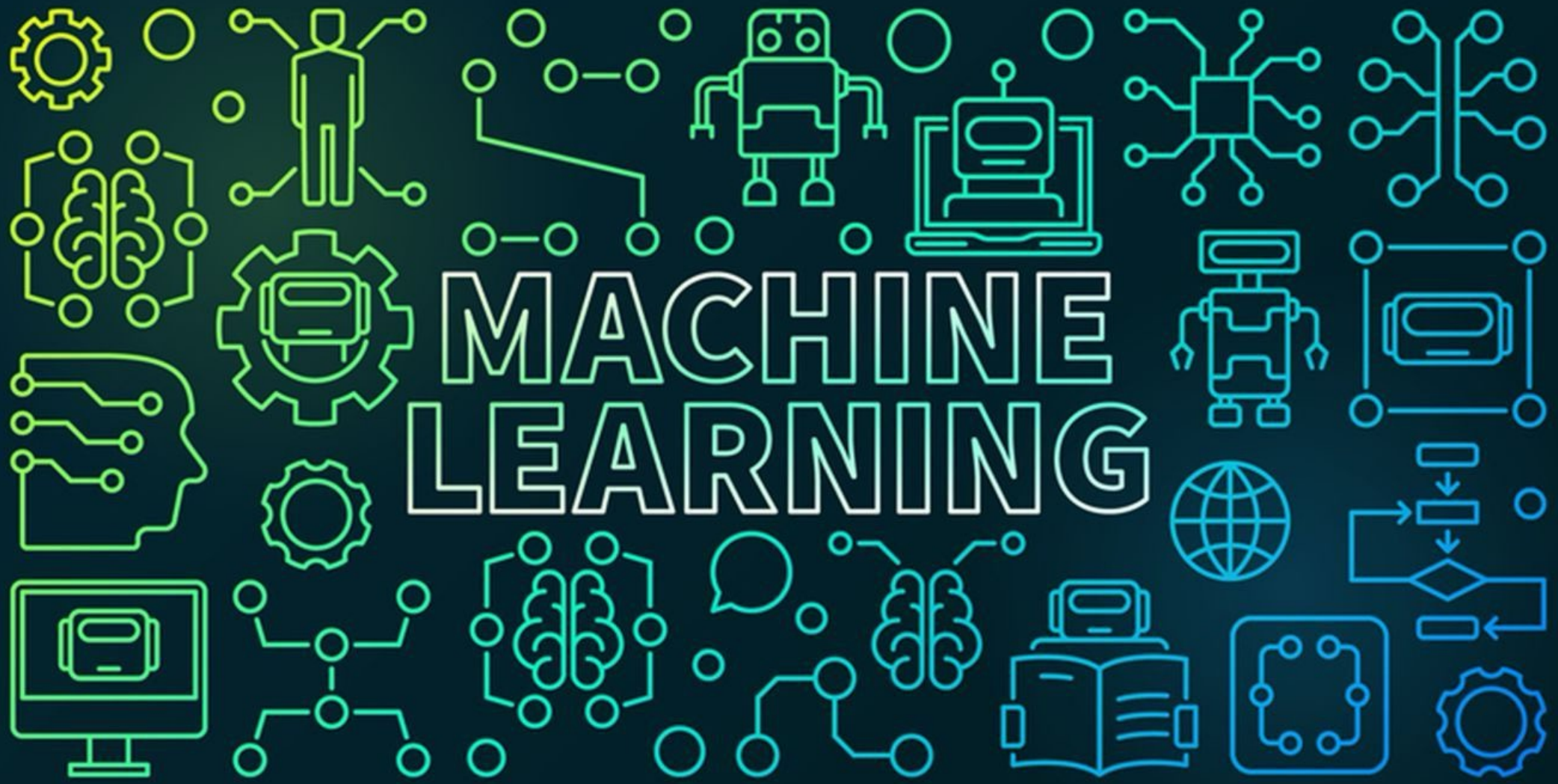


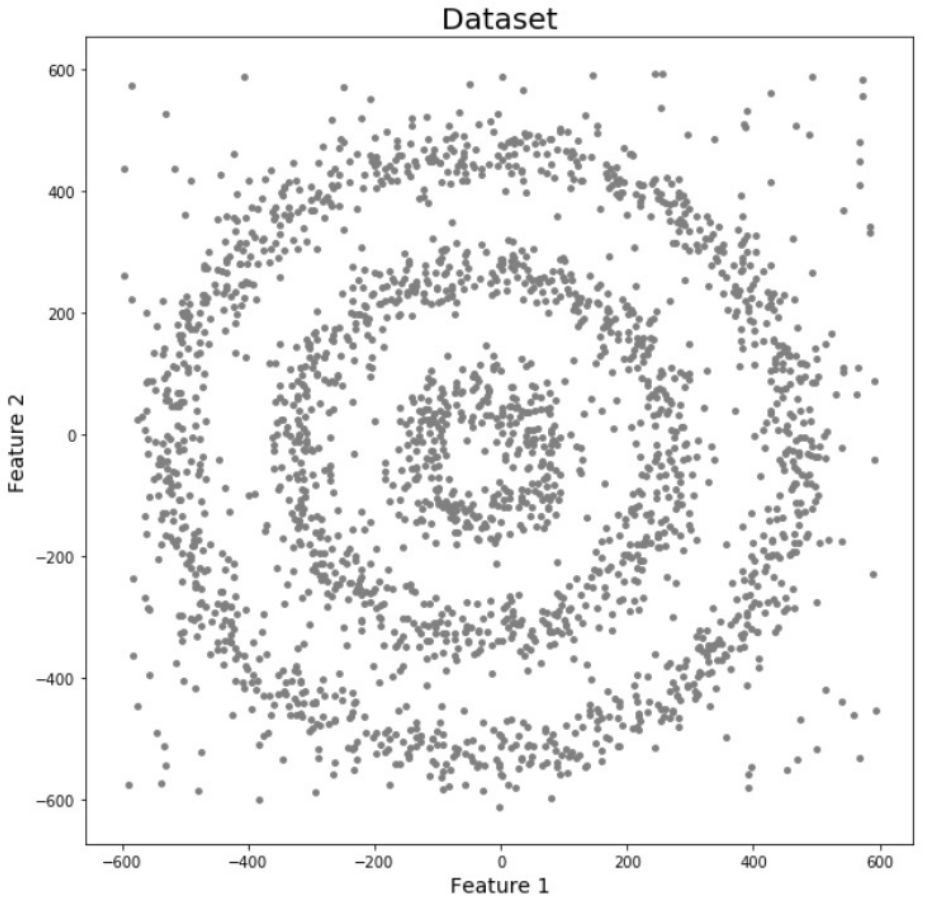
Muhammad Atif Saeed (Lecturer DS & AI)

DBSCAN Clustering

- Clustering is an unsupervised learning technique where we try to group the data points based on specific characteristics. There are various clustering algorithms with K-Means and Hierarchical being the most used ones. Some of the use cases of clustering algorithms include:
- Document Clustering
- Recommendation Engine
- Image Segmentation
- Market Segmentation
- Search Result Grouping
- and Anomaly Detection.

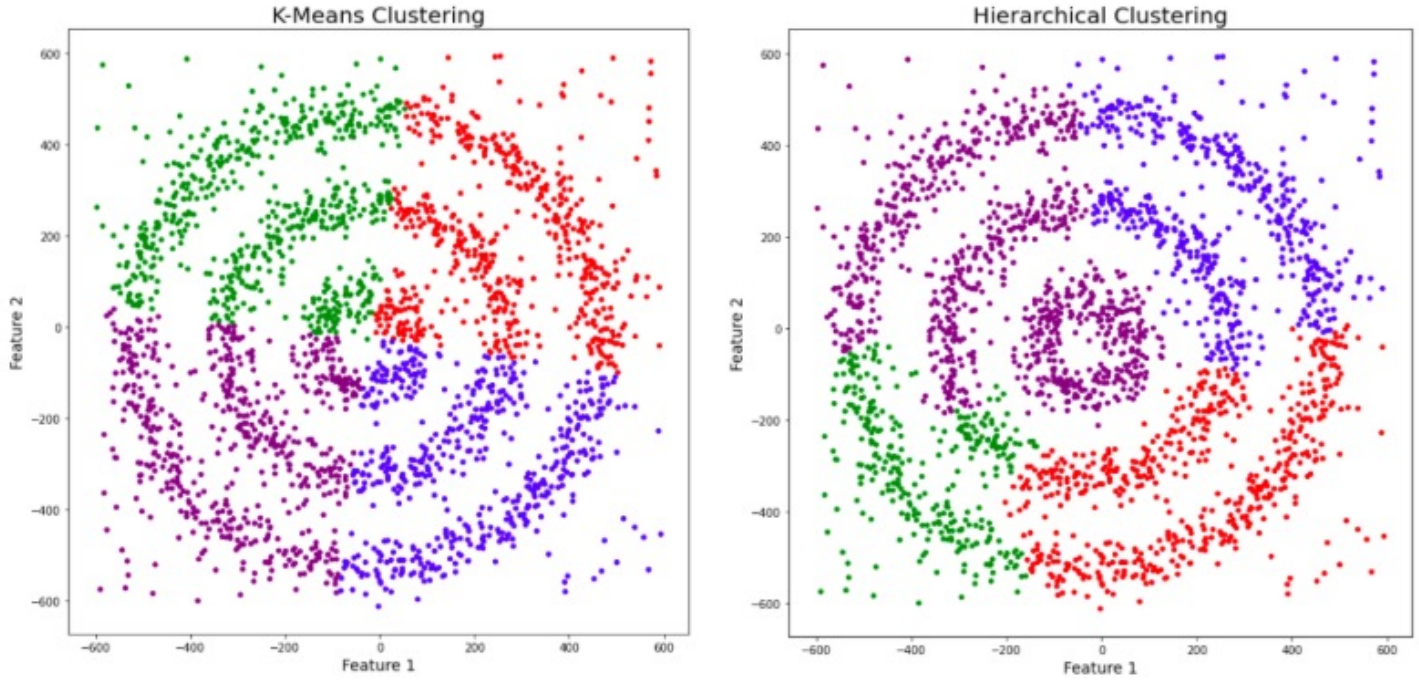
Why do we need DBSCAN Clustering?

- K-Means and Hierarchical Clustering both fail to create clusters of arbitrary shapes. They are not able to form clusters based on varying densities. That's why we need DBSCAN clustering.
- Here we have data points densely present in the form of concentric circles:



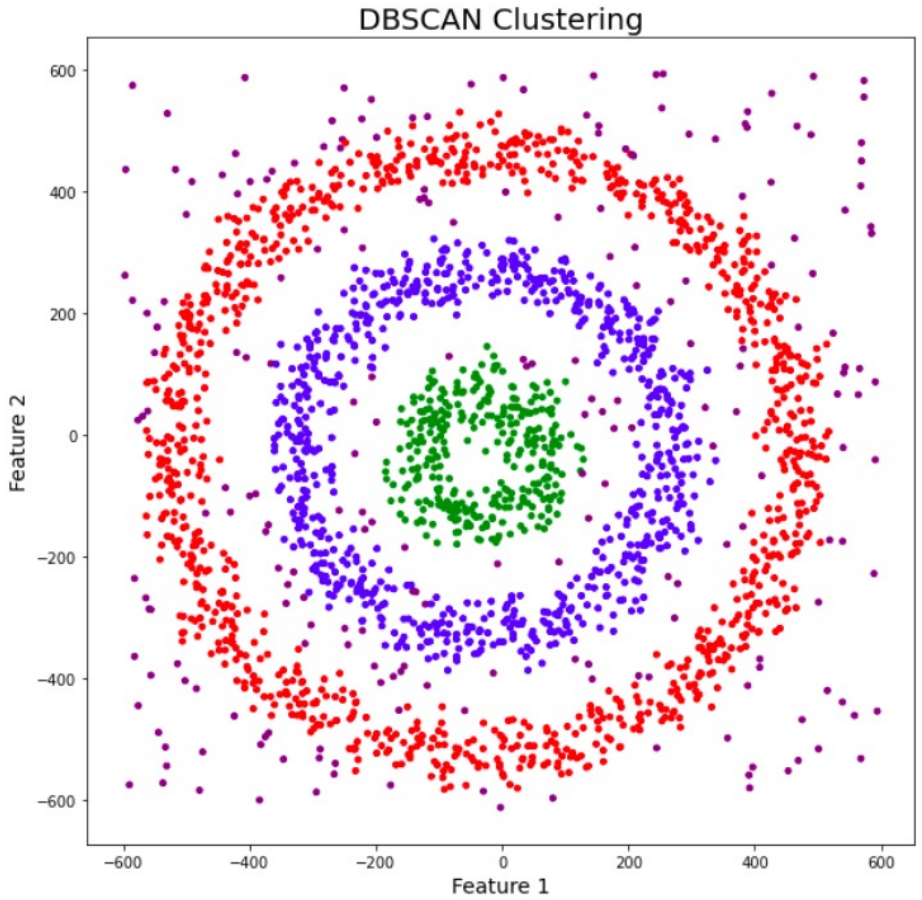
Why do we need DBSCAN Clustering?

- Now, let’s run K-Means and Hierarchical clustering algorithms and see how they cluster these data points.



Why do we need DBSCAN Clustering?

- Both of them failed to cluster the data points. Also, they were not able to detect the noise present in the dataset properly. Now, let's take a look at the results from DBSCAN clustering.



What Parameters Required DBSCAN?

- The DBSCAN algorithm relies on two main parameters to identify clusters in your data:
  - eps (ε): This parameter defines the radius of a neighborhood around a data point. Points within this distance are considered neighbors of the central point.
  - minPts: This parameter represents the minimum number of points required within the ε-neighborhood of a point to classify it as a core point. A core point is considered to be dense enough to be part of a cluster.

What Exactly is DBSCAN Clustering?

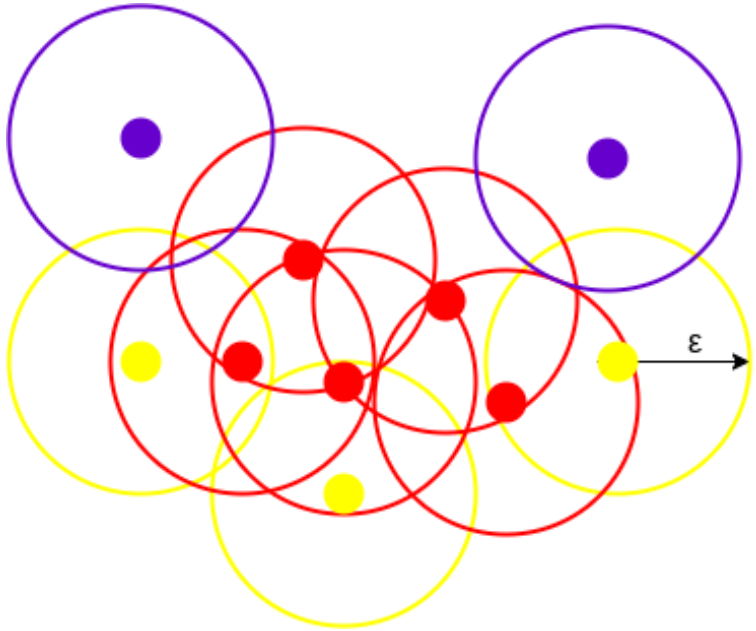
- It groups ‘densely grouped’ data points into a single cluster. It can identify clusters in large spatial datasets by looking at the local density of the data points.
- The most exciting feature of DBSCAN clustering is that it is robust to outliers. Unlike K-Means, where we have to specify the number of centroids, it also does not require the number of clusters to be told beforehand.

What Exactly is DBSCAN Clustering?

- It groups ‘densely grouped’ data points into a single cluster. It can identify clusters in large spatial datasets by looking at the local density of the data points.
- The most exciting feature of DBSCAN clustering is that it is robust to outliers. Unlike K-Means, where we have to specify the number of centroids, it also does not require the number of clusters to be told beforehand.

Example

- DBSCAN algorithm creates a circle of epsilon radius around every data point and classifies them into Core point, Border point, and Noise.
- A data point is a Core point if the circle around it contains at least ‘minPoints’ number of points.
- If the number of points is less than minPoints, then it is classified as Border Point, and if there are no other data points around any data point within epsilon radius, then it treated as Noise.



What Exactly is DBSCAN Clustering?

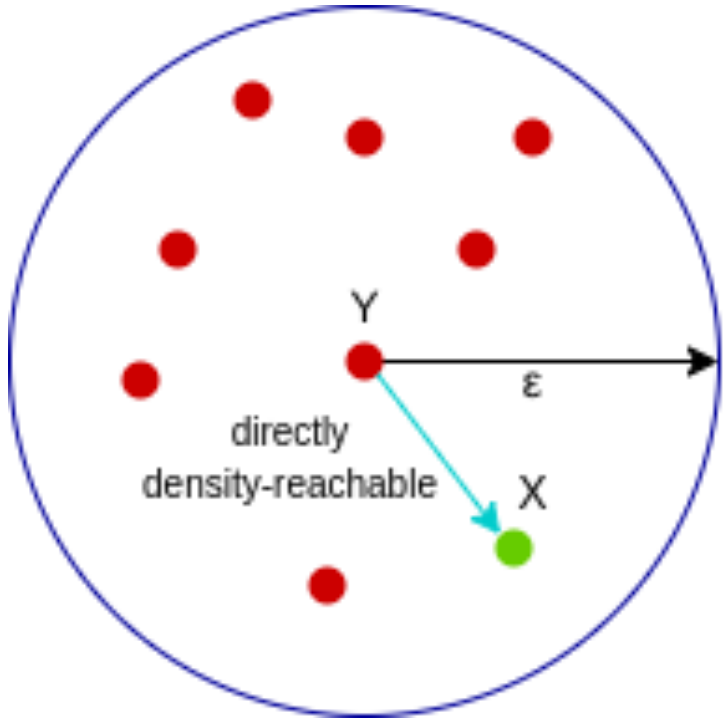
- The above figure shows a cluster created by DBCAN with minPoints = 3. Here, we draw a circle of equal radius epsilon around every data point. These two parameters help create spatial clusters.
- All the data points with at least 3 points in the circle, including itself, are considered as Core points represented by red color. All the data points with less than 3 but greater than 1 point in the circle, including itself, are considered as Border points.
- They are represented by yellow color. Finally, data points with no point other than itself present inside the circle are considered Noise, represented by the purple colour.

Reachability and Connectivity

- Reachability states if a data point can be accessed from another data point directly or indirectly, whereas Connectivity states whether two data points belong to the same cluster or not. In terms of reachability and connectivity, two points in DBSCAN can be referred to as:
- Directly Density-Reachable
- Density-Reachable
- Density-Connected

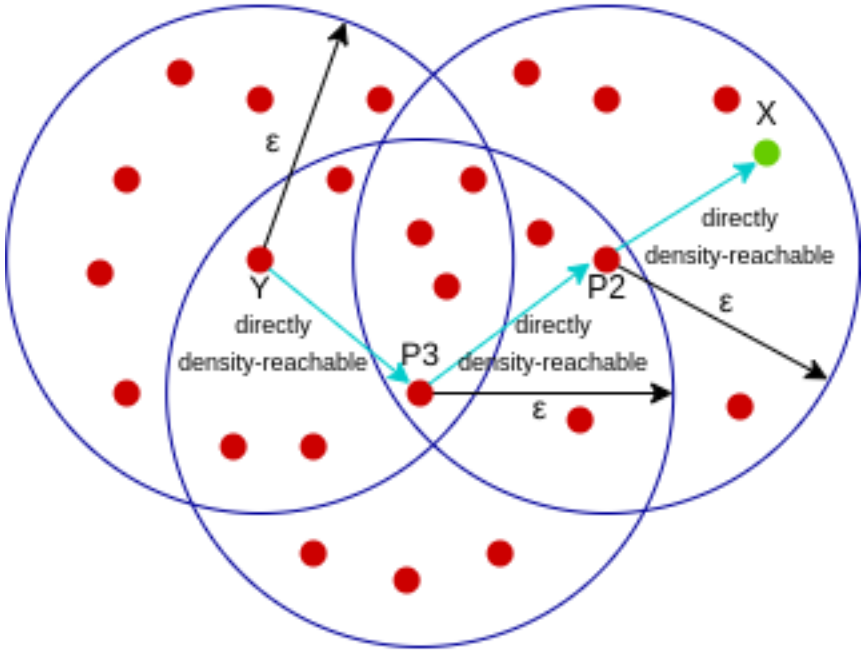
Directly density-reachable

- A point X is directly density-reachable from point Y w.r.t epsilon, minPoints if,
- X belongs to the neighborhood of Y, i.e, dist(X, Y) <= epsilon
- Y is a core point
- Here, X is directly density-reachable from Y, but vice versa is not valid.



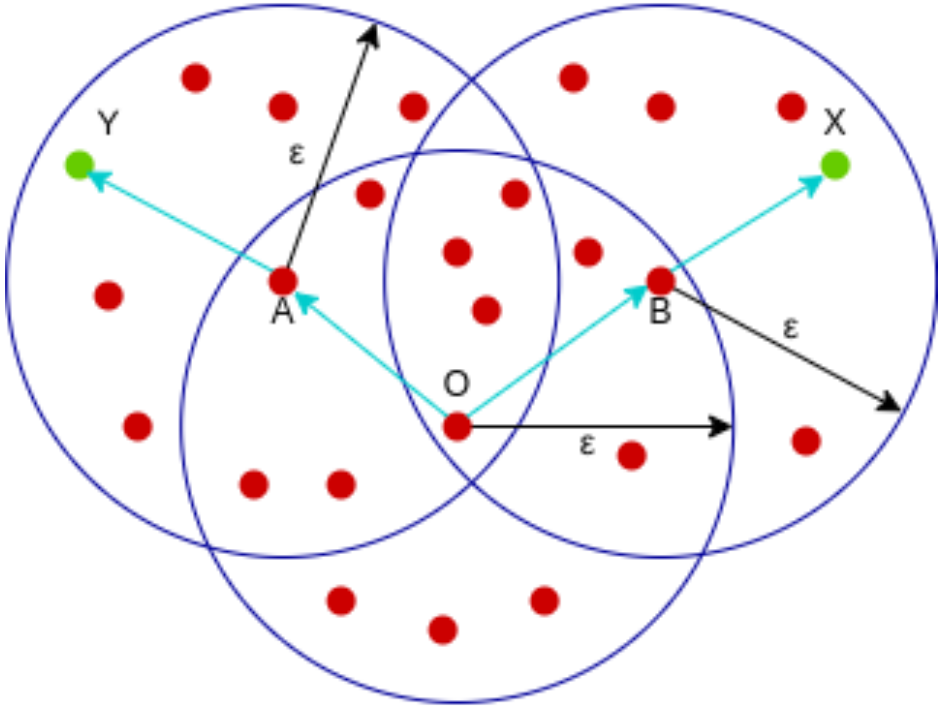
Density-reachable

- A point X is density-reachable from point Y w.r.t epsilon, minPoints if there is a chain of points p1, p2, p3, …, pn and p1=X and pn=Y such that pi+1 is directly density-reachable from pi.
- Here, X is density-reachable from Y with X being directly density-reachable from P2, P2 from P3, and P3 from Y. But, the inverse of this is not valid.



Density-connected

- A point X is density-connected from point Y w.r.t epsilon and minPoints if a point O exists such that both X and Y are density-reachable from O w.r.t to epsilon and minPoints.
- Here, both X and Y are density-reachable from O, therefore, we can say that X is density-connected from Y.



DBSCAN – Math Example

- Cluster-Aish: A(1,2), B(1,3), C(2,2.5), H(2.5,2.2)
- Cluster-Dish: D(8,8), E(8.5,8), F(9,8.5), I(9.5,9)
- Suspected noise: G(0,8)
- We'll use Euclidean distance, ε = 1.2, MinPts = 3

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

DBSCAN – Math Example

- $d(A, B) = \sqrt{\left(1-1\right)^2 + \left(2-3\right)^2} = 1.000$
- $d(A, C) = \sqrt{\left(1-2\right)^2 + \left(2-2.5\right)^2} = \sqrt{1+0.25} = 1.118$
- $d(A, H) = \sqrt{\left(1-2.5\right)^2 + \left(2-2.2\right)^2} = \sqrt{2.25+0.04} = 1.513$
- $d(B, C) = \sqrt{\left(1-2\right)^2 + \left(3-2.5\right)^2} = \sqrt{1+0.25} = 1.118$
- $d(B, H) = \sqrt{\left(1-2.5\right)^2 + \left(3-2.2\right)^2} = \sqrt{2.25+0.64} = 1.700$
- $d(C, H) = \sqrt{\left(2-2.5\right)^2 + \left(2.5-2.2\right)^2} = \sqrt{0.25+0.09} = 0.583$

DBSCAN – Math Example

- $d(D, E) = \sqrt{\left(8 - 8.5\right)^2 + \left(8 - 8\right)^2} = 0.5$
- $d(D, F) = \sqrt{\left(8 - 9\right)^2 + \left(8 - 8.5\right)^2} = \sqrt{1 + 0.25} = 1.118$
- $d(E, F) = \sqrt{\left(8.5 - 9\right)^2 + \left(8 - 8.5\right)^2} = \sqrt{0.25 + 0.25} = 0.707$
- $d(F, I) = \sqrt{\left(9 - 9.5\right)^2 + \left(8.5 - 9\right)^2} = \sqrt{0.25 + 0.25} = 0.707$
- $d(E, I) = \sqrt{\left(8.5 - 9.5\right)^2 + \left(8 - 9\right)^2} = \sqrt{1 + 1} = 1.414$
- $d(D, I) = \sqrt{\left(8 - 9.5\right)^2 + \left(8 - 9\right)^2} = \sqrt{2.25 + 1} = 1.803$

DBSCAN – Math Example

- ε-neighborhoods (ε = 1.2)
- An ε-neighborhood of a point p is all points q with $d(p, q) \leq \varepsilon$)including p).
- **A:** neighbors {A,B(1.0),C(1.118)} ⇒ |Nε(A)| = 3
- **B:** neighbors {B,A(1.0),C(1.118)} ⇒ |Nε(B)| = 3
- **C:** neighbors {C,A(1.118),B(1.118),H(0.583)} ⇒ |Nε(C)| = 4
- **H:** neighbors {H,C(0.583)} ⇒ |Nε(H)| = 2
- **D:** neighbors {D,E(0.5),F(1.118)} ⇒ |Nε(D)| = 3
- **E:** neighbors {E,D(0.5),F(0.707)} ⇒ |Nε(E)| = 3
- **F:** neighbors {F,D(1.118),E(0.707),I(0.707)} ⇒ |Nε(F)| = 4
- **I:** neighbors {I,F(0.707)} ⇒ |Nε(I)| = 2
- **G:** neighbors {G} ⇒ |Nε(G)| = 1

DBSCAN – Math Example

- Core / Border / Noise
- Core point: $|N\varepsilon(p)| \geq \text{MinPts}$ (=3 here)
- Border point: not core, but within ε of a core point
- Noise: neither core nor border
- From the counts:
  - Core: A, B, C, D, E, F
  - Border: H (within ε of core C), I (within ε of core F)
  - Noise: G

Density reachability & cluster construction

- DBSCAN expands clusters from cores by adding all points density-reachable (via chains of ε-neighbors through cores).
- Pick A (unvisited). Core ⇒ start Cluster 1 = {A}.
- Add its ε-neighbors: {B, C}. Now {A,B,C}.
- Visit B (core). Add its ε-neighbors (A,C) — already in cluster.
- Visit C (core). Add its ε-neighbors (A,B,H). H is new ⇒ add H.
- H is border (not core), so do not expand from H.
- Cluster 1 finalized: {A, B, C, H}

Density reachability & cluster construction

- Cluster 2 (continue with next unvisited core, D):
- Pick D (core). Start Cluster 2 = {D}; add ε-neighbors {E,F}. ⇒ {D,E,F}
- Visit E (core). ε-neighbors {D,F} already in cluster.
- Visit F (core). ε-neighbors {D,E,I}. I is new ⇒ add I.
- I is border; do not expand from it.
- Cluster 2 finalized: {D, E, F, I}
- Remaining point
  - G never added ⇒ noise.

Density reachability & cluster construction

- Cluster 2 (continue with next unvisited core, D):
- Pick D (core). Start Cluster 2 = {D}; add ε-neighbors {E,F}. ⇒ {D,E,F}
- Visit E (core). ε-neighbors {D,F} already in cluster.
- Visit F (core). ε-neighbors {D,E,I}. I is new ⇒ add I.
- I is border; do not expand from it.
- Cluster 2 finalized: {D, E, F, I}
- Remaining point
  - G never added ⇒ noise.

Distance matrix

	A	B	C	H	D	E	F	G	I
A	0.000	1.000	1.118	1.513	9.220	9.605	10.308	6.083	11.011
B	1.000	0.000	1.118	1.700	8.602	9.014	9.708	5.099	10.404
C	1.118	1.118	0.000	0.583	8.139	8.515	9.220	5.852	9.925
H	1.513	1.700	0.583	0.000	7.993	8.345	9.052	6.316	9.759
D	9.220	8.602	8.139	7.993	0.000	0.500	1.118	8.000	1.803
E	9.605	9.014	8.515	8.345	0.500	0.000	0.707	8.500	1.414
F	10.308	9.708	9.220	9.052	1.118	0.707	0.000	9.014	0.707
G	6.083	5.099	5.852	6.316	8.000	8.500	9.014	0.000	9.552
I	11.011	10.404	9.925	9.759	1.803	1.414	0.707	9.552	0.000