**Space Server Dataset Sky Classification**

1. **Problem Definition:-**

In this project the data I have used from Sloan digital sky survey. 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. Our aim is to classify entire data into stars, quasars and galaxies. I have used Jupiter notebook environment to build and apply different machine learning classification algorithms to choose the best among them, which can classify the entire data accurately abiding by the evaluation metrics, within a reasonable amount of time. We have used certain standard evaluation metrics to measure the performance of the classifier. The entire result of this classification is mentioned in this article.

1. **Data Analysis:-**

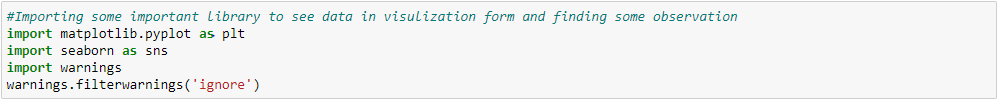
* **Dataset Information:-**

In this dataset we will use classification technique to predict and classify the stars, galaxies or quasars.  
I will use features of the dataset to build the model to predict the required results mentioned above.  
Here are the description of the features present in the dataset:-  
1.objid - Object Identifier  
2.ra = J2000 Right Ascension (r-band)  
3.dec = J2000 Declination (r-band)  
4.u = better of DeV/Exp magnitude fit  
5.g = better of DeV/Exp magnitude fit  
6.r = better of DeV/Exp magnitude fit  
7.i = better of DeV/Exp magnitude fit  
8.z = better of DeV/Exp magnitude fit  
9.run = Run Number  
10.rereun = Rerun Number  
11.camcol = Camera column  
12.field = Field number  
13.specobjid = Object Identifier  
14.class = object class (galaxy, star or quasar object)  
15.redshift = Final Redshift  
16.plate = plate number  
17.mjd = MJD of observation  
18.fiberid = fiber ID

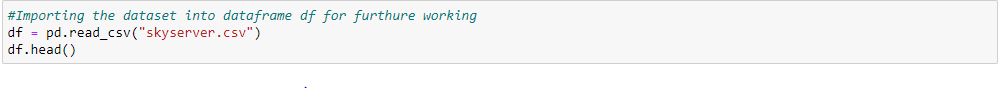
* **Data Import and Analyse:-**

Importing some necessary library-

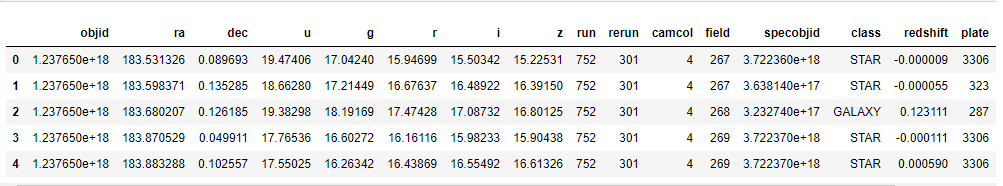


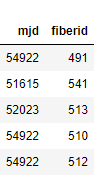


Importing the data-



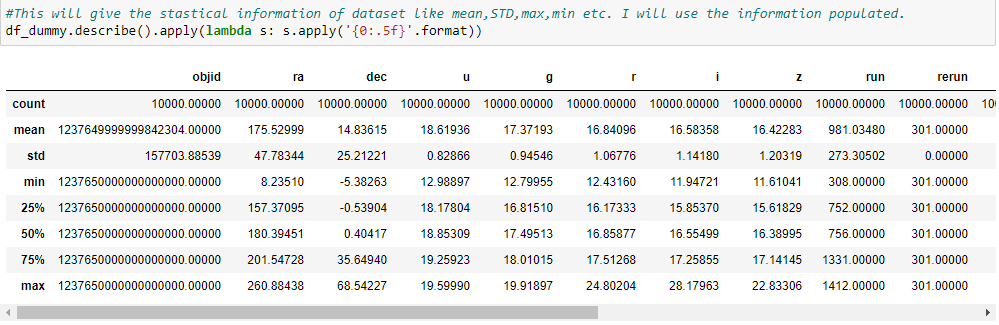
The data is looking like:-





We can see here that there is 18 columns present in the dataset and class column is our target variable which we have to predict by using rest 17 features present in the dataset.

Summary Statistics of the data:-



Key Observation:-

1. The count of values is 10000 across all column which we can see, so by this result the information I am getting that there is no missing value present in the dataset.

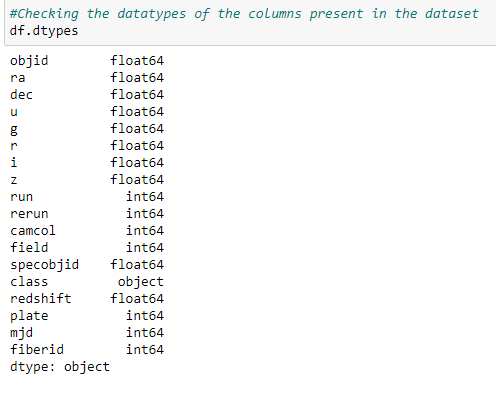
2. Standard deviation is high in the columns like objid ra,dec,run,field,plate,mjd and fiberid. It means the range is possibly high in these columns.

3. We can also see that there is huge differences in min value and max value in the above mentioned columns. So it seems that the data is spread wide and range is high. Possibly the outliers present in the dataset which we will confirm later in data visualization.

4. Median is high than mean in columns like objid,ra,camcol and mean is higher than median in columns like fiberid,mjd,plate,Specobjid etc. It suggest that dataset has skewnees which both positive and negative. Some columns are left skewed and some are right skewed.

5. The difference between maximum and 75 percentile is high in almost all columns which suggest that there must be outliers present in the dataset and the database not normally distributed

Checking data types of all columns present in the dataset:-



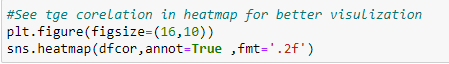
After observing the data types of the column present in the dataset I observed that the all columns data type is integer accept of my target column class which is object. I have to map that column for get more insights from data.

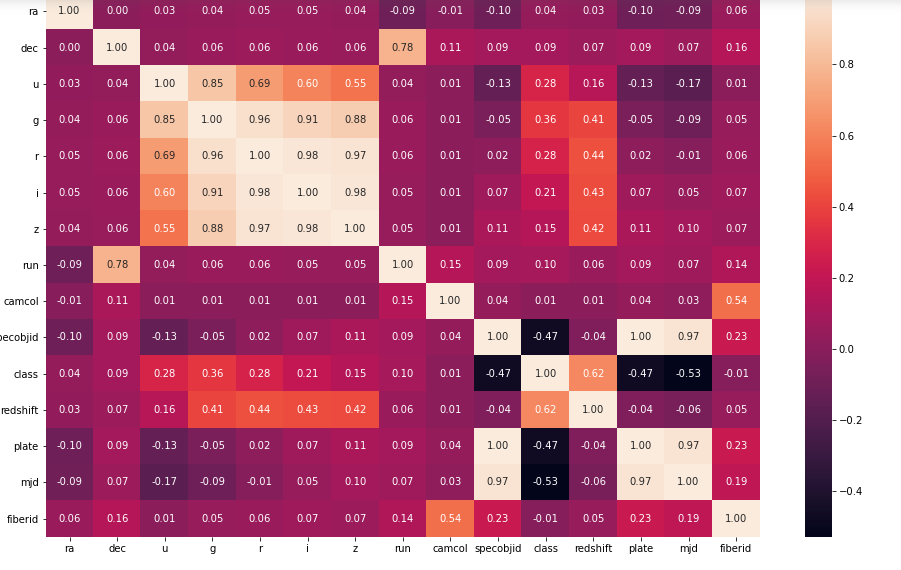
1. **Exploratory Data Analysis:-**

The SDSS database contains two main tables which are of interest to us: SpecObj and PhotoObj. The SpecObj table contains spectroscopic data (including the class) of spectroscopically classified objects, and the PhotoObj table contains photometric data (the colour data) of all identified objects. The SpecObj table contains a small subset of the objects in the PhotoObj table (those that have been classified).

* Checking correlation of all features with our target feature Class.

Here we are checking the correlation of all features with the target class by which we will decide that which feature is good correlating with class and which is bad. It will also give use information that which feature is positively correlated with the target variable and which feature is negatively correlated with target feature.





Observation:-

1. After observing the correlation of other columns with my target column class there is +ve correlation and -ve correlation both are present in the dataset.
2. Highest +ve correlated column is redshift whereas highest -ve correlated column is mjd.
3. Positive correlated columns are :- redshift,camcol,run,z,i,r,g,u,dec,ra
4. Negative correlated columns are: - mjd, plate, specobjid.
5. There are some columns like camcol, run, ra and dec which is very less correlated with my target variable.

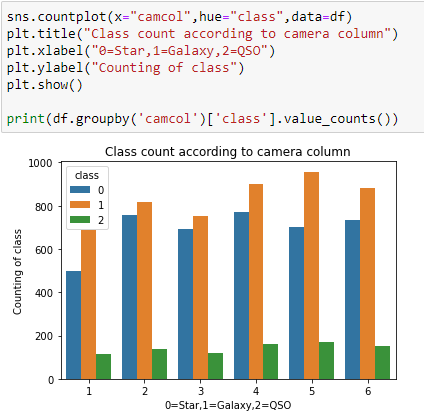
Checking Count of all classes present in the dataset:-

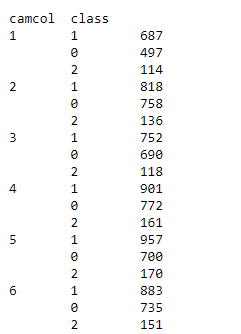


Here we can see that there is 4152 Stars, 4998 Galaxies and 850 Quasars present in the dataset we have.

Checking distribution of different class according to camera column or camcol:-

In this comparison we will see the different class star, galaxy and quasar distribution according to different camcol labels. Also we will see that different class count present according to different camcol label.



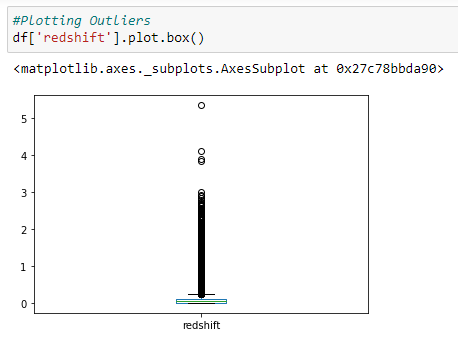


Here are there are 6 labels present in the camcol column which is short form of camera column. The [camera column](http://www.sdss3.org/dr10/help/glossary.php#camcol), or "camcol," a number from 1 to 6, identifying the scan line within the run.

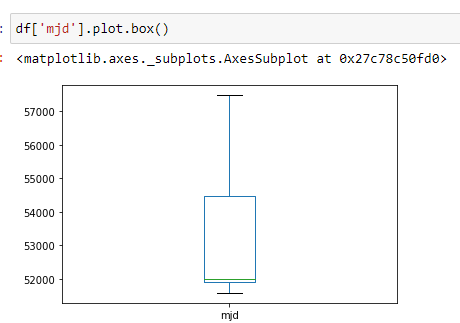
Checking outliers present in the dataset:-

In this we see that if there is any unusual/Outliers data present in the dataset. As if there will be outliers present then that will impact on our model building and model will not perform well and give good accuracy score.

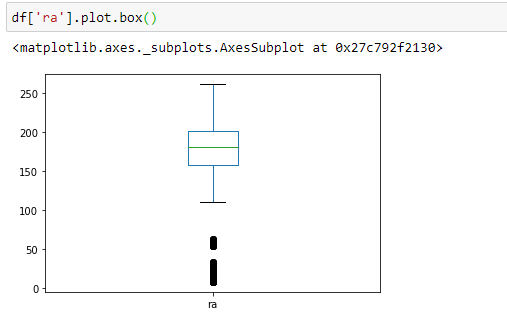
Checking outliers in column redshift:-



Checking outliers in column mjd:-



Checking outliers present in column ra:-

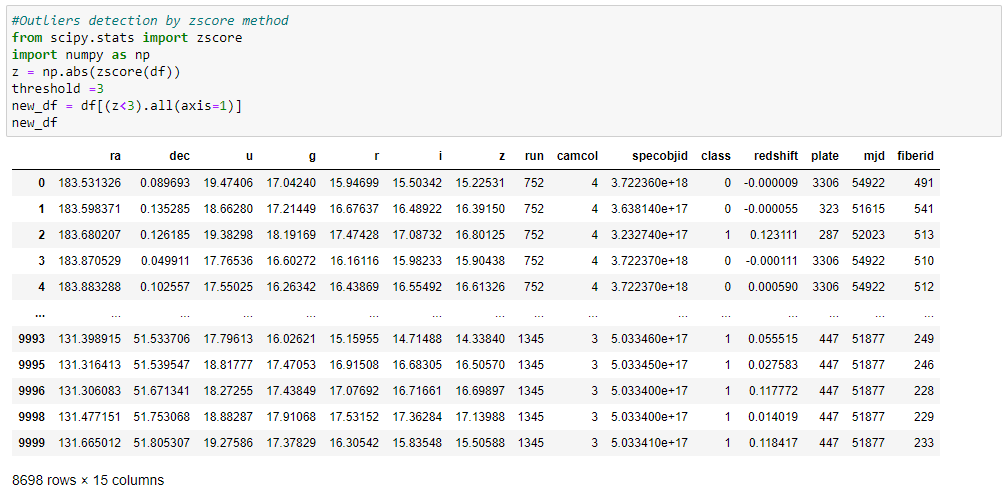


By above observation we can understand that there is outliers present in the dataset and we have to remove the outliers from the dataset to make sure that model will perform well and give good result.

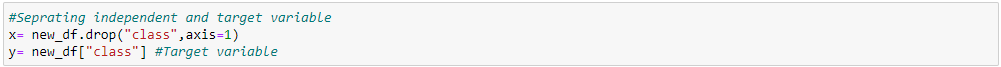
1. **Pre-processing Pipeline:-**

We have to pre-process data before the data is ready for build the model. If there is outliers, skewnees and any missing value present in the data then the model will not perform well and give correct prediction. I will use certain steps to prepare the data for model building mentioned below:-

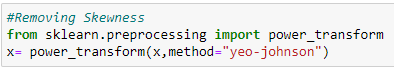
**Step1:-** I will remove outliers by using zscore method by taking threshold value 3 and I will remove those values which zscore is above 3 and select only those data which zscore is under 3. Here for the data frame df I have created new data frame called new\_df. In this new data frame we have all data within the threshold value 3. Below is the code for that. After removing the outliers the shape of new data is 8698 rows × 15 columns.



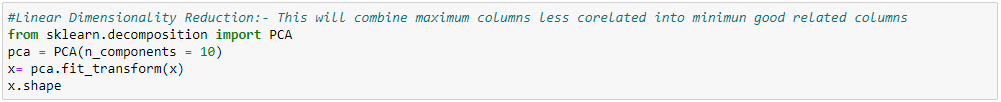
**Step2:-** After removing outliers I will separate new data frame into two part X and Y. In Y I am taking target variable class and in X I have taken rest columns.



**Step3:-** In this step I will remove skewnees from the data so my model will behave well and give good accuracy. Power transforms are a family of parametric, monotonic transformations that are applied to make data more Gaussian-like. This is useful for modelling issues related to heteroscedasticity (non-constant variance), or other situations where normality is desired. Currently, Power Transformer supports the Box-Cox transform and the Yeo-Johnson transform. The optimal parameter for stabilizing variance and minimizing skewnees is estimated through maximum likelihood. I will use power transform with “yeo-Johnson” method to transform the data and remove skewnees. I will remove skewnees in only x variable as the skewnees and variance is necessary in target variable.



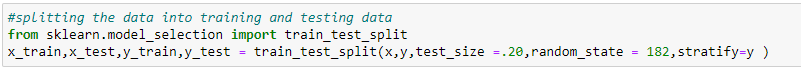
**Step4:-** Principal Component Analysis (PCA) is used to explain the variance-covariance structure of a set of variables through linear combinations. It is often used as a dimensionality-reduction technique. The most important use of PCA is to represent a multivariate data table as smaller set of variables (summary indices) in order to observe trends, jumps, clusters and outliers. This overview may uncover the relationships between observations and variables, and among the variables. I will use PCA to minimize the number of columns to make less and more impactful and effective which is good correlated with the target variables.



1. **Building Machine Learning Models:-**

Now I will start building model and the whole process is divided into some important steps listed below:-

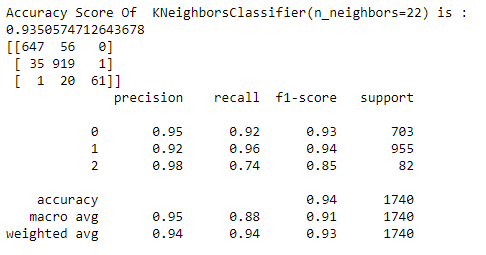
**Step1:- In** this step I will split the data into training and test data. By training data I will train the model and later I will check the performance of model by using test data. I will use train\_test\_split function to split the x variable into train and test data.



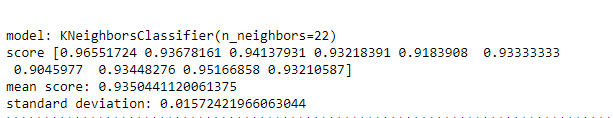
**Step2:-** I will use multiple algorithms and see the model performance and accuracy. As in this project I am using classification techniques so for checking the model performance I will use some important metrics. I will use classification report, accuracy score, confusion matrix and F1 score to see the model performance and decide which algorithm I will select and finalise for model.

Below are the different models used and their performance report:-

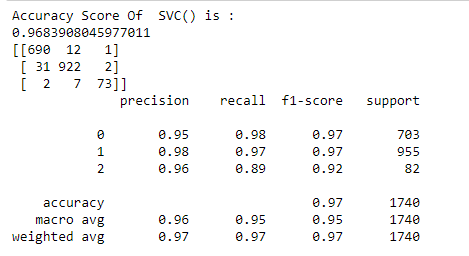
1. KNeighborsClassifier:-



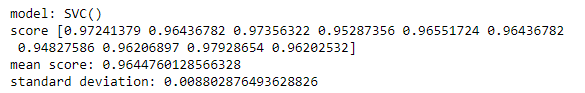
Cross Validation Score:-



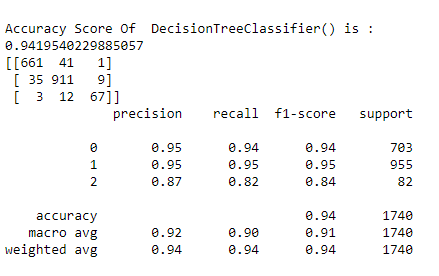
1. Support Vector Classifier :-



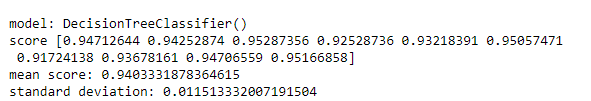
Cross Validation Score:-



1. Decision Tree Classifier:-



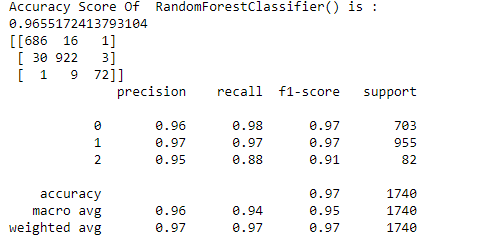
Cross Validation Score:-



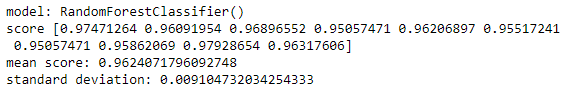
From above all observation I found that SVC giving best score with accuracy score of 96.83% at the same time cross\_val\_score is 96.44%. Now I will use some boosting technique to boost up the score and then I will decide that which I will select and save.

**Ensemble Technique to boost up the score:-**

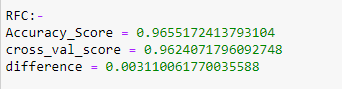
1. Random Forest Classifier:-



Cross Validation Score:-



I checked with multiple ensemble techniques like Adaboostclassifier, GradientBoostingClassifier, and ExtraTreesClassifier etc. I found that after using multiple algorithm I found Random Forest classifier is performing well with below score:



What is Random Forest:-

Random forests are ensembles of decision trees. Decision trees build classification models in the form of tree structures. A dataset is broken down into smaller and smaller subsets, while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Each leaf node represents a classification or decision. It constructs the tree using an information-theoretic entropy function. The entropy function represents the information gain of each of the data points. The internal unit model used in the random forest algorithm is that of a decision tree. The random forest algorithm builds a set of decision tree separately. The algorithm produces different trees by injecting some randomness in the construction of each decision tree. It then combines the predictions of each tree to predict the final class of the object, which reduces the variance and improves the performance of the predictions made on the

Test data.

Reason for Choosing the random forest algorithm:-

The random forest algorithm not only gives good results for binary classification, but it also can be extended for multiclass classification. It does not require feature scaling. The algorithm is capable of capturing nonlinear information inherent in the data. Additionally, it can capture feature interactions. Also we have already seen that this gives best accuracy score and cross validation score.

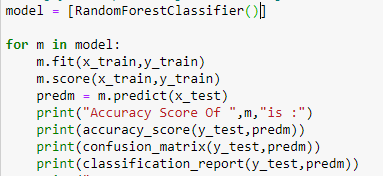
What is Accuracy Score:-

Accuracy score is the score when we test the predicted result of x test data with actual y test data and see how much percentage it is predicting good. As we splitted train and test data into x\_train, y\_train, x\_test and y\_test. So we train the model by using x\_train and y\_train and test with x\_test. When the model give the result then we test it with actual y\_test data and see that how much percentage it has predicted correctly and that score is accuracy score.

What is cross validation score:-

We use cross validation technique which is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. Cross validation score is the score in your validation set. We see that that our model is prevent from over fitting or under fitting we use cross validation technique.

**Code to use for Random Forest Classifier technique:-**



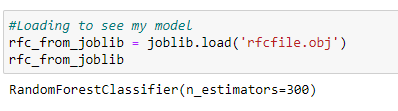
1. **Concluding Remarks:-**

Now I will save the model for production and also we will produce the result after using the same model. As in this case I have selected the Random Forest Classifier so I will save this. Let see in some steps:-

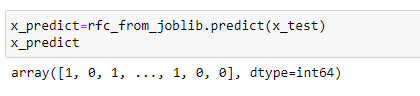
Step1:- In this I will save the model.



Step2:- Loading the Model for production

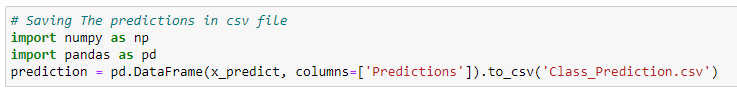


Step3:- Predicting the result:-



Here we have already know that 0= Star, 1= galaxy and 2= QSO (Quasars).

Step4:- Saving the result in CSV(Comma separated value) format.



**Conclusion:-**

The random forest algorithm worked well for SDSS data. We ran the algorithm over the data using the Jupyter Notebook framework. We were able to classify the entire dataset into stars, quasars and galaxies with an accuracy of 96%. These objects could be used to identify the distribution of quasars, stars and galaxies in the sky, which in turn can lead to new insights about them. It seems unlikely that any substantial gain in classification accuracy would result using any other algorithms or approaches on the same dataset, given the classical wisdom that ‘invariably, simple models and a lot of data trump more elaborate models based on less data. However, some significant gains could well occur if more spectroscopically classified data (training data) were available. However, given more such training data, better algorithms may also be possible.

I hope the article will help you to understand how I classified and predict the Sloan digital sky server dataset whether it is star, galaxy or quasar.

Thank You…….

Author:-

Vikash Kumar



The simplest way to describe me would be a technophile. I have an extremely curious mind with a knack to tinker with trending technologies and resolving real-world problems by trying to visualize the problem from a different perspective. Last but not least at the end of the day I really want my work to create a constructive impact on society for its upliftment.