Yinan Guo

guoy221@wfu.edu

2021/12/08

# Executive Summary

## Problem

This task is to build machine learning models to help predict likely default loan for a major financial institution. In this case, we need to make sure the models are explainable and the predictions that we create can be explained.

## Key Findings

1. Customers with G grade has the largest rate of loan default, which has a default rate about 0.3, which is 2-times about average whereas default rate for customers with A grade is about 0.1, which is far below the average. Thus, grade is an important factor. With higher grade (like A, B), customers are less likely to make default loan.
2. Customers who make loans for small business has the largest rate of loan default, which has a default rate about 0.25, 1.5-times about average whereas default rate for customers with cars or weddings has default rate about 0.15, which is far below the average.
3. Customers who do not own a home has the greatest default rate, which is about 0.25, far beyond the average default rate of 0.15.

## Model Performance Summary & Interpretation

The performance for my final model is shown in the chart.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | AUC | Precision | Recall |
| XGBoost | 0.9750392 | 0.9919192 | 0.938 | 0.883 |

1. My final model is an XGBoots model with 1401 trees, min\_n of 14, tree\_depth of 14 and learn rate of 0.012312104.
2. This model has an AUC of 0.992, which shows that this model can tell right from wrong 99.2% of time. It has a precision of 0.938, and a recall of 0.883, which shows that it has a great trade-off between these two metrics. 93.8% of the records that are predicted as positive are actually positive, while 88.3% of the records that are actual positive are detected as positive.
3. The most important predictors are last payment day, last payment amount, last credit pull, member id and the number of payments on the loan.

## Recommendations

1. Since customers with low credit grade are more likely to commit fraud, financial institutions should strengthen their review when users with low credit ratings submit loan applications.
2. When the Customer’s loan reason is to do business, financial organizations should strengthen their assessment of their business conditions. It could be a good way to combine the credit score, home ownership condition and historical data of the lender to make the assessment.

# Detailed Analysis & Steps

### File(s) Summary

| **File Name** | **Record count** | **Column count** | **Numeric columns** | **Character columns** |
| --- | --- | --- | --- | --- |
| loan\_traini.csv | 29777 | 52 | 29 | 23 |
| loan\_holdout.csv | 12761 | 51 | 29 | 22 |

### Field Summary

For character variables:

| **Name** | **Data Type** | **Feature Type** | **Count** | **# Distinct** | **% Distinct** | **# Missing** | **% Missing** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| term | character | Categorical | 29774 | 2 | 0.01% | 3 | 0.01% |
| int\_rate | character | Categorical | 29774 | 390 | 1.31% | 3 | 0.01% |
| grade | character | Categorical | 29774 | 7 | 0.02% | 3 | 0.01% |
| sub\_grade | character | Categorical | 29774 | 35 | 0.12% | 3 | 0.01% |
| emp\_title | character | Categorical | 27960 | 22143 | 74.36% | 1817 | 6.10% |
| emp\_length | character | Categorical | 29774 | 12 | 0.04% | 3 | 0.01% |
| home\_ownership | character | Categorical | 29774 | 5 | 0.02% | 3 | 0.01% |
| verification\_status | character | Categorical | 29774 | 3 | 0.01% | 3 | 0.01% |
| issue\_d | character | Date | 29774 | 55 | 0.18% | 3 | 0.01% |
| loan\_status | character | Target | 29777 | 2 | 0.01% | 0 | 0.00% |
| pymnt\_plan | character | Categorical | 29774 | 2 | 0.01% | 3 | 0.01% |
| url | character | URL | 29774 | 29774 | 99.99% | 3 | 0.01% |
| desc | character | Text | 20345 | 20310 | 68.21% | 9432 | 31.68% |
| purpose | character | Categorical | 29774 | 14 | 0.05% | 3 | 0.01% |
| title | character | Categorical | 29764 | 15200 | 51.05% | 13 | 0.04% |
| zip\_code | character | Categorical | 29774 | 819 | 2.75% | 3 | 0.01% |
| addr\_state | character | Categorical | 29774 | 50 | 0.17% | 3 | 0.01% |
| earliest\_cr\_line | character | Date | 29754 | 516 | 1.73% | 23 | 0.08% |
| revol\_util | character | Categorical | 29710 | 1094 | 3.67% | 67 | 0.23% |
| last\_pymnt\_d | character | Date | 29710 | 106 | 0.36% | 67 | 0.23% |
| next\_pymnt\_d | character | Date | 2352 | 96 | 0.32% | 27425 | 92.10% |
| last\_credit\_pull\_d | character | Date | 29772 | 109 | 0.37% | 5 | 0.02% |
| application\_type | character | Categorical | 29774 | 1 | 0.00% | 3 | 0.01% |

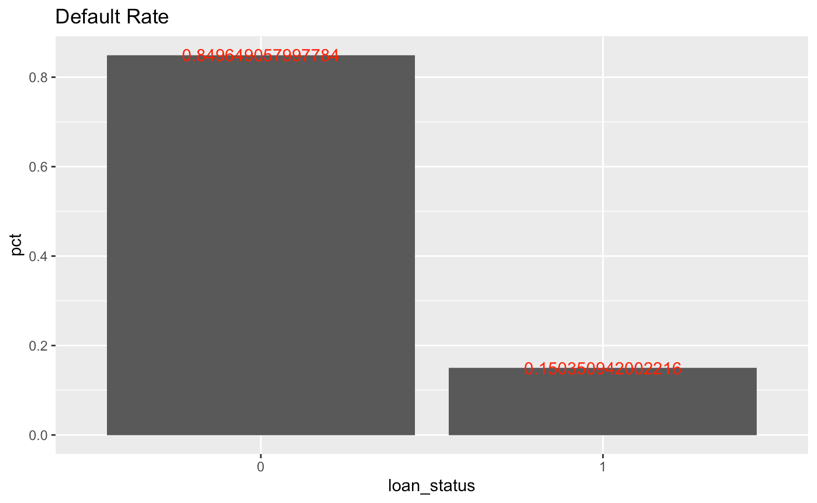
For numeric variables:

| **column** | **n** | **nmiss** | **mean** | **min** | **max** |
| --- | --- | --- | --- | --- | --- |
| id | 29774 | 3 | 663006.185 | 54734 | 1077501 |
| member\_id | 29774 | 3 | 823568.146 | 70473 | 1314167 |
| loan\_amnt | 29774 | 3 | 11109.4344 | 500 | 35000 |
| funded\_amnt | 29774 | 3 | 10843.6371 | 500 | 35000 |
| funded\_amnt\_inv | 29774 | 3 | 10149.6553 | 0 | 35000 |
| installment | 29774 | 3 | 323.808152 | 15.67 | 1305.19 |
| annual\_inc | 29773 | 4 | 69201.2323 | 2000 | 6000000 |
| dti | 29774 | 3 | 13.3840257 | 0 | 29.99 |
| delinq\_2yrs | 29754 | 23 | 0.15503798 | 0 | 13 |
| fico\_range\_low | 29774 | 3 | 713.053167 | 610 | 825 |
| fico\_range\_high | 29774 | 3 | 717.053167 | 614 | 829 |
| inq\_last\_6mths | 29754 | 23 | 1.08408953 | 0 | 33 |
| mths\_since\_last\_delinq | 10870 | 18907 | 34.7158234 | 0 | 120 |
| mths\_since\_last\_record | 2569 | 27208 | 59.2253795 | 0 | 129 |
| open\_acc | 29754 | 23 | 9.33901324 | 1 | 47 |
| pub\_rec | 29754 | 23 | 0.05854675 | 0 | 5 |
| revol\_bal | 29774 | 3 | 14310.0006 | 0 | 1207359 |
| total\_acc | 29754 | 23 | 22.0827788 | 1 | 81 |
| out\_prncp | 29774 | 3 | 11.7962884 | 0 | 3126.61 |
| out\_prncp\_inv | 29774 | 3 | 11.7643152 | 0 | 3123.44 |
| out\_prncp\_inv | 29774 | 3 | 11.7643152 | 0 | 3123.44 |
| total\_rec\_late\_fee | 29774 | 3 | 1.50478266 | 0 | 180.2 |
| last\_pymnt\_amnt | 29774 | 3 | 2615.40548 | 0 | 36115.2 |
| collections\_12\_mths\_ex\_med | 29673 | 104 | 0 | 0 | 0 |
| policy\_code | 29774 | 3 | 1 | 1 | 1 |
| acc\_now\_delinq | 29754 | 23 | 0.00013444 | 0 | 1 |
| chargeoff\_within\_12\_mths | 29673 | 104 | 0 | 0 | 0 |
| delinq\_amnt | 29754 | 23 | 0.20434227 | 0 | 6053 |
| pub\_rec\_bankruptcies | 28811 | 966 | 0.04532991 | 0 | 2 |
| tax\_liens | 29698 | 79 | 3.3672E-05 | 0 | 1 |
|  |  |  |  |  |  |

## Target Summary

* The default rate is 15.04%.

|  |  |  |
| --- | --- | --- |
| Default | n | Pct |
| No | 25300 | 0.8496491 |
| Yes | 4477 | 0.1503509 |



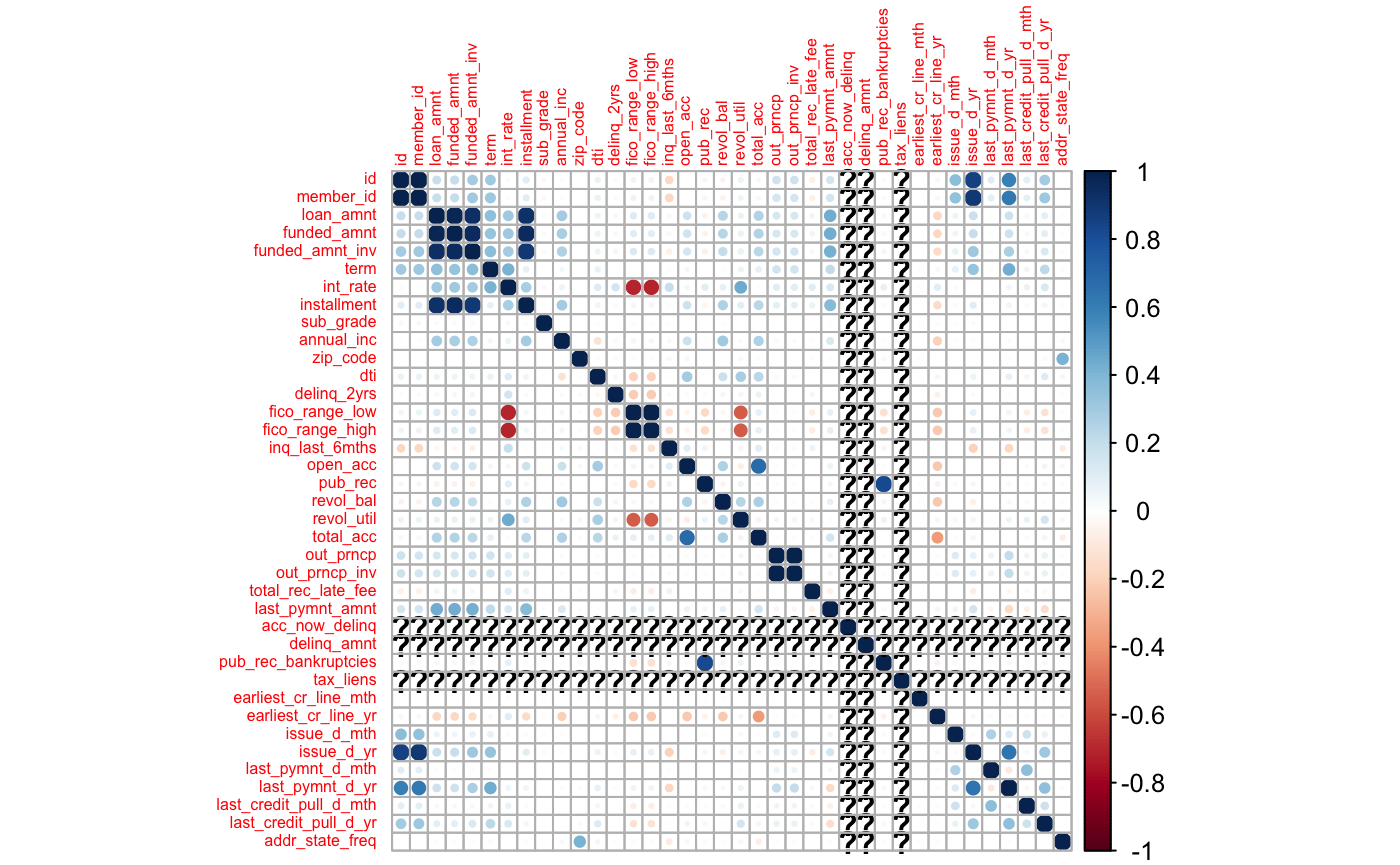
## Exploratory Data Analysis & Screening

### Descriptive Statistics

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Here are some samples for the boxplots of the numeric variables. From the graphs we can tell that funded\_amnt, int\_rate, zip\_code, fico\_range\_low, fico\_range\_high and revol\_till have different distributions within different groups divided by default or not, indicating that these may be important factors to affect loan default.

### Correlation Analysis



Among all the numeric variables, id is highly related with member id, loan amount is highly related to funded amount and installment, which is reasonable according to the common-sense. Also, fico range is negatively related to term.

### Frequency Analysis

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Here are some samples for the barplots of the character variables. According to the plots, if different columns in a graph have different proportions of blue part, which indicates the proportion of default, then the default rate is strongly correlated with this variable. (i.e. The barplot for grade shows a strong difference in proportion of default in different grades, indicating that the grade may affect loan default decisions for the customers.)

## Initial Screening & Exploration

|  |  |
| --- | --- |
|  |  |
| Customers with G grade has the largest rate of loan default, which has a default rate 2-times about average whereas default rate for customers with A grade is far below the average. Thus, grade is an important factor. | Customers who make loans for small business has the largest rate of loan default, which has a default rate 1.5-times about average whereas default rate for customers with cars or weddings is far below the average. Thus, loan purpose is an important factor. |
|  |  |
| The distribution of fico range in different groups are different and those who default the loan will have both lower low boundary and high boundary for the FICO range so that fico\_range\_high / fico\_range\_low are useful predictors | |
|  |  |
| The distribution for member id in different groups are almost the same so that it is not a useful predictor. | The distribution of revolving line utilization rate in different groups are different and those who default the loan have higher revol\_util so that it is a useful predictor. |

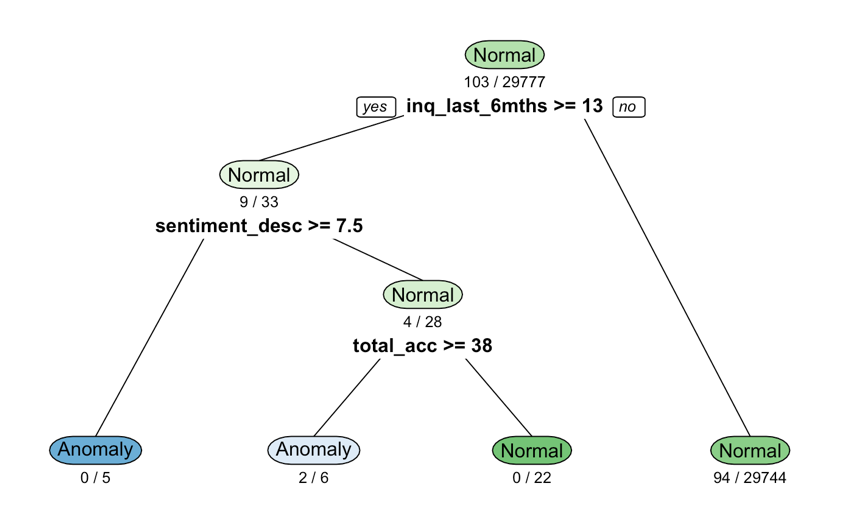
## Anomaly Detection

1. Choose an average\_depth

|  |  |
| --- | --- |
|  |  |

average\_depth is chosen to be 7.2 in this model. Accordingly, the anomaly score is around 0.615.

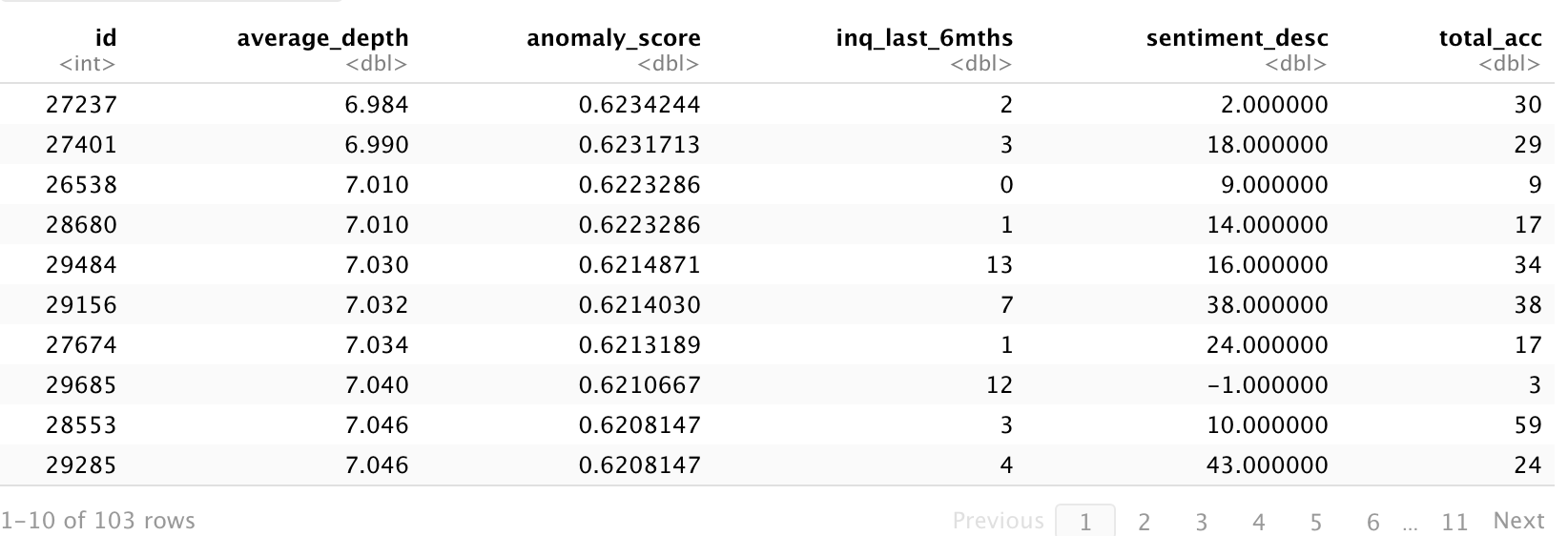
1. Plot the decision tree



The decision tree for anomaly detection rules is shown as above:

* When inq\_last\_6mths>=13, the anomaly coverage = 94/29744, which shows that it is more likely to be normal in this situation. (1)
* When inq\_last\_6mths<13 and sentiment\_desc<7.5, the amomaly coverage = 100%, which shows that we are almost sure that nodes in this situation are anomal. (2)
* When inq\_last\_6mths<13 and sentiment\_desc>=7.5 and total\_acc>=38, the amomaly coverage = 0%, which shows that we are almost sure that nodes in this situation are normal. (3)
* When inq\_last\_6mths<13 and sentiment\_desc>=7.5 and total\_acc<38, the amomaly coverage = 66.6%, which shows that it is more likely to be anormal in this situation. (4)

1. Explanation for records

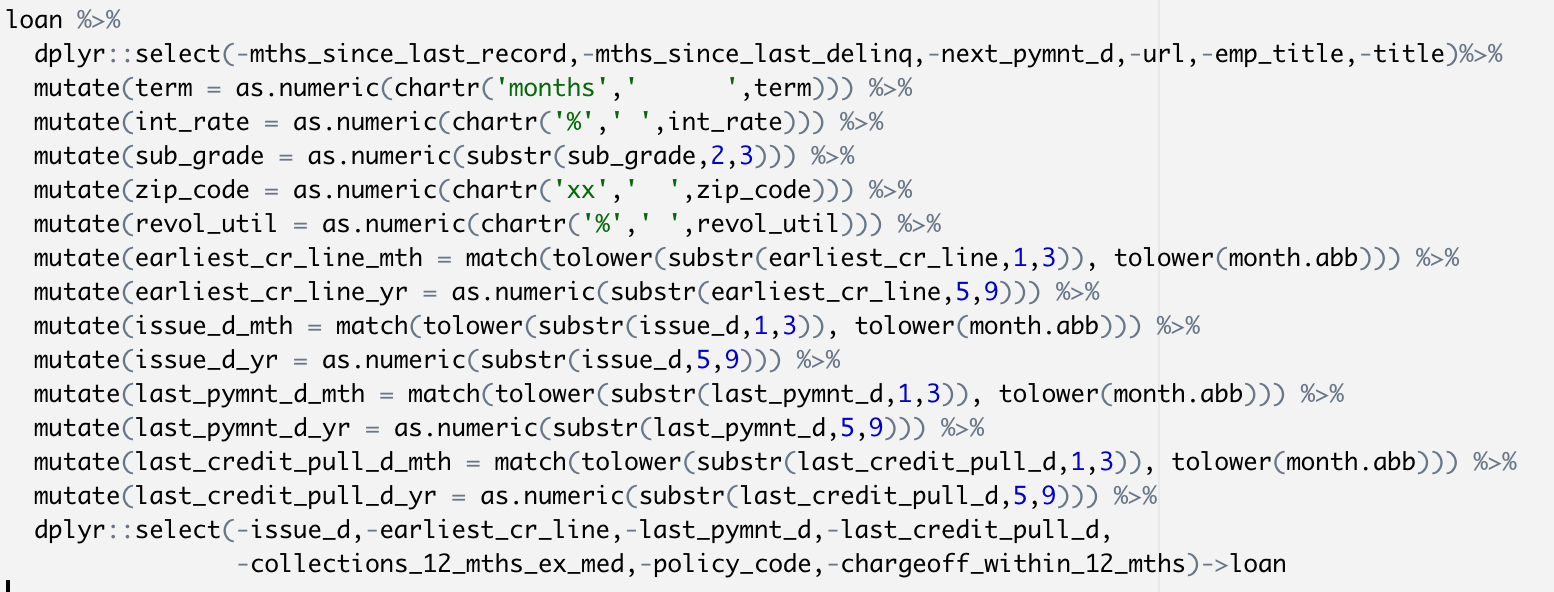


There are 103 records in training set defined as anomal.

* The first record on the picture has average\_depth of 6.984, which is lower than 7.2, with inq\_last\_6mths<13 and sentiment\_desc<7.5, which falls in 1st situation.
* The second record on the picture has average\_depth of 6.990, which is lower than 7.2, with inq\_last\_6mths<13 and sentiment\_desc>=7.5 and total\_acc<38, which falls in 4th situation.
* The third record on the picture has average\_depth of 7.010, which is lower than 7.2, with inq\_last\_6mths<13 and sentiment\_desc>=7.5 and total\_acc<38, which falls in 4th situation.
* The fourth record on the picture has average\_depth of 7.010, which is lower than 7.2, with inq\_last\_6mths<13 and sentiment\_desc>=7.5 and total\_acc<38, which falls in 4th situation.
* The fifth record on the picture has average\_depth of 7.030, which is lower than 7.2, with inq\_last\_6mths>=13, which falls in 1st situation.
* ……

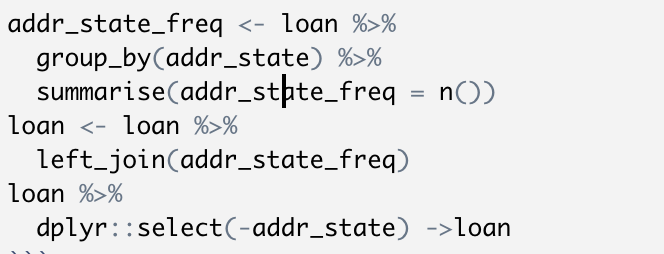
## Data Preparation & Transformation

1. Transfer some characters into useful numeric variables
2. Remove text from data and keep only numbers
3. Split the month-year data into months and years
4. Remove redundant variables



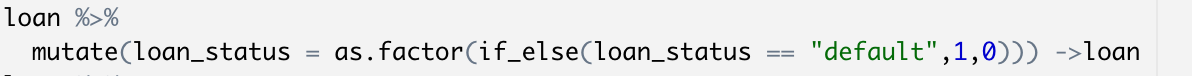
1. Frequency Encoding

Convert character variables with more than 10 unique values to their frequency in the data



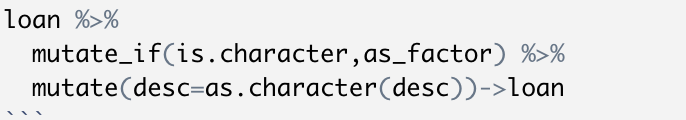
1. Target Encoding

Transfer loan status from current/default to 0/1.



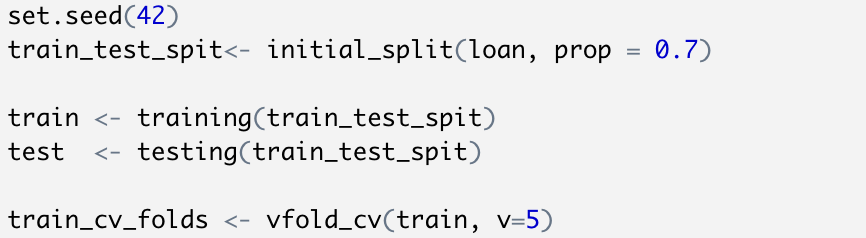
1. Transfer characters to factors

Remain the text variable for further analysing.



## Model Building

70% of data are used to build the model and 30% are used for validation. And 5 folds are splited for further tuning grid.

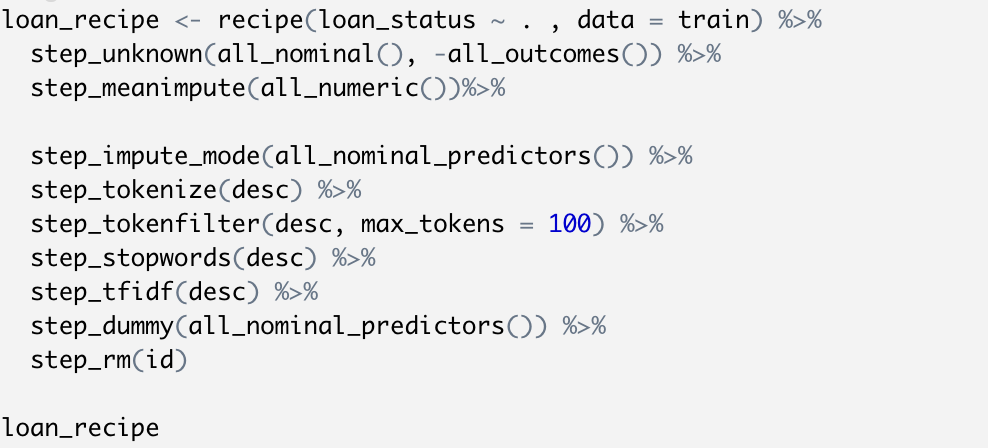


## Model Training

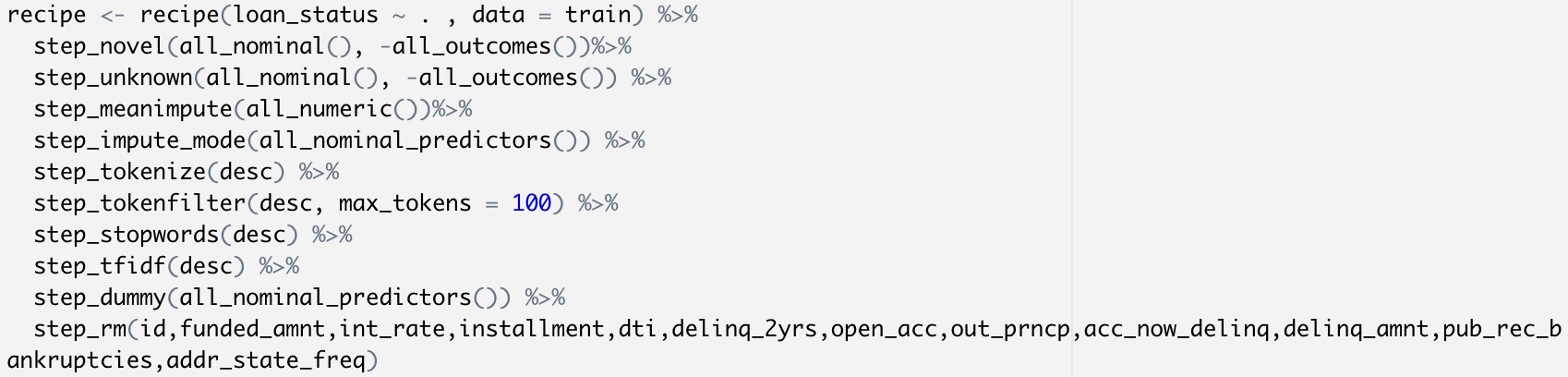
* Document the variables and the roles used

| **Name** | **Data Type** | **Random Forest / XGBoost** | **Reduced logistic** |
| --- | --- | --- | --- |
| loan\_status | character(target) | / | / |
| term | character | 1 | 1 |
| int\_rate | character | 1 | 0 |
| grade | character | 1 | 1 |
| sub\_grade | character | 1 | 1 |
| emp\_title | character | 0 | 0 |
| emp\_length | character | 1 | 1 |
| home\_ownership | character | 1 | 1 |
| verification\_status | character | 1 | 1 |
| issue\_d | character | 1 | 1 |
| pymnt\_plan | character | 1 | 1 |
| url | character | 0 | 0 |
| desc | character | 1 | 1 |
| purpose | character | 1 | 1 |
| title | character | 0 | 0 |
| zip\_code | character | 1 | 1 |
| addr\_state | character | 1 | 1 |
| earliest\_cr\_line | character | 1 | 1 |
| revol\_util | character | 1 | 1 |
| last\_pymnt\_d | character | 1 | 1 |
| next\_pymnt\_d | character | 0 | 0 |
| last\_credit\_pull\_d | character | 1 | 1 |
| application\_type | character | 1 | 1 |
| id | numeric | 0 | 0 |
| member\_id | numeric | 1 | 1 |
| loan\_amnt | numeric | 1 | 1 |
| funded\_amnt | numeric | 1 | 0 |
| funded\_amnt\_inv | numeric | 1 | 1 |
| installment | numeric | 1 | 0 |
| annual\_inc | numeric | 1 | 1 |
| dti | numeric | 1 | 0 |
| delinq\_2yrs | numeric | 1 | 0 |
| fico\_range\_low | numeric | 1 | 1 |
| fico\_range\_high | numeric | 1 | 1 |
| inq\_last\_6mths | numeric | 1 | 1 |
| mths\_since\_last\_delinq | numeric | 0 | 0 |
| mths\_since\_last\_record | numeric | 0 | 0 |
| open\_acc | numeric | 1 | 0 |
| pub\_rec | numeric | 1 | 0 |
| revol\_bal | numeric | 1 | 1 |
| total\_acc | numeric | 1 | 1 |
| out\_prncp | numeric | 1 | 0 |
| out\_prncp\_inv | numeric | 1 | 1 |
| total\_rec\_late\_fee | numeric | 1 | 1 |
| last\_pymnt\_amnt | numeric | 1 | 1 |
| collections\_12\_mths\_ex\_med | numeric | 1 | 1 |
| policy\_code | numeric | 1 | 1 |
| acc\_now\_delinq | numeric | 1 | 1 |
| chargeoff\_within\_12\_mths | numeric | 1 | 1 |
| delinq\_amnt | numeric | 1 | 1 |
| pub\_rec\_bankruptcies | numeric | 1 | 0 |
| tax\_liens | numeric | 1 | 1 |

* Since the random forest model and the XGBoost model are using the same recipe, there are only 2 recipes.
* The first recipe only removed some variables with only a few lines with different value but most of them are the same (i.e. mths\_since\_last\_delinq) and variables with more than 20% missing rate (i.e. next\_pymnt\_d) and some meaningless variables (i.e. url).
* The second recipe removed some variables that has p-value over 0.05 for logistic regression.
* Define Recipes
* Random Forest / XGBoost



* Reduced logistic



* Define your Model

Except what has been removed in the recipes, all of other predictors are used in the model.

The top 5 important factors for each model are:

|  |  |  |
| --- | --- | --- |
| Reduced logistic | Random Forest | XGBoost |
| Last payment day | Last payment amount | Last payment day |
| Last credit pull | Last payment day | Last payment amount |
| Last payment amount | Last credit pull | Last credit pull |
| Term (The number of payments on the loan.) | Total recorded late fee | Member id |
| Total recorded late fee | Member id | Term (The number of payments on the loan.) |

* Create a workflow and fit the model
* Evaluate metrics on Train and Test:
  + Classification:
    - table of Metrics:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | | AUC | | LogLoss | |
| **Train** | **Test** | **Train** | **Test** | **Train** | **Test** |
| Reduced logistic | 0.9520702 | 0.9463846 | 0.9761141 | 0.9727271 | 0.1338607 | 0.1415634 |
| Random Forest | 0.9876697 | 0.9581375 | 0.9996219 | 0.9818348 | 0.09392358 | 0.14858047 |
| XGBoost | 0.9931392 | 0.9750392 | 0.9993909 | 0.9919192 | 0.03127839 | 0.07166708 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Precision | | Recall | | F1 | |
| **Train** | **Test** | **Train** | **Test** | **Train** | **Test** |
| Reduced logistic | 0.868 | 0.893 | 0.791 | 0.734 | 0.828 | 0.806 |
| Random Forest | 0.996 | 0.931 | 0.928 | 0.781 | 0.961 | 0.849 |
| XGBoost | 0.965 | 0.938 | 0.930 | 0.883 | 0.947 | 0.910 |

* + - ROC comparing trains and tests

|  |  |  |
| --- | --- | --- |
| Reduced logistic | Random Forest | XGBoost |
|  |  |  |

* + - Confusion matrix comparing train and test

|  |  |
| --- | --- |
| Train | Test |
| Reduced Logistic | |
|  |  |
| Random Forest | |
|  |  |
| XGBoost | |
|  |  |

* + - Test Score Threshold
* KS for models:

According to KS, the best operational threshold for

1. logistic model is 0.11, where fpr is 0.06 and tpr is 0.925;
2. random forest model is 0.255, where fpr is 0.05 and tpr is 0.925;
3. xgboost model is 0.1, where fpr is 0.03 and tpr is 0.962

|  |
| --- |
| Reduced Logistic |
|  |
| Random Forest |
|  |
| XGBoost |
|  |

* Roc Chart w. Score Threshold

|  |  |
| --- | --- |
| Reduced Logistic | Random Forest |
|  |  |
| XGBoost |  |
|  |  |

* Recall/Precision/Threshold

Given the best operational thresholds, we can find related recall and precision.

1. For logistic model, the recall is 0.92, which means 92% of actual positive records are predicted as positive. The precision is 0.749, which means 74.9% of records predicted as positive are actually positive. And the F1 score is 0.826 at this threshold.
2. For random forest model, the recall is 0.92, which means 92% of actual positive records are predicted as positive. The precision is 0.771, which means 77.1% of records predicted as positive are actually positive. And the F1 score is 0.839 at this threshold.
3. For xgboost model, the recall is 0.96, which means 96% of actual positive records are predicted as positive. The precision is 0.771, which means 85.3% of records predicted as positive are actually positive. And the F1 score is 0.903 at this threshold.

|  |  |
| --- | --- |
| Reduced logistic | Random Forest |
|  |  |
| XGBoost |  |
|  |  |

* + - chart of variable importance
* Top 10 important variables for models

|  |  |
| --- | --- |
| Reduced Logistic | Random Forest |
|  |  |
| XGBoost |  |
|  |  |

* + - score distributions for test dataset

|  |  |
| --- | --- |
| Reduced Logistic | Random Forest |
|  |  |
| XGBoost |  |
|  |  |

## Model Comparison

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | | AUC | | LogLoss | |
| **Train** | **Test** | **Train** | **Test** | **Train** | **Test** |
| Reduced logistic | 0.9520702 | 0.9463846 | 0.9761141 | 0.9727271 | 0.1338607 | 0.1415634 |
| Random Forest | 0.9876697 | 0.9581375 | 0.9996219 | 0.9818348 | 0.09392358 | 0.14858047 |
| XGBoost | 0.9931392 | 0.9750392 | 0.9993909 | 0.9919192 | 0.03127839 | 0.07166708 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Precision | | Recall | | F1 | |
| **Train** | **Test** | **Train** | **Test** | **Train** | **Test** |
| Reduced logistic | 0.868 | 0.893 | 0.791 | 0.734 | 0.828 | 0.806 |
| Random Forest | 0.996 | 0.931 | 0.928 | 0.781 | 0.961 | 0.849 |
| XGBoost | 0.965 | 0.938 | 0.930 | 0.883 | 0.947 | 0.910 |

* Compare and contrast 3 models

Metrics are shown as above.

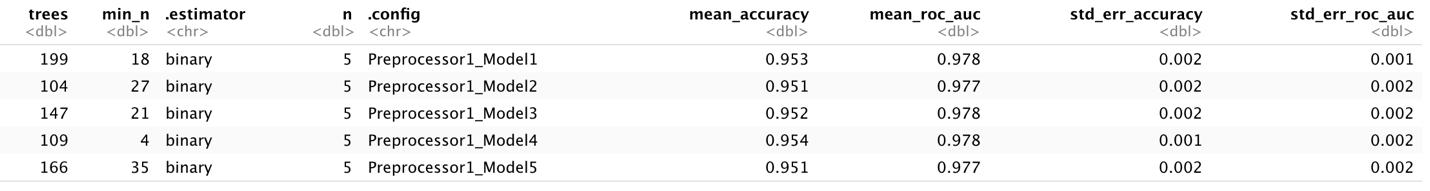
* Accuracy is the rate for (TP+TN)/(TP+TN+FP+FN)
* Precision is the rate for TP/(TP+FP), which represents the positive predictive value and important to a model.
* Recall is the rate for TP/(TP+FN), which represents the true positive rate and important to a model.
* AUC is the area under the ROC.
* Log Loss is the negative average of the log of corrected predicted probabilities for each instance. The lower, the better.
* F1 equals to 2\*precision\*recall/(precision+recall), which proves accuracy for the model.
* Which model performs better and why?

XGBoost model performs the best.

1. XGBoost model has the largest accuracy, auc, precision, recall and F1, which shows that this model can tell right from wrong in most of the time. Also, it has a really low log loss, which means that this model rarely gives out wrong result.
2. The difference on accuracy and AUC is relatively small, which means that it doesn’t have severe overfitting problem.

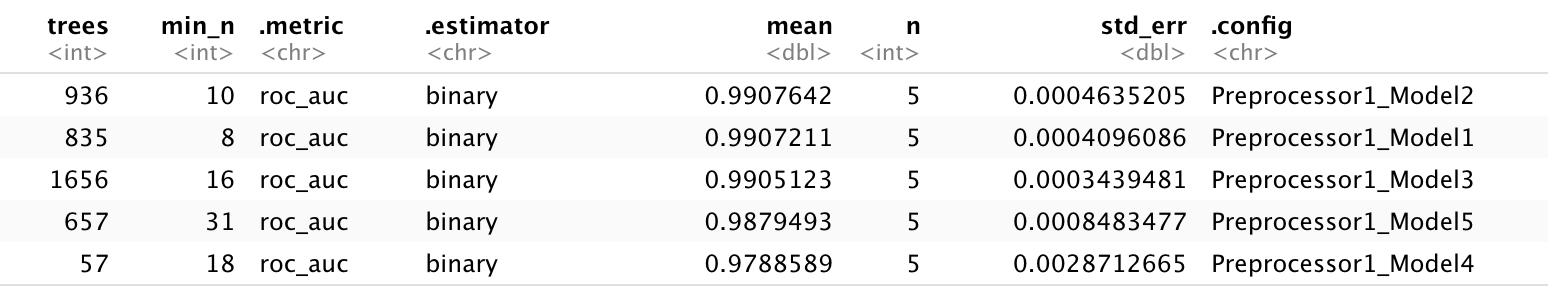
## Comparing tuning performance of RF and XGB models

* Random Forest



Tune grid randomly selected tree numbers between 100 and 200, min\_n between 5 and 30 and generated 5 models with different tunning parameters. The best random forest model is model 1, which gives out a mean\_accuracy of 0.953 and a mean\_roc\_auc of 0.978.

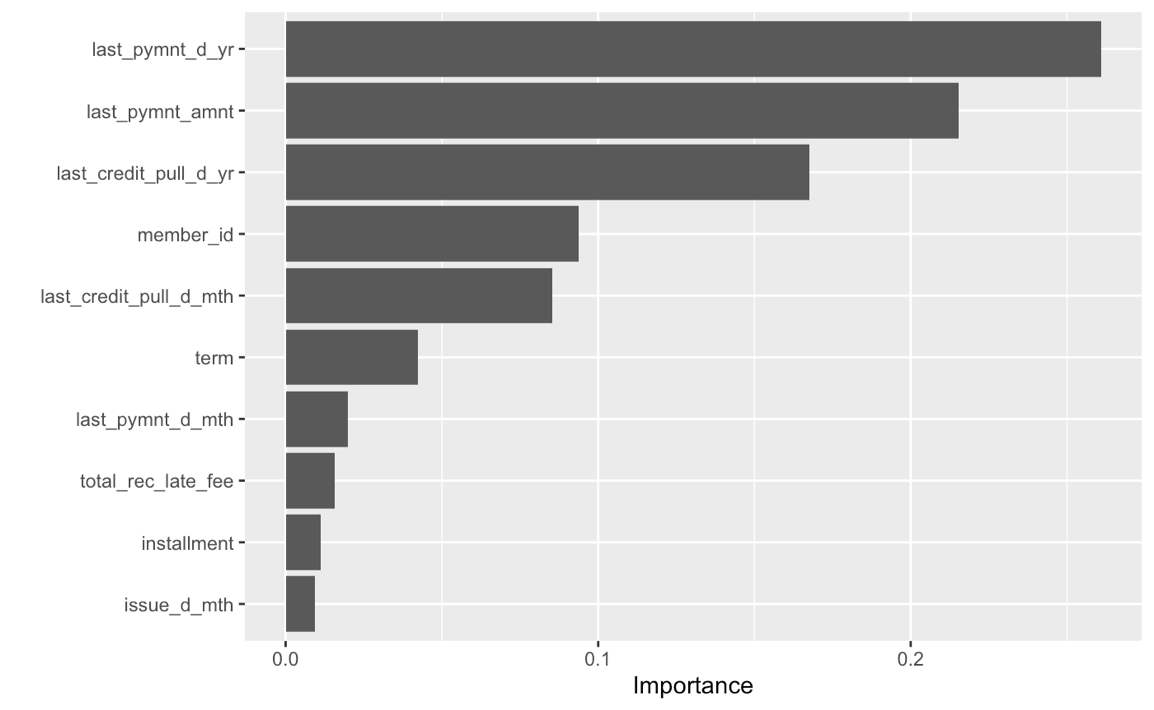
* XGBoost



Tune grid randomly selected tree numbers between 100 and 2000, min\_n between 5 and 50, and generated 5 models with different tunning parameters. The best XGBoost model is model 1, which gives out a mean\_roc\_auc of 0.9908.

## Explanation for my best model:

### Variance Importance



From top to bottom are top 10 important predictors to the model, which is last payment day, last payment amount, member id (which is related to sign in time), last credit pull day, term, total recorded late fee, installment and issue day.

### Partial dependency plot of top variables

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

The partial dependence profile plots show how predicted results would change if an independent predictor changes. For example, when the last payment year is 2008, the average probability for records to be 1 is 0.35, when it changes to 2016, the average probability for records to be 1 becomes 0.05.

### DALEX

|  |  |  |
| --- | --- | --- |
| TP | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| FP | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **FN** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Here are DALEX for top 10 records for TP, FP, FN where the absolute value of contribution is over 0.01. The DALEX plots show how predictors impact prediction for single record. When the contribution is positive, it shows that the predictor positively impacts the predicted result while when it is negative, it shows that the predictor negatively impacts the predicted result. The absolute value of contributor shows how much the predictor influences the prediction.

1. For example, the first plot for TP shows that last credit pull is incredibly important to this prediction since it has contribution of 0.33, which is high. Also, last payment amount, term, member id has contributed much to this result. And all of these predictors have positively affected this prediction.
2. For another example, the first plot for FN shows that last payment day has great positive impact on this prediction. Also, the member id, annual income, sentiment for desc and some other predictors have negative impact that prevent this prediction to be this result.