Q1.Choose any 3 real world datasets:-

Dataset 1 for **Density based clustering**:-

<https://www.kaggle.com/datasets/datascientistanna/customers-dataset>

S

Dataset 2 for **Hierarchical based clustering:**-

<https://www.kaggle.com/datasets/mylesoneill/world-university-rankings>

Dataset 3 for **Prototype l based clustering**:-

<https://www.kaggle.com/datasets/thedevastator/restaurant-and-consumer-context-aware-recommenda>

Q2.**Density based clustering**

(b) Mention two advantages and disadvantages of DBSCAN.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular

density-based clustering algorithm used for identifying clusters and noise in data.

The key idea behind DBSCAN is to group data points that are densely packed and separated by areas of lower point density.

It is different from k-means clustering.

Advantages of DBSCAN:-

Advantages:

1. DBSCAN is reasonably efficient for clustering huge datasets when compared to other clustering techniques such as hierarchical clustering. Its time complexity is frequently lower than that of techniques requiring pairwise distance matrix calculations.
2. Adaptive to Density Variation: DBSCAN can adjust to varied data point densities, which is important in cases where clusters have varying densities. It can manage compact or sparse clusters without requiring manual parameter adjustments for each cluster.
3. DBSCAN does not require you to specify the number of clusters in advance, unlike K-Means, which necessitates a priori knowledge of the desired cluster count. DBSCAN can automatically discover the optimal number of clusters based on the data's inherent density structure

DisAdvantages

1. May Not Find Global Clusters , DBSCAN identifies clusters based on local density relationships, which means it may not always find global clusters that span a large portion of the dataset. Some clusters may remain undiscovered if they are not well-connected to a core point.
2. Difficulty in Clustering High-Dimensional Data , DBSCAN's performance can degrade significantly in high-dimensional spaces. In high-dimensional data, the concept of distance becomes less meaningful, and data points tend to be far apart, which makes it challenging to define meaningful neighborhoods and densities

Q2(c).

**HDBSCAN, or Hierarchical Density-Based Spatial Clustering** of Applications with Noise, is a density-based clustering algorithm that extends the concepts of DBSCAN.

It allows a more a more versatile and automated approach to density-based clustering.

The following are two benefits of HDBSCAN over DBSCAN:

1. Automatic Cluster Hierarchy: HDBSCAN creates a cluster hierarchy, allowing you to view the data at various degrees of granularity. This hierarchical representation of the data provides insights into both global and local clustering structures, making it more adaptable than DBSCAN, which often identifies flat clusters.
2. Simplified Parameter Selection: HDBSCAN simplifies the process of choosing clustering parameters. In DBSCAN, you need to decide on both the distance parameter (ε or epsilon) and the minimum number of points (MinPts) for a cluster, which can be a daunting task. HDBSCAN, on the other hand, requires only one parameter, "minimum cluster size."

Q4

(a)

(i) **Clustering Outliers**:

K-means is susceptible to data outliers. Data points that substantially differ from the bulk of the data points in a cluster are called outliers.

The impact of outliers on K-means is as follows:

The cluster centroids' positions can be distorted by outliers. K-means aims to reduce the total squared distances between data points and the cluster centroids that correspond to them. The centroids may be drawn towards outliers, which could result in incorrect cluster assignments for the bulk of the data points.

It's possible for outliers to create their own clusters: K-means occasionally classifies outliers as belonging to a cluster even when they don't. As a result, outlier clusters that don't accurately reflect the data's trends may form.

To deal with the matter Alternative clustering techniques, including K-medoids (PAM - Partitioning Around Medoids) and DBSCAN (Density-Based Spatial Clustering of Applications with Noise), can be more resilient to the problem of outliers since they are less affected by their outliers existence.

(ii) Scaling with Number of Dimensions:

The performance of K-means clustering may decrease as the quantity of dimensions in the data rises. It's commonly known as the "curse of dimensionality."

High-dimensional data affects K-means in the following ways:

Increasing computational complexity: The processing cost of calculating the distances between data points rises substantially as the number of dimensions increases, while the distances themselves become less meaningful. In high-dimensional spaces, K-means becomes more computationally demanding since it computes the distances between data points and cluster centroids in each dimension.

Overfitting and decreased cluster separability: Data points are more likely to be close to one another in terms of distance when they are scattered sparsely in high-dimensional spaces. This may result in the formation of numerous tiny, densely populated clusters

Principal Component Analysis (PCA) and spectral clusterig are two dimensionality reduction approaches that can be used before clustering to lower the number of dimensions and enhance K-means performance in order to combat the curse of dimensionality.

(b)

(i) Limitation 1: Clustering Outliers

we can utilise the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm to overcome the drawback of clustering outliers.

Method Used by DBSCAN: DBSCAN is a density-based clustering method that isolates noise points (outliers) and clusters data points that are densely adjacent to one another.

It identifies border points—data points that are close to core points but not in high-density areas—noise points, and core points—data points in high-density zones.

Two important parameters are used by DBSCAN:

MinPts, which is the lowest number of data points needed to build a dense zone, and epsilon (ε), which is the maximum radius of a neighbourhood.

Starting from a core point and building a cluster by connecting core points that are easily accessible, it finds clusters. There are no noise points allocated to any

(ii) Limitation 2: Scaling with Number of Dimensions

Algorithm: Spectral Clustering

* Spectral clustering is a graph-based clustering algorithm that works well in high-dimensional spaces and can mitigate the curse of dimensionality.
* It first constructs an affinity matrix that quantifies the similarity between data points. This can be done using various similarity measures, such as Gaussian similarity or k-nearest neighbors.
* Spectral clustering then employs dimensionality reduction techniques, such as eigenvalue decomposition, to project the data into a lower-dimensional space. This reduces the impact of high dimensionality on the clustering process.
* After dimensionality reduction, a standard clustering algorithm like K-means can be applied in the reduced space, and the clusters found in the lower-dimensional representation are mapped back to the original space.
* Spectral clustering is effective in maintaining the structure of the data and mitigating the issues related to high-dimensional data, making it a useful alternative to K-means for such scenarios.

Q3.a

Hierarchical clustering can be categorized into two main categories:

* Agglomerative Clustering (Agglomerative Hierarchical Clustering): This method starts with each data point as its own cluster and then successively merges or agglomerates clusters into larger ones until all data points belong to a single cluster.
* Divisive Clustering (Divisive Hierarchical Clustering): This method starts with all data points in a single cluster and then successively divides or "divides" the cluster into smaller ones until each data point is in its own cluster.

Hence , the Agglomerative Clustering falls into the Agglomerative Hierarchical Clustering ..

Agglomerative clustering is more widely used and popular than divisive clustering for several reasons:

* Complexity: Agglomerative clustering is often computationally more efficient than divisive clustering, especially when dealing with large datasets. Divisive clustering requires the computation of dissimilarities between all pairs of data points, which can be computationally expensive.
* Popular Algorithms: Some of the most popular hierarchical clustering algorithms, such as Ward's method, Single Linkage, and Complete Linkage, are agglomerative in nature. These algorithms are widely available in various data analysis and machine learning libraries.
* Interpretability: Agglomerative clustering results are often easier to interpret and explain, as the hierarchical structure naturally represents the grouping of data points.

Due to these advantages, agglomerative clustering is the preferred choice for most applications, and it is the category of hierarchical clustering that is more commonly used.