MICRO CREDIT DEFAULTER PROJECT

Submitted by:

SUDHANSU MANDAL

INTRODUCTION

➢ Business Problem Framing

A Microfinance Institution (MFI) is an organization that provide Microfinance Services (MFS) to low income populations. MFS helps poor families who is having less source of income and who lives in remote areas. The Microfinance services provided by Microfinance Institution includes services like Group Loans, Agricultural Loans, Individual Business Loan and so on.
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.

MFI is working with Telecom Industry who is dealing with fixed wireless telecommunications network provider. They have launched various products and have developed its business and organization based on the budget operator model, 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 their impact on person's life, thus focusing on providing their services and products to low income families and poor customer that can help them in the need of hour.
So, its very important to chooses their customer for the credit, they need some predictions that could help them in further investment and improvement in selection of customer.
Identifying the default customer is very important step to improve their business and targeting right segment of customer.
Here, if the customer is deviates from the path of paying back the loaned amount within 5 days, then the customer will classify as defaulter. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah) and for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah).

Here the main objective would be building some model which would help them to select the customer for credit and also to identify most important features which will classify customer as defaulter or not.

We have here 2 class 0 and 1 where class '1' indicates that the loan has been payed i.e. Non-defaulter, while, Label '0' indicates that the loan has not been payed i.e. defaulter.

➢ Conceptual Background of the Domain Problem

Earlier loan lending companies used to risk a high rate of defaulting. Some time after approved for loan even perfect candidate also shows very irregular repayment behaviours, such kind of incidents used to be very big challenges for lenders to identify the perfect candidate and also to predict the future whether the candidate would able to repayment on schedule time. So, Machine Learning can help lenders to predict potential defaulter before loan approved with the help of candidature past data like candidate's income, past debt and repayment behaviour.

➢ Review of Literature

This project observed whether candidate is defaulter or not and to identify those features or factor which impact on candidate defaulter.

1. Spend Analyser of User, we will analyse that how many times and how much amount spend on recharge and whether these factors having any relationship with candidate defaulter.
2. Recharge pattern of main account and data account.
3. Loan taken history, how frequently and amount taken for loan.
4. Payback time for loan repayment, whether lenders paying within 5 days or not. If not paying within 5 days then we will call as defaulter.
5. We will build model which can predict candidate defaulter or not based on feature input.

➢ Motivation for the Problem Undertaken

Microfinance Institution play very important role in society which help candidate financially by fulfilling their immediate requirement of finance. Today there are many companies provide loan to the needy people, which help people to enhance their lifestyle.

But providing loan to lenders is very critical and important task to analyse whether candidate can repay the loan on time or not, so based on this, they approve the loan. Here Data Science can help lenders to decide candidate eligibility upon approval of loan based on various history financial or transaction details of the candidate.

ANALYTICAL PROBLEM FRAMING

➢ Mathematical/ Analytical Modeling of the Problem

In this project our dataset having 209593 observation and 37 columns or features including target variable. To understand our data, we have done basic data exploration.

- **Statistic Summary**

Here we have tried to understand the Statistic of data with the help of describe () function. Mention below are the observation made from this summary:

1. No null value in dataset.
2. Unnamed:0 column need to drop as this column would not require for our analysis.
3. We can analyse that data distribution for columns are not normal as there are difference in value of mean and 50% data value.
4. There are some columns where values are in negative so we need to deal this.

- **Feature Selection**

  We have seen that 36 features in dataset, many of which have analysed that not necessary or not much information about the target variable. We have to select only most significant features to make Machine learning models more efficient and cost effective. The method used was “Univariate Selection” using chi-square test.

- **Data Visualization**

  We have found two important insights a. Imbalance of data and b. Distribution not normal.

- **Data Normalization**

  Since the data was not normal, we have normalized all the features except the target variable which was dichotomous (Values '1' and '0').

- **Data Imbalanced**

  Oversampling of Minority class Since the data was expensive, we don’t want to lose out on data by under sampling the majority class. Instead, we have decided to oversample the minority class.

➢ Data Sources and their formats

The data sources have been provided to us in .csv format and also excel file of having details description about the dataset. The .csv file we have imported and converted into Pandas dataframe for analysing purpose. The dataset columns having float, int and object data type as shown in Fig 1.

Fig: 1

```
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
```

➢ Data Preprocessing

Below is the step for data Preprocessing done:

- **Checking unique value of dataset.**
  As shown in Fig 2, we have checked the unique value in dataset, which helps us to know the categorical features level in the particular columns.

Fig: 2

```
Out[8]: label                      2
        msisdn                186243
        aon                     4507
        daily_decr30          147025
        daily_decr90          158669
        rental30              132148
        rental90              141033
        last_rech_date_ma       1186
        last_rech_date_da       1174
        last_rech_amt_ma          70
        cnt_ma_rech30             71
        fr_ma_rech30            1083
        sumamnt_ma_rech30      15141
        medianamnt_ma_rech30     510
        medianmarechprebal30   30428
        cnt_ma_rech90            110
        fr_ma_rech90              89
        sumamnt_ma_rech90      31771
        medianamnt_ma_rech90     608
        medianmarechprebal90   29785
        cnt_da_rech30           1066
        fr_da_rech30            1072
        cnt_da_rech90             27
        fr_da_rech90              46
        cnt_loans30               40
        amnt_loans30              48
        maxamnt_loans30         1050
        medianamnt_loans30         6
        cnt_loans90             1110
        amnt_loans90              69
        maxamnt_loans90            3
        medianamnt_loans90         6
        payback30               1363
        payback90               2381
        pcircle                    1
        pdate                     82
        dtype: int64
```

- **Checking for null value**

As its already has mentioned in problem statement that data doesn't have any null value.

- **Dealing with Negative Value**

As we have seen that there are lot of negative value in columns like aon, daily_decr30, daily_decr90, rental30, rental90, last_rech_date_ma, last_rech_date_da, medianmarechprebal30, medianmarechprebal90.So to avoid any data loss we have not removed these observations from the dataset. Instead we have converted these negative values into 0 for our analysis.

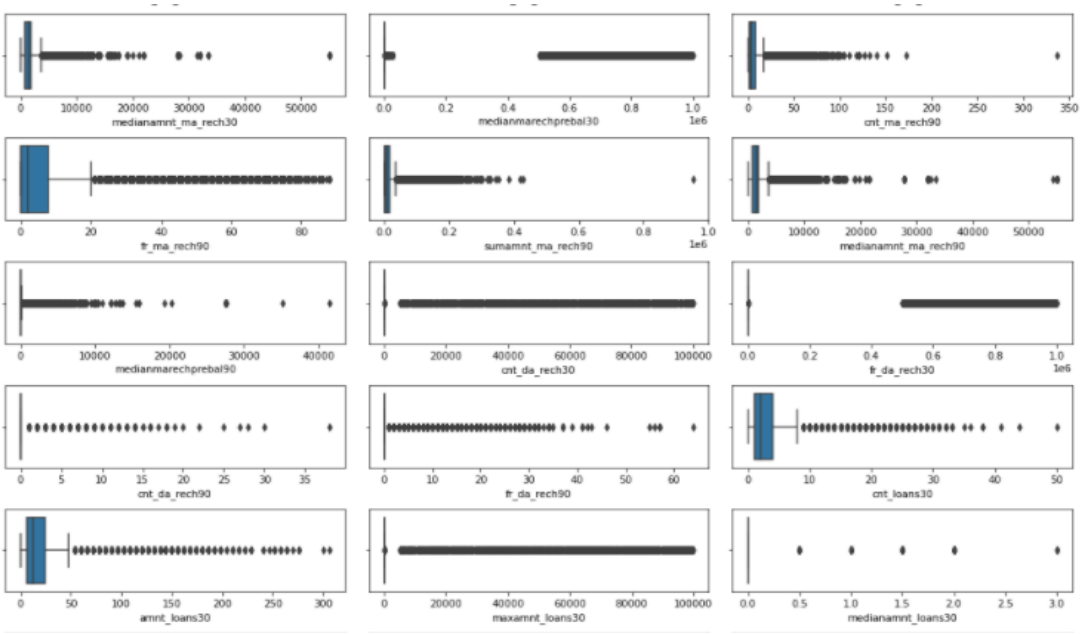
- **Dropping unwanted Columns**

As there are some columns which has only for identification purpose or those columns we believe that those columns are not have any significant for our target variable so we have removed from the dataset. Those columns are Unnamed 0, aon, pcircle and pdate.

- **Checking for outlier's presence in Dataset.**

In EDA we have visualized that as shown in Fig 3, that there are lot of columns where outliers presence and columns data are not normal distributed.

Fig: 3



- **Dealing with Outliers**

Since there were lot of Outliers present in dataset and we don't want to lose any useful information so we have used mean Imputation technique as shown in Fig 4.

Fig: 4

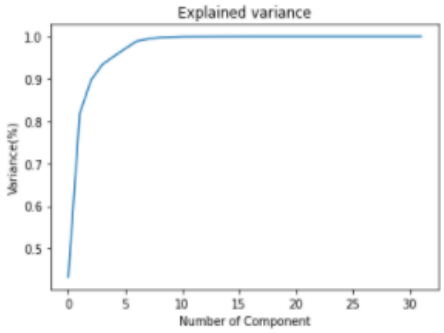
```
upper_limit=data["daily_decr90"].mean()+3*data["daily_decr90"].std()
lower_limit=data["daily_decr90"].mean()-3*data["daily_decr90"].std()

data["daily_decr90"]=np.where(data["daily_decr90"]>upper_limit,
                        upper_limit,
                       np.where(
                           data["daily_decr90"]<lower_limit,
                           lower_limit,
                           data["daily_decr90"]
                       )
                   )
```

➢ Data Inputs- Logic- Output Relationships

In this dataset, our input variable having int, float and object data type. Our target variable is binary data type.
We have seen there are 36 features and 1 target variable in dataset. After doing PCA we can see more than 90% variance explained with 7 features as shown in Fig 5.

Fig: 5



➢ Hardware and Software Requirements and Tools Used

In this project the mention below hardware, software and tools used to complete this project:

- ✓ Hardware:

Processor	Intel(R) Pentium(R) CPU 3825U @ 1.90GHz 1.90 GHz
Installed RAM	4.00 GB
System type	64-bit operating system, x64-based processor

- ✓ Software:

Edition	Windows 10 Home Single Language

Anaconda

- ✓ Library & Tools:

Jupyter Notebook
Pandas
NumPy
Matplotlib
Seaborn

Model/s Development and Evaluation

➢ Identification of possible problem-solving approaches (methods)

As we have seen that our target variable is in binary type, where we have two classes 0 and 1. So we need to apply classification task to the dataset.
We have seen that our data is not distributed normally and having outliers, so we have normalized the features columns with **Standard Scaler**.
Also, we have done PCA **(Principal component Analysis)** for dimension reduction of our feature's dataset, as shown in fig 5, we can choose 7 components for our model, but we will see the performance of model without removing any features and then will decide whether need to use dimension reduction technique or not.
We have seen that without dimension reduction technique model is performing better.

➢ Testing of Identified Approaches (Algorithms)

As we know our target variable having binary class, so we need apply classification algorithm for model building.
Below are the classification algorithms have applied to the model.

1. Logistic Regression
2. Random Forest Classifier
3. Support Vector Classifier

We have applied mentioned above model in our training and test dataset. Here in project we have split training 70% and testing 30% of data.

➢ Run and Evaluate selected models

As shown in Fig 6, we have separated the training and test data set into 70% and 30%.

Fig: 6

```
In [293]: x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.30,random_state=74)
```

Here we have chosen Best Random State for the model as shown in Fig 7.

Fig: 7

```
In [1751]: for i in range(0,100):
               x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.30,random_state=i)
               lr.fit(x_train,y_train)
               pred_train=lr.predict(x_train)
               pred_test=lr.predict(x_test)

               if round(accuracy_score(y_train,pred_train)*100,1)==round(accuracy_score(y_test,pred_test)*100,1):
                   print("At random state",i,"Model perfrom well")
                   print("At random state",i)
                   print("Training accuracy score is-",accuracy_score(y_train,pred_train)*100)
                   print("Testing accuracy score is-",accuracy_score(y_test,pred_test)*100)
```

Now we have evaluated our model, based on Accuracy Score, Confusion Matrix and Classification Report.

1. Logistic Regression

Evaluation Matrix:

Fig: 8

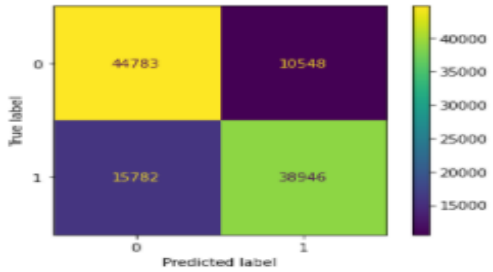
```
In [296]: y_pred=lr.predict(x_test)

In [297]: lr_acc_score=accuracy_score(y_test,y_pred)

In [298]: lr_acc_score
Out[298]: 0.7607646807621367

In [299]: confusion_matrix(y_test,y_pred)
Out[299]: array([[44783, 10548],
                 [15782, 38946]], dtype=int64)

In [300]: print("Logistic Regression")
          plot_confusion_matrix(lr,x_test,y_test,values_format=".6g");
Logistic Regression
```



Here we can see the Logistic Regression Accuracy Score is 76%.

The actual values are:

The Candidate who actually are not defaulter =55251
The Candidate who actually are defaulter = 54728

The Predicted Values are:

The Candidate who actually are not defaulter = 60565
The Candidate who actually are defaulter = 49494

2. Random Forest Classification

Evaluation Matrix:

Fig: 9

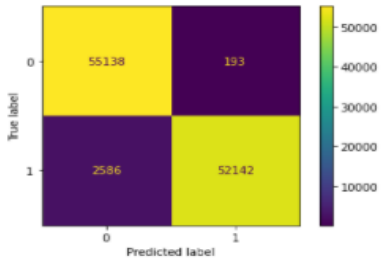
```
In [186]: y_pred=rfc.predict(x_test)

In [187]: rfc_acc_score=accuracy_score(y_test,y_pred)

In [188]: rfc_acc_score
Out[188]: 0.9747499068681343

In [189]: confusion_matrix(y_test,y_pred)
          pd.crosstab(y_test, y_pred, rownames = ['Actual'], colnames =['Predicted'], margins = True)
Out[189]: array([[55138,   193],
                 [ 2586, 52142]], dtype=int64)

In [190]: print("Random Forest Classifier")
          plot_confusion_matrix(rfc,x_test,y_test,values_format=".6g");
Random Forest Classifier
```




```
In [191]: print("Random Forest Classifier")
          print(classification_report(y_test,y_pred,digits=2))
Random Forest Classifier
              precision    recall  f1-score   support

           0       0.96      1.00      0.98     55331
           1       1.00      0.95      0.97     54728

    accuracy                           0.97    110059
   macro avg       0.98      0.97      0.97    110059
weighted avg       0.98      0.97      0.97    110059
```

3. Support Vector Classification

Evaluation Matrix:

Fig: 10

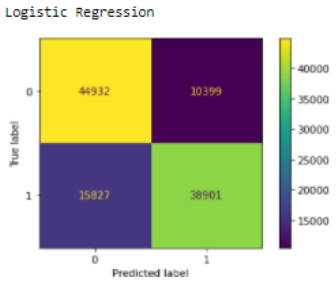
➢ Key Metrics for success in solving problem under consideration

1. Logistic Regression:

Accuracy Score = 76%

Confusion Matrix

Fig: 12



Classification Report

Fig: 13

```
In [301]: print("Logistic Regression")
          print(classification_report(y_test,y_pred,digits=2))

          Logistic Regression
                        precision    recall  f1-score   support

                     0       0.74      0.81      0.77     55331
                     1       0.79      0.71      0.75     54728

              accuracy                           0.76    110059
             macro avg       0.76      0.76      0.76    110059
          weighted avg       0.76      0.76      0.76    110059
```

Precision

As shown in Fig 13, the candidate is defaulter 79% of time is correct and the candidate is not defaulter 74% of time is correct.

Recall

As shown in Fig 13, it measures how accurately model is to identify the relevant data. This is very important because model predicted candidate is not defaulter and actually candidate is defaulter then for lender recovering would be very difficult hence the institute would make loss.

2. Random Forest Classifier:

   Accuracy Score = 97%

   Confusion Matrix:

   Fig: 14

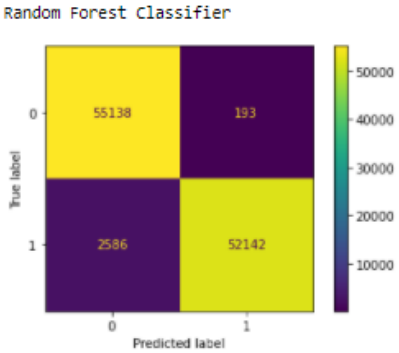


   Fig: 15

Random Forest Classifier

	precision	recall	f1-score	support
0	0.96	1.00	0.98	55331
1	1.00	0.95	0.97	54728
accuracy			0.97	110059
macro avg	0.98	0.97	0.97	110059
weighted avg	0.98	0.97	0.97	110059

Precision

As shown in Fig 15, the candidate is defaulter 100% of time is correct and the candidate is not defaulter 96% of time is correct.

Recall

As shown in Fig 15, it measures how accurately model is to identify the relevant data. This is very important because model predicted candidate is not defaulter and actually candidate is defaulter then for lender recovering would be very difficult hence the institute would make loss.

3. Support Vector Machine Classifier:

   Evaluation Matrix:

   Fig 16:

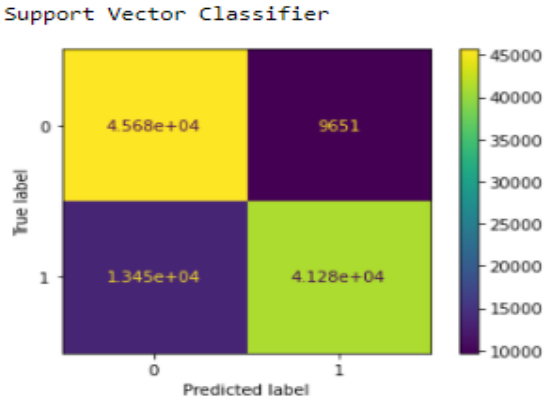


Fig 17:

```
In [205]: print("Support Vector Classifier")
          print(classification_report(y_test,y_pred,digits=2))

Support Vector Classifier
              precision    recall  f1-score   support

           0       0.77      0.83      0.80     55331
           1       0.81      0.75      0.78     54728

    accuracy                           0.79    110059
   macro avg       0.79      0.79      0.79    110059
weighted avg       0.79      0.79      0.79    110059
```

Precision

As shown in Fig 17, the candidate is defaulter 81% of time is correct and the candidate is not defaulter 77% of time is correct.

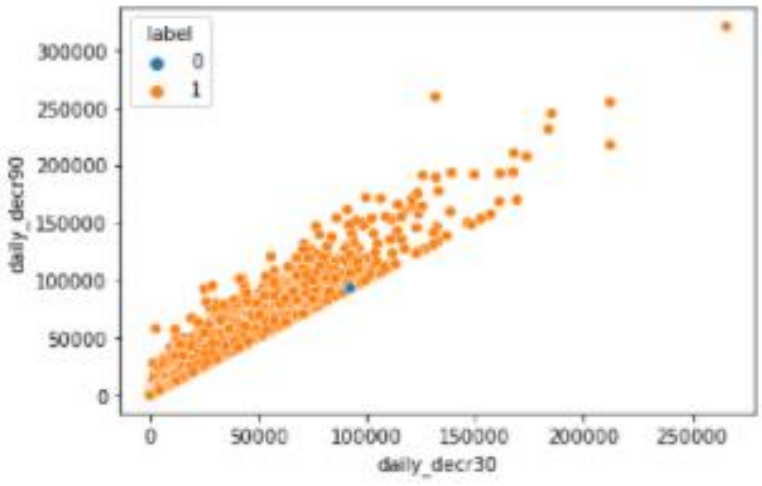
Recall

As shown in Fig 17, it measures how accurately model is to identify the relevant data. This is very important because model predicted candidate is not defaulter and actually candidate is defaulter then for lender recovering would be very difficult hence the institute would make loss.

➢ Visualizations

- **Average Daily Amount Spent**

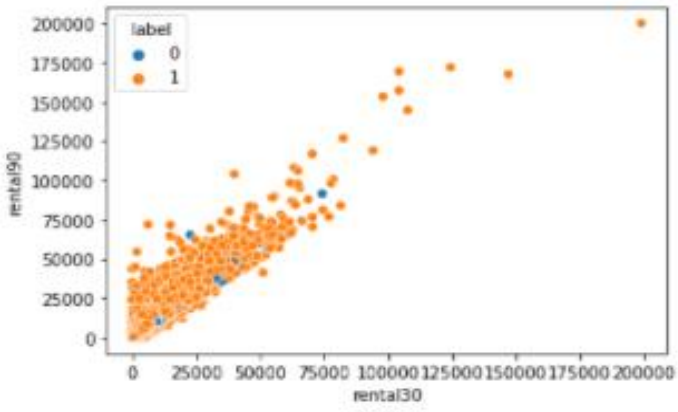
Fig: 20



As we can infer from fig 20 that as daily avg spend amount for 30 days and 90 days is increasing more candidate are defaulter.

- **Average Main Account Balance**

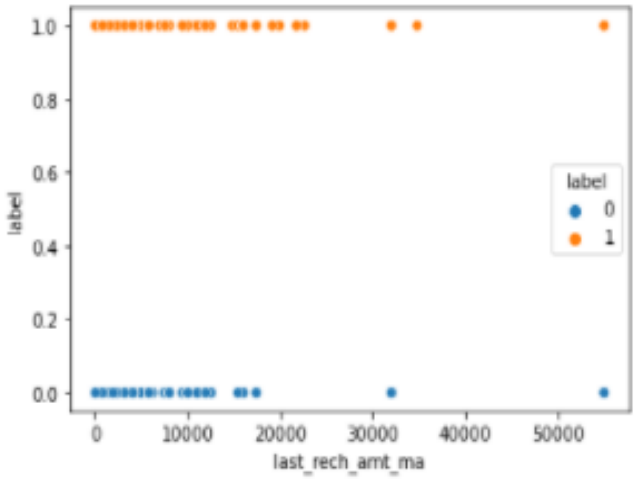
  Fig: 21



  As we can infer from Fig 21, that Avg main account balance for 30 days and 90 days increased we see that more candidate become defaulter.

- **Amount of Last recharge main account**

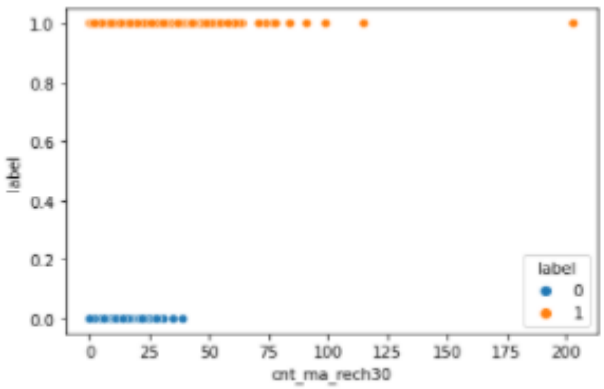
  Fig: 22



  As we can infer from above fig 22, more amount recharge done main account more candidates are defaulter.

- **Number of time main account recharged**

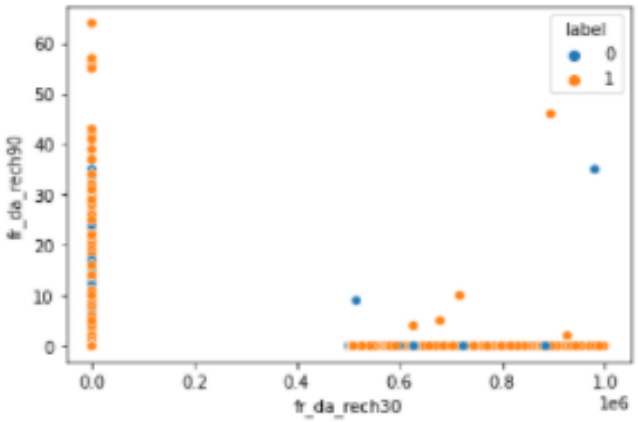
  Fig: 23



We can infer from above Fig 23, more time main account recharged more likely to be defaulter.

- **Frequency of data account recharged**

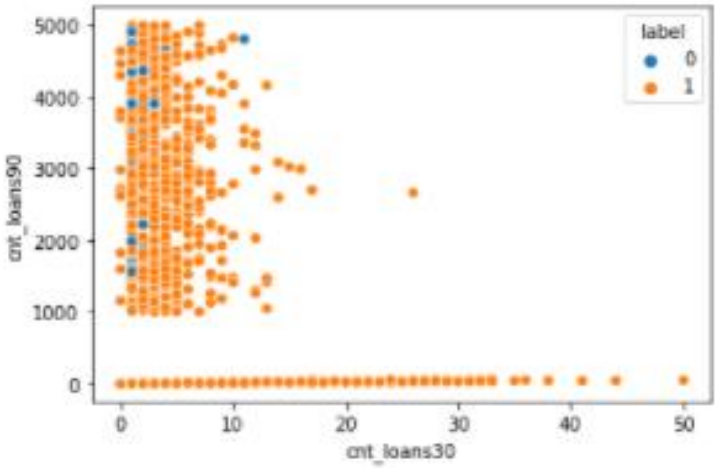
  Fig: 24



We can infer from Fig 24 that less frequency of data account recharged done more likely to be defaulter.

- **Number of loans taken by user**

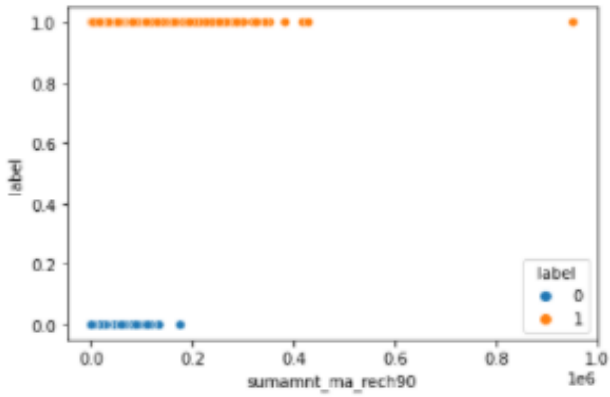
  Fig: 25



We can infer from fig 25 that number of loans taken for 90 days more likely to be defaulter.

- **Total Amount recharge main account**

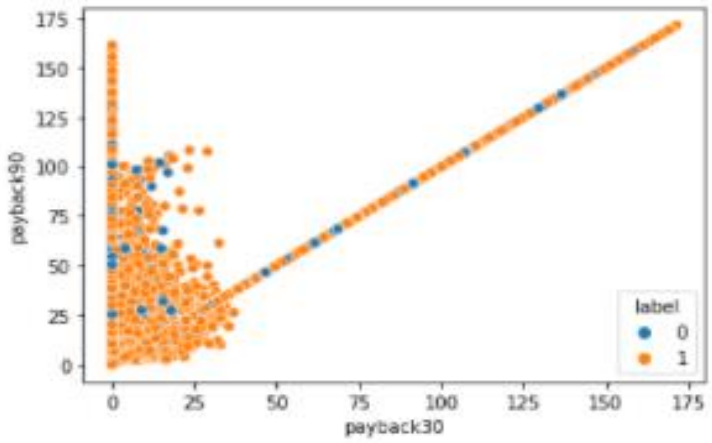
  Fig: 24



We can infer that more amount recharged done main account candidate is more likely to defaulter.

- **Average payback time in days**

Fig: 25



We can infer from Fig 25 that Avg payback time in days for 30 is less more likely to be defaulter.

➢ Interpretation of the Results

Below are the interpreted from the visualizations and modelling:

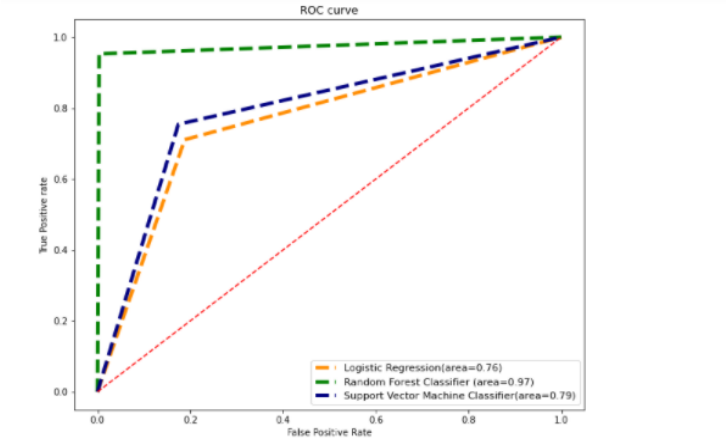
- ✓ We can infer from accuracy score that Random Forest Classifier is performing well in this dataset.
- ✓ As shown in Fig 15, the candidate is defaulter 100% of time is correct and the candidate is not defaulter 96% of time is correct.
- ✓ As shown in Fig 15, it measures how accurately model is to identify the relevant data. This is very important because model predicted candidate is not defaulter and actually candidate is defaulter then for lender recovering would be very difficult hence the institute would make loss.
- ✓ As we can infer from fig 20 that as daily avg spend amount is increasing more candidate are defaulter
- ✓ As we can infer from Fig 21, that Avg main account balance increased we see that more candidate become defaulter.
- ✓ As we can infer from above fig 22, more amount recharge done main account more candidates are defaulter.
- ✓ We can infer from above Fig 23, more time main account recharged more likely to be defaulter.
- ✓ We can infer from Fig 24 that less frequency of data account recharged done more likely to be defaulter.
- ✓ We can infer from Fig 25 that Avg payback time in days for 30 is less more likely to be defaulter.

CONCLUSION

➢ Key Findings and Conclusions of the Study

- ✓ We have shown in Fig 17; Random Forest Classifier is performing better compared to Logistic Regression and Support Vector Classifier.
- ✓ While doing Principal Components Analysis, as shown in Fig 5 we can go with 7 components to get more than 90% of variance explained.
- ✓ While doing visual analysis we have got like daily avg spend amount, Avg main account balance, amount recharge, main account recharged, frequency of data account, Avg payback time in days having more impact on candidate's defaulter.
- ✓ We have Confirmed that based on AUC ROC Curve Random Forest Classifier is performing better. as shown in Fig 26.

Fig: 26



- ✓ To avoid model overfitting, we have checked Cross Validation Score with different CV. And have selected the best model which ever accuracy score and cv score is lesser as shown in below Fig.

Fig: 27

Out[230]:

	Accuracy Score	Cross Validation Score	Difference
LogisticRegression	0.76171	0.759441	0.002269
RandomForestClassifier	0.97475	0.978510	-0.003760
SupportVectorMachineClassifier	0.79014	0.787967	0.002173

➢ Learning Outcomes of the Study in respect of Data Science

Below are the Learning Outcome of the Study:

- ✓ Handling huge dataset more than 2 lakh rows.
- ✓ Having more features and through PCA can select significant features only for model.
- ✓ Feature data pattern are not normally distributed.
- ✓ Having lots of Outliers so we need to choose mean Imputation to avoid data loss.
- ✓ Target variable are highly imbalanced, so we need to apply upscaling technique to avoid any data loss.
- ✓ Selecting the Best Model based on AUC ROC Curve plot and Cross validation score.

➢ Limitations of this work and Scope for Future Work

Mention below are the limitations and future scope steps:

- ✓ Data set having more than 2 lakhs after treating target variable imbalanced, so we can select some sample rows/Observation instead of entire dataset rows/Observation.
- ✓ Due to huge dataset observation here in the project have trained with limited classification model due to time constrained and huge computation power require. We can train model on Decision tree, Boosting Algorithm like ada booster, Gradient boosting classifier.