FLIP ROBO

Micro-Credit Defaulter Model

Submitted By:

Lakshmi Rajendra Thute

ACKNOWLEDGEMENT

I am very much Thankful to FlipRobo Technologies for giving me the opportunity to work with them and to work on this project and also, I am very grateful to Data Trained Education Team for their support and help to understand each and every concept of machine learning which helped me a lot while working on this project. I thought, I am fortunate to become a part of FlipRobo Technology.

References:

Google website

Stack overflow

Analytics Vidya

Medium

Data trained notes

INTRODUCTION

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.

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.

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.

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).

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.

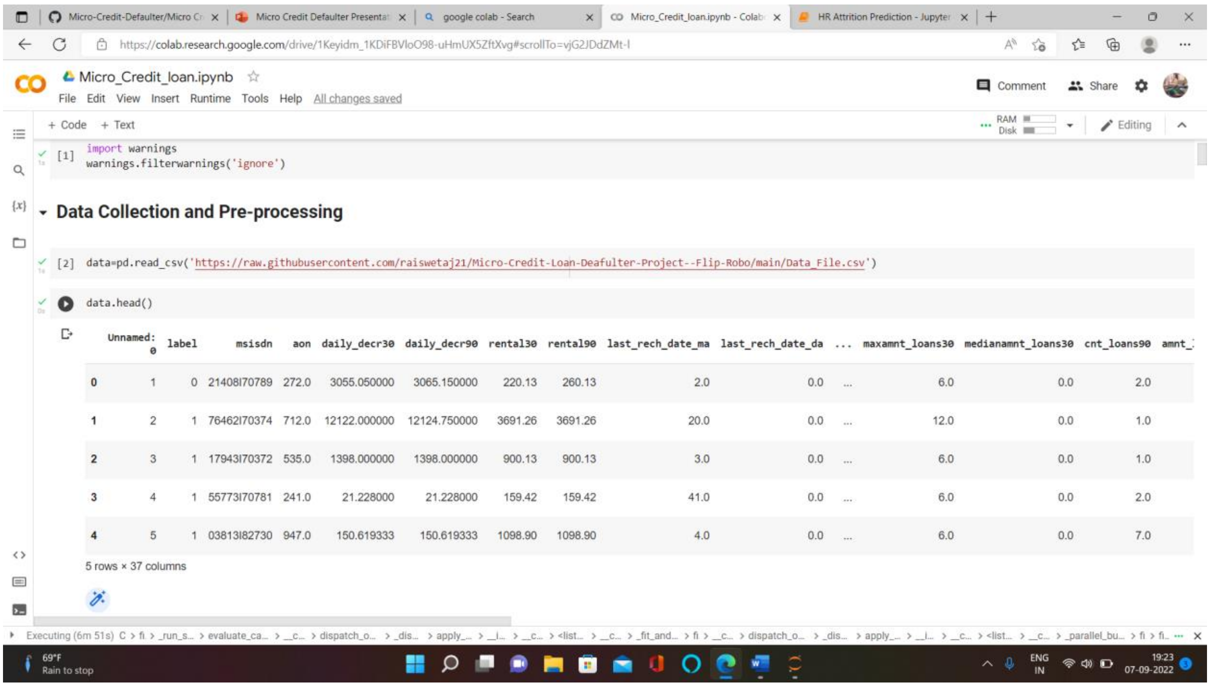
ANALYTICAL PROBLEM FRAMING

- ## Mathematical/ Analytical Modelling of the Problem

  Here I have done Data Pre-processing, Exploratory Data Analysis, then Encoding and lastly model Building and Evaluation.

- ## Data Sources and their formats

  I got the dataset in CSV format and I read the data in Jupyter Notebook using Pandas data frame.



```
[1] import warnings
    warnings.filterwarnings('ignore')
```

Data Collection and Pre-processing

```
[2] data=pd.read_csv('https://raw.githubusercontent.com/raiswetaj21/Micro-Credit-Loan-Deafulter-Project--Flip-Robo/main/Data_File.csv')
```

```
data.head()
```

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	...	maxamnt_loans30	medianamnt_loans30	cnt_loans90	amnt_
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	...	6.0	0.0	2.0	
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	...	12.0	0.0	1.0	
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	...	6.0	0.0	1.0	
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	...	6.0	0.0	2.0	
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	...	6.0	0.0	7.0	

5 rows × 37 columns

- ## Data Pre-processing Done

  The dataset doesn’t contain object data type columns, no missing values. So, there is no need to treat them.

- ## Hardware and Software Requirements and Tools Used

Here for this project, I used Jupyter notebook and tools used pandas and NumPy for mathematical operations, matplotlib and seaborn for various type of data visualizations.

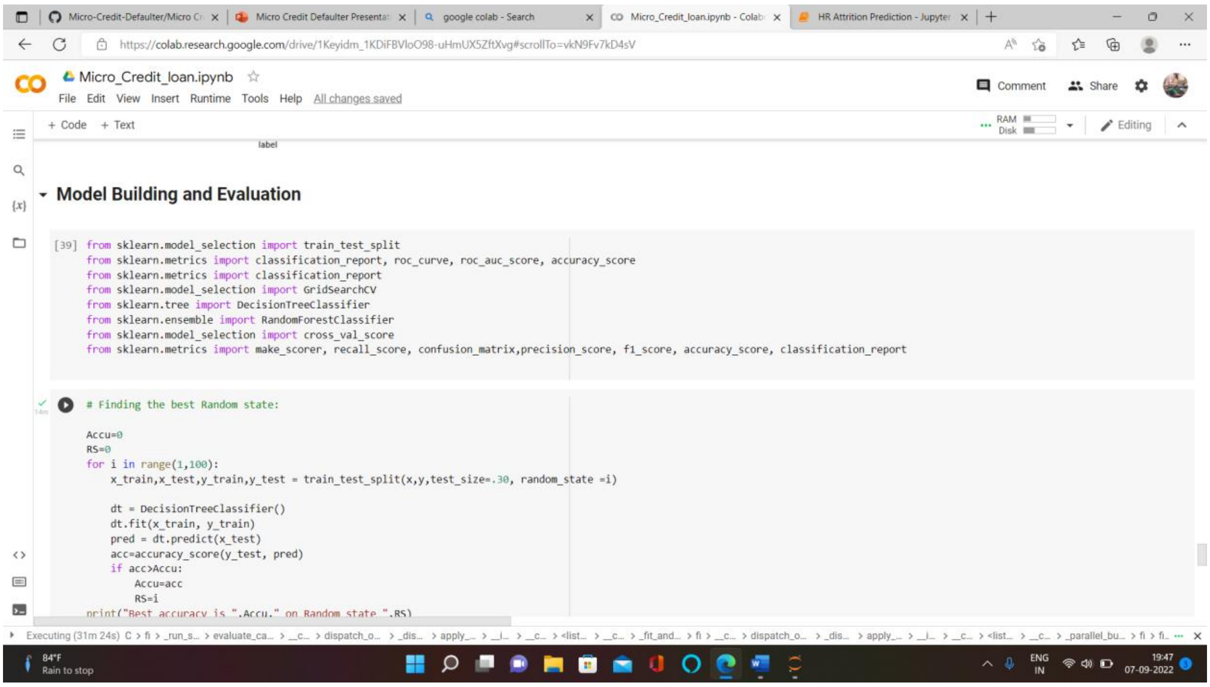
- **Identification of possible problem-solving approaches (methods)**

The statistical summary shows the total count of `209593` rows then mean, min value, max value, standard deviation and quartiles shows up and down values that means the data contains outliers.

- **Testing of Identified Approaches (Algorithms)**

  1. Random Forest Classifier
  2. Decision Tree Classifier

- **Run and evaluate selected models**


Micro-Credit-Defaulter/Micro Cr
Micro Credit Defaulter Presentat
google colab - Search
Micro_Credit_loan.ipynb - Colab
HR Attrition Prediction - Jupyter
https://colab.research.google.com/drive/1Keyidm_1KDiFBVloO98-uHmUX5ZftXvg#scrollTo=vkN9Fv7kD4sV
Micro_Credit_loan.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
label
Executing (31m 24s) C > fi > _run_s... > evaluate_ca... > __c... > dispatch_o... > _dis... > apply_... > __i... > __c... > <list... > __c... > _fit_and... > fi > __c... > dispatch_o... > _dis... > apply_... > __i... > __c... > <list... > __c... > _parallel_bu... > fi > fi...
84°F
Rain to stop
ENG
IN
19:47
07-09-2022

Model Building and Evaluation

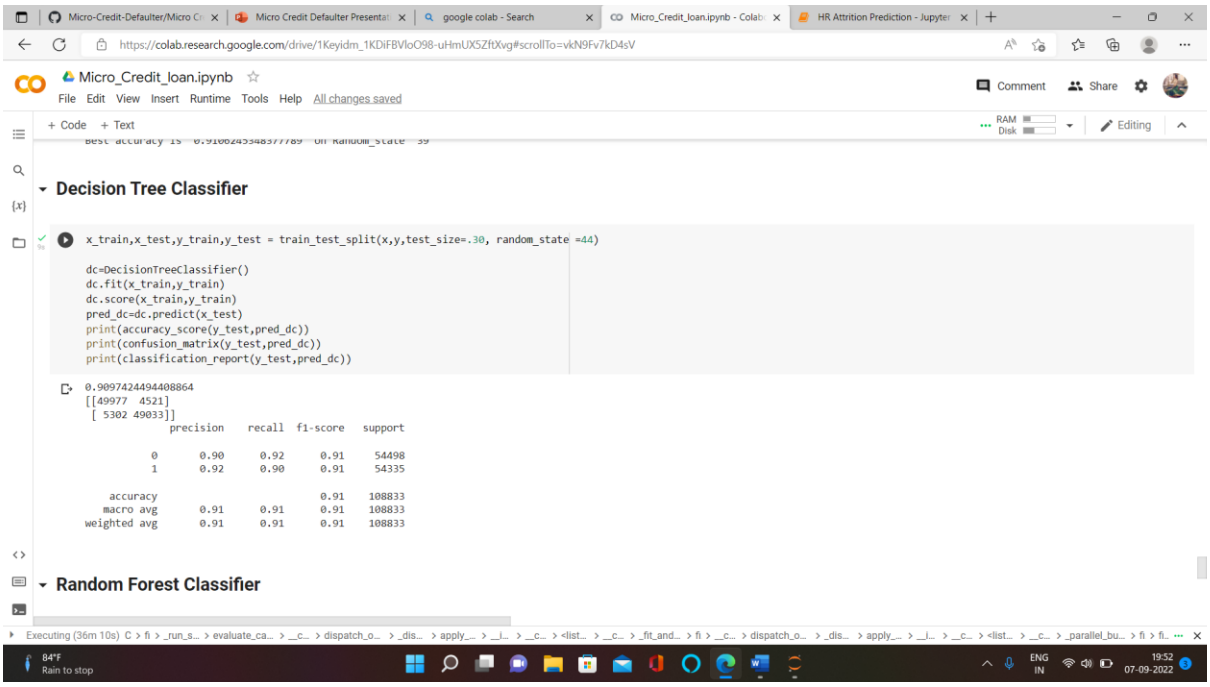
```
[39] from sklearn.model_selection import train_test_split
     from sklearn.metrics import classification_report, roc_curve, roc_auc_score, accuracy_score
     from sklearn.metrics import classification_report
     from sklearn.model_selection import GridSearchCV
     from sklearn.tree import DecisionTreeClassifier
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.model_selection import cross_val_score
     from sklearn.metrics import make_scorer, recall_score, confusion_matrix,precision_score, f1_score, accuracy_score, classification_report
```

```
# Finding the best Random state:

Accu=0
RS=0
for i in range(1,100):
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=.30, random_state =i)

    dt = DecisionTreeClassifier()
    dt.fit(x_train, y_train)
    pred = dt.predict(x_test)
    acc=accuracy_score(y_test, pred)
    if acc>Accu:
        Accu=acc
        RS=i
print("Best accuracy is ",Accu," on Random state ",RS)
```

Decision Tree is a Supervised Machine Learning Algorithm that uses a set of rules to make decisions, similarly to how humans make decisions.



```
Best accuracy is  0.9106245348377789  on Random_state  39
```

Decision Tree Classifier

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=.30, random_state =44)

dc=DecisionTreeClassifier()
dc.fit(x_train,y_train)
dc.score(x_train,y_train)
pred_dc=dc.predict(x_test)
print(accuracy_score(y_test,pred_dc))
print(confusion_matrix(y_test,pred_dc))
print(classification_report(y_test,pred_dc))
```

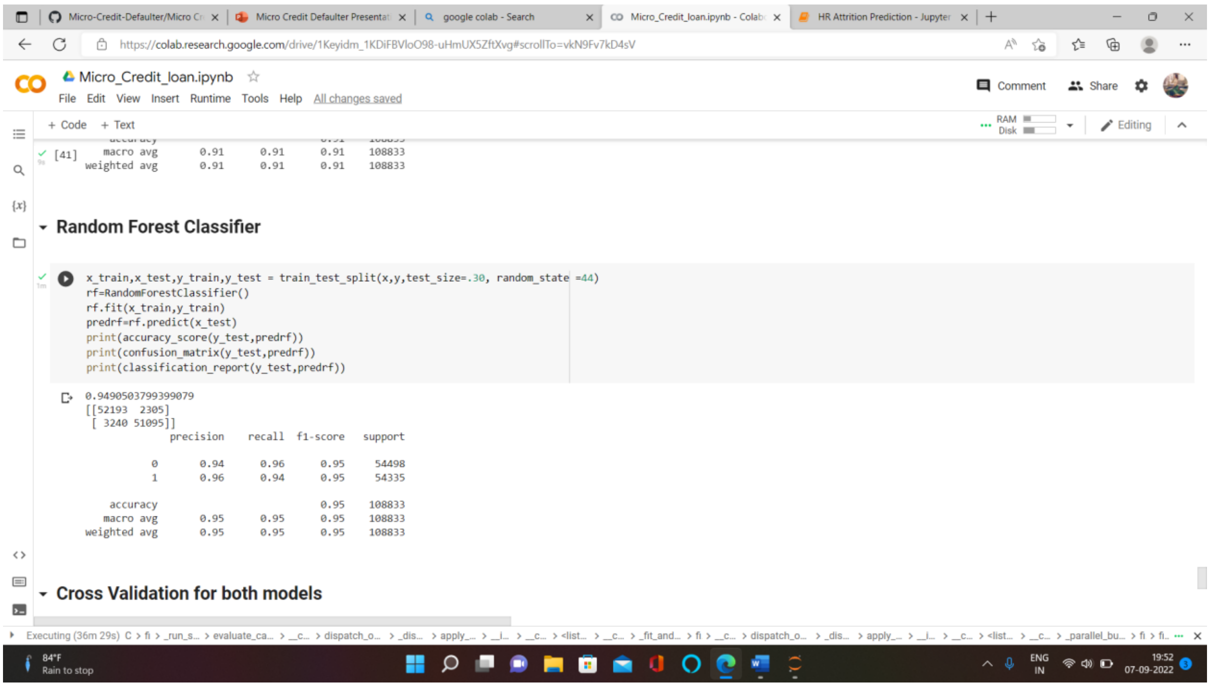
```
0.9097424494408864
[[49977  4521]
 [ 5302 49033]]
              precision    recall  f1-score   support

           0       0.90      0.92      0.91     54498
           1       0.92      0.90      0.91     54335

    accuracy                           0.91    108833
   macro avg       0.91      0.91      0.91    108833
weighted avg       0.91      0.91      0.91    108833
```

Random Forest Classifier

The random forest classifier is a collection of prediction trees. Every tree is dependent on random vectors sampled independently, with similar distribution with every other tree in the random forest.



```
[41]     macro avg       0.91      0.91      0.91    108833
      weighted avg       0.91      0.91      0.91    108833
```

Random Forest Classifier

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=.30, random_state =44)
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))
```

```
0.9490503799399079
[[52193  2305]
 [ 3240 51095]]
              precision    recall  f1-score   support

           0       0.94      0.96      0.95     54498
           1       0.96      0.94      0.95     54335

    accuracy                           0.95    108833
   macro avg       0.95      0.95      0.95    108833
weighted avg       0.95      0.95      0.95    108833
```

Cross Validation for both models

- **Interpretation of the Results**

Here after pre-processing we get the data encoded for framing the model and after visualization, we observe that the data contains skewness and we removed it using power transformation (yeo-john method), After data pre-processing and EDA we build 2 different algorithms for dataset and from them Random Forest Classifier algorithm perform very well, also we checked the cross-validation score. That also says that Random Forest Classifier is best model.