**Loan Application Status**

The article is about predicting whether the applicant is eligible for the loan or not by using various Machine learning Models.

**Problem Definition :-**

Approving the loan is a very important process for banks. Banks should be careful in approving the loan or rejecting the loan. It is very difficult to predict which applicant will repay the loan in correct intervals. With the help of the given data set as a data scientist, we should predict which applicant is eligible to get the loan and which applicant is not.

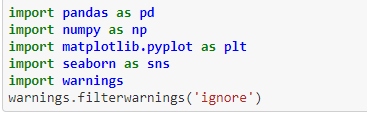


**Data Description :-**

The data set contains details of the applicants who applied for the loan. The data set contains details like applicant gender, education, marital status, employment, the income of the applicant and income of the co-applicant, loan amount, credit history, property area, etc.

**The language used for problem solving is Python .**

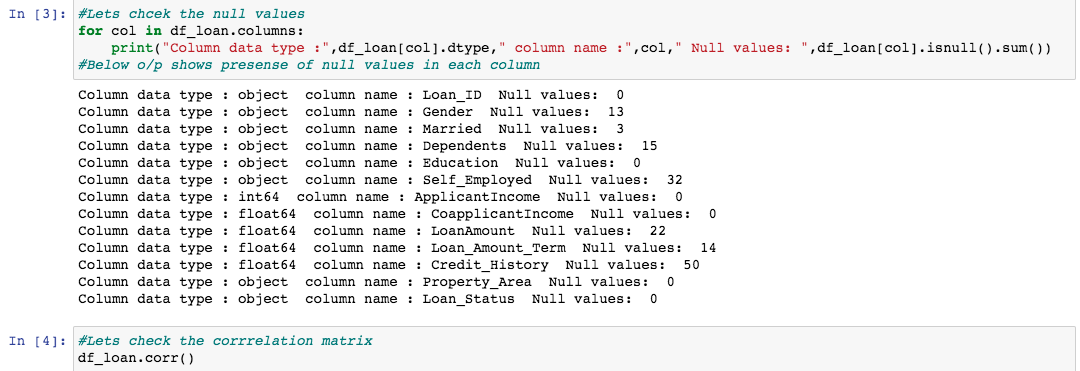
**Importing Required Dependencies :-**



**Loading the data :-**

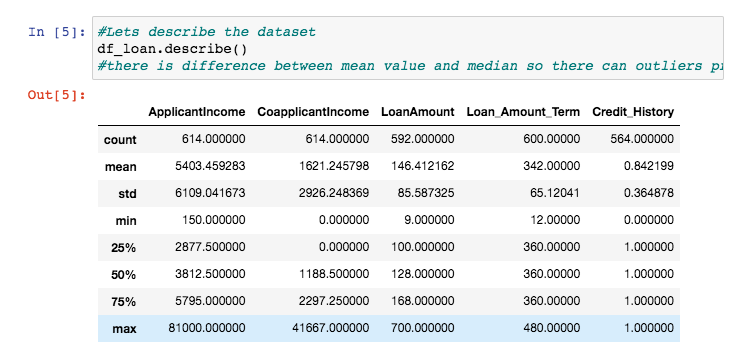
****

**Checking data type and missing values :-**

****

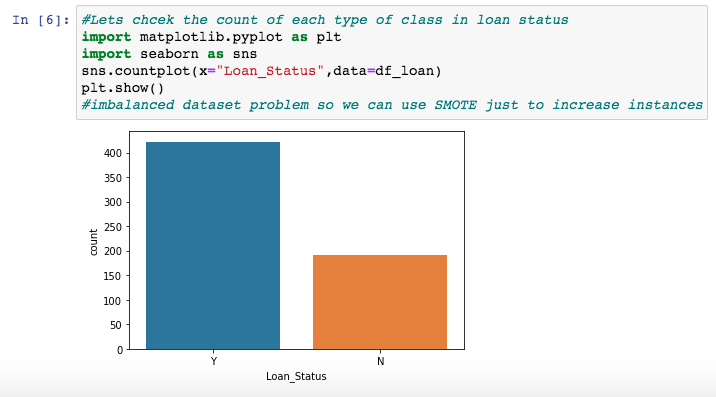
We can see that there are missing values (Nan) values in the given data set. We will treat the values further.

**Descriptive Statistics :-**

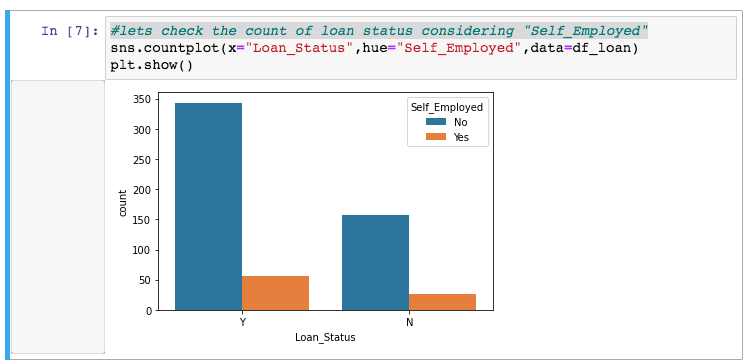
****

Although the column ‘Dependents’ is in integer format which is not visible in the statistical summary. So we will check the values in the column. Here, the min and max values have a high difference and the Standard deviation is also high. So , there must be skewness in the data.

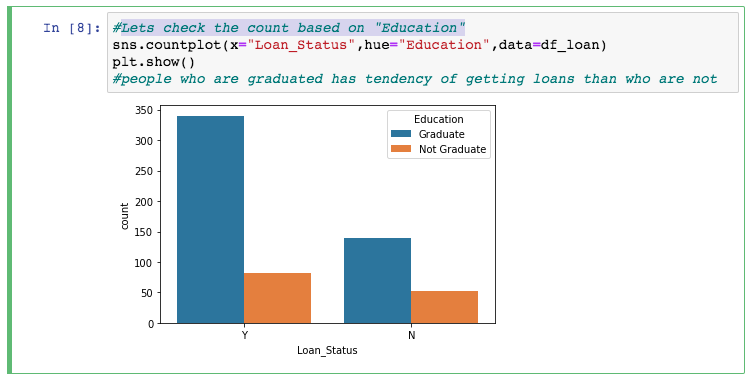
**EDA Visualization :-**

****

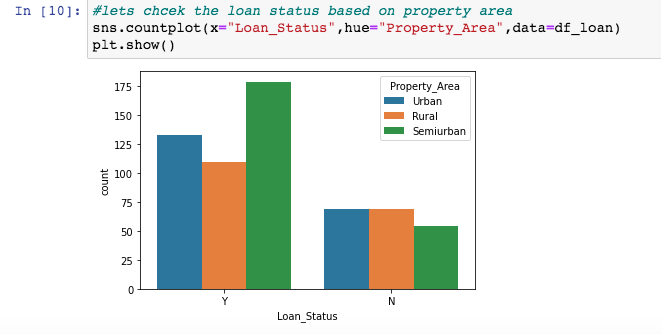
**We will check the count of loan status considering "Self Employed"**

****

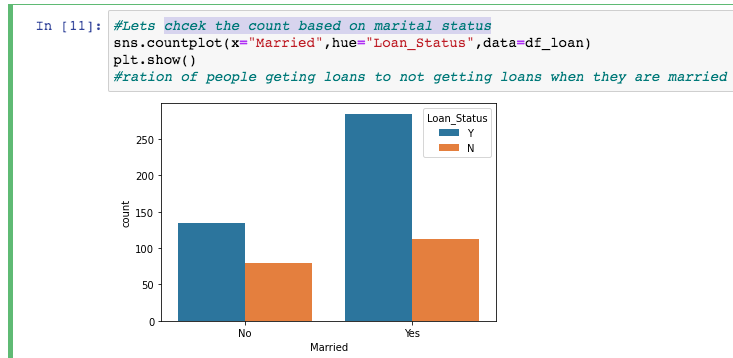
**Now we will check the count based on "Education"**

****

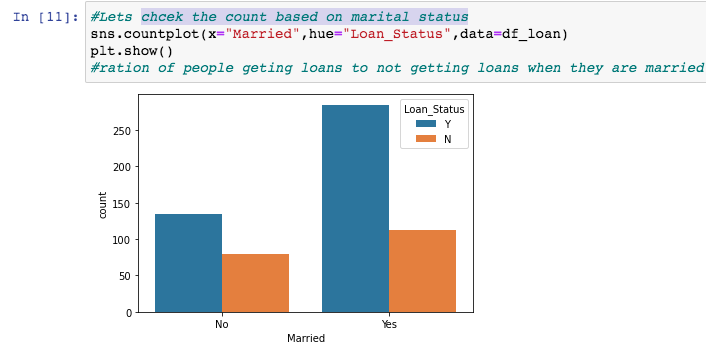
**Checking loan status based on Property Area**

****

**Checking the count based on marital status**

****

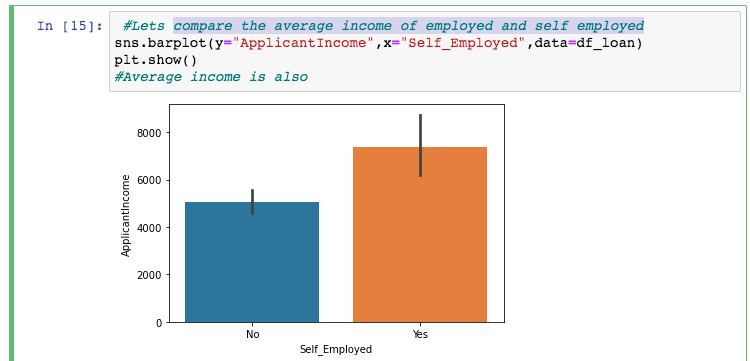
**Check the loan status for applicant income greater than 5000**

****

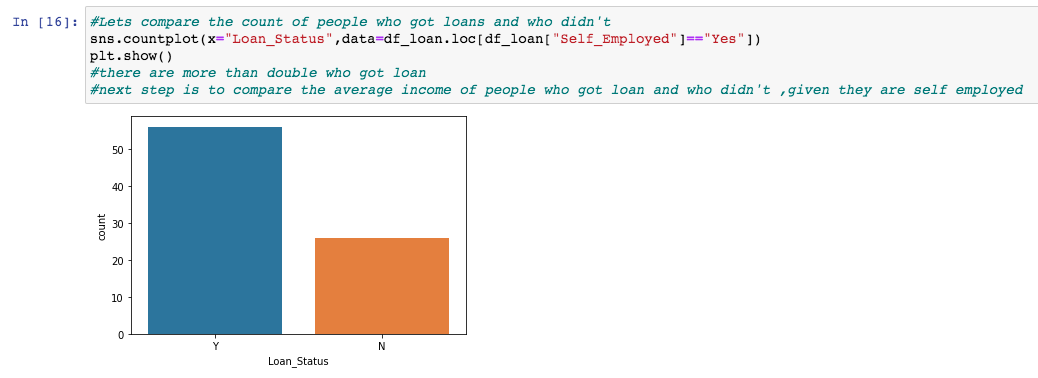
**Check what is the effect on above count if co-applicant income is 0**

****

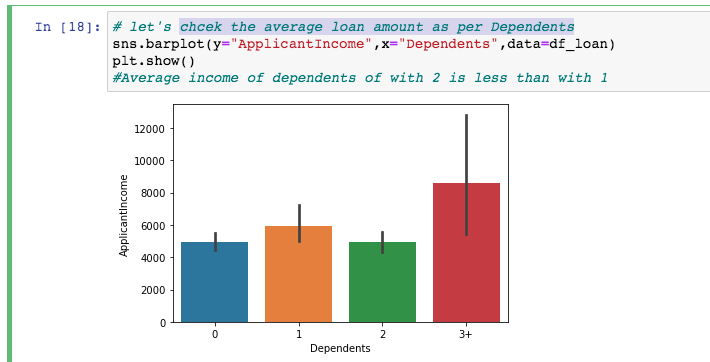
**Comparing the average income of employed and self employed**

****

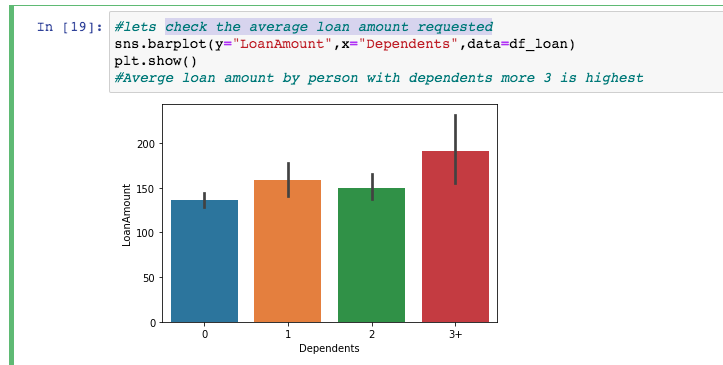
**Now compare the count of people who got loans and who didn't.**

****

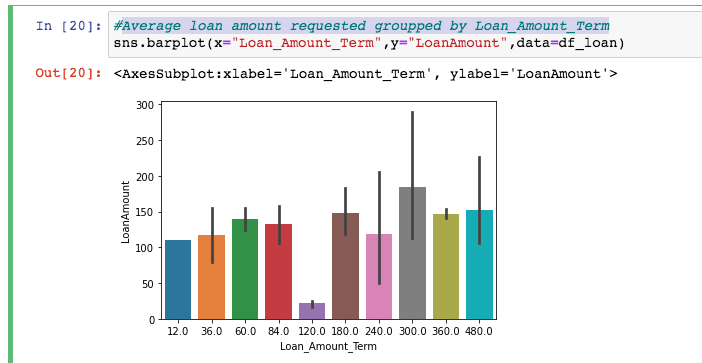
**Now check the average loan amount as per Dependents.**

****

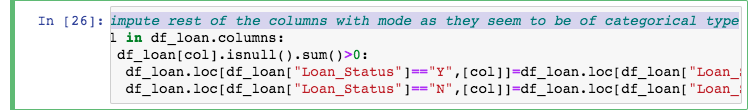
**Check the average loan amount requested.**

****

**For average loan amount requested grouped by Loan Amount Term.**

****

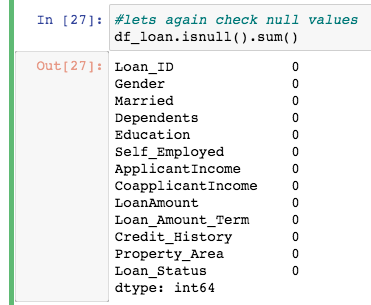
Lets impute rest of the columns with mode as they seem to be of categorical type.



Observations from the categorical data:

* Gender - Male applicants approved more than Female applicants.
* Education – Higher loans approved for the graduates when compared to undergraduates.
* Self Employed – Most loans approved for the people who work for an organization with fixed salaries when compared to self employed people.
* Property Area – Where the property rates high (urban semi- urban) those property owners loans are approved when compared to the owners from rural areas.
* Dependents – Zero dependents applicants loan approved when compared to 1,2,3+ dependents. Also in this column due to the “3+” value, the data type is showing as an object instead of an integer. So we will change the “3+” as “3” in the future process to train the model

**Again check null values :-**

****

**Treating Skewness :-**

What is Skewness and why it is bad?

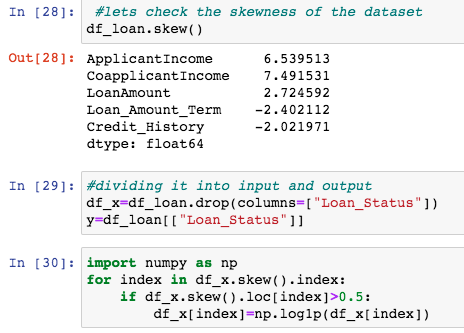
Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive, zero, negative, or undefined.

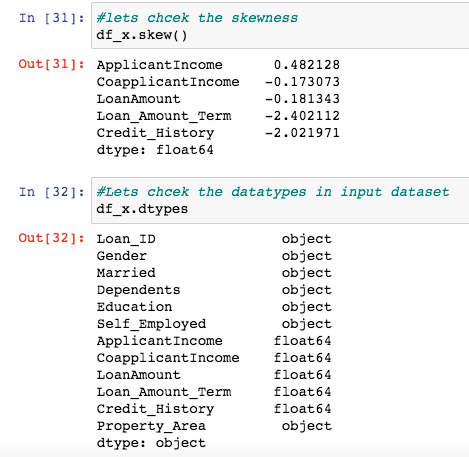
If there is too much skewness in the data, then many statistical models don't work but why. So in skewed data, the tail region may act as an outlier for the statistical model and we know that outliers adversely affect the model's performance. So, it is very important to remove the skewness before you transfer the data to train the model

Observations:

In the columns Applicant income, Co-applicant income, and Loan amount we can see that there is skewness in the data.

We can check the skewness with df.skew()

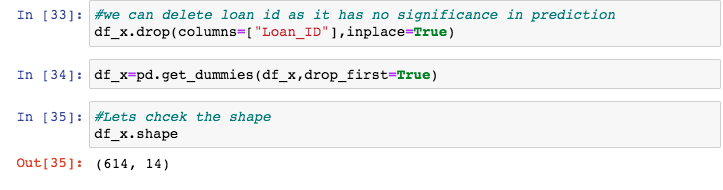
****

****

We removed the skewness from the data, now the numerical variables are normally distributed.

**Treating Missing Values:**

We will delete loan id as it has no significance in prediction.

****

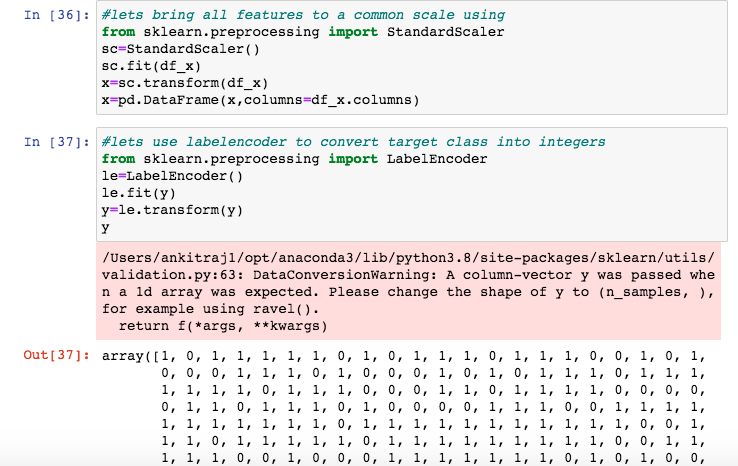
**For model using label encoder :-**

**Scaling :-**

In the descriptive statistics, we saw there is a huge difference in minimum and maximum values. It looks the data is not normalized. So we will do scaling. Here I used the Standard scaling method.

The idea behind standard scaling is that it will transform our given dataset such that its distribution will have a mean value 0 and standard deviation of 1

**We need numerical data to train the model. So we will do an encoding process to change the categorical data into numeric data.**

****

**Feature Engineering :-**

**Feature engineering** is the process of using [domain knowledge](https://en.wikipedia.org/wiki/Domain_knowledge) to extract [features](https://en.wikipedia.org/wiki/Feature_(machine_learning)) (characteristics, properties, attributes) from raw [data](https://en.wikipedia.org/wiki/Data).

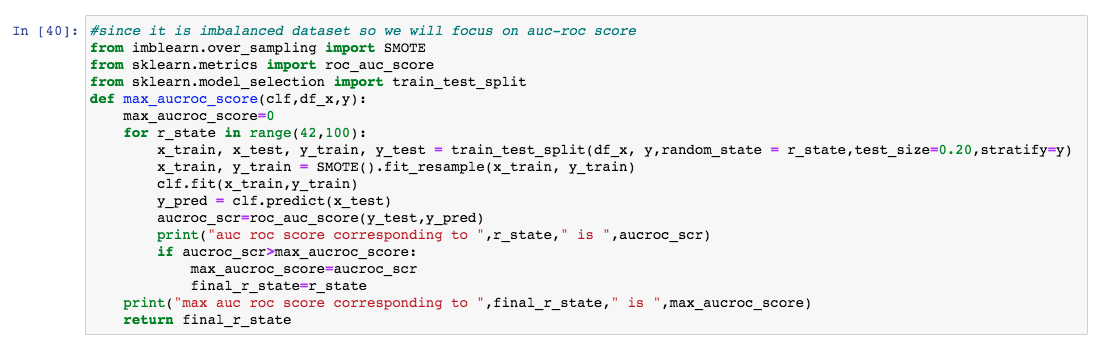
A [feature](https://en.wikipedia.org/wiki/Feature_(machine_learning)) is a property shared by independent units on which analysis or prediction is to be done.[[2]](https://en.wikipedia.org/wiki/Feature_engineering#cite_note-:0-2)

Features are used by [predictive models](https://en.wikipedia.org/wiki/Predictive_modelling) and influence results.

**Checking Auc – Roc score For different Machine learning models :-**

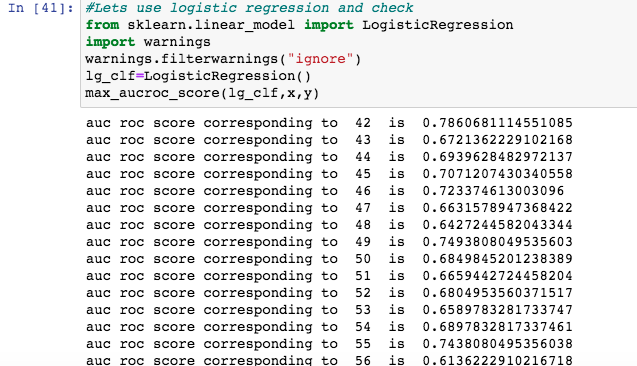
Since our data set is imbalanced we will focus on the auc-roc score.

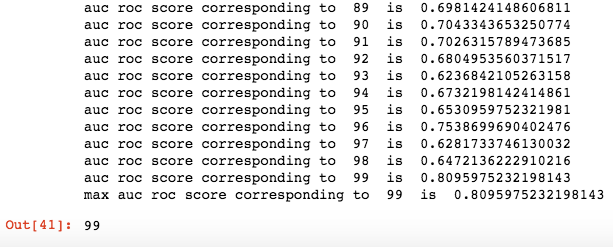
**Required Machine Learning Libraries :-**

****

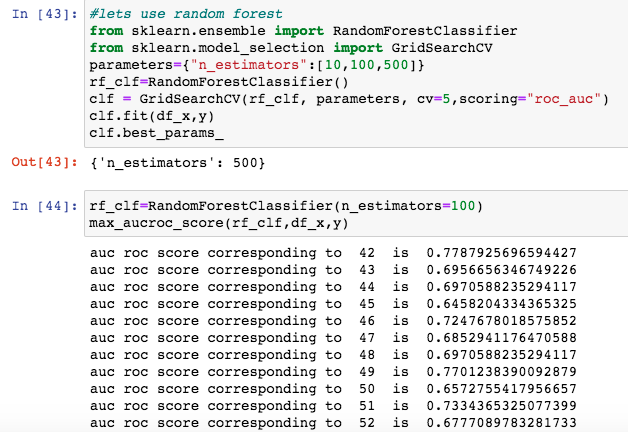
**Now we will check the auc roc score for different machine learning models.**

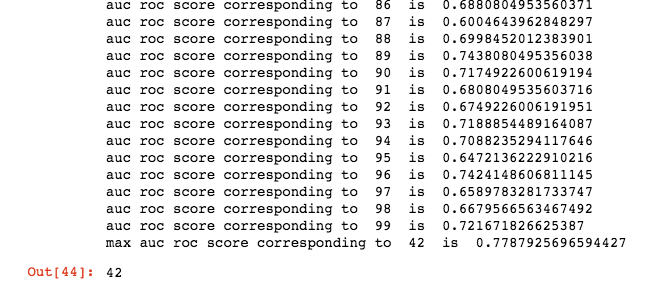
**For Logistic Regression :-**



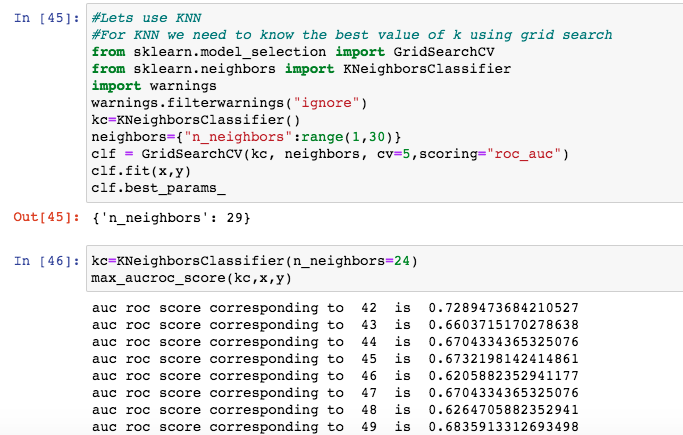


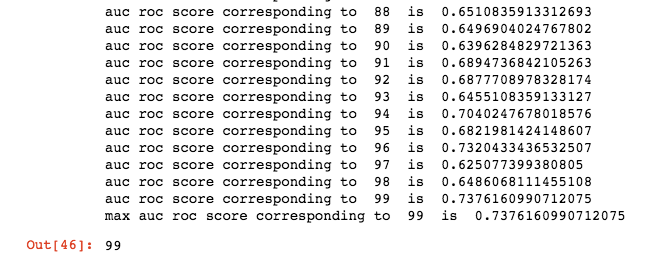
**For Random forest model :-**

****

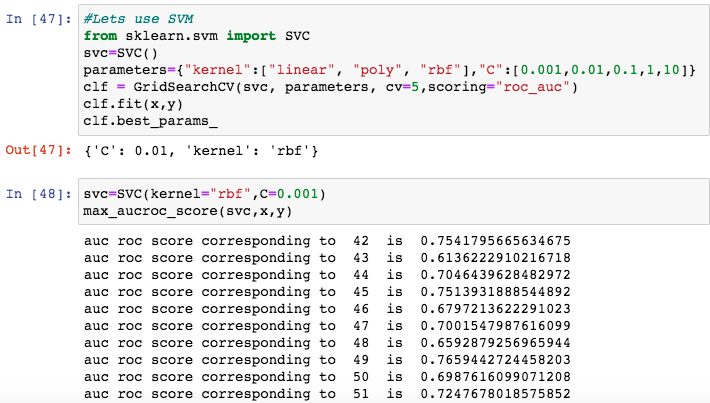
****

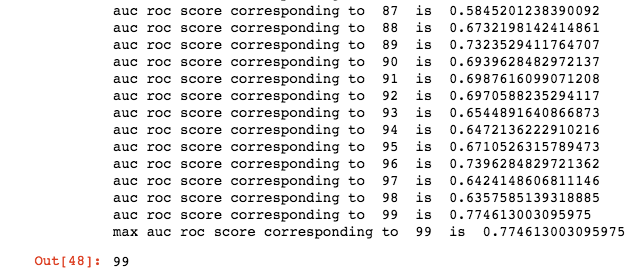
**Using KNN :-**

****

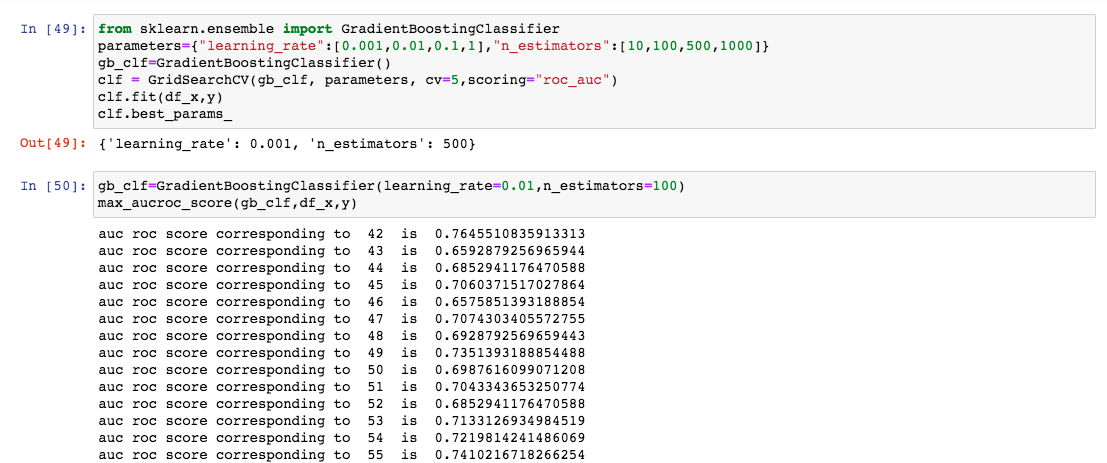
****

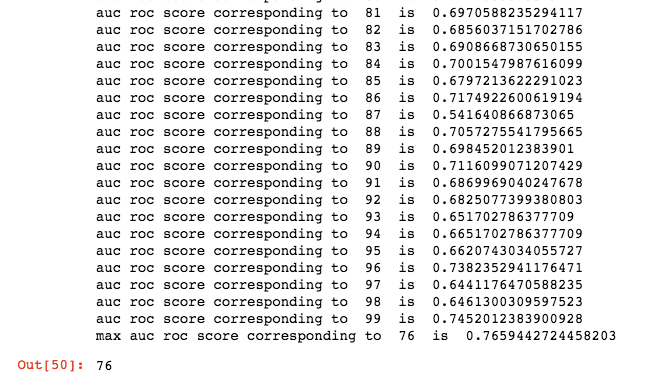
**Let’s use SVM :-**

****

****

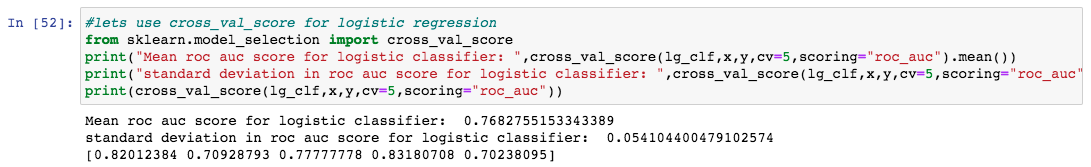
**Using Gradient Booster Classifier:-**

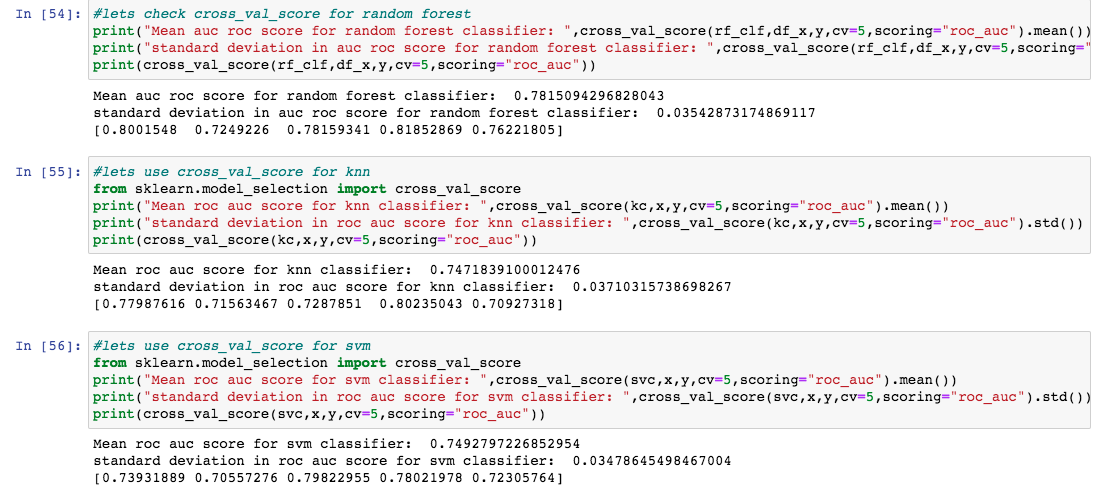
****

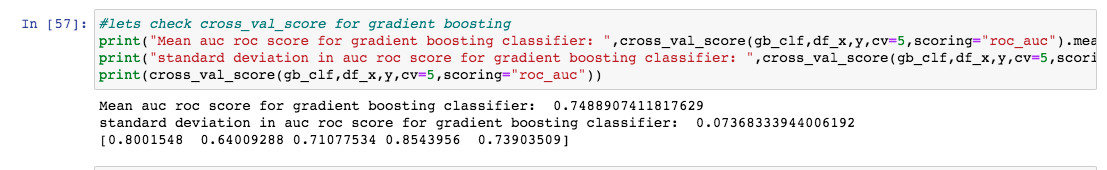
****

We are getting the accuracy with SVM (99%) and Random Forest classifier (42%), Logistic Regression is providing 99% and Gradient Boosting classifier is providing 76% and KNN is giving 99%. But a f1 score is not good. Let's do cross-validation to get better accuracy and fewer errors.

**Cross Validation On Different Models:-**

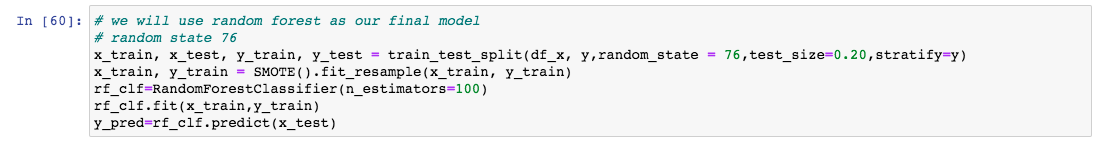
****

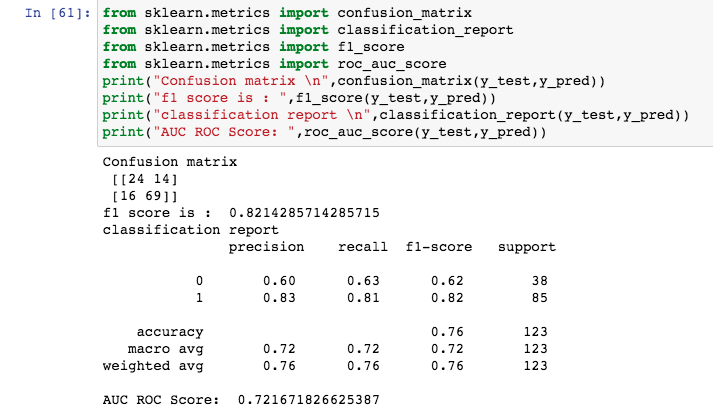
****

****

**Selecting Our Final Machine Learning Model :-**

**We will use random forest as our final model :-**

****

****

Conclusion:

In this type of problem Feature Engineering is the most crucial thing . You can see how we have handled the categorical data by doing encoding process and numerical data by removing skewness and scaling, also replacing missing values with simple imputer and also how we build different ML model on the same dataset, doing cross validation .

**FEEDBACK :-**

For any reviews or feedback regarding article or any [*queries*](https://www.google.com/search?q=queries&spell=1&sa=X&ved=2ahUKEwjp3JmxmI_xAhUHeisKHYk2ASgQBSgAegQIARAw) please feel free to contact me on below mentioned email address. As a data science student it will help me to learn more. You can send your feedback at -

ankitverma87416@gmail.com