FLIP ROBO

MICRO-CREDIT DEFAULTER PROJECT

Submitted by:

Priya Rajput

ACKNOWLEDGMENT

I would like to express my gratitude towards Flip-Robo for providing me this opportunity to show case my talent and also for their constant support and guidance. Also It is indeed a pleasure for me to have worked on this project.

I express my deepest thanks to Miss Sapna Verma, for taking part in useful decision & giving necessary advices and guidance and arranged all facilities to make my project easier. I choose this moment to acknowledge his contribution gratefully.

TABLE OF CONTENTS

1. **Introduction**
   i. Business Problem Framing
   ii. Conceptual Background of the Domain Problem
   iii. Review of literature
   iv. Motivation for the Problem Undertaken

2. **Analytical Problem Framing**
   i. Mathematical/ Analytical Modelling of the Problem
   ii. Data Sources and their formats
   iii. Data Pre-processing Done
   iv. Hardware & Software Requirements & Tools Used

3. **Model/s Development and Evaluation**
   i. Identification of possible problem-solving approaches (methods)
   ii. Visualizations

4. **Conclusions**
   i. Conclusions of the Study
   ii. Limitations of this work and Scope for Future Work

INTRODUCTION

Business Problem Framing

In this project, we have to predict micro credit defaulters using new machine learning models using the dataset. 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 is very useful in such areas, the implementation of MFS has been uneven with both significant challenges and successes.

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.

Conceptual Background of the Domain Problem

1. Collect the data.
2. Check whether the project is a regression type or a classification type.
3. Check whether our dataset is balanced or imbalanced. If it is an imbalanced one, we will apply sampling techniques to balance the dataset.
4. Model building and find the accuracy of the model.
5. Build a model with good accuracy and also go for hyperparameter tuning.

Review of Literature

To predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of insurance of loan. In this case, Label ‘1’ indicates that the loan has been paid i.e., Non- defaulter, while, Label ‘0’ indicates that the loan has not been paid i.e., defaulter.

We have observed that there are no null values present in the dataset given by clients and also there are some customers who don’t have any loan history.

But the data set is quite imbalanced in terms of defaulter and non- defaulters, as the MFI to provide micro-credit on mobile balances to be paid back in 5 days. As per my research, I had seen that most of the users paid the amount within the time frame but if they missed to pay within the time frame of 5 days, they have paid almost like within 7 days, and for that, and I can observe that the client has charged rupiah 6 mostly to the customer.

Motivation

Our main objective of doing this project is to build a model to predict whether the users are paying the loan within the due date or not. We are going to predict by using Machine Learning algorithms.

The sample data is provided to us from our client database. 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.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

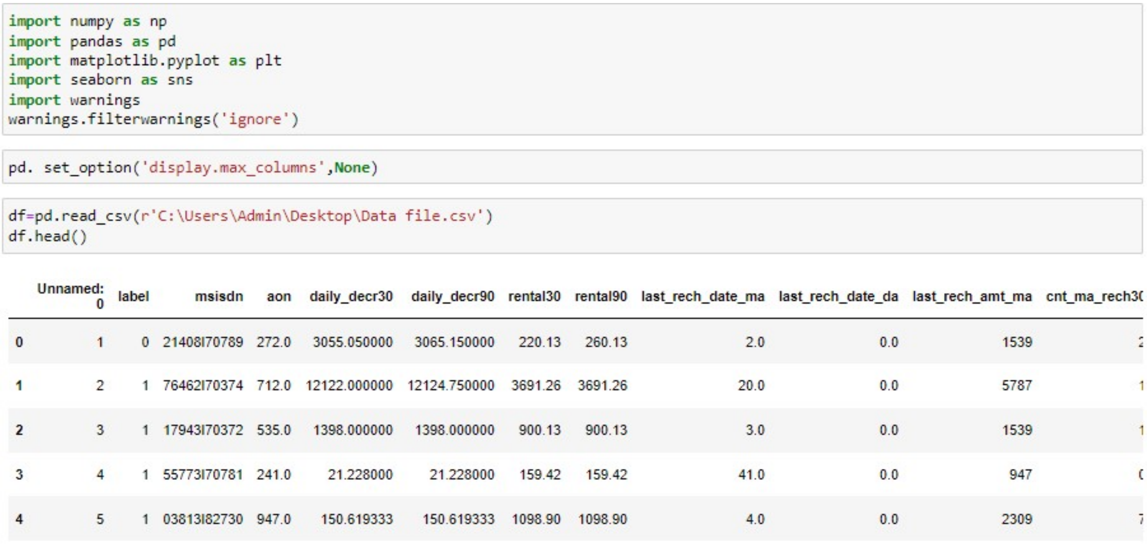
There are various analytics which we have done before moving forward with exploratory analysis, on the basis of accounts which got recharged in the last 30 days. We set the parameter that if the person is not recharging their main account within 3 months, We simply dropped their data because they are not valuable and they might be old customers, but there is no revenue rotating.

Then we had checked the date columns and found that the data belongs to the year 2016. We 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. We 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.After scaling my data, I have sent the data to various classification models and found that Random Forest Classifier Algorithm showing better result.

DATA SOURCES AND THEIR FORMATS

Data Source: The read_csv function of the pandas library is used to read the content of a CSV file into the python environment as a pandas DataFrame. The function can read the files from the OS by using proper path to the file.

Data description: Pandas describe() is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values. When this method is applied to a series of string, it returns a different output which is shown below.

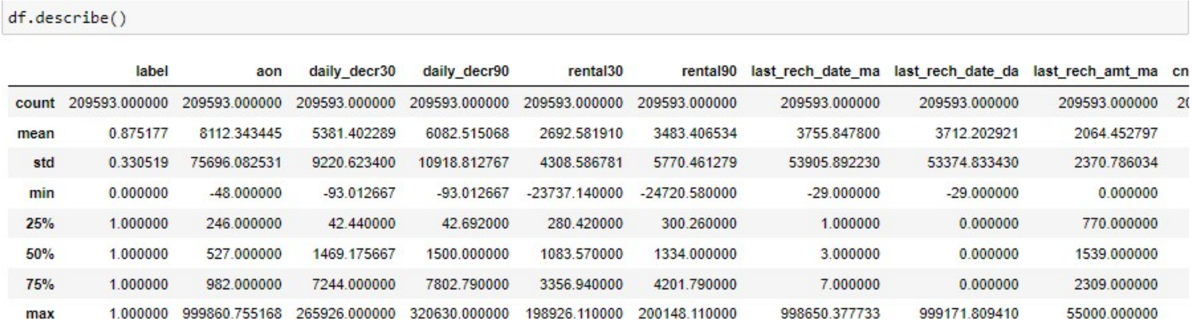


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
pd. set_option('display.max_columns',None)
```

```
df=pd.read_csv(r'C:\Users\Admin\Desktop\Data file.csv')
df.head()
```

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cnt_ma_rech3
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	1539	2
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	5787	1
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	1539	1
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	947	0
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	2309	7



```
df.describe()
```

	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cn
count	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	20
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	

Data Pre-processing Done

- Dropped the column ‘Unnamed:0’ :as it is of no use, it is only giving serial numbers starting from 1 , ’ msisdn’: it tells us about the mobile number of the user which is not so important.
- Checked the value counts of the target variable ’Label’ whether the dataset is balanced or imbalanced.
- Also dropped ‘pcircle’ and ‘pdate’, pcircle is telling us about the telecom circle and pdate is giving us some dates, but these two columns are not playing any significant role.
- Checked the correlation between dependant and independent variables using heatmap. I have seen many columns which are 0% correlated with the target variable. So, I also have dropped all those columns which are not correlated with the target variable ‘Label’.
- Visualization using distribution plots where I can clearly see some outliers and skewness from the plots.
- Checked outliers using boxplots and also removed them using zscore.
- Splitted the dependant and independent variables into x and y.
- Scaled the data using StandardScaler method and made my data ready for model building.

Data Inputs- Logic- Output Relationships

EDA was performed by creating valuable insights using various visualization libraries.

Hardware and Software Requirements and Tools Used

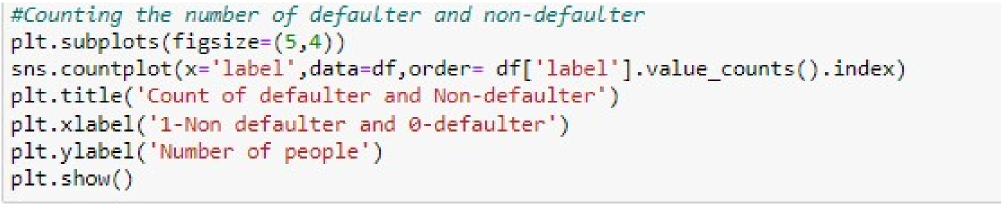
Hardware required:
Processor: core i3
RAM: 8 GB

Software required:

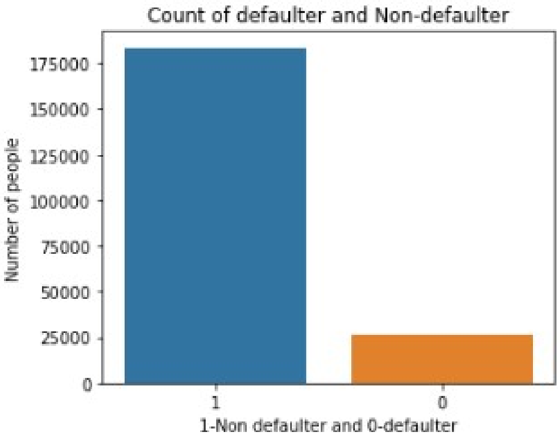
Anaconda 3- language used Python 3
Microsoft Excel

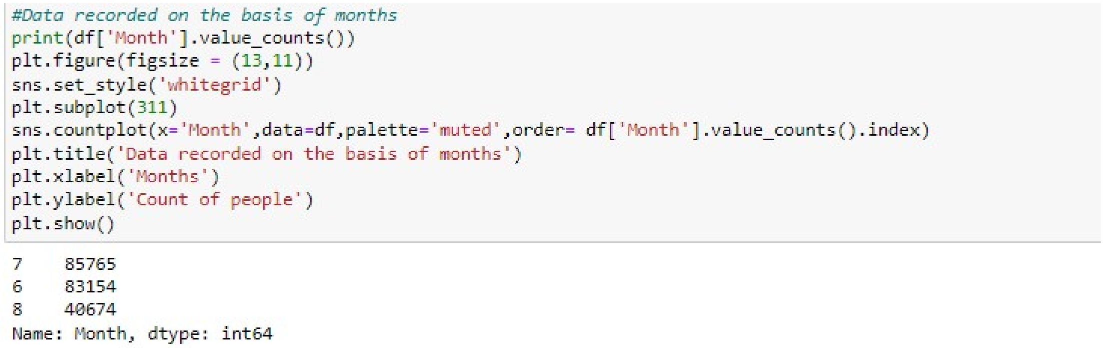
Exploratory Data Analysis

Let us explore our data and visualize



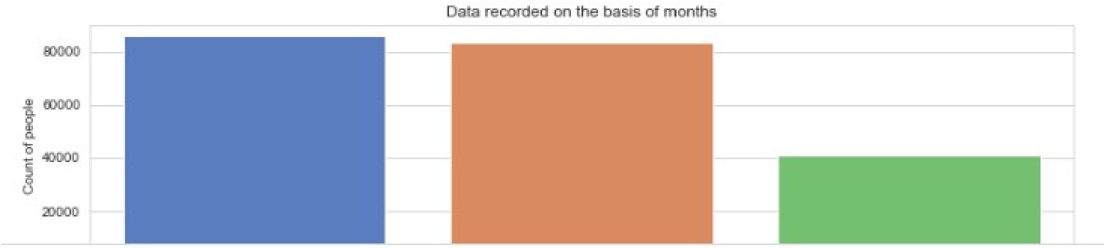
```
#Counting the number of defaulter and non-defaulter
plt.subplots(figsize=(5,4))
sns.countplot(x='label',data=df,order= df['label'].value_counts().index)
plt.title('Count of defaulter and Non-defaulter')
plt.xlabel('1-Non defaulter and 0-defaulter')
plt.ylabel('Number of people')
plt.show()
```

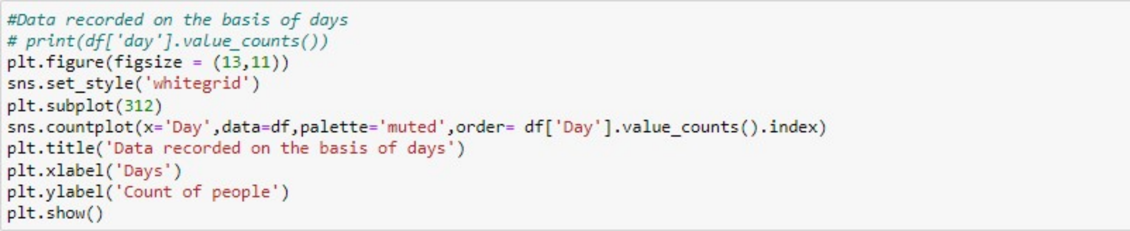




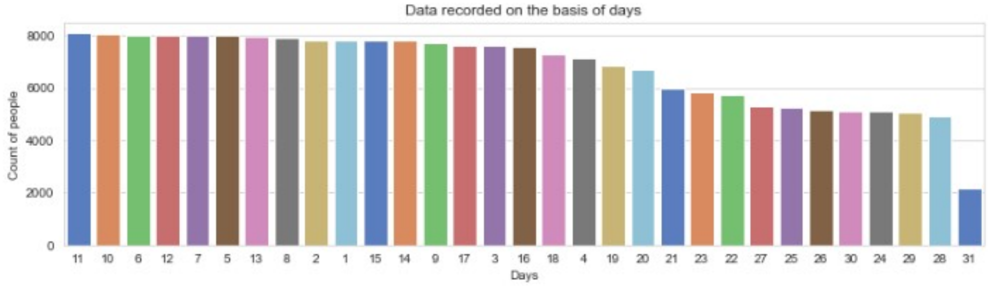
```
#Data recorded on the basis of months
print(df['Month'].value_counts())
plt.figure(figsize = (13,11))
sns.set_style('whitegrid')
plt.subplot(311)
sns.countplot(x='Month',data=df,palette='muted',order= df['Month'].value_counts().index)
plt.title('Data recorded on the basis of months')
plt.xlabel('Months')
plt.ylabel('Count of people')
plt.show()
```

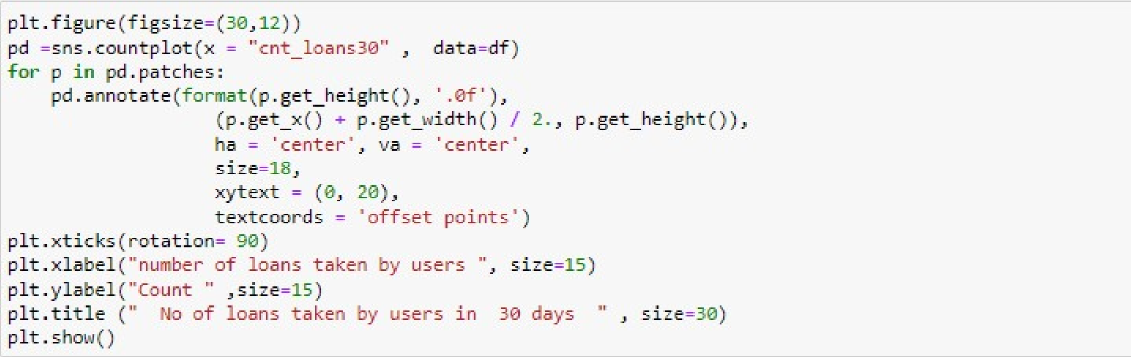
```
7    85765
6    83154
8    40674
Name: Month, dtype: int64
```



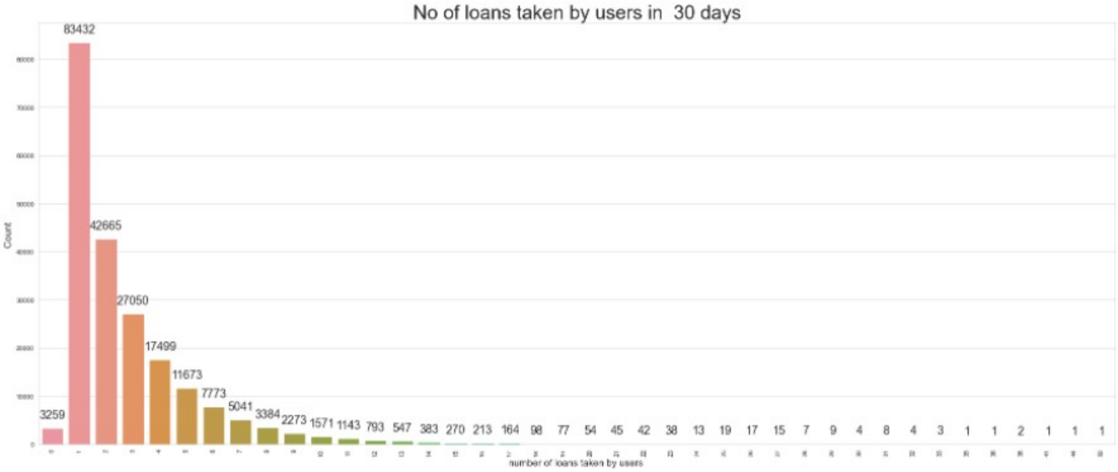


```
#Data recorded on the basis of days
# print(df['day'].value_counts())
plt.figure(figsize = (13,11))
sns.set_style('whitegrid')
plt.subplot(312)
sns.countplot(x='Day',data=df,palette='muted',order= df['Day'].value_counts().index)
plt.title('Data recorded on the basis of days')
plt.xlabel('Days')
plt.ylabel('Count of people')
plt.show()
```

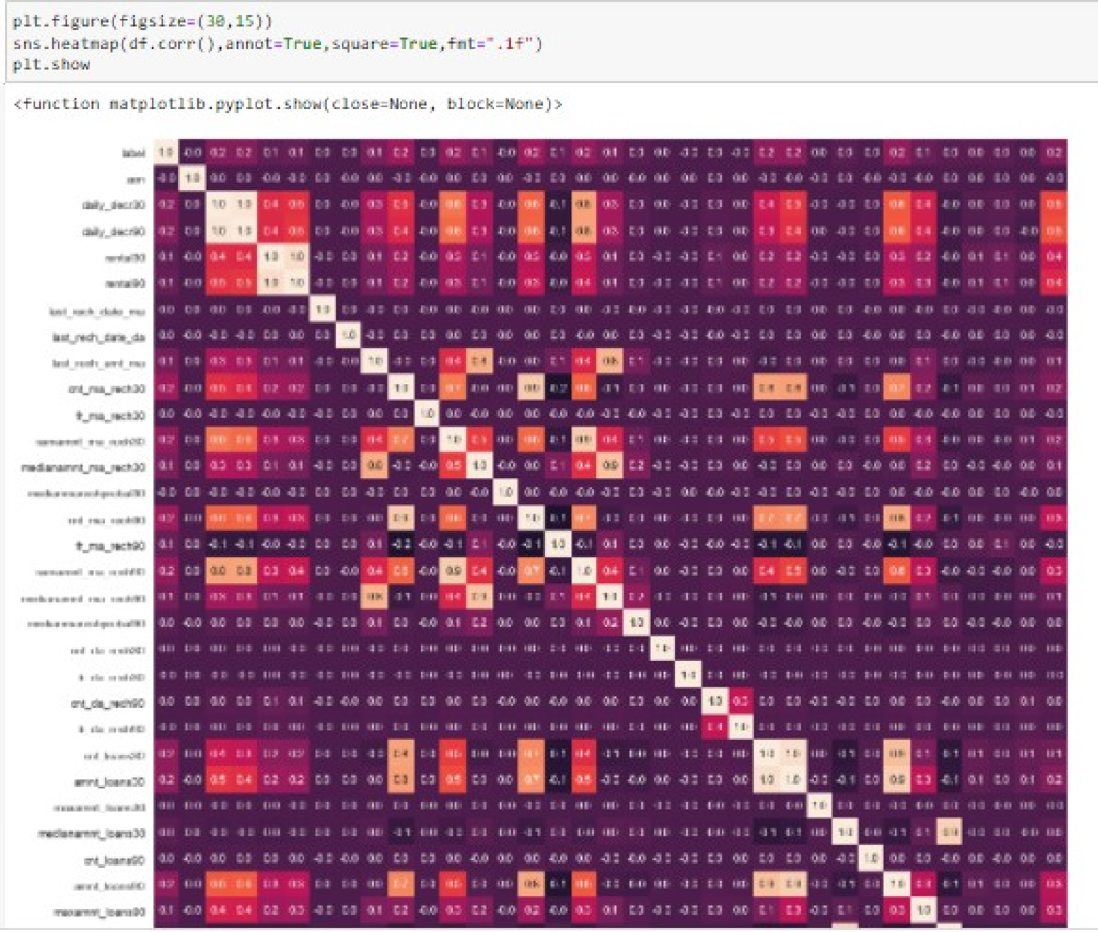




```
plt.figure(figsize=(30,12))
pd =sns.countplot(x = "cnt_loans30" ,  data=df)
for p in pd.patches:
    pd.annotate(format(p.get_height(), '.0f'),
                   (p.get_x() + p.get_width() / 2., p.get_height()),
                   ha = 'center', va = 'center',
                   size=18,
                   xytext = (0, 20),
                   textcoords = 'offset points')
plt.xticks(rotation= 90)
plt.xlabel("number of loans taken by users ", size=15)
plt.ylabel("Count " ,size=15)
plt.title ("  No of loans taken by users in  30 days  " , size=30)
plt.show()
```



Heatmap

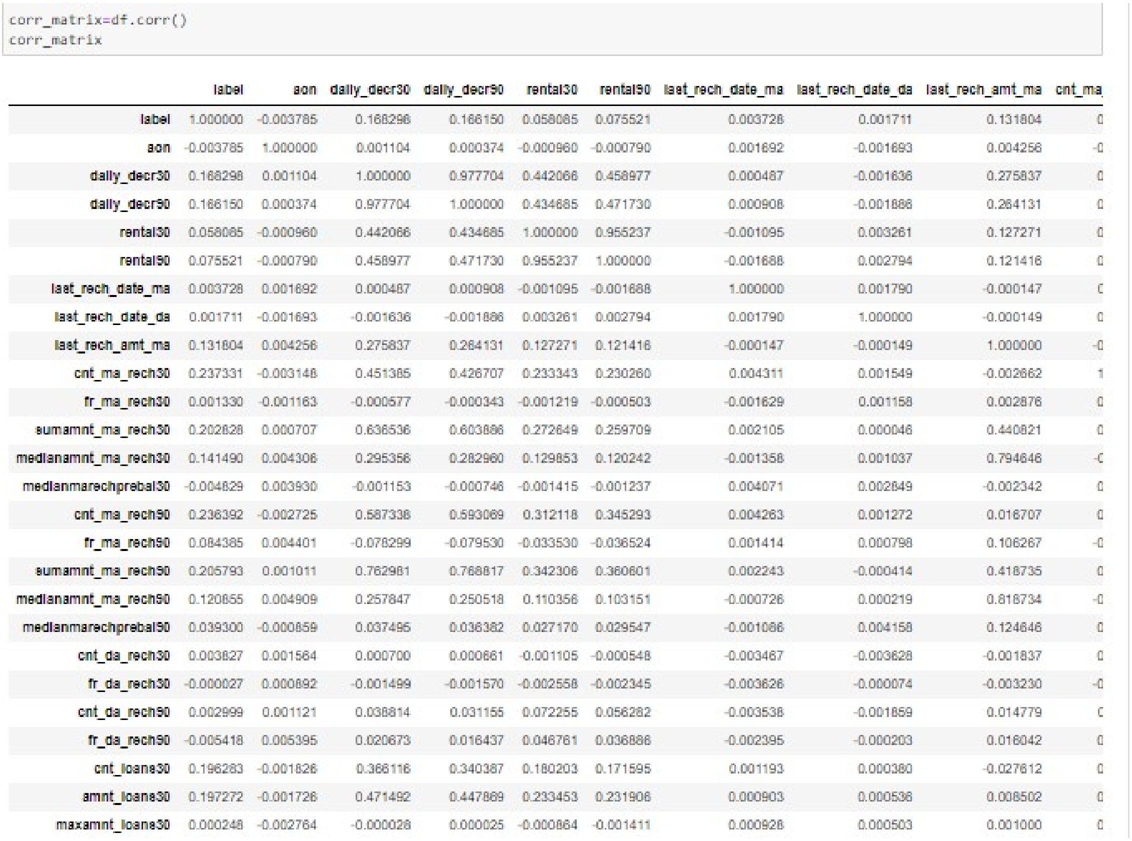


```
plt.figure(figsize=(30,15))
sns.heatmap(df.corr(),annot=True,square=True,fmt=".1f")
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

Correlation matrix

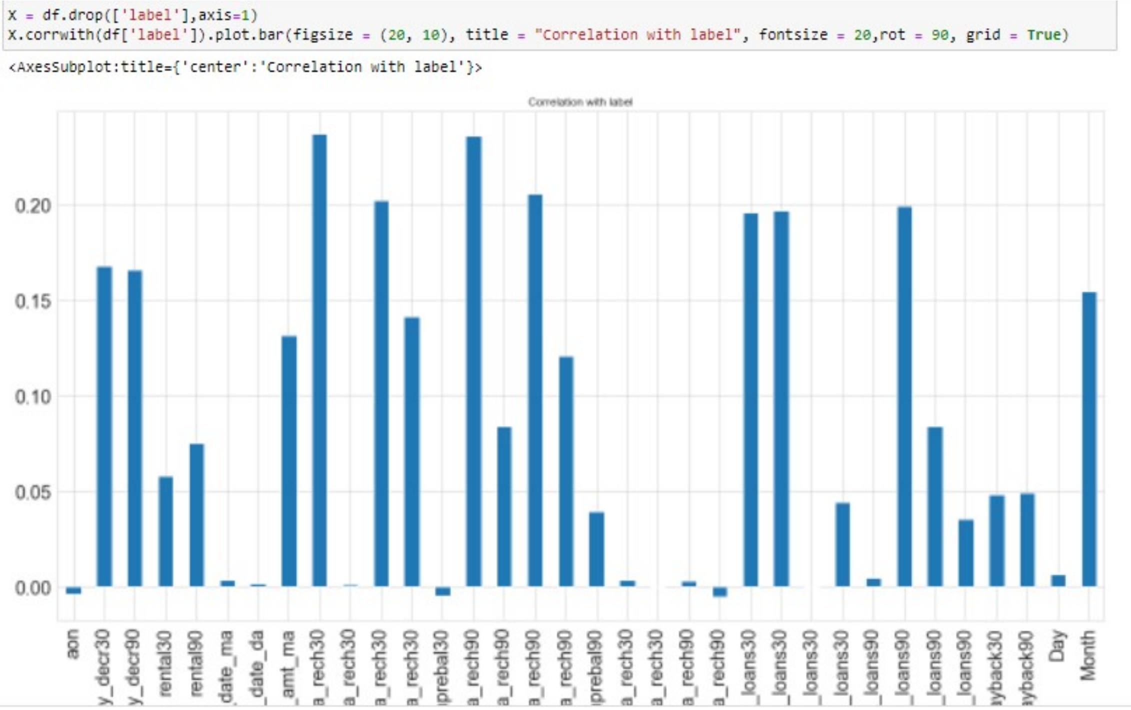
A correlation matrix is simply a table which displays the correlation. The measure is best used in variables that demonstrate a linear relationship between each other. The fit of the data can be visually represented in a heatmap.



```
corr_matrix=df.corr()
corr_matrix
```

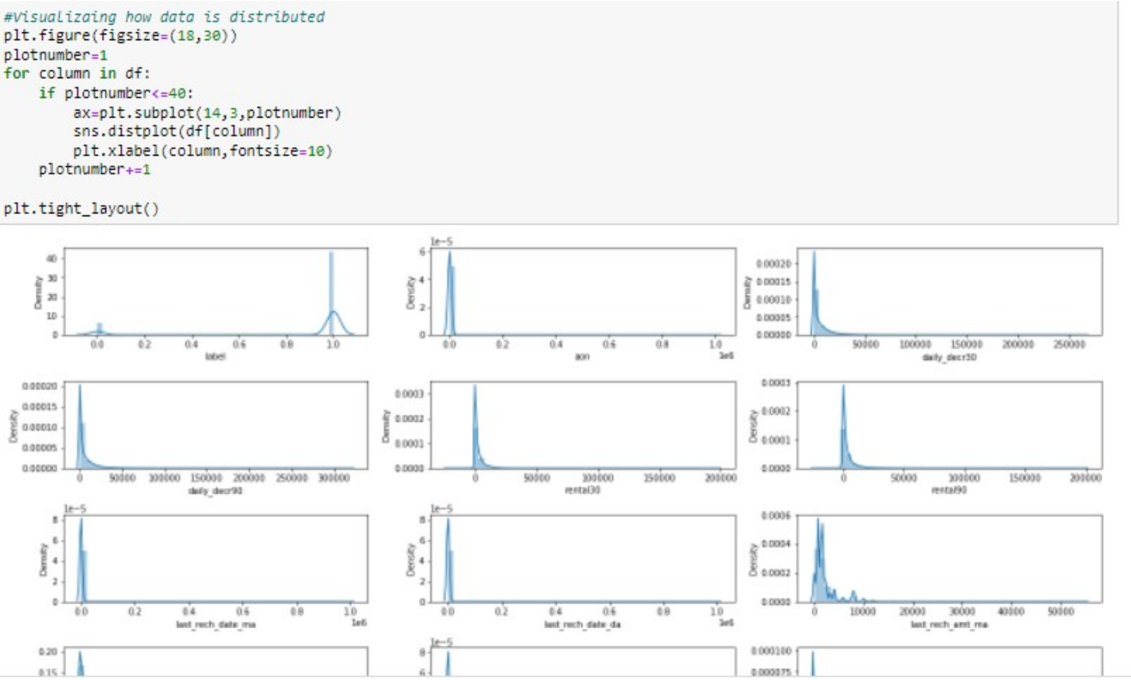
	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cnt_ma
label	1.000000	-0.003785	0.168298	0.166150	0.058085	0.075521	0.003728	0.001711	0.131804	0
aon	-0.003785	1.000000	0.001104	0.000374	-0.000960	-0.000790	0.001692	-0.001693	0.004256	-0
daily_decr30	0.168298	0.001104	1.000000	0.977704	0.442066	0.458977	0.000487	-0.001636	0.275837	0
daily_decr90	0.166150	0.000374	0.977704	1.000000	0.434685	0.471730	0.000908	-0.001886	0.264131	0
rental30	0.058085	-0.000960	0.442066	0.434685	1.000000	0.955237	-0.001095	0.003261	0.127271	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	0.003261	0.002794	0.001790	1.000000	-0.000149	0
last_rech_amt_ma	0.131804	0.004256	0.275837	0.264131	0.127271	0.121416	-0.000147	-0.000149	1.000000	-0
cnt_ma_rech30	0.237331	-0.003148	0.451385	0.426707	0.233343	0.230260	0.004311	0.001549	-0.002662	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	0.141490	0.004306	0.295356	0.282960	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	0.236392	-0.002725	0.587338	0.593069	0.312118	0.345293	0.004263	0.001272	0.016707	0
fr_ma_rech90	0.084385	0.004401	-0.078299	-0.079530	-0.033530	-0.036524	0.001414	0.000798	0.106267	-0
sumamnt_ma_rech90	0.205793	0.001011	0.762981	0.768817	0.342306	0.360601	0.002243	-0.000414	0.418735	0
medianamnt_ma_rech90	0.120855	0.004909	0.257847	0.250518	0.110356	0.103151	-0.000726	0.000219	0.818734	-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	0.000661	-0.001105	-0.000548	-0.003467	-0.003628	-0.001837	0
fr_da_rech30	-0.000027	0.000892	-0.001499	-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	0.366116	0.340387	0.180203	0.171595	0.001193	0.000380	-0.027612	0
amnt_loans30	0.197272	-0.001726	0.471492	0.447869	0.233453	0.231906	0.000903	0.000536	0.008502	0
maxamnt_loans30	0.000248	-0.002764	-0.000028	0.000025	-0.000864	-0.001411	0.000928	0.000503	0.001000	0

This is the heatmap where I have checked the correlation between the data and also got to know that there are columns or independent variables which are 0% correlated with the target variable.



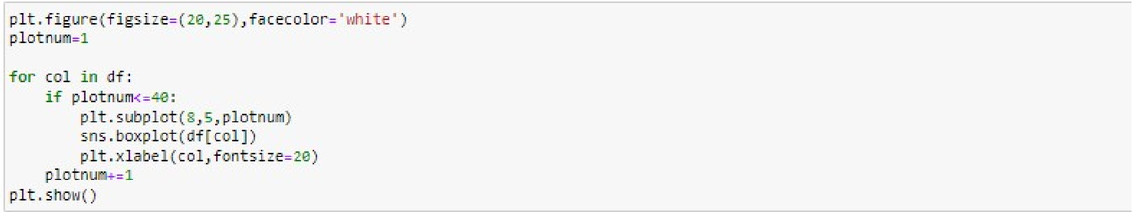
```
X = df.drop(['label'],axis=1)
X.corrwith(df['label']).plot.bar(figsize = (20, 10), title = "Correlation with label", fontsize = 20,rot = 90, grid = True)
```

```
<AxesSubplot:title={'center':'Correlation with label'}>
```



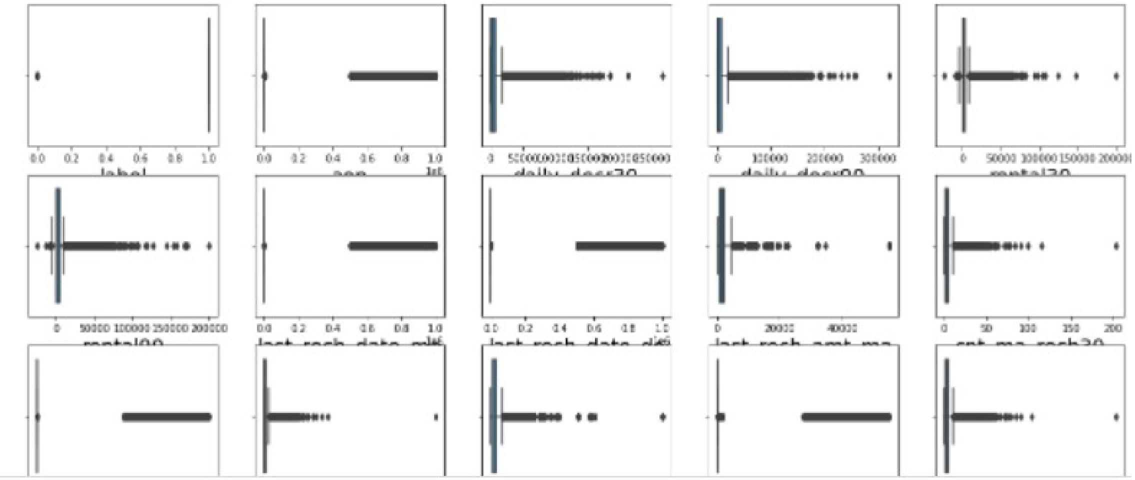
```
#Visualizaing how data is distributed
plt.figure(figsize=(18,30))
plotnumber=1
for column in df:
    if plotnumber<=40:
        ax=plt.subplot(14,3,plotnumber)
        sns.distplot(df[column])
        plt.xlabel(column,fontsize=10)
    plotnumber+=1

plt.tight_layout()
```

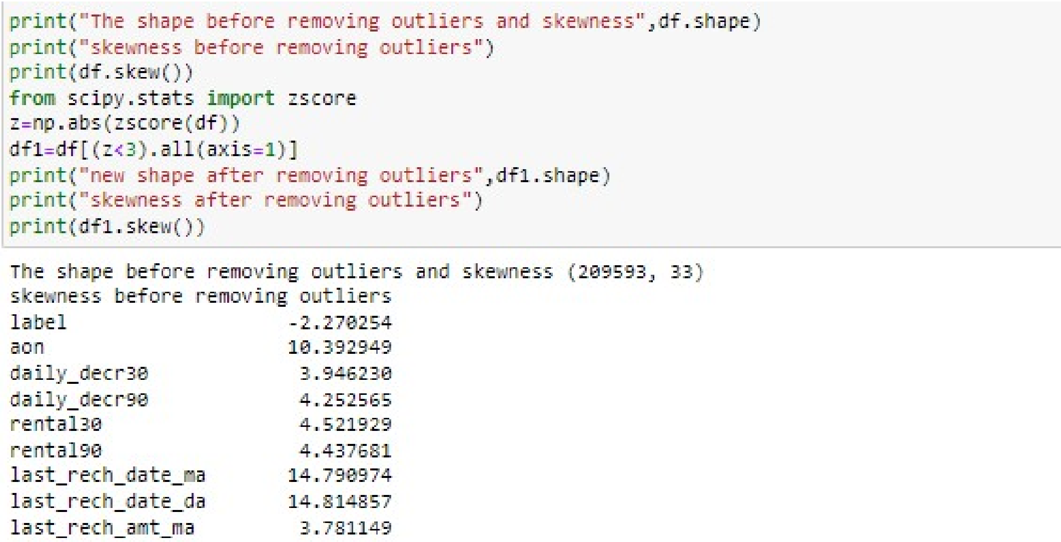


```
plt.figure(figsize=(20,25),facecolor='white')
plotnum=1

for col in df:
    if plotnum<=40:
        plt.subplot(8,5,plotnum)
        sns.boxplot(df[col])
        plt.xlabel(col,fontsize=20)
    plotnum+=1
plt.show()
```



The above screenshot is of the boxplots where we can clearly see there are lots of outliers present in our data.



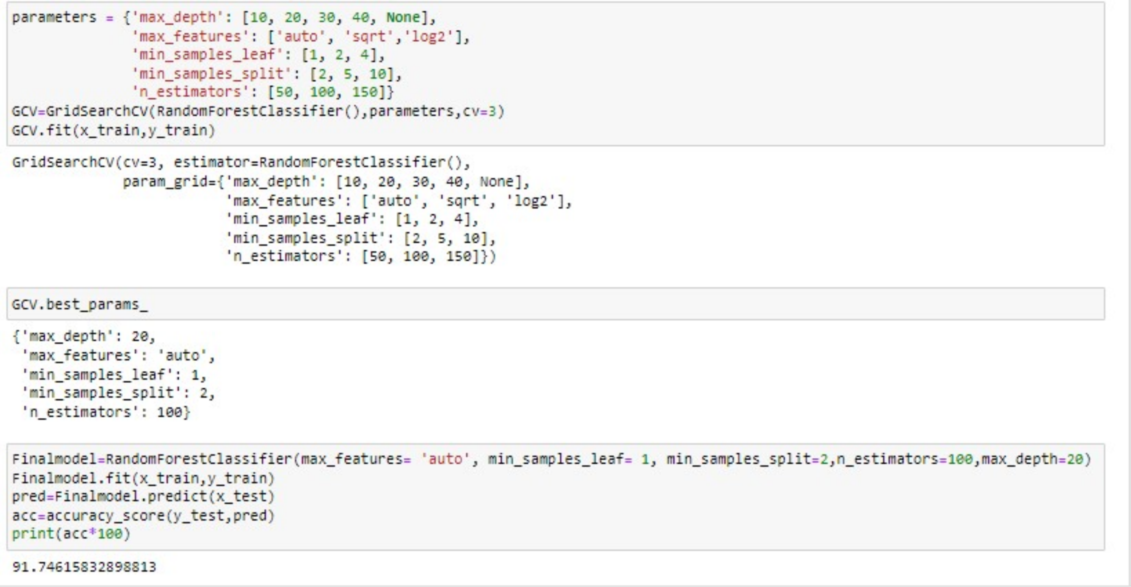
```
print("The shape before removing outliers and skewness",df.shape)
print("skewness before removing outliers")
print(df.skew())
from scipy.stats import zscore
z=np.abs(zscore(df))
df1=df[(z<3).all(axis=1)]
print("new shape after removing outliers",df1.shape)
print("skewness after removing outliers")
print(df1.skew())
```

```
The shape before removing outliers and skewness (209593, 33)
skewness before removing outliers
label                   -2.270254
aon                     10.392949
daily_decr30             3.946230
daily_decr90             4.252565
rental30                 4.521929
rental90                 4.437681
last_rech_date_ma       14.790974
last_rech_date_da       14.814857
last_rech_amt_ma         3.781149
```

Interpretation of the Results

In the visualization part, we have seen how my data looks like using heatmap, boxplot, distribution plots. In the pre-processing part, we have cleaned data using zscore.

In the modeling part, we have designed our model using algorithms like Random Forest Classifier. The accuracy score, confusion_matrix, classification_report are achieved for each model.



```
parameters = {'max_depth': [10, 20, 30, 40, None],
              'max_features': ['auto', 'sqrt','log2'],
              'min_samples_leaf': [1, 2, 4],
              'min_samples_split': [2, 5, 10],
              'n_estimators': [50, 100, 150]}
GCV=GridSearchCV(RandomForestClassifier(),parameters,cv=3)
GCV.fit(x_train,y_train)
```

```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [10, 20, 30, 40, None],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'n_estimators': [50, 100, 150]})
```

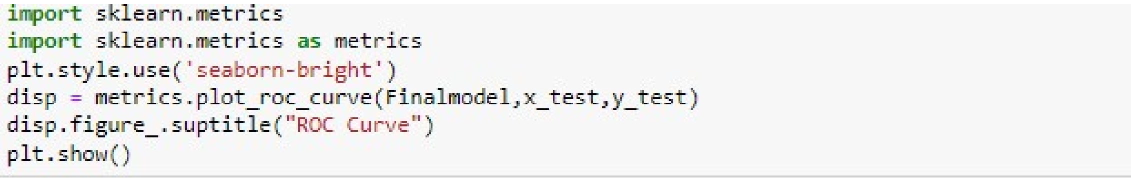
```
GCV.best_params_
```

```
{'max_depth': 20,
 'max_features': 'auto',
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'n_estimators': 100}
```

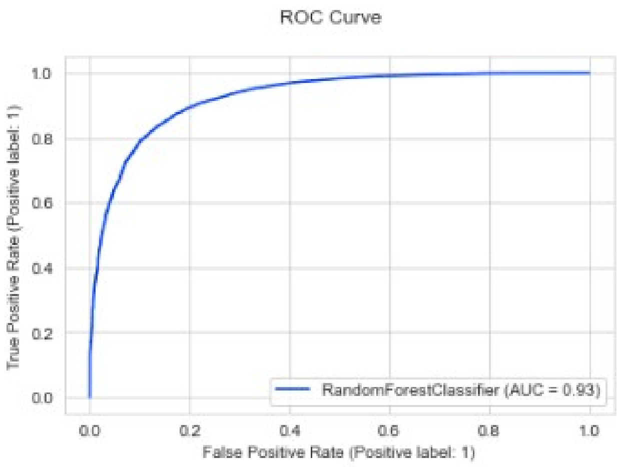
```
Finalmodel=RandomForestClassifier(max_features= 'auto', min_samples_leaf= 1, min_samples_split=2,n_estimators=100,max_depth=20)
Finalmodel.fit(x_train,y_train)
pred=Finalmodel.predict(x_test)
acc=accuracy_score(y_test,pred)
print(acc*100)
```

```
91.74615832898813
```

AUV-ROC Curve

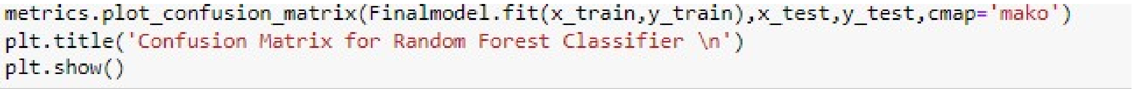


```
import sklearn.metrics
import sklearn.metrics as metrics
plt.style.use('seaborn-bright')
disp = metrics.plot_roc_curve(Finalmodel,x_test,y_test)
disp.figure_.suptitle("ROC Curve")
plt.show()
```

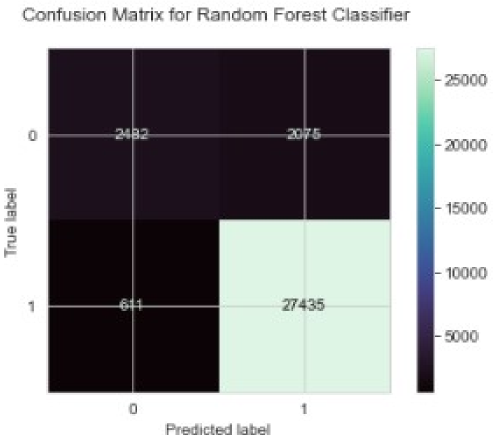


Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.



```
metrics.plot_confusion_matrix(Finalmodel.fit(x_train,y_train),x_test,y_test,cmap='mako')
plt.title('Confusion Matrix for Random Forest Classifier \n')
plt.show()
```



CONCLUSION

Key Findings and Conclusions of the Study

The key findings are we have to study the data very clearly so that we are able to decide which data are relevant for our findings. The techniques that I have used are heatmap, zscore. The conclusion of our study is we have to achieve a model with good accuracy and f1-score.

Learning Outcomes of the Study in respect of Data Science

We will develop relevant programming abilities. We will demonstrate proficiency with statistical analysis of data. We will develop the ability to build and assess data-based models. We will execute statistical analyses with professional statistical software.
The best algorithm for this project according to my work is Random forest Classifier because the accuracy that we have achieved is quite satisfactory than the other model.

Limitations of this work and Scope for Future Work

This study has proposed a comprehensive research and model development for the prediction of the default loans. As the issue related to the high ratio of bad loans is very much critical especially in micro-financing banks of various under develop and developed countries. Although, loan lending has been proven very substantial in the stability of any country's economy in this century such a huge amount of loan defaults is also very critical. To cope up with this problem a comprehensive amount of literature was reviewed to study the significant factors that lead to such problems.

Future scope of this work is we can try different algorithms like Extra tree classifier, Gaussian NB etc for model building and try to achieve a good accuracy and f1-score.