Clustering Analysis Report: Customer Segmentation

1. Objective:

The task aimed at performing customer segmentation using clustering techniques on a dataset that combines **customer profile data** and **transaction data**. The goal was to uncover distinct customer segments that could provide insights for business decision-making.

2. Data Description:

The analysis is based on two datasets:

- **Customers.csv**
  - **CustomerID**: Unique identifier for each customer.
  - **CustomerName**: Name of the customer.
  - **Region**: Continent where the customer resides.
  - **SignupDate**: Date when the customer signed up.
- **Transactions.csv**
  - **TransactionID**: Unique identifier for each transaction.
  - **CustomerID**: ID of the customer who made the transaction.
  - **ProductID**: ID of the product sold.
  - **TransactionDate**: Date of the transaction.
  - **Quantity**: Quantity of the product purchased.
  - **TotalValue**: Total value of the transaction.
  - **Price**: Price of the product sold.

The datasets were merged using **CustomerID** to form a single dataset, which combined the transaction data with customer profile data.

3. Data Preprocessing:

- **Merging**: The `Transactions.csv` and `Customers.csv` datasets were merged on the `CustomerID` column using an **inner join**. This ensured that only customers with transaction records were retained.
- **Features Selection**: The relevant features selected for clustering were:
  - **TotalValue**: The total value of each transaction.
  - **Quantity**: The quantity of products purchased in each transaction.
  - **Price**: The price of the products purchased.

These features were selected as they provide meaningful insight into the purchasing behavior of customers.

- **Scaling**: The selected features (`TotalValue`, `Quantity`, and `Price`) were standardized using **StandardScaler**. This scaling ensures that all features are on a similar scale, which is critical for clustering algorithms like DBSCAN.

4. Clustering Approach:

Two clustering techniques were employed:

- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**: DBSCAN is a density-based clustering algorithm that does not require the number of clusters to be specified beforehand. It identifies clusters based on the density of data points and can also detect outliers (points that do not belong to any cluster).
    - **Parameters** used for DBSCAN:
        - `eps`: The maximum distance between two points to be considered as neighbors. Set to 0.5.
        - `min_samples`: The number of points required to form a dense region. Set to 5.
    - **Cluster Output**: DBSCAN assigns each data point to a cluster. Points that do not meet the density criteria are labeled as noise (`-1`).
- **t-SNE (t-Distributed Stochastic Neighbor Embedding)**: t-SNE is a dimensionality reduction technique that is particularly suited for visualizing high-dimensional data. It projects the data onto a 2D plane for easy visualization of clusters. In this analysis, t-SNE was used to visualize the DBSCAN clusters in two dimensions.

5. Results:

- **Number of Clusters**: DBSCAN identified **4 distinct clusters** in the dataset, as well as some noise points labeled as `-1`.
    - Cluster 0: Contains customers with moderate transaction values.
    - Cluster 1: Contains customers with slightly higher transaction values.
    - Cluster 2: Contains customers with significantly higher transaction values.
    - Cluster 3: Contains customers with the highest transaction values.
- There were no noise points (`-1`), which indicates that all points were assigned to some cluster.
- Note that the dataset, didn't even have any duplicate or missing values .
- **DB Index (Dunn's Index)**: The DB Index score, which evaluates the **separation** between clusters and the **compactness** of the clusters, was calculated and the score was **2.18**.
    - **Interpretation**: A higher DB Index indicates better clustering quality. In this case, a score of 2.18 suggests that the clusters are reasonably well-separated and compact. However, further tuning of DBSCAN parameters could potentially improve the DB Index.

6. Visualizations:

- **t-SNE Visualization**: The t-SNE plot showed the 4 clusters as distinct groups of points, each cluster represented by a different color.

- The separation between clusters is visually clear, and no significant overlap was observed, suggesting that the clustering algorithm performed well in distinguishing between customer segments.

7. Business Insights:

Based on the clustering analysis, the following business insights can be inferred:

1. **Customer Segmentation**:

    - **Cluster 0**: Likely represents customers with average spending habits. Targeting these customers with personalized offers or loyalty programs could help increase their transaction value.
    - **Cluster 1**: Customers in this cluster may be good candidates for upselling. Offering them promotions on higher-value products may lead to increased sales.
    - **Cluster 2**: These customers have a higher transaction value. They could be targeted with exclusive premium products or services.
    - **Cluster 3**: This cluster likely contains the top spenders. Offering VIP treatment, personalised services, or special rewards could increase customer loyalty.

2. **Targeted Marketing**: Understanding these segments allows businesses to design personalised marketing campaigns tailored to each cluster's behaviour.

3. **Product Strategy**: For customers in higher-value clusters, offering exclusive or premium products could drive even higher sales. For lower-value clusters, promotions on popular products could be effective in increasing their transaction value.

8. Conclusion:

- The clustering analysis was successful in segmenting customers into distinct groups based on their transaction data.
- The **DBSCAN** algorithm provided useful insights, with moderate separation between clusters and a reasonably good **DB Index** of 2.18.
- Future improvements could include:
    - Fine-tuning DBSCAN parameters (`eps`, `min_samples`) for better performance.
    - Experimenting with other clustering algorithms such as **KMeans** or **Agglomerative Clustering** for comparison.

This segmentation can be leveraged for targeted marketing, customer retention, and personalized offers to enhance business decision-making.