Machine Learning and Pattern Recognition
Lab 2: Spring Semester 2026

Note: This is an individual assignment. You may not consult your peers or AI tools to do these tasks except where explicitly asked for. Any misconduct will result in a 0 score in this entire assignment and will be noted and reported to the Academic Integrity Committee.

Write a code that generates random data and cluster the datapoints using the K-Means algorithm. For a range of value of K from 1 to 20, compute the within-cluster sum of squares distance (WCSS) to find the optimal number of clusters using the Elbow method.
K Means Clustering | Step-by-Step Tutorials For Data Analysis (analyticsvidhya.com)

The output of the code should be:
a) Elbow Plot showing that 4 clusters are optimal for data generated below
b) The clustered data for 4 clusters

Step 1: Import Necessary Libraries

- Numpy
- Matplotlib
- Kmeans from Sklearn (***from sklearn.cluster import KMeans***)

Step 2: Generate synthetic/random data.

- Define the size of the data.
- Use ***np.random.seed(0)*** for reproducibility of random data generation. It would generate same random data on every code run thus provide consistency in the results to compare at different iterations.

Step 3: Define mean (mu) and covariance (Sigma) matrices for each cluster.

- Keep values of mean and covariance as given below for the four gaussian distribution.

  mu1 = [2, 2]
  sigma1 = [[0.9, -0.0255], [-0.0255, 0.9]]

  mu2 = [5, 5]
  sigma2 = [[0.5, 0], [0, 0.3]]

  mu3 = [-2, -2]
  sigma3 = [[1, 0], [0, 0.9]]

  mu4 = [-4, 8]
  sigma4 = [[0.8, 0], [0, 0.6]]

Step 4: Generate synthetic data by drawing samples from each distribution.

- Generate synthetic data using ***np.random.multivariate_normal().***

numpy.random.multivariate_normal — NumPy v1.25 Manual

- Stack data points drawn from the four gaussian distributions using ***np.vstack( ).***

Step 5: Initiate Within-Cluster Sum of Squares (WCSS) and store in empty list.

- Loop through a range of cluster counts from let's say 1 to 20.
- Create a K-Means instance with the current cluster count (keep ***random_state=0***) and fit it to generated data. Use ***Kmeans.fit()***
- Find WCSS for each iteration of K from 1 to 20.

Step 6: Perform K-Means Clustering

- Perform k-means clustering with optimal number of clusters that is given by the Elbow Method plot (should be 4)

Step 7: Visualize the Data and Clustering Results

- Data Plot
- Elbow Method Plot
- Clustered Data Plot

Report:

Answer the following questions within your report:

Q1: What might happen if the value of the number of clusters is set too high or too low in K-Means?

Q2: What is the effect of initializing centroids far apart or too close together in K-Means, and how might this affect the final clustering?

Q3: How might you optimise the centroid initialisation to lead to better clustering?

Q4: Why might the Elbow Method not always provide a clear solution for choosing the optimal number of clusters?

Q5: How can WCSS be influenced by the presence of outliers or noise in the data?

Submission Instructions:

- Items to be uploaded on LMS: Main code file along with the outputs and the report in one file in PDF format. Ensure that your file contains the following plots:
    - Data Plot (PNG/JPG)
    - Elbow Method plot (PNG/JPG)
    - Clustered Data Plot (PNG/JPG)

- You can use any IDEs you want.

- The file should specify the steps taken, followed by observations and answering the question as specified in the assignment.

- Write your comments along with the code for each step as required.

- Your file should be as follows. **Yourname_lab2.pdf**

- Due time and date are given on LMS. Submit it before the deadline.

Output reference: