##### **PREDICTING THE LOAN APPLICATION STATUS**

##### **Problem Statement:**

Now days Loans are the core business of banks. The profit comes directly from the Loans

Interest. The loan companies given a loan after an intensive process of verification and Validation. However, they still don’t have assurance if the applicant is able to repay the Loan or not. Banks wants to automate the loan eligibility process on real time based on customer Details provided while filling online application form. These details are Gender, Marital Status Education, Number of Dependents, income, Loan-Amount, Credit History and others.

In this project, the dataset includes details of applicants who have applied for loan. The dataset includes details like credit history, loan amount, their income, dependents etc.  We must build a model that can predict whether the loan of the applicant will be approved or not based on the details provided in the dataset.

* **Data Analysis:**

Source:: <https://github.com/mygithub9319/Datatrained-Project-Evaluation-Phase>

In the dataset having 12 independent variables and 1 target variable, I.e Loan status in the training dataset .and we have 614 rows and 13 columns in the train dataset. The column names are following.

* **Independent Variables:**

Load\_ID

Gender

Married

Dependents

Education

Self-Employed

ApplicantIncome

CoapplicantIncome

Loan\_Amount

Loan\_Amount\_Term

Credit History

Property\_Area

Dependent Variable (Target Variable):

Loan\_Status- Y/N (approved/not approved)

There are 3 types of data format:

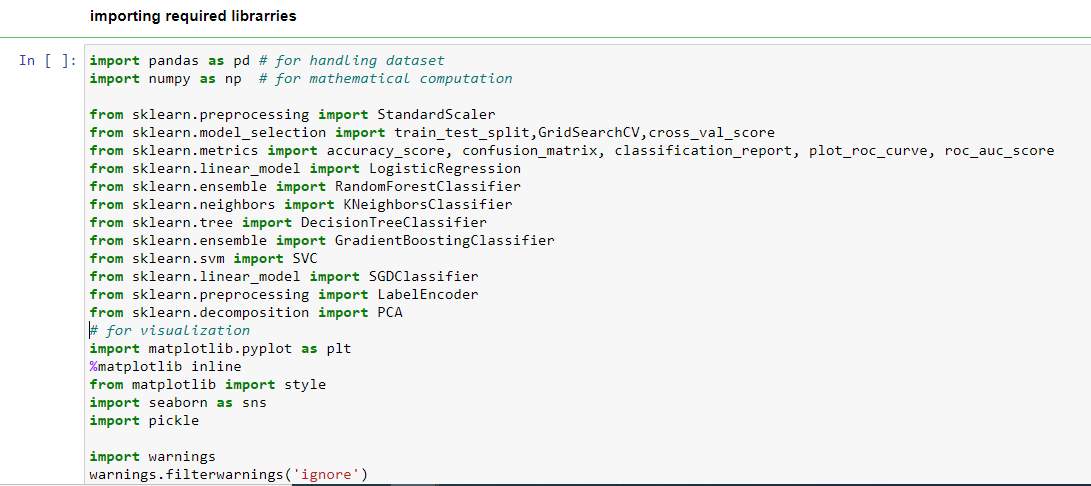
Object: Object format means variables are categorical. Categorical variables in our dataset are Load\_id

Gender, Married, Dependents, Education, Self\_Employed, Property\_Area, Loan\_Status

Int64: it represents the integer variables. ApplicantIncome is this format.

Float64: it represents the variable that has some decimal values involved. They are also numerical

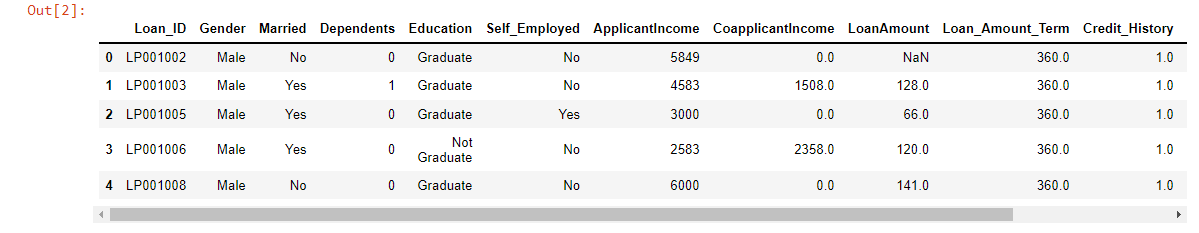
* **Importing the libraries:**



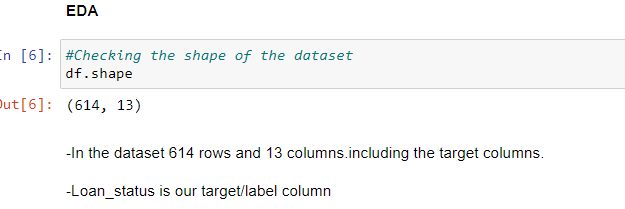
* **Getting the data:**



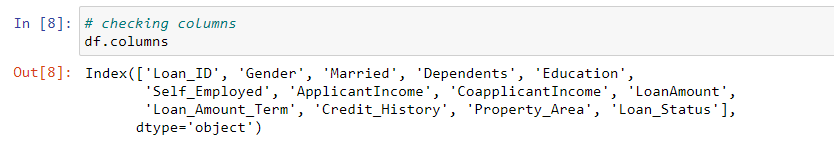
* **Data Analysis:**
* **Checking the first 5 rows of the dataset**



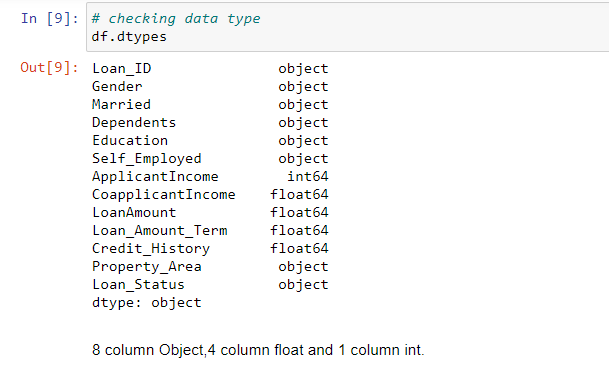
* **Checking the shape of the dataset**



* **Checking the columns of the dataset**

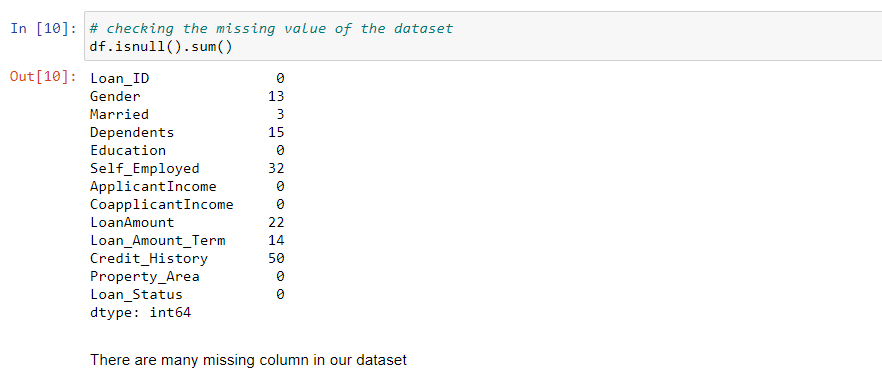


* **Checking the datatypes of the dataset**



There are 8 object type features which we must encode in our future steps so that machine can understand.

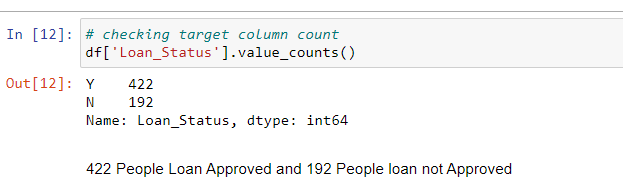
* **Checking for null values in the dataset**



There are null values present in our dataset which we must work on in our future steps to make our data clean and ready for machine building process.

Now, we will start cleaning our data in our next steps so that we can build an efficient machine learning model.

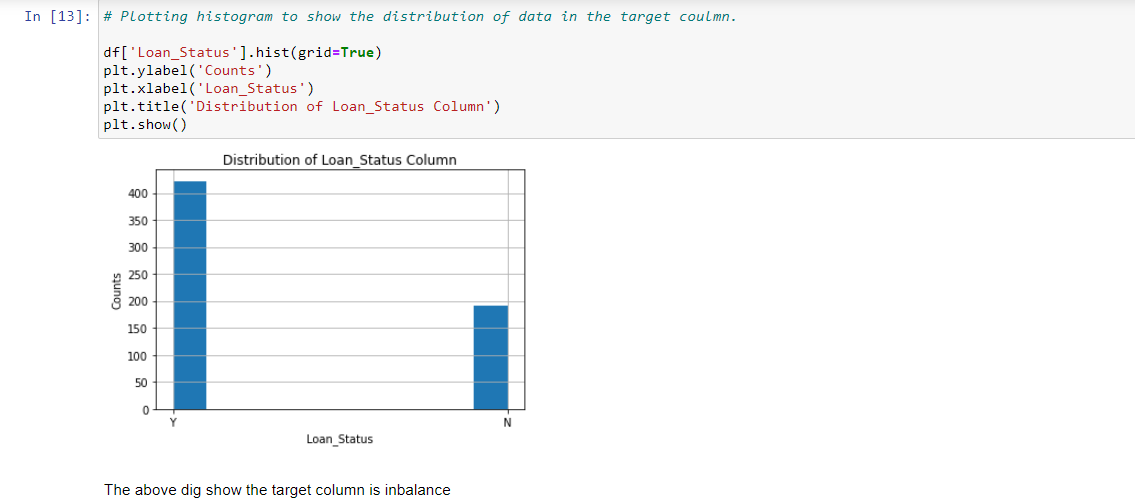
* **Analyzing the target column**



In the Loan status Y Stand for Loan approved and N Stand for Loan reject. In the above snap 422 People Loan approved and 192

People Loan not approved.

* **Plotting histogram to show the distribution of data in the target column.**

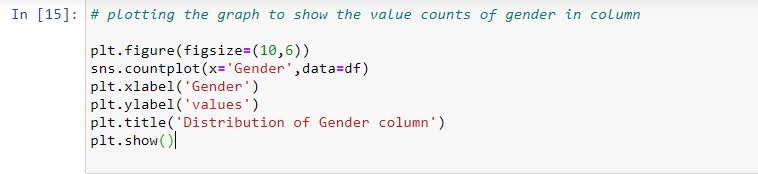


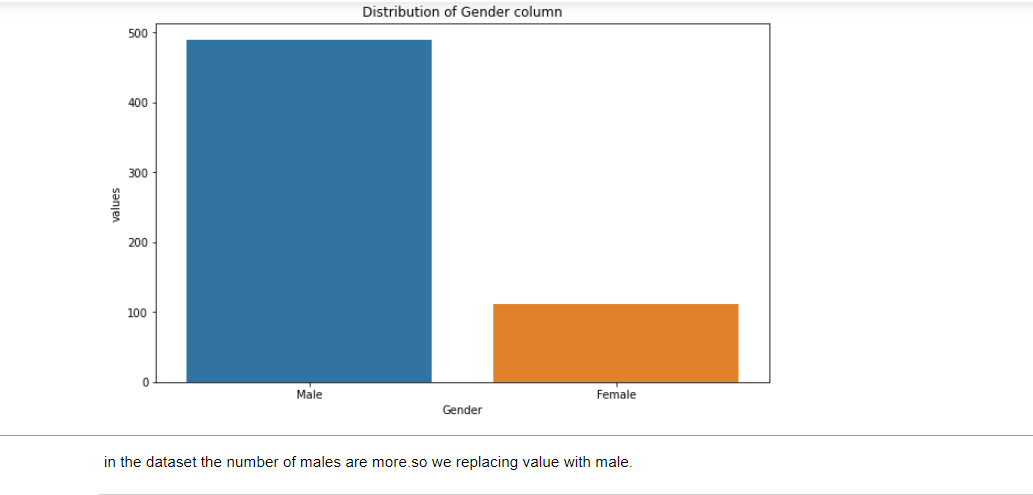
In the Above screenshot show Loan status

Now working on missing values.

* **Gender**

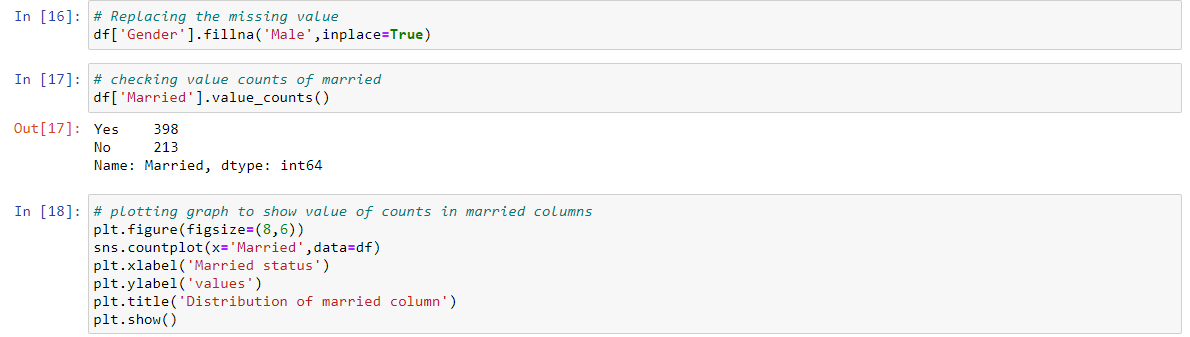


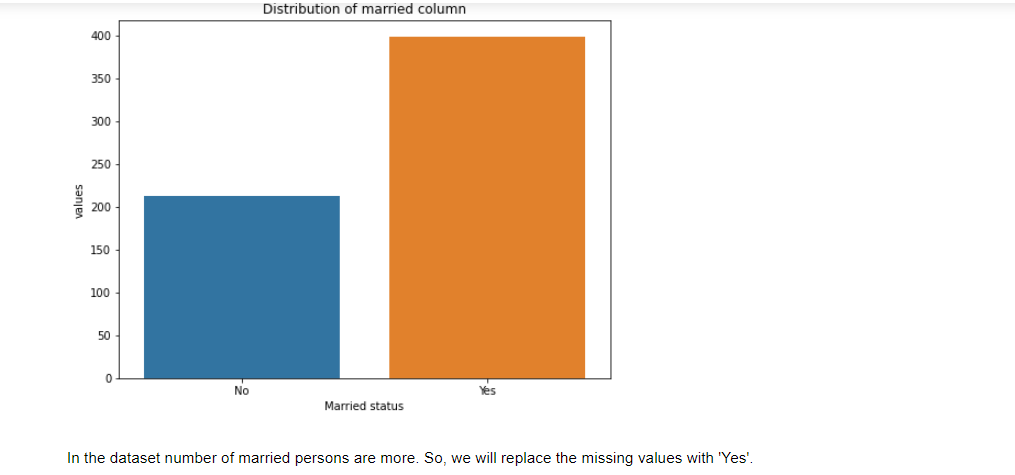




In the dataset the number of males are more.so we replacing value with male.

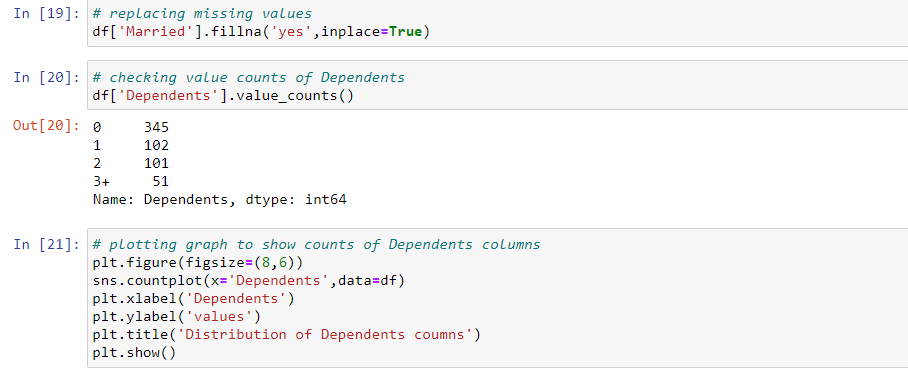
* **Married**

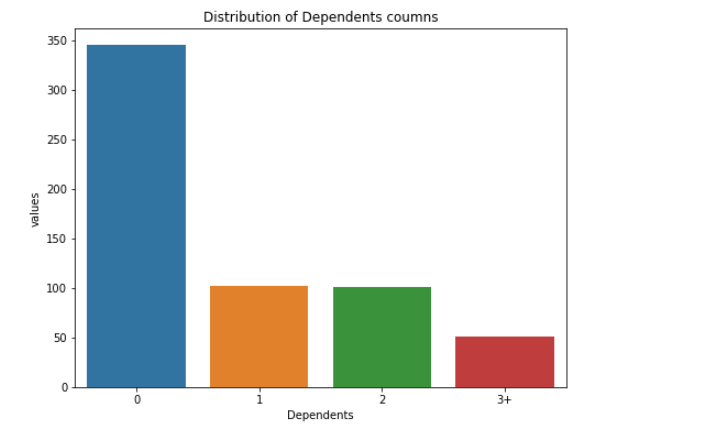




In the dataset numbers of married persons are more. So, we will replace the missing values with 'Yes'

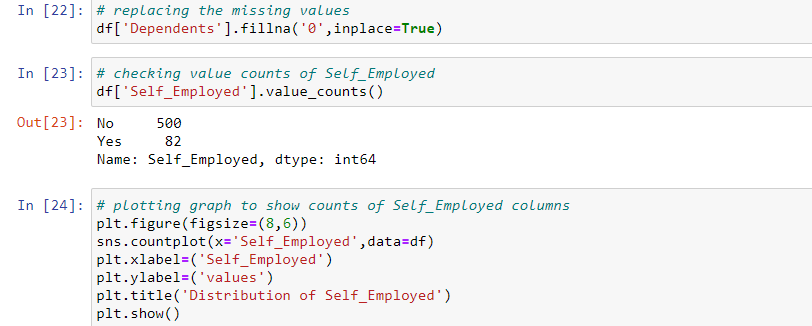
* **Dependents**

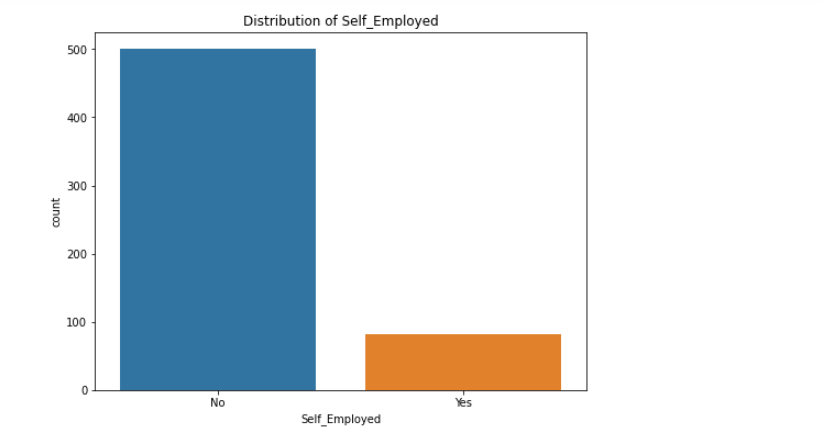




In the dataset 0 for most of the applicants, so we'll replace the missing values as 0 dependents.

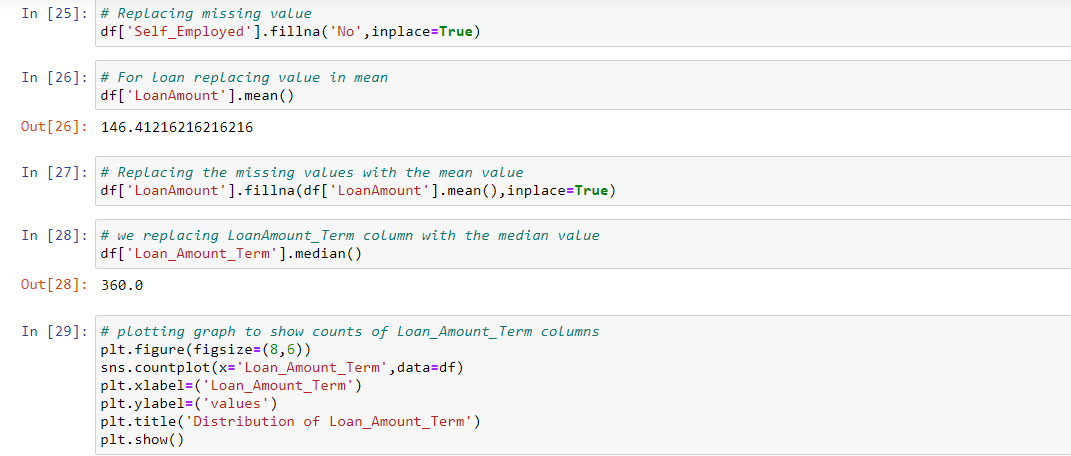
* **Self\_Employed**

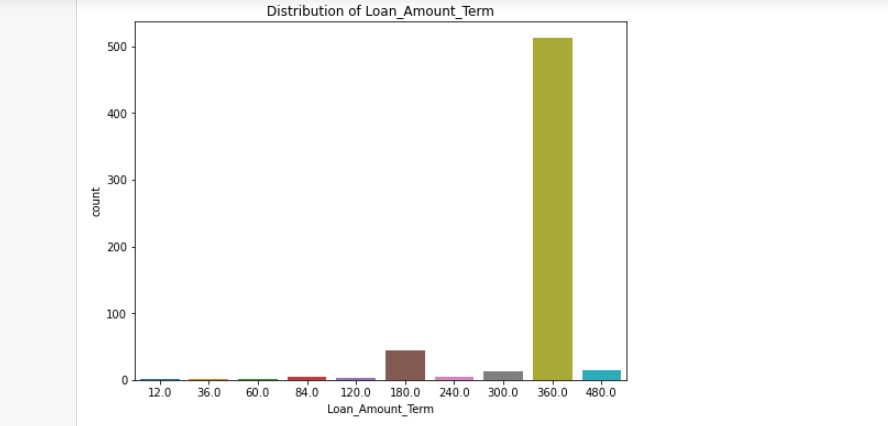




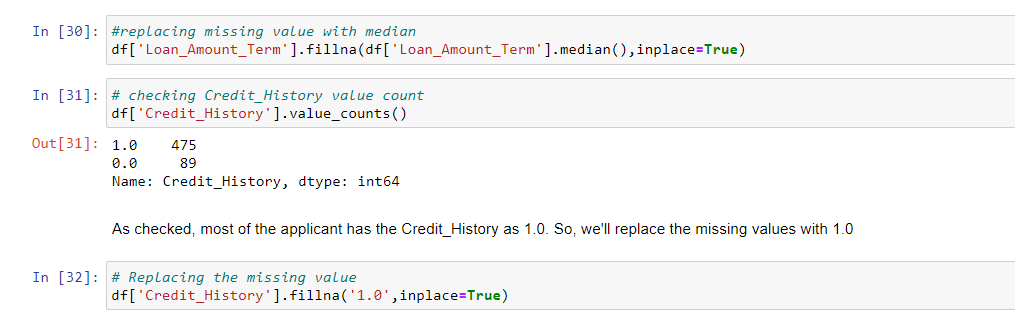
In the dataset the number of self-employed applicants is less, so we'll replace the missing values with 'No'

* **LoanAmount**





In the above snap Loan Amount missing value replaced by median

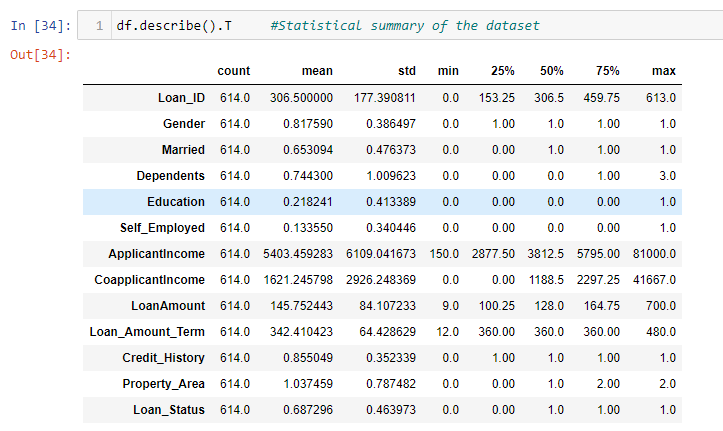


* **Now let’s check if any null values are left again to remove.**

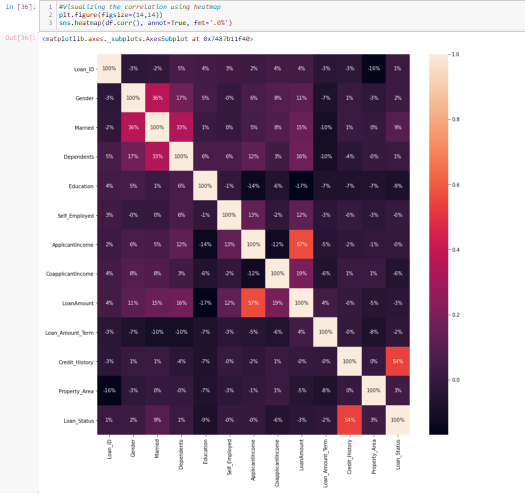


From the above screenshot, I can say that null values are not present in my dataset now.

Now I will now go through the statistical summary of the dataset. I can draw some conclusions looking at the mean and standard deviation whether any outliers are present in the dataset or not. I think there are some outliers in the data and some skewed data present. I will look deep into it in my future steps.

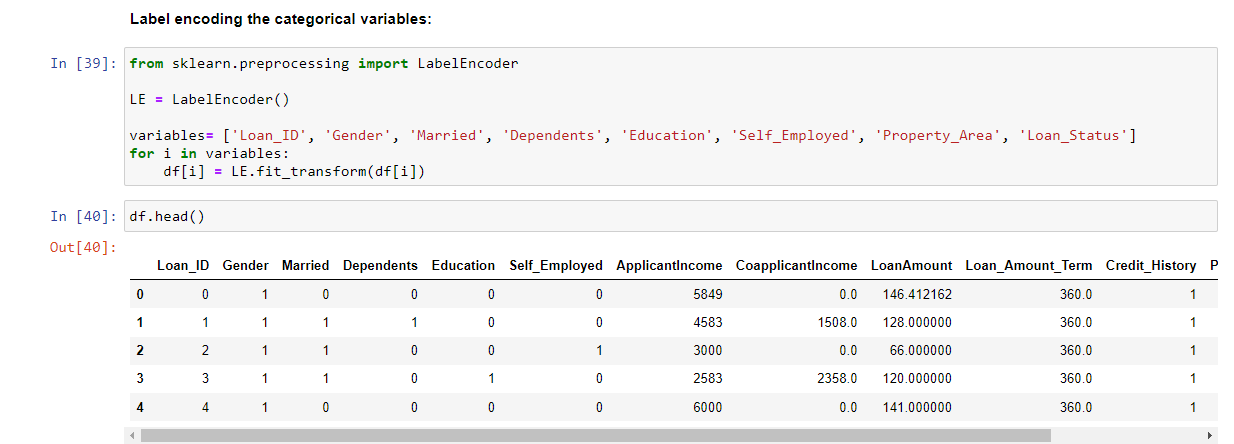


Let’s visualize how my data are correlated with each other and with the target variable, i.e. Loan\_Status.

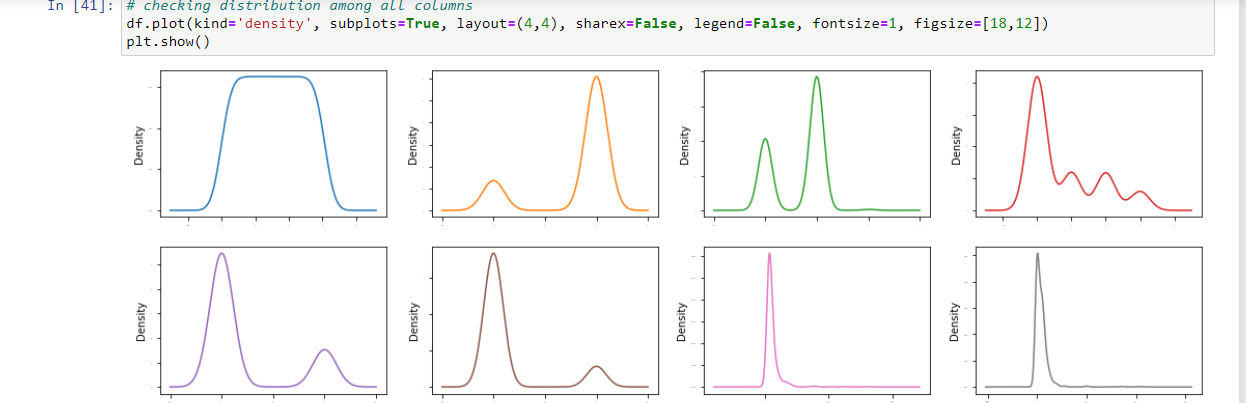


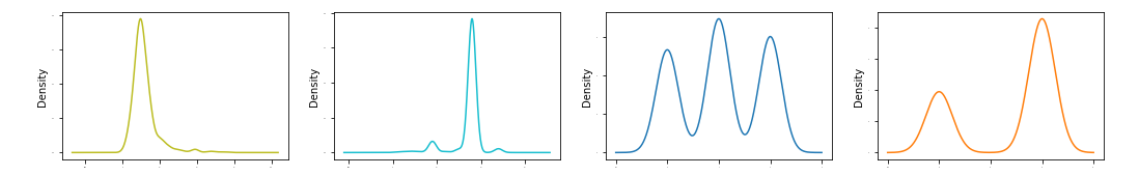
From the heatmap, I can say that the feature ‘Credit\_History’ is highly correlated with the target variable ‘Loan\_Status’.

Now encoding the categorical columns using LabelEncoder



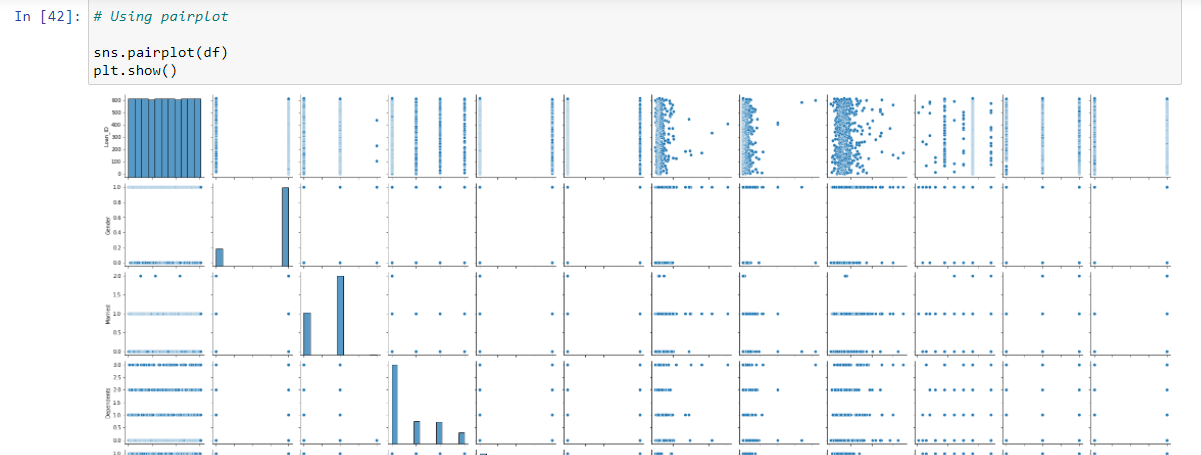
* **Checking Distribution along with all columns**



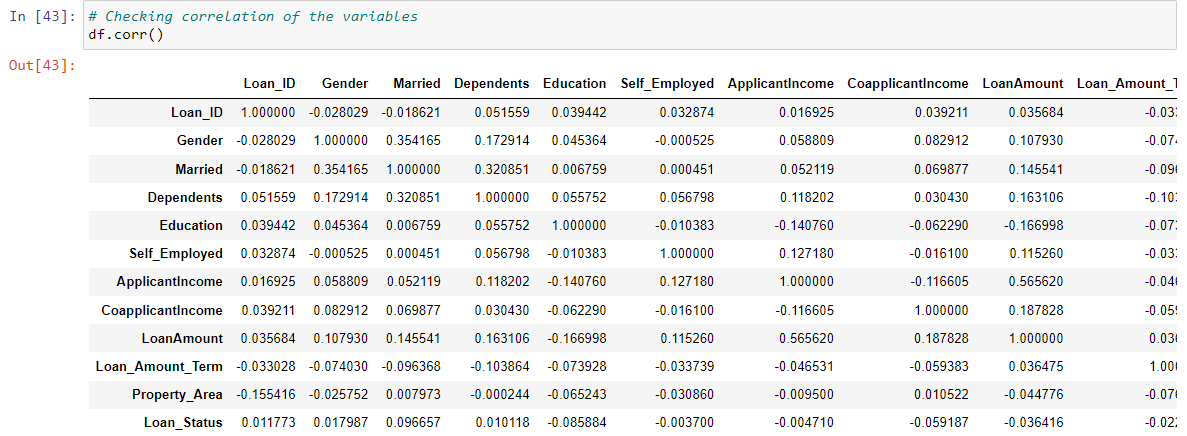


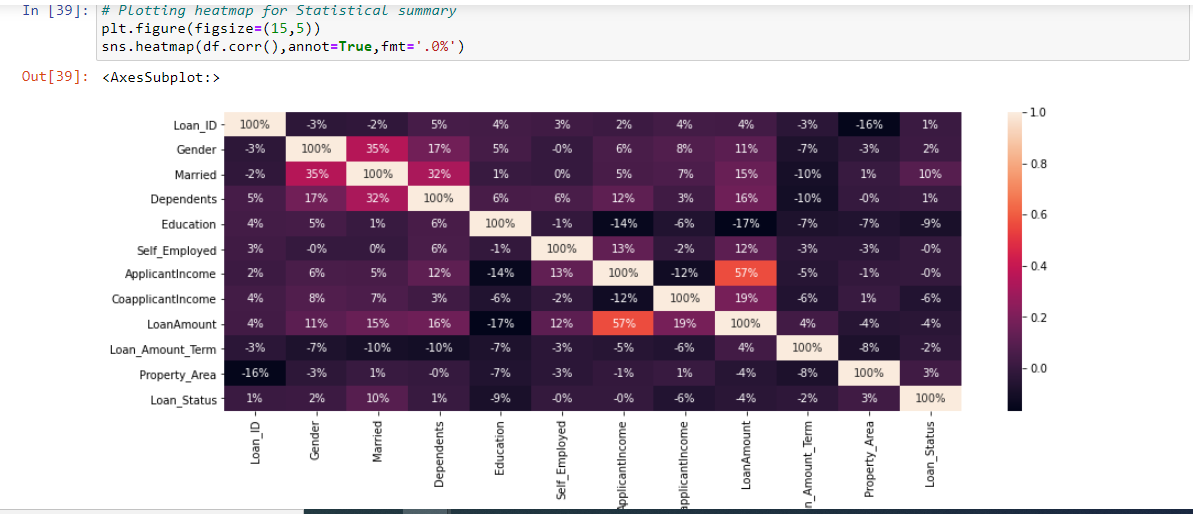
In the above snap all columns are skewness present

* **Using Pair plot to show the distribution**



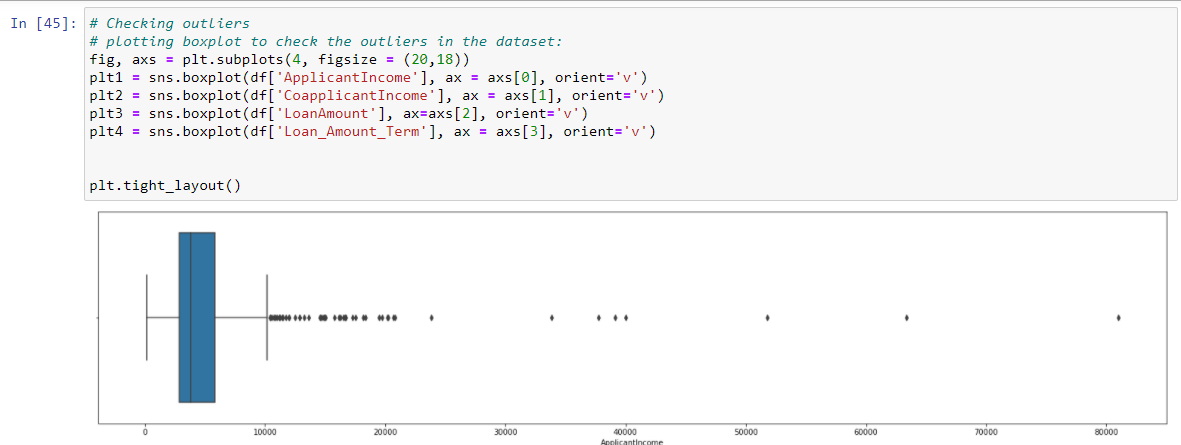
* **Checking correlation of the variables.**

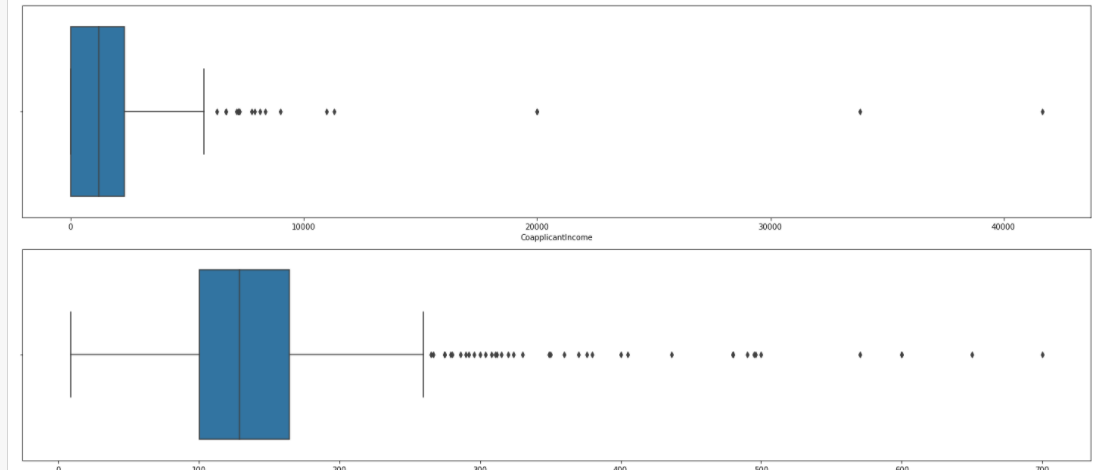
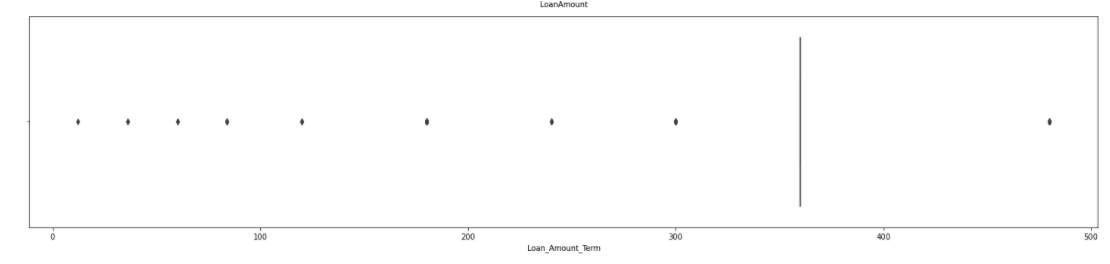




The relation between Applicantincome vs loanamount is high.and there is no any relationship between loan status vs Applicantincome and also no any relationship between loan status vs Self employed

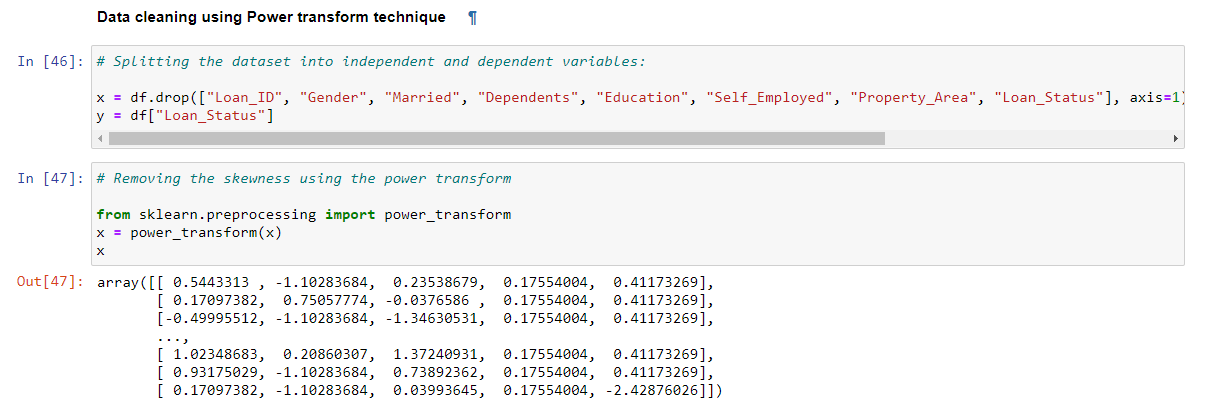
* **Checking Outlier using boxplot.**



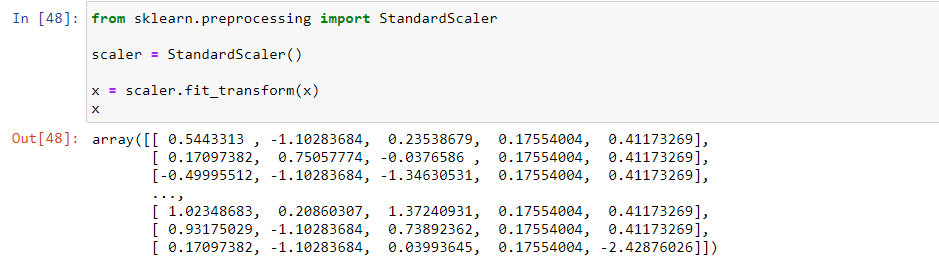


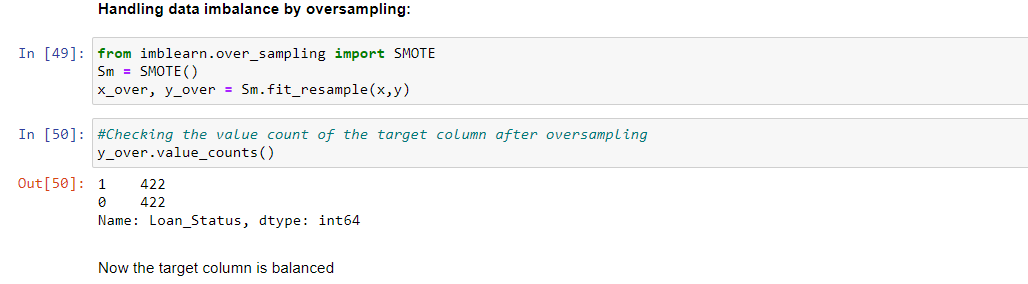
Still outlier present in our dataset

* **Now working on data cleaning using Power transform technique**



* **Scaling the data using StandardScaler:**





* **EDA concluding remark:**

After going through data analysis and data processing, I can conclude that the raw data that I have received is now cleaned and is ready for Machine Learning process. Steps that I follow are listed below:

1. At the beginning, I have analysed the data by checking its shape, its datatypes and information regarding presence of null values.
2. After checking for null values, I have seen that there are some null values present. I removed them using Labelencoder method.
3. While checking for datatypes, I have seen many object type columns are present, I have encoded them into numeric so that Machine can understand. The encoding process that I have used for categorical type data is LabelEncoder and for continuous type data I have used OrdinalEncoder.
4. I have visualized the data using dist plot and barplots.
5. Then I have checked the statistical summary of the dataset and checked the correlation between the features and the target variable using heatmap.
6. I have removed the outliers present in the dataset using Zscore method and minimised the skewness in data using ‘log1p’ method.
7. When my DataFrame gets ready for Machine Learning process, I have split the independent variables and target variable into x and y.
8. Then I scaled my data into a standard form using StandardScaler

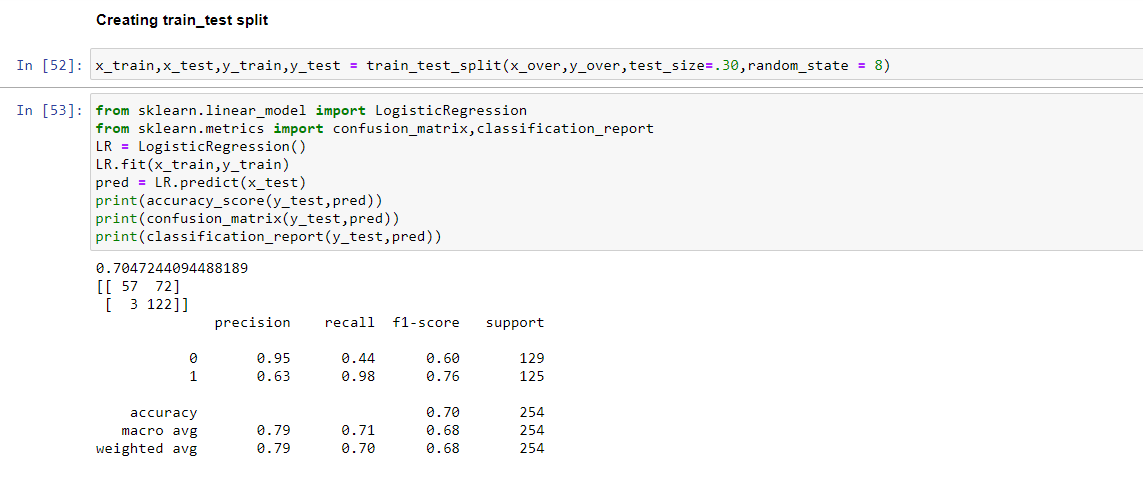
Now Building Machine Learning Models:

* **Finding best Randomstate.**



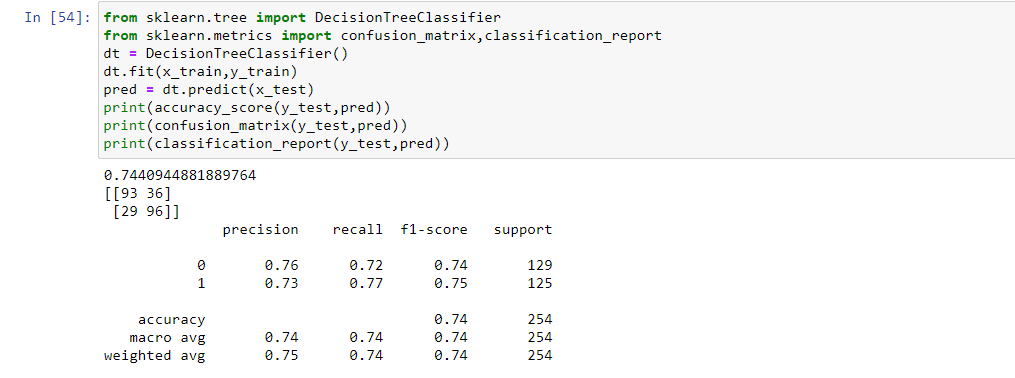
We have found best random state values as 8 we will create our train\_test\_split this random state(8)

* **Now creating train\_test\_split with above random\_state And find accuracy using LogisticRegression**



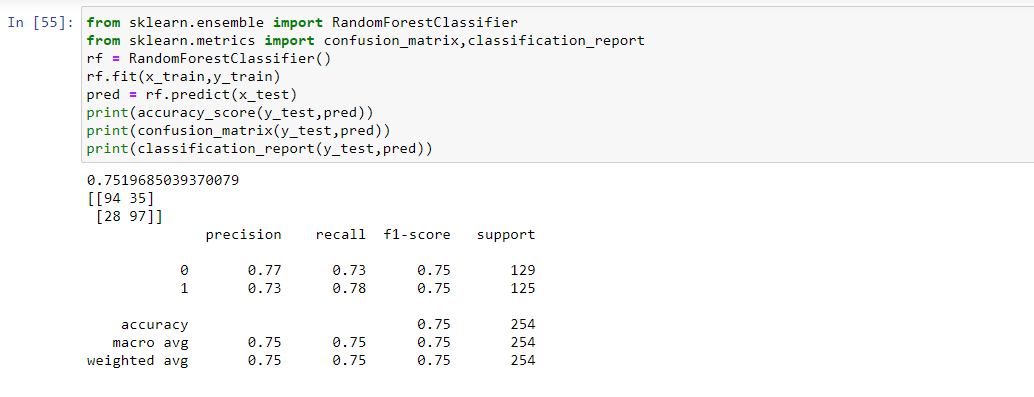
LogisticRegression Accuracy is 71%

* **find accuracy using DecisionTreeClassifier**



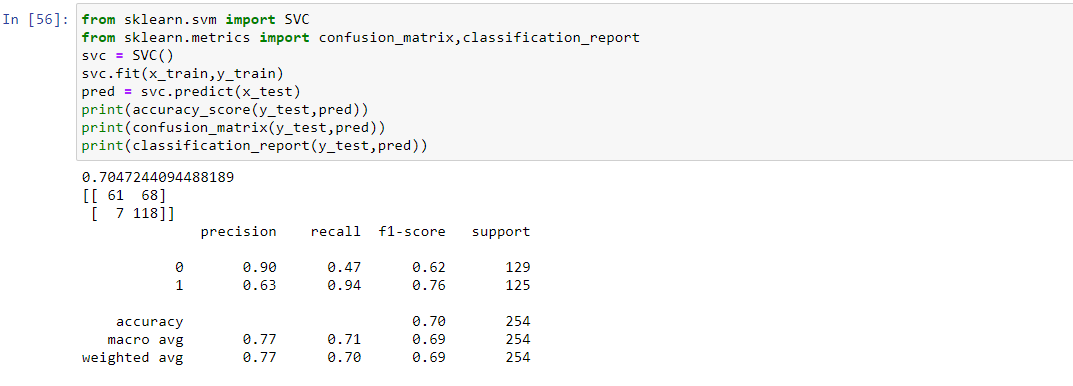
DecisionTreeClassifier Accuracy is 74%

* **find accuracy using RandomForestClassifier**



RandomForestClassifier Accuracy is 75%

* **find accuracy using SVC**



SVC Accuracy is 71%

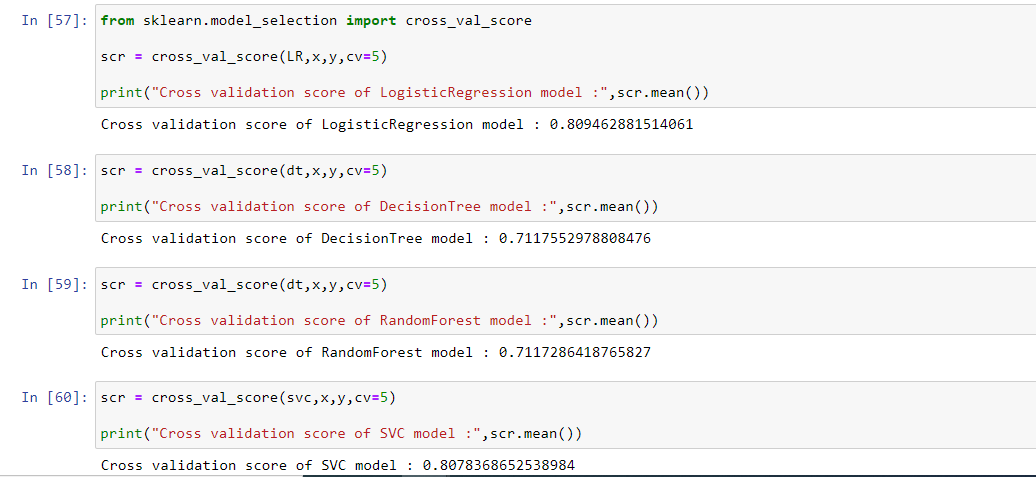
As we can see, RandomForestClassifier is giving a good accuracy of 74% among all the classification models at a random state 8. But, let us check, how RandomForestClassifier performs when we use cross validation.

* **K Fold Cross-Validation:**

K-Fold Cross Validation randomly splits the training data into **K subsets called folds**. Let’s image we would split our data into 4 folds (K = 5). Our model would be trained and evaluated 5 times, using a different fold for evaluation every time, while it would be trained on the remaining 4 folds.

The image below shows the process, using 4 folds (K = 4). Every row represents one training + evaluation process. In the first row, the model gets trained on the first, second and third subset and evaluated on the fourth. In the second row, the model gets trained on the second, third and fourth subset and evaluated on the first. K-Fold Cross Validation repeats this process till every fold acted once as an evaluation fold.

* **Cross validation to check the score**



After cross validation we check RandomForestClassifier is our best model..

* **Hyperparameter tuning:**

A Machine Learning model is defined as a mathematical model with several parameters that need to be learned from the data. By training a model with existing data, we can fit the model parameters. However, there is another kind of parameters, known as Hyperparameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

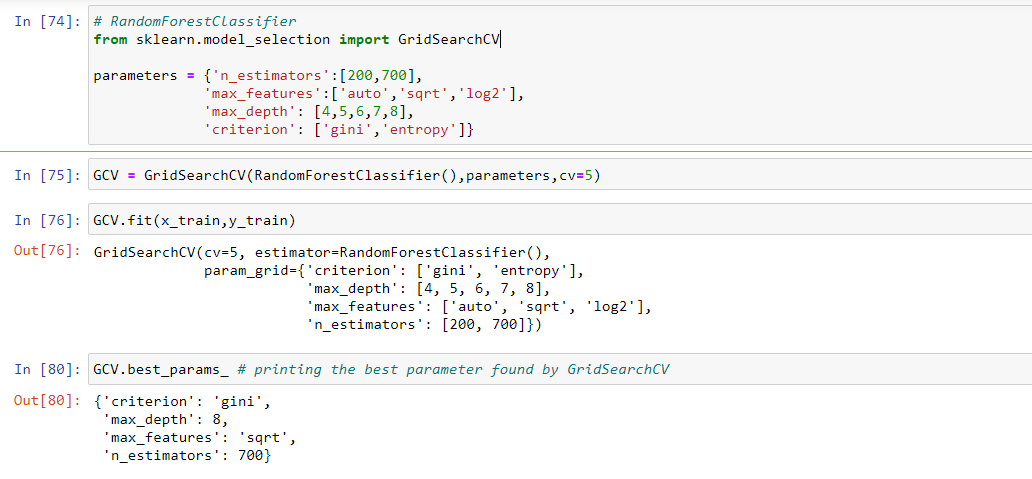
Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. Two best strategies for Hyperparameter tuning are:

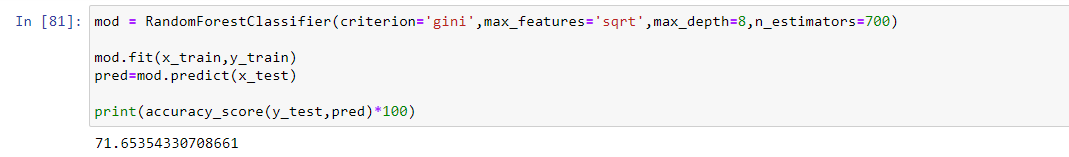
* **GridSearchCV:**

In GridSearchCV approach, machine learning model is evaluated for a range of hyperparameter values. This approach is called GridSearchCV, because it searches for best set of hyperparameters from a grid of hyperparameters values.

* **RandomizedSearchCV:**

RandomizedSearchCV solves the drawbacks of GridSearchCV, as it goes through only a fixed number of hyperparameter settings. It moves within the grid in random fashion to find the best set hyperparameters. This approach reduces unnecessary computation.

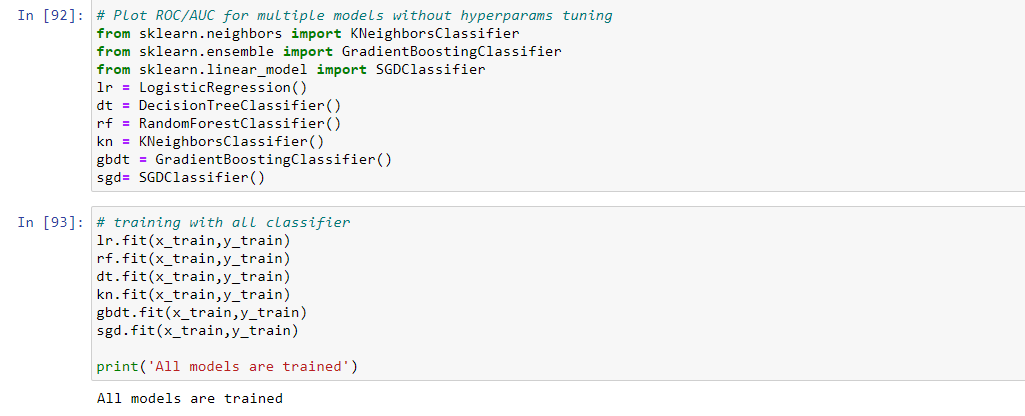


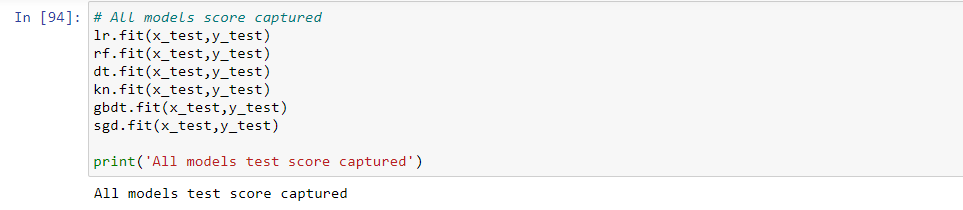


The accuracy after hyperparameter tuning is also 71.6%.best model is RandomForestClassifier

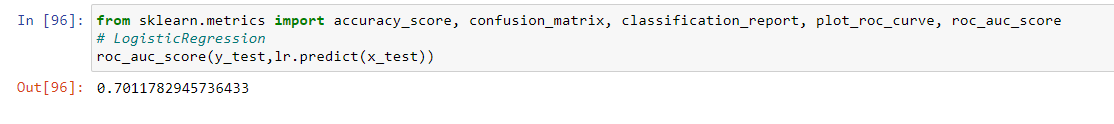
* **ROC AUC Curve:**

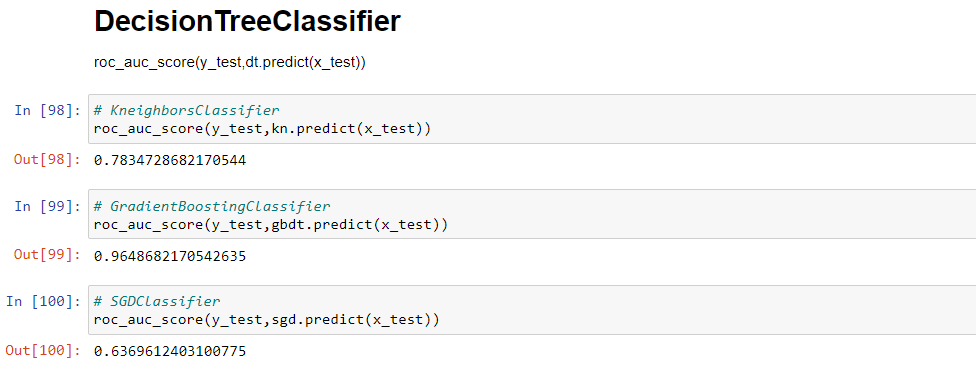
Another way to evaluate and compare our binary classifier is provided by the ROC AUC Curve. This curve plots the true positive rate (also called recall) against the false positive rate (ratio of incorrectly classified negative instances).

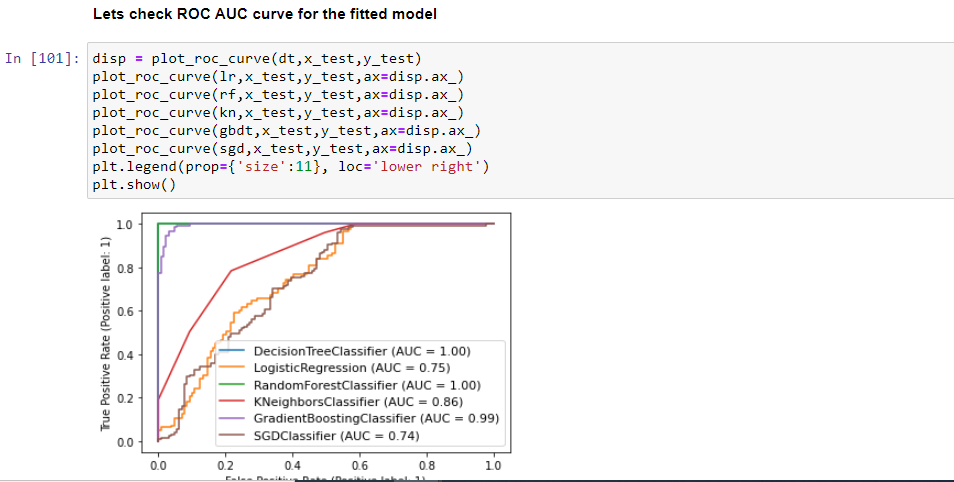




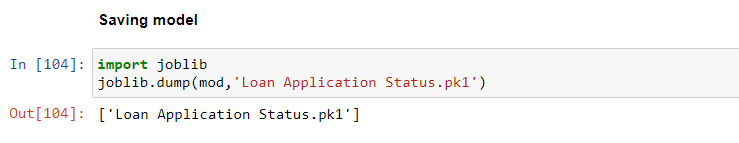
### Lets find ROC AUC score







* **Now Saving the model**



* **Conclusion:**

We started with the data exploration where we got a feeling for the dataset, checked about missing data, and learned which features are important. During this process we used seaborn and matplotlib to do the visualizations. During the data pre-processing part, we converted features into numeric ones,grouped values into categories and computed missing values. Afterwards we started training 5 different machine learning models, picked one of them (LogisticRegression) and applied cross validation on it. Then we discussed how LogisticRegression works, took a look at the importance it assigns to the different features and tuned its performance through optimizing its hyperparameter values