CS3DM practice questions - model answers

READ THIS FIRST

This document provides model answers for the two practice questions. The code bits indicate the calculations that the student is supposed to show. Students are not expected to write any code on the exam - their responses should contain regular mathematical notation, plus their explanations.

The model answers below are provided in code blocks to highlight what is being calculated, as well as the results of those calculations. Model answers to conceptual questions are provided as commented code blocks.

Question 1

a. Based on the diagram in Figure 1, calculate the total Entropy of the this initial data.

```r
# The student is expected to show their working to reach this solution
p.edible = 7/18
p.poison = 11/18
entropy = -(p.edible * log2(p.edible) + p.poison * log2(p.poison))
cat("\nEntropy = ", signif(entropy, 3))
```

```
##
## Entropy =  0.964
```

b. Suppose you decide that your root node will have the expression (Bush height > 0.4m). Calculate the Information gain of this split.

```r
# The student is expected to show their working to reach this solution
np = 18
n1 = 11
n2 = 7

Entr.1 = -(2/11 * log2(2/11) + 9/11 * log2(9/11))
Entr.2 = -(5/7 * log2(5/7) + 2/7 * log2(2/7))
I = n1/np * Entr.1 + n2/np * Entr.2
```

```
Gain = entropy - I
cat("\nEntropy of parent = ", signif(entropy, 3),
    "\nEntropy of child 1 = ", signif(Entr.1, 3),
    "\nEntropy of child 2 = ", signif(Entr.2, 3),
    "\nAfter-split Impurity = ", signif(I, 3),
    "\nInformation Gain = ", signif(Gain, 3))
```

```
##
## Entropy of parent =  0.964
## Entropy of child 1 =  0.684
## Entropy of child 2 =  0.863
## After-split Impurity =  0.754
## Information Gain =  0.21
```

c. For the subset of data that have (Bush height $>$ 0.4m) : True, add a leaf node. Use the class of the majority of points that reached this node to determine the node's predicted class.

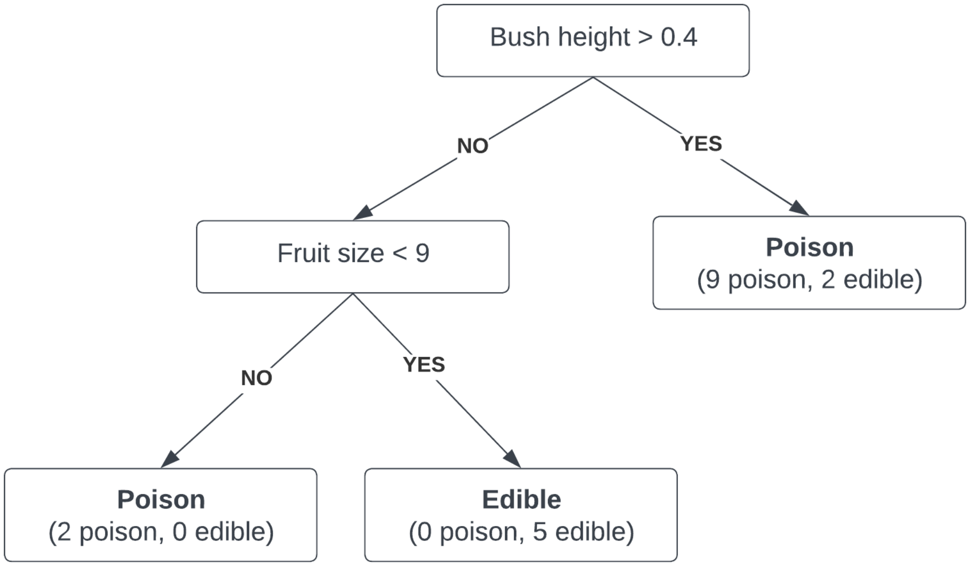
```
# Expected answer:
# This subset has 11 points, 9 of which are of class "poison".
# Therefore, the corresponding leaf node predicts class "poison".
```

d. For the subset of data that have (Bush height $>$ 0.4m) : False, add an internal node with expression (Average fruit size $<$ 9mm). Calculate the Information gain of this split and add two leaf nodes after this internal node. Use the class of the majority of points that reached each leaf node to determine the node's predicted class.

```
np  =  7
n1  =  5
n2 = 2
Entr.p = -(5/7 * log2(5/7) + 2/7 * log2(2/7))
Entr.1 = -(5/5 * log2(5/5) + 0) # 0*Log0=0
Entr.2 = -(0 + 2/2 * log(2/2)) # 0*Log0=0
I = n1/np * Entr.1 + n2/np * Entr.2
Gain = Entr.p - I
cat("\nEntropy of parent = ", signif(Entr.p, 3),
    "\nEntropy of child 1 = ", signif(Entr.1, 3),
    "\nEntropy of child 2 = ", signif(Entr.2, 3),
    "\nAfter-split Impurity = ", signif(I, 3),
    "\nInformation Gain = ", signif(Gain, 3))
```

```
##
## Entropy of parent =  0.863
## Entropy of child 1 =  0
## Entropy of child 2 =  0
## After-split Impurity =  0
## Information Gain =  0.863
##
```

e. Draw your decision tree diagram, including the expressions in the root and internal nodes, the number of points from each class and the predicted classes at each leaf node.



f. Calculate the numbers of True Positives, True Negatives, False Positives and False Negatives from your decision tree, based on the counts of points at the leaf nodes from the previous item. Assume that "edible" is the positive class.

```
# The student is expected to reference the leaf nodes from the tree above to
# derive these numbers
TP = 5
TN = 11
FP = 0
FN = 2
```

g. Draw the Confusion Matrix for your model, and use it to estimate the model's Accuracy.

```
                     Actual: Edible Actual: Poison

## Pred: Edible             5              0
## Pred: Poison             2             11
```

h. Based on the confusion matrix, calculate the Precision of your model - i.e., the probability that a fruit is actually edible, if the model predicts it as being edible.

```
PPV = TP / TP + FP
cat("PPV = ", signif(PPV, 3))
```

```
## PPV =  1
```

i. Imagine that you find a plant with Bush height = 0.6m and Average fruit size = 10mm. Based on your decision tree, should you eat the berries from this bush or not?

```
# This new observation would navigate down the DT as follows:
# (Bush~height>0.4m): Yes
# Therefore it is classified as "poison", regardless of the fruit size.
```

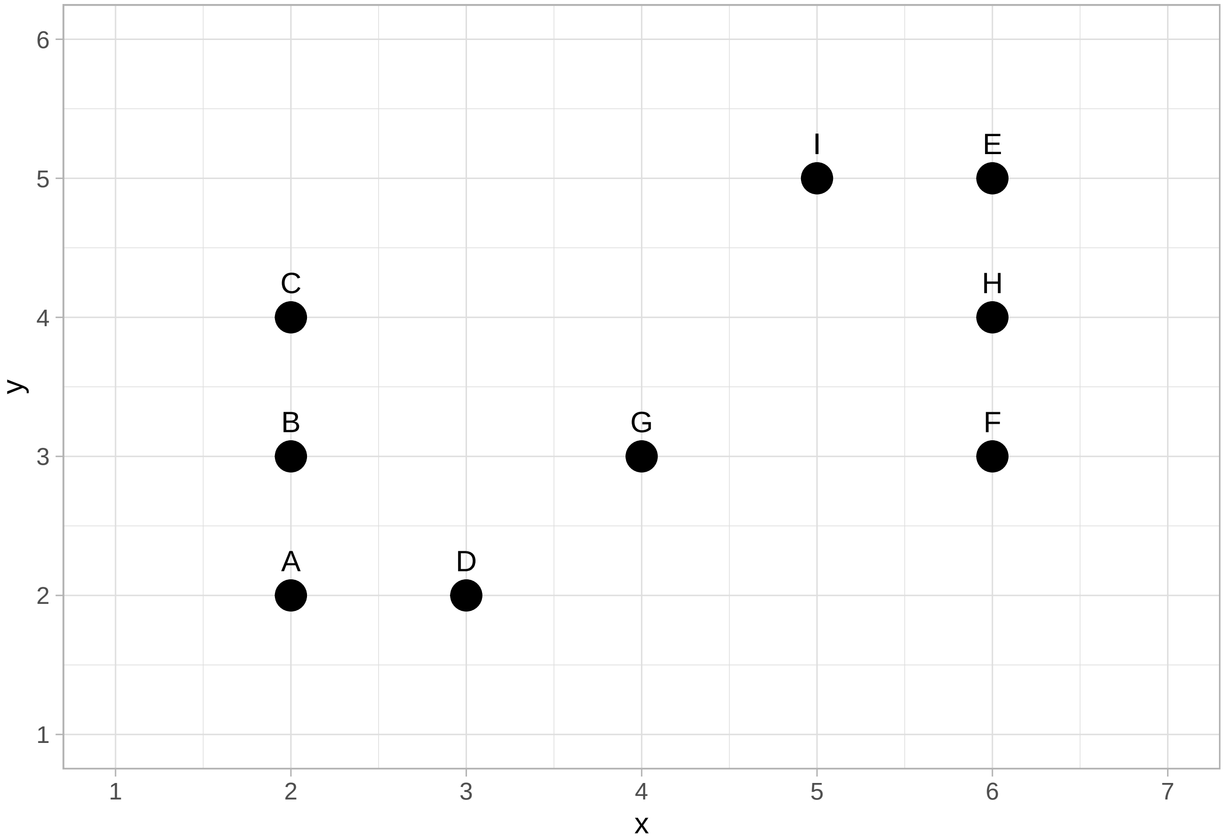
j. In this exercise we used the same data for training and evaluating our model. Discuss why this may be a problem, and suggest a simple strategy to mitigate it.

```
# Assessing a model in the same data that was used for training the model
# may not provide a good estimate of the true generalisation performance
# of the model. In many cases, the performance estimate is overly optimistic.
#
# In general, the approach to mitigate this risk would be to hold out a portion
# of the data, which would not be used to build the model. This hold-out data
# would be used to calculate the performance, providing an estimate of how the
# model is likely to perform when faced with unseen data.

# In the case of very small data (as is this case), a leave-one-out cross validation
# could be used as well (although it is a demanding process that averages
# performance over N models, each trained on N-1 data points, which could be an
# obstacle in the desert island scenario of this problem).
```

Question 3

Suppose you are given the data in the figure below:



You are asked to use the simple k-means clustering to group this data. The specifications of your k-means clustering method are as follows:

- Use $k = 2$ with initial centroids $c_1 = (4, 1)$ and $c_2 = (5, 1)$
- Use the *Manhattan distance* to calculate all distances.

Perform TWO iterations of the k-means algorithm as defined above. You should detail your process and calculations, and report: (i) the final position of the centroids, and (ii) the final attribution of points to each cluster (use the letters from the figure to refer to points allocated to each cluster).

```
# The student is expected to show their working at each step by:
# - Calculate the distance between each point and each centroid
# - attribute points to clusters based on minimal distance
# - update centroid position
#
# Note: The student does not need to produce a diagram with the allocated points,
# only the description.

# Initial centroids
C1 = c(4,1)
C2 = c(5,1)


# ===== First Iteration =====
# Calculate distances. "x" is a matrix containing the data points
# variables d1, d2 will contain the Manhattan distance to each centroid, and
# variable "cluster" will contain the consequent cluster attribution.
```

```
##
##       x     y Labels    d1    d2 Cluster

## 1     2     2 A          3     4       1
## 2     2     3 B          4     5       1
## 3     2     4 C          5     6       1
## 4     3     2 D          2     3       1
## 5     6     5 E          6     5       2
## 6     6     3 F          4     3       2
## 7     4     3 G          2     3       1
## 8     6     4 H          5     4       2
## 9     5     5 I          5     4       2
```

```
# Update centroids
(C1 = c(mean(x[Cluster == 1]), mean(y[Cluster == 1])))
```

```
## [1] 2.6 2.8
```

```
(C2 = c(mean(x[Cluster == 2]), mean(y[Cluster == 2])))
```

```
## [1] 5.75 4.25
```

```
# ===== Second Iteration =====
```

Repeat same process as above to update centroids

```
## Final Centroids:
C1
```

```
## [1] 2.6 2.8
```

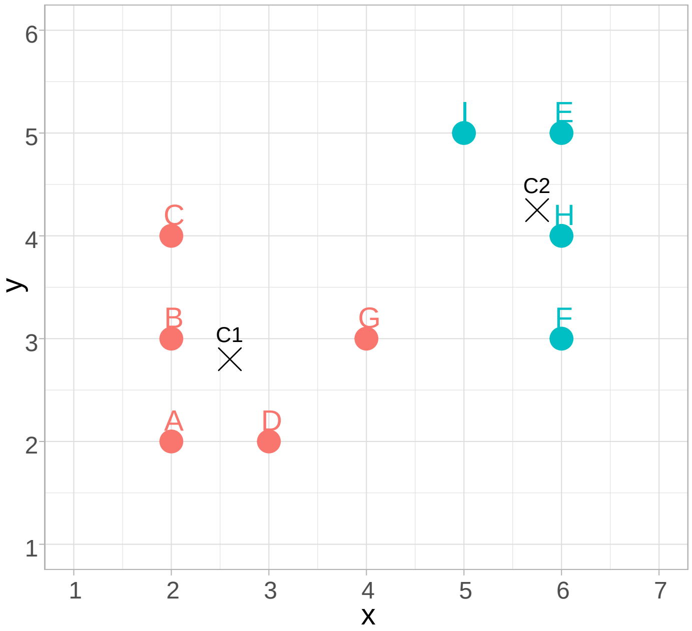
```
C2
```

```
## [1] 5.75 4.25
```

```
## Final Allocation:

##   Labels Cluster

  ##    A        1
  ##    B        1
  ##    C        1
  ##    D        1
  ##    G        1
  ##    E        2
  ##    F        2
  ##    H        2
  ##   I        2
```



- Choose either DBSCAN or agglomerative hierarchical clustering (pick only one) and provide a short explanation of chosen method works. Include the main general idea behind the chosen method, and a sketch of the main steps (or a pseudocode) indicating how the method works. Indicate at least one advantage and one drawback of your selected method.

```
# The student is expected to explain the main aspects of one of the two methods
# and provide a broad step-by-step explanation of how the
# chosen method works.


  - For DBSCAN this includes the definition of the two main parameters
# (eps and minPts), the definition of core, border and noise points, and the
# definition of clusters based on density edges.
# Advantages of DBSCAN can be, e.g. ability to work with arbitrarily-shaped clusters
# or to sutomatically determine the final number of clusters.
# Disadvantages can include sensitivity to parameter values or inability to deal
# with clusters of different densities.
# (Other dis/advantages can also be listed by the student, assuming they make sense).

  - For Agglomerative clustering the student is expected to discuss the choice of
# a linkage function and the progressive joining of points or clusters based on
# the linkage value, until a single supercluster emerges.
# Advantages can include the ability to operate in high-dimensional spaces and the
# fact that the number of clusters does not need to be determined in advance.
# Disadvantages can include a high computational cost (high time complexity) or
# sensitivity to the choice of linkage function.
```