Iris setosa

Iris virginica

Iris versicolor

Iris Mythica

COMP307

ASSIGNMENT 1

Zane Rawson | 300367145 | 05/04/2019

Part 1- K-Nearest-Neighbour

1. K=1:

```
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-setosa || Guessed Class: Iris-setosa
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-versicolor
```

```
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Incorrect -> Predicted class: Iris-versicolor || Guessed Class:
Iris-virginica
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Incorrect -> Predicted class: Iris-versicolor || Guessed Class:
Iris-virginica
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-versicolor || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
```

```
Incorrect -> Predicted class: Iris-virginica || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Incorrect -> Predicted class: Iris-virginica || Guessed Class: Iris-
versicolor
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica
Correct -> Predicated class: Iris-virginica || Guessed Class: Iris-
virginica

Number Passed: 71/75
Number Failed: 4/75
Accuracy: 0.947%

Most Common Class: Iris-setosa
```

2. **Report the classification accuracy on the test set of the k-nearest neighbour method where k=3, and compare and comment on the performance of the two classifiers (k=1 and k=3)**

There is a small difference in the performance from K=1 and K=3 with K=3 being slightly less accurate with the number failed increasing by one. This is slightly surprising as while normally when you increase the value of K you expect to see an increase in performance up to a certain value of K in this case with there being three clear clusters, I would have expected the results to be identical. This additional incorrect prediction would be from the overlap that occurs between the Iris-virginica and Iris versicolor as there is some overlap between them. We can expect to see 100% for most values of K with the Iris Setosa as they do not overlap with any of the other flower types.

3.

Benefits	Drawbacks
- Quite effective if the training set is large - Simple to implement - It can handle multiclass learning algorithms - It is robust to noisy training data	- Need to determine the number of nearest neighbours - Distance based learning is not clear on which type of distance to use and which attribute will yield the best results - Computation cost is quite high

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.**
    a. Split the data up into 5 folds
    b. For each fold we treat one as the test set and the other K-1 fold as the training data
    c. Repeat for each fold so that each one is used as the test data once
    d. Average the results to produce a single estimation on how the system will perform on real data.
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:**
   In this example it would make sense to use K-means clustering if there aren't any labels in either the training or test data as it would be able to split the irises into three clusters using K = 3.
   Steps:
    - Make K centroids
    - Place them randomly within the test data
    - For each one loop through them and calculate the mean for each one
    - Set the mean to each centroid and make it the centre of each cluster
    - Keep going until there is no further change

PART 2 – DECISION TREES

1. Output:

```
2 categories
16 attributes
Read 100 instances
 FEMALE = True:
    Class live, probability = 1.00
 FEMALE = False:
    SGOT = True:
       BILIRUBIN = True:
          FATIGUE = True:
```

```
            Class live, probability = 1.00
         FATIGUE = False:
            ANTIVIRALS = True:
               HISTOLOGY = True:
                  BIGLIVER = True:
                     AGE = True:
                        ANOREXIA = True:
                           Class die, probability = 1.00
                        ANOREXIA = False:
                           Class live, probability = 1.00
                     AGE = False:
                        SPLEENPALPABLE = True:
                           ASCITES = True:
                              Class live, probability = 1.00
                           ASCITES = False:
                              STEROID = True:
                                 Class die, probability = 1.00
                              STEROID = False:
                                 FIRMLIVER = True:
                                    Class live, probability = 1.00
                                 FIRMLIVER = False:
                                    Class die, probability = 1.00
                        SPLEENPALPABLE = False:
                           Class die, probability = 1.00
                  BIGLIVER = False:
                     Class live, probability = 1.00
               HISTOLOGY = False:
                  SPIDERS = True:
                     Class live, probability = 1.00
                  SPIDERS = False:
                     Class die, probability = 1.00
            ANTIVIRALS = False:
               Class live, probability = 1.00
      BILIRUBIN = False:
         Class live, probability = 1.00
   SGOT = False:
      MALAISE = True:
         VARICES = True:
            Class live, probability = 0.94
         VARICES = False:
            Class die, probability = 1.00
      MALAISE = False:
         Class live, probability = 0.54
Read 37 instances

Number Correct: 34 /37
Accuracy: 91.89%
```

2. First run – 91.89%
   Second Run – 30/37 - 81.08%
   Third Run – 31/37 – 83.78%
   Fourth Run – 28/37 – 75.68%
   Fifth Run - 33/37 – 89.19%
   Sixth Run – 29/37 – 78.38%
   Seventh Run - 29/37 – 78.38%
   Eighth run - 30/37 - 81.08%
   Ninth run - 31/37 – 83.78%

Tenth Run - 30/37 - 81.08%

3. **"Pruning" (removing) some of leaves of the decision tree will always make the decision tree less accurate on the training set. Explain (a) How you could prune leaves from the decision tree; (b) Why it would reduce accuracy on the training set, and (c)Why it might improve accuracy on the test set**
   a. Pruning: We can improve the accuracy of the test data by pruning unnecessary leaves to reduce the chance of overfitting. To do this we can run the training data as we normally would and measure its performance. From here we can pick a node and its children and measure how it will impact performance. Then we can use remove the nodes that give us the best results and repeat this until any further pruning would have a negative impact on the results
   b. When we produce a decision tree it keeps making leaves until all the nodes are pure. This gives us 100% accuracy on the training set but will often make it less accurate on the test data due to how much detail it goes into. By pruning we make it less accurate on the training data but will in turn make it more accurate on the test data
   c. When running on a dataset that hasn't been seen yet we can see an improvement as the issue of overfitting will be reduced depending on the number of leaves we have pruned.
4. When looking at the purity of two classes it should only be zero if all of the instances belong to just one class however with 3 or more classes it is possible to have an impurity of zero with instances belonging to more than one class. In a situation like this it would make the impurity measure a poor choice.

PART 3 – PERCEPTRON

1. Output:

```
All images passed
Number of repetitions: 100

Ending learning Rate: 0.0984



Pixel 0: Column: 7, Row: 8
Pixel 1: Column: 4, Row: 9
Pixel 2: Column: 0, Row: 0
Pixel 3: Column: 6, Row: 9
Weight: -0.378


Pixel 0: Column: 3, Row: 6
Pixel 1: Column: 7, Row: 1
Pixel 2: Column: 3, Row: 0
Pixel 3: Column: 2, Row: 4
Weight: 0.6210


Pixel 0: Column: 6, Row: 4
```

```
Pixel 1: Column: 1, Row: 3
Pixel 2: Column: 0, Row: 5
Pixel 3: Column: 1, Row: 9
Weight: 0.6234


Pixel 0: Column: 2, Row: 2
Pixel 1: Column: 4, Row: 4
Pixel 2: Column: 4, Row: 7
Pixel 3: Column: 6, Row: 5
Weight: -1.376


Pixel 0: Column: 6, Row: 7
Pixel 1: Column: 5, Row: 3
Pixel 2: Column: 4, Row: 6
Pixel 3: Column: 1, Row: 8
Weight: -1.376


Pixel 0: Column: 2, Row: 5
Pixel 1: Column: 8, Row: 4
Pixel 2: Column: 7, Row: 5
Pixel 3: Column: 7, Row: 4
Weight: -0.376


Pixel 0: Column: 9, Row: 0
Pixel 1: Column: 1, Row: 6
Pixel 2: Column: 3, Row: 8
Pixel 3: Column: 9, Row: 0
Weight: -0.376


Pixel 0: Column: 3, Row: 9
Pixel 1: Column: 4, Row: 7
Pixel 2: Column: 7, Row: 8
Pixel 3: Column: 5, Row: 5
Weight: -0.376


Pixel 0: Column: 8, Row: 3
Pixel 1: Column: 5, Row: 5
Pixel 2: Column: 7, Row: 7
Pixel 3: Column: 0, Row: 0
Weight: -0.376


Pixel 0: Column: 3, Row: 2
Pixel 1: Column: 8, Row: 7
Pixel 2: Column: 0, Row: 7
Pixel 3: Column: 6, Row: 4
Weight: -0.376


Pixel 0: Column: 0, Row: 7
Pixel 1: Column: 0, Row: 8
Pixel 2: Column: 6, Row: 0
```

```
Pixel 3: Column: 0, Row: 2
Weight: -1.377


Pixel 0: Column: 6, Row: 5
Pixel 1: Column: 3, Row: 0
Pixel 2: Column: 9, Row: 1
Pixel 3: Column: 4, Row: 8
Weight: 0.6226


Pixel 0: Column: 9, Row: 8
Pixel 1: Column: 4, Row: 8
Pixel 2: Column: 0, Row: 0
Pixel 3: Column: 1, Row: 1
Weight: -0.377


Pixel 0: Column: 5, Row: 6
Pixel 1: Column: 7, Row: 9
Pixel 2: Column: 3, Row: 0
Pixel 3: Column: 4, Row: 6
Weight: -1.377


Pixel 0: Column: 2, Row: 1
Pixel 1: Column: 0, Row: 1
Pixel 2: Column: 7, Row: 3
Pixel 3: Column: 9, Row: 4
Weight: -0.377


Pixel 0: Column: 8, Row: 0
Pixel 1: Column: 3, Row: 2
Pixel 2: Column: 0, Row: 3
Pixel 3: Column: 4, Row: 8
Weight: -1.377


Pixel 0: Column: 0, Row: 4
Pixel 1: Column: 6, Row: 1
Pixel 2: Column: 3, Row: 1
Pixel 3: Column: 9, Row: 3
Weight: -0.377


Pixel 0: Column: 6, Row: 3
Pixel 1: Column: 7, Row: 0
Pixel 2: Column: 6, Row: 1
Pixel 3: Column: 3, Row: 9
Weight: -2.377


Pixel 0: Column: 7, Row: 3
Pixel 1: Column: 8, Row: 3
Pixel 2: Column: 9, Row: 1
Pixel 3: Column: 3, Row: 8
Weight: -1.380
```

```
Pixel 0: Column: 3, Row: 4
Pixel 1: Column: 1, Row: 4
Pixel 2: Column: 2, Row: 1
Pixel 3: Column: 6, Row: 3
Weight: 0.6190


Pixel 0: Column: 5, Row: 6
Pixel 1: Column: 4, Row: 3
Pixel 2: Column: 3, Row: 0
Pixel 3: Column: 3, Row: 7
Weight: 0.6193


Pixel 0: Column: 1, Row: 5
Pixel 1: Column: 7, Row: 4
Pixel 2: Column: 6, Row: 2
Pixel 3: Column: 7, Row: 2
Weight: -0.380


Pixel 0: Column: 8, Row: 0
Pixel 1: Column: 7, Row: 6
Pixel 2: Column: 2, Row: 5
Pixel 3: Column: 1, Row: 0
Weight: 0.6188


Pixel 0: Column: 4, Row: 8
Pixel 1: Column: 0, Row: 7
Pixel 2: Column: 5, Row: 4
Pixel 3: Column: 7, Row: 4
Weight: -2.381


Pixel 0: Column: 7, Row: 3
Pixel 1: Column: 3, Row: 6
Pixel 2: Column: 5, Row: 3
Pixel 3: Column: 2, Row: 9
Weight: -0.381


Pixel 0: Column: 2, Row: 1
Pixel 1: Column: 6, Row: 7
Pixel 2: Column: 9, Row: 3
Pixel 3: Column: 8, Row: 5
Weight: -2.381


Pixel 0: Column: 0, Row: 6
Pixel 1: Column: 9, Row: 9
Pixel 2: Column: 3, Row: 6
Pixel 3: Column: 2, Row: 1
Weight: -0.381
```

```
Pixel 0: Column: 5, Row: 5
Pixel 1: Column: 0, Row: 0
Pixel 2: Column: 6, Row: 6
Pixel 3: Column: 6, Row: 6
Weight: -1.381


Pixel 0: Column: 0, Row: 7
Pixel 1: Column: 3, Row: 8
Pixel 2: Column: 8, Row: 4
Pixel 3: Column: 1, Row: 9
Weight: 0.6186


Pixel 0: Column: 4, Row: 8
Pixel 1: Column: 6, Row: 7
Pixel 2: Column: 0, Row: 6
Pixel 3: Column: 6, Row: 5
Weight: -1.381


Pixel 0: Column: 1, Row: 1
Pixel 1: Column: 8, Row: 1
Pixel 2: Column: 6, Row: 9
Pixel 3: Column: 1, Row: 3
Weight: -0.381


Pixel 0: Column: 7, Row: 1
Pixel 1: Column: 1, Row: 2
Pixel 2: Column: 9, Row: 9
Pixel 3: Column: 4, Row: 8
Weight: -0.381


Pixel 0: Column: 1, Row: 4
Pixel 1: Column: 2, Row: 1
Pixel 2: Column: 9, Row: 7
Pixel 3: Column: 1, Row: 2
Weight: 0.6182


Pixel 0: Column: 7, Row: 2
Pixel 1: Column: 5, Row: 0
Pixel 2: Column: 2, Row: 7
Pixel 3: Column: 5, Row: 7
Weight: -0.381


Pixel 0: Column: 0, Row: 2
Pixel 1: Column: 9, Row: 5
Pixel 2: Column: 9, Row: 3
Pixel 3: Column: 6, Row: 2
Weight: -0.379


Pixel 0: Column: 6, Row: 0
Pixel 1: Column: 2, Row: 4
```

```
Pixel 2: Column: 3, Row: 4
Pixel 3: Column: 0, Row: 6
Weight: -0.379


Pixel 0: Column: 8, Row: 2
Pixel 1: Column: 4, Row: 2
Pixel 2: Column: 3, Row: 1
Pixel 3: Column: 5, Row: 1
Weight: -0.379


Pixel 0: Column: 4, Row: 1
Pixel 1: Column: 7, Row: 3
Pixel 2: Column: 6, Row: 3
Pixel 3: Column: 0, Row: 7
Weight: 0.6206


Pixel 0: Column: 1, Row: 6
Pixel 1: Column: 9, Row: 7
Pixel 2: Column: 2, Row: 6
Pixel 3: Column: 5, Row: 3
Weight: -0.379


Pixel 0: Column: 7, Row: 5
Pixel 1: Column: 2, Row: 6
Pixel 2: Column: 6, Row: 6
Pixel 3: Column: 0, Row: 8
Weight: -1.379


Pixel 0: Column: 2, Row: 9
Pixel 1: Column: 3, Row: 5
Pixel 2: Column: 6, Row: 4
Pixel 3: Column: 5, Row: 1
Weight: -0.379


Pixel 0: Column: 5, Row: 7
Pixel 1: Column: 6, Row: 6
Pixel 2: Column: 9, Row: 4
Pixel 3: Column: 9, Row: 7
Weight: 0.6203


Pixel 0: Column: 3, Row: 2
Pixel 1: Column: 9, Row: 0
Pixel 2: Column: 3, Row: 7
Pixel 3: Column: 3, Row: 5
Weight: -0.380


Pixel 0: Column: 8, Row: 1
Pixel 1: Column: 1, Row: 9
Pixel 2: Column: 6, Row: 9
Pixel 3: Column: 9, Row: 1
```

```
Weight: 0.6197


Pixel 0: Column: 3, Row: 5
Pixel 1: Column: 5, Row: 8
Pixel 2: Column: 9, Row: 5
Pixel 3: Column: 4, Row: 4
Weight: -1.379


Pixel 0: Column: 9, Row: 4
Pixel 1: Column: 6, Row: 9
Pixel 2: Column: 2, Row: 7
Pixel 3: Column: 0, Row: 9
Weight: -0.380


Pixel 0: Column: 6, Row: 9
Pixel 1: Column: 8, Row: 1
Pixel 2: Column: 6, Row: 0
Pixel 3: Column: 4, Row: 5
Weight: -2.380


Pixel 0: Column: 0, Row: 8
Pixel 1: Column: 1, Row: 2
Pixel 2: Column: 9, Row: 0
Pixel 3: Column: 8, Row: 1
Weight: -1.380


Pixel 0: Column: 4, Row: 0
Pixel 1: Column: 4, Row: 0
Pixel 2: Column: 0, Row: 9
Pixel 3: Column: 3, Row: 6
Weight: -1.380


Pixel 0: Column: 0, Row: 8
Pixel 1: Column: 5, Row: 1
Pixel 2: Column: 3, Row: 8
Pixel 3: Column: 8, Row: 0
Weight: 0.6196
```

2. Measuring the performance based on the training data because it has already learnt it. The best way to fully test the performance is to use test data that the system hasn't seen before