Customer Shopping Behavior Analysis

Executive Summary

This analytical investigation examines customer shopping behavior through transactional data encompassing 3,900 purchases across diverse product categories. The project was designed to extract actionable insights into spending patterns, customer segmentation, product preferences, and subscription dynamics—ultimately informing strategic business decisions grounded in data-driven evidence.

---

1. Project Overview

The objective of this engagement was to construct a comprehensive analytical framework that transforms raw transactional data into strategic intelligence. By examining customer behavior patterns, we sought to identify opportunities for revenue optimization, customer retention, and targeted marketing interventions.

This project bridges the gap between operational data and strategic insight through disciplined data engineering, rigorous analysis, and visualization-driven storytelling.

---

2. Dataset Composition

Data Dimensions:

- **Sample Size:** 3,900 transactions
- **Feature Count:** 18 variables
- **Data Quality:** 37 missing values in review ratings (handled through systematic imputation)

Core Feature Categories:

- **Demographic Dimensions:** Age, Gender, Geographic Location, Subscription Status
- **Transaction Details:** Item Purchased, Product Category, Purchase Amount, Seasonal Context, Product Size, Color Preferences
- **Behavioral Indicators:** Discount Application, Promotional Code Adoption, Purchase History, Purchase Frequency, Review Ratings, Shipping Preference

---

3. Data Engineering & Preparation (Python)

The analytical process began with rigorous data preparation and quality assurance:

Data Assessment & Exploration

- Imported transactional dataset using pandas framework
- Conducted structural analysis via info() method to identify data types and formatting consistency

Data Quality Management

- Conducted comprehensive null-value analysis across all features
- Applied category-specific imputation strategy for 37 missing review ratings using median values within each product category

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

- Verified data integrity and consistency across all records

Feature Optimization

- Standardized column nomenclature to snake_case format for improved readability and reproducibility

	customer_id	age	gender	item_purchased	category	purchase_amount	location	size	color	season	review_rating	subscription_status	shipping_type
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping
5	6	46	Male	Sneakers	Footwear	20	Wyoming	M	White	Summer	2.9	Yes	Standard
6	7	63	Male	Shirt	Clothing	85	Montana	M	Gray	Fall	3.2	Yes	Free Shipping
7	8	27	Male	Shorts	Clothing	34	Louisiana	L	Charcoal	Winter	3.2	Yes	Free Shipping
8	9	26	Male	Coat	Outerwear	97	West Virginia	L	Silver	Summer	2.6	Yes	Express
9	10	57	Male	Handbag	Accessories	31	Missouri	M	Pink	Spring	4.8	Yes	2-Day Shipping

- Engineered age_group feature through quantile-based binning to facilitate cohort analysis

```
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ', '_')
df = df.rename(columns={'purchase_amount_(usd)' : 'purchase_amount'})
```

```
# creating a new column age_group to classify data further
# qcut splits the data into 'q=4' parts and assigns labels according to them into new column 'age_group'

labels = ['Young Adult', 'Adult', 'Middle-aged', 'Senior' ]
df['age_group'] = pd.qcut(df['age'], q=4, labels=labels)
```

- Derived purchase_frequency_days variable from temporal transaction data for behavioral insight

```
# create new column 'purchase_frequency_days'
# we create a dictionary to assign numeric values to texual data and map it in the new column

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

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

Redundancy Analysis & Refinement

- Evaluated relationship between discount_applied and promo_code_used features
- Identified feature redundancy and removed promo_code_used to optimize model parsimony
- Ensured all remaining features provided distinct analytical value

```
(df.discount_applied == df.promo_code_used).all()
```

```
np.True_
```

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

Data Infrastructure Integration

- Established connection between Python processing environment and PostgreSQL relational database
- Loaded cleaned and engineered DataFrame into production database schema
- Enabled structured SQL-based analysis on validated dataset

```
pip install psycopg2-binary sqlalchemy
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: psycopg2-binary in c:\users\akash\appdata\roaming\python\python313\site-packages (2.9.11)
Requirement already satisfied: sqlalchemy in c:\programdata\anaconda3\lib\site-packages (2.0.43)
Requirement already satisfied: greenlet>=1 in c:\programdata\anaconda3\lib\site-packages (from sqlalchemy) (3.2.4)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\programdata\anaconda3\lib\site-packages (from sqlalchemy) (4.15.0)
Note: you may need to restart the kernel to use updated packages.
```

```
from sqlalchemy import create_engine


# Step 1: Connect to PostgreSQL
# Replace placeholders with your actual details

username = "postgres"            # default user
password = "skyblurry"         # the password you set during installation
host = "localhost"                 # if running Locally
port = "5432"                      # default PostgreSQL port

database = "customer_behavior"         # the database you created in pgAdmin

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

#Step 2: Load Dataframe into PostgreSQL
table_name = "customer"
df.to_sql(table_name, engine, if_exists='replace', index=False)

print(f"Data successfully loaded into table '{table_name}' in database '{database}'.")
```

```
Data successfully loaded into table 'customer' in database 'customer_behavior'.
```

4. Business Analytics Via SQL

A series of structured SQL queries were executed to address key business intelligence questions:

1. Revenue Attribution by Gender

- Quantified total revenue generated across male and female customer segments
- Identified differential revenue contribution patterns by gender

	gender text	revenue numeric
1	Female	75191
2	Male	157890

2. High-Value Discount Adoption

- Isolated customers who applied discount codes yet maintained above-average purchase amounts
- Flagged price-sensitive customers with demonstrated spending capacity

	customer_id bigint	purchase_amount bigint
1	2	64
2	3	73
3	4	90
4	7	85
5	9	97
6	12	68
7	13	72
8	16	81
9	20	90

Total rows: 839 | Query complete 00:00:00.142

3. Product Quality Assessment

- Identified top five products by average review rating
- Established quality benchmarks for competitive positioning

	item_purchased text	Average Product Rating numeric
1	Gloves	3.86
2	Sandals	3.84
3	Boots	3.82
4	Hat	3.80
5	Skirt	3.78

4. Shipping Preference Economics

- Compared average transaction values between standard and express shipping options
- Evaluated willingness-to-pay for expedited delivery

	shipping_type text	round numeric
1	Standard	58.46
2	Express	60.48

5. Subscription Model Analysis

- Compared average spend profiles of subscribers versus non-subscribers
- Calculated revenue contribution differential across subscription cohorts

	subscription_status text	total_customers bigint	avg_spend numeric	total_revenue numeric
1	Yes	1053	59.49	62645.00
2	No	2847	59.87	170436.00

6. Discount Dependency Mapping

- Identified five products with highest percentage of discounted transactions
- Flagged products requiring margin-aware promotional strategy

	item_purchased text	discount_rate numeric
1	Hat	50.00
2	Sneakers	49.00
3	Coat	49.00
4	Sweater	48.00
5	Pants	47.00

7. Customer Lifecycle Segmentation

- Classified customers into New, Returning, and Loyal segments based on purchase history
- Established behavioral cohorts for targeted intervention strategies

	customer_segment text	Number of Customers bigint
1	Loyal	3116
2	New	83
3	Returning	701

8. Category-Level Product Performance

- Ranked top three products within each category by transaction frequency
- Identified portfolio concentration and category strength

	item_rank bigint	category text	item_purchased text	total_orders bigint
1	1	Accessori…	Jewelry	171
2	2	Accessori…	Sunglasses	161
3	3	Accessori…	Belt	161
4	1	Clothing	Blouse	171
5	2	Clothing	Pants	171
6	3	Clothing	Shirt	169
7	1	Footwear	Sandals	160
8	2	Footwear	Shoes	150
9	3	Footwear	Sneakers	145
10	1	Outerwear	Jacket	163
11	2	Outerwear	Coat	161

9. Repeat Purchase & Subscription Correlation

- Analyzed relationship between purchase frequency (>5 transactions) and subscription likelihood
- Quantified customer lifetime value predictors

	subscription_status text	repeat_buyers bigint
1	No	2518
2	Yes	958

10. Revenue Contribution by Age Cohort

- Calculated total and average revenue attribution across age groups
- Identified high-value demographic segments for targeted acquisition

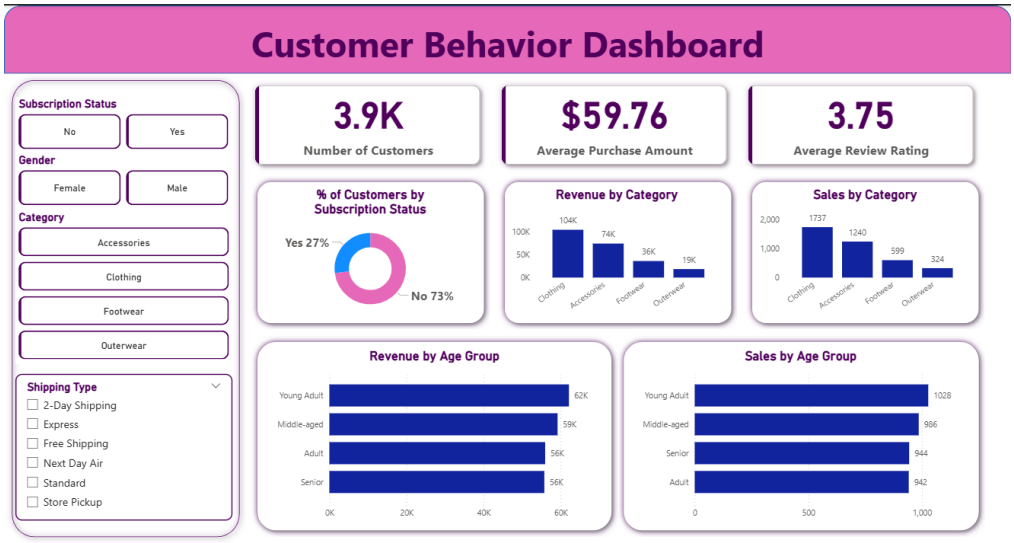
	age_group text	total_revenue numeric
1	Young Adult	62143
2	Middle-aged	59197
3	Adult	55978
4	Senior	55763

---

5. Visualization & Business Intelligence

An interactive Power BI dashboard was developed to present analytical findings in accessible, actionable format. The dashboard enables stakeholders to:

- Monitor key performance indicators in real-time
- Explore data through multidimensional filtering and drill-down capabilities
- Identify trends and patterns across customer, product, and temporal dimensions
- Make informed decisions based on visual evidence



6. Strategic Recommendations

- **Subscription Growth Initiative**
  Develop and promote subscription-exclusive benefits packages. The data indicates differential spending between subscriber and non-subscriber cohorts. Strategic incentives can migrate price-sensitive customers into higher-lifetime-value subscription models.
- **Customer Loyalty Architecture**
  Implement tiered loyalty programs designed to elevate returning customers into the "Loyal" segment. Behavioral economics research supports the effectiveness of recognition and reward structures in driving repeat purchase behavior.
- **Promotional Efficiency Optimization**
  Conduct systematic review of discount policies to optimize the margin-growth tradeoff. The data reveals products with excessive discount dependency; strategic repricing or value-bundling can reduce margin erosion while maintaining competitiveness.
- **Product Portfolio Positioning**
  Amplify marketing and merchandising emphasis on products demonstrating both high review ratings and strong sales velocity. Consumer psychology literature indicates that quality signals drive conversion and reduce price sensitivity.
- **Demographic-Targeted Marketing**
  Concentrate marketing investment on high-revenue age cohorts and express-shipping user segments. Behavioral segmentation allows efficient capital allocation and improved return-on-marketing-spend metrics.

Conclusion

This analytical engagement demonstrates the transformative potential of disciplined data engineering, rigorous statistical analysis, and evidence-based storytelling. By converting transactional complexity into strategic insight, organizations can move beyond intuition-driven decision making toward a data-informed operational model.

The recommendations outlined above provide a foundation for incremental improvements in customer acquisition efficiency, lifetime value realization, and margin optimization.

Report Prepared: January 2026
Methodology: Python Data Engineering | PostgreSQL Analysis | Power BI Visualization
Data Sample: 3,900 Transactions | 18 Features | Multi-Category Coverage