XGBoost Model and Its Application to Personal Credit Evaluation

Hua Li
Zhengzhou University
Henan Academy of Big Data
Henan Key Laboratory of Financial Engineering

Yumeng Cao
Zhengzhou University

Siwen Li
Zhengzhou University

Jianbin Zhao
Zhengzhou University

Yutong Sun
Macau University of Science and Technology

***Abstract*—This article investigates the application of the eXtreme Gradient Boosting (XGB) method to the credit evaluation problem based on big data. We first study the theoretical modeling of the credit classification problem using XGB algorithm, and then we apply the XGB model to the personal loan scenario based on the open data set from Lending Club Platform in USA. The empirical study shows that the XGB model has obvious advantages in both feature selection and classification performance compared to the logistic regression and the other three tree-based models.**

In this article, the eXtreme Gradient Boosting (XGB) method is introduced to establish a credit evaluation model based on big data, which is applied to personal credit evaluation, and the performance of this model is evaluated by different measures.

Digital Object Identifier 10.1109/MIS.2020.2972533
Date of publication 10 February 2020; date of current version 6 July 2020.

Credit evaluation models can be divided into three types—expert system, statistical models, and artificial intelligence (AI) methods. The traditional methods depend on a large number of structured historical data, which cannot be covered by the vast majority of individuals, small and micro enterprises and other credit subjects. The credit evaluation method based on big data refers to that by analyzing and mining massive, diversified, and dynamic data, and then using machine learning algorithm to design the credit

1541-1672 © 2020 IEEE

evaluation model, to depict the “portrait” of the credit subject in multidimensions and present the default rate and credit status of the credit subjects to information users. Therefore, due to its high efficiency, high performance and excellent processing ability based on big data samples, many researchers have applied various AI algorithms to the field of credit risk prediction.

The most commonly used algorithm based on big data for credit evaluation in banks is logistic regression (LR) because of its simple structure, high interpretability, and significant accuracy.[1] Although many advanced AI models, such as deep learning technology, have shown remarkable accuracy for credit prediction, its lack of interpretability and its poor performance in processing relatively smaller data sets prevent it from being widely used in credit evaluation system since most real-world scenarios of loans could not provide enough massive data for model training.[2] Herein, machine learning algorithms represented by decision tree (DT), random forest (RF), gradient boosting decision tree (GBDT), and XGB that have a superior predictive performance on smaller data sets are likely to be widely used in banks in the future. Machine learning algorithms have good performance in prediction of classification and regression problems and it can get better predictions in relatively short training time, among which, DT is a representative algorithm to predict individuals’ credit because of its significant performance in efficiency, accuracy, and interpretability.[3] However, a single machine learning method often leads to overfitting, and it is difficult to deal with a large number of unbalanced data sets emerging in actual problems. In order to make up for the disadvantages of a single machine learning method, the ensemble learning technology emerged and gradually become the mainstream method in the field of machine learning research.

Ensemble learning is to combine multiple models to make the ensemble model have a better ability of generalization. The theory of ensemble learning originated from the equivalence principle of strong learning and weak learning proposed by Kearns and Valiant,[4] which means in order to get an excellent stronger learning model, we could combine several simple weak learning models to “improve.” Considering that the DT algorithm has better performance among all kinds of machine learning algorithms and its training time is relatively shorter, the method combining ensemble learning and DT was proposed, and RF, GBDT, and XGB are typical ensemble algorithms based on the combination of ensemble learning and DT. Breiman[5] first proposed the RF algorithm by combining the ensemble method, DT algorithm and random subspace method. Friedman[6] proposed the GBDT algorithm to solve regression and classification problems. Subsequently, Chen and Guestrin[7] proposed an improved GBDT algorithm: XGB algorithm. XGB is a highly efficient boosting ensemble learning model originated in the DT model, which uses the tree classifier for better results of prediction and higher operation efficiency, and it not only has achieved excellent results in the competitions of Kaggle, a data competition platform, but also has been widely used in many fields, such as bank bankruptcy prediction, financial trading, network intrusion detection, and so on. In recent years, Nguyen[8] compared LR, DT, neural network with XGB, and verified the superior performance in credit evaluation of XGB via the confusion matrix and Monte Carlo simulation benchmarks. Li *et al.*[9] proposed an ensemble model by using the stacking method set the framework to model XGB, support vector machine and RF, and indicated that the proposed ensemble model has better prediction in default risking. Several recent researches show that ensemble learning algorithms have the ability to help banks effectively avoid risks.

Based on the analysis above, this article first reorganized the XGB model for classification problems and gives the theoretical basis for using the XGB model to select features. Then, this article used the latest 2018 personal credit loan data (loan approved in 2019 still have no explicit relevant repayment status) open provided to the world by Lending Club platform[10] to build models, and then improved the feature selection method by using nonlinear method based on XGB algorithm instead of traditional linear method based on IV values. The empirical results indicate that the feature engineering method based on the XGB model has a significant improvement effect on all five models. Subsequently, the credit evaluation effects of five models including XGB, LR, DT, GBDT, and RF are

Table 1. Symbol and meaning.

Symbol	Meaning
n	Total number of samples
m	Total number of features
x_i	Features information of the ith sample, $x_i \in R^m$
y_i	The actual label (or value) of the ith sample
$\hat{y}_i$	The predicted label (or value) of the ith sample
$\hat{y}_i^{(t)}$	The predicted value up to the tth tree
$l(y_i, \hat{y}_i)$	The loss function of the ith sample
$L(y, \hat{y})$	The loss function of total sample
$\Omega(f_k)$	Regular term of objective function to prevent overfitting, where f_k represents the kth DT.

compared and analyzed, which show the optimal prediction ability of the XGB model in the context of binary personal loan credit evaluation.

XGB MODEL

The basic idea of ensemble learning is to combine a series of weak learning models into a strong learning model to improve the performance of machine learning, which provides better prediction results than single models. This article mainly focuses on XGB, an ensemble algorithm of boosting for classification DT.

Model Description

The symbols and their meanings are shown in Table 1.

Given a data set containing n samples and m features, where $D = \{(x_i, y_i)|x_i \in R^m, y_i \in R\}$ and $x_i = \{x_{i1}, x_{i2}, \ldots, x_{im}|i = 1, 2, \ldots, n\}$. The chief task of the XGB model is to build t trees so that the predicted value $\hat{y}_i^{(t)}$ up to the tth tree satisfies formula given as

$$\begin{aligned}
&\hat{y}_i^{(0)} = 0\\
&\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)\\
&\hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i)\\
&\quad\ldots\\
&\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i).
\end{aligned} \tag{1}$$

In each iteration of the gradient boosting algorithm, a weak classifier $f_k(x_i)$ (i.e., a DT) is generated, and the predicted value $\hat{y}_i^{(t)}$ of this iteration is the sum of the predicted value of the previous iteration $\hat{y}_i^{(t-1)}$ and the DT result of this round $f_t(x_i)$.

Therefore, there are three key problems that have to be solved to build the XGB model. 1) How to establish the DT in each round of iteration, or how do leaf nodes split? 2) How to determine the predicted value of leaf nodes on each DT? 3) How does each DT relate to the previous one? These three problems above are determined by the objective function as follows. The objective function $L^{(t)}$ of this algorithm can be expressed as

$$\min L^{(t)}(y, \hat{y}^{(t)}) = \min\left(\sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k)\right) \tag{2}$$

where $l(y_i, \hat{y}_i^{(t)})$ can be loss functions of different types according to actual problems, and is usually used to measure the degree of inconsistency between the real value y_i and the predicted value $\hat{y}_i^{(t)}$. $\sum_{k=1}^{t} \Omega(f_k)$ is the regularization term (i.e., the penalty term) of the model, which is used to measure the complexity of the whole model, and it can be determined as follows:

$$\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda \sum_{j=1}^{T_k} \omega_{kj}^2 \tag{3}$$

where T_k is the number of leaf nodes in the kth tree, γ is the contraction coefficient of the number of leaf nodes T, ω_{kj} is the score of the jth leaf node in the kth tree, λ is the penalty coefficient of the score of leaf node ω, and the value of $\Omega(f_k)$ can be optimized through cross validation.

According to formula (1), substitute the predicted value $\hat{y}_i^{(t)}$ of the ith sample in the t round iteration into the objective function of formula (2), and then using second-order approximation of Taylor expansion at $\hat{y}_i^{(t-1)}$, the following equation can be deduce as (referring to Chen's derivation[1]):

$$\min L^{(t)} = \min\left(\sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t)\right) \tag{4}$$

where g_i and h_i are the first and second derivatives of the loss function $l(y_i, \hat{y}_i)$, respectively.

Define $I_j = \{i|q(x_i) = j\}$ as the collection of sample points on the jth leaf node in the DT, where the structure function $q(x)$ maps the sample point x to the position j of the leaf node and ω represents the score of the leaf node, so the result of the DT can be represented by $\omega_{q(x)}$. Consider that each DT $f(x)$ contains an independent tree structure $q(x)$ and the results $\omega_{q(x)}$ of this tree, which can be denoted as follows:

$$f(x) = \omega_{q(x)}, \quad \omega \in R^T, \quad q : R^d \rightarrow \{1, 2, \ldots, T\}. \tag{5}$$

Substitute formula (5) into formula (4), the following equation can be derived:

$$\min L^{(t)} = \min\left(\sum_{j=1}^{T_t}\left[G_j w_{t,j} + \frac{1}{2}\left(H_j + \lambda\right)\omega_{t,j}^2\right] + \gamma T_t\right) \tag{6}$$

where $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$.

Since this article takes the binary classification problem as an example, and the value of the samples' actual label y_i is 1 or 0, this article chooses the commonly used logloss function as loss function

$$l\left(y_i, \hat{y}_i^{(t)}\right) = -\left(y_i \log\left(p_i\right) + (1 - y_i)\log\left(1 - p_i\right)\right)$$
$$\text{where} \quad p_i = \frac{1}{1 + e^{-\hat{y}_i^{(t)}}}. \tag{7}$$

The derivation of logloss function is as follows. Given a bunch of samples $(x_1, y_1), \ldots, (x_n, y_n)$, and the value of label column y_i is 0 or 1 in the binary classification problem. The probability of n samples $Y_i = y_i (i = 1, 2, \ldots, n)$ can be obtained according to the probability formula of binomial distribution

$$P\{Y_I = y_1, Y_2 = y_2, \ldots, Y_n = y_n\} = \prod_{i=1}^{n} p_i^{y_i}\left(1 - p_i\right)^{1 - y_i}. \tag{8}$$

In order to maximize the probability P meeting this condition (maximum likelihood), we take the logarithm of formula (8)

$$\ln P = \sum_{i=1}^{n} y_i \ln p_i + \sum_{i=1}^{n} (1 - y_i)\ln(1 - p_i). \tag{9}$$

In this case, we need to satisfy the maximum likelihood function $\ln P$ maximum. If we set the loss function as $-\ln P$, it can be equivalent to the minimum value of the loss function. Then, the loss function can be expressed as formula (7), and the value of g_i and h_i can be deduced as

$$g_i = p_i - y_i \tag{10}$$

$$h_i = p_i(1 - p_i). \tag{11}$$

Similarly, for multiclassification problems, we can choose the Softmax function as the loss function

$$l(y, p) = -\sum_{m=1}^{M} y_m \log p_m, \ \text{where} \ \ p_m = \frac{e^{\hat{y}_i^{(t)}}}{\sum_{m=1}^{M} e^{\hat{y}_i^{(t)}}} \tag{12}$$

where m represents the category of labels and M represents a total of M categories. When $M = 2$, formula (12) degenerates into formula (7).

Key Problems

Determination of the Predicted Value of Leaf Node Assumed that our tree structure $q(x)$ is fixed and remains unchanged. According to formula (6), since $G_j \omega_{t,j} + \frac{1}{2}\left(H_j + \lambda\right)\omega_{t,j}^2$ is an upward parabola curve about the predicted value $\omega_{t,j}$ of leaf node (coefficient $\frac{1}{2}\left(H_j + \lambda\right) > 0$ is constant), by taking the derivative of the variable $\omega_{t,j}$, we could find the optimal value point $\omega_{t,j}^*$ to minimize the value of the objective function $L^{(t)}$

$$\omega_{t,j}^* = -\frac{G_j}{H_j + \lambda}. \tag{13}$$

In this case, the minimum value of the objective function is

$$\tilde{L}^{(t)} = -\frac{1}{2}\sum_{j=1}^{T_t}\frac{G_j^2}{H_j + \lambda} + \gamma T_t \tag{14}$$

where $\tilde{L}^{(t)}$ can be regarded as a scoring function to measure the quality of the tree structure $q(x)$ since the smaller the value of $\tilde{L}^{(t)}$, the better the structure of the tree. In other words, the optimal value can be obtained as long as the tree structure $q(x)$ can be determined.

Determination of the Splitting Mode of Leaf Node The tth tree is built based on the predicted value $\hat{y}_i^{(t-1)}$ and the actual labels y_i. First, selecting all or part of the features as candidate features. Then, using greedy algorithm to try to join a split to leaf nodes in each iteration, and optimal splitting point can be found by calculating the Gain score. The process would be completed until the stopping condition has been reached, and the whole construction process of a DT can also be named the determination of tree structure $q(x)$.

Supposed I_L and I_R represent the sample sets on the left and right nodes, respectively, after the splitting of a leaf node, $I = I_L \cup I_R$. For each feature, the scoring function value $\tilde{L}^{(t)}_{\text{No-Split}}$ of leaf nodes without splitting and the scoring function value $\tilde{L}^{(t)}_{\text{Split}}$ of leaf nodes after splitting in each iteration can be calculated as follows:

$$\begin{aligned}\tilde{L}^{(t)}_{\text{split}} &= -\frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda}\right] + \gamma T_{\text{Split}}\\ \tilde{L}^{(t)}_{\text{No-Split}} &= -\frac{1}{2}\frac{(G_L+G_R)^2}{H_L+H_R+\lambda} + \gamma T_{\text{No-Split}}\end{aligned} \tag{15}$$

where G represent the sum of the first derivatives after splitting, H represent the sum of the second derivatives after splitting, and L, R represent the left node and the right note, respectively.

Subtract $\tilde{L}^{(t)}_{\text{Split}}$ from $\tilde{L}^{(t)}_{\text{No-Split}}$ in formula (15), then the loss Gain of leaf node of the tth tree after splitting can be obtained

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma. \tag{16}$$

For each leaf node, the feature node with the largest Gain value is selected as the splitting point, and the generation of the tth tree would be completed until the stopping condition of splitting has been reached.

After the generation of each tree, add it to the original model to calculate the updated predicted value $\hat{y}_i$, and then continue to iterate according to the above rules. Finally, the construction of the whole model could be completed.

The Importance Value of Features Since XGB usually takes the Gain value in formula (16) as an indicator to evaluate the importance of features, which could be used to measure the ability to distinguish between good and bad samples of a certain feature, the features could be filtered according to the importance values.

Considering that in the splitting process of a leaf node, the feature with largest Gain value is always selected for splitting (which means this feature has the strongest distinguishing ability) and XGB is an additive model, the influence of multiple trees has to be considered at the same time. Therefore, we define the importance index of the rth feature variable by calculating the ratio of $\sum_{k=1}^{t} \text{Gain}_r^{(k)}$ (the sum of Gain values of the rth feature on all trees) to $\sum_{r=1}^{m}(\sum_{k=1}^{t} \text{Gain}_r^{(k)})$ (the sum of Gain values of all features on all trees)

$$\text{Imp}_r = \frac{\sum_{k=1}^{t} \text{Gain}_r^{(k)}}{\sum_{r=1}^{m}\left(\sum_{k=1}^{t} \text{Gain}_r^{(k)}\right)} \tag{17}$$

where $\text{Gain}_r^{(k)}$ represents the value of the rth feature variable in the kth iteration, t is the total number of iterations of the algorithm, and m is the total number of feature variables. The higher the importance index is, the stronger the ability of the feature variable to distinguish between good and bad samples is.

ALGORITHM

In this section, we will present the algorithm of XGB.

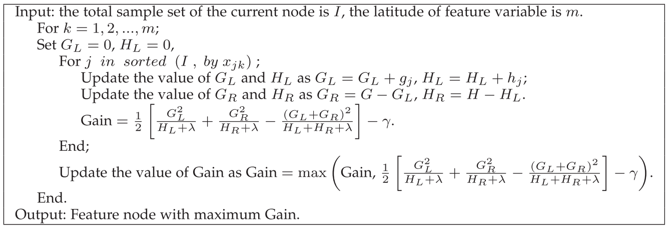
Step 1: Initialization

According to formula (10) and formula (11): $g_i = p_i - y_i, h_i = p_i(1-p_i)$.

$\hat{y}_i^{(t-1)}$ is the predicted value of sample x_i from the $(t-1)$th tree and y_i is the actual value of x_i. And the predicted value of the 0th tree is equal to 0, which means $\hat{y}_i^{(0)} = 0$.

Step 2: Determine the Splitting Mode

For the determination of the current root node, the Gain value of all or part of the features needs to be first traversed and calculated, to find the feature node with the maximum Gain score as the current root node. The pseudocode of the iteration process is shown in Figure 1.



```
Input: the total sample set of the current node is $I$, the latitude of feature variable is $m$.
    For $k = 1, 2, ..., m$;
    Set $G_L = 0$, $H_L = 0$,
        For $j$ $in$ $sorted$ $(I\ ,\ by\ x_{jk})$ ;
            Update the value of $G_L$ and $H_L$ as $G_L = G_L + g_j$, $H_L = H_L + h_j$;
            Update the value of $G_R$ and $H_R$ as $G_R = G - G_L$, $H_R = H - H_L$.
            Gain $= \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma$.
        End;
        Update the value of Gain as Gain $= \max\left(\text{Gain}, \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma\right)$.
    End.
Output: Feature node with maximum Gain.
```

Figure 1. Iteration process of Step 2.

Step 3: Establish the Current Binary Leaf Node Set

For the current root node, the sample set is divided into two parts according to the feature with maximum Gain found in the second step to obtain two leaf node sample sets. The second step above is repeated for the sets of two leaf nodes respectively until the Gain score is negative or meets other stopping conditions, and then the whole tree would be established.

Step 4: Calculate the Predicted Value of the Whole Leaf Node

According to formula (13), the predicted value of leaf node ω_j can be calculated as $\omega_j = -\frac{G_j}{H_j+\lambda}$, and the prediction results of the second tree can be expressed as $\hat{y}_i^{(2)} = \hat{y}_i^{(1)} + f_2(x_i)$ according to formula (1) and (6).

Then, the second tree has been established.

Step 5: Establish More Trees

Repeat *Step 1* to *Step 4* until enough number of trees have been established.

According to formula (1), the prediction results of the model $\hat{y}_i^{(t)}$ can be expressed as $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$, where $\hat{y}_i^{(t)}$ represents the prediction result of t trees on sample x_i.

Then, the tth tree has been established.

Step 6: Determine the Classification Result of Sample

Using $p_i = \frac{1}{1+e^{-\hat{y}_i^{(t)}}}$ in formula (7) to convert the final predicted value $\hat{y}_i^{(t)}$ of the sample into probability. When $p_i \geq 0.5$, the classification of sample x_i is 1 (means default), otherwise it is 0 (means no default).

To make the predicted value much closer to the real value in each round, each tree is built based on the prediction result of the previous tree, so as to improve the prediction effect of the model.

EMPIRICAL STUDY

We applied the XGB algorithm on a real-time loan data set and compared it with other four kinds of machine learning to verify the excellent performance of XGB algorithm in both feature selection and classification. We first conducted a comprehensive review of the raw data (see section "Data Description"). Then, we preprocessed the data set since real-time data sets are prone to interfere with classification results due to various quality problems, such as unbalanced data, null values, different data structures, etc. (see section "Data Preprocessing"). And then, feature selection was applied to the data (see section "Feature Selection"). The models were then trained, tested, and evaluated (see section "Empirical Results"). Here, due to the space limit, we omit features that are removed in each step of processing, referring to the Appendix, which is available in the IEEE Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/MIS.2020.2972533.

Data Description

This research is based on a public data set provided by Lending Club,[10] an online lending platform in the U.S., which is updated quarterly. The raw data set includes 127 702 samples of borrowers who applied for personal credit card loans on the website between January 1, 2018, and December 31, 2018, which was downloaded on August 5, 2019, and contains 143 feature variables and 1 label column (loan status).

Considering the seven kinds of loan status contained in the label column, and banks often use the term "good customer" to refer to customers with low default rates and the "bad customer" to refer to who have high default rates, we divided Fully Paid, Current, and In Grace Period into "Good" samples, and divided Default, Late (16–30 days), Late (31–120 days), and Charged Off into "Bad" samples, taking 0 and 1 corresponding to "Good" and "Bad," respectively. The distribution of the data set is shown in Table 2.

Data Preprocessing

Unbalanced Data Adjusting Considering the proportion of Good and Bad samples in the total data set is about 22:1, which is an extremely

Table 2. Distribution of raw data set.

Category	The number of samples	Proportion
Good	122 112	95.62%
Bad	5590	4.38%
Total	127 702	100%

Table 3. Distribution of original data set.

Category	The Number of Samples	Proportion
Good	22 360	80%
Bad	5590	20%
Total	27 950	100%

unbalanced data set, this article adopted the method of stratified random sampling to adjust the distribution of Good and Bad samples in the data set to 4:1. First, the total data set is divided into 12 layers by month. Second, for each layer, extract all the Bad samples from this layer, and then randomly extract part of good samples in this layer to ensure that the number of Good samples is four times of Bad samples.

After stratified random sampling, 27 950 samples remained, including 22 360 "Good" samples and 5590 "Bad" samples, the distribution of Good and Bad samples in the data set is finally adjusted to 4:1, which is shown in Table 3.

Manual Screening According to practical experience, 29 features that did not satisfy the criteria for entering the models were manually removed, which is shown in the Appendix C available online, including some information of borrowers such as address, date, zip code that have no significant impact on the prediction results of the label column, as well as the loan repayment information which new borrowers have no data with it.

Finally, 114 features were selected.

Null Value Dealing Considering that a certain feature is no longer representative when it has a large number of null values, so 35 features whose missing proportion over 50% were deleted directly, which is shown in Appendix C available online.

Herein, 79 features were selected.

Standardization and Null Value Filling The text data has been first converted to numeric data that can be recognized by the computer, then the null values of continuous data columns are filled with the average value of all non-null values in this column, and the null values of discrete data column are all filled with 0.

After the above steps, 79 features were selected.

Feature Selection

To verify the effectiveness of features selected by XGB, we first use a linear method of IV to select features, and then select features into the model according to a nonlinear method that using the feature importance index returned by XGB.

Feature Selection Based on IV

1) The prediction ability analysis of IV.

IV reflects the predictive ability of features, and the higher the IV is, the stronger the predictive ability is. The IV of 79 features were calculated and then 52 features whose IV less than 0.02 were deleted, which is shown in Appendix C available online.

Herein, 27 features were selected.

2) Correlation analysis between features.

Pearson correlation graph was used to find the correlation coefficient between any two features. To improve model efficiency and reduce data redundancy, for two features with the correlation coefficient greater than 0.6, only the features with higher IV could be retained. Then, 10 features were deleted, which is shown in Appendix C available online.

Finally, 17 features were selected.

3) Correlation analysis of features and label column.

In order to exclude the influence of features highly correlated with the label column on the predicted results, features with a correlation greater than 0.95 to the label column will be deleted. After calculation, the correlation coefficient between all 17 feature columns and the label column is lower than 0.91, meaning that 0 features will be removed in this step.

After the above three steps, a total of 17 features were selected, whose significance is shown in Appendix A, available online, meaning that the final data set after processing will only contain these 17 features. And, the selection result is reasonable since features contained in Appendix A, available online, have fully considered the fixed assets, the repayment ability and the credit situation of the borrower.

Feature Selection Based on XGB The data set with 79 features is used to enter into the XGB model as a training set and the function embedded in the XGB library in Python is used to return the feature importance index table ranked from high to low. Then, 17 features with the highest importance indicator (higher than 0.01) were selected, whose financial significance is shown in Appendix B available online and the selection result is reasonable. Compared with the features selected by the previous method, there are nine different features.

After the above steps, 17 features were selected.

Empirical Results

In this section, the prediction results of XGB are compared with LR, DT, RF, and GBDT to evaluate the performance of models.

Model Training

1) Divide training set and testing set.

The original data set was randomly divided into five parts on the premise that the proportion of Good and Bad samples remains unchanged in each part, and the prediction effect of the model will be verified by five-fold cross validation, which means four parts will be randomly selected as the training set and the remaining one as the testing set.

Specifically, the original data will be randomly split into five sets $s1, s2, \ldots, s5$, so that the size and distribution of the five sets are equal. And then, one of the five sets will be used as the testing set and the rest of four sets will be used as the training sets, which means, for example, $s1$ will be first considered as the testing set and $s2, s3, s4, s5$ will be the training sets. Keep repeating this process until each set was once served as the testing set. Finally, the mean value of these five testing sets will be used as the final prediction results of models.

2) Model training and prediction.

The GridSearch function in Python is used to return the optimal parameters of each model. The prediction results of XGB corresponding to the two methods of feature selection are shown in Table 4, indicating that the XGB model based on XGB feature selection is significantly better at distinguishing between Good and Bad samples.

Table 4. Confusion matrix of XGB.

Confusion matrix of XGB based on IV feature selection		
	Prediction: Good	Prediction: Bad
Actual: Good	4437	85
Actual: Bad	328	740
Confusion matrix of XGB based on XGB feature selection		
	Prediction: Good	Prediction: Bad
Actual: Good	4471	51
Actual: Bad	301	767

Model Evaluation and Comparison

1) Select evaluation indicators.

The most common evaluation indicator in the classification problem is Accuracy, and Kappa is considered together since it is usually used to measure the classification accuracy of unbalanced data set, and the higher the Kappa is, the higher the classification accuracy is.

The ROC and AUC are also selected. The ROC curve is often used to measure the prediction performance of models, and AUC (area under the ROC curve) was used to evaluate the performance of a binary classification system. The value range of AUC is [0,1], and the larger the AUC is (which also means the closer the ROC curve gets to the top left corner), the better the classification effect of the model is.

Besides, the KS that is often used to describe the ability of models to distinguish between two samples is selected. KS is the corresponding value where the two lines are furthest apart in the KS curve, whose range is [0,1], and the larger the KS is (which also means the greater the distance between two lines in the KS curve), the greater the model's ability to distinguish Good and Bad samples.

2) Output evaluation indicators.

Using Python, we can quickly get results (all taken as the mean value by the method of five-fold cross validation and four decimal places are reserved).

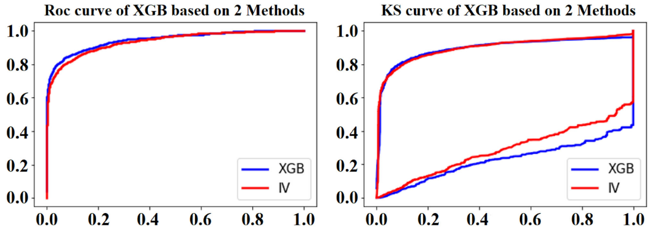


Figure 2. Figure ROC and KS curve of XGB based on two methods of feature selection.

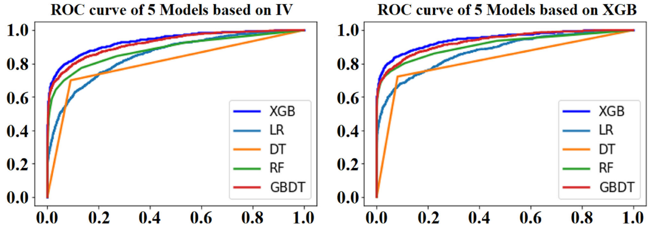


Figure 3. ROC curve of five models based on two methods of feature selection.

The performance of five models based on IV feature selection and XGB feature selection are respectively shown in Tables 5 and 6. The ROC and KS curve of XGB based on two methods of feature selection are shown in Figure 2 and the ROC curve of five models based on two methods are shown in Figure 3.

3) Analysis of prediction results.

According to Tables 5 and 6, Figures 2 and 3, selecting features by XGB can effectively improve the classification effect of various models, which is manifested by different degrees of improvement of evaluation indicators of all five models.

Moreover, both in Tables 5 and 6, XGB has the highest value of Accuracy, Kappa, AUC, and KS, indicating that the performance of XGB model is significantly better than that of other four models.

Table 5. Performance of five models based on IV feature selection.

	Accuracy	Kappa	AUC	KS
XGB	**0.9261**	**0.7382**	**0.9383**	**0.7332**
LR	0.8644	0.4690	0.8498	0.5390
DT	0.8665	0.5849	0.8042	0.6083
RFs	0.9095	0.6644	0.8823	0.6370
GBDT	0.9204	0.7074	0.9226	0.6970

Table 6. Performance of five models based on XGB feature selection.

	Accuracy	Kappa	AUC	KS
XGB	**0.9370**	**0.7763**	**0.9481**	**0.7700**
LR	0.8921	0.5956	0.8730	0.5878
DT	0.8801	0.6222	0.8194	0.6387
RFs	0.9249	0.7264	0.9077	0.6995
GBDT	0.9279	0.7370	0.9343	0.7203

CONCLUSION

In this article, we first studied the theoretical modeling of the credit classification problem using XGB algorithm, and then we applied XGB to the personal loan scenario based on the open data set from Lending Club Platform. Compared with the performance of LR, DT, RFs, and GBDT, the empirical study verified the obvious advantages of XGB in feature selection and classification performance. In future research, the possible interesting problem is to setup a credit evaluation model using XGB based on multiclassification problems or regression problems.

ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China under Grant 11501523.

REFERENCES

1. B. W. Chi and C. C. Hsu, “A hybrid approach to integrate genetic algorithm into dual scoring model in enhancing the performance of credit scoring model,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2650–2661, 2012.
2. D. West, “Neural network credit scoring models,” *Comput. Oper. Res.*, vol. 27, no. 11/12, pp. 1131–1152, 2000.
3. A. Shen, R. Tong, and Y. Deng, “Application of classification models on credit card fraud detection,” in *Proc. Int. Conf. Service Syst. Service Manage.*, 2007, pp. 1–4.
4. M. Kearns and L. G. Valiant, “Cryptographic limitations on learning Boolean formulae and finite automata,” *J. ACM*, 1989, pp. 433–444.

5. L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
6. J. H. Friedman, “Greedy Function approximation: A gradient boosting machine,” *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.
7. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, 785–794.
8. C. Nguyen, “The credit risk evaluation models: An application of data mining techniques,” in *Proc. SAIS*, 2019, Paper 36.
9. G. Li, Y. Shi, and Z. Zhang, “P2P default risk prediction based on XGBoost, SVM and RF fusion model,” in *Proc. 1st Int. Conf. Bus., Econ., Manage. Sci.*, 2019, pp. 470–475.
10. 2018. [Online]. Available:https://www.lendingclub.com/

Hua Li is currently a Professor with the School of Mathematics and Statistics, Zhengzhou University, Zhengzhou, China. She is the Head of Henan Key Laboratory of Financial Engineering. She was the recipient of the Outstanding Natural Science Paper Award of Henan province (2013). Her research interests focus on machine learning and financial engineering. She is the corresponding author of this article. Contact her at huali08@zzu.edu.cn.

Yumeng Cao is currently working toward a master’s degree with the School of Mathematics and Statistics, Zhengzhou University, Zhengzhou, China. Her research interests focus on machine learning and financial engineering. She received the B.S. degree in 2018 from Zhengzhou University. Contact her at 296111624@qq.com.

Siwen Li is currently working toward a master’s degree with the School of Mathematics and Statistics, Zhengzhou University, Zhengzhou, China. Her research interests focus on machine learning and financial engineering. She received the B.S. degree in 2018 from Zhengzhou University. Contact her at 1029892591@qq.com.

Jianbin Zhao is currently working toward the Ph.D. degree with the School of Mathematics, Tongji University, Shanghai, China. He is currently a Lecturer with the School of Mathematics and Statistics, Zhengzhou University. His research interests focus on statistics, machine learning, and financial engineering. He is the corresponding author of this article. Contact him at zhaojianbin@zzu.edu.cn.

Yutong Sun is currently working toward an undergraduate degree with the School of Business, Macau University of Science and Technology, Macau. Her research interests focus on machine learning and financial engineering. Contact her at sun_yutong@163.com.