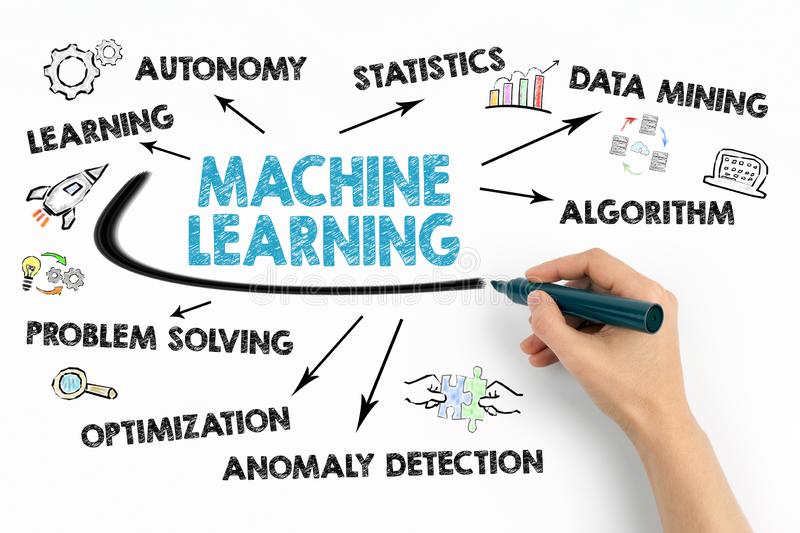
**Loan Application Prediction**

In this Python machine learning project, using the Python libraries scikit-learn, numpy, pandas, Seaborn, Matplotlib we will build a model to predict the approval of loan application. We’ll load the data, get the features and perform feature extractions. We explore few models and calculate the accuracy of the model and fine tuning for prediction.



Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

“Prediction” refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome.

**Prerequisite : python, Basics of Machine learning**

1. **Problem Definition:**

This dataset includes details of applicants who have applied for loan.

The dataset includes details like credit history, loan amount, their income, dependents etc.

**Independent Variables:**

Loan\_ID

Gender

Married

Dependents

Education

Self\_Employed

ApplicantIncome

CoapplicantIncome

Loan\_Amount

Loan\_Amount\_Term

Credit History

Property\_Area

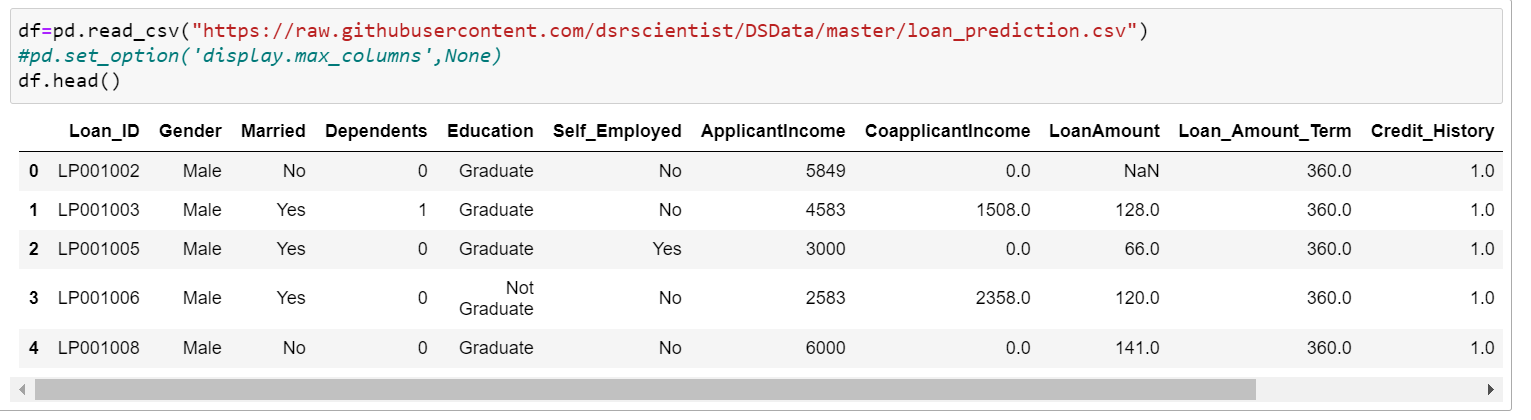
**Dependent Variable (Target Variable):**

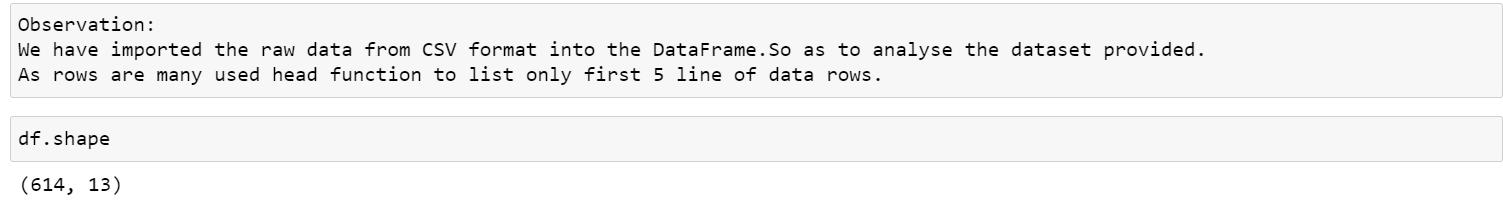
Loan\_Status

Lets build a model that can predict the loan status of the applicant will be approved or not on the basis of the factors provided in the dataset.

1. **Data Analysis:**

Data analysis helps us to know about the given dataset and what to predict. So before getting in to the details, let’s try to understand about data columns to get an idea about the dataset.

****

****

* There are 614 rows and 13 columns. Its comparatively a small dataset.
* Also we can notice that we have Independent columns and dependent columns so it is Supervised learning.
* Target Variable is “Loan\_status” which has to be predicted.
* As we need to classify whether the “Loan\_Status” is Yes or No, the above problem is a classification problem

Now let us look deep into the variables :

Loan ID : As the name suggests each person is having an unique loan ID.

Gender : It states whether the applicant is male or female.

Married : Applicant who is married, is represented by Y and not married as N.

Dependents : This mentions the number of people dependent on the applicant who has taken loan.

Education : This describes about the applicants education status as graduate or non -graduate.

Self\_Employed : As the name suggests Self Employed means , applicant is employed for himself/herself only. An applicant who is self employed is represented by Y and the one who is not is represented by N.

Applicant Income : Applicant Income suggests the income of the Applicant.

Co Applicant income: This represents the income of co-applicant.

Loan Amount : This amount represents the loan amount in thousands.

Loan\_Amount\_Term : This represents the number of months required to repay the loan.

Credit\_History : A credit history is a record of the repayment of debts. Class value : 1 denotes that the credit history is good and Class value : 0 denotes not good.

Property\_Area : The area where the applicants belong. The three types given are : Urban , Semi Urban and Rural

Loan\_Status: This column explains the applicant is eligible for loan or not. If eligible it represents ‘Y’ and if not eligible it represents ‘N’.

**EDA:**

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It can also help determine if the statistical techniques you are considering for data analysis are appropriate.

The above mentioned column description, gives us some information for Exploratory data analysis:

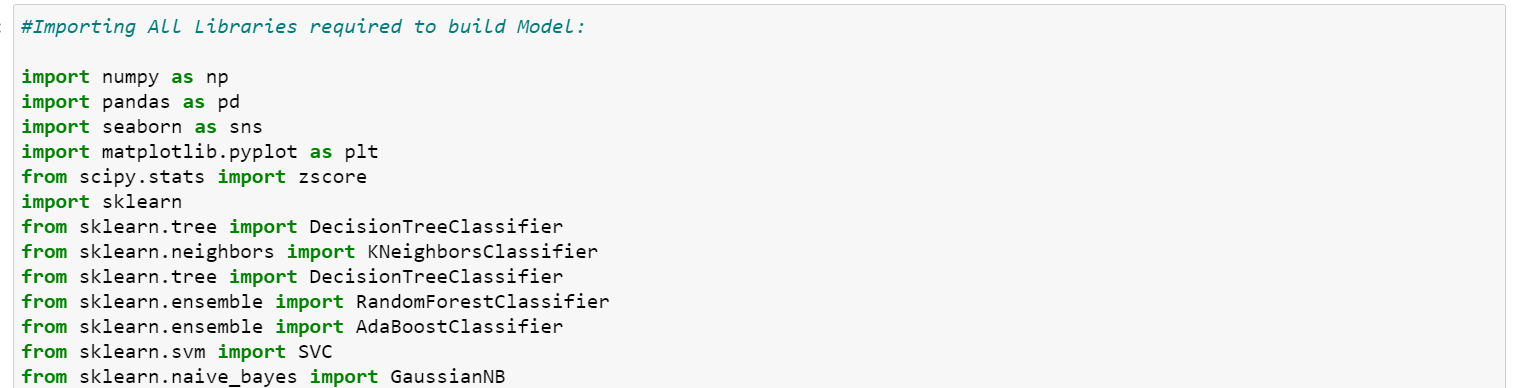
The applicant whose salary is more may have a greater chance of loan approval.

And the applicant who is Graduate may have a better chance of loan approval.

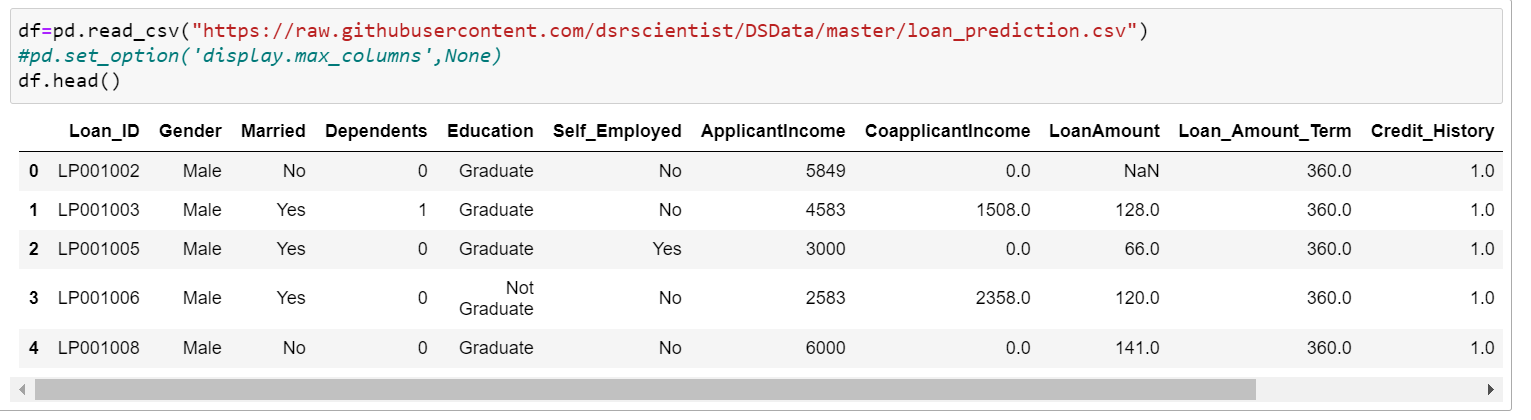
Also lesser the loan amount higher the chance for getting loan.

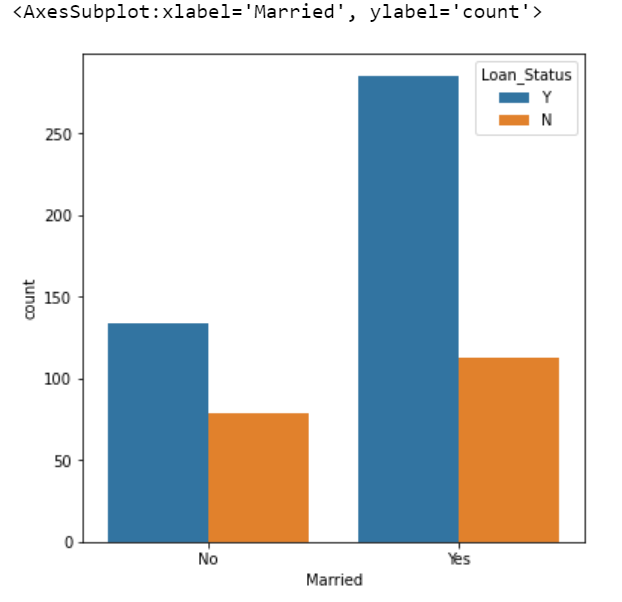
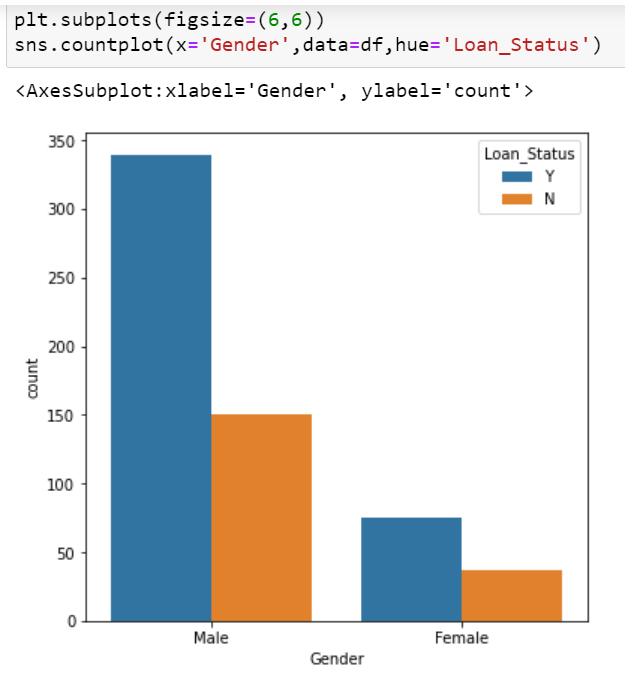
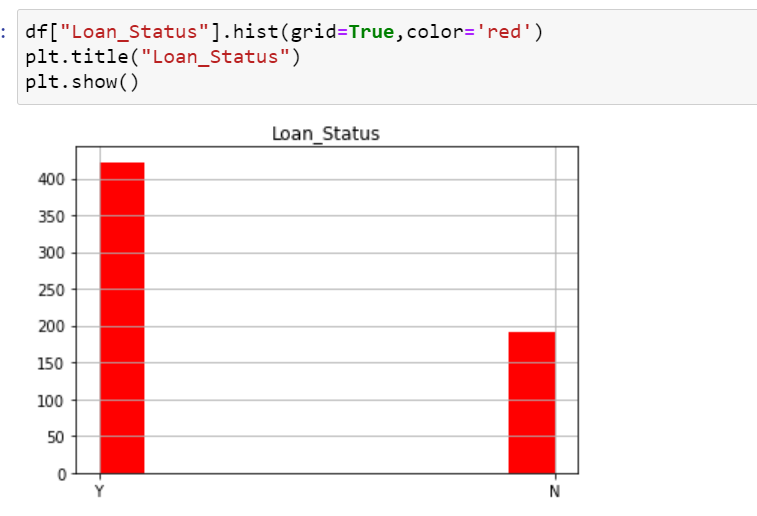
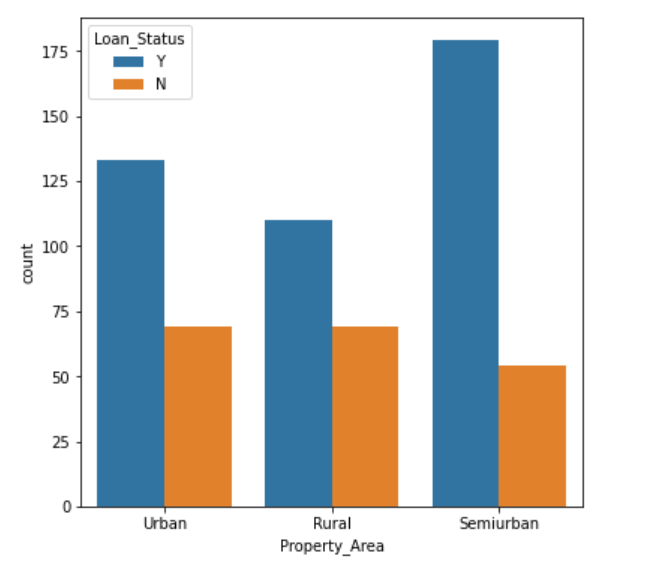
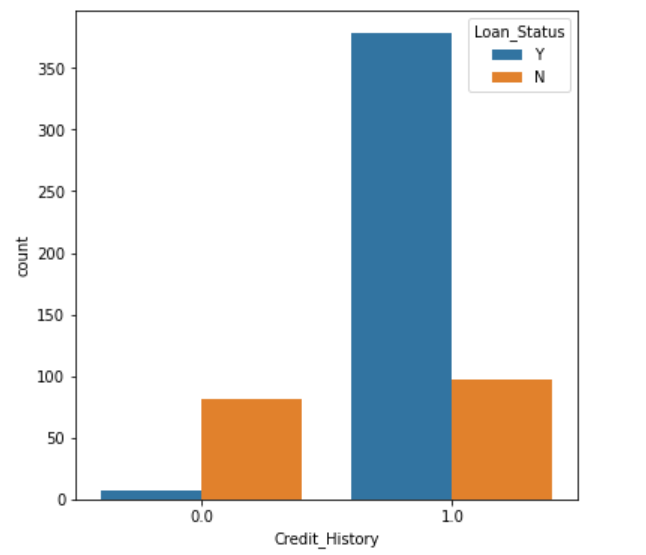
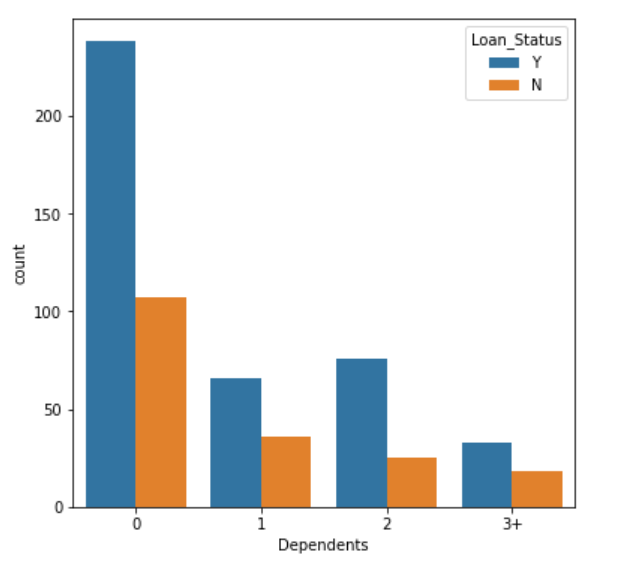
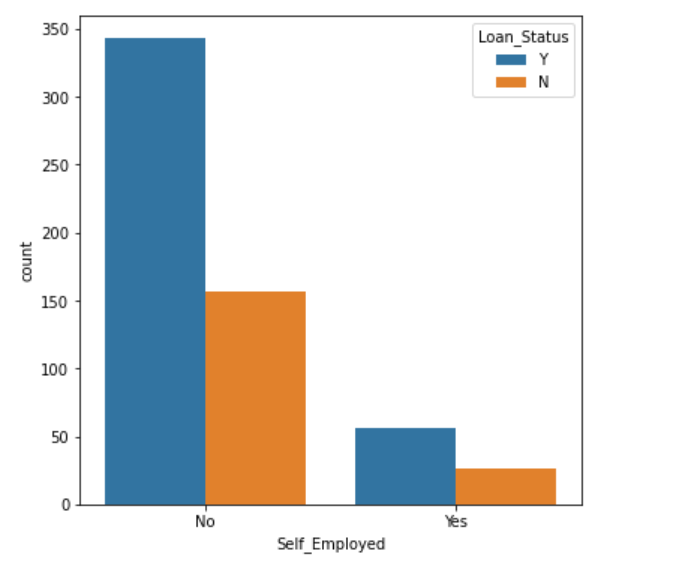
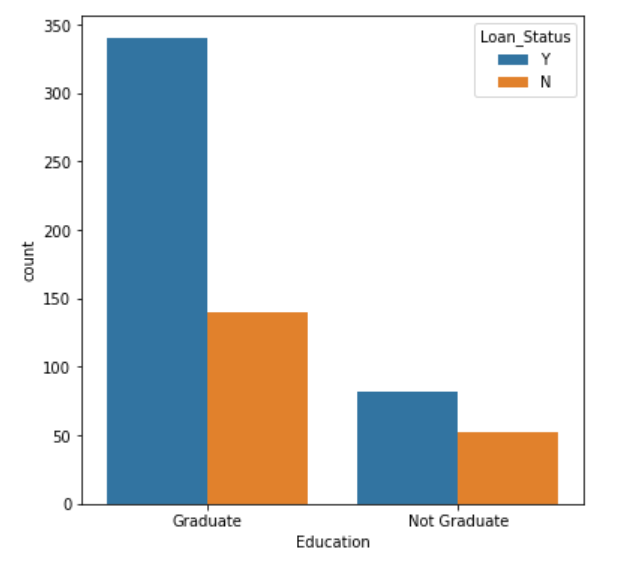
Let’s find out the actual with exact exploration with Python.

Initially we have imported the libraries necessary for further processes of EDA and model building.



Then imported the dataset in DataFrame (df). From CSV format.

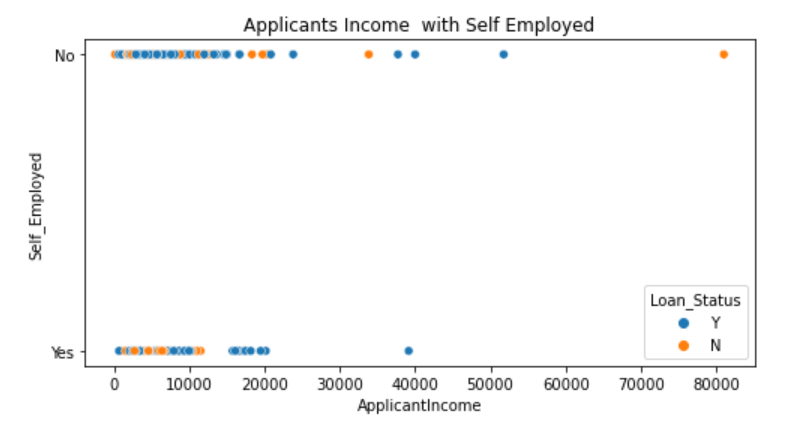
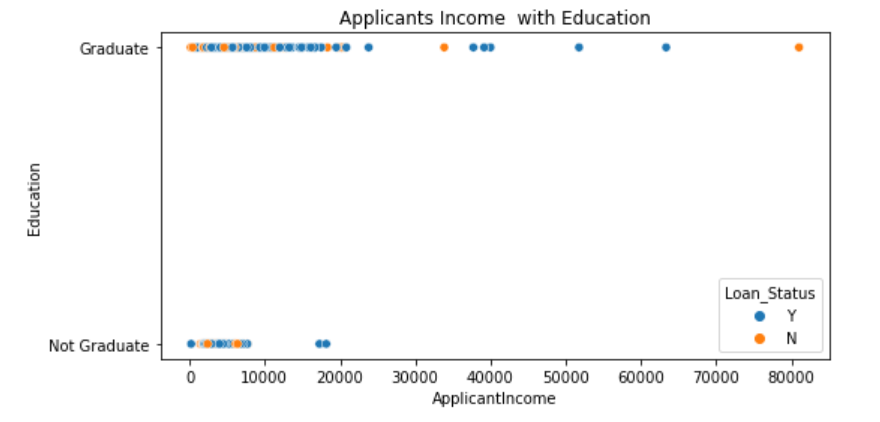
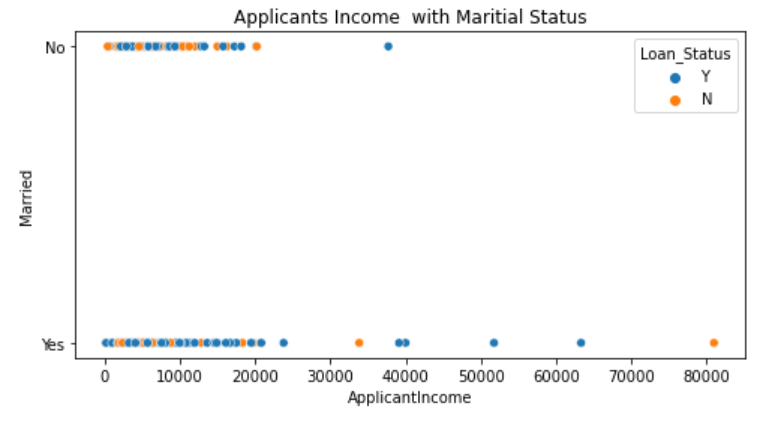
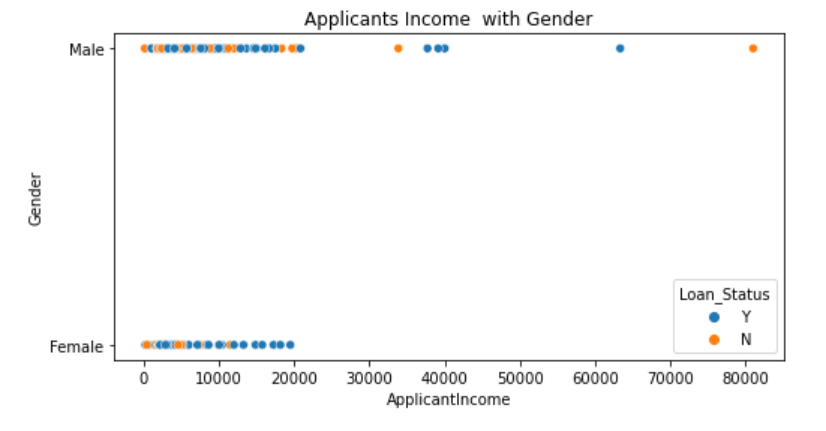
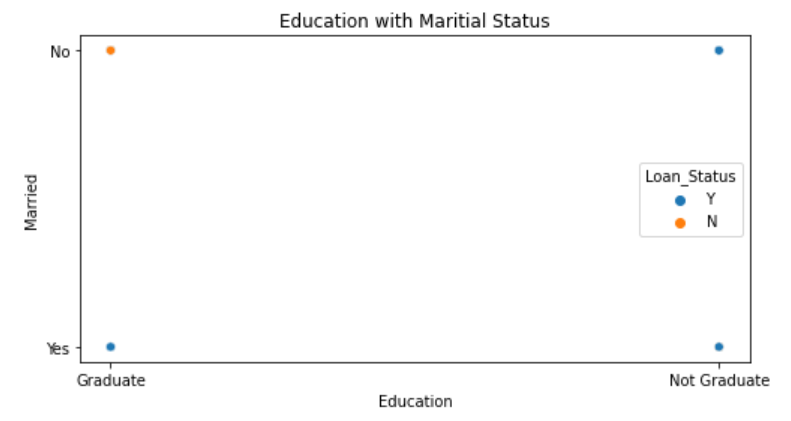
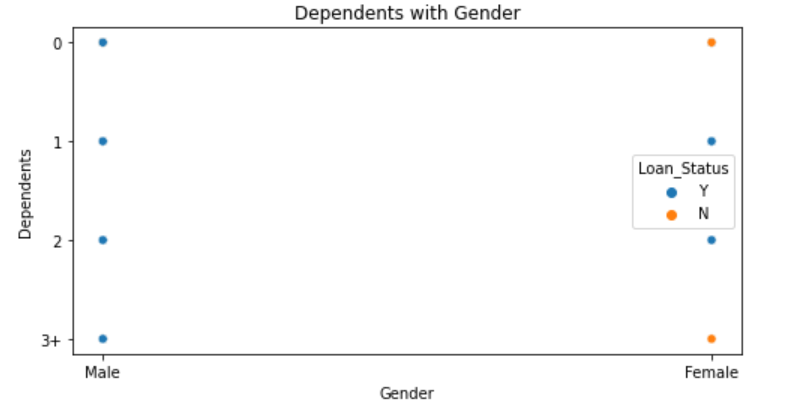
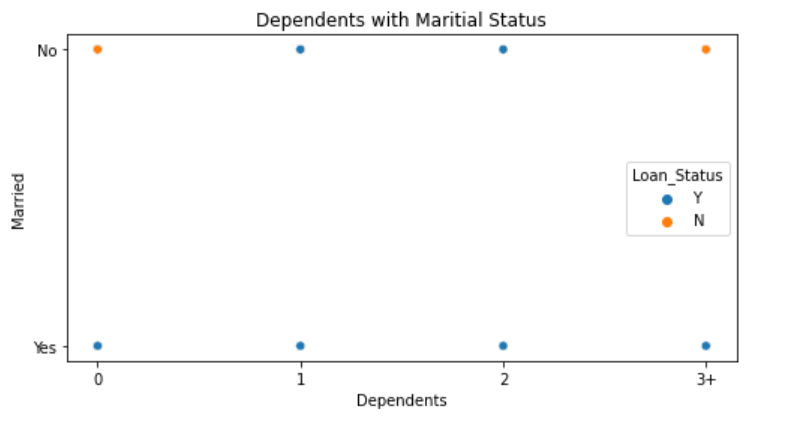
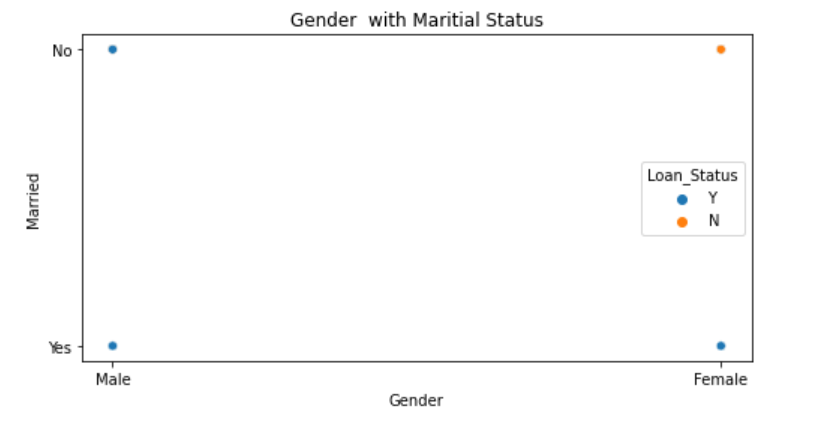


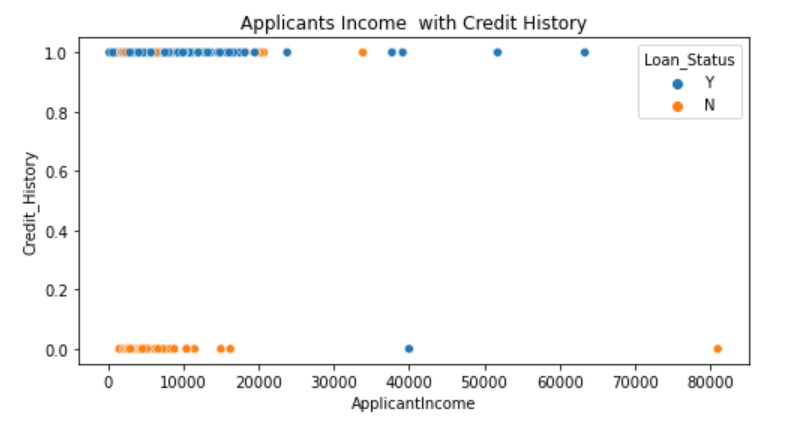
Now we shall Analyse the dataset by **Univariate analysis:**  

**Univariate** involves the **analysis** of a single variable

* The percentage of applicants for whom the loan has been approved has higher percentage rather than the percentage of applicant for whom the loan has been declined.
* We can see that Male ratio count is higher than female count.
* We can also notice that the count of Married applicants have high ratio of loan approval status yes than Unmarried applicants.
* Percentage of applicants with no dependents and next is with 2 dependents is higher.
* There are more number of graduates than non graduates.
* The Count of Self employed is lower than that of who is self employed.
* Larger Percentage of people have a good credit history (1).
* Semi Urban people is slightly higher than Urban people among the applicants.

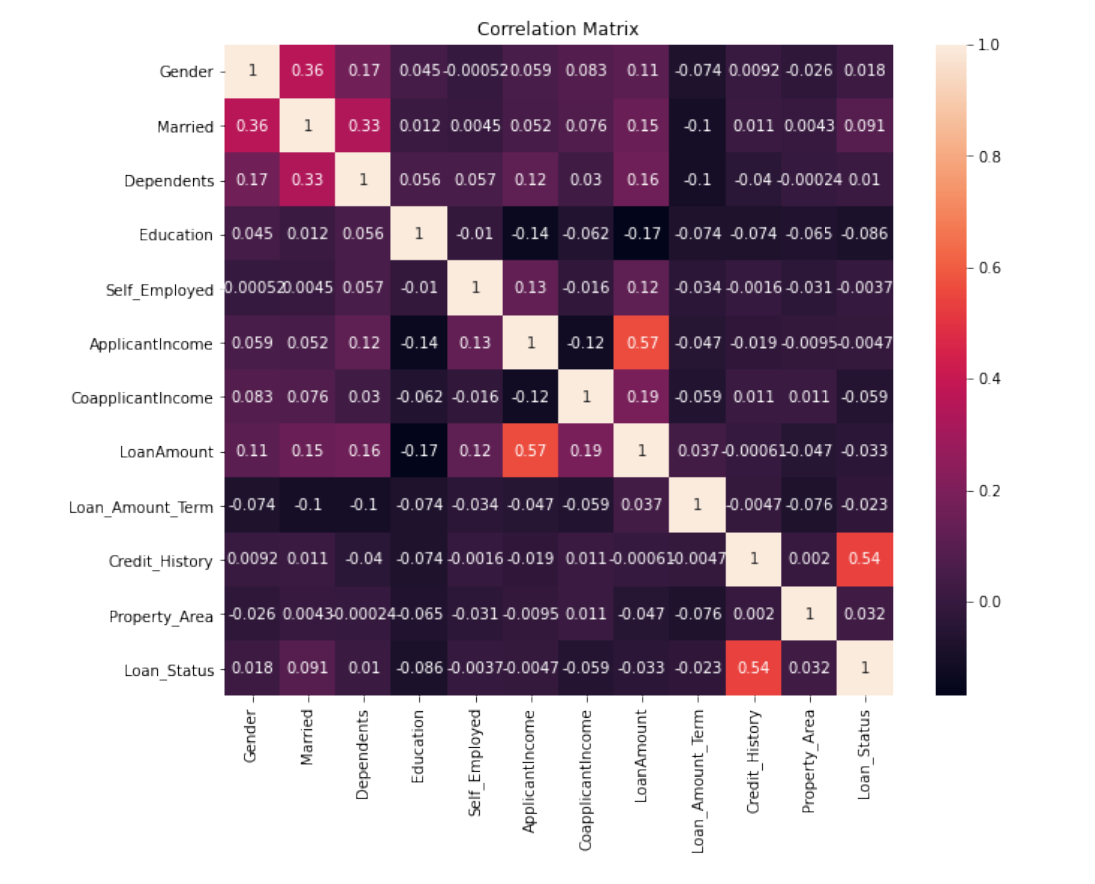
**Multivariate Analysis:**





Below is a correlation matrix:

It’s a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses.

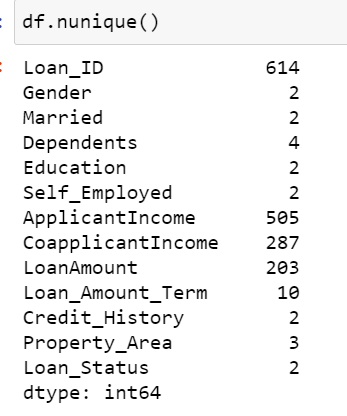
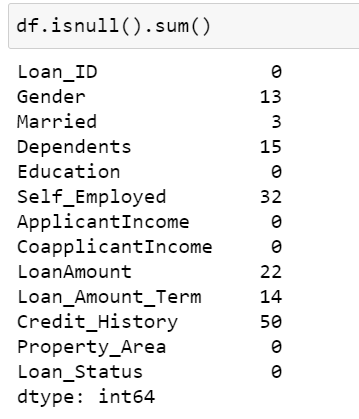


1. **EDA Concluding Remarks:**

* Married male applicants has high probability of loan approval when compared with Females.
* Married applicants who has dependents 2 and no dependents has high probability of loan approval as when compared with those of not married.
* Male applicants has higher percentage of Income.
* Applicants who are married also has an high ratio of loan approval status ‘y’.
* The percentage of applicants who are graduates have got their loan approved rather than the one who are not graduates.
* Applicants with Credit History with zero and income less than 20000 seems to have no approval for loan status where as applicants with Credit history of 1 has high possibilities of loan approval status.
* Loan Amount and Applicants Income is highly correlated.
* We can notice that Loan\_Status is highlt correlated with Cerdit\_History.
* So it says that Loan\_status is highly dependent on Credit History.
* Married , Gender , Dependents are all inter correlated with each other.
* Loan amount is also little correlated with CoapplicantsIncome, Dependents, Married, gender columns.

1. **Pre-Processing Pipeline :**

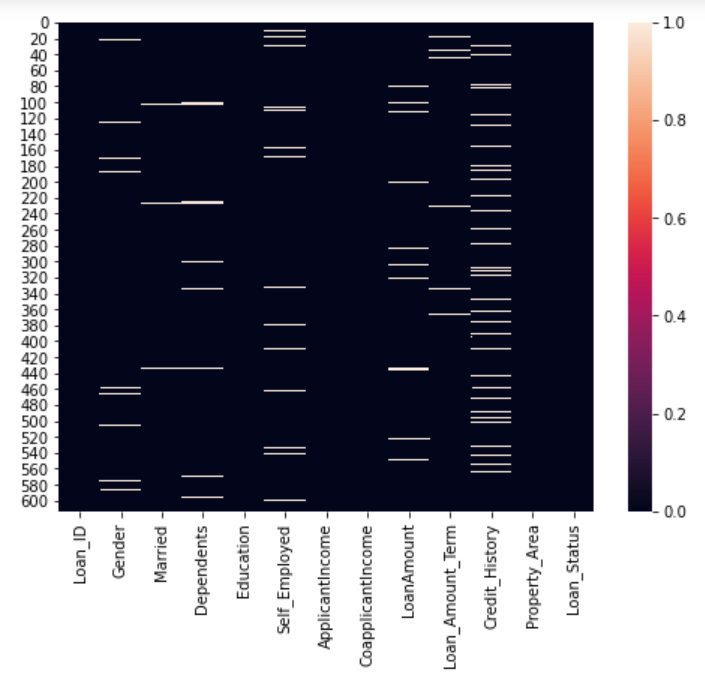
The main objective of the preprocessing task in dataset is to clean the data.

Let’s check the missing values using Heat map graphical representation using following code:

plt.figure(figsize=[8,6])

sns.heatmap(df.isnull())



We have represented the null values using an Heat map.

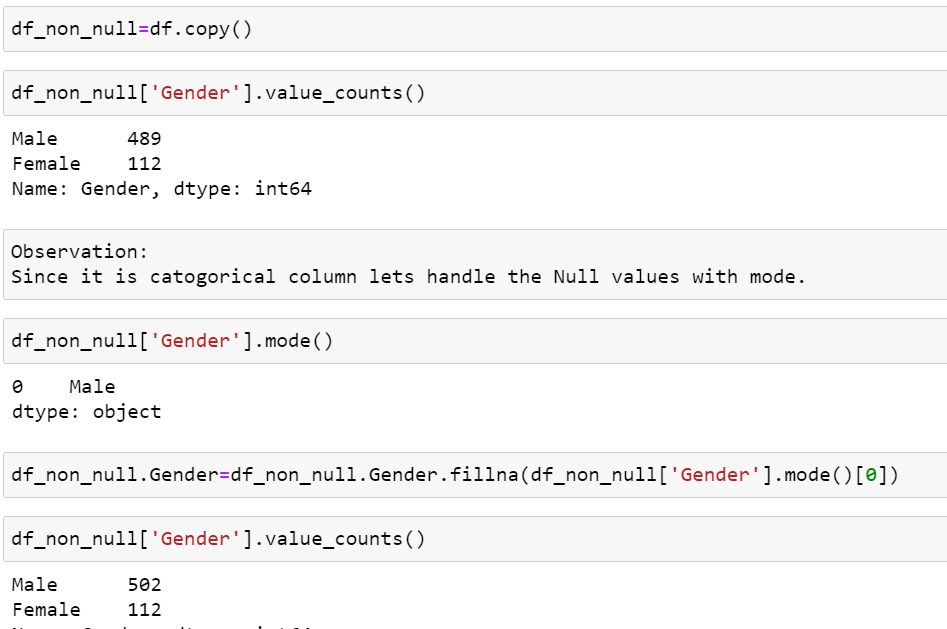
We can notice that we have Null values in many columns.

Null values are on both type of columns i.e on Categorical and Numerical.

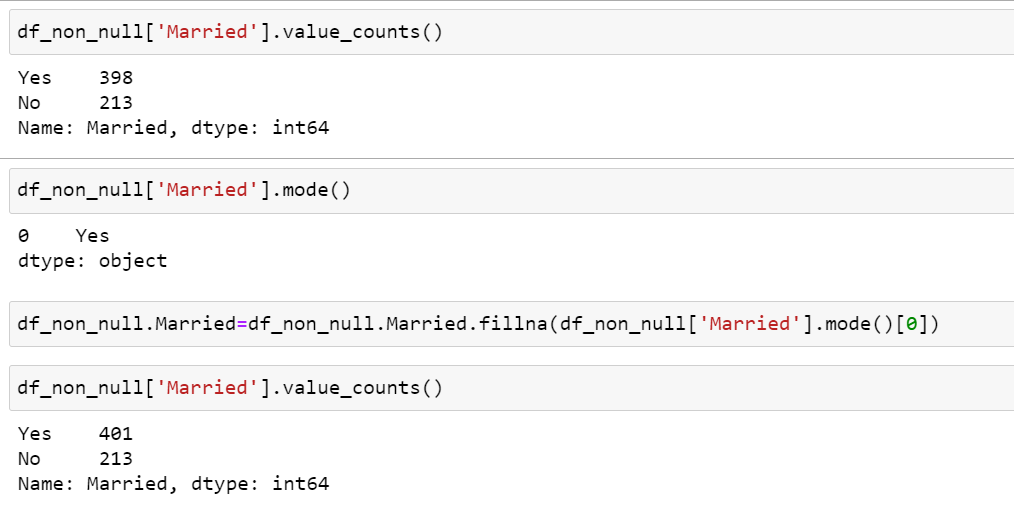
We can treat it based on Mean, Median, Mode depending upon the type and data of column.

* Mean is nothing but the average value in that column.
* where as median is nothing but the central value.
* and mode the most occurring value.

**Gender:** Its Catogorical column, so better to handle the null values with the mode as it make some sense. (In below code we can see the change in count of Male 489 to 502)



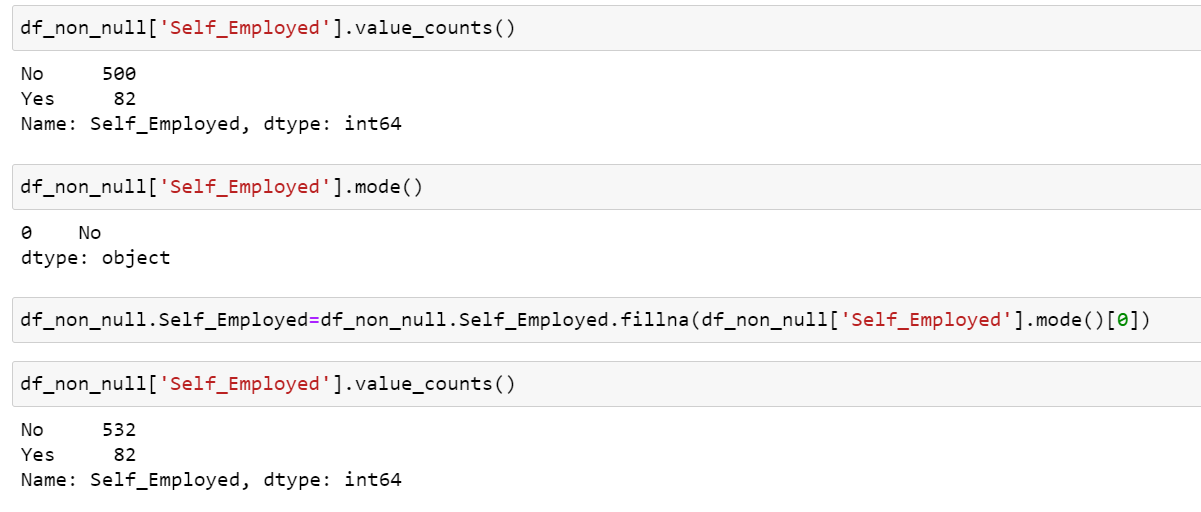
**Married:** Now null value is replaced with mode value (Yes ) is seen before (398-Yes) and after (401-Yes). 3 null values replaced.

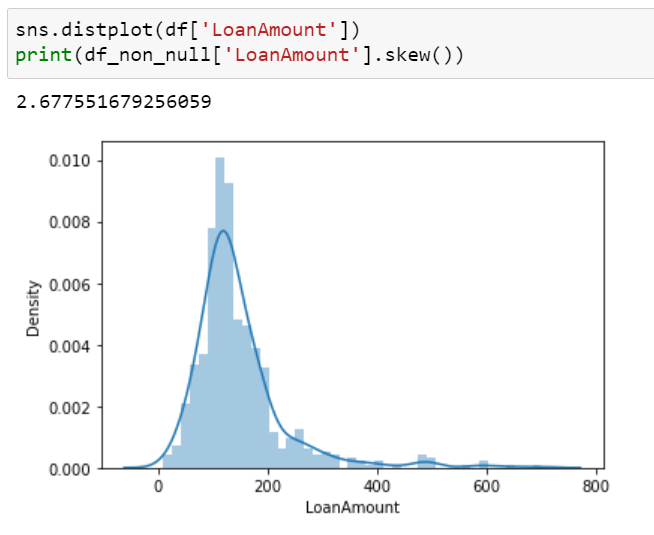
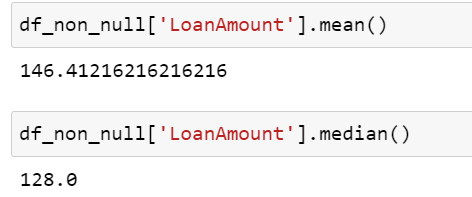


**Self\_Employed:** Same as above this column is Categorical.

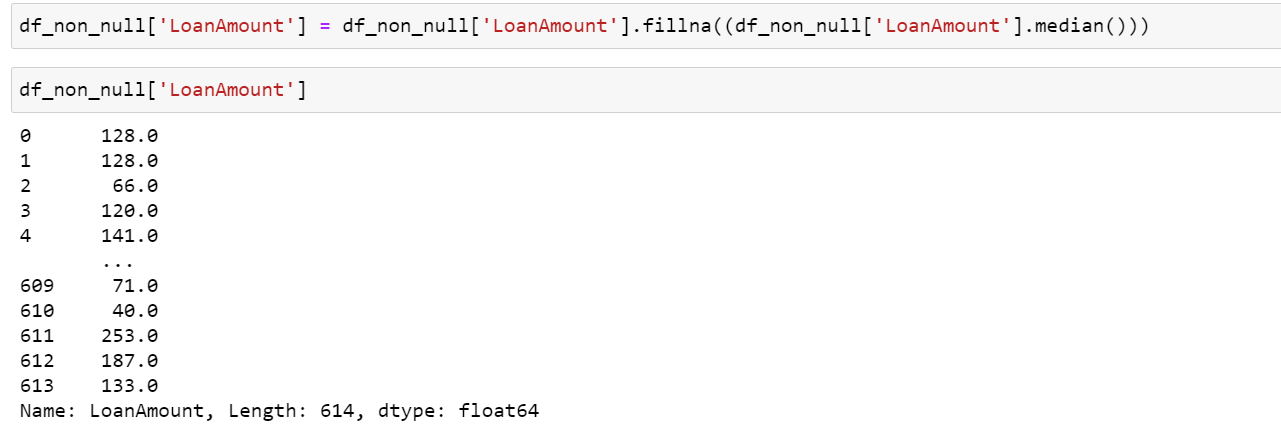
Now null value is replaced with mode value (No ) is seen before (500-No) and after (532-No).

32 null values replaced.

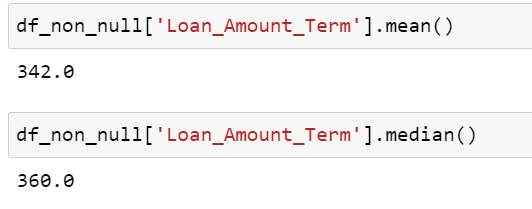


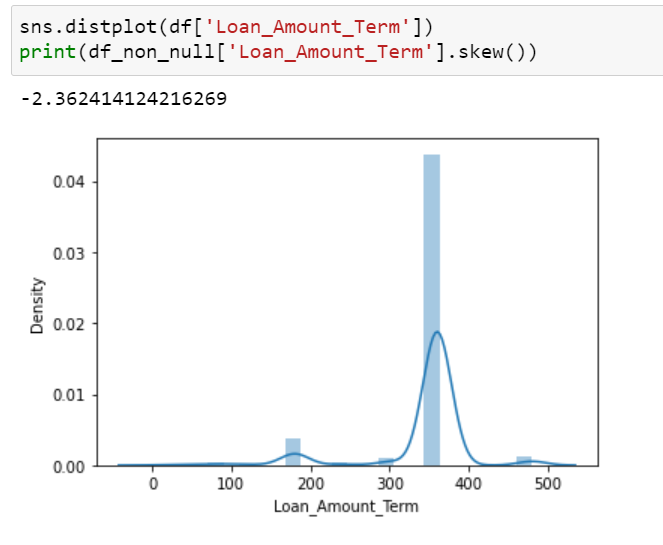
**LoanAmount:** Its an continuous and numerical column. So lets handle with mean or median.

Skewness is there so let proceed with 'Median'. If it is normal distributed we can use 'mean' option.



**Loan\_Amount\_Term:**  This is an continuous and numerical column. So let’s handle with mean or median.



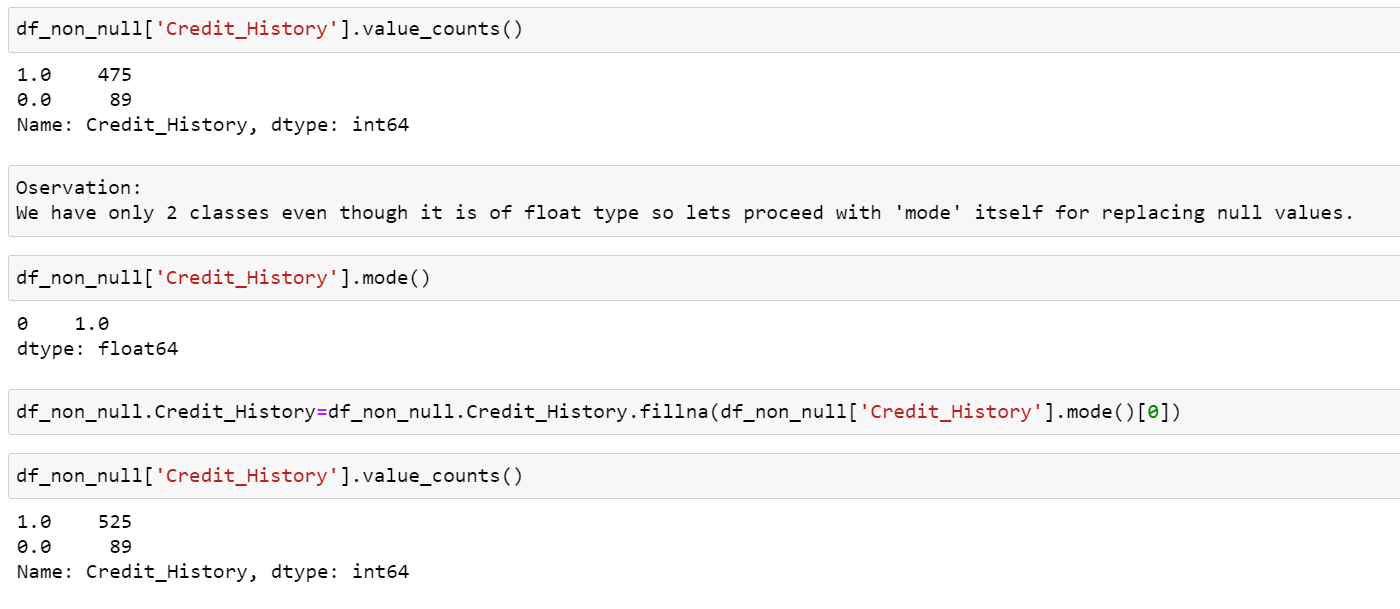


Skewness is there so let proceed replacing the null values with 'Median'.



**Credit\_History:** Now null value is replaced with mode value (1 ) .Before it was seen as (475 counts) and after replacing null with mode it shows(525 counts).

32 null values replaced.



**Dependents:** null value is replaced with mode value (0 ).

We can observe that before it was (345 counts) and after replacing its(360).

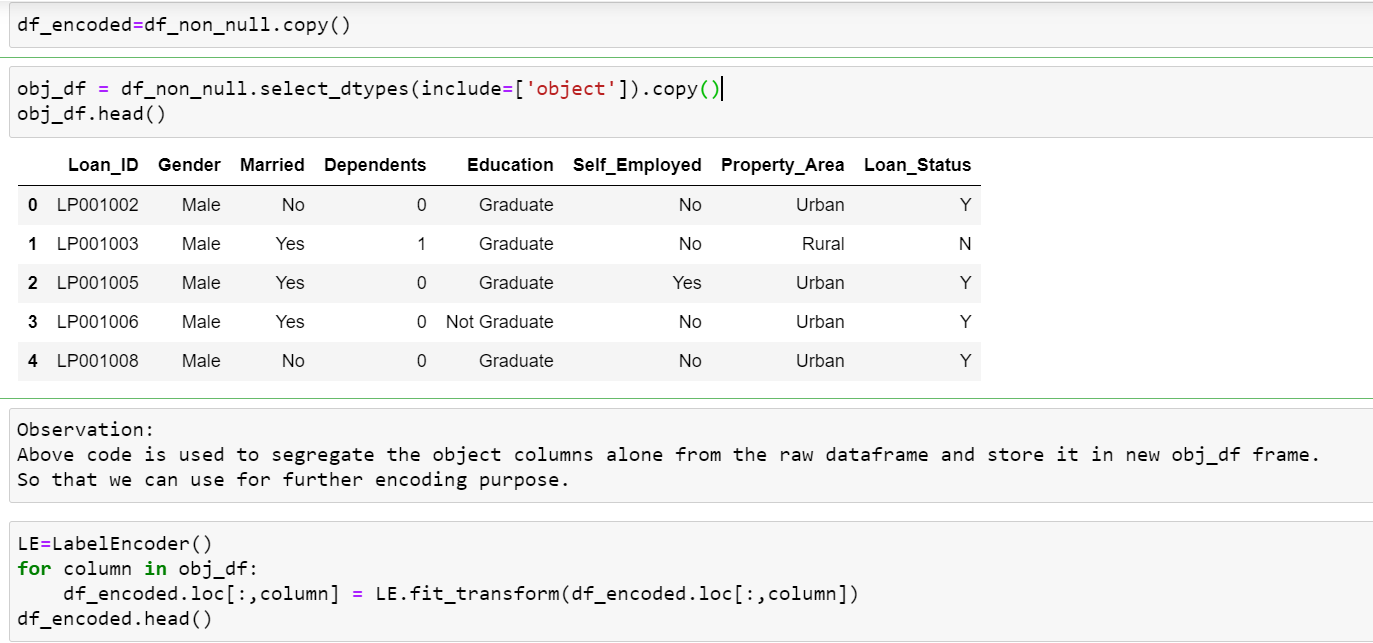
15 null values replaced.

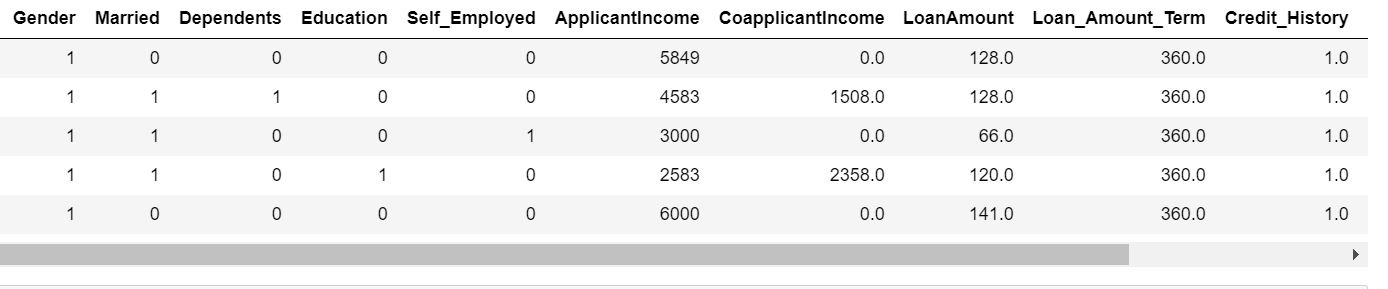


**Label Encoding:** As we are having a lot of categorical variables that are affecting Loan Status. We need to convert each of them in to numeric data for modeling.

Below code is used to segregate the object columns alone from the raw dataframe dataset and store it in new obj\_df frame.

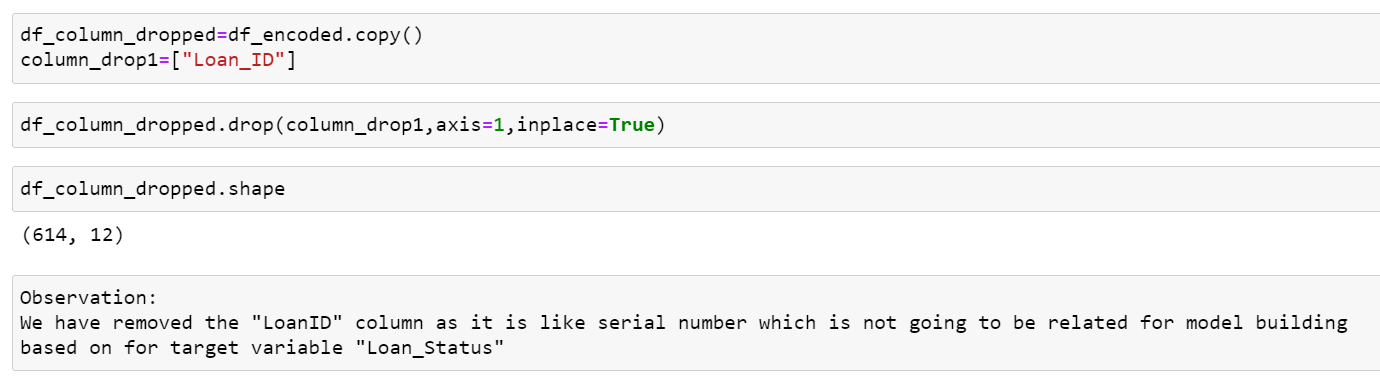
So that we can use for further encoding purpose.

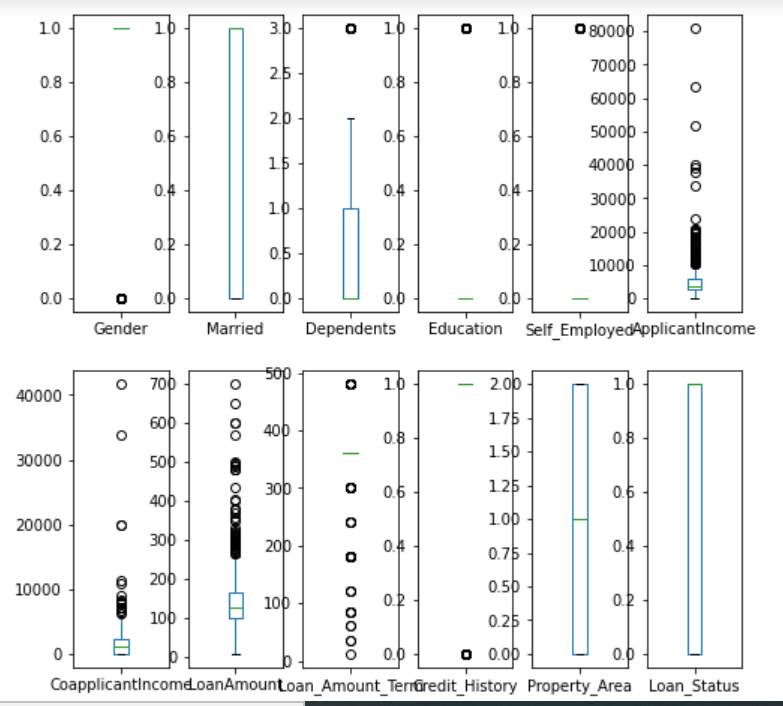




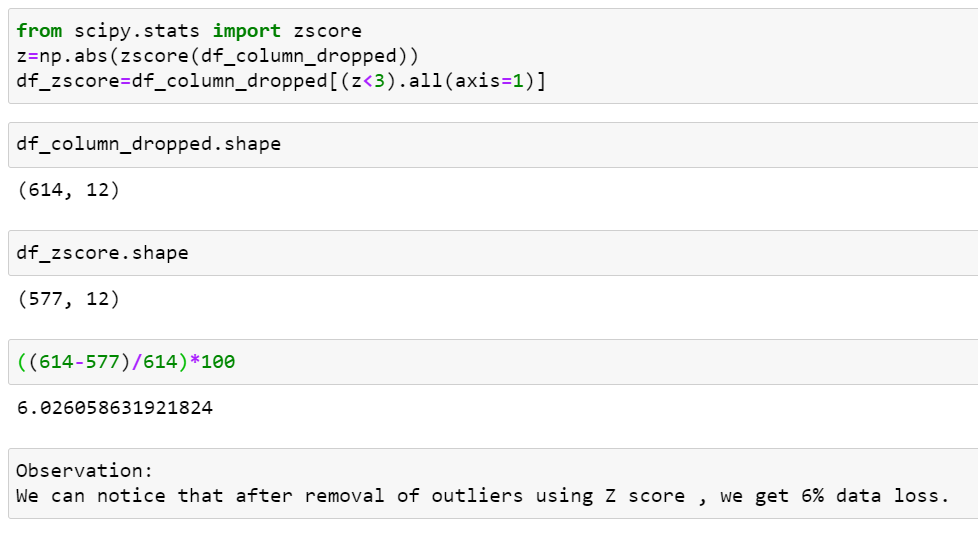
We can notice that label encoding have been done and column values have been converted into numerical representation.

**Dropping column:** We have removed the "LoanID" column as it is like serial number which is not going to be related for model building based on target variable "Loan\_Status".

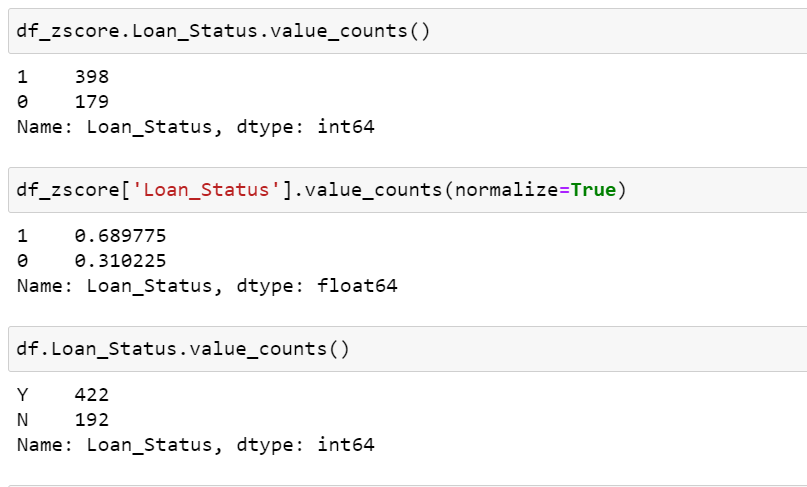


Outliers: We can notice that there is outliers with some columns like LoanAmount, ApplicantsIncome, Coapplicantsinncome

Lets try to reduce the outliers using zscore method.



Sampling:

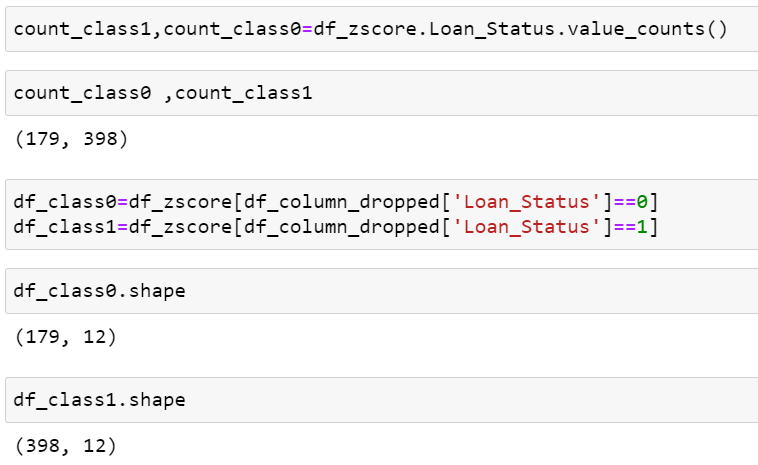


As we mentioned in start we have imbalenced data. So we nee to balance it before applying in model building.

Or else we will not get good accuracy.

So let’s proceed with Over-Sampling as it will not cut down any data or data loss will not be there.

So let do sampling with Over-Sampling for the lower count class in Target variable.

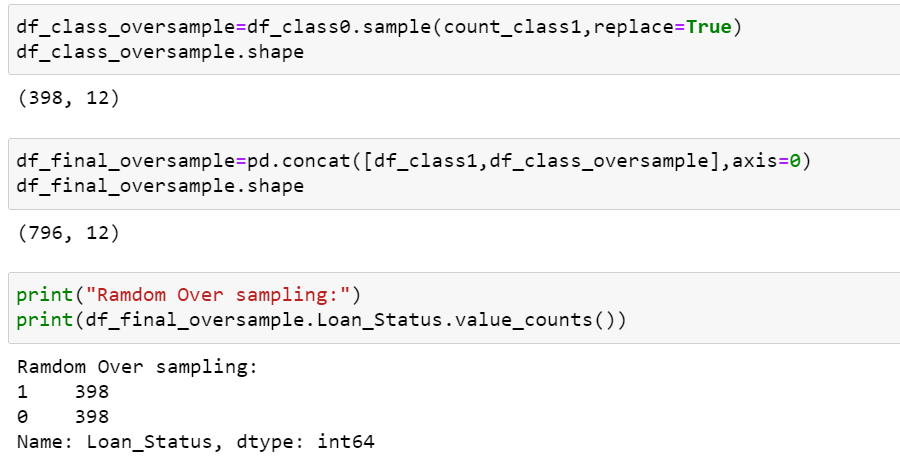


We have split the Target variable value counts into 2 different classes:

count\_class0,

count\_class1,

and 2 dataframes

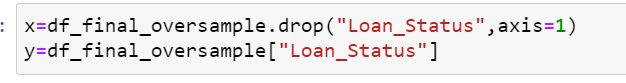


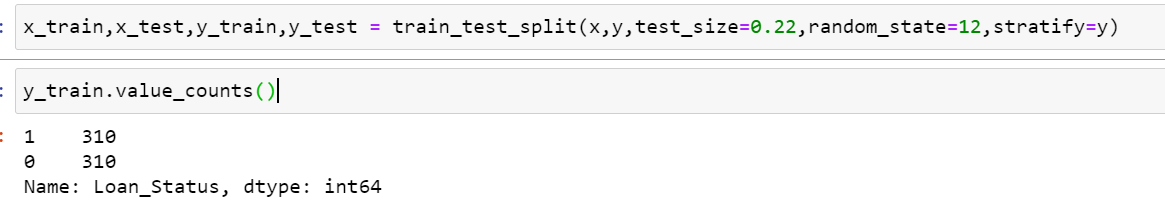
We can see that the two types of classes which are balance in the Target Variable.

**5. Building Machine Learning Models:**

Lets split the variables into x and y with independent and dependent variables respectively.

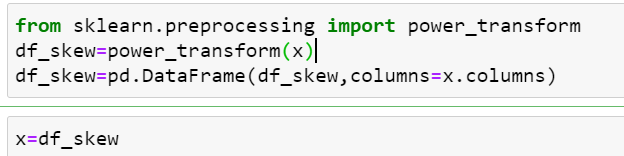
so that we can standardize and pass it in models for accuracy checking and model selection.





We have checked for the y\_train also coming in a balanced state. So lets proceed further with modeling.

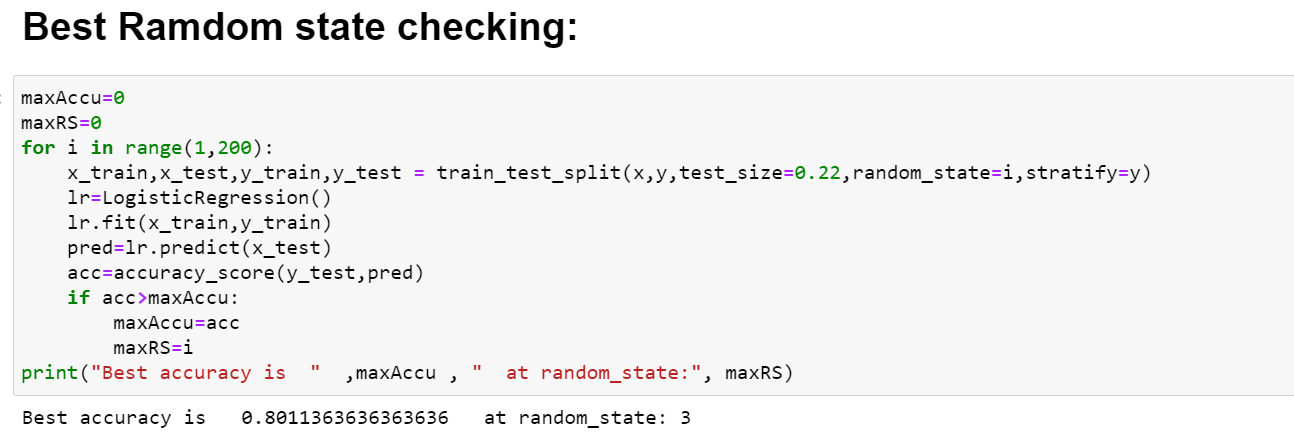
**Skewness removal:** Let’s use power transform method to remove skewness.



**Best random\_state checking:**

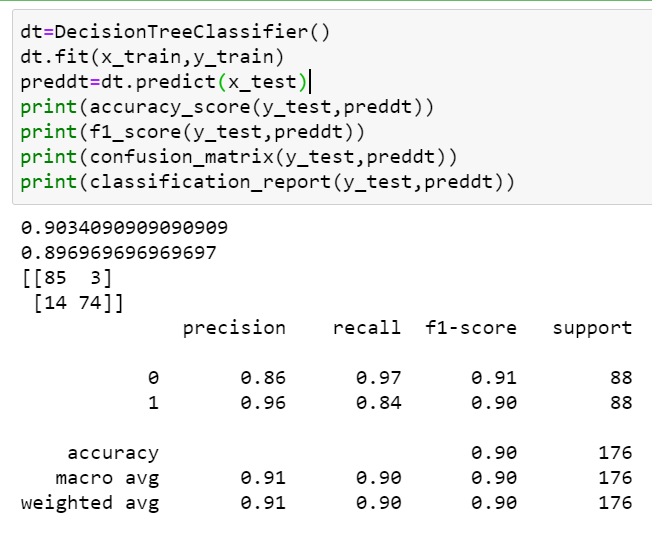
Use any one model to find best random\_state or else lot of computational time consumed.

We have checked the best random state with the below code block so that we can use that in further steps in applying in models.

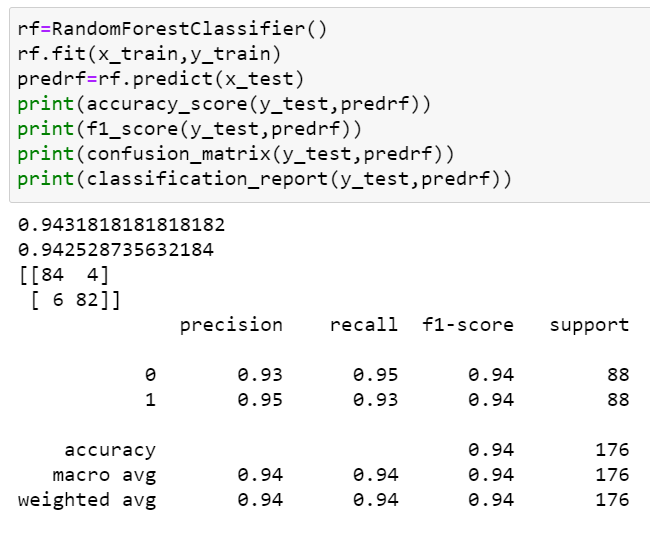


Lets check with 3 to 4 models for checking the best score from which models.

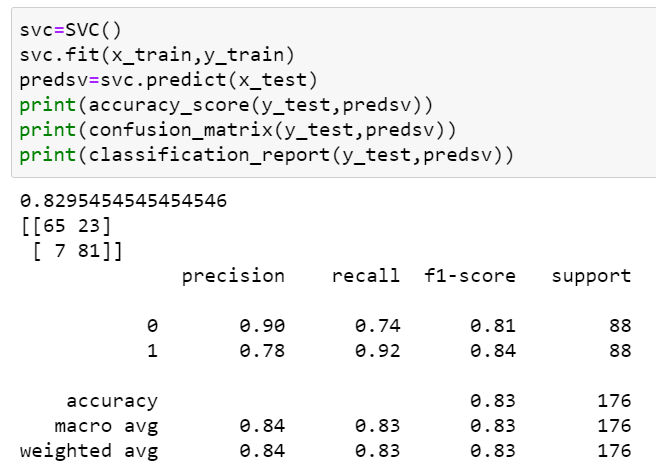
1.DecisionTreeClassifier() ,



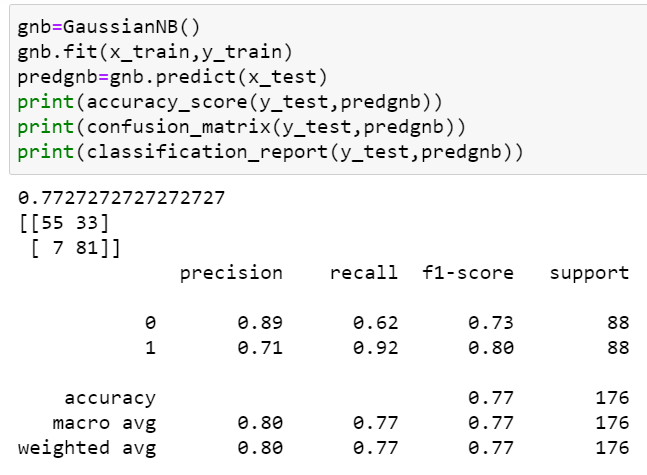
2. RandomForestClassifier(),



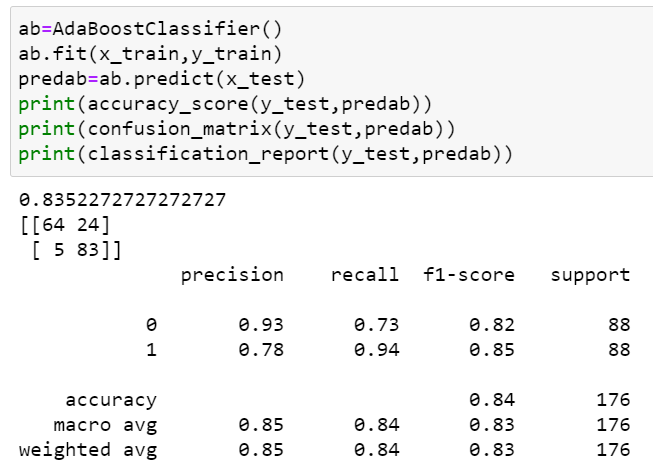
3.SVC() ,



4.GaussianNB(),



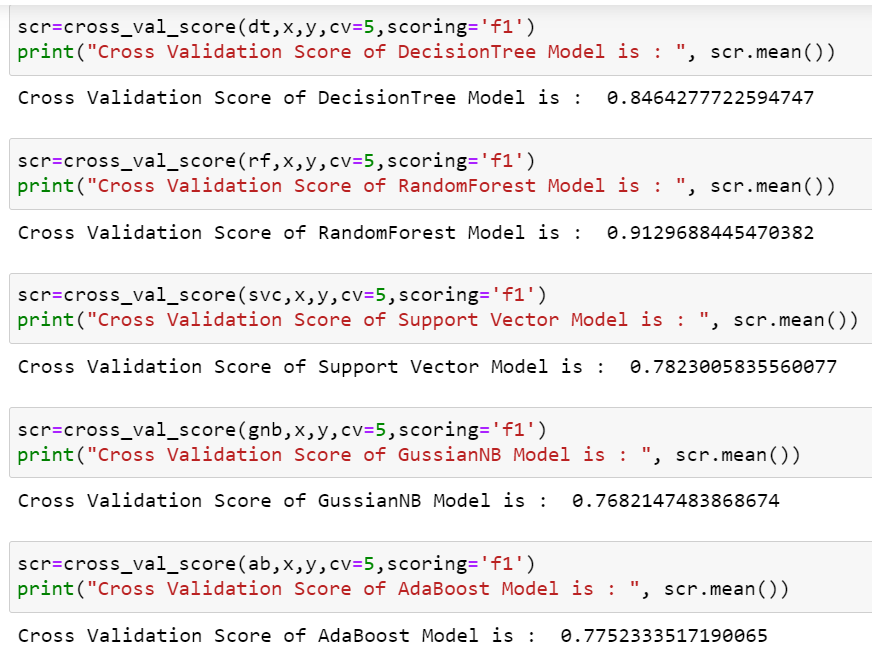
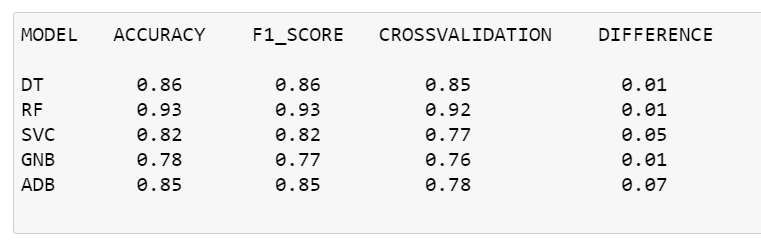
5.AdaBoostClassifier()



 There may be Over fitting or Under fitting with accuracy scores with some models due to bias, so we will check with Cross\_validation scores also.

We could consolidate and finalize the best model with the help of calculating the difference between the accuracy score and Cross validation score of different models and select the best with minimum difference model so that it is efficient and best.

Note: If suppose we have Target variable imbalance we can use F1 score and cross validation score for difference calculation to get best model.

CrossValidation with all Classification models for checking which model is better: 

From the above table we can see the accuracy score,f1 score,crossvalidation scores fo different models.

Also we have take difference of F1 score and cross validation so that we can selet the best performace model.

We use F1 score difference with crossvalidation instead of accuracy score since because its was imbalanced dataset and

we get good with F1 scores.

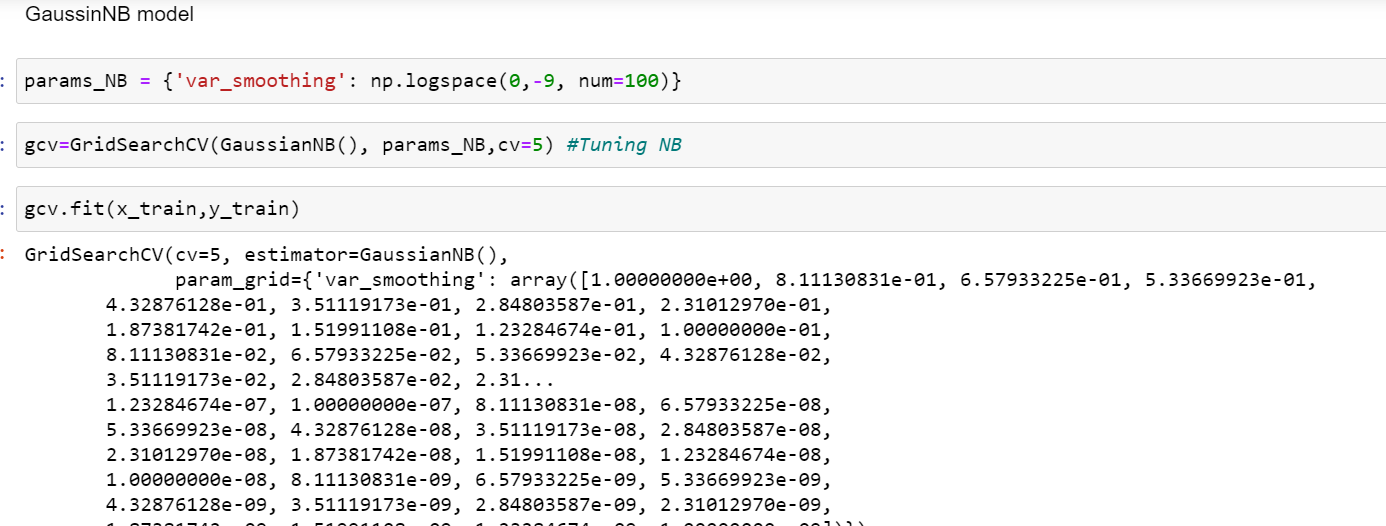
Also we can notice that least diference with more than a model.(RF,DT,GNB).So lets tune futher more with these models and check for performances.

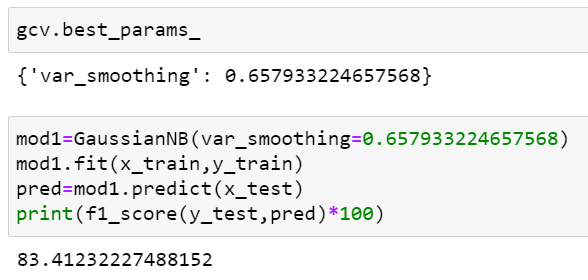
**Hyper Parameter Tunings:**

Tuning DecisionTreeClassifier:



Tuning GaussianNB:



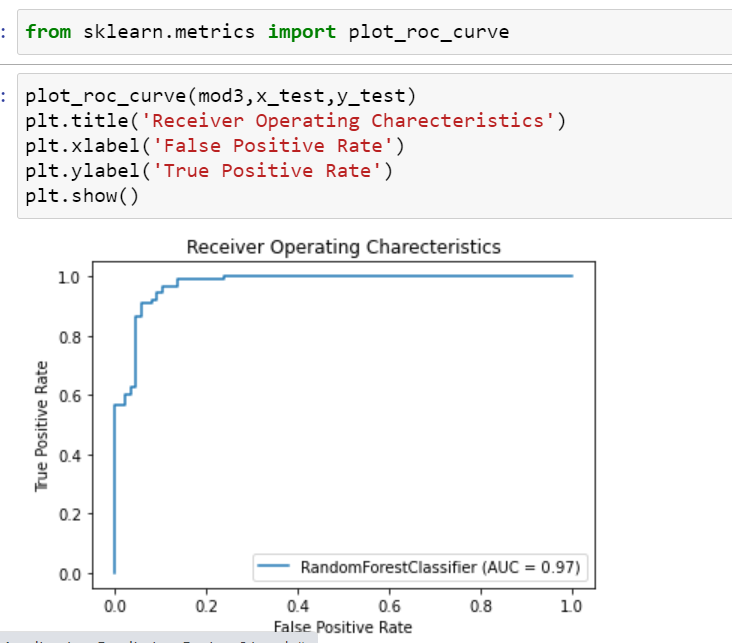


Tuning RandomForestClassifier:



**After parameter tuning ,We have got 92.4% accuracy from Random Forest Classifier model. Hence it is best model to proceed for Prediction.**

### Lets Check with AUC-ROC curve % :

****

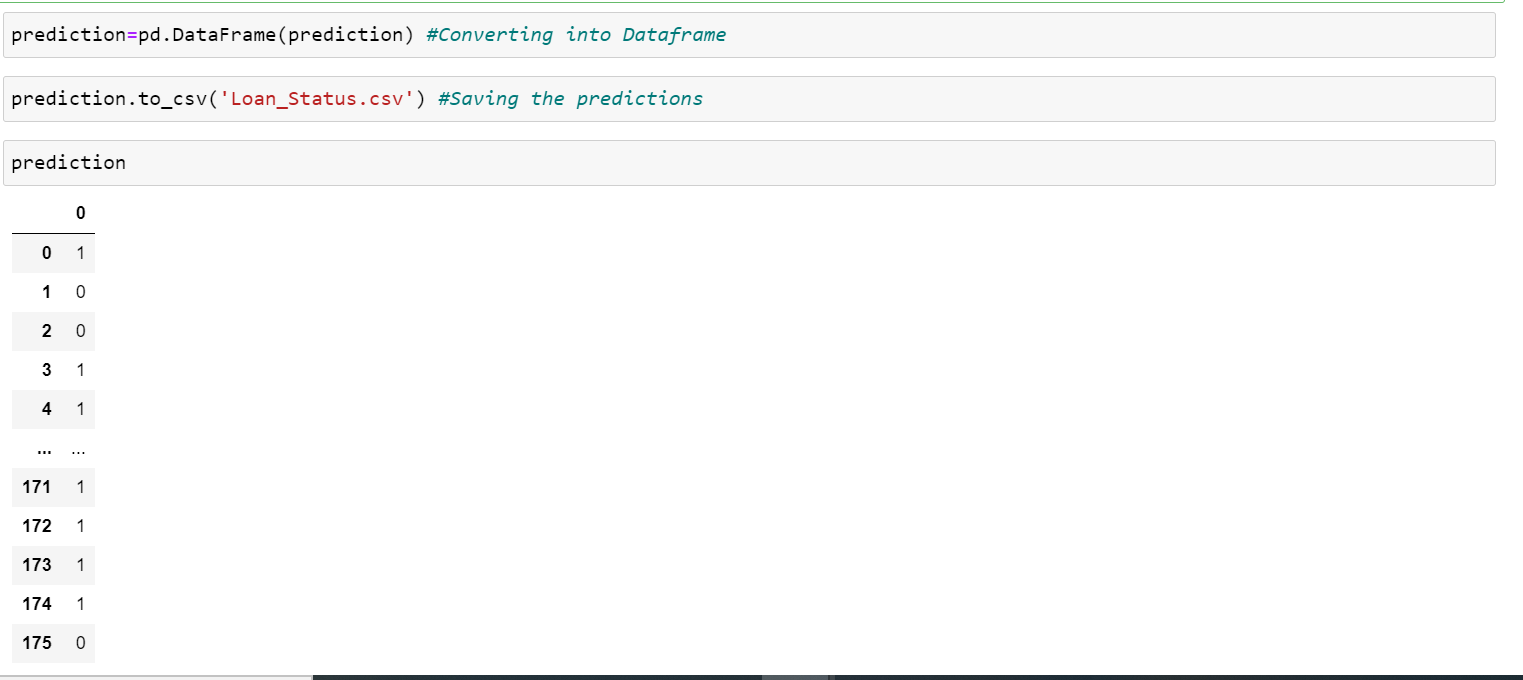
Saving and Loading the Models:



Observation:

We have saved and loaded in above codes and checked for the accuracy too.

Now we can convert that in to DataFrame and Hence saved to CSV format and exported.



**6. Concluding Remarks:**

**After parameter tuning ,We have got 92.4% accuracy from Random Forest Classifier model. Hence it is best model to proceed for Prediction.**

Hence we have saved the model and Checked by loading whether its working fine. and checked with predicting.

In this Machine learning Project,we have learned to identify the status of the loan approval using various factors. We used Random forest Classifier for this and made use of sklearn libraries to prepare the dataset.