FLIP ROBO

NAME OF THE PROJECT

MICRO-CREDIT-PROJECT

Submitted by:

TWINKLE PATEL

ACKNOWLEDGMENT

https://elitedatascience.com/imbalanced-classes

From this site i was able to understand the concept of imbalanced classes and how to use that to my project

INTRODUCTION

- **Business Problem Framing**

Here in this project, we have a dataset of the telecom industry. They have several plans for users. This project was a highly inspired project as it includes the real-time problem for Microfinance Institution (MFI), and to the poor families in remote areas with low income, MFI to give micro-credit on mobile balances to paid back in 5 days.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,now with respect to telecom industry they payback period has been set up as 5 days,if a user fails to pay the loan then he is a defaulter.In real life scenario ,this problem is a logical approach to handle the classification between a defaulter and non defaulter,this problem becomes more special and unique because it’s just not simply a credit loan problem,but it is more associated with telecom industry and the loan here is for communication purpose(the mobile data packages),So surely this is a smart move in this era wherein communication plays a vital role .

- **Conceptual Background of the Domain Problem**

Generally,Credit Scores play a vital role for loan approvals and are very important in today’s financial analysis for an individual.So, the main domain here is the financial domain as the main focus is credit (loan taken for mobile plans). Also this data belongs to the telecom industry,So their plans and usage frequency of recharge done by the user and all such features are included in the dataset,so some important knowledge regarding telecom sector is also required.

- **Review of Literature**

Analysing the dataset,

- Shape of dataset is as follows (209593, 36)
- Almost all are numerical values (int and float)
- No null values are present
- Target variable is label
- Ratio among them is- counts for label 1 is 183431 and counts for 0 label is 26162(ratio 87.5% and 12.5%)

So, I have to solve a classification problem which is imbalanced and has the presence of outliers. Almost all the columns apart from the circle and phone number are of importance.

- **Motivation for the Problem Undertaken**

.

This project was a highly motivated project as it includes the real time problem for Microfinance Institution (MFI), and to the poor families in remote areas with low income, and it is related to financial sectors, as I believe that with growing technologies.

Analytical Problem Framing

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

This problem is a classification problem, the target variable is itself a statistical parameter.we have 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 loan insurance. In this case, Label '1' indicates that the loan has been paid i.e. Non- defaulter, while, Label '0' indicates that the loan has not been paid. For a loan amount of 5 payback amount should be 6, and for loan amount of 10 payback amount is 12.

- **Data Sources and their formats**

The source of the data is from a telecom industry having 36 columns and 209593 rows,the data was in a excel file for which i saved a csv copy and uploaded that on my

Data Sources and their formats

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 Indian Rupee)
medianamnt_ma_rech90: Median of amount of recharges done in main account over last 90 days at user level (in Indian Rupee)
medianmarechprebal90: Median of main account balance just before recharge in last 90 days at user level (in Indian Rupee)
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

jupyter notebook,

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209593 entries, 0 to 209592
Data columns (total 36 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   label                 209593 non-null  int64  
 1   msisdn                209593 non-null  object 
 2   aon                   209593 non-null  float64
 3   daily_decr30          209593 non-null  float64
 4   daily_decr90          209593 non-null  float64
 5   rental30              209593 non-null  float64
 6   rental90              209593 non-null  float64
 7   last_rech_date_ma     209593 non-null  float64
 8   last_rech_date_da     209593 non-null  float64
 9   last_rech_amt_ma      209593 non-null  int64  
 10  cnt_ma_rech30         209593 non-null  int64  
 11  fr_ma_rech30          209593 non-null  float64
 12  sumamnt_ma_rech30     209593 non-null  float64
 13  medianamnt_ma_rech30  209593 non-null  float64
 14  medianmarechprebal30  209593 non-null  float64
 15  cnt_ma_rech90         209593 non-null  int64  
 16  fr_ma_rech90          209593 non-null  int64  
 17  sumamnt_ma_rech90     209593 non-null  int64  
 18  medianamnt_ma_rech90  209593 non-null  float64
 19  medianmarechprebal90  209593 non-null  float64
 20  cnt_da_rech30         209593 non-null  float64
 21  fr_da_rech30          209593 non-null  float64
 22  cnt_da_rech90         209593 non-null  int64  
 23  fr_da_rech90          209593 non-null  int64  
 24  cnt_loans30           209593 non-null  int64  
 25  amnt_loans30          209593 non-null  int64  
 26  maxamnt_loans30       209593 non-null  float64
 27  medianamnt_loans30    209593 non-null  float64
 28  cnt_loans90           209593 non-null  float64
 29  amnt_loans90          209593 non-null  int64  
 30  maxamnt_loans90       209593 non-null  int64  
 31  medianamnt_loans90    209593 non-null  float64
 32  payback30             209593 non-null  float64
 33  payback90             209593 non-null  float64
 34  pcircle               209593 non-null  object 
 35  pdate                 209593 non-null  object 
dtypes: float64(21), int64(12), object(3)
memory usage: 57.6+ MB
```

- **Data Preprocessing Done**

- We created multiple groups based on min, 25% to 75%, above 75% and we compared it VS payback within 5 days.
- I identified the outliers for features whose Z-score>5, and then did mean imputing and also applied cube root to bring the data closer to distribution.
- I checked the correlation of the independent and dependent features and dropped the negative and less important features with the help of correlation matrix. .
- Applied StandardScaler to our dependent features.

	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cn
count	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	2(
mean	0.875177	8112.343445	5381.402289	6082.515068	2692.581910	3483.406534	3755.847800	3712.202921	2064.452797	
std	0.330519	75696.082531	9220.623400	10918.812767	4308.586781	5770.461279	53905.892230	53374.833430	2370.786034	
min	0.000000	-48.000000	-93.012667	-93.012667	-23737.140000	-24720.580000	-29.000000	-29.000000	0.000000	
25%	1.000000	246.000000	42.440000	42.692000	280.420000	300.260000	1.000000	0.000000	770.000000	
50%	1.000000	527.000000	1469.175667	1500.000000	1083.570000	1334.000000	3.000000	0.000000	1539.000000	
75%	1.000000	982.000000	7244.000000	7802.790000	3356.940000	4201.790000	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	

Data Inputs- Logic- Output Relationships

```
In [34]:
# Checking the label corelation with other features
df.corr()['label'].sort_values()
```

```
Out[34]: fr_da_rech90            -0.005418
         medianmarechprebal30    -0.004829
         aon                     -0.003785
         fr_da_rech30            -0.000027
         maxamnt_loans30          0.000248
         fr_ma_rech30             0.001330
         last_rech_date_da        0.001711
         cnt_da_rech90            0.002999
         last_rech_date_ma        0.003728
         cnt_da_rech30            0.003827
         cnt_loans90              0.004733
         medianamnt_loans90       0.035747
         medianmarechprebal90     0.039300
         medianamnt_loans30       0.044589
         payback30                0.048336
         payback90                0.049183
         rental30                 0.058085
         rental90                 0.075521
         maxamnt_loans90          0.084144
         fr_ma_rech90             0.084385
         medianamnt_ma_rech90     0.120855
         last_rech_amt_ma         0.131804
         medianamnt_ma_rech30     0.141490
         daily_decr90             0.166150
         daily_decr30             0.168298
         cnt_loans30              0.196283
         amnt_loans30             0.197272
         amnt_loans90             0.199788
         sumamnt_ma_rech30        0.202828
         sumamnt_ma_rech90        0.205793
         cnt_ma_rech90            0.236392
         cnt_ma_rech30            0.237331
         label                    1.000000
         Name: label, dtype: float64
```

The table displayed that there are some features which are moderately correlated with the target variable and some have very less correlation with the target variable.

Describe the relationship behind the data input, its format, the logic in between and the output. Describe how the input affects the output.

Now my output is the label class and rest 32 columns serve as the input,as discussed this is a classification problem with two classes 1 and 0(0 is defaulter unable to pay the loan within 5 days and 1 is non defaulter that is the user had payed the loan),so now this depends on these 32 columns,comprising of recharge(usage) and loan taken and payback that is the days in which it has been paid back,and other columns are somewhat extension to these like median,frequency,counts etc.

- **Hardware and Software Requirements and Tools Used**

Hardware: 8GB RAM, 64-bit, i7 processor.

Software: Excel, Jupyter Notebook, python 3.6., google colab

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix,classification_report
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_val_predict
import warnings
warnings.filterwarnings('ignore')
```

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Describe the approaches you followed, both statistical and analytical, for solving of this problem.

- Testing of Identified Approaches (Algorithms)

  Listing down all the algorithms used for the training and testing.

Algorithms used are as below-

LR=LogisticRegression()
DT=DecisionTreeClassifier()
GNB=GaussianNB()
RFC=RandomForestClassifier()
GBC=GradientBoostingClassifier()
ABC=AdaBoostClassifier()
ETC=ExtraTreesClassifier()

```
accuracy = []
precision = []
recall = []
f1_score = []
logLoss = []
Auc_roc_score=[]
cvs=[]

def calculate_metrics(y_test, y_pred):
    acc = metrics.accuracy_score(y_true = y_test, y_pred = y_pred)
    pre = metrics.precision_score(y_true = y_test, y_pred = y_pred)
    rec = metrics.recall_score(y_true = y_test, y_pred = y_pred)
    f1 = metrics.f1_score(y_true = y_test, y_pred = y_pred)
    log_loss = metrics.log_loss(y_true = y_test, y_pred = y_pred)


    accuracy.append(acc)
    precision.append(pre)
    recall.append(rec)
    f1_score.append(f1)
    logLoss.append(log_loss)
```

- Run and Evaluate selected models
  Describe all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics.

Logistic Regression

It is the mostly used algorithm across industry ,we will import libraries such as Logistic Regression from sklearn.linear_model, importing confusion matrix and classification report we get the following results

```
from sklearn.linear_model import LogisticRegression
lg=LogisticRegression()
lg.fit(train_x,train_y)
pred=lg.predict(test_x)
print("accuracy_score:",accuracy_score(test_y,pred))
print(classification_report(test_y,pred))
```

```
accuracy_score: 0.7461184526145929
              precision    recall  f1-score   support

           0       0.74      0.75      0.75     45713
           1       0.75      0.74      0.75     46003

    accuracy                           0.75     91716
   macro avg       0.75      0.75      0.75     91716
weighted avg       0.75      0.75      0.75     91716
```

2. Decision Tree Classifier

A decision tree classifier is a tree in which internal nodes are labelled by features,We have two criterion ‘gini’ and ‘entropy’

Importing DecisionTreeClasifier from sklearn.tree

Here are the results

```python
from sklearn.tree import DecisionTreeClassifier
```

```python
dct=DecisionTreeClassifier()
dct.fit(train_x,train_y)
preddct=dct.predict(test_x)
```

```python
print(classification_report(test_y,preddct))
```

```
              precision    recall  f1-score   support

           0       0.92      1.00      0.95     45713
           1       1.00      0.91      0.95     46003

    accuracy                           0.95     91716
   macro avg       0.96      0.95      0.95     91716
weighted avg       0.96      0.95      0.95     91716
```

3 Random Forest Classifier

```python
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(train_x, train_y)
rf_predict=rf.predict(test_x)
```

```python
rf_conf_matrix = confusion_matrix(test_y, rf_predict)
rf_acc_score = accuracy_score(test_y, rf_predict)
print(rf_conf_matrix)
print(rf_acc_score)
```

```
[[45590   123]
 [ 2088 43915]]
0.9758929739631035
```

4 AdaBoostClassifier

```python
from sklearn.ensemble import AdaBoostClassifier
ad=AdaBoostClassifier(n_estimators=7)
ad.fit(train_x,train_y)
ad_pred=ad.predict(test_x)
print(accuracy_score(test_y,ad_pred))
print(confusion_matrix(test_y,ad_pred))
print(classification_report(test_y,ad_pred))


```

```
0.748037419861311
[[36405  9308]
 [13801 32202]]
              precision    recall  f1-score   support

           0       0.73      0.80      0.76     45713
           1       0.78      0.70      0.74     46003

    accuracy                           0.75     91716
   macro avg       0.75      0.75      0.75     91716
weighted avg       0.75      0.75      0.75     91716
```

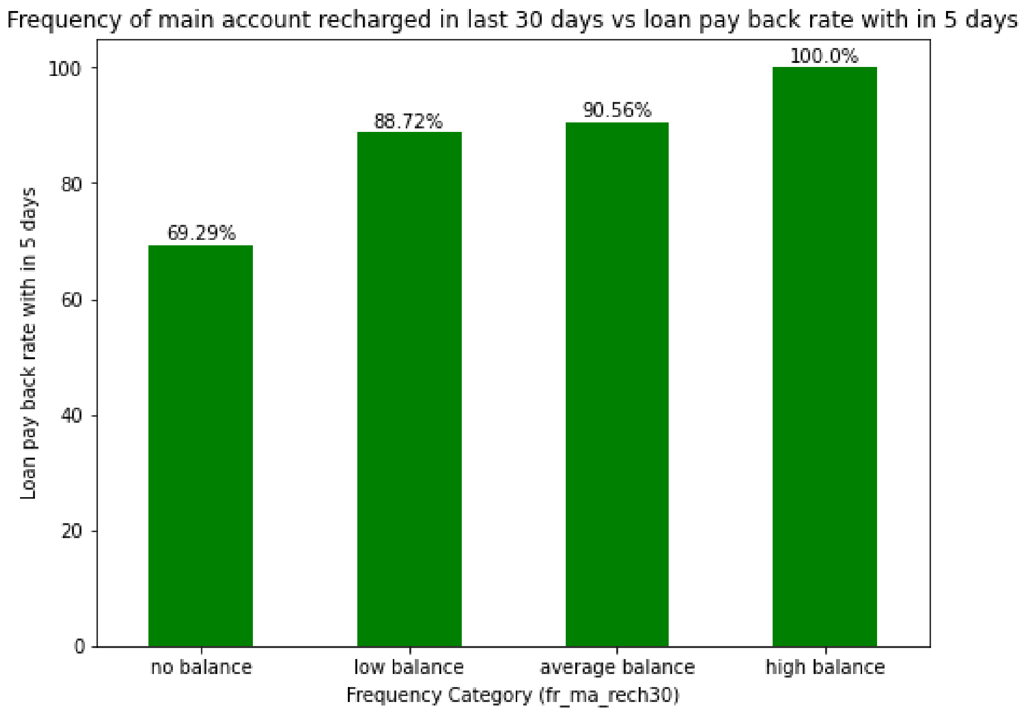
5. GaussianNB

```python
from sklearn.naive_bayes import GaussianNB
gnb=GaussianNB()
gnb.fit(train_x,train_y)
predgnb=gnb.predict(test_x)
print(accuracy_score(predgnb,test_y))
print(classification_report(test_y,predgnb))
```

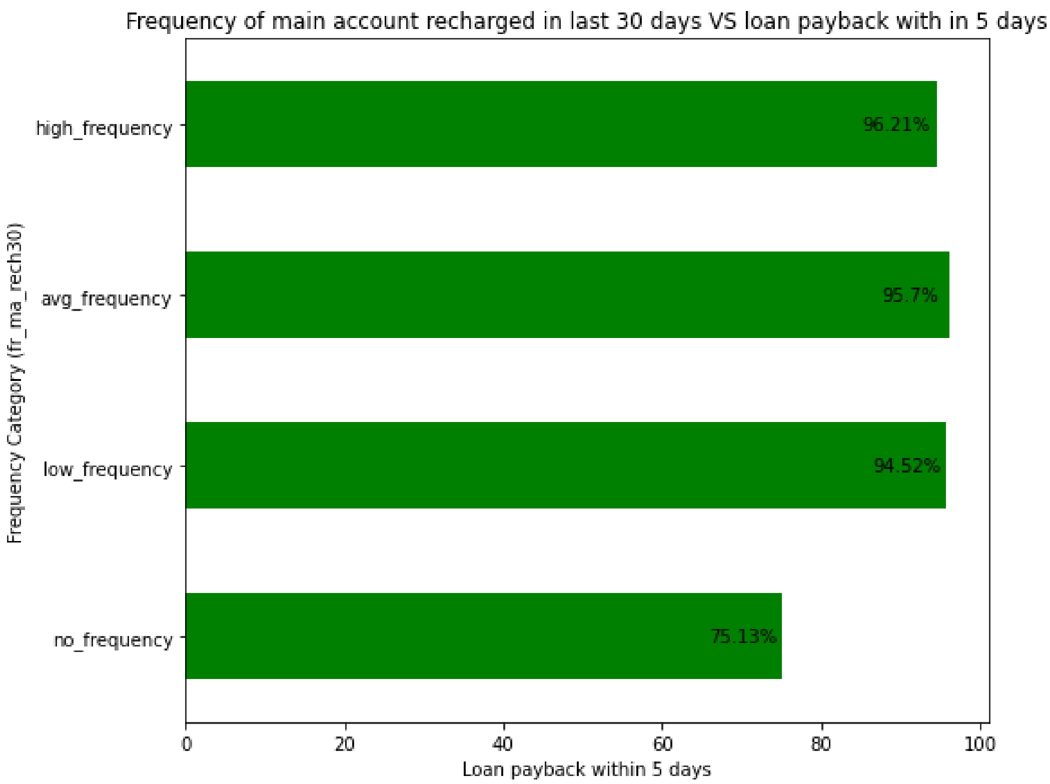
```
0.6519800252954774
              precision    recall  f1-score   support

           0       0.81      0.40      0.53     45713
           1       0.60      0.90      0.72     46003

    accuracy                           0.65     91716
   macro avg       0.70      0.65      0.63     91716
weighted avg       0.70      0.65      0.63     91716
```

- problem under consideration
- The best key metric was random forest classifier and decision tree classifier both giving a high accuracy as

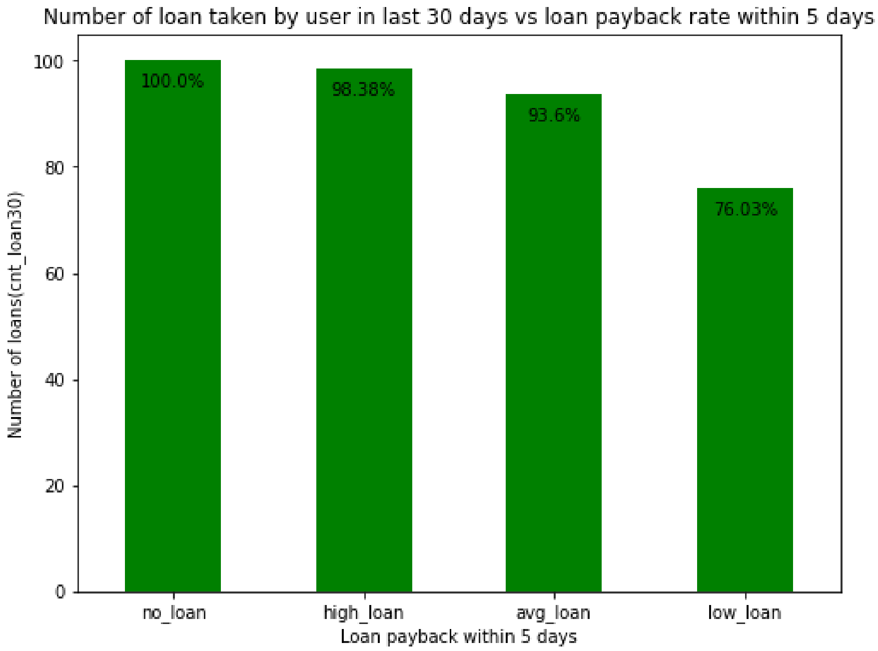
- Visualizations



From the above we can see that users with high balance always pays back the loan within 5 days and average and low category only 9% - 12% users failed to payback the loan within 5%, and users with zero balance around 30% users are not paying the loan back within 5 days.



From the above we can see there are no 100% rate in any frequency group to payback the loan within 5 days, and all average low and high frequency have at least 6% to 4% users who didn't payback the loan within 5 days. Coming to
users have no frequency 25% users didn't pay back the loan within 5 days, till now we can see that users with no balance and no frequency are costing huge losses, companies should implement some kind of strategies to reduce that like send SMS alerts for notification.



From the we can see that majority user who took high loans in last 30 days are more likely to payback within 5 days and 1.62% users failed to payback within 5 days, and among average loan user 7% users failed to pay back the loan within 5 days, and users with low loan have 24% didn't payback as expected might be defaulted.

CONCLUSION

- Key Findings and Conclusions of the Study

  By looking at the auc roc score for the upsampled model ,the roc auc score came out to be at 74 percent whereas before upsampling the auc roc score was 54. Whereas earlier accuracy was of 80 percent and after upsampling it was around 75.So precision recall f1 score,and auc roc score was the decision criteria for my model’s performance .Among all the metrics used decision tree and random forest worked really well.

- Learning Outcomes of the Study in respect of Data Science

  So from this project I learned how apart from a traditional loan problem,there are credit realted problems revolving around other sectors too like in this case it was telecom industry,this model classified the loans given to the users were paid back within 5 days or not,so this was a great move and with data science the telecom company can study the behaviour of users ,how much loan they can take and thereby identifying there potential users also providing these types of credit to all incomes groups especially poor.

  The major limitation faced was imbalance dataset which was solved by resampling and generating upsamples,for future work and rnd purposes i would like to explore more on this and the perfect ratio for two classes (because in this scenario i generated equal samples for both the classes with upsampling) for a classification problem ,so if ratio varies than how the results will differ will be another task.