FLIP ROBO

Micro-Credit Defaulter Model

Submitted by:

YASH JAISWAL

ACKNOWLEDGMENT

I would like to thank you Flip Robo Technologies for providing me with the opportunity to work on this project from which I have learned a lot

INTRODUCTION

- Business Problem Framing

Loan are one of the necessary need of each and every person around the globe and therefore Micro loan Focusing on changing trends in loan sales and purchases predictive modelling market mix modelling Using machine learning in order to predict the actual values of the prospective and decide whether to amount to the loan.

- Conceptual Background of the Domain Problem

Micro credit defaulter model and problem patterns making sure last-minute purchases are loan Keeping the loan as full as they want it raising prices on a rate of loan which is filling up in order to reduce rate of loan and hold back inventory for those expensive lastminute expensive purchases This usually happens as an attempt to maximize revenue based on Micro credit defaulter

- Review of Literature

This is a comprehensive summary of the research done on the behalf You have to data set at least 209593 rows of data. You can data set more data as well, it’s up to you, More the data better the model In this section you have to loan of the data of micro crdit from different websites

- Motivation for the Problem Undertaken

Micro credit defaulter model problem are likely to rate of less loan of the data set Time of purchase patterns making sure last-minute purchases are loan Keeping the micro credit as full as they want it raising prices on a loan which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases loan of the rate of intrest

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

We are building a model in machine learning to predict the actual value of the prospective properties and decide whether to invest in them or not. So this model will help us to determine which variable are import predict the label of variable and also how these variable describe the label of the Micro credit defaulter.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209593 entries, 0 to 209592
Data columns (total 37 columns):
 #    Column                 Non-Null Count   Dtype
---   ------                 --------------   -----
 0    Unnamed: 0             209593 non-null  int64
 1    label                  209593 non-null  int64
 2    msisdn                 209593 non-null  object
 3    aon                    209593 non-null  float64
 4    daily_decr30           209593 non-null  float64
 5    daily_decr90           209593 non-null  float64
 6    rental30               209593 non-null  float64
 7    rental90               209593 non-null  float64
 8    last_rech_date_ma      209593 non-null  float64
 9    last_rech_date_da      209593 non-null  float64
 10   last_rech_amt_ma       209593 non-null  int64
 11   cnt_ma_rech30          209593 non-null  int64
 12   fr_ma_rech30           209593 non-null  float64
 13   sumamnt_ma_rech30      209593 non-null  float64
 14   medianamnt_ma_rech30   209593 non-null  float64
 15   medianmarechprebal30   209593 non-null  float64
```

- Data Sources and their formats

```
df.describe()
```

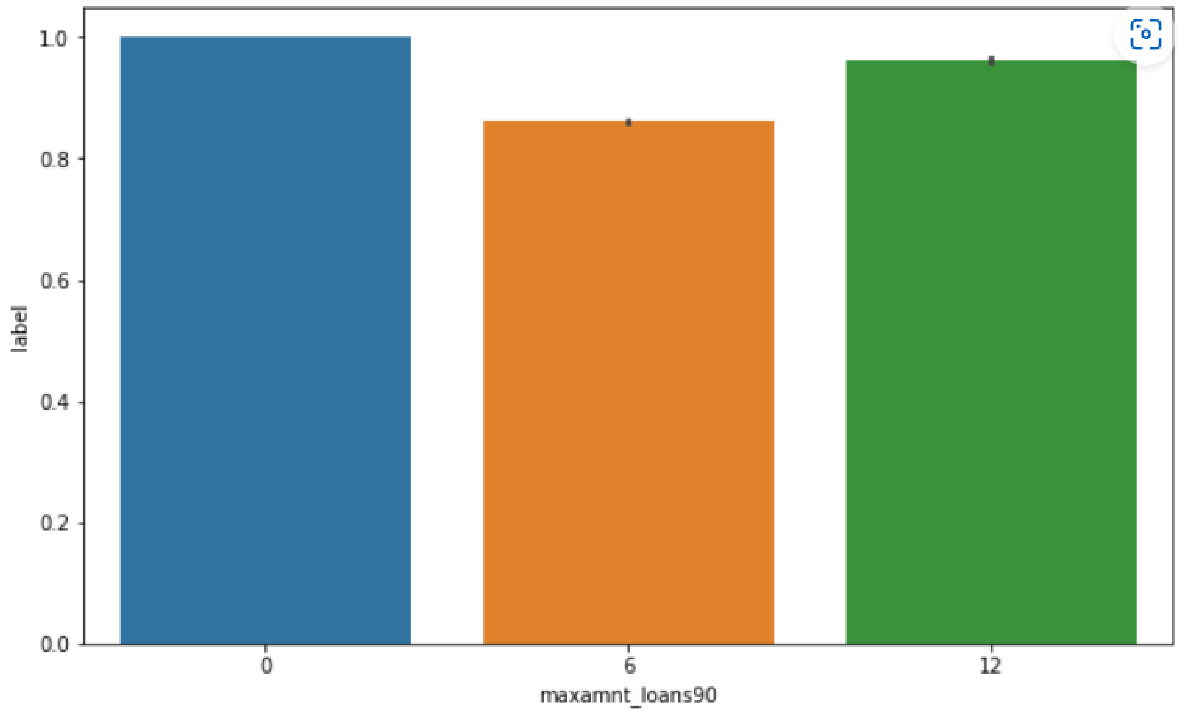
	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da
count	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000
mean	0.875177	8112.343445	5381.402289	6082.515068	2692.581910	3483.406534	3755.847800	3712.202921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

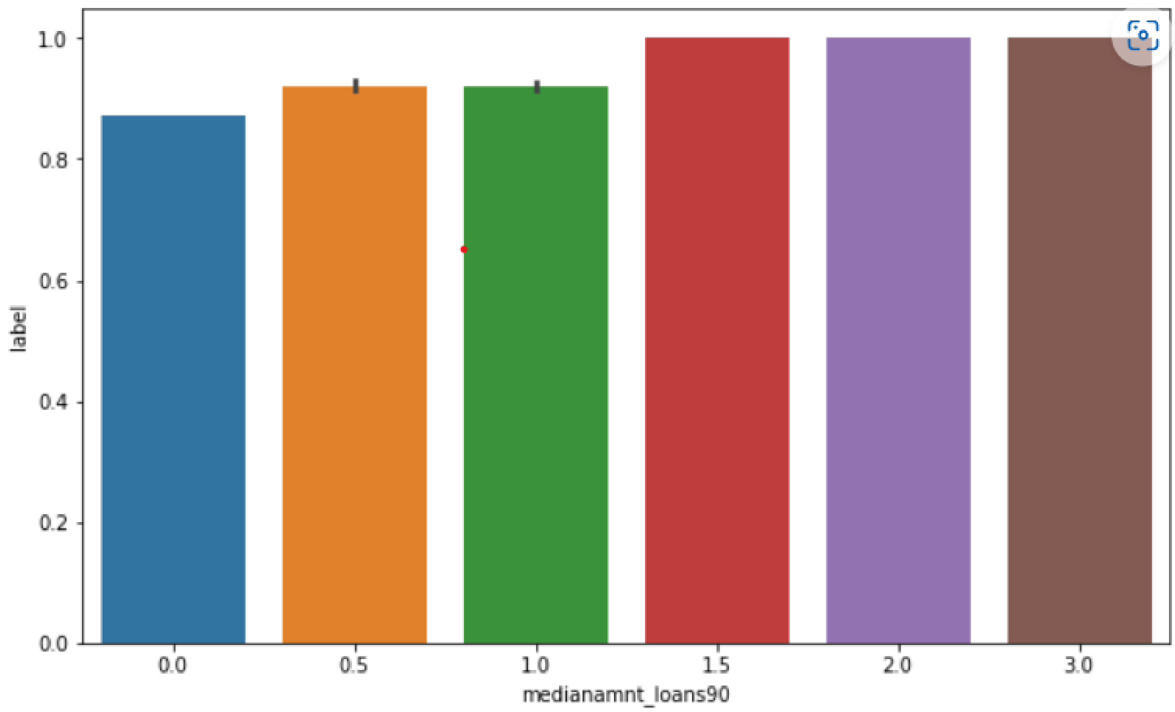
8 rows × 36 columns

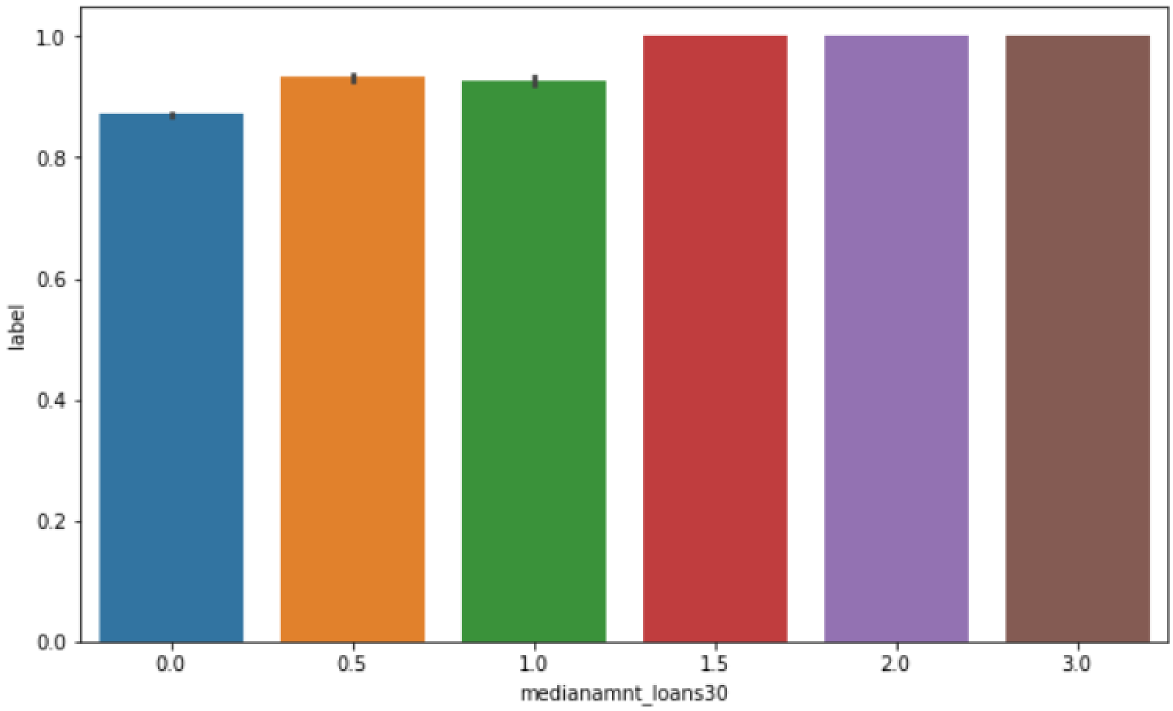
- Data Preprocessing Done

Loading the training data set as a dataframe , used pandas to set display I ensuring we do not see any truncated information , checked the number of rows and columns present in our training data set , checked for missing data and the number of rows with null values , verified the percentage of missing data in each columns are decide to dicard the once that value more than , dropped all the unwanted columns are duplicated data present in our data frame, separated categorical columns and numeric columns name in separate list variable for ease in visulazation , checked the unique values information in each column to get a gist for categorical data. Used pandas profiling during the visulazing phase along with pie plot count plot scatter plot and the other , with the help of label encoding technique converted all object data type columns to numeric data types.

- Data Inputs- Logic- Output Relationships

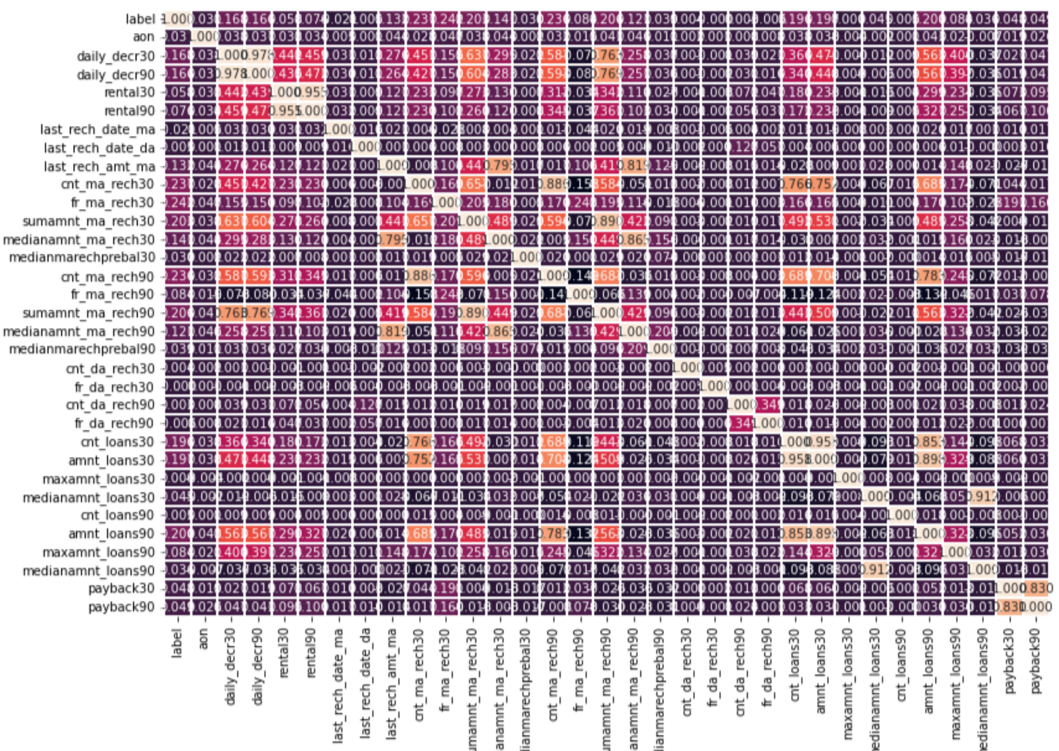

1.0
0.8
0.6
0.4
0.2
0.0
label
0
6
12
maxamnt_loans90

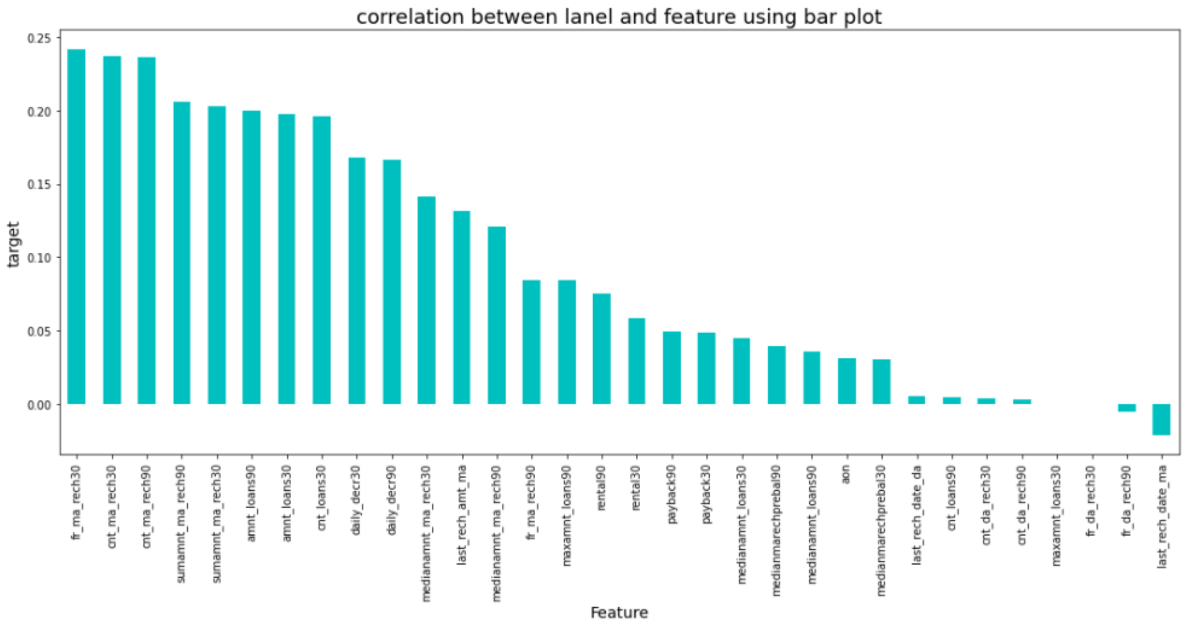

1.0
0.8
0.6
0.4
0.2
0.0
label
0.0
0.5
1.0
1.5
2.0
3.0
medianamnt_loans90



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

	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	0.031059	0.168298	0.166150	0.058085	0.075521	-0.021145	0.005148	0.131804	0
aon	0.031059	1.000000	0.038201	0.036621	0.032971	0.034296	0.004506	0.000850	0.043888	0
daily_decr30	0.168298	0.038201	1.000000	0.977704	0.442066	0.458977	0.030848	0.012523	0.275837	0
daily_decr90	0.166150	0.036621	0.977704	1.000000	0.434685	0.471730	0.029938	0.011449	0.264131	0
rental30	0.058085	0.032971	0.442066	0.434685	1.000000	0.955237	0.031032	0.009448	0.127271	0
rental90	0.075521	0.034296	0.458977	0.471730	0.955237	1.000000	0.031689	0.008932	0.121416	0
last_rech_date_ma	-0.021145	0.004506	0.030848	0.029938	0.031032	0.031689	1.000000	0.016242	0.020869	0
last_rech_date_da	0.005148	0.000850	0.012523	0.011449	0.009448	0.008932	0.016242	1.000000	0.000945	0
last_rech_amt_ma	0.131804	0.043888	0.275837	0.264131	0.127271	0.121416	0.020869	0.000945	1.000000	-0
cnt_ma_rech30	0.237331	0.028098	0.451385	0.426707	0.233343	0.230260	0.005913	0.005890	-0.002662	1
fr_ma_rech30	0.241959	0.047073	0.155130	0.150488	0.097296	0.102007	-0.022697	0.008408	0.103657	0
sumamnt_ma_rech30	0.202828	0.037597	0.636536	0.603886	0.272649	0.259709	0.008209	0.002992	0.440821	0
medianamnt_ma_rech30	0.141490	0.044464	0.295356	0.282960	0.129853	0.120242	0.004108	0.000170	0.794646	-0
medianmarechprebal30	0.030221	0.002183	0.021568	0.021540	0.006849	0.007847	0.000937	0.003049	0.017312	0
cnt_ma_rech90	0.236392	0.032304	0.587338	0.593069	0.312118	0.345293	0.017390	0.006332	0.016707	0
fr_ma_rech90	0.084385	0.015462	-0.078299	-0.079530	-0.033530	-0.036524	-0.044023	0.002382	0.106267	-0
sumamnt_ma_rech90	0.205793	0.041187	0.762981	0.768817	0.342306	0.360601	0.020217	0.004588	0.418735	0
medianamnt_ma_rech90	0.120855	0.045527	0.257847	0.250518	0.110356	0.103151	0.018941	0.000235	0.818734	-0





- Hardware and Software Requirements and Tools Used

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

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

:

```python
x=df.drop(columns='label')#Feature
y=df.label#Target
```

```python
#Lets import standardscaler
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x_scaled=scaler.fit_transform(x)
x_scaled
```

	features	vif
0	aon	1.006505
1	daily_decr30	29.106991
2	daily_decr90	31.999107
3	rental30	13.142587
4	rental90	13.813196
5	last_rech_date_ma	1.006197
6	last_rech_date_da	1.017277
7	last_rech_amt_ma	3.440456
8	cnt_ma_rech30	14.993557
9	fr_ma_rech30	1.211726
10	sumamnt_ma_rech30	12.707707

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

I have used both statistical and analytical approaches to solve the problem which mainly inculd the pre-processing of the data and EDA to check the correlation of independent and dependent features also before building the model I made sure that the point data was cleaned and claed before it was fed into machine learing models for this project we need to predict the loan label meaning our target columns is continue so this is a Logistic problem I have used various classification algorthim I have sekect random forest classification as the best suitalble algorithm for our final models as it is giving a good and least different in and cv-score among all the algorithm used other classification algorithm are also giving accuracy but some are over fitting and some are under fitting the result which may be because of less label performance as well as accuracy and to check my model from overfitting and under fitting I have made use of the k fold and then hyper tuning the final model once I was able to get my desired final model I ensured to save that models i before i loaded the testing data and stared performance the data as training data set and obtaining the predicted label out of the classification machine learning model.

- Testing of Identified Approaches (Algorithms)

1) logistic regression
2) decision tree
3) xg boost
4) adaboost
5) random forest

Here we select Random Forest Tree Classification for the model building.

```
grid_param={
    'criterion':['ginni','entropy'],
    'max_depth': range(10,15),
    'min_samples_leaf':range(2,6),
    'min_samples_split':range(3,8),
    'max_leaf_nodes':range(5,10)}
```

```
grid_search=GridSearchCV(estimator=rf,
                         param_grid=grid_param,
                         cv=5,
                        n_jobs=-1)
```

```
grid_search.fit(x_train,y_train)
```

```
cnn.score(x_train,y_train)
```

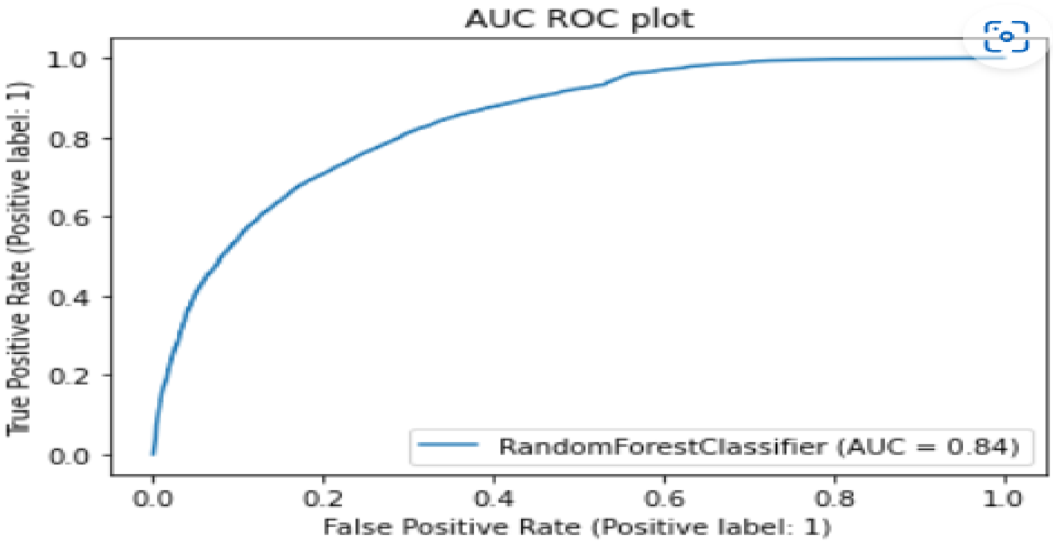
0.9013256994853969

```
cnn.score(x_test,y_test)
```

0.9010305671299977

- Visualizations

```
from sklearn.metrics import plot_roc_curve
plot_roc_curve(cnn, x_test, y_test)
plt.title("AUC ROC plot")
plt.show()
```



CONCLUSION

- Key Findings and Conclusions of the Study

Post models building and choosing the appropriate model I want ahead and scrape the data and join the dataset. After applying all the data pre processing steps as the dataset I was then able to get the predicted label result. Once the dataset with feature columns are predicted label was format I exported the value in a comma separated value file to be accessed as needed.

Conclusion

```
loaded_model=pickle.load(open('Micro_Credit_Defaulter','rb'))
result=loaded_model.score(x_test,y_test)
print(result*100)
```

```
91.19882947930914
```