respostas_as_perguntas

June 29, 2024

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

if 'Unnamed: 0' in df.columns:
    df.drop(columns=['Unnamed: 0'], inplace=True)
```

[2]:
```
# 2a. Qual filme você recomendaria para uma pessoa que você não conhece?

# Calcular a média entre IMDB Rating e Meta Score
df['IMDB_Meta_avg'] = (df['IMDB_Rating'] + (df['Meta_score'] / 10)) / 2

# Selecionar o filme com a maior média
filme_recomendado = df.loc[df['IMDB_Meta_avg'].idxmax()]
print(f"Filme recomendado: {filme_recomendado['Series_Title']}")
print(f"IMDB Rating: {filme_recomendado['IMDB_Rating']}")
print(f"Meta Score: {filme_recomendado['Meta_score']}")
```

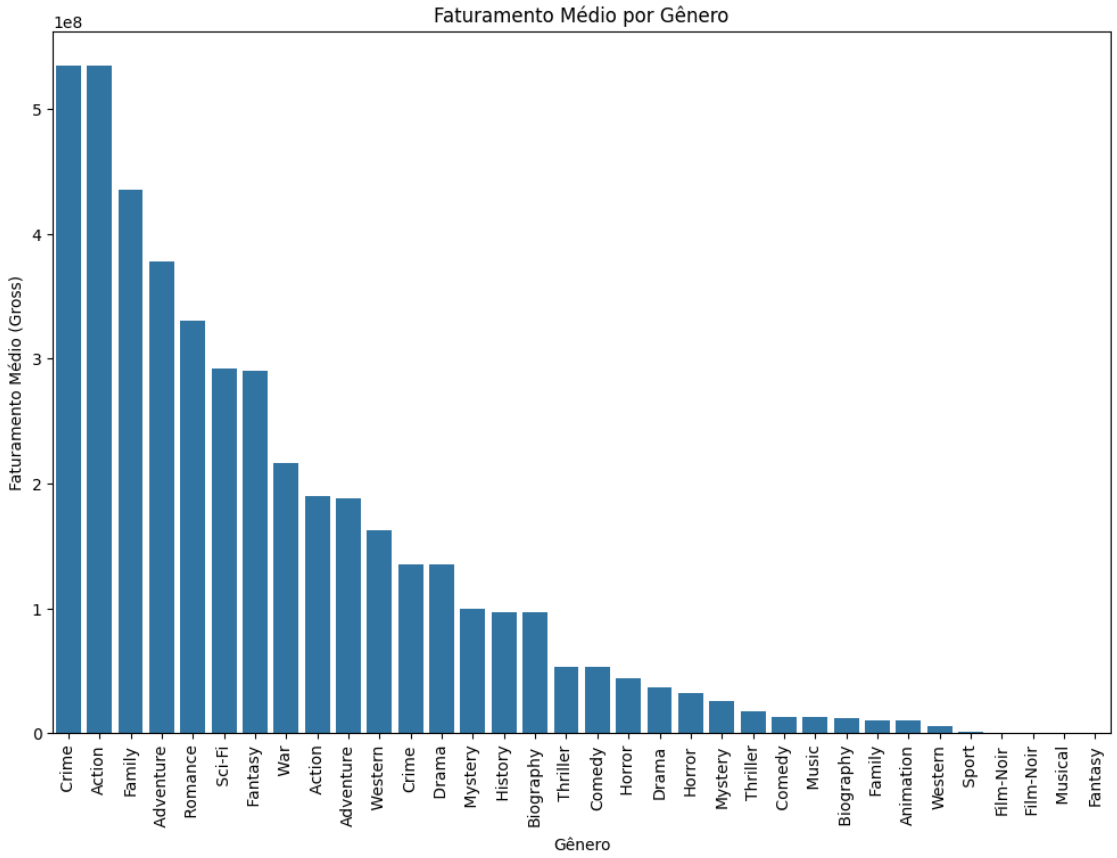
```
Filme recomendado: The Godfather
IMDB Rating: 9.2
Meta Score: 100.0
```

[3]:
```
# 2b. Quais são os principais fatores que estão relacionados com alta␣
 ↪expectativa de faturamento de um filme?
#podemos realisar uma analise dos dados numericos e encontrar sua correlação␣
 ↪com o faturamento de um filme, para isso:

# Limpar e converter a coluna 'Gross'
df['Gross'] = df['Gross'].str.replace(',', '')  # Remover separadores de␣
 ↪milhares
df['Gross'] = pd.to_numeric(df['Gross'], errors='coerce')  # Converter para␣
 ↪numérico, substituindo erros por NaN

# Verificar as colunas numéricas
numeric_df = df.select_dtypes(include=['float64', 'int64'])
```

```
# Calcular a matriz de correlação
corr_matrix = numeric_df.corr()

# Ordenar correlações com a coluna 'Gross'
corr_factors = corr_matrix['Gross'].sort_values(ascending=False)
print(corr_factors)
```

```
Gross            1.000000
No_of_Votes      0.589527
IMDB_Rating      0.099393
IMDB_Meta_avg   -0.000268
Meta_score      -0.030480
Name: Gross, dtype: float64
```

[6]:
```
#podemos encontrar outras correlações, como com o genero do filme

# Separar os gêneros
df_genres = df['Genre'].str.split(',', expand=True).stack().
 ↪reset_index(level=1, drop=True)
df_genres.name = 'Genre'

# Combinar os gêneros separados com o DataFrame original
df = df.drop(columns=['Genre']).join(df_genres).reset_index(drop=True)

# Remover duplicatas
df = df.drop_duplicates(subset=['Genre'])

# Calcular o faturamento médio por gênero
faturamento_por_genero = df.groupby('Genre')['Gross'].mean().
 ↪sort_values(ascending=False)

# Visualizar os resultados com um gráfico de barras
plt.figure(figsize=(12, 8))
sns.barplot(x=faturamento_por_genero.index, y=faturamento_por_genero.values)
plt.title('Faturamento Médio por Gênero')
plt.xlabel('Gênero')
plt.ylabel('Faturamento Médio (Gross)')
plt.xticks(rotation=90)  # Rotacionar os rótulos no eixo x se necessário
plt.show()
```



[13]:
```
# 2c. Quais insights podem ser tirados com a coluna Overview? É possível␣
 ↪inferir o gênero do filme a partir dessa coluna?

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB

# Prever gênero a partir do Overview
vectorizer = CountVectorizer(stop_words='english')
X = vectorizer.fit_transform(df['Overview'])
y = df['Genre']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
model = MultinomialNB()
model.fit(X_train, y_train)
accuracy = model.score(X_test, y_test)
```

[12]:
```
# podemos usar esse modelo simples para prever com certa precisão o gênero de␣
 ↪um filme
novo_resumo = ["An organized crime dynasty's aging patriarch transfers control␣
 ↪of his clandestine empire to his reluctant son."]

# Transformar o resumo
X_novo = vectorizer.transform(novo_resumo)

# Fazer a previsão
previsao = model.predict(X_novo)

# Mostrar o gênero previsto
print(f'Gênero previsto: {previsao[0]}')
```

```
Gênero previsto:  Drama
```

[ ]:
```
#3 Como você faria a previsão da nota do IMDb a partir dos dados?
#Usaria um modelo que, atravez de um treino usando os dados do arquivo␣
 ↪desafio_indicium_imdb.csv seria capaz de prever o IMDb de um filme
```

[ ]:
```
#3 Quais variáveis e/ou suas transformações você utilizou e por quê?
#Variáveis Numéricas: Incluem ano de lançamento, duração, número de votos,␣
 ↪receita bruta, etc.
#Essas variáveis geralmente são mantidas em seu formato numérico original, ou␣
 ↪convertidas se estiverem em formato de texto.

#Variáveis Categóricas: Incluem gênero, certificado, diretor, etc.
#Essas variáveis são transformadas em valores numéricos usando técnicas como␣
 ↪Label Encoding ou One-Hot Encoding, para que possam ser processadas por␣
 ↪algoritmos de machine learning.
```

[ ]:
```
#3 Qual tipo de problema estamos resolvendo (regressão, classificação)?
#Estamos resolvendo um problema de regressão, onde o objetivo é prever um valor␣
 ↪contínuo (a nota do IMDb).
```

[ ]:
```
#3  Qual modelo melhor se aproxima dos dados e quais seus prós e contras?
#Um modelo adequado para esse tipo de problema é o Random Forest Regressor.
```

[ ]:
```
#3 Qual medida de performance do modelo foi escolhida e por quê?
#R2 Score (Coeficiente de Determinação): Mede a proporção da variância da␣
 ↪variável dependente explicada pelo modelo.
#É útil para entender o quão bem os dados de treino se ajustam ao modelo.

#RMSE (Root Mean Squared Error): Mede a diferença média entre os valores␣
 ↪previstos e os valores reais, na mesma unidade da variável dependente.
#É uma métrica fácil de interpretar e fornece uma ideia da precisão do modelo.
```

```
#Essas métricas foram escolhidas porque oferecem uma visão clara da performance␣
↪do modelo em termos de explicação da variância e precisão das predições.
```