FLIP ROBO

Micro-Credit Defaulter Model

Submitted by:

Roshan Kumar Verma

ACKNOWLEDGMENT

This Project would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I would like to thank Flip Robo Technologies Bangalore for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & my SME of Flip Robo Mohd. Kashif for their kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

INTRODUCTION

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.

Business Problem Framing

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 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).

Conceptual Background of the Domain Problem

The conceptual background of this domain problem that client want to know whether their users are paying the loaned amount within the time frame or not.

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.

Review of Literature

To predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of insurance of loan. In this case, Label ‘1’ indicates that the loan has been payed i.e. Non- defaulter, while, Label ‘0’ indicates that the loan has not been payed i.e. defaulter.

After observing the data first thing we observing, there are no null values in the dataset. But data set is quite imbalanced in terms of defaulter and non-defaulters

Motivation for the Problem Undertaken

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. 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.

Our main objective of doing this project is to build a model to predict whether the users are paying the loan within the due date or not.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

There are multiple mathematical and analytical analytics can be done before moving forward to the proper Exploratory Data Analysis.

Though the data is quite imbalance and many columns doesn't have that except the maximum value so we will have to drop that columns. Later checking the skewed data and trying to treat the skewed data before modeling process.

While we would be removing the unwanted data i.e. the outliers from that found that almost 30000 data has been chopped.

Though the data given by the client having almost 30 columns so I have reduced the columns using principle component analysis and then I scaled the data.

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.

This is the data snapshot which shows us the format of the data provided to us

```
In [3]: data
```

Out[3]:

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	...	maxamnt_loans30	mec
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	...	6.0	
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	...	12.0	
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	...	6.0	
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	...	6.0	
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	...	6.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
209588	209589	1	22758I85348	404.0	151.872333	151.872333	1089.19	1089.19	1.0	0.0	...	6.0	
209589	209590	1	95583I84455	1075.0	36.936000	36.936000	1728.36	1728.36	4.0	0.0	...	6.0	
209590	209591	1	28556I85350	1013.0	11843.111667	11904.350000	5861.83	8893.20	3.0	0.0	...	12.0	
209591	209592	1	59712I82733	1732.0	12488.228333	12574.370000	411.83	984.58	2.0	38.0	...	12.0	
209592	209593	1	65061I85339	1581.0	4489.362000	4534.820000	483.92	631.20	13.0	0.0	...	12.0	

209593 rows × 37 columns

There are no null values in the dataset.

```
In [5]: data.isnull().sum().sum()
Out[5]: 0
```

If we see to the Data information as an overview

```
Data columns (total 37 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Unnamed: 0            209593 non-null  int64
 1   label                 209593 non-null  int64
 2   msisdn                209593 non-null  object
 3   aon                   209593 non-null  float64
 4   daily_decr30          209593 non-null  float64
 5   daily_decr90          209593 non-null  float64
 6   rental30              209593 non-null  float64
 7   rental90              209593 non-null  float64
 8   last_rech_date_ma     209593 non-null  float64
 9   last_rech_date_da     209593 non-null  float64
 10  last_rech_amt_ma      209593 non-null  int64
 11  cnt_ma_rech30         209593 non-null  int64
 12  fr_ma_rech30          209593 non-null  float64
 13  sumamnt_ma_rech30     209593 non-null  float64
 14  medianamnt_ma_rech30  209593 non-null  float64
 15  medianmarechprebal30  209593 non-null  float64
 16  cnt_ma_rech90         209593 non-null  int64
 17  fr_ma_rech90          209593 non-null  int64
 18  sumamnt_ma_rech90     209593 non-null  int64
 19  medianamnt_ma_rech90  209593 non-null  float64
 20  medianmarechprebal90  209593 non-null  float64
 21  cnt_da_rech30         209593 non-null  float64
 22  fr_da_rech30          209593 non-null  float64
 23  cnt_da_rech90         209593 non-null  int64
 24  fr_da_rech90          209593 non-null  int64
 25  cnt_loans30           209593 non-null  int64
 26  amnt_loans30          209593 non-null  int64
 27  maxamnt_loans30       209593 non-null  float64
 28  medianamnt_loans30    209593 non-null  float64
 29  cnt_loans90           209593 non-null  float64
 30  amnt_loans90          209593 non-null  int64
 31  maxamnt_loans90       209593 non-null  int64
 32  medianamnt_loans90    209593 non-null  float64
 33  payback30             209593 non-null  float64
 34  payback90             209593 non-null  float64
 35  pcircle               209593 non-null  object
 36  pdate                 209593 non-null  object
dtypes: float64(21), int64(13), object(3)
memory usage: 59.2+ MB
```

From here we can observe that there is three object type attributes are there .They are msisdn,pcircle,pdate

Further if we see to the data description we see that

In [10]:

```
data.describe().T
```

Out[10]:

	count	mean	std	min	25%	50%	75%	max
label	209593.0	0.875177	0.330519	0.000000	1.000	1.000000	1.00	1.000000
aon	209593.0	8112.343445	75696.082531	-48.000000	246.000	527.000000	982.00	999860.755168
daily_decr30	209593.0	5381.402289	9220.623400	-93.012667	42.440	1469.175667	7244.00	265926.000000
daily_decr90	209593.0	6082.515068	10918.812767	-93.012667	42.692	1500.000000	7802.79	320630.000000
rental30	209593.0	2692.581910	4308.586781	-23737.140000	280.420	1083.570000	3356.94	198926.110000
rental90	209593.0	3483.406534	5770.461279	-24720.580000	300.260	1334.000000	4201.79	200148.110000
last_rech_date_ma	209593.0	3755.847800	53905.892230	-29.000000	1.000	3.000000	7.00	998650.377733
last_rech_date_da	209593.0	3712.202921	53374.833430	-29.000000	0.000	0.000000	0.00	999171.809410
last_rech_amt_ma	209593.0	2064.452797	2370.786034	0.000000	770.000	1539.000000	2309.00	55000.000000
cnt_ma_rech30	209593.0	3.978057	4.256090	0.000000	1.000	3.000000	5.00	203.000000
fr_ma_rech30	209593.0	3737.355121	53643.625172	0.000000	0.000	2.000000	6.00	999606.368132
sumamnt_ma_rech30	209593.0	7704.501157	10139.621714	0.000000	1540.000	4628.000000	10010.00	810096.000000
medianamnt_ma_rech30	209593.0	1812.817952	2070.864620	0.000000	770.000	1539.000000	1924.00	55000.000000
medianmarechprebal30	209593.0	3851.927942	54006.374433	-200.000000	11.000	33.900000	83.00	999479.419319
cnt_ma_rech90	209593.0	6.315430	7.193470	0.000000	2.000	4.000000	8.00	336.000000
fr_ma_rech90	209593.0	7.716780	12.590251	0.000000	0.000	2.000000	8.00	88.000000
sumamnt_ma_rech90	209593.0	12396.218352	16857.793882	0.000000	2317.000	7226.000000	16000.00	953036.000000
medianamnt_ma_rech90	209593.0	1864.595821	2081.680664	0.000000	773.000	1539.000000	1924.00	55000.000000
medianmarechprebal90	209593.0	92.025541	369.215658	-200.000000	14.600	36.000000	79.31	41456.500000

In [11]:

```
#Here we check the summary of object and datetime columns
data.describe(include=['object','datetime'])
```

Out[11]:

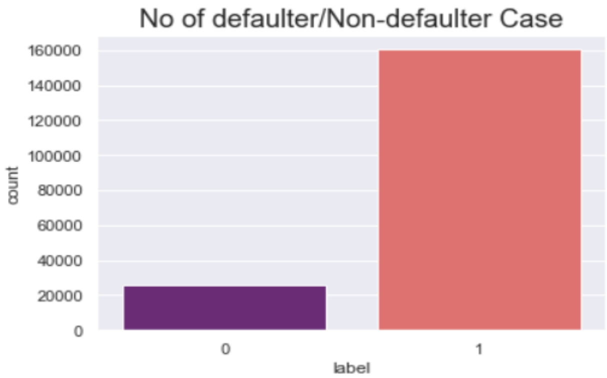
	msisdn	pcircle	pdate
count	209593	209593	209593
unique	186243	1	82
top	04581I85330	UPW	2016-07-04
freq	7	209593	3150

Summary statistics shows all the statistics of our dataset i.e. mean, median and other calculation. Mean is greater than median in all the columns so aur data is right skewed. The difference between 75% and maximum is higher that's why outliers are removed which needs to be removed. The pdate column tells the date when the data is collect. It contains only three month data. msidn is a mobile number of user and mobile number is unique for every customers. There are only 186243 unique number out of 209593 so rest of the data is duplicates entry so we have to remove those entry.

Data Pre-Processing Done

The dataset is imbalanced. Label ‘1’ has approximately 87.5% records, while, label ‘0’ has approximately 12.5% records.

In [28]:
```
#Checking the number of Fraud cases.
sns.countplot(x='label', data=data, palette='magma')
plt.title('No of defaulter/Non-defaulter Case',fontsize=18)
plt.show()

print(data['label'].value_counts())
```




```
1    160383
0     25860
Name: label, dtype: int64
```

After seeing the label column which is also our target feature for this dataset it is clearly shown that 86.11% of data is label 1 and only 13.8% of data is label 0 so our dataset is imbalanced. So before making the ML model first we have to do sampling to get rid-off imbalanced dataset.

For pre-processing of the data as we can observe above that how my data is imbalanced?

Making the new column Day, Month and year from pdate column and after fetching the data from pdate column now we are going to drop it because it has not any significant role.

In [22]:
```
#Making the new column Day, Month and year from pdate column
data['pDay']=pd.to_datetime(data['pdate'],format='%Y/%m/%d').dt.day
data['pMonth']=pd.to_datetime(data['pdate'],format='%Y/%m/%d').dt.month
data['pYear']=pd.to_datetime(data['pdate'],format='%Y/%m/%d').dt.year
```

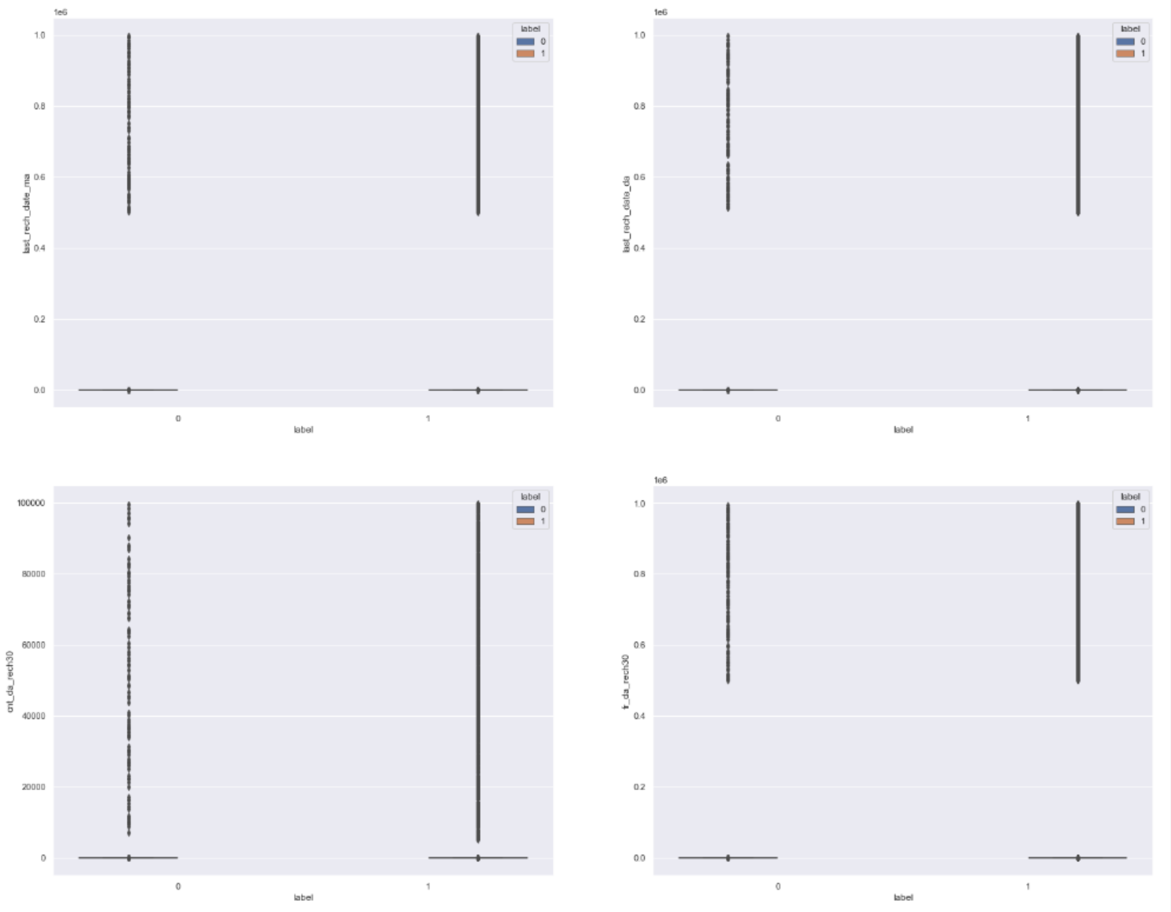
In [25]:
```
#After fetching the data from pdate column now we are
data.drop(columns=['pdate'],axis=1, inplace = True)
```

Separating the categorical columns and Numerical columns

In [26]:
```python
#Seprate the categorical columns and Numerical columns
cat_data,num_data=[],[]

for i in data.columns:
    if data[i].dtype==object:
        cat_data.append(i)
    elif (data[i].dtypes=='int64') | (data[i].dtypes=='float64') | (data[i].dtypes=='int32'):
        num_data.append(i)
    else: continue

print('>>> Total Number of Feature::', data.shape[1])
print('>>> Number of categorical features::', len(cat_data))
print('>>> Number of Numerical Feature::', len(num_data))
```

```
>>> Total Number of Feature:: 34
>>> Number of categorical features:: 2
>>> Number of Numerical Feature:: 32
```

If we plot Outliers and see

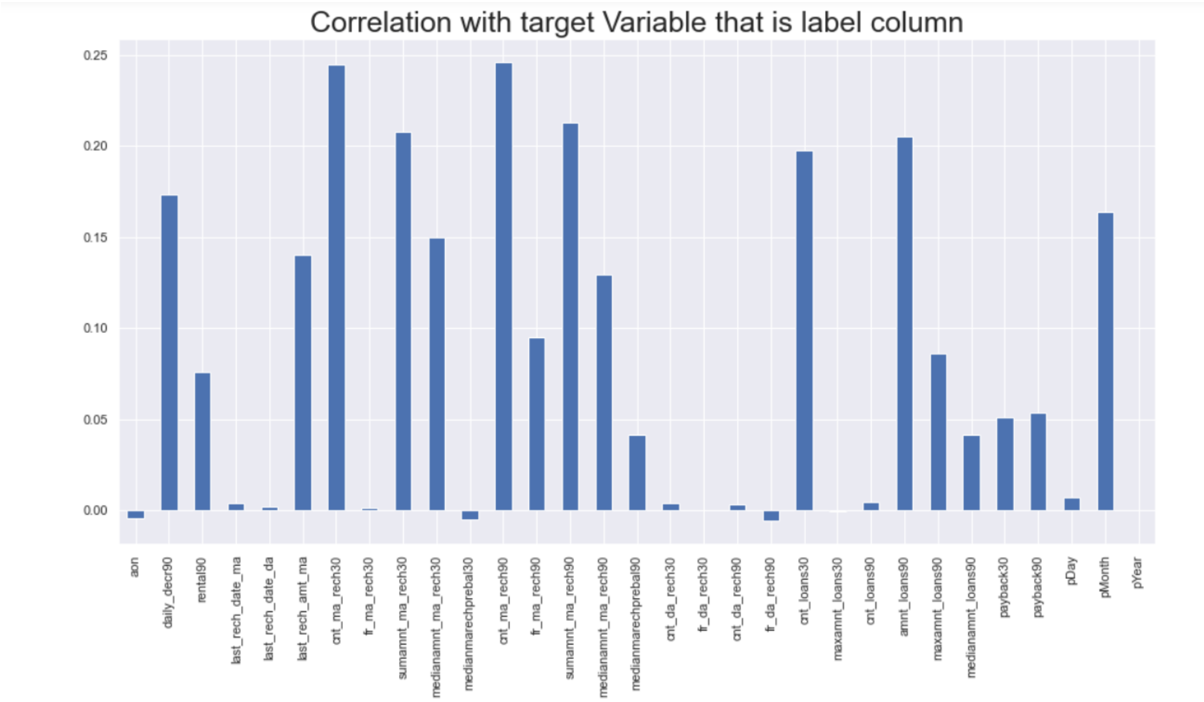


There are too many outliers present in our dataset. So we need to remove it. But before removing please check that only 8 to 10% of data removed .

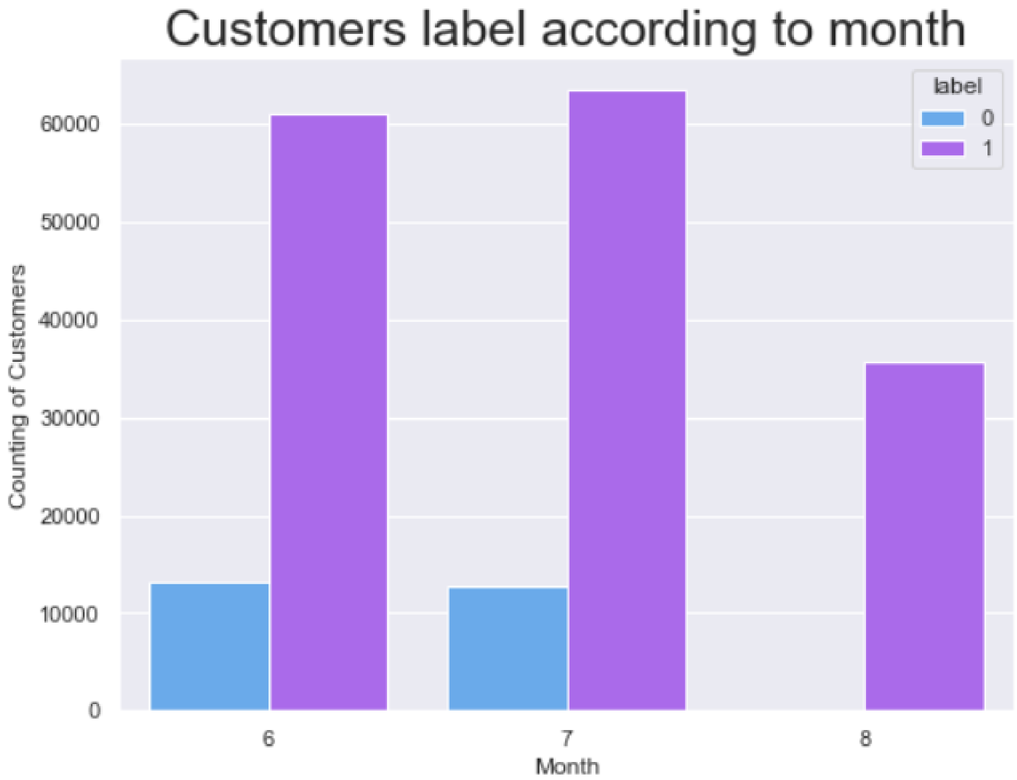
Data Inputs- Logic- Output Relationships

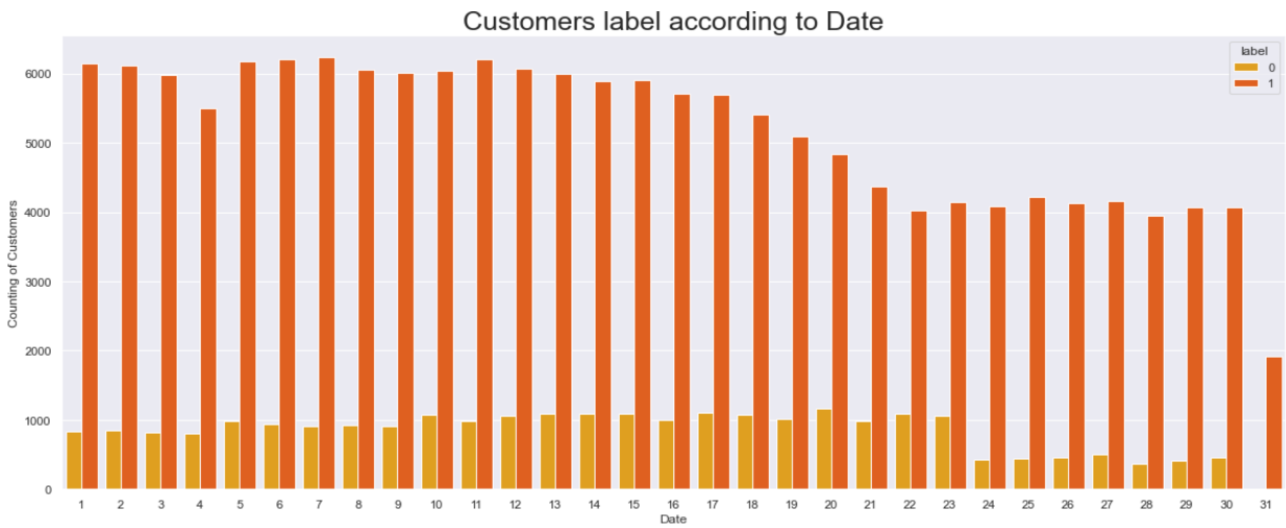
As the data given to us in csv format and client has described it well that he need to predict the defaulters are paying the amount within 5 days or not so it's very clear to us that we have to predict the binary value which is Label '1' indicates that the loan has been payed i.e. Non- defaulter, while, Label '0' indicates that the loan has not been payed i.e. defaulter.

Correlation with target Variable that is label column Here we see the correlation of the columns with respect to the target column that is label.



Understanding Customer label according to Date , Month





The first figure which is date vs label shows that the customers who did not pay their loans are from date 10 to 23. There are several customers at June and July month who did not pay their loan.

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

I have made an assumption that any telecom company keeps the data of customer within 3 months so I have chopped off my data on basis of that.

I have dropped the 2016 year from pdate columns because the data is from the year 2016, only the date and months are different separated months and week to a different columns. Then I separately check the defaulter's data and found that many valuable users are defaulters, they might forgot to pay or their busy life so I separated them so that company can deal politely, because we cannot loose these customers.

Hardware and Software Requirements and Tools Used

So for doing these project the hardware use is a laptop with high end specification, an internet connection. While coming to software I have used anaconda navigator in that I have used **Jupyter notebook** to do my python programming and analysis, for csv file excel is needed.
So in Jupyter notebook I have used lots of python libraries to carry out this project I will be pasting down below with proper justification.

1. Pandas- pandas is used to read the data, visualization and analysis of data.
2. Numpy - used for working with array and various mathematical techniques.
3. Seaborn - I used seaborn for plotting different types of plot.
4. Ploty - Is also used to plot the different types of plot.
5. Matplotlib - It provides an object-oriented API for embedding plots into applications
6. zscore - To remove outliers.
7. skew- to treat skewed data using various transformation like sqrt, log, boxcox.
8. PCA- I used this to remove the data columns to 10.
9. standard scaler- I used this data to scale my data before sending it to model.
10. train_test_split - to split the test and train data.
12. joblib - this is used to save the model pickle file.

Model/s Development and Evaluation

Testing of Identified Approaches (Algorithms)

from sklearn.neighbors import KNeighborsClassifier

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.naive_bayes import GaussianNB

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import StandardScaler

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

Run and Evaluate selected models

What are the algorithm I have used to study this research I have listed above, how I performed these algorithm I have attached the code below.

In [58]:

```
models = []
models.append(('KNeighborsClassifier', KNN))
models.append(('LogisticRegression', LR))
models.append(('DecisionTreeClassifier',DT))
models.append(('GaussianNB', GNB))
models.append(('RandomForestClassifier', RF))
```

In [59]:

```
# lets make the for loop and call the algorithm one by one and save data to respective model using append function.
Model=[]
score=[]
cvs=[]
rocscore=[]
for name,model in models:
    print('******************************',name,'***************************')
    print('\n')
    Model.append(name)
    model.fit(x_train,y_train)
    print(model)
    pre=model.predict(x_test)
    print('\n')
    AS=accuracy_score(y_test,pre)
    print('accuracy_score=',AS)
    score.append(AS*100)
    print('\n')
    sc=cross_val_score(model,x,y,cv=10,scoring='accuracy').mean()
    print('cross_val_score',sc)
    cvs.append(sc*100)
    print('\n')
    false_positive_rate,true_positive_rate,thresholds=roc_curve(y_test,pre)
    roc_auc= auc(false_positive_rate,true_positive_rate)
    print('roc_auc_score = ',roc_auc)
    rocscore.append(roc_auc*100)
```

Key Metrics for success in solving problem under consideration

We can observe that I imported the metrics to find the accuracy score, roc_auc_curve, confusion_matrix, classification_report, in order to interpret the models output. Then I also selected the model to find the cross_validation_score and cross validation prediction.

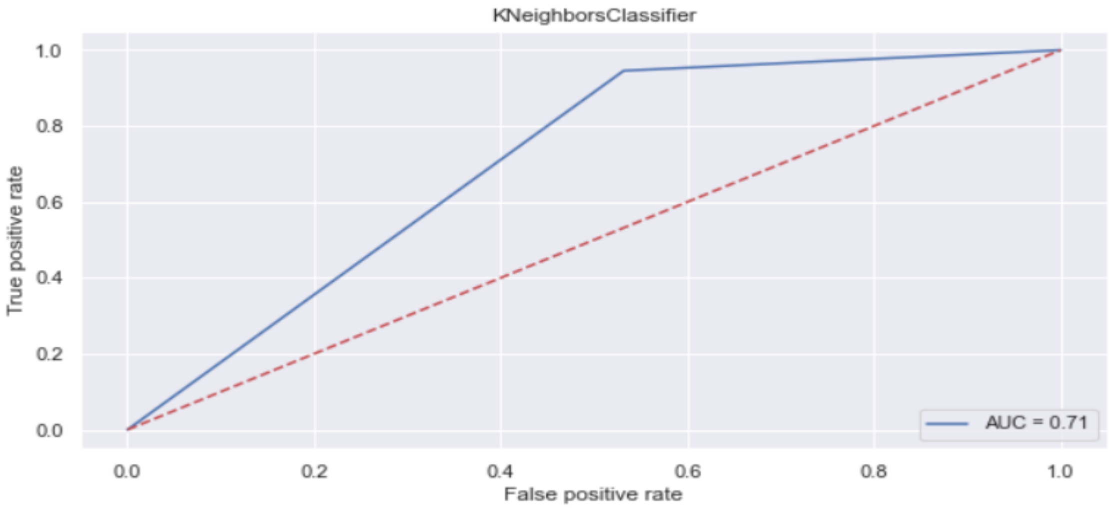
```
LogisticRegression()

accuracy_score= 0.862680877338989

cross_val_score 0.862695504970133

roc_auc_score =  0.5124839149252255

classification_report              precision    recall  f1-score   support

           0       0.62      0.03      0.05      5172
           1       0.86      1.00      0.93     32077

    accuracy                           0.86     37249
   macro avg       0.74      0.51      0.49     37249
weighted avg       0.83      0.86      0.80     37249


[[  143  5029]
 [   86 31991]]


KNeighborsClassifier(n_neighbors=10)

accuracy_score= 0.8792182340465516

cross_val_score 0.8786692399501833

roc_auc_score =  0.7068828737482297

classification_report              precision    recall  f1-score   support

           0       0.58      0.47      0.52      5172
           1       0.92      0.95      0.93     32077

    accuracy                           0.88     37249
   macro avg       0.75      0.71      0.72     37249
weighted avg       0.87      0.88      0.87     37249


[[ 2422  2750]
 [ 1749 30328]]
```

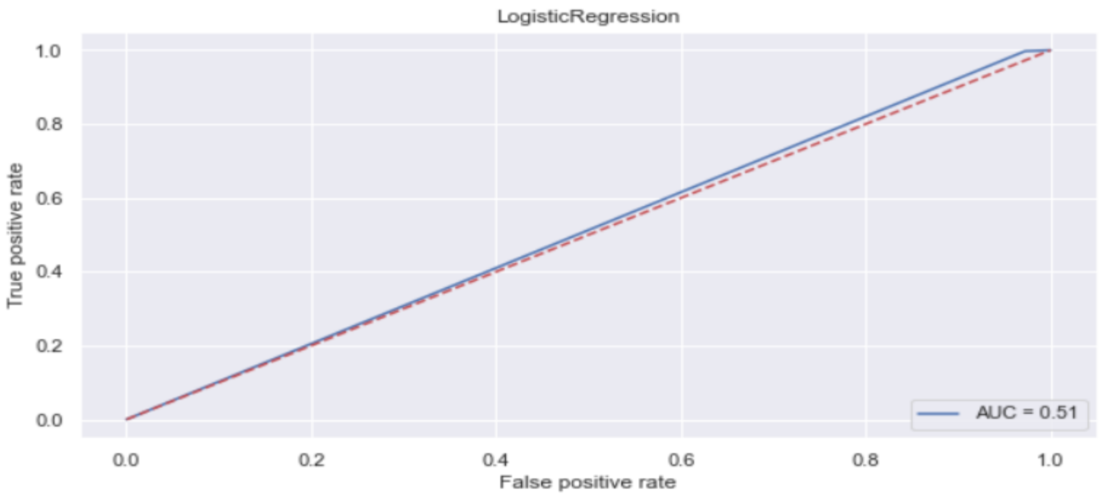
```
DecisionTreeClassifier(random_state=20)

accuracy_score= 0.873848962388252

cross_val_score 0.8742771088470676

roc_auc_score =  0.7459305674008846

classification_report                precision    recall  f1-score   support

           0       0.54      0.57      0.56      5172
           1       0.93      0.92      0.93     32077

    accuracy                           0.87     37249
   macro avg       0.74      0.75      0.74     37249
weighted avg       0.88      0.87      0.88     37249


[[ 2942  2230]
 [ 2469 29608]]
```

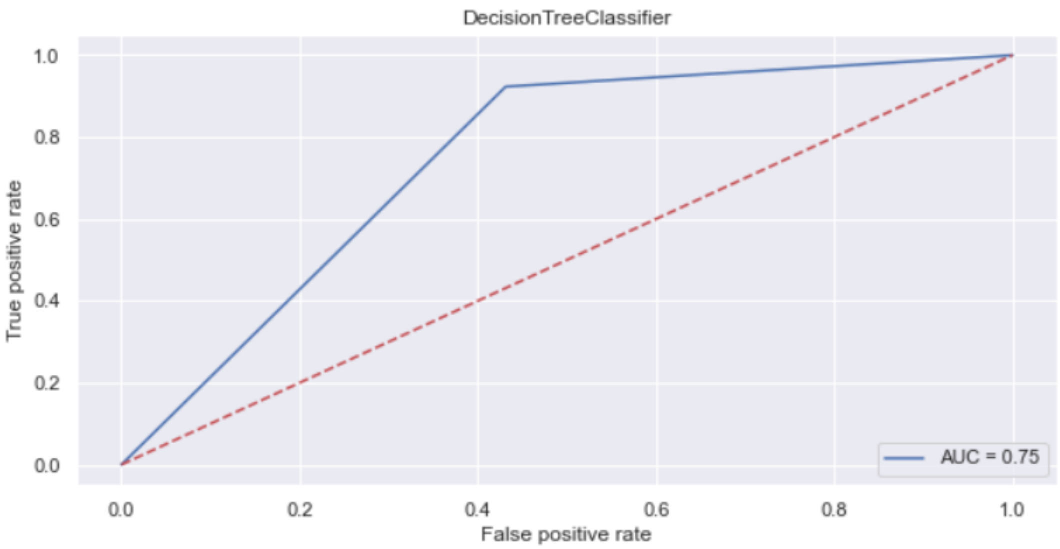
```
GaussianNB()

accuracy_score= 0.6129560525114768

cross_val_score 0.6079689868308771

roc_auc_score =  0.7170547192839658

classification_report                precision    recall  f1-score   support

           0       0.25      0.86      0.38      5172
           1       0.96      0.57      0.72     32077

    accuracy                           0.61     37249
   macro avg       0.60      0.72      0.55     37249
weighted avg       0.86      0.61      0.67     37249


[[ 4454   718]
 [13699 18378]]
```

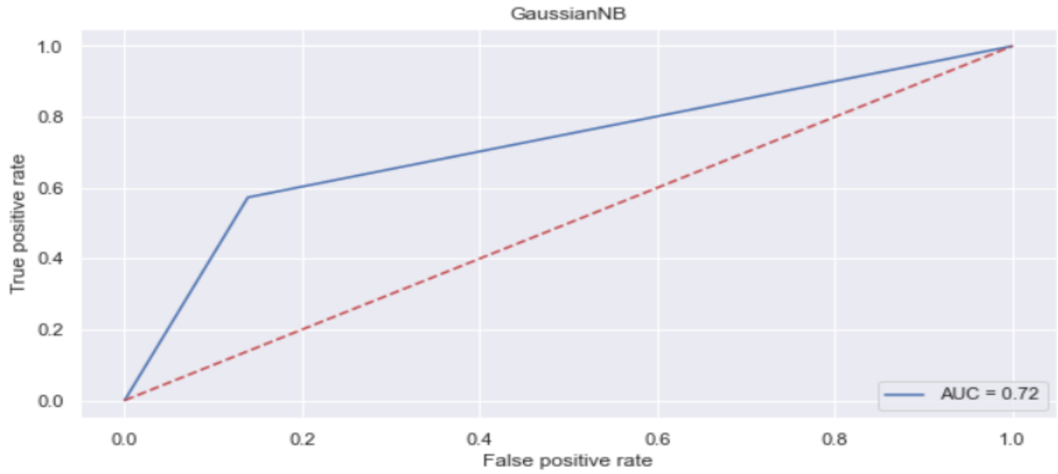
```
RandomForestClassifier()

accuracy_score= 0.9136352653762517

cross_val_score 0.9128718855946862

roc_auc_score =  0.7485973306063298

classification_report                precision    recall  f1-score   support

           0       0.79      0.52      0.63      5172
           1       0.93      0.98      0.95     32077

    accuracy                           0.91     37249
   macro avg       0.86      0.75      0.79     37249
weighted avg       0.91      0.91      0.91     37249


[[ 2690  2482]
 [  735 31342]]
```

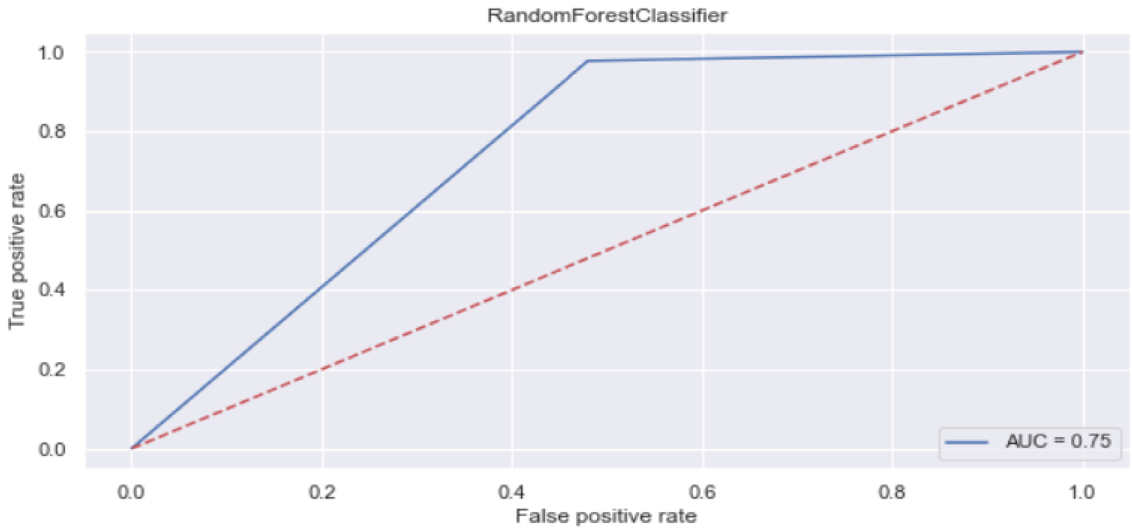
Visualizations








GaussianNB
1.0
0.8
0.6
0.4
0.2
0.0
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
AUC = 0.72


RandomForestClassifier
1.0
0.8
0.6
0.4
0.2
0.0
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
AUC = 0.75

Interpretation of the Results

```
In [60]:  1 result=pd.DataFrame({'Model': Model, 'Accuracy_score': score, 'Cross_val_scor
          2 result
```

Out[60]:

	Model	Accuracy_score	Cross_val_score	Roc_auc_curve
0	KNeighborsClassifier	87.921823	87.866924	70.688287
1	LogisticRegression	86.268088	86.269550	51.248391
2	DecisionTreeClassifier	87.384896	87.427711	74.593057
3	GaussianNB	61.295605	60.796899	71.705472
4	RandomForestClassifier	91.363527	91.287189	74.859733

As I load the dataset I found the data is quite imbalanced and, first thing I found that the age of users in negative count of days so I replaced with 0, then I settled parameters to 130 days for the users who is doing there recharge. Then I found that the company is giving the loan only Rupiah 6 and Rupiah 12 so I have set the parameters and found that 1047 data that is showing more amount of loan which is not possible so instead of replacing the data with most frequent data I just dropped those entire data. Then as I was exploring the defaulter's data found that the users are valuable to our clients but somehow they failed to pay the loan within time frame, so I visualized them individually. As I described earlier that the most number of loan amount is rupiah 6 so which means the users want less amount of loan instead of big amount, hence the client must focus on less amount of loan. The early users or I can say the users who take less number of loans they are more defaulters so the client must be strict to those users. As the data was skewed left from its axis so I used sqrt transformation 3 to reduce down the skewness of data, then I removed the outliers using zscore. Then finally I have prepared the cleaned data for model buildings.

CONCLUSION

Key Findings and Conclusions of the Study

As I load the dataset I found the data is quite imbalanced and, first thing I found that the age of users in negative count of days so I replaced with 0, then I settled parameters to 130 days for the users who is doing there recharge. Then I found that the company is giving the loan only Rupiah 6 and Rupiah 12 so I have set the parameters and found that 1047 data that is showing more amount of loan which is not possible so instead of replacing the data with most frequent data I just dropped those entire data. Then as I was exploring the defaulter's data found that the users are valuable to our clients but somehow they failed to pay the loan within time frame, so I visualised them individually. As I described earlier that the most number of loan amount is rupiah 6 so which means the users want less amount of loan instead of big amount, hence the client must focus on less amount of loan. The early users or I can say the users who take less number of loans they are more defaulters so the client must be strict to those users. As the data was skewed left from its axis so I used sqrt transformation 3 to reduce down the skewness of data, then I removed the outliers using zscore. Then finally I have prepared the cleaned data for model buildings.

Learning Outcomes of the Study in respect of Data Science

As exploring this dataset I came across various challenges to tackle first, I saw that the data is so imbalanced that irrespective of any null data the dataset was so complicated, while visualizing this data I can say how company can boom with the help of Data science, as I studied this dataset I can say company should increase their days of payment cause most of the users paid their loan within 7-8 days, company should focused more on less amount of loan. As building this model I can say the KNeighbors and Random forest is working quite well with accuracy score of almost 90% with roc_auc_score of 73%. After building the model I predicted the same input and got quite good response from the model. Though the dataset were quite imbalanced and challenging but I have learned so many thing while exploring it.

Limitations of this work and Scope for Future Work

So after building this model, I can't say that this research is enough, there is no word of enough in Data Science, people with other mind set can explore this data in different angle, the limitation for this dataset is, we have to use the data set more carefully cause the dataset is having lots of outliers and data is much more skewed and as per the client we can't lose the dataset more than 8%. Finally that's all from my side future improvement can be possible with this dataset.