K Nearest Neighbors Wine Classifier

The objective of this homework is to gain an understanding of k nearest neighbors classifiers when used on datasets and the results that occur from manipulation of the dataset. For this assignment, I used Wine Data Set from the UCI Machine Learning Repository. The data are the results of a chemical analysis of wines grown in the same region but derived from different cultivars. The analysis determined the quantities of the 13 constituents found in each of the three types of wine. The attributes are:

1) Alcohol   
2) Malic acid   
3) Ash   
4) Alcalinity of ash   
5) Magnesium   
6) Total phenols   
7) Flavanoids   
8) Nonflavanoid phenols   
9) Proanthocyanins   
10)Color intensity   
11)Hue   
12)OD280/OD315 of diluted wines   
13)Proline

For initialization, some data manipulation was required in order to fulfill the parameters of the given knnclassify function. The first column in the data represents the corresponding cultivar the wine was from. To fulfill the group parameter requirement, I extracted the first column from the data and inputted it into the group matrix, with each index corresponding to the related attribute data in the training data. I then had to replace all of the commas in the data into spaces so that it could be correctly formatted into a matrix in matlab.

The default rate equation that I used for this homework was:

The resulting default rate for the data set is 39.8876%. To test the dataset with knnclassify, I first randomized the matrix rows. I then cut the data in half and used the first half as my training set, which was inputted in the training matrix parameter, and the second half as my test set, which was inputted in the sample matrix parameter. After running knnclassify function with this data, the resulting accuracy rate using the same equation was 39.3248%, which is slightly lower than the default rate.

Next, I cut the training data in half and ran the function with the same sample matrix. The resulting accuracy of this data was 49.44%. By having less data in the training matrix, there is less restriction on the overall classification of the sample data. This results in more sample data being classified on the looser bound of the most prevalent cultivar. More data in the training matrix results in a tighter bound and a more distinct classifier.

I then examined the result of removing features one by one using the original train and test sets. Removing the alcohol attribute still had the same result as the original. The same result occurred when I took out ash, total phenols, Nonflavanoid phenols, Proanthocyanins, Hue   
12)OD280/OD315 of diluted wines. Taking out the magnesium attribute increased the rate to 41.57%. The only attribute that lowered the accuracy when taken out was ash. When removed, the accuracy lowered to 38.2%.

The next modification I did was add a column to both sample and training matrices, with each entry of the column containing a randomly generated number. This was done by using the rand function and looping through each matrix to add each new index. After adding a column of random values, I recalculated the clustering. The accuracy did not change from the default rate with the new column. I then added another column and recalculated the clustering. Once again the accuracy remained the same. I then continued to add and test columns until the resulting clustering changed. The results stayed the same until I added an 11th random column to the matrices. With the 11th addition, the accuracy increased to 40.45%. The next time that the accuracy changed is on the 24th addition, where it decreased back to the 39.3248%.