COMP 307 Assignment 1

Table of Contents

Student Name: Yu Shen

Student ID: 300210617

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-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-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 the training or/and the test datasets are much larger. 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 can convert 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.
- 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 datasets. K defines how many folds you will divide the data. 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 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 to take the size of the dataset into consideration too.

Steps:

- Divide the data evenly into five datasets.
- 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 (result) 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 three centroids.
- Step2: Group each instance into a category/centroids.
    - You can use Euclidian Distance to decide on which category.
- Step3: Calculated three 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

Decision Tree Output

The decision tree classification accuracy is 0.8148, while as the baseline classification accuracy is 0.8519. Yes, the predicted accuracy of the decision tree is worse than the baseline classification.

However, this may not be a bad thing. In the case of huge class imbalance, the baseline classification will be more accurate, but the mode will be useless in the problem domain with different datasets.

The following is the decision tree classifier structure:

```
FEMALE = false, instances count = 97
  FATIGUE = false, instances count = 61
    ASCITES = false, instances count = 12
      BIGLIVER = false, instances count = 1
        Class live, prob = 1.0
      BIGLIVER = true, instances count = 11
        ANTIVIRALS = true, instances count = 11
          BILIRUBIN = false, instances count = 1
            Class die, prob = 1.0
          BILIRUBIN = true, instances count = 10
            AGE = false, instances count = 8
              HISTOLOGY = false, instances count = 1
                Class live, prob = 1.0
              HISTOLOGY = true, instances count = 7
                MALAISE = false, instances count = 6
                  SPLEENPALPABLE = true, instances count = 6
                    SPIDERS = false, instances count = 5
                      SGOT = false, instances count = 1
                        Class die, prob = 1.0
                      SGOT = true, instances count = 4
                        ANOREXIA = false, instances count = 3
                          Class die, prob = 0.6666666666666666
                        ANOREXIA = true, instances count = 1
                          Class die, prob = 1.0
                    SPIDERS = true, instances count = 1
                      Class die, prob = 1.0
                MALAISE = true, instances count = 1
                  Class die, prob = 1.0
```

AGE = true, instances count = 2

Class die, prob = 1.0

ASCITES = true, instances count = 49

SPIDERS = false, instances count = 19

SPLEENPALPABLE = false, instances count = 7

ANTIVIRALS = true, instances count = 7

BILIRUBIN = true, instances count = 7

BIGLIVER = false, instances count = 1

Class die, prob = 1.0

BIGLIVER = true, instances count = 6

AGE = false, instances count = 3

Class die, prob = 1.0

AGE = true, instances count = 3

Class live, prob = 0.6666666666666666

SPLEENPALPABLE = true, instances count = 12

AGE = false, instances count = 11

VARICES = false, instances count = 1

Class live, prob = 1.0

VARICES = true, instances count = 10

ANOREXIA = false, instances count = 3

Class live, prob = 1.0

ANOREXIA = true, instances count = 7

MALAISE = false, instances count = 3

STEROID = false, instances count = 3

ANTIVIRALS = false, instances count = 1

Class die, prob = 1.0

ANTIVIRALS = true, instances count = 2

Class live, prob = 0.5

MALAISE = true, instances count = 4

ANTIVIRALS = false, instances count = 1

Class live, prob = 1.0

ANTIVIRALS = true, instances count = 3

Class live, prob = 0.6666666666666666

AGE = true, instances count = 1

Class die, prob = 1.0

SPIDERS = true, instances count = 30

VARICES = false, instances count = 1

Class die, prob = 1.0

VARICES = true, instances count = 29

SPLEENPALPABLE = false, instances count = 2

Class live, prob = 1.0

SPLEENPALPABLE = true, instances count = 27

BIGLIVER = false, instances count = 4

Class live, prob = 1.0

BIGLIVER = true, instances count = 23

ANOREXIA = false, instances count = 4

Class live, prob = 1.0

ANOREXIA = true, instances count = 19

SGOT = false, instances count = 16

HISTOLOGY = false, instances count = 13

AGE = false, instances count = 10

MALAISE = false, instances count = 5

Class live, prob = 0.8

MALAISE = true, instances count = 5

Class live, prob = 1.0

AGE = true, instances count = 3

Class live, prob = 1.0

HISTOLOGY = true, instances count = 3

Class live, prob = 1.0

SGOT = true, instances count = 3

Class live, prob = 1.0

FATIGUE = true, instances count = 36

MALAISE = true, instances count = 36

ANOREXIA = true, instances count = 36

ASCITES = true, instances count = 36

SPLEENPALPABLE = false, instances count = 4

Class live, prob = 1.0

SPLEENPALPABLE = true, instances count = 32

BIGLIVER = false, instances count = 4

Class live, prob = 1.0

BIGLIVER = true, instances count = 28

ANTIVIRALS = false, instances count = 2

Class live, prob = 1.0

ANTIVIRALS = true, instances count = 26

SGOT = false, instances count = 22

VARICES = false, instances count = 1

Class die, prob = 1.0

VARICES = true, instances count = 21

SPIDERS = false, instances count = 1

Class live, prob = 1.0

SPIDERS = true, instances count = 20

Class live, prob = 0.95

SGOT = true, instances count = 4

Class live, prob = 1.0

FEMALE = true, instances count = 13

Class live, prob = 1.0

The Results of 10 Test Runs

The below is the results of the 10:

Baseline classifier accuracy: 0.918918918918919

Accuracy = 0.8648648648648649

Baseline classifier accuracy: 0.8918918918918919

Accuracy = 0.8378378378378378

Baseline classifier accuracy: 0.8648648648648649

Accuracy = 0.8378378378378378

Baseline classifier accuracy: 0.7837837837837838

Accuracy = 0.7297297297297297

Baseline classifier accuracy: 0.8918918918918919

Accuracy = 0.8108108108108109

Baseline classifier accuracy: 0.7837837837837838

Accuracy = 0.7837837837837838

Baseline classifier accuracy: 0.8918918918918919

Accuracy = 0.8648648648648649

Baseline classifier accuracy: 0.8648648648648649

Accuracy = 0.7567567567567568

Baseline classifier accuracy: 0.8378378378378378

Accuracy = 0.8108108108108109

Baseline classifier accuracy: 0.8108108108108109

Accuracy = 0.8648648648648649

Average accuracy = 0.8162162162162161

Pruning

Method 1

- Step 1: Remove a subtree in a reverse tree building manner.
- Step 2: Check if there is any improvement in accuracy.
- Step 3: If there is an improvement in accuracy, remove the subtree and replace the common node with a leaf node. The leaf node should be classified as the majority class of its instances.
- Repeat step 1, 2 and 3 until there is no improvement in accuracy.

Method 2

- Step 1: Decide on the number of steps m.
- Step 2: Start from step = 1.
- Step 3: Decide on which sub-tree to remove.
    - Remove each deepest sub-tree
    - Now you will have a new Tree Ti
    - Calculate the error rate. Error = Ti – T(i-1)
    - Choose the subtree with the smallest error rate to remove
- Step 4: Go back to Step 3 until reaches step = m.

When Three or More Classes

The impurity measure is not a good measure if there are three or more classes that the decision tree must distinguish. The reason for the above statement is because three or more classes will cause false positives. For example, if we have three classes in the dataset. In node A, we found probability of the three classes are P(A) = 50%, P(B) = 50%, and P(C) = 0%. So the Gini impurity is 3*P(A) *P(B) *P(C) = 0.5 * 0.5 * 0 = 0. The result zero reports the node is pure, but it is not true because it still contains both class A and B.

Part 3: Perceptron

The performance of my programme was bit random. I guess that was because the features generated were random too. Although, the error count was never below 25 when trained on 100 images. The best run produced a correct set of weights at training cycle 178 when I set the learning rate to 0.2. I have screenshot the results below because I have no idea when it will happen again.

On an average run, the error count will fluctuate between 14 to 23 (total image count = 100).

```
Training cycle # = 168, Classification error count = 2
Training cycle # = 169, Classification error count = 3
Training cycle # = 170, Classification error count = 4
Training cycle # = 171, Classification error count = 3
Training cycle # = 172, Classification error count = 2
Training cycle # = 173, Classification error count = 2
Training cycle # = 174, Classification error count = 2
Training cycle # = 175, Classification error count = 2
Training cycle # = 176, Classification error count = 2
Training cycle # = 177, Classification error count = 1
weights:
-1.6075376101596421
-1.0797345113947356
2.500690894786508
-4.638933427709906
-0.041760115369192274
-2.821689564101749
-4.512610500075433
0.2997669667748669
-0.7977123450196515
-1.4394446872633482
3.073968837969532
-5.003732742365585
0.25436993224968263
-5.288320261716903
1.8728053591539686
5.393132661238779
-7.526190097489645
0.42735449236472584
1.8807780612359195
0.83965318552782
5.3444095316425715
3.832945016116717 3
-0.5544570411489187
-2.173605439977456
3.580708110295884
-3.2044390605852042
1.7740675107977275
-1.613176880451582
7.288053845476602
1.9750552848028262
-1.477094983090091
4.087890944703398
0.6705845360384192
-5.578716862513377
3.3022331284291035
-5.519516419607598
6.773055638898055
0.7651150120257271
-1.8695152567629463
-4.077193309961921
6.792695573791231
-0.18088622697899337
0.4039582588044527 4
-2.3463362485212085
-4.9386207724126585
-1.6247852127631761
0.7236381758179442
4.7440698530883365
5.145826783340902
-1.4130162783018039
-5.167704741884284
```

Evaluating the Perceptron's Performance on Training Data

Evaluating the Perceptron's Performance on training data is not a good measure of its effectiveness. That is because the measure will not uncover any overfitting issues. The trained perceptron will know the exact results in the training data. Therefore its predictive power will be useless here.

To avoid that, you can either find more test data or using the cross-fold validation technique to split data into training and testing datasets. For example, by using 10-fold validation, you can use the one dataset for training and the remaining nine for fitting tests. Of course, the original single dataset should be large enough for this technique to be successful.