**Project Report   
on   
Micro Credit Loan Defaulters**

Submitted by:

Saikrishna Padamatinti

**Problem Statement:**

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.

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).

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.

The goal here is to build a model 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 payed i.e. Non- defaulter, while, Label ‘0’ indicates that the loan has not been payed i.e. defaulter.

**Dataset Description**:

1. label: Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan {1: success, 0: failure}

2. msisdn: mobile number of user

3. aon: age on cellular network in days

4. daily\_decr30: Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)

5. daily\_decr90: Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)

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 (in Indonesian Rupiah)

11. cnt\_ma\_rech30: Number of times main account got recharged in last 30 days

12. fr\_ma\_rech30: Frequency of main account recharged in last 30 days

13. sumamnt\_ma\_rech30: Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)

14. medianamnt\_ma\_rech30: Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)

15. medianmarechprebal30: Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)

16. cnt\_ma\_rech90: Number of times main account got recharged in last 90 days

17. fr\_ma\_rech90: Frequency of main account recharged in last 90 days

18. sumamnt\_ma\_rech90: Total amount of recharge in main account over last 90 days (in Indian Rupee)

19. medianamnt\_ma\_rech90: Median of amount of recharges done in main account over last 90 days at user level (in Indian Rupee)

20. medianmarechprebal90: Median of main account balance just before recharge in last 90 days at user level (in Indian Rupee)

21. cnt\_da\_rech30: Number of times data account got recharged in last 30 days

22. fr\_da\_rech30: Frequency of data account recharged in last 30 days

23. cnt\_da\_rech90: Number of times data account got recharged in last 90 days

24. fr\_da\_rech90: Frequency of data account recharged 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: maximum amount of loan taken by the user 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

**-**

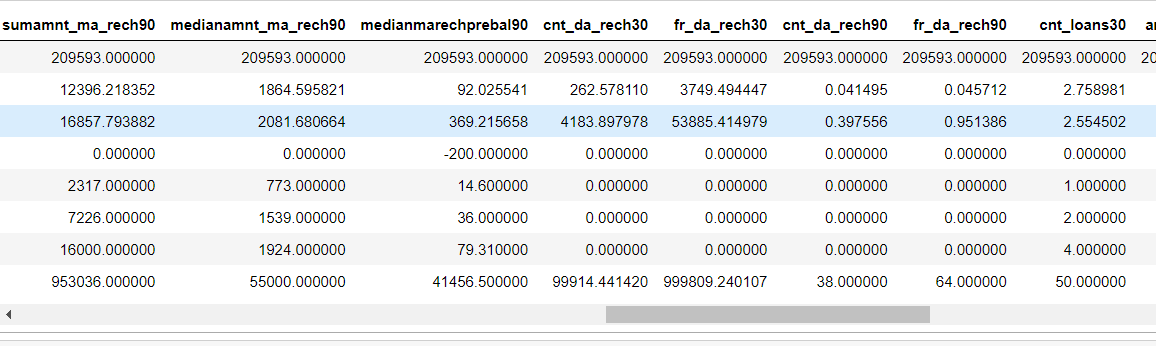
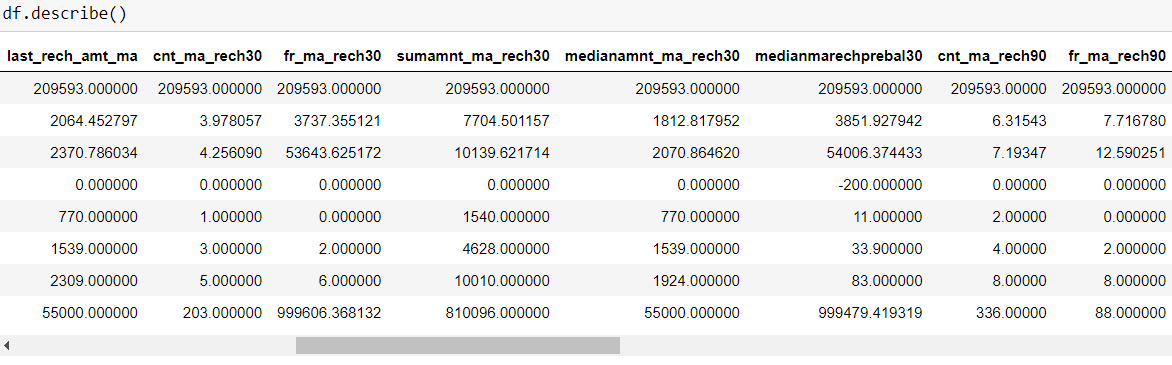
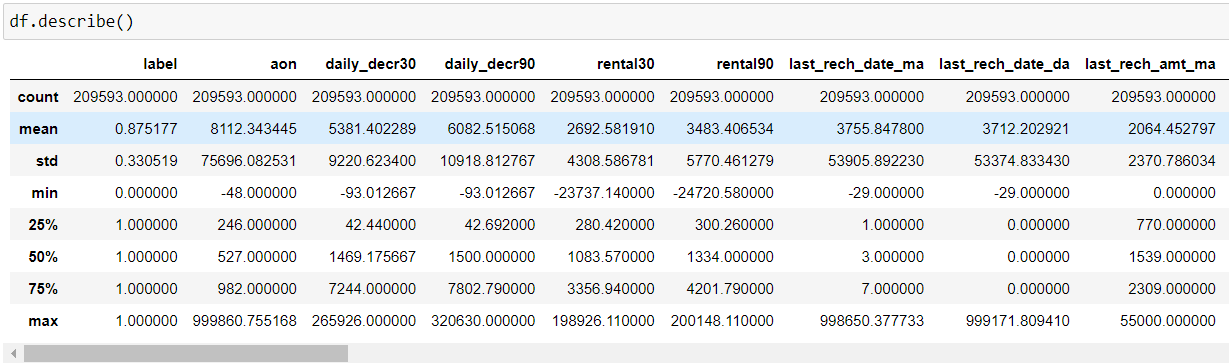
**Exploratory Data Analysis**

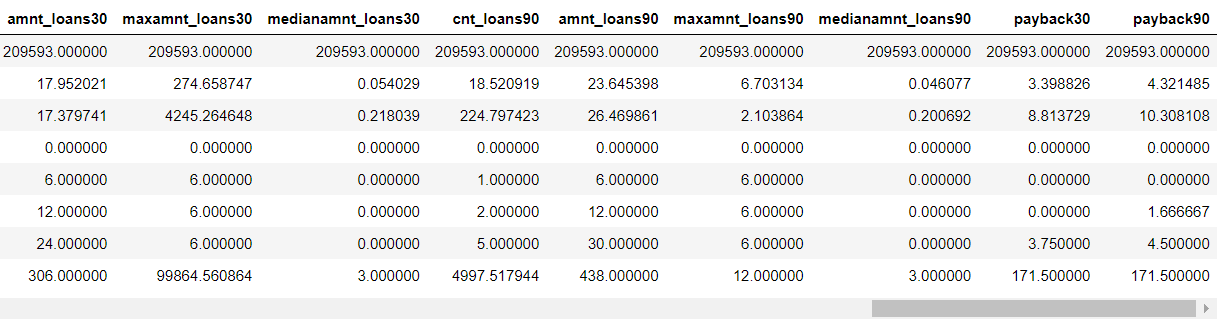
****

The size of the dataset is 209593 rows with 36 columns.

The target feature “Label” is categorical. Hence it is classification problem.

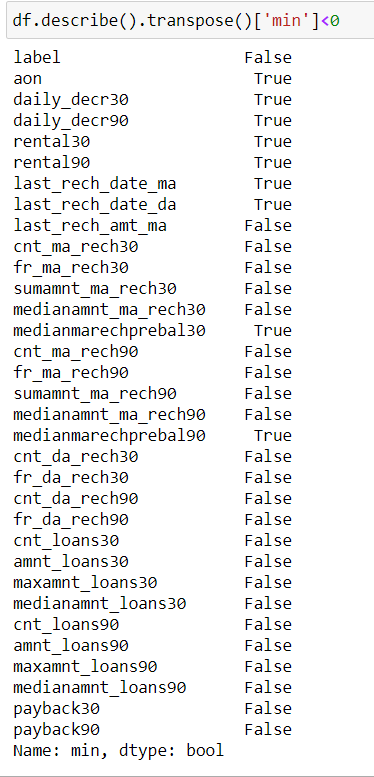
To get the overview of the data, looking at first 5 rows.





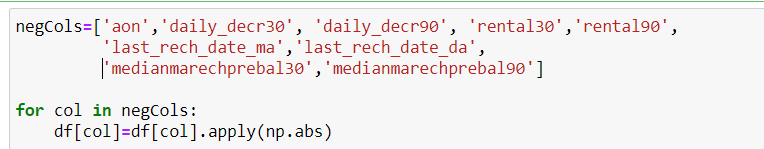
It can be see that all the data is not in the same range. So we have to scale the data.

It can also be seen that the min values are less than 0.

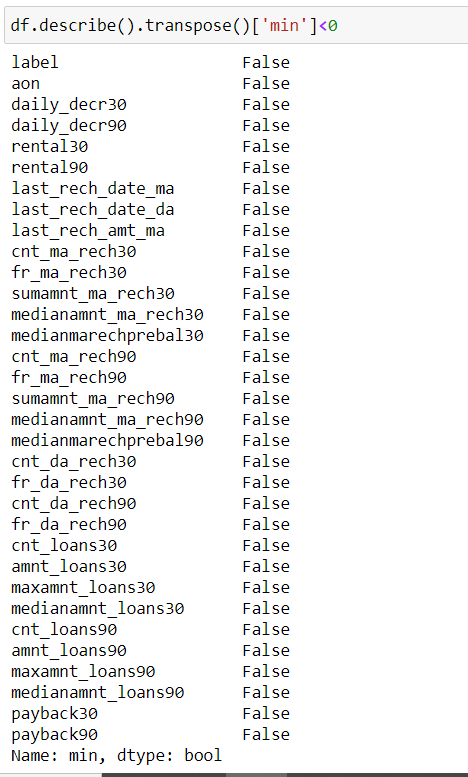


aon: age on cellular network cannot be a negative number, so assuming that the it is a data entry mistake and modifying it to positive value.

Similarly for 'daily\_decr30', 'daily\_decr90', 'rental30', 'rental90', 'last\_rech\_date\_ma', 'last\_rech\_date\_da', 'medianmarechprebal30', 'medianmarechprebal90'. So modifying all the negative values to positive values.

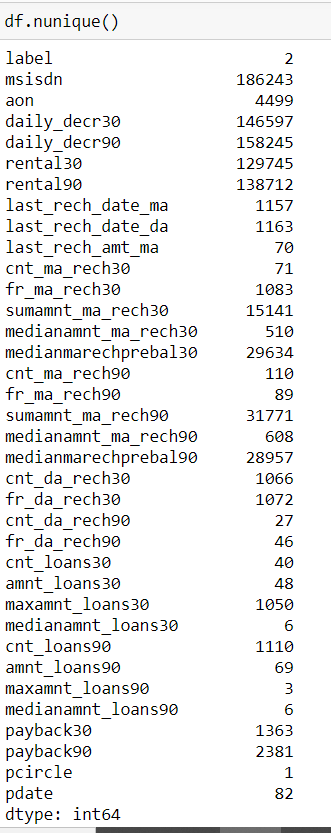


Modifying the values to positive.

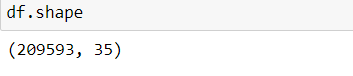


After modifying all the values to postives there are no more negative values.

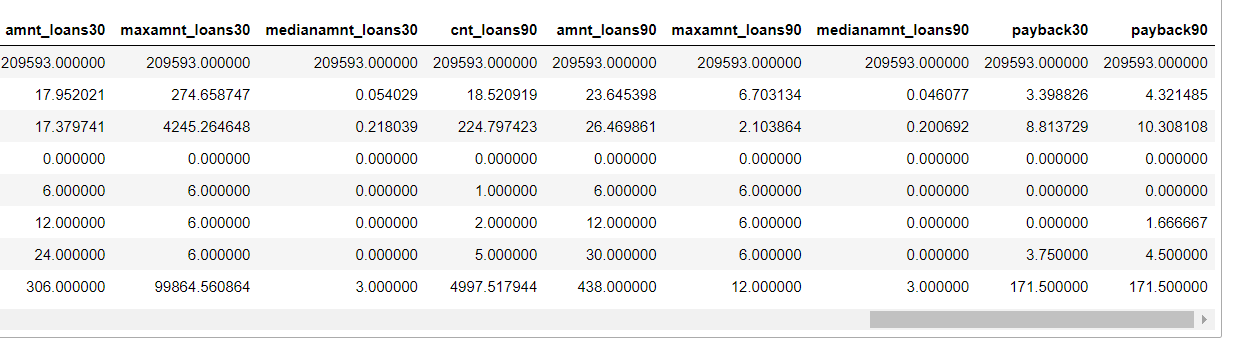
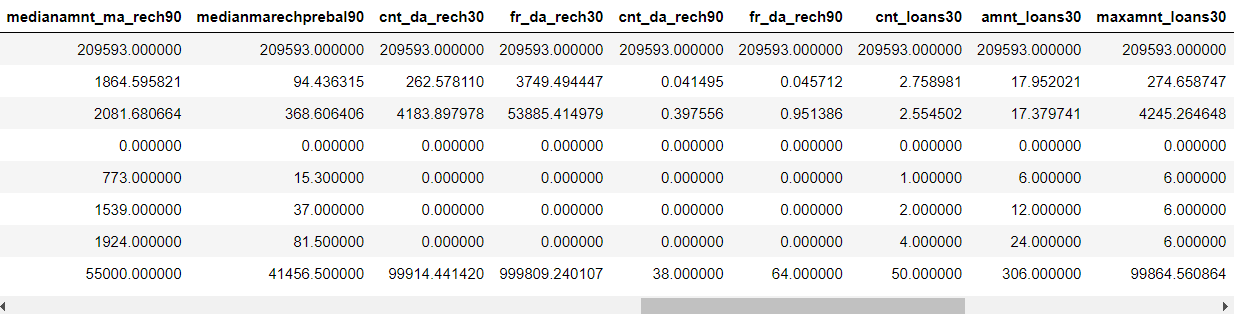
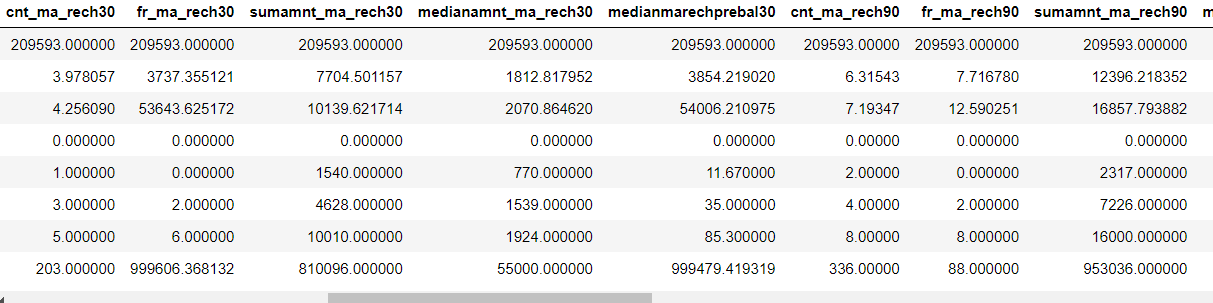
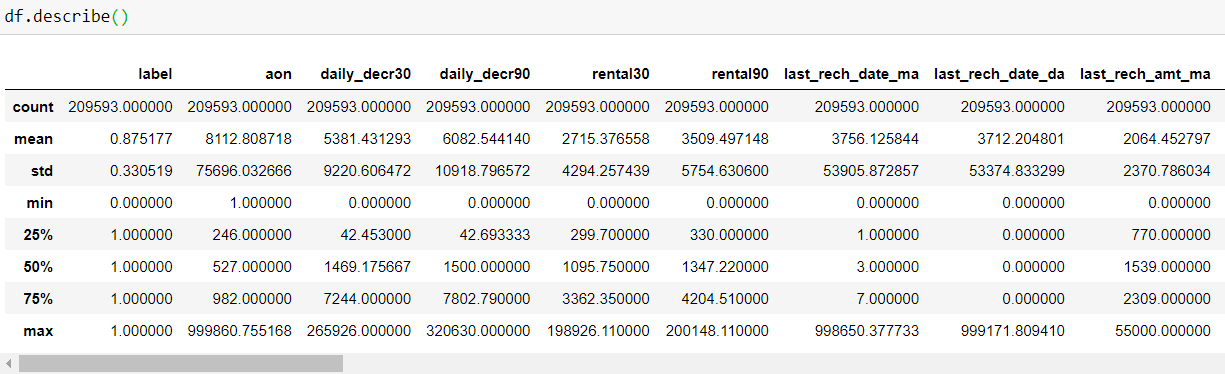
Checking the number of unique values in each feature.

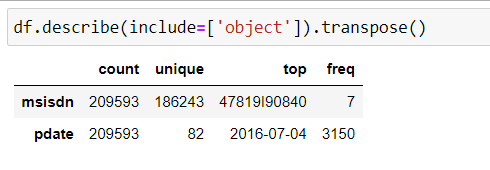


Pcircle has only one unique value, So dropping the feature.



After dropping pcirlce feature there are 35 features now.



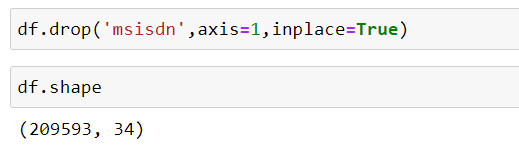


\* Summary statistics shows all the statistics of our dataset i.e. mean, median and other calculation.

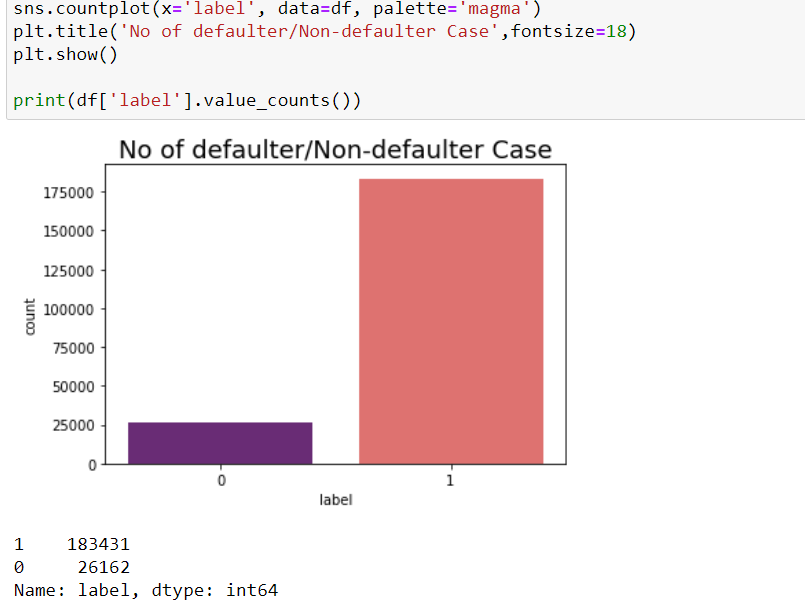
\* Mean is greater than median in all the columns so our data is right skewed.

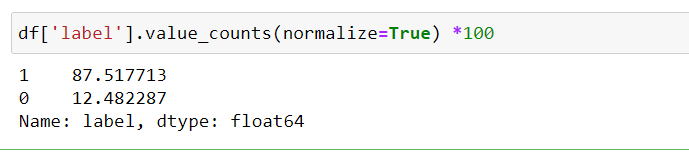
\* The difference between 75% and maximum is higher that's why outliers are there, which needs to be removed.

\* The pdate column tells the date when the data is collected. It contains only three months data.

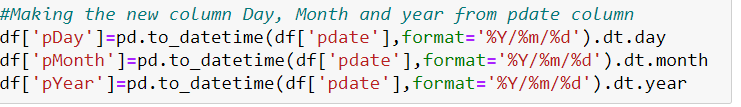


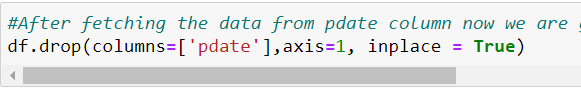
\* msidn is a mobile number of user and mobile number is unique for every customers. And it does not help in predicting whether a customer will pay back the loan so dropping it.





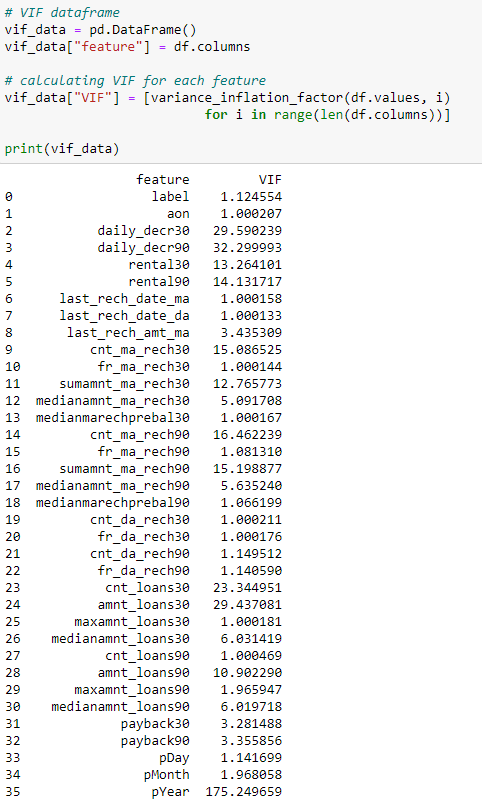
\* After seeing the target feature - label column for this dataset it is clearly shown that 86.11% of data is label 1 and only 13.8% of data is label 0 so our dataset is imbalanced.





Created pDay, pMonth, pYear features from pdate and dropped the pdate feature.

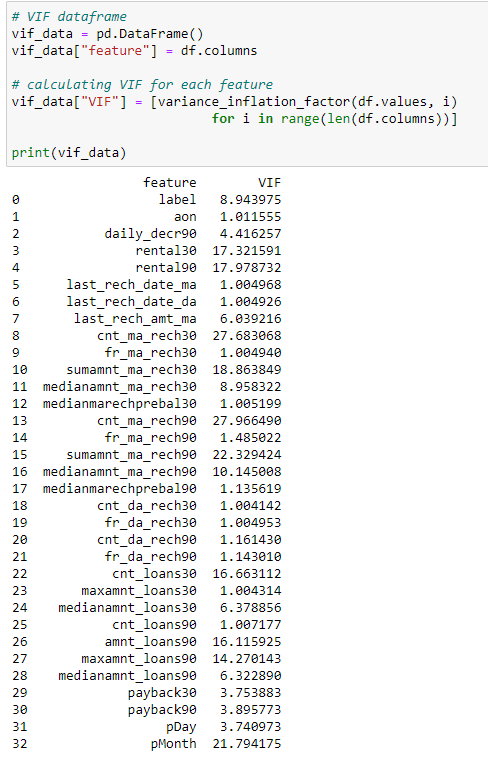
Checking variance inflation factor



VIF should be between 1 to 5 so looking at others and dropping one by one dropping pYear as it has only one uniqie value however

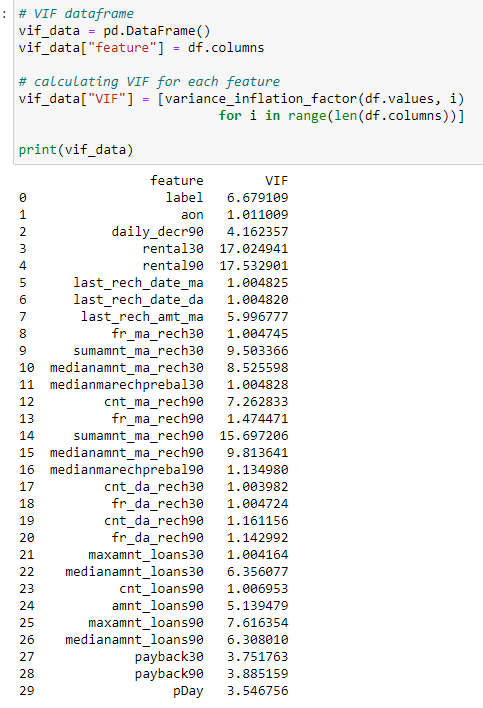
dropping daily\_decr30, amnt\_loans30, pYear.





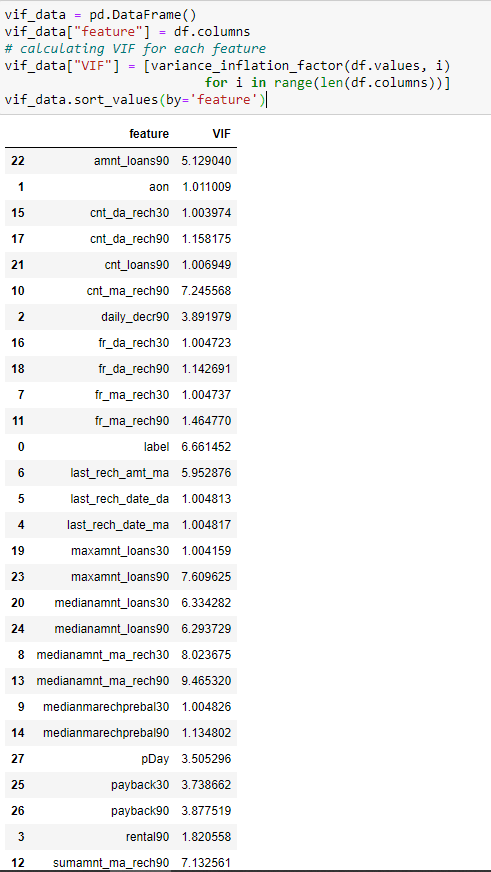
Checking the variance inflation factor again and dropping cnt\_ma\_rech30, pMonth, cnt\_loans30





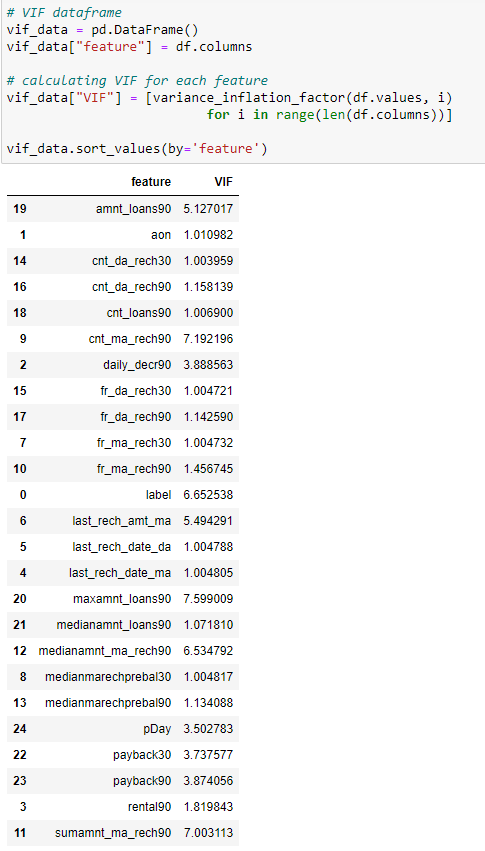
Checking the variance inflation factor again and dropping rental30, sumamnt\_ma\_rech30





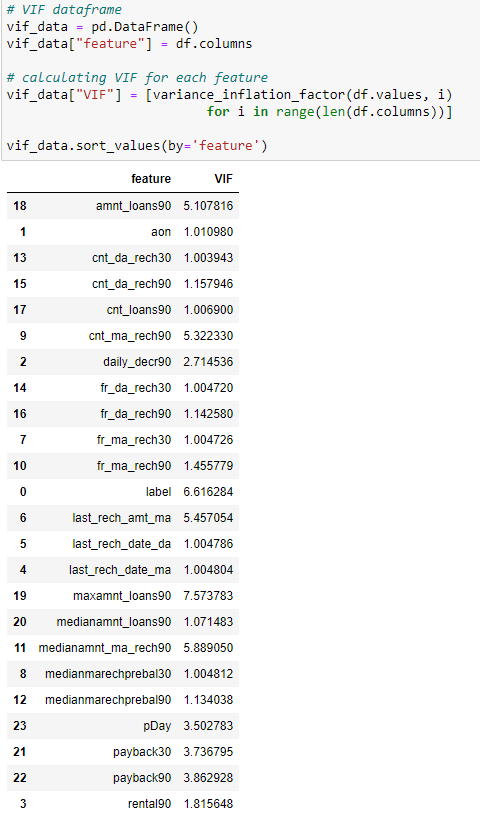
Checking the variance inflation factor again and dropping maxamnt\_loans30, medianamnt\_loans30, medianamnt\_ma\_rech30

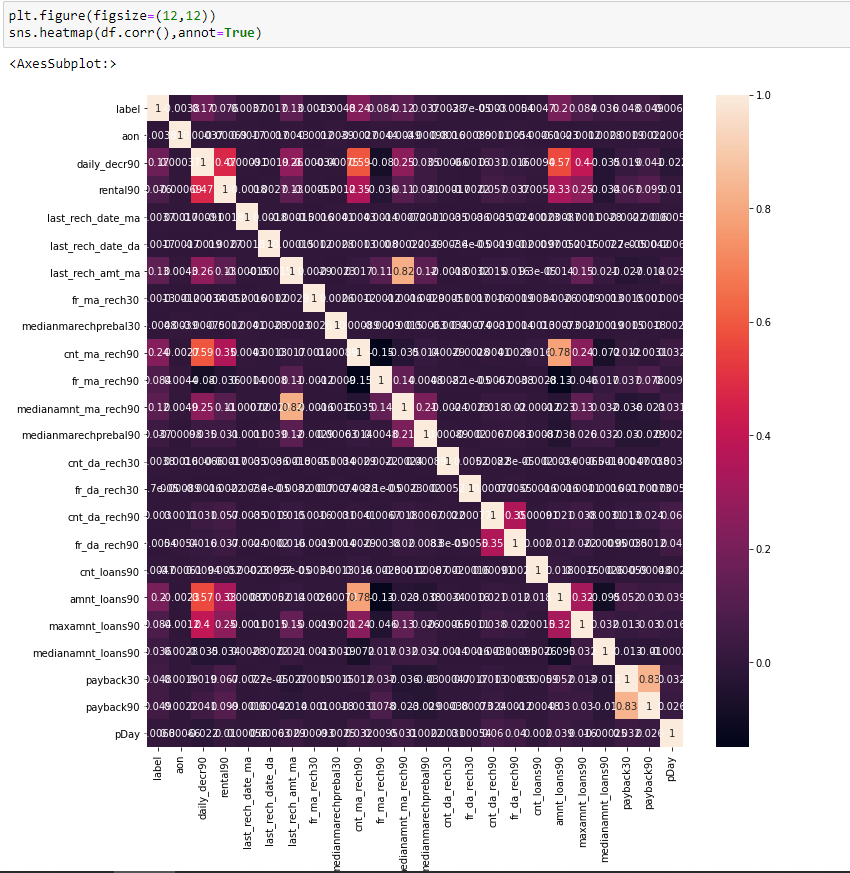


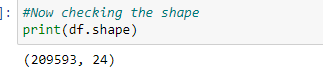


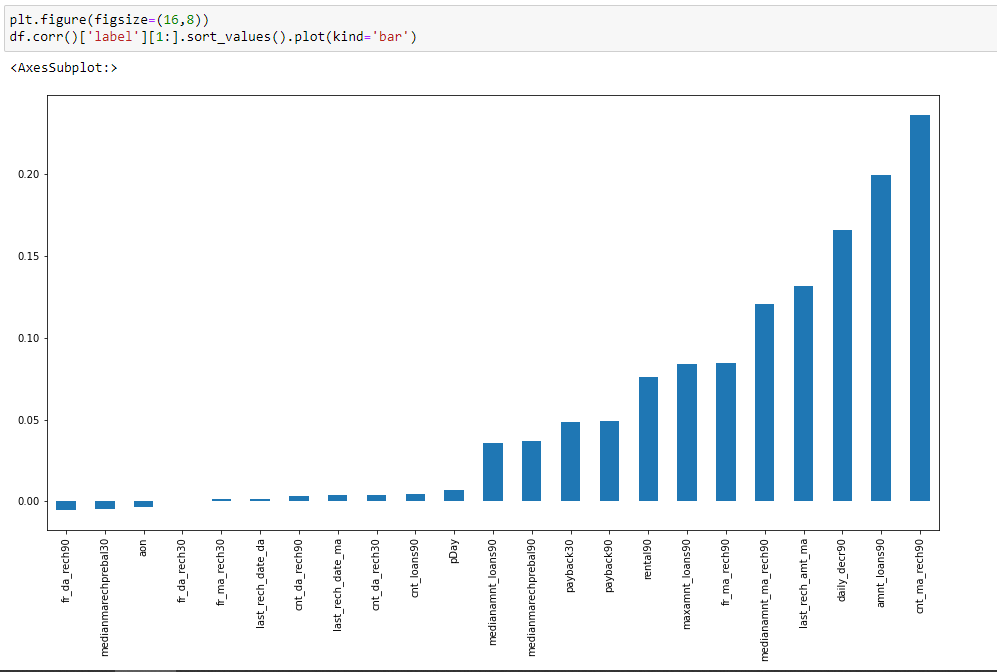
Checking the variance inflation factor again and dropping sumamnt\_ma\_rech30.



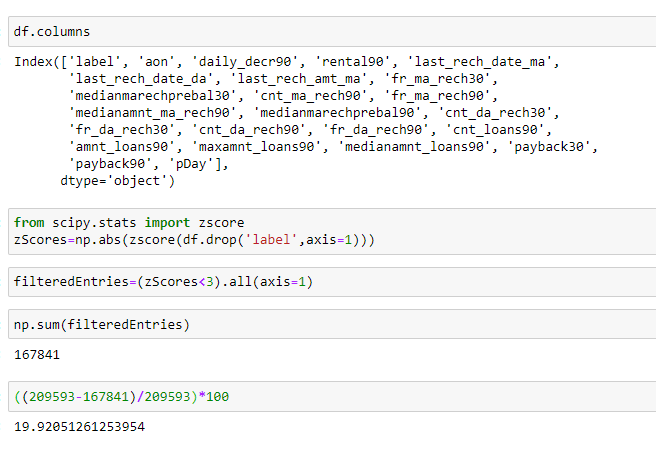






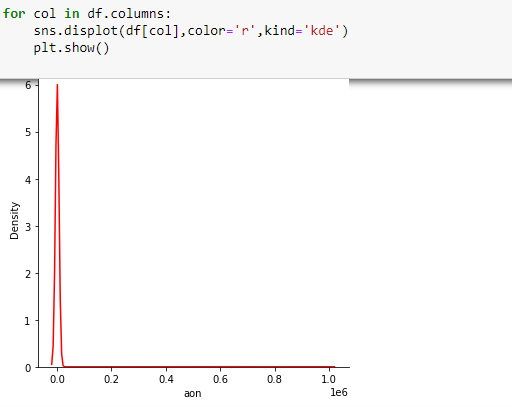


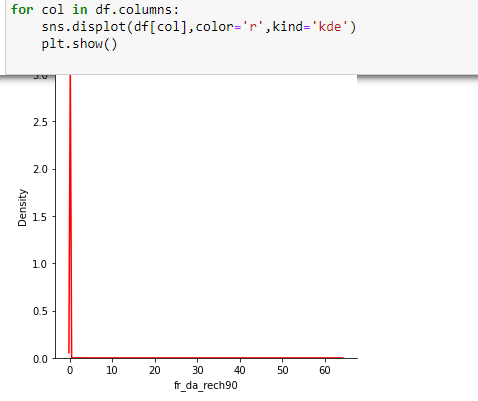
**Checking for Outliers:**

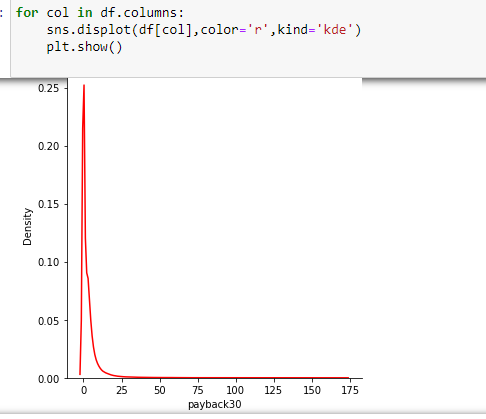


There are 20 percent of outliers, so not removing them as we loose a huge amount of data.

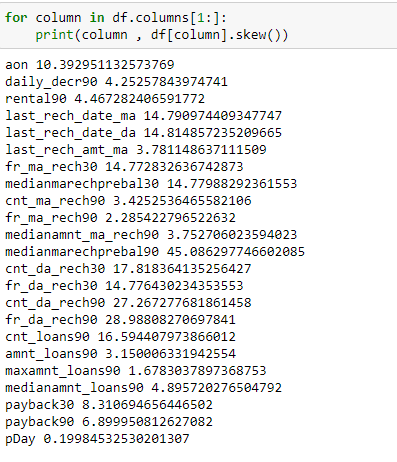
Checking skewness:



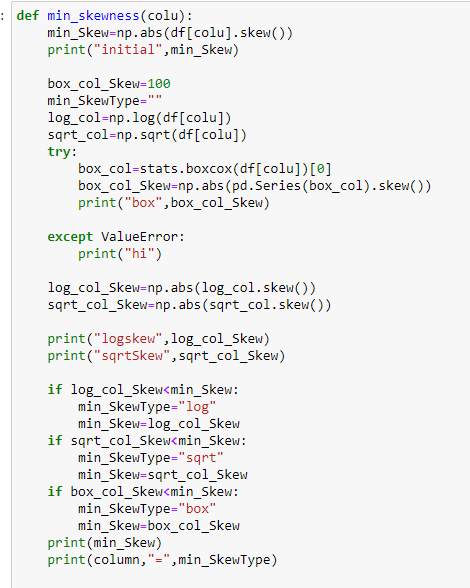




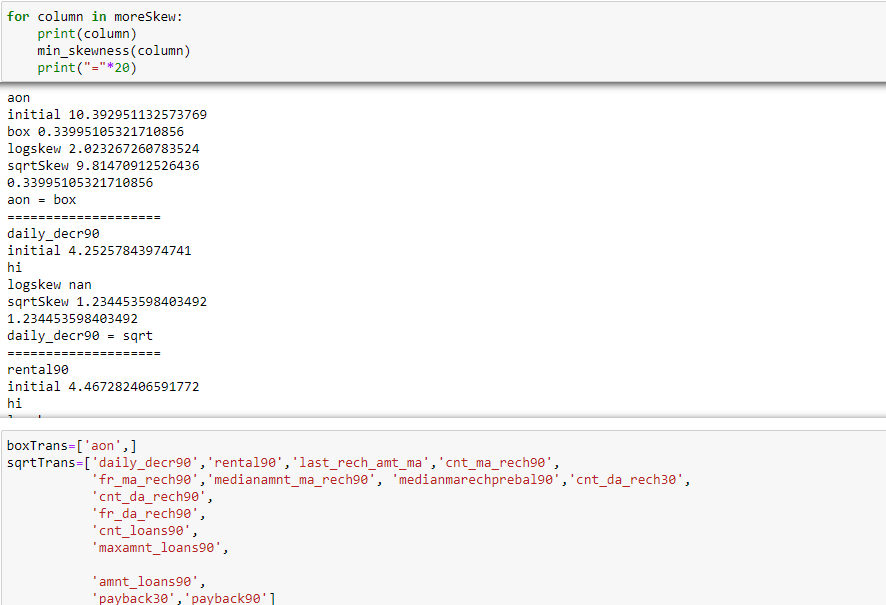
Almost all features are right skewed.



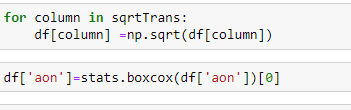
Almost all the features have a skewness more than 0.5.

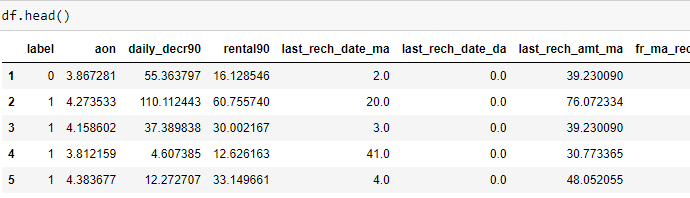


Created a function to know which transformation gives less skewness.

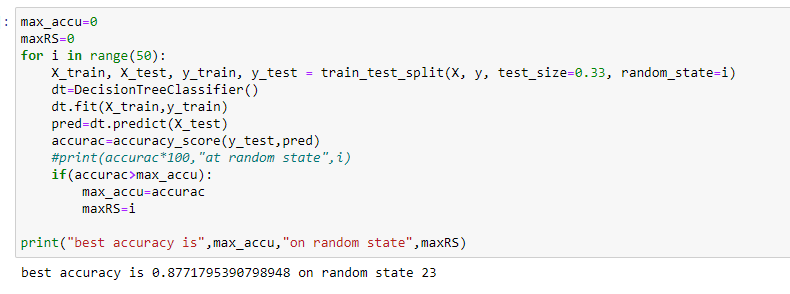


For ‘aon’ boxCox transformation gives less skewness and for all other features square root transformation gives less skewness.



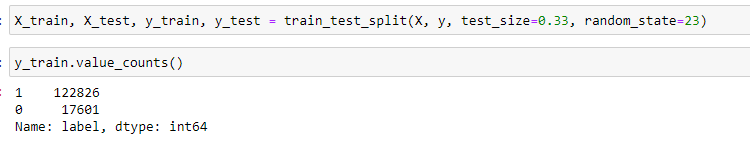


Checking out different random states to see which random state gives more accuracy.



Random state 23 gives more accuracy of 87%.

Splitting the data in to train and test set with random state od 23.



Y train value counts show that the target class is imbalanced with count of label 0 is17601 and count of label 1 is 122826.

To overcome the class imbalance issue there are two approaches – Upsampling and Downsampling.

**Upsampling** is a procedure where synthetically generated data points (corresponding to minority class) are injected into the dataset. After this process, the counts of both labels are almost the same. This equalization procedure prevents the model from inclining towards the majority class. Furthermore, the interaction (boundary line) between the target classes remains unaltered. And also, the Upsampling mechanism introduces bias into the system

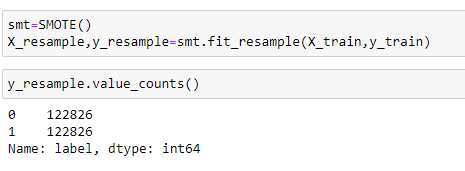
because of the additional information.

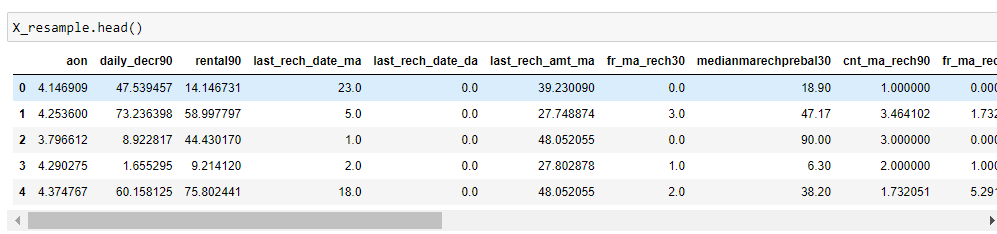
**Downsampling** is a mechanism that reduces the count of training samples falling under the majority class. As it helps to even up the counts of target categories. By removing the collected data, we tend to lose so much valuable information.

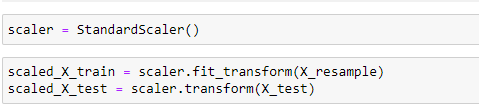
So it is better to do Upsampling and here we are applying SMOTE an implementation for Upsampling.

**SMOTE (Synthetic Minority Oversampling Technique) — Upsampling :-**

It works based on the K Nearest Neighbours algorithm, synthetically generating data points that fall in the proximity of the already existing outnumbered group. The input records should not contain any null values when applying this approach.



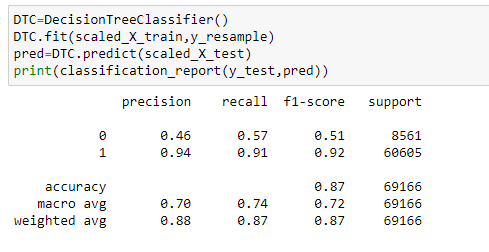




All the data in the training set is not in the same range. So applying Standard Scaler on the training data and transforming the training and testing data. Once the standard scaler is applied on the training and testing set all the values will be in the same range.

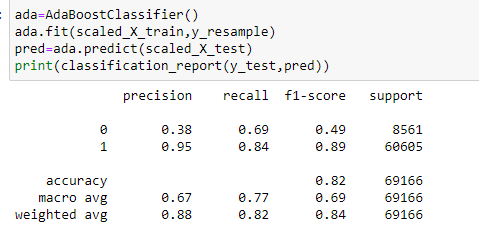
**Model Training**:

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.



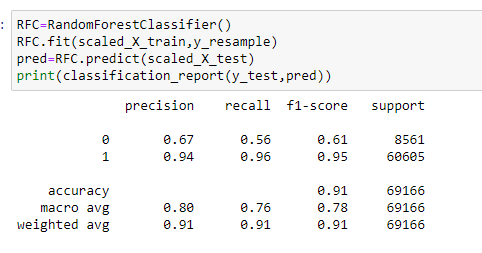
The accuracy on test set with Decision Tree classifier is 87% and F1 scores are 51 and 92 for label 0 and label 1 respectively.

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

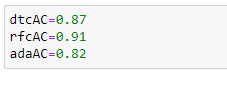


The accuracy achieved on test set with AdaBoost classifier is 82 percent and f1 scores are 49 and 89 on label 0 and 1 respectively.

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max\_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

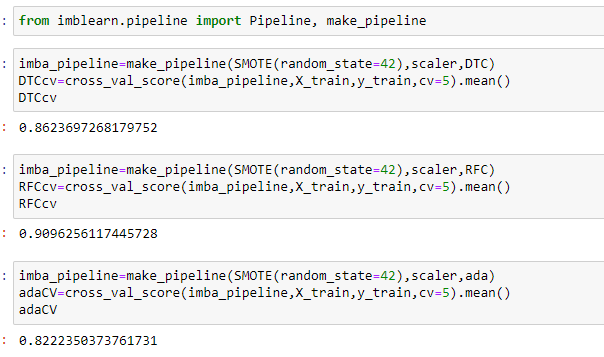


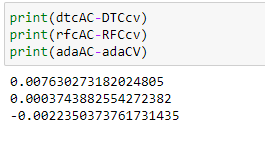
The accuracy achieved on test set with Random Forest classifier is 91 percent and f1 scores are 61 and 95 on label 0 and 1 respectively.



Checking the cross validation scores:

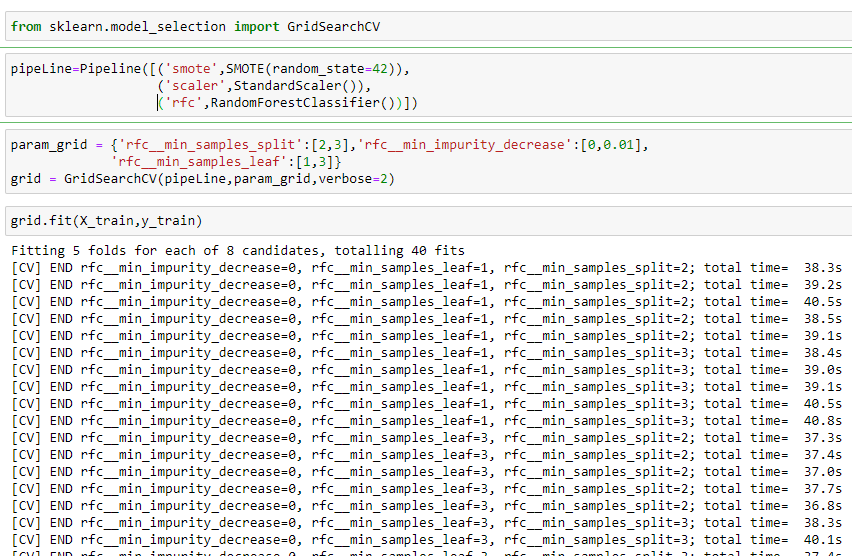
Sequentially apply a list of transforms and a final estimator. Intermediate steps of the pipeline must be ‘transforms’, that is, they must implement fit and transform methods. The final estimator only needs to implement fit. The transformers in the pipeline can be cached using memory argument.





Random Forest Classifier is the best model as it has the highest precision and recall and accuracy as well. Random Forest classifier has the less difference between the accuracy scores(on test set) and Cross Val score(on validation), so it is the best model.

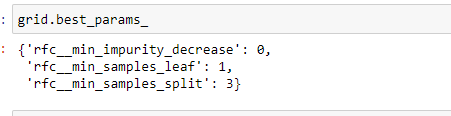
**Hyper Parameter Tuning – Random Forest Classifier:**

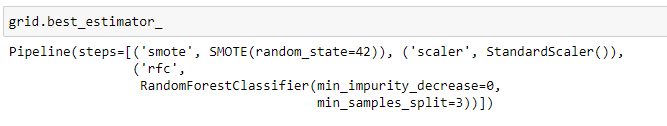
****

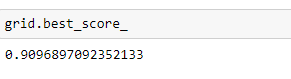
Hyperparameters are the variables that the user specify usually while building the Machine Learning model. Thus, hyperparameters are specified before specifying the parameters or we can say that hyperparameters are used to evaluate optimal parameters of the model. the best part about hyperparameters is that their values are decided by the user who is building the model. For example, max\_depth in Random Forest Algorithms, k in KNN Classifier.

Grid Search uses a different combination of all the specified hyperparameters and their values and calculates the performance for each combination and selects the best value for the hyperparameters. This makes the processing time-consuming and expensive based on the number of hyperparameters involved.

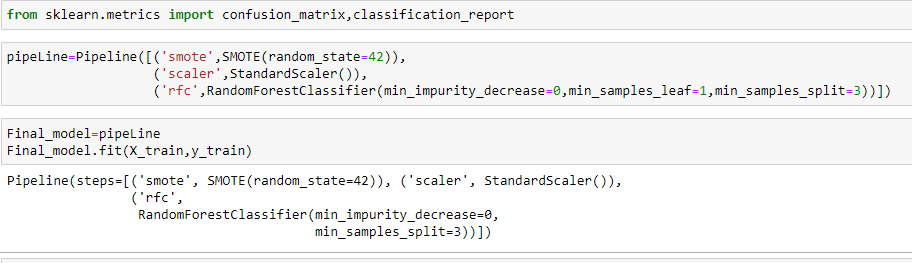
In GridSearchCV, along with Grid Search, cross-validation is also performed. Cross-Validation is used while training the model. As we know that before training the model with data, we divide the data into two parts – train data and test data. In cross-validation, the process divides the train data further into two parts – the train data and the validation data.

****

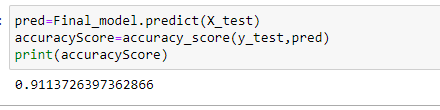


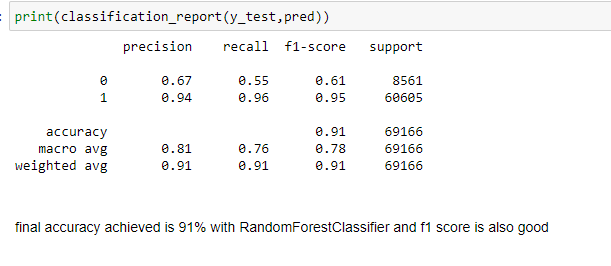


**Training the final model with the best parameters based on GridSearchCV:**

Creating a pipeline with Smote for Upsampling the training data, scaler – fitting the scaler on training data, transforming the training and testing data, Random Forest Classifier  with min\_samples\_leaf = 1, min\_samples\_split = 3, min\_impurity\_decrease = 0.

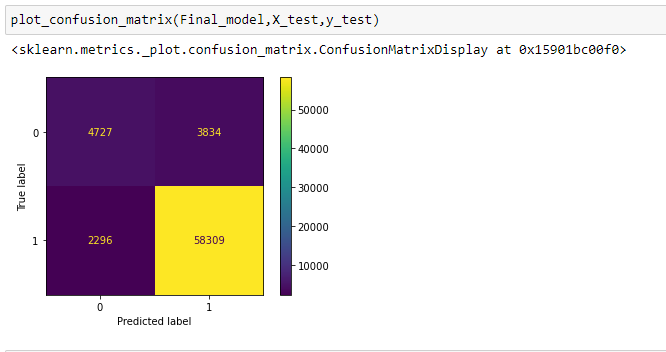
After training the model on the training data with the best parameters , accuracy of 85% is achieved on the testing data.





Classification report shows the f1 scores 0.61 and 0.95 on label 0 and label 1 respectively.

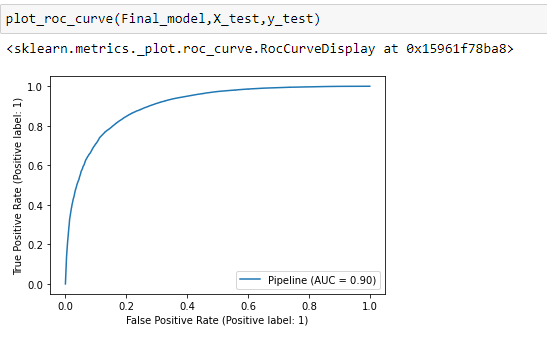
Final Accuracy is 91%.



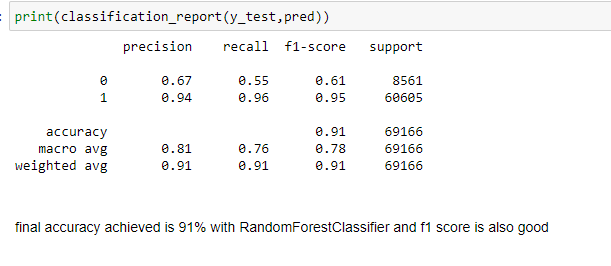
Confusion matrix reveals that out of 60605 records from class1, 58309 are correctly predicted as class 1 and 2296 records is falsely predicted as class 0. Out of 8561 records in class 0, 4727 are correctly predicted as class 0 and 3834 records are incorrectly classified as class 1.

A receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting the fraction of true positives out of the positives (TPR = true positive rate) vs. the fraction of false positives out of the negatives (FPR = false positive rate), at various threshold settings. TPR is also known as sensitivity, and FPR is one minus the specificity or true negative rate.

The roc\_auc\_score function computes the area under the receiver operating characteristic (ROC) curve, which is also denoted by AUC or AUROC. By computing the area under the roc curve, the curve information is summarized in one number.



The area under the curve is 0.90 which is very good.



In Conclusion, the Random Forest classifier is chosen as the best model and the accuracy is 91%.