A

PROJECT REPORT

ON

Micro Credit Defaulter

Submitted by:

Deepesh Singh

ACKNOWLEDGMENT

On the very outset of this report, I would like to extend my sincere & heartfelt obligation towards all the personages who have helped me in this endeavour. Without their active guidance, help, cooperation & encouragement, I would not have made headway in the project.

I am ineffably indebted to my SME for their conscientious guidance and encouragement to accomplish this assignment.

I am extremely thankful and pay my gratitude to my SME for his valuable guidance and support on completion of this project.

I extend my gratitude to FLIP ROBO Technologies for giving me this opportunity.

Any omission in this brief acknowledgement does not mean lack of gratitude.

Thanking you

DEEPESH SINGH

INTRODUCTION

- ## Business Problem Micro Finance

  Microfinance means providing very poor families with very small loans (micro credit) to help them engage in productive activities/small businesses. Over time, microfinance has come to include a broader range of services (credit, savings, insurance, etc.) as we have come to realize that the poor and the very poor who lack access to traditional formal financial institutions require a variety of financial products. The range of activities undertaken in microfinance include group lending, individual lending, the provision of savings and insurance, capacity building, and agricultural business development services. Whatever the form of activity however, the overarching goal that unifies all actors in the provision of microfinance is the creation of social value.

- ## Conceptual Background of the Domain Problem

  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.

- ## Review of Literature

  Microfinance is defined as any activity that includes the provision of financial services such as credit, savings, and insurance to low income which fall just above the national defined poverty line and poor individuals which fall below that poverty line, with the goal of creating social value.

- **Motivation for the Problem Undertaken**

  Motivation behind to making this project is that all the poor people and those families comes from middle class get the microfinance (credit) easily and checking who pay the loan amount within the 5 days of the credit and who may not pay the amount. In between the date of amount taken and the maximum tenure of the loan payment, who pays in between that comes in non-defaulter and who pays after 5 days they comes in defaulter. By making this project we can easily predict who is defaulter and non-defaulter.

Analytical Problem of Micro Finance

- **Mathematical/ Analytical Modelling of the Problem**

  Its classification problem and I use some libraries like Pandas, NumPy, seaborn, matplotlib, for building the project and done EDA with the help of these libraries. And use correlation matrix through heatmap to show the correlation between the each of the columns.

- **Data Sources and their formats**

  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)

DATA 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 Indonesian Rupiah)

medianamnt_ma_rech90: Median of amount of recharges done in main account over last 90 days at user level (in Indonesian Rupiah)

medianmarechprebal90 Median of main account balance just before recharge in last 90 days at user level (in Indonesian 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

  In this first I drop the unusual columns which is not necessary for data building and after that plot the count-plot of the label which showing the how many defaulter and non-defaulter present in the data. And after that in data cleaning firstly check the skewness in the data and remove it by the power transform and then checking for the outliers and removed the outliers by the Zscore.

- ## Hardware and Software Requirements and Tools Used

  In Hardware I use the laptop with the i5 processor and of 8 GB of ram. And in software I use an anaconda and in anaconda jupyter notebook software is used. In jupyter notebook I use python for making my project and libraries used for making the project is Pandas, NumPy, seaborn, matplotlib. Through pandas I loaded the dataset into the python.

Model/s Development and Evaluation

- **Testing of Identified Approaches (Algorithms)**

I use the Train-Test split for the training and the testing of the data and uses the Logistic Regression for finding the best random state. Attached the snap of the coding below:

Finding the best random_state

In [38]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
```

In [39]:
```
maxAccu=0
maxRS=0
for i in range(1,200):
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30,random_state=i)
    LR=LogisticRegression()
    LR.fit(x_train,y_train)
    predrf=LR.predict(x_test)
    acc=accuracy_score(y_test,predrf)
    if acc>maxAccu:
        maxAccu=acc
        maxRS=i
print('best accuracy is',maxAccu,'on Random_state',maxRS)
```

```
best accuracy is 0.8856515792487039 on Random_state 66
```

we found the best random_state as 66 now we will create our train test split using the best random_state (66)

Creating train test split

In [40]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30,random_state =66)
```

- **Run and Evaluate selected models**

  I use the five Algorithms for building the model. Snapshot of all those are as follows:

Model Building

In [41]:
```
LR=LogisticRegression()
LR.fit(x_train,y_train)
predlr=LR.predict(x_test)
print('accuracy score:',)
print(accuracy_score(y_test,predlr))
print(confusion_matrix(y_test,predlr))
print(classification_report(y_test,predlr))
```

```
accuracy score:
0.8856515792487039
[[  941  6774]
 [  416 54747]]
              precision    recall  f1-score   support

           0       0.69      0.12      0.21      7715
           1       0.89      0.99      0.94     55163

    accuracy                           0.89     62878
   macro avg       0.79      0.56      0.57     62878
weighted avg       0.87      0.89      0.85     62878
```

In [42]:
```
DTC=DecisionTreeClassifier()
DTC.fit(x_train,y_train)
preddt=DTC.predict(x_test)
print('accuracy score:',)
print(accuracy_score(y_test,preddt))
print(confusion_matrix(y_test,preddt))
print(classification_report(y_test,preddt))
```

```
accuracy score:
0.8652628900410319
[[ 3884  3831]
 [ 4641 50522]]
              precision    recall  f1-score   support

           0       0.46      0.50      0.48      7715
           1       0.93      0.92      0.92     55163

    accuracy                           0.87     62878
   macro avg       0.69      0.71      0.70     62878
weighted avg       0.87      0.87      0.87     62878
```

In [43]:
```
GNB=GaussianNB()
GNB.fit(x_train,y_train)
predgnb=GNB.predict(x_test)
print('accuracy score:',)
print(accuracy_score(y_test,predgnb))
print(confusion_matrix(y_test,predgnb))
print(classification_report(y_test,predgnb))
```

```
accuracy score:
0.7365056140462483
[[ 5871  1844]
 [14724 40439]]
              precision    recall  f1-score   support

           0       0.29      0.76      0.41      7715
           1       0.96      0.73      0.83     55163

    accuracy                           0.74     62878
   macro avg       0.62      0.75      0.62     62878
weighted avg       0.87      0.74      0.78     62878
```

In [44]:
```
RF=RandomForestClassifier()
RF.fit(x_train,y_train)
predrf=RF.predict(x_test)
print('accuracy score:',)
print(accuracy_score(y_test,predrf))
print(confusion_matrix(y_test,predrf))
print(classification_report(y_test,predrf))
```

```
accuracy score:
0.912751677852349
[[ 3414  4301]
 [ 1185 53978]]
              precision    recall  f1-score   support

           0       0.74      0.44      0.55      7715
           1       0.93      0.98      0.95     55163

    accuracy                           0.91     62878
   macro avg       0.83      0.71      0.75     62878
weighted avg       0.90      0.91      0.90     62878
```

In [45]:
```
AD=AdaBoostClassifier()
AD.fit(x_train,y_train)
predad=AD.predict(x_test)
print('accuracy score:',)
print(accuracy_score(y_test,predad))
print(confusion_matrix(y_test,predad))
print(classification_report(y_test,predad))
```

```
accuracy score:
0.904942905308693
[[ 2308  5415]
 [  562 54601]]
              precision    recall  f1-score   support

           0       0.80      0.30      0.43      7715
           1       0.91      0.99      0.95     55163

    accuracy                           0.90     62878
   macro avg       0.86      0.64      0.69     62878
weighted avg       0.90      0.90      0.89     62878
```

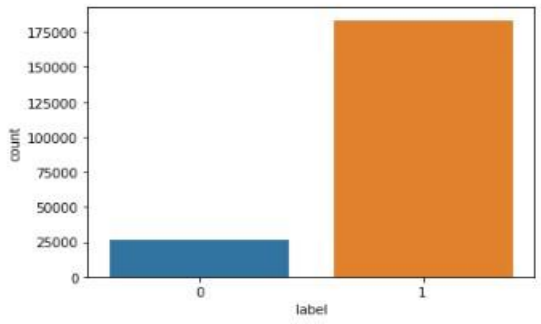
from above model building we can see the randomforestclassifier has the highest accuracy of 91.24% but it is due to of overfitting/underfitting so we go for cross validation score

- **Visualizations**

  Plot the count-plot

```
In [17]: sns.countplot(data['label'])
```

```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x28a03396a30>
```



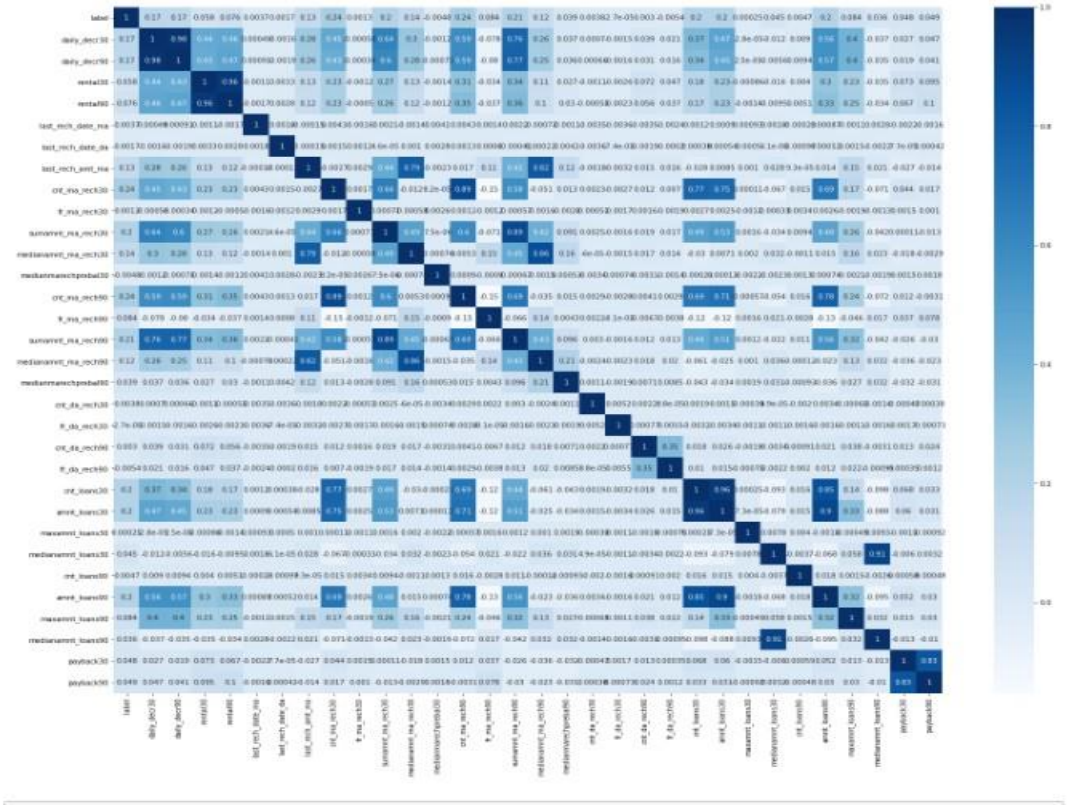
we can see there is large no. success rate in payback within 5 days

In this count plot we can easily see the large no. of success rate of payback within 5 days.

Plotting the heatmap of the correlation matrix of the data:

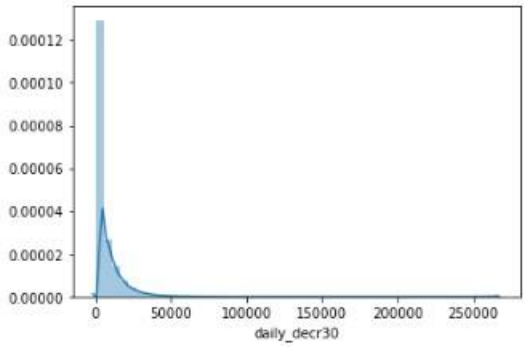
```
In [19]: fig=plt.figure(figsize=(25,20))
         hc=data.corr(method='pearson')
         sns.heatmap(hc,annot=True,cmap='Blues')
```

```
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x28a035c6eb0>
```

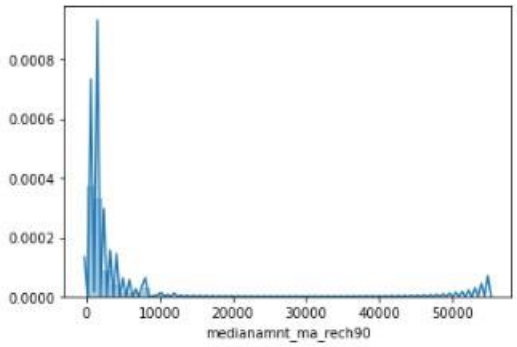


Plotting the distplot of the columns daily_decr30 and medianamnt_ma_rech90

```
In [20]: sns.distplot(data['daily_decr30'])
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x28a0a718670>
```

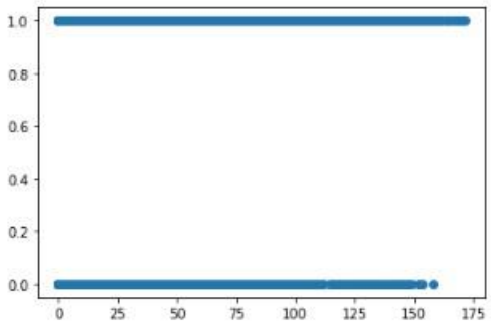



```
In [21]: sns.distplot(data['medianamnt_ma_rech90'])
Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x28a0a78c970>
```



Plotting the scatter plot of the payback90:

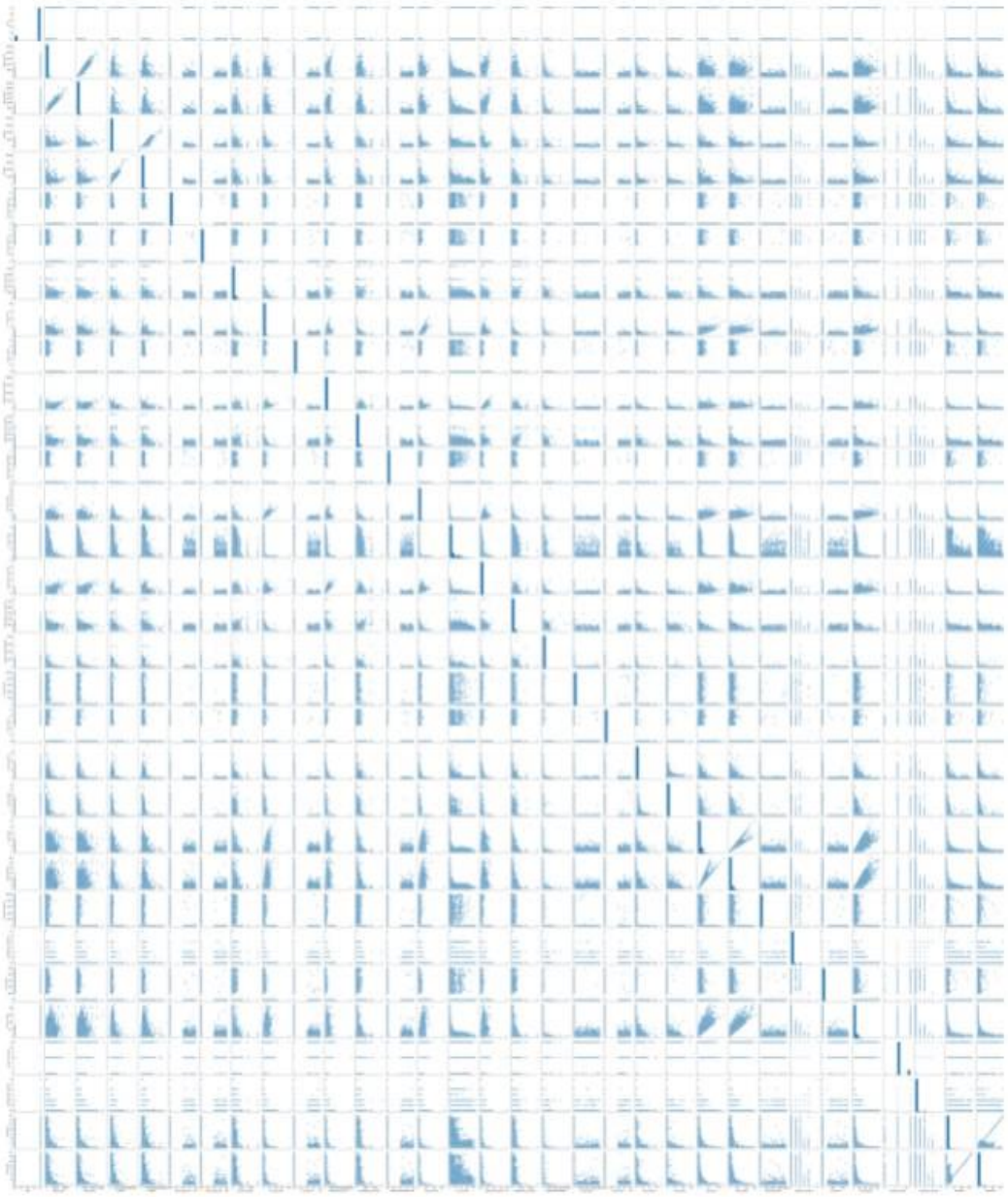
```
In [22]: plt.scatter(data['payback90'],data['label'])
Out[22]: <matplotlib.collections.PathCollection at 0x28a09802f10>
```



scatter plot of showing Average payback time in days over last 90 days with the user paid back the credit amount within 5 days of issuing the loan.

Plotting the pair-plot of the all the data:

```
In [23]: sns.pairplot(data)
Out[23]: <seaborn.axisgrid.PairGrid at 0x28a0985b0d0>
```



In the pair-plot of the data we can see all the relation between the columns.

CONCLUSION

- ## Key Findings and Conclusions of the Study

  I find the highest accuracy score of 91% with the random Forest classification after the EDA and data cleaning and the model building of the data and done a cross validation check in all the 5 algorithms and go with the best model for the hyper parameter tuning and predict the payback with in the 5 days and after 5 days i.e. defaulter and non-defaulter.

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

  In data cleaning I firstly check the skewness of the data and removed it through the power transform and after that I checked for the outliers and there is lots of outliers are present in the data. Then I removed those outliers by using Z-score. After that I find the best random state and the create a train-test-split. And after that using various algorithms and check for the accuracy score, confusion matrix and classification report of the all the algorithm. And from all the algorithm Random forest classification works best.

  In hyperparameter tuning I face the problem for finding the best parameter and running the code after the guidance of my SME I resolve the problem and completed my project.

  The study showed that number of dependents, type of loan, adequacy of loan, duration for repayment of loan, period within the year the loan was acquired and how the customers rank the interest

charged on the loan were significant determinants of micro credit default. Based on the findings of this study.

- **Limitations of this work and Scope for future work**

In this project we only find the defaulter and non-defaulter of the micro credit finance with the help of model building by different algorithms.

The Microfinance institutions should adopt the group loan policy as the main mode through which microcredit may be issued to suitable applicants. Considering the current value of the Ghana credit relative to the exchange rates and the economy as a whole, the MFT'S should consider increasing the size of loan amounts.

The MFT'S should give out long term loan.