Credit evaluation with a data mining approach based on gradient boosting decision tree

Zhenlong Liu1* and Yuheng Zhang2

1,2 School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

*Corresponding author’s e-mail: S181231018@stu.cqupt.edu.cn

Abstract. In recent years, credit evaluation has become an issue of increasing concern for financial institutions. However, since most research focuses on the risk classification process, the problem of data imbalance is ignored. In real data sets, there are often far more users with good credit than users with bad credit, and the imbalance of data often easily leads to a decline in the classification performance of the model. Therefore, previous research is very limited in practical application scenarios. In this paper, we establish a new integration method for credit evaluation, which is classified into three steps: First, data preprocessing. Before inputting samples into the model, we take a series of preprocessing steps, such as missing data processing, data dimensionality reduction. Secondly, in view of the imbalance problem, the data is divided into multiple clusters using an unsupervised clustering algorithm, and the SMOTE method is used to generate minority samples in the clusters whose ratio exceeds the threshold. Finally, the GBDT2NN and Factorization Machine methods are integrated to classify the samples. In order to verify the effectiveness of this method, we use the Kaggle competition data set for verification. The results show that this method is better than other algorithms in the field of credit evaluation in terms of recall rate and AUC value.

1. Introduction
Since the subprime mortgage crisis in the United States, financial institutions have suffered huge losses. Therefore, credit evaluation has become more and more important in the financial industry, and more and more financial institutions have begun to pay attention to credit risks. In order to effectively manage credit risks and reduce loan risks of financial institutions, a set of efficient credit evaluation models have been developed.

It’s a fact that currently, many models have been used to deal with credit evaluation problems. In the field of machine learning, credit evaluation is usually used as a two-classification task. Through the past historical data and personal information of the user, effective features are excavated, learning and training are carried out, and the user's credit is judged. Because of its simplicity and accuracy, linear discriminant and logistic regression are still the most popular credit evaluation models [1]. However, current research shows that a single classifier cannot solve the credit evaluation problem well [2], because different data sets have different structures and characteristics. For the ensemble model, multiple basic classifiers can be used to extract the features of the data from multiple dimensions, and the results of multiple basic classifiers can be combined for comparison to obtain a better classification result. According to the research of Lessmann et al. (2015) [3], the ensemble method is better than a single artificial intelligence and statistical method.
In real data, there are often a lot of problems, such as data imbalance, data sparseness, and data missing. The first is the imbalance problem. In the collected data set, the number of applicants with good credit is usually much greater than the number of credit defaulters. Due to the lack of sufficient data, the classifier's ability to describe sparse samples is insufficient, and it is difficult to effectively classify these sparse samples. The second is the problem of data sparseness. Since credit evaluation requires a certain understanding of personal information, there will be a large number of text characteristics, such as occupation, marital status. Finally, there is the problem of missing. In the real data, it is impossible to ensure that every sample has complete data. Missing data is inevitable, the processing of missing values will have a non-negligible impact on the final accuracy of the model.

In order to solve the above problems, this article proposes a set of solutions from the perspectives of data imbalance processing, data preprocessing, and classification models. For data imbalance, the K-means and SMOTE method is used to generate minority samples; for the data preprocessing stage, we use a series of dimensionality reduction algorithms and data missing processing operations to perform experiments on the data; in the final classification model stage, a combination of GBDT2NN and FM is used to integrate high-level feature combinations and low-level feature combinations to more comprehensively distinguish between good and bad credit.

The rest of this article is organized as follows. In the second chapter, several methods or models related to credit evaluation are described. Chapter 3 mainly introduces our own processing process and methods. In Chapter 4, in order to further verify the effectiveness of the method, we use a typical credit evaluation data set to verify its effectiveness. Finally, in Chapter 5, we summarize the content of this article.

2. Related Work

This chapter will introduce the research of credit evaluation related fields, mainly from two aspects of imbalance processing and classification algorithms.

2.1. SMOTE(Synthetic Minority Over-sampling Technique)

The serious imbalance between the categories in the classification problem is a very common problem. For example, in credit evaluation and other fraud problems, fraud observations are a minority in the sample set. In order to solve this problem, in 2002, Chawla NV and others proposed the SMOTE algorithm [4], which is a broad and effective method to solve the problem of class imbalance. The core idea of the SMOTE algorithm is: SMOTE uses KNN technology to simulate and generate new sample data. First, use Euclidean distance to calculate the K neighbors of each minority sample, and then randomly select N samples from these K neighbors for random linearity Interpolate, and construct new minority samples, and finally synthesize the new samples with the original data to generate a new data set. The specific synthesis formula is as follows:

$$x_{\text{new}} = x_i + (x_{i}^{k} - x_i) \times \delta$$

Where $x_{\text{new}}$ represents the newly generated sample, $x_{i}^{k}$ is one of all the neighbors of $x_i$, and $\delta$ is a random value belonging to (0,1).

However, SMOTE is not perfect. The algorithm is prone to the problem of marginalization in the process of generating minority samples. Although the ambiguity of the boundary improves the balance of the data set, it increases the difficulty of the classification algorithm. In recent years, there have been many improvements to the SMOTE method. For example, in 2005, Han H et al. proposed an algorithm called Borderline-SMOTE [5], which was improved on the basis of the SMOTE algorithm to solve The problem of generating sample overlap (Overlapping) is solved.

2.2. GBDT(Gradient Boosting Decision Tree)

Many previous studies have shown that the integrated method is better than a single artificial intelligence and statistical method. Among the integrated methods, gradient boosting has always been one of the most popular techniques. Gradient boosting is a method used for regression and classification. Boosting is an algorithm that can upgrade a basic learner to a strong learner, and belongs to the category of
ensemble learning[6]. The Boosting method is based on the idea that for a complex task, the judgment obtained by appropriately synthesizing the judgments of multiple experts is better than the judgment of any one of the experts alone. The Boosting algorithm builds models in a stage-wise manner, and the weak learner is built at each step of the iteration to make up for the deficiencies of the existing model. At the same time, the learner based on the gradient boosting algorithm is called GBM (Gradient Boosting Machine). In theory, we can choose a variety of different learning algorithms as the basic learner. But in reality, the most commonly used decision tree is the decision tree, which can be considered a collection of if-then rules, which is easy to understand, interpretable, and fast in prediction. At present, there have been a lot of researches based on Boosting algorithm. For example, as early as 1997, Y Freund et al. proposed the AdaBoost algorithm [7]. Yufei Xia et al used the XGBoost[8] model in credit evaluation in 2017, and used the Bayesian hyperparameter optimization [9] method to adaptively adjust the XGBoost hyperparameters [10], and achieved good experimental results. Two commonly used Boosting algorithms will be introduced in detail below:

2.2.1. LightGBM. LightGBM is a gradient boosting framework proposed by Guolin Ke[11] in 2017. Like XGBoost, it is an efficient implementation of GBDT. In principle, it is similar to GBDT and XGBoost. Both use the negative gradient of the loss function as the current decision tree. Approximate residuals to fit a new decision tree. LightGBM performs better than XGBoost in many aspects, such as faster training speed, lower memory usage, support for parallel learning and so on.

2.2.2. DeepGBM. DeepGBM is an algorithm proposed by Ke G et al. in 2019[12]. The algorithm consists of two neural network components: (1) catnn, which focuses on sparse classification features. (2) GBDT2NN, using the knowledge extracted by GBDT2NN, focuses on intensive numerical features. With the support of these two components, deepgbm can simultaneously utilize classification and numerical features while maintaining the ability to efficiently update online. Among them, GBDT2NN uses distillation technology, uses the index output of the neural network and the model, and concisely obtains the output of the tree model unit, thereby learning the structure of the tree in the neural network.

3. Methods
In order to better describe the research method, a real data set on the Kaggle website competition-Home-Credit-Default-Risk is used. The data set will be introduced in detail in the experimental chapter.

3.1. Data preprocessing
Because in the real credit evaluation data set, the original data set often has a large number of problems, and the quality of the data greatly affects the accuracy of model prediction or classification. Next, we will introduce us from two aspects for credit evaluation. How to complete the data preprocessing, including missing value processing and discrete data processing. The specific process is shown in Figure 1.

![Figure 1. Process of the pretreatment stage.](image)

First, in our data set, there are a large number of missing values in both the training set and the test set. The missing ratios are shown in Figure 2. For these missing data, the missing ratios are calculated separately. When the missing ratios are greater than 80%, The coverage rate of features is low, and a large amount of filling data is easy to mislead the model, so these features are directly deleted from the
data set. In the remaining missing data, if the data conforms to a uniform distribution, the mean value of the variable is used to fill in the missing data. If the data has a sloping distribution, the median is used to fill in. At the same time, if the missing feature is a discrete feature with few different values, convert it into a dummy variable. For example, gender features, there are male, female, and blank missing values, and the column feature values can be converted to SEX_MALE, SEX_FEMALE, SEX_NA. For some features with a small missing ratio (less than 10%), simple prediction models such as decision trees, linear regression and other models can be used for prediction.

Secondly, for discrete data, most classification models require that the input data must be numeric when inputting. A commonly used encoding method is one-hot encoding. For discrete features with many different values, such as occupations. One-hot usually causes the encoded data set to be too sparse, and a large number of feature dimensions are increased. The sparse feature has a large number of 0 values, which causes the model to learn the sample features slowly and affects the classification effect. The usual approach is to perform dimensionality reduction processing on features. Several common dimensionality reduction methods include PCA (Principal Component Analysis) [13], LDA (Linear Discriminant Analysis) [14] and other methods. The goal of PCA is to map high-dimensional data to a low-dimensional space through some kind of linear projection, and expect the largest variance of the data in the projected dimension, so as to use less data dimensions while retaining more the characteristics of the original data point. LDA is a supervised linear dimensionality reduction algorithm. LDA minimizes the intra-class variance and the largest inter-class variance after projection. Compared with PCA, LDA is more suitable for classification tasks. For most data, a single dimensionality reduction method may not achieve good results, and some important features should not be dimensionality reduction compressed. We first use LightGBM to conduct basic experiments on the data set, and in the final model weight, select the top 5 important features and list them separately. At the same time, we conduct experiments on different dimensionality reduction algorithms on different types of data sets. The final result is that under the premise of using LightGBM as the basic classifier, LDA is compressed for numerical samples, for the sparse data after one-hot conversion, Use FM as the embedding method to get the best effect.

3.2. Unbalanced treatment
In the field of credit evaluation, data imbalance is a common problem, and whether the classification model can effectively mine users with poor credit has also become one of the criteria for measuring the quality of the model. Figure 2 shows the ratio of good credit to bad credit in our data set. At the same time, in credit evaluation, the evaluation criteria between users may also be very different: for example, young people are judged differently from middle-aged people. We propose to use the KMeans and SMOTE method to generate minority samples. First, divide the data into K clusters, divide users of the same type into one cluster, then determine the proportion of positive and negative samples in each cluster, and finally when the proportion of a small number of samples in a cluster is greater than a threshold $\alpha$, then in this cluster, SMOTE generation is performed on two samples. The advantage of this method is that, first, it can avoid the problem of confusion of decision-making boundaries. Through the data samples after clustering, the positive samples and the negative samples are distinguished, and the SMOTE method is generated only in the clusters where the proportion of negative samples exceeds the threshold; Second, avoid the confusion of the generated data. The credit evaluation data is concentrated, and the data distribution has its own rules. After clustering, the samples in each cluster will conform to the rules of the cluster, and then the samples generated by SMOTE will also conform to the rules. These laws; the third is to supplement data without loss of information. The advantage of SMOTE is that without discarding the original data, new data is generated without losing the original information.
3.3. Classification model

Classification model has always been the most critical module of credit evaluation. In recent years, a large number of studies have proved that the integrated method is better than a single artificial intelligence and statistical method. At the same time, GBDT can iteratively select the features with the largest statistical information gain to construct Tree [15,16]. However, GBDT also has its shortcomings. After the classification features are converted to sparse and high-dimensional one-hot encoding, the statistical information gain on the sparse features will become very small, because the gains obtained by using sparse features for unbalanced partitioning are compared with those obtained without partitioning. The gain is almost the same. Contrary to GBDT, since the backpropagation algorithm of neural network, the learning ability of sparse classification features through embedded structure has been proved [17,18], but for dense numerical features, it is caused by its fully connected model structure. Optimizing the hyperplane is very complicated, and it is easy to fall into the local optimum [14]. Since the two methods have their own advantages, Ioannou et al. [19] proposed to construct a tree-like neural network, such as GoogLeNet [20], which has tree-like decision-making ability to a certain extent. But GoogLeNet is more used in the field of computer vision, and does not consider table type data. At present, GBDT2NN uses Leaf Embedding Distillation and tree grouping methods, which can effectively extract multiple trees in GBDT into a compact NN model. In addition, in addition to the output of the tree, the feature selection and structural knowledge of the tree are also effectively extracted into the neural network model. This article separates the GBDT2NN part of the DeepGBM model separately. Because GBDT2NN only considers the impact of high-level feature combinations on the classification results, ignoring low-level features, we introduce the Factorization Machine [18] method, adding low-level features to the credit evaluation classification. The weight, taking into account the combination of high-level and low-level features, improves the model’s ability to judge good and bad credit. The specific formula is as follows:

\[ \hat{y}(x) = \sigma(w_1 \times y_{GBDT2NN}(x) + w_2 \times y_{FM}(x)) \]

Where \( \hat{y}(x) \) represents the final result, \( w \) is the weight, \( y_{GBDT2NN}(x) \) is the output result of GBDN2NN, and \( y_{FM}(x) \) is the output result of FM.

4. Experimental

4.1. Evaluation index

The evaluation index is an important factor to verify the reliability of the algorithm or model after the evaluation training is completed. In order to verify the classification performance of our method in the field of credit evaluation, we chose two performance indicators to measure the overall classification performance of FM-GBDT2NN and the three benchmark models. They are the area under the ROC curve (AUC) and the recall rate. In addition, the recall rate plays a vital role in credit evaluation. In institutions such as finance or banking, the cost of judging users with bad credit as good credit is much greater than the cost of judging good credit as bad credit. Therefore, the ability to detect users with poor credit is an important evaluation indicator of the model, and the recall rate can effectively evaluate the proportion of the target category in the focus area.
4.2. Experimental data set
We used the home-credit-default-risk dataset in the Kaggle competition. In order to effectively compare the experimental results, we randomly re-divide the data set, where three-quarters is the training set and one-quarter is the test set. The new training set has a total of 120 features and a total of 230,633 lines, including basic personal information, financial information, historical information and other credit information. Additional information about the data set is shown in Table 1:

|          | Home Credit Default Risk | Data size | The positive sample size | The negative sample size | Feature size |
|----------|--------------------------|-----------|--------------------------|--------------------------|--------------|
| **Train**|                          | 230633    | 18628                    | 212005                   | 120          |
| **Test** |                          | 76878     | 6197                     | 70681                    | 120          |

4.3. Experimental results
In order to verify the effectiveness of the method, we chose three benchmark models for comparison, namely logistic regression (LR), LightGBM, and GBT2NN. The experimental results are shown in the following table 2 and table 3:

| AUC   | Balance | Imbalance |
|-------|---------|-----------|
| GBDT2NN | 0.74106 | 0.75583   |
| LR     | 0.60581 | 0.62101   |
| LightGBM | 0.73466 | 0.75821   |
| FM-GBDT2NN | 0.75851 | 0.76564   |

| Recall | Balance | Imbalance |
|--------|---------|-----------|
| positive | negative | positive | negative |
| GBDT2NN   | 0.72379 | 0.74413  | 0.87092  | 0.33512  |
| LR        | 0.61613 | 0.54058  | 0.99998  | 0.        |
| LightGBM  | 0.64284 | 0.74024  | 1.        | 0.        |
| FM-GBDT2NN| 0.72418 | 0.74515  | 0.87254  | 0.26552  |

5. Conclusion
This paper proposes a credit scoring integrated classification method based on integrated learning Boosting. In the proposed method, the missing and sparse data are pre-processed firstly. Secondly, KMeans and SMOTE methods are used to generate data for imbalance. Finally, FM and GBDT2BB methods are integrated to combine high-order feature combinations with low-order feature combinations to construct a classifier. A series of solutions are also provided. Finally, a combination of FM and GBDT2NN methods is used to combine high-order feature combinations with low-order feature combinations to construct a classifier. The outputs of different classifiers are combined through weighted voting. Through theoretical analysis and experimental research, we prove that the proposed method can be generated in real data sets and achieve better results. Therefore, it is an effective and promising credit scoring method.

References
[1] Nie, G., Rowe, W., Zhang, L., Tian, Y., & Shi, Y. (2011). Credit card churn forecasting by logistic regression and decision tree. Expert Systems with Applications, 38, 15273–15285.
[2] Xia Y., Liu C., Li Y Y., et al. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring[J]. Expert Systems with Applications, 2017, 78: 225-241.
[3] Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of re-search. European Journal of
[4] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of artificial intelligence research, 2002, 16: 321-357.

[5] Han H, Wang W Y, Mao B H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning[J]. Lecture Notes in Computer Science, 2005.

[6] Freund Y, Schapire R, Abe N. A short introduction to boosting[J]. Journal-Japanese Society For Artificial Intelligence, 1999, 14(771-780): 1612.

[7] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting[J]. Journal of computer and system sciences, 1997, 55(1): 119-139.

[8] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.

[9] Golovin D, Solnik B, Moitra S, et al. Google Vizier: A Service for Black-Box Optimization[C]//the 23rd ACM SIGKDD International Conference. ACM, 2017.

[10] Xia Y, Liu C, Li Y Y, et al. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring[J]. Expert Systems with Applications, 2017, 78: 225-241.

[11] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[C]//Advances in Neural Information Processing Systems. 2017: 3146-3154.

[12] Ke G, Xu Z, Zhang J, et al. DeepGBM: A deep learning framework distilled by GBDT for online prediction tasks[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 384-394.

[13] Wold S, Esbensen K, Geladi P. Principal component analysis[J]. Chemometrics and intelligent laboratory systems, 1987, 2(1-3): 37-52.

[14] Balakrishnama S, Ganapathiraju A. Linear discriminant analysis-a brief tutorial[C]//Institute for Signal and information Processing. 1998, 18(1998): 1-8.

[15] Krzysztof Grabczewski and Norbert Jankowski. 2005. Feature selection with decision tree criterion. In null. IEEE, 212–217.

[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1–9.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).

[18] Steffen Rendle. 2010. Factorization machines. In Data Mining (ICDM), 2010 IEEE 10th International Conference on. IEEE, 995–1000.

[19] Yani Ioannou, Duncan Robertson, Darko Zikic, Peter Kontschieder, Jamie Shotton, Matthew Brown, and Antonio Criminisi. 2016. Decision forests, convolutional networks and the models in-between. arXiv preprint arXiv:1603.01250 (2016).

[20] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.