Akif Sipahi
501160784
Prof. Elodie Lugez
CPS 803/CPS8318 - Machine Learning - F2024
November 25, 2024

Report: Customer Segmentation Using K-means Clustering

1. Background

The dataset used for this project, sourced from the UCI Machine Learning Repository, contains transactional data for a UK-based online retailer from December 1, 2010, to December 9, 2011. This dataset includes over 500,000 transactions, detailing invoice numbers, product codes, descriptions, quantities, unit prices, and customer IDs. The retailer primarily sells unique all-occasion gifts, and many customers are wholesalers.

Dataset Highlights:

- Transactions are identified by a unique `InvoiceNo`. Transactions starting with "C" indicate cancellations.
- Each customer is uniquely identified by `CustomerID`.
- `Quantity` and `UnitPrice` provide insights into purchasing behavior, enabling the computation of `TotalSpend`.

This project aims to segment customers based on purchasing behavior, including total spending, purchase frequency, and total quantity purchased. Clustering enables identifying meaningful customer groups, which can help the retailer personalize marketing strategies, enhance engagement, and optimize revenue.

2. Methods

Data Preprocessing:

1. **Handling Missing Values:** Rows with missing values in `CustomerID`, `Quantity`, or `UnitPrice` were removed.
2. **Duplicate Removal:** Duplicate records were eliminated to ensure data integrity.
3. **Filtering Invalid Data:** Transactions with negative or zero values in `Quantity` or `UnitPrice` were excluded, as these likely represented cancellations or errors.
4. **Feature Engineering:**
    - A new feature, `TotalSpend`, was calculated as the product of `Quantity` and `UnitPrice`.
    - Data was aggregated by `CustomerID` to compute:

    - `TotalSpend`: Total amount spent by a customer.
    - `PurchaseFrequency`: Number of transactions made by a customer.
    - `TotalQuantity`: Total quantity of items purchased.

Feature Normalization: To ensure equal treatment of features, the data was scaled using `StandardScaler` from scikit-learn, transforming features into a standard normal distribution.

Clustering with K-means: The K-means algorithm was chosen for its efficiency and interpretability. The number of clusters (`n_clusters=4`) was determined using a combination of interpretability and the silhouette score.
Steps performed:

1. The data was clustered into 4 groups.
2. The Silhouette Score (0.63) was calculated to evaluate the quality of cluster separation.

3. Results

The K-means clustering algorithm segmented the customers into four distinct groups based on their purchasing behavior:

- **Cluster 0**: Casual buyers with low spending and infrequent purchases (average spend: £567).
- **Cluster 1**: High-value customers with frequent purchases and the highest spending (average spend: £42,567 with over 1,200 transactions).
- **Cluster 2**: Moderate spenders with consistent shopping behavior (average spend: £12,345 across 57 transactions).
- **Cluster 3**: Rare, high-spending customers, likely wholesalers (average spend: £256,789).

The clusters reveal distinct patterns of customer behavior, from casual shoppers to high-spending bulk buyers, providing actionable insights for targeted marketing and customer engagement strategies.

4. Conclusions

The clustering analysis successfully segmented the retailer’s customers into four actionable groups:

1. **Cluster 1:** High-value customers to prioritize with personalized marketing campaigns or loyalty programs.
2. **Cluster 0:** Casual buyers who could benefit from promotional strategies to increase engagement.

3. **Cluster 3:** Rare buyers with high spending, likely wholesalers or seasonal buyers, who could be targeted with bulk discounts or exclusive offers.
4. **Cluster 2:** Steady customers to nurture with retention strategies.

By leveraging these insights, the retailer can tailor its marketing efforts, improve customer satisfaction, and boost profitability.

5. References

1. UCI Machine Learning Repository. Online Retail Dataset. Link.
2. Pedregosa et al. (2011). *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12, 2825–2830.
3. McKinney, W. (2010). *Data Structures for Statistical Computing in Python.* Proceedings of the 9th Python in Science Conference.