Title:

DBSCAN

Presented by:

Suhani Chandel 500108261

Vibhuti Bighana 500108277

Anisha 500107738

Content

1. History
2. Why DBSCAN
3. Density Based Clustering
4. Hyperparameters
5. Core Points,Border Points,Noise Points
6. Density Connectivity
7. Steps in Algorithm
8. Limitations

1. History

1996

Proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu.

- Published in KDD (Knowledge Discovery and Data Mining).

Motivation

- Traditional clustering (like K-Means) struggled with:
- Arbitrary cluster shapes.
- Noise/outliers.
- Needed a density-based approach for real-world spatial data.

2. Why DBSCAN

Following are some difficulties faced in k means clustering:

1. Requires Predefined Number of Clusters

2. Sensitive to Outliers

3.Poor Performance on Non-Spherical or Irregular Shaped Clusters

3.Density Based Clustering

We cluster our data with the help of density between the data points.A density-based clustering algorithm groups together points that are closely packed (i.e., have many nearby neighbors), and marks points that lie alone in low-density regions as outliers or noise.

4. DBSCAN Hyperparameters

ε (epsilon)

The maximum distance between two points for them to be considered as neighbors.

MinPts

- This is the minimum number of points required within the **eps** radius to form a dense region.

5.*How Does DBSCAN Work?*

DBSCAN works by categorizing data points into three types:

1. **Core Point:** A point is a core point if it has at least MinPts points (including itself) within a distance ε.

Condition: Points within ε ≥ MinPts

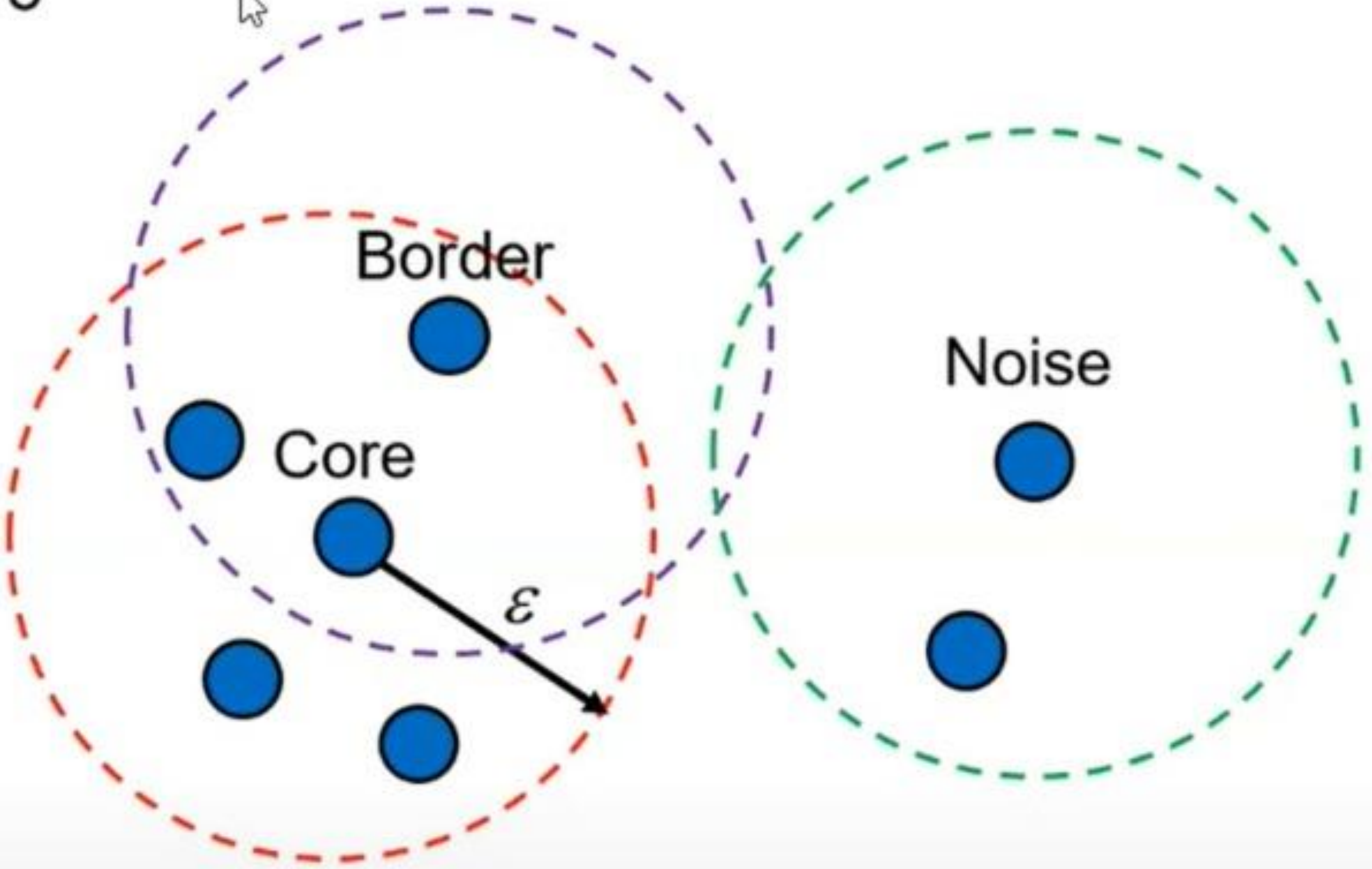
2. Border Point: A point is a border point if it has fewer than MinPts points within ε, but it lies within the ε distance of a core point.

Condition: Points within ε < MinPts and within ε of a Core Point

3. Noise Point: A point is a noise point (outlier) if it is neither a core point nor a border point.

Condition: Not a Core Point and Not within ε of any Core Point

MinPts = 5



6.*Density Connectivity*

Density connectivity

Two points p and q are density-connected if there exists a point o such that both p and q are density-reachable from o.Density connectivity is the basis for forming clusters in DBSCAN. All points in a cluster are mutually density-connected, and if a point is density-connected to any point in the cluster, it also belongs to that cluster.

7.Steps in DBSCAN ALGORITHM

1.Identify Core Points:
For each point in the dataset, count the number of points within its eps neighborhood. If the count meets or exceeds MinPts, mark the point as a core point.

2.Form Clusters:
For each core point that is not already assigned to a cluster, create a new cluster. Recursively find all density-connected points (points within the eps radius of the core point) and add them to the cluster.

3.Density Connectivity:
Two points, a and b, are density-connected if there exists a chain of points where each point is within the eps radius of the next, and at least one point in the chain is a core point. This chaining process ensures that all points in a cluster are connected through a series of dense regions.

4.Label Noise Points:
After processing all points, any point that does not belong to a cluster is labeled as noise.

NUMERICAL

- Apply the DBSCAN algorithm to the given data points and
- Create the clusters with
- minPts = 4 and
- epsilon (ε) = 1.9.

Data Points:

P1: (3, 7)	P2: (4, 6)
P3: (5, 5)	P4: (6, 4)
P5: (7, 3)	P6: (6, 2)
P7: (7, 2)	P8: (8, 4)
P9: (3, 3)	P10: (2, 6)
P11: (3, 5)	P12: (2, 4)

DBSCAN Clustering Algorithm Solved Example – 1

- Use Eucladian distance and calculate the distance between each points.

$$\boldsymbol{Distance(A(x_1, y_1), B(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

minPts = 4 and epsilon (ε) = 1.9

P1: (3, 7)

P2: (4, 6)

P3: (5, 5)

P4: (6, 4)

P5: (7, 3)

P6: (6, 2)

P7: (7, 2)

P8: (8, 4)

P9: (3, 3)

P10: (2, 6)

P11: (3, 5)

P12: (2, 4)

DBSCAN Clustering Algorithm Solved Example – 1

P1: (3, 7)
P2: (4, 6)
P3: (5, 5)
P4: (6, 4)
P5: (7, 3)
P6: (6, 2)
P7: (7, 2)
P8: (8, 4)
P9: (3, 3)
P10: (2, 6)
P11: (3, 5)
P12: (2, 4)

minPts = 4 and epsilon (ε) = 1.9

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0											
P2	1.41	0										
P3	2.83	1.41	0									
P4	4.24	2.83	1.41	0								
P5	5.66	4.24	2.83	1.41	0							
P6	5.83	4.47	3.16	2.00	1.41	0						
P7	6.40	5.00	3.61	2.24	1.00	1.00	0					
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0				
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0			
P10	1.41	2.00	3.16	4.47	5.83	5.66	6.40	6.32	3.16	0		
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0	
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0

minPts = 4 and epsilon (ε) = 1.9

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0											
P2	1.41	0										
P3	2.83	1.41	0									
P4	4.24	2.83	1.41	0								
P5	5.66	4.24	2.83	1.41	0							
P6	5.83	4.47	3.16	2.00	1.41	0						
P7	6.40	5.00	3.61	2.24	1.00	1.00	0					
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0				
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0			
P10	1.41	2.00	3.16	4.47	5.83	5.66	6.40	6.32	3.16	0		
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0	
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

P8: P5

P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

P8: P5

P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11

minPts = 4 and epsilon (ε) = 1.9

Point	Status	
P1	Noise	
P2	Core	
P3	Noise	
P4	Noise	
P5	Core	
P6	Noise	
P7	Noise	
P8	Noise	
P9	Noise	
P10	Noise	
P11	Core	
P12	Noise	

DBSCAN Clustering Algorithm Solved Example – 1

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

P8: P5

P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11

minPts = 4 and epsilon (ε) = 1.9

Point	Status	
P1	Noise	Border
P2	Core	
P3	Noise	Border
P4	Noise	Border
P5	Core	
P6	Noise	Border
P7	Noise	Border
P8	Noise	Border
P9	Noise	
P10	Noise	Border
P11	Core	
P12	Noise	Border

minPts = 4 and epsilon (ε) = 1.9

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

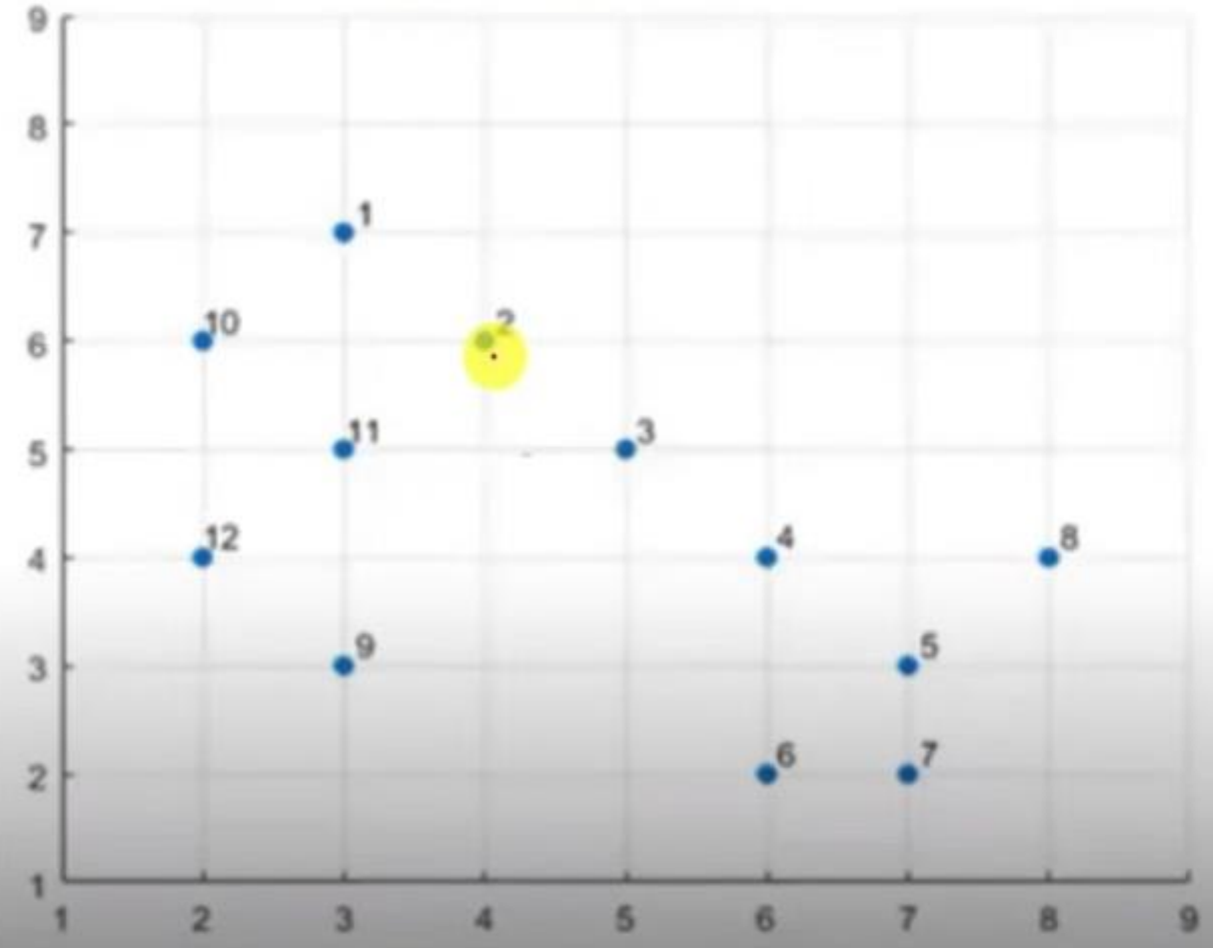
P8: P5

P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11



minPts = 4 and epsilon (ε) = 1.9

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

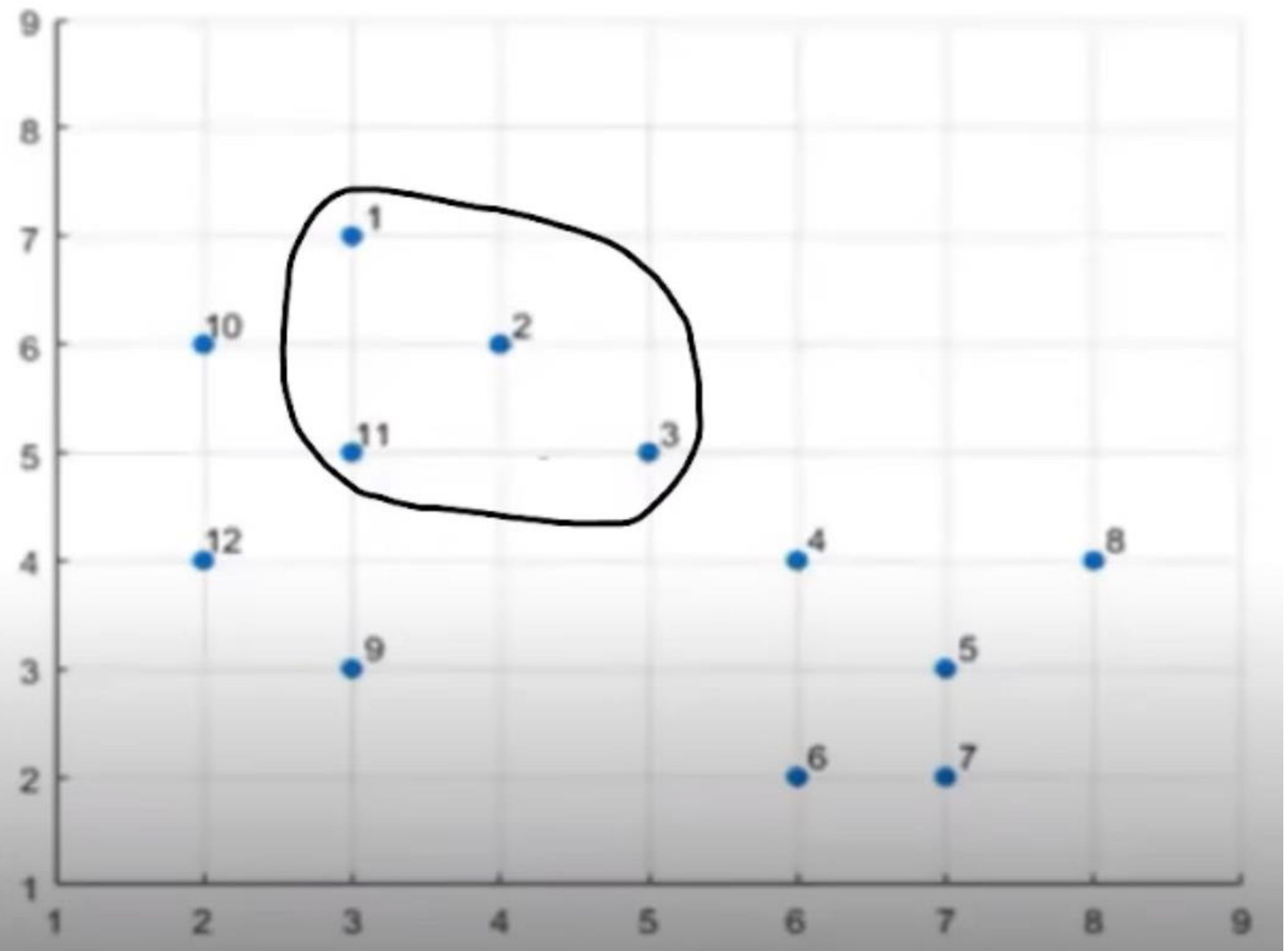
P8: P5

P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11



P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

P8: P5

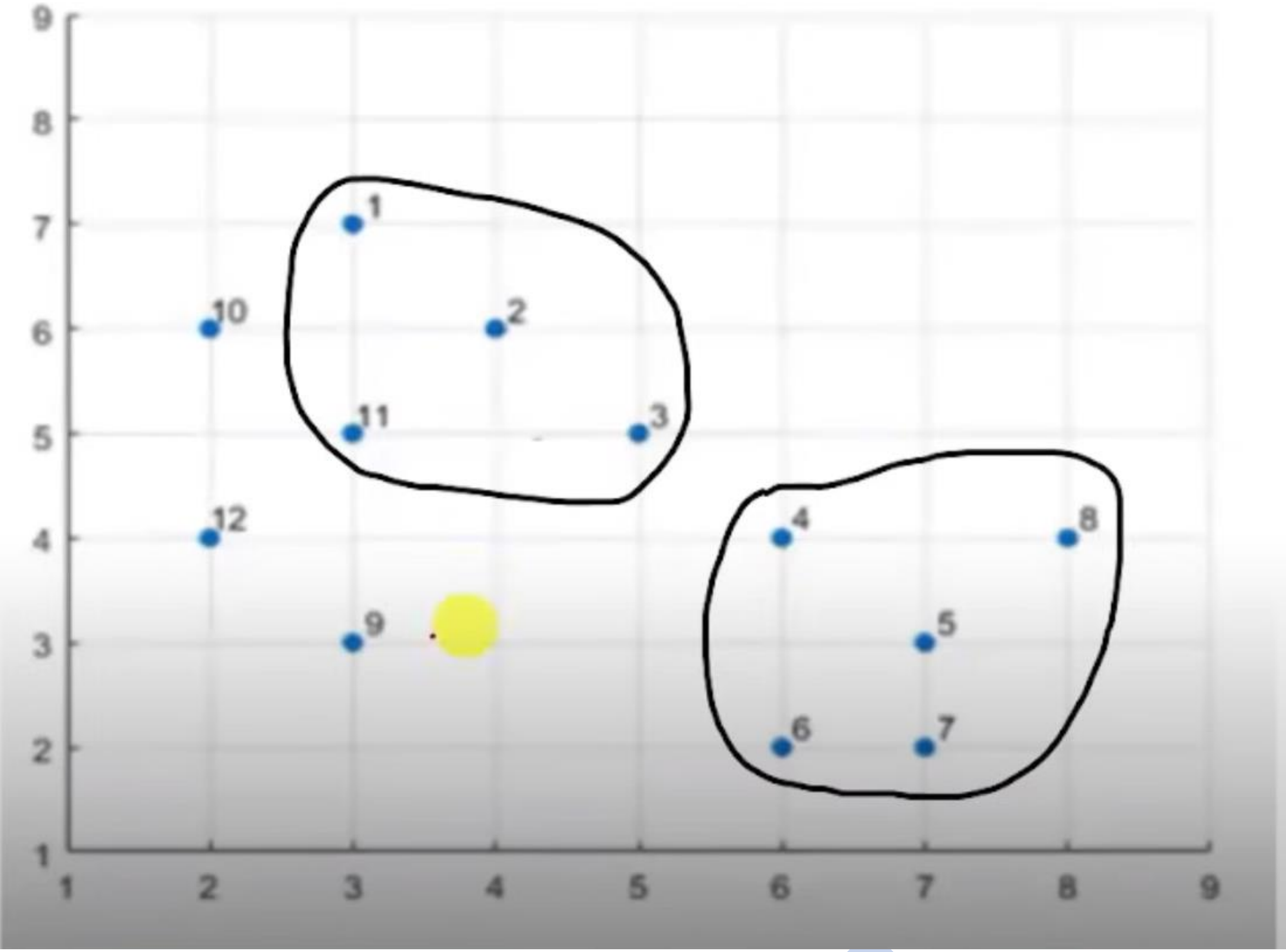
P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11

minPts = 4 and epsilon (ε) = 1.9



P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

P8: P5

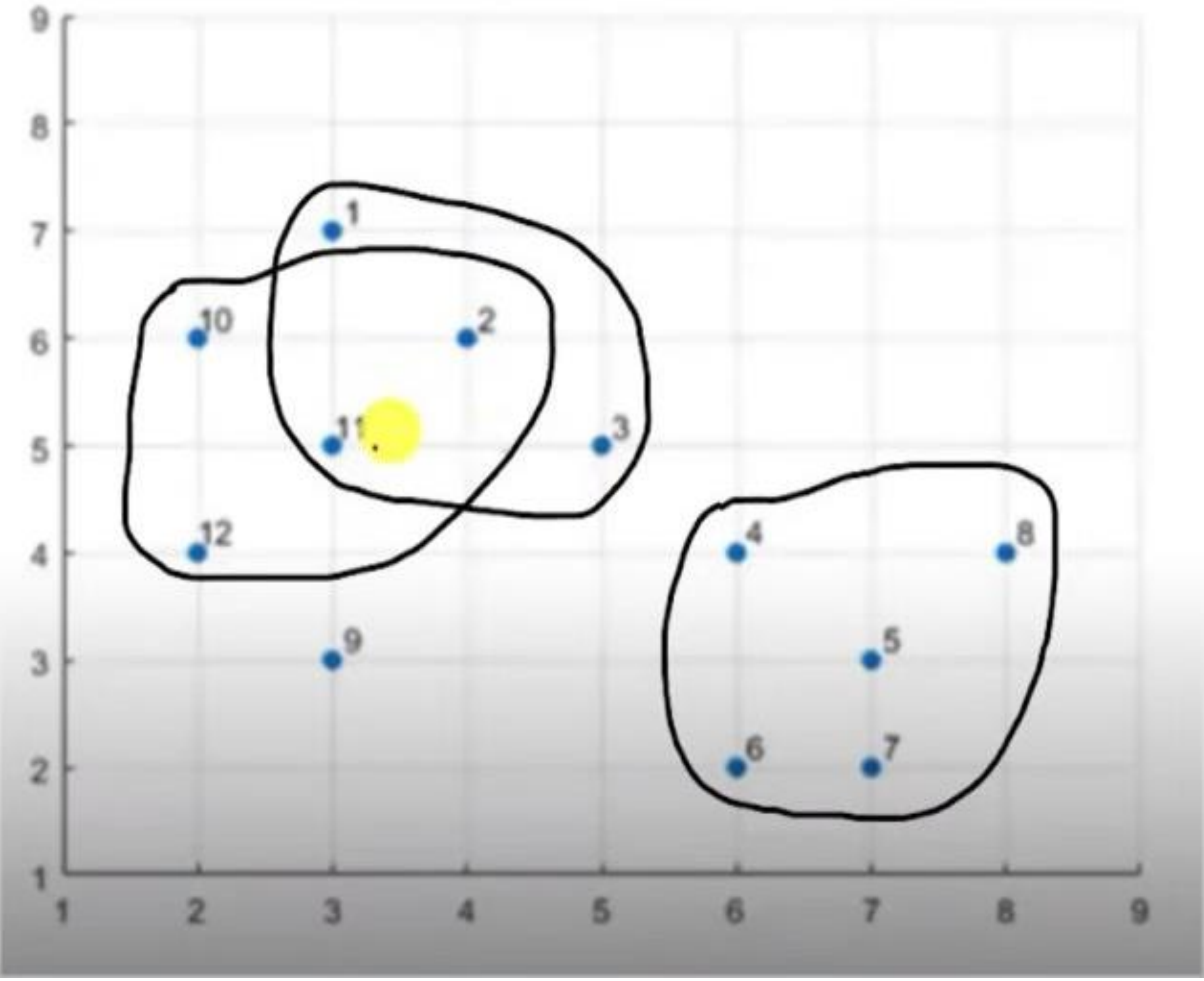
P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11

minPts = 4 and epsilon (ε) = 1.9



8.LIMITATIONS

Sensitivity to Hyperparameters

- Results depend on the right choice of **ε** and **MinPts**.
- No clear method to select them.

Noise Handling

- May wrongly label useful points as noise.
- Struggles in complex or highly noisy data.

Does Not Predict

- No **predict()** function for new data.
- Needs to re-run the algorithm for updates.

Thank You