PROJECT REPORT ON:

"Micro-Credit Defaulter Model"

SUBMITTED BY

Rahul M

ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of this project would be incomplete without mentioning the people who made it possible, without whose constant guidance and encouragement would have made efforts to go in vain. I consider myself privileged to express gratitude and respect towards all those who guided me through the completion of this project.

I respect and thank **Flip Robo Technologies**, for providing me an opportunity to do the project work and giving me all the support and guidance, which helped me complete the project duly.

I am extremely thankful to the SME **Mr.Shwetank Mishra** to helping me out wherever required.

I am thankful to and all fortunate enough to get constant encouragement, support and guidance from all Teaching staffs and Data Scientist of **DataTrained** which helped me in successfully completing 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:

The Problem statement here is Telecom company is collaborating with an MFI to provide micro-credit on mobile balances to customers that has to be paid back in 5 days. 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

1.2 Conceptual Background of the Domain Problem

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.

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

Microfinance: Microfinance is an economic development approach that involves providing financial services, through institutions, to low-income clients, where the market fails to provide appropriate services. The services provided by the Microfinance Institutions (MFIs) include credit saving and insurance services. Many microfinance institutions also provide social intermediation services such as training and education, organizational support, health and skills in line with their development objectives.

Micro-credit: It is a component of microfinance and is the extension of small loans to entrepreneurs, who are too poor to qualify for traditional bank loans. Especially in developing countries, micro-credit enables very poor people to engage in self-employment projects that generate income, thus allowing them to improve the standard of living for themselves and their families.

Micro finance Institutions (MFIs): A microfinance institution is an organization, engaged in extending micro credit loans and other financial services to poor borrowers for income generating and self employment activities. An MFI is usually not a part of the formal banking industry or government. It is usually referred to as a NGO (Non-Government Organization)

CHARACTERISTICS OF MICROFINANCE:

Microfinance gives access to financial and non-financial services to low-income people, who wish to access money for starting or developing an income generation activity. The individual loans and savings of the poor clients are small. Microfinance came into being from the appreciation that micro-entrepreneurs and some poorer clients can be 'bankable', that is, they can repay, both the principal and interest, on time and also make savings, provided financial services are tailored to suit their needs. Microfinance as a discipline has created financial products and services that together have enabled low-income people to become clients of a banking intermediary. The characteristics of microfinance products include:

- Little amounts of loans and savings.
- Short- terms loan (usually up to the term of one year).
- Payment schedules attribute frequent installments (or frequent deposits).

• Installments made up from both principal and interest, which amortized in course of time.

• Higher interest rates on credit (higher than commercial bank rates but lower than loan-shark rates), which reflect the labor-intensive work associated with making small loans and allowing the microfinance intermediary to become sustainable over time.

• Easy entrance to the microfinance intermediary saves the time and money of the client and permits the intermediary to have a better idea about the clients' financial and social status.

• Application procedures are simple.

• Short processing periods (between the completion of the application and the disbursement of the loan).

• The clients who pay on time become eligible for repeat loans with higher amounts.

• The use of tapered interest rates (decreasing interest rates over several loan cycles) as an incentive to repay on time. Large size loans are less costly to the MFI, so some lenders provide large size loans on relatively lower rates.

• No collateral is required contrary to formal banking practices. Instead of collateral, microfinance intermediaries use alternative methods, like, the assessments of clients' repayment potential by running cash flow analyses, which is based on the stream of cash flows, generated by the activities for which loans are taken

METHODOLOGY OF MICROFINANCE :

Majority of the microfinance institutions offer and provide credit on a solidarity-group lending basis without collateral. There is also a range of other methodologies that MFIs follow. Some MFIs start with one methodology and later on move or diversify to another methodology so that they do not exclude certain socio-economic categories of clients. So it becomes important to have a basic understanding of methodologies and activity of MFIs.

Group Lending: Group based lending is one of the most novel approaches of lending small amounts of money to a large number of clients who cannot offer collateral.

Individual Lending: Unlike MFIs, there are very few conventional financial institutions which provide individual loans to low-income people because poorer clients are considered higher risk clients due to their lack of collateral, plus the labor-intensive nature of the credits and hence the lack of profitability of small-credits.

Credit Unions: Credit unions are the organizations that are formed on the basis of financial relation of savings and loans between its members. They accumulate savings from its members and provide short-term credit to the needed members.

Village Banking: Village banking is a kind of financial services model that assists poor communities to establish their own credit and saving associations, or village banks. Village bank provides non–collateralized loans to its members and a place to invest savings and promote social solidarity.

Self Help Groups/Associations: Rotating Savings and Credit Associations (ROSCAs) exist in several parts of the world but recognized under different names, like as Tontines and Susus. They are known to be female dominated organizations that save small amount of money and members can borrow from common pool on a rotating basis. These types of organizations or self help groups, have sometimes been used by MFI for group lending among the members.

SAVINGS MOBILIZATION: Savings mobilization has recently been recognized as a major force in microfinance. In the past, microfinance focused almost exclusively on credit; savings were the "forgotten half" of financial intermediation.The importance of savings mobilization has been highlighted in several papers in the context of microfinance. Few analyses have been shaped in order to take an in-depth look at the savings mobilization strategies, which are employed by various institutions and are then compared to the results

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.

- Imported dataset and checked the formation of data
- Checked for null values – There was no missing value in the dataset
- 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.

- Plotted HEATMAP for clear understanding of missing values ¬ Correlation is done.
- Correlation HEATMAP is visualised. It is found that some of the feature names are correlated and some of the columns have nearly zero correlation with the 'label'

Also, I found outliers and skewness in the dataset so I processes outliers using standard deviation method and I removed skewness using yeo-johnson

method. I have used all the classification algorithms while building model then tuned 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.

- The feature columns are of type int,float64 and object
- As the object type feature columns ('pcircle', 'pdate') is a single data col umn and also doesn't make much impact with 'label' . These columns ar e dropped. All other Data are described and got a statistical inference.
- It was found that all the feature columns has mean higher than the med ian and there was very large difference between 75th percentile and ma x values.
- This gave a insight that mostly all the columns have very high outliers.
- Count for all the columns was same which implies no null values
- The Dataset is of shape 209593 rows and 33 column

```
1 df.describe()
```

	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cn
count	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	2(
mean	0.875177	8112.343445	5381.402289	6082.515068	2692.581910	3483.406534	3755.847800	3712.202921	2064.452797	
std	0.330519	75696.082531	9220.623400	10918.812767	4308.586781	5770.461279	53905.892230	53374.833430	2370.786034	
min	0.000000	-48.000000	-93.012667	-93.012667	-23737.140000	-24720.580000	-29.000000	-29.000000	0.000000	
25%	1.000000	246.000000	42.440000	42.692000	280.420000	300.260000	1.000000	0.000000	770.000000	
50%	1.000000	527.000000	1469.175667	1500.000000	1083.570000	1334.000000	3.000000	0.000000	1539.000000	
75%	1.000000	982.000000	7244.000000	7802.790000	3356.940000	4201.790000	7.000000	0.000000	2309.000000	
max	1.000000	999860.755168	265926.000000	320630.000000	198926.110000	200148.110000	998650.377733	999171.809410	55000.000000	

We can observe Negetive Values in some columns

-aon
- daily_decr30
- daily_decr90
- rental30
- rental90
-last_rech_date_da
- last_rech_date_ma
- medianmarechprebal30
- medianmarechprebal90

And removed the negative values using Absolute function in these colu mn.

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 rech arge 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 rech arge 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, info etc….. .
- ✓ 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 float values in maxamnt_loans90 to 12 as it is specified in the problem statement we can have only 6,12 as maximum amount of loan taken by the user in last 30 days
- ✓ Also there are unrealistic values in aon as the the indonesian telecom industry was established on May 26, 1995, so till 2023 it has approximatelt 10220 days so anything above is unrealistic value.So processed that as well.

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

```
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')
```

as follows:

- ✓ **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.

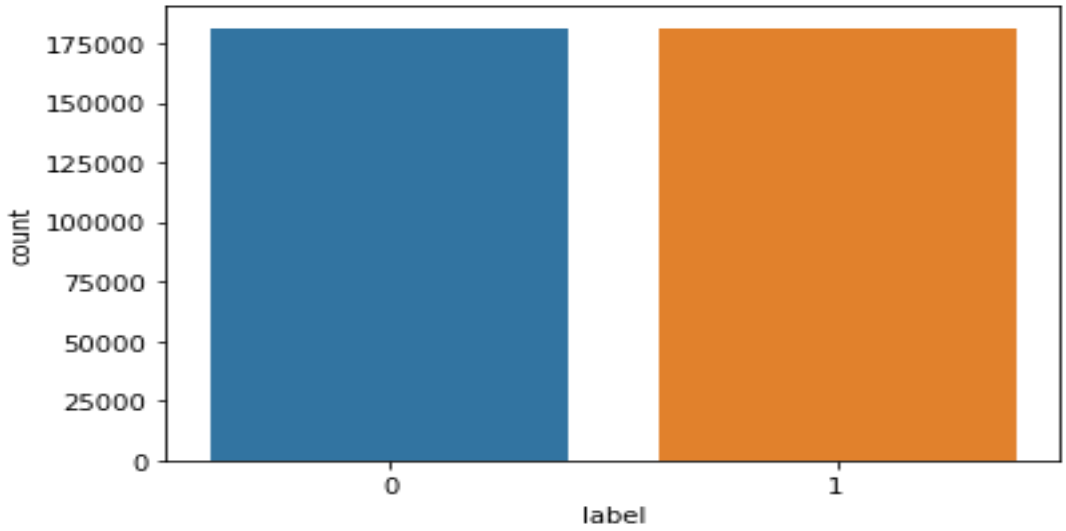
- ✓ from sklearn.ensemble import RandomForestClassifier
- ✓ from sklearn.tree import DecisionTreeClassifier
- ✓ from xgboost import XGBClassifier
- ✓ from sklearn.neighbors import KNeighborsClassifier
- ✓ from sklearn.ensemble import AdaBoostClassifier
- ✓ from sklearn.ensemble import GradientBoostingClassifier
- ✓ 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

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

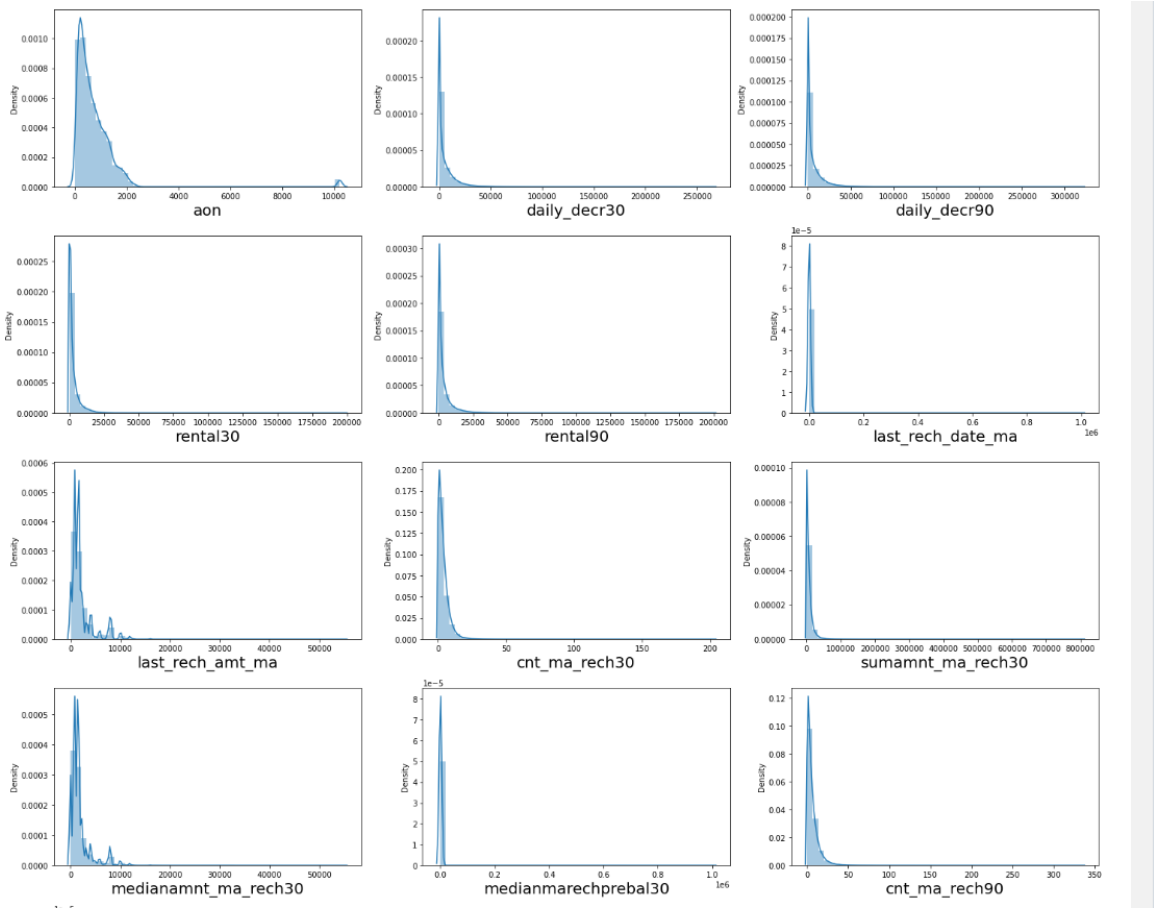
I have used bar plots to see the relation of numerical feature with target and I have used 2 types of plots for numerical columns one is disp plot for univariate and bar plot for bivariate analysis.

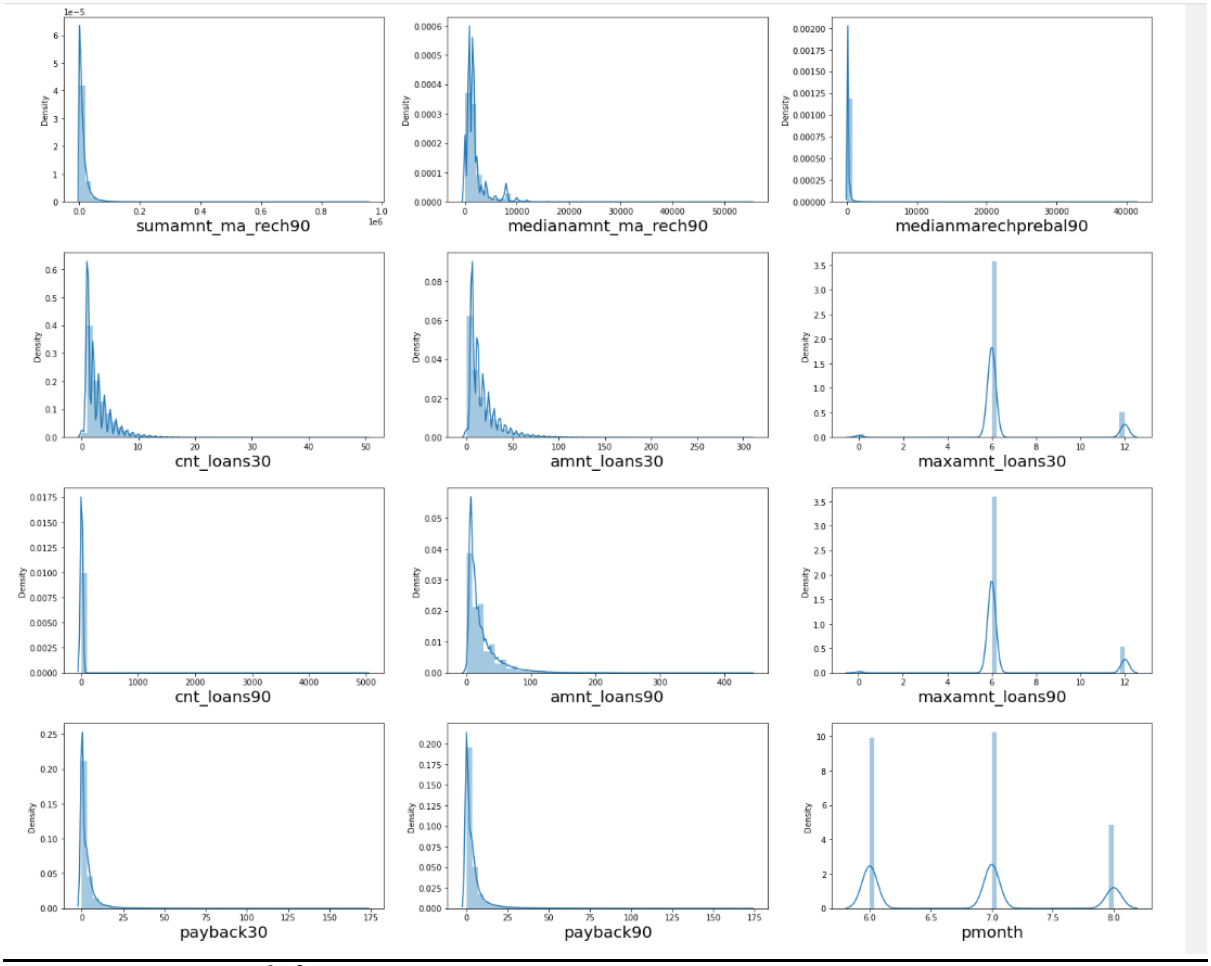
Plotted Distplot : It is found that no feature columns is normally distributed.

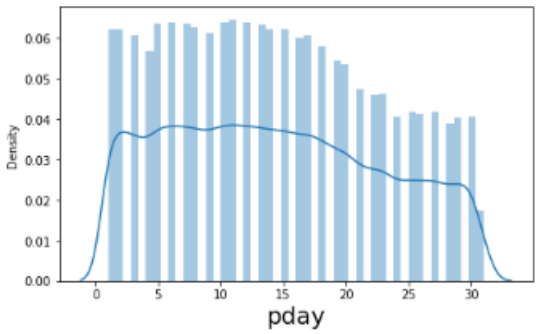
- The available feature column is rightly skewed or has no probability function.
- There is a very high skewness in all the datum that has to be removed . We can also remove feature names which doesn't show probability function or kde as it is zero correlated to columns and have only outliers and zero values which doesn't add any value to the model.
- The unwanted feature columns are identified and removed.
- Plotted Boxplot: This shows the outliers clearly, data on each on different percentiles, whiskers are seen. There are very high outliers in almost every feature columns.
- So we have identified skewness and outliers from graphical representation data. ¬ Removing outliers: I tried different methods like z-score and IQR method, std deviation technique to remove outliers. z-score and IQR outlier removal technique was having high data loss gone with std deviation technique.
- After removing outliers, skewness are removed using Power Transform.
- The data for which skewness is not removed through power transform technique are tried removed using cbrt technique.
- The columns which had High skewness are droped for model performance

- After removing outliers and Skewness, scaled data to check multicollinearity.
- multicollinearity found. Hence proceeded with the droping columns with highest vif value iteratovely based on variance_inflation_factor method
- Thus, data cleaning and EDA done.
- Removing Imbalance in the dataset using SMOTE.

1. Univariate Analysis for numerical columns:

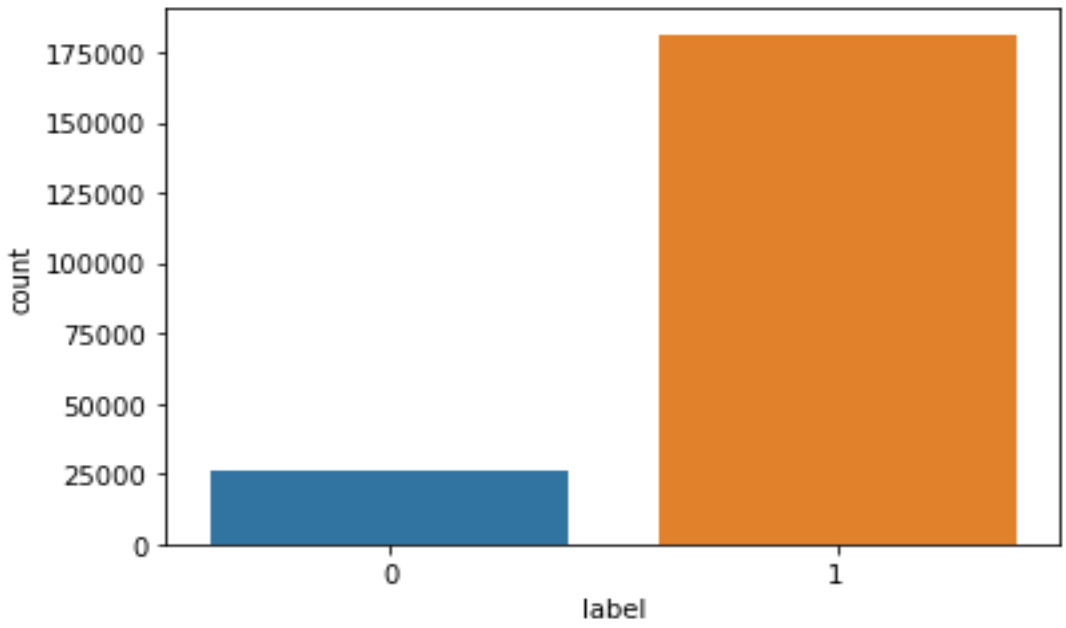






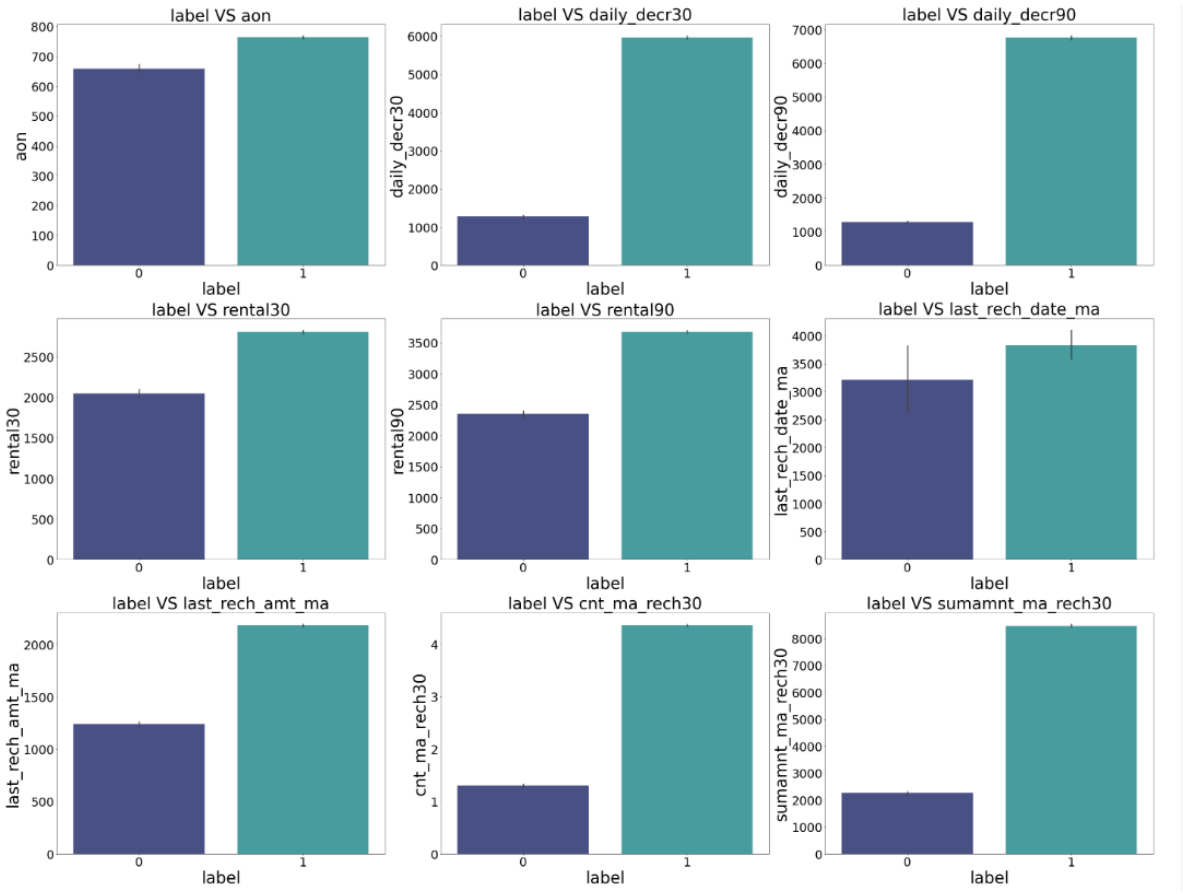
Observations:

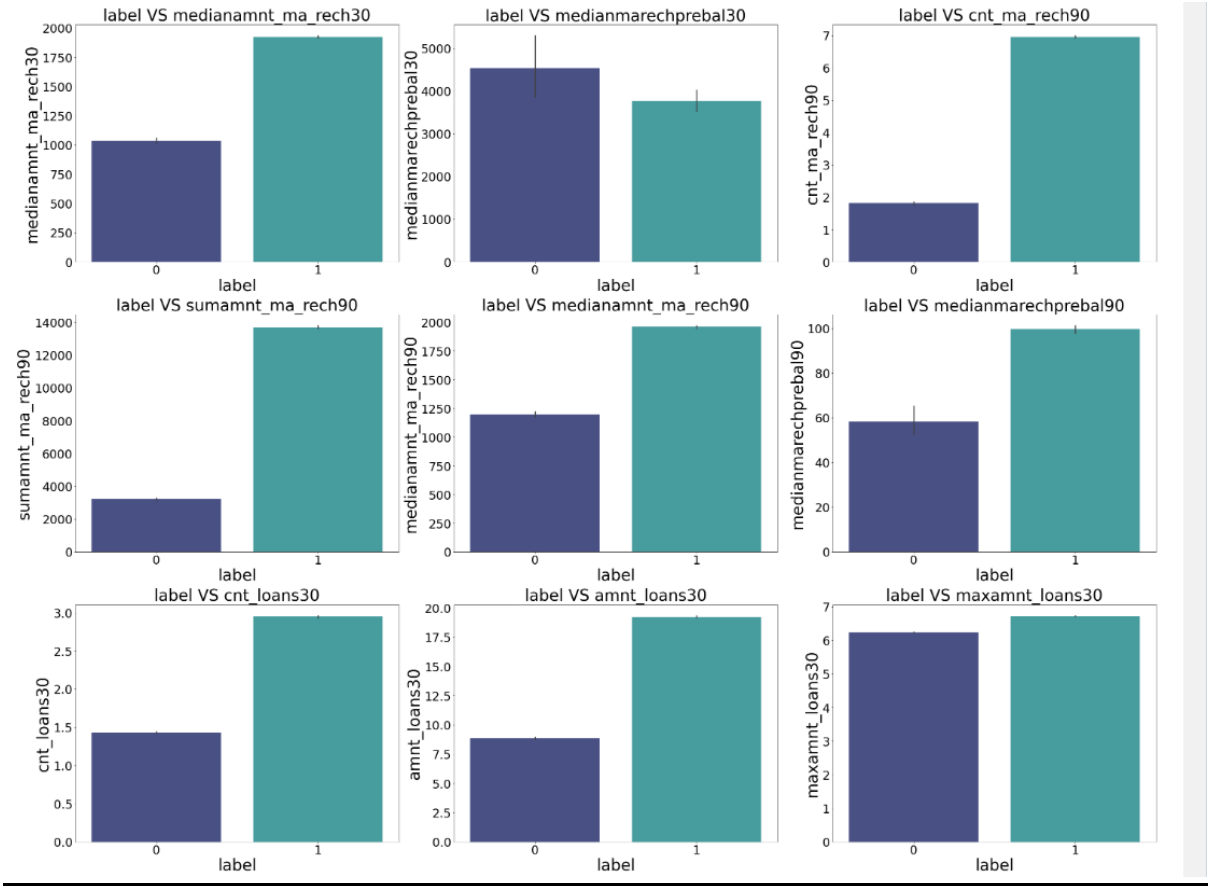
- ✓ We can clearly see that there is skewness in most of the columns so we have to treat them using suitable methods.

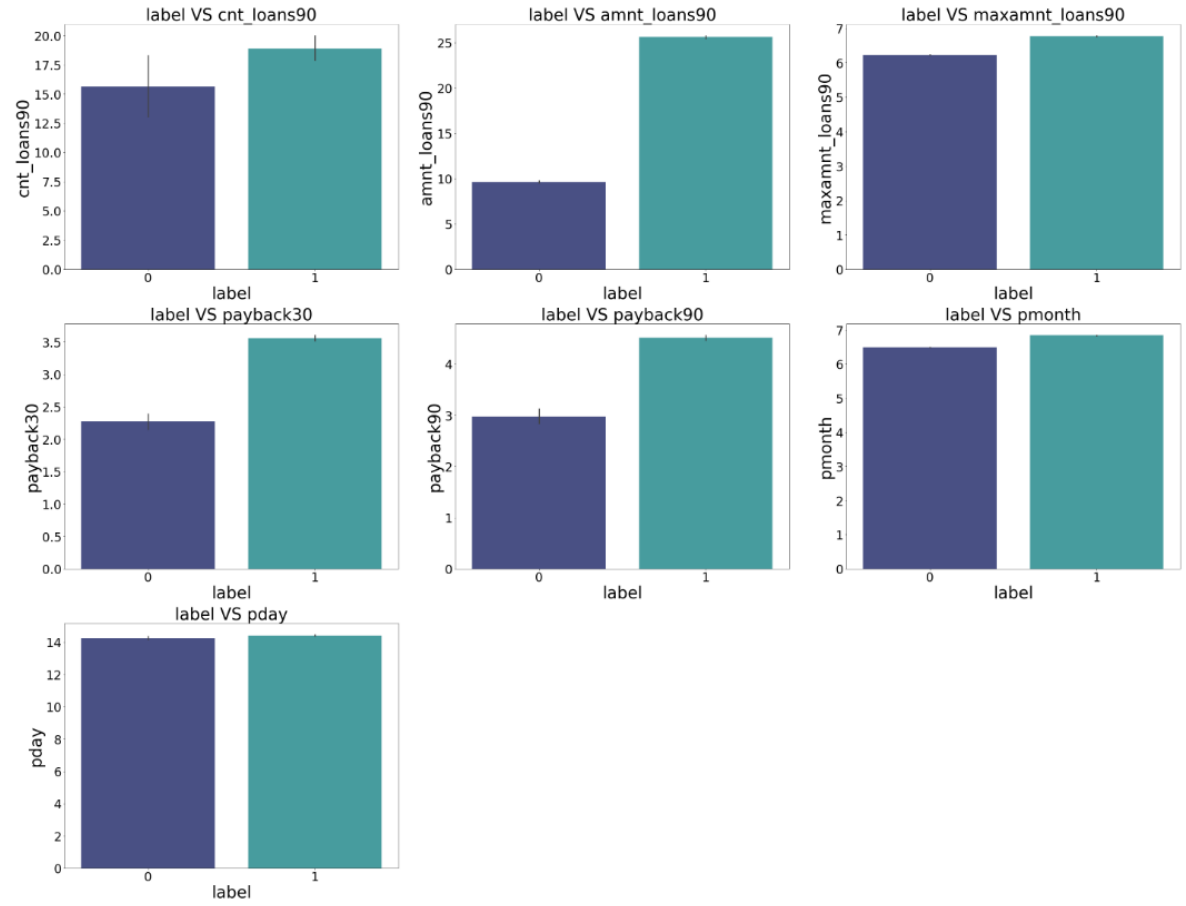


✓ There is a data imbalancing issue so we have to treat this by using oversampling or under sampling.

2. Bivariate analysis for numerical columns:




label VS medianamnt_ma_rech30
medianamnt_ma_rech30
label
label VS medianmarechprebal30
medianmarechprebal30
label
label VS cnt_ma_rech90
cnt_ma_rech90
label
label VS sumamnt_ma_rech90
sumamnt_ma_rech90
label
label VS medianamnt_ma_rech90
medianamnt_ma_rech90
label
label VS medianmarechprebal90
medianmarechprebal90
label
label VS cnt_loans30
cnt_loans30
label
label VS amnt_loans30
amnt_loans30
label
label VS maxamnt_loans30
maxamnt_loans30
label


label VS cnt_loans90
cnt_loans90
label
label VS amnt_loans90
amnt_loans90
label
label VS maxamnt_loans90
maxamnt_loans90
label
label VS payback30
payback30
label
label VS payback90
payback90
label
label VS pmonth
pmonth
label
label VS pday
pday
label

Observations:

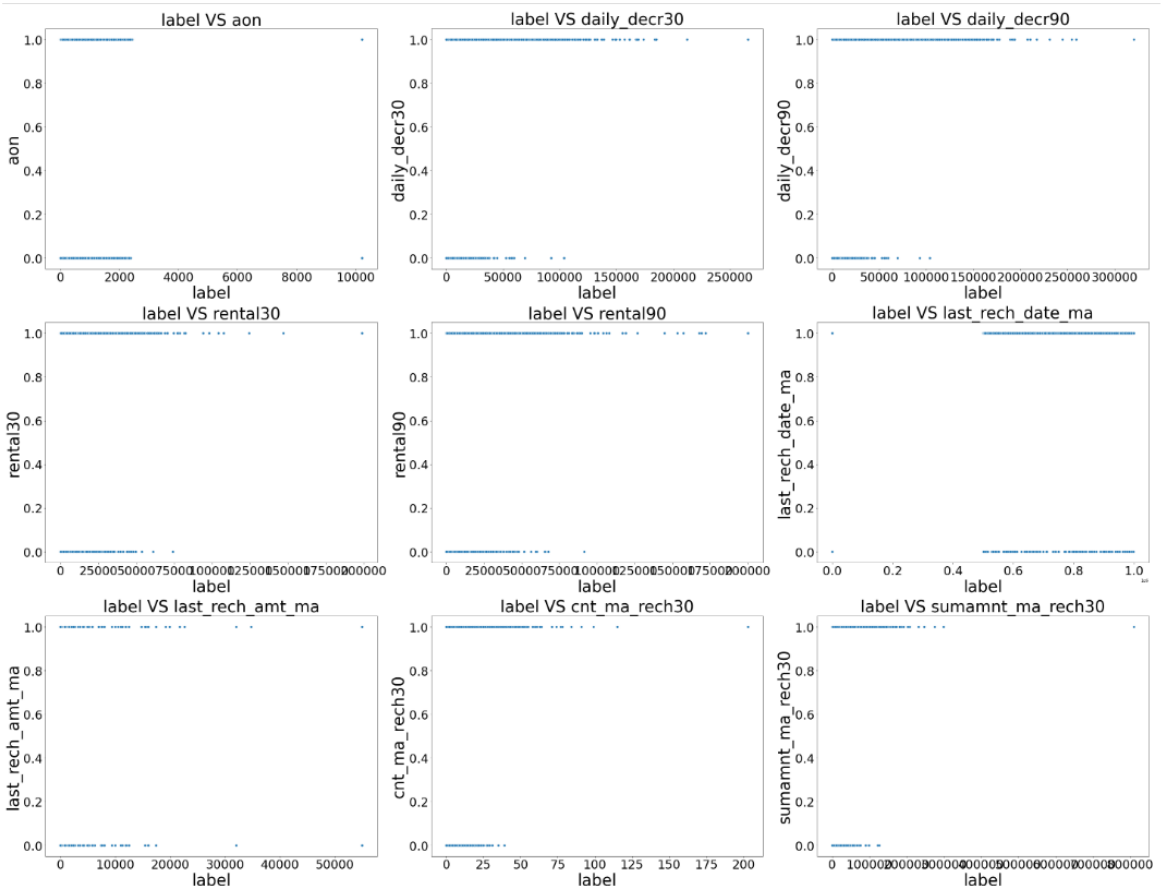
- ✓ 1.Customers with high value of Age on cellular network in days(aon) are maximum Non defaulters(who have paid there loan amount).
- ✓ Non defaulters are more than 700 units
- ✓ 2.We can obseve Customers being high value of Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)(daily_decr30) are maximum Non-defaulters(who have paid there loan amount).
- ✓ 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).
- ✓ 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).
- ✓ 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).
- ✓ 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).
- ✓ 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).

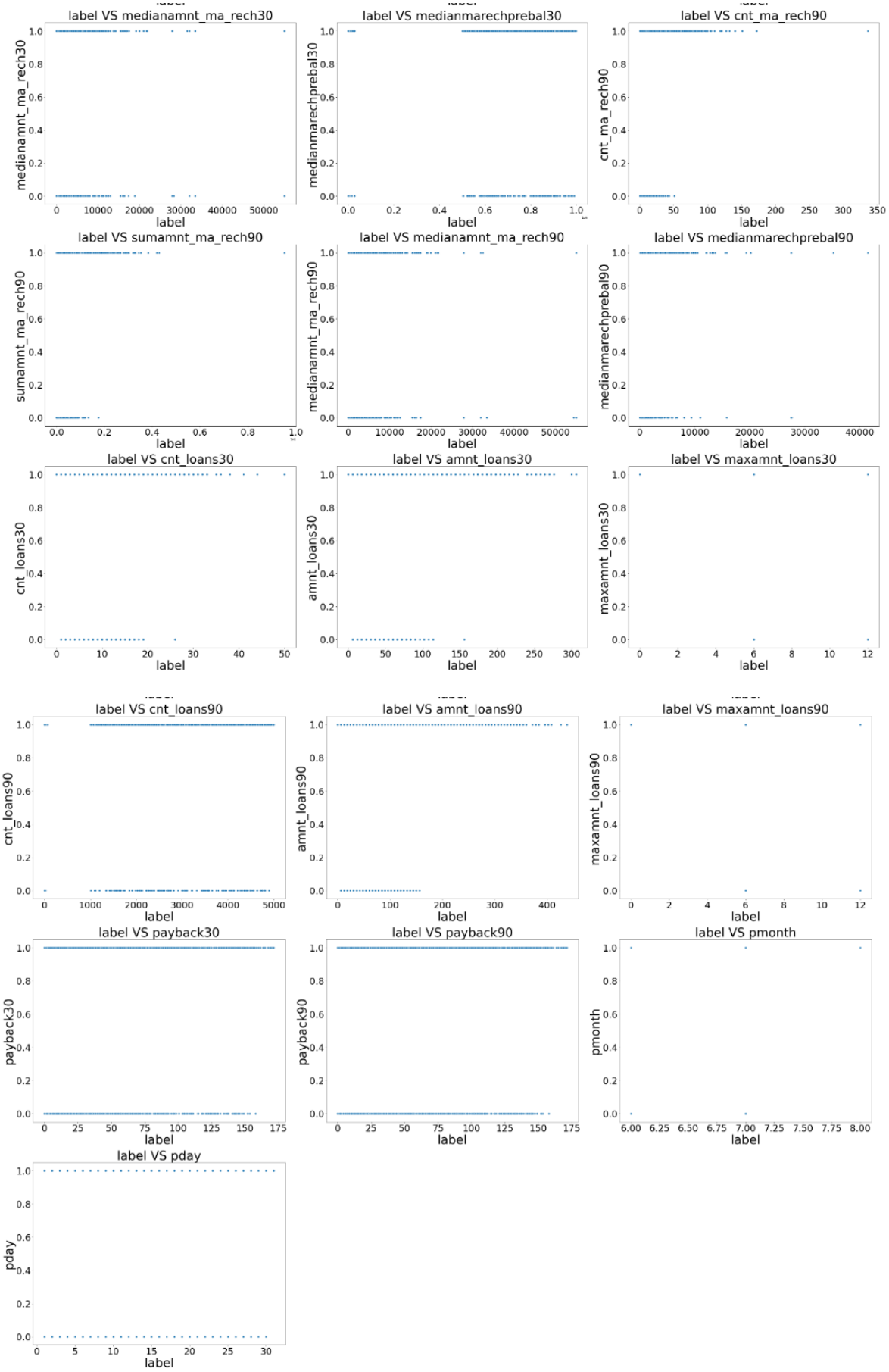
- ✓ 8.Customers with high value of Number of times main account got recharged in last 30 days(cnt_ma_rech30) are maximum Non-defaulters(who have paid there loan amount).
- ✓ 10.Customers with high value of Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)(sumamnt_ma_rech30) are maximum Non-defaulters(who have paid there loan amount).
- ✓ 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).
- ✓ 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).

- ✓ 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).
- ✓ 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).
- ✓ 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).
- ✓ 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).
- ✓ 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).
- ✓ 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).
- ✓ 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
- ✓ 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).
- ✓ 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).
- ✓ 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).
- ✓ 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).
- ✓ 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).

- ✓ 26.In between 6th and 7th month maximum customers both defualters and Non-defaulters have paid there loan amount.
- ✓ 27.Below 14th of each pday all the customers have paid the loan amount.




label VS medianamnt_ma_rech30
medianamnt_ma_rech30
label
label VS medianmarechprebal30
medianmarechprebal30
label
label VS cnt_ma_rech90
cnt_ma_rech90
label
label VS sumamnt_ma_rech90
sumamnt_ma_rech90
label
label VS medianamnt_ma_rech90
medianamnt_ma_rech90
label
label VS medianmarechprebal90
medianmarechprebal90
label
label VS cnt_loans30
cnt_loans30
label
label VS amnt_loans30
amnt_loans30
label
label VS maxamnt_loans30
maxamnt_loans30
label
label VS cnt_loans90
cnt_loans90
label
label VS amnt_loans90
amnt_loans90
label
label VS maxamnt_loans90
maxamnt_loans90
label
label VS payback30
payback30
label
label VS payback90
payback90
label
label VS pmonth
pmonth
label
label VS pday
pday
label

1.In aon column we can observe unifor data distribution between 0 to 2000 units.

2. In daily_decr30 we can obseve the data beinf uniform till 0 to 50000 and rest are scattered throughout .Above 100000 we

can say that customers are Non defaulters

3. In daily_decr90 we can obseve the data beinf uniform till 0 to 50000 and rest are scattered throughout .Above 100000 we

can say that customers are Non defaulters

4. In rental30 we can obseve the data beinf uniform till 0 to 50000 and rest are scattered throughout .

5.In rental90 we can obseve the data beinf uniform till 0 to 50000 and rest are scattered throughout .

6.In last_rech_date_ma the data is distributed above till 1.0 and slightly scattered throughout.

7.In last_rech_amt_ma the data is uniformly Distributed till 8000 and rest are scattered throughout.

8.In cnt_ma_rech30 the data is uniformly distributed till 30 and rest are scattered throughout.

9.In sumamnt_ma_rech30 the data is uniformly distributed till 100000 and rest are scattered throughout.

10.in medianamnt_ma_rech90 the data is uniformly distributed till 100000 and rest are scattered throughout.

11.in medianmarechprebal90 the data is uniformly distributed till less than 100000 and rest are scattered throughout.

12.in cnt_loans30 the data is uniformly distributed till 20 and rest are scattered throughout.

13.in amnt_loans90 the data is uniformly distributed till 150 and rest are scattered throughout.

14.in maxamnt_loans90 the data is scatterefd entirely.

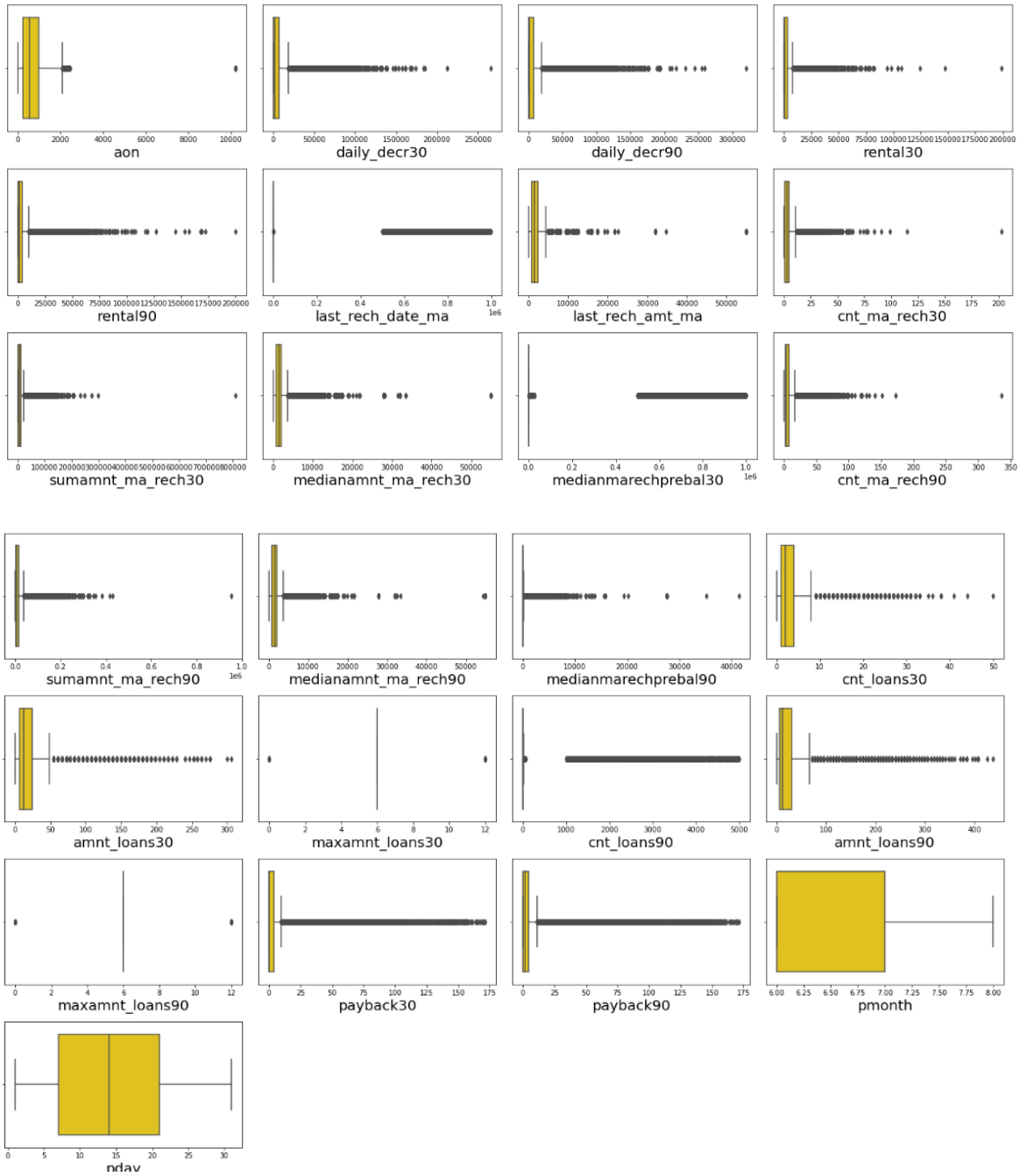
15.in payback30 the data is uniformly scattered till 150 and rest are scattered.

16.in payback90 the data is uniformly scattered till 150 and rest are scattered.

17.in pmonth the data is entirely scattered.

18. in pday the data is entirely scattered.

Outiers Detection



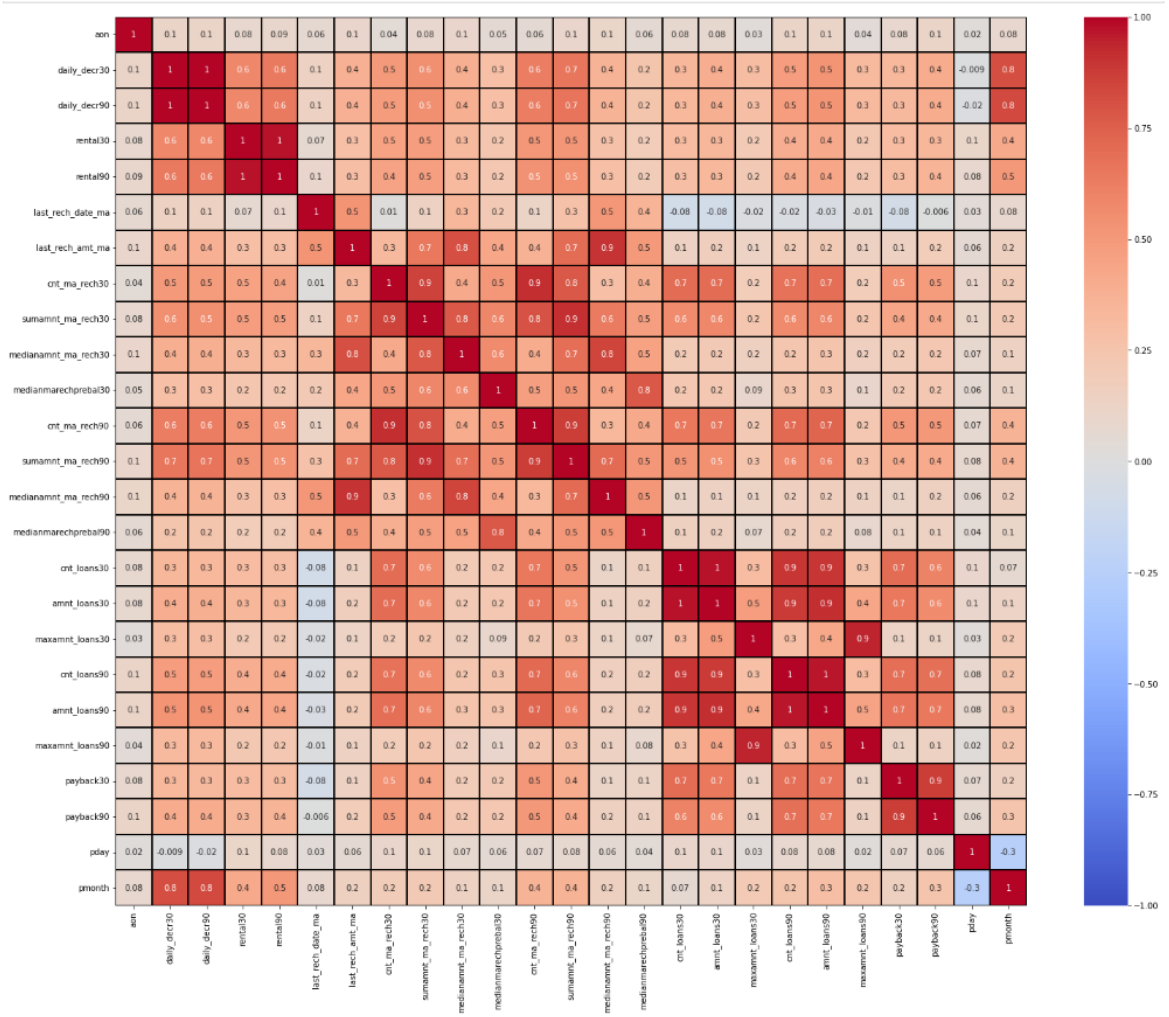
- **Identification of possible problem-solving approaches (methods)**

Since 'label' is of logistic output or categorical data. We can use categorical data parameters and all the available models for logistic regression

- **Identification of possible problem-solving approaches**

Since ‘label’ is of logistic output or categorical data. We can use categorical data parameters and all the available models for logistic regression

Heatmap



We can observe high correlation in some of the features,Lets confirm the same through variance_inflation_factor method and removed those with multicolllinearity by removing the columns iteratively based on the vif score

3.5 Run and Evaluate selected models

1. Model Building:

Listing down all the algorithms used for the training and testing

- **LogisticRegression()**
- **DecisionTreeClassifier()**

- **RandomForestClassifier()**
- **KNeighborsClassifier()**
- **AdaBoostClassifier()**
- **GradientBoostingClassifier()**
- **XGBClassifier()**
- **ExtraTreesClassifier()**

```
#code for Training and Prediction

def eval(x):
    mod=x
    print(mod)
    mod.fit(x_train,y_train)
    pred=mod.predict(x_test)
    print("accuracy score is :",accuracy_score(y_test,pred)*100)
    print("\n")
    print("Confusion Matrix : \n",confusion_matrix(y_test,pred))
    print("\n")
    print("Classification Report  : \n",classification_report(y_test,pred))

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

    #difference of accuracy and cv score
    acc_score=accuracy_score(y_test,pred)*100
    diff = acc_score - scores
    print("difference of accuracy and cross_val_score is :", abs(diff))
```

- **Logistic Regression model is giving an accuracy score of 76.45%**

Logistic Regression

```
eval(lg)
```

```
LogisticRegression()
accuracy score is : 76.22456806808636


Confusion Matrix :
 [[42149 12612]
 [13424 41323]]


Classification Report  :
               precision    recall  f1-score   support

           0       0.76      0.77      0.76     54761
           1       0.77      0.75      0.76     54747

    accuracy                           0.76    109508
   macro avg       0.76      0.76      0.76    109508
weighted avg       0.76      0.76      0.76    109508


Cross validation score : 75.93103021670203
difference of accuracy and cross_val_score is : 0.293537851384329
```

DecisionTreeClassifier model is giving an accuracy score of 91.2%

DECISION TREE CLASSIFIER

```
eval(dtc)
```

```
DecisionTreeClassifier()
accuracy score is : 91.20886145304453


Confusion Matrix :
 [[50378  4383]
 [ 5244 49503]]


Classification Report  :
               precision    recall  f1-score   support

           0       0.91      0.92      0.91     54761
           1       0.92      0.90      0.91     54747

    accuracy                           0.91    109508
   macro avg       0.91      0.91      0.91    109508
weighted avg       0.91      0.91      0.91    109508


Cross validation score : 91.12010548352292
difference of accuracy and cross_val_score is : 0.08875596952161402
```

- **RandomForestClassifier model is giving an accuracy score of 95.1%**

RANDOM FOREST CLASSIFIER

```
eval(rf)
```

```
RandomForestClassifier()
accuracy score is : 95.19030573108815


Confusion Matrix :
 [[52440  2321]
 [ 2946 51801]]


Classification Report  :
              precision    recall  f1-score   support

           0       0.95      0.96      0.95     54761
           1       0.96      0.95      0.95     54747

    accuracy                           0.95    109508
   macro avg       0.95      0.95      0.95    109508
weighted avg       0.95      0.95      0.95    109508


Cross validation score : 95.12776269707663
difference of accuracy and cross_val_score is : 0.06254303401151162
```

- **KNeighborsClassifier model is giving an accuracy score of 89.9%**

KNeighborsClassifier

```
eval(knn)
```

```
KNeighborsClassifier()
accuracy score is : 89.90758666033531


Confusion Matrix :
 [[54228   533]
 [10519 44228]]


Classification Report  :
              precision    recall  f1-score   support

           0       0.84      0.99      0.91     54761
           1       0.99      0.81      0.89     54747

    accuracy                           0.90    109508
   macro avg       0.91      0.90      0.90    109508
weighted avg       0.91      0.90      0.90    109508


Cross validation score : 90.37931622399938
difference of accuracy and cross_val_score is : 0.47172956366406993
```

AdaBoostClassifier model is giving an accuracy score

of 84.39%

Adaboost Classifier

```
eval(ada)
```

```
AdaBoostClassifier()
accuracy score is : 84.52989735909705


Confusion Matrix :
 [[47096  7665]
 [ 9276 45471]]


Classification Report  :
              precision    recall  f1-score   support

           0       0.84      0.86      0.85     54761
           1       0.86      0.83      0.84     54747

    accuracy                           0.85    109508
   macro avg       0.85      0.85      0.85    109508
weighted avg       0.85      0.85      0.85    109508


Cross validation score : 84.39728411126669
difference of accuracy and cross_val_score is : 0.13261324783036343
```

GradientBoostingClassifier model is giving an accuracy score of 89.65%

GradientBoostingClassifier

```
eval(gd)
```

```
GradientBoostingClassifier()
accuracy score is : 89.65646345472477


Confusion Matrix :
 [[49986  4775]
 [ 6552 48195]]


Classification Report  :
               precision    recall  f1-score   support

           0       0.88      0.91      0.90     54761
           1       0.91      0.88      0.89     54747

    accuracy                           0.90    109508
   macro avg       0.90      0.90      0.90    109508
weighted avg       0.90      0.90      0.90    109508


Cross validation score : 89.39939885921075
difference of accuracy and cross_val_score is : 0.2570645955140236
```

XGBClassifier model is giving an accuracy score of 89.65%

XGBClassifier

```
eval(xgb)
```

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, gamma=None,
              gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, n_estimators=100, n_jobs=None,
              num_parallel_tree=None, predictor=None, random_state=None,
              reg_alpha=None, reg_lambda=None, ...)
accuracy score is : 95.01588925010044


Confusion Matrix :
 [[51504  3257]
 [ 2201 52546]]


Classification Report  :
               precision    recall  f1-score   support

           0       0.96      0.94      0.95     54761
           1       0.94      0.96      0.95     54747

    accuracy                           0.95    109508
   macro avg       0.95      0.95      0.95    109508
weighted avg       0.95      0.95      0.95    109508


Cross validation score : 93.62268745229784
difference of accuracy and cross_val_score is : 1.393201797802604
```

ExtraTreesClassifier model is giving an accuracy score of 95.01%

ExtraTreesClassifier

```
eval(etc)
```

```
ExtraTreesClassifier()
accuracy score is : 95.93363041969536


Confusion Matrix :
 [[53496  1265]
 [ 3188 51559]]


Classification Report  :
               precision    recall  f1-score   support

           0       0.94      0.98      0.96     54761
           1       0.98      0.94      0.96     54747

    accuracy                           0.96    109508
   macro avg       0.96      0.96      0.96    109508
weighted avg       0.96      0.96      0.96    109508


Cross validation score : 96.39286269834253
difference of accuracy and cross_val_score is : 0.4592322786471641
```

- Based on the above performance we can confirm the best model as RandomForest classifier with least difference between Accuracy and Crossvalidation score of 0.062.

2. ROC-AUC Curve:

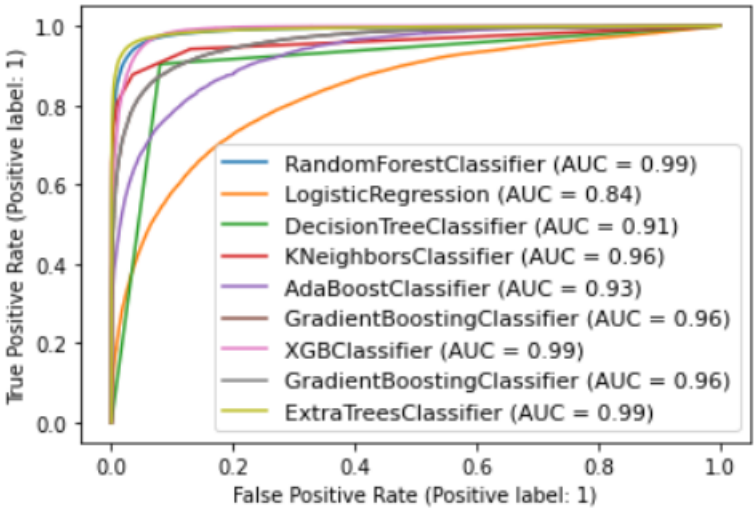
ROC-AUC curve:

```
# 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(best_model,x_test,y_test)

plot_roc_curve(lg, x_test, y_test, ax=disp.ax_)
plot_roc_curve(dtc, x_test, y_test, ax=disp.ax_)
plot_roc_curve(knn, x_test, y_test, ax=disp.ax_)
plot_roc_curve(ada, x_test, y_test, ax=disp.ax_)
plot_roc_curve(gd, x_test, y_test, ax=disp.ax_)
plot_roc_curve(xgb, x_test, y_test, ax=disp.ax_)
plot_roc_curve(gd, x_test, y_test, ax=disp.ax_)
plot_roc_curve(etc, x_test, y_test, ax=disp.ax_)

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



- Random forest model with hypertuned model has the highest performance area.

3. Hyper Parameter Tunning:

HyperParameterTuning RandomForestClassifer using GridSearchCV

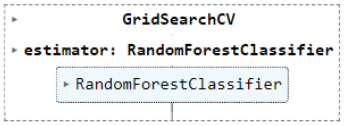
```
#importing gridsearch
from sklearn.model_selection import GridSearchCV


#parameters of RandomForestClassifier
parameters = {
    'n_estimators':[100,200,300],
    'max_features': [None, 'sqrt', 'log2'],
    'criterion' :['gini', 'entropy','log_loss']
}
```

```

grid=GridSearchCV(estimator=rf,param_grid=parameters,cv=5,n_jobs=-1)
print(grid)
#training the model
grid.fit(x_train,y_train)
```

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(), n_jobs=-1,
             param_grid={'criterion': ['gini', 'entropy', 'log_loss'],
                         'max_features': [None, 'sqrt', 'log2'],
                         'n_estimators': [100, 200, 300]})
```



```
print(grid.best_score_)
print(grid.best_estimator_.n_estimators)
print(grid.best_params_)
```

```
0.950109196556034
300
{'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 300}
```

```
#Tuning the model with the best parameters

best_model=RandomForestClassifier(criterion='entropy',n_estimators=300,max_features='log2')
best_model.fit(x_train,y_train)
pred=best_model.predict(x_test)
print("score: ",best_model.score(x_train,y_train))
print('accuracy_score:',accuracy_score(y_test,pred))
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
score:  0.9999804319069499
accuracy_score: 0.9530627899331555
[[52485  2276]
 [ 2864 51883]]
              precision    recall  f1-score   support

           0       0.95      0.96      0.95     54761
           1       0.96      0.95      0.95     54747

    accuracy                           0.95    109508
   macro avg       0.95      0.95      0.95    109508
weighted avg       0.95      0.95      0.95    109508
```

Saving the model

Saving the model using pickle.dump method from the pickle library

```
import pickle
filename2='micro_credit_defaulter_prediction.sav'
pickle.dump(best_model,open(filename2,'wb'))
```

Re-Loading the model for testing

```
load_model=pickle.load(open(filename2,'rb'))
```

```
predictions=load_model.predict_proba(x_test[20000:20100])
predictions
```

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

3.6 Interpretation of the Results

The results interpreted from the visualizations, preprocessing and modelling is that 'RandomForestClassifier' algorithm created the best model to predict whether customer is defaulter or not. Such that the MFI can select customers to provide future micro credit loans

After Training and Testing six algorithm model. The best accuracy model was determined as random forest classifier with 97% true accuracy after all the data cleaning, pre-processing, training and prediction as well as evaluation phase

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. 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

The limitations of this solution provided is the model created by normalising a lot of unrealistic and imbalanced data found in the dataset. If there is availability of more realistic date in the dataset will help in further extending this study and improving the results.