Junaid Aslam

20SW084 - II

DS-A Assignment 02

Submitted To: Mam Rafia Shaikh

Supervised Algorithms

Naive Bayes:

Naive Bayes is a probablistic machine learning algorithm widely used in various applications, particularly in text classification and spam email detection. It is based on Bayes Theorem, and makes a naive assumptions that features are independent of each other, simplifying calculations. Despite its simplicity, Naive bayes often yields impressive results in real world scenarios offering an efficient and easily interpretable method of classifying data points based on their probabilities and likelihoods, making it a valuable tool in the field of supervised learning.

Now, utilize K-fold cross validation (3-fold) should be applied.

→ **Fold 1:**

No.	Refund	Status	Tax	Class
1.	Yes	Single	125 (H)	Yes
2.	Yes	Single	80 (M)	No
3.	No	Single	80 (M)	No

A: Attributes

R: Refund
S: Status
T: Tax income

1→ $P(A \mid Yes) = P(R \mid Yes) \times P(S \mid Yes) \times P(T \mid Yes)$
$= 2/5 \times 3/5 \times 3/5 \times P(Yes)$
$= 18/125 \times 5/9 \Rightarrow 2/25 \approx 0.0$

$P(A \mid No) = P(R \mid No) \times P(S \mid No) \times P(T \mid No) \times P(No)$
$= 2/4 \times 3/4 \times 1/4 \times 4/9$
$= 1/24 \approx 0.041667$

2→ $P(A \mid Yes) = P(R \mid Yes) \times P(S \mid Yes) \times P(T \mid Yes) \times P(Yes)$
$= 2/5 \times 3/5 \times 2/5 \times 5/9$
$= 4/75 \approx 0.0533$

$P(A \mid No) = P(R \mid No) \times P(S \mid No) \times P(T \mid No) \times P(No)$
$= 2/4 \times 3/4 \times 2/4 \times 4/9$
$= 1/12 \approx 0.0833$

3→ $P(A \mid Yes) = P(R \mid Yes) \times P(S \mid Yes) \times P(T \mid Yes) \times P(Yes)$
$= 3/5 \times 3/5 \times 2/5 \times 5/9$
$= 2/25 \approx 0.08$

$P(A \mid No) = P(R \mid No) \times P(S \mid No) \times P(T \mid No) \times P(No)$
$= 2/4 \times 3/4 \times 2/4 \times 4/9$
$= 1/12 \approx 0.0833$

→ **Fold 2:**

Refund	Status	Tax	Class
No	Single	65(mid)	No
Yes	Single	60 (low)	No
No	Married	120(High)	Yes.

$P(A|Yes) = P(R|Yes) \times P(S|Yes) \times P(T|Yes) \times P(Yes)$

$= 3/5 \times 3/5 \times 2/5 \times 5/9$

$= 2/25 \cong 0.08$

$P(A|No) = 2/4 \times 3/4 \times 2/4 \times 4/9$

$= 1/12 \cong 0.0833$

$P(A|Yes) = P(R|Yes) \times P(S|Yes) \times P(T|Yes) \times P(Yes)$

$= 2/5 \times 3/5 \times 0/5 \times 5/9$

$= 0$

$P(A|No) = 2/4 \times 3/4 \times 1/4 \times 4/9$

$= 1/24 \cong 0.04166$

$P(A|Yes) = P(R|Yes) \times P(S|Yes) \times P(T|Yes) \times P(Yes)$

$= 2/5 \times 2/5 \times 3/5 \times 5/9$

$= 4/75 \cong 0.0533$

$P(A|No) = 2/4 \times 1/4 \times 1/4 \times 4/9$

$= 1/72 \cong 0.01388.$

Fold 3:

S.No	Refund	Status	Tax	Class
1.	No	Married	80 (H)	Yes
2.	Yes	Married	90 (H)	Yes
3.	No	Single	95 (H)	Yes

1→ $P(A|Yes) = P(R|Yes) * P(S|Yes) * P(T|Yes) * P(Yes)$

$= 3/5 \times 2/5 \times 2/\cancel{8} \times \cancel{8}/\cancel{9}\,3$

$= 4/75 \simeq 0.0533$

$P(A|No) = 2/4 \times 1/4 \times 2/\cancel{4} \times \cancel{4}/9$

$= 1/36 \simeq 0.0277$

2→ $P(A|Yes) = P(R|Yes) * P(S|Yes) * P(T|Yes) * P(Yes)$

$= 2/5 \times 2/5 \times \cancel{3}/\cancel{8} \times \cancel{8}/\cancel{9}\,3$

$= 4/75 \simeq 0.0533$

$P(A|No) = 2/4 \times 1/4 \times 1/\cancel{4} \times \cancel{4}/9$

$= 1/72 \simeq 0.01388$

3→ $P(A|Yes) = P(R|Yes) * P(S|Yes) * P(T|Yes) * P(Yes)$

$= 3/5 \times \cancel{3}/5 \times 3/\cancel{8} \times \cancel{8}/\cancel{9}$

$= 3/25 \simeq 0.12$

$P(A|No) = 2/4 \times \cancel{3}/4 \times 2/\cancel{4} \times \cancel{4}/\cancel{9}\,3$

$= 2/48 \simeq 0.041667$

Confusion Matrix:

Actual	Predicted
No	Yes
Yes	No
No	No
Yes	No
No	No
Yes	Yes
No	Yes
Yes	Yes
Yes	Yes

Predicted

Actual	Yes	No	Total
Yes	TP (3)	FN(2)	5
No	FP (2)	TN(2)	4
Total	5	4	9

Accuracy: $\frac{TP + TN}{total} = \frac{3+2}{9} = 0.05$

Error Rate: $\frac{FP + FN}{total} = \frac{2+2}{9} = 0.44$

Precision: $\frac{TP}{\text{Predicted Yes}} = \frac{3}{5} = 0.6$

Recall: $\frac{TP}{\text{Actual Yes}} = \frac{3}{5} = 0.6$

(iii) ## KNN Classifier:

The K-Nearest neighbour algorithm is a simple and effective supervised machine learning algorithm used for classif-ication and regression tasks. It operates on the principle of proximity, where an instance is classified based on the majority class of its K-nearest neighbours in features of space. KNN is non-parametric, making it suitable for various datasets and can be employed in multi-class classification. However, it can be sensitive to the choice of K and requires efficient data structures for optimal performance.

Now, utilize K-fold cross validation (3-fold mi

→ **Fold 01:**

S.No	Height	Weight	Class
1	5.35	49	underweight
2	5.25	55	underweight
3	5.91	75	normal

Now, calculate distance from test data to all training data.

Scanned with CamScanner

(1) $|5.35 - 5.09| + |49 - 50| = 1.26$
(2) $|5.35 - 5.41| + |49 - 62| = 13.06$
(3) $|5.35 - 5.64| + |49 - 68| = 19.29$
(4) $|5.35 - 5.18| + |49 - 48| = 1.17$
(5) $|5.35 - 6.07| + |49 - 80| = 31.72$
(6) $|5.35 - 4.92| + |49 - 45| = 4.43$

(1) $|5.25 - 5.09| + |55 - 50| = 5.16$
(2) $|5.25 - 5.41| + |55 - 62| = 7.16$
(3) $|5.25 - 5.64| + |55 - 68| = 13.39$
(4) $|5.25 - 5.18| + |55 - 48| = 7.07$
(5) $|5.25 - 6.07| + |55 - 80| = 25.82$
(6) $|5.25 - 4.92| + |55 - 45| = 10.33$

(1) $|5.91 - 5.09| + |75 - 50| = 25.82$
(2) $|5.91 - 5.41| + |75 - 62| = 13.5$
(3) $|5.91 - 5.64| + |75 - 68| = 7.27$
(4) $|5.91 - 5.18| + |75 - 48| = 27.73$
(5) $|5.91 - 6.07| + |75 - 80| = 5.16$
(6) $|5.91 - 4.92| + |75 - 45| = 30.99$

→ **Fold 2:**

S.no	Height	Weight	Class
(1)	5.09	50	Underweight
(2)	5.41	62	Normal
(3)	5.64	68	Normal

1. $|5.09 - 5.35| + |50 - 49| = 1.26$
2. $|5.09 - 5.25| + |50 - 55| = 5.16$
3. $|5.09 - 5.91| + |50 - 75| = 25.82$
4. $|5.09 - 5.18| + |50 - 48| = 2.09$
5. $|5.09 - 6.07| + |50 - 80| = 30.98$
6. $|5.09 - 4.92| + |50 - 45| = 5.17$

1. $|5.41 - 5.35| + |62 - 49| = 13.06$
2. $|5.41 - 5.25| + |62 - 55| = 7.16$
3. $|5.41 - 5.91| + |62 - 75| = 13.5$
4. $|5.41 - 5.18| + |62 - 48| = 14.23$
5. $|5.41 - 6.07| + |62 - 80| = 18.66$
6. $|5.41 - 4.92| + |62 - 45| = 13.49$

1. $|5.64 - 5.35| + |68 - 49| = 19.29$
2. $|5.64 - 5.25| + |68 - 55| = 13.39$
3. $|5.64 - 5.91| + |68 - 75| = 7.27$
4. $|5.64 - 5.18| + |68 - 48| = 16.46$
5. $|5.64 - 6.07| + |68 - 80| = 12.43$
6. $|5.64 - 4.92| + |68 - 45| = 23.72$

→ **Fold 3:**

S.no	Height	Weight	Class
1	5.18	48	Underweight
2	6.07	80	Normal
3	4.92	45	Underweight

1) $|5.18 - 5.35| + |48 - 49| = 1.17$

2) $|5.18 - 5.25| + |48 - 55| = 7.07$

3) $|5.18 - 5.91| + |48 - 75| = 27.78$

4) $|5.18 - 5.09| + |48 - 50| = 2.09$

5) $|5.18 - 5.41| + |48 - 62| = 14.23$

6) $|5.18 - 5.64| + |48 - 68| = 20.48$

1) $|6.07 - 5.35| + |80 - 49| = 31.72$

2) $|6.07 - 5.25| + |80 - 55| = 25.82$

3) $|6.07 - 5.91| + |80 - 75| = 5.16$

4) $|6.07 - 5.09| + |80 - 50| = 30.98$

5) $|6.07 - 5.41| + |80 - 62| = 18.66$

6) $|6.07 - 5.64| + |80 - 68| = 12.43$

1) $|4.92 - 5.35| + |45 - 49| = 4.43$

2) $|4.92 - 5.25| + |45 - 55| = 10.33$

3) $|4.92 - 5.91| + |45 - 75| = 30.99$

4) $|4.92 - 5.09| + |45 - 50| = 5.17$

5) $|4.92 - 5.41| + |45 - 62| = 17.49$

6) $|4.92 - 5.64| + |45 - 68| = 23.72$

→ **Confusion Matrix:**

No.	Actual	Predicted.
1)	Underweight	Underweight
2)	Normal	Underweight
3)	Normal	Normal
4)	Underweight	Underweight.

(5)	Normal	Normal
(6)	Normal	Normal
(7)	Underweight	Underweight
(8)	Normal	Normal
(9)	Underweight	Underweight

	Normal	Underweight	Total
Normal	TP(4)	FN (1)	5
Underweight	FP(0)	TN (4)	4
Total	4	5	9

Accuracy: $\frac{TP + TN}{Total} = \frac{4+4}{9} = 0.88 = 88\%$

Error Rate: $\frac{FP + FN}{total} = \frac{0+1}{9} = 0.11 = 11\%$

Precision: $\frac{TP}{\text{Predicted Normal}} = \frac{4}{4} = 1 = 100\%$

Recall: $\frac{TP}{\text{Predicted Normal}} = \frac{4}{5} = 0.8 = 80\%$.

(iii) **Decision Tree:**

A Decision Tree is a widely used machine learning algorithm that resembles a tree like structure. It is primarily employed for classification and regression tasks. Each node in the tree represents a decision or test on a features and the branches represents possible outcomes. Decision Trees are advantageous for their interoperability and the ability to handle both categorical and numerical data, making them a valuable tool for decision-making and predictive modeling in various domains.

→ Now, it's time for the implementation of algorithms. First, find entropy of all attributes.

① Refund:

→ When refund value is equal to ~~No~~ Yes, info

$[[2, 3]] = \text{entropy}(2/5, 3/5)$

$= -2/5 \log(2/5) - 3/5 \log(3/5) \Rightarrow 0.97$

→ When refund value is equal to No, info

$[[2, 2]] = \text{entropy}(2/4, 2/4)$

$= -2/4 \log(2/4) - 2/4 \log(2/4) \Rightarrow 1$

→ Expected information ~~gain~~ from attributes

$\text{info}(E(2,2))(E[2,3]) = 4/9 \times 1 + 5/9 \times 0.97$

$= 0.44 + 0.58 \Rightarrow 0.978$

② Status:

→ When status is equal to single:

$\text{info}[[3, 3]] = \text{entropy}(3/6, 3/6)$

$= -3/6 \log(3/6) - 3/6 \log(3/6) \Rightarrow 1$

→ When status is equal to married.

$\text{info}[[1, 2]] = \text{entropy}(1/3, 2/3)$

$= -1/3 \log(1/3) - 2/3 \log(2/3) = 0.921$

→ Expected info:

$\text{info}[[3, 3][1, 2]] = 3/9 \times 0.9207 + 6/9 \times 1$

$= 0.9735$

(3) Tax Income:

→ When tax is equal to high

info[[1,3]] = entropy (1/4, 3/4)

$= -1/4 \log(1/4) - 3/4 \log(3/4) \Rightarrow 0$

→ When tax is equal to medium

info[[2,2]] = entropy (2/4, 2/4)

$= -2/4 \log(2/4) - 2/4 \log(2/4) \Rightarrow 1$

→ When tax income is low

info[[1,0]] = entropy (1/1, 0/1)

$= -1 \log(1) - 0 \log(0) \Rightarrow 0$

→ Expected info:

info[[1,3][2,2][1,0]] = $4/9 \times 0.811 + 4/9 \times 1 + \frac{1}{9} \times 0$

$\Rightarrow 0.804$

4 Now compute the Information Gain.

info gain = (Info before split) − (Info after split)

→ info before split:

info[[4,5]] = entropy (4/9, 5/9)

$= -4/9 \log(4/9) - 5/9 \log(5/9) \Rightarrow 0.99$

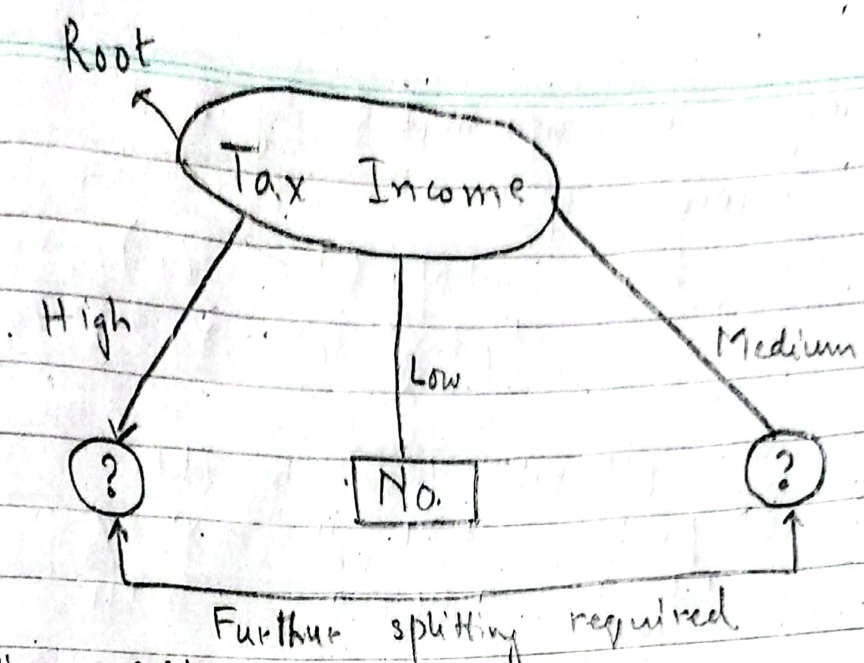
→ Info. gain for attributes from this data

Gain (Refund): 0.9922 − 0.978 = 0.0142

Gain (Status): 0.9922 − 0.9736 = 0.0186

Gain (Tax Income): 0.9922 − 0.804 = 0.1882

Tax Income has the highest information gain value, so, we start our decision tree from Tax Income.



* Continue splitting:

1) Refund:

→ When refund is equal to Yes

info [[1, 1]] = entropy (1/2 , 1/2)

$= -\frac{1}{2}\log(\frac{1}{2}) - \frac{1}{2}\log(\frac{1}{2}) \Rightarrow 1$

→ When refund is equal to no:

info [[2, 0]] = entropy (2/2 , 0/2)

$= -\frac{2}{2}\log(\frac{2}{2}) - \frac{0}{2}\log(\frac{0}{2}) = 0$

→ Expected info:

$\text{info}([1,1][2,0]) = \frac{2}{4} \times 1 + \frac{2}{4} \times 0 = 0.5$

2) Status:

→ When status is equal to single

info ([1, 1]) = entropy (1/2 , 1/2)

$= -\frac{1}{2}\log(\frac{1}{2}) - \frac{1}{2}\log(\frac{1}{2}) \Rightarrow 1$

→ When status is equal to married

info([2, 0]) = entropy (2/2 , 0/2)

$= -\frac{2}{2}\log(\frac{2}{2}) - \frac{0}{2}\log(\frac{0}{2}) \Rightarrow 0$

→ Expected info:

$\text{info}([1,1][2,0]) = \frac{2}{4} \times 1 + \frac{2}{4} \times 0 = 0.5$

* Computing the information gain.

→ info before split:

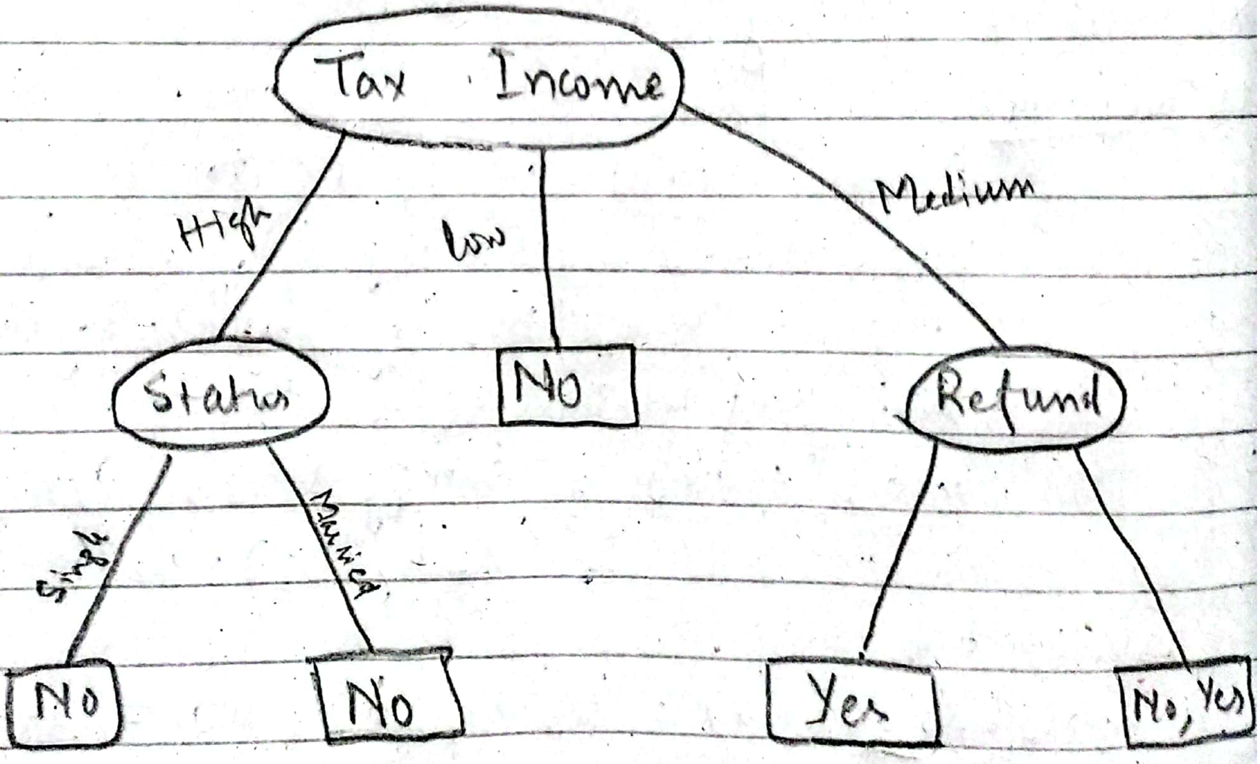
info([1, 3]) = entropy(1/4, 3/4)

$= -\frac{1}{4}\log(\frac{1}{4}) - \frac{3}{4}\log(\frac{3}{4}) \Rightarrow 0.811$

→ info gain for attributes from this data:

Gain(Refund): 0.811 − 0.5 = 0.311

Gain(Status): 0.811 − 0.5 = 0.311



Un-Supervised Algorithm ⑧

Question #01,

Solve

Step 1. Initial Seeds.

Cluster 1 : A_1 (2, 10)

Cluster 2 : A_4 (5, 8)

Cluster 3 : A_7 (1, 2)

Step 2: Euclidean Distance from Cluster to each point

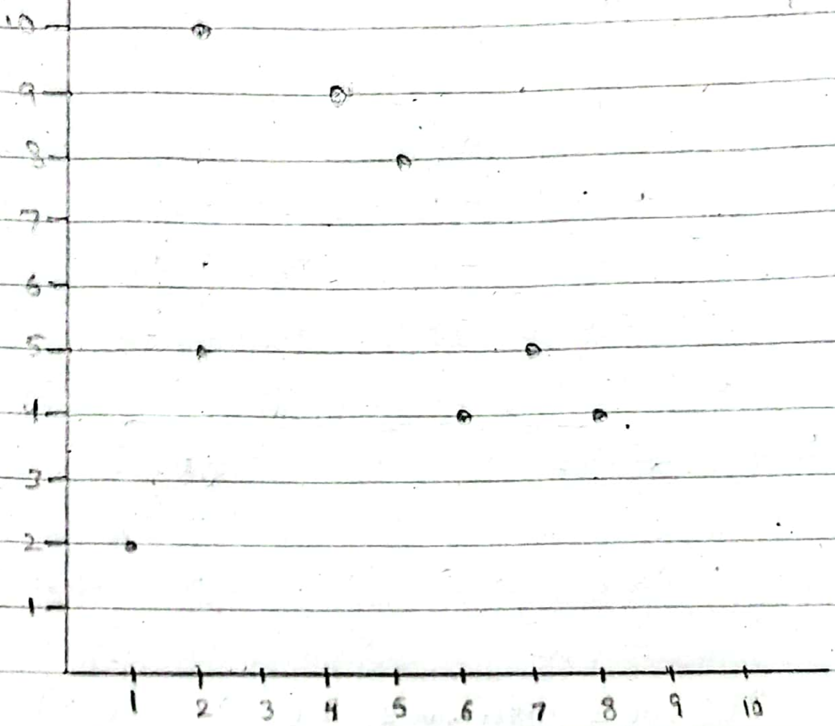
$$E = \sqrt{(x_2 - x_1)^2 + (Y_2 - y_1)^2}$$

Now, Assign each point to cluster with near centroid

Points	Attribute 1	Attribute 2	$D_1(2,10)$	$D_2(5,8)$	$D_3(1,2)$	Cluster
A_1	2	10	0	3.1623	8.062	1
A_2	2	5	5	4.242	3.162	3
A_3	8	4	8.485	5	7.280	2
A_4	5	8	3.605	0	7.211	2
A_5	7	5	7.07	3.605	6.708	2
A_6	6	4	7.211	4.123	5.385	2
A_7	1	2	8.0622	7.211	0	3
A_8	4	9	2.2360	1.414	7.615	2

Centroid after first iteration:

initial ←	Att: 1	Attr: 2	Att: 1	Att: 2 → after first iteration
Centroid 1	2	10	2	10
Centroid 2	5	8	6	6
Centroid 3	1	2	1.5	3.5



Cluster has been shifted by K-means but yet not stable, it may take more iterations for convergence depending on data and initial conditions to achieve consistent cluster assignments.

* Question # 02:

Solve

Step 01: Euclidean distance matrix between all pair

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_1	0.00							
A_2	5.00	0.00						
A_3	8.48	6.08	0.00					
A_4	3.60	8.48	5.00	0.00				
A_5	7.07	5.00	1.40	3.60	0.00			
A_6	7.21	4.12	2.00	4.12	1.41	0.00		
A_7	8.06	3.16	7.28	7.21	6.70	5.38	0.00	
A_8	2.23	4.47	6.40	1.41	5.00	5.38	7.61	0.00

Single Linkage:

Iteration 1	A_1	A_2	$A_3 A_5$	A_4	A_6	A_7	A_8
A_1	0.00						
A_2	5.00	0.00					
$A_3 A_5$	7.07	5.00	0.00				
A_4	3.60	8.48	3.60	0.00			
A_6	7.21	4.12	1.41	4.12	0.00		
A_7	8.06	3.16	6.70	7.21	5.38	0.00.	
A_8	2.23	4.47	5.00	1.41	5.38	7.61	0.00

Iteration 2	A_1	A_2	$A_3 A_5$	$A_4 A_8$	A_6	A_7
A_1	0.0					
A_2	5.00	0.0				
$A_3 A_5$	7.07	5.00	0.0			
$A_4 A_8$	2.23	4.47	3.60	0.0		
A_6	7.21	4.12	1.41	4.12	0.0	
A_7	8.06	3.16	6.70	7.21	5.38	0.0

Iteration 3	A_1	A_2	$A_3 A_5 A_6$	$A_4 A_8$	A_7
A_1	0.0				
A_2	5.00	0.0			
$A_3 A_5 A_6$	7.07	4.12	0.0		
$A_4 A_8$	2.23	4.47	3.60	0.0	
A_7	8.06	3.16	5.38	7.21	0.0

Iteration 4	$A_1 A_4 A_8$	A_2	$A_3 A_5 A_6$	A_7
$A_1 A_4 A_8$	0.0			
A_2	4.47	0.0		
$A_3 A_5 A_6$	3.60	4.12	0.0	
A_7	7.21	3.16	5.38	0.0

Iteration 5	$A_1 A_4 A_8$	$A_2 A_7$	$A_3 A_5 A_6$
$A_1 A_4 A_8$	0.0		
$A_2 A_7$	4.47	0.0	
$A_3 A_5 A_6$	3.60	4.12	0.0

Iteration 6	$A_1 A_3 A_4 A_5 A_6 A_8$	$A_2 A_7$
$A_1 A_3 A_4 A_5 A_6 A_8$	0.0	
$A_2 A_7$	4.12	0.0

Final cluster is merged as:
$A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8$, which contains all original 8 objects

→ Complete Linkage:

Iteration 1	A_1	A_2	$A_3 A_5$	A_4	A_6	A_7	A_8
A_1	0.0						
A_2	5.00	0.0					
$A_3 A_5$	8.48	6.08	0.0				
A_4	3.60	8.48	5.00	0.0			
A_6	7.21	4.12	2.00	4.12	0.0		
A_7	8.06	3.16	7.28	7.21	5.38	0.0	
A_8	2.23	4.47	6.40	1.41	5.38	7.61	0.0

Iteration 2	A_1	A_2	$A_3 A_5$	$A_4 A_8$	A_6	A_7
A_1	0.0					
A_2	5.00	0.0				
$A_3 A_5$	8.48	6.08	0.0			
$A_4 A_8$	3.60	8.48	6.40	0.0		
A_6	7.21	4.12	7.28	5.38	0.0	
A_7	8.06	3.16	6.40	7.61	5.38	0.0

Iteration 3	A_1	$A_2 A_7$	$A_3 A_5$	$A_4 A_8$	A_6
A_1	0.0				
$A_2 A_7$	8.06	0.0			
$A_3 A_5$	8.48	6.40	0.0		
$A_4 A_8$	3.60	8.48	6.40	0.0	
A_6	7.21	5.38	7.28	5.38	0.0

Iteration 4	$A_1 A_4 A_8$	$A_2 A_7$	$A_3 A_5$	A_6
$A_1 A_4 A_8$	0.0			
$A_2 A_7$	8.48	0.0		
$A_3 A_5$	8.48	6.40	0.0	
A_6	7.21	5.38	7.28	0.0

Iteration 5	$A_1 A_4 A_8$	$A_2 A_7 A_6$	$A_3 A_5$
$A_1 A_4 A_8$	0.0		
$A_2 A_7 A_6$	8.48	0.0	
$A_3 A_5$	8.48	7.28	0.0

Iteration 6	$A_1 A_4 A_8$	$A_2 A_7 A_6 A_3 A_5$
$A_1 A_4 A_8$	0.0	
$A_2 A_7 A_6 A_3 A_5$	8.48	0.0

Final Stage cluster $A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8$ are merged cluster from result, contains all eight (8) objects -

→ **Group Average:**

Iteration 1	A_1	A_2	$A_3 A_5$	A_4	A_6	A_7	A_8
A_1	0.0						
A_2	5.00	0.0					
$A_3 A_5$	7.77	5.54	0.0				
A_4	3.60	8.48	4.30	0.0			
A_6	7.21	4.12	1.70	4.12	0.0		
A_7	8.06	3.16	6.99	7.21	5.38	0.0	
A_8	2.23	4.4	5.70	1.41	5.38	7.61	0.0

Iteration 2	A_1	A_2	$A_3 A_5$	$A_4 A_8$	A_6	A_7
A_1	0.0					
A_2	5.0	0.0				
$A_3 A_5$	7.77	5.54	0.0			
$A_4 A_8$	2.91	6.47	5.00	0.0		
A_6	7.21	4.12	7.70	4.75	0.0	
A_7	8.06	3.16	6.99	7.41	5.38	0.0

Iteration 3	A_1	A_2	$A_3 A_5 A_6$	$A_4 A_8$	A_7
A_1	0.0				
A_2	5.00	0.0			
$A_3 A_5 A_6$	7.49	4.83	0.0		
$A_4 A_8$	2.91	6.47	3.225	0.0	
A_7	8.06	3.96	6.18	7.41	0.0

Iteration 4	$A_1 A_4 A_8$	A_2	$A_3 A_5 A_6$	A_7
$A_1 A_4 A_8$	0.0			
A_2	5.73	0.0		
$A_3 A_5 A_6$	5.35	4.83	0.0	
A_7	7.70	3.16	6.18	0.0

Iteration 5	$A_1 A_4 A_8$	$A_2 A_7$	$A_4 A_5 A_6$
$A_1 A_4 A_8$	0.0		
$A_2 A_7$	6.71	0.0	
$A_3 A_5 A_6$	5.35	5.50	0.0

Iteration 6	$A_1 A_4 A_8 A_3 A_5 A_6$	$A_2 A_7$
$A_1 A_4 A_8 A_3 A_5 A_6$	0.0	
$A_2 A_7$	6.10	0.0

Distance Between Centroids:

Iteration 1	A_1	A_2	$A_3 A_5$	A_4	A_6	A_7	A_8
A_1	0.0						
A_2	5.00	0.0					
$A_3 A_5$	1.41	1.08	0.0				
A_4	3.60	8.48	1.40	0.0			
A_6	7.21	4.12	0.59	4.12	0.0		
A_7	8.06	3.16	0.58	7.21	5.38	0.0	
A_8	2.23	4.47	1.40	1.41	5.38	7.61	0.0

Iteration 2	A_1	A_2	$A_3A_5A_7$	A_4	A_6	A_8
A_1	0.0					
A_2	5.00	0.0				
$A_3A_5A_7$	6.65	2.08	0.0			
A_4	3.60	8.48	6.81	0.0		
A_6	7.21	4.12	4.79	4.12	0.0	
A_8	2.23	4.47	6.21	1.41	5.38	0.0

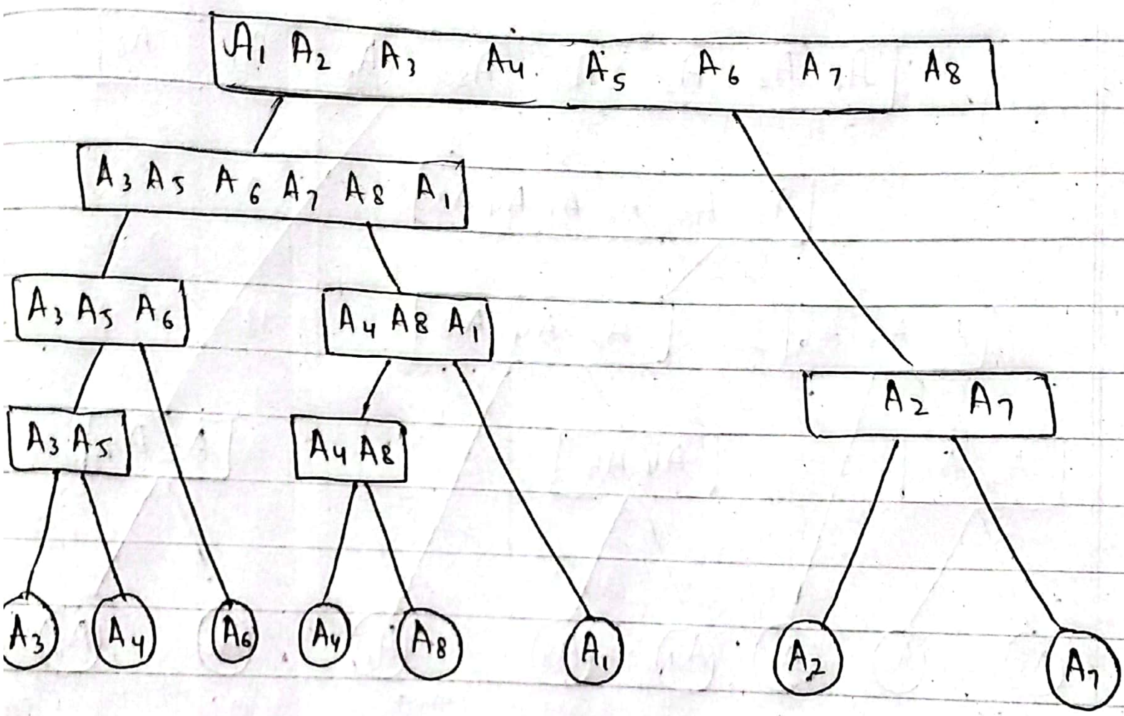
Iteration 3	A_1	A_2	$A_3A_5A_7$	A_4A_8	A_6
A_1	0.0				
A_2	5.00	0.0			
$A_3A_5A_7$	6.65	2.08	0.0		
A_4A_8	1.37	4.01	0.40	0.0	
A_6	7.21	4.12	4.79	1.26	0.0

Iteration 4	A_1	A_2	$A_3A_4A_5A_7A_8$	A_6
A_1	0.0			
A_2	5.00	0.0		
$A_3A_4A_5A_7A_8$	5.28	1.93	0.0	
A_6	7.21	4.12	3.53	0.0

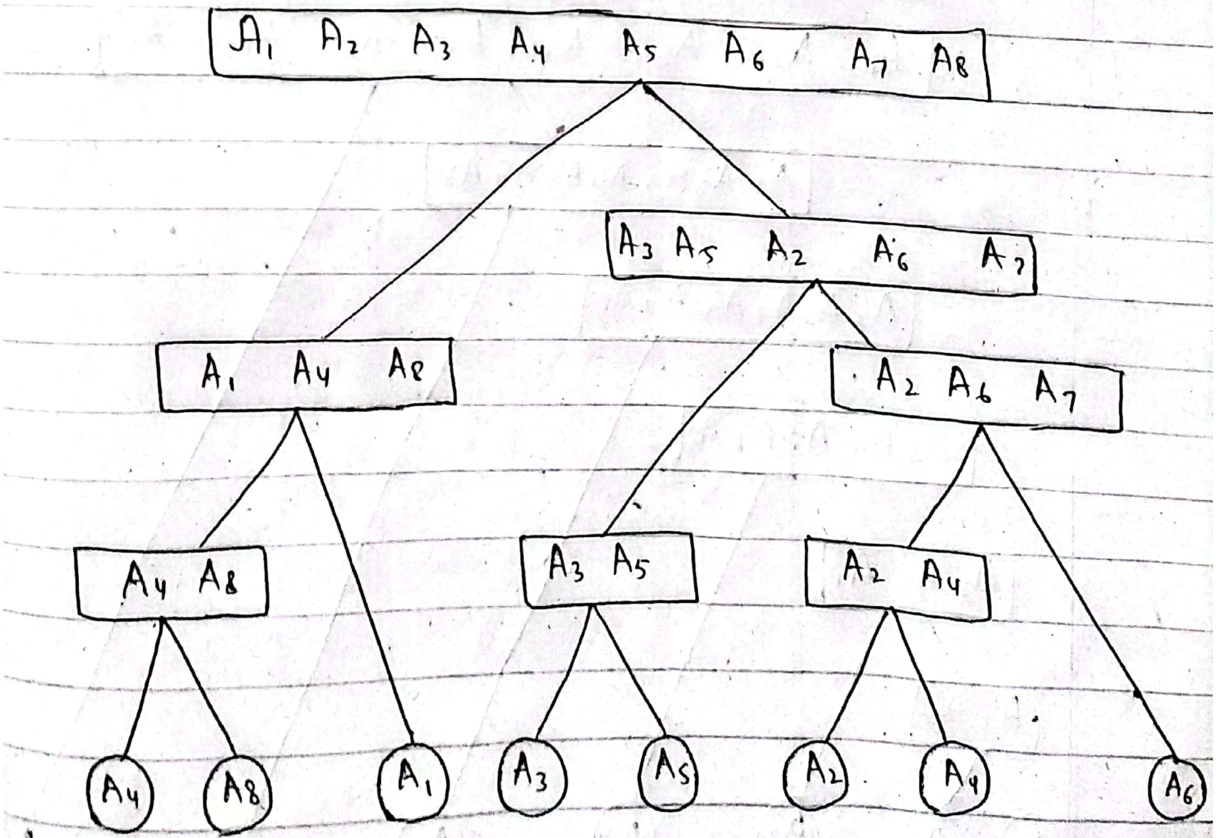
Iteration 5	A_1	$A_2A_3A_4A_5A_7A_8$	A_6
A_1	0.0		
$A_2A_3A_4A_5A_7A_8$	0.28	0.0	
A_6	7.21	0.59	0.0

Iteration 6	$A_1A_2A_3A_4A_5A_8A_7$	A_6
$A_1A_2A_3A_4A_5A_7A_8$	0.0	
A_6	6.62	0.0

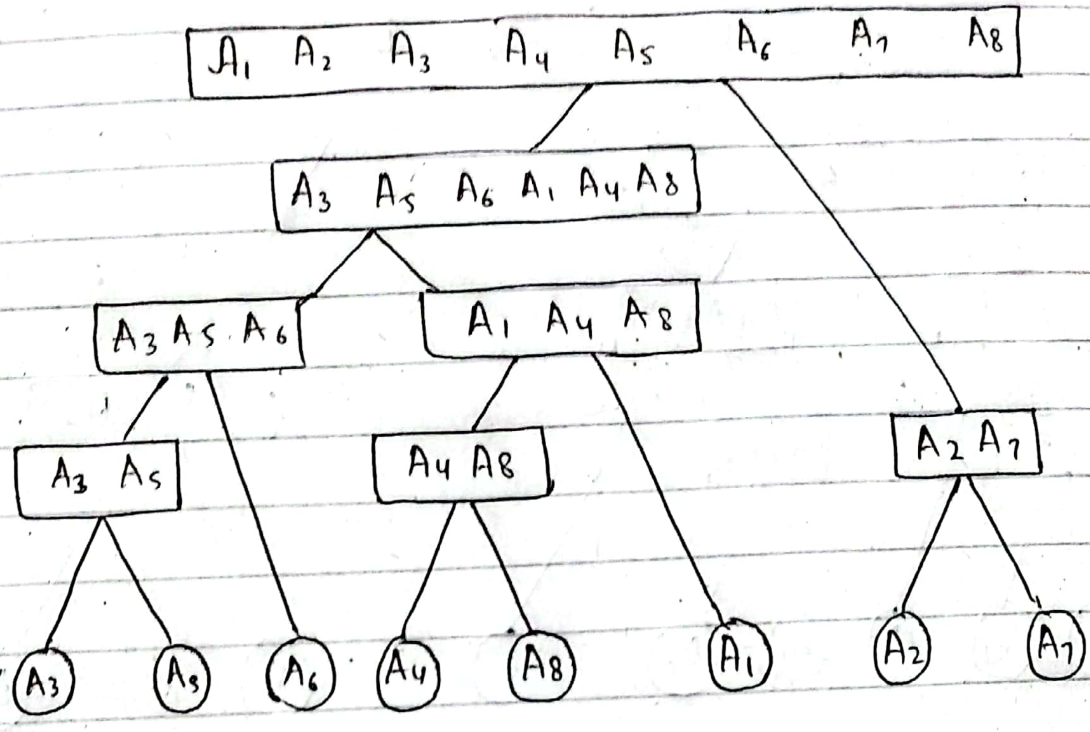
Single Linkage:



Complete Linkage:



→ Group Average:



→ Distance Between Centroids: