1. Importing Libraries

- Libraries are essential tools for handling data, visualization, and machine learning.
  - pandas: Used for data manipulation (e.g., reading and merging datasets).
  - numpy: Handles numerical operations.
  - matplotlib.pyplot & seaborn: For creating visualizations like scatterplots and line charts.
  - sklearn.preprocessing.StandardScaler: Scales data to ensure fair comparisons across features during clustering.
  - sklearn.cluster.KMeans: Performs K-Means clustering.
  - sklearn.metrics: Used to compute clustering evaluation metrics like silhouette scores and DB scores.

2. Loading the Datasets

- Three CSV files are read into **DataFrames**:
  - Customers.csv: Contains customer details, such as IDs, names, and regions.
  - Products.csv: Contains product information like product IDs, names, and prices.
  - Transactions.csv: Logs of purchases, including product IDs, customer IDs, quantities, and transaction values.
- The pd.read_csv() function is used to load these files.

3. Merging the Datasets

- The three datasets are combined to create a unified dataset containing:
  - Customer details (from Customers.csv).
  - Product details (from Products.csv).
  - Transaction details (from Transactions.csv).
- **Keys for Merging**:
  - CustomerID: Links customers to their transactions.
  - ProductID: Links transactions to product details.
- The pd.merge() function merges datasets by matching values in these keys.

Example:

If a customer buys a product:

- The CustomerID ensures the customer's name, region, and ID are included.
- The ProductID links to the product's name and price.

4. Exploring the Merged Data

- After merging, a **combined DataFrame** is created with columns like:
  - TransactionID, ProductName, CustomerName, Region, Price, Quantity, TotalValue, etc.
- Dimensions: 1000 rows and 13 columns (these represent all transaction records).

Example Row:

TransactionID	ProductName	CustomerName	Region	Quantity	TotalValue
101	Coffee Maker	John Doe	West	2	$200

5. Cleaning the Data

- After merging, redundant or unnecessary columns are removed:
  - Price_x and Price_y (duplicates created during merging) are dropped.
  - The remaining Price column is kept for consistency.

6. Aggregating Data for Clustering

- Aggregation groups customer-level data to create meaningful clustering features:
  - Group by CustomerID:
    - Total Spending (sum of TotalValue).
    - Total Quantity Purchased (sum of Quantity).
    - Average Price (mean of Price).
- Result: A compact DataFrame where each row represents a single customer with aggregated values.

Example:

CustomerID	TotalSpending	TotalQuantity	AvgPrice
101	$500	10	$50

7. Scaling the Data

- **Why Scaling?**
  - Features like Total Spending and Total Quantity have different ranges (e.g., $500 vs. 10 units). If not scaled, clustering will give more weight to features with larger values.
- **How?**
  - Use StandardScaler to transform the data into a standard normal distribution (mean = 0, standard deviation = 1).
  - Scaled features ensure fair comparisons during clustering.

8. K-Means Clustering

- K-Means groups customers into clusters based on their similarity in the scaled data.
- Clusters are determined by:
    - Distance between points (customers) and cluster centroids.
    - Iterative adjustments of centroids until the best grouping is achieved.

9. Evaluating Clusters

- Metrics are calculated for cluster numbers ranging from 2 to 10:
    - **WCSS (Within-Cluster Sum of Squares)**:
        - Measures how tightly grouped the points in each cluster are. Lower WCSS means better compactness.
    - **Silhouette Score**:
        - Ranges from -1 to 1.
        - A higher score indicates well-defined clusters with good separation.
    - **DB (Davies-Bouldin) Score**:
        - Measures how distinct the clusters are. Lower values indicate better-defined clusters.

Results of Metrics:

- For each number of clusters (k), WCSS, silhouette score, and DB score are calculated and printed.
- **Best cluster size (k)**:
    - Cluster size 8 is chosen because it has:
        - A reasonable silhouette score.
        - A low DB score (0.876).

10. Visualizing the Elbow Plot

- The **Elbow Method** is used to identify the optimal number of clusters by plotting WCSS against the number of clusters (k).
- The plot shows:
    - As k increases, WCSS decreases (clusters become more compact).
    - The "elbow" (a point where the slope flattens) suggests the best k. For this dataset, 8 clusters are chosen.

11. Final Clustering

- K-Means is run with k=8 to group customers into 8 distinct segments.

- Each customer is assigned a **cluster label** (0 to 7) based on their group.

12. Visualizing the Clusters

- A scatterplot is created to visualize the 8 clusters:
  - X-axis: Total Spending
  - Y-axis: Total Quantity Purchased
  - Each point represents a customer, and colors represent their cluster labels.

Observations:

- Customers with similar spending and purchasing patterns are grouped together.
- For example:
  - Cluster 0: High spenders with high quantity.
  - Cluster 5: Low spenders with low quantity.

Key Takeaways

- This analysis helps identify customer segments for targeted marketing:
  - High spenders (Cluster 0) may receive premium offers.
  - Low spenders (Cluster 5) may get discounts to encourage more purchases.
- Business decisions can be personalized for each cluster, improving customer satisfaction and sales.