

**Micro-Credit Defaulter Model**

Submitted by:

Khushboo Khatri

Acknowledgement

I would like to appreciate FlipRobo for giving us the opportunities, to work on this project. It really helps me to widen my knowledge about different subjects matters.

I really would like to appreciate Mr. Shwetank Mishra to clear my doubts, while building the ML model. He helps me to understand the subject in a better way.

None the least I would like to appreciate Datatrained academy and their mentors to help us teach the basics required to code the entire machine learning algorithm

Introduction

**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).

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.

**Problem Framing:**

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 paid i.e. Non- defaulter, while, Label ‘0’ indicates that the loan has not been paid i.e. defaulter.

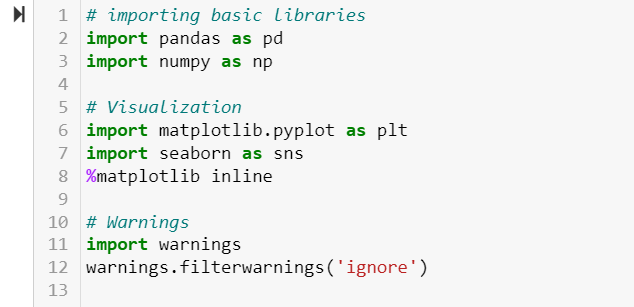
# Analytical Problem Framing

## Mathematical/ Analytical Modeling of the Problem:

Micro credit Loan Defaulter is a dataset consist of 37 columns and 209593 rows, here the main objective is to find out if the loan will be paid on time (non defaulter, 1), or will not be able to pay the loan on time (Defaulter, 0). We can clearly observe that in the label the number of 0’s are way less than number of 1’s in the dataframe. Which implies that our label is imbalance. One of the two things can be done here in this case, we can leave it as it is , as it denotes 0 or We can apply sampling (over/under) in order to manage this imbalance.

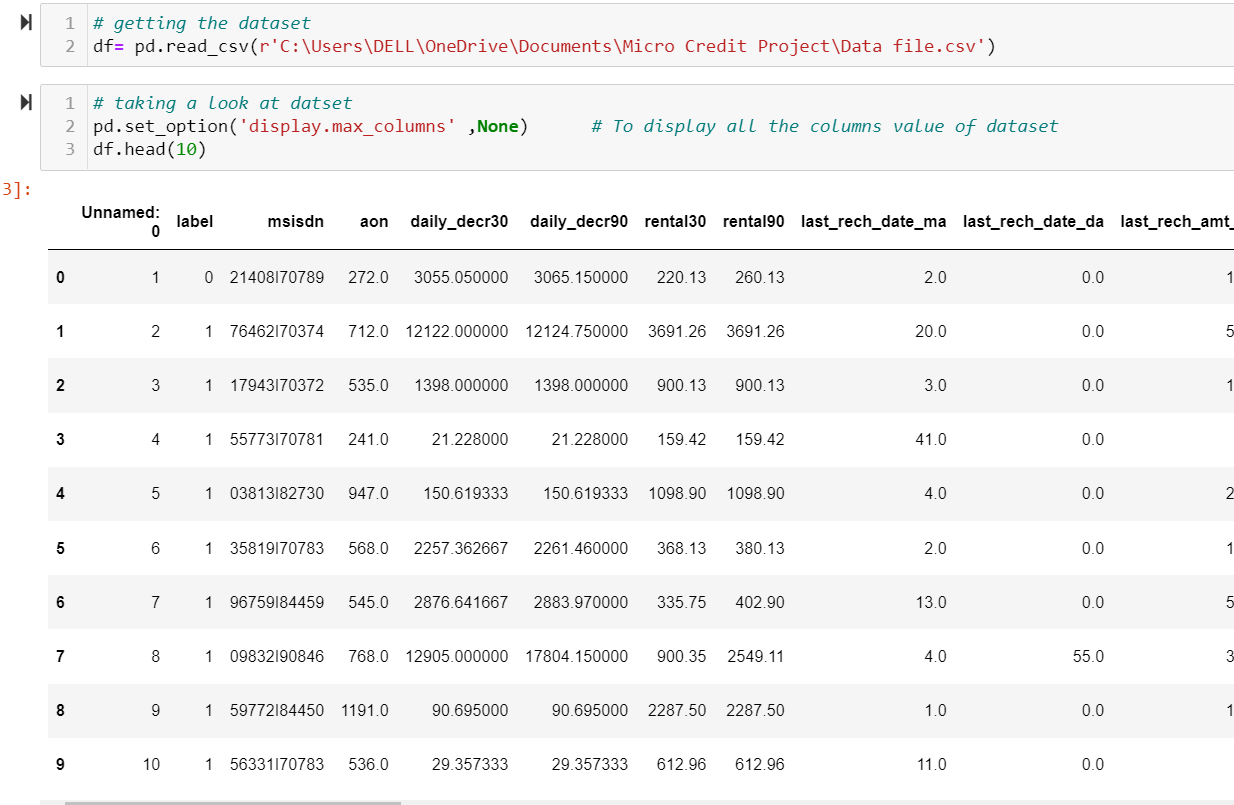
As this data is directly provided by the authority, so its not a good idea to disturb the dataset. Instead, we can look for precision score or recall just to check how our model is working.

Let’s start with importing the basic libraries, so that we can start working on our dataset. Starts with loading.



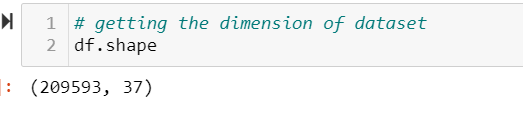
As our libraries has been imported, now we can load the dataset, on which we need to build the model.

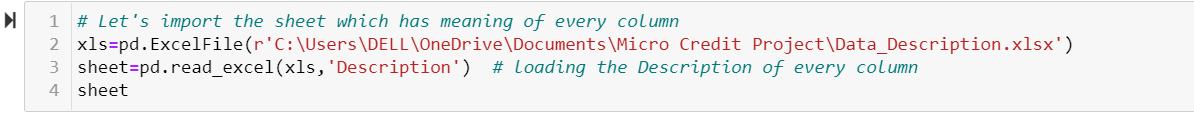
## Data Source and Their Format:



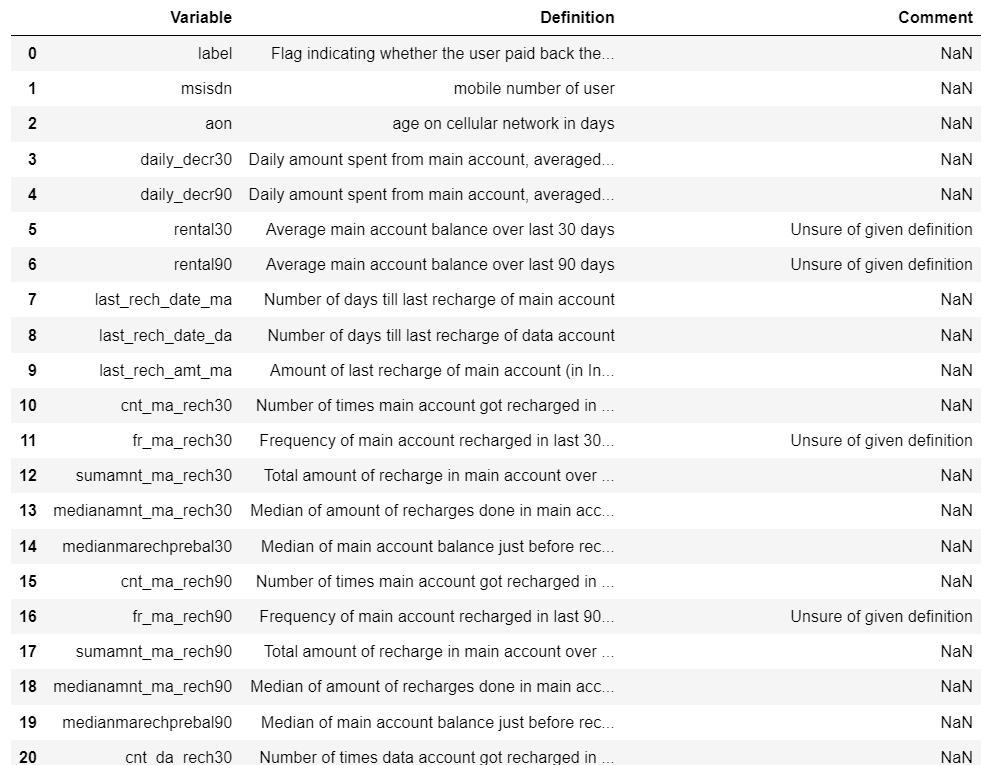
We use pd.set\_option () in order to display all the columns of our dataset.

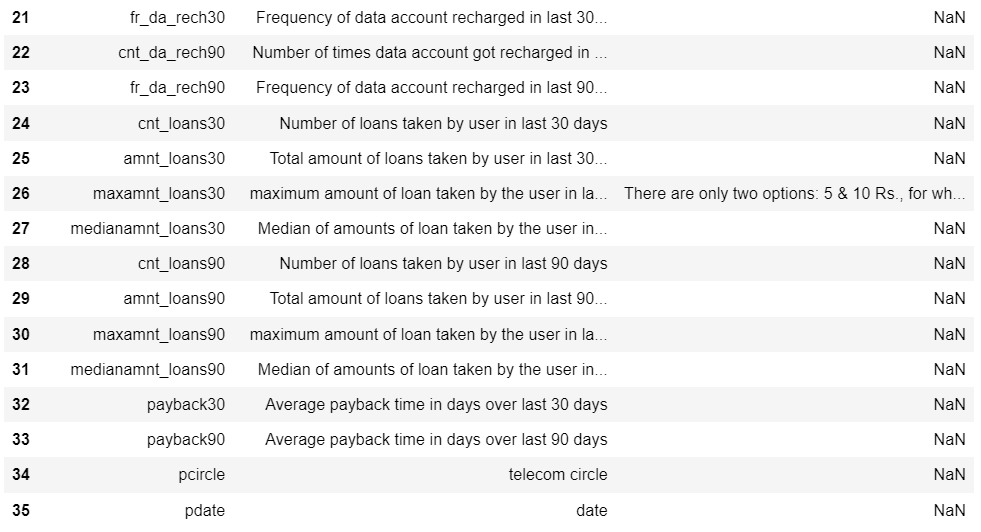
Let’s see the dimension of the dataset, and also let’s figure out the meaning of each columns of the dataset.



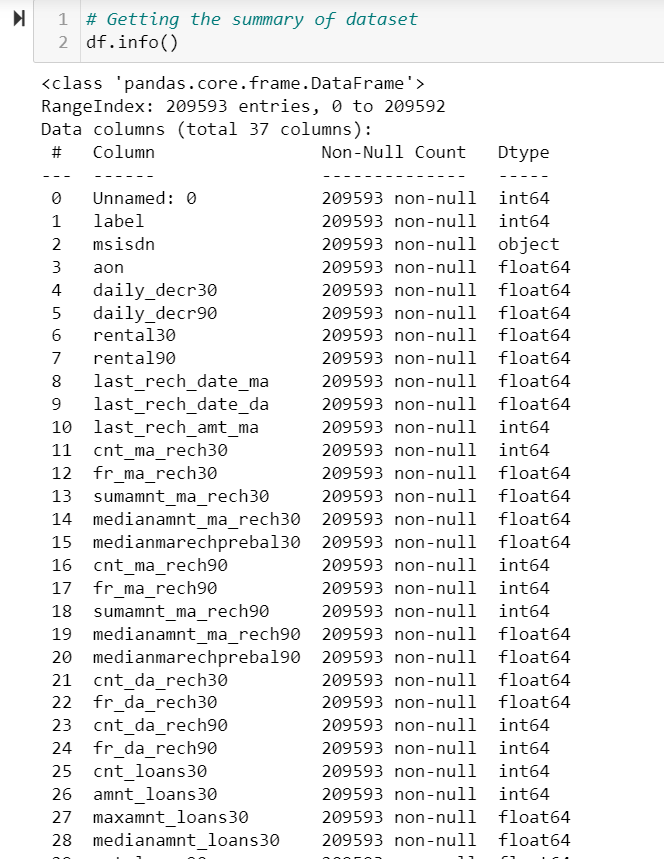


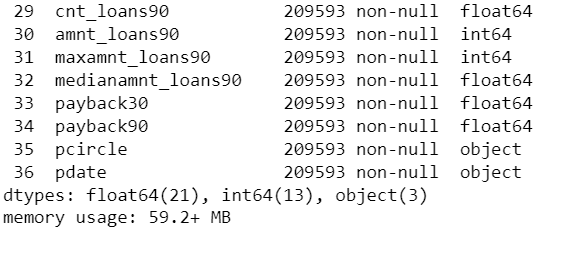
We are provided with another excel sheet which consist of the meaning of every column in the dataframe. To load that in our notebook we use above codes.





Now, let’s go ahead and get precise summary of the dataset. For that we will use .info()

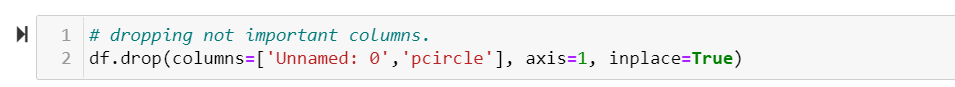




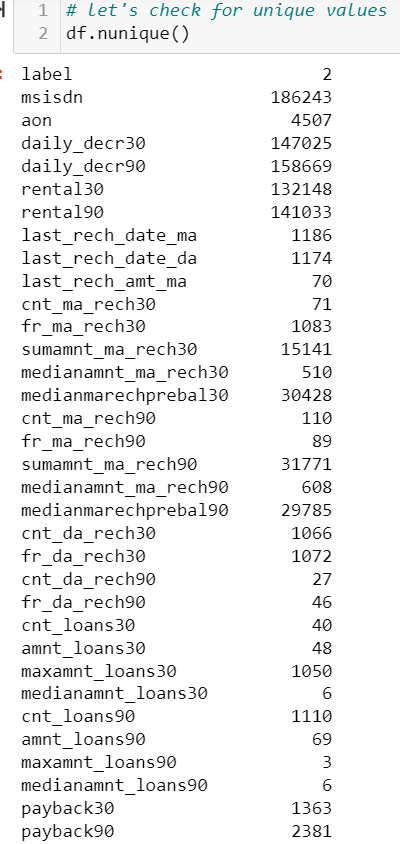
We can observe that except msisdn, pcircle, pdate all others are numerical columns. Also there are no null values in the dataset, which is quite a relief , we can directly move ahead with EDA, feature Engineering and Data Cleansing part.

## Data Preprocessing:

From info() we observe that Unnamed: 0 column is there which also don’t add much value in ML model making, as it is indexing column only so we can drop it. Also pcircle has only one value, as the data is collected from one operator only. For that we will use drop().

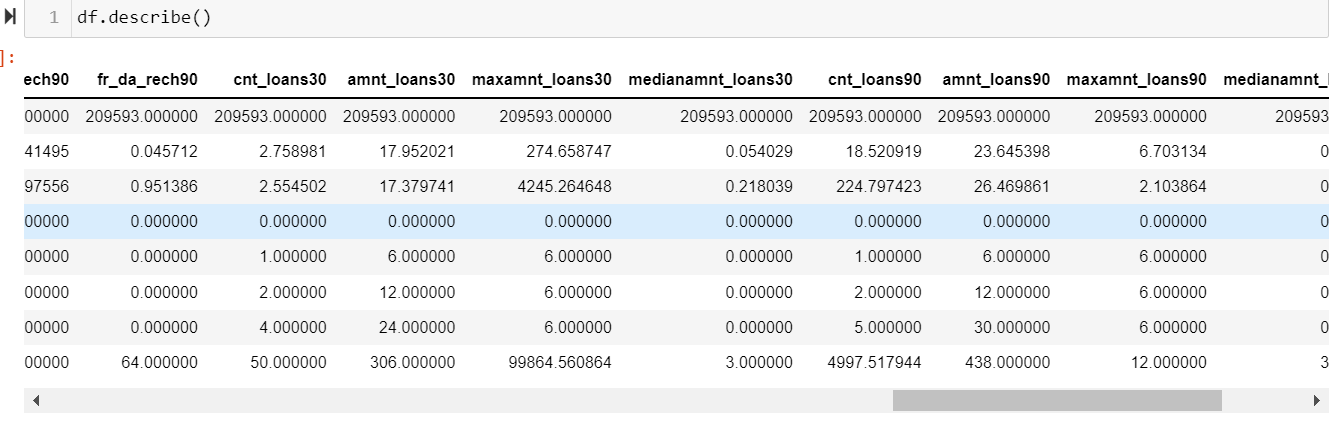


Let’s check for unique value in each column.





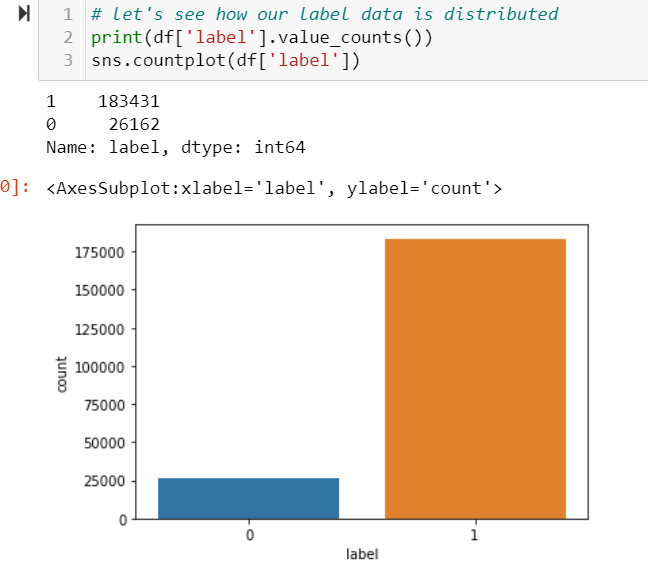
We can observe that the label has only two unique classes, indication defaulter/not defaulter. It is a classification problem with target variable as label. In mobile number of user we can see that there are some mobile number which are repeated in the columns.



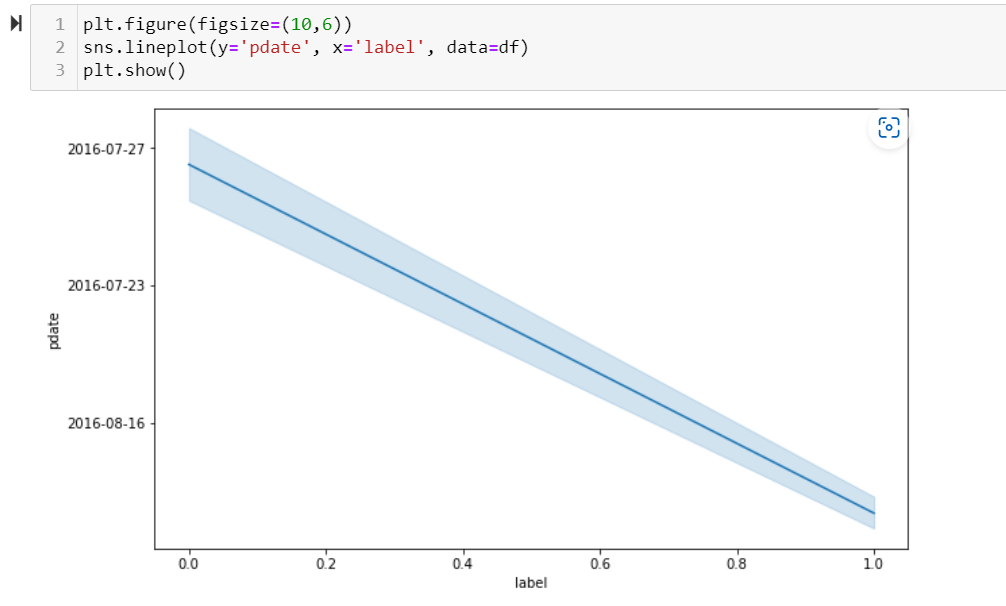
From above we did statistical analysis of the dataset, and we observe that the minimum zero value denotes that , some customers have no loan history.

## EDA

Now, Lets start with distribution of our label column

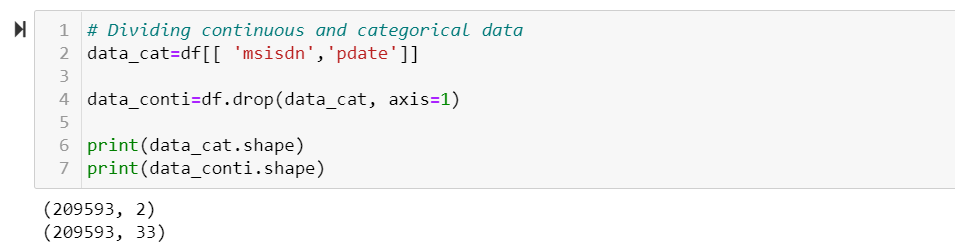


We can observe that the value for '1' and '0' is highly imbalanced, the number of 0 is way less than number of 1. But as this data is provided by the telecom company itself, we will not use any over sampling technique to handle this imbalance and undersampling can lead to information loss. '1' indicates non defaulter, '0' indicates defaulter.



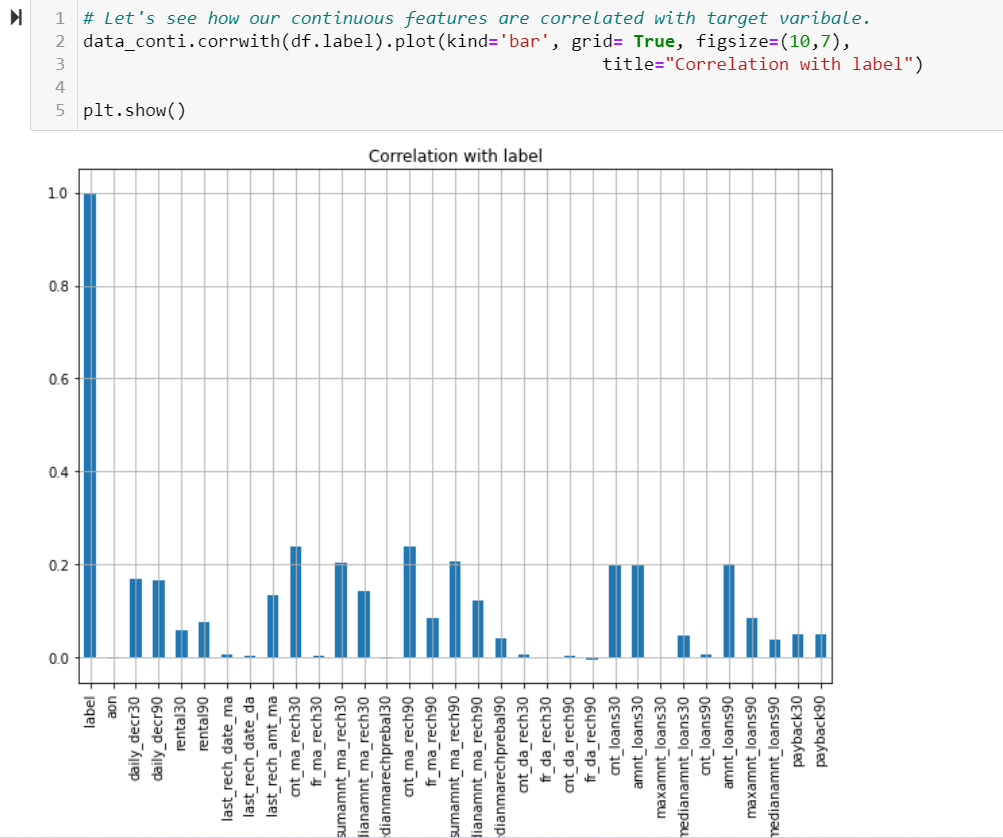
We can observe that on certain dates we got more defaulters , there might be some connection of date with label, so its better to feature engineer this columns, in order to get meaningful information from here.

Before moving ahead let’s separate categorical and continuous columns.



We can observe that we have only two categorical column and 33 continuous columns which includes our target variable i.e. our dataset so far has 32 continuous features , 2 categorical features and one target column.

Let us find out how our continuous features are correlated with our target variable, for this we can use .corrwith() function.

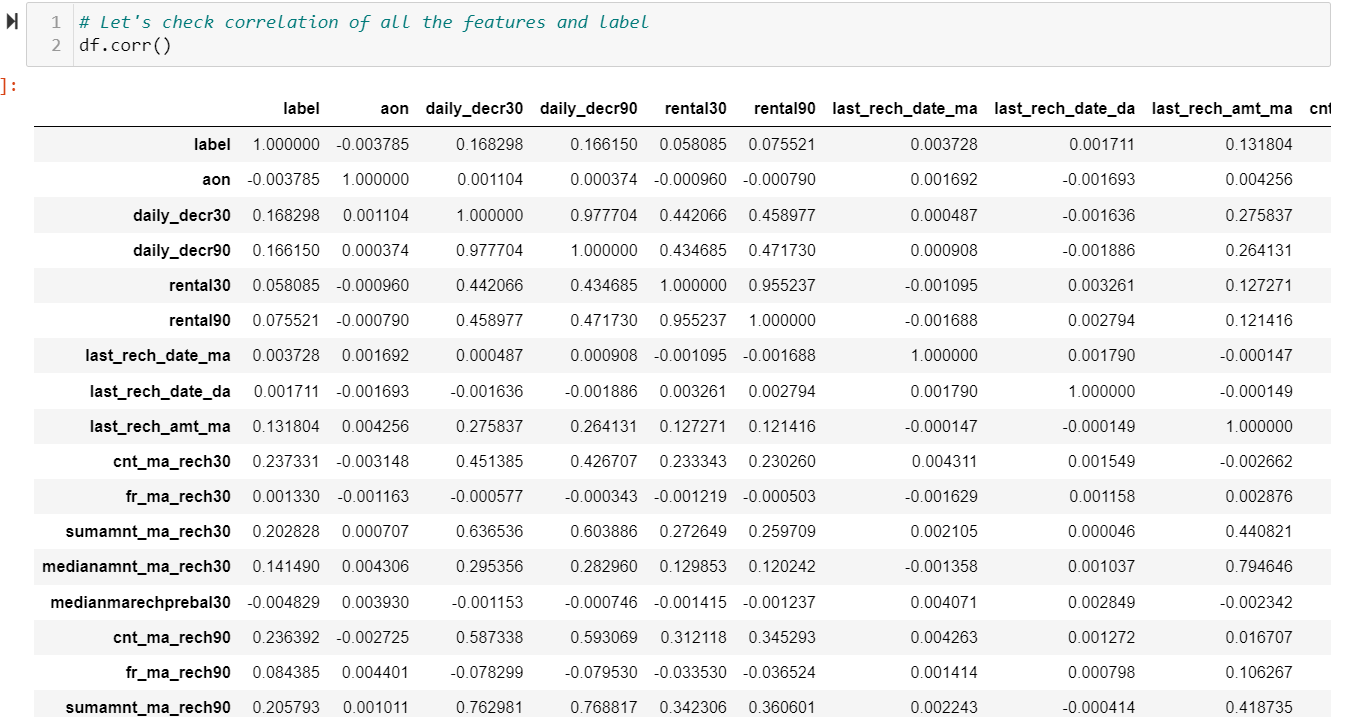


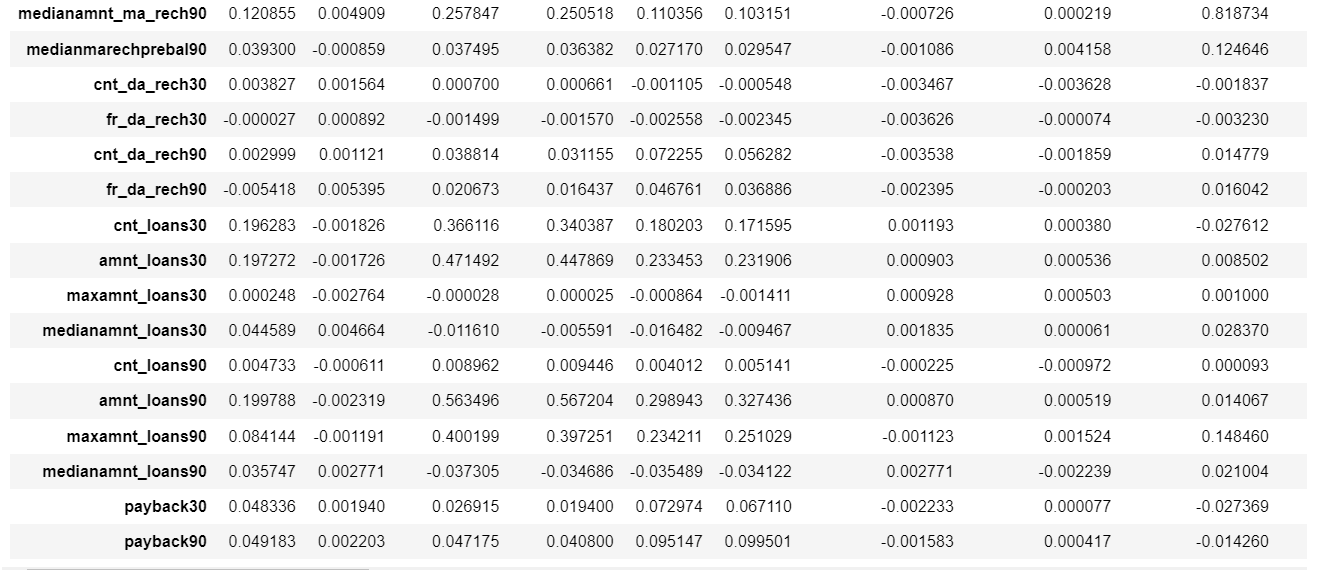
From above We can observe that cnt\_ma\_rech30 and cn\_ma\_rech90 seems to be most correlated features with or target variable label.

While aon, fr\_da\_rech30, maxamnt\_loans30, medianmarechprebal30 are not atall correlated with label. Other features like last\_rech\_date\_ma, last\_rech\_date\_da, fr\_ma\_rech30, cnt\_da\_rech30, cnt\_da\_rech90, fr\_da\_rech90 and cnt\_loans90 also shows nearly zero correlation with label.

Let's confirm it using correlation matrix, then we can drop these features columns with don't show any or nearly zero correlation with label.

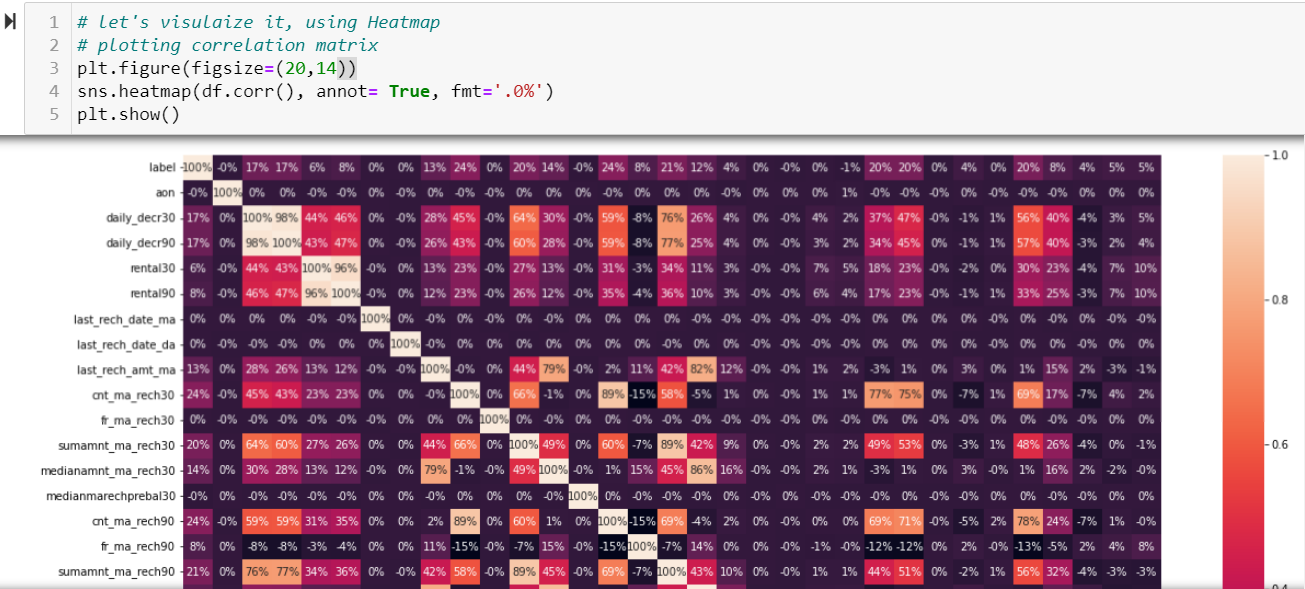
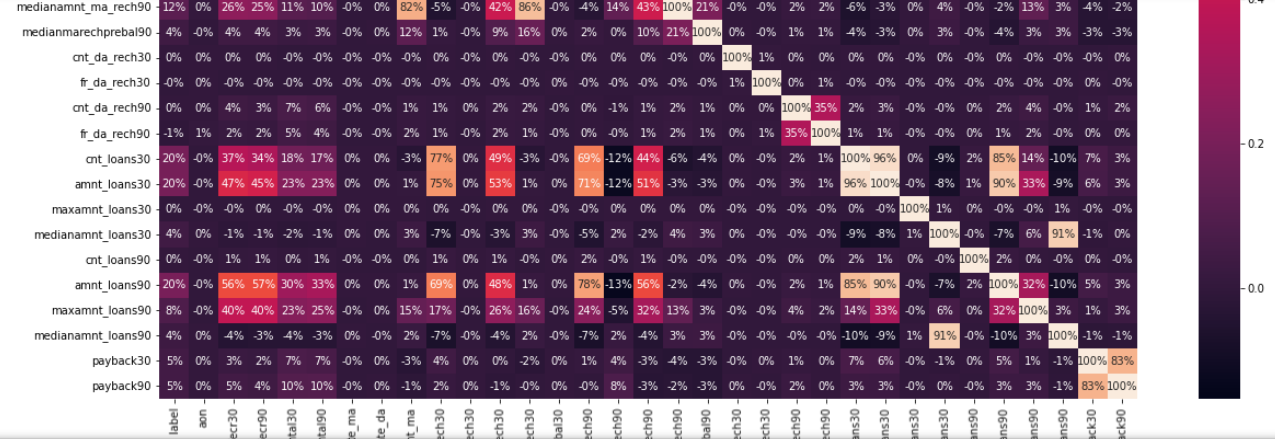
## Correlation Matrix





Above is the mathematical representation of correlation matrix, let’s visualize it. Visualization gave us clear picture of how our columns are correlated with each other and also with the target variable.

Correlation matrix also helps us to identify the multicollinearity present in the dataset. We can confirm multicollinearity using scatterplot.

 Correlation is **a statistical measure that expresses the extent to which two variables are linearly related** 

Observations: From above observation it is clear that, there are few columns which are 0 or almost 0 % correlated tp label, so we will drop them for our ML model building.

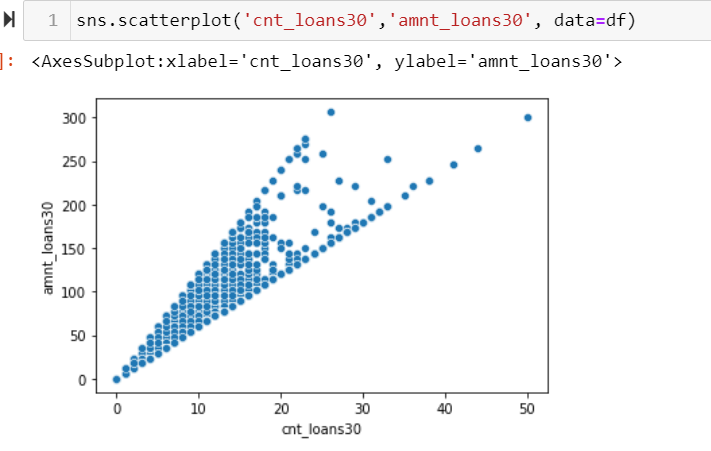
These includes : 'cnt\_loans90', 'maxamnt\_loans30', 'cnt\_da\_rech90', 'fr\_da\_rech30', 'cnt\_da\_rech30', 'medianmarechprebal30', fr\_ma\_rech30', last\_rech\_date\_ma', last\_rech\_date\_da', 'aon'. we can drop these, as they are zero or almost 0% correlated with label.

We can observe that some features shows strong correlation with each cnt\_loans30 - amnt\_loans30, daily\_decr30 - daily\_decr90, rental30- rental90, medianamnt\_loans90- medianamt\_loans30, sumamnt\_ma\_rech30 -sumamnt\_ma\_rech90.

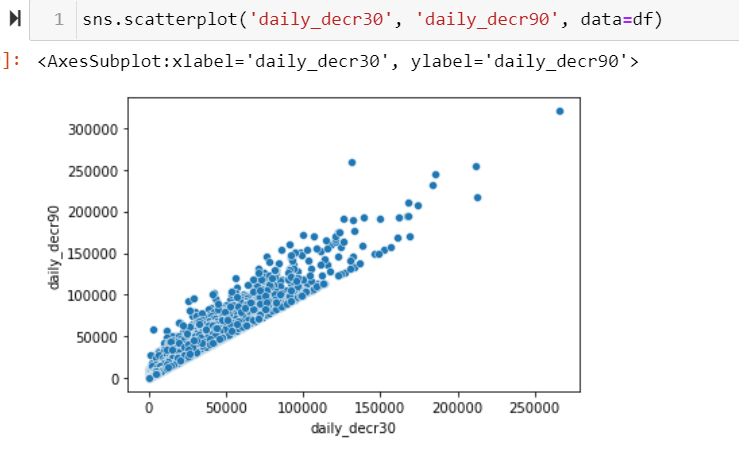
Let’s start with dropping those columns which are not correlated with labels.



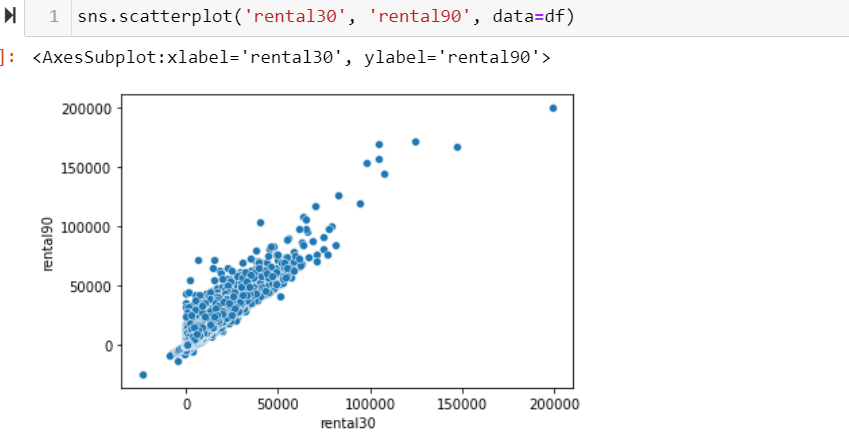
Now, let’s check for columns which shows strong correlation with each other, will use scatter plot to see if they are correlated.



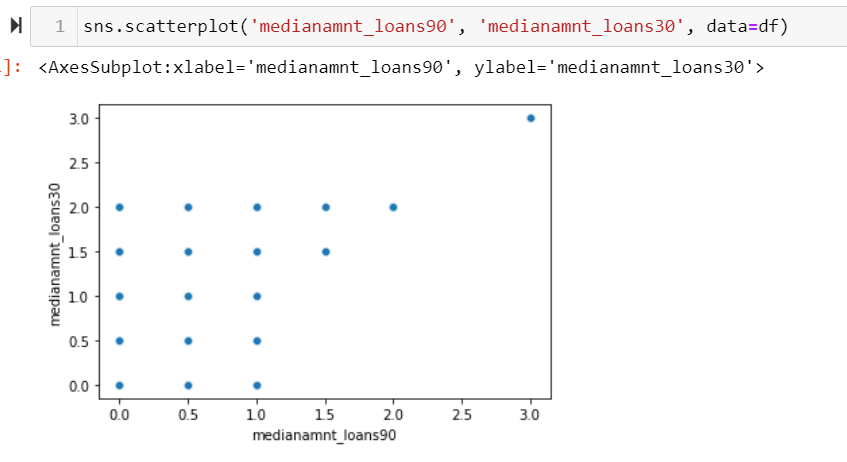
We can observe some positive correlation , as value of one increase the value of other also increases.



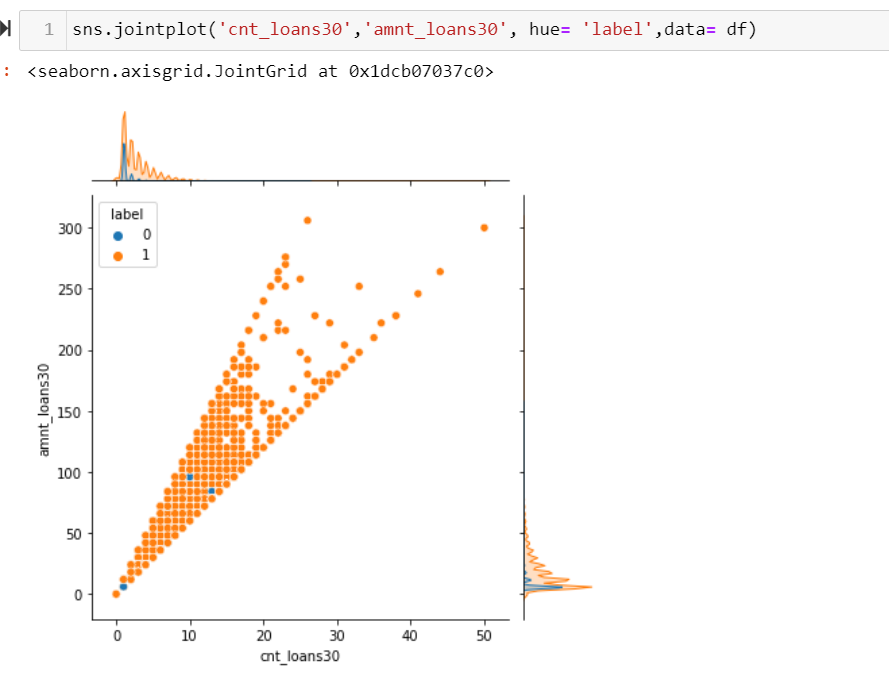
We can observe highly positive correlation between daily\_decr90 and daily\_decr30, we can drop one of them to avoid multicollinearity in the ML Model building.



We can observe strong highly positive correlation between the two, we can drop one of them in order to reduce multicollinearity in our ML model building.



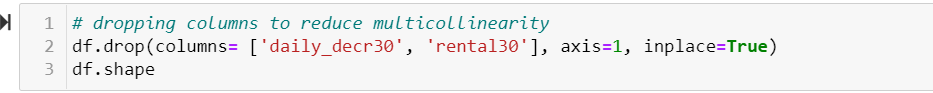
The correlation between above two features are not so clear using scatter plot, there is no definite relation exist between the two.

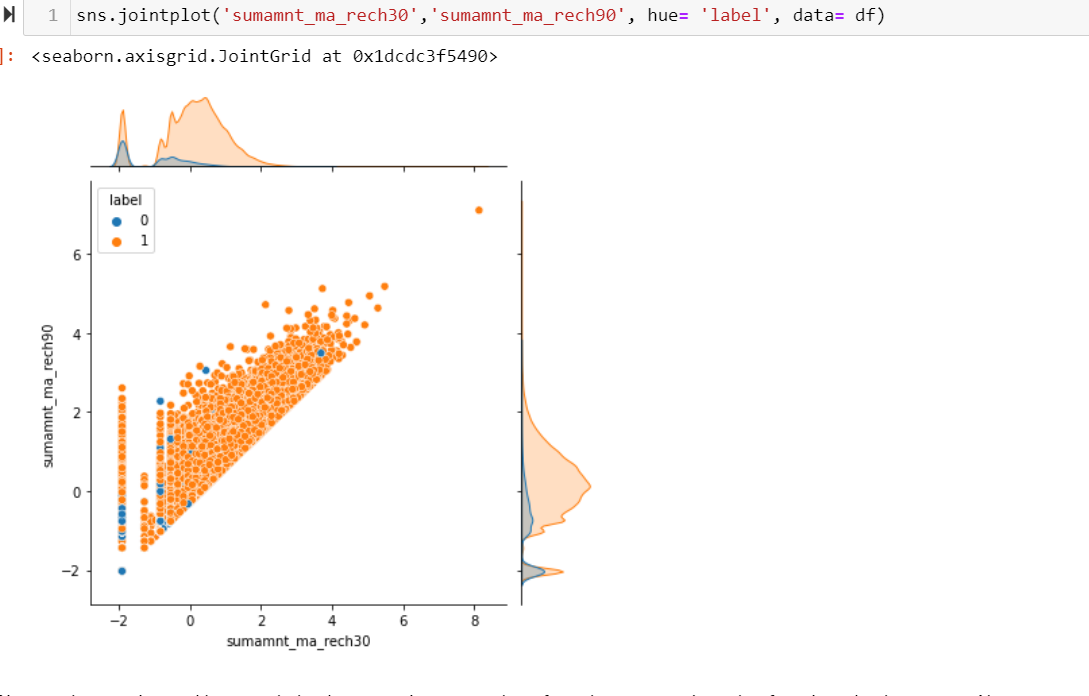


We can Observe that both are correlated , and there correlation with label seems to be the same. We can drop one of them. Let's drop cnt\_loans30.

Let’s drop one of the correlated columns in order to avoid multicollinearity.

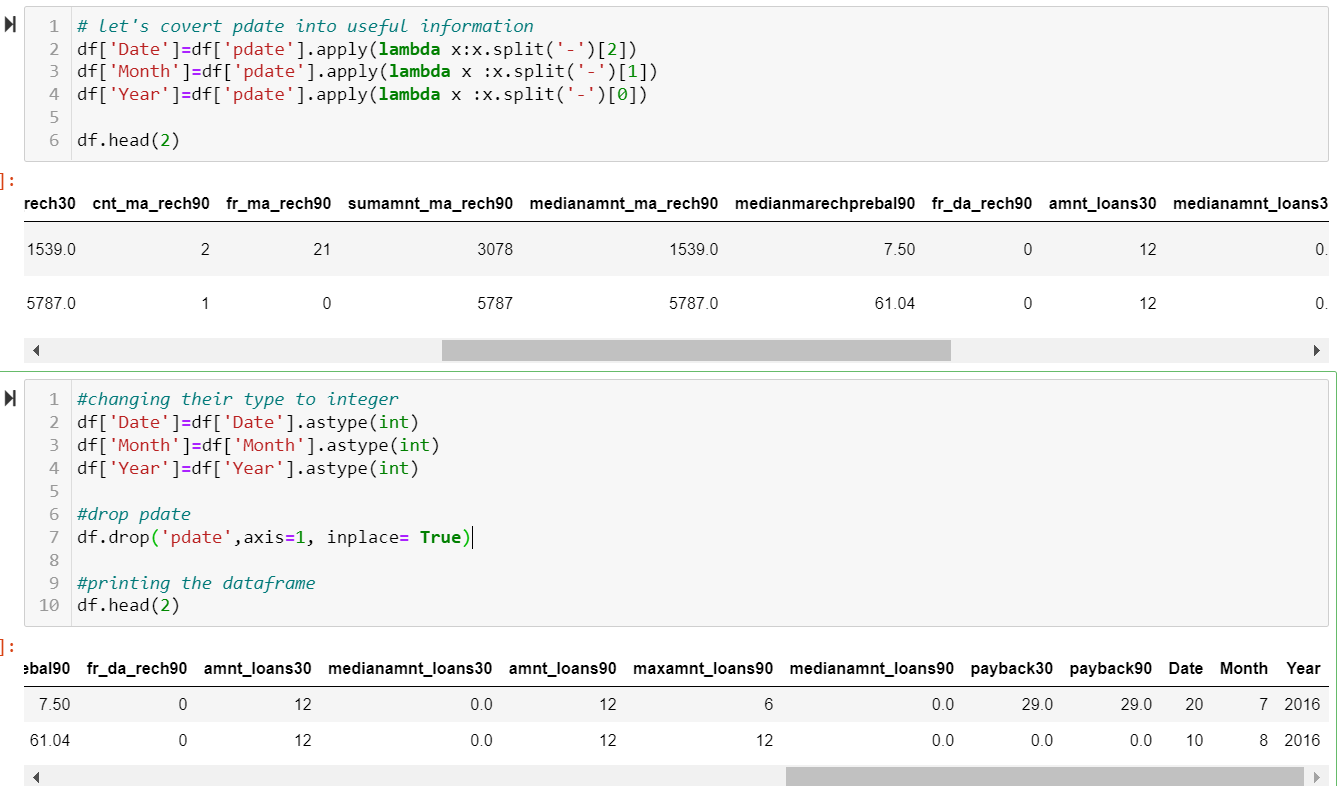




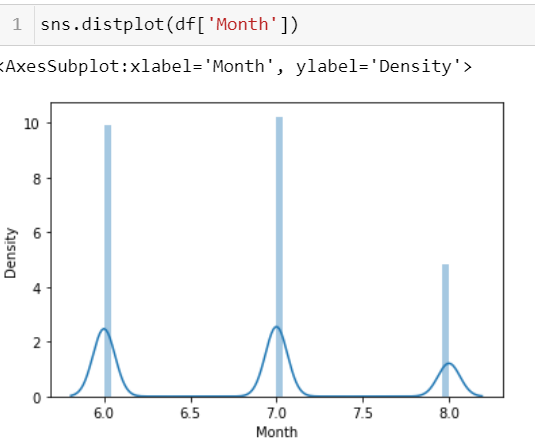


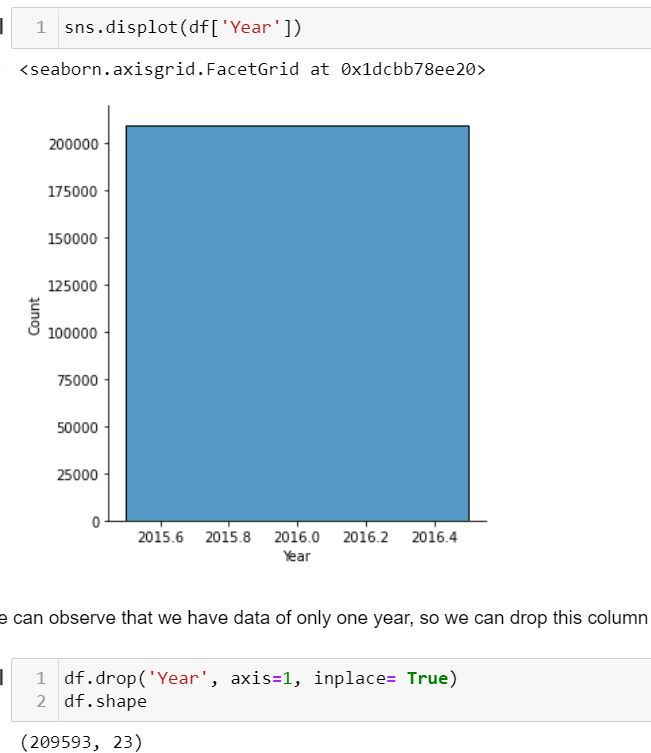
We can observe the positive correlation between them, as value of one increases , the value for other also increases. Also we can see they shows exact plots with label. So it would be good if we drop one of them.

Let’s do some feature engineering with pdate column.

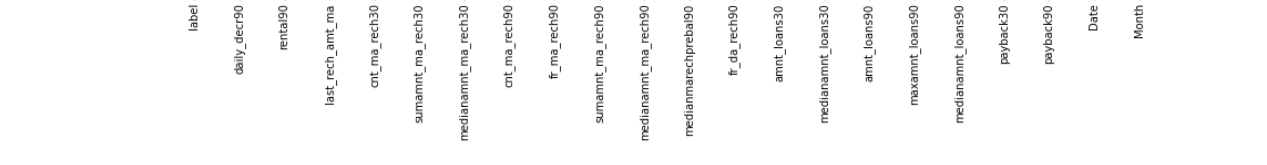


Now let’s see their distribution , to decide on them.

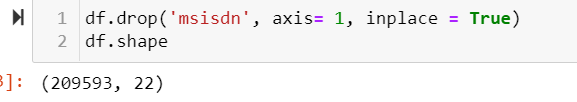
 



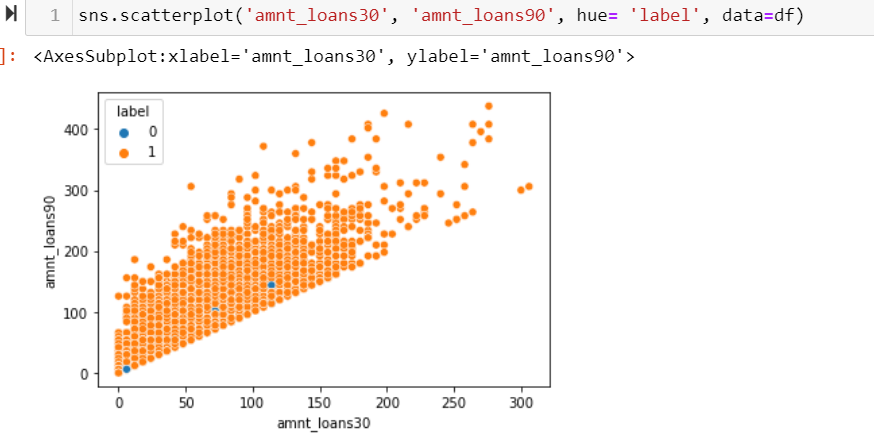
Let’s check correlation matrix one more time, to see how our new columns are correlated with label.

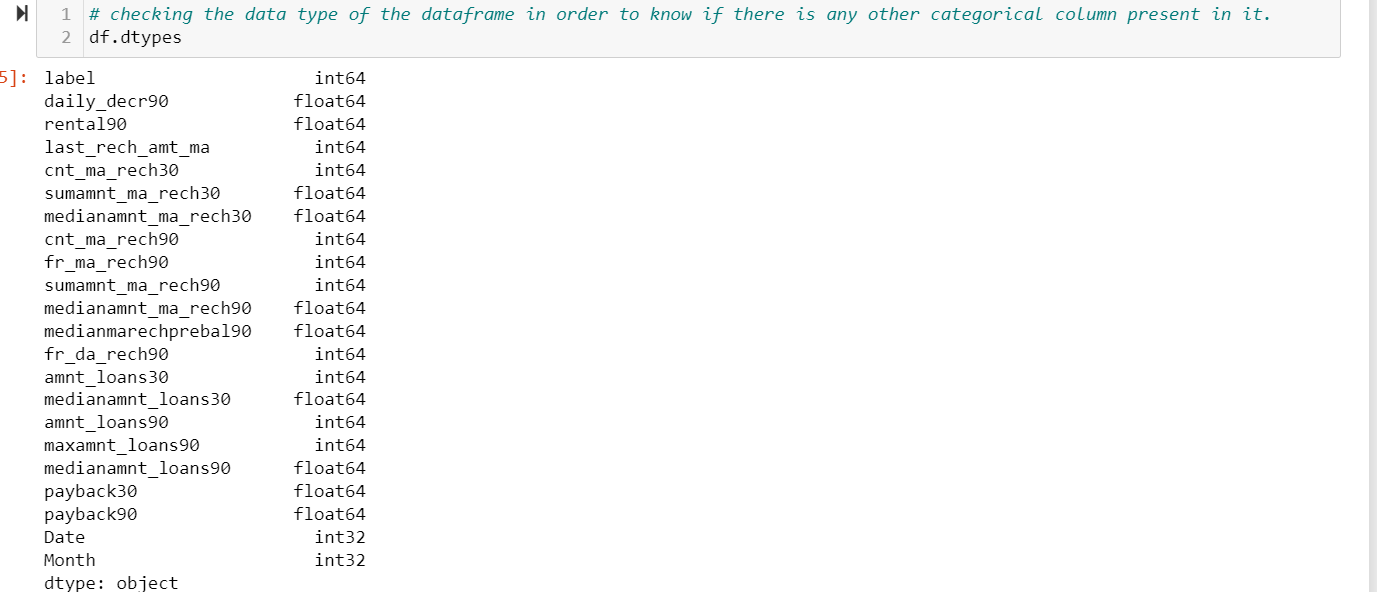
So far, the data looks good. Let's go ahead and do some drop user mobile number column, as it is merely an information about user identity, like name etc. It is important for record keeping, but I don't think it is of any use for ML Model Building.



From below plot , we want to check the correlation between amnt\_loans30- amnt\_loans90, wrt label.



We can observe that both of them are positively correlated, but the variation in the response is quite big, so will keep both of them while going further.

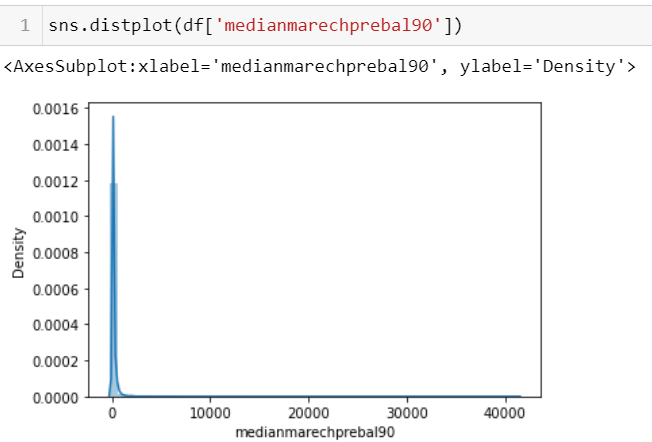
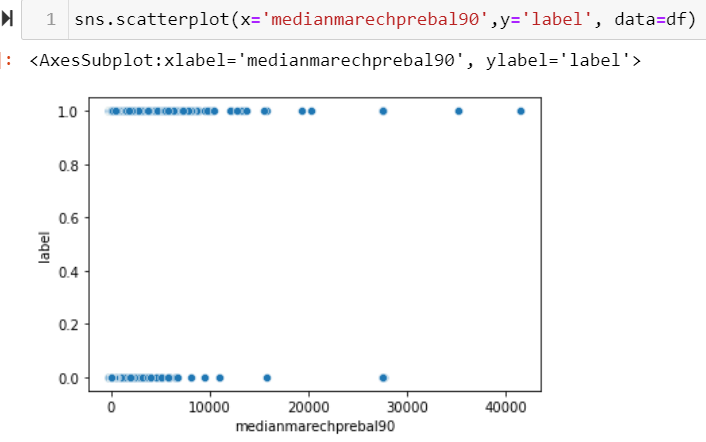


We can observe that all our columns are in numerical data type, So we can safely move ahead with our next step of skewness and outlier detection and removal, as encoding is not required.

## Skewness and Outliers detection and removal:

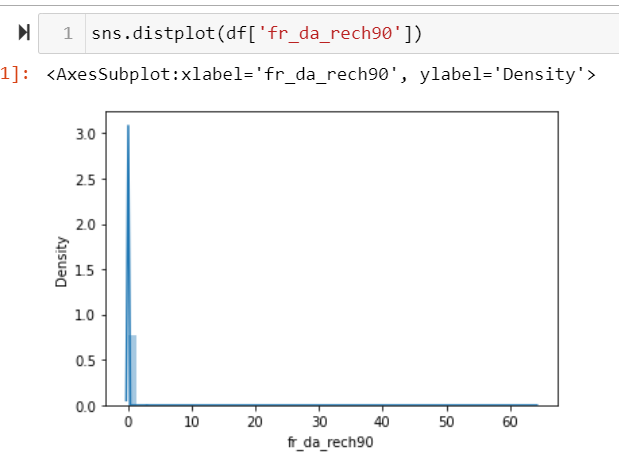
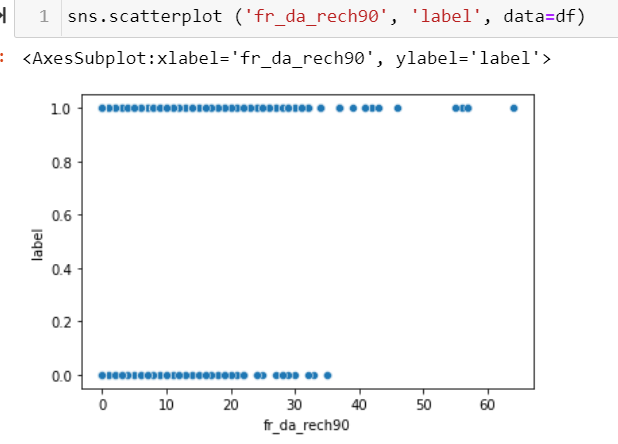


We can observe that our dataset is skewed, with high skewness score on medianmarechpedal90 and fr\_da\_rech90. Let’s visualize their distribution and connection with label before moving ahead with skewness removal technique.

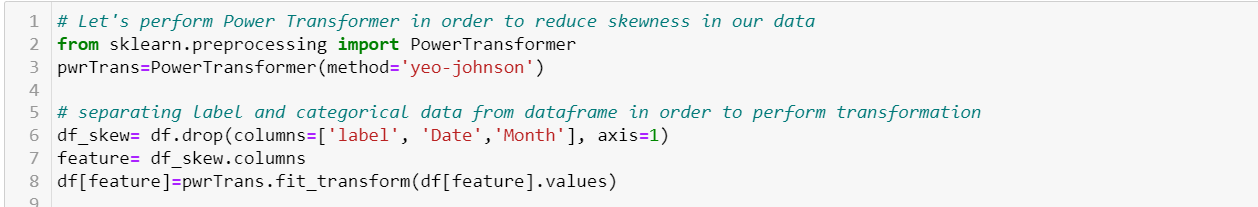
From above two plots it is clear that most of the values of medianmarechprebal90 lies around first highest peak and has wide distribution after that, that might be counted as outlier here, but maybe they are some valuable information, we need to be careful when removing outliers from here.

Now let’s check for same connection in fr\_da\_rech90.

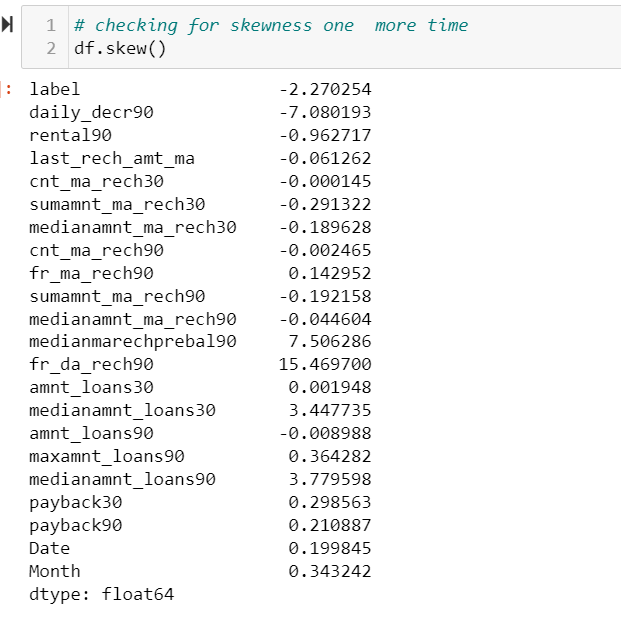
 

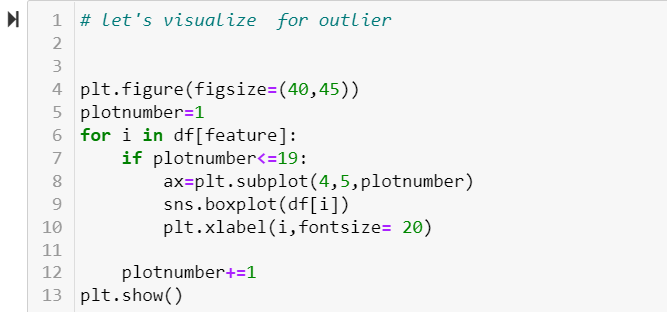
From above two plots it is clear that fr\_da\_rech90 has lots of outliers, but we can observe from the second plot that we got higher value for 1 in higher distribution, so they might be some useful info, we need to be extra careful while dealing with outliers.

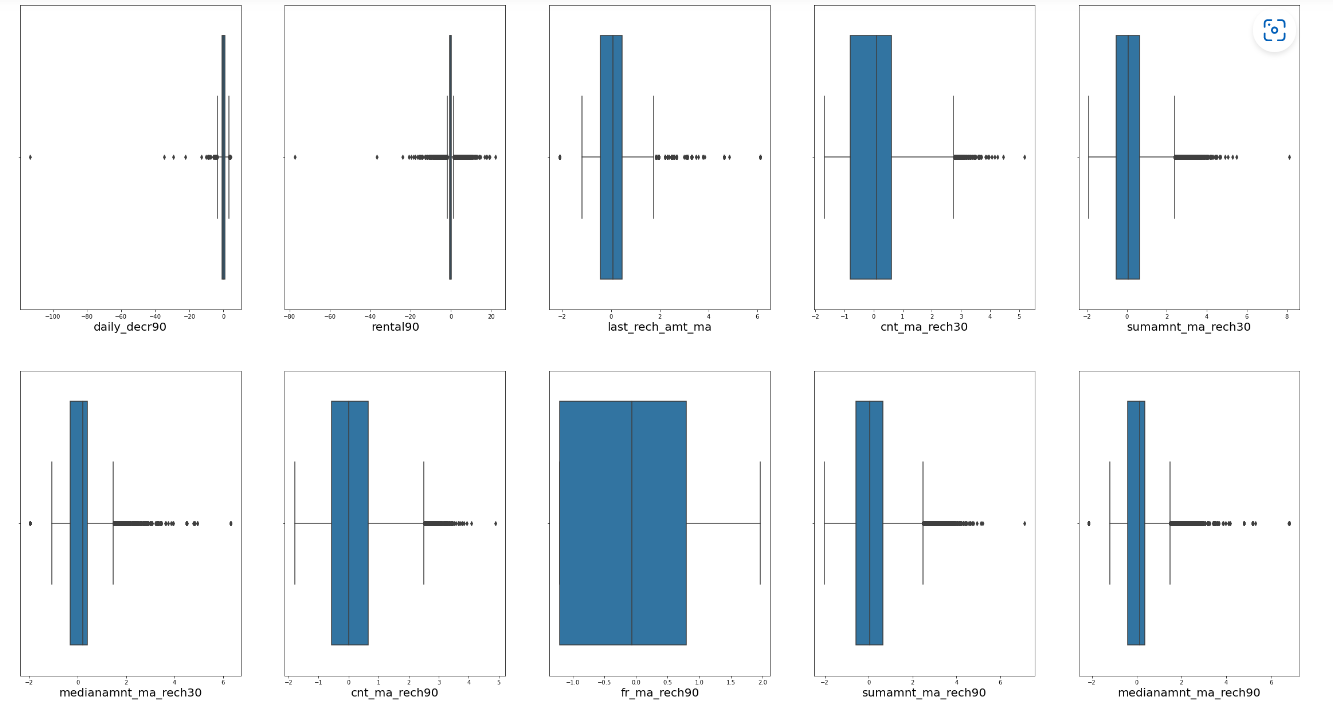
Let’s Perform power transformer in order to reduce skewness from the dataset. It must be noted that skewness and Outliers removal can be performed only on continuous features, not on categorical; features or target variable. So, in order to perform Power Transform to reduce or remove skewness we need to separate label and categorical columns from the main dataframe and then perform the fit\_transform().

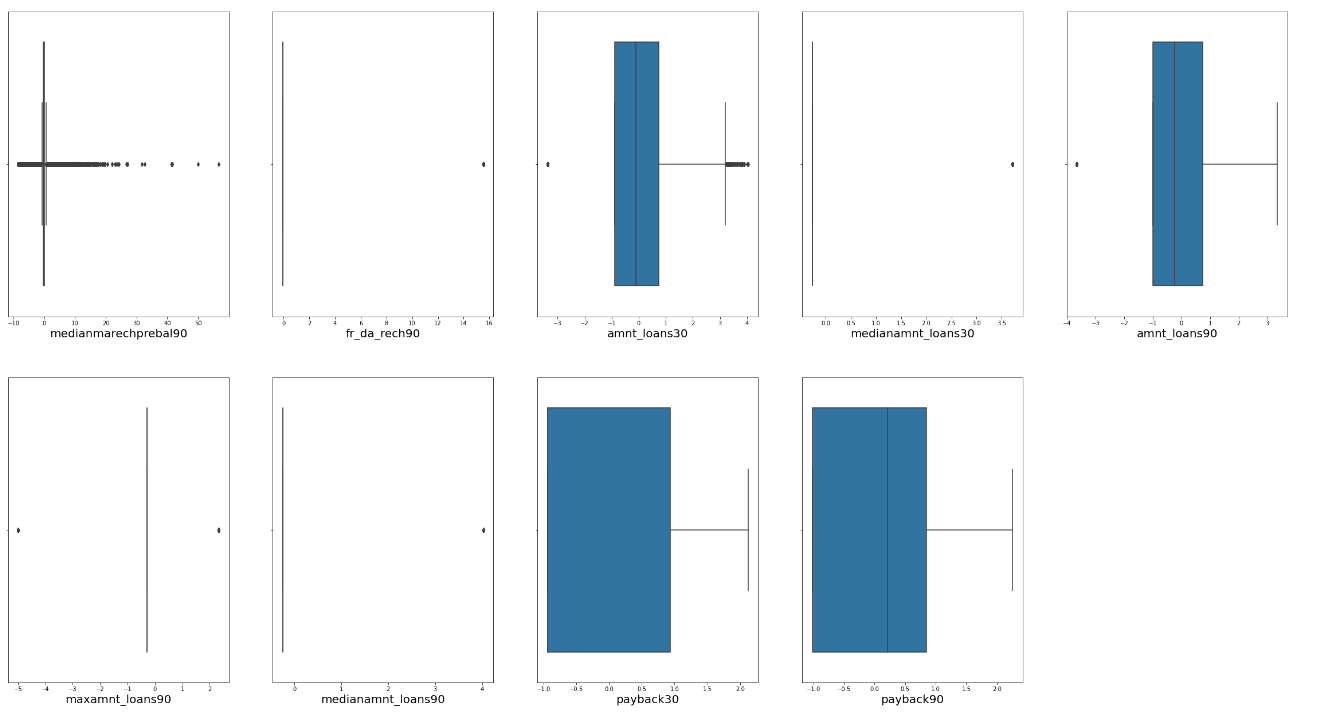


Power Transformation is done, let’s check for skewness in our dataset one more time, before moving ahead.



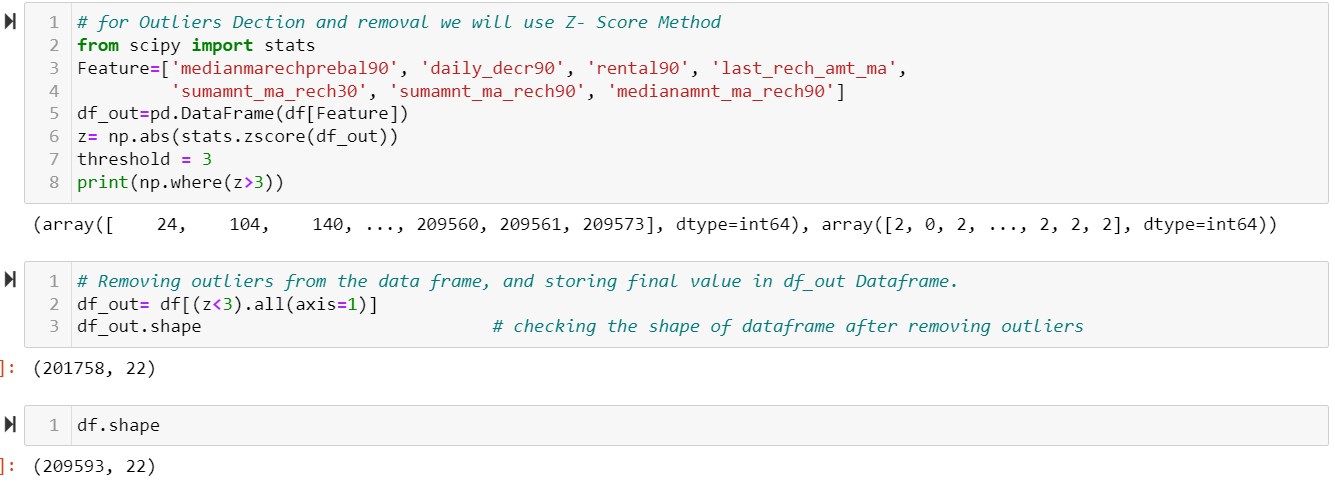
We can observe that skewness has been reduced to great decree, now we can move ahead with outliers detection. We can visualize it using box plot.





We can observe that some columns has some amount of outliers like medianmarechprebal90, daily\_derc90, rental90, last\_rech\_amt\_ma, sumamnt\_ma\_rech30, sumamnt\_ma\_rech90, medianamnt\_ma\_rech90.

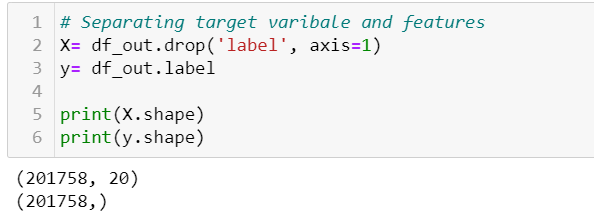
We can use z score technique to remove outliers from our dataset.



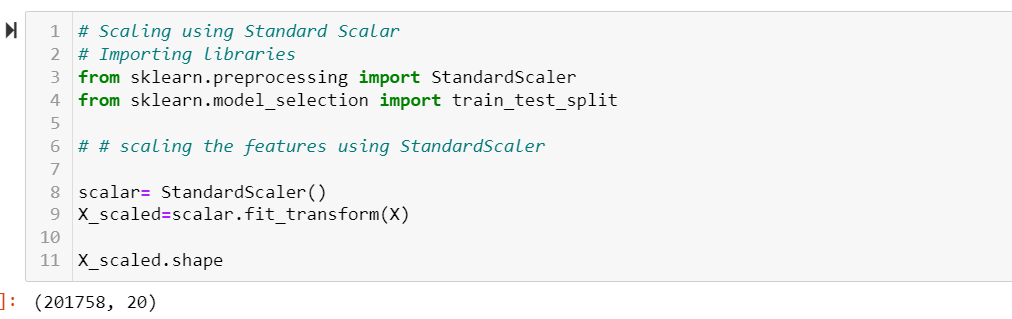
The data loss is around 4% which is acceptable, we will move ahead without clean data, which is df\_out. Let's separate label and features, and scale our data.

## Scaling:

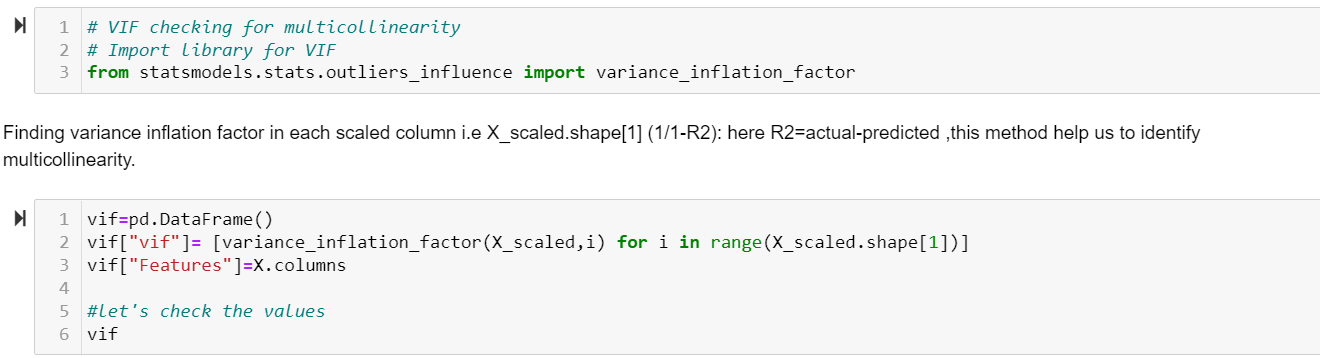
Before scaling the data , we need to first separate target variable from features.

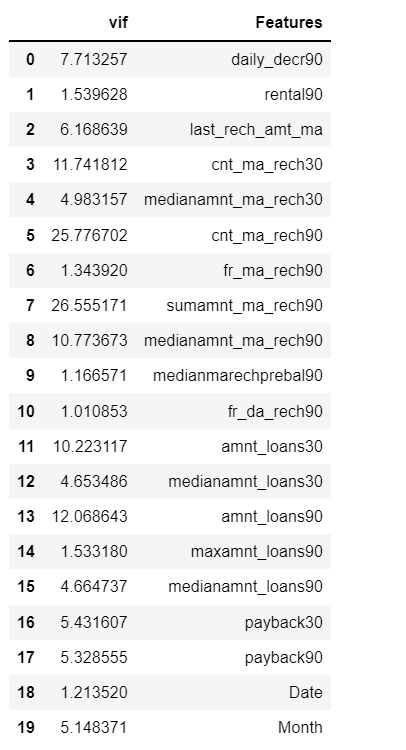


X is the feature dataset while y is the target variable dataset. Now we can scale the feature dataset.

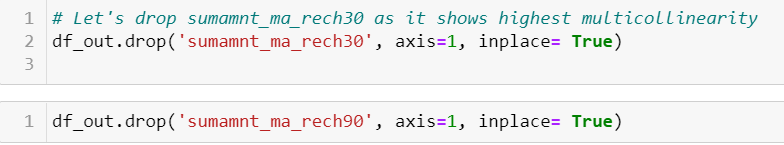


As our feature dataset is scaled, let’s check for multicollinearity using Variance Inflation factor(VIF).

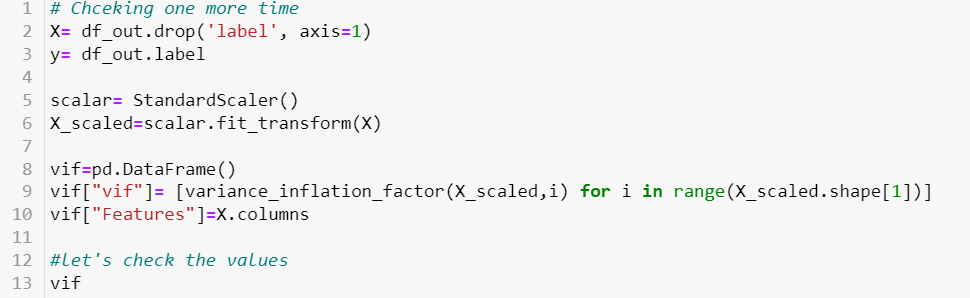




We can see multicollinearity is present in our dataset, let's remove highest score from the list. We are taking the threshold value of +/-10.



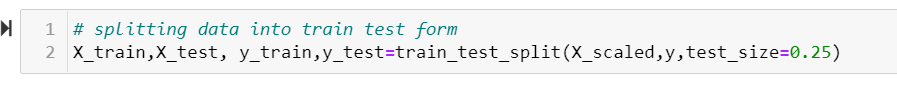
Checking for Multicollinearity one more time.



There is still some amount of multicollinearity present in the data, but let's take the next step, and go for model building.

# Model Building:

First step for any ML Model Building is to splitting data into test and train sets. Here I am splitting data into 25-75% ratio, i.e. 25% of my data is for test set and remain 75% for train the model.

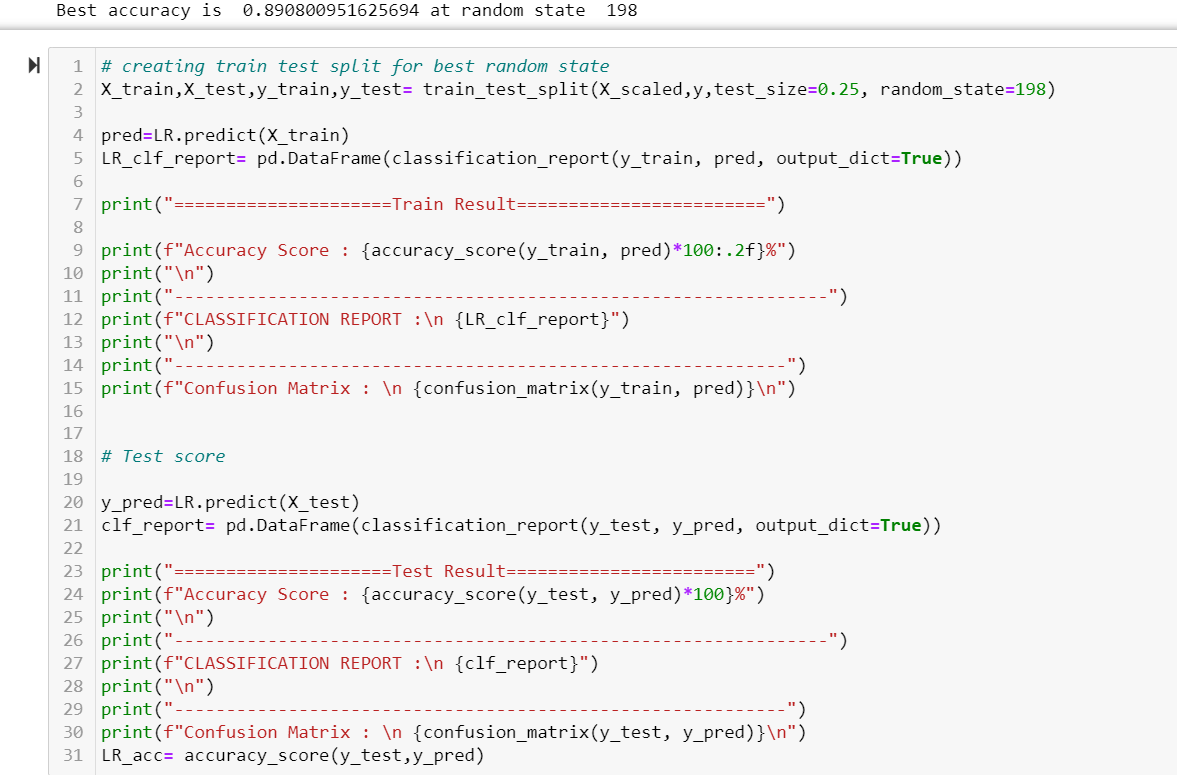


As we know that it is a Classification Problem, let’s starts with Logistic regression Model. As there is not much to hyper parameter tuning in Logistic Regression we start our model with finding the random state with maximum accuracy.

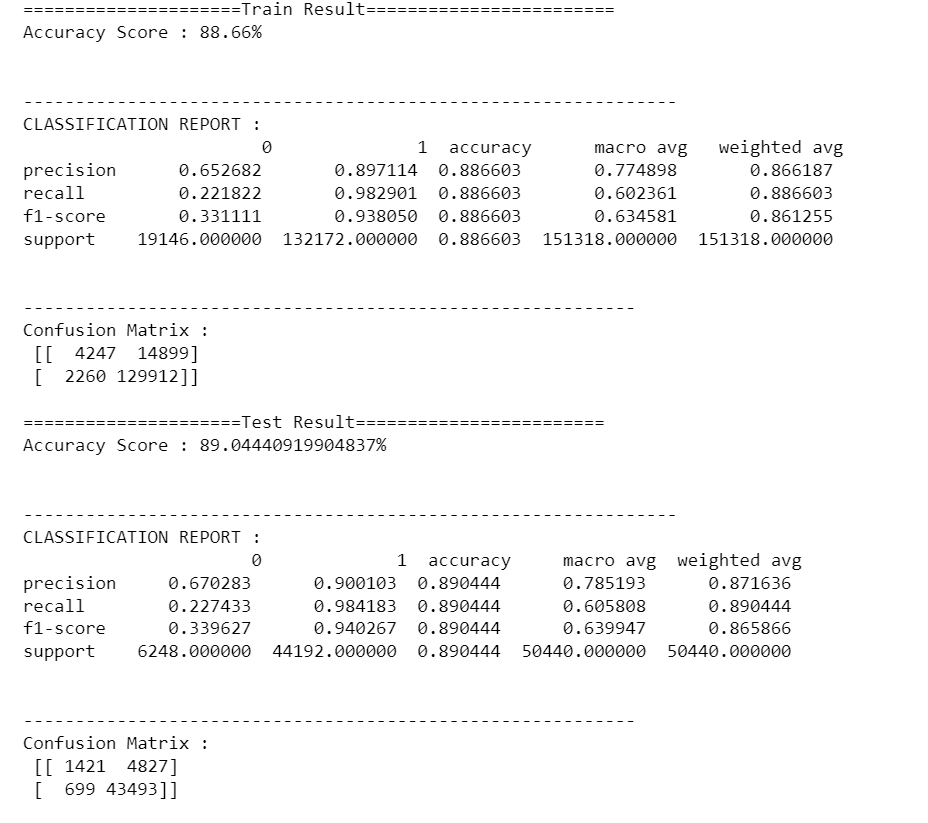
## Logistic regression



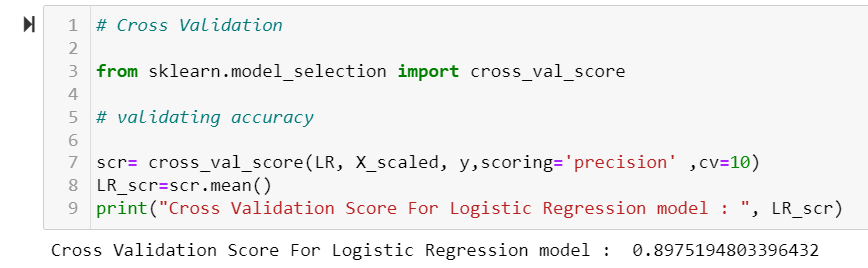
Here we import the library and metrics required to measure the model accuracy.



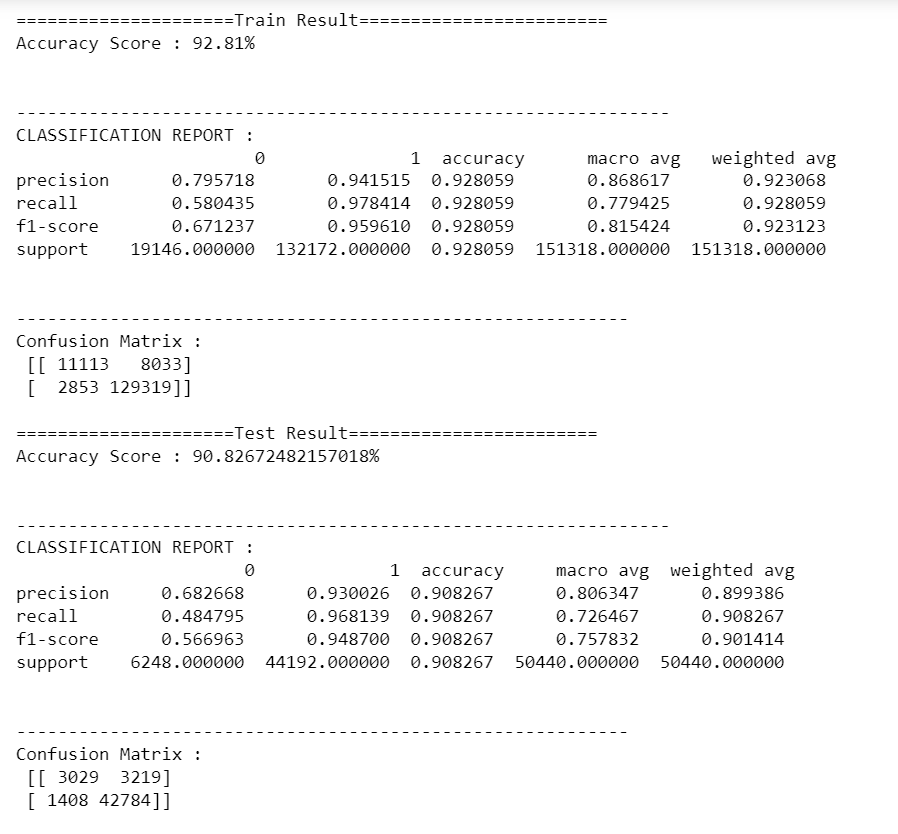
The scores for the above model is as follows:

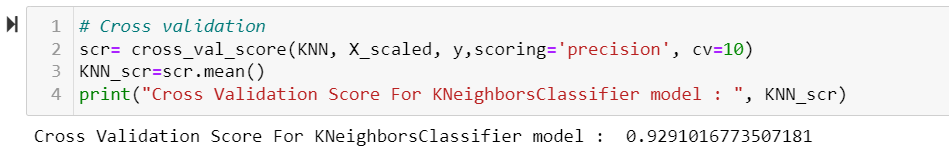


We can see both training score and testing score, as it helps us to see how our model fits. Let’s cross validate our score, here we are using precision score for cross validation, as it helps to justify our model in a better manner.

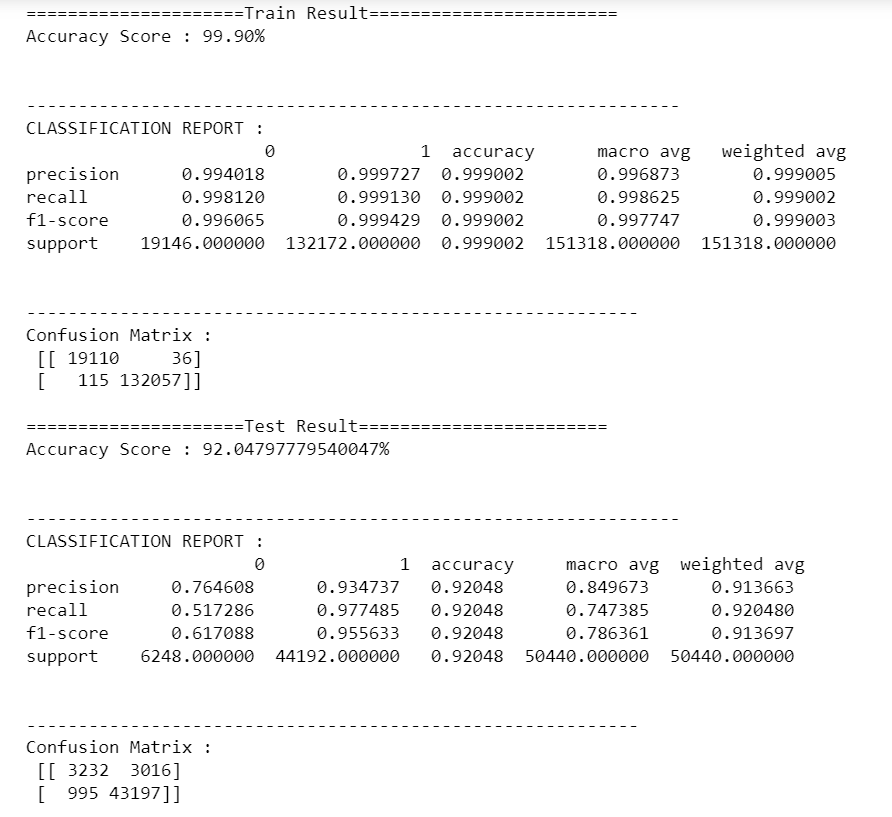
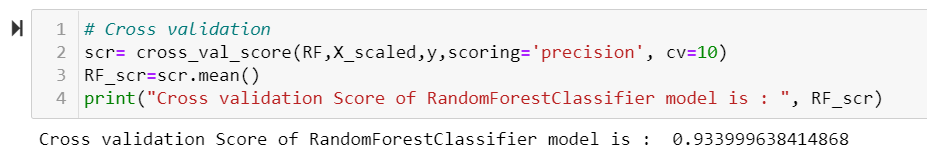


## KNN Classifier



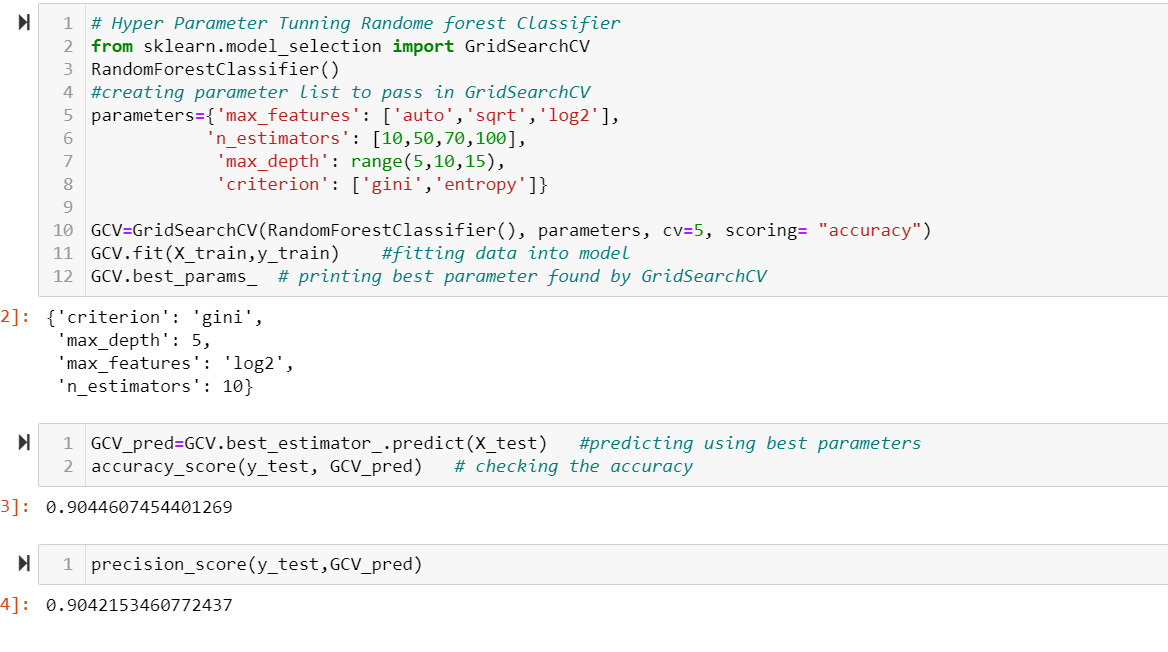
## Ensemble technique (Random Forest Classification)

We can observe that it is performing better than our previous two model, let’s perform hyper parameter tuning in order to improve the efficiency of the model.

We can do Hyper Parameter Tuning either by GridSearchCV approach or by RandomizedSearchCV.

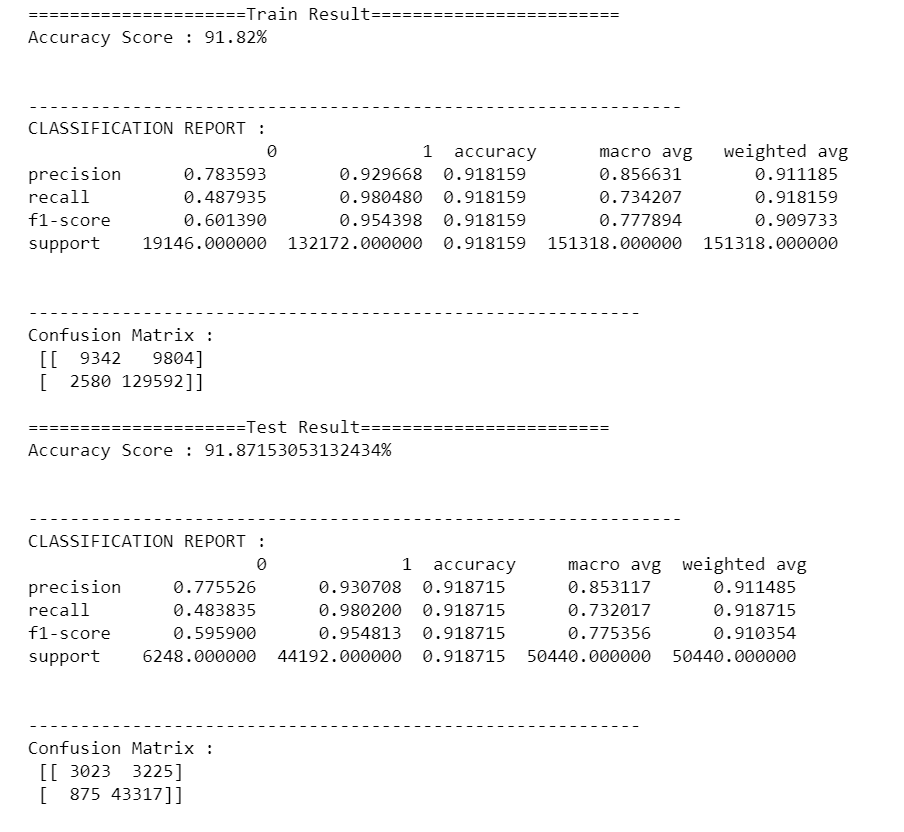
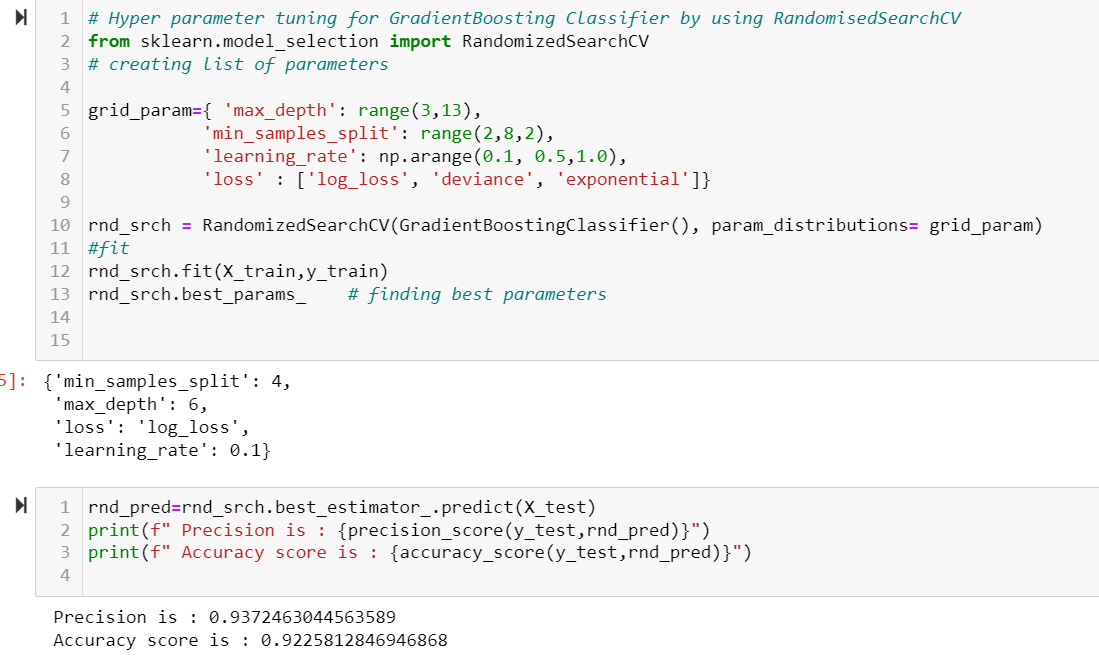
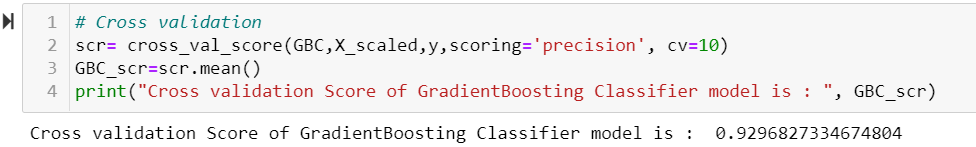
Both the approach has their own pros and cons. Depending on the model and dimension of dataset one can do hyper parameter tuning.



We can observe that with the given parameters range, we don’t find any significant improvement in over all score, So, one can do some alteration and perform it again. So far, I am good with the scores, let’s go ahead and build a boost model, which helps us to compare how our model is performing.

## Boosting Technique (Gradient Boosting Classifier)

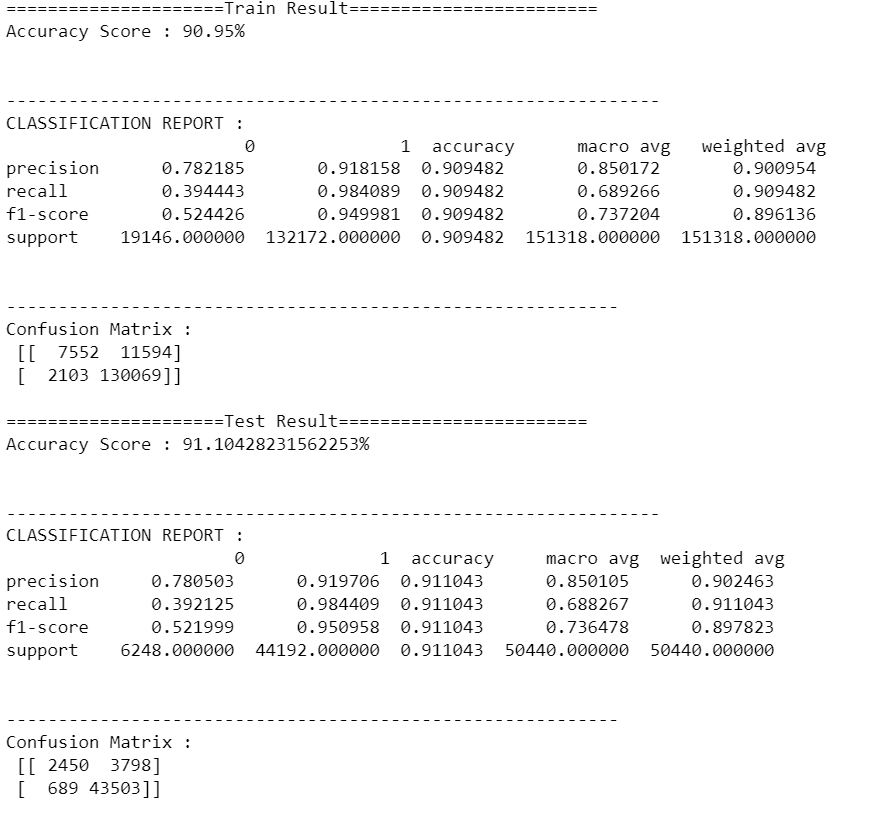


We can observe that it’s score is even better than random forest, and this model is neither overfitted nor underfitted. It means this could be our final model for given dataset. While performing hyper parameter tuning, overall scores also get improved.

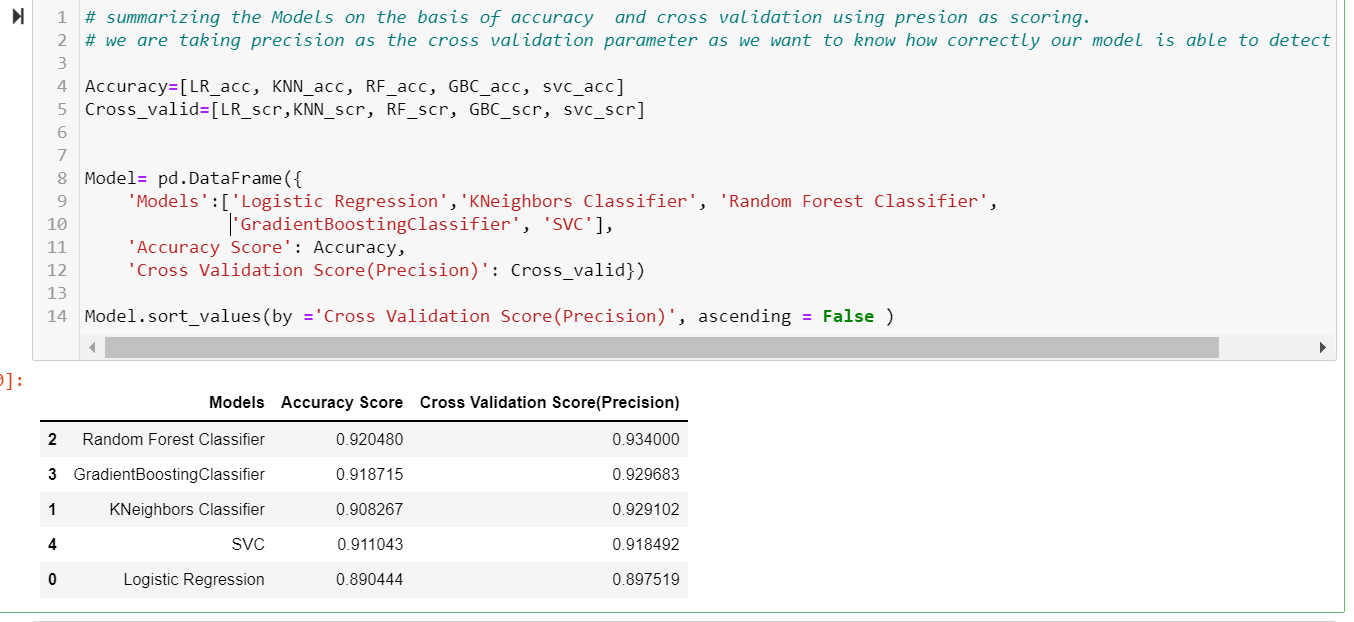
Let’s make one more model before finalizing our decision.

## SVC

Above we made 5 algorithm to decide on best model, let’s summarize each of them , and then we will decide which to consider as our final model.

# Summarizing Each Model :



From above we can see that RandomForestClassifier and GradientBoostingClassifier works best for our data set, To decide on which model is our final model, let’s have a look on the training score of both the model as well. As we don’t want our model to be over fitted or under fitted.

From there we can see that RandomForest training accuracy score is 99.90% while testing score is 92.04% , as both the scores are well above 90% we can say the model is performing really well. We did perform HYPER PARAMETER Tuning in order to improve the efficiency of the model.

And for Gradient BoostingClassifier, we can see that the training score are almost in line of testing scores, from classification report as well, this is confirm that our model neither overfitted nor under fitted. We also perform hyper parameter tuning here using RandomizedSearchCV, it is fastest and our dataset is quite big to perform GridSearchCV.

We can also decide on model by ploting roc auc curve.

# Conclusion:

As we can observe that RandomForest Classifier gives both highest accuracy score and precision score, it will be our final model.

We need to work more on hyper parameter tuning, by changing the range of parameters and run the function few more times we get desired increase in the efficiency/.

We can go ahead and save RandomForestClassifier Model.