Customer Shopping Behavior Analysis

A leading retail company aims to better understand customer shopping behavior to improve sales performance, customer satisfaction, and long-term loyalty. Management observed changing purchasing patterns across demographics, product categories, and sales channels.

Business Problem Overview

A leading retail company aims to better understand customer shopping behavior to improve sales performance, customer satisfaction, and long-term loyalty. Management observed changing purchasing patterns across demographics, product categories, and sales channels. The core business question guiding this project is:

> "How can the company leverage consumer shopping data to identify trends, improve customer engagement, and optimize marketing and product strategies?"

Data Sources and Files Used

This report is developed using the following project artifacts:

Raw Dataset

customer_shopping_behavior.csv

Python Analysis

Code.ipynb

SQL Queries

Customer_behavior_SQL_queries.sql

Business Context

Business Problem Document

Power BI Dashboard

customer_behavior_dashboard.pbix and exported customer_behavior_dashboard.pdf

All analysis, visuals, and insights presented below are directly derived from these files.

Data Preparation & Exploratory Data Analysis (Python)

Python and pandas were used for data loading, inspection, and initial exploration. The following code snippets are taken directly from the Jupyter Notebook used in this project.

Importing Libraries

```
import numpy as np
import pandas as pd
```

Reading the Dataset

```
# reading the csv file
df = pd.read_csv('customer_shopping_behavior.csv')
```

This step loads the raw customer shopping behavior dataset into a pandas DataFrame for analysis.

Initial Data Inspection

```
# checking first few records that how data looks like
df.head()
```

The output confirms that the dataset contains customer demographics, purchase details, and behavioral attributes such as discount usage, shipping type, subscription status, and review ratings.

Dataset Dimensions

```
# check number of rows and columns of the data
df.shape
```

Result: (3900, 18)

The dataset consists of 3,900 records and 18 features, which aligns with the project requirements.

Data Quality and Transformation

Data Types and Structure

```
# check the data types of columns
df.info()
```

This output confirms:

- 18 total columns
- 3,900 records
- A mix of numerical (int64, float64) and categorical (object) variables
- Missing values present in the Review Rating column (only 3,863 non-null values)

Summary Statistics

```
# check the summary statistics
df.describe(include='all')
```

The summary statistics provide insights into:

- Distribution of numerical variables such as age, purchase amount, and previous purchases
- Frequency and uniqueness of categorical variables such as gender, category, and shipping type
- Central tendency and variability of review ratings

Missing Value Analysis

```
# check the missing values
df.isnull().sum()
```

The output shows that all columns contain complete data except **Review Rating**, which has **37 missing values**.

Missing Value Imputation

To avoid bias from global mean imputation, missing review ratings were filled using the **median rating within each product category**.

```
df['Review Rating'] = df.groupby('Category')['Review Rating'] \
    .transform(lambda x: x.fillna(x.median()))
```

After imputation, the dataset was revalidated.

```
df.isnull().sum()
```

The result confirms that no missing values remain in the dataset.

Feature Engineering and Data Standardization

Column Name Standardization

To ensure consistency and SQL compatibility, all column names were converted to **snake_case**.

```
# renaming all columns in snake case
df.columns = df.columns.str.lower().str.replace('_', '_')

df.rename(columns={'purchase_amount_(usd)': 'purchase_amount'}, inplace=True)

df.columns
```

This step standardizes column naming conventions and prepares the dataset for seamless database ingestion.

Feature Engineering – Age Group Creation

Customer ages were segmented into meaningful groups to support demographic-based analysis.

```
# Creating a column of age group
labels = ['Young Adult', 'Adult', 'Middle-aged', 'Senior']
df['age_group'] = pd.qcut(df['age'], q=4, labels=labels)

df[['age', 'age_group']].head()
```

This transformation enables age-based revenue and behavior analysis used later in SQL and Power BI.

Feature Engineering – Purchase Frequency (Days)

Purchase frequency values were originally stored as text. These were converted into numeric day equivalents to support quantitative analysis.

```
frequency_mapping = {
    'Fortnightly': 14,
    'Weekly': 7,
    'Monthly': 30,
    'Quarterly': 90,
    'Bi-Weekly': 14,
    'Annually': 365,
    'Every 3 Months': 90
}

df['purchase_frequency_days'] =
df['frequency_of_purchases'].map(frequency_mapping)

df[['purchase_frequency_days', 'frequency_of_purchases']].head(10)
```

This numeric representation enables more accurate behavioral comparisons.

Redundancy Check – Discount vs Promo Code

A validation step was performed to confirm whether `discount_applied` and `promo_code_used` represented different information.

```
df[['discount_applied', 'promo_code_used']].head()
(df['discount_applied'] == df['promo_code_used']).all()
```

Since both columns contained identical values across all records, `promo_code_used` was identified as redundant and removed.

```
df = df.drop('promo_code_used', axis=1)
df.columns
```

Database Integration (PostgreSQL)

The cleaned and transformed dataset was loaded into PostgreSQL using SQLAlchemy for further SQL-based analysis.

```
from sqlalchemy import create_engine

username = 'postgres'
password = '123456'
host = 'localhost'
port = '5432'
database = 'customer_behavior'

engine = create_engine(
    f"postgresql+psycopg2://{username}:{password}@{host}:{port}/{database}"
)

# Load the DataFrame into PostgreSQL
table_name = 'customer'
df.to_sql(table_name, engine, if_exists='replace', index=False)
```

This step completes the data preparation pipeline and enables structured SQL analysis on the finalized dataset.

Business Analysis Using SQL

Structured SQL queries were executed on the cleaned dataset to answer key business questions. The queries below are taken **directly and verbatim** from `Customer_behavior_SQL_queries.sql`.

Revenue Analysis by Gender

```
select gender, sum(purchase_amount) as revenue
from customer
group by gender;
```

High-Spending Discount Users

```
select customer_id, purchase_amount
from customer
where discount_applied = 'Yes'
and purchase_amount >= (select
avg(purchase_amount) from customer);
```

Top 5 Products by Average Review Rating

```
select item_purchased,
round(avg(review_rating::numeric), 2) as
average_product_rating
from customer
group by item_purchased
order by avg(review_rating) desc
limit 5;
```

Average Purchase Amount by Shipping Type

```
select shipping_type,
round(avg(purchase_amount), 2) as avg_spend
from customer
where shipping_type in ('Standard', 'Express')
group by shipping_type;
```

Subscribers vs Non-Subscribers Spending

```
select subscription_status,
count(*) as total_customer,
round(avg(purchase_amount), 2) as avg_spend,
round(sum(purchase_amount), 2) as total_revenue
from customer
group by subscription_status
order by total_revenue desc, avg_spend desc;
```

Products with Highest Discount Dependency

```
select item_purchased,
round(100 * sum(case when discount_applied =
'Yes' then 1 else 0 end)
/count(*), 2) as disc_rate
from customer
group by item_purchased
order by disc_rate desc
limit 5;
```

Customer Segmentation (New, Returning, Loyal)

```
with customer_type as (
select customer_id,
previous_purchases,
case
when previous_purchases = 1 then 'New'
when previous_purchases between 2 and 10 then
'Returning'
else 'Loyal'
end as customer_segment
from customer
)
select customer_segment, count(*) as
no_of_customers
from customer_type
group by customer_segment;
```

Top 3 Products per Category

```
with item_counts 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
    group by category, item_purchased
)
select item_rank, category, item_purchased,
total_orders
from item_counts
where item_rank <= 3;
```

Repeat Buyers and Subscription Status

```
select subscription_status, count(customer_id) as
repeat_buyers
from customer
where previous_purchases > 5
group by subscription_status;
```

Revenue Contribution by Age Group

```
select age_group, sum(purchase_amount) as
total_revenue
from customer
group by age_group
order by total_revenue desc;
```

Data Visualization & Dashboard (Power BI)

The Power BI dashboard was exported as a PDF and each page is included below as a visual figure. These dashboards provide an interactive and visual summary of customer behavior, revenue trends, and operational insights.

Power BI Dashboard – Page 1 (KPI & Customer Overview)

This dashboard highlights overall business performance, including total revenue, total orders, average order value, average review rating, revenue by gender, and customer segmentation (New, Returning, Loyal).

Power BI Dashboard – Page 2 (Revenue & Operational Insights)

This dashboard represents a **drill-through view** from the main dashboard, focusing specifically on **individual product categories**. It allows users to analyze category-wise performance in greater detail, including total sales, customer count, and average review ratings for the selected category. This drill-through functionality enables deeper exploration of product performance and supports more granular decision-making at the category level.

Dashboard Filters and Interactivity

The Power BI dashboard incorporates multiple **slicers** and **interactive filters** to allow dynamic exploration of customer behavior and business performance.

The following slicers were implemented:

- **Season Slicer:** Enables users to filter insights by shopping season (e.g., Spring, Summer, Winter), helping identify seasonal purchasing trends and demand fluctuations.
- **Location Slicer:** Allows analysis of customer behavior and revenue distribution across different geographic locations.
- **Category Slicer:** Enables users to focus on specific product categories and dynamically update all visuals accordingly.

These slicers are synchronized across all relevant visuals, ensuring that any selection instantly updates KPIs, charts, and tables. This interactivity allows stakeholders to drill down into specific customer segments, compare performance across dimensions, and uncover hidden patterns without modifying the underlying data model.

Overall, the use of slicers enhances usability, supports self-service analytics, and enables faster, data-driven decision-making.

Key Insights

Gender Revenue Distribution

Male customers contribute a significantly higher share of total revenue.

Customer Loyalty Impact

Loyal and returning customers generate the majority of sales.

Subscription Value

Subscribers show higher average spend and stronger retention.

Shipping Preferences

Express and free shipping options are associated with higher purchase values.

Age Group Performance

Middle-aged and young adult customers are the highest revenue-generating age groups.

Business Recommendations & Conclusion

Business Recommendations

Increase Subscription Adoption

Promote exclusive offers and benefits for subscribers.

Strengthen Loyalty Programs

Incentivize repeat purchases to convert returning customers into loyal customers.

Optimize Discount Strategy

Focus discounts on high-margin products and loyal customers.

Product Promotion

Highlight top-rated and best-selling products in marketing campaigns.

Targeted Marketing

Focus on high-revenue age groups and customers preferring express shipping.

Conclusion

This project demonstrates an end-to-end data analytics workflow using Python, SQL, and Power BI. By integrating data preparation, business analysis, and visualization, the company can better understand customer behavior and make informed strategic decisions that drive growth, engagement, and profitability.