FLIP ROBO

Micro Credit Loan Defaulter Project

Submitted by:

Kundan Patil

ACKNOWLEDGMENT

"We would like to express our sincere gratitude to all those who have contributed to the success of our Micro Credit Loan Defaulter Project. Our deepest appreciation goes to our project supervisor for their guidance and support throughout the project. We would also like to thank our colleagues and peers for their valuable input and suggestions. Lastly, we would like to acknowledge the support and cooperation of the micro credit loan defaulters for their participation in this project. Thank you all for your dedication and hard work."

1) https://www.google.com/
2) https://www.youtube.com/
3) https://scikit-learn.org/stable/user_guide.html

4) https://github.com/
5) https://www.kaggle.com/

INTRODUCTION

- ## 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.

- ## Conceptual Background of the Domain Problem

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.

They 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).

- ## Review of Literature

1. What is Microfinance?

"Microfinance" is often seen as financial services for poor and low-income clients. In practice, the term is often used more narrowly to refer to loans and other services from providers that identify themselves as "microfinance institutions" (MFIs). Microfinance can also be described as a setup of a number of different operators focusing on the financially under-served people with the aim of satisfying their need for poverty alleviation, social promotion, emancipation, and inclusion. Microfinance institutions reach and serve their target market in very innovative ways. Microfinance operations differ in principle, from the standard disciplines of general and entrepreneurial finance. This difference can be attributed to the fact that the size of the loans granted with microcredit is typically too small to finance growth-oriented business projects. Some unique features of microfinance as follows:

i. Delivery of very small loans to unsalaried workers.

ii. Little or no collateral requirements.

iii. Group lending and liability.

iv. Pre-loan savings requirement.

v. Gradually increasing loan sizes.

Implicit guarantee of ready access to future loans if present loans are repaid fully and promptly Microfinance is seen as a catalyst for poverty alleviation, delivered in innovative and sustainable ways to assist the underserved poor, especially in developing countries.

2. Default in Microfinance

Default in microfinance is the failure of a client to repay a loan. The default could be in terms of the amount to be paid or the timing of the payment.

- Motivation for the Problem Undertaken

This problem is of great importance as it can help the financial institution to reduce the default rate and improve their overall performance. By using machine learning techniques, we can analyze the data and make predictions that can aid in identifying potential defaulters, helping the institution to make informed decisions about lending. Additionally, the use of this model can help the institution to better understand their customer's behavior and tailor their services to better meet their needs. Overall, the goal of this project is to provide valuable insights that can help the institution to improve their performance and better serve their customers.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

n summary, the analytical problem being addressed in this project is to build a model that can accurately predict whether a micro credit loan borrower will default on their loan or not. The data provided by the client includes information about the loan and the borrower's account activity. The problem was framed by first identifying the target variable (default vs non-default) and then analyzing the data to identify patterns and relationships that could be used to make predictions. Data cleaning and preprocessing techniques were applied to handle missing data, outliers, and skewed data.

The analytical approach taken was to use machine learning algorithms to analyze the data and make predictions. The data was separated into two groups, defaulters and non-defaulters, and the valuable customers were identified based on their monthly revenue. The data was then scaled and various classification models were applied, and the Extra Trees Classifier algorithm was found to be the most effective. This algorithm will be used to make predictions about future loan defaults and help the financial institution to improve their lending decisions.

- Data Sources and their formats

The data is been provided by one of our clients from telecom industry. They are a fixed wireless telecommunications network provider and 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.

The data is been given by Indonesian telecom company and they gave it to us in a CSV file, with data description file in excel format. They also had provided the problem statement by explaining what they need from us and also the required criteria to be satisfied.

Let’s check the data now. Below I have attached the snapshot below to give an overview.

```python
#import needed libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno

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

from imblearn.over_sampling import SMOTE
from sklearn.preprocessing import LabelEncoder
from scipy.stats import zscore
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import PowerTransformer
from sklearn.pipeline import Pipeline
from statsmodels.stats.outliers_influence import variance_inflation_factor

#models
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import ExtraTreesClassifier

#model selection
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV

#metrics
from sklearn.metrics import roc_auc_score,roc_curve
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix

#save the model
import joblib
```

In [7]:

```python
#load dataset
df=pd.read_csv('Data file.csv')
```

In [8]:

```python
#display top 5 rows with all columns
pd.set_option('display.max_columns',None)
df.head()
```

Out[8]:

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cnt_ma_rech30	fr_ma_rech30	s
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	1539	2	21.0	
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	5787	1	0.0	
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	1539	1	0.0	
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	947	0	0.0	
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	2309	7	2.0	

Here we are taking a look at the first 5 rows of our dataset. It shows that we have a total of 209593 rows and 37 columns present in our dataframe. We have the label column that stores the defaulter and non-defaulter values marked with 0 and 1 making this a Classification problem!

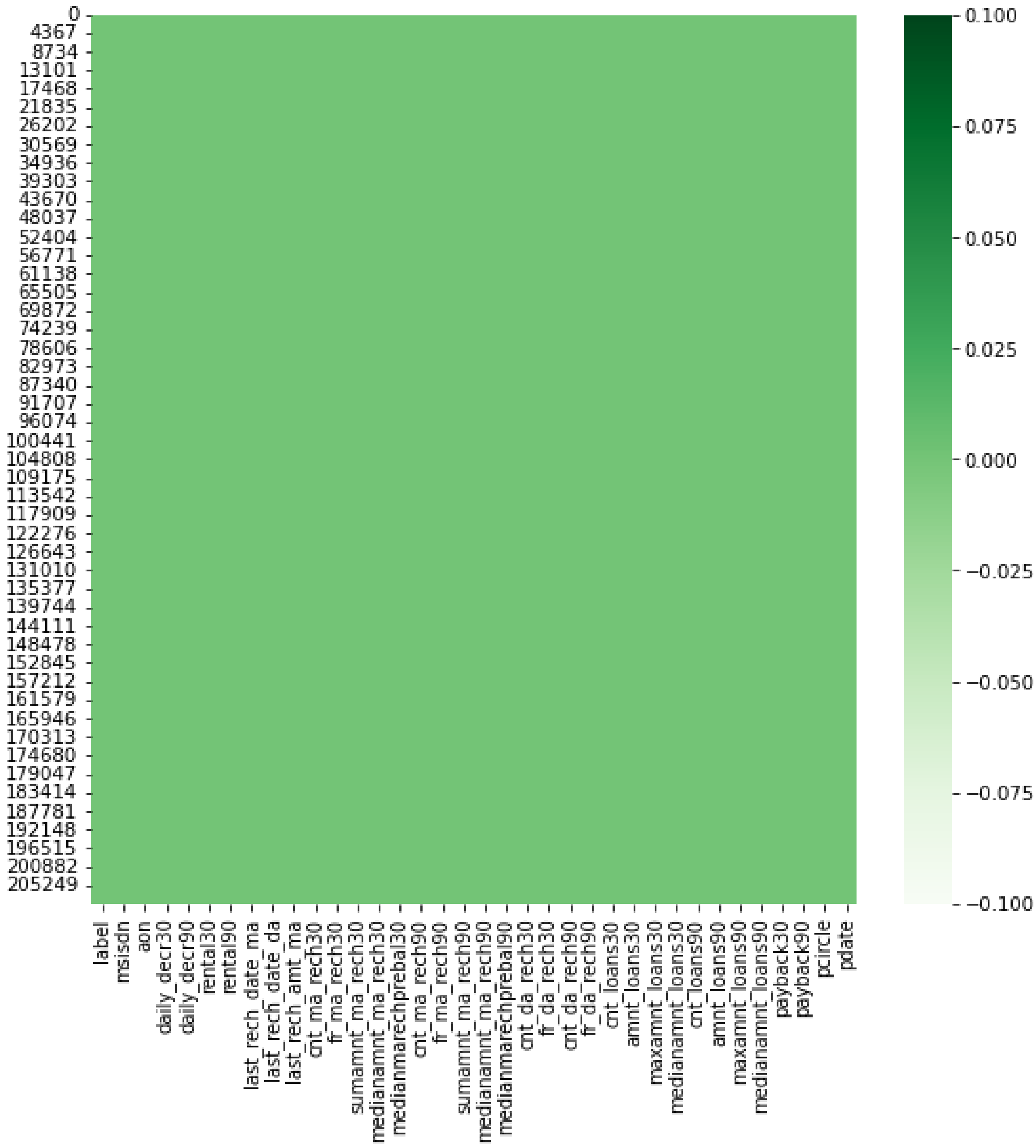
- ## Data Preprocessing Done

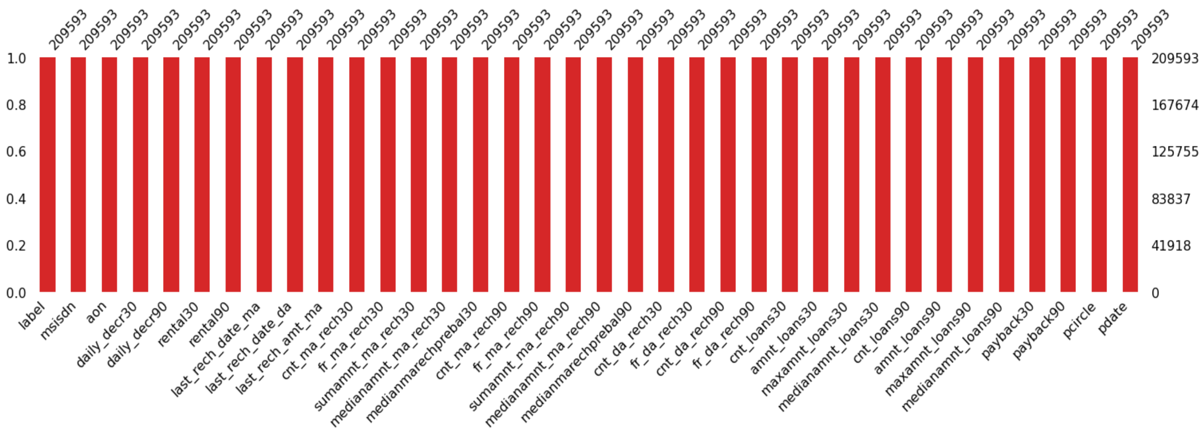
Checked for missing values to confirm the information of no null values present provided in the problem statement.

```
#count null values in all variables
df.isnull().sum()
```

```
label                    0
msisdn                   0
aon                      0
daily_decr30             0
daily_decr90             0
rental30                 0
rental90                 0
last_rech_date_ma        0
last_rech_date_da        0
last_rech_amt_ma         0
cnt_ma_rech30            0
fr_ma_rech30             0
sumamnt_ma_rech30        0
medianamnt_ma_rech30     0
medianmarechprebal30     0
cnt_ma_rech90            0
fr_ma_rech90             0
sumamnt_ma_rech90        0
medianamnt_ma_rech90     0
medianmarechprebal90     0
cnt_da_rech30            0
fr_da_rech30             0
cnt_da_rech90            0
fr_da_rech90             0
cnt_loans30              0
amnt_loans30             0
maxamnt_loans30          0
medianamnt_loans30       0
cnt_loans90              0
amnt_loans90             0
maxamnt_loans90          0
medianamnt_loans90       0
payback30                0
payback90                0
pcircle                  0
pdate                    0
dtype: int64
```

Took a visual on the missing data information as well.


0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
label
msisdn
aon
daily_decr30
daily_decr90
rental30
rental90
last_rech_date_ma
last_rech_date_da
last_rech_amt_ma
cnt_ma_rech30
fr_ma_rech30
sumamnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30
cnt_ma_rech90
fr_ma_rech90
sumamnt_ma_rech90
medianamnt_ma_rech90
medianmarechprebal90
cnt_da_rech30
fr_da_rech30
cnt_da_rech90
fr_da_rech90
cnt_loans30
amnt_loans30
maxamnt_loans30
medianamnt_loans30
cnt_loans90
amnt_loans90
maxamnt_loans90
medianamnt_loans90
payback30
payback90
pcircle
pdate


209593
1.0
0.8
0.6
0.4
0.2
0.0
167674
125755
83837
41918
0

Using the info method, we are able to confirm the non-null count details as well as the datatype information. We have 21 float/decimal datatype, 12 integer datatype and 3 object/categorical datatype columns. We will need to convert the object datatype columns to numerical data before we input the information in our machine learning models.

```
#some more information of dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209593 entries, 0 to 209592
Data columns (total 36 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   label                 209593 non-null  int64
 1   msisdn                209593 non-null  object
 2   aon                   209593 non-null  float64
 3   daily_decr30          209593 non-null  float64
 4   daily_decr90          209593 non-null  float64
 5   rental30              209593 non-null  float64
 6   rental90              209593 non-null  float64
 7   last_rech_date_ma     209593 non-null  float64
 8   last_rech_date_da     209593 non-null  float64
 9   last_rech_amt_ma      209593 non-null  int64
 10  cnt_ma_rech30         209593 non-null  int64
 11  fr_ma_rech30          209593 non-null  float64
 12  sumamnt_ma_rech30     209593 non-null  float64
 13  medianamnt_ma_rech30  209593 non-null  float64
 14  medianmarechprebal30  209593 non-null  float64
 15  cnt_ma_rech90         209593 non-null  int64
 16  fr_ma_rech90          209593 non-null  int64
 17  sumamnt_ma_rech90     209593 non-null  int64
 18  medianamnt_ma_rech90  209593 non-null  float64
 19  medianmarechprebal90  209593 non-null  float64
 20  cnt_da_rech30         209593 non-null  float64
 21  fr_da_rech30          209593 non-null  float64
 22  cnt_da_rech90         209593 non-null  int64
 23  fr_da_rech90          209593 non-null  int64
 24  cnt_loans30           209593 non-null  int64
 25  amnt_loans30          209593 non-null  int64
 26  maxamnt_loans30       209593 non-null  float64
 27  medianamnt_loans30    209593 non-null  float64
 28  cnt_loans90           209593 non-null  float64
 29  amnt_loans90          209593 non-null  int64
 30  maxamnt_loans90       209593 non-null  int64
 31  medianamnt_loans90    209593 non-null  float64
 32  payback30             209593 non-null  float64
 33  payback90             209593 non-null  float64
 34  pcircle               209593 non-null  object
 35  pdate                 209593 non-null  object
dtypes: float64(21), int64(12), object(3)
memory usage: 57.6+ MB
```

- Data Inputs- Logic- Output Relationships

Data description on each column present in our dataset.

label : Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan {1: success, 0: failure}

msisdn : Mobile number of users

aon : Age on cellular network in days

daily_decr30 : Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)

daily_decr90 : Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)

rental30 : Average main account balance over last 30 days

rental90 : Average main account balance over last 90 days

last_rech_date_ma : Number of days till last recharge of main account

last_rech_date_da : Number of days till last recharge of data account

last_rech_amt_ma : Amount of last recharge of main account (in Indonesian Rupiah)

cnt_ma_rech30 : Number of times main account got recharged in last 30 days

fr_ma_rech30 : Frequency of main account recharged in last 30 days

sumamnt_ma_rech30 : Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)

medianamnt_ma_rech30 : Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)

medianmarechprebal30 : Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)

cnt_ma_rech90 : Number of times main account got recharged in last 90 days

fr_ma_rech90 : Frequency of main account recharged in last 90 days

sumamnt_ma_rech90 : Total amount of recharge in main account over last 90 days (in Indonasian Rupiah) medianamnt_ma_rech90 : Median of amount of recharges done in main account over last 90 days at user level (in Indonasian Rupiah)

medianmarechprebal90 : Median of main account balance just before recharge in last 90 days at user level (in Indonasian Rupiah)

cnt_da_rech30 : Number of times data account got recharged in last 30 days

fr_da_rech30 : Frequency of data account recharged in last 30 days

cnt_da_rech90 : Number of times data account got recharged in last 90 days

fr_da_rech90 : Frequency of data account recharged in last 90 days

cnt_loans30 : Number of loans taken by user in last 30 days

amnt_loans30 : Total amount of loans taken by user in last 30 days

maxamnt_loans30 : Maximum amount of loan taken by the user in last 30 days

medianamnt_loans30: Median of amounts of loan taken by the user in last 30 days

cnt_loans90 : Number of loans taken by user in last 90 days

amnt_loans90 : Total amount of loans taken by user in last 90 days

maxamnt_loans90 : Maximum amount of loan taken by the user in last 90 days

medianamnt_loans90: Median of amounts of loan taken by the user in last 90 days

payback30 : Average payback time in days over last 30 days

payback90 : Average payback time in days over last 90 days

pcircle : Telecom circle

pdate : Date

Data description in a tabular format:

1	Variable	Definition	Comment
2	label	Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure}	
3	msisdn	mobile number of user	
4	aon	age on cellular network in days	
5	daily_decr30	Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)	
6	daily_decr90	Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)	
7	rental30	Average main account balance over last 30 days	Unsure of given definition
8	rental90	Average main account balance over last 90 days	Unsure of given definition
9	last_rech_date_ma	Number of days till last recharge of main account	
10	last_rech_date_da	Number of days till last recharge of data account	
11	last_rech_amt_ma	Amount of last recharge of main account (in Indonesian Rupiah)	
12	cnt_ma_rech30	Number of times main account got recharged in last 30 days	
13	fr_ma_rech30	Frequency of main account recharged in last 30 days	Unsure of given definition
14	sumamnt_ma_rech30	Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)	
15	medianamnt_ma_rech3(	Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)	
16	medianmarechprebal30	Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)	
17	cnt_ma_rech90	Number of times main account got recharged in last 90 days	
18	fr_ma_rech90	Frequency of main account recharged in last 90 days	Unsure of given definition
19	sumamnt_ma_rech90	Total amount of recharge in main account over last 90 days (in Indonasian Rupiah)	
20	medianamnt_ma_rech9(	Median of amount of recharges done in main account over last 90 days at user level (in Indonasian Rupiah)	
21	medianmarechprebal90	Median of main account balance just before recharge in last 90 days at user level (in Indonasian Rupiah)	
22	cnt_da_rech30	Number of times data account got recharged in last 30 days	
23	fr_da_rech30	Frequency of data account recharged in last 30 days	
24	cnt_da_rech90	Number of times data account got recharged in last 90 days	
25	fr_da_rech90	Frequency of data account recharged in last 90 days	
26	cnt_loans30	Number of loans taken by user in last 30 days	
27	amnt_loans30	Total amount of loans taken by user in last 30 days	
28	maxamnt_loans30	maximum amount of loan taken by the user in last 30 days	There are only two options:
28	maxamnt_loans30	maximum amount of loan taken by the user in last 30 days	There are only two options: 5 & 10 Rs., for which the user needs to p
29	medianamnt_loans30	Median of amounts of loan taken by the user in last 30 days	
30	cnt_loans90	Number of loans taken by user in last 90 days	
31	amnt_loans90	Total amount of loans taken by user in last 90 days	
32	maxamnt_loans90	maximum amount of loan taken by the user in last 90 days	
33	medianamnt_loans90	Median of amounts of loan taken by the user in last 90 days	
34	payback30	Average payback time in days over last 30 days	
35	payback90	Average payback time in days over last 90 days	
36	pcircle	telecom circle	
37	pdate	date	

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

I had made an assumption that any telecom company keeps the data of customer within 3 months so I have chopped off my data on basis of that.

I have dropped the 2016 year from pdate columns because the data is from the year 2016, only the date and months are different. We separated months and days to different columns.

Then I separately checked the defaulter’s data and found that many valuable users are defaulters as they might have forgotten to pay or they are having a busy life. I separated them so that company can deal politely, because we cannot lose these customers.

- ## Hardware and Software Requirements and Tools Used

Hardware technology being used.

RAM : 4 GB

CPU : AMD E2 with 2.10 GHz

Software technology being used.

Programming language : Python

Distribution : Anaconda Navigator

Browser based language shell : Jupyter Notebook

Libraries/Packages : Pandas , NumPy, matplotlib, seaborn, scikit-learn, missingno

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  We have used the describe method to check the numerical data details. There are 38 columns which have numerical values in them and it looks like the count, mean, standard deviation, minimum value, 25% quartile, 50% quartile, 75% quartile and maximum value are all mostly properly distributed in terms of data points but I do see some abnormality that we will confirm with a visual on it.

In [61]:

```
df.describe().T
```

Out[61]:

	count	mean	std	min	25%	50%	75%	max
label	209593.0	0.875177	0.330519	0.000000	1.000	1.000000	1.00	1.000000
aon	209593.0	8112.343445	75696.082531	-48.000000	246.000	527.000000	982.00	999860.755200
daily_decr30	209593.0	5381.402289	9220.623400	-93.012667	42.440	1469.175667	7244.00	265926.000000
daily_decr90	209593.0	6082.515068	10918.812767	-93.012667	42.692	1500.000000	7802.79	320630.000000
rental30	209593.0	2692.581910	4308.586781	-23737.140000	280.420	1083.570000	3356.94	198926.110000
rental90	209593.0	3483.406534	5770.461279	-24720.580000	300.260	1334.000000	4201.79	200148.110000
last_rech_date_ma	209593.0	3755.847800	53905.892230	-29.000000	1.000	3.000000	7.00	998650.377700
last_rech_date_da	209593.0	3712.202921	53374.833430	-29.000000	0.000	0.000000	0.00	999171.809400
last_rech_amt_ma	209593.0	2064.452797	2370.786034	0.000000	770.000	1539.000000	2309.00	55000.000000
cnt_ma_rech30	209593.0	3.978057	4.256090	0.000000	1.000	3.000000	5.00	203.000000
fr_ma_rech30	209593.0	3737.355121	53643.625172	0.000000	0.000	2.000000	6.00	999606.368100
sumamnt_ma_rech30	209593.0	7704.501157	10139.621714	0.000000	1540.000	4628.000000	10010.00	810096.000000
medianamnt_ma_rech30	209593.0	1812.817952	2070.864620	0.000000	770.000	1539.000000	1924.00	55000.000000
medianmarechprebal30	209593.0	3851.927942	54006.374433	-200.000000	11.000	33.900000	83.00	999479.419300
cnt_ma_rech90	209593.0	6.315430	7.193470	0.000000	2.000	4.000000	8.00	336.000000
fr_ma_rech90	209593.0	7.716780	12.590251	0.000000	0.000	2.000000	8.00	88.000000
sumamnt_ma_rech90	209593.0	12396.218352	16857.793882	0.000000	2317.000	7226.000000	16000.00	953036.000000
medianamnt_ma_rech90	209593.0	1864.595821	2081.680664	0.000000	773.000	1539.000000	1924.00	55000.000000
medianmarechprebal90	209593.0	92.025541	369.215658	-200.000000	14.600	36.000000	79.31	41456.500000
cnt_da_rech30	209593.0	262.578110	4183.897978	0.000000	0.000	0.000000	0.00	99914.441420
fr_da_rech30	209593.0	3749.494447	53885.414979	0.000000	0.000	0.000000	0.00	999809.240100
cnt_da_rech90	209593.0	0.041495	0.397556	0.000000	0.000	0.000000	0.00	38.000000
fr_da_rech90	209593.0	0.045712	0.951386	0.000000	0.000	0.000000	0.00	64.000000
cnt_loans30	209593.0	2.758981	2.554502	0.000000	1.000	2.000000	4.00	50.000000
amnt_loans30	209593.0	17.952021	17.379741	0.000000	6.000	12.000000	24.00	306.000000
maxamnt_loans30	209593.0	274.658747	4245.264648	0.000000	6.000	6.000000	6.00	99864.560860
medianamnt_loans30	209593.0	0.054029	0.218039	0.000000	0.000	0.000000	0.00	3.000000

cnt_da_rech30	209593.0	262.578110	4183.897978	0.000000	0.000	0.000000	0.00	99914.441420
fr_da_rech30	209593.0	3749.494447	53885.414979	0.000000	0.000	0.000000	0.00	999809.240100
cnt_da_rech90	209593.0	0.041495	0.397556	0.000000	0.000	0.000000	0.00	38.000000
fr_da_rech90	209593.0	0.045712	0.951386	0.000000	0.000	0.000000	0.00	64.000000
cnt_loans30	209593.0	2.758981	2.554502	0.000000	1.000	2.000000	4.00	50.000000
amnt_loans30	209593.0	17.952021	17.379741	0.000000	6.000	12.000000	24.00	306.000000
maxamnt_loans30	209593.0	274.658747	4245.264648	0.000000	6.000	6.000000	6.00	99864.560860
medianamnt_loans30	209593.0	0.054029	0.218039	0.000000	0.000	0.000000	0.00	3.000000
cnt_loans90	209593.0	18.520919	224.797423	0.000000	1.000	2.000000	5.00	4997.517944
amnt_loans90	209593.0	23.645398	26.469861	0.000000	6.000	12.000000	30.00	438.000000
maxamnt_loans90	209593.0	6.703134	2.103864	0.000000	6.000	6.000000	6.00	12.000000
medianamnt_loans90	209593.0	0.046077	0.200692	0.000000	0.000	0.000000	0.00	3.000000
payback30	209593.0	3.398826	8.813729	0.000000	0.000	0.000000	3.75	171.500000
payback90	209593.0	4.321485	10.308108	0.000000	0.000	1.666667	4.50	171.500000
Pay_Back_Day	209593.0	14.398940	8.438900	1.000000	7.000	14.000000	21.00	31.000000
Pay_Back_Month	209593.0	6.797321	0.741435	6.000000	6.000	7.000000	7.00	8.000000
Pay_Back_Year	209593.0	2016.000000	0.000000	2016.000000	2016.000	2016.000000	2016.00	2016.000000

In the above report we can see that the maximum value for columns aon, daily_decr30, daily_decr90, rental30, rental90, last_rech_date_ma, last_rech_date_da,fr_ma_rech30,sumamnt_ma_rech30,medianmarechprebal30, sumamnt_ma_rech90 and fr_da_rech30 have quite a high number than the other column values.

- ## Testing of Identified Approaches (Algorithms)

  Listing down all the 7 classification machine learning algorithms used for the training and testing.

```
lg=LogisticRegression()
dtc=DecisionTreeClassifier()
knc=KNeighborsClassifier()
rfc=RandomForestClassifier()
abc=AdaBoostClassifier()
gbc=GradientBoostingClassifier()
etc=ExtraTreesClassifier()
```

```
models=[lg,dtc,knc,rfc,abc,gbc,etc]
```

- ## Run and Evaluate selected models

  I created a Classification Model function incorporating the evaluation metrics so that we can get the required data for all the models.

  ExtraTreesClassifier is Best Model

```
def confusion_plot():
    print('accuracy_score:-',accuracy_score(pred,y_test))
    print(classification_report(pred,y_test))
    matrix=confusion_matrix(pred,y_test)
    sns.heatmap(matrix,annot=True,cmap='Oranges',square=True,fmt="d")
    plt.title('Predicted Values')
    plt.show()
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30,random_state=43)
etc.fit(x_train,y_train)
pred=etc.predict(x_test)
confusion_plot()
```

```
accuracy_score:- 0.9411679190252501
              precision    recall  f1-score   support

           0       0.93      0.95      0.94     53908
           1       0.95      0.93      0.94     56151

    accuracy                           0.94    110059
   macro avg       0.94      0.94      0.94    110059
weighted avg       0.94      0.94      0.94    110059
```

- ## Key Metrics for success in solving problem under consideration

  The key metrics used here were accuracy_score, cross_val_score, classification report, auc_score and confusion matrix. We tried to find out the best parameters and also to increase our scores by using Hyperparameter Tuning and we will be using GridSearchCV method.

  1. Cross Validation:

  Cross-validation helps to find out the over fitting and under fitting of the model. In the cross validation the model is made to run on different subsets of the dataset which will get multiple measures of the model. If we take 5 folds, the data will be divided into 5 pieces where each part being 20% of full dataset. While running the Cross-validation the 1st part (20%) of the 5 parts will be kept out as a holdout set for validation and everything else is used for training data. This way we will get the first estimate of the model quality of the dataset.

In the similar way further iterations are made for the second 20% of the dataset is held as a holdout set and remaining 4 parts are used for training data during process. This way we will get the second estimate of the model quality of the dataset. These steps are repeated during the cross-validation process to get the remaining estimate of the model quality.

Cross validation

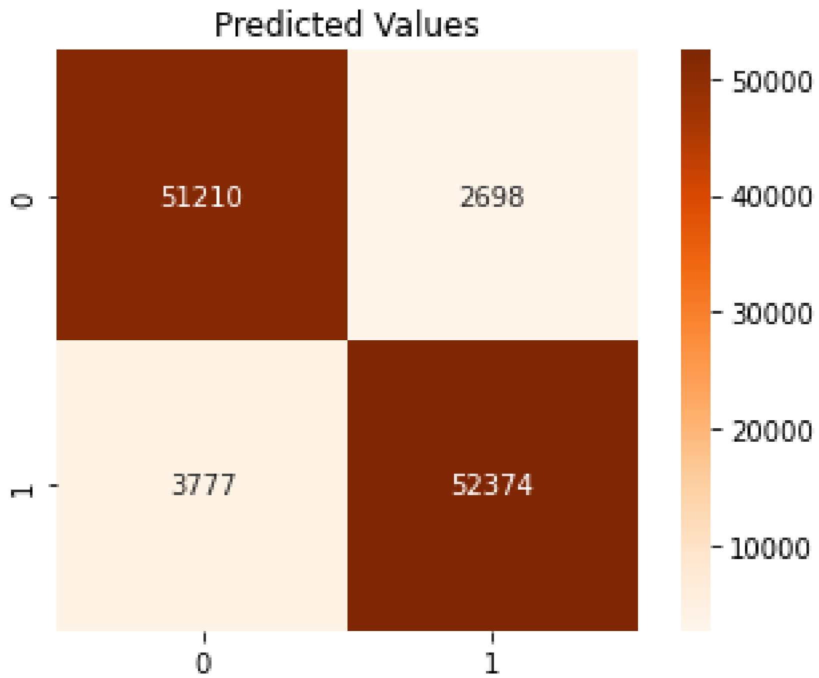
```
for model in models:
    print(model)
    score=cross_val_score(model,x,y,cv=5)
    print(score)
    print(score.mean())
    print('---------------------------------------------------------')
```

```
LogisticRegression()
[0.76848432 0.80695896 0.80485744 0.80410783 0.80702448]
0.7982866067620265
---------------------------------------------------------
DecisionTreeClassifier()
[0.79279844 0.91986153 0.91791146 0.91864744 0.92066456]
0.8939766868561024
---------------------------------------------------------
KNeighborsClassifier()
[0.86151582 0.90609625 0.9035872  0.90537262 0.90458213]
0.8962308029982434
---------------------------------------------------------
RandomForestClassifier()
[0.81243782 0.96427841 0.96444148 0.96367824 0.96430518]
0.9338282267997853
---------------------------------------------------------
AdaBoostClassifier()
[0.80036253 0.8772573  0.87416181 0.87499319 0.87786894]
0.8609287531487955
---------------------------------------------------------
GradientBoostingClassifier()
[0.82267319 0.90653238 0.90486834 0.90527722 0.90811209]
0.889492642409731
---------------------------------------------------------
ExtraTreesClassifier()
[0.84792771 0.95984899 0.95866271 0.95995748 0.95918061]
0.9371154997091462
---------------------------------------------------------
```

2. Confusion Matrix:

A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see whether the system is confusing two classes (i.e., commonly mislabelling one as another).

It is a special kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions (each combination of dimension and class is a variable in the contingency table).



3. Classification Report:

The classification report visualizer displays the precision, recall, F1, and support scores for the model. There are four ways to check if the predictions are right or wrong:

1. TN / True Negative: the case was negative and predicted negative

2. TP / True Positive: the case was positive and predicted positive
3. FN / False Negative: the case was positive but predicted negative
4. FP / False Positive: the case was negative but predicted positive

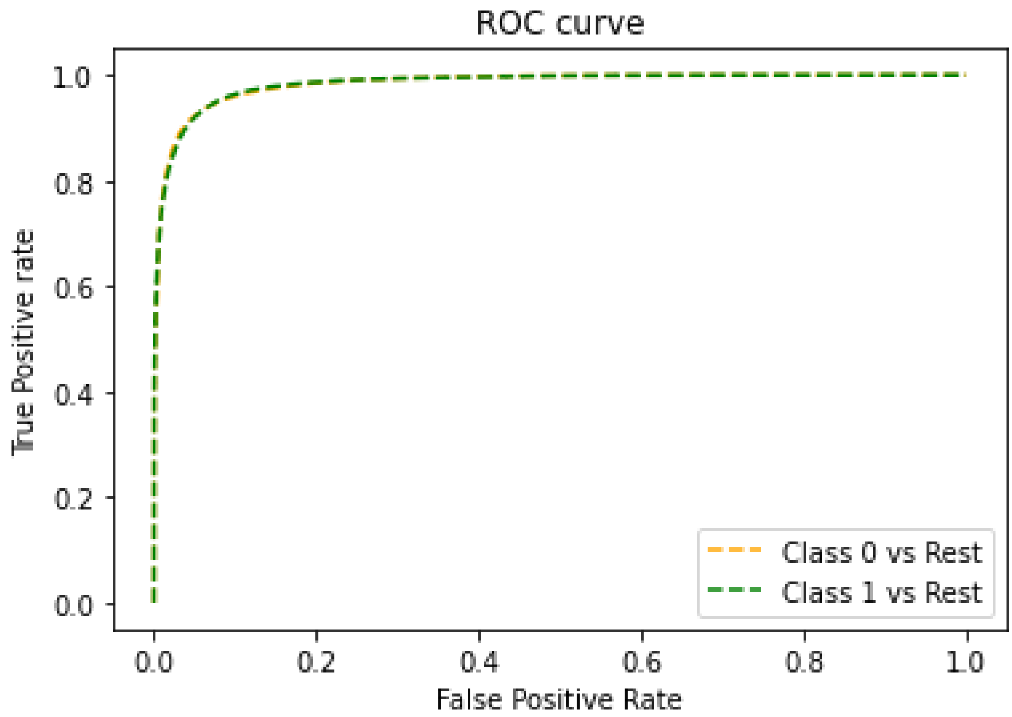
Precision: Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class, it is defined as the ratio of true positives to the sum of a true positive and false positive. It is the accuracy of positive predictions. The formula of precision is given below: *Precision = TP/ (TP + FP)*
Recall: Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives. It is also the fraction of positives that were correctly identified. The formula of recall is given below:
Recall = TP/(TP+FN)
F1 score: The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy. The formula is:
F1 Score = 2(Recall * Precision) / (Recall + Precision)*
Support: Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing. Support doesn't change between models but instead diagnoses the evaluation process.

Classification Report

	precision	recall	f1-score	support
0	0.93	0.95	0.94	53908
1	0.95	0.93	0.94	56151
accuracy			0.94	110059
macro avg	0.94	0.94	0.94	110059
weighted avg	0.94	0.94	0.94	110059

4. AUC-ROC Curve and score:

AUC (Area Under the Curve) - ROC (Receiver Operating Characteristics) curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represent the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s. By analogy, the Higher the AUC, the better the model is at distinguishing between patients with the disease and no disease.
The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.
Score is the area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.



5. Hyperparameter Tuning:

There is a list of different machine learning models. They all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named as Hyperparameters. These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for your model. You must select from a specific list of

hyperparameters for a given model as it varies from model to model.
We are not aware of optimal values for hyperparameters which would generate the best model output. So, what we tell the model is to explore and select the optimal model architecture automatically. This selection procedure for hyperparameter is known as Hyperparameter Tuning. We can do tuning by using GridSearchCV.
GridSearchCV is a function that comes in Scikit-learn (or SK-learn) model selection package. An important point here to note is that we need to have Scikit-learn library installed on the computer. This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters.

Hyper Parameter Tuning is the best Classification ML Model

```
parameter={
    'criterion':['gini','entropy'],
    'max_depth':[30,40],
    'n_estimators':[300,350],
    'min_samples_split':[3,4],
    'random_state':[42,72]
}
GSCV=GridSearchCV(ExtraTreesClassifier(),parameter,cv=5)
GSCV.fit(x_train,y_train)
GSCV.best_params_
```

```
final_model=ExtraTreesClassifier(criterion='entropy',max_depth=30,n_estimators=350,min_samples_split=3,random_state=72)
final_model.fit(x_train,y_train)
```

```
ExtraTreesClassifier(criterion='entropy', max_depth=30, min_samples_split=3,
                     n_estimators=350, random_state=72)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

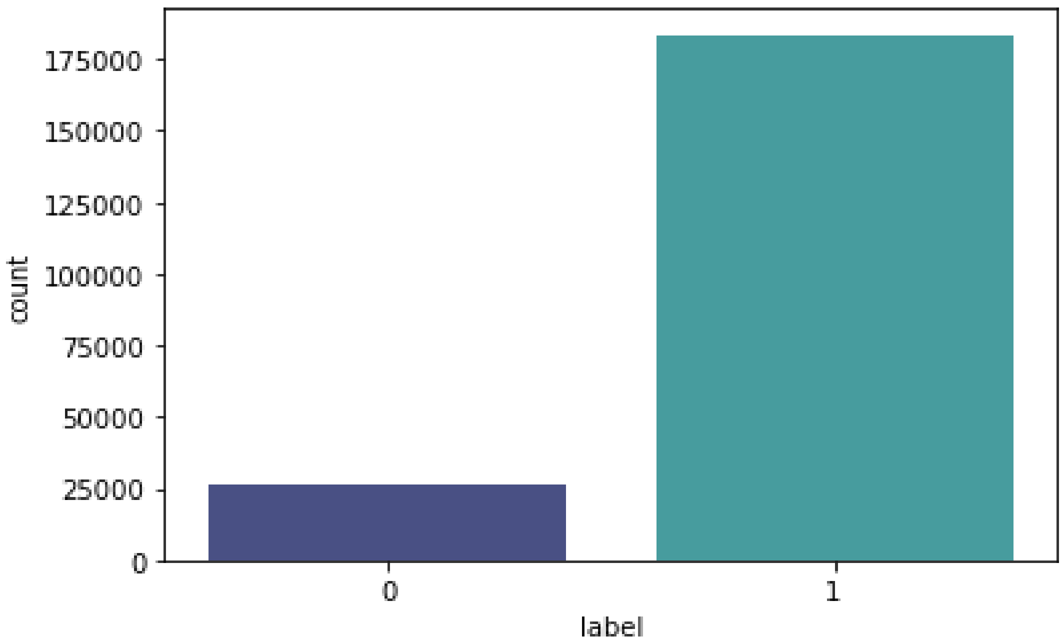
```
final_model_pred=final_model.predict(x_test)
print('Accuracy score for the best model is:',accuracy_score(final_model_pred,y_test)*100)
```

```
Accuracy score for the best model is: 93.6543126868316
```

- Visualizations

Now, we will see the different plots done with this dataset in order to know the insight of the data present. Below are the codes given for the plots and the output obtained:

Label Count Plot



- Interpretation of the Results

for feature aon:

Data ranges from -48 to 999860 with Mean value of 8112.34.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature daily_descr30:

Data ranges from -93 to 265926 with Mean value of 5381.4.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature daily_descr90:

Data ranges from -93 to 320630 with Mean value of 6082.52.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature rental30:

Data ranges from -23737.14 to 198926 with Mean value of 2692.58.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature rental90:

Data ranges from -24720 to 200148 with Mean value of 3483.41.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature last_rech_date_ma:

Data ranges from -29 to 998650 with Mean value of 3755.85.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature last_rech_date_da:

Data ranges from -29 to 999178 with Mean value of 3712.2.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature last_rech_amt_ma:

Data ranges from 0 to 55000 with Mean value of 2064.45.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature cnt_ma_rech30:

Data ranges from 0 to 203 with Mean value of 3.98.

Data is not distributed normally or in well curve.

Data is spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature fr_ma_rech30:

Data ranges from 0 to 999606 with Mean value of 3737.36.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature sumamnt_ma_rech30:

Data ranges from 0 to 810096 with Mean value of 7704.5.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature medianamnt_ma_rech30:

Data ranges from 0 to 55000 with Mean value of 1812.82.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature medianmarechprebal30:

Data ranges from -200 to 999479 with Mean value of 3851.93.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature cnt_ma_rech90:

Data ranges from 0 to 336 with Mean value of 6.32.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature fr_ma_rech90:

Data ranges from 0 to 88 with Mean value of 7.72.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature sumamnt_ma_rech90:

Data ranges from 0 to 953036 with Mean value of 12396.22.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature medianamnt_ma_rech90:

Data ranges from 0 to 55000 with Mean value of 1864.6.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature medianmarechprebal90:

Data ranges from -200 to 41456 with Mean value of 92.03.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature cnt_da_rech30:

Data ranges from 0 to 99914 with Mean value of 262.58.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature fr_da_rech30:

Data ranges from 0 to 999809 with Mean value of 3749.49.

Data is not distributed normally or in well curve.

Data is highly spreaded and needs to be treated accordingly.

Data is positively skewed and needs to be treated accordingly.

for feature cnt_da_rech90:

Data ranges from 0 to 38 with Mean value of 0.04.

Data is distributed normally but not in well curve.

Data is positively skewed and needs to be treated accordingly.

for feature fr_da_rech90:

Data ranges from 0 to 64 with Mean value of 0.05.

Data is not distributed normally or in well curve.

Data is positively skewed and needs to be treated accordingly.

for feature cnt_loans30:

Data ranges from 0 to 50 with Mean value of 2.76.

Data is not distributed normally or in well curve.

Data is positively skewed and needs to be treated accordingly.

for feature amnt_loans30:

Data ranges from 0 to 306 with Mean value of 17.95.

Data is not distributed normally or in well curve.

Data is positively skewed and needs to be treated accordingly.

for feature maxamnt_loans30:

Data ranges from 0 to 99864 with Mean value of 274.66.

Data is not distributed normally or in well curve.

Data is positively skewed and needs to be treated accordingly.

for feature medianamnt_loans30:

Data ranges from 0 to 3 with Mean value of 0.05.

Data is not distributed normally or in well curve and it is understandable as feature has only limited set of values.

Data is positively skewed and needs to be treated accordingly.

for feature cnt_loans90:

Data ranges from 0 to 4997.52 with Mean value of 18.52.

Data is not distributed normally or in well curve.

Data is positively skewed and needs to be treated accordingly.

for feature amnt_loans90:

Data ranges from 0 to 438 with Mean value of 23.65.

Data is not distributed normally or in well curve.

Data is positively skewed and needs to be treated accordingly.

for feature maxamnt_loans90:

Data ranges from 0 to 12 with Mean value of 6.7.

Data is not distributed normally or in well curve and it understandable as user has two option for loans i.e., 5 and 10 for with 6 and 12 has to be paid.

Data is positively skewed and needs to be treated accordingly.

for feature medianamnt_loans90:

Data ranges from 0 to 3 with Mean value of 0.05.

Data is not distributed normally or in well curve.

Data is positively skewed and needs to be treated accordingly.

for feature payback30:

Data ranges from 0 to 171.5 with Mean value of 3.4.

Data is not distributed normally or in well curve.

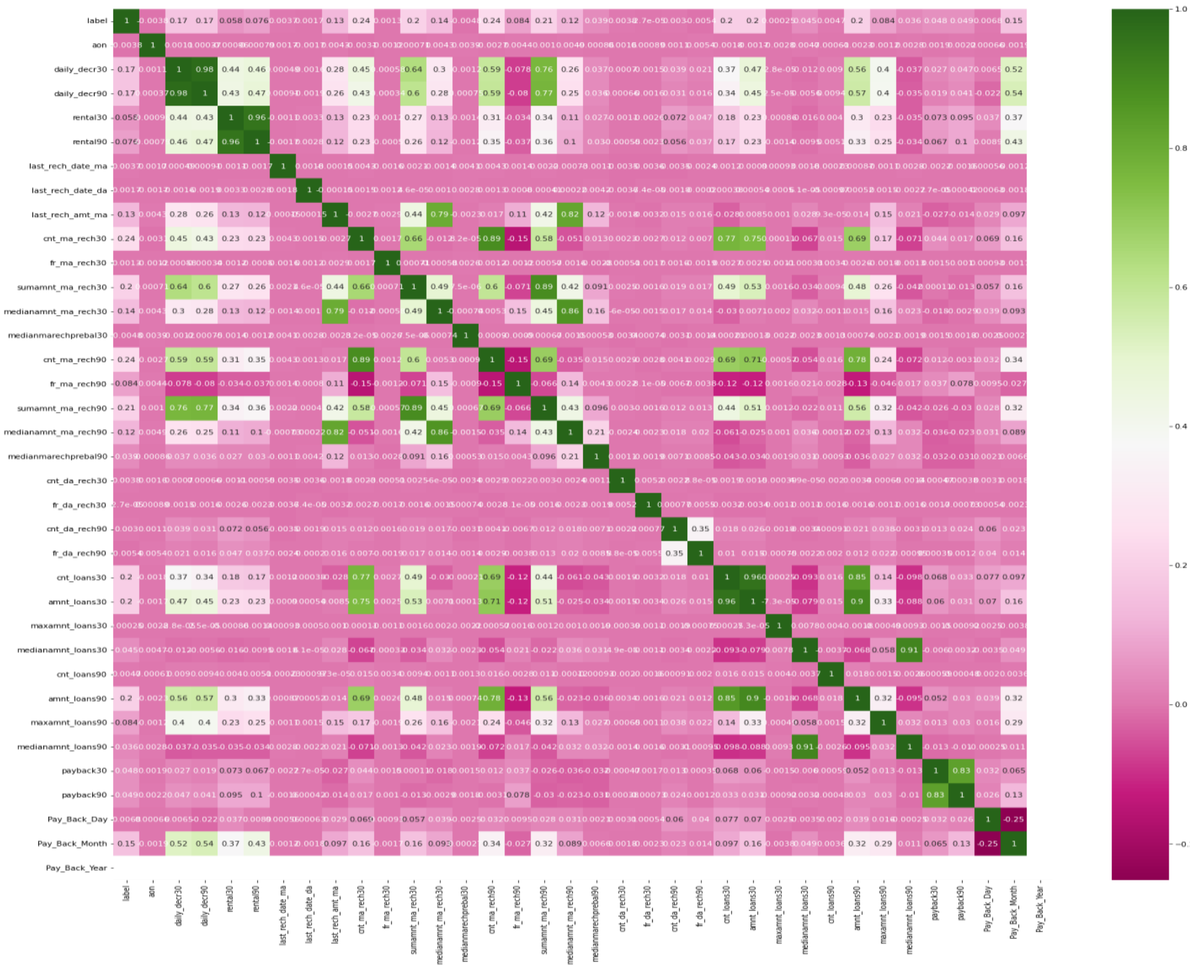
Data is positively skewed and needs to be treated accordingly.

for feature payback90:

Data ranges from 0 to 171.5 with Mean value of 4.32.

Data is not distributed normally or in well curve.

Data is positively skewed and needs to be treated accordingly.



CONCLUSION

- Key Findings and Conclusions of the Study

  From the final model MFI can find if a person will return money or not and should an MFI provide a load to that person or not judging from the various features taken into consideration.

- Learning Outcomes of the Study in respect of Data Science

  I built multiple classification models and did not rely on one single model for getting better accuracy and using cross validation comparison I ensured that the model does not fall into overfitting and underfitting issues. I picked the best one and performed hyper parameter tuning on it to enhance the scores.

- Limitations of this work and Scope for Future Work

Another limitation of this work is that it is based on a single dataset from a single financial institution, and the results may not generalize to other institutions or contexts. Additionally, the model is only as good as the data it is trained on, so if the data is incomplete or inaccurate, the predictions may not be reliable.

In terms of future work, one possibility would be to gather more data from other institutions and use this to improve the model's accuracy. Additionally, it would be interesting to explore the use of other machine learning techniques, such as neural networks, to see if they can improve the model's performance. Finally, it would be valuable to conduct more thorough testing and validation to ensure that the model is robust and reliable.

As you mentioned, this model can also be used by marketing companies to predict which customers are more likely to buy expensive services based on their personal details such as number of times account got recharged in last 30 days, daily amount spent from main account, averaged over last 30 days. Also, it can be used in other financial institutions and other industries such as insurance, banking to predict the potential default risk of their clients.