Customer Segmentation using KMeans Clustering

Overview

This project applies KMeans clustering to segment customers based on their spending behavior and transaction patterns. The goal is to identify customer segments that can be targeted for personalized marketing, product recommendations, or other business strategies.

Steps Involved

1. Data Loading and Preprocessing

- **Datasets**: The project works with three datasets: Customers.csv, Products.csv, and Transactions.csv. These datasets are loaded using the pandas library.
- **Datetime Parsing**: The columns TransactionDate from the Transactions.csv and SignupDate from the Customers.csv are converted into datetime format for further analysis.
- **Merging Datasets**: The datasets are merged into a single unified dataset based on common identifiers like CustomerID and ProductID.

2. Feature Engineering

- For each customer, key features are engineered:
  - **Total Spending**: The total value of all transactions made by the customer.
  - **Transaction Count**: The number of transactions made by the customer.
  - **Average Purchase Value**: The average value of products purchased by the customer.
- Additional features like **Region** are retained for further analysis.

3. Data Transformation

- **Categorical Encoding**: The Region feature is encoded into dummy variables, transforming it into multiple binary columns for each region.
- **Normalization**: The numerical features (Total Spending, Transaction Count, Average Purchase Value) are normalized using StandardScaler to ensure they are on the same scale before applying clustering.

4. Clustering using KMeans

- **KMeans Clustering**: The KMeans algorithm is applied for different cluster counts ranging from 2 to 10. The number of clusters is determined by evaluating the Davies-Bouldin Index (DB Index), which measures the separation between clusters. A lower DB Index indicates better clustering.
- **Optimal Cluster Selection**: The optimal number of clusters is selected based on the lowest DB Index.

5. Visualization

- A scatter plot is generated to visualize the customer segments. The plot is created using Total Spending and Average Purchase Value as the axes, with the clusters represented by different colors.

6. Output

- The final dataset, which includes the assigned cluster labels for each customer, is saved as Customer_Clusters.csv for further analysis or business use.

Requirements

This project requires the following Python libraries:

- pandas
- numpy
- scikit-learn
- matplotlib
- seaborn

To install the dependencies, you can use the following commands:

```
pip install pandas numpy scikit-learn matplotlib seaborn
```

How to Use

1. **Prepare Your Datasets**: Ensure you have the Customers.csv, Products.csv, and Transactions.csv files in the working directory.
2. **Run the Script**: Execute the script to load, preprocess, and apply KMeans clustering to the data.
3. **Check the Results**:
   - The optimal number of clusters and their characteristics will be printed.
   - A scatter plot will be displayed showing customer segments based on spending behavior.
4. **Output**: The clustered data will be saved to Customer_Clusters.csv for further use.

Conclusion

This project showcases the power of clustering for customer segmentation. By grouping customers based on spending and transaction patterns, businesses can better understand their customer base and tailor marketing efforts to each segment.