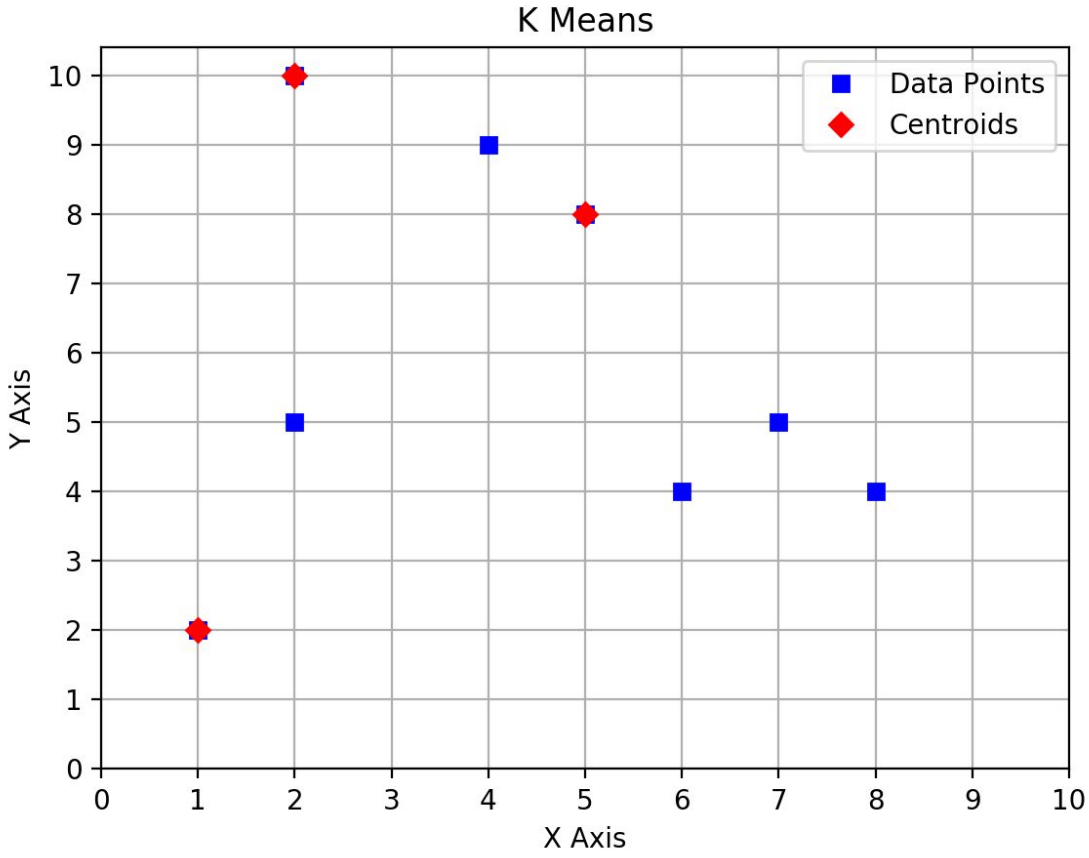
1. K-Means

Given 8 dots: A (4, 9) B (2, 10) C (1, 2) D (2, 5) E (6, 4) F (8, 4) G (7, 5) H (5, 8)

3 centroids: α (2, 10) β (1, 2) γ (5, 8)

1)



Compute the Euclidian distance

A -> α = $\sqrt{5}$ B -> α = 0 C -> α = $\sqrt{65}$ D -> α = 5

A -> β = $\sqrt{58}$ B -> β = $\sqrt{65}$ C -> β = 0 D -> β = $\sqrt{10}$

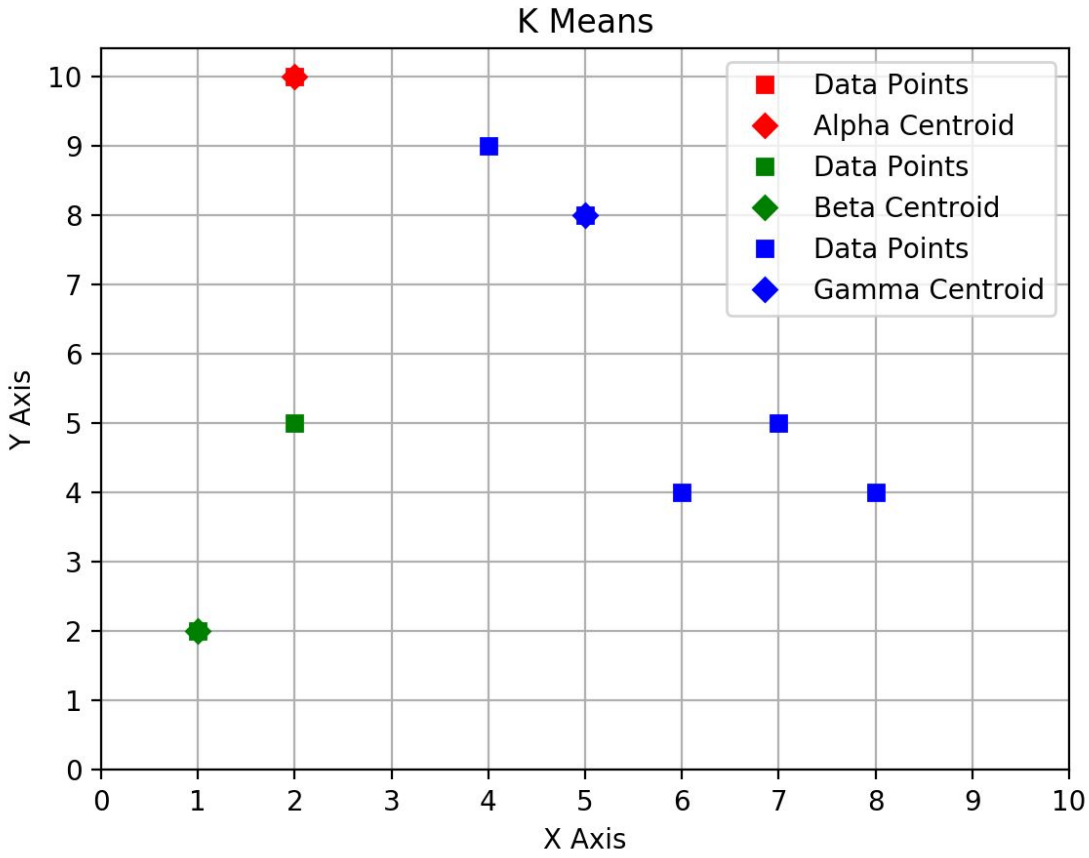
A -> γ = $\sqrt{2}$ B -> γ = $\sqrt{13}$ C -> γ = $\sqrt{52}$ D -> γ = $\sqrt{18}$

E -> α = $\sqrt{52}$ F -> α = $\sqrt{72}$ G -> α = $\sqrt{50}$ H -> α = $\sqrt{13}$

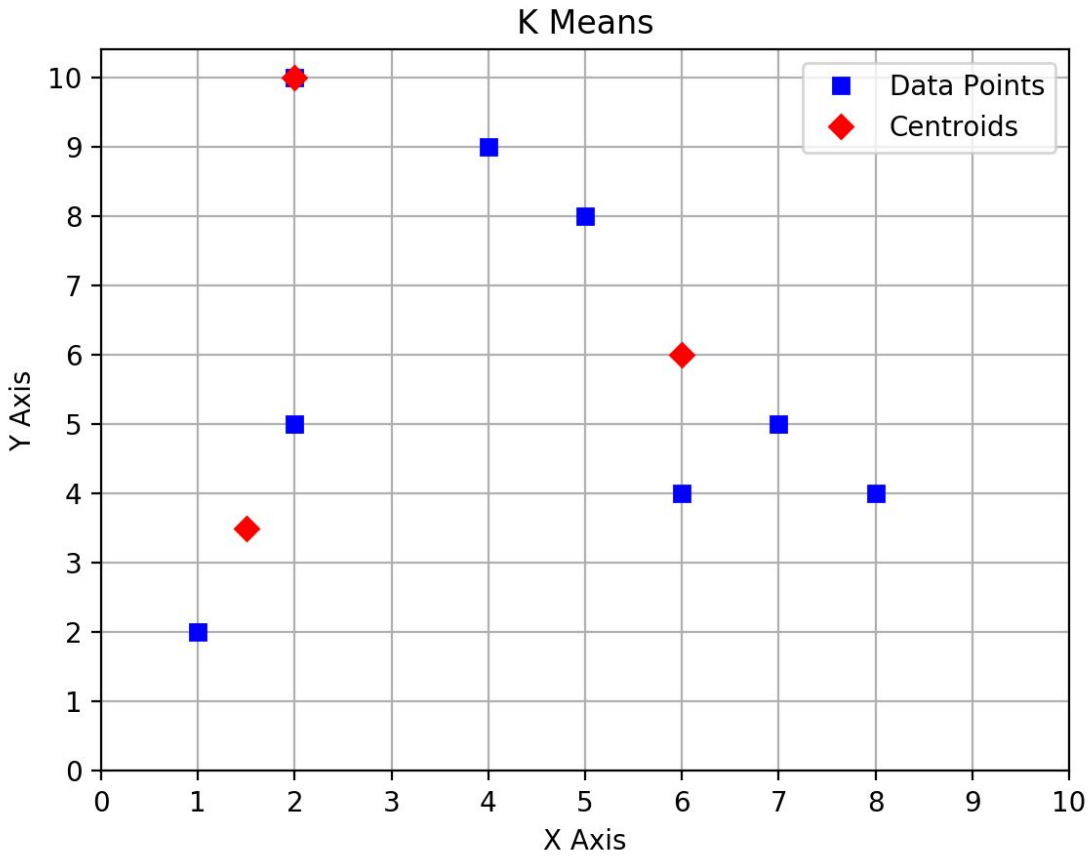
E -> β = $\sqrt{29}$ F -> β = $\sqrt{53}$ G -> β = $\sqrt{45}$ H -> β = $\sqrt{52}$

E -> γ = $\sqrt{17}$ F -> γ = 5 G -> γ = $\sqrt{13}$ H -> γ = 0

Centroid Assignment



2) Recompute centroids for each cluster
Given 8 dots: A (4, 9) B (2, 10) C (1, 2) D (2, 5) E (6, 4) F (8, 4) G (7, 5) H (5, 8)
New centroids: α (2, 10) β (1.5, 3.5) γ (6, 6)



Compute the Euclidian distance

A -> α = $\sqrt{5}$ B -> α = 0 C -> α = $\sqrt{65}$ D -> α = 5

A -> β = $\sqrt{36.5}$ B -> β = $\sqrt{42.5}$ C -> β = $\sqrt{2.5}$ D -> β = $\sqrt{2.5}$

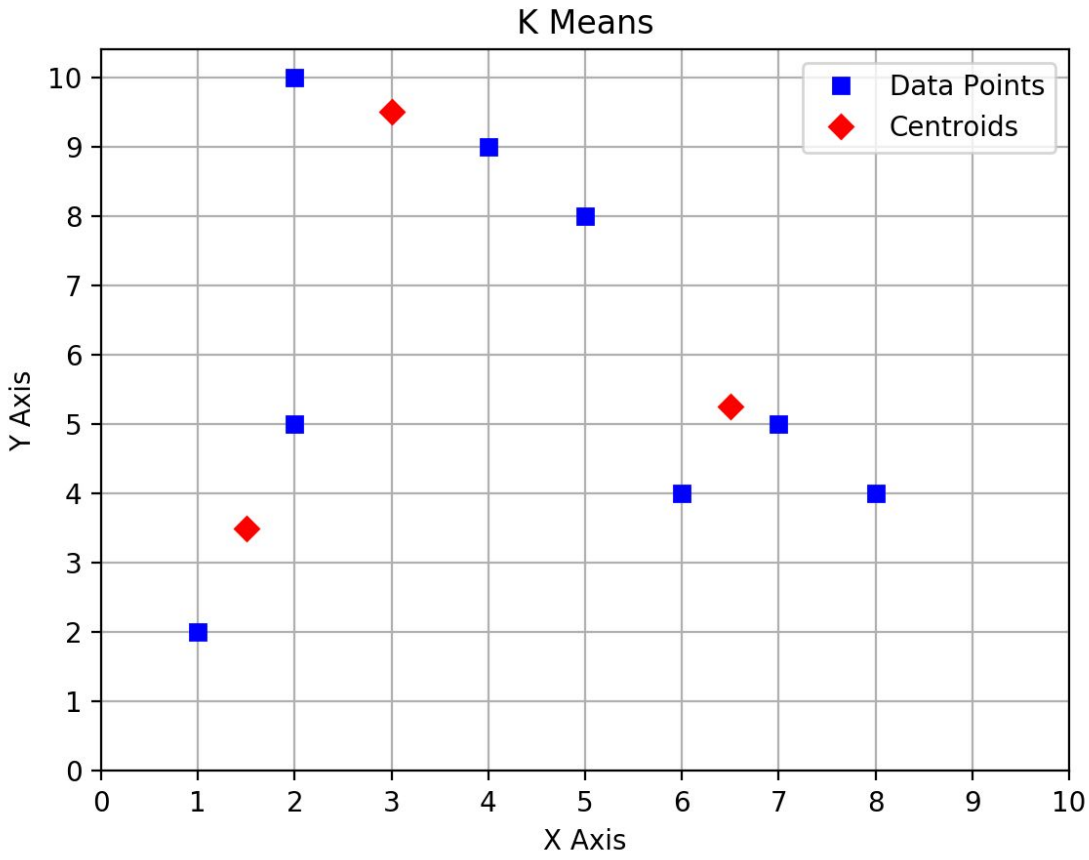
A -> γ = $\sqrt{13}$ B -> γ = $\sqrt{32}$ C -> γ = $\sqrt{41}$ D -> γ = $\sqrt{17}$

E -> α = $\sqrt{52}$ F -> α = $\sqrt{128}$ G -> α = $\sqrt{50}$ H -> α = $\sqrt{13}$

E -> β = $\sqrt{20.5}$ F -> β = $\sqrt{42.5}$ G -> β = $\sqrt{32.5}$ H -> β = $\sqrt{32.5}$

E -> γ = 2 F -> γ = $\sqrt{8}$ G -> γ = $\sqrt{2}$ H -> γ = $\sqrt{5}$

3) Recompute centroids for each cluster
Given 8 dots: A (4, 9) B (2, 10) C (1, 2) D (2, 5) E (6, 4) F (8, 4) G (7, 5) H (5, 8)
New centroids: α (3, 9.5) β (1.5, 3.5) γ (6.5, 5.25)



Compute the Euclidian distance

A -> α = $\sqrt{1.25}$ B -> α = $\sqrt{1.25}$ C -> α = $\sqrt{60.25}$ D -> α = $\sqrt{21.25}$

A -> β = $\sqrt{36.5}$ B -> β = $\sqrt{42.5}$ C -> β = $\sqrt{2.5}$ D -> β = $\sqrt{2.5}$

A -> γ = $\sqrt{20.3125}$ B -> γ = $\sqrt{42.8125}$ C -> γ = $\sqrt{40.8125}$ D -> γ = $\sqrt{20.3125}$

E -> α = $\sqrt{39.25}$ F -> α = $\sqrt{55.25}$ G -> α = $\sqrt{36.25}$ H -> α = $\sqrt{6.25}$

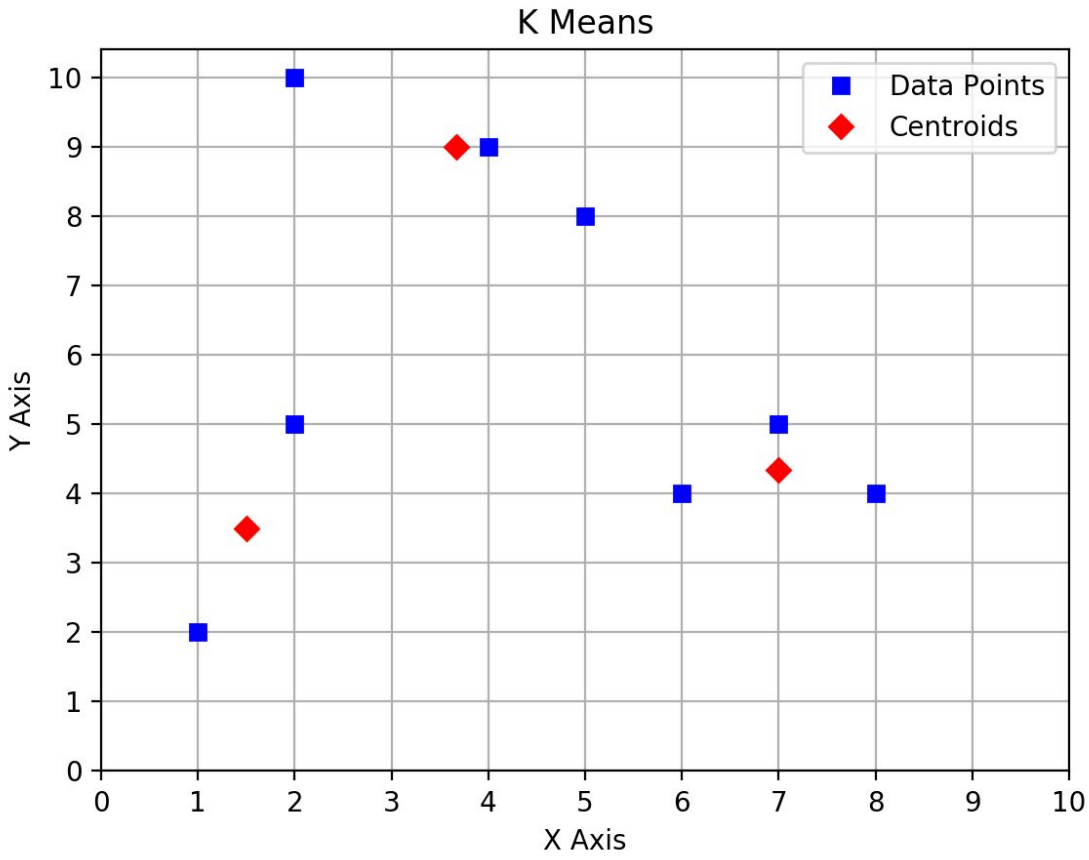
E -> β = $\sqrt{20.5}$ F -> β = $\sqrt{42.5}$ G -> β = $\sqrt{32.5}$ H -> β = $\sqrt{32.5}$

E -> γ = $\sqrt{1.8125}$ F -> γ = $\sqrt{3.8125}$ G -> γ = $\sqrt{0.3125}$ H -> γ = $\sqrt{9.8125}$

4) Recompute centroids for each cluster

Given 8 dots: A (4, 9) B (2, 10) C (1, 2) D (2, 5) E (6, 4) F (8, 4) G (7, 5) H (5, 8)

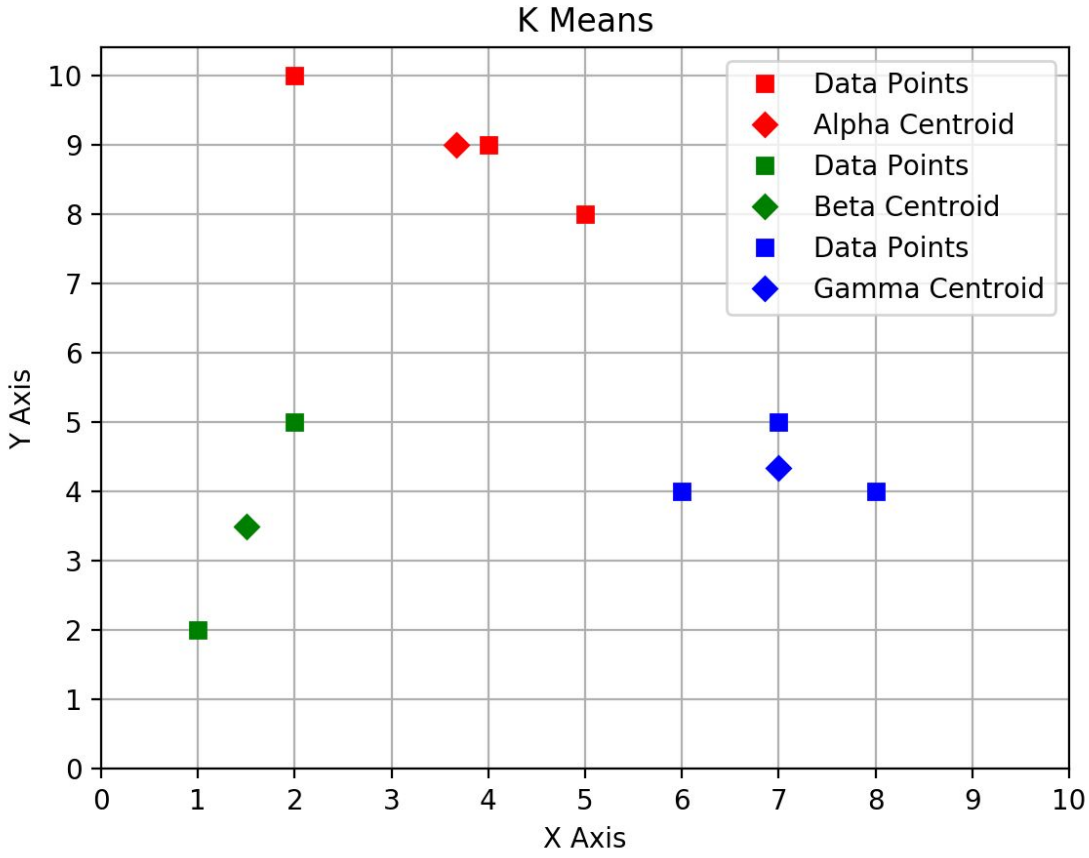
New centroids: α (11/3, 9) β (1.5, 3.5) γ (7, 13/3)



Compute the Euclidian distance

A -> α = 1/3	B -> α = 1.94	C -> α = 7.49	D -> α = 4.33
A -> β = $\sqrt{36.5}$	B -> β = $\sqrt{42.5}$	C -> β = $\sqrt{2.5}$	D -> β = $\sqrt{2.5}$
A -> γ = 5.55	B -> γ = 7.56	C -> γ = 6.44	D -> γ = 5.04
E -> α = 5.52	F -> α = 6.62	G -> α = 5.21	H -> α = 1.67
E -> β = $\sqrt{20.5}$	F -> β = $\sqrt{42.5}$	G -> β = $\sqrt{32.5}$	H -> β = $\sqrt{32.5}$
E -> γ = 1.05	F -> γ = 1.05	G -> γ = 0.67	H -> γ = 4.18

Realizing that the centroids are not changing this time. K-Means converges. The plot is drawn as below.



2. Agglomerative Hierarchical
MIN Link:

	x	y
p1	0.40	0.53
p2	0.21	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1						
p2	0.242074					
p3	0.21587	0.152315				
p4	0.367696	0.196469	0.158114			
p5	0.34176	0.133417	0.284605	0.284253		
p6	0.235372	0.252982	0.10198	0.219545	0.386005	

First Cluster: {3, 6}

d({1},{3,6}) = min(d({1},{3}), d({1},{6})) = d({1},{3})
d({2},{3,6}) = min(d({2},{3}), d({2},{6})) = d({2},{3})
d({4},{3,6}) = min(d({4},{3}), d({4},{6})) = d({4},{3})
d({5},{3,6}) = min(d({5},{3}), d({5},{6})) = d({5},{3})

Merge two closest clusters for MIN:

	x	y
p1	0.40	0.53
p2	0.21	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1						
p2	0.242074					
p3	0.21587	0.152315				
p4	0.367696	0.196469	0.158114			
p5	0.34176	0.133417	0.284605	0.284253		
p6	~~0.235372~~	~~0.252982~~	~~0.10198~~	~~0.219545~~	~~0.386005~~	

Second Cluster: {2, 5}

d({1},{2,5}) = min(d({1},{2}), d({1},{5})) = d({1},{2})
d({3},{2,5}) = min(d({2},{3}), d({3},{5})) = d({2},{3})
d({4},{2,5}) = min(d({4},{2}), d({4},{5})) = d({4},{2})

Merge two closest clusters for MIN:

	x	y
p1	0.40	0.53
p2	0.21	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1						
p2	0.242074					
p3	0.21587	0.152315				
p4	0.367696	0.196469	0.158114			
p5	~~0.34176~~	~~0.133417~~	~~0.284605~~	~~0.284253~~		
p6	~~0.235372~~	~~0.252982~~	~~0.10198~~	~~0.219545~~	~~0.386005~~	

Third Cluster: {2, 3}

d({1},{2, 3}) = min(d({1},{2}), d({1},{3})) = d({1},{3})
d({4},{2, 3}) = min(d({2},{4}), d({3},{4})) = d({3},{4})

Merge two closest clusters for MIN:

	x	y
p1	0.40	0.53
p2	0.21	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1						
p2	~~0.242074~~					
p3	0.21587	~~0.152315~~				
p4	0.367696	~~0.196469~~	0.158114			
p5	~~0.34176~~	~~0.133417~~	0.284605	~~0.284253~~		
p6	~~0.235372~~	~~0.252982~~	~~0.10198~~	~~0.219545~~	~~0.386005~~	

Fourth Cluster: {3, 4}

Therefore, {1} and {2, 3, 4, 5, 6}

MAX Link:

	x	y
p1	0.40	0.53
p2	0.21	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1						
p2	0.242074					
p3	0.21587	0.152315				
p4	0.367696	0.196469	0.158114			
p5	0.34176	0.133417	0.284605	0.284253		
p6	0.235372	0.252982	0.10198	0.219545	0.386005	

Merge two closest clusters for MAX:

	x	y
p1	0.40	0.53
p2	0.21	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1						
p2	0.242074					
p3	~~0.21587~~	~~0.152315~~				
p4	0.367696	0.196469	~~0.158114~~			
p5	0.34176	0.133417	~~0.284605~~	0.284253		
p6	0.235372	0.252982	~~0.10198~~	0.219545	0.386005	

First Cluster: {3, 6}

d({1},{3,6}) = max(d({1},{3}), d({1},{6})) = d({1},{6})
d({2},{3,6}) = max(d({2},{3}), d({2},{6})) = d({2},{6})
d({4},{3,6}) = max(d({4},{3}), d({4},{6})) = d({4},{6})
d({5},{3,6}) = max(d({5},{3}), d({5},{6})) = d({5},{6})

Merge two closest clusters for MAX:

	x	y
p1	0.40	0.53
p2	0.21	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1						
p2	~~0.242074~~					
p3	~~0.21587~~	~~0.152315~~				
p4	0.367696	~~0.196469~~	~~0.158114~~			
p5	0.34176	~~0.133417~~	~~0.284605~~	0.284253		
p6	0.235372	~~0.252982~~	~~0.10198~~	0.219545	0.386005	

Second Cluster: {2, 5}

d({1},{2,5}) = max(d({1},{2}), d({1},{5})) = d({1},{5})
d({4},{2,5}) = max(d({2},{4}), d({4},{5})) = d({4},{5})
d({6},{2,5}) = max(d({2},{6}), d({5},{6})) = d({5},{6})

Merge two closest clusters for MAX:

	x	y
p1	0.40	0.53
p2	0.21	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1						
p2	~~0.242074~~					
p3	~~0.21587~~	~~0.152315~~				
p4	0.367696	~~0.196469~~	~~0.158114~~			
p5	0.34176	~~0.133417~~	~~0.284605~~	~~0.284253~~		
p6	~~0.235372~~	~~0.252982~~	~~0.10198~~	~~0.219545~~	0.386005	

Third Cluster: {4, 6}

d({1},{4,6}) = max(d({1},{4}), d({1},{6})) = d({1},{4})
d({5},{4,6}) = max(d({4},{5}), d({5},{6})) = d({5},{6})

Merge two closest clusters for MAX:

	x	y
p1	0.40	0.53
p2	0.21	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1						
p2	~~0.242074~~					
p3	~~0.21587~~	~~0.152315~~				
p4	0.367696	~~0.196469~~	~~0.158114~~			
p5	0.34176	~~0.133417~~	~~0.284605~~	~~0.284253~~		
p6	~~0.235372~~	~~0.252982~~	~~0.10198~~	~~0.219545~~	0.386005	

Fourth Cluster: {1, 5}

Therefore, {1, 2, 5} and {3, 4, 6}

AVG:

	x	y
p1	0.40	0.53
p2	0.21	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1						
p2	0.242074					
p3	0.21587	0.152315				
p4	0.367696	0.196469	0.158114			
p5	0.34176	0.133417	0.284605	0.284253		
p6	0.235372	0.252982	**0.10198**	0.219545	0.386005	

First Cluster: {3, 6}

	p1	p2	p4	p5	p3,p6
p2	0.242074				
p4	0.367696	0.196469			
p5	0.34176	**0.133417**	0.284253		
p3, p6	0.225621	0.2026485	0.1888295	0.335305	

Second Cluster: {2,5}

	p1	p4	p2, p5	p3, p6
p4	0.367696			
p2, p5	0.291917	0.240361		
p3, p6	0.225621	**0.1888295**	0.26897675	

Third Cluster: {3,4,6}

Therefore, {1}, {2, 3, 4, 5, 6}

3.
Condition: (ε = 7.5, MinPts = 3).
pt 0: 2 < MinPts, so cluster=-1
pt 1: 3 ≥ MinPts, so cluster=0, to visit=[40, 75], visited={1}
- pt 40: cluster=0, 3 ≥ MinPts, so adding neighbors to_visit=[75, 28], visited={1, 40}
- pt 75: cluster=0, 3 ≥ MinPts, so adding neighbors to_visit=[28, 4], visited={1, 40, 75}
- pt 28: cluster=0, 3 ≥ MinPts, so adding neighbors to_visit=[4, 12], visited={1, 28, 40, 75}
- pt 4: cluster=0, 3 ≥ MinPts, so adding neighbors to_visit=[12, 56], visited={1, 4, 28, 40, 75}
- pt 12: cluster=0, 2 < MinPts, to_visit=[56], visited={1, 4, 12, 28, 40, 75}
- pt 56: cluster=0, 3 ≥ MinPts, so adding neighbors to_visit=[66], visited={1, 4, 12, 28, 40, 56, 75}
- pt 66: cluster=0, 2 < MinPts to_visit=[], visited={1, 4, 12, 28, 40, 56, 66, 75}

pt 2: 1 < MinPts, so cluster=-1
pt 3: 2 < MinPts, so cluster=-1
pt 4: cluster=0, so skip
pt 5: 3 ≥ MinPts, so cluster=1 to visit=[70, 74], visited={5}:
- pt 70: cluster=1, 5 ≥ MinPts, so adding neighbors to_visit=[74, 32, 69, 72], visited={5, 70}
- pt 74: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[32, 69, 72, 19, 54], visited={5, 70, 74}
- pt 32: cluster=1, 5 ≥ MinPts, so adding neighbors to_visit=[69, 72, 19, 54, 63, 69], visited={5, 32, 70, 74}
- pt 69: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[72, 19, 54, 63], visited={5, 32, 69, 70, 74}
- pt 72: cluster=1, 7 ≥ MinPts, so adding neighbors to_visit=[19, 54, 63, 8, 60], visited={5, 32, 69, 70, 72, 74}
- pt 19: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[54, 63, 8, 60], visited={5, 19, 32, 69, 70, 72, 74}
- pt 54: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[63, 8, 60, 25], visited={5, 19, 32, 54, 69, 70, 72, 74}
- pt 63: cluster=1, 7 ≥ MinPts, so adding neighbors to_visit=[8, 60, 25], visited={5, 19, 32, 54, 63, 69, 70, 72, 74}
- pt 8: cluster=1, 5 ≥ MinPts, so adding neighbors to_visit=[60, 25, 11], visited={5, 8, 19, 32, 54, 63, 69, 70, 72, 74}
- pt 60: cluster=1, 6 ≥ MinPts, so adding neighbors to_visit=[25, 11, 50, 68], visited={5, 8, 19, 32, 54, 60, 63, 69, 70, 72, 74}
- pt 25: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[11, 50, 68, 26, 67], visited={5, 8, 19, 25, 32, 54, 60, 63, 69, 70, 72, 74}
- pt 11: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[50, 68, 26, 67, 14], visited={5, 8, 11, 19, 25, 32, 54, 60, 63, 69, 70, 72, 74}
- pt 50: cluster=1, 5 ≥ MinPts, so adding neighbors to_visit=[68, 26, 67, 14, 39], visited={5, 8, 11, 19, 25, 32, 50, 54, 60, 63, 69, 70, 72, 74}

• pt 68: cluster=1, 5 ≥ MinPts, so adding neighbors to_visit=[26, 67, 14, 39], visited={5, 8, 11, 19, 25, 32, 50, 54, 60, 63, 68, 69, 70, 72, 74}

• pt 26: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[67, 14, 39, 34], visited={5, 8, 11, 19, 25, 26, 32, 50, 54, 60, 63, 68, 69, 70, 72, 74}

• pt 67: cluster=1, 2 < MinPts, to_visit=[14, 39, 34], visited={5, 8, 11, 19, 25, 26, 32, 50, 54, 60, 63, 67, 68, 69, 70, 72, 74}

• pt 14: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[39, 34, 6], visited={5, 8, 11, 14, 19, 25, 26, 32, 50, 54, 60, 63, 68, 69, 70, 72, 74}

• pt 39: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[34, 6, 10, 71], visited={5, 8, 11, 14, 19, 25, 26, 32, 39, 50, 54, 60, 63, 68, 69, 70, 72, 74}

• pt 34: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[6, 10, 71, 29, 46], visited={5, 8, 11, 14, 19, 25, 26, 32, 34, 39, 50, 54, 60, 63, 68, 69, 70, 72, 74}

• pt 6: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[10, 71, 29, 46, 42], visited={5, 6, 8, 11, 14, 19, 25, 26, 32, 34, 39, 50, 54, 60, 63, 68, 69, 70, 72, 74}

• pt 10: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[71, 29, 46, 42, 22], visited={5, 6, 8, 10, 11, 14, 19, 25, 26, 32, 34, 39, 50, 54, 60, 63, 68, 69, 70, 72, 74}

• pt 71: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[29, 46, 42, 22], visited={5, 6, 8, 10, 11, 14, 19, 25, 26, 32, 34, 39, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 29: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[46, 42, 22, 16], visited={5, 6, 8, 10, 11, 14, 19, 25, 26, 29, 32, 34, 39, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 46: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[42, 22, 16], visited={5, 6, 8, 10, 11, 14, 19, 25, 26, 29, 32, 34, 39, 46, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 42: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[22, 16, 17, 20], visited={5, 6, 8, 10, 11, 14, 19, 25, 26, 29, 32, 34, 39, 42, 46, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 22: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[16, 17, 20], visited={5, 6, 8, 10, 11, 14, 19, 22, 25, 26, 29, 32, 34, 39, 42, 46, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 16: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[17, 20, 48], visited={5, 6, 8, 10, 11, 14, 16, 19, 22, 25, 26, 29, 32, 34, 39, 42, 46, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 17: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[20, 48], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 22, 25, 26, 29, 32, 34, 39, 42, 46, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 20: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[48, 38], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 32, 34, 39, 42, 46, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 48: cluster=1, 2 < MinPts, to_visit=[38], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 32, 34, 39, 42, 46, 48, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 38: cluster=1, 5 ≥ MinPts, so adding neighbors to_visit=[30, 37, 45], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 32, 34, 38, 39, 42, 46, 48, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 30: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[37, 45, 52], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 30, 32, 34, 38, 39, 42, 46, 48, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 37: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[45, 52, 53], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 30, 32, 34, 37, 38, 39, 42, 46, 48, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 45: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[52, 53], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 30, 32, 34, 37, 38, 39, 42, 45, 46, 48, 50, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 52: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[53, 49, 64], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 30, 32, 34, 37, 38, 39, 42, 45, 46, 48, 50, 52, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 53: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[49, 64, 47], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 30, 32, 34, 37, 38, 39, 42, 45, 46, 48, 50, 52, 53, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 49: cluster=1, 4 ≥ MinPts, so adding neighbors to_visit=[64, 47, 31, 76], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 30, 32, 34, 37, 38, 39, 42, 45, 46, 48, 49, 50, 52, 53, 54, 60, 63, 68, 69, 70, 71, 72, 74}

• pt 64: cluster=1, 2 < MinPts, to_visit=[47, 31, 76], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 30, 32, 34, 37, 38, 39, 42, 45, 46, 48, 49, 50, 52, 53, 54, 60, 63, 64, 68, 69, 70, 71, 72, 74}

• pt 47: cluster=1, 2 < MinPts, to_visit=[31, 76], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 30, 32, 34, 37, 38, 39, 42, 45, 46, 47, 48, 49, 50, 52, 53, 54, 60, 63, 64, 68, 69, 70, 71, 72, 74}

• pt 31: cluster=1, 2 < MinPts, to_visit=[76], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 30, 31, 32, 34, 37, 38, 39, 42, 45, 46, 47, 48, 49, 50, 52, 53, 54, 60, 63, 64, 68, 69, 70, 71, 72, 74}

• pt 76: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[21], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 29, 30, 31, 32, 34, 37, 38, 39, 42, 45, 46, 47, 48, 49, 50, 52, 53, 54, 60, 63, 64, 68, 69, 70, 71, 72, 74}

• pt 21: cluster=1, 3 ≥ MinPts, so adding neighbors to_visit=[], visited={5, 6, 8, 10, 11, 14, 16, 17, 19, 20, 21, 22, 25, 26, 29, 30, 31, 32, 34, 37, 38, 39, 42, 45, 46, 47, 48, 49, 50, 52, 53, 54, 60, 63, 64, 68, 69, 70, 71, 72, 74}

pt 5: cluster=1, so skip

pt 6: cluster=1, so skip

pt 7: 1 < MinPts, so cluster=-1

pt 8: cluster=1, so skip

pt 9: 3 ≥ MinPts, so cluster=2, to visit=[33, 78], visited={9}

• pt 33: cluster=2, 3 ≥ MinPts, so adding neighbors to_visit=[78], visited={9, 33}

• pt 78: cluster=2, 3 ≥ MinPts, so adding neighbors to_visit=[], visited={9, 33, 78}

pt 10: cluster=1, so skip

pt 11: cluster=1, so skip

pt 12: cluster=0, so skip

pt 13: 2 < MinPts, so cluster=-1

pt 14: cluster=1, so skip

pt 15: 1 < MinPts, so cluster=-1

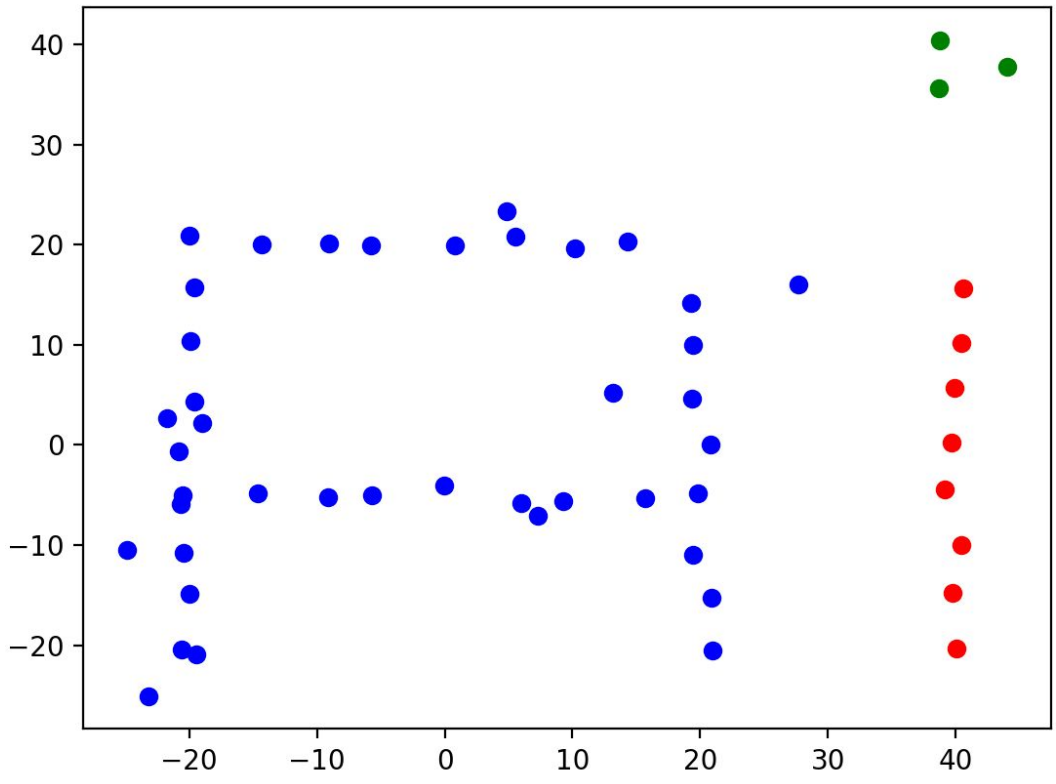
pt 16: cluster=1, so skip

pt 17: cluster=1, so skip

pt 18: 1 < MinPts, so cluster=-1

pt 19: cluster=1, so skip
pt 20: cluster=1, so skip
pt 21: cluster=1, so skip
pt 22: cluster=1, so skip
pt 23: 1 < MinPts, so cluster=-1
pt 24: 1 < MinPts, so cluster=-1
pt 25: cluster=1, so skip
pt 26: cluster=1, so skip
pt 27: 2 < MinPts, so cluster=-1
pt 28: cluster=0, so skip
pt 29: cluster=1, so skip
pt 30: cluster=1, so skip
pt 31: cluster=1, so skip
pt 32: cluster=1, so skip
pt 33: cluster=2, so skip
pt 34: cluster=1, so skip
pt 35: 2 < MinPts, so cluster=-1
pt 36: 1 < MinPts, so cluster=-1
pt 37: cluster=1, so skip
pt 38: cluster=1, so skip
pt 39: cluster=1, so skip
pt 40: cluster=0, so skip
pt 41: 1 < MinPts, so cluster=-1
pt 42: cluster=1, so skip
pt 43: 2 < MinPts, so cluster=-1
pt 44: 1 < MinPts, so cluster=-1
pt 45: cluster=1, so skip
pt 46: cluster=1, so skip
pt 47: cluster=1, so skip
pt 48: cluster=1, so skip
pt 49: cluster=1, so skip
pt 50: cluster=1, so skip
pt 51: 2 < MinPts, so cluster=-1
pt 52: cluster=1, so skip
pt 53: cluster=1, so skip
pt 54: cluster=1, so skip
pt 55: 2 < MinPts, so cluster=-1
pt 56: cluster=0, so skip
pt 57: 1 < MinPts, so cluster=-1
pt 58: 1 < MinPts, so cluster=-1
pt 59: 2 < MinPts, so cluster=-1
pt 60: cluster=1, so skip
pt 61: 1 < MinPts, so cluster=-1

pt 62: 2 < MinPts, so cluster=-1
pt 63: cluster=1, so skip
pt 64: cluster=1, so skip
pt 65: 1 < MinPts, so cluster=-1
pt 66: cluster=0, so skip
pt 67: cluster=1, so skip
pt 68: cluster=1, so skip
pt 69: cluster=1, so skip
pt 70: cluster=1, so skip
pt 71: cluster=1, so skip
pt 72: cluster=1, so skip
pt 73: 1 < MinPts, so cluster=-1
pt 74: cluster=1, so skip
pt 75: cluster=0, so skip
pt 76: cluster=1, so skip
pt 77: 2 < MinPts, so cluster=-1
pt 78: cluster=2, so skip
pt 79: 1 < MinPts, so cluster=-1



Extra credit:

Name: Thomas G. Dietterich
Employer: Oregon State University

3 interesting facts:
Born in South Weymouth MA
Known for Executive Editor of Machine Learning (92-98), founder of field machine learning
Honored Distinguished Professor by Oregon State in spring of 2013