Intelligent Information Management, 2022, 14, 157-164
https://www.scirp.org/journal/iim
ISSN Online: 2160-5920
ISSN Print: 2160-5912

Machine Learning Approaches to Predict Loan Default

Wanjun Wu

Bytedance Data Analysis Group, Beijing, China
Email: wuwanjun.jennifer@bytedance.com

How to cite this paper: Wu, W.J. (2022) Machine Learning Approaches to Predict Loan Default. *Intelligent Information Management*, **14**, 157-164.
https://doi.org/10.4236/iim.2022.145011

Received: August 11, 2022
Accepted: September 25, 2022
Published: September 28, 2022

Abstract

Loan lending plays an important role in our everyday life and powerfully promotes the growth of consumption and the economy. Loan default has been unavoidable, which carries a great risk and may even end up in a financial crisis. Therefore, it is particularly important to identify whether a candidate is eligible for receiving a loan. In this paper, we apply Random Forest and XGBoost algorithms to train the prediction model and compare their performance in prediction accuracy. In the feature engineering part, we use the variance threshold method and Variance Inflation Factor method to filter out unimportant features, and then we input those selected features into Random Forest and XGBoost models. It turns out that Random Forest and XGBoost show little difference in the accuracy of their predictions since both get high accuracy of around 0.9 in the loan default cases.

Keywords

Machine Learning, Random Forest, Loan Default, Prediction Model

1. Introduction

Loan lending plays an important role in our everyday life and powerfully promotes the growth of consumption and the economy [1]. Taking a loan has been inevitable for people since individuals around the world depend on loans to overcome financial constraints to achieve their personal goals, and organizations rely on loans to expand their production [2]. In most cases, loan lending is beneficial to both the borrowers and the lenders. However, loan default is still unavoidable, which carries a great risk and may even end up in a financial crisis. Therefore, it is particularly important to identify whether a candidate is eligible for receiving a loan.

In the past, the evaluation primarily depended on manual review, which was

time-consuming and labor-intensive [3]. Recently, banks have opted for machine learning approaches to automatically predict the loan default since it can highly enhance the accuracy and the efficiency of the prediction. On the one hand, banks can collect a massive amount of transaction data due to the prosperity of online shopping and mobile payments. On the other hand, machine learning models are rapidly evolving and have successful applications in various fields, motivating the bank industry to use them to predict loan default. Researchers have found that Random Forest performed better than other models such as logistic regression, decision trees, and support vector machines in some loan lending cases [4]. We will also apply XGBoost to predict the loan default to make a comprehensive comparation since XGBoost is one of the most advanced methods for machine learning that has been developed in recent years [5].

In this paper, based on the loan default data provided by Imperial College London, we predict whether a loan will default, as well as the loss incurred if it does default. We choose Random Forest and XGBooost to build the prediction model and decide which one performs better.

The rest of this paper is organized as follows. Section 2 describes the characteristics of the raw data and shows the methodology of this paper, including feature engineering and introductions of Random Forest and XGBooost. In Section 3 we show the experiment process of applying models and evaluate the results. Finally, we draw our conclusion and discuss the potential applications of our outcomes in Section 4.

2. Literature Review

At present, researchers generally use machine learning methods to predict loan defaults, including Logistic Regression, Decision Trees, Random Forest, XGBoost, and other advanced techniques.

The main advantages of Logistic Regression lie in its simple understanding, sturdy performance, and easy implementation [6]. Logistic Regression naturally outperforms Linear Regression in predicting the probability of loan default since its outcome contains a continuous range of grades between 0 and 1, which represents the likelihood of an event occurring [7]. Han used Logistic Regression and Cox proportional hazard algorithm to predict student loan default, whose findings indicated that the main affected factors that led to student loan default lie in age, household income, monthly repayable amount, and the college major. The Logistic Regression model that they developed gained an AUC of 0.697 for the test data, which showed the accuracy and robustness of LR [8].

Decision Trees generates a structure like a tree by classifying the instances and using recursive portioning algorithm [7]. Each leaf node represents a class label and branches present the outcomes for the test, which are represented by internal nodes for an attribute [9]. In order to predicting the businesses' past due in service accounts, Wang developed models using Logistic Regression and Decision Trees in SAS and compared their results. It turned out that Decision Trees

outperformed Logistic Regression when there were small amount of attributes in a large enough sample [10].

Random Forest runs by constructing multiple decision trees while training and outputting the class that is the mode of the classes output by individual trees, which outperformed single decision trees [11]. Malekipirbazari and Aksakalli built a Random Forest based classification model to identify high-quality peer-to-peer borrowers. They compared the different machine learning techniques and found out that the Random Forest based model performed significantly better than the FICO credit scores [4].

XGBoost has been shown to achieve state-of-art results on many machine learning tasks. It is an improvement of Gradient Boosting algorithm and a decision tree based on the gradient boosting algorithm. Li constructed an XGBoost-based model to predict peer-to-peer loan default and compare its outcome with Logistic Regression and Decision Trees. The results indicated that the accuracy of the XGBoost-based model achieved 97.705%, which fitted the actual results better [12].

3. Experiment

The framework of our experiment process is shown in **Figure 1**.

3.1. Data Processing

First, we take a quick look at the whole data. The dataset provided by Imperial College London includes 105,471 records and 771 columns, containing customers' ids, 778 features, and the loss of the record. The detailed description is shown in **Table 1**.

Next, we clean the NAs in the dataset. There are 525 columns containing NAs. When dealing with the numeric columns, we fill those NAs with the average of the columns.

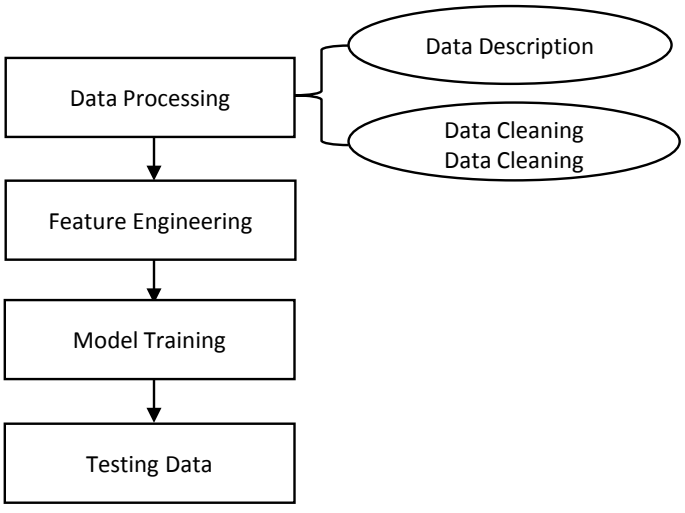

Figure 1. The framework of our experiment.

Table 1. The detailed description of the dataset, including the columns and the data types.

Columns	Description
id	The ids of customers. Float type.
F1, F2, …, F778	Different features of the customer behaviors. Numeric types.
Loss	The loss incurred. Float type. Note: If the record does default, the number shows the severity of the loss that result. If there is no default, the number is 0.

3.2. Feature Engineering

Feature selection aims to drop those redundant columns which contain little useful information and reduce the number of input features when developing an effective model.

First, we use the variance threshold method to quickly filter out those columns whose variance equals 0, since those columns do not contain useful information for classification. After applying the variance threshold method, we get 760 columns left.

Second, we apply Variance Inflation Factor method to reduce multicollinearity. Multicollinearity inflates unnecessarily the standard errors of the coefficients, and increased standard errors indicates that the coefficients of some features might be close to 0, which will make some features insignificant when they should be significant. A useful way to measure multicollinearity is Variance Inflation Factor (VIF). If no features are correlated, the VIF will be 1. If the VIF is greater than 10, it shows that the regression coefficients are poorly estimated due to multicollinearity [13]. In this paper, we remove all those features whose VIF is greater than 10.

After filtering features by the variance threshold method and the VIF, we have 419 columns left.

3.3. Model Training & Testing Data

In this paper, we apply Random Forest and XGBoost algorithms to train the model and compare their performance in prediction accuracy. Empirical studies show that the best results are obtained if we use 20% - 30% of the data for testing, and the remaining 70% - 80% of the data for training [14]. Thus, we separate the dataset randomly into 2 parts. The first part is the training dataset, which contains 80% data, and the remaining 20% of data belong to the testing dataset. We use the training dataset to train the model and use the testing dataset to evaluate the efficiency of the model.

3.3.1. Random Forest

Random Forest is a decision tree based supervised learning algorithm, which implements the classification by constructing multiple decision trees. The metric we choose for splitting attributes in decision trees is the Gini index. For a candi-

date split attribute X_i, denote possible levels as $L_1, L_2, \cdots, L_j$, the Gini Index is calculated as [15]:

$$G(X_i) = \sum_{j=1}^{J} \Pr(X_i = L_j)\left(1 - \Pr(X_i = L_j)\right) = 1 - \sum_{j=1}^{J} \Pr(X_i = L_j)^2$$

The steps of Random Forest algorithm is explained as follows.

1) Start with the selection of random samples.

2) Construct a decision tree for every sample, and get the result from every decision tree.

3) Perform voting for every result.

4) Select the most voted result as the final prediction result.

Random Forest has its own predominance. On the one hand, it costs relatively little time on a large dataset. On the other hand, it performs well for estimating missing data, which makes the accuracy of the prediction model robust even when a large amount of data is missing.

In this paper, we use Random Forest Classifier from the sklearn package in python to build the model.

3.3.2. XGBoost

XGBoost is also a decision tree based algorithm and is an improvement of the gradient boost algorithm. The steps of Random Forest algorithm is explained as follows.

1) First, we input the training set $\{(x_i, y_i)\}$, a loss function $L(y, F(x))$, weak learners M and a learning rate α.

2) Initialize model with a constant

$$\hat{f}_{(0)}(x) = \arg\min_{\theta} \sum_{i=1}^{N} L(y_i, \theta)$$

3) For $m = 1$ to M:

a) Compute the gradient and the hessians:

$$\hat{g}_m(x_i) = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}_{(m-1)}(x)}$$

$$\hat{h}_m(x_i) = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)^2}\right]_{f(x)=\hat{f}_{(m-1)}(x)}$$

b) Fit a base learner using the training set $\left\{x_i, -\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\right\}_{i=1}^{N}$ by solving the optimization question below:

$$\hat{\varphi}_m = \arg\min_{\varphi} \sum_{i=1}^{N} \frac{1}{2} \hat{h}_m(x_i) \left[-\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} - \varphi(x_i)\right]^2$$

$$\hat{f}_m(x) = \alpha \hat{\varphi}_m(x)$$

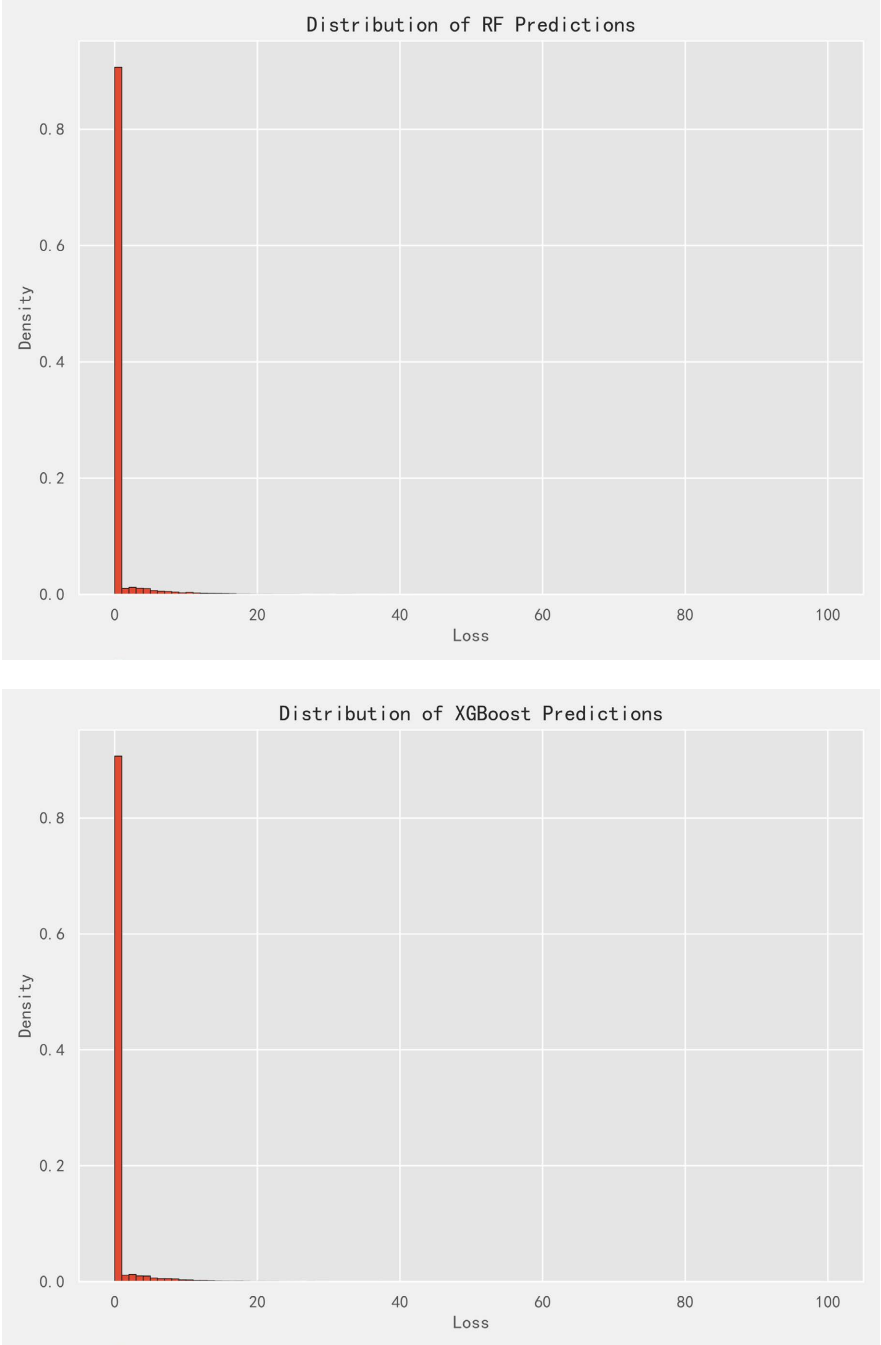

Figure 2. The prediction distributions of Random Forest and XGBoost.

c) Update the model:

$$\hat{f}_m(x) = \hat{f}_{(m-1)}(x) + \hat{f}_m(x)$$

4) Output

$$\hat{f}(x) = \hat{f}_{(X)}(x) \sum_{m=0}^{M} \hat{f}_m(x)$$

In this paper, we use XGBClassifier from the xgboost package in python to

build the model.

3.4. The Prediction Accuracy

The prediction accuracy of the Random Forest model is 0.90657, while the prediction accuracy of the XGBoost model is 0.90635. The result indicates that Random Forest and XGBoost show little difference in the accuracy of their predictions, and both get high accuracy in the loan default cases. The prediction distributions of Random Forest and XGBoost are shown in Figure 2.

4. Conclusions & Discussion

This paper verifies the ability of Random Forest as well as XGBoost to predict loan default. In the feature engineering part we use the variance threshold method and Variance Inflation Factor method to filter out unimportant features, and then we input those selected features into Random Forest and XGBoost models. It turns out that Random Forest and XGBoost show little difference in the accuracy of their predictions since both get high accuracy in the loan default cases. In future research, we will implement comparative studies using other advanced machine learning algorithms such as Neural Network, KNN, and MLP, or the mixed model of them, to find the most suitable model for loan default prediction.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Lai, L. (2020) Loan Default Prediction with Machine Learning Techniques. 2020 *International Conference on Computer Communication and Network Security* (*CCNS*), Xi'an, 21-23 August 2020, 5-9. https://doi.org/10.1109/CCNS50731.2020.00009

[2] Aslam, U., Tariq Aziz, H.I., Sohail, A., *et al.* (2019) An Empirical Study on Loan Default Prediction Models. *Journal of Computational and Theoretical Nanoscience*, **16**, 3483-3488. https://doi.org/10.1166/jctn.2019.8312

[3] Madaan, M., Kumar, A., Keshri, C., *et al.* (2021) Loan Default Prediction Using Decision Trees and Random Forest: A Comparative Study. *IOP Conference Series: Materials Science and Engineering*, **1022**, 012042. https://doi.org/10.1088/1757-899X/1022/1/012042

[4] Malekipirbazari, M. and Aksakalli, V. (2015) Risk Assessment in Social Lending via Random Forests. *Expert Systems with Applications*, **42**, 4621-4631. https://doi.org/10.1016/j.eswa.2015.02.001

[5] Ma, X., Sha, J., Wang, D., *et al.* (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/10.1016/j.elerap.2018.08.002

[6] Nalić, J. and Švraka, A. (2018) Using Data Mining Approaches to Build Credit Scor-

ing Model: Case Study—Implementation of Credit Scoring Model in Microfinance Institution. 2018 17*th International Symposium Infoteh-Jahorina* (*INFOTEH*), East Sarajevo, 21-23 March 2018, 1-5. https://doi.org/10.1109/INFOTEH.2018.8345543

[7] Baesens, B., Roesch, D. and Scheule, H. (2016) Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS. John Wiley & Sons, Hoboken. https://doi.org/10.1002/9781119449560

[8] Han, J.T., Choi, J.S., Kim, M.J., *et al*. (2018) Developing a Risk Group Predictive Model for Korean Students Falling into Bad Debt. *Asian Economic Journal*, **32**, 3-14. https://doi.org/10.1111/asej.12139

[9] Marqués, A.I., García, V. and Sánchez, J.S. (2012) Exploring the Behaviour of Base Classifiers in Credit Scoring Ensembles. *Expert Systems with Applications*, **39**, 10244-10250. https://doi.org/10.1016/j.eswa.2012.02.092

[10] Wang, Y. and Priestley, J.L. (2017) Binary Classification on Past Due of Service Accounts Using Logistic Regression and Decision Tree. Grey Literature from PhD Candidates. http://digitalcommons.kennesaw.edu/dataphdgreylit/

[11] Li, X.H. (2013) Using "Random Forest" for Classification and Regression. *Chinese Journal of Applied Entomology*, **50**, 1190-1197.

[12] Li, Z., Li, S., Li, Z., *et al*. (2021) Application of XGBoost in P2P Default Prediction. *Journal of Physics*: *Conference Series*, **1871**, 012115. https://doi.org/10.1088/1742-6596/1871/1/012115

[13] Akinwande, M.O., Dikko, H.G. and Samson, A. (2015) Variance Inflation Factor: As a Condition for the Inclusion of Suppressor Variable(s) in Regression Analysis. *Open Journal of Statistics*, **5**, 754-767. https://doi.org/10.4236/ojs.2015.57075

[14] Gholamy, A., Kreinovich, V. and Kosheleva, O. (2018) Why 70/30 or 80/20 Relation between Training and Testing Sets: A Pedagogical Explanation. Departmental Technical Reports (CS). https://scholarworks.utep.edu/cs_techrep/1209

[15] Breiman, L. (2001) Random Forests. *Machine Learning*, **45**, 5-32. https://doi.org/10.1023/A:1010933404324