TASK – 3: Customer Segmentation / Clustering

Objective: The goal of this task is to perform customer segmentation using clustering techniques based on both **profile information** (from the Customers.csv file) and **transaction data** (from the Transactions.csv file). The primary objective is to group customers with similar behaviors and characteristics into clusters.

Dataset

- **Customers.csv** contains customer profile information such as CustomerID, CustomerName, Region, and SignupDate.
- **Transactions.csv** contains transaction-related data, including CustomerID, TransactionID, ProductID, Quantity, TotalValue, and TransactionDate.

Data Preprocessing

The first step was to load both the Customers.csv and Transactions.csv datasets using pandas and merge them based on the CustomerID column to create a combined dataset.

Feature Engineering

- TotalSpend: The total spending for each customer.
- AverageSpend: The average spending per transaction for each customer.
- TransactionCount: The number of transactions for each customer.
- Days: Calculating the customer's tenure in days by subtracting their SignupDate from the current date.

One hot encoding

The Region column is encoded using OneHotEncoder, converting it into binary columns representing the presence of each region. This makes it suitable for use in machine learning models.

Normalization

The StandardScaler is used to normalize the numerical features to ensure that the clustering algorithm performs well. Normalization is important because it ensures that all features contribute equally to the distance metric used by KMeans.

KMeans Clustering

- I chose KMeans with 5 clusters, which I can fine-tune if necessary.
- After fitting the model, I predicted the cluster labels for each customer and added them to the customer_features dataframe.

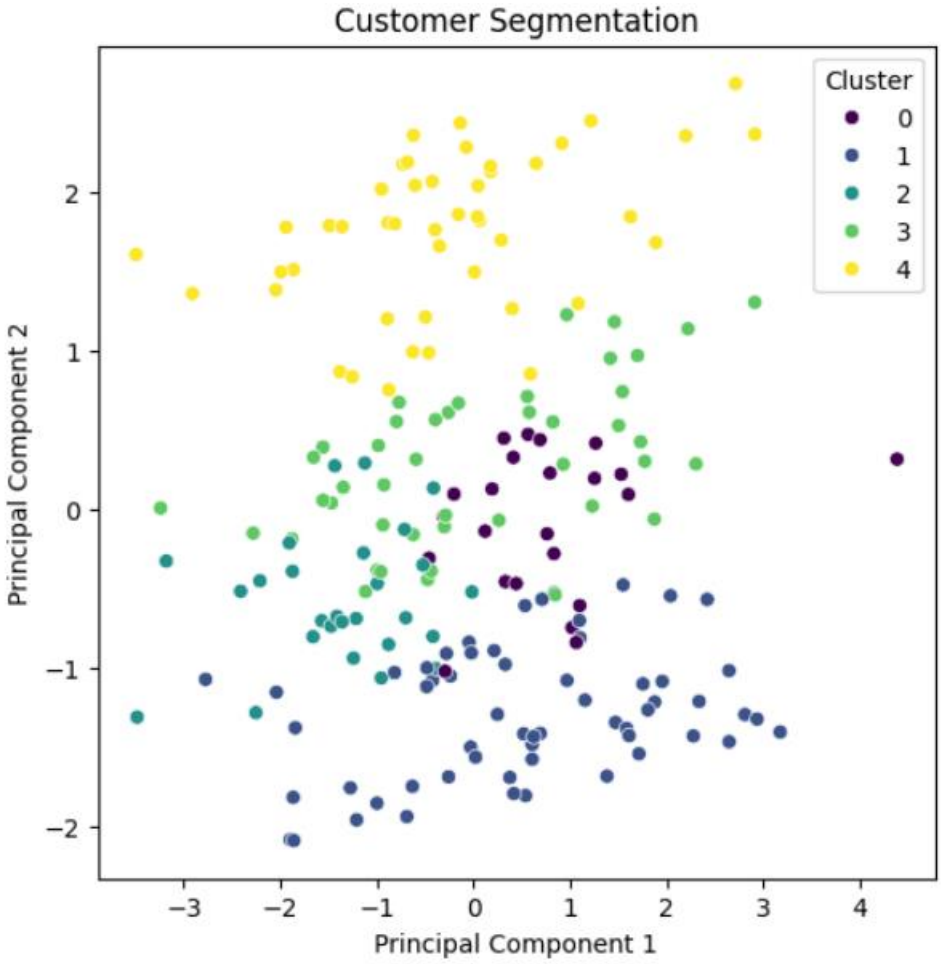
DB Index

I calculated the **Davies-Bouldin Index (DBI)** to evaluate the clustering performance. A lower DBI indicates better clustering, and it will help in choosing the optimal number of clusters.

Dimensionality Reduction (PCA)

Using **Principal Component Analysis (PCA)**, I reduced the data's dimensions to two principal components for visualization purposes. PCA is useful for visualizing high-dimensional data in a 2D space.

Visualization I used **Seaborn's scatter plot** to visualize the clustering results, where each point represents a customer, and the colors represent the assigned cluster.



Finally, I saved the customer_features dataframe to a CSV file to store the clustering results along with the features.