**Question #1)**

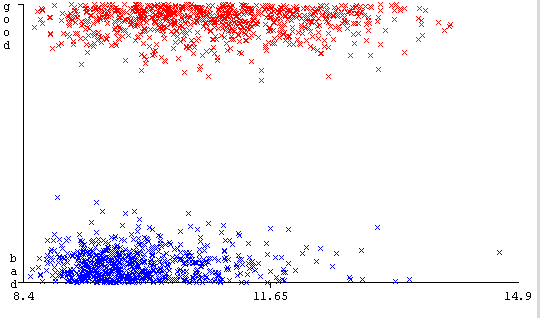
The attributes that do not appear to have a “hump” distribution:

* Citric Acid
* Total Sulphur Dioxide
* Free Sulphur Dioxide

These attributes appear to have outliers:

* Residual Sugar
* Chlorides
* Free Sulphur Dioxide
* Total Sulphur Dioxide
* Sulphates

The attribute that appears to be most useful in determining the quality of the wine would be alcohol content as the good quality tend to be more clumped together at the lower end of the scale where at the bad win tends to be more spread out.

****

**Question #2)**

The accuracy achieved by ZeroR was 53.4709%. This means that the ZeroR classifier, which is the simplest classification method, got an accuracy of 53.4709%. This will be used as a baseline benchmark for other classifications to determine their performance.

**Question#3)**

The alcohol attribute was the first split of the decision tree and thus contains the most useful data to determine the quality of the wine and this result does match my prediction in question one. The split value was <=10.5 for good, this was also right in my prediction of the clustering on data

**Question#4)**

Use Training Set) Correctly Classified Instances 1465 91.6198 %

Cross-Validation) Correctly Classified Instances 1182 73.9212 %

**What is 10-fold Cross validation?**

Cross validation is the basis of understanding what 10-fold cross-validation is. Cross-validation is when a data set is split into an arbitrary number of sections, the number of splits is called “k”. The classifier is then tested on the selected section of split data, the remainder of the data is used for training. Once the first section has been used for testing and the remainder has been used for training the next section is then selected and used as testing while the remainder is used for training. This is done till all section have been both used as training and testing data. Once all sections have done this, the average of the test results are what are displayed as “Correctly Classified Instances”. 10-fold cross-validation means that the “k” value is 10, thus meaning the data set is split into 10 section and the model is run 10 times to go through every section.

**What is the main reason for the difference between the percentage of Correctly Classified Instances when you used the entire training set directly versus when you ran 10-fold cross-validation on the training set?**

The main reason between the two would be that the model trained on the entire data set was also tested on data that the model was already trained on. This leads to major pitfalls in the validity of the model, as instead of testing whether the model knows how to classify new data, you’re testing it on whether is sort of “remembers” what each data point was. This is known as Overfitting. Cross-validation trains and tests the classifier on all data but in sections and the averaged so the training data is never the same as the data used to test that model.

**Why is cross-validation important**

Cross-validation is very important as it allows the model to be trained on all the data but also tested on all the data, which makes the results from the cross-validation a lot more reliable if the model is fed new data.

**Question#5)**

“RandomForest -P 100 -I 200 -num-slots 12 -K 0 -M 1.0 -V 0.001 -S 1”

At 10-fold Cross-validation: Correctly Classified Instances 1330 83.177 %

**Question#6)**

After trying different classifiers, the tree classifiers tended to yield higher accuracy, after trying few “RandomForest” gave the best result. I then modified the parameters so that the number of iterations “Trees in forest” was higher and got to 200 being the optimal number but this was taking a significant amount of time so I increased the amount of threads Weka could use for the modelling to the max available on my machine, 12 threads.

**Question#7)**

bagSizePercent – Determines how big each bag is in relation to training set size

batchSize – if batch prediction is being performed, this is the preferred number of instances

breakTiesRandomly – when attributes look equally as good this can be set to true, to break ties at random

calcOutOfBag – Boolean variable to calculate out of bag error

maxDepth – tells the model how far down its allowed to go and the default is 0 which is unlimited

numDecimalPlaces – Tells the model the accuracy of decimal places we would like

numExecutionSlots – This tells the model how many threads of the CPU it can use to run each modelling

numFeatures – Tells the model if and how many randomly chosen attributes it should pick for the model

numIterations – Tells the model how iterations it should do do per model

No other parameters are relevant or worth explaining

maxDepth was set to 5 this decreased the accuracy to 76.7355%, this would be due to the model being limited in how thorough the tree can be

numIterations was set to the default 100 which decreased the accuracy to 82.8643%, this would be because the model isn't going through as many iterations which allows to pick up on things better

breakTiesRandomly was set to true this brought the accuracy to 82.7392%, this would be because instead of figuring out the difference between attributes that look equally as good, it is randomly throwing in unpredictability into the model