A

Assessment Report

on

“Customer Segmentation in E-commerce”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE - AIML

By

Ronak Goel (202401100400159)

Under the supervision of

“Abhishek Shukla”

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

Introduction

Customer segmentation plays a pivotal role in modern e-commerce and marketing strategies. By categorizing customers into distinct groups based on shared characteristics or behaviors, businesses can create targeted campaigns that are more relevant and personalized, which often leads to improved customer satisfaction and increased sales. Understanding the nuances of customer purchasing behavior allows companies to deliver tailored experiences, optimize their marketing efforts, and ultimately increase profitability. For example, businesses can offer exclusive promotions to high-value customers, create retention strategies for frequent buyers, or introduce new product offerings to customers with diverse preferences.

In the context of e-commerce, customer segmentation typically involves analyzing transactional data to group customers according to their spending habits, purchase frequency, and product variety. These insights allow businesses to optimize resource allocation, develop effective loyalty programs, and improve inventory management. Furthermore, customer segmentation can be instrumental in identifying emerging trends, helping businesses stay ahead of the curve by anticipating customer needs.

This report focuses on applying K-Means clustering, an unsupervised machine learning algorithm, to segment customers based on their purchasing behavior. The dataset we use includes key variables such as transaction quantity, unit price, and customer identification. By processing this data, we aim to uncover patterns that highlight distinct groups of customers, each with unique characteristics.

The analysis starts by cleaning and preparing the data, followed by feature engineering to create meaningful customer attributes such as total money spent, the frequency of purchases, and the variety of items purchased. These features are then standardized and fed into the K-Means algorithm, which groups customers into clusters. The results are visualized through Principal Component Analysis (PCA), reducing the data's dimensionality for easier interpretation, and a heatmap is generated to summarize the average behavior of each customer segment.

The ultimate goal of this analysis is not just to classify customers into arbitrary groups but to provide actionable insights that can drive strategic business decisions. By understanding which customer segments generate the most revenue, which segments are at risk of churn, and how diverse customer preferences affect product offerings, businesses can fine-tune their marketing, product, and customer service strategies to maximize customer engagement and lifetime value.

Methodology

1. Data Preprocessing

The first step in the methodology is data preprocessing, which involves:

- **Data Cleaning**: Rows with zero or negative values in the Quantity and UnitPrice columns were removed to ensure valid and accurate analysis. Any rows with erroneous values were filtered out.
- **Feature Engineering**: A new feature, TotalPrice, was created by multiplying Quantity by UnitPrice to reflect the total expenditure for each transaction.

2. Aggregation by Customer

The data was then aggregated at the CustomerID level to create customer-specific features:

- total_spent: The total amount of money spent by each customer.
- total_purchases: The number of unique invoices (transactions) made by each customer.
- unique_items: The number of distinct items purchased by each customer.

3. Data Scaling

Since the features vary in scale (e.g., total_spent can be in hundreds, while total_purchases might be in the range of tens), it is important to standardize the data. This was done using StandardScaler to ensure that all features have zero mean and unit variance, making the data suitable for clustering algorithms like K-Means.

4. K-Means Clustering

K-Means clustering was applied to segment customers into four clusters. K-Means is an iterative algorithm that assigns customers to one of the clusters based on their feature similarity. The number of clusters was selected arbitrarily (4), but different methods (e.g., the elbow method) could be used to determine the optimal number of clusters.

5. PCA for Visualization

Since the customer data has more than two features, we used **Principal Component Analysis (PCA)** to reduce the data’s dimensionality to two components, making it easier to visualize. This transformation allowed us to plot the clusters in a two-dimensional space, providing an intuitive understanding of the customer segments.

6. Heatmap for Cluster Analysis

To gain insights into the average behavior of customers within each cluster, a heatmap was created. This heatmap shows the average values of total_spent, total_purchases, and unique_items for each cluster, allowing for an interpretation of the distinct characteristics of each group.

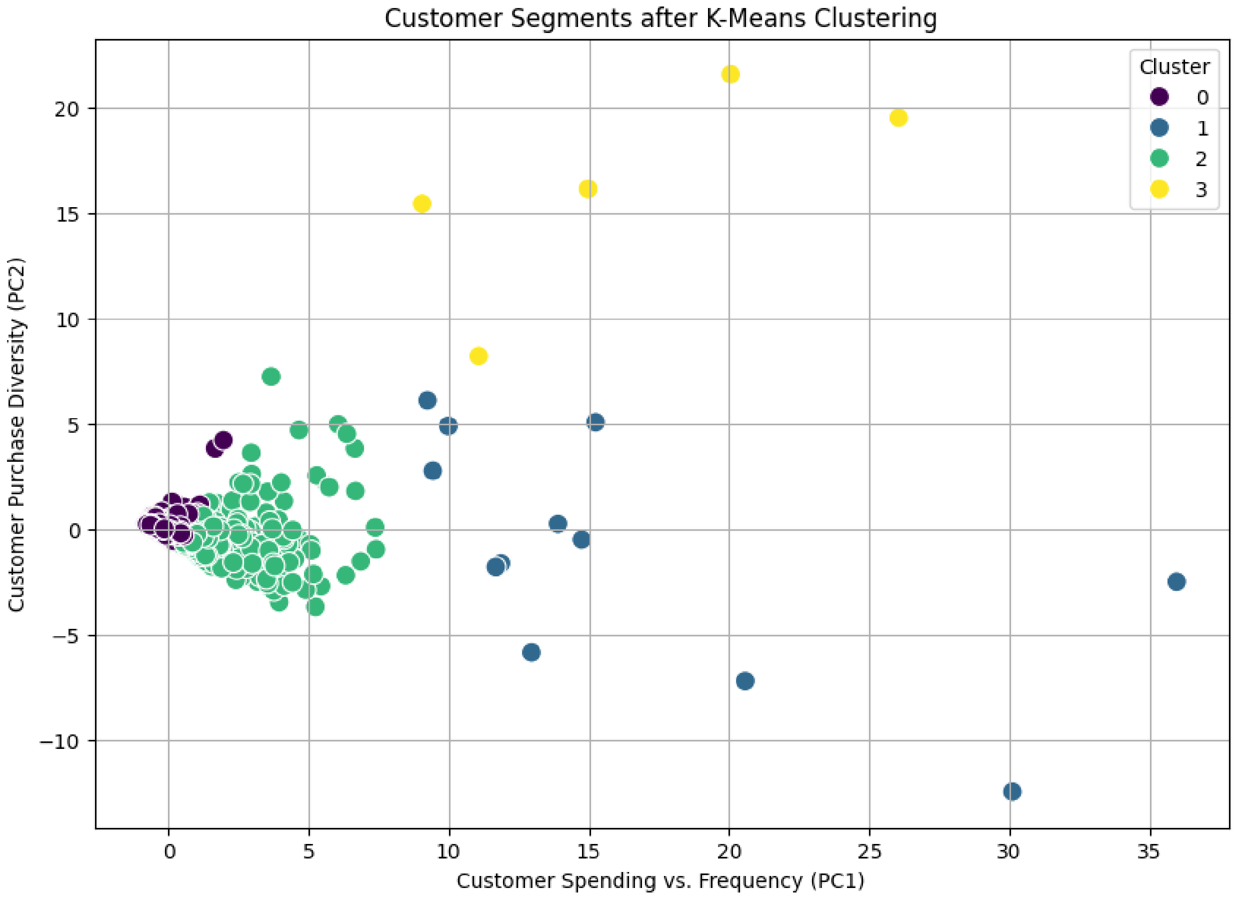
Code

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# Step 1: Load dataset
data = pd.read_csv('9. Customer Segmentation in E-commerce.csv')

# Step 2: Clean the data
# Filter out rows with non-positive quantities or unit prices
data = data[(data['Quantity'] > 0) & (data['UnitPrice'] > 0)].copy()

# Create 'TotalPrice' feature: Quantity * UnitPrice
data['TotalPrice'] = data['Quantity'] * data['UnitPrice']

# Step 3: Feature engineering
# Aggregate data by 'CustomerID' to get total spent, total purchases, and unique items
customer_data = data.groupby('CustomerID').agg(
    total_spent=('TotalPrice', 'sum'),  # Total amount spent by each customer
    total_purchases=('InvoiceNo', 'nunique'),  # Number of unique invoices
    unique_items=('StockCode', 'nunique')  # Number of unique items purchased
).reset_index()


# Step 4: Scale the data
# Standardize features to have zero mean and unit variance
scaler = StandardScaler()
scaled_data = scaler.fit_transform(customer_data[['total_spent', 'total_purchases', 'unique_items']])


# Step 5: Apply K-means clustering
# Use K-Means to segment customers into 4 clusters
kmeans = KMeans(n_clusters=4, random_state=42)
customer_data['Cluster'] = kmeans.fit_predict(scaled_data)


# Step 6: Visualize clusters with PCA
# Reduce data to 2 components using PCA for 2D visualization
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)
customer_data['PCA1'], customer_data['PCA2'] = pca_data[:, 0], pca_data[:, 1]


# Plot the PCA scatter plot to visualize customer clusters
plt.figure(figsize=(10, 7))
sns.scatterplot(data=customer_data, x='PCA1', y='PCA2', hue='Cluster', palette='viridis', s=100)
plt.title("Customer Segments after K-Means Clustering")
plt.xlabel("Customer Spending vs. Frequency (PC1)")
plt.ylabel("Customer Purchase Diversity (PC2)")
plt.legend(title="Cluster")
plt.grid(True)
plt.show()

# Step 7: Visualize average behavior per cluster with a heatmap
# Calculate the average feature values for each cluster
cluster_summary = customer_data.groupby('Cluster')[['total_spent', 'total_purchases', 'unique_items']].mean().round(1)

# Plot a heatmap of average customer behavior per cluster
plt.figure(figsize=(10, 6))
sns.heatmap(cluster_summary, annot=True, cmap='YlGnBu', fmt='.1f')
plt.title('Average Customer Behavior per Cluster')
plt.xlabel('Customer Behavior Features')
plt.ylabel('Cluster')
plt.show()
```

Output/Result

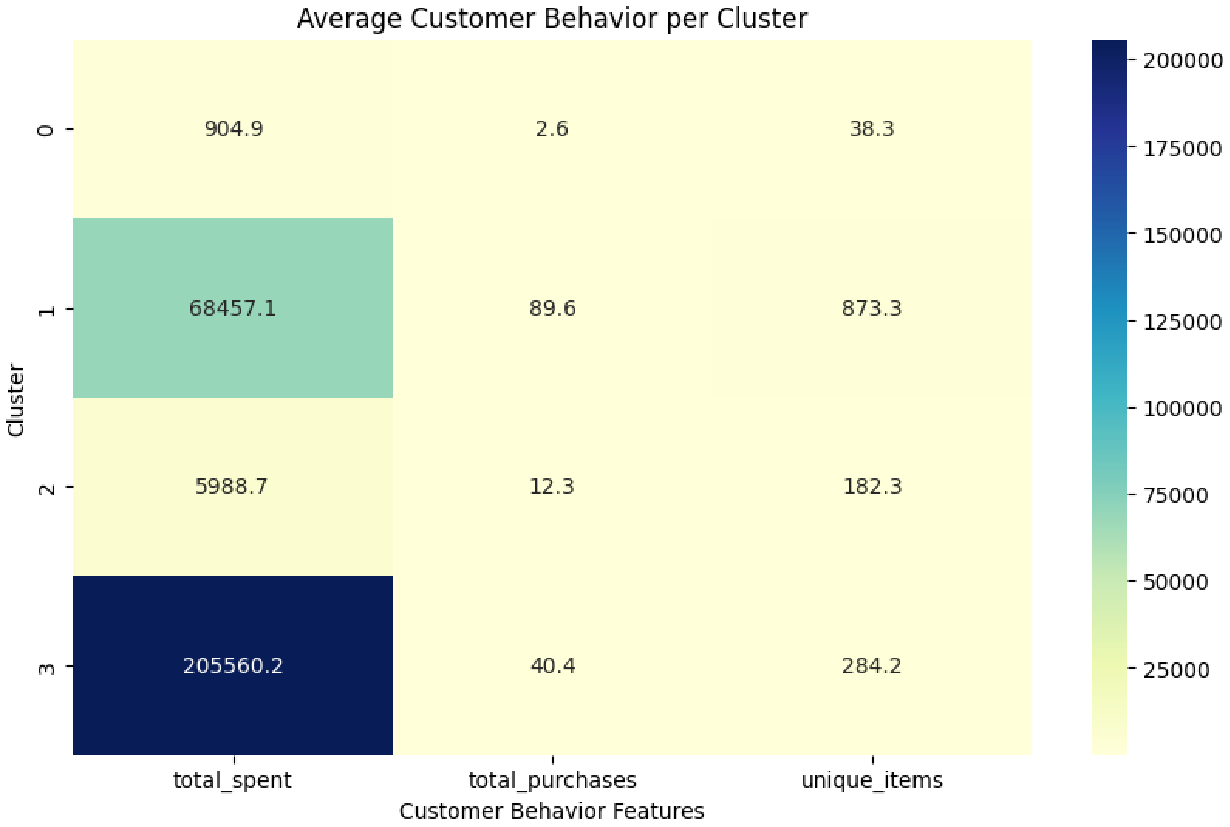
1. PCA Scatter Plot

The PCA scatter plot is used to visualize the distribution of customers across different clusters in a 2D space. The colors represent different clusters, which have been separated based on their purchasing behaviors. From this plot, we can observe distinct customer groups, indicating successful segmentation.



2. Heatmap of Average Customer Behavior per Cluster

The heatmap provides an overview of the average behavior of each cluster. Each cluster shows different average values for total_spent, total_purchases, and unique_items, which indicates the distinct characteristics of each customer group.



References

- **K-Means Clustering**: Scikit-learn Documentation - K-Means
- **PCA for Dimensionality Reduction**: Scikit-learn Documentation - PCA
- **StandardScaler**: Scikit-learn Documentation - StandardScaler
- **Matplotlib**: Matplotlib Documentation
- **Seaborn**: Seaborn Documentation
- **Dataset**: Online Retail Dataset from UCI
- **Tools used**: Python, Pandas, Scikit-learn, Seaborn, Matplotlib