Proceedings of the 3rd International Conference on Financial Technology and Business Analysis
DOI: 10.54254/2754-1169/154/2024.19542

Evaluate Lending Club Loan Status by Machine Learning

Zilan Liu[1,a,*]

[1]*Xiamen University Malaysia, Jalan Sunsuria, Bandar Sunsuria, 43900 Sepang, Selangor, Malaysia*
a. Fin1909322@xmu.edu.my
**corresponding author*

Abstract: To reduce the information asymmetry between investors and lenders and enable P2P lending platforms to develop more accurate lending standards, this study uses Lending Club data to build a credit scoring model based on machine learning (ML) and artificial neural network (ANN) algorithms. The data is first cleaned and preprocessed through feature engineering techniques, and then trained using XGBoost to evaluate feature importance. To address the class imbalance problem, undersampling is used to divide the dataset into training and test sets. The study also uses grid search and cross-validation methods to determine the best hyperparameters for four algorithms (logistic regression (LR), random forest (RF), lightweight gradient boosting machine (LightGBM), and XGBoost). Finally, this paper compares and analyzes the performance of different models and points out the practicality of different models. The results of this process provide insights into the most important features for predicting creditworthiness and identify the most effective model for improving lending decisions on P2P platforms.

Keywords: Evaluate Lending Club Loan Status, logistic regression (LR), random forest (RF), lightweight gradient boosting machine (LightGBM).

1. Introduction

Lending Club represents the largest peer-to-peer (P2P) lending platform in the United States. By the conclusion of 2016, the company had facilitated the issuance of over two million loans, with a collective value of 24.6 billion U.S. dollars. The question thus arises as to how this result was achieved. The following section will provide an overview of the Lending Club. Lending Club is a peer-to-peer lending platform that facilitates connections betiten borroitrs seeking personal loans and investors willing to provide funding for those loans. This platform has achieved financial technology innovation by enabling borroitrs and lenders to circumvent traditional financial institutions as intermediaries for borrowing services through peer-to-peer (P2P) lending.

Thomas found that incorporating credit history into scoring models can improve the ability to predict [1]. A study by Ahmeti found that financial information, especially information about debt levels and profitability, is more effective in predicting financial distress than other factors traditionally considered, such as credit history [2]. When it comes to the P2P lending platforms, Iyer et al. found that the credit score, borrower's current and total defaulted loans, debt-to-income ratio, and loan amount are the most important default factors [3]. Everett considered the credit score, age of the borroitr, house ownership, guarantor, and loan amount as important factors [4, 5].

DOI: 10.54254/2754-1169/154/2024.19542

The business problem to be solved is to construct a notebook credit model, often referred to as a behavioral model, for existing accounts. The model aims to predict which existing accounts are likely to become non-performing. Because such borrowing patterns are very risky for investors, there is liquidity risk and credit risk. The paper uses machine learning methods to predict lending club loan status, including logistic regression (LR), support vector machines (SVM), decision trees (DT), random forests (RF), and extreme gradient boosting (XGBoost).

2. Data Source and Data Description

The paper uses the open P2P loan data provided by the Lending Club. This dataset contains all of the information collected by the platform during its loan process. In this data, it has 887379 entries and 74 rows. From the picture below, it included 2 int64, 49 float64, and 23 objects. Besides that, it also shows different names, such as the Non-Null Count and Dtype.

2.1. Preprocessing and Model Build

This problem is a credit type score which is used to predict whether a loan is good or bad. Set y=1 for the goods and 0 for the bads.

2.2. Data Cleaning

To make it easy to extract the characteristics and train the models, it cleaned the data by applying the concept of feature engineering. There are five steps, including removing redundant features, converting features, dealing with missing data and scaling, under-sampling, and feature selection.

First, it removed the features that were empty columns and transformed the issue dates by year. This code takes the current date and transforms it into a year-month format. Since most of the data, it is in the form of categories, which is not suitable for model training the data needed to be converted into numerical forms. So this paper set y=1 for the goods and 0 for the bads. This paper will remove early defaults (16-30) and grace period loans because they have not yet deteriorated definitively. It will also set a good/bad flag, just like a typical credit score, where higher is better (Table 1).

Table 1: Data cleaning

Category	Loan Condition	adjusted Loan Status
Good Loan	1	Fully Paid, Current, Issued
Bad Loan	0	Default, Charged Off, Late (31-120 days)

This paper creates a new column and converts emp_length to integers. In the loan_status column, There are a total of 876,020 records, including 817,962 good ones and 58,058 bad ones, for a defect ratio of 0.066% (Table 2).

Table 2: Data cleaning for emp_length

	loan_status	count
0	Current	601779
1	Fully Paid	207723
2	Charged off	45248
3	Late(31-120 days)	11591
4	Issued	8460
5	Default	1219

DOI: 10.54254/2754-1169/154/2024.19542

It will remove loans that 'do not comply with credit policies because of their uncertainty, and you may not want the model to learn from them. Since there it is missing values in the data, it needed to deal with this issue before processing the model training data. Some features it is related to historical records,

There are also some other data-cleaning parts, such as deleting fields that it thinks may reveal the target. This paper deletes the interest_rate and purpose columns. Delete some fields that it has already converted, as shown in the table below. Delete some duplicate features (Table 3).

Table 3: More data cleaning

delete	but keep
loan_condition	loan_condition_int
emp_length	emp_length_int

Delete fields with too many missing values. There are many columns with a lot of missing data. Here it deletes columns with more than 30% missing data, which means that only columns with at least 70% of the total number of rows in the dataset with non-empty values will be retained. This is not always the best approach, as sometimes even fields with a small amount of data may contain valuable information. However, without direct business guidance, if it keeps these fields, it is more likely that it will overfit (Table 4).

Table 4: More data cleaning for empty values

	Missing Count	Missing percent	Type
tot_cur_bal	67434	7.697769	float64
total_rev_hi_lim	67434	7.697769	float64
emp_length_int	444111	5.069633	float64
revol_util	458	0.052282	float64

Finally, it filled in the missing fields, also known as imputation. It looked up the count and percentage of missing values, looked for another related field, grouped by that related field, and calculated the mean/median of the missing field.

2.3. Build Variables

Through data cleaning, it has established a good environment for subsequent steps, such as feature selection and model building. The variables are standardized and categorized, while our data is more accurate, clear, and complete. The paper first creates variables and then deletes outliers as needed. The best way to create variables is to gain an in-depth understanding of the data and the business problem. The paper tries to create clever variables that are as relevant as possible to the prediction goal. The paper starts with numeric fields and then works on categorical fields. First, save the record number (id) and the dependent variable y. Then Make the best guess to automatically set fields as either numeric or categorical.

There are a few steps to this. First, save 'id' as data frame X and 'target' as data frame Y. Next set the fields to categorical or numerical. In this step, this paper uses a unique function to identify the numerical fields. The rest are categorical fields. Manual adjustments are also made as needed. The third step is to print the numerical and categorical tables. In this step, it has a few things to delete: 1. Delete 'policy_code' because it only has one value; 2. Delete 'id' because it is a record indicator 3. Delete 'target' because it is not using it to construct variables

DOI: 10.54254/2754-1169/154/2024.19542

After completing these steps, it will operate according to the two aspects of numeric fields and categorical fields. Numeric fields require us to calculate all possible data, and it needs to use eps to ensure that the denominator is not 0. In categorical fields, it creates a single binary variable from a categorical variable with a small cardinality by encoding it singly, ensuring that its unique value ≤ 15. In Target Encoding, do 1- and 2-d target encoding for all low-cardinality categoricals. It will also create a dataset containing only categorical variables. Then run the create 'X_id_save', delete 'id', and use the z-score function to calculate the mean and standard deviation of each field.

Now it has built all the variables and they contain all the data it needs. By classifying variables into categoricals and numerics, it are ready for future analysis. It helps us to build a better-supervised model, loiter the risk, and boost prediction accuracy.

2.4. Feature Selection

According to Keogh, too many features will make the multidimensional feature space too sparse, which will easily make the distance function meaningless [6]. Therefore, this study screened out several important features to avoid this kind of dimensional disaster. In the last step, it has built over 300 variables.

At this juncture, it employs feature selection to reduce the number of variables. By reducing the number of variables, this paper reduces the dimensionality, thereby facilitating the construction of a greater number of candidate variables while simultaneously optimizing the model structure and hyperparameters.

2.5. Filters

This section will focus on filters and wrappers. Filters are used to measure the univariate relationship between each independent variable and the dependent variable. Therefore, this paper uses Python to calculate the filter score of each independent variable. By comparing the scores, it can be seen that the stronger the relationship between the independent variable and the 'target' field, the higher the score. Finally, the top 40 variables will be selected. (291->40) (Figure 1).

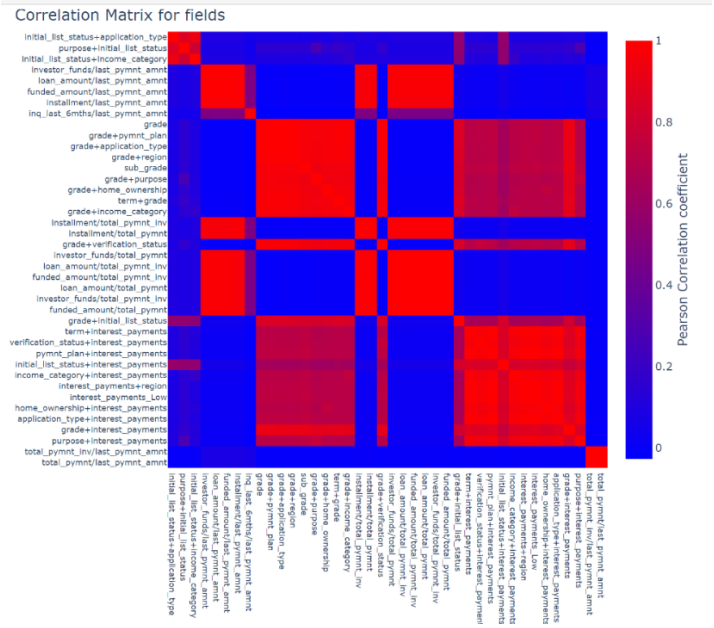

Figure 1: Correlation Matrix for fields (Photo/Picture credit: Original).

DOI: 10.54254/2754-1169/154/2024.19542

2.6. Wrapper

Wrapper is a method for eliminating the correlation between independent variables, which complements the previous method. In general, there are three types of wrappers: forward selection, backward selection, and general stepwise selection.

In Figure 2, when the number of features is equal to 6, the performance level is unchanged, but to be conservative, 10 features are selected. Through two steps of feature selection, it reduces the number of independent variables from 291 to 10, which is efficient.

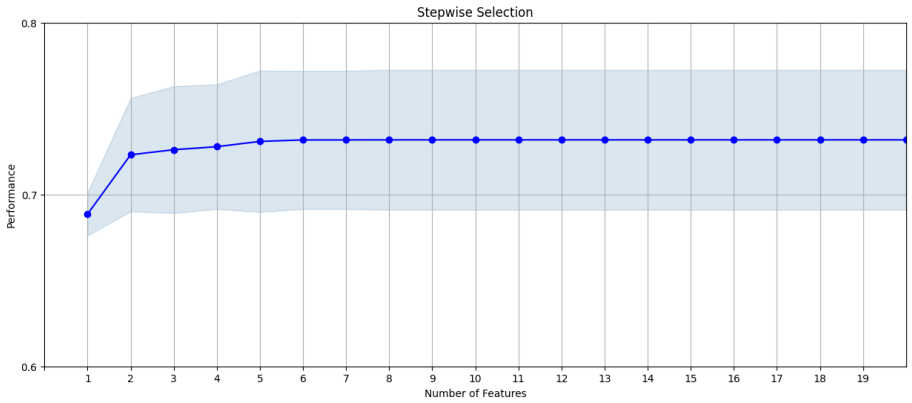

Figure 2: Stepwise selection (Photo/Picture credit: Original).

3. Model selection

This time, it used three different data sets: 1. X_model (normal); 2. X_model_smote (oversampled using SMOTE – large amount of; 3. X_model_sample (retains all bad data and samples good data). It will use the following four methods to measure the pros and cons and select the model:

LR: original data (log reg), sampled data (log reg sampled), and SMOTE data (log reg smote). Traditionally, in statistics, it solves this kind of problem by using probit or logit models, which assume that the probability of event occurrence obeys a certain probability distribution. If the probability of event occurrence obeys the cumulative standard normal distribution, it usually uses a probit model.

RF: raw data (RF) and sample data (RF sampled). The RF is an ensemble learning method that was first proposed by Ho and developed to have a bagging (bootstrap aggregation) method and a random subspace method [7, 8]. The algorithm can help solve the problem of DTs being easier to become a highly irregular pattern when they grow deeper.

CatBoost: raw data (CatBoost) and sampled data (CatBoost sampled). CatBoost is a GBDT framework based on symmetric decision trees (oblivious trees)as a base learner that has finite parameters, supports categorical variables, and has high accuracy. The main pain point it solves is the efficient and reasonable processing of categorical features, as can be seen from its name, CatBoost is made up of Categorical and Boosting. In addition, CatBoost also solves the problems of gradient bias (Gradient Bias)and prediction shift (Prediction shift)to reduce overfitting and improve the accuracy and generalisation ability of the algorithm.

LightGBM: raw data (LGBM), sampled data (LGBM sampled), and SMOTE (LGBM smote) data. According to Ke, ightGBM is a distributed gradient-boosting framework for ML algorithms originally developed by Microsoft [9]. Similar to XGBoost, LightGBM is also an extension of the GBDT algorithm. Ke proposed two methods, i.e., gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB), to reduce the amount of data and features without affecting the prediction capability of the model, which can help reach a balance between efficiency and accuracy [9, 10].

DOI: 10.54254/2754-1169/154/2024.19542

Here are the steps on how to obtain the ideal model: First, use 'score (DR%)' (detection rate) to evaluate the model performance of different iterations and datasets. For each model, trn_score, tst_score, and val_score represent the performance metrics on the training, test, and validation datasets, respectively. Store the results in 'Modeling_output' and print the average results. Then, rearrange the metrics to be suitable for plotting. and calculate the mean and standard deviation of the detection rate for each model. Use box plots to compare the metrics for the training, test, and validation sets. Finally, delete 'trn' and 'tst' and only view the validation set to select a model with good and robust performance: select CatBoost Classidier (high detection rate).

From Figure 3, the max value/best value is almost 0.52. It shows the relationship betiten training, validation, and test data sets used by the different models and scores.

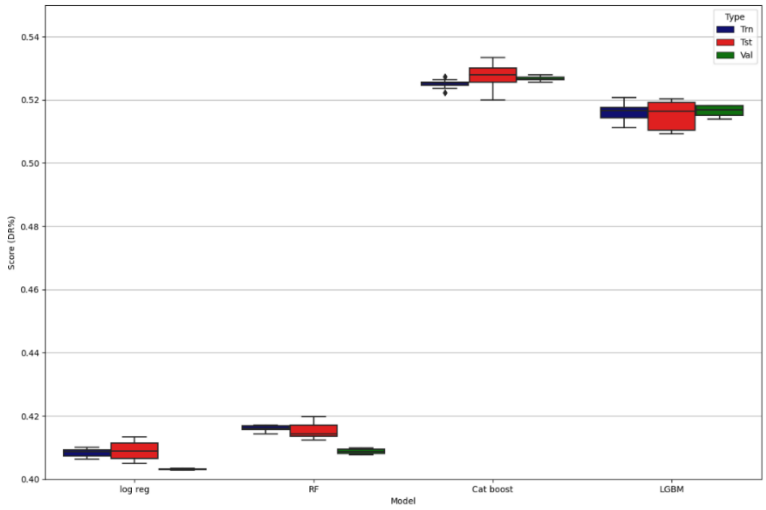

Figure 3: Model selection (Photo/Picture credit: Original).

4. Result

In the paper, the random forest model was selected was selected. Then, the algorithm was run to find the most predictive variables.

Table 1: Different model results

Model	accuracy	precision	recall	f1-score
RF	0.74	0.88	0.80	0.84
XGBoost	0.75	0.88	0.81	0.85
LightGBM	0.75	0.88	0.81	0.84
CatBoost	0.83	0.85	0.96	0.91

As can be seen from Table 1, CatBoost performs best in four indicators, especially in the recall, which is significantly better than other models, indicating that it is very sensitive in identifying positive examples. XGBoost and LightGBM have very similar performance and are well balanced; while RF is relatively weak in all indicators, especially in F1 score. If the task requires high recall and tolerates a certain amount of false positives, CatBoost is the best choice; if it needs to balance precision and recall, XGBoost or LightGBM is more suitable; and for smaller data sets or simple tasks, Random Forest may be sufficient.

DOI: 10.54254/2754-1169/154/2024.19542

5. Conclusion

This article mainly focuses on data processing, including feature creation, feature selection, and model selection. First, this article lays the foundation for subsequent feature selection and model construction through data cleaning and variable construction. By standardizing and classifying the data, the article ensures the accuracy and completeness of the data. During the feature creation process, this article carefully creates numerical and categorical variables according to business needs and data characteristics and removes outliers.

In the feature selection stage, this article first uses the filter and wrapper method to filter features. Finally, it reduces the number of features from 291 to 10 through two rounds of feature selection, ensuring the optimization of data dimensions and improving the model's prediction accuracy. Then, four different methods (LR, RF, CatBoost, LightGBM) are used to select and evaluate models for different data sets. In the end, the CatBoost model performs best in accuracy (0.83), recall (0.96), and F1 score (0.91), especially in recall, which is significantly better than other models and is suitable for tasks that require high recall.

In summary, CatBoost performs best in all indicators, XGBoost and LightGBM have similar performance and are suitable for tasks that require a balance between precision and recall, while Random Forest is relatively weak and is suitable for processing smaller data sets or simple tasks. Finally, the experimental results of the article prove that careful data preprocessing and feature selection can significantly improve the performance and prediction accuracy of the model.

References

[1] *PGeorgios, P. (2022). Data-assisted modeling of complex chemical and biological systems (Doctoral dissertation). Johns Hopkins University.*

[2] *Laureta, A., & Zubanovic, A. (2020). The predictive power of financial ratios on bankruptcy: A quantitative study of non-listed limited liability SMEs companies in Sweden.*

[3] *Bishop, M. (2006). Pattern recognition and machine learning. Springer.*

[4] *Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer.*

[5] *He, K. M., Zhang, X. Y., Ren, S. Q., et al. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. IEEE International Conference on Computer Vision, 1026–1034.*

[6] *Weikuan, J. (2022). Feature dimensionality reduction: A review. Complex & Intelligent Systems, 8(3), 2663–2693.*

[7] *Bolton, P. (2010). Logistic regression and its application in credit scoring (Master's thesis). University of Pretoria*

[8] *Guyon, I., & ElNoeeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157–1182.*

[9] *Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning, 448–456.*

[10] *Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 1097–1105.*