

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

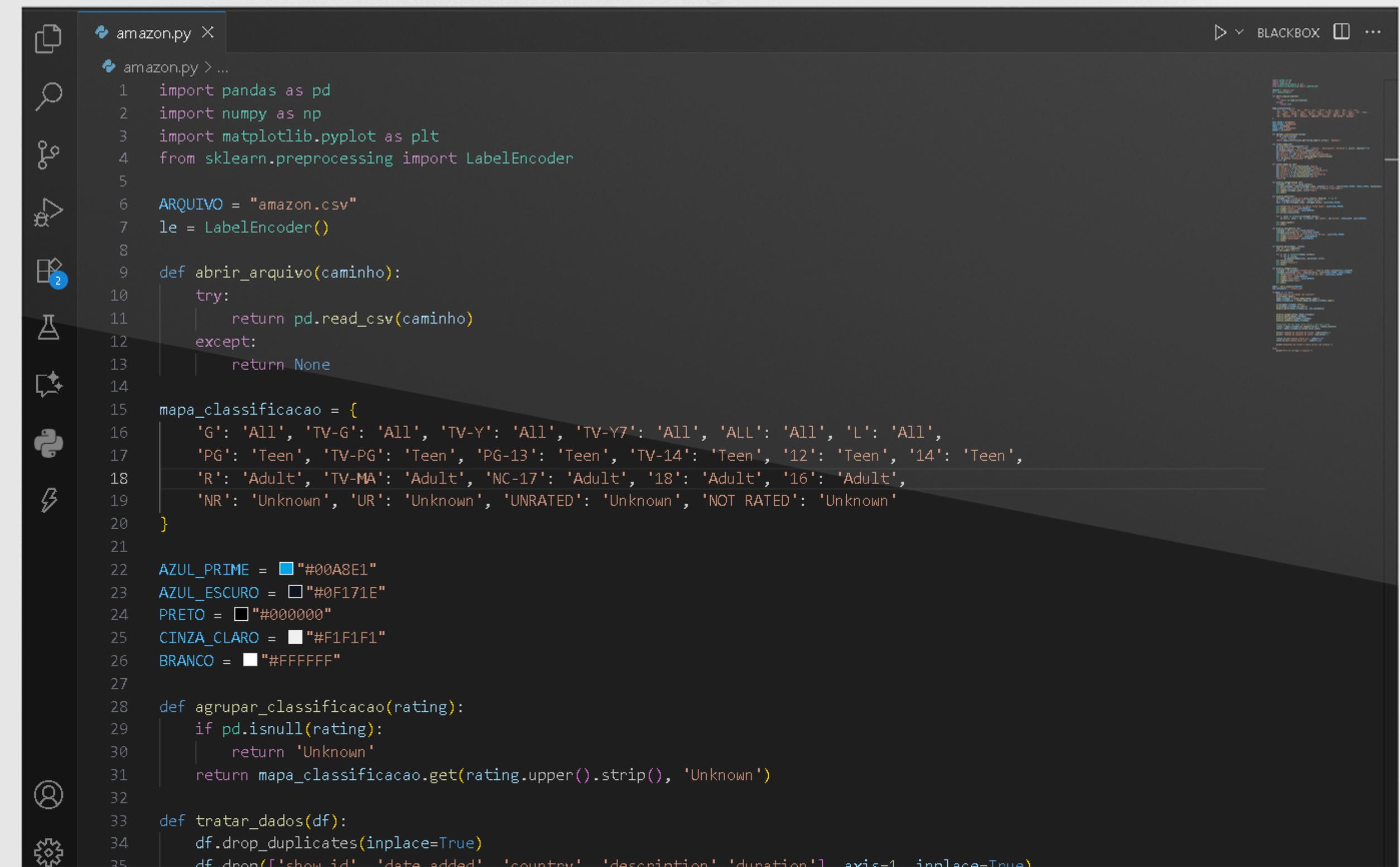
`rating` → classificação indicativa (faixa etária)

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

`duration` → duração (minutos ou temporadas)

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

Carregamento do Dataset

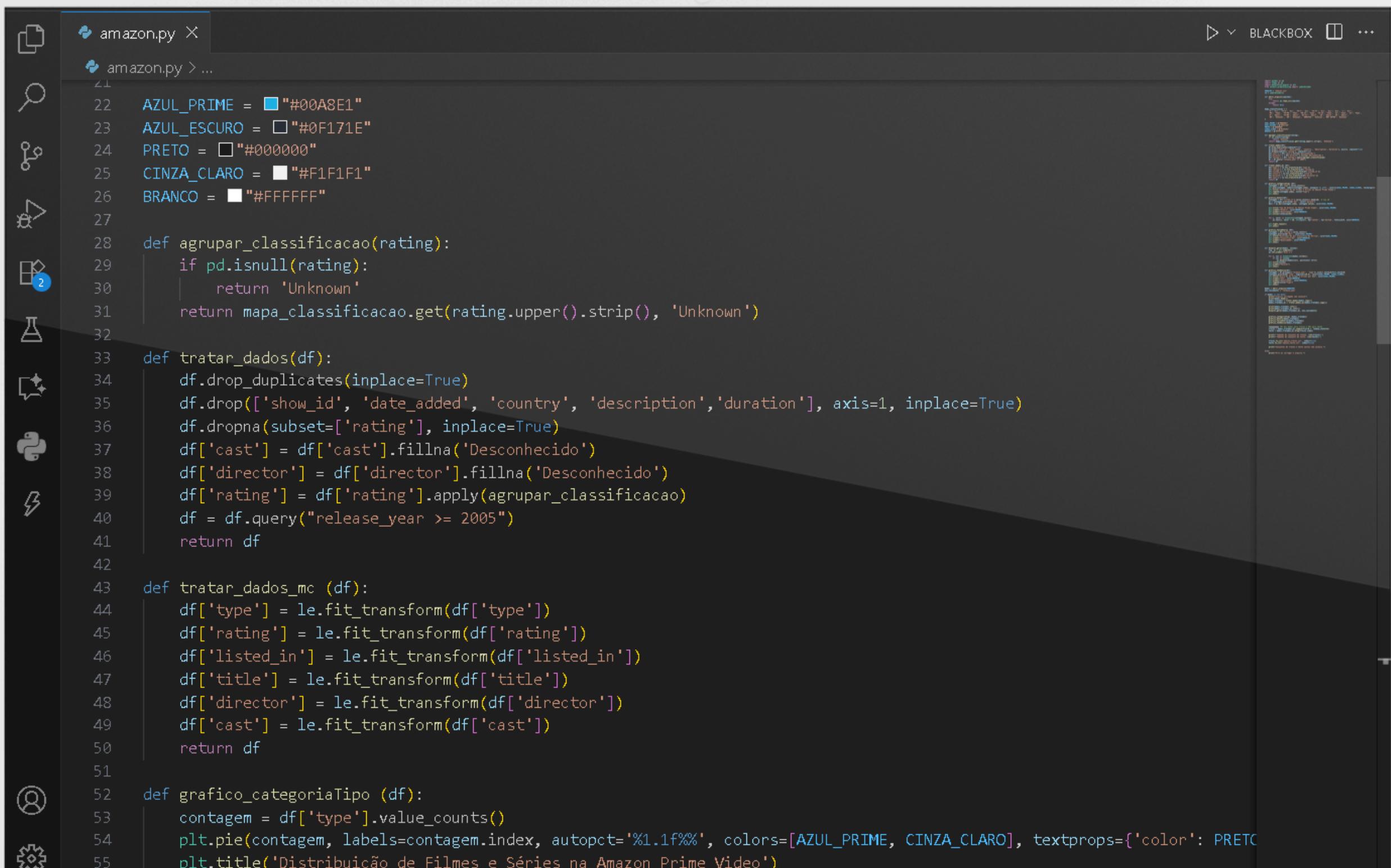


The image shows a Jupyter Notebook interface with a dark theme. The left sidebar contains various icons for file operations, search, and notebook management. The main area displays a Python script named `amazon.py`. The code performs the following tasks:

- Imports necessary libraries: `pandas`, `numpy`, `matplotlib.pyplot`, and `sklearn.preprocessing.LabelEncoder`.
- Specifies the CSV file path: `ARQUIVO = "amazon.csv"`.
- Creates a `LabelEncoder` object: `le = LabelEncoder()`.
- Defines a function `abrir_arquivo(caminho)` to read the CSV file using `pd.read_csv(caminho)`. It handles exceptions and returns `None` if reading fails.
- Creates a mapping `mapa_classificacao` from movie ratings to categories. The mapping includes:
 - 'G': 'All', 'TV-G': 'All', 'TV-Y': 'All', 'TV-Y7': 'All', 'ALL': 'All', 'L': 'All'
 - 'PG': 'Teen', 'TV-PG': 'Teen', 'PG-13': 'Teen', 'TV-14': 'Teen', '12': 'Teen', '14': 'Teen'
 - 'R': 'Adult', 'TV-MA': 'Adult', 'NC-17': 'Adult', '18': 'Adult', '16': 'Adult'
 - 'NR': 'Unknown', 'UR': 'Unknown', 'UNRATED': 'Unknown', 'NOT RATED': 'Unknown'
- Defines color constants: `AZUL_PRIME` (#00A8E1), `AZUL_ESCURO` (#0F171E), `PRETO` (#000000), `CINZA_CLARO` (#F1F1F1), and `BRANCO` (#FFFFFF).
- Defines a function `agrupar_classificacao(rating)` that returns 'Unknown' if the rating is null or uses the mapping to get the category for the uppercased and stripped rating.
- Defines a function `tratar_dados(df)` that drops duplicate rows and removes columns `'show_id'`, `'date_added'`, `'country'`, `'description'`, and `'duration'` along axis 1.

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

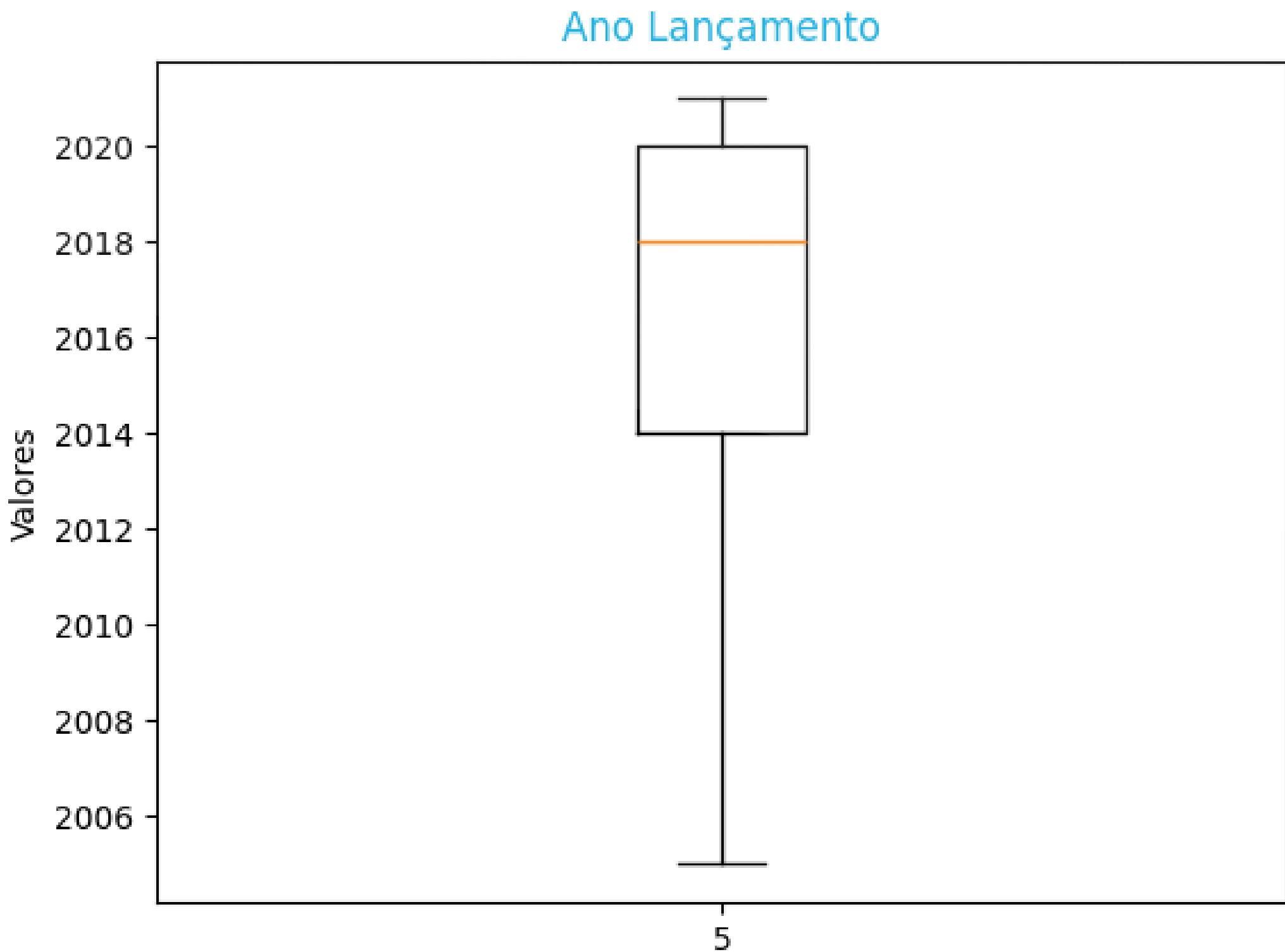
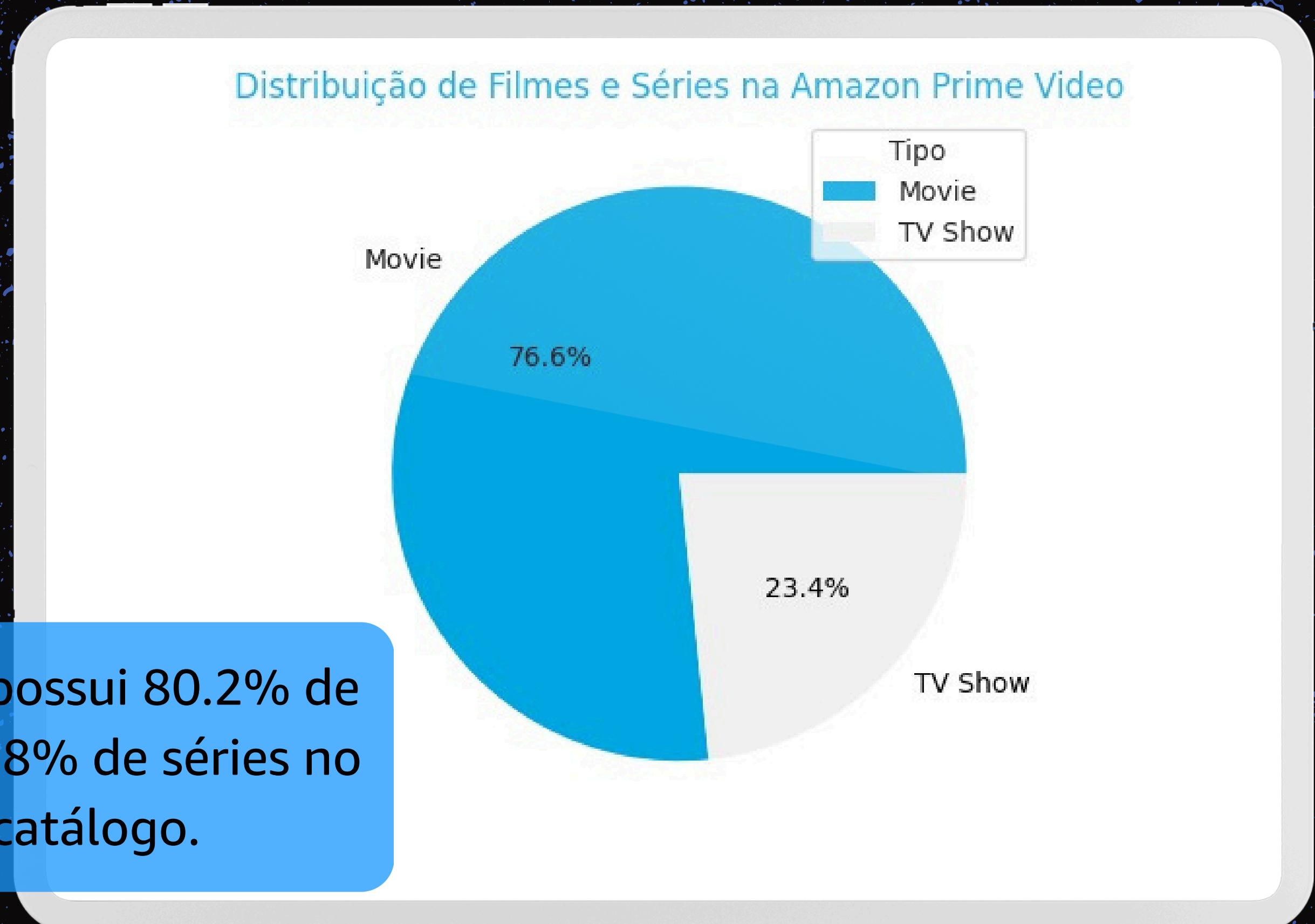


Gráfico Distribuição de Filmes vs Séries



O gênero que mais se destaque é Drama

Gráfico

Top 10 Gêneros

Top 10 Gêneros na Amazon Prime Video

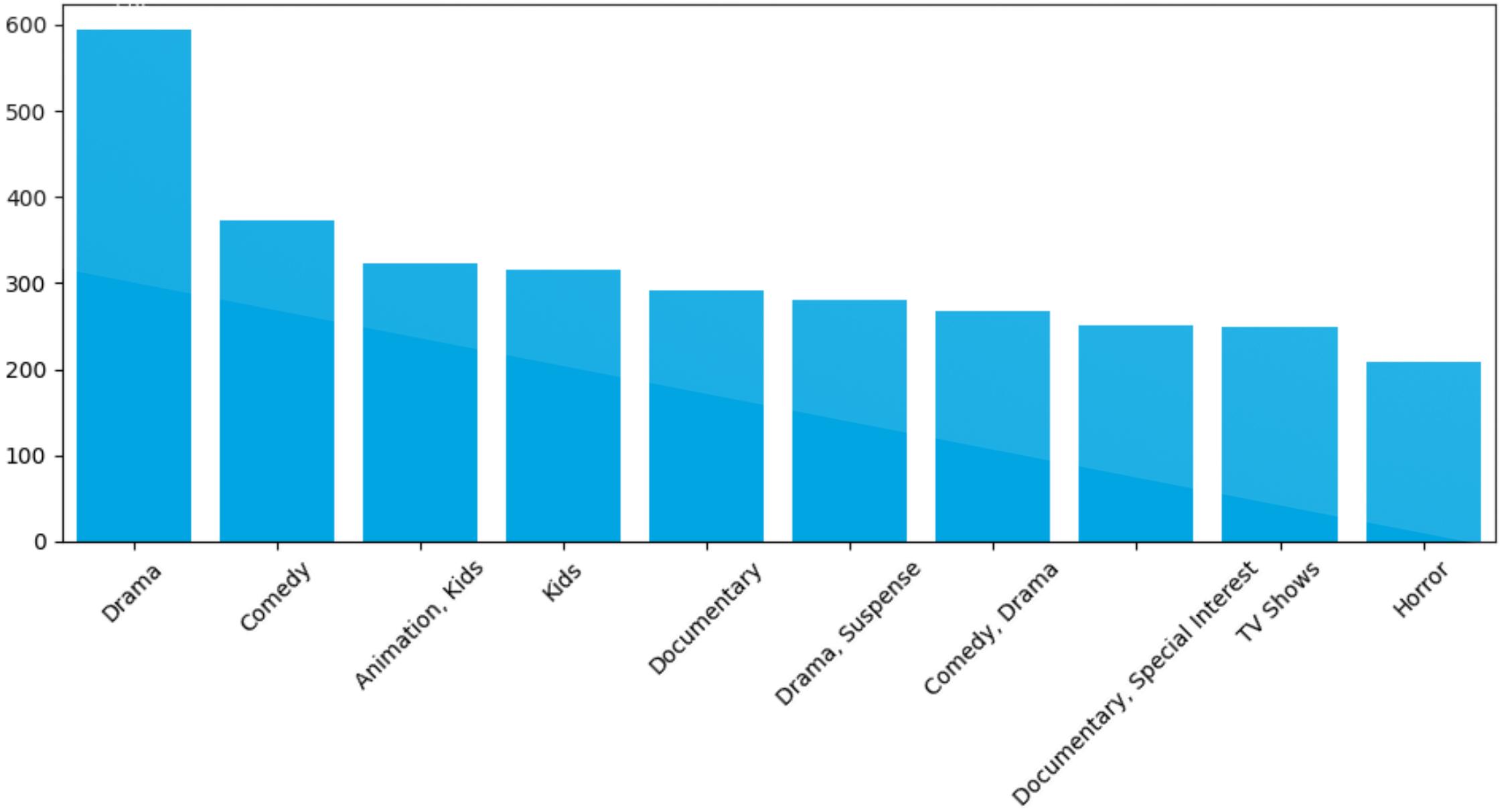
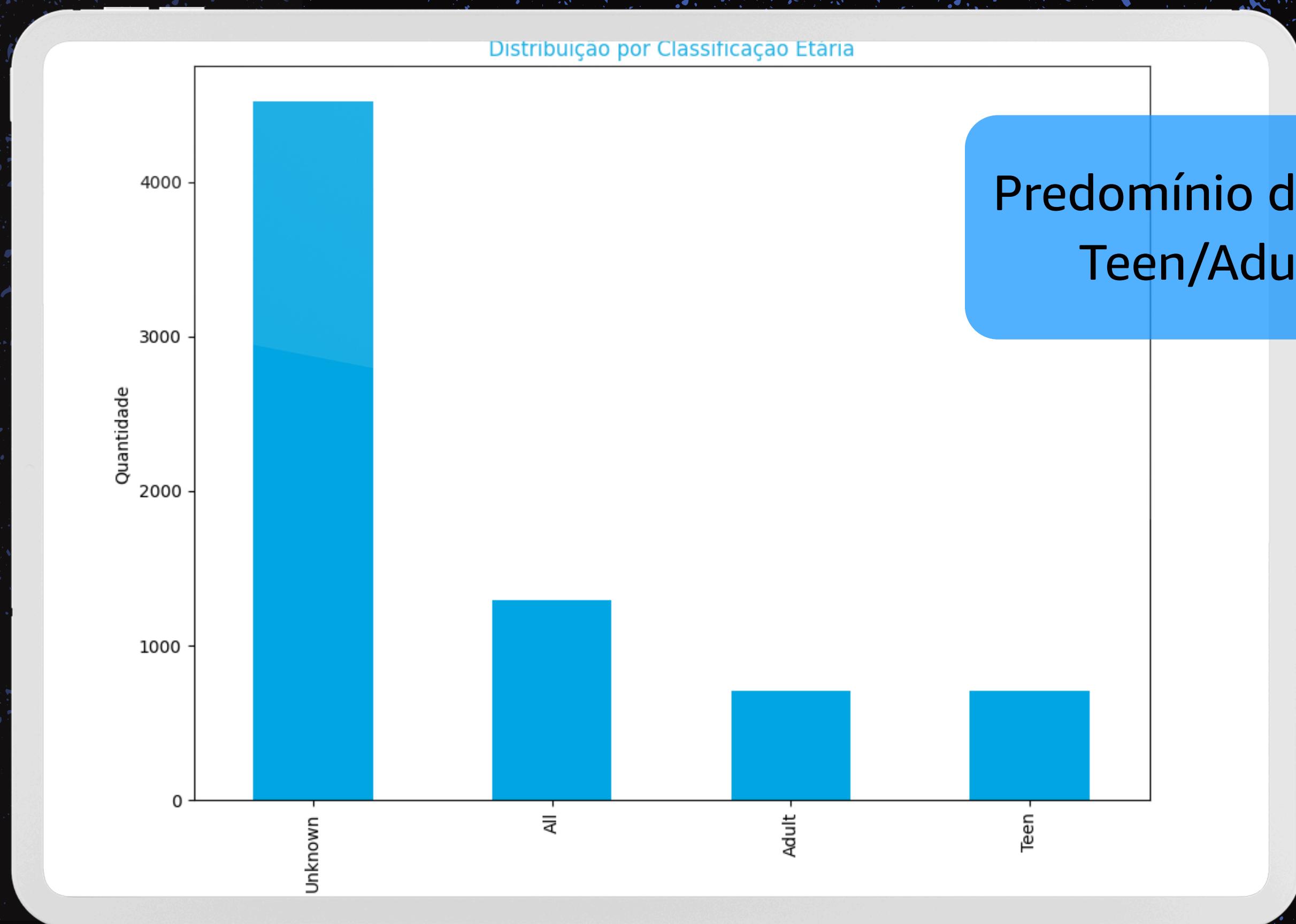


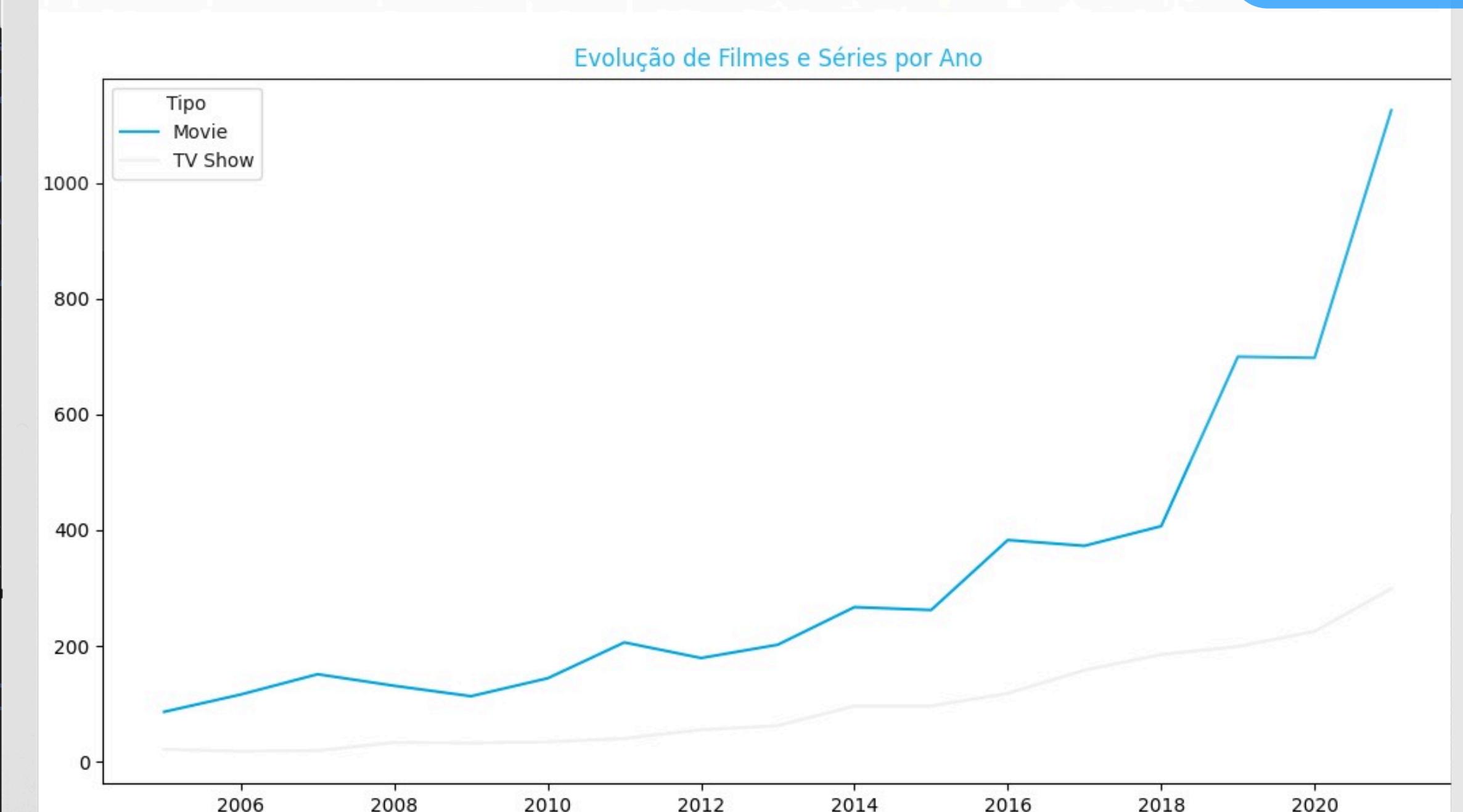
Gráfico Distribuição por Classificação Etária



Predomínio do público
Teen/Adult/etc

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 `grafico_categoriaTipo`.
- Handles data loading: prints success message, prints head, creates `dados_tratados` and `dados_tratados_mc` from `dados`, prints info for both.
- Creates boxplots: `boxplot_geral(dados_tratados_mc, ano_lancamento, "Ano Lançamento")`.
- Creates various plots: `grafico_categoriaTipo(dados_tratados)`, `grafico_Genero(dados_tratados)`, `grafico_faixaEtaria(dados_tratados)`, `grafico_tendencia(dados_tratados)`.
- Splits the data: separates 70% for training and 30% for testing using `sample(frac=0.7, random_state=42)`.
- Prints the sizes of the training and testing sets.
- Saves the datasets to CSV files: `treino.to_csv('amazon_treino.csv', index=False)` and `teste.to_csv('amazon_teste.csv', index=False)`.
- Prints a success message: "Conjuntos de treino e teste salvos com sucesso."
- Catches errors: prints "Erro ao carregar o arquivo." if an error occurs.

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

