PROJECT REPORT

Title:

Machine Learning Classification of Galaxies and Quasars and estimation of its Photometric Redshift

About the dataset:

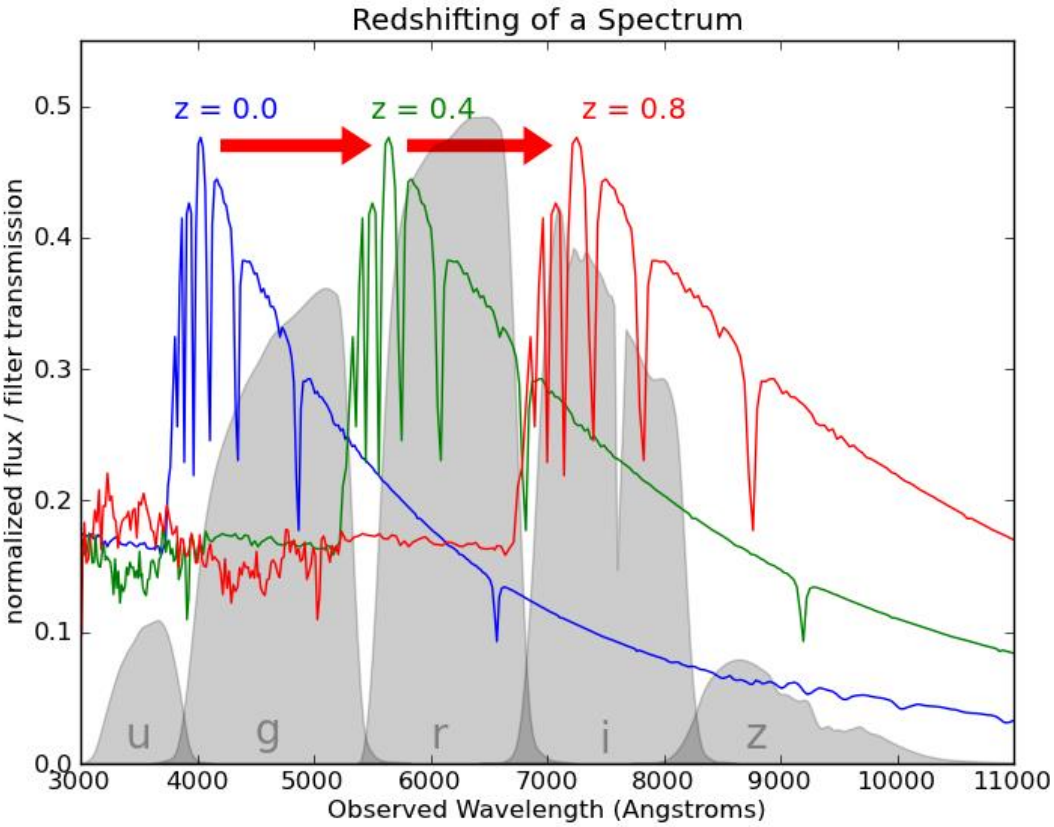
We will be using the dataset obtained from the Photometric Redshift data of the Sloan Digital Sky Survey(SDSS).

The data we are using is a Numpy Binary File consisting of 8 columns. There are 6 feature variable and 1 class variable, which we use for our project.

The description of those variables are given in the table below

Table 1: Variables Description

U	The intensity of light(flux) with a wavelength of 3551 Angstrom emitted by the object
G	The intensity of light(flux) with a wavelength of 4686 Angstrom emitted by the object
R	The intensity of light(flux) with a wavelength of 6166 Angstrom emitted by the object
I	The intensity of light(flux) with a wavelength of 7480 Angstrom emitted by the object
Z	The intensity of light(flux) with a wavelength of 8932 Angstrom emitted by the object
Redshift	**Redshift**, displacement of the spectrum of an astronomical object toward longer (red) wavelengths. It is attributed to the Doppler effect, a change in wavelength that results when a given source of waves (e.g., light or radio waves) and an observer are in motion with respect to each other**.**
Spec_class	Classification of the object as Galaxy or QSO(Quasi Steller Object)


Redshifting of a Spectrum
z = 0.0
z = 0.4
z = 0.8
u
g
r
i
z
normalized flux / filter transmission
Observed Wavelength (Angstroms)
0.0
0.1
0.2
0.3
0.4
0.5
3000
4000
5000
6000
7000
8000
9000
10000
11000

Features and number of classes

The *astronomical colours* (or *colour indices*) is the difference between the magnitudes of two filters, i.e. *u – g, g-r, r-i and i - z*.

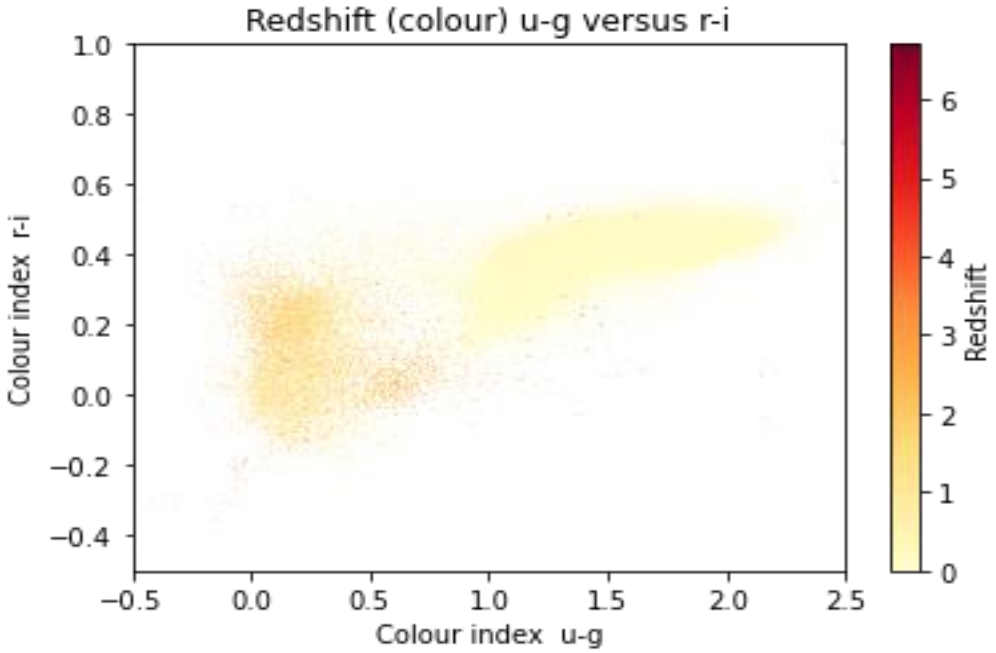
Colour indices act as an approximation for the spectrum of the object and are useful for classifying stars into different types

In this project we will use the colour indices(u-g, g-r, r-i and i-z) as our inputs.

For the regression part we will use a subset of sources with photometric redshifts as our training sample

For the Classification part we will train the model with ‘spec_class’ column which consists of two classes ‘Galaxies’ and ‘QSOs’

There are QSOs (8525) than galaxies (41,475) in the dataset.



Here we plot a colour-index vs colour-index plot using an additional colour scale to show redshift.

It shows that we get reasonably well defined regions where redshifts are similar. If we were to make a contour map of the redshifts in the colour index vs colour index space we would be able to get an estimate of the redshift for new data points based on a combination of their colour indices.

This is the reason why we selected colour indices as our feature to estimate Photometric Redshift using regression.

Cross Validation Techniques

We use two types of Cross Validation methods in this project:

1. **Hold-out validation**: In this we split the data in two, one to test with and other to train with. The measured accuracy will depend on how we split the data into testing and training subsets.

2. K-Fold Validation: Here split the data into k subsets. We train and test the model k times, recording the accuracy each time. Each time we use a different combination of k−1 subsets to train the model and the final kth subset to test. We take the average of the k accuracy measurements to be the overall accuracy of the the model.

We make use of the Kfold library. The **KFold** library is designed to split the data into training and testing subsets. It does this by offering an iterable object that can be initialised with

kf = KFold(n_splits=k, shuffle=True)

The **n_splits=k** specifies the number of subsets to use.

Classification Report

A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report.

The report shows the main classification metrics precision, recall and f1-score on a per-class basis. The metrics are calculated by using true and false positives, true and false negatives. Positive and negative in this case are generic

names for the predicted classes. There are four ways to check if the predictions are right or wrong:

1. **TN / True Negative:** when a case was negative and predicted negative
2. **TP / True Positive:** when a case was positive and predicted positive
3. **FN / False Negative:** when a case was positive but predicted negative
4. **FP / False Positive:** when a case was negative but predicted positive

- Accuracy - Percentage of true predictions:

$$\frac{TP+TN}{TP+FP+TN+FN}$$

- Precision - Percentage of predicted true positive predictions

$$\frac{TP}{TP+FP}$$

- Recall - Percentage of the positive cases that were predicted true:

$$\frac{TP}{TP+FN}$$

- F1-score - Weighted average of precision and recall; the best score of 1 and worst score of 0:

$$\frac{2}{\frac{1}{precision}+\frac{1}{recall}} = \frac{2*precision*recall}{precision+recall}$$

- ‘macro avg’ - Unweighted average of each metric. This did not account for class imbalance
- ‘weighted avg’ - Weighted average of each metric. This accounted for class imbalance.

Linear Regression

For finding the photometric redshift, I applied Linear Regression taking colour indices as my input and redshifts as the training sample also calculated the median residual error of our model, i.e. the median difference between our predicted and actual redshifts, making use of the formula:

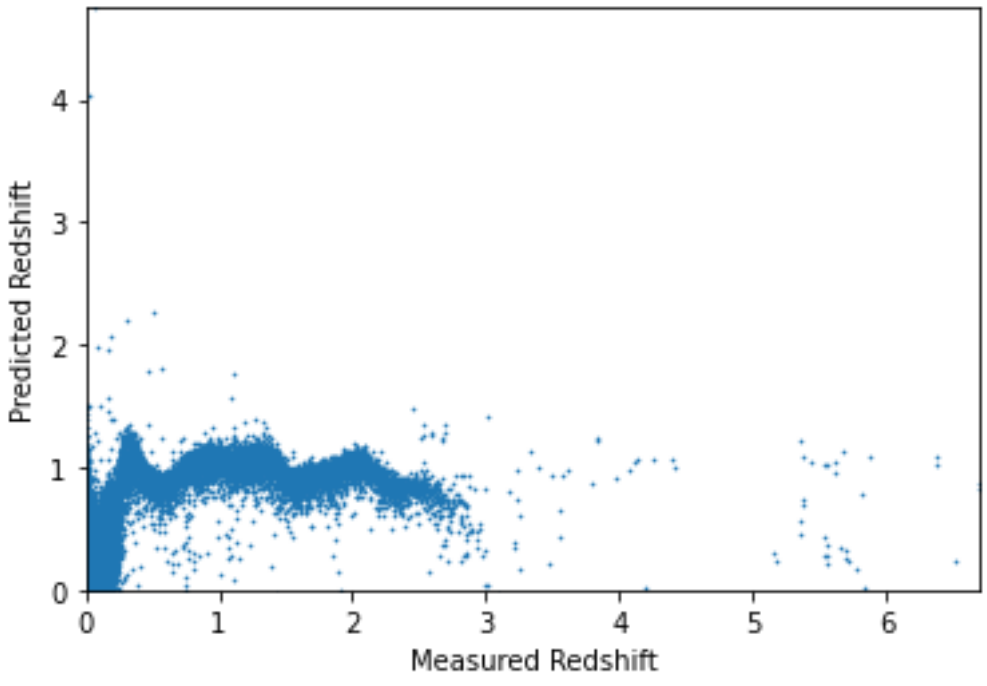
$$\text{med\_diff} = \text{median}(|Yi,\text{pred} - Yi,\text{act}|)$$

I also calculated the coefficient of determination(regression score function) and Mean Squared Error. The results are as following:

Table 2

	Median Difference	Coefficient of determination	Mean Squared Error
Linear Regression	0.18	0.44	0.16

The plot between measured redshift and predicted shift is given below:



We understood that the data is non-linear so we will try different techniques

I also calculated median residuals of Galaxies and QSOs in the dataset.

QSOs have a greater median residual (≈0.451) than the galaxies (≈0.019) the possible reasons are explained in the next section

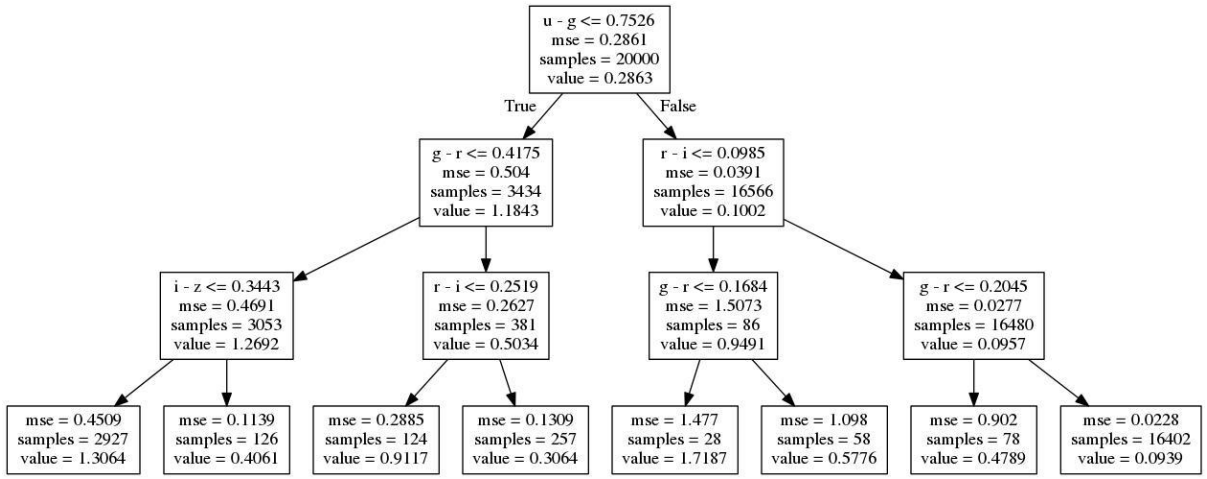
Decision Tree Regressor

Decision trees are a tool that can be used for both classification and regression. Decision trees map a set of input features to their corresponding output targets. This is done through a series of individual decisions where each decision represents a node (or branching) of the tree.

The decision at each branch is determined from the training data by the decision tree learning algorithm. Each algorithm employs a different metric (e.g. Gini impurity or information gain) to find the decision that splits the data most effectively.

The inputs to our decision tree are the colour indices from photometric imaging and our output is a photometric redshift. Our training data uses accurate spectroscopic measurements.

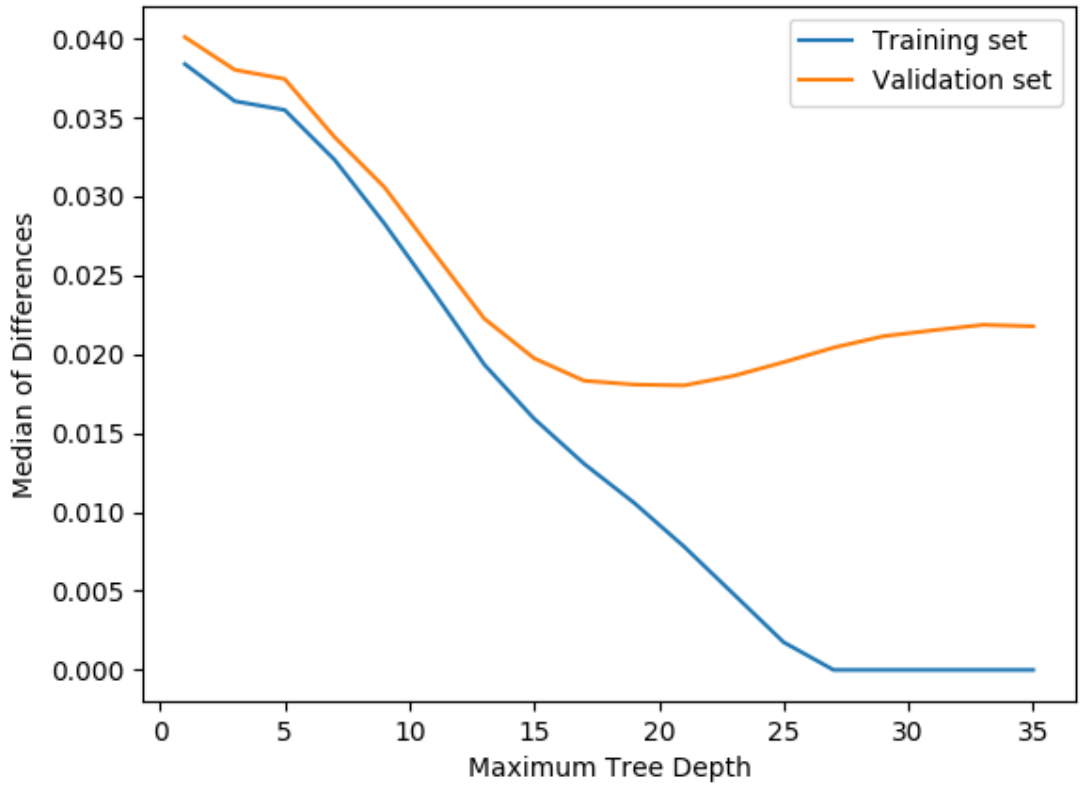
The decision tree will look something like the following.



Eventhough Decision Trees are easy to implement they tend to overfit the data.

So I ran decision trees for various tree depth and then found the median difference between the testing and training data.

The result is as follows:



We can see that the accuracy of the decision tree on the *training set* gets better as we allow the tree to grow to greater depths. In fact, at a depth of 27 our errors goes to zero.

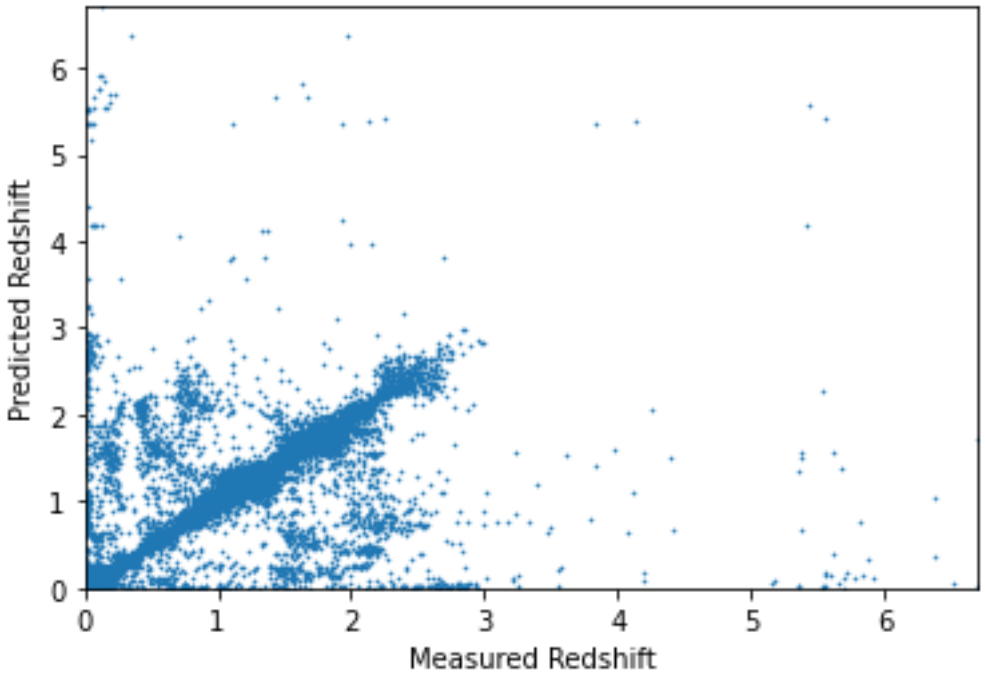
Conversly, the accuracy measure of the predictions for the test set gets better initially and then worse at larger tree depths. At a tree depth ~19 the decision tree starts to overfit the data. This means it tries to take into account outliers in the training set and loses its general predictive accuracy.

So inorder to tackle overfitting we set tree depth to 19.

I also used K-fold cross-validation with k=10, splitting the datasets into 10 subsets. The results are as follows:

	Median Difference	Coefficient of determination	Mean Squared Error
Decision Tree Regressor	0.017	0.93	0.02

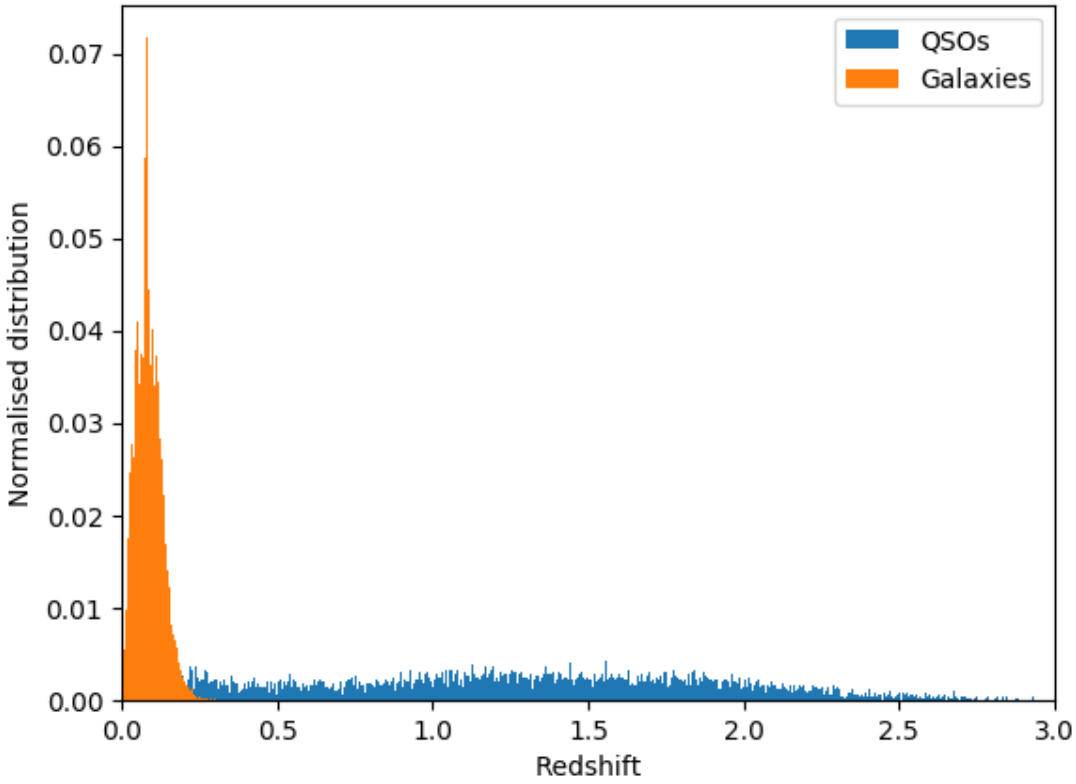
The plot between measured redshift and predicted shift is given below:



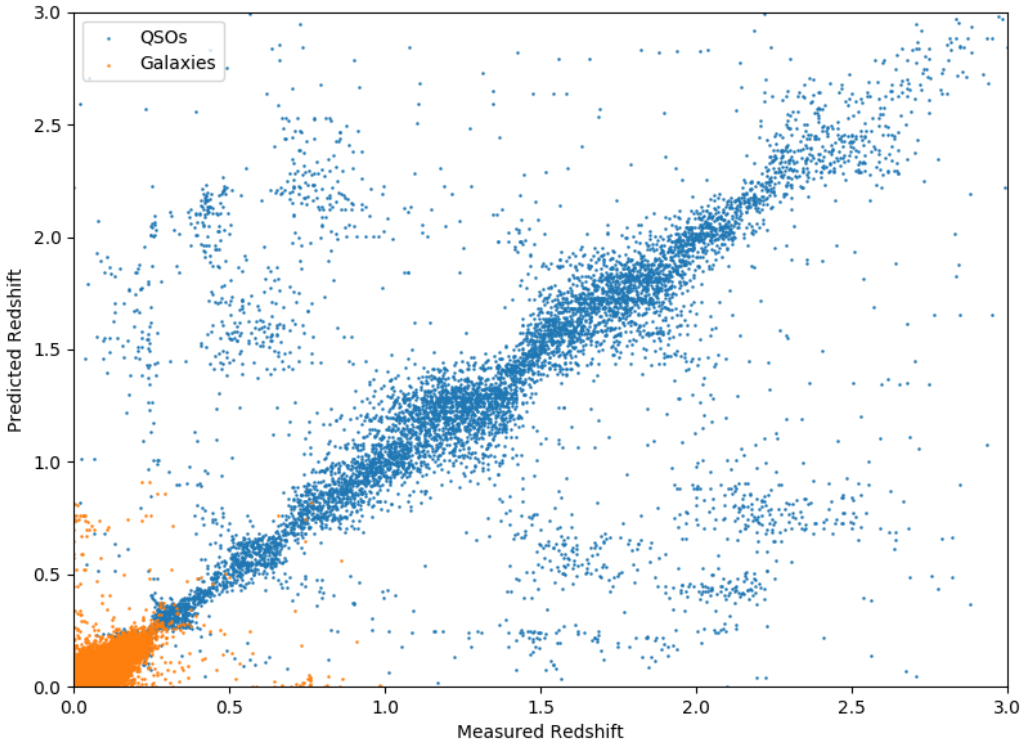
I also calculated median residuals of Galaxies and QSOs in the dataset.

QSOs have a greater median residual (≈0.074) than the galaxies (≈0.016) the possibilities for this are:

- There are far fewer QSOs (8525) than galaxies (41,475).
- Galaxies aren't as bright as QSOs so they become too faint to be detected with SDSS at redshifts ≈ 0.4. This creates a measurement bias.



We can see that the majority of galaxies form a peak around 0.10 while the QSOs are resonably evenly distributed out to redshift ≈2.5. This can lead to a measurement bias. In the case of the galaxies we have trained our decision tree with target redshifts approximately less than 0.4. As such the predictions from this model will not be larger than the maximum target value. So the maximum difference (or residual) for each galaxy in this set will be a lot smaller than the maximum residual for the QSOs.
We can often get a clearer view of this by looking at the predicted redshifts vs actual redshifts in a plot.



Logistic Regression

Logistic regression is an algorithm that is used in solving classification problems. It is a predictive analysis that describes data and explains the relationship between variables. Logistic regression is applied to an input variable (X) where the output variable (y) is a discrete value that ranges between 1 (yes) and 0 (no).

$$0 \leq h_\theta(x) \leq 1$$

It uses the logistic (sigmoid) function to find the relationship between variables. The sigmoid function is an S-shaped curve that can take any real-valued number and map it to a value between 0 and 1, but never exactly at those limits.

Since the dependent class variable is categorical, with two values or classes, I used the multinomial logistic regression model, which used the softmax function and cross-entropy loss function to predict

multiple classifications. I also used K-fold cross-validation with k=10, splitting the datasets into 10 subsets.

Then, a classification report, including precision, recall, f1-score, their macro and weighted averages, and accuracy, along with the cross-validation score were calculated.

```
Classification Report:
              precision    recall  f1-score   support

   b'GALAXY'       0.99      0.99      0.99     41475
      b'QSO'       0.95      0.93      0.94      8525

    accuracy                           0.98     50000
   macro avg       0.97      0.96      0.96     50000
weighted avg       0.98      0.98      0.98     50000


Accuracy: 0.98024
```

As shown by the classification report, all of the performance metrics, with 1 as the maximum and 0 as the minimum, are well above 0.93 and mostly around 0.98, which is a very good performance from a machine learning model by any standards and indicates that the logistic regression model fitted the data well. It is also worth considering the difference between accuracy and f1-score, while accuracy focuses on the true positives and true negative, the f1-score focuses on the false positive and false negative and is a better measure of the falsely classified cases than accuracy. Also, accuracy is generally fit for when the dataset is balanced, whereas the f1-score is a better metric for when the dataset is imbalanced. Since imbalance is a possible issue for the SDSS dataset, the f-1 score is a better metric than accuracy in this study

The confusion matrix heat map of the logistic regression

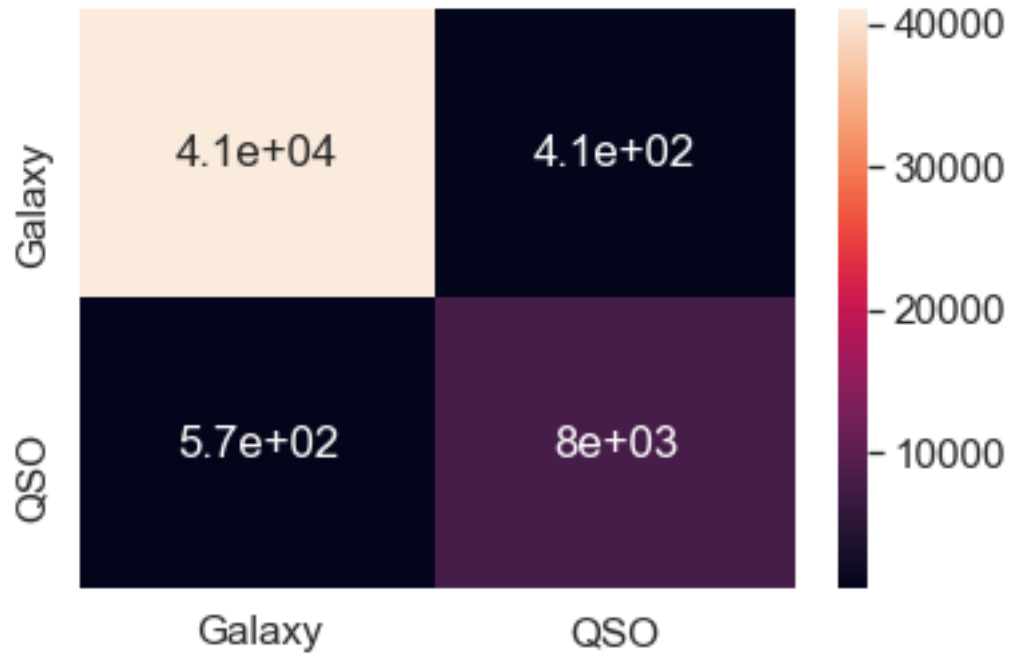


Figure shows the heat map of the multinomial logistic regression model's confusion matrix. A confusion matrix presents the count of the true positives, false positives, true negatives, and false negatives of each class. The two squares at the top left, and the bottom right show the counts, as indicated by the color intensity, of the true positives of the classes, with the rest of the squares showing the counts of false classifications. Overall, the majority, around ninetly three to ninety eight, of the cases in each class were correctly classified, further showing the model's performance is fairly well. However, upon further inspection, it can be seen that each of those two to seven percent of wrong classifications for each class was centered around one particular predicted class. This centering of wrong classifications around one wrong label could indicate intrinsic problems or internal unfitness of the multinomial logistic regression model

Support Vector Machine

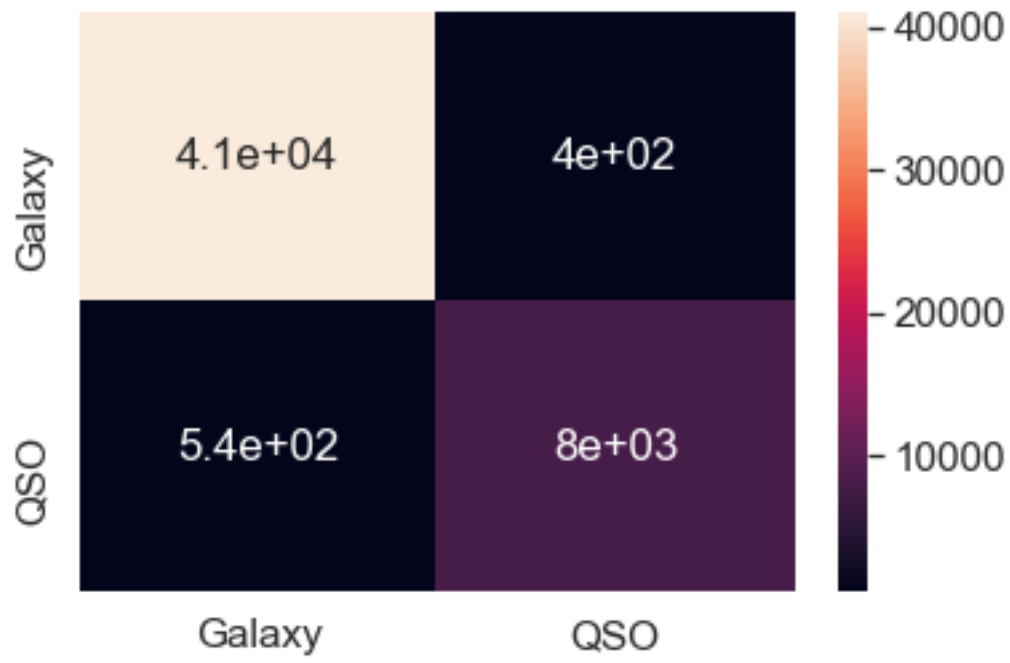
The support vector machine is a model used for both classification and regression problems though it is mostly used to solve classification problems. The algorithm creates a hyperplane or

line(decision boundary) which separates data into classes. It uses the kernel trick to find the best line separator (decision boundary that has same distance from the boundary point of both classes). It is a clear and more powerful way of learning complex non linear functions.

Since Logistic Regression was showing goo results, I tried a SVM Model with linear kernel. I also used K-Fold Cross validation with k=10, splitting the dataset into 10 subsets.

Then, a classification report, including precision, recall, f1-score, their macro and weighted averages, and accuracy, along with the cross-validation score were calculated.

```
Classification Report:
              precision    recall  f1-score   support

   b'GALAXY'       0.99      0.99      0.99     41475
      b'QSO'       0.95      0.94      0.94      8525

    accuracy                           0.98     50000
   macro avg       0.97      0.96      0.97     50000
weighted avg       0.98      0.98      0.98     50000


Accuracy: 0.98132
```



The confusion matrix and classification reports shows pretty much the same results as Logistic Regression. But SVM has slightly better accuracy

Random Forest

A random forest is a collection of decision trees that have each been independently trained using different subsets of the training data and/or different combinations of features in those subsets.

When making a prediction, every tree in the forest gives its own prediction and the most common classification is taken as the overall forest prediction (in regression the mean prediction is used).

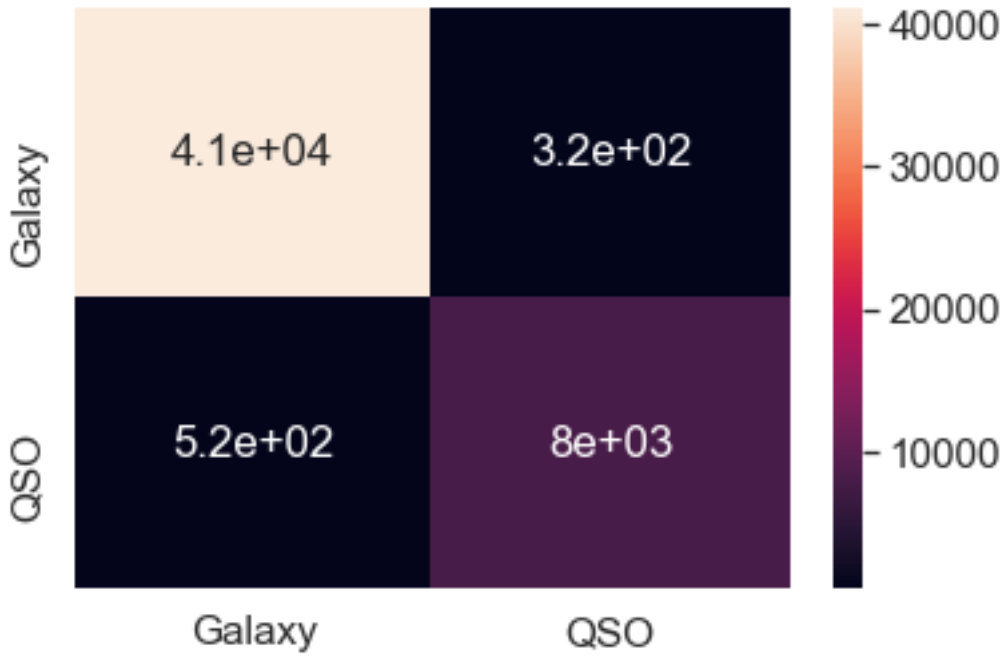
Random forests help to mitigate overfitting in decision trees.

Training data is spread across decision trees. The subsets are created by taking random samples *with* replacement. This means that a given data point can be used in several subsets. (This is different from the subsets used in cross validation where each data point belongs to one subset).

Individual trees are trained with different subsets of features. By using different combinations of input features you create expert trees that are can better identify classes by a given feature.

I made a random forest model with 10 decision trees. I also used 10-fold Cross Validation

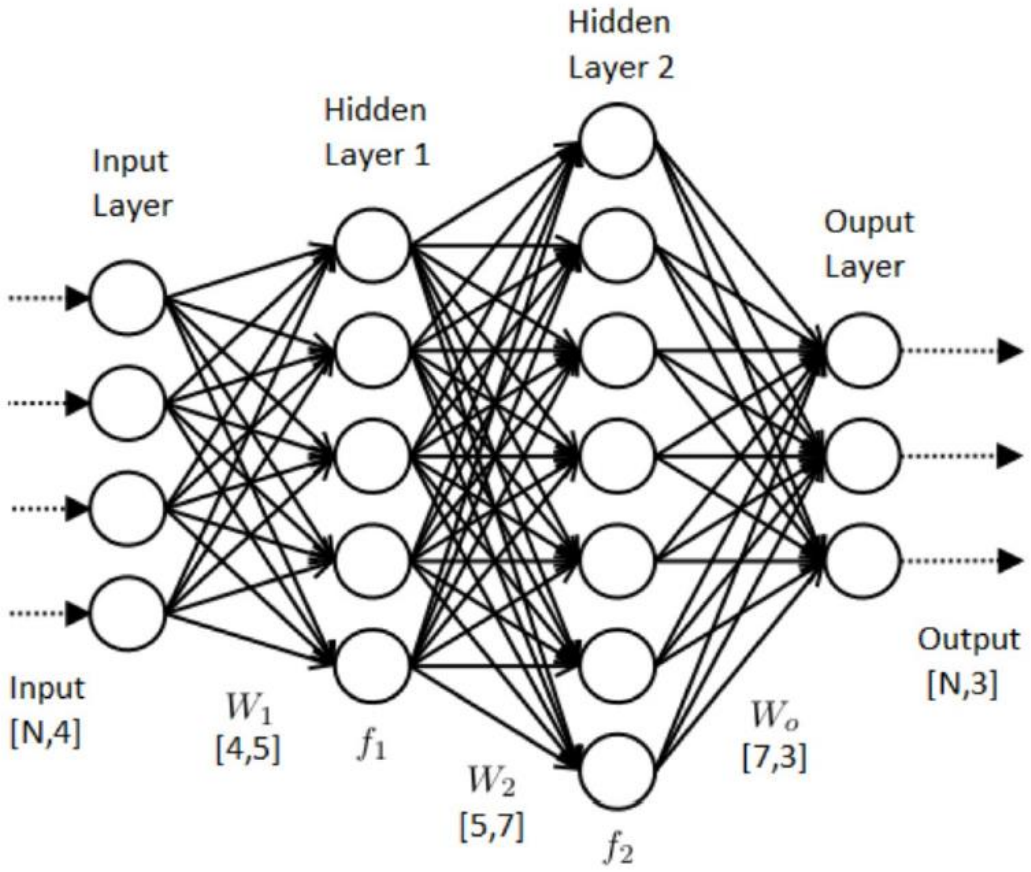
```
Classification Report:
              precision    recall  f1-score   support

   b'GALAXY'       0.99      0.99      0.99     41475
     b'QSO'        0.96      0.94      0.95      8525

    accuracy                           0.98     50000
   macro avg       0.97      0.97      0.97     50000
weighted avg       0.98      0.98      0.98     50000


Accuracy: 0.98334
```



Looking at Classification report, we were able to see improvement in the overall accuracy of the model

Artificial Neural Networks

Artificial neural networks are a set of algorithms with structures that are vaguely inspired by the biological neural networks that constitute the human brain. Their flexible structure and non-linearity allows one to perform a wide variety of tasks, including classification and regression, clustering, and dimensionality reduc tion, making them extremely popular in Astronomy



The above is an illustration of a shallow neural network architecture. The network consists of an input layer, output layer, and several hid den layers, where each of these contain neurons that transmit information to the neurons in the succeeding layer. The input data is transmitted from the input layer, through the hidden layers, and reaches the output layer, where the target variable is predicted. The value of every neuron in the network (apart from the neurons in the

input layer) is a linear combination of the neurons in the previous layer, followed by an application of a non-linear activation function.

I made the Neural Network Model given below:

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 256)               1280
_________________________________________________________________
dense_5 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_6 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_7 (Dense)              (None, 2)                 130
=================================================================
Total params: 42,562
Trainable params: 42,562
Non-trainable params: 0
_________________________________________________________________
None
```

I compiled the model with categorical_crossentropy loss function an d ‘adam’ optimizer. I did 10-fold cross validation and reached an accu racy of **0.9813600063323975**
I also found the f1, accuracy and precision scores:

```
y_predannf = np.argmax(ypredann,axis=1)
yann_ta = np.argmax(dummy_y,axis=1)
yann_ta
 #Print f1, precision, and recall scores
print("Precision Score=",precision_score(yann_ta, y_predannf , average="macro"))
print("Recall Score=",recall_score(yann_ta, y_predannf , average="macro"))
print("f1_score=",f1_score(yann_ta, y_predannf , average="macro"))#Yessss baby
```

```
Precision Score= 0.9822410167664197
Recall Score= 0.9684530305681796
f1_score= 0.9751823929860267
```

Unsupervised Learning

Unsupervised Learning is a general term that incorporates a large set of statistical tools, used to perform data exploration, such as clustering analysis, dimensionality reduction, visualization, and outlier detection. Such tools are particularly important in scientific

research, since they can be used to make new discoveries or extract new knowledge from the dataset.

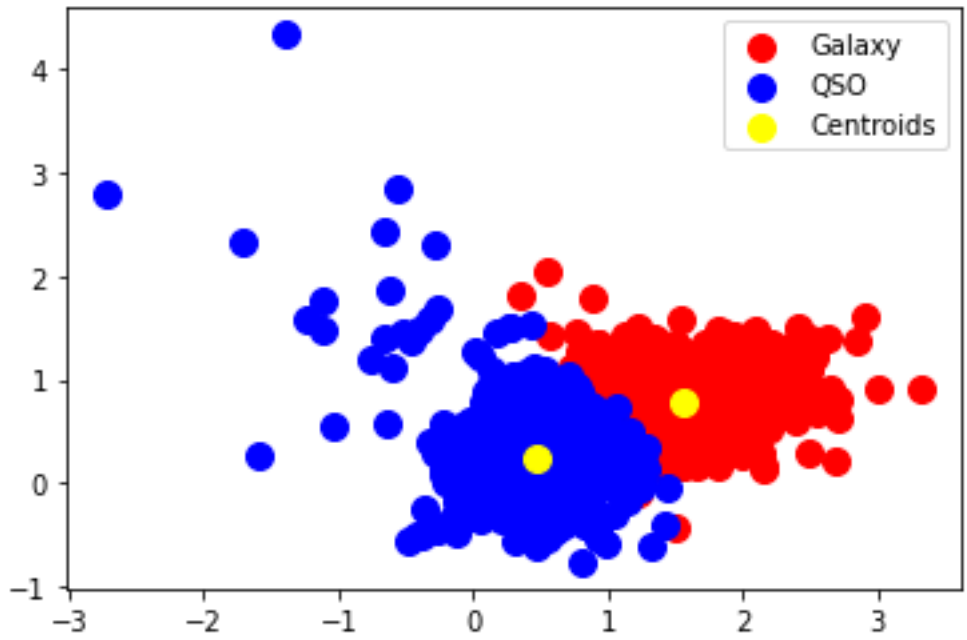
Clustering

Clustering refers to grouping similar data points together, based on their attributes or features.

K-Means Clustering

k-means clustering is a distance-based algorithm. This means that it tries to group the closest points to form a cluster.

I tried K Means clustering with input data as colour indices giving number clusters=2(since we have 2 classes). I also normalised the data before doing clustering since it is a distance based algorithm



This is the plot I got after clustering the data and plotting it against the first 2 features.

After comparing the output variable with my target variable which I used for supervised techniques, I found out the accuracy of the manually i.e by checking the number of classes that were correctly clustered

Accuracy=number of classes correctly clusetered/total number of classes

and I got an accuracy of **0.91316**

I also calculated the Rand Index, which computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings

It was 0.6180

Gaussian Mixture Model

Gaussian Mixture Models (GMMs) assume that there are a certain number of Gaussian distributions, and each of these distributions represent a cluster. Hence, a Gaussian Mixture Model tends to group the data points belonging to a single distribution together.

I tried Gaussian Mixture Model with 2 components and then found out its Rand Index

Its Rand Index was: 0.7334398326062945

Comparative Analysis

Estimation of Photometric Redshift:

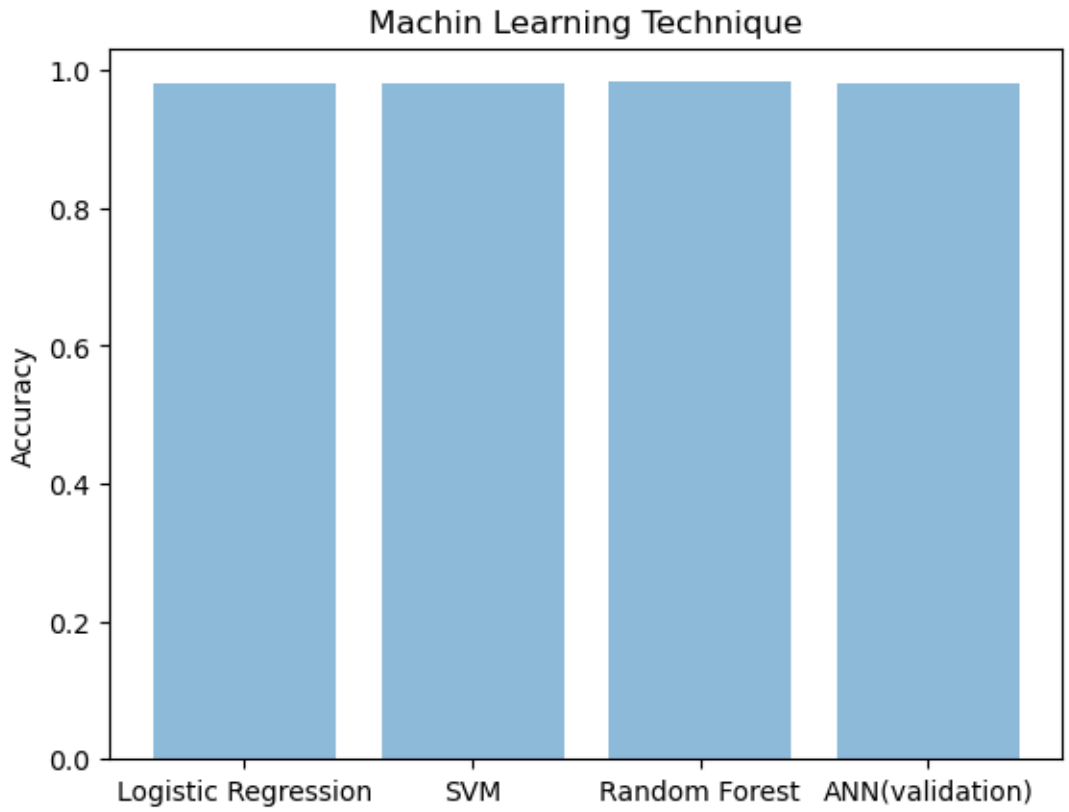
I tried this problem with two techniques, Decision Tree Regressor were giving much better results than Linear Regression.

But decision trees were prone to overfitting. By comparing the accuracy of the model on the training set with that of the test set for different tree depths we found that a maximum tree depth of 19 was suitable for our model.

We also looked at k-fold cross validation and the various methods that can be used to implement it. k-fold cross validation mitigated the risk that the training set has a unique or specific population of the data set; For example if all the training data contained QSOs and the testing set regular galaxies. k-folds cross validation also allowed us to get a prediction for all the points in your data set.

We also looked at the sub-population of QSOs and how their accuracy measurement was significantly worse than that of the other galaxies. On closer inspection we found that this was a measurement bias resulting from the difference in the range of redshifts in each population.

Classification using Supervised Learning



Machine Learning technique	Accuracy
Logistic Regression	0.98024
SVM	0.98132
Random Forest	0.98334
ANN	0.98136

- All 4 methods did pretty well in classifying the data
- Logistic Regression had the least accuracy among the four but it is a more simpler and faster algorithm
- SVM also did well it is faster and simpler than Random Forest and ANN but slower than Logistic Regression. It can handle outliers better than Logistic Regression
- Artificial Neural Network had more accuracy than most of the other techniques and but it is in the same time more complex than the other methods used. Since we were using a simple

binary classification problem I didn’t find much advantage in using ANN over other techniques

- Random Forest outperformed all other methods in Classification but it is still more complex and took more time than Logistic Regression and SVM

Unsupervised Learning:

For this dataset, which was measuring what were probably normally distributed features, the GMM clustering worked better at finding the actual species labels, as measured by the **adjusted Rand score**.