MICRO CREDIT DEFAULTER

Submitted by Anum Yusuf

ACKNOWLEDGMENT

My special gratitude to Flip Robo Technologies who has shared the data set and help us in Completing the project. This project helps us to

understand the EDA, Data cleaning and making the prediction using various machine learning model.

It helps me to understand what is micro credit and how it works? It helps the people who is not able to pay the huge loan amount with high interest rate. How the finance company go through this process, how they analysis?

References:

Microcredit - Overview, How It Works, History, and Disadvantages (corporatefinanceinstitute.com)

Microcredit (investopedia.com)

Microcredit: impacts and limitations | The Abdul Latif Jameel Poverty Action Lab

INTRODUCTION

Business Problem Framing:

- 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.
- 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).
- ✦ 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.

Conceptual Background of the Domain Problem:

- 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).
- 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.

- We have to build a model and make the prediction that where the people will pay the loan within 5 days or not. Based on this prediction Micro Finance Company will work based on the prediction. To make the better prediction we have clearly understand each feature. As the data is huge, we need to do the properly cleaning of data.

Review of Literature

An attempt has been made in this report to review the available literature in the area of microfinance. Approaches to microfinance, issues related to measuring social impact versus profitability of MFIs, issue of sustainability, variables impacting sustainability, effect of regulations of profitability and impact assessment of MFIs have been summarized in the below report. We hope that the below report of literature will provide a platform for further research and help the industry to combine theory and practice to take microfinance forward and contribute to alleviating the poor from poverty.

Motivation for the Problem Undertaken

I have to build a model with available independent variable data set by thorough analysis of data. The model will go to management for further research. This micro credit model will help the Finance Company to decide which is defaulter and nondefaulter. Who will return the loan amount within 5 days? So, they can focus on the area which will yield in high return. The relationship between the prediction and economy is important, that will drive a motivation in understand the problem and providing the solution for that problem.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

In this problem, our target variable is label. It has 2 classes 1 and 0. 1 indicated that the loan amount will get payed within 5 days which is non defaulter. 0 indicated that loan amount will not be payed within 5 days which is a defaulter. There is no null value present in the data. In some columns there are more than 90% values have zeros, So, we have dropped those columns to avoid the multicollinearity issues. We have checked the correlation of data. In most of the columns Outliers are present, we have removed the outliers using the zscore method. We have used the various visualization plot for all features. We used distribution plot, violin plot, scatterplot, strip plot, count plot. In the distribution plot we observed that there is a skewness present in data. We have removed the skewness using the yeo-johnson method. Then we have built the model, checked their accuracy score, confusion matrix and classification report. We also checked their cross-validation score of each model. We have done the hyper parameter tuning to check if we can increase the accuracy or not. Then final we have made the prediction from the saved model.

Data Sources and their formats

The data was provided by the Flip Robo Technologies and it is in the .csv format. The data is huge it has 209593 rows and 36 columns. It contains integer, float and object data types. From this data set it ask to predict which is defaulter and nondefaulter. Our target variable in this data set is label.

Data Features Description:

label	Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure}
msisdn	mobile number of user
aon	age on cellular network in days
daily_decr30	Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)
daily_decr90	Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)
rental30	Average main account balance over last 30 days
rental90	Average main account balance over last 90 days
last_rech_date_ma	Number of days till last recharge of main account
last_rech_date_da	Number of days till last recharge of data account
last_rech_amt_ma	Amount of last recharge of main account (in Indonesian Rupiah)
cnt_ma_rech30	Number of times main account got recharged in last 30 days
fr_ma_rech30	Frequency of main account recharged in last 30 days
sumamnt_ma_rech30	Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)
medianamnt_ma_rech30	Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)
medianmarechprebal30	Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)
cnt_ma_rech90	Number of times main account got recharged in last 90 days
fr_ma_rech90	Frequency of main account recharged in last 90 days
sumamnt_ma_rech90	Total amount of recharge in main account over last 90 days (in Indonasian Rupiah)
medianamnt_ma_rech90	Median of amount of recharges done in main account over last 90 days at user level (in Indonasian Rupiah)
medianmarechprebal90	Median of main account balance just before recharge in last 90 days at user level (in Indonasian Rupiah)
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

- First step, I have imported the necessary Libraries and imported the data set.
- Checked the nunique, info of data, shape of data, statistical summary of data.
- Our data contains integer, float and object data type. We have converted the categorical data to numerical data using the Label Encoder.
- In some column there are more than 90% zero values, we have dropped that column.
- We observed that data contains the Outliers and skewness. We removed the Outliers using zscore method and removed the skewness using the Power Transformer (yeojohnson) method.
- Our target variable is label, it is imbalance. We used the over sampling technique to make our data balance using the SMOTE technique.
- We have extracted the day, month from pdate column, then dropped the pdate column after the extraction.

Data Inputs- Logic- Output Relationships

To checked the relationship between the features and target, we used the distribution plot, strip plot, scatter plot. We observe that relationship with the target variable count is high which is non defaulter. In our target variable it has 2 classes 1 and 0. 1 indicates the non-defaulter which is the people will pay the loan amount with in 5 days. 0 indicates that they will not pay the loan amount within 5 days which is defaulter. With the visualization plot we can understand the relationship between each feature with the target variables.

Hardware and Software Requirements and Tools Used

Here is the hardware and software used in the project.

- Processor – core i3
- RAM – 12 GB
- SSD – 250 GB **Software:**

Anaconda Jupyter Notebook.

Libraries:

```
# Importing Libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
import warnings
warnings.filterwarnings('ignore')
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import SGDClassifier
```

import pandas as pd:

pandas are a popular Python-based data analysis toolkit which can be imported using import pandas as pd. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a numpy matrix array. This makes pandas a trusted ally in data science and machine learning.

import numpy as np:

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a

multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

import seaborn as sns:

Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas' data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data. **import matplotlib.pyplot as plt**:

matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

We used the various machine learning model for training and testing the data for making the prediction.

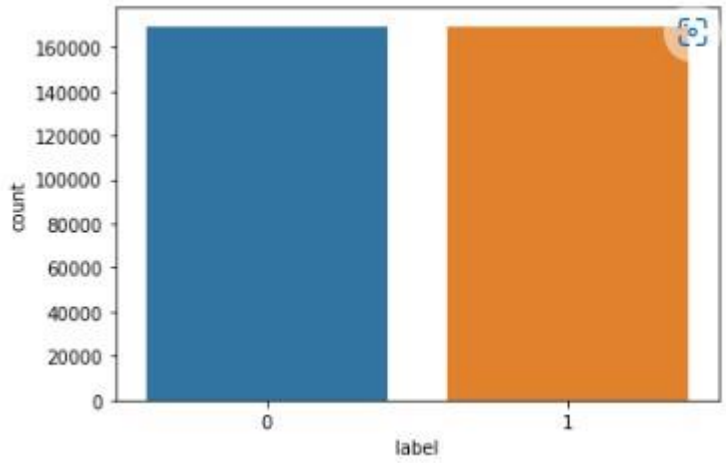
Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

In our data set Outliers are present, we removed the data set using the zscore method. Also, skewness is present in most of the columns. We removed the skewness using the yeo-johnson method. We checked the correlation of data. We dropped the columns that are having more than 90% zero values. Our target variable is imbalance, we balanced the data using the Over sampling technique SMOTE. We scaled the data for our features independent variables.

```
y.value_counts()
```

```
0    169493
1    169493
Name: label, dtype: int64
```



Testing of Identified Approaches (Algorithms)

Our target variable is label and it has 2 classes 1 non-defaulter and 0-Defaulter. So, it is a binary classification problem. I have to use the Classification model for training and testing. I have checked the cross-validation score, confusion matrix and classification report for each model. Then we have done the hyper parameter tuning to increase the accuracy for our final model.

Here is the classification algorithm used in the Micro Credit Defaulter project.

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Gradient Boosting Classifier
- SGD Classifier
- KNeighbors Classifier

Key Metrics for success in solving problem under consideration

The following metrics used in the project.

Precision can be seen as a measure of quality; higher precision means that an algorithm returns more relevant results than irrelevant ones.

· **Recall** is used as a measure of quantity and high recall means that an algorithm returns most of the relevant results.

· **Accuracy score** is used when the True Positives and True negatives are more important. Accuracy can be used when the class distribution is similar.

· **F1-score** is used when the False Negatives and False Positives are crucial. While F1-score is a better metric when there are imbalanced classes.

· **Cross_val_score:** To run cross-validation on multiple metrics and also to return train scores, fit times and score times. Get predictions from each split of crossvalidation for diagnostic purposes. Make a scorer from a performance metric or loss function.

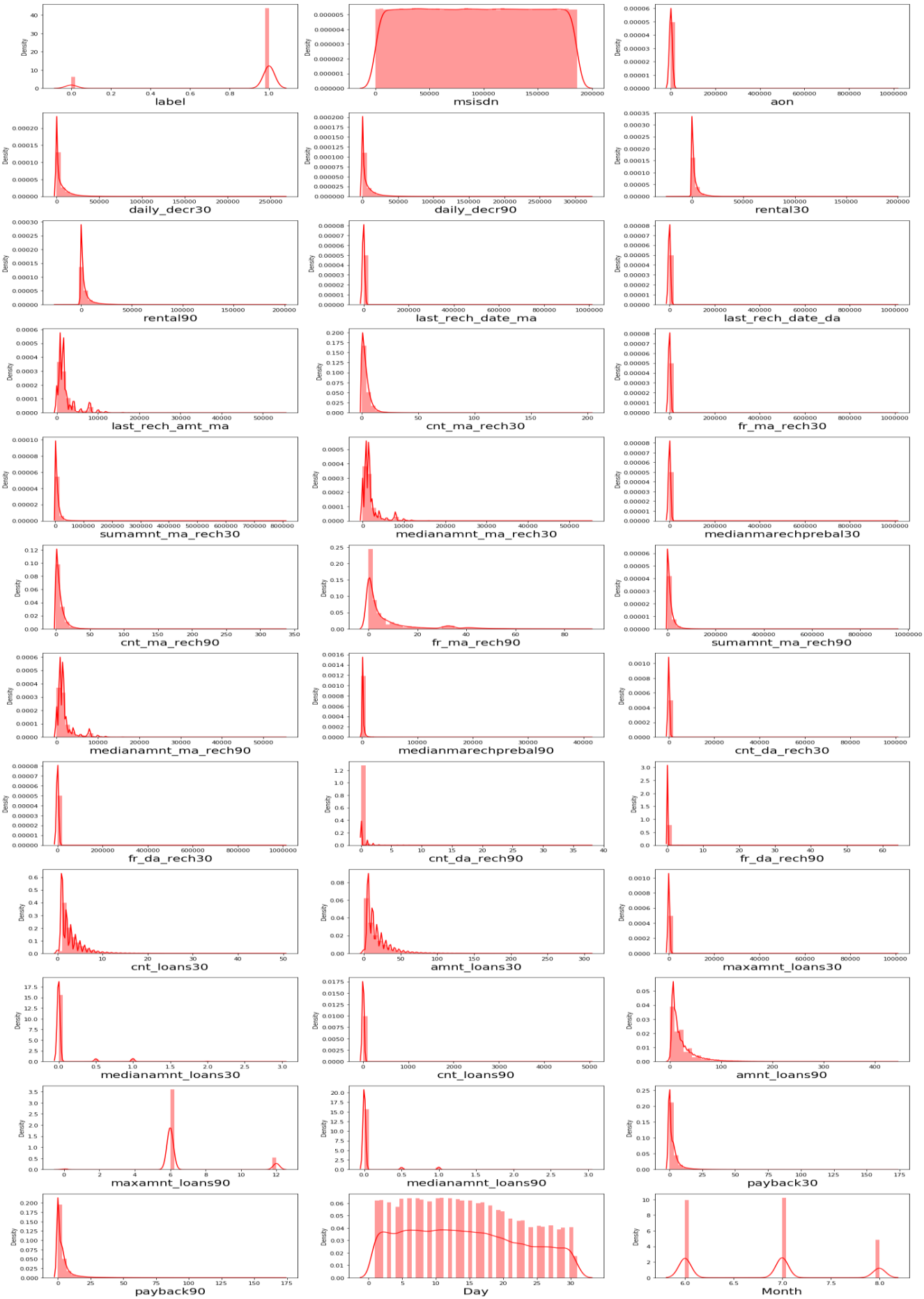
· **AUC_ROC _score: ROC curve.** It is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0

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

from sklearn. metrics import accuracy_score, confusion_matrix, classification report

Visualizations

In have used distribution plot, scatter plot, count plot, strip plot, violin plot for visualization of data.


label
msisdn
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
medianmarechprebal90
cnt_da_rech30
fr_da_rech30
cnt_da_rech90
fr_da_rech90
cnt_loans30
amnt_loans30
maxamnt_loans30
medianamnt_loans30
cnt_loans90
amnt_loans90
maxamnt_loans90
medianamnt_loans90
payback30
payback90
Day
Month


label
msisdn
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
medianmarechprebal90
cnt_da_rech30
fr_da_rech30
cnt_da_rech90
fr_da_rech90
cnt_loans30
amnt_loans30
maxamnt_loans30
medianamnt_loans30
cnt_loans90
amnt_loans90
maxamnt_loans90
medianamnt_loans90
payback30
payback90
Day
Month

Observations:

1. Customers with high value of Age on cellular network in days(aon) are maximum defaulters(who have not paid there loan amount-0).

2. Customers wit th high value of Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)(daily_decr30) are maximum Nondefaulters(who have paid there loan amount-1).

3. Customers with high value of Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)(daily_decr90) are maximum Nondefaulters(who have paid there loan amount-1).

4. Customers with high value of Average main account balance over last 30 days(rental30) are maximum Non-defaulters(who have paid there loan amount-1).

5. Customers with high value of Average main account balance over last 90 days(rental90) are maximum Non-defaulters(who have paid there loan amount-1).

6. Customers with high Number of days till last recharge of main account(last_rech_date_ma) are maximum Non-defaulters(who have paid there loan amount-1).

7. Customers with high value of Amount of last recharge of main account (in Indonesian Rupiah)(last_rech_amt_ma) are maximum Non-defaulters(who have paid there loan amount-1).

8.Customers with high value of Number of times main account got recharged in last 30 days(cnt_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1).

9. Customers with high value of Frequency of main account recharged in last 30 days(fr_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1) and also the count is high for defaulters comparitively Non-defaulters are more in number.

10. Customers with high value of Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)(sumamnt_ma_rech30) are maximum Nondefaulters(who have paid there loan amount-1).

11. Customers with high value of Median of amount of recharges done in main account over last 30 days at user level (in Indonesian

Rupiah)(medianamnt_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1).

12. Customers with high value of Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)(medianmarechprebal30) are maximum defaulters(who have not paid there loan amount-0).

13. Customers with high value of Number of times main account got recharged in last 90 days(cnt_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).

14. Customers with high value of Frequency of main account recharged in last 90 days(fr_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).

15. Customers with high value of Total amount of recharge in main account over last 90 days (in Indonasian Rupiah)(sumamnt_ma_rech90) are maximum Nondefaulters(who have paid there loan amount-1).

16. Customers with high value of Median of amount of recharges done in main account over last 90 days at user level (in Indonasian Rupiah)(medianamnt_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).

17. Customers with high value of Median of main account balance just before recharge in last 90 days at user level (in Indonasian Rupiah)(medianmarechprebal90) are maximum Non-defaulters(who have paid there loan amount-1).

18. Customers with high value of Number of loans taken by user in last 30 days(cnt_loans30) are maximum Non-defaulters(who have paid there loan amount-1).

19. Customers with high value of Total amount of loans taken by user in last 30 days(amnt_loans30) are maximum Non-defaulters(who have paid there loan amount-1).

ith high value of maximum amount of loan taken by the user in last 30 days (maxamnt_loans30) are maximum Non-defaulters (who have paid there loan amount-1).

21. Customers with high value of Number of loans taken by user in last 90 days (cnt_loans90) are maximum Non-defaulters (who have paid there loan amount-1).

22. Customers with high value of Total amount of loans taken by user in last 90 days (amnt_loans90) are maximum Non-defaulters (who have paid there loan amount-1).

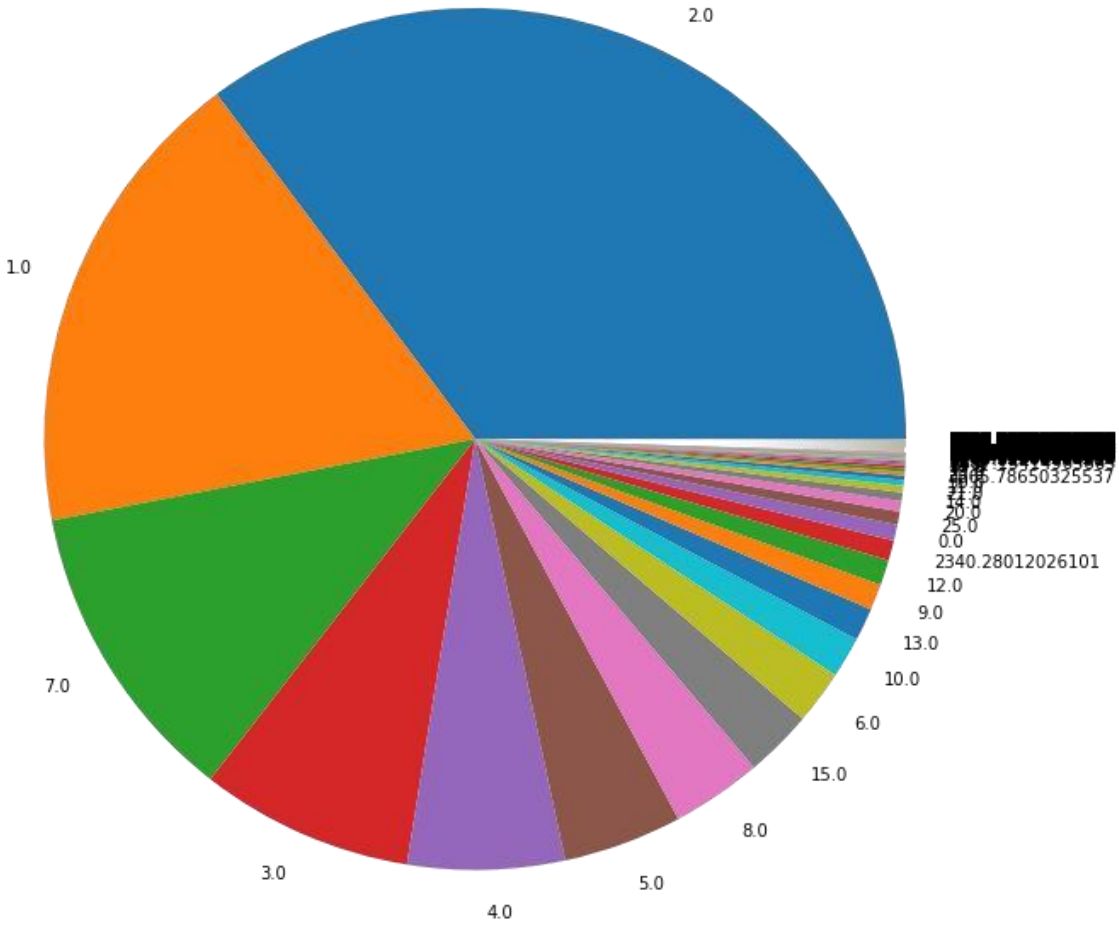
23. Customers with high value of maximum amount of loan taken by the user in last 90 days (maxamnt_loans90) are maximum Non-defaulters (who have paid there loan amount-1).

24. Customers with high value of Average payback time in days over last 30 days (payback30) are maximum Non-defaulters(who have paid there loan amount-1).

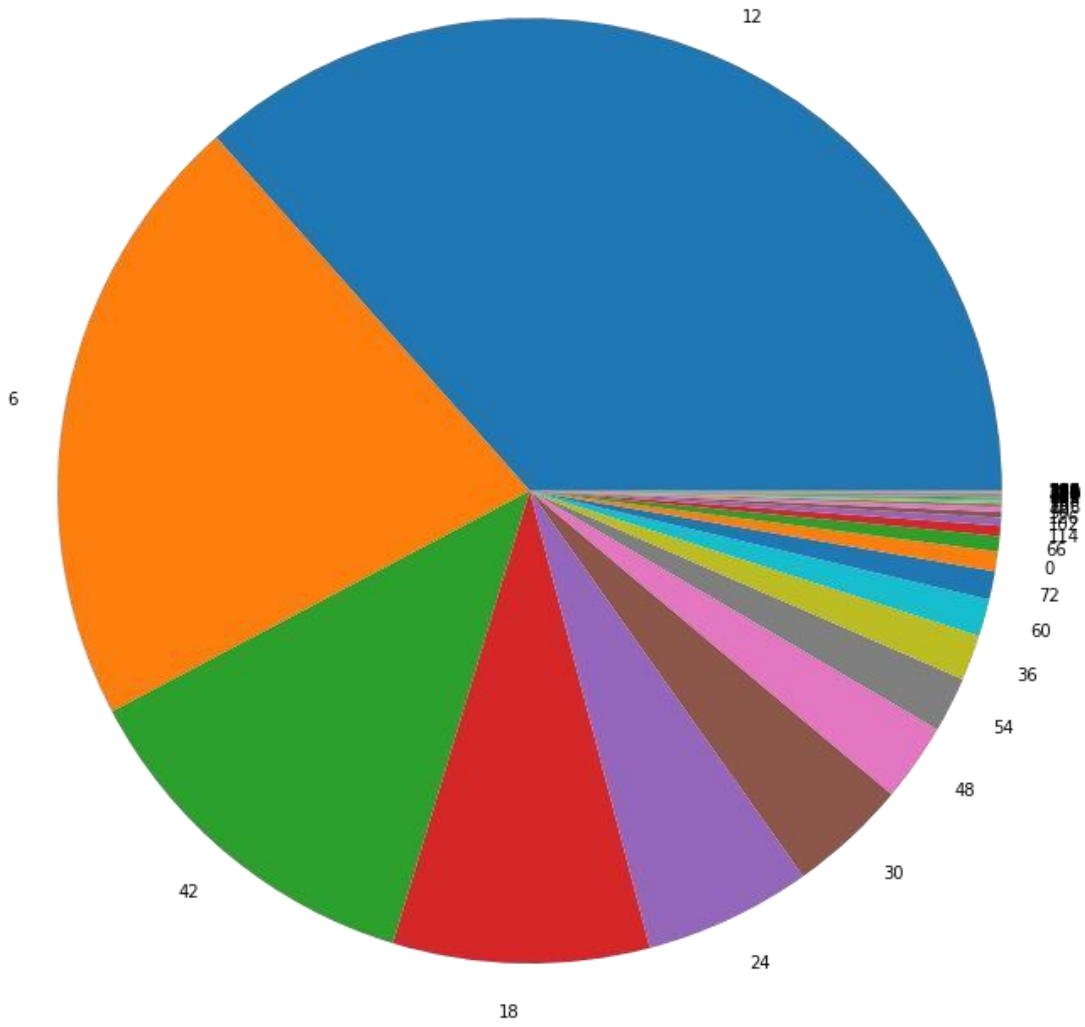
25. Customers with high value of Average payback time in days over last 90 days (payback90) are maximum Non-defaulters(who have paid there loan amount-1).

26. In between 6th and 7th month maximum customers both defaulters and Nondefaulters have paid there loan amount.

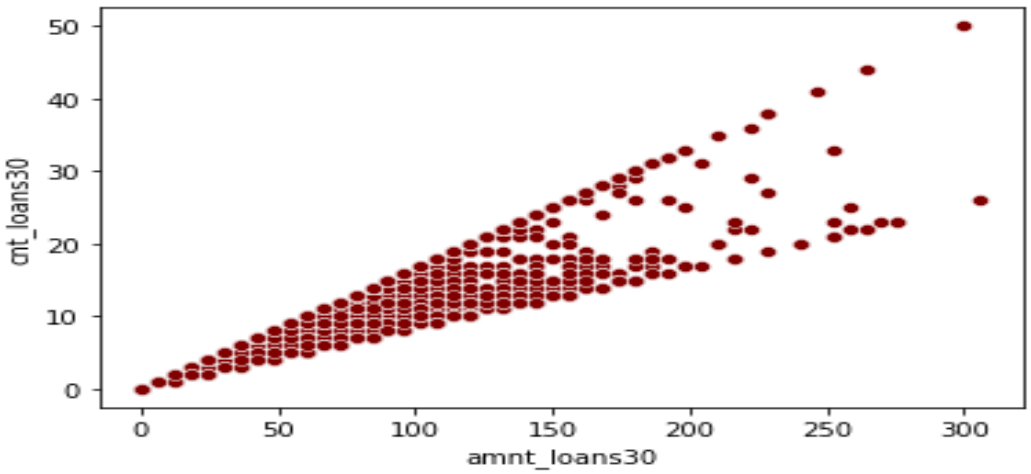
27. Below 14th of each month all the customers have paid their loan amount.



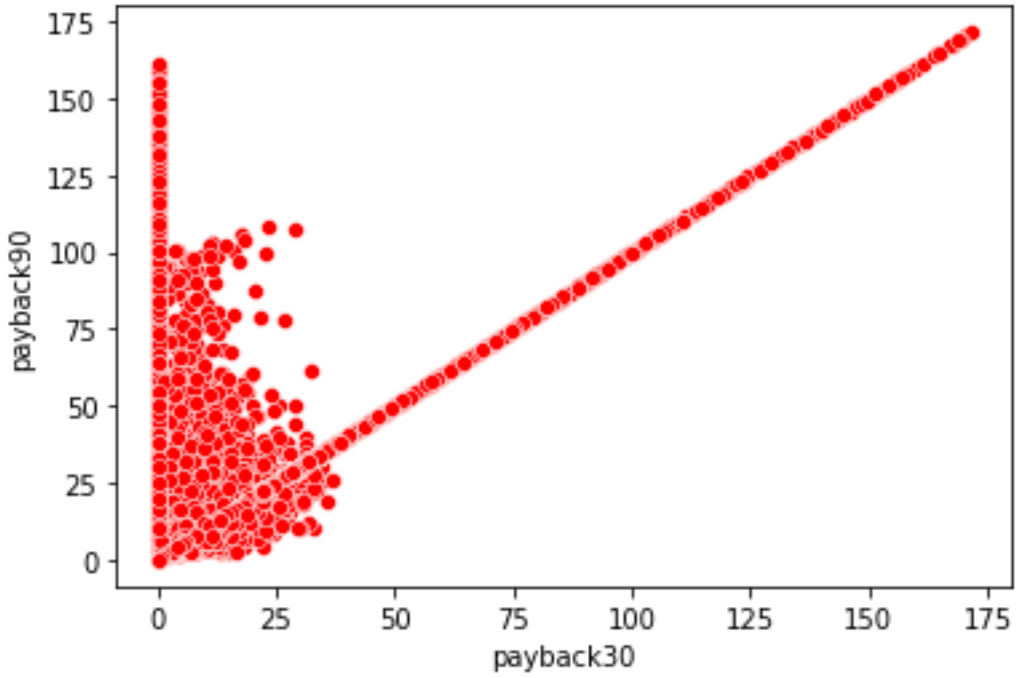
In the pie chart cnt_loans90, we observe that 2.0 has maximum count as compare to others.



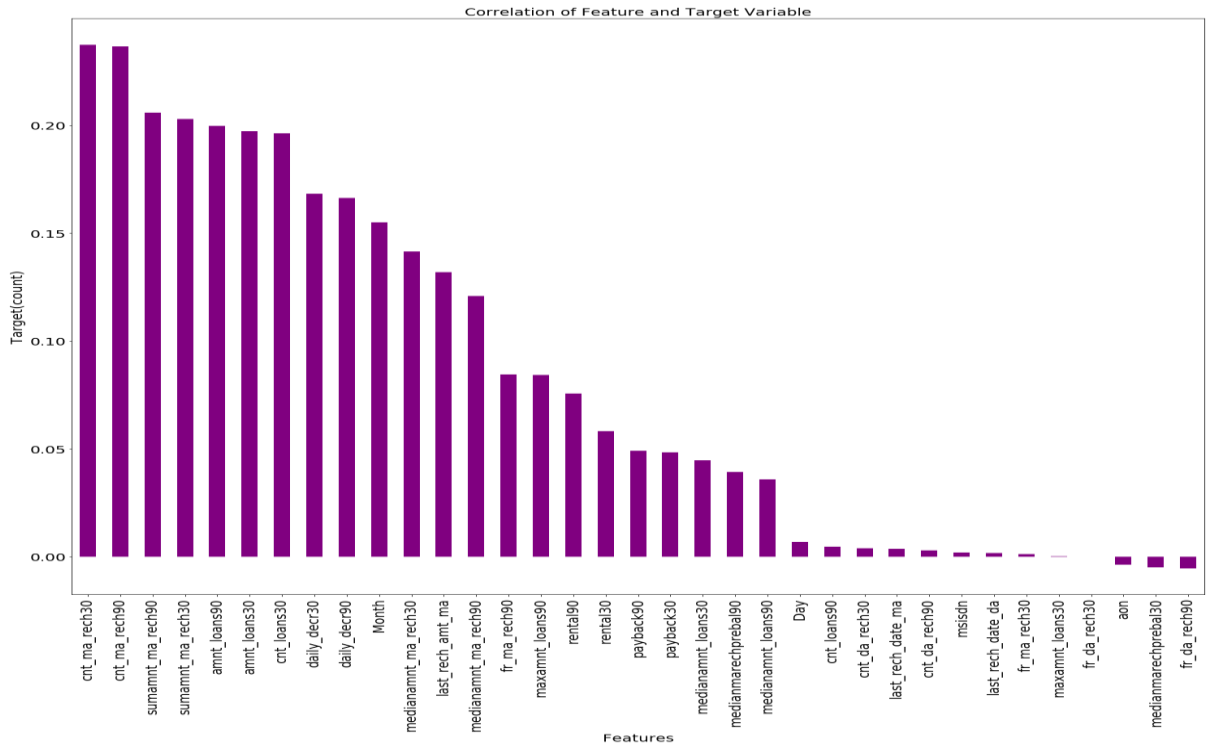
Total amount of loans taken by user in last 30 days count 12 has maximum than any other count.



When we compare the amnt_loans30 and cnt_loans30 using scatterplot we observe that amnt_loans30 is increasing exponentially.

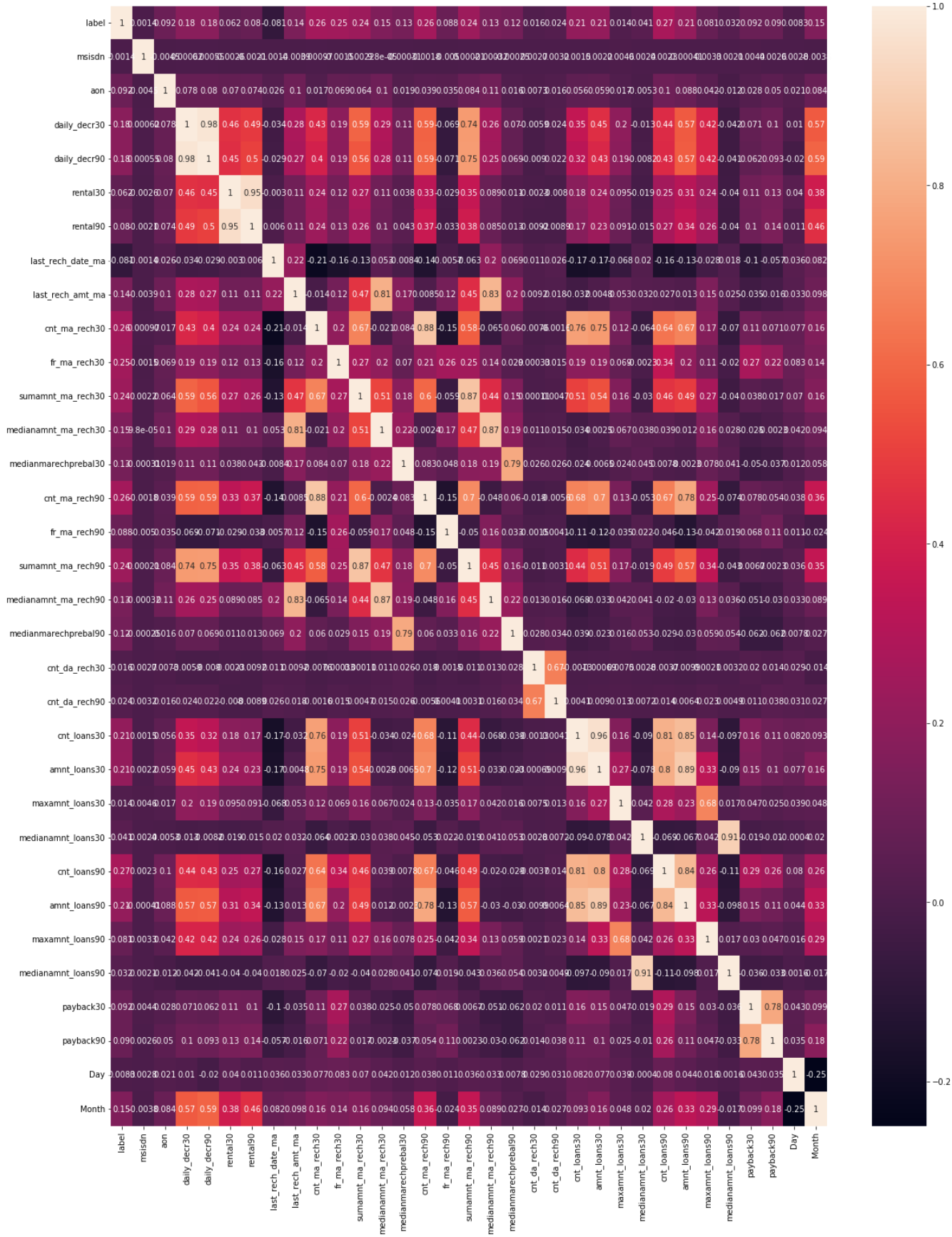


Average payback time in days over last 30 days count is increasing from 0 to 175.



Checked the correlation of data for features and target.

Correlation of data:



Run and Evaluate selected models

Logistic Regression

```
lg=LogisticRegression()
lg.fit(x_train,y_train)
```

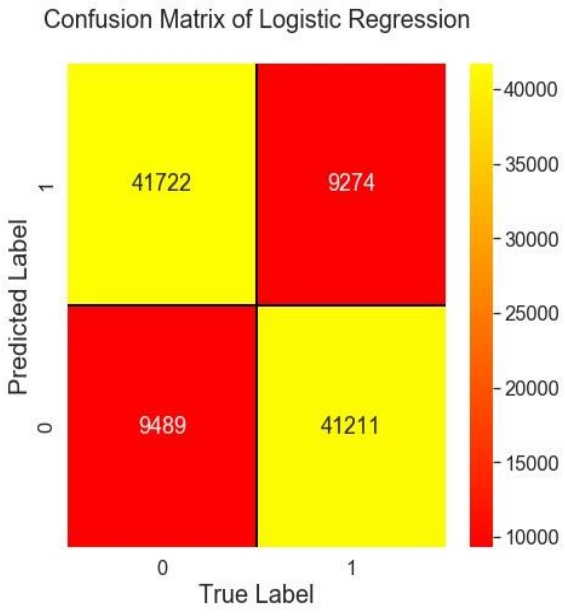
```
LogisticRegression()
```

```
lg_pred=lg.predict(x_test)
print("Predicted value:\n",lg_pred)
```

```
Predicted value:
 [0 0 1 ... 1 0 0]
```

```
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
print("Accuracy Score:",accuracy_score(y_test,lg_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,lg_pred),'\n')
print("Classification Report:\n",classification_report(y_test,lg_pred))
```

```
Accuracy Score: 0.8154991346758967

Confusion Matrix:
 [[41722  9274]
 [ 9489 41211]]

Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.82      0.82     50996
           1       0.82      0.81      0.81     50700

    accuracy                           0.82    101696
   macro avg       0.82      0.82      0.82    101696
weighted avg       0.82      0.82      0.82    101696
```



We got the 81% accuracy in Logistic Regression.

Decision Tree Classifier:

```
dr=DecisionTreeClassifier()
dr.fit(x_train,y_train)
```

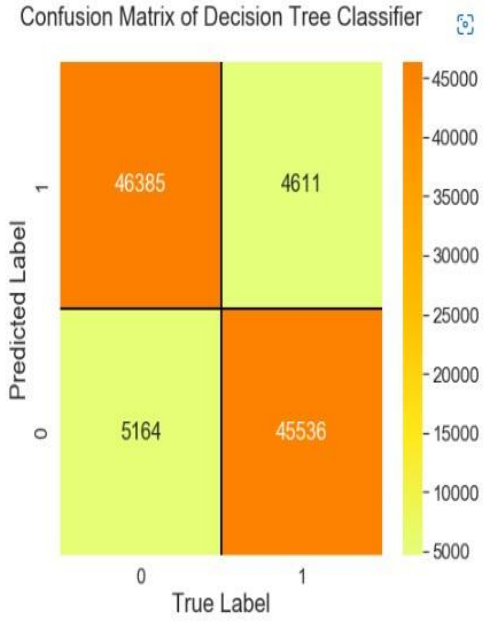
```
DecisionTreeClassifier()
```

```
dr_pred=dr.predict(x_test)
print("Predicted value:\n",dr_pred)
```

```
Predicted value:
 [0 0 1 ... 1 0 1]
```

```
print("Accuracy Score:",accuracy_score(y_test,dr_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,dr_pred),'\n')
print("Classification Report:\n",classification_report(y_test,dr_pred))
```

```
Accuracy Score: 0.9038801919446192

Confusion Matrix:
 [[46385  4611]
 [ 5164 45536]]

Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.91      0.90     50996
           1       0.91      0.90      0.90     50700

    accuracy                           0.90    101696
   macro avg       0.90      0.90      0.90    101696
weighted avg       0.90      0.90      0.90    101696
```



In Decision Tree Classifier, we got 90% accuracy.

Random Forest Classifier:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
RandomForestClassifier()
```

```
rfc_pred=rfc.predict(x_test)
print("Predicted value:\n",rfc_pred)
```

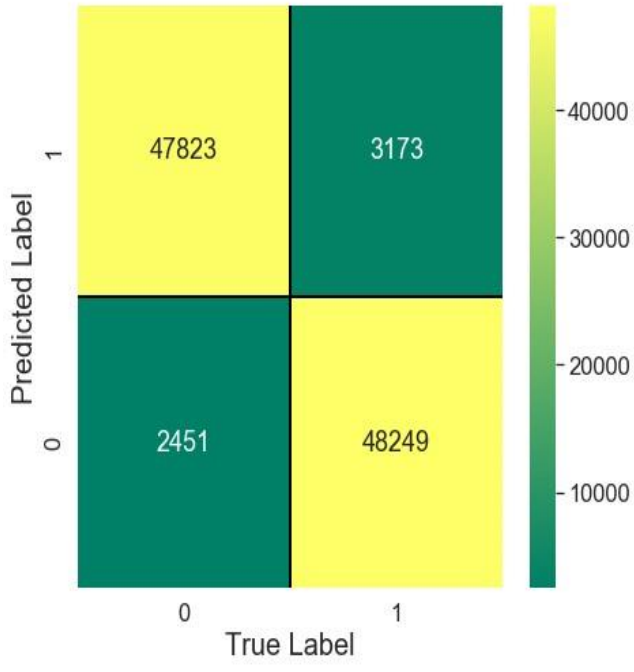
```
Predicted value:
 [0 0 1 ... 1 0 1]
```

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

```
print("Accuracy Score:",accuracy_score(y_test,rfc_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,rfc_pred),'\n')
print("Classification Report:\n",classification_report(y_test,rfc_pred))
```

```
Accuracy Score: 0.9446979232221523

Confusion Matrix:
 [[47823  3173]
 [ 2451 48249]]

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.94      0.94     50996
           1       0.94      0.95      0.94     50700

    accuracy                           0.94    101696
   macro avg       0.94      0.94      0.94    101696
```

Confusion Matrix of Random Forest Classifier



In Random Forest Classifier, we got 94% Accuracy.

Gradient Boosting Classifier:

```
gbc=GradientBoostingClassifier()
gbc.fit(x_train,y_train)
```

```
GradientBoostingClassifier()
```

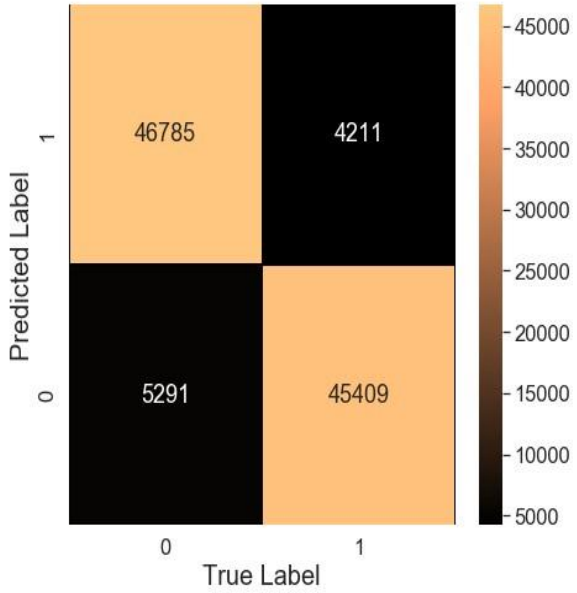
```
gbc_pred=gbc.predict(x_test)
print("Predicted value:\n",gbc_pred)
```

```
Predicted value:
 [0 0 1 ... 1 0 1]
```

```
print("Accuracy Score:",accuracy_score(y_test,gbc_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,gbc_pred),'\n')
print("Classification Report:\n",classification_report(y_test,gbc_pred))
```

```
Accuracy Score: 0.906564663310258

Confusion Matrix:
 [[46785  4211]
 [ 5291 45409]]

Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.92      0.91     50996
           1       0.92      0.90      0.91     50700

    accuracy                           0.91    101696
   macro avg       0.91      0.91      0.91    101696
weighted avg       0.91      0.91      0.91    101696
```

Confusion Matrix of Gradient Boosting Classifier



In Gradient Boosting Classifier, we got 90% accuracy.

SGD Classifier:

```
sgd=SGDClassifier()
sgd.fit(x_train,y_train)
```

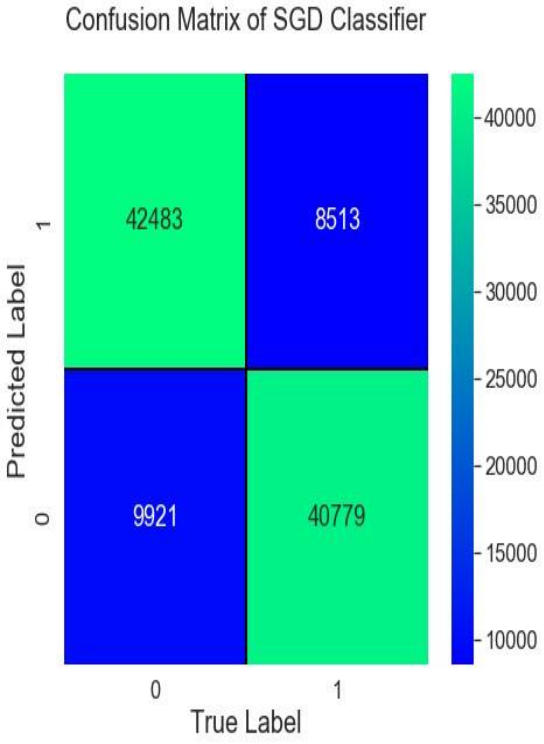
```
SGDClassifier()
```

```
sgd_pred=sgd.predict(x_test)
print("Predicted value:\n",sgd_pred)
```

```
Predicted value:
 [0 0 1 ... 1 0 0]
```

```
print("Accuracy Score:",accuracy_score(y_test,sgd_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,sgd_pred),'\n')
print("Classification Report:\n",classification_report(y_test,sgd_pred))
```

```
Accuracy Score: 0.8187342668344871

Confusion Matrix:
 [[42483  8513]
 [ 9921 40779]]

Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.83      0.82     50996
           1       0.83      0.80      0.82     50700

    accuracy                           0.82    101696
   macro avg       0.82      0.82      0.82    101696
weighted avg       0.82      0.82      0.82    101696
```



In SGD Classifier, we got 81% accuracy.

KNeighbors Classifier:

```
knn=KNeighborsClassifier()
knn.fit(x_train,y_train)
```

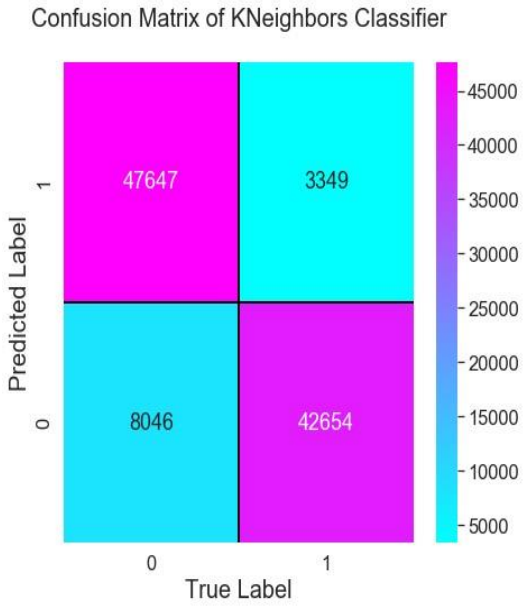
```
KNeighborsClassifier()
```

```
knn_pred=knn.predict(x_test)
print("Predicted value:\n",knn_pred)
```

```
Predicted value:
 [0 0 1 ... 0 0 1]
```

```
print("Accuracy Score:",accuracy_score(y_test,knn_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,knn_pred),'\n')
print("Classification Report:\n",classification_report(y_test,knn_pred))
```

```
Accuracy Score: 0.8879503618628068

Confusion Matrix:
 [[47647  3349]
 [ 8046 42654]]

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.93      0.89     50996
           1       0.93      0.84      0.88     50700

    accuracy                           0.89    101696
   macro avg       0.89      0.89      0.89    101696
weighted avg       0.89      0.89      0.89    101696
```



In KNeighbors Classifier, we got 88% accuracy.

Cross Validation Score

```
print("Cross Validation score for Logistic Regression:",cross_val_score(lg,x,y,cv=5).mean()*100)
print("Cross Validation score for Random Forest Classifier:",cross_val_score(rfc,x,y,cv=5).mean()*100)
print("Cross Validation score for Decision Tree Classifier:",cross_val_score(dr,x,y,cv=5).mean()*100)
print("Cross Validation score for Gradient Bossting Classifier:",cross_val_score(gbc,x,y,cv=5).mean()*100)
print("Cross Validation score for SGD Classifier:",cross_val_score(sgd,x,y,cv=5).mean()*100)
print("Cross Validation score for KNeighbors Classifier:",cross_val_score(knn,x,y,cv=5).mean()*100)
```

```
Cross Validation score for Logistic Regression: 73.94730190558344
Cross Validation score for Random Forest Classifier: 93.79358427578683
Cross Validation score for Decision Tree Classifier: 89.9671667906081
Cross Validation score for Gradient Bossting Classifier: 90.44240428476913
Cross Validation score for SGD Classifier: 61.187803599492995
Cross Validation score for KNeighbors Classifier: 86.00443723692726
```

We have done the cross-validation score for each model. We got the best score for Random Forest Classification.

Hyper Parameter Tuning:

```
from sklearn.model_selection import GridSearchCV
```

```
params={'n_estimators':[50,60],
        'criterion':['gini','entropy'],
        'max_features':['auto','log2']}
```

```
grid_search=GridSearchCV(estimator=rfc,param_grid=params,cv=3,verbose=3)
```

```
grid_search.fit(x_train,y_train)
```

```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_features': ['auto', 'log2'],
                         'n_estimators': [50, 60]},
             verbose=3)
```

```
grid_search.best_params_
```

```
{'criterion': 'gini', 'max_features': 'auto', 'n_estimators': 60}
```

```
grid_search.best_estimator_
```

```
RandomForestClassifier(n_estimators=60)
```

```
grid_search.best_score_
```

```
0.940659112603612
```

We have done the hyper parameter tuning for Random Forest Classifier, we got the best score 94% after tuning the parameter.

ROC-AUC Curve

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

```
fpr,tpr,thresholds=roc_curve(y_test,final_model_pred)
```

```
fpr
```

```
array([0.        , 0.06169111, 1.        ])
```

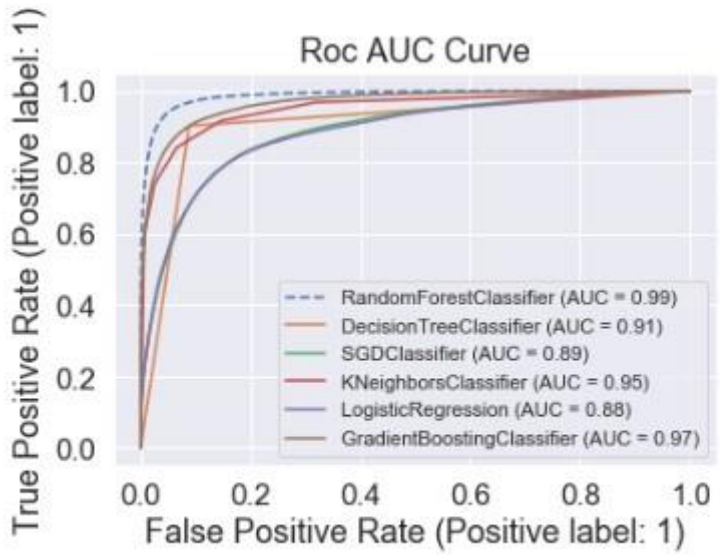
```
tpr
```

```
array([0.        , 0.95104536, 1.        ])
```

```
thresholds
```

```
array([2, 1, 0], dtype=int64)
```

```
dist=plot_roc_curve(rfc,x_test,y_test,linestyle='--')
plot_roc_curve(dr,x_test,y_test,ax=dist.ax_)
plot_roc_curve(sgd,x_test,y_test,ax=dist.ax_)
plot_roc_curve(knn,x_test,y_test,ax=dist.ax_)
plot_roc_curve(lg,x_test,y_test,ax=dist.ax_)
plot_roc_curve(gbc,x_test,y_test,ax=dist.ax_)
plt.title("Roc AUC Curve")

plt.legend(prop={'size':11},loc='lower right')
plt.show()
```



We observe that AUC Score value is high in Random Forest Classifier 99%.

Interpretation of the Results

Saving the Model:

```
import pickle
```

```
filename='micro_credit_defaulter.pickle'
```

```
pickle.dump(final_model,open(filename,'wb'))
```

```
loaded_model=pickle.load(open(filename,'rb'))
```

```
loaded_model.fit(x_train,y_train)
```

```
RandomForestClassifier(n_estimators=60)
```

```
prediction=loaded_model.predict(x_test)
print("Predicted value:",prediction)
```

```
Predicted value: [0 0 1 ... 1 0 1]
```

```
print("Accuracy Score:",accuracy_score(y_test,prediction),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,prediction),'\n')
print("Classification Report:\n",classification_report(y_test,prediction))
```

```
Accuracy Score: 0.9440390969162996


Confusion Matrix:
 [[47817  3179]
 [ 2512 48188]]

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.94      0.94     50996
           1       0.94      0.95      0.94     50700

    accuracy                           0.94    101696
   macro avg       0.94      0.94      0.94    101696
weighted avg       0.94      0.94      0.94    101696
```

```
df=pd.DataFrame([loaded_model.predict(x_test)[:],y_test[:]],index=['Predicted','Actual'])
df
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
Predicted	0	0	1	1	1	0	0	0	1	0	1	0	0	0	0	1	0	1	1	1	1	1	0	0	1	0	1	1	1	0	1	0	1	1	0	1	0	1	0
Actual	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1	1	1	1	1	0	0	1	0	1	1	1	0	1	0	1	1	0	1	0	1	0

Here are the actual and predicted values. The value almost looks similar to Actual value. We saved the final model using the pickle.

CONCLUSION

Key Findings and Conclusions of the Study

In this project report we have used various machine learning classification algorithm, we have done the step by steps analysis. We checked the correlation of features and target variables. We removed the unnecessary columns. We have done the data cleaning. We have scaled the data. We have splitted the data into 70% of data for training and 30% of data for testing. We have done the pre-processing and checked the accuracy score, confusion matrix and classification report for each model. We have done the hyper parameter tuning for the best model. We have saved the model and make the prediction for micro credit defaulter project. It is good that the predicted and actual value is almost same. This help the Micro Finance company to decide which is Defaulter and Non-Defaulter.

Learning Outcomes of the Study in respect of Data Science

The data is interesting, as it contains more feature and the volume is huge. It takes time for visualization of distribution plot of all features. I have used various visualization plot which helped me to understand what data is trying to say. It also provides the description of feature. So, it helps to understand each feature. Data cleaning is a very important steps in any machine learning project. It helps to clean the day, remove the outliers, skewness. Dropped the unnecessary columns which having the more zero values. It helps to avoid the multi-collinearity issues. Correlation helps me to understand which features is highly correlated and which is negatively correlated. I have used the 6-machine learning algorithm to make the prediction for Micro Credit Defaulter project.

Limitations of this work and Scope for Future Work

- In this data set, first drawback is the data is huge and it is difficult to handle. Because of huge data set, it takes lots of time for Visualization, training the model and in hyper parameter tuning.
- The data set contains lots of outliers and skewness present in the data set. So, in the outlier's removal and skewness removal we lost the data 7.4%. After cleaning the data, we got the 94% accuracy score, which is a very good score.
- There are lots of classification algorithm, we have chosen few machines learning algorithm to make the prediction.
- This is an early stage for making the prediction for Micro Credit Defaulter. The Finance company will make the use of this prediction and yield the high return.