Loan Default Prediction with Statistical Models

Muxue Liu, Yaxuan Hou, Xinbei Li, Huiyao Yang

April 2023

1 Background and Introduction

With many small companies going bankrupt due to the huge financial pressure in recent years, banks face a greater risk of default and thus are more careful to evaluate the financial conditions of a company. Therefore, this paper aims to help banks to predict the loan condition and possible losses.

Data used in this paper is from Kaggle named " loan default prediction data set". It is a table of records describing companies' financial attributes and grades by the bank. It contains 35 variables, 53970 training samples, and 13493 test samples. This paper tries various classifiers like Logistic regression, QDA, KNN, Random Forest, and AdaBoost, to predict the loan default based on previous loan history, and use multiple methods to evaluate the results.

2 Methodology

2.1 Exploratory Data Analysis

2.1.1 Type and uniqueness

This paper checks the types and uniqueness of each variable. 9 non-numerical variables, "Batch Enrolled", "Grade", "Sub Grade", "Employment Duration", "Verification Status", "Payment Plan", "Loan Title", "Initial List Status" and "Application Type", are observed. Besides, the variable "Term" only has three values.

To handle the above problems, this paper tries to convert non-numerical into numerical ones and classify some numerical variables into categorical variables.

1. "Payment Plan" only has one unique value 'n' and should be deleted.
2. "Loan Title" is written randomly by borrower and can't contribute to the prediction and should be deleted.
3. "Application Type", "Grade", "Sub Grade", "Employment Duration", "Verification Status", "Batch Enrolled", "Initial List Status" have several different unique values and is meaningful in predicting loan default. They should be converted into numerical data by encoding them with value between 0 and the number of unique value.
4. "Term" has three values, so change to categorical data.

2.1.2 Boxplot

In Figure 1, this paper uses boxplots to visualize the difference in distribution between the two classes of the "Loan Status", and it shows that there is no distinct patterns in the features' distribution in terms of the target variable.

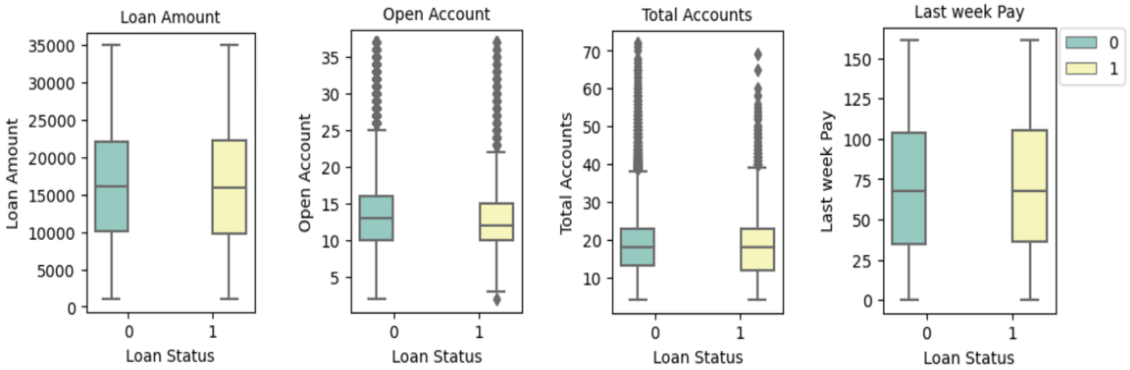

Figure 1: Part of the Box Plot

2.1.3 Pairplot

This paper uses the pair plot for the exploratory data analysis, as shown in Figure 2. The values of most variables for companies that defaulted and those that did not highly overlap, which makes it hard for us to predict whether the company would default, and there is no outstanding feature to separate two classes.

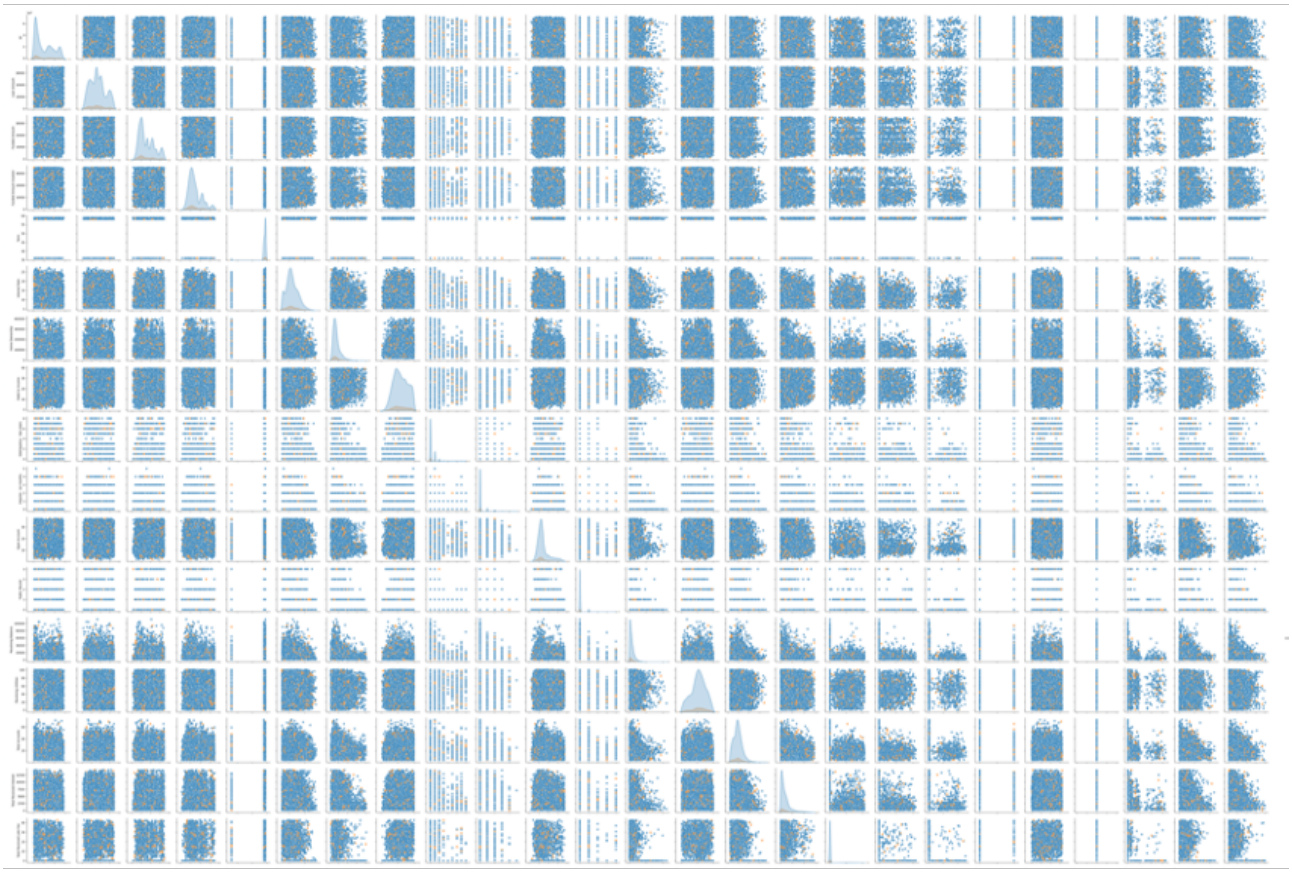

Figure 2: Part of the pair plot

2.1.4 Drop features

Third, by checking the variables' meaning and uniqueness, this paper drops the variable "ID", and "Accounts Delinquent".

1. "ID" indicates the unique ID of the representative, which is meaningless to the following analysis.

2. The values of "Accounts Delinquent" for each sample are the same.

2.2 Missing Value and Correlation

This paper tries to select the remaining variables by checking missing values and correlations between variables. It is shown in Figure 3 that there's no missing value in this data set. At last, the correlation matrix shows that the correlation between variables in this dataset is pretty small and doesn't require modification.

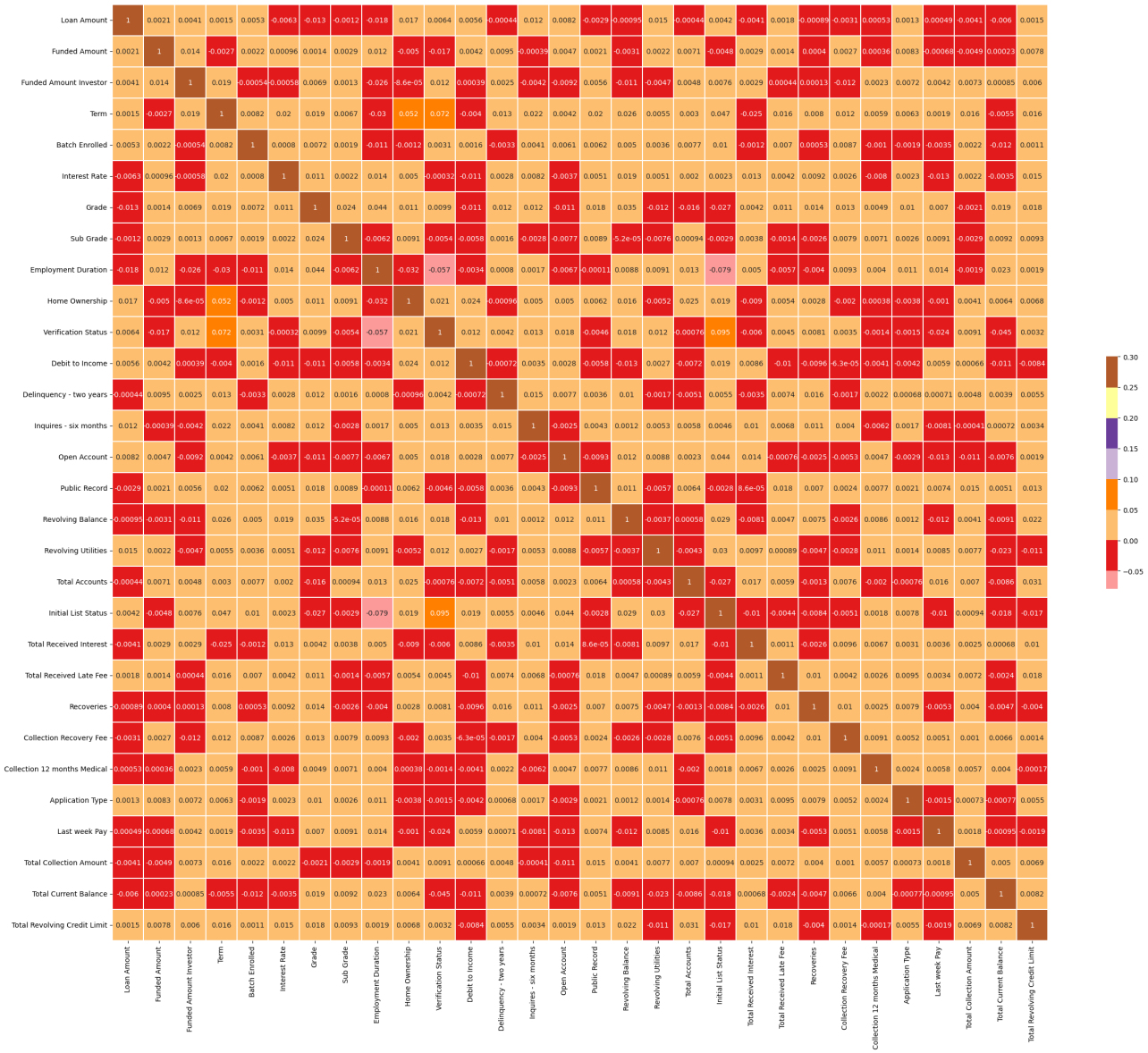

Figure 3: correlation Matrix between variables

2.3 Oversample and SMOTE

In this training set, there are 48977 negative samples and 4993 positive samples. The ratio of the two classes is nearly 10:1. It indicates that the training data is quite imbalanced. This paper uses the SMOTE method to oversample the data rather than downsample because the value of variables in these 2 labels are highly overlapped and downsample would randomly decrease the sample size leading to loss of some information. After SMOTE, the sample size of two classes are equal.

2.4 Statistical models and Metrics

We trained on Logistic regression, Quadratic discriminant analysis, K-nearest neighbor, Random forest, Adaboost models. Besides, the hyperparameters in model are chosen by k-fold validation, and evaluated models based on Confusion matrix, precision, recall, F1 score, BER, and AUC.

3 Results

3.1 Hyperparameters Tuning for KNN, Random Forest, Adaboost

3.1.1 KNN

By 10-fold validation method, the following figure shows the corresponding average balanced accuracy with different value of k. From the Figure 4, k ranges from 3 to 40. The highest average balanced accuracy is achieved when k is equal to 4. Thus, k = 4 is chosen to build KNN.

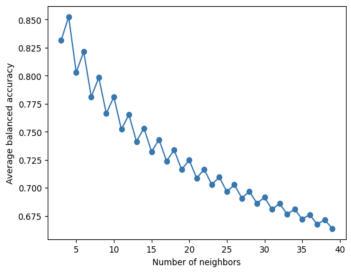

Figure 4: KNN Hyperparamter Tuning

3.1.2 Random Forst

Applying 10-fold validation method to determine the hyperparameters in Random Forest, including number of trees, depth of trees, and number of input variables considered for each split, the following figure shows the corresponding average balanced accuracy for each parameter value. From Figure 5, get max depth = 38, max features = 3, n estimators = 1000 to achieve the highest value, and thus use these values to build the Random Forest model.

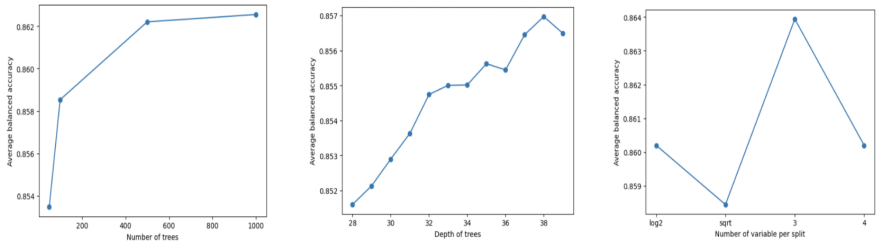

Figure 5: Random Forest Hyperparamters Tuning

3.1.3 Adaboost

Applying 5-fold validation method to determine the hyper parameters in Adaboost, including number of trees, depth of trees, the following figure shows the corresponding average balanced accuracy for each parameter value. From Figure 6, we get max depth = 27, n estimators = 2000 to achieve the highest value, and thus use these values to build the Adaboost value.

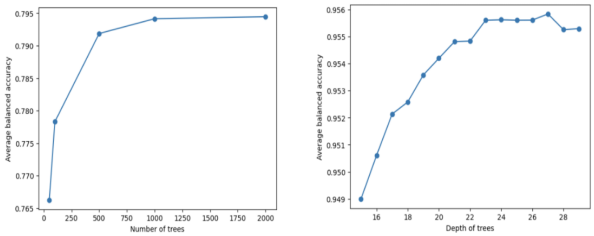

Figure 6: Adaboost Hyperparameters Tuning

3.2 Model Results and Conclusion

3.2.1 Logistic Regression

The Confusion matrix, AUC, and ROC of Logistic Regression are shown in Figure 7. These results show that Logistic Regression performs poorly. For label $y = 0$, the accuracy is around 56.8%, while the accuracy for label $y = 1$ is only 46.0%. Besides, the AUC is around 0.52.

As mentioned in the Exploratory Data analysis part, the two classes' features are highly overlapping, so do not exist a clear linear line to separate the two classes. This may be the reason that Logistic Regression does not perform well.

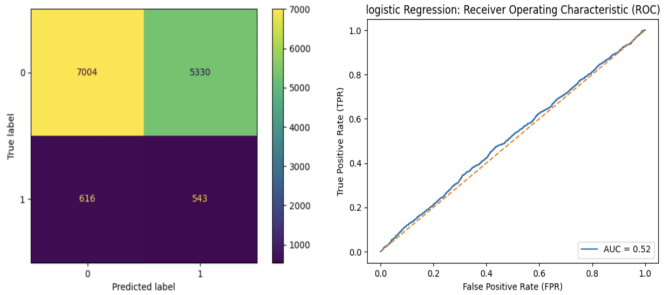

Figure 7: Logistic Regression Result

3.2.2 Quadratic Discriminant Analysis

The Confusion matrix, AUC, and ROC of QDA are shown in Figure 8. These results show that QDA performs better in predicting label $y = 1$ than label $y = 0$. For label $y = 0$, the accuracy is less than 25%, while the accuracy for label $y = 1$ is around third times, 75.92%. Besides, the AUC is around 0.49.

The factors causing this weird and worst performance may be the oversampling method and the data do not satisfy the Gaussian assumption. From the pair plot in the Exploratory Data analysis part, the shape of the data is arbitrary, which can be a square shape or an abnormal shape.

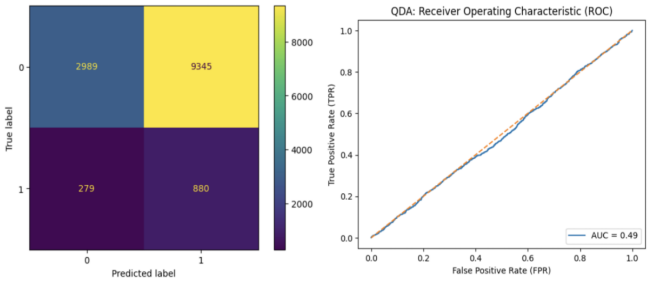

Figure 8: QDA Result

3.2.3 KNN

The Confusion matrix, AUC, and ROC of KNN are shown in Figure 9. These results show that KNN performs better than the previous two models. For label $y = 0$, the accuracy is around 73%, while the accuracy for label $y = 1$ is around 70.2%. Besides, the AUC is around 0.74.

Compared with the above two models, all the evaluation metrics are improved, since KNN does not have assumptions and the data is sparse. However, the problem of the curse of dimensionality may lead to not very high accuracy in two classes, because 30 predictors in the KNN model.

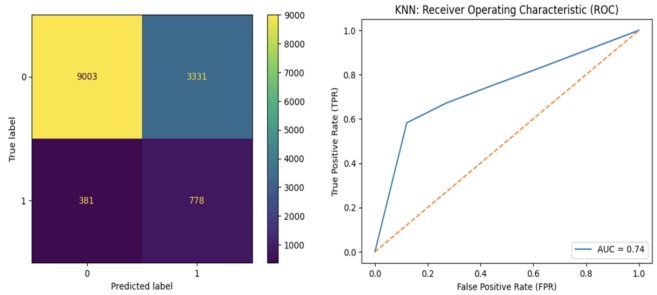

Figure 9: KNN Result

3.2.4 Random Forest

The Confusion matrix, AUC, and ROC of the Random Forest are shown in Figure 10. These results show that Random Forest performs well in label $y = 0$ than in label $y = 1$. For label $y = 0$, the accuracy is around 88.6%, while the accuracy for label $y = 1$ is around 47.8%. Besides, the AUC is around 0.72.

Even before building Random Forest, we over-sampled the label $y = 1$, and the performance of label $y = 1$ is still bad. We tried to adjust sample sizes for two labels around 8:9, but the results shifted. The model performs well with label $y = 1$ and bad $y = 0$. So the reason for the poor Random Forest model may be not the oversampling method, but the Lack of outstanding input features in the data set to classify two classes.

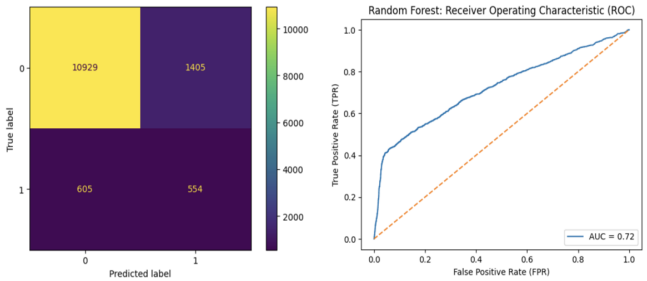

Figure 10: Random Forst Result

3.2.5 Adaboost

The Confusion matrix, AUC, and ROC of Adaboost are shown in Figure 11. These results show that KNN performs better than the previous two models. For label $y = 0$, the accuracy is around 97.3%, while the accuracy for label $y = 1$ is around 55.3%. Besides, the AUC is around 0.78.

Adaboost is the best model among the five models, but the BER value is still not below 0.2, around 0.26. Tunning in not enough parameters and lack of outstanding features to separate two classes may be two reasons.

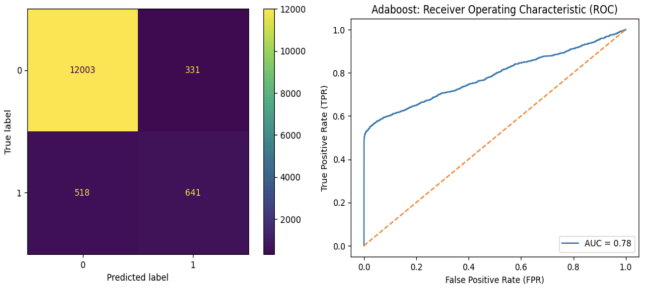

Figure 11: Adaboost Result

3.2.6 Overall Results

Besides the confusion matrix, ROC, and AUC mentioned above. We also calculated Precision, Recall, F1 score, and BER to give a overall results of five model.

The following table is the results of five models.

Models	Precision	Recall	F1 score	BER	AUC
Logistic regression	0.0924	0.4685	0.1544	0.4818	0.5162
QDA	0.0860	0.7593	0.1546	0.4992	0.4935
KNN	0.1893	0.6713	0.2953	0.2994	0.7386
Random Forest	0.2828	0.4779	0.3554	0.3180	0.7218
Adaboost	0.6595	0.5530	0.6016	0.2369	0.7843

From the table and figure, the F1 score, BER, and AUC of Adaboost model is much higher compared with other models, so **Adaboost** model is chosen to predict the loan default.

4 Discussion

The paper explores the challenge of loan defaults faced by banks and the importance of predicting loan defaults before granting loans. After exploratory data analysis and over-sampling with SMOTE, the paper conducts several models including Logistic regression, QDA, KNN, Random Forest, and AdaBoost to predict loan default.

The paper concludes that the AdaBoost model performs the best among the five models based on the F1 score, AUC, and BER. The AdaBoost model achieves an AUC of 0.7843 and a BER of 0.2369. However, the overall model performance is not very satisfactory, which may be caused by several reasons and can be improved in further research.

1. Lack of outstanding input features in the dataset to classify two classes. It is likely that the number of input features is small, or the input features may not be relevant enough for the target variable. To handle this, add more features by referring to features from other related datasets. Also, it is useful for preprocessing data using PCA or tSNE to extract outstanding features to classify classes.

2. Some features need to be more precisely analyzed. This paper drops the variable 'Loan Title' since it is written randomly by the borrower. However, there is still a method to preprocess this feature, like converting the 'Loan Title' to a broadly categorized categorical variable such as medical use, real estate purchase, etc.

3. Oversampling method can be improved. The training set is highly imbalanced, with the ratio of negative to positive samples being nearly 10:1. This paper uses SMOTE to over-sample the positive samples, which may be not the best method to handle unbalanced problem. Further research can use alternative oversampling methods like K-means SMOTE and ADASYN to address the issue of imbalanced datasets.

4. The research is conducted with limited time and restricted compute engines, some complex models such as Random Forest and AdaBoost may not be tuned enough. In further research, the parameters of some advanced models like Random Forest and AdaBoost can be tuned using cross-validation over a wider parameter space, for example, increasing the depth of the tree.

Overall, the paper provides valuable insights into the use of statistical models for predicting loan default. After addressing the above issues, future research can be performed to explore how the findings can be implemented in practice to help banks predict loan default and make more informed lending decisions, ultimately improving their financial stability and the stability of the economy as a whole.