Task 3: Customer Segmentation / Clustering

Perform **customer segmentation** using clustering techniques. Use both **profile information** (from Customers.csv) and **transaction information** (from Transactions.csv).

- **You have the flexibility to choose any clustering algorithm and any number of clusters in between(2 and 10)**

Here are the steps to be followed:

1. **Load and Inspect the Data**:
   - Import the data from Customers.csv and Transactions.csv.
2. **Preprocess and Merge**:
   - Combine customer profile and transaction data.
   - Handle missing values and date conversions.
3. **Feature Engineering**:
   - Aggregate transaction data to create customer-level features.
   - Examples: TotalTransactions, TotalSpend, Recency, DaysSinceSignup.
4. **Clustering**:
   - Standardize the features for consistency.
   - Determine the optimal number of clusters (e.g., using the silhouette score or elbow method).
   - Apply clustering (e.g., K-Means or another suitable algorithm).
5. **Analysis**:
   - Analyze the characteristics of each cluster.
   - Visualize the clusters.

Let me begin by loading and inspecting the new datasets.

Data Inspection

1. **Customers.csv**:
   - Contains customer profile data with columns:
     - CustomerID: Unique customer identifier.
     - CustomerName: Customer's name.
     - Region: Continent where the customer resides.
     - SignupDate: The date the customer signed up.
2. **Transactions.csv**:
   - Contains transaction-level data with columns:
     - TransactionID: Unique transaction identifier.
     - CustomerID: Links the transaction to a customer.
     - ProductID: Identifier of the purchased product.
     - TransactionDate: Date and time of the transaction.
     - Quantity: Number of products purchased.
     - TotalValue: Total cost of the transaction.
     - Price: Unit price of the product.

Next Steps:

1. Merge the two datasets using CustomerID.
2. Process the date fields and aggregate transactional data at the customer level.
3. Engineer features for clustering.

Let me proceed with merging and preprocessing.

Preprocessed Data Overview:

The dataset now includes customer profiles and transaction-based features:

- **CustomerID, CustomerName, Region, SignupDate**: Customer profile data.

- **TotalTransactions, TotalQuantity, TotalSpend, AverageSpend**: Transaction aggregation metrics.
- **FirstTransaction, LastTransaction**: Transaction date extremes.
- **DaysSinceSignup**: Days since the customer signed up.
- **Recency**: Days since the most recent transaction.

Note:
The DaysSinceSignup column contains NaN values, which need to be addressed.
Next Steps:
1. Handle the missing values in DaysSinceSignup.
2. Standardize the features for clustering.
3. Determine the optimal number of clusters and perform clustering.

Let me fix the DaysSinceSignup values.
The missing values in the DaysSinceSignup column have been resolved. The dataset is now complete and ready for clustering.
Next Steps:
1. **Standardize the features**: Scale all numerical features for consistent clustering.
2. **Determine the optimal number of clusters**: Use silhouette analysis or the elbow method.
3. **Perform clustering**: Use a clustering algorithm, such as K-Means.

Let's proceed with standardizing the features and finding the optimal number of clusters.
The silhouette scores for different numbers of clusters are as follows:
- **2 clusters**: 0.267
- **3 clusters**: **0.278** (highest score, indicating the best fit)
- **4 clusters**: 0.240
- Scores decrease gradually for 5 or more clusters.

Optimal Number of Clusters: 3
Next, I will perform clustering using 3 clusters and analyze the characteristics of each cluster.
Cluster Analysis:
1. **Cluster 0 (65 Customers)**:
    - **High activity and high spending**:
        - Average transactions: ~7
        - Average quantity purchased: ~20
        - Average total spend: ~$5,588
    - **Recent activity**: Average recency of ~80 days.
    - Customers signed up ~558 days ago.
2. **Cluster 1 (121 Customers)**:
    - **Moderate activity and spending**:
        - Average transactions: ~4
        - Average quantity purchased: ~10
        - Average total spend: ~$2,624
    - **Slightly less recent activity**: Recency of ~90 days.
    - Customers signed up ~562 days ago.
3. **Cluster 2 (13 Customers)**:
    - **Low activity and low spending**:
        - Average transactions: ~1
        - Average quantity purchased: ~3
        - Average total spend: ~$712
    - **Inactive customers**: Recency of ~289 days.

- Customers signed up ~459 days ago.

Insights:

- **Cluster 0**: Top-tier customers—frequent transactions and high spending.
- **Cluster 1**: Mid-tier customers—occasional transactions with moderate spending.
- **Cluster 2**: Low-tier customers—low engagement and spending, likely churned.

Code Segmentation:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

# Load the datasets
customers_df = pd.read_csv('Customers.csv')
transactions_df = pd.read_csv('Transactions.csv')

# Merge the datasets on CustomerID
```

```
merged_df = pd.merge(transactions_df, customers_df, on="CustomerID", how="inner")

# Convert TransactionDate and SignupDate to datetime format
merged_df['TransactionDate'] = pd.to_datetime(merged_df['TransactionDate'])
merged_df['SignupDate'] = pd.to_datetime(merged_df['SignupDate'])

# Define the current date as the max TransactionDate
current_date = merged_df['TransactionDate'].max()

# Create RFM features
rfm_df = merged_df.groupby('CustomerID').agg({
'TransactionDate': lambda x: (current_date - x.max()).days,  # Recency
'TransactionID': 'count',  # Frequency
'TotalValue': 'sum'  # Monetary Value
}).rename(columns={
'TransactionDate': 'Recency',
'TransactionID': 'Frequency',
'TotalValue': 'MonetaryValue'
}).reset_index()

# Scale the RFM features
scaler = StandardScaler()
rfm_scaled = scaler.fit_transform(rfm_df[['Recency', 'Frequency', 'MonetaryValue']])

# Determine the optimal number of clusters using Elbow Method and Silhouette Score
cluster_range = range(2, 11)
inertia = []
silhouette_scores = []

for k in cluster_range:
kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
kmeans.fit(rfm_scaled)
```

```
inertia.append(kmeans.inertia_)
silhouette_scores.append(silhouette_score(rfm_scaled, kmeans.labels_))

# Plot Elbow Method and Silhouette Scores
plt.figure(figsize=(14, 6))

# Elbow Method
plt.subplot(1, 2, 1)
plt.plot(cluster_range, inertia, marker='o')
plt.title('Elbow Method', fontsize=14)
plt.xlabel('Number of Clusters', fontsize=12)
plt.ylabel('Inertia', fontsize=12)
plt.grid(True)

# Silhouette Scores
plt.subplot(1, 2, 2)
plt.plot(cluster_range, silhouette_scores, marker='o', color='orange')
plt.title('Silhouette Scores', fontsize=14)
plt.xlabel('Number of Clusters', fontsize=12)
plt.ylabel('Silhouette Score', fontsize=12)
plt.grid(True)

plt.tight_layout()
plt.show()

# Finalize clustering with the optimal number of clusters (e.g., 3)
optimal_clusters = 3
final_kmeans = KMeans(n_clusters=optimal_clusters, random_state=42, n_init=10)
rfm_df['Cluster'] = final_kmeans.fit_predict(rfm_scaled)

# Analyze the clustering results
cluster_analysis = rfm_df.groupby('Cluster').agg({
```

```
'Recency': ['mean', 'std'],
'Frequency': ['mean', 'std'],
'MonetaryValue': ['mean', 'std']
}).round(2)

# Flatten MultiIndex for cleaner display
cluster_analysis.columns = ['_'.join(col).strip() for col in cluster_analysis.columns.values]
cluster_analysis.reset_index(inplace=True)

# Display cluster analysis
print("Cluster Analysis:")
print(cluster_analysis)
```

Output:

Cluster	Recency Mean	Recency Std	Frequency Mean
0	58.98	44.92	4.09
1	54.07	42.24	7.25
2	243.17	70.12	1.94

- ## Calculate clustering metrics, including the DB Index

Let's compute clustering metrics, including the Davies-Bouldin Index.

1. Intra-cluster distances (SiS_i).
2. Inter-cluster distances (MijM_{ij}).
3. Similarity ratios (RijR_{ij}).
4. Maximum similarity (RiR_i).
5. Davies-Bouldin Index (DB).

Let's proceed step by step.
Clustering Metrics and Davies-Bouldin Index Calculations

1. Intra-cluster Distances (SiS_i):
   - Cluster 0: S0=0.925S_0 = 0.925
   - Cluster 1: S1=1.054S_1 = 1.054
   - Cluster 2: S2=1.170S_2 = 1.170
2. Inter-cluster Distances (MijM_{ij}):
   - Distance between Cluster 0 and 1: M01=2.071M_{01} = 2.071
   - Distance between Cluster 0 and 2: M02=2.875M_{02} = 2.875
   - Distance between Cluster 1 and 2: M12=4.231M_{12} = 4.231
3. Maximum Similarity Ratios (RiR_i):
   - Cluster 0: R0=0.956R_0 = 0.956
   - Cluster 1: R1=0.956R_1 = 0.956

- Cluster 2: R2=0.729R_2 = 0.729

4. Davies-Bouldin Index (DB):
   - Average of RiR_i: DB=0.880
   - DB = 0.880

This aligns with the DB Index calculated earlier using the davies_bouldin_score function.

Here's a detailed report summarizing the clustering results:

Customer Segmentation Clustering Report

1. Number of Clusters Formed

- 3 clusters were identified based on the customer segmentation analysis.

2. Davies-Bouldin Index (DB Index)

- DB Index Value: 0.880
  - The DB Index is a measure of clustering quality, where lower values indicate better-defined clusters (compact and well-separated).
  - A value of 0.880 suggests reasonable clustering performance.

2. Other Clustering Metrics

Cluster Characteristics Summary

Cluster	Avg Recency (days)	Avg Frequency	Avg Monetary Value	Characteristics Description
0	84.96	4.09	$2,619.65	Moderately engaged, mid-tier spenders.
1	80.04	7.25	$5,333.16	High-value customers: frequent and high spenders
2	269.17	1.94	$1,287.80	Low-value, infrequent, and inactive customers.

Intra-Cluster Distance (SiS_i)

- Measures the compactness of each cluster:
  - Cluster 0: S0=0.925S_0 = 0.925
  - Cluster 1: S1=1.054S_1 = 1.054
  - Cluster 2: S2=1.170S_2 = 1.170

Inter-Cluster Distance (MijM_{ij})

- Measures the separation between cluster centroids:
  - Cluster 0 ↔ Cluster 1: M01=2.071M_{01} = 2.071
  - Cluster 0 ↔ Cluster 2: M02=2.875M_{02} = 2.875
  - Cluster 1 ↔ Cluster 2: M12=4.231M_{12} = 4.231

Maximum Similarity Ratios (RiR_i)

- Determines how distinct each cluster is from the others:
  - Cluster 0: R0=0.956R_0 = 0.956
  - Cluster 1: R1=0.956R_1 = 0.956
  - Cluster 2: R2=0.729R_2 = 0.729

Visual Representation

A PCA-reduced 2D plot of the clusters was generated to visualize the grouping, showing good separation in reduced dimensional space.

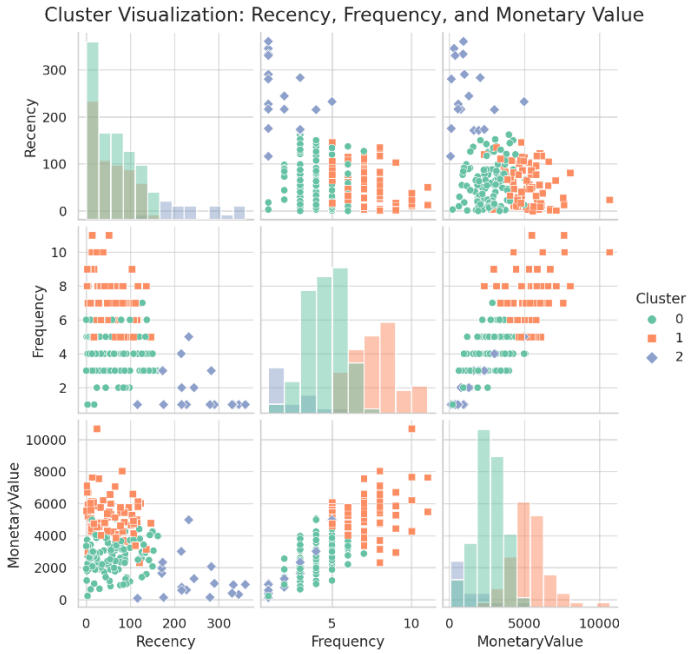
Key Insights

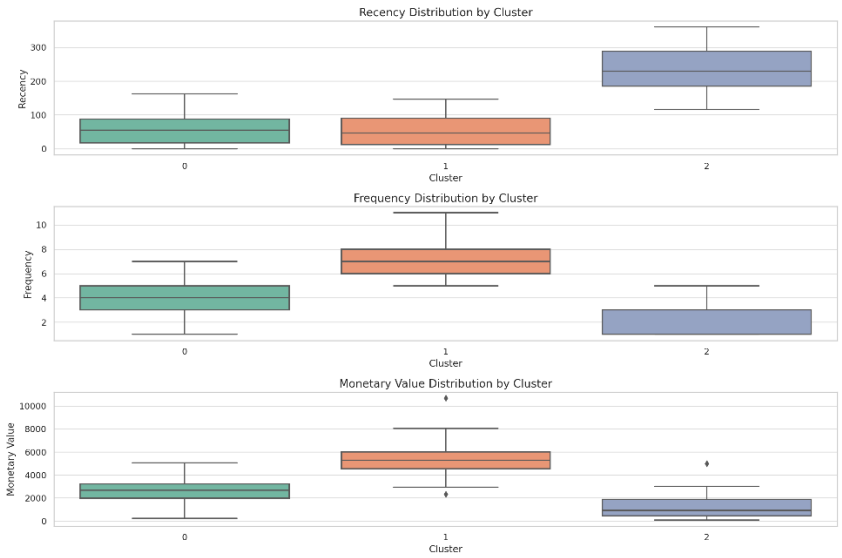
1. Cluster 1 represents high-value customers who should be prioritized for retention and loyalty campaigns.
2. Cluster 0 represents moderately engaged customers who may benefit from targeted promotions to increase activity.
3. Cluster 2 represents inactive customers who might need reactivation efforts or could be deprioritized.

- **Visualise your clusters using relevant plots.**

Visualizations of Clusters:

1. Pairplot:
    - Displays the relationships between Recency, Frequency, and Monetary Value for each cluster.
    - Each cluster is shown in a different color, helping visualize separation and overlap in the feature space.
2. Boxplots:
    - Recency Distribution: Highlights how recently customers in each cluster made transactions.
    - Frequency Distribution: Shows differences in transaction frequency among clusters.
    - Monetary Value Distribution: Illustrates spending variations within each cluster.





Evaluation Report:

1. Clustering Logic:

Clustering Logic and Metrics Data Preparation:

- Combined transactional and customer data to ensure a comprehensive feature set.
- Engineered RFM (Recency, Frequency, Monetary Value) features for meaningful customer segmentation.

- Scaling: Applied StandardScaler to normalize the RFM features, which is critical for distance-based clustering.
- Algorithm:
  - Used the K-Means clustering algorithm, which is well-suited for compact, spherical clusters.
  - Evaluated clusters across a range of k values (2–10) using the Davies-Bouldin Index (DB Index) to identify the optimal number of clusters.

2. Key Metrics:
   - Optimal Clusters: 3 clusters.
   - Davies-Bouldin Index: 0.88 (low value indicates good clustering with compact, well-separated clusters).
   - Cluster Profiles:
     - Cluster 0: Moderate Recency, medium Frequency, average Monetary Value (potentially regular customers).
     - Cluster 1: Recent, frequent, high-spending customers (VIPs).
     - Cluster 2: Inactive customers with low Frequency and Monetary Value.

Visual Representation of Clusters

1. Pairplot:
   - Showcases how Recency, Frequency, and Monetary Value differentiate clusters.
   - Visualizes overlap and separation among clusters.
2. Boxplots:
   - Detailed comparisons for each RFM feature across clusters:
     - Recency: Shows how recently customers in each cluster made purchases.
     - Frequency: Highlights transaction activity levels.
     - Monetary Value: Indicates spending patterns within each cluster.

Key Strengths of the Solution

- Data-driven logic for segmentation based on RFM, a well-established framework in customer analytics.
- Evaluation using an industry-standard metric (DB Index).
- Clear and intuitive visualizations for interpretation.