Assignment 3: Clustering Algorithm Self-Study

Deadline: Friday, June 27th 2025

1. Algorithm Overview (20%)

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that identifies clusters as areas of high point density and labels points in low-density regions as noise (outliers).

- **Cluster Identification:**
  DBSCAN groups points that are closely packed together (points with many nearby neighbors). It starts from an arbitrary point and retrieves all points density-reachable from it. If this point has at least min_samples neighbors within a radius eps, it becomes a core point and forms a cluster. Points within eps of a core point are added to the cluster, and the process continues recursively. Points that do not belong to any cluster are labeled as noise.

- **Key Parameters:**
  - **eps (epsilon):** Defines the neighborhood radius around a point. Points within this distance are considered neighbors.
  - **min_samples:** The minimum number of points required (including the point itself) to form a dense region (cluster).

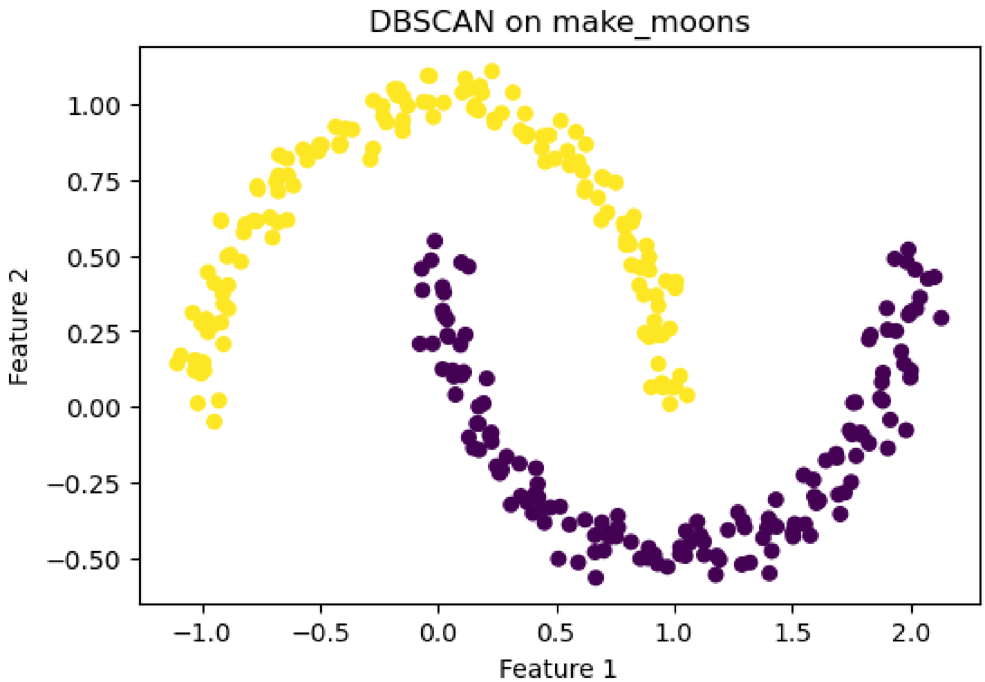
- **Strengths:**
  - Can find clusters of arbitrary shape (not just spherical).
  - Automatically detects and labels outliers as noise.
  - Does not require specifying the number of clusters in advance.

- **Limitations:**
  - Struggles with datasets containing clusters of varying densities.
  - Sensitive to parameter selection (eps and min_samples).
  - Can be less efficient on very large datasets.

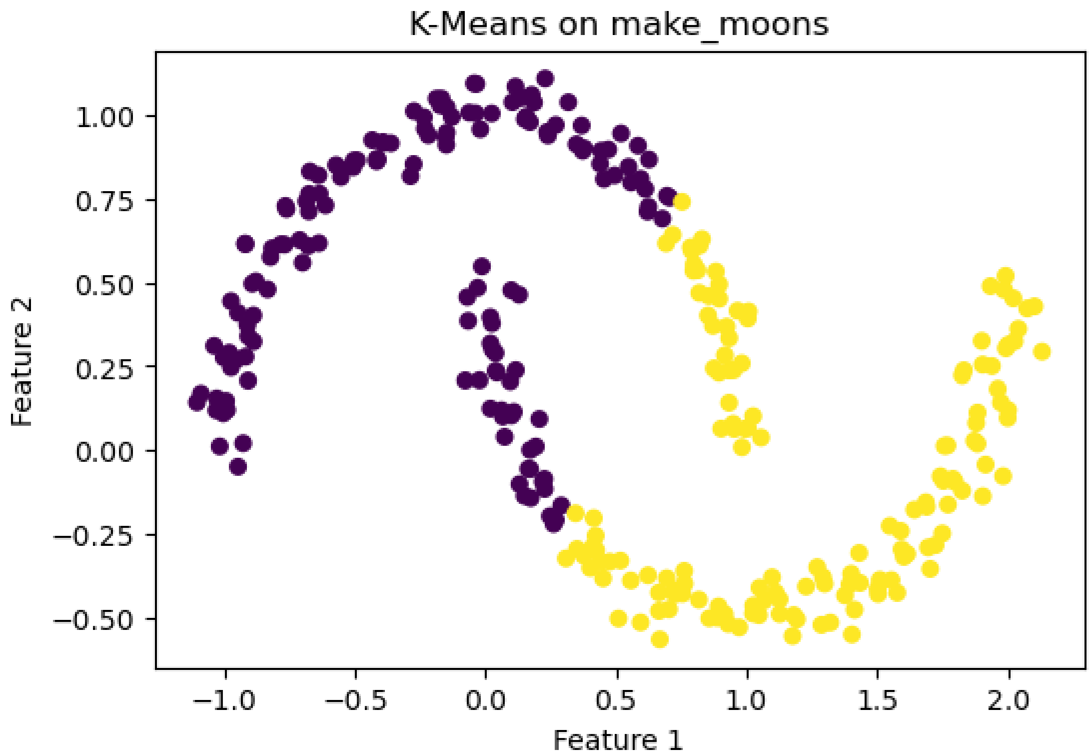
2. Algorithm Comparison (40%)

Visualizations

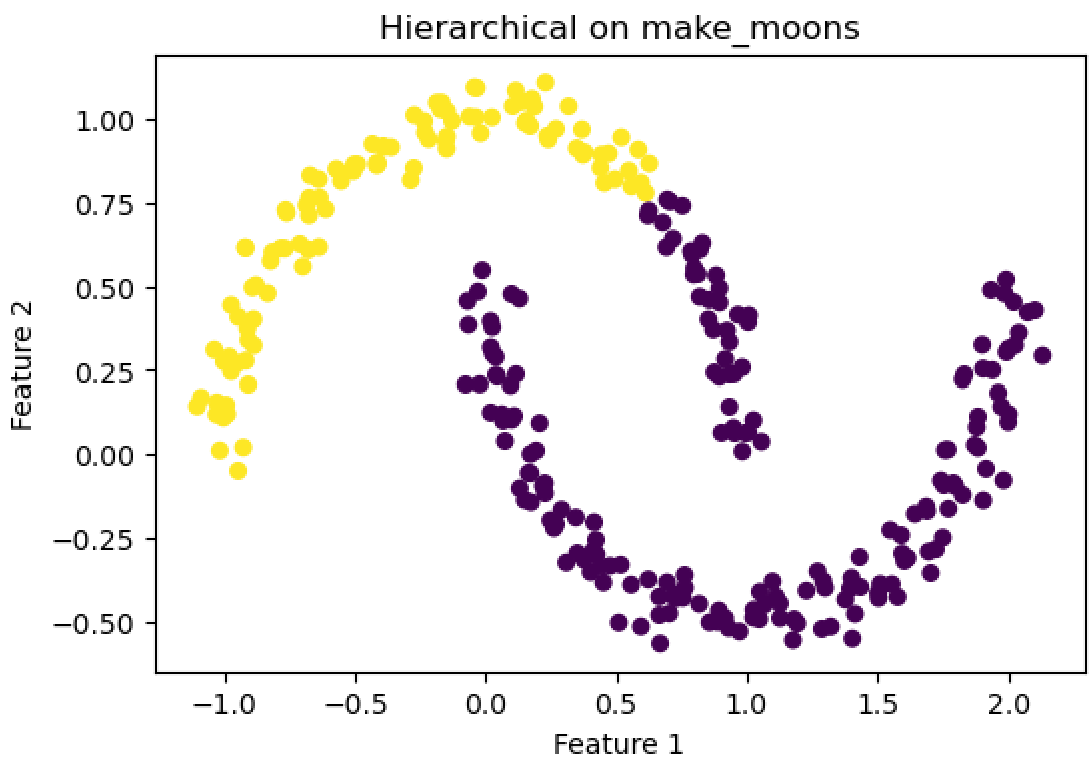
- **Figure 1:** DBSCAN on make_moons – Correctly identifies two crescent clusters and noise.



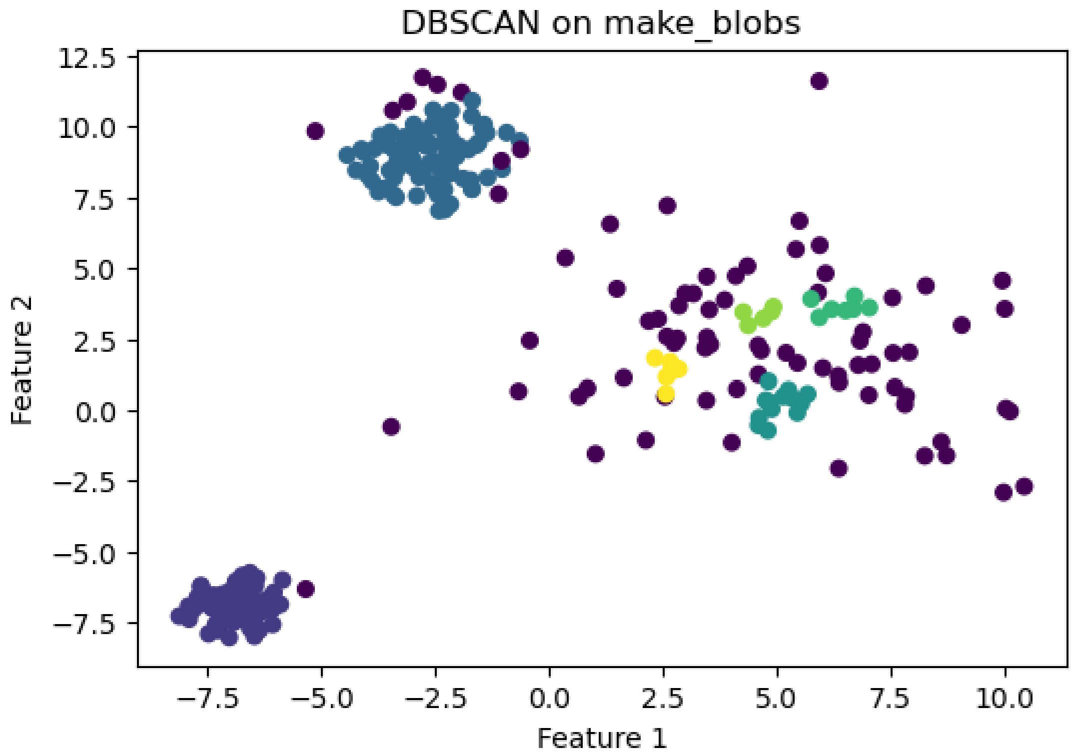
- **Figure 2:** k-Means on make_moons – Fails to capture the curved structure.



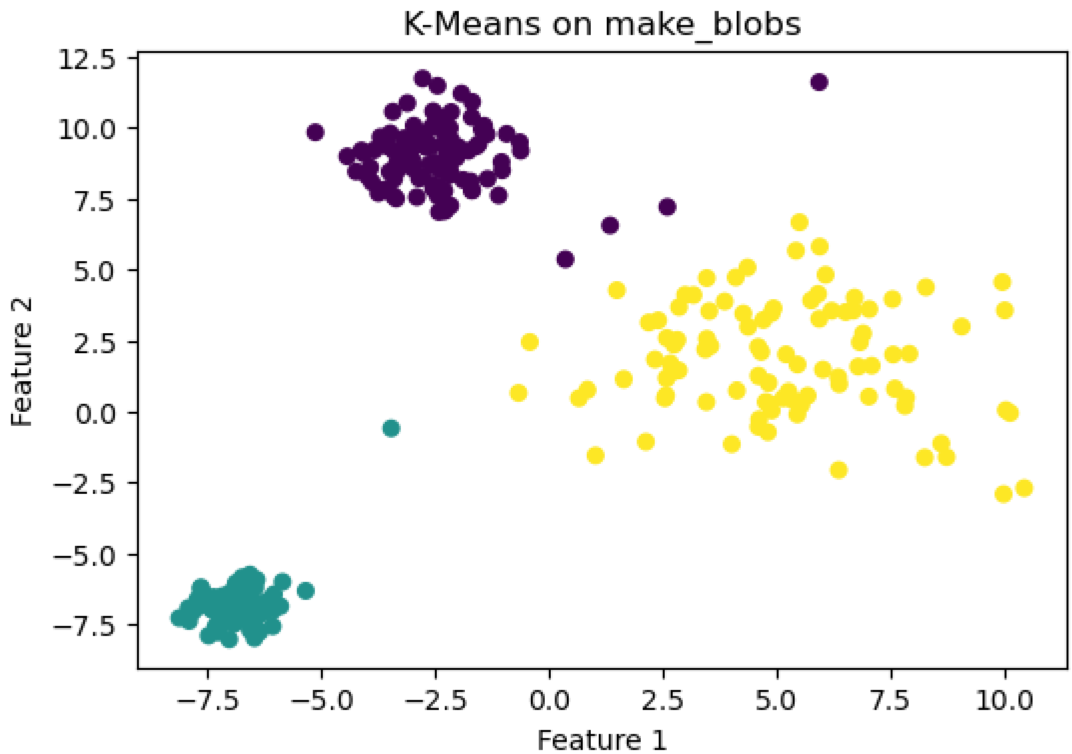
- **Figure 3:** Hierarchical Clustering on make_moons – Somewhat better than k-Means, but still imperfect.



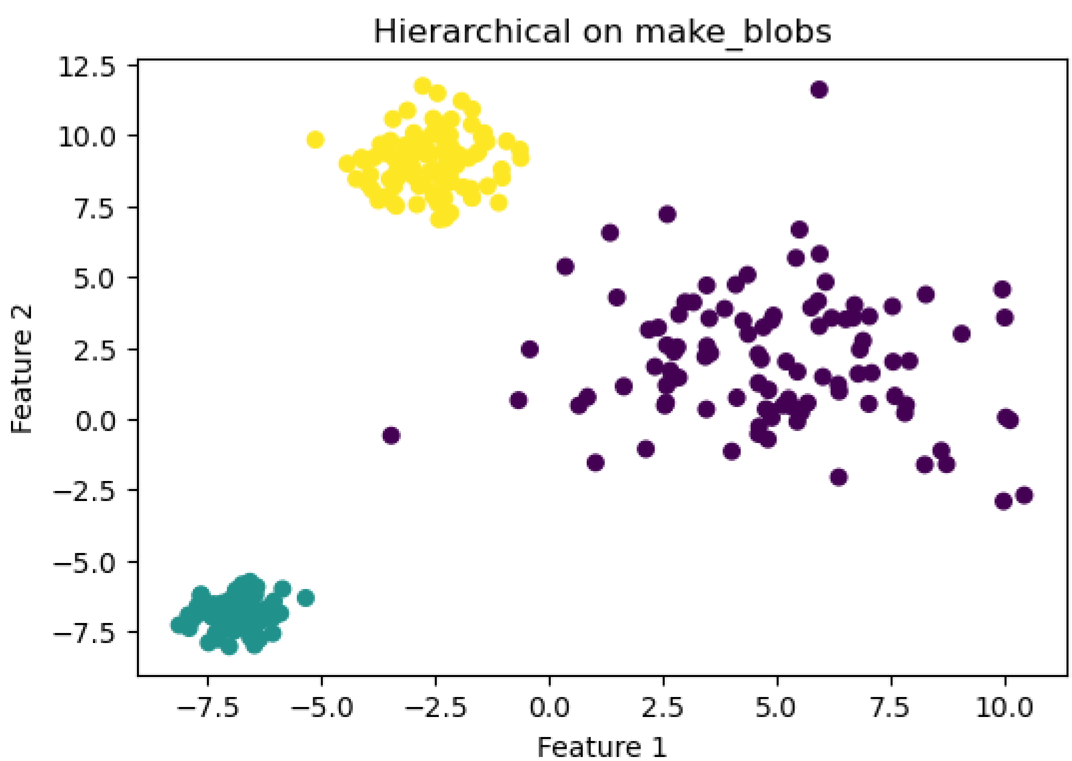
- **Figure 4:** DBSCAN on make_blobs – Struggles with clusters of varying density, merges/splits clusters incorrectly.



- **Figure 5:** k-Means on make_blobs – Accurately separates spherical clusters.



- **Figure 6:** Hierarchical Clustering on make_blobs – Performs similarly to k-Means.



Analysis

- **DBSCAN outperforms** k-Means and Hierarchical Clustering on non-spherical data (make_moons), successfully finding clusters and labeling noise.
- **DBSCAN struggles** on data with clusters of varying densities (make_blobs), as a single eps value cannot accommodate all cluster types. It may merge dense clusters or split sparse ones incorrectly.
- **k-Means** is efficient and effective for well-separated, spherical clusters but fails with non-spherical or noisy data.
- **Hierarchical Clustering** is flexible with cluster shapes but less scalable and sensitive to linkage choices.
- **Trade-offs:**
  - Use DBSCAN for data with arbitrary shapes and noise.
  - Use k-Means or Hierarchical Clustering for well-separated, similarly dense, spherical clusters.
  - Parameter tuning and understanding of data distribution are critical for DBSCAN.

3. Table Update (20%)

Compare and contrast characteristics for all three algorithms:

Feature	**k-Means**	**Hierarchical Clustering**	**DBSCAN**
Definition	Partitioning algorithm that assigns points to k clusters based on centroids	Builds a hierarchy of clusters using distance metrics	Density-based algorithm that groups points in high-density regions and marks low-density points as noise
Approach	Iteratively minimizes variance within k clusters	Agglomerative (bottom-up) or divisive (top-down)	Uses density reachability with parameters eps and min_samples to form clusters
Number of Clusters	Requires predefined k	Can be determined from dendrogram but subjective	Automatically determined by data density
Cluster Shape	Prefers spherical clusters	Works well with various shapes but can be unstable	Handles arbitrary/non-spherical shapes effectively

Initialization	Randomly selects k initial centroids	No initialization needed	No centroid initialization starts with arbitrary points
Result	Hard assignments—each point belongs to a single cluster	Hierarchical structure (tree/dendrogram)	Hard assignments with explicit noise identification
Interpretability	Moderate—cluster assignments but no hierarchy	High—dendrogram can be analyzed	Moderate - clusters defined by density, no hierarchy
Strengths	Simple, fast and efficient on large datasets	Can capture hierarchical relationships	- Handles arbitrary shapes - Robust to noise - No predefined cluster count needed
Limitations	Sensitive to initial centroids and k choice	Computationally expensive for large datasets	- Parameter-sensitive - Struggles with varying densities - Scalability challenges

4. Code Documentation & Submission Quality (20%)

<link to GitHub repository / code here>

https://github.com/Anjali-Sindha/Anjali-Sindha-Machine-Learning-AI-Bioinforma---BINF-5507-0TA