Micro-Credit Defaulter Model

Submitted by:

Aditya Maurya

ACKNOWLEDGMENT

I would like to express my very great appreciation to Mr Mohd Kashif. for his valuable and constructive suggestions during the planning and development of this research work. His willingness to give his time so generously has been very much appreciated.

Research papers that helped me in this project was as follows:

- https://www.researchgate.net/publication/336800562_Credit_Card_Fraud_Detection_using_Machine_Learning_and_Data_Science
- https://www.academia.edu/44389277/Microfinance_in_Bangladesh_A_Case_Study_on_Islamic_Microfinance

Articles that helped me in this project was as follows:

- https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/
- https://medium.com/kitepython/handling-imbalanced-datasets-with-smote-in-python-a94090d031f0

INTRODUCTION

Business Problem Framing

This project was highly motivated project as it includes the real time problem for Microfinance Institution (MFI), and to the poor families in remote areas with low income, MFI to provide micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be defaulter if he deviates from the path of paying back the loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah), while, for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah).

Conceptual Background of the Domain Problem

Generally, Credit Scores plays a vital role for loan approvals, and is very important in today's financial analysis for an individual, Most of the loan lending vendors rely heavily on it, so in our case users has 5 days' time to pay back the loan or else they are listed as defaulters which will impact the loan the credit score heavily, so there are few thing to lookout in this dataset as users who are taking extensive loans, user who have most frequent recharges in their main account have a good chance of 100% payback rate, and user who never recharged their main account for them loan should have never been approved as there is high chance for single user or default user taking multiple connections in name or documents of the family members.

Review of Literature

The project objective is to find out the defaulters (i.e. the users who don't repay the loan within 5 days). Now, Using Different Mathematical and statistical tools
Many assumptions regarding the data is made and data Cleaning is done.
After the Data Cleaning part Model Training takes place in which different models like: KNN, Random Forest Classifier, Decision Tree Classifier Ada Boost Classifier, Gradient Boosting Classifier etc. models are used for the Training of the data.
After Training of the data Hyper-parameter tuning is done and then the best model is designed.

Motivation for the Problem Undertaken

This project was highly motivated project as it includes the real time problem for Microfinance Institution (MFI), and to the poor families in remote areas with low income, and it is related to financial sectors, as I believe that with growing technologies and Idea can make a difference, there are so much in the financial market to explore and analyse and with Data Science the financial world becomes more interesting.

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

- This problem is a classification problem, the target variable is itself a statistical parameter. We have to predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of insurance of loan. In this case, Label ‘1’ indicates that the loan has been payed i.e. Non- defaulter, while, Label ‘0’ indicates that the loan has not been payed .for a loan amount of 5 payback amount should be 6,and for loan amount of 10 payback amount is 12.

	count	mean	std	min	25%	50%	75%	max
label	209593.0	0.875177	0.330519	0.000000	1.000	1.000000	1.00	1.000000
aon	209593.0	8112.343445	75696.082531	-48.000000	246.000	527.000000	982.00	999860.755168
daily_decr30	209593.0	5381.402289	9220.623400	-93.012667	42.440	1469.175667	7244.00	265926.000000
daily_decr90	209593.0	6082.515068	10918.812767	-93.012667	42.692	1500.000000	7802.79	320630.000000
rental30	209593.0	2692.581910	4308.586781	-23737.140000	280.420	1083.570000	3356.94	198926.110000
rental90	209593.0	3483.406534	5770.461279	-24720.580000	300.260	1334.000000	4201.79	200148.110000
last_rech_date_ma	209593.0	3755.847800	53905.892230	-29.000000	1.000	3.000000	7.00	998650.377733
last_rech_date_da	209593.0	3712.202921	53374.833430	-29.000000	0.000	0.000000	0.00	999171.809410
last_rech_amt_ma	209593.0	2064.452797	2370.786034	0.000000	770.000	1539.000000	2309.00	55000.000000
cnt_ma_rech30	209593.0	3.978057	4.256090	0.000000	1.000	3.000000	5.00	203.000000
fr_ma_rech30	209593.0	3737.355121	53643.625172	0.000000	0.000	2.000000	6.00	999606.368132
sumamnt_ma_rech30	209593.0	7704.501157	10139.621714	0.000000	1540.000	4628.000000	10010.00	810096.000000
medianamnt_ma_rech30	209593.0	1812.817952	2070.864620	0.000000	770.000	1539.000000	1924.00	55000.000000
medianmarechprebal30	209593.0	3851.927942	54006.374433	-200.000000	11.000	33.900000	83.00	999479.419319
cnt_ma_rech90	209593.0	6.315430	7.193470	0.000000	2.000	4.000000	8.00	336.000000
fr_ma_rech90	209593.0	7.716780	12.590251	0.000000	0.000	2.000000	8.00	88.000000
sumamnt_ma_rech90	209593.0	12396.218352	16857.793882	0.000000	2317.000	7226.000000	16000.00	953036.000000
medianamnt_ma_rech90	209593.0	1864.595821	2081.680664	0.000000	773.000	1539.000000	1924.00	55000.000000
medianmarechprebal90	209593.0	92.025541	369.215658	-200.000000	14.600	36.000000	79.31	41456.500000
cnt_da_rech30	209593.0	262.578110	4183.897978	0.000000	0.000	0.000000	0.00	99914.441420
fr_da_rech30	209593.0	3749.494447	53885.414979	0.000000	0.000	0.000000	0.00	999809.240107
cnt_da_rech90	209593.0	0.041495	0.397556	0.000000	0.000	0.000000	0.00	38.000000
fr_da_rech90	209593.0	0.045712	0.951386	0.000000	0.000	0.000000	0.00	64.000000
cnt_loans30	209593.0	2.758981	2.554502	0.000000	1.000	2.000000	4.00	50.000000
amnt_loans30	209593.0	17.952021	17.379741	0.000000	6.000	12.000000	24.00	306.000000
maxamnt_loans30	209593.0	274.658747	4245.264648	0.000000	6.000	6.000000	6.00	99864.560864
medianamnt_loans30	209593.0	0.054029	0.218039	0.000000	0.000	0.000000	0.00	3.000000
cnt_loans90	209593.0	18.520919	224.797423	0.000000	1.000	2.000000	5.00	4997.517944
amnt_loans90	209593.0	23.645398	26.469861	0.000000	6.000	12.000000	30.00	438.000000
maxamnt_loans90	209593.0	6.703134	2.103864	0.000000	6.000	6.000000	6.00	12.000000
medianamnt_loans90	209593.0	0.046077	0.200692	0.000000	0.000	0.000000	0.00	3.000000
payback30	209593.0	3.398826	8.813729	0.000000	0.000	0.000000	3.75	171.500000
payback90	209593.0	4.321485	10.308108	0.000000	0.000	1.666667	4.50	171.500000

- From the above statistical summary of the above part of the dataset, **the important thing is that** Some features even have negative values like the age on cellular network, main account last recharge date, data account last recharge date. Negative values in these features make no sense thus these values should be removed.

- The Dataset we are having, consists of some features giving information about the user for the time span of 30 days and 90 days. According to me if we have data of large number of days for a particular user then we could interpret User's behaviour more precisely because many users have the tendency of repeating the same things. Thus the features having the data with a time span of 90 days gives more information about the user as compared to the features with a time span of 30 days.

- All the categories that is being made to make the visualizations easy are solemnly based on the Description i.e. statistical summary of the data plotted above ***for instance*** low comes under (0-25%), average comes under(25-75%) and high comes over 75% of the data values in a given feature.

- Using MS EXCEL I have found the maximum values a feature can have, beyond these values the values are unimaginable.

  *****(for an example beyond the value [2500], the very next value in "aon" feature comes out to be around 2379 years, which means a user is using the telephone services from 359 BCE which is clearly not possible).*****

- I checked the correlation of the independent and dependent features and from the correlation table it is also clear that the features with time span of 30 and 90 days almost have the same correlation thus we can drop one for the same information.

Data Sources and their formats

- **label :** Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure}
- **msisdn :** mobile number of user
- **aon :** age on cellular network in days
- **daily_decr30:** Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)
- **daily_decr90:** Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)
- **rental30:** Average main account balance over last 30 days
- **rental90:** Average main account balance over last 90 days
- **last_rech_date_ma:** Number of days till last recharge of main account
- **last_rech_date_da:** Number of days till last recharge of data account
- **last_rech_amt_ma:** Amount of last recharge of main account (in Indonesian Rupiah)
- **cnt_ma_rech30:** Number of times main account got recharged in last 30 days
- **fr_ma_rech30:** Frequency of main account recharged in last 30 days
- **sumamnt_ma_rech30:** Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)
- **medianamnt_ma_rech30:** Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)
- **medianmarechprebal30:** Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)
- **cnt_ma_rech90:** Number of times main account got recharged in last 90 days
- **fr_ma_rech90:** Frequency of main account recharged in last 90 days
- **sumamnt_ma_rech90 :** Total amount of recharge in main account over last 90 days (in Indian Rupee)

- **medianamnt_ma_rech90:** Median of amount of recharges done in main account over last 90 days at user level (in Indian Rupee)
- **medianmarechprebal90:** Median of main account balance just before recharge in last 90 days at user level (in Indian Rupee)
- **cnt_da_rech30:** Number of times data account got recharged in last 30 days
- **fr_da_rech30:** Frequency of data account recharged in last 30 days
- **cnt_da_rech90:** Number of times data account got recharged in last 90 days
- **fr_da_rech90:** Frequency of data account recharged in last 90 days
- **cnt_loans30:** Number of loans taken by user in last 30 days
- **amnt_loans30:** Total amount of loans taken by user in last 30 days
- **maxamnt_loans30:** maximum amount of loan taken by the user in last 30 days
- **medianamnt_loans30:** Median of amounts of loan taken by the user in last 30 days
- **cnt_loans90:** Number of loans taken by user in last 90 days
- **amnt_loans90:** Total amount of loans taken by user in last 90 days
- **maxamnt_loans90:** maximum amount of loan taken by the user in last 90 days
- **medianamnt_loans90:** Median of amounts of loan taken by the user in last 90 days
- **payback30:** Average payback time in days over last 30 days
- **payback90:** Average payback time in days over last 90 days
- **pcircle:** telecom circle
- **pdate:** date

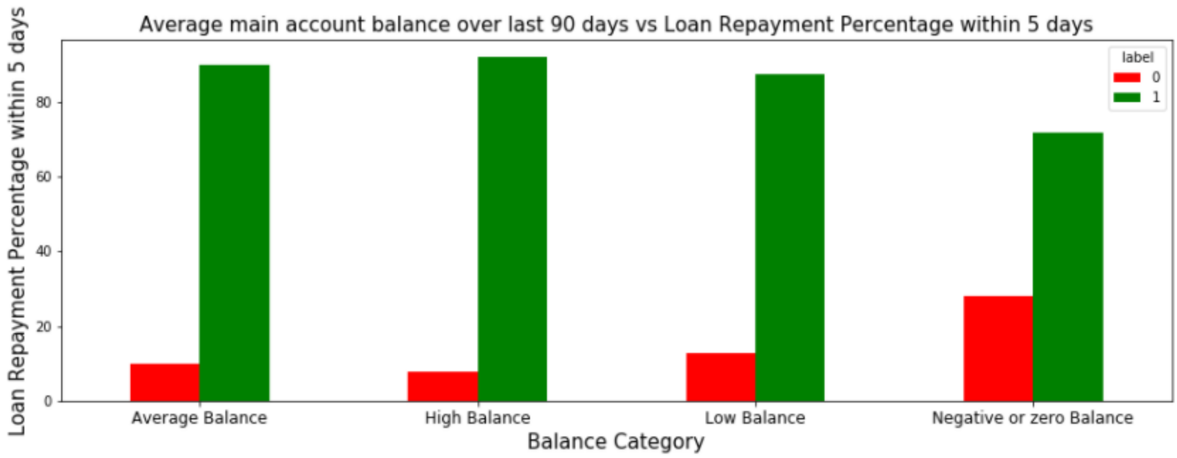
```
Data Types of Features :
 Unnamed: 0                       int64
label                            int64
msisdn                          object
aon                            float64
daily_decr30                   float64
daily_decr90                   float64
rental30                       float64
rental90                       float64
last_rech_date_ma              float64
last_rech_date_da              float64
last_rech_amt_ma                 int64
cnt_ma_rech30                    int64
fr_ma_rech30                   float64
sumamnt_ma_rech30              float64
medianamnt_ma_rech30           float64
medianmarechprebal30           float64
cnt_ma_rech90                    int64
fr_ma_rech90                     int64
sumamnt_ma_rech90                int64
medianamnt_ma_rech90           float64
medianmarechprebal90           float64
cnt_da_rech30                  float64
fr_da_rech30                   float64
cnt_da_rech90                    int64
fr_da_rech90                     int64
cnt_loans30                      int64
amnt_loans30                     int64
maxamnt_loans30                float64
medianamnt_loans30             float64
cnt_loans90                    float64
amnt_loans90                     int64
maxamnt_loans90                  int64
medianamnt_loans90             float64
payback30                      float64
payback90                      float64
pcircle                         object
pdate                   datetime64[ns]
dtype: object

Dataset contains any NaN/Empty cells :  False
```

Data Pre-processing Done

- I checked the correlation of the independent and dependent features and from the correlation table it is also clear that the features with time span of 30 and 90 days almost have the same correlation thus we can drop one for the same information.
- There were data for 30 and 90 days, so considering data for 90 days is adding more information rather than then data of 30 days.
- Some features can't have any negative value, so those features were treated accordingly.
- Outliers are treated manually for the features giving some important information, and then the threshold values were set to make the data free from outliers.
- Data lost is very less i.e **5.9%** which is less than the 7% which was stated in the documentation.
- Applied SMOTETomek, to balance the dataset as the dataset was imbalanced dataset.
- Applied StandardScaler to our dependent features.
- Applied various machine learning model and compared it.
- Applied hyper tuning several models, but couldn't achieve much better results.
- Saving final predictions in file.csv format.

Data Inputs- Logic- Output Relationships

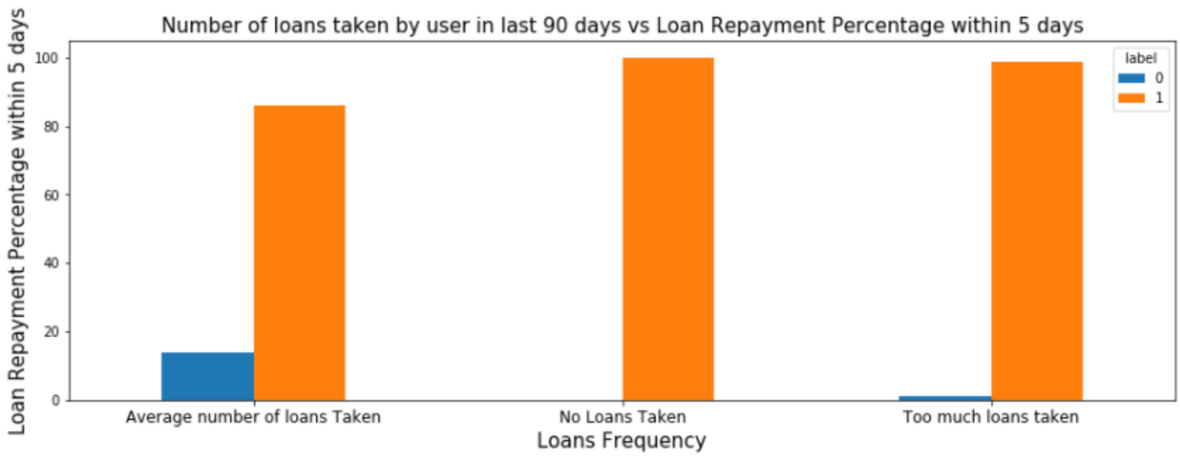
i) Average main account balance over last 90 days vs Loan Repayment Percentage within 5 days



From the above Graph and the crosstab table it is clear that:

1) 28% of Users having negative or zero balance are defaulters, which is very high.
2) 10% to 12% Users are defaulters which falls in the category of Average and Low balance category.
3) Users having high balance and are defaulters are very less in number

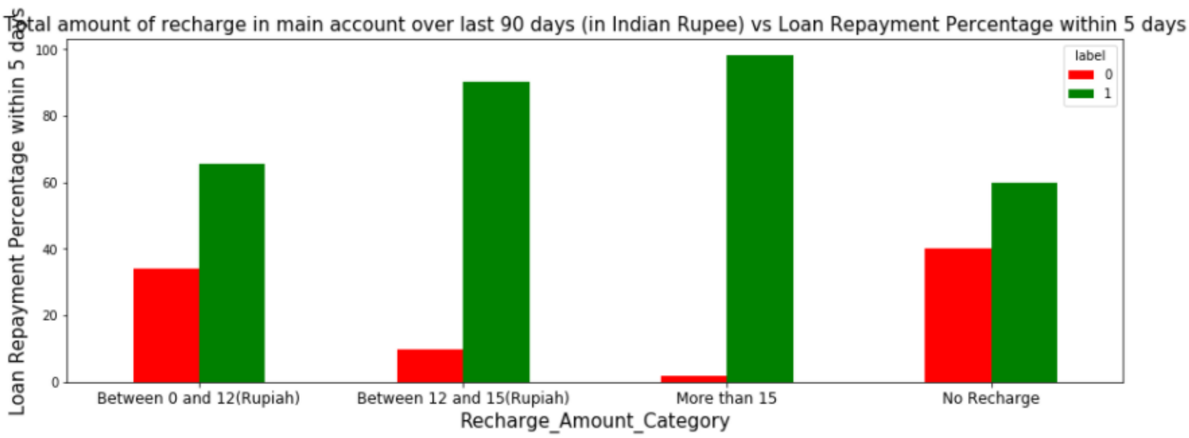
ii) Number of loans taken by user in last 90 days vs Loan Repayment Percentage within 5 days



From the above graph it is clear that:

1) Users who take more number of loans are non-defaulters (i.e. 98% of the category) as they repays the loan within the given time i.e. 5 days.
2) 14% of the Users are are among the average number of loan taken category are defaulters.

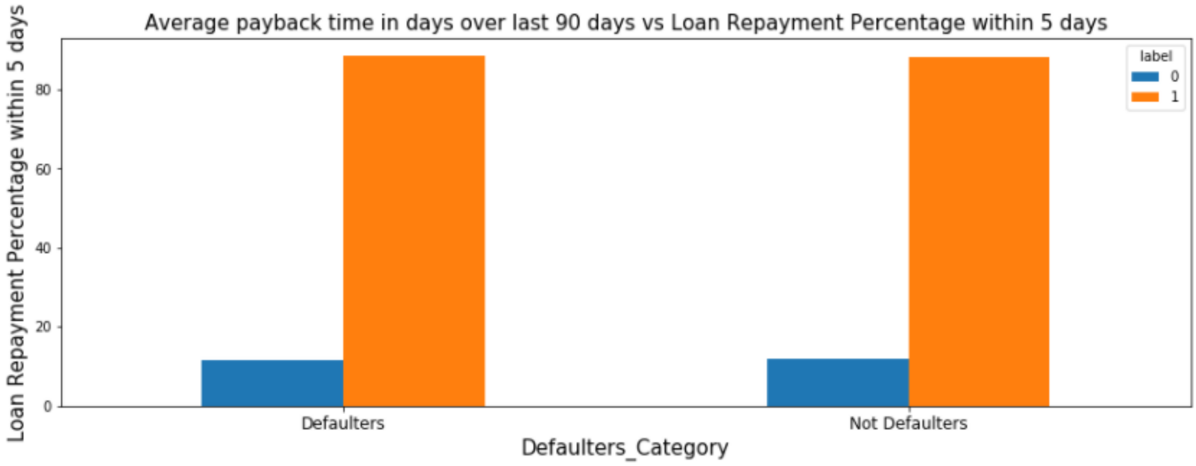
iii) Total amount of recharge in main account over last 90 days (in Indian Rupee) vs Loan Repayment Percentage within 5 days



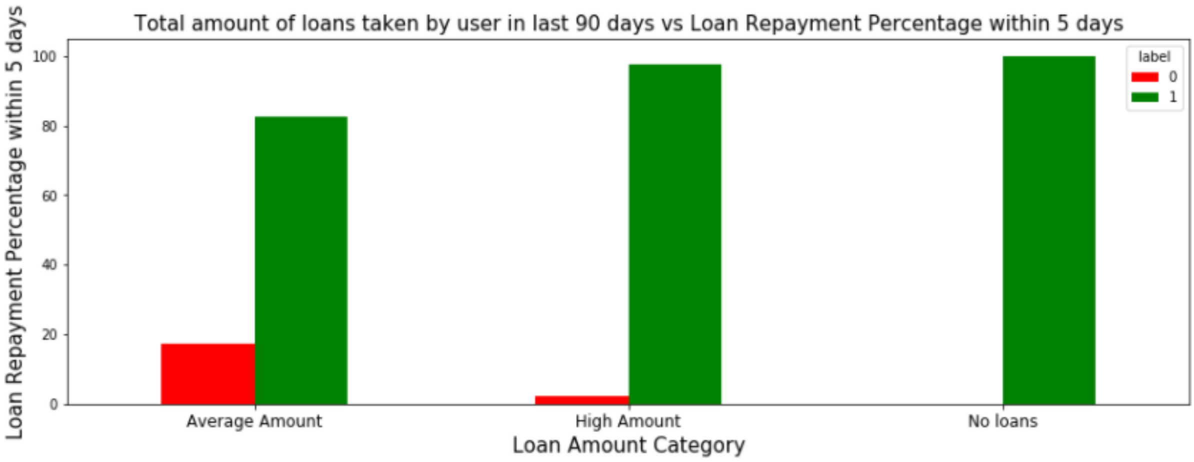
From the above graph it is clear that:

1) 40 % of the Users who do not even recharged in the 90 days are defaulters only.
2) Users who do very high amount of recharge always pays their loans on time. i.e. 98% of them are non-defaulters.
3) 34% of the Users who do less amount of recharge are defaulters.

iv) Average payback time in days over last 90 days vs Loan Repayment Percentage within 5 days



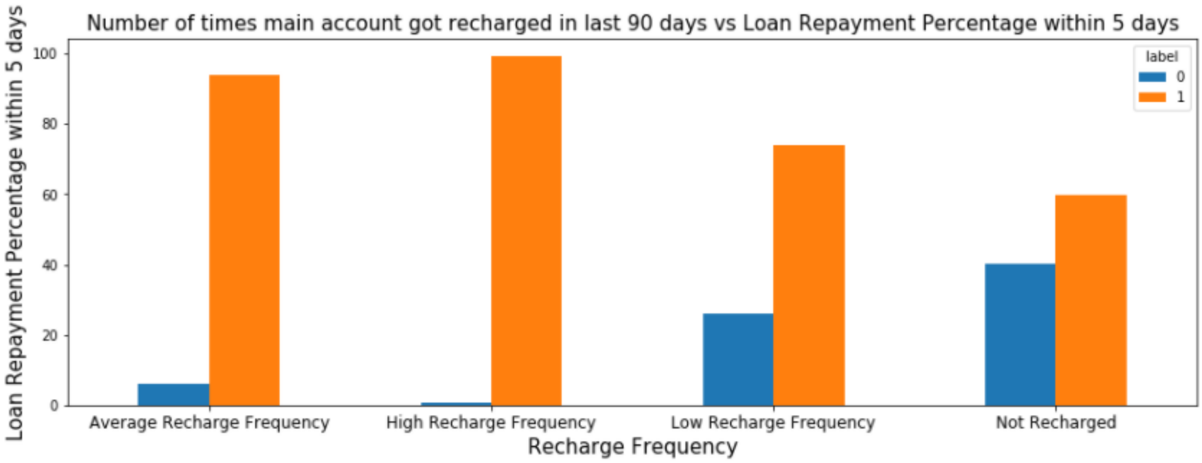
V) Total amount of loans taken by user in last 90 days vs Loan Repayment Percentage within 5 days



From the above graph it is clear that:

1) Users who did not take any loans are non-defaulters.
2) Most of the Users (i.e. 97%) who take large amount of loans comes under non defaulter category.
3) 17% of the users who take small loans are defaulters.

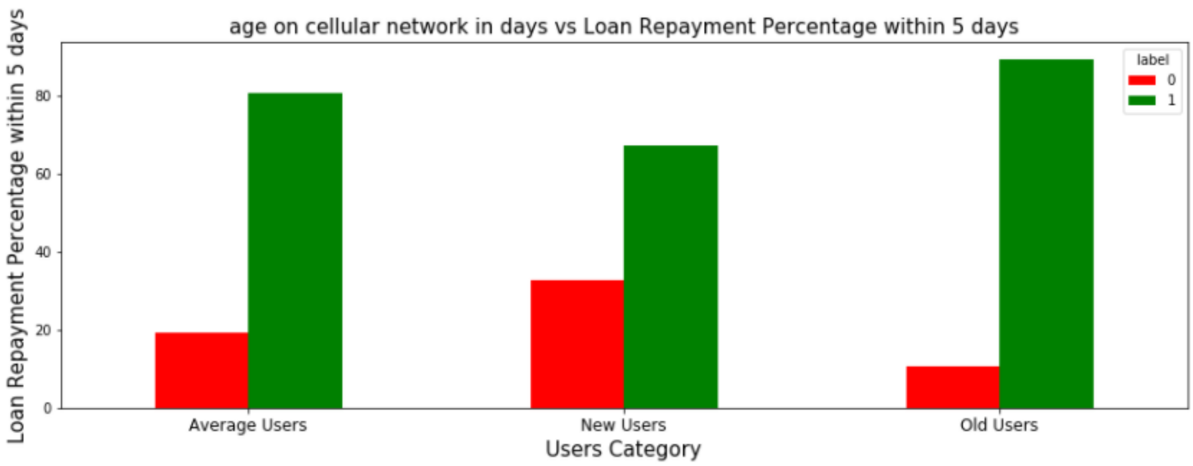
Vi) Number of times main account got recharged in last 90 days vs Loan Repayment Percentage within 5 days



From the above graph it is clear that:

1) Among the Users who have not done a single recharge in 3 months 40% are defaulters.
2) Among the Users who are very frequent in recharging and who always pay their loans on time are more in number i.e. 99% of the total category, which is a good news for the company.

Vii) Age on cellular network in days vs Loan Repayment Percentage within 5 days



From the above graph it is clear that:

1) 32% of the users who are defaulters are the new users.
2) Old Users are trusted and they are mostly non defaulters.

State the set of assumptions (if any) related to the problem under consideration

- From the above statistical summary of the above part of the dataset, **the important thing is that** Some features even have negative values like the age on cellular network, main account last recharge date, data account last recharge date. Negative values in these features make no sense thus these values should be removed.

- The Dataset we are having, consists of some features giving information about the user for the time span of 30 days and 90 days. According to me if we have data of large number of days for a particular user then we could interpret User's behaviour more precisely because many users have the tendency of repeating the same things. Thus the features having the data with a time span of 90 days gives more information about the user as compared to the features with a time span of 30 days.

- All the categories that is being made to make the visualizations easy are solemnly based on the Description i.e. statistical summary of the data plotted above ***for instance*** low comes under (0-25%), average comes under (25-75%) and high comes over 75% of the data values in a given feature. Using MS EXCEL I have found the maximum values a feature can have, beyond these values the values are unimaginable.

- *****(for an example beyond the value [2500], the very next value in "aon" feature comes out to be around 2379 years, which means a user is using the telephone services from 359 BCE which is clearly not possible).*****

- I checked the correlation of the independent and dependent features and from the correlation table it is also clear that the features with time span of 30 and 90 days almost have the same correlation thus we can drop one for the same information.

Hardware and Software Requirements and Tools Used

- Hardware: 8GB RAM, 64-bit, 9th gen i7 processor.
- Software: MS-Excel, Jupyter Notebook, python 3.6.

Libraries used:-

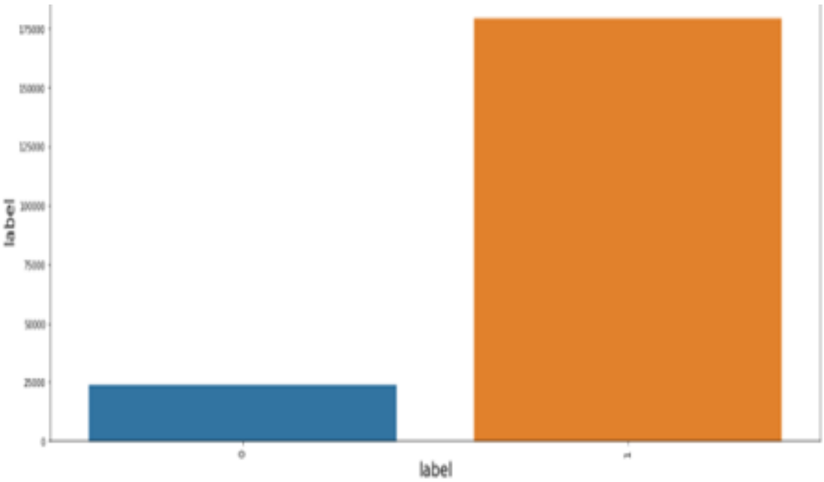
```python
# Importing libraries for data loading and visualization..
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from nltk import flatten

import warnings
warnings.filterwarnings('ignore')
```

```python
# .....................Importing Important libraries for Classification Models................
# Models from Scikit-Learn...
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import GaussianNB
from xgboost import XGBClassifier

# Ensemble Techniques...
# from sklearn.ensemble import GradientBoostingClassifierx apviorn
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier,ExtraTreesClassifier

# Model selection libraries...
from sklearn.model_selection import cross_val_score, cross_val_predict, train_test_split
from sklearn.model_selection import GridSearchCV

# Importing some metrics we can use to evaluate our model performance....
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.metrics import roc_auc_score, roc_curve, auc
from sklearn.metrics import precision_score, recall_score, f1_score
```

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods).

```
1    183431
0     26162
Name: label, dtype: int64
```



- From the above graph it is clear that the data set is highly imbalanced dataset, so applied SMOTETomek to balance the dataset.

Testing of Identified Approaches (Algorithms)

- lr=LogisticRegression()
- DT=DecisionTreeClassifier()
- GBC=GradientBoostingClassifier()
- RF=RandomForestClassifier()

- AD=AdaBoostClassifier()
- ETC=ExtraTreesClassifier()

Run and Evaluate selected models

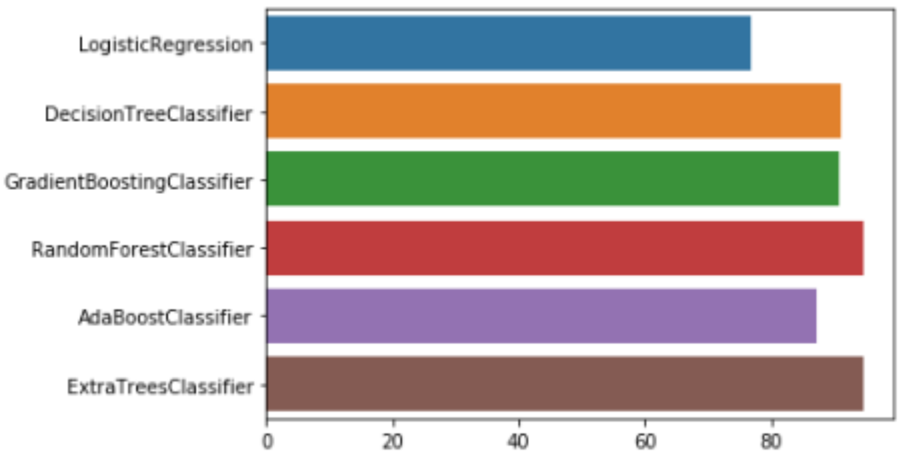
```python
#   Putting Scikit-Learn machine learning Models in a list so that it can be used for
#   further evaluation in loop.
models=[]
models.append(('LogisticRegression',lr))
models.append(('DecisionTreeClassifier',DT))
models.append(('GradientBoostingClassifier',GBC))
models.append(('RandomForestClassifier',RF))
models.append(('AdaBoostClassifier',AD))
models.append(("ExtraTreesClassifier",ETC))
```

```python
#     Lists to store model name, Learning score, Accuracy score, cross_val_score, Auc Roc score .
Model=[]
Score=[]
Acc_score=[]
cvs=[]
rocscore=[]
#            For Loop to Calculate Accuracy Score, Cross Val Score, Classification Report, Confusion Matrix

for name,model in models:
    print('***************************',name,'*****************************')
    print('\n')
    Model.append(name)
    print(model)
    print('\n')

     #        Now here I am calling a function which will calculate the max accuracy score for each model
     #                               and return best random state.
    r_state=max_acc_score(model,x,y)
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=r_state,stratify=y)
    model.fit(x_train,y_train)
#.............Learning Score...........
    score=model.score(x_train,y_train)
    print('Learning Score : ',score)
    Score.append(score*100)
    y_pred=model.predict(x_test)
    acc_score=accuracy_score(y_test,y_pred)
    print('Accuracy Score : ',acc_score)
    Acc_score.append(acc_score*100)
#...............Finding Cross_val_score..................
    cv_score=cross_val_score(model,x,y,cv=10,scoring='accuracy').mean()
    print('Cross Val Score : ', cv_score)
    cvs.append(cv_score*100)

#...............Roc auc score..........................
    false_positive_rate,true_positive_rate, thresholds=roc_curve(y_test,y_pred)
    roc_auc=auc(false_positive_rate, true_positive_rate)
    print('roc auc score : ', roc_auc)
    rocscore.append(roc_auc*100)
    print('\n')
    print('Classification Report:\n',classification_report(y_test,y_pred))
    print('\n')
    print('Confusion Matrix:\n',confusion_matrix(y_test,y_pred))
    print('\n')
    plt.figure(figsize=(10,40))
    plt.subplot(911)
    plt.title(name)
    plt.plot(false_positive_rate,true_positive_rate,label='AUC = %0.2f'% roc_auc)
    plt.plot([0,1],[0,1],'r--')
    plt.legend(loc='lower right')
    plt.ylabel('True_positive_rate')
    plt.xlabel('False_positive_rate')
    print('\n\n')
```

Key Metrics for success in solving problem under consideration

	Model	Learning Score	Accuracy Score	Cross Val Score	Roc_Auc_curve
0	LogisticRegression	76.3163	76.9244	76.4298	76.9244
1	DecisionTreeClassifier	99.976	90.955	90.5398	90.955
2	GradientBoostingClassifier	90.5731	90.8215	90.281	90.8215
3	RandomForestClassifier	99.9764	94.7211	94.2806	94.7211
4	AdaBoostClassifier	87.2164	87.3094	86.7443	87.3094
5	ExtraTreesClassifier	99.9771	94.5904	94.4702	94.5904

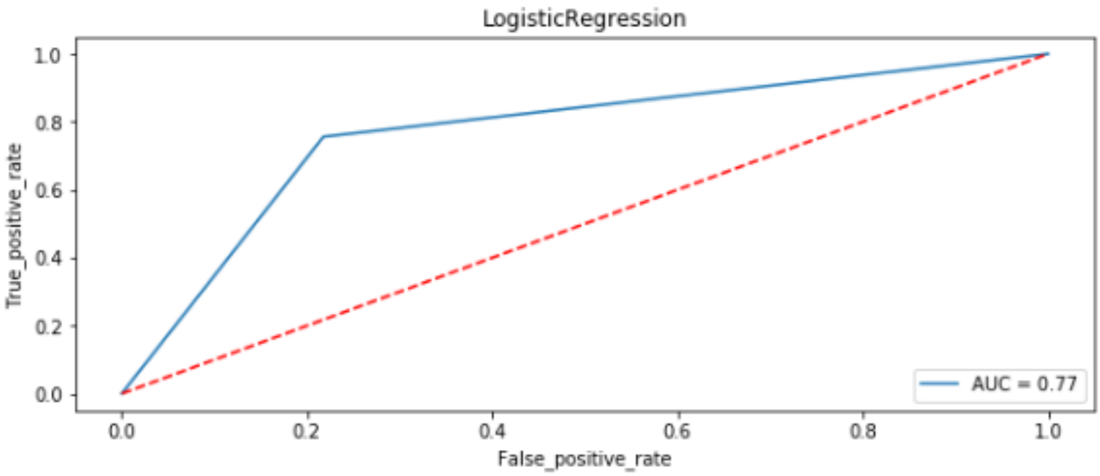
Key Metrices used were the Accuracy Score, Crossvalidation Score and AUC & ROC Curve as this was binary classification problem and we focus more on AUC & ROC curve metrices to observe True Positive Rate and False Positive Rare, for users who paid the loan and falsely marked as default and will their affect the credit score and we already talked about the importance of that in financial sector, and for the users who are marked falsely marked as paid but they didn't, can affect the company revenue.



Visualizations:

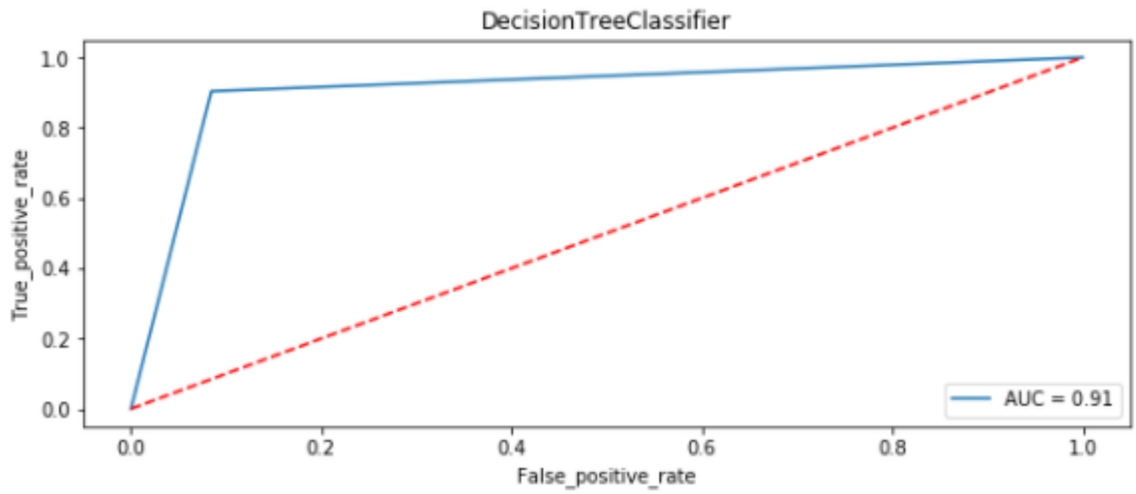
Logistic regression:

```
************************** LogisticRegression ****************************


LogisticRegression()


Max Accuracy Score corresponding to Random State  50 is: 0.7692441708528123


Learning Score :  0.7631628616021197
Accuracy Score :  0.7692441708528123
Cross Val Score :  0.7642978379339211
roc auc score :  0.7692441708528122


Classification Report:
               precision    recall  f1-score   support

           0       0.76      0.78      0.77     34439
           1       0.78      0.76      0.77     34439

    accuracy                           0.77     68878
   macro avg       0.77      0.77      0.77     68878
weighted avg       0.77      0.77      0.77     68878



Confusion Matrix:
 [[26944  7495]
 [ 8399 26040]]
```



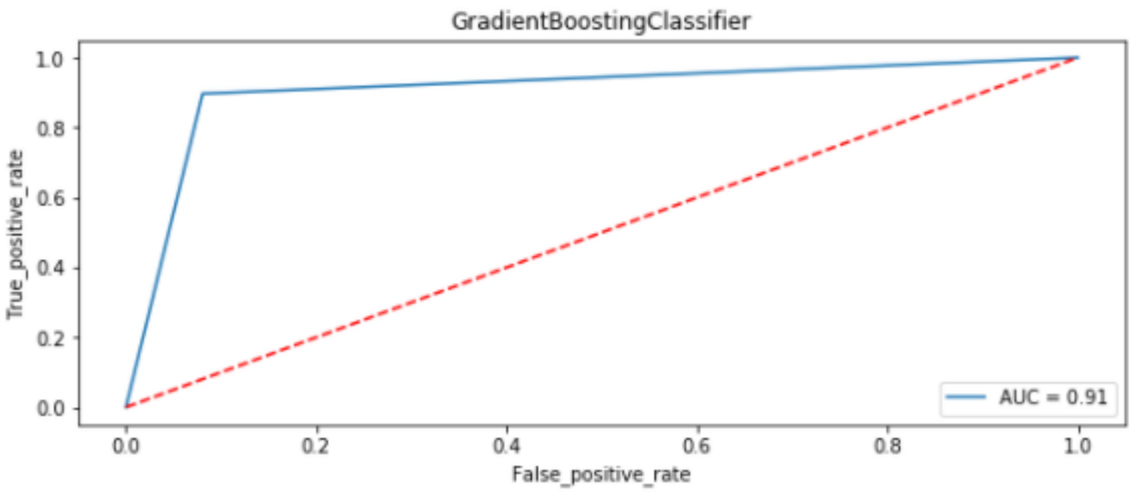
- Decision Tree Classifier:

```
*************************** DecisionTreeClassifier *****************************


DecisionTreeClassifier()


Max Accuracy Score corresponding to Random State  65 is: 0.9096663666192398


Learning Score :  0.9997604442669957
Accuracy Score :  0.9096808850431197
Cross Val Score :  0.9052988346689561
roc auc score :  0.9096808850431197


Classification Report:
               precision    recall  f1-score   support

           0       0.90      0.92      0.91     34439
           1       0.91      0.90      0.91     34439

    accuracy                           0.91     68878
   macro avg       0.91      0.91      0.91     68878
weighted avg       0.91      0.91      0.91     68878



Confusion Matrix:
 [[31530  2909]
 [ 3312 31127]]
```



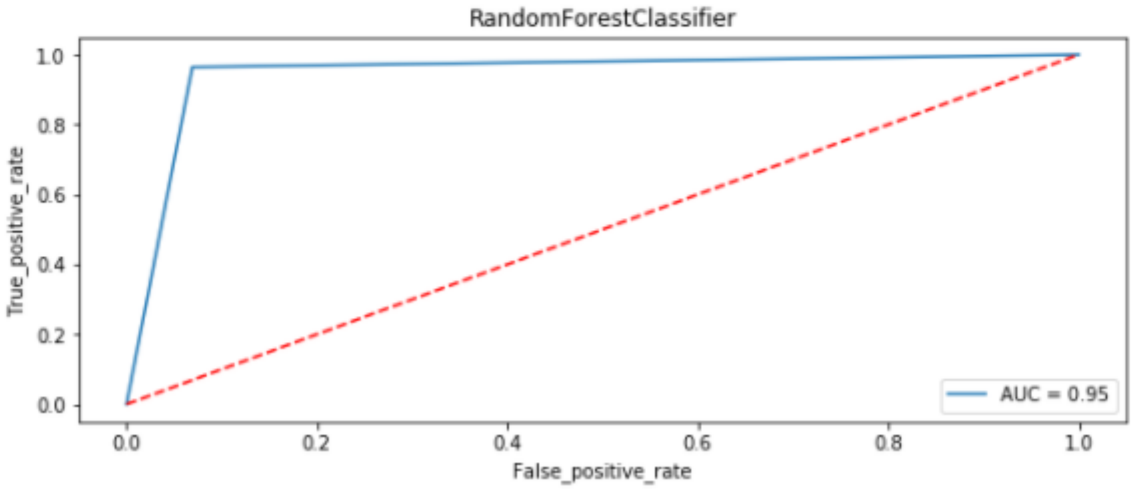
Gradient Boosting Classifier:

```
************************** GradientBoostingClassifier ****************************

GradientBoostingClassifier()

Max Accuracy Score corresponding to Random State  83 is: 0.9082145242312495

Learning Score :  0.9057311894305107
Accuracy Score :  0.9082145242312495
Cross Val Score :  0.9028103600369967
roc auc score :  0.9082145242312495

Classification Report:
               precision    recall  f1-score   support

           0       0.90      0.92      0.91     34439
           1       0.92      0.90      0.91     34439

    accuracy                           0.91     68878
   macro avg       0.91      0.91      0.91     68878
weighted avg       0.91      0.91      0.91     68878


Confusion Matrix:
 [[31663  2776]
 [ 3546 30893]]
```



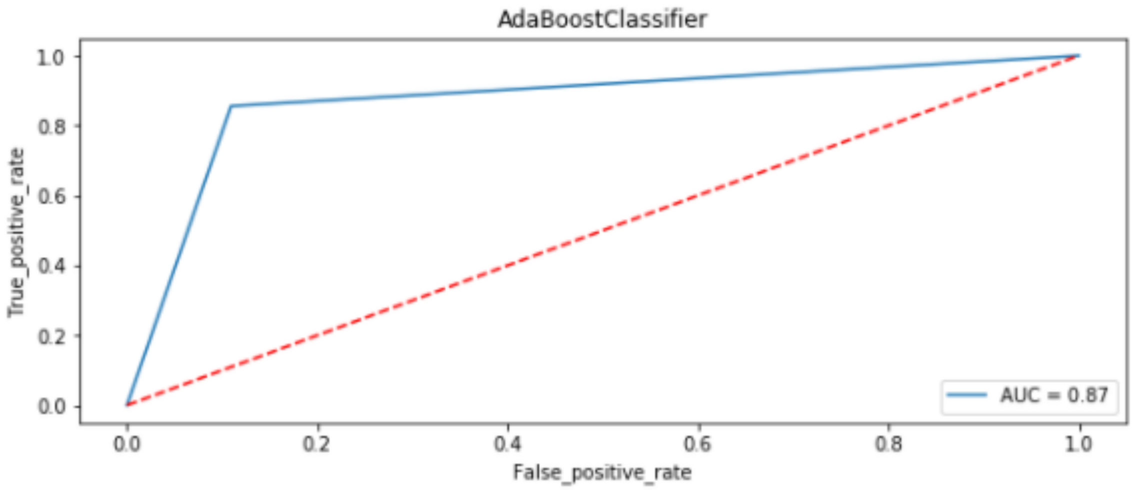
Random Forest Classifier:

```
************************** RandomForestClassifier ****************************

RandomForestClassifier()

Max Accuracy Score corresponding to Random State  94 is: 0.9475594529457882

Learning Score :  0.999764073899314
Accuracy Score :  0.9476610819129475
Cross Val Score :  0.9426027217190684
roc auc score :  0.9476610819129475

Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.93      0.95     34439
           1       0.93      0.96      0.95     34439

    accuracy                           0.95     68878
   macro avg       0.95      0.95      0.95     68878
weighted avg       0.95      0.95      0.95     68878


Confusion Matrix:
 [[32058  2381]
 [ 1224 33215]]
```



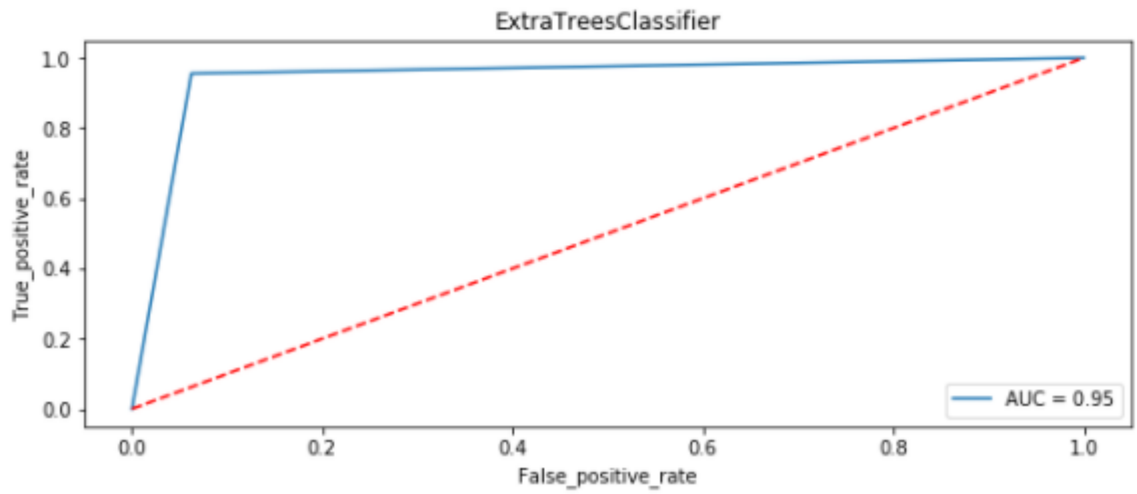
- Ada Boost Classifier:

```
*************************** AdaBoostClassifier *****************************

AdaBoostClassifier()

Max Accuracy Score corresponding to Random State  54 is: 0.8730944568657627

Learning Score :  0.8721643497513701
Accuracy Score :  0.8730944568657627
Cross Val Score :  0.867443223902843
roc auc score :  0.8730944568657627

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.89      0.88     34439
           1       0.89      0.86      0.87     34439

    accuracy                           0.87     68878
   macro avg       0.87      0.87      0.87     68878
weighted avg       0.87      0.87      0.87     68878


Confusion Matrix:
 [[30670  3769]
 [ 4972 29467]]
```



Extra Tree Classifier:

```
************************** ExtraTreesClassifier ****************************


ExtraTreesClassifier()


Max Accuracy Score corresponding to Random State  54 is: 0.9470513081099916


Learning Score :  0.999778592428587
Accuracy Score :  0.946543163274195
Cross Val Score :  0.9450737196054091
roc auc score :  0.946543163274195


Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.94      0.95     34439
           1       0.94      0.96      0.95     34439

    accuracy                           0.95     68878
   macro avg       0.95      0.95      0.95     68878
weighted avg       0.95      0.95      0.95     68878



Confusion Matrix:
 [[32291  2148]
 [ 1534 32905]]
```



After all this process conclusion is that Random Forest Classifier and Extra Tree Classifier are performing well in terms of Accuracy score, Cross val score and Roc_Auc score as compared to other models.

Hyper Parameter Tuning Results:

```
#checking accuracy score using best parameters which calculated from gridsearchCV
rf=RandomForestClassifier(n_estimators=200,max_depth=None, min_samples_leaf= 1, max_features= 'aut
max_acc_score(rf,x,y)
```

```
Max Accuracy Score corresponding to Random State  94 is: 0.9473561950114695
```

```
#checking accuracy score using best parameters which calculated from gridsearchCV
etc=ExtraTreesClassifier(n_estimators=200,max_depth=None, min_samples_leaf= 1, max_features= 2,min
max_acc_score(etc,x,y)
```

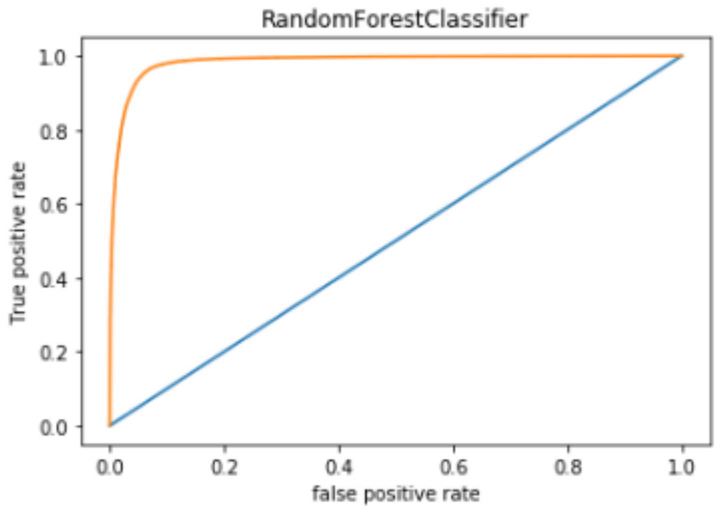
```
Max Accuracy Score corresponding to Random State  83 is: 0.9431458520862975
```

After all this process conclusion of Hyper Parameter is that Random Forest Classifier is giving accuracy of 94.73%. So now I am making a final model using Random Forest Classifier.

Final Model:

```
# Using RandomForestClassifier for final model...
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=94,test_size=.20,stratify=y)
rfc=RandomForestClassifier(n_estimators=200,max_depth=None, min_samples_leaf= 1,
                           max_features= 'auto',min_samples_split=4)
rfc.fit(x_train,y_train)
rfc.score(x_train,y_train)
rfcpred=rfc.predict(x_test)
print('Accuracy Score:',accuracy_score(y_test,rfcpred))
print('Confusion Matrix:',confusion_matrix(y_test,rfcpred))
print('Classification Report:','\n',classification_report(y_test,rfcpred))
```

```
Accuracy Score: 0.9475594529457882
Confusion Matrix: [[32019  2420]
 [ 1192 33247]]
Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.93      0.95     34439
           1       0.93      0.97      0.95     34439

    accuracy                           0.95     68878
   macro avg       0.95      0.95      0.95     68878
weighted avg       0.95      0.95      0.95     68878
```




```
roc_auc_score =  0.9847887122799309
```

From the above visualization and matrices found that the RandomForest Classifier performed the best 98.47% AOC_ROC_SCORE, with precision accuracy score of 96% and recall 97%.

Interpretation of the Results

- From the above visualization and matrices found that the Random Forest Classifier performed the best AUC_ROC_SCORE **i.e. 98.47%.**

CONCLUSION

Key Findings and Conclusions of the Study

1) 28% of Users having negative or zero balance are defaulters, which is very high.
2) 10% to 12% Users are defaulters which falls in the category of Average and Low balance category.
3) Users having high balance and are defaulters are very less in number.
4) Users who take more number of loans are non-defaulters (i.e 98% of the category) as they repays the loan within the given time i.e. 5 days.
5) 14% of the Users are among the average number of loan taken category are defaulters.
6) 40 % of the Users who do not even recharged in the 90 days are defaulters only.
7) Users who do very high amount of recharge always pays their loans on time. i.e 98% of them are non-defaulters.
8) 34% of the Users who do less amount of recharge are defaulters.
9) Users who did not take any loans are non-defaulters.
10) Most of the Users (i.e. 97%) who take large amount of loans comes under non defaulter category.
11) 17% of the users who take small loans are defaulters.
12) Among the Users who have not done a single recharge in 3 months 40% are defaulters.
13) Among the Users who are very frequent in recharging and who always pay their loans on time are more in number i.e. 99% of the total category, which is a good news for the company.q
14) 32% of the users who are defaulters are the new users.
15) Old Users are trusted and they are mostly non defaulters.
16) Random Forest Classifier performed the best AUC_ROC_SCORE **i.e. 94.7%.**

Learning Outcomes of the Study in respect of Data Science

- Visualizations and Data Cleaning part was very crucial as without the cleaning we were not able to judge the data effectively and won’t be able to remove the outliers thus adding in to the errors.
- Visualizations helped a lot in finding out those outliers values and helped in finding out the features having direct relation between the feature and the label.
- We could have experimented by using PCA for dimension reducing and could have tried opting some other technique instead of SMOTETomek.

Limitations of this work and Scope for Future Work

- Machine Learning Algorithms like KNN took enormous amount of time to build the model.
- I could have experimented by using PCA and could have tried opting some other technique instead of SMOTETomek.
- Some altering notification before the deadlines can play a major role, in reducing the defaulter rate, whether it is sms notification or intimation using a call.

- I suggest to combine both the account i.e. data and main account and provide both services on each recharge to increase the profit.