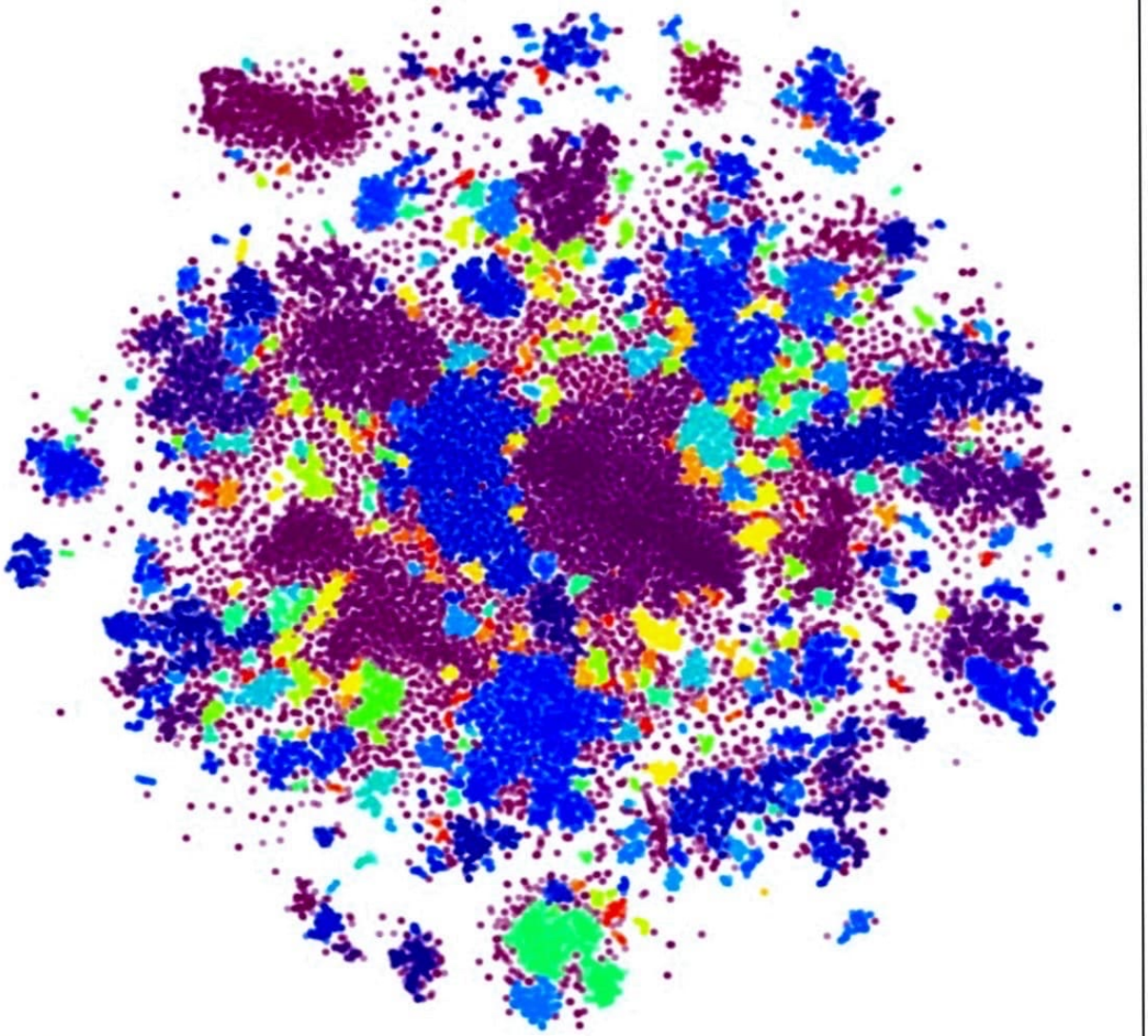
BSCAN

ty-Based Spatial Clustering of Applications with



Inventors of the DBSCAN algorithm :
Ester, Kriegel, Sander and Xu in the
year 1996

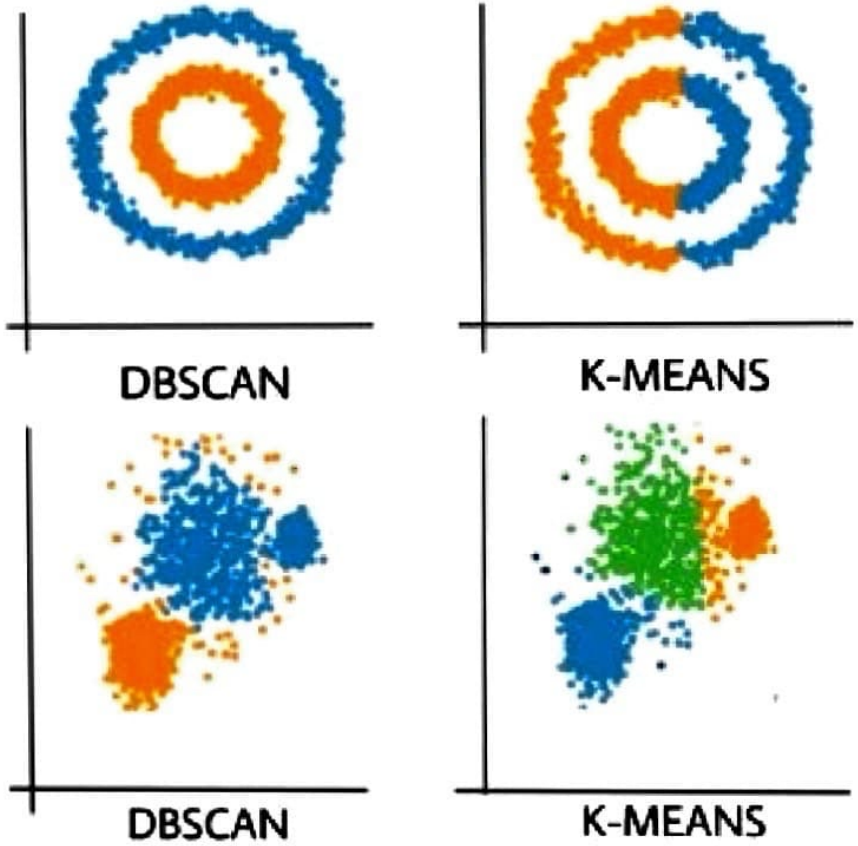
backs of K-means and Hierarchical clusterin

Partitioning method partitions the dataset to k (the main input of the methods) number of groups (clusters). The Partition iterative process allocates each point in the dataset to the group it belongs to.

After the points are allocated in its group, the mean of the group (its centroid) is calculated by taking an average for all the points in the group. The most well-known Partitioning method is K-means.

The partition methods have some significant drawbacks:

- K-Means algorithm requires one to specify the number of clusters a priory (the K value).
- K-means does not perform well on finding non-convex/non-spherical shapes of clusters (see the below image)



backs of K-means and Hierarchical clustering

Hierarchical also has some serious drawbacks:

- It isn’t suitable for big datasets, has high computational complexity.
- Need a metric for merging the clusters (linkage) that affects the clustering result.
- Sensitivity to noise

As we can see the main disadvantages of partitioning and hierarchical methods are: handling noise and getting bad results with finding clusters of nonspherical shape

DBSCAN

DBSCAN is a density-based clustering method that discovers clusters of non spherical shape. The DBSCAN clustering method can represent clusters of arbitrary shape and to handle noise

Clusters of arbitrary shape.

DBSCAN intuition

Differing groups of points by their density is the main idea of the DBSCAN. The DBSCAN groups together points with a dense neighbourhood into clusters.

A point will be considered as crowded if it has many other neighbours points near it. The DBSCAN finds these crowded points and places them and their neighbours in a cluster

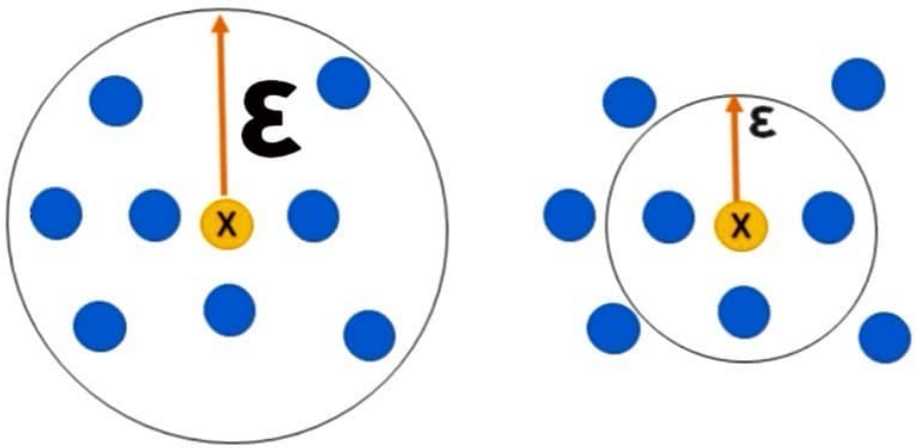
Let’s think about a very big city with a lot of residents and tourists. Imagine that there is a small village just a short drive away.

If we would take an alien to both places, although they are very close, he could easily tell that they are completely different places. Yes, the view, area, buildings, and many other aspects are completely different.

But there is one aspect that is relevant for our case — the density of the places. The city is crowded, with a lot of locals and tourists, whereas the village is small with significantly fewer people.

CAN parameters : ε (or eps or epsilon)

ε (or eps or epsilon) : Defines the size and borders of each neighbourhood. The ε (must be bigger than 0) is a radius. The neighbourhood of point x called the ε-neighbourhood of x, is the circle/ball with radius ε around point x



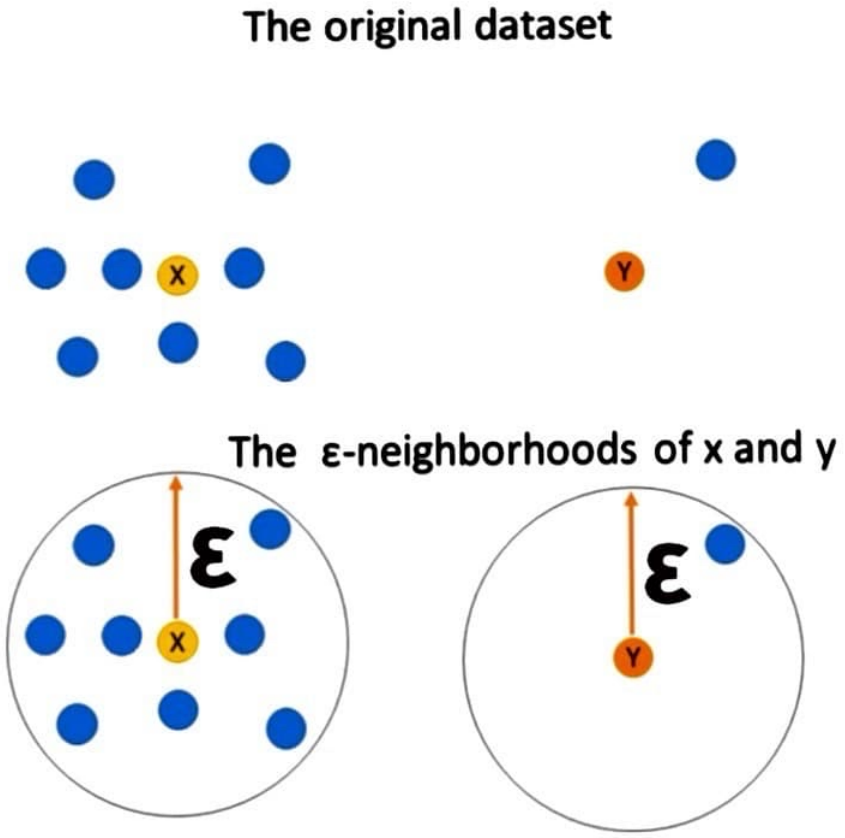
$$N_\epsilon(\mathbf{x}) = B_d(\mathbf{x}, \epsilon) = \{\mathbf{y} \mid \delta(\mathbf{x}, \mathbf{y}) \leq \epsilon\}$$

CAN parameters : ε (or eps or epsilon)

Point x and its neighbors would be in one neighborhood and y and its few neighbors would be in another.

In the end, however, point x and its neighbors would probably be in one cluster whereas point y and its neighbor would be considered as outliers or noise.

This is because the ε-neighborhood of y isn't dense enough. As a neighborhood contains more points, the denser it becomes.

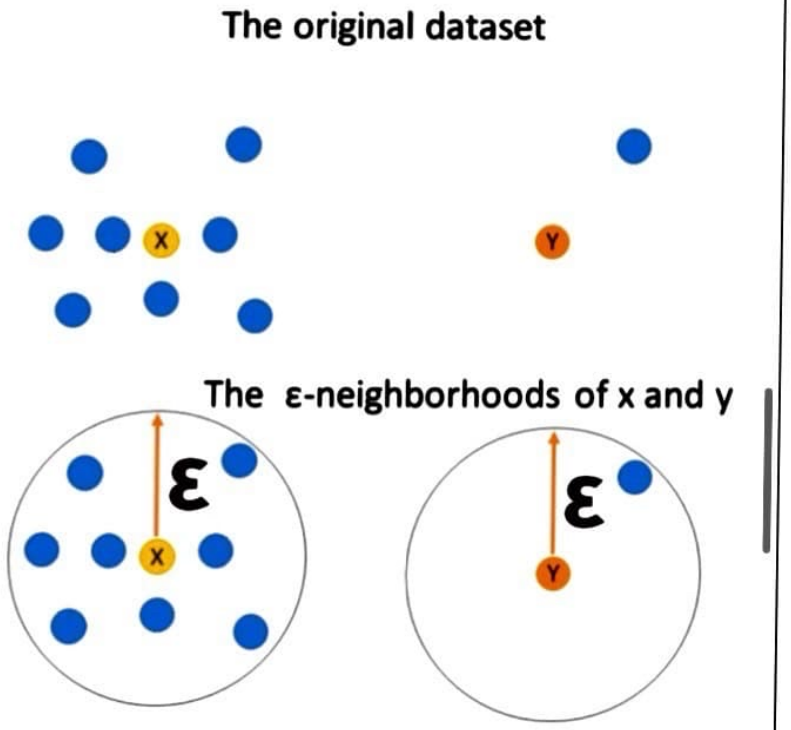


SCAN parameters : MinPts

How can we define if a neighborhood is dense enough? In order to do that, the DBSCAN uses the second parameter MinPts.

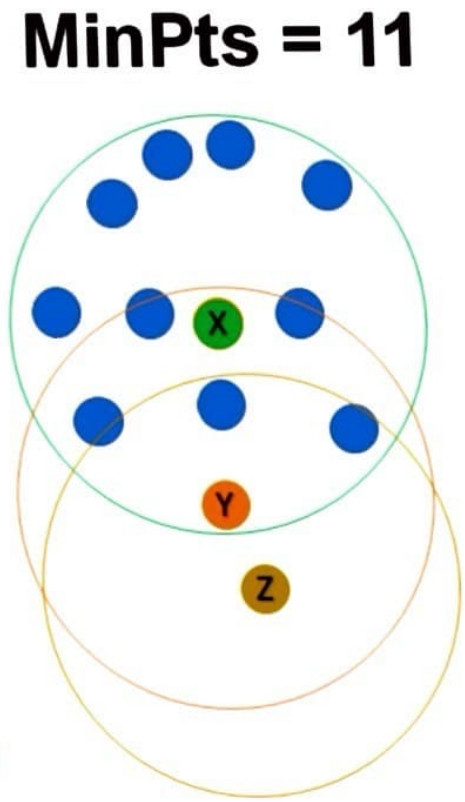
MinPts : The density threshold. If a neighborhood includes at least MinPts points, it will be considered as a dense region. Alternatively, a point will be considered as dense if there are at least the value of MinPts points in its ε-neighborhood. These dense points are called core points.

Let's check the image, if the MinPts parameter is 3, the point x will be a core point because the size of its ε-neighborhood is 9 and it's bigger than 3. Point y won't be a core point because its ε-neighborhood contains two points.



Border Point

A border point has ε-neighborhood that contains less than MinPts points (so it's not a core point), but it belongs to the ε-neighborhood of another core point.
If a point isn't a core point and isn't a border point, it's a noise point or an outlier.



we can see that point x is a core point, because it has more than 11 points in its ε-neighborhood.

Point y isn't a core point because it has less than 11 points in its ε-neighborhood, but because it belongs to the ε-neighborhood of point x, and point x is a core point, point y is a border point. We can easily see that point z isn't a core point. It belongs to the ε-neighborhood of point y, and point y isn't a core point, therefore point z is a noise point.

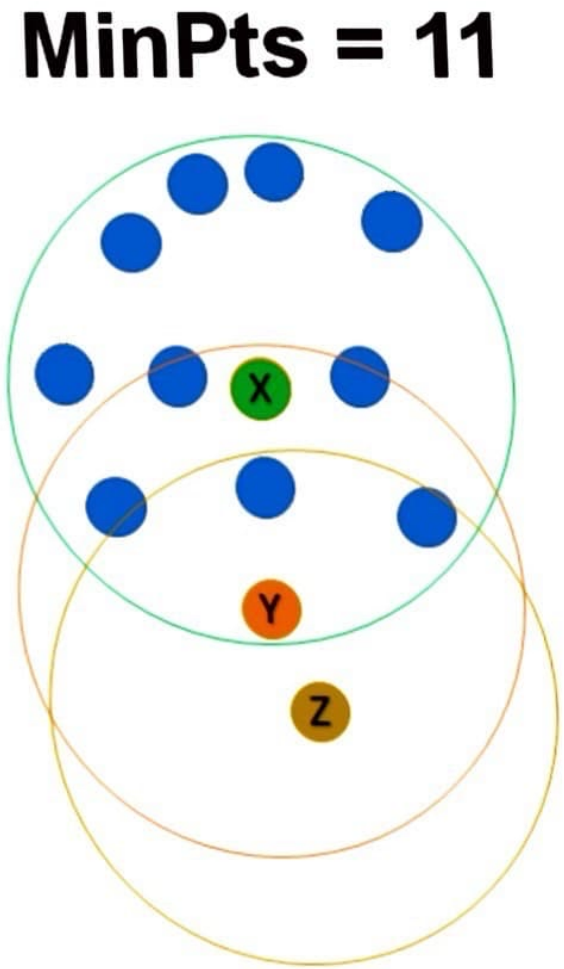
rectly Density Reachable

We are clear about the core points and border points will be in a cluster, now how does the DBSCAN know which point goes to which cluster? In order to answer that, we need to define some definitions:

Directly density reachable:
Point y is directly density reachable from point x if:

- Point y belongs to the ε-neighbourhood of point x
- And Point x is a core point.

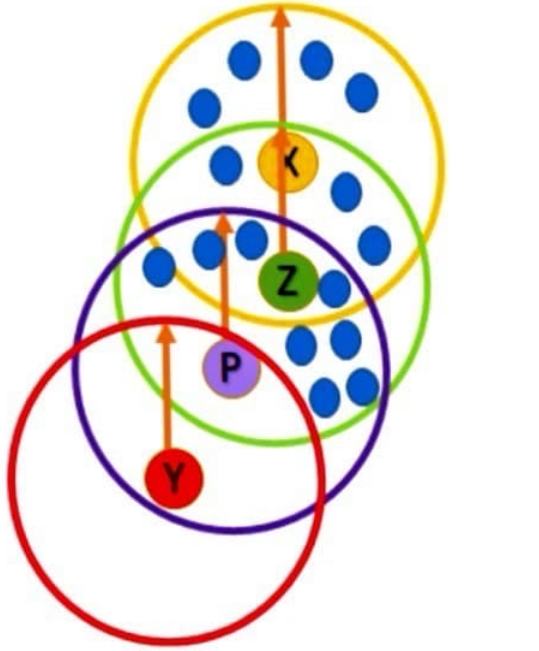
•In the image , point y is directly density reachable from point x. Notice that point z isn't directly density reachable from point y, because point y isn't a core point



rectly Density Reachable

Density reachable : Point y is density reachable from point x, if there is a path of points between point x and point y, where each point in the path is directly reachable from the previous point. This means that all the points on the path are core points, with the possible exception of point y.

A point q is reachable from core point p if there is a path $P_1, P_2 \ldots, P_n$ with $P_1 = p$ and $P_n = q$, that all points on the path are core points, with the possible exception of point q.



w to choose MinPts and Ɛ?

In general, ε should be chosen as small as possible.

Parameters Estimation
If ε has a small value, many points may be considered as outliers because they wouldn't be core points or border points (the ε-neighborhoods will be very small). A large value for ε may cause a huge amount of points to be in the same cluster.

As said before, one of the main advantages of the DBSCAN is that it detects noise. According to a research made in 2017 by Schubert, Sander, et al, the desirable amount of noise will usually be between 1% and 30%. Another insight from that research is that if one of the clusters contains many (20%-50%) points of the dataset, it indicates that you should choose a smaller value for ε or to try another clustering method.

ics for Measuring DBSCAN's Performance

Silhouette Score: The silhouette score is calculated utilizing the mean intra- cluster distance between points, AND the mean nearest-cluster distance. For instance, a cluster with a lot of data points very close to each other (high density) AND is far away from the next nearest cluster (suggesting the cluster is very unique in comparison to the next closest), will have a strong silhouette score. A silhouette score ranges from -1 to 1, with -1 being the worst score possible and 1 being the best score. Silhouette scores of 0 suggest overlapping clusters.

Let’s comprehend

'arameters

DBSCAN algorithm requires two parameters –

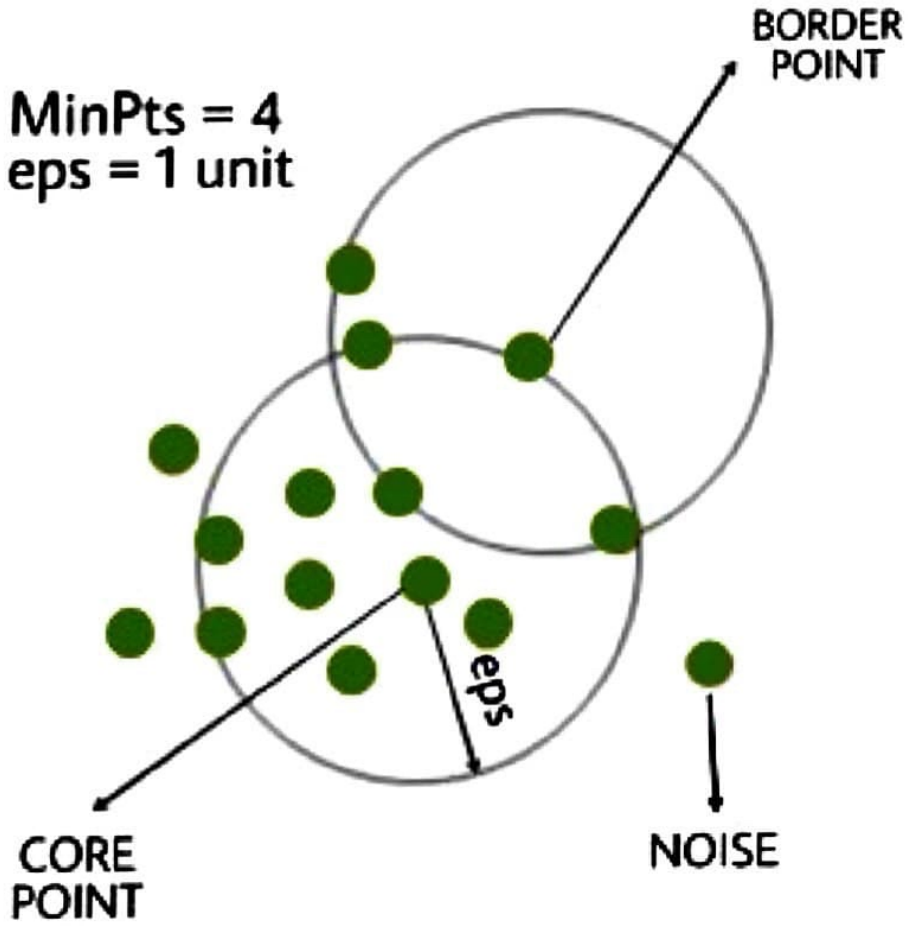
1.eps : It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to 'eps' then they are considered as neighbors. If the eps value is chosen too small then large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and majority of the data points will be in the same clusters. One way to find the eps value is based on the k-distance graph.

2.MinPts: Minimum number of neighbors (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3.

types of data points

In this algorithm, we have 3 types of data points.
Core Point: A point is a core point if it has more than MinPts points within eps.

Border Point: A point which has fewer than MinPts within eps but it is in the neighborhood of a core point.
Noise or outlier: A point which is not a core point or border point.



BSCAN algorithm

DBSCAN algorithm can be summarized in the following steps :

1.Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.

2.For each core point if it is not already assigned to a cluster, create a new cluster.

3.Find recursively all its density connected points and assign them to the same cluster as the core point.

A point a and b are said to be density connected if there exist a point c which has a sufficient number of points in its neighbors and both the points a and b are within the eps distance. This is a chaining process. So, if b is neighbor of c, c is neighbor of d, d is neighbor of e, which in turn is neighbor of a implies that b is neighbor of a.

4.Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

DBSCAN PROS

- Identifies randomly shaped clusters
- doesn’t necessitate to know the number of clusters in the data previously (as opposed to K-means)
- Handles noise

DBSCAN CONS

- Datasets with varying densities are problematic
- Input parameters (ε and MinPts) may be difficult to determine
- computational complexity — when the dimensionality is high, it takes $O(n^2)$