## **Description:**

1. Prediction of Loan Eligibility for Dream Housing Finance company is a Hackathon project on Datahack. ([link](https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/))
2. This project is implemented using Gradient Boosting Classifier.

## **Problem:**

Company wants to automate the loan eligibility process based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have provided a dataset to identify the customers segments that are eligible for loan amount so that they can specifically target these customers.

## **Data Source:**

[Datahack](datahack.analyticsvidhya.com)

Download Data Sets:

* [Training Data Set](https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/download/train-file)
* [Testing Data Set](https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/download/test-file)

### Data Dictionary:

* Training Data:

Train file CSV containing the customers for whom loan eligibility is known as 'Loan\_Status'.

|  |  |
| --- | --- |
| Variable | Description |
| Loan\_ID | Unique Loan ID |
| Gender | Male/ Female |
| Married | Applicant married (Y/N) |
| Dependents | Number of dependents |
| Education | Applicant Education (Graduate/ Under Graduate) |
| Self\_Employed | Self employed (Y/N) |
| ApplicantIncome | Applicant income |
| CoapplicantIncome | Coapplicant income |
| LoanAmount | Loan amount in thousands |
| Loan\_Amount\_Term | Term of loan in months |
| Credit\_History | credit history meets guidelines |
| Property\_Area | Urban/ Semi Urban/ Rural |
| Loan\_Status | (Target) Loan approved (Y/N) |

* Testing Data:

Test file**:** CSVcontaining the customer information for whom loan eligibility is to be predicted.

|  |  |
| --- | --- |
| Variable | Description |
| Loan\_ID | Unique Loan ID |
| Gender | Male/ Female |
| Married | Applicant married (Y/N) |
| Dependents | Number of dependents |
| Education | Applicant Education (Graduate/ Under Graduate) |
| Self\_Employed | Self employed (Y/N) |
| ApplicantIncome | Applicant income |
| CoapplicantIncome | Coapplicant income |
| LoanAmount | Loan amount in thousands |
| Loan\_Amount\_Term | Term of loan in months |
| Credit\_History | credit history meets guidelines |
| Property\_Area | Urban/ Semi Urban/ Rural |

* Final Result Format:

The final submission file format should be in following manner. The Loan\_ID column contains unique loan IDs and the Loan\_Status column contain predicted result of loan status.

The final submission file should be in CSV format

|  |  |
| --- | --- |
| Variable | Description |
| Loan\_ID | Unique Loan ID |
| Loan\_Status | (Target) Loan approved (Y/N) |

## **Why Gradient Boosting Classifier?**

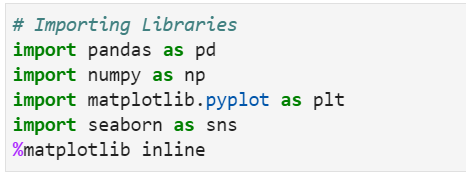
* It provides predictive scores which is better than other classifiers.
* It often provides predictive accuracy that cannot be trumped.
* Optimize on differentiable loss function.
* Provides several Hyper Parameter Tuning options that make the function fit very flexible.

The Project is divided into Two parts:

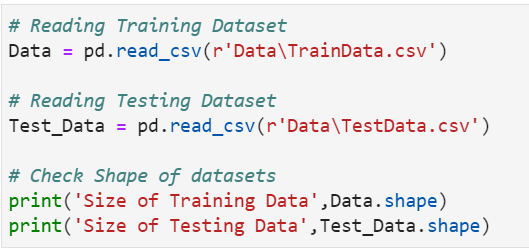
1. **Building Machine Learning Model.**
2. **Predicting the Outcomes of Test Dataset**.

# **Part I: Building ML Model**

## **Step 1st: Importing Important Libraries**

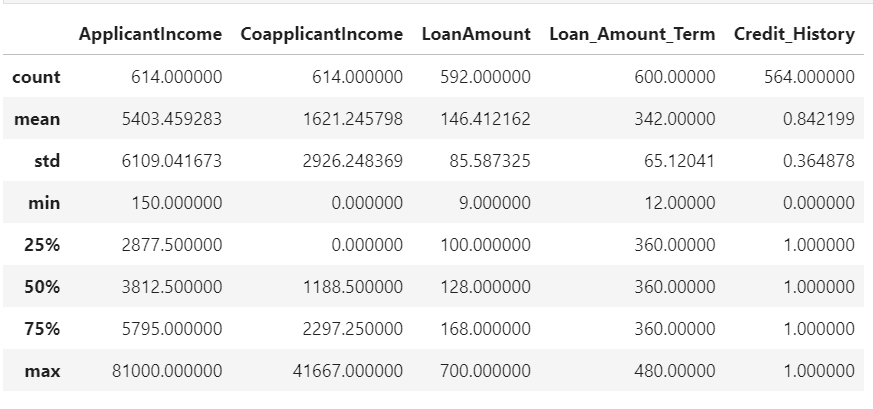
****

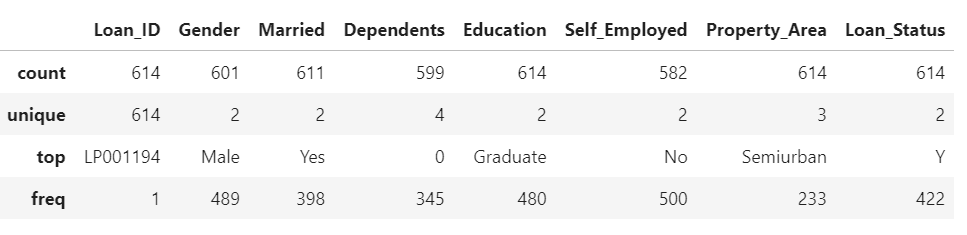
## **Step 2nd: Reading Train & Test Data Set**

****

## **Step 3rd: Descriptive Statistics**

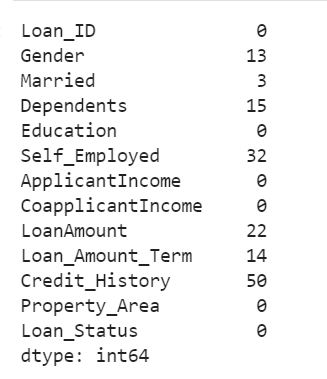
**Descriptive statistic of Quantitative Data**

** Descriptive statistic of Qualitative Data**

****

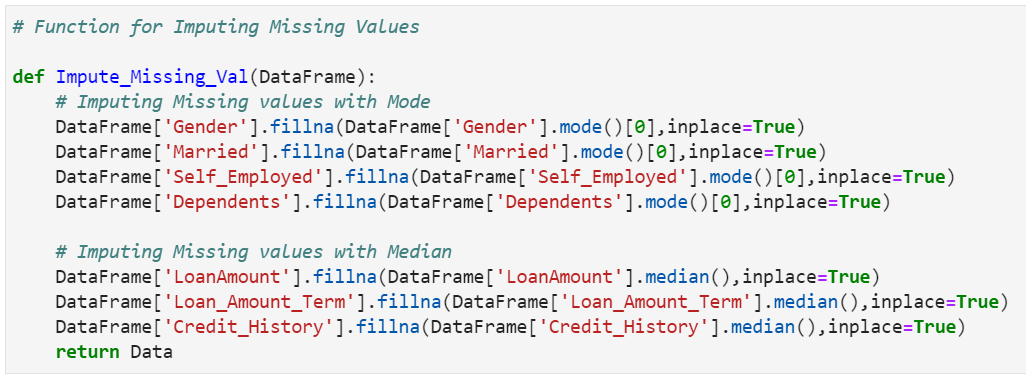
## **Step 4th: Data Cleaning**

The Missing Values present in data set are:

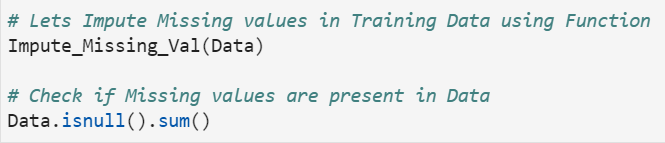
****

### **Imputing Missing Values**

Here we created the function for imputing Null values So we can reuse the function for Testing data set as well. In that function we use the pandas fillna method to fill null values. The Gender, Married , Self Employed, and Dependents features are fill with their respective modes and Loan Amount, Loan Amount History, Credit History are fill with medians.

****

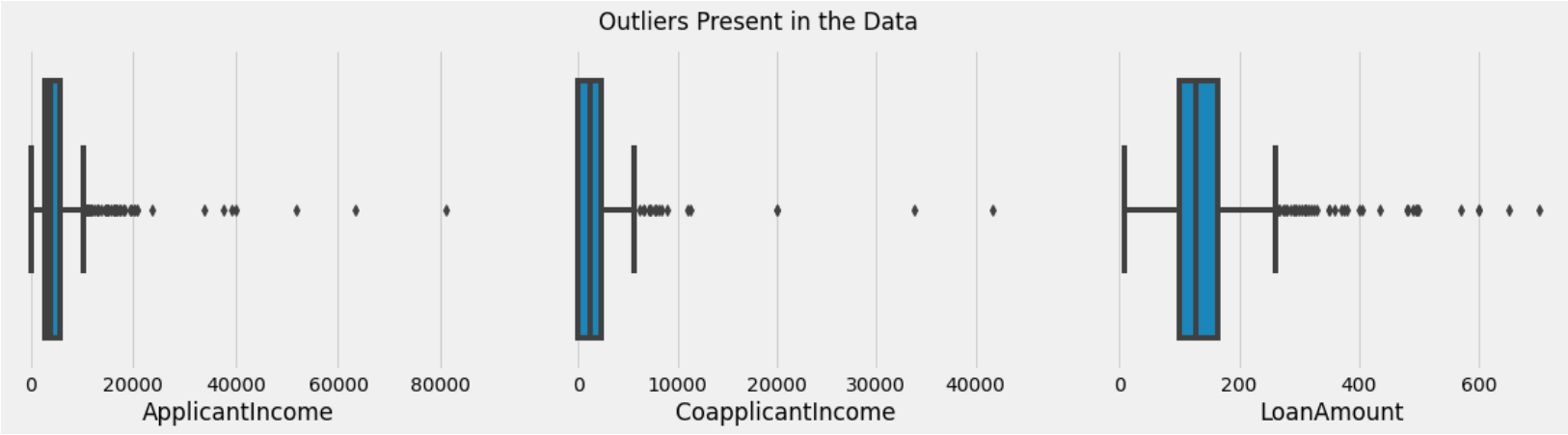
Imputing the Null values using function Impute\_Missing\_Val and check if there are null values present in data after calling a function

****

Total Null values are zero after calling a function

### **Visualize and Remove Outliers**

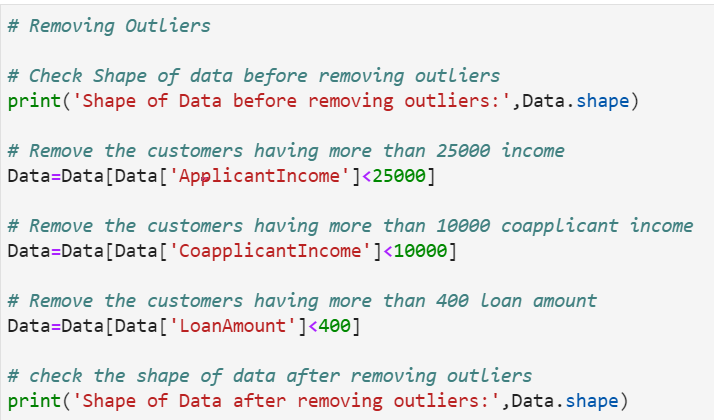
Visualize the Applicant Income, Co-applicant Income and Loan Amount feature of Data using seaborn boxplot method.

****

**Remove Outliers**

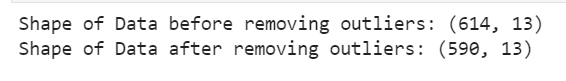
As we see in above plot too many outliers present in the data and we removed them as follows

1. Removed customers having Applicant Income more than 25000.
2. Removed customers having Co-Applicant Income more than 10000.
3. Removed customers having Loan Amount more than 400.

****

We check the shape of the data before and after removing the data

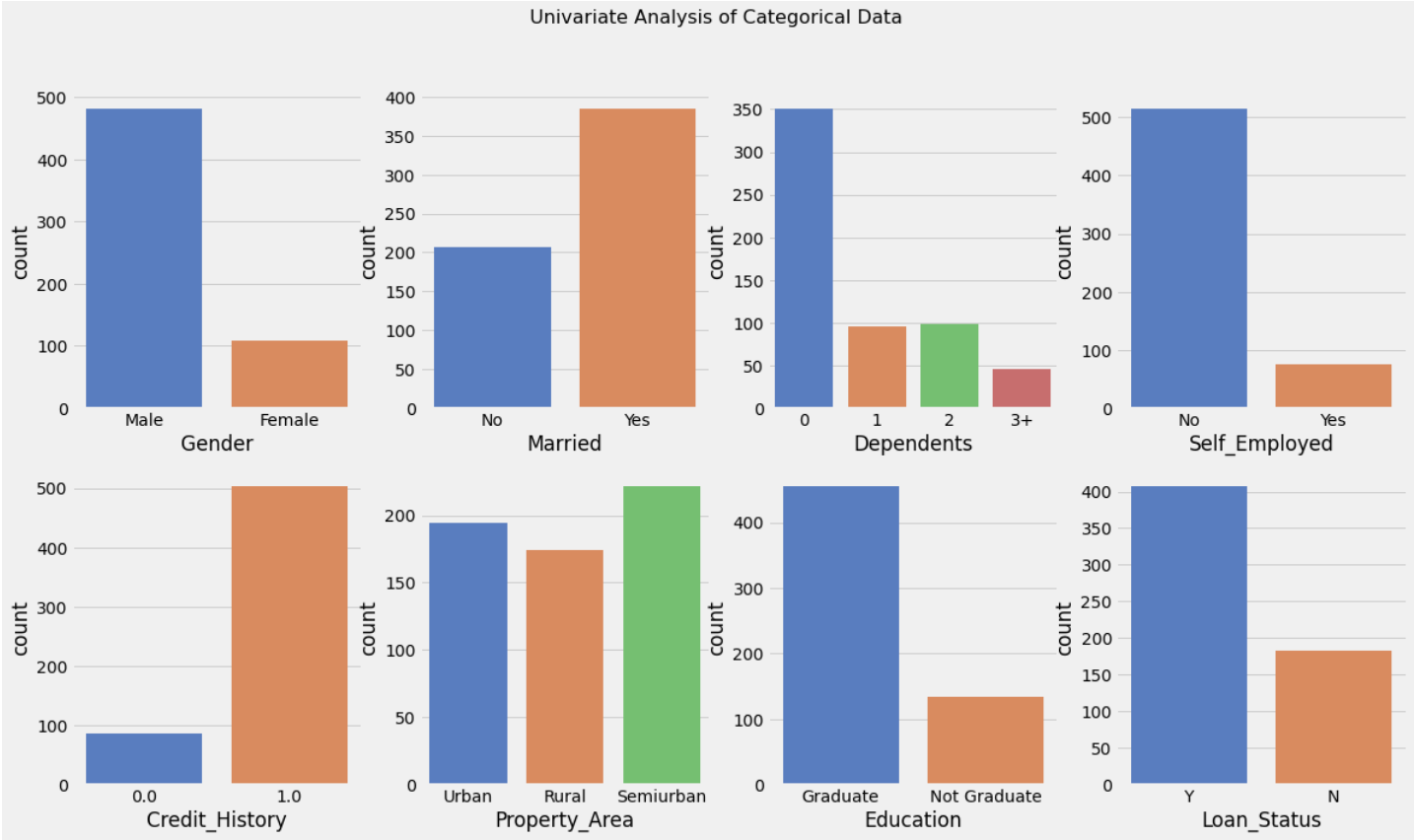
Shape of data before and after removing outliers

****

## **Step 5th: Univariate Data Analysis**

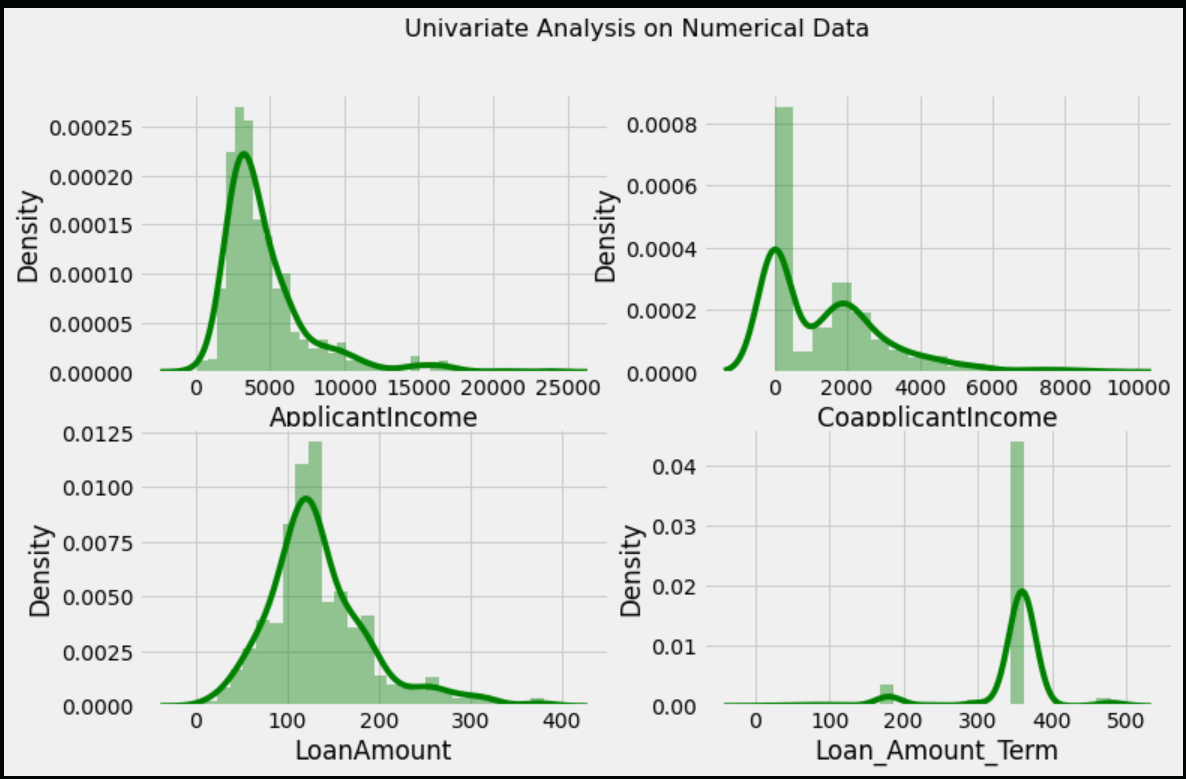
1. **On Qualitative Features of Data**

Visualize the different categorical features of data using coutnplot

****

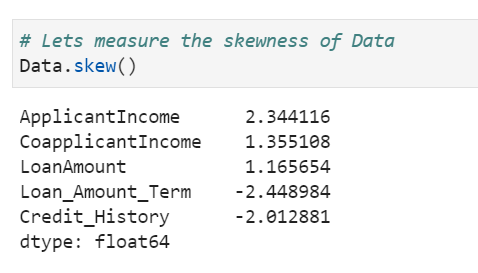
1. **On Quantitative Features of Data**

Visualize the quantitative features of the data using distplot

****

As we see in above plot the data is highly skewed.

Check the skewness of data

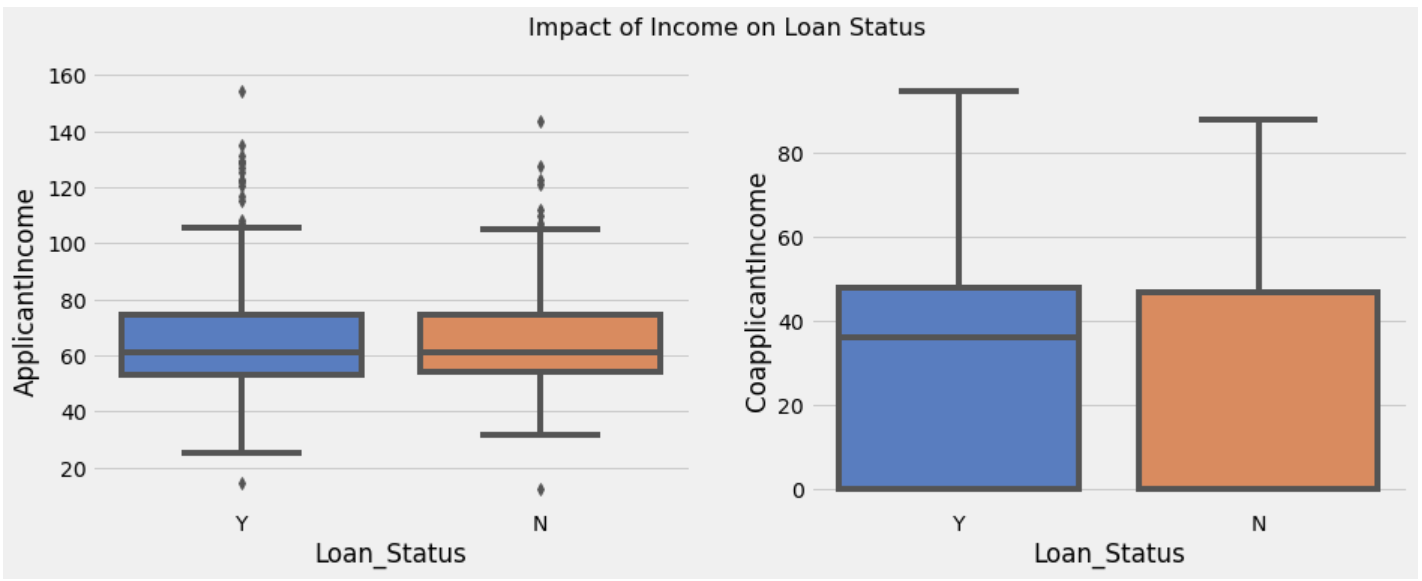
****

The skewness of data is high so we removed in the data preparation

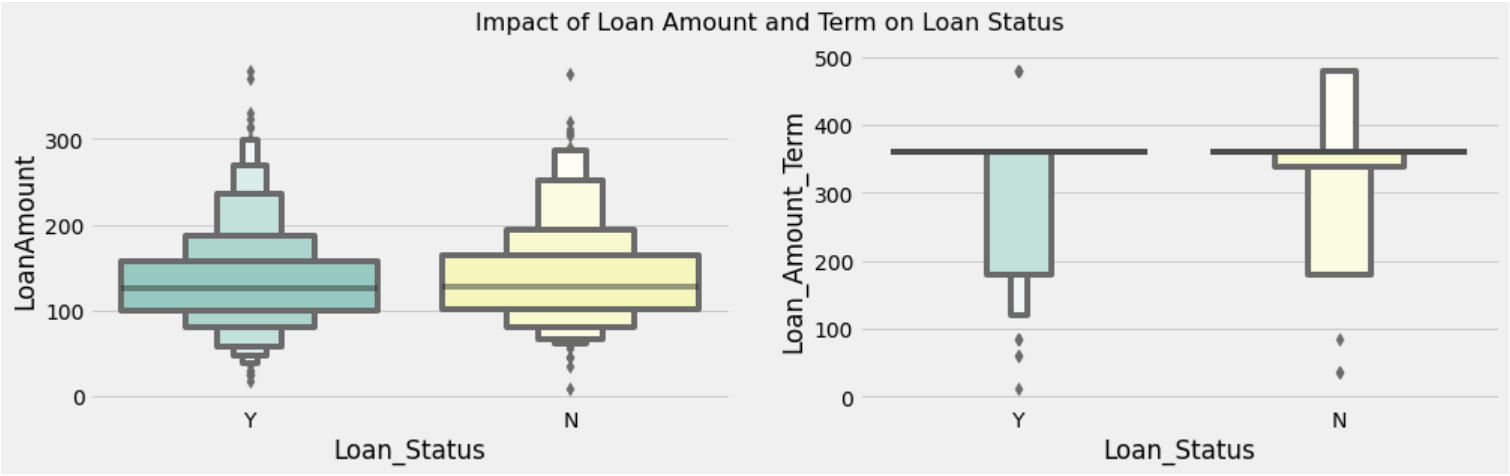
## **Step 6th: Bivariate Data Analysis**

**Bivariate Analysis on Quantitative Features**

Visualize the impact of Applicant Income and Co-applicant Income of the Loan Status

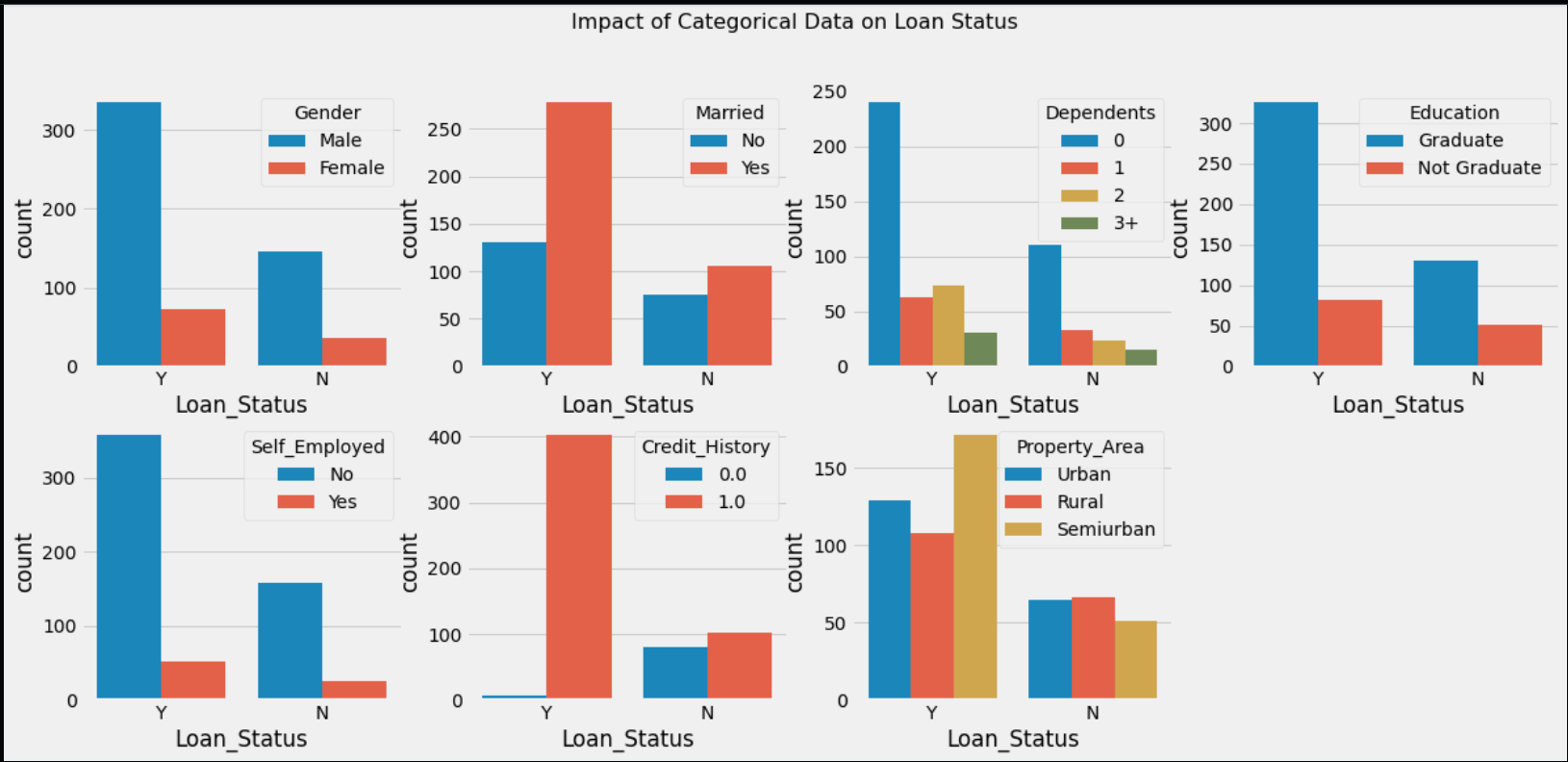
****

Visualize the Impact of Loan Amount and Loan Amount Term on Status of Loan

****

**Bivariate Analysis on Qualitative Features**

Visualize the relation between different categories of Data with Target variable (Loan\_Status)

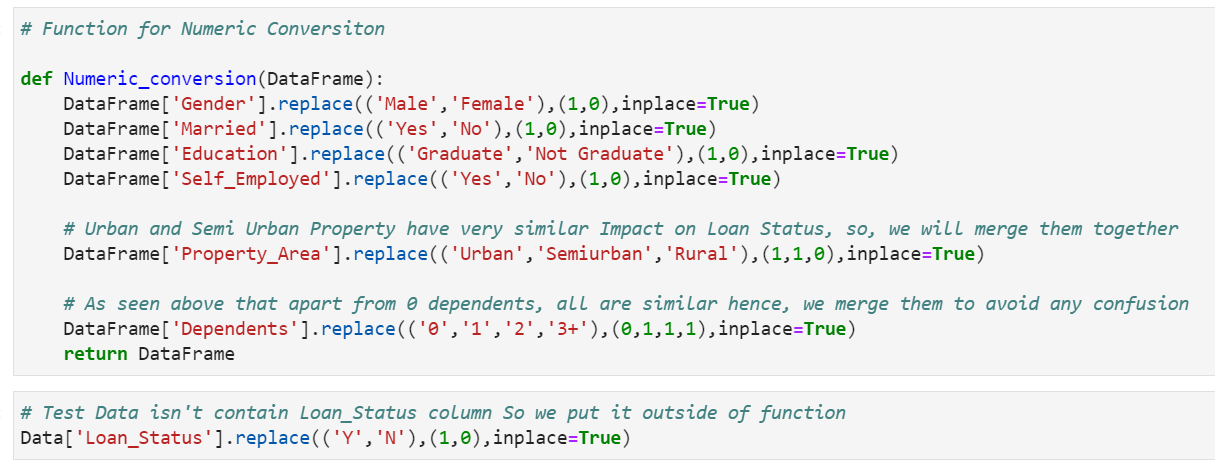
****

**As we see in above chart the Credit history is high relation with Loan Status**

## **Step 7th: Data Preparation**

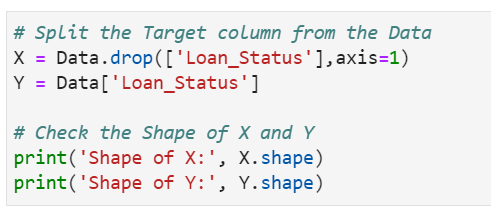
#### **Data Encoding**

We check the all unique values of the categorical columns of data so we can replace them using pandas replace method. Here we created the function to encode the data so we can use it later for Testing Data.

****

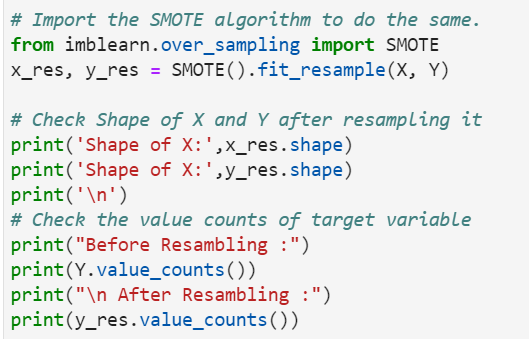
#### **Set Target Column**

We know that Loan Status is our Target column so set Y (dependent variable) and X (independent variable)

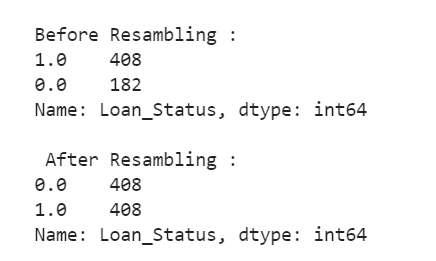
****

#### **Resembling Data**

It is very important to resample the data, as the Target class is Highly imbalanced. Here We are going to use Over Sampling Technique to resample the data. Use SMOTE algorithm to do the same.

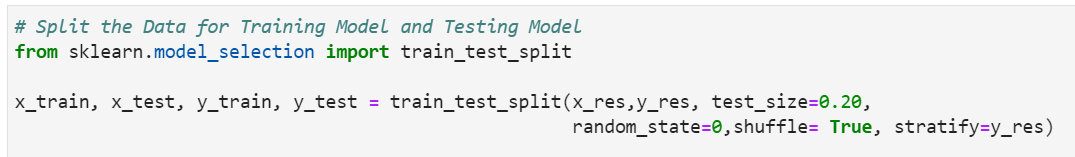
****

Check the data begore sampling and after sampling

****

## **Step 8th: Data Splitting**

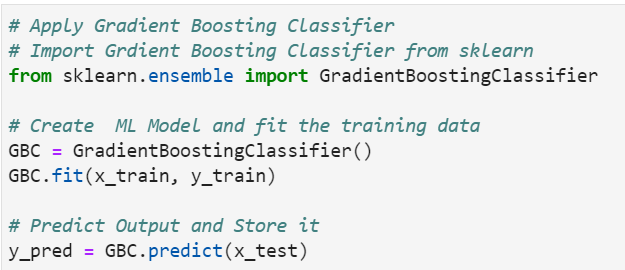
Split the data to Training set and Testing set for building a model and to check the accuracy of model

****

## **Step 9th: Applying ML Model**

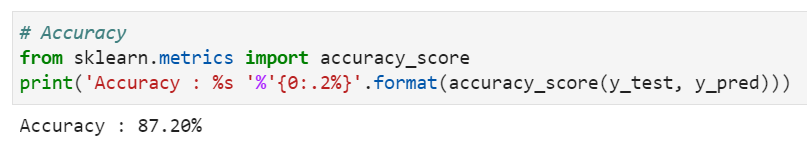
Creating a Gradient Boosting Classifier model and fit the data to training of model

Predict the output of test data set and store it to a variable

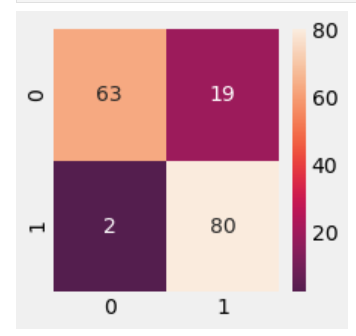
****

## **Step 10th: Check the Accuracy of Model**

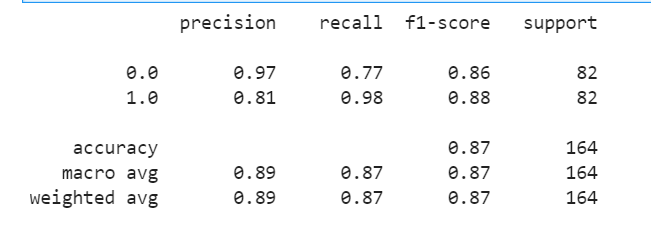
#### **Accuracy: 87.20%**

****

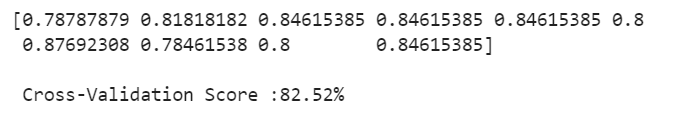
#### **Confusion-Matrix:**

****

#### **Classification Report:**

****

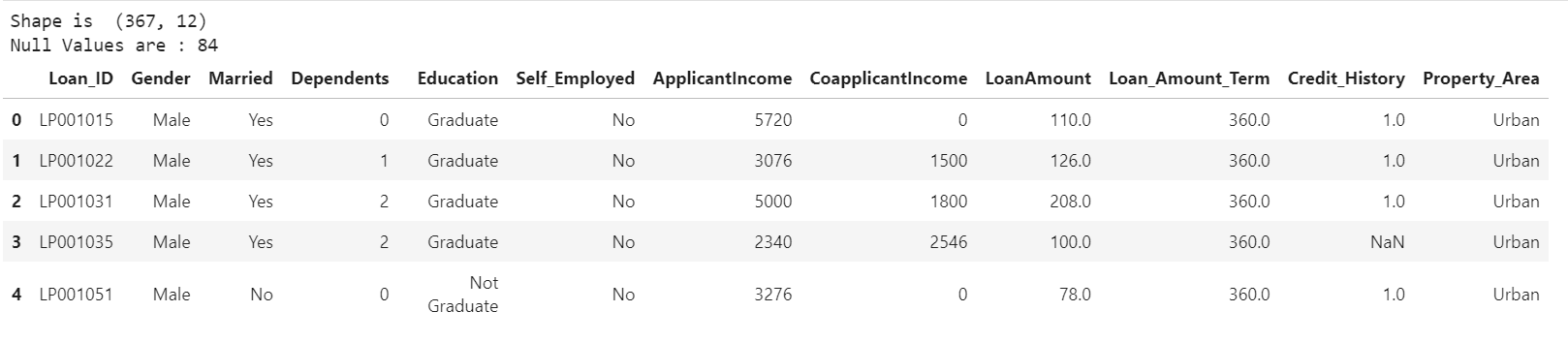
#### **Cross-Validation score: 82.52%**

****

# **Part II: Predict the Outcomes of Test Data**

## **Step 1st: Check Testing Data Set**

Input Data:

****

Shape: (367,12)

Total Null Values present in Testing data: 84

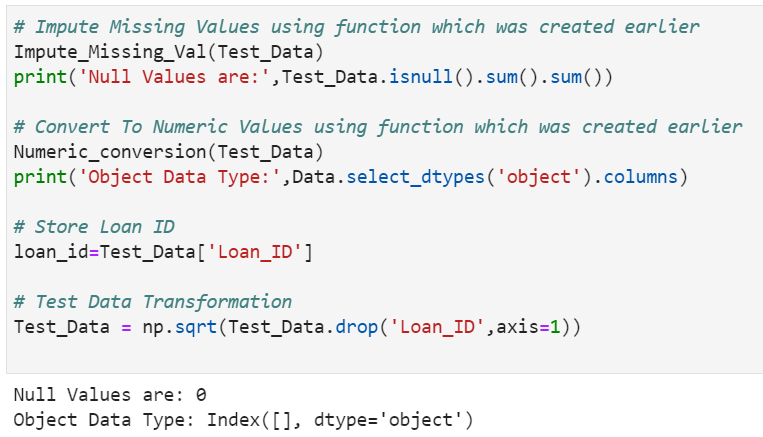
## **Step 2nd: Transform Data**

Total Null Values are 84

Impute missing values using function the function Imput\_Missing\_Val which was created earlier.

Convert object type features to numeric type features using the function Numeric\_Conversion which was created earlier.

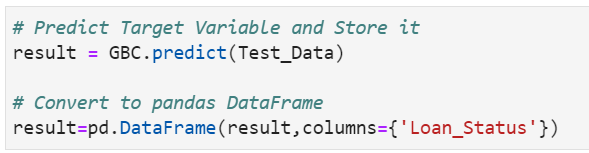
Perform sqare root transformation to transform the data

****

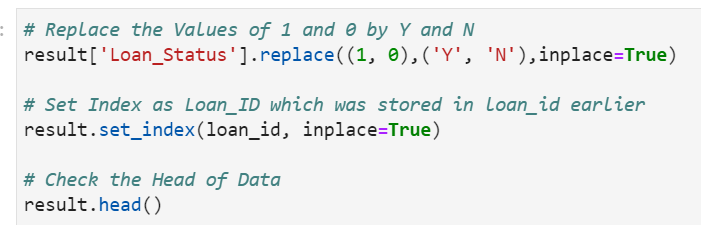
## **Step 3rd: Apply ML Model**

Apply ML model and fit the data to predict the outcomes

The output is in numpy array format so we convert it in pandas DataFrame

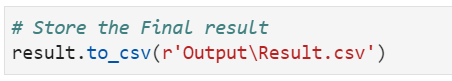
****

The output is in numeric type so we convert it back to normal Y and N type and set index as Loan ID

****

## **Step 4th: Save the Result**

The result is saved in CSV format using pandas**.**

****

## **Result:**

