Lab Assignment 8: Wheat Seed Clustering using PAM Algorithm

Date: 16-04-2025 **Duration: 2 hours**

Objective: Use the Wheat Seeds Dataset to apply the Partitioning Around Medoids (PAM) clustering algorithm and evaluate how well it clusters different types of wheat seeds based on their physical features.

Dataset: https://archive.ics.uci.edu/dataset/236/seeds

1. Data Loading and Exploration (15 minutes)

- Task: Load the dataset and perform initial exploratory analysis.
    - Load the dataset into a pandas DataFrame and display the first few rows.
    - Check for missing values, and handle them either by imputation or removal.
    - Visualize the distribution of each feature and check for any patterns or anomalies.

2. Data Preprocessing (20 minutes)

- Task: Prepare the data for clustering by scaling and reducing dimensionality.
    - Normalization: Use StandardScaler or MinMaxScaler to scale numerical features (Area, Perimeter, etc.) to bring all features to the same scale.
    - Dimensionality Reduction: Apply Principal Component Analysis (PCA) to reduce the dataset to 2 components, retaining at least 95% of the variance. Visualize the explained variance ratio to ensure proper dimensionality reduction.

3. Clustering with PAM Algorithm (40 minutes)

- Task: Perform clustering on the dataset using the Partitioning Around Medoids (PAM) algorithm.
    - Implement the clustering using the pyclustering library with k = 3 (since there are three classes of wheat).

- Visualize the clustering results using the 2D PCA-transformed dataset. Plot the data points and color them based on their assigned clusters.

- Optional: Overlay the true class labels on the plot to compare the clustering result with actual class labels.

4. Evaluation and Comparison (25 minutes)

- Task: Evaluate the clustering results using several metrics.

    - Use Silhouette Score, Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI) to assess clustering performance.

    - Generate a Confusion Matrix comparing the clustering results with the true labels.

    - Display the evaluation metrics and discuss the clustering performance.

5. Report Findings (20 minutes)

- Task: Write a concise report summarizing the results and insights.

    - Discuss the clustering performance, referencing the metrics calculated in the previous section.

    - Highlight any challenges encountered during the preprocessing or clustering stages.

    - Suggest possible improvements, such as using different clustering algorithms or tuning the parameters.

● **Code File:** Submit the Python script or notebook (.py or .ipynb) with explanations in comments. ● **Report:** A short report (about 200 words) summarizing your process, results, and any challenges encountered.

Note: Comment your code to explain the logic behind each step.