Clustering Techniques Report

Q1. Real-World Datasets for Clustering

Dataset 1 for Density-based Clustering
Dataset: Customers Dataset
This dataset contains information about customers, which is useful for performing clustering based on customer features, such as purchase patterns, geographic location, and demographics. It is an ideal candidate for density-based clustering, like DBSCAN, where clusters are formed by dense areas of similar customer behaviors.

Dataset 2 for Hierarchical-based Clustering
Dataset: World University Rankings
The World University Rankings dataset includes various metrics like student satisfaction, research output, and international diversity of universities. Hierarchical clustering methods are suitable here to form clusters of universities based on similarity in rankings and characteristics.

Dataset 3 for Prototype-based Clustering
Dataset: Restaurant and Consumer Context-Aware Recommendation
This dataset provides information about restaurants and customer preferences. Prototype-based clustering, such as k-means, is a great choice for grouping restaurants that share common attributes, like cuisine type or average ratings.

Q2. Density-based Clustering: DBSCAN

(b) Advantages and Disadvantages of DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering technique that focuses on grouping data points that are densely packed while distinguishing noise points that don't belong to any cluster.

Advantages of DBSCAN:

1. **No Need to Specify the Number of Clusters**: Unlike k-means, DBSCAN doesn't require prior knowledge about the number of clusters. It automatically identifies the optimal number of clusters based on density.

2. **Adaptability to Density Variation**: DBSCAN is capable of identifying clusters with varying densities, unlike k-means, which assumes clusters are spherical. This is especially useful for datasets where clusters are irregular or have different densities.

Disadvantages of DBSCAN:

1. **Challenges with Global Clusters**: DBSCAN focuses on local density, which means it may struggle to detect clusters that are globally distributed across the dataset. Some clusters might remain undetected if they don't have enough connected points or if they span a wide area.
2. **Difficulty in High-Dimensional Data**: As the number of dimensions increases, the concept of density and distance becomes less meaningful. DBSCAN performs poorly in high-dimensional spaces due to the "curse of dimensionality," where points are often far apart and density estimations become less reliable.

(c) HDBSCAN Over DBSCAN

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) extends DBSCAN by introducing hierarchical clustering techniques to density-based clustering.

Advantages of HDBSCAN over DBSCAN:

1. **Automatic Cluster Hierarchy**: Unlike DBSCAN, which produces flat clusters, HDBSCAN creates a hierarchy of clusters. This hierarchy allows users to observe the data at different levels of granularity, providing more insights into both global and local structures.
2. **Simplified Parameter Tuning**: DBSCAN requires careful selection of both the distance threshold (ε) and the minimum number of points (MinPts). In contrast, HDBSCAN simplifies the process by needing just one parameter: the minimum cluster size.

Q4. K-means Clustering Limitations and Solutions

(a) Impact of Outliers and Scaling with Dimensions

(i) Clustering Outliers
K-means is sensitive to outliers, and they can significantly affect the resulting clusters. Outliers can shift the centroid of clusters or even form their own incorrect clusters.

- **Impact of Outliers**: K-means minimizes the squared distances between points and centroids, so outliers can distort the centroid's position. This can lead to inaccurate clustering results, where outliers are incorrectly assigned to a cluster or pull centroids away from the true center of clusters.

- **Solutions**: Alternative clustering methods like K-medoids (PAM) and DBSCAN are less affected by outliers. DBSCAN, for instance, can identify outliers as noise and exclude them from the clustering process.

(ii) Scaling with Dimensions
As the number of dimensions increases, K-means performance tends to degrade due to the curse of dimensionality. In higher-dimensional spaces, distances between points become less meaningful, and the computational complexity of the algorithm increases.

- **Impact**: In high-dimensional spaces, data points tend to be closer together, leading to poor separation of clusters and increased computational cost.
- **Solutions**: Dimensionality reduction techniques such as PCA (Principal Component Analysis) can be used before applying K-means to reduce the number of features and improve clustering performance.

(b) Solutions to K-means Limitations

(i) Limitation 1: Clustering Outliers
To handle outliers, DBSCAN is an effective alternative. It works by identifying "core" points that are densely surrounded by other points and distinguishing them from outliers, which are considered noise.

- **DBSCAN Approach**: It uses two key parameters:
    - **MinPts**: Minimum number of points to form a dense region.
    - **ε (epsilon)**: Maximum distance between points for them to be considered neighbors.
    - It classifies points as core points (in dense regions), border points (on the edge of clusters), or noise points (outliers).

(ii) Limitation 2: Scaling with Number of Dimensions
To address the curse of dimensionality, **Spectral Clustering** is a suitable alternative.

- **Spectral Clustering Approach**: It constructs a similarity matrix based on the relationships between data points, then applies dimensionality reduction (e.g., eigenvalue decomposition) to map the data into a lower-dimensional space where clustering is more effective. This technique is especially useful for high-dimensional data.

Q3. Hierarchical Clustering

(a) Agglomerative vs. Divisive Clustering

Hierarchical clustering can be categorized into two types:

1. **Agglomerative Clustering**: This is the most common approach, where each data point initially represents its own cluster. The algorithm then iteratively merges the closest clusters until a single cluster remains. It's a bottom-up approach.
2. **Divisive Clustering**: In contrast, this approach starts with all data points in one cluster and iteratively splits them into smaller clusters until each data point forms its own cluster. It's a top-down approach.

Agglomerative Clustering vs. Divisive Clustering
Agglomerative clustering is more widely used due to several reasons:

- **Computational Efficiency**: Agglomerative clustering is generally more computationally efficient, especially for large datasets. Divisive clustering requires the calculation of dissimilarities between all pairs of data points, which can be expensive.
- **Algorithm Popularity**: Algorithms like Ward's method, Single Linkage, and Complete Linkage, which are widely used in clustering, are all agglomerative in nature.
- **Interpretability**: The hierarchical structure produced by agglomerative clustering is easier to interpret and visualize, which aids in understanding the relationships between clusters.

Due to these advantages, **agglomerative clustering** is the preferred method and more commonly used in practice.