Exploring Top-Performing Films and Genres for New Movie Studio

Introduction

In the rapidly evolving entertainment industry, original video content has become a significant driver of growth and audience engagement. Noticing this trend, Our Company has decided to venture into the movie production arena by establishing a new movie studio. However, the company lacks experience in film creation and is uncertain about which types of films will perform best at the box office.

To ensure the success of this new initiative, it is crucial to make data-driven decisions about the types of films to produce. This project aims to explore the current landscape of top-performing films at the box office using data from IMDb and box office records. By analyzing trends and patterns in the data, we will provide actionable insights to guide the head of the new movie studio in making informed decisions about film production.

Objectives

1. **Data Collection:**

- Load CSV Data using Pandas, data from box office.

In [1]:
```
import pandas as pd

# Load the CSV data into a pandas DataFrame
file_path = './data/bom.movie_gross.csv'
csv_df = pd.read_csv(file_path)

# Display the first few rows of the DataFrame
print(csv_df.head())

# Ensure numeric columns are of correct type
csv_df['domestic_gross'] = pd.to_numeric(csv_df['domestic_gross'], errors
='coerce')
csv_df['foreign_gross'] = pd.to_numeric(csv_df['foreign_gross'], errors='co
erce')

# Check for missing values
print(csv_df.isnull().sum())

# Fill or drop missing values as appropriate (here we will drop them)
csv_df.dropna(subset=['domestic_gross', 'foreign_gross'], inplace=True)

# Check the data types
print(csv_df.dtypes)

# Convert 'year' to integer if necessary
csv_df['year'] = csv_df['year'].astype(int)

# Display basic statistics
print(csv_df.describe())
```

```
                                          title studio  domestic_gross  \
0                                   Toy Story 3     BV     415000000.0   
1                    Alice in Wonderland (2010)     BV     334200000.0   
2  Harry Potter and the Deathly Hallows Part 1     WB     296000000.0   
3                                     Inception     WB     292600000.0   
4                           Shrek Forever After   P/DW     238700000.0   

  foreign_gross  year  
0     652000000  2010  
1     691300000  2010  
2     664300000  2010  
3     535700000  2010  
4     513900000  2010  
title                0
studio               5
domestic_gross      28
foreign_gross     1355
year                 0
dtype: int64
title              object
studio             object
domestic_gross    float64
foreign_gross     float64
year                int64
dtype: object
       domestic_gross  foreign_gross         year
count    2.004000e+03   2.004000e+03  2004.000000
mean     4.566975e+07   7.590713e+07  2013.497006
std      7.637549e+07   1.382501e+08     2.597954
min      4.000000e+02   6.000000e+02  2010.000000
25%      6.617500e+05   3.900000e+06  2011.000000
50%      1.635000e+07   1.955000e+07  2013.000000
75%      5.570000e+07   7.615000e+07  2016.000000
max      7.001000e+08   9.605000e+08  2018.000000
```

- Load SQLite Data using Pandas, data from IMDb

In [2]:
```
import sqlite3

# Connect to the SQLite database
conn = sqlite3.connect('./data/im.db')

# Load SQLite data into a DataFrame
sqlite_query = '''
    SELECT mb.movie_id, mb.primary_title, mb.genres, mb.start_year, mr.aver
agerating, mr.numvotes
    FROM movie_basics mb
    JOIN movie_ratings mr ON mb.movie_id = mr.movie_id
'''
sqlite_df = pd.read_sql_query(sqlite_query, conn)

# Close the connection
conn.close()

print(sqlite_df.head())
```

```
    movie_id                    primary_title                genres  \
0  tt0063540                        Sunghursh    Action,Crime,Drama
1  tt0066787  One Day Before the Rainy Season       Biography,Drama
2  tt0069049       The Other Side of the Wind                 Drama
3  tt0069204                  Sabse Bada Sukh          Comedy,Drama
4  tt0100275         The Wandering Soap Opera  Comedy,Drama,Fantasy

   start_year  averagerating  numvotes
0        2013            7.0        77
1        2019            7.2        43
2        2018            6.9      4517
3        2018            6.1        13
4        2017            6.5       119
```

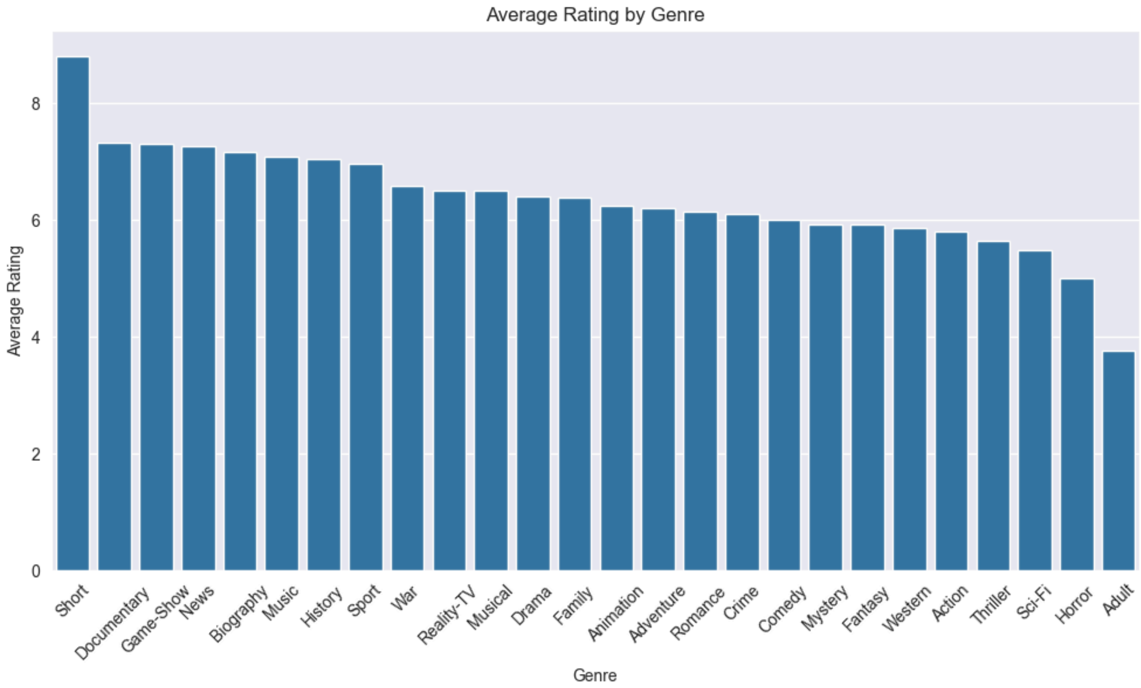
1. **Analyze Data to Determine Top Performing Film Types**

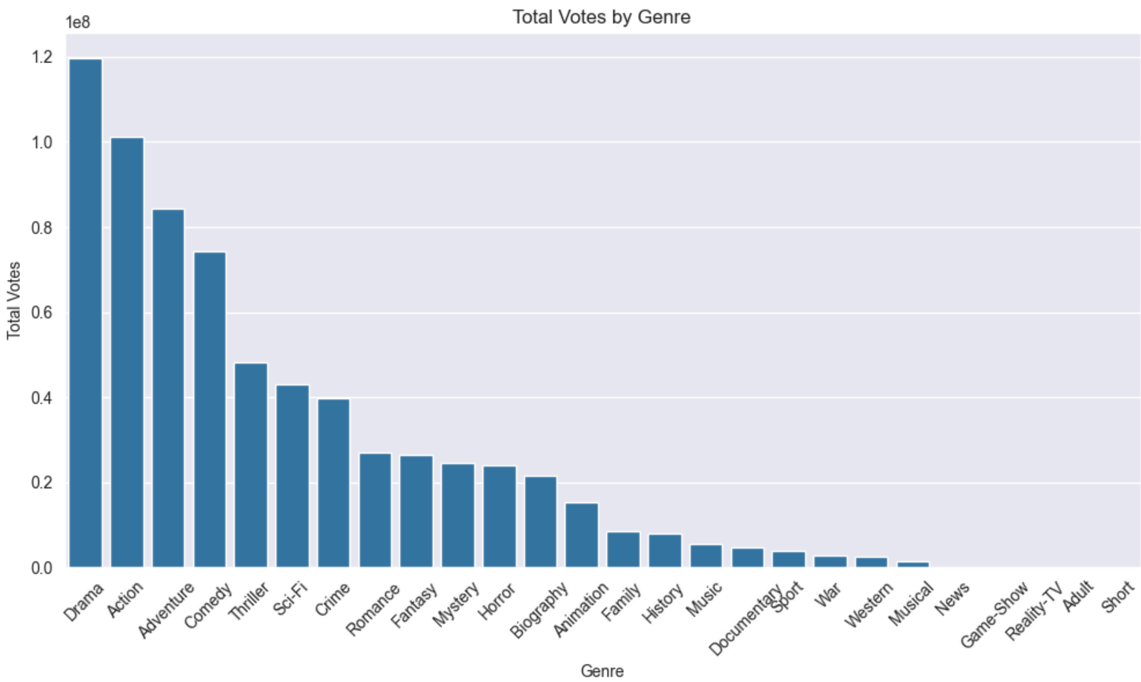
In [3]:
```
# Analyze genres in SQLite data
sqlite_genres = sqlite_df['genres'].str.split(',', expand=True).stack().res
et_index(level=1, drop=True)
sqlite_df_genres = sqlite_df[['primary_title', 'averagerating', 'numvote
s']].join(sqlite_genres.rename('genre'))

# Aggregate data by genre
genre_rating_summary = sqlite_df_genres.groupby('genre').agg({
    'averagerating': 'mean',
    'numvotes': 'sum'
}).reset_index()

print(genre_rating_summary)


# Calculate total gross (domestic + foreign)
csv_df['total_gross'] = csv_df['domestic_gross'] + csv_df['foreign_gross']
# Top 10 movies by total gross
top_movies = csv_df.sort_values(by='total_gross', ascending=False).head(10)
print(top_movies[['title', 'studio', 'total_gross']])
```

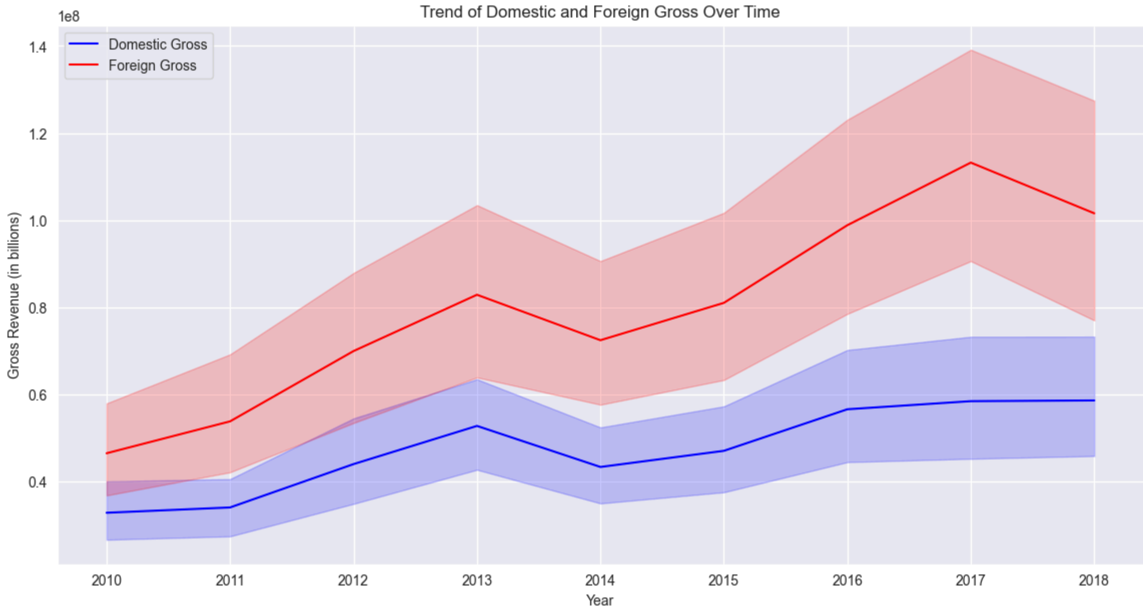
```
          genre  averagerating   numvotes
0        Action       5.810361  101161682
1         Adult       3.766667        164
2     Adventure       6.196201   84232589
3     Animation       6.248308   15353302
4     Biography       7.162274   21609446
5        Comedy       6.002689   74305805
6         Crime       6.115441   39631356
7   Documentary       7.332090    4739345
8         Drama       6.401559  119567500
9        Family       6.394725    8636710
10      Fantasy       5.919473   26335704
11    Game-Show       7.300000       3469
12      History       7.040956    7843349
13       Horror       5.003440   23884695
14        Music       7.091972    5453369
15      Musical       6.498336    1387965
16      Mystery       5.920401   24657286
17         News       7.271330     123319
18   Reality-TV       6.500000        459
19      Romance       6.146608   26913873
20       Sci-Fi       5.489755   42960289
21        Short       8.800000          8
22        Sport       6.961493    3755824
23     Thriller       5.639114   48155313
24          War       6.584291    2684725
25      Western       5.868214    2452376
                                            title studio   total_gross
727                           Marvel's The Avengers     BV  1.518900e+09
1875                        Avengers: Age of Ultron     BV  1.405400e+09
3080                                  Black Panther     BV  1.347000e+09
328   Harry Potter and the Deathly Hallows Part 2     WB  1.341500e+09
2758                       Star Wars: The Last Jedi     BV  1.332600e+09
3081                 Jurassic World: Fallen Kingdom   Uni.  1.309500e+09
1127                                         Frozen     BV  1.276400e+09
2759                       Beauty and the Beast (2017)     BV  1.263500e+09
3082                                   Incredibles 2     BV  1.242800e+09
1128                                     Iron Man 3     BV  1.214800e+09
```

1. **Create Visualizations to Present Findings**

In [4]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Visualization 1: Average Rating by Genre
plt.figure(figsize=(12, 6))
sns.barplot(x='genre', y='averagerating', data=genre_rating_summary.sort_va
lues(by='averagerating', ascending=False))
plt.title('Average Rating by Genre')
plt.xlabel('Genre')
plt.ylabel('Average Rating')
plt.xticks(rotation=45)
plt.show()

# Visualization 2: Total Votes by Genre
plt.figure(figsize=(12, 6))
sns.barplot(x='genre', y='numvotes', data=genre_rating_summary.sort_values
(by='numvotes', ascending=False))
plt.title('Total Votes by Genre')
plt.xlabel('Genre')
plt.ylabel('Total Votes')
plt.xticks(rotation=45)
plt.show()

# Plot domestic and foreign gross over time
plt.figure(figsize=(14, 7))
sns.lineplot(data=csv_df, x='year', y='domestic_gross', label='Domestic Gro
ss', color='blue')
sns.lineplot(data=csv_df, x='year', y='foreign_gross', label='Foreign Gros
s', color='red')
plt.title('Trend of Domestic and Foreign Gross Over Time')
plt.xlabel('Year')
plt.ylabel('Gross Revenue (in billions)')
plt.legend()
plt.grid(True)
plt.show()
```


Average Rating by Genre
Average Rating
0
2
4
6
8
Short
Documentary
Game-Show
News
Biography
Music
History
Sport
War
Reality-TV
Musical
Drama
Family
Animation
Adventure
Romance
Crime
Comedy
Mystery
Fantasy
Western
Action
Thriller
Sci-Fi
Horror
Adult
Genre


Total Votes by Genre
1e8
Total Votes
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Drama
Action
Adventure
Comedy
Thriller
Sci-Fi
Crime
Romance
Fantasy
Mystery
Horror
Biography
Animation
Family
History
Music
Documentary
Sport
War
Western
Musical
News
Game-Show
Reality-TV
Adult
Short
Genre



Conclusion

Summary of Findings

Increasing Foreign and Domestic Gross Over Time: This upward trend suggests a growing market for movies. High-Rated Genres: While genres like Short, Documentaries, Game-Show, and News have high average ratings, they have a relatively low number of votes. This indicates that although these genres are highly rated, their fan base is quite small. Engaging Genres: Drama, Action, Adventure, and Comedy genres receive the highest number of votes. This high vote count indicates that these genres have a large and dedicated fan base.

Recommendations

By focusing on producing movies in popular genres with a large fan base and creating films that appeal to both local and international audiences, our new movie studio has a high chance of success.