Micro Credit Defaulter

Submitted by:

Dattatraya Panda

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my SME (Subject Matter Expert) Mohd Kashifas well as Flip Robo Technologies who gave me the opportunity to do this project on Surprise Housing Price Prediction, which also helped me in doing lots of research where I came to know about so many

new things.

Also, I have utilized a few external resources that helped me to complete the project. I ensured that I learn from the samples and modify things according to my project requirement. All the external resources that were used in creating this project are listed below.

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/
6) https://medium.com/
7) https://towardsdatascience.com/
8) https://www.analyticsvidhya.com/

INTRODUCTION

Problem Statement:

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

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

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

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

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).

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.

Review of Literature

Microfinance, also called microcredit, is a type of banking service provided to unemployed or low-income individuals or groups who otherwise would have no other access to financial services.

While institutions participating in the area of microfinance most often provide lending—microloans can range from as small as $100 to as large as $25,000—many banks offer additional services such as checking and savings accounts as well as micro-insurance products, and some even provide financial and business education. The goal of microfinance is to ultimately give impoverished people an opportunity to become self-sufficient.

Motivation for the Problem Undertaken

Mathematical/ Analytical Modelling of the Problem

There are various analytics which I have done before moving forward with exploratory analysis, on the basis of accounts which got recharged in the last 30 days. I set the parameter that if the person is not recharging their main account within 3 months, I simply dropped their data because they are not valuable and they might be old customers, but there is no revenue rotating. Then I had checked the date columns and found that the data belongs to the year 2016. I extracted the month and day from the date, saved the data in separate columns, and tried to visualize the data on the basis of months and days. I had checked the maximum amount of loan taken by the people and found that the data had more outliers. As per the description given by the client, the loan amount can be paid by the customer is either rupiah 6 or 12 so that I have dropped all the loan amount that shows the loan is taken more than 12 rupiah. Then I separated the defaulter's data and checked the valuable customers in the network and we found that their monthly revenue is more than 10000 rupiah.

Although the data is quite imbalanced and many columns doesn't have that expected maximum value, we dropped that column. We checked the skewed data and try to treat the skewed data before model processing which caused NaN so avoided it. When we try removing the unwanted data, i.e., the outliers, we found that almost 40000+ data has been chopped. Though the data given by the client had almost 37 columns and over 2 lakhs since the data is expensive and we cannot lose more than 7-8% of the data as per company policy so avoided the outlier removal part as well. After scaling my data, I have sent the data to various classification models and found that Extra Trees Classifier Algorithm is working well.

Data Description:

	A	B
1	Variable	Definition
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
8	rental90	Average main account balance over last 90 days
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
14	sumamnt_ma_rech30	Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)
15	medianamnt_ma_rech30	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
19	sumamnt_ma_rech90	Total amount of recharge in main account over last 90 days (in Indonasian Rupiah)
20	medianamnt_ma_rech90	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
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
38		
39		

Hardware

RAM : 8.00 GB/512SSD

CPU : Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz

GPU : NVIDIA GeForce GTX 1650 Ti

Software technology used.

Programming language : Python

Distribution : Anaconda Navigator

Browser based language shell : Jupyter Notebook

Libraries/Packages specifically being used.

Pandas, NumPy, matplotlib, seaborn, scikit-learn, pandas-profiling,

missingno

Data Input

We import required language

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.simplefilter("ignore")
warnings.filterwarnings("ignore")
import joblib

import missingno as msno
import pandas_profiling
from imblearn.over_sampling import SMOTE
from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.neighbors import KNeighborsClassifier
import xgboost as xgb
import lightgbm as lgb


from sklearn import metrics
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import roc_curve, auc, roc_auc_score
```

```
]: df=pd.read_csv("Data file.csv")
```

```
]: df
```

]:

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	...	maxamnt_loans30	mec
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	...	6.0	
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	...	12.0	
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	...	6.0	
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	...	6.0	
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	...	6.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
209588	209589	1	22758I85348	404.0	151.872333	151.872333	1089.19	1089.19	1.0	0.0	...	6.0	
209589	209590	1	95583I84455	1075.0	36.936000	36.936000	1728.36	1728.36	4.0	0.0	...	6.0	
209590	209591	1	28556I85350	1013.0	11843.111667	11904.350000	5861.83	8893.20	3.0	0.0	...	12.0	
209591	209592	1	59712I82733	1732.0	12488.228333	12574.370000	411.83	984.58	2.0	38.0	...	12.0	
209592	209593	1	65061I85339	1581.0	4489.362000	4534.820000	483.92	631.20	13.0	0.0	...	12.0	

We load the csv file name as Data File

Exploratory Data Analysis

```
df.drop("Unnamed: 0", axis=1, inplace=True)
df.head()
```

	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_rec
0	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	1539	2	
1	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	5787	1	
2	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	1539	1	
3	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	947	0	
4	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	2309	7	

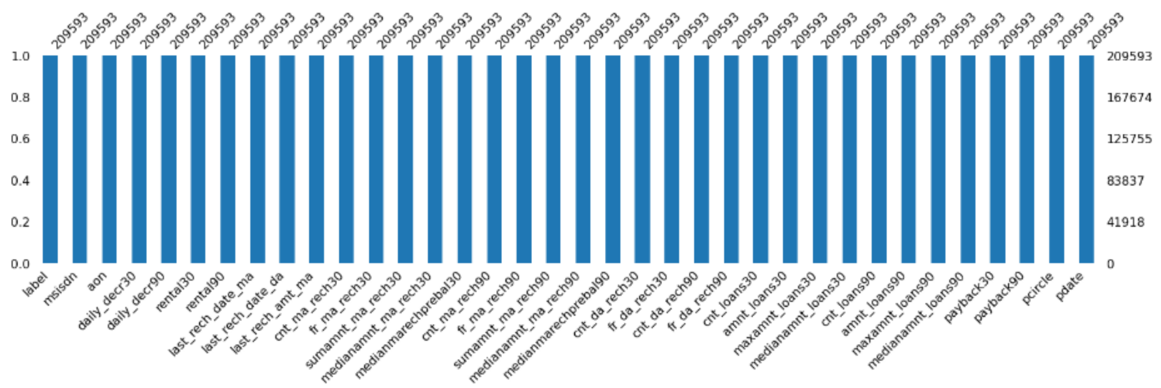
Removed the "Unnamed: 0" column from the dataset since it is just a numbering of rows and not useful for the prediction process.

```
In [10]: df.isna().sum()

Out[10]: 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
```

Using df.isna() we know about missing value and we clearly see that there no missing value present in our dataset

```
msno.bar(df, figsize = (25,5), color="tab:blue")
```

<AxesSubplot:>



```
]: df.info()
   df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 209592 entries, 0 to 209592
Data columns (total 36 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   label                 209592 non-null  int64
 1   msisdn                209592 non-null  object
 2   aon                   209592 non-null  float64
 3   daily_decr30          209592 non-null  float64
 4   daily_decr90          209592 non-null  float64
 5   rental30              209592 non-null  float64
 6   rental90              209592 non-null  float64
 7   last_rech_date_ma     209592 non-null  float64
 8   last_rech_date_da     209592 non-null  float64
 9   last_rech_amt_ma      209592 non-null  int64
 10  cnt_ma_rech30         209592 non-null  int64
 11  fr_ma_rech30          209592 non-null  float64
 12  sumamnt_ma_rech30     209592 non-null  float64
 13  medianamnt_ma_rech30  209592 non-null  float64
 14  medianmarechprebal30  209592 non-null  float64
 15  cnt_ma_rech90         209592 non-null  int64
 16  fr_ma_rech90          209592 non-null  int64
 17  sumamnt_ma_rech90     209592 non-null  int64
 18  medianamnt_ma_rech90  209592 non-null  float64
 19  medianmarechprebal90  209592 non-null  float64
 20  cnt_da_rech30         209592 non-null  float64
 21  fr_da_rech30          209592 non-null  float64
 22  cnt_da_rech90         209592 non-null  int64
 23  fr_da_rech90          209592 non-null  int64
 24  cnt_loans30           209592 non-null  int64
 25  amnt_loans30          209592 non-null  int64
 26  maxamnt_loans30       209592 non-null  float64
 27  medianamnt_loans30    209592 non-null  float64
 28  cnt_loans90           209592 non-null  float64
 29  amnt_loans90          209592 non-null  int64
 30  maxamnt_loans90       209592 non-null  int64
 31  medianamnt_loans90    209592 non-null  float64
 32  payback30             209592 non-null  float64
 33  payback90             209592 non-null  float64
 34  pcircle               209592 non-null  object
 35  pdate                 209592 non-null  object
dtypes: float64(21), int64(12), object(3)
memory usage: 59.2+ MB
```

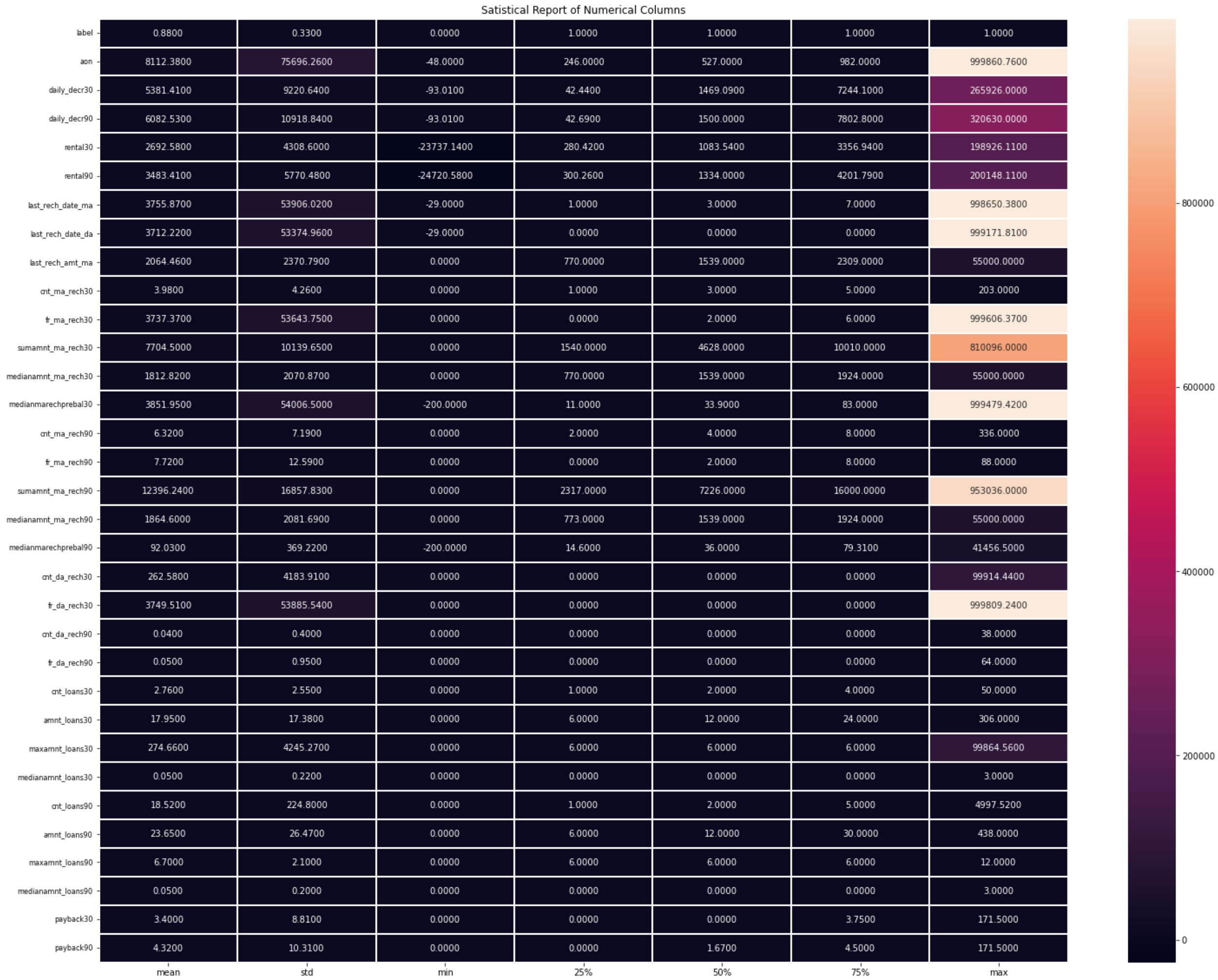
Using the info method we are able to confirm the non null count details as well as the datatype information. We have 21 float/decimal datatype, 12 integer datatype and 3 object/categorical datatype columns. We will need to convert the object datatype columns to numerical data before we input the information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

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

	count	mean	std	min	25%	50%	75%	max
label	209592.0	0.875177	0.330519	0.000000	1.000000	1.000000	1.0000	1.000000
aon	209592.0	8112.380399	75696.261220	-48.000000	246.000000	527.000000	982.0000	999860.755168
daily_decr30	209592.0	5381.412999	9220.644093	-93.012667	42.439500	1469.091833	7244.0960	265926.000000
daily_decr90	209592.0	6082.529123	10918.836919	-93.012667	42.691917	1500.000000	7802.7950	320630.000000
rental30	209592.0	2692.578912	4308.596841	-23737.140000	280.417500	1083.540000	3356.9450	198926.110000
rental90	209592.0	3483.407309	5770.475034	-24720.580000	300.260000	1334.000000	4201.7925	200148.110000
last_rech_date_ma	209592.0	3755.865715	53906.020204	-29.000000	1.000000	3.000000	7.0000	998650.377733
last_rech_date_da	209592.0	3712.220632	53374.960145	-29.000000	0.000000	0.000000	0.0000	999171.809410
last_rech_amt_ma	209592.0	2064.458973	2370.790003	0.000000	770.000000	1539.000000	2309.0000	55000.000000
cnt_ma_rech30	209592.0	3.978053	4.256099	0.000000	1.000000	3.000000	5.0000	203.000000
fr_ma_rech30	209592.0	3737.372947	53643.752523	0.000000	0.000000	2.000000	6.0000	999606.368132
sumamnt_ma_rech30	209592.0	7704.496570	10139.645685	0.000000	1540.000000	4628.000000	10010.0000	810096.000000
medianamnt_ma_rech30	209592.0	1812.819258	2070.869474	0.000000	770.000000	1539.000000	1924.0000	55000.000000
medianmarechprebal30	209592.0	3851.945862	54006.502647	-200.000000	11.000000	33.900000	83.0000	999479.419319
cnt_ma_rech90	209592.0	6.315437	7.193487	0.000000	2.000000	4.000000	8.0000	336.000000
fr_ma_rech90	209592.0	7.716812	12.590273	0.000000	0.000000	2.000000	8.0000	88.000000
sumamnt_ma_rech90	209592.0	12396.236149	16857.832129	0.000000	2317.000000	7226.000000	16000.0000	953036.000000
medianamnt_ma_rech90	209592.0	1864.597375	2081.685508	0.000000	773.000000	1539.000000	1924.0000	55000.000000
medianmarechprebal90	209592.0	92.025522	369.216539	-200.000000	14.600000	36.000000	79.3100	41456.500000
cnt_da_rech30	209592.0	262.579362	4183.907920	0.000000	0.000000	0.000000	0.0000	99914.441420
fr_da_rech30	209592.0	3749.512336	53885.542905	0.000000	0.000000	0.000000	0.0000	999809.240107
cnt_da_rech90	209592.0	0.041495	0.397557	0.000000	0.000000	0.000000	0.0000	38.000000
fr_da_rech90	209592.0	0.045713	0.951388	0.000000	0.000000	0.000000	0.0000	64.000000
cnt_loans30	209592.0	2.758975	2.554507	0.000000	1.000000	2.000000	4.0000	50.000000
amnt_loans30	209592.0	17.951992	17.379778	0.000000	6.000000	12.000000	24.0000	306.000000
maxamnt_loans30	209592.0	274.660029	4245.274734	0.000000	6.000000	6.000000	6.0000	99864.560864
medianamnt_loans30	209592.0	0.054029	0.218039	0.000000	0.000000	0.000000	0.0000	3.000000
cnt_loans90	209592.0	18.520988	224.797957	0.000000	1.000000	2.000000	5.0000	4997.517944
amnt_loans90	209592.0	23.645397	26.469924	0.000000	6.000000	12.000000	30.0000	438.000000
maxamnt_loans90	209592.0	6.703138	2.103869	0.000000	6.000000	6.000000	6.0000	12.000000
medianamnt_loans90	209592.0	0.046078	0.200692	0.000000	0.000000	0.000000	0.0000	3.000000
payback30	209592.0	3.398639	8.813330	0.000000	0.000000	0.000000	3.7500	171.500000
payback90	209592.0	4.321302	10.307791	0.000000	0.000000	1.666667	4.5000	171.500000

We have used the describe method to check the numerical data details. There are 33 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.

```python
# visualizing the statistical description of numeric datatype columns

plt.figure(figsize = (25,20))
sns.heatmap(round(df.describe()[1:].transpose(),2), linewidth = 2, annot= True, fmt = ".4f")
plt.title("Satistical Report of Numerical Columns")
plt.xticks(fontsize = 10)
plt.yticks(fontsize = 8)
plt.show()
```



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.

```
In [17]: df.nunique().sort_values()
```

```
Out[17]: pcircle                    1
         label                      2
         maxamnt_loans90            3
         medianamnt_loans90         6
         medianamnt_loans30         6
         cnt_da_rech90             27
         cnt_loans30               40
         fr_da_rech90              46
         amnt_loans30              48
         amnt_loans90              69
         last_rech_amt_ma          70
         cnt_ma_rech30             71
         pdate                     82
         fr_ma_rech90              89
         cnt_ma_rech90            110
         medianamnt_ma_rech30     510
         medianamnt_ma_rech90     608
         maxamnt_loans30         1050
         cnt_da_rech30           1066
         fr_da_rech30            1072
         fr_ma_rech30            1083
         cnt_loans90             1110
         last_rech_date_da       1174
         last_rech_date_ma       1186
         payback30               1363
         payback90               2381
         aon                     4507
         sumamnt_ma_rech30      15141
         medianmarechprebal90   29785
         medianmarechprebal30   30428
         sumamnt_ma_rech90      31771
         rental30              132148
         rental90              141033
         daily_decr30          147025
         daily_decr90          158669
         msisdn                186243
         dtype: int64
```

In the above list we can see that column pcircle has 1 single data value filled in all the records and therefore does not contribute much towards the output label prediction.

```
df.drop("pcircle", axis=1, inplace=True)
```

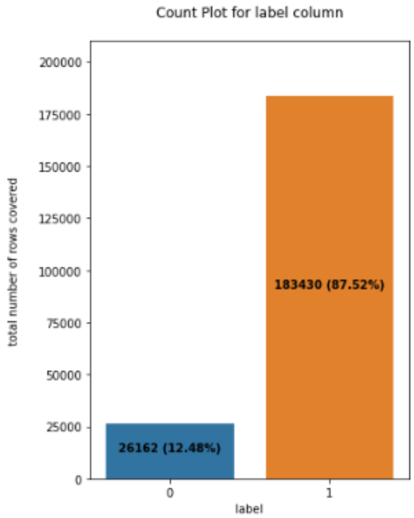
```
df.corr()
```

	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cnt_ma_
label	1.000000	-0.003785	0.168298	0.166151	0.058084	0.075521	0.003728	0.001711	0.131805	0
aon	-0.003785	1.000000	0.001104	0.000374	-0.000960	-0.000790	0.001692	-0.001693	0.004256	-0
daily_decr30	0.168298	0.001104	1.000000	0.977704	0.442066	0.458977	0.000487	-0.001636	0.275837	0
daily_decr90	0.166151	0.000374	0.977704	1.000000	0.434685	0.471730	0.000908	-0.001886	0.264130	0
rental30	0.058084	-0.000960	0.442066	0.434685	1.000000	0.955237	-0.001095	0.003262	0.127272	0
rental90	0.075521	-0.000790	0.458977	0.471730	0.955237	1.000000	-0.001688	0.002794	0.121416	0
last_rech_date_ma	0.003728	0.001692	0.000487	0.000908	-0.001095	-0.001688	1.000000	0.001790	-0.000147	0
last_rech_date_da	0.001711	-0.001693	-0.001636	-0.001886	0.003262	0.002794	0.001790	1.000000	-0.000149	0
last_rech_amt_ma	0.131805	0.004256	0.275837	0.264130	0.127272	0.121416	-0.000147	-0.000149	1.000000	-0
cnt_ma_rech30	0.237331	-0.003148	0.451385	0.426708	0.233343	0.230260	0.004311	0.001549	-0.002661	1
fr_ma_rech30	0.001330	-0.001163	-0.000577	-0.000343	-0.001219	-0.000503	-0.001629	0.001158	0.002876	0
sumamnt_ma_rech30	0.202828	0.000707	0.636536	0.603886	0.272649	0.259709	0.002105	0.000046	0.440821	0
medianamnt_ma_rech30	0.141491	0.004306	0.295356	0.282959	0.129853	0.120242	-0.001358	0.001037	0.794646	-0
medianmarechprebal30	-0.004829	0.003930	-0.001153	-0.000746	-0.001415	-0.001237	0.004071	0.002849	-0.002342	0
cnt_ma_rech90	0.236393	-0.002725	0.587338	0.593069	0.312118	0.345293	0.004263	0.001272	0.016706	0
fr_ma_rech90	0.084386	0.004401	-0.078300	-0.079530	-0.033529	-0.036524	0.001414	0.000798	0.106265	-0
sumamnt_ma_rech90	0.205794	0.001011	0.762981	0.768817	0.342306	0.360601	0.002243	-0.000414	0.418735	0
medianamnt_ma_rech90	0.120855	0.004909	0.257846	0.250518	0.110356	0.103151	-0.000726	0.000219	0.818735	-0
medianmarechprebal90	0.039300	-0.000859	0.037495	0.036382	0.027170	0.029547	-0.001086	0.004158	0.124646	0
cnt_da_rech30	0.003827	0.001564	0.000700	0.000660	-0.001105	-0.000548	-0.003467	-0.003628	-0.001837	0
fr_da_rech30	-0.000026	0.000892	-0.001500	-0.001570	-0.002558	-0.002345	-0.003626	-0.000074	-0.003230	-0
cnt_da_rech90	0.002999	0.001121	0.038814	0.031155	0.072255	0.056282	-0.003538	-0.001859	0.014779	0
fr_da_rech90	-0.005418	0.005395	0.020673	0.016437	0.046761	0.036886	-0.002395	-0.000203	0.016042	0
cnt_loans30	0.196283	-0.001826	0.366117	0.340387	0.180203	0.171595	0.001193	0.000380	-0.027611	0
amnt_loans30	0.197272	-0.001726	0.471492	0.447869	0.233453	0.231906	0.000903	0.000536	0.008503	0
maxamnt_loans30	0.000248	-0.002764	-0.000028	0.000025	-0.000864	-0.001411	0.000928	0.000503	0.001000	0
medianamnt_loans30	0.044590	0.004664	-0.011611	-0.005592	-0.016482	-0.009467	0.001835	0.000061	0.028370	-0
cnt_loans90	0.004733	-0.000611	0.008962	0.009446	0.004012	0.005141	-0.000225	-0.000972	0.000093	0
amnt_loans90	0.199788	-0.002319	0.563496	0.567204	0.298943	0.327436	0.000870	0.000519	0.014067	0
maxamnt_loans90	0.084144	-0.001191	0.400199	0.397251	0.234212	0.251029	-0.001123	0.001524	0.148459	0
medianamnt_loans90	0.035747	0.002771	-0.037305	-0.034686	-0.035489	-0.034122	0.002771	-0.002239	0.021004	-0
payback30	0.048330	0.001942	0.026922	0.019406	0.072974	0.067114	-0.002231	0.000079	-0.027358	0
payback90	0.049178	0.002205	0.047181	0.040806	0.095148	0.099505	-0.001582	0.000418	-0.014251	0

Checking the correlation data for our columns in our entire dataset.

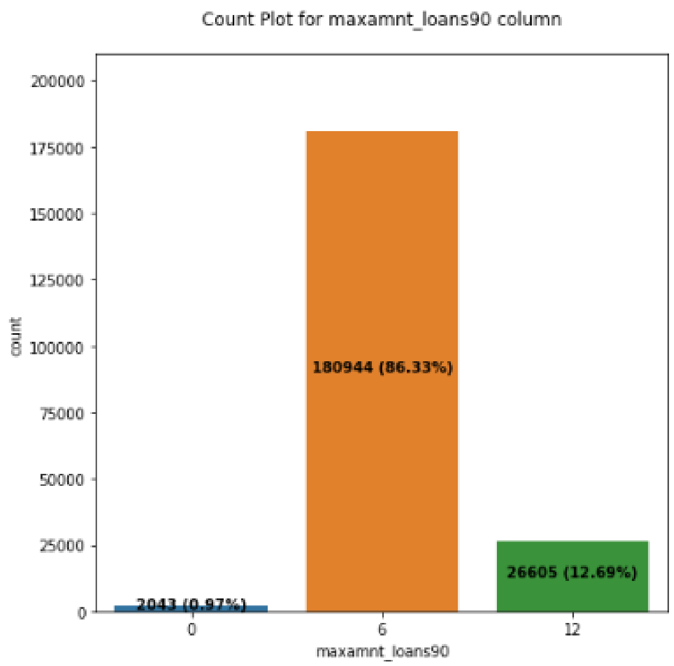
Visualization

```
try:
    x = 'label'
    k=0
    plt.figure(figsize=[5,7])
    axes = sns.countplot(df[x])
    for i in axes.patches:
        ht = i.get_height()
        mr = len(df[x])
        st = f"{ht} ({round(ht*100/mr,2)}%)"
        plt.text(k, ht/2, st, ha='center', fontweight='bold')
        k += 1
    plt.ylim(0,210000)
    plt.title(f'Count Plot for {x} column\n')
    plt.ylabel(f'total number of rows covered\n')
    plt.show()

except Exception as e:
    print("Error:", e)
    pass
```



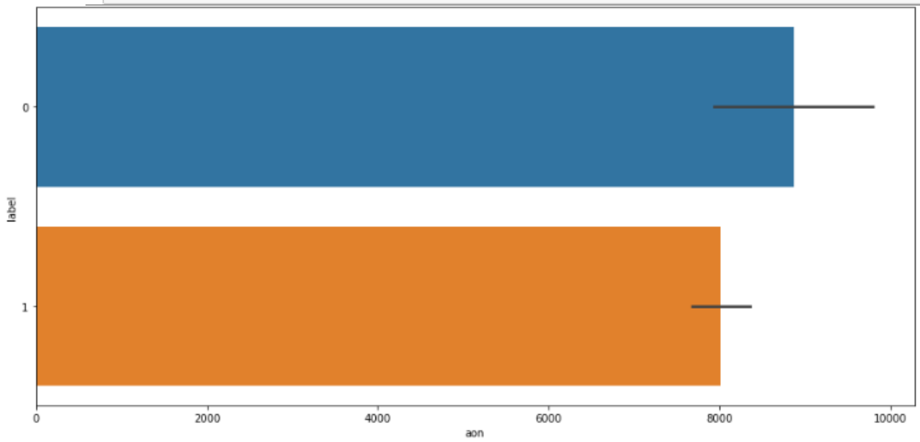
In the above count plot we can see that our label data is imbalanced which will need to balance before we feed information into our calssification machine learning models.

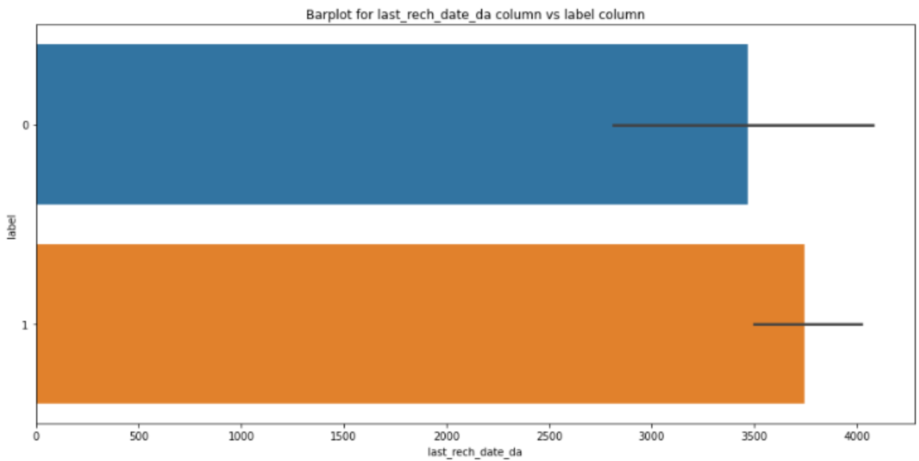
```
try:
    x = 'maxamnt_loans90'
    k=0
    plt.figure(figsize=[7,7])
    axes = sns.countplot(df[x])
    for i in axes.patches:
        ht = i.get_height()
        mr = len(df[x])
        st = f"{ht} ({round(ht*100/mr,2)}%)"
        plt.text(k, ht/2, st, ha='center', fontweight='bold')
        k += 1
    plt.ylim(0,210000)
    plt.title(f'Count Plot for {x} column\n')
    plt.show()

except Exception as e:
    print("Error:", e)
    pass
```



Bivariate Analysis

```
y = 'label'

x = 'aon'
plt.figure(figsize=[15,7])
sns.barplot(x,y,data=df,orient='h')
plt.title(f"Barplot for {x} column vs {y} column")
plt.show()

x = 'last_rech_date_da'
plt.figure(figsize=[15,7])
sns.barplot(x,y,data=df,orient='h')
plt.title(f"Barplot for {x} column vs {y} column")
plt.show()

x = 'last_rech_date_ma'
plt.figure(figsize=[15,7])
sns.barplot(x,y,data=df,orient='h')
plt.title(f"Barplot for {x} column vs {y} column")
plt.show()

x = 'last_rech_amt_ma'
plt.figure(figsize=[15,7])
sns.barplot(x,y,data=df,orient='h')
plt.title(f"Barplot for {x} column vs {y} column")
plt.show()
```

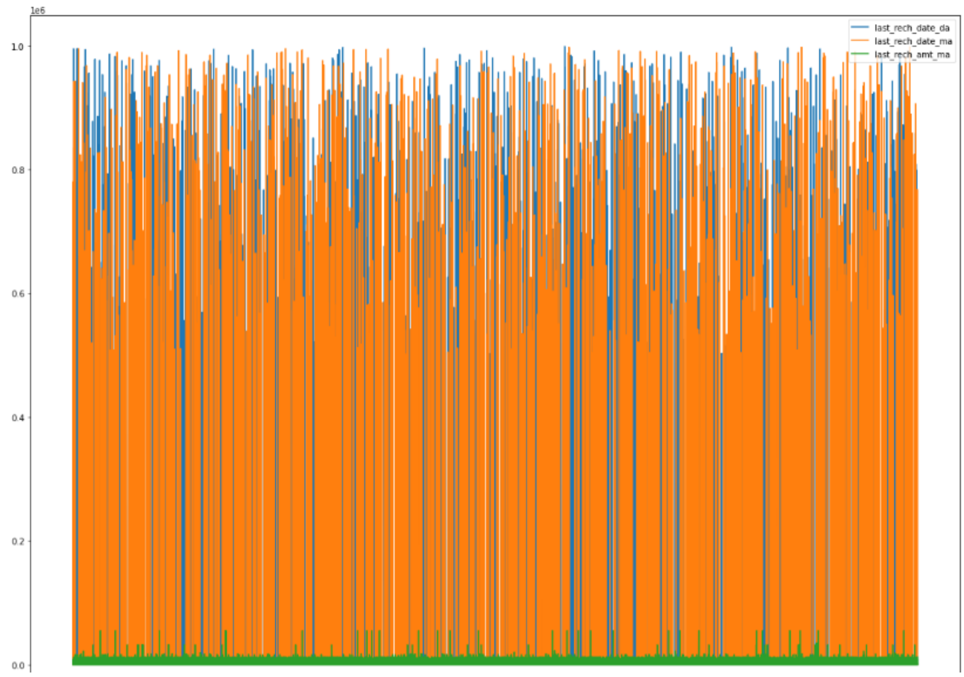


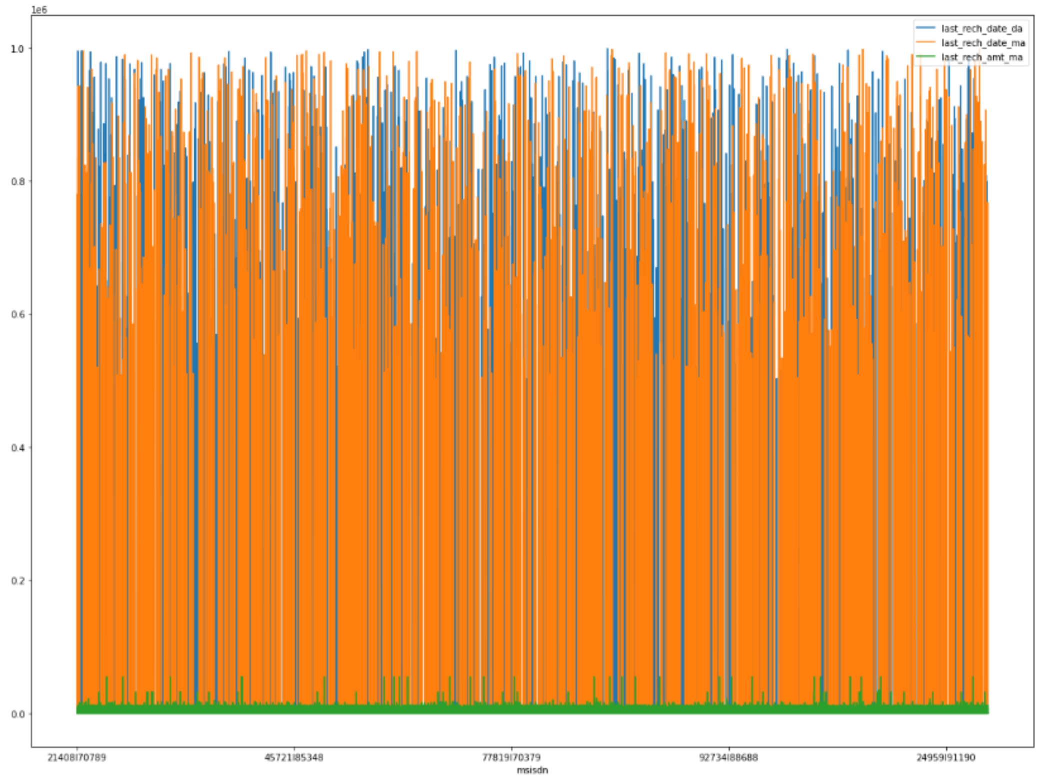


The above bar plots show the success and failure in returning the credit amount by a user depending on the specified feature columns.

```
df.plot(kind="line", x="pdate", y=["last_rech_date_da", "last_rech_date_ma", "last_rech_amt_ma"], figsize=[20,15])

df.plot(kind="line", x="msisdn", y=["last_rech_date_da", "last_rech_date_ma", "last_rech_amt_ma"], figsize=[20,15])
```

```
<AxesSubplot:xlabel='msisdn'>
```





Here we have line plots for date and mobile number data with respect to daily and monthly recharge information along with the amount factor.