FLIP ROBO

MICRO CREDIT DEFAULTER PROJECT

ACKNOWLEDGMENT

Firstly, I would like to thank FlipRobo Technologies for giving me the opportunity to work on this project. Also, I would like to thank the DataTrained team, especially Shankargouda Tegginmani sir for providing me the knowledge and guidance which helped me a lot to work on this project.

References:

https://stackoverflow.com/

https://seaborn.pydata.org/

INTRODUCTION

- Business Problem Framing

  The main objective of this project is to build a model which can be used 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.

- Conceptual Background of the Domain Problem

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

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

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

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

They understand the importance of communication and how it affects a person’s life, thus, focusing on providing their services and products to low income families and poor customers that can help them in the need of hour.

- Review of Literature

They are collaborating with an MFI to provide micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be defaulter if he deviates from the path of paying back the loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah), while, for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah).

- Motivation for the Problem Undertaken

1. The objective behind to take this project is to harness the required data science skills.
2. Improve the analytical thinking.
3. Get into the real world problem solving mechanics.

Analytical Problem Framing

- Data Sources and their formats

The sample data is provided to us from FlipRobo 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. The summary of the dataset are as follows:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209593 entries, 0 to 209592
Data columns (total 35 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   label                 209593 non-null  int64
 1   aon                   209593 non-null  float64
 2   daily_decr30          209593 non-null  float64
 3   daily_decr90          209593 non-null  float64
 4   rental30              209593 non-null  float64
 5   rental90              209593 non-null  float64
 6   last_rech_date_ma     209593 non-null  float64
 7   last_rech_date_da     209593 non-null  float64
 8   last_rech_amt_ma      209593 non-null  int64
 9   cnt_ma_rech30         209593 non-null  int64
 10  fr_ma_rech30          209593 non-null  float64
 11  sumamnt_ma_rech30     209593 non-null  float64
 12  medianamnt_ma_rech30  209593 non-null  float64
 13  medianmarechprebal30  209593 non-null  float64
 14  cnt_ma_rech90         209593 non-null  int64
 15  fr_ma_rech90          209593 non-null  int64
 16  sumamnt_ma_rech90     209593 non-null  int64
 17  medianamnt_ma_rech90  209593 non-null  float64
 18  medianmarechprebal90  209593 non-null  float64
 19  cnt_da_rech30         209593 non-null  float64
 20  fr_da_rech30          209593 non-null  float64
 21  cnt_da_rech90         209593 non-null  int64
 22  fr_da_rech90          209593 non-null  int64
 23  cnt_loans30           209593 non-null  int64
 24  amnt_loans30          209593 non-null  int64
 25  maxamnt_loans30       209593 non-null  float64
 26  medianamnt_loans30    209593 non-null  float64
 27  cnt_loans90           209593 non-null  float64
 28  amnt_loans90          209593 non-null  int64
 29  maxamnt_loans90       209593 non-null  int64
 30  medianamnt_loans90    209593 non-null  float64
 31  payback30             209593 non-null  float64
 32  payback90             209593 non-null  float64
 33  pcircle               209593 non-null  object
 34  pdate                 209593 non-null  object
dtypes: float64(21), int64(12), object(2)
memory usage: 56.0+ MB
```

- Data Preprocessing Done

Below are the steps which we have taken in data pre - processing:

➢ Null Values:

We checked for the null values (missing values) and found that there is no null values in the given dataset.

➢ Data Cleaning:

a) Dropped ‘Unnamed:0’ column as it was not contributing to the dataset.

b) Dropped 'msisdn' as it’ll not help in the model building.

c) Split the ‘pdate’ column into day, month, and year and dropped the ‘pdate’ column.

d) Dropped ‘year’ column as it only contains 2016 as value.

e) Dropped ‘pcircle’ column as it contains single value (UPW).

- Data Inputs- Logic- Output Relationships

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

Importing the required libraries:

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

from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier

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

The main relationship between the input variable and the output variable is their correlation and covariance value. The value must lie between -1 to 1 for correlation and 0 to 1 for covariance for a strong relationship between input and the output.

For example ‘cnt_loans90’ (number of loans taken in last 90 days)

By examining this column we can establish a relation between input and output, whether the user had taken the loan or not if he had taken whether he was able to pay it or not.

- Hardware and Software Requirements and Tools Used

<u>Hardware Configuration:</u>

Operating System: Windows 10

System Type: 64-bit operating system, x64-based processor

Processor: Intel® Core™ i3-5005U @ 2.00 GHz 2.00 GHz

RAM: 4GB

<u>Software & Tools:</u>

a) Jupyter Notebook (used as a notebook to code)

b) Python (used for scientific computation)

c) Pandas (used for scientific computation)

d) Numpy (used for scientific computation)

e) Matplotlib (used for visualization)

f) Seaborn (used for visualization)

g) Scikit-learn (used as algorithmic libraries)

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
  - Performed EDA (Exploratory Data Analysis).
  - Data Cleaning and dropping the columns which were not contributing to the dataset.
  - Checked for the outliers and tried to remove the outliers of the dataset.
  - Checked for the skewness in the dataset and removed the skewness for better model building.
  - Train- Test the dataset into independent and dependent variables.
  - Model Building.
  - Cross validation score to check if the model is over-fitted.
- Testing of Identified Approaches (Algorithms)

  Below are the algorithms used for the training and testing:

  1. Logistic Regression.
  2. Ridge Classifier.
  3. Random Forest Classifier.
  4. Decision Tree Classifier.
  5. Gaussian NB.

- Run and Evaluate selected models

1. Logistic Regression:

```
from sklearn.linear_model import LogisticRegression

LR = LogisticRegression()
LR.fit(x_train, y_train)
predlr = LR.predict(x_test)

print(accuracy_score(y_test, predlr))
print(confusion_matrix(y_test,predlr))
print(classification_report(y_test, predlr))
```

```
0.7794330216230567
[[26798  6843]
 [ 8111 26046]]
              precision    recall  f1-score   support

           0       0.77      0.80      0.78     33641
           1       0.79      0.76      0.78     34157

    accuracy                           0.78     67798
   macro avg       0.78      0.78      0.78     67798
weighted avg       0.78      0.78      0.78     67798
```

From Logistic Regression we got 78% accuracy score.

2. Ridge Classifier:

```
from sklearn.linear_model import RidgeClassifier

RC = RidgeClassifier()
RC.fit(x_train,y_train)
pred_rc = RC.predict(x_test)

print(accuracy_score(y_test, pred_rc))
print(confusion_matrix(y_test, pred_rc))
print(classification_report(y_test, pred_rc))
```

```
0.7773533142570578
[[26336  7305]
 [ 7790 26367]]
              precision    recall  f1-score   support

           0       0.77      0.78      0.78     33641
           1       0.78      0.77      0.78     34157

    accuracy                           0.78     67798
   macro avg       0.78      0.78      0.78     67798
weighted avg       0.78      0.78      0.78     67798
```

From Ridge Classifier we got 78% accuracy score.

3. Random Forest Classifier:

```
from sklearn.ensemble import RandomForestClassifier

RF = RandomForestClassifier()
RF.fit(x_train, y_train)
predrf = RF.predict(x_test)

print(accuracy_score(y_test, predrf))
print(confusion_matrix(y_test, predrf))
print(classification_report(y_test, predrf))
```

```
0.9534647039735685
[[32245  1396]
 [ 1759 32398]]
              precision    recall  f1-score   support

           0       0.95      0.96      0.95     33641
           1       0.96      0.95      0.95     34157

    accuracy                           0.95     67798
   macro avg       0.95      0.95      0.95     67798
weighted avg       0.95      0.95      0.95     67798
```

From Random Forest Classifier we got 95% accuracy score.

4. Decision Tree Classifier:

```
from sklearn.tree import DecisionTreeClassifier

DT = DecisionTreeClassifier()
DT.fit(x_train, y_train)
preddt = DT.predict(x_test)

print(accuracy_score(y_test, preddt))
print(confusion_matrix(y_test, preddt))
print(classification_report(y_test, preddt))
```

```
0.913832266438538
[[30995  2646]
 [ 3196 30961]]
              precision    recall  f1-score   support

           0       0.91      0.92      0.91     33641
           1       0.92      0.91      0.91     34157

    accuracy                           0.91     67798
   macro avg       0.91      0.91      0.91     67798
weighted avg       0.91      0.91      0.91     67798
```

From Decision Tree Classifier we got 91% accuracy score.

5. Gaussian NB:

```
from sklearn.naive_bayes import GaussianNB

gussian = GaussianNB()
gussian.fit(x_train,y_train)
pred_gus = gussian.predict(x_test)

print(accuracy_score(y_test,pred_gus))
print(confusion_matrix(y_test, pred_gus))
print(classification_report(y_test, pred_gus))
```

```
0.7455972152570872
[[26934  6707]
 [10541 23616]]
              precision    recall  f1-score   support

           0       0.72      0.80      0.76     33641
           1       0.78      0.69      0.73     34157

    accuracy                           0.75     67798
   macro avg       0.75      0.75      0.74     67798
weighted avg       0.75      0.75      0.74     67798
```

From Gaussian NB we got 75% accuracy score.

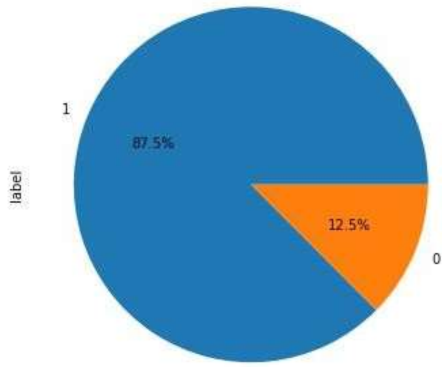
- Key Metrics for success in solving problem under consideration

  The key metrics used are as follows:

  a. Accuracy Score
  b. Confusion Matrix
  c. Classification Report
  d. F1 Score
  e. Precision & Recall
  f. Cross validation score

- Visualizations

  ◆ Checked if the data is balanced or not.

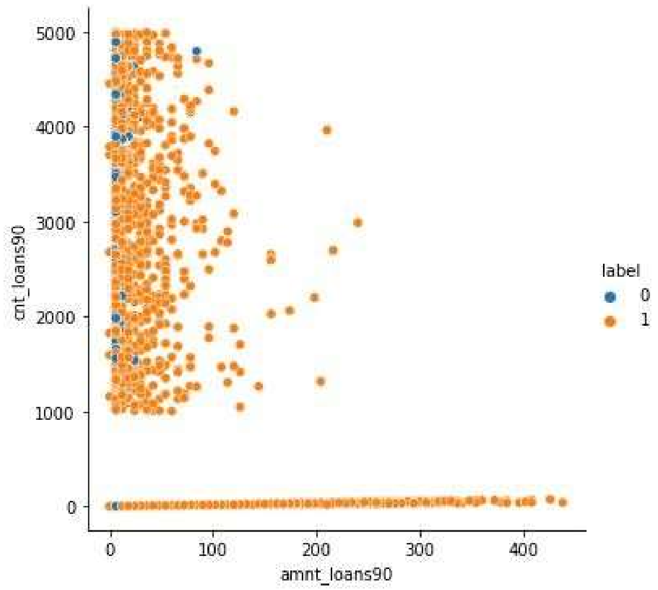


Label ‘1’ indicates Non- defaulters & label ‘0’ indicates defaulters.

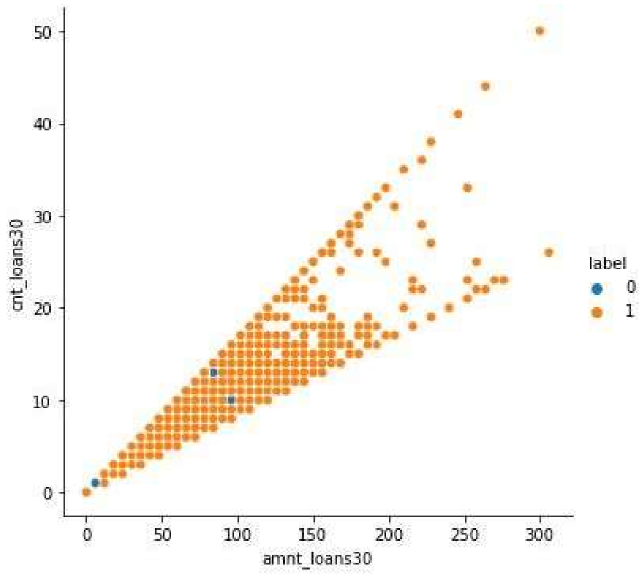
87.5% are non- defaulters and 12.5% are defaulters. This shows that the dataset is imbalance.

- ◆ The number of defaulters are more for 90 days but the loan amount is below 100.

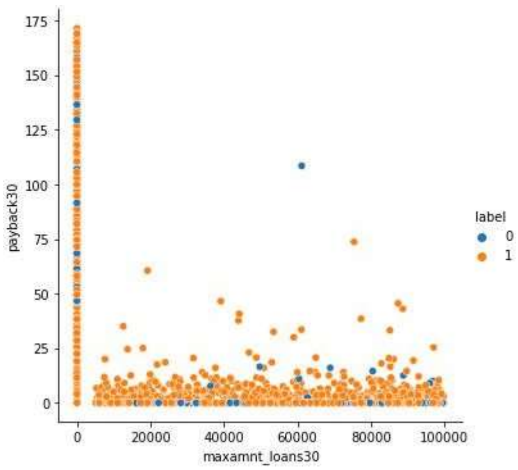
```
<Figure size 720x432 with 0 Axes>
```

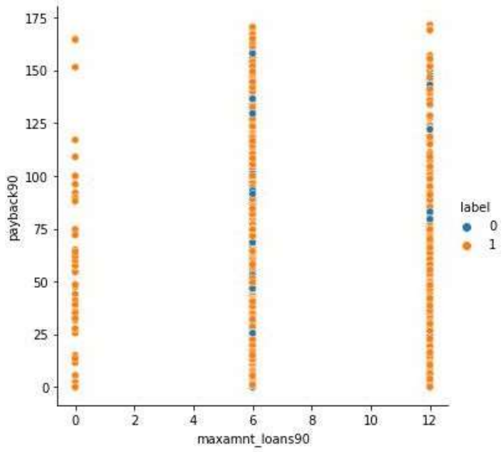


- ◆ The number of loans taken by users in last 30 days is more than 50 but the maximum loan amount taken ranges from 50 to 150.



◆ As the number of days of payback is increasing the number of defaulters are also increasing.



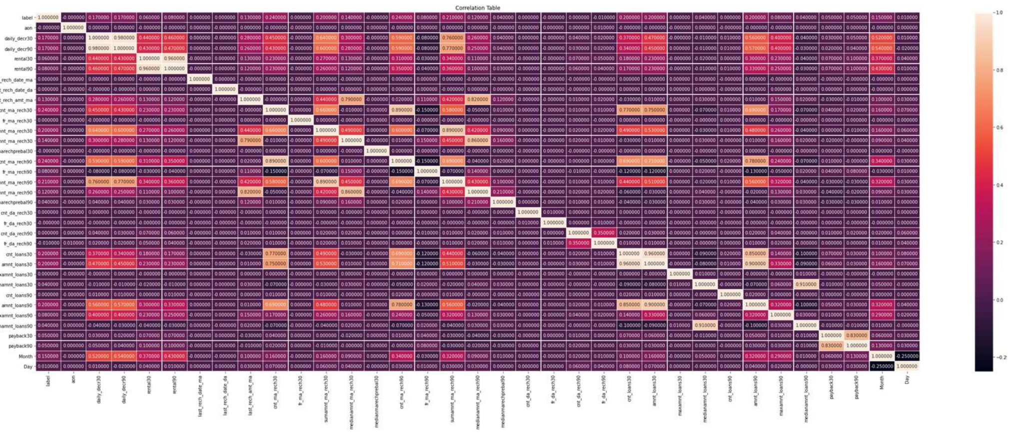


◆ Statistical Summary using Heat-map

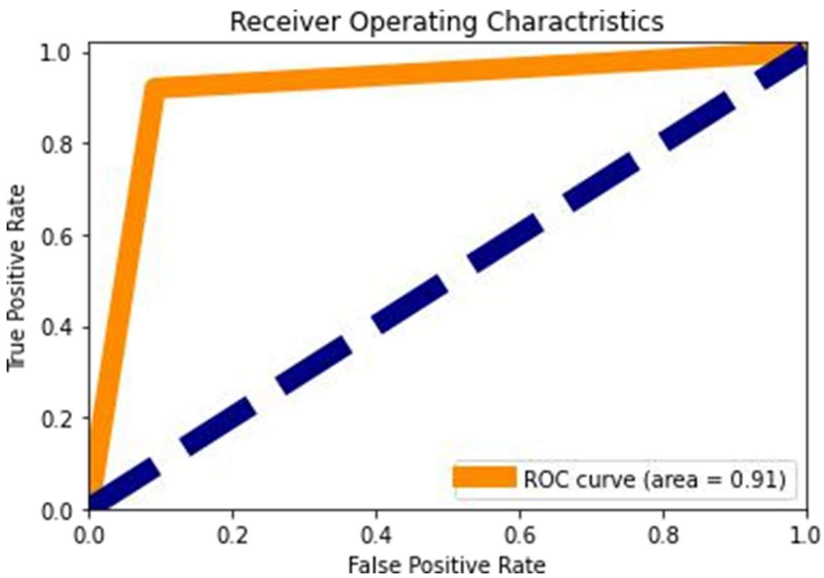
Statistical Summary

	mean	std	min	25%	50%	75%	max
label	0.880000	0.330000	0.000000	1.000000	1.000000	1.000000	1.000000
aon	8112.340000	75696.080000	-48.000000	246.000000	527.000000	982.000000	999860.760000
daily_decr30	5381.400000	9220.620000	-93.010000	42.440000	1469.180000	7244.000000	265926.000000
daily_decr90	6082.520000	10918.810000	-93.010000	42.690000	1500.000000	7802.790000	320630.000000
rental30	2692.580000	4308.590000	-23737.140000	280.420000	1083.570000	3356.940000	198926.110000
rental90	3483.410000	5770.460000	-24720.580000	300.260000	1334.000000	4201.790000	200148.110000
last_rech_date_ma	3755.850000	53905.890000	-29.000000	1.000000	3.000000	7.000000	998650.380000
last_rech_date_da	3712.200000	53374.830000	-29.000000	0.000000	0.000000	0.000000	999171.810000
last_rech_amt_ma	2064.450000	2370.790000	0.000000	770.000000	1539.000000	2309.000000	55000.000000
cnt_ma_rech30	3.980000	4.260000	0.000000	1.000000	3.000000	5.000000	203.000000
fr_ma_rech30	3737.360000	53643.630000	0.000000	0.000000	2.000000	6.000000	999606.370000
sumamnt_ma_rech30	7704.500000	10139.620000	0.000000	1540.000000	4628.000000	10010.000000	810096.000000
medianamnt_ma_rech30	1812.820000	2070.860000	0.000000	770.000000	1539.000000	1924.000000	55000.000000
medianmarechprebal30	3851.930000	54006.370000	-200.000000	11.000000	33.900000	83.000000	999479.420000
cnt_ma_rech90	6.320000	7.190000	0.000000	2.000000	4.000000	8.000000	336.000000
fr_ma_rech90	7.720000	12.590000	0.000000	0.000000	2.000000	8.000000	88.000000
sumamnt_ma_rech90	12396.220000	16857.790000	0.000000	2317.000000	7226.000000	16000.000000	953036.000000
medianamnt_ma_rech90	1864.600000	2081.680000	0.000000	773.000000	1539.000000	1924.000000	55000.000000
medianmarechprebal90	92.030000	369.220000	-200.000000	14.600000	36.000000	79.310000	41456.500000
cnt_da_rech30	262.580000	4183.900000	0.000000	0.000000	0.000000	0.000000	99914.440000
fr_da_rech30	3749.490000	53885.410000	0.000000	0.000000	0.000000	0.000000	999809.240000
cnt_da_rech90	0.040000	0.400000	0.000000	0.000000	0.000000	0.000000	38.000000
fr_da_rech90	0.050000	0.950000	0.000000	0.000000	0.000000	0.000000	64.000000
cnt_loans30	2.760000	2.550000	0.000000	1.000000	2.000000	4.000000	50.000000
amnt_loans30	17.950000	17.380000	0.000000	6.000000	12.000000	24.000000	306.000000
maxamnt_loans30	274.660000	4245.260000	0.000000	6.000000	6.000000	6.000000	99864.560000
medianamnt_loans30	0.050000	0.220000	0.000000	0.000000	0.000000	0.000000	3.000000
cnt_loans90	18.520000	224.800000	0.000000	1.000000	2.000000	5.000000	4997.520000
amnt_loans90	23.650000	26.470000	0.000000	6.000000	12.000000	30.000000	438.000000
maxamnt_loans90	6.700000	2.100000	0.000000	6.000000	6.000000	6.000000	12.000000
medianamnt_loans90	0.050000	0.200000	0.000000	0.000000	0.000000	0.000000	3.000000
payback30	3.400000	8.810000	0.000000	0.000000	0.000000	3.750000	171.500000
payback90	4.320000	10.310000	0.000000	0.000000	1.670000	4.500000	171.500000
Month	6.800000	0.740000	6.000000	6.000000	7.000000	7.000000	8.000000
Day	14.400000	8.440000	1.000000	7.000000	14.000000	21.000000	31.000000

◆ Heat-map for the correlation table:



◆ ROC AUC Curve:



Area for the ROC curve is 0.91.

CONCLUSION

- Key Findings and Conclusions of the Study

  - If the number of days of payback is increasing the chance of defaulters is also increasing. So, we should look for the payback duration.

  - If the loan amount is below 100 and the number of loans taken by users is 90 days, the number of defaulters is increasing.

- Learning Outcomes of the Study in respect of Data Science

  This project helped me to work on the real time industrial data, which helped me to gain the real time experience. In the project I got to work on the different type of algorithms and fitting the best model based on the accuracy score and cross validation score. We achieved accuracy score of 91% using the Decision Tree Classifier.

```
0.9133602761143397
[[30934  2707]
 [ 3167 30990]]
              precision    recall  f1-score   support

           0       0.91      0.92      0.91     33641
           1       0.92      0.91      0.91     34157

    accuracy                           0.91     67798
   macro avg       0.91      0.91      0.91     67798
weighted avg       0.91      0.91      0.91     67798
```

- After hyper parameter tuning we're getting 91% accuracy score.