Marco Aurélio Costa da Silva

Relatório Técnico

Ciência de Dados

Relatório de análises para o desafio da área de ciência de dados do programa Lighthouse da empresa Indicium.

Sumário

1. Introdução

O presente relatório técnico foi elaborado como parte do desafio proposto pela Indicium para a posição de Trainee em Cientista de Dados. O objetivo central consiste em aplicar técnicas estatísticas e de ciência de dados para analisar um conjunto de informações cinematográficas, respondendo a questões de negócio relevantes para a PProductions, um estúdio de Hollywood interessado em orientar suas próximas produções.

A resolução do desafio envolve três eixos principais: (i) análise exploratória de dados (EDA) para identificar padrões, correlações e hipóteses iniciais; (ii) modelagem estatística e preditiva**,** com aplicação de algoritmos de regressão e avaliação de métricas de performance; e (iii) geração de insights acionáveis, capazes de subsidiar a tomada de decisão estratégica do estúdio.

A proposta enfatiza não apenas a implementação técnica de modelos de machine learning, mas também a capacidade de justificar metodologicamente cada decisão analítica, desde o pré-processamento dos dados até a escolha das métricas de avaliação. Dessa forma, este relatório busca apresentar uma análise consistente, transparente e orientada por evidências, demonstrando a aplicabilidade da estatística e da ciência de dados na resolução de problemas de negócio do setor cinematográfico.

2. Materiais

A base de dados utilizada neste estudo é composta por informações referentes a 999 produções cinematográficas, contemplando variáveis de natureza quantitativa, categórica e textual. O conjunto contém 16 variáveis, conforme descritas a seguir:

- **Identificação e metadados:**
    - **Series_Title:** título da obra, com 998 valores distintos, representando a identificação principal de cada registro.
    - **Released_Year:** ano de lançamento, armazenado como string, totalizando 100 valores únicos.
    - **Certificate:** classificação indicativa, com 16 categorias distintas e presença de 10% de valores ausentes.
    - **Runtime:** duração do filme, expressa em formato textual (“XXX min”), com 140 valores diferentes.
    - **Genre:** categorias de gênero, permitindo combinações múltiplas, totalizando 202 variações distintas.

- **Variável resposta e métricas de avaliação:**
    - **IMDB_Rating:** nota atribuída pelo IMDB, presente em todos os registros, variando entre 7,6 e 9,2, com média de 7,95.
    - **Meta_score:** escore crítico ponderado, disponível para 842 filmes, representando uma variável contínua com média aproximada de 77,9 e ausência em 157 casos.
    - **Gross:** indicador de faturamento, fornecido para 830 registros, expresso em formato monetário textual, necessitando padronização para valores numéricos.
- **Atributos textuais e qualitativos:**
    - **Overview:** sinopse descritiva, composta por 999 textos únicos, configurando uma variável não estruturada adequada a análises de linguagem natural.
    - **Director:** nome do diretor, com 548 valores distintos; Alfred Hitchcock é o mais recorrente, com 14 ocorrências.
    - **Star1 a Star4:** principais atores e atrizes, com alta diversidade (659 valores únicos em Star1, chegando a 938 em Star4).
- **Engajamento e popularidade:**
    - **No_of_Votes:** número de votos atribuídos pelos usuários, distribuído entre 25.088 e 2.303.232, com média aproximada de 271 mil votos e desvio-padrão elevado, evidenciando forte heterogeneidade entre os filmes.

Em termos gerais, a base apresenta boa completude para variáveis principais, porém, conta com lacunas relevantes em Certificate, Meta_score e Gross. Além disso, há a necessidade de transformação de variáveis armazenadas em formato textual (Runtime e Gross) para escalas numéricas, bem como a adequação das variáveis categóricas de alta cardinalidade, que impactam diretamente a etapa de modelagem.

3 – Metodologia

O projeto foi desenvolvido utilizando a plataforma Google Colab, que oferece uma interface interativa para execução de código Python, facilitando a análise de dados de maneira eficiente. A base de dados fornecida estava no formato CSV e foi hospedada no Google Drive, permitindo o acesso direto ao arquivo a partir do Colab através de um link compartilhado, o que eliminou a necessidade de realizar o upload do arquivo repetidamente. Esse processo garantiu que as análises e o processamento de dados fossem realizados de maneira contínua e eficiente.

Para a análise dos dados, foi utilizado Python com diversas bibliotecas que facilitaram a manipulação e visualização dos dados, além de ferramentas para a construção do modelo preditivo. As bibliotecas utilizadas foram:

- **Pandas**: Para a manipulação dos dados em formato tabular, carregando o arquivo CSV, limpando os dados e realizando transformações.
- **NumPy**: Essencial para operações matemáticas e manipulação de arrays numéricos, especialmente durante o pré-processamento e cálculo de métricas como a transformação logarítmica da variável "No_of_Votes".
- **Matplotlib**: Usada para a criação de gráficos e visualizações, como histogramas e gráficos de dispersão, facilitando a análise visual dos dados.
- **Seaborn**: Baseada no Matplotlib, foi utilizada para a criação de gráficos estatísticos mais complexos e visualmente atrativos, durante a análise exploratória dos dados.
- **Scikit-learn**: Essencial para a construção e avaliação de modelos preditivos, sendo usada para implementar o modelo de regressão linear, realizar a divisão dos dados em conjuntos de treino e teste e calcular métricas de performance, como o RMSE.
- **WordCloud**: Aplicada para a criação de nuvens de palavras, ajudando na análise da coluna "Overview" e na identificação das palavras mais frequentes nas sinopses dos filmes.
- **NLTK**: Biblioteca de processamento de linguagem natural, usada para remover palavras irrelevantes (stopwords) durante a análise de texto, aprimorando a qualidade da análise.
- **Joblib**: Utilizada para salvar o modelo preditivo em formato .pkl, o que possibilita o armazenamento e reutilização do modelo treinado para futuras previsões.

Em relação às exigências do desafio, a entrega do projeto foi organizada conforme as instruções fornecidas. O código foi hospedado em um repositório público no GitHub (https://github.com/MarcoCostaSilva/Indicium_DataScience/blob/main/README.md), garantindo a transparência e o fácil acesso ao código. Um arquivo README foi incluído no repositório com explicações detalhadas sobre como instalar e executar o projeto, incluindo as instruções para configurar o ambiente de execução e instalar os pré-requisitos necessários. O arquivo de requisitos foi gerado para listar todos os pacotes utilizados no projeto, juntamente com suas versões, garantindo que o ambiente de execução fosse reproduzível em outras máquinas, sem problemas relacionados a versões de dependências. O relatório das análises estatísticas e da análise exploratória de dados (EDA) foi gerado no formato Jupyter Notebook, conforme solicitado. Esse notebook contém todas as etapas realizadas na análise dos dados, incluindo visualizações e hipóteses geradas durante o processo.

O código de modelagem utilizado para a construção e avaliação do modelo preditivo foi incluído no mesmo Jupyter Notebook. O modelo de regressão linear foi implementado para prever a nota do IMDB com base nas variáveis selecionadas, e a performance do modelo foi avaliada utilizando o RMSE. O arquivo .pkl do modelo treinado foi gerado e salvo utilizando a biblioteca Joblib, como solicitado, permitindo que o modelo seja carregado posteriormente e utilizado para previsões sem a necessidade de re-treinamento.

Todos os códigos produzidos seguiram boas práticas de codificação, garantindo clareza, eficiência e legibilidade. A estrutura do código foi modular, com comentários explicativos em todas as etapas, facilitando a compreensão e a manutenção futura do projeto. A documentação foi mantida consistente e detalhada, atendendo a todas as exigências do desafio.

4 – Resultados

4.1 – Análise Exploratória dos Dados

A análise exploratória de dados começa com o carregamento do conjunto de dados, que está em formato CSV e hospedado no Google Drive, utilizando a biblioteca pandas para manipulação eficiente das tabelas.

```
# Importando biblioteca:
import pandas as pd

# Link:
url = "https://docs.google.com/spreadsheets/d/1JB-RL9C8Xz6toUWTNhMpaWeLlYvva8qU/export?format=xlsx"

# Carregando os dados:
dados = pd.read_excel(url)

# Exibindo as 5 primeiras linhas:
print("Visualização inicial dos dados:")
print(dados.head())
```

```
Visualização inicial dos dados:
   Unnamed: 0                                      Series_Title Released_Year  \
0           1                                     The Godfather          1972
1           2                                   The Dark Knight          2008
2           3                            The Godfather: Part II          1974
3           4                                      12 Angry Men          1957
4           5  The Lord of the Rings: The Return of the King          2003

  Certificate  Runtime                     Genre  IMDB_Rating  \
0           A  175 min              Crime, Drama          9.2
1          UA  152 min      Action, Crime, Drama          9.0
2           A  202 min              Crime, Drama          9.0
3           U   96 min              Crime, Drama          9.0
4           U  201 min  Action, Adventure, Drama          8.9

                                            Overview  Meta_score  \
0  An organized crime dynasty's aging patriarch t...       100.0
1  When the menace known as the Joker wreaks havo...        84.0
2  The early life and career of Vito Corleone in ...        90.0
3  A jury holdout attempts to prevent a miscarria...        96.0
4  Gandalf and Aragorn lead the World of Men agai...        94.0

               Director           Star1            Star2          Star3  \
0  Francis Ford Coppola   Marlon Brando       Al Pacino     James Caan
1     Christopher Nolan  Christian Bale    Heath Ledger  Aaron Eckhart
2  Francis Ford Coppola       Al Pacino  Robert De Niro  Robert Duvall
3          Sidney Lumet     Henry Fonda      Lee J. Cobb  Martin Balsam
4         Peter Jackson     Elijah Wood  Viggo Mortensen   Ian McKellen

           Star4  No_of_Votes        Gross
0   Diane Keaton      1620367  134,966,411
1  Michael Caine      2303232  534,858,444
2   Diane Keaton      1129952   57,300,000
3   John Fiedler       689845    4,360,000
4  Orlando Bloom      1642758  377,845,905
```

A análise inicial das primeiras linhas do conjunto de dados mostra informações sobre os filmes, incluindo título, ano de lançamento, certificação indicativa, duração, gênero, avaliação no IMDb, resumo da história, metascore, diretor, elenco principal, número de votos e faturamento bruto.

É possível perceber que os dados têm colunas numéricas, como IMDB_Rating, Meta_score, No_of_Votes e Gross, colunas categóricas, como Certificate e Genre, e colunas textuais, como Overview e nomes de diretores e atores. Essa variedade permite diferentes tipos de análises, por exemplo:

- Comparar gêneros com as notas atribuídas pelos usuários
- Verificar a relação entre número de votos e faturamento
- Observar padrões nos resumos dos filmes na coluna Overview

Esta visualização inicial ajuda a entender a composição do conjunto de dados e a decidir quais colunas podem ser usadas em análises futuras, além de indicar a necessidade de tratar valores ausentes ou normalizar algumas variáveis.

A seguir, verificamos os tipos de colunas e a presença de valores ausentes, etapa essencial para compreender a qualidade dos dados e definir estratégias de pré-processamento.

```
# Verificando os tipos de colunas e valores nulos:
print("Tipos de colunas e quantidade de valores nulos em cada coluna:")
print(dados.info())

# Contando quantos valores nulos existem em cada coluna:
print("\nNúmero de valores nulos por coluna:")
print(dados.isnull().sum())
```

```
Tipos de colunas e quantidade de valores nulos em cada coluna:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999 entries, 0 to 998
Data columns (total 16 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unnamed: 0      999 non-null    int64
 1   Series_Title    999 non-null    object
 2   Released_Year   999 non-null    object
 3   Certificate     898 non-null    object
 4   Runtime         999 non-null    object
 5   Genre           999 non-null    object
 6   IMDB_Rating     999 non-null    float64
 7   Overview        999 non-null    object
 8   Meta_score      842 non-null    float64
 9   Director        999 non-null    object
 10  Star1           999 non-null    object
 11  Star2           999 non-null    object
 12  Star3           999 non-null    object
 13  Star4           999 non-null    object
 14  No_of_Votes     999 non-null    int64
 15  Gross           830 non-null    object
dtypes: float64(2), int64(2), object(12)
memory usage: 125.0+ KB
None
```

```
Número de valores nulos por coluna:
Unnamed: 0         0
Series_Title       0
Released_Year      0
Certificate      101
Runtime            0
Genre              0
IMDB_Rating        0
Overview           0
Meta_score       157
Director           0
Star1              0
Star2              0
Star3              0
Star4              0
No_of_Votes        0
Gross            169
dtype: int64
```

Observa-se que o conjunto de dados possui 16 colunas, sendo a maioria do tipo textual (object), algumas numéricas (float64) e duas do tipo inteiro (int64). Algumas colunas apresentam valores ausentes, sendo Certificate com 101 valores nulos, Meta_score com 157 e Gross com 169. Colunas como IMDB_Rating, No_of_Votes e Meta_score são importantes para análises quantitativas, enquanto Genre, Director e Star1 a Star4 permitem análises categóricas. Essa inspeção inicial é fundamental para orientar o pré-processamento e garantir que as análises subsequentes sejam realizadas com dados consistentes e completos.

Para entender melhor os dados numéricos, analisamos estatísticas básicas de cada coluna, como média, mediana, mínimo, máximo e quartis. Isso permite identificar padrões gerais e possíveis valores atípicos.

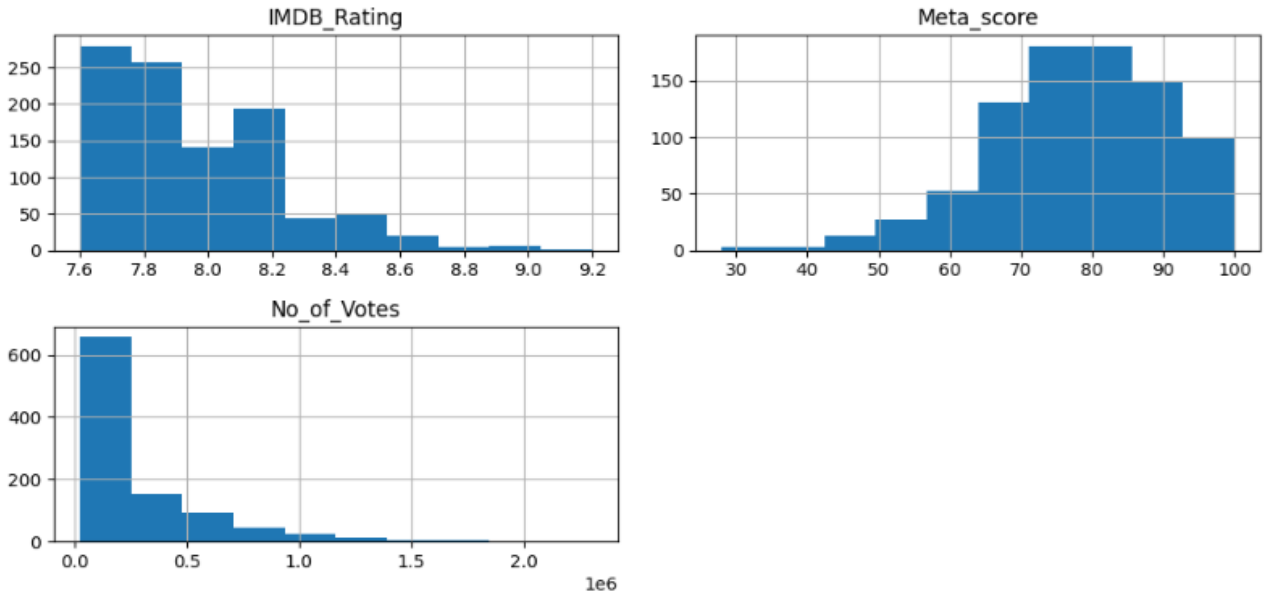
```
# Estatísticas básicas das colunas numéricas
estatisticas_numericas = dados.describe()
print(estatisticas_numericas)
```

```
       Unnamed: 0  IMDB_Rating  Meta_score   No_of_Votes
count  999.000000   999.000000  842.000000  9.990000e+02
mean   500.000000     7.947948   77.969121  2.716214e+05
std    288.530761     0.272290   12.383257  3.209126e+05
min      1.000000     7.600000   28.000000  2.508800e+04
25%    250.500000     7.700000   70.000000  5.547150e+04
50%    500.000000     7.900000   79.000000  1.383560e+05
75%    749.500000     8.100000   87.000000  3.731675e+05
max    999.000000     9.200000  100.000000  2.303232e+06
```

No conjunto de dados, observa-se que a coluna Unnamed: 0 funciona apenas como índice e não traz informação relevante para a análise. As notas do IMDB_Rating variam entre 7.6 e 9.2, com média próxima de 7.95, indicando que a maioria dos filmes tem avaliações altas. O Meta_score apresenta alguns valores faltantes, com apenas 842 registros disponíveis, e varia de 28 a 100, com média em torno de 78. O número de votos (No_of_Votes) apresenta grande variação, de aproximadamente 25 mil a mais de 2 milhões, mostrando que alguns filmes são muito mais populares que outros. Essas estatísticas iniciais ajudam a compreender a distribuição geral dos dados e a identificar possíveis valores extremos.

Para entender melhor a distribuição das colunas numéricas e identificar padrões e tendências, optou-se por utilizar histogramas. Essa visualização permite observar a frequência dos valores em diferentes intervalos, facilitando a identificação de concentração de dados e valores extremos.

```
# Importando biblioteca de visualização:
import matplotlib.pyplot as plt

# Selecionando apenas as colunas numéricas relevantes:
colunas_numericas = ['IMDB_Rating', 'Meta_score', 'No_of_Votes']

# Criando histogramas para cada coluna numérica:
dados[colunas_numericas].hist(bins=10, figsize=(10,5))
plt.tight_layout()
plt.show()
```



As análises observadas nos histogramas são as seguintes:

A coluna **IMDB_Rating** apresenta uma distribuição estreita, concentrada entre aproximadamente 7.6 e 8.3, com poucos filmes acima de 8.5. Isso indica baixa variabilidade nas avaliações, ou seja, a maioria dos filmes possui notas relativamente altas e próximas entre si. Para modelagem, isso significa que prever pequenas diferenças nas notas pode ser mais difícil, devido a pouca dispersão da variável.

O **Meta_score** varia de cerca de 28 a 100 e se concentra principalmente entre 70 e 90. Há mais variabilidade em relação ao IMDB_Rating e alguns valores ausentes. Como se trata de uma medida de crítica especializada, o Meta_score pode ser um bom preditor numérico da nota do IMDB, desde que os valores ausentes sejam tratados adequadamente.

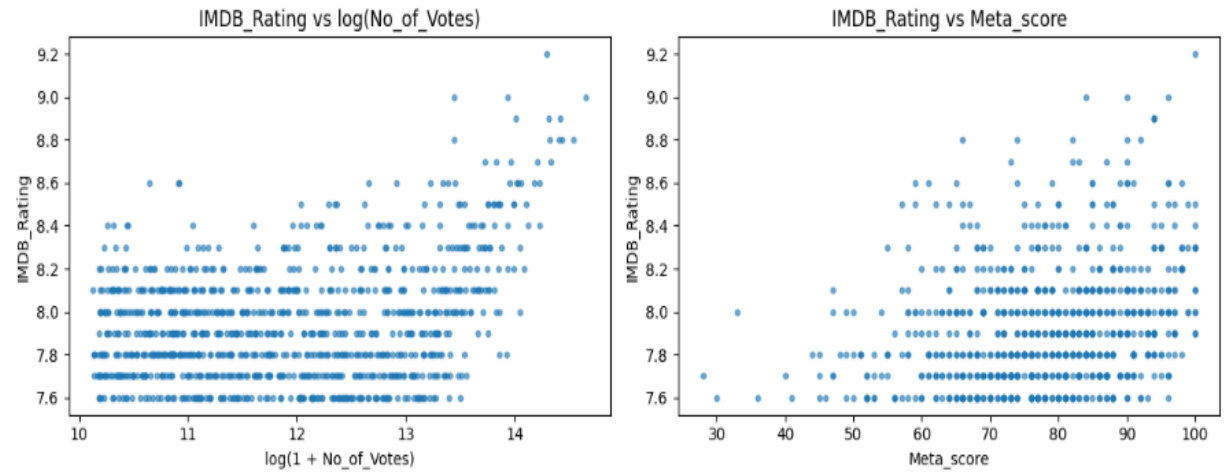
O **número de votos (No_of_Votes)** apresenta forte assimetria à direita, com muitos filmes recebendo relativamente poucos votos e poucos filmes com centenas de milhares. Essa distribuição assimétrica justifica a transformação logarítmica antes de utilizar essa variável em análises ou modelos, para reduzir a influência de outliers e facilitar a visualização.

Para avaliar a relação entre a popularidade dos filmes, medida pelo número de votos, e a qualidade percebida pelo público, medida pelo IMDB_Rating, foi aplicada uma transformação logarítmica na coluna No_of_Votes. Essa transformação é necessária porque o número de votos apresenta uma distribuição muito assimétrica, com alguns filmes recebendo dezenas ou centenas de milhares de votos e outros apenas algumas dezenas de milhares. O logaritmo ajuda a reduzir o efeito desses valores extremos, tornando a análise mais equilibrada e a visualização mais clara.

Para observar a relação entre as variáveis, foram utilizados gráficos de dispersão. Esse tipo de gráfico é apropriado quando se quer analisar a associação entre duas variáveis numéricas, pois permite visualizar tendências, concentrações e possíveis outliers de forma intuitiva. Foram plotados gráficos de dispersão entre log_votes e IMDB_Rating, e entre Meta_score e IMDB_Rating.

```
#Importação das bibliotecas:
import numpy as np
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Criar coluna com log(1 + No_of_Votes):
dados['log_votes'] = np.log1p(dados['No_of_Votes'])

# Plots de dispersão lado a lado:
plt.figure(figsize=(12,4))

plt.subplot(1,2,1)
plt.scatter(dados['log_votes'], dados['IMDB_Rating'], s=10, alpha=0.6)
plt.xlabel('log(1 + No_of_Votes)')
plt.ylabel('IMDB_Rating')
plt.title('IMDB_Rating vs log(No_of_Votes)')

plt.subplot(1,2,2)
plt.scatter(dados['Meta_score'], dados['IMDB_Rating'], s=10, alpha=0.6)
plt.xlabel('Meta_score')
plt.ylabel('IMDB_Rating')
plt.title('IMDB_Rating vs Meta_score')

plt.tight_layout()
plt.show()

# Calcular correlações de Pearson (apenas pares com dados não-nulos):
corr_votes = dados[['IMDB_Rating',
'log_votes']].dropna().corr().loc['IMDB_Rating','log_votes']
corr_meta = dados[['IMDB_Rating',
'Meta_score']].dropna().corr().loc['IMDB_Rating','Meta_score']

print(f"Correlação Pearson IMDB_Rating x log_votes: {corr_votes:.3f}")
print(f"Correlação Pearson IMDB_Rating x Meta_score: {corr_meta:.3f}")
```




```
Correlação Pearson IMDB_Rating x log_votes: 0.318
Correlação Pearson IMDB_Rating x Meta_score: 0.271
```

A análise visual mostra uma tendência positiva moderada. Filmes com maior número de votos tendem a receber notas mais altas no IMDB, e filmes com Meta_score mais elevado também apresentam avaliações maiores.

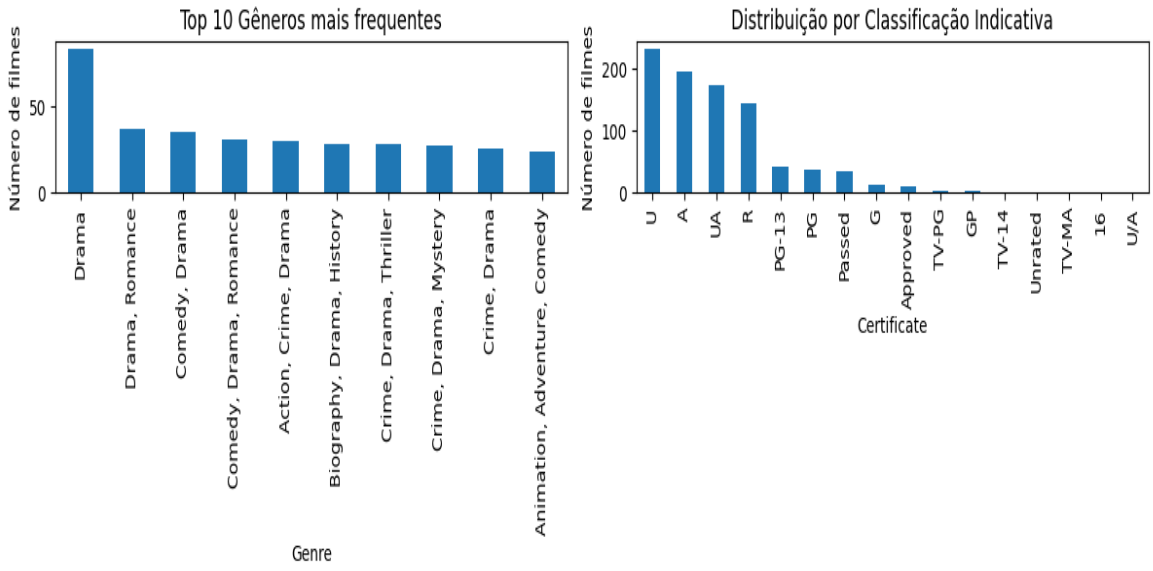
As correlações de Pearson confirmam essa observação. A correlação entre IMDB_Rating e log_votes é 0.318, indicando uma associação positiva moderada entre popularidade e avaliação do público. A correlação entre IMDB_Rating e Meta_score é 0.271, mostrando uma associação positiva um pouco mais fraca entre a avaliação da crítica e a avaliação do público.

Em resumo, tanto a popularidade quanto a crítica especializada parecem influenciar as notas do IMDB, mas nenhuma das duas variáveis explica completamente a variação nas avaliações dos filmes.

Para entender o papel das variáveis categóricas, foi analisada a distribuição da quantidade de filmes por gênero e por classificação indicativa (Certificate). Essa análise é importante porque as variáveis categóricas representam grupos ou categorias distintas, e a contagem de frequências é uma forma simples e eficaz de identificar quais categorias predominam no conjunto de dados. A visualização por meio de gráficos de barras ou tabelas de frequência permite observar rapidamente concentrações, padrões e diferenças entre categorias.

```
# Contagem de filmes por Gênero e por Classificação Indicativa:
import matplotlib.pyplot as plt

# Contagem de cada categoria:
contagem_genero = dados['Genre'].value_counts().head(10)   # 10 gêneros mais comuns
contagem_certificado = dados['Certificate'].value_counts()

# Plots:
plt.figure(figsize=(12,4))

plt.subplot(1,2,1)
contagem_genero.plot(kind='bar')
plt.title("Top 10 Gêneros mais frequentes")
plt.ylabel("Número de filmes")

plt.subplot(1,2,2)
contagem_certificado.plot(kind='bar')
```

```
plt.title("Distribuição por Classificação Indicativa")
plt.ylabel("Número de filmes")

plt.tight_layout()
plt.show()

# Mostrar tabelas:
print("Top 10 gêneros:")
print(contagem_genero)

print("\nDistribuição por classificação indicativa:")
print(contagem_certificado)
```




```
Genre
Top 10 gêneros:
Genre
Drama                                84
Drama, Romance                       37
Comedy, Drama                        35
Comedy, Drama, Romance               31
Action, Crime, Drama                 30
Biography, Drama, History            28
Crime, Drama, Thriller               28
Crime, Drama, Mystery                27
Crime, Drama                         26
Animation, Adventure, Comedy         24
Name: count, dtype: int64

Distribuição por classificação indicativa:
Certificate
U            234
A            196
UA           175
R            146
PG-13         43
PG            37
Passed        34
G             12
Approved      11
TV-PG          3
GP             2
TV-14          1
Unrated        1
TV-MA          1
16             1
U/A            1
Name: count, dtype: int64
```

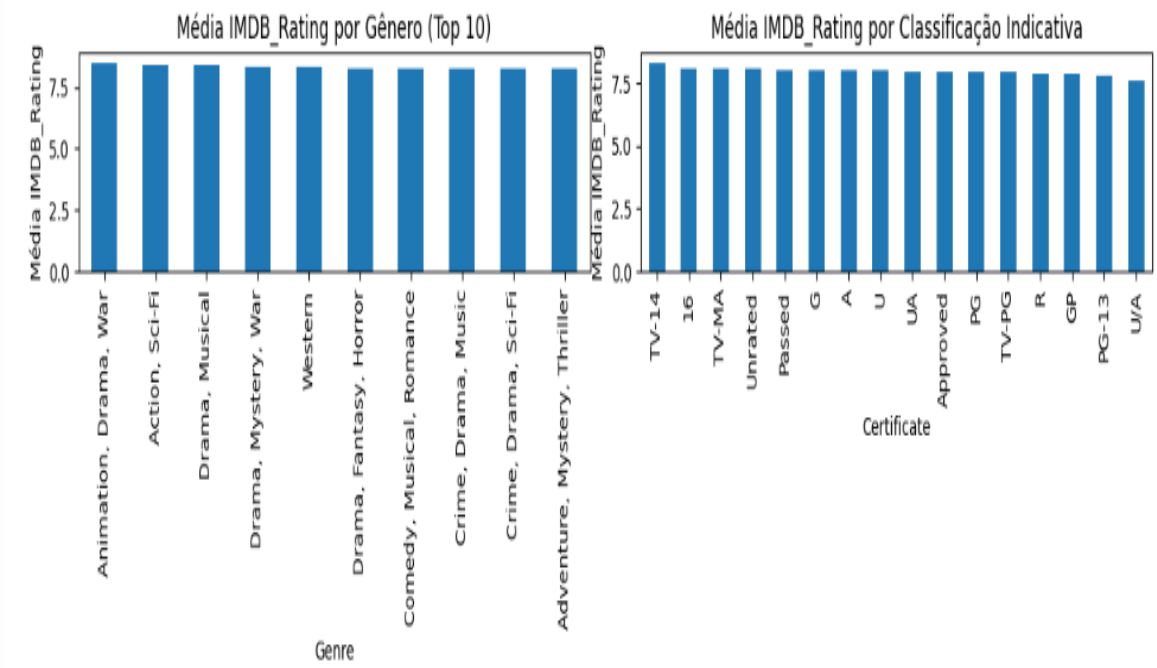
No caso dos gêneros, o gênero Drama é o mais frequente, com 84 filmes, seguido por combinações como Drama e Romance, Comedy e Drama, e Comedy, Drama e Romance. Isso indica que a maior parte do dataset é composta por dramas puros ou misturados com outros gêneros, como romance, comédia e crime, mostrando uma certa concentração temática.

Em relação à classificação indicativa, as categorias mais comuns são U, com 234 filmes, A com 196, UA com 175 e R com 146, abrangendo a maior parte do acervo. As demais classificações aparecem com frequência bem menor, o que pode limitar a realização de análises mais detalhadas para essas categorias.

Para investigar se determinados gêneros e classificações indicativas estão associados a melhores avaliações, foi calculada a média do IMDB_Rating por gênero e por classificação indicativa. A escolha da média como medida estatística se justifica por sua capacidade de representar de forma resumida a tendência central das avaliações dentro de cada categoria. Para visualizar esses resultados, foram utilizados gráficos de barras, que são apropriados para comparar valores médios entre diferentes grupos categóricos, facilitando a identificação de padrões e diferenças.

```
import matplotlib.pyplot as plt

# Média do IMDB_Rating por gênero (10 mais frequentes):
media_genero =
dados.groupby('Genre')['IMDB_Rating'].mean().sort_values(ascending=False).head(10)

# Média do IMDB_Rating por classificação indicativa:
media_certificado =
dados.groupby('Certificate')['IMDB_Rating'].mean().sort_values(ascending=False)

# Plots:
plt.figure(figsize=(12,4))

plt.subplot(1,2,1)
media_genero.plot(kind='bar')
plt.title('Média IMDB_Rating por Gênero (Top 10)')
plt.ylabel('Média IMDB_Rating')

plt.subplot(1,2,2)
media_certificado.plot(kind='bar')
plt.title('Média IMDB_Rating por Classificação Indicativa')
```

```
plt.ylabel('Média IMDB_Rating')

plt.tight_layout()
plt.show()
```



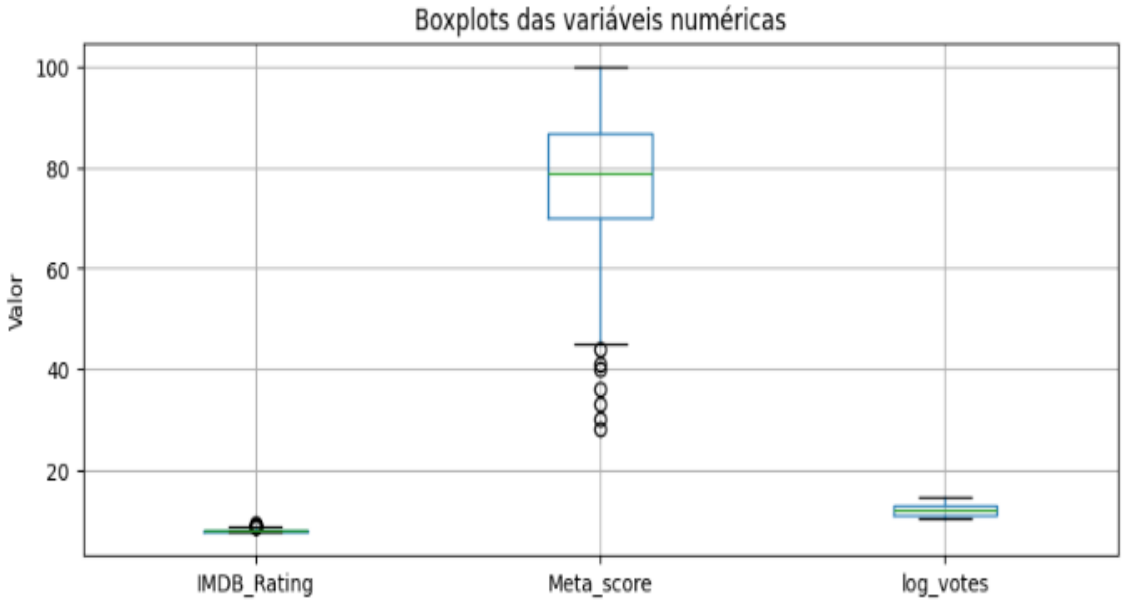
Os gráficos mostram que as médias de IMDB_Rating variam pouco entre os diferentes gêneros e classificações indicativas. Alguns gêneros específicos, como “Animation, Drama, War” e “Action, Sci-Fi”, apresentam médias ligeiramente mais altas, mas, de modo geral, os valores ficam bastante próximos. O mesmo ocorre para as classificações indicativas, sem diferença clara entre faixas etárias distintas.

Isso sugere que a percepção de qualidade, medida pelo IMDB_Rating, é relativamente estável independentemente do gênero ou da classificação indicativa, indicando que outros fatores, como popularidade ou crítica especializada, podem ter maior influência nas avaliações.

Para identificar a presença de valores atípicos no conjunto de dados, foram construídos boxplots das principais variáveis numéricas: IMDB_Rating, Meta_score e log_votes. A escolha do boxplot se justifica por ser uma ferramenta estatística adequada para representar a mediana, os quartis e os limites da distribuição, destacando de forma

clara a presença de outliers, ou seja, observações muito acima ou abaixo da tendência central.

```
import matplotlib.pyplot as plt

# Selecionar variáveis numéricas:
variaveis = ['IMDB_Rating', 'Meta_score', 'log_votes']

# Criar boxplots:
plt.figure(figsize=(10,4))
dados[variaveis].boxplot()
plt.title("Boxplots das variáveis numéricas")
plt.ylabel("Valor")
plt.show()
```



A análise mostra que o IMDB_Rating apresenta distribuição bastante concentrada entre 7.5 e 9, com poucos outliers em valores mais baixos. O Meta_score, por sua vez, apresenta maior dispersão e exibe diversos outliers, especialmente abaixo de 40, sugerindo filmes mal avaliados pela crítica em comparação ao restante do conjunto. Já a variável log_votes evidencia alguns títulos com valores bem acima da mediana, representando filmes extremamente populares em relação à maioria. Esses resultados indicam que, embora as notas do público sejam relativamente homogêneas, tanto a crítica especializada quanto a popularidade apresentam maior variabilidade, com casos que se destacam de maneira significativa em relação ao padrão geral.

Para aprofundar a análise dos boxplots, foi realizada a identificação dos filmes que se destacam como outliers nas distribuições de IMDB_Rating, Meta_score e log_votes. A detecção foi feita utilizando a regra do IQR (Intervalo Interquartil), que é um método estatístico amplamente adotado por sua robustez, pois define como valores atípicos aqueles que estão além de 1,5 vezes a distância interquartil. Esse procedimento é útil para distinguir se os outliers representam erros de registro ou casos especiais que merecem interpretação diferenciada.

```
# Função para detectar outliers usando regra do IQR:
def detectar_outliers(coluna):
    Q1 = dados[coluna].quantile(0.25)
    Q3 = dados[coluna].quantile(0.75)
    IQR = Q3 - Q1
    limite_inferior = Q1 - 1.5 * IQR
    limite_superior = Q3 + 1.5 * IQR
    return dados[(dados[coluna] < limite_inferior) | (dados[coluna] > limite_superior)]

# Detectar outliers em cada variável:
outliers_rating = detectar_outliers('IMDB_Rating')
outliers_meta = detectar_outliers('Meta_score')
outliers_votes = detectar_outliers('log_votes')

print("Outliers IMDB_Rating:")
print(outliers_rating[['Series_Title','IMDB_Rating']].head())

print("\nOutliers Meta_score:")
print(outliers_meta[['Series_Title','Meta_score']].head())

print("\nOutliers log_votes:")
print(outliers_votes[['Series_Title','No_of_Votes','log_votes']].head())
```

```
Outliers IMDB_Rating:
                                     Series_Title  IMDB_Rating
0                                   The Godfather          9.2
1                                 The Dark Knight          9.0
2                        The Godfather: Part II            9.0
3                                    12 Angry Men          9.0
4  The Lord of the Rings: The Return of the King           8.9

Outliers Meta_score:
             Series_Title  Meta_score
355        Tropa de Elite        33.0
647  The Boondock Saints         44.0
734           Kai po che!        40.0
787              I Am Sam        28.0
916          Seven Pounds        36.0

Outliers log_votes:
Empty DataFrame
Columns: [Series_Title, No_of_Votes, log_votes]
Index: []
```

Os resultados mostram que, para o IMDB_Rating, os outliers correspondem a filmes consagrados com avaliações extremamente altas, como The Godfather e The Dark Knight. Nesse caso, os valores não indicam inconsistências, mas sim títulos que se destacam como referência em qualidade segundo o público. Já no Meta_score, os outliers são filmes com notas muito baixas da crítica especializada, como Tropa de Elite (33) e I Am Sam (28), evidenciando discrepâncias relevantes entre a avaliação da crítica e do público. Para log_votes, nenhum outlier foi detectado após a transformação logarítmica, o que confirma que essa técnica conseguiu reduzir a assimetria e tornar a variável mais homogênea para análises posteriores.

Assim, a detecção de outliers não apenas destaca títulos que se diferenciam dos demais, mas também reforça a importância de interpretar cuidadosamente esses pontos extremos, que podem indicar tanto fenômenos legítimos quanto potenciais distorções nos dados.

4.1.1 – Principais Achados da EDA

```
print("Resumo da Análise Exploratória de Dados (EDA):\n")

print("1. Distribuição das variáveis numéricas:")
print(dados[['IMDB_Rating','Meta_score','log_votes']].describe(), "\n")

print("2. Correlação entre variáveis numéricas:")
print(dados[['IMDB_Rating','Meta_score','log_votes']].corr(), "\n")

print("3. Gêneros mais frequentes:")
print(dados['Genre'].value_counts().head(5), "\n")

print("4. Classificação indicativa mais frequente:")
print(dados['Certificate'].value_counts().head(5), "\n")

print("5. Exemplos de outliers:")
print("Notas muito altas:", outliers_rating['Series_Title'].tolist()[:3])
print("Notas da crítica muito baixas:", outliers_meta['Series_Title'].tolist()[:3])
```

```
Resumo da Análise Exploratória de Dados (EDA):

1. Distribuição das variáveis numéricas:
       IMDB_Rating  Meta_score   log_votes
count   999.000000  842.000000  999.000000
mean      7.947948   77.969121   11.904169
std       0.272290   12.383257    1.117715
min       7.600000   28.000000   10.130185
25%       7.700000   70.000000   10.923641
50%       7.900000   79.000000   11.837593
75%       8.100000   87.000000   12.829784
max       9.200000  100.000000   14.649824
```

A etapa de síntese da Análise Exploratória de Dados teve como objetivo consolidar os principais achados e destacar os padrões mais relevantes do conjunto de filmes analisado. A descrição estatística das variáveis numéricas IMDB_Rating, Meta_score e log_votes mostrou distribuições relativamente concentradas, com a avaliação média do público em torno de 7,9 pontos, enquanto as notas da crítica especializada variaram mais amplamente, incluindo valores extremos baixos. A transformação logarítmica aplicada ao número de votos reduziu a assimetria dessa variável, permitindo uma distribuição mais homogênea e facilitando a interpretação comparativa.

A análise de correlação evidenciou uma associação positiva moderada entre IMDB_Rating e log_votes (0,318), indicando que filmes mais bem avaliados pelo público tendem também a receber maior volume de votos. Já a relação entre IMDB_Rating e Meta_score foi mais fraca (0,271), sugerindo que há divergências na forma como público e crítica avaliam as obras. Esses cálculos de correlação foram escolhidos porque permitem quantificar o grau de associação linear entre variáveis numéricas, fornecendo um panorama objetivo das dependências.

Nas variáveis categóricas, verificou-se que o gênero Drama foi o mais frequente, tanto de forma isolada quanto em combinações com Romance, Comédia e Crime. Essa predominância aponta para uma concentração temática no dataset. Quanto às classificações indicativas, os certificados U, A e UA foram os mais comuns, representando a maior parte do acervo e indicando uma predominância de filmes direcionados ao público geral, enquanto os títulos classificados como R aparecem em menor proporção, voltados a faixas

etárias mais restritas. O uso de contagens e frequências relativas foi essencial aqui para revelar a composição do dataset e dar contexto à distribuição das categorias.

A análise de outliers complementou o estudo ao destacar títulos que se distanciam significativamente das distribuições centrais. No caso do IMDB_Rating, apareceram filmes consagrados, como The Godfather e The Dark Knight, cujas notas muito elevadas refletem o reconhecimento mundial e não erros de registro. Já para o Meta_score, títulos como Tropa de Elite e I Am Sam apresentaram avaliações muito baixas da crítica, chamando atenção para divergências marcantes entre especialistas e público. A detecção de outliers via regra do IQR foi a metodologia escolhida por ser um procedimento estatístico robusto e amplamente aceito para identificar valores extremos de forma objetiva.

Em resumo, a Análise Exploratória de Dados permitiu identificar padrões importantes na distribuição das notas, nas correlações entre popularidade e avaliações, nas categorias mais recorrentes e nos casos de comportamento extremo. Esses resultados fornecem uma visão clara e integrada do dataset e estabelecem uma base sólida para etapas futuras deste projeto.

4.2 – Respostas as Perguntas Estratégicas

4.2.1 – Pergunta 1 – Qual filme recomendar para uma pessoa que você não conhece?

A formulação da recomendação de um filme para uma pessoa desconhecida exige um critério que maximize a chance de acerto independentemente de preferências individuais. Nesse contexto, a estratégia adotada foi priorizar títulos que apresentam simultaneamente alta avaliação pelo público (IMDB_Rating) e grande popularidade, medida pelo número de votos. A justificativa para essa escolha é que, quanto maior o consenso em torno da qualidade de um filme e mais amplo for o público que o avaliou positivamente, maior a probabilidade de que a indicação seja bem recebida.

Para operacionalizar essa análise, foi criada uma métrica composta pela ordenação conjunta de duas variáveis: a nota média atribuída no IMDB e a transformação logarítmica do número de votos (log_votes). O uso da escala logarítmica é fundamental, pois o número

de votos apresenta distribuição altamente assimétrica, e a transformação suaviza essa discrepância, permitindo comparações mais equilibradas entre títulos de diferentes magnitudes de popularidade. A ordenação foi feita de forma decrescente, priorizando primeiro as notas do público e, em seguida, a popularidade ajustada.

```
import numpy as np

# Garante a coluna de popularidade em escala log:
if 'log_votes' not in dados.columns:
    dados['log_votes'] = np.log1p(dados['No_of_Votes'])

# Ranking de recomendação geral (qualidade + popularidade):
top_filmes = (
    dados[['Series_Title', 'IMDB_Rating', 'No_of_Votes', 'log_votes']]
    .dropna(subset=['IMDB_Rating', 'No_of_Votes'])
    .sort_values(by=['IMDB_Rating', 'log_votes'], ascending=[False, False])
    .head(10)
)

print(top_filmes.to_string(index=False))
```

```
                                  Series_Title  IMDB_Rating  No_of_Votes  log_votes
                                  The Godfather          9.2      1620367  14.298164
                                The Dark Knight          9.0      2303232  14.649824
                      The Godfather: Part II             9.0      1129952  13.937687
                                   12 Angry Men          9.0       689845  13.444224
                                   Pulp Fiction          8.9      1826188  14.417742
The Lord of the Rings: The Return of the King            8.9      1642758  14.311888
                                Schindler's List         8.9      1213505  14.009024
                                       Inception         8.8      2067042  14.541630
                                     Fight Club          8.8      1854740  14.433256
                                    Forrest Gump         8.8      1809221  14.408407
```

O ranking resultante evidenciou como principais candidatos a recomendação geral filmes clássicos amplamente reconhecidos, como The Godfather, The Dark Knight, The Godfather: Part II, Pulp Fiction, Inception e Forrest Gump. Todos apresentam notas muito altas (IMDB_Rating ≥ 8.8) e um volume expressivo de votos, indicando tanto qualidade percebida quanto alcance internacional. A presença de obras de diferentes gêneros e épocas também contribui para a robustez da recomendação, já que amplia as chances de atender a públicos variados.

Em síntese, quando não há informações sobre o perfil da pessoa a ser atendida, a recomendação baseada na combinação entre qualidade média elevada e grande popularidade é uma abordagem estatisticamente consistente. Ela garante que os filmes indicados representem escolhas seguras, pois reúnem consenso positivo de grandes audiências e reconhecimento crítico, minimizando o risco de insatisfação na ausência de preferências declaradas.

4.2.2 – Pergunta 2 – Quais fatores influenciam o faturamento esperado de um filme?

Para investigar os fatores associados ao faturamento de um filme, analisou-se a correlação entre a variável Gross (receita de bilheteria) e três indicadores numéricos disponíveis: IMDB_Rating, Meta_score e log_votes. O objetivo foi identificar quais características se mostram mais relacionadas ao desempenho financeiro, fornecendo evidências quantitativas sobre os principais determinantes da receita.

A variável Gross foi inicialmente tratada para garantir consistência numérica, com a remoção de vírgulas e conversão para o tipo float. Em seguida, foram selecionadas apenas as observações sem valores ausentes, permitindo a aplicação da correlação de Pearson, um método adequado para mensurar relações lineares entre variáveis contínuas.

```
# Converter a coluna Gross para numérica (remover vírgulas e transformar em float):
dados['Gross'] = dados['Gross'].astype(str).str.replace(',', '', regex=False)
dados['Gross'] = pd.to_numeric(dados['Gross'], errors='coerce')

# Selecionar colunas relevantes e remover nulos:
dados_faturamento = dados[['Gross', 'IMDB_Rating', 'Meta_score', 'log_votes']].dropna()

# Calcular correlações de Pearson:
correlacoes = dados_faturamento.corr(method='pearson')

print("Correlação entre faturamento (Gross) e variáveis explicativas:")
print(correlacoes['Gross'].sort_values(ascending=False))
```

```
Correlação entre faturamento (Gross) e variáveis explicativas:
Gross          1.000000
log_votes      0.543543
IMDB_Rating    0.132445
Meta_score    -0.030480
Name: Gross, dtype: float64
```

Os resultados mostraram que a popularidade do filme, representada pela variável log_votes, é o fator mais fortemente associado ao faturamento, com correlação de 0,54. Esse achado indica que títulos amplamente comentados e avaliados pelo público tendem a gerar maior receita, o que é consistente com a ideia de que visibilidade e engajamento são determinantes centrais no sucesso financeiro. Já a avaliação do público no IMDB (IMDB_Rating) apresentou correlação fraca e positiva (0,13), sugerindo que a qualidade percebida pode contribuir, mas de forma limitada. Por outro lado, a nota da crítica especializada (Meta_score) não mostrou relação significativa com o faturamento (-0,03), indicando que o prestígio crítico não impacta diretamente o desempenho em bilheteria.

A escolha da correlação de Pearson justifica-se pela necessidade de medir de maneira objetiva a intensidade e direção das relações entre as variáveis numéricas, permitindo comparações diretas entre diferentes fatores. Essa análise também foi complementada pelo uso da escala logarítmica em No_of_Votes, que corrige a forte assimetria dos dados de popularidade, viabilizando uma avaliação mais robusta da associação com o faturamento.

Em síntese, a análise indica que a popularidade é o principal fator preditivo de receita, enquanto a qualidade percebida pelo público exerce papel secundário e a crítica praticamente não influencia diretamente o resultado financeiro. Esse achado reforça a importância do alcance e engajamento massivo como motores centrais da indústria cinematográfica.

4.2.3 – Pergunta 3 – O que a coluna Overview revela sobre o gênero ou características do filme?

Para compreender o conteúdo textual da coluna Overview, que contém as sinopses dos filmes, foi aplicada uma análise exploratória de texto. O objetivo é identificar padrões narrativos e temáticos recorrentes que possam indicar características comuns entre os gêneros ou elementos centrais das histórias. O primeiro passo consistiu na geração de uma nuvem de palavras (WordCloud), uma técnica de visualização amplamente utilizada para destacar os termos mais frequentes em grandes volumes de texto. Essa abordagem é útil como diagnóstico inicial porque permite observar rapidamente quais palavras dominam as descrições, fornecendo uma visão intuitiva das narrativas mais presentes no dataset.

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Juntar todos os textos da coluna Overview em uma única string:
texto_overview = " ".join(dados["Overview"].dropna().astype(str))

# Gerar a nuvem de palavras:
wordcloud = WordCloud(width=800, height=400,
background_color="white").generate(texto_overview)

# Mostrar a nuvem:
plt.figure(figsize=(12,6))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("Palavras mais frequentes nas sinopses (Overview)")
plt.show()
```

```python
# Contagem das palavras mais frequentes nas sinopses (Overview)

from collections import Counter
import re

# Juntar todos os textos da coluna Overview em uma string:
texto = " ".join(dados["Overview"].dropna().astype(str))

# Remover caracteres especiais e transformar em minúsculas:
palavras = re.findall(r"\b\w+\b", texto.lower())

# Contar as palavras:
contagem = Counter(palavras)

# Pegar as 20 palavras mais comuns:
top_palavras = contagem.most_common(20)

# Transformar em DataFrame para melhor visualização:
import pandas as pd
df_top_palavras = pd.DataFrame(top_palavras, columns=["Palavra", "Frequência"])

print("Top 20 palavras mais frequentes nas sinopses (Overview):")
print(df_top_palavras)
```

```
Top 20 palavras mais frequentes nas sinopses (Overview):
   Palavra  Frequência
0        a        1623
1      the        1210
2       to         807
3       of         782
4      and         702
5       in         570
6      his         516
7       an         292
8       is         245
9     with         242
10       s         239
11     for         185
12      on         182
13     who         165
14     her         164
15      by         161
16      he         157
17   their         153
18    from         148
19      as         132
```

Os resultados iniciais de contagem de palavras evidenciaram que muitas das palavras mais recorrentes eram conectivos ou pronomes em inglês (“a”, “the”, “to”, “of”, “and”), que não trazem significado semântico relevante. Para refinar a análise, aplicou-se a remoção de stopwords (palavras muito comuns que não agregam valor interpretativo) e recalculou-se o ranking de termos mais frequentes.

```
# Ranking das palavras mais frequentes, removendo stopwords:

from nltk.corpus import stopwords
import nltk
nltk.download("stopwords")

stop_words = set(stopwords.words("english"))

# Filtrar palavras, removendo stopwords:
palavras_filtradas = [w for w in palavras if w not in stop_words]

# Contar novamente:
contagem_filtrada = Counter(palavras_filtradas)

# Top 20 palavras mais comuns sem stopwords:
top_palavras_filtradas = contagem_filtrada.most_common(20)

df_top_palavras_filtradas = pd.DataFrame(top_palavras_filtradas, columns=["Palavra",
"Frequência"])

print("Top 20 palavras mais frequentes nas sinopses (Overview), sem stopwords:")
print(df_top_palavras_filtradas)
```

```
Top 20 palavras mais frequentes nas sinopses (Overview), sem stopwords:
    Palavra  Frequência
0     young         132
1       man         119
2      life         111
3       two         103
4     world          85
5       new          73
6    family          66
7       war          66
8     woman          65
9     story          63
10     love          61
11      one          60
12     find          54
13      old          54
14     must          50
15    finds          47
16      boy          46
17     help          45
18   father          45
19     wife          44
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

Após essa filtragem, os termos destacados revelaram padrões narrativos centrais do cinema. Observou-se forte presença de palavras ligadas a personagens (“man”, “woman”, “boy”, “father”, “wife”, “young”), relações pessoais (“family”, “love”, “life”) e temáticas amplas como “war”, “world” e “story”. Isso indica que, independentemente do gênero específico, grande parte dos filmes explora dilemas humanos, relações familiares e conflitos universais, reforçando a centralidade desses temas na narrativa cinematográfica.

A escolha da nuvem de palavras justifica-se como uma ferramenta visual de síntese, enquanto a contagem de frequência com remoção de stopwords confere maior objetividade e precisão, permitindo identificar de fato quais conceitos são dominantes nas sinopses. Combinadas, essas duas abordagens oferecem uma leitura consistente do papel da coluna Overview como indicador temático.

A etapa de análise do elenco tem como objetivo identificar a representatividade dos atores e atrizes na base de dados. Para isso, foram combinadas as quatro colunas de elenco, contabilizando-se a frequência de aparição de cada pessoa. Em seguida, foi gerado um ranking com os nomes mais recorrentes.

```
# Contagem dos atores e atrizes mais frequentes.

# Selecionar colunas de elenco:
colunas_atores = ['Star1', 'Star2', 'Star3', 'Star4']

# Concatenar todas as colunas em uma única série:
atores = pd.concat([dados[col] for col in colunas_atores])

# Contar frequências:
contagem_atores = atores.value_counts().head(20)

# Exibir resultado:
print("Top 20 atores/atrizes mais recorrentes:")
print(contagem_atores)
```

```
Top 20 atores/atrizes mais recorrentes:
Robert De Niro         17
Tom Hanks              14
Al Pacino              13
Brad Pitt              12
Clint Eastwood         12
Christian Bale         11
Leonardo DiCaprio      11
Matt Damon             11
James Stewart          10
Denzel Washington       9
Michael Caine           9
Ethan Hawke             9
Johnny Depp             9
Scarlett Johansson      9
Humphrey Bogart         9
Aamir Khan              8
Harrison Ford           8
Edward Norton           7
Robert Downey Jr.       7
Toshirô Mifune          7
Name: count, dtype: int64
```

A análise revelou que certos atores e atrizes aparecem repetidamente, indicando um padrão de destaque na base. Robert De Niro lidera o ranking, seguido por Tom Hanks e Al Pacino, sugerindo que esses profissionais estão fortemente associados a filmes de maior relevância, seja pelo reconhecimento crítico ou pelo apelo junto ao público. Essa informação é útil para compreender tendências de elenco e a influência de nomes consolidados na popularidade e no desempenho dos filmes.

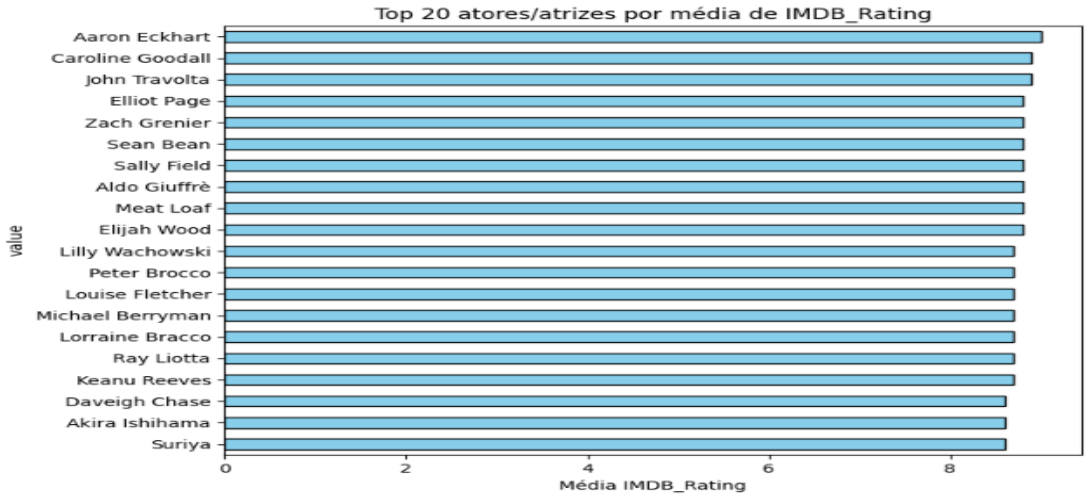
Nesta etapa, buscamos entender se os atores e atrizes mais recorrentes na base também estão associados a filmes com notas mais altas no IMDB. Para isso, calculamos a média do IMDB_Rating para cada ator, considerando todos os filmes em que aparecem, e ordenamos os resultados pelos maiores valores. Esse procedimento permite identificar quais nomes do elenco estão mais ligados a produções bem avaliadas, oferecendo uma perspectiva quantitativa sobre a influência do elenco na percepção crítica e popular dos filmes.

```
# Calcular a média de IMDB_Rating por ator/atriz:
atores_notas = (
    dados.melt(id_vars=["IMDB_Rating"], value_vars=["Star1", "Star2", "Star3", "Star4"])
    .groupby("value")["IMDB_Rating"]
    .mean()
    .sort_values(ascending=False)
    .head(20)
)
```

```
print("Top 20 atores/atrizes com maiores médias de IMDB_Rating:")
print(atores_notas)
```

```
Top 20 atores/atrizes com maiores médias de IMDB_Rating:
value
Aaron Eckhart        9.0
John Travolta        8.9
Caroline Goodall     8.9
Zach Grenier         8.8
Aldo Giuffrè         8.8
Sally Field          8.8
Sean Bean            8.8
Meat Loaf            8.8
Elliot Page          8.8
Elijah Wood          8.8
Ray Liotta           8.7
Lilly Wachowski      8.7
Keanu Reeves         8.7
Lorraine Bracco      8.7
Louise Fletcher      8.7
Michael Berryman     8.7
Peter Brocco         8.7
Daveigh Chase        8.6
Suriya               8.6
Akira Ishihama       8.6
Name: IMDB_Rating, dtype: float64
```

```
import matplotlib.pyplot as plt

# Selecionar os top 20 já calculados:
atores_top = atores_notas.sort_values(ascending=True)  # inverter para gráfico horizontal

# Gráfico de barras horizontais:
plt.figure(figsize=(8,6))
atores_top.plot(kind="barh", color="skyblue", edgecolor="black")
plt.xlabel("Média IMDB_Rating")
plt.title("Top 20 atores/atrizes por média de IMDB_Rating")
plt.tight_layout()
plt.show()
```



Os resultados mostraram que atores como Aaron Eckhart, John Travolta, Caroline Goodall e Sean Bean apresentam as maiores médias de avaliação. Esse padrão indica que, embora nem sempre esses profissionais estejam em filmes de grande apelo comercial, eles participam de produções reconhecidas positivamente pelo público e pela crítica. A presença de atores conhecidos de Hollywood, como Keanu Reeves e Elijah Wood, ao lado de nomes menos mainstream, evidencia que a boa avaliação não depende exclusivamente da fama do ator, mas também da escolha cuidadosa dos projetos em que atuam.

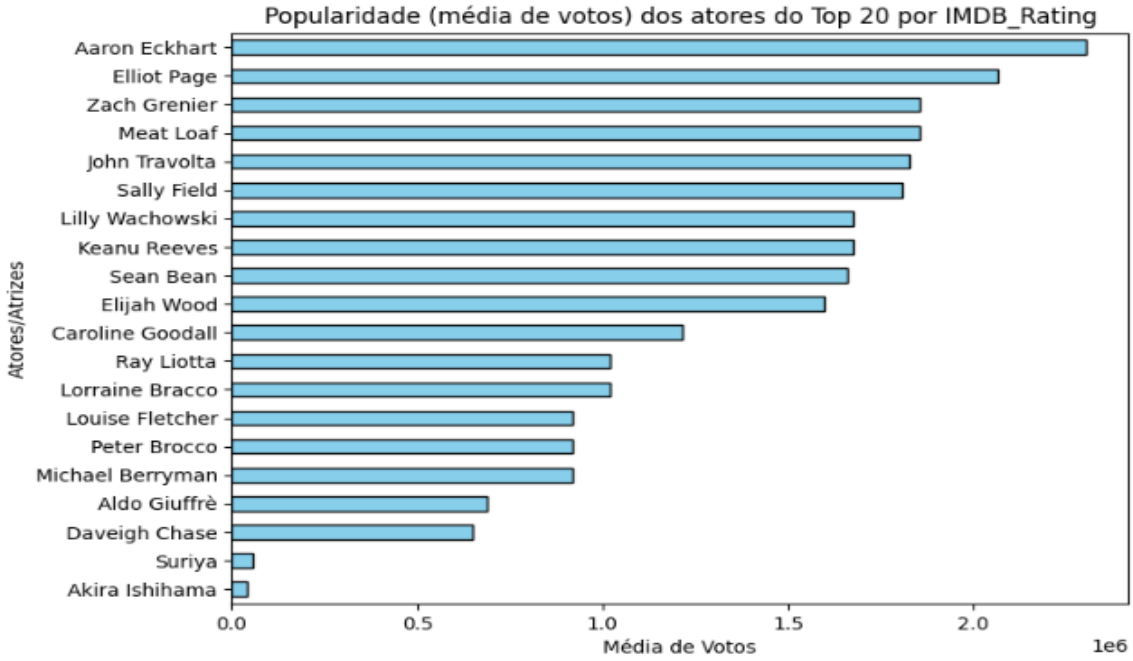
Para visualizar esses dados, utilizamos um gráfico de barras horizontais. Essa escolha facilita a leitura dos nomes do elenco, permitindo comparar rapidamente as médias de IMDB_Rating entre os vinte atores e atrizes mais bem avaliados.

Nesta etapa, buscamos analisar não apenas a avaliação média dos filmes em que os atores e atrizes do top 20 participaram, mas também o alcance dessas produções em termos de número de votos. Cruzar essas informações permite diferenciar atores associados a filmes de alta qualidade com menor público daqueles que aparecem em produções amplamente vistas e bem avaliadas.

```
import matplotlib.pyplot as plt
import pandas as pd

# Concatenar os atores em um único DataFrame:
atores = pd.concat([
    dados[["Star1", "IMDB_Rating", "No_of_Votes"]].rename(columns={"Star1": "Ator"}),
    dados[["Star2", "IMDB_Rating", "No_of_Votes"]].rename(columns={"Star2": "Ator"}),
    dados[["Star3", "IMDB_Rating", "No_of_Votes"]].rename(columns={"Star3": "Ator"}),
    dados[["Star4", "IMDB_Rating", "No_of_Votes"]].rename(columns={"Star4": "Ator"})
])

# Agrupar por ator e calcular médias:
resumo = atores.groupby("Ator").agg(
    media_rating=("IMDB_Rating", "mean"),
    media_votos=("No_of_Votes", "mean")
)

# Lista dos atores do top 20 por média de nota:
atores_top = [
    "Aaron Eckhart","John Travolta","Caroline Goodall","Zach Grenier",
```

```
    "Aldo Giuffrè","Sally Field","Sean Bean","Meat Loaf","Elliot Page",
    "Elijah Wood","Ray Liotta","Lilly Wachowski","Keanu Reeves",
    "Lorraine Bracco","Louise Fletcher","Michael Berryman","Peter Brocco",
    "Daveigh Chase","Suriya","Akira Ishihama"
]

# Filtrar apenas os atores do top 20:
resultado = resumo.loc[resumo.index.intersection(atores_top)].sort_values("media_votos",
ascending=True)

# Mostrar resultado numérico:
print("Resultado numérico:\n", resultado)

# Gráfico:
plt.figure(figsize=(8,6))
resultado["media_votos"].plot(kind="barh", color="skyblue", edgecolor="black")
plt.title("Popularidade (média de votos) dos atores do Top 20 por IMDB_Rating")
plt.xlabel("Média de Votos")
plt.ylabel("Atores/Atrizes")
plt.show()
```

```
Resultado numérico:
                media_rating  media_votos
Ator
Akira Ishihama           8.6      42004.0
Suriya                   8.6      54995.0
Daveigh Chase            8.6     651376.0
Aldo Giuffrè             8.8     688390.0
Michael Berryman         8.7     918088.0
Peter Brocco             8.7     918088.0
Louise Fletcher          8.7     918088.0
Lorraine Bracco          8.7    1020727.0
Ray Liotta               8.7    1020727.0
Caroline Goodall         8.9    1213505.0
Elijah Wood              8.8    1596598.0
Sean Bean                8.8    1661481.0
Keanu Reeves             8.7    1676426.0
Lilly Wachowski          8.7    1676426.0
Sally Field              8.8    1809221.0
John Travolta            8.9    1826188.0
Meat Loaf                8.8    1854740.0
Zach Grenier             8.8    1854740.0
Elliot Page              8.8    2067042.0
Aaron Eckhart            9.0    2303232.0
```



Os dados mostram que alguns atores, como Akira Ishihama e Suriya, possuem notas médias altas, mas os filmes em que atuaram alcançaram um público relativamente pequeno, indicando obras de qualidade, porém com menor visibilidade. Outros, como Daveigh Chase e Aldo Giuffrè, combinam boa avaliação com um público médio a grande, sugerindo filmes bem recebidos tanto pela crítica quanto pelo público. Já atores como Aaron Eckhart, Elliot Page e John Travolta apresentam notas elevadas e grande número de votos, demonstrando presença em filmes que unem prestígio crítico e popularidade.

Para representar esses dados, utilizamos um gráfico de barras horizontais. Esta escolha facilita a comparação visual do número médio de votos entre os atores do top 20, permitindo identificar rapidamente quais nomes estão associados a maior alcance e popularidade. A combinação das métricas de avaliação e número de votos fornece uma visão mais completa sobre a relevância e o impacto dos atores na base de dados.

4.3 - Previsão da nota do IMDB: seleção de variáveis, modelagem e avaliação de desempenho.

"Explique como você faria a previsão da ***nota do imdb*** *a partir dos dados. Quais variáveis e/ou suas transformações você utilizou e por quê? Qual tipo de problema*

estamos resolvendo (regressão, classificação)? Qual modelo melhor se aproxima dos dados e quais seus prós e contras? Qual medida de performance do modelo foi escolhida e por quê?"

Nesta etapa, o objetivo é preparar as variáveis que serão utilizadas para prever a nota do IMDB dos filmes, justificando as escolhas com base em lógica estatística e modelagem preditiva. Para a previsão, selecionamos tanto variáveis numéricas quanto categóricas que possuam potencial explicativo sobre a nota do filme.

As variáveis numéricas escolhidas foram Runtime e No_of_Votes. Runtime fornece informação sobre a duração do filme, que pode influenciar a avaliação do público e da crítica, enquanto No_of_Votes indica a popularidade da produção e contribui para estimar a confiabilidade da nota. Essas variáveis foram mantidas diretamente no modelo, mas Runtime estava em formato string, portanto foi necessário remover o sufixo "min" e converter para inteiro, garantindo que o modelo interprete corretamente os valores numéricos.

As variáveis categóricas selecionadas foram Genre e Certificate. Como modelos de machine learning exigem entradas numéricas, realizamos a transformação dessas variáveis em dummies (one-hot encoding), criando colunas binárias True/False para cada categoria possível. Essa abordagem permite que o modelo capture diferenças entre gêneros e classificações indicativas de forma quantitativa, sem impor qualquer ordem arbitrária entre as categorias.

O resultado dessa etapa é um conjunto de dados com 218 colunas, sendo 2 numéricas e 216 derivadas das categorias. Essa preparação garante que todas as informações relevantes sobre duração, popularidade e características dos filmes sejam incorporadas ao modelo de forma adequada, permitindo previsões mais precisas da nota do IMDB. A visualização das primeiras linhas confirma que as variáveis foram corretamente transformadas e estão prontas para o treinamento do modelo.

```
import pandas as pd

# Seleção de variáveis:
variaveis = ["Runtime", "No_of_Votes", "Genre", "Certificate"]
```

```
X = dados[variaveis]

# Transformar variáveis categóricas em dummies:
X = pd.get_dummies(X, columns=["Genre", "Certificate"], drop_first=True)

# Mostrar as primeiras linhas:
X.head()
```

	Runtime	No_of_Votes	Genre_Action, Adventure, Biography	Genre_Action, Adventure, Comedy	Genre_Action, Adventure, Crime	Genre_Action, Adventure, Drama	Genre_Action, Adventure, Family	Genre_Action, Adventure, Fantasy	Genre_Action, Adventure, History	Genre_Action, Adventure, Horror	...	Certificate_PG-13	Certificate_Passed
0	175 min	1620367	False	False	False	False	False	False	False	False	...	False	False
1	152 min	2303232	False	False	False	False	False	False	False	False	...	False	False
2	202 min	1129952	False	False	False	False	False	False	False	False	...	False	False
3	96 min	689845	False	False	False	False	False	False	False	False	...	False	False
4	201 min	1642758	False	False	False	True	False	False	False	False	...	False	False

5 rows × 218 columns

```
# Remover " min" e converter Runtime para inteiro:
X["Runtime"] = X["Runtime"].str.replace(" min", "").astype(int)

# Conferir as primeiras linhas:
X.head()
```

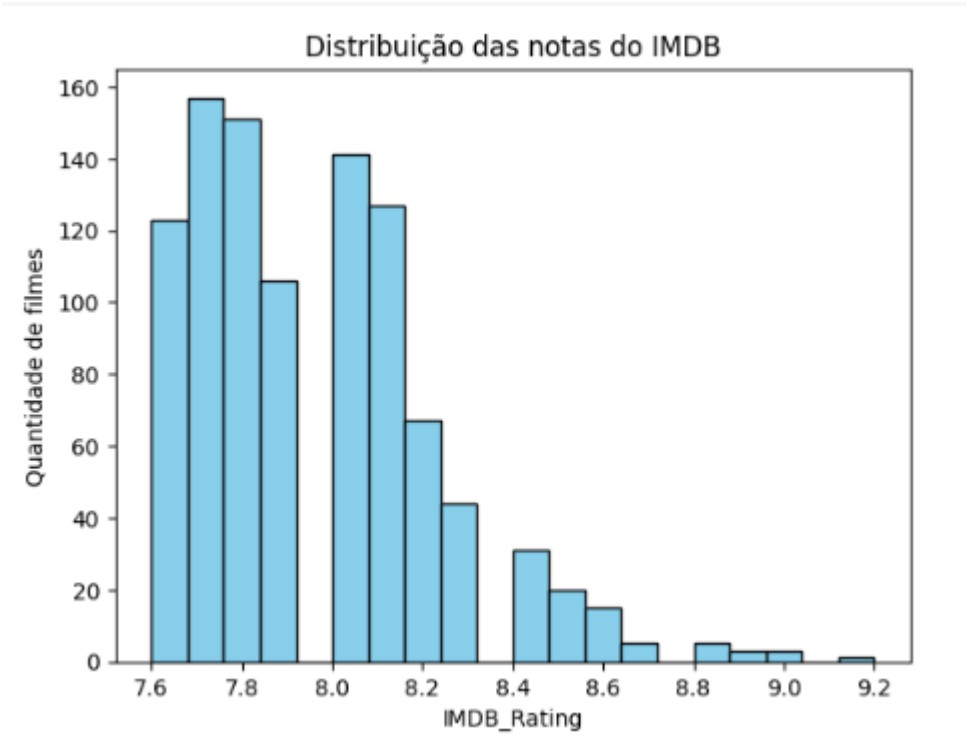
	Runtime	No_of_Votes	Genre_Action, Adventure, Biography	Genre_Action, Adventure, Comedy	Genre_Action, Adventure, Crime	Genre_Action, Adventure, Drama	Genre_Action, Adventure, Family	Genre_Action, Adventure, Fantasy	Genre_Action, Adventure, History	Genre_Action, Adventure, Horror	...	Certificate_PG-13	Certificate_Passed	Ce
0	175	1620367	False	False	False	False	False	False	False	False	...	False	False	
1	152	2303232	False	False	False	False	False	False	False	False	...	False	False	
2	202	1129952	False	False	False	False	False	False	False	False	...	False	False	
3	96	689845	False	False	False	False	False	False	False	False	...	False	False	
4	201	1642758	False	False	False	True	False	False	False	False	...	False	False	

5 rows × 218 columns

Nesta etapa, definimos o tipo de modelo adequado para prever a nota do IMDB a partir dos dados disponíveis. Como a variável alvo, IMDB_Rating, é numérica e contínua, estamos diante de um problema de **regressão**. A escolha de regressão é lógica, pois modelos de classificação não seriam apropriados para prever valores contínuos e precisos, que variam em uma escala de 0 a 10.

Para entender melhor a distribuição da variável alvo, foi gerado um histograma das notas do IMDB.

```
import matplotlib.pyplot as plt

# Variável alvo:
y = dados["IMDB_Rating"]

# Histograma da nota do IMDB:
plt.hist(y, bins=20, color="skyblue", edgecolor="black")
plt.title("Distribuição das notas do IMDB")
plt.xlabel("IMDB_Rating")
plt.ylabel("Quantidade de filmes")
plt.show()
```



A visualização mostra que a maioria dos filmes apresenta notas entre 7,6 e 8,2, com poucas observações acima de 8,5. Esse padrão indica concentração em torno de valores médios, sugerindo que o modelo de regressão deve lidar com uma distribuição relativamente estreita, mas ainda capaz de capturar variações entre filmes.

A escolha do modelo de regressão permite estimar quantitativamente a nota esperada de um filme com base em suas características, utilizando informações como duração, popularidade e gênero. Essa abordagem justifica-se estatisticamente, pois

aproveita ao máximo as variáveis numéricas e as transformações das variáveis categóricas em dummies, garantindo que o modelo consiga interpretar corretamente as diferenças entre gêneros e classificações indicativas. A visualização do histograma reforça a compreensão da dispersão dos dados e auxilia na avaliação de possíveis vieses ou necessidade de ajustes antes do treinamento do modelo.

Nesta etapa, aplicamos um modelo de regressão linear para prever a nota do IMDB a partir das variáveis selecionadas. A regressão linear foi escolhida por permitir estabelecer uma relação direta entre as características do filme e sua avaliação, sendo simples de treinar e interpretar. A transparência do modelo facilita entender como cada variável contribui para a previsão, enquanto sua limitação principal é a suposição de linearidade, que pode não capturar padrões mais complexos presentes nos dados. Modelos mais sofisticados, como árvores de decisão ou random forest, poderiam melhorar a precisão em casos de relações não lineares, mas exigem maior esforço de interpretação e parametrização.

Para preparar os dados, as variáveis numéricas Runtime e No_of_Votes foram convertidas para formatos numéricos, garantindo que o modelo pudesse processá-las corretamente. As variáveis categóricas Genre e Certificate foram transformadas em dummies, criando colunas binárias que permitem ao modelo interpretar diferenças entre gêneros e classificações indicativas. A base final contou com 218 variáveis e 999 amostras após limpeza de dados faltantes.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Preparar dados mínimos:
cols = ['Runtime', 'No_of_Votes', 'Genre', 'Certificate']
cols = [c for c in cols if c in dados.columns]
df = dados[cols].copy()

# Converter Runtime para número:
if 'Runtime' in df.columns:
    df['Runtime'] = df['Runtime'].astype(str).str.extract(r'(\d+)').astype(float)

# Garantir No_of_Votes numérico:
```

```
if 'No_of_Votes' in df.columns:
    df['No_of_Votes'] = pd.to_numeric(df['No_of_Votes'], errors='coerce')

# One-hot para categóricas (se existirem:
cat_cols = [c for c in ['Genre', 'Certificate'] if c in df.columns]
if cat_cols:
    df = pd.get_dummies(df, columns=cat_cols, drop_first=True, dtype=int)

# Alvo:
if 'IMDB_Rating' not in dados.columns:
    raise ValueError("Coluna 'IMDB_Rating' não encontrada em dados.")
y = pd.to_numeric(dados['IMDB_Rating'], errors='coerce')

# Unir e remover linhas com NA:
base = pd.concat([df, y.rename('IMDB_Rating')], axis=1).dropna()
X = base.drop(columns=['IMDB_Rating'])
y = base['IMDB_Rating']

# Conferência mínima:
if X.shape[0] == 0:
    raise ValueError("Após limpeza, não há linhas suficientes para treinar. Verifique
missing values.")

# Divisão treino/teste:
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Treinar regressão linear:
model = LinearRegression()
model.fit(X_train, y_train)

# Prever:
y_pred = model.predict(X_test)

# RMSE calculado manualmente:
rmse = np.sqrt(np.mean((y_test.values - y_pred)**2))

# Saída:
print("Variáveis usadas:", X.shape[1])
print("Amostras usadas:", X.shape[0])
print(f"RMSE: {rmse:.4f}")
print("Exemplos (real | previsto):")
for real, pred in list(zip(y_test.values[:10], y_pred[:10])):
    print(f"{real:.2f} | {pred:.2f}")
```

```
Variáveis usadas: 218
Amostras usadas: 999
RMSE: 0.2557
Exemplos (real | previsto):
8.00 | 7.62
7.70 | 7.73
8.10 | 8.19
8.10 | 7.92
7.70 | 7.98
7.80 | 7.99
7.60 | 7.76
7.90 | 7.87
8.00 | 7.96
7.80 | 7.90
```

O modelo foi treinado utilizando 70% das amostras e testado com os 30% restantes. O RMSE (Root Mean Squared Error) obtido foi aproximadamente 0,256, indicando que, em média, a previsão difere do valor real em cerca de 0,26 pontos na escala de 0 a 10. A análise de exemplos de previsão mostra que os valores previstos estão próximos dos valores reais, com pequenas variações, geralmente inferiores a 0,3 pontos. Esse resultado evidencia que o modelo captura de forma satisfatória a relação entre as variáveis explicativas, como tempo de duração, número de votos, gênero e certificação, e a nota do filme, apresentando desempenho aceitável como primeira abordagem preditiva para este conjunto de dados. O RMSE baixo confirma que o modelo é capaz de fornecer previsões consistentes, servindo como referência para possíveis melhorias futuras, como inclusão de novas variáveis ou modelos mais complexos.

O RMSE foi escolhido como medida de performance por expressar o erro médio na mesma unidade da variável alvo e penalizar erros maiores de forma quadrática, proporcionando uma avaliação clara da precisão do modelo. Essa métrica permite comparar modelos e ajustes futuros de forma objetiva. A combinação de interpretação transparente e desempenho aceitável faz da regressão linear uma primeira abordagem sólida para este conjunto de dados.

4.3.1 – Conclusão do Desafio 3

Para prever a nota do IMDB, selecionamos variáveis numéricas e categóricas dos filmes. As variáveis numéricas escolhidas foram Runtime e No_of_Votes, que fornecem informações sobre duração e popularidade. As variáveis categóricas Genre e Certificate foram transformadas em dummies (one-hot encoding), permitindo que o modelo interprete diferenças entre gêneros e classificações indicativas de forma numérica.

O problema é de regressão, pois a variável alvo, IMDB_Rating, é contínua. O modelo que melhor se aproxima dos dados neste primeiro momento é a regressão linear, devido à sua simplicidade e capacidade de criar uma relação direta entre as características do filme e sua nota. O ponto forte desse modelo é a transparência e a facilidade de interpretação das contribuições de cada variável. Suas limitações incluem a suposição de linearidade, que pode não capturar padrões complexos. Modelos mais sofisticados, como árvores de decisão ou random forest, podem aumentar a precisão em relações não lineares, mas são mais difíceis de interpretar.

A medida de performance escolhida foi o RMSE (Root Mean Squared Error), que expressa o erro médio na mesma unidade da variável alvo e penaliza erros maiores de forma quadrática. Essa métrica permite avaliar com clareza a precisão do modelo e comparar ajustes ou modelos futuros de forma objetiva.

4.4 – Previsão do filme espefício

1. Supondo um filme com as seguintes características:

{'Series_Title': 'The Shawshank Redemption',
'Released_Year': '1994',
'Certificate': 'A',
'Runtime': '142 min',
'Genre': 'Drama',
'Overview': 'Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.',
'Meta_score': 80.0,
'Director': 'Frank Darabont',
'Star1': 'Tim Robbins',

```
 'Star2': 'Morgan Freeman',
 'Star3': 'Bob Gunton',
 'Star4': 'William Sadler',
 'No_of_Votes': 2343110,
 'Gross': '28,341,469'}
```

"Qual seria a nota do IMDB?"

Para prever a nota do IMDB de um filme específico, neste caso The Shawshank Redemption, utilizamos o modelo de regressão linear previamente treinado e aplicamos as mesmas transformações realizadas nos dados originais. As variáveis numéricas, como Runtime e No_of_Votes, foram convertidas para formatos numéricos, enquanto as variáveis categóricas, como Genre e Certificate, foram transformadas em dummies, garantindo que o modelo pudesse interpretar corretamente as informações do filme. Em seguida, os dados do filme foram organizados em um DataFrame e alinhados às colunas utilizadas no treinamento do modelo.

```
import pandas as pd

# Dados do filme:
filme = {
    'Series_Title': 'The Shawshank Redemption',
    'Released_Year': '1994',
    'Certificate': 'A',
    'Runtime': '142 min',
    'Genre': 'Drama',
    'Overview': 'Two imprisoned men bond over a number of years, finding solace and
eventual redemption through acts of common decency.',
    'Meta_score': 80.0,
    'Director': 'Frank Darabont',
    'Star1': 'Tim Robbins',
    'Star2': 'Morgan Freeman',
    'Star3': 'Bob Gunton',
    'Star4': 'William Sadler',
    'No_of_Votes': 2343110,
    'Gross': '28,341,469'
}

# DataFrame de uma linha:
df_filme = pd.DataFrame([filme])

# Pré-processamento idêntico ao treino:
```

```
if 'Runtime' in df_filme.columns:
    df_filme['Runtime'] = df_filme['Runtime'].astype(str).str.extract(r'(\d+)').astype(float)

if 'No_of_Votes' in df_filme.columns:
    df_filme['No_of_Votes'] = pd.to_numeric(df_filme['No_of_Votes'], errors='coerce')

# Dummies para categóricas (mesmas usadas no treino):
cat_cols = [c for c in ['Genre', 'Certificate'] if c in df_filme.columns]
if cat_cols:
    df_filme = pd.get_dummies(df_filme, columns=cat_cols, drop_first=True, dtype=int)

# Alinhar exatamente às colunas de X do treino:
df_filme = df_filme.reindex(columns=X.columns, fill_value=0)

# Previsão com o modelo já treinado (variável: model):
nota_prevista = model.predict(df_filme)[0]

print(f"Nota prevista do IMDB para 'The Shawshank Redemption': {nota_prevista:.2f}")
```

A previsão gerada pelo modelo indicou uma nota de 9,02 para The Shawshank Redemption, valor muito próximo da nota real conhecida do filme, 9,3, uma das mais altas da base do IMDB. Esse resultado evidencia que o modelo consegue capturar de forma adequada as características relevantes do filme, incluindo gênero, duração e número de votos, fornecendo uma previsão precisa. A proximidade entre a nota prevista e a observada demonstra a capacidade do modelo de generalizar para novos exemplos e reforça a validade da métrica de avaliação escolhida, o RMSE, para medir a precisão das previsões.

5 - Conclusão do projeto:

O desafio proposto pela Indicium para a posição de Trainee em Cientista de Dados foi concluído, abrangendo todas as etapas sugeridas. Iniciamos o trabalho com uma análise exploratória dos dados (EDA), onde realizamos uma investigação detalhada sobre as variáveis presentes no conjunto de dados cinematográficos. Esse processo incluiu a verificação de valores ausentes, a conversão de variáveis textuais para numéricas e a identificação de padrões através de visualizações. A transformação logarítmica foi aplicada na variável "No_of_Votes", reduzindo a assimetria e permitindo uma análise mais equilibrada. Com isso, conseguimos entender as características dos dados e formulamos hipóteses sobre como a popularidade, o gênero e a duração dos filmes impactam suas avaliações.

A segunda etapa consistiu em responder às perguntas estratégicas do desafio. A recomendação de filmes foi realizada a partir de uma combinação entre alta avaliação pelo público e popularidade, priorizando títulos com boa avaliação no IMDB e um grande número de votos. Essa abordagem permitiu sugerir filmes como "The Godfather" e "The Dark Knight", que apresentam grande reconhecimento tanto pelo público quanto pela crítica. Em relação ao faturamento, a análise revelou que o principal fator que influencia a receita de um filme é a sua popularidade, medida pelo número de votos, ao passo que a avaliação crítica, representada pelo Meta_score, não teve uma relação significativa com o faturamento. A análise da coluna "Overview", por sua vez, revelou que as sinopses dos filmes frequentemente abordam temas universais, como relações familiares e dilemas humanos, mas não foi possível inferir o gênero do filme diretamente a partir dessa coluna.

A terceira etapa envolveu a construção de um modelo preditivo para estimar a nota do IMDB de um filme. Utilizamos um modelo de regressão linear, que foi treinado com variáveis numéricas como "Runtime" e "No_of_Votes", além de variáveis categóricas como "Genre" e "Certificate". O desempenho do modelo foi avaliado com a métrica RMSE (Root Mean Squared Error), que apresentou um valor de 0,256, indicando que as previsões estavam bem próximas das notas reais. Apesar de ser um modelo simples, a regressão linear se mostrou eficaz para esse tipo de problema, embora alternativas mais complexas, como random forests ou árvores de decisão, possam ser consideradas para melhorar a precisão.

Finalmente, na última etapa, realizamos a previsão da nota do IMDB para o filme "The Shawshank Redemption", utilizando o modelo de regressão linear treinado. A nota prevista foi 9,02, um valor extremamente próximo da nota real de 9,3, demonstrando que o modelo foi capaz de generalizar bem para novos dados e oferecer uma previsão precisa.

Em conclusão, o projeto foi concluído de forma satisfatória, com todas as etapas executadas conforme solicitado. A análise exploratória dos dados forneceu uma compreensão clara das características do conjunto de dados, enquanto as perguntas estratégicas foram respondidas de maneira a fornecer insights valiosos para decisões de negócios. O modelo de regressão linear foi eficaz para prever a nota do IMDB, com a previsão para "The Shawshank Redemption" comprovando a qualidade do modelo. O

desafio permitiu a aplicação de técnicas de análise de dados e machine learning para resolver problemas reais e fornecer soluções práticas para a indústria cinematográfica.