**LOAN APPLICATION STATUS PREDICTION**

**BY YASHNA SHAH**

**STUDENT**

**DATATRAINED INSTITUTE**

****

**INTRODUCTION**

Loan Origination is a process by which a borrower applies for a new loan and a lender processes that application based on financial history of the applicant. The customer has to go through many manual documentation processes, which are time consuming. Getting the loan approved has become a difficult task nowadays. Loan distribution is considered as the core part of every banks business. The main asset of the bank is loans , because banks earn a lot from loans. Risk being the major part of loan, every bank wants their money to go in safe hands. There is no guarantee whether the customer will repay the loan amount or not.

Machine learning helps in predicting the model by learning each and every aspect of data and give accurate decisions whether the loan can be given to the customer or not. This model will save time and efforts of bank employees.

**PROBLEM DEFINITION**

The purpose of this blog is to provide fast and effective ways to choose deserving applicants for loan based on their repaying capability.

We have a dataset that includes details of applicants who have applied for loans, the details in the dataset include credit history, loan amount, applicant’s income, etc. On the basis of the details given in the dataset, we need to build a model that predicts whether the loan of the applicant will be approved or not.

Independent Variables**:**

* Loan\_ID
* Gender
* Married
* Dependents
* Education
* Self\_ employed
* Applicant Income
* Co-applicant Income
* Loan\_Amount
* Loan\_Amount\_Term
* Credit History
* Property\_Area

Dependent Variable:(Target Variable)

* Loan\_Status

Our problem statement clearly depicts that our target variable is categorical, which means that it is a Classification Problem. Hence, we would be using classification machine learning algorithms to build a model that predicts the loan application status of the applicant.

**DATA ANALYSIS**

Data Analysis is all about gathering raw data and converting the same into informative data which helps in making decisions for the profitability of the business. First the data is collected and then analyzed.

We have one dataset. Here, will first train the model and when the test data will be provided to the model, it will predict the loan status on whether to approve the loan or not.

* Our dataset has 614 rows and 13 columns.
* Most of the features are categorical and few are numerical.
* Will check the statistical summary of the data with the help of describe () method.

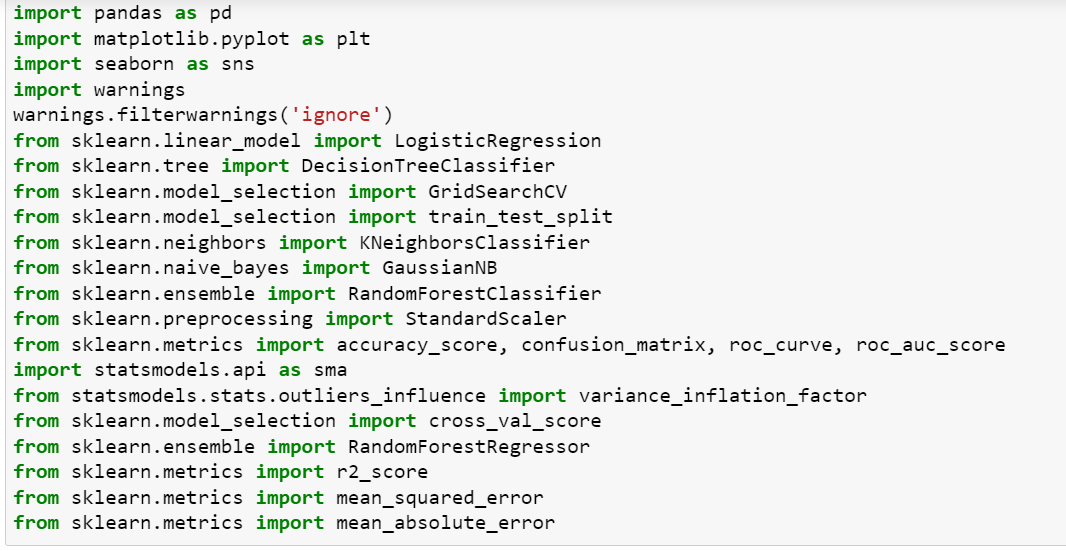
Observations:

1. Since our data seems to be scattered, we observe, for few features Standard Deviation is far from mean.
2. The maximum amount of loan applied is 700 with the minimum loan amount term being 480.
3. The minimum amount of loan applied is 9 having the minimum term of 12.
4. Few features are having null values, which need to fill as per the data types of the features.
5. The next step would be to check the unique values of each columns and drop the columns with the unique value and does not help us in predicting the target variable.

**EXPLORATORY DATA ANALYSIS**

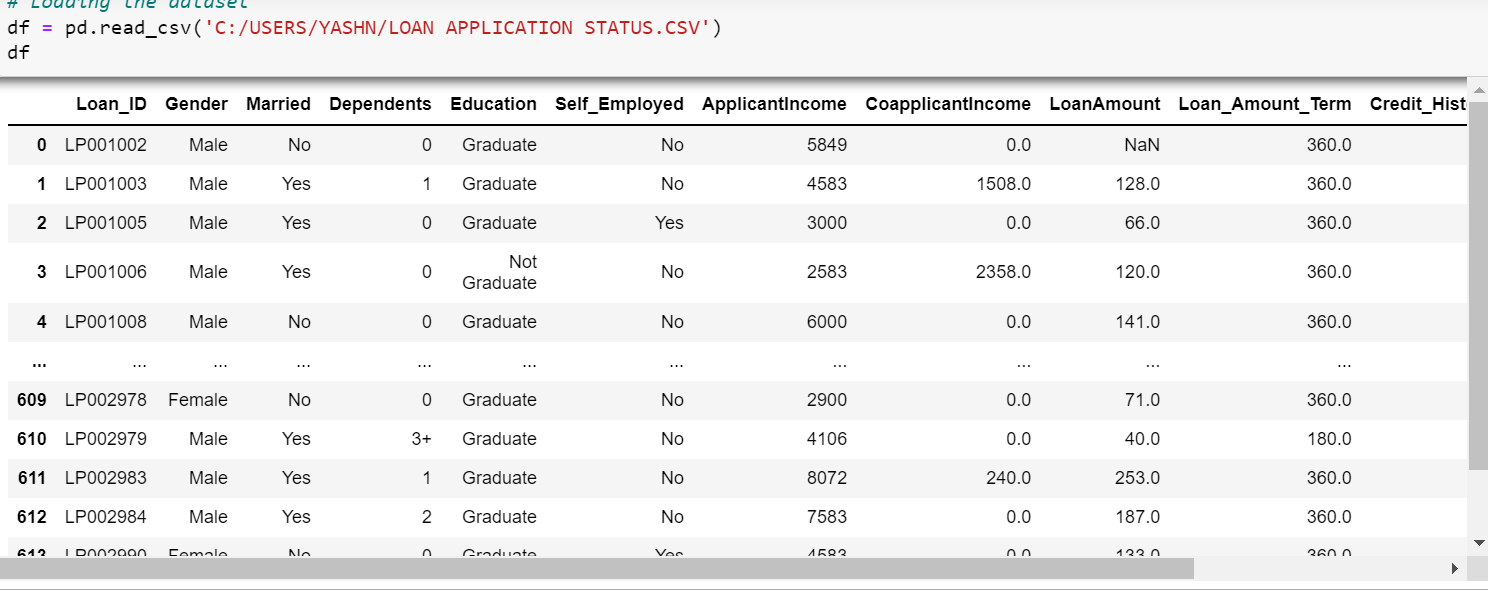
Importing the required libraries:

The first step would be to import the necessary packages like Pandas, Numpy, Seaborn etc, to carry the necessary operations further.



Loading the dataset:

Next, will load the dataset using Pandas. For this we used read\_csv.



Checking the Dimensions:

We will check the shape of our data by using df.shape which gives us the total number of rows and columns in our dataset.



Checking Null Values in our dataset:

Before proceeding further, we need to check null values in our dataset using df.isnull().sum()

Loan\_ID 0

Gender 13

Married 3

Dependents 15

Education 0

Self\_Employed 32

ApplicantIncome 0

CoapplicantIncome 0

LoanAmount 22

Loan\_Amount\_Term 14

Credit\_History 50

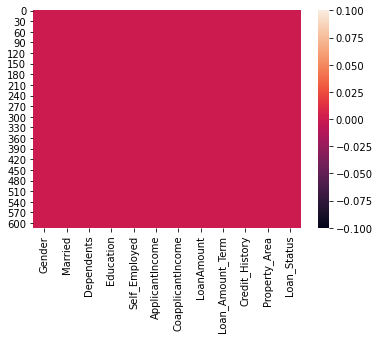
Property\_Area 0

Loan\_Status 0

dtype: int64

From the above output, we can observe, that we have many null values in our dataset which we need to fill before proceeding further. Hence, null values have been filled using fillna().

Plotting the heatmap



We have no null values in our dataset. All the null values have been filled.

Visualizing the data distribution:

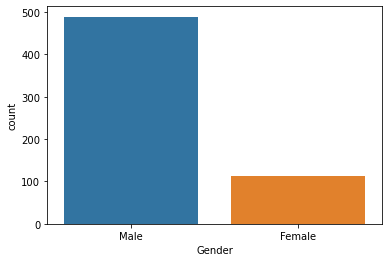
Checking the unique counts of features.

1. Gender:

Male 489

Female 112

Name: Gender, dtype: int64



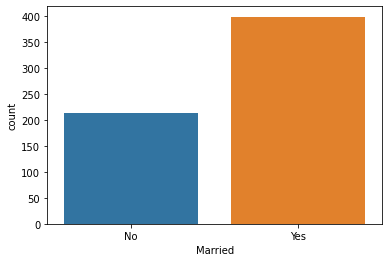
Here, we observe, the total counts for male is more than female. Which means, more males have applied for loans as compared to females.

1. Married:

Yes 398

No 213

Name: Married, dtype: int64



We observe, total count for married is higher than unmarried. Which means more of married people have applied for loan as compared to unmarried people.

1. Dependents:

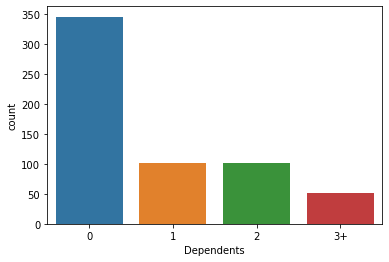
0 345

1 102

2 101

3+ 51

Name: Dependents, dtype: int64



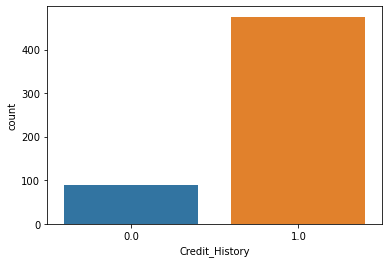
0 dependents have the highest counts whereas, 3+ has the lowest counts. Hence, we can say that, people with no dependents have applied for the loan most.

1. Credit History:

1.0 475

0.0 89

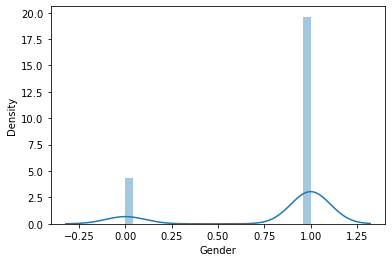
Name: Credit\_History, dtype: int64



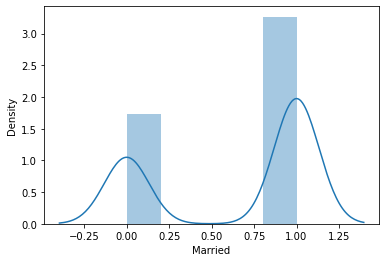
People having 1 credit history has more chances of loan getting approved compared to other.

Graphical Representation of all the features:

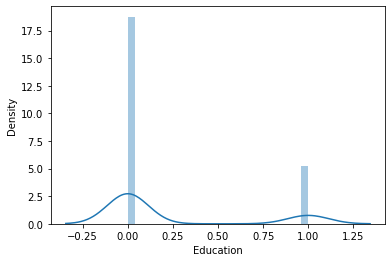
Gender



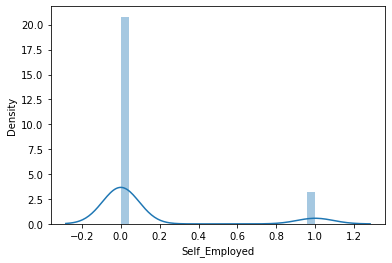
Married



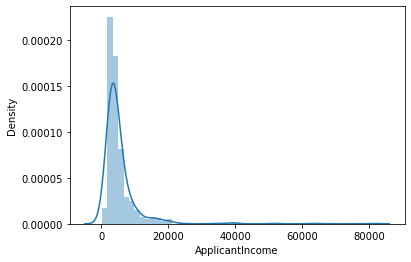
Education



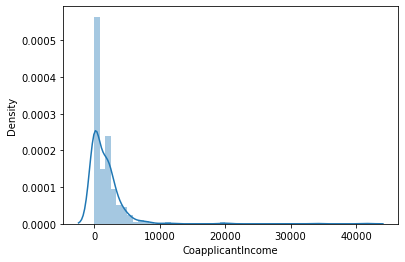
Self Employed



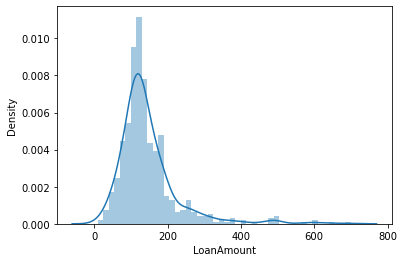
ApplicantIncome



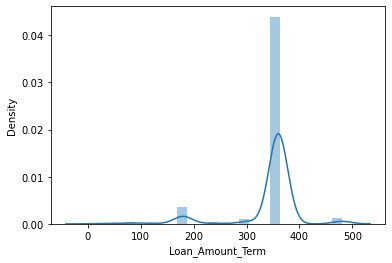
CoapplicantIncome



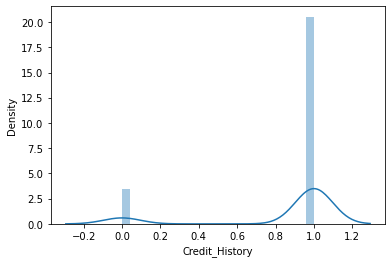
LoanAmount



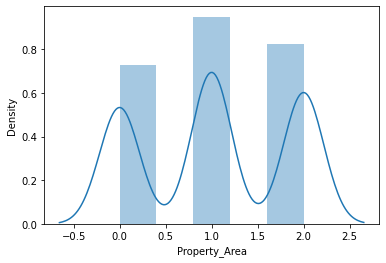
Loan Amount Term

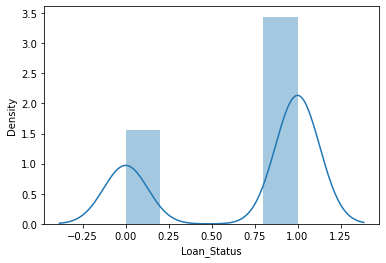


Credit\_History



Property\_Area

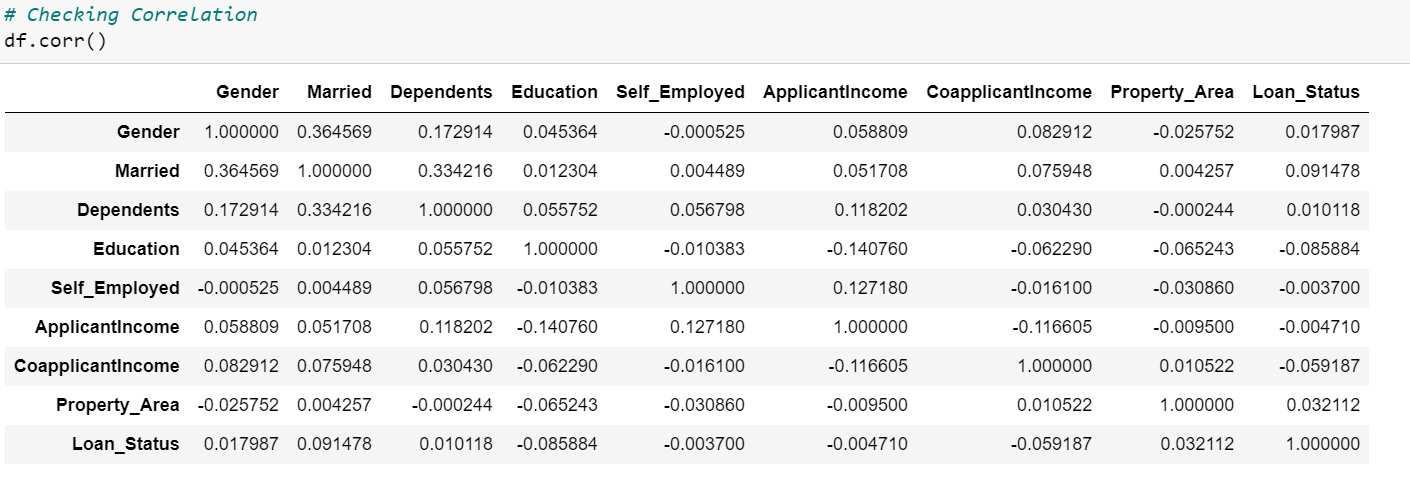


Loan\_Status 

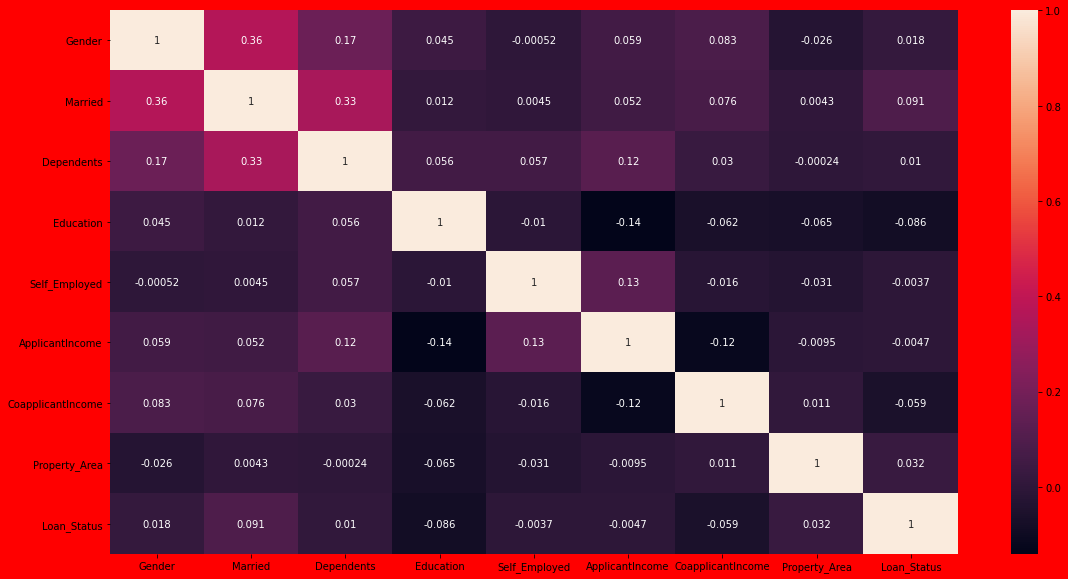
From the above graphs, we observe that most of the plots are bimodal. Skewness is also present in some columns.

Checking Correlation between Variables:

We will check the correlation between the variables using df.corr()



Visualizing correlation with a heatmap which helps us to check multicollinearity between the variables.

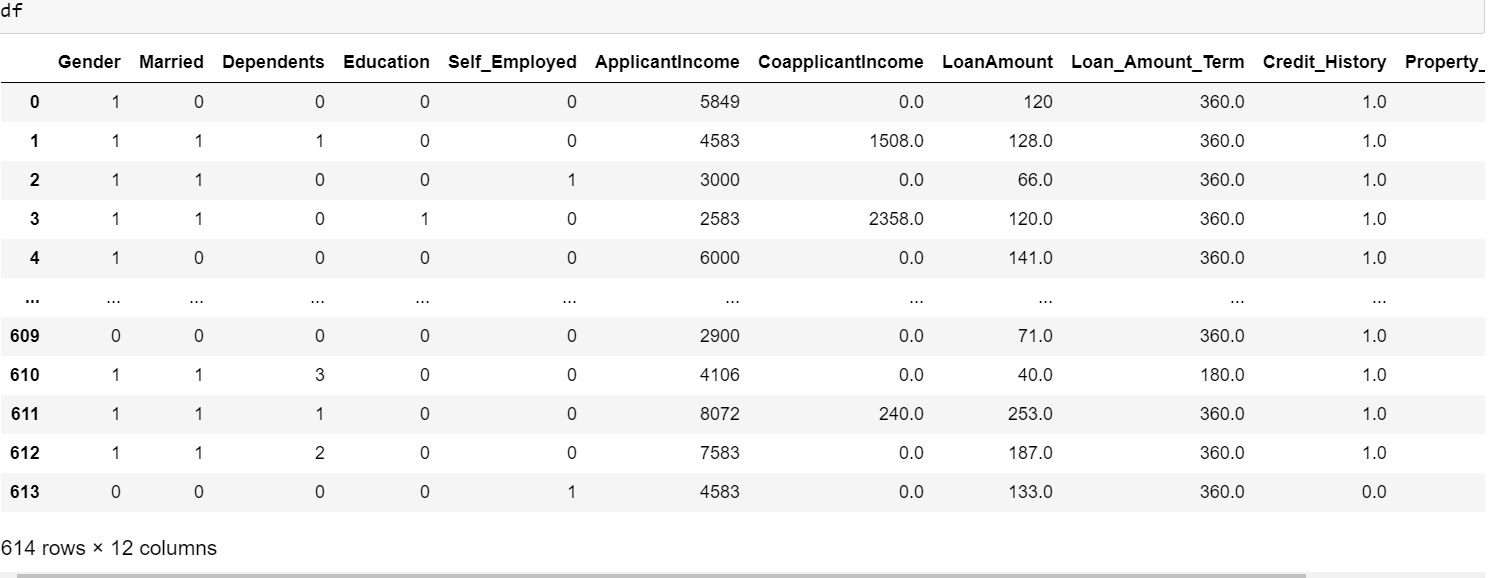


Observations:

1. Credit history is highly correlated with target variable, whereas, education is least correlated with Loan\_Status.
2. There is a multicollinearity between applicant income, co-applicant income and loan amount

Dealing with Categorical Features:

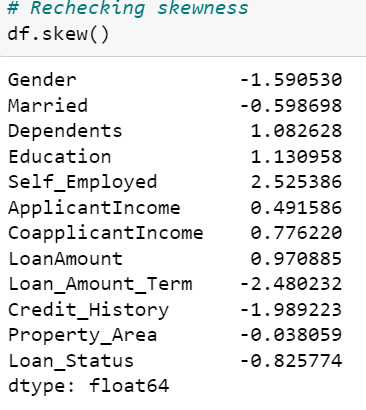
As we have many categorical columns in our dataset, we need to convert them into numerical data for modeling, for which we will be using LabelEncoder for converting each features.



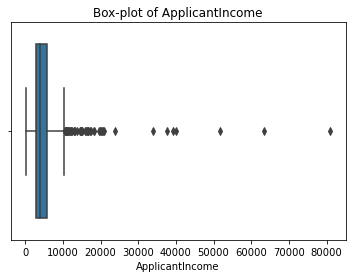
Checking for Outliers/ Skewness:

There are outliers in few features, which we need to remove before building a model.

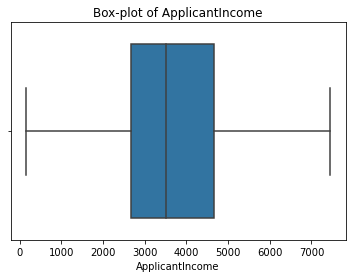
Will check for the skewness in the data by using df.skew().

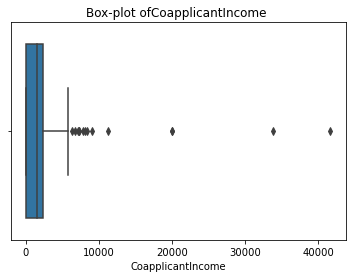


We need not handle the skewnes if its between -0.5 to 0.5.

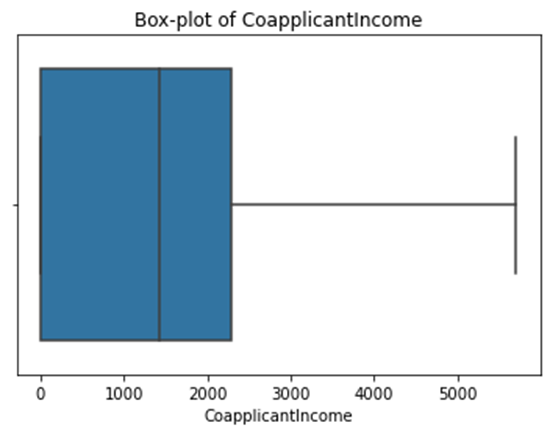


After Removing Outliers using df=df[df['CoapplicantIncome']<7500] method Here we can see, there are outliers from 7500 .





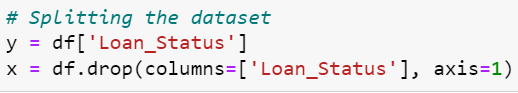
After Removing Outliers using df=df[df['CoapplicantIncome']<6000] method. Here, we can see, there are outliers from 6000.



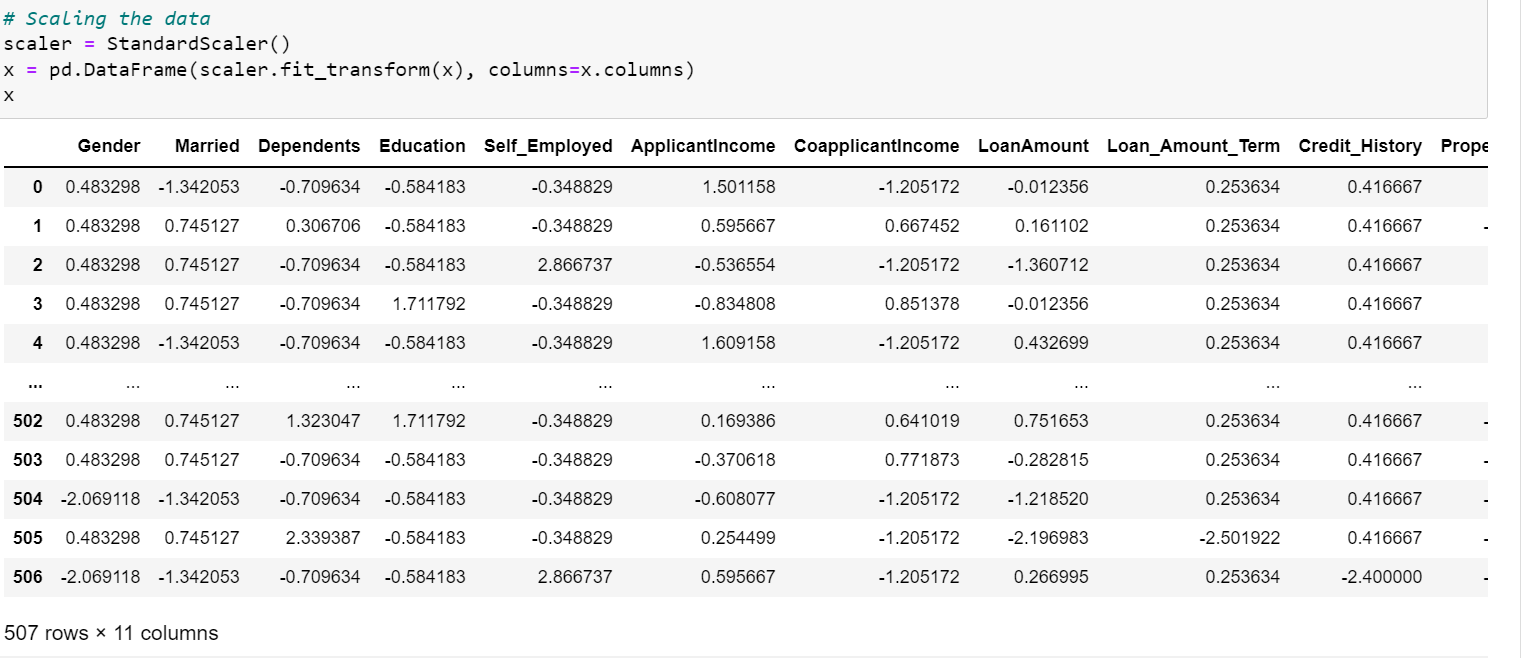
**PREPROCESSING PIPELINE**

Splitting the dataset into test and train:

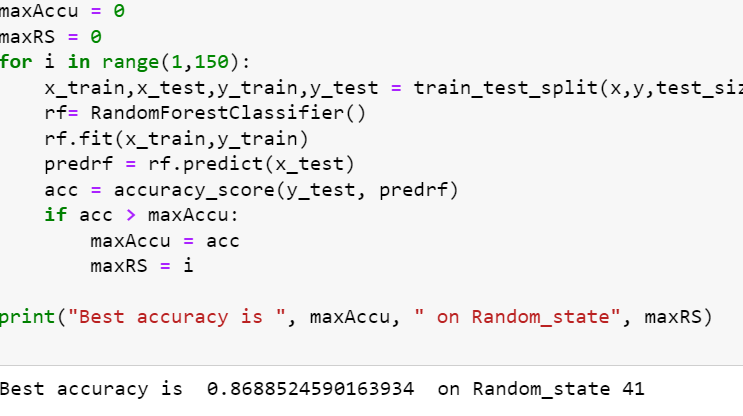
1. The next step is to split the data into test and train and drop the loan status column from the data set as we need to predict the loan approval. Splitting the train-test data , 70% of the data will be trained and 30% will be tested.



1. As the data is not scaled, we need to scale the data with the Standard Scalar Method. Scaling is required because there is too much difference between minimum and maximum values of the features.



1. To obtain the Best Random State, we used Random Forest Classifier.



1. The new dimension of the dataset is 507 rows and 11 columns.
2. Will split the train and test data in 0.24 test size and best random state.

**BUILDING MACHINE LEARNING MODELS**

Model Creation:

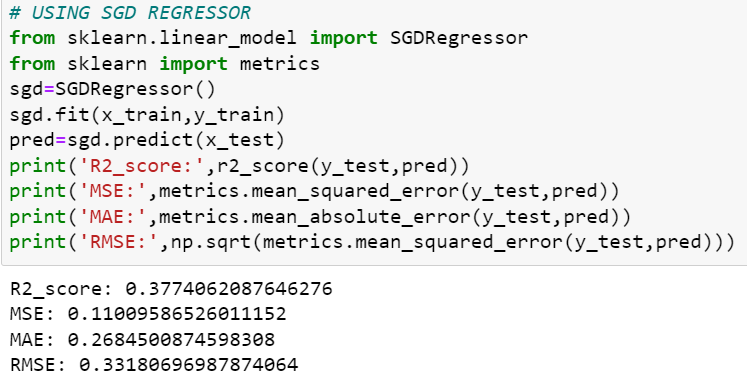
After splitting the data into train and test using train- test – split method, we have x-train, x-test, y-train and y-test which is required for building the machine learning models. Will be building multiple classification models, so that we get the best accuracy score, classification report and confusion matrix for all the models. The algorithms used for model building and predictions are listed below:

* Logistic Regression
* Random Forest Classifier
* Decision Tree

These algorithms have been used for both training and testing the model and the model have been evaluated on the basis of classification metrics like confusion matrix, precision and recall.

Regularization methods like LASSO and RIDGE are used to mitigate the problems of over-fitting and under- fitting.

SGD Regressor

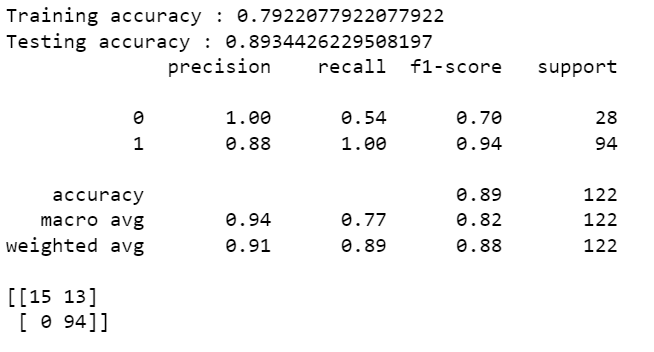


LASSO

****

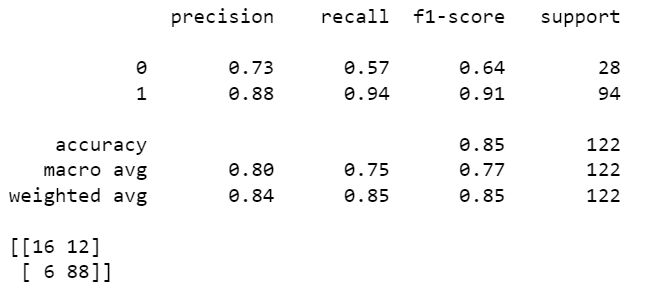


Logistic Regression: it is used to describe the data and explain the relationship between dependent and independent variable. It is a classification model and is used when the target variable is categorical in nature.

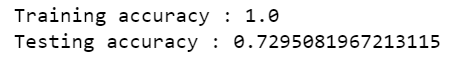


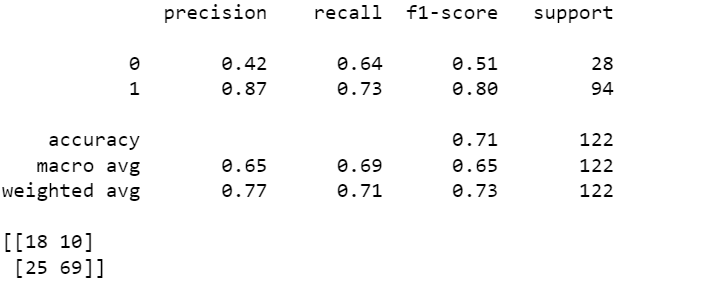
Random Forest Classifier: it is a classification algorithm which consists of many decision trees. It can perform both regression and classification tasks. Random Forest produces good predictions that can be understood easily. It searches for the best features among a random subset of features.





Decision Tree: the main advantage of decision tree classifier is its ability to use different feature subsets and decision rules at different stages of classification. It often involves higher time to train the model. The tree can be explained by two entities, namely decision nodes and leaves.





Observations:

For every above model, we have trained the data i.e; x\_train , y\_train and we did the prediction using x\_test. We got the accuracy score with thye help of y\_test and the prediction value. Out of all the models , we are getting the best score with Random Forest Classifier.

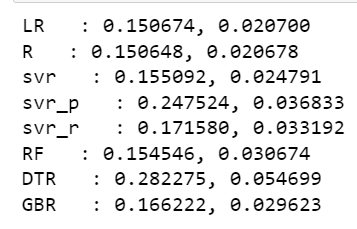
Random Forest Classifier has the highest accuracy score of 85% and the difference between Cross Validation Score and Accuracy score is also less. So will use Random Forest Classifier to learn the model.

Reasons of using Random Forest Classifier:

1. Random Forest reduces over-fitting in decision tree and helps to improve accuracy.
2. It is a rule based approach.
3. It automates missing values present in the data.
4. It works well with both categorical and continuous values.

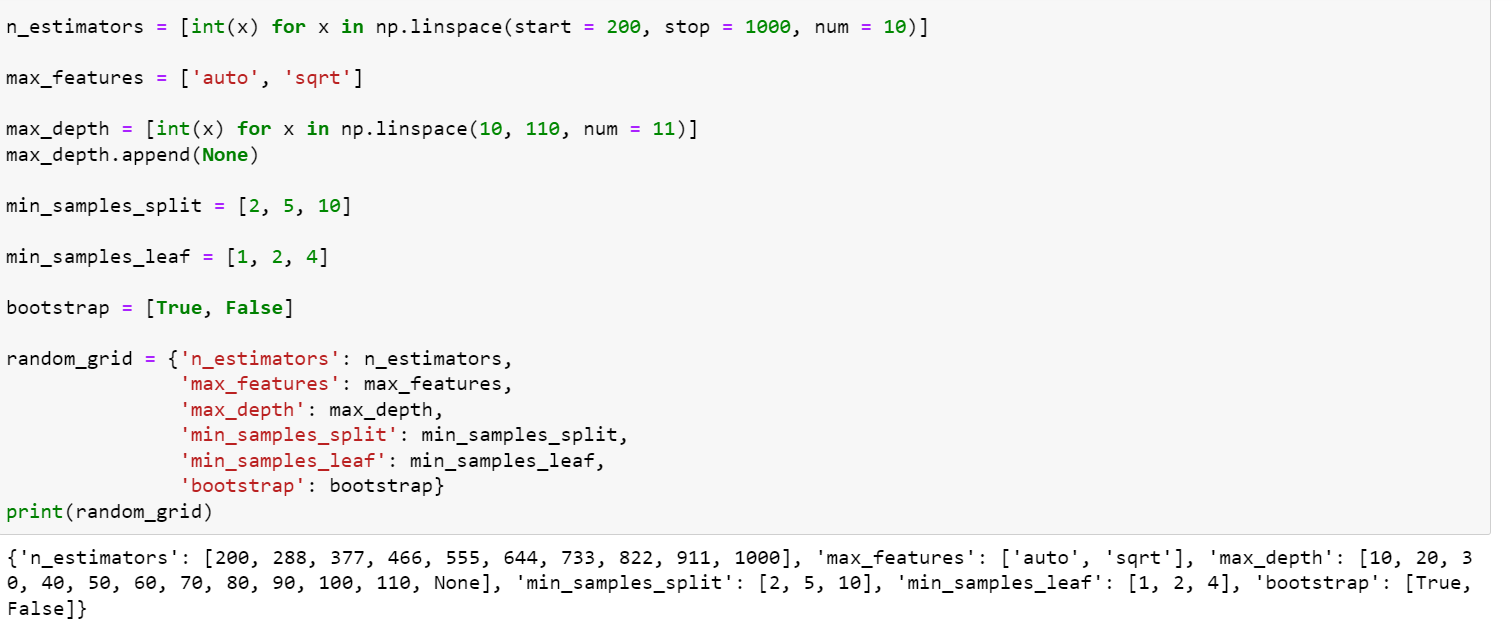
Cross Validation:

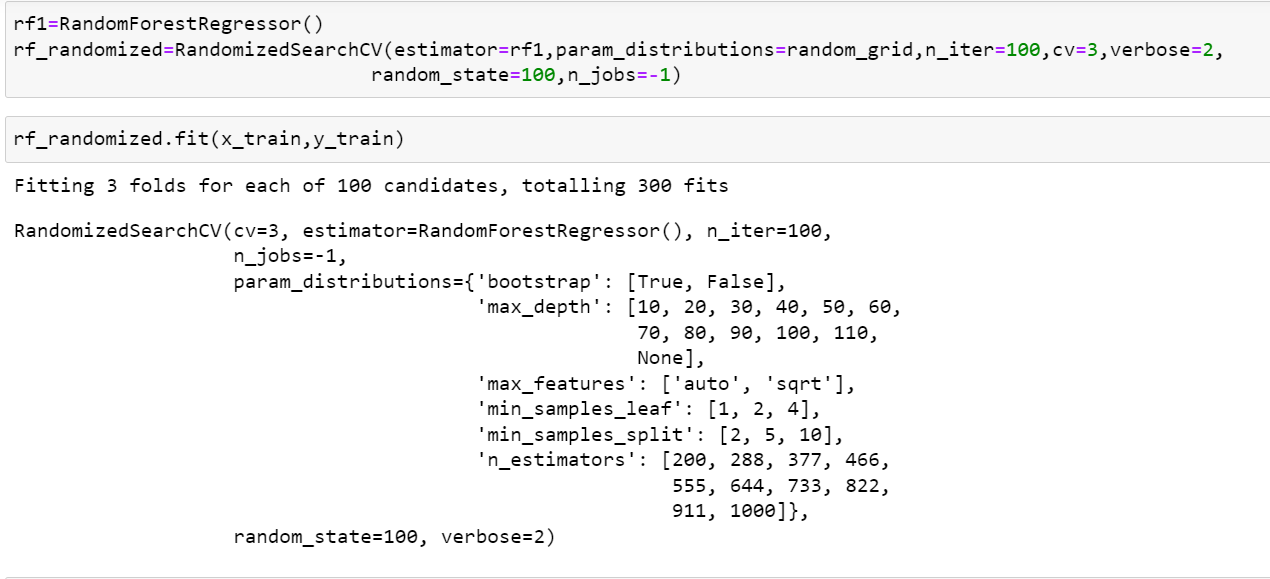
Cross Validation will be performed for every model using sklearn.model\_selection import cross\_val\_score. Whichever model gives the best score with less difference between cross validation score and the accuracy score, is considered the best fit model.

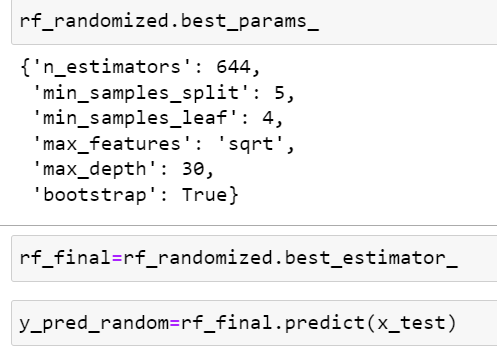


Hyper parameter Tuning:

We will perform hyper parameter tuning to get good and more accurate result from the model. For that, we will try to increase the accuracy of the model by giving the best parameters and again try to increase the accuracy score.



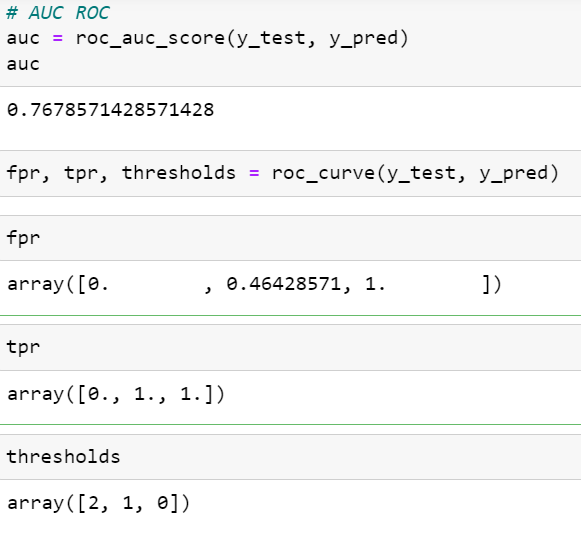




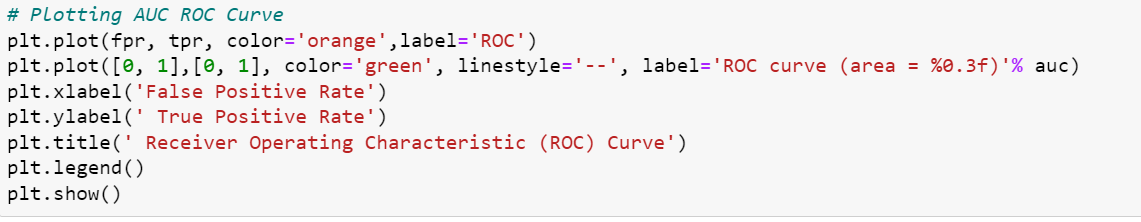
Once, we have built the machine learning model, our next step would be to evaluate and validate how good or bad it is and then deciding whether to implement it or not. AUC ROC curve helps in visualizing how well our machine learning classifier is performing.

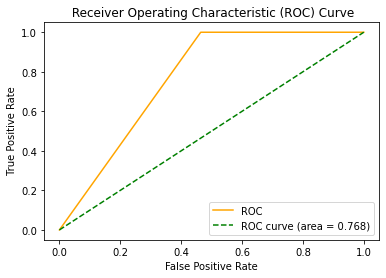
AUC-ROC Curve:

AUC-ROC Curve is a performance measurement for classification problem at various thresholds settings. ROC is considered as a probability curve whereas AUC represents degree or measures of seperability. It tells us how much model is capable of distinguishing between classes.



Plotting AUC ROC Curve to see the false positive rate and true positive rate.

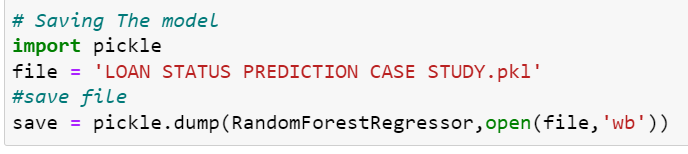




**CONCLUDING REMARKS**

Saving the model:

The model is ready and we will save the model in ‘pkl’ format for future reference.



Predict the model:

Once, we are done with saving the model for future reference, we will predict the model with the actual trained data set.



Here, we observe that the predicted model shows the similar relationship with the actual loan status from the train data set, which means the model is predicted correctly.

In this case study, a Machine Learning model is developed to predict the loan application status. With the help of above techniques, proposed model is able to predict the loan application status.

This will help the banks in predicting which customer would be eligible and get the loan approved based on these features. This will even help banks in saving their time. This in turn will also help applicants to predict whether they can apply for the loan or not based on the banks eligibility.

In this way, Machine Learning techniques can solve this issue in predicting and even reduce manual efforts.