1. **What is K-Means Clustering?**
   * K-Means is an unsupervised learning algorithm that groups data into a specified number of clusters (k). It minimizes the variance within each cluster.
2. **How do you determine the optimal number of clusters in K-Means?**
   * The **Elbow Method** is used, which involves plotting the inertia (sum of squared distances) against the number of clusters and finding the 'elbow' point where the decrease in inertia slows.
3. **What is Hierarchical Clustering?**
   * It is a method of clustering that builds a hierarchy of clusters using an agglomerative (bottom-up) or divisive (top-down) approach.
4. **What is the role of the Dendrogram in Hierarchical Clustering?**
   * The dendrogram visually represents how clusters are formed step by step. The vertical line's height shows the distance at which clusters are merged, helping to choose the number of clusters.
5. **What is the difference between K-Means and Hierarchical Clustering?**
   * K-Means requires the number of clusters (k) as input, while Hierarchical Clustering doesn't. Hierarchical clustering is more interpretable but computationally intensive for large datasets.
6. **How does K-Means initialization work?**
   * K-Means++ initializes centroids to be as far apart as possible, improving convergence and avoiding poor clustering due to random initialization.
7. **What is Inertia in K-Means?**
   * Inertia measures the sum of squared distances between each data point and its assigned cluster center. It indicates the compactness of clusters; lower inertia means better clustering.
8. **What is the 'Ward' method in Hierarchical Clustering?**
   * The 'Ward' method minimizes the variance within each cluster when merging. It selects pairs of clusters to merge based on minimum variance.
9. **Why is normalization important before clustering?**
   * Features may have different scales (e.g., income in thousands vs. age in years). Normalization ensures each feature contributes equally to the clustering process.
10. **Can K-Means handle outliers?**

* K-Means is sensitive to outliers because they can distort the cluster centroids and result in poor clustering.

**What is Clustering?**

* **Clustering** is an unsupervised machine learning technique used to group similar data points together. It divides the dataset into subsets (clusters), where points in the same cluster are more similar to each other than to points in other clusters.

**2. Where is Clustering used in real life?**

* Clustering is used in **market segmentation** to group customers based on their behavior, in **image segmentation** to identify distinct objects in an image, in **social network analysis** to find communities, and in **anomaly detection** to identify unusual patterns in data like fraud detection.

**3. When to use K-Means vs K-Medians?**

* **K-Means** is used when the dataset is relatively free of outliers and follows a normal distribution, as it minimizes the average squared distance. **K-Medians** is better when there are outliers, as it minimizes the sum of absolute differences, making it more robust to anomalies.

**4. Which methods belong to clustering?**

* Common clustering methods include:
  + **K-Means Clustering**
  + **Hierarchical Clustering**
  + **DBSCAN (Density-Based Spatial Clustering)**
  + **Gaussian Mixture Models (GMM)**
  + **Agglomerative Clustering**

**5. What is the difference between Classification and Clustering?**

* **Classification** is a supervised learning method where the model is trained on labeled data to predict the category of new data. **Clustering**, on the other hand, is unsupervised and groups data based on similarity without pre-existing labels.

**6. Which Clustering Algorithm is best?**

* The best clustering algorithm depends on the dataset:
  + **K-Means** is efficient for large datasets with spherical clusters.
  + **DBSCAN** is effective for datasets with noise and clusters of varying shapes.
  + **Hierarchical Clustering** is better for small to medium-sized datasets requiring a hierarchy of clusters.

**7. Which algorithm is used by clustering in machine learning?**

* Common algorithms used for clustering in machine learning include **K-Means**, **DBSCAN**, **Agglomerative Hierarchical Clustering**, and **Gaussian Mixture Models (GMM)**. Each algorithm has its specific use cases and is chosen based on the nature of the dataset.

# Importing necessary libraries

import warnings

warnings.filterwarnings('ignore') # Ignores warnings to keep output clean

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Reading the dataset

df = pd.read\_csv('4.csv')

df.head() # Displays the first 5 rows of the dataframe

# Getting information about the dataset

df.info() # Provides summary info about the dataset, including data types and non-null counts

# Plotting distribution of features

plt.style.use("fivethirtyeight")

plt.figure(1, (15, 8))

n = 0

for x in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:

n += 1

plt.subplot(1, 3, n)

plt.subplots\_adjust(hspace=0.5, wspace=0.5)

sns.histplot(df[x], bins=40) # Histogram for age, annual income, and spending score

plt.show()

# Gender Count Plot

plt.figure(1, (15, 5))

sns.countplot(y="Gender", data=df) # Shows the count of Male and Female customers

plt.show()

# Pairwise Regression Plots

plt.style.use("fivethirtyeight")

plt.figure(1, (15, 7))

n = 0

for x in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:

for y in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:

n += 1

plt.subplot(3, 3, n)

plt.subplots\_adjust(hspace=0.5, wspace=0.5)

sns.regplot(x=x, y=y, data=df)

plt.show()

# Checking for missing values

df.isnull().sum() # Returns the count of missing values for each column

# Checking for duplicate values

df.drop\_duplicates(inplace=True) # Removes duplicate rows

# Scatter Plot for Income vs Spending Score

plt.scatter(df['Annual Income (k$)'], df['Spending Score (1-100)'])

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.show()

# Extracting features for clustering

X = df.iloc[:, [3, 4]].values # Selecting 'Annual Income' and 'Spending Score'

# K-Means Clustering

from sklearn.cluster import KMeans

kmeans = KMeans(n\_clusters=5, init='k-means++', random\_state=42)

y\_kmeans = kmeans.fit\_predict(X) # Predicting clusters

# Visualizing K-Means Clusters

for i in range(5):

sns.scatterplot(x=X[y\_kmeans == i, 0], y=X[y\_kmeans == i, 1], label=f'Cluster {i+1}')

sns.scatterplot(x=kmeans.cluster\_centers\_[:, 0], y=kmeans.cluster\_centers\_[:, 1], color='red', marker='\*', s=300)

plt.title('Clusters of Customers')

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.legend()

plt.show()

# Elbow Method for Optimal Number of Clusters

cluster = []

for k in range(1, 11):

kmean = KMeans(n\_clusters=k).fit(X)

cluster.append(kmean.inertia\_)

plt.plot(range(1, 11), cluster, 'r-')

plt.xlabel('Number of Clusters')

plt.ylabel('Inertia')

plt.title('Elbow Method')

plt.show()

# Hierarchical Clustering Dendrogram

import scipy.cluster.hierarchy as sch

dendrogram = sch.dendrogram(sch.linkage(X, method='ward'))

plt.title("Dendrogram")

plt.xlabel("Customers")

plt.ylabel("Euclidean Distance")

plt.show()

# Agglomerative Hierarchical Clustering

from sklearn.cluster import AgglomerativeClustering

hc = AgglomerativeClustering(n\_clusters=5, affinity="euclidean", linkage='ward')

y\_pred = hc.fit\_predict(X)

# Visualizing Hierarchical Clusters

for i in range(5):

plt.scatter(X[y\_pred == i, 0], X[y\_pred == i, 1], label=f'Cluster {i+1}')

plt.xlabel('Annual Income')

plt.ylabel('Spending Score')

plt.legend()

plt.title('Clusters of Customers')

plt.show()