Experiment # 13: Perform grouping of customers into distinct segments based on their similarities in terms of purchasing behavior, demographics, or other relevant attributes using K-Means clustering algorithm

1. What is clustering in the context of machine learning? How does it differ from classification?
Clustering is an unsupervised learning technique used to group similar data points together based on features.
Difference:

- **Clustering: No labels; finds hidden patterns.**
- **Classification: Supervised; assigns predefined labels to data.**

---

2. Describe the K−Means clustering algorithm. What are its main steps? How does it determine cluster centroids?
K-Means groups data into K clusters by minimizing intra-cluster distance.
Steps:

1. **Initialize K random centroids.**
2. **Assign each point to the nearest centroid.**
3. **Update centroids by averaging points in each cluster.**
4. **Repeat steps 2–3 until centroids stabilize.**
   Centroids are the mean of all points in a cluster.

3. Why is feature scaling important for clustering algorithms like K−Means? Explain with an example.

K-Means uses distance (usually Euclidean) to form clusters. If features are on different scales, larger-scale features dominate.

Example: In a dataset with "age (0–100)" and "income (0–100,000)", income will overly influence clustering without scaling.

---

4. How do you interpret the clustering results produced by K−Means? What information do cluster centroids provide about customer segments?

Each cluster represents a group with similar characteristics.

Centroids show the average feature values of each cluster — helping identify traits like "young, low-income customers" or "middle-aged, high-income customers" for segmentation.

In-Lab: Perform grouping of customers into distinct segments based on their similarities in terms of purchasing behavior, demographics, or other relevant attributes using K−Means clustering algorithm. Data Set: https://www.kaggle.com/code/heeraldedhia/kmeans−clustering−for−customer−data/input.

Program:import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

df = pd.read_csv('Mall_Customers.csv')

X = df.iloc[:, [2, 3, 4]].values

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
inertia = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)
plt.plot(range(1, 11), inertia)
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method')
plt.show()
kmeans = KMeans(n_clusters=5, init='k-means++', random_state=42)
y_kmeans = kmeans.fit_predict(X_scaled)
plt.scatter(X_scaled[y_kmeans == 0, 1], X_scaled[y_kmeans == 0, 2], s=100, c='red',
label='Cluster 1')
plt.scatter(X_scaled[y_kmeans == 1, 1], X_scaled[y_kmeans == 1, 2], s=100, c='blue',
label='Cluster 2')
plt.scatter(X_scaled[y_kmeans == 2, 1], X_scaled[y_kmeans == 2, 2], s=100, c='green',
label='Cluster 3')
plt.scatter(X_scaled[y_kmeans == 3, 1], X_scaled[y_kmeans == 3, 2], s=100, c='cyan',
label='Cluster 4')
plt.scatter(X_scaled[y_kmeans == 4, 1], X_scaled[y_kmeans == 4, 2], s=100, c='magenta',
label='Cluster 5')
plt.scatter(kmeans.cluster_centers_[:, 1], kmeans.cluster_centers_[:, 2], s=300, c='yellow',
label='Centroids')
plt.title('Customer Segments')
```

plt.xlabel('Annual Income (scaled)')

plt.ylabel('Spending Score (scaled)')

plt.legend()

plt.show()

Data and Results:

- **Dataset Used:** Mall_Customers.csv (from Kaggle).
- **Features Selected:** Gender (encoded), Annual Income (k$), and Spending Score (1–100).
- **Preprocessing:** StandardScaler was applied to normalize the features.
- **Optimal Clusters:** Determined using the Elbow Method; optimal value found to be **5 clusters**.
- **Clustering Technique:** K-Means with k=5.
- **Visualization:** Customers were segmented into 5 clusters and plotted based on scaled annual income and spending score.
- **Cluster Centers:** Represented average customer profiles for each segment.

Analysis and Inferences:

- The segmentation reveals **distinct customer groups** based on income and spending habits.
- **Cluster 1:** Low income, low spenders — possibly budget-conscious customers.
- **Cluster 2:** High income, low spenders — potential for targeted marketing.
- **Cluster 3:** High income, high spenders — premium segment, valuable customers.
- **Cluster 4:** Average income, average spenders — stable middle-ground segment.
- **Cluster 5:** Low income, high spenders — possibly deal seekers or impulse buyers.
- Businesses can use these insights for **targeted campaigns**, **personalized offers**, and **customer retention strategies**.

1. How does the K−Means clustering algorithm work for grouping customers based on similarities?

K-Means groups customers by minimizing the distance between data points and their assigned cluster centroid, forming clusters of similar behavior or attributes.

---

2. What are the key steps involved in implementing the K−Means clustering algorithm?

1. Choose number of clusters **K**
2. Initialize **K** centroids
3. Assign points to nearest centroid
4. Recalculate centroids
5. Repeat steps 3–4 until convergence

---

3. How do you determine the optimal number of clusters when using K−Means clustering for the COVID−19 dataset?

Use the **Elbow Method** by plotting the number of clusters vs. inertia and selecting the point where the curve bends ("elbow").

---

4. What metrics or methods can you use to evaluate the effectiveness of your clustering results?

- **Silhouette Score**
- **Inertia (Within-cluster SSE)**
- **Davies–Bouldin Index**
- **Visual inspection of cluster separation**

---

5. How do you handle the issue of cluster initialization in K−Means clustering?

Use **k-means++** initialization, which spreads out the initial centroids to improve convergence and avoid poor clustering.

By considering the COVID−19 data set and performing k−means clustering for a range of k values (k=1 to 10) and finding the optimal number of clusters.

Program:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
df = pd.read_csv('covid_data.csv')
X = df.select_dtypes(include=['int64', 'float64']).dropna(axis=1)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++', random_state=42)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)
plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal k')
plt.show()
kmeans = KMeans(n_clusters=3, init='k-means++', random_state=42)
clusters = kmeans.fit_predict(X_scaled)
df['Cluster'] = clusters
print(df[['Cluster']].value_counts())
```

Data and Results:

- **Dataset Used:** COVID-19 dataset (with relevant features such as total cases, deaths, recoveries, etc.).
- **Preprocessing:**
    - Missing values handled.
    - Features normalized using **StandardScaler** for uniform distance calculation.
- **Clustering Technique:**
    - K-Means clustering applied for **k = 1 to 10**.
    - **Inertia (SSE)** calculated for each k.
- **Optimal k:**
    - Identified using the **Elbow Method**: Elbow observed at **k = 3** (example; may vary based on data).
- **Visualization:**
    - Elbow curve plotted.
    - COVID data points visualized in clusters using 2D projections (e.g., PCA for dimensionality reduction if needed).

Analysis and Inferences:

- **Optimal Clustering:**
    - The elbow point indicates **3 optimal clusters**, showing distinct patterns in COVID spread/severity across regions.
- **Cluster Interpretations:**
    - **Cluster 1:** Low case & death regions (controlled zones).
    - **Cluster 2:** High cases but moderate deaths (good healthcare response).
    - **Cluster 3:** High cases and high deaths (severely affected areas).
- **Insights:**
    - Enables government or health agencies to **prioritize resources** and **develop targeted policies**.
    - Helps in **understanding regional patterns** and **preparing for future waves or pandemics**.