LAB 3: UNSUPERVISED LEARNING

(Duration: 2 sessions)

(PART A: CLUSTERING)

Exercise 1: Manual K-Means Clustering on a 2D Dataset (Implement K-Means from Scratch)

You are given the following 6 points in 2D space:

Point	x	y
A	1	1
B	1	2
C	1	3
D	5	1
E	5	2
F	5	3

Assume you want to cluster these points into **k = 2 clusters** using the **K-Means algorithm**.

1. **Distance Computation and Assign Points**

   You are given the initial centroids: Centroid 1: (1, 2), Centroid 2: (5, 2)

   a) Use the Euclidean distance to compute the distance from each point to both centroids.

   b) Assign each point to the nearest centroid.

   c) Complete the following table:

Point	Distance to Centroid 1	Distance to Centroid 2	Assigned Cluster
A			
B			
C			
D			
E			
F			

2. **Update Centroids**

   a) Compute the new centroids by averaging the coordinates of the points assigned to each cluster.

   b) Report the new centroid positions

   c) Repeat the process until convergence

3. **Second Initialization (Problematic)**

   Now, repeat the process using the following alternative centroids: Centroid 1: (2.5, 1.5) and Centroid 2: (2.5, 2.5)

   a) Again, compute the distances and assign points.

   b) Compute the new centroids and check for convergence.

   c) Compare the final clusters from this run with those from the first initialization.

4. **Reflection**
    a) Are the final clusters from both initializations the same? If not, explain why.
    b) How does the second initialization illustrate a weakness of K-Means?
    c) What does this example show about K-Means' sensitivity to initialization?
    d) What methods exist to reduce the risk of bad initialization?
    e) Can you say that K-Means always finds the global optimum? Justify with your results.
    f) If you added an outlier to the dataset (e.g., point G = (9, 1)), how might it affect the result?

Exercise 2 – K-Means and Initialization Impact

1. **Dataset Loading**
    a. Download the dataset and load it [link].
    b. Visualize the dataset using a scatter plot.
    c. Describe the expected clustering structure.
2. **Clustering with Random Initialization**
    a) Apply the K-Means algorithm with the following parameters:
        - n_clusters = 3, init = 'random', n_init = 1, random_state = 0
    b) Print the predicted labels and final centroids.
    c) Plot the clustering result and centroids.
    d) Compute and report the inertia.
3. **Clustering with K-Means++ Initialization**
    a) Repeat the same clustering task but using:
        - init = 'k-means++', n_init = 10, random_state = 0
    b) Again, print the predicted labels, centroids, and inertia.
    c) Plot the clustering result.
4. **Comparison and Reflection**
    a) Are the cluster assignments the same in both cases?
    b) Which method resulted in the lowest inertia? Why?
    c) How do the centroid positions differ between the two runs?
    d) Explain how k-means++ improves clustering reliability.
    e) Try running the same code with several random_state values using init='random'. Does the result always look good? Why not?

Exercise 3 – Customer Segmentation Using Clustering Techniques

1. Load and Explore the Dataset
    a. Load the dataset Mall_Customers.csv (use this link to download it).
    b. Keep the following features only: Age, Annual Income (k$), Spending Score (1-100).
    c. What types of variables are these?
    d. Why might it be necessary to scale these features before clustering?
2. Visualize the Raw Data
    a. Create a 2D scatter plot: (x-axis = Annual Income, y-axis = Spending Score)

   b. What patterns do you observe in this plot?
   c. Based on the visual, how many clusters would you expect?

3. Apply K-Means Clustering
   a. Apply KMeans(n_clusters=5) to the scaled data.
   b. Plot the results and show the cluster centroids.
   c. What does each cluster seem to represent in marketing terms?
   d. Are any clusters significantly larger or more compact than others?

4. Evaluate K-Means Performance
   a. Compute the Silhouette Score.
   b. Is the score close to 1 (well-separated) or near 0 (overlapping)?
   c. What are the limits of using this score in unsupervised learning?

5. Apply DBSCAN
   a. Apply DBSCAN to the same dataset (experiment with eps and min_samples).
   b. Plot the result, highlighting noise points (label = -1).
   c. How many clusters were found?
   d. How does DBSCAN handle outliers compared to K-Means?
   e. Do the clusters have different shapes?

6. Apply Agglomerative Clustering
   a. Apply Agglomerative Clustering with n_clusters=5 and 'ward' linkage.
   b. Create a dendrogram (optional).
   c. How do the resulting clusters compare to those from K-Means and DBSCAN?
   d. Which method gives the most stable and intuitive segmentation?

7. Comparative Analysis
   a. Which clustering method gives the most useful segmentation from a business perspective?
   b. Which method is most robust to scale, shape, and outliers?

(Part B: Dimensionality Reduction)

Exercise 1 – Why Reduce Dimensions?

1. Create synthetic datasets in 2D, 10D, and 100D (e.g., using np.random.randn()).
2. For each dataset, compute the pairwise distances between points using Euclidean distance.
3. Plot the histogram of distances for each dimensionality.
4. What do you observe as dimensionality increases?
5. Why are distances becoming more similar in high dimensions?
6. What problems might this cause in clustering or classification?

Exercise 2 – PCA on a Real Dataset

1. Load the Iris dataset and remove the labels.
2. Standardize the data.
3. Apply PCA and keep only 2 components.

4. Plot the dataset using the 2 principal components.
5. How much variance is preserved with the first 2 components?
6. Do you observe visible clusters or patterns?
7. What does the direction of the components represent?

Exercise 3 – How Many Components to Keep?

1. Load the Wine or Breast Cancer dataset.
2. Standardize the features and apply PCA with `n_components=None`.
3. Plot:
    a. Scree plot (explained variance by component)
    b. Cumulative variance
4. How many components do you need to keep 95% of the variance?
5. Why is this useful when working with large feature sets?
6. Would you use all components if you only need to cluster the data?

Exercise 4 – PCA Before Clustering

1. Take the Digits dataset (sklearn.datasets.load_digits).
2. Apply PCA to reduce dimensions to 10, then to 2.
3. Apply K-Means clustering on both PCA versions.
4. Visualization and Analysis:
    a. Plot the 2D clusters.
    b. Do clusters look more or less clear after PCA?
    c. How does PCA affect the performance of K-Means?
    d. Could PCA remove important clustering information?

(Part C: Anomaly Detection)

Exercise 1 – Isolation Forest on Real Data

1. Load the Breast Cancer or Wine Quality dataset.
2. Apply `IsolationForest` from scikit-learn.
3. Plot the anomaly scores and mark which samples were flagged as outliers.
4. How does Isolation Forest isolate anomalies?
5. What is the meaning of the "anomaly score"?
6. How would you adjust the model to detect fewer or more anomalies?