Customer segmentation/clustering

Explanation of the Code and Process

This script focuses on clustering customers based on their transaction behavior and evaluates the quality of the clusters using the Davies-Bouldin Index.

Steps in the Script

Import Required Libraries Libraries used:

NumPy and Pandas: For data manipulation.

Scikit-learn: For data scaling, clustering, and clustering evaluation.

Load Data Data is read from three CSV files containing customer, product, and transaction information:

Customers.csv

Products.csv

Transactions.csv

```
Cust_data = pd.read_csv('Customers.csv')

Prod_data = pd.read_csv('Products.csv')

Trans_data = pd.read_csv('Transactions.csv')
```

Merge Datasets Transactions are merged with product data using ProductID, and the resulting dataset is further merged with customer data using CustomerID.

```
transactions = pd.merge(Trans_data, Prod_data, on="ProductID", how="left")

data = pd.merge(transactions, Cust_data, on="CustomerID", how="left")
```

Aggregate Customer Features The script calculates customer-level features:

total_spent: Total value of all transactions (TotalValue).

transaction_count: Number of transactions (TransactionID).

distinct_products: Number of unique products purchased (ProductID).

The result is stored in the customer_features DataFrame.

```
customer_features = data.groupby("CustomerID").agg(
    total_spent=("TotalValue", "sum"),
    transaction_count=("TransactionID", "count"),
    distinct_products=("ProductID", "nunique")
).reset_index()
```

Normalize Features

Features are scaled to the range [0, 1] using MinMaxScaler to ensure all variables contribute equally during clustering.

```
scaler = MinMaxScaler()
customer_features_scaled = scaler.fit_transform(customer_features.iloc[:, 1:])
```

Apply K-Means Clustering

A K-Means clustering model with 4 clusters is trained using the normalized customer features.

The predicted cluster labels (clusters) are added as a new column to the customer_features DataFrame.

```
kmeans = KMeans(n_clusters=4, random_state=42)
clusters = kmeans.fit_predict(customer_features_scaled)
customer_features["Cluster"] = clusters
```

Evaluate Clustering with Davies-Bouldin Index

The Davies-Bouldin Index (DBI) measures clustering quality. A lower DBI indicates better-defined clusters.

DBI is computed using the scaled features and cluster labels.

```
db_index = davies_bouldin_score(customer_features_scaled, clusters)
```

Key Outputs

Clustered Customer Features:

Each customer is assigned a cluster label (0, 1, 2, or 3), indicating the group they belong to.

The customer_features DataFrame includes columns like total_spent, transaction_count, distinct_products, and the new Cluster column.

Davies-Bouldin Index:

The computed db_index represents the quality of the clusters. Lower values (closer to 0) indicate better clustering performance.

Interpretation of Results

Clusters:

Customers are grouped into 4 clusters based on their spending, transaction count, and diversity of products purchased.

Each cluster represents a segment with similar purchasing behavior.

Davies-Bouldin Index:

A high DBI indicates overlapping or poorly defined clusters, while a low DBI suggests well-separated clusters.

Example Use Cases

Marketing Campaigns:

Target specific clusters with tailored marketing strategies (e.g., high-spending customers in one cluster).

Product Recommendations:

Suggest products based on popular purchases within a cluster.

Customer Retention:

Identify low-spending clusters and design strategies to boost engagement.

Next Steps

Visualization:

Use PCA or t-SNE for dimensionality reduction and visualize clusters.

Create bar plots or pie charts showing the distribution of customers across clusters.

Cluster Analysis:

Analyze each cluster to understand its characteristics (e.g., average spending, product diversity).

Parameter Tuning:

Experiment with different values of n_clusters in K-Means to optimize the clustering performance. Evaluate each using DBI or other metrics.

This workflow provides a solid foundation for customer segmentation and targeted interventions.