FLIP ROBO

PROJECT REPORT ON:

"Micro-Credit Defaulter Model"

SUBMITTED BY

Abhay Surma

ACKNOWLEDGMENT

I would like to express my special gratitude to "Flip Robo" team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzation skills. And I want to express my huge gratitude to Ms. Khushboo Garg (SME Flip Robo), she is the person who has helped me to get out of all the difficulties I faced while doing the project. And also Mr. Sajid Choudhary (SME Flip Robo) who has inspired me in so many aspects and also encouraged me a lot with his valuable words and with his unconditional support I have ended up with a beautiful Project.

A huge thanks to my academic team "Data trained" who are the reason behind what I am today. Last but not least my parents who have been my backbone in every step of my life. And also thank you for many other persons who has helped me directly or indirectly to complete the project.

Contents:

1. Introduction

1.1 Business Problem Framing:
1.2 Conceptual Background of the Domain Problem
1.3 Review of Literature
1.4 Motivation for the Problem Undertaken

2. Analytical Problem Framing

2.1 Mathematical/ Analytical Modeling of the Problem
2.2 Data Sources and their formats
2.3 Data Preprocessing Done
2.4 Data Inputs-Logic-Output Relationships
2.5 Hardware and Software Requirements and Tools Used

3. Data Analysis and Visualization

3.1 Identification of possible problem-solving approaches (methods)
3.2 Testing of Identified Approaches (Algorithms)
3.3 Key Metrics for success in solving problem under consideration
3.4 Visualization
3.5 Run and Evaluate selected models
3.6 Interpretation of the Results

4. Conclusion

4.1 Key Findings and Conclusions of the Study
4.2 Learning Outcomes of the Study in respect of Data Science
4.3 Limitations of this work and Scope for Future Work

1.INTRODUCTION

1.1 Business Problem Framing:

A Microfinance Institution (MFI) is an organization that offers financial services to low income populations. MFS becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income. The Microfinance services (MFS) provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans and so on.

Many microfinance institutions (MFI), experts and donors are supporting the idea of using mobile financial services (MFS) which they feel are more convenient and efficient, and cost saving, than the traditional high-touch model used since long for the purpose of delivering microfinance services. Though, the MFI industry is primarily focusing on low income families and are very useful in such areas, the implementation of MFS has been uneven with both significant challenges and successes.

Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients.

We are working with one such client that is in Telecom Industry. They are a fixed wireless telecommunications network provider. They have launched various products and have developed its business and organization based on the budget operator model, offering better products at Lower Prices to all value conscious customers through a strategy of disruptive innovation that focuses on the subscriber.

1.2 Conceptual Background of the Domain Problem

Telecom Industries understand the importance of communication and how it affects a person's life, thus, focusing on providing their services and products to low income families and poor customers that can help them in the need of hour.

They are collaborating with an 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).

The sample data is provided to us from our client database. It is hereby given to you for this exercise. In order to improve the selection of customers for the credit, the client wants some predictions that could help them in further investment and improvement in selection of customers.

We have to build a model which can be used 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 i.e. defaulter.

1.3 Review of Literature

An attempt has been made in this report to review the available literature in the area of microfinance. Approaches to microfinance, issues related to measuring social impact versus profitability of MFIs, issue of sustainability, variables impacting sustainability, affect of regulations of profitability and impact assessment of MFIs have been summarized in the below report. We hope that the below report of literature will provide a platform for further research and help the industry to combine theory and practice to take microfinance forward and contribute to alleviating the poor from poverty.

1.4 Motivation for the Problem Undertaken

I have to model the micro credit defaulters with the available independent variables. This model will then be used by the management to understand how the customer is considered as defaulter or non-defaulter based on the independent variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand whether the customer will be paying back the loaned amount within 5 days of insurance of loan. The

relationship between predicting defaulter and the economy is an important motivating factor for predicting micro credit defaulter model.

2.Analytical Problem Framing

2.1 Mathematical/ Analytical Modeling of the Problem

In this perticular problem I had label as my target column and it was having two classes Label '1' indicates that the loan has been payed i.e. Non- defaulter, while, Label '0' indicates that the loan has not been payed i.e. defaulter. So clearly it is a binary classification problem and I have to use all classification algorithms while building the model. There was no null values in the dataset. Also, I observed some unnecessary entries in some of the columns like in some columns I found more than 90% zero values so I decided to drop those columns. If I keep those columns as it is, it will create high skewness in the model. To get better insight on the features I have used ploting like distribution plot, bar plot and count plot. With these ploting I was able to understand the relation between the features in better manner. Also, I found outliers and skewness in the dataset so I removed outliers using percentile method and I removed skewness using yeo-johnson method. I have used all the classification aldorithms while building model then tunned the best model and saved the best model. At last I have predicted the label using saved model.

2.2 Data Sources and their formats

The data was collected for my internship company – Flip Robo technologies in excel format. The sample data is provided to us from our client database. It is hereby given to us for this exercise. In order to improve the selection of customers for the credit, the client wants some predictions that could help them in further investment and improvement in selection of customers.

Also, my dataset was having 209593 rows and 36 columns including target. In t his perticular datasets I have object, float and integer types of data. The inform ation about features is as follows.

Features Information:

1. label : Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure}
2. msisdn : mobile number of user
3. aon : age on cellular network in days
4. daily_decr30 : Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)
5. daily_decr90 : Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)
6. rental30 : Average main account balance over last 30 days
7. rental90 : Average main account balance over last 90 days
8. last_rech_date_ma : Number of days till last recharge of main account
9. last_rech_date_da: Number of days till last recharge of data account
10. last_rech_amt_ma : Amount of last recharge of main account (in Indonesia n Rupiah)
11. cnt_ma_rech30 : Number of times main account got recharged in last 30 da ys
12. fr_ma_rech30 : Frequency of main account recharged in last 30 days
13. sumamnt_ma_rech30 : Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)
14. medianamnt_ma_rech30 : Median of amount of recharges done in main ac count over last 30 days at user level (in Indonesian Rupiah)
15. medianmarechprebal30 : Median of main account balance just before recha rge in last 30 days at user level (in Indonesian Rupiah)
16. cnt_ma_rech90 : Number of times main account got recharged in last 90 da ys
17. fr_ma_rech90 : Frequency of main account recharged in last 90 days
18. sumamnt_ma_rech90 : Total amount of recharge in main account over last 90 days (in Indonasian Rupiah)
19. medianamnt_ma_rech90 : Median of amount of recharges done in main ac count over last 90 days at user level (in Indonasian Rupiah)
20. medianmarechprebal90 : Median of main account balance just before recha rge in last 90 days at user level (in Indonasian Rupiah)
21. cnt_da_rech30 : Number of times data account got recharged in last 30 day s
22. fr_da_rech30: Frequency of data account recharged in last 30 days
23. cnt_da_rech90 : Number of times data account got recharged in last 90 day s
24. fr_da_rech90 : Frequency of data account recharged in last 90 days
25. cnt_loans30 : Number of loans taken by user in last 30 days

26. amnt_loans30: Total amount of loans taken by user in last 30 days
27. maxamnt_loans30 : maximum amount of loan taken by the user in last 30 d ays
28. medianamnt_loans30 : Median of amounts of loan taken by the user in last 30 days
29. cnt_loans90 : Number of loans taken by user in last 90 days
30. amnt_loans90 : Total amount of loans taken by user in last 90 days
31. maxamnt_loans90 : maximum amount of loan taken by the user in last 90 d ays
32. medianamnt_loans90 : Median of amounts of loan taken by the user in last 90 days
33. payback30 : Average payback time in days over last 30 days
34. payback90 : Average payback time in days over last 90 days
35. pcircle : telecom circle
36. pdate : date

2.3 Data Preprocessing Done

- ✓ As a first step I have imported required libraries and I have imported the dataset which was in csv format.
- ✓ Then I did all the statistical analysis like checking shape, nunique, value counts, info etc…..
- ✓ Then while looking into the value counts I found some columns with more than 90% zero values this creates skewness in the model and there are chances of getting model bias so I have dropped those columns with more than 90% zero values.
- ✓ While checking for null values I found no null values in the dataset.
- ✓ I have also droped Unnamed:0, msisdn and pcircle column as I found they are useless.
- ✓ Next as a part of feature extraction I converted the pdate column to pyear, pmonth and pday. Thinking that this data will help us more than pdate.
- ✓ In some columns I found negative values which were unrealistic so I have converted those negative values to positive using abs command.
- ✓ Also, I have converted all the flaot values in maxamnt_loans90 to zero as it is specified in the problem statement we can have only 0,6,12 as maximum amount of loan taken by the user in last 30 days. As well I

have droped all the data with amnt_loans90=0 as it gives the persons who have not taken any loans.

2.4 Data Inputs- Logic- Output Relationships

- ✓ Since I had all numerical columns I have plotted dist plot to see the distribution of each column data.
- ✓ I have used box plot for each pair of categorical features that shows the relation between label and independent features. Also we can observe wheather the person pays back the loan within the date based on features.
- ✓ In maximum features relation with target I observed Non-defaulter count is high compared to defaulters.

2.5 Hardware and Software Requirements and Tools Used

While taking up the project we should be familiar with the Hardware and software required for the successful completion of the project. Here we need the following hardware and software.

Hardware required: -

1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software/s required: -

1.Anaconda

Libraries required :-

- ✓ To run the program and to build the model we need some basic libraries as follows:

```
In [1]: #importing required libraries
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import datetime as dt

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

- ✓ **import pandas as pd**: **pandas** is a popular Python-based data analysis toolkit which can be imported using import pandas as pd. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a numpy matrix array. This makes pandas a trusted ally in data science and machine learning.
- ✓ **import numpy as np**: NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.
- ✓ **import seaborn as sns:** Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.
- ✓ **Import matplotlib.pyplot as plt:** matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.
- ✓ from sklearn.preprocessing import OrdinalEncoder
- ✓ from sklearn.preprocessing import MinMaxScaler
- ✓ from sklearn.ensemble import RandomForestClassifier
- ✓ from sklearn.tree import DecisionTreeClassifier
- ✓ from xgboost import XGBClassifier
- ✓ from sklearn.ensemble import GradientBoostingClassifier
- ✓ from sklearn.ensemble import ExtraTreesClassifier

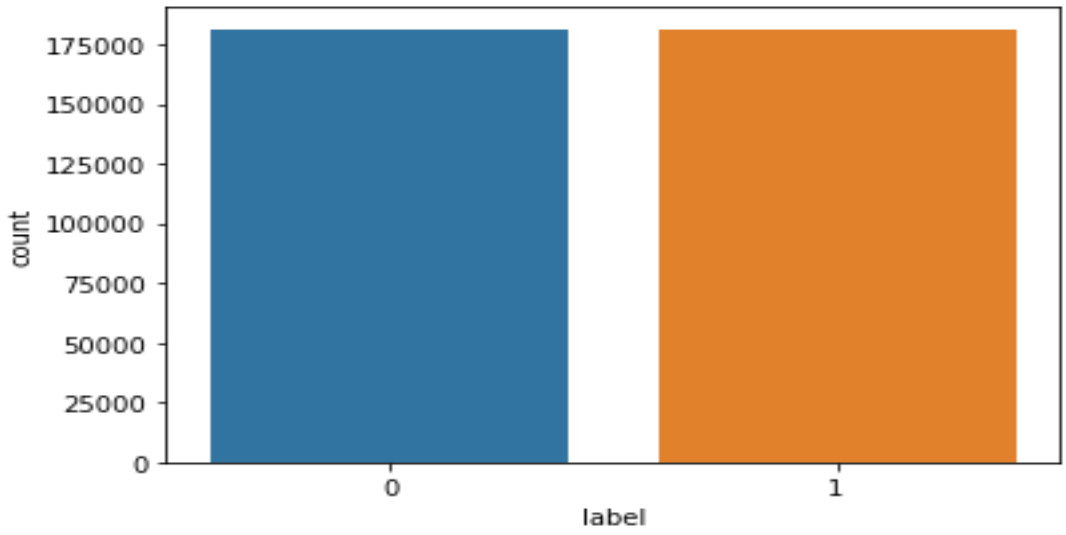
- ✓ from sklearn.metrics import classification_report
- ✓ from sklearn.metrics import accuracy_score
- ✓ from sklearn.model_selection import cross_val_score

With this sufficient libraries we can go ahead with our model building.

3.Data Analysis and Visualization

3.1 Identification of possible problem-solving approaches (methods)

- ✓ To remove outliers I have used percentile method. And to remove skewness I have used yeo-johnson method. We have dropped all the unnecessary columns in the dataset according to our understanding. Use of Pearson's correlation coefficient to check the correlation between dependent and independent features. Also I have used Normalization to scale the data. After scaling we have to balance the target column using oversampling. Then followed by model building with all Classification algorithms. I have used oversampling (SMOTE) to get rid of data imbalancing. The balanced output looks like this.



3.2 Testing of Identified Approaches (Algorithms)

Since label was my target and it was a classification column with 0-defaulter and 1-Non-defaulter, so this perticular problem was Classification problem. And I have used all Classification algorithms to build my model. By looking into the difference of accuracy score and cross validation score I found RandomForestClassifier as a best model with least difference. Also to get the best model we have to run through multiple models and to avoid the confusion of overfitting we have go through cross validation. Below are the list of classification algorithms I have used in my project.

- XGBClassifier
- DecisionTreeClassifier
- BaggingClassifier
- AdaBoostClassifier

3.3 Key Metrics for success in solving problem under consideration

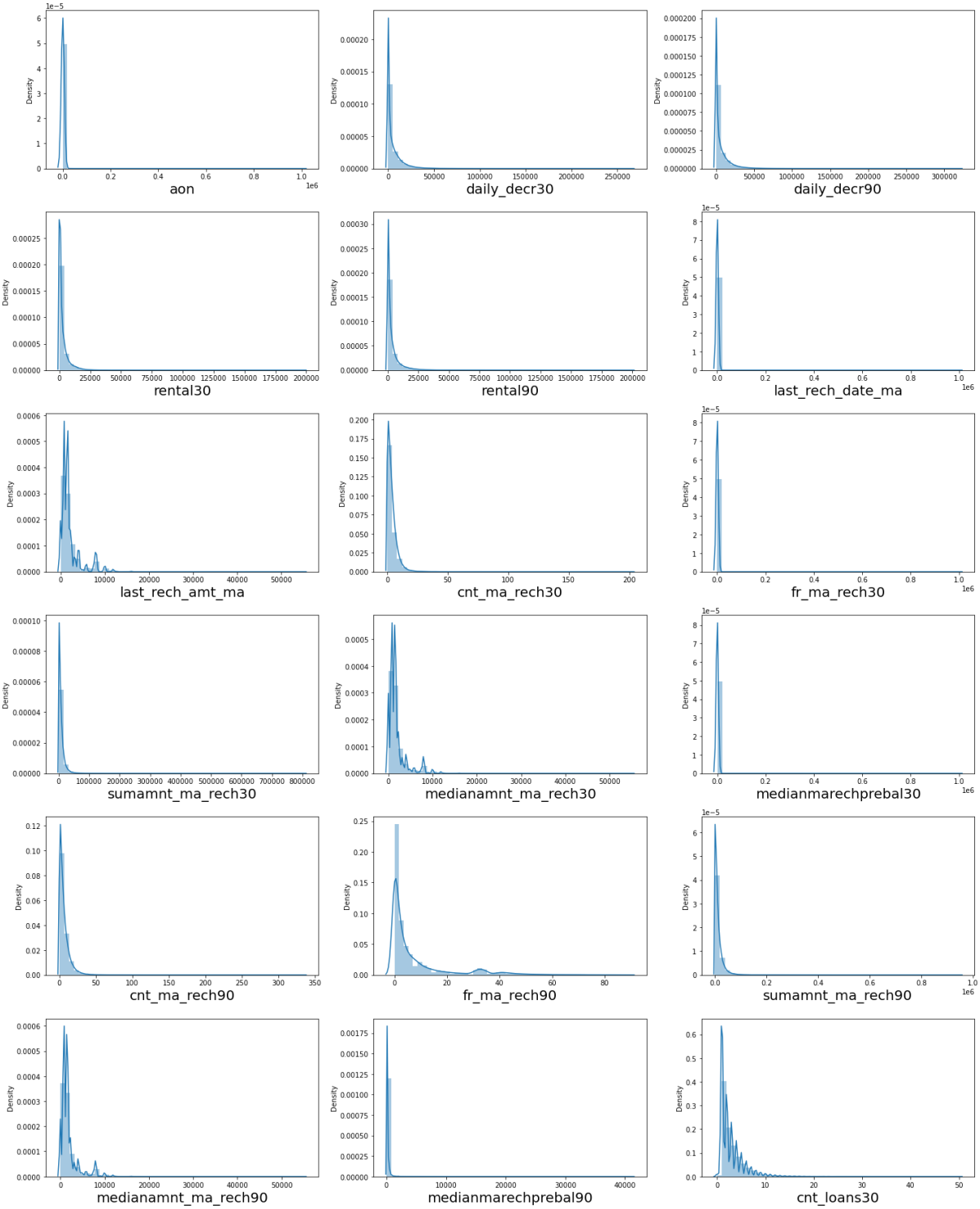
I have used the following metrics for evaluation:

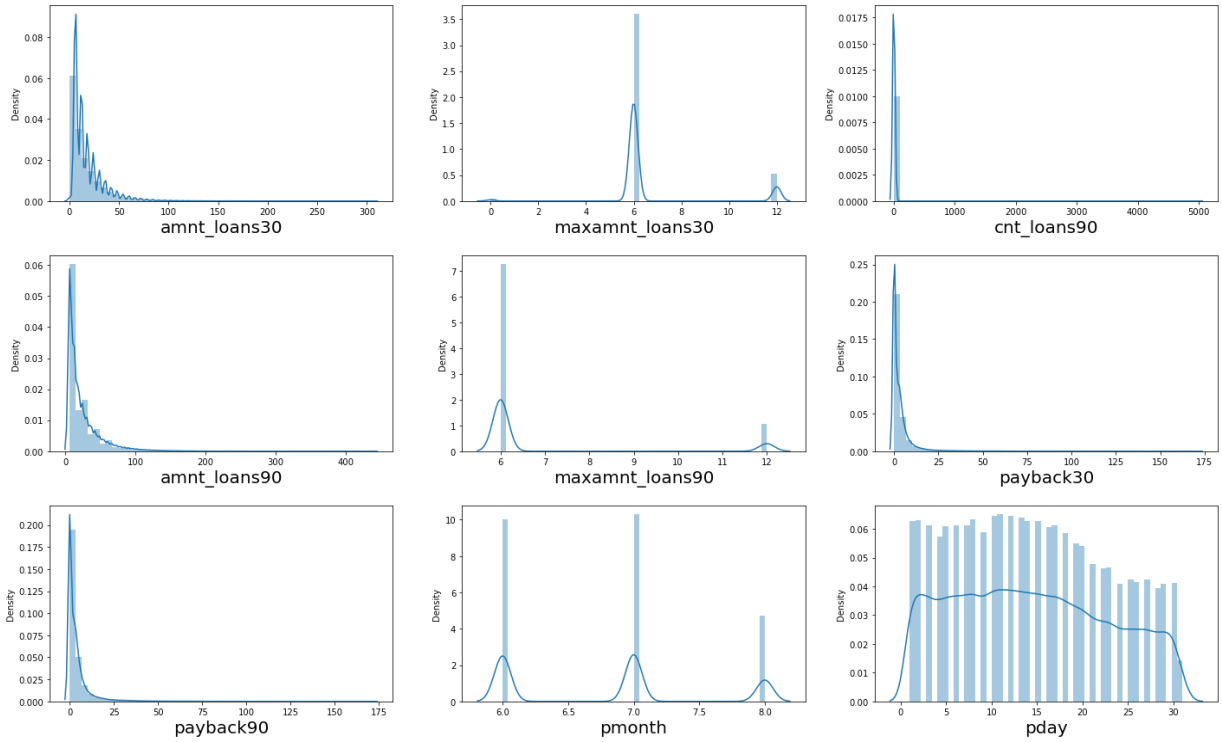
- **Precision** can be seen as a measure of quality, higher precision means that an algorithm returns more relevant results than irrelevant ones.
- **Recall** is used as a measure of quantity and high recall means that an algorithm returns most of the relevant results.
- **Accuracy score** is used when the True Positives and True negatives are more important. Accuracy can be used when the class distribution is similar.
- **F1-score** is used when the False Negatives and False Positives are crucial. While F1-score is a better metric when there are imbalanced classes.
- **Cross_val_score**: To run cross-validation on multiple metrics and also to return train scores, fit times and score times. Get predictions from each split of cross-validation for diagnostic purposes. Make a scorer from a performance metric or loss function.
- **AUC_ROC _score**: ROC curve. It is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0
- I have used accuracy_score since I have balanced my data using oversampling.

3.4 Visualizations

I have used bar plots to see the relation of numerical feature with target and I have used 2 types of plots for numerical columns one is disp plot for univariate and bar plot for bivariate analysis.

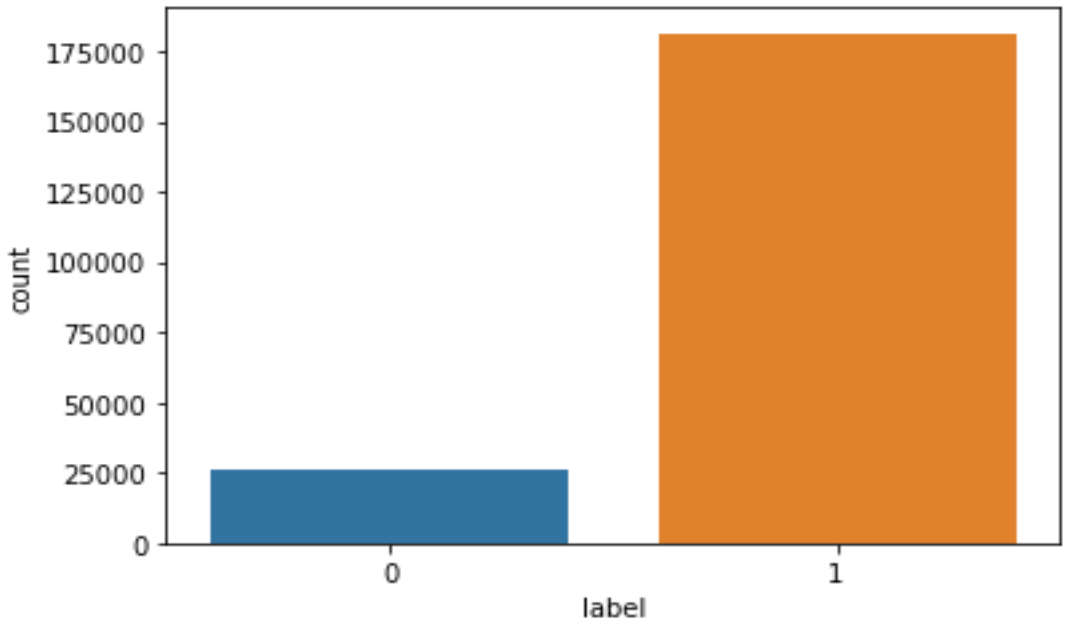
1. Univariate Analysis for numerical columns:





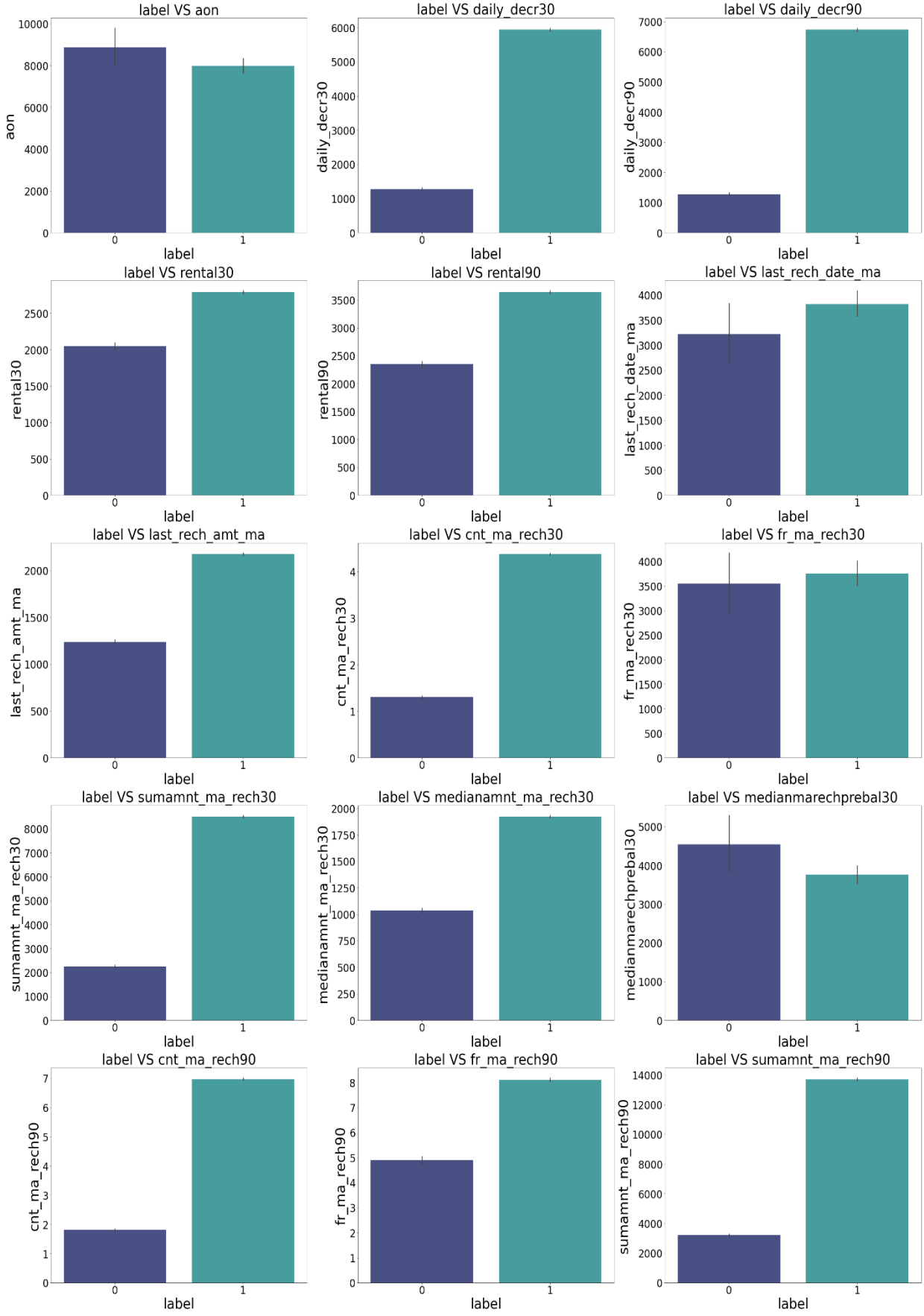
Observations:

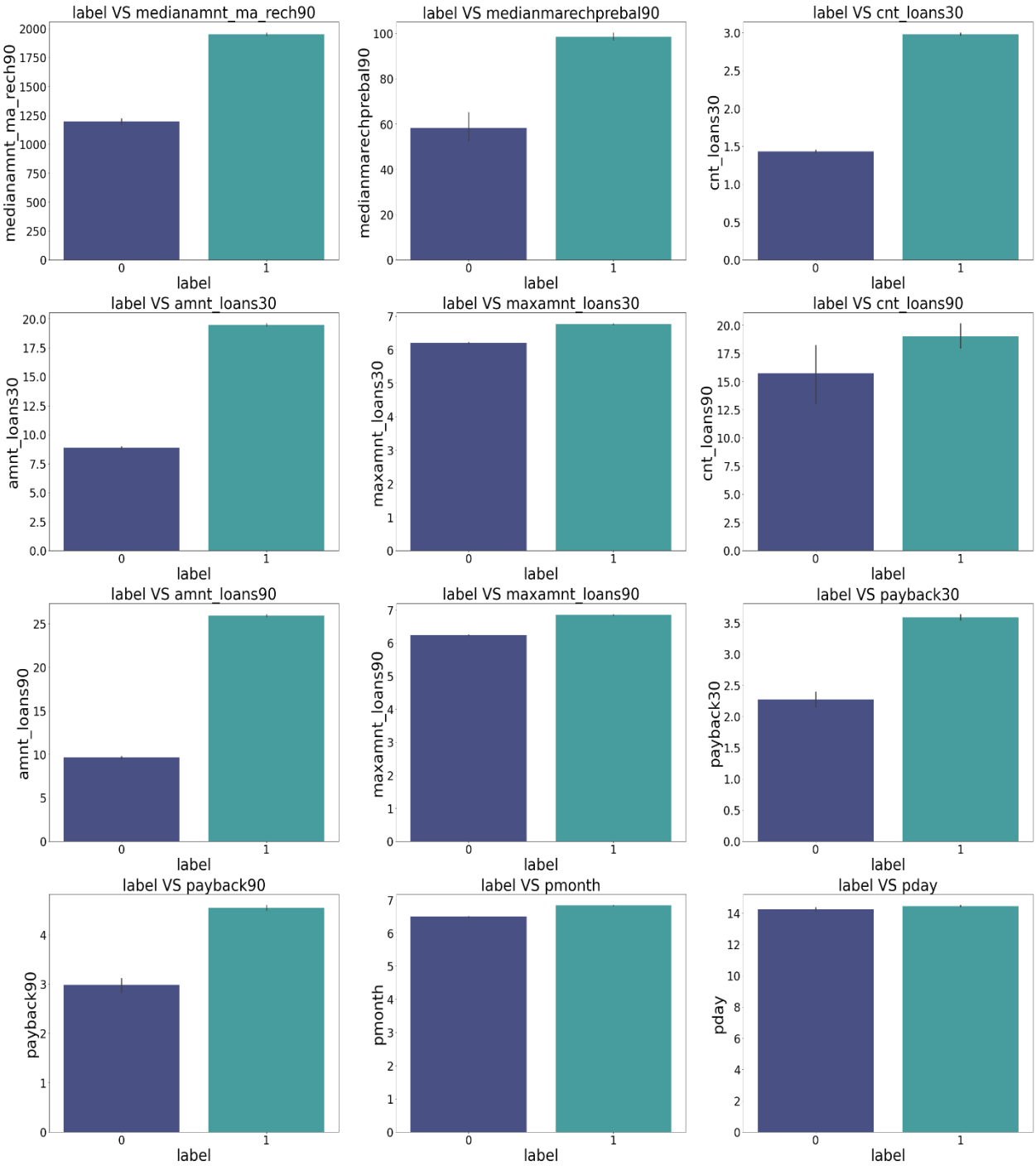
- ✓ We can clearly see that there is skewness in most of the columns so we have to treat them using suitable methods.



- ✓ There is a data imbalancing issue so we have to treat this by using oversampling or under sampling.

2. Bivariate analysis for numerical columns:





Observations:

- ✓ 1.Customers with high value of Age on cellular network in days(aon) are maximum defaulters(who have not paid there loan amount-0).
- ✓ 2.Customers with high value of Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)(daily_decr30) are maximum Non-defaulters(who have paid there loan amount-1).

- ✓ 3.Customers with high value of Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)(daily_decr90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 4.Customers with high value of Average main account balance over last 30 days(rental30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 5.Customers with high value of Average main account balance over last 90 days(rental90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 6.Customers with high Number of days till last recharge of main account(last_rech_date_ma) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 7.Customers with high value of Amount of last recharge of main account (in Indonesian Rupiah)(last_rech_amt_ma) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 8.Customers with high value of Number of times main account got recharged in last 30 days(cnt_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 9.Customers with high value of Frequency of main account recharged in last 30 days(fr_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1) and also the count is high for defaulters comparitively Non-defaulters are more in number.
- ✓ 10.Customers with high value of Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)(sumamnt_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 11.Customers with high value of Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)(medianamnt_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 12.Customers with high value of Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)(medianmarechprebal30) are maximum defaulters(who have not paid there loan amount-0).
- ✓ 13.Customers with high value of Number of times main account got recharged in last 90 days(cnt_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 14.Customers with high value of Frequency of main account recharged in last 90 days(fr_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).

- ✓ 15.Customers with high value of Total amount of recharge in main account over last 90 days (in Indonasian Rupiah)(sumamnt_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 16.Customers with high value of Median of amount of recharges done in main account over last 90 days at user level (in Indonasian Rupiah)(medianamnt_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 17.Customers with high value of Median of main account balance just before recharge in last 90 days at user level (in Indonasian Rupiah)(medianmarechprebal90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 18.Customers with high value of Number of loans taken by user in last 30 days(cnt_loans30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 19.Customers with high value of Total amount of loans taken by user in last 30 days(amnt_loans30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 20.Customers with high value of maximum amount of loan taken by the user in last 30 days(maxamnt_loans30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 21.Customers with high value of Number of loans taken by user in last 90 days(cnt_loans90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 22.Customers with high value of Total amount of loans taken by user in last 90 days(amnt_loans90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 23.Customers with high value of maximum amount of loan taken by the user in last 90 days(maxamnt_loans90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 24.Customers with high value of Average payback time in days over last 30 days(payback30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 25.Customers with high value of Average payback time in days over last 90 days(payback90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 26.In between 6th and 7th month maximum customers both defualters and Non-defaulters have paid there loan amount.
- ✓ 27.Below 14th of each month all the customers have paid there loan amount.

3.5 Run and Evaluate selected models

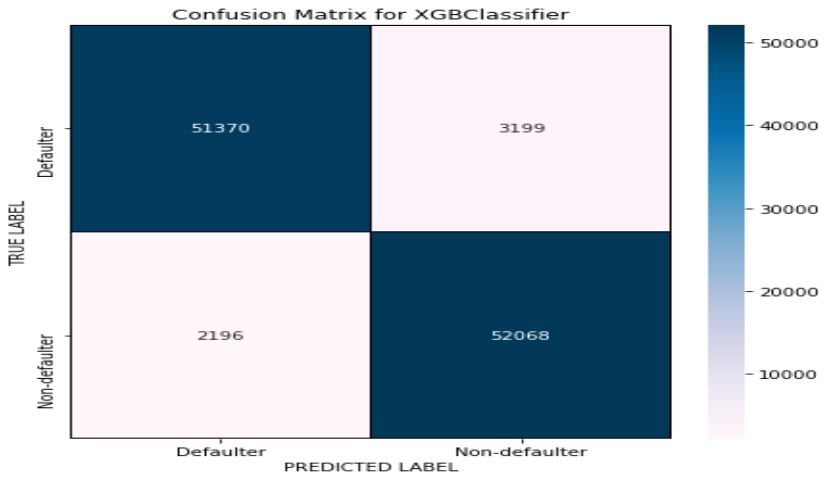
1. Model Building:

1) XGBClassifier:

```
In [74]: XGB=XGBClassifier(verbosity=0)
         XGB.fit(X_train,y_train)
         predxg=XGB.predict(X_test)
         Accuracy_Score = accuracy_score(y_test, predxg)*100
         print('Accuracy Score:',Accuracy_Score)
         print('Confusion Matrix:',confusion_matrix(y_test, predxg))
         print(classification_report(y_test,predxg))

         #cross validation score
         scores = cross_val_score(XGB, X, y, cv = 5).mean()*100
         print("\nCross validation score :", scores)

         #difference of accuracy and cv score
         diff = Accuracy_Score - scores
         print("\Accuracy_Score - Cross Validation Score :", diff)
```

```
Accuracy Score: 95.04286383725524
Confusion Matrix: [[51370  3199]
 [ 2196 52068]]
              precision    recall  f1-score   support

           0       0.96      0.94      0.95     54569
           1       0.94      0.96      0.95     54264

    accuracy                           0.95    108833
   macro avg       0.95      0.95      0.95    108833
weighted avg       0.95      0.95      0.95    108833

Cross validation score : 93.56273689827947
\Accuracy_Score - Cross Validation Score : 1.480126938975772
```

- XGBClassifier has given me 95% accuracy and the difference between model accuracy and cross validation score is 1.48%, but still we have to look into multiple models.



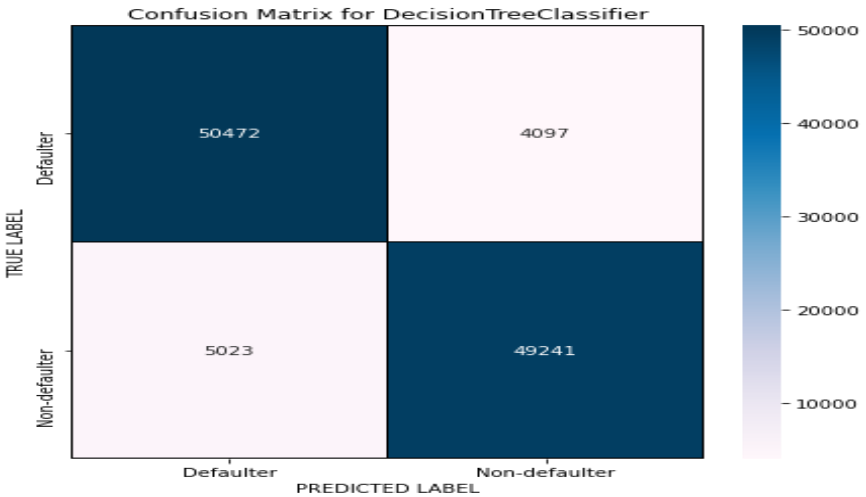
- We can see the true values and predicted values in XGB Classifier model using confusion matrix.

2) DecisionTreeClassifier:

```
In [80]: DTC=DecisionTreeClassifier()
         DTC.fit(X_train,y_train)
         preddt=DTC.predict(X_test)
         Accuracy_Score = accuracy_score(y_test, preddt)*100
         print('Accuracy Score:',Accuracy_Score)
         print('Confusion Matrix:',confusion_matrix(y_test, preddt))
         print(classification_report(y_test,preddt))

         #cross validation score
         scores = cross_val_score(DTC, X, y, cv = 5).mean()*100
         print("\nCross validation score :", scores)

         #difference of accuracy and cv score
         diff = Accuracy_Score - scores
         print("\Accuracy_Score - Cross Validation Score :", diff)
```

```
Accuracy Score: 91.62018872952137
Confusion Matrix: [[50472  4097]
 [ 5023 49241]]
              precision    recall  f1-score   support

           0       0.91      0.92      0.92     54569
           1       0.92      0.91      0.92     54264

    accuracy                           0.92    108833
   macro avg       0.92      0.92      0.92    108833
weighted avg       0.92      0.92      0.92    108833


Cross validation score : 90.84616203659101
\Accuracy_Score - Cross Validation Score : 0.774026692930363
```

- DecisionTreeClassifier is giving me 92% accuracy and the difference between model accuracy and cross validation score is 0.77%.



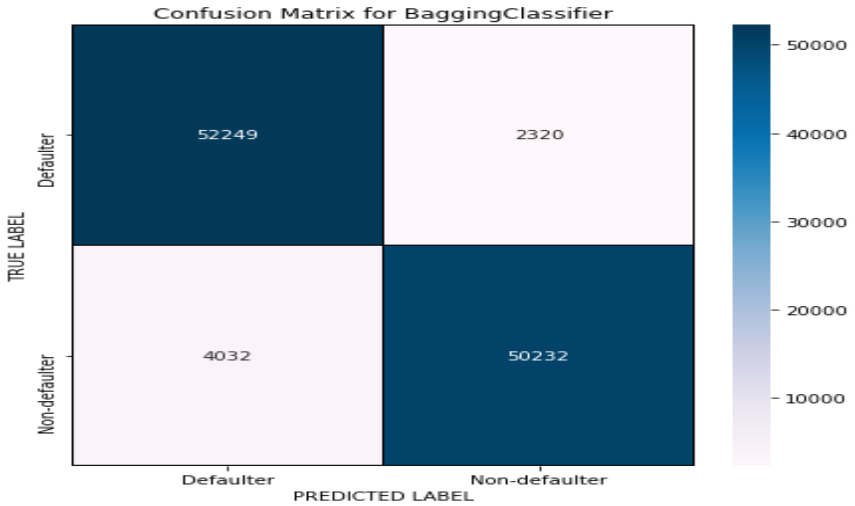
- We can see the true values and predicted values in DecisionTreeClassifier model using confusion matrix.

3) BaggingClassifier:

```
In [82]: BC=BaggingClassifier()
         BC.fit(X_train,y_train)
         predbc=BC.predict(X_test)
         Accuracy_Score = accuracy_score(y_test, predbc)*100
         print('Accuracy Score:',Accuracy_Score)
         print('Confusion Matrix:',confusion_matrix(y_test, predbc))
         print(classification_report(y_test,predbc))

         #cross validation score
         scores = cross_val_score(BC, X, y, cv = 5).mean()*100
         print("\nCross validation score :", scores)

         #difference of accuracy and cv score
         diff = Accuracy_Score - scores
         print("\Accuracy_Score - Cross Validation Score :", diff)
```

```
Accuracy Score: 94.16353495722805
Confusion Matrix: [[52249  2320]
 [ 4032 50232]]
              precision    recall  f1-score   support

           0       0.93      0.96      0.94     54569
           1       0.96      0.93      0.94     54264

    accuracy                           0.94    108833
   macro avg       0.94      0.94      0.94    108833
weighted avg       0.94      0.94      0.94    108833


Cross validation score : 93.71983473253724
\Accuracy_Score - Cross Validation Score : 0.44370022469081505
```

- BaggingClassifier is giving me 94% accuracy and the difference between model accuracy and cross validation score is 0.44%.



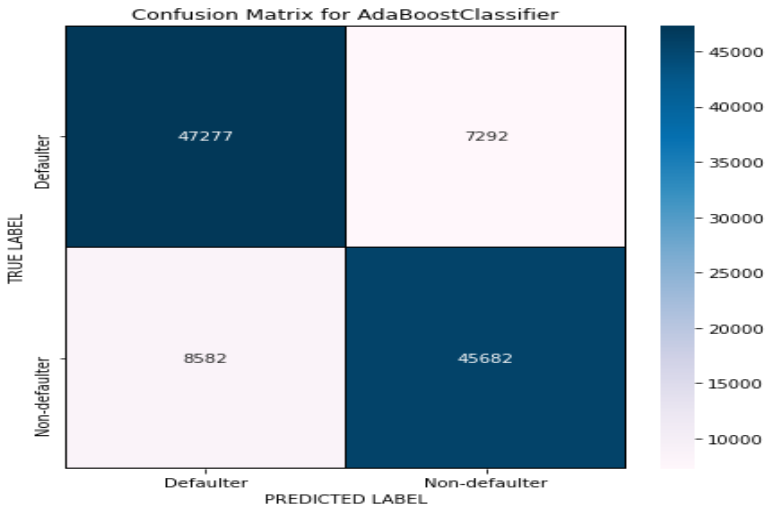
- We can see the true values and predicted values in BaggingClassifier model using confusion matrix.

4) AdaBoostClassifier:

```
In [84]: ABC=AdaBoostClassifier()
         ABC.fit(X_train,y_train)
         predab=ABC.predict(X_test)
         Accuracy_Score = accuracy_score(y_test, predab)*100
         print('Accuracy Score:',Accuracy_Score)
         print('Confusion Matrix:',confusion_matrix(y_test, predab))
         print(classification_report(y_test,predab))

         #cross validation score
         scores = cross_val_score(ABC, X, y, cv = 5).mean()*100
         print("\nCross validation score :", scores)

         #difference of accuracy and cv score
         diff = Accuracy_Score - scores
         print("\Accuracy_Score - Cross Validation Score :", diff)
```

```
Accuracy Score: 85.4143504268007
Confusion Matrix: [[47277  7292]
 [ 8582 45682]]
              precision    recall  f1-score   support

           0       0.85      0.87      0.86     54569
           1       0.86      0.84      0.85     54264

    accuracy                           0.85    108833
   macro avg       0.85      0.85      0.85    108833
weighted avg       0.85      0.85      0.85    108833


Cross validation score : 84.93754741109406
\Accuracy_Score - Cross Validation Score : 0.4768030157066363
```

- AdaBoostClassifier is giving me 85% accuracy and the difference between model accuracy and cross validation score is 0.48%.

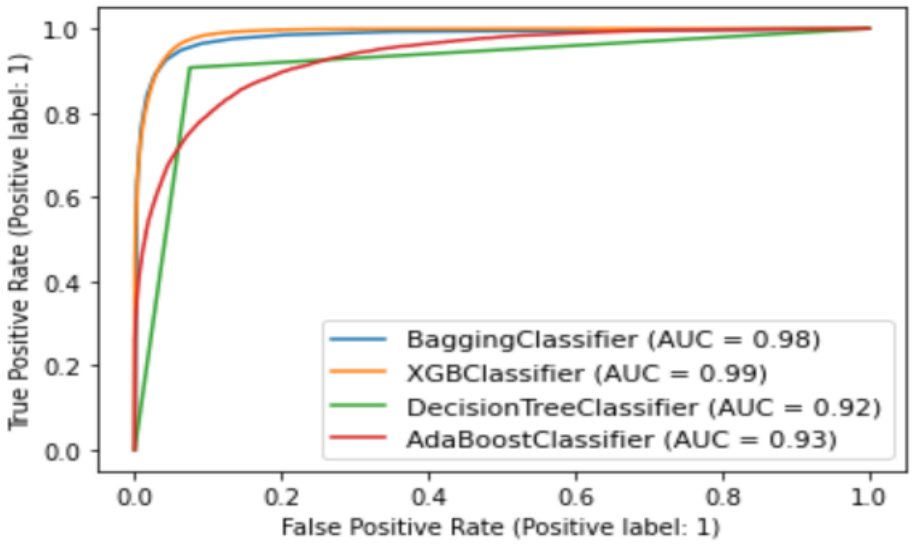


- We can see the true values and predicted values in AdaboostClassifier model using confusion matrix.

✓ **By looking into the difference of model accuracy and cross validation score i found BaggingClassifier as the best model with 95.16% accuracy and the difference between model accuracy and cross validation score is 0.44.**

2. ROC-AUC Curve:

In [86]:
```
# Plotting ROC for all the models used here
from sklearn import datasets
from sklearn import metrics
from sklearn import model_selection
from sklearn.metrics import plot_roc_curve
disp = plot_roc_curve(BC,X_test,y_test)
plot_roc_curve(XGB, X_test, y_test, ax=disp.ax_)
plot_roc_curve(DTC, X_test, y_test, ax=disp.ax_)
plot_roc_curve(ABC, X_test, y_test, ax=disp.ax_)

plt.legend(prop={'size':11}, loc='lower right')
plt.show()
```



- AUC value is high for XGBClassifier and BaggingClassifier.I got least difference in model accuracy and cross validation score for BaggingClassifier so BC is my best model.

3. Hyper Parameter Tunning:

```
In [87]: #importing necessary libraries
         from sklearn.model_selection import GridSearchCV
```

```
In [91]: parameter = {'bootstrap':['True','False'],
                     'n_jobs': [-2,-1,1,2],
                     'n_estimators':[10,20,30,40],
                     'warm_start':['True','False']}
```

Giving the parameters list for BaggingClassifier model.

```
In [92]: GCV=GridSearchCV(BaggingClassifier(),parameter,cv=5)
```

Running grid search CV for BaggingClassifier.

```
In [93]: GCV.fit(X_train,y_train)
```

```
Out[93]: GridSearchCV(cv=5, estimator=BaggingClassifier(),
                      param_grid={'bootstrap': ['True', 'False'],
                                  'n_estimators': [10, 20, 30, 40],
                                  'n_jobs': [-2, -1, 1, 2],
                                  'warm_start': ['True', 'False']})
```

Training the model with GCV.

```
In [94]: GCV.best_params_
```

```
Out[94]: {'bootstrap': 'True', 'n_estimators': 40, 'n_jobs': -1, 'warm_start': 'True'}
```

Got the best parameters for BaggingClassifier.

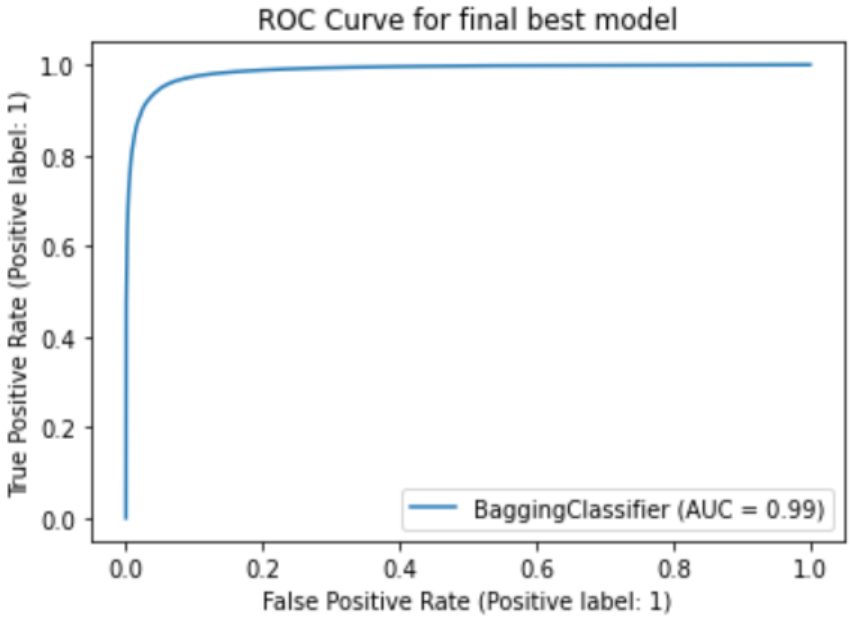
```
In [95]: Final_mod=BaggingClassifier(bootstrap='True', n_jobs=-1,warm_start='True', n_estimators=40)
         Final_mod.fit(X_train,y_train)
         pred=Final_mod.predict(X_test)
         acc=accuracy_score(y_test, pred)

         print('Accuracy Score:',(accuracy_score(y_test,pred)*100))
         print('Confusion matrix:',confusion_matrix(y_test,pred))
         print(classification_report(y_test,pred))
```

```
Accuracy Score: 94.81958597116683
Confusion matrix: [[52057  2512]
 [ 3126 51138]]
              precision    recall  f1-score   support

           0       0.94      0.95      0.95     54569
           1       0.95      0.94      0.95     54264

    accuracy                           0.95    108833
   macro avg       0.95      0.95      0.95    108833
weighted avg       0.95      0.95      0.95    108833
```

- **I have choosed all parameters of BaggingClassifier, after tunning the model with best parameters I have incresed my model accuracy from 94.16% to 94.82%.**

ROC Curve for final model:

```
In [96]: #Ploting ROC curve for final best model
         plot_roc_curve(Final_mod, X_test, y_test)
         plt.title('ROC Curve for final best model')
         plt.show()
```



- Great after hyperparameter tuning we got improvement in roc curve and AUC also.

3. Saving the model and Predictions:

- I have saved my best model using .pkl as follows**.**

```
In [98]: #Saving the model as .pkl file
         import joblib
         joblib.dump(Final_mod,"MicroCreditLoan.pkl")
Out[98]: ['MicroCreditLoan.pkl']
```

- Now loading my saved model and predicting the test values.

```
In [99]: # Loading the saved model
         model=joblib.load("MicroCreditLoan.pkl")

         #Prediction
         prediction = model.predict(X_test)
         prediction

Out[99]: array([1, 0, 0, ..., 1, 0, 0], dtype=int64)
```

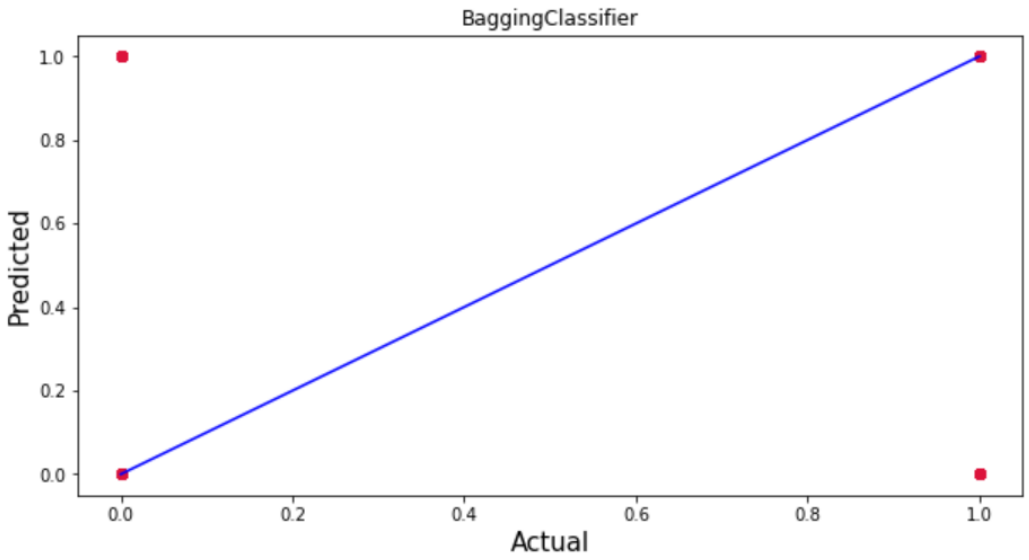
```
In [100]: pd.DataFrame([model.predict(X_test)[:],y_test[:]],index=["Predicted","Actual"])
```

Out[100]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
Predicted	1	0	0	0	0	1	0	1	1	0	0	0	0	0	1	1	1	1	0	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	0	1
Actual	1	0	0	0	0	1	0	1	1	0	1	0	0	0	1	1	1	1	0	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	1	1

Above are the predicted values and the actual values.They are almost similar.

```
In [101]: plt.figure(figsize=(10,5))
          plt.scatter(y_test, prediction, c='crimson')
          p1 = max(max(prediction), max(y_test))
          p2 = min(min(prediction), min(y_test))
          plt.plot([p1, p2], [p1, p2], 'b-')
          plt.xlabel('Actual', fontsize=15)
          plt.ylabel('Predicted', fontsize=15)
          plt.title("BaggingClassifier")
          plt.show()
```



- Plotting Actual vs Predicted, To get better insight. Bule line is the actual line and red dots are the predicted values.

3.6 Interpretation of the Results

- ✓ The dataset was very challenging to handle it had 37 features with 30days and 90days information of customers.
- ✓ Firstly, the datasets were not having any null values.
- ✓ But there was huge number of zero entries in maximum columns so we have to be careful while going through the statistical analysis of the datasets.
- ✓ And proper ploting for proper type of features will help us to get better insight on the data. I found maximum numerical columns in the dataset so I have choosen bar plot to see the relation between target and features.
- ✓ I notice a huge amount of outliers and skewness in the data so we have choose proper methods to deal with the outliers and skewness. If we ignore this outliers and skewness we may end up with a bad model which has less accuracy.
- ✓ Then scaling dataset has a good impact like it will help the model not to get baised. Since we have not removed outliers and skewness completely from the dataset so we have to choose Normalization.
- ✓ We have to use multiple models while building model using dataset as to get the best model out of it.
- ✓ And we have to use multiple metrics like F1_score, precision, recall and accuracy_score which will help us to decide the best model.
- ✓ I found BaggingClassifier as the best model with 94.16% accuracy_score. Also I have improved the accuracy of the best model by running hyper parameter tunning.
- ✓ At last I have predicted wheather the loan is paid back or not using saved model. It was good!! that I was able to get the predictions near to actual values.

4.CONCLUSION

4.1 Key Findings and Conclusions of the Study

In this project report, we have used machine learning algorithms to predict the micro credit defaulters. We have mentioned the step by step procedure to analyze the dataset and finding the correlation between the features. Thus we

can select the features which are correlated to each other and are independent in nature. These feature set were then given as an input to four algorithms and a hyper parameter tunning was done to the best model and the accuracy has been improved. Hence we calculated the performance of each model using different performance metrics and compared them based on these metrics. Then we have also saved the best model and predicted the label. It was good the the predicted and actual values were almost same.

4.2 Learning Outcomes of the Study in respect of Data Science

I found that the dataset was quite interesting to handle as it contains all types of data in it. Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed and analysed. New analytical techniques of machine learning can be used in property research. The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say. Data cleaning is one of the most important steps to remove unrealistic values and zero values. This study is an exploratory attempt to use four machine learning algorithms in estimating micro credit defaulter, and then compare their results.

To conclude, the application of machine learning in micro credit is still at an early stage. We hope this study has moved a small step ahead in providing some methodological and empirical contributions to crediting institutes, and presenting an alternative approach to the valuation of defaulters. Future direction of research may consider incorporating additional micro credit transaction data from a larger economical background with more features.

4.3 Limitations of this work and Scope for Future Work

- ✓ First draw back is the length of the dataset it is very huge and hard to handle.
- ✓ Followed by more number of outliers and skewness these two will reduce our model accuracy.

- ✓ Also, we have tried best to deal with outliers, skewness and zero values. So it looks quite good that we have achieved a accuracy of 94.82% even after dealing all these drawbacks.
- ✓ Also, this study will not cover all Classification algorithms instead, it is focused on the chosen algorithm, starting from the basic ensembling techniques to the advanced ones.