Data Analytics
Final Project Report
—— Fannie Mae Single-Family Loan Default

Baiting Gai

661991250

Abstract and Introduction

As there are more financial constraints in the ever-lasting financial environment, loan has been one of the most important products in the banking service. Banks are seeking effective and efficient ways to help their customers get the most satisfied loans. However, some customers will perform bad and even make loan default when their loans are approved. So, we need to find ways to minimize the loss that loan default causes. Machine learning is widely used in the banking industry to predict borrower default and it tends to work very well. In this paper, I will establish classification models to predict whether the borrower will default the loan and find important factors that affect the result most.

At the beginning, I made some hypotheses about which factors will affect the loan default. I assume that whether the borrower will default the loan is related to the credit score, the DTI(debt-to-income) ratio and the borrower's location. To testify these hypotheses, I find the Fannie Mae Single-Family loan data that includes all the information I want and I will use the data to conduct data modeling.

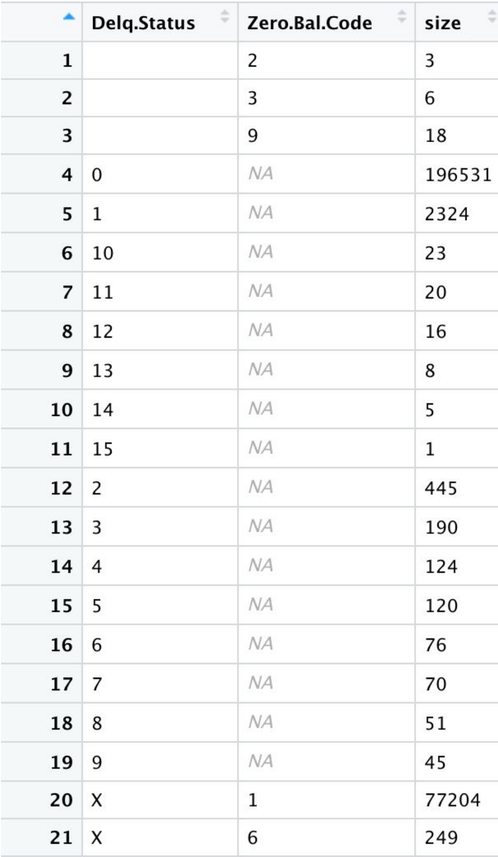
Fannie Mae is a government-sponsored enterprise founded in 1938 by Congress during the Great Depression as part of the New Deal. The aim of Fannie Mae is to stimulate the housing market by providing more mortgages to low-income and single-family borrowers. The organization is reliable, as well as the data source. In this report, I analyzed factors that may affect loan default based on the Fannie Mae single-family loan data in 2019 Q1 and gave some actionable recommendations.

Data Description and Exploratory Data Analytics

Fannie Mae provides loan performance data on a portion of its single-family mortgage loans to promote better understanding of the credit performance of Fannie Mae mortgage loans. The Fannie Mae single-family loan data I used contains two parts: 'Acquisition' data and 'Performance' data. The 'Acquisition' file includes static mortgage loan information like loan id, organization channel, seller name, original amount, original interest rate, loan purpose, credit score, etc. at the time of the mortgage loan origination and delivery to Fannie Mae. The 'Performance' file provides monthly performance data for each loan like maturity date, delinquency status, zero balance code, etc. from acquisition up until its current status as of the previous quarter.

The data are in 'txt' format, so I named the columns and set the type of each feature. There are 25 features in the 'Acquisition' data and 31 features in the 'Performance' data. In this research, I combined the two dataset according to the loan id so that we can see how each loan performs and whether there is a loan default. After combing the data, I found each loan has not only one record since there will be many performance records for one loan. I deleted the duplicate records and just kept the latest one for the following analysis.

The aim of the project is to predict loan default. Since there is not a very specific column that indicates whether the loan defaults, I need to define what kind of loans are classified as defaulted loans. I defined loans that have more than 1 month delinquency may lead to a default, otherwise the loans are classified as healthy loans. However, there are some loans that have an ‘unknown’ delinquency status. To determine what class these loans belong to, I checked the relationship between loan delinquency status and zero balance code indicating the reason the mortgage loan's balance was reduced to zero. I found the loans that have an ‘unknown’ delinquency status are most prepaid and matured (zero balance code = ‘01’) and the rest are repurchased (Plot 1). So, these loans are classified as healthy loans.



	Delq.Status	Zero.Bal.Code	size
1		2	3
2		3	6
3		9	18
4	0	NA	196531
5	1	NA	2324
6	10	NA	23
7	11	NA	20
8	12	NA	16
9	13	NA	8
10	14	NA	5
11	15	NA	1
12	2	NA	445
13	3	NA	190
14	4	NA	124
15	5	NA	120
16	6	NA	76
17	7	NA	70
18	8	NA	51
19	9	NA	45
20	X	1	77204
21	X	6	249

Plot 1: Relationship between delinquency status and zero balance code

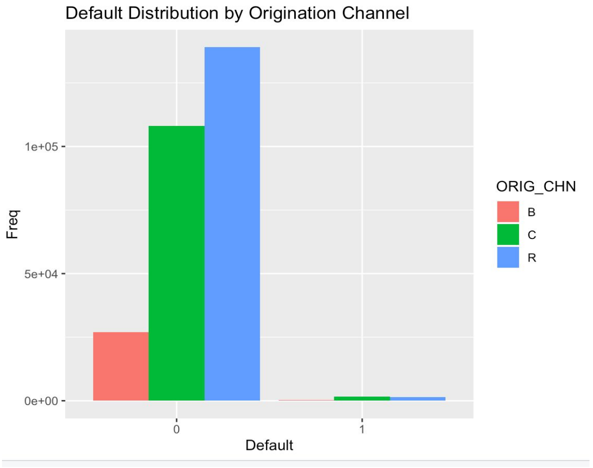
In addition to defining the ‘default’ loans, I also checked each column in the combined data and I found not all columns are relevant to the research. So, I just keep the columns including delinquency status, disposition date and zero balance code in the ‘Performance’ data and all information in the ‘Acquisition’ data. Besides, I combined and created some features based on the original dataset. I calculated the original home value according to the original loan amount and original loan-to-value. I also calculated the minimum credit score using the borrower’s and co-borrower’s credit score.

The challenge in the dataset is that the dataset is very unbalanced (0.01%). I used resampling methods to tackle the problem and I will explain more in the modeling part.

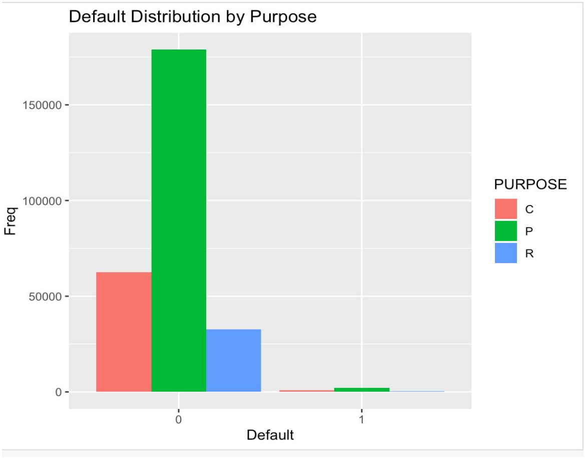
Analysis

After combining two datasets, defining the ‘default’ loans, selecting relevant features and creating some useful features, I did some exploratory analysis to check how each categorical and numeric will affect the ‘default’ value.

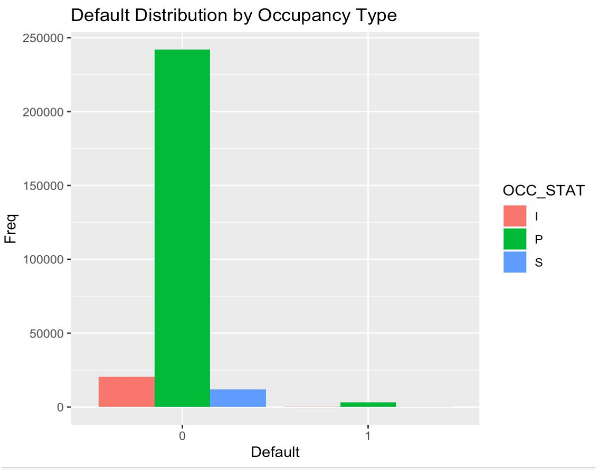
First, I analyzed how categorical variables including origination channel, seller name, whether the borrower is the first time home buyer, loan purpose, property type, occupancy status, whether the loan is a relocation mortgage and state affect loan default. From Plot 2, we can see that most loans come from R but loans from C have the highest default rate. In terms of loan purpose, loans based on purpose like purchase, cash-out refinance and no cash-out refinance almost show the same default rate (Plot 3). The default rate of loans that are secured by a principal residence, second home or investment property are almost the same. In Plot 4, we can see how default rate distributes in different states. LA has the most default rate and followed by MS. The default rate difference between each state indicates that state will be a relatively important feature in the classification model.



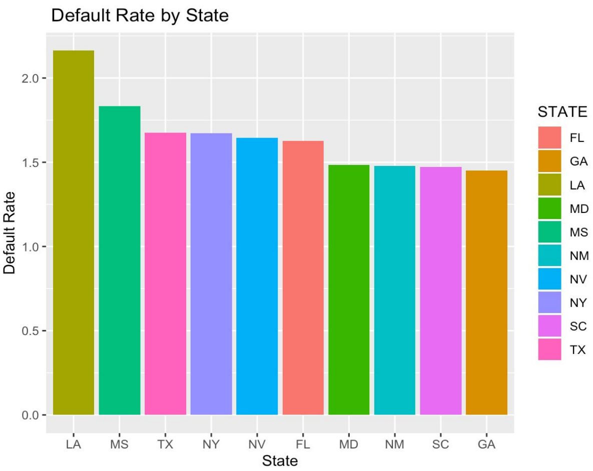
Plot 2: Default Distribution by Origination Channel



Plot 3: Default Distribution by Purpose

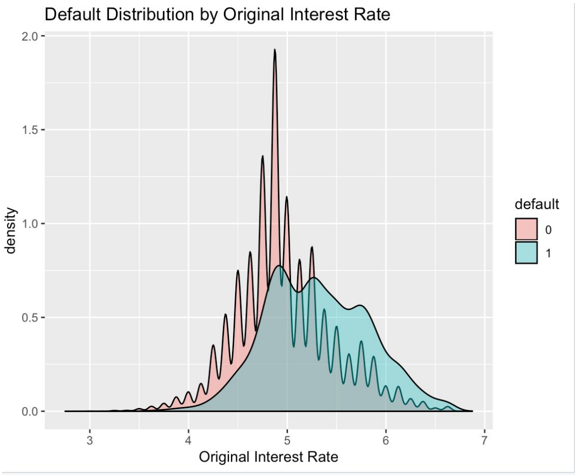


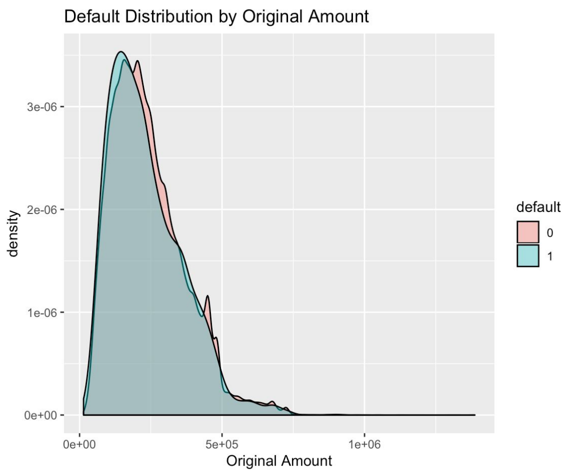
Plot 4: Default Distribution by Occupancy Type



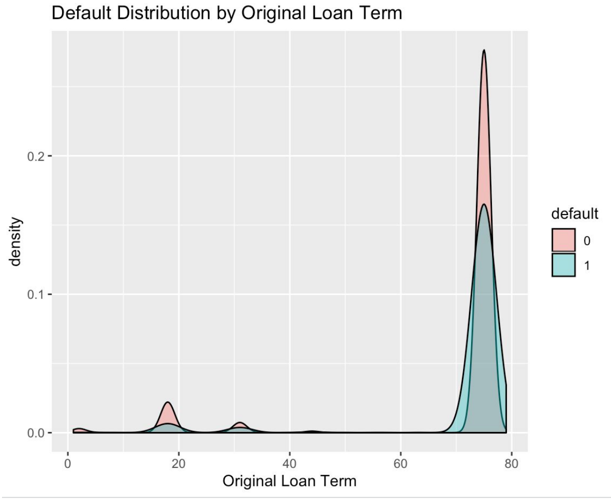
Plot 5: Default Rate by State

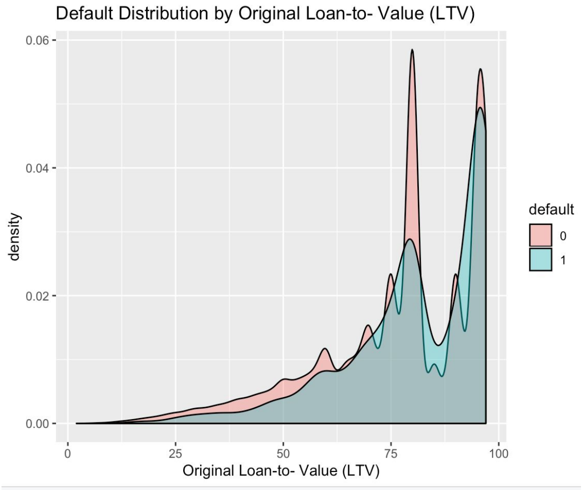
Second, I analyzed how numeric variables including original interest rate, original amony, original loan term, original loan-to-value (OLTV), debt-to-income rate (DTI), original value and credit score affect loan default. From the plots below, we can see that there is not too much difference between health loans and default loans in terms of original amount, original value, original loan term and OLTV, (Plot 7, Plot 8, Plot 9 and Plot 11). By contrast, original interest rate, DTI and credit score have an effect in classifying default loans and healthy loans. In terms of original interest rate, when the interest rate is higher than 5.3%, loan default is likely to happen (Plot 6). In terms of DTI, we can see that when DTI has achieved 35%, loans are more likely default loans (Plot 10). The difference of default loans and healthy loans show the most in the aspect of credit score. The distributions are totally different (Plot 12). Loans with a credit score smaller than 705 are more likely loan defaults. This is a very important insight from preliminary analysis and I will pay attention to the feature when I establish models.



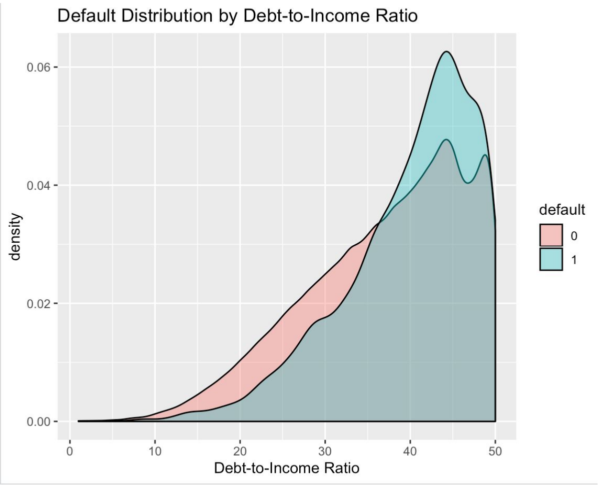


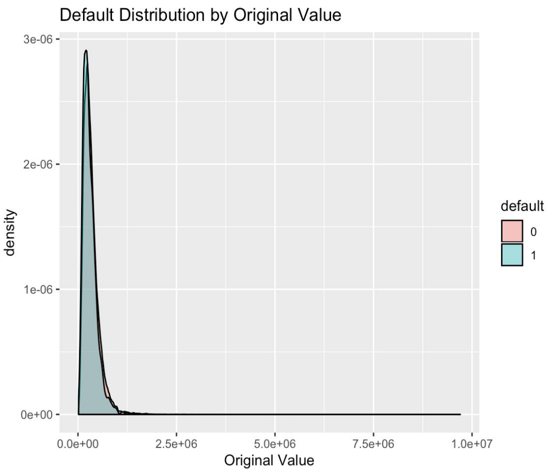
Plot 6: Default Distribution by Original Interest Rate Plot 7: Default Distribution by Original Amount





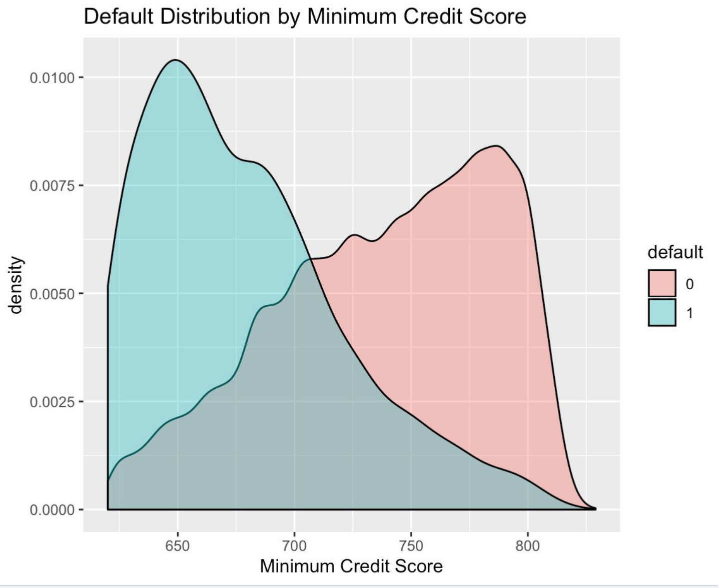
Plot 8: Default Distribution by Original Loan Term Plot 9: Default Distribution by OLTV





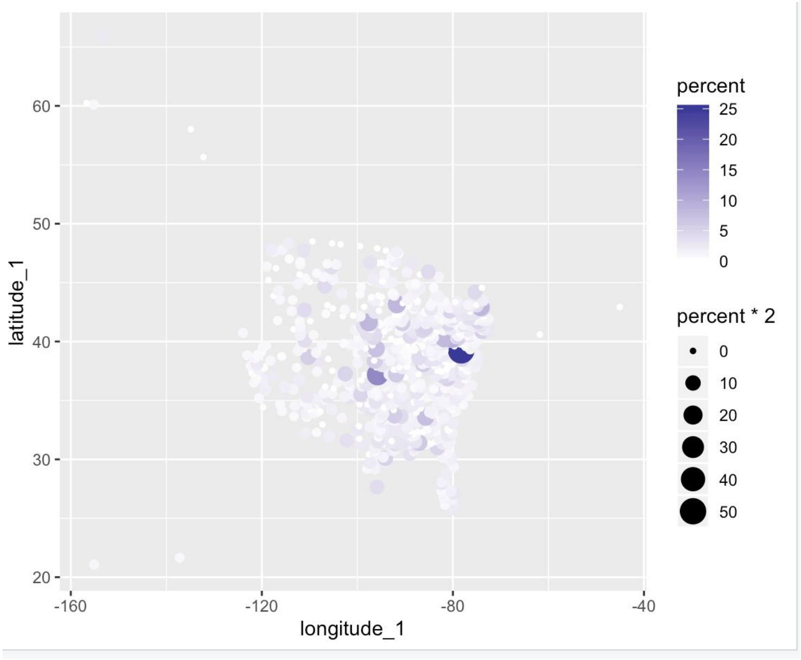
Plot 10: Default Distribution by DTI

Plot 11: Default Distribution by Original Value



Plot 12: Default Distribution by Minimum Credit Score

In addition to analyzing these variables, I also explored the geographic location. The data provides the first three digit zip code of each loan, but we do not know the specific location (longitude and latitude). I searched the corresponding longitude and latitude of each location online and calculated the average longitude and latitude with zip codes that have the same first three digit code. Then, I combined the manipulated data with our dataset and drew a scatter plot with longitude and latitude at the two axises (Plot 13). In this plot, each point represents the default rate of each same zip first three digit code. The deeper the color is and the bigger the point is, the default rate is higher. The plots show that the location with longitude of about -93 and latitude of about 35 and point with longitude of about-80 and latitude of about 40 have higher default rate.



Plot 13: Default rate Geographic location

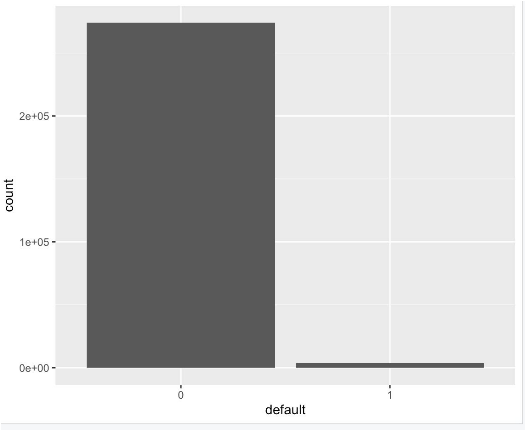
Model Development and Application of model(s)

Before modeling, I made some data transformation based on the analysis in the last part. From the distributions of original value, original amount, OLTV, OC LTV and DTI, we can see that the data are not normally distributed. Therefore, I made log transformations on these features. In addition to transformation, I also create dummy variables for some important categorical variables including origination channel, number of borrowers, loan purpose, property type, state, relocation flag and occupancy status. Finally, I dropped some useless raw columns that have been transformed to new columns.

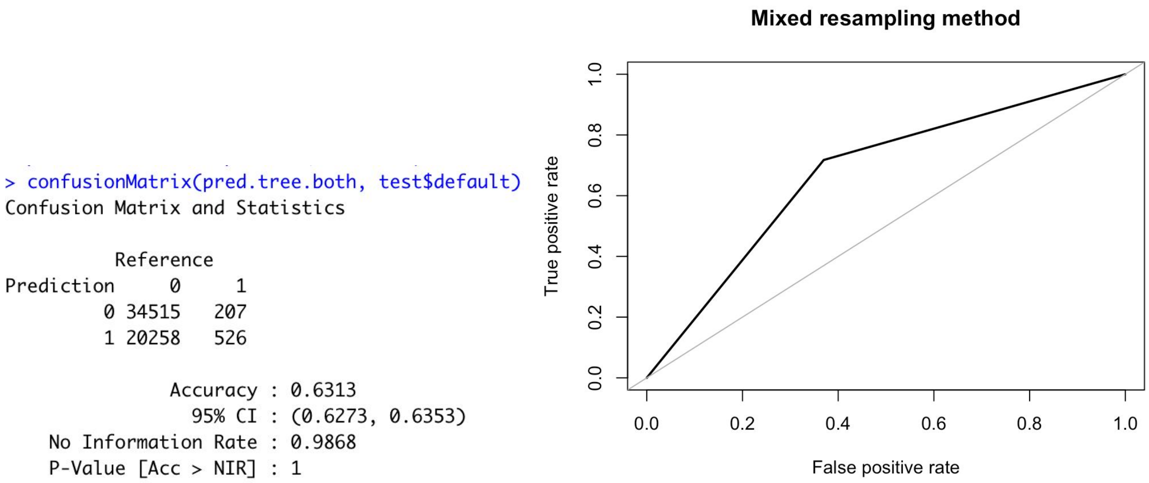
To avoid overfitting, I splitted the whole dataset into train and test dataset with a proportion of 8:2. I used train data to establish models and used test data to testify the model performance and optimize models.

Decision Tree

I first tried to establish a decision tree model with the dataset, but the accuracy is very low and the data are most misclassified. I found the reason behind the bad performance is the data is very unbalanced (0.01% minority of the data) (Plot 14). Therefore, I used resampling methods to tackle the problem. I tried oversampling, undersampling and mixed methods. Oversampling refers to duplicating samples from the minority class and undersampling refers to deleting samples from the majority class. From Plot 15, 16 and 17, we can see that the undersampling method performs well whether in confusion matrix and ROC curve. The decision tree accuracy using the undersampling data achieved 85%, which is pretty good. So, I will use the undersampled data to train the model and still use the test data to validate the model.

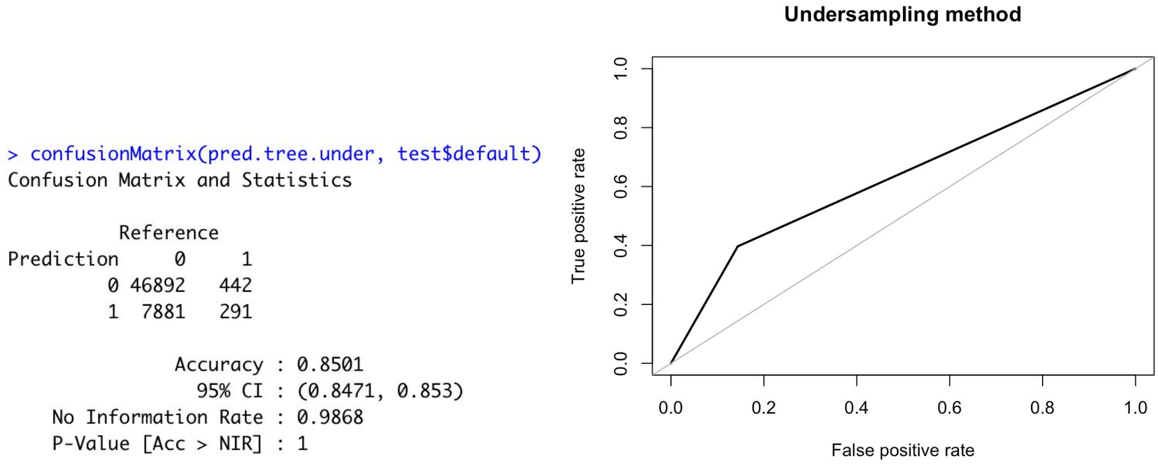


Plot 14: Imbalanced Data Barplot



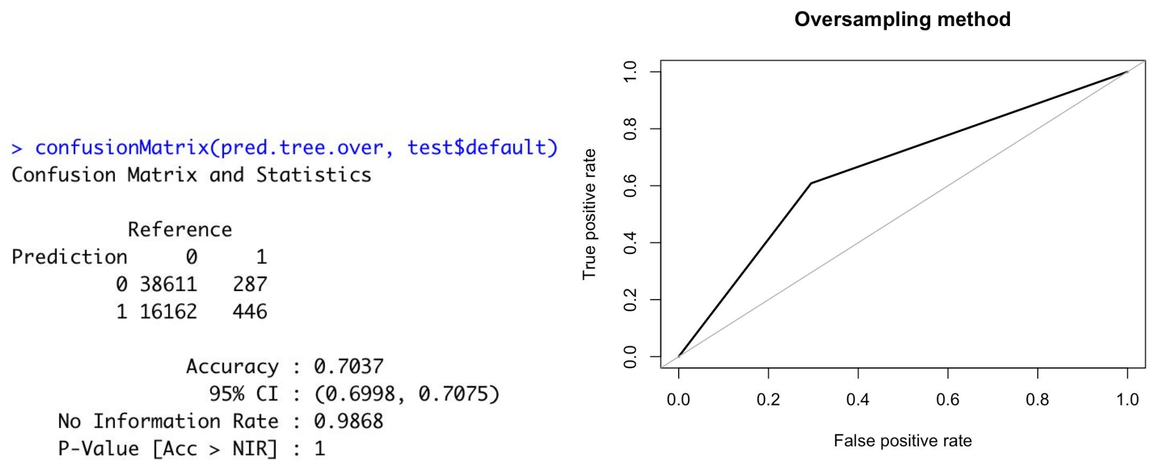
```
> confusionMatrix(pred.tree.both, test$default)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 34515   207
         1 20258   526

               Accuracy : 0.6313
                 95% CI : (0.6273, 0.6353)
    No Information Rate : 0.9868
    P-Value [Acc > NIR] : 1
```

Plot 15: Decision Tree Confusion matrix and ROC curve based on mixed resampling method



```
> confusionMatrix(pred.tree.under, test$default)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 46892   442
         1  7881   291

               Accuracy : 0.8501
                 95% CI : (0.8471, 0.853)
    No Information Rate : 0.9868
    P-Value [Acc > NIR] : 1
```

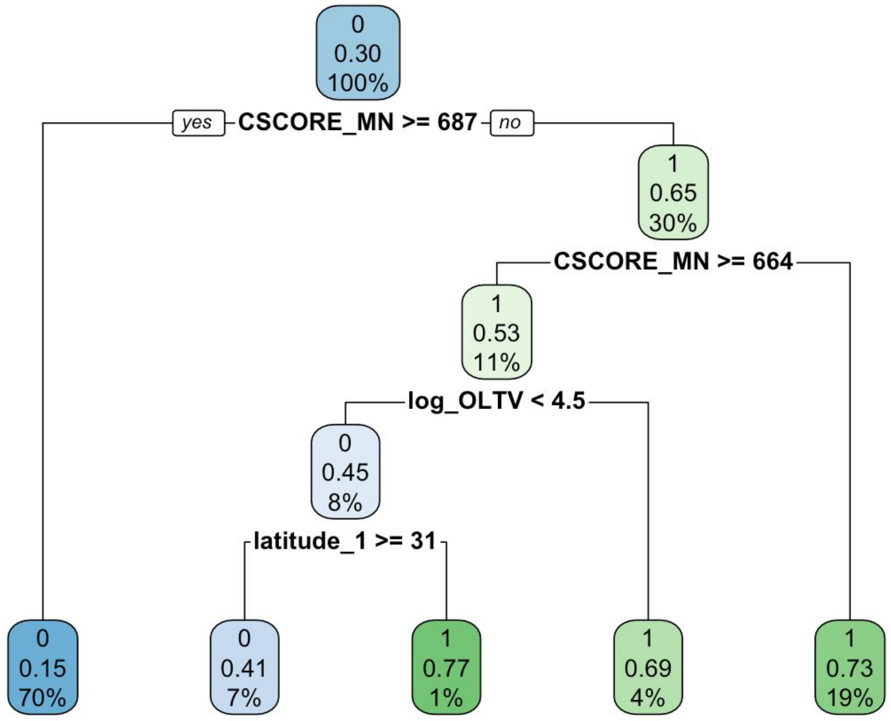
Plot 16: Decision Tree Confusion matrix and ROC curve based on undersampling method



```
> confusionMatrix(pred.tree.over, test$default)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 38611   287
         1 16162   446

               Accuracy : 0.7037
                 95% CI : (0.6998, 0.7075)
    No Information Rate : 0.9868
    P-Value [Acc > NIR] : 1
```

Plot 17: Decision Tree Confusion matrix and ROC curve based on oversampling method

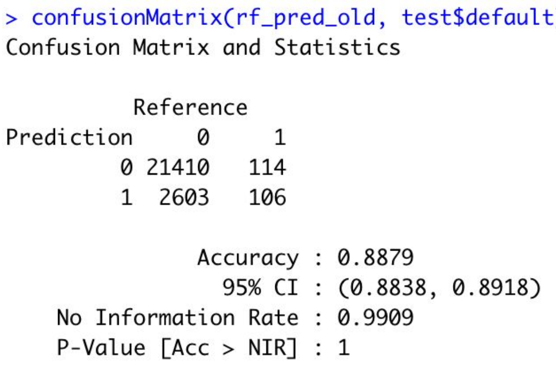
In Plot 18, we can clearly see how the loans are classified based on these features. Credit score is the most important feature that determines whether the loan is a default loan or healthy loan because it is the first classification criteria in the tree. If the credit score of the borrower is greater than 687, there is a 70% possibility that the loan is healthy. If the minimum credit score between borrowers is smaller than 664, the loan is more likely a loan default. If the minimum credit score between borrowers is in the range of 664 and 687 and OLTV is greater than 90%, the loan is likely a loan default. Otherwise, if the loan is a healthy loan unless the borrower is located at some specific state like LA.



Plot 18: Decision Tree visualization

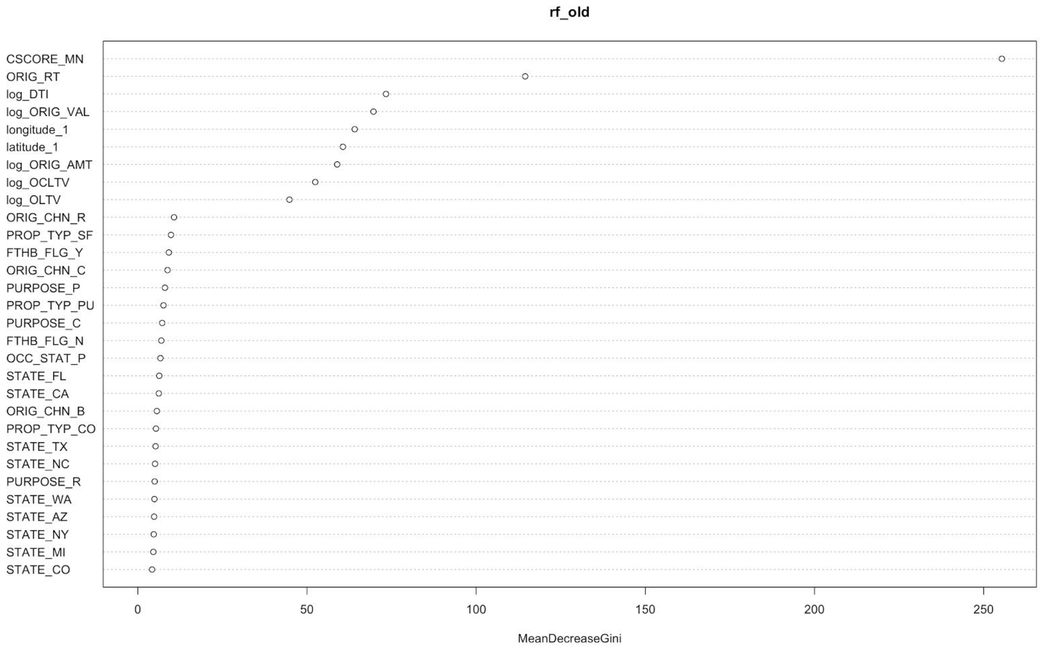
Random Forest

Based on the not bad decision tree model result, I implemented the random forest model. It will classify better than the decision tree because it will construct many trees and combine the results from these trees. I chose to build 20 random trees in the model and the accuracy has improved 3.7% (Plot 19). I think the model performance improvements are due to the different feature selections in two models. From Plot 20, we can see the most important feature is still the minimum credit score between borrowers, but the second important feature in random forest is the original interest rate. The next importance level features include original DTI and original value. As for how these features will affect the result, I gave a specific explanation in the next part.



```
> confusionMatrix(rf_pred_old, test$default)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 21410   114
         1  2603   106

               Accuracy : 0.8879
                 95% CI : (0.8838, 0.8918)
    No Information Rate : 0.9909
    P-Value [Acc > NIR] : 1
```

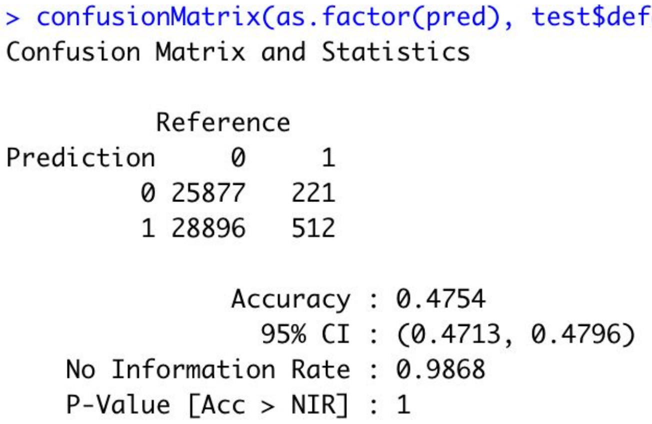
Plot 19: Random Forest Confusion Matrix



Plot 20: First Random Forest Feature Importance

XGBoost

XGBoost is a more complicated boosting algorithm. Random Forest improves the model performance through combing a large number of trees using averages or ‘majority’ rules at the end of the process, while XGBoost tries to combine decision trees at the beginning. Based on the very good result from Random Forest, I tried XGBoost because it always performs well and runs faster. However, the result does not perform well. The accuracy even not achieved 0.5, which is the accuracy of random classification. I think the reason behind the bad result XGBoost shows is that I did not clean data as the XGBoos model needs. Besides, there are too many parameters that need to be tuned in the XGBoost.



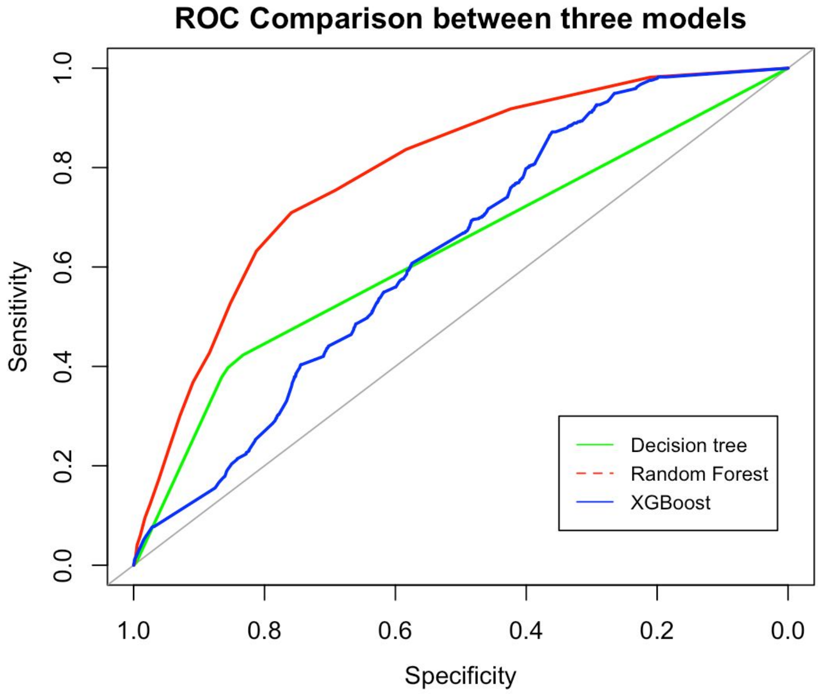
```
> confusionMatrix(as.factor(pred), test$default)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 25877   221
         1 28896   512

               Accuracy : 0.4754
                 95% CI : (0.4713, 0.4796)
    No Information Rate : 0.9868
    P-Value [Acc > NIR] : 1
```

Plot 21: XGBoost Confusion Matrix

Model Comparison

After establishing three tree-based models, I compare the model performances using the ROC curve. ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds. As for the ROC curve, the faster the turning point gets closer to the left-top and the bigger the area under the ROC curve, the model performs better. In Plot 22, the red line representing the Random Forest model performs best.

In addition to the ROC curve performance and AUC, the specificity is also very important to the very unbalanced data. Therefore, I also analyzed how sensitivity will change with the change of specificity. Specificity refers to how the model will correctly classify abnormal class. Sensitivity refers to how the model will correctly classify the normal class. In the ROC plot, it shows how much sensitivity will be sacrificed when the model improves specificity. The Random Forest I built shows when it improves specificity from 0 to 0.8, the sensitivity will just sacrifice 0.2 (Plot 22). The results are much better than the other two models.



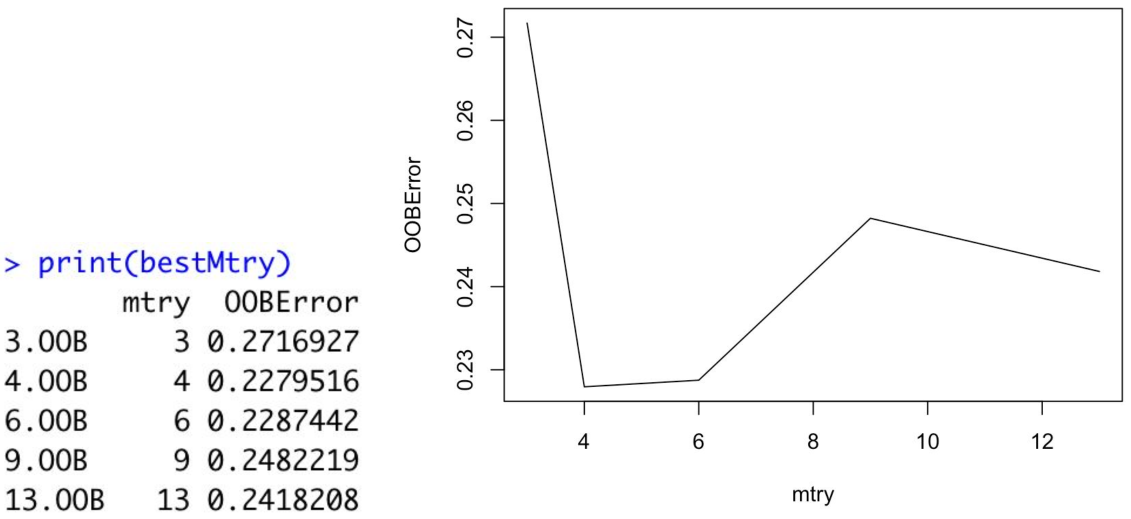
Plot 22: ROC curve comparison between three models

Best Model Optimization & Analysis

According to the model comparison, the best model is Random Forest. I tried to further optimize the model performance by tuning the 'mtry' parameter in Random Forest. The 'mtry' parameters refers to the number of variables randomly sampled as candidates at each split. I used tuneRF() function that searches for choosing optimal mtry values. From Plot 23, we can see that the most accurate value for 'mtry' was 4 with an OOBError of 0.2279516 when I set the 'ntree' parameter with 20. Then, I use the optimized parameters to build a new Random Forest and the accuracy has improved 5% (Plot 24).

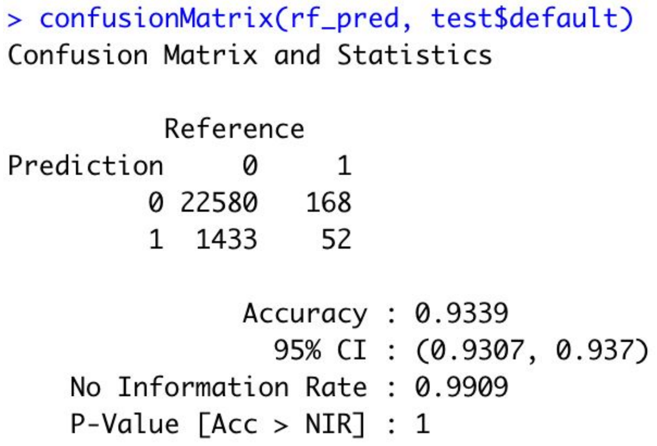
The top 3 important features are as same as that in the first Random Forest, which are minimum credit score between borrowers, original interest rate and debt-to-income ratio (DTI) (Plot 25). The difference is that the new Random Forest model considers the feature of OCLTV as important as DTI. From Plot 26, we can see that the features are grouped more clearly than that in the first Random Forest in Plot 20. Credit score is the most important feature and is much more important than others, with the MeanDecreaseGini of 113. The second important feature is the original interest rate, with the MeanDecreaseGini of 75. DTI and OCLTV are in the third-tier importance level, with the MeanDecreaseGini around 42-45. Longitude, latitude, original home value and original loan amount are in the fourth-tier importance level, with the MeanDecreaseGini around 23-24.

Overall, the optimized Random forest model performs very well. I drew some plots to show how these important features affect the loan default. I gave the detailed insights and recommendations I got from the plots in the conclusion part.



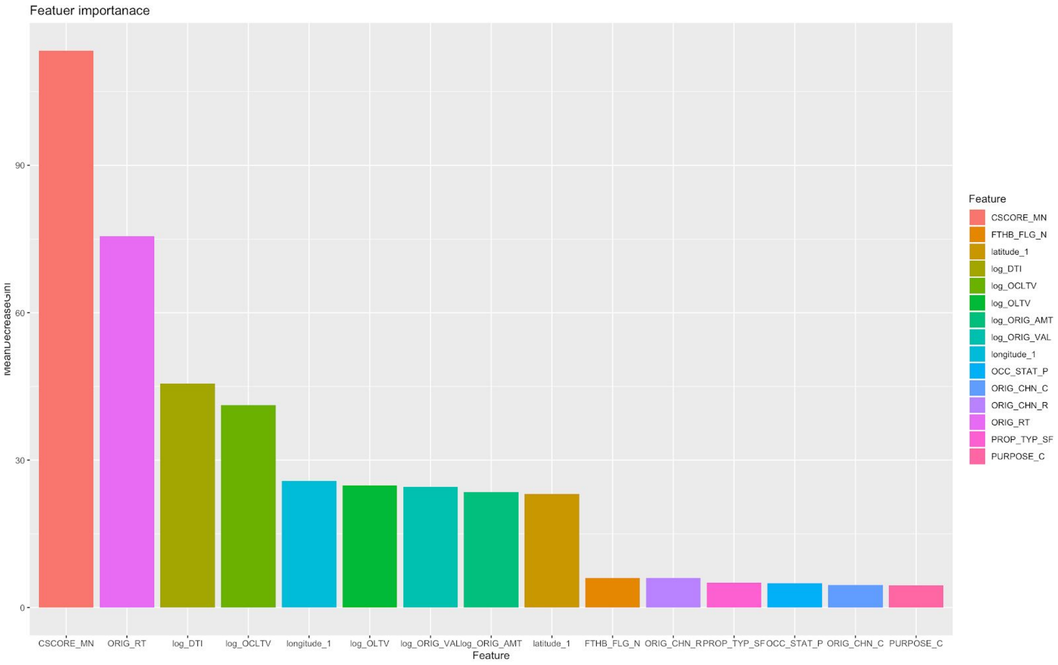
```
> print(bestMtry)
       mtry  OOBError
3.OOB     3 0.2716927
4.OOB     4 0.2279516
6.OOB     6 0.2287442
9.OOB     9 0.2482219
13.OOB   13 0.2418208
```

Plot 23: Random Forest 'Mtry' Parameter Tuning

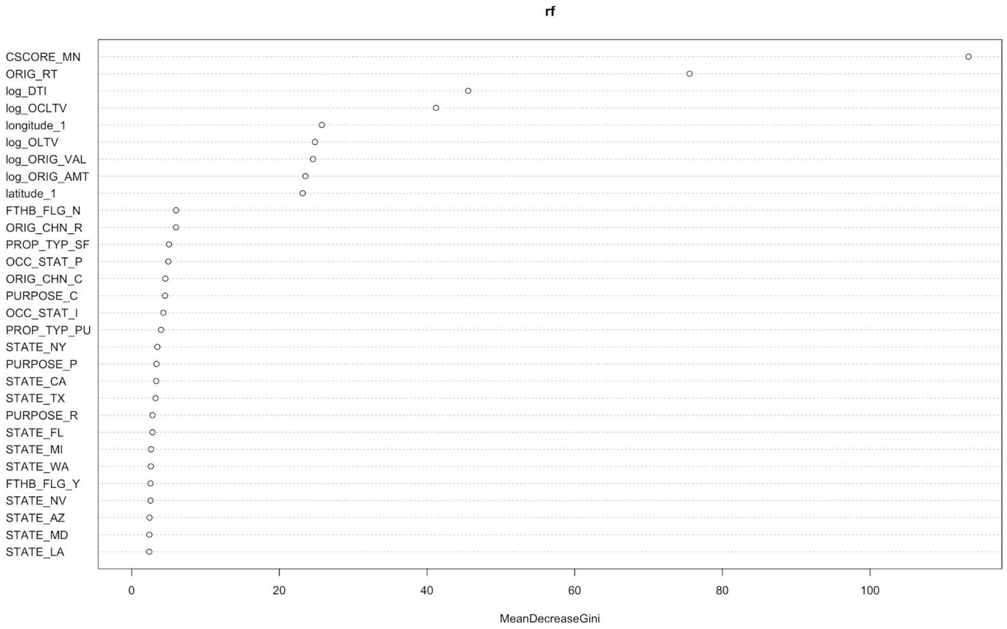


```
> confusionMatrix(rf_pred, test$default)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 22580   168
         1  1433    52

               Accuracy : 0.9339
                 95% CI : (0.9307, 0.937)
    No Information Rate : 0.9909
    P-Value [Acc > NIR] : 1
```

Plot 24: Optimized Random Forest Confusion Matrix



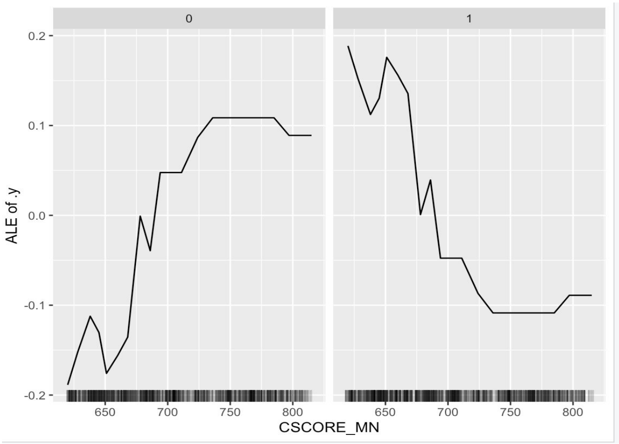
Plot 25: Top 15 Random Forest Feature Importance Barplot



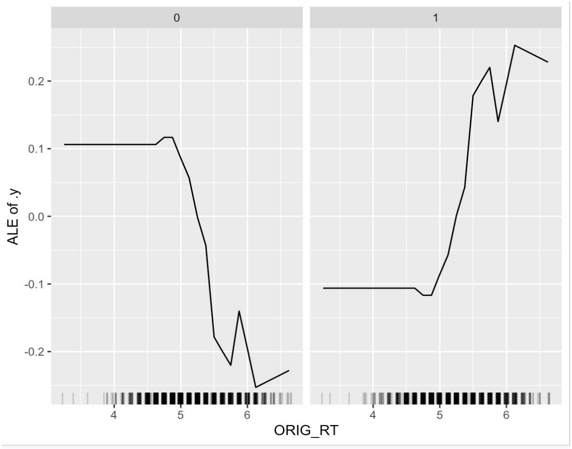
Plot 26: Random Forest Feature Importance

Conclusions and Discussion

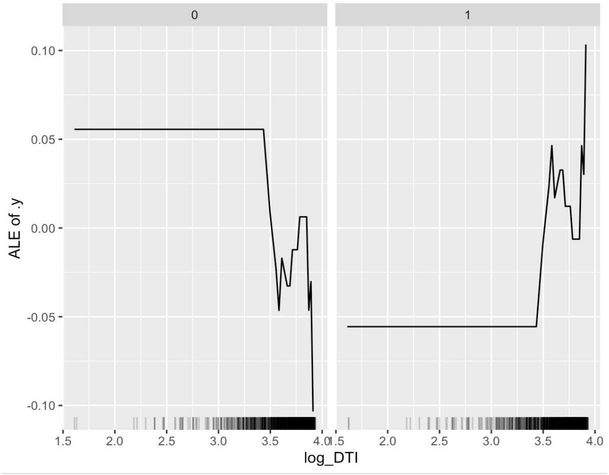
According to the modeling performance result, it is obvious that the optimized Random Forest is the best model. I extracted the most four important features from the model and analyzed how these features will affect the loan default. In Plot 27, Plot 28, Plot 29 and Plot 30, the left plots show each feature's distributions of healthy loans and the right plots show each feature's distributions of defaulted loans. We can see that the curves in each left and right plots in these four plots are totally different. From Plot 27, we can see that when the credit score is lower than 730, the loan is more likely a defaulted loan. If the borrower's credit score is below 680, the possibility of defaulted loan is higher. As for the original interest rate, when the interest rate is lower than 5%, the loan is more likely to be a healthy loan, otherwise, the loan is more likely to be a defaulted loan, especially when the interest rate achieves 5.5% (Plot 28). In terms of DTI, when the DTI achieves 43% or higher, the loan is highly possible to be a defaulted loan (Plot 29). The Plot 30 shows that when the loan amount achieves 90% of the home value, the loan is more likely to be a defaulted loan.



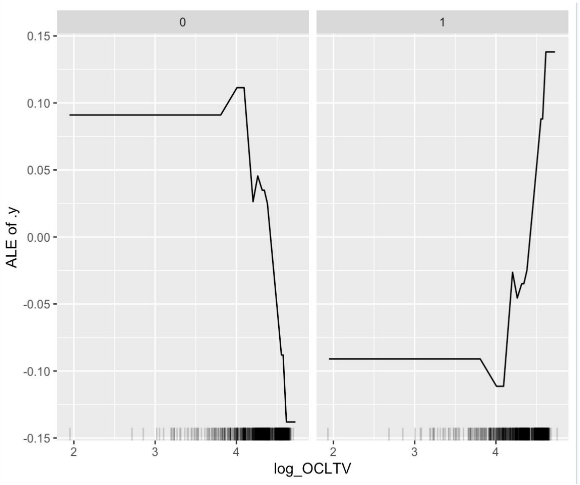
Plot 27: Random Forest——Credit Score



Plot 28: Random Forest——Original Interest Rate



Plot 29: Random Forest——DTI



Plot 30: Random Forest——OLCTV

Using the information I summarized, I recommend the banks try to strictly control the credit score criteria. If the borrower's credit score is lower than 650, the banks should pay attention to the borrowers. The banks could provide less loan and high default fee for this kind of borrower. In addition, when the loan amount is more than 90% of the original home value or the interest rate is high, the banks also need to pay attention to the loans. The banks could sign long-term and low-month-paid contracts with the borrowers to make sure they are affordable the monthly pay.

This is an end-to-end project, where I started from proposing an idea, collecting relevant data, cleaning and exploring data, establishing classification models to extracting useful information from models. Foe the next step, I will further optimize the models and try more models. Besides, I will group the borrowers using clustering methods and try to give some targeted and actionable recommendations for the banks.

References

Data source:
https://capmrkt.fanniemae.com/portal/funding-the-market/data/loan-performance-data.html

Github link:

https://github.com/baiting7/DataAnalytics2020_Baiting_Gai/tree/master/Final%20Project