Desafio Santander 2023 API ETL

Pandas para análise dos dados

```
import pandas as pd

pd.read_csv("shopping_trends_updated.csv")
```

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Se
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	W
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	W
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	S
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	S
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	S
...	...	...	...	...	...	...	...	...	...	
3895	3896	40	Female	Hoodie	Clothing	28	Virginia	L	Turquoise	Sur
3896	3897	52	Female	Backpack	Accessories	49	Iowa	L	White	S
3897	3898	46	Female	Belt	Accessories	33	New Jersey	L	Green	S
3898	3899	44	Female	Shoes	Footwear	77	Minnesota	S	Brown	Sur
3899	3900	52	Female	Handbag	Accessories	81	California	M	Beige	S

3900 rows × 18 columns

Criação de um DataFrame nome tendencias e consulta 5 valores mais recorrentes

```
import pandas as pd

tendencias = pd.read_csv("shopping_trends_updated.csv")
tendencias.head()
```

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring

Consulta de todos os valores da coluna: Item Purchased (item adquirido)

```
tendencias['Item Purchased']
```

```
0         Blouse
1        Sweater
2          Jeans
3        Sandals
4         Blouse
          ...
3895      Hoodie
3896    Backpack
3897        Belt
3898       Shoes
3899     Handbag
Name: Item Purchased, Length: 3900, dtype: object
```

Consulta frequência de compra de cada item comprado, quantidade de vezes que foi comprado

```
tendencias['Item Purchased'].value_counts()
```

```
Blouse        171
Jewelry       171
Pants         171
Shirt         169
Dress         166
Sweater       164
Jacket        163
Belt          161
Sunglasses    161
Coat          161
Sandals       160
Socks         159
Skirt         158
Shorts        157
Scarf         157
Hat           154
Handbag       153
Hoodie        151
Shoes         150
T-shirt       147
Sneakers      145
Boots         144
Backpack      143
Gloves        140
Jeans         124
Name: Item Purchased, dtype: int64
```

Consulta nome itens da coluna Item Purchased

```
itens = tendencias['Item Purchased'].unique()
print(itens)
```

```
['Blouse' 'Sweater' 'Jeans' 'Sandals' 'Sneakers' 'Shirt' 'Shorts' 'Coat'
 'Handbag' 'Shoes' 'Dress' 'Skirt' 'Sunglasses' 'Pants' 'Jacket' 'Hoodie'
 'Jewelry' 'T-shirt' 'Scarf' 'Hat' 'Socks' 'Backpack' 'Belt' 'Boots'
 'Gloves']
```

Consulta média frequência de compra dos itens

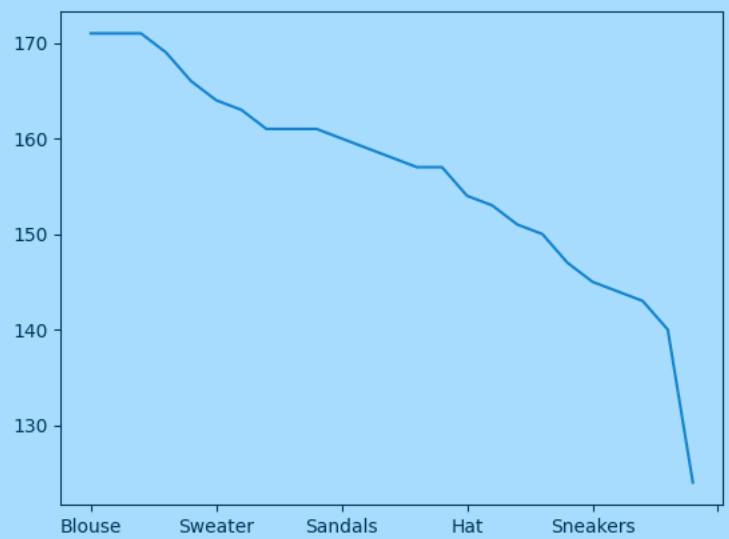
```
quantidade = tendencias['Item Purchased'].value_counts()
media = quantidade.mean()
print(f'Média = {media:.0f}')
```

```
Média = 156
```

Gráfico quantidade comprada cada item

```
quantidade.plot()
```

```
<Axes: >
```



Média Idade Consumidor

```
media_idade_consumidor = tendencias.Age.mean()
print(f'Idade média do consumidor: {media_idade_consumidor:.0f}')
```

```
Idade média do consumidor: 44
```

Consulta quantidade consumidores por gênero

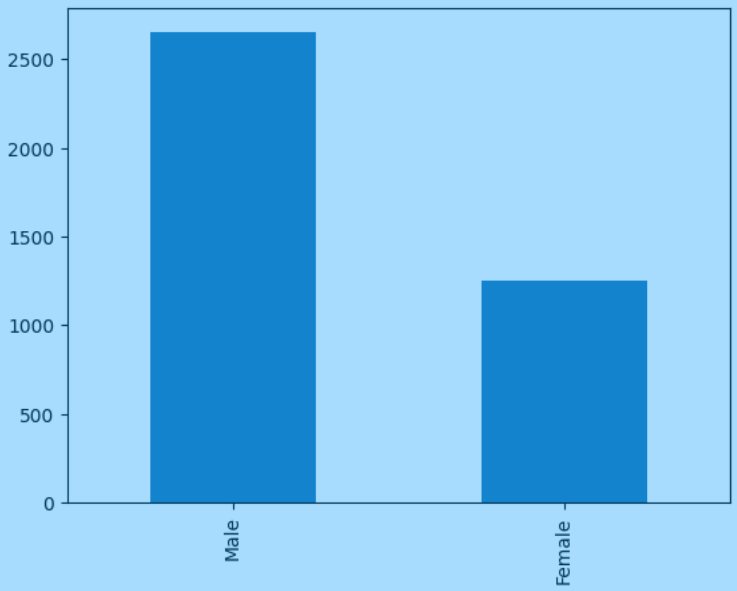
```
tendencias['Gender'].value_counts()
```

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

Gráfico quantidade consumidores por gênero

```
quantidade_comprada_por_genero = tendencias['Gender'].value_counts()
quantidade_comprada_por_genero.plot(kind='bar')
```

```
<Axes: >
```




```
!pip install openai
```

```
Requirement already satisfied: openai in /usr/local/lib/python3.10/dist-packages (0.28.1)
Requirement already satisfied: requests>=2.20 in /usr/local/lib/python3.10/dist-packages (from openai) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from openai) (4.66.1)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from openai) (3.8.6)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai) (2023.7.22)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (23.1.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (6.0.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (4.0.3)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (1.9.2)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (1.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (1.3.1)
```

API

```
import openai
import dotenv
import os

dotenv.load_dotenv()

openai.api_key = os.getenv(OPENAI_API_KEY)
result = openai.ChatCompletion.create(
    model = "gpt-3.5-turbo",
    messages = [{"role": "user", "content": "Gere 3 argumentos para compra de roupas para homens com mais de 40 anos"},
    ])
print(result)
```