← Go Back to Data Analysis & Visualization

Course Content

# DBSCAN

**Density-Based Spatial Clustering of Applications with Noise (DBSCAN)**

DBSCAN is another type of clustering algorithm. It works based on the concept of density difference. Density is interpreted as the concentration of the samples present in a region. The algorithm differentiates between high and low-density regions and clusters the points accordingly.

There are two parameters to the algorithm which is used to define the term dense.

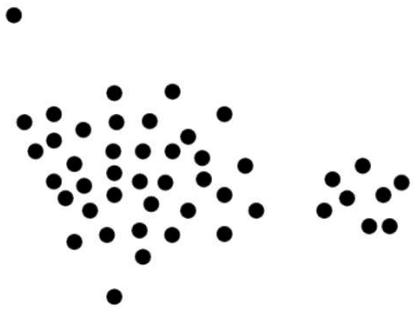
**epsilon (eps)**: It specifies how close points should be to each other to be considered as part of a cluster.

Two points are considered neighbors if the distance between the two points is below the threshold epsilon.

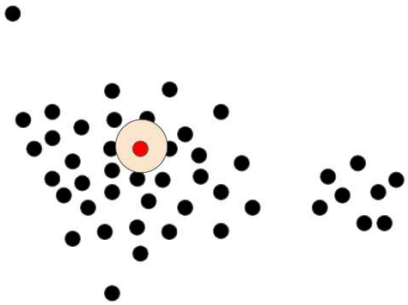
For example, let's say we are defining epsilon = 5, then all the points having a distance less than 5 will be considered a single cluster. The Euclidean distance is calculated by default but we can also use distance measurements.

**minPoints**: The minimum number of points to form a dense region. For example, if we set the minPoints parameter as 4, then we need at least 4 points to form a dense region.

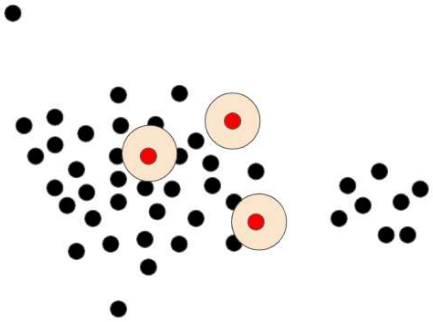
**Let's understand how the algorithm works by considering the data points shown in the below image:**



**Step 1:** We can take any random point and draw a circle around it. The radius of the circle (epsilon) should be passed as a parameter.



**Step 2:** The above step should be continued for all the data points present

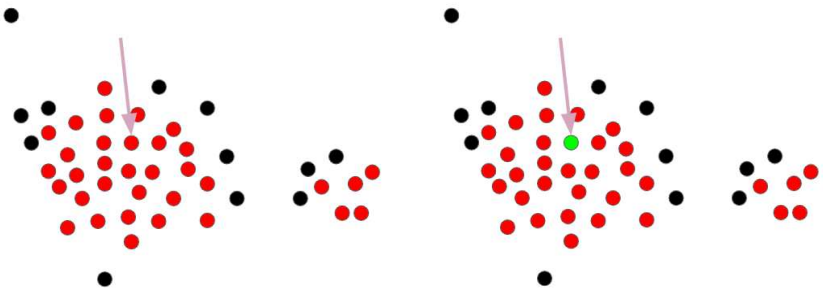


**Step 3:** Identifying **Core points**. To Classify a point as a Core, it should satisfy two criteria. The point which we are considering (red) should **at least** be

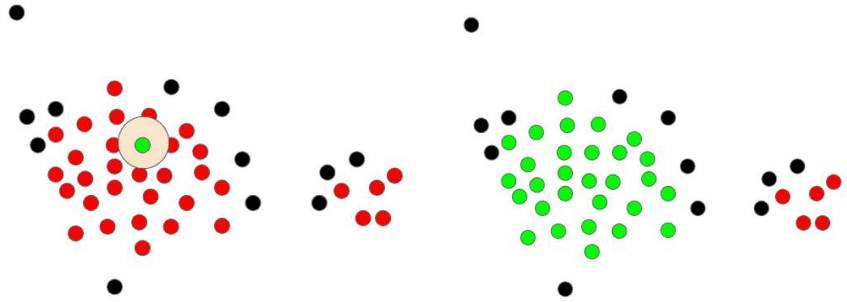
- surrounded by the minimum number of points specified (minpoints)
- within a circle of radius (epsilon)

Example: We will take a point and draw a circle (epsilon=5) around it and minPoints as 4 . In the first image above we can find the conditions being satisfied as the circle is overlapping at least 4 points but in the second image the condition is not satisfied as the overlapping is less than 4, such points are called **non-core points**.

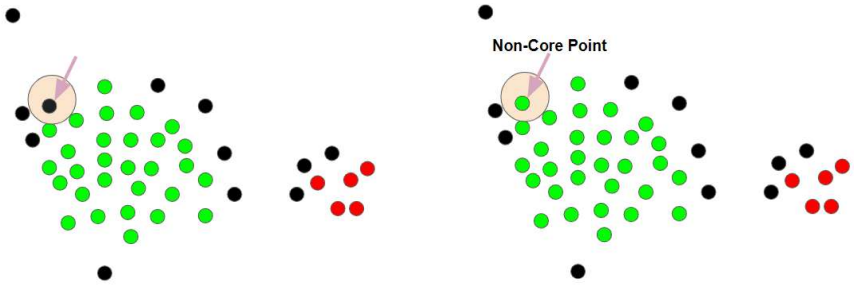
**Step 4a:** The below image shows core points (red) and non-core points (black). We randomly pick a Core point and assign it to a cluster.



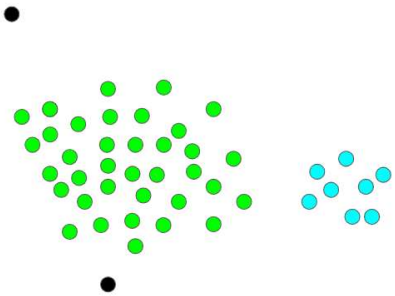
**Step 4b:** The core points that are within the radius of epsilon are all added to the cluster. This continues until there is no core point in the radius of epsilon. Initially, it adds only the core points to the cluster.



**Step 5:** When the cluster reaches a non-core point and if it falls in the epsilon range, it will be added to the cluster but can't be used to extend the cluster.




**Step 6:** Once a cluster is completely formed, then a new core point will be chosen to form another cluster and the process continues. There will be some points that will not be a part of any cluster and they will be treated as outliers.



In the above images, the green represents points that belong to the first cluster, blue points represent points that belong to the