Cluster Analysis Report
Author: Tri Lam
ID: 1916079

Task 1: Compute Purity
Objective:
Write a function compute_purity(y_true, y_pred) to calculate the purity of clustering results. The function takes as input the true labels y_true and predicted cluster labels y_pred and returns a single number representing purity. Purity is defined as the sum of the majority class examples of all clusters divided by the total number of data points.
Implementation:

- The compute_purity function calculates purity using the provided formula.
- Example provided in the prompt was used to validate the function:
  - y_pred = [2, 2, 1, 2, 2, 2, 0, 0, 0, 1, 2, 1, 1, 1, 1, 1]
  - y_true = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2]
  - Computed purity: (4 + 3 + 5) / 16 = 0.75.

Results:

- The function was tested with multiple examples and produced correct results in all cases.
- This function is used in subsequent tasks to calculate purity.

Conclusion:
The compute_purity function correctly calculates the purity of clustering results and will be applied to other tasks.

Task 2: K-Means with k=2
Objective:
Run K-Means clustering with k=2. Compute:

1. The percentage of data points assigned to each cluster.
2. The overall purity of the clustering.
3. The purity for each cluster. Identify the cluster with the highest purity.

Implementation:

- K-Means clustering was applied with k=2 using scikit-learn's default parameters.
- Features were normalized using feature_norm.
- Purity and cluster-wise statistics were calculated.

Results:

- Overall Purity: 0.6789
- Percentage of Data Points in Each Cluster:
  - Cluster 0: 51.51 percent
  - Cluster 1: 48.49 percent
- Cluster-wise Purity:
  - Cluster 0: 0.6558
  - Cluster 1: 0.7034
- Highest Purity Cluster: Cluster 1 with a purity of 0.7034.

Conclusion:
Cluster 1 demonstrates higher purity compared to Cluster 0. The overall purity reflects moderate clustering performance.

Task 3: K-Means with Varying k
Objective:
Run K-Means clustering with k=2, 10, 30, 50, 100. For each value of k, run K-Means 10 times and calculate:

1. The average purity across 10 runs.
2. The average silhouette coefficient across 10 runs.
   Identify the value of k that gives the best clustering results in terms of purity and silhouette coefficient.

Implementation:

- For each k, K-Means clustering was run 10 times with different random states.
- Purity and silhouette coefficient were averaged across the 10 runs.
- Silhouette coefficient was computed using Euclidean distance.

Results:
k=2, Avg Purity=0.6789, Avg Silhouette=0.1977
k=10, Avg Purity=0.6789, Avg Silhouette=0.3117
k=30, Avg Purity=0.7037, Avg Silhouette=0.4727
k=50, Avg Purity=0.7334, Avg Silhouette=0.3649
k=100, Avg Purity=0.7963, Avg Silhouette=0.2323

- Best k for Purity: k=100 (purity=0.7963).
- Best k for Silhouette Coefficient: k=30 (silhouette=0.4727).

Conclusion:

- Increasing k improves purity as clusters align more closely with data points.
- Silhouette coefficient peaks at k=30, suggesting this value provides the best trade-off between cluster cohesion and separation.

Task 4: DBSCAN Analysis
Objective:
Run DBSCAN on the normalized data with epsilon=0.3, 0.5, 0.7, fixing minPts=5, using Euclidean distance as the metric. Compute:

1. The total number of clusters.
2. The total number of anomalies.
3. The purity of the clustering. Identify the epsilon that gives the best clustering result in terms of purity.

Implementation:

- Features were normalized using feature_norm.
- DBSCAN clustering was applied with varying epsilon values.
- Purity, number of clusters, and anomalies were computed for each configuration.

Results:
epsilon=0.3, Number of Clusters=18, Number of Anomalies=146, Purity=0.6890
epsilon=0.5, Number of Clusters=22, Number of Anomalies=21, Purity=0.6890
epsilon=0.7, Number of Clusters=22, Number of Anomalies=13, Purity=0.6957

- Best epsilon: epsilon=0.7 (purity=0.6957).

Conclusion:
DBSCAN performs best with epsilon=0.7, producing fewer anomalies and higher purity. Smaller epsilon values lead to more clusters and anomalies, while larger epsilon values decrease anomalies and increase purity slightly.