Clustering Algorithm Implementation

KMeans Clustering

KMeans clustering is an iterative algorithm that partitions data into K distinct clusters based on feature similarity. The algorithm follows these steps:

Initialization: Randomly select K centroids (initial cluster centers) from the dataset.
Assignment: Assign each data point to the nearest centroid based on Euclidean distance.
Update: Recalculate the centroids by taking the mean of all data points assigned to each cluster.
Repeat: Iterate through the assignment and update steps until the centroids no longer change significantly or a predetermined number of iterations is reached.

KMeans is suitable for the Iris dataset due to the following reasons:

Numerical Features: The dataset consists entirely of numerical features, making it compatible with KMeans, which relies on distance metrics.
Distinct Clusters: The Iris dataset is known to contain three distinct species, suggesting the potential for well-defined clusters.
Scalability: KMeans is efficient and can handle larger datasets, although it performs well even on smaller datasets like Iris.

Loading and Preprocessing: Load the Iris dataset and drop the species column to focus solely on the numerical features.
Choosing the Number of Clusters (k)
Applying KMeans Clustering: Fit the KMeans model to the preprocessed data using the selected value of k.

Once KMeans clustering is applied, we can visualize the clusters in a scatter plot, typically using two of the four features to illustrate the separations among the clusters. In this plot, data points will be colored according to the identified clusters, showcasing how well the clusters separate based on their measured characteristics. The centroids can also be marked to visualize the center of each cluster.

Hierarchical Clustering

Hierarchical clustering builds a hierarchy of clusters using either an agglomerative (bottom-up) or divisive (top-down) approach.
In agglomerative clustering, the steps are as follows:
Start: Treat each data point as a separate cluster.

Merge: At each iteration, identify the two closest clusters based on a distance metric and merge them.
Repeat: Continue merging until all data points are in a single cluster or a desired number of clusters is reached.
The result is often displayed as a dendrogram, which visually represents the cluster hierarchy.

Divisive Clustering: This approach starts with one cluster containing all data points and recursively splits it into smaller clusters. This method is less common due to its computational complexity.

Hierarchical clustering is suitable for the Iris dataset for several reasons:

No Need to Specify K: Unlike KMeans, hierarchical clustering does not require a pre-defined number of clusters, making it advantageous when the optimal number of clusters is unknown.
Dendrogram Visualization: The dendrogram provides an intuitive way to visualize cluster formation and relationships among the data points, aiding in interpretation.Well-Separated Clear Structure: The dataset contains distinct features (e.g., sepal length, sepal width, petal length, and petal width) that are likely to form identifiable clusters.
Handling Various Shapes: Hierarchical clustering can accommodate clusters of different shapes and sizes, aligning well with the diversity in the data.

Applying Hierarchical Clustering:

- Use the AgglomerativeClustering method from the sklearn library.
- Select an appropriate number of clusters, typically three based on prior knowledge or exploratory analysis.
- Fit the model to the preprocessed data.

Visualization of Clusters:

- Generate a dendrogram to visualize the hierarchical relationships between clusters.
- Create a scatter plot to visualize the clusters in two dimensions, using key features like petal

By applying these two clustering techniques to the Iris dataset, we can explore and identify natural groupings within the data, enhancing our understanding of the relationships between different iris species based on their features.