Retail_Customer_Analysis_Pipeline

Project Overview

This project analyzes customer shopping behavior using transactional data from 3,900 purchases across various product categories. The goal is to uncover insights into spending patterns, customer segments, product preferences, and subscription behavior to guide strategic business decisions.

The Core Objectives

According to the problem statement, the company's goals are focused on three main pillars:

- **Improve Sales & Growth:** Identifying which factors (discounts, seasons) drive decisions.
- **Customer Satisfaction & Loyalty:** Understanding repeat purchase behavior and preferences.
- **Strategic Optimization:** Refining marketing and product strategies based on shopping channels (online vs. offline) and demographics.
-

"Data shows that 25-34 year-olds shopping online are highly influenced by reviews but rarely use discounts. Recommendation: Shift marketing budget from 'Promo Codes' to 'Influencer/Review Campaigns' for this segment."

Whether you’re applying for Data Analyst jobs or building your portfolio, this project has everything:

✅ Define a real business problem statement from a retail company use case

✅ Load, clean, and transform data using Python (pandas)

✅ Simulate business transactions and run advanced queries using SQL

✅ Visualize insights through a beautiful Power BI dashboard

✅ Create a project report for documentation (mandatory in real teams)

✅ Build a presentation deck using Gamma AI

✅ Publish everything on GitHub to make your portfolio stand out

Here is how to approach each deliverable to ensure your portfolio stands out:

1. Data Preparation & Modeling (Python)

Your goal here is to transform "raw" data into "clean" data.

- **Handle Messy Data: Check for missing values in critical columns like Payment Preferences or Reviews.**
- **Feature Engineering: Create a new column for "Season" based on transaction dates and a "Repeat Customer" flag to satisfy the management's interest in loyalty.**
- **Transformation: Use Pandas to ensure all data types are correct (e.g., dates are in datetime format) before exporting to SQL.**

2. Data Analysis (SQL)

Use SQL to perform the heavy lifting of segmenting your customers.

- **Segmentation Queries: Write queries to group customers by demographics (age, gender) and sales channel (online vs. offline).**
- **Driver Analysis: Calculate the average spend or purchase frequency for customers who used discounts versus those who didn't.**
- **Join Operations: If your data is in multiple tables, demonstrate your ability to use JOIN to link customer profiles with transaction history.**

3. Visualization & Insights (Power BI)

Don't just make a report; make a strategic tool.

- **Interactive Filters: Include slicers for Season, Product Category, and Sales Channel so stakeholders can drill down into specific patterns.**
- **KPI Cards: Highlight "Customer Lifetime Value" or "Repeat Purchase Rate" prominently.**
- **Trend Lines: Use line charts to show how purchasing patterns have changed over time across different demographics.**

4. Report and Presentation

This is where you answer the "So What?"

- **Executive Summary: Lead with the most impactful finding (e.g., "Online shoppers are 30% more likely to repeat purchase when reviews are present").**
- **Actionable Recommendations: Suggest specific marketing shifts based on your findings, such as targeting specific seasons for discount-heavy campaigns.**

5. GitHub Repository

This serves as your "proof of work."

- **ReadMe File: Include a high-level summary of the business problem and your findings.**
- **Structure: Organize folders clearly: /Scripts for Python, /Queries for SQL, and /Dashboard for your Power BI file.**

Dataset Summary

- Rows: 3,900
- Columns: 18
- Key Features:
  - Customer demographics (Age, Gender, Location, Subscription Status)
  - Purchase details (Item Purchased, Category, Purchase Amount, Season, Size, Color)
  - Shopping behavior (Discount Applied, Promo Code Used, Previous Purchases, Frequency of Purchases, Review Rating, Shipping Type)
- Missing Data: 37 values in Review Rating column

How to use Google Colab with your Excel file:

1. **Open Google Colab:** Go to colab.google and click **New Notebook**.
2. **Upload your Excel file:**
   - On the left sidebar, click the **Folder icon** 📁.
   - Click the **Upload icon** (upward arrow) and select your Excel file from your computer.
3. **Wait for the upload:** Once the progress bar finishes, you will see your file in the sidebar.
4. **Read the file with Python:** Use the following code in a cell:

Python

```
import pandas as pd
# 1. Right-click your file in the sidebar and select "Copy path"
# 2. Paste it inside the quotes below
path = '/content/your_uploaded_file.xlsx'

df = pd.read_excel(path)

# Display the first few rows
df.head()
```

Why Google Colab is Great for Your Portfolio

- **Zero Setup:** You don't have to worry about pip install or environment errors.
- **Shareable:** You can click the "Share" button just like a Google Doc and include that link in your portfolio so recruiters can see your live code.
- **Google Drive Integration:** You can save your datasets in Google Drive and "mount" your drive to Colab so you don't have to re-upload your file every time you open the notebook.

Exploratory Data Analysis using Python

The "Big Picture" Check
Before cleaning, you need to see what you're working with—column names, data types, and any obviously missing data.

We began with data preparation and cleaning in Python:

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

```
# Check the shape (Rows, Columns) and data types
print(df.info())

# Check for missing values in each column
print("\nMissing Values:\n", df.isnull().sum())
```

-

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discou Appli
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3863.000000	3900	3900	39
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	NaN	2	6	
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	NaN	No	Free Shipping	
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	NaN	2847	675	22
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	3.750065	NaN	NaN	Na
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	0.716983	NaN	NaN	Na
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	2.500000	NaN	NaN	Na
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	3.100000	NaN	NaN	Na
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	3.800000	NaN	NaN	Na
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	4.400000	NaN	NaN	Na
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	5.000000	NaN	NaN	Na

Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
3900	3900	3900.000000	3900	3900
2	2	NaN	6	7
No	No	NaN	PayPal	Every 3 Months
2223	2223	NaN	677	584
NaN	NaN	25.351538	NaN	NaN
NaN	NaN	14.447125	NaN	NaN
NaN	NaN	1.000000	NaN	NaN
NaN	NaN	13.000000	NaN	NaN
NaN	NaN	25.000000	NaN	NaN
NaN	NaN	38.000000	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.

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

- **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'})

- **Feature Engineering:**
  - Created **age_group** column by binning customer ages.
  - Created **purchase_frequency_days** column from purchase data.
- **Data Consistency Check:** Verified if `discount_applied` and `promo_code_used` were redundant; dropped `promo_code_used`.

```
# Create column 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)

# Check the new column alongside the original
```

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

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

Data Analysis using SQL (Business Transactions)

The Business Questions

1. What is the total revenue generated by male vs. female customers?
2. Which customers used a discount but still spent more than the average purchase amount?
3. Which are the top 5 products with the highest average review rating?
4. Compare the average Purchase Amounts between Standard and Express Shipping.
5. Do subscribed customers spend more? Compare average spend and total revenue between subscribers and non-subscribers.
6. Which 5 products have the highest percentage of purchases with discounts applied?
7. Segment customers into New, Returning, and Loyal based on their total number of previous purchases, and show the count of each segment.
8. What are the top 3 most purchased products within each category?
9. Are customers who are repeat buyers (more than 5 previous purchases) also likely to subscribe?
10. What is the revenue contribution of each age group?

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.

	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
10	22	62
11	24	88

Total rows: 839 Query complete 00:00:(

3. **Top 5 Products by Rating** – Found products with the highest average review ratings.

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

	shipping_type text	round numeric
1	Standard	58.46
2	Express	60.48

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

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

	item_purchased text	discount_rate numeric
1	Hat	50.00
2	Sneakers	49.66
3	Coat	49.07
4	Sweater	48.17
5	Pants	47.37

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

	customer_segment text	Number 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.

	item_rank bigint	category text	item_purchased text	total_orders bigint
1	1	Accessories	Jewelry	171
2	2	Accessories	Sunglasses	161
3	3	Accessories	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	repeat_buyers 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

Data Pipeline Block Diagram

- Phase 1: Raw Data (CSV/Excel) → Google Colab (Python Cleaning).
- Phase 2: Clean Data → SQLAlchemy Tunnel → Neon Cloud (PostgreSQL).
- Phase 3: Neon Cloud → DirectQuery → Power BI Service (Live Dashboard).

Dashboard in Power BI

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

Business Recommendations

- **Boost Subscriptions** – Promote exclusive benefits for subscribers.
- **Customer Loyalty Programs** – Reward repeat buyers to move them into the “Loyal” segment.
- **Review Discount Policy** – Balance sales boosts with margin control.
- **Product Positioning** – Highlight top-rated and best-selling products in campaigns.
- **Targeted Marketing** – Focus efforts on high-revenue age groups and express-shipping users.