**Loan Application Status Prediction**

***Problem statement***:

**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.

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

Will 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.

**TYPE OF PROBLEM:**

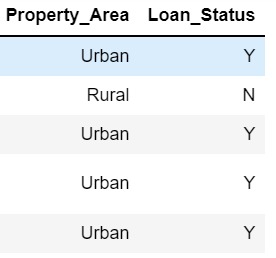
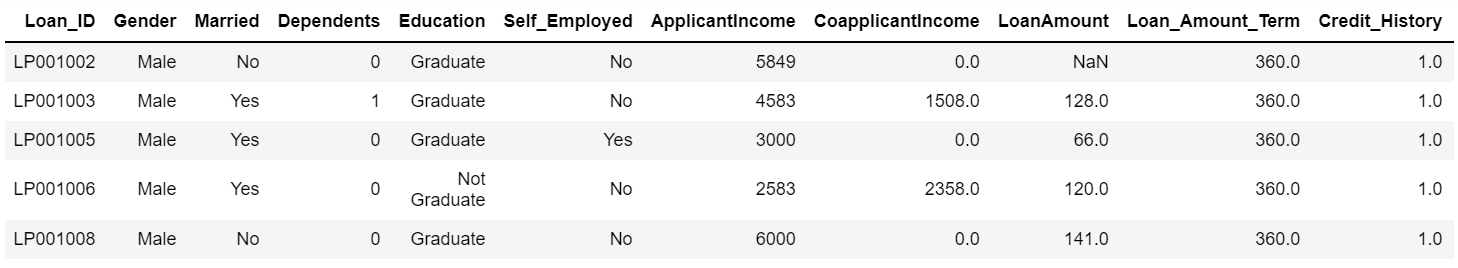
The above problem is a clear classification problem as we need to classify whether the Loan\_Status is yes or no. So this can be solved by any of the classification techniques like

1. Logistic Regression .
2. Decision Tree Algorithm.
3. Random Forest Technique.

I have mentioned only few. We will be dealing with each of techniques later in this blog.

**Description about the Data Columns**:

There is only 1 data set, which needs to be train & test with the help of the actual target variable as ‘Loan Status’. Train file will be used for training the model, i.e. the model will learn from this file. It contains all the independent variables and the target variable. Size of training set: 614 rows & 13 columns.



Loan\_ID: Loan ID for the Applicant applying for a loan

Gender: Gender of the Applicant

Married: Applicant’s marital status

Dependents: Number of dependents of the Applicant

Education: Applicant’s education status (Graduate/Under Graduate)

Self\_Employed: Applicant is self-employed or not

ApplicantIncome: Applicant’s Income

CoapplicantIncome: Co-applicant’s Income

LoanAmount: Loan Amount taken

Loan\_Amount\_Term: Term of the loan in months

Credit\_History: Applicant’s previous credit history meeting guidelines

Property\_Area: Urban, Semi-Urban, or Rural Areas

Loan\_Status: Loan Approval status (Target Variable)

Now let us look in to the each variable and can make some assumptions.(It’s just assumptions right, there is no harm in just assuming few statements)

Loan ID -> As the name suggests each person should have a unique loan ID.

Gender -> In general it is male or female. No offence for not including the third gender.

Married -> Applicant who is married is represented by Y and not married is represented as N. The information regarding whether the applicant who is married is divorced or not has not been provided. So we don’t need to worry regarding all these.

Dependents -> the number of people dependent on the applicant who has taken loan has been provided.

Education -> It is either non -graduate or graduate. The assumption I can make is “ The probability of clearing the loan amount would be higher if the applicant is a graduate”.

Self\_Employed -> As the name suggests Self Employed means , he/she is employed for himself/herself only. So freelancer or having a own business might come in this category. 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 by Applicant.So the general assumption that i can make would be “The one who earns more have a high probability of clearing loan amount and would be highly eligible for loan ”

Co Applicant income -> this represents the income of co-applicant. I can also assume that “ If co applicant income is higher , the probability of being eligible would be higher “

Loan Amount -> This amount represents the loan amount in thousands. One assumption I can make is that “ If Loan amount is higher , the probability of repaying would be lesser and vice versa”

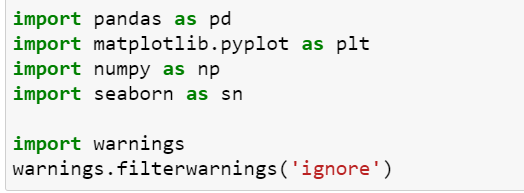
Loan\_Amount\_Term -> This represents the number of months required to repay the loan.

Credit\_History -> When I googled it , I got this information. A credit history is a record of a borrower’s responsible repayment of debts. It suggests → 1 denotes that the credit history is good and 0 otherwise.

Property\_Area -> The area where they belong to is my general assumption as nothing more is told. Here it can be three types. Urban or Semi Urban or Rural

Loan\_Status -> If the applicant is eligible for loan it’s yes represented by Y else it’s no represented by N.

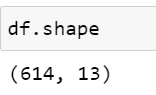
***Data Analysis:***Now let me walk through the code. Firstly I just imported the necessary packages like pandas, numpy, seaborn etc. so that i can carry the necessary operations further.

******

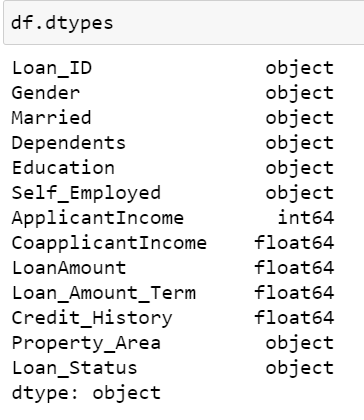
Now I am going to upload or read the files/data-sets using pandas. For this we used read\_csv



By using df.shape I understood that there are 614 rows and 13 columns

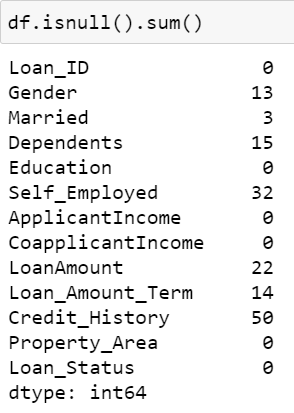


By using df.dtypes we understood about the types of each columns. We have maximum columns as categorical including labels & few are integer

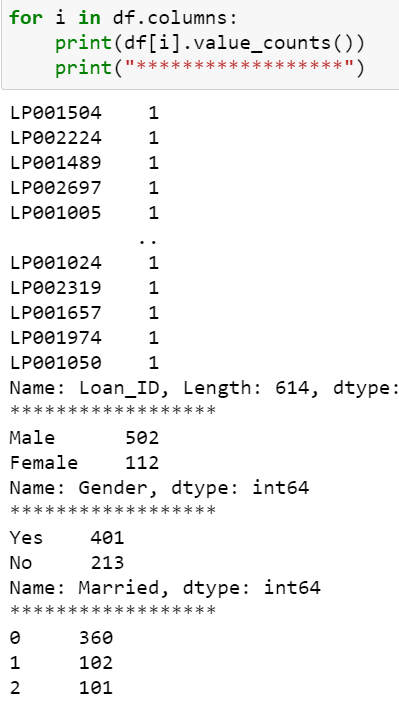


Dealing with null values

Out of all these columns, we can see few columns are having missing values. Now, we need to check the data type & accordingly we would fill the null values.



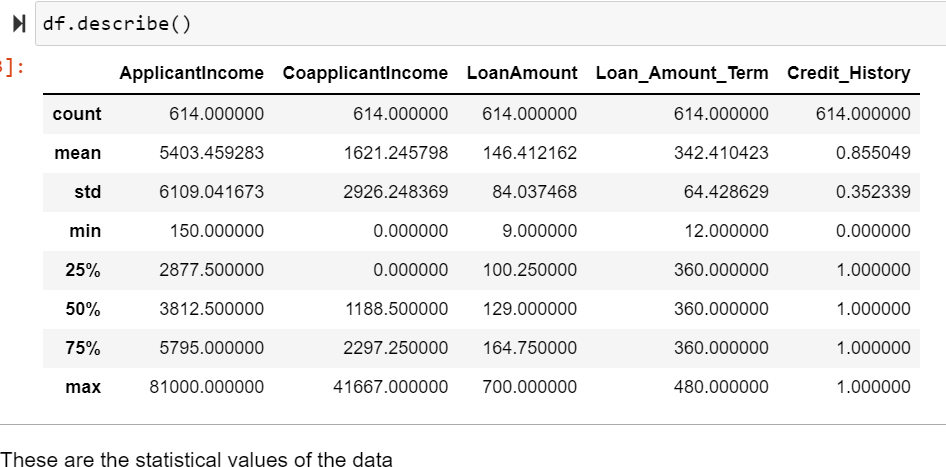
Now let us analyse the data using single variable:



**Conclusions**: (Through Single Variable Analysis)

1. We can see that approximately 81% are Male and 19% are female.
2. Percentage of applicants with no dependents is higher.
3. There are more number of graduates than non graduates.
4. Semi Urban people is slightly higher than Urban people among the applicants.
5. Larger Percentage of people have a good credit history.
6. The percentage of people that the loan has been approved has been higher rather than the percentage of applicant for which the loan has been declined.

understanding the data at a high level. Checking the statistics of the data set



Findings:

· For a few features standard deviation is far from the mean, it may be because of data is more scattered.

· Co-applicant is having a minimum income of zero.

· Maximum loan applied is 700 having a minimum term of 480.

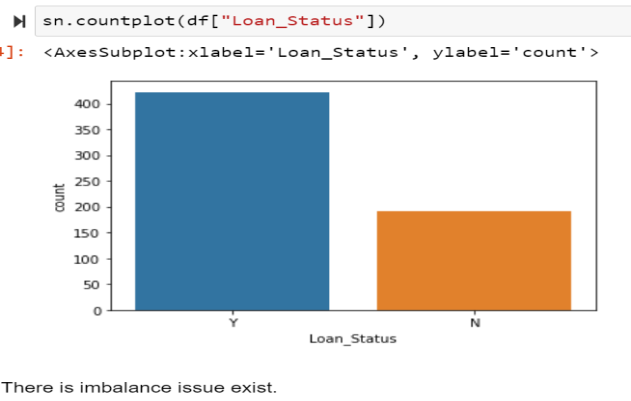
· Minimum loan applied is 9 having minimum term 12.

· Missing values in the Loan amount, loan amount term & credit history.

**Exploratory Data Analysis:**

[EDA]

Univariate analysis



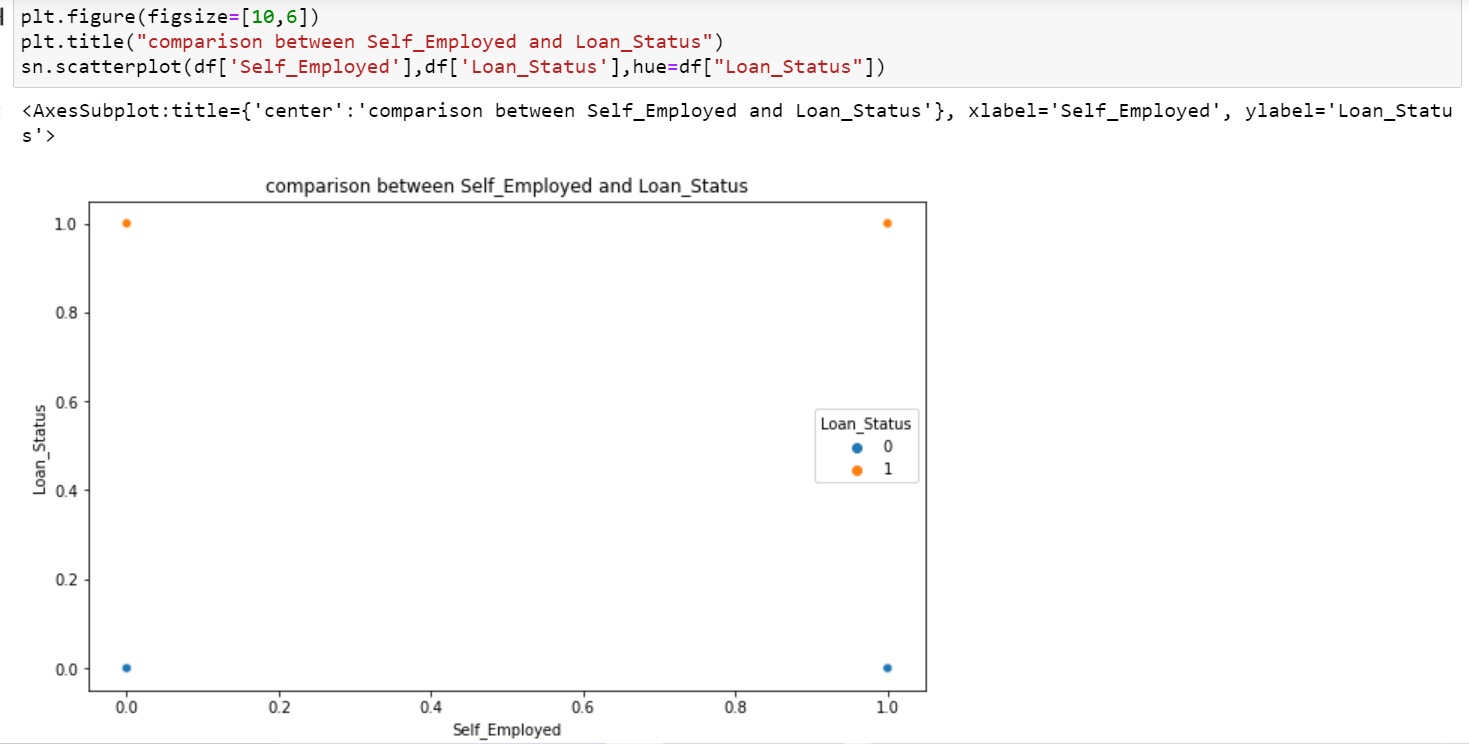
Well don’t get to worry about the fancy names like exploratory data analysis and all. By looking at the columns description in the above paragraph, we can make many assumptions like

1. The one whose salary is more can have a greater chance of loan approval.
2. The one who is graduate has a better chance of loan approval.
3. Married people would have a upper hand than unmarried people for loan approval .
4. The applicant who has less number of dependents have a high probability for loan approval.
5. The lesser the loan amount the higher the chance for getting loan.

Why are we doing EDA?

Like these there are many more we can assume. But one basic question you may get it …”Why are we doing all these ? Why can’t we do directly modeling the data instead of knowing all these…..” Well in some cases we can easily come to conclusion if we just to do EDA. Then there is no necessary for going through next models.

Bivariate analysis



most of the data we have from not employed person

Here we see if the person is not self employed there is high chances of approval

There are more than 350 people are approved and more than 150 are not approved those are not self\_employed

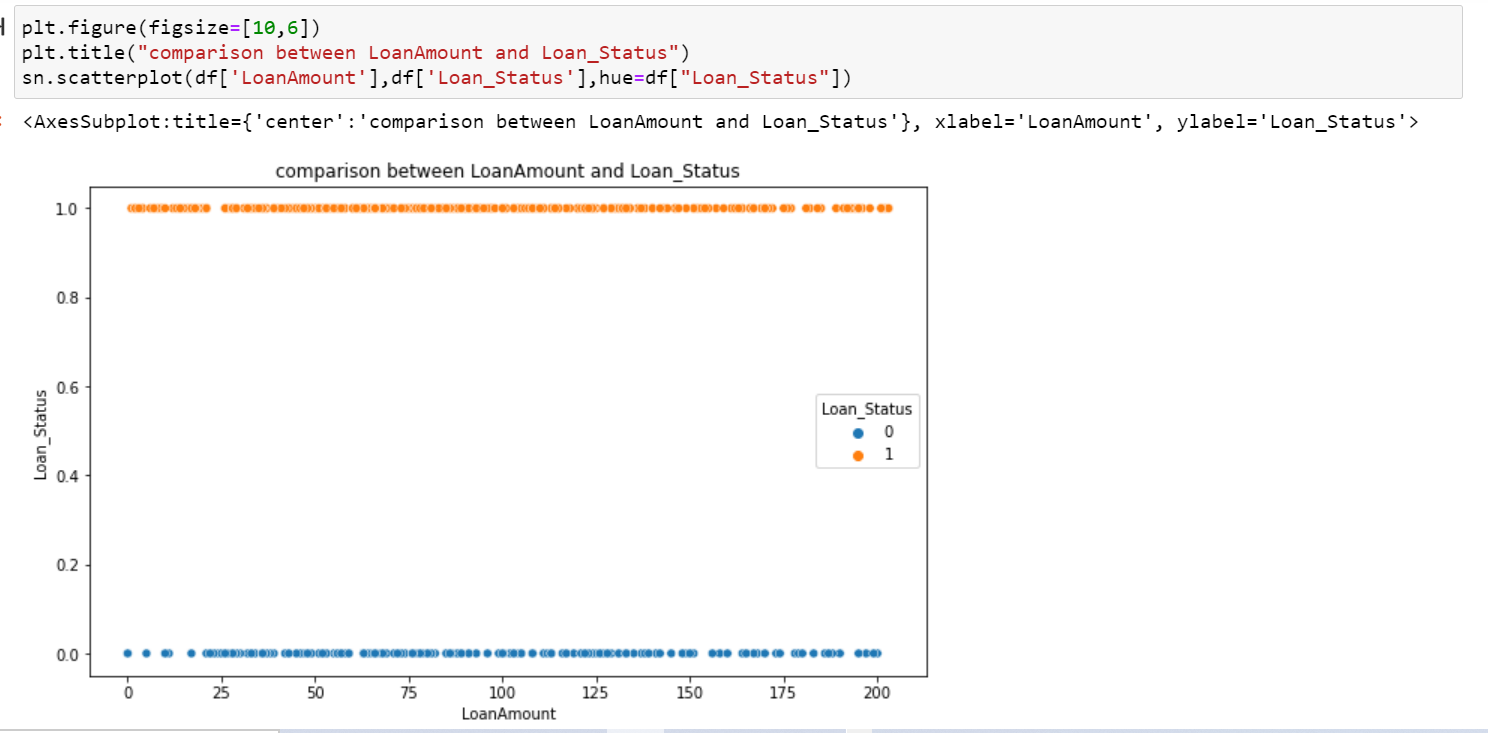
In self\_employed we see there around 50% of diffrence between approved and not approved people.

**

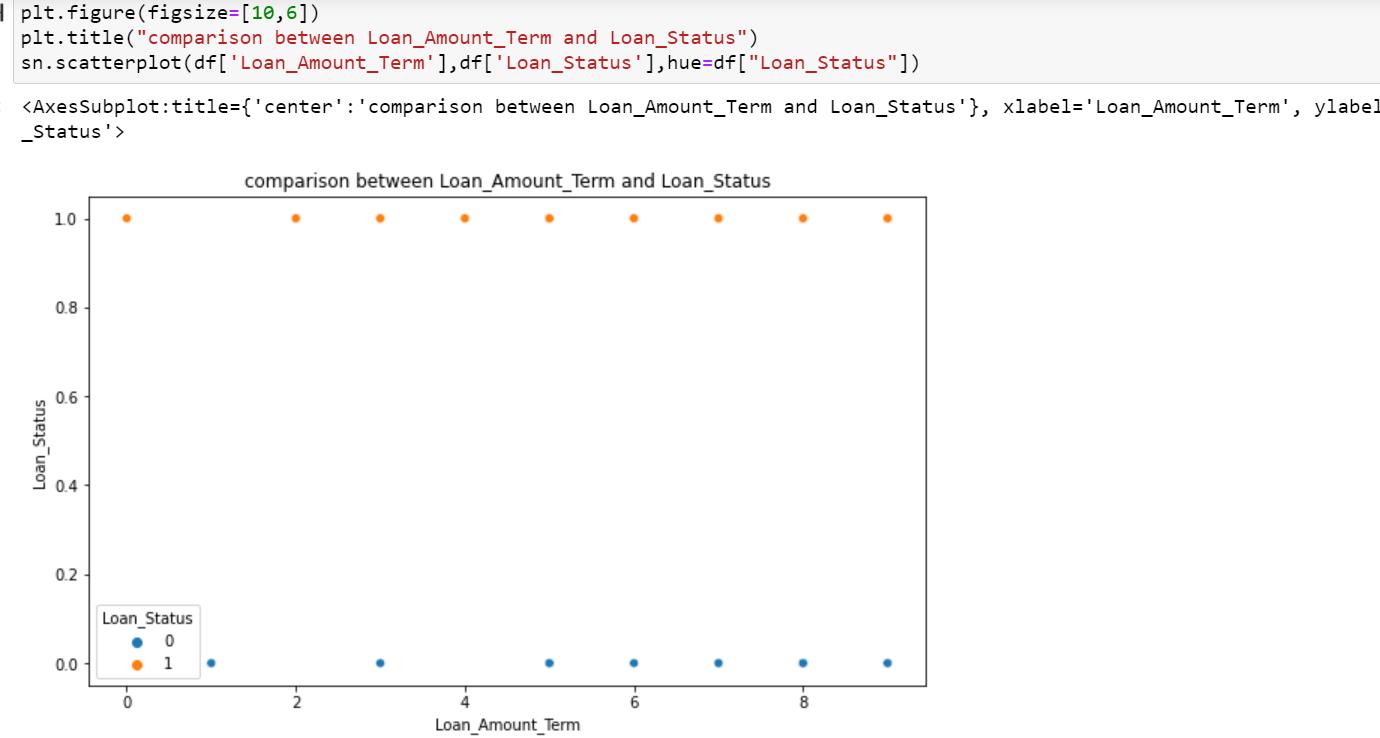
This shows the applicantincome against loanstatus. This values are having almost equal number of chances.



No. of max income value is less, which also gets not approved for a loan. Few co-applicants have 0 income, still, they got approval for the loan as the corresponding applicant showed income.



Higher values of the loan were approved. Data looks scattered, it has skewness/Outliers as maybe few people might have applied for higher loans. Let’s impute missing values for Loan Amount with the median as values are already scattered.

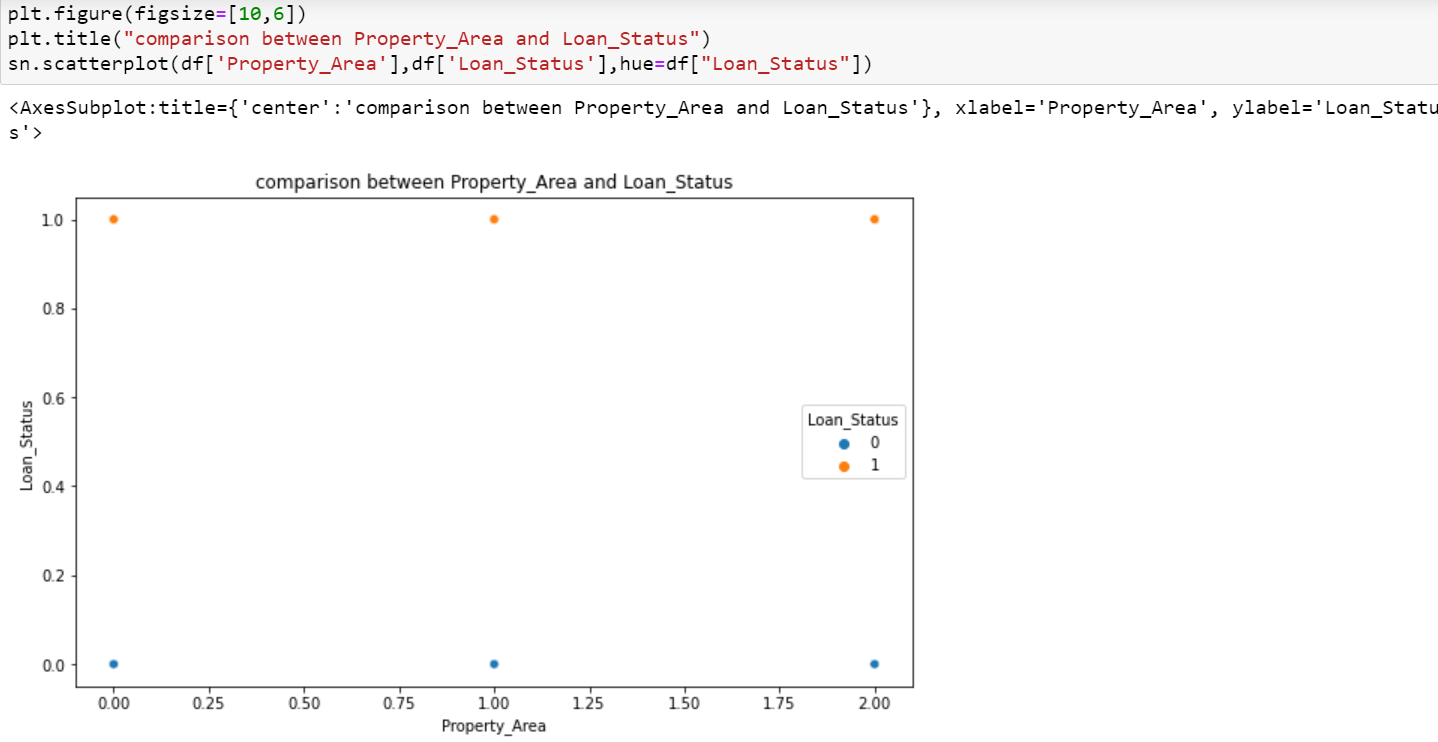


Most of the people who applied for loan asked for 360 months of term more half are getting approved

People those are taking loan for 480 months, most of them are getting approval



Here we have the logical figure most of the people those are not getting approval, not having credit history. The people those having credit history, most of them getting approval



In RuralProperty\_area not approval chances are increasing

In Semiurban area there are high chances to get approval

**Checking the correlation between variables:**







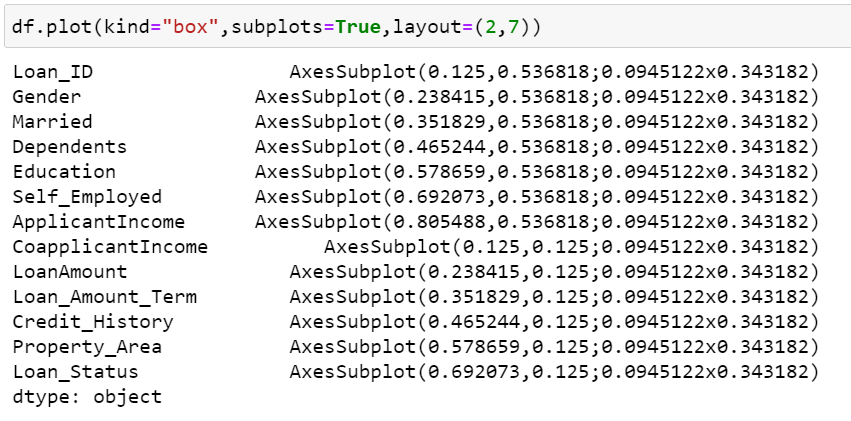
## Action Taken based on observation:

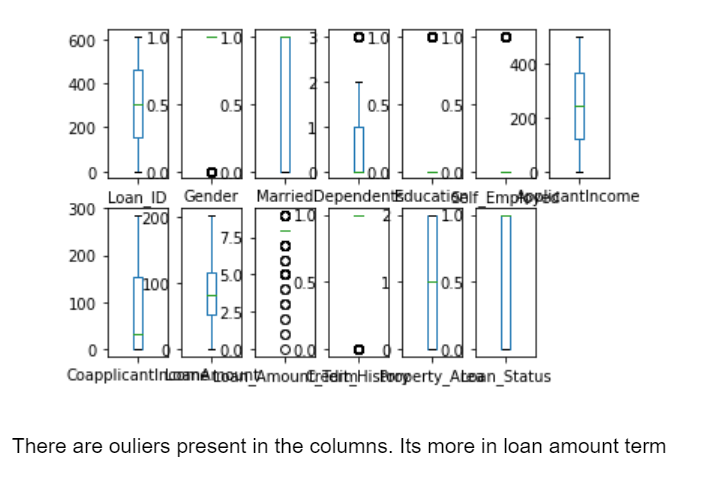
· We have seen how categorical & continuous data is distributed & correlated with the target variable. Also, I have mentioned the findings below each plot.

· Credit history is more correlated with target variable and we have multi-collinearity issues among applicant income, co-applicants & loan amount. Hence, we have added both Applicant Income and Coapplicant Income to Total Income as co-applicant income has some 0 values & total income would make more sense. Also, dropped columns Coapplicant Income & Applicant Income.

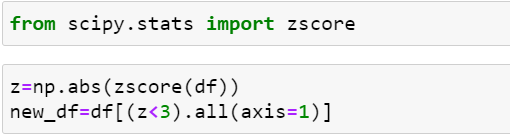
· Performed encoding by using the Label encoder method for all the categorical columns.

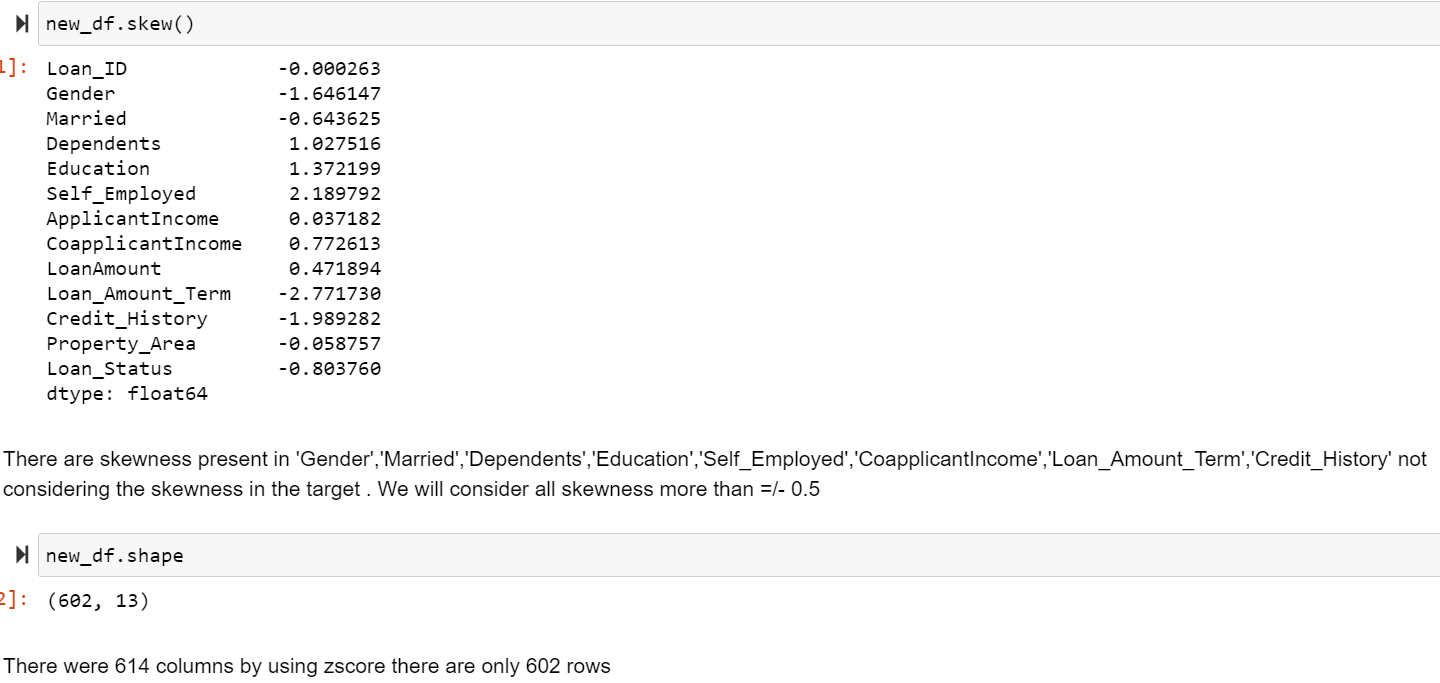
**Checking the outliers**



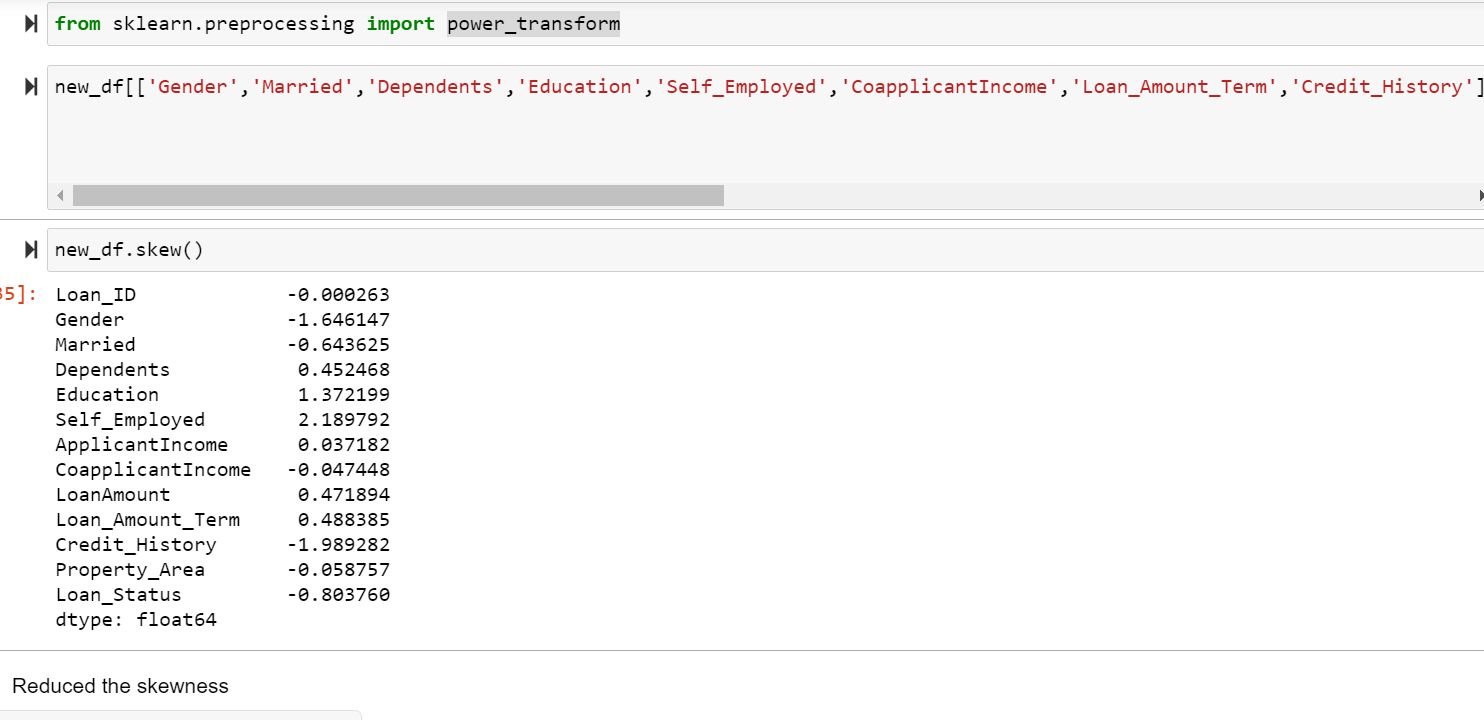


**Outlier Remover/ skewness**



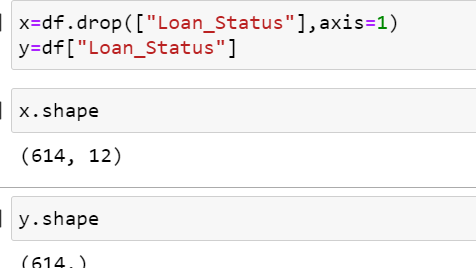


As we can see we have lost almost 5% of data to remove outlier’s which is fine. We have a maximum column as categorical, we need to check the skewness for continuous features: Loan amount & Total income. Let’s remove skewness with  power\_transform



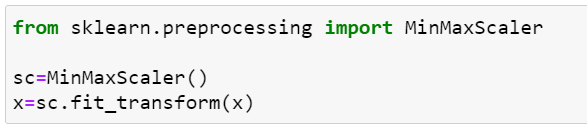
# *Pre-Processing Pipeline*

# Dividing Data into x and y



Divided the data into two parts features (x) and target (y) for model

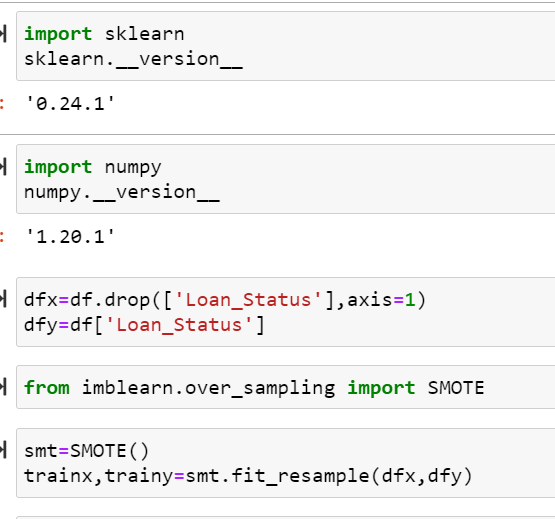
# Scaling

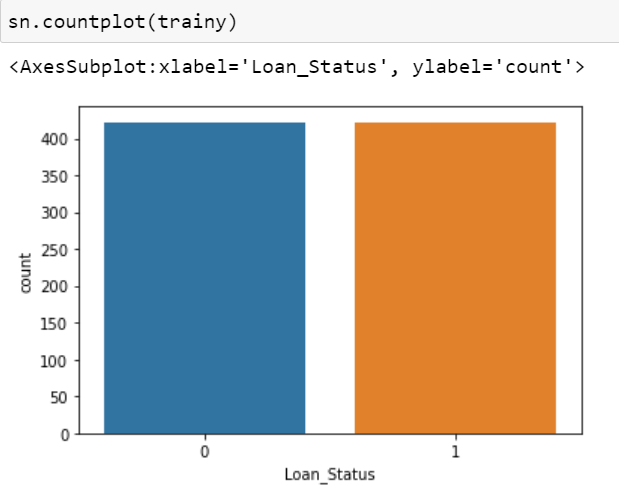


Scaled the data by using the min-max scaler method.

**Balancing**

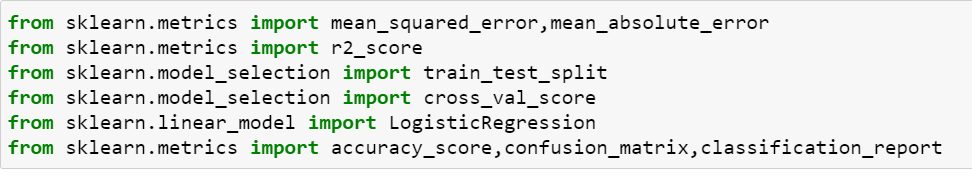
Using OverSamling\_BorderlineSMOTE



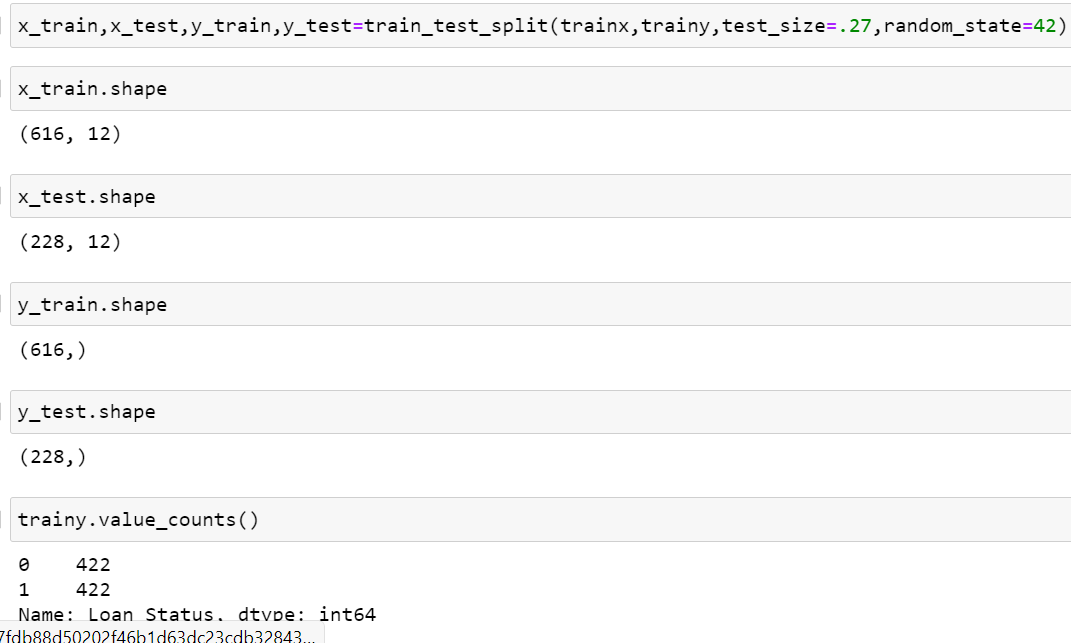


Out data is balanced not we have same number of data for both category now we can go for model building.

# Splitting the Data for Training and Testing



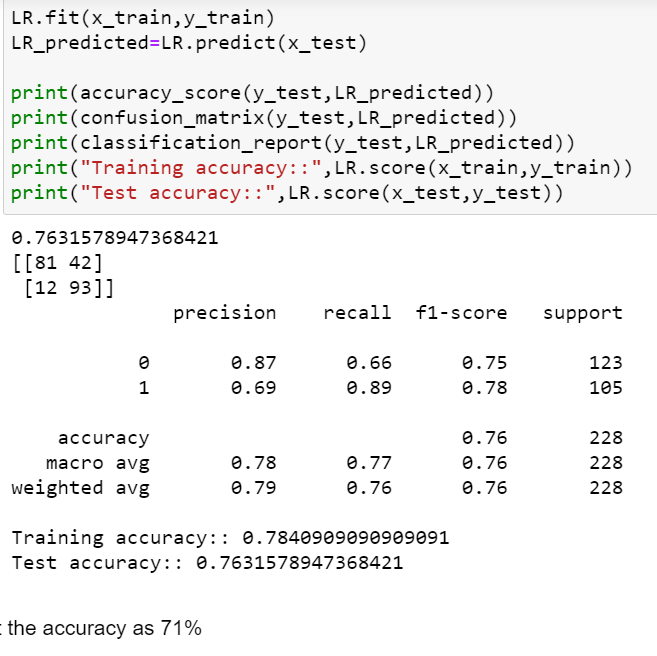
# Find the best random states



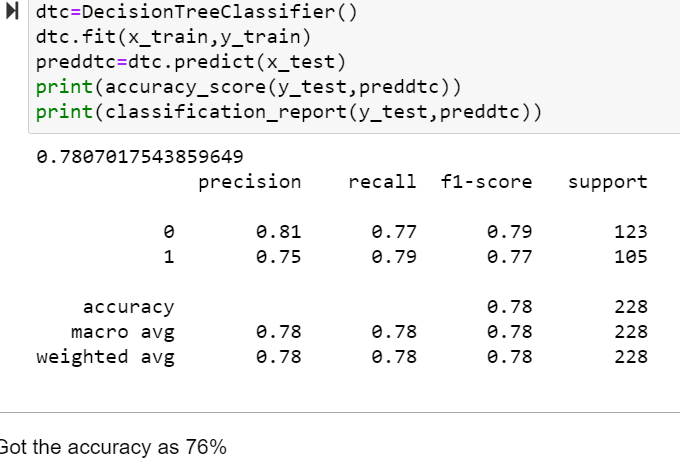
# *Model building*

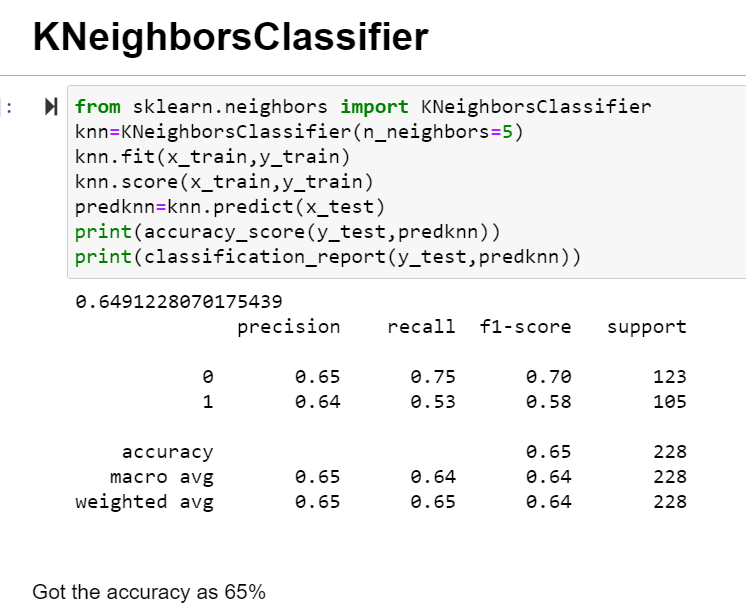
Now, after performing the train test split, we have x\_train, x\_test, y\_train & y\_test, which are required to build Machine learning models. We would build multiple classification models to get the best accuracy, confusion matrix & classification report for all the below models:

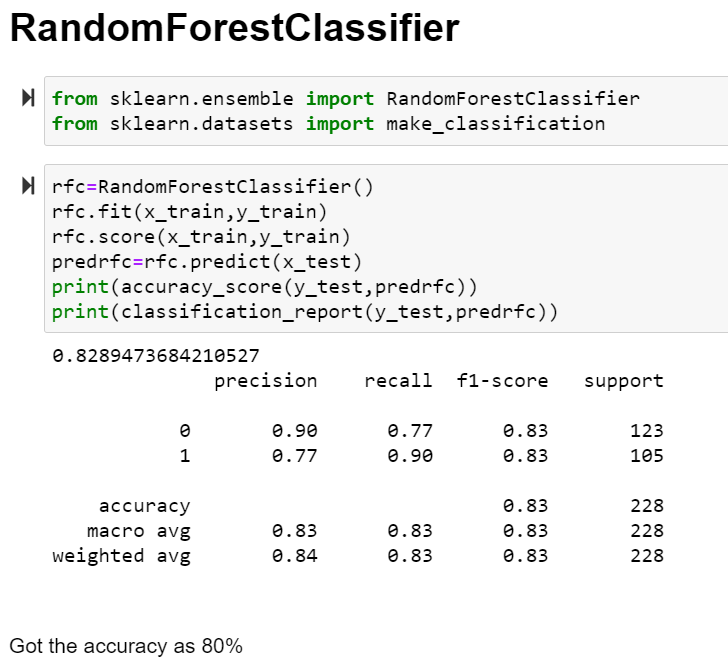
**Logistic Regression**

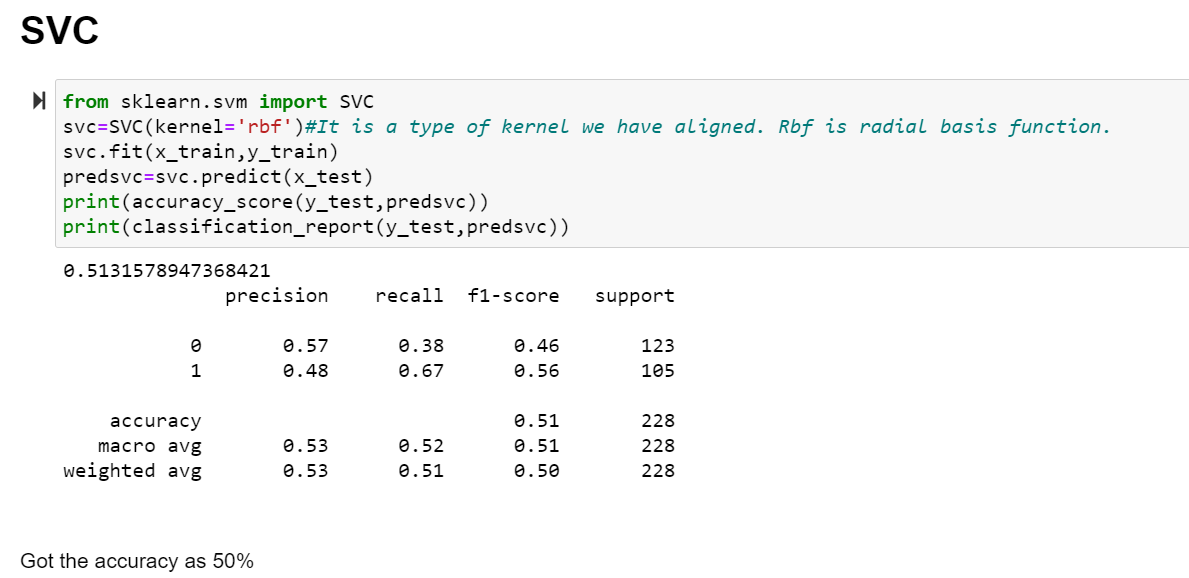


# DecisionTreeClassifier

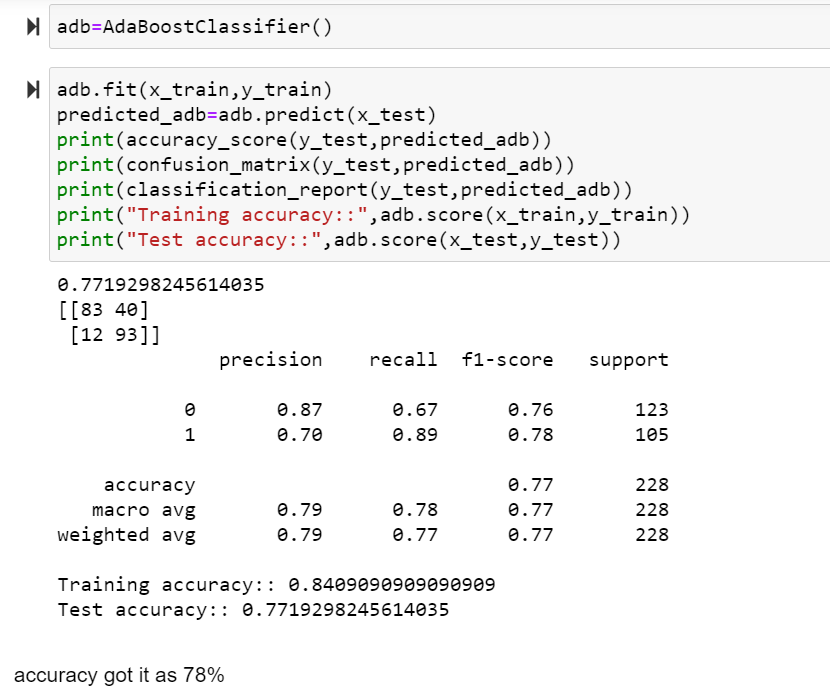




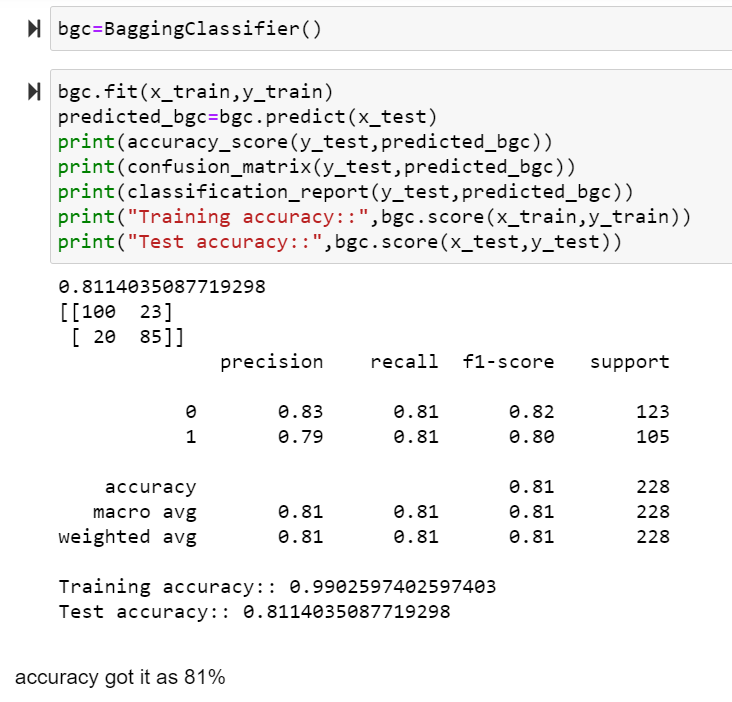




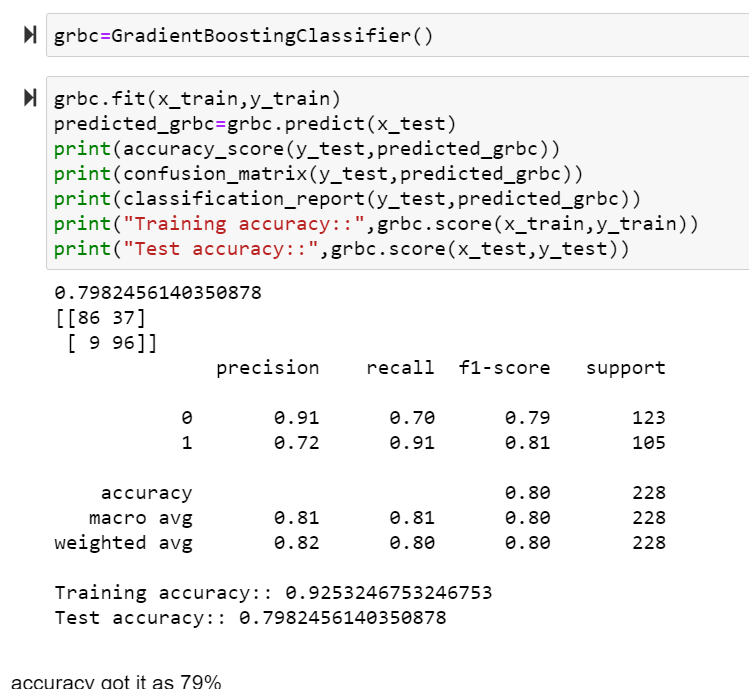
# ADA BOOST CLASSIFIER

****

**Bagging classifier**

****

**GradientBoostingClassifier**

****

# cross validation score

# Observation****:****

For every model, we have train the data (x\_train, y\_train) & predict with the help of x\_test. Now, with the help of the y\_test & prediction value, we got the accuracy score. So, out of all models, we are getting the highest accuracy for some algorithms with less type-I&II error, but it could be due to overfitting, so we need to check the cross-validation if the model is overfitted or not.

We would perform cross-validation for every model by using **sklearn.model\_selection import cross\_val\_score** & compare with the accuracy score, whichever model gives the less difference between cross-validation score & accuracy score is the best fit model.

|  |  |  |
| --- | --- | --- |
| MODEL | ACCURACY | CROSS VALIDATION |
| KNeighborsClassifier | 65 | 78 |
| DecisionTreeClassifier | 65 | 69 |
| SVC | 50 | 81 |
| RandomForestClassifier | 83 | 77 |
| ADA BOOST CLASSIFIER | 78 | 69 |
| Bagging classifier | 81 | 70 |
| GradientBoostingClassifier | 79 | 71 |

As we have seen, DecisionTreeClassifier,Gradient boosting classifier is giving less difference as compare to other models. Also, dtc, is giving the lowest difference but it is having high type-I & II errors.

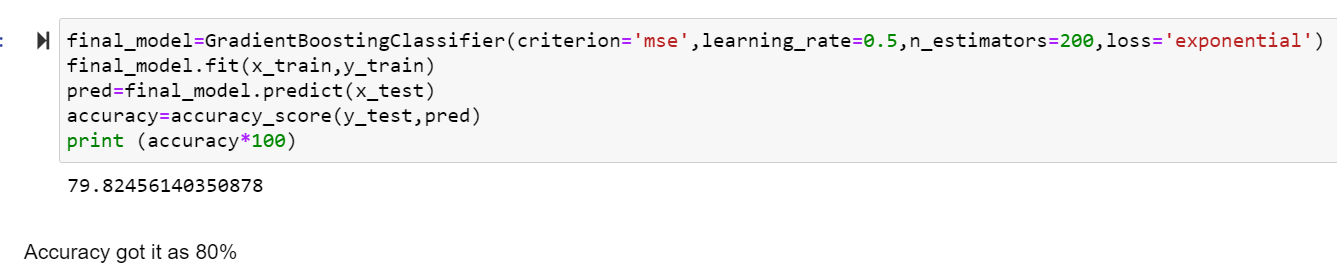
# *HYPER PARAMETER TUNING*:

To enhance the accuracy tuning the model .I have used the **GridSearchCV** grid for tuning the model so we can get the best parameters

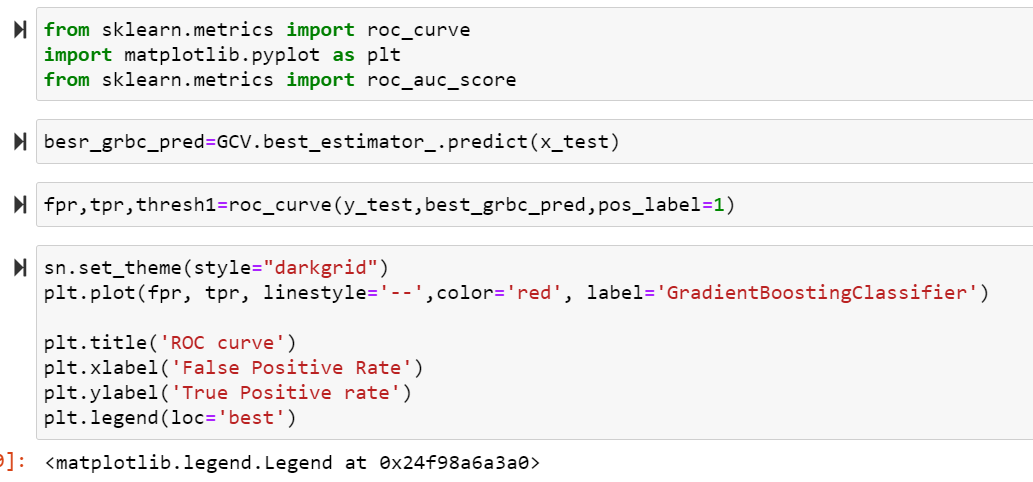
****

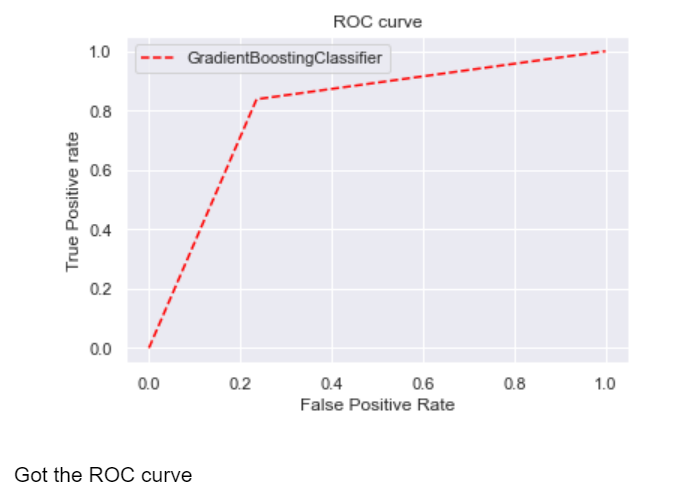
After tuning the model we got the best parameters for model

Lets train the model with best parameters



***roc\_auc\_score*:**

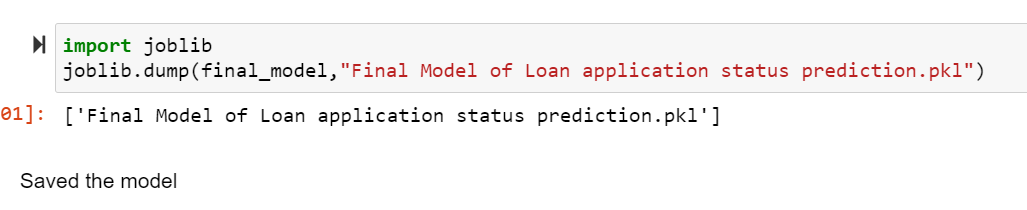




# Concluding Remarks

**Saving the model**

Saving the model: The model is ready & we have saved the model in ‘pkl’ format by using “joblib”



## *Final conclusion*:

As we have seen, the prediction is showing a similar relationship with the actual loan status from the train data set, which means the model predicted correctly & this could help banks to save time & predict which all customers could get the approval based on these features. It could also help customers to predict if they would be able to apply for a loan or not based on the bank requirements and accordingly, they would be able to apply for a loan, and by using these model both banks & customers can save time as this process of applying loan is hectic & time-consuming. Hence by using Machine learning techniques we can solve this problem & reduce manual efforts.

Key Findings and Conclusions of the Study:

* 1. There are high chances of approval for Graduate people comparing not graduate
  2. There is high chances of Loan Approved when you have credit history, people those are not having any credit history mostly getting not approved
  3. People from Semiurban area are having high chances to get their loan approved comparing people from other area.
  4. There are high chances for loan approval when you taking loan for less tenurity