FLIP ROBO

MICRO CREDIT DEFAULTER PROJECT

Submitted by:

Siddhant Sethi

ACKNOWLEDGMENT

This is to mention that extensive help was taken from Datatrained notes to help implement the model and gain valuable insights. Besides it, new balancing of dataset technique was learnt through the internet papers. I would like to thank the FLIPROBO team along with my mentor Mr. Mishra for their guidance throughout the project along with providing the dataset and documentation sample to make the process exciting and enriching.

INTRODUCTION

- Business Problem

The problem relates to the field of telecom industry whereby a company is trying to leverage the upcoming unconventional type of banking in the form of Micro-finance institution. The company seeks to implement a business model in which it can help the poor and needy to continue with its services by providing them a loan of 5 or 10(Indonesian rupiah). The condition is that the customer ought to return the amount (loan + charges) within 5 days to avoid the default. Keeping this in mind, we are required to analyse and build a model to make predictions using the dataset provided.

This is a direct practical problem which can be related to wide areas as well that include banking services. Very often we hear about growing NPA's in the context of Indian banking sector. Thus, the problem is very relatable to predict a default in future or not.

- Conceptual Background

Since the problem explicitly relates to telecom industry along with the usage of banking activities, it would be very useful to make ourselves accustomed to some basic knowledge of the respective fields. For the telecom part, we would come across with terms like phone number, the no of days for which the connection is active, the amount of balance, how much is spent on call and data, what was the recharge amount, the no of times the recharge is performed, what telecom circle does the connection falls in and the date of it.

In the context of telecom sector, we would like to know the amount of loan taken, the frequency of loans taken, what was the max. and average value of amounts, also the no of days to payback the complete amount

- Review of Literature

  The research included understanding the domain knowledge of the two sectors-telecom and MFIs to get an overall grip of the problem. This included understanding the features of the dataset and whether it made complete sense of the data provided. Some unnecessary columns were to be dropped and some strange values that appeared had to be dealt with. For this, extensive usage of internet was done to be able to make sense of the parameters provided in the dataset. The following features had some unrealistic values: -

  last_rech_date_ma, last_rech_date_da, cnt_ma_rech30, fr_ma_rech30, cnt_ma_rech90, fr_ma_rech90, cnt_da_rech30, fr_da_rech30, cnt_da_rech90, fr_da_rech90, cnt_loans30, maxamnt_loans30, cnt_loans90

- Motivation for the Problem Undertaken

  The objective behind the problem is to build a model to make predictions if the customer would be able to return the loan amount within 5 days or not. This would help the company to better cater to its business model and keep its books balanced and profit on an increasing curve.
  Besides, the motivation was to be a part of a great practical problem that has wider learnings in the machine learning field. Thus, the insights gathered and the model was a great learning and application-based experience.

Analytical Problem Framing

- ## Mathematical/ Analytical Modelling of the Problem

The problem being a classification problem, some important classification algorithms were imported to jupyter interface. Being a large dataset, a general fit was done using the algorithms. The following algorithms were used: -

GaussianNB(), LogisticRegression(), DecisionTreeClassifier(), KNeighborsClassifier(), RandomForestClassifier(), GradientBoostingClassifier(), AdaBoostClassifier(base_estimator=dtc,lr)

SMOTE was used to balance the dataset and the balanced data was fed into the train_test_split. For the metrics, accuracy score, cross validation score, roc_auc score, classification report and confusion matrix were derived. The best performance was judged and it was followed by hyperparameter tuning using RandomizedSearchCV (for the convenience of large dataset). Then, an optimum random state had to be found using a for loop in (42,100) range in splitting function. Thus, now we were equipped with the necessary weapons to be able to get the final result. Finally, the model was saved using joblib.dump.

- ## Data Sources and their formats

A csv file was provided in the form of dataset as the problem was provided by the client to look into insights and build a model for making predictions regarding default of loan amount. The dataset had 37 columns in all along with 209593 rows. The dtypes of the features were as follows

```
df.dtypes
```

```
Unnamed: 0                int64
label                     int64
msisdn                   object
aon                     float64
daily_decr30            float64
daily_decr90            float64
rental30                float64
rental90                float64
last_rech_date_ma       float64
last_rech_date_da       float64
last_rech_amt_ma          int64
cnt_ma_rech30             int64
fr_ma_rech30            float64
sumamnt_ma_rech30       float64
medianamnt_ma_rech30    float64
medianmarechprebal30    float64
cnt_ma_rech90             int64
fr_ma_rech90              int64
sumamnt_ma_rech90         int64
medianamnt_ma_rech90    float64
medianmarechprebal90    float64
cnt_da_rech30           float64
fr_da_rech30            float64
cnt_da_rech90             int64
fr_da_rech90              int64
cnt_loans30               int64
amnt_loans30              int64
maxamnt_loans30         float64
medianamnt_loans30      float64
cnt_loans90             float64
amnt_loans90              int64
maxamnt_loans90           int64
medianamnt_loans90      float64
payback30               float64
payback90               float64
pcircle                  object
pdate                    object
dtype: object
```

Th data description of the aforesaid features were as follows: -

- Label=default or non-default
- Msisdn= number
- Aon= cellular network(days)
- daily_decr30= daily amount spent from main account over last 30 days
- daily_decr90= daily amount spent from main account over last 90 days
- last_rech_date_ma= days till last recharge of main account
- last_rech_date_da= days till last recharge of data account
- last_rech_amt_ma=amount of last recharge of main
- cnt_ma_rech30= no of times main acct got recharged in last 30 days
- sumamnt_ma_rech30= Total amount of recharge in main account over last 30 days (

- medianamnt_ma_rech30= Median of amount of recharges done in main account over last 30 days
- medianmarechprebal30= Median of main account balance just before recharge in last 30 days
- cnt_ma_rech90= Number of times main account got recharged in last 90 days
- sumamnt_ma_rech90= Total amount of recharge in main account over last 90 days
- medianamnt_ma_rech90= Median of amount of recharges done in main account over last 90 days
- medianmarechprebal90= Median of main account balance just before recharge in last 90 days
- cnt_da_rech30= Number of times data account got recharged in last 30 days
- fr_da_rech30= Frequency of data account recharged in last 30 days
- cnt_da_rech90= Number of times data account got recharged in last 90 days
- fr_da_rech90= Frequency of data account recharged in last 90 days
- cnt_loans30= Number of loans taken by user in last 30 days
- amnt_loans30= Total amount of loans taken by user in last 30 days
- maxamnt_loans30= maximum amount of loan taken by the user in last 30 days
- medianamnt_loans30= Median of amounts of loan taken by the user in last 30 days
- cnt_loans90= Number of loans taken by user in last 90 days
- amnt_loans90= Total amount of loans taken by user in last 90 days
- maxamnt_loans90= maximum amount of loan taken by the user in last 90 days
- medianamnt_loans90= Median of amounts of loan taken by the user in last 90 days
- payback30= Average payback time in days over last 30 days
- payback90= Average payback time in days over last 90 days
- pcircle= telecom circle
- pdate= date

- ## Data Pre-processing Done

  To clean the data, following steps were followed: -

  1. Heatmap was built to look out for any missing values
  2. Pcircle & unnamed columns were dropped (since pcircle had a singular value & of no use)
  3. Pdate was converted to numerical values by splitting it into year, month & day
  4. Visualizations such as violinplot, catplot, countplot and pairplot were drawn to gather insights
  5. The features showing unrealistic values were looked into and methods were thought of treating the same

6. Since the column of phone number had repeating values thus label encoder was used to convert object dtype into numerical value
7. Boxplot and distplot were drawn to look for any skewness and outliers
8. A threshold of five was chosen to remove outliers since the data was very expensive using zscore
9. Finally, the dataset was treated with logarithm function to treat for its skewness & then standard scaler was used to put the input values at same scale

- ## Data Inputs- Logic- Output Relationships

The format of data input was of the following type: -

```
df.dtypes
```

```
Unnamed: 0                int64
label                     int64
msisdn                   object
aon                     float64
daily_decr30            float64
daily_decr90            float64
rental30                float64
rental90                float64
last_rech_date_ma       float64
last_rech_date_da       float64
last_rech_amt_ma          int64
cnt_ma_rech30             int64
fr_ma_rech30            float64
sumamnt_ma_rech30       float64
medianamnt_ma_rech30    float64
medianmarechprebal30    float64
cnt_ma_rech90             int64
fr_ma_rech90              int64
sumamnt_ma_rech90         int64
medianamnt_ma_rech90    float64
medianmarechprebal90    float64
cnt_da_rech30           float64
fr_da_rech30            float64
cnt_da_rech90             int64
fr_da_rech90              int64
cnt_loans30               int64
amnt_loans30              int64
maxamnt_loans30         float64
medianamnt_loans30      float64
cnt_loans90             float64
amnt_loans90              int64
maxamnt_loans90           int64
medianamnt_loans90      float64
payback30               float64
payback90               float64
pcircle                  object
pdate                    object
dtype: object
```

The relationship between the input and output variables was found using catplot, pairplot & correlation matrix.

FINDINGS (The max count of the above parameters wrt label is as follows) -->

-Label=1 shows highest frequency in relation to year,month and day without any exception
High positive correlation exists between--
daily_decr30 v/s daily_decr90
rental30 v/s rental90
last_rech_amt_ma v/s medianamnt_ma_rech90
cnt_ma_rech30 v/s cnt_ma_rech90
sumamnt_ma_rech30 v/s sumamnt_ma_rech90
medianamnt_ma_rech30 v/s medianamnt_ma_rech90
cnt_da_rech30 v/s cnt_da_rech90
cnt_loans30 v/s amnt_loans30
cnt_loans30 v/s cnt_loans90
cnt_loans30 v/s amnt_loans90
amnt_loans30 v/s cnt_loans90
amnt_loans30 v/s amnt_loans90
maxamnt_loans30 v/s maxamnt_loans90
cnt_loans90 v/s amnt_loans90
payback30 v/s payback90

High negative correlation exists between
last_rech_date_ma v/s cnt_ma_rech30
last_rech_date_ma v/s cnt_loans30
month v/s day

- The set of assumptions

The following columns had unrealistic data entries that were recurring: -

- last_reach_date_ma
- last_rech_date_da
- cnt_ma_rech30
- fr_ma_rech30
- cnt_ma_rech90
- fr_ma_rech90
- cnt_da_rech30

- fr_da_rech30
- cnt_da_rech90
- fr_da_rech90
- cnt_loans30
- maxamnt_loans30
- cnt_loans90

Threshold of 100 value was chosen since all the above features are expected to have a double-digit figure. Thus all these flawed values were treated with the respective median.

- ## Hardware and Software Requirements and Tools Used

Hardware requirements: -

- i5 processor
- 8 GB RAM

Software requirements: -

- Anaconda
- Jupyter interface
- Libraries & tools: numpy, pandas, matplotlib.pyplot, seaborn, warnings, statistics, LablelEncoder, sklearn, scipy.stats, zscore, StandardScaler, train_test_split, KNeighborsClassifier, accuracy_score,confusion_matrix,classification_report, GradientBoostingClassifier, SVC, RandomForestClassifier, AdaBoostClassifier, LogisticRegression, DecisionTreeClassifier, GaussianNB, joblib, imblearn.over_sampling, SMOTE, RandomizedSearchCV, roc_auc_score, roc_curve,auc, cross_val_score

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  The following analytical approach was followed: -

  - The data pre-processing was done followed by EDA & finally insights were gathered using various visualizations. Also, using the correlation matrix the input features which were having high positive correlation amongst themselves, some of those columns were dropped to avoid excessive multicollinearity

  The following statistical approach was used: -

  - A statistical summary was sought using df.describe() function. By using this, we got to know about respective standard deviations, the relation between mean & medians which tells us about the nature of skewness, and also about the relative distance between 75th percentile and the maximum value. This gives us an idea about the underlining outliers in the dataset
  - Since our dataset was of imbalanced nature, SMOTE technique was used to balance it. These balanced parameters were then fed into train_test_split. Now using various classification algorithms, a general fit was made and various metric scores were calculated. This helped us to finalize our best performing algorithm. Then, we continued by using RandomizedSearchCV. Also, by using the optimum random state, we reached our final model that was used to predict xtest

- Testing of Identified Approaches (Algorithms)

  The following algorithms were used in the project: -

  - GaussianNB()
  - LogisticRegression()

- DecisionTreeClassifier()
- KNeighborsClassifier()
- RandomForestClassifier()
- GradientBoostingClassifier()
- AdaBoostClassifier(base_estimator=dtc)
- AdaBoostClassifier(base_estimator=lr)

- Run and Evaluate selected models

  Algorithms are as listed above

  Snapshot of code: -

```
models=[]
models.append(('GNB',gn))
models.append(('LogisticRegression',lr))
models.append(('DTC',dtc))
models.append(('kNeighborsClassifier',knc))
models.append(('RFC',rfc))
models.append(('GBC',gbc))
models.append(('ADACLASS1',ada1))
models.append(('ADACLASS2',ada2))
```

```
Model=[]
score=[]
cvs=[]
rocscore=[]
for name,model in models:
    print('****',name,'****')
    print('\n')
    Model.append(name)
    model.fit(X_train,y_train.ravel())
    print(model)
    pre=model.predict(X_test)
    print('\n')
    AS=accuracy_score(y_test,pre)
    print('accuracy score',AS)
    score.append(AS*100)
    print('\n')
    sc=cross_val_score(model,X_train,y_train,cv=5,scoring='accuracy').mean()
    print('Cross val score',sc)
    cvs.append(sc*100)
    print('\n')
    false_positive_rate,true_positive_rate,thresholds=roc_curve(y_test,pre)
    roc_auc=auc(false_positive_rate,true_positive_rate)
    print('roc_auc score',roc_auc)
    rocscore.append(roc_auc*100)
    print('\n')
    print('classification report\n',classification_report(y_test,pre))
    print('\n')
    cm=confusion_matrix(y_test,pre)
    print(cm)
    print('\n')
    plt.figure(figsize=(10,60))
    plt.subplot(911)
    plt.title(name)
    print(sns.heatmap(cm,annot=True))
    plt.subplot(912)
    plt.title(name)
    plt.plot( false_positive_rate,true_positive_rate,label='AUC = %0.2f'% roc_auc)
    plt.plot([0,1],[0,1],'r--')
    plt.legend(loc='lower right')
    plt.ylabel('true positive rate')
    plt.xlabel('false positive rate')
    print('\n\n')
```

Results: -

	Model	Accuracy_score	Cross val score	Roc_auc_curve
0	GNB	64.602937	66.540453	64.491983
1	LogisticRegression	76.982351	77.003953	76.979639
2	DTC	91.449698	91.130260	91.446680
3	kNeighborsClassifier	88.582310	87.749371	88.540067
4	RFC	95.348458	95.157245	95.348439
5	GBC	90.517612	90.485275	90.512898
6	ADACLASS1	92.078005	92.219063	92.072692
7	ADACLASS2	76.798601	76.828352	76.797667

- Key Metrics for success in solving problem under consideration

The key metrics used were as follows: -

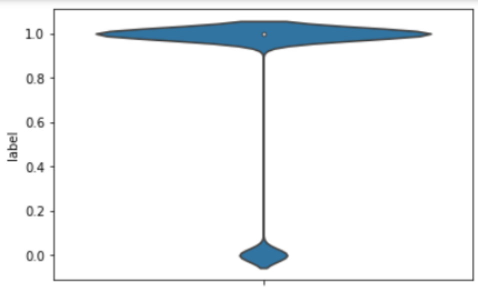
- Accuracy score: gives us the score between training and testing
- crossvalidation score: validates that the model is not overfitting

- Roc_auc_curve: a very useful tool in case of binary classification in imbalanced dataset
- Classification report: gives us useful values like recall & f1score
- Confusion matrix: gives us a clear picture about true positives & true negatives

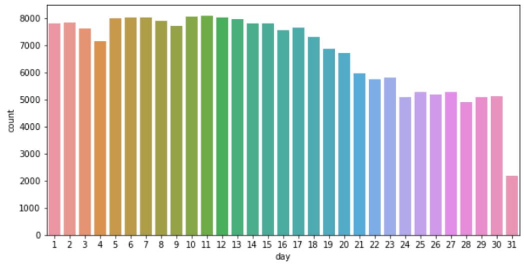
- Visualizations

Following were the plots made: -

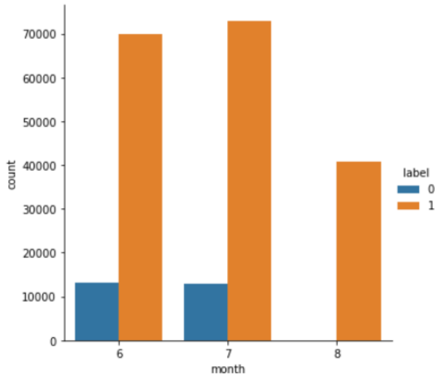
- Violinplot: used to check probability density (or distribution) of data for various variable

e.g. 

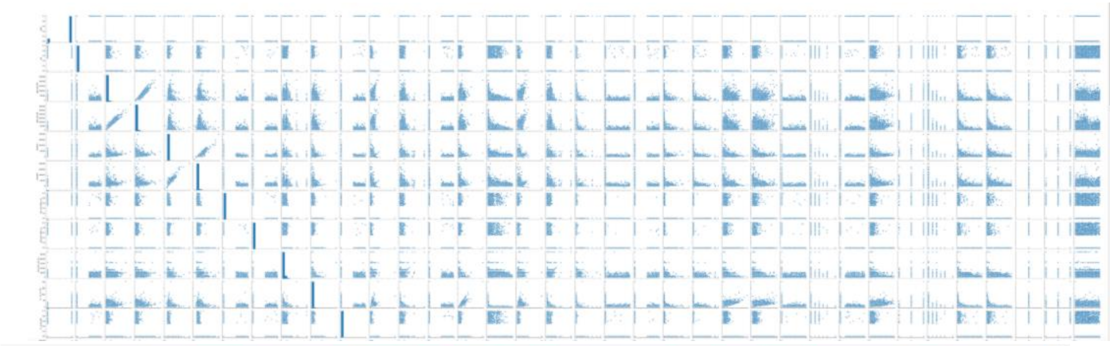
- Countplot: used to check frequency of data points for variables with discrete values

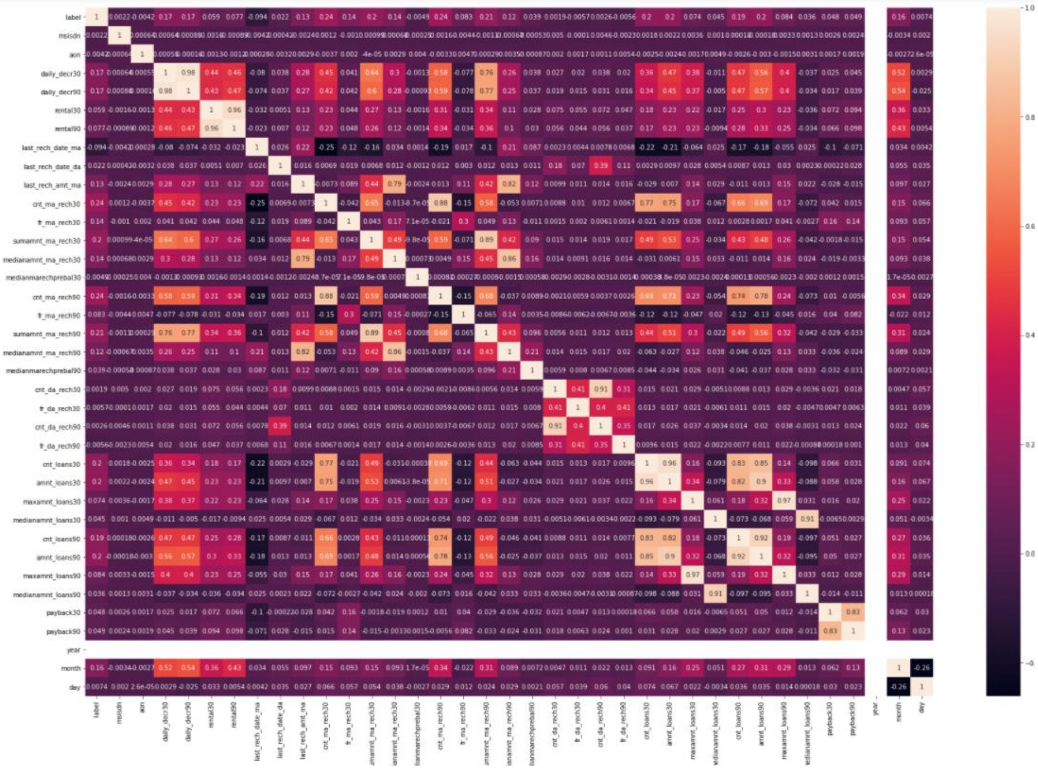
e.g. 

- Catplot: used to check frequency of data points split with defined legend

e.g. 

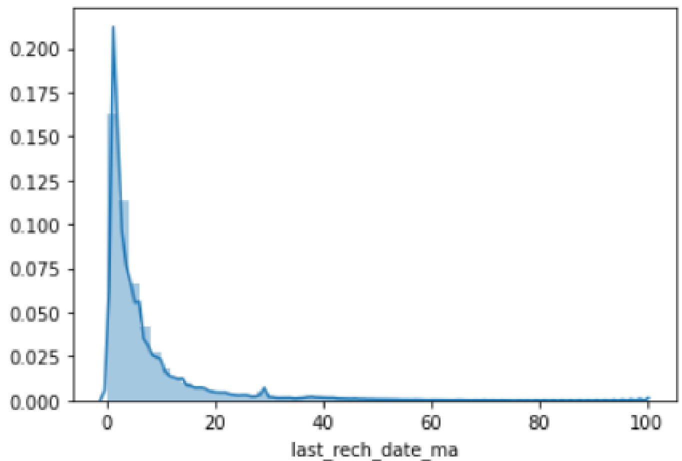
- Pairplot & correlation matrix: highlights correlation between different variables





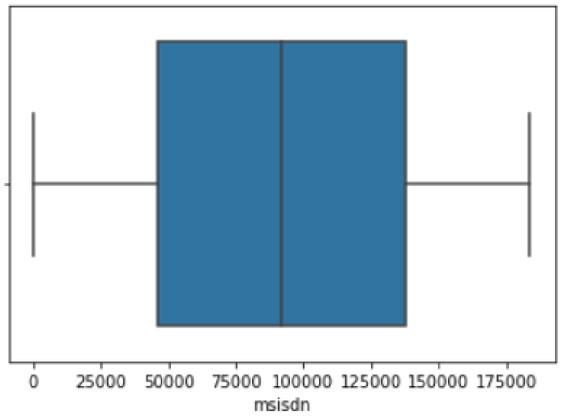
e.g.

- Distplot: gives distribution of data for various features



e.g.

- Boxplot: helps in identifying outliers

e.g. 

- ## Interpretation of the Results

Results from visualizations were as follows:-

-Violinplot was used to get an idea regarding the distribution of the data. The results from it was

```
FINDINGS(The max distribution of data is as follows)-->
-label=1(ie non defaulter)
-aon< 0.2*10^6
-daily_decr30<25000
-daily_decr90<25000
-rental30<25000
-rental90<25000
-last_rech_date_ma<0.2*10^6
-last_rech_date_da<0.2*10^6
-last_rech_amt_ma<5000
-cnt_ma_rech30<25
-fr_ma_rech30<0.2*10^6
-sumamnt_ma_rech30<50000
-medianamnt_ma_rech30<5000
-medianmarechprebal30<0.2*10^6
-cnt_ma_rech90<25
-fr_ma_rech90<10
-sumamnt_ma_rech90<0.2*10^6
-medianamnt_ma_rech90<5000
-medianmarechprebal90~0
-cnt_da_rech30~0
-fr_da_rech30<0.2*10^6
-cnt_da_rech90~0
-fr_da_rech90~0
-cnt_loans30<10
-amnt_loans30<50
-maxamnt_loans30~0
-medianamnt_loans30~0
-cnt_loans90~0
-amnt_loans90<100
-maxamnt_loans90~6
-medianamnt_loans90~0
-payback30<12.5
-payback90<12.5
-year=2016
-month=7
-day~10
```

Then, we used the countplot to get the frequency of discrete values. The results were

```
FINDINGS(The max count of the above parameters is as follows)-->
-year=2016
-month=7
-day=11
-label=1
```

Using the catplot with hue as ‘label’, we obtained

```
FINDINGS(The max count of the above parameters wrt label is as follows)-->
-Label=1 shows highest frequency in relation to year,month and day without any exception
```

For correlation between different features, we used pairplot and correlation matrix

```
High positive correlation exists between--
daily_decr30 v/s daily_decr90
rental30 v/s rental90
last_rech_amt_ma v/s medianamnt_ma_rech90
cnt_ma_rech30 v/s cnt_ma_rech90
sumamnt_ma_rech30 v/s sumamnt_ma_rech90
medianamnt_ma_rech30 v/s medianamnt_ma_rech90
cnt_da_rech30 v/s cnt_da_rech90
cnt_loans30 v/s amnt_loans30
cnt_loans30 v/s cnt_loans90
cnt_loans30 v/s amnt_loans90
amnt_loans30 v/s cnt_loans90
amnt_loans30 v/s amnt_loans90
maxamnt_loans30 v/s maxamnt_loans90
cnt_loans90 v/s amnt_loans90
payback30 v/s payback90

High negative correlation exists between
last_rech_date_ma v/s cnt_ma_rech30
last_rech_date_ma v/s cnt_loans30
month v/s day
```

Resuts from data preprocessing were as follows

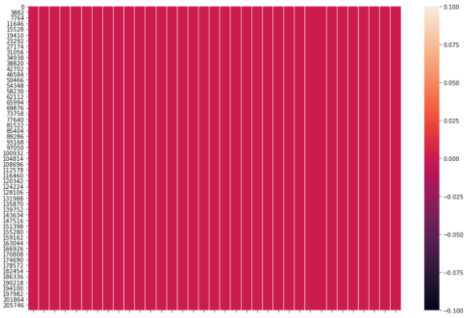
-Getting the unique value for each feature

label	2
msisdn	186243
aon	4507
daily_decr30	147026
daily_decr90	158670
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	

It shows that entire dataset has only singular pcircle, thus its of no use in modelling

-Similarly, the column of ‘unnamed’ too could be dropped.

-The dataset was complete with 0 missing value



-Also, the major chunk of data was of numeric dtype

```
label                      int64
msisdn                    object
aon                      float64
daily_decr30             float64
daily_decr90             float64
rental30                 float64
rental90                 float64
last_rech_date_ma        float64
last_rech_date_da        float64
last_rech_amt_ma           int64
cnt_ma_rech30              int64
fr_ma_rech30             float64
sumamnt_ma_rech30        float64
medianamnt_ma_rech30     float64
medianmarechprebal30     float64
cnt_ma_rech90              int64
fr_ma_rech90               int64
sumamnt_ma_rech90          int64
medianamnt_ma_rech90     float64
medianmarechprebal90     float64
cnt_da_rech30            float64
fr_da_rech30             float64
cnt_da_rech90              int64
fr_da_rech90               int64
cnt_loans30                int64
amnt_loans30               int64
maxamnt_loans30          float64
medianamnt_loans30       float64
cnt_loans90              float64
amnt_loans90               int64
maxamnt_loans90            int64
medianamnt_loans90       float64
payback30                float64
payback90                float64
year                       int64
month                      int64
day                        int64
```

-Some unrealistic values were noticed in the following features

- last_reach_date_ma
- last_rech_date_da
- cnt_ma_rech30
- fr_ma_rech30
- cnt_ma_rech90
- fr_ma_rech90
- cnt_da_rech30

- fr_da_rech30
- cnt_da_rech90
- fr_da_rech90
- cnt_loans30
- maxamnt_loans30
- cnt_loans90

Therefore, such values had to be treated with median statistic. Median was chosen due to its continuous character and the need of absolute values.

-Statistical summary gave us the following conclusions

```
Findings--
-STD DEVIATION is relatively higher in case of---
msisdn,aon,daily_decr30,daily_decr90,rental30,rental90,last_rech_amt_ma,sumamnt_ma_rech30,medianamnt_ma_rech30,medianmarechpreba
l30,sumamnt_ma_rech90,medianamnt_ma_rech90,medianmarechprebal90
(Thus, largely distributed data might be there and hav to be looked into)
-MEAN<MEDIAN in case of--label(ie the case of left skewness)
-Gap b/w max and 75th percentile is relatively higher in case of---
aon,daily_decr30,daily_decr90,rental30,rental90,last_rech_date_ma,last_rech_date_da,last_rech_amt_ma,cnt_ma_rech30,fr_ma_rech30,
sumamnt_ma_rech30,medianamnt_ma_rech30,medianmarechprebal30,cnt_ma_rech90,fr_ma_rech90,sumamnt_ma_rech90,medianamnt_ma_rech90,me
dianmarechprebal90,cnt_da_rech30,fr_da_rech30,cnt_da_rech90,fr_da_rech90,cnt_loans30,amnt_loans30
(ie chances of outliers being there as high)
```

-The dataset was having clear outliers which had to be dealt with using zscore

```
z_score=abs(zscore(df))
print(df.shape)
dffinal=df.loc[(z_score<5).all(axis=1)]
print(dffinal.shape)
```

-Also, the dataset had some skewness

```
label                      -2.278195
msisdn                     -0.000996
aon                         0.953842
daily_decr30                2.328861
rental30                    2.482312
last_rech_date_ma           2.733742
last_rech_date_da          12.015965
last_rech_amt_ma            2.217159
cnt_ma_rech30               1.714459
fr_ma_rech30                1.804465
sumamnt_ma_rech30           2.084924
medianamnt_ma_rech30        2.431959
medianmarechprebal30       10.674438
fr_ma_rech90                2.163293
medianmarechprebal90        4.849542
cnt_da_rech30               9.884992
fr_da_rech30              438.910014
fr_da_rech90               80.890174
maxamnt_loans30             1.479818
medianamnt_loans30          4.099560
amnt_loans90                2.188093
medianamnt_loans90          4.488738
payback30                   3.666016
month                       0.360607
day                         0.203293
dtype: float64
```

Modelling results were as follows:-

	Model	Accuracy_score	Cross val score	Roc_auc_curve
0	GNB	64.602937	66.540453	64.491983
1	LogisticRegression	76.982351	77.003953	76.979639
2	DTC	91.449698	91.130260	91.446680
3	kNeighborsClassifier	88.582310	87.749371	88.540067
4	RFC	95.348458	95.157245	95.348439
5	GBC	90.517612	90.485275	90.512898
6	ADACLASS1	92.078005	92.219063	92.072692
7	ADACLASS2	76.798601	76.828352	76.797667

```
parameters={'n_estimators':[50,100,150,200,250,300]}
clf=RandomizedSearchCV(rfc,parameters,cv=5)
clf.fit(X_train,y_train)
print('best RFC parameters is :',clf.best_params_)
print('\n')
```

```
best RFC parameters is : {'n_estimators': 250}
```

```
rfc=RandomForestClassifier(n_estimators=250)
r_state=maxacc_score(rfc,X_train_res,y_train_res)
print('\n')
```

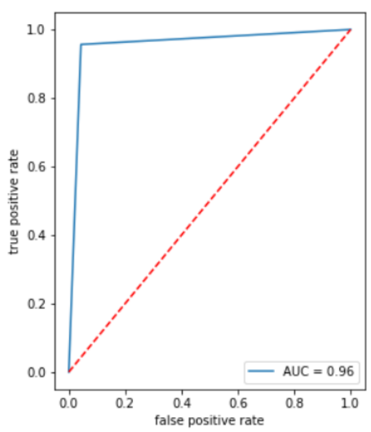
```
max acc score corresponding to 78 is 0.9568928470874146
```

Final model showed

```
accuracy score of RFC is 0.9566557503371219
[[32316  1455]
 [ 1470 32242]]
              precision    recall  f1-score   support

           0       0.96      0.96      0.96     33771
           1       0.96      0.96      0.96     33712

    accuracy                           0.96     67483
   macro avg       0.96      0.96      0.96     67483
weighted avg       0.96      0.96      0.96     67483
```



CONCLUSION

- ## Key Findings and Conclusions of the Study

To summarize, the major observations were
-Some features that were not needed at all
-Presence of unrealistic values
-Object dtype that had to be converted to numeric one for usage
-Dropping some columns having high positive collinearity
-Presence of skewness and outliers keeping in mind the expensive data
-Using suitable algorithms and metrics

- ## Learning Outcomes of the Study in respect of Data Science

Visualizations using matplotlib and seaborn is a great asset that is used to gather insights from a huge dataset. It helps us get the count, relative relations, distribution of data, normal v/s skew, outliers,etc depending upon the demands of problem.

On the other hand, data cleaning is of paramount importance before the data is fed into model training. More our data is cleaned and valid, more accurate we get our results.

For the algorithms part, each algorithm has its own advantages and use. Still, for apt results, all major algorithms were used fro the projects and the best performing one was picked up for our final model training.

- ## Limitations of this work and Scope for Future Work

-The major hurdle in the problem was the presence of quite a lot of unrealistic values in some columns. Thus, we had to assume a threshold above which the probability of getting such values was unreal, followed by treating them with median statistics library. Along with it, the dataset was largely an imbalanced one.

It would have been better if the dataset was devoid of misleading values and the results therefore could have been more powerful.