EDA

June 27, 2024

[1]:
```
import pandas as pd
df = pd.read_csv(r"C:\Users\gusta\Documents\desafio_LH\desafio_indicium_imdb.
↪csv")
```

[12]:
```
#Visualização Inicial:
df.drop(columns=['Unnamed: 0'], inplace=True)
print(df.head())
print(df.info())
print(df.describe())
```

```
                                  Series_Title Released_Year Certificate  \
0                                The Godfather          1972           A
1                              The Dark Knight          2008          UA
2                      The Godfather: Part II          1974           A
3                                  12 Angry Men          1957           U
4  The Lord of the Rings: The Return of the King          2003           U

   Runtime                     Genre  IMDB_Rating  \
0  175 min              Crime, Drama          9.2
1  152 min      Action, Crime, Drama          9.0
2  202 min              Crime, Drama          9.0
3   96 min              Crime, Drama          9.0
4  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
```

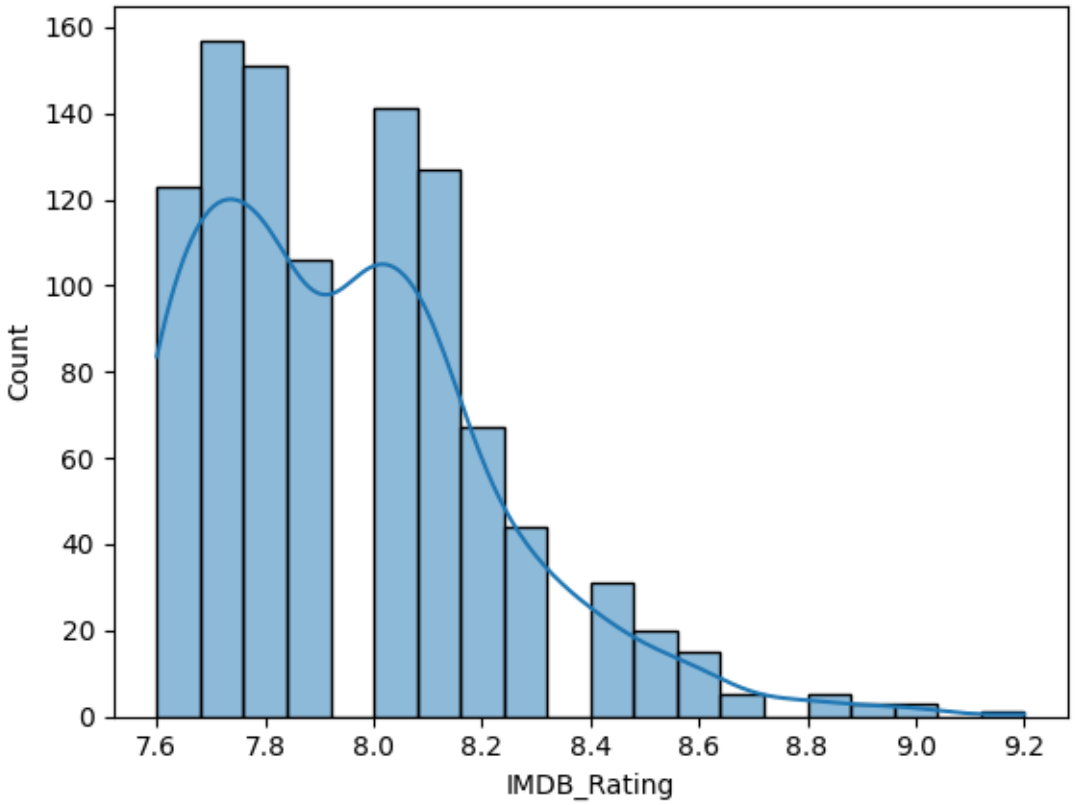
```
           Star4  No_of_Votes        Gross
0   Diane Keaton      1620367  134,966,411
1  Michael Caine      2303232  534,858,444
2   Diane Keaton      1129952   57,300,000
3   John Fiedler       689845    4,360,000
4  Orlando Bloom      1642758  377,845,905
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 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
None
       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
```

[5]:
```
# Encontrar dados faltantes para um possivel tratamento
missing_data = df.isnull().sum()
print(missing_data)
```

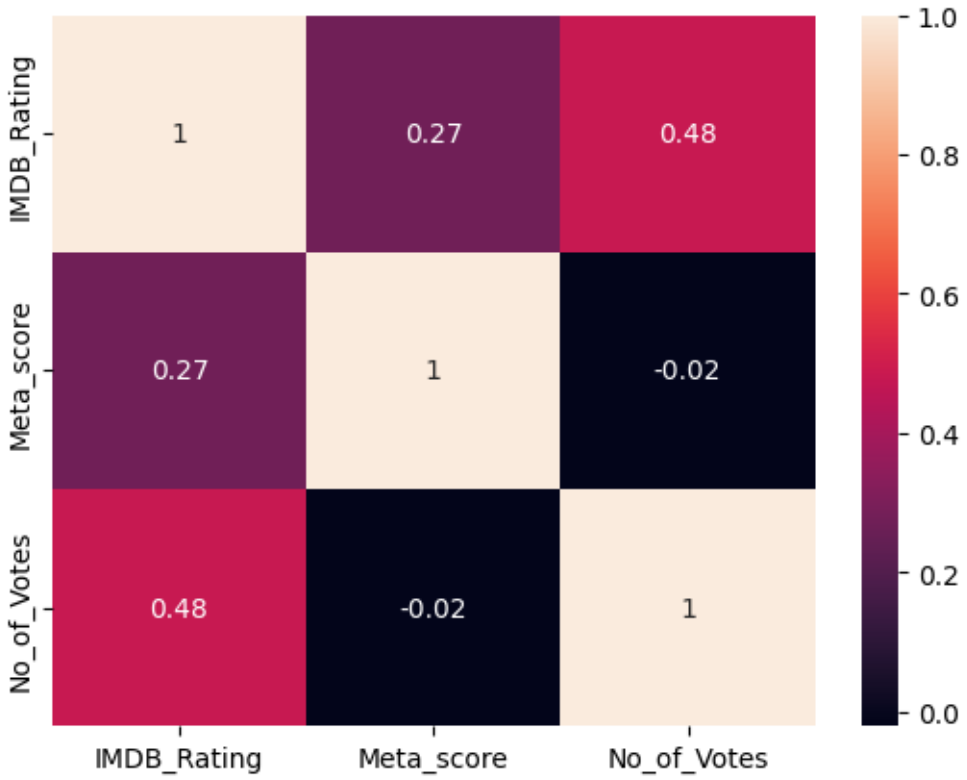
```
Unnamed: 0         0
Series_Title       0
Released_Year      0
Certificate      101
Runtime            0
```

```
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
```

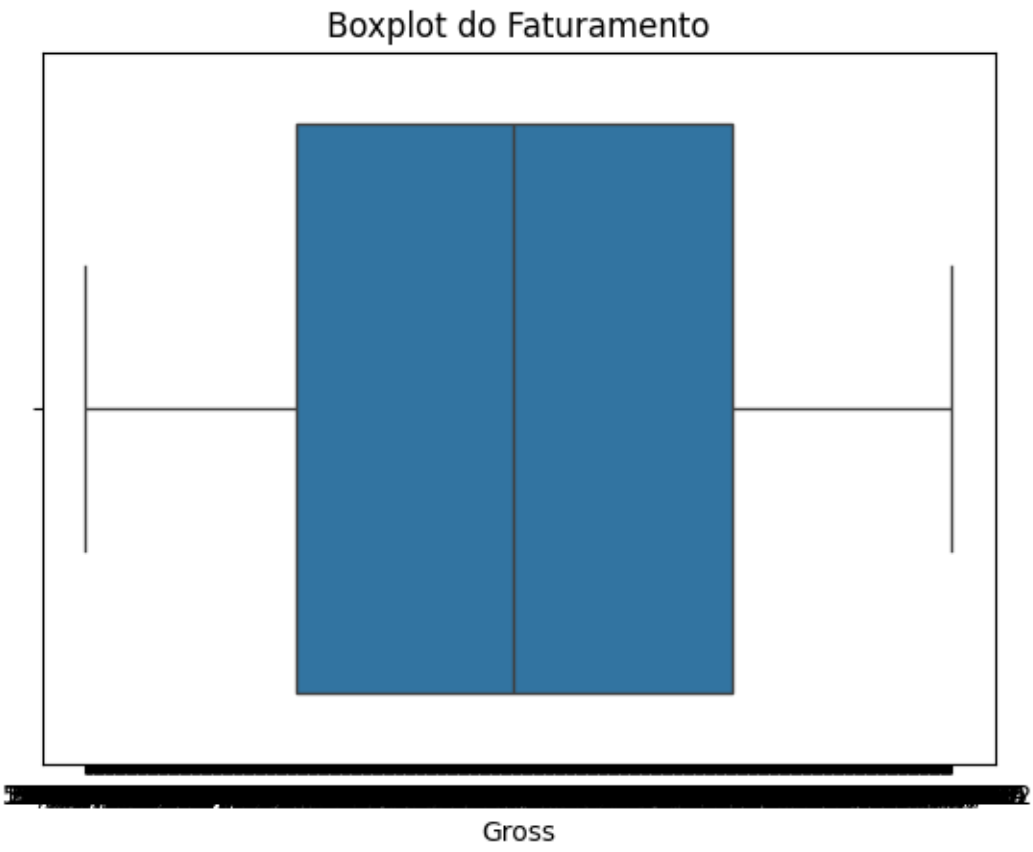
[10]:
```python
#Visualização de Distribuições:
import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(df['IMDB_Rating'], kde=True)
plt.show()
```



[14]:
```
#nálise de Correlação dos doados numericos
if 'Unnamed: 0' in df.columns:
    df.drop(columns=['Unnamed: 0'], inplace=True)
numeric_df = df.select_dtypes(include=['float64', 'int64'])
corr_matrix = numeric_df.corr()
sns.heatmap(corr_matrix, annot=True)
plt.show()
```



[15]:
```
#Exploração de Outliers:
sns.boxplot(x=df['Gross'])
plt.title('Boxplot do Faturamento')
plt.show()
```


Boxplot do Faturamento
Gross

[ ]: