Customer Shopping Behavior Analysis for Retail Optimization

Project Overview

This project analyzes a retail company's consumer shopping data to uncover patterns in customer behavior, trends across demographics, product categories, and shipping channels. The insights aim to improve sales, increase customer satisfaction, and enhance long-term loyalty.

Business Problem

The company wants to understand which factors influence customer decisions and repeat purchases, including discounts, reviews, seasons, payment preferences, and shipping channels.

Key question:

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

In [1]:

```
# Step 1: Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from scipy import stats
warnings.filterwarnings('ignore')

# Set display options
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 100)

# Set visualization style
sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (12, 6)
pd.set_option('display.max_columns', None)

#Load Dataset
df = pd.read_csv('customer_shopping_behavior.csv')
```

In [2]:

```
#Intial Exploration
print(df.info())
print(df.describe())
```

```
<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
None
       Customer ID          Age  Purchase Amount (USD)  Review Rating  \
count  3900.000000  3900.000000            3900.000000    3863.000000   
mean   1950.500000    44.068462              59.764359       3.750065   
std    1125.977353    15.207589              23.685392       0.716983   
min       1.000000    18.000000              20.000000       2.500000   
25%     975.750000    31.000000              39.000000       3.100000   
50%    1950.500000    44.000000              60.000000       3.800000   
75%    2925.250000    57.000000              81.000000       4.400000   
max    3900.000000    70.000000             100.000000       5.000000   

       Previous Purchases  
count         3900.000000  
mean            25.351538  
std             14.447125  
min              1.000000  
25%             13.000000  
50%             25.000000  
75%             38.000000  
max             50.000000  
```

In [3]:

```
print(df.head())
```

```
   Customer ID  Age Gender Item Purchased  Category  Purchase Amount (USD)  \
0            1   55   Male         Blouse  Clothing                     53   
1            2   19   Male        Sweater  Clothing                     64   
2            3   50   Male          Jeans  Clothing                     73   
3            4   21   Male        Sandals  Footwear                     90   
4            5   45   Male         Blouse  Clothing                     49   

        Location Size      Color  Season  Review Rating Subscription Status  \
0       Kentucky    L       Gray  Winter            3.1                 Yes   
1          Maine    L     Maroon  Winter            3.1                 Yes   
2  Massachusetts    S     Maroon  Spring            3.1                 Yes   
3   Rhode Island    M     Maroon  Spring            3.5                 Yes   
4         Oregon    M  Turquoise  Spring            2.7                 Yes   

   Shipping Type Discount Applied Promo Code Used  Previous Purchases  \
0        Express              Yes             Yes                  14   
1        Express              Yes             Yes                   2   
2  Free Shipping              Yes             Yes                  23   
3   Next Day Air              Yes             Yes                  49   
4  Free Shipping              Yes             Yes                  31   

  Payment Method Frequency of Purchases  
0          Venmo            Fortnightly  
1           Cash            Fortnightly  
2    Credit Card                 Weekly  
3         PayPal                 Weekly  
4         PayPal               Annually  
```

In [4]:
```python
# Renaming columns according to snake casing for better readability and document
df.columns = df.columns.str.replace(' ','_')
```

In [5]:
```python
#Check columns
print(df.columns)
```

```
Index(['Customer_ID', 'Age', 'Gender', 'Item_Purchased', 'Category',
       'Purchase_Amount_(USD)', 'Location', 'Size', 'Color', 'Season',
       'Review_Rating', 'Subscription_Status', 'Shipping_Type',
       'Discount_Applied', 'Promo_Code_Used', 'Previous_Purchases',
       'Payment_Method', 'Frequency_of_Purchases'],
      dtype='object')
```

In [6]:
```python
df = df.rename(columns={'purchase_amount_(usd)':'purchase_amount'})
```

In [7]:
```python
df = df.rename(columns={'purchase_amount':'Purchase_Amount'})
```

In [8]:
```python
df.columns
```

Out[8]:
```
Index(['Customer_ID', 'Age', 'Gender', 'Item_Purchased', 'Category',
       'Purchase_Amount_(USD)', 'Location', 'Size', 'Color', 'Season',
       'Review_Rating', 'Subscription_Status', 'Shipping_Type',
       'Discount_Applied', 'Promo_Code_Used', 'Previous_Purchases',
       'Payment_Method', 'Frequency_of_Purchases'],
      dtype='object')
```

In [12]:
```python
#Step 2: Data Cleaning & Preprocessing

#Data Quality Check
print("\nData Quality Check")
print("="*70)
```

```
#Check missing values
print("\nMissing Values")
missing = df.isnull().sum()
missing_pct = (missing / len(df) * 100).round(2)
missing_df = pd.DataFrame ({'Missing Count': missing, 'Percentage': missing_pct}
print(missing_df[missing_df['Missing Count'] >0])

#Handle missing values Review Rating col (fill with median)
if df['Review_Rating'].isnull().sum() > 0:
    median_rating = df['Review_Rating'].median()
    df['Review_Rating'].fillna(median_rating, inplace=True)
    print(f"\nFilled {missing['Review_Rating']} missing Review Ratings with medi

#Check duplicates
duplicates = df.duplicated().sum()
print(f"\nDuplicate Rows: {duplicates}")
if duplicates > 0:
    df.drop_duplicates(inplace=True)
    print(f"Removed{duplicates} duplicate rows")

#Clean column names (remove spaces)
df.rename(columns={'Purchase_Amount_(USD)':'Purchase_Amount'}, inplace=True)
```

```
Data Quality Check
======================================================================

Missing Values
               Missing Count  Percentage
Review_Rating             37        0.95

Filled 37 missing Review Ratings with median: 3.8

Duplicate Rows: 0
```

In [13]:
```
df.columns
```

Out[13]:
```
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')
```

In [18]:
```
# ============================================================================
# Step 3: Feature Engineering (process of modifying features)
# ============================================================================

print("\n" + "="*70)
print("FEATURE ENGINEERING")
print("="*70)

# 1. Customer Lifetime Value metrics
customer_metrics = df.groupby('Customer_ID').agg({
    'Purchase_Amount': ['sum', 'mean', 'count'],
    'Review_Rating': 'mean'
}).round(2)

customer_metrics.columns = ['Total_Spending', 'Avg_Purchase', 'Purchase_Count',
customer_metrics.reset_index(inplace=True)
```

```python
# Remove old columns if they exist to prevent _x/_y duplicates
cols_to_drop = ['Total_Spending', 'Avg_Purchase', 'Purchase_Count', 'Avg_Rating'
df = df.drop(columns=[col for col in cols_to_drop if col in df.columns])

# Merge only the necessary new columns
df = df.merge(customer_metrics, on='Customer_ID', how='left')

# 2. Customer Segmentation based on spending
spending_quartiles = df['Total_Spending'].quantile([0.25, 0.5, 0.75])
df['Customer_Segment'] = pd.cut(df['Total_Spending'],
                                bins=[0, spending_quartiles[0.25], spending_quar
                                      spending_quartiles[0.75], df['Total_Spendi
                                labels=['Low Value', 'Medium Value', 'High Value
                                include_lowest=True)

# 3. Age Groups
df['Age_Group'] = pd.cut(df['Age'], bins=[0, 25, 35, 45, 55, 100],
                         labels=['18-25', '26-35', '36-45', '46-55', '55+'],
                         include_lowest=True)

# 4. Discount Effectiveness
df['Discount_Flag'] = df['Discount_Applied'].apply(lambda x: 1 if str(x).strip()
df['Effective_Price'] = df['Purchase_Amount']

# 5. High Spender Flag
median_purchase = df['Purchase_Amount'].median()
df['High_Spender'] = (df['Purchase_Amount'] > median_purchase).astype(int)

# 6. Purchase Frequency in days
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_mappi

# Optional: rename columns to remove _x if any appear (extra safeguard)
df = df.rename(columns=lambda x: x.rstrip('_x'))

print("\nNew Features Created:")
print("- Total_Spending, Avg_Purchase, Purchase_Count, Avg_Rating (per customer)
print("- Customer_Segment (Low/Medium/High/Premium Value)")
print("- Age_Group (18-25, 26-35, etc.)")
print("- Discount_Flag (1 = Discount Applied, 0 = No Discount)")
print("- Effective_Price (kept same or adjusted if discount amount known)")
print("- High_Spender flag")
print("- Purchase_Frequency_Days")
```

```
======================================================================
FEATURE ENGINEERING
======================================================================

New Features Created:
- Total_Spending, Avg_Purchase, Purchase_Count, Avg_Rating (per customer)
- Customer_Segment (Low/Medium/High/Premium Value)
- Age_Group (18-25, 26-35, etc.)
- Discount_Flag (1 = Discount Applied, 0 = No Discount)
- Effective_Price (kept same or adjusted if discount known)
- High_Spender flag
- Purchase_Frequency_Days
```

In [19]:

```python
#Step 4: Exploratory Data Analysis (EDA)

print("\n" + "="*70)
print("DESCRIPTIVE STATISTICS")
print("="*70)

# Numerical summary
print("\nNumerical Variables Summary:")
print(df[['Age', 'Purchase_Amount', 'Review_Rating', 'Discount_Applied',
          'Previous_Purchases']].describe().round(2))

# Categorical summary
print("\nCategorical Variables - Value Counts:")
categorical_cols = ['Gender', 'Category', 'Season', 'Subscription_Status',
                    'Payment_Method', 'Frequency_of_Purchases']

for col in categorical_cols:
    print(f"\n{col}:")
    print(df[col].value_counts().head())
```

```
======================================================================
DESCRIPTIVE STATISTICS
======================================================================

Numerical Variables Summary:
           Age  Purchase_Amount  Review_Rating  Previous_Purchases
count  3900.00          3900.00        3900.00             3900.00
mean     44.07            59.76           3.75               25.35
std      15.21            23.69           0.71               14.45
min      18.00            20.00           2.50                1.00
25%      31.00            39.00           3.10               13.00
50%      44.00            60.00           3.80               25.00
75%      57.00            81.00           4.40               38.00
max      70.00           100.00           5.00               50.00

Categorical Variables - Value Counts:

Gender:
Gender
Male      2652
Female    1248
Name: count, dtype: int64

Category:
Category
Clothing       1737
Accessories    1240
Footwear        599
Outerwear       324
Name: count, dtype: int64

Season:
Season
Spring    999
Fall      975
Winter    971
Summer    955
Name: count, dtype: int64

Subscription_Status:
Subscription_Status
No     2847
Yes    1053
Name: count, dtype: int64

Payment_Method:
Payment_Method
PayPal         677
Credit Card    671
Cash           670
Debit Card     636
Venmo          634
Name: count, dtype: int64

Frequency_of_Purchases:
Frequency_of_Purchases
Every 3 Months    584
Annually          572
Quarterly         563
Monthly           553
```

```
Bi-Weekly         547
Name: count, dtype: int64
```

In [20]:

```
# Save processed dataset
df.to_csv('customer_shopping_analysis_latest.csv', index=False)
print("\n✓ Processed dataset saved: customer_shopping_analysis_complete.csv")
```

```
✓ Processed dataset saved: customer_shopping_analysis_complete.csv
```