

Micro Credit Defaulter Project

Submitted by:

Mekhala Misra

**ACKNOWLEDGMENT**

I took help from following websites:

1)Geek for geeks

2)Pandas documentation

3)towardsdatascience.com

4)data trained

**INTRODUCTION**

* Business Problem Framing

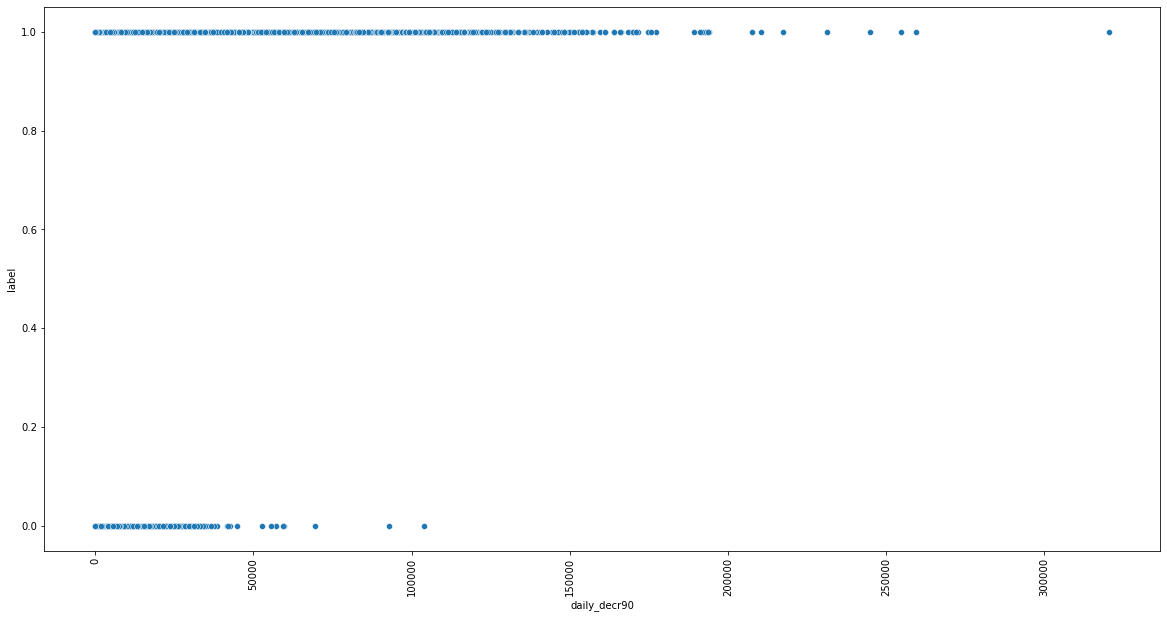
A Microfinance Institution (MFI) is an organization that offers financial services to low-income populations known as Micro Finance Services (MFS). MFS include group loans, agricultural loans, individual business loans and so on.

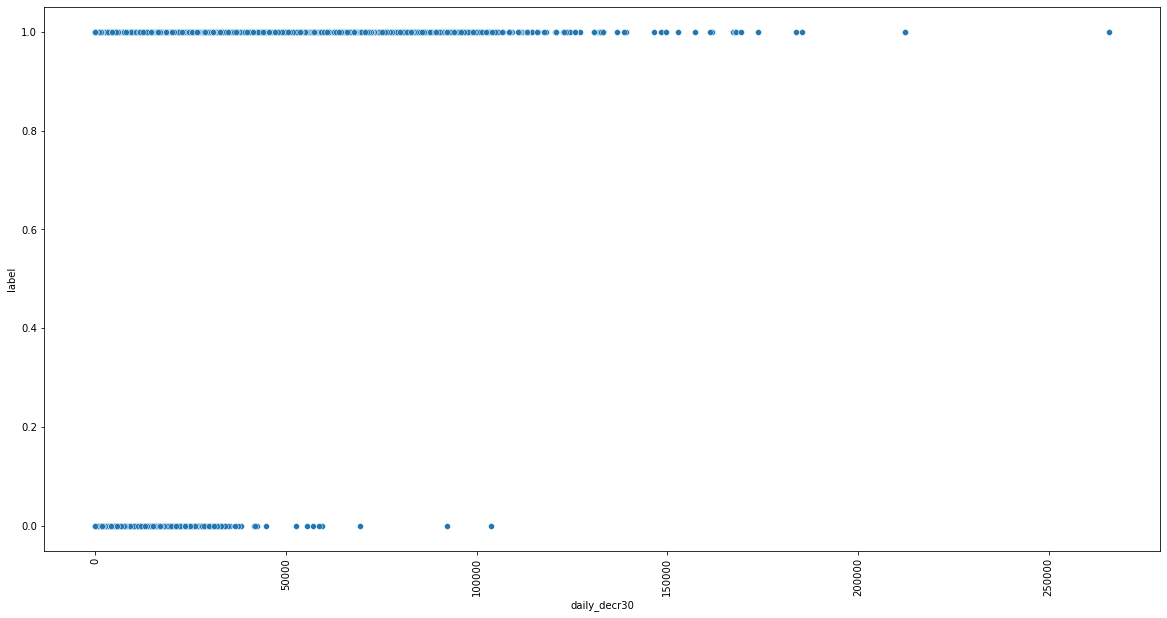
Microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients. In this project we must work for a telecom microfinance client who has collaborated with an MFI to provide micro-credit on mobile balances to be paid 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).

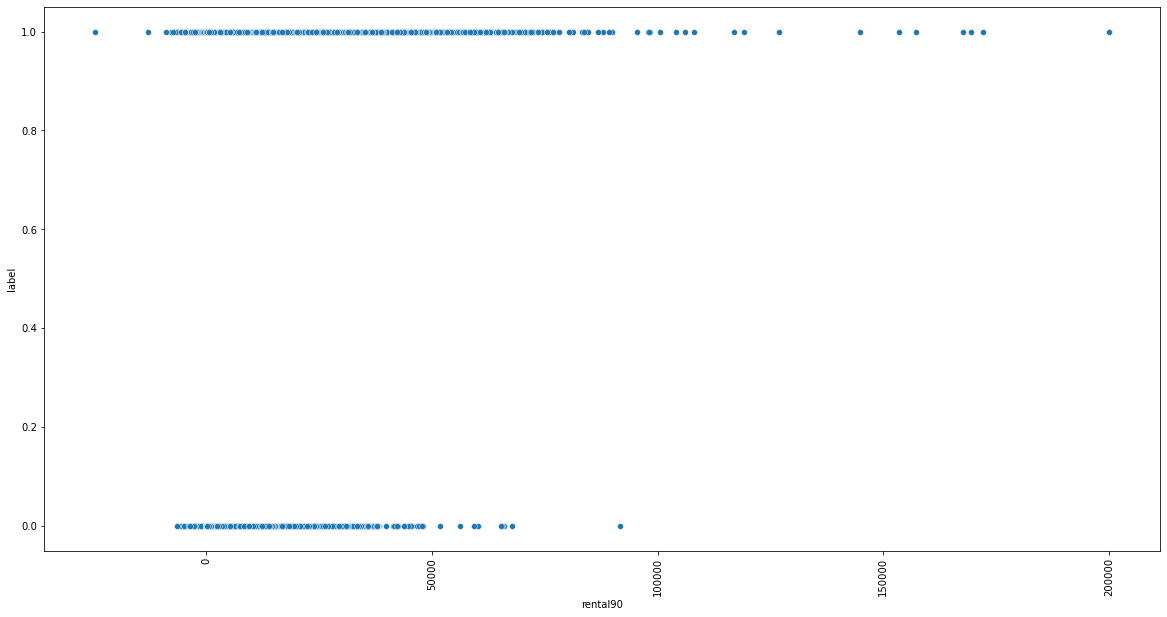
Review of Literature

Using various scatter graph I came to certain conclusion:

1. Here we can see that whether a customer is defaulter or not doesn’t depend on age on network (aon) .
2. We can see that defaulters average daily amount spend from main account is much higher than the non-defaulters averaged over 30 days and averaged over 90 days.



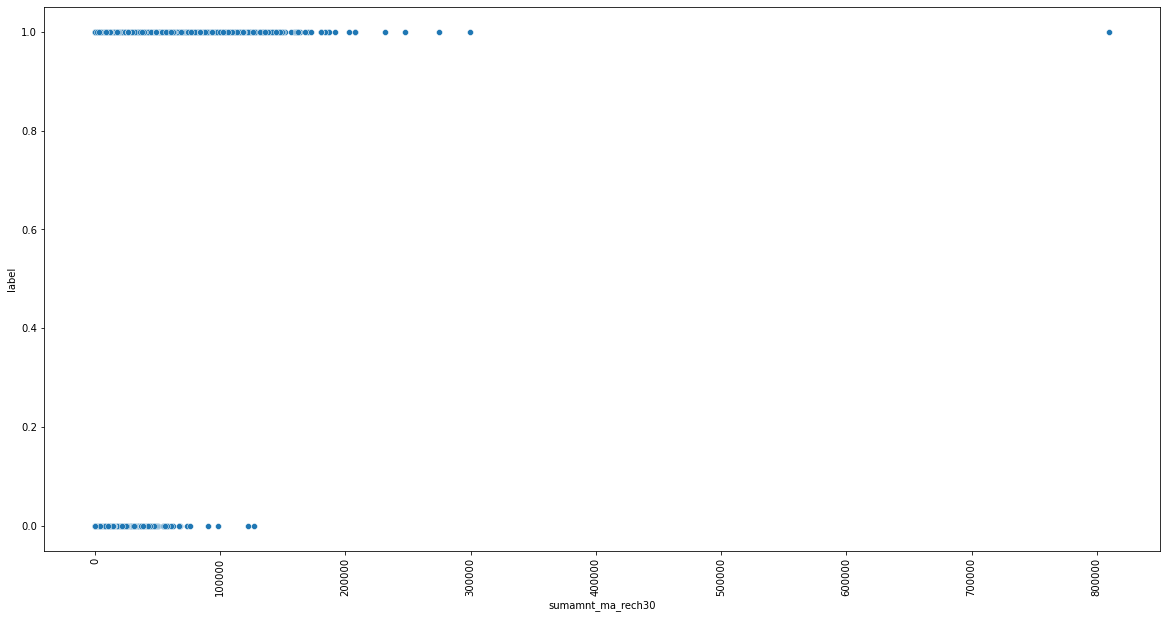




1. Average main account balance over last 30 days and over last 90 days is higher for defaulters than non-defaulters.





1. Defaulters get there main account recharge more number of times than non-defaulters.
2. 

Total amount of recharge in last 30 days is much higher for defaulters than non defaulters.

1. 

Defaulters recharge there data more number of times in 90 days than non defaulters.

1. 

Defaulters take more number of times loan than non defaulters.

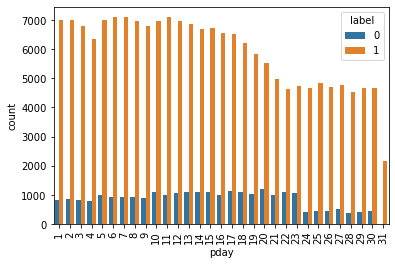
1. 

The maximum loan amount taken in last 30 days is higher for defaulters than non defaulter.

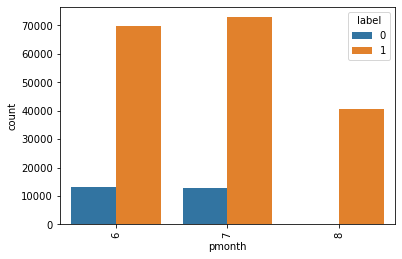
1. 



Loan taken in last 30 days and loan taken in last 90 days, the average days of payback for non-defaulters is mostly within 150 days and for defaulters is 175 days.

1. 

We assume that pday means payment date so we can see from the above graph that customers whose payment date is in the starting of the month are the defaulters.

1. 
2. This dataset contains information for three months only. If a customer takes a loan in the month of august then there is highest probability of being a defaulter.

* Motivation for the Problem Undertaken

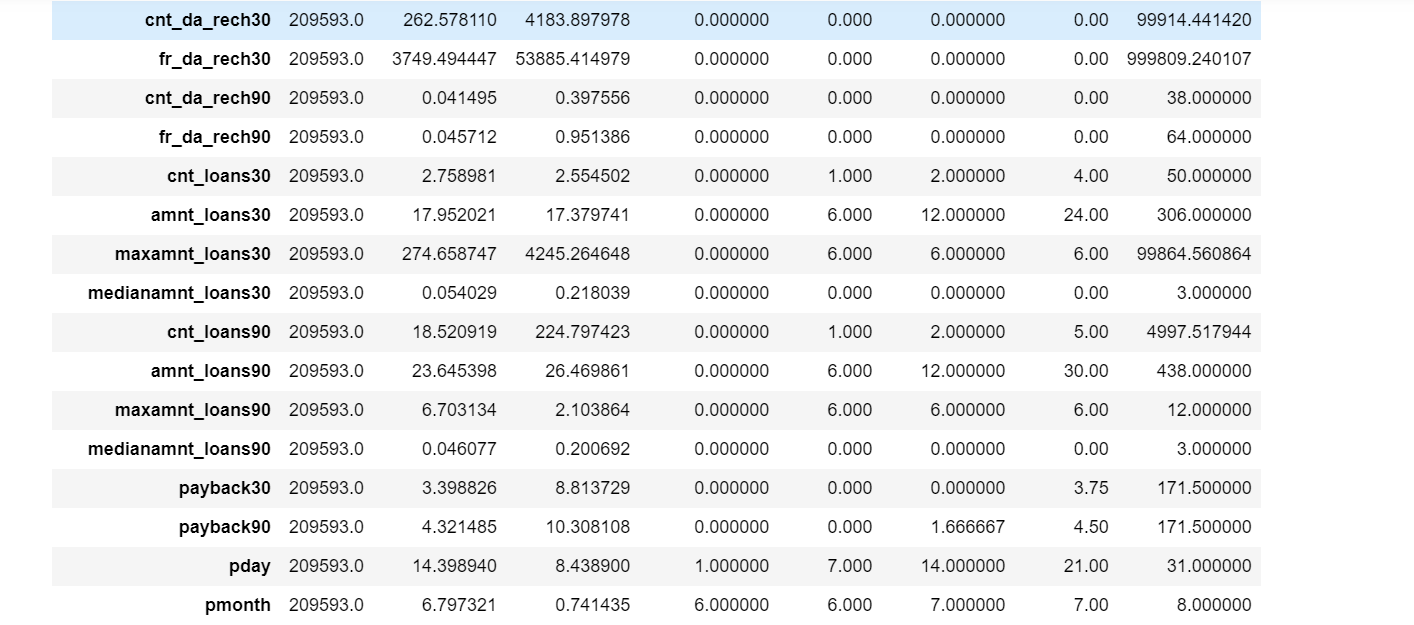
In todays time when there are so many kinds of frauds including cyber frauds, it is very important to predict whether a customer will be fraudster or a good payer of loan amount and also within given time duaration.

**Analytical Problem Framing**

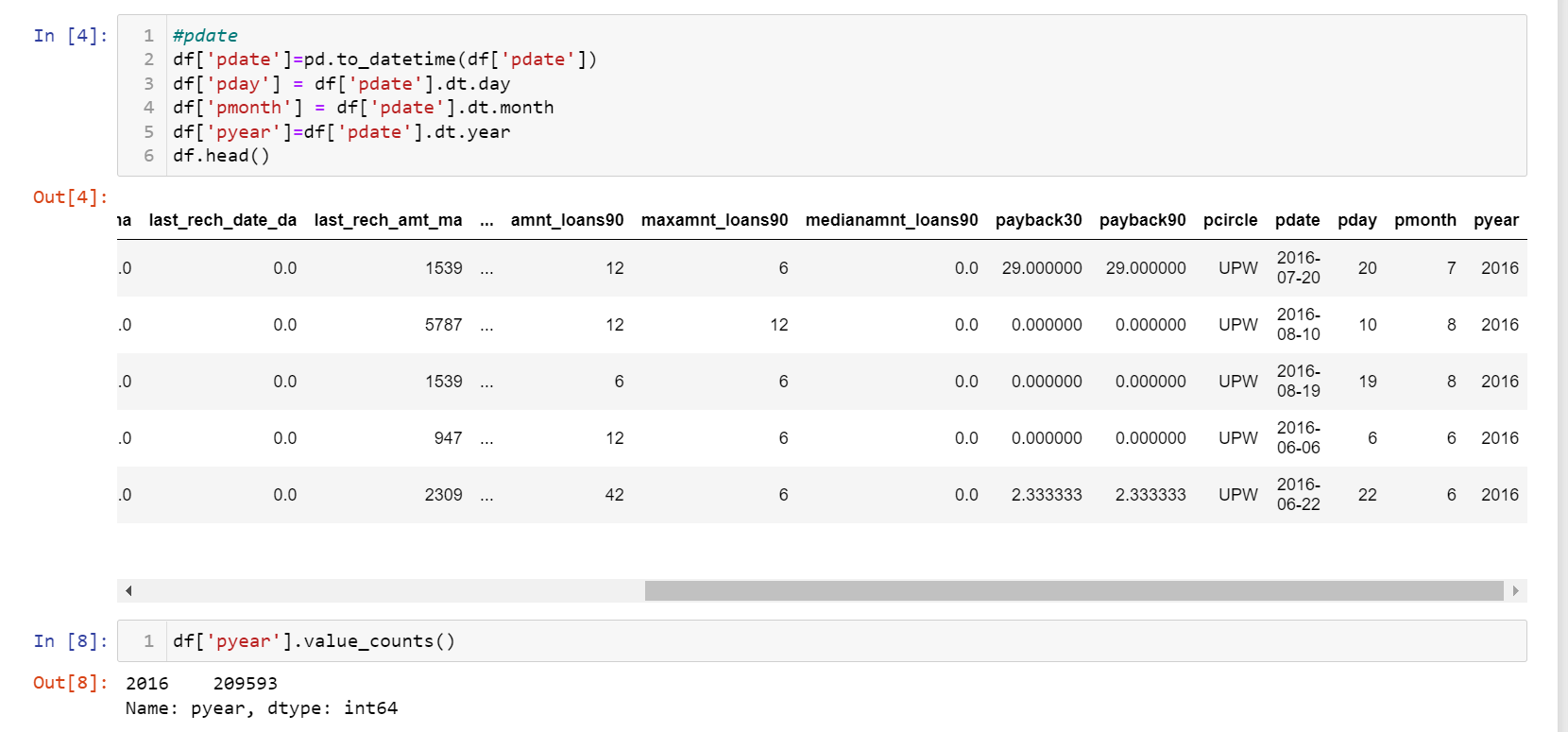
* Mathematical/ Analytical Modelling of the Problem

1. The dataset had no null values but **there were many unrealistic negative values** in many features.
2. When we described the data we found that aon,daily\_decr30,daily\_decr90, rental30,rental90,last\_rech\_date\_ma,last\_rech\_date\_da,medianmarechprebal30,medianmarechprebal90 minimum values are in negative. aon, daily\_decr30,daily\_decr90 values cannot be negative as age on network can either br 0 or any positive number, age can never be negative. Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah) i.e. daily\_decr30 can also never be negative as its an amount spent it can either be 0 or positive. Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah) i.e. daily\_decr90 can also never be negative as its an amount spent it can either be 0 or positive. rental30 and rental 90 being the average account balance over 30 days and 90 days respectively can be considered as negative. Number of days till last recharge of main account i.e. last\_rech\_date\_ma can never be in negative so we would replace negative values by 0. Number of days till last recharge of data i.e. last\_rech\_date\_da can never be in negative so we would replace negative values by 0. medianmarechprebal30 and medianmarechprebal90 can be negative as this is the medium of account balance which could be in negative.

****

****

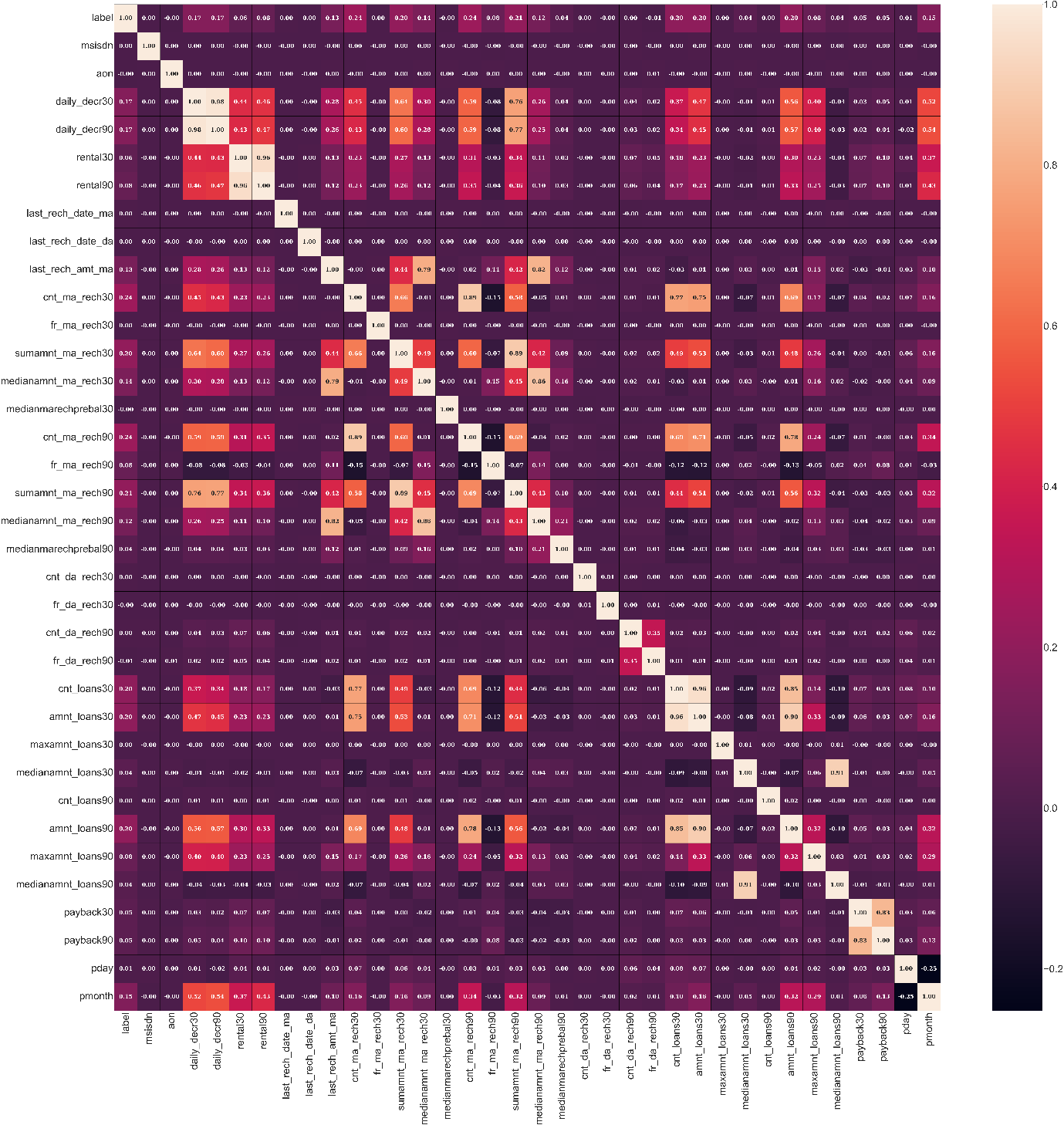
1. **Feature Engineering-**Initially we had a feature named pdate so to better analyse this feature we splitted pdate into pyear, pmonth and pday. So we found that all the data was of year 2016 so we dropped pdate and pyear column and analysed rest.

****

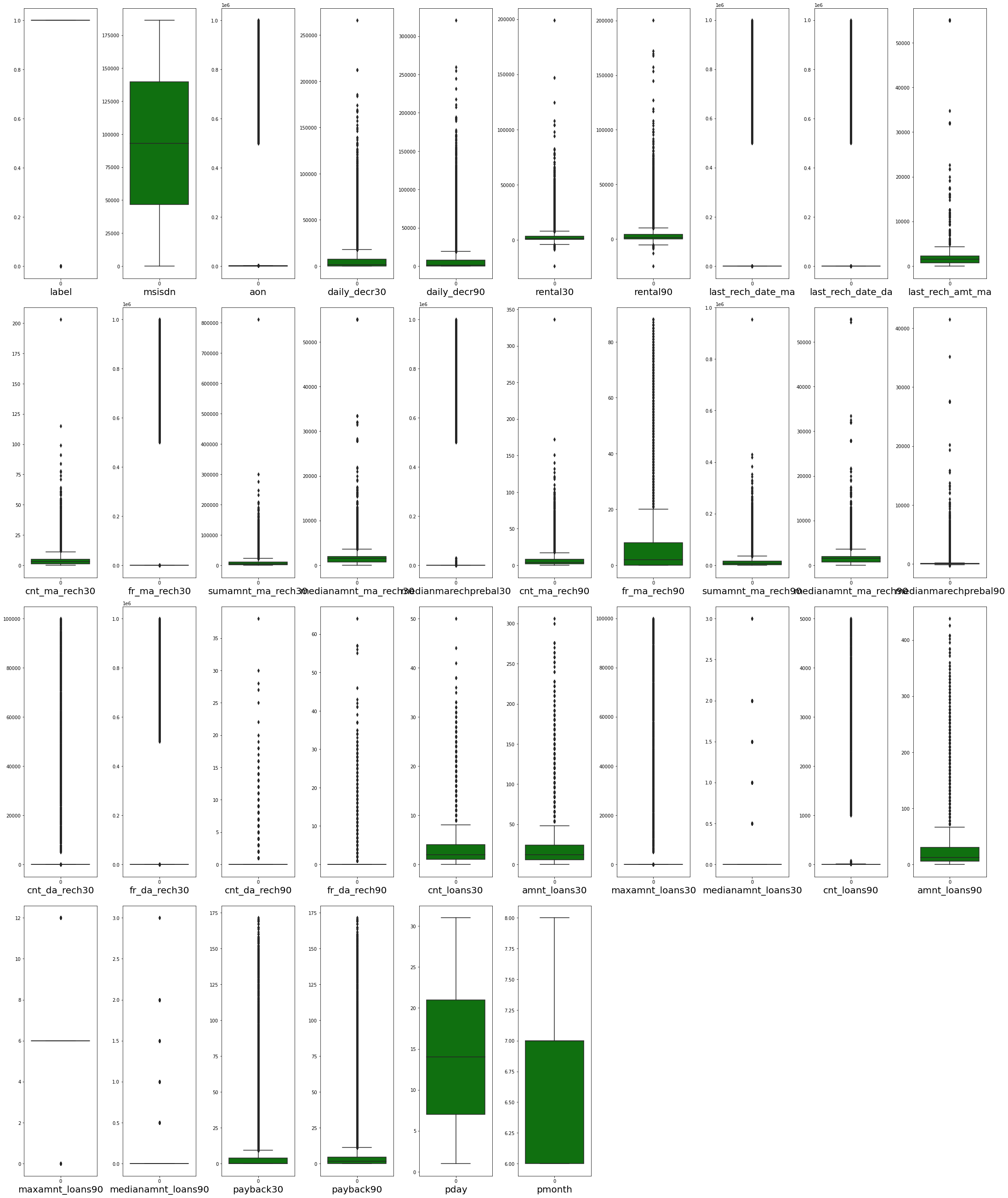
1. There was a feature named pcircle, we found out that entire dataset has the same value for pcircle so it seems that all the mobile phone has the same telecom circle. Thus no use of this feature so we droped it as well.

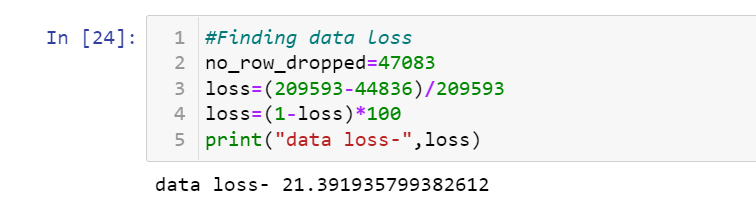
****

1. After creating heatmap as well as after finding vif score of features we found that there was a problem of multicollinearity as many features vif score was greater than 10. So all those features were deleted that caused multicollinearity.

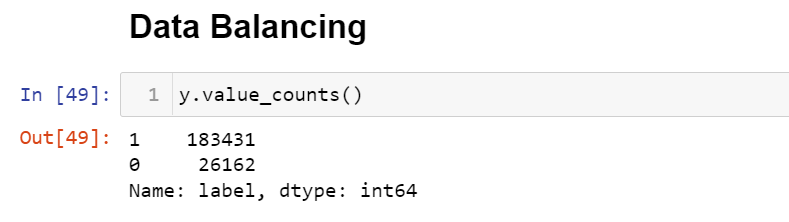
****

1. Lot many outliers are present in the data set but if we consider deleting them then there will be 21% data loss so deleting outliers is not a good idea. So to overcome this we applied power transformation technique.

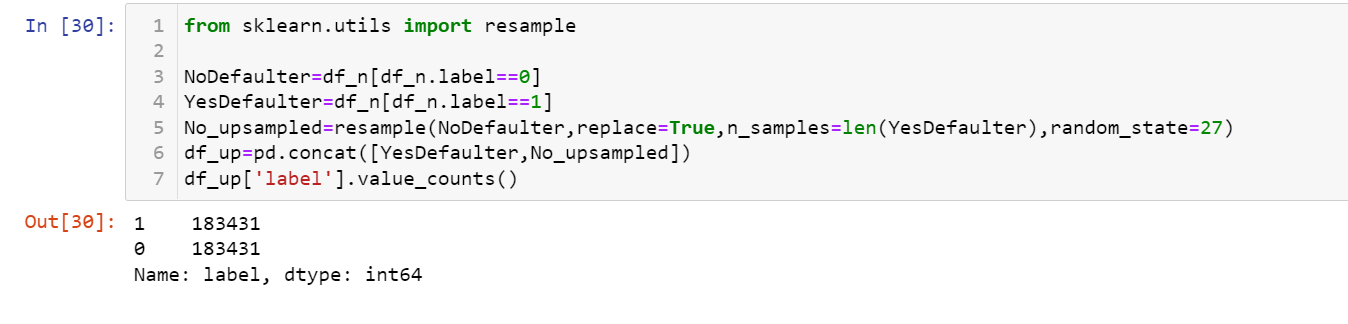
****

****

1. The applied power transform method will not only deal with outliers but also remove skewness. Due to presence of outliers the data is also skewed a lot.
2. We can observe that data is not at all balanced so we balanced it by resampling and up sampling it.

****

The screenshot after data is balanced:



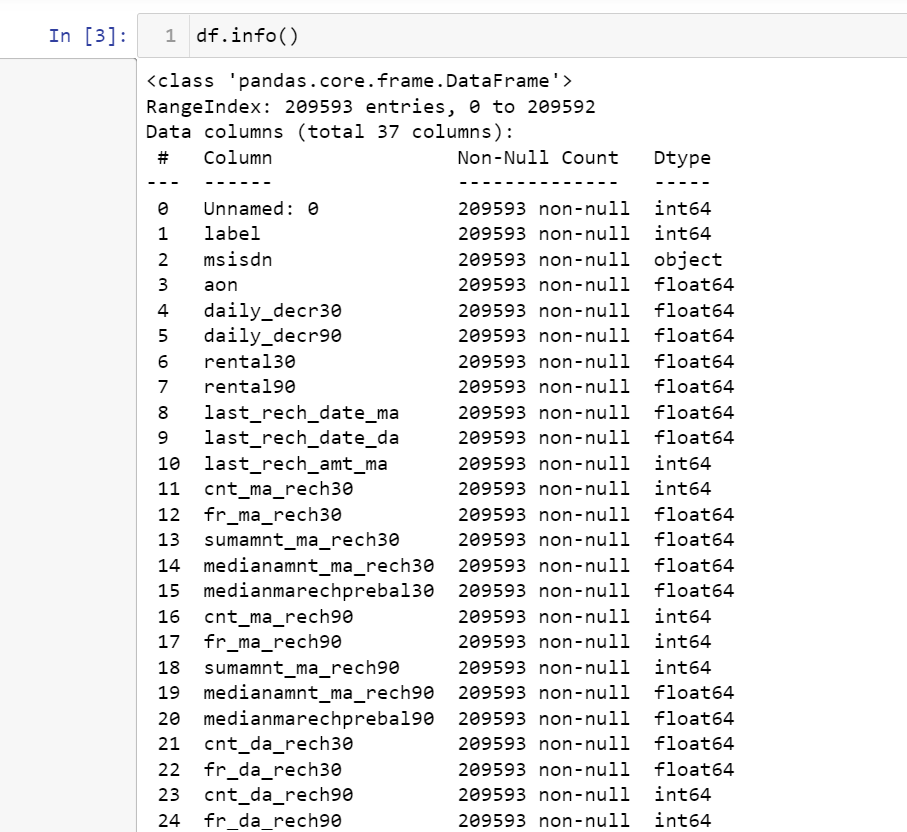
1. Data was then normally distributed using standardisation technique.
2. We split the data into training set and testing set using train test split method.
3. On this train and test data we applied various models: logistic regression, decision tree classifier, random forest classifier , Knn classifier and XGBoost Classifier.
4. The best performing model is random forest as its confusion matrix, auc-roc curve and recall, f1-score is the best among all. Since the model is already giving its best accuracy so we won’t apply hyperparameter tuning as it’s a very big data so applying hyperparameter tuning is very time consuming and the scope of accuracy improvement is very less.
5. So, we saved our previous random forest classifier model.

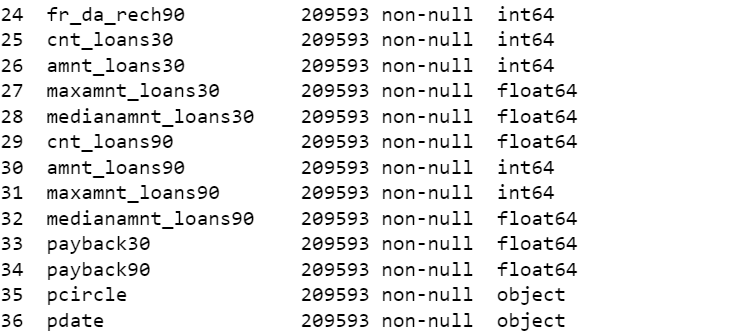
* Data Sources and their formats

Data set is provided by the FlipRobo technologies and it has 209593 rows and 33 columns.

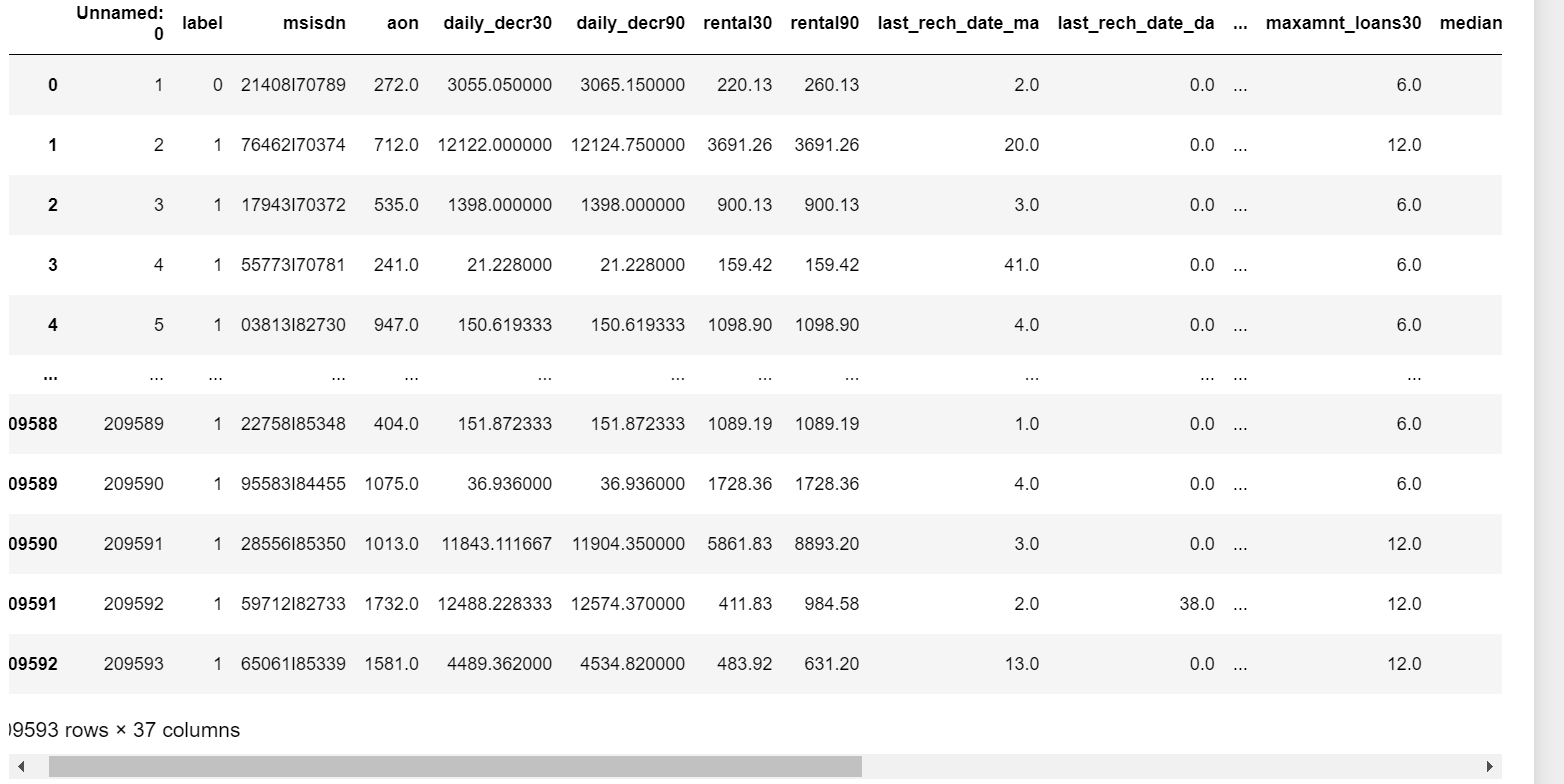
****

Following are the columns present and there respective data types:





Here is the glimpse of data:



* Data Pre-processing Done
* The dataset had no null values but **there were many unrealistic negative values** in many features.
* When we described the data we found that aon,daily\_decr30,daily\_decr90, rental30,rental90,last\_rech\_date\_ma,last\_rech\_date\_da,medianmarechprebal30,medianmarechprebal90 minimum values are in negative. aon, daily\_decr30,daily\_decr90 values cannot be negative as age on network can either br 0 or any positive number, age can never be negative. Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah) i.e. daily\_decr30 can also never be negative as its an amount spent it can either be 0 or positive. Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah) i.e. daily\_decr90 can also never be negative as its an amount spent it can either be 0 or positive. rental30 and rental 90 being the average account balance over 30 days and 90 days respectively can be considered as negative. Number of days till last recharge of main account i.e. last\_rech\_date\_ma can never be in negative so we would replace negative values by 0. Number of days till last recharge of data i.e. last\_rech\_date\_da can never be in negative so we would replace negative values by 0. medianmarechprebal30 and medianmarechprebal90 can be negative as this is the medium of account balance which could be in negative.
* **Feature Engineering-**Initially we had a feature named pdate so to better analyse this feature we splitted pdate into pyear, pmonth and pday. So we found that all the data was of year 2016 so we dropped pdate and pyear column and analysed rest.
* There was a feature named pcircle, we found out that entire dataset has the same value for pcircle so it seems that all the mobile phone has the same telecom circle. Thus no use of this feature so we droped it as well.
* There is a feature named msisdn i.e. mobile number of a user which is of object type so we encoded it using ordinal encoder.
* After creating heatmap as well as after finding vif score of features we found that there was a problem of multicollinearity as many features vif score was greater than 10. So all those features were deleted that caused multicollinearity.
* Lot many outliers are present in the data set but if we consider deleting them then there will be 21% data loss so deleting outliers is not a good idea. So to overcome this we applied power transformation technique.
* The applied power transform method will not only deal with outliers but also remove skewness. Due to presence of outliers the data is also skewed a lot.
* We can observe that data is not at all balanced so we balanced it by resampling and up sampling it.
* Data was then normally distributed using standardisation technique.
* Hardware and Software Requirements and Tools Used

We imported following packages:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import OrdinalEncoder

from sklearn.preprocessing import LabelEncoder

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

from scipy.stats import zscore,boxcox

from sklearn.model\_selection import GridSearchCV

from sklearn.preprocessing import StandardScaler

from sklearn.preprocessing import power\_transform

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import cross\_val\_score

from sklearn import metrics

from sklearn.metrics import roc\_curve,auc,classification\_report,accuracy\_score

We also installed XGBoost Classifier.

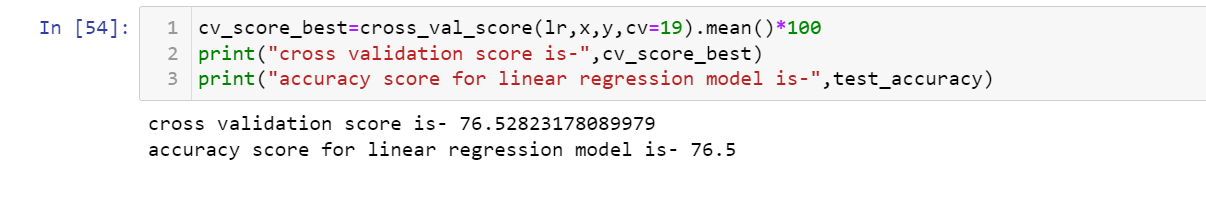
**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)
* Since the data has lot many outliers although we used power transformation to lower the affect of outliers still there is some existence of outliers. So, for better prediction I mostly used tree based algorithms because they are less sensitive to outliers.
* Testing of Identified Approaches (Algorithms)
* We applied Logistic regression, Decision Tree classifier,random forest classifier, knn classifier and XGBoost algorithms on the clean data and got 76.5 %,96.3%, 98%,89.6% and 88.9% accuracy respectively.
* Depending on the model accuracy, confusion matrix, auc-roc curve and classification report we opted random forest classifier.
* Since random forest classifier is giving best accuracy already so I did not apply hyper parameter tuning on dataset as the dataset is very large and time consuming and scope of accuracy improvement is not much high. So, we saved our previous random forest classifier model.
* **Run and Evaluate selected models**

1. **Logistic Regression**

We applied this algorithm and found the train accuracy to be 76.5% and test accuracy to be 76.5%.

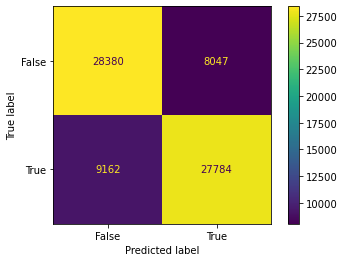




The test accuracy for logistic regression is 76.5% and its cv score is 76.5% thus making us sure that the model is not overfitted.

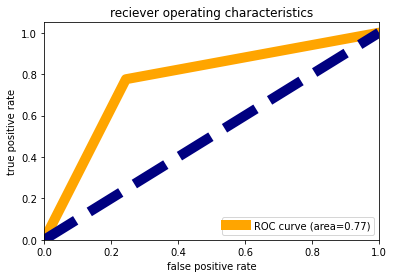
We also tested this models on other metrics:

1. Confusion matrix:



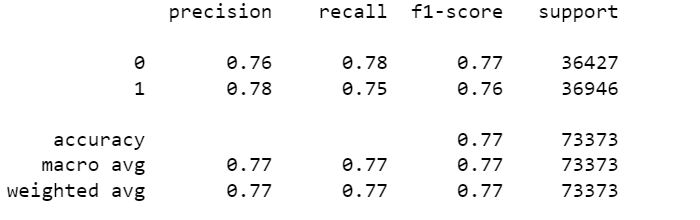
Model is good in predicting false classes but bad in predicting true as well as false classes. Although we can also see that the samples for true classes and samples for false classes are also equal.

1. AUC-ROC Curve:



We observed that the area under the curve is 77% that means 77% times model is predicting accurately and rest all other time it gives wrong prediction.

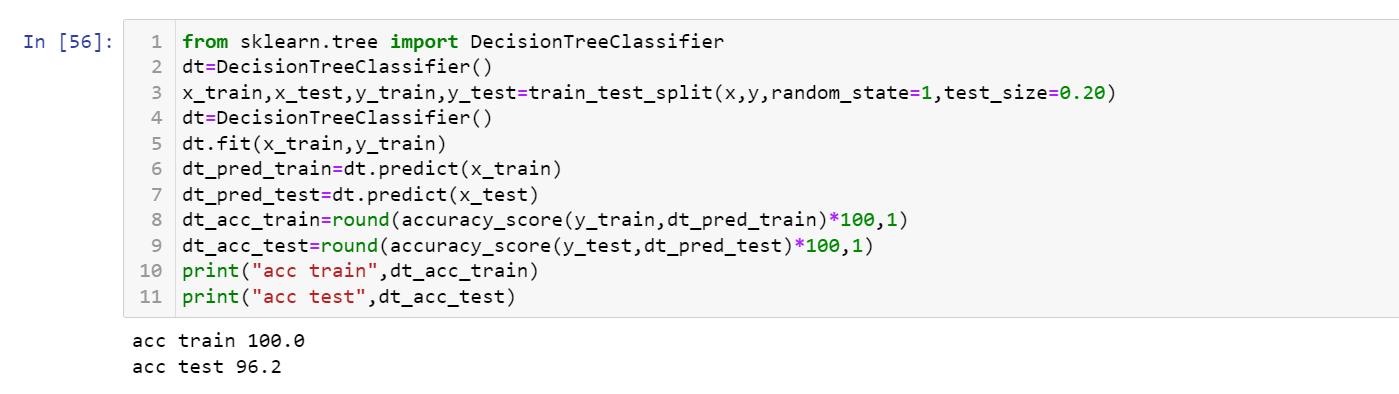
1. Classification Report:

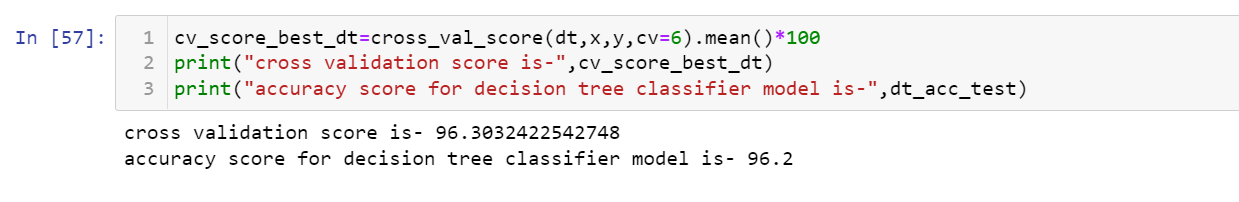


Model is poor in recalling class 1 i.e., True than class 0 i.e. False. F1-score of class 0 is higher than that of class 1.

1. **Decision Tree Classifier**

We applied this algorithm and found the train accuracy to be 100% and test accuracy to be 96.2%.

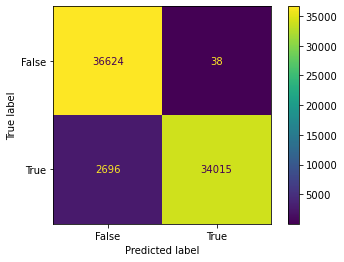




The test accuracy for logistic regression is 96.2% and its cv score is 96.3% thus making us sure that the model is not overfitted, although decision tree classifier is prone to overfitting.

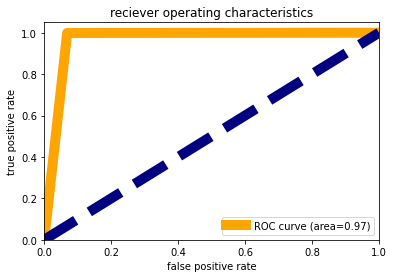
We also tested this model on other metrics:

1. Confusion matrix:



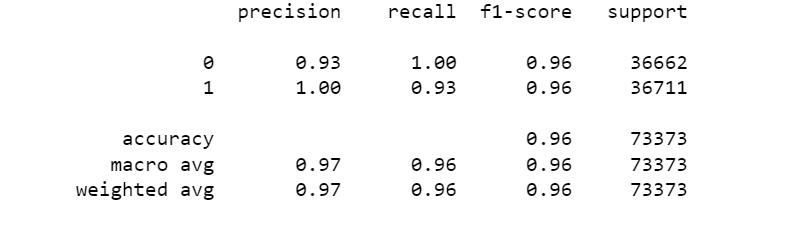
Model is good in predicting false classes but bad in predicting true as well as false classes. Although we can also see that the samples for true classes and samples for false classes are also equal.

1. AUC-ROC Curve:



We observed that the area under the curve is 97% that means 97% times model is predicting accurately and rest all other time it gives wrong prediction. This accuracy is a good one lets check on other metrics as well.

1. Classification Report:

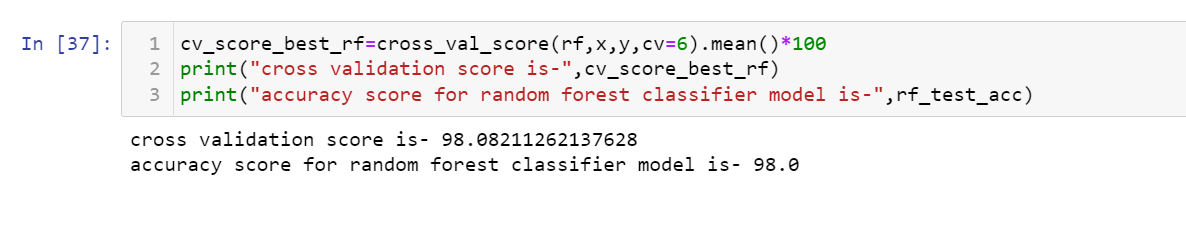


Model is poor in recalling class 1 i.e., True than class 0 i.e. False, although precision for class 1 is higher than class 0. F1-score of class 0 is similar to that of class 1.

1. **Random Forest Classifier**

We applied this algorithm and found the train accuracy to be 100% and test accuracy to be 98%.

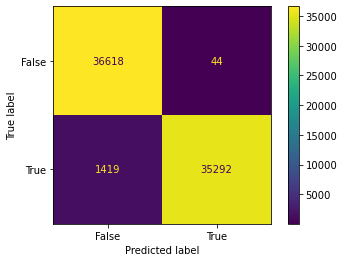




The test accuracy for logistic regression is 98% and its cv score is 98% thus making us sure that the model is not overfitted, although decision tree classifier is prone to overfitting.

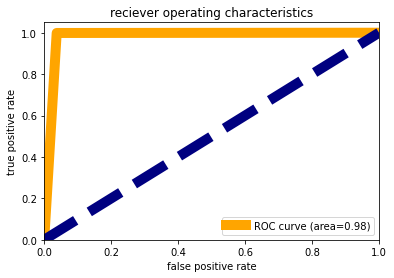
We also tested this model on other metrics:

1. Confusion matrix:



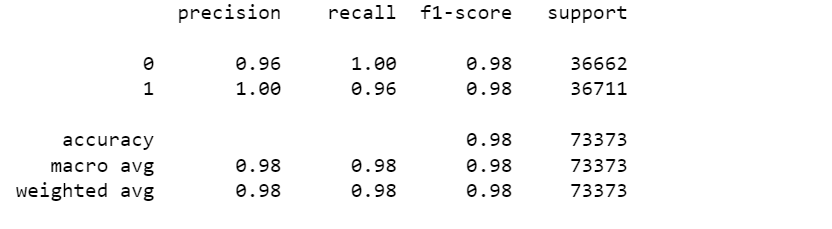
Model is good in predicting false classes but bad in predicting true as well as false classes. Although we can also see that the samples for true classes and samples for false classes are also equal.

1. AUC-ROC Curve:



We observed that the area under the curve is 98% that means 98% times model is predicting accurately and rest all other time it gives wrong prediction. This accuracy is a good one let’s check on other metrics as well.

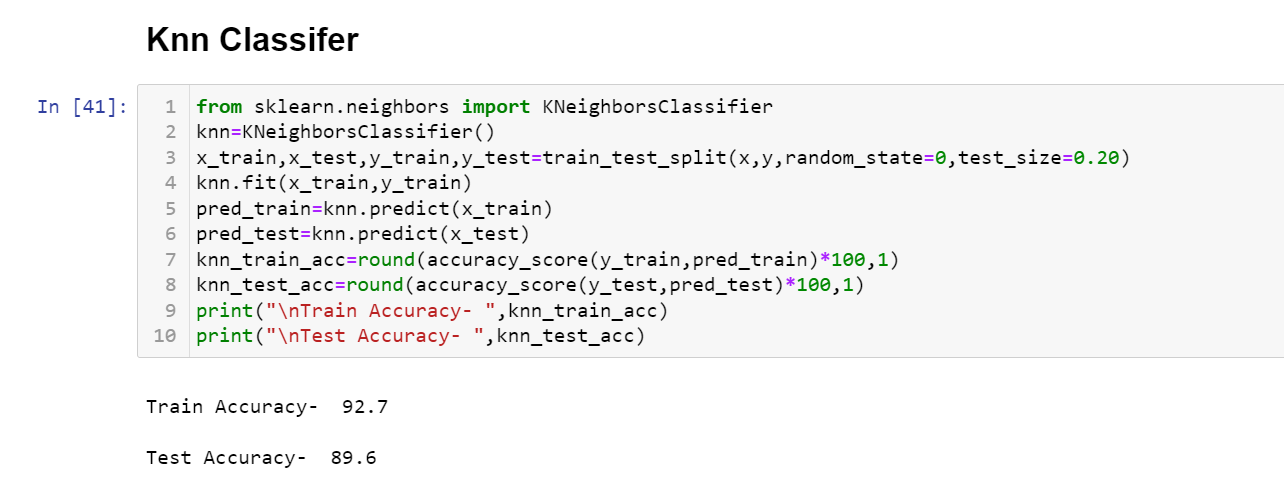
1. Classification Report:

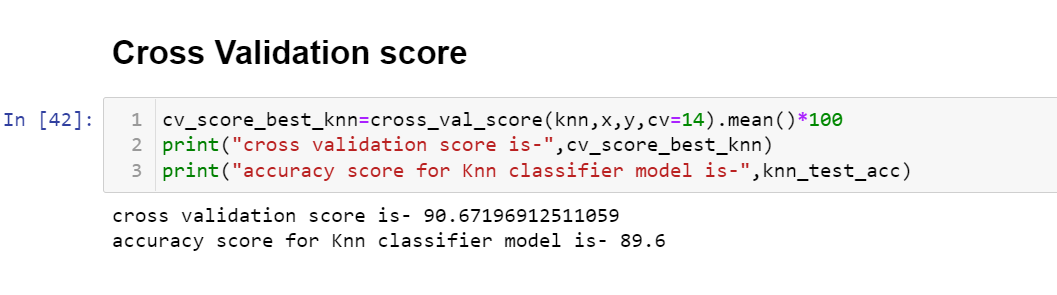


Model is poor in recalling class 1 i.e., True than class 0 i.e. False, although precision for class 1 is higher than class 0. F1-score of class 0 is similar to that of class 1.

1. **Knn Classifier**

We applied this algorithm and found the train accuracy to be 92.7% and test accuracy to be 89.6%.

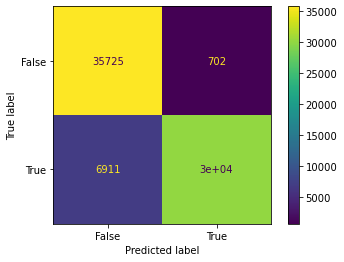




The test accuracy for logistic regression is 89.6% and its cv score is 90.6% thus making us sure that the model is not overfitted, although decision tree classifier is prone to overfitting.

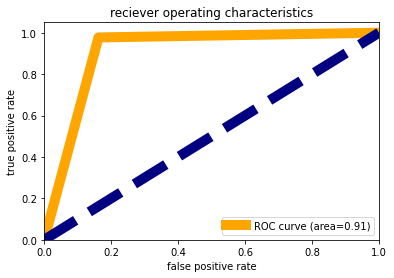
We also tested this model on other metrics:

1. Confusion matrix:



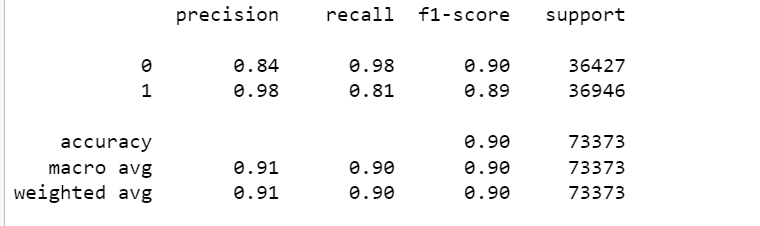
Model is good in predicting false classes but bad in predicting true as well as false classes. Although we can also see that the samples for true classes and samples for false classes are also equal.

1. AUC-ROC Curve:



We observed that the area under the curve is 91% that means 91% times model is predicting accurately and rest all other time it gives wrong prediction. This accuracy is a good one let’s check on other metrics as well.

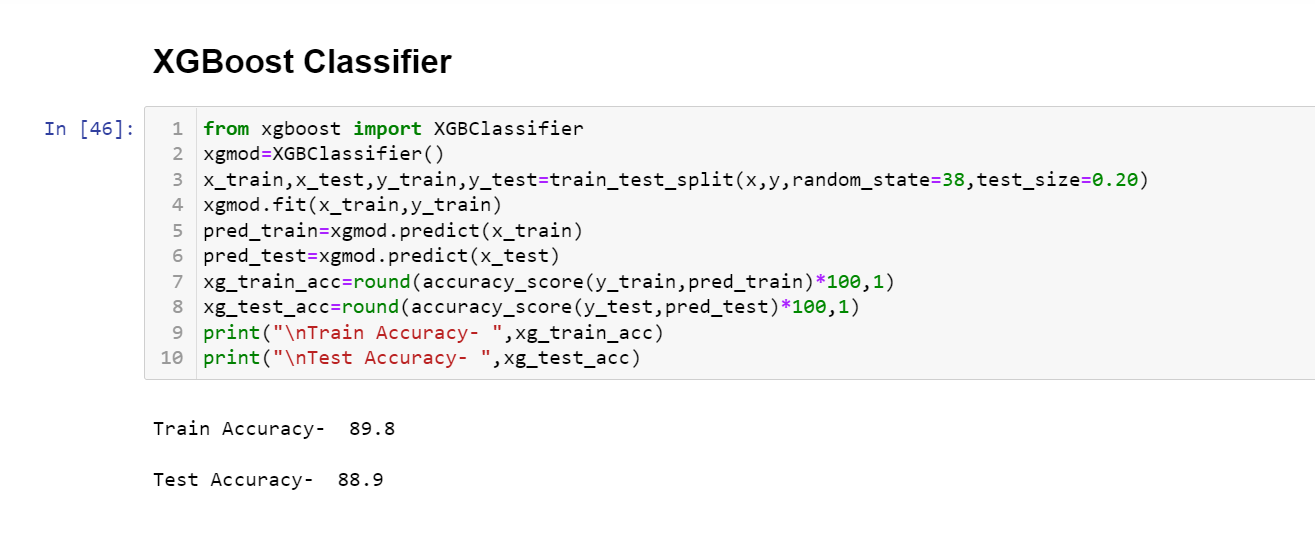
1. Classification Report:

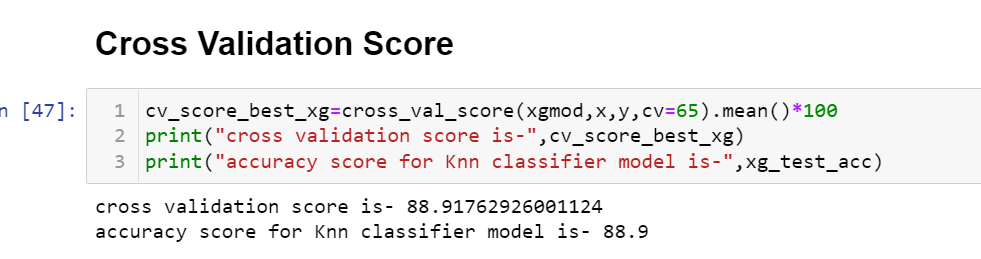


Model is poor in recalling class 1 i.e., True than class 0 i.e. False, although precision for class 1 is higher than class 0. F1-score of class 0 is similar to that of class 1.

1. **XGBoost Classifier**

We applied this algorithm and found the train accuracy to be 89.8% and test accuracy to be 88.9%.

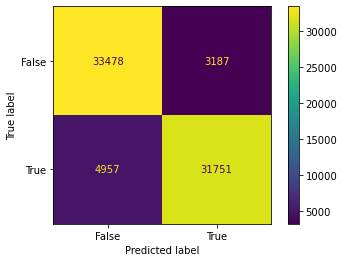




The test accuracy for logistic regression is 88.9% and its cv score is 88.9% thus making us sure that the model is not overfitted, although decision tree classifier is prone to overfitting.

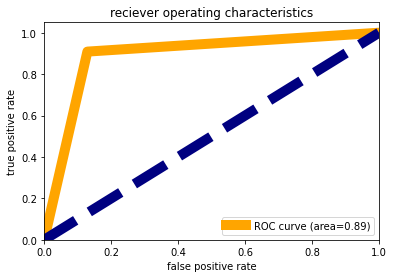
We also tested this model on other metrics:

1. Confusion matrix:



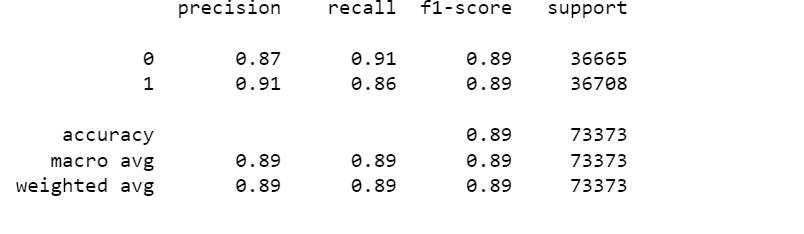
Model is good in predicting false classes but bad in predicting true as well as false classes. Although we can also see that the samples for true classes and samples for false classes are also equal.

1. AUC-ROC Curve:



We observed that the area under the curve is 89% that means 89% times model is predicting accurately and rest all other time it gives wrong prediction. This accuracy is a good one let’s check on other metrics as well.

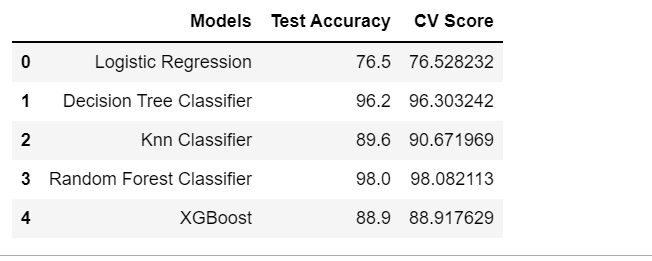
1. Classification Report:



Model is poor in recalling class 1 i.e., True than class 0 i.e. False, although precision for class 1 is higher than class 0. F1-score of class 0 is similar to that of class 1.

**Conclusion for Model**

There are three models with least difference between accuracy and cv score but the best performing model is random forest as its confusion matrix, auc-roc curve and recall, f1-score is the best among all. Since the model is already giving its best accuracy so we wont apply hyperparameter tuning as it’s a very big data so we will save our random forest model.



***CONCLUSION***

1. Customer paying loan back within 150 days are mostly non-defaulters.
2. Customers taking higher amount of loan and higher frequency of recharge are mostly defaulters.
3. Average main account balance is higher for defaulters than non-defaulters.
4. Average daily amount spend from main account is much higher for defaulters than non-defaulters.