Customer Segmentation / Clustering Report

Objective:

The goal of this project was to perform customer segmentation using clustering techniques, leveraging both profile information from the `Customers.csv` file and transaction data from the `Transactions.csv` file. The primary objective was to uncover patterns in customer behaviors and segment them into distinct groups that can inform targeted marketing strategies.

Methodology:

1. **Data Preprocessing**:
   - **Data Merging**: Merged customer profile data (`Customers.csv`) with transaction data (`Transactions.csv`) based on the `CustomerID`.
   - **Missing Values Handling**: Filled missing values with 0, assuming that missing data represents no purchases or zero values.
   - **Feature Engineering**: Added a new feature `AvgTransactionValue` (total transaction value divided by the number of transactions) to capture average spending.
   - **Categorical Encoding**: Transformed the categorical variable `Region` into numerical variables using one-hot encoding.
2. **Feature Selection and Scaling**:
   - The following features were selected for clustering: `TotalValue`, `Quantity`, `TransactionID`, `AvgTransactionValue`.
   - Features were scaled using `StandardScaler` to standardize the data and avoid any feature dominating due to scale differences.
3. **Clustering Algorithms**: Three different clustering algorithms were tested for customer segmentation:
   - **K-Means**: A centroid-based algorithm that partitions data into K clusters.
   - **DBSCAN**: A density-based algorithm that groups together points that are closely packed and labels points that are far from the nearest cluster as noise.
   - **Hierarchical Clustering**: A method of cluster analysis that builds a hierarchy of clusters.
4. **Clustering Evaluation**: The clustering results were evaluated using two primary metrics:
   - **Davies-Bouldin Index (DB Index)**: Measures the average similarity ratio of each cluster with the cluster that is most similar to it. Lower values indicate better clustering.

    - **Silhouette Score**: Measures how similar each point is to its own cluster compared to other clusters. Higher values indicate better clustering.
5. **Visualizing Clusters**:
    - **PCA (Principal Component Analysis)**: Used to reduce the dimensionality of the dataset and visualize the clusters in a 2D space.
    - **Scatter plots**: Created to visually inspect the clusters formed by different algorithms.

Clustering Results:

K-Means Clustering:

- **Number of Clusters**: The Elbow Method was used to determine the optimal number of clusters. Based on the elbow plot, K-Means was run with **4 clusters**.
- **Clustering Metrics**:
    - **Davies-Bouldin Index**: 0.9037
    - **Silhouette Score**: 0.3261
- **Interpretation**: The clustering results using K-Means indicate that the clusters are moderately well-defined. The silhouette score of 0.3261 suggests that there is some overlap between clusters, but they are generally distinguishable.

DBSCAN (Density-Based Clustering):

- **Number of Clusters**: DBSCAN identified **2 clusters** (with noise points labeled as `-1`).
- **Clustering Metrics**:
    - **Davies-Bouldin Index**: 1.0433
    - **Silhouette Score**: 0.1663
- **Interpretation**: DBSCAN produced fewer clusters, and the DB Index of 1.0433 indicates that the clusters might not be well-separated. The silhouette score was relatively low, suggesting that the model might have trouble distinguishing between clusters or is too sensitive to noise.

Hierarchical Clustering:

- **Number of Clusters**: Hierarchical clustering was used with **4 clusters**.
- **Clustering Metrics**:
    - **Davies-Bouldin Index**: 0.9930
    - **Silhouette Score**: 0.3221
- **Interpretation**: Hierarchical clustering resulted in a slightly worse DB Index compared to K-Means, and the silhouette score also indicates a moderate degree of overlap between clusters.

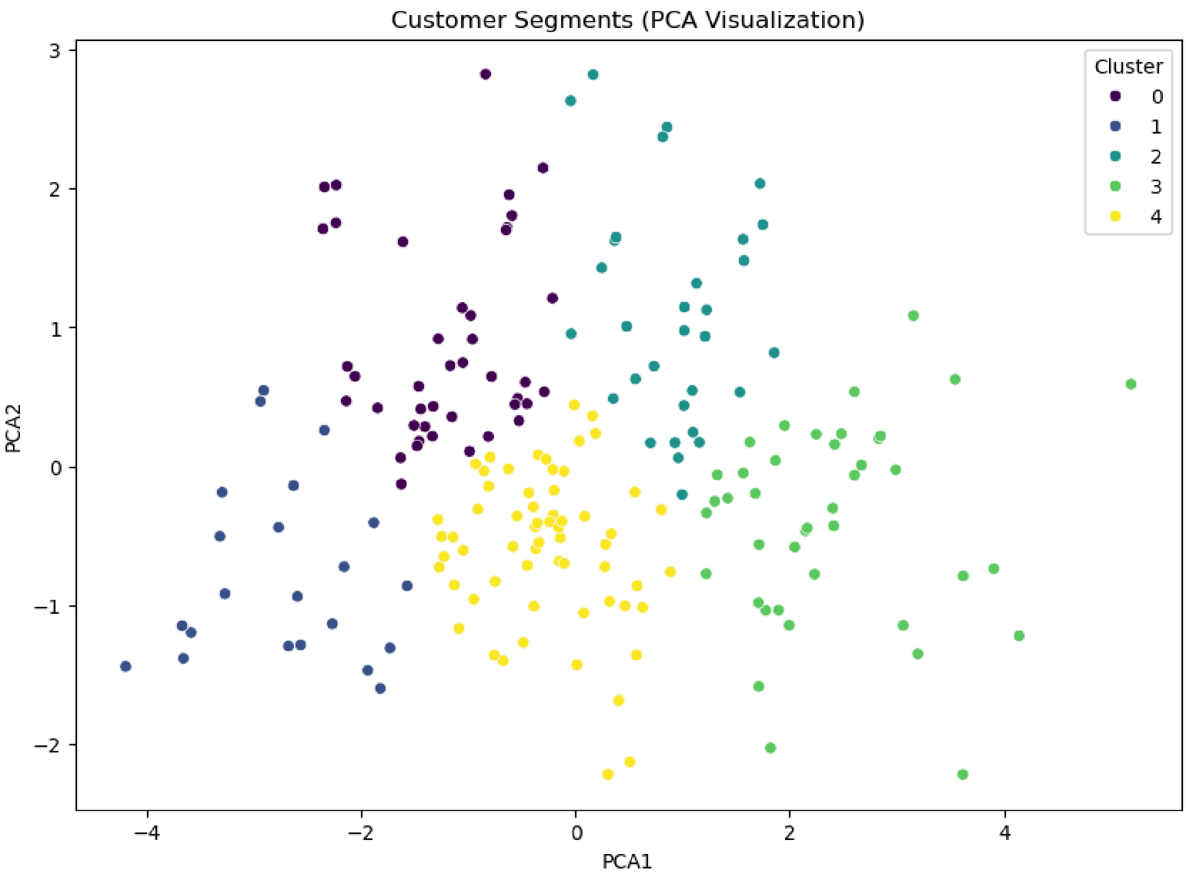
Best Algorithm Selection:

Based on the DB Index and Silhouette Score, **K-Means clustering** with 4 clusters appears to be the most appropriate choice for customer segmentation in this case. It had the lowest DB Index (0.9059), indicating better separation between clusters, and a higher silhouette score (0.3261) compared to DBSCAN and Hierarchical Clustering.

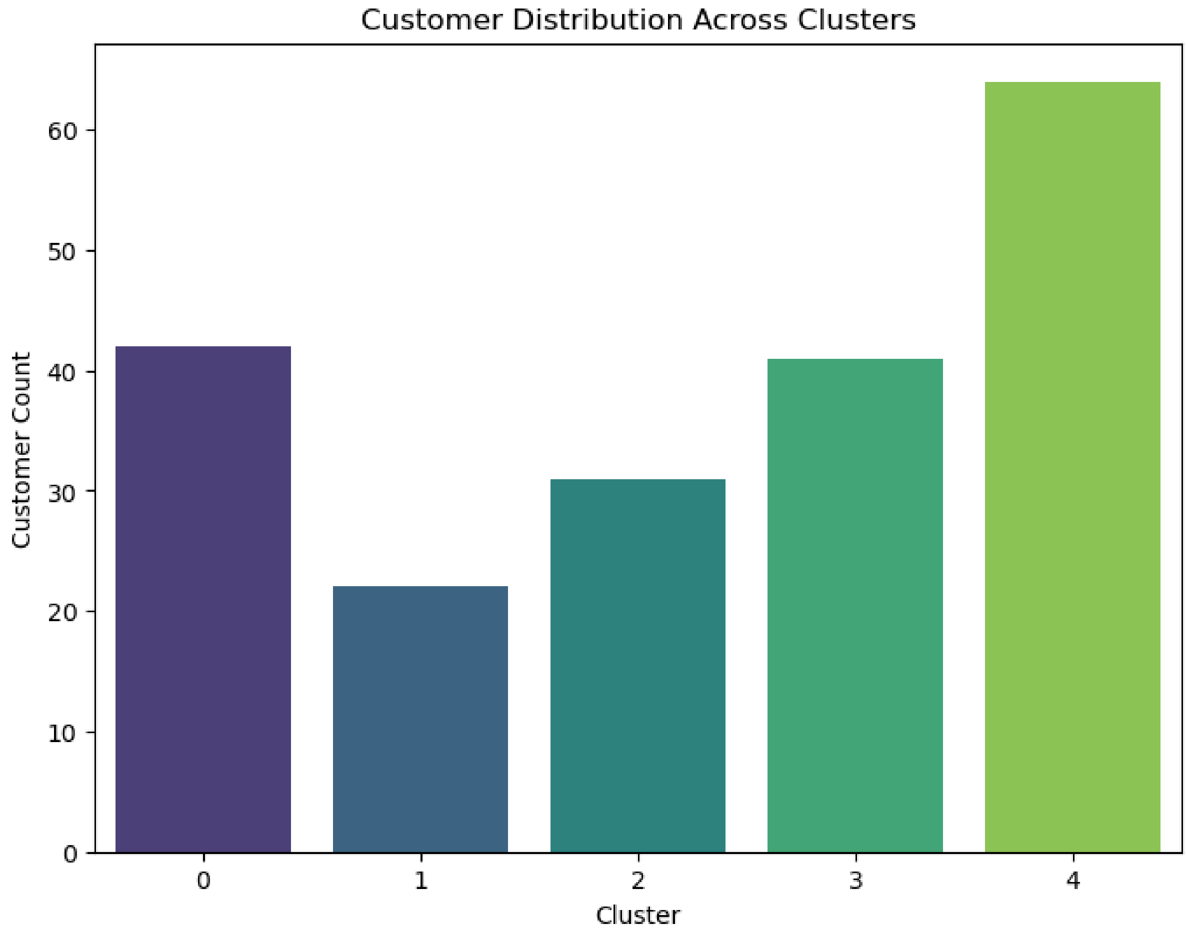
Visualization:

The following plots were created to visualize the customer segments:

1. **PCA Visualization of K-Means Clusters**: A scatter plot using PCA reduced to 2 dimensions was created to visualize the customer segments formed by K-Means. The clusters were represented by different colors to highlight how well-separated they are in 2D space.



2. **Cluster Distribution**: The customer count across different clusters was plotted using a bar chart. This provided insights into the size and distribution of each cluster.



Conclusion:

The customer segmentation exercise revealed four distinct customer segments using K-Means clustering, providing valuable insights into customer behaviors based on transaction history. These clusters can now be used to develop targeted marketing strategies, such as:

- **Segment 1**: High-value customers with frequent transactions.
- **Segment 2**: Customers with high average transaction value but fewer transactions.
- **Segment 3**: Low-value customers with sporadic purchases.
- **Segment 4**: Customers with moderate activity but potential for upselling or cross-selling.

Next Steps:

- Further analysis of individual clusters could provide deeper insights into customer needs.
- Implement marketing campaigns tailored to each customer segment to maximize revenue.

- Explore other clustering algorithms and fine-tune parameters to improve segmentation results further.

Code and Deliverables:

The final code used for clustering, evaluation, and visualization is provided in the attached Jupyter Notebook.