DBSCAN Algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that groups together points that are closely packed together while marking points that lie alone in low-density regions as outliers.

Steps of the DBSCAN Algorithm

1. **Initialization**:
    - Choose the parameters eps (epsilon) and min_samples.
        - eps: The maximum distance between two samples for them to be considered as in the same neighborhood.
        - min_samples: The minimum number of samples in a neighborhood for a point to be considered a core point.
2. **Classify Points**:
    - For each point in the dataset, check if it is a core point:
        - A core point has at least min_samples points (including itself) within eps distance.
3. **Expand Clusters**:
    - Start with an unvisited point and perform the following steps:
        - If the point is a core point, create a new cluster.
        - Add all points within eps distance of the core point to the cluster.
        - Recursively visit each point in the cluster's neighborhood and add their neighbors to the cluster if they are also core points.
4. **Handle Noise**:
    - Points that are not core points and not reachable from any core point are considered noise or outliers.

Example

Consider a small dataset with the following points and parameters: eps = 2 and min_samples = 3.

Sample Dataset

Points: (1, 2), (2, 2), (2, 3), (8, 7), (8, 8), (25, 80)

Step-by-Step Execution

1. **Initialize Parameters**:
    - eps = 2
    - min_samples = 3

2. **Classify Points**:
    - Calculate the number of points within eps distance for each point.
3. **Expand Clusters**:
    - For each unvisited point, check if it is a core point and expand the cluster.

Detailed Execution

1. **Point (1, 2)**:
    - Neighbors within eps: (1, 2), (2, 2), (2, 3)
    - Core point: Yes (3 neighbors including itself)
    - Create a new cluster: C1 = {(1, 2), (2, 2), (2, 3)}
2. **Point (2, 2)**:
    - Already visited and part of cluster C1.
3. **Point (2, 3)**:
    - Already visited and part of cluster C1.
4. **Point (8, 7)**:
    - Neighbors within eps: (8, 7), (8, 8)
    - Core point: No (2 neighbors, less than min_samples)
    - Not a core point, check neighbors.
        - Point (8, 8) is a core point with neighbors (8, 7), (8, 8).
        - Expand cluster: C2 = {(8, 7), (8, 8)}
5. **Point (8, 8)**:
    - Already visited and part of cluster C2.
6. **Point (25, 80)**:
    - Neighbors within eps: (25, 80)
    - Core point: No (1 neighbor, less than min_samples)
    - Not a core point, mark as noise.

Resulting Clusters and Noise

- Clusters:
    - C1: {(1, 2), (2, 2), (2, 3)}
    - C2: {(8, 7), (8, 8)}
- Noise: {(25, 80)}