Unsupervised learning- you do not know the number of clusters or cluster membership

DBscan a density based algorithm, clusters are chosen by density

Parameters of Interest:

eps (radius) - The maximum distance between two samples for one to be considered as in the neighborhood of the other. This is not a maximum bound on the distances of points within a cluster. This is the most important DBSCAN parameter to choose appropriately for your data set and distance function.

min_samples - The number of samples (or total weight) in a neighborhood for a point to be considered as a core point. This includes the point itself.

Time complexity

O(nlogn)

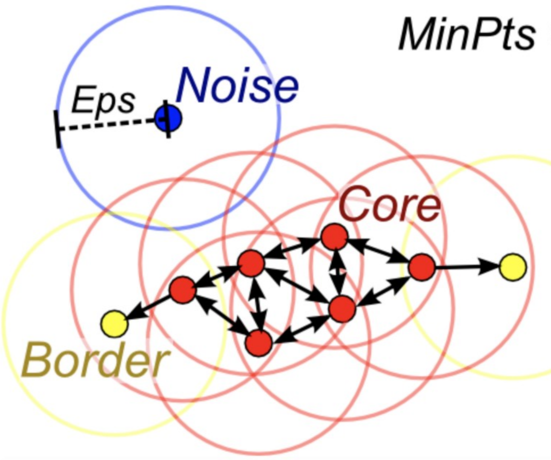
Problems

Vulnerable to overlapping clusters
Have to pick eps and min_samples
Is not guaranteed to produce the same result every run (it depends on which cluster the algorithm starts with)
choosing min_samples and eps means that you are interested in a minimum density of points (this many points within a radius of eps). Other, less dense, clusters are missed.

Algorithm (Choose eps, min_samples)

```
    1. Randomly pick a point
    2. while( other points to process)
           If there are at least min_samples within eps distance of that point, it is a core point
Now all core points are classified
    3. Randomly pick a non clustered core point
    4. Assign to a cluster
    5. for every non assigned core point withen eps of CP
    6.        assign all these core points to that cluster
    7. Add all non core points withen eps of a core point to the cluster
    8. If any leftover, unassigned core points, go to step 3
```

Any remaining points are called outliers



Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

HDBscan similar to DBscan with the addition of handling varying density clusters

Parameters of Interest:
min_cluster_size: minimum number of points needed to be considered a cluster
(note that eps is gone, which makes sense since different cluster densities will have varying eps. And if eps varies, we dont fix it as a hyperparameter)
min_samples: same as DBscan, minimum number of neighbors to a core point. Make this high, then clusters are dense and more points in non-dense space are marked as outliers.

Time complexity
O(nlogn)

Problems:
Not part of scikit-learn but performs like scikit learn estimators.
min_samples parameter is somewhat unintuitive

Algorithm
Not applicable for this course