,30]
Catboost Classifier	learning_rate	continuous	[0,0.2]
	max_depth	discrete	[1,5]
	min_data_in_leaf	discrete	[10,30]
	one_hot_max_size	discrete	[1,10]
	max_ctr_complexity	discrete	[1,5]

Model evaluation

The evaluation metrics are employed mainly around accuracy and stability. Most of these metrics were calculated based on the confusion matrix. The detailed calculation formulas are shown in Table 3 and Table 4.

Table 3 Confusion matrix

	Prediction	
	Positive	Negative
Truth Positive	TP	FN
Negative	FP	TN

Table 4 Evaluation metrics

Evaluation metrics	**Formulas**
Accuracy (proportion of correctly classified samples)	$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$
Recall rate (% of bad customers)	$Recall = \frac{TP}{TP + FN}$
Precision (ability to predict bad customers)	$Precision = \frac{TP}{TP + FP}$
F1 value (considering both recall and accuracy)	$F1\ score = \frac{2TP}{2TP + FP + FN}$
Area under the ROC curve (model learning performance)	$AUC = \frac{1}{2}(1 + \frac{TP}{TP + FN} - \frac{FP}{FP + TN})$

It should be noted that accuracy does not reflect the true level of the classifier under imbalanced data, so then classifiers that do not use resampling techniques need to be evaluated in combination with other evaluation metrics. AUC is more friendly to imbalanced datasets (Garcia, 2015). Also, as we mentioned in the previous section, unlike other application scenarios, the predictive power of the positive sample (bad credit) is more important.

Data Analysis and Results

We have adopted the German Credit dataset from the UCL database. It is a benchmark dataset which has been used by a large number of scholars for comparison of various models (Baesens et al., 2003; Hand & Henley, 1997), with a total of 1000 samples, 18 features and a binary label (we have removed the gender feature due to possible legal issues involved). All empirical analyses were run via Python 3.10.9 64-bit (Spyder IDE).

Based on the IV screening and the embedding method, which uses Random Forest to assess the importance of features, we listed six features with an IV less than 0.02: guarantors, people_liable, residence, telephone, job, and number_credit, where number_credits, job, people_liable, and telephone are also at the bottom of the feature gain ranking. Therefore, we excluded all the features with an IV less than 0.02 and the one with the lowest variable gain (foreign_worker). The remaining 12 features were used for model training (Table 5).

Table 5. German credit dataset

Feature	Type	Details
status	categorial	status of existing checking account
duration	numerical	credit Duration in month
credit_history	categorial	credit_history
purpose	categorial	purpose
credit_amount	numerical	credit amount
savings	categorial	savings account/bonds
employment_duration	categorial	present employment duration
~~guarantors~~	categorical	other debtors
~~residence~~	numerical	present residence since
property	ordinal	property
age	numerical	age
installment_plans	categorical	other installment plans
installment rate	numerical	installment rate in percentage of disposable income
housing	categorical	house ownership status
~~number_credits~~	ordinal	number of existing credits at this bank
~~job~~	ordinal	employment ability
~~people_liable~~	ordinal	number of people being liable to provide maintenance for
~~telephone~~	categorical	landline in the debtor's name
~~foreign_work~~	categorical	whether foreign worker
Label	**Type**	**Details**
credit_risk	categorical	credit contract been complied with good or bad

Three of the 12 features are continuous features (duration, credit_amount, and age), which have been log–transformed. All types of models in this paper are able to perform self–binning of the continuous variables and have good calling ability for categorical features (among which CatBoost can automatically perform one–hot encoding). The distribution of each variable is shown in Figure 3. We split the dataset into a training set and a test set in the ratio of 8:2. The validation set is extracted from the training set.

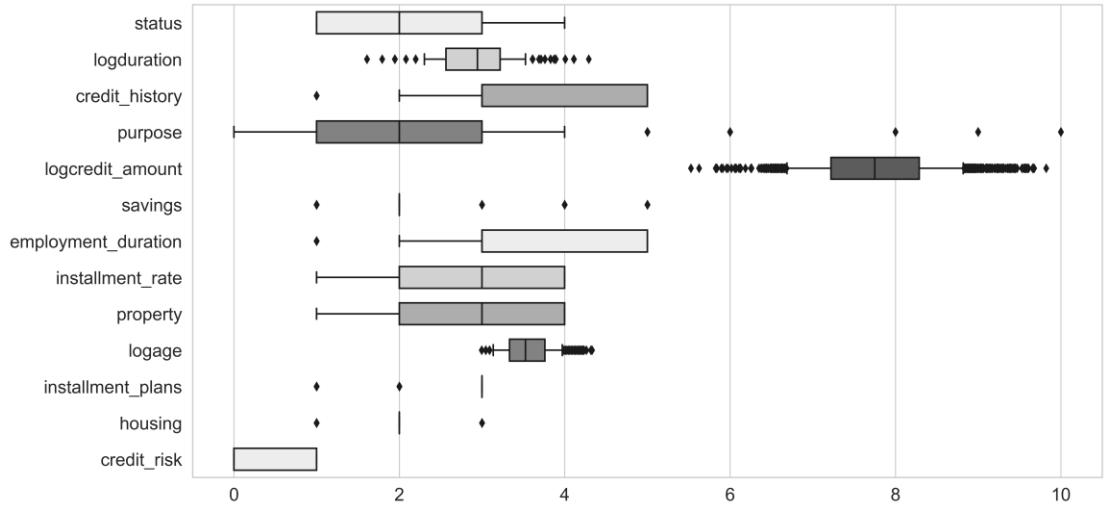

Fig.3 Boxplot of selected features after standard scale. The diamond-shaped points represent some discrete points

Labeled with credit_risk, the ratio of positive to negative samples in the dataset is 7:3, which is a relatively serious imbalanced dataset. We mark “bad credit” as 1, “good credit” as 0, which means positive samples and negative samples. SMOTE, ADASYN, SMOTETomek and ClusterCentroid will be applied to the training set separately, which transfers the original training set into four different datasets. In the following, the original data are referred to "original" and the new data sets obtained by applying the four types of resampling techniques are referred to as "SMO", "ADA", "TOM" and "CC" respectively.

The ratio of positive and negative samples is adjusted to 1:1 for SMO, TOM, and CC (Table 6). It should be noted that in order to avoid data leakage, we will apply the data balancing techniques to the test set separately when model training is finished completely.

Table 6. Proportion of positive and negative samples on different training sets

Training set	Good credit (Negative sample)	Bad credit (Positive sample)	positive percentage
Original	559	241	30.13%
SMO	559	559	50.00%
ADA	559	534	48.86%
TOM	541	541	50.00%
CC	241	241	50.00%

Different sets of five training data sets were cross-validated 10-fold, and the resulting validation sets obtained different scores. We filtered the optimal hyperparameter combinations to fit the DT, RF, XGB, LGBM, and CatBoost five models based on accuracy and AUC metrics. All

scores of different models on different datasets are given on the test set (Table 7). The following are our conclusions based on the scores:

(1) All models outperformed the original dataset substantially on the dataset where class balancing techniques were used. Although the original dataset still achieves good scores in AUC and accuracy, the scores in Precision, Recall, and F1 are very low, and the models become extremely unstable. This indicates that effective gains were obtained using the balancing techniques. This is because in the credit risk review process, we should usually try to avoid having customers with bad credit incorrectly judged as good. Models with high accuracy rates cannot be useful in real-life scenarios either.

(2) RF and XGB classifiers work better than other classifiers by getting higher accuracy and stability scores. In turn, all the ensemble classifiers have better performance than a single decision tree. It means that for small datasets, the traditional integrated tree model is sufficient to cope with them. More advanced models would instead lead to a degradation of model performance due to overfitting.

(3) In terms of balancing techniques, SMO and TOM datasets perform better than ADA and CC on all types of classifiers. Among them, SMO has better scores on RF and LGBM. TOM has higher scores on DT and XGB. It is worth mentioning that CC obtained the best score on the CatBoost. This indicates that CatBoost still has excellent anti-overfitting ability when facing small samples. However, generally, we still do not recommend under-sampling techniques on datasets that are not large enough. The integrated sampling technique SMOTE-Tomelink is a better choice.

Table 7. Performance of ensemble classifiers on different test sets

	Test set	AUC	Accuracy	Precision	Recall	F1
DT Classifier	Original	0.6543	0.7350	0.5625	0.4576	0.5047
	SMO	0.7163	0.7163	0.6993	0.7589	0.7279
	ADA	0.7002	0.7007	0.6768	0.7762	0.7231
	TOM	**0.7566**	**0.7566**	0.7417	**0.7876**	**0.7639**
	CC	0.7458	0.7458	**0.7843**	0.6780	0.7273
RF Classifier	Original	0.6715	0.7800	0.7273	0.4068	0.5217
	SMO	**0.8121**	0.8120	**0.8492**	**0.7589**	0.8015
	ADA	0.7961	0.7958	0.8295	0.7483	0.7867
	TOM	0.7965	0.7965	0.8252	0.7522	0.7870
	CC	0.8136	**0.8136**	0.7937	0.8475	**0.8197**
XGB Classifier	Original	0.6707	0.765	0.6500	0.4407	0.5252
	SMO	0.8156	0.8156	0.8560	0.7589	0.8045
	ADA	0.8032	0.8028	0.8372	0.7552	0.7941
	TOM	**0.8186**	**0.8186**	**0.8600**	0.7611	0.8075
	CC	0.8051	0.8051	0.7903	**0.8305**	**0.8099**
LGBM Classifier	Original	0.6849	0.7850	0.7222	0.4407	0.5474
	SMO	**0.7908**	**0.7908**	0.8306	**0.7305**	**0.7774**
	ADA	0.7680	0.7676	0.8080	0.7063	0.7537
	TOM	0.7788	0.7787	**0.8387**	0.6927	0.7573
	CC	0.7458	0.7458	0.7544	0.7288	0.7414
Catboost Classifier	Original	0.6811	0.7500	0.6200	0.500	0.5536
	SMO	0.7500	0.7501	0.7041	**0.8623**	0.7752
	ADA	0.7955	0.7953	0.7879	0.7939	0.7909
	TOM	0.7573	0.753	0.7524	0.7670	0.7596
	CC	**0.8065**	**0.8065**	**0.8167**	0.7903	**0.8033**

All types of ensemble algorithms come with their own balancing techniques. A hyper–parameter, "scale_pos_weight," of XGB allows the model to focus on the information contained in a small number of samples by setting the proportion of positive and negative samples. We set "scale_pos_weight at a range of [1, 30] to observe its effect on the AUC score (Figure 4). We found that the original set could only obtain a maximum AUC score of 0.7631, which is much lower than the score with the balancing technique (even CC could obtain an AUC score of 0.8051 on the XGB). On the other hand, we also adjusted the 'class_weight' parameter in LGBM to learn the class ratio of the data before starting training. Unfortunately, this setting still only helped our original set get an accuracy score of 0.735 and an F1 score of 0.5827, which is still a poor performance. The conclusion indicates that even though models themselves carry a mechanism to

deal with imbalance problems, applying balancing techniques to the dataset can significantly improve learning ability and stability.

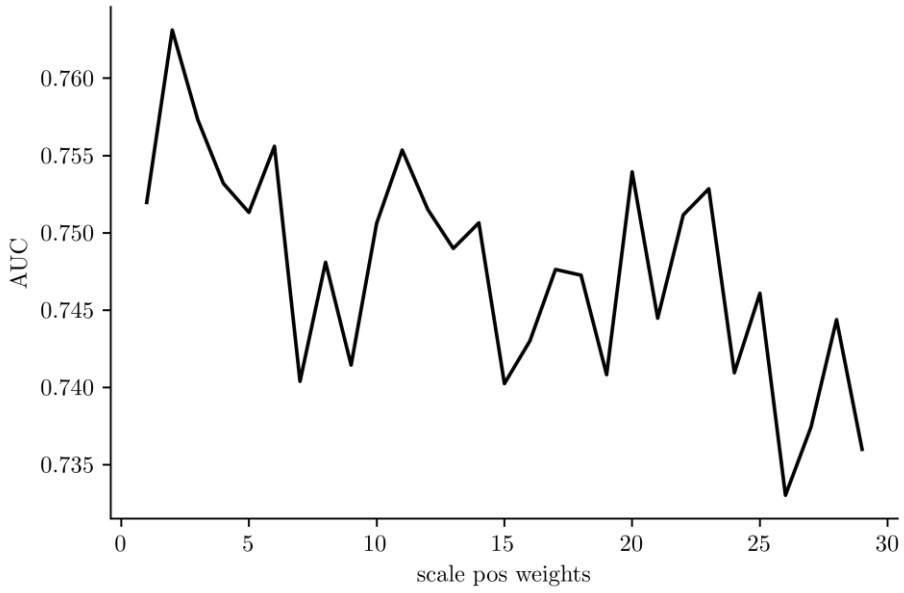

Fig. 4 Hyper–parameter 'scale_pos_weights' effect on AUC score in XGB classifier

DT or ensemble models with decision trees have relatively good interpretability. By getting feature importance scores, we can conclude that:

(1) Different models score each feature differently and provide different rankings. However, in general, "credit_amount" is rated as the first influential feature by RF, LGBM, and CatBoost; "status" is rated as the first influential feature by XGB and the second important feature by RF and CatBoost. "Age" and "credit history" are also considered significant features by all classifiers (Figures 5, 6, 7, 8).

(2) The Original, ADA, SMO and TOM datasets showed almost no difference in feature importance ranking in the RF model, but CC dataset show a different result. RF considered "age" and "installation_rate" to be the most important feature on CC dataset (Figure 5). This indicates that the reduced sample size due to the under–sampling technique could cause the loss of data information, which affects the judgment of the model. This phenomenon also exists in XGB, LGBM and CatBoost as well (Figure 6, 7, 8).

(3) The SMO, ADA, and TOM datasets are not significantly different in each model. Among them, SMO and TOM are closer and can be regarded as almost identical datasets. It indicates that SMOTE oversampling is more stable than ADASYN oversampling.

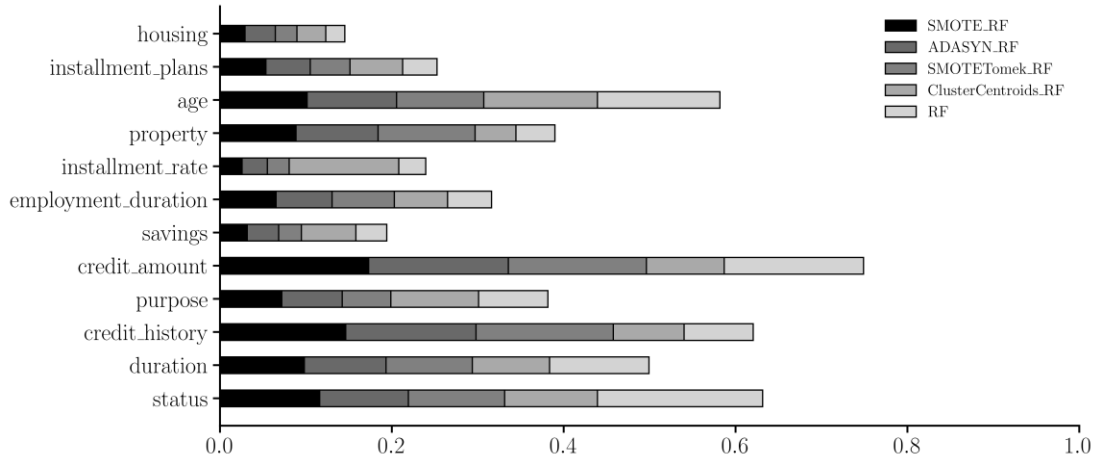

Fig. 5 Bar chart of RF feature importance on different test sets. Importance scores are indicated by the length of the different colored bars (e.g., the black bar (SMOTE_RF) and the dimgray bar (ADASYN_RF) in 'status' are of similar length, indicating that status has the same importance score in both datasets).

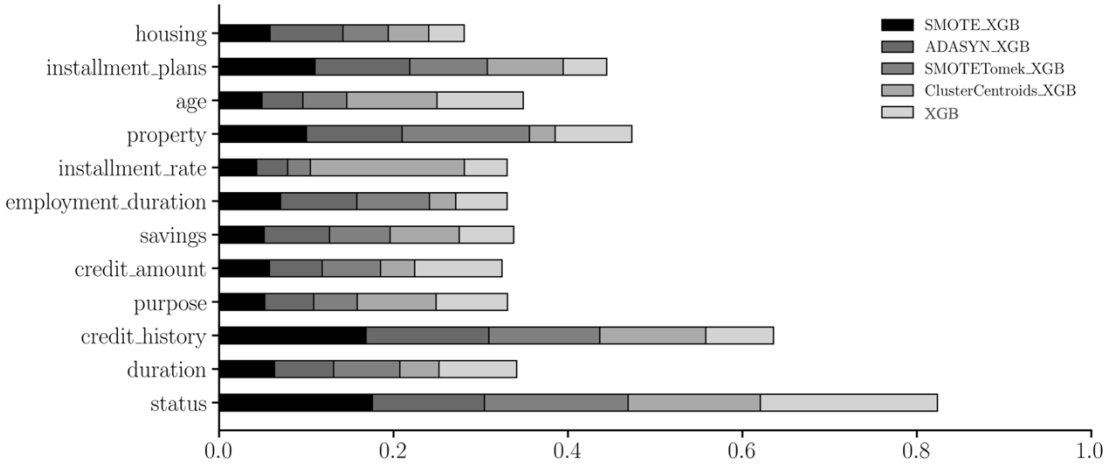

Fig. 6 Bar chart of XGB feature importance on different test sets.

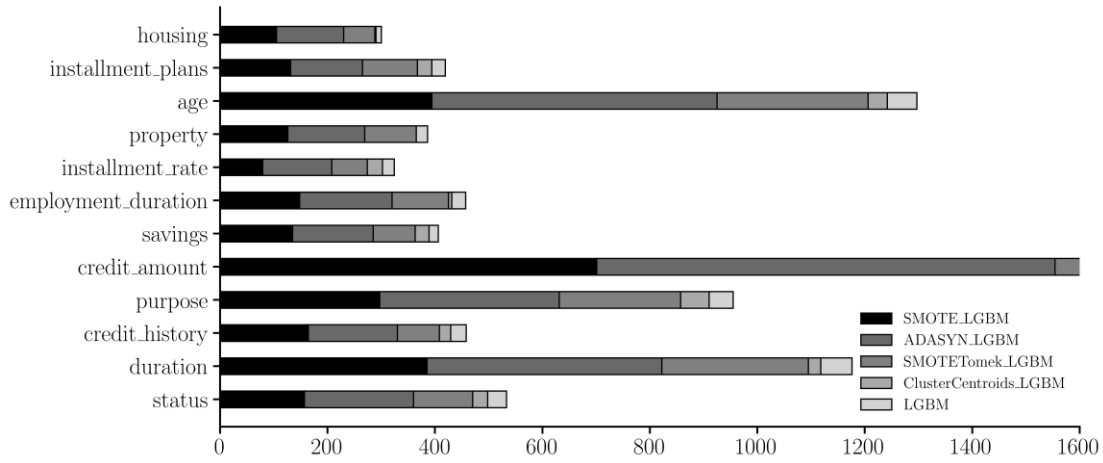

Fig. 7 Bar chart of LGBM feature importance on different test sets

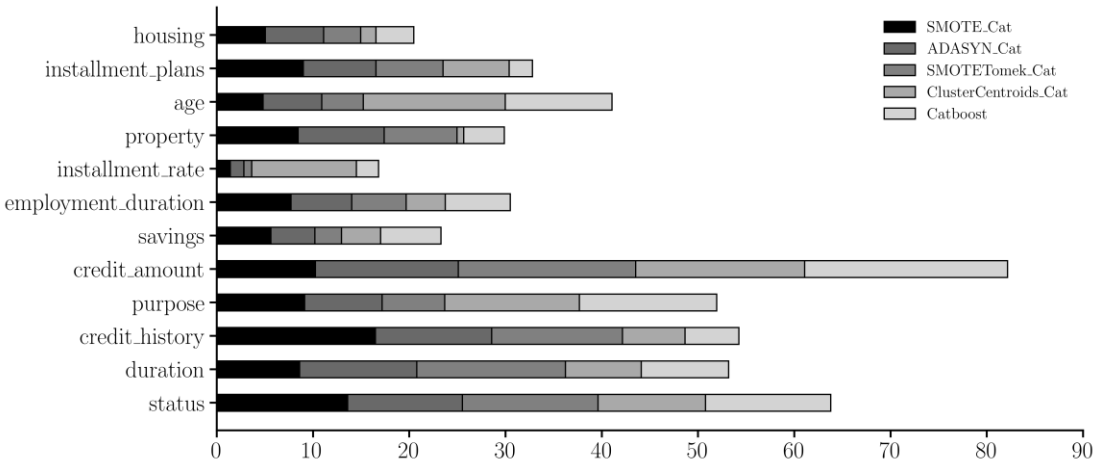

Fig. 8 Bar chart of CatBoost feature importance on different test sets.

Conclusion

This paper focuses on evaluating the performance of decision tree–based ensemble learning classifiers on datasets with different class balancing techniques. The conclusions show that using balancing techniques in advance for the datasets can significantly improve the learning and predictive abilities of the models. Among them, we more often recommend using SMOTE and SMOTETomek techniques on small datasets. However, oversampling and integrated sampling techniques may exacerbate the risk of overfitting, especially in ensemble learning of tree structures.

It is necessary to pair them with appropriate pruning strategies to prevent overfitting. If oversampling or integrated sampling techniques are applied, careful tuning of hyperparameters is required, and even applying fast tuning means like TPE Bayes can cost more in response time than financial institutions can afford in a complex classifier like CatBoost. Therefore, how to balance model performance and time consumption remains a topic for continued research in credit risk prediction.

Acknowledgments

This research was supported by Yunnan University of Finance and Economics Scientific Research Fund Project of China (Grant No. 2021B01) and National Institute of Development Administration.

Reference

Alessi, L., & Detken, C. (2018). Identifying Excessive Credit Growth and Leverage. *Journal of Financial Stability*, *35,* 215–225. https://doi.org/https://doi.org/10.1016/j.jfs.2017.06.005

Auronen, L. (2003). *Asymmetric Information: Theory and Applications.* Helsinki University of Technology. Retrieved from https://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1 &type=pdf&doi=10.1.1.198.9252

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking State–of–the–art Classification Algorithms for Credit Scoring. *Journal of the Operational Research Society, 54*, 627–635.

Caouette, J.B., Altman, E.I., Narayanan, P., & Nimmo, R. (2008). Economic Capital and Capital Allocation: Chapter 19. *In Managing Credit Risk: The Great Challenge for the Global Financial Markets* (2nd ed.). Wiley. https://doi.org/10.1002/9781118266236.ch19

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over–sampling technique. *Journal of Artificial Intelligence Research, 16,* 321–357.

Chawla, N., Lazarevic, A., Hall, L., & Bowyer, K. (2003). *SMOTEBoost: Improving Prediction of the Minority Class in Boosting* (Vol. 2838). https://doi.org/10.1007/978–3–540–39804–2_12

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

Daoud, E. A. (2019). Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset. *International Journal of Computer and Information Engineering*, *145,* 6–10. https://publications.waset.org/pdf/10009954

Dastile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing, 91,* 106263. https://doi.org/https://doi.org/10.1016/j.asoc.2020.106263

Dobbie, W., & Skiba, P. M. (2013). Information Asymmetries in Consumer Credit Markets: Evidence from Payday Lending. *American Economic Journal: Applied Economics, 5*(4), 256–282. doi:10.1257/app.5.4.256

Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv*:1810.11363.

Galindo, J., & Tamayo, P. (2000). Credit Risk Assessment Using Statistical and Machine Learning: Basic Methodology and Risk Modeling Applications. *Computational Economics,* 15(1), 107–143. https://doi.org/10.1023/A:1008699112516

Gamini, P., Yerramsetti, S., Darapu, G., Pentakoti, V., Prudhvi, V. (2021). Detection of Credit Card Fraudulent Transactions using Boosting Algorithms. *Journal of Emerging Technologies and Innovative Research*, 8(2), 2031–2036.

García, V., Marqués, A. I., & Sánchez, J. S. (2015). An insight into the experimental design for credit risk and corporate bankruptcy prediction systems. *Journal of Intelligent Information Systems, 44*(1), 159–189. https://doi.org/10.1007/s10844–014–0333–4

Haixiang, G., Li, Y., Shang, J., Mingyun, G., Yuanyue, H., & Gong, B. (2016). Learning from class–imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73.* https://doi.org/10.1016/j.eswa.2016.12.035

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society), 160*(3), 523–541.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Paper presented at *the 2008 IEEE International Joint Conference on Neural Networks* (IEEE world congress on computational intelligence).

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., . . . Liu, T.–Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems,* 30.

Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review, 39,* 261–283.

Kumar, S., Ahmed, R., Bharany, S., Shuaib, M., Ahmad, T., Tag Eldin, E., . . . Shafiq, M. (2022). Exploitation of Machine Learning Algorithms for Detecting Financial Crimes Based on Customers' Behavior. *Sustainability, 14*(21), 13875. https://doi.org/10.3390/su142113875

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.

Ma, X., Sha, J., Wang, D., Yu, Y., Yang, Q., & Niu, X. (2018). Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning. *Electronic Commerce Research and Applications, 31*, 24–39. https://doi.org/https://doi.org/10.1016/j.elerap.2018.08.002

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems,* 31.

Stiglitz, J. E., & Weiss, A. (1981). Credit Rationing in Markets with Rationing Credit Information Imperfect. *The American Economic Review*, *71,* 393–410.

Tsai, C.–F. (2009). Feature selection in bankruptcy prediction. *Knowl. – Based Syst.*, *22*(2), 120–127.

Wang, T., Zhao, S., Zhu, G., & Zheng, H. (2021). A machine learning–based early warning system for systemic banking crises. *Applied Economics*, *53*(26), 2974–2992. https://doi.org/10.1080/00036846.2020.1870657

Witten, I. H., & Frank, E. (2002). Data mining: practical machine learning tools and techniques with Java implementations. *Acm Sigmod Record, 31*(1), 76–77.

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

Xia, Y., Liu, C., Da, B., & Xie, F. (2018). A novel heterogeneous ensemble credit scoring model based on bstacking approach. *Expert Syst. Appl.*, *93*(C), 182–199. https://doi.org/10.1016/j.eswa.2017.10.022

Yang, L., & Shami, A. (2020). On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. *Neurocomputing*, *415,* 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

Zhou, L., Fujita, H., Ding, H., & Ma, R. (2021). Credit Risk Modeling on Data with Two Timestamps in Peer–to–Peer Lending by Gradient Boosting. *Applied Soft Computing*, *110*, 107672. https://doi.org/https://doi.org/10.1016/j.asoc.2021.107672