(Affiliated to University of Mumbai)	***End Semester Examination (R-24/R-25) SH 2026 Answer Key with marking scheme***
Branch: EXTC/CSE-IOT/IT	Course: MDM-Machine Learning
Year/ Semester: T.E/V	Course code: MDMC5012
Time: 03 hours	Marks: 80

		Marks
Q. 1	All questions are compulsory.	
A.	ML workflow steps – 2 marks Data preprocessing and its importance – 1 mark Data splitting and its purpose – 1 mark Model evaluation and performance metrics – 1 mark	
B.	**Step 1: Truth Table for OR Gate** Assume weights $w_1 = 1,\ w_2 = 1$ and bias $b = -1$. Net input: $z = x_1 w_1 + x_2 w_2 + b$ Activation function: $y = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$ x_1 | x_2 | Expected OR output | Net input $z = x_1 + x_2 - 1$ 0 | 0 | 0 | −1 0 | 1 | 1 | 0 1 | 0 | 1 | 0 1 | 1 | 1 | 1 **Step 2: Determine the Threshold** For the McCulloch–Pitts neuron: $y = \begin{cases} 1, & \sum w_i x_i \geq \theta \\ 0, & \sum w_i x_i < \theta \end{cases}$ $\boxed{\theta = 1}$ Thus, $w_1 = 1,\ w_2 = 1,\ \theta = 1$ correctly implement the OR operation. Truth table – 1 mark Selection of weights and bias – 1 mark Net input and predicted outputs – 2 marks Threshold determination – 1 mark	
C.	Working of SVM algorithm – 2 marks Neat labelled diagram – 1 mark Hyperplane and support vectors – 1 mark Margin and its significance – 1 mark	

D.	Bias and its explanation – 1 mark Variance and its explanation – 1 mark Underfitting and overfitting – 2 marks Bias–variance trade-off – 1 mark	
E.	**Given:** • Dataset: $X = \{2, 4, 6, 8, 10, 12\}$ • Number of clusters: $K = 2$ • Initial centroids: $C_1 = 2,\ C_2 = 6$ • Distance measure: Euclidean distance For one-dimensional data: $d(x, C) = \lvert x - C \rvert$ **Iteration 1** Initial centroids: $C_1 = 2,\ C_2 = 6$	

Data point	Distance from $C_1 = 2$	Distance from $C_2 = 6$	Cluster
2	0	4	C1
4	2	2	C1*
6	4	0	C2
8	6	2	C2
10	8	4	C2
12	10	6	C2

For equal distances, assign the point to Cluster 1.

Updated clusters:

$C_1 = \{2, 4\}$

$C_2 = \{6, 8, 10, 12\}$

New centroids:

$$C_1 = \frac{2+4}{2} = \boxed{3}$$

$$C_2 = \frac{6+8+10+12}{4} = \boxed{9}$$

Iteration 2

Updated centroids: $C_1 = 3,\ C_2 = 9$

Data point	Distance from $C_1 = 3$	Distance from $C_2 = 9$	Cluster
2	1	7	C1
4	1	5	C1
6	3	3	C1*
8	5	1	C2
10	7	1	C2
12	9	3	C2

Updated clusters:

$C_1 = \{2, 4, 6\}$

$C_2 = \{8, 10, 12\}$

New centroids:

$$C_1 = \frac{2+4+6}{3} = \boxed{4}$$

$$C_2 = \frac{8+10+12}{3} = \boxed{10}$$

Iteration 3

Updated centroids: $C_1 = 4,\ C_2 = 10$

Data point	Distance from $C_1 = 4$	Distance from $C_2 = 10$	Cluster
2	2	8	C1
4	0	6	C1
6	2	4	C1
8	4	2	C2
10	6	0	C2
12	8	2	C2

New centroids:

$$C_1 = \frac{2+4+6}{3} = \boxed{4}$$

$$C_2 = \frac{8+10+12}{3} = \boxed{10}$$

Since the centroids remain unchanged, **the algorithm has converged.**

- **Cluster 1:** 6, Centroid = 4
- **Cluster 2:** 12, Centroid = 10
- **Number of iterations:** 3 (including convergence verification)

Iteration 1 – Distance calculations and cluster formation – 1 mark
Iteration 1 – Centroid calculation – 1 mark
Iteration 2 – Distance calculations and centroid update – 1 mark
Iteration 3 – Convergence verification – 1 mark
Final clusters and centroids – 1 mark

F. Five relevant differences between bagging and boosting – 1 mark each.

Q.2A

Given data:

Obs.	X_1	X_2	Y
1	10	5	60
2	20	5	80
3	10	10	75
4	20	10	95

Step 1: Multiple Linear Regression Equation

The general equation is:

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2$$

Where:

- b_0 = Intercept
- b_1 = Coefficient of advertisement expenditure
- b_2 = Coefficient of discount

Step 2: Calculate the Required Values

X_1	X_2	Y	X_1^2	X_2^2	X_1X_2	X_1Y	X_2Y
10	5	60	100	25	50	600	300
20	5	80	400	25	100	1600	400
10	10	75	100	100	100	750	750
20	10	95	400	100	200	1900	950
60	**30**	**310**	**1000**	**250**	**450**	**4850**	**2400**

Thus,

$$n = 4, \quad \sum X_1 = 60, \quad \sum X_2 = 30$$
$$\sum Y = 310, \quad \sum X_1^2 = 1000, \quad \sum X_2^2 = 250$$
$$\sum X_1X_2 = 450, \quad \sum X_1Y = 4850, \quad \sum X_2Y = 2400$$

Step 3: Form the Normal Equations

The least-squares normal equations are:

$$\sum Y = nb_0 + b_1 \sum X_1 + b_2 \sum X_2$$

$$\sum X_1Y = b_0 \sum X_1 + b_1 \sum X_1^2 + b_2 \sum X_1X_2$$

$$\sum X_2Y = b_0 \sum X_2 + b_1 \sum X_1X_2 + b_2 \sum X_2^2$$

Substituting the calculated values:

$$4b_0 + 60b_1 + 30b_2 = 310 \quad (1)$$

$$60b_0 + 1000b_1 + 450b_2 = 4850 \quad (2)$$

$$30b_0 + 450b_1 + 250b_2 = 2400 \quad (3)$$

Step 4: Solve the Equations

$$\boxed{b_0 = 25} \quad \boxed{b_1 = 2} \quad \boxed{b_2 = 3}$$

Therefore, the multiple linear regression equation is:

$$\boxed{\hat{Y} = 25 + 2X_1 + 3X_2}$$

Step 5: Predict Monthly Sales

Given:

Advertisement expenditure = ₹30,000, hence $X_1 = 30$

Discount = 10%, hence $X_2 = 10$

$$\hat{Y} = 25 + 2(30) + 3(10)$$

$$\hat{Y} = 25 + 60 + 30$$

$$\boxed{\hat{Y} = 115}$$

Since sales are expressed in ₹ thousands:

Predicted monthly sales = ₹1,15,000.

Regression equation formulation – 1 mark
Calculation of required summations – 2 marks
Formation of normal equations – 2 marks
Calculation of regression coefficients – 3 marks
Final regression equation – 1 mark
Prediction of monthly sales – 1 mark

OR

Q.2B.

Step 1: Calculate Entropy of the Dataset

Total samples = 8

- Apply for Loan = Yes: 5
- Apply for Loan = No: 3

Entropy formula:

$$H(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

$$H(S) = -\frac{5}{8}\log_2 \frac{5}{8} - \frac{3}{8}\log_2 \frac{3}{8}$$

$$\boxed{H(S) = 0.9544}$$

Step 2: Calculate Information Gain for Each Attribute

Information Gain:

$$IG(S, A) = H(S) - \sum_v \frac{|S_v|}{|S|} H(S_v)$$

A. Income

Income	Yes	No	Entropy
High	3	1	0.8113
Low	2	2	1.0000

$$IG(S, \text{Income}) = 0.9544 - \left[\frac{4}{8}(0.8113) + \frac{4}{8}(1)\right]$$

$$\boxed{IG(S, \text{Income}) = 0.0488}$$

B. Employment

Employment	Yes	No	Entropy
Salaried	3	1	0.8113
Self-employed	2	2	1.0000

$$IG(S, \text{Employment}) = 0.9544 - \left[\frac{4}{8}(0.8113) + \frac{4}{8}(1)\right]$$

$$\boxed{IG(S, \text{Employment}) = 0.0488}$$

C. Credit History

Credit History	Yes	No	Entropy
Good	4	0	0
Poor	1	3	0.8113

$$IG(S, \text{Credit}) = 0.9544 - \left[\frac{4}{8}(0) + \frac{4}{8}(0.8113)\right]$$

$$\boxed{IG(S, \text{Credit}) = 0.5488}$$

Step 3: Select the Root Node

Attribute	Information Gain
Income	0.0488
Employment	0.0488
Credit History	**0.5488**

Since Credit History has the highest Information Gain, **Credit History is selected as the root node.**

Step 4: Construct the Decision Tree

For Credit History = Good, all four customers applied for loans (Yes). Hence, no further splitting is required.

For Credit History = Poor, there are 4 samples (1 Yes, 3 No), with entropy 0.8113.

Information Gain within this subset:

$$IG(\text{Income}) = 0.8113 - \left[\frac{2}{4}(1) + \frac{2}{4}(0)\right] = 0.3113$$

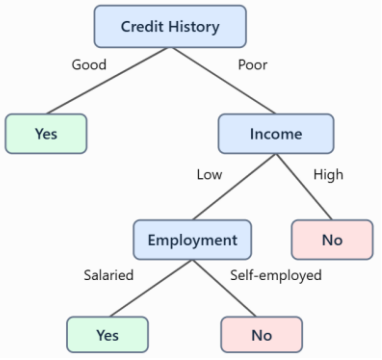
$$IG(\text{Employment}) = 0.3113$$

Both attributes have equal Information Gain. Select Income to break the tie.

For Poor credit history:

- Income = Low → No
- Income = High → split further using Employment
  - Salaried → Yes
  - Self-employed → No

Step 5: Final Decision Tree



Entropy of complete dataset – 2 marks
Information Gain for three attributes – 3 marks
Selection of root node – 1 mark
Further splitting and Information Gain – 2 marks
Final Decision Tree – 2 marks

Q.3A

Given:

- Initial weights: $w_1 = 0,\ w_2 = 0$
- Initial bias: $b = 0$
- Learning rate: $\alpha = 1$
- Threshold: $\theta = 0.5$

Net input:

$$z = w_1 x_1 + w_2 x_2 + b$$

Activation function:

$$y = \begin{cases} 1, & z \geq 0.5 \\ 0, & z < 0.5 \end{cases}$$

Weight and bias update rules:

$$w_i^{new} = w_i^{old} + \alpha(t - y)x_i$$

$$b^{new} = b^{old} + \alpha(t - y)$$

Step 1: OR Gate Truth Table

x_1	x_2	Target t
0	0	0
0	1	1
1	0	1
1	1	1

Step 2: Epoch 1

Initial: $w_1 = 0,\ w_2 = 0,\ b = 0$

x_1	x_2	t	Net z	y	Error $t - y$	Updated (w_1, w_2, b)
0	0	0	0	0	0	(0, 0, 0)
0	1	1	0	0	1	(0, 1, 1)
1	0	1	1	1	0	(0, 1, 1)
1	1	1	2	1	0	(0, 1, 1)

Weights after Epoch 1:

$$\boxed{w_1 = 0, \quad w_2 = 1, \quad b = 1}$$

Step 3: Epoch 2

Initial: $w_1 = 0,\ w_2 = 1,\ b = 1$

x_1	x_2	t	Net z	y	Error $t - y$	Updated (w_1, w_2, b)
0	0	0	1	1	−1	(0, 1, 0)
0	1	1	1	1	0	(0, 1, 0)
1	0	1	0	0	1	(1, 1, 1)
1	1	1	3	1	0	(1, 1, 1)

Weights after Epoch 2:

$$\boxed{w_1 = 1, \quad w_2 = 1, \quad b = 1}$$

Final Result

After two epochs:

$$\boxed{z = x_1 + x_2 + 1}$$

The perceptron has not yet converged, since the final parameters incorrectly classify input (0,0) as 1. Further training is required to correctly implement the OR gate.

OR gate truth table and activation function – 1 mark
Weight and bias update equations – 2 marks
Epoch 1 calculations – 3 marks
Epoch 2 calculations – 3 marks
Final weights, bias and conclusion – 1 mark

OR

Q.3B

Step 1: Given Parameters

- Initial weights: $w_1 = 0.2,\ w_2 = 0.2$
- Initial bias: $b = 0.2$
- Learning rate: $\alpha = 0.2$
- Number of epochs: 1

AND Gate Truth Table

x_1	x_2	Target t
0	0	0
0	1	0
1	0	0
1	1	1

Step 2: ADALINE Learning Equations

Net input (linear output):

$$y_{in} = w_1x_1 + w_2x_2 + b$$

Error: $e = t - y_{in}$

Weight update: $w_i^{new} = w_i^{old} + \alpha e x_i$

Bias update: $b^{new} = b^{old} + \alpha e$

Step 3: Training – Epoch 1

Weights and bias are updated after each input pattern. Calculations use the continuous linear output, not the thresholded binary output.

Inputs (x_1, x_2)	Target	Net input	Error	Updated w_1	Updated w_2	Updated b
(0,0)	0	0.2000	−0.2000	0.2000	0.2000	0.1600
(0,1)	0	0.3600	−0.3600	0.2000	0.1280	0.0880
(1,0)	0	0.2880	−0.2880	0.1424	0.1280	0.0304
(1,1)	1	0.3008	0.6992	0.28224	0.26784	0.17024

Step 4: Sample Calculations

Pattern 1: (0,0), Target = 0

$$y_{in} = 0.2(0) + 0.2(0) + 0.2 = 0.2$$

$$e = 0 - 0.2 = -0.2$$

$$w_1 = 0.2, \quad w_2 = 0.2$$

$$b = 0.2 + 0.2(-0.2) = 0.16$$

Pattern 2: (0,1), Target = 0

$$y_{in} = 0.2(0) + 0.2(1) + 0.16 = 0.36$$

$$e = -0.36$$

$$w_1 = 0.2$$

$$w_2 = 0.2 + 0.2(-0.36)(1) = 0.128$$

$$b = 0.16 + 0.2(-0.36) = 0.088$$

Pattern 3: (1,0), Target = 0

$$y_{in} = 0.2(1) + 0.128(0) + 0.088 = 0.288$$

$$e = -0.288$$

$$w_1 = 0.2 + 0.2(-0.288) = 0.1424$$

$$w_2 = 0.128$$

$$b = 0.088 + 0.2(-0.288) = 0.0304$$

Pattern 4: (1,1), Target = 1

$$y_{in} = 0.1424 + 0.128 + 0.0304 = 0.3008$$

$$e = 1 - 0.3008 = 0.6992$$

$$w_1 = 0.1424 + 0.2(0.6992) = 0.28224$$

$$w_2 = 0.128 + 0.2(0.6992) = 0.26784$$

$$b = 0.0304 + 0.2(0.6992) = 0.17024$$

Step 5: Final Network Parameters

After one complete epoch:

$$\boxed{w_1 = 0.28224}$$

$$\boxed{w_2 = 0.26784}$$

$$\boxed{b = 0.17024}$$

Final ADALINE linear output equation:

$$\boxed{y_{in} = 0.28224x_1 + 0.26784x_2 + 0.17024}$$

AND gate truth table – 1 mark
ADALINE learning equations – 2 marks
Pattern 1 calculations – 1 mark
Pattern 2 calculations – 2 marks
Pattern 3 calculations – 2 marks
Pattern 4 calculations – 1 mark
Final weights and bias – 1 mark

Q.4A

Given Dataset:

Observation	X_1	X_2
1	1	2
2	5	2
3	3	1
4	3	3

Step 1: Calculate the Mean

$$\bar{X}_1 = \frac{1+5+3+3}{4} = 3$$

$$\bar{X}_2 = \frac{2+2+1+3}{4} = 2$$

Mean vector:

$$\mu = \begin{bmatrix} 3 & 2 \end{bmatrix}$$

Step 2: Mean Centering

Subtract the mean from each observation.

Observation	$X_1 - 3$	$X_2 - 2$
1	−2	0
2	2	0
3	0	−1
4	0	1

Mean-centered data:

$$X_c = \begin{bmatrix} -2 & 0 \\ 2 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix}$$

Step 3: Calculate Covariance Matrix

Using sample covariance:

$$C = \frac{1}{n-1} X_c^T X_c$$

$$X_c^T X_c = \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix}$$

Therefore,

$$C = \frac{1}{3} \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\boxed{C = \begin{bmatrix} 2.667 & 0 \\ 0 & 0.667 \end{bmatrix}}$$

Step 4: Calculate Eigenvalues and Eigenvectors

Characteristic equation:

$$|C - \lambda I| = 0$$

$$(2.667 - \lambda)(0.667 - \lambda) = 0$$

Eigenvalues:

$$\boxed{\lambda_1 = \frac{8}{3} = 2.667}$$

$$\boxed{\lambda_2 = \frac{2}{3} = 0.667}$$

Corresponding eigenvectors:

$$v_1 = \begin{bmatrix}1\\0\end{bmatrix}, \qquad v_2 = \begin{bmatrix}0\\1\end{bmatrix}$$

Step 5: Select the Principal Component

Since $\lambda_1 > \lambda_2$, select the eigenvector corresponding to the largest eigenvalue.

$$\boxed{PC_1 = \begin{bmatrix}1\\0\end{bmatrix}}$$

Explained variance ratio:

$$\frac{\lambda_1}{\lambda_1 + \lambda_2} \times 100 = \frac{2.667}{3.334} \times 100$$

$$\boxed{80\%}$$

Step 6: Project Data onto the First Principal Component

$$Z = X_c \times PC_1$$

$$Z = \begin{bmatrix}-2 & 0\\ 2 & 0\\ 0 & -1\\ 0 & 1\end{bmatrix} \begin{bmatrix}1\\0\end{bmatrix}$$

$$\boxed{Z = \begin{bmatrix}-2\\2\\0\\0\end{bmatrix}}$$

Observation	Original (X_1, X_2)	Reduced 1D Data
1	(1, 2)	−2
2	(5, 2)	2
3	(3, 1)	0
4	(3, 3)	0

PCA reduces the dataset from two dimensions to one dimension while retaining 80% of the total variance. The signs of all projected values may be reversed if the eigenvector (−1, 0) is selected instead; both answers are correct.

Mean calculation – 1 mark
Mean centering – 2 marks
Covariance matrix calculation – 2 marks
Eigenvalues and eigenvectors – 2 marks
Selection of principal component – 1 mark
Projection onto 1D – 2 marks

OR

Q.4B

Step 1: Given Dataset

Employee	X_1	X_2	Y
1	2	1	40
2	4	3	50
3	6	2	60
4	8	5	70
5	10	4	80

Step 2: Pearson's Correlation Formula

$$r_{XY} = \frac{n\sum XY - (\sum X)(\sum Y)}{\sqrt{[n\sum X^2 - (\sum X)^2][n\sum Y^2 - (\sum Y)^2]}}$$

Step 3: Calculate Required Values

X_1	X_2	Y	X_1^2	X_2^2	Y^2	X_1Y	X_2Y
2	1	40	4	1	1600	80	40
4	3	50	16	9	2500	200	150
6	2	60	36	4	3600	360	120
8	5	70	64	25	4900	560	350
10	4	80	100	16	6400	800	320
30	**15**	**300**	**220**	**55**	**19000**	**2000**	**980**

Step 4: Correlation Between X_1 and Y

$$r_{X_1Y} = \frac{5(2000) - (30)(300)}{\sqrt{[5(220) - 30^2][5(19000) - 300^2]}}$$

$$= \frac{10000 - 9000}{\sqrt{(1100 - 900)(95000 - 90000)}}$$

$$= \frac{1000}{\sqrt{200 \times 5000}}$$

$$\boxed{r_{X_1Y} = 1.000}$$

Step 5: Correlation Between X_2 and Y

$$r_{X_2Y} = \frac{5(980) - (15)(300)}{\sqrt{[5(55) - 15^2][5(19000) - 300^2]}}$$

$$= \frac{4900 - 4500}{\sqrt{(275 - 225)(95000 - 90000)}}$$

$$= \frac{400}{\sqrt{50 \times 5000}}$$

$$\boxed{r_{X_2Y} = 0.800}$$

Step 6: Feature Ranking

Rank	Feature	Pearson's r	Absolute correlation
1	Training Hours (X_1)	1.000	1.000
2	Experience (X_2)	0.800	0.800

Training Hours X1 is the most relevant feature based on Pearson's correlation, since it has a perfect positive linear correlation with employee performance (r=1).

	Experience X2 also shows a strong positive correlation (r=0.8), but it is comparatively lower. Pearson's correlation formula – 1 mark Calculation of required summations – 2 marks Correlation between X1 and Y – 2 marks Correlation between X2 and Y – 3 marks Feature ranking – 1 mark Identification of most relevant feature – 1 mark	
Q.5A	Stacking technique – Working and training procedure – 2 marks Stacking – Neat labelled diagram – 1 mark Blending technique – Working and training procedure – 2 marks Blending – Neat labelled diagram – 1 mark Differences in training procedures – 2 marks Role of the meta-model – 2 marks	
	OR	
Q.5B	Working of AdaBoost algorithm – 2 marks Suitable example – 2 marks Neat labelled diagram – 1 mark Sample weight updates with equations – 2 marks Weak learner importance (\(\alpha\)) calculation – 2 marks Effect on final ensemble performance – 1 mark	
Q.6A	**Given:** • Number of clusters: $K = 2$ • Initial medoids: $M_1 = A(1,1)$, $M_2 = D(7,7)$ • Distance measure: Manhattan distance **Step 1: Manhattan Distance Formula** $d(P,Q) = \|x_1 - x_2\| + \|y_1 - y_2\|$ **Step 2: Iteration 1** Initial medoids: $M_1 = A(1,1)$, $M_2 = D(7,7)$ Point | Distance from A | Distance from D | Cluster A(1,1) | 0 | 12 | C1 B(2,1) | 1 | 11 | C1 C(2,2) | 2 | 10 | C1 D(7,7) | 12 | 0 | C2 E(8,7) | 13 | 1 | C2 F(7,8) | 13 | 1 | C2 **Clusters formed:** $C_1 = \{A, B, C\}$ $C_2 = \{D, E, F\}$	

Step 3: Calculate New Medoids

For each cluster, calculate the sum of Manhattan distances from each candidate point to all other points in that cluster.

Cluster 1:

Candidate medoid	Total distance
A	0 + 1 + 2 = 3
B	1 + 0 + 1 = 2
C	2 + 1 + 0 = 3

Minimum distance = 2

New medoid: $M_1 = B(2,1)$

Cluster 2:

Candidate medoid	Total distance
D	0 + 1 + 1 = 2
E	1 + 0 + 2 = 3
F	1 + 2 + 0 = 3

Minimum distance = 2

New medoid: $M_2 = D(7,7)$

Medoids after Iteration 1:

$$\boxed{M_1 = B(2,1), \quad M_2 = D(7,7)}$$

Step 4: Iteration 2

Updated medoids: $M_1 = B(2,1), M_2 = D(7,7)$

Point	Distance from B	Distance from D	Cluster
A(1,1)	1	12	C1
B(2,1)	0	11	C1
C(2,2)	1	10	C1
D(7,7)	11	0	C2
E(8,7)	12	1	C2
F(7,8)	12	1	C2

Clusters formed:

$C_1 = \{A, B, C\}$

$C_2 = \{D, E, F\}$

Medoid cost calculations remain unchanged:

- Cluster 1: B has minimum cost = 2
- Cluster 2: D has minimum cost = 2

Therefore,

$$\boxed{M_1 = B(2,1), \quad M_2 = D(7,7)}$$

Since the medoids remain unchanged, the algorithm has converged.

Step 5: Final Result

Cluster	Data points	Final medoid	Clustering cost
C1	A, B, C	B(2,1)	2
C2	D, E, F	D(7,7)	2

Total clustering cost:

$$J = 2 + 2 = \boxed{4}$$

After two iterations, the algorithm converges with final medoids B(2,1) and D(7,7).

Manhattan distance formula – 1 mark
Iteration 1 – Distance calculations and cluster formation – 2 marks
Iteration 1 – Medoid cost calculations and update – 2 marks
Iteration 2 – Distance calculations and cluster formation – 2 marks
Iteration 2 – Medoid verification – 1 mark
Final medoids, clusters and total cost – 2 marks

OR

Q.6B

Given data points:

- A = (1, 1)
- B = (2, 1)
- C = (5, 4)
- D = (6, 5)

Distance measure: Euclidean distance

Linkage method: Complete linkage (maximum distance between points belonging to two clusters).

Step 1: Calculate Euclidean Distances

$$d(P,Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$d(A,B) = \sqrt{(2-1)^2 + (1-1)^2} = 1$$

$$d(A,C) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

$$d(A,D) = \sqrt{(6-1)^2 + (5-1)^2} = \sqrt{41} = 6.403$$

$$d(B,C) = \sqrt{(5-2)^2 + (4-1)^2} = \sqrt{18} = 4.243$$

$$d(B,D) = \sqrt{(6-2)^2 + (5-1)^2} = \sqrt{32} = 5.657$$

$$d(C,D) = \sqrt{(6-5)^2 + (5-4)^2} = \sqrt{2} = 1.414$$

Step 2: Initial Distance Matrix

	A	B	C	D
A	0	1.000	5.000	6.403
B	1.000	0	4.243	5.657
C	5.000	4.243	0	1.414
D	6.403	5.657	1.414	0

Step 3: First Merge

Minimum distance = 1.000, between A and B.

Merge A and B to form cluster .

Using complete linkage:

$$d(\{A,B\},C) = \max(5, 4.243) = 5$$

$$d(\{A,B\},D) = \max(6.403, 5.657) = 6.403$$

Updated Distance Matrix – After Merge 1

		C	D
	0	5.000	6.403
C	5.000	0	1.414
D	6.403	1.414	0

Step 4: Second Merge

Minimum distance = 1.414, between C and D.

Merge C and D to form cluster .

$$d(\{A,B\},\{C,D\})$$

$$= \max\{d(A,C), d(A,D), d(B,C), d(B,D)\}$$

$$= \max\{5, 6.403, 4.243, 5.657\}$$

$$= \boxed{6.403}$$

Updated Distance Matrix – After Merge 2

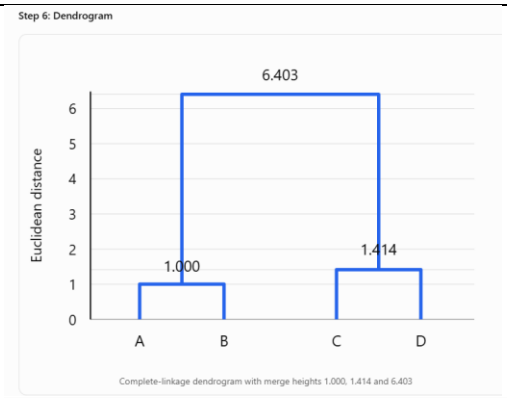
0	6.403
6.403	0

Step 5: Third Merge

Merge clusters and at distance 6.403.

Final cluster:

$$\boxed{\{A,B,C,D\}}$$



Step	Clusters merged	Distance
1	A and B	1.000
2	C and D	1.414
3	and	6.403

Agglomerative hierarchical clustering produces two intermediate clusters, and , which are finally merged at a complete-linkage distance of 6.403.

Euclidean distance formula and calculations – 2 marks
Initial distance matrix – 2 marks
First merge and updated matrix – 2 marks
Second merge and updated matrix – 2 marks
Final merge – 1 mark
Correct dendrogram – 1 mark

********* **All the Best**********