

**Assignment Code: DA-AG-017**

Clustering | **Assignment**

**Instructions:** Carefully read each question. Use Google Docs, Microsoft Word, or a similar tool to create a document where you type out each question along with its answer. Save the document as a PDF, and then upload it to the LMS. Please do not zip or archive the files before uploading them. Each question carries 20 marks.

**Total Marks**: 200

**Question 1:** What is the difference between K-Means and Hierarchical Clustering? Provide a use case for each.

**Answer:**

| K-Means is a centroid-based algorithm that partitions data into *k* clusters, minimizing within-cluster variance. It is efficient for large datasets but requires the number of clusters beforehand. Hierarchical clustering builds a tree-like structure (dendrogram) without needing the number of clusters initially, making it more interpretable but computationally heavy for large data.   * Use case K-Means: Customer segmentation for marketing. * Use case Hierarchical: Document or gene similarity analysis. |
| --- |

**Question 2:** Explain the purpose of the Silhouette Score in evaluating clustering algorithms.

**Answer:**

| The Silhouette Score measures how similar a point is to its own cluster compared to other clusters. It ranges from -1 to 1, where higher values mean well-separated clusters. A value near 1 indicates good clustering, 0 means overlapping clusters, and negative values indicate wrong clustering. It is useful to compare different clustering algorithms and to validate the number of clusters chosen. |
| --- |

1



**Question 3:** What are the core parameters of DBSCAN, and how do they influence the clustering process?

**Answer:**

| The two main parameters are eps (ε) and min\_samples.   * eps defines the neighborhood radius around a point. * min\_samples sets the minimum points needed to form a dense region.  If a point has at least min\_samples within eps, it becomes a core point. Points close to core points are border points, while others are labeled as noise (outliers). Together, these parameters control cluster shape and density detection. |
| --- |

**Question 4:** Why is feature scaling important when applying clustering algorithms like K-Means and DBSCAN?

**Answer:**

| Clustering algorithms like K-Means and DBSCAN rely on distance measures (Euclidean or Manhattan). If features are on different scales (e.g., age in years vs income in lakhs), larger values dominate, leading to biased clusters. Standardization or normalization ensures equal weight for all features, improving accuracy and stability of clusters. |
| --- |

**Question 5:** What is the Elbow Method in K-Means clustering and how does it help determine the optimal number of clusters?

**Answer:**

| The Elbow Method is a heuristic used in K-Means to find the optimal number of clusters. It involves plotting the Within-Cluster Sum of Squares (WCSS) against different values of k. Initially, WCSS decreases sharply as k increases, but after a certain point (“elbow”), the improvement slows down. The elbow point represents the best trade-off between cluster compactness and simplicity. |
| --- |

2



**Dataset:**

Use make\_blobs, make\_moons, and sklearn.datasets.load\_wine() as specified.

**Question 6:** Generate synthetic data using make\_blobs(n\_samples=300, centers=4), apply KMeans clustering, and visualize the results with cluster centers.

(*Include your Python code and output in the code box below.*)

**Answer:**

| from sklearn.datasets import make\_blobs  from sklearn.cluster import KMeans  import matplotlib.pyplot as plt  X, y = make\_blobs(n\_samples=300, centers=4, random\_state=42)  kmeans = KMeans(n\_clusters=4, random\_state=42).fit(X)  plt.scatter(X[:,0], X[:,1], c=kmeans.labels\_, cmap='viridis')  plt.scatter(kmeans.cluster\_centers\_[:,0], kmeans.cluster\_centers\_[:,1],  c='red', marker='X', s=200, label='Centers')  plt.legend()  plt.show()  **OUTPUT:** |
| --- |

**Question 7**: Load the Wine dataset, apply StandardScaler , and then train a DBSCAN model. Print the number of clusters found (excluding noise).

(*Include your Python code and output in the code box below.*)

**Answer:**

| from sklearn.datasets import load\_wine  from sklearn.preprocessing import StandardScaler  from sklearn.cluster import DBSCAN  wine = load\_wine()  X = StandardScaler().fit\_transform(wine.data)  db = DBSCAN(eps=1.5, min\_samples=5).fit(X)  labels = db.labels\_  n\_clusters = len(set(labels)) - (1 if -1 in labels else 0)  print("Number of clusters found:", n\_clusters)  **OUTPUT:**  Number of clusters found: 0 |
| --- |

**Question 8**: Generate moon-shaped synthetic data using

make\_moons(n\_samples=200, noise=0.1), apply DBSCAN, and highlight the outliers in the plot.

(*Include your Python code and output in the code box below.*)

**Answer:**

3



| from sklearn.datasets import make\_moons  import numpy as np  X, y = make\_moons(n\_samples=200, noise=0.1, random\_state=42)  db = DBSCAN(eps=0.3, min\_samples=5).fit(X)  plt.scatter(X[:,0], X[:,1], c=db.labels\_, cmap='plasma')  plt.scatter(X[db.labels\_==-1,0], X[db.labels\_==-1,1], c='red', marker='x', label='Outliers')  plt.legend()  plt.show()  OUTPUT: |
| --- |

**Question 9**: Load the Wine dataset, reduce it to 2D using PCA, then apply Agglomerative Clustering and visualize the result in 2D with a scatter plot. (*Include your Python code and output in the code box below.*)

**Answer:**

| from sklearn.decomposition import PCA  from sklearn.cluster import AgglomerativeClustering  X = StandardScaler().fit\_transform(wine.data)  X\_pca = PCA(n\_components=2).fit\_transform(X)  agg = AgglomerativeClustering(n\_clusters=3).fit(X\_pca)  plt.scatter(X\_pca[:,0], X\_pca[:,1], c=agg.labels\_, cmap='rainbow')  plt.show()  **OUTPUT:** |
| --- |

**Question 10:** You are working as a data analyst at an e-commerce company. The marketing team wants to segment customers based on their purchasing behavior to run targeted promotions. The dataset contains customer demographics and their product purchase history across categories.

Describe your real-world data science workflow using clustering:

● Which clustering algorithm(s) would you use and why?

● How would you preprocess the data (missing values, scaling)?

● How would you determine the number of clusters?

● How would the marketing team benefit from your clustering analysis? (*Include your Python code and output in the code box below.*)

**Answer:**

4



| **Algorithm Choice**: Use K-Means for large datasets, DBSCAN for noise/outliers, and Hierarchical for interpretability.  **Preprocessing**: Handle missing values (imputation), normalize data with StandardScaler.  **Cluster Number Selection**: Apply Elbow method, Silhouette score, or dendrogram for hierarchical clustering.  **Benefit to Marketing**: Segments customers into meaningful groups (e.g., high spenders, discount seekers, loyal buyers). This enables personalized promotions, improves customer retention, and enhances sales strategy through data-driven targeting. |
| --- |

5