Create a random dataset of 500 rows and 10 columns.

Columns 1 to 4 are defined between [-10, 10];

Columns 5 to 8 are defined between [10, 20];

Columns 9 to 10 are defined between [-100, 100].

Apply following clustering algorithms; determine the optimal number of clusters and plot distance metric

Graph using each algorithm.

1. K-Mean clustering
2. Hierarchical clustering

Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

From sklearn.cluster import KMeans

From scipy.cluster.hierarchy import dendrogram, linkage

From scipy.spatial.distance import cdist

# Create random dataset

Np.random.seed(42)

Data = np.hstack((

Np.random.uniform(-10, 10, (500, 4)),

Np.random.uniform(10, 20, (500, 4)),

Np.random.uniform(-100, 100, (500, 2))

))

# Convert to DataFrame

Df = pd.DataFrame(data, columns=[f’Feature\_{i+1}’ for I in range(10)])

# K-Means Clustering

Inertia = []

K = range(1, 15)

For k in K:

Kmeans = KMeans(n\_clusters=k, random\_state=42)

Kmeans.fit(df)

Inertia.append(kmeans.inertia\_)

# Plot Elbow Method For Optimal k

Plt.figure(figsize=(10, 6))

Plt.plot(K, inertia, ‘bx-‘)

Plt.xlabel(‘Number of clusters’)

Plt.ylabel(‘Inertia’)

Plt.title(‘Elbow Method For Optimal k’)

Plt.show()

# Hierarchical Clustering

Linked = linkage(df, method=’ward’)

# Plot Dendrogram

Plt.figure(figsize=(12, 6))

Dendrogram(linked, truncate\_mode=’lastp’, p=30, show\_leaf\_counts=True)

Plt.title(‘Hierarchical Clustering Dendrogram’)

Plt.xlabel(‘Sample index or (cluster size)’)

Plt.ylabel(‘Distance’)

Plt.show()