COMP307 Assignment 1 Joel Chu 300524008

Part1

Q1

```
Where k= 1
Accuracy:  94.3820224719101  ==  84 / 89
```

Q2

```
Where k= 1
Accuracy:  94.3820224719101  ==  84 / 89
```

```
Where k= 3
Accuracy:  95.50561797752809  ==  85 / 89
```

When k=1, it is using the singular closest point to determine it's classification. This could mean that if a point is closest to an unusually placed point, it will be classed as the unusually placed point.

When k=3, it takes into account the 3 nearest points to determine its classification. It will do this by getting the most frequent class out of the 3 nearest points and use this to determine the class of the given point. When there are 3 different classes, it will use the class that occurs the most in the training dataset.

From the given wine dataset, I used wine-training and wine-testing as the inputs. There was only a difference of 1 correct classification between k=1 and k=3 with k=3 coming out on top. Both results were relatively similar which suggests that the dataset has very well defined borders between each of the 3 classes. K=3 did slightly better because there must have been some test points which were close to two different classes where the k=3 could determine the correct class by finding the most frequent class out of the 3 closest points to the input point.

Q3

Advantages:

NO training period required – Since all the points are stored, the system doesn't need to undergo any form of training. The output is determined by analysing existing data. Due to no training period, it saves time making this algorithm very quick to use.

Easy to add new data – When there is new data available, it can easily be added. It can be added as input to the algorithm and can be used from there.

Easy to implement – The only form of computation in the algorithm is calculating the Euclidian distance between two points.

Disadvantages:

Does not work well with large datasets – A distance needs to be calculated for each point. When the dataset gets large, there would have to be a lot for calculations just to determine the class for one data point. Potentially taking a long time.

Easily impacted by noise and outliers – Since the algorithm finds the closest point(s) to the input point, if there are outliers or noisy data present, there will be data points where they realistically shouldn't be. This means that when a data point is close to noisy or outlying data, it will use these to determine it's class even when it may not be correct. This disadvantage may be seen more often

when the k values are smaller as there are less data points being used to determine the class of the input value.

Does not work well in large dimensions

When the dimension gets large, this means that calculating the distance between each point will cost more. There will be more parameters in the distance calculation since it needs to account for every single attribute making the calculation more costly. Overall, it will mean that finding the class of a particular input point may take longer.

Q4

Since k=5,

Split data set into 5 groups

Split the training set into 5 sets where each time, 1 of the 5 sets will be the test set and the other 4 will be used for training.

We want to compare the algorithm where k=3 and where k=1

Run 5 times for each algorithm k=1 and k=3

The KNN search will be performed 5 times where k=3 and k=1. Each time, the training and test set will be rotated so it will be different each time. The accuracy for each run will be calculated and stored.

First run, the first 5th will be used to testing, the remaining 4/5 will be used for the dataset("training"). The accuracy will be stored. In second run, the second 5^{th} will be used for testing, the remaining 4/5 will be used for the dataset("training"). This is repeated 5 times in total so that each fifth of the full dataset are used for testing once.

Get average and find the best algorithm

At the end of the testing, there should be 5 different accuracies, one set of 5 for k=1 and one set of 5 for k=3. The average accuracy for k=1 and k=3 will be calculated. The better algorithm will be the one with the higher average.

Q5

If there were no classes available, we would have to go through and find these classes ourselves. A method of doing this would be to use **k means clustering**.

1) We would start off by picking a number of clusters that we want to find. Let the number be n.
2) We would then go and find n random points in the dataset. Every data value in the set will then be associated with it's mean. This being it's closest point out of the n random initial points. This determines the initial n clusters.
3) At each iteration, in each of the clusters, a new mean will be found by calculating the point in the middle of all datapoints in that cluster.

4) The previous step will be repeated until all n clusters converge. This meaning that there isn't any change in the means anymore or there is a satisfactory amount of change of the clusters midpoint.
5) There will now be n clusters which will be the groups in the dataset

Part 2

Q1

```
PS D:\2022\comp307\Assignment1\ass1_data\part2> py DecisionTree.py hepatitis-training hepatitis-test
ASCITES  = TRUE:
        SPIDERS  = TRUE:
                VARICES  = TRUE:
                        STEROID  = TRUE:
                                live , prob =  1.0
                        STEROID  = FALSE:
                                SPLEENPALPABLE  = TRUE:
                                        FIRMLIVER  = TRUE:
                                                live , prob =  1.0
                                        FIRMLIVER  = FALSE:
                                                BIGLIVER  = TRUE:
                                                        SGOT  = TRUE:
                                                                live , prob =  1.0
                                                        SGOT  = FALSE:
                                                                FEMALE  = TRUE:
                                                                        live , prob =  1.0
                                                                FEMALE  = FALSE:
                                                                        ANOREXIA  = TRUE:
                                                                                die , prob =  1.0
                                                                        ANOREXIA  = FALSE:
                                                                                live , prob =  1.0
                                                BIGLIVER  = FALSE:
                                                        live , prob =  1.0
                                SPLEENPALPABLE  = FALSE:
                                        HISTOLOGY  = TRUE:
                                                die , prob =  1.0
                                        HISTOLOGY  = FALSE:
                                                live , prob =  1.0
                VARICES  = FALSE:
                        die , prob =  1.0
        SPIDERS  = FALSE:
                BILIRUBIN  = TRUE:
                        FATIGUE  = TRUE:
                                AGE  = TRUE:
                                        live , prob =  1.0
                                AGE  = FALSE:
                                        die , prob =  1.0
                        FATIGUE  = FALSE:
                                ANTIVIRALS  = TRUE:
                                        MALAISE  = TRUE:
                                                live , prob =  0.75
                                        MALAISE  = FALSE:
                                                live , prob =  0.7
                                ANTIVIRALS  = FALSE:
                                        live , prob =  1.0
                BILIRUBIN  = FALSE:
                        live , prob =  0.8888888888888888
ASCITES  = FALSE:
        die , prob =  0.7333333333333333
Baseline: 0.8
Test accuracy: 0.76
```

The baseline classifier has an accuracy of 0.8. The decision tree had an accuracy of 0.76. The

difference was 0.04. The Baseline accuracy was more accurate than the decision tree when using hepatitis-training and hepatitis-test.

Q2

```
0)0.7666666666666667
1)0.8
2)0.6666666666666666
3)0.7333333333333333
4)0.8
5)0.7
6)0.8
7)0.8333333333333334
8)0.6333333333333333
9)0.7666666666666667
Ten fold average = 0.7499999999999999
```

The 10 fold average fluctuates each time it is run. This is due to the random aspect. Average can change by about +-0.01.

Working:

```
def doTenFold():
    tenFoldList = []
    for i in range(10):
        attributes.clear()
        trainingInstances.clear()
        testInstances.clear()
        decisions.clear()
        loadSets("hepatitis-training-run-"+str(i), trainingInstances)
        loadSets("hepatitis-test-run-"+str(i), testInstances)
        root = buildTree(trainingInstances, attributes)
        tenFoldList.append(testTree(testInstances, root))
        print(str(i)+")"+str(testTree(testInstances, root)))
    print("Ten fold average = "+str(average(tenFoldList)))
```

Q3

a) Pruning a tree can be done when a tree reaches a certain depth or if a particular section doesn't help the tree in classification. To do this, a node's children (whole branch) could be removed from the tree. The node that the pruning occurred will become a leaf node.

b) This will reduce accuracy because a complete tree will be the most accurate tree possible for the training set. When pruning occurs, branches will be removed and the tree becomes more generalised making it less specific to the training data.

c) Pruning might improve accuracy on the test set since the tree will be more general. Pruning helps prevent overfitting the tree less specific towards the training set. This in turn can potentially make it improve the accuracy due to the tree now being more generic.

d) The impurity measurement from the lecture measures
When have more than 2 classes, the impurity the probability will get smaller and smaller. Since we are multiplying probabilities in the impurity calculations, if the probabilities are

getting so small that they are essentially 0, it means that we can no longer distinguish the difference between the probabilities.
When there is a node in the calculation where there is none of one class, it automatically makes the impurity 0. This rules out the probabilities for the other classes since the 0 will make the impurity value 0. So having more than 2 classes increases the chance of events like this happening where the impurity is 0.

Part 3

Q1

```
Using weight:  -40  And Bias:  0  This was the highest accuracy that I could
find
it has been  407 Iterations. The results are above
Accuracy =  315 / 351     89.74358974358975
```

I brute forced a range of learning rate from -10 ,10 a range of biases from -20 to 20 and a range of weights from -100 to 100. All results were very similar. The accuracy would start off very low, rapidly increasing in the first 25 iterations. The accuracy would start to plateau just after the 25th iteration. It would level out around 290 correct answers, constantly alternating accuracies between 280 and 300. At some points, this would go up to 315 which is the highest number of correct calculations that I've seen. I found that using a smaller learning rate meant that there isn't as much fluctuation when the accuracy plateaus. The best accuracy that I could find was weight = -40, bias = 0, Learningrate = 1. With an accuracy of 315/315 = 89.74% after 407 iterations. After this max rate, it would then drop back down to the 290 correct answers mark and stay around there for the rest of the run.

It appeared that the dataset wasn't linearly separable, so the perceptron never fully converged.

Q2

Using the training data on itself isn't a good measure of effectiveness because it means that if the Perceptron starts to overfit, this will favour the training set evaluation. When the perceptron overfits, it will become too specific and the error will might be greater if another test dataset was used. This will also make the perceptron have a very high level of accuracy when making guesses on the training data but using another set of data may be different since the perceptron has never seen the test data before. Using a different test set than the training set to measure effectiveness will also test if the perceptron is generalised enough but also accurate enough.

```
PS D:\2022\comp307\Assignment1\SourceCode\part3> py PerceptronSplitSets.py ionosphere.data
Using weight:  -28  And Bias:  -10  This was the highest accuracy that I could find
it has been  394 Iterations. The results are above
Accuracy =  187 / 211     88.62559241706161

Using test data which has length of:  140
Accuracy =  130 / 140     92.85714285714286
```

After splitting the training and data set into two groups, I had 60% of the data as training and the other 40% was test data. I went and found the best bias and weight for this specific dataset which was -28 starting weight, bias=-10 and I left the learning rate at 1.