Data Analysis Report: Uncovering insights into Customer Shopping Behavior

1. Executive Summary

This comprehensive analysis examines transactional data from 3,900 customers across various product categories to identify key drivers of revenue and consumer shopping patterns. The analysis highlights that while the Clothing category dominates sales, there is a significant opportunity to convert a large segment of repeat buyers into subscribers. By leveraging targeted marketing for the high-spending Young Adult demographic and optimizing loyalty programs, the business can drive sustainable growth.

2. Project Overview & Data Summary

The goal of this project is to uncover actionable insights into spending patterns, customer segments, and product preferences.

- **Dataset Scope**: 3,900 records with 18 features, including demographics, purchase details, and behavioral metrics.
- **Data Quality**: Missing values in the **Review Rating** column were addressed using category-based median imputation to ensure analysis integrity.
- **Technical Stack**: Data cleaning and EDA were performed in **Python (Pandas)**, structured queries in **PostgreSQL**, and final visualizations in **Power BI**.

3. Exploratory Data Analysis Using Python

- **Data Loading**: Imported dataset using pandas.
- **Initial Exploration**: Used df.info() to check structure and .describe() for summary statistics.

```
[2]: 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
```

```
[3]: df.describe(include="all")
```

[3]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900
unique	NaN	NaN	2	25	4	NaN	50	4	25	4
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN

	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
count	3863.000000	3900	3900	3900	3900	3900.000000	3900	3900
unique	NaN	2	6	2	2	NaN	6	7
top	NaN	No	Free Shipping	No	No	NaN	PayPal	Every 3 Months
freq	NaN	2847	675	2223	2223	NaN	677	584
mean	3.750065	NaN	NaN	NaN	NaN	25.351538	NaN	NaN
std	0.716983	NaN	NaN	NaN	NaN	14.447125	NaN	NaN
min	2.500000	NaN	NaN	NaN	NaN	1.000000	NaN	NaN
25%	3.100000	NaN	NaN	NaN	NaN	13.000000	NaN	NaN
50%	3.800000	NaN	NaN	NaN	NaN	25.000000	NaN	NaN
75%	4.400000	NaN	NaN	NaN	NaN	38.000000	NaN	NaN
max	5.000000	NaN	NaN	NaN	NaN	50.000000	NaN	NaN

- **Missing Data Handling**: Checked for null values and imputed missing values in the Review Rating column using the median rating of each product category.

```
[4]: df.isnull().sum()

[4]: Customer ID               0
     Age                       0
     Gender                    0
     Item Purchased            0
     Category                  0
     Purchase Amount (USD)     0
     Location                  0
     Size                      0
     Color                     0
     Season                    0
     Review Rating            37
     Subscription Status       0
     Shipping Type             0
     Discount Applied          0
     Promo Code Used           0
     Previous Purchases        0
     Payment Method            0
     Frequency of Purchases    0
     dtype: int64
```

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

[6]: df.isnull().sum()

[6]: Customer ID               0
     Age                       0
     Gender                    0
     Item Purchased            0
     Category                  0
     Purchase Amount (USD)     0
     Location                  0
     Size                      0
     Color                     0
     Season                    0
     Review Rating             0
     Subscription Status       0
     Shipping Type             0
     Discount Applied          0
     Promo Code Used           0
     Previous Purchases        0
     Payment Method            0
     Frequency of Purchases    0
     dtype: int64
```

- **Column Standardization**: Renamed columns to snake case for better readability and documentation.

=

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

8]:
```
df.columns
```

8]:
```
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')
```

- **Feature Engineering**:
    - Created age_group column by binning customer ages.
    - Created purchase_frequency_days column from purchase data.

```
labels = ["Young Adult", "Adult", "Middle-aged", "Senior"]
df["age_group"] = pd.qcut(df["age"], q=4, labels=labels)
# qcut = QUantile cut, it splits data into equal-sized groups
```

[10]:
```
df[["age", "age_group"]].head(10)
```

[10]:

	age	age_group
0	55	Middle-aged
1	19	Young Adult
2	50	Middle-aged
3	21	Young Adult
4	45	Middle-aged
5	46	Middle-aged
6	63	Senior
7	27	Young Adult
8	26	Young Adult
9	57	Middle-aged

[11]:
```
# create column purchase_frequency_days
frequency_mapping = {
    "Fortnightly": 14,
    "Weekly": 7,
    "Monthly": 30,
    "Quarterly": 90,
    "Bi-Weegly": 14,
    "Annually": 365,
    "Every 3 Months": 90
}
df["purchase_frequency_days"] = df["frequency_of_purchases"].map(frequency_mapping)
```

[12]:
```
df[["purchase_frequency_days", "frequency_of_purchases"]].head(10)
```

[12]:

	purchase_frequency_days	frequency_of_purchases
0	14.0	Fortnightly
1	14.0	Fortnightly
2	7.0	Weekly
3	7.0	Weekly
4	365.0	Annually
5	7.0	Weekly
6	90.0	Quarterly
7	7.0	Weekly
8	365.0	Annually
9	90.0	Quarterly

- **Data consistency Check**: Verified if discount_applied and promo_code_used were redundant; dropped promo_code_used.

```
[15]: df = df.drop("promo_code_used", axis=1)

[16]: df.columns

[16]: Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
             'purchase_amount', 'location', 'size', 'color', 'season',
             'review_rating', 'subscription_status', 'shipping_type',
             'discount_applied', 'previous_purchases', 'payment_method',
             'frequency_of_purchases', 'age_group', 'purchase_frequency_days'],
            dtype='object')
```

- **Database Integration**: Connexted Python script to PostgreSQL and loaded the cleaned DataFrame into the database for SQL analysis.

4. Dara Analysis using SQL

We performed structured analysis in PostgreSQL to answer key business questions:

1. **Revenue by Gender** – Compared total revenue generated by male vs. female customers.

	gender text	revenue numeric
1	Female	75191
2	Male	157890

2. **High-Spending Discount Users** – Identified customers who used discounts but still spent above the average purchase amount.

Data Output Messages Notifications

	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

Total rows: 839 Query complete 00:00:0

3. **Top 5 Product by rating** – Found products with the highest average review ratings.

Data Output Messages Notifications

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

Total rows: 5 Query complete 00:00:00.1

4. **Shipping Type Comparison** – Compared average purchase amounts between Standard and Express shipping.

Data Output Messages Notificati

	round numeric	shipping_type text
1	58.46	Standard
2	60.48	Express

5. **Subscribers vs. Non-Subscribers** – Compared average spend and total revenue across subscription status.

Data Output Messages Notifications

	subscription_status text	total_customers bigint	average_spend numeric	revenue numeric
1	Yes	1053	59.49	62645.00
2	No	2847	59.87	170436.00

6. **Discount-Dependent Products** – Identified 5 products with the highest percentage of discounted purchases.

Data Output Messages Notifications

	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 Segmentation** – Classified customers into New, Returning and Loyal segments based on purchase history.

Data Output Messages Notifications

SQL

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

8. **Top 3 Products per Category** – Listed the most purchased products within each category.

Data Output Messages Notifications

SQL

	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 Buyers & Subscriptions** – Checked whether customers with >5 purchases are more likely to subscribe.

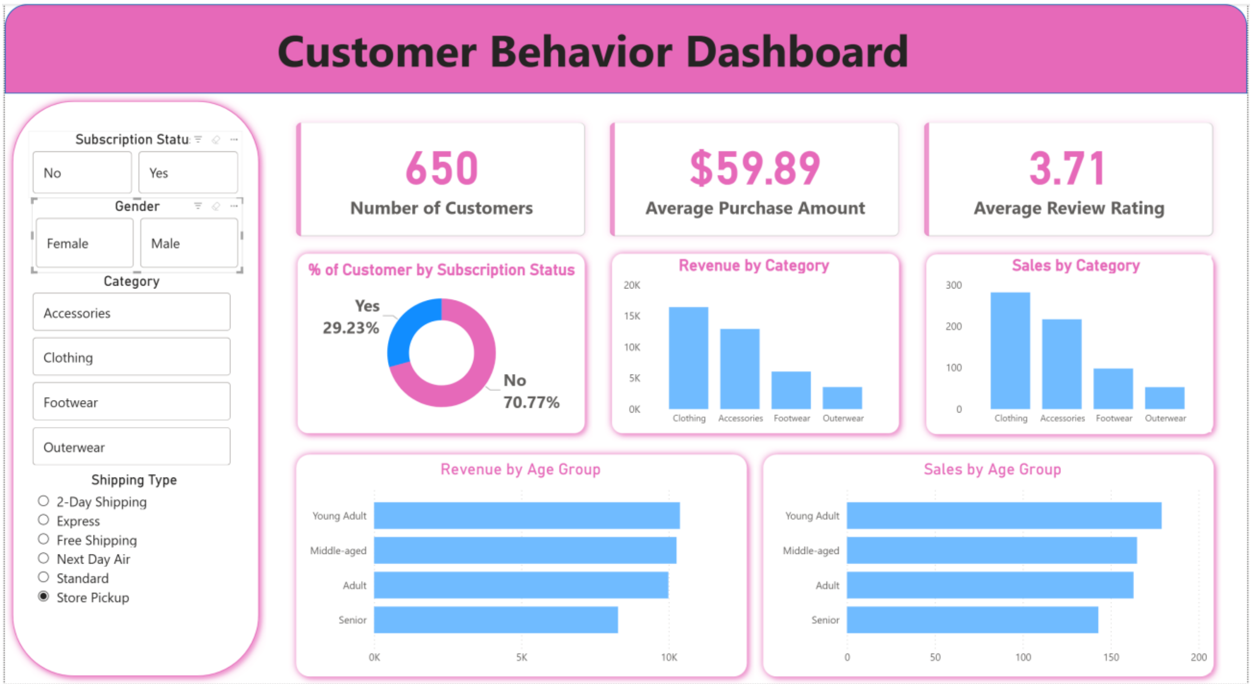
	subscription_status text	count bigint
1	No	2518
2	Yes	958

10. **Revenue by Age Group** – Calculated total revenue contribution of each age group.

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

5. Dashboard in Power BI

Finally, we built an interactive dashboard in Power BI to present insights visually.



6. Strategic Recommendations

Based on the data, the following strategies are recommended:

1. **Subscription Push:** Launch a targeted email campaign for the **958 repeat buyers** who haven't subscribed yet, offering a "Subscriber-Only" discount to bridge the gap.
2. **Age-Specific Campaigns:** Develop social media marketing specifically tailored for the **Young Adult** demographic to capitalize on their high spending power.
3. **Loyalty Rewards:** Implement a tiered loyalty program that rewards the "Loyal" segment with early access to new collections in the **Clothing** category.
4. **Shipping Upsell:** Offer "Free Express Shipping" for orders over a certain threshold to increase the average order value.