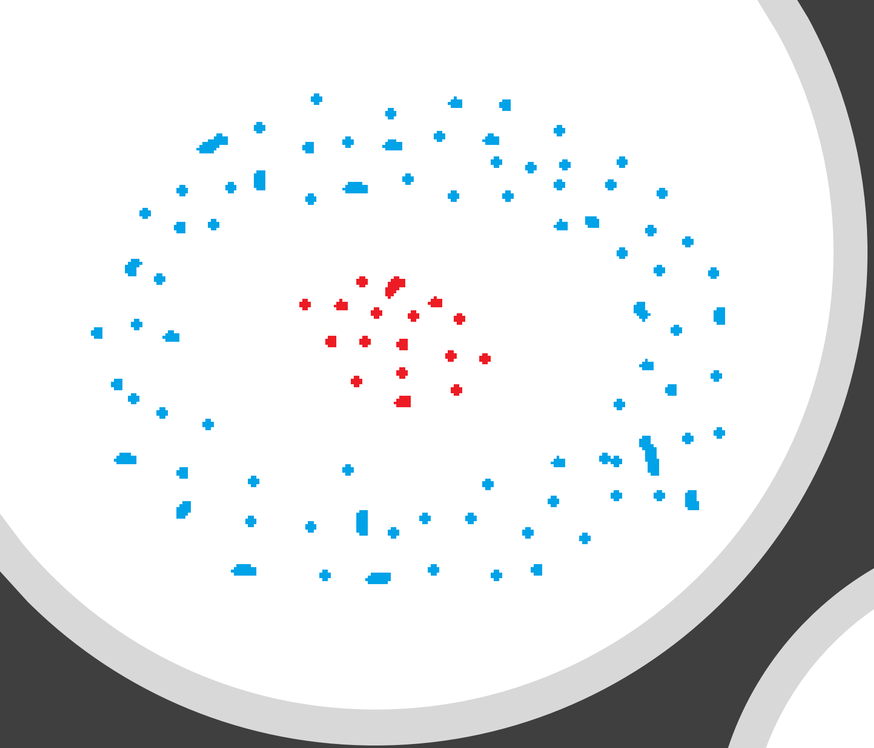
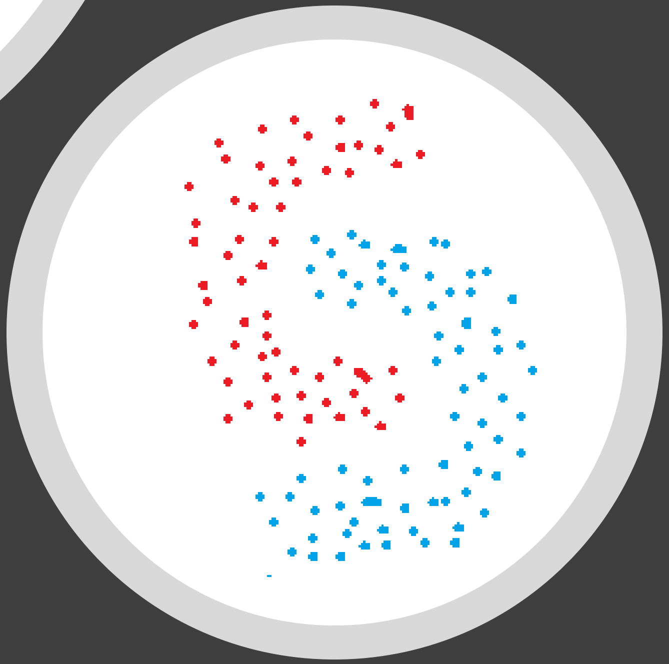
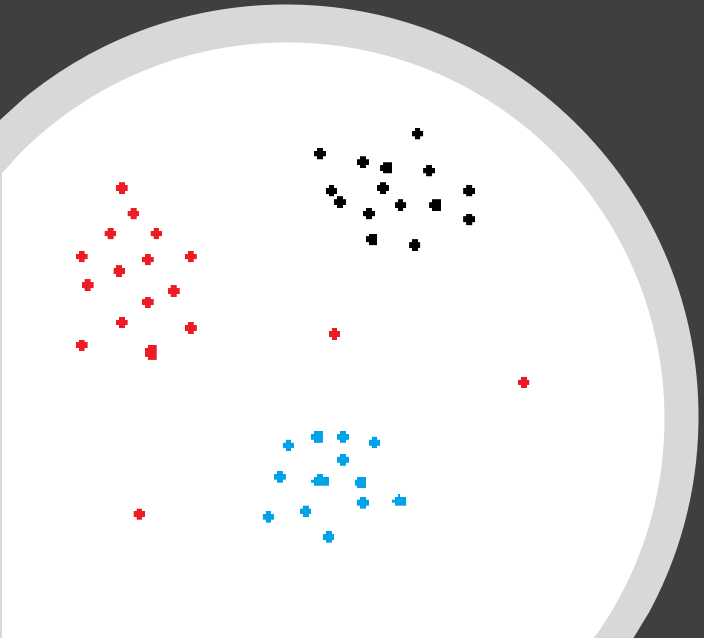
DBSCAN

Grace, Scott, Regina, Hang







Introduction to DBSCAN

- An unsupervised learning method, the algorithm tries to find the underlying structure of the data.
- Density-based spatial clustering of applications with noise.
- Clusters dataset based on distance between nearest points.
    - There must be a minimum number of points within that distance of each other to be considered a cluster.

Steps needed for data processing

Required:

- Standardization of values so that all features are on the same scale
- Missing value imputation/removal

Not required:

- Outlier mitigation, as DBSCAN is robust to outliers

Main Hyperparameters

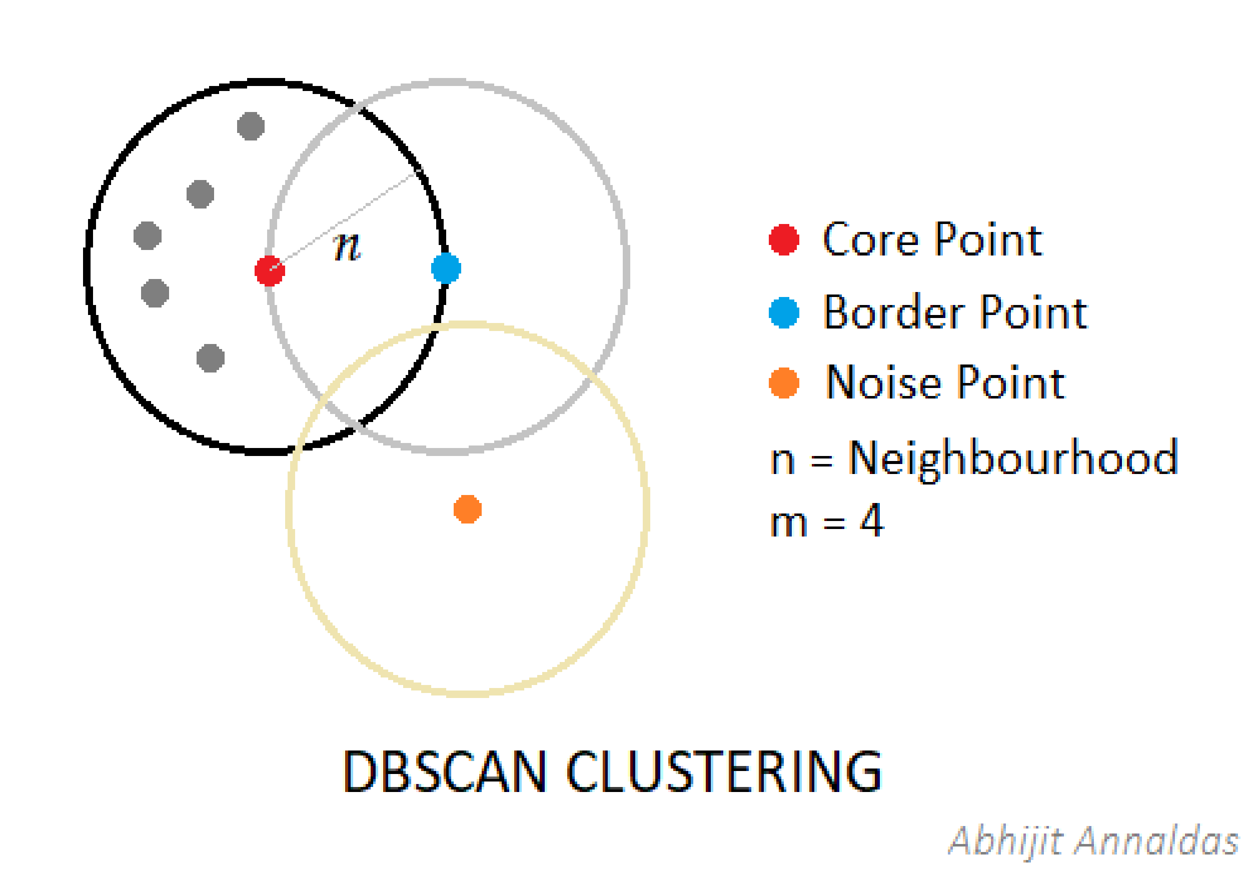
- **Epsilon (eps)**: the maximum distance between two points for one to be considered in the neighborhood of the other.
  - Typically chosen using nearest neighbor graph (k-graph), where k = minPts – 1. Prefer small value of epsilon.
- **MinPts (min_samples)**: the number of points (or samples) in a neighborhood for a point to be considered a core point. This includes the point itself.
  - Must be ≥ 3. As a rule of thumb, minPts = 2 * dimension.
- **Distance function (metric)**: need to be chosen appropriately for each dataset. By default, it is the Euclidean distance.

Other Hyperparameters

- **metric_params**: additional keyword arguments for metric (distance function)
- **algorithm**: the algorithm used by the NearestNeighbors module to compute distances and find neighborhood
- **leaf_size**: leaf size passed to BallTree or KDTree (nearest neighbor algorithms), which affects the speed of the query, and the memory needed to store the tree
- **p**: the power of the Minkowski metric (to calculate distance between points)
- **n_jobs**: the number of jobs to run (how many core processors to use)

Terms

- **Core point**: a data point is considered a core point if it has the minimum number of neighboring data points (minPts) at an epsilon distance from it.
- **Border point**: a data point that has less than the minimum number of data points (minPts) but has at least one core point in its neighborhood.
- **Noise point**: a data point that is not a core point or a border point is considered noise or an outlier.



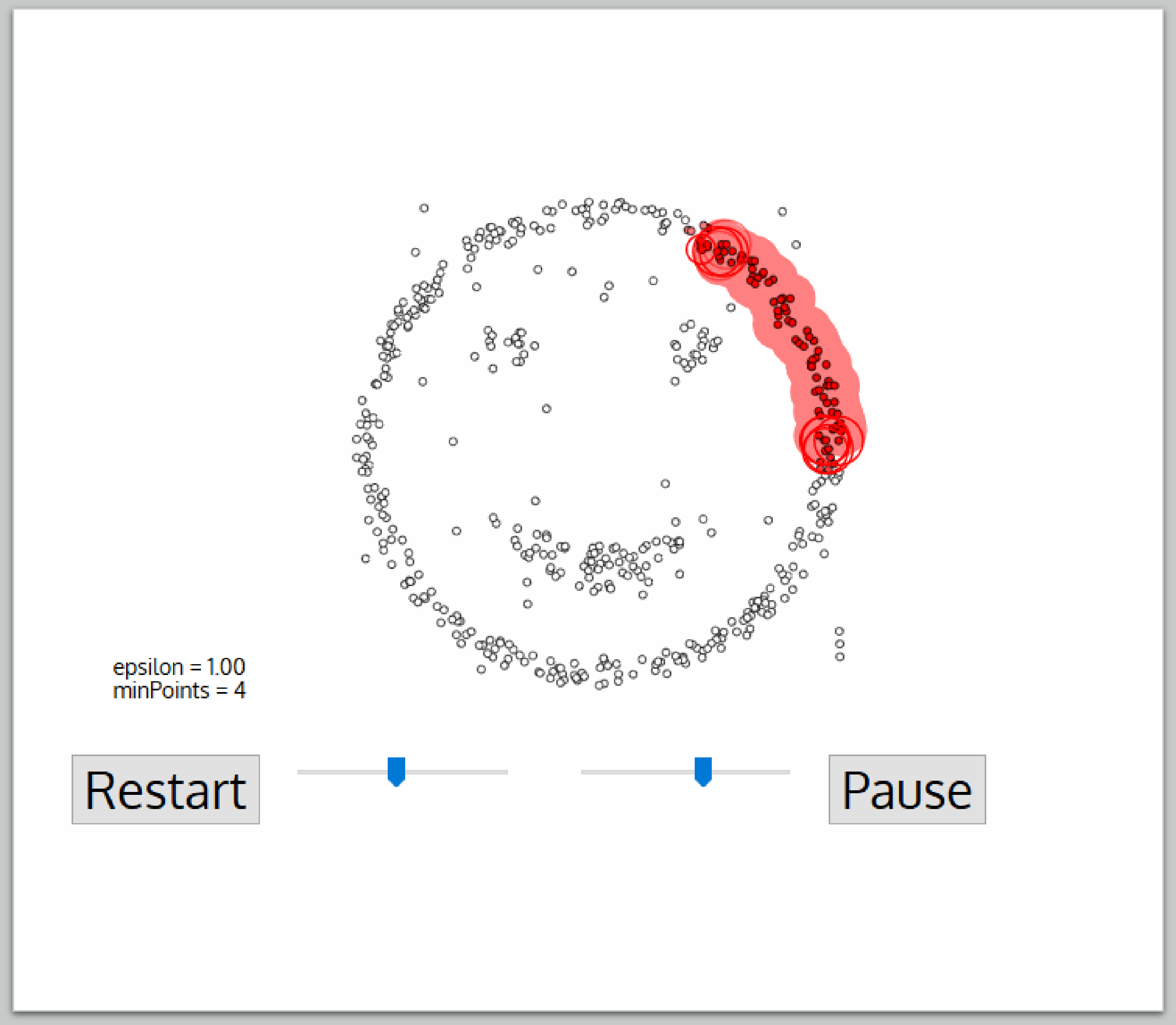
Algorithm Steps

Randomly select a point.

If within epsilon distance, there exists points >= minPoints, group these points into a cluster. Else, classify that point as noise.

Iterate through all neighboring points within the epsilon distance and expand the cluster until all points the neighborhood has been visited.

Repeat the process for a new unvisited point, and until all the points in the dataset have been visited.



Advantages	Disadvantages
• No need to specify the number of clusters (saves time for trial and error). • Able to discover clusters of arbitrary shapes. • Able to detect outliers in the data.	• Does not work very well for sparse datasets or datasets with varying density. • Not suitable for high-dimensional data, as distance calculation becomes difficult.

Conclusion

- DBSCAN is an unsupervised learning method.
  - Determines relationships between data points by forming clusters based on the proximity between the points and the number of points in an area (density of the points).
- DBSCAN can work with arbitrary shapes and outliers, but it does not work well with sparse datasets or high-dimensional data.
- Applications include:
  market research, pattern recognition, data analysis, and image processing.

Resources:

- https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556
- https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html
- https://www.digitalvidya.com/blog/the-top-5-clustering-algorithms-data-scientists-should-know/
- https://towardsdatascience.com/k-means-vs-dbscan-clustering-49f8e627de27
- https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556
- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html?highlight=dbscan

Questions?