Loan Eligibility Prediction Using Machine Learning

**Problem definition:**

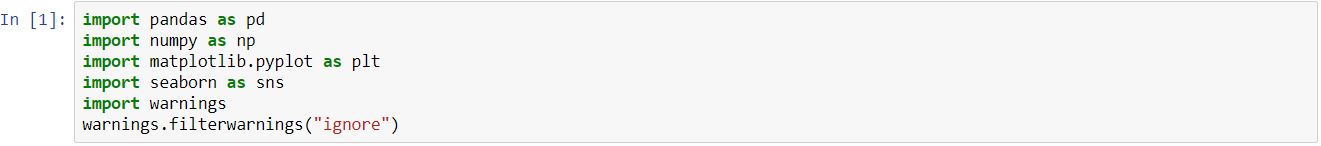
Aim is to automate the Loan Eligibility Prediction process from the given customer details by EDA and Machine Learning Models

Given Details:

|  |
| --- |
| Loan ID |
| Gender |
| Married |
| Dependents |
| Education |
| Self Employed |
| Applicant Income |
| Co applicant Income |
| Loan Amount |
| Loan Amount Term |
| Credit History |
| Property Area |
| Loan Status (Target Variable) |

**Import libraries:**

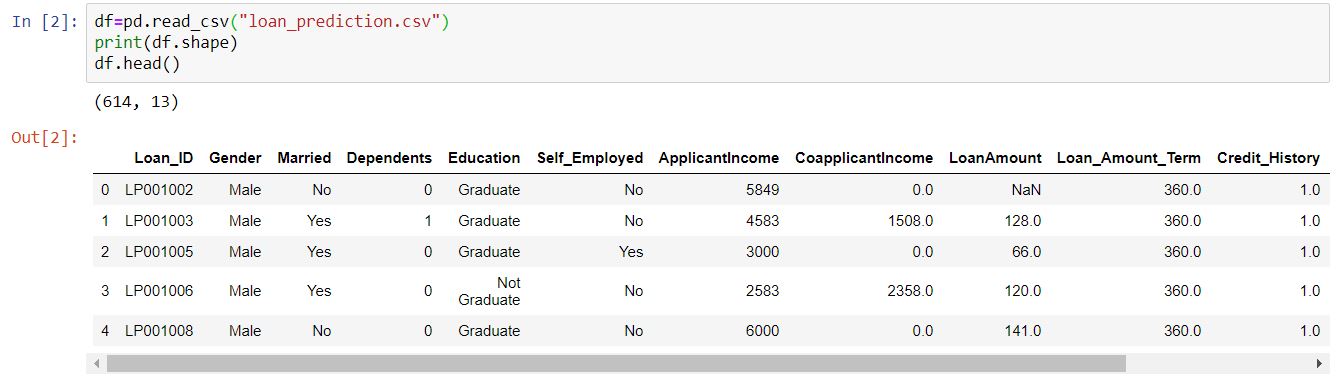
Have to import the basic required libraries for Exploratory Data Analysis process



Here Pandas and numpy are libraries used for data processing, matplotlib and seaborn are visualization library used to visual the given data graphically in a easy understandable way.

**Reading Data:**

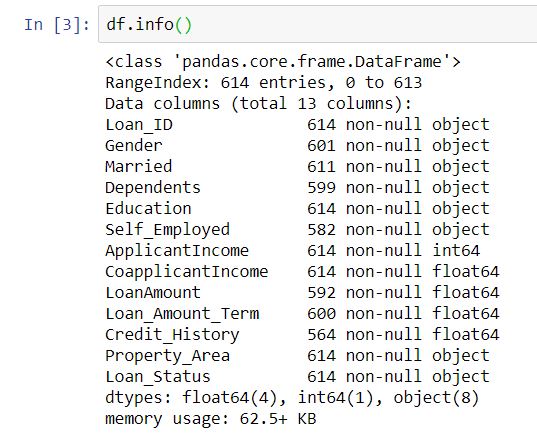
To load the data by using pandas

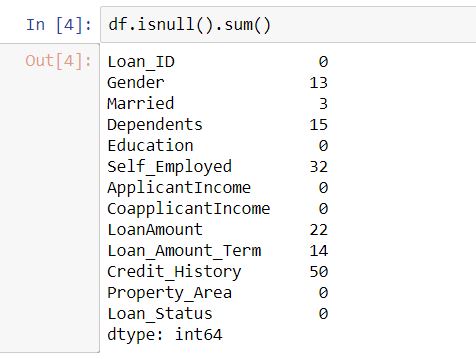


Above it shows that shape of whole dataset and it’s top 5 rows.

**Exploratory Data Analysis:**

To know the data type and check the presence of null value can done by



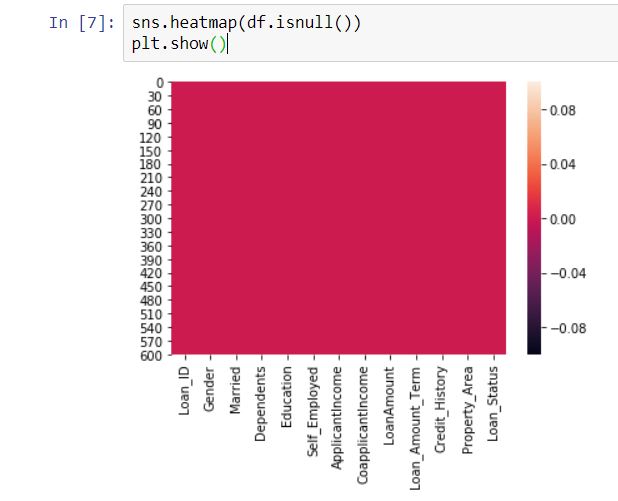
It show the count of null values in respective columns 

**Treating of null values**:

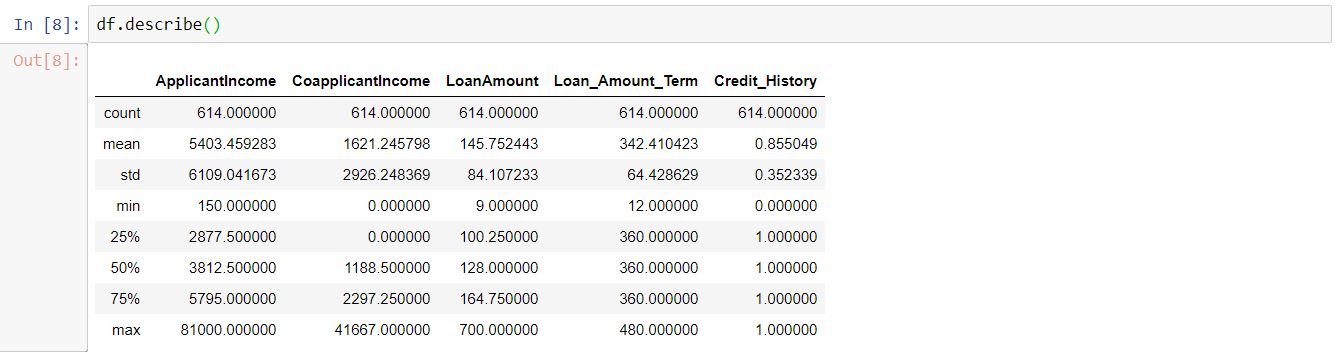
Replace the null values with mean or median (but median is better than mean because of presence of outliers) for unique numerical data and for categorical data it can be fill by mode operation or also can use drop option (but not recommended).



To check whether null value presence is there are not can again done by above format (df.isnull().sum()) or by Graphical method



**Describe the numerical data:**



The above figure shows that presence of outliers in Applicant income, Co applicant Income, Loan Amount Term because it have huge difference between variation in (min,25%,50%,75%) and max(maximum number)

If the standard deviation is 1 then that column will not use for prediction because it won’t affect or make impact in model

It also show presence of data is skewed

**Drop the unwanted columns:**

8.JPG

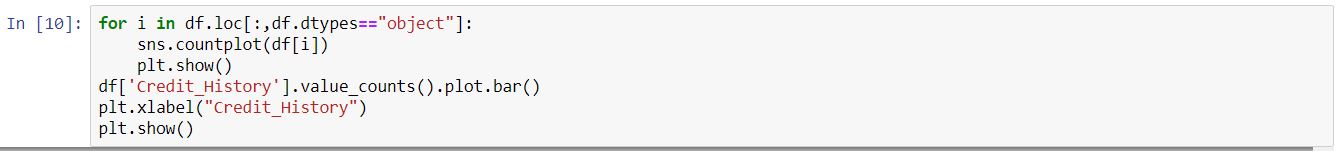
It is a ID number all the data is unique

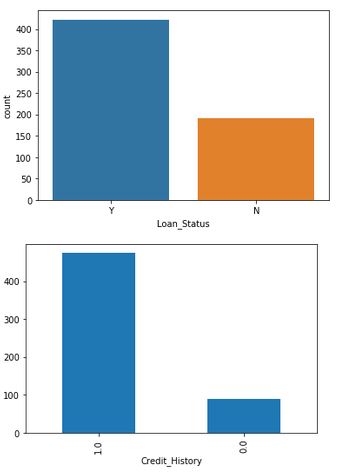
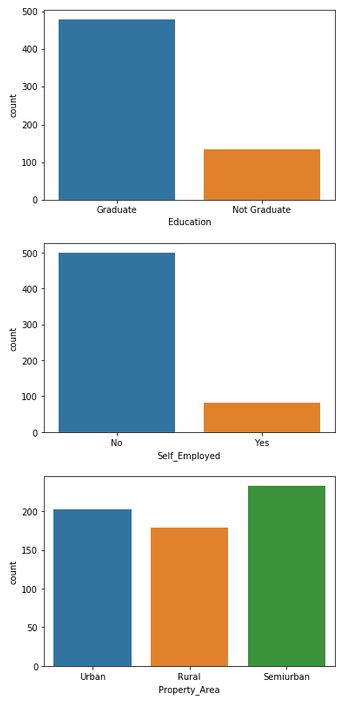
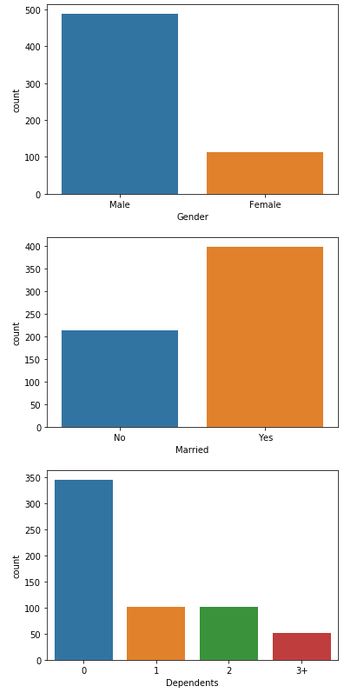
**Visualization:**

Visualize the given data Graphically is to make a quick decision about the data

**Categorical data:**

**Uni- variant analysis:**





It shows that

In Gender column only male and female are present there it also shows male who apply for loan is greater than female.

In Education column no of count is high for Graduate person when compare with Not Graduate.

Married persons who apply for loan is higher than unmarried persons.

Self Employed persons are less in applied count for loan.

The person with no dependents had applied for loan the most.

Loan application is equally came from all the three property area (Urban, Rural, Semi urban).

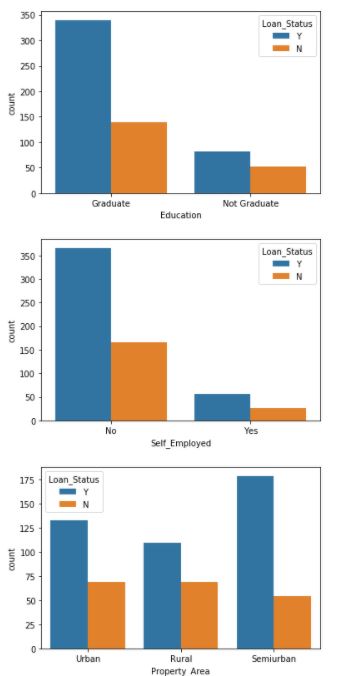
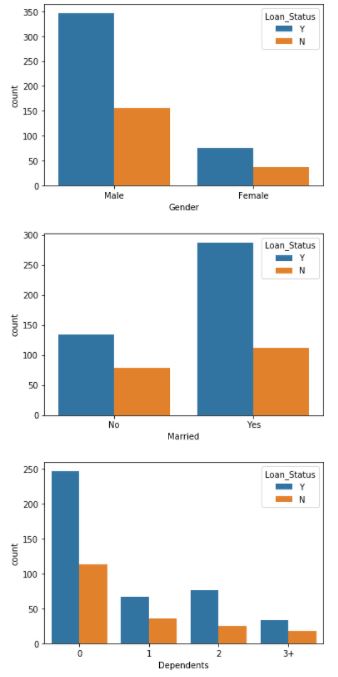
Most of persons apply for loan had good Credit History (1) only few have bad Credit History (0).

When Compare the Loan Eligibility is Yes or Not Eligibility count is high for Yes.

**Bi variant analysis:**

Compare the object data type with loan Status:



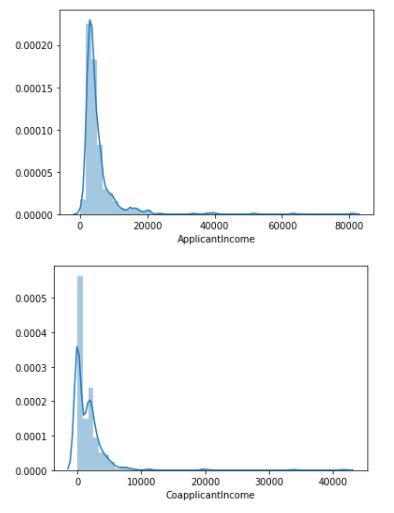


Here it detailed that count for eligibility based on category of data

In all category Loan eligibility percentage is high than not eligibility percentage

To visualize the distribution of numerical data by





Most of Loan applicant Income is from 0 to 20000

Co applicant income is mostly from 0 to 10000

**Encode the data:**

To convert object data into integer data

Before apply the Machine Learning model we have to check required data is in integer, if not we have to convert it as integer data, because the model will only consider integer data as their input and output data to train and test the data.

Encoding technique can help to convert it as a required type.



Encode can be done by either label encoder or one hot encoder these are available in scikit learn library.

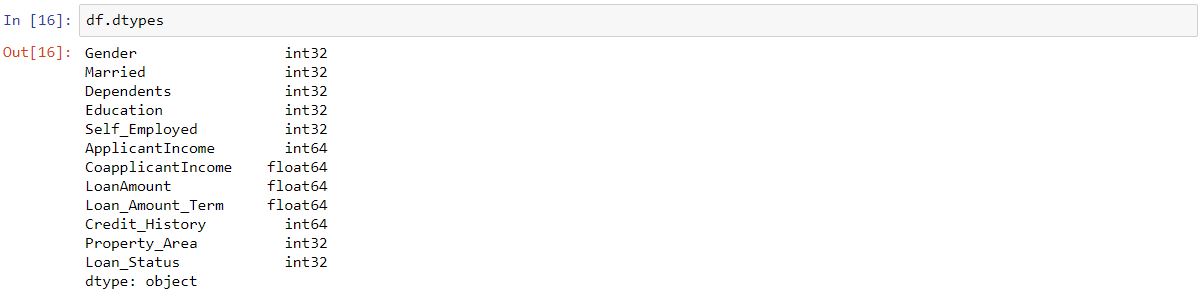
One Hot Encoder is for to encode the categorical data column with only two possible answers (yes or no) or (True or False).

Label Encoder is applicable for the object data with two and more variety of data in a column.

Here I used Label Encoder because some of columns like Dependents and Property Area have more than two types of data.

There One Hot Encoder is can’t applicable.

To check all the columns are converted to integer data:



Here all the data type of columns are showing either integer or float , there are no presence of object data, So it will be a good data for to apply in model.

Finding of correlation between data present in dataset:

Correlation is can either positive correlation or negative correlation

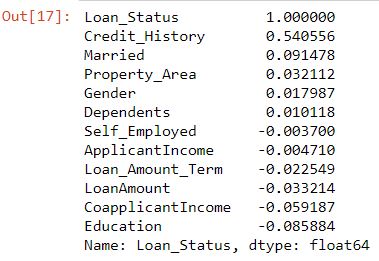
Positive Correlation - (0 to 1)

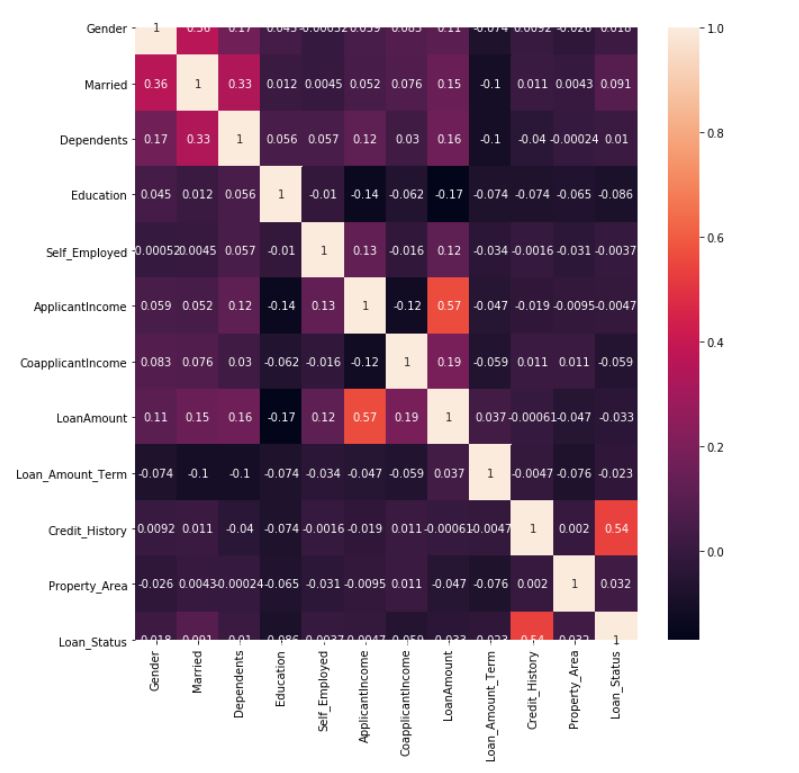
Negative Correlation – (0 to -1)

If the correlation value is nearer to 1 or -1 this data will give great impact in values to the corresponding data.

Or if the correlation value is nearer to 0 this data will not impact in values to the corresponding data.







It shows if Credit History is good positive correlation it leads to loan status either Yes or No, Education is giving good negative impact on Loan Status

That states that if Credit History decease Loan possibility will decrease in rate.

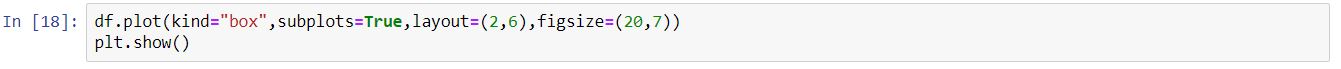
**Outliers Mining:**

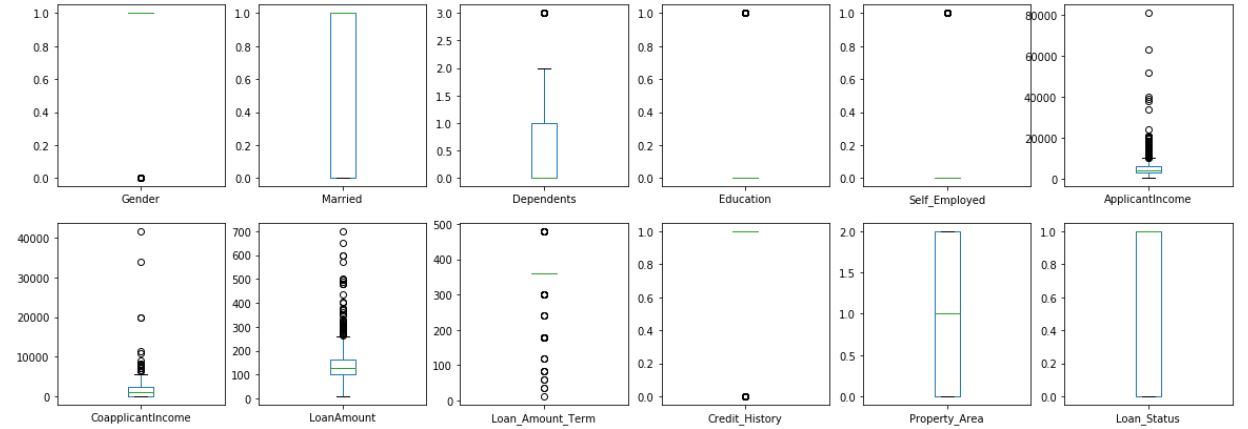
Outliers are presence of unwanted data present in dataset, it also may be a fake or by typing mistake.

To remove the presence of outliers we have apply Z-scores on the data, before that we have to find the presence of outliers in the given data.

**Outliers Identification:**

It can be able to done by box plot.





Circles present on top of maximum value and bottom of minimum value in given data will be consider as outlier

Here it showed that there is present of outliers.

**Outliers Removal:**

It is going to done by Z-Score

Z-score measures how far the data points are present away from mean of the data, and also there will be a presence of threshold value(-3 to +3)

If the distance between data points and mean of data is above that fixed value it will be consider as a outliers



26.JPG

There is occur of reduction in row size, behalf of applying Z-score

**Skew value Identifier:**

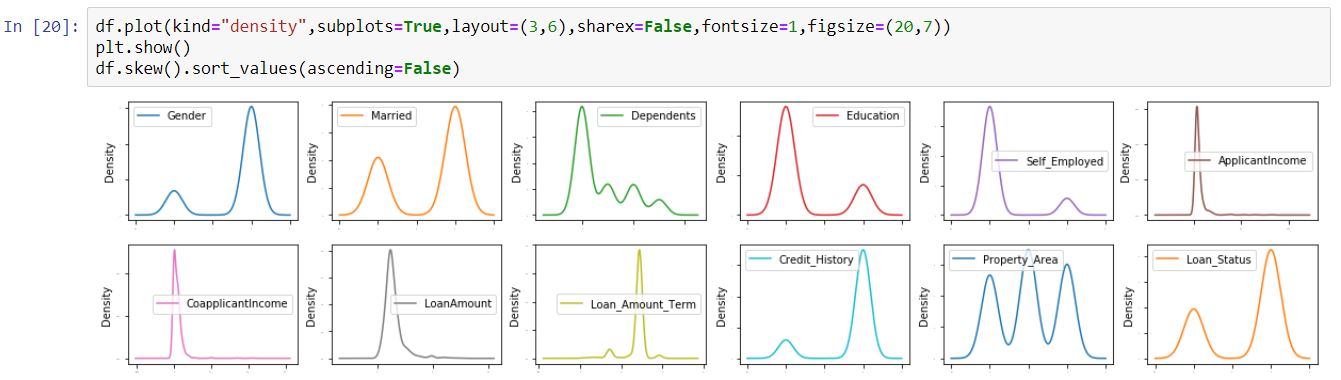
Skewed data refers as it is not like a Normal Distribution (Standard Deviation = 1), there curve will be asymmetric that is curve may be left or right skewed

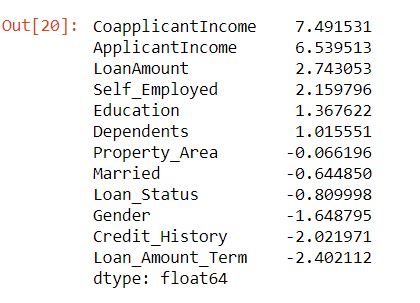
Left Skewed = Negative

Right Skewed = Positive

Normal Distribution = Zero Skewed value

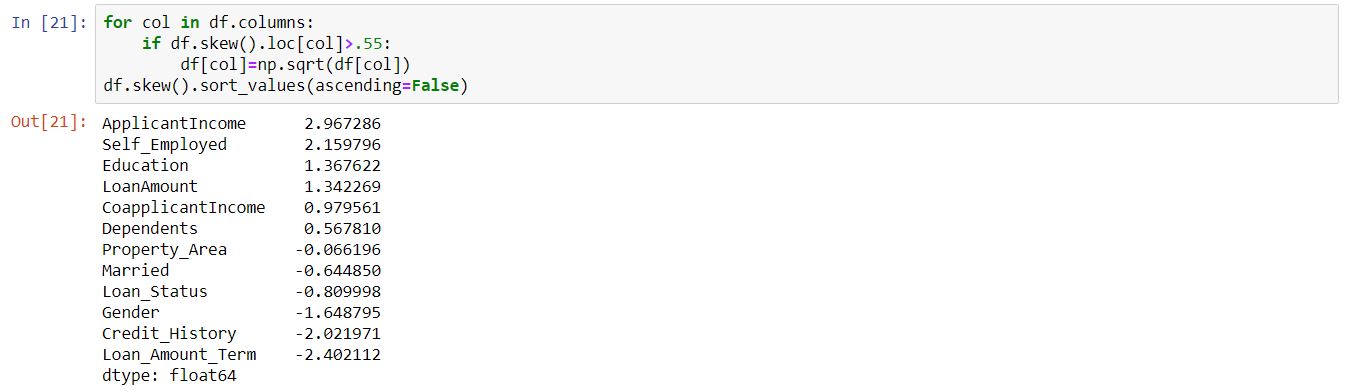
If the data is right skewed or left skewed it will affect the model accuracy, else if it is Normal Distribution data it will good for model to predict





For to get a good normal curve the skew value should between (-0.55 to +0.55)

**Reduction of skew values:**



There some reduction of skew values is occur because of applying Square root transformation

**EDA Concluding Remark:**

There occur of removal of null values and shrinking and expand of required columns are occur.

Some unwanted columns are get removed.

All columns data are converted as integer type.

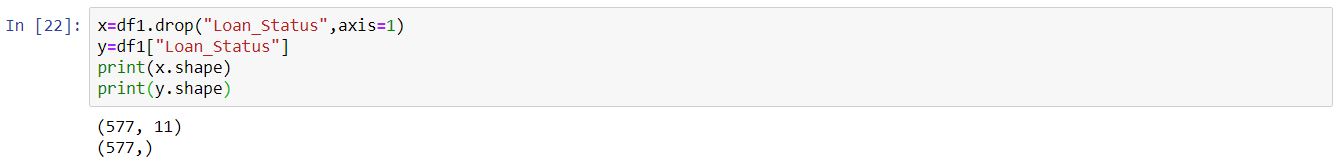
Removal of outliers is done.

Skew values are get reduced.

Checking of correlation of target variable is done.

**Split data process:**

Before building Models separate train (input data) and test data (output data)



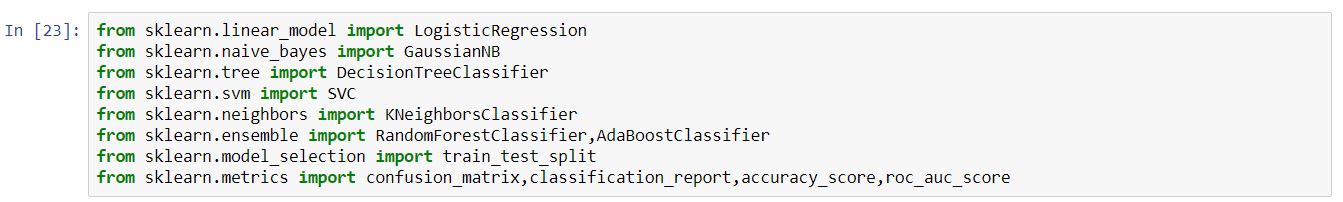
Except Loan Status every data columns will be consider as a input data

Loan Status column will consider as output data

**Building Machine Learning Models:**

**Model Import :**

Import the required model to apply in prediction process



These models are not inbuilt in python so we have to import it from sklearn (Scikit Learn)

Models are stored in particular repository and these repositories are have to import from sklearn

Here the given output data have only two outcome values 1 or 0, So it will be consider as classification data model

Model comes under Classification are

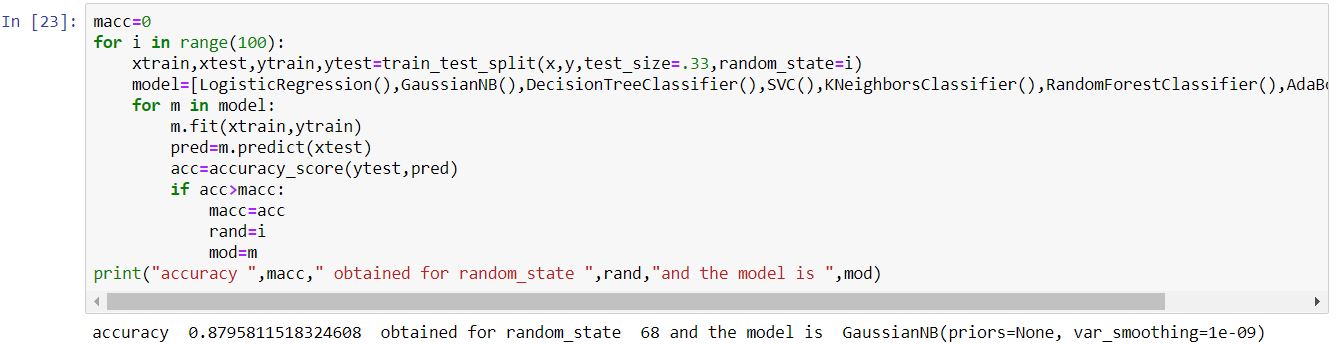
1. LogisticRegression
2. GaussianNB
3. DecisionTreeClassifier
4. SVC
5. KNeighborsClassifier
6. RandomForestClassifier
7. AdaBoostClassifier

In metrics repository

1. confusion\_matrix is for to check the amount of perfectly predicted value and wrongly predicted value by ( True Positive ,True Negative ,False Positive ,False Negative ).
2. classification\_report is for to check the precision, recall, f1-score .
3. accuracy\_score is for to check the accuracy score of the prediction value.
4. roc\_auc\_score it is a performance measurement of model. It shows model capability to distinguishing classes.

**To find best random state and model:**

For select best model and random state which give high accuracy score can able to find by iterate different random state and classification model through the model train and test process



Train test split is helps to split the train and test data with some tuning parameter.

Test size = It decides ratio of train and test data if 0.30 is test size then test data percentage is 30% and train data percentage will 70%.

Random state = It is a type of tuning where it randomly desired which data should go for train process and which data is for test process.

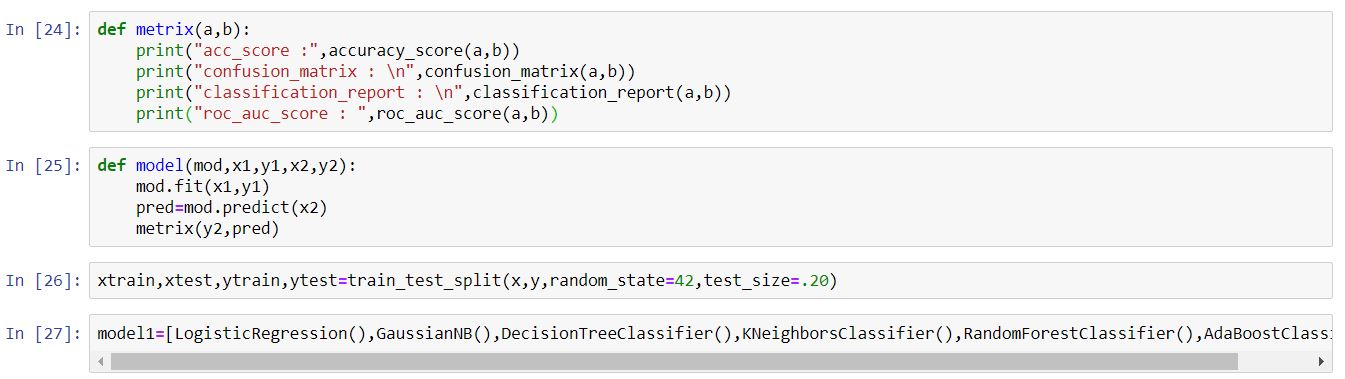
In training phase we have to fit both the input and target train data.

Prediction is have to done for input test data to get target test data.

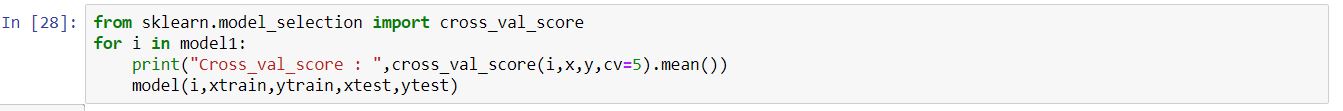
For to find the accuracy score we have to compare the actual target test data with predicted target test data.

**Cross validation:**

There may be over fitting can occur to avoid that we have to do cross validation by cross\_val\_score model in sklearn library.



Here we implement model train and test structure in user defined function named as model and accuracy score, confusion matrix, classification report in matrix user defined function.



All classification models in the list iterated one by one through cross validation technique and user defined function model to get the data without over fitting and its accuracy score.

Cross\_val\_score : 0.8164267866066967

acc\_score : 0.8275862068965517

confusion\_matrix :

[[12 19]

[ 1 84]]

classification\_report :

precision recall f1-score support

0 0.92 0.39 0.55 31

1 0.82 0.99 0.89 85

accuracy 0.83 116

macro avg 0.87 0.69 0.72 116

weighted avg 0.84 0.83 0.80 116

roc\_auc\_score : 0.6876660341555978

Cross\_val\_score : 0.8059614929377416

acc\_score : 0.8275862068965517

confusion\_matrix :

[[12 19]

[ 1 84]]

classification\_report :

precision recall f1-score support

0 0.92 0.39 0.55 31

1 0.82 0.99 0.89 85

accuracy 0.83 116

macro avg 0.87 0.69 0.72 116

weighted avg 0.84 0.83 0.80 116

roc\_auc\_score : 0.6876660341555978

Cross\_val\_score : 0.7210252768352665

acc\_score : 0.7241379310344828

confusion\_matrix :

[[16 15]

[17 68]]

classification\_report :

precision recall f1-score support

0 0.48 0.52 0.50 31

1 0.82 0.80 0.81 85

accuracy 0.72 116

macro avg 0.65 0.66 0.65 116

weighted avg 0.73 0.72 0.73 116

roc\_auc\_score : 0.6580645161290323

Cross\_val\_score : 0.6151910886661931

acc\_score : 0.6120689655172413

confusion\_matrix :

[[ 4 27]

[18 67]]

classification\_report :

precision recall f1-score support

0 0.18 0.13 0.15 31

1 0.71 0.79 0.75 85

accuracy 0.61 116

macro avg 0.45 0.46 0.45 116

weighted avg 0.57 0.61 0.59 116

roc\_auc\_score : 0.4586337760910816

Cross\_val\_score : 0.7729611510034458

acc\_score : 0.7413793103448276

confusion\_matrix :

[[16 15]

[15 70]]

classification\_report :

precision recall f1-score support

0 0.52 0.52 0.52 31

1 0.82 0.82 0.82 85

accuracy 0.74 116

macro avg 0.67 0.67 0.67 116

weighted avg 0.74 0.74 0.74 116

roc\_auc\_score : 0.6698292220113852

Cross\_val\_score : 0.7835476998342934

acc\_score : 0.8103448275862069

confusion\_matrix :

[[14 17]

[ 5 80]]

classification\_report :

precision recall f1-score support

0 0.74 0.45 0.56 31

1 0.82 0.94 0.88 85

accuracy 0.81 116

macro avg 0.78 0.70 0.72 116

weighted avg 0.80 0.81 0.79 116

roc\_auc\_score : 0.6963946869070209

From the above model logistic regression had a good accuracy score because the difference of accuracy score and cross validation score is minimum for Logistic Regression when compare with other model accuracy value.

**Hyper parameter tuning:**

Hyper parameter tuning can be done by Grid Search CV.



For Logistic regression there are certain parameters to tune for getting good accuracy score.

Penality-l1 (lasso), l2 (ridge) - it species the norm in penalization

C- Inverse of regularization strength.

After applying Grid Search CV for Logistic Regression model there may increase in accuracy score by the process of tuning.

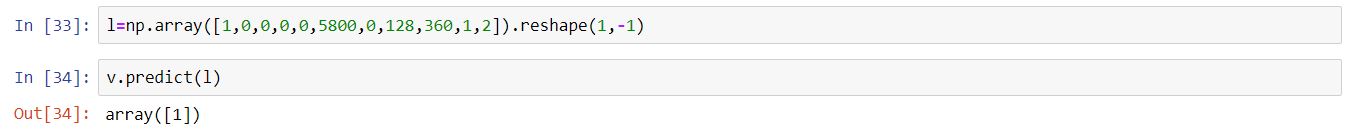
**Conclusion:**

Dump the model prediction in a file for the developer use



Joblib or pickle will help you to dump the predicted model in object file.

It can be saved for later prediction process for the same size of input data.



V stores the model for prediction.

L stores the required input value (personal and loan detail) of a person who apply for loan.

With the help of model saved in v and input data stored l we can able to predict the outcome either the person will get loan or not.