COMP 307
Tyler Shamsuddoha
300428076

1.

When K = 1
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
the predicted class: class IrisNode was incorrect
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct

predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
the predicted class: class IrisNode was incorrect
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
the predicted class: class IrisNode was incorrect
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
the predicted class: class IrisNode was incorrect
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct
predicated class: class IrisNode is correct

95% is accurate
Number of matches is 71
Number of nonMatches is 4
K was set at 1

2. When K = 3

94% is accurate
Number of matches is 70
Number of nonMatches is 5
K was set at 3

The two performances from my algorithm is very similar for k=1 and k=3. This could be due to multiple things. I think that this is because the nearest neighbour around the value of the node was apart of a cluster of same values. Meaning that even if it takes the 3 nearest neighbours, it would be the same as taking 1 nearest neighbour as they are all of the same type. Hence why the performance of k = 1 and k = 3 are very similar in terms of accuracy.

For example if k was equal to the size of the training data, it would result in predicting the class to be the most common class in all the test data. When K = 1 however, it is only taking the nearest neighbour or closest Iris (in this case). This isn't as accurate as it is not taking into account clusters. The one Iris that it is using could potentially be an outlier which would mean an inaccurate prediction. That's why increasing K should generally increase accuracy, because it takes into account clustering.

3.Discuss the main advantages and disadvantages of k-Nearest Neighbour method.

Advantages:

- Easy to implement the algorithm
- K Nearest neighbour handles multi classification, this can be compared to DecisionTrees which does not do it as well.
- The accuracy increases if the training set is larger

Disadvantages:

- Can be costly because each set has to be compared to the entire training set. This means that the Big-O cost would be costly, being O(n*m) which is not good in terms of efficiency, especially when using larger datasets
- Would need to determine the optimal K Value based on the dataset.
- Could be harder to implement as we need to specify which distance to be used, as well as specify which attributes will be taken in account.

4. Assuming that you are asked to apply the k-fold cross validation method for the above problem with k=5, what would you do? State the major steps

- Parse the data in the dataset.
- Split the data, so that there are subsets.
- Run K Nearest neighbour 5 times, where one of the sets is the test set, whilst the remaining sets are the training sets. Eg. Test set: 2 Training Set:1,3,4,5
- Calculate the accuracy for each test set
- Get the average accuracy from all five different runs with each different test set.

5. In the above problem, assuming that the class labels are not available in the training set and the test set, and that there are three clusters, which method would you use to group the examples in the data set? State the major steps

- Parse the data set
- Set K to 3, due to clusters being apparent
- Create k cluters by linking each instance with the nearest mean, which is based on a measure of distance that you choose
- Replace old means with centroid of each k cluster (as new means)
- Repeat until there is no convergence