Business Report

Overview

The project aims to analyze customer trends in shopping by using data acquired from 3900 purchases across different franchises of the company across the states in the US. We seek to find insights into spending patterns, customer segments, product preferences, subscription trends to make strategic business decisions.

All the amounts were in the dollar currency but were changed into rands without using the financial conversion rate. The figures still make relative sense but greatly impact on the perceived size of the shop.

Data Information

Rows: 3900

Columns: 18

Important Features

- Customer Demographics (Age, Gender, Location, Subscription status)
- Purchase Details (Item Purchased, Purchase Amount, Size, Season)
- Shopping Trends (Discount Applied, Promo Code usage, Frequency of purchases)

Quick glance of the data looks like this:

```
df.head()
```

customer_id	age	gender	item_purchased	category	purchase_amount	location	size	color	season	review_rating	subscription_status	shipping_type	disco
1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	
2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	
3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	
4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	
5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	

Exploratory Data Analysis in Python

- In the first part of the project, data had to be cleaned, columns were renamed accordingly to fit a consistent naming convention.
- A column for creating different age groups was created in such a way that there would be 4 age groups

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

```
[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')
```

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

```
[9]: 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')
```

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

- Checked for redundancy between columns and removed it

```
[15]: #Check if there is a redundancy between the columns
(df['discount_applied'] == df['promo_code_used']).all()
```

```
[15]: np.True_
```

```
[16]: df = df.drop('promo_code_used', axis =1)
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')
```

- The data had 37 values missing in the Review Rating column

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

```
[5]: 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
```

```
[6]: #Replace the missing variable with the median review rating of that particular item (category)
#Clothing and footwear might have vastly different medians so it is important that we become specific with assigning the missing values

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

- Created a new feature to accurately assign purchase frequency by employing a frequency mapping

```
[12]: #Create a feature called purchase_frequency_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_mapping)
```

- Convert the DataFrame into a csv file so it can then be imported into SQL Server Management Studio

```
[23]: df.to_csv("customer_behavior.csv", index=False)
```

Data Analysis With SQL

In this section of the project, we had to implement analysis in SQL Server Management Studio to answer business questions. The SQL script used to answer the questions is included in the project folder.

1. Total revenue generated by each gender

	gender	revenue
1	Male	157890.00
2	Female	75191.00

2. Which customers used a discount but still spent more than the average purchase amount?

- The query returned 839 customers

Results Messages

	customer_id	purchase_amount
188	386	93.00
189	388	93.00
190	389	82.00
191	391	71.00
192	392	86.00
193	393	82.00
194	394	65.00
195	396	65.00
196	397	88.00
197	399	67.00
198	401	66.00
199	402	91.00
200	403	78.00
201	405	93.00
202	406	74.00

Query executed successfully.

3. What are the top 5 products with the highest average review rating? (Rating max =5)

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

4. Compare the average purchase amounts between Standard and Express shipping

	shipping_type	Average Purchase Amount by Shipping
1	Standard	58.460000
2	Express	60.480000

5. Do subscribed customers spend more? Comparing the average spend and total revenue between subscribers and non-subscribers

	subscription_status	total_customers	avg_spend	total_revenue
1	Yes	1053	59.490000	62645.00
2	No	2847	59.870000	170436.00

6. Which 5 products have the highest percentage of purchases with discounts applied?

Results Messages

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

7. Segment customers into New, Returning and Loyal based on their total number of previous purchases and show the count of each segment.

Customers who had only visited the store once are "New", while those with visits between 2 and 13 times are "Returning". Customers with more than 13 visits are "Loyal"

	customer_segment	Number of Customers
1	Returning	931
2	Loyal	2886
3	New	83

8. Are customers who are repeat buyers (more than 5 previous purchases) more likely to subscribe?

Results Messages

	subscription_status	repeat_buyers
1	Yes	958
2	No	2518

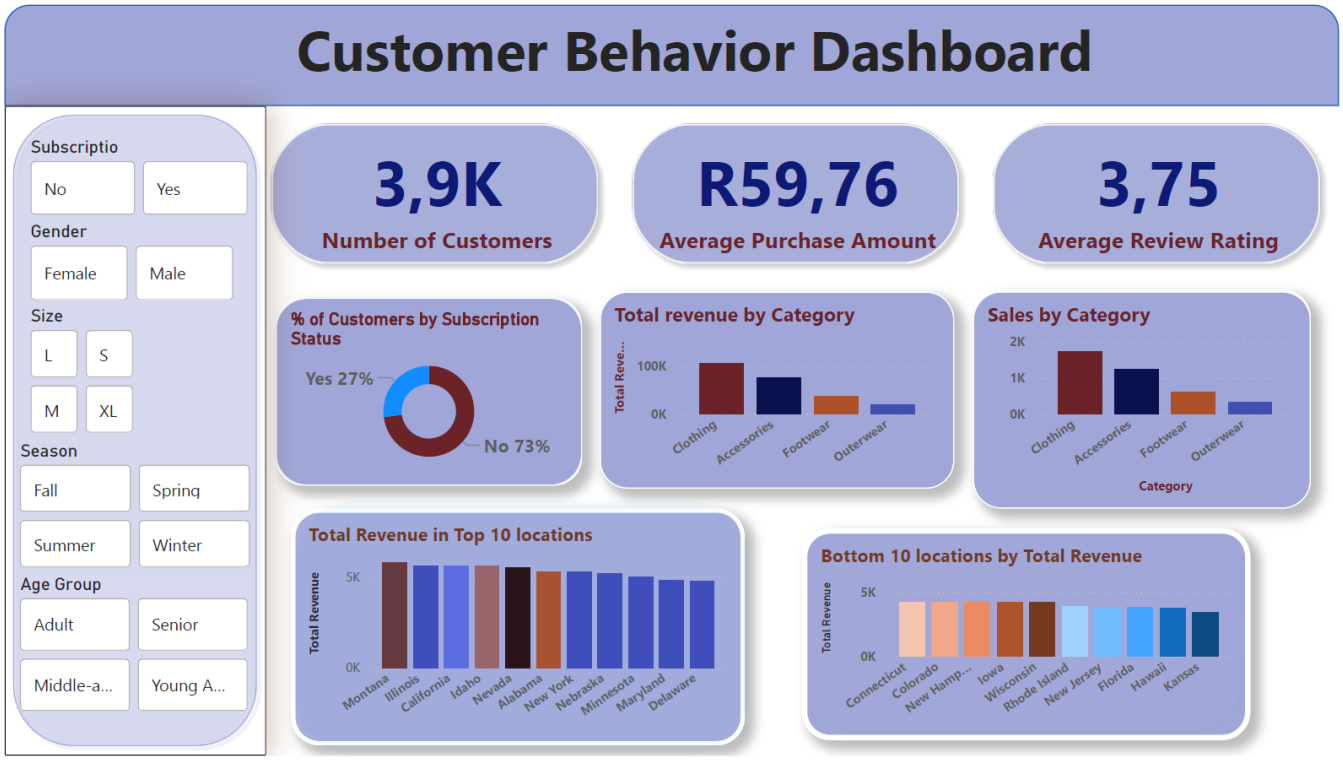
9. What is the total revenue contribution to each age group?

Results | Messages

	age_group	total_revenue
1	Young Adult	62143.00
2	Middle-aged	59197.00
3	Adult	55978.00
4	Senior	55763.00

Visualization with Power BI

The last part of the project is to build an *interactive* dashboard to present insights visually.



Business Recommendations

PAYMENT METHOD	USAGE %	AVG TRANSACTION
CREDIT CARD	28%	R 61.20
PAYPAL	24%	R 59.80
VENMO	22%	R 58.40

DEBIT CARD	15%	R 57.10
CASH	8%	R 52.30
BANK TRANSFER	3%	R 55.00

Business Implications:

- Digital payments dominate (74% combined)
- Ensure seamless checkout for top payment methods
- Consider offering small incentives for preferred payment types

Product Category Performance

CATEGORY	TOTAL REVENUE	AVG RATING	DISCOUNT USAGE
CLOTHING	R 145,000	3.6	45%
ACCESSORIES	R 92,000	3.8	48%
FOOTWEAR	R 68,000	3.7	42%
OUTERWEAR	R 35,000	3.5	40%

Business Implications:

- Clothing dominates revenue so ensure bestsellers are always *in stock*
- Accessories have highest ratings so *promote* as add-ons at checkout
- Outerwear has lowest revenue, consider *bundling* with other items