Problem description:

The dataset contains 80,000 samples, as well as spectral properties for each sample and the object type.The goal of this assignment is to categorize sky objects (stars, galaxies, and qua sars) based on their spectral properties. Following statements can be consider as problem statements:

- Exploratory data analysis: Figuring out have missing value or not, find out have outliers or not, feature selection and feature engineering.
- Are there any relationship exists between the features or not?
- Which classification algorithms performance in the class selection has a better accuracy?

This study goes into depth on the process of preparing data, choosing features, designing n ew features, and utilizing predictive modeling techniques to develop a robust model capabl e of properly categorizing the item type based on its spectral properties.

Exploratory data analysis:

Firstly, obj_ID (Object Identifier, the unique value that identifies the object in the image catalog), run_ID (Run Number used to identify the specific scan), rerun_ID (Rerun Number to specify how the image was processed), cam_col (Camera column to identify the scanline within the run),field_ID(Field number to identify each field), spec_obj_ID (Unique ID used for optical spectroscopic objects), plate (plate ID, identifies each plate in SDSS), MJD (Modified Julian Date, used to indicate when a given piece of SDSS data was taken), fiber_ID (fiber ID that identifies the fiber that pointed the light at the focal plane in each observation)

These columns have no bearing on class selection; they are used to identify each sample uniquely, which information is not relevant or has no bearing on our classification problem. As a result, we will no longer consider this column for our categorization dataset.

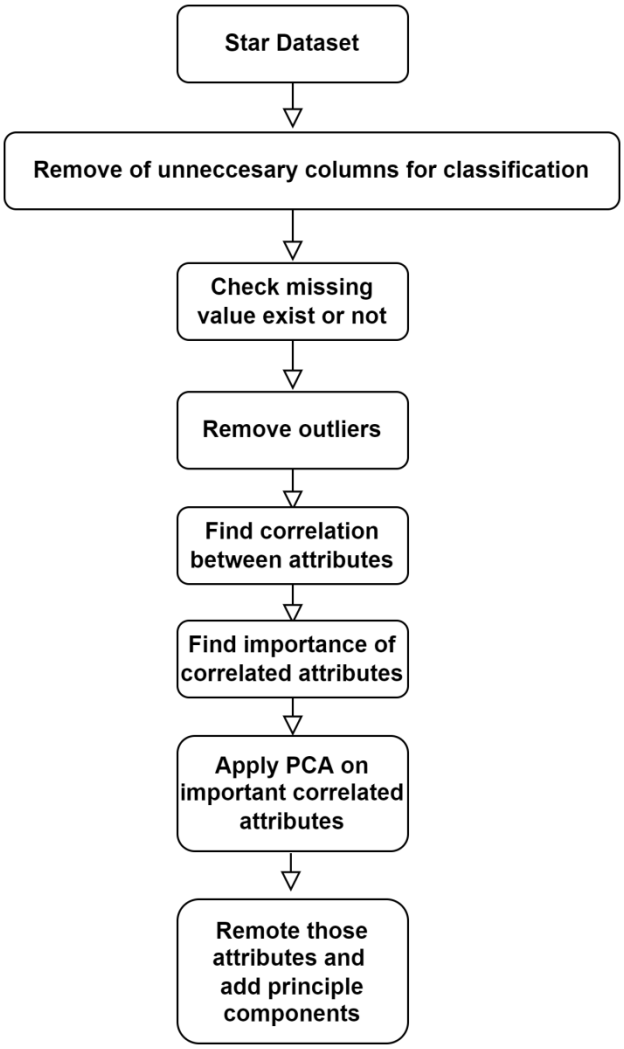
So, in the remaining dataset, the "class" variable is the dependent one, the other 8 variables are independent, and the class variable is dependent on these 8 variables. All eight independent variables are numbers, but the class column was character, thus we converted the values of this column to factors.

If we compare the value with the mean value of the columns and the maximum values of the columns, the minimum value in 'u', 'g', and 'z' seems strange "-9999," which might be deemed an error in data. We investigate the histograms of all the variables. From the graph, we can see that the column alpha distribution resembles a bimodal data distribution, and the delta distribution resembles a bimodal data distribution, but it is somewhat skewed in the first quarter around the values of zero. The distribution of the r and I columns resembles a Gaussian distribution. The class "Galaxy" appears to be the main class, whereas QSO and STAR are minor classes. The 'redshift' distribution is skewed toward zero, with

the bulk of values falling between 0 and 0.6/0.7. Some values are between 1 and 3, while some data may be between 3 and 7. However, u, g, and z appear to have outlier issues; the graph suggests that certain samples are quite far from zero, which we predicted before by seeing the minimum value of this column -9999. Then, for each column, we try to determine how many data points are less than zero. By using these instructions, we attempt to determine how many samles have values smaller than zero. However, only one sample is less than zero in all three columns, and it is -9999 in all cases. So, to avoid this anomaly, we may either remove the sample or set this -9999 to 0. As there is just one sample in the range that is less than zero, we simply trim the column from 0 to their maximum value. Then we find find attributes that are highly corrected using different cutoff values. The absolute correlation between the characteristics 'r,"i,"g,' and 'z' is more than 0.5. The absolute correlation between the attributes 'r,"i,' and 'g' is more than 0.75. (which sometimes consider as ideal). The absolute correlation between 'r' and I is greater than 0.90. Now, we use LDA as a prototype classification algorithm to assess the relevance of the features in order to rank the attributes based on their characteristics. We already have a list of strongly associated traits; once we know how significant the features are, we determine whether to remove them or redesign them.

So, based on the rank of the data, we build new features. Normally, we could erase duplicate characteristics, especially those with high correlation, but the significance plot reveals that features 'r', 'i', 'g', 'z' are all highly significant, and of these four, 'z', 'i' are more important. As a result, because redundant features are critical, we cannot simply eliminate them. Instead, we may incorporate the strongly associated traits into new features, lowering the number of duplicate features while maintaining the most important ones. We know PCA can assist us in achieving this goal. So, we did PCA using the optimal circumstance cut of 0.75.

The first principal component PC1 accounts for 94% of the variation in the data, whereas the second component PC2 accounts for 5%. So we may focus on these two and dismiss the others. So, disregard the three 'r','i','g' characteristics and replace them with these two primary components in the dataset.



Result analysis:

Table: Classification techniques performance according to accuracy and kappa.

Classification Type	**Accuracy**	**Kappa**
KNN	0.9366	0.8869
Linear Discriminant Analysis	0.8519	0.7204
Logistic Regression	0.9641	0.936
Classification Trees	0.9496	0.909
Random Forests	**0.9796**	**0.9637**
Artificial Neural Networks	0.9729	0.9518
Support Vector machine	0.9619	0.9321
Boosting Trees	0.9776	0.9601
ANN-20	0.973	0.952

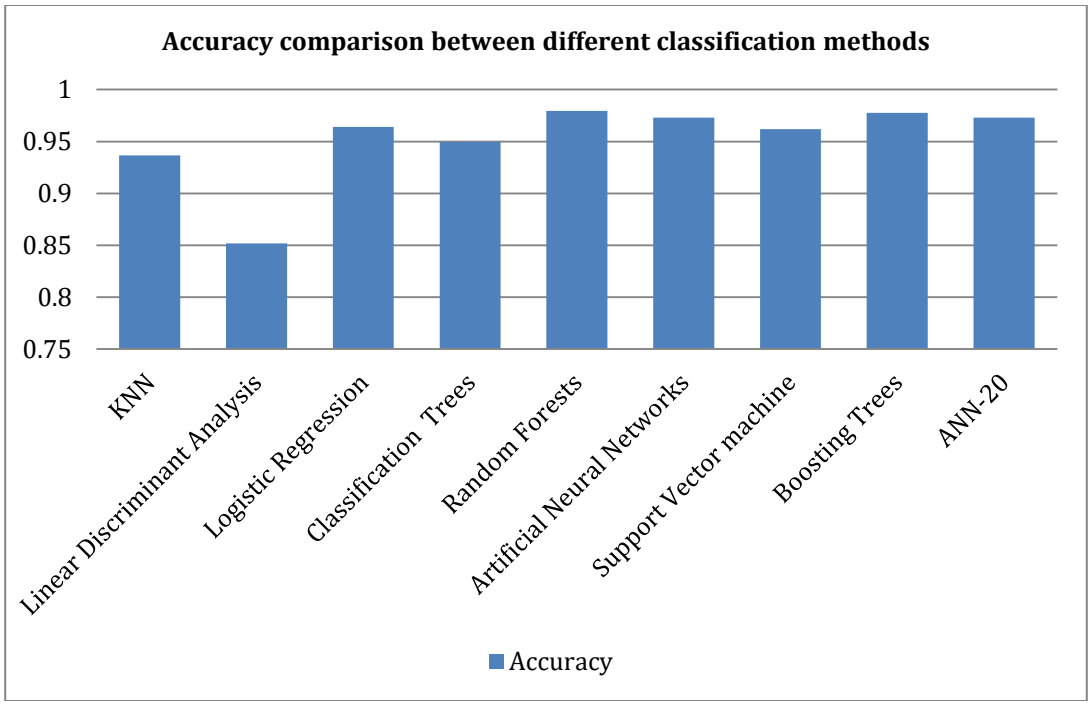


Fig: Performance analysis of different classification methods using accuracy

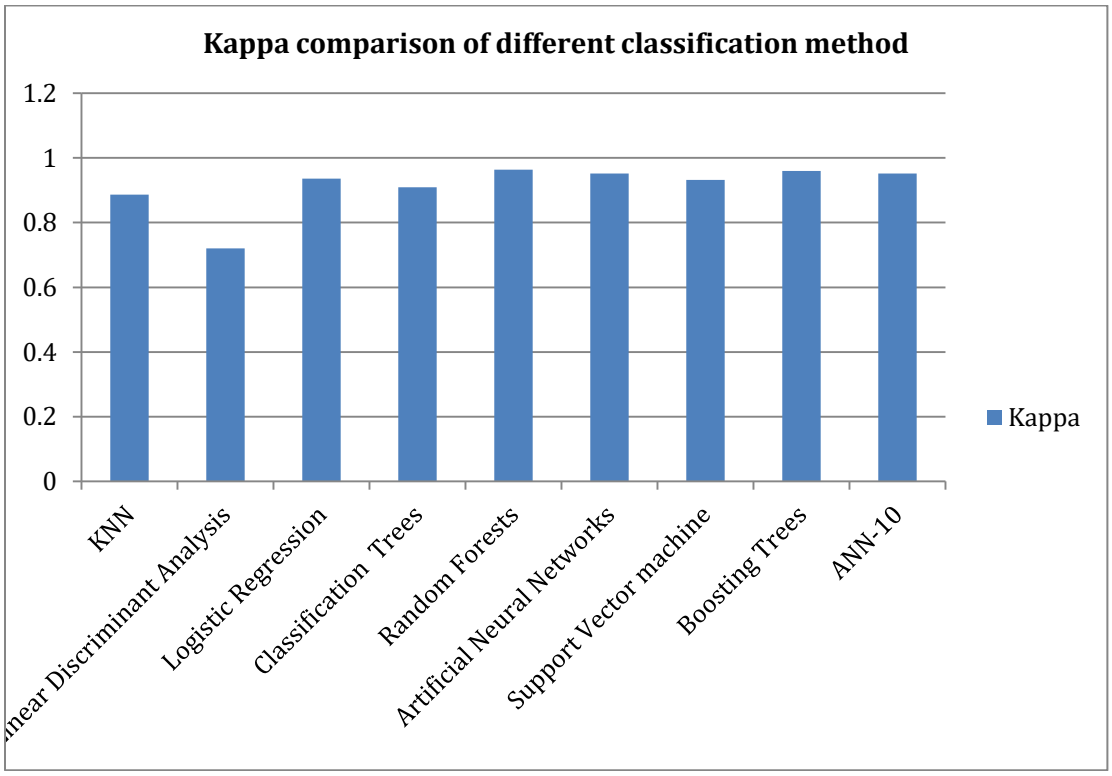


Fig: Performance analysis of different classification methods using Kappa

Table: Classification techniques performance **according to Sensitivity and Specificity.**

Model	Class	Sensitivity	Specificity	Overall Accuracy
KNN	GALAXY	0.9549	0.9135	0.9366
	QSO	0.9114	0.9919	
	STAR	0.9084	0.9723	
Linear Discriminant Analysis	GALAXY	0.9530	0.7122	0.8519
	QSO	0.8396	0.9923	
	STAR	0.5833	0.9679	
Logistic Regression	GALAXY	0.9764	0.9464	0.9641
	QSO	0.8854	0.9896	
	STAR	1.0000	0.9927	
Classification Trees	GALAXY	0.9775	0.9088	0.9496
	QSO	0.8054	0.9897	
	STAR	1.0000	0.9935	
Random Forests	GALAXY	**0.9883**	**0.9670**	**0.9796**
	QSO	**0.9295**	**0.9926**	
	STAR	**1.0000**	**0.9987**	
Artificial Neural Networks	GALAXY	0.9801	0.9625	0.9729
	QSO	0.9222	0.9918	
	STAR	0.9979	0.9933	
Support Vector machine	GALAXY	0.9726	0.9464	0.9619
	QSO	0.8854	0.9905	
	STAR	1.0000	0.9889	
Boosting Trees	GALAXY	0.9879	0.9627	0.9776
	QSO	0.9203	0.9922	
	STAR	1.0000	0.9989	
ANN-20	GALAXY	0.9790	0.9647	0.973
	QSO	0.9261	0.9920	
	STAR	0.9979	0.9920	

Model Selection: To compare the best performing classification technique we consider here from different performance index: Accuracy, Sensitivity, Specificity and Kappa.

- Sensitivity = TP/(TP + FN) = (Number of true positive assessment)/(Number of all positive assessment)
- Specificity = TN/(TN + FP) = (Number of true negative assessment)/(Number of all negative assessment)
- Accuracy = (TN + TP)/(TN+TP+FN+FP) = (Number of correct assessments)/Number of all assessments)
- The kappa coefficient calculates the degree of agreement between categorization and truth values. A kappa value of one indicates complete agreement, whereas a value of zero indicates no agreement. The kappa coefficient is calculated in the following way:

$$\kappa = \frac{N \sum_{i=1}^{n} m_{i,i} - \sum_{i=1}^{n} (G_i \, C_i)}{N^2 - \sum_{i=1}^{n} (G_i \, C_i)}$$

- The letter i represents the class number.
- N denotes the total number of categorized values vs truth values.
- mi,i is the number of truth class I values that have also been classed as class i. (i.e., values found along the diagonal of the confusion matrix)
- Ci denotes the total number of anticipated values in class i.
- Gi denotes the total number of truth values in class i.

After comparing all the model results Random Forest have the highest accuracy so far if we consider the results up to four decimal places. But the time required to train the Random Forest model was the highest one what I realized during the training time. At the same time Random forest archive higher value in kappa, sensitivity and specificity as well comparing to all other methods. We compare the specificity and sensitivity here class wise. If we need to consider the time constraints then the next best performing model is Boosting tree. It takes less time compares to Random forest. But boosting tree archive almost same performance having some different considering the value up to 4 decimal place.

Final Model selection: Considering the time constraint I am going to select boosting tree as a final model over others as it achieved significant performance taking less time in training and prediction.

Conclusion: Comparing all the models we can see that all the models have more than 90 percent accuracy and kappa score except linear discriminant Analysis. And all the models archive significant sensitivity and specificity as well which is over almost 90 percent in all the classification model except LDA.

Note: I didn't compare the classification result of the classification I did on clustered dataset. I applied ANN on dataset I created after k means clustering.