DBSCAN

Density based spatial clustering of application with noise

Introduction: Clustering analysis is an unsupervised learning method that separates the data points into several specific bunches or groups, such that the data points in the same groups have similar properties and data points in different groups have different properties in some sense.

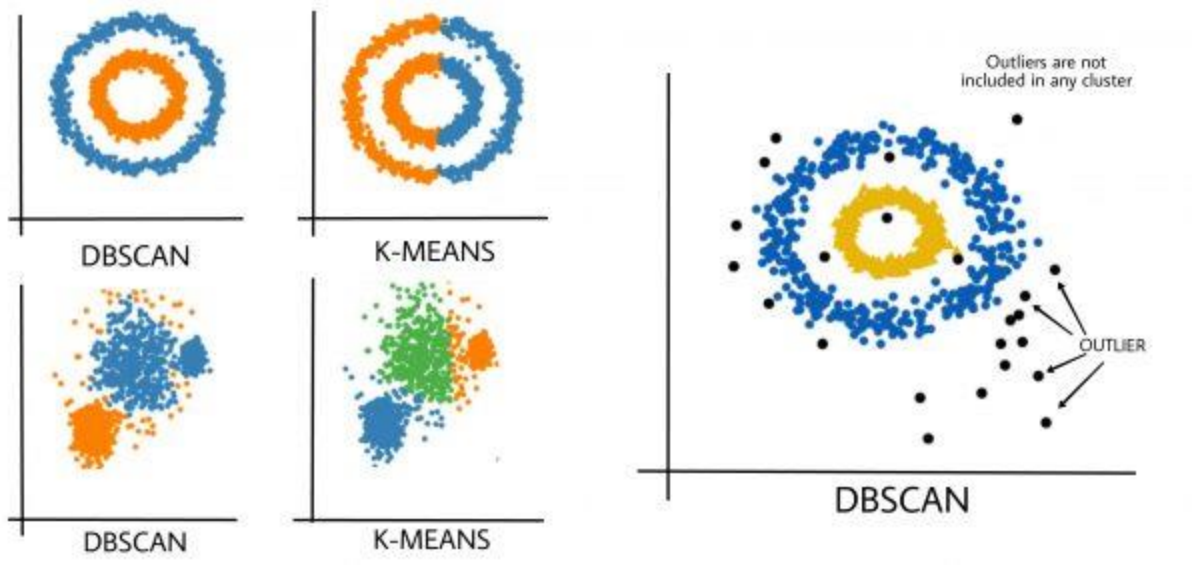
DBSCAN: DBSCAN stands for density based spatial clustering of applications with noise, DBSCAN is a clustering method that used in machine learning to separate clusters of high density from clusters of low density.

Why do we need a Density-Based clustering algorithm like DBSCAN when we already have K-means clustering?

K-Means clustering may cluster loosely related observations together. Every observation becomes a part of some cluster eventually, even if the observations are scattered far away in the vector space. Since clusters depend on the mean value of cluster elements, each data point plays a role in forming the clusters. A slight change in data points *might* affect the clustering outcome. This problem is greatly reduced in DBSCAN due to the way clusters are formed. This is usually not a big problem unless we come across some odd shape data.

Another challenge with *k*-means is that you need to specify the number of clusters ("*k*") in order to use it. Much of the time, we won't know what a reasonable *k* value is *a priori*.

What's nice about DBSCAN is that you don't have to specify the number of clusters to use it. All you need is a function to calculate the distance between values and some guidance for what amount of distance is considered "close". DBSCAN also produces more reasonable results than *k*-means across a variety of different distributions.

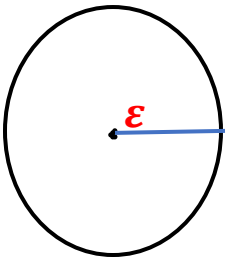


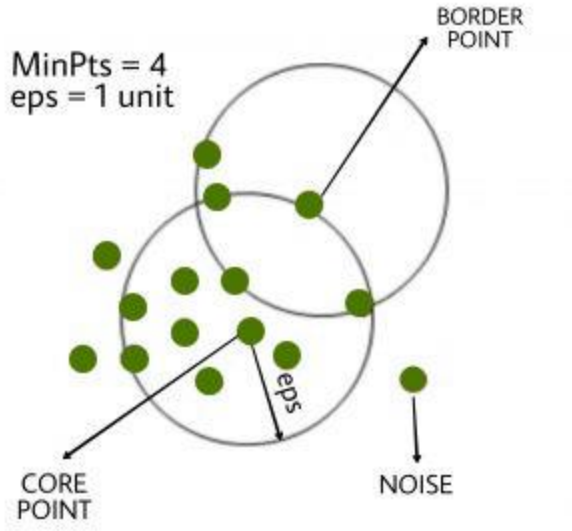
DBSCAN algorithm requires two parameters:

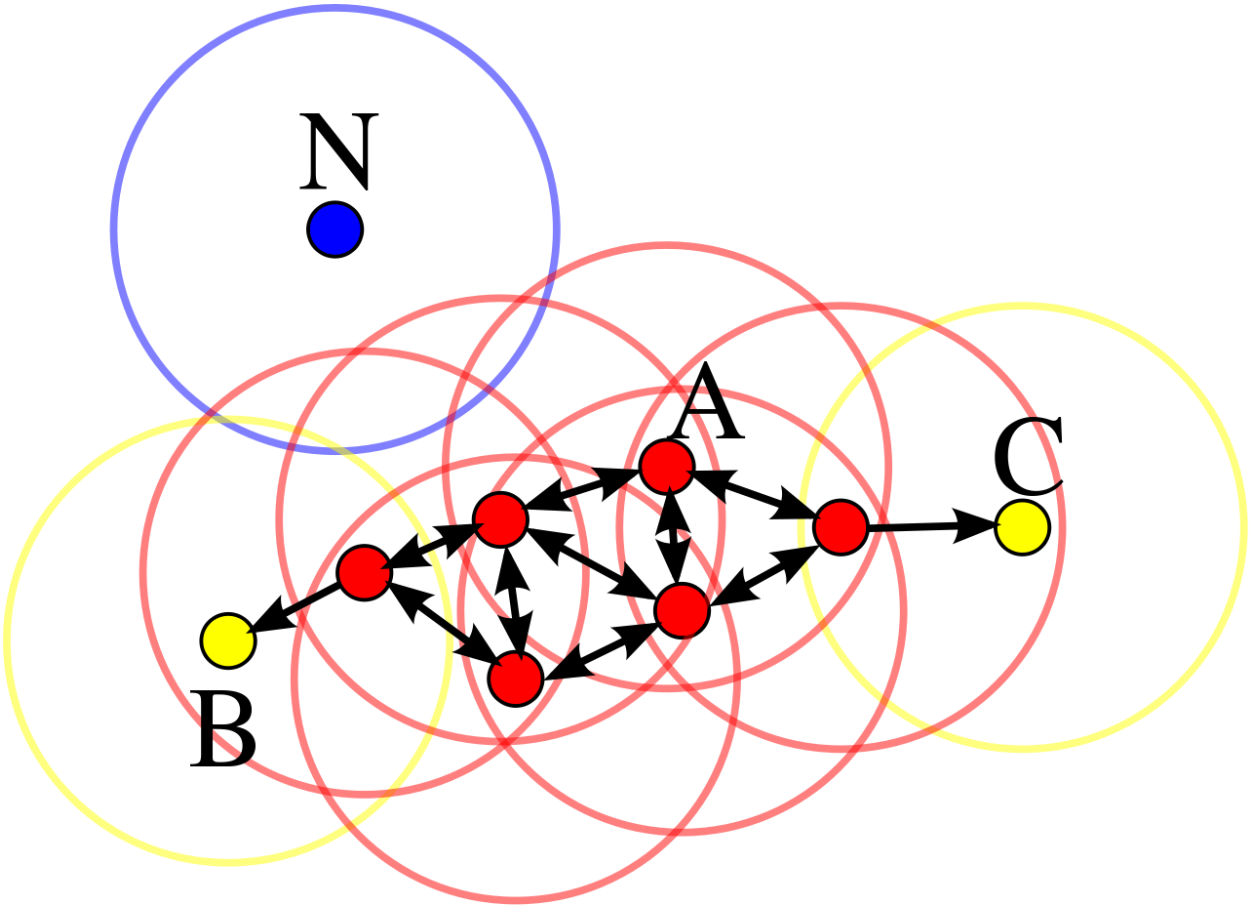
1. **Eps (ε):** It defines the neighborhood around a data point i.e., if the distant between two points is lower or equal to 'eps' then they are considered as neighbors. If the 'eps' value is chosen too small then large part of the data will be considered as outliers. If it is chosen very large then 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.** **You can say it normally the circle radius.**
2. **MinPts:** Minimum number of neighbors (data points) within 'eps' radius. Larger the dataset, the larger value of Minimum-point 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.

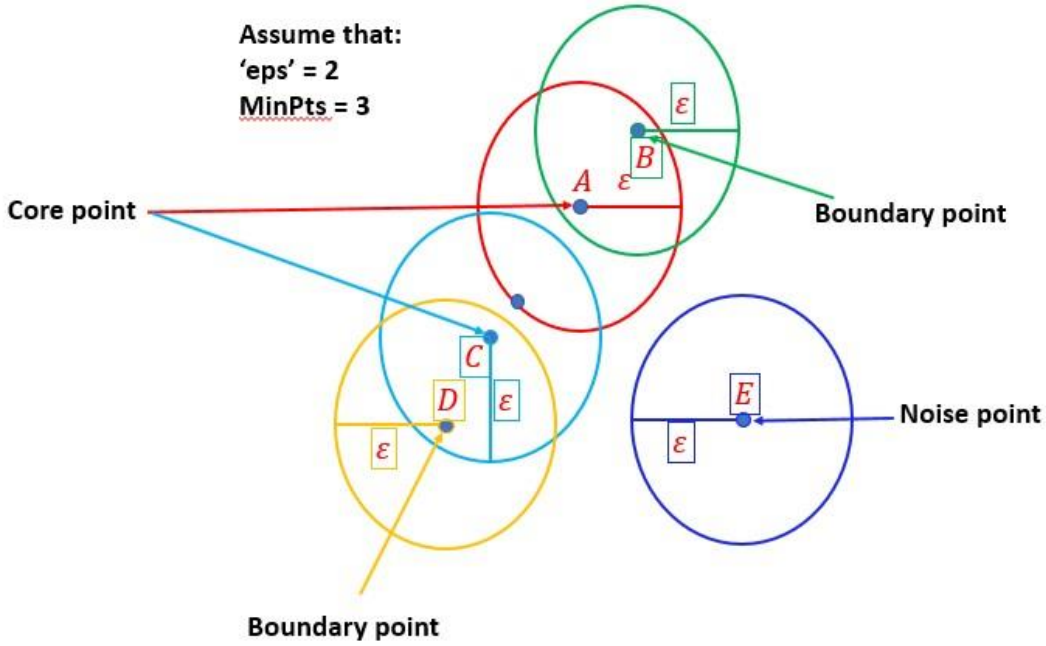
In this algorithm, we have 3 types of data points.

1. Core Point: A point is a core point if it has more than MinPts points within 'eps'
2. Border Point: A point which has fewer than MinPts within 'eps' but is in the **neighborhood of a core point**.
3. Noise or outlier: A point which is not core point or border point.




MinPts = 4
eps = 1 unit
BORDER POINT
eps
CORE POINT
NOISE


N
A
C
B



1) ***Boundary point is nothing but the neighbors point of any core point.***
2) ***Core point is nothing but that point based on this point we create circle.***

DBSCAN algorithm can be abstracted 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.