Prediction of Foreclosure of a Loan by Using Supervised Learning.

1. Introduction:

Foreclosure is a legal process in which a lender attempts to recover the balance of a loan from a borrower who has stopped making payments to the lender by forcing the sale of the asset used as the collateral for the loan.
As soon a borrower fails to make a loan or mortgage payment on time, the loan becomes delinquent. The foreclosure process begins when a borrower defaults or misses a loan or mortgage payment. At this point, a homeowner in default will be notified by the lender.

Problem Definition:

The goal of this project is to predict the probability, as to which borrower of a loan from a financial institution may or may not result in an NPA.

Inherent Value of Project:

Foreclosure has negative consequences for both parties, namely the Borrower and the Lender. A foreclosed property or asset usually doesn't recover the amount that is being defaulted, Which is added to the portfolio of a foreclosed property. Such assets sell at a much lower rate than the market value which hurts the lender.
For the Borrower, it leads to a poor credit report which greatly diminishes their credibility and may cause severe hindrance in procuring loans in the foreseeable future. It also has marked negative consequences on the mental health of the borrower and his kin, which could be mitigated if the lender and borrower can reach a mutual understanding as to how to navigate out of the situation.

2. Data :

https://www.kaggle.com/absags/predict-forclosure-probability

[Disclaimer - At the time when I had started working on the dataset I wasn't aware of its Data Quality, which essentially means that the data provided here is simulated. The output of the analysis or modelling in parts did not seem intuitive but I continued working on this dataset as it made for a great exercise for concepts like data cleaning, wrangling, Time-series etc in one dataset. So please take the result with a pinch of salt].

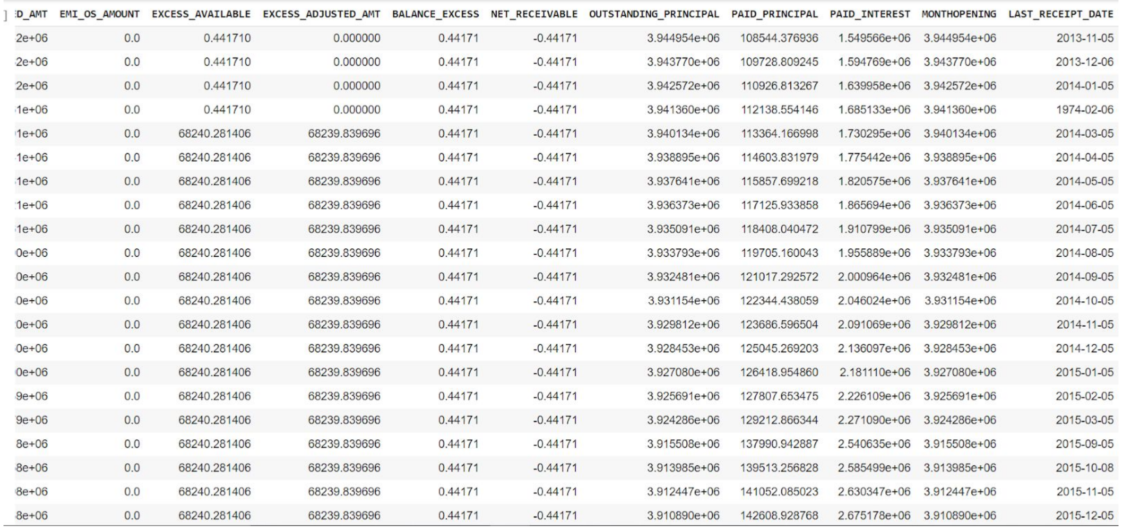
The dataset comprised of 5 separate excel files that contained the following information:

1. Customers_31JAN2019.xlsx - It had information about the personal details of each customer. The details were regarding age, sex, qualification, income and other demographic details.
2. LMS_31JAN2019.xlsx - It has the details of all the loan transactions that happened between each customer and their respective bank under a loan agreement id. Each loan has a varying duration of transaction history, meaning some agreement ids have a history for a few months to some having over several extended months(more than 30)
3. RF_Final_Data.xlsx - This file has information about any communication that happened between a customer and their bank in the form of an email. Details include email subject, letter body, query type etc.
4. Train_foreclosure.xlsx - This file has the unique agreement id of some of the loans along with the target variable that we need to predict i.e. Foreclosure
5. Test_foreclosure.xlsx - It contains only the agreement ids of the rest of the loans that weren't part of the previous file.

Other than the RF_Final_Data.xlsx file, the rest of the files were made use of to solve the given Data science problem at hand.

3. Data Wrangling & Cleaning

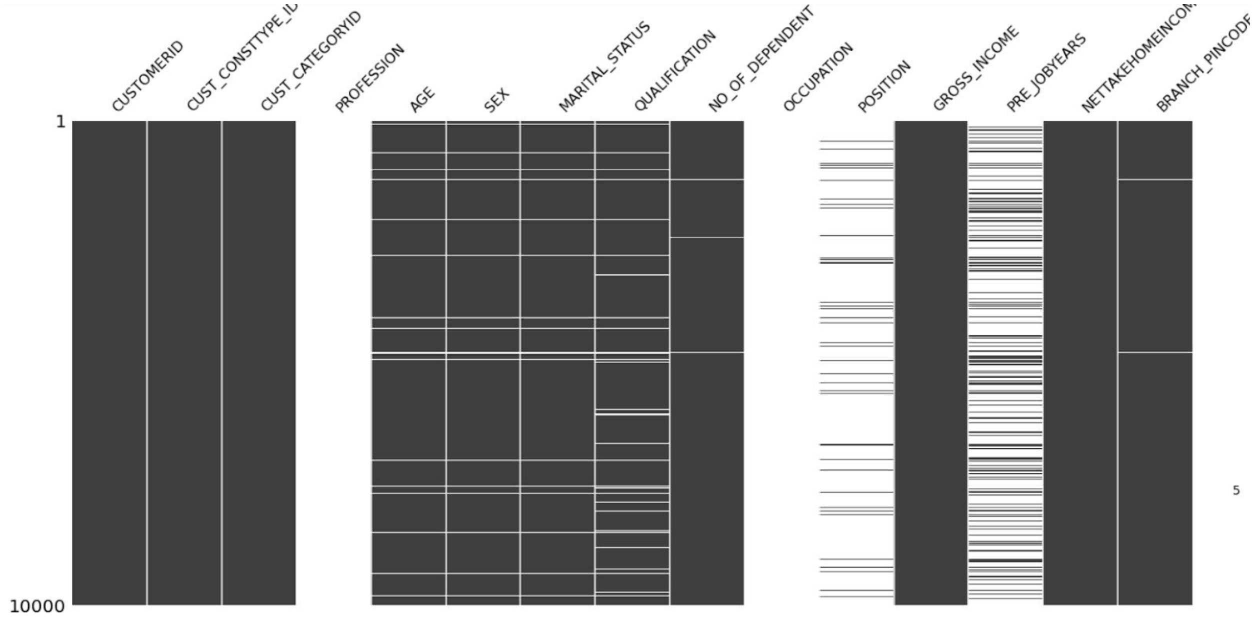
- In LMS_31JAN2019.xlsx file, upon loading the dataset, the last receipt date column is inspected, which serves as the time index for all the variables in a loan transaction of a single customer under a particular agreement id. This feature had a data entry error which needed to be rectified.



D_AMT	EMI_OS_AMOUNT	EXCESS_AVAILABLE	EXCESS_ADJUSTED_AMT	BALANCE_EXCESS	NET_RECEIVABLE	OUTSTANDING_PRINCIPAL	PAID_PRINCIPAL	PAID_INTEREST	MONTHOPENING	LAST_RECEIPT_DATE
2e+06	0.0	0.441710	0.000000	0.44171	-0.44171	3.944954e+06	108544.376936	1.549566e+06	3.944954e+06	2013-11-05
2e+06	0.0	0.441710	0.000000	0.44171	-0.44171	3.943770e+06	109728.809245	1.594769e+06	3.943770e+06	2013-12-06
2e+06	0.0	0.441710	0.000000	0.44171	-0.44171	3.942572e+06	110926.813267	1.639958e+06	3.942572e+06	2014-01-05
1e+06	0.0	0.441710	0.000000	0.44171	-0.44171	3.941360e+06	112138.554146	1.685133e+06	3.941360e+06	1974-02-06
1e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.940134e+06	113364.166998	1.730295e+06	3.940134e+06	2014-03-05
1e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.938895e+06	114603.831979	1.775442e+06	3.938895e+06	2014-04-05
1e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.937641e+06	115857.699218	1.820575e+06	3.937641e+06	2014-05-05
1e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.936373e+06	117125.933858	1.865694e+06	3.936373e+06	2014-06-05
1e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.935091e+06	118408.040472	1.910799e+06	3.935091e+06	2014-07-05
0e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.933793e+06	119705.160043	1.955889e+06	3.933793e+06	2014-08-05
0e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.932481e+06	121017.292572	2.000964e+06	3.932481e+06	2014-09-05
0e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.931154e+06	122344.438059	2.046024e+06	3.931154e+06	2014-10-05
0e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.929812e+06	123686.596504	2.091069e+06	3.929812e+06	2014-11-05
0e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.928453e+06	125045.269203	2.136097e+06	3.928453e+06	2014-12-05
0e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.927080e+06	126418.954860	2.181110e+06	3.927080e+06	2015-01-05
9e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.925691e+06	127807.653475	2.226109e+06	3.925691e+06	2015-02-05
9e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.924286e+06	129212.866344	2.271090e+06	3.924286e+06	2015-03-05
8e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.915508e+06	137990.942887	2.540635e+06	3.915508e+06	2015-09-05
8e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.913985e+06	139513.256828	2.585499e+06	3.913985e+06	2015-10-08
8e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.912447e+06	141052.085023	2.630347e+06	3.912447e+06	2015-11-05
8e+06	0.0	68240.281406	68239.839696	0.44171	-0.44171	3.910890e+06	142608.928768	2.675178e+06	3.910890e+06	2015-12-05

- Few of the columns belonging to the loan transaction data were dropped based on the lack of utility i.e. they don't serve as good predictor variables

- The customer demographics data file is loaded for data cleaning. On inspection, it is observed some of the variables have lots of missing values, more than 70% in a few cases. All such variables are dropped.



- In other variables, with missing values, the rows are imputed with the median values that appear in the respective features, which is inferred by looking at the descriptive statistic summary of all the features.
- It is to be mentioned that the customer demographic data didn't include information about all the customers in the LMS file. There are 32,895 customers in the loan transaction system, while personal details are there only for 10,000 customers, which leaves out a lot of missing values interspersed in the dataset.Such missing values are replaced with -99,999 so that during the modelling phase these values would be ignored by the machine learning algorithm as outliers.
- All the various categorical variables that are present in the complete merge data frames are numerically encoded.
- Few of the observations were dropped because they were loan transactions which were not assigned to any of the customers, which was deduced to be noise in the dataset.

4. Data Preparation

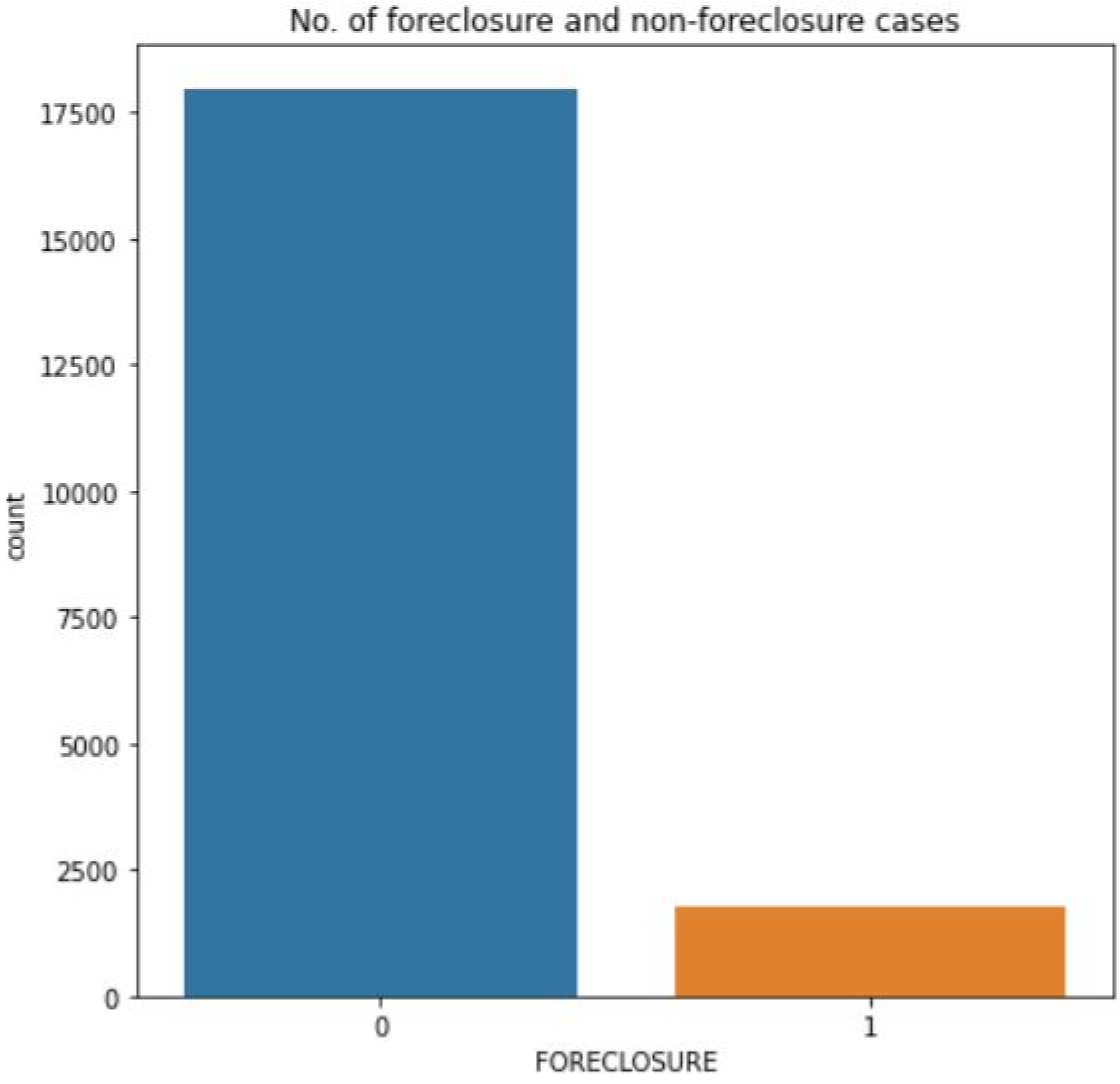
- To make use of the final merged data, the dataset needs to be converted from a long format to a wide format. Concretely, the individual rows should be reduced to unique customer id, agreement id and the last loan transaction in each case of the loan transaction history. This is achieved by making use of the last receipt date feature and the entire dataset is sorted in a manner in which the agreement and customer id are sorted in ascending order with the loan transaction history in descending order, with the last transaction displayed at first.

- we observe that the task at hand is inherently related to time series problems. Hence, we explore Lag Features and present our findings. In this study, we consider 6-months history of each of the customers with one lag variable for each month from each of the chosen predictor variables. In future experiments, we plan to leverage a deeper understanding of Finance to expand the possible set of features.

5. Exploratory Data Analysis

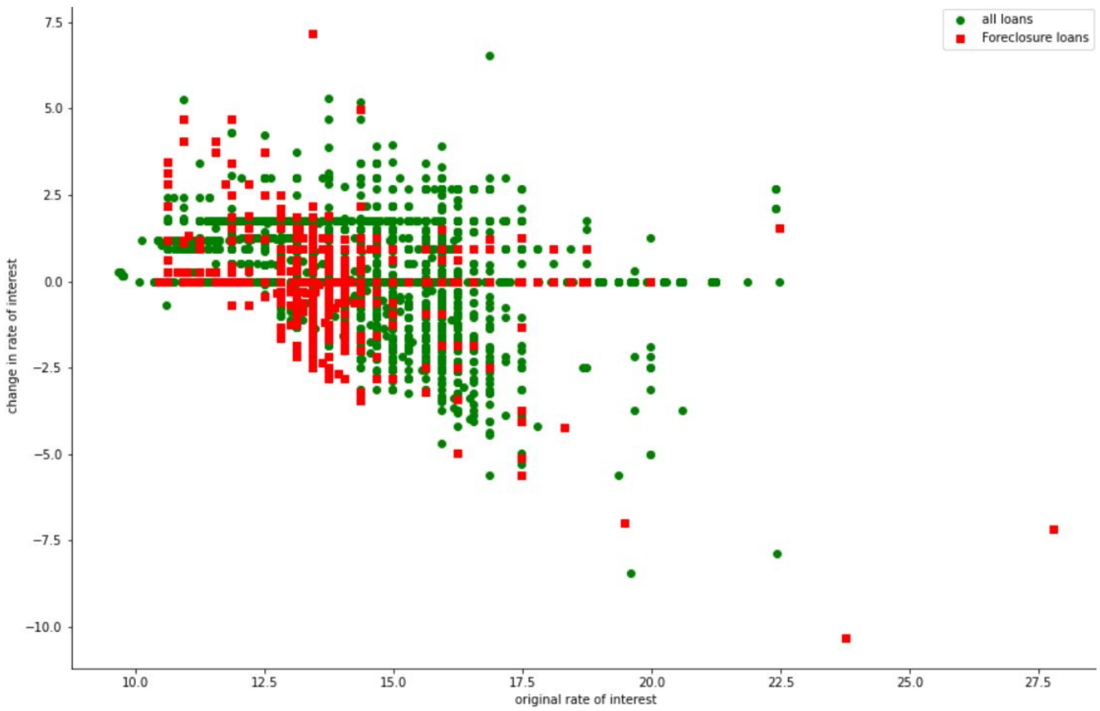
After merging the relevant files to create the training data, we explore all the features and the different interactions between them.

- Distribution of target variable



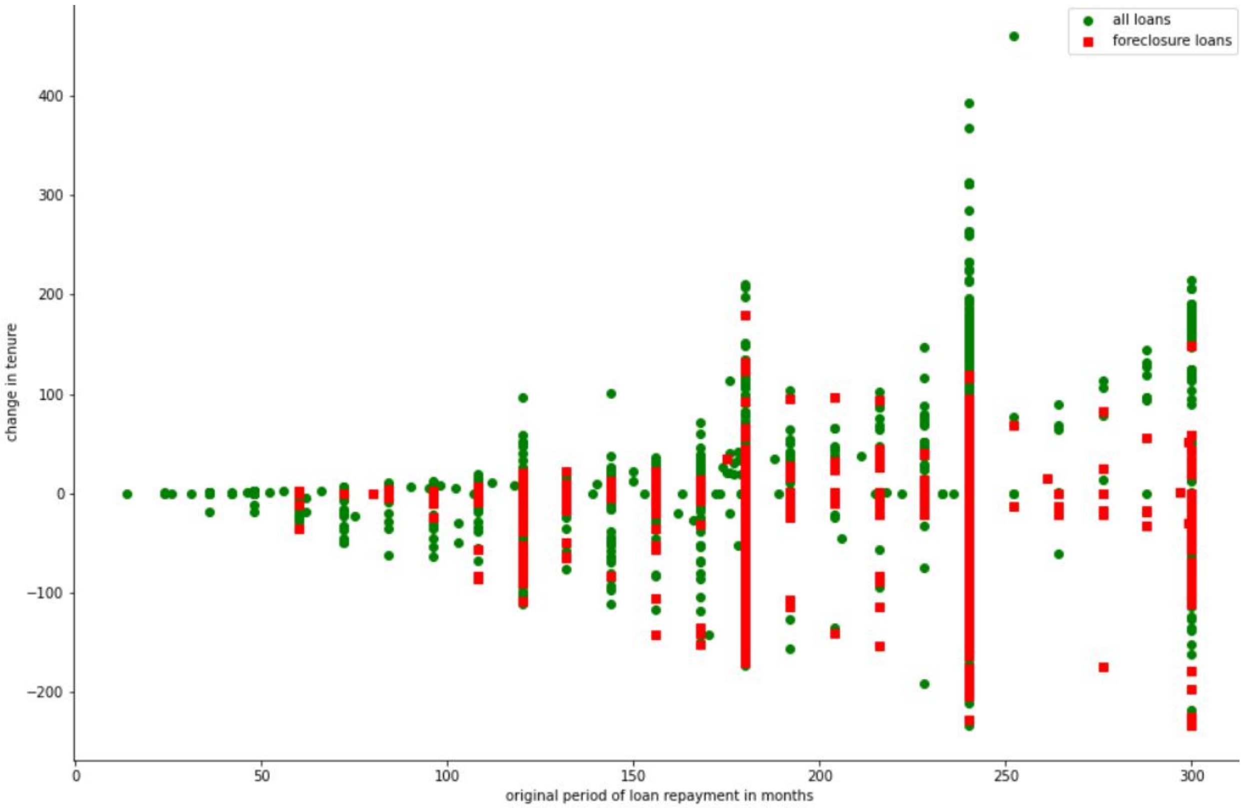
From the plot, we observe that the classes are imbalanced. In other words, we need to incorporate the weights of the classes during model training.

- The change in the rate of interest for loans foreclosed vs all loans



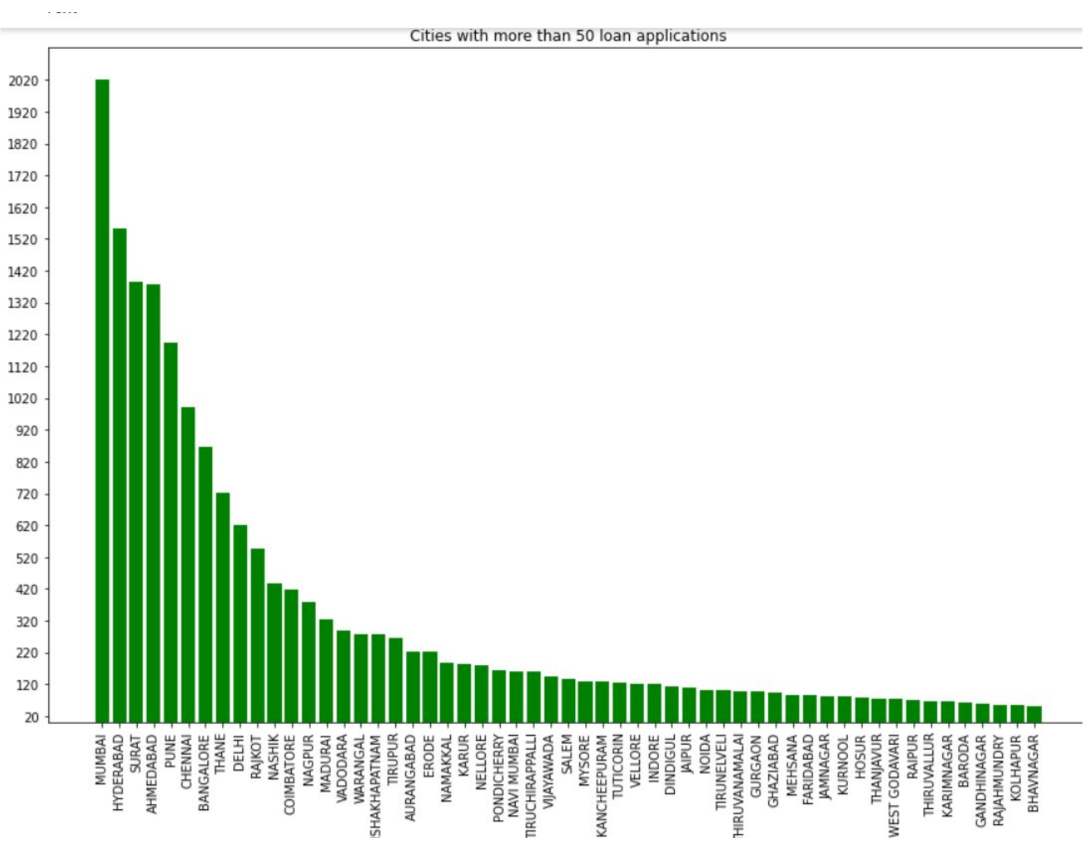
[No clear pattern to the change in rate of interest for foreclosed cases is observed.Can be investigated further]

- The change is the tenure of loan repayment foreclosed vs all loans

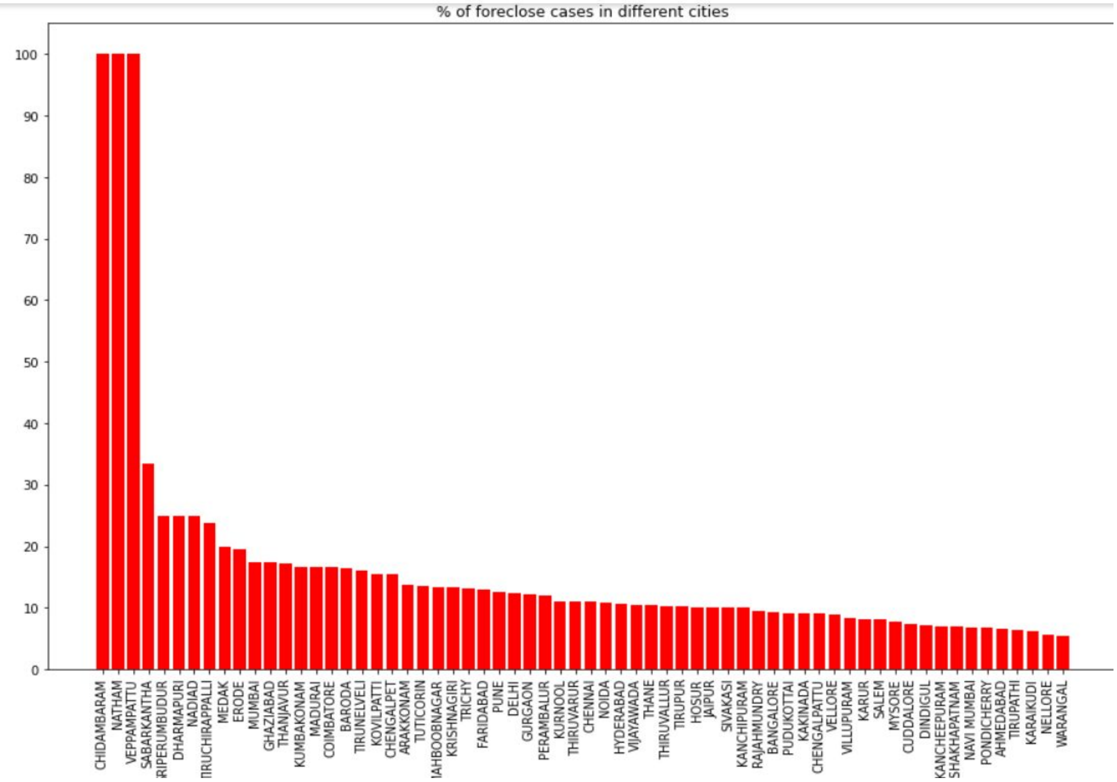


[We can clearly see that the loan repayment tenure has reduced for Foreclosed loans]

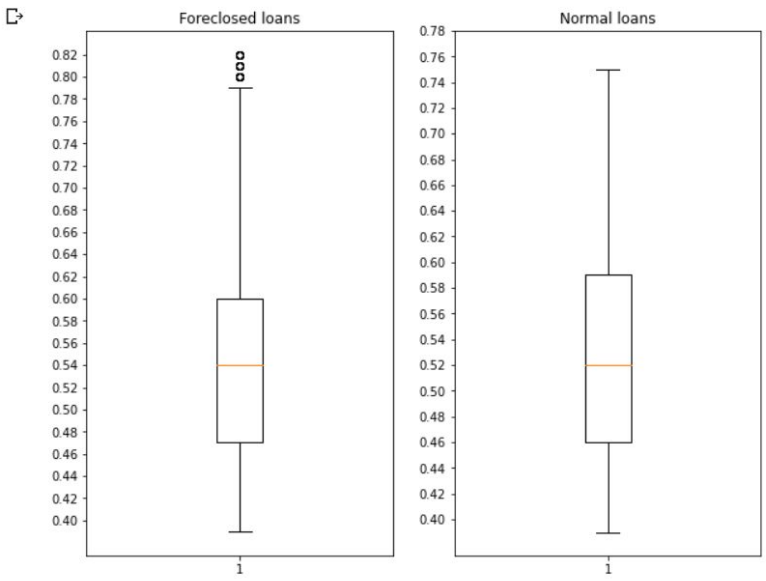
- The no. of loan applications from different cities



- Some Cities with a high percentage of foreclosed loans

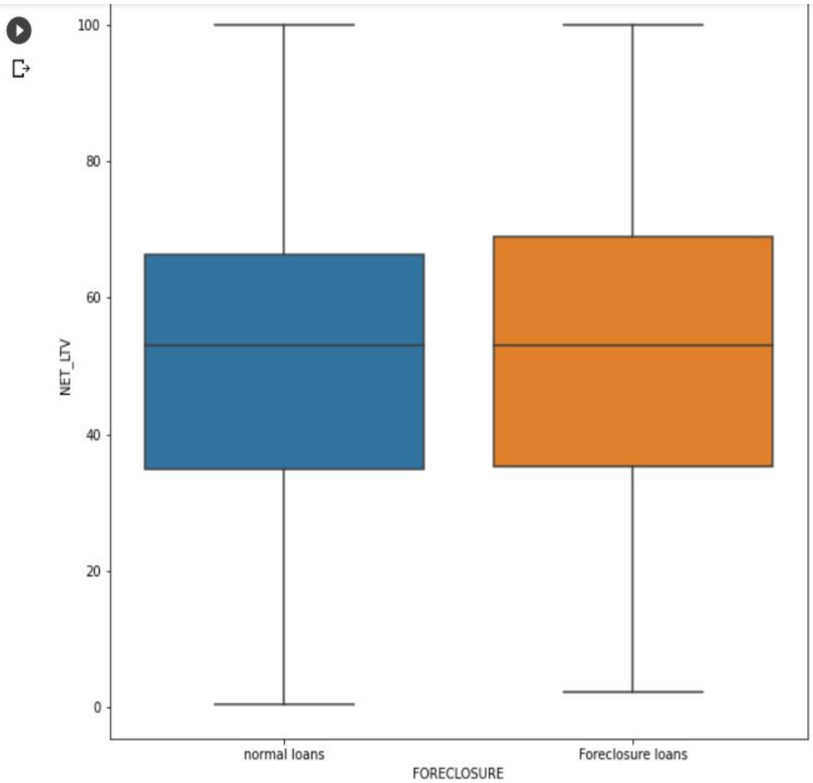


- FOIR - Fixed Obligation to income ratio



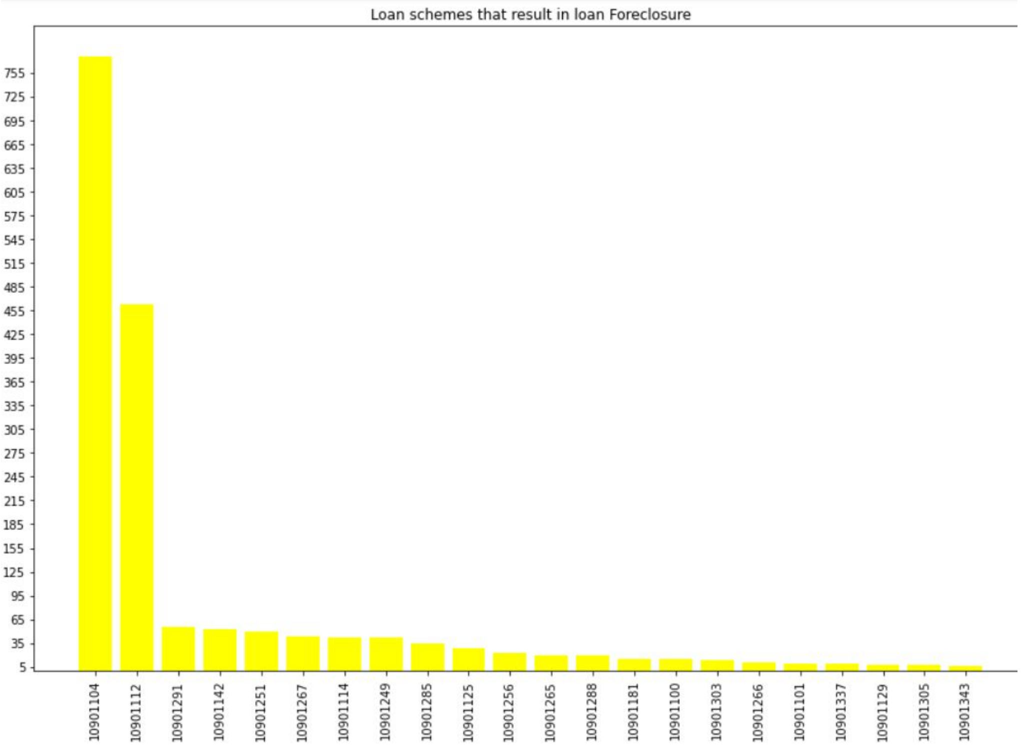
FOIR - Fixed Obligation to income ratio is calculated to assess how much of the income would go into paying the installments. A higer FOIR is a bad sign. On close inspection of the two box plots it seems that normal loans have a lower FOIR than Foreclosed loans.

- Net loan to volume ratio



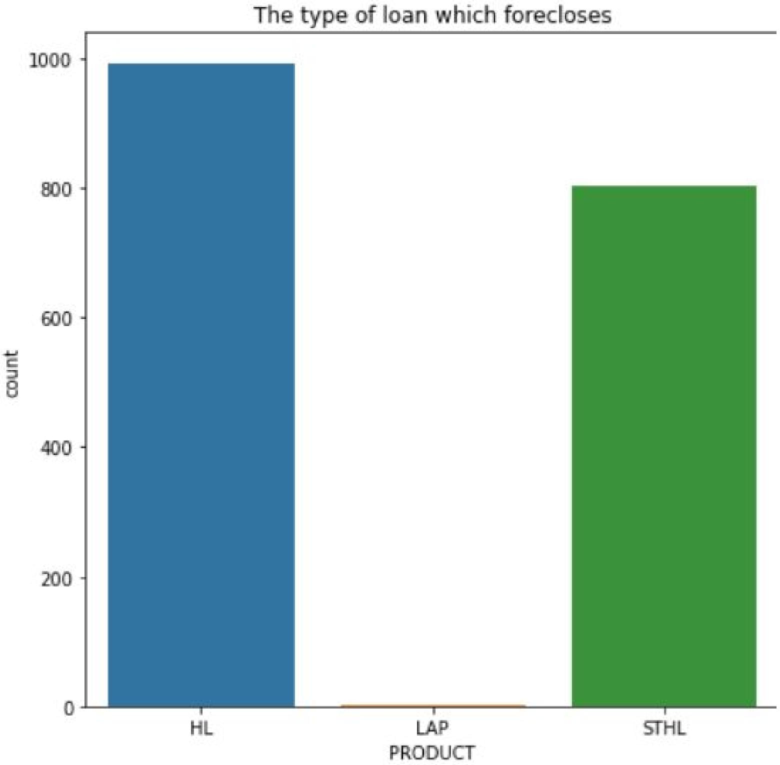
The Boxplot for the normal loans seems to be left skewed and the foreclosed one seems fairly symmetric. This means the mean Net-to-loan ratio is lower in case of normal loans than in Foreclosed ones. Higher NTL ratio indicates inability to repay loans.

- The different scheme ids that result in foreclosed loans

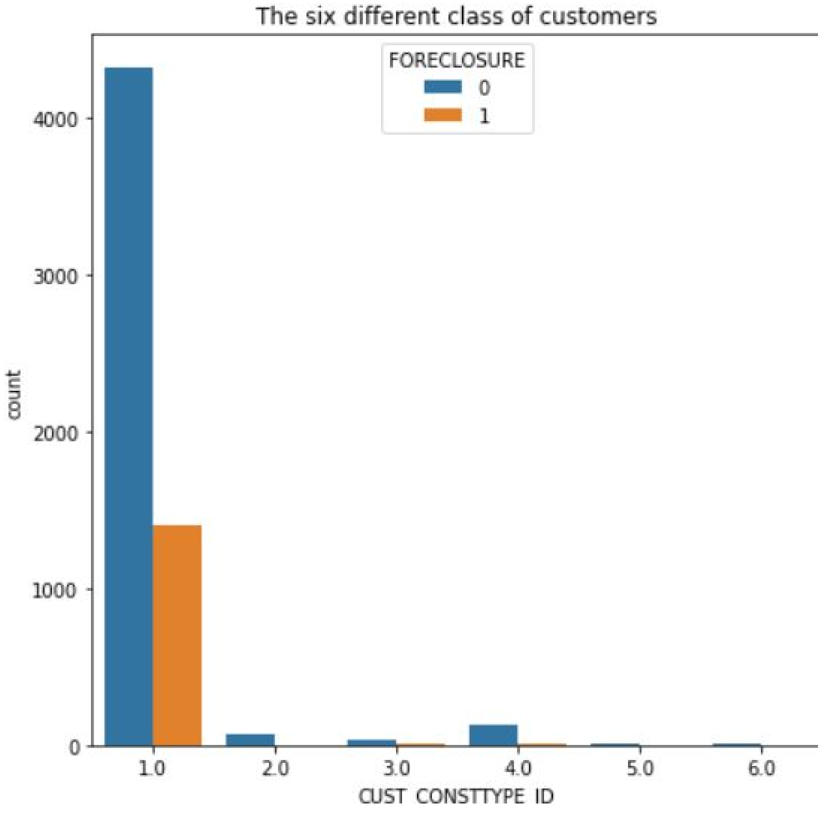


[Scheme no.- 10901104 & 10901112 result in majority of the foreclosed loans]

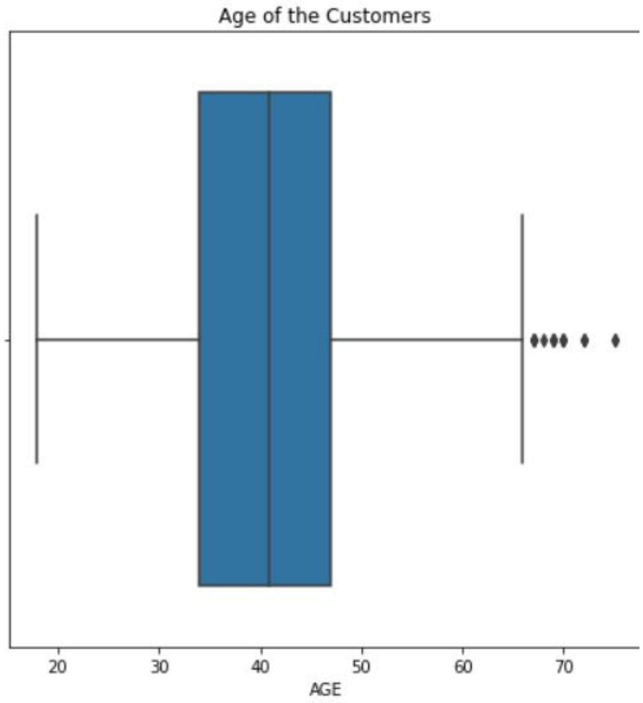
- The type of loan product that results in Foreclosure



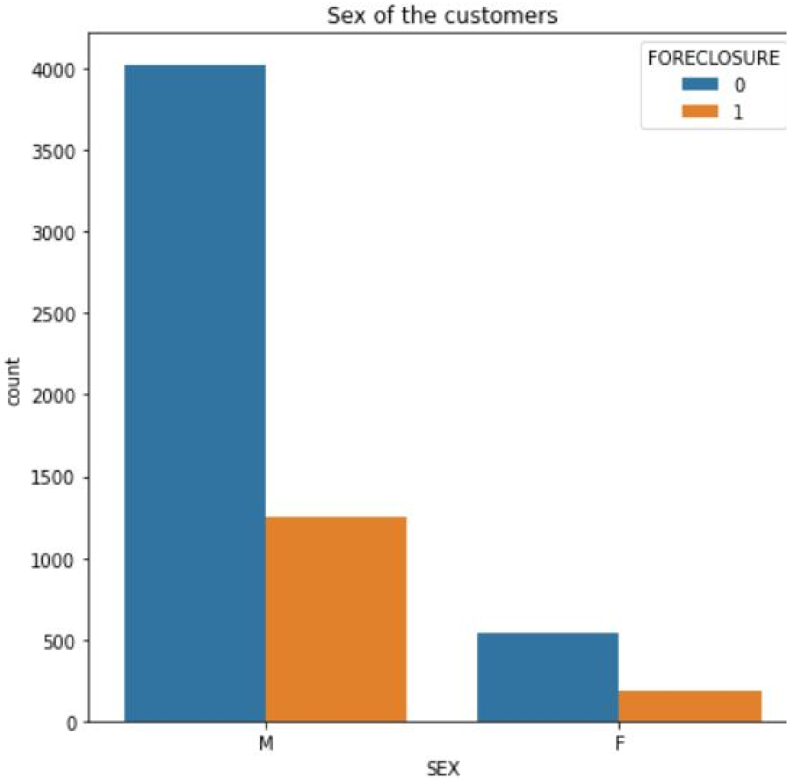
- The customers were segregated in 6 different classes. The plot describes the class that is most likely to default



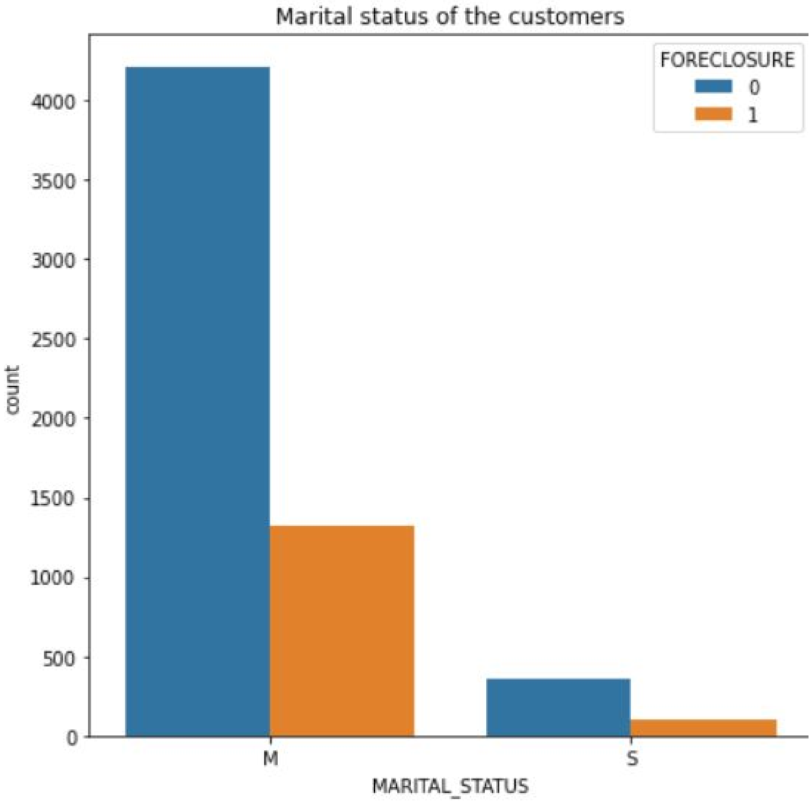
- Distribution of customer age



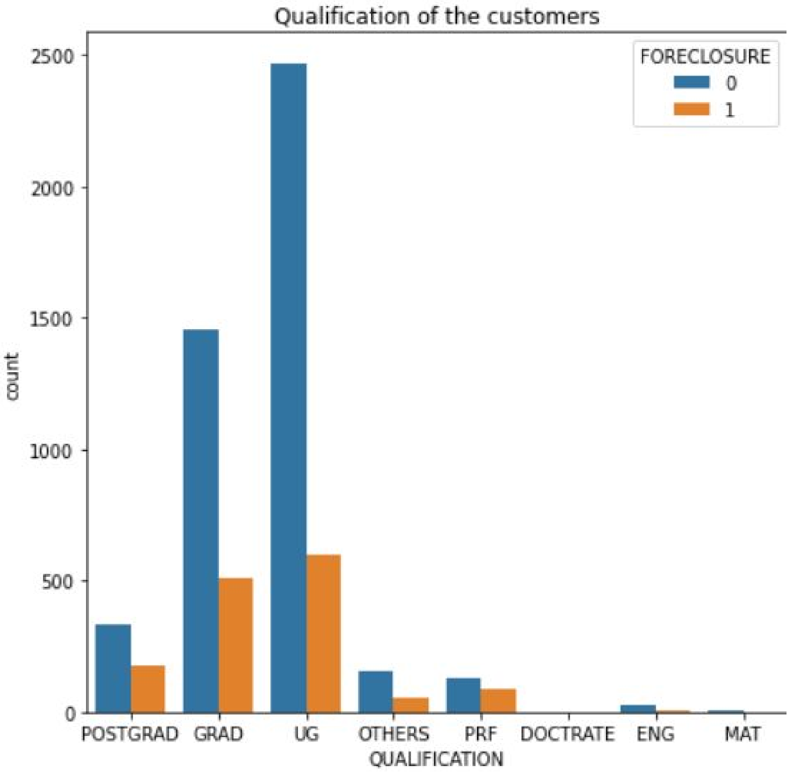
- Gender distribution of customers



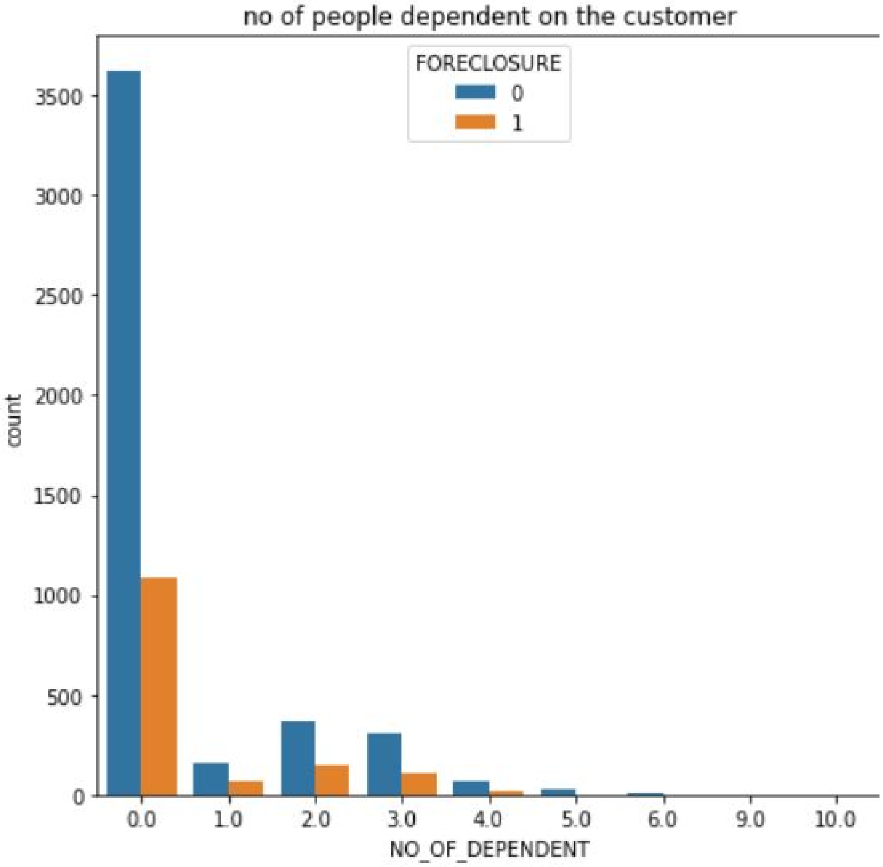
- Marital Status of the customers



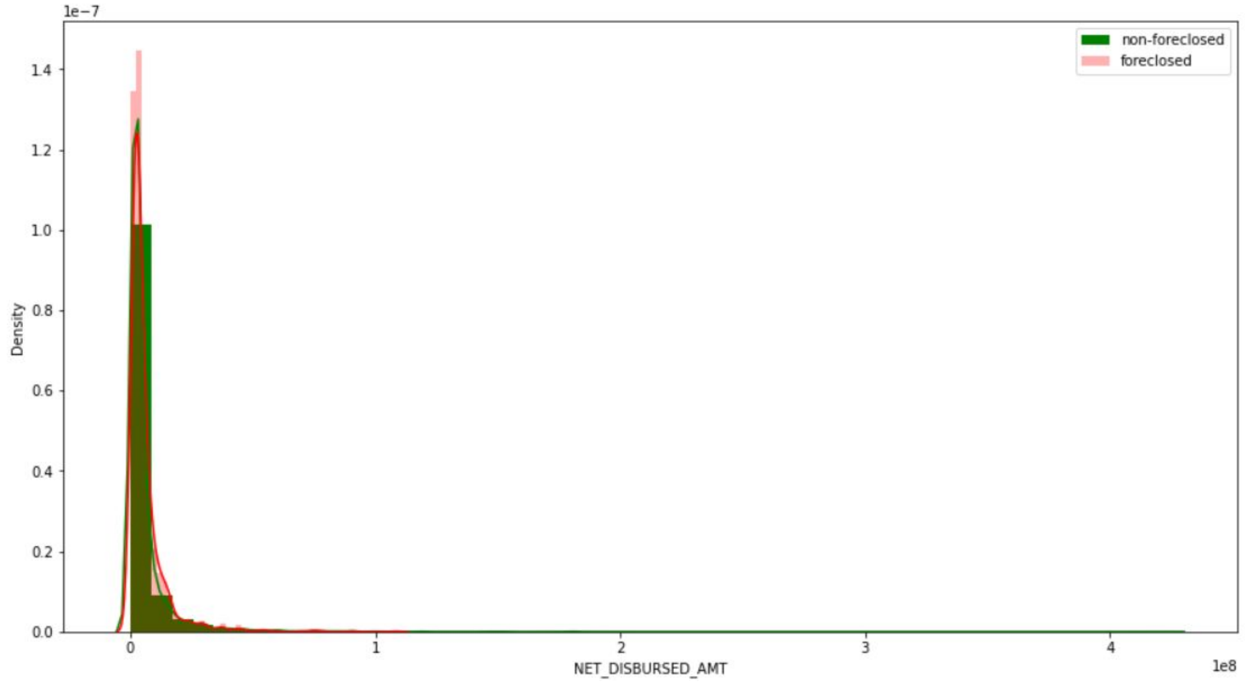
- Qualification distribution of Customers



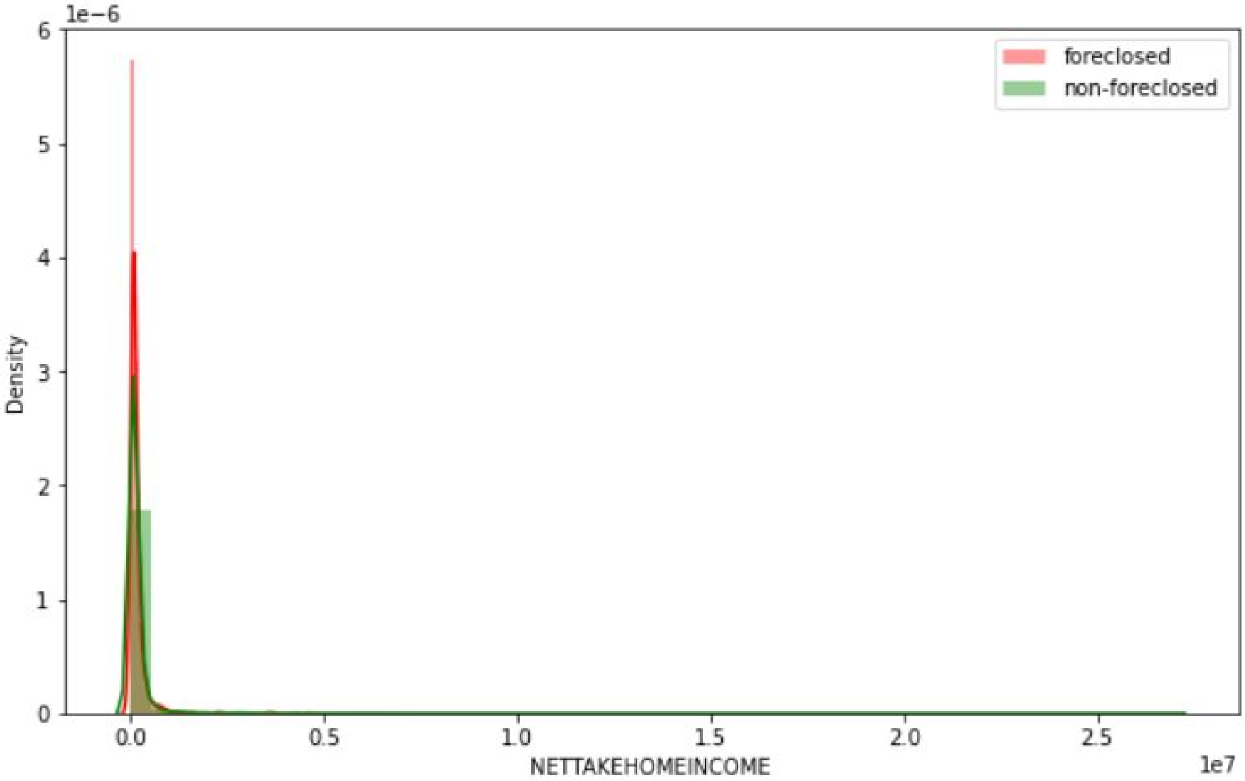
- No of dependent members on the customers



- Distribution of amount given out as loans - Foreclosed vs Non foreclosed



- The distribution of income of all customers - Foreclosed vs Non-foreclosed



6. Hypothesis Testing

In the plot above, where the change in the rate of interest of loans for customers throughout their loan repayment period is shown, there wasn’t a clear pattern to differentiate between the two cases.

It is to be investigated further. The null hypothesis is formulated as follows -

- There is no significant difference in the change in the rate of interest between foreclosed and normal loan cases.

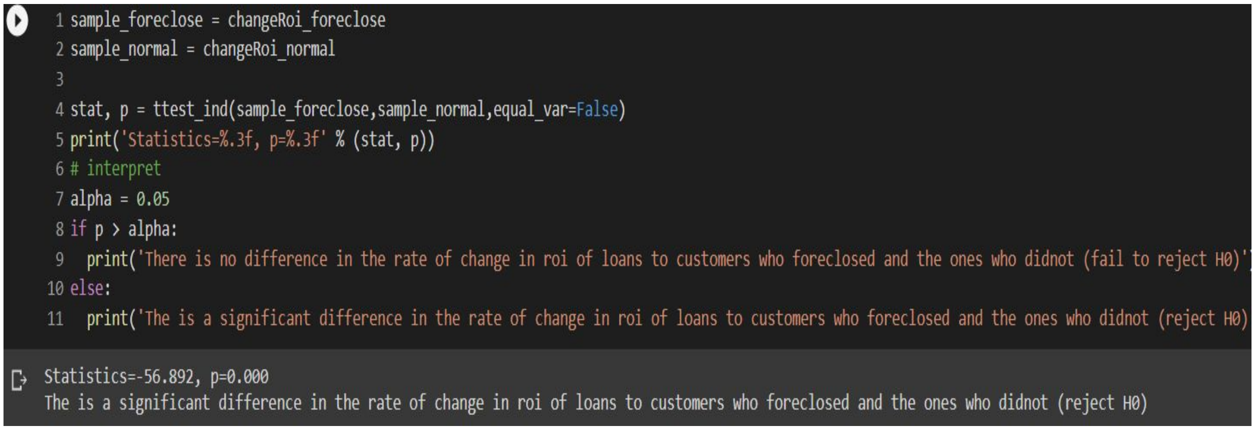
Alternate Hypothesis -

- There is a significant difference in the change in the rate of interest between foreclosed and normal loan cases.

To determine this we would make use of the welch's t-test as our test for statistical significance.

Welch’s t-test[https://www.statisticshowto.com/welchs-test-for-unequal-variances/] is used because it doesn’t have the assumption that the variances of the two distributions are equal. As far as the assumption(the distributions that would be compared should follow a normal distribution)of normality is concerned,
we make use of the Central Limit Theorem to ensure that.

The outcome of the test -



```
sample_foreclose = changeRoi_foreclose
sample_normal = changeRoi_normal

stat, p = ttest_ind(sample_foreclose,sample_normal,equal_var=False)
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
  print('There is no difference in the rate of change in roi of loans to customers who foreclosed and the ones who didnot (fail to reject H0)'
else:
  print('The is a significant difference in the rate of change in roi of loans to customers who foreclosed and the ones who didnot (reject H0)
```

```
Statistics=-56.892, p=0.000
The is a significant difference in the rate of change in roi of loans to customers who foreclosed and the ones who didnot (reject H0)
```

7. Predictive Modelling

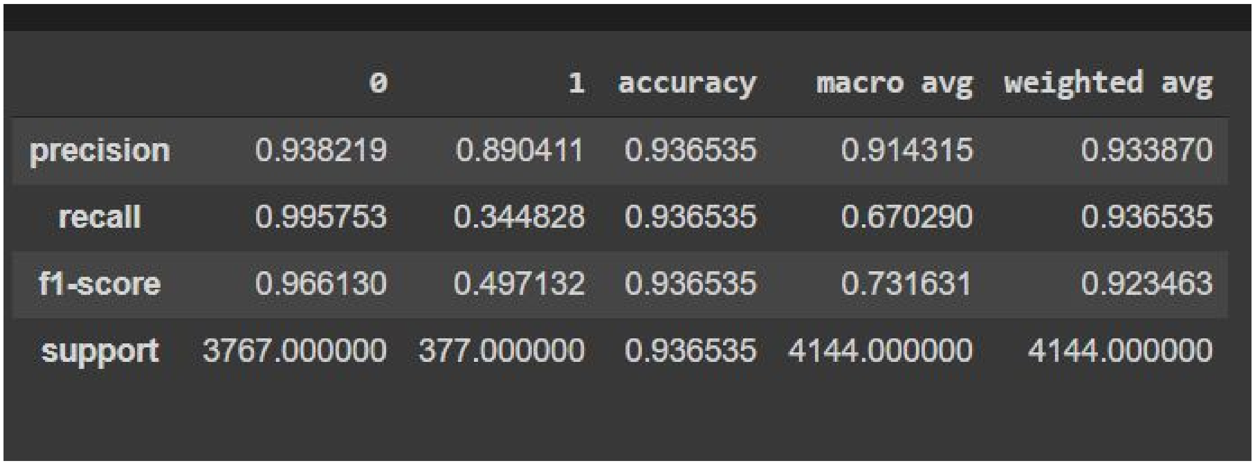
With the training data that was prepared earlier, we start with the modelling process. We consider the Random Forest [https://builtin.com/data-science/random-forest-algorithm] for all our experiments and observe the performance of the model for different sets of features.

For the above task, we will make use of Tree-based learning approaches, more specifically Random Forest. The reasons for that are -

1. The training data contains lots of missing values which can only be dealt with decision trees.
2. There are a lot of predictor variables and the relationship between the response and predictor isn't quite linear. This can easily be tackled by Tree-based methods.
3. Tree-based learning is quite explainable and helps in determining how important each of the predictor variables are.

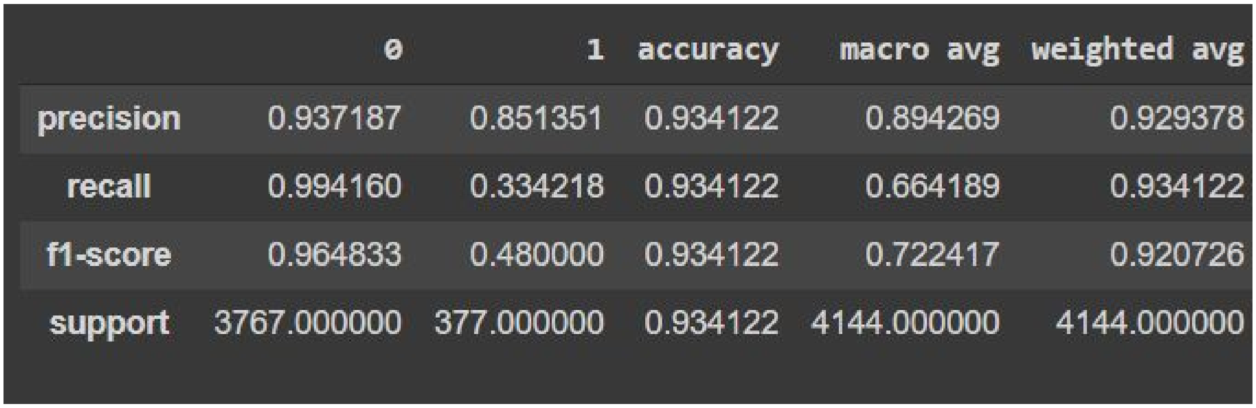
Before we start the modelling process, we change all the categorical features into numeric form by Label Encoding them. Now the entire training data is split into train, validation and test set for modelling. The models would be trained using Train data and will be tested again Validation set, the test set remains untouched. It is ensured that all sets follow the same distribution of the response variable in the original Training data.

I. The first model takes the train data, as it is, without resolving the class imbalance problem and also does not include the lag features that were created in the data preparation stage. This is the first attempt at modelling to gauge the initial performance.



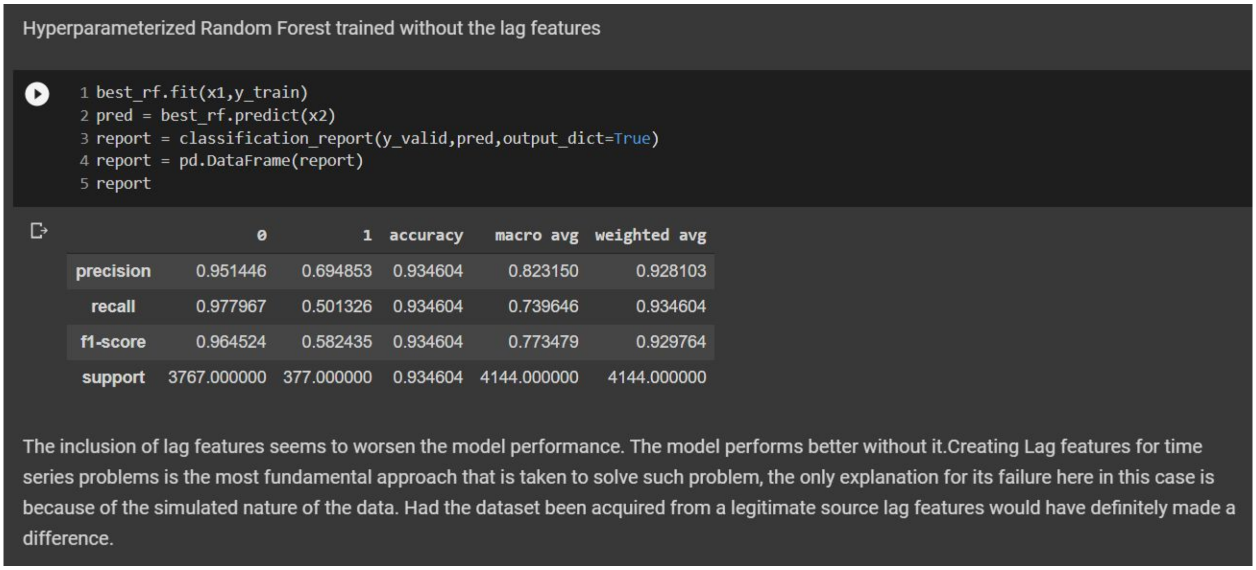
	0	1	accuracy	macro avg	weighted avg
precision	0.938219	0.890411	0.936535	0.914315	0.933870
recall	0.995753	0.344828	0.936535	0.670290	0.936535
f1-score	0.966130	0.497132	0.936535	0.731631	0.923463
support	3767.000000	377.000000	0.936535	4144.000000	4144.000000

II. High accuracy is observed but the overall F1-Score is low, which suggests that the model is doing well at identifying the majority class but doesn't cover all cases of foreclosure. Now we would try including lag features to see if there is any enhancement in performance.



	0	1	accuracy	macro avg	weighted avg
precision	0.937187	0.851351	0.934122	0.894269	0.929378
recall	0.994160	0.334218	0.934122	0.664189	0.934122
f1-score	0.964833	0.480000	0.934122	0.722417	0.920726
support	3767.000000	377.000000	0.934122	4144.000000	4144.000000

III. Addition of lag features didn't help the model at all, on the contrary, the model performance reduces indicated by the lower F1-Score Now we try to tune our random forest model to boost the performance and we repeat the same exercise of comparing the models with the newly tuned model.



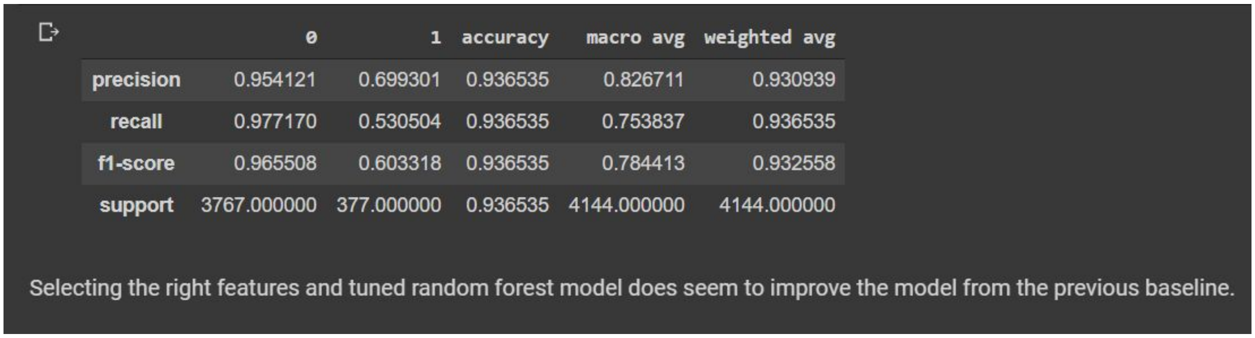
Hyperparameterized Random Forest trained without the lag features

```
best_rf.fit(x1,y_train)
pred = best_rf.predict(x2)
report = classification_report(y_valid,pred,output_dict=True)
report = pd.DataFrame(report)
report
```

	0	1	accuracy	macro avg	weighted avg
precision	0.951446	0.694853	0.934604	0.823150	0.928103
recall	0.977967	0.501326	0.934604	0.739646	0.934604
f1-score	0.964524	0.582435	0.934604	0.773479	0.929764
support	3767.000000	377.000000	0.934604	4144.000000	4144.000000

The inclusion of lag features seems to worsen the model performance. The model performs better without it.Creating Lag features for time series problems is the most fundamental approach that is taken to solve such problem, the only explanation for its failure here in this case is because of the simulated nature of the data. Had the dataset been acquired from a legitimate source lag features would have definitely made a difference.

IV. The random forest model was tuned and then it was trained on two separate train data, one which included lag features and the other that doesn't. It shows a similar performance like in the previous step, where the inclusion of lag features didn't help the cause. Now we would proceed in modelling without the use of lag features and try to reduce the number of predictors to find the sweet spot where we get great predictive power with a minimal number of features. This is done by making use of the Backward Feature Elimination technique.

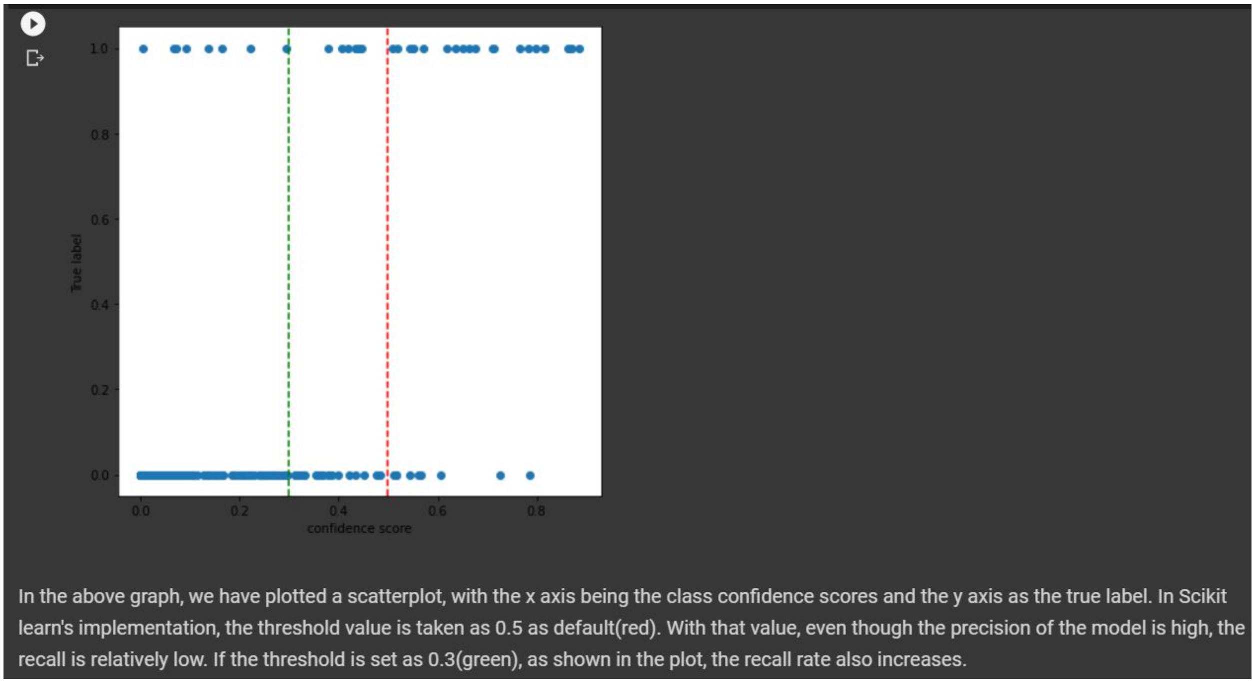


	0	1	accuracy	macro avg	weighted avg
precision	0.954121	0.699301	0.936535	0.826711	0.930939
recall	0.977170	0.530504	0.936535	0.753837	0.936535
f1-score	0.965508	0.603318	0.936535	0.784413	0.932558
support	3767.000000	377.000000	0.936535	4144.000000	4144.000000

Selecting the right features and tuned random forest model does seem to improve the model from the previous baseline.

V. Now an attempt is made to tackle the class imbalance problem. Several techniques are popular to resolve the issue, for example, upsampling of the minority class, downsampling majority class, introducing synthetic examples to increase more no. of minority cases etc. We upsampled the minority class with synthetic examples which are generated by an algorithm called SMOTE. Then the model is trained on this new data and the performance is analysed. The outcome of the experiment was that the overall Recall of foreclosed cases increased but the Precision decreased greatly. Hence upsampling did not help in the case.

VI. Out of all the variants of Random forest, the model with the optimal performance turned out to be the one where the right set of features were selected and the model tuned. But the performance could be improved, well past the 0.60 F1 score. The trick here is to increase the recall rate, which could be achieved by manipulating the threshold. What it means is that we could plot a graph to infer an optimal value, using which the model flags a certain case as a Foreclosure or not. This variant already has a decent precision rate of 0.699



In the above graph, we have plotted a scatterplot, with the x axis being the class confidence scores and the y axis as the true label. In Scikit learn's implementation, the threshold value is taken as 0.5 as default(red). With that value, even though the precision of the model is high, the recall is relatively low. If the threshold is set as 0.3(green), as shown in the plot, the recall rate also increases.

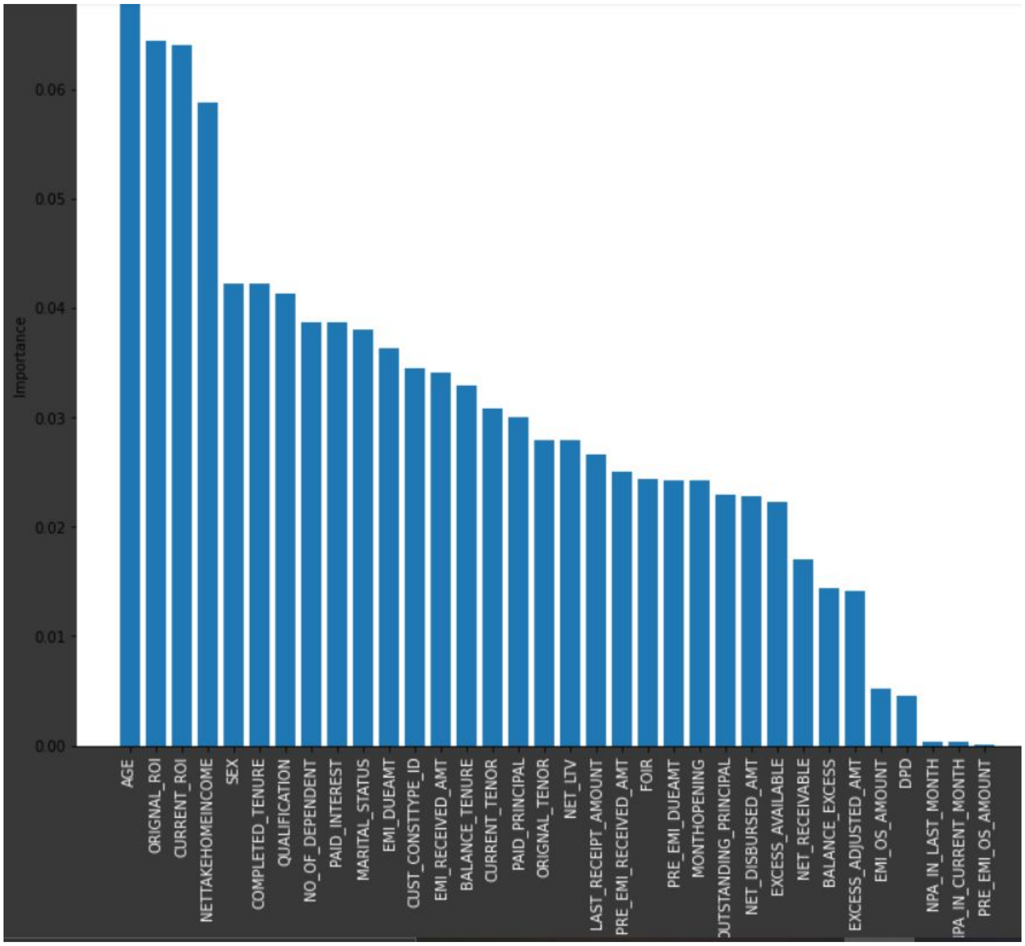
VII. The optimal threshold was inferred to be 0.3. Now keeping this as the new threshold, if the probability for foreclosure is above this value, an alarm can be raised and the particular loan application can be looked into by an expert to confirm. Another trade-off, which is visible in the plot, is that the new threshold increases the chances of false-positive cases, which is an acceptable caveat of the model.

8. Conclusion

From the above analysis and modelling, we have determined the following set of features that should be monitored closely, for every ongoing loan application, to anticipate whether a customer would foreclose his loan.

1. Age - The age of the customer plays the most important factor. Younger applicants do have the chance of clearing the loan than older ones.

2. Original & Current Roi - The change in the rate of interest over the period of loan repayment plays a vital role in determining loan foreclosure. This was confirmed by Hypothesis Testing, where it was concluded that there is a significant difference between Foreclosure and normal loan cases.



3. Net income - The most intuitive feature so far, Higher income means easier for the customer to repay the loan

4. Completed Tenure - How further along a customer is in his/her loan repayment journey also seems a logical indicator of foreclosure.

5. No of Dependent - A customer with a bigger family, means greater commitments on their behalf hence harder to repay the loan.

The result of this experiment can be improved with better feature engineering under the guidance of a subject matter expert and trying other types of Tree-based learning methods.