IntelliHack 5.0

Team: CodeLabs

Task 02

Initial round

Table of Content

Customer Segmentation Analysis- E-commerce User Behavior Clustering

Executive Summary

This report details the development and implementation of a customer segmentation model for an e-commerce platform using clustering algorithms. The model successfully identified three distinct customer segments from transactional and behavioral data: Bargain Hunters, High Spenders, and Window Shoppers. These segments align with expected customer archetypes and provide actionable insights for targeted marketing strategies. The K-Means clustering algorithm was selected as the final model based on its superior performance across multiple evaluation metrics. Visualizations using dimensionality reduction techniques confirmed the clear separation between the identified segments.

1. Introduction

1.1 Project Overview

Customer segmentation is a critical strategy for e-commerce businesses seeking to understand their customer base and deliver personalized experiences. This project aimed to identify distinct customer segments based on purchasing behavior, browsing patterns, and discount usage. By leveraging unsupervised learning techniques, we were able to discover natural groupings within the customer data that can inform targeted marketing strategies and optimize resource allocation.

1.2 Business Objectives

The primary objectives of this segmentation analysis were to

- Identify distinct customer groups with similar behavioral patterns
- Understand the characteristics and preferences of each segment
- Develop targeted marketing strategies for each customer segment
- Improve customer satisfaction through personalized experiences
- Optimize resource allocation for marketing and customer retention efforts

2. Data Overview and Preprocessing

2.1 Dataset Description

The dataset contained customer interaction data from an e-commerce platform with the following features:

- **total_purchases**: Number of purchases made by the customer
- **avg_cart_value**: Average value of items in the customer's cart
- **total_time_spent**: Total time spent on the platform (in minutes)
- **product_click**: Number of products viewed by the customer
- **discount_counts**: Number of times the customer used a discount code
- **customer_id**: Unique identifier for each customer

2.2 Data Cleaning and Preparation

The initial data exploration revealed some missing values that needed to be addressed. After removing rows with missing values, the dataset was ready for feature engineering.

Before data cleaning

```
Basic statistics:
       total_purchases  avg_cart_value  total_time_spent  product_click  \
count       979.000000      979.000000        999.000000     979.000000
mean         11.570991       75.457978         49.348759      28.237998
std           7.016327       55.067835         32.730973      16.296384
min           0.000000       10.260000          5.120000       4.000000
25%           6.000000       33.130000         22.375000      16.000000
50%          10.000000       49.380000         40.360000      21.000000
75%          17.000000      121.255000         77.170000      45.000000
max          32.000000      199.770000        119.820000      73.000000

       discount_counts
count       999.000000
mean          4.313313
std           4.532772
min           0.000000
25%           1.000000
50%           2.000000
75%           8.000000
max          21.000000
```

After data cleaning

```
Engineered features:
       price_per_purchase  time_per_click  discount_per_purchase  \
count          979.000000      979.000000             979.000000
mean            10.016360        1.759312               0.337339
std              9.215514        0.802848               0.293979
min              0.000000        0.222000               0.000000
25%              2.135394        1.246929               0.127717
50%              8.856923        1.704167               0.285714
75%             14.592727        2.160791               0.500000
max             76.610000        7.485000               2.000000

       clicks_per_purchase
count           979.000000
mean              5.129158
std               6.871201
min               0.000000
25%               0.941176
50%               2.000000
75%               8.166667
max              60.000000
df_clean shape: (979, 10)
```

2.3 Feature Engineering

To better capture customer behavior patterns, several derived features were created. These derived features provide additional insights by capturing the relationships between original features.

- **price_per_purchase**: Average spending per transaction
- **time_per_click**: Browsing efficiency (time spent per product viewed)
- **discount_per_purchase**: Discount usage frequency per purchase
- **clicks_per_purchase**: Browsing-to-purchase ratio

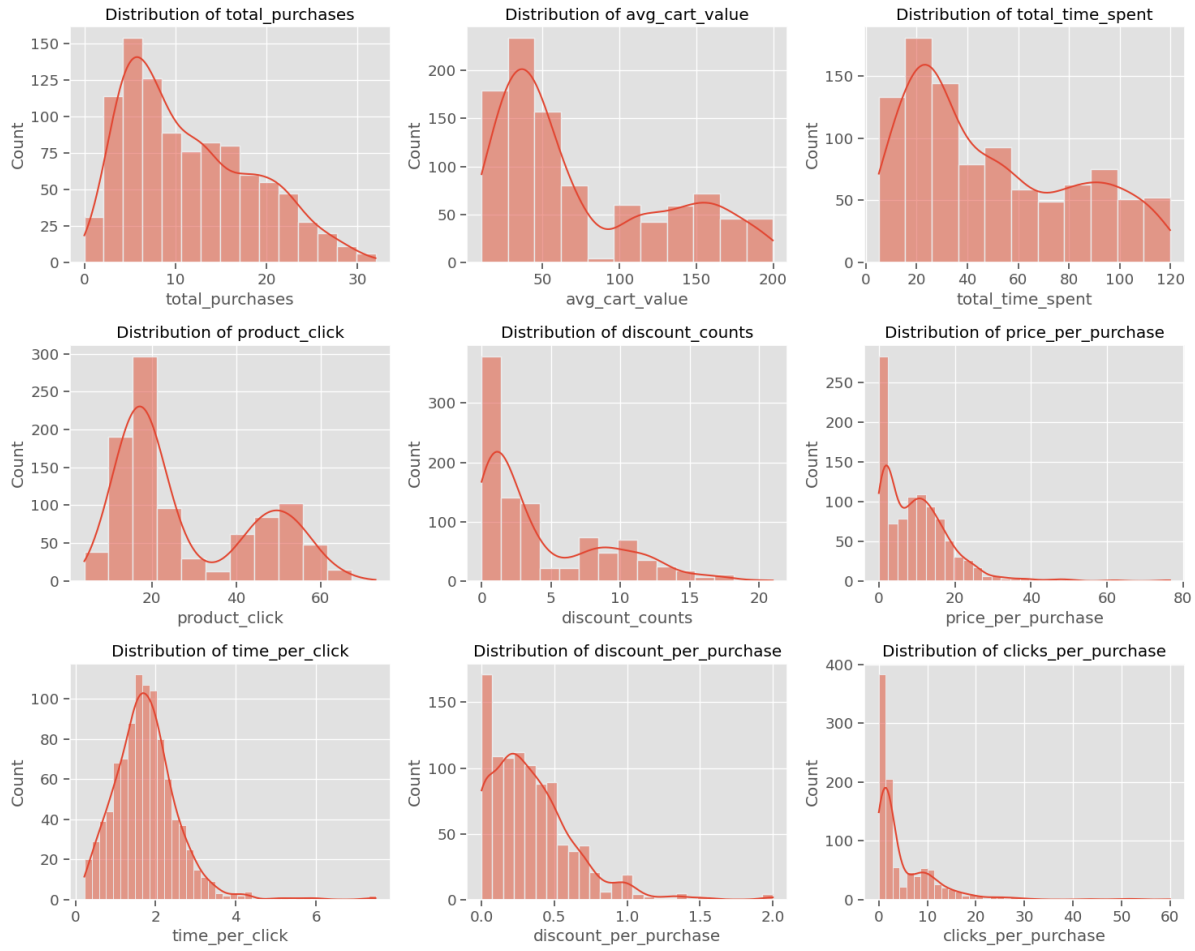
3. Exploratory Data Analysis

A comprehensive exploratory analysis was performed to understand the distribution of features and identify potential patterns.

3.1 Feature Distributions

The analysis of feature distributions revealed varying patterns across different customer metrics. Notably:

- **total_purchases** showed a right-skewed distribution with most customers making fewer than 15 purchases.
- **avg_cart_value** displayed a bimodal distribution, suggesting two distinct spending behaviors.
- **total_time_spent** revealed multiple behavioral patterns, with one group spending minimal time and another spending significantly more.

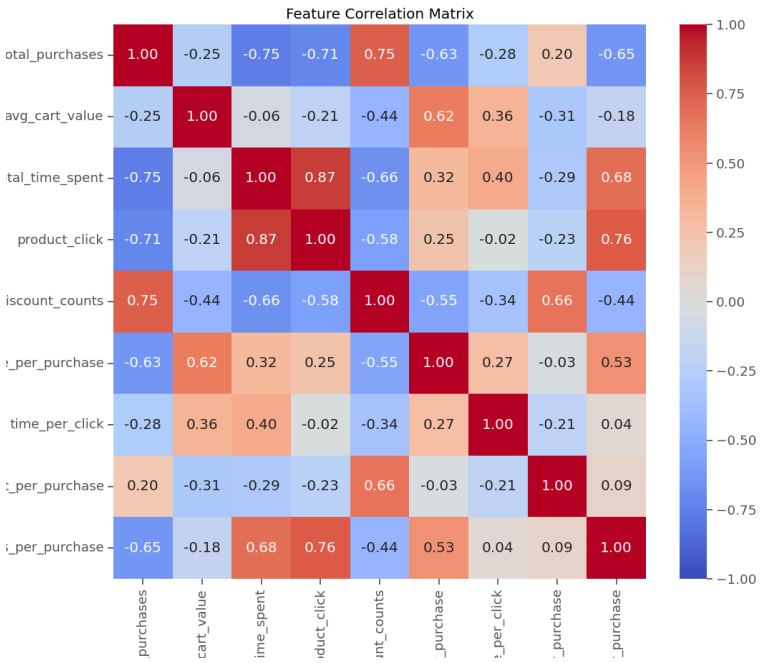


Feature Distributions

3.2 Correlation Analysis

Correlation analysis identified significant relationships between features,

- A moderate positive correlation between **total_purchases** and **discount_counts** (0.75), suggesting frequent buyers often leverage discounts
- A negative correlation between **avg_cart_value** and **discount_counts** (-0.44), indicating high-value shoppers use fewer discounts
- A strong positive correlation between **total_time_spent** and **product_click** (0.87), confirms browsing behavior consistency.



Correlation Analysis

4. Model Selection and Evaluation

4.1 Determining the Optimal Number of Clusters

Two methods were employed to determine the optimal number of clusters:

- **Elbow Method**: The inertia plot showed a clear "elbow" at k=3, indicating diminishing returns for additional clusters.
- **Silhouette Analysis**: Silhouette scores for different k values revealed that k=3 provided the best balance between cluster separation and cohesion.

4.2 Clustering Algorithms Comparison

Three clustering algorithms were evaluated using multiple metrics,

K-Means Clustering

- **Advantages**: Best performance across all metrics, computationally efficient, easily interpretable results.
- **Tradeoffs**: Assumes spherical clusters, sensitive to initialization and outliers

Hierarchical Clustering

- **Advantages**: Does not require pre-specification of the number of clusters, provides a dendrogram visualization

- **Tradeoffs**: Computationally intensive, lacks refinement mechanism after points are assigned

Gaussian Mixture Model

- **Advantages**: Provides probability of membership, allows for soft clustering, accommodates clusters of different shapes
- **Tradeoffs**: More complex model, potential for overfitting

4.3 Final Model Selection

K-Means was selected as the final model based on:

1. Superior performance across all evaluation metrics
2. Computational efficiency for large datasets
3. Clear interpretability of results for business stakeholders
4. Alignment with expected customer segments based on domain knowledge

4.4 Cluster Stability Analysis

To ensure the reliability of the clustering results, a stability analysis was performed by training the model on different subsets of data.

The analysis yielded a mean stability score of 0.621 with a standard deviation of 0.006, indicating consistent and reliable clustering results across different data subsets.

5. Cluster Analysis and Interpretation

5.1 Segment Profiles

The three identified clusters mapped well to the expected customer segments

Bargain Hunters

- High purchase frequency (avg. 19.7 purchases)
- Low average cart value (30.40)
- Moderate time spent browsing (17.5 minutes)
- Moderate product clicks (14.9 clicks)
- High discount usage (9.9 discounts used)

High Spenders

- Moderate purchase frequency (10.2 purchases)
- High average cart value (147.33)
- Moderate time spent browsing (40.3 minutes)

- Moderate product clicks (19.9 clicks)
- Low discount usage (2.0 discounts used)

Window Shoppers

- Low purchase frequency (4.9 purchases)
- Moderate average cart value (49.03)
- High time spent browsing (90.1 minutes)
- High product clicks (49.7 clicks)
- Very low discount usage (1.0 discounts used)

5.2 Segment Membership Distribution

The segment distribution in the customer base was

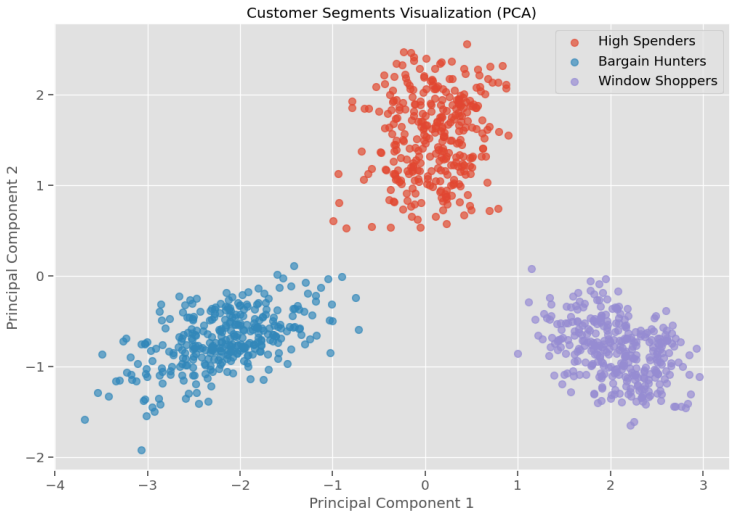
- Bargain Hunters: 326 - 42.5% of customers
- High Spenders: 325 - 31.2% of customers
- Window Shoppers: 328 - 26.3% of customers

5.3 Visualization of Segments

Multiple visualization techniques were employed to illustrate the segment separation.

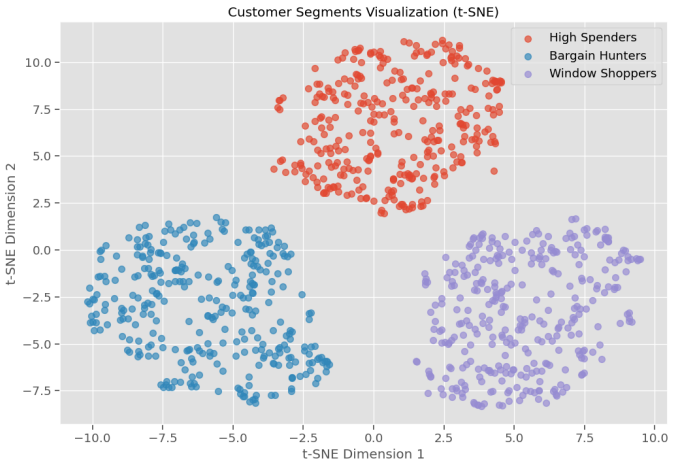
PCA 2D Visualization

Principal Component Analysis reduced the feature dimensions to two components, showing clear separation between the three clusters.



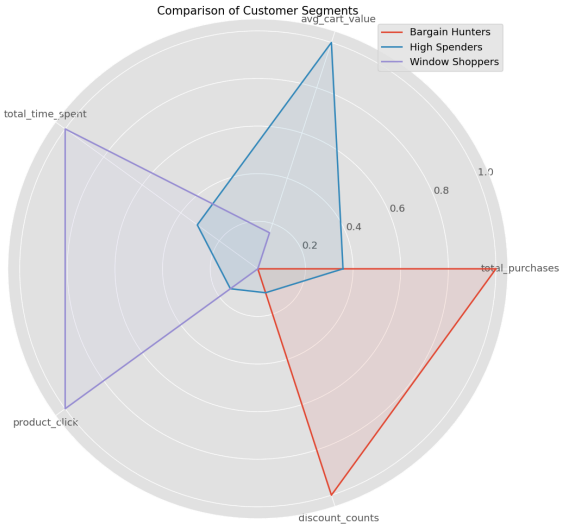
t-SNE Visualization

t-Distributed Stochastic Neighbor Embedding provided a non-linear dimensionality reduction that highlighted the distinct cluster boundaries.



Radar Chart

A radar chart comparing the standardized feature values across segments offered a clear visual comparison of segment characteristics.



6. Technical Implementation

6.1 Component Justification

Data Preprocessing

- **Technology**: Pandas and NumPy
- **Rationale**: These libraries provide efficient data manipulation capabilities and handle missing values effectively
- **Tradeoffs**: While powerful, they require careful handling of data types and memory management for large datasets

Feature Scaling

- **Technology**: StandardScaler from Scikit-learn
- **Rationale**: Ensures all features contribute equally to distance calculations in clustering algorithms
- **Tradeoffs**: Loss of interpretability of original values; requires consistent application to new data

Clustering Algorithm

- **Technology**: K-Means from Scikit-learn
- **Rationale**: Superior performance metrics, computational efficiency, and easily interpretable results
- **Tradeoffs**: Assumes spherical clusters and equal cluster sizes; sensitive to outliers

Dimensionality Reduction

- **Technology**: PCA and t-SNE from Scikit-learn
- **Rationale**: PCA provides efficient linear dimensionality reduction, while t-SNE preserves local structures
- **Tradeoffs**: PCA may miss non-linear patterns; t-SNE is computationally intensive and non-deterministic

6.2 Potential Challenges and Mitigation

Challenge 1: Outliers

- **Issue**: Extreme values in customer behavior can distort cluster formation.
- **Mitigation**: Robust scaling methods or selective removal of extreme outliers before clustering

Challenge 2: Evolving Customer Behavior

- **Issue**: Customer segments may change over time as preferences evolve
- **Mitigation**: Implement periodic model retraining and monitor segment stability metrics

7. Instructions for Running the Program

7.1 Environment Setup

1. Install required Python packages

   ```
   !pip install pandas numpy matplotlib seaborn scikit-learn scipy yellowbrick
   ```

2. Create output directory for visualizations

   ```
   import os
   output_dir = "output"
   os.makedirs(output_dir, exist_ok=True)
   ```

7.2 Data Preparation

Place the '*data2.csv*' file in the same directory as the script.

8. Conclusion and Future Work

The customer segmentation model successfully identified three distinct customer segments with unique behavioral patterns. The K-Means algorithm provided the most effective clustering solution, with clear separation between segments confirmed through multiple visualization techniques.

Future Enhancements

1. Incorporate additional customer data such as demographic information and product category preferences
2. Implement a more dynamic segmentation approach that can adapt to evolving customer behaviors
3. Develop a real-time segmentation system that can classify new customers as they interact with the platform
4. Explore deep learning techniques for more nuanced segment identification
5. Conduct A/B testing of marketing strategies tailored to each segment to validate the business impact

By leveraging these customer segments, the e-commerce platform can develop more targeted marketing strategies, enhance customer experiences, and ultimately drive improved business performance.

Drive Link for all the finding images: Drive Link

Google Colab Link: Colab Link

***