The Lending Club Data Analytics Report

1. INTRODUCTION

In this project, we will analyze the Lending Club data to predict whether the borrowers will repay their loan to better manage risk of loan service. Lending Club is an American person-to-person loan company, headquartered in San Francisco, California. We choose to analyze the data of Lending Club because it is the world's largest person-to-person (p2p) lending platform, which has a credible dataset with a huge amount of digital footprints.

The main problem we focus on is to decide whether we should lend money to our potential clients. We built Logistic Regression, Random Forest, Neural Network, and SVM models to predict whether a client was "fully paid" or "charged off" the loan in the year of 2018. Since borrowers who default cause the most losses to the lender, if we could identify these borrowers with models, we can reduce the losses to the lender.

We used accuracy score, false positive rate, and AUC score to evaluate the performance of models.We stated the hypothesis that the best model to predict loan status of fully paid and charged off is the Random Forest model. Since random forest can reduce error, less impact of outliers, and avoid overfitting.

2. METHODOLOGY

2.1 Description of Dataset

We used the Lending Club dataset obtained from Kaggle. We had 56,237 observations, 18 explanatory variables, and 1 response variable for this project. It originally had approximately 2 millions observations and 151 variables. First, we checked the missing values of each variable and dropped variables with a high percentage of missing values. Then, we selected variables based on our objective. In feature selection, we discovered that some variables were highly correlated with the response variable, but could not be used to answer our questions. After the data selection process, we had 56,237 observations and 19 variables.

In exploratory data analysis, we drew histograms to check the relationship between categorical variables and the response. We found that the user tends to repay their loan if their income status were verified. In numerical variables, we made a heatmap and found that there is a high correlation between the installment and the amount of loan. The fico rate has strong correlation with the response.

2.2 Description of algorithms

2.2.1 Logistic Regression

Logistic Regression was a Machine Learning algorithm that prefers to be used in binary classification. We used it for a dataset that has only two outcomes '1' and '0', in our project we determined whether the loan status is fully paid or charged off. We chose logistic regression as one of the models since it was easy to implement and effective to train. However, compared with other models, logistic regression might not be as powerful as other algorithms. In addition, Logistic regression offered an interpretation of the model coefficients, which could be suggested as indicators of feature importance.

$$log(\frac{\pi}{1-\pi}) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

2.2.2 SVM

SVM was a supervised machine learning algorithm that was used in binary classification. It aimed to differentiate the entire dataset into disparate classes and find an optimal boundary between "fully paid" and "charged off". This boundary was the best separating line, could be linear or nonlinear, that maximizes the distance between the hyperplanes of decision boundaries based on the features we choose. This made the division of vector space into two sets better. SVM was used in high dimension data in binary classification, and dealt with outliers. However, it took time to process the model and caused poor performance for overlapped classes.

2.2.3 K-nearest neighbor algorithm (KNN)

KNN was a mature method in theory and one of the simplest machine learning algorithms. The idea was that a sample belongs to a category if most of the k most similar (that is, the closest neighbors in the feature space) samples in the feature space belong to that category. In the KNN algorithm, the selected neighbors were correctly classified objects. In order to find the nearest "neighbors", the algorithm applied the concept of Euclidean distance. In the Euclidean plane, if two dimensions, we let point p1 have Cartesian coordinates(p1, p1), and q had coordinates (q1, q2). Then the distance between p and q was given by $d(p,\ q)\ =\ \sqrt{(q1\ -\ p1)^2\ +\ (q2\ -\ p2)^2}$. In higher dimensions, for points given by Cartesian coordinates in n-dimensions, we had $d(p,\ q)\ =\sqrt{(q1\ -\ p1)^2\ +\ (q2\ -\ p2)^2\ +\ (q3\ -\ p3)^2\ +\ \ldots\ +\ (qn\ -\ pn)^2}$.

By this mathematical concept, we could easily find its shortest distance neighbor. Based on the parameter k, we could find out its k nearest neighbors. However, it also left some disadvantages. For example, the test sample classification required a large amount of computation and memory overhead, because the distance between each text to be classified and all known samples must be calculated before its K nearest neighbor points could be obtained. At present, the commonly used method is to clip the known sample points in advance, and remove the samples that have little effect on classification in advance. In addition, the selection of parameter k will need deep consideration. For example, when the sample was imbalanced, such as the sample size of one class is very large while the sample size of other classes was very small, it may lead to the majority of the K neighbors of the sample with large volume class when a new sample was imported.

2.2.4 Random Forest

Random Forest was a supervised machine learning algorithm that was based on decision trees. It was based on the logic of bagging. Random Forest served as an ideal method we can implement in this project. First, Random Forest was good at dealing with high dimensional data since it worked with subsets of data. This was good for our project since we have in total 18 predictors which were 18 dimensions. Second, Random Forest could help with finding the importance of variables, so we could view the relationship between predictors and the response. We could find out which predictors affected the outcome of subscription of a term deposit more and which affected less. Also, the prediction of Random Forest was robust to multicollinearity, so this could handle the issue that there were few pairs of variables having multicollinearity in our dataset. In addition, compared with other methods, the random forest tended not to overfit. As the more trees we added into the random forest, the tendency to overfit decreases.

2.2.5 Neural Network

A neural network was a series of algorithms that try to recognize underlying relationships in a set of data through a process that mimics the way the human brain works. A neural network can build nonlinear models of complex relationships. In our dataset, the relationship between predictors and responses was nonlinear and complex. Second, the neural network had a strong predictive ability. The model can effectively infer unknown relationships between unknown data, so that the model can generalize and predict unknown data.

3. IMPLEMENTATION DETAILS

We randomly splited the data with 70% of training data and 30% of test data. We did feature scaling on the training data for better model training. We built models using the training data and predicted with the test data.

3.1 Logistic Regression

We first normalize the data, then all the variables have small p-values close to 0, we can conclude that all the variables are sufficiently important. We build the logistic regression model with default set parameter c =1, inverse of regularization strength, to reduce the generalization error to prevent overfitting the training data.

3.2 SVM

We used SVM with a linear Kernel to separate using a single line since we have a large size of dataset, the linear kernel is faster than other kernels. We try different c regularization parameters when building the model.

3.3 K-nearest neighbor algorithm (KNN)

In order to simulate the situation of finding potential neighbors of a customer to see the proportion of the class between "charged off" and "fully paid" to decide whether we should lend the money to him. We randomly selected 51 rows of our dataset, and chose the first row as our customer information. By our algorithm, we were able to find which class most of his neighbors belong to, and put him in the class with the most people. This helped us to decide whether we should lend money to him.

3.4 Random Forest

We used a random forest algorithm from sklearn. In the selection of tuning parameters, we seted max_features to 6, making the maximum number of features a random forest is allowed to try in individual trees is 6. We set the number of estimators to 100 since it makes the balance between the performance of the model and the speed of the algorithm. We set the minimum sample leaf to 1 since a smaller leaf makes the model more prone to capturing noise in train data. We also found the importance of variables with random forest.

3.5 Neural Network

We used a neural network algorithm from sklearn. It has preset on the input layer and output layers, which are 18 and 1. We set 3 hidden layers, and each contains 10 neurons. We tried multiple combinations of hidden layers and found this is the most powerful. We applied the solver of ‘lbfgs’ as an optimizer

because it can converge faster and perform better. We set the L2 penalty to 0.0001 to avoid overfitting. We set the maximum number of iterations to 100.

4. RESULTS AND INTERPRETATION

	Logistic regression	Random Forest	SVM	KNN	Neural Network
Test Accuracy	**0.976**	**0.987**	**0.976**	**0.969**	**0.988**
AUC Score	**0.950**	**0.976**	**0.951**	**0.925**	**0.976**
False positive/ Total Population	**0.014**	**0.009**	**0.014**	**0.022**	**0.0069**

From the above table, we can find that the Neural Network has the highest accuracy. In surprise, we find the performance of Random Forest and Neural Network are approximately equal. They perform well in both classification tasks. In addition, they have a great performance on non-linear relations of our large dataset. They alse balance bias and variance trade-off well.

We decided to choose a Neural Network as our optimal model. According to risk management, we should control the probability of those who are charged off but in the model we predict as fully paid, which are those false positive values. The neural network has the smallest number of false positive values than others. This is better for the objective of our project that reduces the potential losses to the lender.

In KNN, when the sample data is imbalanced, such as the sample size of one class is very large while the sample size of other classes is very small, it may lead to the majority of the K neighbors of the sample with large volume class when a new sample is imported. Because of this problem, it is hard to decide the parameter k and leads to relative bad prediction.

Also, the overlapped classes cause poor performance in SVM, and selecting different kernel functions may cause the accuracy to be different. In the further study, we could try other kernels like the RBF kernel since our dataset may not be linearly divided by plane. Logistic regression has poor performance because the relationship between predictors and response in our dataset is not linear enough.

5. APPENDIX

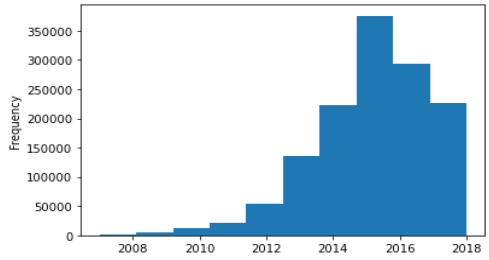

Figure 1. Selection Data by year

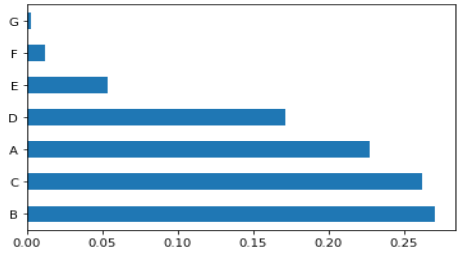

Figure 2. Percentage of each grade

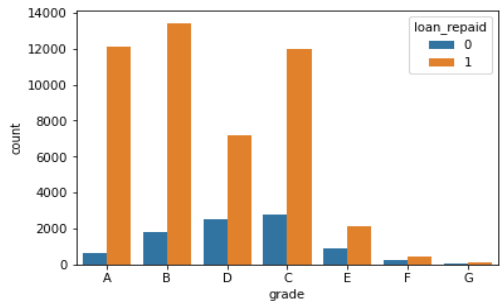

Figure 3. Loan_repaid by grade

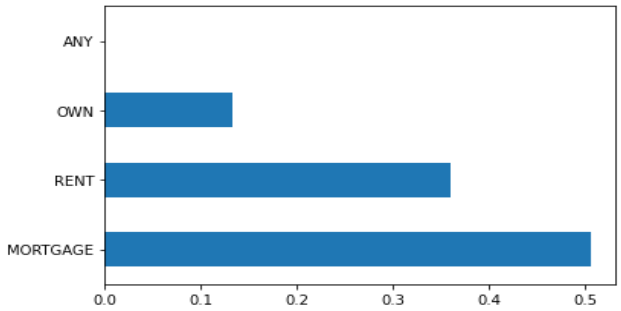

Figure 4. Percentage of home_ownership

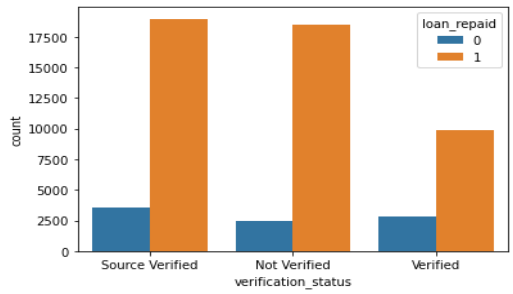

Figure 5. Loan_repaid by verification_status

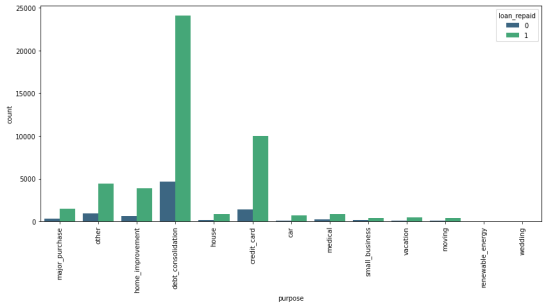

Figure 6. Loan_repaid by purpose

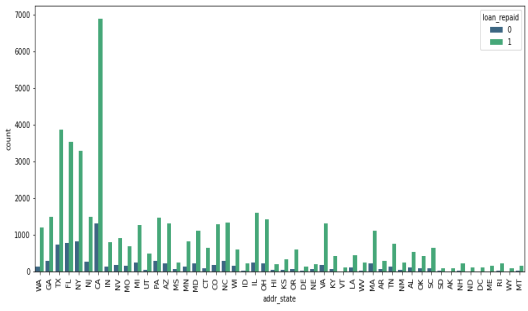

Figure 7. Address state by loan_status

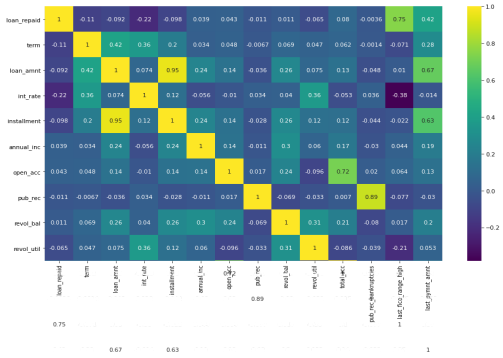

Figure 8. Heatmap

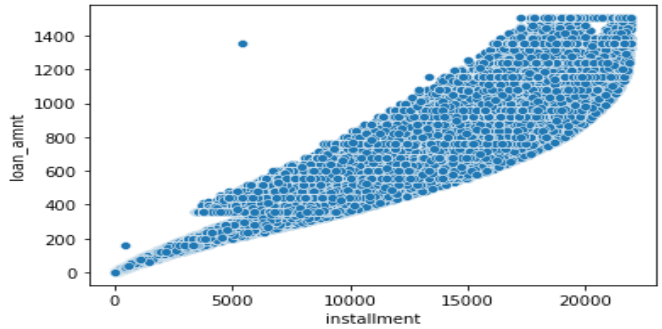

Figure 9. Scatterplot of installment vs loan_repaid

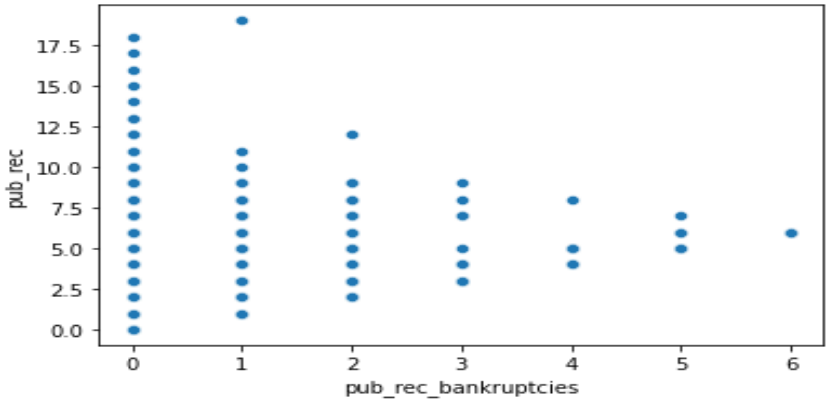

Figure 10. Scatter Plot of pub_rec_bankruptcies vs pub_rec

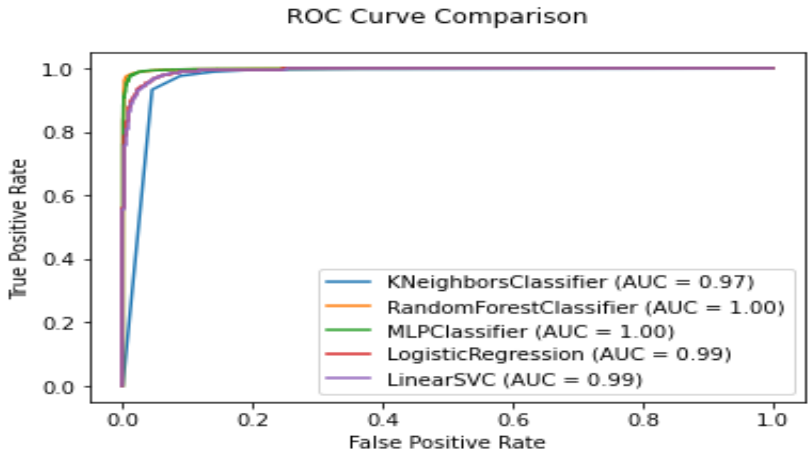

Figure 11. AUC with each model