The algorithm used by **DBSCAN** to create clusters can be described in 4 steps.

1. Classify data points:

The DBSCAN Algorithm uses Density to cluster the data points and eliminate noise.

The variables used in this algorithm are

1. **Epsilon** : the radius or "neighborhood" of each point
2. **Min_Samples**: least number of points or "Threshold" for creating core clusters

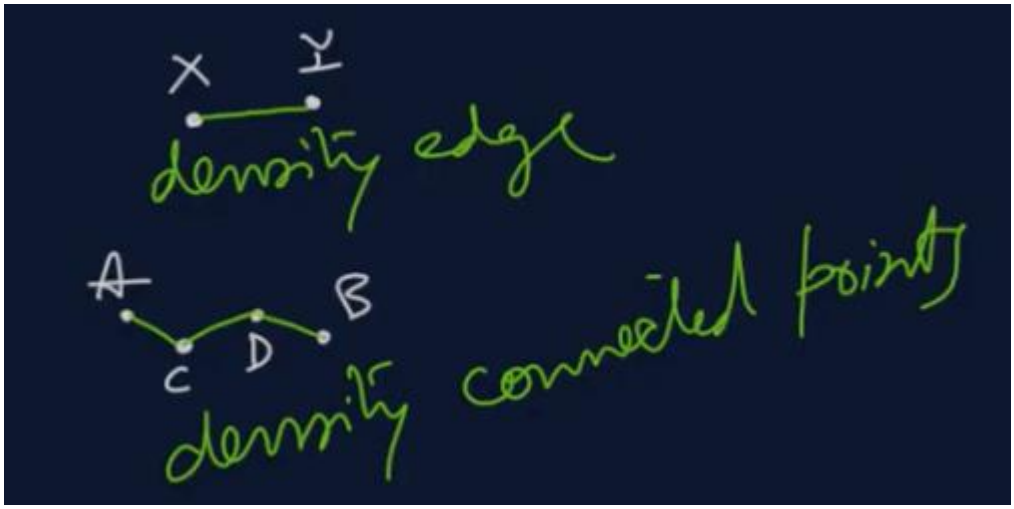
The different types of clusters are:

1. **Core Points**: have >= min samples in neighborhood
2. **Boundary Points**: have < than min samples, but are in the neighborhood of at least one core point
3. **Noise Point**: Neither a core point or a boundary point.

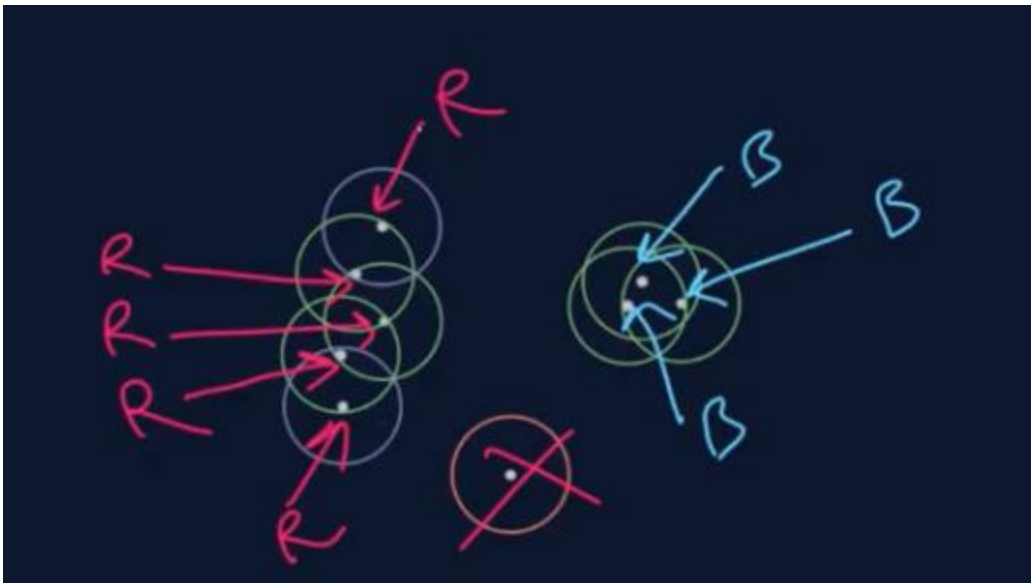
Lines connect core points to create Density Edges

DCP - **Density connected points**:

When two core points which are not neighbors are connected via density edges. Only core points can be Density connected points.



2. Discard noise points
3. Assign a cluster to a core point
4. Identify all DCPs of the core points and assign them to the same cluster
5. Identify boundary points and assign them to the nearest core point cluster



The DBSCAN algorithm is very sensitive to its initial parameters.
Even slight changes to the Epsilon or min_samples can produce very different results.

So how do we determine our initial epsilon and min_samples?

Method 1

* As a rule of thumb, if you have a 2 dimensional dataset, it is recommended to start with a min_sample between 5 - 10 and then calibrate as needed for best results.
* Of course this depends heavily on the density of your dataset and the domain knowledge of your dataset as well as what results you are aiming for.
* This becomes much less effective if you have 3 or more dimensions to your dataset.
* In fact, DBSCAN is not recommended for datasets with a dimensionality greater than 2.

Method 2

* Line plot the distance of the 10th neighbor of each datapoint in a random sample.
* The distance should gradually increase until reaching a threshold point and create a sort of Elbow as it jumps outward.
* If you draw a horizontal line at the elbow point, that Y should be your Epsilon.
* However, an elbow joint is not always generated with real world data so this is not always possible.