**Predicting the Loan Status.**

**Load Application Status Prediction**is a task that can be done based on historical information of the customer and bank. By checking the dataset already existed regarding the status of the Load Application and creating a model will help us to Predict the further Loan Application Status.

In this post, we are going to build a model that can predict whether the loan of the applicant will be approved or not on the basis of the details provided in the dataset. Dataset includes details of applicants who have applied for loan. The dataset includes details like credit history, loan amount, their income, dependents etc. Let’s start.

# Dataset:

# We have a dataset with several features. The dataset comprises of 2 things: Independent variables and dependent variables.

* The **independent variable** is the cause. Its value is independent of other variables in your study.
* The **dependent variable** is the effect. Its value depends on changes in the independent variable.

# Below is the list of independent and dependent variables in our dataset: -

**Independent Variables:**

- Loan\_ID

- Gender

- Married

- Dependents

- Education

- Self\_Employed

- ApplicantIncome

- CoapplicantIncome

- Loan\_Amount

- Loan\_Amount\_Term

- Credit History

- Property\_Area

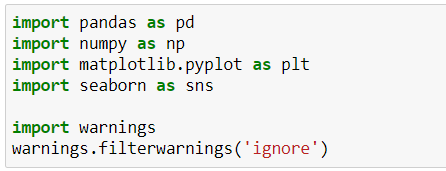
**Dependent Variable (Target Variable):**

- Loan\_Status

**Problem Definition**: We need to build a model that can predict whether the loan of the applicant will get approved or not on the basis of the details provided.

# Importing Required Libraries:

We started the project by importing basic libraries that are required to start a project on python 3 notebook.



As seen above I have imported below libraries and creates instances for all which helps to perform different tasks in my project:

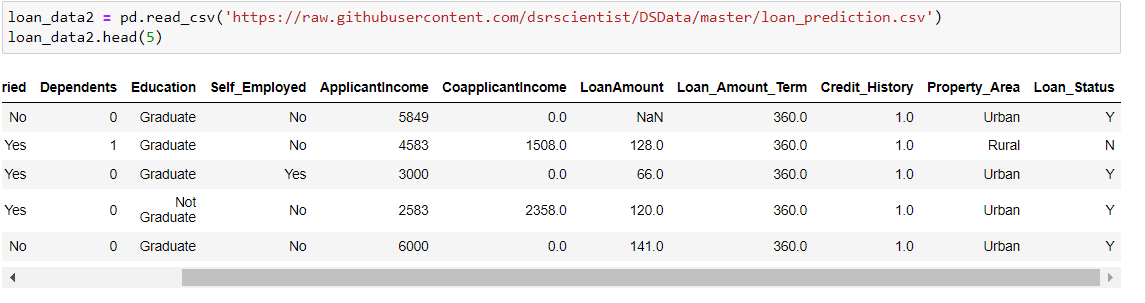
1. **Pandas**: Helps us with the data analysis.
2. **NumPy**: Helps us in performing several mathematical operations.
3. **Matplotlib**.pyplot: Helps us with visualization for 2D plots array.
4. **Seaborn**: Which is used for data visualization and exploratory data analysis.
5. **Warnings**: Warnings are provided to warn the developer of situations that aren't necessarily exceptions. Usually, a warning occurs when there is some obsolete of certain programming elements, such as keyword, function or class, etc. A warning in a program is distinct from an error.

Let’s start building a machine learning model for loan application prediction status. These are the few steps that we will follow while handling the problem: -

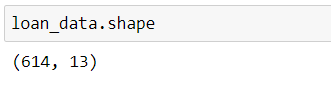
1. Importing dataset and understanding the problem.
2. EDA.
3. Data Cleansing
4. Feature Engineering
5. Encoding
6. Split the data into training and test data sets.
7. Model Selection
8. Model Validation
9. Interpret the result
10. Saving Model

**Importing the dataset and understanding the problem:**

After importing necessary libraries, we imported the dataset, on which we need to work using the pandas read function. The dataset is in csv file so we used **pd.read\_csv** to fetch the dataset.

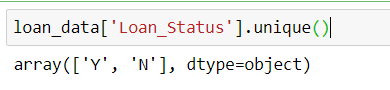


Above is the image showing command to import the variable and saved the dataset to an instance. Also, the above dataset shows first 5 lines of our dataset. The last column of the dataset that is Loan\_Status is our target variable which is a categorical column.



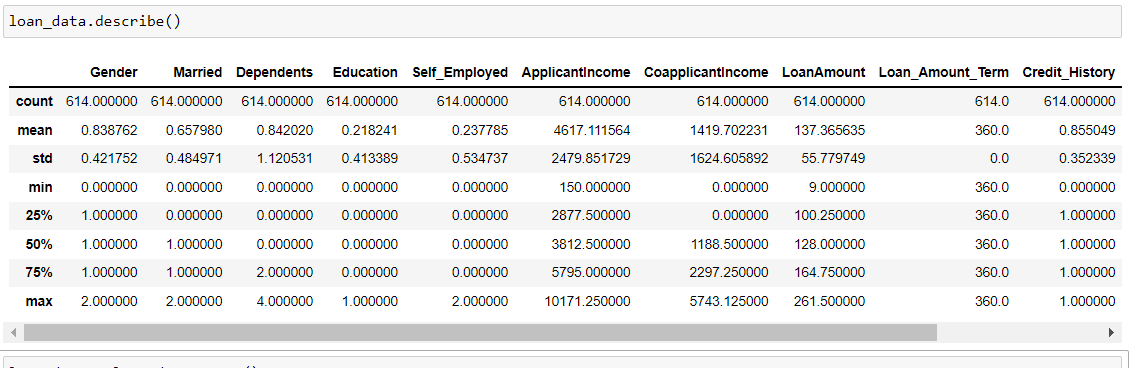
The above image shows the number of rows, which is ***614*** and number of columns which is ***13*** in our dataset.

Once we checked the dataset, we now checked the target variable which is loan status: a-



From the above code we have identified that we are having a categorical output column consists of two values or categories, Yes and No. Hence, we have to build a classification model to predict the loan\_satatus. We have 12 independent columns which helps us in predicting the model.

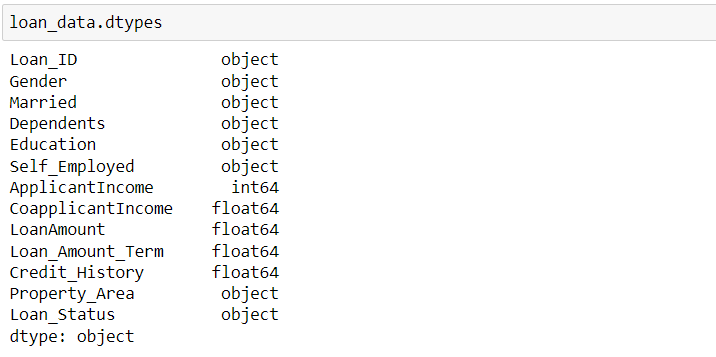
After checking data will **check description** and get to know minimum value, max value, standard deviation etc.:



Describe functions gives us the values of mean, standard deviation, 25th percentile, 75th percentile etc. mathematical values for each column of the dataset.

**Checking data types our dataset:**

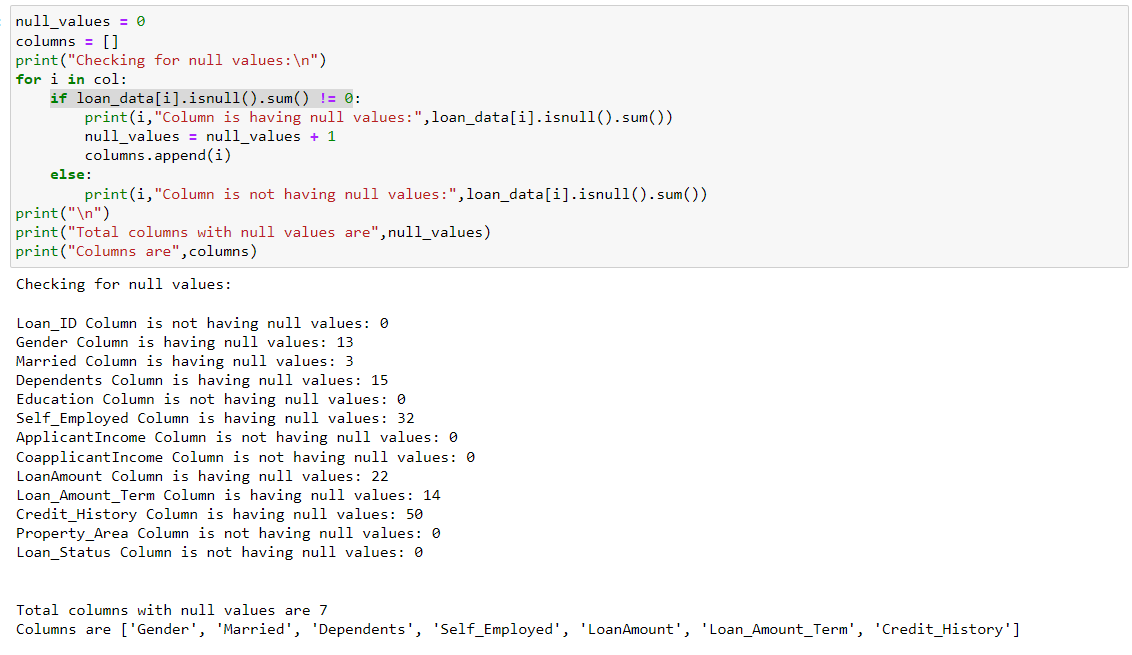
Now, we are checking the datatypes of our dataset: -

****

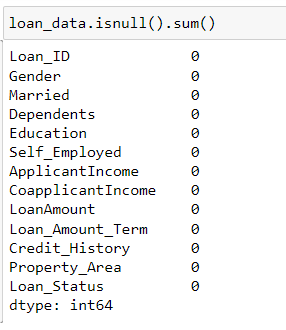
Above we can check that we have 8 columns in Object datatype non-numeric and 5 columns as int datatype numeric form. We need to encode the dataset as our machine algorithms doesn’t understand the object or string data.

**Checking for the Null Values: -**

Checking for the null values and handling them accordingly using mean, median and mode.



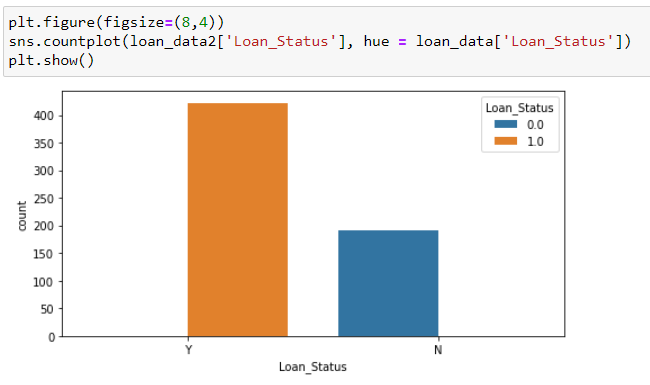
After Handling null values,

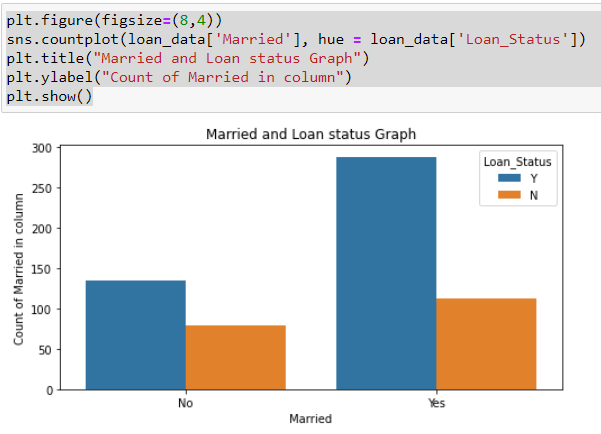
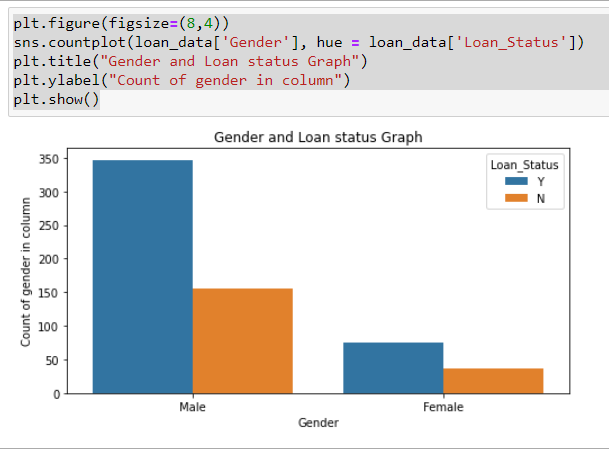


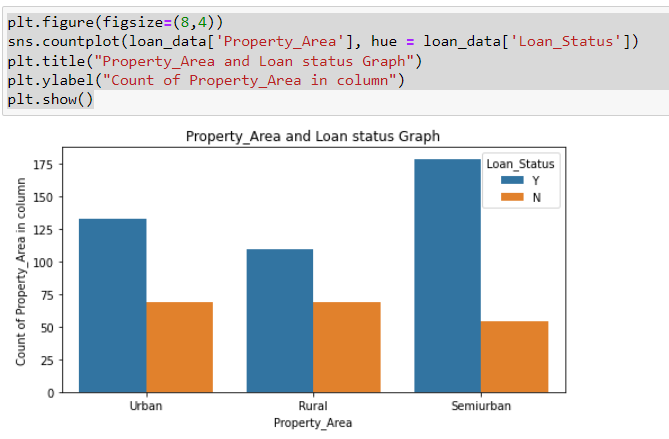
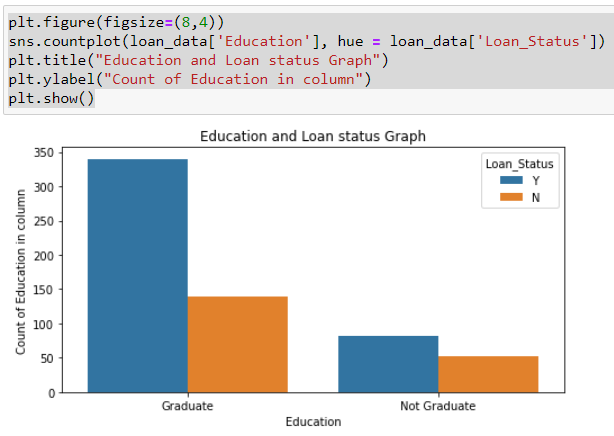
We, finally have removed all the null values from the dataset.

**Visualization:**

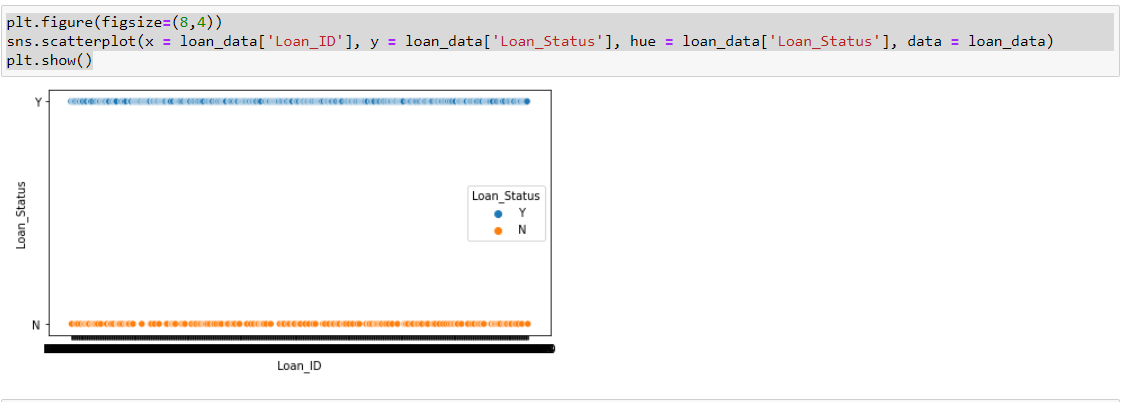
We have plotted the target variable that is Loan\_Status, we can check that the column is having categorical data and also number of Yes are greater than No, which states that our data is having class imbalancing.

****





Few count plots used to analyze the nominal columns. These columns represents the count of several independent variables and how many of the loans are approve or not.

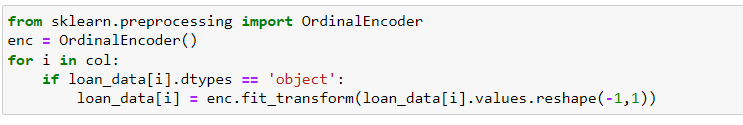






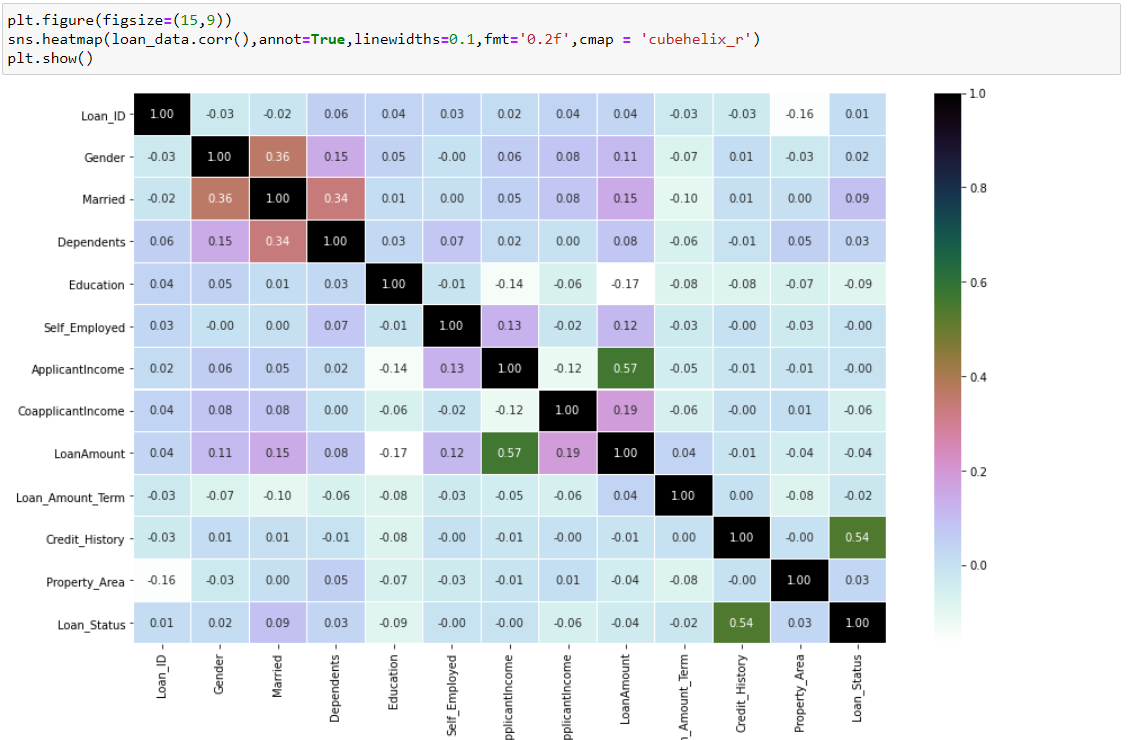
**Data Encoding: -**

As we can check that we have a dataset where mixed types of data are present, so we need to convert the data into numeric. So, we used ordinal encoder to get the job done. We can call ordinal encoder from the sklearn.preprocessing.



**Checking Correlation between Feature variable and Target Variable:**

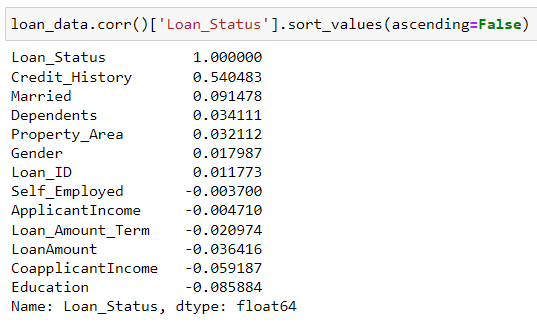
As, we have performed basic action on the dataset we will now check the collinearity of the dataset. Collinearity shows the relation of independent variables with the target variable. Same can be checked using a heatmap (multivariate analysis).



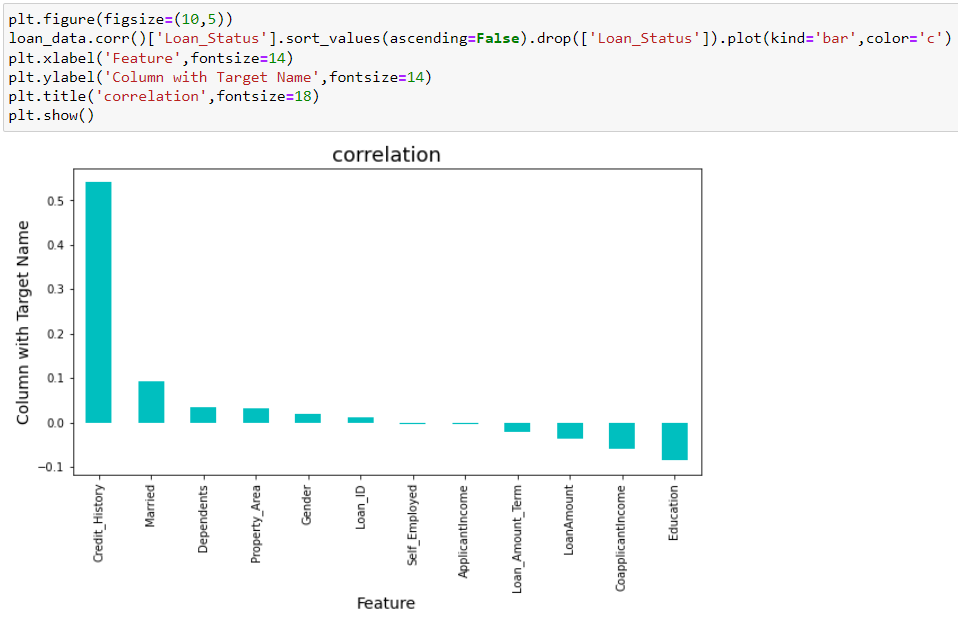
The columns in the above heatmap shows the relation of the variables with each other and with the target variable. Dark color represents that the columns are highly corelated with each other. Light color represents the opposite. We can see that few independent columns are correlated with other independent columns which states that we are having some multicollinearity.

Also, we can check that credit history column is having a very high correlation with the loan\_satatus (Target Variable).

Below, we have arranged all the columns with their correlation with the target variable.

****

We can check that credit history is highly related to the target variable and education is the least related.

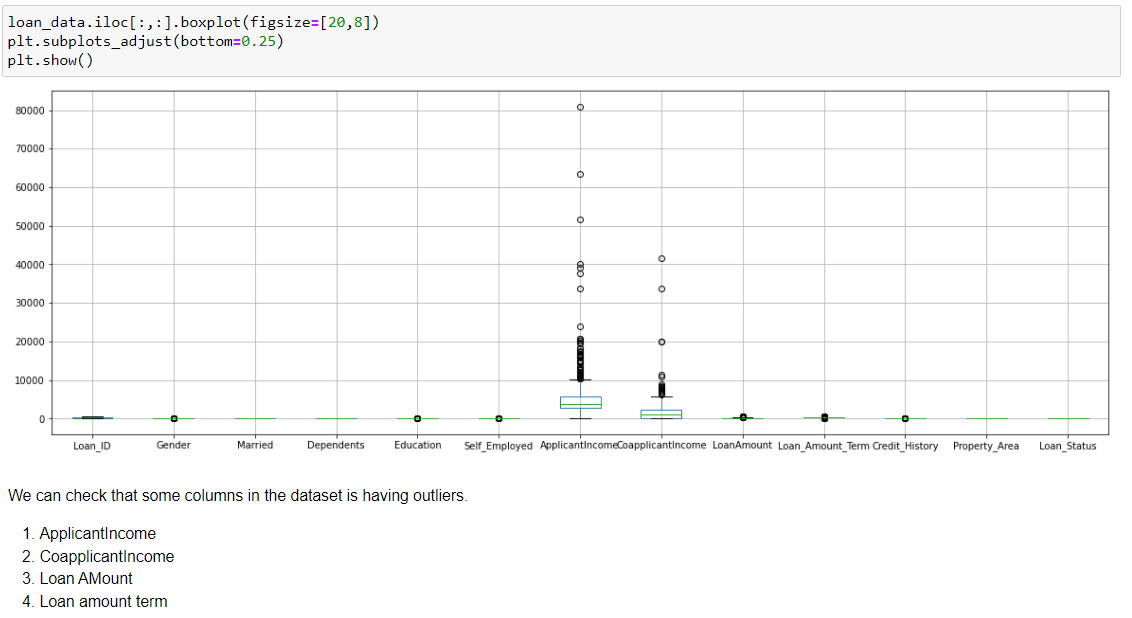


The above is the graphical representation of the correlation of independent variables with the target variable. Here also we can see that credit history is highly corelated and education is the least corelated.

**Outlier Detection and Removal:**

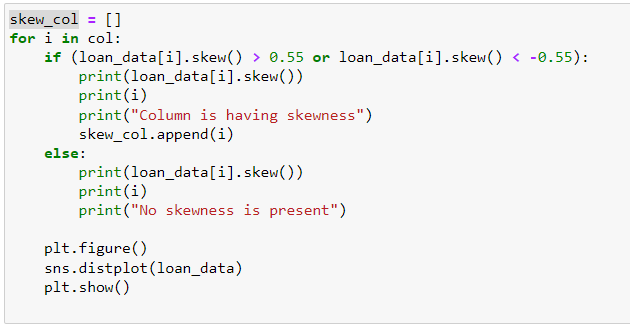
An **outlier** is a data point that is noticeably different from the rest. They represent errors in measurement, bad data collection, or simply show variables not considered when collecting the data.

Below is the graph showing outliers present in the dataset.

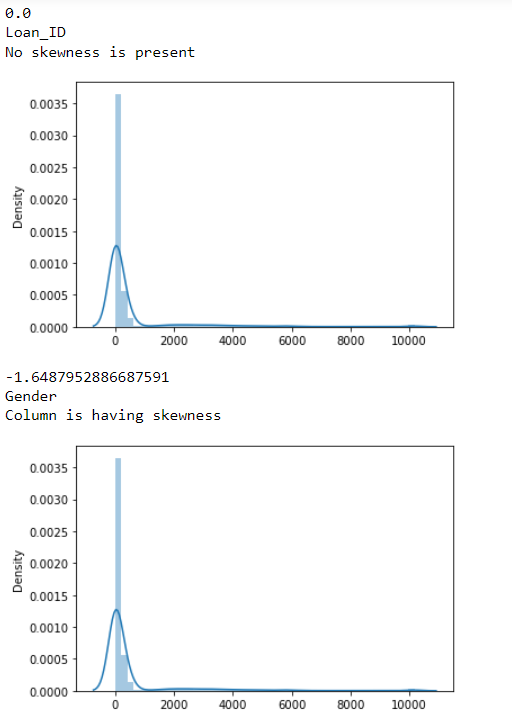
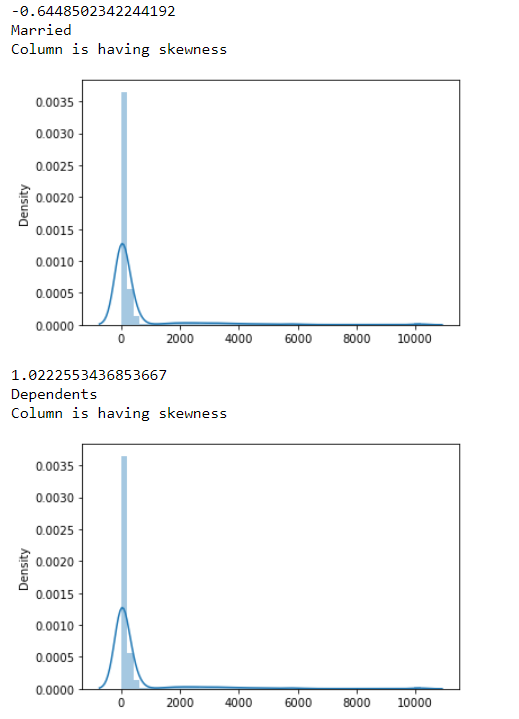


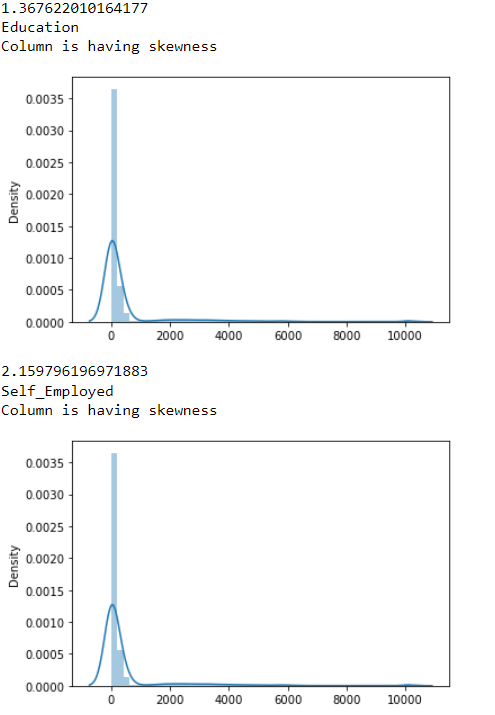
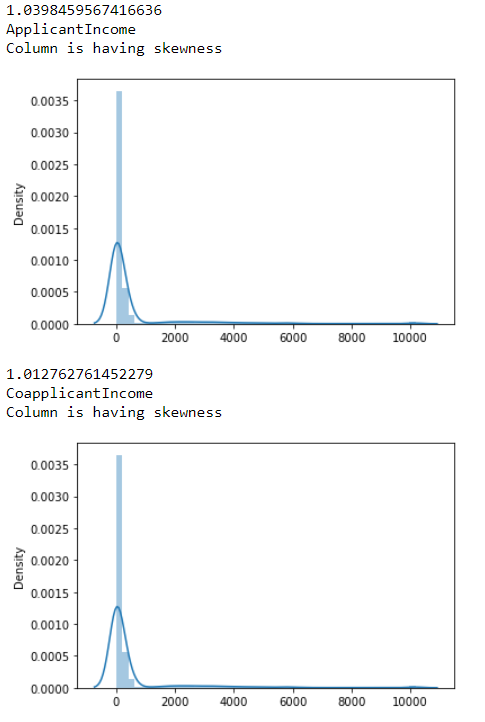
We were getting a data loss of 6.5% using z-score, so we removed outliers with the help IQR method.

**Detecting Skewness and Removal:**

Skewed data is common in data science; skew is the degree of distortion from a normal distribution. Skewness refers to a distortion or asymmetry that deviates from the symmetrical bell curve, or [normal distribution](https://www.investopedia.com/terms/n/normaldistribution.asp), in a set of data. 

We use a for loop to check the skewness, below is the distplot showing distribution of the data. We considered that is a column is having skewness greater than 0.55 and less than -0.55, the data is skewed in that column.

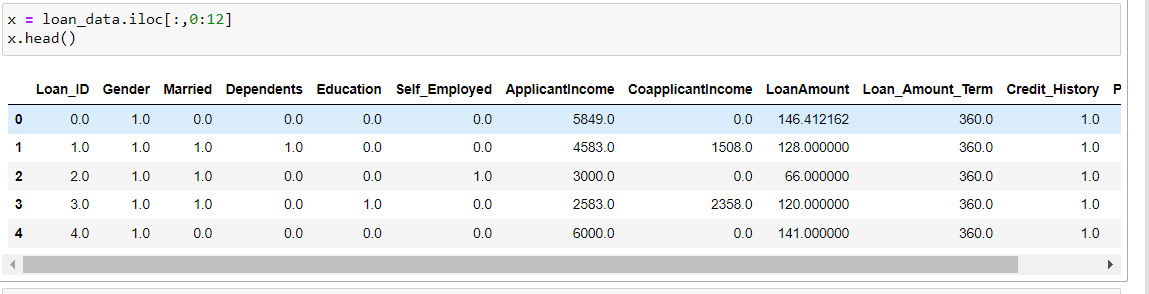
 

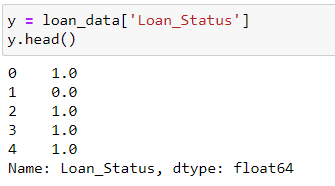
 

Above is the graphical representation of the distribution plot, which represents that distribution of data over the column. We can see that many of the columns are skewed means that their data is not normally distributed or it is out of the range. Most of the data is skewed to right side. We will use power transform (Yeo-Johnson) to remove the skewness from the dataset.

**Splitting the dataset: -**

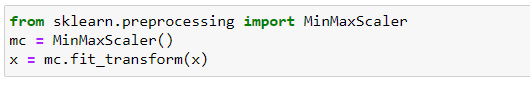
We will now split the dataset into X and Y, where X contains all the independent variable and Y contains the target variable. Please see the picture below: -





**Scaling of Values:**

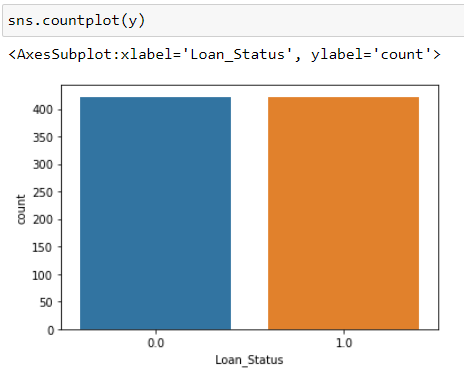
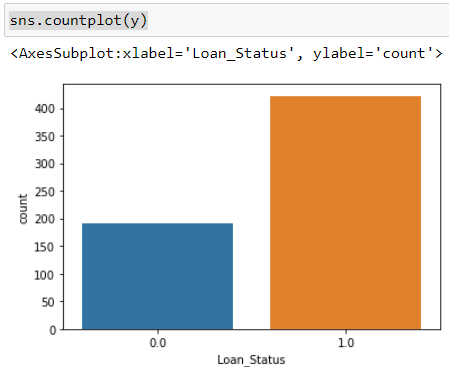
As the range of data is varying we can use Min-Max scaler to limit the range of data between 0 to 1. Transform **features by scaling each feature to a given range**. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one.



Min-max normalization method **guarantees all features will have the exact same scale**

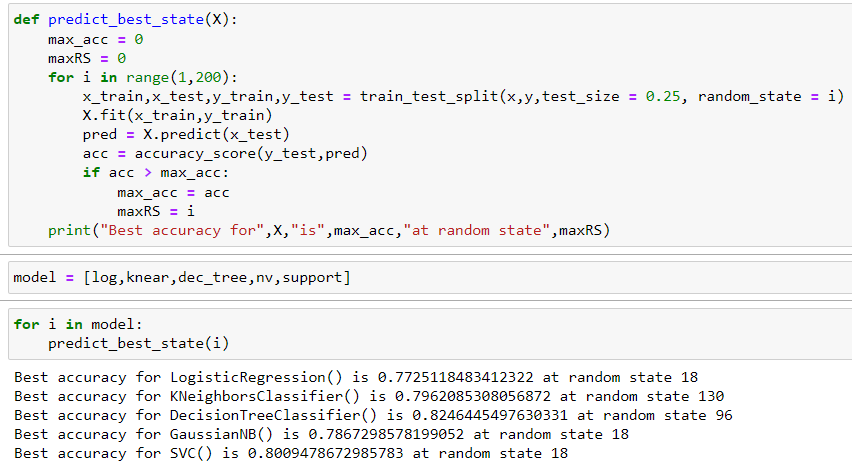
**Balancing the Classes:**

**As, we have checked earlier that our dataset is having an issue of classes unbalancing. We will now use SMOTE technique to balance the classes.**

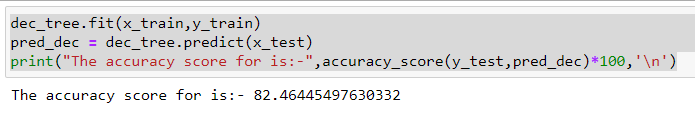
**** ****

**MODEL TRAINING:**

Here the data was fetched into the machine for making predictions. We first finding best random state for the algorithms.



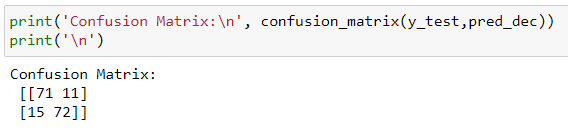
We can check that we are getting 82% accuracy with decision tree at random state 96. So, will proceed with decision tree. Below we can check the results for decision Tree.



Here we applied the random state that we got above and try to run the model and we can check that we are getting 82% for decision tree algorithm.

**Confusion Matrix:**

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

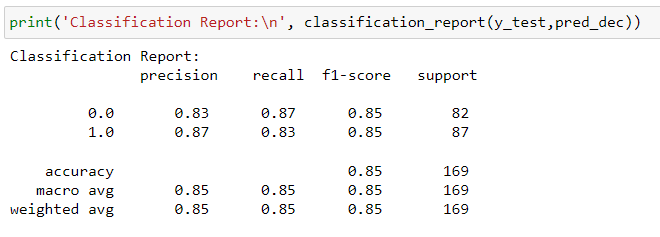
****

**We can check that: -**

1. **We have 71 True positive which means, 71** predicted value matches the actual value**.**
2. 11 predicated values were falsely predicted. The actual value was negative but the model predicted a positive value
3. 15 predicted values were falsely predicted. The actual value was positive but the model predicted a negative value
4. 72 predicted values matches the actual value. The actual value was negative and the model predicted a negative value

**Classification Report:**

A classification report is **a performance evaluation metric in machine learning**. It is used to show the precision, recall, F1 Score, and support of your trained classification model.

****

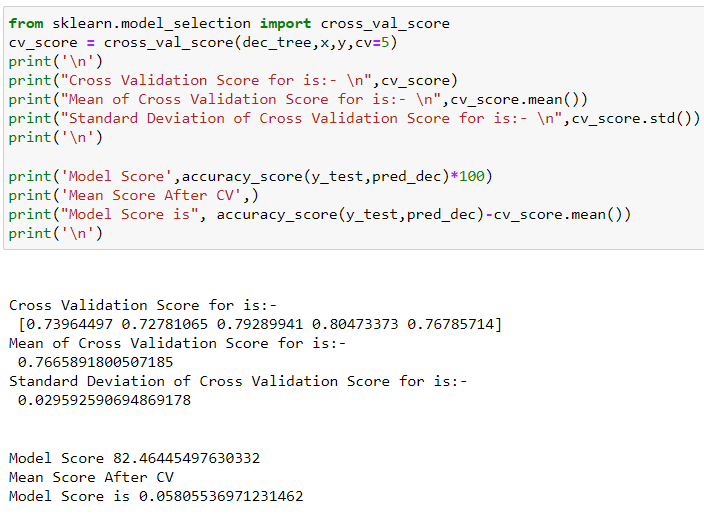
**From the classification report we can say that:-**

1. **We are having 83% precision for predicting 0 and 87% for predicting 1.**
2. **Weighted harmonic mean score is 85%.**

**Cross Validation for Decision Tree Classifier:**

Cross-validation is a statistical method used to estimate the skill of machine learning models.

It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.



Here, we calculated the CV score for the decision tree and we can check that the difference between cv score mean is 0.05, which makes decision tree suitable for this problem.

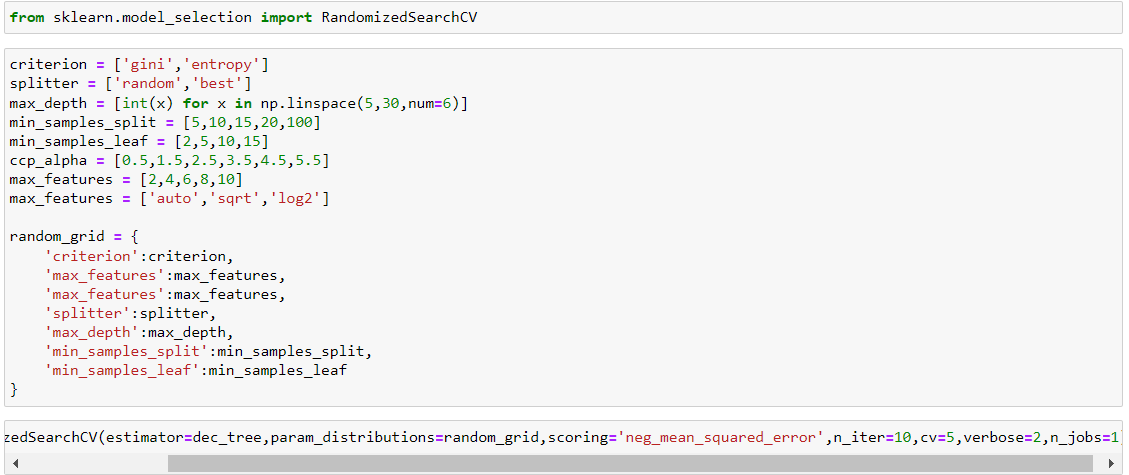
**Observation of the CV Score:**

Minimum difference between Accuracy and Cross validation is of **Decision Tree Classifier** which, will be proven the Best Algorithm for this model.

**Hyper Parameter Tuning:**

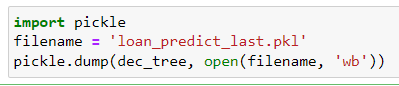
Hyperparameter tuning is **choosing a set of optimal hyperparameters for a learning algorithm**. A hyperparameter is a model argument whose value is set before the learning process begins. The key to machine learning algorithms is hyperparameter tuning.

**We used randomized search CV for improving the performance of our model.**

****

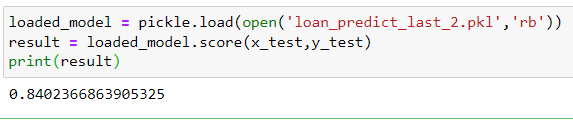
**SAVING THE MODEL:-**

**Hence we found that Decision tree I best suitable algorithm for this model. Hence, we are finally saving this model**

****

**Pickle libraries is used to save the model.**

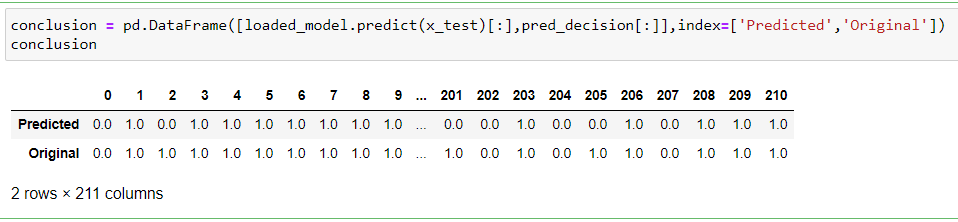
**Loading Model:**

****

**Conclusion:**

**We have successfully created a machine learning model to predict the loan status with an accuracy of 84%.**

**Below we can see that predicted and original values for the model.**

****

**Thanks!**