MACHINE LEARNING- ASSIGNMENT- 03

Algorithm overview: DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

CLUSTER IDENTIFICATION:

DBSCAN groups points based on their density rather than relying on predefined cluster numbers. It classifies points into three categories:

1. Core Points: Points with at least min_samples neighbors within a radius of eps.
2. Broder Points: Points that are within eps of a core point but do not meet the min_samples requirements.
3. Noise Points (outliers): Points that are neither core nor border points.

Clusters are formed by expanding from core points and connecting other core points or border points within eps. Noise points remain unclustered.

KEY PARAMETERS:

1. eps (epsilon)
   - Defines the neighborhood radius around a point
   - A larger eps results in fewer, larger clusters, while smaller eps can lead to many small clusters or more noise.
2. min_samples:
   - The minimum number of points required in an eps-radius neighborhood to consider a point as a core point.
   - Higher values make cluster more compact and resistant to noise but may cause small clusters to be ignored.

STRENGTHS:

- Detects Arbitrary Shapes: Unlike k-means, DBSCAN can find non-spherical clusters.
- Handle noise well: It can classify outliers explicitly instead of forcing them into clusters.
- No need to predefine cluster count: Unlike k-means, DBSCAN does not require the number of clusters as an input.

LIMITATIONS:

- Parameter sensitivity: Choosing the right eps and min_samples is challenging and often requires tuning.
- Varying density problems: Struggles when clusters have different densities, as a single eps may not fit all.
- Scalability issues: Has a complexity of O(n log n) but may become inefficient for very large datasets.

ANALYSIS:

Ans:01 DBSCAN excels when the data contains clusters of arbitrary shape or when there are outliers/noise points. In the “moons” and “circles” datasets, DBSCAN outperforms k-means and hierarchical clustering because it can correctly separate non-spherical clusters and identify noise points. k-means, on the other hand, tries to form circular clusters and struggles with curved shapes, while hierarchical clustering depends heavily on the linkage criteria and can also fail to capture complex shapes unless carefully turned.

Ans:02 DBSCAN relies on finding high density regions separated by lower density areas.
It struggles when clusters have varying densities because one setting for the neighborhood size may not work for all clusters. It also faces issues with high dimensional data where distance measurements become less reliable, and it can fail when there aren’t clear dense regions, leading to misclassification of points as noise.

Ans:03 Factors influencing the choice include the shape of clusters (spherical vs. arbitrary), the presence of outliers, dataset size and dimensionality, and parameter sensitivity. For example, k-means is fast and works well with spherical clusters but struggles with outliers and non-linear shapes, whereas DBSCAN handles arbitrary shapes and noise but requires careful tuning of parameters. Hierarchical clustering, meanwhile, can reveal nested structures but may be computationally expensive and sensitive to chosen linkage method.

TABLE

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 based on region density
Approach	Iteratively minimizes variance within k clusters	Agglomerative (bottom-up) or divisive (top-down)	Expands clusters from high-density regions using eps and min_samples parameters
Number of clusters	Requires predefined k	Can be determined from dendrogram but subjective	Automatically determines clusters based on density, no present number needed
Cluster shape	Prefers spherical clusters	Works well with various shapes but can be unstable	Handles clusters of arbitrary shapes (eg. “moons” and “circles” in visualizations)
Initialization	Randomly selects k initial centroids	No initialization needed	No initializations; clusters form based on density threshold
Result	Hard assignments—each point belongs to a single cluster	Hierarchical structure (tree/dendrogram)	Hard assignments with additional identification of noise/ outliers
Interpretability	Moderate—cluster assignments but no hierarchy	High—dendrogram can be analyzed	Moderate; clusters are defined by density, though setting optimal parameters may require tuning
Strengths	Simple, fast and efficient on large datasets	Can capture hierarchical relationships	Excels in identifying non-spherical clusters and handling noise
Limitations	Sensitive to initial centroids and k choice	Computationally expensive for large datasets	Struggles with clusters of varying density and is sensitive to eps and min_samples parameters.

GITHUB- https://github.com/Binal-1805/BINF-5507/blob/main/Assignment-3.ipynb