DS3030: Data Analytics Lab
Assignment 11

Date: Oct 27, 2025

Timing: 2:00 to 4:45 PM Max marks: 12

Instructions

- Part 2 results for WEKA should be demonstrated in the lab itself
- Submit one .ipynb file named as
  [student name]_assignment[number]_part[number].ipynb
  each for Parts 1,3, containing all questions (text blocks) and solutions (code blocks)
- Write **justifications/comments** as required.

Part I: K-Means Clustering (Python)

Total Marks: 3
Dataset: "*BankChurners.csv*"

1. (1 point) **Data Preprocessing & Implementation**

   Load the BankChurners.csv dataset and perform K-Means clustering with $k = 5$ clusters using three different initialization methods: 'k-means++', 'random', and Bisecting K-Means.

   - Load the dataset and select relevant numerical features (at least 4-5 features)
   - Standardize the features using StandardScaler
   - Apply all three initialization methods and display the cluster centers for each

2. (1 point) **Stability & Convergence Analysis**

   - Run 'random' initialization 10 times with different random states
   - Record: mean inertia, standard deviation, and number of iterations to converge
   - Compare the stability and convergence speed with k-means++
   - Create a box plot showing inertia distribution across runs

3. (1 point) **Visualization and Business Insights**

   - Create a 2D visualization using PCA to reduce dimensions (plot PC1 vs PC2)
   - Show clustering results for all three methods in separate subplots

Part II: Hierarchical Clustering (WEKA)

Total Marks: 4
Dataset: "*iris.arff*"

1. (0.5 points) Load the dataset and visualize the ground truth of the number of instances in each class.

2. (1.5 points) Perform Hierarchical Clustering on the dataset.

   - Choose the appropriate values for "*numClusters*" (refer to ground truth)
   - Visualize the clustering results (verify it with the ground truth)
   - Visualize the associated dendrogram tree and give your interpretation

3. (2.0 points) Compare and interpret the clustering results and dendrogram tree by changing

   - "*linkage type*" : Complete, Centroid
   - "*Distance Function*" : Euclidean, Manhattan

Part III: KMeans and DBSCAN Clustering Comparison (Python)

Total Marks: 4
Libraries: `sklearn, numpy, matplotlib`

1. (1 point) **Q1. Synthetic Dataset with Equal-Density Clusters**

   Generate a synthetic dataset with three clusters of roughly the same density and similar number of points per cluster. (**Parameters:** `n_samples=600, centers=3, cluster_std=0.6, random_state=42`)

   Tasks:

   - Apply **KMeans** clustering on this dataset and report the **SSE** using the built-in `inertia_` attribute.
   - Apply **DBSCAN** clustering and compute **SSE**.
   - Plot the clusters obtained.

2. (1 point) **Q2. Synthetic Dataset with Varying-Density Clusters**

   Generate a synthetic dataset with three clusters of varying densities and/or sizes, so that some clusters are tight and others are more spread out. (**Parameters:** `n_samples=[200, 300, 100], centers=[[0, 0], [5, 5], [0, 5]], cluster_std=[0.2, 0.8, 1.5], random_state=42`)

   Tasks:

   - Apply **KMeans** clustering on this dataset and report the **SSE**.
   - Apply **DBSCAN** clustering and compute **SSE**.
   - Plot the clusters obtained.

3. (2.0 points) **Q3. Observation & Conclusion**

   Based on your results from Q1 and Q2:

   - Report which algorithm performs better for the **Q1**.
   - Report which algorithm performs better for the **Q2**.
   - Explain why there is a difference in performance.