FLIP ROBO

Micro-Credit Defaulter Model

Submitted by:

SHIVANCHAL ASTHANA

ACKNOWLEDGMENT

It is my sensual gratification to present this report. Working on this project was an incredible experience that will have a tremendous impact on my career. I would like to express my sincere thanks to the company Flip Robo Technologies for a regular follow up and valuable suggestions provided throughout. They always been an origin of spark and direction. I also thank all the respondents who have given their valuable time, views and valid information for this project.

Shivanchal Asthana

INTRODUCTION

1. Business Problem Framing:

The main problem is to how to improve the selection of customers to reduce the micro credit defaulters.

2. Conceptual Background of the Domain Problem:

Microcredit was built on the concept that people with skills and more entrepreneurial mindsets also came from impoverished countries that did not necessarily have access to financial services that could suit them.

3. Motivation for the Problem Undertaken:

Motive about 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. 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.

Analytical Problem Framing

4. Mathematical/ Analytical Modeling of the Problem:

Starting with the dataset, when I looked through the statistical description, we come to see that most of the data are unbalanced and highly skewed. Some columns are negatively skewed and some have high zero values. To remove the outliers, I used IQR method, and many columns have some zero values, so, we replaced them by column mean value. The visualization also helped to identify the skewness present in the data. That skewness was also corrected using Log transformation. At last, after data pre-processing, we come the model building section, were I used Logistic Regression, Decision Tree, KNN, Support vector machine, Random Forest Classifier and bagging classifier. To improve accuracy, we use hyperparameter tuning.

5. Data Sources and their formats:

This data is been provided by a Telecom company. 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. The company shared around 2 lakh data of their customer with different transaction behaviour to understand and to predict their future behaviour. The data is been provided in CSV format with 37 different variables in different columns and 209593 rows.

6. Data Preprocessing Done:

In this dataset, most of the data are skewed and columns contains full of outliers and zero values.

We replaced zero value by their respective column mean value. We remove skewness by log transformation method and we remove outliers by using IQR method.

7. Data Inputs- Logic- Output Relationships

The input data provided, helps to understand the behaviour of the customer, their various transaction records, their frequency of transaction during a period of time etc, all these helps to predict the customer's intension toward the repayment of loan.

8. State the set of assumptions (if any) related to the problem under consideration:

No as such assumption been done related to the circumstances.

9. Hardware and Software Requirements and Tools Used:

Hardware is used by me that is i5 intel core, 8GB RAM, 64Bit processor, and software is Jupyter Notebook for coding along with MS Word to make useful report, MS – PowerPoint to make presentation. For coding, we need to install NumPy, Pandas, Sklearn libraries.

Model/s Development and Evaluation

10. Identification of possible problem-solving approaches (methods):

The data set contain more than 2 lakh data with no null values related to the customer. The dataset is imbalanced. Label 1 has 87.5% of data whereas label 0 has approximately 12.5%. As I went through the dataset, I found lot of outliers and skewness are present in the dataset. The outliers were corrected by replacing them with IQR method. The skewness was also reduced using Log transformation wherever applicable. There were certain columns which had least importance with our target variable, hence those were dropped. After data cleaning and data transformation, data visualization was done to represent data graphically. At last, the most important part was to build model for the data set.

11. Testing of Identified Approaches (Algorithms):

1. Logistic Regression
2. KNN Classifier
3. Decision Tree
4. Support vector machine
5. Random forest classifier

12. Visualizations and observations:

```
In [4]: df.info() #check the datatype of all columns

<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
 16  cnt_ma_rech90         209593 non-null  int64
 17  fr_ma_rech90          209593 non-null  int64
 18  sumamnt_ma_rech90     209593 non-null  int64
 19  medianamnt_ma_rech90  209593 non-null  float64
 20  medianmarechprebal90  209593 non-null  float64
 21  cnt_da_rech30         209593 non-null  float64
 22  fr_da_rech30          209593 non-null  float64
 23  cnt_da_rech90         209593 non-null  int64
 24  fr_da_rech90          209593 non-null  int64
 25  cnt_loans30           209593 non-null  int64
 26  amnt_loans30          209593 non-null  int64
 27  maxamnt_loans30       209593 non-null  float64
 28  medianamnt_loans30    209593 non-null  float64
 29  cnt_loans90           209593 non-null  float64
 30  amnt_loans90          209593 non-null  int64
 31  maxamnt_loans90       209593 non-null  int64
 32  medianamnt_loans90    209593 non-null  float64
 33  payback30             209593 non-null  float64
 34  payback90             209593 non-null  float64
 35  pcircle               209593 non-null  object
 36  pdate                 209593 non-null  object
dtypes: float64(21), int64(13), object(3)
memory usage: 59.2+ MB
```

-

```
In [6]: df.duplicated().sum() #check the duplicate values

Out[6]: 0
```

```
In [3]: df.shape #check the shape of the dataset

Out[3]: (209593, 37)
```

Observations:

1. By observing info of this dataset, we can clearly see, the datatypes of this dataset, mostly are in int and float, three are in object.
2. There are no duplicate values in this dataset.
3. We can see the shape of the dataset.

```
In [8]: df.skew() #check the skewness
```

```
Out[8]: Unnamed: 0              0.000000
        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
        cnt_ma_rech30           3.283842
        fr_ma_rech30           14.772833
        sumamnt_ma_rech30       6.386787
        medianamnt_ma_rech30    3.512324
        medianmarechprebal30   14.779875
        cnt_ma_rech90           3.425254
        fr_ma_rech90            2.285423
        sumamnt_ma_rech90       4.897950
        medianamnt_ma_rech90    3.752706
        medianmarechprebal90   44.880503
        cnt_da_rech30          17.818364
        fr_da_rech30           14.776430
        cnt_da_rech90          27.267278
        fr_da_rech90           28.988083
        cnt_loans30             2.713421
        amnt_loans30            2.975719
        maxamnt_loans30        17.658052
        medianamnt_loans30      4.551043
        cnt_loans90            16.594408
        amnt_loans90            3.150006
        maxamnt_loans90         1.678304
        medianamnt_loans90      4.895720
        payback30               8.310695
        payback90               6.899951
        dtype: float64
```

Observations:

1. We can see the skewness of every column.
2. Mostly columns are positively skewed.
3. Our Target column is negatively skewed.
4. If we see, second image, mostly column contains zero value.

```
In [9]: df.all() #check the zero values
Out[9]: Unnamed: 0                True
        label                    False
        msisdn                    True
        aon                       True
        daily_decr30             False
        daily_decr90             False
        rental30                 False
        rental90                 False
        last_rech_date_ma        False
        last_rech_date_da        False
        last_rech_amt_ma         False
        cnt_ma_rech30            False
        fr_ma_rech30             False
        sumamnt_ma_rech30        False
        medianamnt_ma_rech30     False
        medianmarechprebal30     False
        cnt_ma_rech90            False
        fr_ma_rech90             False
        sumamnt_ma_rech90        False
        medianamnt_ma_rech90     False
        medianmarechprebal90     False
        cnt_da_rech30            False
        fr_da_rech30             False
        cnt_da_rech90            False
        fr_da_rech90             False
        cnt_loans30              False
        amnt_loans30             False
        maxamnt_loans30          False
        medianamnt_loans30       False
        cnt_loans90              False
        amnt_loans90             False
        maxamnt_loans90          False
        medianamnt_loans90       False
        payback30                False
        payback90                False
        pcircle                   True
        pdate                     True
        dtype: bool
```

```
df.groupby('label')['last_rech_date_ma'].value_counts()
#check the relationship between column and label
```

```
label  last_rech_date_ma
0      0.000000             6273
       2.000000             1809
       1.000000             1271
       3.000000             1222
       4.000000             1005
                            ...
1      993905.540090           1
       994295.955985           1
       994622.180471           1
       997717.809631           1
       998650.377733           1
Name: last_rech_date_ma, Length: 1214, dtype: int64
```

```
df.groupby('label')['daily_decr30'].value_counts()
#check the relationship between column and label
```

```
label  daily_decr30
0      5421.149495      3572
       500.000000        529
       1000.000000       270
       700.000000        155
       600.000000        143
                        ...
1      183850.000000       1
       185313.000000       1
       212202.000000       1
       212364.000000       1
       265926.000000       1
Name: daily_decr30, Length: 149037, dtype: int64
```

```
df.groupby('label')['daily_decr90'].value_counts()
#check the relationship between column and label
```

```
label  daily_decr90
0      0.00            3572
       500.00           529
       1000.00          271
       700.00           155
       600.00           143
                        ...
1      231228.81          1
       244906.76          1
       254657.13          1
       259525.00          1
       320630.00          1
Name: daily_decr90, Length: 160564, dtype: int64
```

```
df['last_rech_date_ma'].value_counts()
```

```
1.000000          44170
2.000000          25627
3.000000          19067
0.000000          15959
4.000000          14655
                  ...
824616.273632         1
931546.437088         1
740555.920987         1
826835.412416         1
700461.145723         1
Name: last_rech_date_ma, Length: 1121,
```

Observations:

1. Mostly, life of cellular network is less in days.
2. Mostly, peoples are spending daily amount is less 26500 in (Indonesian rupiah)

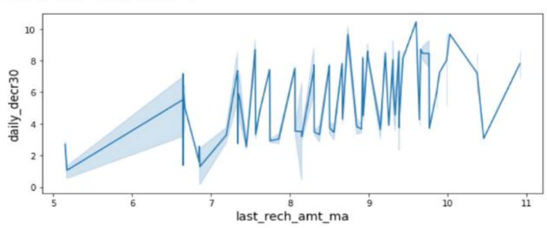
3. We can see, mainly people unable to pay back the amount where daily average amount is less.

4. We can see, mostly failure in loans when daily average spent amount is less.

5. Number of days of last recharge of main accounts are very less, means user are frequently recharge their main account.

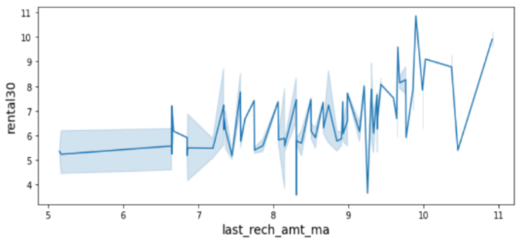
```
#visualization
plt.figure(figsize=(10,4))
sns.lineplot(x= 'last_rech_amt_ma', y = 'daily_decr30', data=df)
plt.xlabel('last_rech_amt_ma', fontsize = 14)
plt.ylabel('daily_decr30', fontsize = 14)
```

```
Text(0, 0.5, 'daily_decr30')
```



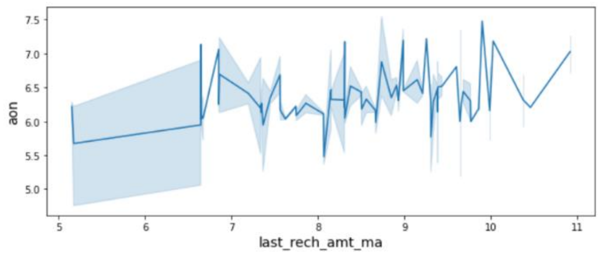

```
#visualization
plt.figure(figsize=(10,4))
sns.lineplot(x= 'last_rech_amt_ma', y = 'rental30', data=df)
plt.xlabel('last_rech_amt_ma', fontsize = 14)
plt.ylabel('rental30', fontsize = 14)
```

```
Text(0, 0.5, 'rental30')
```




```
#visualization
plt.figure(figsize=(10,4))
sns.lineplot(x= 'last_rech_amt_ma', y = 'aon', data=df)
plt.xlabel('last_rech_amt_ma', fontsize = 14)
plt.ylabel('aon', fontsize = 14)
```

```
Text(0, 0.5, 'aon')
```



<u>Observations:</u>

1. We can clearly analyze by value counts method, that people who do low amount of recharge are higher than others.

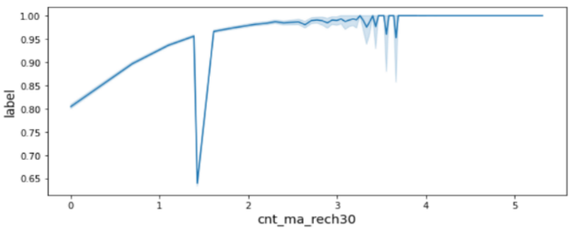
2. We can see here, the daily amount spent averaged over last 30 days is increasing as the last recharge amount is increasing

3. Average main account balance is also proportional to last recharge amount

4. As the age of cellular network increases, the last recharge amount is increases.

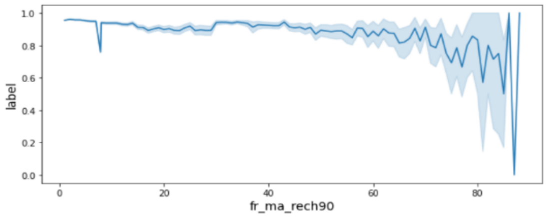
```python
#visualization
plt.figure(figsize=(10,4))
sns.lineplot(x= 'cnt_ma_rech30', y = 'label', data=df)
plt.xlabel('cnt_ma_rech30', fontsize = 14)
plt.ylabel('label', fontsize = 14)
```

Text(0, 0.5, 'label')



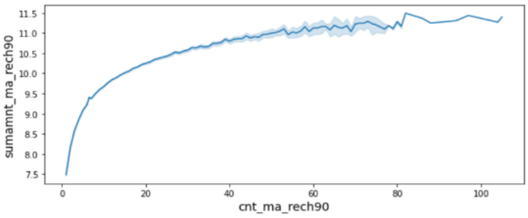

```python
#visualization
plt.figure(figsize=(10,4))
sns.lineplot(x= 'fr_ma_rech90', y = 'label', data=df)
plt.xlabel('fr_ma_rech90', fontsize = 14)
plt.ylabel('label', fontsize = 14)
```

Text(0, 0.5, 'label')




```python
#visualization
plt.figure(figsize=(10,4))
sns.lineplot(x= 'cnt_ma_rech90', y = 'sumamnt_ma_rech90', data=df)
plt.xlabel('cnt_ma_rech90', fontsize = 14)
plt.ylabel('sumamnt_ma_rech90', fontsize = 14)
```

Text(0, 0.5, 'sumamnt_ma_rech90')



Observations:

1. One times recharge is more than others
2. Mostly people have done 4 times recharge when we compare it to age of celluar network in days
3. As the daily amount spent is increases, the main account got recharged more
4. As we see, Number of times main account got recharged, the average main amount balance increases.
5. We can see clearly, people do recharge 4 times in 30 days, when their recharge amount is less
6. We can see, people who do 1 0r 2 times recharge their success rate is less than others.
7. People who do less amount of recharge their success rate is very less
8. As the number of frequency of recharge increases, recharge amount is also increases.
9. 3 times data account is recharged for approximately low to medium recharge values

10. As the daily amount spent increases, number of loans are also increases

11. Average amount balance increases increases when number of loans also increases

12. More peoples are recharged their account and takes 4 times loan

13. As the age of cellular network increases, number of times loans taken by users are increases slightly.

14. Success rate is high when amount of taking loan is high

15. Amount of loans is high because Amount of daily spent is high

16. We can see clearly, amount of loan increases, when the total amount of recharge in main account increases.

17. success is higher for users taking total amount of loans in last 90 days

18. amount is higher that’s why users taking more amount of loans

19. Main account balance is high because users taking more amount of loans

13. Run and Evaluate selected models:

- Logistic Regression

```
Accuracy_score of Logistic regression:-------->   0.8944000669148091
Confusion_matrix:
 [[  300  4971]
 [   79 42472]]
Classification_report:
              precision    recall  f1-score   support

           0       0.79      0.06      0.11      5271
           1       0.90      1.00      0.94     42551

    accuracy                           0.89     47822
   macro avg       0.84      0.53      0.53     47822
weighted avg       0.88      0.89      0.85     47822
```

- Decision Tree Classifier

```
Accuracy_score of Decision Tree:-------->   0.8848228848647066
Confusion_matrix:
 [[ 2689  2582]
 [ 2926 39625]]
Classification_report:
               precision    recall  f1-score   support

           0       0.48      0.51      0.49      5271
           1       0.94      0.93      0.94     42551

    accuracy                           0.88     47822
   macro avg       0.71      0.72      0.71     47822
weighted avg       0.89      0.88      0.89     47822
```

- Random Forest Classifier

```
Accuracy_score of Random forest:-------->   0.92106143615909
Confusion_matrix:
 [[ 2352  2919]
 [  856 41695]]
Classification_report:
               precision    recall  f1-score   support

           0       0.73      0.45      0.55      5271
           1       0.93      0.98      0.96     42551

    accuracy                           0.92     47822
   macro avg       0.83      0.71      0.76     47822
weighted avg       0.91      0.92      0.91     47822
```

- KNN Classifier

```
Accuracy_score of KNeighbors Classifiers:-------->   0.91175609552089
Confusion_matrix:
 [[ 2330  2941]
 [ 1279 41272]]
Classification_report:
               precision    recall  f1-score   support

           0       0.65      0.44      0.52      5271
           1       0.93      0.97      0.95     42551

    accuracy                           0.91     47822
   macro avg       0.79      0.71      0.74     47822
weighted avg       0.90      0.91      0.90     47822
```

- Support Vector Classifier

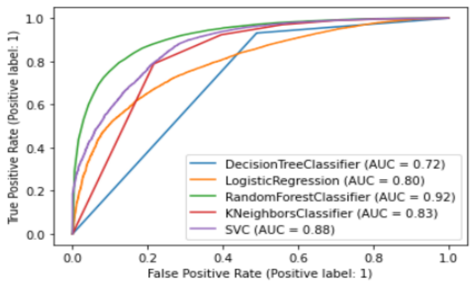
```
Accuracy_score of Support Vector Machine:-------->   0.9152900338756221
Confusion_matrix:
 [[ 1749  3522]
 [  529 42022]]
Classification_report:
               precision    recall  f1-score   support

           0       0.77      0.33      0.46      5271
           1       0.92      0.99      0.95     42551

    accuracy                           0.92     47822
   macro avg       0.85      0.66      0.71     47822
weighted avg       0.91      0.92      0.90     47822
```

We can see the CV scores of 5 models below:

```
Logistic regression CV Score:
0.8939493750163916

 **************************************************
Decision Tree CV Score:
0.8852451283754915

 **************************************************
Random Forest CV Score:
0.9212958613230793

 **************************************************
KNeighbour Classifier CV Score:
0.9113003593394055

 **************************************************
Support Vector Machine CV Score:
0.9162353743908837
```

Plot ROC/AUC for multiple models

```
#how well out model works on test data
disp = plot_roc_curve(dt,x_test,y_test)
plot_roc_curve(log_reg,x_test,y_test, ax= disp.ax_)
plot_roc_curve(rf,x_test,y_test, ax= disp.ax_)
plot_roc_curve(knn,x_test,y_test, ax= disp.ax_)
plot_roc_curve(svc,x_test,y_test, ax= disp.ax_)
plt.show()
```



Observations:

We can conclude easily now by observing the Accuracy scores, CV Scores, difference between the Accuracy scores and cv score, and AUC Score, We can say Random Forest is our best model because it is giving us best parameters.

14. Key Metrics for success in solving problem under consideration:

The dataset is unbalanced with 87.5% of label 1 and 12.5% of label 0, which made it clear that, we cannot blindly rely on accuracy score for the prediction as it can lead to biasness. Hence, I have used confusion matrix and AUC ROC curve to determine the accuracy of the model.

15. Interpretation of the Results:

From the dataset, it was clear that most of the customers are inclined to pay the loan as 87.5% of the customer repaid it and only 12.5% of the customers are defaulter.

CONCLUSION

- **Key Findings and Conclusions of the Study** Mostly, the customers have the intension of repaying. There are certain cases, when the customers have no intension of repayment but the number of such customers are few. With the model built, we can certainly determine customers having intension of repayment or not.
- **Learning Outcomes of the Study in respect of Data Science** The dataset was full of outliers, skewness and unbalanced data which was the biggest challenge to overcome. Hence data cleaning was very important to get proper prediction. I have used Logistic

Regression, Decision Tree, Support vector machine, KNN classifier, Bagging classifier and Random Forest Classifier. Among all of them algorithms Random Forest Classifier gave the best outcome. As the dataset was unbalanced, the other algorithm may overfit and can come out with wrong prediction whereas Random forest can control overfitting and give best prediction.

- **Limitations of this work and Scope for Future Work** The solution can be applied to the customer having a transaction history but the model may not perform well with customer having new profile and no transaction history. Nevertheless, the model will perform well with customer having transaction history and can predict whether a person will be a defaulter or non-defaulter. Hence, we can say that this statistical model will be helpful in future for the prediction of micro credit defaulter and non-defaulter customer.