1 Multiple Choice Questions (MCQs)

1. **K-Means Initial Centroids**

   Scenario: A shopkeeper uses K-Means to cluster customers based on visit frequency and spend. The dataset has six customers: C1(2, 500), C2(10, 800), C3(4, 300), C4(11, 1200), C5(3, 350), C6(9, 1000). Initial centroids are set as C1(2, 500) and C4(11, 1200).

```python
import numpy as np
data = np.array([[2, 500], [10, 800], [4, 300], [11, 1200], [3, 350], [9, 1000]])
initial_centroids = np.array([[2, 500], [11, 1200]])
print(f"Initial centroids: {initial_centroids}")
```

   Output: `Initial centroids: [[2, 500], [11, 1200]]`
   Question: What is the purpose of setting initial centroids manually?

   A) To ensure reproducibility

   B) To increase clustering speed

   C) To reduce data dimensionality

   D) To eliminate outliers

   Answer: **A**
   Explanation: Manual centroid initialization ensures consistent results across runs, aiding reproducibility, especially for small datasets or specific scenarios. It doesn't directly speed up clustering, reduce dimensions, or remove outliers, which are handled by other techniques like PCA or preprocessing.

2. **Feature Scaling Necessity**

   Scenario: A retailer clusters customers using frequency (1–50 visits) and monetary spend (100–5000 units). Without scaling, what issue arises?

```python
from sklearn.preprocessing import StandardScaler
import numpy as np
data = np.array([[5, 2000], [15, 500], [3, 4500]])
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)
```

```python
print(f"Scaled data: {scaled_data.round(2)}")
```

Output: `Scaled data: [[-0.51 0.2] [ 1.4 -1.11] [-0.89 1.31]]`
Question: Why is feature scaling necessary?

A) To balance feature influence

B) To increase data variance

C) To remove negative values

D) To reduce dataset size

Answer: **A**
Explanation: Features like frequency and spend have different scales, causing spend to dominate distance calculations in K-Means. StandardScaler ($x' = \frac{x-\mu}{\sigma}$) ensures equal feature contribution, preventing bias. Scaling doesn't increase variance, remove negatives, or reduce size.

3. **Elbow Method Application**
Scenario: A retailer applies the elbow method to determine the optimal number of clusters for customer data. Inertia values for $k = 1$ to 5 are [1000, 600, 400, 350, 320].

```python
import matplotlib.pyplot as plt
k_range = range(1, 6)
inertia = [1000, 600, 400, 350, 320]
plt.plot(k_range, inertia, 'bo-')
plt.xlabel('Number of clusters (k)')
plt.ylabel('Inertia')
plt.title('Elbow Method')
plt.show()
```

Question: What is the likely optimal k?

A) 2

B) 3

C) 4

D) 5

Answer: **B**
Explanation: The elbow method identifies k where inertia's rate of decrease slows, forming an "elbow." From 1000 to 600 ($k = 2$) and 600 to 400 ($k = 3$) are sharp drops, but from 400 to 350 ($k = 4$) and 350 to 320 ($k = 5$) the decrease slows, suggesting $k = 3$ as optimal, balancing coherence and simplicity.

4. **Silhouette Score Interpretation (Moderate)**
Scenario: A shopkeeper evaluates K-Means clustering with silhouette scores for $k = 2$ to 4: [0.69, 0.62, 0.55].

```python
from sklearn.metrics import silhouette_score
from sklearn.cluster import KMeans
import numpy as np
data = np.array([[2, 500], [10, 800], [4, 300], [11, 1200], [3, 350], [9, 1000]])
for k in range(2, 5):
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = kmeans.fit_predict(data)
    score = silhouette_score(data, labels)
    print(f"k={k}, Silhouette Score: {score:.2f}")
```

Question: Which k indicates the best clustering?

A) 2
B) 3
C) 4
D) None, all are poor

Answer: **A**
Explanation: The silhouette score, $s = \frac{b-a}{\max(a,b)}$, measures cluster cohesion and separation, with higher values (near 1) indicating better-defined clusters. A score of 0.65 ($k = 2$) is higher than 0.45 ($k = 3$) and 0.29 ($k = 4$), suggesting $k = 2$ produces the most distinct clusters.

5. **Agglomerative Clustering Process**
Scenario: A retailer uses agglomerative clustering to group customers.

```python
from sklearn.cluster import AgglomerativeClustering
import numpy as np
data = np.array([[1, 2], [2, 3], [5, 6], [6, 6]])
agg = AgglomerativeClustering(n_clusters=2)
labels = agg.fit_predict(data)
print(f"Cluster labels: {labels}")
```

Question: What is the first step in agglomerative clustering?

A) Split all data into two clusters

B) Assign each point to its own cluster

C) Calculate final centroids

D) Merge all points into one cluster

Answer: **B**
Explanation: Agglomerative clustering is bottom-up, starting with each data point as its own cluster. It iteratively merges the closest pair of clusters based on a distance metric (e.g., Euclidean) until the desired number of clusters is reached. Splitting is divisive clustering, centroids are for K-Means, and merging all points is the end state.

6. **Divisive Clustering Approach**
Scenario: A retailer simulates divisive clustering on customer data.

```python
from sklearn.cluster import KMeans
import numpy as np
data = np.array([[1, 2], [2, 3], [5, 6], [6, 6]])
kmeans = KMeans(n_clusters=2, random_state=42,
   n_init=10)
labels = kmeans.fit_predict(data)
print(f"Initial split labels: {labels}")
```

Question: What characterizes divisive clustering?

A) Merging closest clusters

B) Starting with one cluster and splitting

C) Random centroid initialization

D) Fixed number of iterations

Answer: **B**
Explanation: Divisive clustering is top-down, starting with all points in one cluster and recursively splitting (e.g., using K-Means with $k = 2$) into smaller clusters until a stopping criterion is met. Merging is agglomerative, centroids are K-Means-specific, and iterations vary.

7. **PCA for Visualization**
Scenario: A retailer uses PCA to visualize customer clusters based on frequency, quantity, and monetary features.

```python
from sklearn.decomposition import PCA
import numpy as np
scaled_data = np.array([[0.1, 0.2, 0.3], [1.2, 0.8,
    0.9], [-0.5, -0.4, -0.3]])
pca = PCA(n_components=2)
principal_components = pca.fit_transform(scaled_data)
print(f"PCA components:
    {principal_components.round(2)}")
```

Question: Why is PCA used here?

A) To scale features

B) To reduce dimensionality for plotting

C) To compute cluster centroids

D) To calculate silhouette scores

Answer: **B**
Explanation: PCA reduces high-dimensional data (e.g., three features) to two principal components, capturing most variance for 2D visualization of clusters. Scaling is done by StandardScaler, centroids by K-Means, and silhouette scores measure cluster quality, not visualization.
Hint: Consider why 3D data needs simplification for scatter plots.

8. **Inertia in K-Means**
Scenario: A retailer runs K-Means with $k = 3$, and the inertia is 450.

```python
from sklearn.cluster import KMeans
import numpy as np
data = np.array([[2, 500], [10, 800], [4, 300], [11, 1200], [3, 350], [9, 1000]])
kmeans = KMeans(n_clusters=3, random_state=42, n_init=10)
kmeans.fit(data)
print(f"Inertia: {kmeans.inertia_:.0f}")
```

Question: What does inertia represent?

A) Number of clusters

B) Sum of squared distances to centroids

C) Average silhouette score

D) Total data variance

Answer: **B**
Explanation: Inertia is the sum of squared Euclidean distances from each point to its assigned centroid, measuring cluster cohesion. Lower inertia indicates tighter clusters. It's not the number of clusters, silhouette score, or total variance, which are distinct metrics.

2 Numeric Type Questions (NTQs)

1. **Euclidean Distance Calculation**
Scenario: In K-Means, a customer at (5, 600) is compared to a centroid at (10, 800).
Question: What is the Euclidean distance, rounded to two decimal places?
Answer: **205.91**
Explanation: Euclidean distance is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. For (5, 600) and (10, 800): $\sqrt{(10-5)^2 + (800-600)^2} = \sqrt{25 + 40000} = \sqrt{40025} \approx$ 205.91. This distance determines cluster assignment.

2. **Silhouette Score for** $k = 2$
Scenario: A retailer runs K-Means with $k = 2$ on scaled data, yielding a silhouette score of 0.6923.

Question: What is the silhouette score, rounded to two decimal places?
Answer: **0.69**
Explanation: The silhouette score, $s = \frac{b-a}{\max(a,b)}$, is given as 0.6923, rounded to 0.69. This high score indicates well-separated clusters, as scores near 1 suggest strong cohesion and separation.
Hint: Round the provided score to two decimal places.

3. **Inertia for $k = 3$**
Scenario: A K-Means model with $k = 3$ on customer data has an inertia of 450.23.
Question: What is the inertia, rounded to the nearest integer?
Answer: **450**
Explanation: Inertia, the sum of squared distances to centroids, is given as 450.23, rounded to 450. This measures how tightly clustered the points are around their centroids.
Hint: Round the inertia value to the nearest integer.

4. **PCA Variance Explained**
Scenario: A retailer applies PCA to three scaled features, reducing to two components with explained variance ratios [0.65, 0.25].
Question: What is the total variance explained by the two components, as a percentage rounded to the nearest integer?
Answer: **90**
Explanation: The explained variance ratios [0.65, 0.25] sum to $0.65 + 0.25 = 0.90$, or 90%. This indicates the two components capture 90% of the data's variance, useful for visualization.

3 Multiple Select Questions (MSQs)

1. **K-Means Characteristics**
Scenario: A retailer uses K-Means to segment customers.
Question: Which statements are true about K-Means?

A) Uses Euclidean distance for assignments

B) Requires feature scaling

C) Guarantees globally optimal clusters

D) Iterates until centroids stabilize

Answers: **A, B, D**
Explanation:
- A) K-Means assigns points to the nearest centroid using Euclidean distance, true.
- B) Feature scaling ensures equal feature influence, true.
- C) K-Means may converge to local optima, not always global, false.
- D) It iterates until centroids no longer change, true.

2. **Clustering Evaluation Metrics**
Scenario: A shopkeeper evaluates clustering quality.
Question: Which metrics help determine the optimal number of clusters?

A) Elbow method

B) Silhouette score

C) Euclidean distance

D) PCA variance

Answers: **A, B**
Explanation:
- A) The elbow method plots inertia vs. k, identifying an optimal k, true.
- B) Silhouette score measures cluster cohesion and separation, true.
- C) Euclidean distance is for assignments, not choosing k, false.
- D) PCA variance aids visualization, not cluster number selection, false.

3. **Hierarchical Clustering Features**
Scenario: A retailer compares agglomerative and divisive clustering.
Question: Which statements are true about hierarchical clustering?

A) Agglomerative starts with each point as a cluster

B) Divisive starts with all points in one cluster

C) Both require a fixed number of clusters upfront

D) Agglomerative is more commonly used

Answers: **A, B, D**
Explanation:
- A) Agglomerative begins with each point as a cluster, merging iteratively, true.
- B) Divisive starts with one cluster, splitting recursively, true.
- C) Hierarchical methods produce a dendrogram, allowing flexible cluster numbers, false.
- D) Agglomerative is more common due to simpler implementation, true.