**Loan Application Status Prediction**

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from noisy, structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains.



**1. Problem Statement**

This dataset includes details of applicants who have applied for loan. The dataset includes details like credit history, loan amount, their income, dependents etc.

In this Assignment, we are going to solve the Loan Approval Prediction. This is a Binary classification problem in which we need to classify whether the loan will be approved by the bank or not. Classification refers to a predictive modelling problem where a class label is predicted for a given example of input data. A few examples of classification problems are Spam Email detection, Cancer detection, Sentiment Analysis, etc.

**2. Data Analysis**

There are all kind of customer across urban, semi-urban and rural areas. The customer first applies for loan and after that,the company validates the customer eligibility for the loan.

The company wants to automate the loan eligibility process based on customer detail provided while filling out online application forms. To automate this process, they have provided a dataset to identify the customer segments that are eligible for loan amounts so that they can specifically target these customers.

**2.1 Feature Understanding**

* Loan ID : Unique Identification of every person's loan account.
* Gender : Male or Female
* Dependents : No. of Dependents
* Education : Graduate or Non-Graduate
* Self Employed : Weather a person is Self Employed or Salaried
* Applicant Income : Person's own Income.
* C- applicant Income: Income of Co-applicant
* Loan Amount: Total amount applicant owe to the bank.
* Loan Amount Term : Tenure of Loan Amount in 'Months'
* Credit History : Previous Loan History
* Property Area : Urban, Rural, Semiurban
* Loan Status : Yes or No. Weather Loan is approved or not

**2.2 Objective**

We have 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.

**3. Exploratory Data Analysis**

**Importing Libraries**

Here we are importing some necessary libraries we’ll need to import and analyse our dataset.

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler,LabelEncoder

from sklearn.model\_selection import train\_test\_split,cross\_val\_score

import joblib as jl

from joblib import dump,load

import warnings

warnings.filterwarnings("ignore")

#Importing Classification Models

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

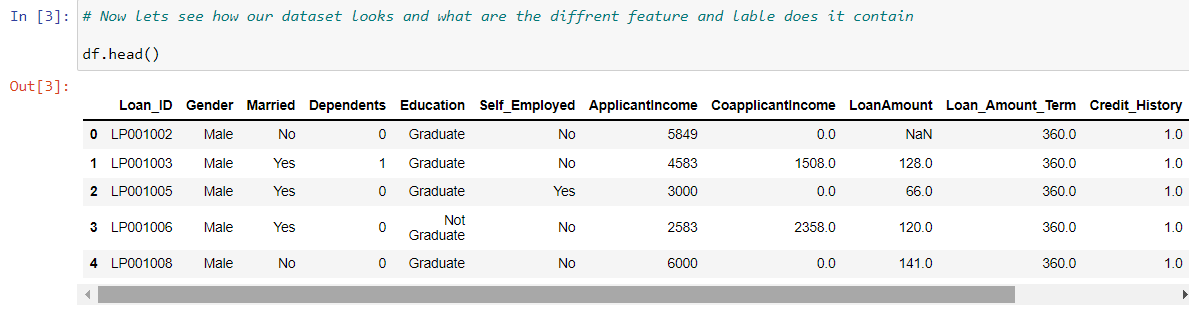
from sklearn.model\_selection import cross\_val\_score

from sklearn.metrics import classification\_report,accuracy\_score,f1\_score,PrecisionRecallDisplay,plot\_confusion\_matrix,precision\_score,recall\_score

**Importing Dataset**

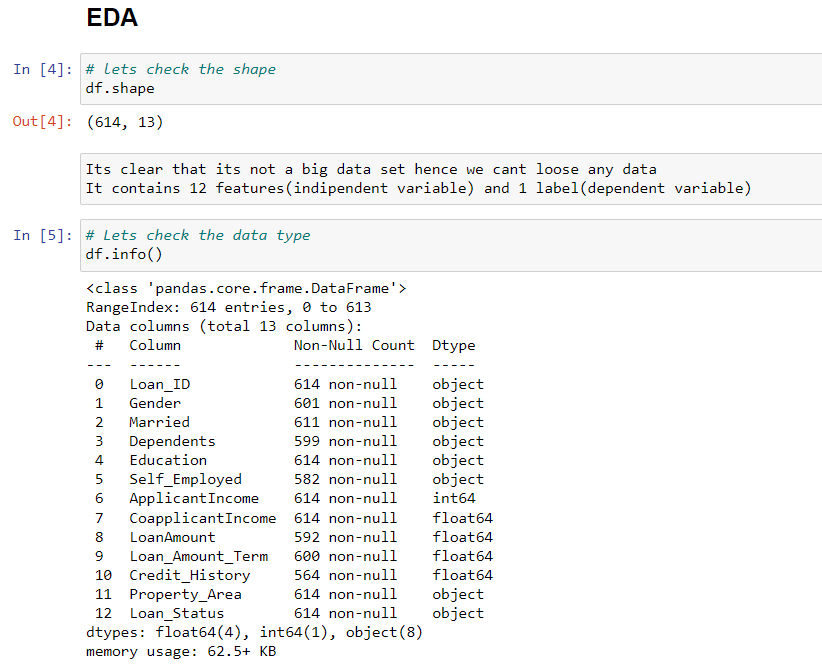
df=pd.read\_csv(r"C:\Users\hp\Dropbox\PC\Desktop\Evaluation Projects\week 2\Loan application status prediction\loan\_prediction.csv")

we have imported our dataset by giving the location at which our dataset is stored. Below is the simple representation how our data looks.

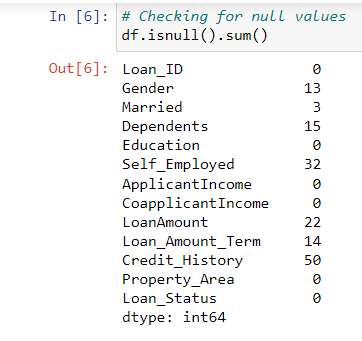


**Analysing Data**

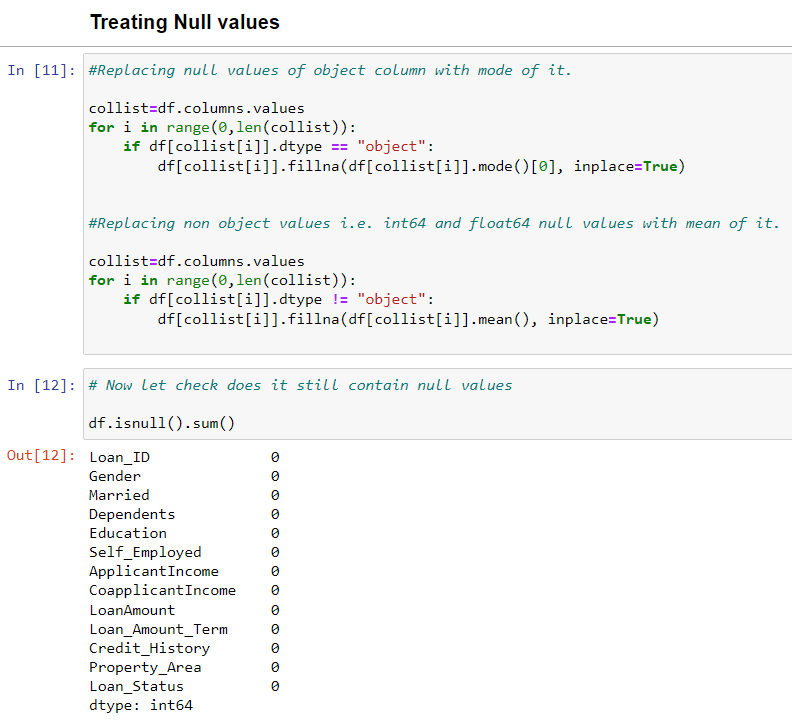
I first check the shape of our Dataset to better understand our dataset and then checked information about the dataset, where I get the better understanding of data type of each column ,and to check our dataset for Null value

****

From df.info I found that my data contain null values which I need to treat,before moving further.let first see the counts of null value different column contain.

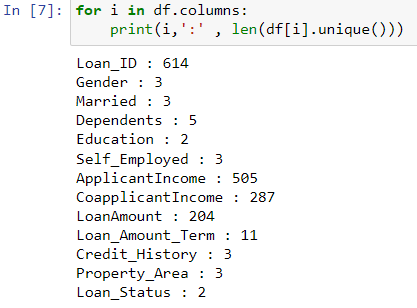


This is how i get an idea of null values different column contains. I then treated object type column with mode of that particular and integer type column with mean of that particular column.

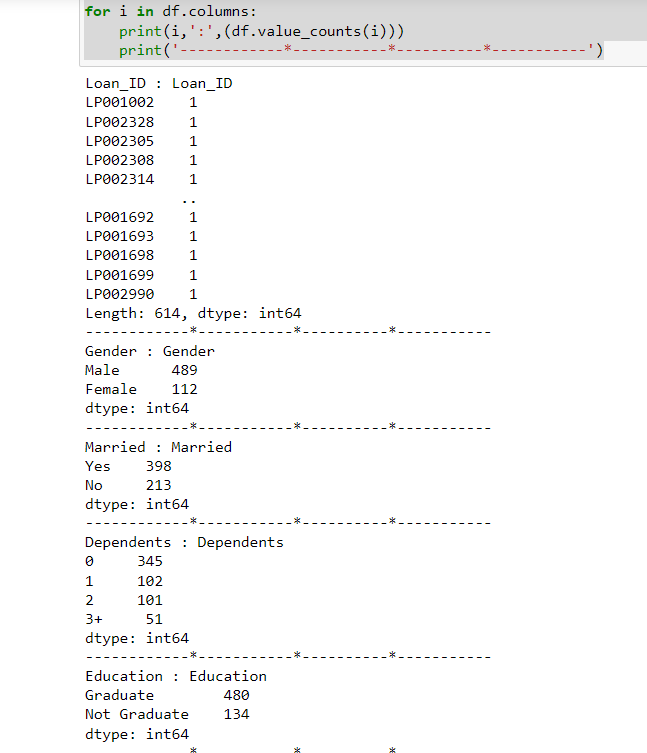


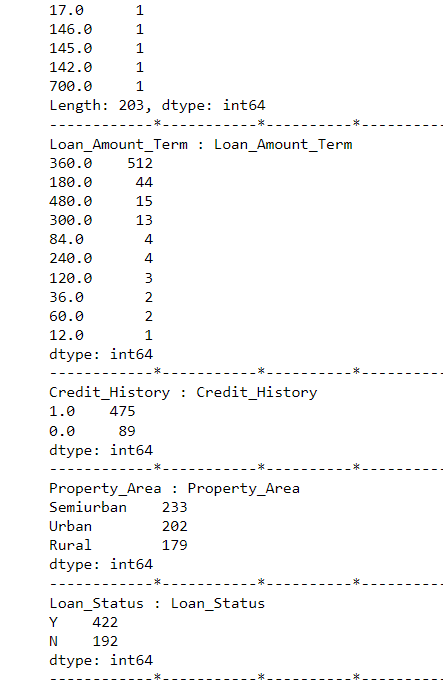
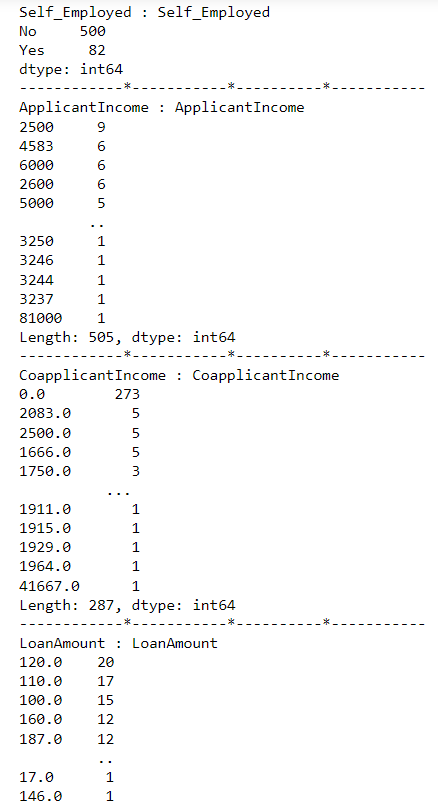
Now then after treating null values our data is free from null values as can be seen above and we can proceed further.

I further checked my data for Unique values each column contain :

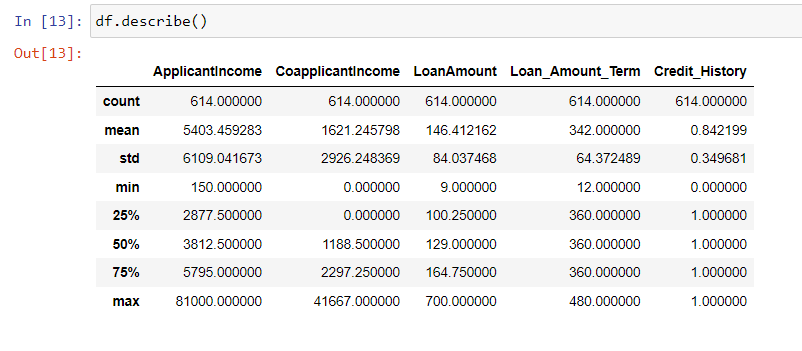


Now we’ll check the value counts for better insight of our data.





Above is the information about value counts of each feature. Now lets check the statistics of our data.



**Observation :**

From above data analysis these are the observation:

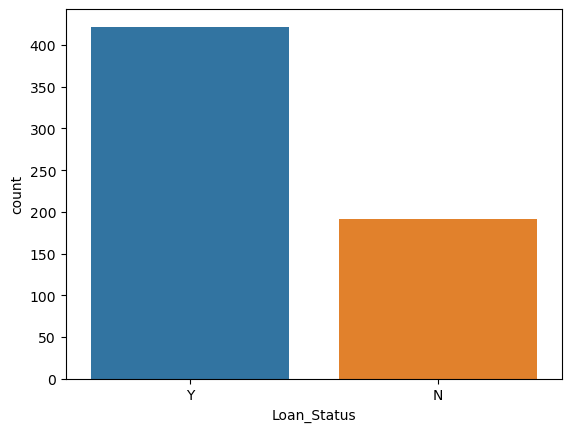
* Average income of an loan applicant is 5403.45 and maximum is 81000.
* Average income of an Coapplicant is 1621.24 and maximum is 81000.
* Average loan amount is 146.41 and maximum is 700.
* Average loan term is 342 and maximum is 480 Months.
* Credit History seems to be a categorical data either it’s 1 or 0 as we also saw in value count of this feature.

**4. Data Visualization**

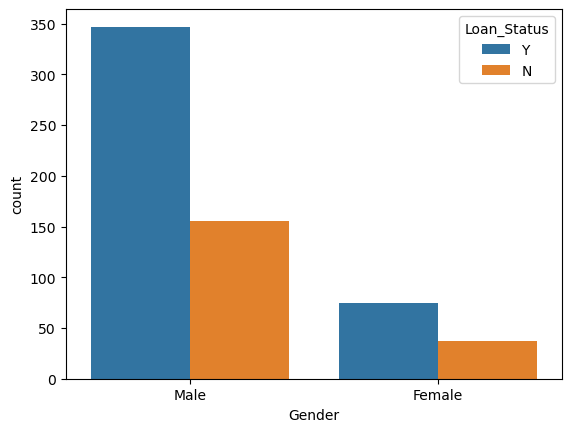
Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. The main goal of data visualization is to make it easier to identify patterns, trends and outliers in large [data sets](https://whatis.techtarget.com/definition/data-set).

Visualizing impact different features have on our label:

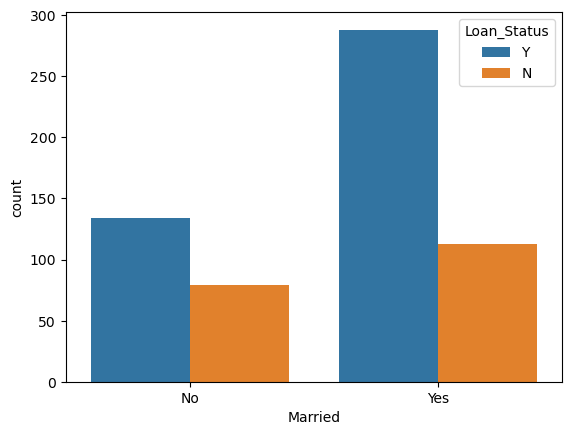
sns.countplot('Loan\_Status',data=df)



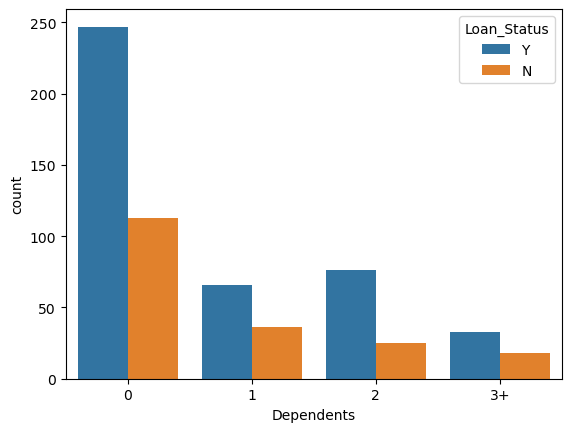
sns.countplot(x='Gender',data=df,hue='Loan\_Status')



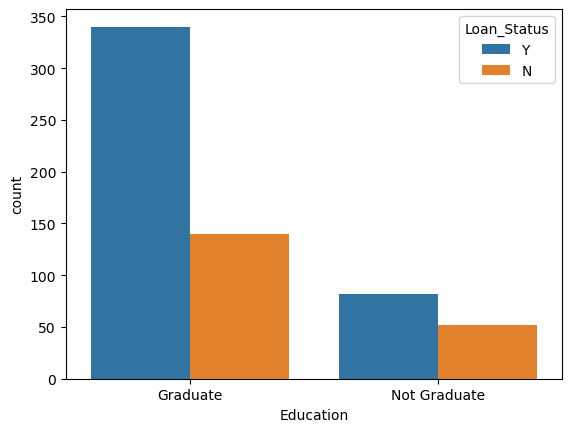
sns.countplot(x='Married',data=df,hue='Loan\_Status')



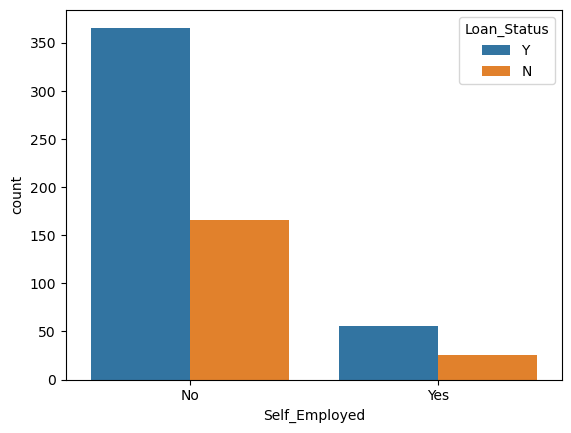
sns.countplot(x='Dependents',data=df,hue='Loan\_Status')



sns.countplot(x='Education',data=df,hue='Loan\_Status')



sns.countplot(x='Self\_Employed',data=df,hue='Loan\_Status')



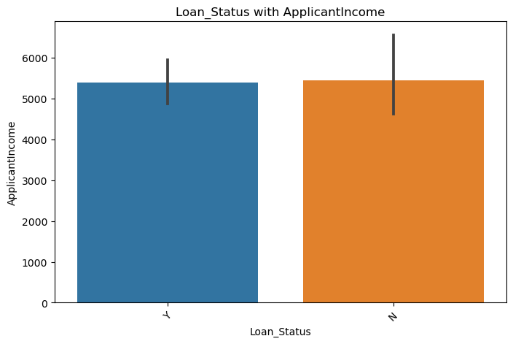
plt.figure(figsize=(8,5))

sns.barplot(y='ApplicantIncome',x='Loan\_Status',data=df)

plt.title("Loan\_Status with ApplicantIncome")

plt.xticks(rotation=45)

plt.show()



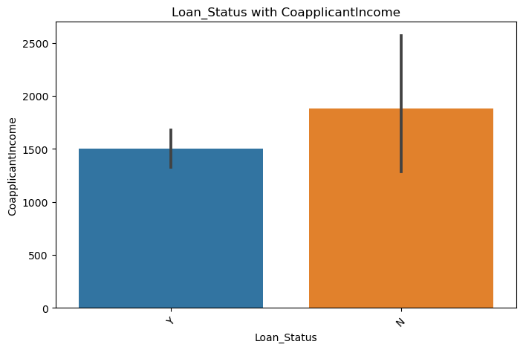
plt.figure(figsize=(8,5))

sns.barplot(y='CoapplicantIncome',x='Loan\_Status',data=df)

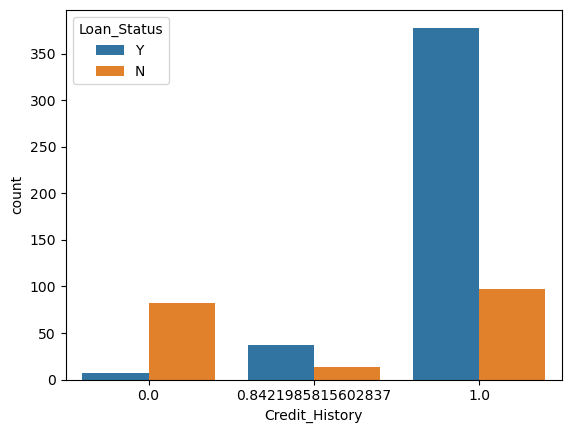
plt.title("Loan\_Status with CoapplicantIncome")

plt.xticks(rotation=45)

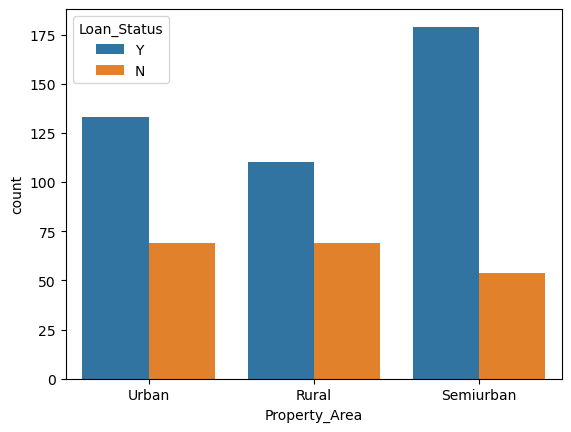
plt.show()



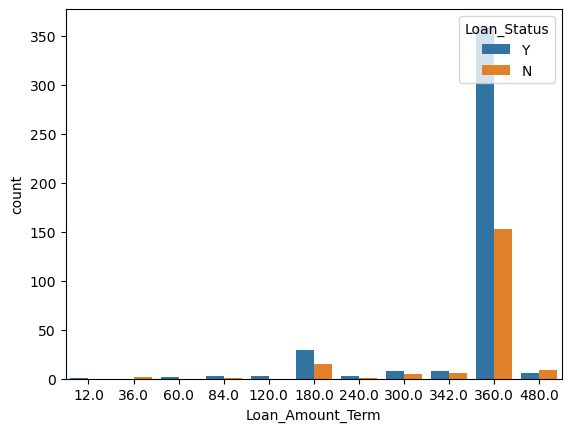
sns.countplot(x='Credit\_History',data=df,hue='Loan\_Status')



sns.countplot(x='Property\_Area',data=df,hue='Loan\_Status')



sns.countplot(x='Loan\_Amount\_Term',hue='Loan\_Status',data=df)

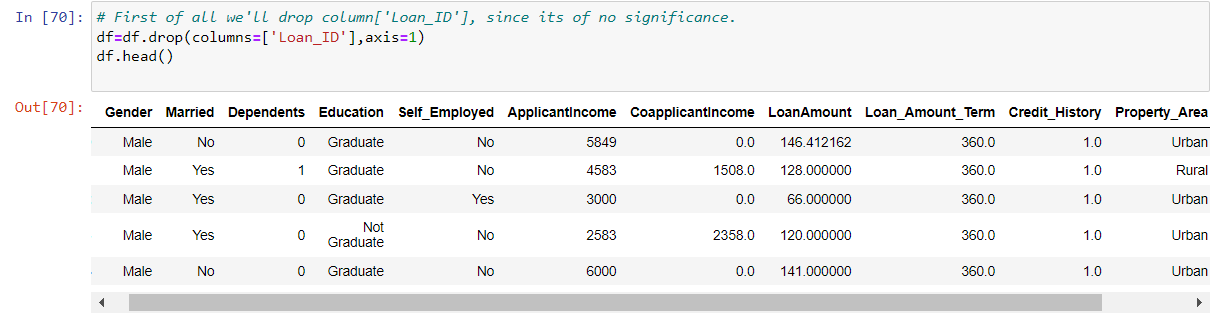


**Observation from data visualization:**

1. Count of application getting approved is high.
2. It seems like 'Male' applies for loan more often, and approval rate are also higher for 'male', Hence gender plays an important role.
3. It appears 'Married' applies for loan more often, and also seems have higher approval rate.
4. Applicant with 0 dependent mostly applies for loan, but applicant with 2 dependents have best approval rate.
5. Graduates oftenly apply for loan, education level plays an important role in approval of loan.
6. Salaried persons having higher count in approval of loan
7. People having Good Credit history can avail loan easily
8. Most of the applicant who applies for loan are from "Semiurban" area and there approval rates are quiet good, applicants from "Rural area " have worst approval rate.
9. Most of the people who applied for loan asked for 360 months of term more then half are getting approved
10. People those are taking loan for 480 months, most of them are not getting approval.

**5. Pre-Processing Pipeline.**

From all above analysis it’s clear that Loan Id is of no use in Determining whether the loan is get approved or is rejected, so we’ll drop it.

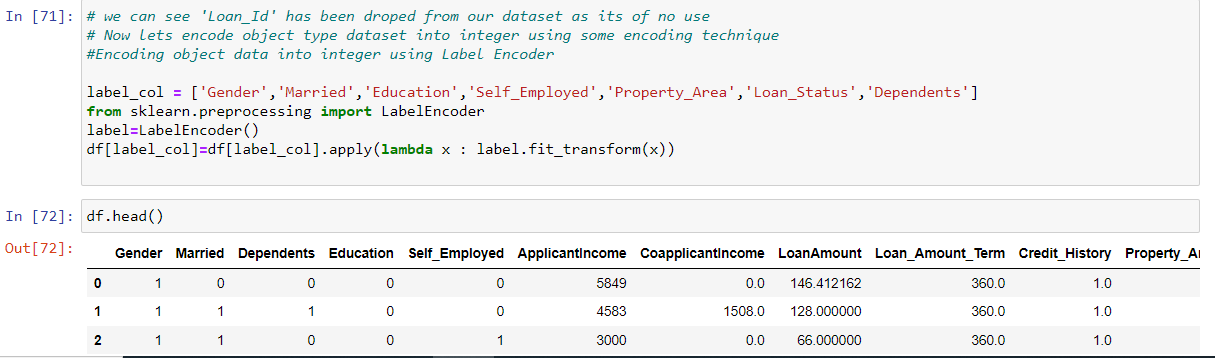


We can see above that the unnecessary column has been dropped.

**Label Encoding of categorical column:**

Encoding or continuization is the transformation of categorical variables to binary or numerical counterparts. An example is to treat male or female for gender as 1 or 0.

There are multiple encoding technique , I’m opting for label encoding.

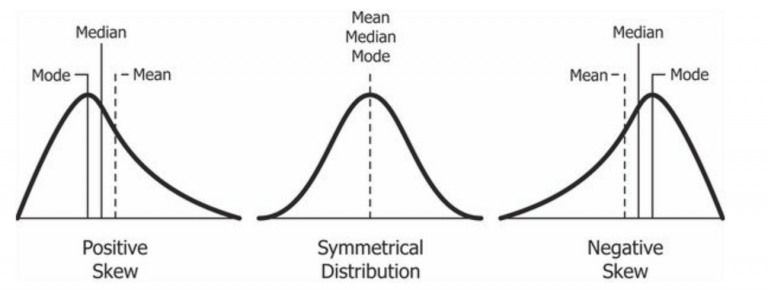


Above I have imported label encoder from sklearn.preprocessing and encoded all the categorical features, and it can also be seen how our data looks after encoding.

**Skewness**

Skewness is the measure of the asymmetry of an ideally symmetric probability distribution and is given by the third standardized moment. If that sounds way too complex, don’t worry! Let me break it down for you.

In simple words, skewness is the measure of how much the probability distribution of a random variable deviates from the normal distribution.



# Checking Data Distribution

plt.figure(figsize=(20,25))

plotnumber=1

for column in df:

if plotnumber<=15 :

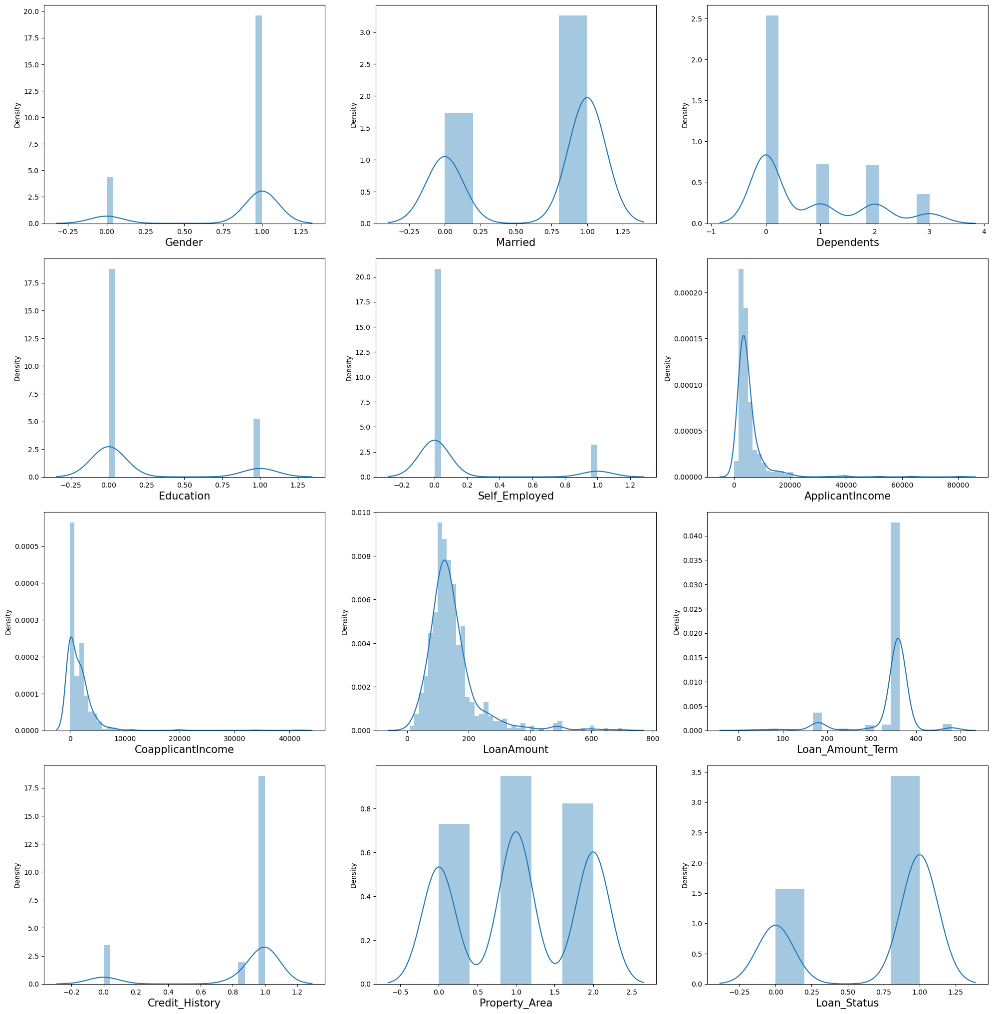
ax=plt.subplot(5,3,plotnumber)

sns.distplot(df[column])

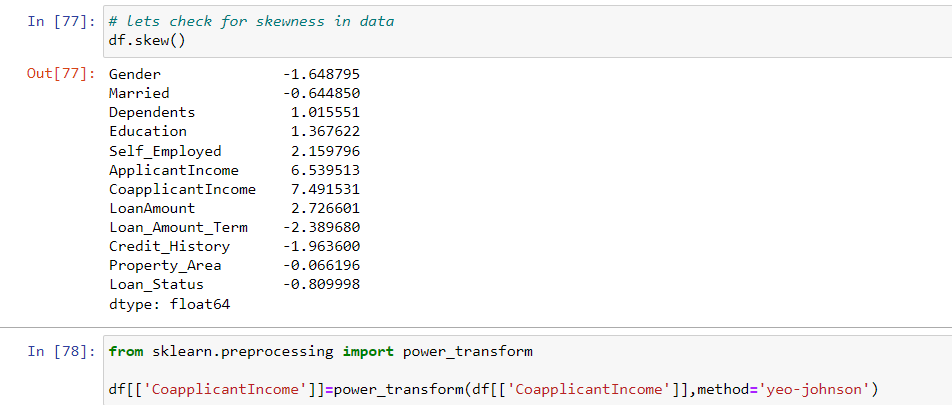
plt.xlabel(column , fontsize=15)

plotnumber+=1

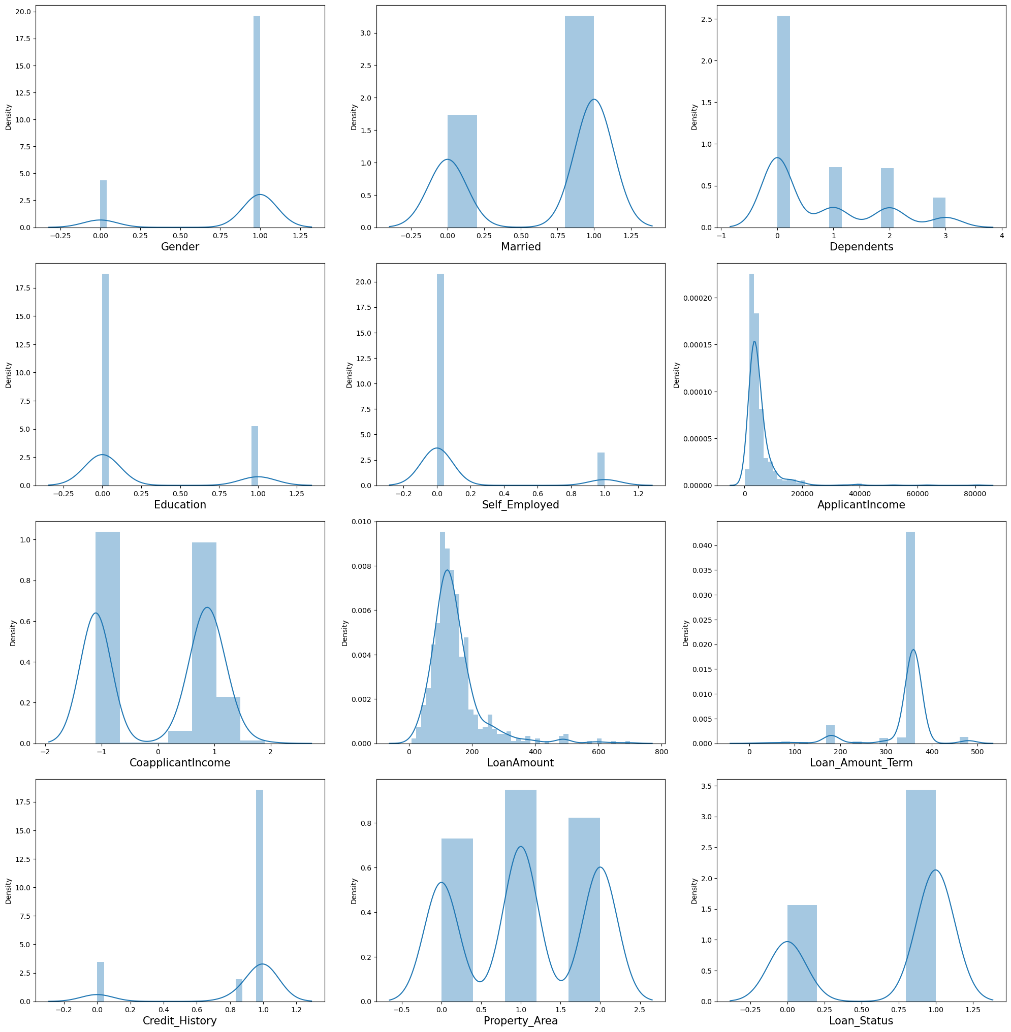
plt.tight\_layout()



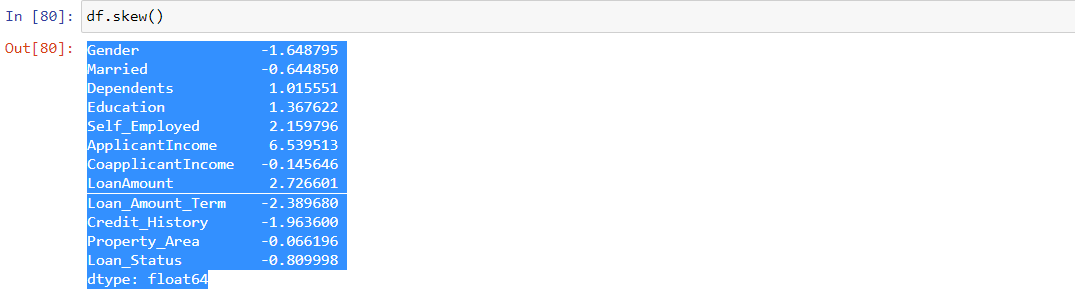
Keeping aside the categorical data , our contineous data distribution seems to be little skeweed lets treat it.



Above we can see I have imported power transform and treated the skewness.



Above is the distribution after skewness correction.



As we have treated skewnees , lets check our dataset for outliers.

**Outlier Detection**

An outlier is an object(s) that deviates significantly from the rest of the object collection. It is an abnormal observation during the Data Analysis stage, that data point lies far away from other values. An outlier is an observation that diverges from well-structured data.

Let’s detect the outlier using box plot:

# checking if there is any kind of outlier presence in data or not using box plot

plt.figure(figsize=(25,20))

plotnumber=1

for column in df:

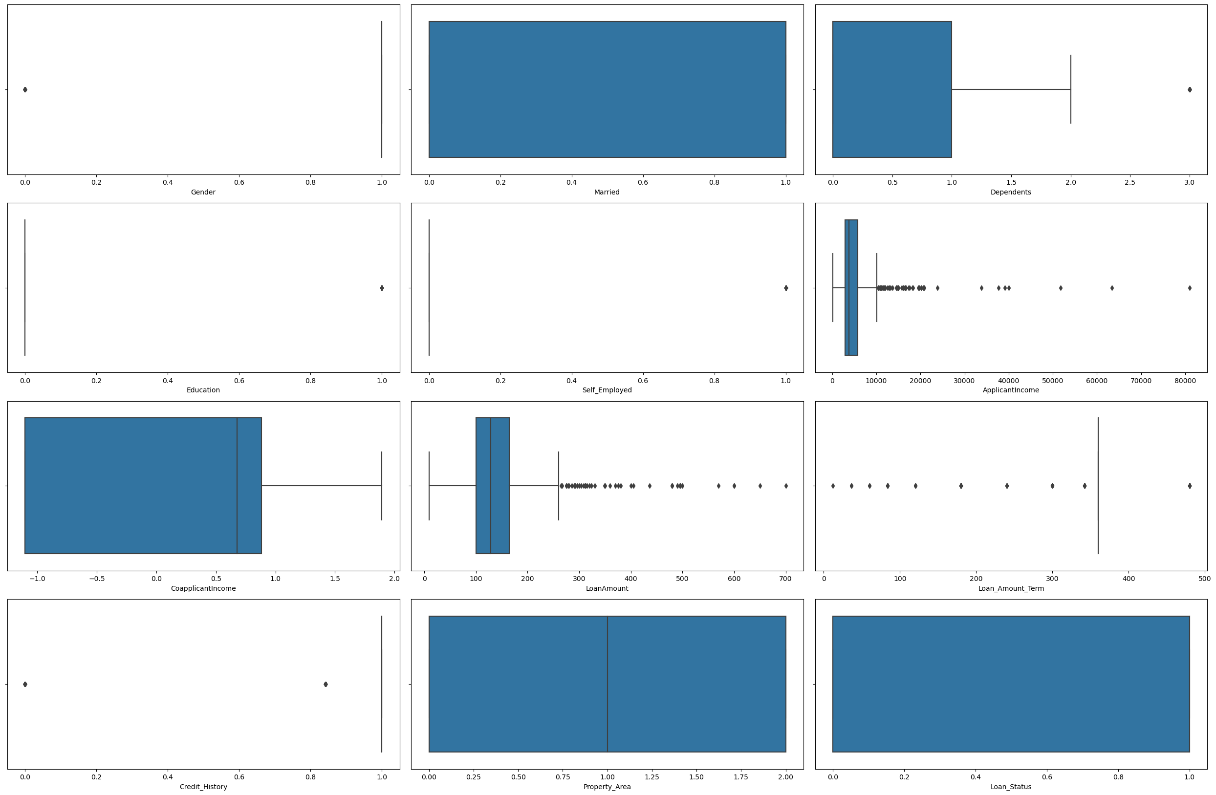
if plotnumber<=15 :

ax=plt.subplot(5,3,plotnumber)

sns.boxplot(df[column]

plotnumber+=1

plt.tight\_layout()



since it seem there are few outlier present in our dataset ,lets treat them before moving further. There are different method to remove outliers

Some of the most popular methods for outlier detection are:

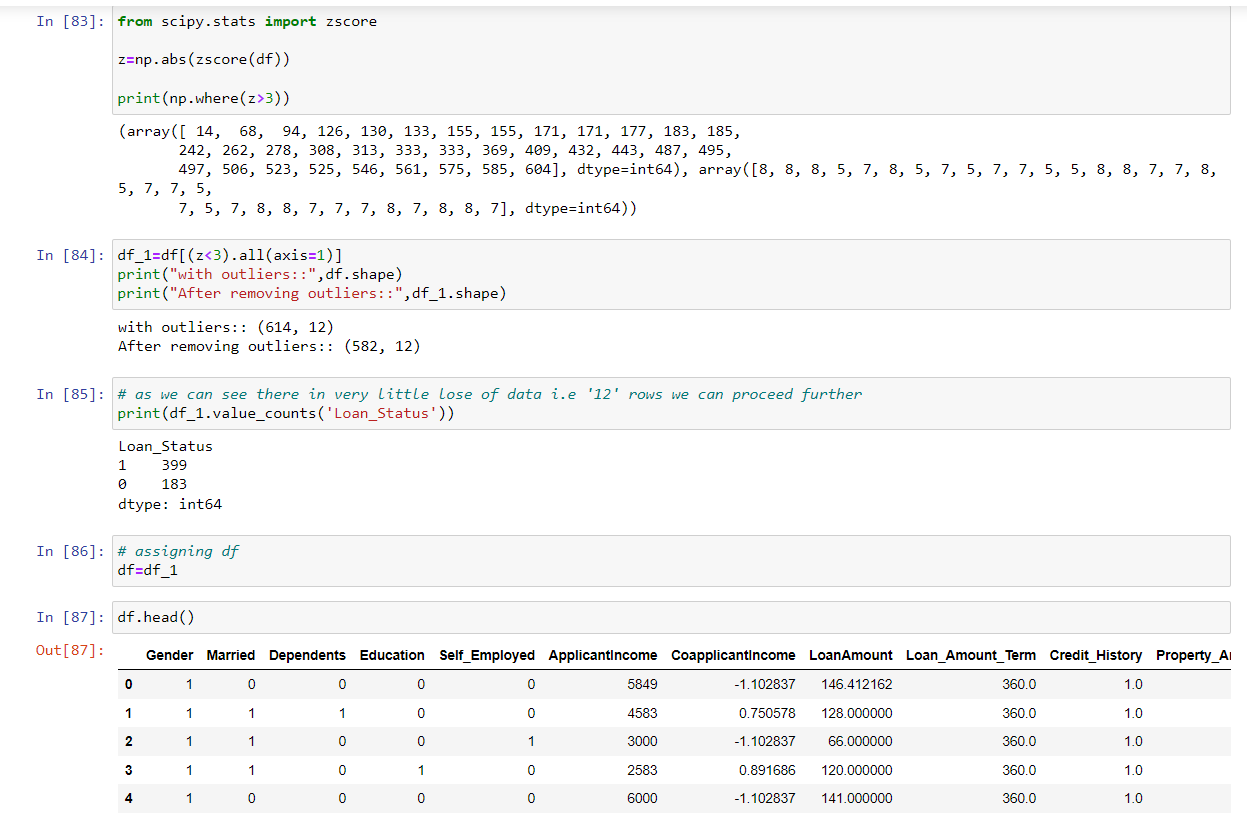
* Z-Score or Extreme Value Analysis (parametric)
* Probabilistic and Statistical Modeling (parametric)
* Linear Regression Models (PCA, LMS)
* Proximity Based Models (non-parametric)
* Information Theory Models
* High Dimensional Outlier Detection Methods (high dimensional sparse data)

Selecting z-score

Z-score is a simple, yet powerful method to get rid of outliers in data if you are dealing with parametric distributions in a low dimensional feature space.

 The z-score of any data point can be calculated with the following expression:





Above you can see that I have imported z score and removed all the outliers.

And we can also see that a very little data has been lost ,so we are good to go.

**6. Building Machine Learning Models.**

**Separating features and label:**

X=df.drop(columns=['Loan\_Status'],axis=1)

y=df['Loan\_Status']

**Scaling**

Most of the times, our dataset will contain features highly varying in magnitudes, units and range. But since, most of the machine learning algorithms use Euclidean distance between two data points in their computations, this is a problem. If left alone, these algorithms only take in the magnitude of features neglecting the units. The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes. To suppress this effect, we need to bring all features to the same level of magnitudes. This can be achieved by scaling. Statistical formula of standardisation is as follows:-

Stand_eq.gif

**Scaling features**

scalar=StandardScaler()

x\_scaled=scalar.fit\_transform(X)

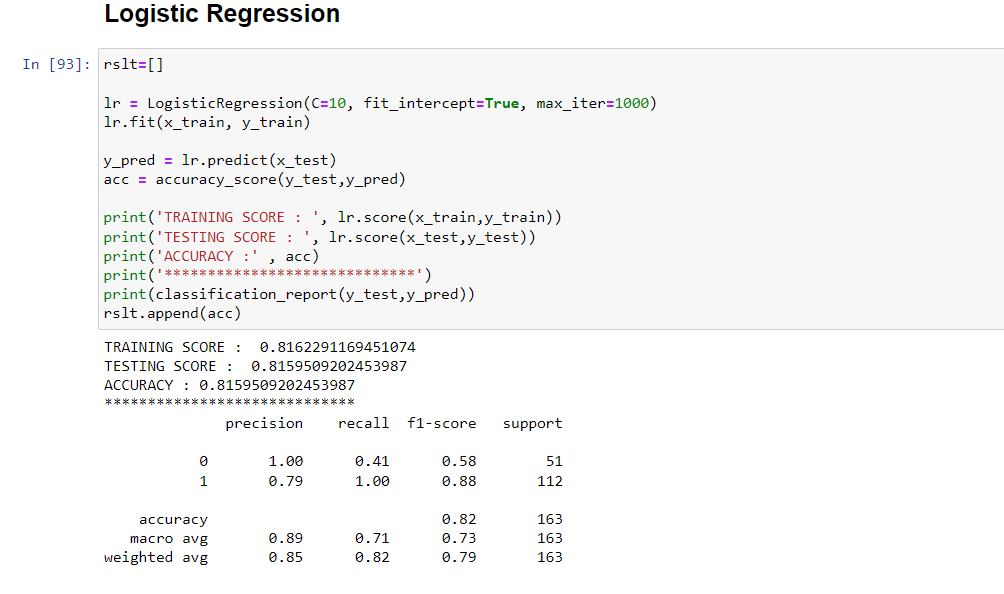
**Spliting our dataset into train and test**

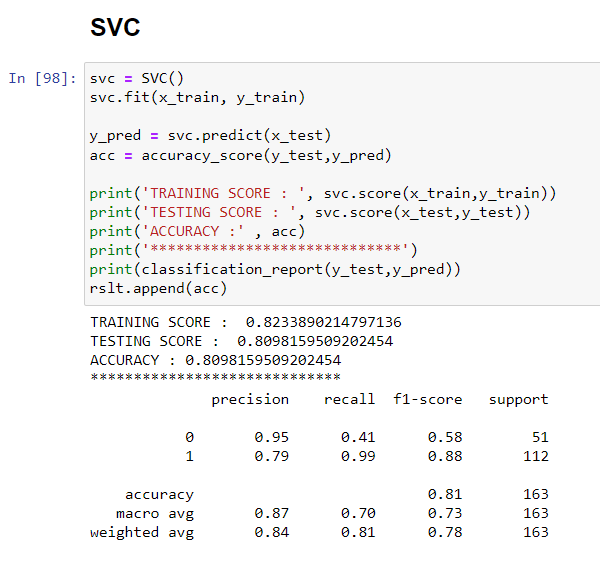
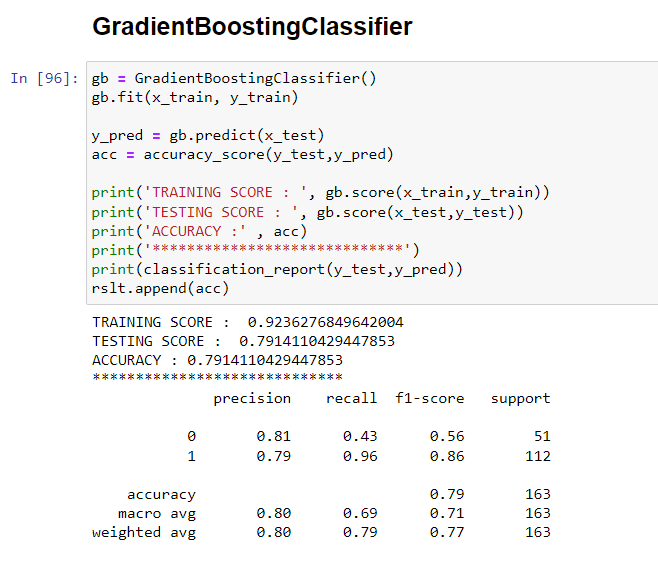
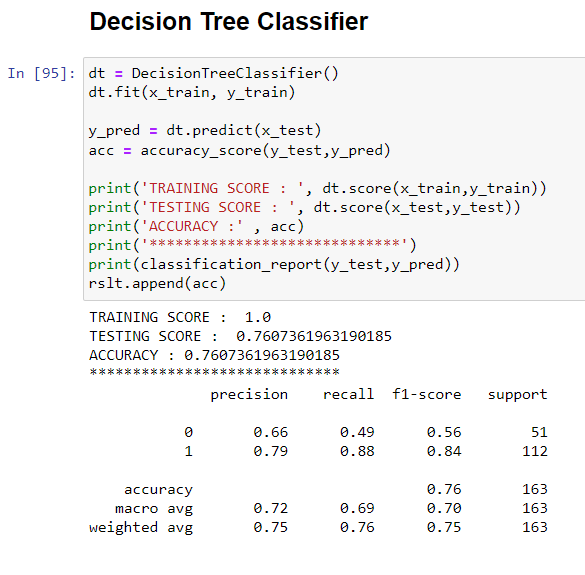
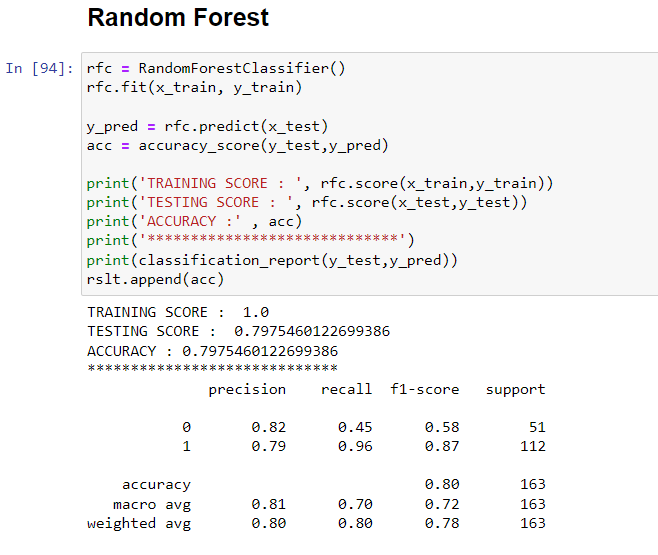
we are splitting our dataset into train and test so that we can train aur data in training data set and test it on testing dataset:

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x\_scaled,y,test\_size=0.28,random\_state=45)

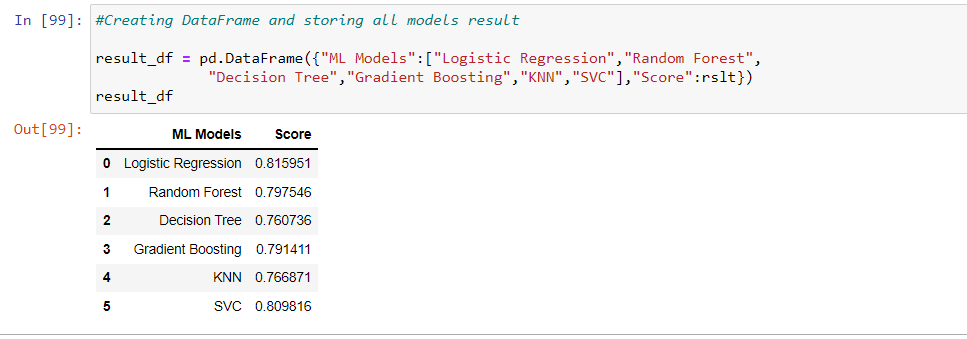
**Model Building**

In here we will use various classification algorithm to predict our target. And determine the accuracy score of each model.

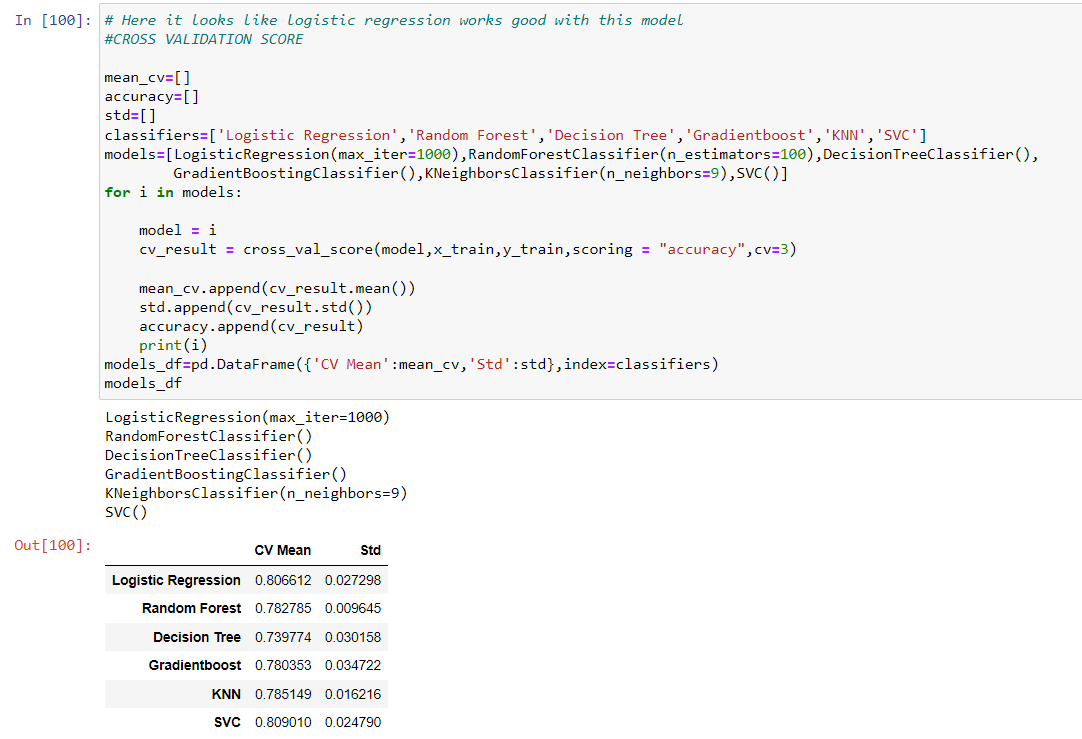
****

****

Here we have applied different classification algorithm and determine accuracy and different factor help in deciding best fit model.

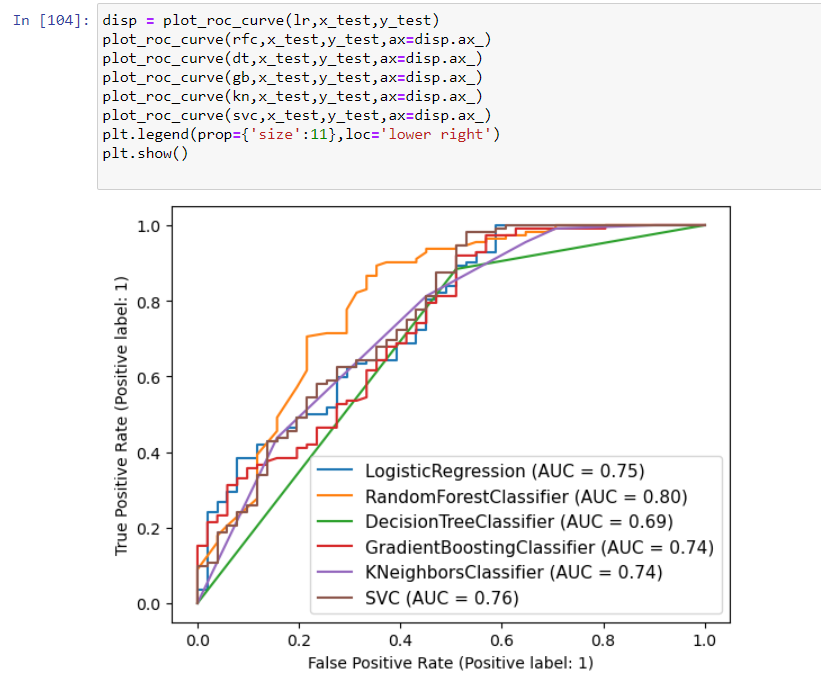


From above dataframe we can determine logistic regression is performing good but let’s confirm it. By cross validating each model and determine if by chance our model is over or under fit.



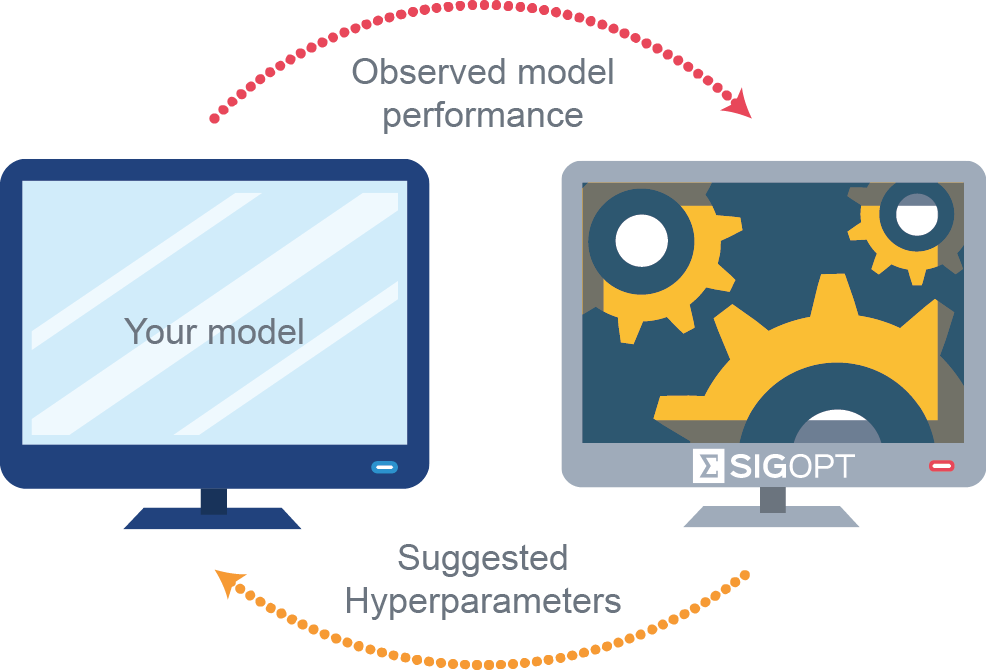
Let us plot roc auc curve to confirm the best performing Model .

Roc-Auc Curve: Roc- Auc curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing.

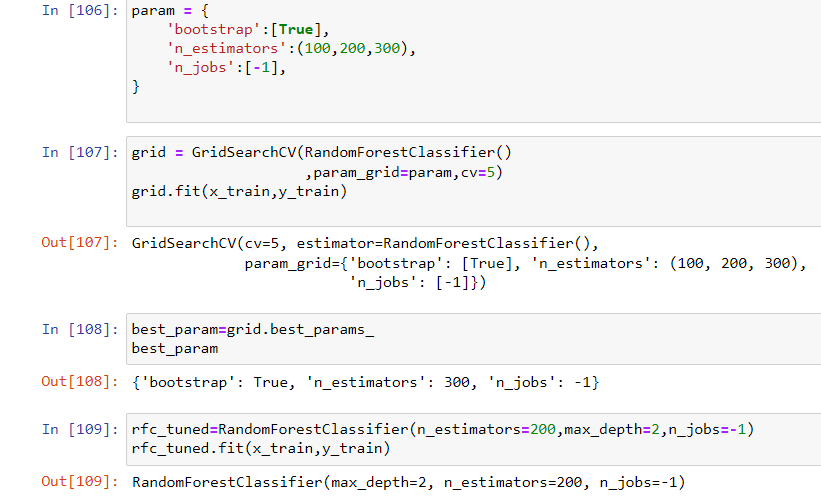


#### **According to Accuracy Score and CV mean Logistic regression is our Best fitted Model.But, as we move on to ROC-AUC curve, it shows that Random Forest is best model to predict.**

**Hyper parameter tuning:**

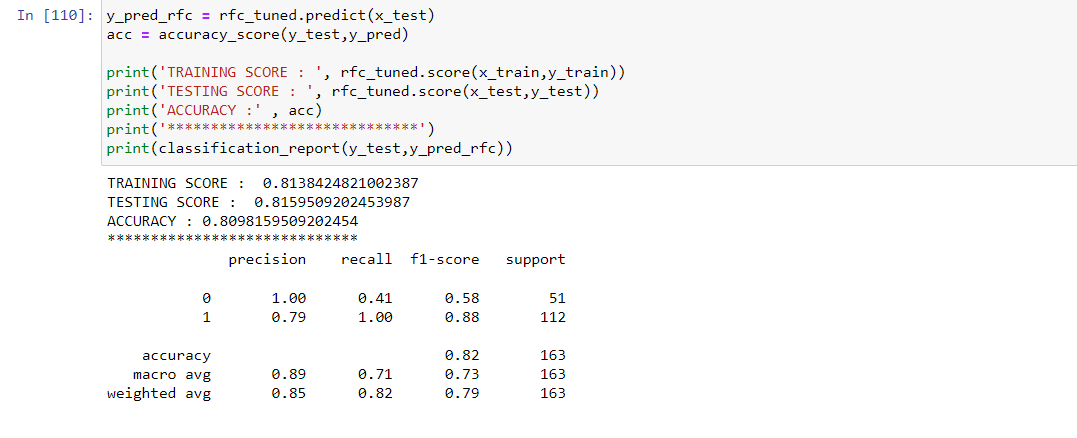
****

A hyperparameter is a **parameter whose value is used to control the learning process**.



Above we can see I have hyper tuned our best model using Grid Search CV. Now let’s train our tuned model and see the performance.

Below we can see the performance of our tuned model.



## As we can see I have achieved better accuracy and training score with the help of hyper parameter tuning.

## Saving Model

## It’s very important to save our model after successfully building it.

## import joblib

## joblib.dump(rfc\_tuned,'Loan\_Application\_Status.pkl')

**7. Conclusion**

1. **Factor most impacting the approval of loan are Gender ,Marital Status , Dependent member ,Educational status , whether applicant is salaried or not , there credit score and where they live.**
2. **Male applicant, married applicant , applicants with 2 dependents , graduates , salaries applicant applicant having good credit score , applicant from semiurban area , applicant who demanded 360 month loan term have good approval rate.**
3. **I Hope this model will help companies to validates the customer eligibility for the loan.**
4. **This model will save a lot of time and hard word .**

