CUSTOMER SEGMENTATION USING K-MEANS CLUSTERING ON E-COMMERCE DATA

Executive Summary

This case study demonstrates the application of unsupervised learning—specifically K-Means clustering—to segment customers of an e-commerce company based on their purchasing behavior. It guides students through data cleaning, feature scaling, elbow method, cluster interpretation, and visualization. The project offers a highly practical use case of clustering that is often assigned in data science courses or job interviews, helping students bridge theory and business application.

1. Introduction

Customer segmentation is a foundational application of machine learning in marketing analytics. It allows companies to identify meaningful groups of customers and target them with personalized strategies. K-Means clustering is a simple yet powerful algorithm to divide customers based on behavioral patterns such as purchase frequency, average spend, and recency.

2. Problem Statement

Segment customers into distinct groups based on their purchasing activity, using variables such as:

- Recency (days since last purchase)
- Frequency (total number of purchases)
- Monetary Value (total amount spent)
- Tenure (days since becoming a customer)

3. Dataset Overview

- **Source**: Simulated transactional data from a retail e-commerce platform
- **Observations**: 5,000 unique customers
- **Features**:
    - CustomerID
    - Recency (days)

- Frequency (counts)
- Monetary ($)
- Tenure (days)

4. Methodology

Step 1: Preprocessing

- Removed missing values
- Normalized features using MinMaxScaler (scaling improves clustering accuracy)

Step 2: Elbow Method to Determine Optimal Clusters

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_data)
    sse.append(kmeans.inertia_)

plt.plot(range(1, 11), sse)
plt.xlabel("Number of Clusters")
plt.ylabel("SSE")
```

- Optimal clusters: **4**

Step 3: Apply K-Means

```
kmeans = KMeans(n_clusters=4, random_state=42)
clusters = kmeans.fit_predict(scaled_data)
df['Cluster'] = clusters
```

5. Cluster Interpretation

Cluster	Recency	Frequency	Monetary	Tenure	Interpretation
0	High	Low	Low	Short	New customers, not yet engaged
1	Low	High	High	Long	Loyal high-value customers
2	Medium	Medium	Medium	Medium	Average customers
3	High	High	Low	Long	Frequent, low-spending users

6. Visualizations

- **Scatter plot** of Monetary vs Frequency colored by cluster
- **Boxplot** of Recency across clusters
- **Radar chart** comparing average values of each cluster

7. Business Insights

- **Cluster 1** should be targeted with loyalty rewards and early access offers
- **Cluster 0** can be nurtured with onboarding emails and discounts
- **Cluster 3** needs conversion strategies to increase spend per order

8. Conclusion

This case study demonstrates how unsupervised learning can be used to understand customer behavior at scale. K-Means clustering revealed patterns that are not immediately visible from raw data. Such segmentation helps students connect statistical modelling with tangible business actions.

9. Learning Outcomes for Students

- Understand the logic and math behind K-Means
- Learn practical preprocessing techniques for unsupervised learning
- Visualize and interpret clusters meaningfully
- Translate analysis into business recommendations

10. Suggested Enhancements

- Compare with DBSCAN or Hierarchical clustering
- Perform dimensionality reduction using PCA for better visuals
- Apply cluster-based marketing strategies using A/B testing logic
- Extend to product-based clustering

11. References

- Han, Kamber, & Pei. *Data Mining: Concepts and Techniques*
- Scikit-Learn Documentation – KMeans
- Kaggle: E-Commerce Data Customer Segmentation
- MIT OpenCourseWare – Clustering Algorithms