

DBSCAN & HDBSCAN

*Matt Brems
Data Science Immersive, GA*

DBSCAN & HDBSCAN

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

→ no y variable as part of training data

- 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?

k -means

- + pretty fast
- + quickly ID clusters
- + control k
- performs poorly on certain data
- random instantiation of centroid impacts our results

hierarchical

- + don't need to specify # of clusters at outset
- + dendrogram is a useful visualization
- computation time
- not great for bigger datasets

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

↳ outliers

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. → areas with lots of data
 - We avoid clustering in low-density areas.
- ↳ areas where data is sparse

DBSCAN

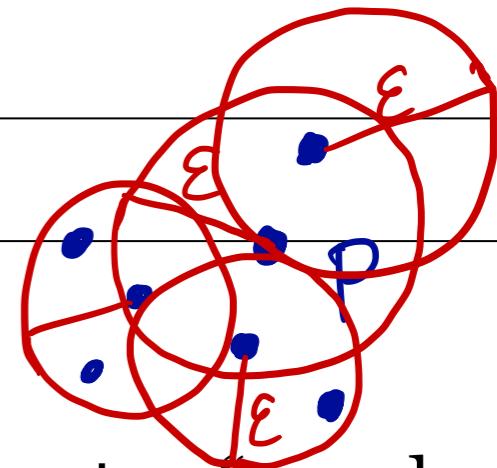
- DBSCAN requires you to specify two hyperparameters:
 - **min_samples**: the minimum number of points needed to form a cluster.
 - **epsilon**: the “searching” distance when attempting to build a cluster.

VISUALIZING DBSCAN

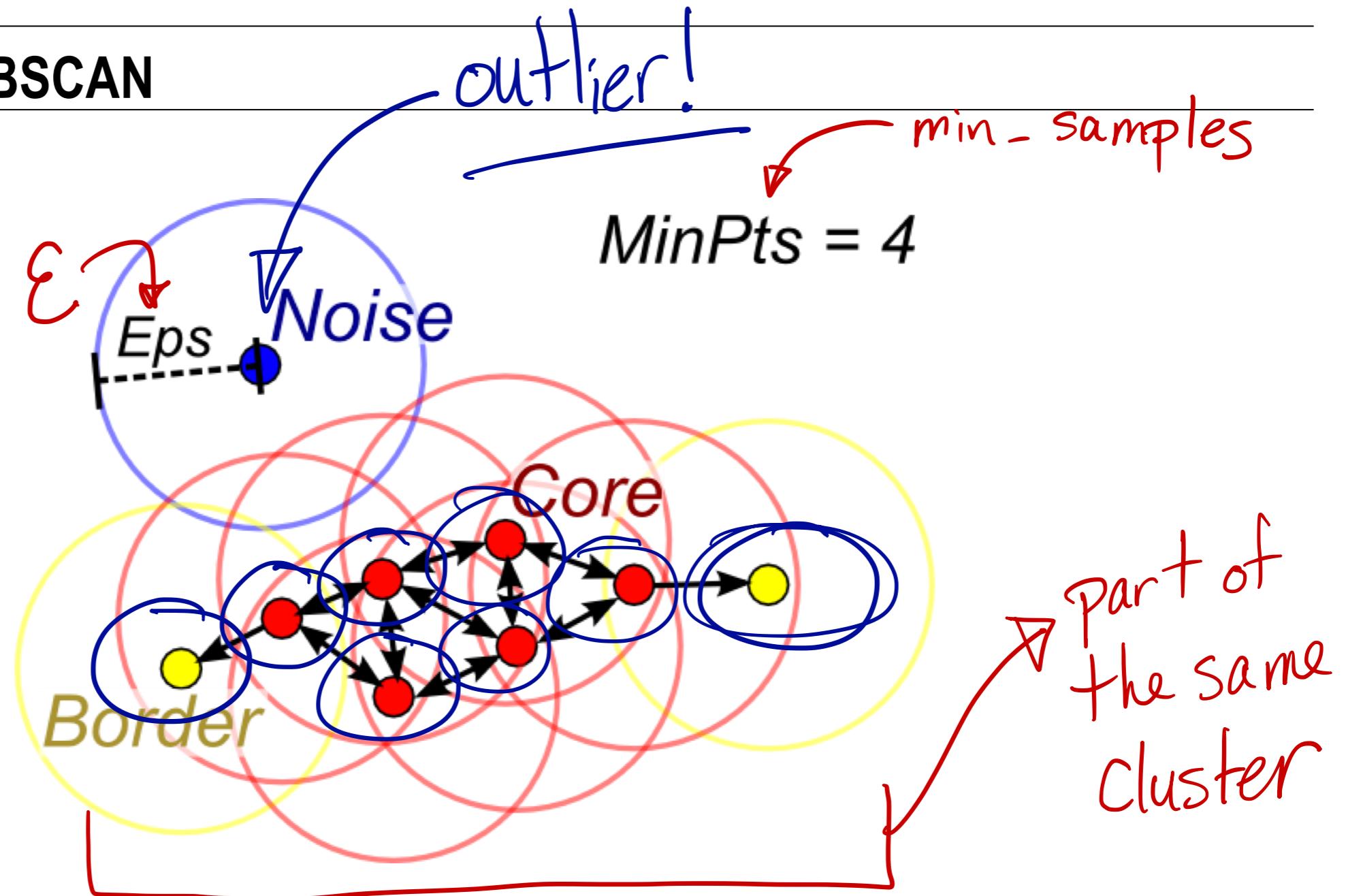
- <https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/>

HOW DOES DBSCAN WORK?

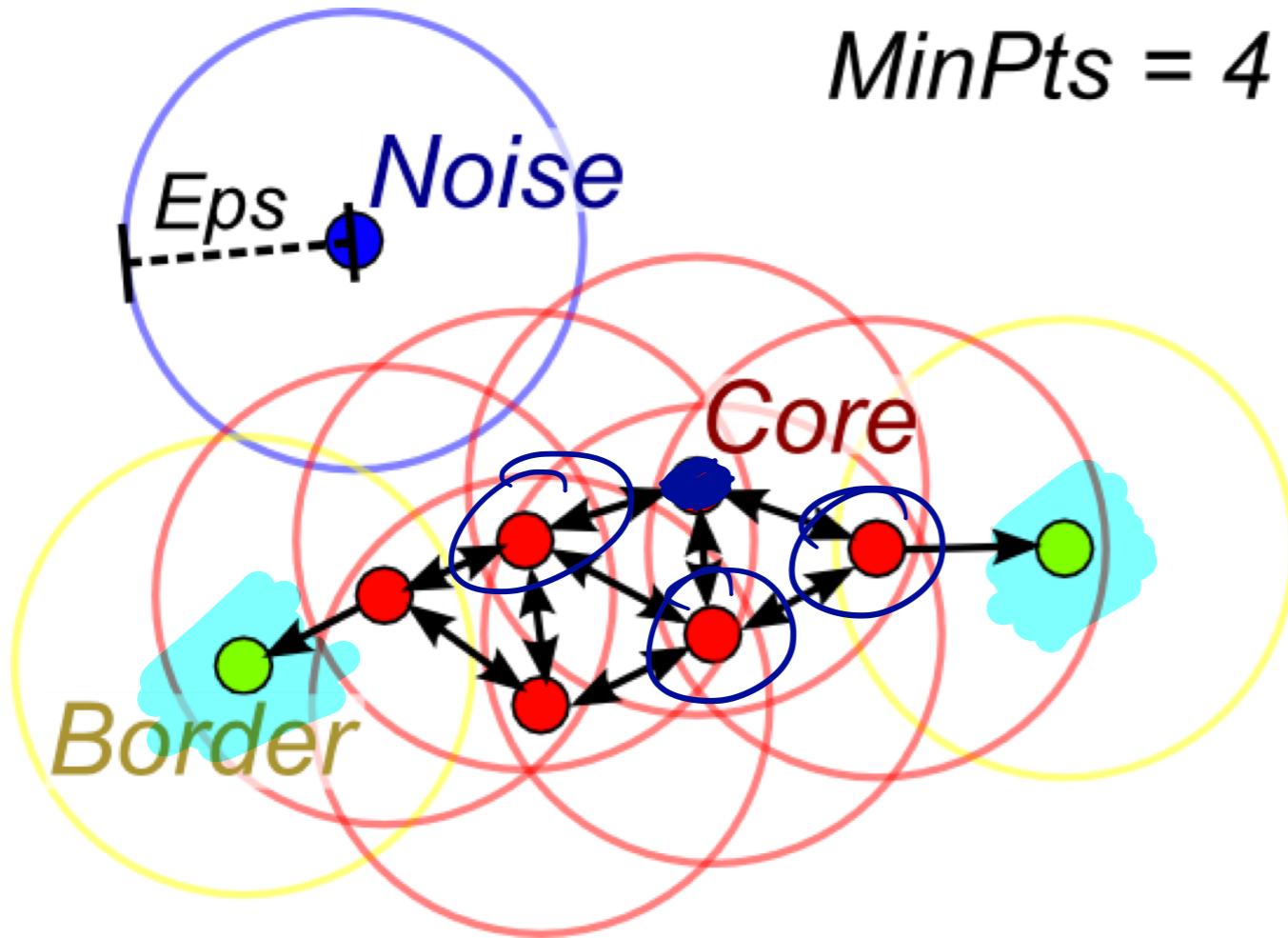
- ϵ , min_samples
1. Choose your hyperparameters.
 2. Pick a random, un-clustered observation p to start and create a “pseudo-cluster” that consists of p and all observations within ϵ of p .
 3. Move through every point in the pseudo-cluster, search a distance of ϵ , and include all observations within ϵ 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



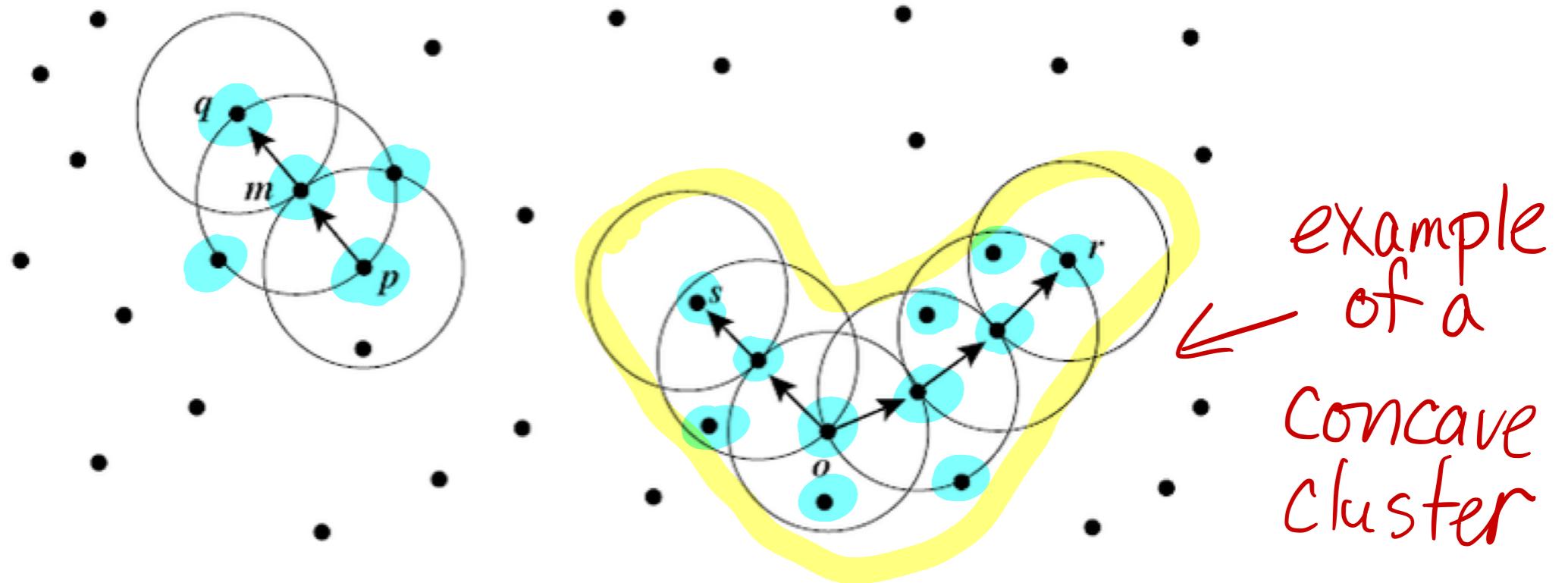
VISUALIZING DBSCAN



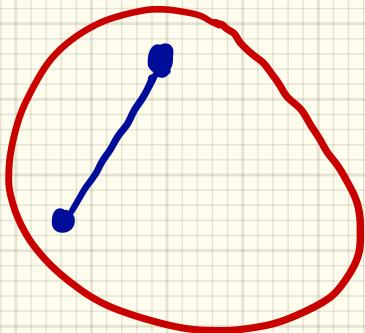
- **Core points:** Points inside a cluster that have at least `min_samples` points within `epsilon`.
- **Border points:** Points inside a cluster that do not have at least `min_samples` points within `epsilon`.
- **Noise:** Points that belong to no cluster.

WHY DBSCAN?

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

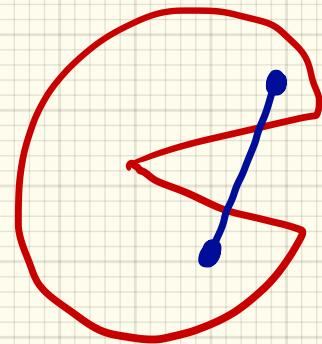


CONVEX



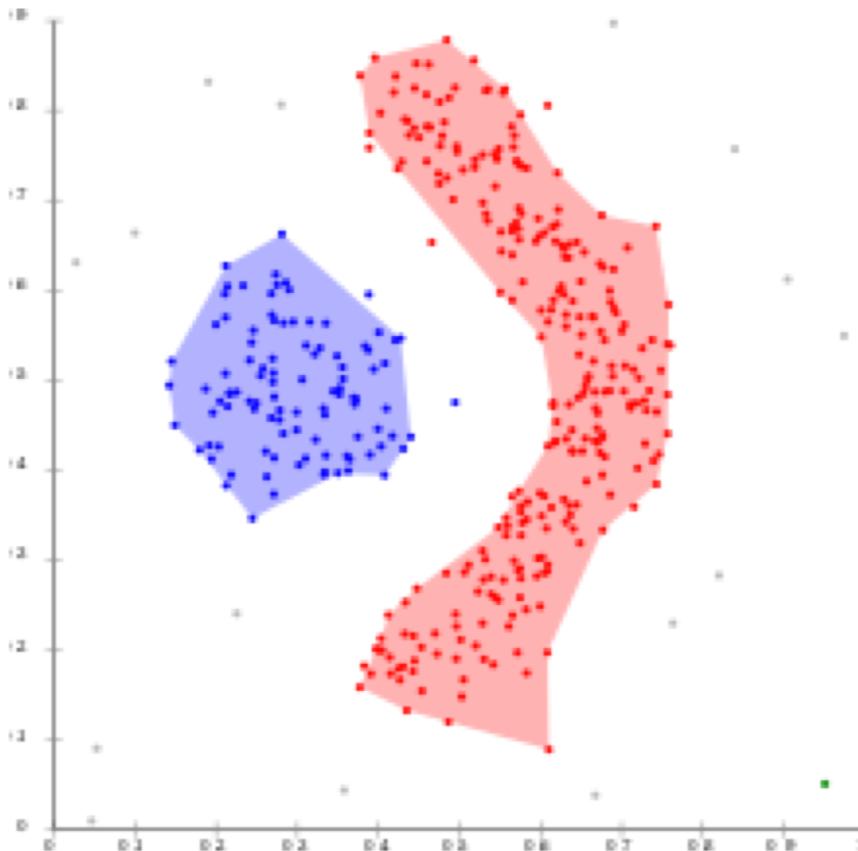
k-means clusters
must be convex.

CONCAVE

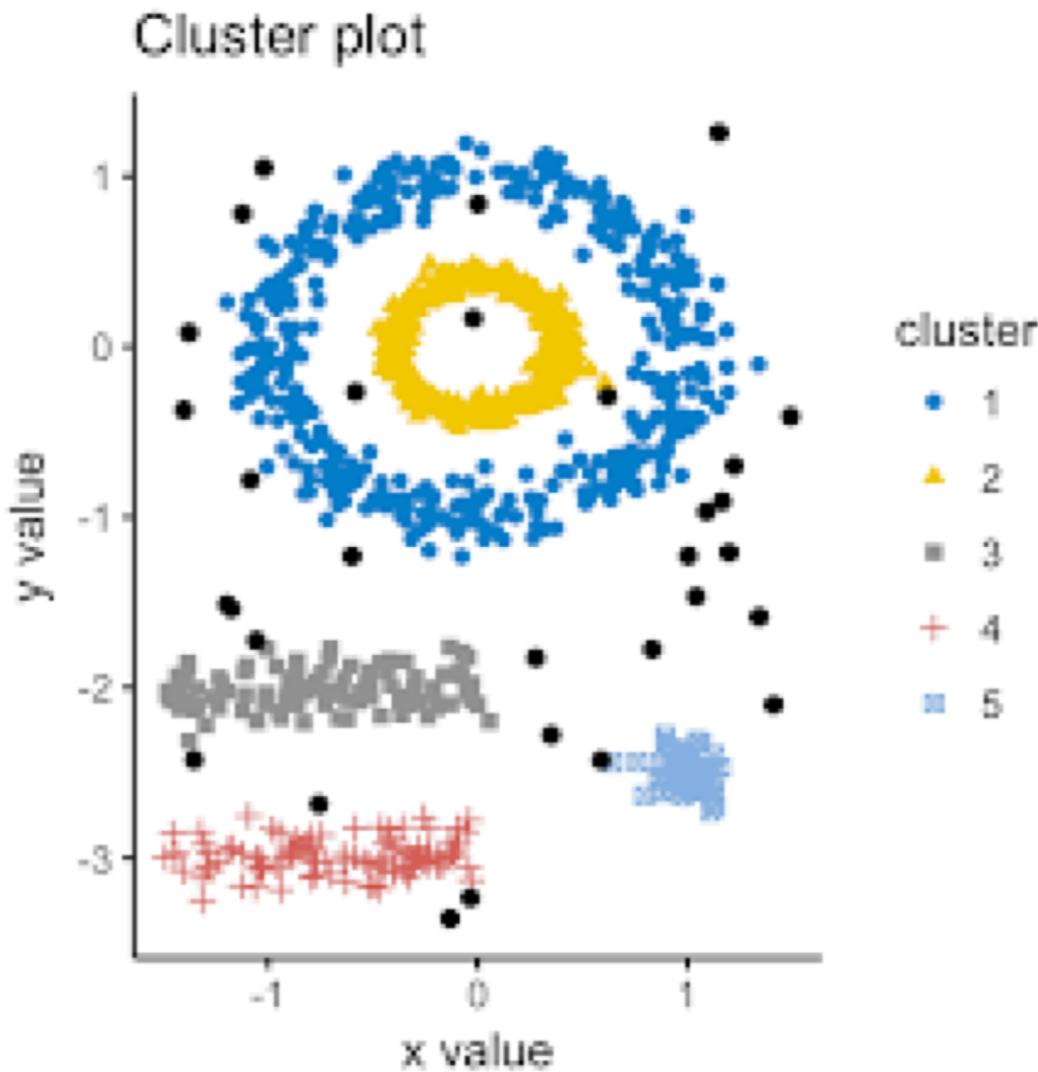


DBSCAN clusters
can be concave.

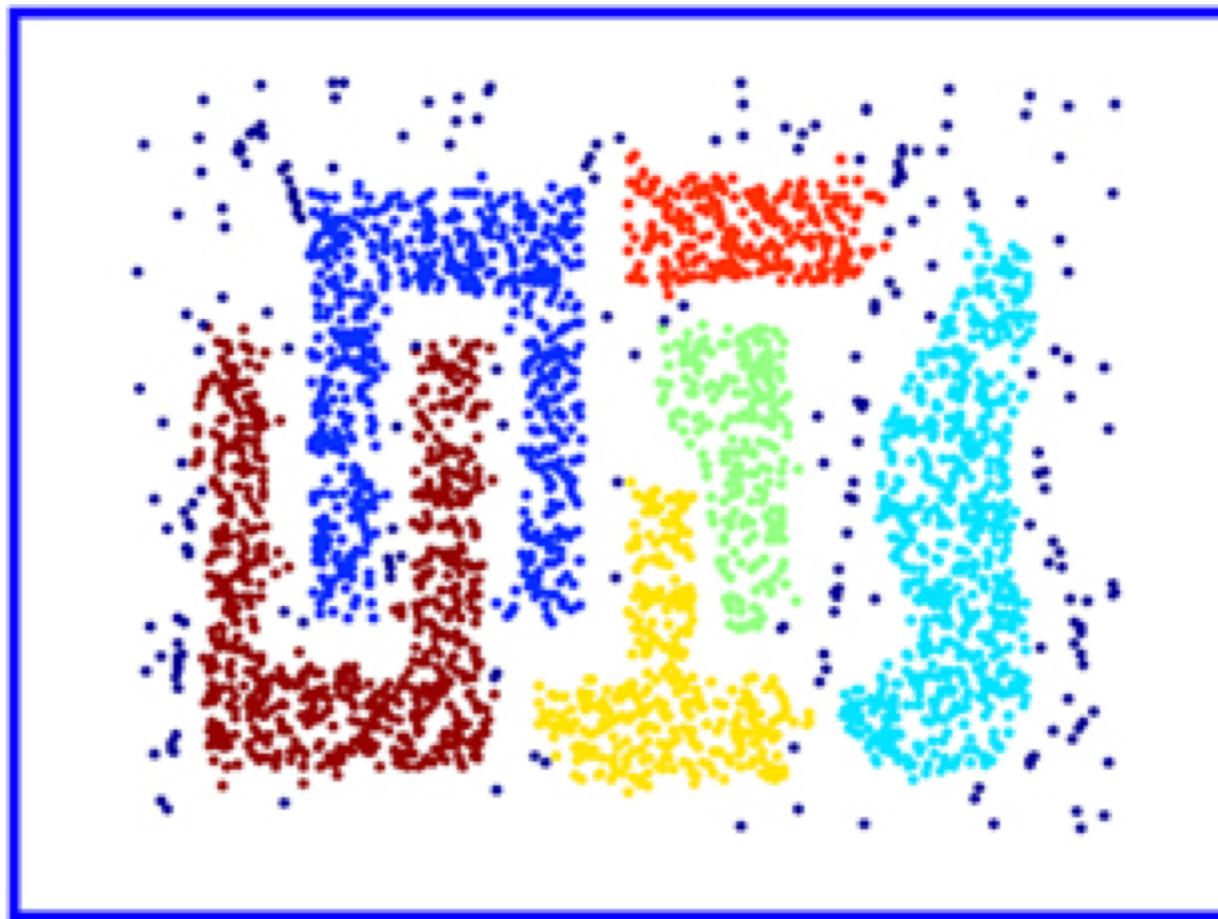
WHY DBSCAN?



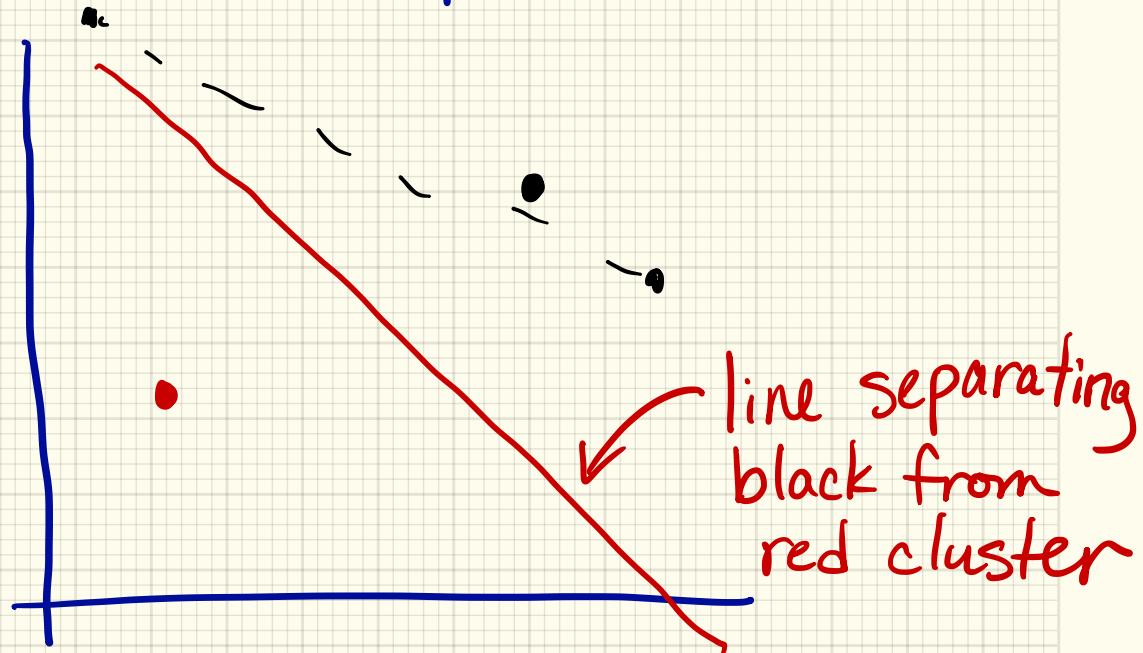
WHY DBSCAN?



WHY DBSCAN?



k-means example



WHY DBSCAN?

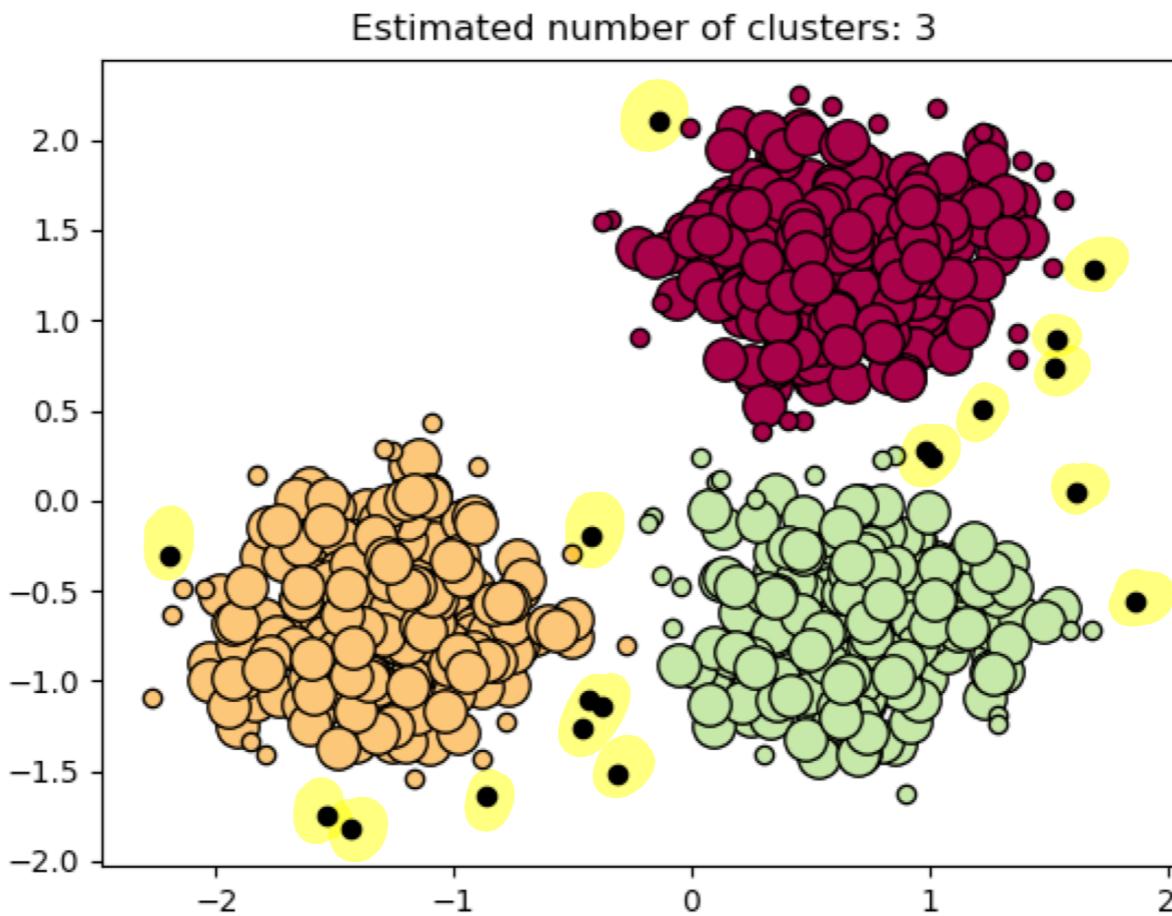
- DBSCAN allows us to detect some cluster patterns that k-Means and Hierarchical clustering might not be able to detect.
- We don't need to pre-specify the number of clusters; the algorithm will determine how many clusters are appropriate given fixed `min_samples` and `epsilon` values.
 - This is particularly valuable when we are clustering data in more than two or three dimensions.

↳ This is also valuable when we do not have subject-matter expertise to define a specific number of clusters ahead of time.

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?

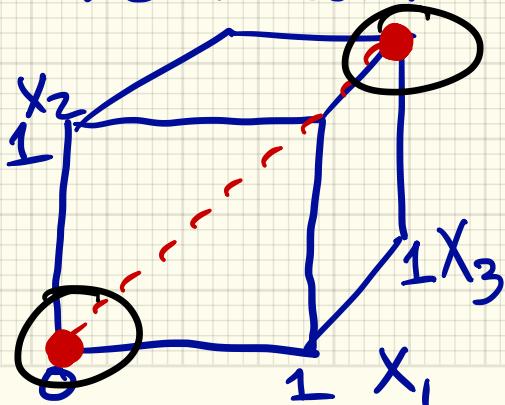


Euclidean distance

$$d(A, B) = \sqrt{(x_1^A - x_1^B)^2 + (x_2^A - x_2^B)^2 + \dots + (x_p^A - x_p^B)^2}$$

Suppose that each variable is scaled using Min Max Scaler

→ each column is bound b/w 0 and 1.



$$= \sqrt{\sum_{i=1}^p 1} = \sqrt{p}$$

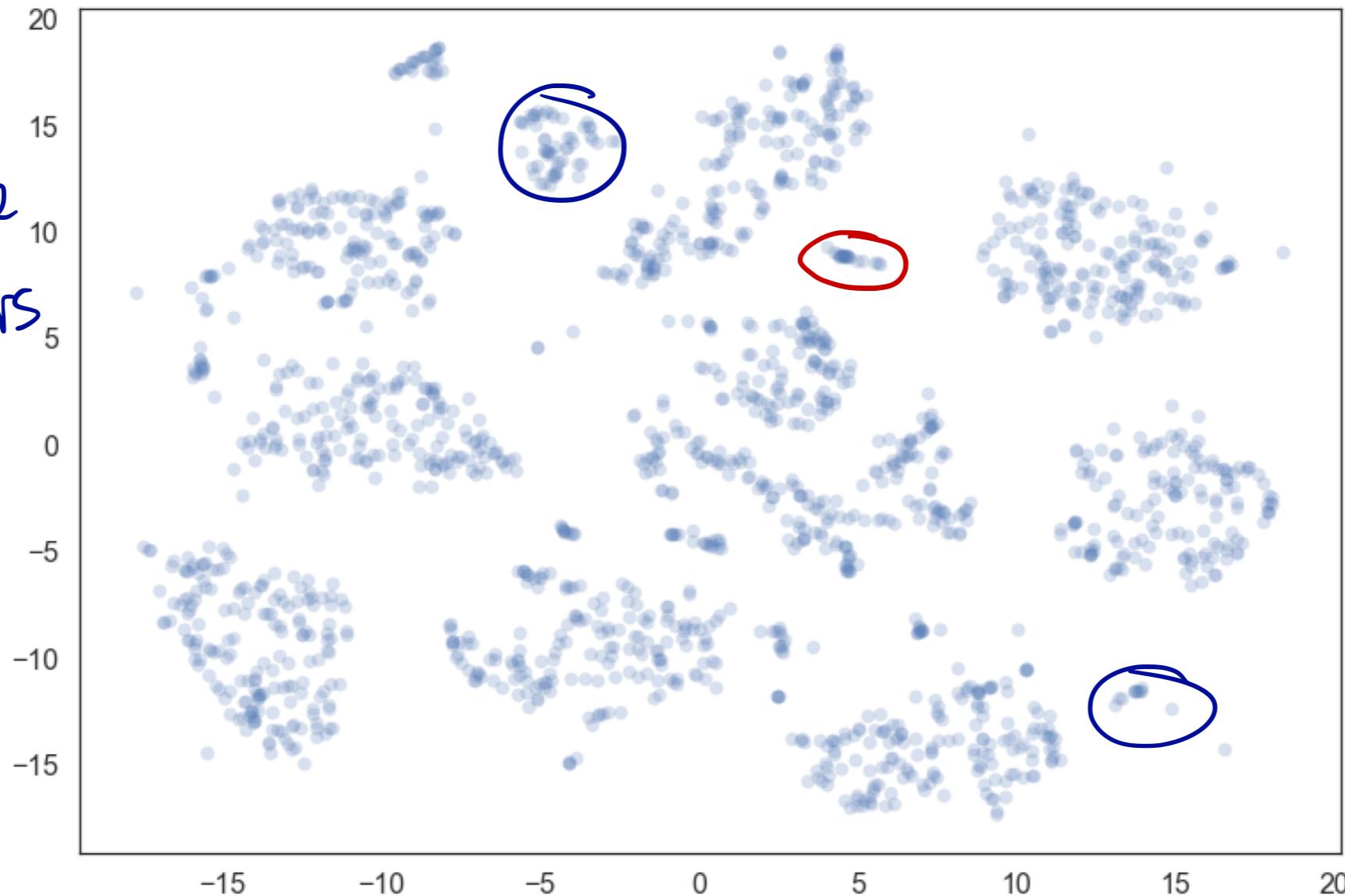
max dist.
in p dimensions
if MinMax Scaled.

DISADVANTAGES OF DBSCAN

- DBSCAN requires us to tune two parameters. ϵ , min-samples
- 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

Some clusters are more sparse than others

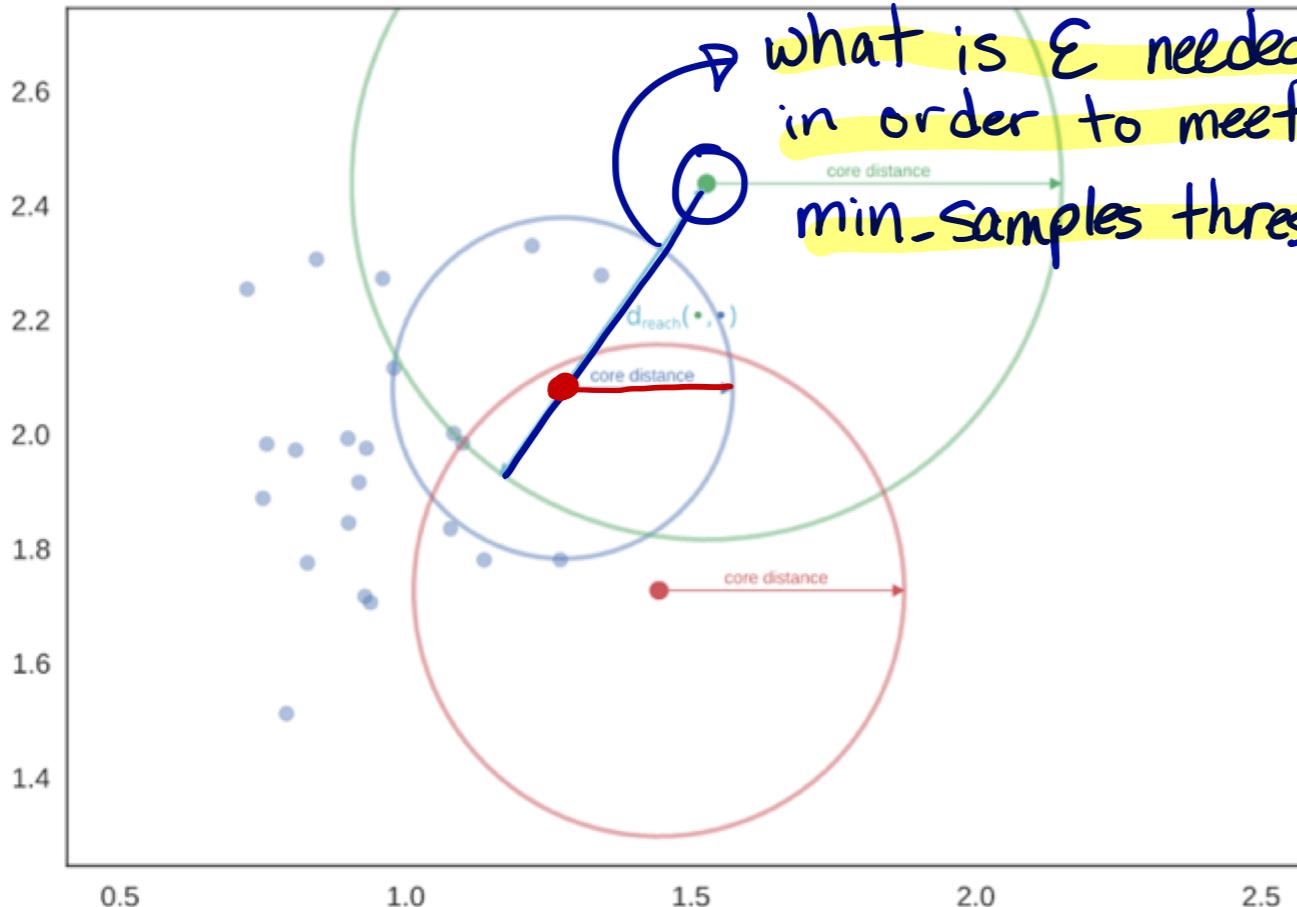


HDBSCAN

- **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

min_samples

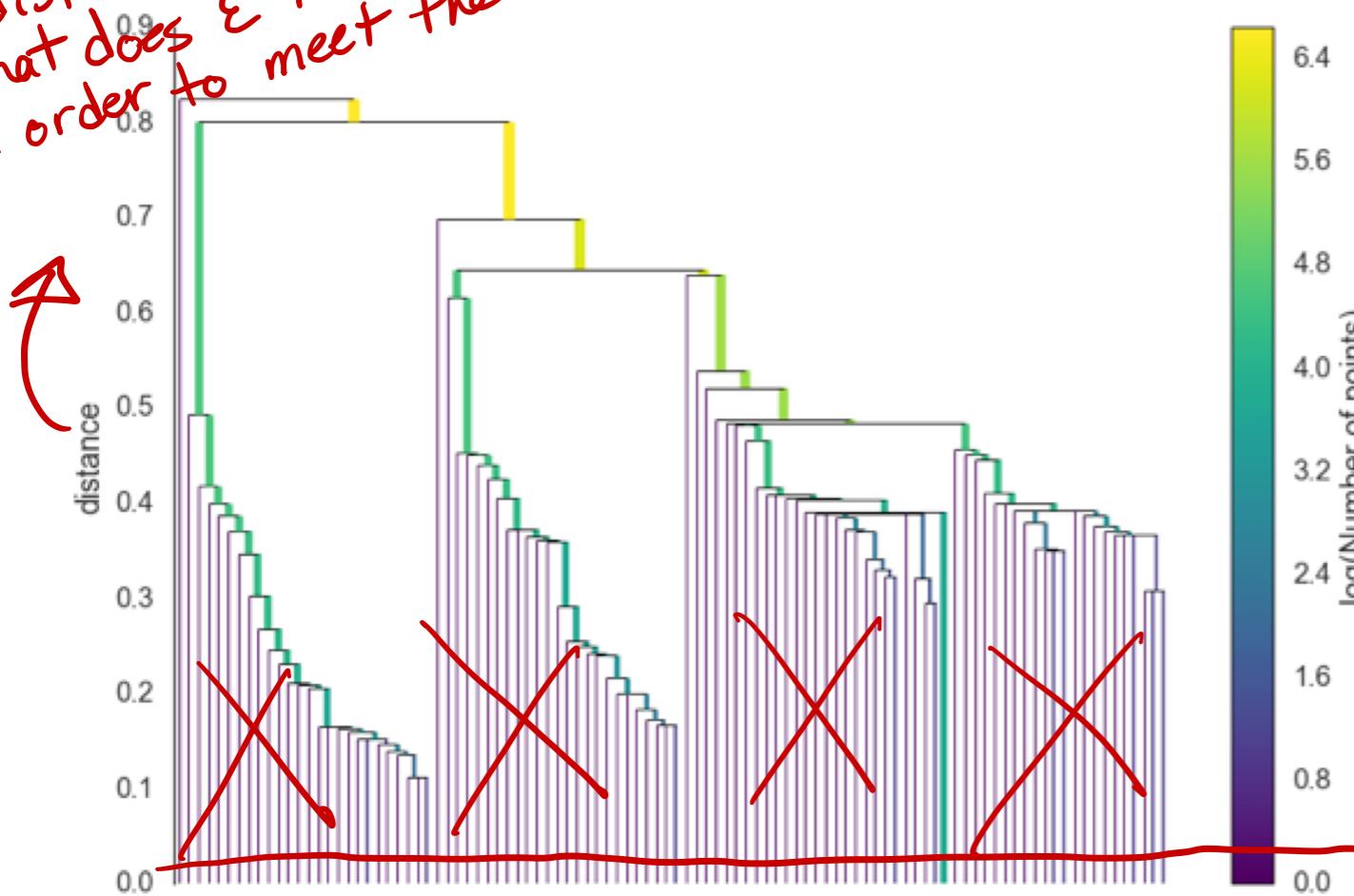


For each point, we calculate how large epsilon would need to be in order to meet the `min_samples` threshold.

- We call this the “core distance” for each point.

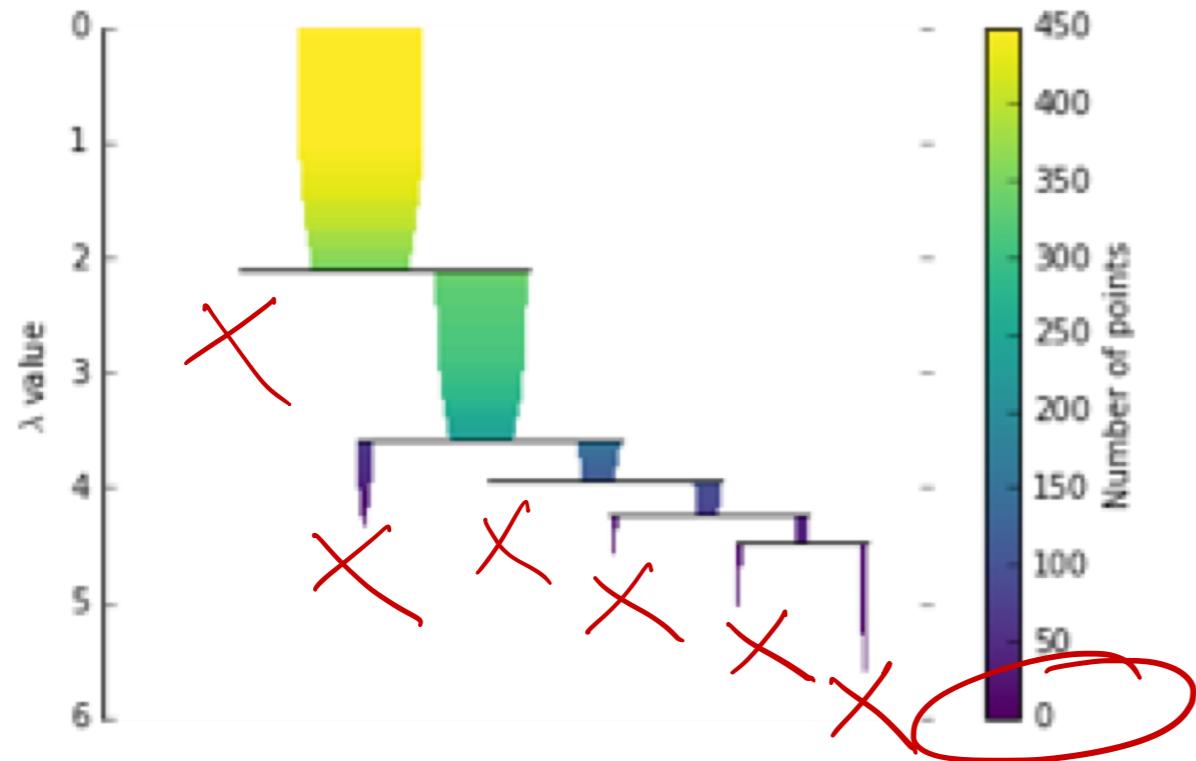
HDBSCAN

core distance, a.k.a what does ϵ need to be in order to meet the min-samples threshold?



- 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