Customer Segmentation Clustering Report

Overview:

The goal of this task was to perform customer segmentation using clustering techniques, leveraging both profile information (from the Customers.csv dataset) and transaction data (from the Transactions.csv dataset). The dataset was preprocessed and normalized before applying the KMeans clustering algorithm, followed by the evaluation of clustering quality using various metrics.

Data Preprocessing:

1. **Customer Information**: The Region column was one-hot encoded, and other relevant profile features were retained.
2. **Transaction Data**: Transaction data was aggregated to calculate key metrics like TotalSpent, AvgSpent, and PurchaseCount for each customer.
3. **Feature Selection**: Non-relevant columns such as CustomerID, CustomerName, and SignupDate were excluded from the clustering features.

Clustering Algorithm:

- **KMeans** clustering was used, and the number of clusters was selected after testing different values. Based on the results, 7 clusters were chosen as the optimal number of clusters.

Metrics Evaluation:

1. **Number of Clusters Formed**:
    - **7 clusters** were formed based on the KMeans algorithm. This number was selected after testing different options, with 7 showing the best balance of clustering quality and interpretability.
2. **DB Index (Davies-Bouldin Index)**:
    - **DB Index**: 0.8535
        - The DB Index evaluates the separation between clusters. A lower value indicates better clustering, as the clusters are more distinct and separated. In this case, the DB Index of 0.8535 suggests that the clusters are fairly well-separated, although there may be some overlap in certain areas that could be improved.
3. **Silhouette Score**:
    - **Silhouette Score**: 0.4269
        - The Silhouette Score measures how similar each data point is to its own cluster compared to other clusters. A higher score indicates well-defined clusters. A score of 0.4269 indicates that the clusters are moderately well-separated, though there might be some overlap in certain cases, which can be addressed by experimenting with the number of clusters or using a different clustering algorithm.
4. **KMeans Inertia**:

- **KMeans Inertia**: 393.9124
    - Inertia is a measure of how well the data points fit within their assigned clusters. Lower inertia values indicate better clustering performance. An inertia of 393.9124 suggests that the clusters are somewhat compact but not perfectly so. Reducing inertia further could be achieved by fine-tuning the clustering process or adjusting the number of clusters.

Cluster Visualization:

- **PCA-based 2D Plot**: The clusters were visualized using Principal Component Analysis (PCA) to reduce the data to 2 dimensions. The visualization showed the separation between the different clusters, with customers grouped into distinct regions based on their profile and transaction features.

Conclusion and Next Steps:

- **Clustering Logic**: The KMeans clustering algorithm successfully segmented the customers into 7 groups based on their profile and transaction data. The DB Index, Silhouette Score, and Inertia values indicate reasonable clustering quality, but there is room for further improvement.
- **Improvement**:
    - Experimenting with other clustering algorithms like **DBSCAN** or **Agglomerative Clustering** might provide different insights.
    - Further tuning of the number of clusters could improve the separation between customer groups.
    - Additional features, such as customer demographics or product preferences, could be incorporated to refine the segmentation.