Desafio Cientista de dados

Lucas Nunes da Silveira

ANÁLISE EXPLORATÓRIA DOS DADOS (EDA)

A análise exploratória de dados (AED) desempenha um papel fundamental em qualquer projeto de análise ou modelagem de dados. Ela não se limita apenas a uma simples inspeção superficial dos dados, mas sim a uma exploração profunda e sistemática que revela insights cruciais para entender a natureza dos dados em questão.

Primeiramente, a AED nos ajuda a compreender a estrutura dos dados. Isso inclui a identificação dos tipos de variáveis presentes, como numéricas ou categóricas, e a distribuição desses dados ao longo das variáveis. Além disso, nos permite detectar a presença de valores ausentes ou outliers que podem distorcer análises estatísticas posteriores, garantindo assim uma limpeza e preparação adequada dos dados.

A validação de suposições também é uma parte crucial da análise exploratória. Antes de aplicar métodos estatísticos mais complexos, é essencial verificar se as condições necessárias, como a normalidade dos dados ou a independência entre variáveis, são atendidas. Isso assegura a robustez e a confiabilidade dos resultados obtidos.

Outro benefício significativo da AED é a seleção de variáveis. Ao explorar visualmente as relações entre diferentes variáveis, podemos identificar quais são mais relevantes para o problema em questão. Isso não apenas melhora a precisão dos modelos subsequentes, mas também evita o overfitting, onde um modelo se ajusta excessivamente aos dados de treinamento e não generaliza bem para novos dados.

Além disso, a AED proporciona uma base sólida para a preparação de dados. Ela revela a necessidade de transformações, como normalização de distribuições ou tratamento de dados ausentes, preparando assim os dados de forma adequada para análises mais avançadas.

Visualizações e resumos obtidos durante a análise exploratória não são apenas ferramentas para especialistas em dados. Eles também facilitam a comunicação de insights e descobertas para partes interessadas não técnicas, como tomadores de decisão e colegas de outras áreas.

Por fim, a AED não apenas valida hipóteses existentes, mas muitas vezes inspira novas perguntas de pesquisa e insights que podem guiar investigações mais profundas e estratégias analíticas mais refinadas.

Em resumo, a análise exploratória de dados é um passo essencial e poderoso que não só prepara o terreno para análises estatísticas mais avançadas e modelagem preditiva, mas também revela insights profundos que impulsionam a compreensão e a tomada de decisão baseada em dados.

1. Conhecendo os dados

Começamos a importar bibliotecas usadas no código, em python:

```
#aqui é as importação das bibliotecas que irá ser usual para execução do código
import pandas as pd
import numpy as np
import seaborn as sns
import missingno as msno
import matplotlib.pyplot as plt
import plotly.express as px
import nbformat
from collections import  Counter
from sklearn.preprocessing import MinMaxScaler
from nltk.corpus import stopwords
import warnings
warnings.filterwarnings('ignore')
import nltk
nltk.download('punkt')
nltk.download('stopwords')
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas
```

Após feita a importação iremos começa a ver com os dados se comporta:

```
#Criei o dataset com  os dados que estar em formato .csv
df = pd.read_csv("desafio_indicium_imdb.csv")
df.head()
```

Criação do Dataset, do aquivo do que estar e formato .csv , usando a função head() para mostrar as primeiras linhas gerando o output

	Unnamed: 0	Series_Title	Released_Year	Certificate	Runtime	Genre	IMDB_Rating	Overview	Meta_score	Director	Star1	Star2	Star3	Star4	No_of_Votes	Gross
0	1	The Godfather	1972	A	175 min	Crime, Drama	9.2	An organized crime dynasty's aging patriarch t...	100.0	Francis Ford Coppola	Marlon Brando	Al Pacino	James Caan	Diane Keaton	1620367	134,966,411
1	2	The Dark Knight	2008	UA	152 min	Action, Crime, Drama	9.0	When the menace known as the Joker wreaks havo...	84.0	Christopher Nolan	Christian Bale	Heath Ledger	Aaron Eckhart	Michael Caine	2303232	534,858,444
2	3	The Godfather: Part II	1974	A	202 min	Crime, Drama	9.0	The early life and career of Vito Corleone in ...	90.0	Francis Ford Coppola	Al Pacino	Robert De Niro	Robert Duvall	Diane Keaton	1129952	57,300,000
3	4	12 Angry Men	1957	U	96 min	Crime, Drama	9.0	A jury holdout attempts to prevent a miscarria...	96.0	Sidney Lumet	Henry Fonda	Lee J. Cobb	Martin Balsam	John Fiedler	689845	4,360,000
4	5	The Lord of the Rings: The Return of the King	2003	U	201 min	Action, Adventure, Drama	8.9	Gandalf and Aragorn lead the World of Men agai...	94.0	Peter Jackson	Elijah Wood	Viggo Mortensen	Ian McKellen	Orlando Bloom	1642758	377,845,905

```
#aqui eu apliquei o Drop para dropar, a coluna colocando o nome da coluna(Unnamed:0) e o Eixo(axis=1)
df = df.drop('Unnamed: 0', axis=1)
```

```
#aqui serve para mostrar o tipo das variáveis e a quantidade de registros não nulos.
df.info()
```

Output: gera as informações da base de dados, tipo dos dados , valores não nulos a quantidade de registros na base de dados entre número de colunas que são 15 de (0 a 14), os nomes de colunas.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999 entries, 0 to 998
Data columns (total 15 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Series_Title   999 non-null    object 
 1   Released_Year  999 non-null    object 
 2   Certificate    898 non-null    object 
 3   Runtime        999 non-null    object 
 4   Genre          999 non-null    object 
 5   IMDB_Rating    999 non-null    float64
 6   Overview       999 non-null    object 
 7   Meta_score     842 non-null    float64
 8   Director       999 non-null    object 
 9   Star1          999 non-null    object 
 10  Star2          999 non-null    object 
 11  Star3          999 non-null    object 
 12  Star4          999 non-null    object 
 13  No_of_Votes    999 non-null    int64  
 14  Gross          830 non-null    object 
dtypes: float64(2), int64(1), object(12)
memory usage: 117.2+ KB
```

```
df.describe()
```

Output : gera valores em cont, média, desvio padrão, mín, primeiro quartil(25%), segundo quartil(50%), e terceiro quartil (75%), e o valores máximos.

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

```
df[df.duplicated()]
```

Output: Gera uma saída mostrando os dados duplicados, visto que não mostrou nenhum dados logo constata que não à existência de dados duplicados, dentro da base:

Series_Title	Released_Year	Certificate	Runtime	Genre	IMDB_Rating	Overview	Meta_score	Director	Star1	Star2	Star3	Star4	No_of_Votes	Gross

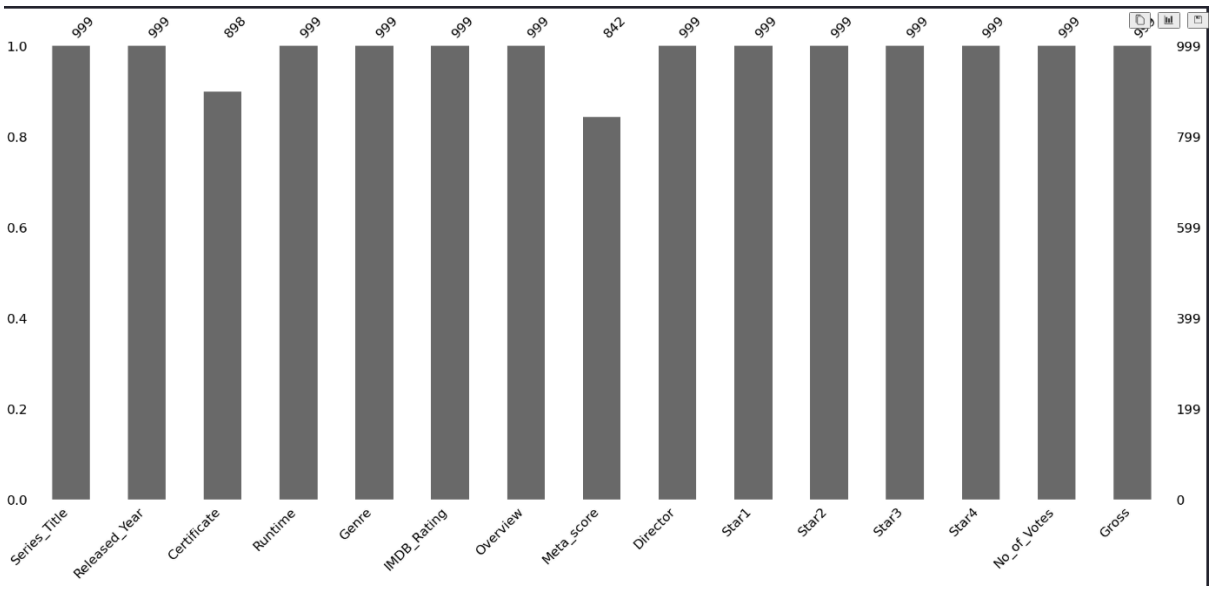
df.isnull().sum()

Output: Mostra o somatório de todos os dados que estão Nulos(Null), aí nesse dados dá para verificar a quantidade de valores nulos em cada coluna visto.Onde na parte ***Certificate*** consta que tem **101 registros nulos**, e no **Meta_score** tem **157 nulos**, e o **Gross** tem **169 nulos.**

```
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
```

A visualização da ausência dos dados, pelo gráfico, usando o código :

```
msno.bar(df)
```




```
df['Certificate'].unique()
```

Output: Ele faz uma análise para mostrar os valores únicos (mesmo que tenha mais de um registro igual) , dentro de um array, serve para verificar o comportamento do registro de uma coluna específica, nesse caso a coluna é Certificate.

```
array(['A', 'UA', 'U', 'PG-13', 'R', nan, 'PG', 'G', 'Passed', 'TV-14',
       '16', 'TV-MA', 'Unrated', 'GP', 'Approved', 'TV-PG', 'U/A'],
      dtype=object)
```

```
df['Released_Year'].value_counts()
```

Output: Mostra a quantidade de registro por cada ano, dentro da coluna “**Released_Year”**, onde em 2014 mostra 32 registros, 2004, 31 registros …

```
Released_Year
2014    32
2004    31
2009    29
2013    28
2016    28
        ..
1920     1
1930     1
1922     1
1943     1
PG       1
Name: count, Length: 100, dtype: int64
```

```
df.columns
```

Output: Mostra as Colunas da base de dados, onde podemos ver que tem várias colunas.

```
Index(['Series_Title', 'Released_Year', 'Certificate', 'Runtime', 'Genre',
       'IMDB_Rating', 'Overview', 'Meta_score', 'Director', 'Star1', 'Star2',
       'Star3', 'Star4', 'No_of_Votes', 'Gross'],
      dtype='object')
```

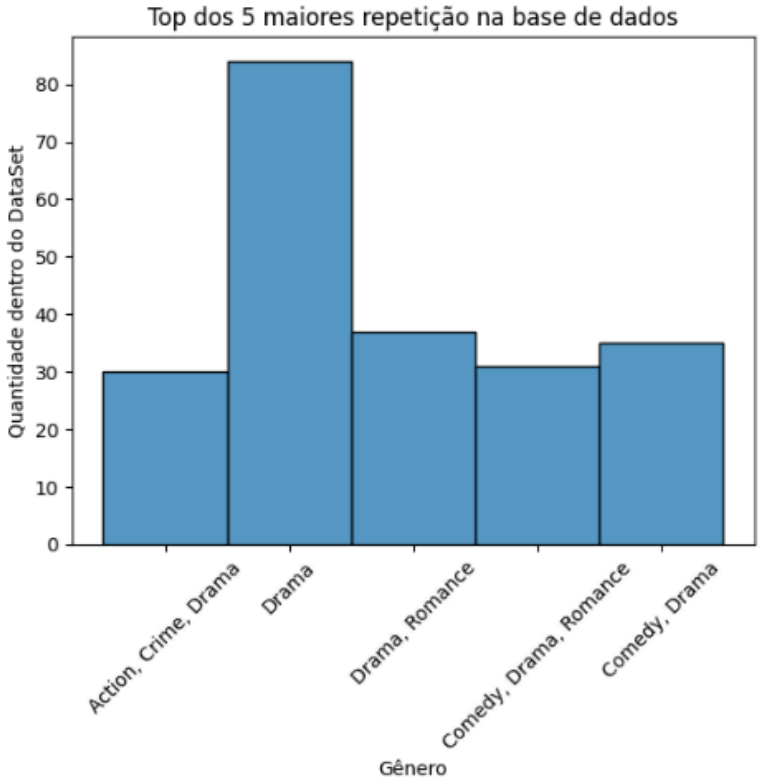
```
df.shape
```

Output: mostra uma tupla(linhas, colunas) do dataset

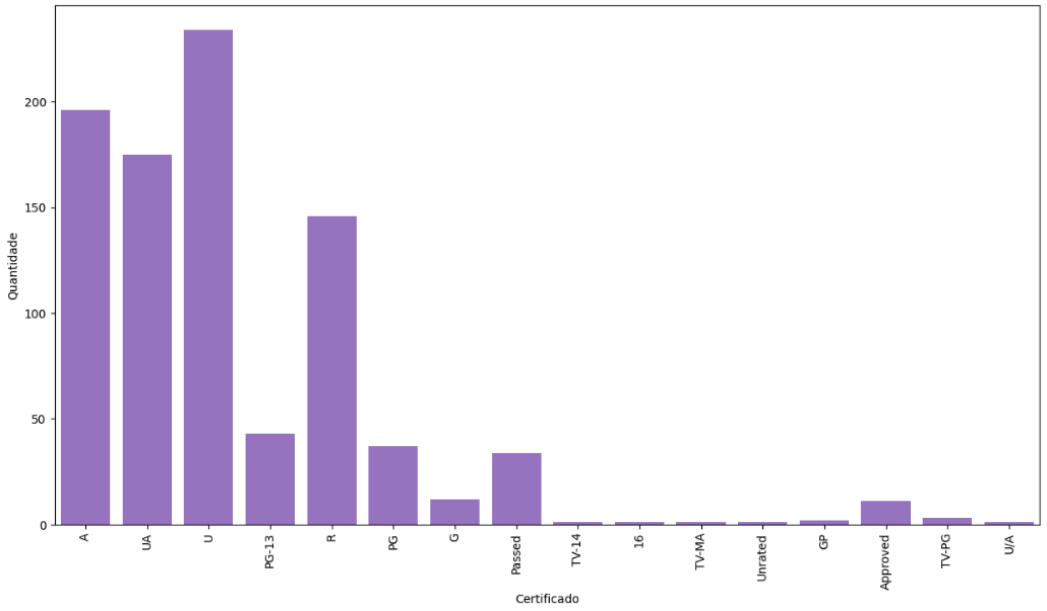
```
(998, 15)
```

Agora iremos sair do código em si para uma parte mais visual, onde podemos ver os dados:

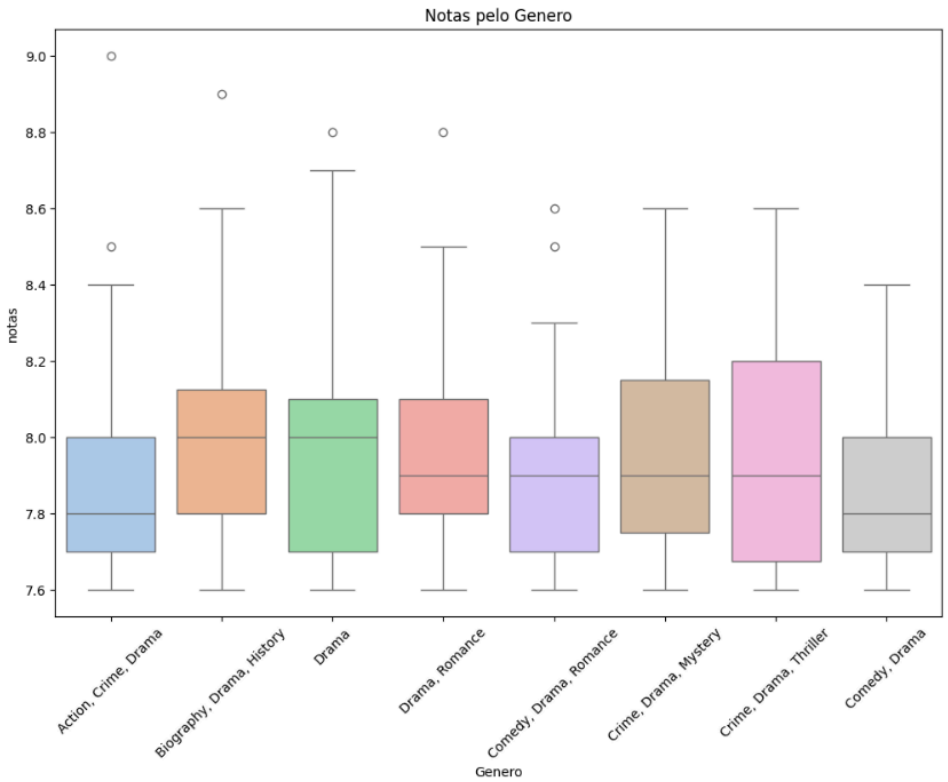
Output: O gráfico mostra o Top 5 dos gêneros, onde podemos ver que dentro da base de dados o valor da coluna Drama é que tem mais dados.



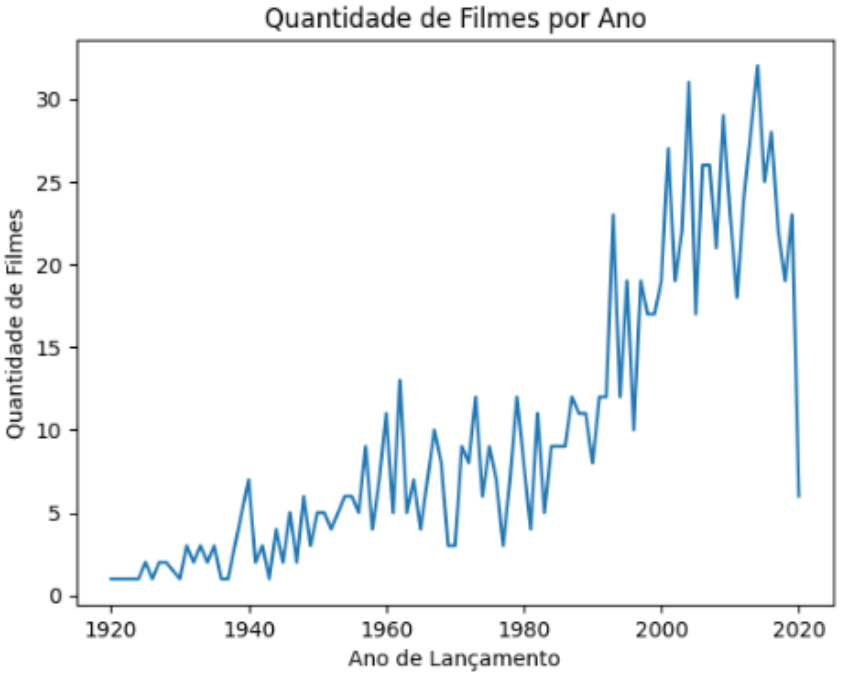
Output: O gráfico analisa as classificações dos filmes dentro da base de dados onde mostra que a classificação “U” referente a classificação de filme livre.

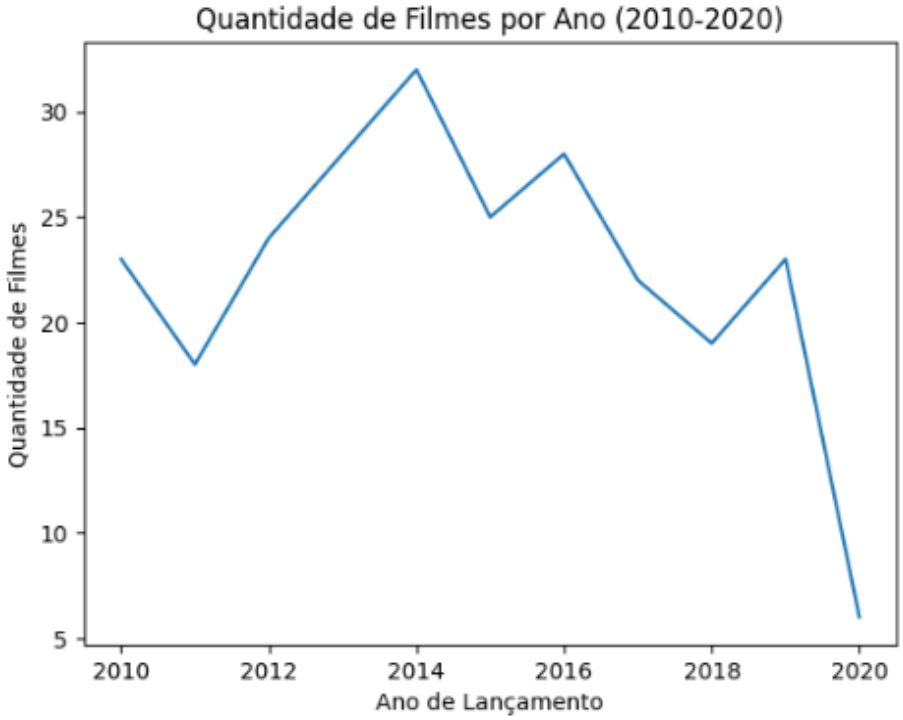


Output: O Gráfico boxplot, ele explica estatisticamente o comportamento de algumas variáveis, podemos fazer um corte no “Action, Crime, Drama”, onde podemos constatar que existe dois outlier (pontos que Discrepante em relação a base de dados), podemos ver que média ficou 7.8, A uma variação muito grande entre terceiro quartil (75%) ao ponto máximo (8.4) ponto mínimo (7.6)

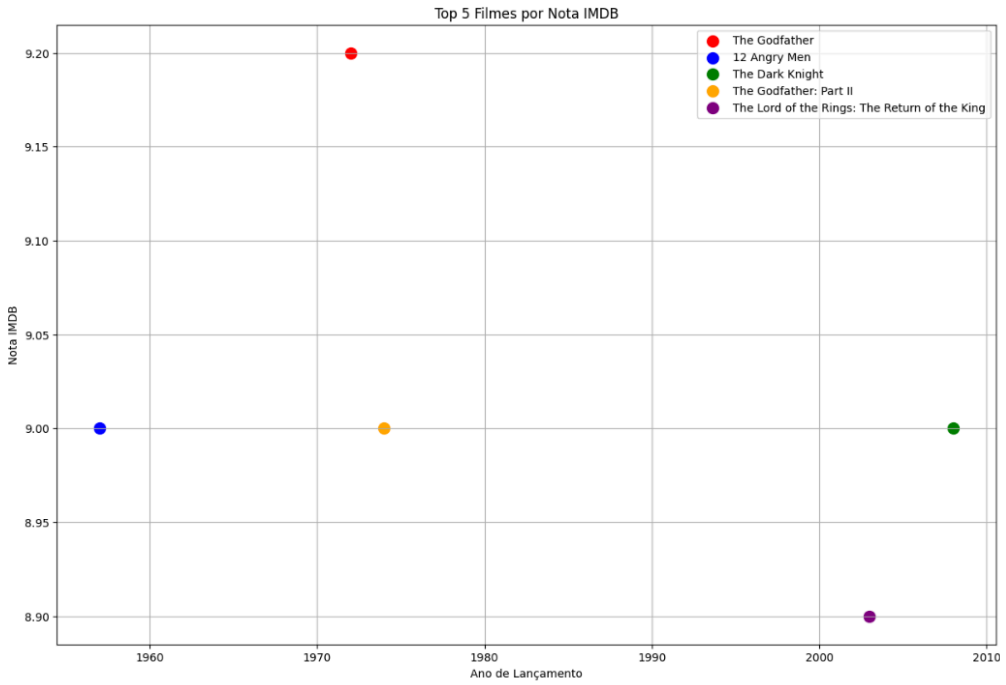


Output: O Gráfico de linhas mostra o comportamento da base de dados em relação à quantidade de filmes produzidos por ano e podemos analisar que em 2010 a 2020 temos uma queda considerável na produção dos filmes, onde irá ficar mais claro no segundo gráfico onde eu fiz um corte graficamente. Essa queda repentina foi por conta do coronavírus da COVID-19, que foi uma pandemia onde a produção de filmes e cinemas foram fechados na época.





Output: O gráfico em de dispersão foi feito pontos, para verificar os 5 maiores filmes com maiores nota e podemos perceber que o “The godfather” é o que teve maior nota 9.20.



2 Filme recomendado para um desconhecido

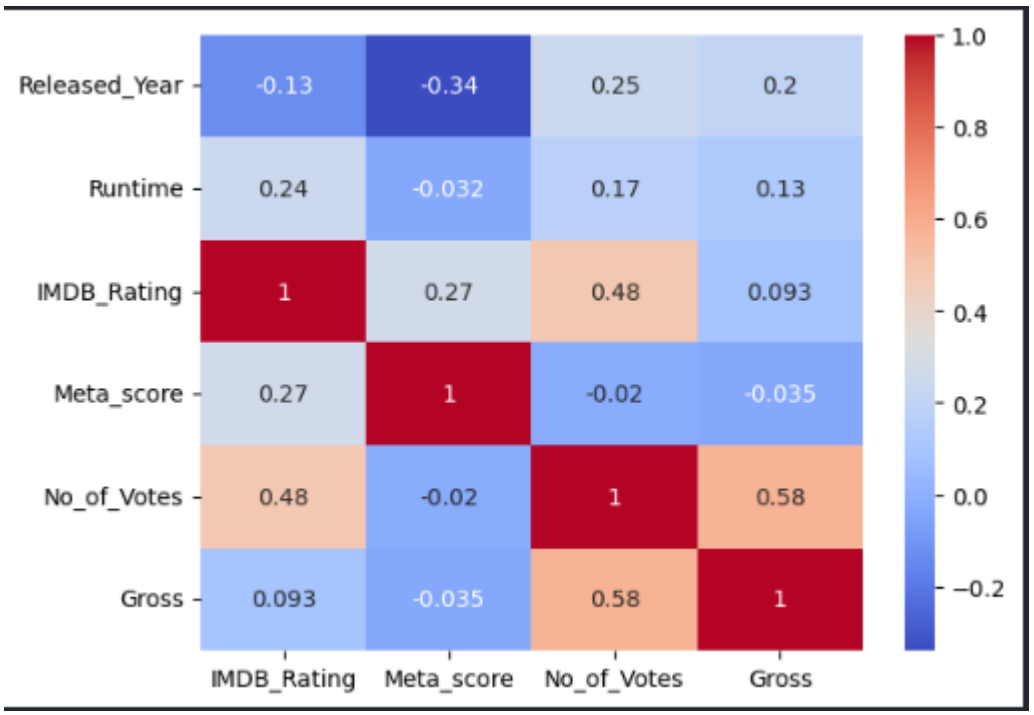
```
highest_rated_movie = df[df['IMDB_Rating'] == df['IMDB_Rating'].max()]
Nome_do_filme = highest_rated_movie['Series_Title'].values[0]
nota_do_filme = highest_rated_movie['IMDB_Rating'].values[0]
print(f"Eu iria recomendar {Nome_do_filme}, pois pela nota do IMDB ele
tem a nota {nota_do_filme}")
```

```
Eu iria recomendar The Godfather, pois pela nota do IMDB ele tem a nota 9.2
```

3 Principais fatores que estão relacionados com alta expectativa de faturamento :

Comprovamos que a Relação de faturamento tem correlação alta na quantidade de votos, isso significa que quanto maior a quantidade de votos é relacionado a quantidade do valor do faturamento, logo se tem vários votos terá um um faturamento alto.

```
correlation_matrix = df.select_dtypes(include=['number']).corr()
selected_columns = ['IMDB_Rating', 'Meta_score', 'No_of_Votes', 'Gross']
sns.heatmap(correlation_matrix[selected_columns], annot=True,
cmap='coolwarm')
plt.show()
```



3. Quais insights podem ser tirados com a coluna Overview? É possível inferior gênero filme a partir dessa coluna?

A partir da coluna 'Overview', é possível analisar as palavras-chave que podem sinalizar os principais temas do filme. Ao utilizar esses dados, realizamos cruzamentos com o gênero do filme. Além disso,é viável 3. Quais insights podem ser tirados com a coluna Overview? É possível inferior gênero filme a partir dessa coluna? aplicar algoritmos de inferência mais sofisticados utilizando aprendizado de máquina.

```
def infer_genre(overview):
    overview = overview.lower()
    overview = ''.join(e for e in overview if e.isalnum() or
e.isspace())
    overview = nltk.word_tokenize(overview)
    stop_words = set(nltk.corpus.stopwords.words('english'))
    overview = [word for word in overview if word not in stop_words]
     genre_keywords = {
        'action': ['fight', 'battle', 'explosion'],
        'comedy': ['funny', 'humor', 'laugh'],
        'drama': ['love', 'loss', 'family'],
        'horror': ['scary', 'ghost', 'monster'],
        'romance': ['love', 'relationship', 'kiss']
    }
    genre_scores = {}
    for genre, keywords in genre_keywords.items():
        score = sum(1 for word in overview if word in keywords)
```

```
        genre_scores[genre] = score
    inferred_genre = max(genre_scores, key=genre_scores.get)
    return inferred_genre
movie_overview = "An organized crime dynasty's aging patriarch transfers
control of his clandestine empire to his reluctant son."
inferred_genre = infer_genre(movie_overview)
print(f"O genero é : {inferred_genre}")
```

output:

```
···   O genero é : action
```