DBSCAN

* Density Based Spatial Clustering of Application With Noise

* Why?

  - → K-Mean specify no of clusters DBSCAN automatically does this
  - → DBSCAN can handle outlier automatically
  - → DBSCAN can handle irregular shapes.

* Developed by Martin Ester and Peter Kriegel in 1996.
  - Limitation of K-Means → fix.

General

Same as K-means
(Different in Mathematical approach)

Mathematical

Point	X_1	X_2
A	1	1
B	2	2
C	2	3
D	8	8
E	8.5	8.5
F	7.5	8
G	25	80
H	24	81

Two factors

① Epsilon (ε)

we set $\varepsilon = 2$

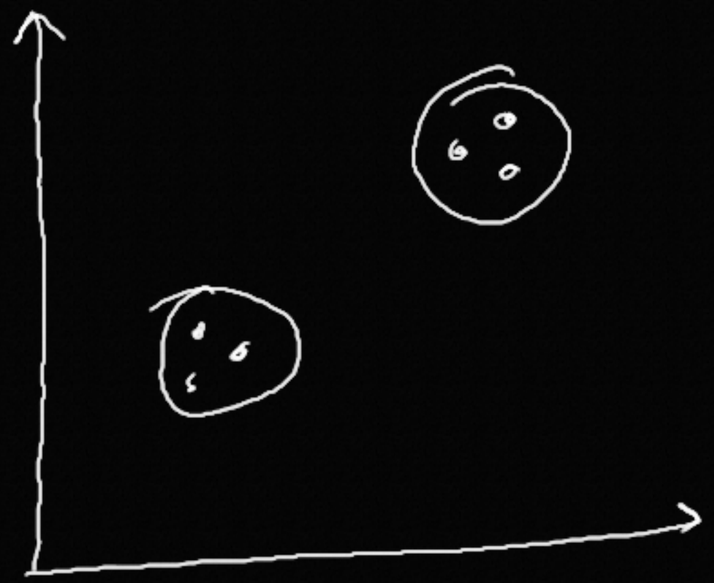
$A \leftrightarrow B = 1.8 \rightarrow$ Neigbour

$A \leftrightarrow C = 1.5 \rightarrow$ "

$A \leftrightarrow D = 4 \rightarrow$ Not neighbour

② MinPts → We set MinPts = 3 ⇒

Min points reqd to make a cluster



Classifying criteria

① Core point ⇒ Point that satisfy MinPts condition

② Border point ⇒ Not a core point but is within ε distance of core point

③ Noise point → Neither core point Nor border points

Working

MinPts = 3 $\quad$ $\varepsilon = 2$

① Start with A.

Calculate all points distance with A

$A \leftrightarrow A$, $A \rightarrow B$, $A \leftrightarrow C$, $A \leftrightarrow D$, - - - - - $A \leftrightarrow H$

Using euclidean distance formula

Check which distance is less than ε

Neighbour points ⇒ A, B ⇒ 2 (Not satisfied)

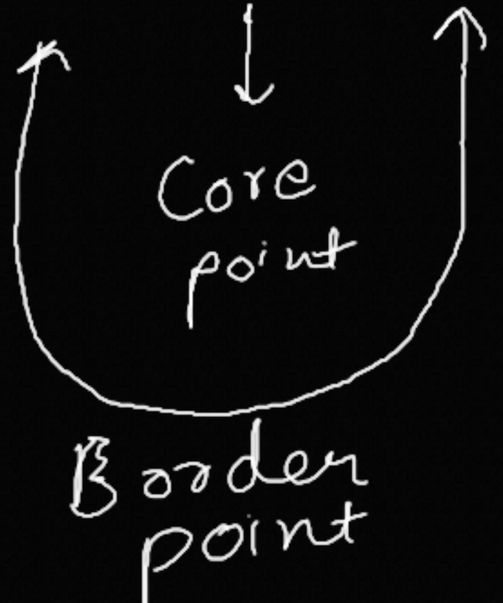
It is not a core point

② Now point B.

Neighbour points = A, B, C ⇒ 3 (satisfied condition)

(B is a core point)

Cluster 1 = {A, B, C}



③ Point D

Neighbours = D, E, F ⇒ 3 (satisfied condition)

(D is a core point)

Cluster 2 = {D, E, F}

D — core

E, F — border point

④ Remaing points = {G, H} = 2

↳ Never ever satisfy condition

↓

(Noise point)

final Clusters

Cluster 1

{A, B, C}

cluster 2

{D, E, F}

Noise

{G, H}

Model trained

Prediction

New point = (7.9, 7.9)

Calculate all distances again

Neighbours = {D, E, F}
($\varepsilon = 2$)

Check majority of point belong to which cluster.

Output ⇒ Cluster 2

When to Use

* Irregular cluster shape
* Data has noise / outlier
* No of cluster is unknown
* Cluster are of varying sizes.

When Not to Use

* Dataset is very large & high-dimensional (Computational Cost ↑)