

Projeto de Análise de Dados

amazon prime video

Exploração e tratamento de dados de
filmes e séries

Ana Clara Lomeu Rosa
Isabel Azar de Holanda
Júlio César de Souza





O dataset contém informações sobre filmes e séries da Amazon Prime Video.

As colunas principais que utilizamos:

`title` → título do filme/série

`type` → Filme ou Série

`release_year` → ano de lançamento

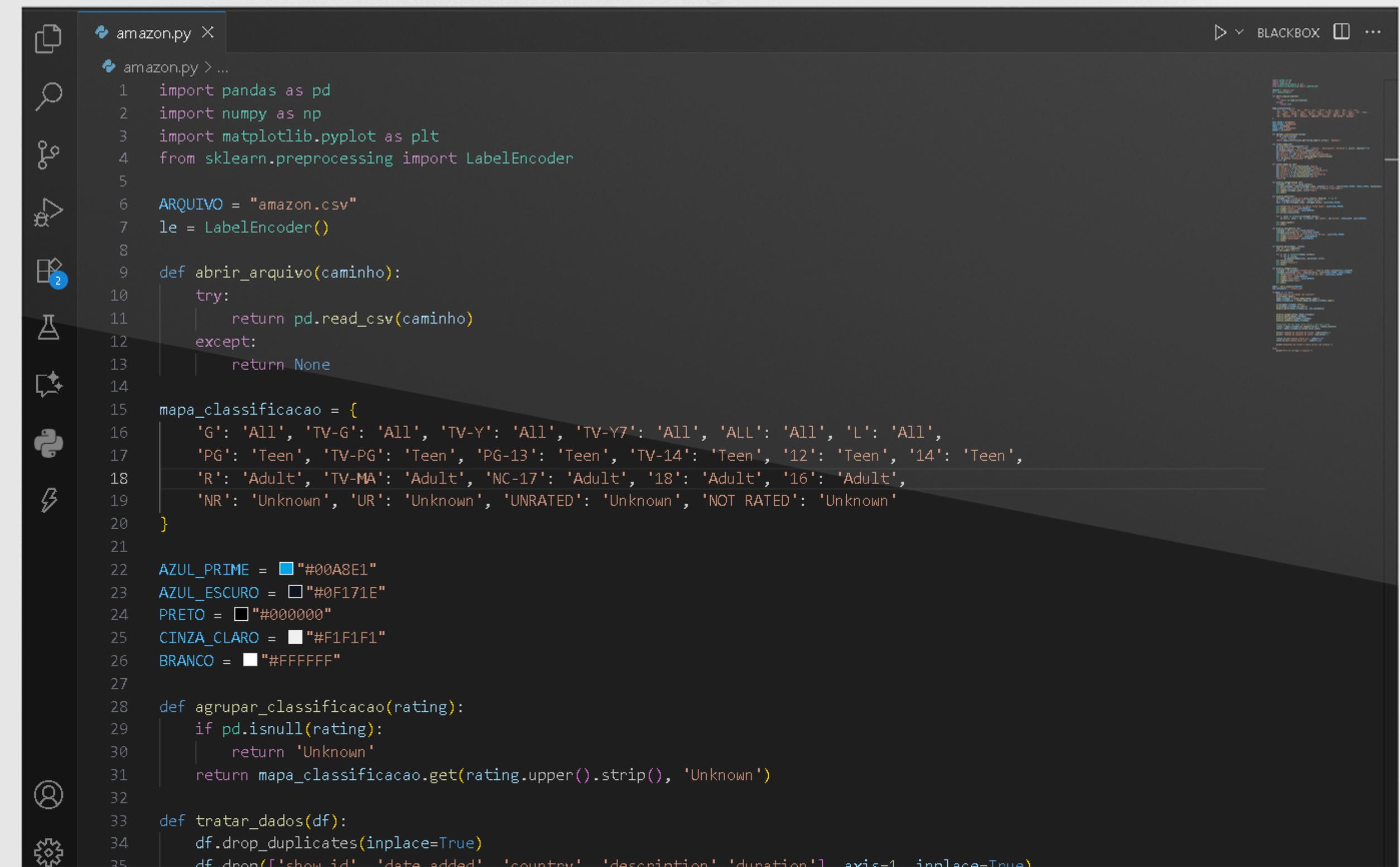
`rating` → classificação indicativa (faixa etária)

`listed_in` → gêneros (ex: Drama, Comédia, Documentário)

`duration` → duração (minutos ou temporadas)

`director / cast` → diretor(es) e elenco principal

Carregamento do Dataset

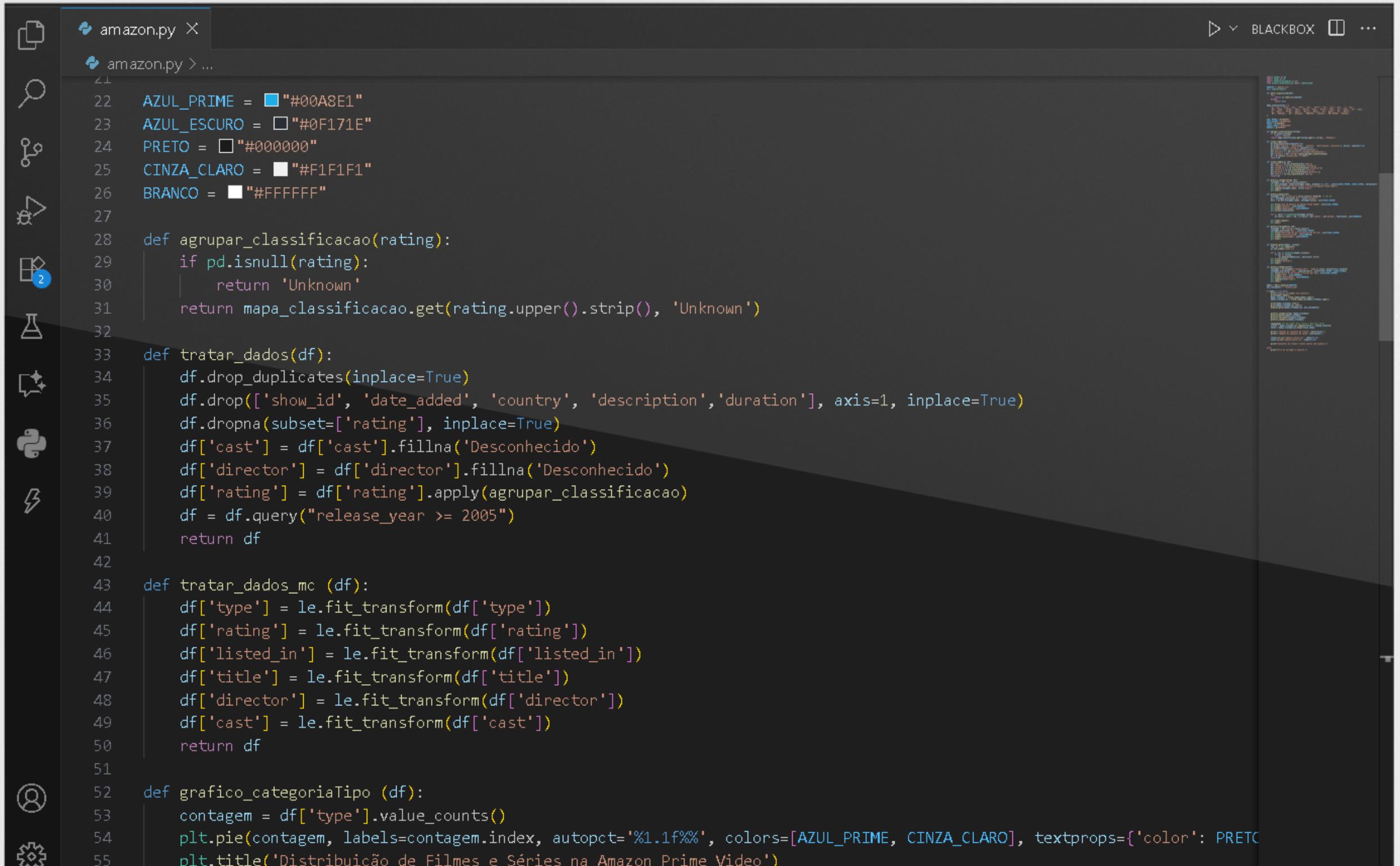


The image shows a Jupyter Notebook interface with a dark theme. On the left, there's a sidebar with various icons for file operations, search, and other notebook functions. The main area displays a Python script named 'amazon.py'. The code is as follows:

```
amazon.py X
amazon.py > ...
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from sklearn.preprocessing import LabelEncoder
5
6 ARQUIVO = "amazon.csv"
7 le = LabelEncoder()
8
9 def abrir_arquivo(caminho):
10     try:
11         return pd.read_csv(caminho)
12     except:
13         return None
14
15 mapa_classificacao = {
16     'G': 'All', 'TV-G': 'All', 'TV-Y': 'All', 'TV-Y7': 'All', 'ALL': 'All', 'L': 'All',
17     'PG': 'Teen', 'TV-PG': 'Teen', 'PG-13': 'Teen', 'TV-14': 'Teen', '12': 'Teen', '14': 'Teen',
18     'R': 'Adult', 'TV-MA': 'Adult', 'NC-17': 'Adult', '18': 'Adult', '16': 'Adult',
19     'NR': 'Unknown', 'UR': 'Unknown', 'UNRATED': 'Unknown', 'NOT RATED': 'Unknown'
20 }
21
22 AZUL_PRIME = "#00A8E1"
23 AZUL_ESCURO = "#0F171E"
24 PRETO = "#000000"
25 CINZA_CLARO = "#F1F1F1"
26 BRANCO = "#FFFFFF"
27
28 def agrupar_classificacao(rating):
29     if pd.isnull(rating):
30         return 'Unknown'
31     return mapa_classificacao.get(rating.upper().strip(), 'Unknown')
32
33 def tratar_dados(df):
34     df.drop_duplicates(inplace=True)
35     df.drop(['show_id', 'date_added', 'country', 'description', 'duration'], axis=1, inplace=True)
```

Tratamento dos Dados

- Remoção de duplicados e colunas irrelevantes;
- Tratamento de valores ausentes(nulos);
- Padronização da classificação etária (All, teen, Adult, Unknown);
- Filtro de ano de lançamento.



The screenshot shows a Jupyter Notebook interface with a dark theme. The left sidebar contains icons for file operations, search, and other notebook functions. The main area displays the following Python code:

```
azul_prime = "#00A8E1"
azul_escuro = "#0F171E"
preto = "#000000"
cinza_claro = "#F1F1F1"
branco = "#FFFFFF"

def agrupar_classificacao(rating):
    if pd.isnull(rating):
        return 'Unknown'
    return mapa_classificacao.get(rating.upper().strip(), 'Unknown')

def tratar_dados(df):
    df.drop_duplicates(inplace=True)
    df.drop(['show_id', 'date_added', 'country', 'description', 'duration'], axis=1, inplace=True)
    df.dropna(subset=['rating'], inplace=True)
    df['cast'] = df['cast'].fillna('Desconhecido')
    df['director'] = df['director'].fillna('Desconhecido')
    df['rating'] = df['rating'].apply(agrupar_classificacao)
    df = df.query("release_year >= 2005")
    return df

def tratar_dados_mc(df):
    le = LabelEncoder()
    df['type'] = le.fit_transform(df['type'])
    df['rating'] = le.fit_transform(df['rating'])
    df['listed_in'] = le.fit_transform(df['listed_in'])
    df['title'] = le.fit_transform(df['title'])
    df['director'] = le.fit_transform(df['director'])
    df['cast'] = le.fit_transform(df['cast'])
    return df

def grafico_categoriaTipo(df):
    contagem = df['type'].value_counts()
    plt.pie(contagem, labels=contagem.index, autopct='%1.1f%%', colors=[azul_prime, cinza_claro], textprops={'color': preto})
    plt.title('Distribuição de Filmes e Séries na Amazon Prime Video')
```

Gráfico Ano Lançamento

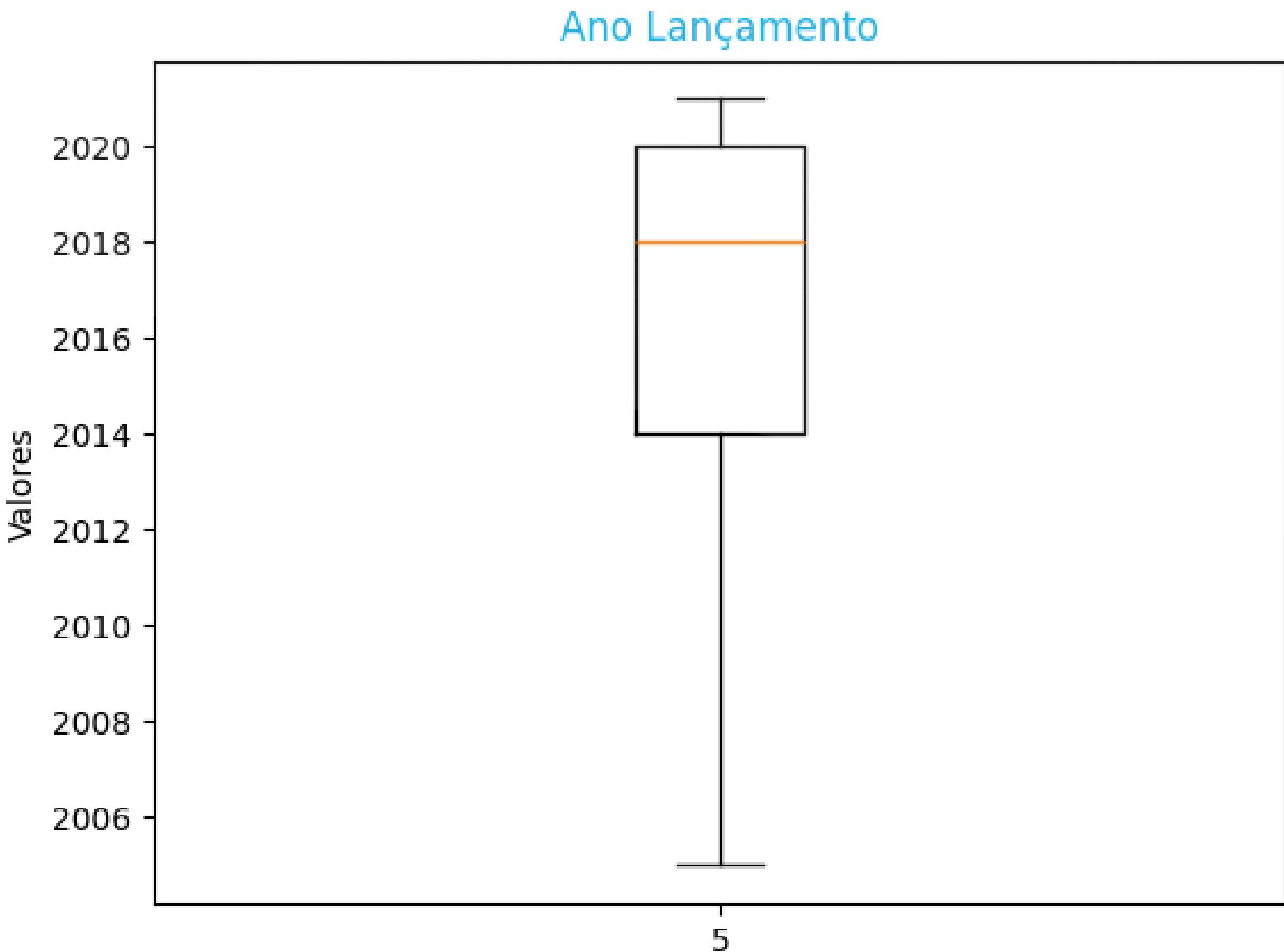
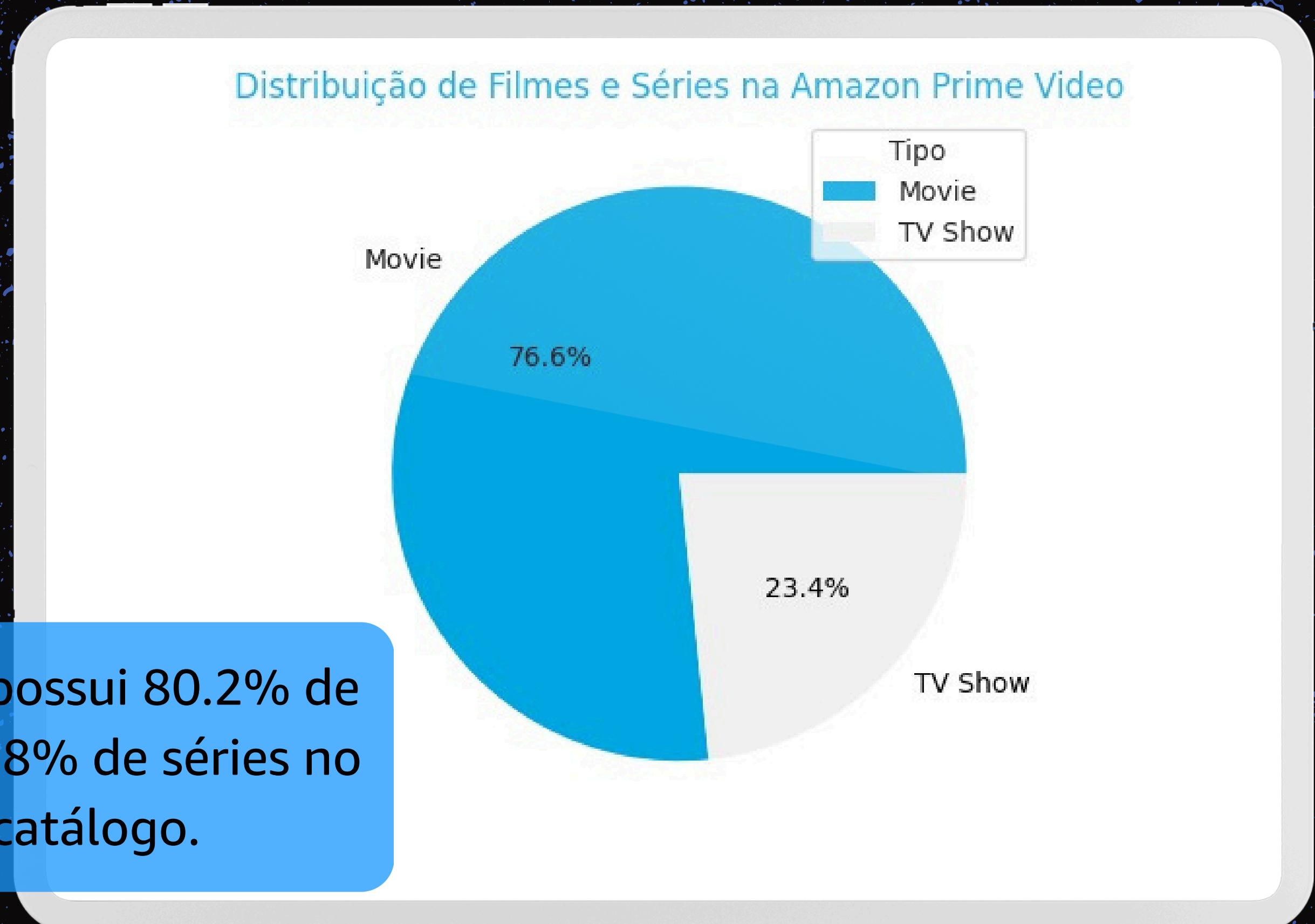


Gráfico Distribuição de Filmes vs Séries



O gênero que mais se destaque é Drama

Gráfico

Top 10 Gêneros

Top 10 Gêneros na Amazon Prime Video

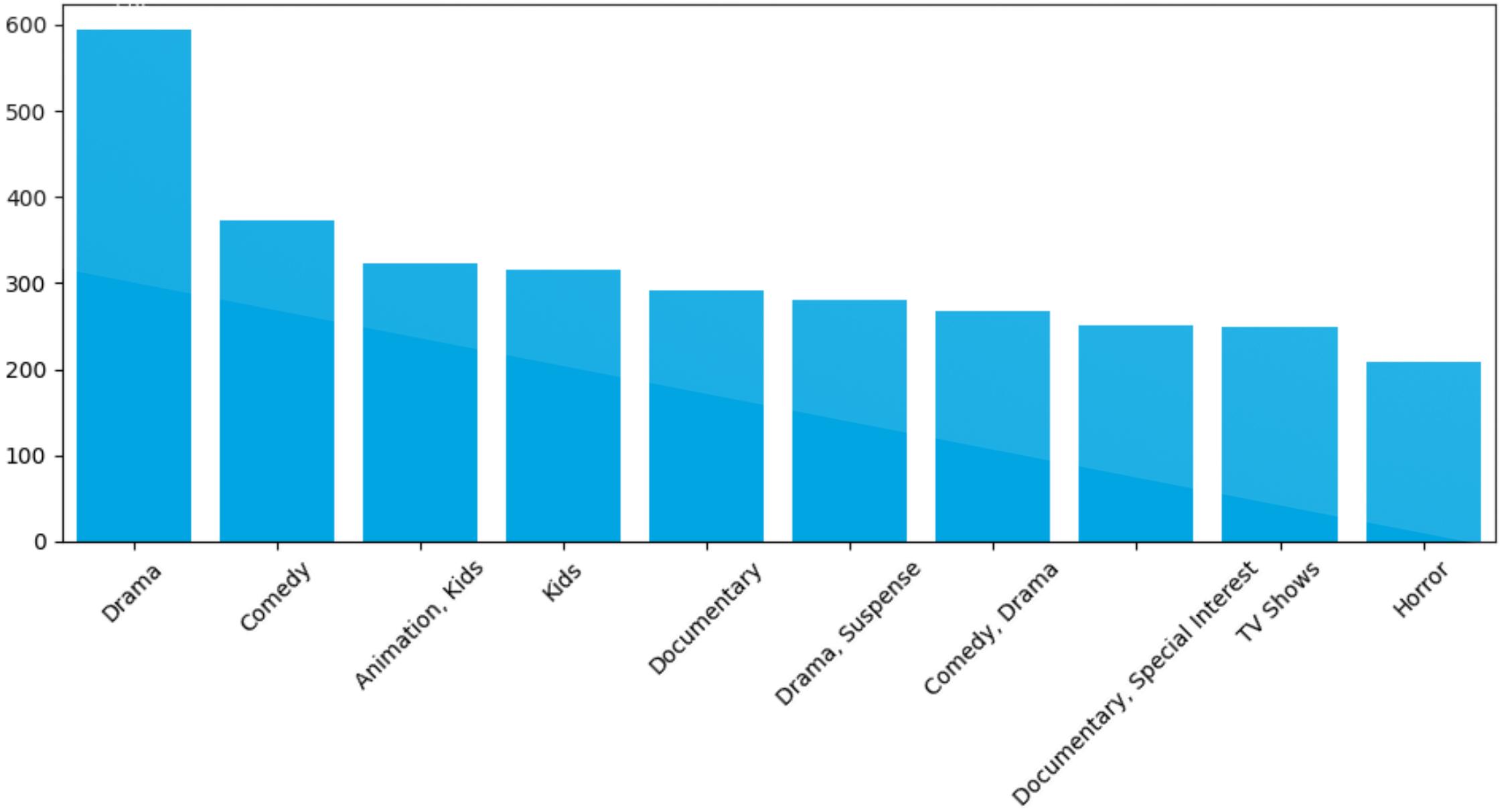


Gráfico Distribuição por Classificação Etária

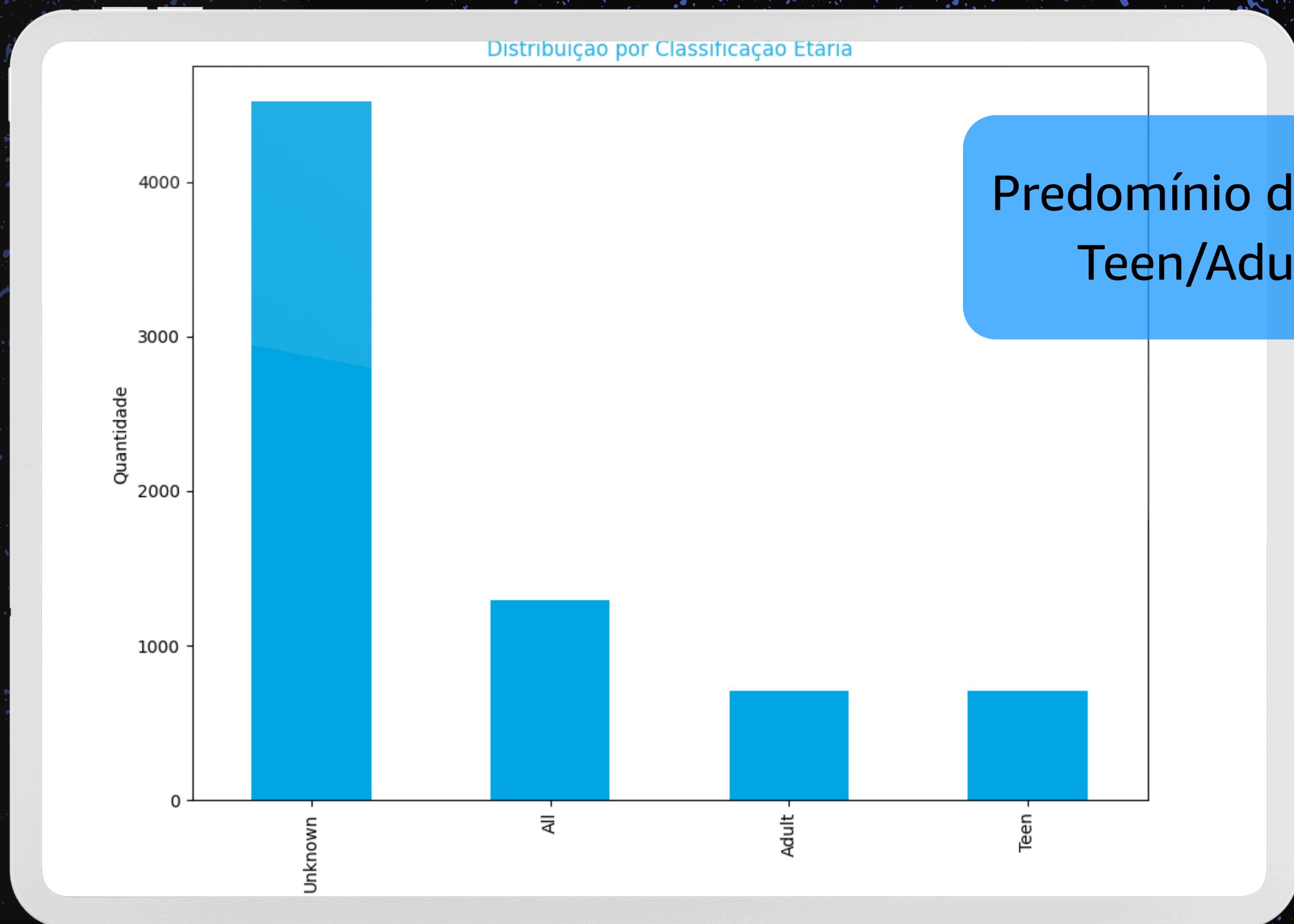
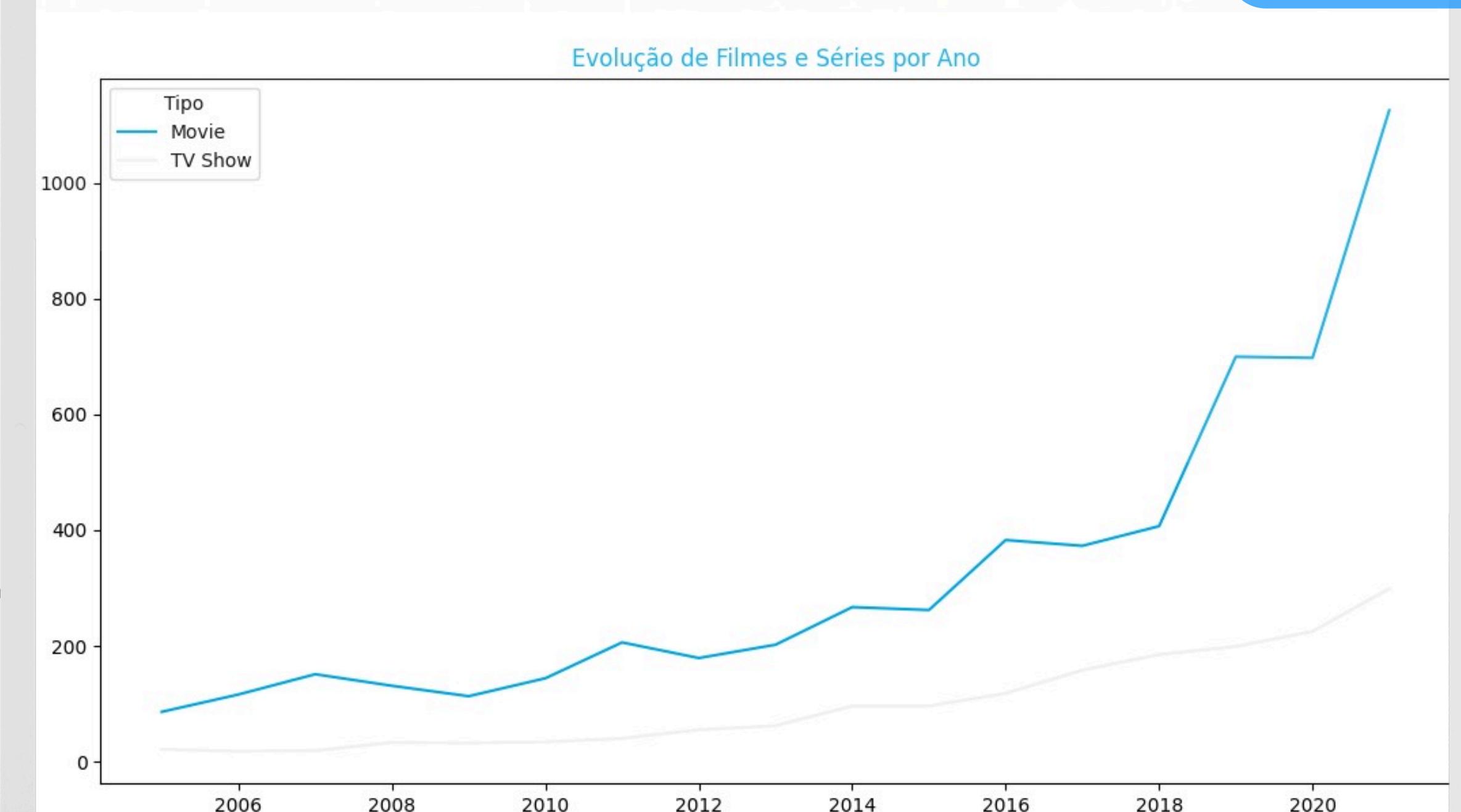


Gráfico Tendência por Ano

Maior concentração de títulos após 2000



O nosso dataset
foi dividido em
70% treino e
30% teste



The screenshot shows a Jupyter Notebook interface with a dark theme. The left sidebar contains icons for file operations, search, and other notebook functions. The main area displays a Python script named `amazon.py`. The code performs the following steps:

- Imports and handles data loading.
- Prints the first few rows of the loaded data.
- Creates two versions of the data: `dados_tratados` and `dados_tratados_mc`.
- Prints basic information about the treated data.
- Creates a boxplot for the year of release.
- Generates four types of plots: `grafico_categoriaTipo`, `grafico_Genero`, `grafico_faixaEtaria`, and `grafico_tendencia`.
- Splits the data into training and testing sets, maintaining a 70% to 30% ratio.
- Prints the sizes of the training and testing datasets.
- Saves the training and testing datasets to CSV files: `amazon_treino.csv` and `amazon_teste.csv`.
- Prints a success message indicating the datasets were saved successfully.
- If there's an error, it prints an error message.

```
amazon.py X
amazon.py > grafico_categoriaTipo
106
107 if dados is not None:
108     print("Arquivo carregado com sucesso")
109     print(dados.head())
110     dados_tratados = tratar_dados(dados.copy())
111     dados_tratados_mc = tratar_dados_mc(dados_tratados.copy())
112
113     print(dados_tratados.info())
114     print(dados_tratados_mc.info())
115     boxplot_geral(dados_tratados_mc, ano_lancamento, "Ano Lançamento")
116
117
118     grafico_categoriaTipo (dados_tratados)
119     grafico_Genero(dados_tratados)
120     grafico_faixaEtaria(dados_tratados)
121     grafico_tendencia(dados_tratados)
122
123 #Separando 70% dos dados para treino e 30% para teste
124 treino = dados_tratados_mc.sample(frac=0.7, random_state=42)
125 teste = dados_tratados_mc.drop(treino.index)
126
127     print(f'Tamanho do conjunto de treino: {len(treino)}')
128     print(f'Tamanho do conjunto de teste: {len(teste)}')
129
130     treino.to_csv('amazon_treino.csv', index=False)
131     teste.to_csv('amazon_teste.csv', index=False)
132
133     print("Conjuntos de treino e teste salvos com sucesso.")
134
135 else:
136     print("Erro ao carregar o arquivo.")
```

O dataset da Amazon Prime Video permite insights como:

- Maior proporção de filmes em relação a series;
- Gêneros mais populares na plataforma;
- Classificações etárias predominantes;
- Crescimento do catálogo nos últimos anos;

Futuras análises
avaliar sucesso dos títulos, analisar
atores/diretores mais frequentes.

OBRIGADO!
amazon
prime video

