FLIP ROBO

# MICRO CREDIT LOAN PROJECT

Submitted by:

SATYANATH DAS

# ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to **Flip Robo Technologies** who gave me the golden opportunity to do this internship project on the topic **Micro Credit Loan Defaulter**, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them. I would like to express my special thanks to my SME Miss Swati Rustagi who gave me the golden opportunity to do this wonderful project.

The sample data is provided to us from FlipRobo’s client database. Kaggle, Github, scikit-learn.org, www.scipy.org are the websites which helped me in completing the project.

# 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.

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.

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.

- ## Conceptual Background of the Domain Problem

The Company provide the previous loan data of customer. So we can build a **Machine Learning Model** which can help the company to determine who is defaulter customer. We use **Exploratory Data Analysis** and **Visualization** to determine the defaulter customer and using python programming we build a predictive model.

- ## Review of Literature

The Data provided to us by company has 3 months of loan data which company was given and there are 12.5% defaulters.

The client wants some predictions that could help them to invest and improvement in customer selection.

First we do all the data processing steps, view statistical information and do EDA to visualize the data graphically and after that we make a machine learning model in order to improve the selection of customers for the credit.

## • Motivation for the Problem Undertaken

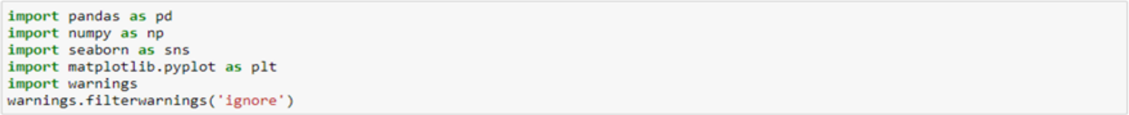
The client wants some predictions that could help them in further investment and improvement in selection of customers. So to help them.
My motivation behind this project is to do the proper research because we are working with one such client that is in Telecom Industry. They are a fixed wireless telecommunications 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.

# Analytical Problem Framing

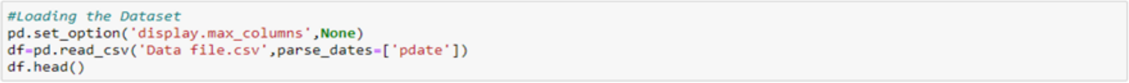
## • Mathematical/ Analytical Modeling of the Problem

Let’s import our  csv data file into by importing some important library and loading into our Jupyter Notebook

### Importing Important Library



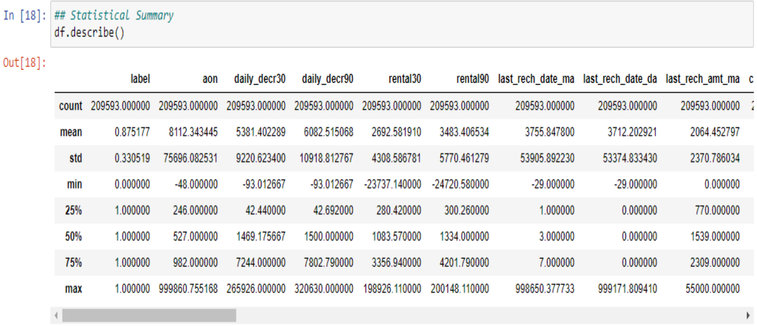
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```



```
#Loading the Dataset
pd.set_option('display.max_columns',None)
df=pd.read_csv('Data file.csv',parse_dates=['pdate'])
df.head()
```

The Data Set has 209593 Rows which means that is Customer Number. There are 37 columns which means 37 different attributes including target column.

### Statistical Summery:

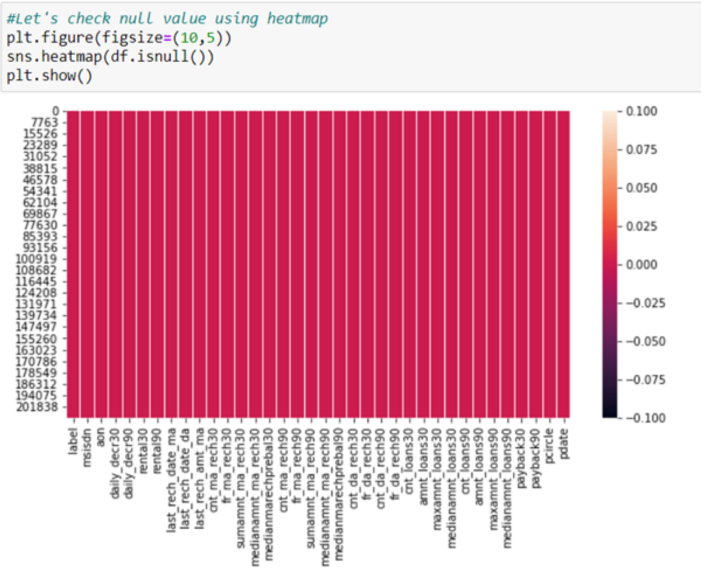


```
In [18]: ## Statistical Summary
         df.describe()
```

Out[18]:

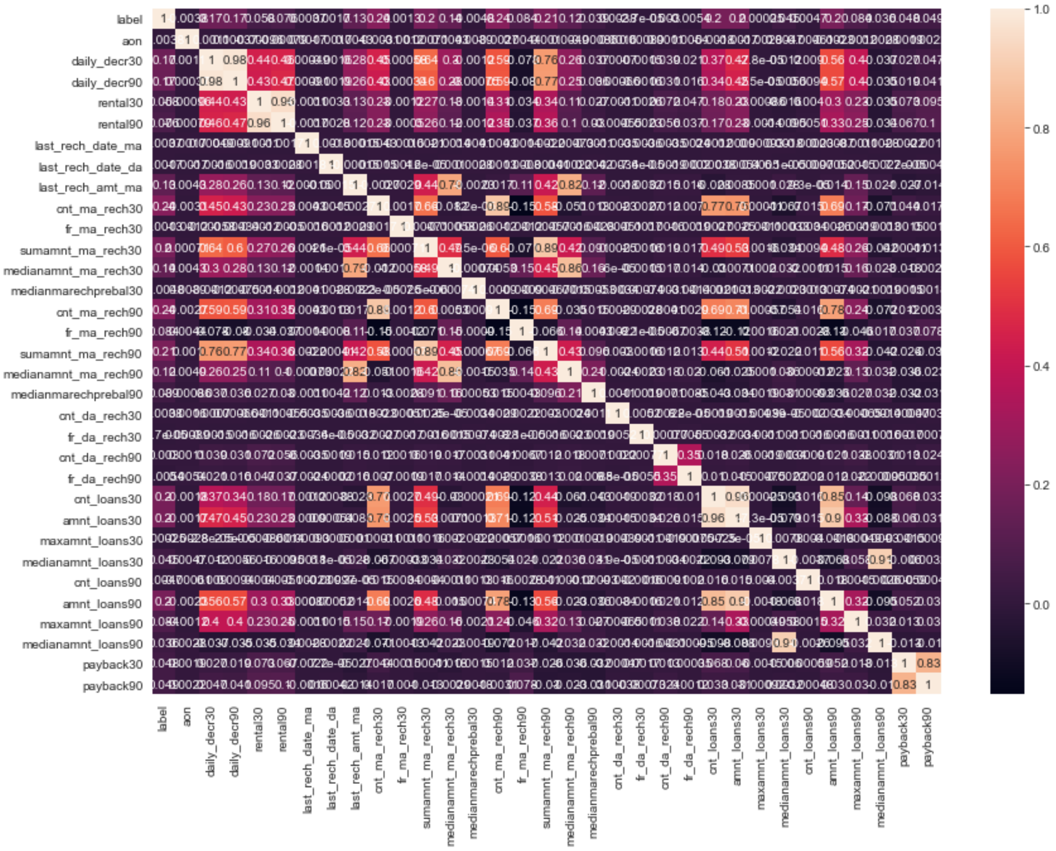
| | label | aon | daily_decr30 | daily_decr90 | rental30 | rental90 | last_rech_date_ma | last_rech_date_da | last_rech_amt_ma |
|---|---|---|---|---|---|---|---|---|---|
| count | 209593.000000 | 209593.000000 | 209593.000000 | 209593.000000 | 209593.000000 | 209593.000000 | 209593.000000 | 209593.000000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

### Checking Null Values:



```
#Let's check null value using heatmap
plt.figure(figsize=(10,5))
sns.heatmap(df.isnull())
plt.show()
```

## Correlation Between Target and Other Columns:

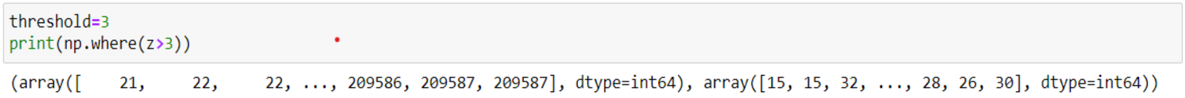


- Data Preprocessing Done

We learn about the data by columns. There are some columns available which contains some personal information about customers like Mobile no, Network Circle, Loan date these columns have no requirement in model building. So I dropped those columns.

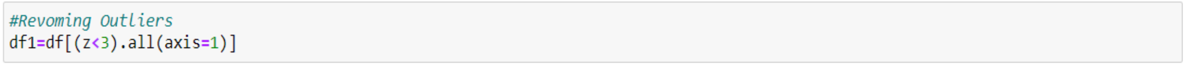
After checking the dataset we also found that there are no null values. If null values are present we replace them with mean, median, mode according to Columns or attributes.

We also Check for Outliers in Dataset and found that there are outliers present in every column and we try to remove them with zscore, where threshold=3.Bt the outlier data contain more than 20% of whole data. If we remove those data we may lose the data about some of defaulters. So we could not remove outlier

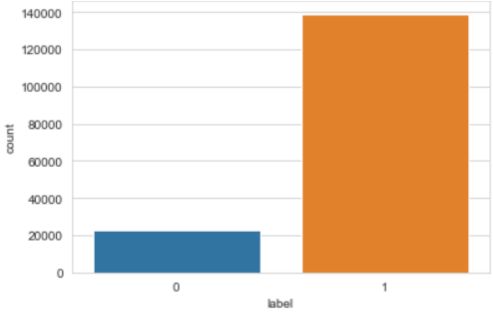


```
threshold=3
print(np.where(z>3))
```

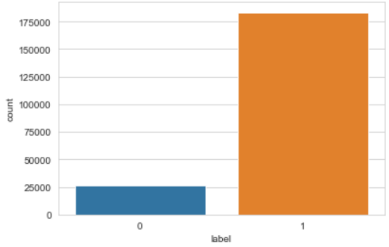
```
(array([     21,      22,      22, ..., 209586, 209587, 209587], dtype=int64), array([15, 15, 32, ..., 28, 26, 30], dtype=int64))
```



```
#Revoming Outliers
df1=df[(z<3).all(axis=1)]
```



Shape of Only Numerical Column After Outliers Removed



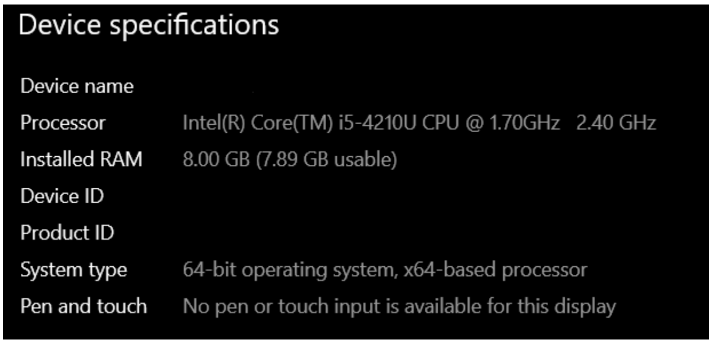
So as we can see that if we remove outliers we lose so much data. Maximum row detect as outliers but not remove the outliers beacuse dataset is inbalance and if we remove the outliers than some row detect which contain information about defaulter.

- Hardware and Software Requirements and Tools Used

Hardware:
Processor—Intel (R) Core(TM) i5-4210U CPU @1.70GHZ @2.40GHz

Installed Memory (RAM)—8.00 GB

System type—64-bit Operating System
Software: Windows 10



Device specifications

| | |
|---|---|
| Device name | |
| Processor | Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz   2.40 GHz |
| Installed RAM | 8.00 GB (7.89 GB usable) |
| Device ID | |
| Product ID | |
| System type | 64-bit operating system, x64-based processor |
| Pen and touch | No pen or touch input is available for this display |

We have used Python Package because it is powerful and general purpose programming language.

NumPy—It is a math library to work with N dimensional arrays. It enables us to do computation effectively and regularly. For working with arrays, dictionary, functions data type we need to know NumPy

Pandas—It is high level Python library and easy to use for data importing, manipulation and data analysis.

Matplotlib—It is a plotting that provide 2D and 3D plotting.

Seaborn-- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

SciPy—It is a collection of numerical algorithm and domain specific tool boxes including optimization, statistics and much more.

Scikit-learn—It is a collection of tools and algorithm for machine learning. It works with NumPy and SciPy and it is easy to implement machine learning models.

# Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

We know that It is a **Classification Problem** so we use accuracy score, classification report, confusion matrix for evaluation matrix. Then we use precision, recall, F1 score, auc roc score, auc roc score, cross validation score for finalizing the model.

- Testing of Identified Approaches (Algorithms)

Listing down all the algorithms used for the training and testing

Logistic Regression

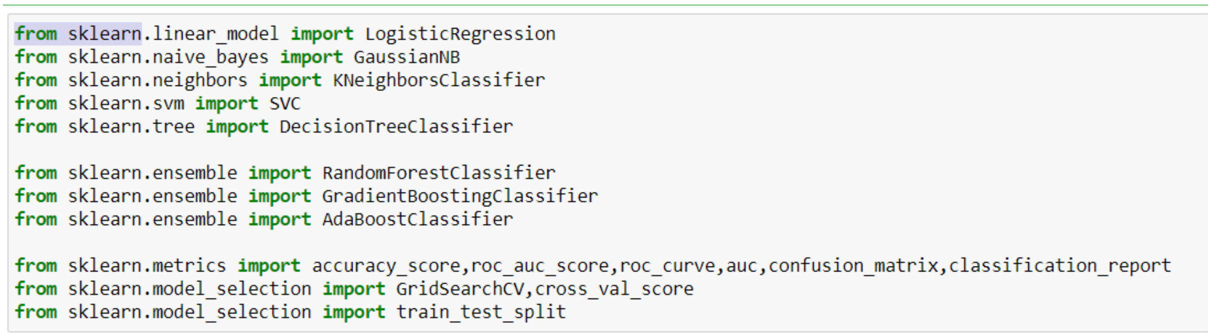
KNeighbors Classifier

GausianNB

Random Forest Classifier

Ada Boost Classifier

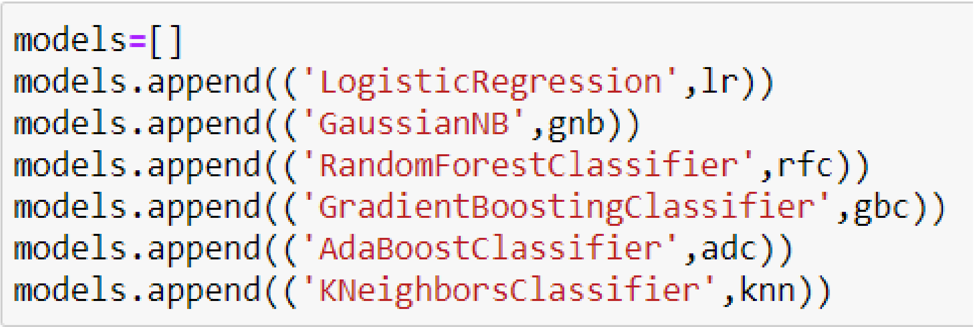
- Run and Evaluate selected models



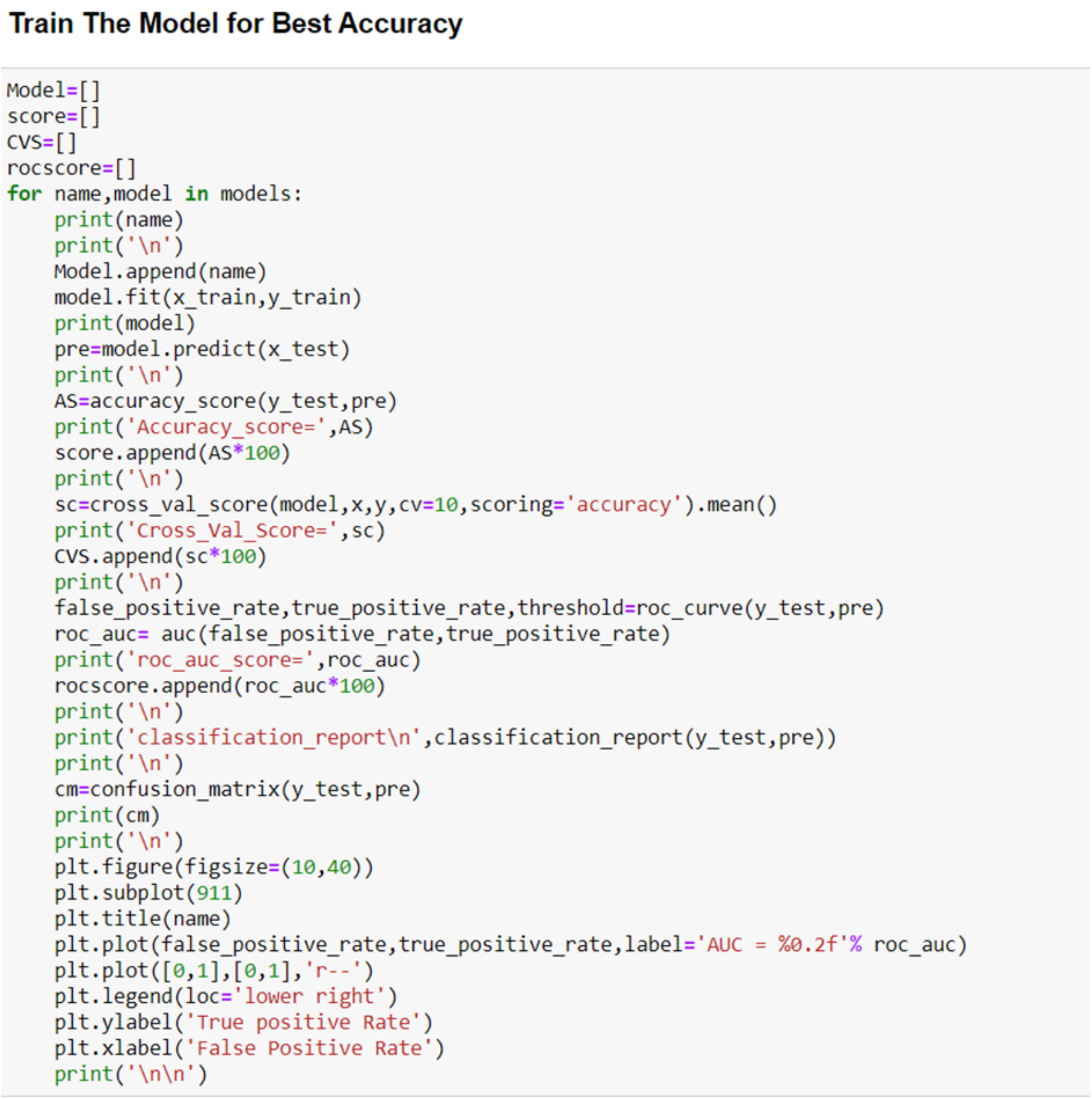
```
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import AdaBoostClassifier

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



```
models=[]
models.append(('LogisticRegression',lr))
models.append(('GaussianNB',gnb))
models.append(('RandomForestClassifier',rfc))
models.append(('GradientBoostingClassifier',gbc))
models.append(('AdaBoostClassifier',adc))
models.append(('KNeighborsClassifier',knn))
```



### Train The Model for Best Accuracy

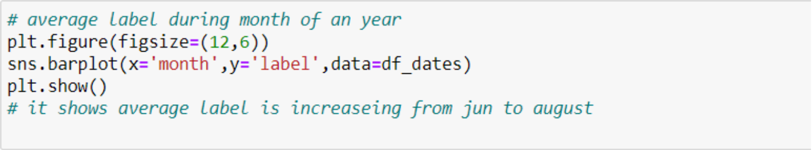
```
Model=[]
score=[]
CVS=[]
rocscore=[]
for name,model in models:
    print(name)
    print('\n')
    Model.append(name)
    model.fit(x_train,y_train)
    print(model)
    pre=model.predict(x_test)
    print('\n')
    AS=accuracy_score(y_test,pre)
    print('Accuracy_score=',AS)
    score.append(AS*100)
    print('\n')
    sc=cross_val_score(model,x,y,cv=10,scoring='accuracy').mean()
    print('Cross_Val_Score=',sc)
    CVS.append(sc*100)
    print('\n')
    false_positive_rate,true_positive_rate,threshold=roc_curve(y_test,pre)
    roc_auc= auc(false_positive_rate,true_positive_rate)
    print('roc_auc_score=',roc_auc)
    rocscore.append(roc_auc*100)
    print('\n')
    print('classification_report\n',classification_report(y_test,pre))
    print('\n')
    cm=confusion_matrix(y_test,pre)
    print(cm)
    print('\n')
    plt.figure(figsize=(10,40))
    plt.subplot(911)
    plt.title(name)
    plt.plot(false_positive_rate,true_positive_rate,label='AUC = %0.2f'% roc_auc)
    plt.plot([0,1],[0,1],'r--')
    plt.legend(loc='lower right')
    plt.ylabel('True positive Rate')
    plt.xlabel('False Positive Rate')
    print('\n\n')
```

- Key Metrics for success in solving problem under consideration

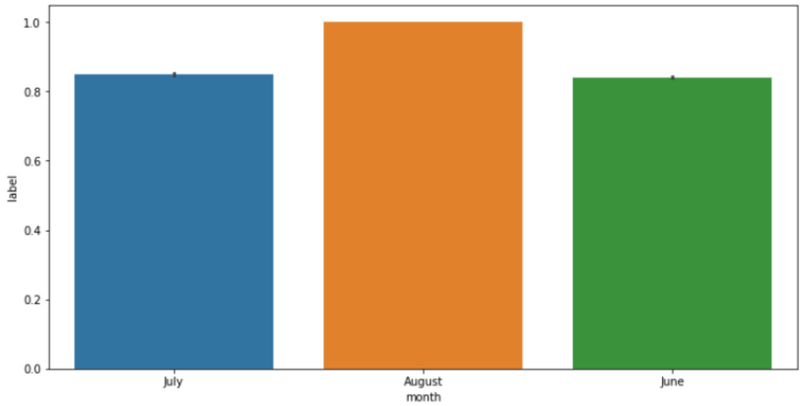
  As we know that our dataset is imbalance so we do not much focus upon the accuracy score. We mainly see the precision and recall value of our model.

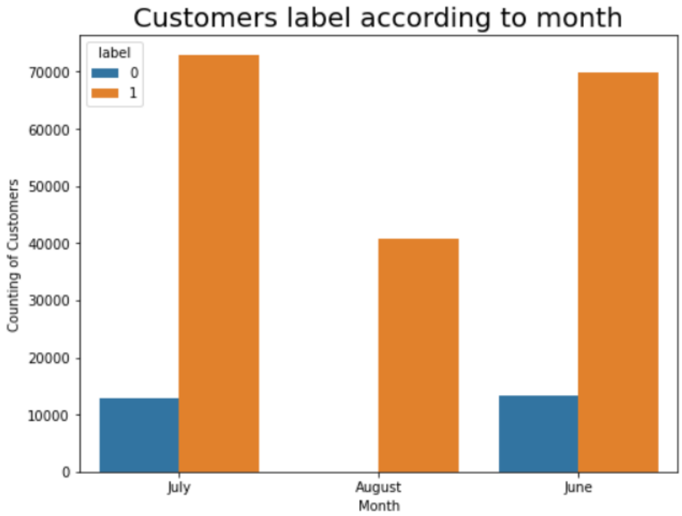
Precision talks about all the correct predictions out of total positive predictions. Recall means how many individuals were classified correctly out of all the actual positive individuals.
Along that we also consider AUC ROC score as our key matrix. Because Area under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

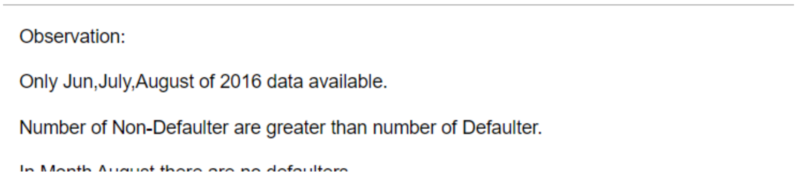
- Visualization



```
# average label during month of an year
plt.figure(figsize=(12,6))
sns.barplot(x='month',y='label',data=df_dates)
plt.show()
# it shows average label is increaseing from jun to august
```







Observation:

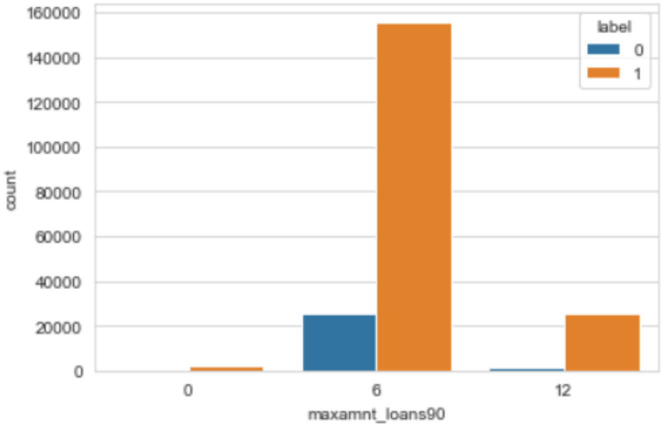
Only Jun,July,August of 2016 data available.

Number of Non-Defaulter are greater than number of Defaulter.

In Month August there are no defaulters

```
# Lets check the count of the maximum loan amount in 90 days taking by users alo
sns.set_style('whitegrid')
sns.countplot(x='maxamnt_loans90',hue='label',data=df)
```

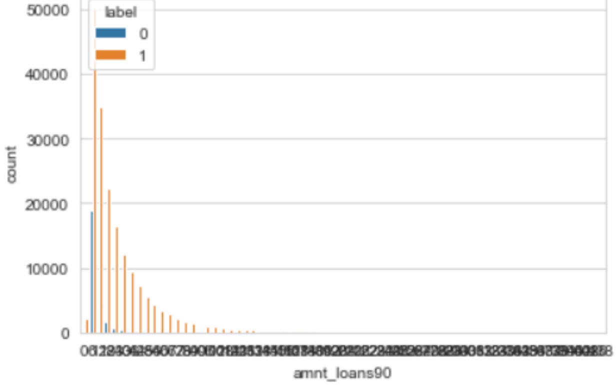
```
<AxesSubplot:xlabel='maxamnt_loans90', ylabel='count'>
```



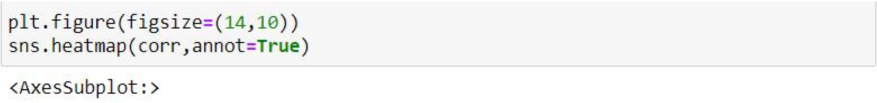
We can observe that Maximun users take 6 rupiah and few take 12 rupiah loan in last 90 days.And there are 12.5% defaulters.

```
sns.set_style('whitegrid')
sns.countplot(x='amnt_loans90',hue='label',data=df)
```

```
<AxesSubplot:xlabel='amnt_loans90', ylabel='count'>
```

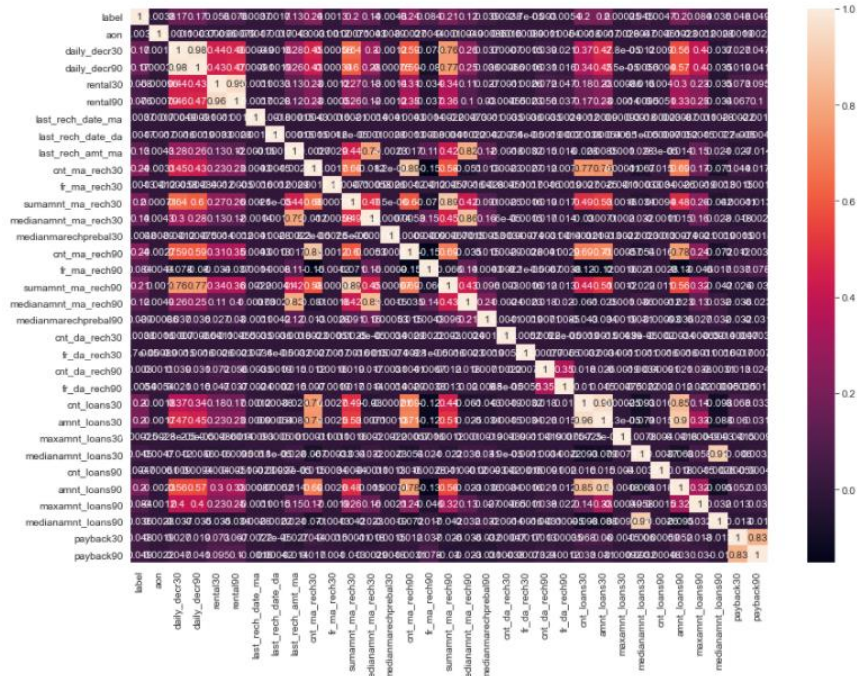


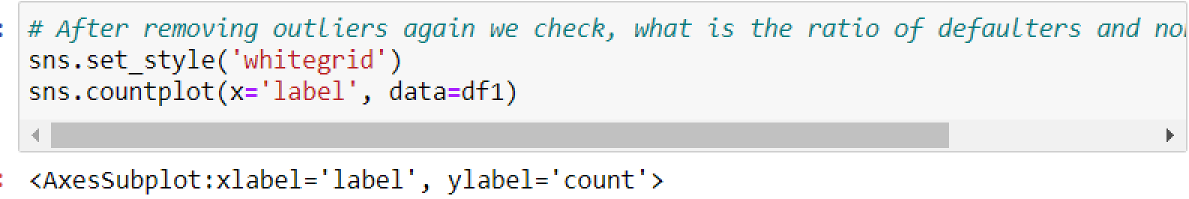
Here also we can see that Maximun users take 6 rupiah loan and most os the defaulters are 6 rupiah loan takers.



```
plt.figure(figsize=(14,10))
sns.heatmap(corr,annot=True)
```

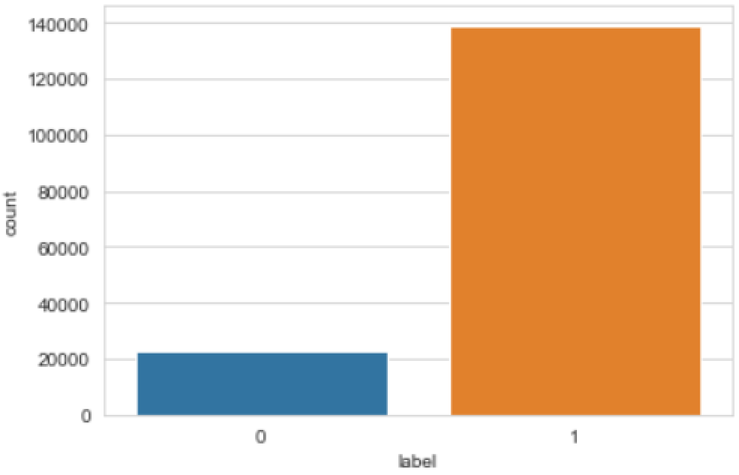
```
<AxesSubplot:>
```





```
# After removing outliers again we check, what is the ratio of defaulters and no
sns.set_style('whitegrid')
sns.countplot(x='label', data=df1)
```

```
<AxesSubplot:xlabel='label', ylabel='count'>
```



Shape of Only Numerical Column After Outliers Removed

## CONCLUSION

- Key Findings and Conclusions of the Study

There are no Null value.

The dataset is imbalanced. Label ‘1’ has approximately 86% records, while, label ‘0’ has approximately 14% records.

maxamnt_loans90 columns gives information about customers with no loan history.

msisdn feature some values which might not be realistic. So drop the row which contain not realistic value.

There are some rows which is repeated means duplicate entries are present in our dataset.

The collected data is only for one area circle(UPW).

## • Learning Outcomes of the Study in respect of Data Science

Here I learned about the micro credit industry, visualization, data cleaning, handling outliers and using various algorithms on huge dataset. This was the first time I worked on such huge dataset. It took a lot of time to train all the algorithms to find out the best one to work with. Working with such huge dataset that took a lot of time to train the algorithms and tuning it for the best prams was worth knowing in this project.
And I also know that the best way to finalize the model is trying every possibility with patience

## • Limitations of this work and Scope for Future Work

As the data set content more than 20% of Outliers we can not remove than so that is a big limitation. If outliers were removed we can build more efficient model.

If there were only one month data available we can tune the data more efficiently.

Here I learned about the micro credit industry, visualization, data cleaning, handling outliers and using various algorithms on huge dataset. This was the first time I worked on such huge dataset. It took a lot of time to hyper tune all the algorithms to find out the best one to work with. Working with such huge dataset that took a lot of time to train the algorithms and tuning it for the best parameter was worth knowing in this project.

# THANK YOU