**PROBLEM DEFINITION**

The data consists of 10,000 observations of space taken by the SDSS. Every observation is described by 17 feature columns and 1 class column which identifies it to be either a star, galaxy or quasar.

**Inspiration:**

The aim is to try and predict/determine the whether the observed space object is a Star, galaxy or quasar.

**DATA DESCRIPTION**

Fields or columns present in the dataset :

* objid - Object Identifier - Single value
* ra - J2000 Right Ascension (r-band) - Float - Continuous

Right ascension (abbreviated RA) is the angular distance measured eastward along the celestial equator from the Sun at the March equinox to the hour circle of the point above the earth in question.

* dec - J2000 Declination (r-band) - Float - Continuous
* u, g, r, i, z - Ultraviolet, Green, Red, Infrared - represent the five bands of a telescope - Float - Continuous
* run - identifies the specific scan - Integer - Continuous
* camcol - 'camera column' a number from 1 to 6, helps in identifying the scan line within the run - Integer - Continuous
* field - starts at 11 (after an initial rampup time), and can be as large as 800 - Integer - Continuous

Run, rerun, camcol and field are features which describe a field within an image taken by the SDSS. A field is basically a part of the entire image corresponding to 2048 by 1489 pixels.

* specobjid - generated from the plate number, mjd, and fiberid - Integer - Continuous
* redshift - measure of the recession velocity of a galaxy or other sky object - Float - Continuous
* plate - unique serial number - Integer - Discrete

Redshift happens when light or other electromagnetic radiation from an object is increased in wavelength, or shifted to the red end of the spectrum. Each spectroscopic exposure employs a large, thin, circular metal plate that positions optical fibers via holes drilled at the locations of the images in the telescope focal plane.

* mjd - Modified Julian Date, used to indicate the date that a given piece of SDSS data was taken - Integer - Continuous
* fiberid - each object is assigned a corresponding fiberID - Integer - Continuous

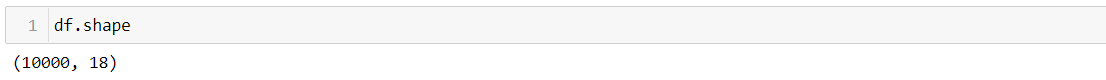
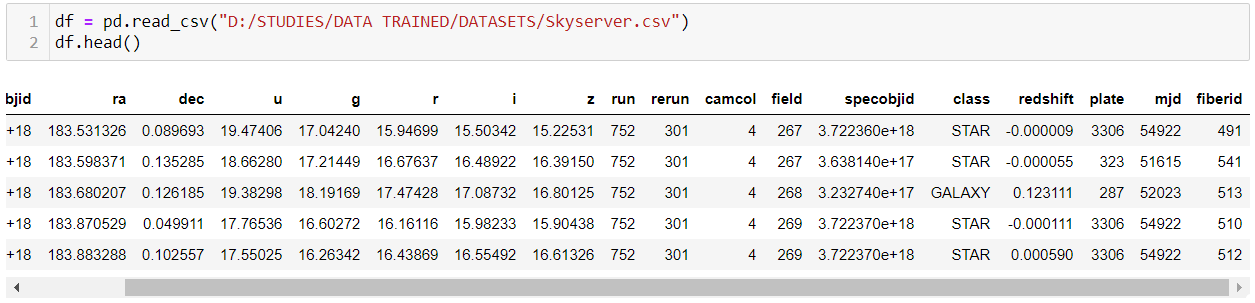
**DATA ANALYSIS & PRE-PROCESSING**

Importing Libraries:



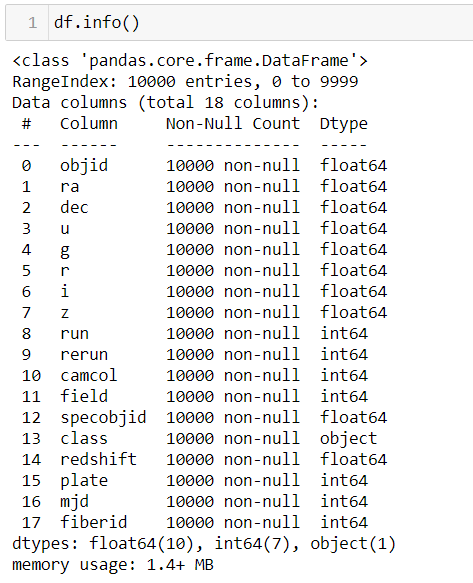
These are all the libraries that are required at various stages - EDA, visualization, prediction and evaluation - for executing this project.

**Loading Dataset:**

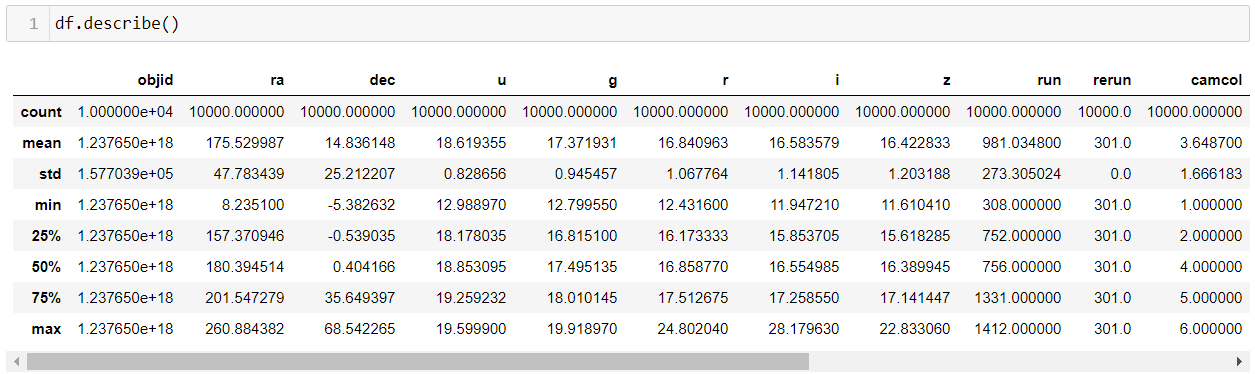


I have loaded the data into ‘df’ variable. The dataset contains 10,000 rows and 18 columns.

**Statistical Data Analysis:**

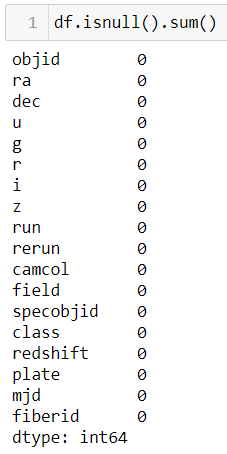


Except the ‘class’ column all the other columns are either of *integer* or *float* data type.



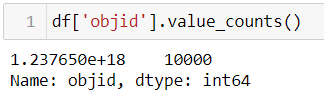
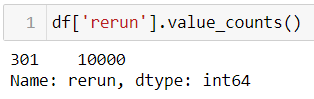
* ‘rerun’ and ‘objid’ columns have only one value and is constant
* ‘ra’ column has values ranging from 8.23 to 260.88 (Indicates Movement from East to West)
* ‘dec’ column has values ranging from -5.38 to 68.54 (Indicates Movement from South to North)
* ‘field’ column ranges from 11 to 768 - contains outliers - considerably big difference between 3rd quartile and max
* ‘plate’ column ranges from 266 to 8410 - contains outliers - considerably big difference between 3rd quartile and max

**Checking Null Values:**



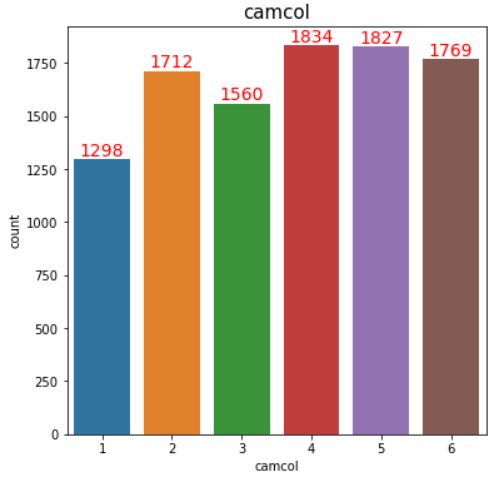
No null values are present in the dataset.

**Uni-variate analysis:**

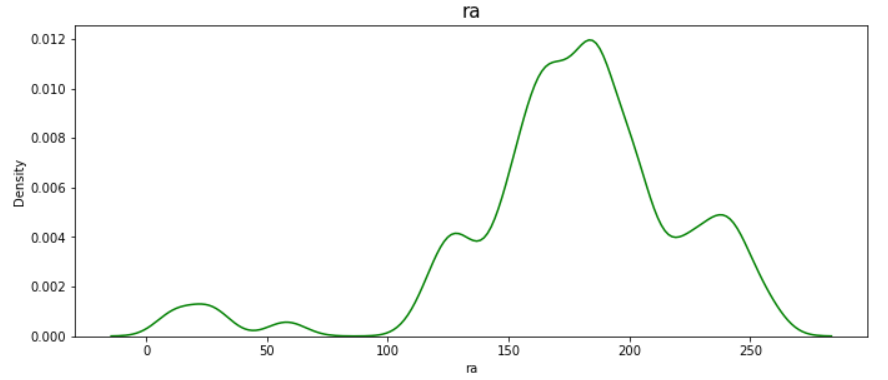
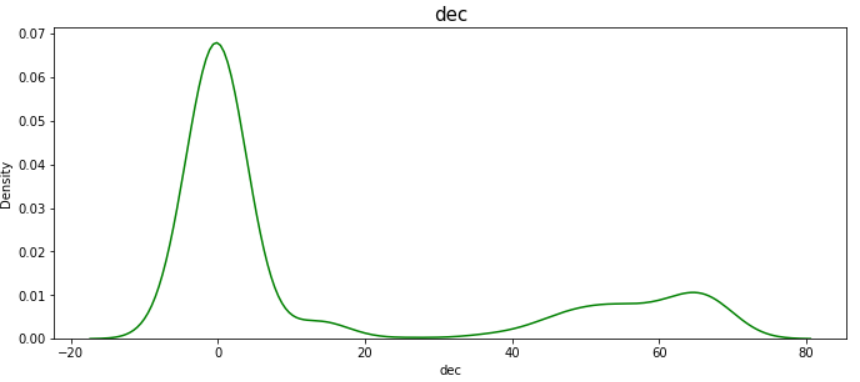
‘objid’, ‘rerun’ columns have only one value and hence these columns can be dropped before the model building process.

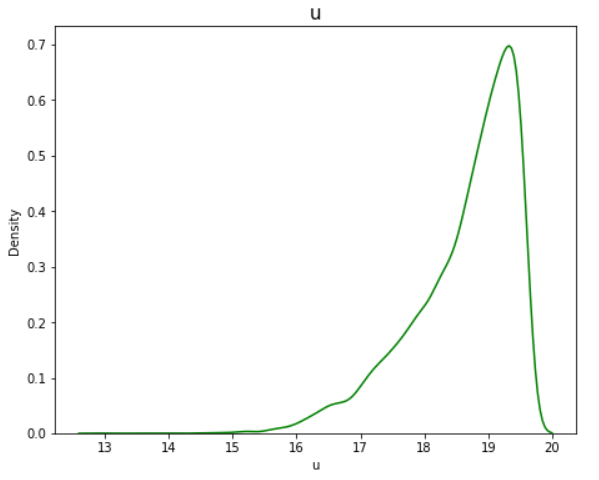
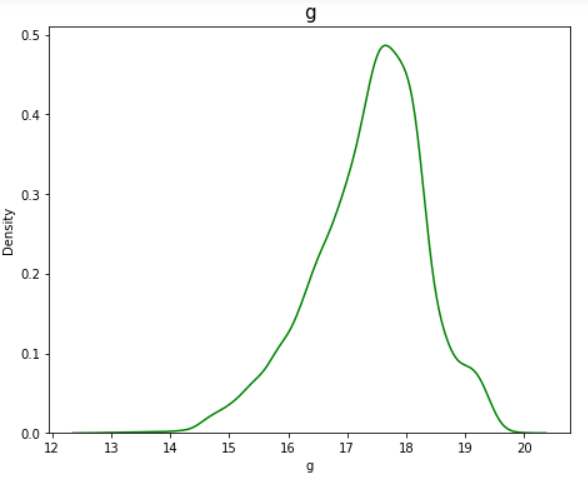
Initially the count of different categories in the categorical variables are checked using countplots. A countplot is kind of like a histogram or a bar graph for some categorical area. It simply shows the number of occurrences of an item based on a certain type of category.

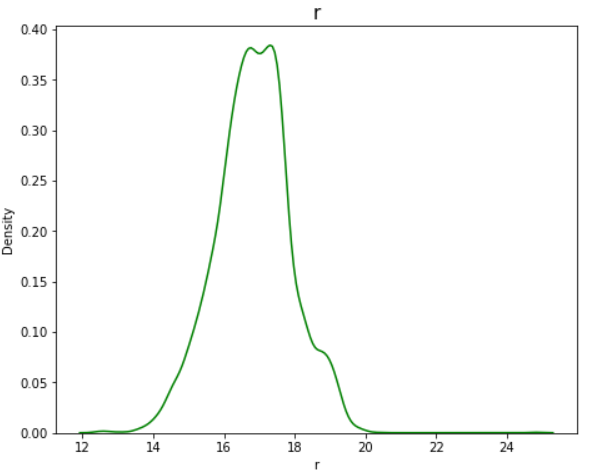
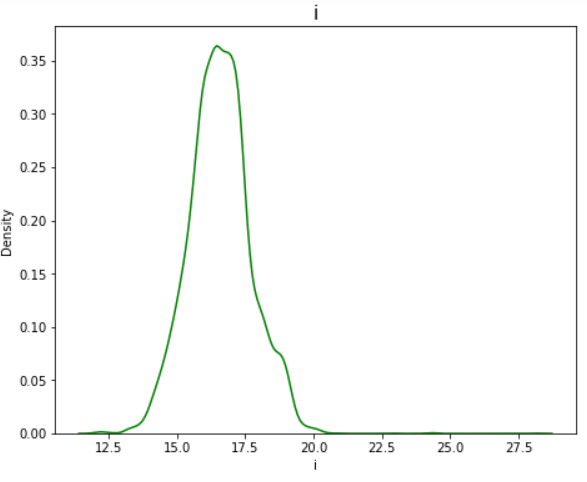
 

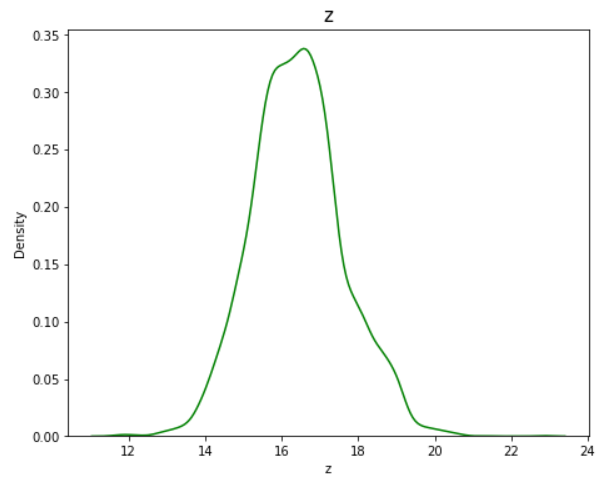
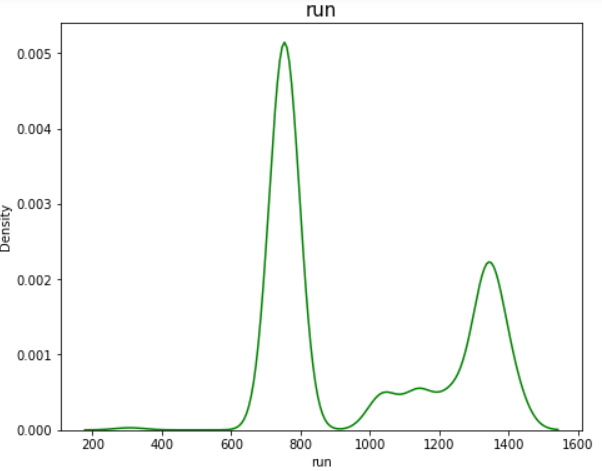
* More galaxies were observed than any other celestial body. QSO have the least number of observations.
* Highest number of observations for 'camcol' value 4 and lowest number of observations for 'camcol' value 1.

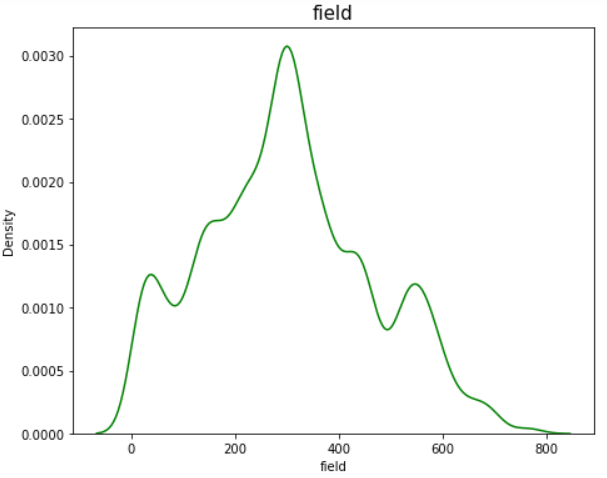
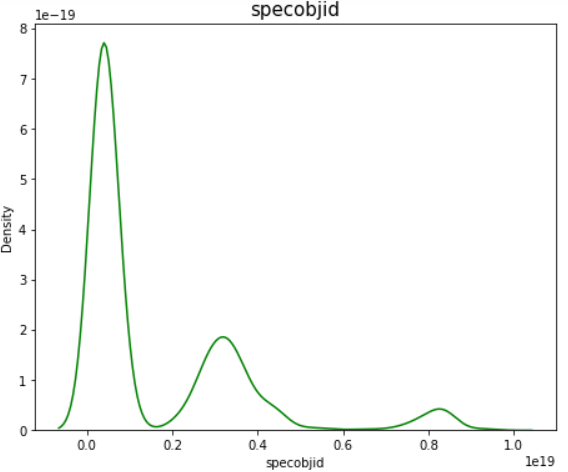
The distribution of continuous variables can be checked using distribution plots. A **distribution plot** displays a distribution and range of a set of numeric values plotted against a dimension. Distribution plot helps in identifying if the field is skewed or normally distributed.

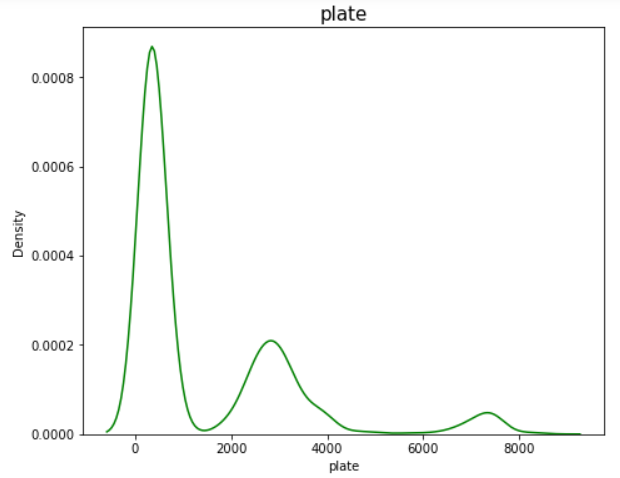
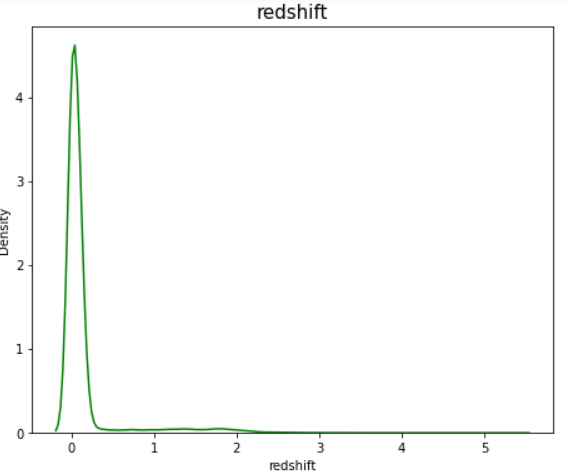
 

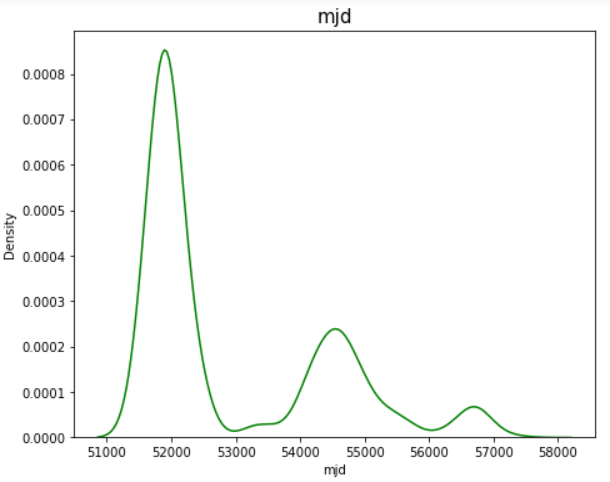
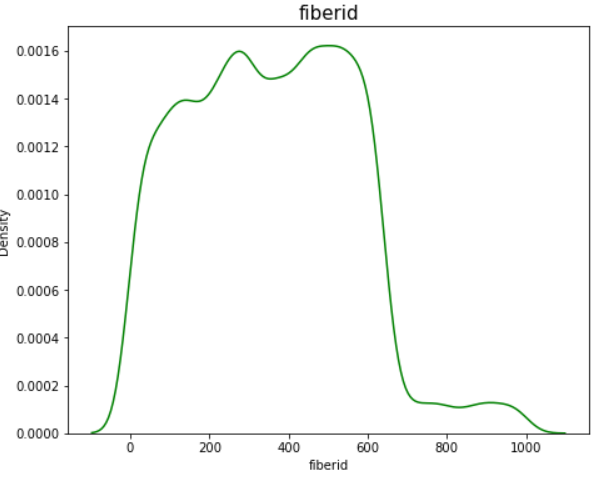
 

From the above distribution plots the following insights can be drawn :

* ‘ra’ column is multi-modal and has slight left skewness.
* ‘dec’ column has high right skewness.
* ‘u’, ’g’, ’r’, ’i’, ’z’ columns are uni-modal.
* ‘u’ & ‘g’ columns have slight left skewness.
* ‘r’ & ‘i’ columns have slight right skewness.
* ‘z’ column is normally distributed.
* ‘run’ column is bi-modal.
* ‘field’ column is multi-modal.
* ‘specobjid’, ‘plate’ & ‘redshift’ columns are right skewed.
* ‘mjd’ column is multi-modal and is right skewed.

The **outliers** of the continuous variables can be checked using boxplots. **Boxplots** are a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”).

First quartile (Q1/25th Percentile): The middle value between the smallest number (not the “minimum”) and the median of the dataset.

Median (Q2/50th Percentile): The middle value of the dataset.

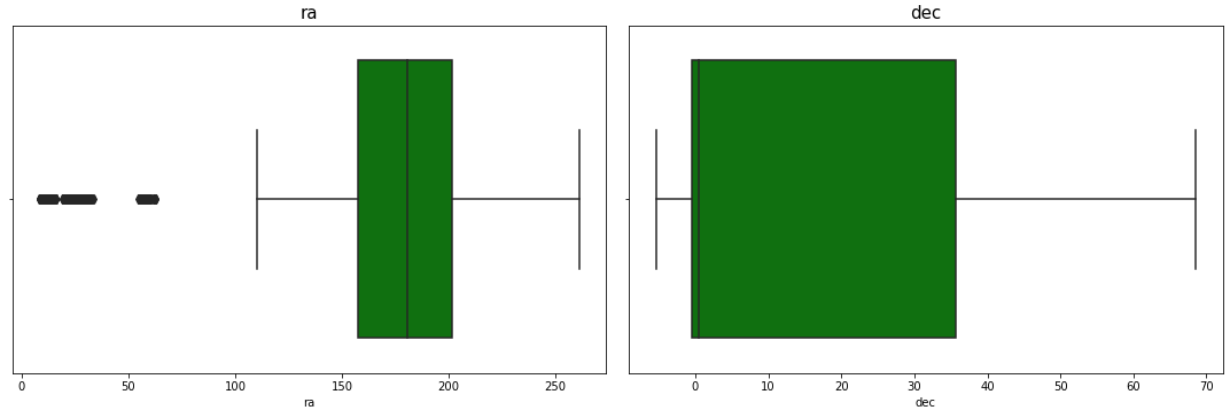
Third quartile (Q3/75th Percentile): The middle value between the median and the highest value (not the “maximum”) of the dataset.

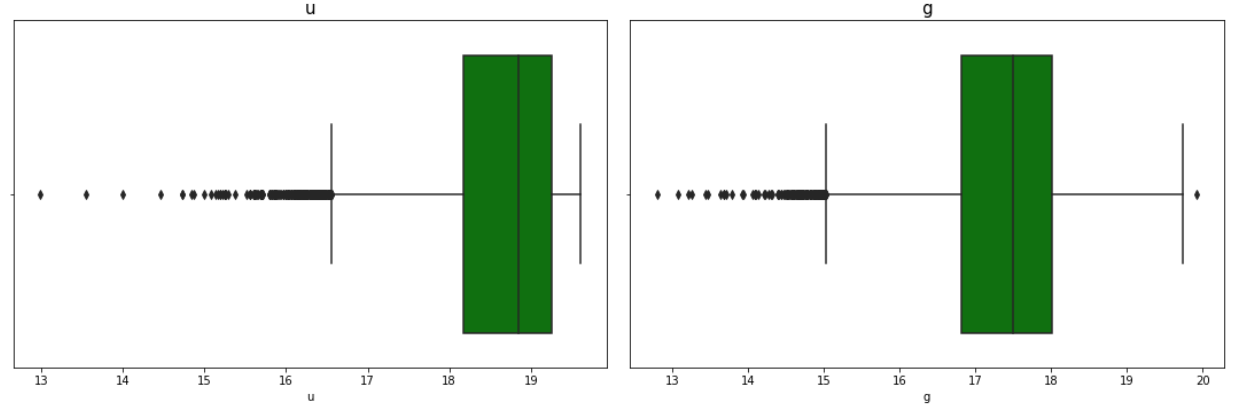
Inter-quartile range (IQR): Range of values between the 25th to the 75th percentile.

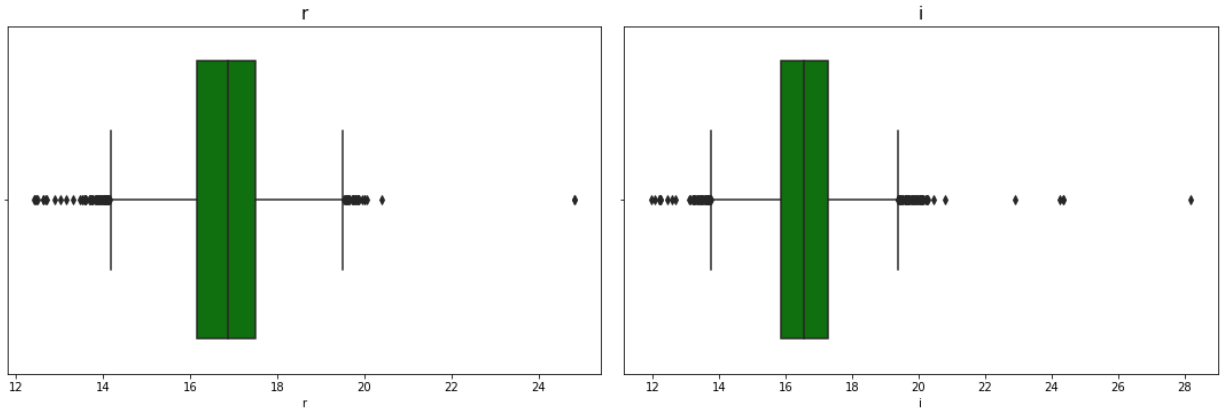
Maximum: Q3 + 1.5 \* IQR

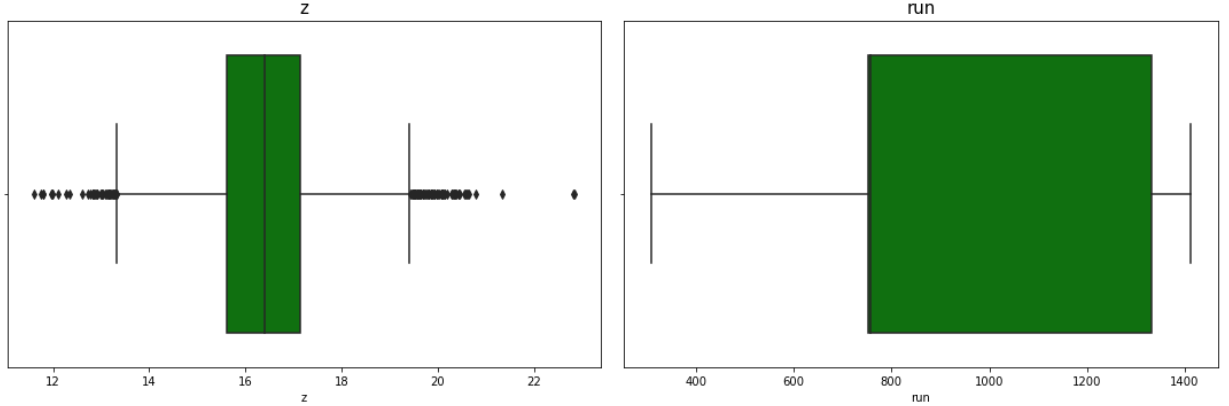
Minimum: Q1 - 1.5 \* IQR

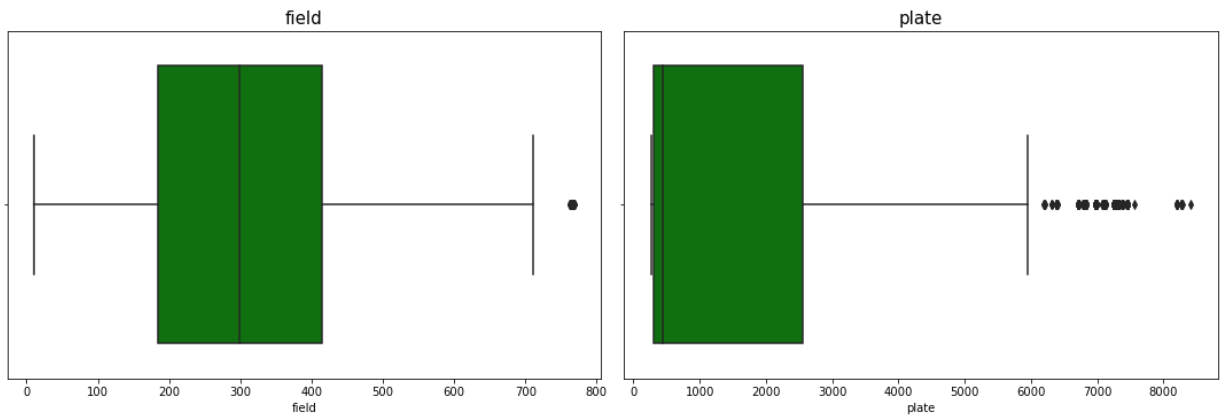
Any value greater than maximum and less than minimum is considered as an outlier.

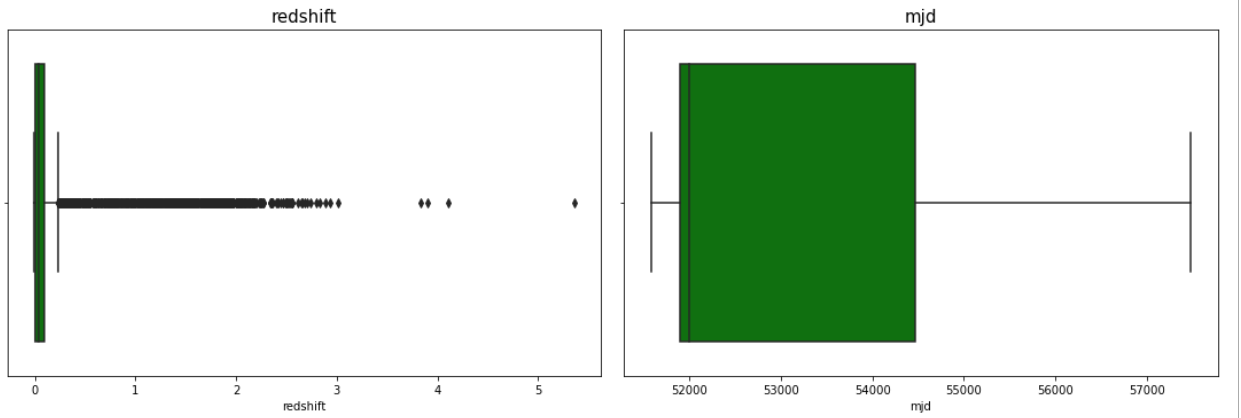


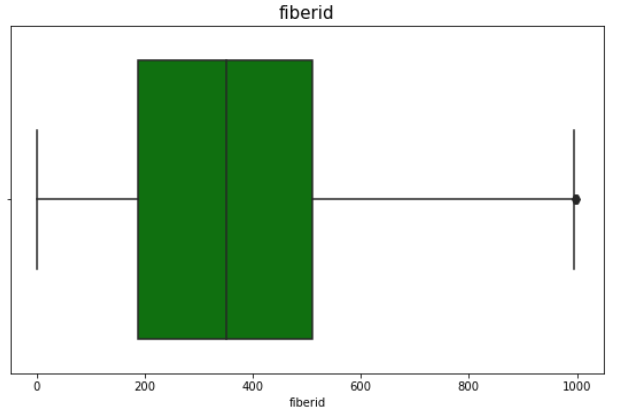












From the above box-plots the following insights can be drawn :

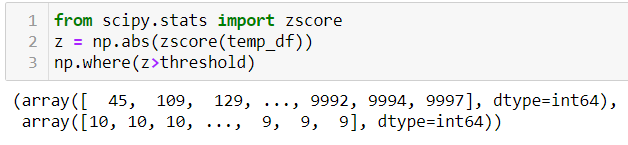
* There are many outliers present in 'ra', 'u', 'g', 'r', 'i', 'z', 'plate', 'redshift' columns

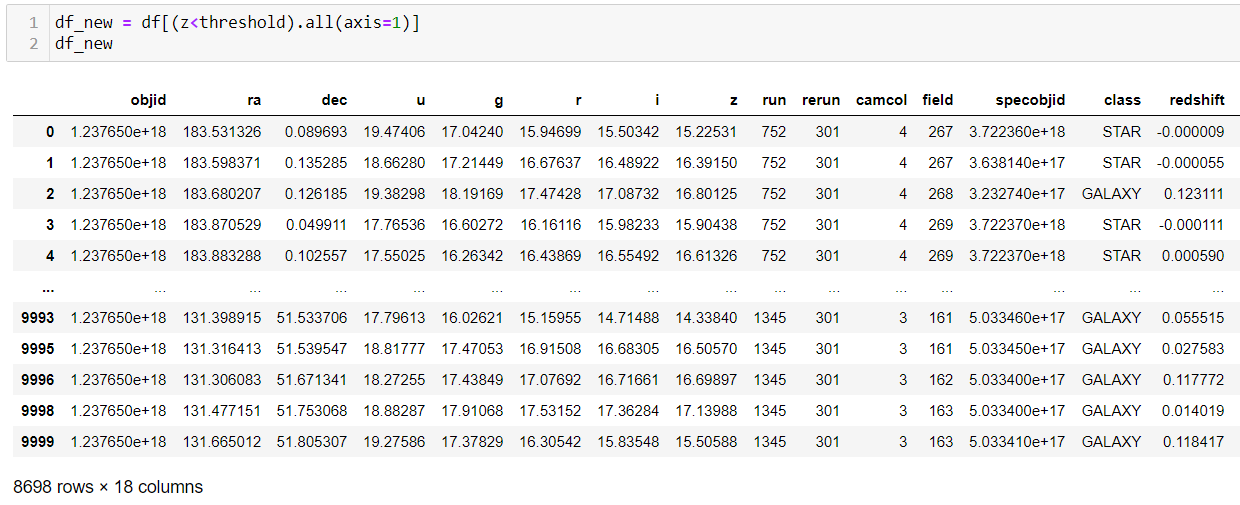
**Outlier Removal:**

Data outliers can spoil and mislead the training process resulting in longer training times, less accurate models, and, ultimately, more mediocre results. So, we must remove the outliers present in the dataset. But, we must be careful not to delete large amounts of data from the dataset as it may cause over fitting of the models. Up to 5% loss in data is considered acceptable. We can remove the outliers by two methods :

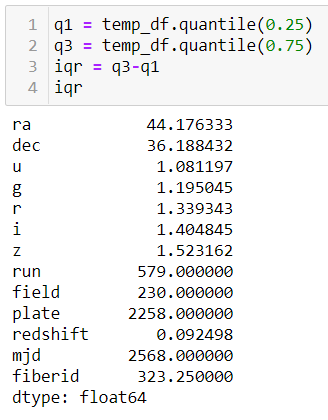
1. Z- Score Method - **Z-score** is a numerical measurement of how many [standard deviations](https://www.statisticshowto.com/probability-and-statistics/standard-deviation/) below or above the [population mean](https://www.statisticshowto.com/population-mean/)a [raw score](https://www.statisticshowto.com/raw-score/) or value is. If a Z-score is 0, it indicates that the data point's score is identical to the mean value. A Z-score of 1.0 would indicate a value that is one standard deviation away from the mean. Z-scores may be positive or negative, with a positive value indicating the score is above the mean and a negative score indicating it is below the mean. . Z-scores range from -3 standard deviations (which would fall to the far left of the normal distribution curve) up to +3 standard deviations (which would fall to the far right of the normal distribution curve). Therefore, we set threshold as 3 and delete all the values that are greater than the threshold.
2. Inter-Quartile Range(IQR) Method - In this method, IQR, “maximum” and “minimum” are calculated. Values less than minimum and greater than the maximum are deleted.

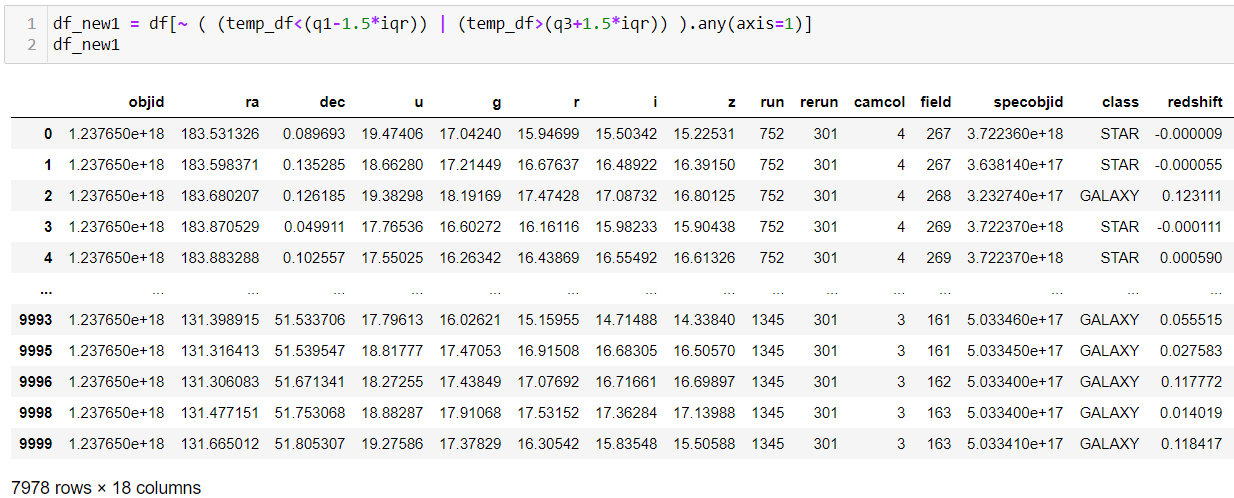
First, we try with the Z-score method, if the data loss is more than 5%, then we can try with IQR method.





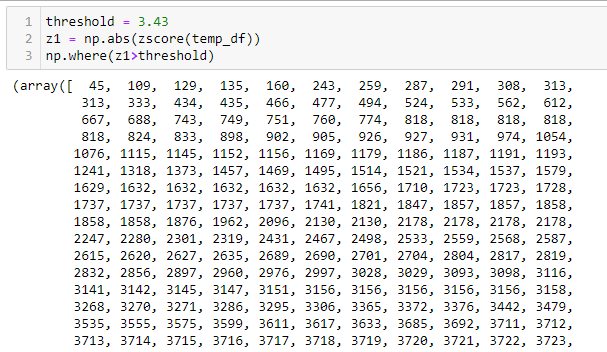
13% data is lost through this method, therefore we proceed with IQR method.

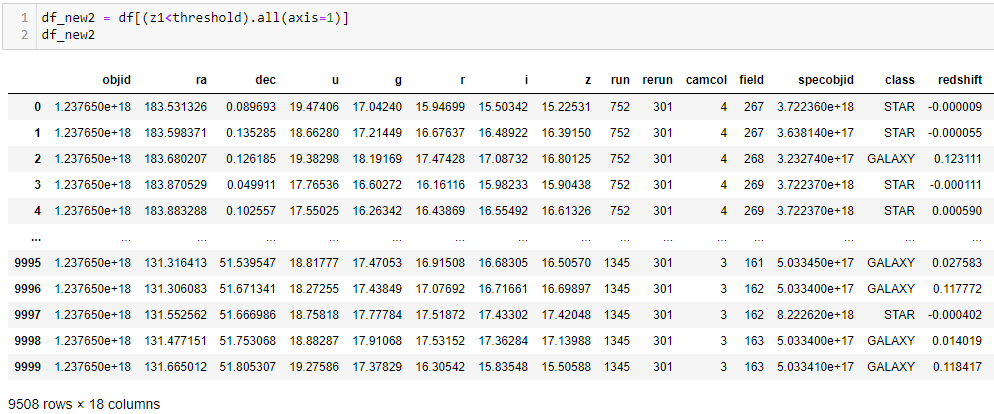




20% data is lost in IQR method which is much more than the z-score method.

Hence, we proceed by increasing the threshold to 3.4 in z-score method





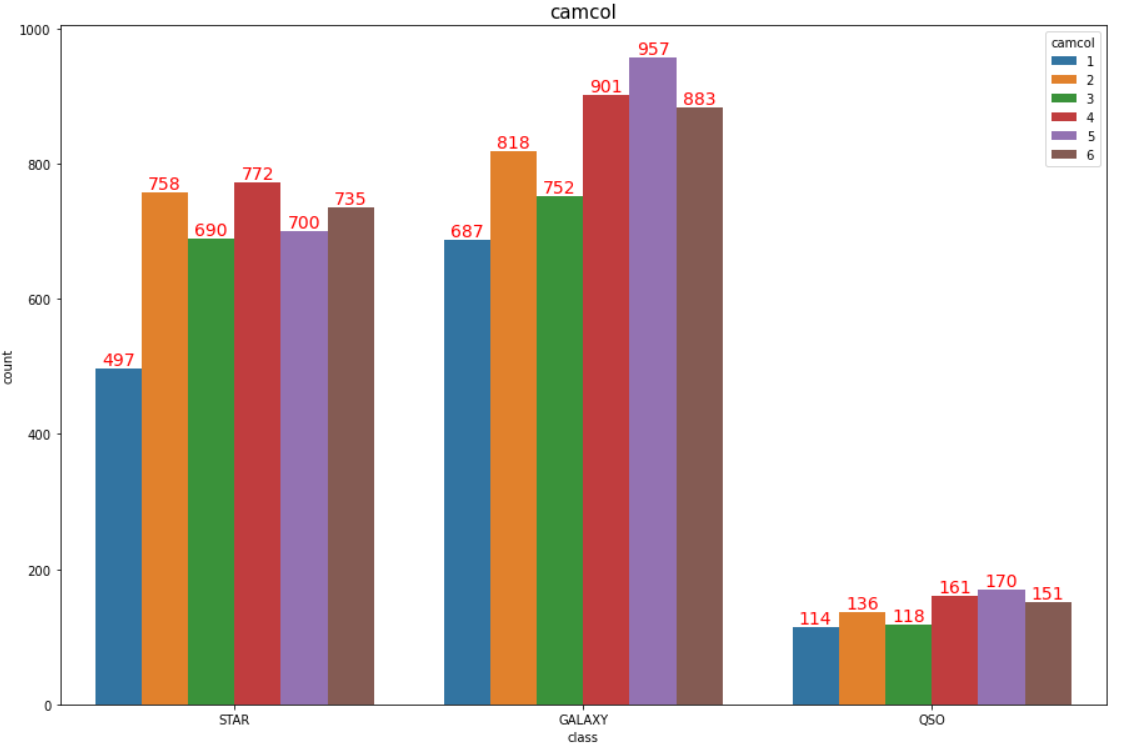
The new dataset has 9508 rows and 10 columns. After changing the threshold the data loss percentage reduced to 4.9% which is lesser than the permissible percent. Therefore, we can proceed.

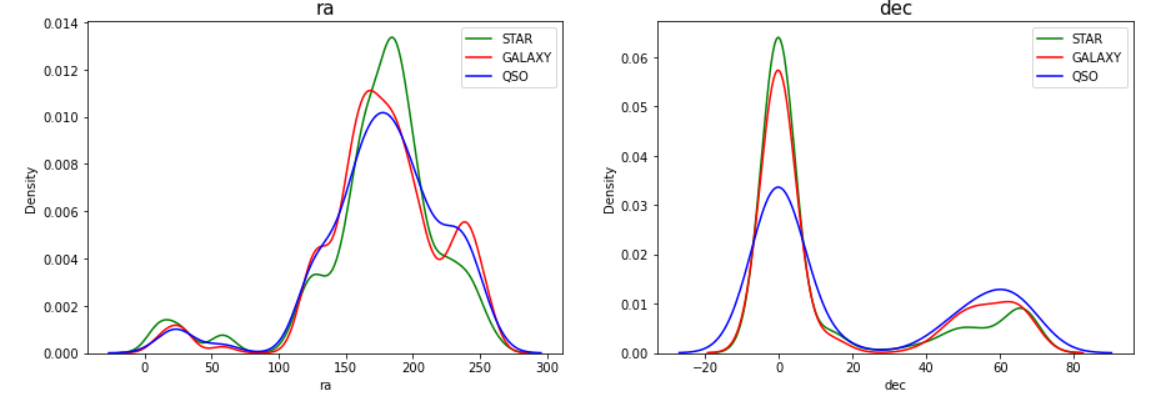
**Bi-variate Analysis:**

**Bi-variate analysis** is stated to be an analysis of any concurrent relation between two variables or attributes. This study explores the relationship of two variables as well as the depth of this relationship to figure out if there are any discrepancies between two variables and any causes of this difference.

Since our target variable is categorical we analyse how the other variables vary for each class in our target variable.

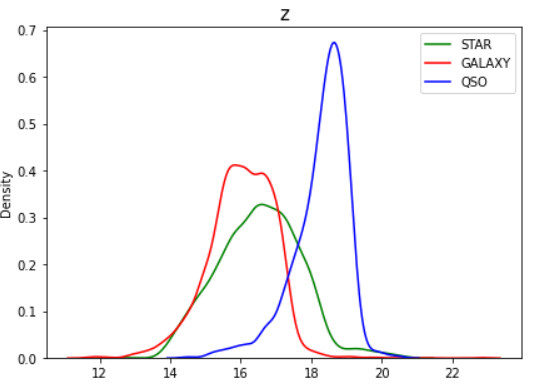
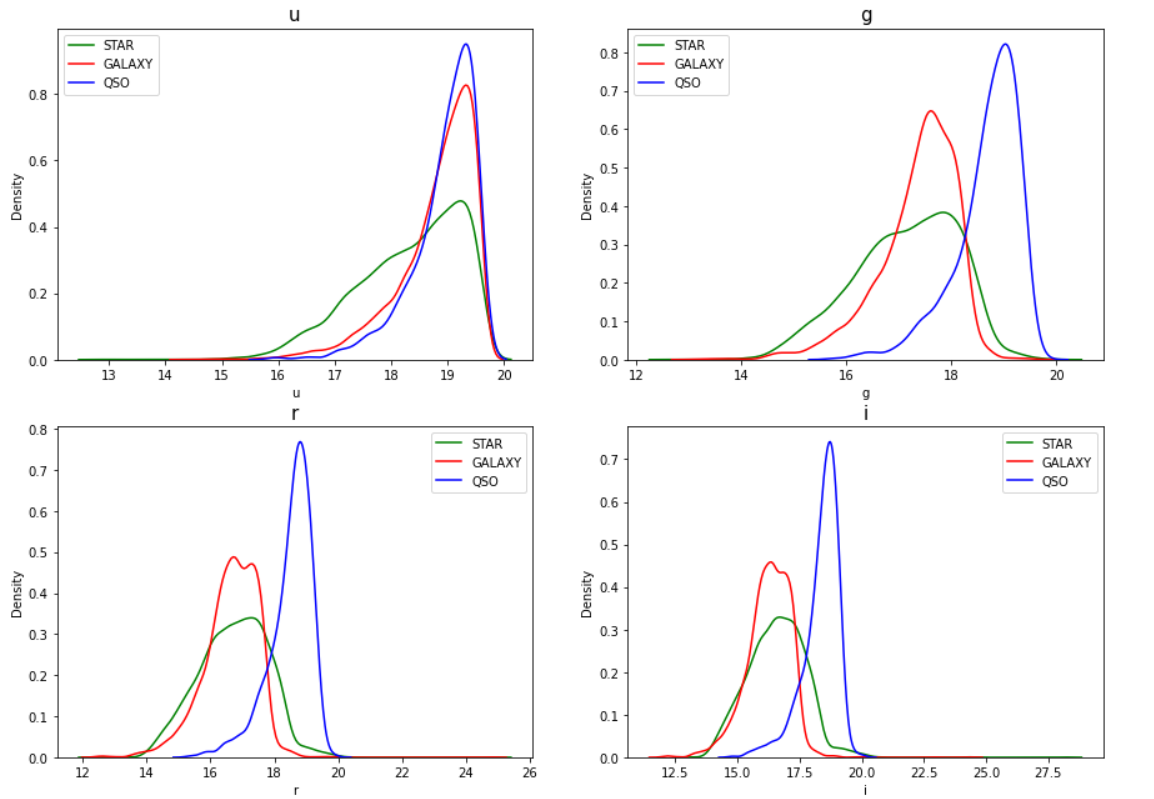
First, we analyze how the categorical variables vary with the target using count-plots.

The ‘camcol’ field is distributed in similar ratios across all the different classes/categories of the target variable. So we can’t draw great insights from this comparison.



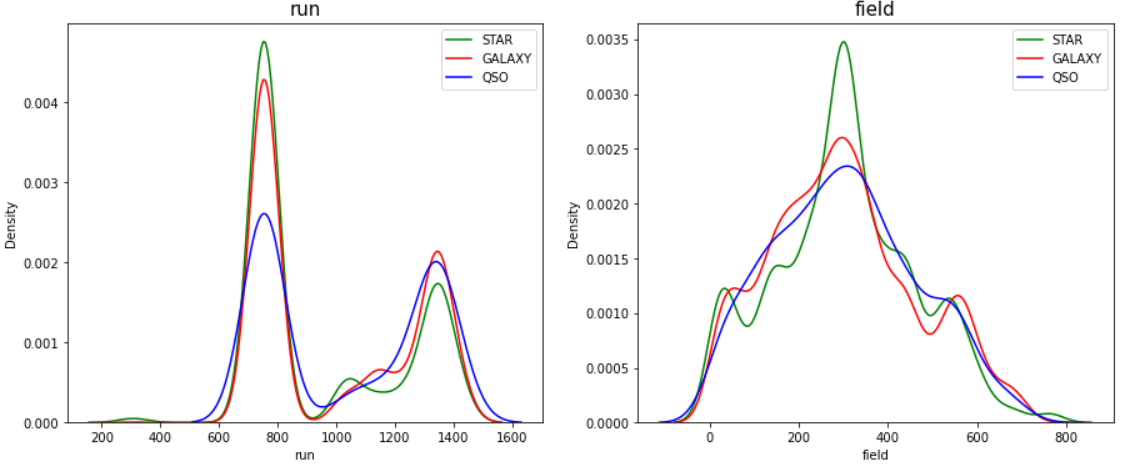
The ‘ra’ column has a slightly higher value when the the celestial body observed is a STAR.

The ‘dec’ field is distributed in similar ratios across all the different classes/categories of the target variable.



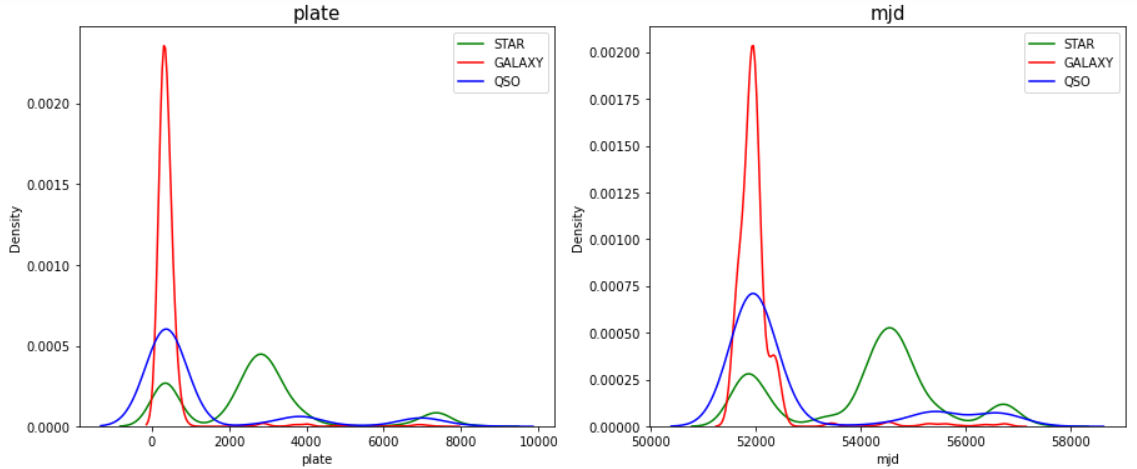
The ‘u’ field is distributed in similar ratios across all the different classes/categories of the target variable.

Values for fields 'g', 'r', 'i', 'z' are slightly higher for the class 'QSO' as compared to others,

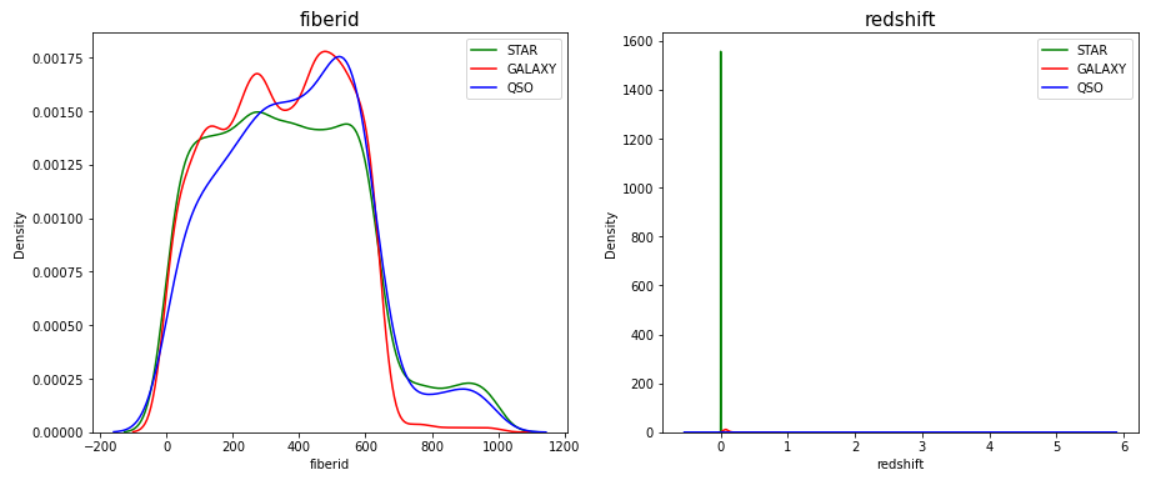


‘run’ column is distributed in similar ratios across all the different classes/categories of the target variable.

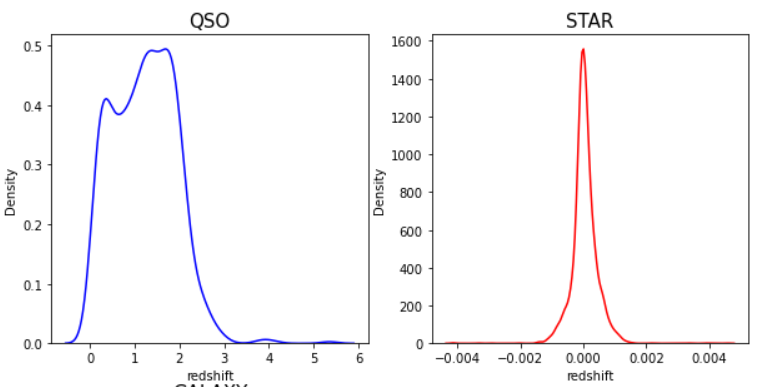
‘field’ column is normally distributed when the celestial body observed is QSO.

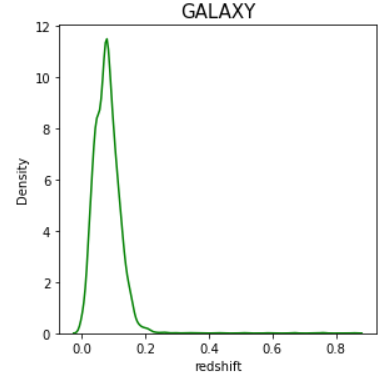


Both ‘plate’ and ‘mjd’ columns have a similar distribution across classes of the target variable.



‘fiberid’ column has a lower value when the the celestial body observed is a GALAXY.



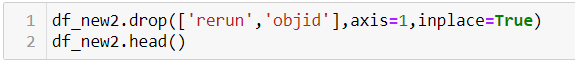


‘redshift’ values for QSO varies from 0 to 6 - mainly distributed around the range (0-3)

‘redshift’ values for STAR varies from 0.0 to 0.8-mainly distributed around the range (0.0-0.2)

‘redshift’ values for GALAXY varies from -0.004 to 0.004 - mainly distributed around the range (-0.002-0.002)

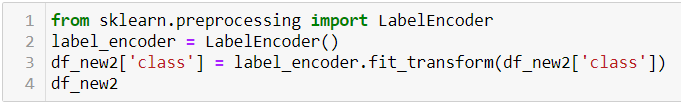
**We can drop columns ‘rerun’ and ‘objid’ since they have only one value**

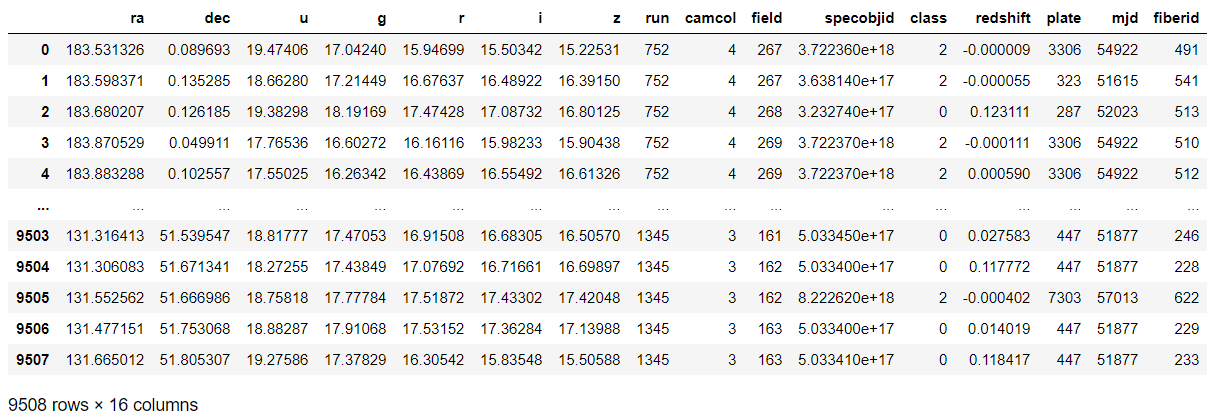
****

**Label Encoding:**

**Label Encoding** refers to the process of converting the labels into numeric form so as to convert it into the machine-readable form.

We use the LabelEncoder module from the sklearn library for this purpose.





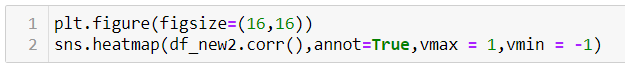
**Correlation:**

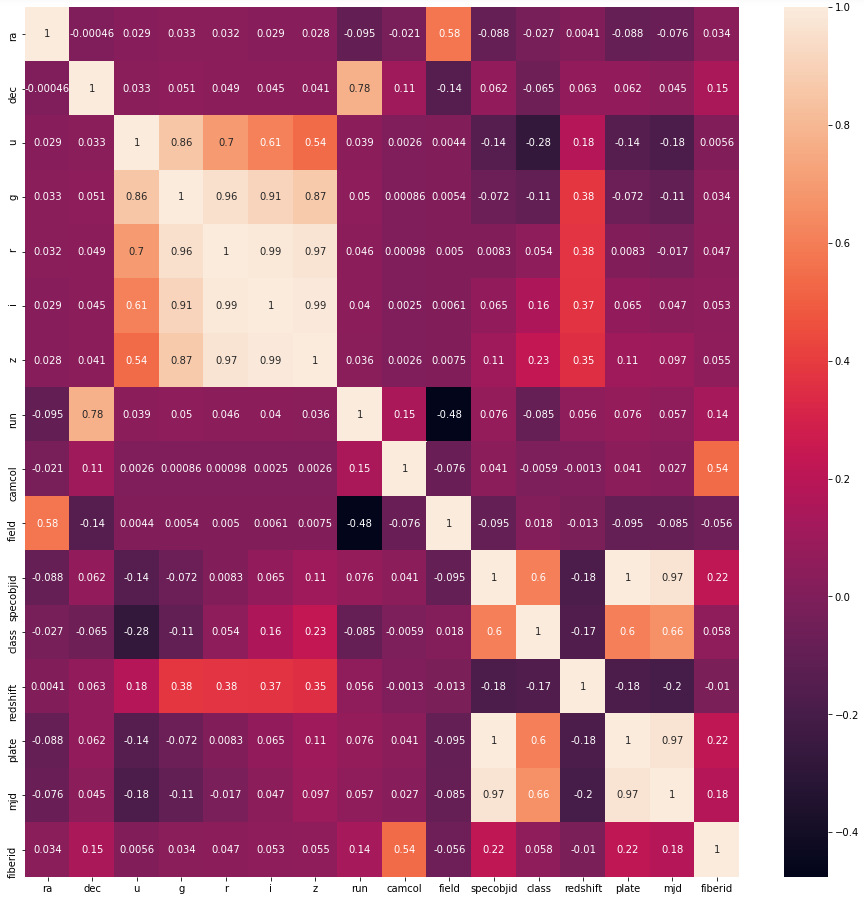
Correlation Matrix is basically a covariance matrix. A summary measure called the correlation describes the strength of the linear association. Correlation summarizes the strength and direction of the linear (straight-line) association between two quantitative variables. Denoted by r, it takes values between -1 and +1.

Correlation value of each column is categorized into mainly 2 parts that are:

* Positive correlated value means that when one variable decreases as the other variable decreases, or one variable increases while the other increases.
* Negative correlated value is the vice versa of positive correlated value.

Correlation matrix is plotted using the seaborn heatmap.





* 'class' column is strongly correlated with 'plate', 'mjd' and 'specobjid'.
* One column out of 'plate' and 'specobjid' since they have a correlation of 1.
* Since 'u', 'g', 'r', 'i', 'z' have similar values they can be reduced using PCA.

**EDA Conclusion:**

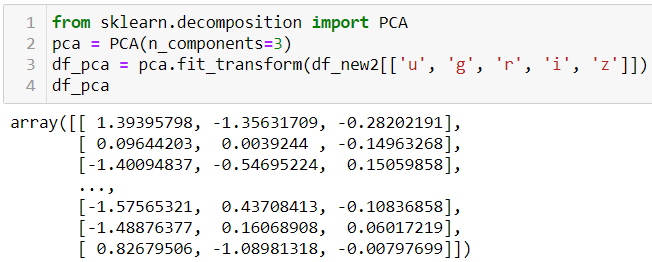
* All but one field are not normally distributed and most of them are multi-modal.
* Galaxies and stars are more frequently and easily observable compared to quasars.
* Stars have higher ‘ra’ values than the others.
* Quasars have higher values for 'g', 'r', 'i', 'z' than the others.
* 'class' column is very strongly correlated with 'plate', 'mjd' and 'specobjid'.
* Columns 'plate' and 'specobjid' have a correlation of 1.

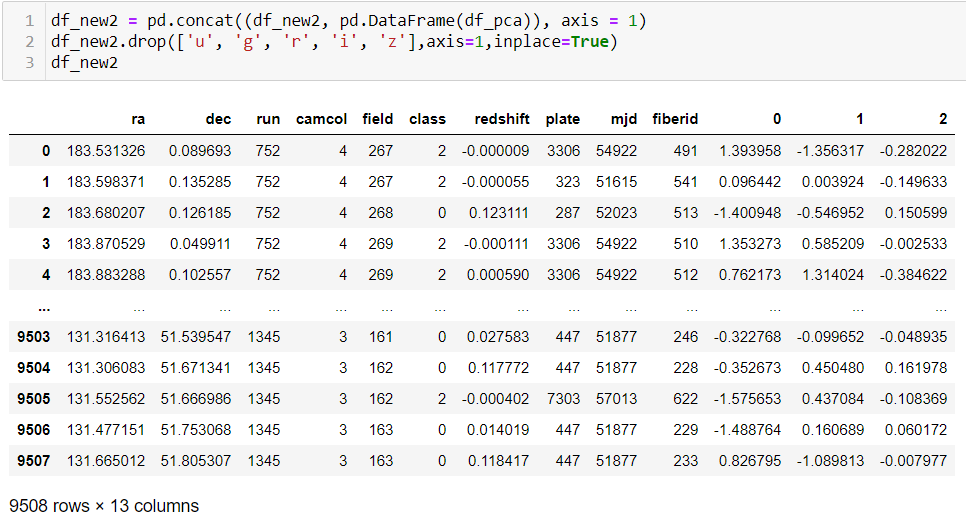
**Dropping the ‘specobjid’ column.**



**Prinicpal Component Analysis:**

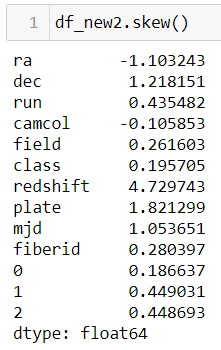
Prinicpal Component Analysis(PCA) is a dimensionality reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.





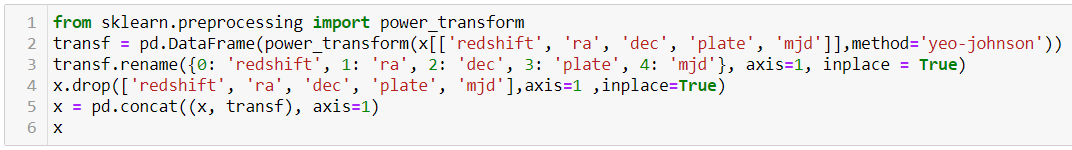
**Removing Skewness:**

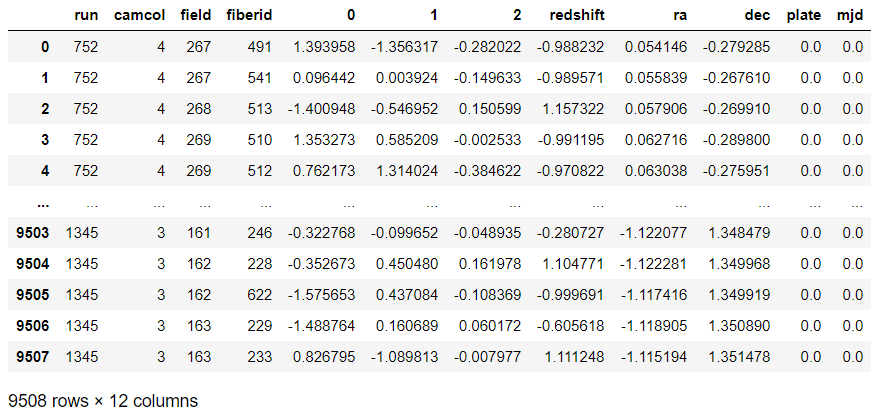
We first check for skewness in the continuous columns.



Columns 'redshift', 'ra', 'dec', 'plate', 'mjd' have skewness more than the threshold(+/-.5)

We can remove skewness in many ways, I am currently using Yeo-Johnson transformation - this is one of the older transformation techniques which is very similar to Box-cox transformation but does not require the values to be strictly positive. This transformation also has the ability to make the distribution more symmetric.

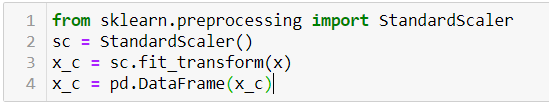




**Feature Scaling:**

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

I have used the StandardScaler (It transforms the data in such a manner that it has mean as 0 and standard deviation as 1) from the sklearn library to scale the features.

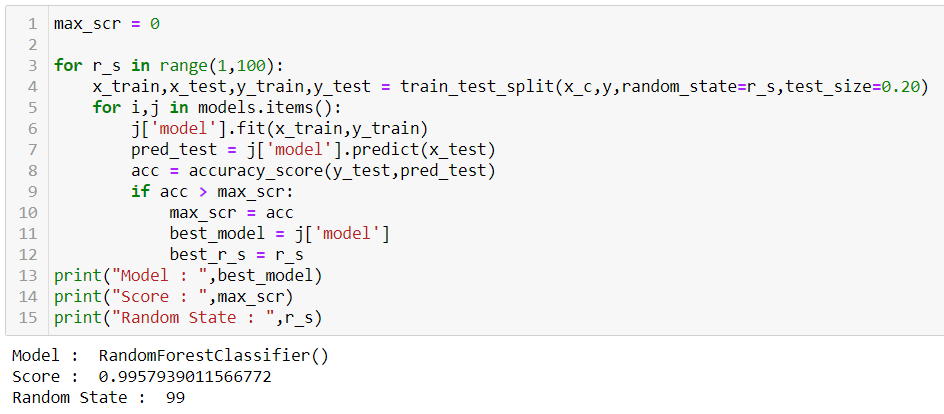


**MODEL BUILDING**

The given problem is a classification, so initially we can define a dictionary with some of best the classifier models and their parameters.

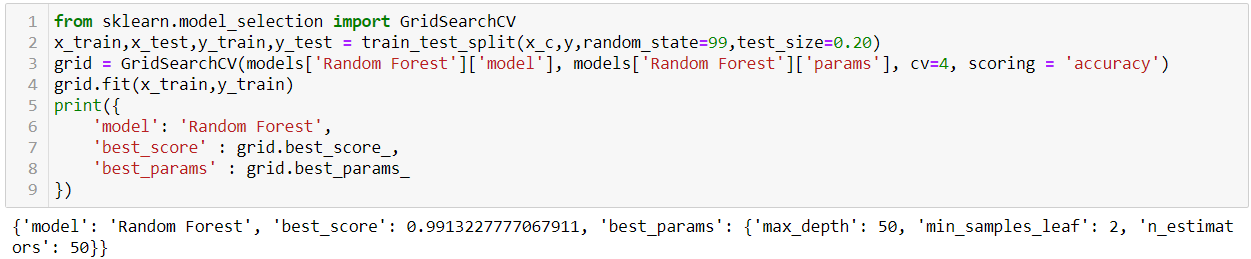


I am using the for loop which help me to provide the accuracy score at each random state and for the best state where accuracy score is maximum is displayed come as output value.

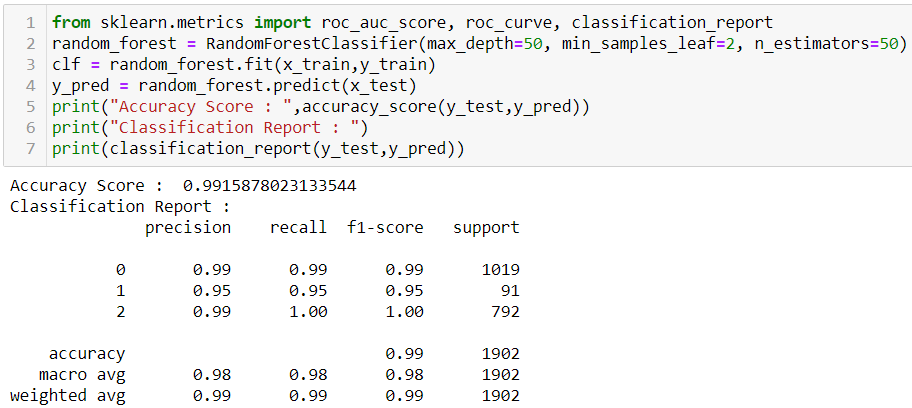


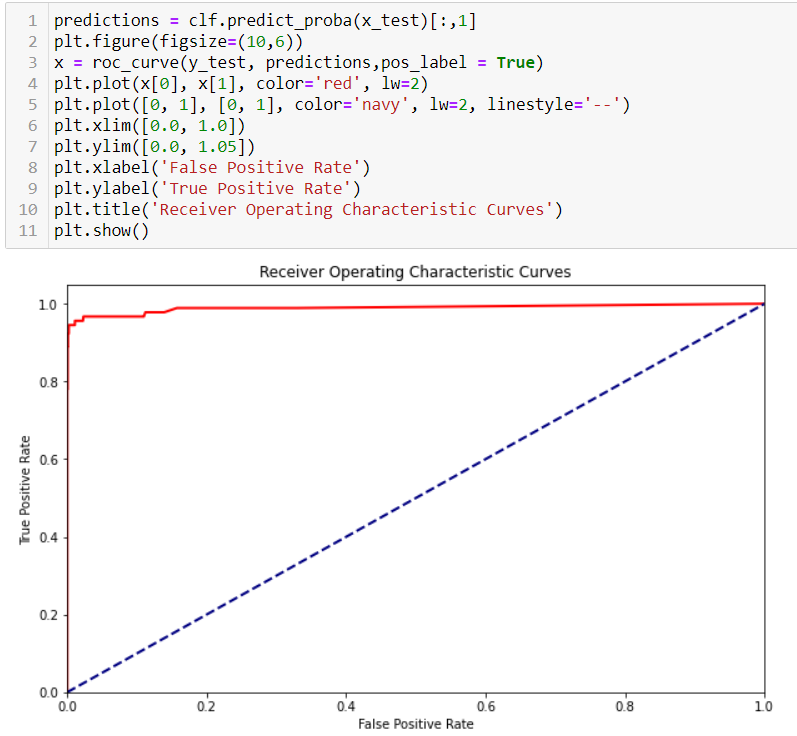
RANDOM FOREST CLASSSIFIER MODEL HAS THE HIGHEST ACCURACY OF 0.99 AND PERFORMS BEST AT RANDOM STATE 99.

After obtaining the best model we find the best parameters using GridSearchCV. GridSearchCV uses cross validation to evaluate the model’s performance and this helps in preventing the over-fitting of the model.



We can now evaluate the model’s performance.





As we can see from above that the model’s performance is excellent.

**CONCLUDING REMARKS**

* Random Forest Classifier was found to be the best classifier for the dataset with an accuracy score of 0.99.
* The Random Forest Classifier model performed best for the following parameters : *max\_depth=50, min\_samples\_leaf=2, n\_estimators=50.*
* The model’s accuracy score was high even after performing GridSearchCV to prevent over-fitting.