A Report on

Customer Engagement Analytics With an e-commerce dataset

Project Overview

This project examines customer behavior and spending patterns using an ecommerce dataset of 3900 purchase records. The analysis identifies key customer segments, high-value products, and top-selling categories. It also explores subscription tendencies and purchasing frequency to highlight growth opportunities and support data-driven decisions for improving sales and product strategy.

Dataset Summary

- **Size:** 3900 rows and 18 columns
- **Feature Groups:**

  Customer Demographics & Profile:

  - Age, Gender, Location, Subscription Status

  Purchase Details & Product Attributes:

  - Item Purchased, Category, Season, Size, Color, Purchase Amount, Discount Applied, Promo Code Used, Shipping Type

  Behavioral & Engagement Metrics:

  - Previous Purchases, Frequency of Purchases, Review Rating

- **Missing Data:** 37 null values in Review Rating
- **Tools Used:** Python (Pandas), MySQL, Power BI

3. Exploratory Data Analysis (Python)

- **Data Preparation:** Loaded the dataset using Pandas.

```
[1]: import pandas as pd
```

```
[2]: df = pd.read_csv("customer_shopping_behavior.csv")
```

- **Data Loading & Initial Exploration:** Imported the dataset and inspected it with df.head(), df.info() and df.describe() to review sample rows, structure, non-null counts, and summary statistics.

```
[11]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Customer ID             3900 non-null   int64  
 1   Age                     3900 non-null   int64  
 2   Gender                  3900 non-null   object 
 3   Item Purchased          3900 non-null   object 
 4   Category                3900 non-null   object 
 5   Purchase Amount (USD)   3900 non-null   int64  
 6   Location                3900 non-null   object 
 7   Size                    3900 non-null   object 
 8   Color                   3900 non-null   object 
 9   Season                  3900 non-null   object 
 10  Review Rating           3863 non-null   float64
 11  Subscription Status     3900 non-null   object 
 12  Shipping Type           3900 non-null   object 
 13  Discount Applied        3900 non-null   object 
 14  Promo Code Used         3900 non-null   object 
 15  Previous Purchases      3900 non-null   int64  
 16  Payment Method          3900 non-null   object 
 17  Frequency of Purchases  3900 non-null   object 
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
```

[12]:
```
df.describe()
```

[12]:

	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3863.000000	3900.000000
mean	1950.500000	44.068462	59.764359	3.750065	25.351538
std	1125.977353	15.207589	23.685392	0.716983	14.447125
min	1.000000	18.000000	20.000000	2.500000	1.000000
25%	975.750000	31.000000	39.000000	3.100000	13.000000
50%	1950.500000	44.000000	60.000000	3.800000	25.000000
75%	2925.250000	57.000000	81.000000	4.400000	38.000000
max	3900.000000	70.000000	100.000000	5.000000	50.000000

- **Missing Value Treatment:** Imputed 37 missing review ratings using the median rating within each product category.

[9]:
```
#filling out the 37 nulls in review_rating with the category median
df["Review Rating"] = df.groupby("Category")["Review Rating"].transform(lambda x: x.fillna(x.median()))
```

- Converted column names to **snake_case** for clarity and to prevent issues from spaces or special characters.

[4]:
```
#renaming column
df.rename(columns ={"Purchase Amount (USD)": "purchase_amount"}, inplace=True)
```

[5]:
```
#identifying column names
df.columns
```

[5]:
```
Index(['Customer ID', 'Age', 'Gender', 'Item Purchased', 'Category',
       'purchase_amount', 'Location', 'Size', 'Color', 'Season',
       'Review Rating', 'Subscription Status', 'Shipping Type',
       'Discount Applied', 'Promo Code Used', 'Previous Purchases',
       'Payment Method', 'Frequency of Purchases'],
      dtype='object')
```

[6]:
```
#renaming columns
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ', '_')
```

- **Feature Engineering:**
    - Created ***age_group*** to categorize customers by age
    - Derived ***purchase_frequency_days*** to assess buying frequency

```
[8]: #creating a new column 'age_group'
     labels = ["Young Adult", "Adult", "Middle-aged", "Senior"]
     df["age_group"]= pd.qcut(df["age"], q=4, labels = labels)
```

```
[36]: #finding unique items of 'frequency_of_purchases' column
      df['frequency_of_purchases'].unique()
```

```
[36]: array(['Fortnightly', 'Weekly', 'Annually', 'Quarterly', 'Bi-Weekly',
             'Monthly', 'Every 3 Months'], dtype=object)
```

```
[40]: #creating a column name purchasing_frequency
      frequency_map = {"Weekly": 7,
                       "Fortnightly": 14,
                       "Bi-Weekly": 14,
                       "Monthly": 30,
                       "Every 3 Months": 90,
                       "Quarterly": 90,
                       "Annually": 365
                      }
      df["purchase_frequency_days"] = df["frequency_of_purchases"].map(frequency_map)
```

- **Data Consistency Review:** Identified redundancy between Discount Applied and Promo Code Used; removed Promo Code Used to maintain a clean, relevant dataset.

```
[14]: (df["discount_applied"] == df["promo_code_used"]).all()
```

```
[14]: np.True_
```

```
[15]: df.drop(columns=["promo_code_used"], inplace=True)
```

- **Database Integration:** Loaded the cleaned dataset into **MySQL** for structured analysis.

```
!pip install mysql-connector-python sqlalchemy pymysql
```

```
Requirement already satisfied: mysql-connector-python in c:\p
Requirement already satisfied: sqlalchemy in c:\purvad\anacon
Requirement already satisfied: pymysql in c:\purvad\anaconda3
Requirement already satisfied: greenlet!=0.4.17 in c:\purvad\
Requirement already satisfied: typing-extensions>=4.6.0 in c:
```

```python
from sqlalchemy import text

with engine.connect() as connection:
    result = connection.execute(text("SHOW DATABASES;"))
    for db in result:
        print(db[0])
```

```python
import pandas as pd
from sqlalchemy import create_engine, text
```

```python
engine = create_engine("mysql+pymysql://root:Password1@localhost:3306")
```

```python
with engine.connect() as connection:
    connection.execute(text("CREATE DATABASE IF NOT EXISTS customer;"))
```

```python
engine = create_engine("mysql+pymysql://root:Password1@localhost:3306/customer")
```

```python
df.to_sql('customer_shopping_behavior', con=engine, if_exists='replace', index=False)
```

```
3900
```

```python
with engine.connect() as connection:
    result = connection.execute(text("SHOW TABLES;"))
    for table in result:
        print(table[0])
```

```
customer_shopping_behavior
```

```python
query = "SELECT * FROM customer_shopping_behavior LIMIT 5;"
df_mysql = pd.read_sql(query, engine)
df_mysql
```

4. SQL-Based Business Analysis

Conducted targeted SQL queries to understand customer behavior, product performance, and transactional patterns, supporting data-driven decisions for sales optimization and customer strategy.

Business Problem

The company is experiencing inconsistent revenue growth and wants to understand what drives customer spending across different segments. Management needs clarity on which customer groups generate the most value, which products perform best, and how factors such as discounts, subscriptions, shipping preferences, and age groups influence revenue. The goal is to uncover spending patterns, identify high-value customer segments, optimize product listings, and strengthen retention strategies to improve profitability and guide future marketing and sales decisions.

1. Subscriber Spending Analysis

```sql
SELECT
    subscription_status,
    COUNT(customer_id) AS 'Number of Customers',
    ROUND(AVG(purchase_amount), 2) AS average_spend,
    ROUND(SUM(purchase_amount), 2) AS total_revenue
FROM
    customer_shopping_behavior
GROUP BY subscription_status
ORDER BY average_spend DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

subscription_status	Number of Customers	average_spend	total_revenue
No	2847	59.87	170436
Yes	1053	59.49	62645

Shows whether customers with a subscription spend more than those without, helping us understand if the subscription program drives higher revenue.

2. Gender-Based Revenue Comparison

```sql
SELECT
    gender, SUM(purchase_amount) AS revenue
FROM
    customer_shopping_behavior
GROUP BY gender;
```

Result Grid | Filter Rows: | Export: | W

gender	revenue
Male	157890
Female	75191

Compares how much men and women contribute to total revenue so the business can see which group spends more.

3. Age Group Revenue Insights

```sql
SELECT
    age_group,
    COUNT(customer_id) AS customers_count,
    ROUND(SUM(purchase_amount), 2) AS revenue_contribution
FROM
    customer_shopping_behavior
GROUP BY age_group
ORDER BY customers_count DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

age_group	customers_count	revenue_contribution
Young Adult	1028	62143
Middle-aged	986	59197
Senior	944	55763
Adult	942	55978

Breaks down revenue by age groups to reveal which age segments bring the most value to the business.

4. Top Rated Products Analysis

```sql
SELECT
    item_purchased, ROUND(AVG(review_rating), 2) AS top_products
FROM
    customer_shopping_behavior
GROUP BY item_purchased
ORDER BY AVG(review_rating) DESC
LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch r

item_purchased	top_products
Gloves	3.86
Sandals	3.84
Boots	3.82
Hat	3.8
Skirt	3.79

Identifies the five products customers rate the highest, helping the business highlight quality performers.

5. Category-Wise Top Products

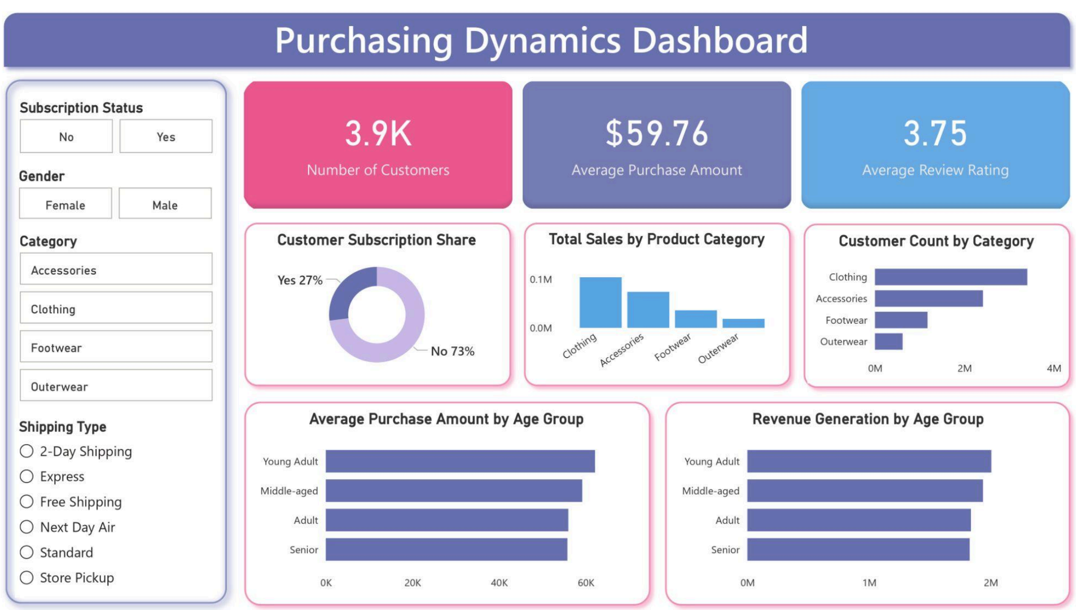
```sql
WITH product_count AS
(SELECT category,
    item_purchased,
    COUNT(customer_id) AS total_orders,
        ROW_NUMBER() OVER(PARTITION BY category ORDER BY COUNT(customer_id) DESC) AS item_rank
FROM customer_shopping_behavior
GROUP BY
    category, item_purchased
)
SELECT
    item_rank, category, item_purchased, total_orders
FROM
    product_count
WHERE
    item_rank <= 3;
```

item_rank	category	item_purchased	total_orders
1	Accessories	Jewelry	171
2	Accessories	Sunglasses	161
3	Accessories	Belt	161
1	Clothing	Blouse	171
2	Clothing	Pants	171
3	Clothing	Shirt	169
1	Footwear	Sandals	160
2	Footwear	Shoes	150
3	Footwear	Sneakers	145
1	Outerwear	Jacket	163
2	Outerwear	Coat	161

Finds the three most purchased items in every category to show what customers prefer the most in each product group.

Power BI Dashboard

Finally, developed the Purchasing Dynamics Dashboard in Power BI to visually communicate the analytical findings and support data-driven decision making.



Business Recommendations & Key Insights

1. Strengthen strategies for male customers, who drive the majority of revenue

Male customers contribute more than double the revenue of female customers (157,890 vs 75,191). The business should prioritize targeted promotions, personalized recommendations, and product placement for this segment to maximize returns while exploring why female engagement is significantly lower.

2. Optimize discount-driven campaigns to convert high-spending shoppers

A sizable 839 customers used discounts yet still spent above the average purchase amount, showing strong purchase intent even with incentives. The business should design segmented discount strategies for these high-value shoppers, such as tiered discounts or exclusive early access.

3. Promote top-rated and most-purchased products to boost sales

Products with the highest review ratings (3.79–3.86) and those leading category purchases should be highlighted in campaigns, cross-sell placements, and premium listings. These items already demonstrate strong customer approval and demand, making them ideal for feature spots and bundling.

4. Review the subscription model, as subscribers are not spending more

Despite expectations, subscribers show slightly lower average spend (59.49 vs 59.87) and contribute only ~27% of revenue. The business should evaluate subscription benefits, strengthen loyalty incentives, and consider redesigning the value proposition to encourage higher engagement and repeat purchases.

5. Focus on retaining loyal customers, who form the bulk of the customer base

The dataset shows 3324 loyal customers, far outweighing new and returning segments. Since loyal shoppers are the core revenue drivers, retention strategies such as personalized product alerts, early access sales, and member-only pricing should be prioritized over broad acquisition campaigns.

6. Tailor product listings and marketing by age group performance

All age groups contribute meaningful revenue, but young adults lead, generating the highest revenue (62,143). Marketing efforts, product assortments, and seasonal campaigns should align with young adult preferences while creating targeted nudges for senior and middle-aged buyers where revenue is slightly lower.

7. Highlight express shipping as a value option for slightly higher spenders

Customers choosing express shipping spend more on average (60.48 vs 58.46). The business should continue promoting express delivery as a premium option and consider bundling it with high-demand or top-rated products.