FLIP ROBO

Micro-Credit Defaulter Model

Submitted by:

kamalakanta sahu

ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped you and guided you in completion of the project.

INTRODUCTION

- Business Problem Framing

  Predict whether a customer will be paying back the loaned amount within 5 days of insurance of loan. If finance company come to know like which customer can return the loan amount then company will give the loan only to those people inspite of all people applied for loan .Hence less risk and high profit .

- Conceptual Background of the Domain Problem

  We need to find out whether a person is a defaulter or payer from the given data.For this first we need to check whether a person paid the loan amount in last 5 days or not .

- Review of Literature

  From the problem statement it is clear that this is a classification problem.There are many classification algorithm available to solve such type of problem.In this research we have used four algorithm naming random forest,decissiontree,kneighbor,logestic regression.We have gone through data preprocessing,EDA,outlier detection to make data proper .

- Motivation for the Problem Undertaken

  Now a days there are lots of micro finance companies which provide small loans to its customer and this research will be helpfull to solve various problem in same domain .

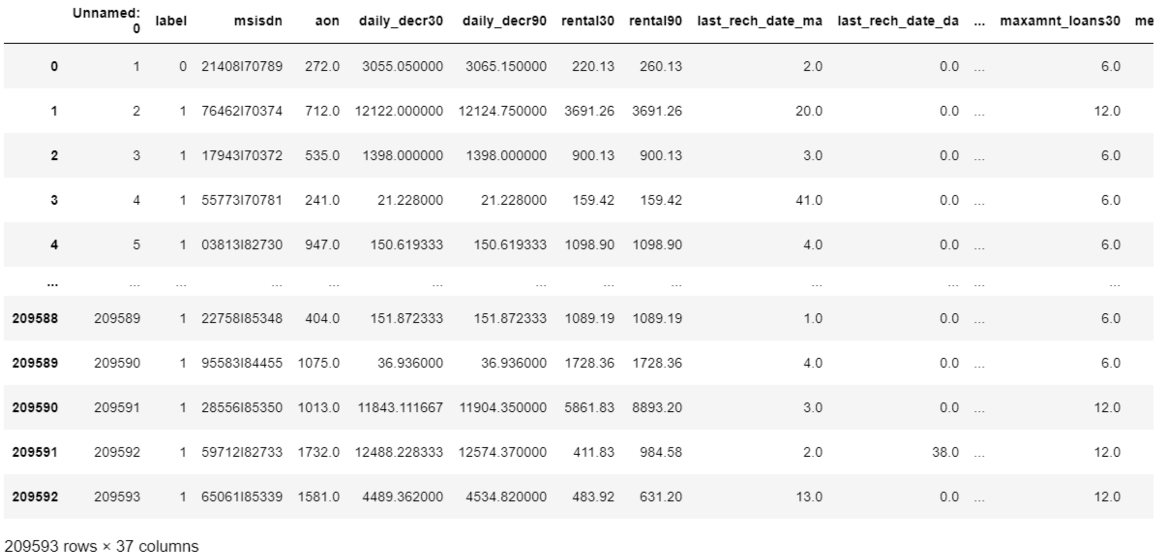
Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  In this project we have used mean,standard deviation,min,max,zscore,correlation like mathemetical/statistical methods to find variance,outleir and relations between features .

- Data Sources and their formats

We have processed csv file shared by fliprobo .We have loaded this file into our panda dataframe to do further manipulation .It contains almost 37 columns and 209593 records .From these 37 columns we have one target column ie. "label" and rest 36 columns are feature columns.



	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	...	maxamnt_loans30	me
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	...	6.0	
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	...	12.0	
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	...	6.0	
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	...	6.0	
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	...	6.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
209588	209589	1	22758I85348	404.0	151.872333	151.872333	1089.19	1089.19	1.0	0.0	...	6.0	
209589	209590	1	95583I84455	1075.0	36.936000	36.936000	1728.36	1728.36	4.0	0.0	...	6.0	
209590	209591	1	28556I85350	1013.0	11843.111667	11904.350000	5861.83	8893.20	3.0	0.0	...	12.0	
209591	209592	1	59712I82733	1732.0	12488.228333	12574.370000	411.83	984.58	2.0	38.0	...	12.0	
209592	209593	1	65061I85339	1581.0	4489.362000	4534.820000	483.92	631.20	13.0	0.0	...	12.0	

209593 rows × 37 columns

- Data Preprocessing Done

Before creating any model it is necessary to make our data proper .If we feed invalid data then our model will not give good prediction.So here if you see the dataset you can find lots of fictional data present in our dataset .Column 1 is just an identifier .So we can drop this column .Also we dropped pdate column as it is not relevant to our test case .In our dataset there is no NaN value present in data .So no need to treat NaN value in such case .Msisdn column also an unique presentor of a customer .We can remove this too.Circle coulmn contains same value across all the row . This will not add any value to our model .So this column can be dropped.Aon column contains no of days .In real world no of days can not be in negative value.So we dropped those records.maxamnt_loans90 and maxamnt_loans30 column contains various type of values.But as per document a user can take only two types of loan.One is of loan amount Rs 5 and other one is of loan amount Rs 10.So max loan amount can be of 0(if no loan),6 or 12(with interest).But if you see the data then you can find data with huge number .We can filter out those data.There are few columns which contains 0 values more than 90% .In such case we can drop those columns instead of imputing mean value .

columns'daily_decr90','cnt_loans90','maxamnt_loans90','cnt_loans30' ,'rental90' are highly corelated with other feature.So we can drop these columns.Then we have removed skewed data and outliers .After all these process we have standardise the input data as it contains various measures .

- Data Inputs- Logic- Output Relationships

We have taken first 20 record of dataset where label =1 and in descending order by amnt_loans30 column to identify the pattern in dataset. Here we found below obsererervations .

1.here people with level1 returning loan amount in time that is within 2 days .

2.Also loan count is more compare to lavel0

3.People are recharging their main account in every one or two days.

4.Label column is highly dependant on payback day .

After analysing correlation value among feature we found that columns'daily_decr90','cnt_loans90','maxamnt_loans90','cnt_loans30' ,'rental90' are highly corelated with other features .So we have removed these columns .

- State the set of assumptions (if any) related to the problem under consideration

From dataset we assume that target variable is highly corelated with payback days,loan amount and daily main balance spend .

- Hardware and Software Requirements and Tools Used

Listing down the hardware and software requirements along with the tools, libraries and packages used. Describe all the software tools used along with a detailed description of tasks done with those tools.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

From problem statement we can understand that it is a classification problem.So here we have applied few classification model to find best prediction ratio.

- Testing of Identified Approaches (Algorithms)

We have used four classification algorithm for our project which are listed below .

1.LogisticRegression

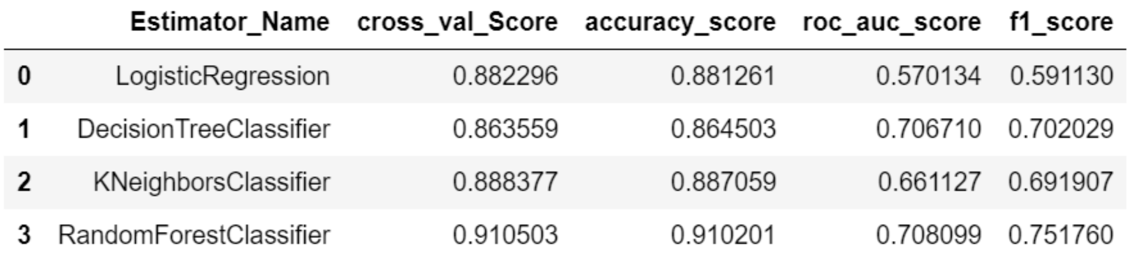
2.DecisionTreeClassifier

3.KNeighborsClassifier

4.RandomForestClassifier

- Run and Evaluate selected models

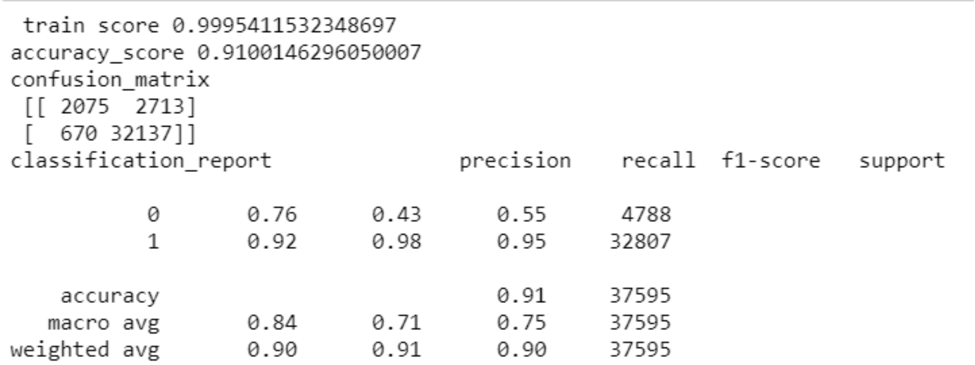
This dataset is an inbalanced dataset .So for such type of dataset tree type algorithm works better compare to others .And not surprisingly decisiontree and randomforest model metrics better than other two .



	Estimator_Name	cross_val_Score	accuracy_score	roc_auc_score	f1_score
0	LogisticRegression	0.882296	0.881261	0.570134	0.591130
1	DecisionTreeClassifier	0.863559	0.864503	0.706710	0.702029
2	KNeighborsClassifier	0.888377	0.887059	0.661127	0.691907
3	RandomForestClassifier	0.910503	0.910201	0.708099	0.751760

- Key Metrics for success in solving problem under consideration

If we see above snapshot RandomForest has average f1-score and roc score compare to other model .So considering this model as final model for my problem .

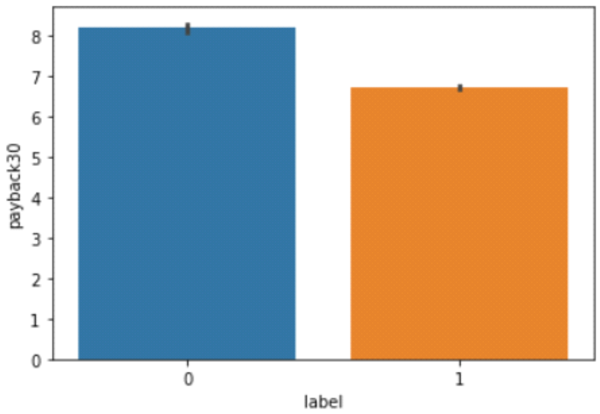


```
 train score 0.9995411532348697
accuracy_score 0.9100146296050007
confusion_matrix
 [[ 2075  2713]
 [  670 32137]]
classification_report               precision    recall  f1-score   support

           0       0.76      0.43      0.55      4788
           1       0.92      0.98      0.95     32807

    accuracy                           0.91     37595
   macro avg       0.84      0.71      0.75     37595
weighted avg       0.90      0.91      0.90     37595
```

- Visualizations

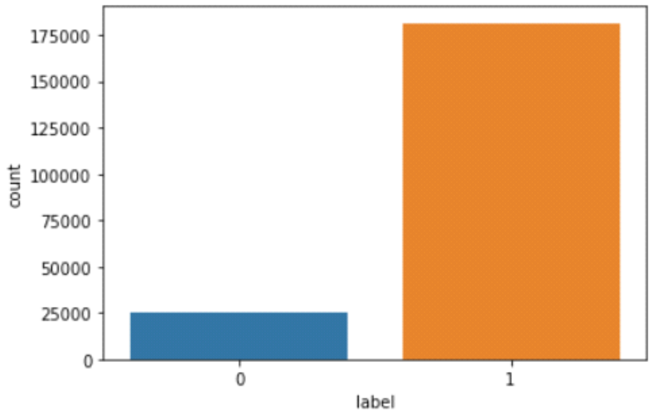
```
sns.barplot(data=df_30,x='label',y='payback30')
#Lavel1 user payback rate is good compare to lavel0 user
```

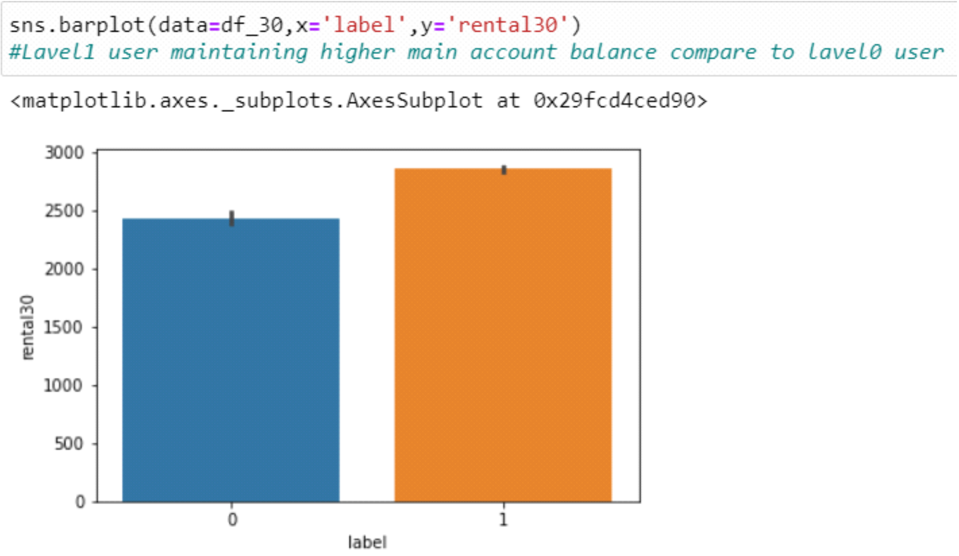
```
<matplotlib.axes._subplots.AxesSubplot at 0x29fc6162b50>
```



```
sns.countplot(data=df_30,x='label')
#Label-1 has more count compare to Label-0
#Means defaulter count is less .
```

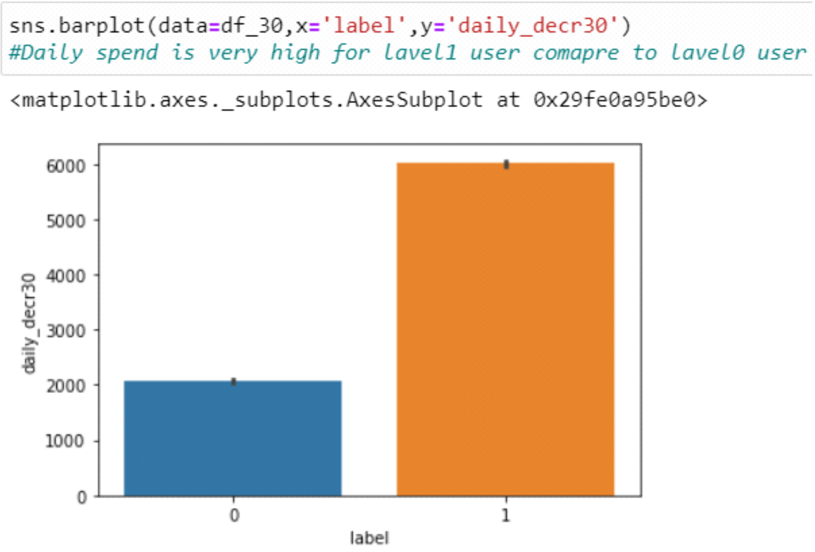
```
<matplotlib.axes._subplots.AxesSubplot at 0x29fc61a5d60>
```





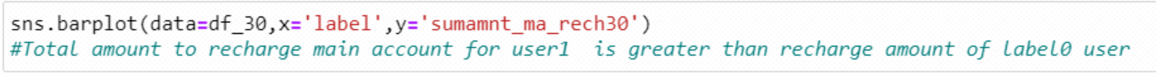
```
sns.barplot(data=df_30,x='label',y='rental30')
#Lavel1 user maintaining higher main account balance compare to lavel0 user
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x29fcd4ced90>
```



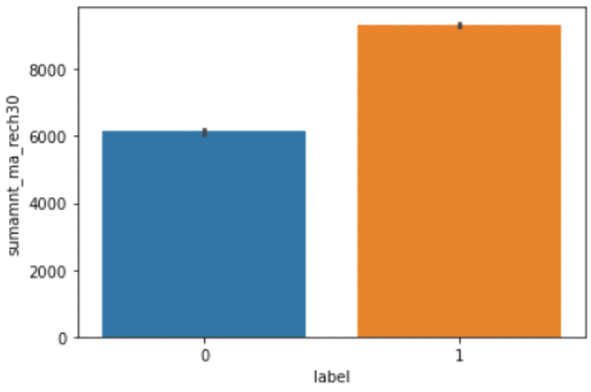
```
sns.barplot(data=df_30,x='label',y='daily_decr30')
#Daily spend is very high for lavel1 user comapre to lavel0 user
```

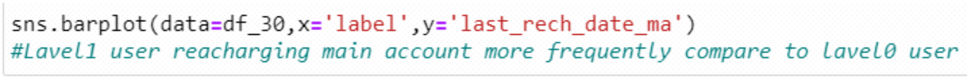
```
<matplotlib.axes._subplots.AxesSubplot at 0x29fe0a95be0>
```



```
sns.barplot(data=df_30,x='label',y='sumamnt_ma_rech30')
#Total amount to recharge main account for user1  is greater than recharge amount of label0 user
```

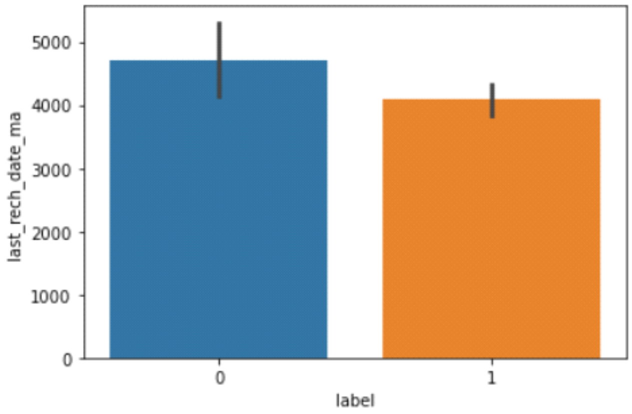
```
<matplotlib.axes._subplots.AxesSubplot at 0x29fdfdbeb50>
```

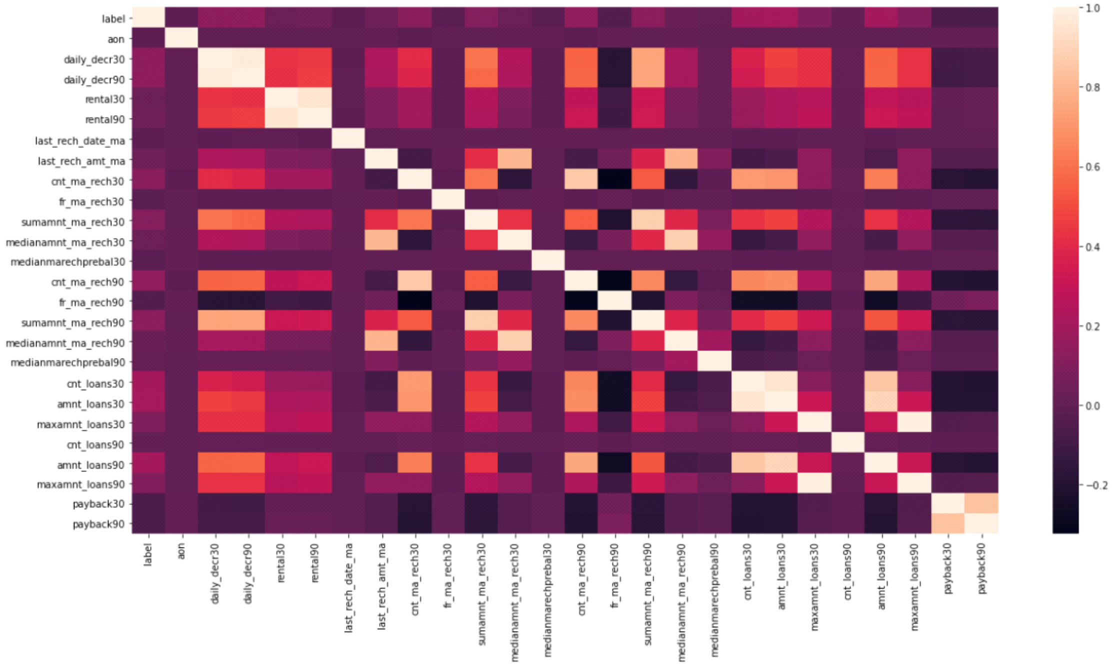




```
sns.barplot(data=df_30,x='label',y='last_rech_date_ma')
#Lavel1 user reacharging main account more frequently compare to lavel0 user
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x29fe0a95520>
```





From above image we found below observation

#daily_decr_30 is highly corelated with daily_decr_90.We can drop any one of the column .

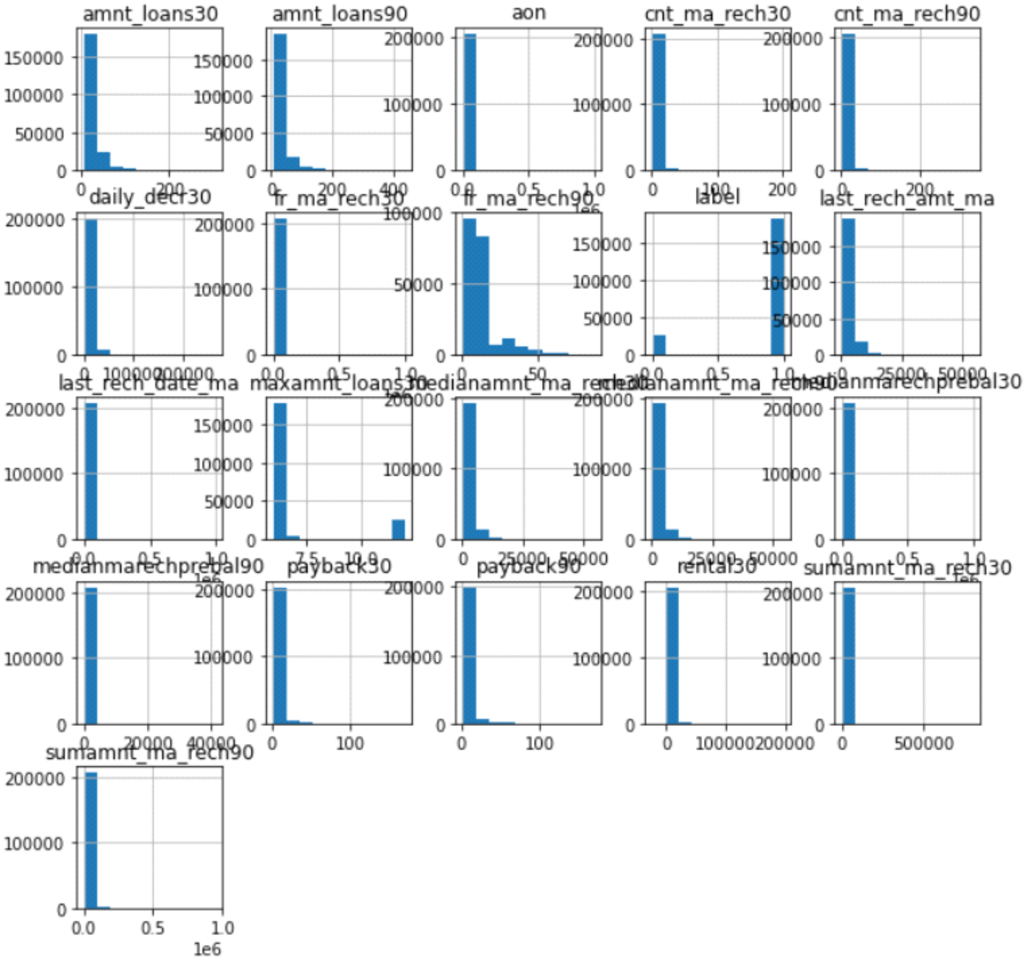
#cnt_loans30 is highly corelated with cnt_loans90

#maxamnt_loans30 is highly corelated with maxamnt_loans90

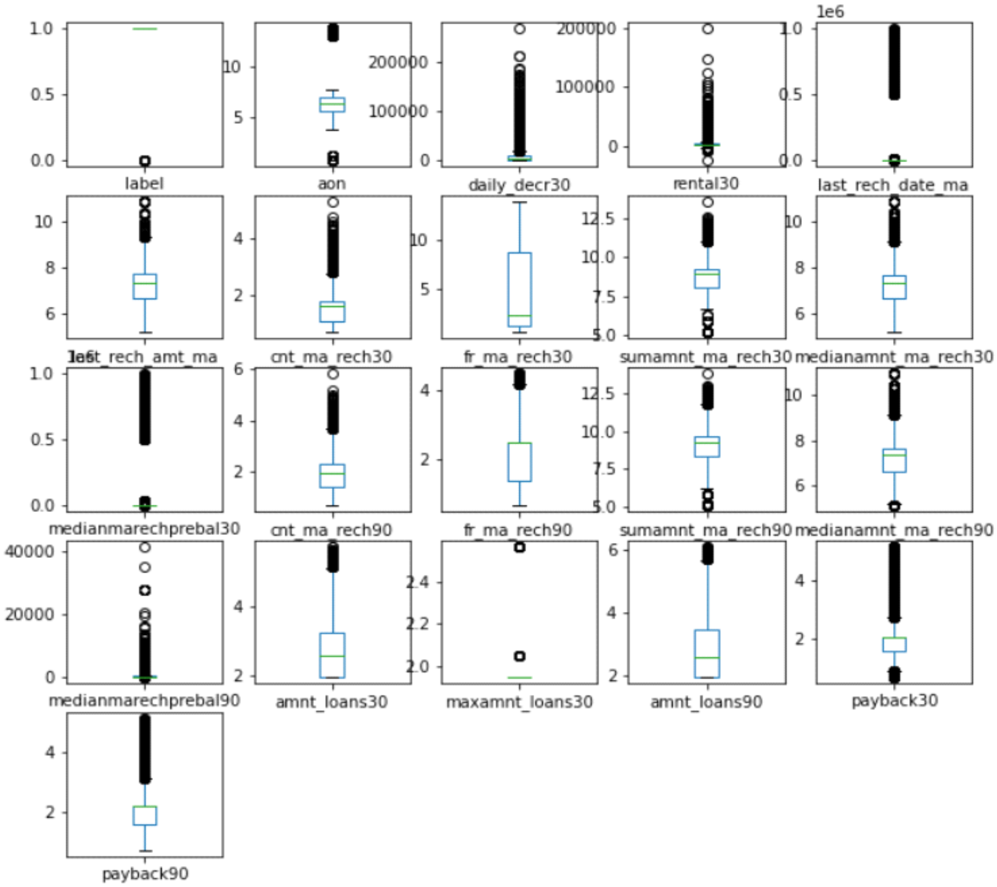
#amnt_loans30 and amnt_loans90

#amnt_loans30 and cnt_loan30

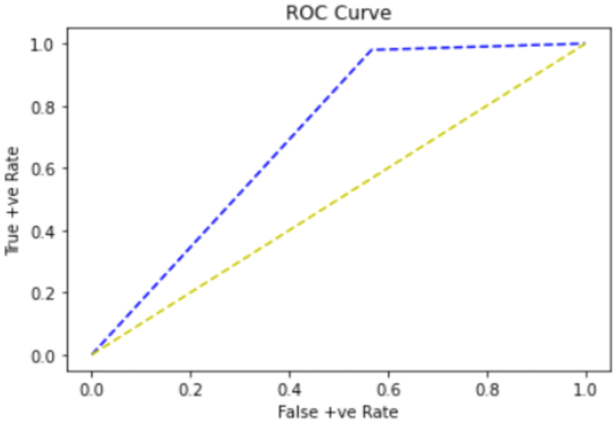
#rental30 and rental90



Most of the data are right skewed .



Most of the data contains outlier.



- Interpretation of the Results

By using random forest with hyper parameter we got an accuracy of 91% ,f1-score of 75% and roc of 70% .

CONCLUSION

- Key Findings and Conclusions of the Study

  For me it took time for data cleaning and finding insights of data.Now what I understood is data preprocessing is the main job in ml .

- Learning Outcomes of the Study in respect of Data Science

  Learned many things with respect to machine learing .

  1.Project usecase understanding

  2.Data cleaning

  3.Bar plot to find the insight

  4.Heatmap for corelation

  5.Histogram for finding variance

  6.Boxplot for finding of outliers

- Limitations of this work and Scope for Future Work

  Only I can say that this dataset is an inbalnce dataset .So might be it is good if we balance it by adding under balanced catergorial data or decreasing over balanced data .