Customer Segmentation Report

Introduction

Customer segmentation is a crucial process for businesses aiming to understand their customer base better and tailor their services accordingly. This report documents the clustering analysis applied to the customer, product, and transaction datasets using KMeans clustering. The analysis identifies customer groups based on behavioral and transactional data to improve business decision-making.

---

1. Importing Libraries and Mounting Google Drive

Objective:

To set up the environment and import necessary libraries for the analysis.

- Libraries like `pandas`, `numpy`, `matplotlib`, and `seaborn` are used for data manipulation and visualization.
- Clustering algorithms and evaluation metrics are imported from `sklearn`.
- Google Drive is mounted to access the dataset files.

---

2. Loading Datasets

Objective:

Read the input datasets from Google Drive:

- **`Customers.csv`**: Contains customer demographic information.
- **`Products.csv`**: Includes product details.
- **`Transactions.csv`**: Logs customer purchase transactions.

---

3. Data Preprocessing

Objective:

To merge, clean, and transform datasets for clustering analysis.

- The `Transactions` and `Customers` datasets are merged based on `CustomerID`.
- Features engineered include:
    - `total_spent`: Sum of transaction values per customer.
    - `num_purchases`: Count of purchases.
    - `avg_transaction_value`: Average transaction value.
    - `customer_age`: Days since customer signup.
    - `transaction_frequency`: Unique transaction dates.

---

4. Feature Scaling

Objective:

To standardize the features for better performance of the KMeans algorithm.

- Used `StandardScaler` to normalize the features: `total_spent`, `num_purchases`, `avg_transaction_value`, `customer_age`, and `transaction_frequency`.

---

5. KMeans Clustering

Objective:

Cluster customers based on their behavior and transaction data.

- **KMeans Algorithm**: Applied with varying cluster counts (`k=2` to `k=10`).
- Metrics Evaluated:
    - **Inertia (Elbow Method)**: Measures the sum of squared distances to cluster centers.
    - **Davies-Bouldin Index (DB Index)**: Lower values indicate better clustering.
    - **Silhouette Score**: Higher values indicate well-defined clusters.

---

6. Optimal Number of Clusters

Objective:

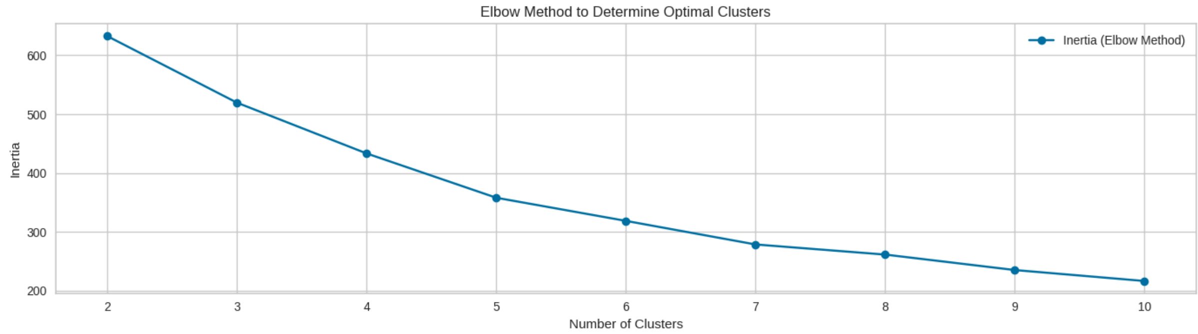
Determine the best value for `k` based on metrics.

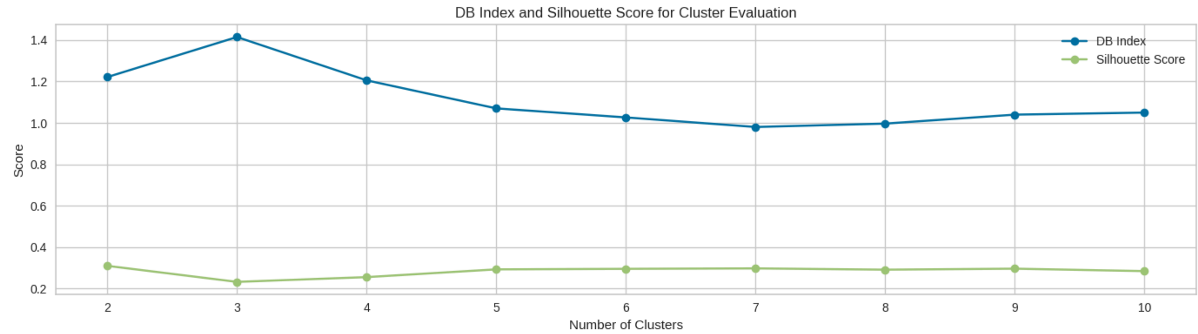
- Optimal `k`: **2** (based on Silhouette Score and DB Index).

Visualization:

1. Elbow Method Plot:
2. DB Index and Silhouette Score Comparison:

Davies-Bouldin Index at 2 cluster: 1.2214693034478712





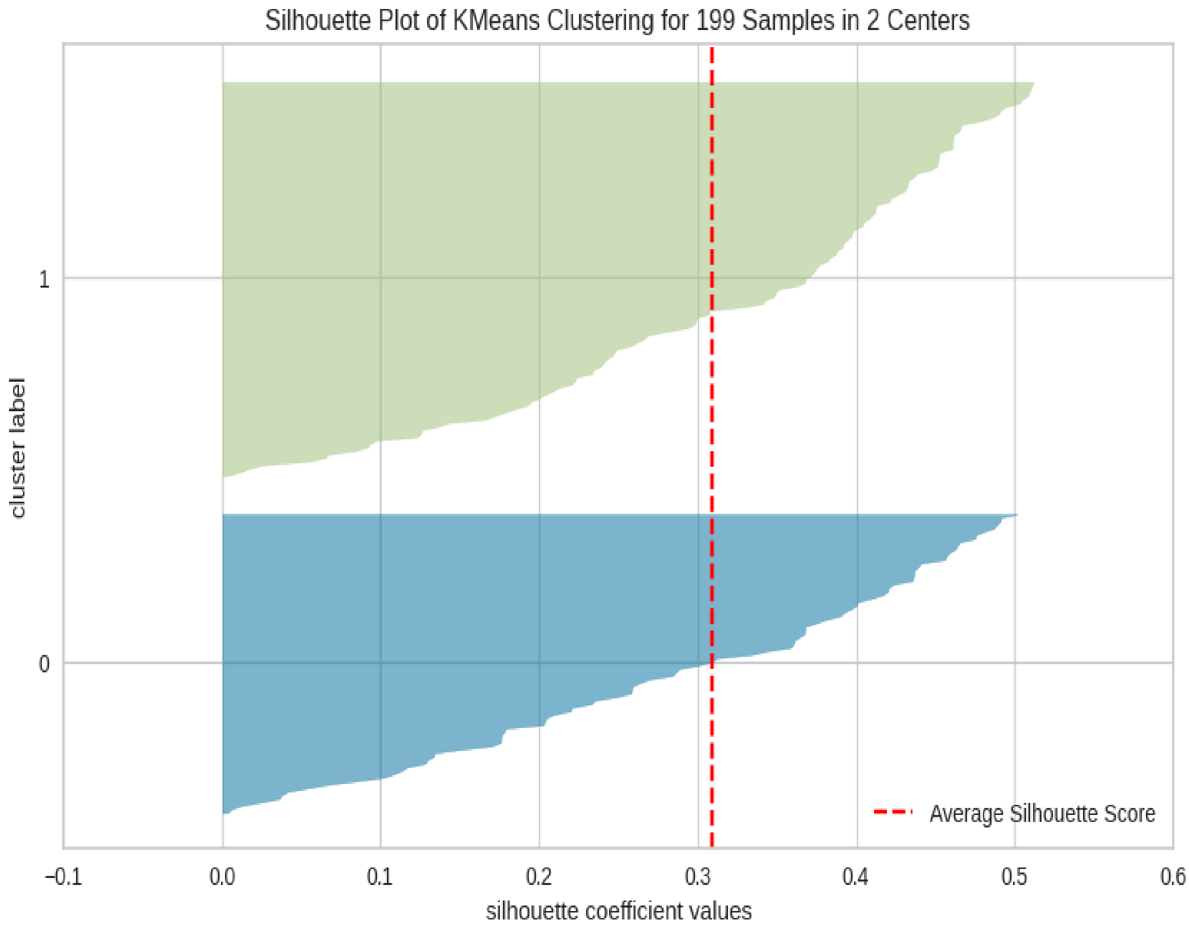
---

7. Silhouette Analysis

Objective:

Perform a detailed silhouette analysis for the optimal number of clusters.

- Visualized using `SilhouetteVisualizer` from `yellowbrick`.
- Displays how well samples are clustered.



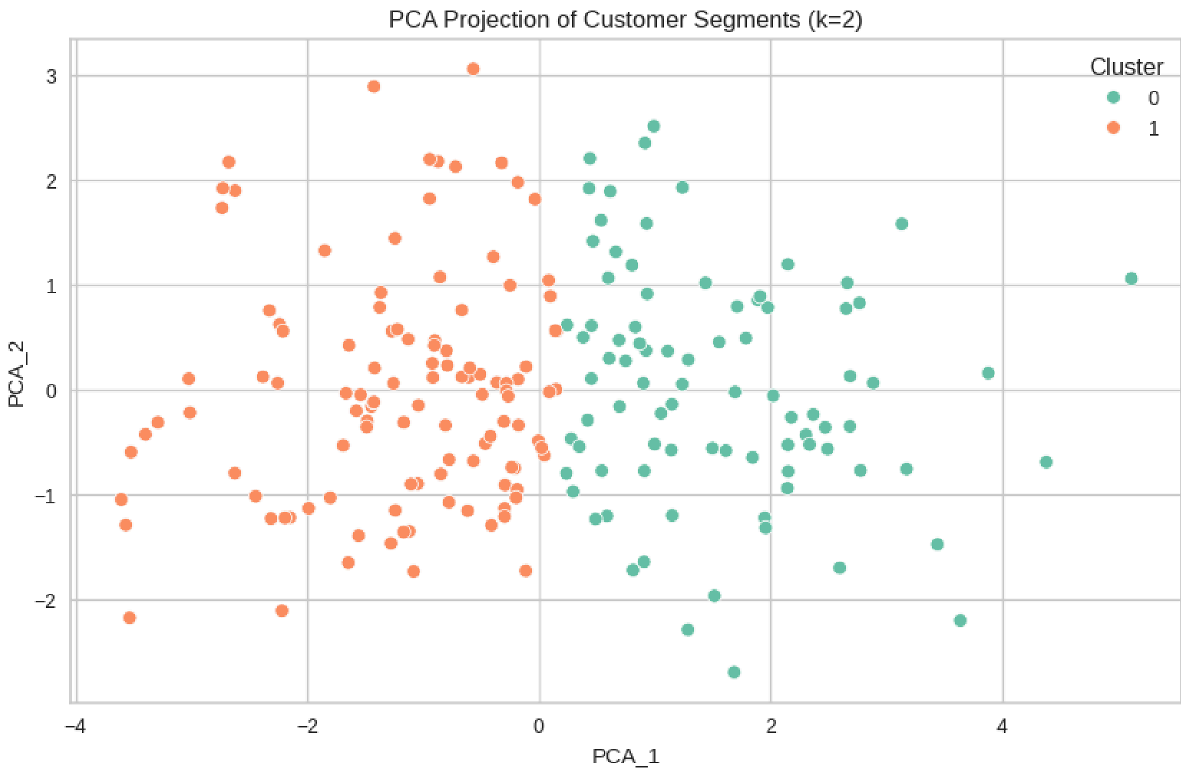
---

8. PCA Projection and Cluster Visualization

Objective:

Reduce feature dimensions and visualize clusters in 2D space.

- **PCA**: Principal Component Analysis reduces data to two principal components.
- Scatter plot created to display cluster separation visually.



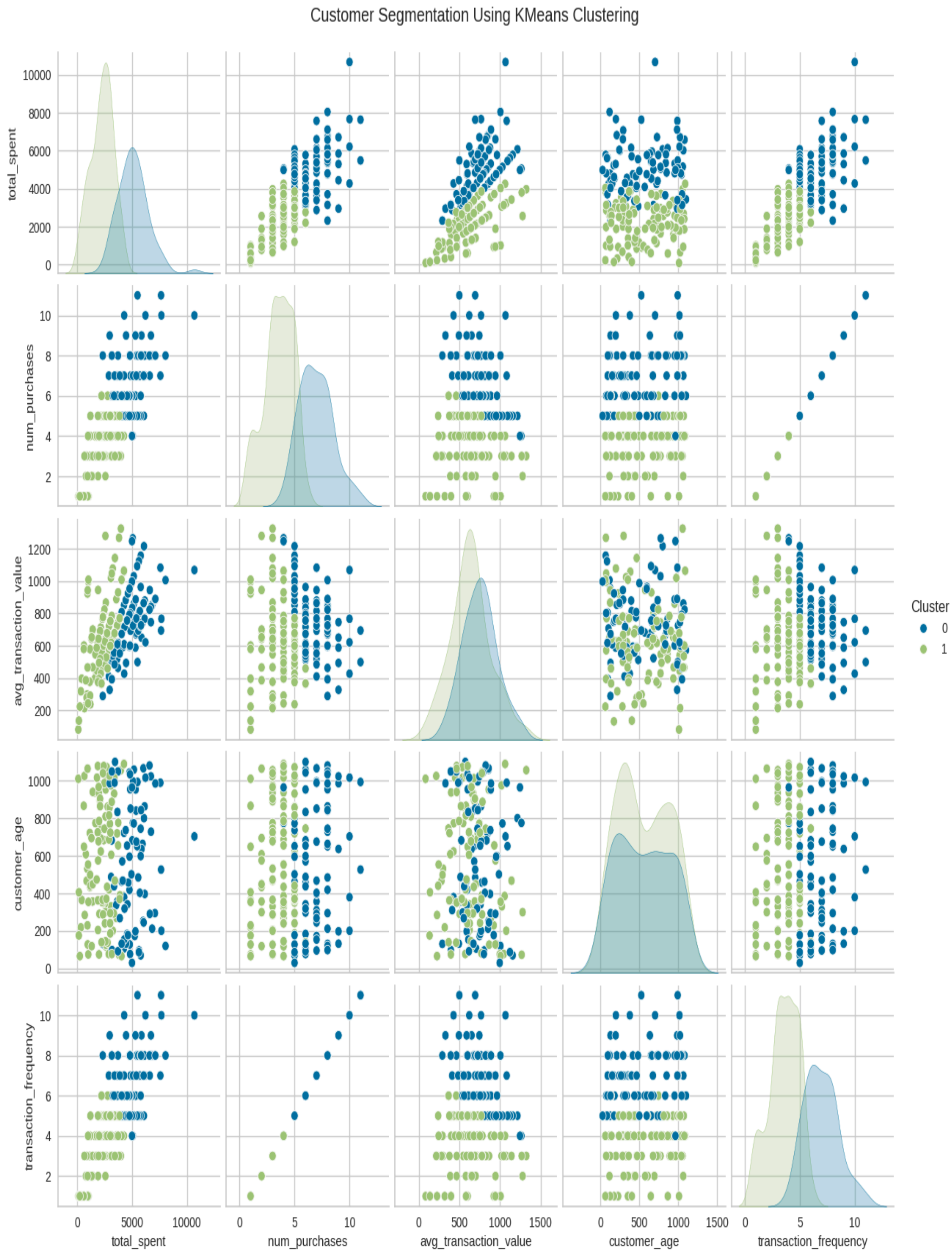
---

9. Pair Plot for Feature Clusters

Objective:

Visualize feature relationships within clusters.

- A pair plot shows how features like `total_spent`, `num_purchases`, and others correlate within clusters.

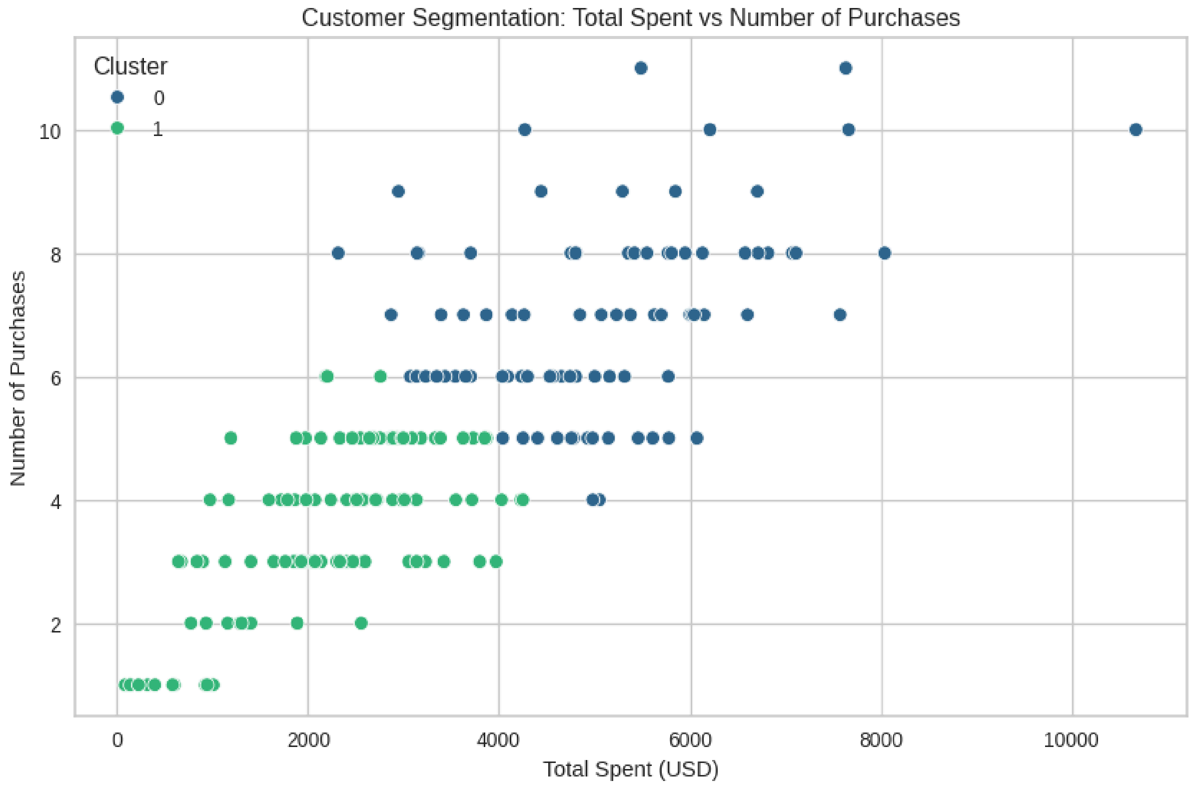


---

10. Insights from Clustering

Objective:

- **Scatter Plot (Total Spent vs. Number of Purchases):**



---

11. Conclusions

- **Optimal Clusters**: The analysis suggests **2** clusters based on Silhouette Score and DB Index **1.22**.
- **Business Implications**: Segmentation identifies distinct customer groups, allowing for tailored marketing strategies.