GENERAL ASSEMBLY

DBSCAN & HDBSCAN

Matt Brems
Data Science Immersive, GA

LEARNING OBJECTIVES

- By the end of this lesson, students should be able to:
  - Describe the effect of `epsilon` and `min_points` on DBSCAN.
  - Implement DBSCAN.
  - Identify advantages and disadvantages of DBSCAN.
  - Explain and implement HDBSCAN.

K-MEANS & HIERARCHICAL CLUSTERING

- In unsupervised learning, one strategy is to cluster observations into groups.
  - Observations in the same group are more similar than observations in different groups.

- We've learned two methods of clustering thus far: *k*-Means and hierarchical.

K-MEANS & HIERARCHICAL CLUSTERING

- What are the pros/cons to using *k*-means or hierarchical clustering?

DBSCAN

- There's another method of clustering that can sidestep some of the disadvantages of *k*-means and hierarchical clustering: **DBSCAN**.
    - Density-Based Spatial Clustering of Applications with Noise

DBSCAN

- There's another method of clustering that can sidestep some of the disadvantages of *k*-means and hierarchical clustering: **DBSCAN**.
    - Density-Based Spatial Clustering of Applications with Noise
    - We detect areas of high and low density.
        - We cluster in high-density areas.
        - We avoid clustering in low-density areas.

DBSCAN

- DBSCAN requires you to specify two hyperparameters:
  - **`min_samples`**: the minimum number of points needed to form a cluster.
  - **`epsilon`**: the "searching" distance when attempting to build a cluster.

VISUALIZING DBSCAN

- https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

HOW DOES DBSCAN WORK?

1. Choose your hyperparameters.

2. Pick a random, un-clustered observation p to start and create a "pseudo-cluster" that consists of p and all observations within `epsilon` of p.

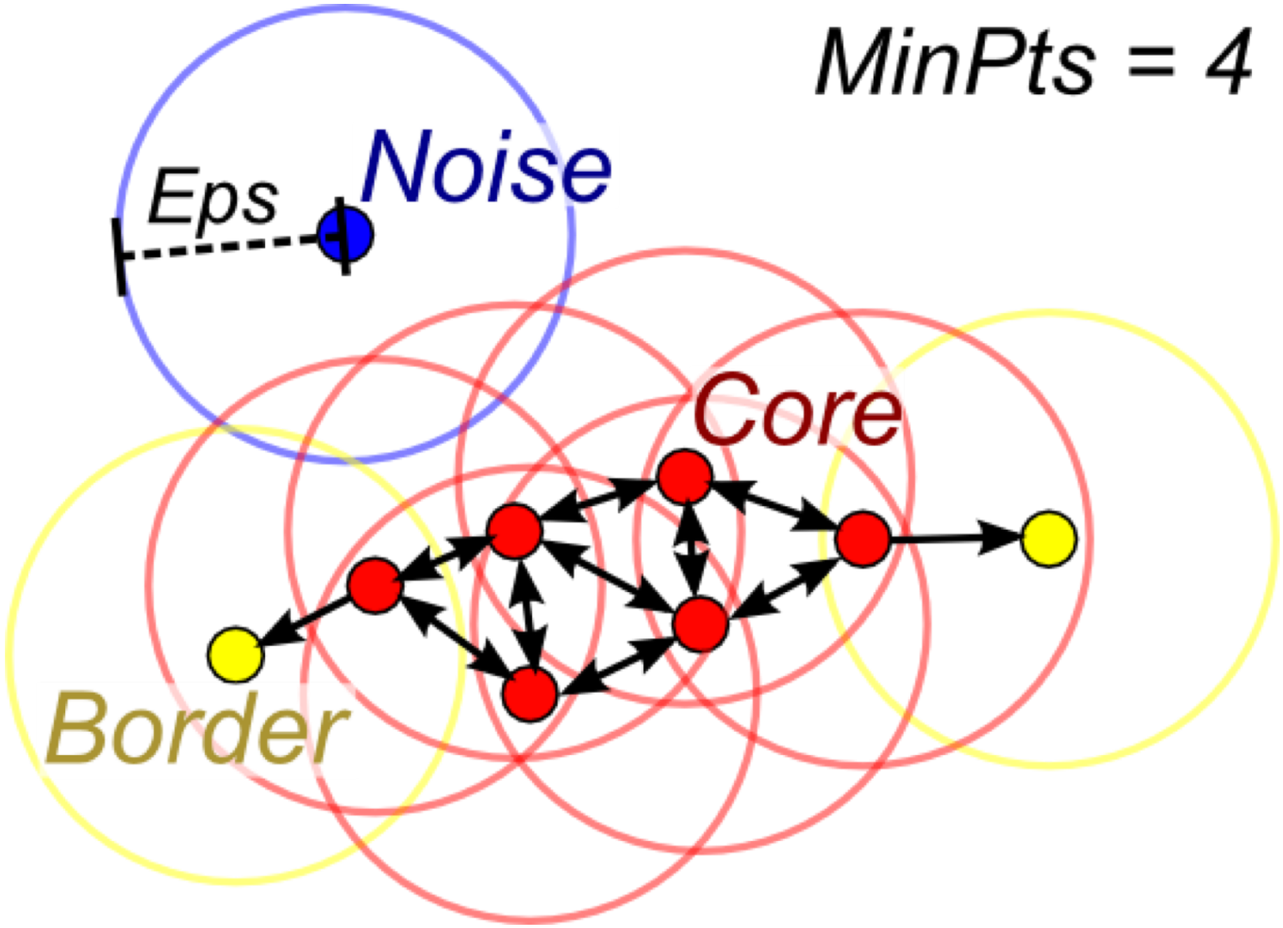
3. Move through every point in the pseudo-cluster, search a distance of epsilon, and include all observations within `epsilon` in this pseudo-cluster.

4. Repeat 3. until we stop adding observations to the pseudo-cluster.

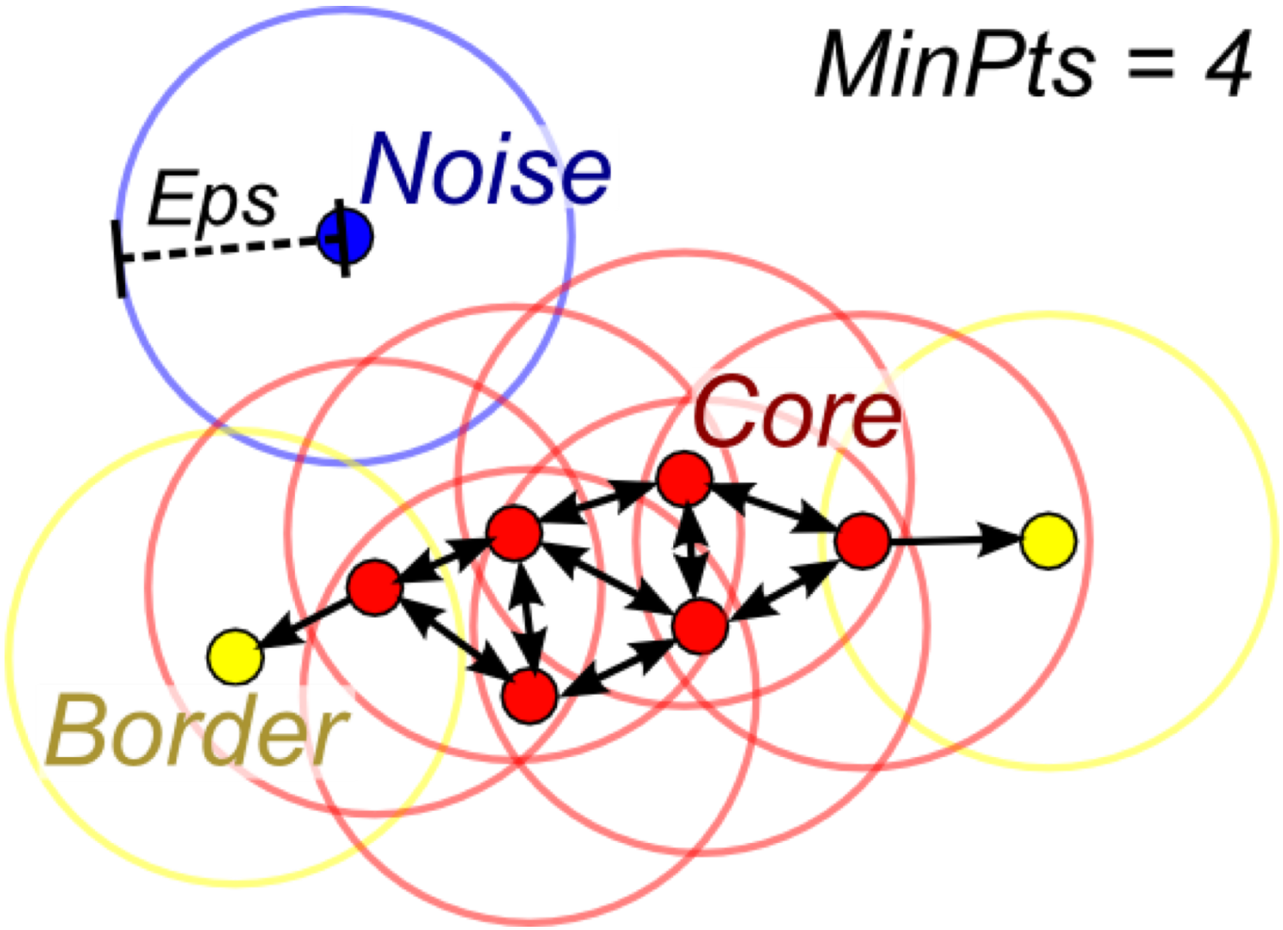
5. Compare the size of the pseudo-cluster to the `min_samples` threshold.
   - If the size of the pseudo-cluster is at least `min_samples`, great! Call that a cluster.
   - If not, disband the pseudo-cluster.

6. Return to step 2. Repeat until every point has been checked.

VISUALIZING DBSCAN



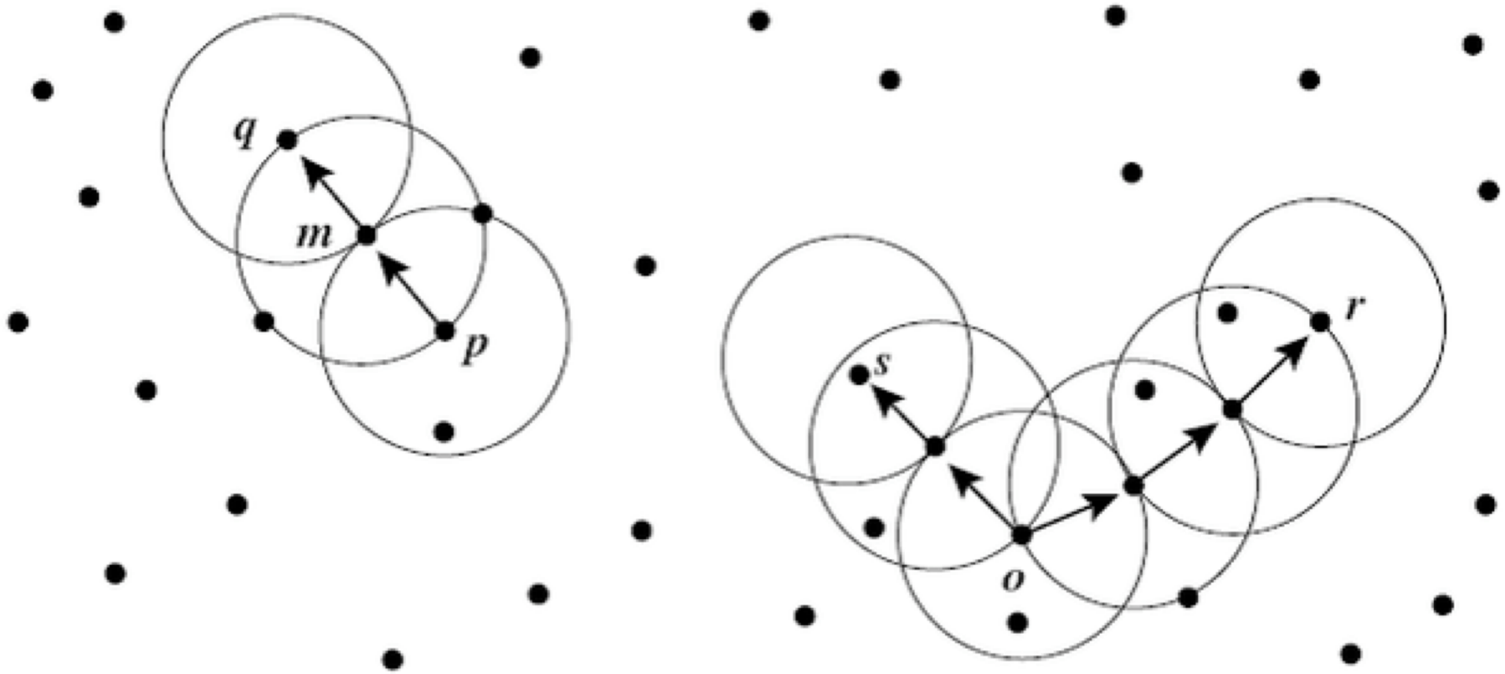
VISUALIZING DBSCAN



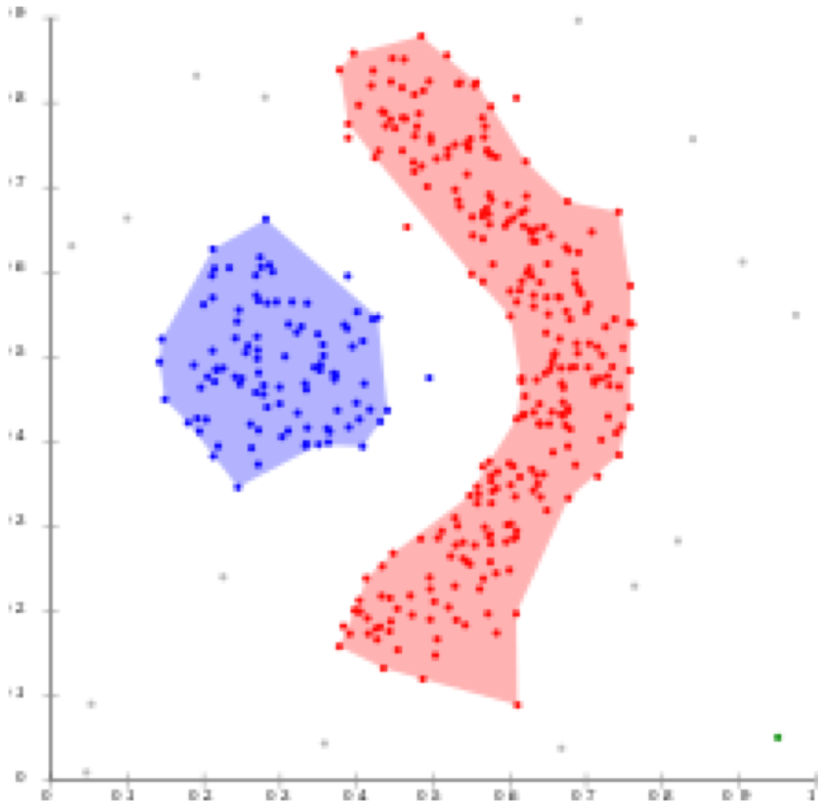
- **Core points**: Points inside a cluster that have at least `min_samples` points within `epsilon`.
- **Border points**: Points inside a cluster that <u>do not</u> have at least `min_samples` points within `epsilon`.
- **Noise:** Points that belong to no cluster.

WHY DBSCAN?

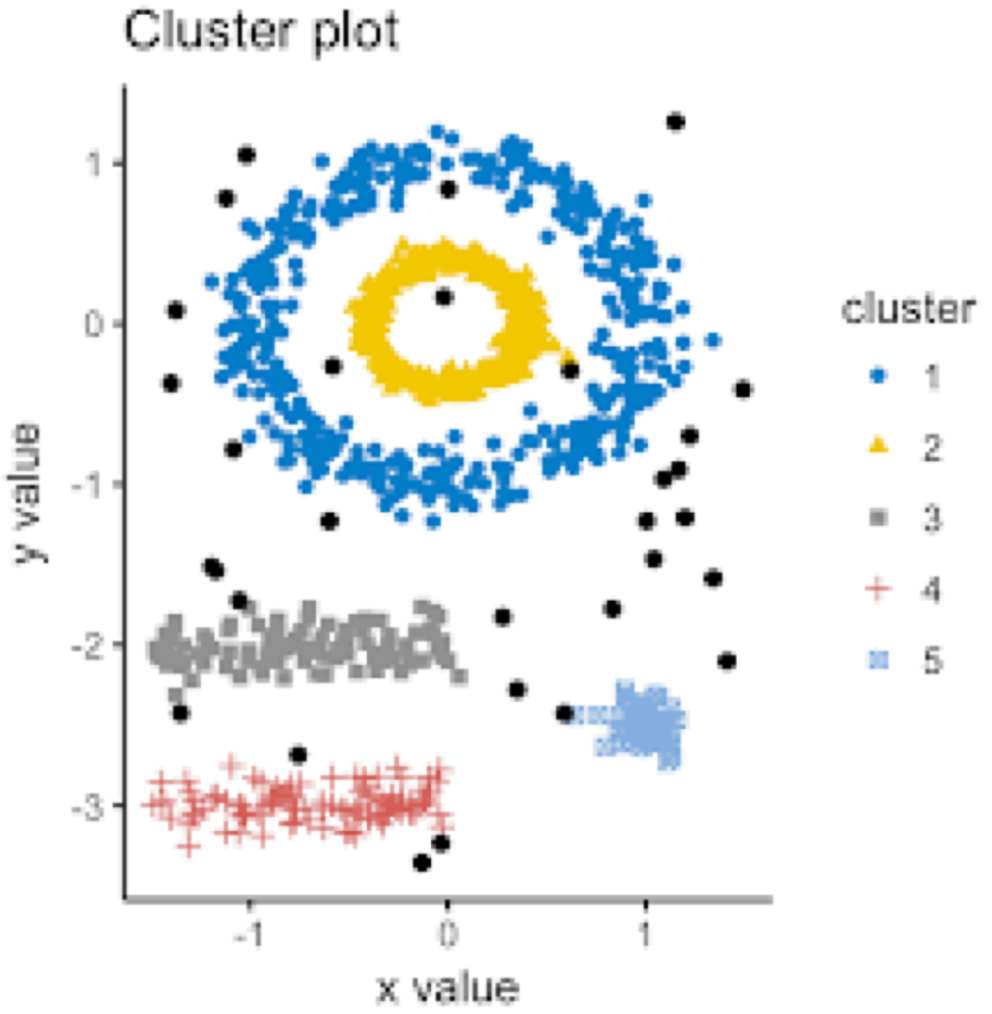
- DBSCAN allows us to detect some cluster patterns that k-Means and Hierarchical clustering might not be able to detect.



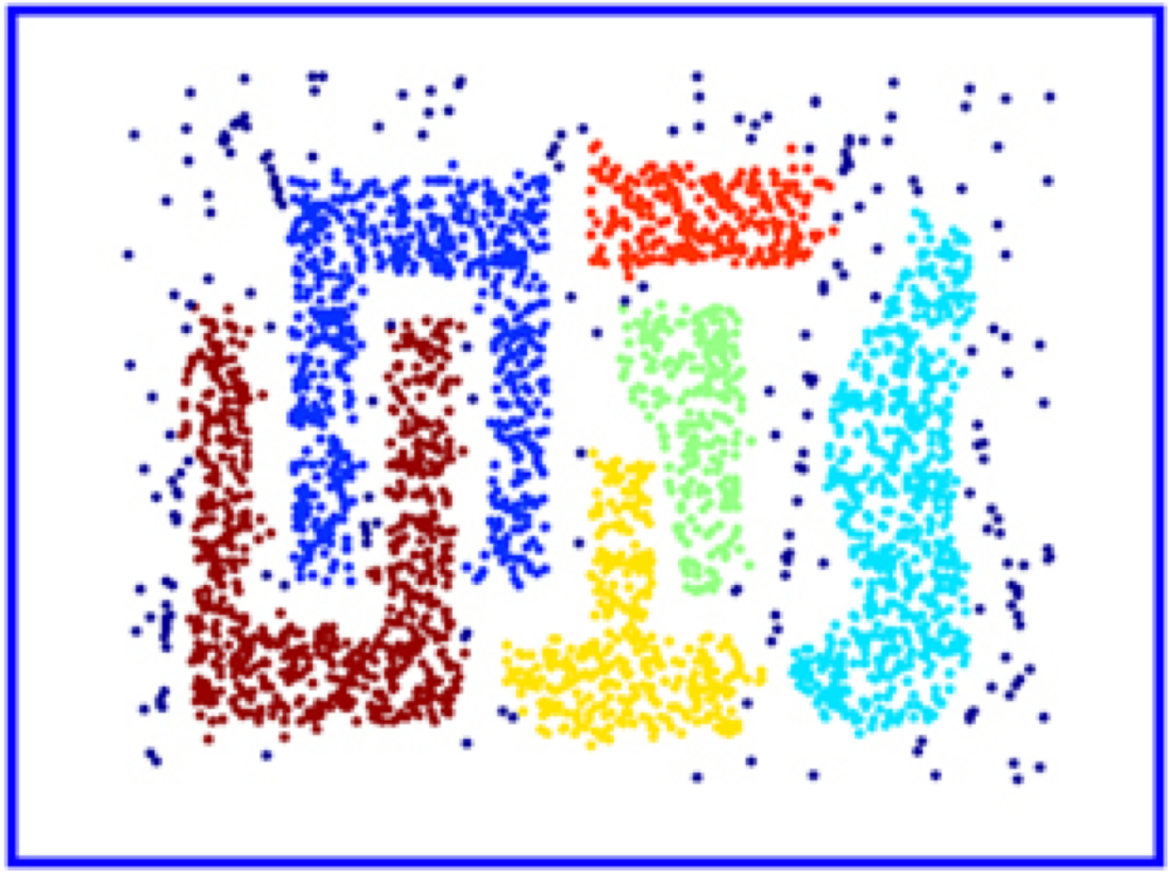
WHY DBSCAN?



WHY DBSCAN?



WHY DBSCAN?



WHY DBSCAN?

- DBSCAN allows us to detect some cluster patterns that k-Means and Hierarchical clustering might not be able to detect.

- We don’t need to pre-specify the number of clusters; the algorithm will determine how many clusters are appropriate given fixed `min_samples` and `epsilon` values.
  - This is particularly valuable when we are clustering data in more than two or three dimensions.

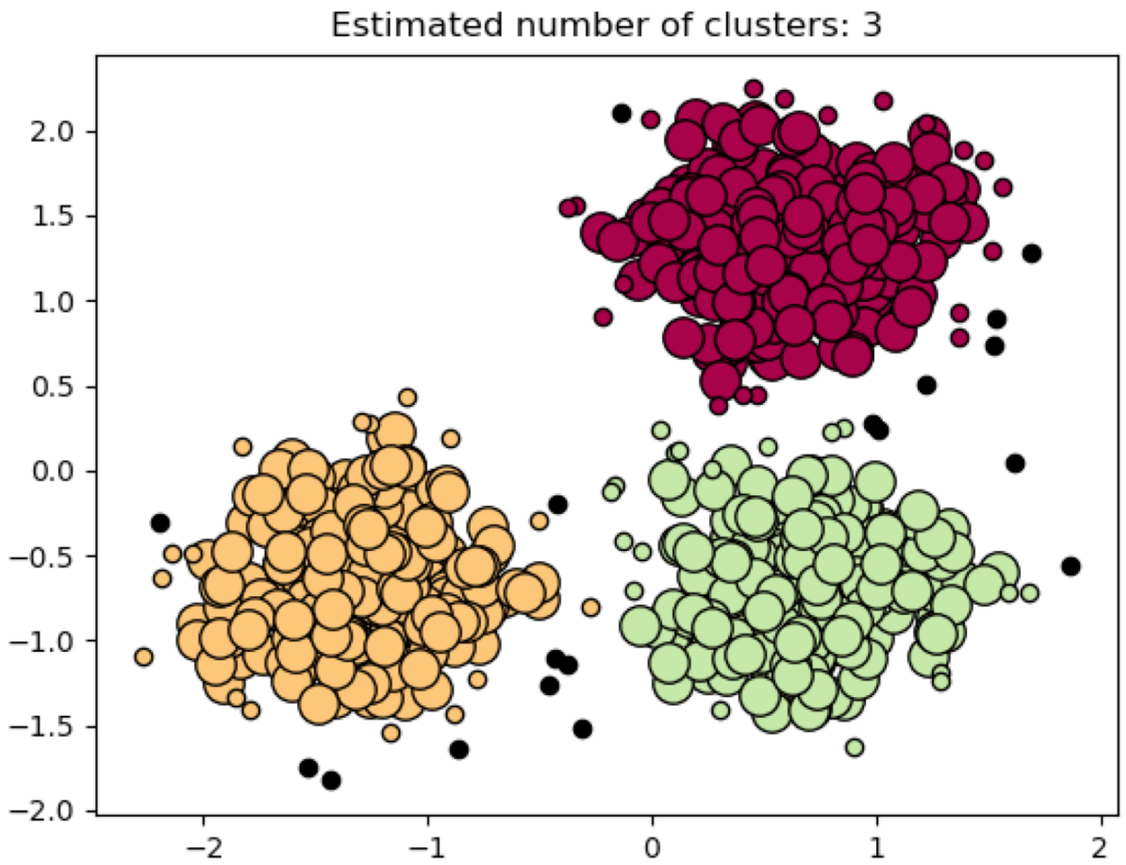
WHY DBSCAN?

- DBSCAN allows us to detect some cluster patterns that k-Means and Hierarchical clustering might not be able to detect.

- We don’t need to pre-specify the number of clusters; the algorithm will determine how many clusters are appropriate given fixed `min_samples` and `epsilon` values.
  - This is particularly valuable when we are clustering data in more than two or three dimensions.

- Not every point is clustered!
  - Good for **identifying outliers**.

WHY DBSCAN?

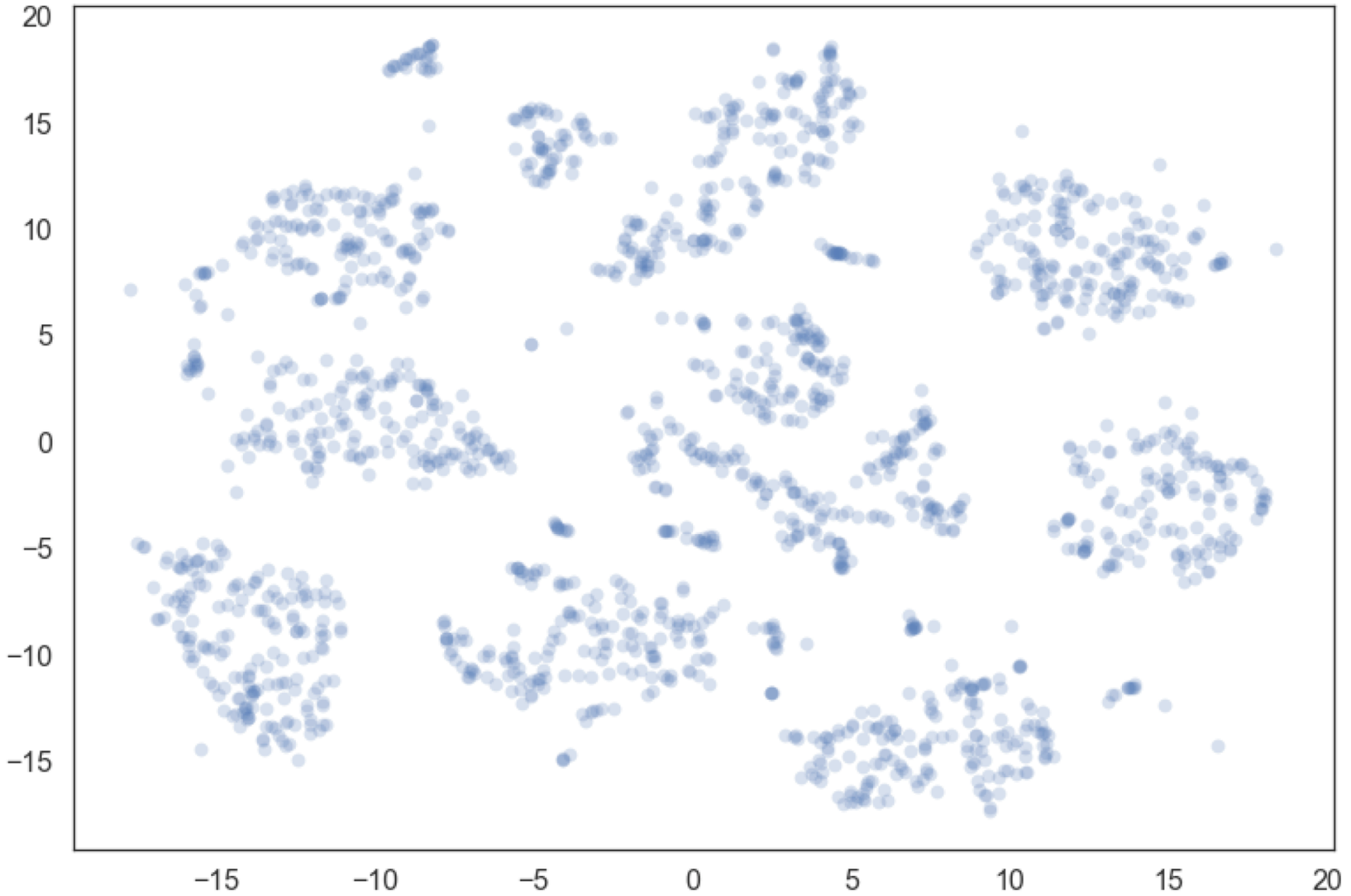


DISADVANTAGES OF DBSCAN

- DBSCAN requires us to tune two parameters.

- DBSCAN works well when clusters are of a different density than the overall data, but does not work well when the clusters themselves are of varying density.
  - Fixed `epsilon`.

DISADVANTAGES OF DBSCAN

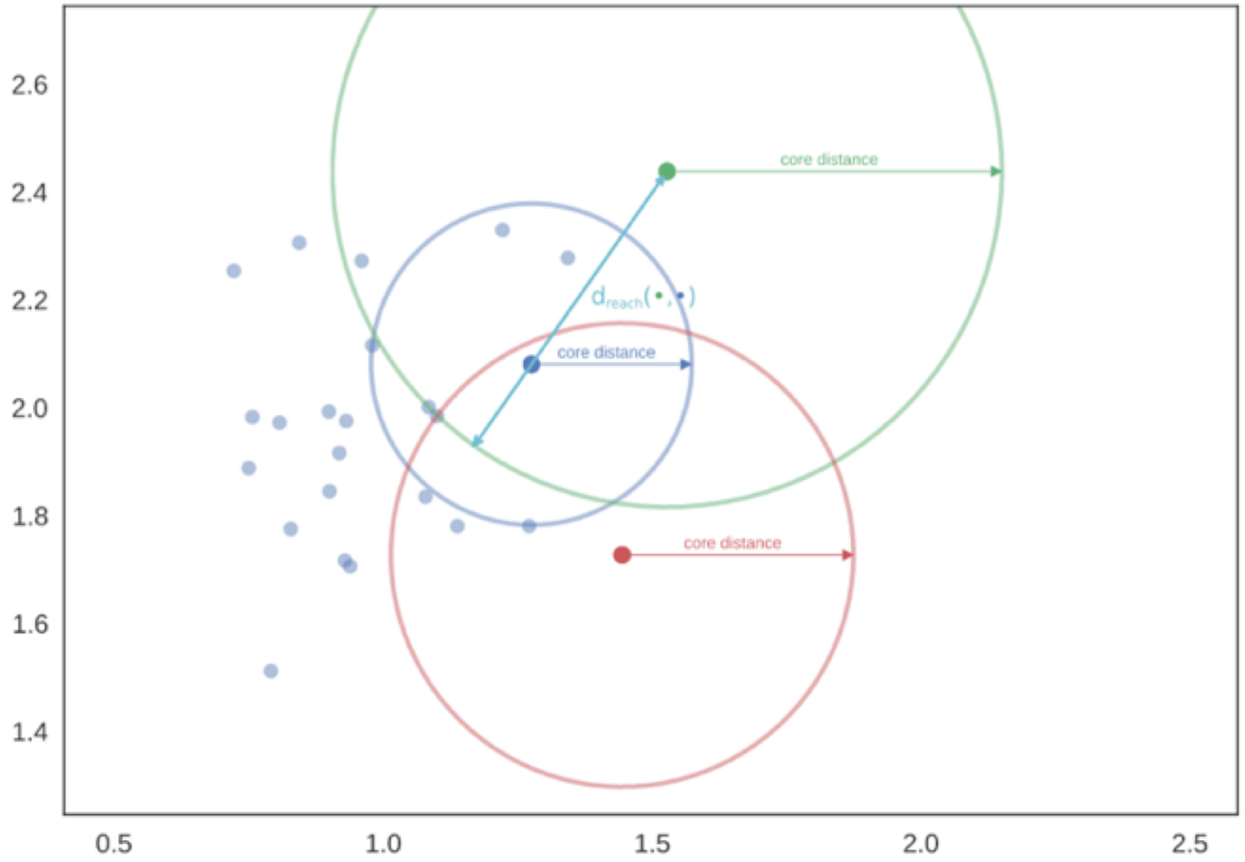


HDBSCAN

- **HDBSCAN** is a tweak on DBSCAN that will allow us to also check for clusters of varying density.
  - **Hierarchical DBSCAN**.

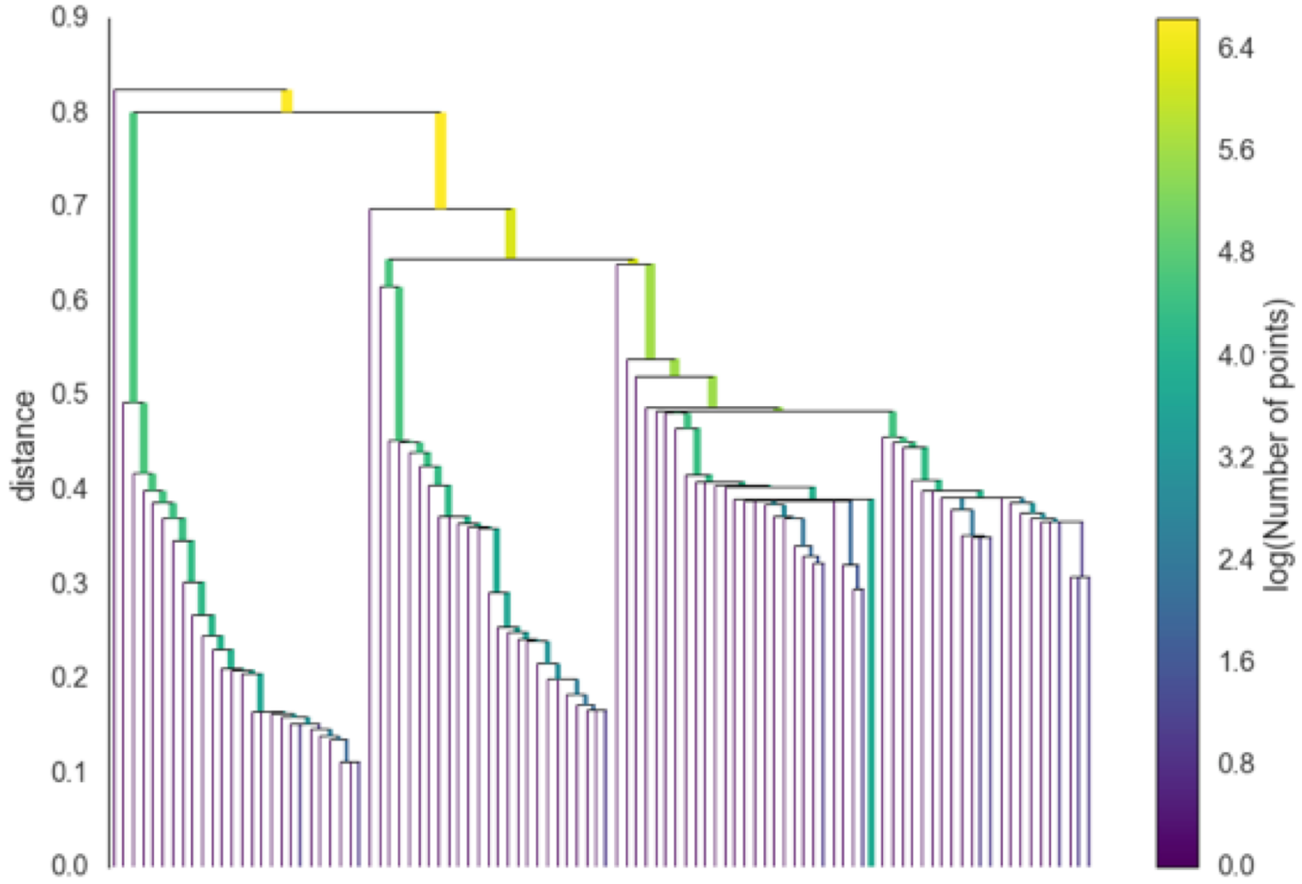
  - We combine DBSCAN with hierarchical clustering by using `epsilon` as our distance metric for our dendrogram.

HDBSCAN



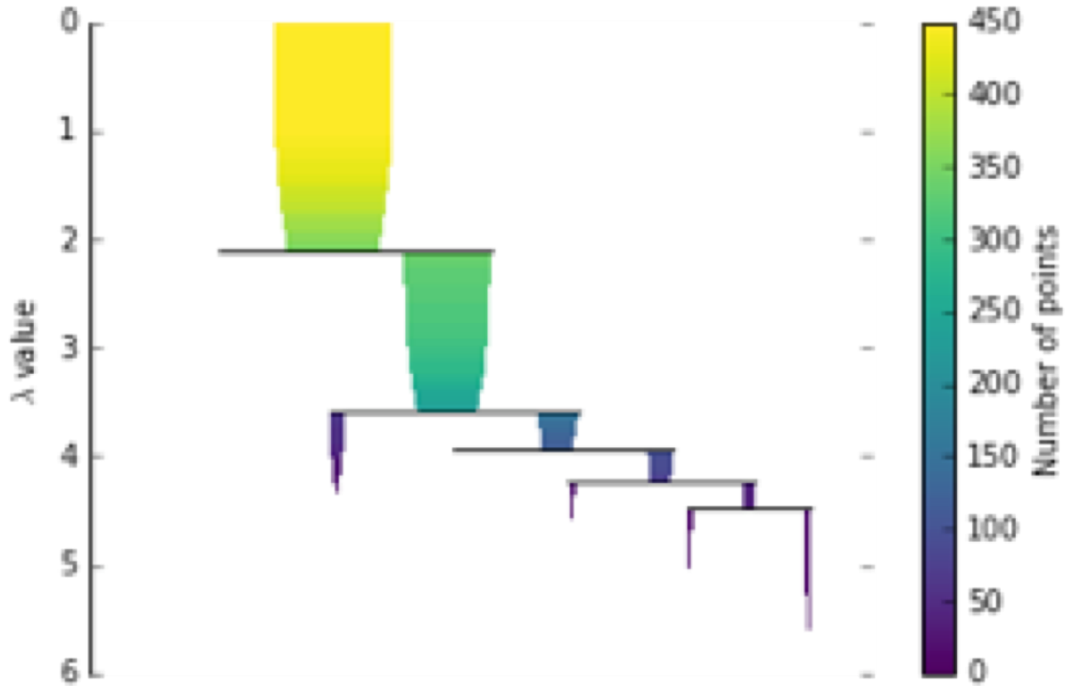
- For each point, we calculate how large `epsilon` would need to be in order to meet the `min_samples` threshold.
- We call this the “core distance” for each point.

HDBSCAN



- The algorithm then plots these potential clusters in a dendrogram by plotting all values of `epsilon`.

HDBSCAN



- The algorithm then “prunes” the dendrogram based on `min_samples`.

HDBSCAN

- HDBSCAN only requires us to specify `min_samples`; we do not need to identify `epsilon`. (The algorithm varies `epsilon` for us.)

- HDBSCAN is substantially faster than the `sklearn` implementation of DBSCAN or *k*-Means.
  - http://hdbscan.readthedocs.io/en/latest/performance_and_scalability.html#comparison-of-high-performance-implementations