### **1. Algorithm Overview (20%)**

**DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** is a density-based clustering algorithm that groups together points that are closely packed (dense regions) and marks points in low-density regions as noise or outliers.

* **Cluster Identification:**  
  DBSCAN starts with an arbitrary point and explores its neighborhood defined by eps (epsilon). If this point has at least min\_samples neighbors within eps, it forms a dense region (cluster core). The cluster expands by recursively including neighbors' neighbors. Points that cannot be assigned to any cluster are labeled as noise.
* **Key Parameters:**
  + eps (epsilon): Radius around a point to consider its neighborhood.
  + min\_samples: Minimum number of points required within the eps radius to qualify as a core point.
* **Strengths:**
  + Can identify clusters of arbitrary shapes.
  + Robust to noise and outliers (labels them as noise).
  + Does not require specifying the number of clusters beforehand.
* **Limitations:**
  + Sensitive to choice of eps and min\_samples.
  + Struggles with datasets of varying densities (single eps may not work for all).
  + Computationally expensive on very large datasets.

### 

### 

### 

### 

### 

### **2. Algorithm Comparison (40%)**

### **2. Algorithm Comparison (40%)**

**Visual Analysis Summary**

We applied all three algorithms—DBSCAN, k-Means, and Hierarchical Clustering—on two datasets:

1. **make\_moons**: Non-spherical clusters with noise.
   * **DBSCAN** performed best, capturing the curved shapes and handling noise well.
   * **k-Means** failed to separate the arcs correctly due to its preference for spherical clusters.
   * **Hierarchical Clustering** performed reasonably but with less clarity in boundaries.
2. **make\_blobs with varying densities**:
   * **k-Means** and **Hierarchical** did well with distinct cluster centers.
   * **DBSCAN** struggled to identify low-density clusters and misclassified some as noise.

**Performance & Trade-offs:**

* **DBSCAN** is superior when dealing with **noise** and **non-spherical clusters**, such as spatial data and geospatial applications.
* **k-Means** is fast and works well when clusters are **well-separated and equally sized**.
* **Hierarchical Clustering** is excellent for **small datasets** and **hierarchical analysis**, but computationally intensive for larger data.

### **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 clustering that groups points with dense neighborhoods and identifies noise |
| **Approach** | Iteratively minimizes variance within k clusters | Agglomerative (bottom-up) or divisive (top-down) | Groups points based on density connectivity |
| **Number of Clusters** | Requires predefined k | Can be determined from dendrogram but subjective | Automatically determines clusters based on density parameters |
| **Cluster Shape** | Prefers spherical clusters | Works well with various shapes but can be unstable | Detects arbitrarily shaped clusters |
| **Initialization** | Randomly selects k initial centroids | No initialization needed | No initialization; depends on parameters eps and min\_samples |
| **Result** | Hard assignments—each point belongs to a single cluster | Hierarchical structure (tree/dendrogram) | Hard assignments with noise points labeled as outliers |
| **Interpretability** | Moderate—cluster assignments but no hierarchy | High—dendrogram can be analyzed | Moderate — clusters and noise detected, no hierarchy |
| **Strengths** | Simple, fast and efficient on large datasets | Can capture hierarchical relationships | Can find clusters of arbitrary shape and handle noise well |
| **Limitations** | Sensitive to initial centroids and k choice | Computationally expensive for large datasets | Sensitive to eps and min\_samples; struggles with varying densities and scalability |

### 

### 

### **4. Code Documentation & Submission Quality (20%)**

<link to GitHub repository / code here>

[aryanrathod23/dbscan\_clustering\_assignment](https://github.com/aryanrathod23/dbscan_clustering_assignment)