1. Loading and Merging Data:

- Load both customer and transaction data into DataFrames (customers, transactions).
- Merge them using a common identifier (CustomerID).

2. Feature Selection:

- We select customer profile features like Age and Income along with transaction-related features like TotalValue and Quantity .

3. Normalization:

- Normalize the features using StandardScaler to ensure all features contribute equally to the clustering process.

4. Clustering:

- We apply **KMeans clustering** to segment the customers into clusters. We specify n_clusters=4 here, but you can experiment with values between 2 and 10.
- The cluster assignments are stored in the Cluster column.

5. Evaluation - Davies-Bouldin Index:

- The **Davies-Bouldin Index (DBI)** is calculated using the davies_bouldin_score function. Lower DBI values indicate better clustering.

6. Visualization:

- The high-dimensional data (after normalization) is reduced to 2D using **PCA** and **t-SNE** for visualization.
- We plot the clusters on a 2D plane, where each point represents a customer and colors represent different clusters.

Clustering on Price, TotalValue, and Quantity can reveal interesting patterns in your transaction data.

Potential Insights:

- **Identify Product Bundles/Groups:**
    - Products with similar pricing, total value, and quantity sold might be frequently purchased together or belong to the same product category.
    - This can help with product recommendations, cross-selling, and merchandising.
- **Customer Segmentation:**
    - **High-value customers:** Cluster products with high TotalValue and potentially high Quantity to identify products favored by high-spending customers.
    - **Value-conscious customers:** Cluster products with lower Price or higher Quantity per TotalValue, suggesting customers seeking value or buying in bulk.

    - **Impulse purchases:** Cluster products with low Quantity but potentially higher Price, indicating items that might be bought on a whim.

- **Inventory Management:**
    - Identify slow-moving vs. fast-moving products based on Quantity sold.
    - Optimize inventory levels based on product clusters and demand patterns.

Clustering Techniques:

- **K-Means:** A popular choice for its simplicity. However, it can be sensitive to outliers and may not perform well with non-spherical clusters.
- **DBSCAN:** Can identify clusters of arbitrary shapes and can handle noise effectively.