Application of XGBoost in P2P Default Prediction

Zhiqiang Li1,*, Shouyan Li1, Zhilong Li1, Yixiang Hu1, Hanlin Gao1

1 College of Business Jiangxi University of Science and Technology Nanchang, Jiangxi
*lzq200786608@gmail.com

Abstract. P2P network lending is a "peer-to-peer" loan through a third-party Internet platform built by P2P companies. The application of machine learning algorithms to the field of P2P loan default prediction will improve the operating capabilities of the platform, and also effectively regulate the lending market. In this paper, we use Lending Club's loan data and feature engineering technology to apply the XGBoost algorithm to construct a P2P loan default prediction model, and we choose five performances: accuracy, AUC value, error rate, model robustness, and model run time to compare it with Logistic Regression and Decision Trees. The result shows that the prediction accuracy rate of the XGBoost algorithm is 97.705%, which fits the actual results better, and can effectively control the loss cost caused by model errors. In addition, we also select 10 features that have the greatest impact on loan default rates based on the XGBoost algorithm, and provide a reference for P2P lending platforms.

1. Introduction

P2P network lending (referred to as P2P network loan, peer-to-peer lending) is a direct personal-to-person credit model. It breaks away from using traditional banks as a medium to form a loan contract between the fund supplier and the demander in an unsecured form on the Internet platform.

At present, scholars mainly conduct research from three aspects: P2P lending platforms, borrowers’ default factors, and building platform prediction models. This paper mainly focuses on the construction of the default prediction model. In recent years, many scholars have tried to adopt different methods to predict the loan default rate of users. Li Liao et al. believe that the identification of borrowers' public information can judge different default risks [1]. Srrano-Cinca et al. (2015) used survival analysis to screen the variables into the model, and then built a loan default prediction model based on Logistic Regression [2]. Milad Malckipirbazari (2015) based on the real data of the lending club, by comparing the Random Forests algorithm, FICO score and LC rank identification, it is proved that the Random Forests algorithm is significantly better than the other two methods [3]. PetrTeply and MichalPolena (2019) selected Lending Club loan data from 2009 to 2013, and selected a variety of different classifiers, such as Logistic Regression, Artificial Neural Network, Linear Discriminant Analysis, k-Nearest Neighbor, classification and regression decision trees, through comparison, it is found that Logistic Regression, Artificial Neural Network, and Linear Discriminant Analysis are the three best performing algorithms [4]. Vedala and Kumar (2012) constructed a Bayesian model of multiple events for loan default prediction in empirical research [5]. Ning Zhang and Qin Chen (2018) used the information retrieval TF-IDF algorithm to construct an investor's reverse investment scale factor, quantitatively measured investor differences, optimized the investment human rights recalculation factor in the model to predict, and broadened the method of loan default prediction [6].

Tianqi Chen and Carlos Guestrin (2016) used the data of Lending Clubs to conduct empirical analysis.
By comparing with the GBDT algorithm, they explained the principle of the XGBoost algorithm, indicating that the XGBoost algorithm is an efficient and widely used machine learning algorithm [7]. This paper will also use the XGBoost algorithm to build the P2P loan default model and compare it with Logistic Regression and Decision Tree in order to have a deeper understanding of the XGBoost algorithm and expand its scope of use. At the same time, we hope to contribute to reducing the number of P2P loan defaults through the study of the model.

2. Data processing based on Feature Engineering

2.1 Data collection

The experiment data used in this paper comes from the official website of Lending Club [8] (https://www.lendingclub.com/), which includes the loan data released by Lending Club from 2007 to 2018, with a total of 2,260,699 lines of observation data and 151 variables.

2.2 Data cleaning

The data used in this paper is directly downloaded from the official website of Lending Club. The original data has problems such as missing values and label leakage, so data cleaning is required first. The original data contains some post-loan information, such as out_prncp (the total remaining outstanding principal), last_pymnt_amnt (repayment received last month), etc., need to be deleted. The missing value of the original data is serious, and the proportion of missing values of 38 variables is more than 90%. In this paper, the threshold is set to 50%, that is, the proportion of missing values exceeding 50% needs to be deleted, such as settlement_amount (the loan amount the borrower has agreed to repay), desc (the loan description provided by the lender) and other variables. Some variables in the original data have single characteristic, this paper deletes these variables through homogenization. Too many features of data variables will overfit the model with strong learning ability. For example, emp-title (job title of the borrower) has more than 500,000 different values and needs to be deleted. Finally, this paper deleted 67 useless variables, leaving 84 variables.

This paper mainly uses special value filling and statistical value filling for missing value filling. For the variable emp-length (the borrower's working life), the missing value will be filled with 0, which means that the working life is less than 1 year. The remaining non-continuous variables will use the last value of the missing value to fill in the missing value. And continuous variables will be filled with the mode, thus making full use of the data set without losing key data information.

2.3 Data exploratory analysis

After data cleaning, this paper analyzes the data of the Lending Club platform as a whole to explore data characteristics and data dimensions, thereby effectively improving data quality, and improving algorithm operation speed and accuracy. Figure 1 shows the distribution of borrowers’ working years. It can be seen that most browsers have worked for 10 years or more, accounting for 35.41%, while other working years are basically around 4%~8%, indicating that Lending Club is for borrowers There are very high requirements for working life, and it is difficult to apply for a loan if it does not reach a certain working life.
Table 1 shows the loan status information of the original data. In this paper, Fully Paid and Current are defined as good loans (0), and the remaining statuses are defined as bad loans (1). The label still presents an imbalance after division, and subsequent model training needs to take this into consideration. Figure 2 shows the proportion of good loans and bad loans, where good loans accounted for 84.48% and bad loans accounted for 13.52%. From the data results, Lending Club, as the largest P2P platform in the United States, still has a loan default rate of 13.52%.

Table 1. Original data loan status information

| Loan status                        | Count  |
|------------------------------------|--------|
| Fully Paid                         | 1076763|
| Current                            | 878333 |
| Charged Off                        | 268561 |
| Late (31-120 days)                 | 21467  |
| In Grace Period                    | 8436   |
| Late (16-30 days)                  | 4349   |
| Does not meet the credit policy. Status: Fully Paid | 1989 |
| Does not meet the credit policy. Status: Charged Off | 761 |
| Default                            | 40     |

Figure 1. Distribution of user working years

Figure 2. The proportion of good loans and bad loans
Figure 3 shows the defaults for different loan purposes. It can be seen that education and small business-related loans have a higher default rate, while car-related loans have the lowest default rate.

Figure 4 shows the situation of loans divided by loan grades, and the loan default rate increases significantly as the loan grade (A-G) declines. However, the lower the loan grade, the lower the proportion, so the overall default rate is between B, C, and D, which is 16.18%.

![Distribution of loan purposes](image)

**Figure 3. Distribution of loan purposes**

![Loan status by loan grade](image)

**Figure 4. Loan status by loan grade**

In order to be clearer about the structure of the data and to facilitate the subsequent analysis of this article, Table 2 shows the description of some data.

| Name               | Describe                                      |
|--------------------|-----------------------------------------------|
| loan_amnt          | the listed amount of the loan applied for by the borrower. |
| funded_amnt        | loan amount at loan time                       |
| total_rec_late_fee | late fees received to data                     |
| int_rate           | lending rates                                  |
| emp_title          | job title                                      |
Feature construction and feature selection

This paper deals with variables mainly in dimensionless and feature abstraction. Firstly, using the feature standardization method, the feature is scaled by calling the StandardScaler method of the preprocessing sub-module of the scikit-learn module, and the feature value is converted into a standard normal distribution value. Secondly, perform numerical mapping for non-continuous variables, such as mapping the grade (loan grade) variable from A-G to 1-7. Finally, choose the variables that have practical significance and influence on the model as derived variables from the original variables. For example, installment _feat (percentage of monthly repayment to monthly income) is obtained through installment (monthly repayment amount)/annual_inc (monthly income).

Feature selection reduces the dimensionality of variables by removing irrelevant features, which can make the model's generalization ability better, greatly reduce the risk of overfitting, and also enhance the interpretability of features and feature values. This article uses Wrapper method and statistical feature selection method to remove irrelevant and redundant variables according to Pearson Correlation Coefficient. Finally, there are 40 features left in the data.

3. XGBoost model introduction

XGBoost [9] is an improvement of the gradient boosting algorithm and a decision tree based on the gradient boosting algorithm. Through a large number of iterations, each iteration produces a weak classifier, and each weak classifier is trained on the bias of the result of the previous classifier.

\[ \hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i) \]  

(1)

Where \( f_k \) is the regression tree, \( K \) is the number of regression trees, \( f_k(x_i) \) is the score of the i-th observation given by the k-th tree.

In order to make the prediction results more accurate, a penalty term is added to the prediction function, which will reduce the occurrence of overfitting and increase the generalization ability of the model function. The model objective function will become as follows:

\[ Obj(\phi) = \sum_i^l l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \]

\[ \Omega(f_k) = \gamma T + \frac{1}{2} \lambda \| \omega \|^2 \]

(2)

\( y_i \) is the true value of the training sample, \( \hat{y}_i \) is the predicted value of the training sample. \( l \) is a differentiable convex loss function that measures the difference between the prediction \( \hat{y}_i \) and the target \( y_i \). \( \Omega \) is a penalty term to prevent the model from being too complicated. \( \gamma \) is a parameter that controls the number of \( T \) nodes, \( \lambda \) is the parameter that controls the weight of the leaf node.
However, the seemingly perfect equation (2) is difficult to optimize the Euclidean Spaces through traditional methods. Therefore, XGBoost adopts the greedy idea and adds $f_i$ into the objective function to improve the performance of the model. The objective function becomes the following form:

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_i(x_i)\right) + \Omega(f_i)$$  \hspace{1cm} (3)$$

$\hat{y}_i^{(t)}$ is the predicted value of the $i$-th instance in the $t$-th iteration. In general, the gradient of the objective function is difficult to obtain. Because second-order approximation can be rapidly optimized [10], we use the second-order Taylor expansion to simplify the equation.

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i)\right) + \Omega(f_i)$$  \hspace{1cm} (4)$$

where $g_i = \frac{\partial}{\partial y_i} l\left(y_i, \hat{y}_i^{(t-1)}\right)$ and $h_i = \frac{\partial^2}{\partial y_i^2} l\left(y_i, \hat{y}_i^{(t-1)}\right)$ are first and second order gradient statistics on the loss function.

Define $I_j = \{i | q(x_i) = j\}$ as the set of leaf nodes $j$, by extending the $\Omega$ and remove the constant term at $t$-th iteration, we can rewrite the equation (5)

$$\widetilde{Obj}^{(t)} = \sum_{i=1}^{n} \left[ g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i)\right] + \gamma T + \frac{1}{2} \sum_{j=1}^{T} \omega_j^2$$

$$= \sum_{j=1}^{T} \left[ \left(\sum_{i \in I_j} g_i\right) \omega_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda\right) \omega_j^2\right] + \gamma T$$  \hspace{1cm} (5)$$

In order to obtain a stable tree structure, the weight of leaf node $j$ $\omega_j$ is expressed as $\omega_j = \frac{\sum g_i}{\sum h_i + \lambda}$. And then we can calculate the corresponding optimal value:

$$\widetilde{Obj}^{(q)} = -\frac{1}{2} \sum_{j=1}^{T} \left(\sum_{i \in I_j} g_i\right)^2 \sum_{i \in I_j} h_i + \lambda + \gamma T$$  \hspace{1cm} (6)$$

Equation (6) is the evaluation equation of tree structure $q$. However, it is not practical to list all the tree structures in general. Instead, we use a greedy algorithm to continuously divide leaf nodes and iteratively add subtrees. The following equation is used to evaluate the split node:

$$Obj_{split} = \frac{1}{2} \left[ \left(\sum_{i \in I_l} g_i\right)^2 + \left(\sum_{i \in I_r} g_i\right)^2 - \left(\sum_{i \in I_l} h_i + \lambda\right) \left(\sum_{i \in I_r} h_i + \lambda\right) \right] - \gamma$$  \hspace{1cm} (7)$$

Where $I_L$ and $I_R$ are the left and right nodes of the instance set after leaf node $I$ is split, respectively.

4. XGBoost model establishment

This paper uses Python tools and the xgboost standard library to perform a two-class prediction analysis on Lending Club loan data. First, we use cross-validation and grid search to calculate the optimal parameters of the model (shown in Table 3), so that the model obtains the optimal classification accuracy parameters, and then the two-class model training is carried out based on the XGBoost algorithm.

| Parameter            | the optimal parameters |
|----------------------|------------------------|
| n_estimators         | 200                    |
| min_child_weight     | 6                      |
The parameters used were:

| Parameter       | Value |
|-----------------|-------|
| max_depth       | 7     |
| gamma           | 0.5   |
| learning_rate   | 0.1   |
| subsample       | 0.6   |
| colsample_bytree| 0.9   |
| reg_alpha       | 2     |
| reg_lambda      | 0.05  |

Where `n_estimators` is the number of trees. `min_child_weight` defines the minimum weight sum for all observations of a subtree. `max_depth` is the depth of the tree. `gamma` specifies the minimum loss function decline in value. `learning_rate` is the learning rate. `subsample` controls the random proportion of each tree. `colsample_bytree` controls every tree of the characteristics of random sampling. `reg_alpha` is the weight of L1 regularization. `reg_lambda` is the weight for L2 regularization.

As shown in Figures 5-6, when the model is iterated 200 times, the error rate of the test set and the error rate of the training set reach the minimum at the same time. At this time, the error rate of the training set is 2.206%, and the error rate of the test set is 2.295%. The error rate and loss value of the model test set and training set show a simultaneous downward trend, indicating that the model has not overfitted.

**Figure 5. XGBoost log loss**

**Figure 6. XGBoost classification error**

Figure 7 shows the contribution of features to the model. The top 10 feature variables from high to low are last_fico_range_high (the last upper boundary of range the borrower’s FICO belongs to pulled.), fico_range_low (the lower boundary of range the borrower’s FICO belongs to), total_rec_late_fee (late fees received to data), installment_feat (percentage of monthly repayment to monthly income), mo_sin_old_rev_tl_op (months since oldest revolving account opened), grade (LC assigned loan grade), loan_amnt (the listed amount of the loan applied for by the borrower), max_bal_bc (max balance of bankcard accounts), dti (the borrower’s debt to income ratio), addr_state (the address state provided by the borrower during loan application).
Figure 7. Feature importance

It can be seen from Figure 7 that last_fico_range_high has a greater impact on the model. To prevent the risk of overfitting, we can delete it and check the model effect. After verification, the model has no over-fitting phenomenon.

5. Comprehensive model analysis
In order to analyze the difference between the performance of the default prediction model constructed by XGBoost and other algorithms, this paper selects Logistic Regression and Decision Trees to construct P2P default prediction models respectively. The classification effect of each model is compared from five aspects: accuracy, AUC value, error rate, model robustness, and model running speed. According to Table 4:

(1) The classification accuracy rate can reflect the default prediction ability of the model. From the perspective of test set classification accuracy, XGBoost has the highest accuracy, followed by Decision Trees, and finally Logistic Regression.

(2) The AUC value objectively judges the classification accuracy. The larger the AUC value, the better the model classification effect. Therefore, it can be judged from the AUC value that XGBoost has the highest accuracy, followed by Decision Trees, and finally Logistic Regression.

(3) The robustness of the model represents the change in accuracy, that is, the degree of similarity between the test set and the training set classification. Judging by the fluctuation of the AUC value, XGBoost has the best robustness, followed by Decision Trees, and finally Logistic Regression.

(4) Model error rates are divided into two types: the first type of error is the probability of predicting non-default as default, and the second type of error is the probability of predicting default as non-default. Because the loss cost of the second type of error in default is much greater than the loss cost of the first type of error, the second type of error rate is used to measure the loss cost of prefetching misclassification of each model. It can be judged from the second type of error rate value on the test set that XGBoost has the lowest cost of predicting misclassification, followed by Decision Trees, and finally Logistic Regression.

(5) It can be found from the running rate of the model that the Logistic Regression runs the fastest, followed by the Decision Trees, and finally XGBoost.

Table 4. Classification and prediction performance indicators of each model under 5-fold cross validation

| Model                  | XGBoost | Logistic Regression | Decision Tree |
|------------------------|---------|---------------------|---------------|
| the accuracy of the test set (%) | 97.705  | 92.12               | 95.88         |
| type 1 error rate of the test set (%) | 2.138   | 4.87                | 3.47          |
| type 2 error rate of the test set (%) | 0.157   | 3.01                | 0.65          |
6. Summary
This paper verifies the ability of XGBoost to predict the default of P2P loans, and selects the 10 most important feature variables that affect the prediction results. At the same time, this paper also compared with Logistic Regression and Decision Tree, and found that XGBoost is significantly better than traditional machine learning algorithms in most aspects. But it does not mean that the XGBoost algorithm has sufficient performance in P2P default prediction. In future research, we will conduct a comparative study on GBDT's cutting-edge algorithms LightGBM and Catboost and artificial neural network models in order to analyze XGBoost deeply and find the most suitable model algorithm for studying P2P loan default prediction.

References
[1] Li L, Mengran L, Zhengwei W. (2014) Research on Regional Discrimination in China's Internet Finance. The Journal of Quantitative & Technical Economics., 31(05):54–70.
[2] Carlos Serrano-Cinca, Begoña Gutiérrez-Nieto, Luz López-Palacios. (2015) Determinants of Default in P2P Lending[J]. PLOS ONE., 10(10).
[3] Milad Malekipirbazari, Vural Aksakalli. (2015) Risk assessment in social lending via Random Forests. Expert Systems With Applications., 42(10).
[4] Petr T, Michal P. (2020) Best classification algorithms in peer-to-peer lending. North American Journal of Economics and Finance., 51.
[5] Vedala R.Kumar B.R. (2012) An application of Naïve Bayes classification for credit scoring in e-lending platform. 2012 International Conference on Data Science & Engineering (ICDSE). pp. 81-84
[6] Ning Z, Qin C. (2018) P2P loan default prediction model based on TF-IDF algorithm. Journal of Computer Applications., 38(10):3042-3047
[7] Chen T, Guestrin C. (2016) XGBoost: A Scalable Tree Boosting System. the 22nd ACM SIGKDD International Conference., ACM:785-794.
[8] Nowak A, Ross A, Yencha C. (2015) Small Business Borrowing and Peer-to-Peer Lending: Evidence from Lending Club. Working Papers., 36(4):129-159.
[9] Chen T, Tong H and Benesty M. 2016 xgboost: Extreme Gradient Boosting.
[10] Jerome F, Trevor H, Robert T. (2000) Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). The Annals of Statistics.,28(2).