Customer Segmentation / Clustering

-by Kowshik R 27/01/2025 official.kowshik.r@gmail.com

Summary

This implementation focuses on clustering customer data using the **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** method. The dataset is preprocessed by merging customer and transaction data, engineering features such as transaction count, average value, and days since signup, and encoding categorical regions. Features are scaled for consistency, and DBSCAN is applied with eps=1.2 and min_samples=1 to identify dense clusters while excluding noise points. The clustering performance is evaluated using the **Davies-Bouldin Index**, which measures compactness and separation of clusters. A scatterplot visualizes clusters, showing group distributions based on scaled TotalValue and AvgTransactionValue, highlighting the results.

Approach

1. **Data Loading and Preprocessing**
    - The Customers.csv and Transactions.csv datasets are merged on CustomerID.
    - Transaction dates and signup dates are converted to datetime objects to compute new features like DaysSinceSignup.
2. **Feature Engineering**
    - Aggregated metrics such as TotalValue (sum), AvgTransactionValue (mean), TransactionCount, and AvgDaysSinceSignup are computed for each customer.
    - The categorical Region column is encoded into numerical values using **Label Encoding**.
3. **Feature Normalization**
    - Since clustering algorithms are sensitive to scales, numerical features are standardized using **StandardScaler** to ensure uniformity.
4. **Clustering with DBSCAN**
    - DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is used as the clustering algorithm. It identifies clusters based on data density, making it robust to noise and adaptable for non-linear cluster shapes.
    - The eps (neighborhood radius) is set to 1.2, and min_samples (minimum data points in a cluster) is set to 1, allowing for the identification of small, dense groups in the data.
    - Clusters are evaluated using the **Davies-Bouldin Index (DB Index)**, which measures cluster separation and compactness (lower values indicate better clustering).

5. **Visualization**
    - The results are visualized using scatter plots of the most relevant features (TotalValue and AvgTransactionValue), where clusters are color-coded for easy interpretation.

DBSCAN for Clustering

DBSCAN is chosen because it offers significant advantages:

- It automatically identifies noise points (outliers) that do not belong to any cluster, improving result reliability.
- Unlike KMeans or hierarchical clustering, DBSCAN does not require specifying the number of clusters beforehand, which is particularly useful when the structure of data is unknown.
- It can identify clusters of arbitrary shapes and is less sensitive to initialization, unlike KMeans.

In this dataset, where customer transactions vary significantly, DBSCAN provides better flexibility to group customers based on density rather than forcing them into pre-defined cluster counts.

Results

- **Clusters Identified:** The DBSCAN algorithm identified several clusters (and some noise points).
- **Davies-Bouldin Index:** The **DB Index for DBSCAN is 0.66**, indicating reasonably compact and well-separated clusters.
- **Visualization:** The scatterplot of clusters (based on TotalValue and AvgTransactionValue) clearly depicts customer groups with distinct purchasing behaviors. Noise points are visualized separately, representing customers with atypical transaction patterns.

Overall, DBSCAN effectively grouped customers with similar behaviors while excluding outliers, offering actionable insights for segmentation and targeting.

Output:

DBSCAN DB Index (filtered noise): 0.66024704948174
Number of clusters (excluding noise): 15