PAREENITA A.SHIRSATH B.E.A.I.&.D.S. ROLL.NO : 57

BDA EXPERIMENT NO :

```
import numpy as np
from scipy.spatial.distance import euclidean
import matplotlib.pyplot as plt

def select_representative_points(points, num_representatives):
    if len(points) <= num_representatives:
        return points
    indices = np.random.choice(len(points), size=num_representatives, replace=False)
    return points[indices]

def calculate_min_representative_distance(cluster1_reps, cluster2_reps):
    min_dist = float('inf')
    for rep1 in cluster1_reps:
        for rep2 in cluster2_reps:
            dist = euclidean(rep1, rep2)
            min_dist = min(min_dist, dist)
    return min_dist


# Cluster 1 points
cluster1_points = np.array([[0.5, 1], [1.2, 1.5], [0.8, 2]])

# Cluster 2 points
cluster2_points = np.array([[6.5, 4.5], [7.2, 5.5], [5.9, 4.8]])

# Cluster 3 points
cluster3_points = np.array([[2.5, 5], [1.8, 5.2], [2.2, 4.8]])


clusters = {
    "Cluster 1": cluster1_points,
    "Cluster 2": cluster2_points,
    "Cluster 3": cluster3_points
}

num_representatives = 2
cluster_representatives = {}
for name, points in clusters.items():
    cluster_representatives[name] = select_representative_points(points, num_representatives)
    print(f"Representative points for {name}:\n{cluster_representatives[name]}")

print("\nCalculating pairwise distances between representative points:")

min_distances = {}
cluster_names = list(clusters.keys())
for i in range(len(cluster_names)):
    for j in range(i + 1, len(cluster_names)):
        name1 = cluster_names[i]
        name2 = cluster_names[j]
        reps1 = cluster_representatives[name1]
        reps2 = cluster_representatives[name2]
        dist = calculate_min_representative_distance(reps1, reps2)
        min_distances[(name1, name2)] = dist
        print(f"Min distance between {name1} and {name2}: {dist:.2f}")

closest_pair = None
min_overall_dist = float('inf')
for (name1, name2), dist in min_distances.items():
    if dist < min_overall_dist:
        min_overall_dist = dist
        closest_pair = (name1, name2)

print(f"\nClosest clusters to merge: {closest_pair} with a minimum representative distance of {min_overall_dist:.2f}")

plt.figure(figsize=(8, 6))
colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k']
for i, (name, points) in enumerate(clusters.items()):
    plt.scatter(points[:, 0], points[:, 1], color=colors[i % len(colors)], label=name, alpha=0.6)
    reps = cluster_representatives[name]
    plt.scatter(reps[:, 0], reps[:, 1], color=colors[i % len(colors)], marker='D', s=150, edgecolor='black', linewidth=1, label

plt.title("Simplified CURE Demonstration: Initial Clusters and Representative Points")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.legend()
plt.grid(True)
plt.show()
```

```
Representative points for Cluster 1:
[[1.2 1.5]
 [0.5 1. ]]
Representative points for Cluster 2:
[[5.9 4.8]
 [7.2 5.5]]
Representative points for Cluster 3:
[[2.2 4.8]
 [2.5 5. ]]

Calculating pairwise distances between representative points:
Min distance between Cluster 1 and Cluster 2: 5.74
Min distance between Cluster 1 and Cluster 3: 3.45
Min distance between Cluster 2 and Cluster 3: 3.41

Closest clusters to merge: ('Cluster 2', 'Cluster 3') with a minimum representative distance of 3.41
```