Micro-Credit Defaulter Model

Submitted by:

Arjita Saxena

ACKNOWLEDGMENT

I would like to express my sincere thanks to Ms. Swati Mahaseth for her timely and valuable support and guidance in completing my project.

I would also like to express my gratitude towards my family & members of Flip Robo Technologies for their kind co-operation and encouragement which helped me in completion of this project. I would like to express my special gratitude and thanks to industry persons for giving me this great opportunity to do a project on ‘Micro-Credit Defaulter Model’.

INTRODUCTION

• Business Problem Framing

The problem is focused on microfinance institutions (MFI) providing their services and products to low income families and poor customers that can help them in the need of hour to provide micro-credit (insurance of loan) on mobile balances to be paid back in 5 days. The Consumer is believed to be defaulter if he deviates from the path of paying back the loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah), while, for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah).

We are building 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.

• Conceptual Background of the Domain Problem

Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients.

There is one such client that is in Telecom Industry. They are a fixed wireless telecommunications network provider. They have launched various products and have developed its business and organization based on the budget operator model, offering better products at Lower Prices to all value conscious customers through a strategy of disruptive innovation that focuses on the subscriber.

They understand the importance of communication and how it affects a person's life, thus, focusing on providing their services and products to low income families and poor customers that can help them in the need of hour. They are collaborating with an MFI to provide micro-credit on mobile balances to be paid back in 5 days.

• Review of Literature

The Consumer is believed to be defaulter if he deviates from the path of paying back the loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah), while, for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah).

- Motivation for the Problem Undertaken

The project is basically focused on providing their services and products to low income families and poor customers that can help them in the need of hour.

In order to improve the selection of customers for the credit, the client wants some predictions that could help them in further investment and improvement in selection of customers.

The target variable in the data set is 'Label' that will be represented by binary classes. Label ‘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. Hence it is a Binary classification problem.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

In the project below are the mathematical, statistical and analytics modelling done:

- Feature engineering:
  Feature engineering is the process of using domain knowledge to reconfigure data and create “features” that optimize machine learning algorithms.

- Data Pre-processing :
  To ensure high-quality data, it's crucial to pre-process it. To make the process easier, data pre-processing is divided into four stages: data cleaning, data integration, data reduction, and data transformation.

- Exploratory Data Analysis:
  Exploratory Data Analysis (EDA) is the crucial process of using summary statistics and graphical representations to perform preliminary investigations on data in order to uncover patterns, detect anomalies, test hypotheses, and verify assumptions.

- Data cleaning :
  Data cleaning is the process of ensuring data is correct, consistent and usable. You can clean data by identifying errors or corruptions, correcting or deleting them, or manually processing data as needed to prevent the same errors from occurring.

- Correlation :
  Correlation is a statistical measure. Correlation explains how one or more variables are related to each other. These variables can be input data features which have been used to forecast our target variable.

- Data Summary :
  The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values).

- Statistic summary :
  It is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values. When this method is applied to a series

of string, it returns a different output like count of values, unique values, top and frequency of occurrence in this case.

- Data Sources and their formats

The sample data provided is from the client database, in order to improve the selection of customers for the credit, the client wants some predictions that could help them in further investment and improvement in selection of customers. Below are the formats :

```
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 target variable in the data set is 'Label' that will be represented by binary classes. Label ‘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. Hence it is a Binary classification problem.

- Data Pre-processing Done
  - Changing columns data type to the relevant one
  - Dropping irrelevant columns that are not contributing much in the model learning
  - Removing invalid / null values

- Checking and removing duplicates present in the dataset
- Treating outliers and skewness
- Checked and treated Multi-colinearity
- Scaling

- Data Inputs- Logic- Output Relationships

  Columns 'daily_decr30', 'daily_decr90', 'last_rech_amt_ma', 'cnt_ma_rech30', 'sumamnt_ma_rech30', 'medianamnt_ma_rech30', 'cnt_ma_rech30', 'cnt_ma_rech90' have good correlation with target column 'label'.

  'rental30' has strong positive correlation with 'daily_decr30', 'daily_decr90'. 'rental90' has strong positive correlation with 'daily_decr30', 'daily_decr90'. 'last_rech_amt_ma' has good correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90'. 'cnt_ma_rech30' has good correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90', 'last_rech_amt_ma'. 'sumamnt_ma_rech30' has strong correlation with 'daily_decr30', 'daily_decr90', 'last_rech_amt_ma', 'cnt_ma_rech30'. 'medianamnt_ma_rech30' has very strong postive correlation with 'last_rech_amt_ma' and 'sumamnt_ma_rech30'. 'cnt_ma_rech90' has good/strong positive correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90', 'last_rech_amt_ma', 'cnt_ma_rech30', 'sumamnt_ma_rech30', 'medianamnt_ma_rech30'. 'sumamnt_ma_rech90' has good/strong positive correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90', 'last_rech_amt_ma', 'cnt_ma_rech30', 'sumamnt_ma_rech30', 'medianamnt_ma_rech30', 'cnt_ma_rech90'. 'medianamnt_ma_rech90' has good/strong positive correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90', 'last_rech_amt_ma', 'cnt_ma_rech30', 'sumamnt_ma_rech30', 'medianamnt_ma_rech30', 'cnt_ma_rech90','sumamnt_ma_rech90'. 'fr_da_rech30' has strong correlation with 'fr_da_rech90'. 'pdate_month' has strong correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90' and good coorlation with few other columns. Many independent columns have strong linear correlation with each other.

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

- There are no null values in the dataset.
- There may be some customers with no loan history.
- The dataset is imbalanced. Label ‘1’ has approximately 87.5% records, while, label ‘0’ has approximately 12.5% records, hence class imbalance issue exists that is needed to be treated.
- For some features, there may be values which might not be realistic. We have observed them and treat them with a suitable explanation.

- We came across outliers in some features which we handled as per our understanding and relevant threshold.

• Hardware and Software Requirements and Tools Used

Hardware & Software requirements :

- Modern Operating System (**Windows:** 7 or newer)
- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM.
- 5 GB free disk space.
- Processor: Minimum 1 GHz; Recommended 2GHz or more
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above
- Sound card w/speakers
- Some classes require a camera and microphone

Tools, libraries and packages used :

- Tools : Jupyter Notebook (Anaconda)
- Libraries : Pandas, Numpy, Matplotlib, Seaborn, SciPy, Scikit-learn
- Statsmodels, imblearn, pickle, joblib.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
  - Problem Formulation and understanding
  - Data Preparation and Pre-processing
  - Feature Engineering and Selection
  - Exploratory Data Analysis
  - Data Cleaning
  - Model Development
  - Hyper parameter Tuning and Cross-Validation
  - Model Evaluation
  - Conclusion

- Testing of Identified Approaches (Algorithms)
  - Logistic Regression
  - Decision Tree Classifier
  - Random Forest Classifier
  - Gradient Boosting Classifier

- Run and Evaluate selected models
  - Logistic Regression

```
LogisticRegression()
Accuracy(Training) :  77.39878326455101  Accuracy(Test)  75.7473322613653  Mean absolute error :  0.24252667738634703
[[ 5057  1391]
 [11882 36398]]
              precision    recall  f1-score   support

           0       0.30      0.78      0.43      6448
           1       0.96      0.75      0.85     48280

    accuracy                           0.76     54728
   macro avg       0.63      0.77      0.64     54728
weighted avg       0.88      0.76      0.80     54728
```

  - Decision Tree Classifier

```
DecisionTreeClassifier()
Accuracy(Training) :  99.99911057350221  Accuracy(Test)  87.39584856015202  Mean absolute error :  0.12604151439847974
[[ 3867  2581]
 [ 4317 43963]]
              precision    recall  f1-score   support

           0       0.47      0.60      0.53      6448
           1       0.94      0.91      0.93     48280

    accuracy                           0.87     54728
   macro avg       0.71      0.76      0.73     54728
weighted avg       0.89      0.87      0.88     54728
```

  - Random Forest Classifier

```
RandomForestClassifier()
Accuracy(Training) :  99.99911057350221  Accuracy(Test)  91.51805291624031  Mean absolute error :  0.08481947083759685
[[ 3895  2553]
 [ 2089 46191]]
              precision    recall  f1-score   support

           0       0.65      0.60      0.63      6448
           1       0.95      0.96      0.95     48280

    accuracy                           0.92     54728
   macro avg       0.80      0.78      0.79     54728
weighted avg       0.91      0.92      0.91     54728
```

- Gradient Boosting Classifier

```
GradientBoostingClassifier()
Accuracy(Training) :  90.56096129215881  Accuracy(Test)  87.33737757637772  Mean absolute error :  0.12662622423622277
[[ 4801  1647]
 [ 5283 42997]]
              precision    recall  f1-score   support

           0       0.48      0.74      0.58      6448
           1       0.96      0.89      0.93     48280

    accuracy                           0.87     54728
   macro avg       0.72      0.82      0.75     54728
weighted avg       0.91      0.87      0.88     54728
```

RandomForestClassifier() has the highest accuracy among all the models selected for model training.

- Key Metrics for success in solving problem under consideration

Cross validation Score is used to select the best model

```
# Computing cross validation score of all the models used
from sklearn.model_selection import cross_val_score

for i in algo_list :
    print('CV mean of ',i,' is ',cross_val_score(i,x2,y,cv=5).mean())
```

```
CV mean of  LogisticRegression()  is  0.8832780594764973
CV mean of  DecisionTreeClassifier()  is  0.8855255584486775
CV mean of  RandomForestClassifier()  is  0.9214910237083733
CV mean of  GradientBoostingClassifier()  is  0.9203563108126627
```

```
# Finding out the difference of Accuracy and Cross validation mean of all the models used
   Accuracy CVmean Diff
LR   76       88    12
DTC  87       89    2
RFC  92       92    0
GBC  87       92    5
```

Looking at difference of accuracy and cv mean, opting for RandomForestClassifier() as our best model with accuracy of 92% and having least difference between accuracy and cross validation mean.

Hyper parameter Tuning is used for better accuracy and to avoid over fitting issues with best parameters on selected model.

```
# Using hyper parameter tuning on selected model for better accuracy and to avoid overfitting issues
from sklearn.model_selection import GridSearchCV

parameters = {'criterion':['gini', 'entropy'],
              'max_features':['auto', 'sqrt', 'log2'],
              'class_weight':['balanced','balanced_subsample',None],
              'min_samples_leaf':[1,2,3]}
             #'verbose':[0,1,2,3,4,5],
             #'n_estimators':[10,50,100]}

rf=RandomForestClassifier()
GCV=GridSearchCV(rf,parameters,cv=5)
GCV.fit(trainx,trainy)
GCV.best_params_
```

```
{'class_weight': None,
 'criterion': 'entropy',
 'max_features': 'log2',
 'min_samples_leaf': 3}
```

```
rf=RandomForestClassifier(class_weight=None,criterion='entropy',max_features='log2',min_samples_leaf=3)
rf.fit(trainx,trainy)
pred=rf.predict(x_test)
acc=accuracy_score(y_test,pred)
cv=cross_val_score(rf,x2,y,cv=5).mean()

print('Accuracy : ',acc,' CV mean : ',cv)
```

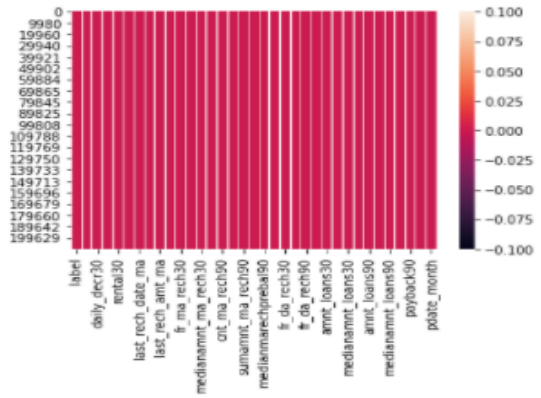
```
Accuracy :  0.9128416898114311  CV mean :  0.9221269014663559
```

We are getting model accuracy as 91% and cross validation mean as 0.92, this shows our model is performing good.

- Visualizations
  - **Checking for null values in the dataset**

```
# Visualizing nulls
sns.heatmap(df.isnull())
```

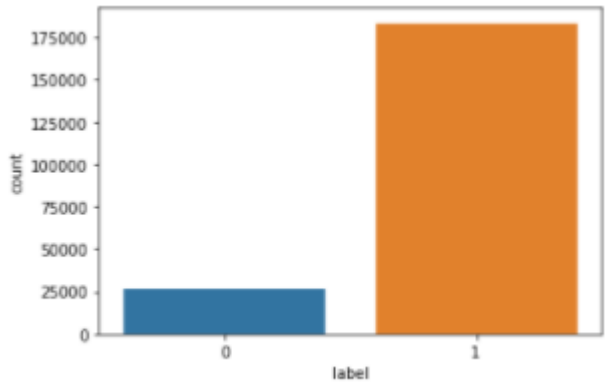
```
<AxesSubplot:>
```

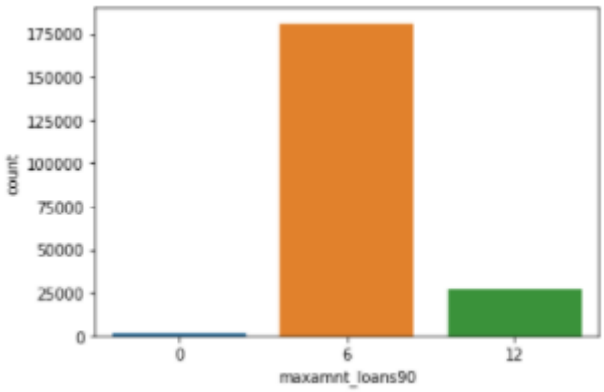


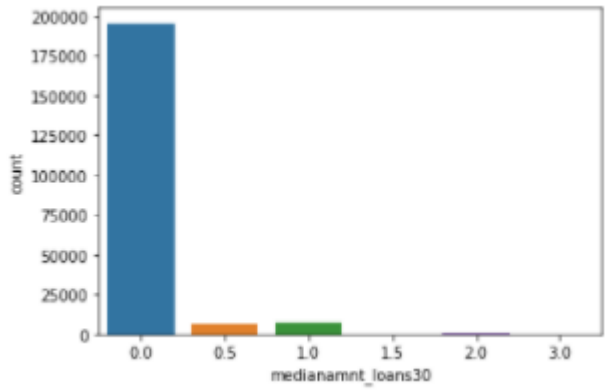
Heatmap showing no null values are present in the dataset.

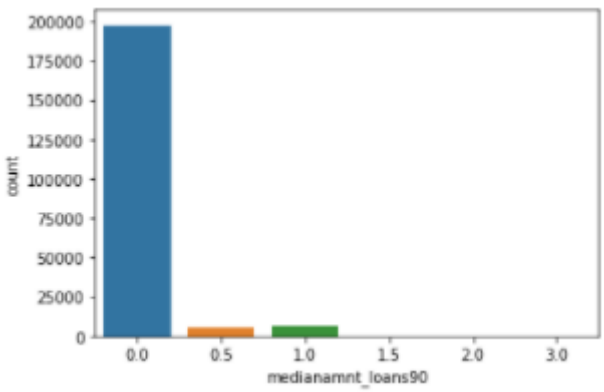
  - **Checking for the value counts using count plot**

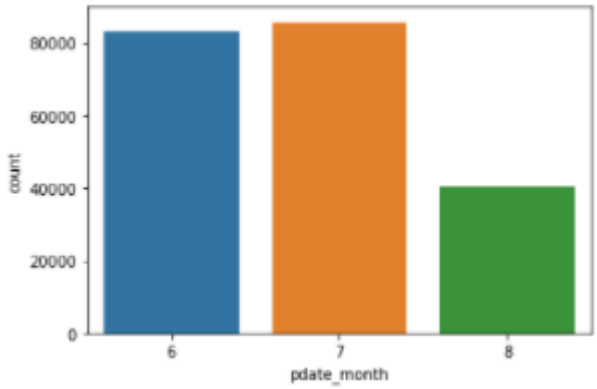
```
# Visualizing value counts for object type the columns
for i in df.columns :
    if df[i].nunique()<15 :
        #plt.figure(figsize=(15,4))
        sns.countplot(df[i])
        plt.show()
```











With the countplot we can see that 'medianamnt_loans30' column has 0.0 Median of amounts of loan taken by the user in last 30 days in majority and other values have very less count comparatively. Column 'maxamnt_loans90' has 6 as maximum amount of loan taken by the user in last 90 days n a very high count and other values comparatively have very less count. Column 'medianamnt_loans90' has 0.0 as Median of amounts of loan taken by the user in last 90 days in a very high count as compared to other values. There are 3 'pdate_months' June, July,August out of which July has highest count followed by June and least count is in August. This kind of distribution could result in skewness.

Column 'label' 1 (Non-defaulter) has a huge count compared to 0 (Defaulter) and since this is a target variable this columns needs class balancing.

- **Statistic Summary**

```
# Describe dataset
df.describe()
#df.describe(include='all')
```

	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cr
count	209562.000000	209562.000000	209562.000000	209562.000000	209562.000000	209562.000000	209562.000000	209562.000000	209562.000000	2
mean	0.875297	8113.512796	5382.170031	6083.386523	2692.964058	3483.905668	3756.403389	3712.752058	2064.754512	
std	0.330383	75701.620014	9221.086606	10919.382966	4308.784971	5770.737718	53909.859829	53378.762010	2370.831005	
min	0.000000	-48.000000	-93.012667	-93.012667	-23737.140000	-24720.580000	-29.000000	-29.000000	0.000000	
25%	1.000000	246.000000	42.480000	42.713250	280.800000	300.370000	1.000000	0.000000	770.000000	
50%	1.000000	527.000000	1470.466667	1500.000000	1083.940000	1334.400000	3.000000	0.000000	1539.000000	
75%	1.000000	982.000000	7246.000000	7804.000000	3357.462500	4202.537500	7.000000	0.000000	2309.000000	
max	1.000000	999860.755168	265926.000000	320630.000000	198926.110000	200148.110000	998650.377733	999171.809410	55000.000000	

8 rows × 35 columns

'label' has values 0 and 1, 'aon' ranges from -48 to 999860.755168, 'daily_decr30' ranges from -93.012667 to 265926, 'daily_decr90' ranges from -93.012667 to 320630, 'rental30' ranges from -23737.140000 to 198926.110000, 'rental90' ranges from -24720.580000 to 200148.110000, 'last_rech_date_ma' ranges from -29.000000 to 998650.377733, 'last_rech_date_da' -29.000000 to 999171.809410, 'last_rech_amt_ma' ranges from 0-55000, 'cnt_ma_rech30' ranges from 0-203, 'maxamnt_loans30' ranges from 0-99864.560864, 'medianamnt_loans30' ranges from 0 to 3, 'cnt_loans90' ranges from 0-4997.517944, 'amnt_loans90' ranges from 0-438, 'maxamnt_loans90' ranges from 0-12, 'medianamnt_loans90' ranges from 0-3, 'payback30' ranges from 0-171.500000, 'payback90' ranges from 0-171.500000, 'pdate_day' ranges from 1-31, 'pdate_month' ranges from 6-8.

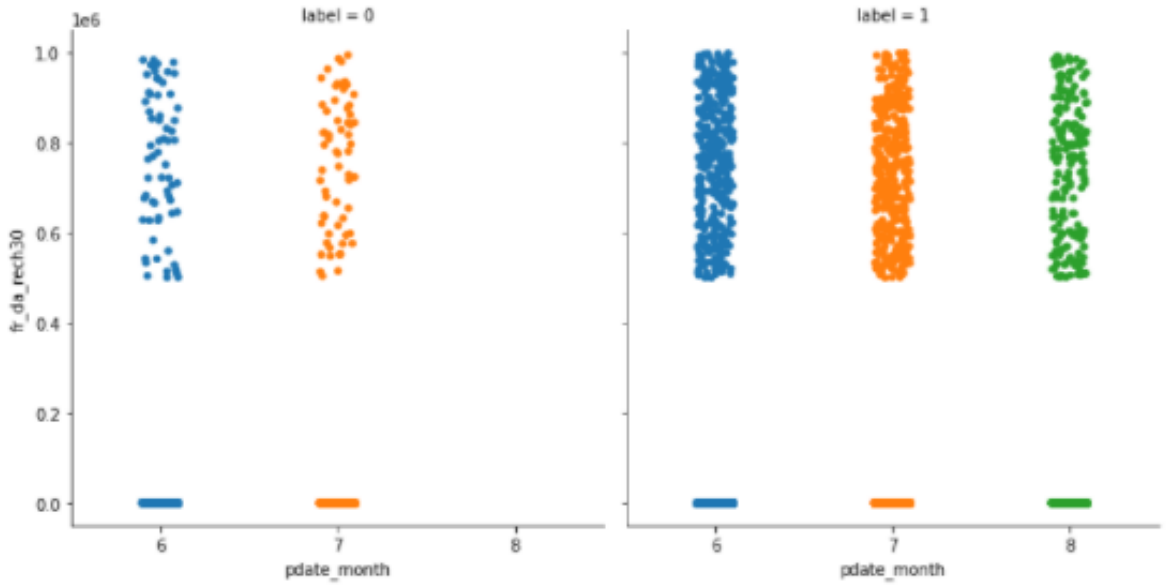
Columns aon, daily_decr30, daily_decr90, rental30, rental90, last_rech_date_ma, last_rech_date_da, last_rech_amt_ma, maxamnt_loans30, cnt_loans90, amnt_loans90, payback30, payback90 have mean>median hence right skewness is there and the difference between 75% and max values are high for some of the columns hence outliers might be there.

Standard deviation seem high for aon, daily_decr30, daily_decr90, rental30, rental90, last_rech_date_ma, last_rech_date_da, last_rech_amt_ma, maxamnt_loans30, cnt_loans90.

- **Category plots**

```
# Category plot for categorical data
sns.catplot(x='pdate_month',y='fr_da_rech30',data=df, col='label')
```

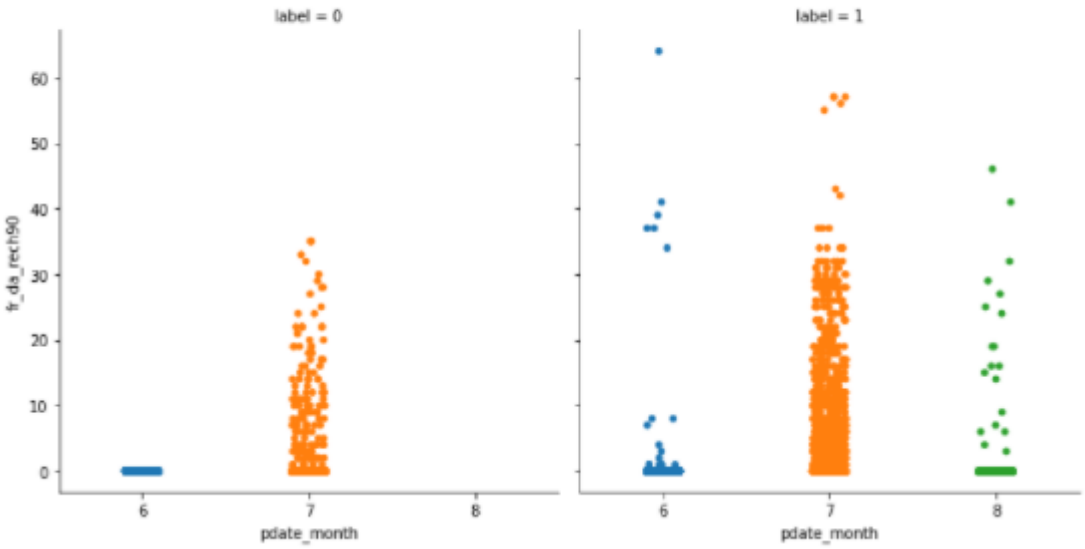
```
<seaborn.axisgrid.FacetGrid at 0x1df0ba58220>
```



Frequency of data account recharged in last 30 days is highest for the users paid back the credit amount within 5 days of issuing the loan that is also highest in July month and lowest in August month. Users did not pay back the credit amount within 5 days of issuing the loan have 0 Frequency of data account recharged in last 30 days and very few in other months.

```
sns.catplot(x='pdate_month',y='fr_da_rech90',data=df, col='label')
```

```
<seaborn.axisgrid.FacetGrid at 0x1df0bc91fa0>
```

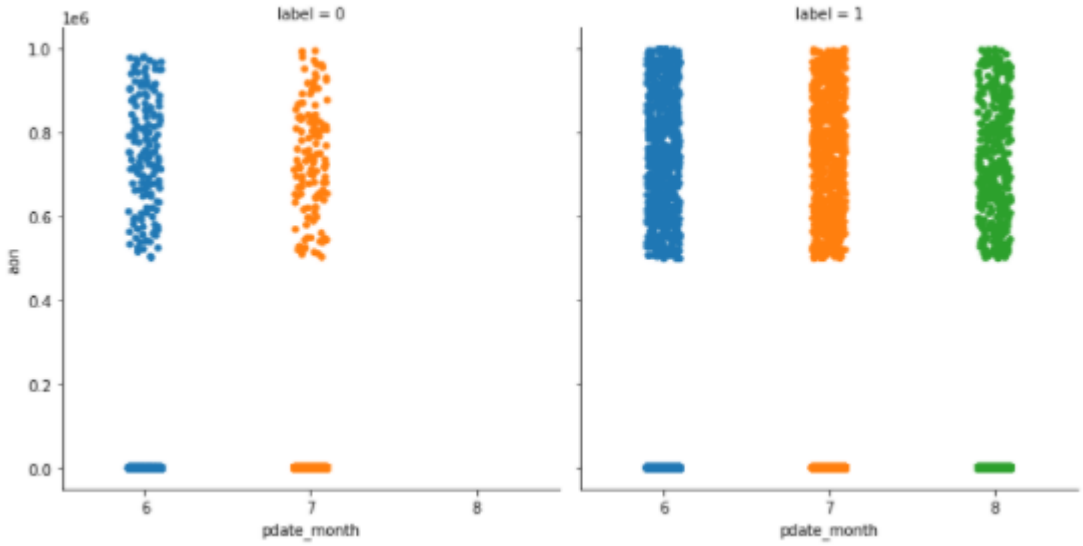


Number of times data account got recharged in last 90 days is highest for the users paid back the credit amount within 5 days of issuing the loan in July month and few in other months. Users did not pay back the credit amount within 5 days of issuing the loan have 0 Number of times data account got recharged in last 90

days in August and count is comparatively high in the month of July and very few in June month.

```
sns.catplot(x='pdate_month',y='aon',data=df, col='label')
```

```
<seaborn.axisgrid.FacetGrid at 0x1df0e2968e0>
```



Majority of age on cellular network in days is majorly for the users paid back the credit amount within 5 days of issuing the loan and highest n the month of July. Users did not paid back the credit amount within 5 days of issuing the loan have no count in the month of August they lie in June and July months.

- **Crosstab showing pivot tables between 2 variables**

```
# Pivot table showing counts
pd.crosstab(df['label'],df['medianamnt_loans30'])
```

medianamnt_loans30	0.0	0.5	1.0	1.5	2.0	3.0
label						
0	25173	440	520	0	0	0
1	170241	6098	6629	38	420	3

0.0 as Median of amounts of loan taken by the user in last 30 days has the highest count of the users that paid back and did not paid back the credit amount within 5 days of issuing the loan. There are no user that has not paid back the credit amount within 5 days of issuing the loan with the median amount of 1.5, 2.0 and 3.0 .are mostly used for online purchase.

```
pd.crosstab(df['label'],df['maxamnt_loans90'])
```

maxamnt_loans90	0	6	12
label			
0	0	25111	1022
1	2043	155803	25583

Maximum amount of loan taken by the users that paid back and did not paid back the credit amount within 5 days of issuing the loan, in last 90 days is highest for 6 as maximum amount followed by 12.

```
pd.crosstab(df['pdate_month'],df['medianamnt_loans90'])
```

medianamnt_loans90	0.0	0.5	1.0	1.5	2.0	3.0
pdate_month						
6	77789	2477	2863	0	0	0
7	81552	2286	1922	0	0	0
8	38052	905	1387	19	307	3

Median of amounts of loan taken by the user in last 90 days as 0.0 is highest in all the mentioned months but among these months July has the highest count followed by June and lastly August. Median of amounts of loan taken by the user in last 90 days above 1.0 has no users in the months June and July but August has few counts.

```
pd.crosstab(df['pdate_month'],df['maxamnt_loans90'])
```

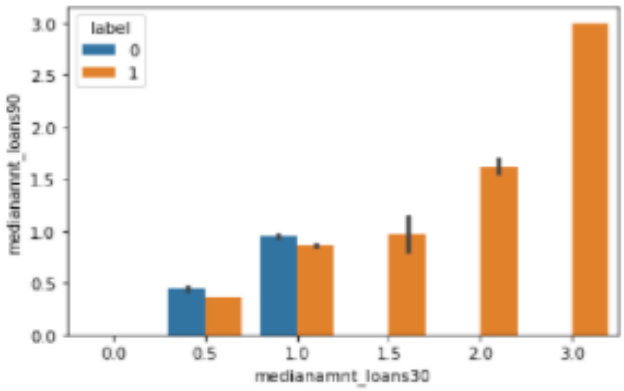
maxamnt_loans90	0	6	12
pdate_month			
6	23	82627	479
7	385	72370	13005
8	1635	25917	13121

Maximum amount of loan taken by the user in last 90 days as 6 has highest count in all the months data given followed by 12 as maximum amount of loan taken by the user in last 90 days and least counts are in 0. Maximum amount of loan of 0 has highest cont in August, Maximum amount 6 has highest count in June and Maximum amount 12 has highest count in August.

- **Barplot showing relations between variables**

```
# Barplot
sns.barplot(x='medianamnt_loans30',y='medianamnt_loans90',data=df, hue='label')
```

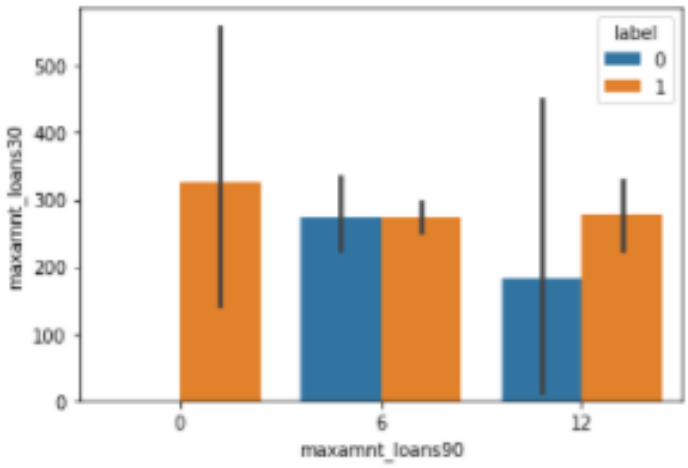
```
<AxesSubplot:xlabel='medianamnt_loans30', ylabel='medianamnt_loans90'>
```



Median of amounts of loan taken by the user in last 30 days as 0.5 and 1.0 only has the users who did not pay back the credit amount within 5 days of issuing the loan having higher Median of amounts of loan taken by the user in last 90 days comparatively. There are no users in 0.0 as medianamnt_loans30 and medianamnt_loans90. There are no users who did not pay back the credit amount within 5 days of issuing the loan in medianamnt_loans30 above 1.0.

```
sns.barplot(x='maxamnt_loans90',y='maxamnt_loans30',data=df, hue='label')
```

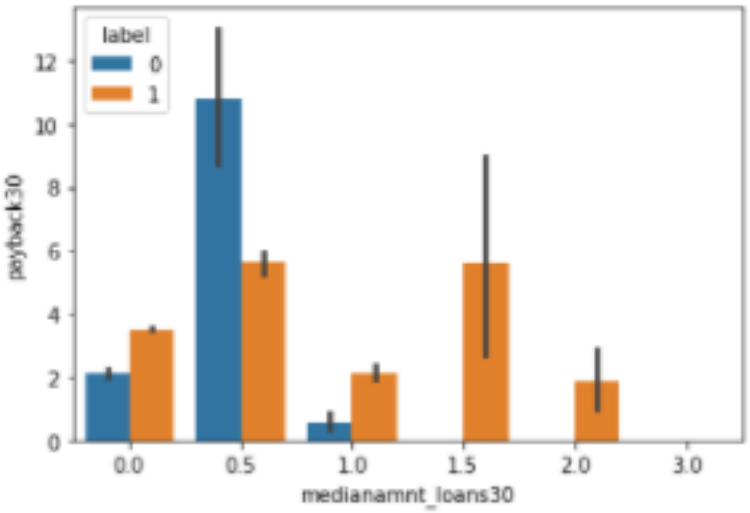
```
<AxesSubplot:xlabel='maxamnt_loans90', ylabel='maxamnt_loans30'>
```



There are no users who did not pay back the credit amount within 5 days of issuing the loan where maxamnt_loans90 is 0. maxamnt_loans30 is highest for 0 as maxamnt_loans90 where all the users have paid back the credit amount within 5 days of issuing the loan.

```
sns.barplot(x='medianamnt_loans30',y='payback30',data=df, hue='label')
```

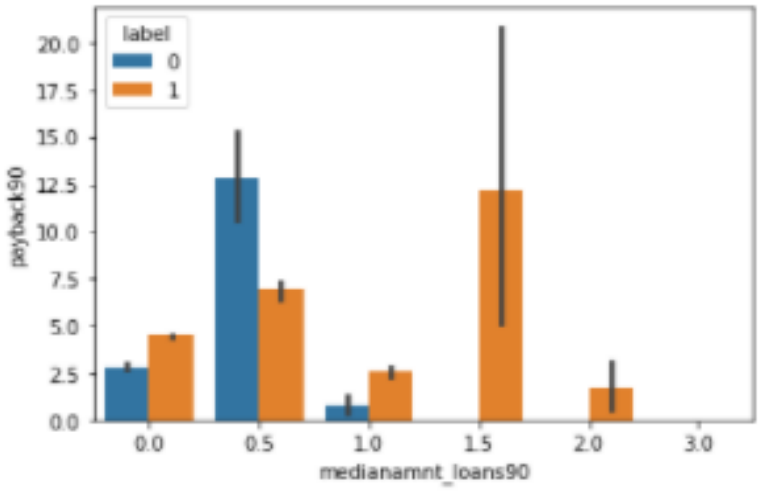
```
<AxesSubplot:xlabel='medianamnt_loans30', ylabel='payback30'>
```



Median of amounts of loan taken by the user in last 30 days above 1.0 has no users who did not pay back the credit amount within 5 days of issuing the loan, they lie in the range of 0 to 1. Users who did not pay back the credit amount within 5 days of issuing the loan has highest Average payback time in days over last 30 days at medianamnt_loans30 as 0.5.

```
sns.barplot(x='medianamnt_loans90',y='payback90',data=df, hue='label')
```

```
<AxesSubplot:xlabel='medianamnt_loans90', ylabel='payback90'>
```

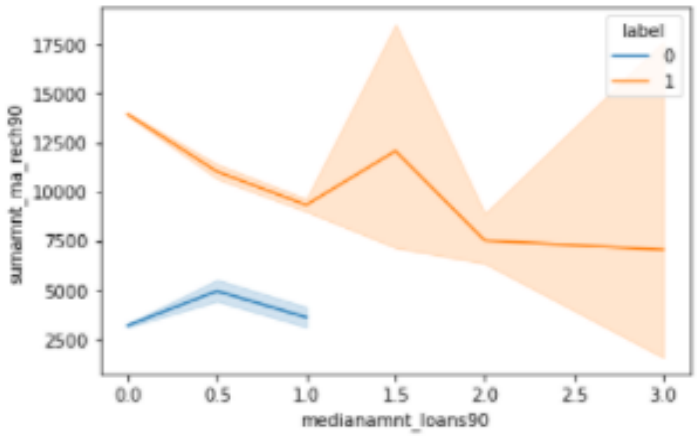


Median of amounts of loan taken by the user in last 90 days above 1.0 has no users who did not pay back the credit amount within 5 days of issuing the loan, they lie in the range of 0 to 1. Users who did not pay back the credit amount within 5 days of issuing the loan has highest Average payback time in days over last 90 days at medianamnt_loans30 as 0.5. Users who paid back the credit amount within 5 days of issuing the loan has highest payback90 when medianamnt_loans90 is 1.5.

- **Lineplot showing relation between**

```
# Lineplot
sns.lineplot(x='medianamnt_loans90',y='sumamnt_ma_rech90',data=df,hue='label')
```

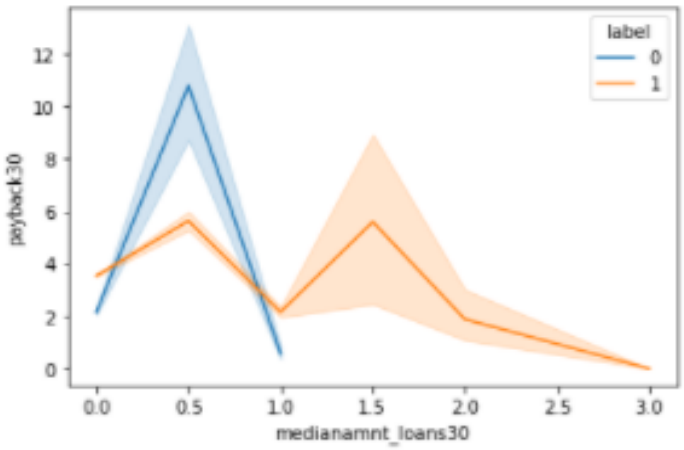
```
<AxesSubplot:xlabel='medianamnt_loans90', ylabel='sumamnt_ma_rech90'>
```



Total amount of recharge in main account over last 90 days (in Indonasian Rupiah) decreases with increase in Median of amounts of loan taken by the user in last 90 days with a hike in between where Median of amounts of loan taken by the user in last 90 days is 1.5 for the users paid back the credit amount within 5 days of issuing the loan. For the users did not pay back the credit amount within 5 days of issuing the loan medianamnt_loans90 lies between 0.0 to 1.0 and sumamnt_ma_rech90 first increases till the peak and then again decreases. sumamnt_ma_rech90 has a huge difference of amounts for the defaulters and non-defaulters.

```
sns.lineplot(x='medianamnt_loans30',y='payback30',data=df,hue='label')
```

```
<AxesSubplot:xlabel='medianamnt_loans30', ylabel='payback30'>
```

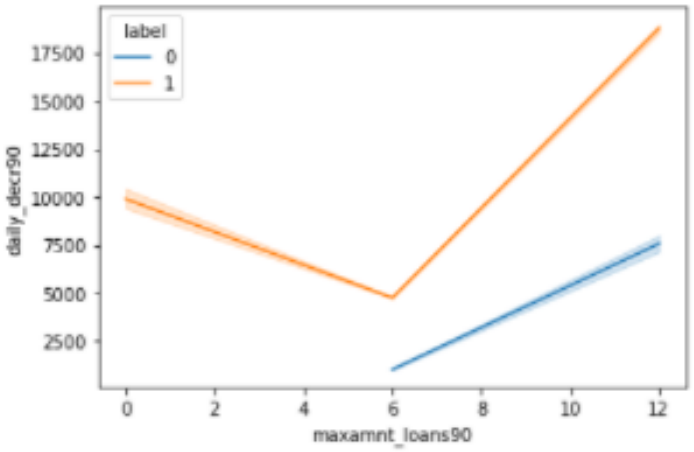


Average payback time in days over last 30 days decreases with increase in Median of amounts of loan taken by the user in last 30 days with hikes in between where Median of amounts of loan taken by the user in last 30 days is 0.5 and 1.5 for the users paid back the credit amount within 5 days of issuing the loan. For the users

did not pay back the credit amount within 5 days of issuing the loan Average payback time in days over last 30 days lies between 2 to 12 and medianamnt_loans30 lies between 0.0 to 1.0 and payback30 first increases till the peak and then again decreases.

```
sns.lineplot(x='maxamnt_loans90',y='daily_decr90',data=df,hue='label')
```

```
<AxesSubplot:xlabel='maxamnt_loans90', ylabel='daily_decr90'>
```

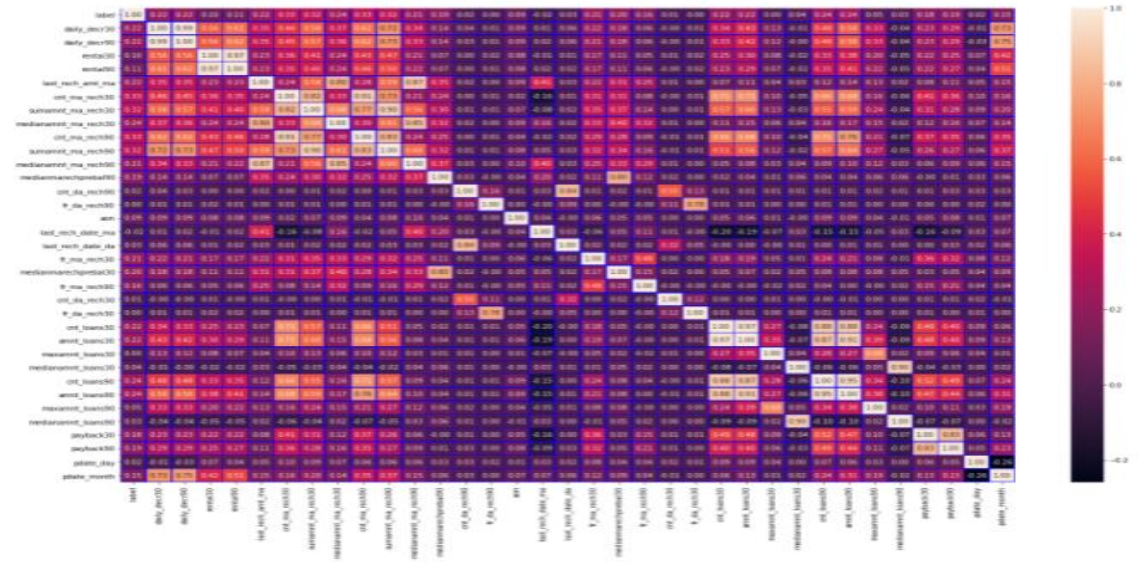


Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah) for the users paid back the credit amount within 5 days of issuing the loan first decreases and then increases linearly to the highest amount with increase in maximum amount of loan taken by the user in last 90 days. For the users did not pay back the credit amount within 5 days of issuing the loan maxamnt_loans90 is in the range of 6 to 12 which is increasing with the increase in daily_decr90. Daily amount spent from main account, averaged over last 90 days has amount difference for defaulters and non-defaulters.

- **Visualizing Correlation**

```
# Visualizing correlation
plt.figure(figsize=(25,15))
sns.heatmap(df.corr(),annot=True,linewidths=0.5,linecolor='b',fmt='.2f')
```

```
<AxesSubplot:>
```



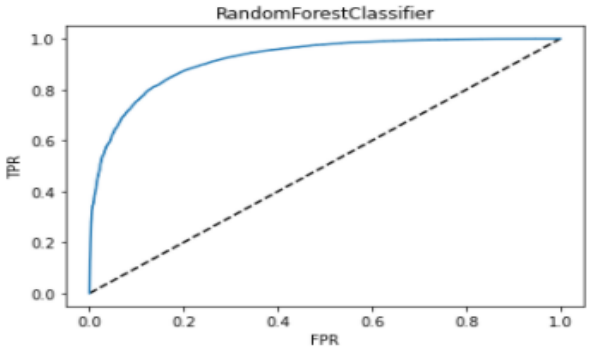
Columns 'daily_decr30', 'daily_decr90', 'last_rech_amt_ma', 'cnt_ma_rech30', 'sumamnt_ma_rech30', 'medianamnt_ma_rech30', 'cnt_ma_rech30', 'cnt_ma_rech90' has good correlation with target column 'label'.

'rental30' has strong positive correlation with 'daily_decr30', 'daily_decr90'. 'rental90' has strong positive correlation with 'daily_decr30', 'daily_decr90'. 'last_rech_amt_ma' has good correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90'. 'cnt_ma_rech30' has good correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90', 'last_rech_amt_ma'. 'sumamnt_ma_rech30' has strong correlation with 'daily_decr30', 'daily_decr90', 'last_rech_amt_ma', 'cnt_ma_rech30'. 'medianamnt_ma_rech30' has very strong postive correlation with 'last_rech_amt_ma' and 'sumamnt_ma_rech30'. 'cnt_ma_rech90' has good/strong positive correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90', 'last_rech_amt_ma', 'cnt_ma_rech30', 'sumamnt_ma_rech30', 'medianamnt_ma_rech30'. 'sumamnt_ma_rech90' has good/strong positive correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90', 'last_rech_amt_ma', 'cnt_ma_rech30', 'sumamnt_ma_rech30', 'medianamnt_ma_rech30', 'cnt_ma_rech90'. 'medianamnt_ma_rech90' has good/strong positive correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90', 'last_rech_amt_ma', 'cnt_ma_rech30', 'sumamnt_ma_rech30', 'medianamnt_ma_rech30', 'cnt_ma_rech90','sumamnt_ma_rech90'. 'fr_da_rech30' has strong correlation with 'fr_da_rech90'. 'pdate_month' has strong correlation with 'daily_decr30', 'daily_decr90', 'rental30', 'rental90' and good correlation with few other columns. Many independent columns have strong correlation with each other hence problem of multicollinearity exists here.

- **Plotting AUC-ROC Curve**

```
# Plotting the curve

plt.plot([0,1],[0,1],'k--')
plt.plot(fpr,tpr,label='RandomForestClassifier')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('RandomForestClassifier')
plt.show()
```



Getting good accuracy score and AUC-ROC score for the model selected

- Interpretation of the Results

AUC-ROC Curve

```
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score

rf=RandomForestClassifier(class_weight=None,criterion='entropy',max_features='log2',min_samples_leaf=3)
rf.fit(trainx,trainy)
pred=rf.predict(x_test)
acc1=accuracy_score(y_test,pred)
```

Applying AUC-ROC curve on selected model i.e. RandomForestClassifier()

```
pred_proba=rf.predict_proba(x_test)[:,1]
fpr,tpr,thresholds=roc_curve(y_test,pred_proba)
#print(fpr,tpr,thresholds)
```

Calculating fpr, tpr, thresholds for selected model

```
auc_score=roc_auc_score(y_test,pred_proba)
print('Accuracy : ',acc1)
print('ROC_AUC score : ',auc_score)
```

```
Accuracy :  0.9127503288992838
ROC_AUC score :  0.919904438169302
```

We are getting good accuracy score and AUC-ROC score for the model selected.

CONCLUSION

- ## Key Findings and Conclusions of the Study

  We are getting good accuracy of training and testing the given dataset after cleaning the data using Random Forest Classifier for model development and evaluation.

  Most of the Target variables are non-defaulters i.e. the user paid back the credit amount within 5 days of issuing the loan, which is a good sign of growing the business and taking it to the next level of progression.

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

  It helped in developing relevant programming abilities, demonstrated proficiency with statistical analysis of data, developed the ability to build and assess data-based models, executed statistical analyses and demonstrated skill in data management.

  Supervised learning algorithms are employed where the training data has output variables corresponding to the input variables. The algorithm analyses the input data and learns a function to map the relationship between the input and output variables. Supervised learning can further be classified into Regression, Classification, Forecasting, and Anomaly Detection.

  Logistic regression is a machine learning method used in the classification problem when you need to distinguish one class from another. The simplest case is a binary classification. It predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. Instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

  Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems. Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. We can represent any Boolean function on discrete attributes using the decision tree.

  Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as **bagging**. The basic idea behind this is to combine multiple decision trees in determining the final output rather than

relying on individual decision trees. Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

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

Data will change with time and hence relation between them as well. There are only two options: 5 & 10 Rs., for which the user needs to pay back 6 & 12 Rs respectively. The days of frequency of recharge, total amount taken, maximum amount recharged, average main account balance are considered for 30 days and 90 days only.

Future scope is this study can be proved very beneficial to low income families who are in need just with more awareness and by provisioning more amenities on the options given to the customers.

To further extend this study and improve the results the data set could be gathered from more and more regions to analyse the progress and outcomes of our contributions. The pay back money options can be increased based on the concession and benefits provided. The days considered for every amount related details can also be increased for better understanding and precise estimation.