Machine Learning applied: Prediction of Micro-Credit Defaulter based on data from mobile financial services (MFS)

Submitted by:
Madhurima Srivastava
Internship-30

ACKNOWLEDGMENT

A Microfinance Institution (MFI) is an organization that offers financial services to lowincome 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. 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 service 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. And we have collected the following data of mobile network userfrom our client database in using which we are training our model for prediction.

1. label - 1 not-Defaulter, 0 Defaulter (TARGET VARIABLE)
2. msisdn - mobile number of users
3. aon - age on cellular network in days
4. daily_decr30 - averaged over last 30 days
5. daily_decr90 - averaged over last 90 days
6. rental30 - Average main account balance over last 30 days
7. rental90 - Average main account balance over last 90 days
8. last_rech_date_ma - Number of days till last recharge of main account
9. last_rech_date_da - Number of days till last recharge of data account
10. last_rech_amt_ma - Amount of last recharge of main account
11. cnt_ma_rech30 - No. of times recharge in last 30 days
12. fr_ma_rech30 - Frequency of recharge in last 30 days
13. sumamnt_ma_rech30 - Sum of recharge in 30 days
14. medianamnt_ma_rech30 - median of recharge in 30 days
15. medianmarechprebal30 - median of balance in last 30 days
16. cnt_ma_rech90 - No. of times recharge in last 90 days
17. fr_ma_rech90 - Frequency of recharge in last 90 days
18. sumamnt_ma_rech90 - Sum of recharge in 90 days
19. medianamnt_ma_rech90 - median of recharge in 90 days
20. medianmarechprebal90 - median of balance in last 90 days
21. cnt_da_rech30 - No. of times data recharge in last 30 days
22. fr_da_rech30 - Frequency of data recharge in last 30 days

23. cnt_da_rech90 - No. of times data recharge in last 90 days

24. fr_da_rech90 - Frequency of data recharge in last 90 days

25. cnt_loans30 - Number of loans taken by user in last 30 days

26. amnt_loans30 - Total amount of loans taken by user in last 30 days

27. maxamnt_loans30 - max amount taken in last 30 days

28. medianamnt_loans30 - Median of amounts of loan taken by the user in last 30 days

29. cnt_loans90 - Number of loans taken by user in last 90 days

30. amnt_loans90 - Total amount of loans taken by user in last 90 days

31. maxamnt_loans90 - maximum amount of loan taken by the user in last 90 days

32. medianamnt_loans90 - Median of amounts of loan taken by the user in last 90 days

33. payback30 - Average payback time in days over last 30 days

34. payback90 - Average payback time in days over last 90 days

35. pcircle - telecom circle

36. pdate – Date

Many (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 forthe 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. The sample data is provided to us from the client database. In understanding the above data, I have tried to plot said factors and with the use of machine learning, tried and successfully arrived at a model that can detect whether the customer will pay back the loaned amount within 5 days of insurance of loan. 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 i.e., defaulter bases these factors. We can use this model in order to improve the selection of customers for the credit and do some predictions that could help in further investment and improvement in selection of customers.

INTRODUCTION

• Business Problem Framing.

The Data is provided to us from the Client Database which has 209593 records of mobile phone userin Indonesia. These records are about the mobile phone user does transaction with the mobile network service providers such as recharges done in 30 days and 90 days, loan taken within last 30 and 90 days, the trend of repayment for every loan taken and also the tenure of the users in the same network. Understanding the above factors as said I have tried to plot and predict using machine learning, tried and successfully arrived at a model. So, using the model in the future the Client can predict and analyse the customer behaviour and also come to a conclusion whether to sanction a loan to a specified customer or not and also in order to improve the onboarding ofthe customers for the credit.

• Conceptual Background of the Domain Problem.

Comprehensive support from financial institutions is required in effort to drive community empowerment, particularly middle to low-income society and micro, small and medium enterprises (UMKM). This group of enterprises has limited access to formal financial institutions so far. Therefore, in order to deal with such problems, many non-bank financial institutions have grown and developed in society, running services in business development and community empowerment, and are established by government or society. Those institutions are well-known as microfinance institution (MFIs). However, many of the MFIs still do not have legal entity or business license yet. In order to provide a strong legal groundwork for MFIs` operation, Law Number 1 of 2013 on MFIs has been issued on January 8, 2013.

Legal Groundwork

1. Law Number 1 of 2013 on microfinance institutions (MFI Law).

2. Government Regulation Number 89 of 2014 on loan interest rate or yield of financing and MFI`s business coverage.

3. OJK Regulations (POJK):

(a) OJK Regulation Number 12/POJK.05/2014 on business licensing and institutional matters of MFIs.

(b) OJK Regulation Number 13/POJK.05/2014 on business management of MFIs.

(c) OJK Regulation Number 14/POJK.05/2014 on fostering and supervision of MFIs.

MFI`s Business Activities

1) MFI`s business activities cover business development and community empowerment through loan or financing for micro-scaled business of MFI members and society, deposit management, or giving consultancy services in business development.

2) Business activities as mentioned above can be carried out using conventional practices or based on Sharia principles.

Objectives of MFI

1) To improve access to micro-scaled funding for society;

2) To help improving economic empowerment and productivity in society; and

3) To help increasing society`s income and prosperity, mainly of disadvantaged and/or low-income society.

MFI Ownership An MFI can be owned by:

1) Indonesian citizen;

2) Village/rural enterprise;

3) Regional government (regency/city); and/or

4) Cooperative

MFI is not allowed to be owned, whether directly or indirectly, by foreign citizen and/or enterprise owned in whole or in part by foreign citizen or foreign enterprise.

• Review of Literature

This paper provides a systematic assessment of customer behaviour and the trend of loan taken Vs loan paid back to the Microfinance Institution (MFI) in Indonesia bases on the above given factors. From our analysis and research done one can understand that depending on what factors customers are ending up on being a defaulter, using this study a Microfinance Institution (MFI) can predict that a customer can be a defaulter or not basis on which Microfinance Institution (MFI) can confer short-term loans.

Further our study helps the Micro finance institution to understand, analysis and predict the futuristic behaviour of a customer that could help them in further speculation and improvement in selection of customers.

• Motivation for the Problem Undertaken

Build a model using the best machine learning algorithm which can be used 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 paid i.e., Non- defaulter, while, Label '0' indicates that the loan has not been paid i.e., defaulter.

Analytical Problem Framing

• Mathematical/ Analytical Modelling of the Problem

Our data consist of 36 columns with 209592 records using which we have done few Modelling and derived mathematical summaries and we have also got many insights from the same. Let's visualize the modelling in detail.

1. **Mathematical summary of the data.**

	count	mean	std	min	25%	50%	75%	max
label	209592.0	0.875177	0.330519	0.000000	1.000000	1.000000	1.0000	1.000000
aon	209592.0	8112.380399	75696.261220	-48.000000	246.000000	527.000000	982.0000	999860.755168
daily_decr30	209592.0	5381.412999	9220.644093	-93.012667	42.439500	1469.091833	7244.0960	265926.000000
daily_decr90	209592.0	6082.529123	10918.836919	-93.012667	42.691917	1500.000000	7802.7950	320630.000000
rental30	209592.0	2692.578912	4308.596841	-23737.140000	280.417500	1083.540000	3356.9450	198926.110000
rental90	209592.0	3483.407309	5770.475034	-24720.580000	300.260000	1334.000000	4201.7925	200148.110000
last_rech_date_ma	209592.0	3755.865715	53906.020204	-29.000000	1.000000	3.000000	7.0000	998650.377733
last_rech_date_da	209592.0	3712.220632	53374.960145	-29.000000	0.000000	0.000000	0.0000	999171.809410
last_rech_amt_ma	209592.0	2064.458973	2370.790003	0.000000	770.000000	1539.000000	2309.0000	55000.000000
cnt_ma_rech30	209592.0	3.978053	4.256099	0.000000	1.000000	3.000000	5.0000	203.000000
fr_ma_rech30	209592.0	3737.372947	53643.752523	0.000000	0.000000	2.000000	6.0000	999606.368132
sumamnt_ma_rech30	209592.0	7704.496570	10139.645685	0.000000	1540.000000	4628.000000	10010.0000	810096.000000
medianamnt_ma_rech30	209592.0	1812.819258	2070.869474	0.000000	770.000000	1539.000000	1924.0000	55000.000000
medianmarechprebal30	209592.0	3851.945862	54006.502647	-200.000000	11.000000	33.900000	83.0000	999479.419319
cnt_ma_rech90	209592.0	6.315437	7.193487	0.000000	2.000000	4.000000	8.0000	336.000000
fr_ma_rech90	209592.0	7.716812	12.590273	0.000000	0.000000	2.000000	8.0000	88.000000
sumamnt_ma_rech90	209592.0	12396.236149	16857.832129	0.000000	2317.000000	7226.000000	16000.0000	953036.000000
medianamnt_ma_rech90	209592.0	1864.597375	2081.685508	0.000000	773.000000	1539.000000	1924.0000	55000.000000
medianmarechprebal90	209592.0	92.025522	369.216539	-200.000000	14.600000	36.000000	79.3100	41456.500000
cnt_da_rech30	209592.0	262.579362	4183.907920	0.000000	0.000000	0.000000	0.0000	99914.441420
fr_da_rech30	209592.0	3749.512336	53885.542905	0.000000	0.000000	0.000000	0.0000	999809.240107
cnt_da_rech90	209592.0	0.041495	0.397557	0.000000	0.000000	0.000000	0.0000	38.000000
fr_da_rech90	209592.0	0.045713	0.951388	0.000000	0.000000	0.000000	0.0000	64.000000
cnt_loans30	209592.0	2.758975	2.554507	0.000000	1.000000	2.000000	4.0000	50.000000
amnt_loans30	209592.0	17.951992	17.379778	0.000000	6.000000	12.000000	24.0000	306.000000
maxamnt_loans30	209592.0	274.660029	4245.274734	0.000000	6.000000	6.000000	6.0000	99864.560864
medianamnt_loans30	209592.0	0.054029	0.218039	0.000000	0.000000	0.000000	0.0000	3.000000
cnt_loans90	209592.0	18.520988	224.797957	0.000000	1.000000	2.000000	5.0000	4997.517944
amnt_loans90	209592.0	23.645397	26.469924	0.000000	6.000000	12.000000	30.0000	438.000000
maxamnt_loans90	209592.0	6.703138	2.103869	0.000000	6.000000	6.000000	6.0000	12.000000
medianamnt_loans90	209592.0	0.046078	0.200692	0.000000	0.000000	0.000000	0.0000	3.000000
payback30	209592.0	3.398639	8.813330	0.000000	0.000000	0.000000	3.7500	171.500000
payback90	209592.0	4.321302	10.307791	0.000000	0.000000	1.666667	4.5000	171.500000
pdate_day	209592.0	14.398899	8.438899	1.000000	7.000000	14.000000	21.0000	31.000000
pdate_month	209592.0	6.797321	0.741437	6.000000	6.000000	7.000000	7.0000	8.000000

Key observations:

1. From the above data it is clear that the data has no null values.

2. Categorical Columns: “label”

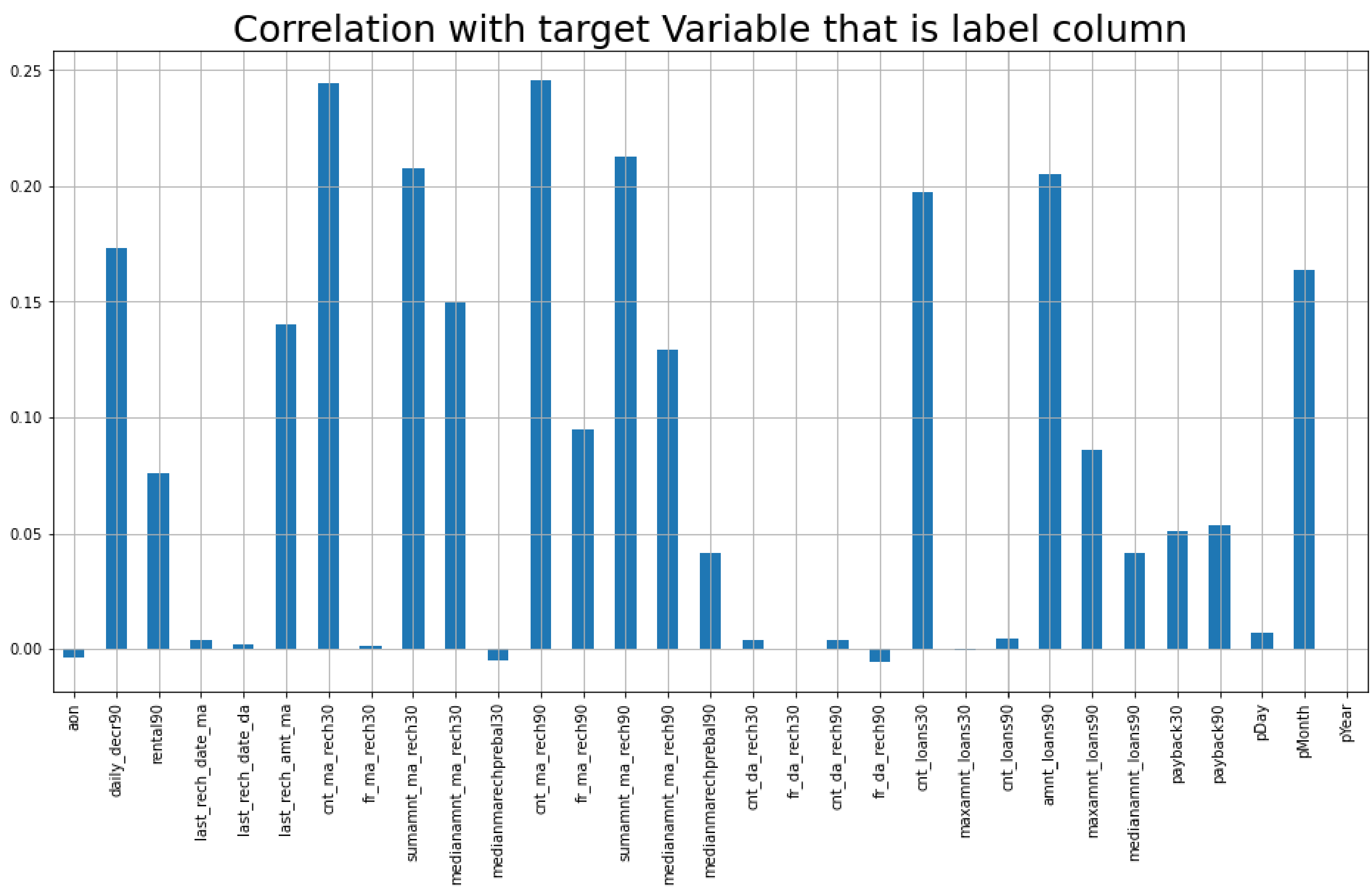
3. Continuous Data Columns: Remaining all 35 Columns are continuous data

4. There is large difference between 75% percentile and Max Values which means it has more outliers.

5. Mean is greater than median which also means data have skewness present.

2. Correlation of Features with the Target column

Let’s see the correlations of the data with Target variable so we can analyse depending on which feature variable the Target variable is decided.



• Data Sources and their formats :

The Data is provided to us from the Client Database which has 209593 records of mobile phone user in Indonesia. These records are about the mobile phone user does transaction with the mobile network service providers such as recharges done in 30 days and 90 days, loan taken within last 30 and 90 days, the trend of repayment for every loan taken and also the tenure of the users in the same network. These Data are collected and stored in CSV format. Which we are using for the study and analysis.

• *Data Pre-processing:*

Done We started our pre-processing pipeline in importing the required libraries and we imported the given data.

```
In [1]: import pandas as pd # for data wrangling purpose
        import numpy as np # Basic computation library
        import seaborn as sns # For Visualization
        import matplotlib.pyplot as plt # ploting package
        %matplotlib inline
        import warnings # Filtering warnings
        warnings.filterwarnings('ignore')
```

```
In [2]: df=pd.read_csv(r"C:\Users\Lenovo\OneDrive\Desktop\Data_file.csv",index_col=0,parse_dates=['pdate'])
        df.head(10)
```

Out[2]:

	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	...	maxamnt_loans
1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	1539	...	6
2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	5787	...	12
3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	1539	...	6
4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	947	...	6
5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	2309	...	6
6	1	35819I70783	568.0	2257.362667	2261.460000	368.13	380.13	2.0	0.0	1539	...	6
7	1	96759I84459	545.0	2876.641667	2883.970000	335.75	402.90	13.0	0.0	5787	...	6
8	1	09832I90846	768.0	12905.000000	17804.150000	900.35	2549.11	4.0	55.0	3178	...	6
9	1	59772I84450	1191.0	90.695000	90.695000	2287.50	2287.50	1.0	0.0	1539	...	6

1. Remove columns where number of unique value is only 1.

Let's look at no of unique values for each column. We will remove all columns where number of unique value is only 1 because that will not make any sense in the analysis.

```
In [9]: unique = df.nunique()
        unique = unique[unique.values == 1]
```

```
In [10]: df.drop(labels = list(unique.index), axis =1, inplace=True)
         print("So now we are left with",df.shape ,"rows & columns.")

         So now we are left with (209593, 35) rows & columns.
```

```
In [11]: df.describe().transpose()
```

Out[11]:

	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

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.

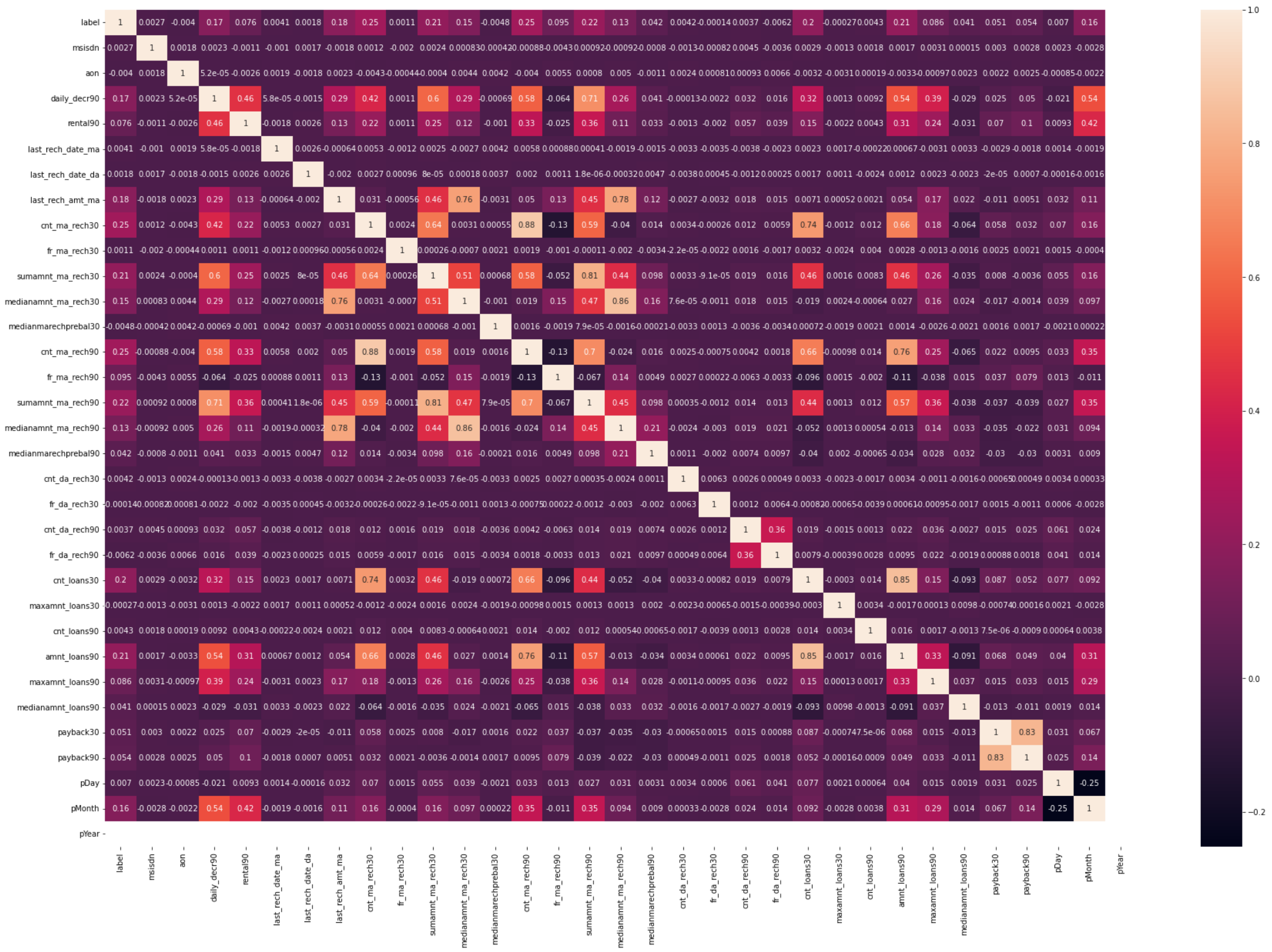
msidn is a mobile number of user and mobile number is unique for every customers. There are only 186243 unique number out of 209593 so rest of the data is duplicates entry so we have to remove those entry.

Data Exploration

```
In [19]: #Checking the number of number of defaulter and non defaulter customers.
         df['label'].value_counts()

Out[19]: 1    160383
         0     25860
         Name: label, dtype: int64
```

```
In [20]: #Checking the defaulter customers percentage wise.
         df['label'].value_counts(normalize=True) *100

Out[20]: 1    86.114914
         0    13.885086
         Name: label, dtype: float64
```

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 implanced. So before making the ML model first we have to do sampling to get rid off imblance dataset.



Observations:

daily_decr30 and daily_decr90 features are highly correlated with each other.

rental30 and rental90 features are highly correlated with each other.

cnt_loans30 and amount_loans30 columns are highly correlated with each other.

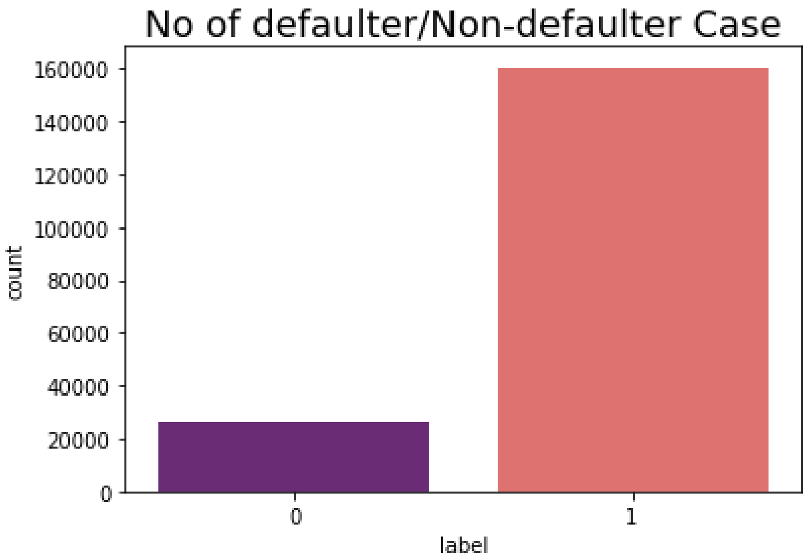
amount_loans30 is also highly correlated with amount_loans90 column.

medianamnt_loans30 and medianamnt_loans90 is highly correlated with each other.

We have to drop one of the features which are highly correlated with other feayures. And if we dont do this then our model will face multicolinearity problem.

Data Visualization

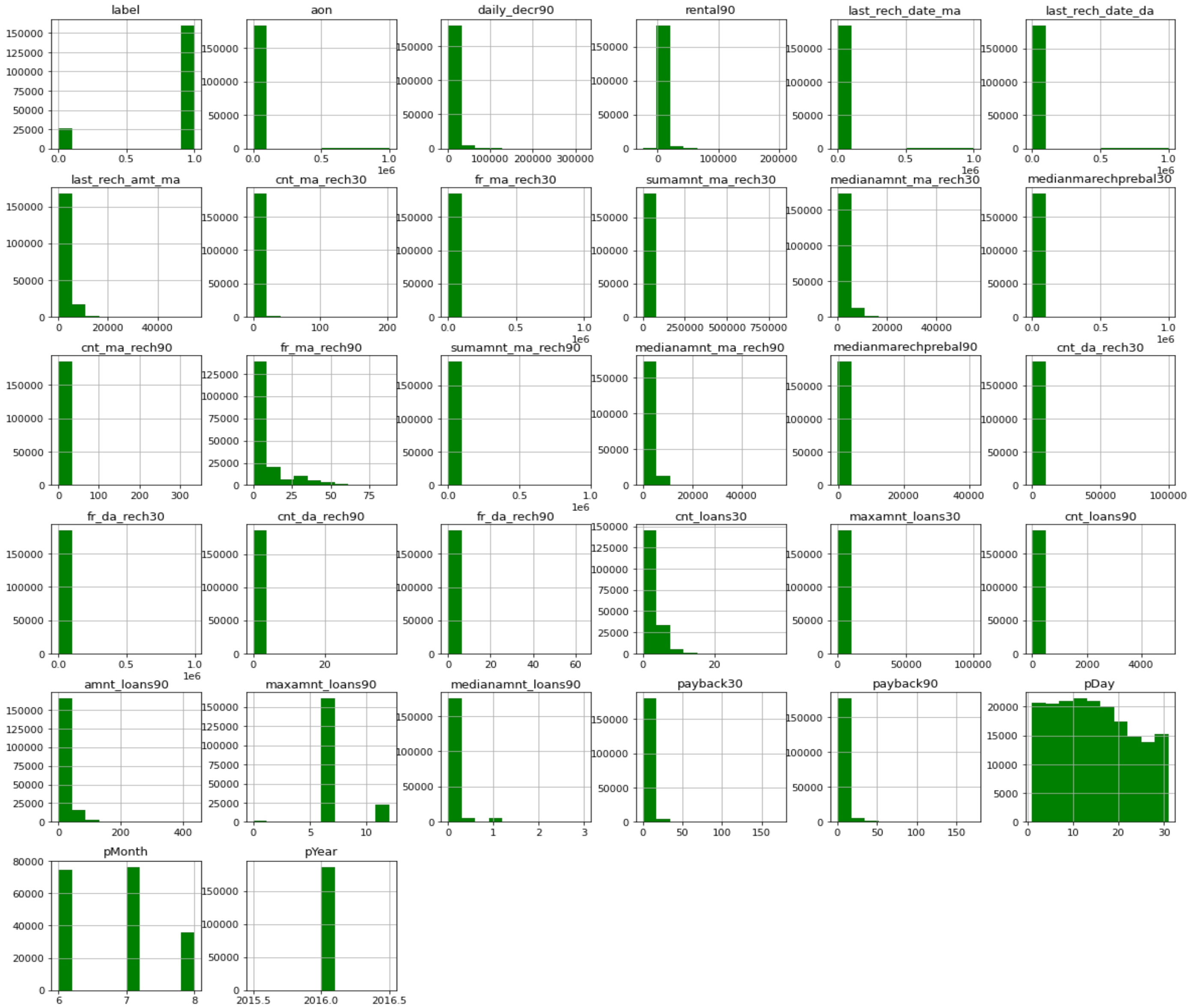
In [31]:
```
#Checking the number of Fraud cases.
sns.countplot(x='label', data=df, palette='magma')
plt.title('No of defaulter/Non-defaulter Case',fontsize=18)
plt.show()

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



Label 1 indicates loan has been paid i.e Non-Defaulter and label 0 indicates that the loan has not been paid i.e. defaulter.

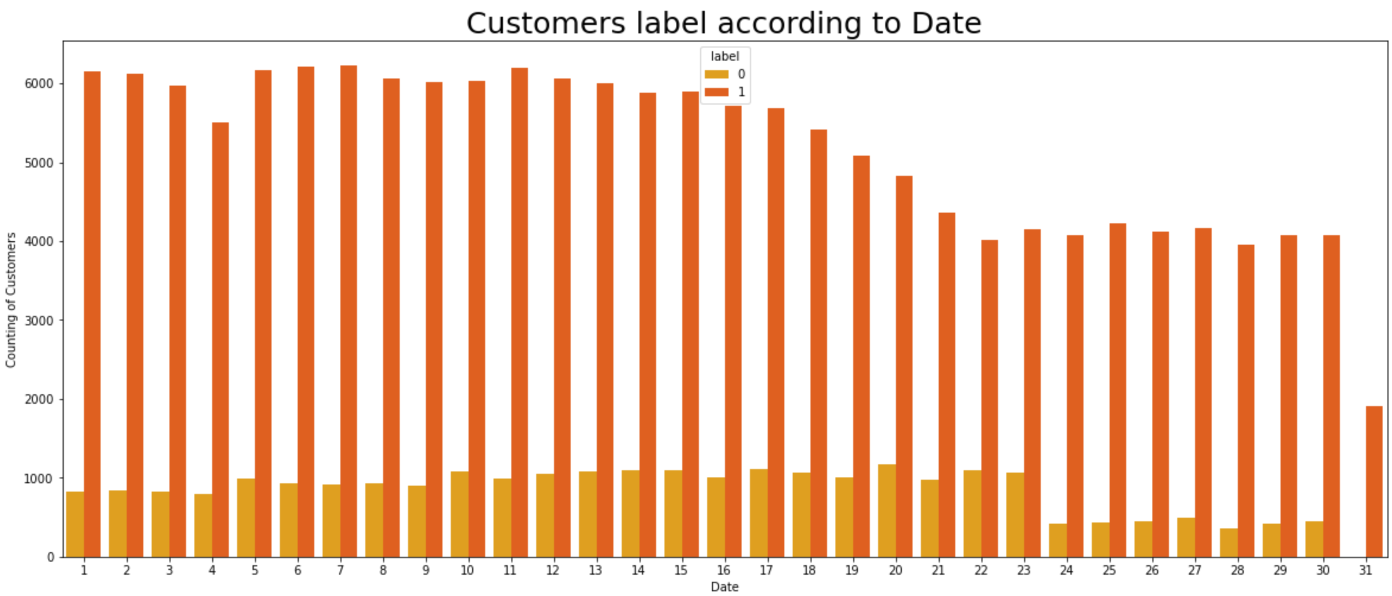
In [32]:
```
#Plotting the Histogram
df.hist(figsize=(20,20),color='green')
plt.show()
```

We plot the histogram to display the shape and spread of continuous sample data.In a histogram, each bar groups numbers into ranges. Taller bars show that more data falls in that range

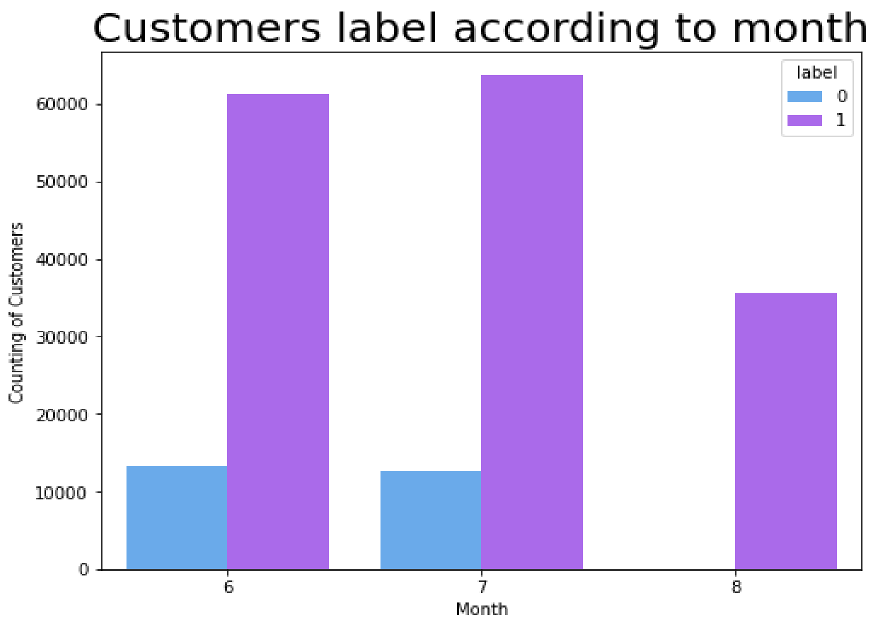


In [33]:

```
#Customer Label according to Date
plt.figure(figsize=(20,8))
sns.countplot(x="pDay", hue='label', data=df, palette='autumn_r')
plt.title("Customers label according to Date", fontsize=25)
plt.xlabel('Date')
plt.ylabel('Counting of Customers')
plt.show()
```



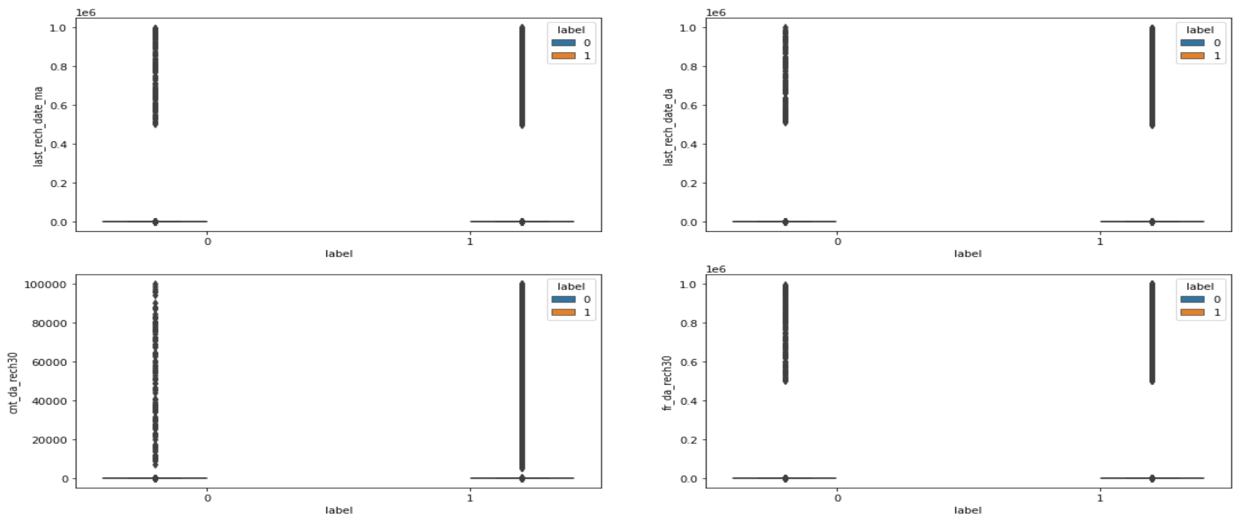
In [34]:
```python
#Customer label according to Month
plt.figure(figsize=(8,6))
sns.countplot(x="pMonth", hue='label', data=df, palette='cool')
plt.title("Customers label according to month", fontsize=25)
plt.xlabel('Month')
plt.ylabel('Counting of Customers')
plt.show()
```



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.

Outliers:

In [37]:
```python
#plotting outliers

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=2, ncols=2, figsize = (18, 10))
sns.boxplot(ax=ax1, x = 'label', y = 'last_rech_date_ma', hue = 'label', data = df)
sns.boxplot(ax=ax2, x = 'label', y = 'last_rech_date_da', hue = 'label', data = df)
sns.boxplot(ax=ax3, x = 'label', y = 'cnt_da_rech30', hue = 'label', data = df)
sns.boxplot(ax=ax4, x = 'label', y = 'fr_da_rech30', hue = 'label', data = df)
```



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.

Lets find and clean the outliers:

In [40]:
```
from scipy.stats import zscore
z=np.abs(zscore(df1))
z
```

Out[40]:

	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cnt_ma_rech30	...	cr
1	2.647896	0.103577	0.252299	0.276346	0.573844	0.558583	0.069637	0.069550	0.221637	0.464760	...	
2	0.377658	0.097764	0.731037	0.553380	0.231788	0.036020	0.069303	0.069550	1.570178	0.699718	...	
3	0.377658	0.100102	0.432011	0.429033	0.416020	0.447674	0.069619	0.069550	0.221637	0.699718	...	
4	0.377658	0.103986	0.581326	0.555125	0.587935	0.576036	0.068914	0.069550	0.471344	0.934677	...	
5	0.377658	0.094660	0.567293	0.543274	0.369886	0.413227	0.069600	0.069550	0.103151	0.710030	...	
...	...	...	...	...	...	...	...	...	...	...	...	
209589	0.377658	0.101833	0.567157	0.543159	0.372140	0.414910	0.069656	0.069550	0.836664	0.229802	...	
209590	0.377658	0.092969	0.579622	0.553686	0.223791	0.304144	0.069600	0.069550	0.544737	0.005156	...	
209591	0.377658	0.093788	0.700790	0.533194	0.735567	0.937500	0.069619	0.069550	0.221637	0.240114	...	
209592	0.377658	0.084289	0.770755	0.594558	0.529352	0.433039	0.069637	0.068838	0.544737	0.240114	...	
209593	0.377658	0.086284	0.096744	0.141746	0.512620	0.494278	0.069433	0.069550	2.303692	0.464760	...	

209593 rows × 33 columns

In [41]:
```
threshold=3
print(np.where(z>3))
```

```
(array([     21,      22,      22, ..., 209586, 209587, 209587], dtype=int64), array([15, 15, 32, ..., 28, 26, 30], dtype=int6
4))
```

Model Training

In [49]:
```
#Scaling in input variables
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
x=ss.fit_transform(x)
```

In [50]:
```
#Splitting the data into training and testing data

from sklearn.model_selection import train_test_split,cross_val_score
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.20,random_state=42,stratify=y)
```

In [51]:
```
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
```

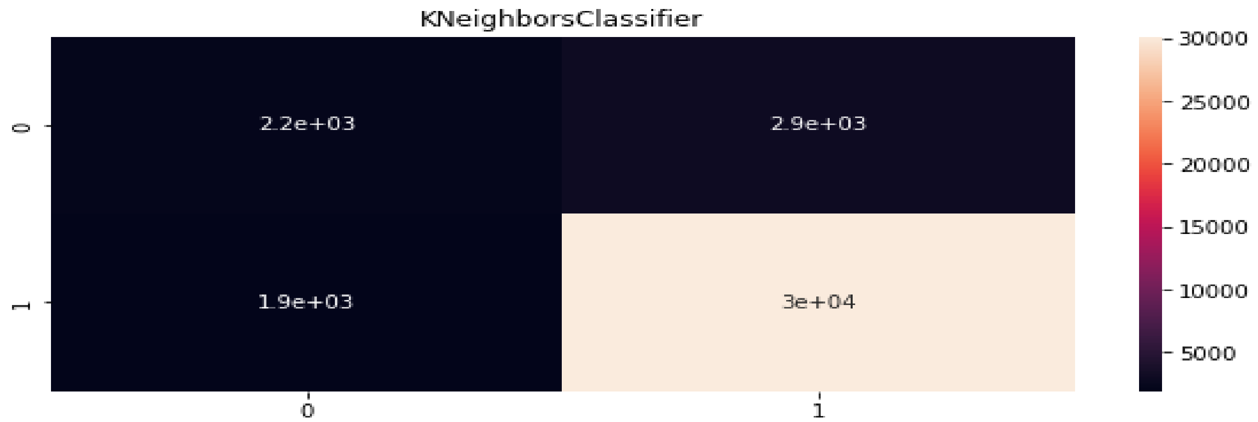
In [52]:
```
KNN=KNeighborsClassifier(n_neighbors=10)
LR=LogisticRegression()
DT=DecisionTreeClassifier(random_state=20)
GNB=GaussianNB()
RF=RandomForestClassifier()
```

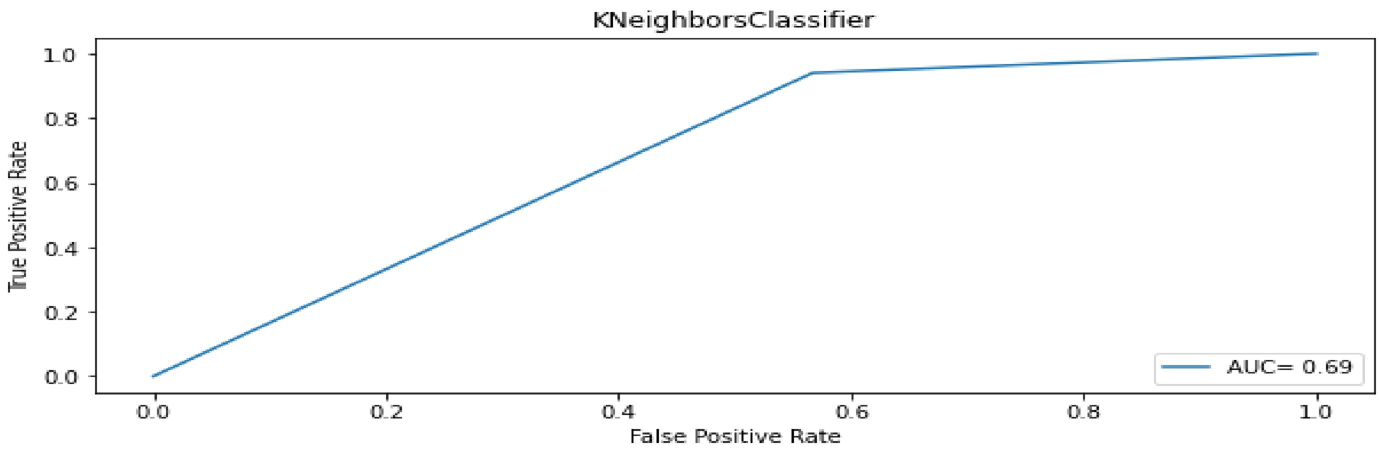
In [53]:
```
models = []
models.append(('KNeighborsClassifier', KNN))
models.append(('LogisticRegression', LR))
models.append(('DecisionTreeClassifier',DT))
models.append(('GaussianNB', GNB))
models.append(('RandomForestClassifier', RF))
```

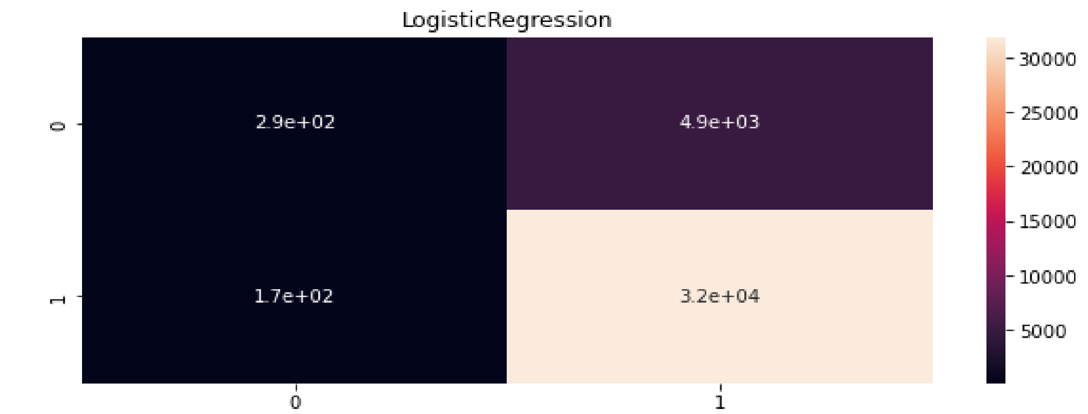
Evaluation started with importing the required libraries and with help of cross validation I have filtered and shortlisted the best algorithm model and I have hyper tunes the same. We will visualize the shortlisting of best performed model as follows

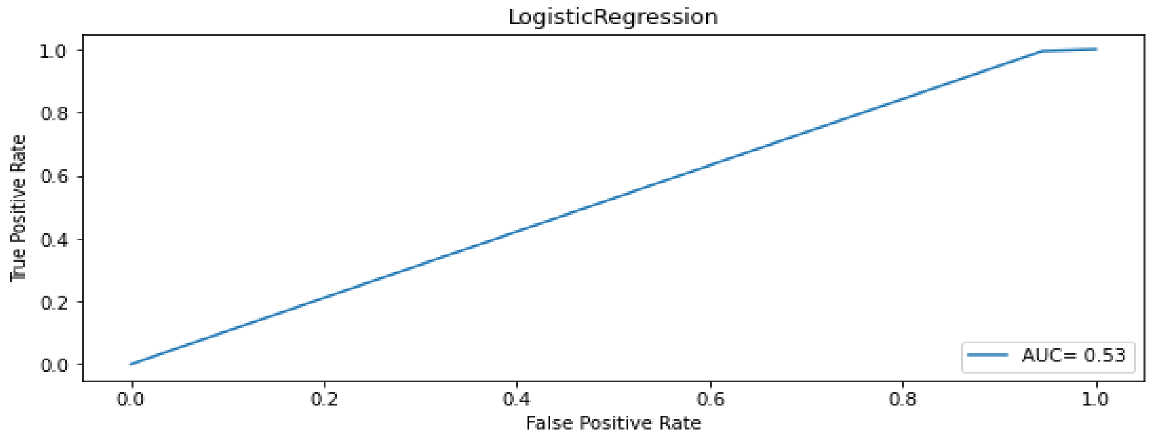
In [54]:
```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score,roc_curve,auc
```

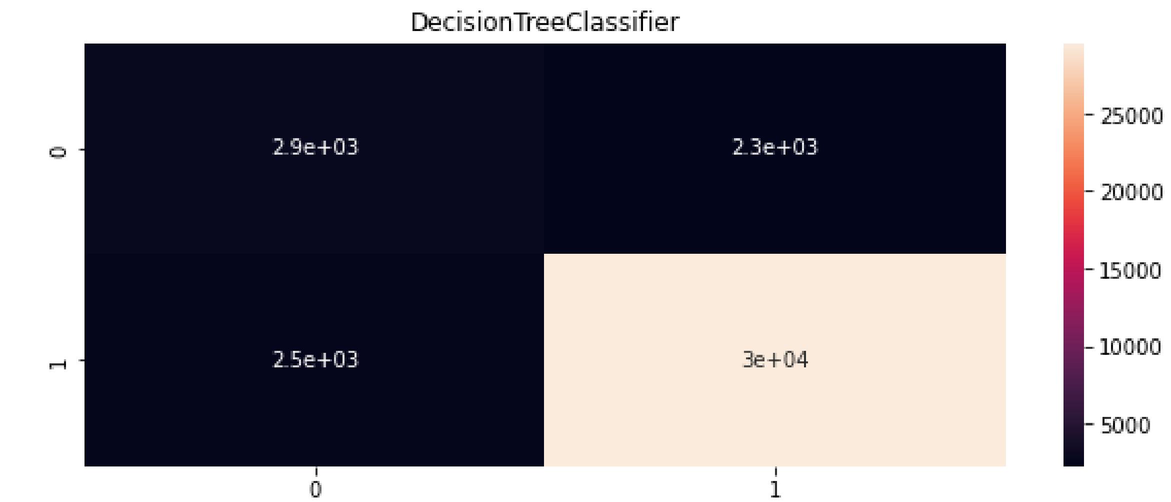
In [55]:
```
Model=[]
score=[]
cvs=[]
rocscore=[]
for name,model in models:
    print('****************************',name,'********************************')
    print('\n')
    Model.append(name)
    model.fit(x_train,y_train.values.ravel())
    print(model)
    pre=model.predict(x_test)
    print('\n')
    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)
    print('\n')
    print('classification_report\n',classification_report(y_test,pre))
    print('\n')
    cm=confusion_matrix(y_test,pre)
    print(cm)
    print('\n')
    plt.figure(figsize=(10,40))
    plt.subplot(911)
    plt.title(name)
    print(sns.heatmap(cm,annot=True))
    plt.subplot(912)
    plt.title(name)
    plt.plot(false_positive_rate, true_positive_rate, label = 'AUC= %0.2f'%roc_auc)
    plt.legend(loc='lower right')
    plt.ylabel('True Positive Rate')
    plt.xlabel('False Positive Rate')
    print('\n\n')
```

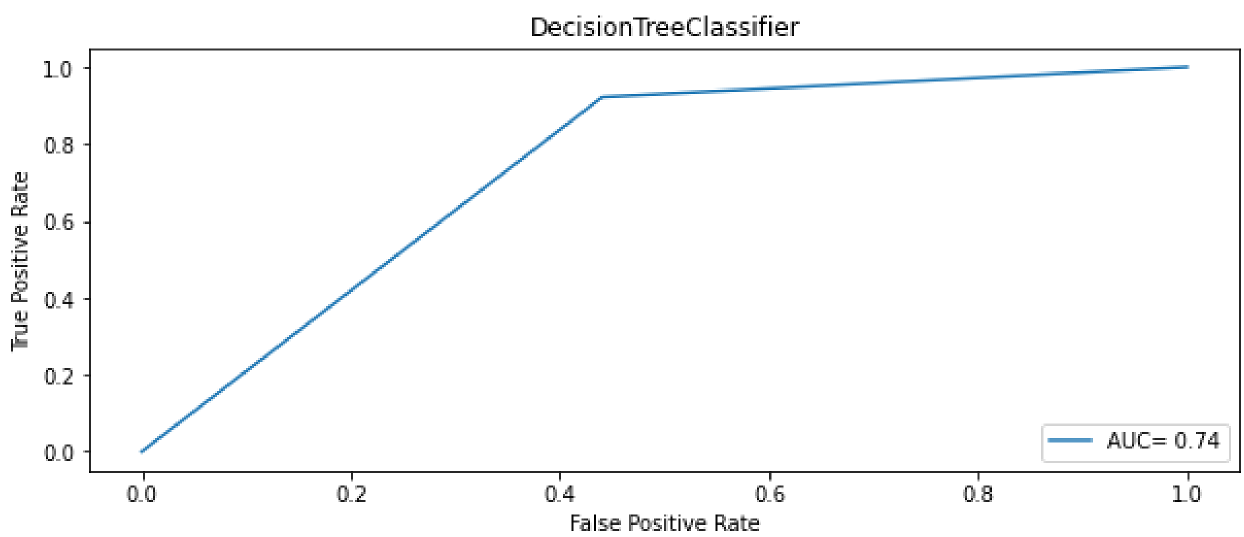


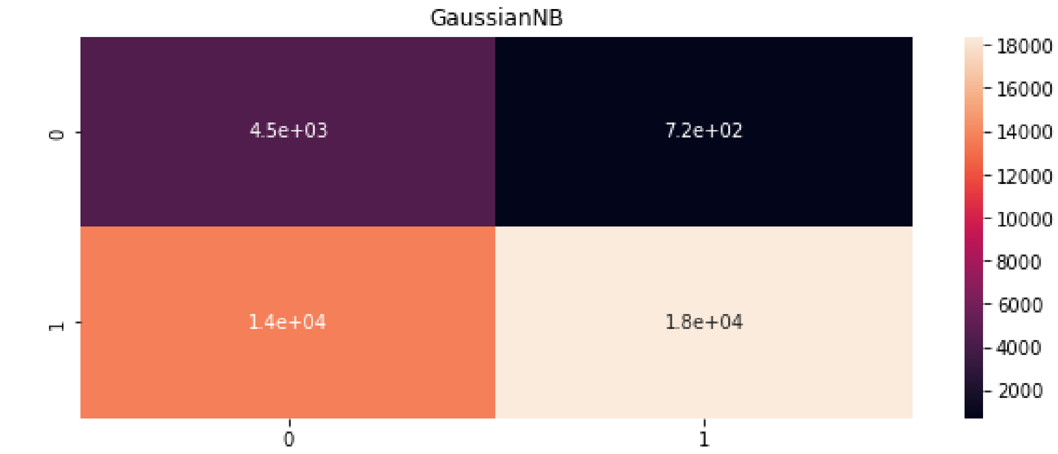


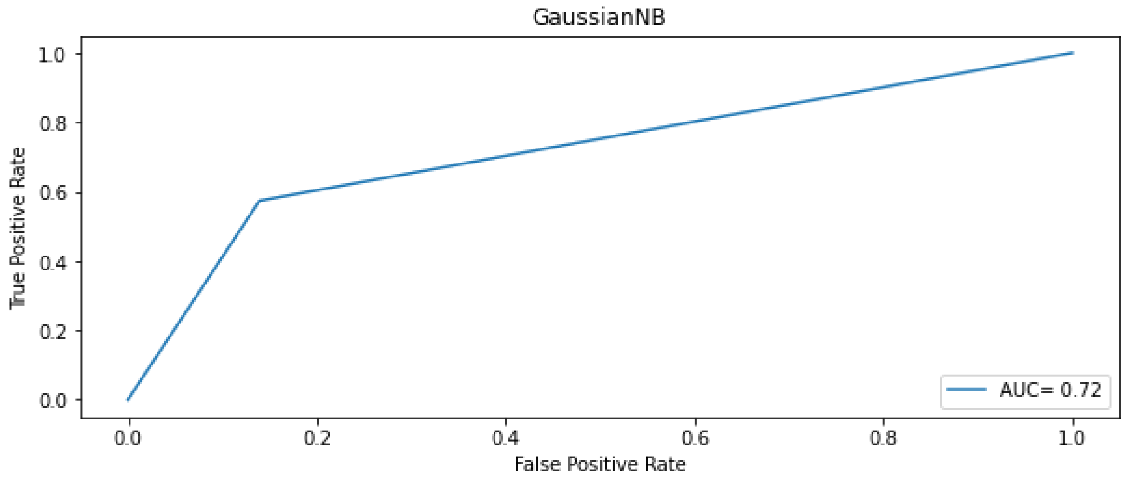

LogisticRegression
2.9e+02
4.9e+03
1.7e+02
3.2e+04
0
1
0
1
30000
25000
20000
15000
10000
5000

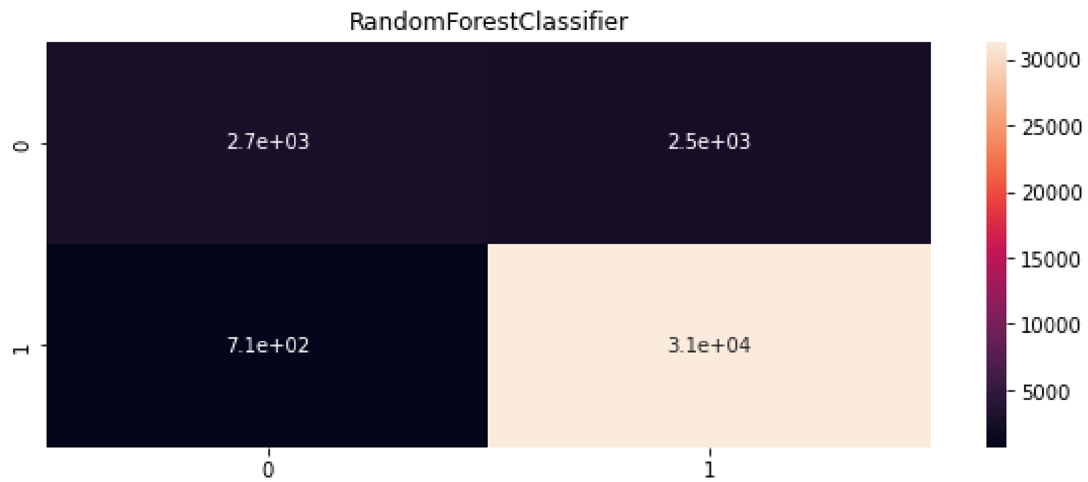

LogisticRegression
True Positive Rate
False Positive Rate
AUC= 0.53
0.0
0.2
0.4
0.6
0.8
1.0

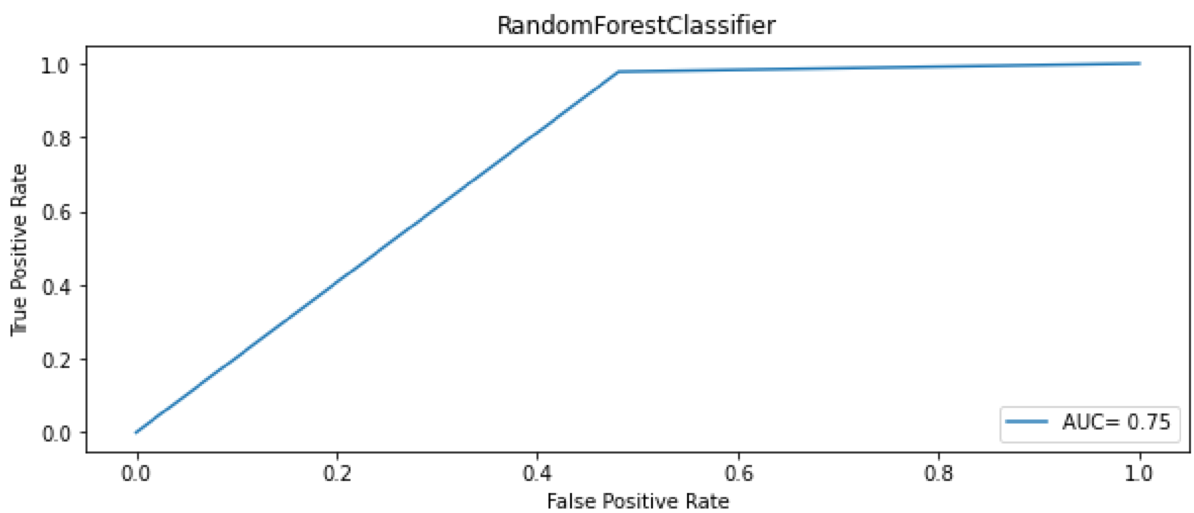

DecisionTreeClassifier
2.9e+03
2.3e+03
2.5e+03
3e+04
0
1
0
1
25000
20000
15000
10000
5000


DecisionTreeClassifier
True Positive Rate
False Positive Rate
AUC= 0.74
0.0
0.2
0.4
0.6
0.8
1.0


GaussianNB
4.5e+03
7.2e+02
1.4e+04
1.8e+04
0
1
0
1
18000
16000
14000
12000
10000
8000
6000
4000
2000


GaussianNB
True Positive Rate
False Positive Rate
AUC= 0.72
0.0
0.2
0.4
0.6
0.8
1.0


RandomForestClassifier
2.7e+03
2.5e+03
7.1e+02
3.1e+04
0
1
0
1
30000
25000
20000
15000
10000
5000


RandomForestClassifier
True Positive Rate
False Positive Rate
AUC= 0.75
0.0
0.2
0.4
0.6
0.8
1.0

```
In [56]: result=pd.DataFrame({'Model': Model, 'Accuracy_score': score, 'Cross_val_score':cvs, 'Roc_auc_curve':rocscore})
result
```

Out[56]:

	Model	Accuracy_score	Cross_val_score	Roc_auc_curve
0	KNeighborsClassifier	86.990255	87.139379	68.671620
1	LogisticRegression	86.423797	86.423650	52.506450
2	DecisionTreeClassifier	87.175495	87.465835	74.082257
3	GaussianNB	61.362721	60.837705	71.720115
4	RandomForestClassifier	91.425273	91.345178	74.846932

• *Interpretation of the Results*

As per our pre- assumption and visualization that we have observed the 209593 records of data of Telecomcustomers and also as per our machine learning model we can understand how a customer is ending up being a defaulter and also how a customer are ending up being a non-defaulterfrom the given factors. The customers who maintain lesser average balance in the account in 30days and 90 days and lesser number of loans taken by a customer ending up being not a defaulter. Customer who onboard with same network for longer period of time and high number of loans taken customer are not being a defaulter. But customer who does high recharge for data in 30 to 90 days, and customer who maintain high balance, who have taken lesser number of loans are ending up being a defaulter.

Post the hyperparameter tuning I have achieved model f1 - score increase from f1: 0.917606 to f1: 0.91829 with accuracy of 92.36%. Thus, we have finally achieved maximum .performed model for the prediction of Defaulters and non-Defaulters for mobile financial services (MFS)

CONCLUSION :

• Key Findings and Conclusions of the Study From our above analysis we understand that which factors are responsible and using which a Microfinance Institution (MFI) can predict that a customer can be a defaulter or not a defaulter basis on which Microfinance Institution (MFI) can give a short-term loan and predictions that could help them in further investment and improvement in selection of customers.

• Limitations of this work and Scope for Future Work Limitations of the study arise from the use of a single dataset obtained from one company. However, the large number of loans and customers considered and generic applicability of credit scoring for mobile credit suggests that the variables and models investigated are relevant for other business applications. Further work may include the use of demographic information which could be obtained from mobile network operators and consideration of additional pay-as-you-go mobile products.

• Learning Outcomes of the Study in respect of Data Science As per our pre- assumption and visualization that we have observed the 209593 records of data of Telecomcustomers and also as per our machine learning model we can understand how a customer is ending up being a defaulter and also how a customer are ending up being a non-defaulterfrom the given factors. The customers who maintain lesser average balance in the account in 30days and 90 days and lesser number of loans taken by a customer ending up being not a defaulter. Customer who onboard with same network for longer period of time and high number of loans taken customer are not being a defaulter. But customer who does high recharge for data in 30 to 90 days, and customer who maintain high balance, who have taken lesser number of loans are ending up being a defaulter.