COMP 307 Assignment 1

Table of Contents

[Part 1: K Nearest Neighbour (KNN) 1](#_Toc511199085)

[Results When K = 1 1](#_Toc511199086)

[Part 2: Decision Tree 1](#_Toc511199087)

[Part 3: Perceptron 1](#_Toc511199088)

# Part 1: K Nearest Neighbour (KNN)

## Results When k = 1

The classification accuracy is 0.92 when k = 1. The following is the class labels of each instance predicted by the algorithm in the “Predicted” column.

|  |  |
| --- | --- |
| ***Actual*** | ***Predicted*** |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-setosa | Iris-setosa |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-virginica |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-versicolor | Iris-versicolor |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-versicolor |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-versicolor |
| Iris-virginica | Iris-versicolor |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-versicolor |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-virginica |
| Iris-virginica | Iris-versicolor |

## Accuracy When k = 3

The classification accuracy is 0.94667 when k is set to 3. It is an almost 3% accuracy improvement.

Performance wise, K1 took 45118 microseconds, while as K3 took 47788 microseconds. K3 took slightly longer to run. However, both are within a second. Therefore, it is probably not human recognisable. The difference will become more significant when there is training or/and test data. Of course, the execution time also depends on the software and hardware platform on which you ran the programme.

In theory, KNN will require the same amount of storage regardless the value of K.

## Advantages

* Simple to implement.
* This algorithm can deal with no linear separable data sets.

## Disadvantage

* It cannot deal with linear data, i.e. continuous numbers. Although, you may covert numeric data to categorical data by using ranges.
* The algorithm is expensive. There is no training. For each test instance, you will have to calculate its distance to every training instances. Therefore, this algorithm will struggle with performance when comes to performance.
* It cannot deal with missing data gracefully. You probably can mask missing data to overcome this issue but you should be very careful about not skewing the results.

## K-fold Cross Validation When K=5

K-fold cross validation solves a problem where there is only a single data file. In this scenario, K fold validation is a method of splitting data to a training data set and test data sets. K defines how many folds you will divide data into. Typically, you will use the first fold as the training data, and the remaining folds as test data sets. Each test data set will produce a result. The final result is the average of those sub-results.

The optimal value of K depends on if you want to minimise the training time, or if you will prefer to optimise the training accuracy. Of course, you will want take the data set itself into consideration.

*Steps:*

* Divide the data evenly into 5 data sets.
* Train your data model against the first/training data set.
* Test the trained classifier against each of the remaining 4 test data sets, and record the accuracy of each test.
* Once the classifier has completed all test runs, average the test results.

## When The Class Labels Are Not Available

When the class labels are not available, I will use the K Means Clustering method to group the examples.

*Steps:*

* Step1: Randomly choose 3 centroids.
* Step2: Group each instance into a category/centroids.
  + You can use Euclidian Distance to decide on which category.
* Step3: Calculated 3 new centroids. One for each group.
* Step4: Repeat the step 2 and the step 3 until convergence has been reached (the centroids don’t change anymore).

# Part 2: Decision Tree

# Part 3: Perceptron