project_imdbreview

October 19, 2024

IMDb MOVIE REVIEWS PROJECT

[3]:
```
# Import necesary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from bs4 import BeautifulSoup
import re
import requests
```

Step 1: Webpage Request

[4]:
```
# Fetching movies list from the IMDB website
moviesurl = "https://www.imdb.com/search/title/?title_type=feature"
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36␣
 ↪(KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'
}

# Request to fetch the data from the URL
pgrequest = requests.get(moviesurl, headers=headers)
if pgrequest.status_code == 200:
    print(f"Webpage was successfully fetched. STATUS CODE: {pgrequest.
 ↪status_code}")
else:
    print(f"Error in retrieving the webpage. STATUS CODE: {pgrequest.
 ↪status_code}")
```

```
Webpage was successfully fetched. STATUS CODE: 200
```

Step 2: Parsing the HTML data content - Using beautifulsoup for webscrapping

[5]:
```
soup = BeautifulSoup(pgrequest.text, 'html.parser')
print(type(soup))
#print(soup.prettify())
```

```
<class 'bs4.BeautifulSoup'>
```

[6]:
```
# scrapped movie names
movies = soup.find_all("li", class_="ipc-metadata-list-summary-item")
print(f"Total number of movies found: {len(movies)}")
```

```
Total number of movies found: 25
```

Step 3:Extract the movie details

[7]:
```
# Initialize the list
movies_data = []
# Iterating through each movie and getting relevant details
for movie in movies:
    title = movie.find('h3', class_="ipc-title__text").text.split('.')[1]
    year = movie.find("div", class_="sc-732ea2d-5 kHnTQb dli-title-metadata").
 ↪find_all('span')[0].text
    duration = movie.find("div", class_="sc-732ea2d-5 kHnTQb␣
 ↪dli-title-metadata").find_all('span')[1].text
    film_rating = movie.find("div", class_="sc-732ea2d-5 kHnTQb␣
 ↪dli-title-metadata").find_all('span')[2].text
    star_rating = movie.find("span", class_="ipc-rating-star--rating").text if␣
 ↪movie.find("span", class_="ipc-rating-star--rating") else np.nan
    voteCount = movie.find("span", class_="ipc-rating-star--voteCount").text if␣
 ↪movie.find("span", class_="ipc-rating-star--voteCount") else np.nan
    metascore = movie.find("span", class_="sc-b0901df4-0 bXIOoL␣
 ↪metacritic-score-box").text if movie.find("span", class_="sc-b0901df4-0␣
 ↪bXIOoL metacritic-score-box") else np.nan
    description= movie.find("div", class_="ipc-html-content-inner-div").
 ↪get_text(strip=True) if movie.find("div",␣
 ↪class_="ipc-html-content-inner-div") else np.nan

    movies_data.append({
        "Movie Title": title,
        "Release Year": year,
        "Movie Duration":duration,
        "MPA Rating":film_rating,
        "Audience Rating":star_rating,
        "Audience Votes":voteCount,
        "Metascore":metascore,
        "Movie Description":description
    })
```

[8]:
```
#movies_data
# Convert the movie details data into pandas dataframe
df = pd.DataFrame(movies_data)
df
```

[8]:

	Movie Title	Release Year	Movie Duration	MPA Rating	Audience Rating	Audience Votes	Metascore
0	Joker: Folie à Deux	2024	2h 18m	R	5.3	(80K)	45
1	Terrifier 3	2024	2h 5m	Not Rated	7.1	(9.3K)	61
2	The Substance	2024	2h 21m	R	7.7	(57K)	78
3	Salem's Lot	2024	1h 54m	R	5.7	(20K)	47
4	Beetlejuice Beetlejuice	2024	1h 45m	PG-13	6.9	(77K)	62
5	It's What's Inside	2024	1h 43m	R	6.6	(16K)	57
6	Deadpool & Wolverine	2024	2h 8m	R	7.8	(337K)	56
7	Speak No Evil	2024	1h 50m	R	6.9	(44K)	66
8	The Platform 2	2024	1h 39m	TV-MA	5.0	(23K)	45
9	Megalopolis	2024	2h 18m	R	5.0	(15K)	55
10	Joker	2019	2h 2m	R	8.4	(1.5M)	59
11	The Wild Robot	2024	1h 42m	PG	8.4	(28K)	85
12	Wolfs	2024	1h 48m	R	6.5	(40K)	60
13	Saturday Night	2024	1h 49m	R	7.4	(4K)	63
14	Beetlejuice	1988	1h 32m	PG	7.5	(381K)	71
15	Terrifier	2016	1h 25m	Unrated	5.6	(64K)	NaN
16	Terrifier 2	2022	2h 18m	Not Rated	6.1	(56K)	59
17	Hellboy: The Crooked Man	2024	1h 39m	R	4.5	(5.8K)	45
18	Daddio	2023	1h 40m	R	6.6	(6.9K)	62
19	The Platform	2019	1h 34m	TV-MA			
20	The Apprentice	2024	2h 2m	R			
21	Caddo Lake	2024	1h 39m	PG-13			
22	Strange Darling	2023	1h 37m	R			
23	Inside Out 2	2024	1h 36m	PG			
24	Blink Twice	2024	1h 42m	R			

```
19              7.0         (291K)         73
20              7.1         (5.1K)         64
21              6.9          (10K)         55
22              7.2          (19K)         80
23              7.6         (161K)         73
24              6.5          (45K)         66
```

```
                                    Movie Description
0   Struggling with his dual identity, failed come…
1   Art the Clown is set to unleash chaos on the u…
2   A fading celebrity takes a black-market drug: …
3   An author returns to his hometown of Jerusalem…
4   After a family tragedy, three generations of t…
5   A group of friends gather for a pre-wedding pa…
6   Deadpool is offered a place in the Marvel Cine…
7   A family is invited to spend a whole weekend i…
8   A thrilling physical journey that allows an ap…
9   The city of New Rome faces the duel between Ce…
10  Arthur Fleck, a party clown and a failed stand…
11  After a shipwreck, an intelligent robot called…
12  Two rival fixers cross paths when they're both…
13  At 11:30pm on October 11th, 1975, a ferocious …
14  The spirits of a deceased couple are harassed …
15  A maniac named Art the Clown terrorizes two fr…
16  After being resurrected by a sinister entity, …
17  Hellboy and a rookie B.P.R.D. agent in the 195…
18  A woman taking a cab ride from JFK engages in …
19  In a prison where inmates are fed on a descend…
20  The story of how a young Donald Trump started …
21  When an 8-year-old girl disappears on Caddo La…
22  Nothing is what it seems when a twisted one-ni…
23  A sequel that features Riley entering puberty …
24  When tech billionaire Slater King meets cockta…
```

[136]:
```
df.to_csv('IMDB_Scrappedmovies.csv', index=False)
print("Initial scrapped movies data saved in CSV format")
```

```
Initial scrapped movies data saved in CSV format
```

Step 4: Data Cleaning - Check for None or NaN value: need to amend the **None** value - Check for missing values - Check for duplicate movie name entry, & maybe movie description! (only work for movie title; rest field can have same/duplicate values) - Audience Votes column has brackets: need to remove the brackets from the values

[20]:
```
# Count None values per row
# This should also give us missing values/ null values OR else could use df.
  ↪isnull function
nan_counts = df.isna().sum(axis=1)
```

```
print(f"Total number of NaN values:\n {df.isna().sum()}")
```

```
Total number of NaN values:
 Movie Title          0
Release Year          0
Movie Duration        0
MPA Rating            0
Audience Rating       0
Audience Votes        0
Metascore             1
Movie Description     0
dtype: int64
```

[55]:
```
duplicates = df.duplicated(keep=False)
df.duplicated().sum()
print(f"There are {df.duplicated().sum()} duplicate entries.")
```

```
There are 0 duplicate entries.
```

[56]:
```
duplicates_movie_title = df['Movie Title'].duplicated(keep=False)
duplicates_movie_description = df['Movie Description'].duplicated(keep=False)
print(f"There are {duplicates_movie_title.sum()} duplicates in Movie Title␣
 ↪column, and {duplicates_movie_description.sum()} duplicates in Movie␣
 ↪description column.")
```

```
There are 0 duplicates in Movie Title column, and 0 duplicates in Movie
description column.
```

[57]:
```
# N/A or non-available metascores are assigned integer value 0
df['Metascore'] = df['Metascore'].map(lambda x: 0 if x is None else x)
# Metascore has 'Nonetype' attribute: Converting it into numeric integer value␣
 ↪for future use
df['Metascore'] = pd.to_numeric(df['Metascore'], errors='coerce').fillna(0).
 ↪astype(int)
```

[58]:
```
# Removing the brackets from elements of Audience Votes column
df['Audience Votes'] = df['Audience Votes'].str.replace(r'[\[\]\(\)]', '',␣
 ↪regex=True)
df
```

[58]:

	Movie Title	Release Year	Movie Duration	MPA Rating	\
0	Joker: Folie à Deux	2024	2h 18m	R	
1	Terrifier 3	2024	2h 5m	Not Rated	
2	The Substance	2024	2h 21m	R	
3	Salem's Lot	2024	1h 54m	R	
4	Beetlejuice Beetlejuice	2024	1h 45m	PG-13	
5	It's What's Inside	2024	1h 43m	R	
6	Deadpool & Wolverine	2024	2h 8m	R	
7	Speak No Evil	2024	1h 50m	R	

8	The Platform 2	2024	1h 39m	TV-MA
9	Megalopolis	2024	2h 18m	R
10	Joker	2019	2h 2m	R
11	The Wild Robot	2024	1h 42m	PG
12	Wolfs	2024	1h 48m	R
13	Saturday Night	2024	1h 49m	R
14	Beetlejuice	1988	1h 32m	PG
15	Terrifier	2016	1h 25m	Unrated
16	Terrifier 2	2022	2h 18m	Not Rated
17	Hellboy: The Crooked Man	2024	1h 39m	R
18	Daddio	2023	1h 40m	R
19	The Platform	2019	1h 34m	TV-MA
20	The Apprentice	2024	2h 2m	R
21	Caddo Lake	2024	1h 39m	PG-13
22	Strange Darling	2023	1h 37m	R
23	Inside Out 2	2024	1h 36m	PG
24	Blink Twice	2024	1h 42m	R

	Audience Rating	Audience Votes	Metascore	\
0	5.3	80K	45	
1	7.1	9.3K	61	
2	7.7	57K	78	
3	5.7	20K	47	
4	6.9	77K	62	
5	6.6	16K	57	
6	7.8	337K	56	
7	6.9	44K	66	
8	5.0	23K	45	
9	5.0	15K	55	
10	8.4	1.5M	59	
11	8.4	28K	85	
12	6.5	40K	60	
13	7.4	4K	63	
14	7.5	381K	71	
15	5.6	64K	0	
16	6.1	56K	59	
17	4.5	5.8K	45	
18	6.6	6.9K	62	
19	7.0	291K	73	
20	7.1	5.1K	64	
21	6.9	10K	55	
22	7.2	19K	80	
23	7.6	161K	73	
24	6.5	45K	66	

	Movie Description
0	Struggling with his dual identity, failed come…

```
1   Art the Clown is set to unleash chaos on the u…
2   A fading celebrity takes a black-market drug: …
3   An author returns to his hometown of Jerusalem…
4   After a family tragedy, three generations of t…
5   A group of friends gather for a pre-wedding pa…
6   Deadpool is offered a place in the Marvel Cine…
7   A family is invited to spend a whole weekend i…
8   A thrilling physical journey that allows an ap…
9   The city of New Rome faces the duel between Ce…
10  Arthur Fleck, a party clown and a failed stand…
11  After a shipwreck, an intelligent robot called…
12  Two rival fixers cross paths when they're both…
13  At 11:30pm on October 11th, 1975, a ferocious …
14  The spirits of a deceased couple are harassed …
15  A maniac named Art the Clown terrorizes two fr…
16  After being resurrected by a sinister entity, …
17  Hellboy and a rookie B.P.R.D. agent in the 195…
18  A woman taking a cab ride from JFK engages in …
19  In a prison where inmates are fed on a descend…
20  The story of how a young Donald Trump started …
21  When an 8-year-old girl disappears on Caddo La…
22  Nothing is what it seems when a twisted one-ni…
23  A sequel that features Riley entering puberty …
24  When tech billionaire Slater King meets cockta…
```

Step 5: Data Transformation - Convert movie duration to minutes - Convert Audience Votes to numeric format (by removing 'K' and 'M') - Handle missing data by filling the values with relevent values - Tokenize the movie description for sentiment analysis (I will use Textblob library)

[59]:
```python
def convert_duration_to_minutes(duration):
    if isinstance(duration, str):
        parts = duration.split()
        hours = int(parts[0][:-1])  # Remove 'h' and convert to int
        minutes = int(parts[1][:-1])  # Remove 'm' and convert to int
        return hours * 60 + minutes
    return None
df['Movie Duration'] = df['Movie Duration'].apply(convert_duration_to_minutes)
df.head()
```

[59]:

	Movie Title	Release Year	Movie Duration	MPA Rating	Audience Rating	Audience Votes	Metascore
0	Joker: Folie à Deux	2024	138	R	5.3	80K	45
1	Terrifier 3	2024	125	Not Rated			
2	The Substance	2024	141	R			
3	Salem's Lot	2024	114	R			
4	Beetlejuice Beetlejuice	2024	105	PG-13			

```
1             7.1          9.3K         61
2             7.7           57K         78
3             5.7           20K         47
4             6.9           77K         62

                                  Movie Description
0  Struggling with his dual identity, failed come…
1  Art the Clown is set to unleash chaos on the u…
2  A fading celebrity takes a black-market drug: …
3  An author returns to his hometown of Jerusalem…
4  After a family tragedy, three generations of t…
```

[60]:
```python
# Clean Audience Votes: Remove non-numeric characters (like 'K' for thousands) and convert to numeric
def convert_audience_votes(votes):
    if isinstance(votes, str):
        votes = votes.strip().upper()  # Remove spaces and standardize to uppercase
        if 'K' in votes:
            votes = votes.replace('K', '')
            return float(votes) * 1000  # Convert 'K' to thousands
        elif 'M' in votes:
            votes = votes.replace('M', '')
            return float(votes) * 1000000  # Convert 'M' to millions
        else:
            # Remove any other non-numeric characters and convert to numeric
            votes = votes.replace(',', '').replace(' ', '')
    return pd.to_numeric(votes, errors='coerce')  # Handle other cases like plain numbers

# Apply this function to the 'Audience Votes' column
df['Audience Votes'] = df['Audience Votes'].apply(convert_audience_votes)
df.head()
```

[60]:
```
                Movie Title Release Year  Movie Duration MPA Rating  \
0       Joker: Folie à Deux         2024             138          R
1                Terrifier 3         2024             125  Not Rated
2             The Substance         2024             141          R
3               Salem's Lot         2024             114          R
4   Beetlejuice Beetlejuice         2024             105      PG-13

  Audience Rating  Audience Votes  Metascore  \
0             5.3         80000.0         45
1             7.1          9300.0         61
2             7.7         57000.0         78
3             5.7         20000.0         47
4             6.9         77000.0         62
```

```
                                    Movie Description
0  Struggling with his dual identity, failed come…
1  Art the Clown is set to unleash chaos on the u…
2  A fading celebrity takes a black-market drug: …
3  An author returns to his hometown of Jerusalem…
4  After a family tragedy, three generations of t…
```

Step 5: Tokenization - Tokenize movie description for future analysis. - Future reference: Perform sentiment analysis to describe mood (liveliness, neutral, gloomy) of the movie

[61]:
```
# Defining a custom tokenzier that removes duplicates
def customtokenizer(text):
    # Lowercase the text
    text = text.lower()

    # Removing punctuation marks.
    punctuation = ['!', '.', ',', '?', ';', ':', '-', '(', ')', '[', ']', '{',␣
 ↪'}', "'", '"']
    # Loop through each punctuation character and replace it with an empty␣
 ↪string
    for p in punctuation:
        text = text.replace(p, "")
    tokens = text.split()

    # Remove duplicates while preserving order
    seenwords = set()
    tokens2 = [token for token in tokens if not (token in seenwords or␣
 ↪seenwords.add(token))]

    return tokens2
```

[62]:
```
# Applying the custom tokenizer to the Movie Description column in df
df['Tokens'] = df['Movie Description'].apply(customtokenizer)
df.head()
```

[62]:
```
                Movie Title Release Year  Movie Duration MPA Rating  \
0       Joker: Folie à Deux         2024             138          R
1                Terrifier 3         2024             125  Not Rated
2             The Substance         2024             141          R
3                Salem's Lot         2024             114          R
4   Beetlejuice Beetlejuice         2024             105      PG-13

  Audience Rating  Audience Votes  Metascore  \
0             5.3         80000.0         45
1             7.1          9300.0         61
2             7.7         57000.0         78
```

```
3              5.7         20000.0         47
4              6.9         77000.0         62

                                    Movie Description  \
0  Struggling with his dual identity, failed come…
1  Art the Clown is set to unleash chaos on the u…
2  A fading celebrity takes a black-market drug: …
3  An author returns to his hometown of Jerusalem…
4  After a family tragedy, three generations of t…

                                              Tokens
0  [struggling, with, his, dual, identity, failed…
1  [art, the, clown, is, set, to, unleash, chaos,…
2  [a, fading, celebrity, takes, blackmarket, dru…
3  [an, author, returns, to, his, hometown, of, j…
4  [after, a, family, tragedy, three, generations…
```

[63]:
```
df[['Movie Title', 'Tokens']].head()
```

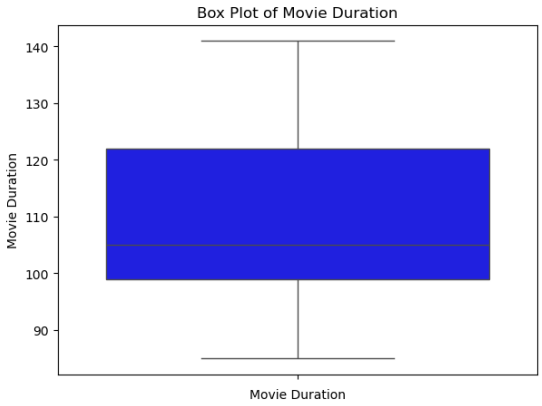
[63]:
```
                Movie Title                                             Tokens
0       Joker: Folie à Deux  [struggling, with, his, dual, identity, failed…
1                Terrifier 3  [art, the, clown, is, set, to, unleash, chaos,…
2              The Substance  [a, fading, celebrity, takes, blackmarket, dru…
3                Salem's Lot  [an, author, returns, to, his, hometown, of, j…
4  Beetlejuice Beetlejuice    [after, a, family, tragedy, three, generations…
```

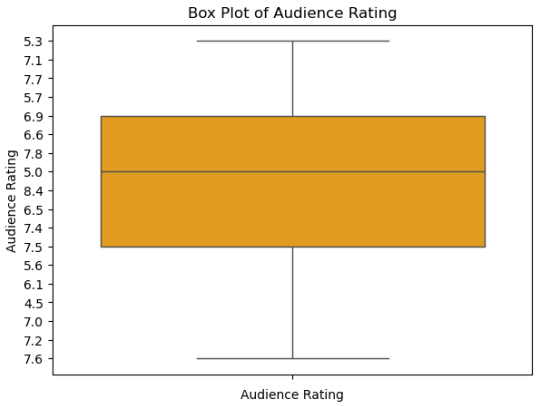
Data Visualization -Relevant numeric data for comparison: Movie Duration, Audience Rating and (maybe! Audience votes)

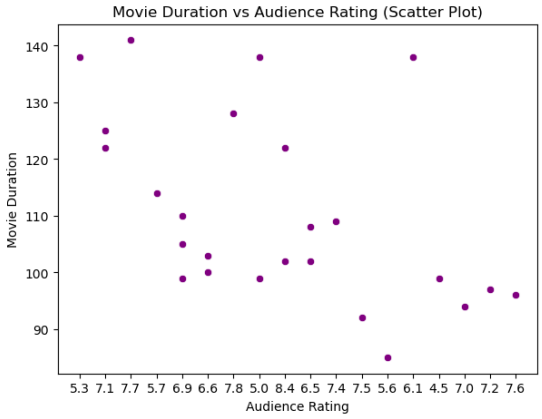
[64]:
```
#Box plot
plt.figure(figsize = (15,17))

# Box plot for Movie Duration
plt.subplot(3,2,1)
sns.boxplot(df['Movie Duration'], color = 'blue')
plt.title('Box Plot of Movie Duration')
plt.xlabel('Movie Duration')

# Box plot for Audience Rating
plt.subplot(3,2,2)
sns.boxplot(df['Audience Rating'], color = 'orange')
plt.title('Box Plot of Audience Rating')
plt.xlabel('Audience Rating')

# Scatter plot for Audience Rating and Movie Duration
plt.subplot(3, 2, 3)
sns.scatterplot(x=df['Audience Rating'], y=df['Movie Duration'], color='purple')
plt.title('Movie Duration vs Audience Rating (Scatter Plot)')
```

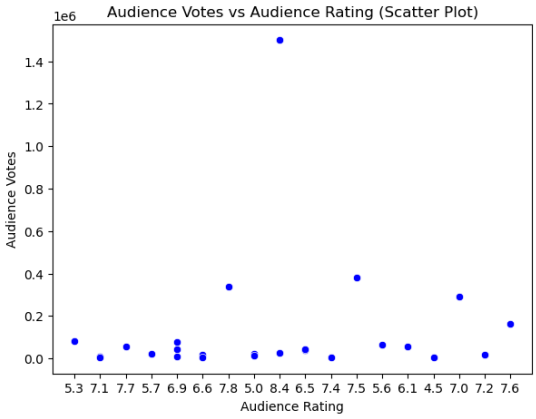
```
plt.xlabel('Audience Rating')
plt.ylabel('Movie Duration')

# Scatter plot for Audience Rating and Audience Votes
plt.subplot(3, 2, 4)
sns.scatterplot(x=df['Audience Rating'], y=df['Audience Votes'], color='blue')
plt.title('Audience Votes vs Audience Rating (Scatter Plot)')
plt.xlabel('Audience Rating')
plt.ylabel('Audience Votes')

# Scatter plot for Audience Rating and Metascore
plt.subplot(3, 2, 5)
sns.scatterplot(x=df['Audience Rating'], y=df['Metascore'], color='red')
plt.title('Metascore vs Audience Rating (Scatter Plot)')
plt.xlabel('Audience Rating')
plt.ylabel('Metascore')
```

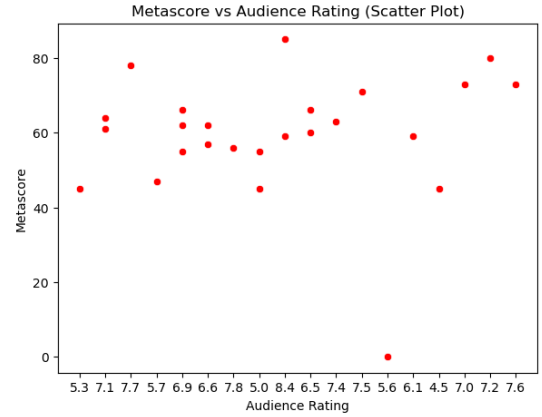
[64]: `Text(0, 0.5, 'Metascore')`



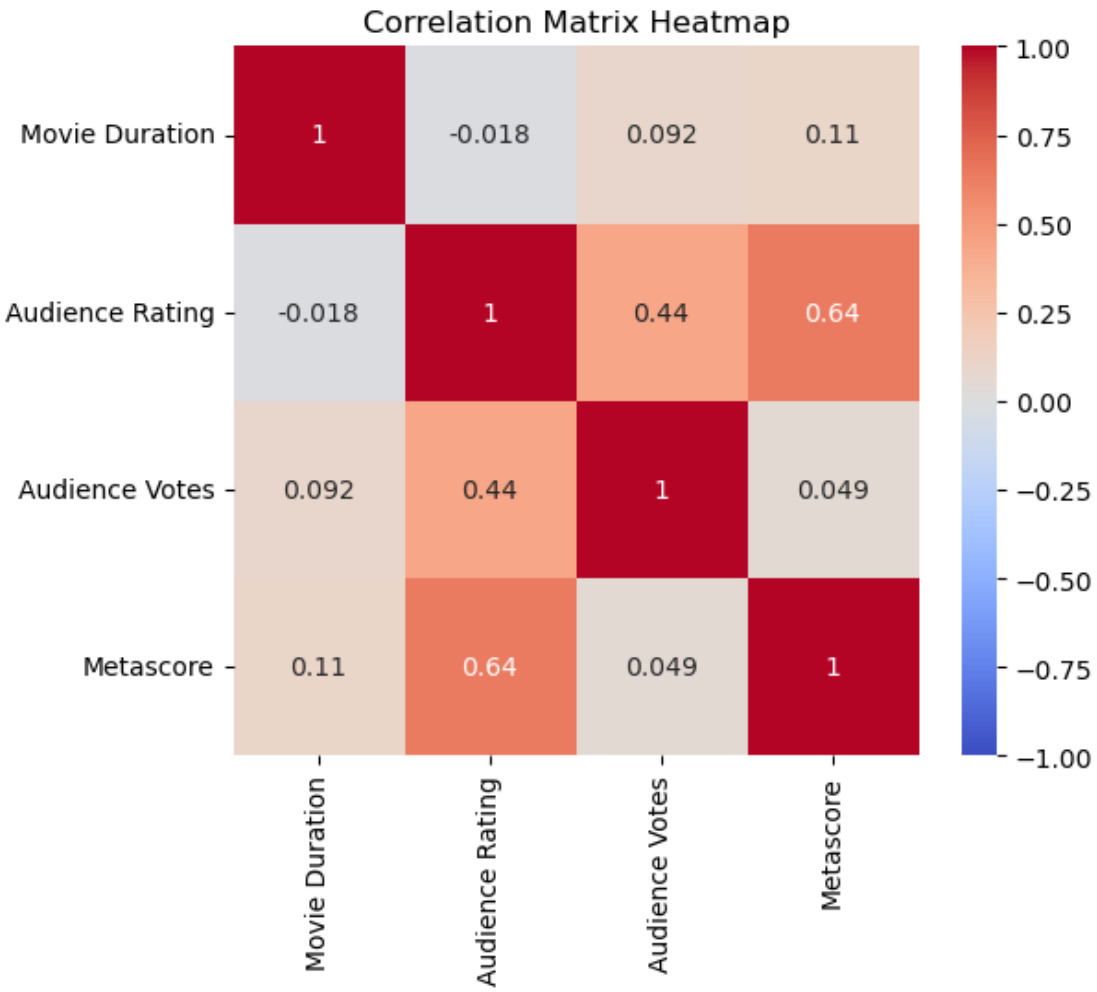








[65]:
```
# Correlation Matrix Heatmap
df_reduced = df.filter(items=['Movie Duration', 'Audience Rating', 'Audience␣
 ↪Votes', 'Metascore'])
correlation_matrix = df_reduced.corr()
plt.figure(figsize = (6,5))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Matrix Heatmap')
plt.show()
```



Step 6: Save the final customized data to csv file

[66]:
```
df.to_csv('IMDBmoviesfinal.csv', index=False)
print("Final movies data saved in CSV format")
```

```
Final movies data saved in CSV format
```