**Implementation of any one Hierarchical Clustering method**

import numpy as np

import matplotlib.pyplot as plt

from scipy.cluster.hierarchy import dendrogram, linkage, fcluster

from sklearn.datasets import make\_blobs

X, y\_true = make\_blobs(n\_samples=300, centers=4, cluster\_std=0.60, random\_state=42)

plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)

plt.scatter(X[:, 0], X[:, 1], c=y\_true, cmap='viridis')

plt.title('True Clusters')

Z = linkage(X, method='ward')

plt.subplot(1, 3, 2)

plt.title('Dendrogram')

dendrogram(Z)

plt.xlabel('Sample Index')

plt.ylabel('Distance')

max\_d = 8

clusters = fcluster(Z, max\_d, criterion='distance')

plt.subplot(1, 3, 3)

plt.scatter(X[:, 0], X[:, 1], c=clusters, cmap='viridis')

plt.title('Agglomerative Clustering Results')

plt.tight\_layout()

plt.show()

plt.figure(figsize=(15, 4))

plt.subplot(1, 4, 1)

dendrogram(Z, truncate\_mode='lastp', p=12)

plt.title('Truncated Dendrogram')

plt.subplot(1, 4, 2)

clusters\_3 = fcluster(Z, 3, criterion='maxclust')

plt.scatter(X[:, 0], X[:, 1], c=clusters\_3, cmap='viridis')

plt.title('3 Clusters')

plt.subplot(1, 4, 3)

clusters\_4 = fcluster(Z, 4, criterion='maxclust')

plt.scatter(X[:, 0], X[:, 1], c=clusters\_4, cmap='viridis')

plt.title('4 Clusters')

plt.subplot(1, 4, 4)

clusters\_5 = fcluster(Z, 5, criterion='maxclust')

plt.scatter(X[:, 0], X[:, 1], c=clusters\_5, cmap='viridis')

plt.title('5 Clusters')

plt.tight\_layout()

plt.show()

print(f"Number of clusters formed: {len(np.unique(clusters))}")

print(f"Cluster sizes: {np.bincount(clusters)}")