

## Pass Task 7.1P: K-Means and Hierarchical Clustering

```

to 21)
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.cluster import dendrogram, linkage

# Set seed for reproducibility
np.random.seed(0)

# Create synthetic dataset
X, y = make_blobs(n_samples=100, centers=[[0, 0], [10, 4], [10, 10], [0, 10], [0, 0]], cluster_std=1)

# Split the dataset into 5 clusters
plt.figure(figsize=(10, 10))
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, marker='o')
plt.title('Synthetic Dataset')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.grid(True)
plt.show()

# Hierarchical clustering
linkage = linkage(X, method='complete', metric='euclidean', n_jobs=-1)
dendrogram(linkage)

# K-Means clustering
kmeans = KMeans(n_clusters=5, random_state=0)
kmeans.fit(X)

# Plot the clusters with their centroids
plt.figure(figsize=(10, 10))
plt.scatter(X[:, 0], X[:, 1], c=kmeans.labels_, s=50, marker='o')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], c='red', s=100, marker='x')
plt.title('K-Means Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.grid(True)
plt.show()

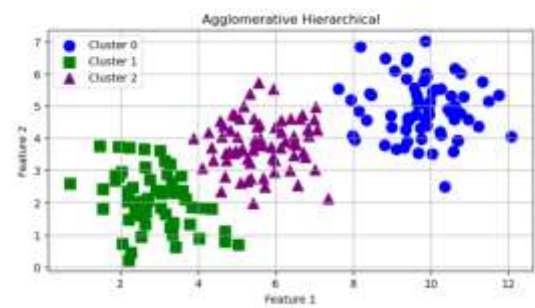
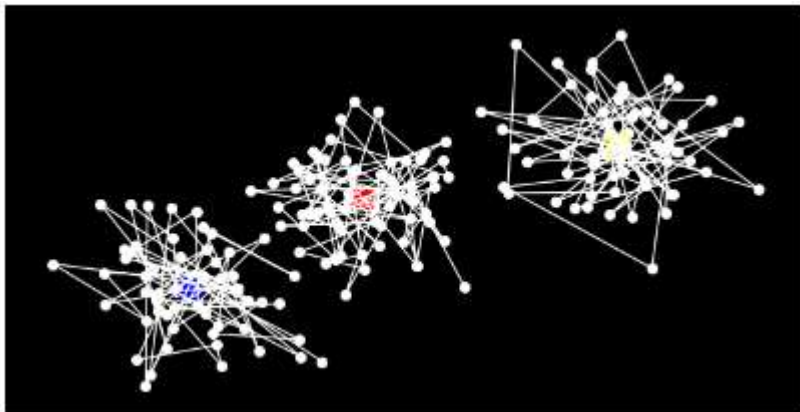
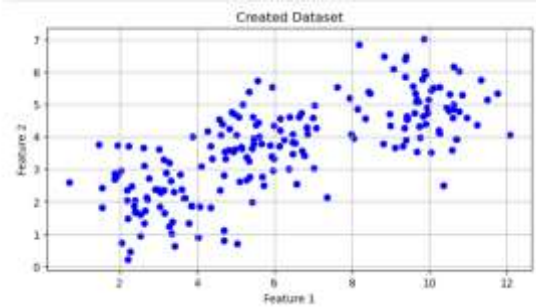
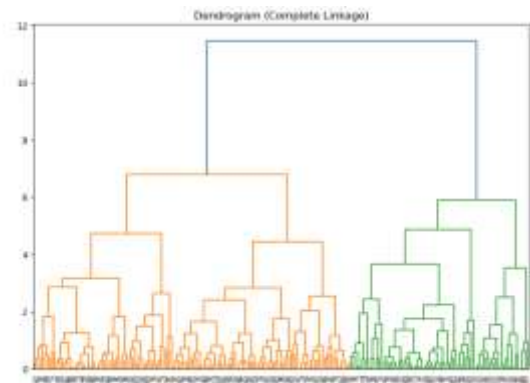
# Agglomerative Hierarchical clustering
agg_clustering = AgglomerativeClustering(linkage='complete', distance_threshold=1.5)
agg_clustering.fit(X)

# Define metrics for clusters
def cluster_metrics(X, y):
    metrics = {}
    for i in range(5):
        cluster = X[y == i]
        metrics[i] = {'size': len(cluster), 'centroid': cluster.mean(0), 'std': cluster.std(0)}
    return metrics

# Plot the clusters with their metrics
plt.figure(figsize=(10, 10))
plt.scatter(X[:, 0], X[:, 1], c=agg_clustering.labels_, s=50, marker='o')
plt.title('Agglomerative Hierarchical Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.grid(True)
plt.show()

# Plot the clusters with their metrics
plt.figure(figsize=(10, 10))
plt.scatter(X[:, 0], X[:, 1], c=agg_clustering.labels_, s=50, marker='o')
plt.title('Agglomerative Hierarchical Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.grid(True)
plt.show()

```



Two crucial unsupervised learning algorithms—K-Means and Agglomerative Hierarchical clustering—are applied practically in this Python code. The method first uses ``make_blobs`` to create a synthetic dataset, from which it generates 200 data points distributed among three different clusters. The dataset is subsequently split into three clusters using the K-Means algorithm, with the centroids prominently shown. This illustrates how K-Means iteratively improves cluster centers. The distinct visual representation, including colored clusters on a black background, highlights the algorithm's capacity to efficiently group related data points.

Afterwards, Agglomerative Hierarchical clustering is implemented by the code, which merges the nearest clusters one after the other to create a hierarchical tree. The whole linkage approach used to compute the distance matrix is projected on a dendrogram, providing a detailed perspective of the clustering hierarchy. The resulting clusters are visualized with unique markers. The relevance of data visualization in evaluating machine learning results is emphasized by this activity, which gives a practical grasp of both clustering algorithms. The understanding of these potent analytical tools is further improved by reinforcing the ideas of cluster centroids, hierarchical linkages, and the significance of linkage criteria in hierarchical clustering.