IME672 Group Project Summer 2022

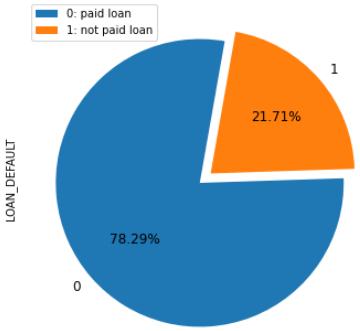
Group 9

Sahil Singh 200838
Sanskriti Sharma 200876
Sanober Ali 200872

Problem Title: Predicticting Loan Default (Classification)

Problem Description: The banking industry uses classification models to predict the defaulters or raise flags on customers who are likely to default. The dataset used has many problems initially, including outliers, missing data, proper model usage, class imbalance and so on. We have tried to tackle these problems here. The data set is preprocessed, then suitable machine learning models are applied and then the output is postprocessed.

Data Understanding:

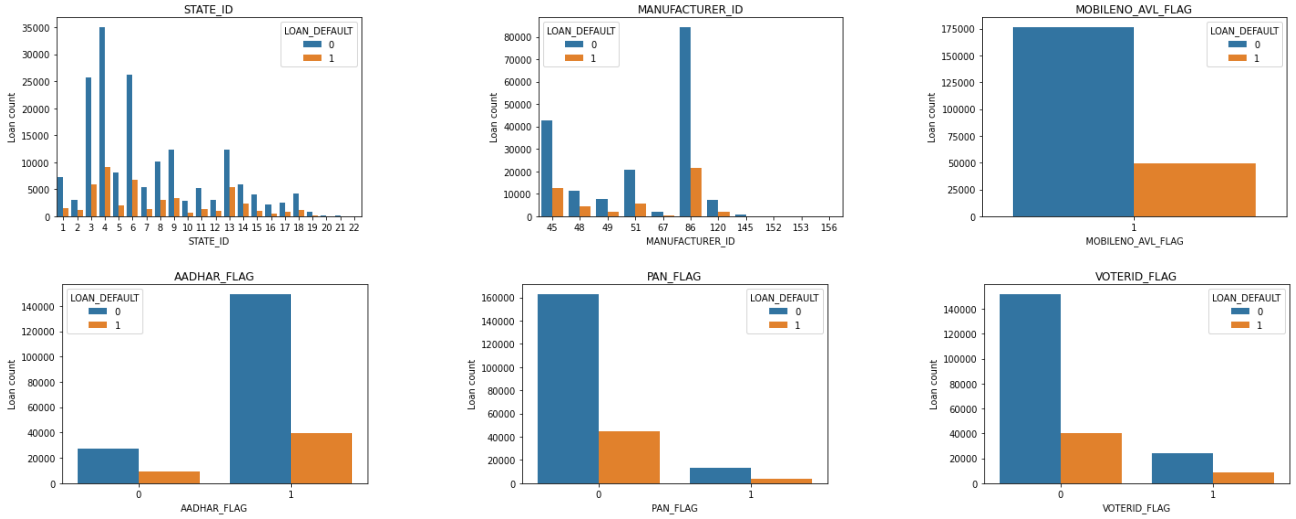


The given data-set has 40 attributes (leaving classifier) and around 233 thousand data of customers (rows). Most of the attributes are integer (or float) type. Few attributes like Employment Type, Date of Birth, Disbursal Date, Perform CNS Score Description are object type. The attribute Employment Type has 7661 missing data in count which accounts to 0.032858 percentage of the total data size.
The classifier ratio of the training set can be seen in *Figure*: This is a case of class imbalance with 78.29% negatives.
The statistics of the data are studied by grouping according to defaulters or non-defaulters in the code.

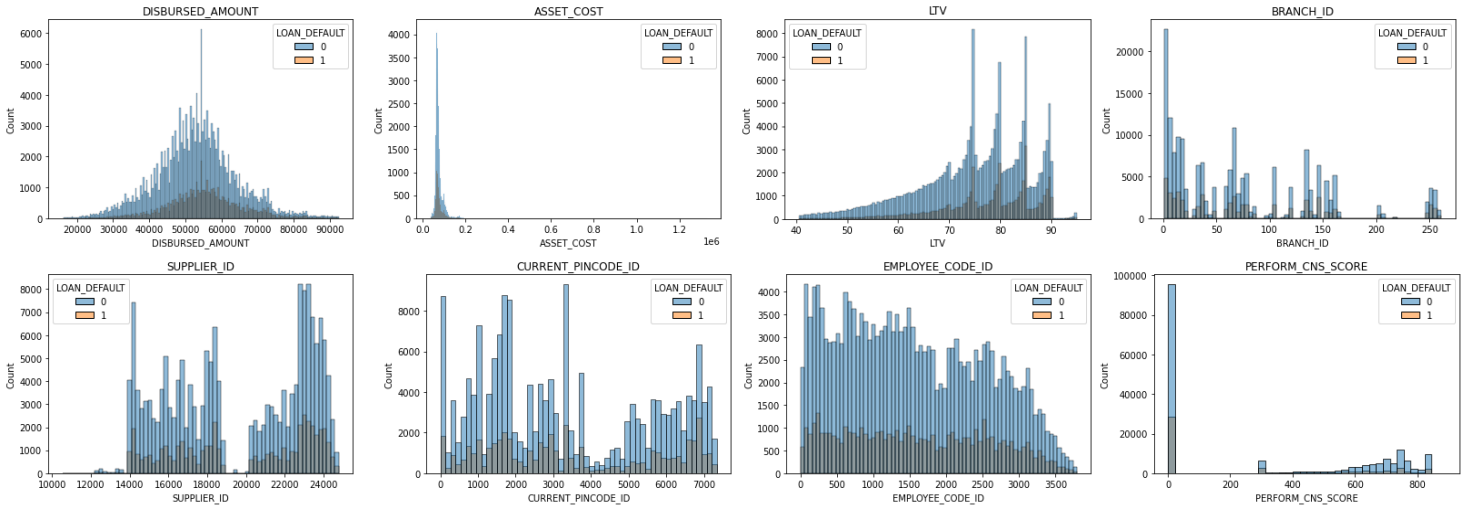
Data Preprocessing:

```
DISBURSED_AMOUNT
No. of observations in column:  225493
Statistics: Mean=54240.729, Std dev=12775.562
Identified outliers: 3036

LTV
No. of observations in column:  225493
Statistics: Mean=74.807, Std dev=11.442
Identified outliers: 2645

PERFORM_CNS_SCORE
No. of observations in column:  225493
Statistics: Mean=293.040, Std dev=338.874
Identified outliers: 0

PRI_NO_OF_ACCTS
No. of observations in column:  225493
Statistics: Mean=2.462, Std dev=5.223
Identified outliers: 3991

PRI_OVERDUE_ACCTS
No. of observations in column:  225493
Statistics: Mean=0.159, Std dev=0.553
Identified outliers: 6194

PRI_CURRENT_BALANCE
No. of observations in column:  225493
Statistics: Mean=168481.316, Std dev=951667.062
Identified outliers: 2152

PRI_SANCTIONED_AMOUNT
No. of observations in column:  225493
Statistics: Mean=222073.639, Std dev=2411716.168
Identified outliers: 592

PRI_DISBURSED_AMOUNT
No. of observations in column:  225493
Statistics: Mean=221609.814, Std dev=2414692.085
Identified outliers: 588
```

The important attributes like Disbursed Amount, Perform CNS Score, etc are made to go through a outlier removal algorithm. This function replaces the value of the outlier in these attributes to the mean of that attribute. The outlier data of these attributes can be seen in *Figure 2*.
Useless Features, like Date of Birth, IDs, etc which have no influence on the classifier are dropped from the data-set now. The attribute Disbursal Date (originally present as DD-MM-YYYY) is now converted into number of months and stored as integer attribute. The attributes Average Acct Age and Credit History Length (originally present as X yrs Y mon) are now converted to number of months till present date and stored as integer type.
After these operations, it is seen that the data now has two categorical attributes namely: Employment type and Perform CNS Score Description.
The missing values are dropped from the dataset. They constitute a very small percentage of the data and dropping them would not change the ratio of defaulters to non-defaulters.
The attributes (numerical) which have a unique values of less than 30 are studied here in the form of countplots. The following plots are shown below in *Figure* :
Some inferences from these countplots: Manufacturer ID 86 has most number of defaulters. People with Aahdar Flag as 1, Pan Flag as 0, Voter ID Flag as 0 are most likely to be defaulters. People in State IDs 3,4 and 6 are defaulting most.



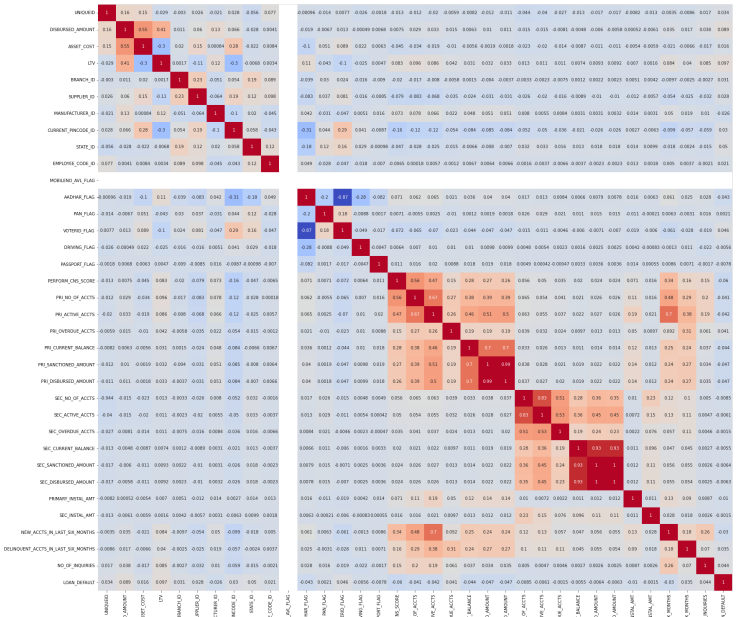
Bar charts are plotted for the continuous attributes as shown below in *Figure* :



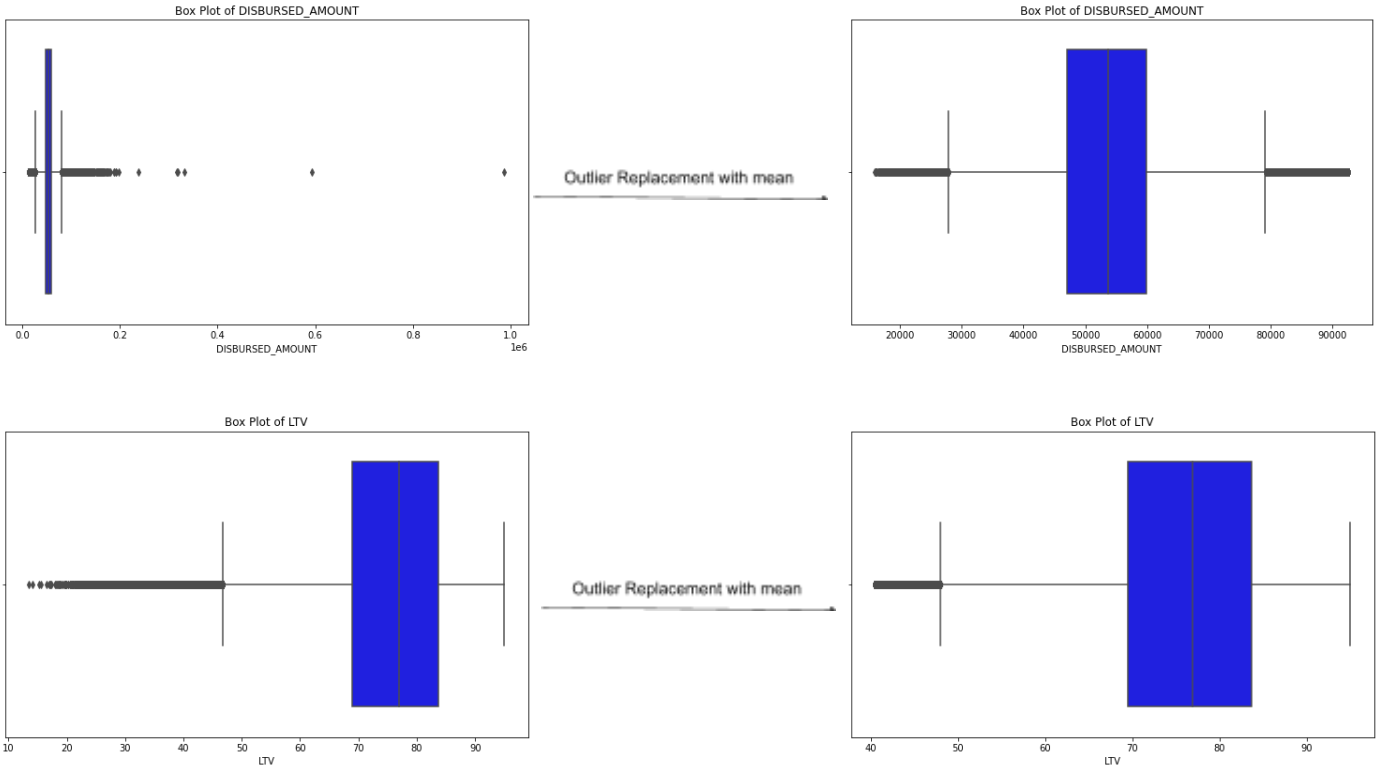
Most of the defaulters and non-defaulters are present in the middle of the range of the Disburbsed Amount. People with CNS Score 0 apply the most for loan and are mostly defaulters. Higher LTV value customers have increasing number of defaulters.

Heatmap can be plotted for studying the correlation between the attributes.

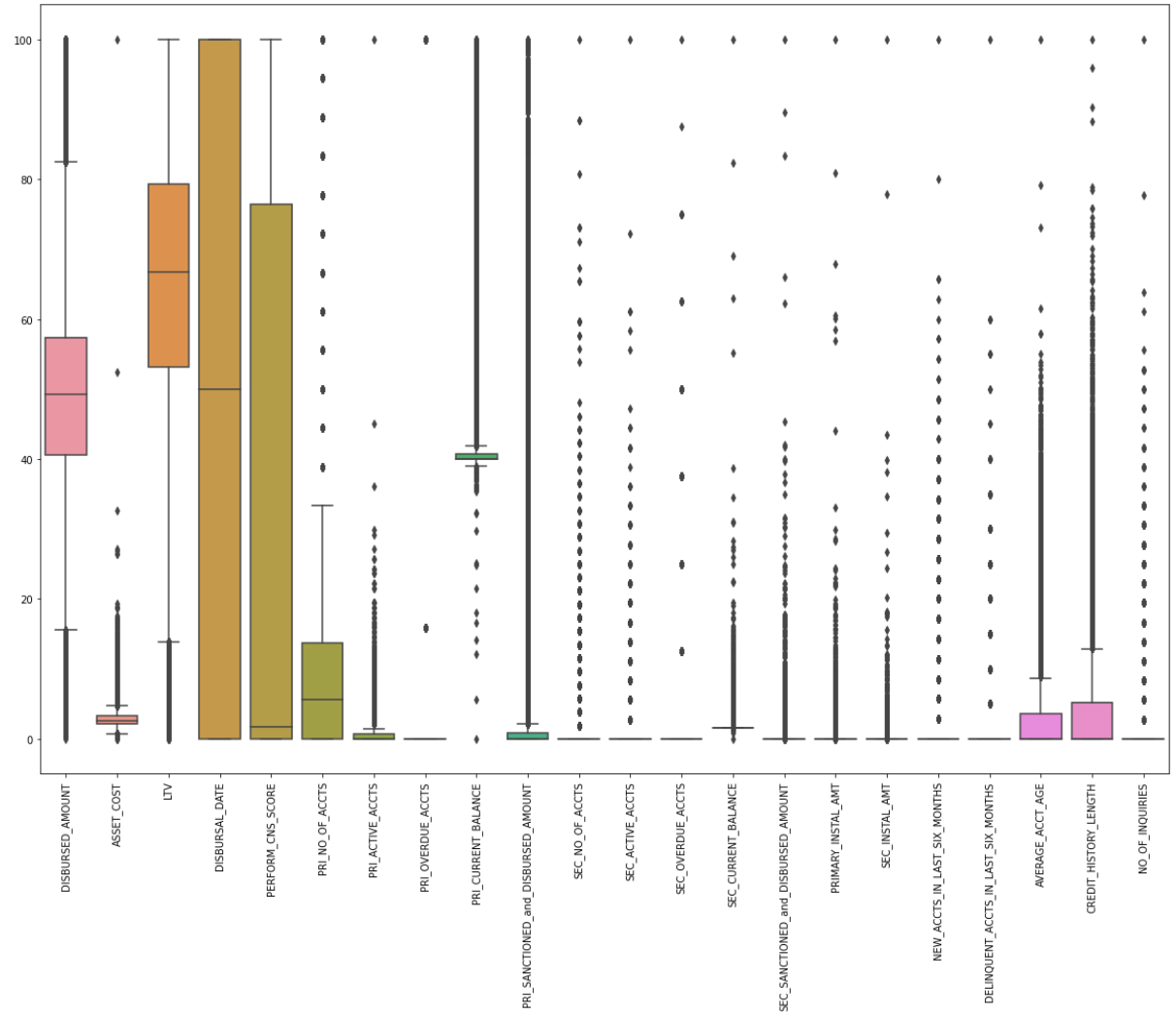
From the heatmap, it can be inferred that the attributes Primary Sanctioned Amount and Primary Disbursed Amount are highly correlated (same for Secondary). Hence we can club these two attributes as a single one. Mobile No, Aadhar Flag, Voter ID, Driving Flag, Passport Flag are just for book-keeping purpose. Disbursed Amount, Sanctioned Amount and Current Balance are also correlated to an appreciable extent.



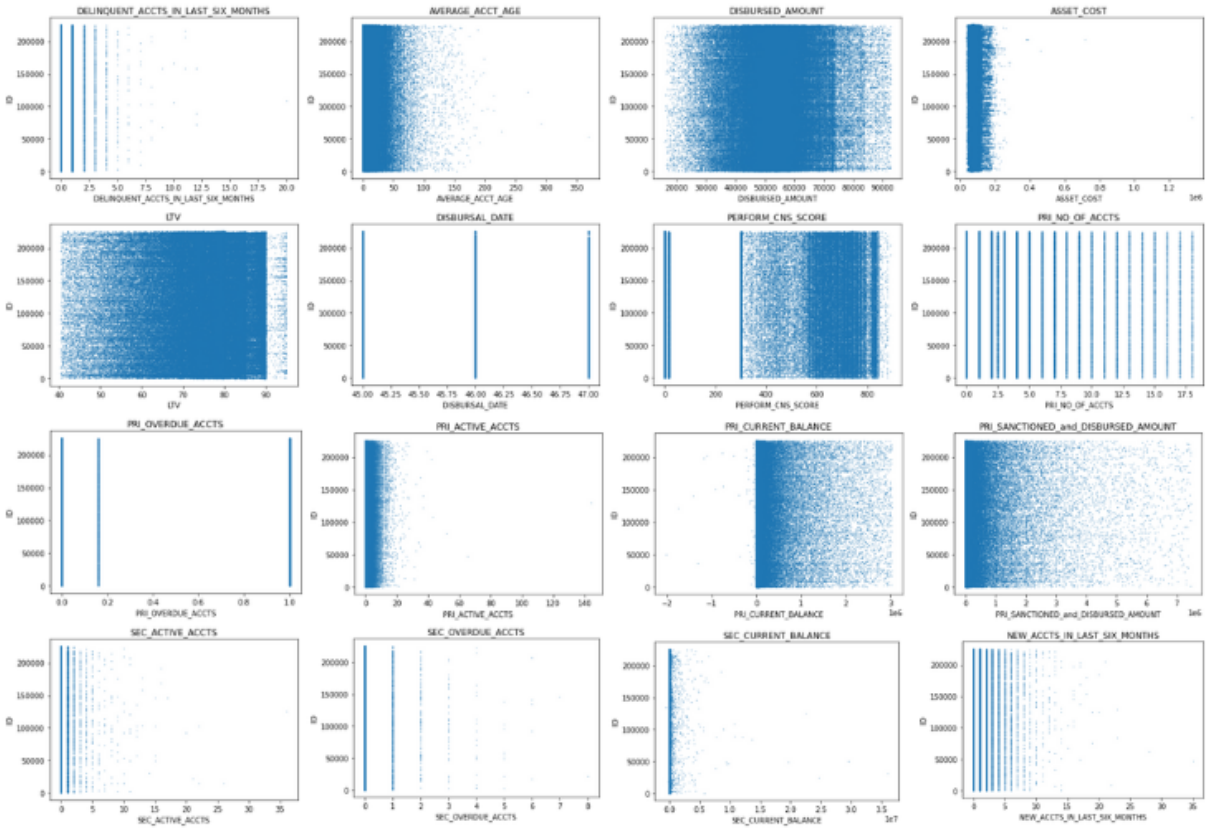
Indivizual Boxplots of the attributes whose outliers where replaced with mean are plotted:



The attributes of the data are now normalized using Min-Max technique with range [0,1]. The combined boxplot is now plotted in *Figure*.



ScatterPlots of attributes are shown below in *Figure 8*. Patterns can be seen in these plots.



Chi-squared test is performed on Employment Type and Loan Defaulters (class label). The contingency table can be seen in *Figure 9*. The chi-squared value is approximately 1.30387. It can be said that both these attributes are not related.

Calculated Table

	Salaried	Self Employed	Total
Loan Default	537	941	1478
Loan Not Default	2112	3453	5565
Total	2649	4394	7043

Expected Table

	Salaried	Self Employed	Total
Loan Default	555.902598	922.097402	1478.0
Loan Not Default	2093.097402	3471.902598	5565.0
Total	2649.000000	4394.000000	7043.0

Chi Squared value = 1.3038705107692794

The attributes Employment Type and Perform CNS Score Description are changed to binary data by adding dummies of all the unique elements in them. The data is then split into training and test set (test size =0.25).

Model Building, Evaluation and Results:

8 different models are applied on the data set.

Random Forest

Confusion Matrix

	Positive	Negative
Positive	41518	2606
Negative	11125	1125

Accuracy 75.6430269273069
F1 Score 0.14079219072648771
Recall Score 0.09183673469387756
Balanced Accuracy Score 0.5163879530599295

Logistic Regression

Confusion Matrix

	Positive	Negative
Positive	44124	0
Negative	12250	0

Accuracy 100
F1 Score 0.0
Recall Score 0.0
Balanced Accuracy Score 0.5

Linear Regression

Confusion Matrix

	Positive	Negative
Positive	44099	25
Negative	12238	12

Accuracy 78.0
F1 Score 0.0019532839586554893
Recall Score 0.0009795918367346938
Balanced Accuracy Score 0.5002065033791597

For Linear Regression, the output classifier is treated as such: if it is [0,0.5] it is labeled as 0; if it is (0.5,1] it is labelled as 1.

Gradient Descent

Confusion Matrix

	Positive	Negative
Positive	20939	23185
Negative	4895	7355

Accuracy 50.0
F1 Score 0.34377190932460855
Recall Score 0.6004081632653061
Balanced Accuracy Score 0.5374785807714437

Naive Bayes

Confusion Matrix

	Positive	Negative
Positive	6316	37808
Negative	1128	11122

Accuracy 31.0
F1 Score 0.3635828702190258
Recall Score 0.9079183673469388
Balanced Accuracy Score 0.5255302107788995

Decision Tree

Confusion Matrix

	Positive	Negative
Positive	34446	9678
Negative	9044	3206

Accuracy 67.0
F1 Score 0.2551125964828519
Recall Score 0.26171428571428573
Balanced Accuracy Score 0.5211889350790629

Neural Network

Confusion Matrix

	Positive	Negative
Positive	44123	1
Negative	12250	0

Accuracy 78.0
F1 Score 0.0
Recall Score 0.0
Balanced Accuracy Score 0.4999886682984317

Ada Boost

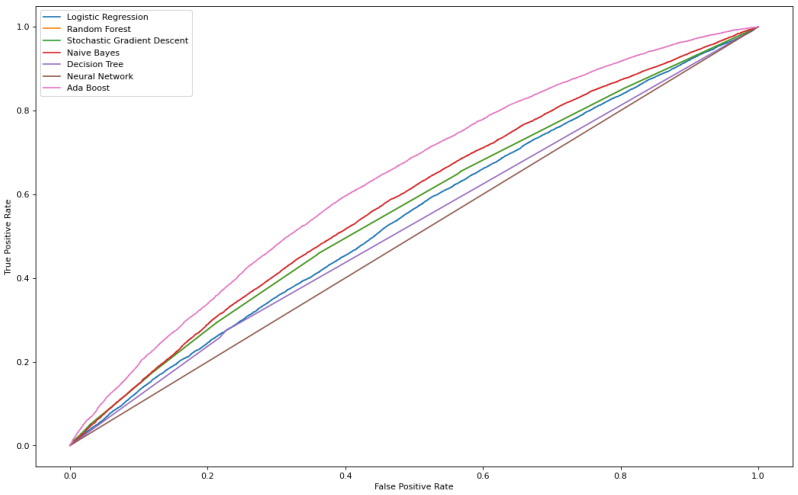
Confusion Matrix

	Positive	Negative
Positive	44014	110
Negative	12166	84

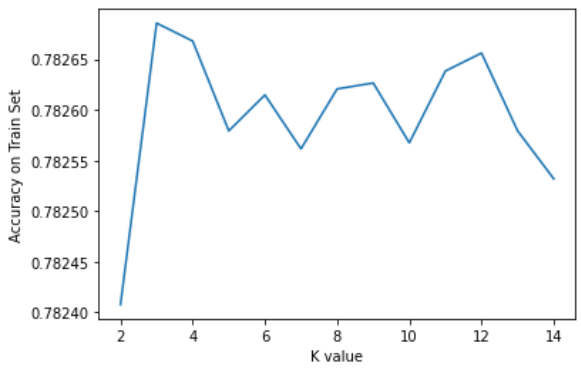
Accuracy 78.0
F1 Score 0.013500482160077145
Recall Score 0.006857142857142857
Balanced Accuracy Score 0.5021820842560576

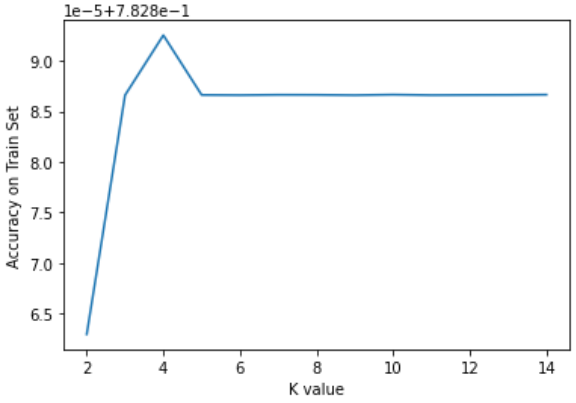
ROC curve have been plotted to understand how the models perform comparatively in *Figure*. It can be seen that the Ada Boost model is the most efficient in this case. The decreasing order of the model's efficiency is:

a. Ada Boost
b. Naive Bayes
c. Stochastic Gradient Descent
d. Random Forest
e. Logistic Regression
f. Decision Tree
g. Neural Network

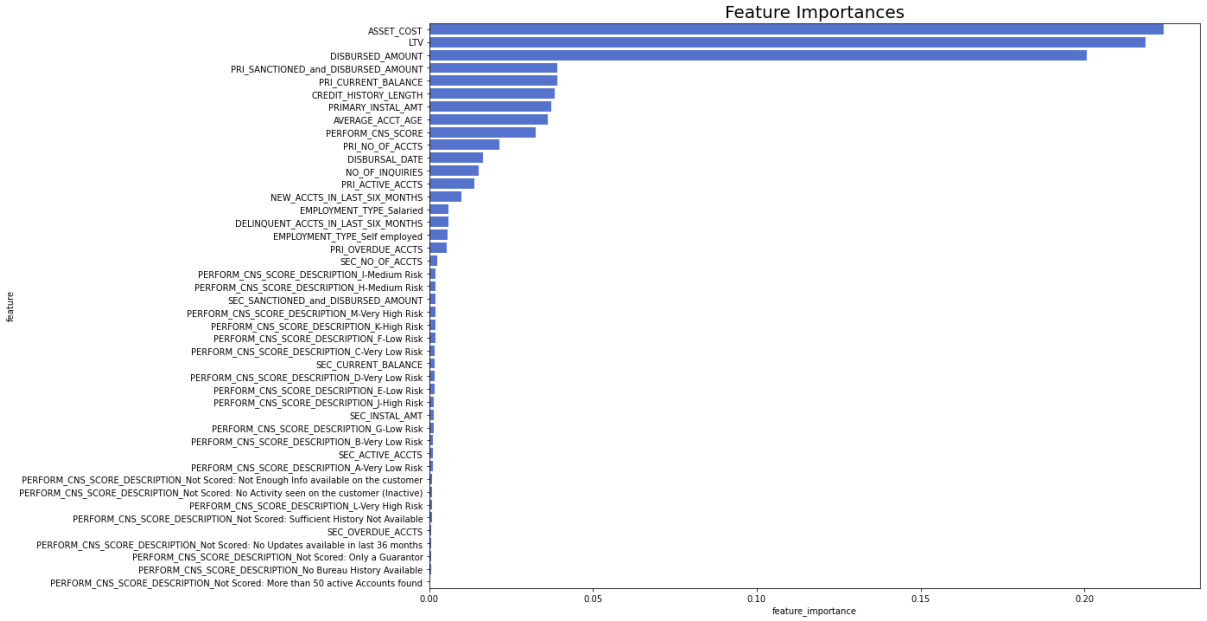


K-Fold Cross Validation can be applied. We have chosen Ada Boost (as it is already giving good results) and Logistic Regression to apply the same and to figure out the value of k: a range of K is chosen (2,15) and the K with the maximum accuracy is chosen. *Figure 12* shows the plot of Accuracy vs K values. Maximum accuracy value 78.27% is found out for a K = 3 for Ada Boost. Maximum accuracy value 78.30% is found out for a K = 4 for Logistic Regression.Left Figure: Ada Boost Right Figure: Logistic Regression





The features importance can be calculated using one of the models (which supports features_importance feature). Random Forest is used for the same here. *Figure below* shows the importance of the features of the data set comparatively. It can be seen that Asset Cost, LTV, Disbursed Amount are very important features while Sanctioned and Disbursed Amount, Current Balance, Credit History Length, Acct Age, Perform CNS Score are also important.



Our data is highly class imbalanced. SMOTE or Synthetic Minority Oversampling Technique is used to create synthetic data. SMOTE uses a nearest neighbors algorithm to generate new and synthetic data we can use for training our model.
The output details of the model using SMOTE are shown below.

Random Forest

```
[[38615  5509]
 [10049  2201]]
72.0
Accuracy of model  0.7240217121368007
F1 Score  0.22054108216432866
Recall Score  0.1796734693877551
Balanced Accuracy Score  0.5274103907540716
```

Logistic Regression

```
[[21427 22697]
 [ 4568  7682]]
52.0
Accuracy of model  0.5163550572959166
F1 Score  0.36041192615355744
Recall Score  0.6271020408163265
Balanced Accuracy Score  0.556355389912288
```

Gradient Descent

```
[[25271 18853]
 [ 7506  4744]]
53.0
Accuracy of model  0.5324262958101252
F1 Score  0.2646804474572489
Recall Score  0.38726530612244897
Balanced Accuracy Score  0.4799960833939232
```

Naive Bayes

```
[[ 3581 40543]
 [  626 11624]]
27.0
Accuracy of model  0.26971653599176926
F1 Score  0.3608985205768664
Recall Score  0.9488979591836735
Balanced Accuracy Score  0.5150278029079459
```

Decision Tree

```
[[37087  7037]
 [ 9724  2526]]
70.0
Accuracy of model  0.7026820874871395
F1 Score  0.23160500618896984
Recall Score  0.20620408163265305
Balanced Accuracy Score  0.5233608568801467
```

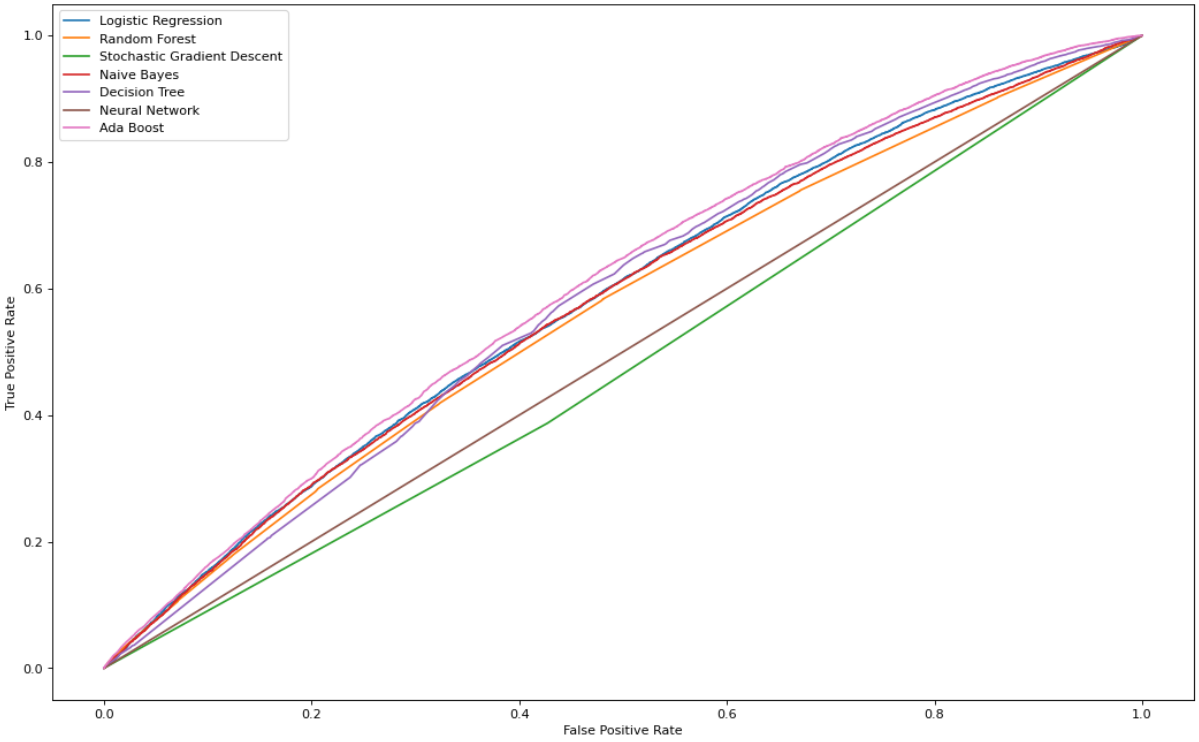
Neural Network

```
[[44120     4]
 [12250     0]]
78.0
Accuracy of model  0.7826302905594777
F1 Score  0.0
Recall Score  0.0
Balanced Accuracy Score  0.4999546731937268
```

Ada Boost

```
[[39835  4289]
 [10305  1945]]
74.0
Accuracy of model  0.7411217937347004
F1 Score  0.21045228305561567
Recall Score  0.15877551020408162
Balanced Accuracy Score  0.53078608707557
```

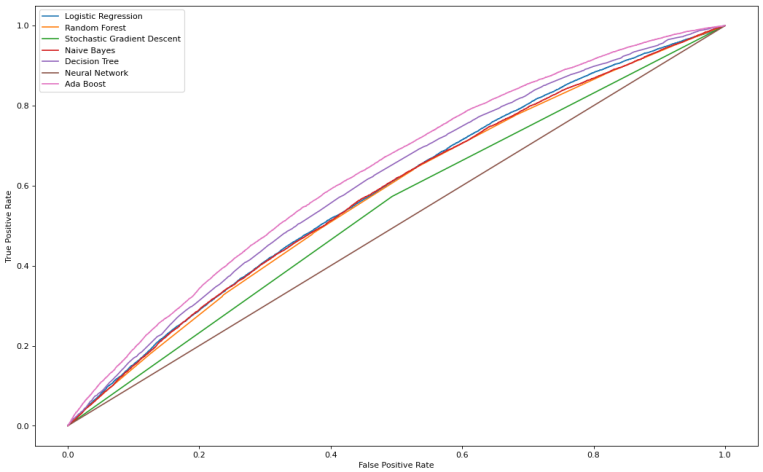
From Figure above, it can be seen that the although the accuracy of the models using SMOTE is decreasing, but they are classifying defaults better. The F1 score has improved. Although the accuracy of Ada Boost has fallen by 4%, but the F1 score has increased by 19.69% and the balanced accuracy score has increased by 2.86%. The SMOTE Ada Boost performs better than the class imbalanced models.

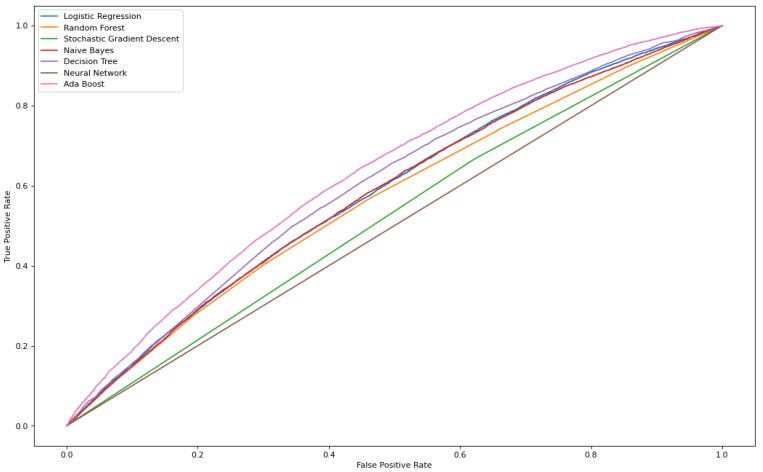


ROC curve have been plotted to understand how the models perform comparatively in *Figure above* using the SMOTE class balance method. It can be seen that the Ada Boost model is still the most efficient in this case. The overall accuracy of the model has increased than the imbalanced data set. The decreasing order of the model's efficiency is:

a. Ada Boost
b. Decision Tree
c. Logistic Regression
d. Naive Bayes
e. Random Forest
f. Neural Network
g. Stochastic Gradient Descent

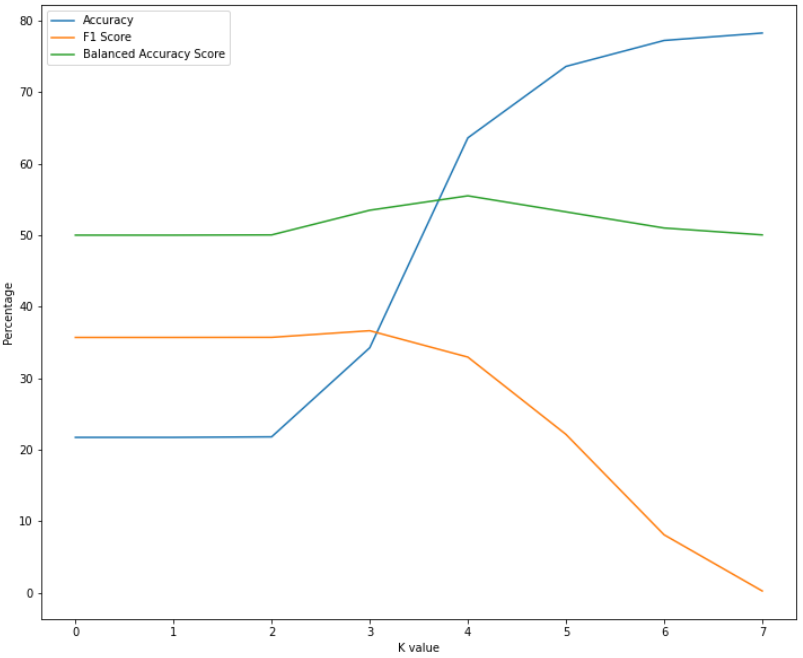
Downsampling and Upsampling can also be applied to the data-set. Figure below shows ROC curve for downsampling and Figure 18 shows the same for upsampling.





It is seen that Ada Boost performs better again with both Downsampling and Upsampling.

Out of the 7 models used after SMOTE, we can develop an algorithm which generates a 1 if there are atleast K number of 1s in the prediction of the models. The value of K can range from 0 to 7 as there are 7 models used here. The value of K which gives the most efficient model has to be found. Figure shows the different paramters on varying the K value. From the figure it can be inferred that K=4 gives the best balanced accuracy score. Also, the F1 score is also second highest for K=4. K=4 means that we will classify a tuple as 1 if the maximum of the 7 models (atleast 4) predict the output as 1. This is in conjunction with our intuition.



From the different models deployed and the tests performed, we can conclude that the Ada Boost algorithm on SMOTE data performs the best.