Assignment 3: Clustering Algorithm Self-Study

Deadline: Friday, June 27th 2025

Objective

This assignment focuses on the DBSCAN algorithm for clustering. You will:

- Understand DBSCAN's mechanics—how it identifies clusters and handles noise.
- Compare DBSCAN to k-Means and Hierarchical Clustering, highlighting their strengths and weaknesses.
- Expand the comparison table from class by adding DBSCAN.

🔹 **Alternative Option:** If you are particularly interested in a clustering algorithm not covered in class, you may propose it for approval before proceeding.

Assignment Instructions

1. Algorithm Overview (20%)

Write a clear and concise explanation of the DBSCAN algorithm, covering:

- **Cluster Identification:** Explain how DBSCAN groups points based on density and identifies noise.
- **Key Parameters:** Describe the roles of:
  - `eps` (epsilon): Defines the neighborhood radius around a point.
  - `min_samples`: The minimum number of points required to form a dense region.
- **Strengths and Limitations:**
  - When does DBSCAN work well? (e.g., detecting arbitrarily shaped clusters, handling noise)
  - When does DBSCAN struggle? (e.g., difficulty in tuning `eps`, varying density issues, scalability)

---

2. Algorithm Comparison (40%)

✅ Visualizations (20%)

Apply **DBSCAN, k-Means, and Hierarchical Clustering** to at least **two different datasets** and present results using clearly labeled plots:

- **Dataset where DBSCAN excels** (e.g., non-spherical clusters, datasets with noise).

- **Dataset where DBSCAN struggles** (e.g., varying densities, difficulty tuning `eps`).

Dataset Selection (Using `sklearn` Datasets)

To ensure consistency, you may use datasets from `sklearn.datasets` such as:

- `make_moons(n_samples=300, noise=0.05)` ➜ Good for DBSCAN (non-spherical clusters).
- `make_blobs(n_samples=300, centers=3, cluster_std=[1.0, 2.5, 0.5])` ➜ Shows DBSCAN struggles with varying densities.
- `make_circles(n_samples=300, factor=0.5, noise=0.05)` ➜ Highlights DBSCAN's ability to handle non-linearly separable data.

Ensure plots are easy to interpret with appropriate **axis labels, legends, and titles**.

✅ **Analysis (20%)**

- Compare performance: When does DBSCAN outperform k-Means and Hierarchical Clustering?
- Discuss failure cases: When does DBSCAN struggle, and why?
- Trade-offs: What factors influence the choice between these clustering methods?

---

3. Table Update (20%)

Expand the **comparison table from class** by adding a new column for DBSCAN (or your chosen algorithm; table can be completed directly in the report template).

🔹 **Expectations for the Table:**

- Keep descriptions **concise yet informative**—use **bullet points where necessary**.
- Support key points with **examples from your visualizations** (e.g., if DBSCAN handles noise well, reference your dataset).

4. Code Documentation & Submission Quality (20%)

📌 **Deliverables:**

1. **Structured report (2-3 pages maximum, uploaded as a PDF)** containing:
    - DBSCAN overview
    - Comparative analysis with visualizations
    - Expanded comparison table
2. **Code Submission:**
    - Provide a **link to your GitHub repository** where the full code is available.
    - Ensure scripts or notebooks are well-documented with **inline comments**.
    - Organize code clearly with **functions where appropriate**.

Grading Breakdown

Section	Percentage	Evaluation Criteria
Algorithm Overview	20%	Clarity, completeness, explanation of parameters and trade-offs
Algorithm Comparison	40%	Quality of visualizations, insightful comparisons, real-world discussion
Table Update	20%	Completeness, clarity, and accuracy of comparisons
Code & Submission Quality	20%	Code readability, organization, documentation, and report structure

---

Final Notes:

- If you wish to use a different dataset, get approval first.
- Reports must be **well-structured** and concise (avoid excessive text).
- Late submissions may incur penalties as per course policy.