Part 1: Conceptual Questions

Question 1: What is the difference between K-Means and Hierarchical Clustering?

Provide a use case for each.
Answer:

- **K-Means Clustering:** An algorithm that partitions data into k predefined clusters by minimizing the distance between data points and their respective cluster centers. It is generally faster and more efficient for large datasets.
    - **Use Case:** Segmenting customers into three distinct groups (e.g., Low, Medium, High spenders) for a marketing campaign.
- **Hierarchical Clustering:** A method that builds a tree-like hierarchy of clusters (a dendrogram) either from the bottom-up (agglomerative) or top-down (divisive). It does not require the number of clusters to be specified upfront.
    - **Use Case:** Building a taxonomy of animal species or organizing a library of documents into nested categories.

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

Answer: The Silhouette Score measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

- The score ranges from **-1 to +1**.
- A **high score** indicates that the object is well-matched to its own cluster and poorly matched to neighboring clusters.
- It helps in determining the optimal number of clusters and comparing the performance of different algorithms.

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

Answer: DBSCAN (Density-Based Spatial Clustering of Applications with Noise) relies on two primary parameters:

1. **eps (ϵ):** Specifies the maximum distance between two samples for one to be considered as in the neighborhood of the other. If ϵ is too small, most data will be marked as noise; if too large, clusters will merge.
2. **min_samples:** The number of samples in a neighborhood for a point to be considered a "core point." Higher values require denser regions to form a cluster.

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

Answer: Most clustering algorithms calculate the **Euclidean distance** between points to determine similarity.

- If features have different scales (e.g., Age 0–100 vs. Annual Income 0–1,000,000), the feature with the larger range will dominate the distance calculation.
- Scaling (like StandardScaler) ensures all features contribute equally to the distance metric, leading to more accurate and meaningful 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 involves plotting the **Within-Cluster Sum of Squares (WCSS)** or inertia against the number of clusters (k).

- As k increases, WCSS decreases.
- The "elbow" is the point where the rate of decrease shifts significantly, forming an angle like an elbow.
- This point represents the optimal balance between cluster compactness and the number of clusters.

Part 2: Practical Implementation

Question 6: Generate synthetic data using make_blobs, apply KMeans, and visualize results.

Answer:

Python
```
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans

# Generate data
X, y = make_blobs(n_samples=300, centers=4, random_state=42)

# Apply KMeans
kmeans = KMeans(n_clusters=4, random_state=42)
y_kmeans = kmeans.fit_predict(X)

# Visualize
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, alpha=0.75, marker='X')
plt.title("K-Means Clustering with 4 Centers")
plt.show()
```

Question 7: Load Wine dataset, apply StandardScaler, and train DBSCAN.

Answer:

Python
```
from sklearn.datasets import load_wine
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN

# Load and scale
wine = load_wine()
X_scaled = StandardScaler().fit_transform(wine.data)

# DBSCAN
dbscan = DBSCAN(eps=2.5, min_samples=5)
clusters = dbscan.fit_predict(X_scaled)

# Count clusters (exclude noise labeled as -1)
n_clusters = len(set(clusters)) - (1 if -1 in clusters else 0)
print(f"Number of clusters found (excluding noise): {n_clusters}")
```

Question 8: Generate make_moons data, apply DBSCAN, and highlight outliers.

Answer:

Python
```
from sklearn.datasets import make_moons

# Generate data
X, _ = make_moons(n_samples=200, noise=0.1, random_state=42)

# Apply DBSCAN
dbscan = DBSCAN(eps=0.2, min_samples=5)
labels = dbscan.fit_predict(X)

# Visualize
plt.scatter(X[labels != -1, 0], X[labels != -1, 1], c=labels[labels != -1], cmap='viridis',
label='Clusters')
plt.scatter(X[labels == -1, 0], X[labels == -1, 1], c='red', marker='x', label='Outliers')
plt.legend()
plt.title("DBSCAN: Moon Data with Outliers Highlighted")
plt.show()
```

Question 9: Wine dataset, PCA reduction, and Agglomerative Clustering.

Answer:

Python
```
from sklearn.decomposition import PCA
from sklearn.cluster import AgglomerativeClustering

# PCA to 2D
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled) # Using X_scaled from Q7

# Agglomerative Clustering
agg_clustering = AgglomerativeClustering(n_clusters=3)
labels = agg_clustering.fit_predict(X_pca)

# Visualize
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=labels, cmap='rainbow')
plt.title("Agglomerative Clustering on Wine Data (PCA Reduced)")
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
plt.show()
```

Part 3: Business Scenario

Question 10: E-commerce Customer Segmentation Workflow

+1
1. Algorithm Choice: I would use **K-Means** for its efficiency in handling customer transaction data and ease of interpretation for the marketing team. If the segments have irregular shapes or many outliers, I might also experiment with **DBSCAN**.

2. Preprocessing:

- **Missing Values:** Use mean/median imputation for numerical data or mode for categorical demographics.
- **Encoding:** Convert categorical data (like gender) into numerical format using One-Hot Encoding.
- **Scaling:** Apply StandardScaler to ensure features like "Total Spend" and "Age" are on the same scale.

3. Determining Clusters: I would use the **Elbow Method** to find the point where adding more clusters provides diminishing returns, cross-referenced with the **Silhouette Score** to ensure cluster distinctness.

4. Marketing Benefits:

- **Personalized Campaigns:** Offer discounts on electronics to "Tech Enthusiasts" while sending beauty product alerts to the "Fashion-Forward" segment.
- **Retention:** Identify "At-Risk" customers (low frequency, high past spend) for re-engagement rewards.

Would you like me to help you format these answers into a PDF document for your submission?