MICRO CREDIT DEFAULTER PREDICTION

Submitted by:

SANTOSH H. HULBUTTI

Table of Contents

ACKNOWLEDGMENT

This project is completed using knowledge/information available on internet.

Following are the websites & YouTube Channels, which were used to understand concepts related to ML, AI & Data Visualization.

Websites:

1. towardsdatascience.com
2. medium.com
3. analyticsvidya.com
4. DataTrained LMS Platform
5. Official documentation of ScikitLearn, Matplot library, AutoViz, Sweet Viz, Pandas Library & Seaborn library.
6. Kaggle.com
7. UCI ML Repository
8. Stackoverflow.com
9. YouTube Channels:
    a. Krish Naik
    b. Sidhdhardan
    c. Keith Galli

I would like to thank FlipRobo Technologies, for giving an opportunity to work as an intern during this project period. And also like to thank mentor Ms. Gulshana Chaudhary for assigning the project.

INTRODUCTION

Business Problem Framing

This project includes the real time problem for Microfinance Institution (MFI) 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, MFI provides micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be defaulter if he deviates from the path of paying back the loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah), while, for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah).

Conceptual Background of the Domain Problem

Mobile financial services (MFS) are a very lucrative business as the returns are high but there is considerable risk of default involved. In our specific application, the telecom company in collaboration with a Microfinance Institute (MFI) provides loans of amount 5 and 10 (Indonesian Rupiah) for a very short period and the payback amount is 6 and 12 (Indonesian Rupiah) respectively which corresponds to a high interest rate of 20% in a very short period (usually 5 days). While the return is high, there is considerable risk of default involved, because the loan is being provided to low-income populations.

Therefore, it is necessary to classify all the defaulters to minimize business risk and avoid losses. The sample data is provided to us from our client database to classify defaulters which would help them in further investment and improvement in selection of customers.

We will use machine learning classification algorithms to predict the defaulters based on the sample data provided by the client.

Review of Literature

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

Indonesian Telecom company is 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 which is provided to us 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 used machine learning model to predict the above. Since we have categorical data so Classification model technique has been used.

We will begin our project with the sample dataset which contains loan default status along with associated features. We will look at all the features with following goals in mind:

- Relevance of the feature
- Distribution of the feature
- Cleaning the feature
- Visualization of the feature
- Visualization of the feature as per loan default status for data analysis

After having gone through all the features and cleaning the dataset, we will move on to machine learning classification modelling:

- Pre-processing the dataset for models
- Testing multiple algorithms with multiple evaluation metrics
- Select evaluation metric as per our specific business application
- Hyper-parameter tuning using GridSearchCV for the best model parameter
- And finally saving the best model

Motivation for the Problem Undertaken

The project was the first provided to me by Flip Robo Technologies as a part of the internship programme. The exposure to real world data and the opportunity to deploy my skillset in solving a real time problem has been the primary motivation. Further diving into the dataset, the motive is to help the poor or low-income band to have continuous access to their mobile accounts, and to make emergency calls even when they do not have account balance making use of the loan facility

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

The objective of the project is to prepare a model based on the sample dataset that classifies all loan defaulters and help our client in further investment and improvement in selection of customers.

ANALYTICAL PROBLEM FRAMING

Mathematical/ Analytical Modeling of the Problem

(Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.)

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	209593.0	NaN	NaN	NaN	104797.0	60504.431823	1.0	52399.0	104797.0	157195.0	209593.0
label	209593.0	NaN	NaN	NaN	0.875177	0.330519	0.0	1.0	1.0	1.0	1.0
msisdn	209593	186243	04581l85330	7	NaN	NaN	NaN	NaN	NaN	NaN	NaN
aon	209593.0	NaN	NaN	NaN	8112.343445	75696.082531	-48.0	246.0	527.0	982.0	999860.755168
daily_decr30	209593.0	NaN	NaN	NaN	5381.402289	9220.6234	-93.012667	42.44	1469.175667	7244.0	265926.0
daily_decr90	209593.0	NaN	NaN	NaN	6082.515068	10918.812767	-93.012667	42.692	1500.0	7802.79	320630.0
rental30	209593.0	NaN	NaN	NaN	2692.58191	4308.586781	-23737.14	280.42	1083.57	3356.94	198926.11
rental90	209593.0	NaN	NaN	NaN	3483.406534	5770.461279	-24720.58	300.26	1334.0	4201.79	200148.11
last_rech_date_ma	209593.0	NaN	NaN	NaN	3755.8478	53905.89223	-29.0	1.0	3.0	7.0	998650.377733
last_rech_date_da	209593.0	NaN	NaN	NaN	3712.202921	53374.83343	-29.0	0.0	0.0	0.0	999171.80941
last_rech_amt_ma	209593.0	NaN	NaN	NaN	2064.452797	2370.786034	0.0	770.0	1539.0	2309.0	55000.0
cnt_ma_rech30	209593.0	NaN	NaN	NaN	3.978057	4.25609	0.0	1.0	3.0	5.0	203.0
fr_ma_rech30	209593.0	NaN	NaN	NaN	3737.355121	53643.625172	0.0	0.0	2.0	6.0	999606.368132
sumamnt_ma_rech30	209593.0	NaN	NaN	NaN	7704.501157	10139.621714	0.0	1540.0	4628.0	10010.0	810096.0
medianamnt_ma_rech30	209593.0	NaN	NaN	NaN	1812.817952	2070.86462	0.0	770.0	1539.0	1924.0	55000.0
medianmarechprebal30	209593.0	NaN	NaN	NaN	3851.927942	54006.374433	-200.0	11.0	33.9	83.0	999479.419319
cnt_ma_rech90	209593.0	NaN	NaN	NaN	6.31543	7.19347	0.0	2.0	4.0	8.0	336.0
fr_ma_rech90	209593.0	NaN	NaN	NaN	7.71678	12.590251	0.0	0.0	2.0	8.0	88.0
sumamnt_ma_rech90	209593.0	NaN	NaN	NaN	12396.218352	16857.793882	0.0	2317.0	7226.0	16000.0	953036.0
medianamnt_ma_rech90	209593.0	NaN	NaN	NaN	1864.595821	2081.680664	0.0	773.0	1539.0	1924.0	55000.0
medianmarechprebal90	209593.0	NaN	NaN	NaN	92.025541	369.215658	-200.0	14.6	36.0	79.31	41456.5
cnt_da_rech30	209593.0	NaN	NaN	NaN	262.57811	4183.897978	0.0	0.0	0.0	0.0	99914.44142
fr_da_rech30	209593.0	NaN	NaN	NaN	3749.494447	53885.414979	0.0	0.0	0.0	0.0	999809.240107
cnt_da_rech90	209593.0	NaN	NaN	NaN	0.041495	0.397556	0.0	0.0	0.0	0.0	38.0
fr_da_rech90	209593.0	NaN	NaN	NaN	0.045712	0.951386	0.0	0.0	0.0	0.0	64.0
cnt_loans30	209593.0	NaN	NaN	NaN	2.758981	2.554502	0.0	1.0	2.0	4.0	50.0
amnt_loans30	209593.0	NaN	NaN	NaN	17.952021	17.379741	0.0	6.0	12.0	24.0	306.0
maxamnt_loans30	209593.0	NaN	NaN	NaN	274.658747	4245.264648	0.0	6.0	6.0	6.0	99864.560864
medianamnt_loans30	209593.0	NaN	NaN	NaN	0.054029	0.218039	0.0	0.0	0.0	0.0	3.0
cnt_loans90	209593.0	NaN	NaN	NaN	18.520919	224.797423	0.0	1.0	2.0	5.0	4997.517944
amnt_loans90	209593.0	NaN	NaN	NaN	23.645398	26.469861	0.0	6.0	12.0	30.0	438.0
maxamnt_loans90	209593.0	NaN	NaN	NaN	6.703134	2.103864	0.0	6.0	6.0	6.0	12.0
medianamnt_loans90	209593.0	NaN	NaN	NaN	0.046077	0.200692	0.0	0.0	0.0	0.0	3.0
payback30	209593.0	NaN	NaN	NaN	3.398826	8.813729	0.0	0.0	0.0	3.75	171.5
payback90	209593.0	NaN	NaN	NaN	4.321485	10.308108	0.0	0.0	1.666667	4.5	171.5
pcircle	209593	1	UPW	209593	NaN	NaN	NaN	NaN	NaN	NaN	NaN
pdate	209593	82	2016-07-04	3150	NaN	NaN	NaN	NaN	NaN	NaN	NaN

- There was total **209593** rows of observations.
- 3 features namely, **'Unnamed: 0', 'msisdn','pdate', 'pcircle'** were removed as they carry no value for predicting defaulter/non-Defaulter.
- Statistical techniques used:
    - Skewness check using '**.skew()'** method & removing using power transformation method,
    - Outliers' removal using '**Z-Score'** method (3 Std deviation method),
    - Correlation check using '**.corr()'** & heatmap method,
    - Minimizing Multi collinearity using '**Variance Inflation Factor(VIF)',**

- Scaling input data using **'StandardScaler()'** method**,**
- Removing data imbalance using '**SMOTE**' method,
- Graphical modelling done through seaborn, matplotlib, Tableau, sweetviz & Autoviz.

- After Pre processing we used '**.describe()**' method to check description of the data.
- Machine Learning algorithms used:

```
#Classification models
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier,
                             BaggingClassifier, ExtraTreesClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
```

- Model Evaluation metrics used:

```
#Evaluation metrics
from sklearn.metrics import confusion_matrix, classification_report,
                            f1_score, roc_auc_score, accuracy_score,
                            roc_curve, auc
```

- The final model was tuned using Hyper parameter tuning & validated using Cross validation score.

Data Sources and their formats

- The input data was shared in **CSV** format.
- There were 37 attributes (**36 features and 1 target**).
- The target variable is either 1 or 0 which means non defaulter and defaulter respectively.
- The other key attributes are the account balances, days since last recharge, age on network, average, median balance, recharge, frequency for 30 and 90-days moving average.

Label	Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan {1: non-defaulter, 0: Defaulter}
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 Indonesian Rupee)
medianamnt_ma_rech90	Median of Amount of recharges done in main account over last 90 days at user level (in Indonesian Rupee)
medianmarechprebal90	Median of main account balance just before recharge in last 90 days at user level (in Indonesian Rupee)
cnt_da_rech30	Number of times data account got recharged in last 30 days
fr_da_rech30	Frequency of data account recharged in last 30 days
cnt_da_rech90	Number of times data account got recharged in last 90 days
fr_da_rech90	Frequency of data account recharged in last 90 days
cnt_loans30	Number of loans taken by user in last 30 days
amnt_loans30	Total amount of loans taken by user in last 30 days
maxamnt_loans30	maximum amount of loan taken by the user in last 30 days
medianamnt_loans30	Median of amounts of loan taken by the user in last 30 days
cnt_loans90	Number of loans taken by user in last 90 days
amnt_loans90	Total amount of loans taken by user in last 90 days
maxamnt_loans90	maximum amount of loan taken by the user in last 90 days
medianamnt_loans90	Median of amounts of loan taken by the user in last 90 days
payback30	Average payback time in days over last 30 days
payback90	Average payback time in days over last 90 days
Pcircle	telecom circle
Pdate	Date

Data Pre-processing Done

1. Data Imported using Pandas '**.read_csv()'** method,

```
# Data import
data_url = "Data_file.csv"
data0 = pd.read_csv(data_url)
```

2. Total Rows of observation read:

```
#Total no. of observations
input_data_size = data0.shape[0]
input_data_size
```

```
209593
```

3. Dropping unnecessary columns/features,

```
data0.drop(columns = ['Unnamed: 0', 'msisdn','pdate'], inplace=True)
```

```
data0.drop(columns = ['pcircle'], inplace=True)
```

4. Dropping Duplicate entries,

```
data0.duplicated().sum()
```

```
386
```

```
data0.drop_duplicates(inplace=True)
```

5. Checking for data consistency & unusual data entries,

```
data0.isin([' ','NA','-','?','na']).sum().any()
```

```
False
```

6. Checking for unique entries, null values,

	unique_entries	missing values
pcircle	1	0
label	2	0
maxamnt_loans90	3	0
medianamnt_loans30	6	0
medianamnt_loans90	6	0
cnt_da_rech90	27	0
cnt_loans30	40	0
fr_da_rech90	46	0
amnt_loans30	48	0
amnt_loans90	69	0
last_rech_amt_ma	70	0
cnt_ma_rech30	71	0
fr_ma_rech90	89	0
cnt_ma_rech90	110	0
medianamnt_ma_rech30	510	0
medianamnt_ma_rech90	608	0
maxamnt_loans30	1050	0

	unique_entries	missing values
cnt_da_rech30	1066	0
fr_da_rech30	1072	0
fr_ma_rech30	1083	0
cnt_loans90	1110	0
last_rech_date_da	1174	0
last_rech_date_ma	1186	0
payback30	1363	0
payback90	2381	0
aon	4507	0
sumamnt_ma_rech30	15141	0
medianmarechprebal90	29785	0
medianmarechprebal30	30428	0
sumamnt_ma_rech90	31771	0
rental30	132148	0
rental90	141033	0
daily_decr30	147025	0
daily_decr90	158669	0

7. Checking for datatype count,

```
unique_null_data['Dtypes'].value_counts()
```

```
float64    21
int64      12
object      1
Name: Dtypes, dtype: int64
```

- Age on Cellular network has minimum & maximum values as -48 & 9999860 resply. which is not realistic. we will remove these.

```
data0['aon'].min()
```

```
-48.0
```

```
data0['aon'].max()
```

```
999860.755167902
```

Selecting age threshold to be 65 years.. which translates to 23740 days(including 15 extra days of leap year)

```
to_remove1 = np.where(data0['aon'] >=23740)
len(to_remove1[0])
```

```
2089
```

```
len(np.where(data0['aon'] <=0))
```

```
1
```

Also removing new customers who are yet to complete 60 days using the network.

```
to_remove2 = np.where(data0['aon'] <60)
len(to_remove2[0])
```

```
2503
```

8. *There were 2089 observations which were having unrealistic entry of age on network (more than 65 years) & 2503 observations which were having age less than 60 days, as there were some columns which were having 30 & 90-day simple moving period average having the consisting data with moving average entries was better. also new customers cannot be assessed with no usage history. so, it was considered to create model based on the people who had at least 60 days of usage history & less than 65 years using cellular network.* Total data loss after removing observations based on the age on network was 2.375%.

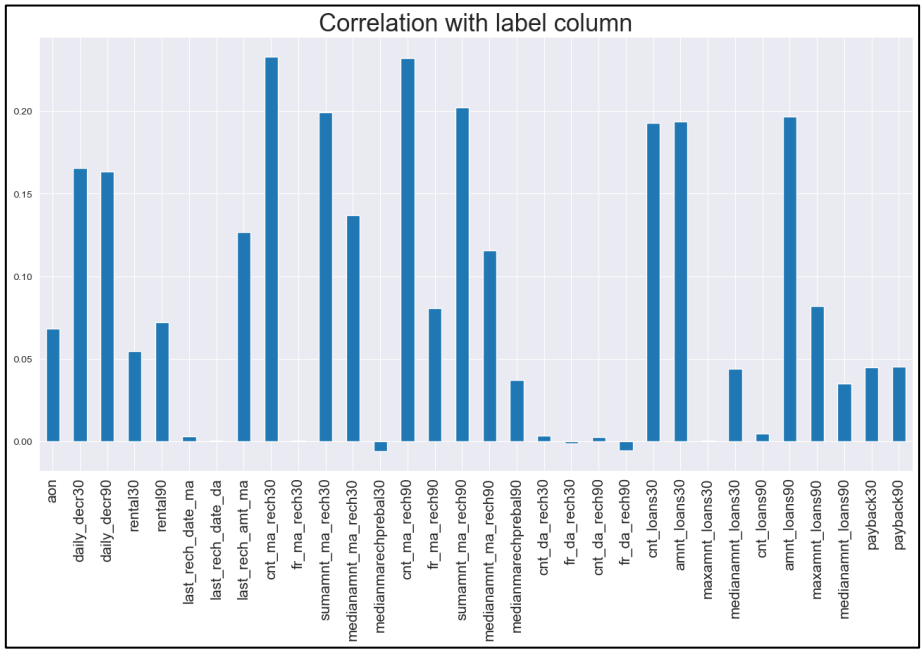
Total data loss after removing the data

```
(input_data_size-data1.shape[0])*100/input_data_size
```

```
2.3750793203971505
```

9. Following features are removed as based on Correlation using seaborn heatmap and .corr() methos & Multicollinearity check using VIF value:

   'amnt_loans30','daily_decr90','cnt_ma_rech90','sumamnt_ma_rech30','rental30','cnt_loans30', 'last_rech_date_ma', 'last_rech_date_da', 'fr_ma_rech30', 'cnt_da_rech30', 'fr_da_rech30', 'cnt_da_rech90', & 'maxamnt_loans30'.



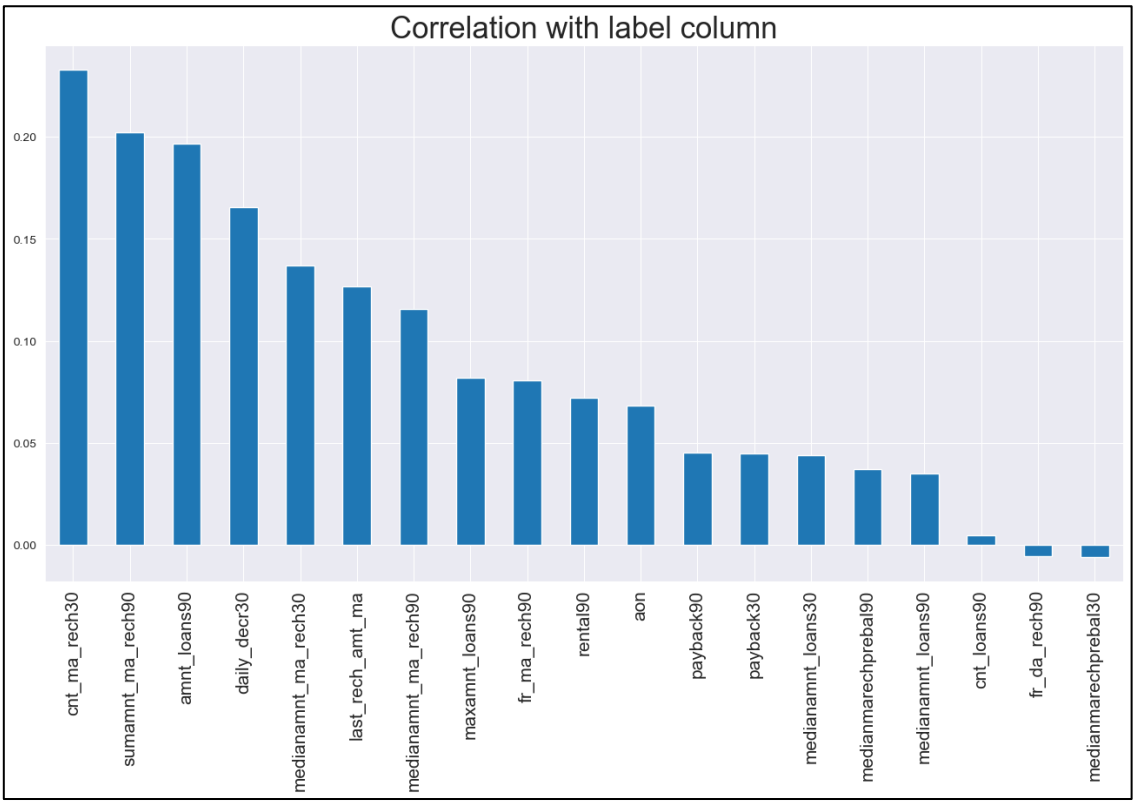
	variables	VIF
23	amnt_loans30	46.829277
2	daily_decr90	41.863041
1	daily_decr30	38.919511
22	cnt_loans30	35.601270
13	cnt_ma_rech90	28.819243
8	cnt_ma_rech30	28.133151
15	sumamnt_ma_rech90	23.333891
10	sumamnt_ma_rech30	20.086702
27	amnt_loans90	19.493348
4	rental90	18.813280
3	rental30	18.244871
16	medianamnt_ma_rech90	10.069494
11	medianamnt_ma_rech30	8.945648
25	medianamnt_loans30	6.306235
29	medianamnt_loans90	6.247993
7	last_rech_amt_ma	6.020194

	variables	VIF
28	maxamnt_loans90	5.648205
31	payback90	3.865342
30	payback30	3.725545
0	aon	2.679163
14	fr_ma_rech90	1.447735
20	cnt_da_rech90	1.158917
21	fr_da_rech90	1.141931
17	medianmarechprebal90	1.131575
26	cnt_loans90	1.007108
12	medianmarechprebal30	1.004849
6	last_rech_date_da	1.004827
5	last_rech_date_ma	1.004764
9	fr_ma_rech30	1.004748
19	fr_da_rech30	1.004748
24	maxamnt_loans30	1.004196
18	cnt_da_rech30	1.003923

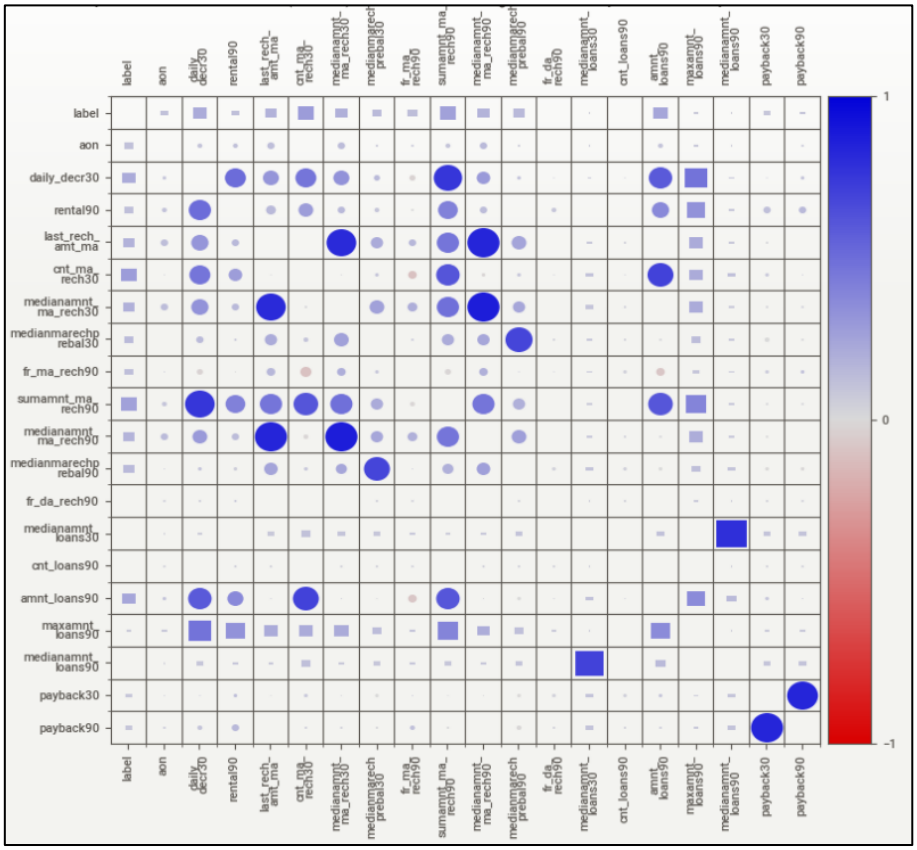
10. Skewness is removed using Yeo-Johnson Power Transformer.
11. Outliers are removed using Z-score method. Data loss observed was 1.13%.
12. Standard scaling is applied on the entire train & test data.
13. We used train_test_split to split data for machine learning.

Data Inputs- Logic- Output Relationships

Following heat map shows the relation between numerical features and target variable, using correlation coefficient.



Following heat map shows the relation between independent numerical features with each other & target variable using correlation coeffient.



State The Set of Assumptions (If Any) Related to The Problem Under Consideration

There were 2089 observations which were having unrealistic entry of age on network (more than 65 years) & 2503 observations which were having age less than 60 days, as there were some columns which were having 30 & 90-day simple moving period average having the consisting data with moving average entries was better. also new customers cannot be assessed with no usage history. so, it was considered to create model based on the people who had at least 60 days of usage history & less than 65 years using cellular network.

Following image shows the Statistical description after data processing/cleaning.

	count	mean	std	min	25%	50%	75%	max
label	204615.0	0.879158	0.325944	0.000000	1.00	1.000	1.000000	1.000000
aon	204615.0	663.928168	497.103727	60.000000	252.00	529.000	971.000000	2440.000000
daily_decr30	204615.0	5453.891350	9255.079343	-93.012667	45.10	1542.851	7364.905667	265926.000000
rental90	204615.0	3517.668752	5789.823010	-24720.580000	315.33	1363.010	4249.970000	200148.110000
last_rech_amt_ma	204615.0	2079.973599	2376.120726	0.000000	770.00	1539.000	2309.000000	55000.000000
cnt_ma_rech30	204615.0	4.020824	4.269696	0.000000	1.00	3.000	5.000000	203.000000
medianamnt_ma_rech30	204615.0	1826.269946	2074.400513	0.000000	770.00	1539.000	1924.000000	55000.000000
medianmarechprebal30	204615.0	3859.439485	54083.320531	-200.000000	11.20	34.400	84.000000	999479.419319
fr_ma_rech90	204615.0	7.739203	12.563664	0.000000	0.00	2.000	8.000000	88.000000
sumamnt_ma_rech90	204615.0	12544.672756	16921.755295	0.000000	2320.00	7518.000	16172.000000	953036.000000
medianamnt_ma_rech90	204615.0	1877.792823	2085.411154	0.000000	773.00	1539.000	1924.000000	55000.000000
medianmarechprebal90	204615.0	92.787623	372.340954	-200.000000	15.00	36.450	80.000000	41456.500000
fr_da_rech90	204615.0	0.045691	0.950159	0.000000	0.00	0.000	0.000000	64.000000
medianamnt_loans30	204615.0	0.053791	0.217474	0.000000	0.00	0.000	0.000000	3.000000
cnt_loans90	204615.0	18.519261	224.549903	0.000000	1.00	2.000	5.000000	4997.517944
amnt_loans90	204615.0	23.897896	26.587967	0.000000	6.00	12.000	30.000000	438.000000
maxamnt_loans90	204615.0	6.713701	2.116385	0.000000	6.00	6.000	6.000000	12.000000
medianamnt_loans90	204615.0	0.045779	0.199904	0.000000	0.00	0.000	0.000000	3.000000
payback30	204615.0	3.430614	8.808454	0.000000	0.00	0.000	3.800000	171.500000
payback90	204615.0	4.365751	10.316636	0.000000	0.00	1.750	4.500000	171.500000

Hardware and Software Requirements and Tools Used

a. Software

i. ➔ Jupyter Notebook (Python 3.9)
ii. ➔ Microsoft Office
iii. ➔ Tableau

b. Hardware

i. ➔ Processor – AMD Ryzen 5
ii. ➔ RAM - 8 GB
iii. ➔ Graphic Memory - 4Gb, Nvidia GEFORCE RTX1650

c. Python Libraries

i. ➔ Pandas
ii. ➔ Numpy
iii. ➔ Matplotlib
iv. ➔ Seaborn
v. ➔ Autoviz & SweetViz
vi. ➔ Scipy
vii. ➔ Sklearn

MODEL/S DEVELOPMENT AND EVALUATION

Identification of possible problem-solving approaches (methods)

The data set was was analysed both statistically and graphically. The statistical analysis showed that,

1. data has outliers, skewness, no null values & zero values
2. independent variables were continuous numerical type data
3. Outliers were removed using z-score method, about 1.31% of data removed.
4. Skewness of some columns were transformed using yeo-Johnson method to have within allowed limits of +/-0.5.
5. Some features were dropped as the entries were did not have any meaning with respect to target variable.
6. Total data loss in pre-processing was 4%.
7. Data imbalance was dealt with using SMOTE method.

Testing of Identified Approaches (Algorithms)

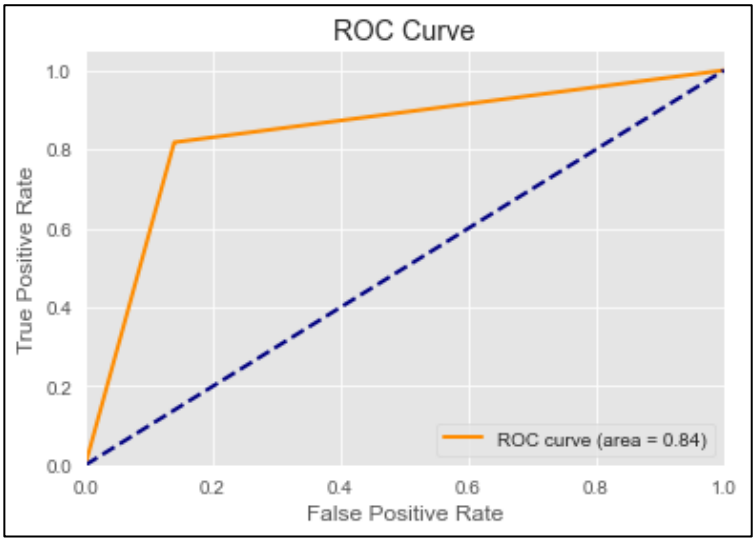
```
#Classification models
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, BaggingClassifier, ExtraTreesClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB

#Evaluation metrics
from sklearn.metrics import confusion_matrix, classification_report, f1_score,
                            roc_auc_score, accuracy_score, roc_curve, auc
```

```
dtc = DecisionTreeClassifier()
etc = ExtraTreesClassifier()
gnb = GaussianNB()
knc = KNeighborsClassifier()
lgr = LogisticRegression()
rfc = RandomForestClassifier()
bgc = BaggingClassifier()
```

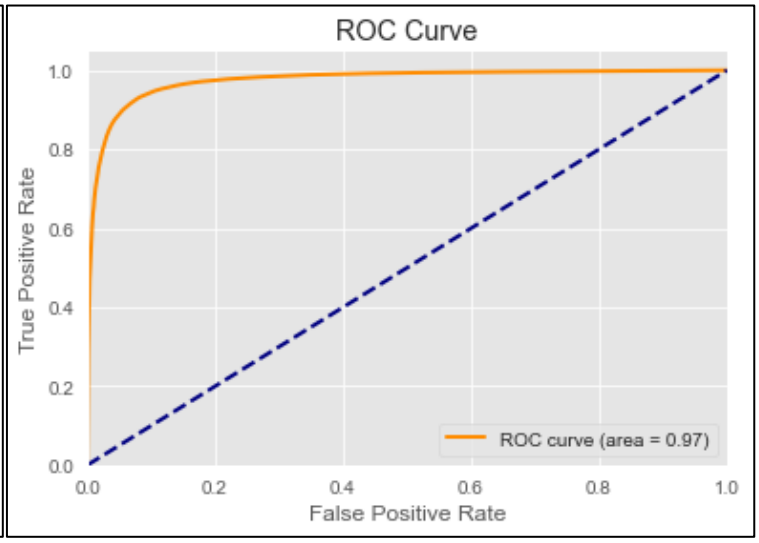
Run and evaluate selected models

```
models = [dtc,etc,gnb,knc,lgr,rfc,bgc]
models_name = ['Decision Tree Classifier','Extra Trees Classifier',
               'Gaussian NB Classifier','KNeighbors Classifier',
               'Logistic Regression','Random Forest Classifier','Bagging Classifier']
```

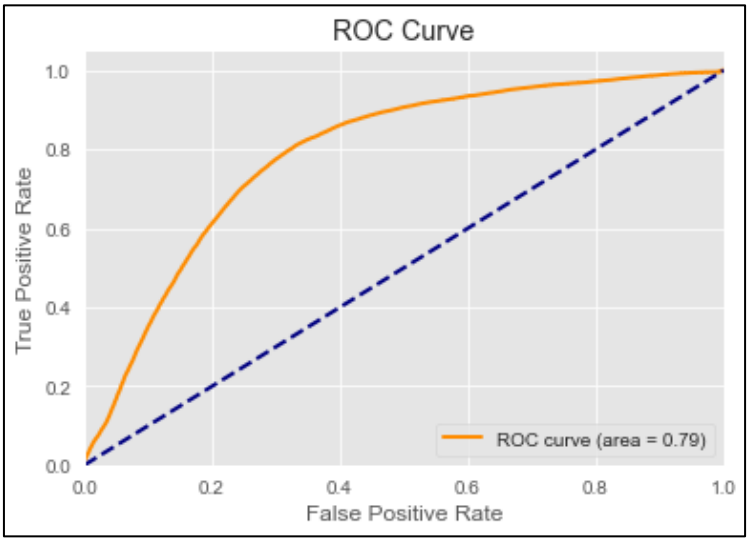
```
for Decision Tree Classifier model..

Best Random_state number for splitting the data is:  15

Accuracy score for Train : 99.98%
Accuracy score for Test : 83.92%
====================
Confusion Matrix :

[[38302  6199]
 [ 8070 36171]]
====================


 Test Classification report
              precision    recall  f1-score   support

           0       0.83      0.86      0.84     44501
           1       0.85      0.82      0.84     44241

    accuracy                           0.84     88742
   macro avg       0.84      0.84      0.84     88742
weighted avg       0.84      0.84      0.84     88742

Cross Validation score at best cv=5 is : 83.96%
```



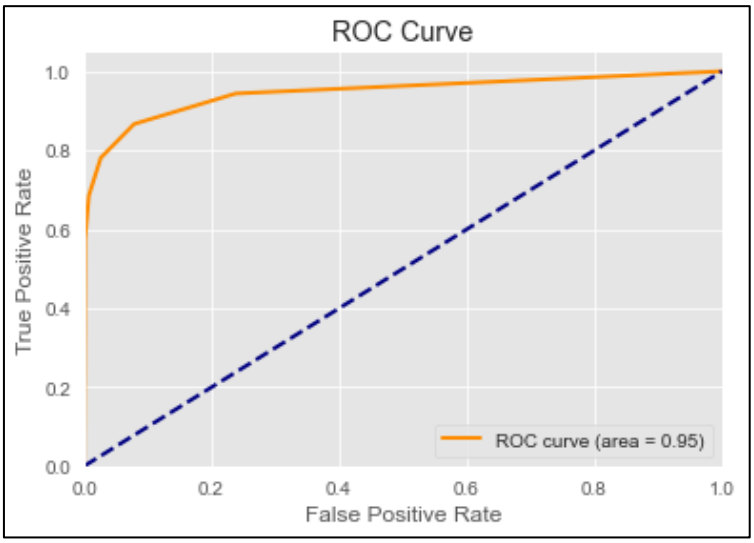
```
for Extra Trees Classifier model..

Best Random_state number for splitting the data is:  12

Accuracy score for Train : 99.98%
Accuracy score for Test : 92.37%
====================
Confusion Matrix :

[[41591  2841]
 [ 3927 40383]]
====================


 Test Classification report
              precision    recall  f1-score   support

           0       0.91      0.94      0.92     44432
           1       0.93      0.91      0.92     44310

    accuracy                           0.92     88742
   macro avg       0.92      0.92      0.92     88742
weighted avg       0.92      0.92      0.92     88742

Cross Validation score at best cv=4 is : 92.43%
```



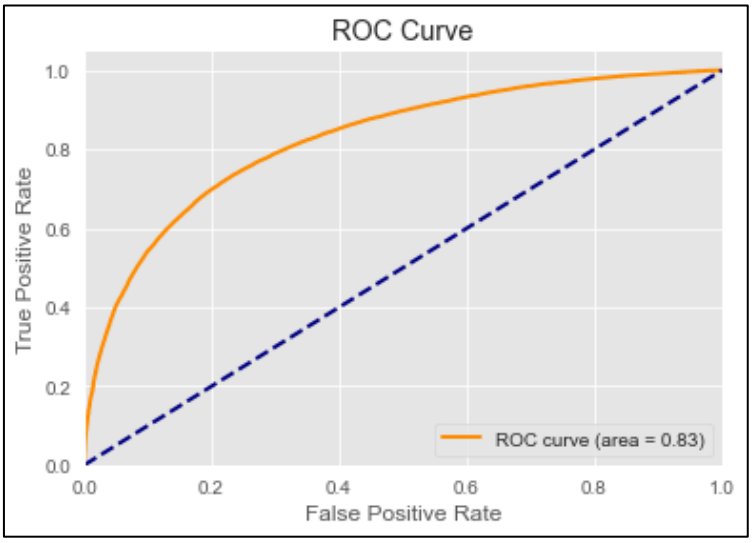

```
for Gaussian NB Classifier model..

Best Random_state number for splitting the data is:  67

Accuracy score for Train : 73.75%
Accuracy score for Test : 73.75%
====================
Confusion Matrix :

[[31191 13231]
 [10062 34258]]
====================


 Test Classification report
              precision    recall  f1-score   support

           0       0.76      0.70      0.73     44422
           1       0.72      0.77      0.75     44320

    accuracy                           0.74     88742
   macro avg       0.74      0.74      0.74     88742
weighted avg       0.74      0.74      0.74     88742

Cross Validation score at best cv=3 is : 73.75%
```



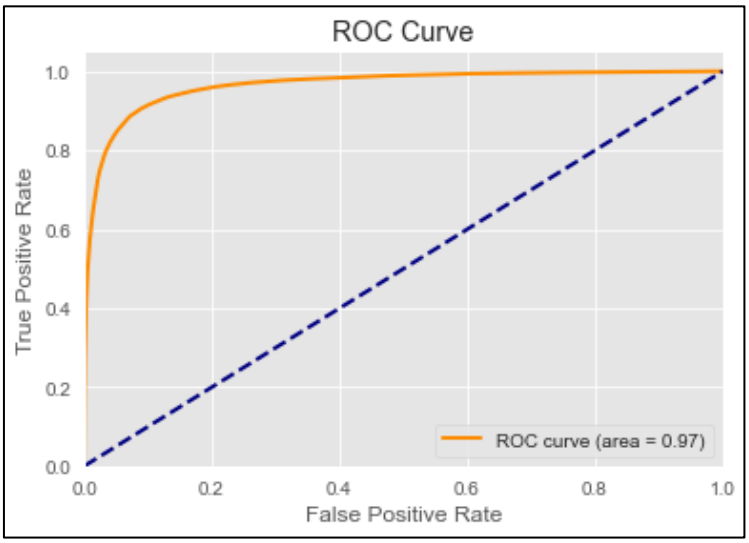

```
for KNeighbors Classifier model..

Best Random_state number for splitting the data is:  75

Accuracy score for Train : 91.22%
Accuracy score for Test : 87.78%
====================
Confusion Matrix :

[[43424  1159]
 [ 9683 34476]]
====================


 Test Classification report
              precision    recall  f1-score   support

           0       0.82      0.97      0.89     44583
           1       0.97      0.78      0.86     44159

    accuracy                           0.88     88742
   macro avg       0.89      0.88      0.88     88742
weighted avg       0.89      0.88      0.88     88742

Cross Validation score at best cv=4 is : 87.76%
```



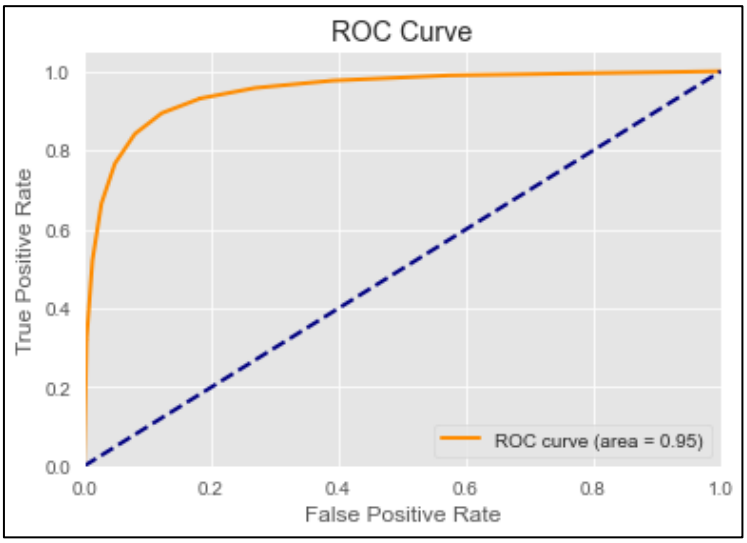
```
for Logistic Regression model..

Best Random_state number for splitting the data is:  94

Accuracy score for Train : 74.82%
Accuracy score for Test : 74.82%
====================
Confusion Matrix :

[[32570 11737]
 [10604 33831]]
====================


 Test Classification report
              precision    recall  f1-score   support

           0       0.75      0.74      0.74     44307
           1       0.74      0.76      0.75     44435

    accuracy                           0.75     88742
   macro avg       0.75      0.75      0.75     88742
weighted avg       0.75      0.75      0.75     88742

Cross Validation score at best cv=7 is : 74.82%
```




```
for Random Forest Classifier model..

Best Random_state number for splitting the data is:  108

Accuracy score for Train : 99.98%
Accuracy score for Test : 90.74%
====================
Confusion Matrix :

[[40881  3401]
 [ 4813 39647]]
====================


 Test Classification report
              precision    recall  f1-score   support

           0       0.89      0.92      0.91     44282
           1       0.92      0.89      0.91     44460

    accuracy                           0.91     88742
   macro avg       0.91      0.91      0.91     88742
weighted avg       0.91      0.91      0.91     88742

Cross Validation score at best cv=4 is : 90.73%
```




```
for Bagging Classifier model..

Best Random_state number for splitting the data is:  119

Accuracy score for Train : 99.30%
Accuracy score for Test : 88.11%
====================
Confusion Matrix :

[[41112  3570]
 [ 6984 37076]]
====================


 Test Classification report
              precision    recall  f1-score   support

           0       0.85      0.92      0.89     44682
           1       0.91      0.84      0.88     44060

    accuracy                           0.88     88742
   macro avg       0.88      0.88      0.88     88742
weighted avg       0.88      0.88      0.88     88742

Cross Validation score at best cv=4 is : 88.04%
```



	algo	best random state	train_accuracy	test_accuracy	Score_diff	best cv fold	cross_val_score
2	Gaussian NB Classifier	67	73.75	73.75	0.00	3	73.746500
4	Logistic Regression	94	74.82	74.82	0.00	7	74.823505
5	Random Forest Classifier	108	99.98	90.74	0.00	4	90.725591
0	Decision Tree Classifier	15	99.98	83.92	0.01	5	83.955082
3	KNeighbors Classifier	75	91.22	87.78	0.02	4	87.762208
6	Bagging Classifier	119	99.30	88.11	0.05	4	88.038573
1	Extra Trees Classifier	12	99.98	92.37	0.05	4	92.431951

We selected ***ExtraTreesClassifier*** for the following reasons:

- High Testing & Cross validation score among all the models.
- Highest area under AUC ROC Curve.

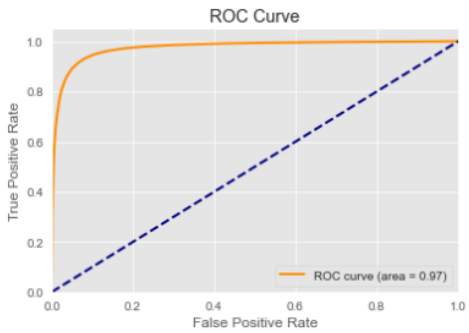
Key Metrics for success in solving problem under consideration

1. **Accuracy Score** - at first our dataset was imbalanced but later we balanced it so first we will look at accuracy score shows best result when it comes to balance the dataset.
2. **F1 - Score** - in this data set the target will decide who is defaulter or not, hence 0 and 1, and as both zero and one is important to us therefore recall and precision what will be our preferred metric and as we all know that, it combines precision and recall into one metric by calculating the harmonic mean between those two and preferred metric is F1 score.
3. **AUC ROC** - We can see a healthy ROC curve, pushed towards the top-left side both for positive and negative classes. It is not clear which one performs better across the board as with FPR = 0.15 positive class is higher and starting from FPR= 0.15 the negative class is above. In order to get one number that tells us how good our curve is, we can calculate the Area Under the ROC Curve, or ROC AUC score. The more top-left your curve is the higher the area and hence higher ROC AUC score.
4. **Cross Validation Score** - to check if our model is overfitting or not, we use cross validation score, higher the cross-validation scores higher the cross-validation score means the model is not overfitting.

Hyperparameter Tuning:

```
In [220]:
x_train, x_test, y_train, y_test = train_test_split(X, y_class, test_size = 0.25, random_state = 12)

In [221]:
param_grid_etc= {'criterion' : ['gini', 'entropy'],
                 'max_features':['auto','sqrt','log2'] }

In [222]:
etc_grid = GridSearchCV(estimator = etc,
                        param_grid = param_grid_etc,
                        verbose = 3,
                        scoring = 'accuracy')

In [223]:
etc_grid.fit(x_train, y_train)

...

In [226]:
etc_grid.best_score_

Out[226]: 0.9123520016273291

In [227]:
etc_grid.best_params_

Out[227]: {'criterion': 'entropy', 'max_features': 'auto'}

In [228]:
etc_final = ExtraTreesClassifier(criterion='entropy',
                                 max_features = 'auto')

In [229]:
etc_final.fit(x_train,y_train)
y_pred=etc_final.predict(x_test)
print('Accuracy Score: ', accuracy_score(y_test, y_pred))

Accuracy Score:  0.9241509093777467
```

```
In [231]:
confusion_matrix_c(y_test, y_pred)
print('\n \n Test Classification report \n', classification_report(y_test, y_pred, digits=2))

cv_score = cross_val_score(etc_final,X, y_class, cv=4, scoring="accuracy").mean()
print(f"Cross Validation score at best cv = 11 is : {cv_score*100:.2f}%")
y_predict_probabilities = etc_final.predict_proba(x_test)[:,1]
fpr, tpr, _ = roc_curve(y_test, y_predict_probabilities)
roc_auc = auc(fpr, tpr)
plot_roc_auc_curve(fpr, tpr)

====================
Confusion Matrix :

[[41625  2807]
 [ 3924 40386]]
====================


 Test Classification report
              precision    recall  f1-score   support

           0       0.91      0.94      0.93     44432
           1       0.94      0.91      0.92     44310

    accuracy                           0.92     88742
   macro avg       0.92      0.92      0.92     88742
weighted avg       0.92      0.92      0.92     88742

Cross Validation score at best cv = 11 is : 93.07%
```



Saving & predictions of the model on Test data provided

In [232]:
```
filename='Micro_credit_defaulter.pkl'
pickle.dump(etc_final,open(filename,'wb'))
```

In [233]:
```
model =pickle.load(open('Micro_credit_defaulter.pkl','rb'))
pred =model.predict(x_test)
result = pd.DataFrame(list(zip(y_test, pred)), columns = ['Actual', 'Predicted'])
result
```

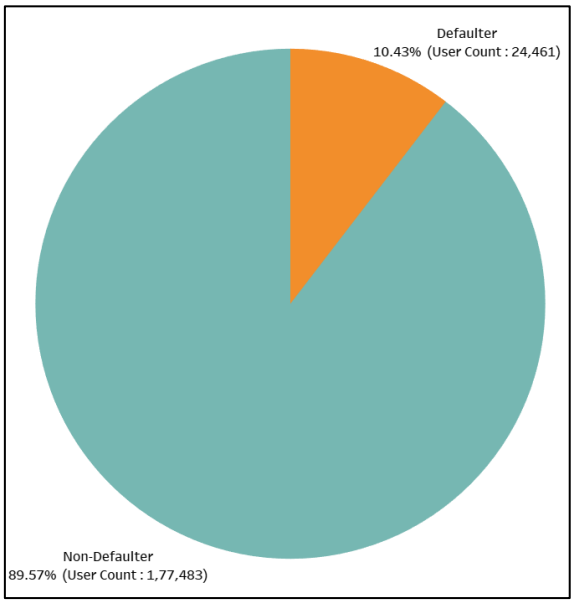
Out[233]:

	Actual	Predicted
0	0	0
1	0	0
2	1	1
3	0	0
4	1	1
...	...	...
88737	1	1
88738	0	0
88739	1	1
88740	1	1
88741	1	1

88742 rows × 2 columns

VISUALIZATIONS & EDA

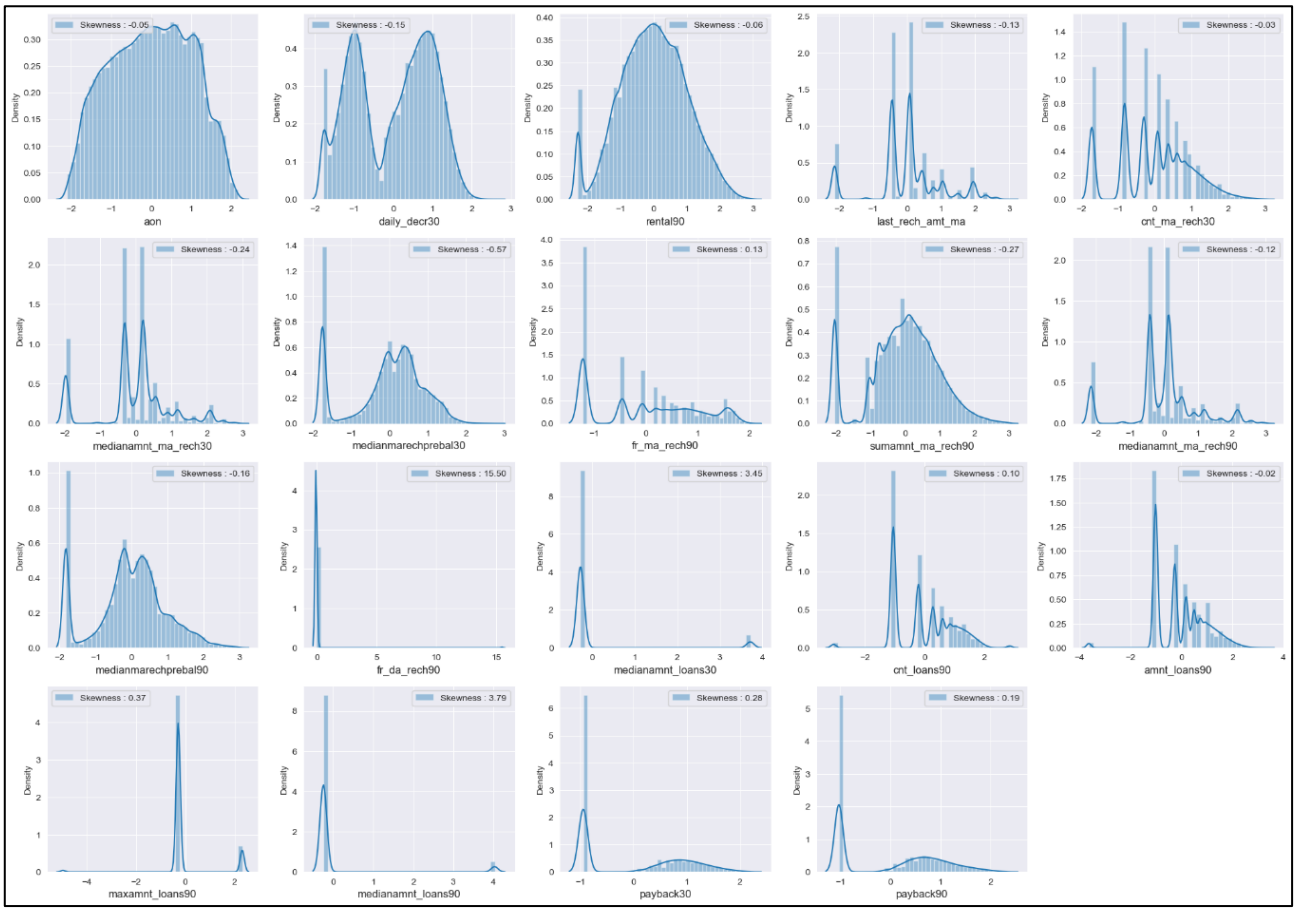
Target Variable:



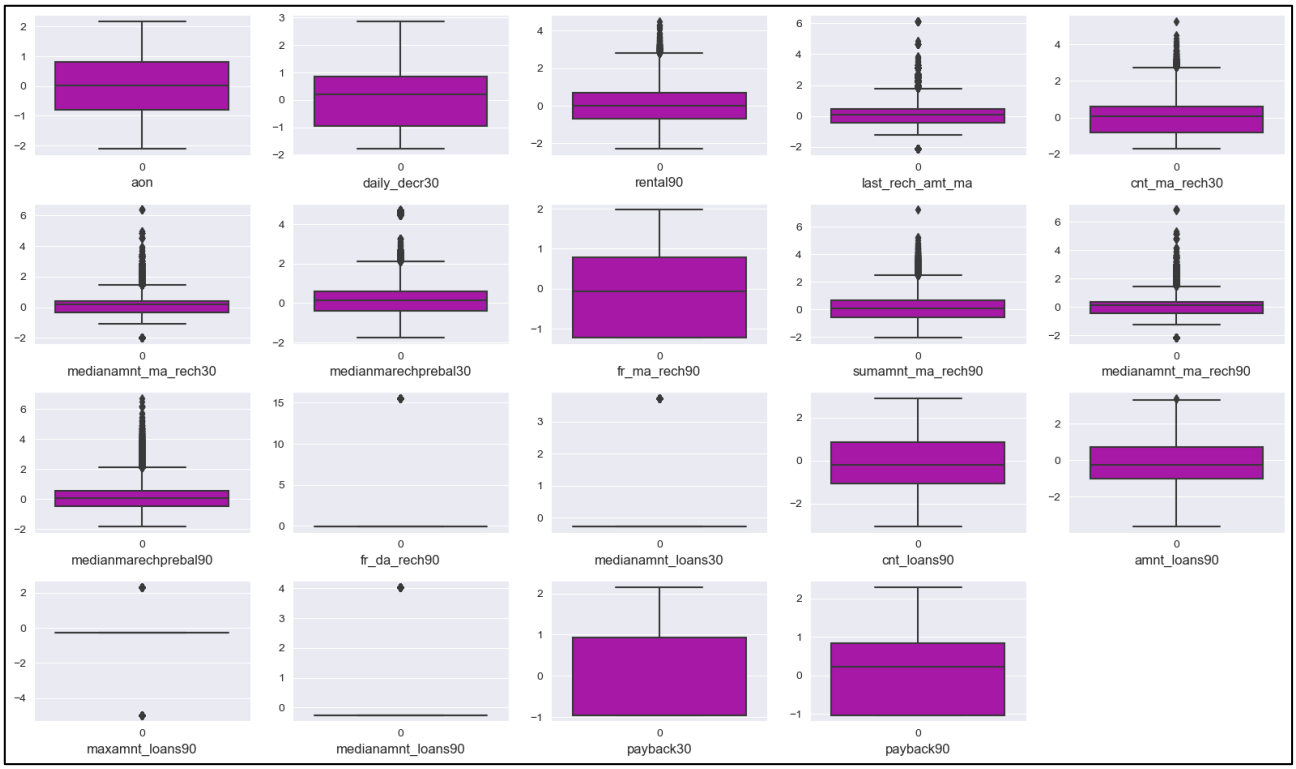
Observation:

We see that input data was highly imbalanced and more than 89% of the observations were non-defaulters. The data was balanced before feeding it to ML models using SMOTE.

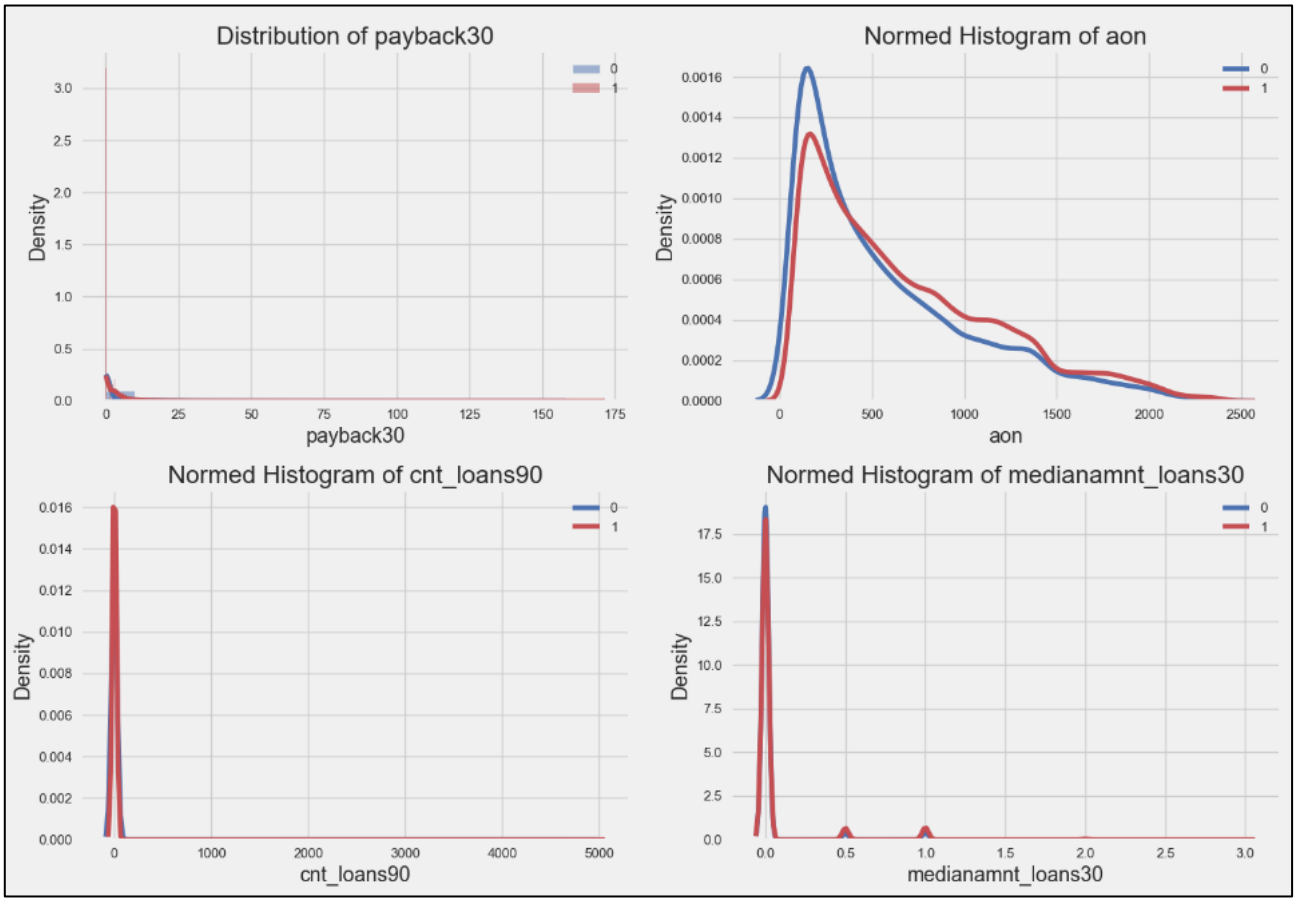
Independent Variables:

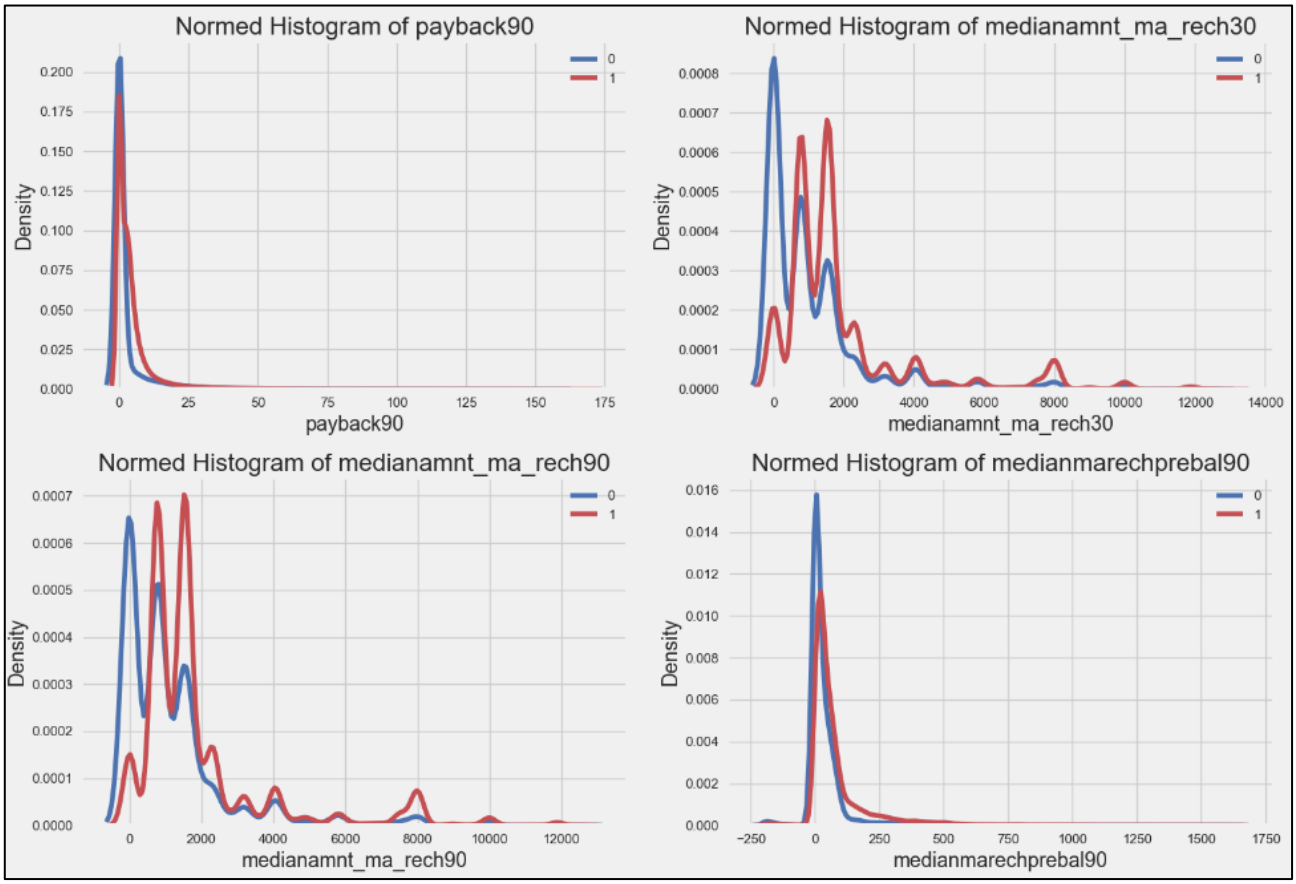


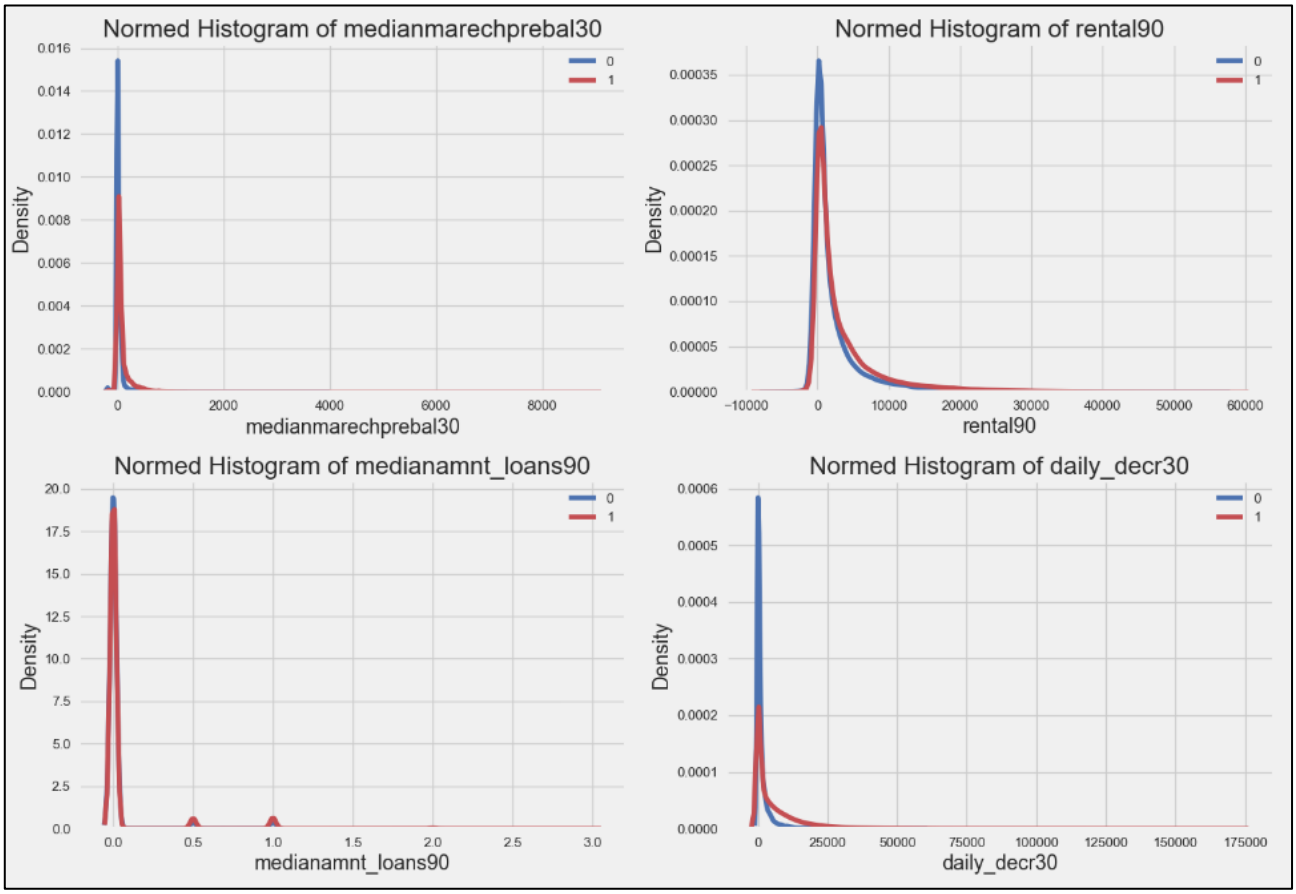
Outliers in Given data:

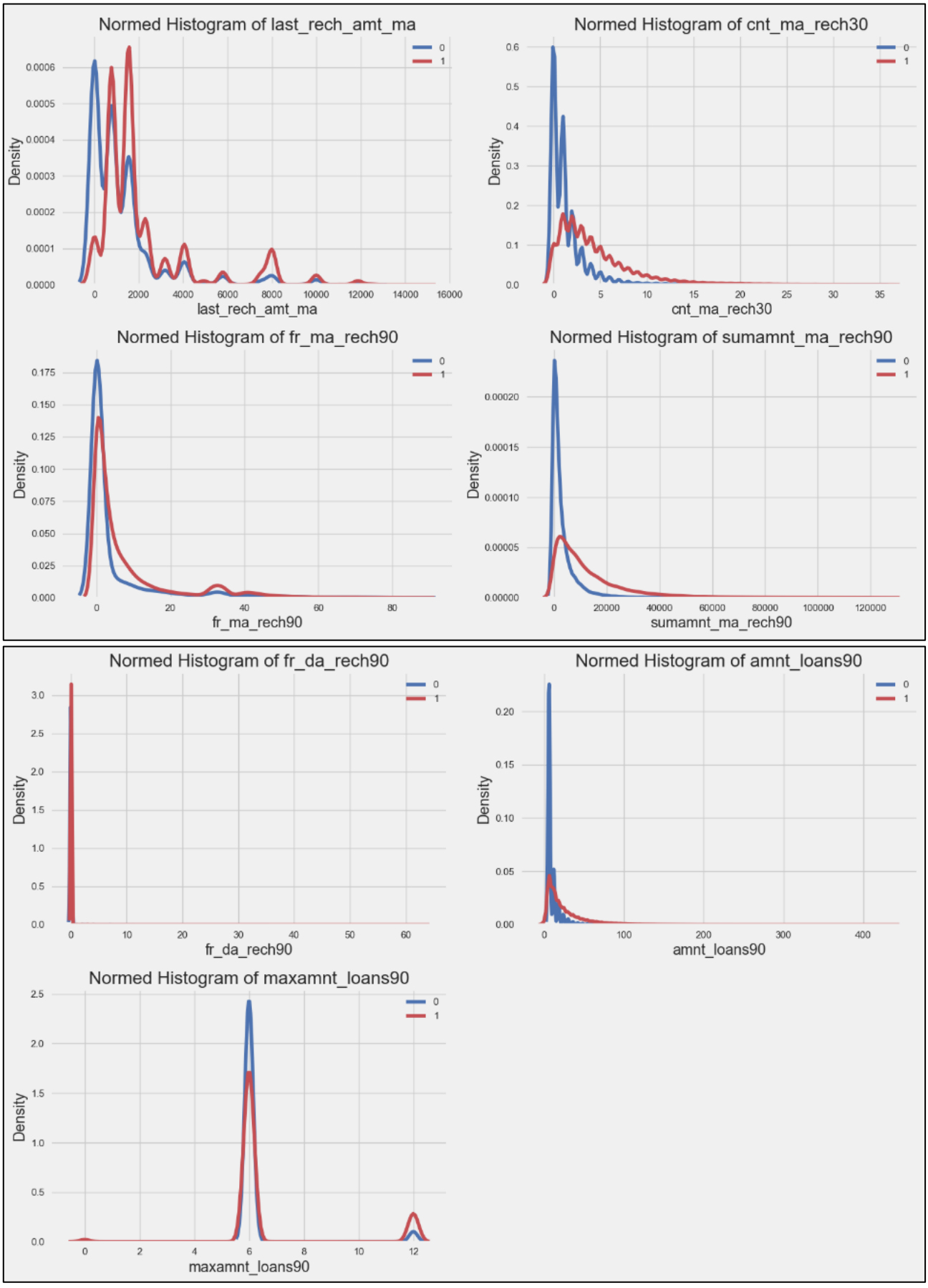


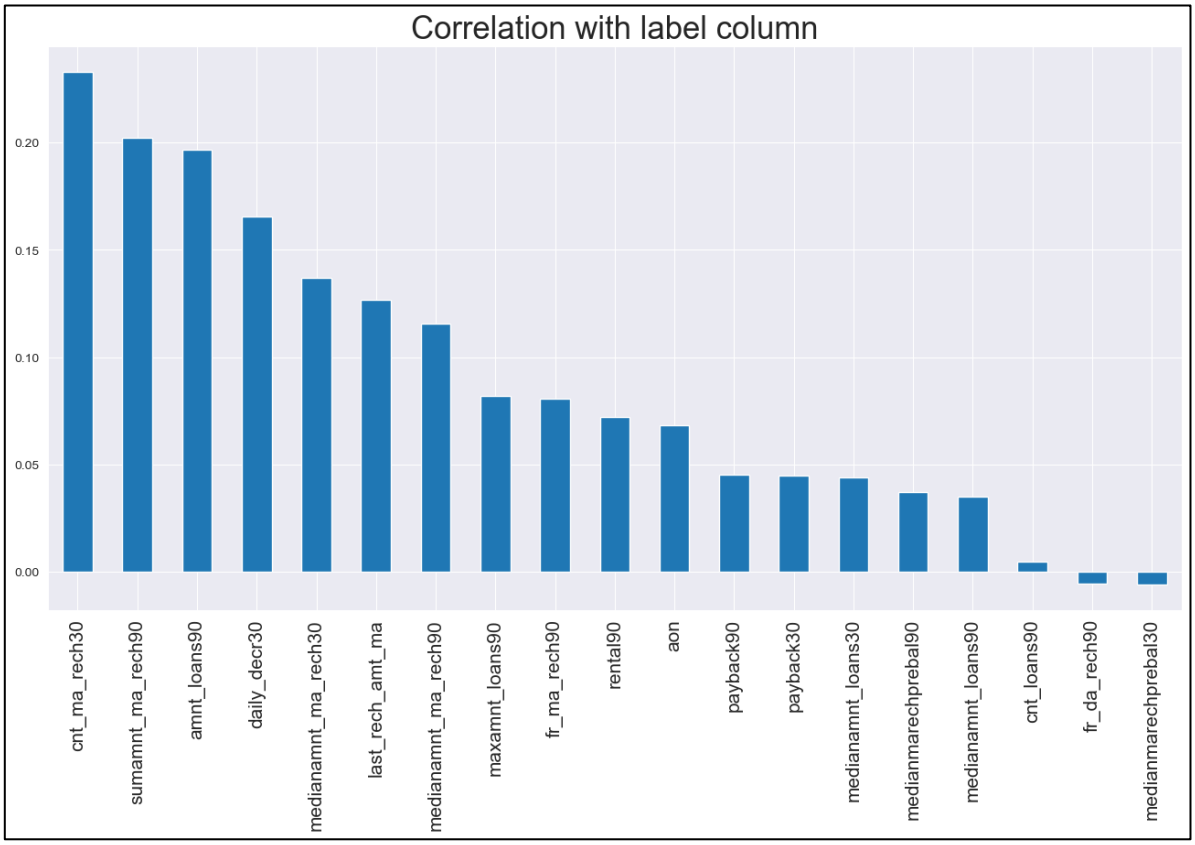
Bivariate Analysis:

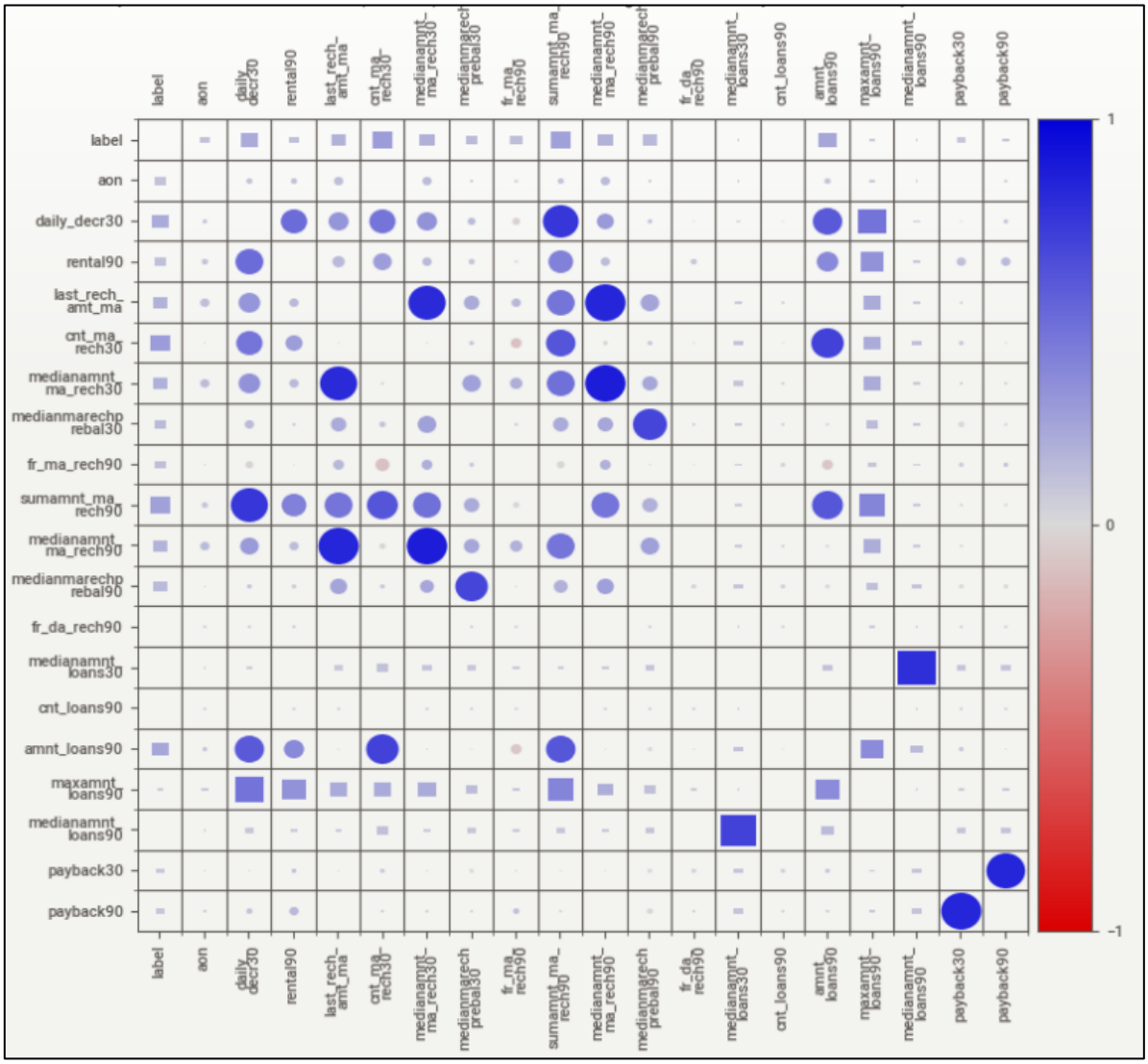


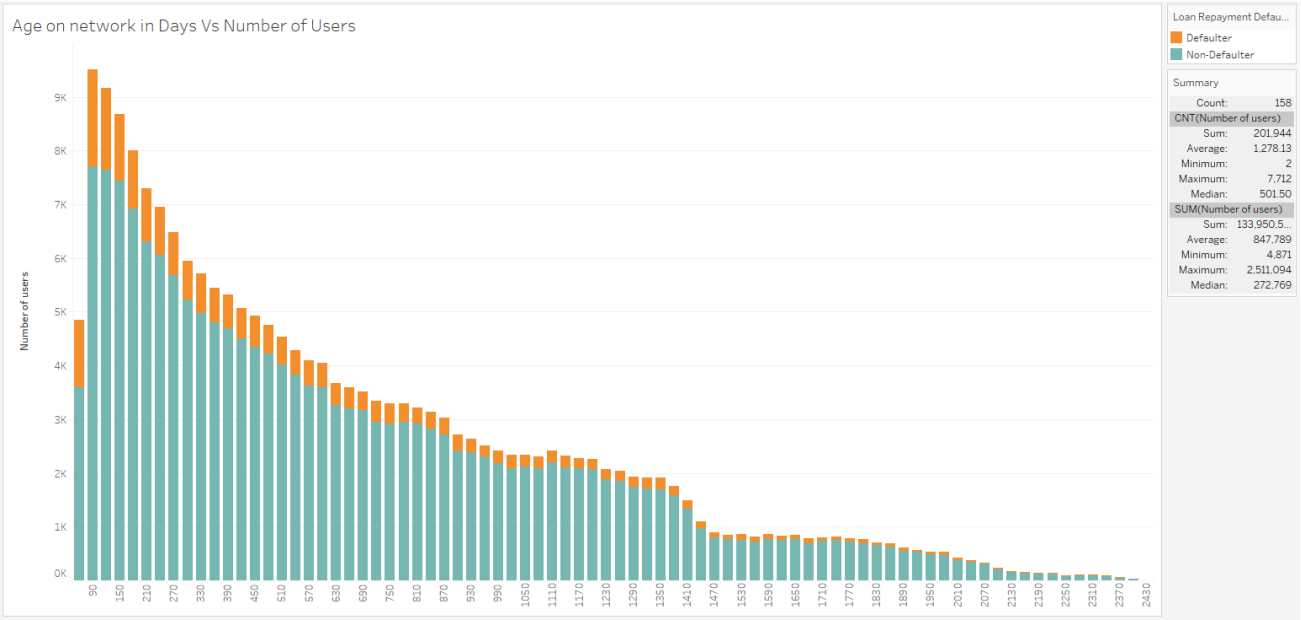

Normed Histogram of payback90
Density
payback90
0
1
Normed Histogram of medianamnt_ma_rech30
Density
medianamnt_ma_rech30
0
1
Normed Histogram of medianamnt_ma_rech90
Density
medianamnt_ma_rech90
0
1
Normed Histogram of medianmarechprebal90
Density
medianmarechprebal90
0
1

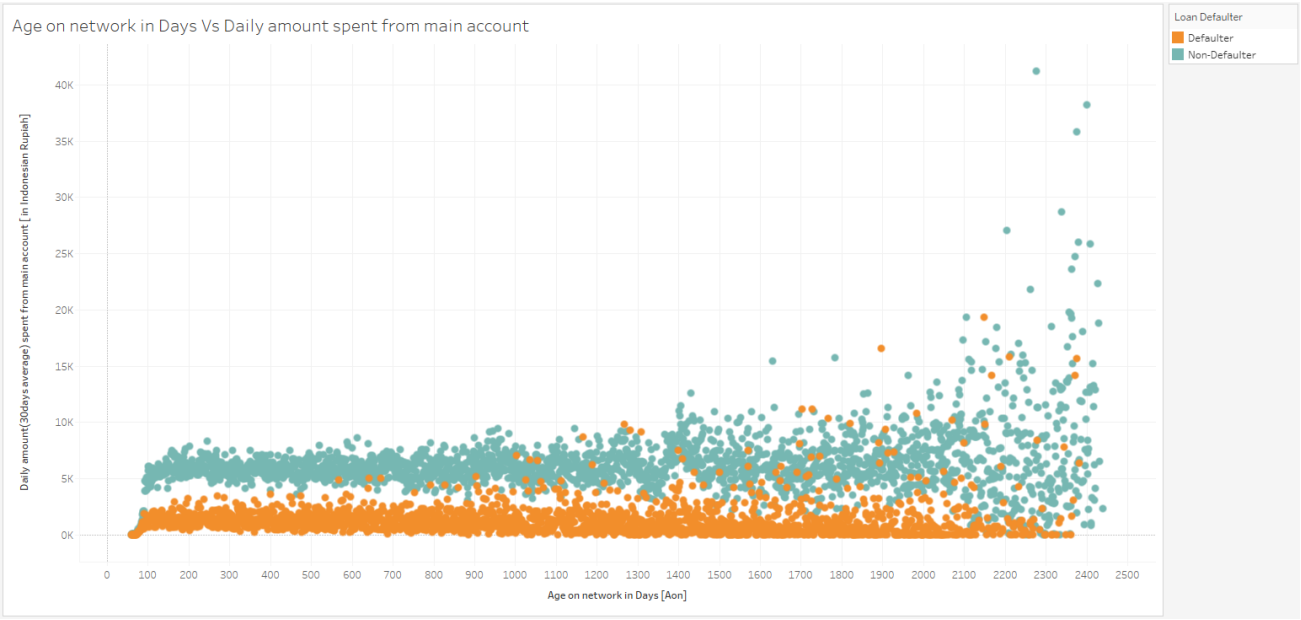

Normed Histogram of medianmarechprebal30
Density
medianmarechprebal30
0
1
Normed Histogram of rental90
Density
rental90
0
1
Normed Histogram of medianamnt_loans90
Density
medianamnt_loans90
0
1
Normed Histogram of daily_decr30
Density
daily_decr30
0
1

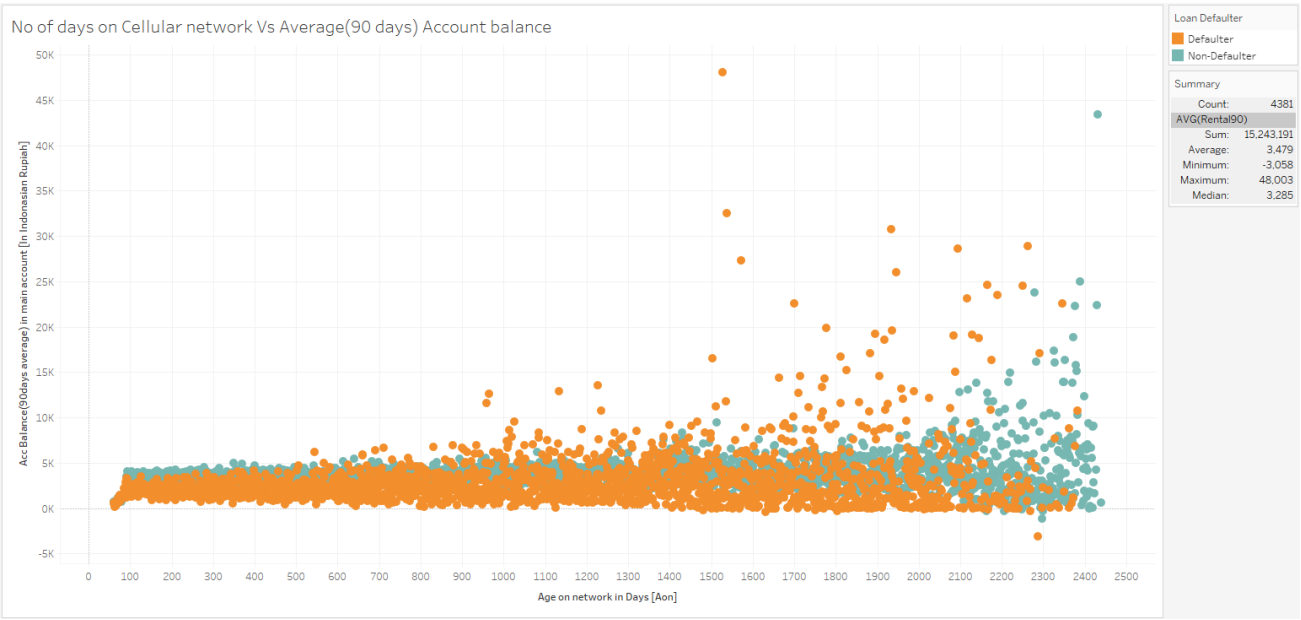

Normed Histogram of last_rech_amt_ma
Density
last_rech_amt_ma
0
1
Normed Histogram of cnt_ma_rech30
Density
cnt_ma_rech30
0
1
Normed Histogram of fr_ma_rech90
Density
fr_ma_rech90
0
1
Normed Histogram of sumamnt_ma_rech90
Density
sumamnt_ma_rech90
0
1
Normed Histogram of fr_da_rech90
Density
fr_da_rech90
0
1
Normed Histogram of amnt_loans90
Density
amnt_loans90
0
1
Normed Histogram of maxamnt_loans90
Density
maxamnt_loans90
0
1

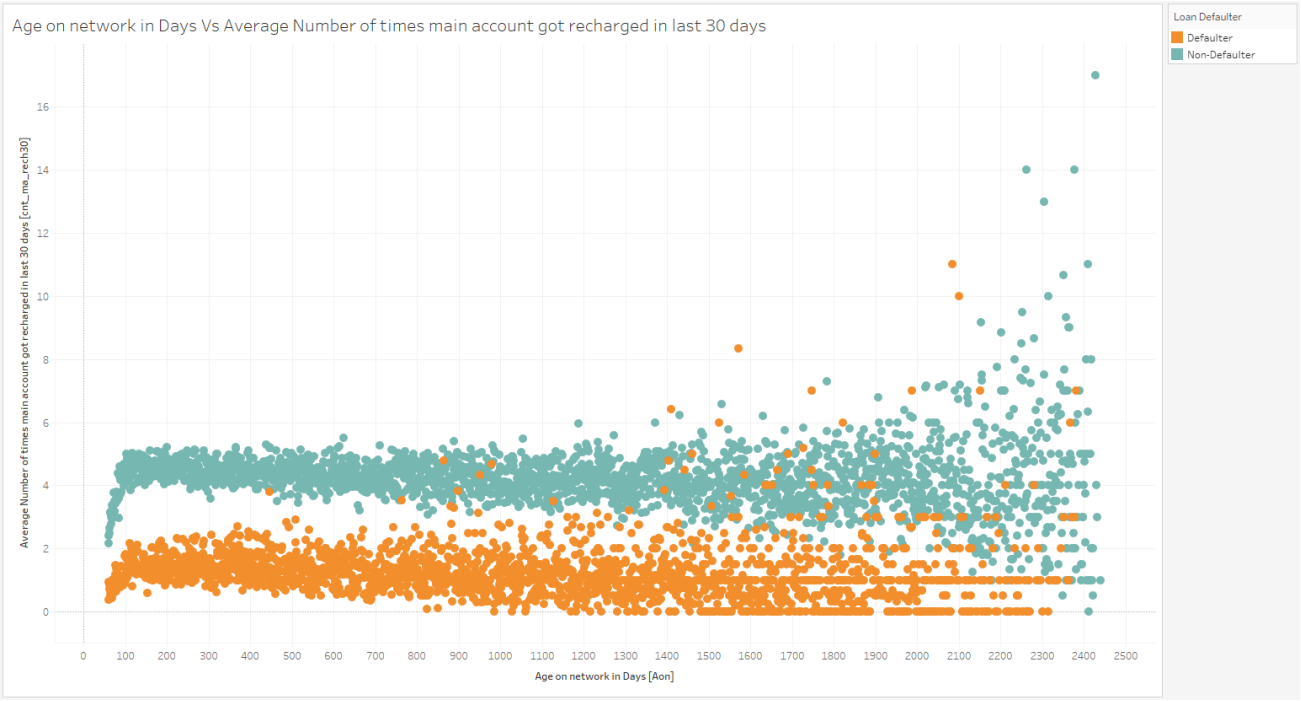

Correlation with label column
0.20
0.15
0.10
0.05
0.00
cnt_ma_rech30
sumamnt_ma_rech90
amnt_loans90
daily_decr30
medianamnt_ma_rech30
last_rech_amt_ma
medianamnt_ma_rech90
maxamnt_loans90
fr_ma_rech90
rental90
aon
payback90
payback30
medianamnt_loans30
medianmarechprebal90
medianamnt_loans90
cnt_loans90
fr_da_rech90
medianmarechprebal30

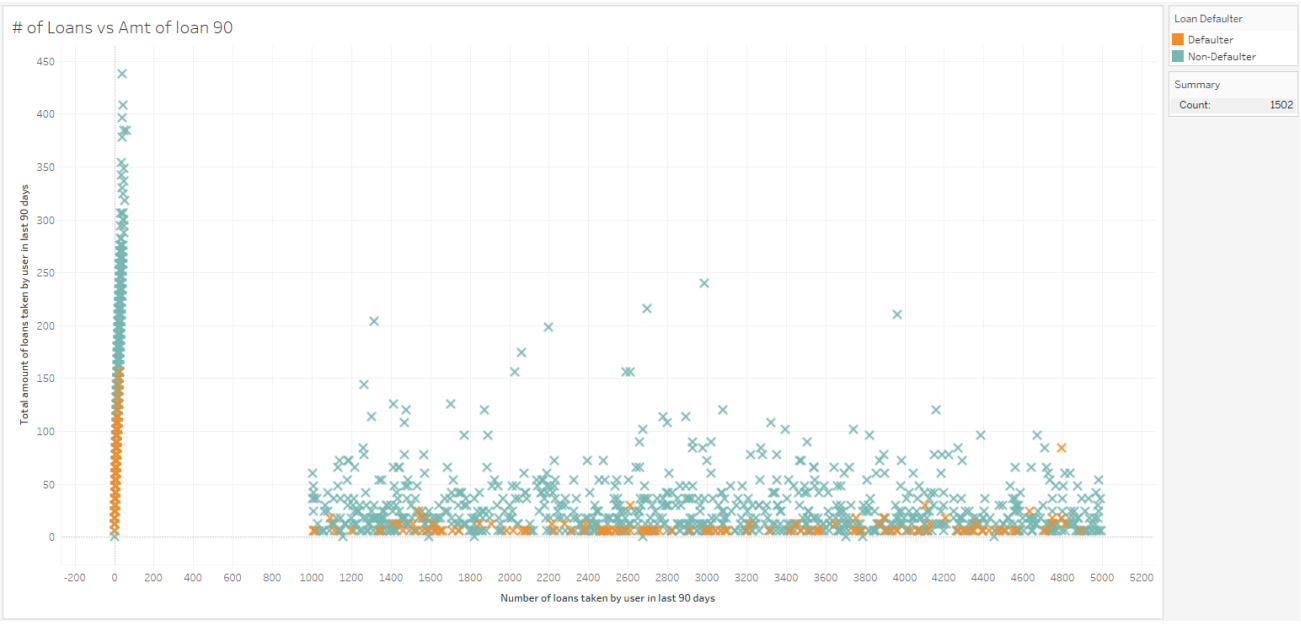

label
aon
daily_decr30
rental90
last_rech_amt_ma
cnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30
fr_ma_rech90
sumamnt_ma_rech90
medianamnt_ma_rech90
medianmarechprebal90
fr_da_rech90
medianamnt_loans30
cnt_loans90
amnt_loans90
maxamnt_loans90
medianamnt_loans90
payback30
payback90
1
0
-1

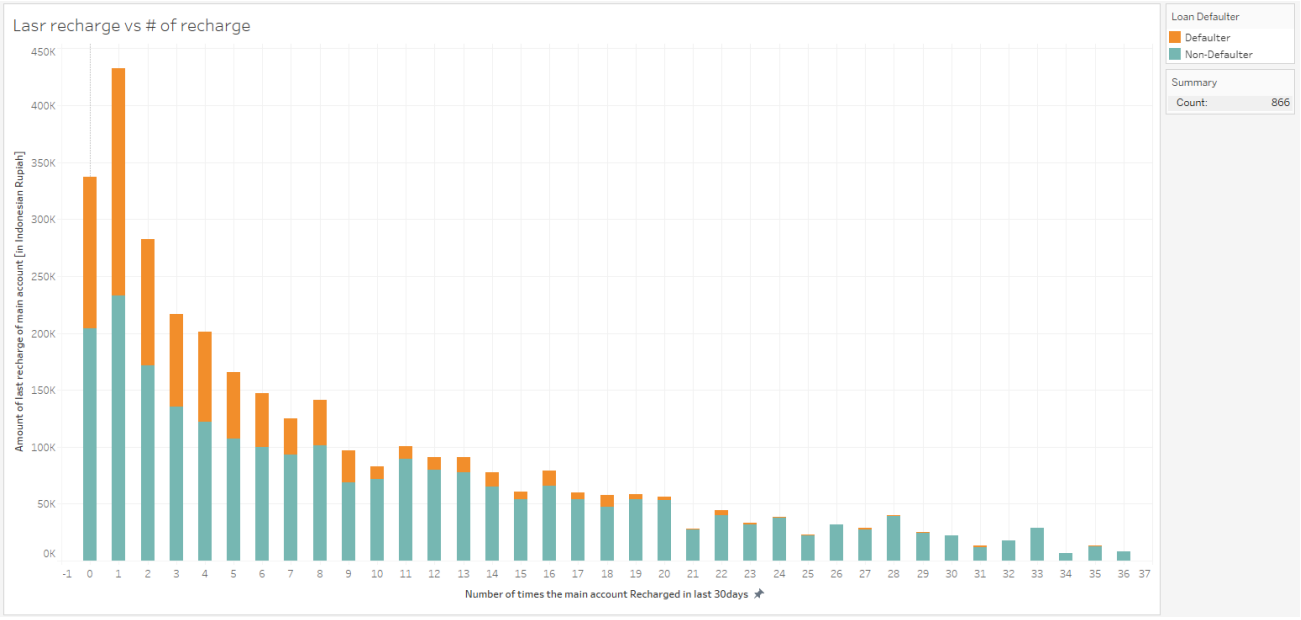

Age on network in Days Vs Number of Users
Number of users
Loan Repayment Defau...
Defaulter
Non-Defaulter
Summary
Count: 158
CNT(Number of users)
Sum: 201,944
Average: 1,278.13
Minimum: 2
Maximum: 7,712
Median: 501.50
SUM(Number of users)
Sum: 133,950,5...
Average: 847,789
Minimum: 4,871
Maximum: 2,511,094
Median: 272,769

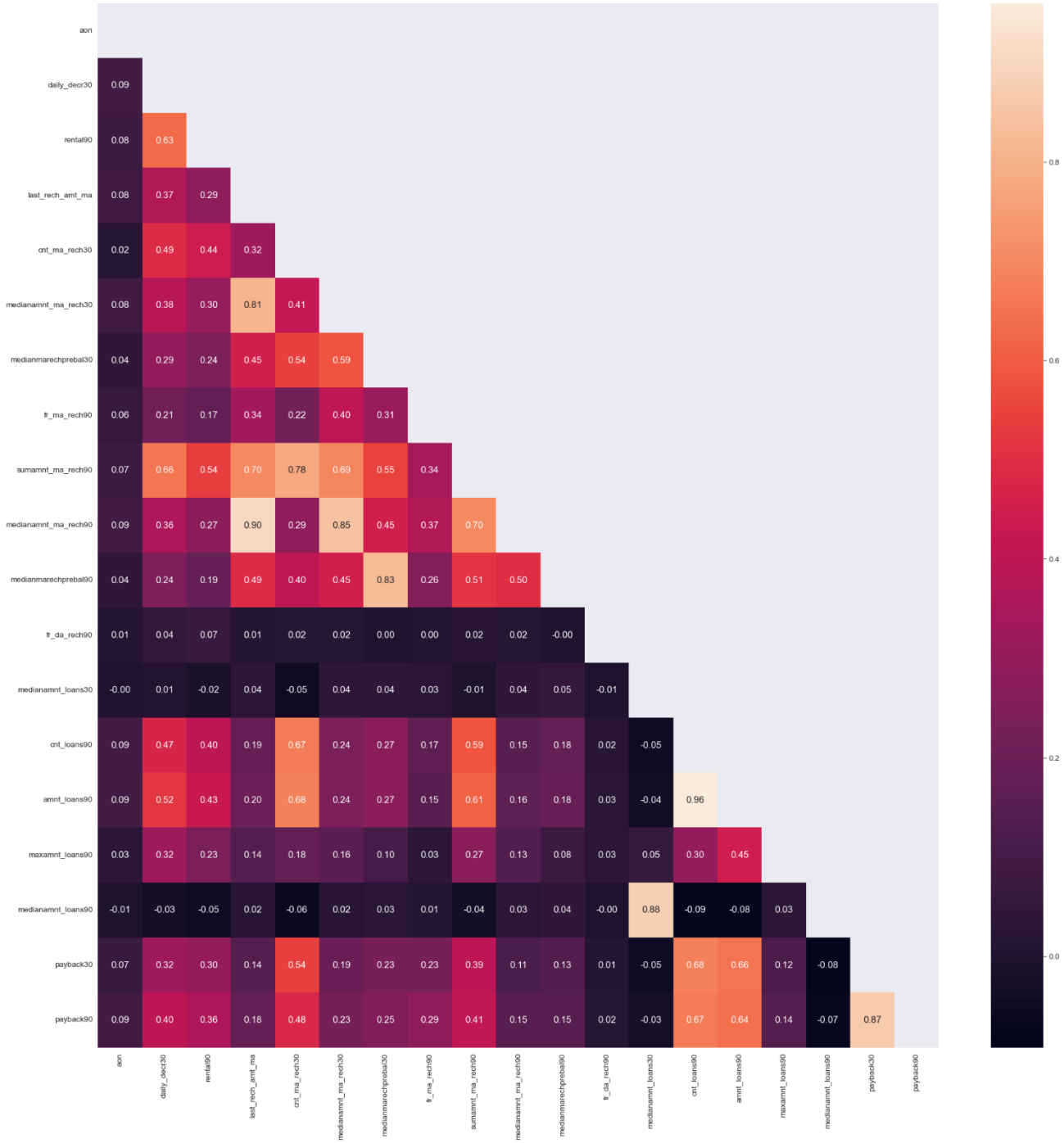

Age on network in Days Vs Daily amount spent from main account
Daily amount(30days average) spent from main account [in Indonesian Rupiah]
Age on network in Days [Aon]
Loan Defaulter
Defaulter
Non-Defaulter


No of days on Cellular network Vs Average(90 days) Account balance
Acc Balance(90days average) in main account [in Indonesian Rupiah]
Age on network in Days [Aon]
Loan Defaulter
Defaulter
Non-Defaulter
Summary
Count: 4381
AVG(Rental90)
Sum: 15,243,191
Average: 3,479
Minimum: -3,058
Maximum: 48,003
Median: 3,285


Age on network in Days Vs Average Number of times main account got recharged in last 30 days
Loan Defaulter
Defaulter
Non-Defaulter
Average Number of times main account got recharged in last 30 days [cnt_ma_rech30]
Age on network in Days [Aon]


# of Loans vs Amt of loan 90
Loan Defaulter
Defaulter
Non-Defaulter
Summary
Count: 1502
Total amount of loans taken by user in last 90 days
Number of loans taken by user in last 90 days


Lasr recharge vs # of recharge
Loan Defaulter
Defaulter
Non-Defaulter
Summary
Count: 866
Amount of last recharge of main account [in Indonesian Rupiah]
Number of times the main account Recharged in last 30days


aon
daily_decr30 0.09
rental90 0.08 0.63
last_rech_amt_ma 0.08 0.37 0.29
cnt_ma_rech30 0.02 0.49 0.44 0.32
medianamnt_ma_rech30 0.08 0.38 0.30 0.81 0.41
medianmarechprebal30 0.04 0.29 0.24 0.45 0.54 0.59
fr_ma_rech90 0.06 0.21 0.17 0.34 0.22 0.40 0.31
sumamnt_ma_rech90 0.07 0.66 0.54 0.70 0.78 0.69 0.55 0.34
medianamnt_ma_rech90 0.09 0.36 0.27 0.90 0.29 0.85 0.45 0.37 0.70
medianmarechprebal90 0.04 0.24 0.19 0.49 0.40 0.45 0.83 0.26 0.51 0.50
fr_da_rech90 0.01 0.04 0.07 0.01 0.02 0.02 0.00 0.00 0.02 0.02 -0.00
medianamnt_loans30 -0.00 0.01 -0.02 0.04 -0.05 0.04 0.04 0.03 -0.01 0.04 0.05 -0.01
cnt_loans90 0.09 0.47 0.40 0.19 0.67 0.24 0.27 0.17 0.59 0.15 0.18 0.02 -0.05
amnt_loans90 0.09 0.52 0.43 0.20 0.68 0.24 0.27 0.15 0.61 0.16 0.18 0.03 -0.04 0.96
maxamnt_loans90 0.03 0.32 0.23 0.14 0.18 0.16 0.10 0.03 0.27 0.13 0.08 0.03 0.05 0.30 0.45
medianamnt_loans90 -0.01 -0.03 -0.05 0.02 -0.06 0.02 0.03 0.01 -0.04 0.03 0.04 -0.00 0.88 -0.09 -0.08 0.03
payback30 0.07 0.32 0.30 0.14 0.54 0.19 0.23 0.23 0.39 0.11 0.13 0.01 -0.05 0.68 0.66 0.12 -0.08
payback90 0.09 0.40 0.36 0.18 0.48 0.23 0.25 0.29 0.41 0.15 0.15 0.02 -0.03 0.67 0.64 0.14 -0.07 0.87
aon
daily_decr30
rental90
last_rech_amt_ma
cnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30
fr_ma_rech90
sumamnt_ma_rech90
medianamnt_ma_rech90
medianmarechprebal90
fr_da_rech90
medianamnt_loans30
cnt_loans90
amnt_loans90
maxamnt_loans90
medianamnt_loans90
payback30
payback90
0.8
0.6
0.4
0.2
0.0

Observations from Bivariate Analysis:

1. Almost all the features have positive correlation with target variable.
2. As seen from 'Aon' feature, there is high number of new customers/users than the old customers/users. Those joined recently were mostly taking loans.
3. Higher the age on network less are the chances of being a defaulter.
4. Most of the users were new users on the network.
5. Most of the users were used to recharge withing10-20 days since last recharge.
6. Those users spending more than 3000 Rupiah from main account are unlikely to be defaulter.
7. Most of the users used to recharge their main account maximum of 1-9 times in last 30 days. Those recharging more than 2-3 times were unlikely to be a defaulter.
8. If age on network is more than 1500 days, it was more likely that users spending more money & being non-defaulters.
9. Those users with total number of loans taken more than 1000 times in last 3 months are unlikely to be a defaulter. Also, those users who exceed the total amount of loan more than 150 rupias are likely to pay back.
10. Out of Users recharging their main account with more than 100K Rupiah & recharge less than 10 times in a month, more than 35% users will default their loan payment.
11. Most of the users paid back within 3 days, so people who didn't return the money till 10 days are most likely to be defaulters.

CONCLUSION

Key Findings and Conclusions of the Study

- Those users with recharge history of 2-3 times in a month are less likely to be a defaulter.
- Those who pay beck the loan within 2-3 days should be given preferences for micro credit.
- Those availing loans more than 1000 times & paying back the same within 3-days should be given micro credit.
- Extra Trees Classifier was best performing model which has an accuracy more than 90% in predicting whether a user will default or not if he given a micro credit loan.

Learning Outcomes of the Study in respect of Data Science

We were able to understand & study about the data given, following points were the learning outcome of the projects:

- Identifying unrealistic observations in a dataset,
- Dealing with duplicate entries, unnecessary features,
- Dealing with highly skewed features,
- Removing Highly correlated independent features & multicollinearity,
- Dealing with data imbalance
-

Limitations of this work and Scope for Future Work

- Date was not consistent, has high skewness & had unrealistic values even after cleaning.
- Highly skewed features were limitation for in depth EDA & unable to understand dependencies of features.
- Most of the entries were meaningless. E.g., Users taking loan more than 2000 times in 90 days period is unrealistic & that for 5 & 10 Rupiah is non-sensical & undigestible.
- Skewness limits the model accuracy.
- Multi collinearity issues in some of the features.
- Having average & median features was not a good idea. More than 15 features were having no relation with target variable.