FLIP ROBO

NAME OF THE PROJECT

# Micro Credit Project

Submitted by:

Grace Amirtham G

## 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
  This is a classic Business problem which helps Micro Financing Institutions and other Lending companies reduce Credit risks by recognizing potential Defaulters.

  Conceptual Background of the Domain Problem
  Before advancement of Data Science, loan lending companies used to risk a high rate of defaulting. Many a times a perfect candidate would display erratic financial and repayment behavior after being approved for loan. Machine Learning can help lenders predict potential defaulters before approving their candidature using their past data. The candidates' income, past debt and repayment behavior can be important metrics for the same.

  Review of Literature
- A Microfinance Institution (MFI) is an organization that offers financial services to low income populations. MFS becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income. The Microfinance services (MFS) provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans and so on.

- Motivation for the Problem Undertaken
- Many microfinance institutions (MFI), experts and donors are supporting the idea of using mobile financial services (MFS) which they feel are more convenient and efficient, and cost saving, than the traditional high-touch model used since long for the purpose of delivering microfinance services. 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.
- We are working with 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. 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).

## Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem
- 1) Data Exploration and Cleaning
- 2) Feature Selection
- 3) Data Visualization
- 4) Data Normalization
- 5) Oversampling of Minority class
- 6) Build Models

Data Sources and their formats

- The sample data is provided to us from our client database. It is hereby given to you for this exercise. 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.
- Data Preprocessing Done
- 1) Data Exploration and Cleaning
- On data exploration, I found that the dataset was imbalanced for the target feature(87.5% for Non-defaulters and 12.5% for Defaulters). Also, I found that the data had some very unrealitic values such as 999860 days which is not possible. Also, there were negative values for variables which must not have one (example:frequency,amount of recharge etc). All these unrealistic values were dropped which caused a data loss of 8% only.
- 2) Feature Selection
- Since there were 36 features, many of which I suspected were redundant because of the data duplication. It was imperative to select only most significant of them to make ML models more efficient and cost effective. The method used was 'Univariate

Selection' using chi-square test. I selected top 20 features which were highly significant.

- 3) Data Visualization
- On visualizing data, there were two important insights I gathered.
- a. Imbalance of data
- b. Distribution was not normal
- 4) Data Normalization
- Since the data was not normal, I normalized all the features except the target variable which was dichotomous(Values '1' and '0').
- 5) Oversampling of Minority class
- Since the data was expensive, I did not want to lose out on data by undersampling the majority class. Instead, I decided to oversample the minority class using SMOTE.
- 6) Build Models
- Since it was a supervised classification problem,
- I built 5 models to evaluate performance of each of them:
- a. Logistic Regression
- b. Linear SVM
- c. Decision Tree
- d. Random forest
- e. Gradient Boost Classifier

## Data Inputs- Logic- Output Relationships

- The dataset has 209593 rows and 37 columns.Label is the target variable. 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.

## Hardware and Software Requirements and Tools Used

Python was used.

Below are the libraries used.

import pandas as pd

import numpy as np[Pandas and numpy was used to import the dataset and create a dataframe]

import seaborn as sns[Used for plotting that is for data visualization]

from sklearn.preprocessing import LabelEncoder[Used for converting objects columns]

from scipy.stats import zscore[Used to remove outliers]

import matplotlib.pyplot as plt[Used for plotting that is for data visualization]

from scipy import stats

from statsmodels.stats.outliers_influence import variance_inflation_factor[Used to check highly correlated columns]

from imblearn.over_sampling import SMOTE[Used for balancing the imbalanced output variable]

from sklearn.metrics import roc_curve[Used to represt the classification report graphically]

import matplotlib.pyplot as plt

from sklearn.metrics import roc_auc_score

## Model/s Development and Evaluation

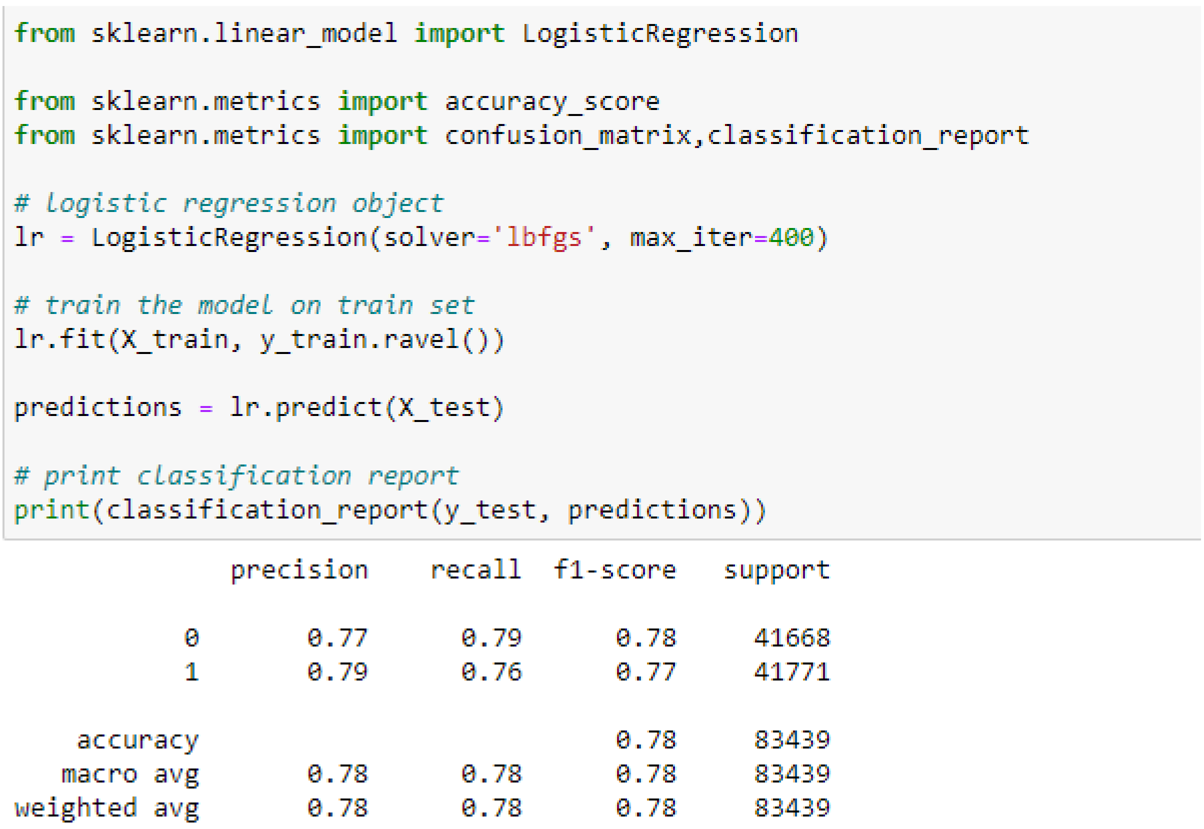
- The output was a classification problem.
  Testing of Identified Approaches (Algorithms)

Used train test split method

- Run and Evaluate selected models

  4 models were used to check the accuracy.

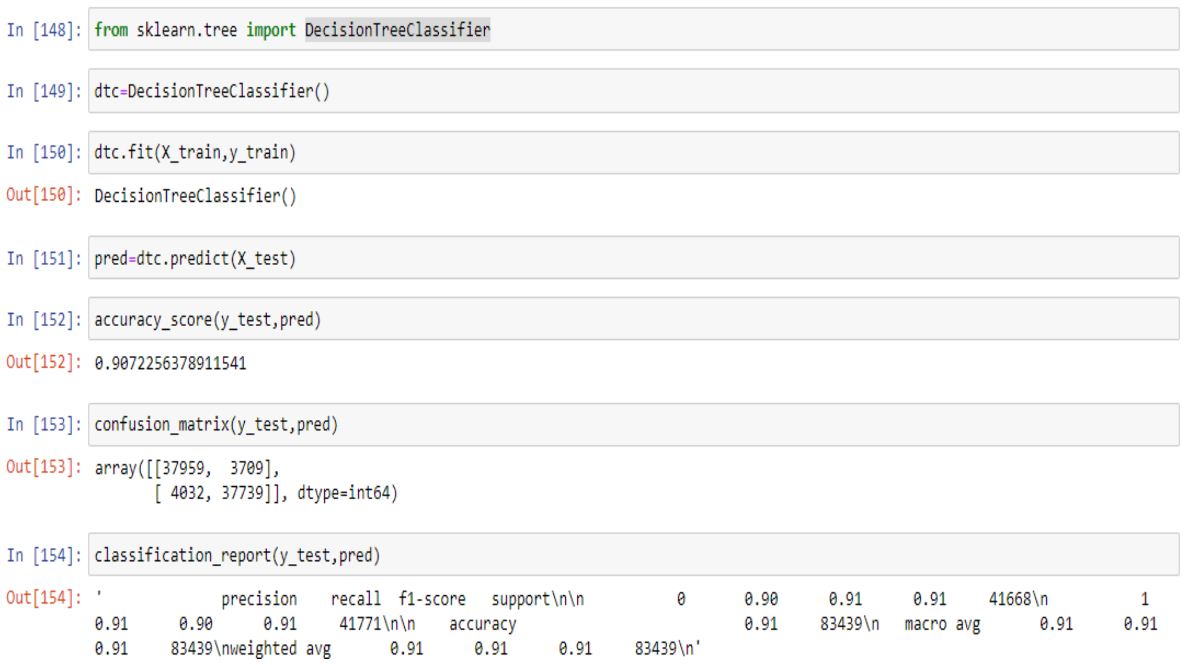
  1)Logistic Regression



```
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix,classification_report

# logistic regression object
lr = LogisticRegression(solver='lbfgs', max_iter=400)

# train the model on train set
lr.fit(X_train, y_train.ravel())

predictions = lr.predict(X_test)

# print classification report
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.77      0.79      0.78     41668
           1       0.79      0.76      0.77     41771

    accuracy                           0.78     83439
   macro avg       0.78      0.78      0.78     83439
weighted avg       0.78      0.78      0.78     83439
```

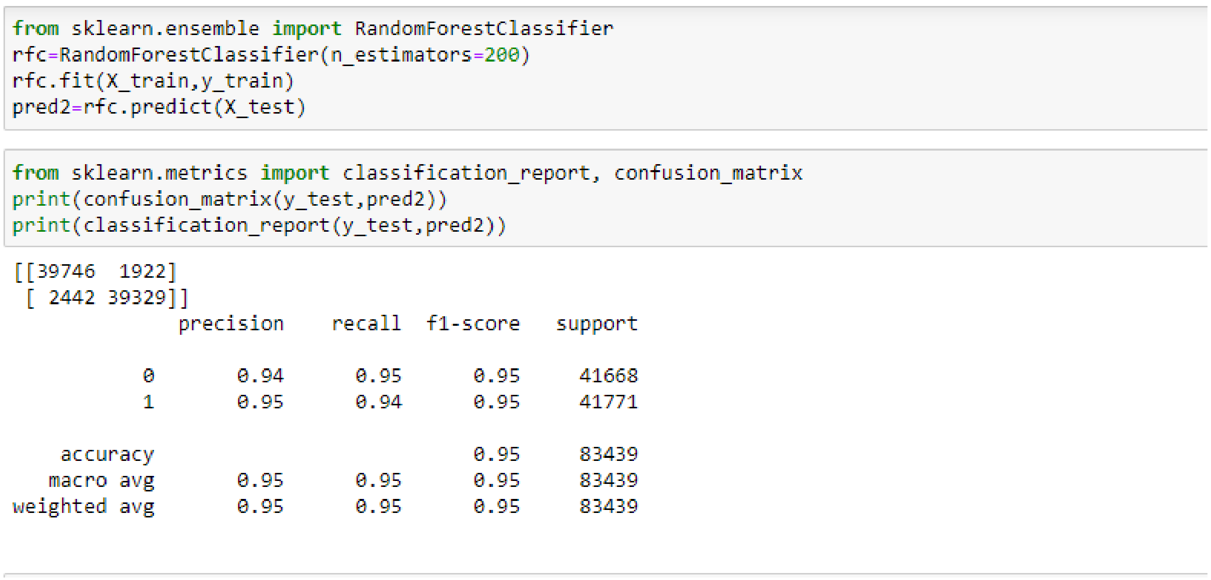
- 2)Decision Tree Classifier



```
In [148]: from sklearn.tree import DecisionTreeClassifier

In [149]: dtc=DecisionTreeClassifier()

In [150]: dtc.fit(X_train,y_train)

Out[150]: DecisionTreeClassifier()

In [151]: pred=dtc.predict(X_test)

In [152]: accuracy_score(y_test,pred)

Out[152]: 0.9072256378911541

In [153]: confusion_matrix(y_test,pred)

Out[153]: array([[37959,  3709],
                 [ 4032, 37739]], dtype=int64)

In [154]: classification_report(y_test,pred)

Out[154]: '              precision    recall  f1-score   support\n\n           0       0.90      0.91      0.91     41668\n           1
          0.91      0.90      0.91     41771\n\n    accuracy                           0.91     83439\n   macro avg       0.91      0.91
          0.91     83439\nweighted avg       0.91      0.91      0.91     83439\n'
```

-

- 3)Random forest classifier

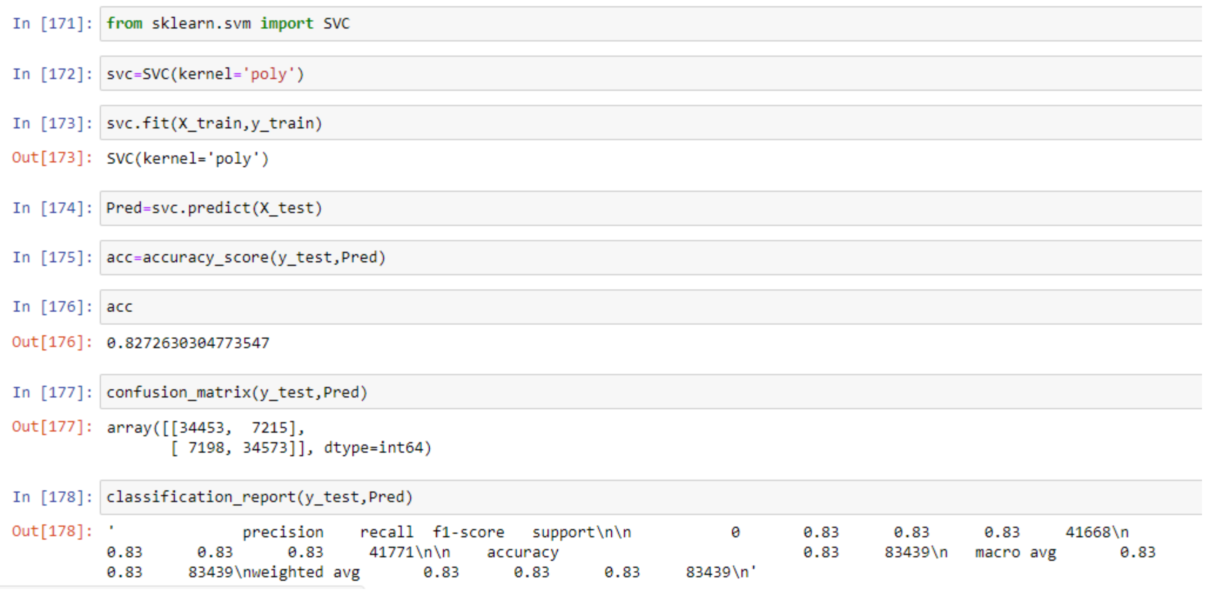


```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier(n_estimators=200)
rfc.fit(X_train,y_train)
pred2=rfc.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,pred2))
print(classification_report(y_test,pred2))
```

```
[[39746  1922]
 [ 2442 39329]]
              precision    recall  f1-score   support

           0       0.94      0.95      0.95     41668
           1       0.95      0.94      0.95     41771

    accuracy                           0.95     83439
   macro avg       0.95      0.95      0.95     83439
weighted avg       0.95      0.95      0.95     83439
```

- 4)SVM



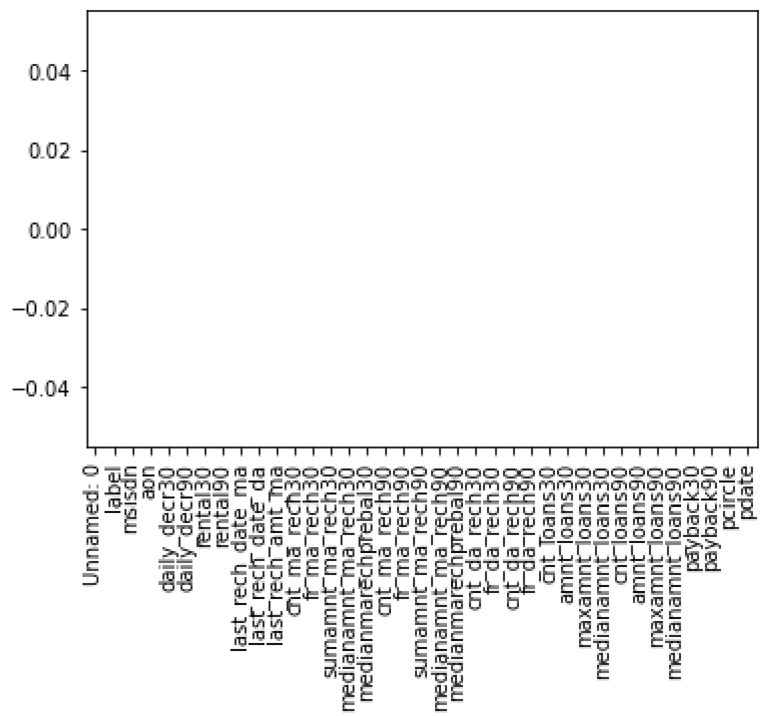
```
In [171]: from sklearn.svm import SVC

In [172]: svc=SVC(kernel='poly')

In [173]: svc.fit(X_train,y_train)
Out[173]: SVC(kernel='poly')

In [174]: Pred=svc.predict(X_test)

In [175]: acc=accuracy_score(y_test,Pred)

In [176]: acc
Out[176]: 0.8272630304773547

In [177]: confusion_matrix(y_test,Pred)
Out[177]: array([[34453,  7215],
                 [ 7198, 34573]], dtype=int64)

In [178]: classification_report(y_test,Pred)
Out[178]: '              precision    recall  f1-score   support\n\n           0       0.83      0.83      0.83     41668\n
          0.83      0.83      0.83     41771\n\n    accuracy                           0.83      83439\n   macro avg       0.83
          0.83     83439\nweighted avg       0.83      0.83      0.83     83439\n'
```

- Key Metrics for success in solving problem under consideration
- Used AUC-ROC CURVE Plot for each model.
- AUC-ROC curve helped us to visualize how well our machine learning classifier is performing.
- The AUC score was also calculated for each of the model.

- Visualizations
- Heatmap was used to check the null values in the dataset.

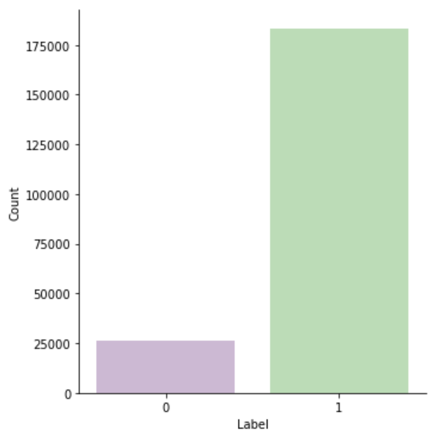
```
df.isna().sum().plot(kind="bar")
```

```
<AxesSubplot:>
```



- 
- From the above plot we can see that there are no null values in the dataset.
- Category plot was used to represent the target variable graphically.We can see that the target variable is

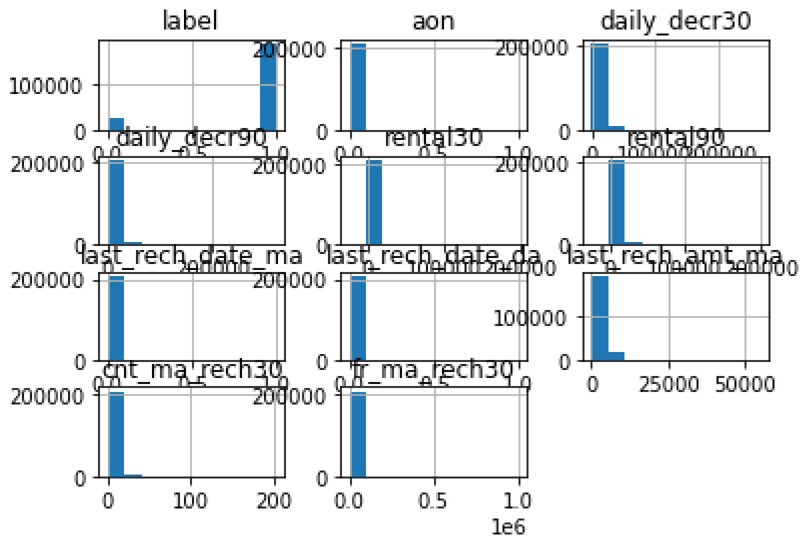
```
g=sns.catplot(x="label",data=df,kind="count",palette="PRGn",alpha=1)
g.set(xlabel="Label",ylabel="Count")
plt.show()
```



- imbalanced.

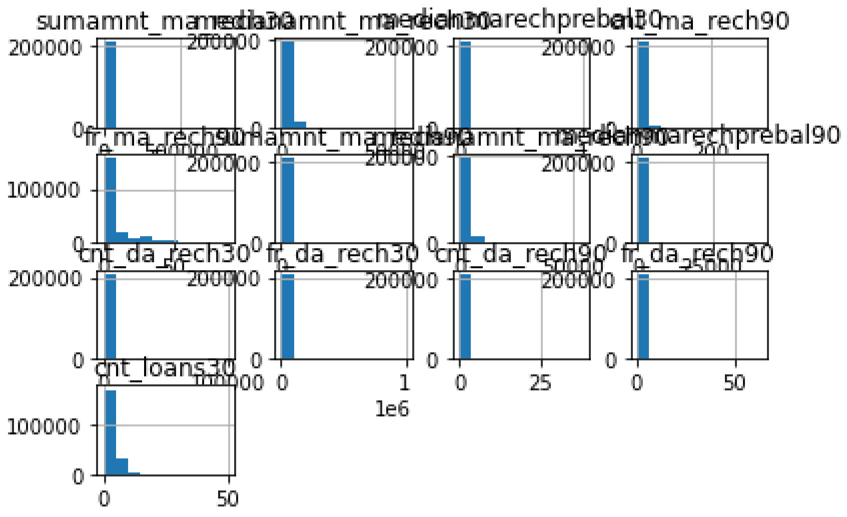
## Histogram plot was used to Visualizing the distribution of each variable

```
hist = df.iloc[:,0:12].hist()
```



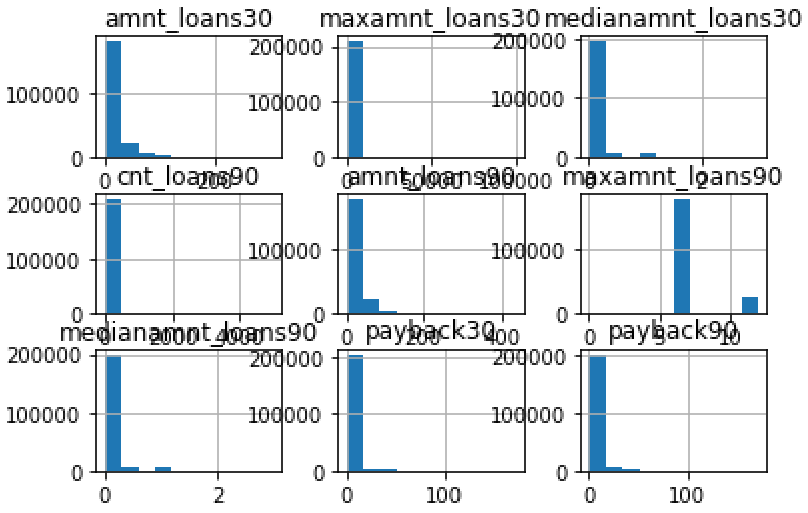
●

```
hist = df.iloc[:,12:25].hist()
```

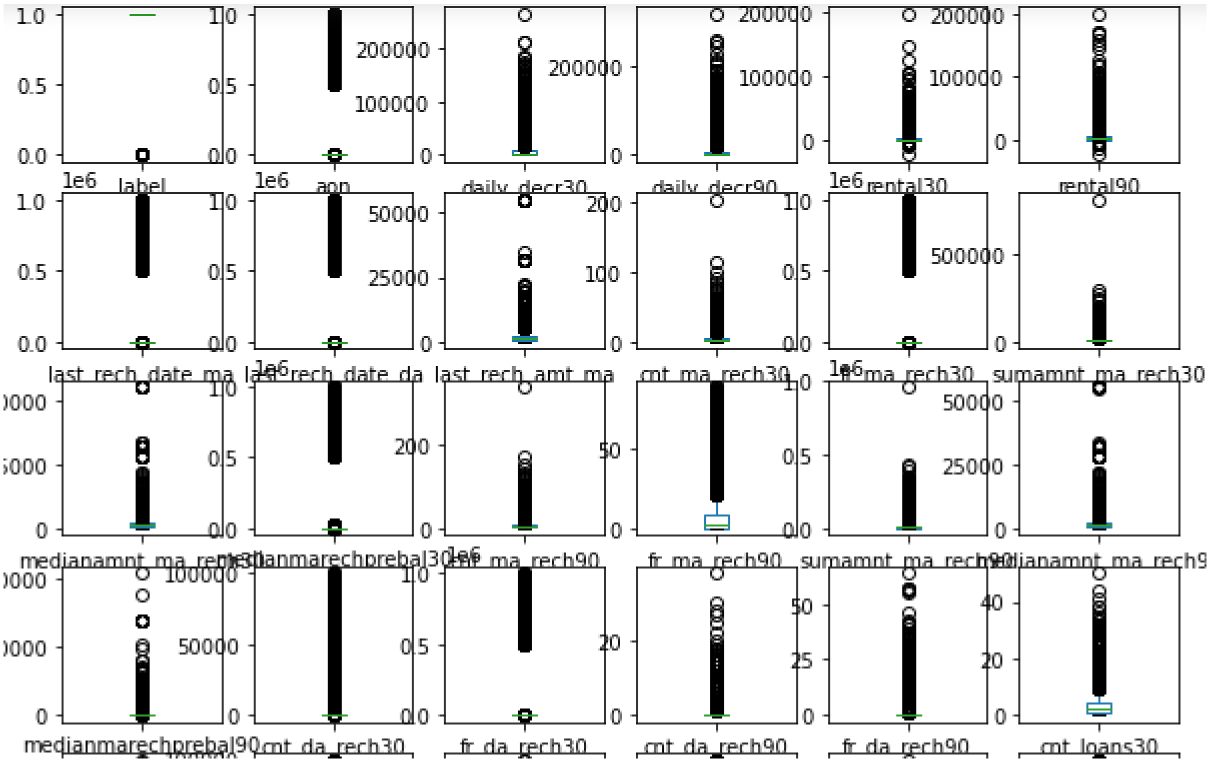


●

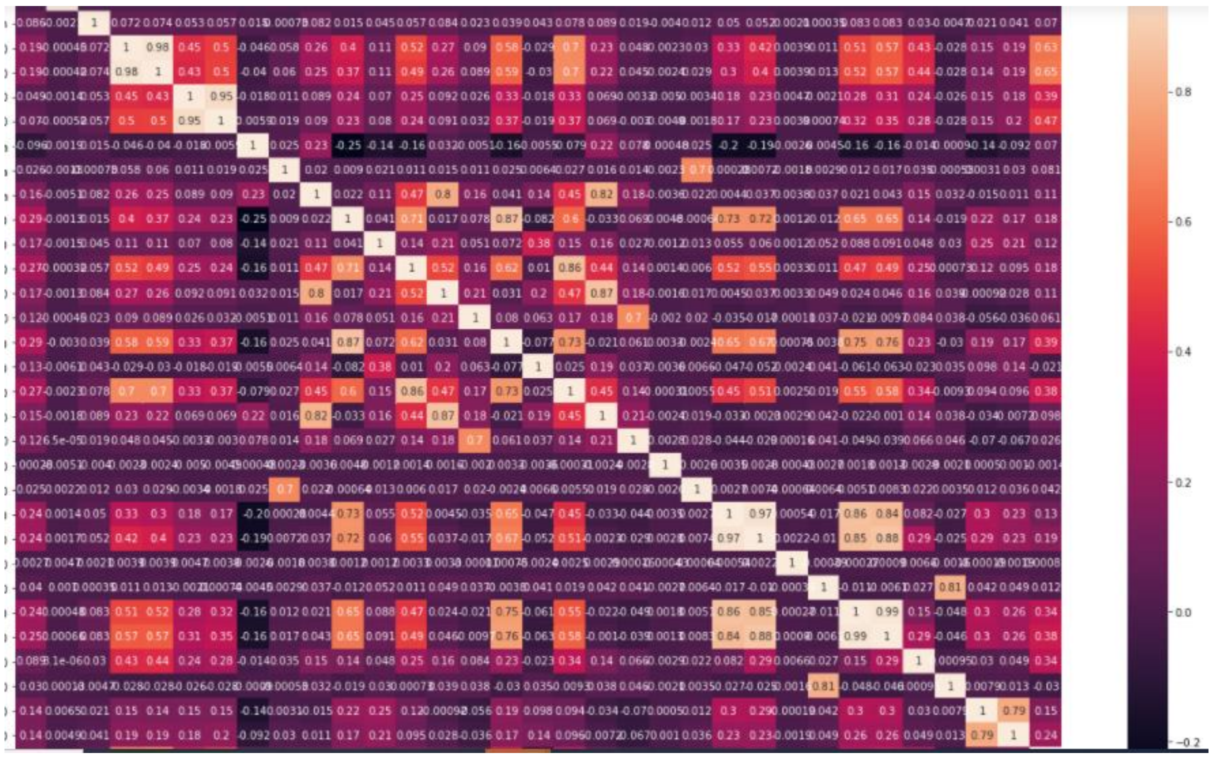
```
hist = df.iloc[:,25:35].hist()
```



- 
- Box plot was used to check the outliers in the dataset.Almost all the columns have outliers



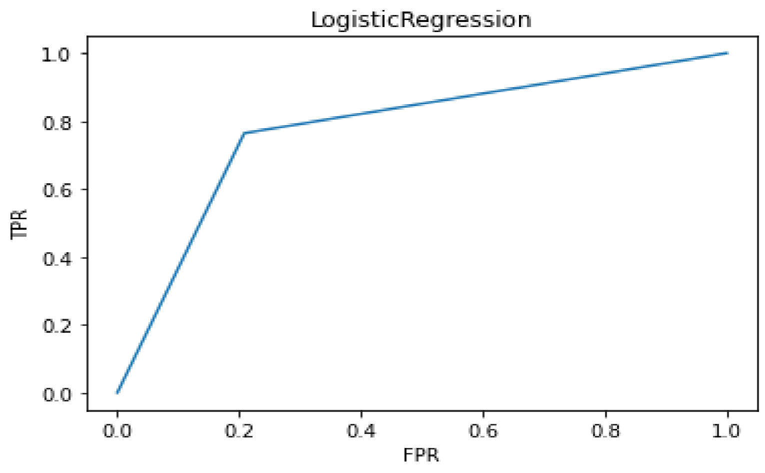
- 
- Heatmap was used to check the correlation between the columns.Lighter shades signify that they are highly correlated to each other.



AUC-ROC Curve was plotted o check how well the model performs.Below is the AUC_ROC curve for Logistic regression.

```
plt.plot(fpr,tpr,label='LogisticRegression')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('LogisticRegression')
```

```
Text(0.5, 1.0, 'LogisticRegression')
```



```
auc_score=roc_auc_score(y_test,lr.predict(X_test))
```

```
auc_score
```

```
0.7778665096745775
```

●

- Interpretation of the Results

  The dataset did not have null values.The dataset had a column with the object datatype.Dropped few columns that was not useful for analysis.Checked for outliers and skewness.Used zscore method to remove outliers.Used power transform method to remove skewness.Heat map was used to check multicollinearity.Dropped the columns that was highly correlated.Used minmax scalaing to scale the input variable.Used SMOTE technique to balance the imbalanced target variable.The target variable was of Classification problem.Used 4 models to check the accuracy.

## CONCLUSION

- Key Findings and Conclusions of the Study
  Comparing the accuracy scores of the four models used and the AUC-ROC curve.Random forest classifier is the best model.