FLIP ROBO

PROJECT REPORT ON:

"Micro-Credit Defaulter Model"

SUBMITTED BY

Akas Mandal

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 Data Trained team, especially Shankar Gouda 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 Pre-processing 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 11

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

Processor: Intel® Core™ i5-1035G1 CPU @ 1.00GHz 1.20 GHz

RAM 8GB

<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))
```

```
0.7796837664827871
[[27073  6992]
 [ 7945 25788]]
              precision    recall  f1-score   support

           0       0.77      0.79      0.78     34065
           1       0.79      0.76      0.78     33733

    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))
```

```
0.7769698221186465
[[26615  7450]
 [ 7671 26062]]
              precision    recall  f1-score   support

           0       0.78      0.78      0.78     34065
           1       0.78      0.77      0.78     33733

    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))
```

```
0.9540251924835541
[[32717  1348]
 [ 1769 31964]]
              precision    recall  f1-score   support

           0       0.95      0.96      0.95     34065
           1       0.96      0.95      0.95     33733

    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))
```

```
0.9169887017316145
[[31479  2586]
 [ 3042 30691]]
              precision    recall  f1-score   support

           0       0.91      0.92      0.92     34065
           1       0.92      0.91      0.92     33733

    accuracy                           0.92     67798
   macro avg       0.92      0.92      0.92     67798
weighted avg       0.92      0.92      0.92     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))
```

```
0.7467919407652143
[[27235  6830]
 [10337 23396]]
              precision    recall  f1-score   support

           0       0.72      0.80      0.76     34065
           1       0.77      0.69      0.73     33733

    accuracy                           0.75     67798
   macro avg       0.75      0.75      0.75     67798
weighted avg       0.75      0.75      0.75     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

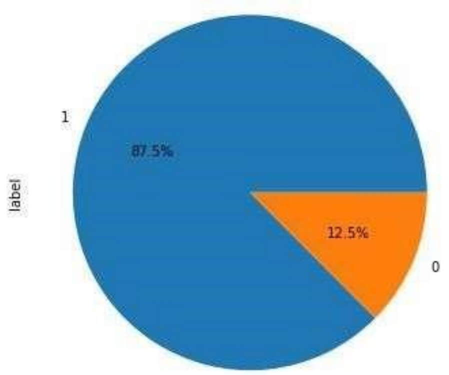
d. F1 Score

e. Precision & Recall

f. Cross validation score

❖ <u>Visualizations</u>

➢ Checked if the data is balanced or not.

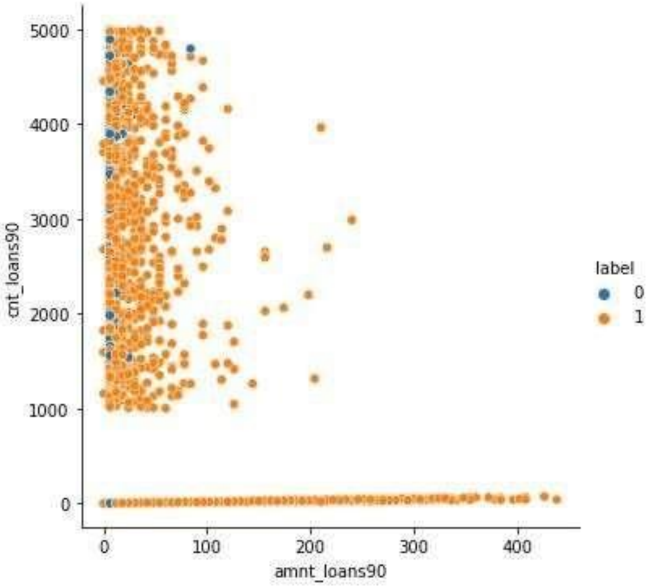


Label ‘1’ indicates non-defaulters & label ‘0’ indicates defaulters.

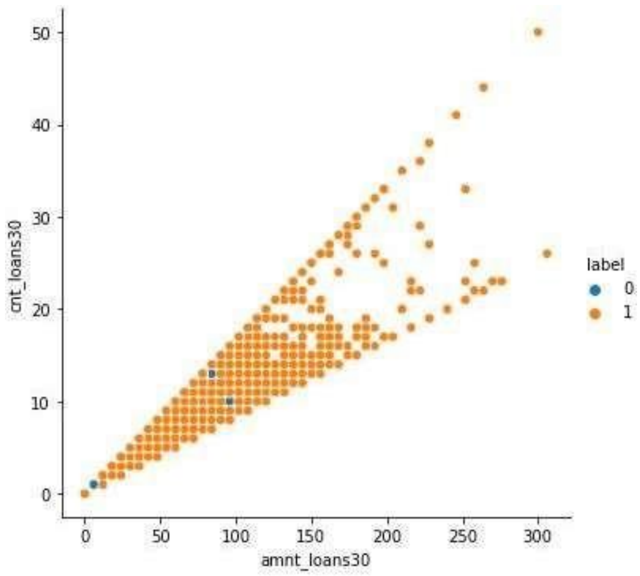
87.5% are non- defaulters and 12.5% are defaulters. This shows that the dataset is imbalance.

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

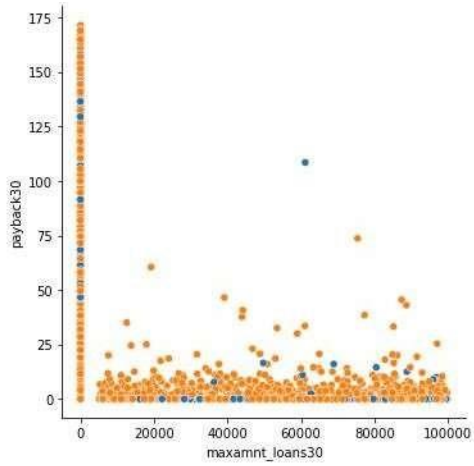
<Figure size 720x432 with 0 Axes>

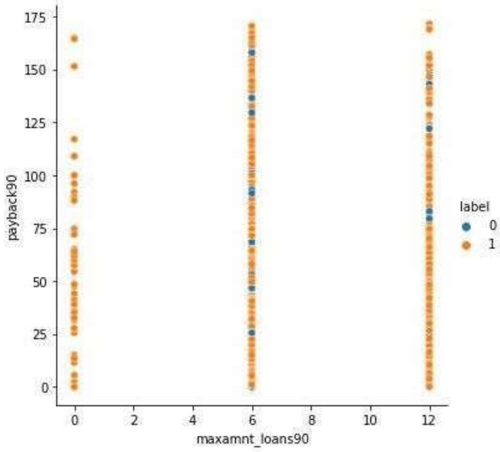


- The number of loans taken by users in last 30 days is more than 50 but the maximum loan amount taken ranges from 50 to 150.



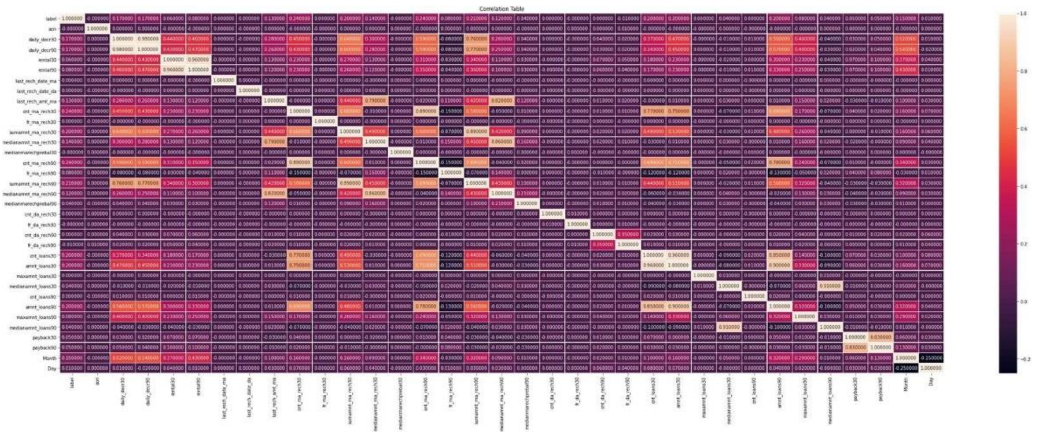
- As the number of days of payback is increasing the number of defaulters are also increasing.



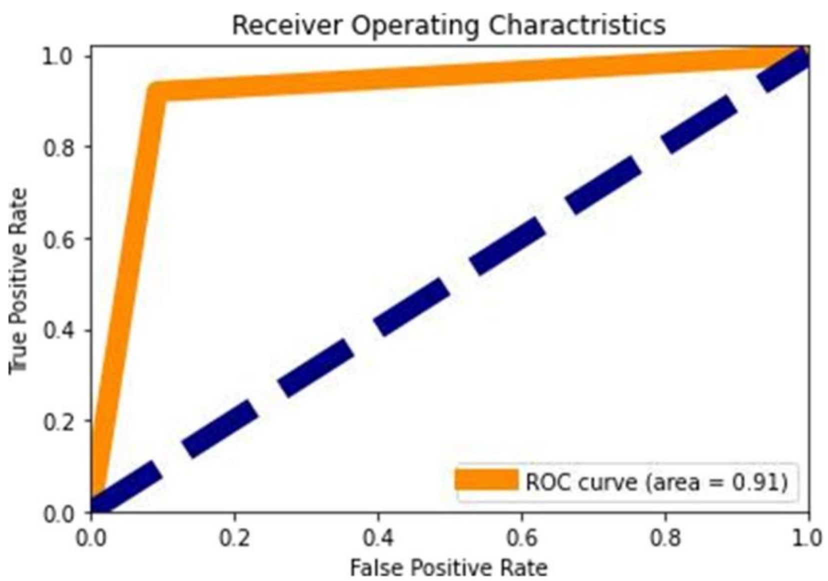


➢ Statistical Summary using Heat-map

➢ 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.

```
0.9133750258119708
[[31308  2757]
 [ 3116 30617]]
              precision    recall  f1-score   support

           0       0.91      0.92      0.91     34065
           1       0.92      0.91      0.91     33733

    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.

Learning Outcomes of the Study in respect of Data Science

I found that the dataset was quite interesting to handle as it contains all types of data in it. Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed and analysed. New analytical techniques of machine learning can be used in property research. The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say. Data cleaning is one of the most important steps to remove unrealistic values and zero values. This study is an exploratory attempt to use four machine learning algorithms in estimating micro credit defaulter, and then compare their results.

To conclude, the application of machine learning in micro credit is still at an early stage. We hope this study has moved a small step ahead in providing some methodological and empirical contributions to crediting institutes, and presenting an alternative approach to the valuation of defaulters. Future direction of research may consider incorporating additional micro credit transaction data from a larger economical background with more features.

Limitations of this work and Scope for Future Work

- ✓ First drawback is the length of the dataset it is very huge and hard to handle.
- ✓ Followed by a greater number of outliers and skewness these two will reduce our model accuracy.
- ✓ Also, we have tried best to deal with outliers, skewness and zero values. So, it looks quite good that we have achieved an accuracy of 94.82% even after dealing all these drawbacks.
- ✓ Also, this study will not cover all Classification algorithms instead, it is focused on the chosen algorithm, starting from the basic ensembling techniques to the advanced ones.