Consider the following confusion table summarising the testing results for iris classification. As, you are aware, the iris data is a classic multi-class benchmark dataset (12 marks)

Confusion Matrix for IRIS dataset		Actual Class		
		Setosa	Versicolor	Virginica
Predicted Class	Setosa	20	0	0
	Versicolor	1	1	1
	Virginica	0	4	16

- **What is the overall classification accuracy? (2 marks)**
- **What is the sensitivity and specificity for each class? (6 marks)**
- **Use the table as an example to explain why confusion matrix is a better way to assess the performance of a classifier than the overall classification accuracy (4 marks)**

Indicative answer:

The overall classification accuracy is: (20+1+16)/(20+1+1+4+1+16) = 86.046%

As this is a multi-class problem, we need to decompose it, so to create "individual" confusion matrices (CM) for each, just to make easier the understanding of the concept. The following CM shows the details.

		Actual Class	
		Yes	Not
Predicted Class	Yes	TP	FP
	Not	FN	TN

Confusion Matrix for IRIS dataset		Actual Class	
		Setosa	Not
Predicted Class	Setosa	20	0
	Not	1	22

Confusion Matrix for IRIS dataset		Actual Class	
		Versicolor	Not
Predicted Class	Versicolor	1	2
	Not	4	36

Confusion Matrix for IRIS dataset		Actual Class	
		Virginica	Not
Predicted Class	Virginica	16	4
	Not	1	22

Sensitivity: TP/(TP+FN)
Specificity: TN/(TN+FP)

For setosa: Sensitivity: 20/21 = 95.23%, specificity: 22/22 = 100%
For Versicolor: Sensitivity: 1/5= 20%, specificity: 36/38 = 94.73%
For Virginica: Sensitivity: 16/17 = 94.11%, specificity: 22/26 = 84.61%

Classification accuracy can be unreliable when assessing unbalanced data. In the above example, the overall classification accuracy of 86% shows the classifier does well in general. It does not reflect the poor performance when classifying the Versicolor class, which has small sample number. The confusion matrix shows a rather poor performance of classifying the Versicolour class, which also leads to low sensitivity.

You are training a Multilayer Perceptron (MLP) neural network for a particular classification task. After, some investigation, your neural network is constructed with 5 input variables, one hidden layer with 12 nodes and one output layer with 3 nodes (the classes). How many network parameters are required to be tuned/trained? Show your detailed calculations. (4 marks)

Indicative answer: (5+1) x 12 + (12+1) x 3 = 111. We have to include also the bias "weights" which is a compulsory component in MLP structure.

Given three clusters, X, *Y* and *Z*, containing a total of six points, where each point is defined by an integer value in one dimension, X = {0, 2, 6}, Y = {3, 9} and *Z* = {11}, which two clusters will be merged at the next iteration stage of Hierarchical Agglomerative Clustering when using the standard Euclidean distance and (i) Single Linkage (4 marks), (ii) Complete Linkage (4 marks). Justify your response for these two cases, by showing all steps of your work.

Indicative answer: For each case of linkage, we need to find the distances of X&Y, X&Z and Y&Z. Then, we need to choose the minimum of them. In single cluster linkage we need to find the minimum of individual combined components, while in the case of complete, the maximum.

<u>Single linkage.</u>
Dist (X,Y) = min {d(0,3), d(0,9), d(2,3), d(2,9), d(6,3), d(6,9)} = min{3,9,1,7,3,3} = 1 (1 mark)
Dist (X,Z) = min {d(0,11), d(2,11), d(6,11)} = min{11, 9, 5} = 5 (1 mark)
Dist (Y,Z) = min {d(3,11), d(9,11)} = min{8, 2} = 2 (1 mark)
For single linkage, the choice will be XY, as it has the minimum distance. (1 mark)

<u>Complete linkage.</u>
Dist (X,Y) = max {d(0,3), d(0,9), d(2,3), d(2,9), d(6,3), d(6,9)} = max{3,9,1,7,3,3} = 9 (1 mark)
Dist (X,Z) = max {d(0,11), d(2,11), d(6,11)} = max{11, 9, 5} = 11 (1 mark)
Dist (Y,Z) = max {d(3,11), d(9,11)} = max{8, 2} = 8 (1 mark)
For complete linkage, the choice will be YZ, as it has the minimum distance. (1 mark)

Question B-1

The following table is a set of *three-course menus* from a famous restaurant. The idea here, is to create a decision tree (using the ID3 algorithm) to correctly classify similar examples. As the aim is to classify menus regarding whether they are good or not, calculate whether this algorithm would use "Starter" or "Main course" as the root of the decision tree. In this specific question you will use the entropy and information gain concepts. For the calculation of $\log_2$, remember, from maths, that $\log_a b = \dfrac{\log_{10} b}{\log_{10} a}$ **.**

Starter	Main Course	Dessert	Good Menu
salad	steak	cheesecake	yes
soup	salmon	profiteroles	yes
salad	variety-roast	Fruit-salad	no
salad	surprise-bake	cheesecake	no
soup	variety-roast	Fruit-salad	yes
salad	salmon	Fruit-salad	yes
salad	variety-roast	Fruit-salad	no

You need to address/calculate the following issues:

- ***Total parent entropy: 2 marks***

- ***Starter case: two individual entropies (2x2= 4 marks), weighted entropy (2 marks), information gain (2 mark)***
- ***Main course case: four individual entropies (4x2=8 marks), weighted entropy (2 marks), information gain (2 marks)***
- ***Final Decision (2 marks)***

Show all steps of your work (24 marks).

Solution:

The total parent entropy (i.e. good menu) is calculated as:

$$Entropy(S) = -\frac{4}{7}\log_2\left(\frac{4}{7}\right) - \frac{3}{7}\log_2\left(\frac{3}{7}\right) = 0.985$$

We need to check the entropy for each one of these two attributes individually and then find which attribute maximizes the information gain.

Starter attribute:

Values	Yes	No	Entropy(S_i)
S_1 (salad)	xx	xxx	[2/5, 3/5]
S_2(soup)	xx		[1, 0]

$$Entropy(S_1) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) = 0.971$$

$$Entropy(S_2) = -\log_2(1) = 0$$

Therefore, the entropy for this attribute is:

$$Entropy(S \mid starter) = \frac{5}{7}\cdot 0.971 + \frac{2}{7}\cdot 0 = 0.693$$

The information gain for Starter is: Entropy(S) – Entropy(S | starter) = 0.985 – 0.693 = 0.292

Main course attribute:

Values	Yes	No	Entropy(S_i)
S_1(steak)	1	0	[1, 0]
S_2(salmon)	2	0	[2/2, 0] = [1, 0]
S_3(variety-roast)	1	2	[1/3, 2/3]
S_4(surprise-bake)	0	1	[0, 1]

$$Entropy(S_1) = -\log_2(1) - 0 = 0$$

$$Entropy(S_2) = -\log_2(1) - 0 = 0$$

$$Entropy(S_3) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) = 0.918$$

$$Entropy(S_4) = 0 - \log_2(1) = 0$$

Therefore, the entropy for this attribute is:

$$Entropy(S \mid main\ course) = \frac{1}{7}\cdot 0 + \frac{2}{7}\cdot 0 + \frac{3}{7}\cdot 0.918 + \frac{1}{7}\cdot 0 = 0.393$$

The information gain for main course is: Entropy(S) – Entropy(S |main course) = 0.985 – 0.393=0.592

Main course produces the highest gain, and thus this attribute will be the root of the tree.

Consider the following confusion matrix (CM) summarising the testing results for a Fruit dataset classification (14 marks in total).

Confusion Matrix for Fruit dataset		Predicted Class		
		Apple	Pears	Grapes
Actual Class	Apple	6	0	2
	Pears	3	9	1
	Grapes	1	0	10

- **What is the overall classification accuracy of this CM? (2 marks)**
- **Decompose the above 3x3 CM into three individual (per fruit) 2x2 CMs (3x2= 6 marks)**
- **What is the sensitivity and specificity for each class? (3x2 = 6 marks)**
  Show all steps of your work.

Indicative answer:
The overall classification accuracy is: (6+9+10))/32 = 78.125%
As this is a multi-class problem, we need to decompose it, so to create "individual" CMs for each fruit, just to make easier the understanding of the concept. The following CM shows the details.

		Predicted Class	
		Yes	Not
Actual Class	Yes	TP	FN
	Not	FP	TN

Confusion Matrix for Fruit dataset		Predicted Class	
		Apple	Not
Actual Class	Apple	6	2
	Not	4	20

Confusion Matrix for Fruit dataset		Predicted Class	
		Pears	Not
Actual Class	Pears	9	4
	Not	0	19

Confusion Matrix for Fruit dataset		Predicted Class	
		Grapes	Not
Actual Class	Grapes	10	1
	Not	3	18

Sensitivity: TP/(TP+FN)
Specificity: TN/(TN+FP)

For apple: Sensitivity: 6/8 = 75%, specificity: 20/24 = 83.33%
For pears: Sensitivity: 9/13=69.23%, specificity: 19/19 = 100%

For grapes: Sensitivity: 10/11 = 90.91%, specificity: 18/21 = 85.71%

A z-score measures exactly how many standard deviations above or below the mean a data point is. Here's the formula for calculating a z-score:

$$z = \frac{\text{data point} - \text{mean}}{\text{standard deviation}}$$

Here's the same formula written with symbols:

$$z = \frac{x - \mu}{\sigma}$$

<u>Here are some important facts about z-scores:</u>

- A positive z-score says the data point is above average.
- A negative z-score says the data point is below average.
- A z-score close to 0 says the data point is close to average.

Suppose, we have the same 4 numbers: 8, 10, 15, 20, and we wish to find their z- score

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

$$\text{Standard deviation} = \sqrt{\frac{\sum(\textit{every individual value of marks} - \textit{mean of marks})^2}{n}}$$

Mean of marks = 8 + 10 + 15 + 20 /4 = 13.25

$$= \sqrt{\frac{(8 - 13.25)^2 + (10 - 13.25)^2 + (15 - 13.25)^2 + (20 - 13.25)^2}{4}}$$

$$= \sqrt{\frac{(-5.25)^2 + (-3.25)^2 + (1.75)^2 + (6.75)^2}{4}}$$

$$= \sqrt{\frac{27.56 + 10.56 + 3.06 + 45.56}{4}} = \sqrt{\frac{86.74}{4}} = \sqrt{21.6} = 4.6$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{8 - 13.25}{4.6} = -1.14$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{10 - 13.25}{4.6} = -0.7$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{15 - 13.25}{4.6} = 0.3$$

$$ZScore = \frac{x - \mu}{\sigma} = \frac{20 - 13.25}{4.6} = 1.4$$

Market Basket Analysis is one of the key techniques used by large retailers to uncover associations between items. It works by looking for combinations of items that occur together frequently in

transactions. Association Rules are widely used to analyse retail basket or transaction data. You have been given the following transaction database that consists of items (a, b, c, d &e) bought in a store by customers.

TID	Items
1	a,b,d,e
2	b,c,d
3	a,b,d,e
4	a,c,d,e
5	b,c,d,e
6	b,d,e
7	c,d
8	a,b,c
9	b,d
10	a,d,e

Find all the closed frequent itemsets which are not maximal, along with their support, for a minsupp threshold of 0.3. Procedure: Define first, all the frequent itemsets, then all the closed frequent itemsets and finally all the closed frequent itemsets which are not maximal. Show all steps/results of your work and justify any decision you have taken in your analysis.

Indicative answer:

We check first the 1-size itemsets

a 5/10 = 0.5 (5 counts divided by the number of transactions)
b 0.7
c 0.5
d 0.9
e 0.6

Everything is above min support so they are candidates for the creation of 2-size itemsets

ab	0.3	
ac	0.2	reject, due to less 0.3
ad	0.4	
ae	0.4	
bc	0.3	
bd	0.6	
be	0.4	
cd	0.4	
ce	0.2	reject, due to less 0.3
de	0.6	

8 2-itemsets are suitable; they will create the 3-size itemsets in the next stage

Abc	reject as it contains ac
Abd	
Abe	
Acd	reject due to ac
Ace	reject due to ac
Ade	
Bcd	
Bce	reject due to ce
Bde	
Cde	reject due to ce

Thus, only 5 3-itemsets are suitable candidates. We need to calculate their support

Abd	0.2	reject, due to less 0.3
Abe	0.2	reject, due to less 0.3

Ade	0.4	
Bcd	0.2	reject, due to less 0.3
Bde	0.4	

These 2 3-itemsets are ok for the creation of 4-itemsets

Abcd		reject as it includes ac
Abce		reject as it includes bc
Abde		reject as it includes abd
Acde	0.1	reject, less 0.3
Acde	0.1	reject, less 0.3

So, finally we have 5 1-itemsets, 8 2-itemsets and 2 3-itemsets (all frequent)

In summary, these are the following frequent itemsets

items	support
[1] {c}	0.5
[2] {a}	0.5
[3] {e}	0.6
[4] {b}	0.7
[5] {d}	0.9
[6] {b,c}	0.3
[7] {c,d}	0.4
[8] {a,e}	0.4
[9] {a,b}	0.3
[10] {a,d}	0.4
[11] {b,e}	0.4
[12] {d,e}	0.6
[13] {b,d}	0.6
[14] {a,d,e}	0.4
[15] {b,d,e}	0.4

Based on definitions:

- **Closed Frequent Itemset:** An itemset is closed if none of its immediate supersets has the same support as that of the itemset.
- **Maximal frequent itemset:** The definition says that an itemset is maximal frequent if none of its immediate supersets is frequent

Let's try to find the closed frequent itemsets

items	**support**
[1] {c}	0.5
[2] {a}	0.5
[3] {b}	0.7
[4] {d}	0.9
[5] {b,c}	0.3
[6] {c,d}	0.4
[7] {a,b}	0.3
[8] {d,e}	0.6
[9] {b,d}	0.6
[10] {a,d,e}	0.4
[11] {b,d,e}	0.4

There are 4 itemsets from the frequent list (e, ae, ad, be) that are not closed, so they have been omitted from the closed list. So, “e” for example has been excluded as “de” has also 0.6. “ae” the same due to “ade”, “ad” due to “ade” and “be” is also omitted due to “bde” (the same support).

But we are interested in those closed itemsets which are not maximal ones. Let’s try to find those which are maximal ones and remove them for the closed list. Then, the remaining ones will be the required answer.

The maximal ones are:

	items	support
[1]	{b,c}	0.3
[2]	{c,d}	0.4
[3]	{a,b}	0.3
[4]	{a,d,e}	0.4
[5]	{b,d,e}	0.4

Thus, if we remove them from the above list of closed ones (11 itemsets), we have the final result.

[1]	{c}	0.5
[2]	{a}	0.5
[3]	{b}	0.7
[4]	{d}	0.9
[8]	{d,e}	0.6
[9]	{b,d}	0.6