**PREDICTING THE LOAN APPLICATION STATUS**



**Problem Definition:**

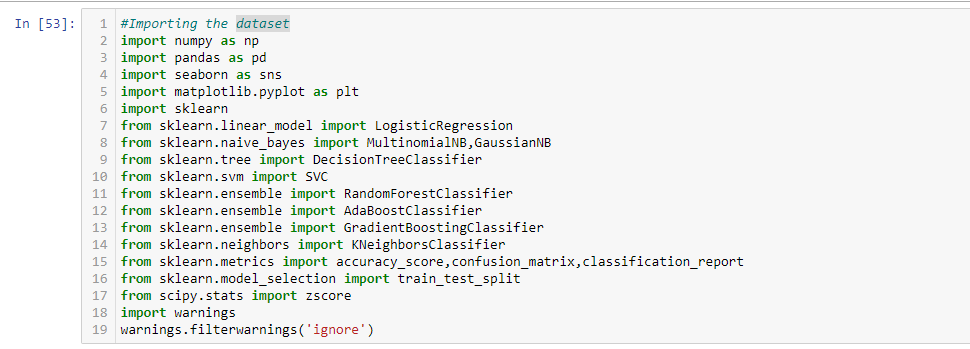
Loan lending has been an important part of daily lives for organizations and individuals alike. With the ever-increasing competition in the financial world and due to a significant amount of financial constraints, the activity of taking a loan has become inevitable. Individuals around the world depend on the activity of loan lending for reasons such as overcoming their financial constraints for them to achieve some personal goals. Similarly, banks, small to large firms depend on the activity of loan lending for the basic purpose of managing their affairs and to function smoothly in times where there are financial constraints.

Although it is quite profitable and beneficial for both the lenders and the borrowers. However, it carries a great risk, which in the domain of loan lending is referred to as Loan risk. Industry experts and researchers around the world assign individuals with numerical scores known as credit scores to measure the risk and their creditworthiness.

Throughout the years, machine learning algorithms have been used to calculate and predict credit risk by evaluating an individual’s historical data.

In this project, the dataset includes details of applicants who have applied for loan. The dataset includes details like credit history, loan amount, their income, dependents etc.  We must build a model that can predict whether the loan of the applicant will be approved or not based on the details provided in the dataset.

**Importing the libraries:**



*Exhibit 1*

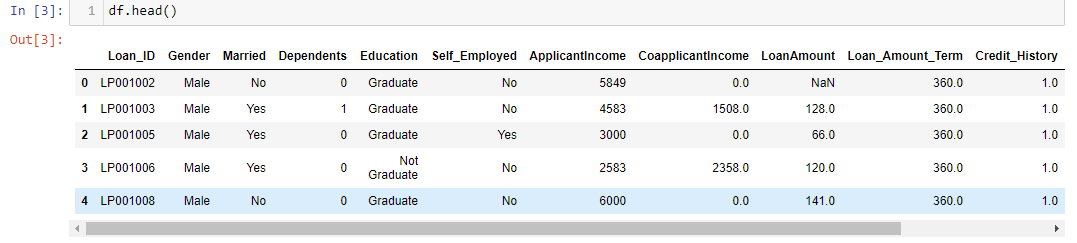
**Getting the data:**



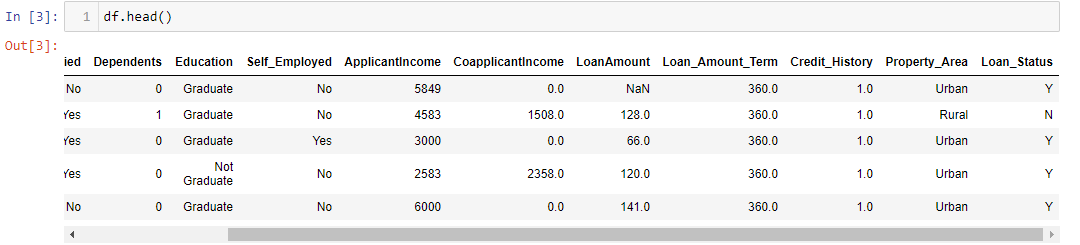
*Exhibit 2*

**Data Analysis:**

* Checking the first 5 rows of the dataset

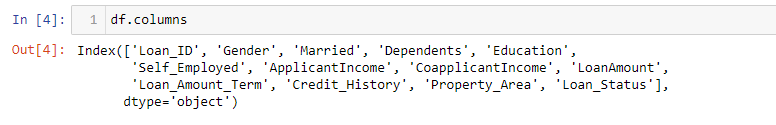


*Exhibit 3*



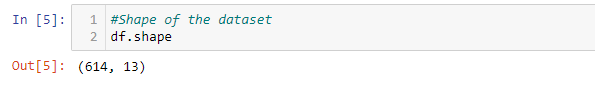
*Exhibit 4*

* Checking the columns of the dataset



*Exhibit 5*

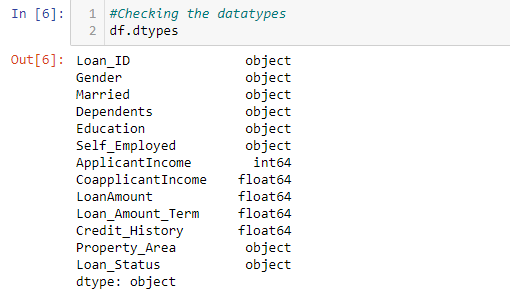
* Checking the shape of the dataset



*Exhibit 6*

The rows in the dataset is 614 and the columns are 13.There are 13 features including the target variable.

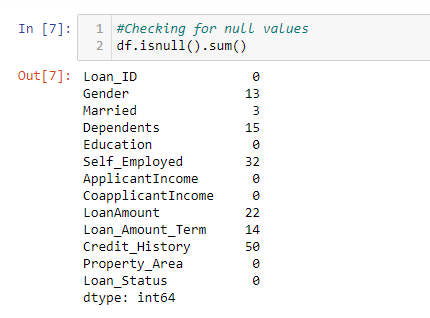
* Checking the datatypes



*Exhibit 7*

There are 8 object type features which we must encode in our future steps so that machine can understand.

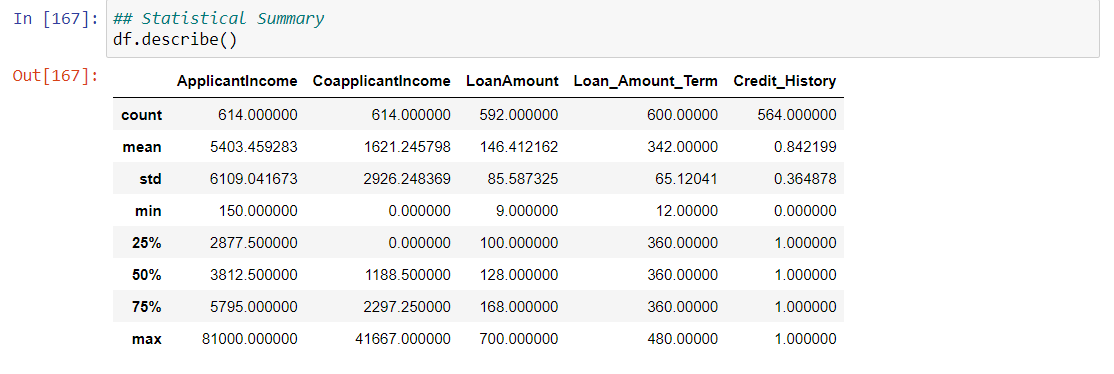
* Checking for null values in the dataset



*Exhibit 8*

There are null values present in our dataset which we must work on in our future steps to make our data clean and ready for machine building process.

* Checking the Statistical Summary



Here we can see that our data is right skewed and left skewed both.

Applicant Income, Co Applicant Income are right skewed because Mean is greater than Median values.

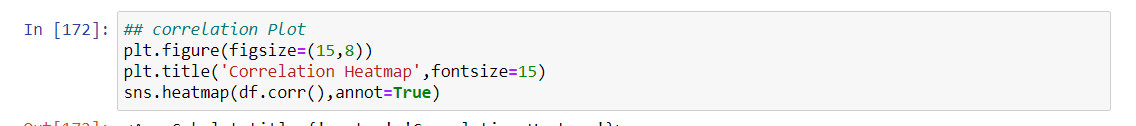
Loan amount, Loan\_Amount\_Term, are left skewed because Median is greater than Mean values.

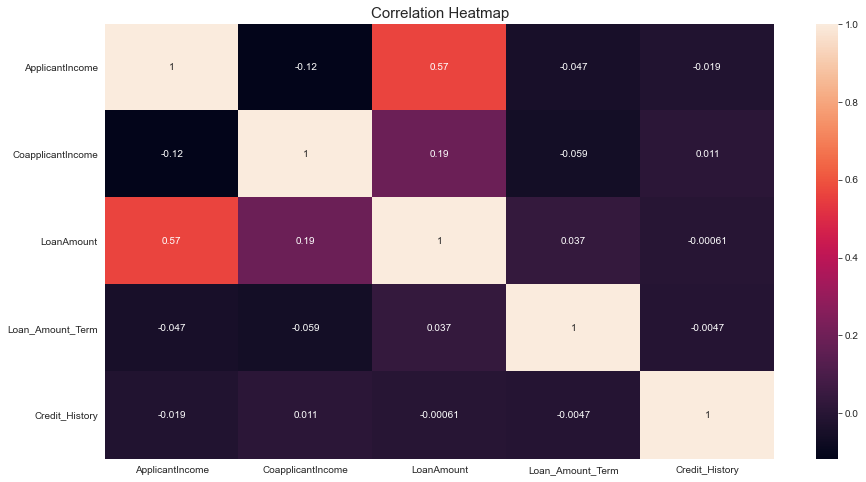
We can see that there are outliers present in our dataset because difference between the value of 3rd quantile and maximum values is more.

Now, we will start cleaning our data in our next steps so that we can build an efficient machine learning model.

**Data Visualization:**

1. We will create the Correlation Matrix to get some insight on the correlation between the attributes. We will use the below code for it.

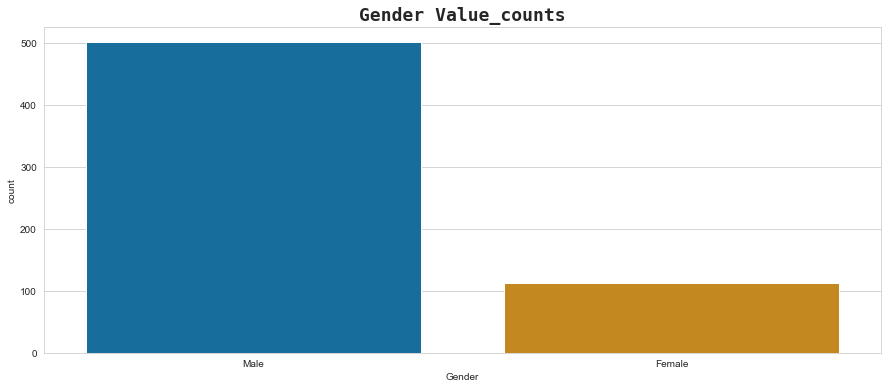


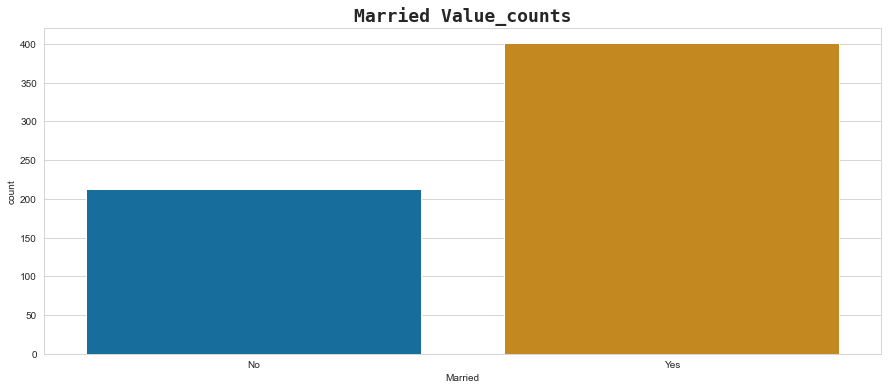


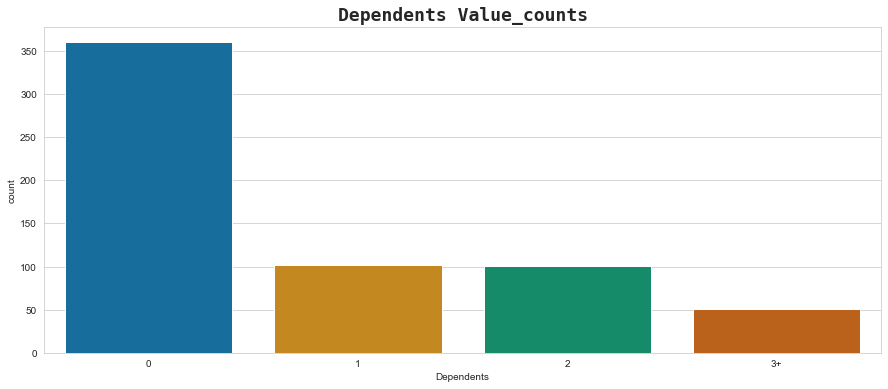
From the above Correlation matrix we can say that the Loan amount is highly correlated with Applicant Income which is 0.57.

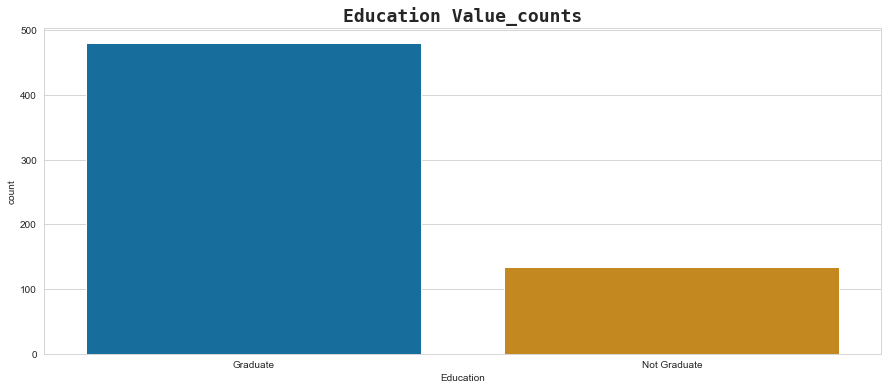
There is neither strong positive and strong negative correlation present in any variable.

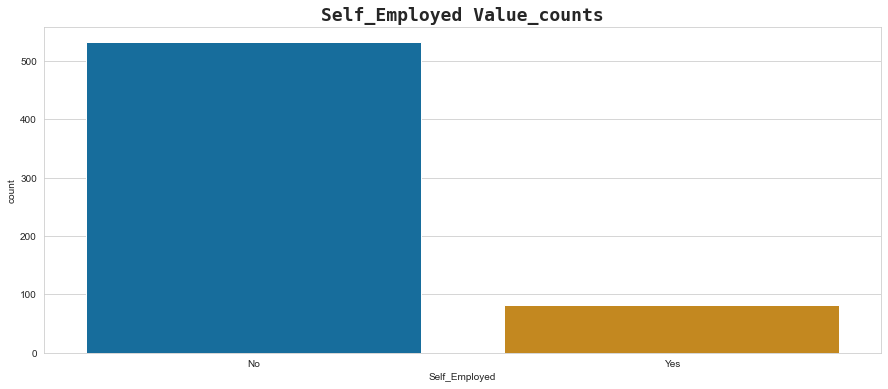
1. Now we will plot a graph of the Value Counts of different columns to get some insights.

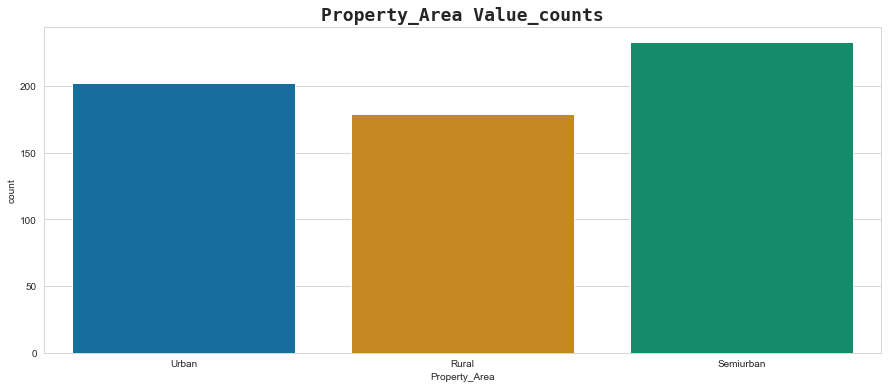


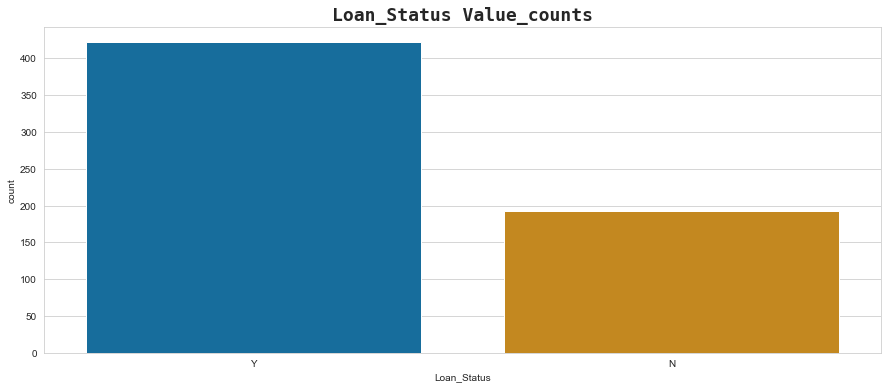












From the above graphs we can conclude that there are around 82% applicants who are male in our dataset.

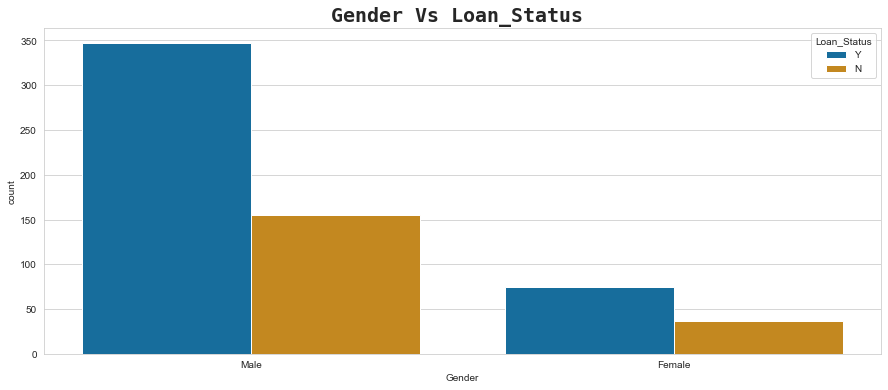
Approximately 47% applicants are married. Majority applicants have zero dependents in our dataset.

The not graduated applicants are below the half of the count of the applicants who

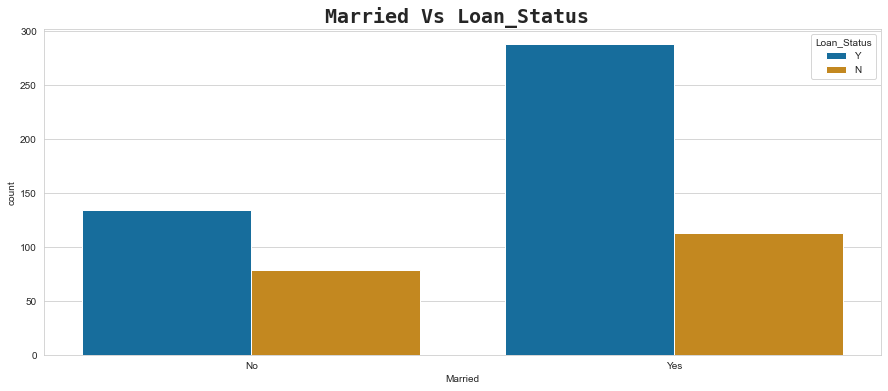
are graduated.

Around 16% applicants are self employed. Most of the loan applicant belong to the semi urban area.

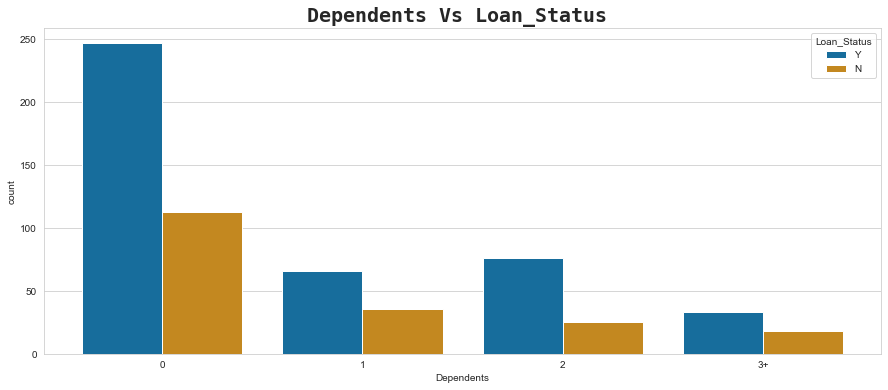
1. Now we will plot the graph for different columns against target column Loan Status and get some insights.



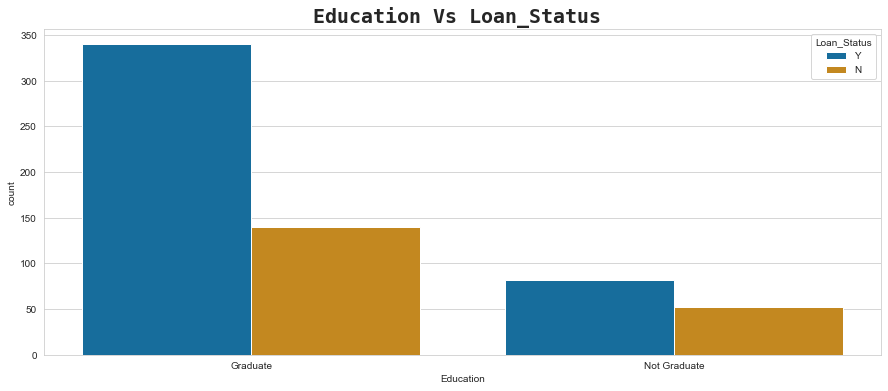
Here we can see that mostly Males sanctioned for loan as compared to Females.



Around 300 applicants are married whose loans are approved as compared to the applicants who are not married but their loans are approved.



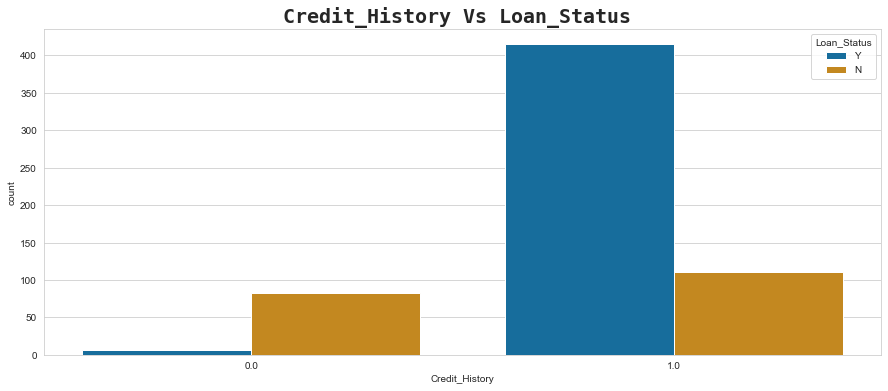
Majority of the applicants whose loans are approved have no or 0 dependency and the minimum loan approved to those who has higher number of dependents.



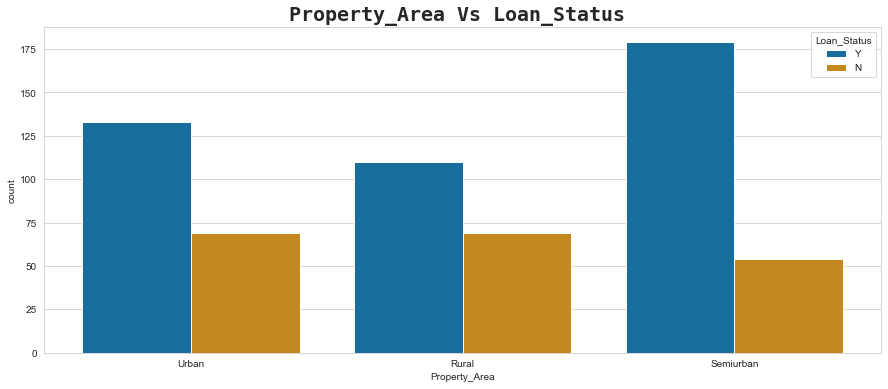
Graduates have high rate of Loan approval than non graduates.



The percentage of self-employed applicants having approved loans is around 15% of the non-self-employed applicants having approved loans.



People who have credit history 1 have the highest loan approval as compared to the 0 credit history. People who have zero credit history are mostly denied for the loan.



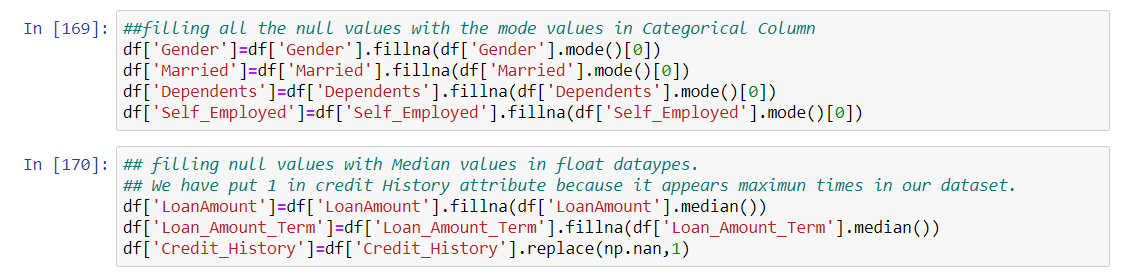
The maximum number of applicants whose loans are approved belongs to the property in semi-urban area.

**Data Pre-processing:**

1. We will first drop some unnecessary columns which does not have any significance to predict the loan status.

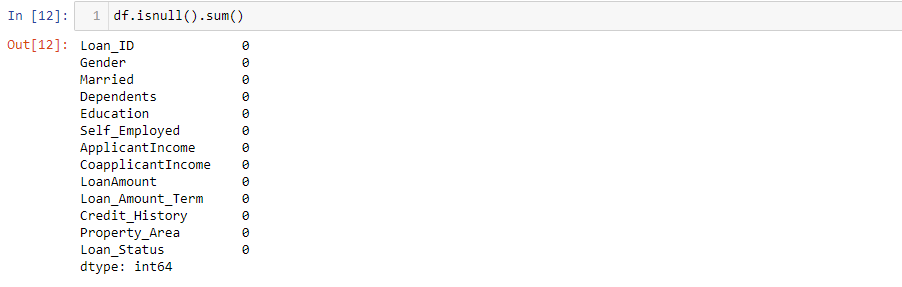


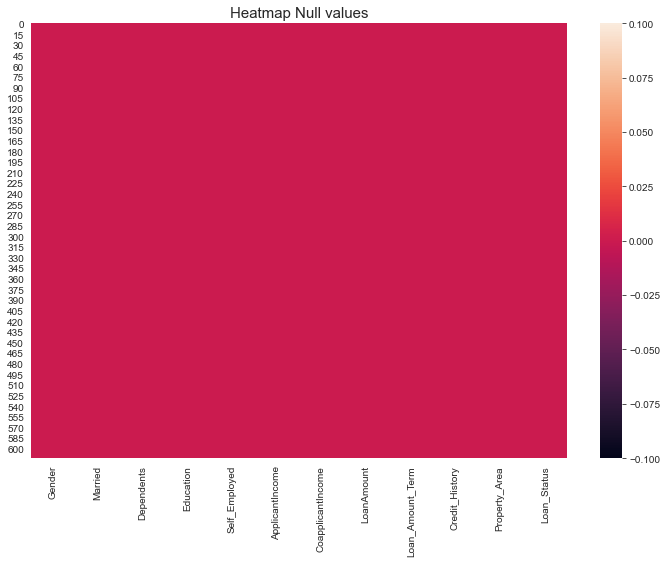
1. As shown below we will treat the null values by replacing them with the mode of the column for the categorical columns and replacing with median for the float datatype columns. We have put 1 in the credit history attribute because it appears maximum times in our data set. Below is the code for the same.



*Exhibit 9*

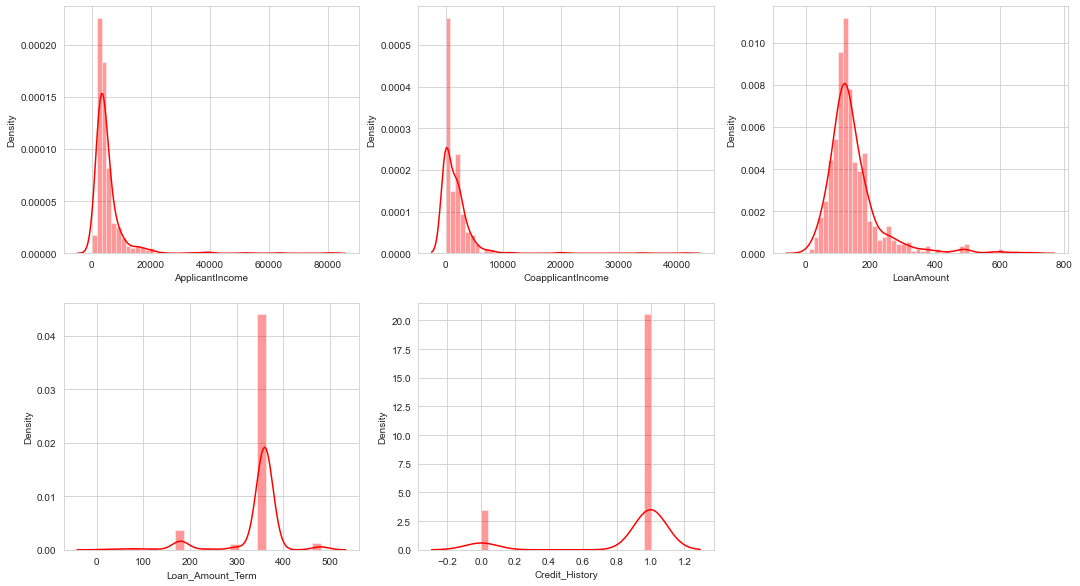
Now let’s check if any null values are left again to remove.





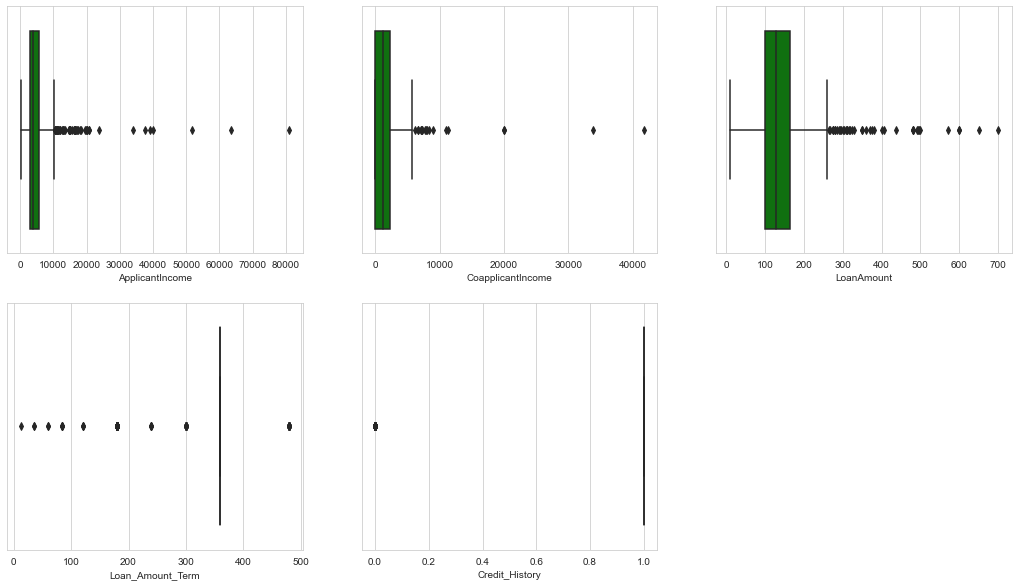
*Exhibit 10*

From the above screenshot,I can say that null values are not present in my dataset now.

1. Now we will check the Distribution of the columns. Below is the graph of all the distribution of columns. 

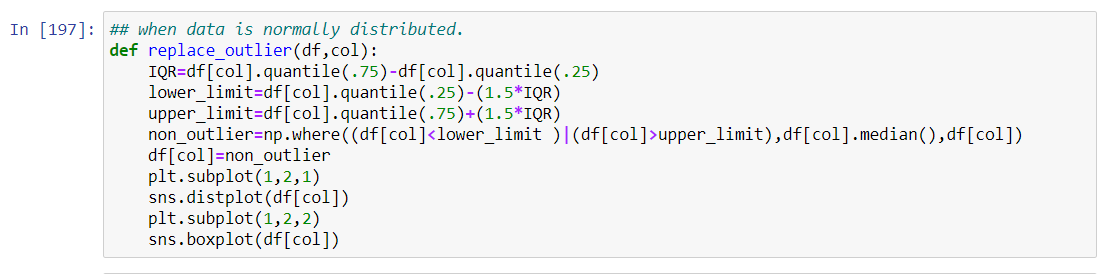
We can see that applicants income, Co- applicants income, Loan Amount are right skewed. Loan amount term has majority values of 360 months. Credit history has only two values (0 or 1), in which majority values are one.

1. Now we will check the outliers in our dataset. We will use the boxplot to identify the outliers in our dataset. Below is the graph.



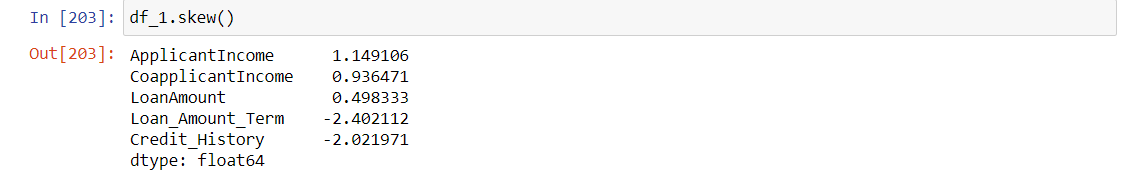
Here we can see that there are outliers present in Applicants Income, Coapplicants income and Loan Amounts.

1. We will treat the outliers by replacing it with the quantiles. We will create a function as shown below and pass the columns which needs to be treated.

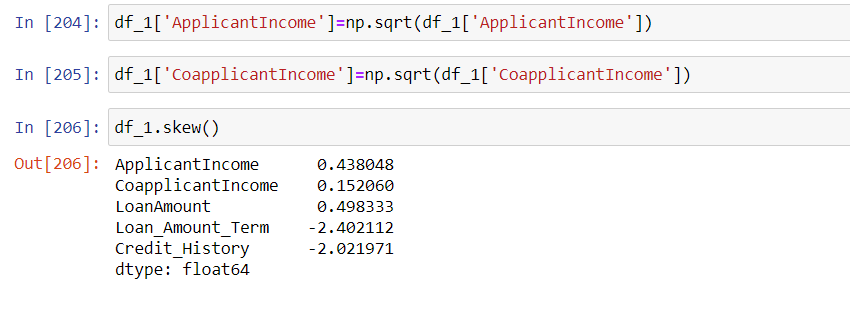


After passing all the columns to the function we have successfully replaced all the outliers from the columns.

1. I will have a look on my skewed data now. I must minimise the skewness in my data if any before machine building.



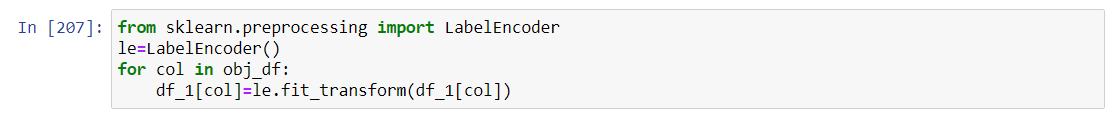
We can see that Applicant Income and Coapplicant Income are right skewed, hence we will treat them by square root transformation method. We will not treat the Loan amount term and credit history as they have only two values. We will use the square root transformation as show in the code below.



As we can see that after the transformation the skewness in the columns are removed.

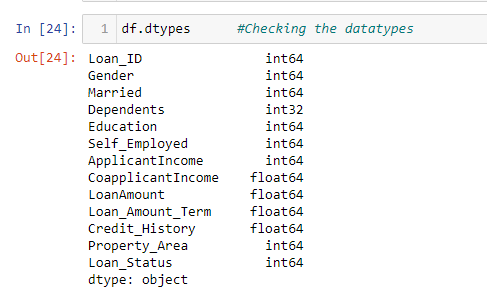
1. Now, I must encode the object type data as we know that a machine can understand only data in numeric form. So, I will use LabelEncoder to encode the object type data. The data are categorical type, so I am using LabelEncoder.

Let’s encode it.



*Exhibit 11*

Checking the datatypes after encoding



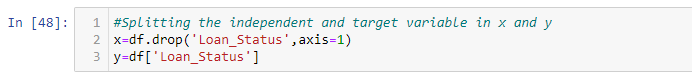
*Exhibit 12*

No object data are available now. I have done partial cleaning of my data.

Data cleaning is completed. My data is getting ready for machine building process.

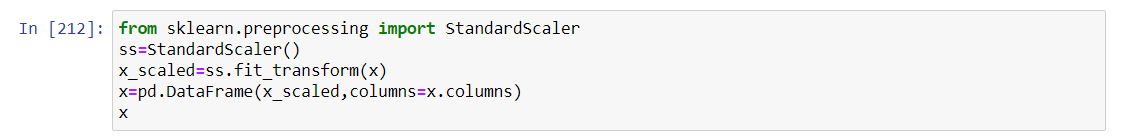
Before I start building the Machine Learning model, I will split my independent data and dependent data into x and y. Then I will scale the data to a standard form using StandardScaler.

**Splitting the data into x and y:**

.

*Exhibit 25*

**Scaling the data using StandardScaler:**



*Exhibit 26*

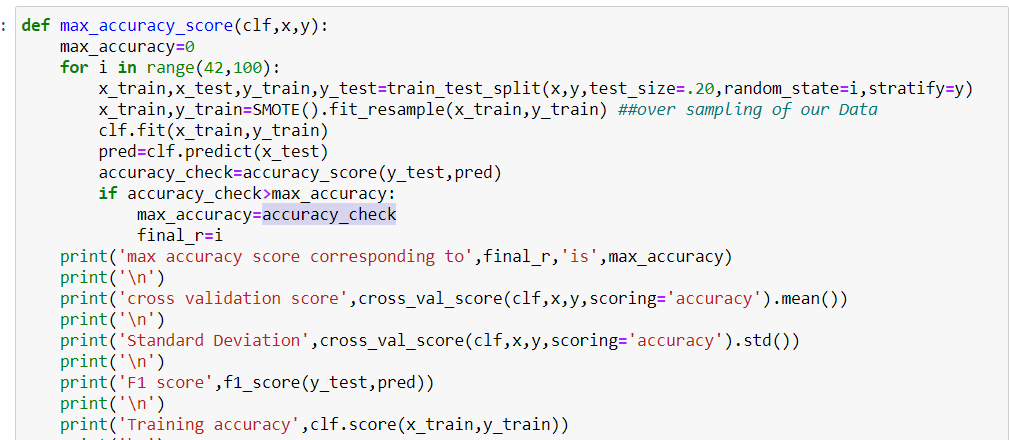
**EDA concluding remark:**

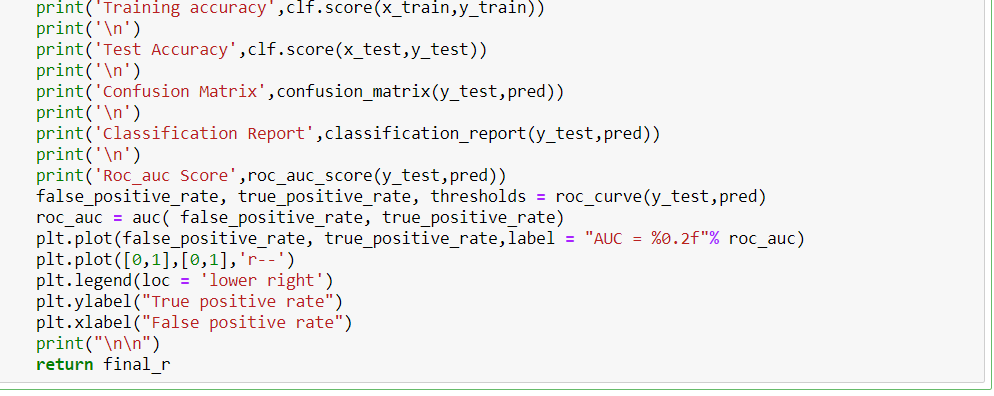
After going through data analysis and data processing, I can conclude that the raw data that I have received is now cleaned and is ready for Machine Learning process. Steps that I follow are listed below:

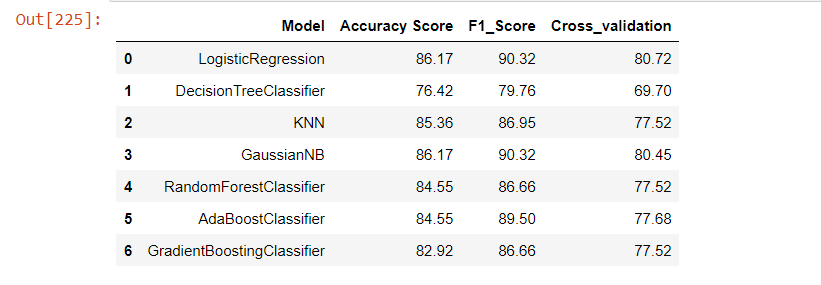
1. At the beginning, I have analysed the data by checking its shape, its datatypes and information regarding presence of null values.
2. After checking for null values, I have seen that there are some null values present. I removed them by replacing them with mode and median.
3. While checking for datatypes, I have seen many object type columns are present, I have encoded them into numeric so that Machine can understand. The encoding process that I have used for categorical type data is LabelEncoder.
4. I also visualized my data using barplots.
5. Then I have checked the statistical summary of the dataset and checked the correlation between the features and the target variable using heatmap.
6. I have removed the outliers present in the dataset by replacing them by quantiles and minimised the skewness in data using ‘sqrt’ method.
7. When my DataFrame gets ready for Machine Learning process, I have split the independent variables and target variable into x and y.
8. Then I scaled my data into a standard form using StandardScaler.
9. As we have seen in the visualization that the class of target variable is imbalanced so we will use the over sampling method SMOTE from the imblearn library.

**Building Machine Learning Models:**

Now I will train several Machine Learning models and compare their results. I will create a function which will take the model instance, X and y variable to give the values. We will pass the various models into the function and check the accuracy score and cross validation score. The model which gives the least difference between the accuracy score and cross validation score becomes are best model.







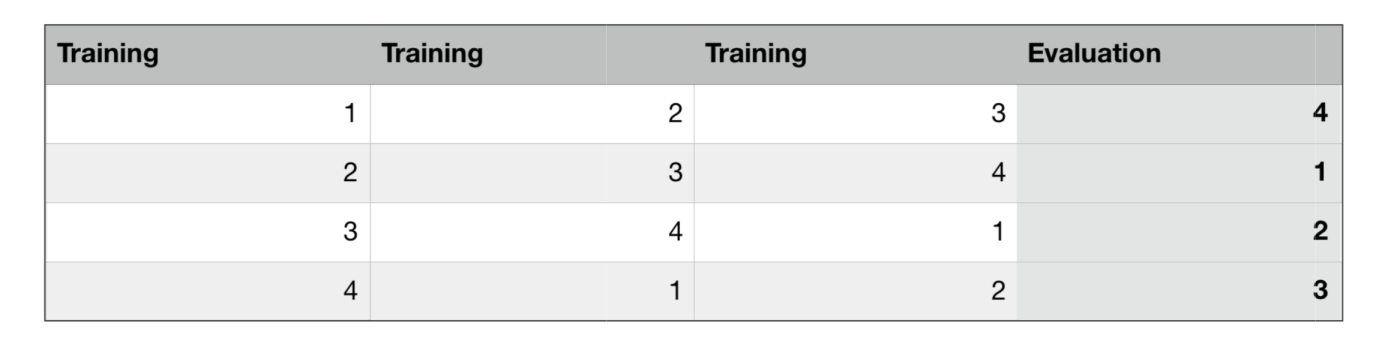
*Exhibit 27*

As we can see, LogisticRegression is giving a good accuracy of 86.17% among all the classification models. It is clear that Logistic Regression is Most generalised model among all because the difference between Accuracy Score and cross validation score is miminum as compared to other models. So this would be our best model. Now we will get some insight about the cross validation.

**K Fold Cross-Validation:**

K-Fold Cross Validation randomly splits the training data into **K subsets called folds**. Let’s image we would split our data into 4 folds (K = 5). Our model would be trained and evaluated 5 times, using a different fold for evaluation every time, while it would be trained on the remaining 4 folds.

The image below shows the process, using 4 folds (K = 4). Every row represents one training + evaluation process. In the first row, the model gets trained on the first, second and third subset and evaluated on the fourth. In the second row, the model gets trained on the second, third and fourth subset and evaluated on the first. K-Fold Cross Validation repeats this process till every fold acted once as an evaluation fold.

****

*Exhibit 28*

*Exhibit 29*

After cross validation, we get the actual accuracy of the model,i.e. 80.72%. Before it was 86.17% because of over-fitting. Now, I will try to increase its performance even better in the following section.

**Hyperparameter tuning:**

A Machine Learning model is defined as a mathematical model with several parameters that need to be learned from the data. By training a model with existing data, we can fit the model parameters. However, there is another kind of parameters, known as **Hyperparameters**, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. Two best strategies for Hyperparameter tuning are:

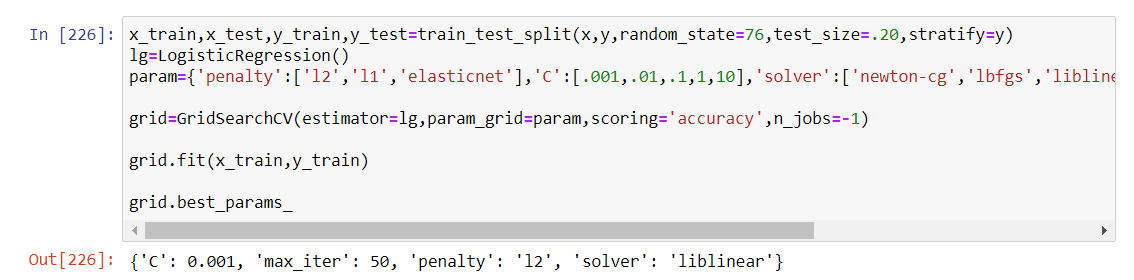
* **GridSearchCV:**

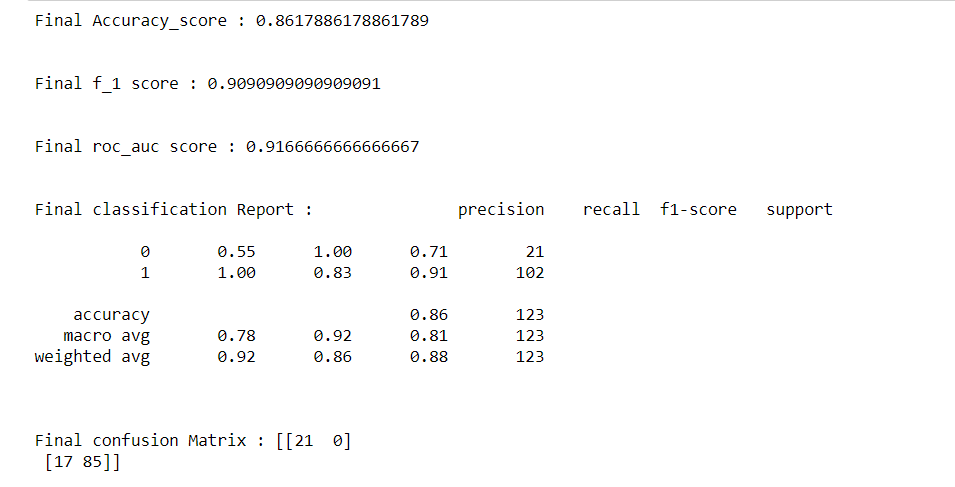
In GridSearchCV approach, machine learning model is evaluated for a range of hyperparameter values. This approach is called GridSearchCV, because it searches for best set of hyperparameters from a grid of hyperparameters values.

* **RandomizedSearchCV:**

RandomizedSearchCV solves the drawbacks of GridSearchCV, as it goes through only a fixed number of hyperparameter settings. It moves within the grid in random fashion to find the best set hyperparameters. This approach reduces unnecessary computation.

Now I will try to tune the hyperparameters and check if we can increase the model's accuracy.



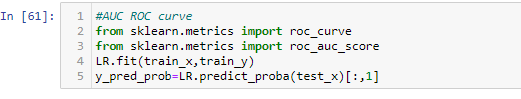


*Exhibit 30*

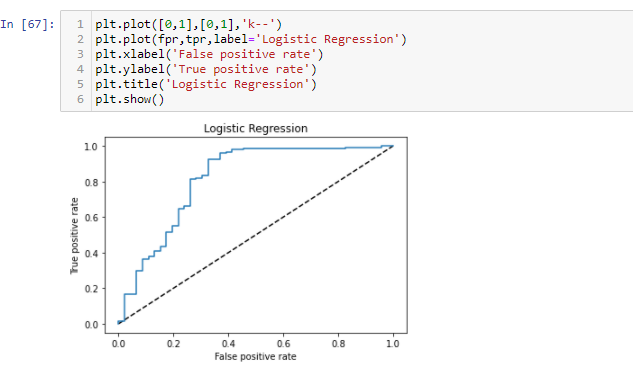
The accuracy after hyperparameter tuning is also 86.17%. The best parameters are {‘C’:0.001, ‘max\_iter’:50, ‘penalty’: ‘l2’, ‘solver’: ‘liblinear’}. We can see that by hyperparameter tuning our F1 score and Roc\_auc score has improved. Our Accuracy score remained same. Hence we will save this as our best model.

**ROC AUC Curve:**

Another way to evaluate and compare our binary classifier is provided by the ROC AUC Curve. This curve plots the true positive rate (also called recall) against the false positive rate (ratio of incorrectly classified negative instances).



*Exhibit 31*

****

*Exhibit 32*

The dotted line in the middle represents a purely random classifier (e.g a coin flip) and therefore our classifier should be as far away from it as possible.

**ROC AUC score:**

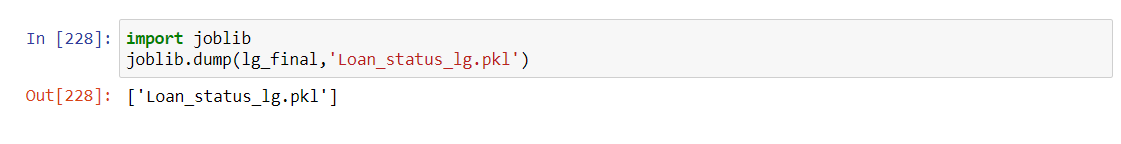
The ROC AUC Score is the corresponding score to the ROC AUC Curve. It is simply computed by measuring the area under the curve, which is called AUC.

A classifier that is 100% correct, would have a ROC AUC Score of 1 and a completely random classifier would have a score of 0.5.

*Exhibit 33*

The score is 0.9166 which is good.

Now, I am saving the model.



*Exhibit 34*

**Conclusion:**

We started with the data exploration where we got a feeling for the dataset, checked about missing data, and learned which features are important. During this process we used seaborn and matplotlib to do the visualizations. During the data pre-processing part, we converted features into numeric ones, grouped values into categories and computed missing values. Afterwards we started training different machine learning models, picked one of them (LogisticRegression) and applied cross validation on it. Then we discussed how LogisticRegression works, took a look at the importance it assigns to the different features and tuned its performance through optimizing its hyperparameter values.