

# Análise do Comportamento de Compra do Cliente

## 1. Visão Geral do Projeto

Este projeto analisa o comportamento de compra dos clientes utilizando dados transacionais de 3.900 compras em diversas categorias de produtos. O objetivo é descobrir insights sobre padrões de gastos, segmentos de clientes, preferências de produtos e comportamento de assinatura para orientar decisões de negócios estratégicas.

## 2. Sumário do Dataset

- Linhas: 3.900

- Colunas: 18

### Principais Características:

- Demografia do cliente (Idade, Gênero, Localização, Status de Assinatura).
- Detalhes da compra (Item Comprado, Categoria, Valor da Compra, Estação, Tamanho, Cor).
- Comportamento de compra (Desconto Aplicado, Código Promocional Utilizado, Compras Anteriores, Frequência de Compras, Média de Avaliação, Tipo de Frete).

**Dados Faltantes:** 37 valores na coluna de Média de Avaliação.

## 3. Análise Exploratória com Python

Iniciamos com a preparação e limpeza dos dados:

- **Carregamento de Dados:** Importação do conjunto de dados com **pandas**.
- **Exploração Inicial:** Uso de **df.info()** e **describe()** para estatísticas sumárias.

```
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
```

df.describe(include='all')

	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
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3863.000000	3900	3900	3900	3900	3900.000000	3900	3900
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	NaN	2	6	2	2	NaN	6	7
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	NaN	No	Free Shipping	No	No	NaN	PayPal	Every 3 Months
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	NaN	2847	675	2223	2223	NaN	677	584
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	3.750065	NaN	NaN	NaN	NaN	25.351538	NaN	NaN
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	0.716983	NaN	NaN	NaN	NaN	14.447125	NaN	NaN
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	2.500000	NaN	NaN	NaN	NaN	1.000000	NaN	NaN
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	3.100000	NaN	NaN	NaN	NaN	13.000000	NaN	NaN
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	3.800000	NaN	NaN	NaN	NaN	25.000000	NaN	NaN
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	4.400000	NaN	NaN	NaN	NaN	38.000000	NaN	NaN
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	5.000000	NaN	NaN	NaN	NaN	50.000000	NaN	NaN

- **Tratamento de Dados Faltantes:** Imputação de valores nulos na coluna de Média de Avaliação utilizando a mediana de cada categoria de produto.

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

df.isnull().sum()

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	

- **Padronização de Colunas:** Renomeação para o formato *snake\_case*.

df.columns = df.columns.str.lower()  
df.columns = df.columns.str.replace(' ', '\_')  
df = df.rename(columns={'purchase\_amount\_(usd)': 'purchase\_amount'})

df.columns

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'
-------	---

- **Engenharia de Recursos:** Criação das colunas *age\_group* e *purchase\_frequency\_days*.

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

df[['age', 'age\_group']].head(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

```
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)
```

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

	purchase_frequency_days	frequency_of_purchases
0	14	Fortnightly
1	14	Fortnightly
2	7	Weekly
3	7	Weekly
4	365	Annually
5	7	Weekly
6	90	Quarterly
7	7	Weekly
8	365	Annually
9	90	Quarterly

- **Consistência de Dados:** Verificou-se redundância entre `discount_applied` e `promo_code_used`; a coluna `promo_code_used` foi removida.

```
(df['discount_applied'] == df['promo_code_used']).all()
```

```
np.True_
```

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

```
df.columns
```

```
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')
```

- **Integração com Banco de Dados:** Conexão com PostgreSQL para carregar o DataFrame limpo para análise via SQL.

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

```
Requirement already satisfied: psycopg2-binary in c:\users\gabri\appdata\local\programs\python\python314\lib\site-packages (2.9.11)
Requirement already satisfied: sqlalchemy in c:\users\gabri\appdata\local\programs\python\python314\lib\site-packages (2.0.46)
Requirement already satisfied: greenlet>=1 in c:\users\gabri\appdata\local\programs\python\python314\lib\site-packages (from sqlalchemy) (3.3.1)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\gabri\appdata\local\programs\python\python314\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
```

```
username = "postgres"
password = "7912449"
host = "localhost"
port = "5432"
database = "customer_behavior"
```

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

```
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. Análise de Dados com SQL (Transação de Negócios)

Foi realizada uma análise estruturada para responder perguntas-chave:

1. **Receita por Gênero:** Comparação da receita total entre clientes masculinos e femininos.

	gender text	revenue numeric
1	Female	75191
2	Male	157890

**2. Usuários de Desconto com Gastos Altos:** Clientes que usaram descontos, mas ainda gastaram acima da média.

	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
12	29	94
13	32	79
14	33	67
15	35	61

Total rows: 839      Query complete 00:00:00.245

**3. Top 5 Produtos por Avaliação:** Produtos encontrados com a maior média de avaliação.

	item_purchased text	Média de Avaliação de Produtos numeric
1	Gloves	3.86
2	Sandals	3.84
3	Boots	3.82
4	Hat	3.80
5	Skirt	3.78

**4. Comparação de Tipo de Entrega:** Média de compra para frete Padrão (Standard) vs. Expresso (Express).

	shipping_type text	round numeric
1	Standard	58.46
2	Express	60.48

**5. Assinantes vs. Não Assinantes:** Comparação da média de gastos dos clientes e da receita total por status de assinatura (Yes ou No em 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. Produtos Dependentes de Desconto:** Identificação de 5 produtos com a maior porcentagem de compras com desconto.

	item_purchased text	Média de Avaliação de Produtos numeric
1	Gloves	3.86
2	Sandals	3.84
3	Boots	3.82
4	Hat	3.80
5	Skirt	3.78

**7. Segmentação de Clientes:** Classificação de clientes como Novo, Recorrente e Fiel baseado no histórico de compras.

	customer_segment text	Número de consumidores bigint
1	Fiel	3116
2	Recorrente	701
3	Novo	83

**8. Top 3 Produtos por Categoria:** Lista dos produtos mais vendidos de cada categoria.

	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. Compradores Repetitivos e Assinaturas:** Verificação de quais clientes com menos de 5 compras estão mais suscetíveis à assinatura.

	subscription_status text	repeat_buyers bigint
1	No	2518
2	Yes	958

**10. Receita por Faixa Etária:** Cálculo total de contribuição na receita por faixa etária.

Young Adult - Jovem Adulto

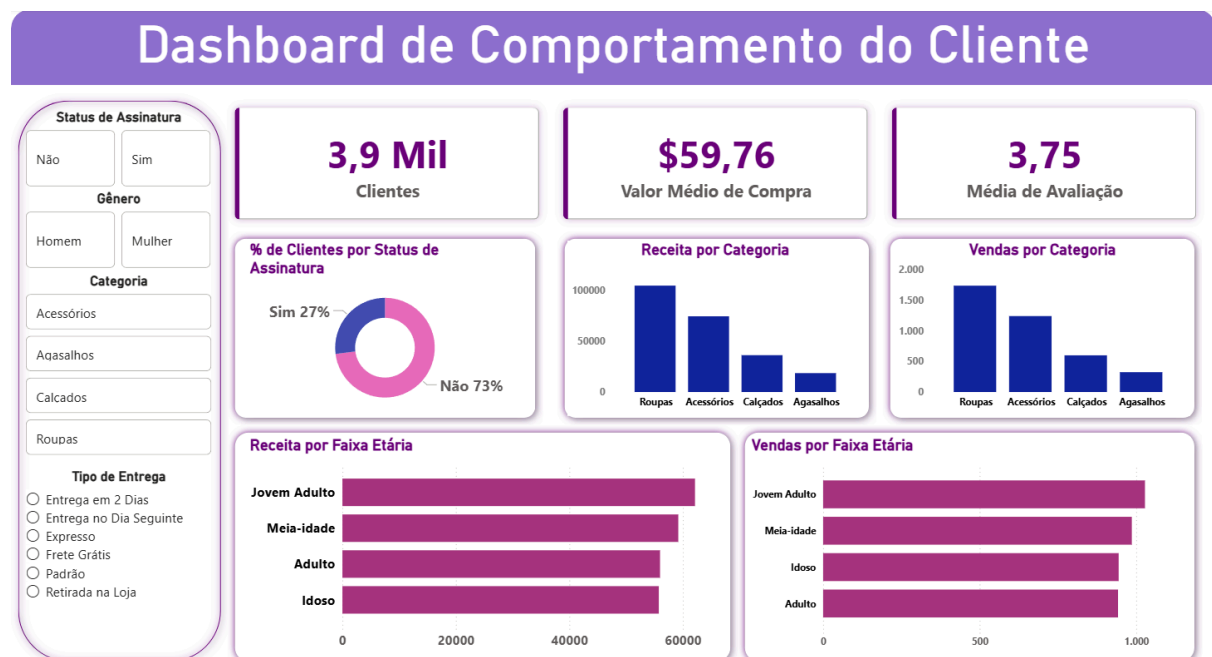
Adult - Adulto

Middle-aged - Meia-idade

Senior - Idoso

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

## 5. Dashboard no PowerBI



## 6. Recomendações de Negócio

- **Impulsionar Assinaturas:** Promover benefícios exclusivos para assinantes.
- **Programas de Fidelidade:** Recompensar compradores repetitivos para migrá-los ao segmento "Fiel".
- **Revisar Política de Descontos:** Equilibrar o aumento de vendas com o controle de margem.
- **Posicionamento de Produto:** Destacar produtos bem avaliados em campanhas.
- **Marketing Direcionado:** Focar esforços em faixas etárias de alta receita e usuários de frete expresso.