# Blog on: Loan Application Status Prediction



**Author: Pratiksha Potghan**

**Batch: 1843**

**Data trained Education**

**Introduction:**

Loans are core business of banks. The main profit comes directly from the loan’s interest. The loan companies grant a loan only after the process of validation and verification.

In this article we are going to solve the problem of Loan Application Status Prediction. Here we will build a predictive model which will predict if an applicant is able to repay the lending company or not.

This is a Classification problem, in which we will need to classify whether the loan will be approved or not.

Classification refers to a predictive modelling problem where a class label is predicted for a given example of input data.

It is a task of Machine Learning which assigns a label value to a specific class and then can identify a particular type to be of one kind or another. The most basic example can be of the mail spam filtration system where one can classify a mail as either “spam” or “not spam”.

**Previous System:**

Bank employees check the details of every applicant manually, and then give the loan to the eligible applicants. This process takes a lot of time. There are also chances of human error. This are the disadvantages of the previous system.

**Proposed System:**

To deal with the problem, we have developed automatic loan prediction Machine Learning models and techniques. We will train the machine with the previous dataset, so that machine will analyse and understand the process. The machine will check the eligibility and give us the results.

Advantages:

* Time period for loan sanctioning will be reduced.
* The total process will be automated and thus the human error will be reduced.
* Time consuming process.

**Problem Statement:**

This dataset includes details of applicants who have applied for loan. Below listed variables are our Independent Variables (Features) and Dependent Variables (Label):

**Independent Variables:**

1. Loan Id
2. Gender
3. Married
4. Dependents
5. Education
6. Self Employed
7. Applicant Income
8. Co-applicant Income
9. Loan Amount
10. Loan Amount Term
11. Credit History
12. Property Area

**Dependent Variable:**

* Loan Status

**Problem Statement Definition:**

* The details of the applicants are given in the dataset which has 12 feature columns and 1 label “Target variable”.
* We will have to build a machine learning model to predict whether the loan of the applicant will be approved or not on the basis of provided details.

**Software requirements and tools used:**

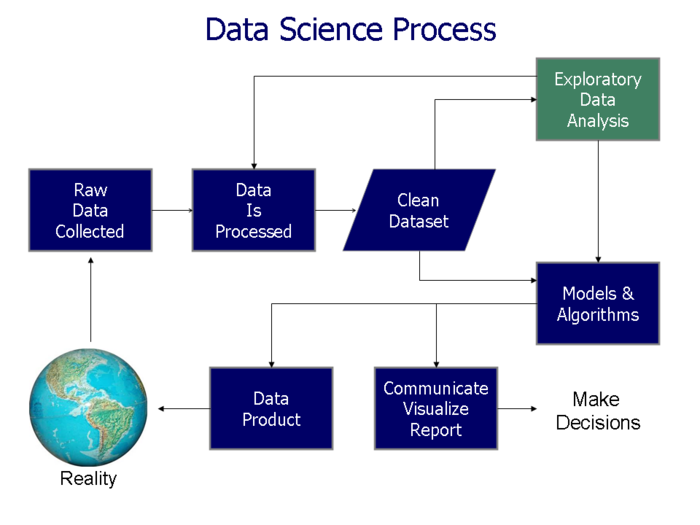
* Jupyter Notebook

**Libraries Used:**

* Pandas
* Numpy
* Python
* Matplotlib
* Seaborn
* Scikit Learn

**Table of Contents :**

1. Getting the system ready and loading the dataset.
2. Understanding the data and problem statement and requirements.
3. Exploratory Data Analysis (EDA)
4. Missing Value and Outlier treatment using zscore.
5. Checking Statistical Data
6. Data Visualizations using several plots.
7. Encoding the categorical data.
8. Checking Correlation
9. Seperating Label and Features
10. Feature Scaling
11. Model Building
12. Evaluation metrics for Classification models
13. Hyper Parameter Tuning on best model
14. AUC-ROC Curve
15. Confusion Matrix
16. Saving the Model



**Importing the required libraries:**

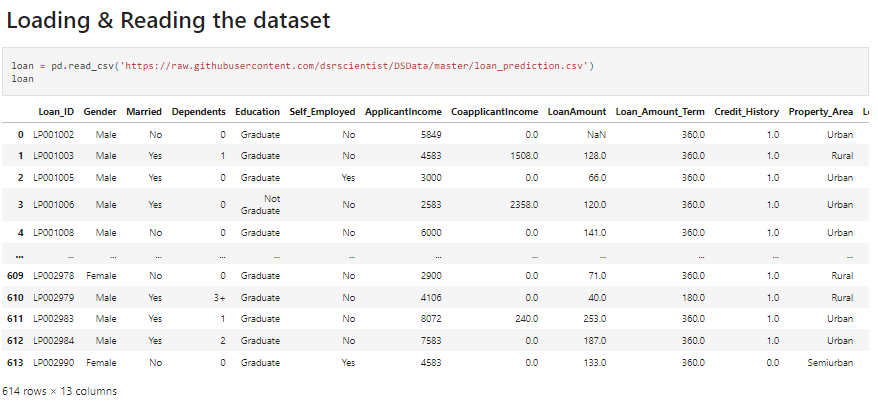
Firstly we will import the libraries which are required to analyse the data.

* We have imported **pandas** as pd, which will help in creating dataframes and analysing data. Also it is the most popular data wrangling package.
* After that we have imported **numpy** as np, which supports for multidimensional arrays.
* **Matplotlib** and **Seaborn** are the libraries which are used for Data visualizations.
* **Joblib** is used for saving the model and it is a set of tools which provides lightweight pipelining in python.
* **StandardScaler** is a python sklearn library which is used to standardize the data values into a standard format.
* **LabelEncoder** is used when there are 2 possible values of a categorical feature, for eg, Yes or No, Male or Female, etc. By using this our categorical column with data ‘Yes’ or ‘No’ will be converted into numerical data as ‘0’ and ‘1’.
* **OrdinalEncoder** is used to assign each unique category value to an integer value.
* We have also used **train-test split** algorithm which will estimate the performance of machine learning algorithms.
* We have also used **Cross\_validation\_score** to estimate the performance and accuracy of our ML models.
* We have used **zscore** technique for removal of outliers, as zscore gives us an idea of how far from the mean a data is.
* **Confusion matrix** is also used here, which is also called as error matrix.
* **Classification report** is used to measure the quality of predictions from a classification algorithm. By using this we will understand that how many predictions are True and how many are False.
* We used **accuracy score** to measure the model performance in terms of measuring the ratio of sum of true positives and true negatives out of all the predictions made.
* Using **R2 score** we can check the proportion of the variance in the dependent variable that is predictable from the independent variables.
* **Mean squared error** is used to average the square of the errors.
* ***Logistic Regression, Random Forest Classifier, SVC, Decision Tree Classifier, SGD Classifier, K Neighbors Classifier, Ada Boost Classifier, Extra Trees Classifier, Gradient Boosting Classifier, XGB, LGB*** are some of the Classification Algorithm used to build the Classification Model.



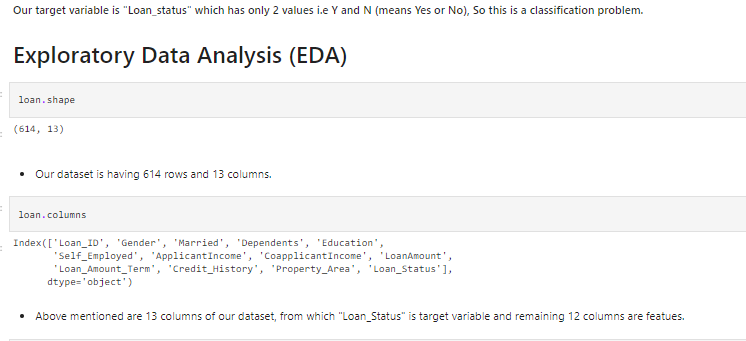
**Loading the Dataset:**

* We have loaded the dataset which was in csv format.
* We have imported it in ‘loan’.
* After observing the dataset we can see that we have both categorical data and numerical data as well.
* Our target variable is **“Loan\_Status”.**
* “Loan Status” has 2 categories of data i.e Y and N, so we can conclude that this is a Classification problem.



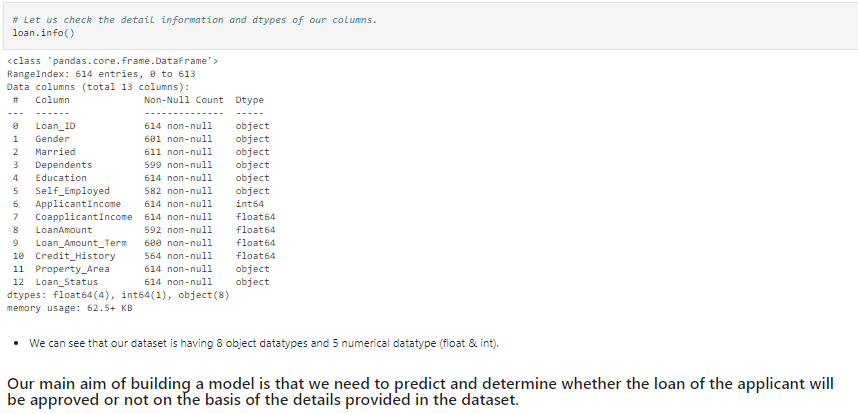
**EDA:**

* We have done the exploratory data analysis on our dataset to check all the details.
* We have first used *“loan.shape”* to check the number of columns and rows of our dataset and we came to know that there are 614 rows and 13 columns in our dataset.
* We also used *“loan.columns”* to check all the name of the columns in our dataset.



**Detail Information:**

* We have used *“loan.info()”* to check the detail information of our dataset, for eg. Non-null count, datatypes count, etc.



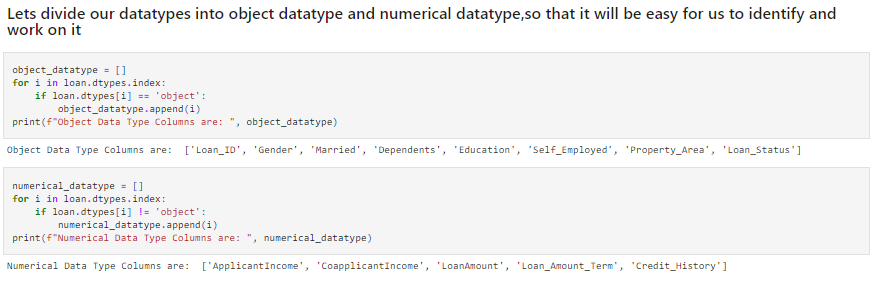
**% Yes and No in Target Variable:**

* By using below syntax I have checked the ***total rows count, columns count, total YES in the target variable, total NO in the target variable and total percentage of YES loans*.**
* It gave me below details.



**Datatypes:**

* I have used below syntax to separate the categorical column i.e object type of data and numerical columns.
* After separating, I have stored the object data in “*object\_datatype*” and numerical data in *“numerical\_datatype”.*



**Unique values:**

* *“loan.nunique()”* gives me the number of unique data present each column.

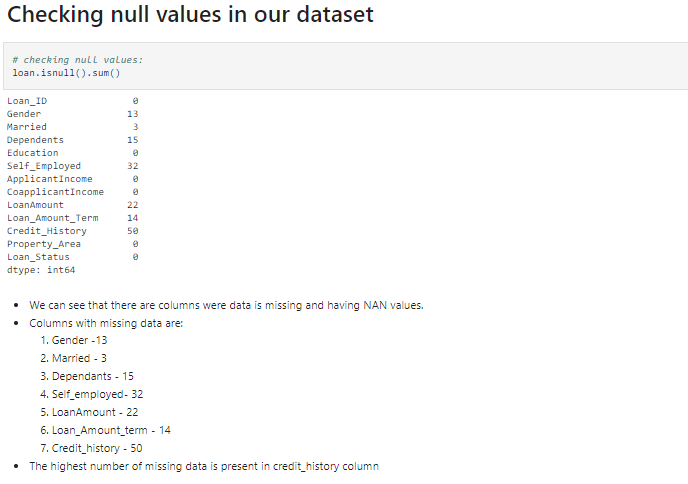


**Duplicate values:**

* This function is used to check the duplicate values in our dataset.

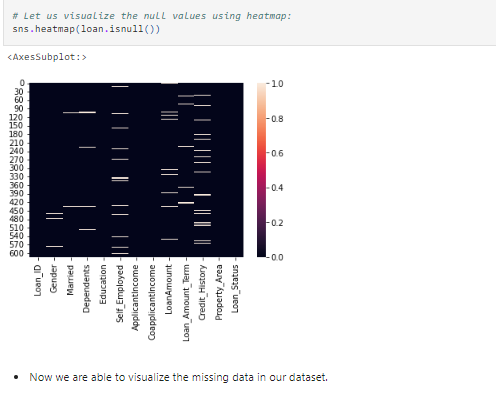


**Null values:**

* By using ***“loan.isnull().sum()”*** I am able to get the null values present in the dataset.
* In this case we had null values in columns *“Gender, Married, Dependents, Self-employed, Loan amount, Loan amount term and credit history”.*
* The highest number of missing values were present in *“Credit\_History”* column.
* 

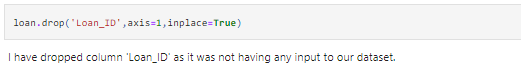
**Visualizing null values using heatmap:**

* By using this function we were able to see and visualize the missing values in the columns.



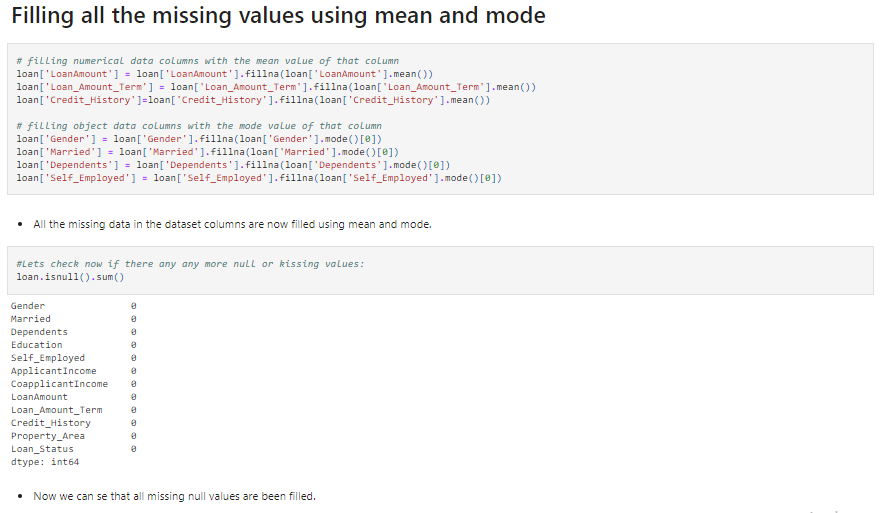
**Dropping Irrelevant columns:**

* I have dropped the column *“Loan\_ID”* which has no relevant input to our dataset. It is just unique numbers given to all the applicants, so we will drop it using **“drop”** function.



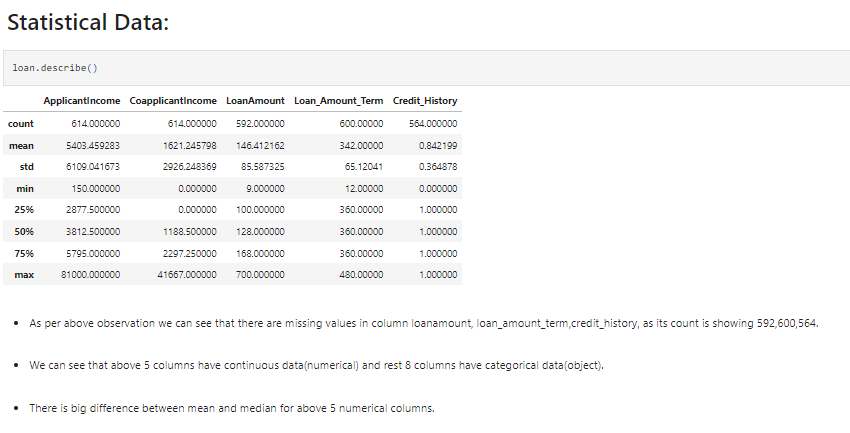
**Filling missing data:**

* After checking null values we came to know that we need to fill all the missing data as we require this columns for building model.
* For filling missing values, we used mean and mode method.
* For the numerical data column we used mean function to fill the data and for the categorical data columns we used mode function to fill the data.
* After filling the missing values, we again checked with loan.isnull().sum() function to confirm if there are any more missing data and we found that now all the data are filled.



**Statistical data:**

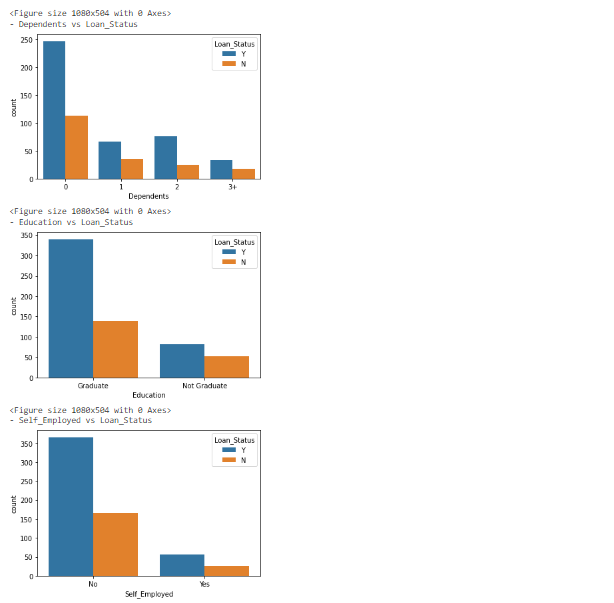
* Here we have used *“loan.describe()”* function to check all the statistical details like count, mean, standard deviation, minimum, 25%, 50%, 75% and maximum values of numerical datatypes.

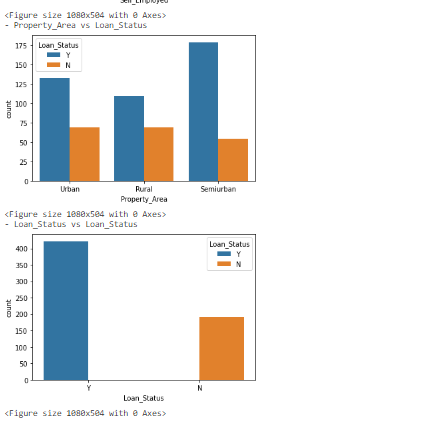


**Data Visualization**

1. **Countplot:**







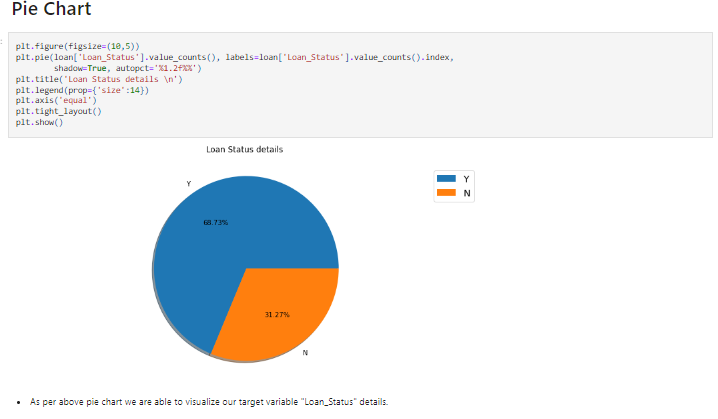
**Observations:**

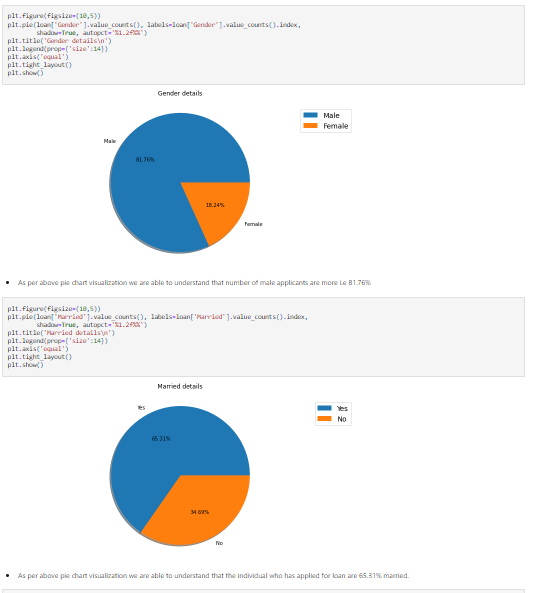
* We can observe that high number of the approved applicants are Male.
* High number of the approved applicants are married.
* High number of the approved applicants have 0 dependents.
* High number of approved applicants are Graduate background.
* High number of approved applicants are not self-employed.
* Highest number of approved applicants are from Semi urban area.

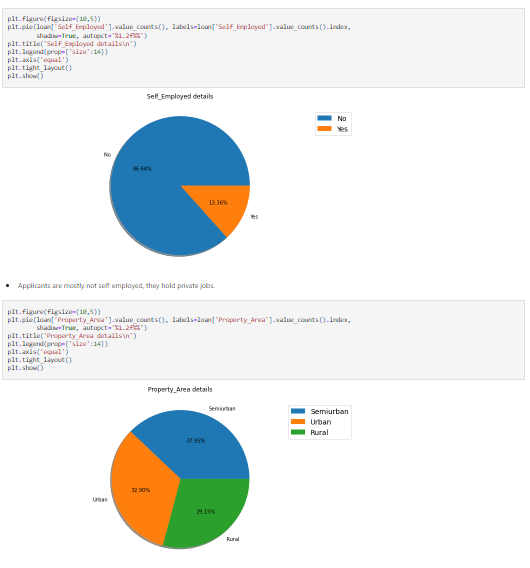
1. **Pie Chart:**

**Observations:**

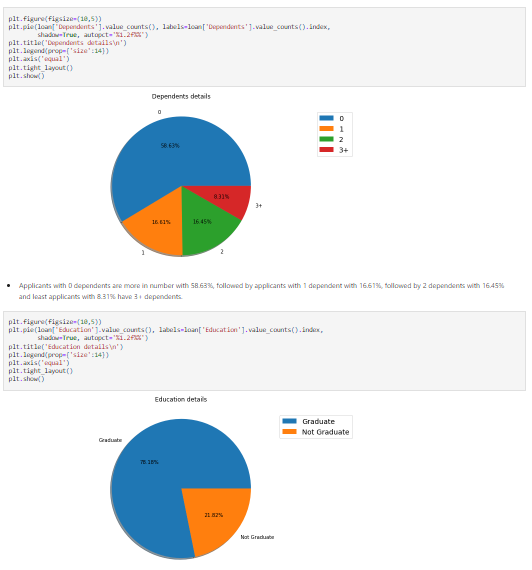
* Pie chart gives us clear idea and ratio of 2 categories.
* **68.73%** of loan status is YES and **31.22%** of loan status is NO.
* **81.75% approved** loan applicants are Male and **18.24%** are female.
* **66.31% of** approved loan applicants are married.



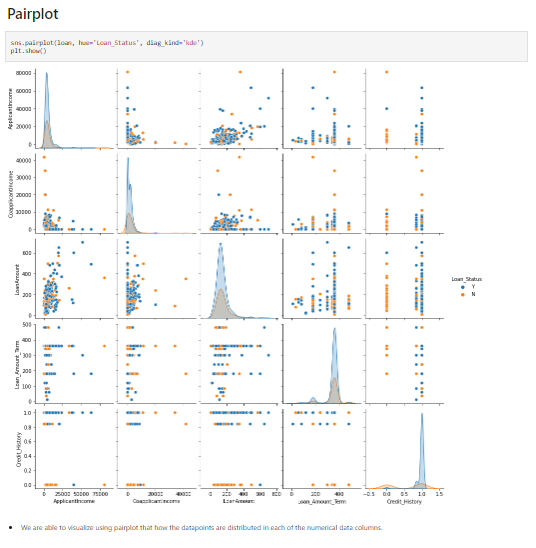




* **86.64%** of approved applicants are not self-employed, so they must be job holders.
* **37.95%** of approved applicants are from Semi Urban areas.
* **58.63%** of approved applicants have 0 dependents.
* **78.18%** of approved applicants are graduate by education.



1. **Pairplot:**



**Observations:**

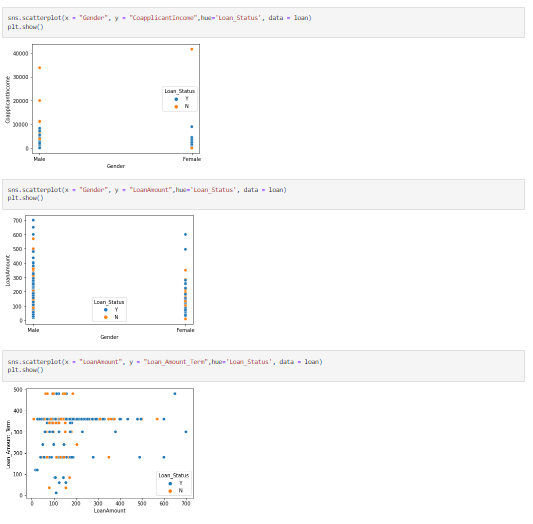
* By using pairplot we can see that how the data points are distributed in each of the numeric datatype columns.

1. **Scatter plot:**

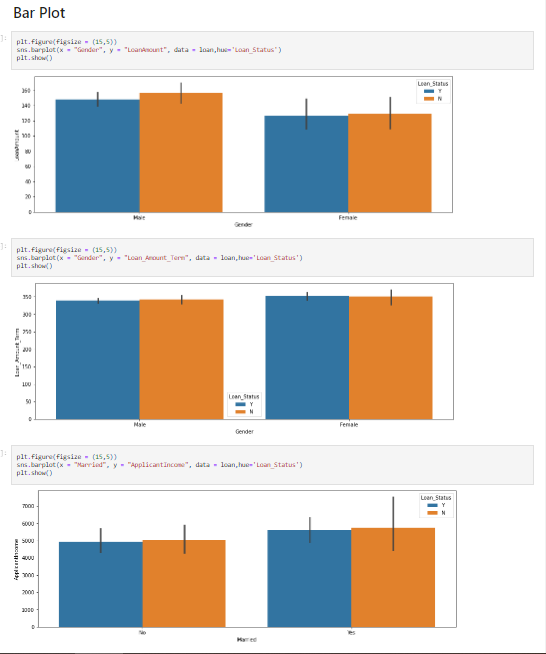
:

**Observations:**

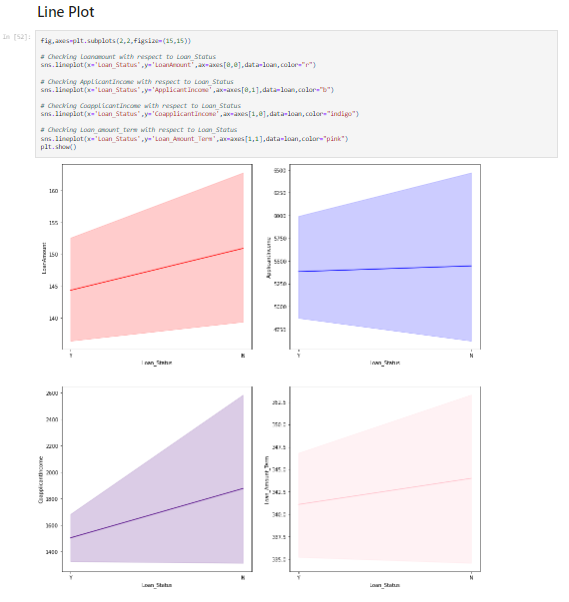
* All the columns are strongly, positively and linearly related with each other.



1. **Barplot:**



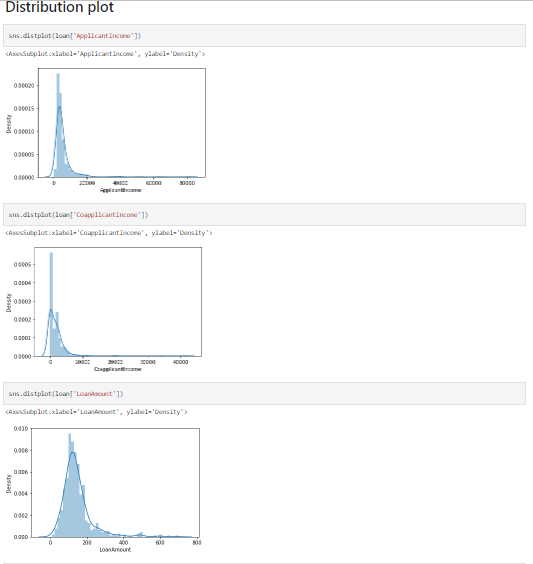
1. **Lineplot:**



**Observations:**

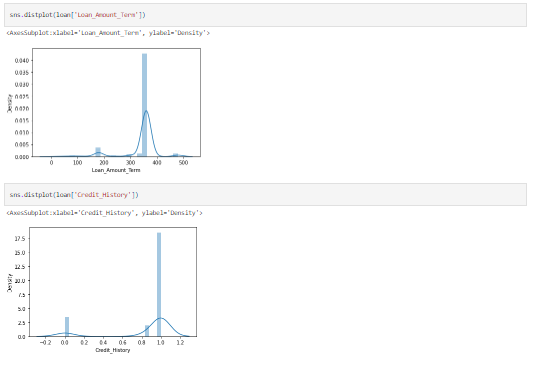
* We can observe that Loan amount, Applicant Income, Coapplicant income and Loan amount term are all linearly and positively correlated with our target variable ‘Loan Status’.
* We can see that if applicant and coapplicant incomes are more there are more chances of loan getting approved.

1. **Distribution plot:**

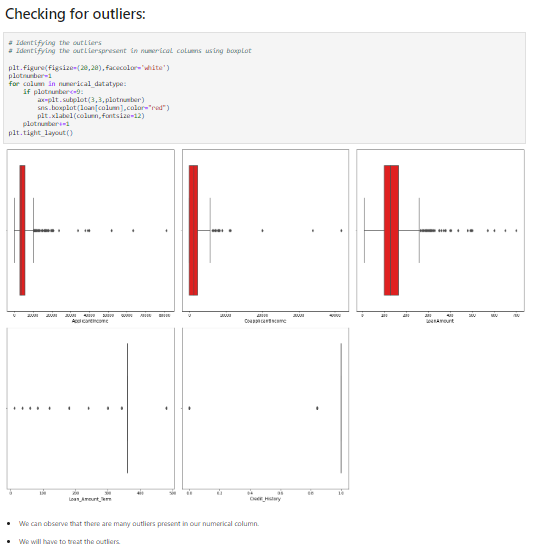


**Observations:**

* By using distribution plot we are able to notice the skewness present in our numerical columns, which we need to treat in future steps.



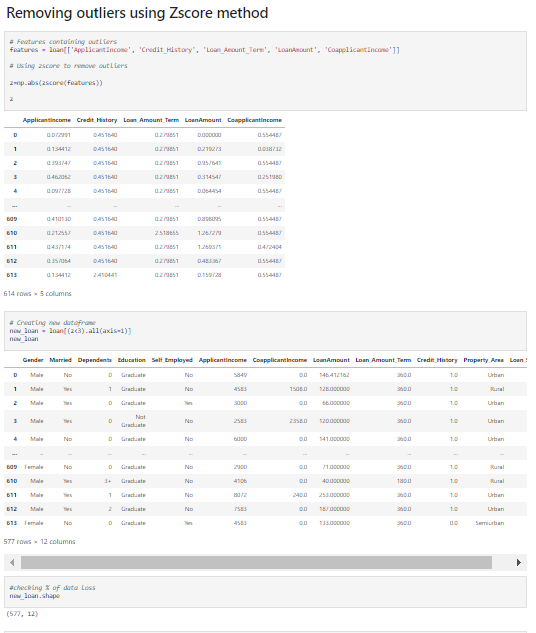
1. **Checking Outliers using boxplot:**



**Observations:**

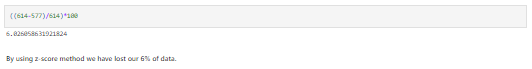
* We can see that all the numerical columns have outliers present in it.
* Higher number of outliers are present in Loan amount column.

**Removing Outliers:**

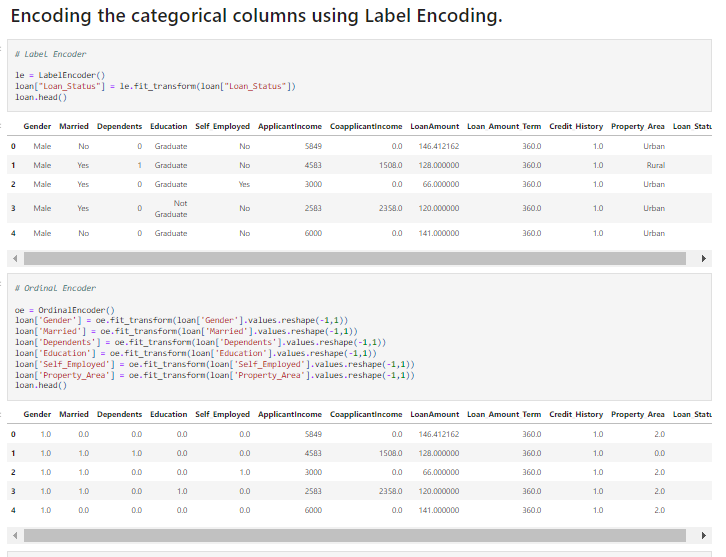


**Observations:**

* We have used zscore method for removing the outliers present in the numerical columns of our dataset.
* We have now treated the outliers and below is the % data loss we got after removing the outliers.
* 6% of data is lost while treating the outliers, but we can afford this because it do not affect our dataset.



**Encoding:**



**Observations:**

* We have used the LabelEncoder() and OrdinalEncoder() method.
* We used label encoder method for our target variable to get the values ‘Y’ and ‘N’ as ‘0’ and ‘1’.
* We used ordinal encoder method for all the remaining categorical columns to get a unique integer values instead of any categorical data.

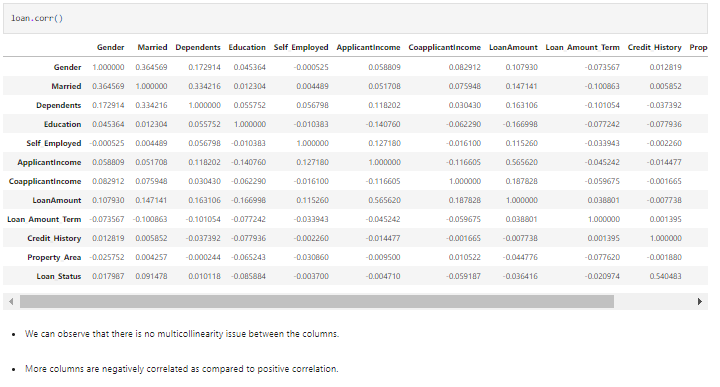
**Skewness:**



**Observations:**

* Skewness is a way of estimating and measuring the shape of distribution.
* Its value can be either positive or negative.
* A positive skew will indicate that the tail is on the right side and it will extend towards the most positive values.
* In this case positively skewed columns are dependents, education, self-employed, applicant income, co-applicant income, loan amount.
* On the other hand, negative skew indicates tail on the left side and will extend towards more negative side.
* In this case negatively skewed columns are gender, married, loan amount term, credit history, property area and loan status.
* If there is zero value then it indicates that there is no skewness in the distribution.

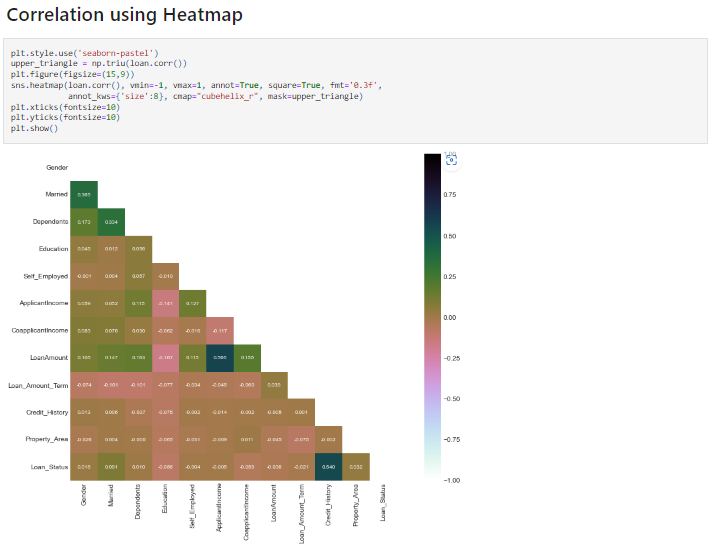
**Correlation:**



**Observations:**

* Correlation is a statistical measure that expresses the extend to which two variables are linearly related with each other.
* Here we can see that gender is 0.017 correlated with loan status, marital status is 0.091 correlated with loan status, Coapplicant income is -0.059 correlated with loan status and so on.
* We can also visualize the correlation using the heatmap shown as below.

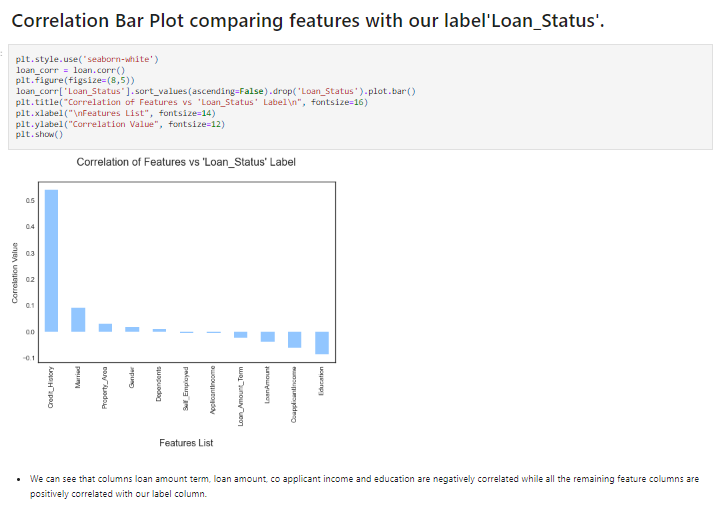
**Correlation using Heatmap:**



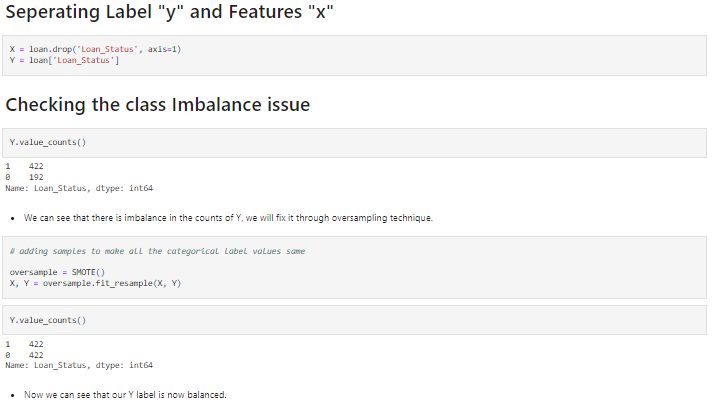
**Correlation Barplot:**

**Observations:**

* We have used here a bar graph which helps us to understand the correlation of target variable with other independent variables in an easy manner.
* We can see that credit history is highly and positively correlated with the loan status, so we can understand that if the credit history of an applicant is good there are high chances of loan getting approved.
* Least correlated columns are Self-employment, applicant income.
* Education column is negatively and highly correlated with the loan status.



**Separating Label(Y) and Features(X):**

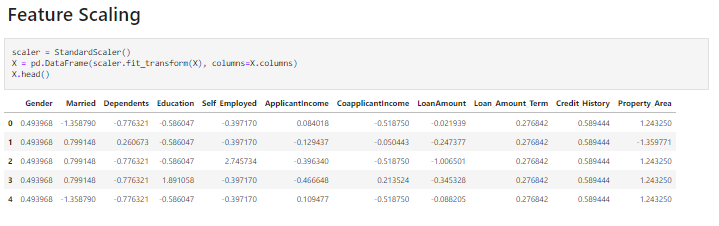


* Firstly I have separated target variable as label ‘y’ and other columns as features ‘x’.

**Checking Class Imbalance issue and resolving it:**

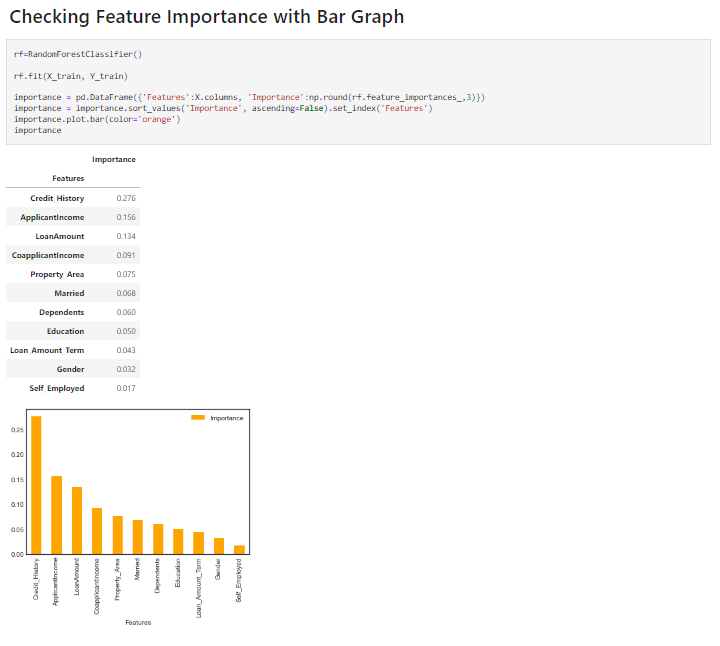
* I also checked the class imbalance for the target variable, to understand if in future I need to use any over-sampling or under-sampling techniques.
* From the above observations, we came to know that the data in target variable is imbalanced and we will have to perform the over-sampling technique to make it balance.
* For oversampling I have used SMOTE technique.
* SMOTE is nothing but synthetic minority oversampling technique which is commonly used as oversampling method to solve the imbalance problem.
* After using this method we can see that the data in Y is now balanced.

**Feature Scaling using StandardScaler:**



* Here in this case we have used StandardScaler for Feature scaling the data, other some methods are MinMaxScaler, Normalization etc.
* StandardScaler is used to resize the distribution of values so that the mean of the observed value is 0 and standard deviation is 1.

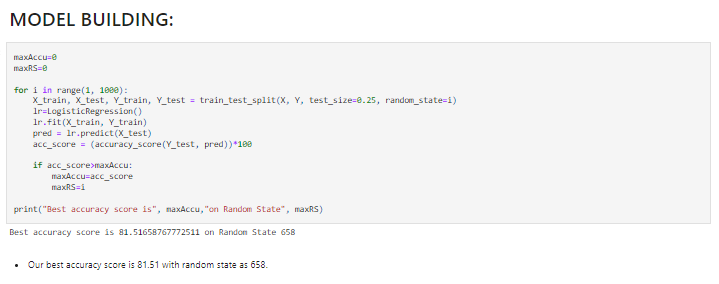
**Feature Importance:**



**Observations:**

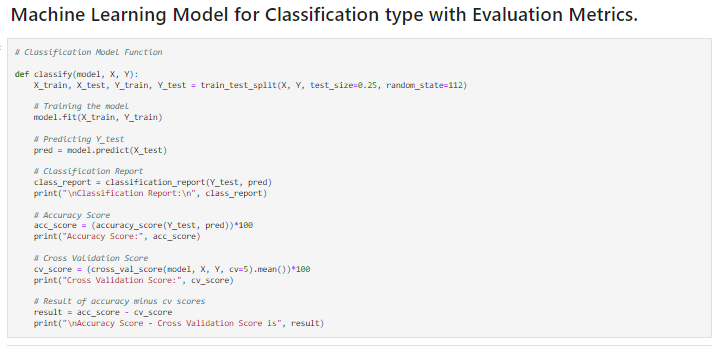
* We have plotted a bar graph which indicates the importance of the feature columns with respect to target variable.
* We can observe that the credit history column is the most important feature with respect to loan status, followed by applicant income, loan amount, coapplicant income, property area, dependents, etc.
* Least important feature with respect to loan status is Self-employed, followed by gender.

**Model Building:**



* We have used 25% data for testing and remaining 75% data is used for training purpose.
* Here we used logistic regression method to check the accuracy score and best random state.

**Machine Learning models**

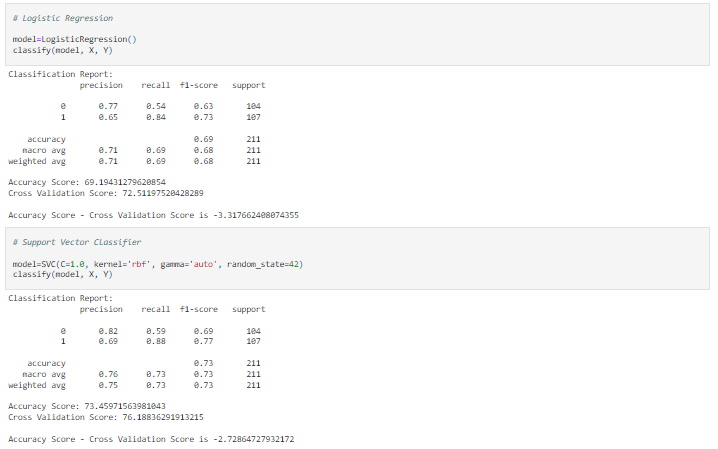


**Observations:**

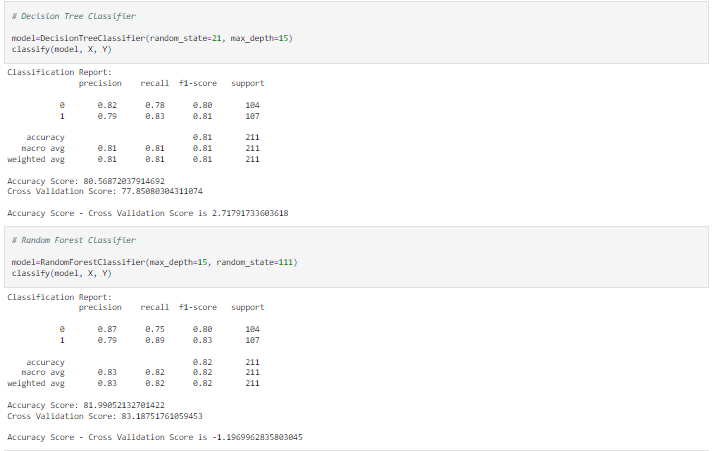
* I have defined a class that will perform the train-test split, training of machine learning model, predicting the label value, getting the accuracy score, generating the classification report, getting the cross validation score and the result of difference between the accuracy score and cross validation score for any machine learning model by using the above function.

**Classification Models**

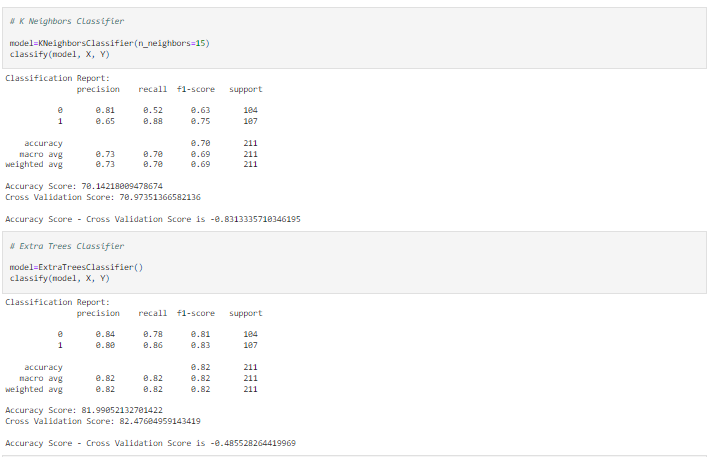
* Logistic Regression
* Support Vector Classifier



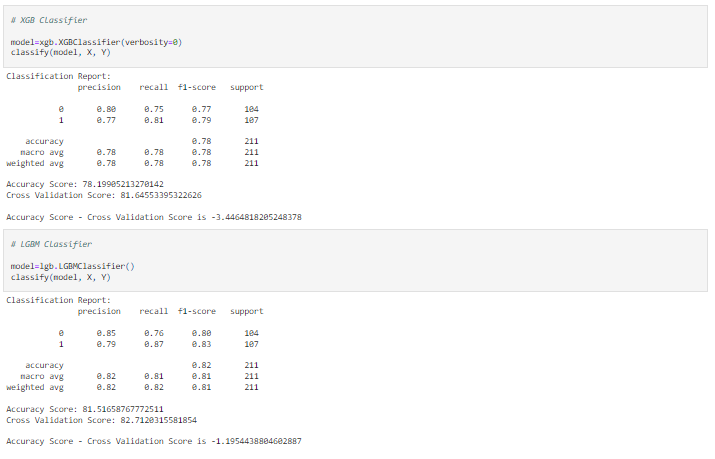
* Decision Tree Classifier
* Random Forest Classifier



* K Neighbors Classifier
* Extra Trees Classifier



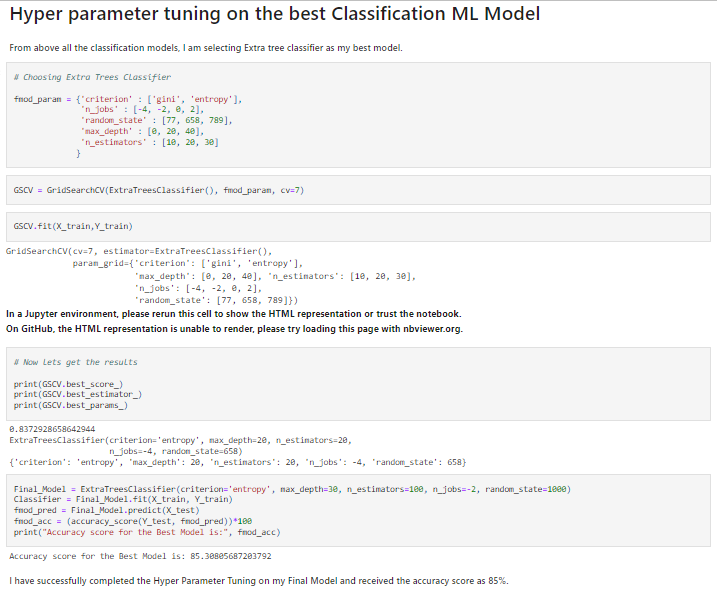
* XGB Classifier
* LGBM Classifier



* I have created all the above classification models and checked the evaluation metrics for all.
* After comparing all the classification models I selected **Extra Trees Classifier** as my best model considering the factor i.e difference between accuracy score and cross validation score with minimum value as -0.48.
* I have listed down the chosen parameters below.

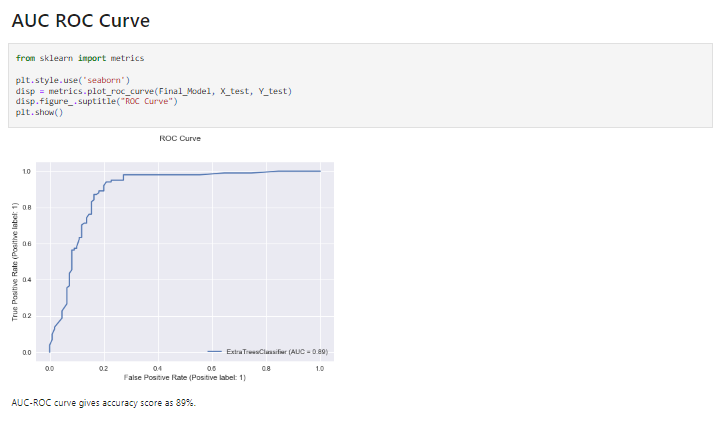
**Hyper Parameter Tuning on Best Model**

* Hyper parameter is used to control the behaviour of a machine learning model.
* If we don’t correctly tune our hyper parameters , our estimated model parameters produce suboptimal results, as they don’t minimize the loss function, this means our model makes more errors.
* Therefore we need to hyper parameter the model correctly.



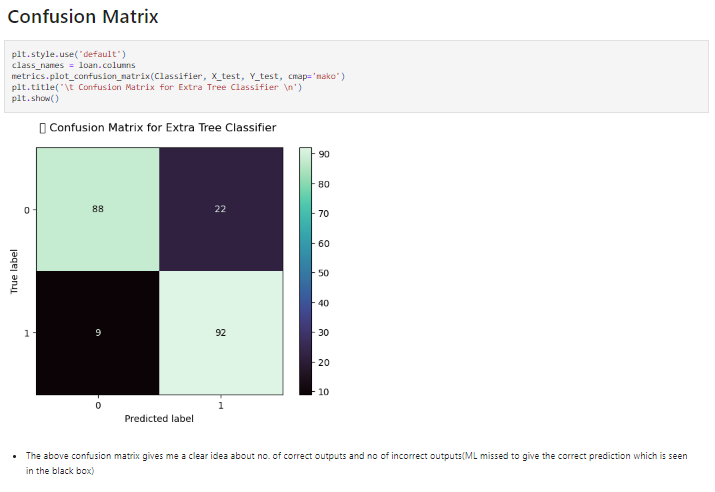
* After successfully completing the hyper parameter tuning on the best model, I received accuracy score as 85%.

**AUC-ROC curve:**



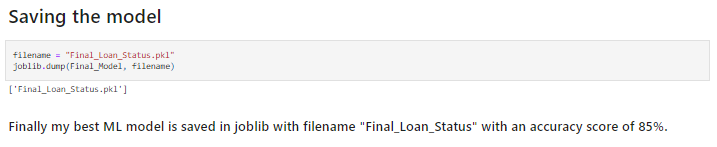
* ROC curve gives an AUC score as 89%.
* So here area under curve is quite good for this model.

**Confusion Matrix:**



* With the help of above confusion matrix I am able to understand the number of times I got the correct outputs and the number of times my ML model missed to provide the correct prediction (depicting in the black boxes).

**Saving Model:**



**Concluding Remarks:**

* At the starting point of the blog we checked the steps of building a Machine Learning Model, now you can see how we have reached the base and finally ended up to the model building and made the model ready for deployment.
* This banking area needs a good vision of data, and in every model building, problem Data Analysis and Feature Engineering is the most crucial part.
* You can see how we have handled missing data, outliers, skewness, Encoding , numerical and categorical data and also how we build different machine learning models on the same dataset.
* Using hyper parameter tuning we can improve our model accuracy, in this case we can see that our accuracy increased from 81% to 85%. Using this machine Learning Model we can easily predict whether the loan application will be approved or not.

**References:**

1. analyticsjobs.in
2. google.com