Prediction of Vehicle Loan Defaulter

Data 1030 Project

https://github.com/zhuoran-han/data1030_project

Zhuoran Han

December 7, 2021

1 Introduction

Since owning a car is necessary for people who live in United States but also expensive, many people choose to apply for vehicle loans from financial institutions. Currently, 85 percent of new car purchase carried a vehicle loan and the statistic shows that there were 113 million open accounts for vehicle loan in the 3rd quarter of 2018. These large numbers can lead finical institutions to face a significant loss when there are vehicle loans defaulters. Therefore, the rejection rate of vehicle loans increases and the need of a prediction model to help finical institutions classify potential defaulters also increases.

This project attempts to create a machine learning model to achieve the goal of classifying potential vehicle loan defaulters. The dataset used in this project is from Kaggle and is provided by LT Financial Service. It contains 255133 data with total 40 features. The features include a unique id,6 features of loan's information including age,employment and identity proof, 14 features of loanee's information including amount of loan disbursed, cost of the Asset and so on, 19 features of Bureau data and history like Bureau score,status of other loans,etc. and a target variable called 'loan default' which determined by whether the payment was made by the first EMI(Equated Monthly Instalments) on due date.

Several authors in Kaggle shared their modeling process. The author Kashyap Narayan tried several model to perform the prediction with the basic splitting method. He concluded the one that perform best is Logistic Regression with SMOTE and it gives a good AUC score on test set of 0.78 and the best f-1 score among all of the other. He noticed that some of the feature have very similar distribution on defaulter and non-defaulter like the age and some of the features have highly skewed distribution like asset cost and disbursement amount. The feature of the Bureau Score appears to be the most relevant feature for prediction.

2 Exploratory Data Analysis

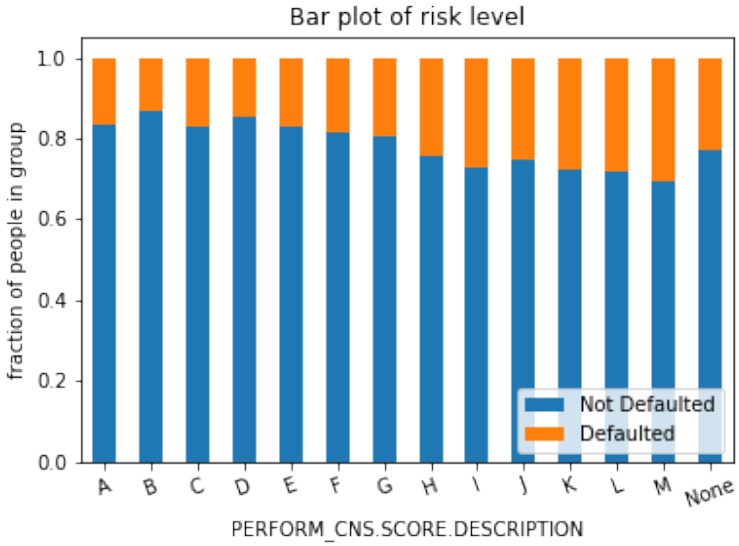

Figure 1: Distribution of risk level

This stacked bar plot shows the distribution of portions of people in each Bureau score description group, segregated by the label of whether the loanee is a defaulter.The letter A through M represents the ordinal risk level from low to high according to the loanee's credit history. There is a positive linear relationship between he fraction of defaulter in each group and the risk level of each group. Represent the ordinal risk level from A to M as 1 to 13, the linear regression equation y=0.014x+0.114 best fit the relationship with error of 0.014. It clearly shows that as the risk level increases, the number of defaulter increases. Therefore, loanee who is marked having higher risk according to credit history, he will be more likely to be a loan defaulter than the loanee who has lower risk. This feature can be important in the prediction model.

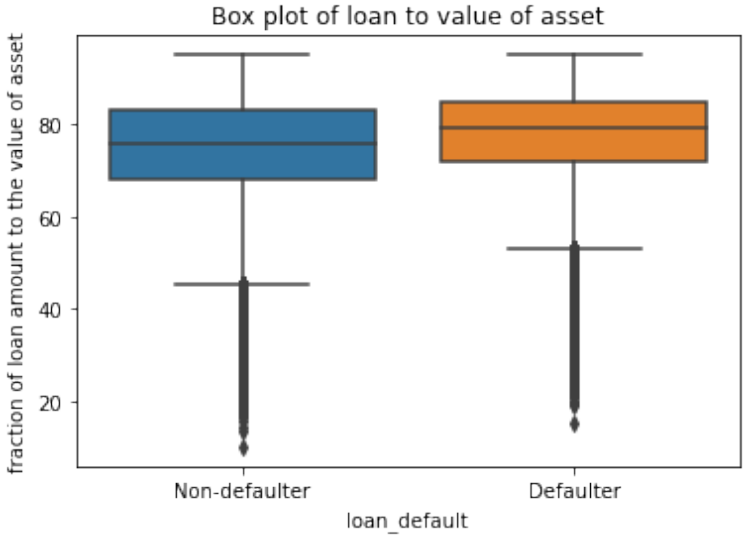

Figure 2: Distribution of the percentage of loan to value of asset segregated by the label of loan default

The mean of the feature in defaulter group is about 2.7% higher than it in non-defaulter group. The values of the three quartiles in defaulter group are also all higher than in non-defaulter group. So one loanee whose loan amount equals to a higher percentage of the cost of the car has a higher chance to not pay their first EMI by due date than those whose loan amount has a lower percentage to the value of the car.

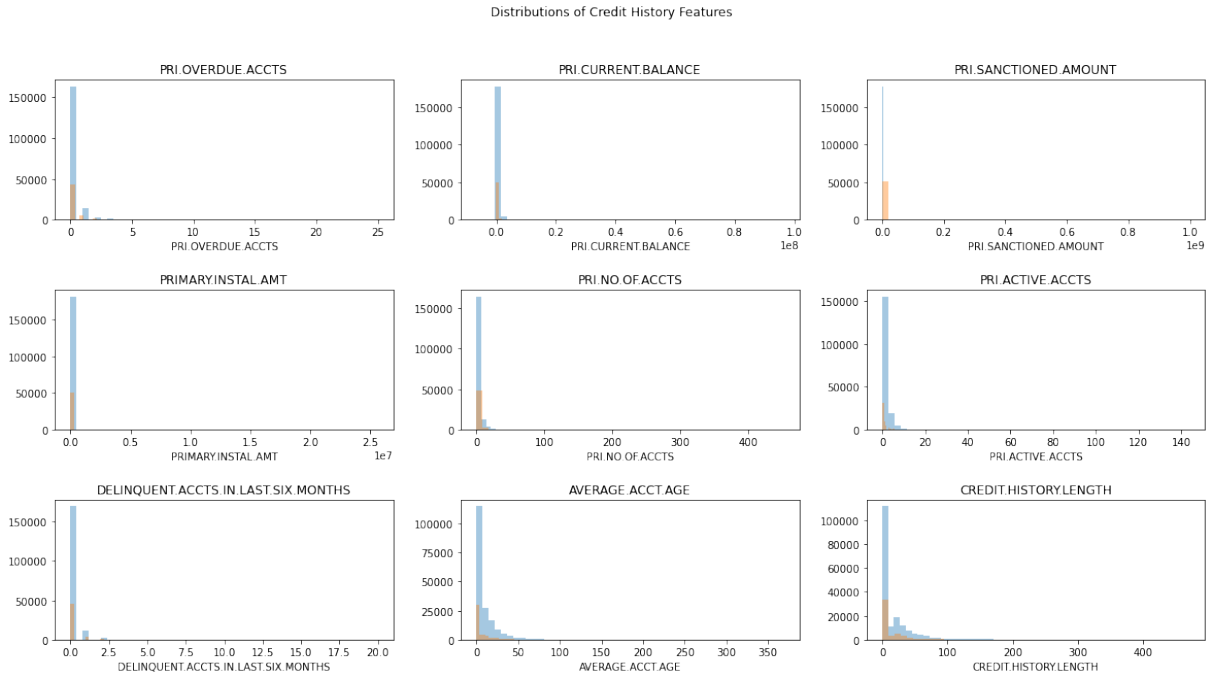

Figure 3: Distribution of features related to credit history

There are total 40 features in the dataset, 18 of them are the information about the loans they have before this disbursement and 14 of the 18 features are about the loanee's primary and secondary accounts. After doing some EDA, there are about 98% zero values in each of the features of secondary accounts. So it is worthier to look at the loan information features other than those feature of secondary account. The figure shows the distribution of features including count of total loans, count of active loans, count of default accounts, total Principal outstanding amount, total amount that was sanctioned and total amount that was disbursed in loanee's primary account, also the number of loans that the loanee defaulted in the last 6 months, the average loan tenure and the time since first loan. Form the figure, all of the graph are right-skewed distribution and 50% to 60% of data in each features have 0 values which means there are a lot of loanee applied for loans for the first time.

3 Method

3.1 Data Splitting and Preprocessing

From the exploratory data analysis step, 78.29% of data are marked as not defaulted and 21.71% are defaulted. Therefore, the dataset is imbalanced and stratified splitting technique should be used. Since there are total 255133 data in the dataset, splitting 20% of the data to test set using stratified splitting so that 23316 data are in the test set and it is large enough to access the model's performance.Then do 5 stratified k fold cross validation on the other 80% set so that there are 188855 data in the training set and 20983 data in the validation set. During each fold, fit and transform the training, then transform the test and validation set. In each set, there are 78.29% non-default data and 21.71% default data. By doing stratified k fold cross validation, the percentage of data in each class of target variable is constant and the k fold cross validation reduces the bias that may caused when randomly splitting the data. Each row of data in the dataset describes one loanee and there is no correlation between each loanee's data. So the dataset is independent and identically distributed and doesn't have time-series or group structure. With this property, the test and validation sets can approximately stimulate the real distribution of the dataset.

The only feature that has missing value is the Employment type, replace the nan value as new class "Unknown", since these missing value can represent that the loanee may not have job or does not will to tell their job, simply dropping the missing value can cause inaccuracy on the model performance.

From EDA, the features about secondary accounts are insignificant to the prediction of target variable but dropping secondary account information is unsafe for banks, so combine these features with the corresponding features for primary accounts and get a set of new features as total account features. Transform the 'Date of Birth' feature to the age of each loanee. Applied MinMaxEncoder on the age feature since it is clearly bounded. For other continuous features, since some of the features are about the amount of money and some of the features are about counts, applied the StandardScaler to make them have the same unit. For categorical features, the only ordered feature is applied PERFORM_CNS.SCORE.DESCRIPTION feature which represents the risk level of the loanee's credit from 'A-very low risk' to 'M-very high risk'. So applied OrdinalEncoder on this feature, transformed risk level from 'A-very low risk' to 'M-very high risk' to 1 to 13 and the 'No history available' class as 14. Applied onehotEncoder on other categorical features since all of the other categorical feature do not have any order in it. As a result, there are 21 features in the preprocessed dataset. Since the target variable is [0,1] in the original dataset, it dosen't need to be preprocessed.

3.2 Model Selection

Four different machine learning models were used: Logistic Regression model with l2 regularization, Random Forest Classifier, K-nearest Neighbors Classifier, and XGBoost Classifier. I chose f1 score to evaluate the performance of each model. Since the dataset is pretty imbalanced, just predicting all the data as class 0 can still get a good accuracy score, so choose f1 score as the evaluation metrics which take both false positives and false negatives into account can measure the performance more accurately. In the pipline, 5 random states are used in order to measure uncertainties that caused by splitting method and non-deterministic methods. For each random state, all models' hyperparameters are tuned by using GridSearchCV with 5 fold stratified cross validation so that the best performance for each model can be achieved. After tuning the hyperparemeter, the f1 score of each model with the best parameters on the holdout set will be stored in the array for comparison. Below is the table of the value of hyperparemeters each model tried in the pipline:

Models	Hyperparameters
L2	C :0.1,1,10,100,1000
RF	max_features: 5,10,20, max_depth:4,5,6
KNN	n_neighbors: 5,10,50,100,200, weights:uniform,distance
xgboost	max_depth:10,15,20,n_estimators:1000,1500,2000,min_child_weight:1,3,5,subsample:0.6,0.8,1

4 Result

4.1 Model performance

Below is the figure that shows the performance of each model in 5 random state.

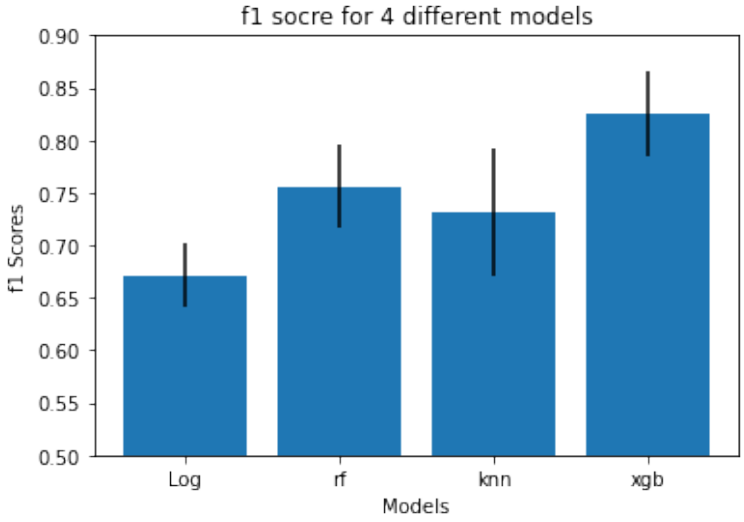

Figure 4: peformance of each model across 5 fold

The XGBoost Classifier gives the best f1 scores of 0.8255 and auc socre of 0.84. The baseline f1 score can be calculated by 2p/p+1 where p is calculated as the fraction of class 1. Over 5 random states, the baseline f1 score has a mean of 0.395 with standard deviation of 0.01. The best model gives an average f1 score of 0.825 with standard deviation of 0.04. So the model's score is 43 standard deviation above the baseline.

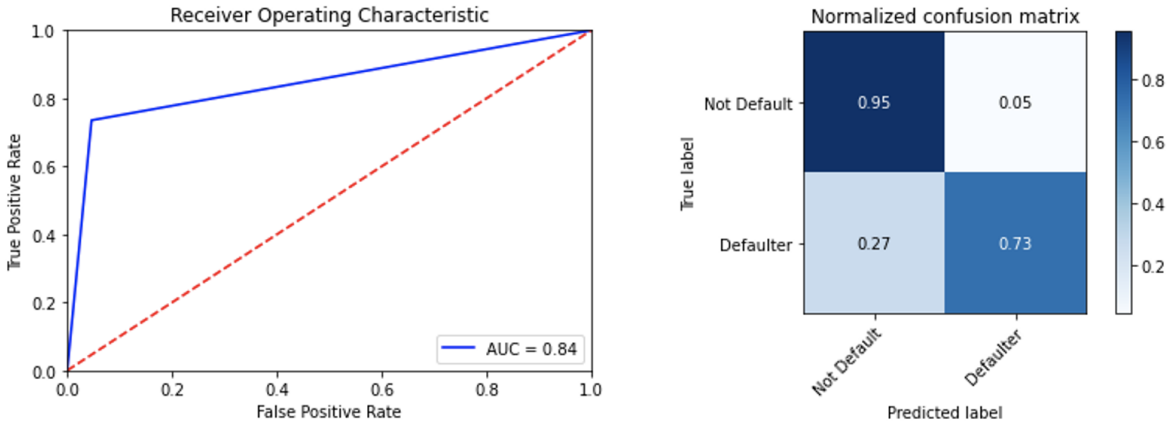

Figure 5: roc curve and confusion matrix

4.2 feature importance

For XGBoost,three different metrics are used to measure the feature importance which are weight,gain,cover. The local feature importance is calculated using Shap.

The most relevant feature according to gain and cover types is PERFORM_CNS.SCORE.DESCRIPTION feature. This is reasonable since this is the feature that directly show the credit level of the loanee.

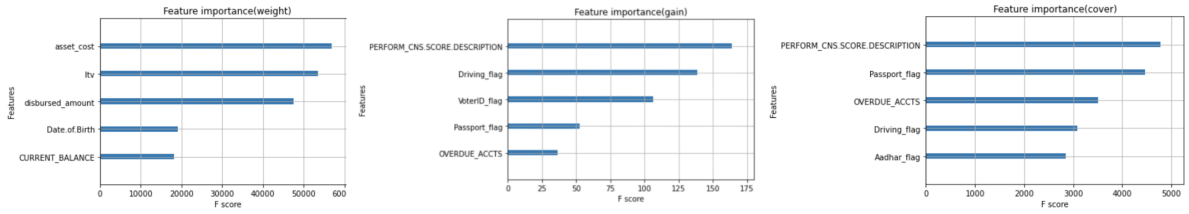

Figure 6: global feature importance

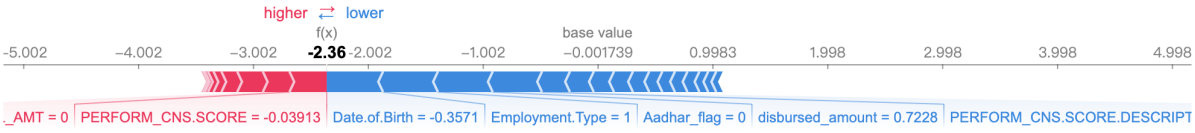

Figure 7: local feature importance

Since type weight is the percentage representing the relative number of times a particular feature occurs in the trees of the model. The most important feature is the cost of the asset which also make sense, since it is the one to measure the amount of loan. For the local feature importance, the feature that push the prediction higher is the age of the loanee and the one which push the prediction lower is the employment type. What surprised me is that the least important feature is total sanctioned amount which should represent how much the bank trust then to give then the loan. The possible reason is that this is the first time to take vehicle loan for many customers so that there is no record for them.

5 Outlook

When training the model, I only dropped the features that are obviously unrelated like the employee' id. However, after looking at the feature importance of the models, there are some features that has really low importance but seems important by the name like account's disbursal length. Dropping those features should lead to a better performance. Also, there are many customers in the dataset had 0 in many of the features like the number of active account,new accounts in last 6 month and so on. If we create a new feature called missing features to count how many features have 0 values for each customers, this feature might be able to improve the performance. In the dataset, only 20% of data are marked as defaulter. To solve this problem of imbalanced, the additional technique can be used is resampling like SMOTE() or use class weight to assign more weight to the class of defaulter. To improve the accuracy of the prediction, we can also collect the customer's loan history for other property like house.

6 Referrence

[1]: R. J. CROSS AND TONY DUTZIK FRONTIER GROUP (2019) The Hidden Costs of Risky Auto Loans to Consumers and Our Communities, ED MIERZWINSKI AND MATT CASALE U.S. PIRG EDUCATION FUND, p.8

[2]: Amjad Abu-Rmileh (2019) The Multiple faces of 'Feature importance' in XGBoost Towards Data Science

[3]: Kaggle: https://www.kaggle.com/mamtadhaker/lt-vehicle-loan-default-prediction