

Business Case: Netflix - Data Exploration and Visualisation

About NETFLIX:

Netflix is one of the most popular media and video streaming platforms. They have over 10000 movies or tv shows available on their platform, as of mid-2021, they have over 222M Subscribers globally. This tabular dataset consists of listings of all the movies and tv shows available on Netflix, along with details such as - cast, directors, ratings, release year, duration, etc.

✓ Business Problem:

Analyze the data and generate insights that could help Netflix in deciding which type of shows/movies to produce and how they can grow the business in different countries

```
!pip install pandas matplotlib seaborn wordcloud
```

```
import pandas as pd
# Load the Netflix dataset
netflix_data = pd.read_csv('netflix.csv')
# Display the first few rows of the dataset to understand its structure
netflix_data.head()
```



	show_id	type	title	director	cast	country	date_added	release_year	rat
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-

✓ Observations on the shape of data, data types of all the attributes

```
columns=netflix_data.columns
print(columns)
data_shape=netflix_data.shape
print(data_shape)
data_info=netflix_data.info()
print(data_info)
```

```
➞ Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
        'release_year', 'rating', 'duration', 'listed_in', 'description'],
        dtype='object')
(186325, 12)
<class 'pandas.core.frame.DataFrame'>
Index: 186325 entries, 0 to 8806
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   show_id               186325 non-null object
1   type                  186325 non-null object
2   title                 186325 non-null object
3   director              186325 non-null object
4   cast                  186325 non-null object
5   country               186325 non-null object
6   date_added            186325 non-null object
7   release_year          186325 non-null int64
8   rating                186325 non-null object
9   duration              186325 non-null object
10  listed_in             186325 non-null object
11  description            186325 non-null object
dtypes: int64(1), object(11)
memory usage: 18.5+ MB
None
```

✓ 1. Basic Analysis

1.1 Un-nesting the columns & Handling null values

```

# Un-nesting the 'cast' column
netflix_data['cast'] = netflix_data['cast'].str.split(', ')
netflix_data = netflix_data.explode('cast')

# Un-nesting the 'listed_in' column
netflix_data['listed_in'] = netflix_data['listed_in'].str.split(', ')
netflix_data = netflix_data.explode('listed_in')

# Un-nesting the 'country' column
netflix_data['country'] = netflix_data['country'].str.split(', ')
netflix_data = netflix_data.explode('country')

# Handling null values for categorical columns
categorical_columns = ['type', 'title', 'director', 'cast', 'country', 'rating', 'listed_in']
for col in categorical_columns:
    netflix_data[col].fillna(f'Unknown {col}', inplace=True)

# Handling null values for continuous columns
continuous_columns = ['duration']
for col in continuous_columns:
    netflix_data[col].fillna(0, inplace=True)

# Display the processed dataframe
print("Processed DataFrame:")

# Storing the final processed data in a new DataFrame
final_df = netflix_data.copy()
final_df.head()

```



Processed DataFrame:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown cast	United States	September 25, 2021	2020	PG-13
1	s2	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA
2	s3	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA

✓ 2. Find the counts of each categorical variable both using graphical and non-graphical analysis.

2.1 For Non-graphical Analysis:

```
# Non-graphical analysis: Value counts
type_counts = final_df['type'].value_counts()
country_counts = final_df['country'].value_counts()
rating_counts = final_df['rating'].value_counts()
listed_in_counts = final_df['listed_in'].value_counts()

print(type_counts)
print(country_counts)
print(rating_counts)
print(listed_in_counts)
```

```
⇒ type
Movie      131857
TV Show    54468
Name: count, dtype: int64
country
United States    54219
India            21147
United Kingdom   12404
Unknown country  11145
Japan            7940
...
Uganda           1
Nicaragua        1
Botswana         1
United States,   1
Kazakhstan       1
Name: count, Length: 128, dtype: int64
rating
TV-MA           67628
TV-14           42028
R               23990
PG-13           15233
TV-PG           13778
PG              9011
TV-Y7           5792
TV-Y            3152
TV-G            2650
NR              1521
G               1151
NC-17           149
TV-Y7-FV        86
UR              86
Unknown rating   67
74 min          1
```

```
84 min      1
66 min      1
Name: count, dtype: int64
listed_in
Dramas      27768
International Movies  26129
Comedies     18229
International TV Shows  12324
Action & Adventure  11124
Independent Movies    8815
TV Dramas           8475
Children & Family Movies  7681
Thrillers           6805
Romantic Movies     6159
TV Comedies         4856
Crime TV Shows      4678
Kids' TV            4552
Horror Movies       3965
Sci-Fi & Fantasy     3565
Romantic TV Shows   3012
Music & Musicals     2829
Anime Series        2313
TV Action & Adventure  2280
Spanish-Language TV Shows  2098
```

✓ Insights and Recommendations

Insights:

1. Content Type Distribution:

The majority of the dataset consists of Movies (131,857) and TV Shows (54,468). This suggests that there is more movie content than television.

2. Country Distribution:

The United States leads with 54,219 titles, followed by India with 21,147 titles, and the United Kingdom with 12,404 titles.

3. Rating Distribution:

The most common ratings are TV-MA (67,628), TV-14 (42,028), and R (23,990).

4. Genre Distribution:

The two most popular categories are International Movies (26,129) and Dramas (27,768). This highlights a strong preference for dramatic and internationally diverse content. Other notable genres include Comedies (18,229), International TV Shows (12,324), and Action & Adventure (11,124).

Recommendations:

1.Focus on High-Demand Content:

Considering the high volume of movie content, expanding the movie library, especially in popular genres like Dramas, International Movies, and Comedies, could attract more viewers. For TV shows, increasing the inventory of TV Dramas, International TV Shows, and TV Comedies could be beneficial.

2.Cater to Popular Ratings:

The high number of TV-MA and TV-14 rated content indicates a mature audience. Thus, curating more content with these ratings might help in retaining and growing the viewer base.

3.Diversify Content Offerings:

While Dramas and International Movies are well-represented, exploring and promoting underrepresented genres such as LGBTQ Movies, Classic Movies, and Science & Nature TV could attract niche audiences.

4.Leverage Regional Content:

Given the substantial amount of content from India and the United Kingdom, consider tailoring marketing strategies to these regions. Additionally, promoting regional content to global audiences can enhance viewer engagement and satisfaction.

2.2 For graphical analysis:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Define unknown country labels
unknown_country_labels = ['Unknown country', 'Not Available', 'NA']

# Filter out rows with unknown countries
filtered_df = final_df[~final_df['country'].isin(unknown_country_labels)]

# Create a figure with subplots
fig, axs = plt.subplots(2, 2, figsize=(15, 10))

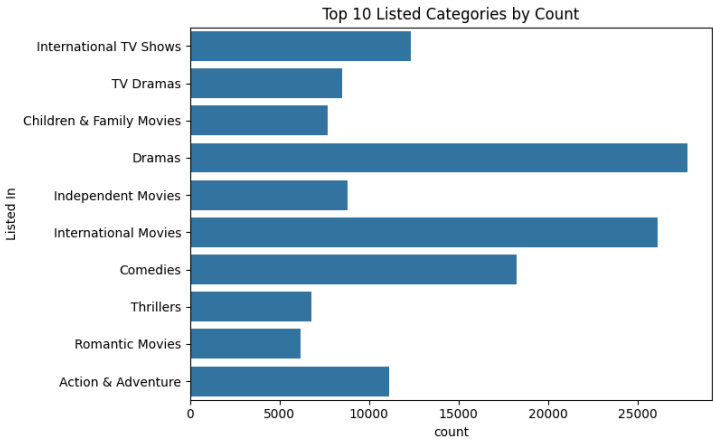
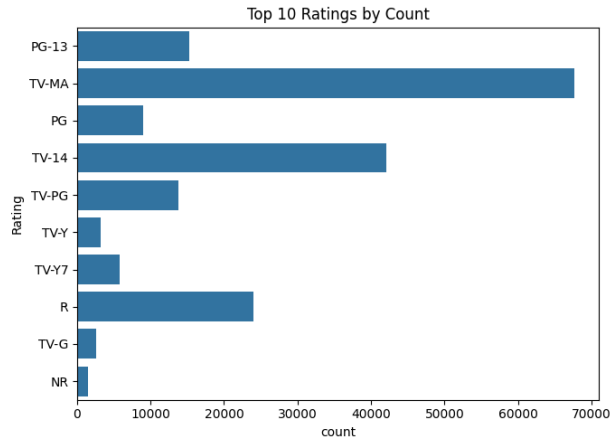
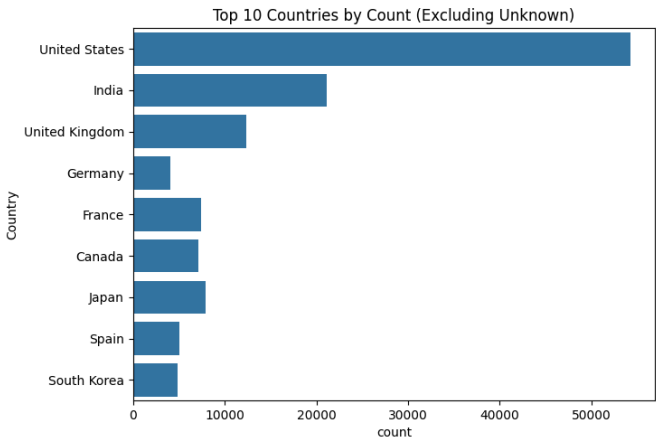
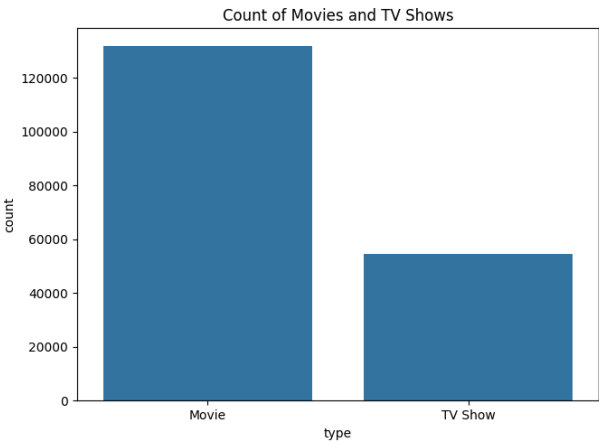
# Count plot for 'type'
sns.countplot(data=final_df, x='type', ax=axs[0, 0])
axs[0, 0].set_title('Count of Movies and TV Shows')

# Count plot for 'country'
top_countries = country_counts.head(10).index
sns.countplot(data=filtered_df[filtered_df['country'].isin(top_countries)], y='country', ax=
axs[0, 1].set_title('Top 10 Countries by Count (Excluding Unknown)')
axs[0, 1].set_ylabel('Country')

# Count plot for 'rating'
top_ratings = rating_counts.head(10).index
sns.countplot(data=final_df[final_df['rating'].isin(top_ratings)], y='rating', ax=axs[1, 0])
axs[1, 0].set_title('Top 10 Ratings by Count')
axs[1, 0].set_ylabel('Rating')

# Count plot for 'listed_in'
top_listed_in = listed_in_counts.head(10).index
sns.countplot(data=final_df[final_df['listed_in'].isin(top_listed_in)], y='listed_in', ax=axs[1, 1])
axs[1, 1].set_title('Top 10 Listed Categories by Count')
axs[1, 1].set_ylabel('Listed In')

# Adjust layout and display
plt.tight_layout()
plt.show()
```



Insights and Recommendations

Insights:

1.Content Type Distribution:

Movies vastly outnumber TV Shows, with approximately 130,000 movies compared to around 55,000 TV shows.

2.Country Distribution:

The United States leads with the highest count of titles, followed by India and the United Kingdom. Other notable countries include Germany, France, Canada, Japan, Spain, and South Korea.

3.Rating Distribution:

TV-MA is the most common rating, followed by TV-14, R, and TV-PG. Ratings such as PG, TV-Y7, TV-Y, TV-G, NR, and G have fewer titles.

4.Genre Distribution:

The most common categories are Dramas, International Movies, and Comedies. Other significant categories include International TV Shows, TV Dramas, Children & Family Movies, Action & Adventure, Independent Movies, Romantic Movies, and Thrillers.

Recommendations:

1.Expand High-Demand Content:

Since movies are more numerous than TV shows, consider expanding the movie library further, especially in popular genres like Dramas, International Movies, and Comedies. For TV shows, increasing content in categories like TV Dramas and International TV Shows can be beneficial.

2.Enhance Regional Focus:

Given the high number of titles from the United States, India, and the United Kingdom, tailoring content and marketing strategies to these regions can attract more viewers. Additionally, promoting content from other countries like Germany, France, and Japan can diversify the viewer base.

3.Optimize Content for Popular Ratings:

Focus on adding more TV-MA and TV-14 rated content, as these are the most common and likely the most popular among viewers. Ensuring a good mix of content with these ratings can help in retaining the audience.

4.Diversify Genre Offerings:

While Dramas and International Movies are well-represented, consider boosting content in underrepresented yet popular categories like Children & Family Movies, Independent Movies, and

Romantic Movies to cater to diverse audience preferences.

✓ 3.Comparison of tv shows vs. movies

3.1 Find the number of movies produced in each country and pick the top 10 countries.

```
# Group by country and count unique movie titles
movies_by_country = final_df[final_df['type'] == 'Movie'].groupby('country')['title'].nunique

# Select top 10 countries by number of unique movie titles
top_10_movies_countries = movies_by_country.sort_values(by='title', ascending=False).head(10)

print("Top 10 Countries by Number of Movies Produced:")
print(top_10_movies_countries)
```

⇒ Top 10 Countries by Number of Movies Produced:

	country	title
114	United States	2751
43	India	962
112	United Kingdom	532
116	Unknown country	440
20	Canada	319
34	France	303
36	Germany	182
100	Spain	171
51	Japan	119
23	China	114

3.2 Find the number of Tv-Shows produced in each country and pick the top 10 countries.

```
# Group by country and count unique TV show titles
tv_shows_by_country = final_df[final_df['type'] == 'TV Show'].groupby('country')['title'].nunique

# Select top 10 countries by number of unique TV show titles
top_10_tv_countries = tv_shows_by_country.sort_values(by='title', ascending=False).head(10)

print("\nTop 10 Countries by Number of TV Shows Produced:")
print(top_10_tv_countries)
```



Top 10 Countries by Number of TV Shows Produced:

	country	title
63	United States	938
64	Unknown country	391
62	United Kingdom	272
30	Japan	199

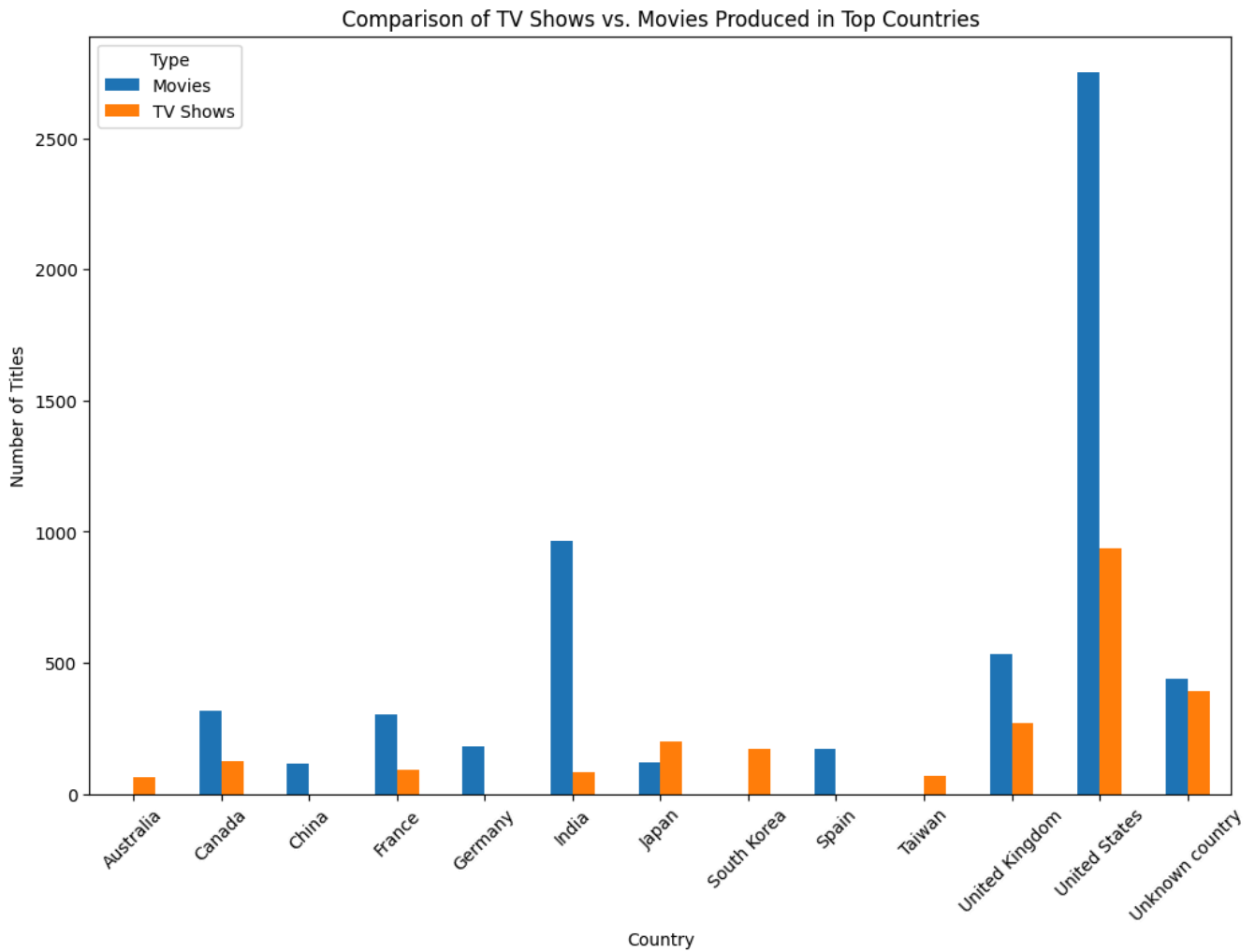
52	South Korea	170
8	Canada	126
19	France	90
25	India	84
57	Taiwan	70
2	Australia	66

```
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate number of movies and TV shows in each top country
top_countries_movies = top_10_movies_countries.set_index('country')['title']
top_countries_tv_shows = top_10_tv_countries.set_index('country')['title']

# Combine into a single DataFrame
comparison_df = pd.DataFrame({
    'Movies': top_countries_movies,
    'TV Shows': top_countries_tv_shows
})

# Plotting
comparison_df.plot(kind='bar', figsize=(12, 8))
plt.title('Comparison of TV Shows vs. Movies Produced in Top Countries')
plt.xlabel('Country')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.legend(title='Type')
plt.show()
```



Insights and Recommendations

Insights:

1. Country-wise Content Distribution:

The United States has the highest number of both movies and TV shows, with movies significantly outnumbering TV shows. India, Canada, and the United Kingdom also have a substantial number of titles, predominantly movies. Other countries like France, Germany, Japan, South Korea, and China have fewer titles, with a noticeable preference for movies over TV shows.

2. Movies vs. TV Shows:

In almost all countries, the number of movies exceeds the number of TV shows. This trend is particularly pronounced in the United States and India. The United Kingdom and Canada also follow this trend, but the gap between the number of movies and TV shows is narrower.

Recommendations:

1. Content Strategy:

Since movies are more prevalent than TV shows across most countries, consider continuing to expand the movie library, especially in high-producing countries like the United States, India, and the United Kingdom. For TV shows, focus on increasing the content in countries where the gap is narrower, such as the United Kingdom and Canada, to balance the library.

2. Regional Focus:

Tailor marketing and promotional strategies to emphasize the strengths of each region. For instance, highlight the vast movie collection from the United States and India to attract movie enthusiasts. Promote TV shows from the United Kingdom and Canada, where the TV show counts are relatively higher, to cater to viewers looking for TV series from these regions.

3. Content Diversification:

Encourage the production and acquisition of more TV shows from countries where they are currently underrepresented, such as Japan, South Korea, and Germany. This can attract viewers interested in international TV series.

✓ 4. What is the best time to launch a TV show?

4.1 Find which is the best week to release the Tv-show or the movie.

```
# Add a new column for the week of the year
final_df['date_added'] = pd.to_datetime(final_df['date_added'], errors='coerce')
final_df['week_added'] = final_df['date_added'].dt.isocalendar().week

# Best week to release
best_week_movies = final_df[final_df['type'] == 'Movie']['week_added'].mode()
best_week_tvshows = final_df[final_df['type'] == 'TV Show']['week_added'].mode()

print("Best week to release movies:", best_week_movies)
print("Best week to release TV shows:", best_week_tvshows)
```

```
➞ Best week to release movies: 0    1
   Name: week_added, dtype: UInt32
   Best week to release TV shows: 0    27
   Name: week_added, dtype: UInt32
```

```
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate the number of releases per week for movies and TV shows
weekly_movie_counts = final_df[final_df['type'] == 'Movie'].groupby('week_added').size()
weekly_tvshow_counts = final_df[final_df['type'] == 'TV Show'].groupby('week_added').size()

# Plotting
plt.figure(figsize=(12, 8))

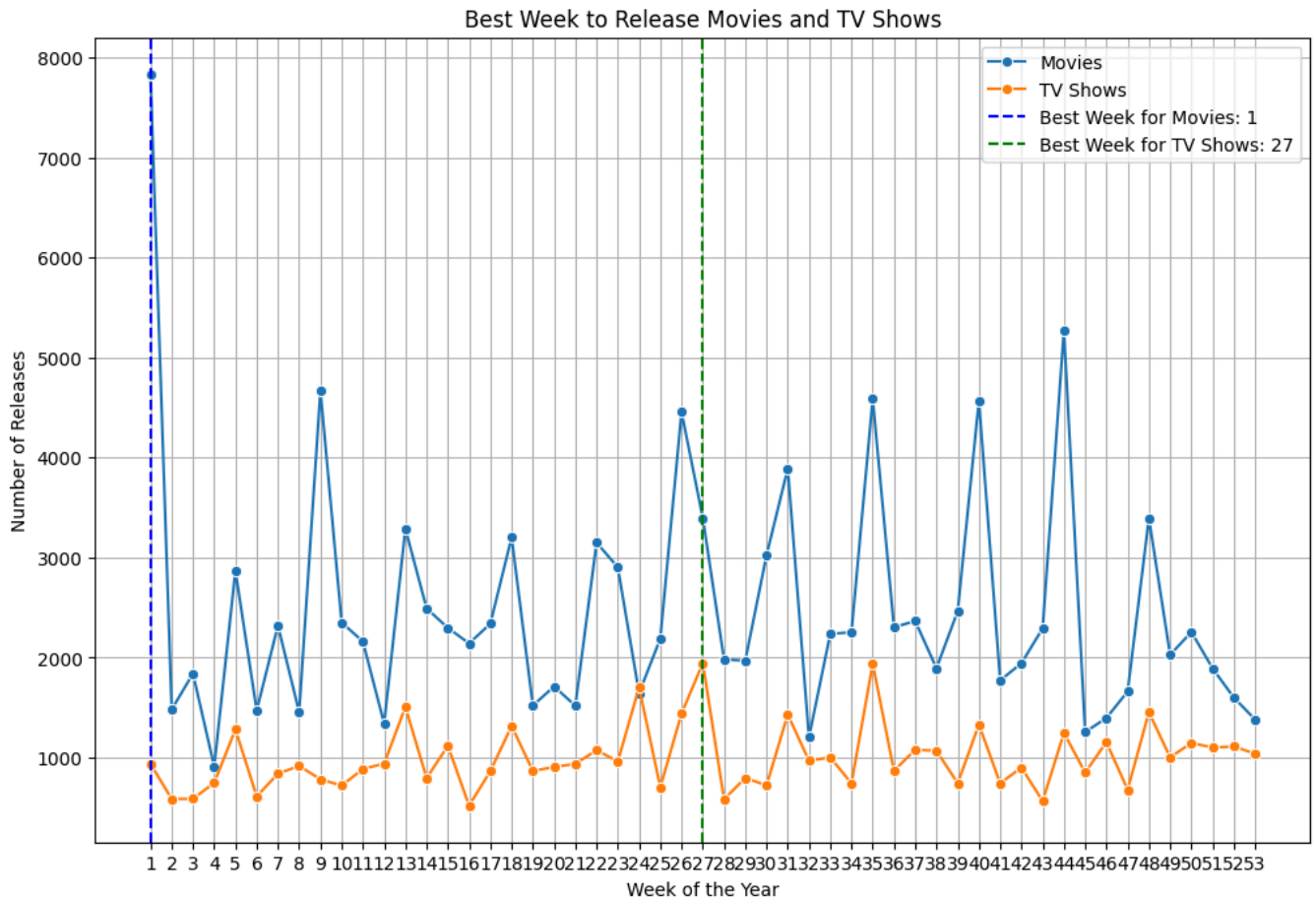
# Line plot for movies
sns.lineplot(x=weekly_movie_counts.index, y=weekly_movie_counts.values, label='Movies', mark

# Line plot for TV shows
sns.lineplot(x=weekly_tvshow_counts.index, y=weekly_tvshow_counts.values, label='TV Shows',

# Highlight the best week to release
best_week_movie = best_week_movies.iloc[0]
best_week_tvshow = best_week_tvshows.iloc[0]

plt.axvline(x=best_week_movie, color='blue', linestyle='--', linewidth=1.5, label=f'Best Wee
plt.axvline(x=best_week_tvshow, color='green', linestyle='--', linewidth=1.5, label=f'Best w

plt.title('Best Week to Release Movies and TV Shows')
plt.xlabel('Week of the Year')
plt.ylabel('Number of Releases')
plt.legend()
plt.xticks(range(1, 54), range(1, 54))
plt.grid(True)
plt.show()
```



✓ Insights and Recommendations

Insights:

1. Movies:

Peak Release Week: The highest number of movie releases occurs in week 1.

Other Notable Peaks: There are significant peaks around weeks 10, 22, 34, and 44.

2. TV Shows:

Peak Release Week: The highest number of TV show releases occurs in week 27.

Other Notable Peaks: There are smaller peaks in weeks 1, 13, and 39.

Recommendations:

1. Movies:

Optimal Time: If you want your movie to be released when there is less competition, avoid releasing it in week 1, as it has the highest number of releases.

Better Alternatives: Consider releasing your movie in weeks with fewer releases to stand out more. Weeks 17, 30, and 48 have relatively lower numbers of movie releases.

2. TV Shows:

Optimal Time: Avoid releasing new TV shows in week 27 due to high competition.

Better Alternatives: Choose weeks with fewer TV show releases. Weeks 5, 14, and 43 have relatively lower numbers of TV show releases.

4.2 Find which is the best month to release the Tv-show or the movie.

```
# Add a new column for the month of the year
final_df['month_added'] = final_df['date_added'].dt.month

# Best month to release
best_month_movies = final_df[final_df['type'] == 'Movie']['month_added'].mode()
best_month_tvshows = final_df[final_df['type'] == 'TV Show']['month_added'].mode()

print("Best month to release movies:", best_month_movies)
print("Best month to release TV shows:", best_month_tvshows)
```

```
➞ Best month to release movies: 0    7.0
   Name: month_added, dtype: float64
   Best month to release TV shows: 0    12.0
   Name: month_added, dtype: float64
```



```
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate the number of releases per month for movies and TV shows
monthly_movie_counts = final_df[final_df['type'] == 'Movie'].groupby('month_added').size()
monthly_tvshow_counts = final_df[final_df['type'] == 'TV Show'].groupby('month_added').size()

# Plotting
plt.figure(figsize=(12, 8))

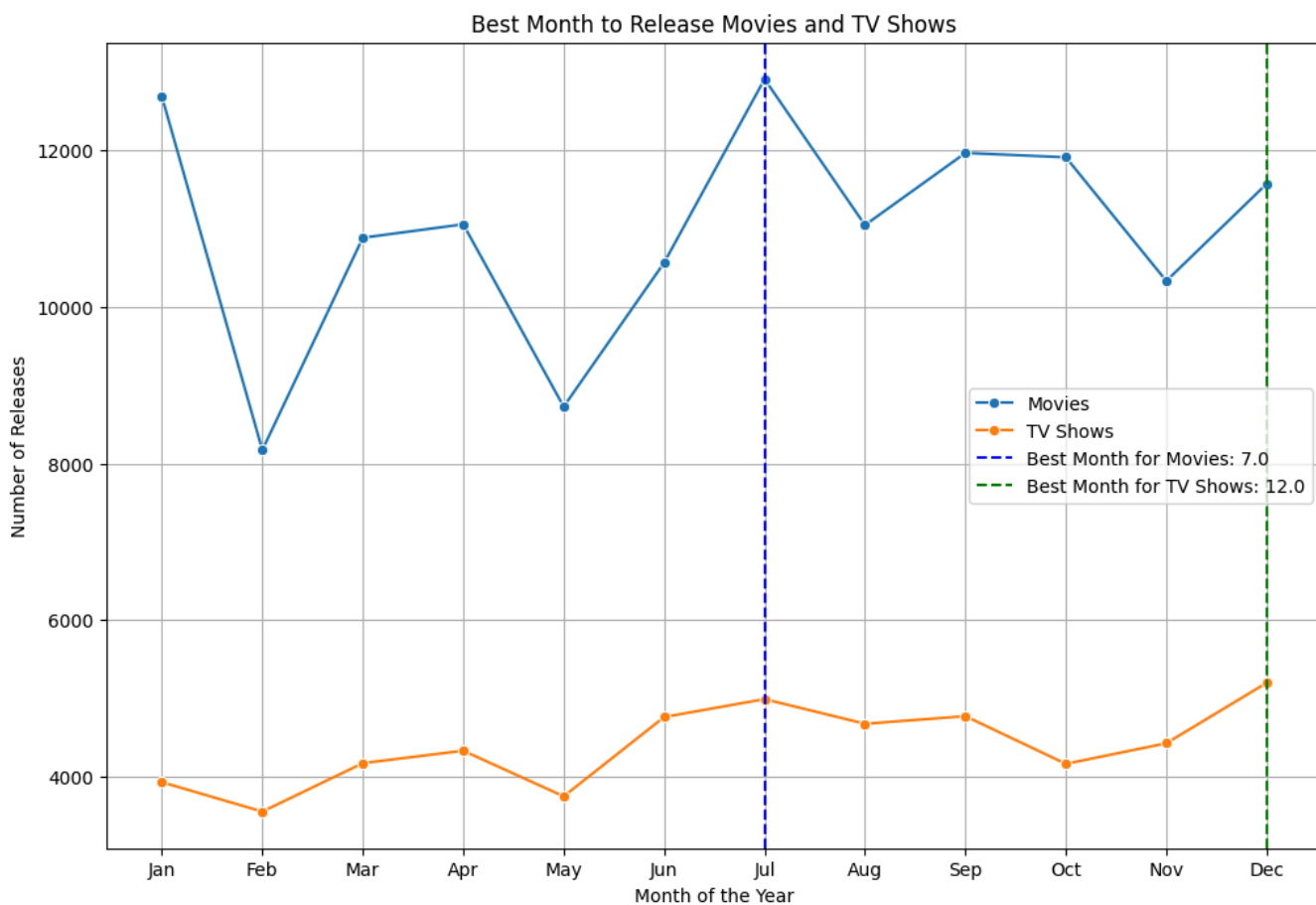
# Line plot for movies
sns.lineplot(x=monthly_movie_counts.index, y=monthly_movie_counts.values, label='Movies', ma

# Line plot for TV shows
sns.lineplot(x=monthly_tvshow_counts.index, y=monthly_tvshow_counts.values, label='TV Shows'

# Highlight the best month to release
best_month_movie = best_month_movies.iloc[0]
best_month_tvshow = best_month_tvshows.iloc[0]

plt.axvline(x=best_month_movie, color='blue', linestyle='--', linewidth=1.5, label=f'Best Mo
plt.axvline(x=best_month_tvshow, color='green', linestyle='--', linewidth=1.5, label=f'Best

plt.title('Best Month to Release Movies and TV Shows')
plt.xlabel('Month of the Year')
plt.ylabel('Number of Releases')
plt.legend()
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oc
plt.grid(True)
plt.show()
```



Insights and Recommendations

Insights:

1. Movies:

The highest number of movie releases occurs in July. The lowest number of movie releases occurs in February. There is a noticeable increase in releases in the middle of the year (around June and July) and a drop towards the beginning of the year (February).

2. TV Shows:

The highest number of TV show releases occurs in December. The lowest number of TV show releases occurs in February. TV show releases are relatively stable throughout the year with slight increases towards the end of the year (November and December).

Recommendations:

1. For Movie Releases:

Best Month to Release: Consider releasing movies in July to take advantage of the high number of releases, which might indicate a popular time for audiences to watch new movies. Avoid February: Since February has the lowest number of releases, it might be less competitive, but also might indicate lower audience engagement.

2. For TV Show Releases:

Best Month to Release: Consider releasing TV shows in December to align with the peak release period. Avoid February: Like movies, February has the lowest number of TV show releases, suggesting it might be a less favorable time for new releases.

5. Analysis of actors/directors of different types of shows/movies.

5.1. Identify the top 10 actors who have appeared in most movies or TV shows

```
# Group by each actor and count unique titles of movies and TV shows
top_actors = final_df.groupby('cast')['title'].nunique().sort_values(ascending=False).head(10)

# Print the top 10 actors
print("Top 10 Actors by Number of Appearances:")
print(top_actors)
```

➡ Top 10 Actors by Number of Appearances:

cast	
Unknown cast	825
Anupam Kher	43
Shah Rukh Khan	35
Julie Tejwani	33
Naseeruddin Shah	32
Takahiro Sakurai	32
Rupa Bhimani	31
Om Puri	30
Akshay Kumar	30
Yuki Kaji	29

Name: title, dtype: int64

```
import seaborn as sns
import matplotlib.pyplot as plt

# Filter out 'Unknown' cast
sub_df = final_df[final_df['cast'] != 'Unknown cast']


# Group by each actor and count unique titles of movies and TV shows
top_actors = sub_df.groupby('cast')['title'].nunique().sort_values(ascending=False).head(10)

# Plotting
plt.figure(figsize=(14, 8))
barplot = sns.barplot(x=top_actors.values, y=top_actors.index, palette='viridis')

# Adding count labels to each bar
for index, value in enumerate(top_actors.values):
    barplot.text(value, index, f'{value}', color='black', ha="left", va="center", fontsize=12)

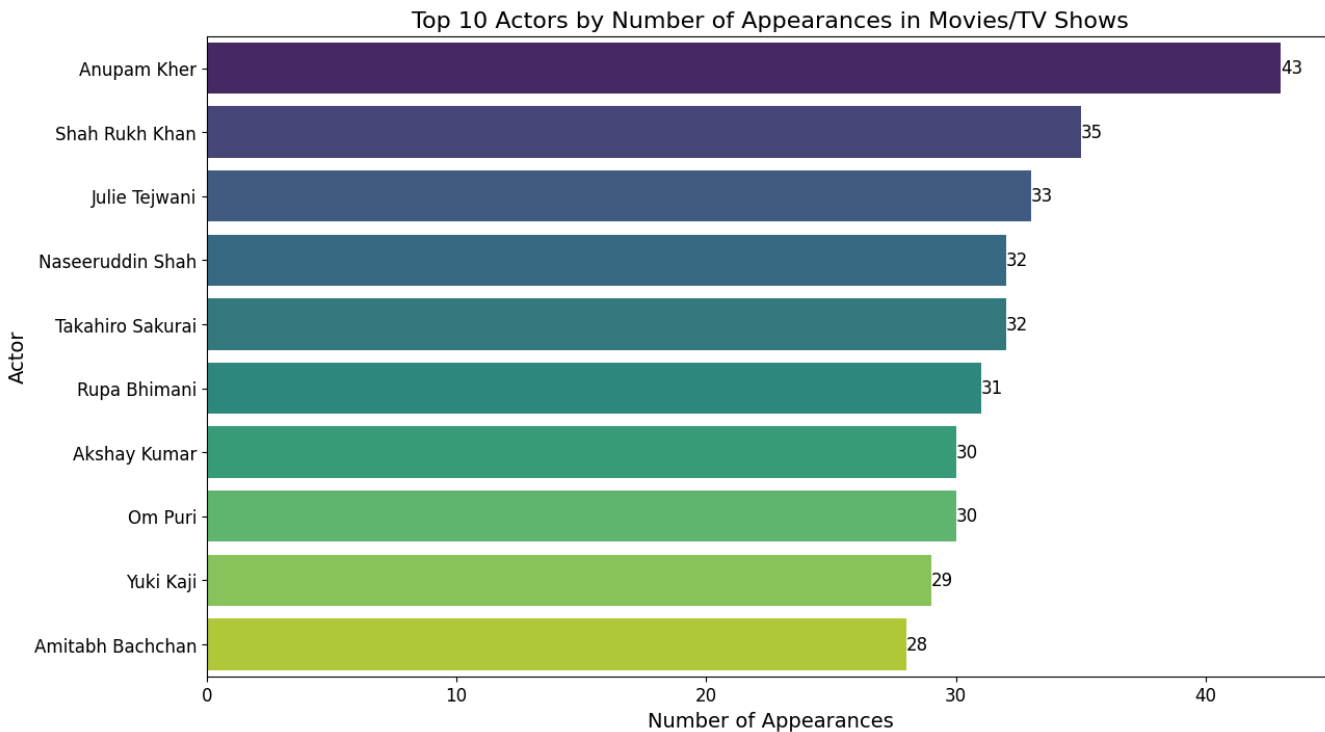
plt.title('Top 10 Actors by Number of Appearances in Movies/TV Shows', fontsize=16)
plt.xlabel('Number of Appearances', fontsize=14)
plt.ylabel('Actor', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

plt.show()
```

 <ipython-input-14-e01ad5ade632>:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

```
barplot = sns.barplot(x=top_actors.values, y=top_actors.index, palette='viridis')
```



✓ Insights and Recommendations

Insights:

1. Anupam Kher Leads:

Anupam Kher has the highest number of appearances in movies/TV shows among the listed actors, with 43 appearances.

2.Popular Actors:

Shah Rukh Khan follows with 35 appearances. Julie Tejjwani is in third place with 33 appearances.

3.Top 10 Actors:

3.1.Anupam Kher is the top actor with the most appearances, totaling 43.

3.2.Shah Rukh Khan comes second with 35 appearances.

3.3.Julie Tejjwani is third with 33 appearances.

3.4.Both Naseeruddin Shah and Takahiro Sakurai are tied with 32 appearances each.

3.5.Rupa Bhimani is next with 31 appearances.

3.6.Akshay Kumar and Om Puri both have 30 appearances each.

3.7.Yuki Kaji has made 29 appearances.

3.8.Amitabh Bachchan rounds out the top 10 with 28 appearances.

Recommendations:

1.Leverage Star Power for Marketing Campaigns:

1.1 High-Profile Collaborations: Utilize the popularity of top actors like Anupam Kher, Shah Rukh Khan, and Amitabh Bachchan in marketing and promotional campaigns to attract a larger audience.

1.2 Endorsements and Sponsorships: Secure endorsements and sponsorships from these top actors to enhance brand visibility and credibility.

2.Content Strategy:

2.1 Diverse Casting: Diversify casting to include actors with a high number of appearances in various genres to appeal to a broader audience.

2.2 International Reach: Consider including international actors like Takahiro Sakurai and Yuki Kaji to attract global audiences and expand market reach.

3.Production Investments:

3.1 Quality Over Quantity: Invest in high-quality productions featuring these top actors to ensure high viewer engagement and positive reviews.

3.2 Frequent Releases: Aim to produce and release content regularly featuring these actors to maintain audience interest and loyalty.

4. Audience Engagement:

4.1 Fan Interaction: Organize events, social media interactions, and live sessions with these top actors to engage with their fan base directly.

4.2 Exclusive Content: Offer exclusive behind-the-scenes content, interviews, and special features with these actors to provide added value to the audience.

5.2 Identify the top 10 directors who have appeared in most movies or TV shows.

```
# Group by each director and count unique titles of movies and TV shows
top_directors = final_df.groupby('director')['title'].nunique().sort_values(ascending=False)

# Print the top 10 directors
print("\nTop 10 Directors by Number of Titles Directed:")
print(top_directors)
```



Top 10 Directors by Number of Titles Directed:

director	
Unknown director	2634
Rajiv Chilaka	19
Raúl Campos, Jan Suter	18
Marcus Raboy	16
Suhas Kadav	16
Jay Karas	14
Cathy Garcia-Molina	13
Martin Scorsese	12
Jay Chapman	12
Youssef Chahine	12

Name: title, dtype: int64

```
import seaborn as sns
import matplotlib.pyplot as plt

# Filter out 'Unknown' directors
sub_df = final_df[final_df['director'] != 'Unknown director']


# Group by each director and count unique titles of movies and TV shows
top_directors = sub_df.groupby('director')['title'].nunique().sort_values(ascending=False).r

# Plotting
plt.figure(figsize=(14, 8))
barplot = sns.barplot(x=top_directors.values, y=top_directors.index, palette='viridis')

# Adding count labels to each bar
for index, value in enumerate(top_directors.values):
    barplot.text(value, index, f'{value}', color='black', ha="left", va="center", fontsize=16)

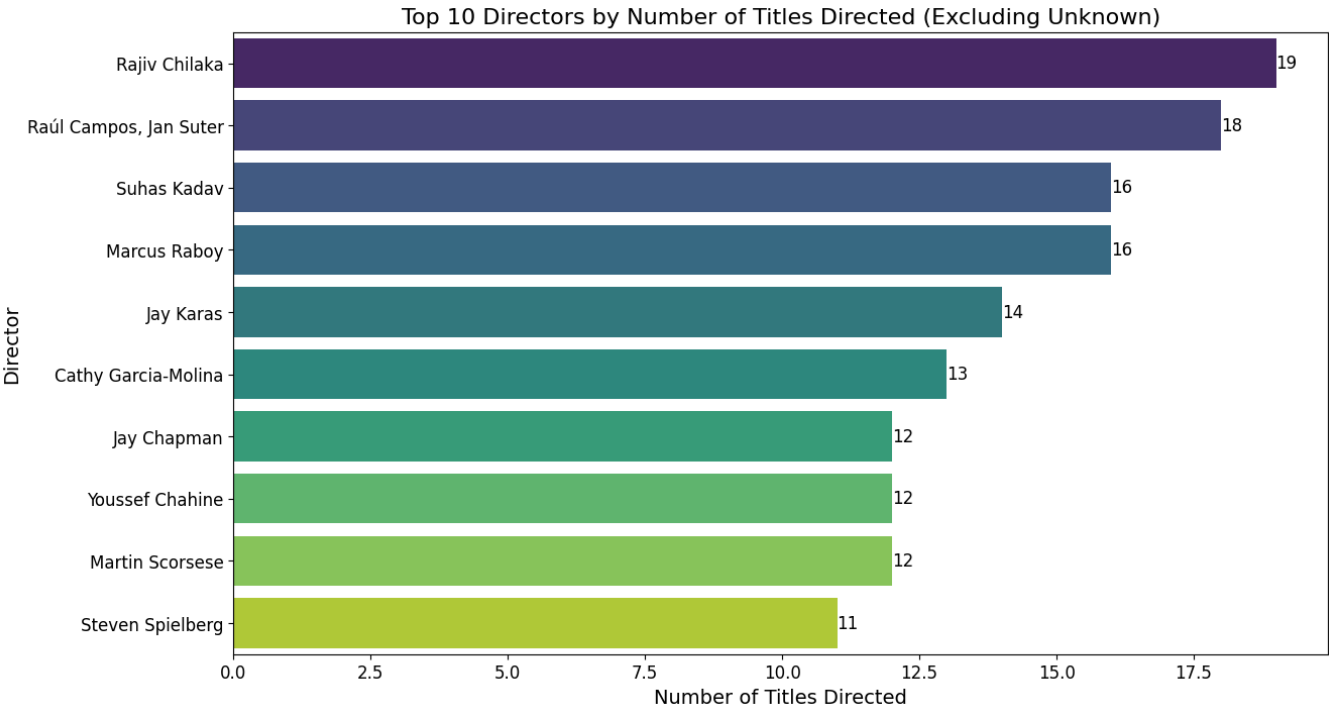
plt.title('Top 10 Directors by Number of Titles Directed (Excluding Unknown)', fontsize=16)
plt.xlabel('Number of Titles Directed', fontsize=14)
plt.ylabel('Director', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

plt.show()
```


 <ipython-input-16-de96cf7b8f8f>:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

```
barplot = sns.barplot(x=top_directors.values, y=top_directors.index, palette='viridis')
```



Insights and Recommendations

Insights:

1. Director Popularity and Volume: The chart highlights the top 10 directors by the number of titles directed, indicating their popularity and productivity in the industry. Rajiv Chilaka leads with 19 titles, followed closely by Raúl Campos and Jan Suter with 18 titles each.

2. Diverse Geographical Representation: The directors come from various regions, suggesting a global diversity in content creation. This includes directors like Rajiv Chilaka (India), Youssef Chahine (Egypt), and Martin Scorsese and Steven Spielberg from the United States.

3. Established vs. Emerging Directors: Some directors like Martin Scorsese and Steven Spielberg are well-established and globally recognized, while others might be emerging talents in specific regions or genres.

4. Genre Specialization: Directors with a high number of titles might specialize in genres with high production rates, such as animation (Rajiv Chilaka) or TV series (Raúl Campos, Jan Suter).

Recommendations:

1. Strategic Collaborations: Partner with top directors like Rajiv Chilaka, Raúl Campos, and Jan Suter to leverage their expertise and audience base. This can help in producing content that has a proven track record of success.

2. Invest in Emerging Talent: Invest in directors who have shown high productivity and are on the rise, such as Suhas Kadav and Jay Karas. This can help in creating innovative content and tapping into new market segments.

3. Global Market Penetration: Utilize the geographical diversity of directors to penetrate new international markets. Collaborating with directors like Youssef Chahine can help in creating content that resonates with Middle Eastern audiences, for instance.

4. Marketing and Promotion: Use the popularity of well-known directors like Steven Spielberg and Martin Scorsese to boost marketing efforts. Their names can draw significant attention and viewership.

5. Content Diversification: Ensure a diverse range of content by working with directors from different regions and backgrounds. This can attract a wider audience and cater to varied tastes and preferences.

6. Content Quality and Innovation: Focus on maintaining high production values and innovative storytelling by learning from the successful projects of these top directors. This can help in setting a benchmark for quality in your productions.

✓ 6. Which genre movies are more popular or produced more

```

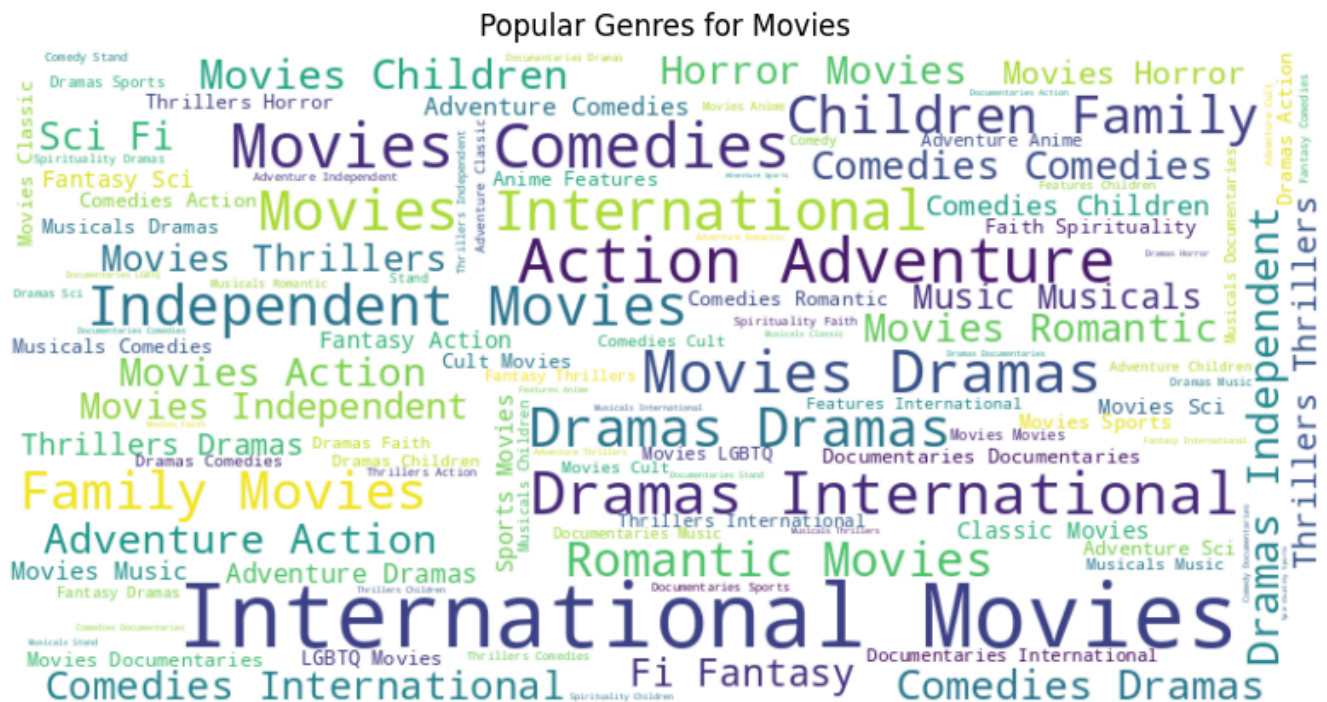
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Filter for movies
movies_df = final_df[final_df['type'] == 'Movie']

# Generate word cloud for genres
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(' '.join(mov

# Plotting
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Popular Genres for Movies')
plt.show()

```



Insights and Recommendations

Insights:

1. Genre Popularity:

International Movies are highly prominent, indicating a significant interest in films from diverse cultures and countries. Comedies and Dramas are also very popular, showing a strong audience

preference for these genres. Family Movies and Children's Movies have substantial visibility, suggesting a steady demand for family-friendly content.

2.Action and Adventure:

Action and Adventure Movies are frequently mentioned, reflecting a consistent demand for high-energy, exciting content.

3.Niche Genres:

Genres like Sci-Fi, Fantasy, Horror, Musicals, and Documentaries have noticeable mentions, indicating niche markets that can be explored for specific audience segments. LGBTQ Movies and Faith & Spirituality films appear, pointing to growing interest in diverse and inclusive content.

4.Independent and International Focus:

Independent Movies have a significant presence, suggesting an audience interested in unique, non-mainstream films. International Movies are a major trend, emphasizing the global nature of contemporary movie audiences.

Recommendations:

1.Expand International Content:

Invest in acquiring and producing International Movies to cater to the growing global audience. Consider collaborations with filmmakers from different countries to diversify the content library.

2.Focus on Popular Genres:

Prioritize production and acquisition of Comedies and Dramas, as these genres have a broad and stable audience base. Develop a strong lineup of Family Movies and Children's Movies to attract family audiences.

3.Explore Niche Markets:

Invest in niche genres like Sci-Fi, Fantasy, and Horror to capture dedicated fan bases. Creating high-quality content in these genres can attract loyal viewers. Develop content for LGBTQ Movies and Faith & Spirituality to cater to these growing and specific audience groups.

4.Leverage Independent Films:

Support and promote Independent Movies to appeal to audiences looking for unique and original content. Consider film festivals, indie film partnerships, and showcasing emerging talent.

5.Action and Adventure Appeal:

Continue to produce and promote Action and Adventure Movies to maintain and grow the audience base that enjoys high-energy, thrilling content.

6.Cross-Genre Opportunities:

Experiment with cross-genre content (e.g., Sci-Fi Comedies, Romantic Thrillers) to create fresh and innovative films that can attract a diverse audience.

7. Family-Friendly Platform:

Develop a family-friendly platform or section within your streaming service dedicated to Family Movies and Children's Movies, ensuring safe and enjoyable content for all ages.

7. Find After how many days the movie will be added to Netflix after the release of the movie

```
import pandas as pd
# Convert 'release_year' to datetime
final_df['release_year'] = pd.to_datetime(final_df['release_year'], format='%Y')

# Convert 'date_added' to datetime, handling mixed formats
final_df['date_added'] = pd.to_datetime(final_df['date_added'], format='%B %d, %Y', errors='')

# Calculate the days between release and added to Netflix
final_df['days_to_add'] = (final_df['date_added'] - final_df['release_year']).dt.days

# Calculate median of the days to add
days_to_add_median = final_df['days_to_add'].median()

print("Median of days to add on Netflix:", days_to_add_median)
```

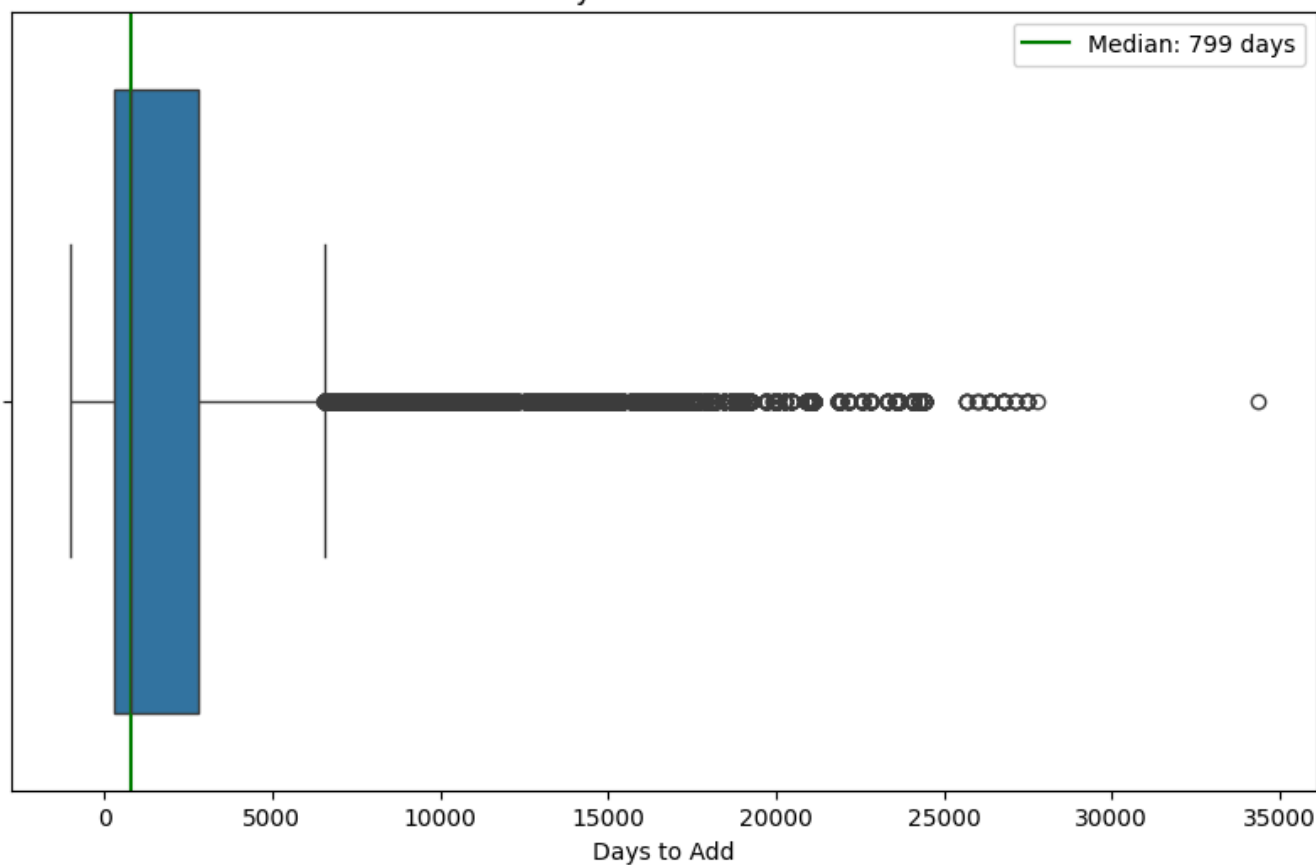
➡ Median of days to add on Netflix: 799.0

```
import matplotlib.pyplot as plt
import seaborn as sns

# Plotting the distribution using a box plot
plt.figure(figsize=(10, 6))
sns.boxplot(x=final_df['days_to_add'].dropna())
plt.axvline(days_to_add_median, color='g', linestyle='--', label=f'Median: {days_to_add_median}')
plt.title('Box Plot of Days to Add Movies to Netflix')
plt.xlabel('Days to Add')
plt.legend()
plt.show()
```



Box Plot of Days to Add Movies to Netflix



✓ Insights and Recommendations

Insights:

1. Median Days to Add:

The median number of days to add movies to Netflix is 799 days, as indicated by the green line.

2. Distribution:

The majority of movies are added within a relatively short period, but there is a significant number of outliers where the addition time extends up to 35,000 days.

3. Outliers:

The presence of extreme outliers suggests that there are some movies that take an exceptionally long time to be added to the platform.

Recommendations:

1.Streamline Acquisition Processes:

Analyze the processes for acquiring movies that fall into the longer tail of the distribution. Identifying and addressing the bottlenecks in these cases can significantly reduce the average time to add new content.

2.Focus on Reducing Variability:

Implement standardized procedures for movie acquisition to reduce the variability in addition times. This could involve negotiating faster licensing deals or improving internal review processes.

3.Prioritize Popular Content:

Prioritize the acquisition of popular or trending movies, especially those that can be added quickly, to keep the platform's content fresh and appealing to subscribers.

4.Enhanced Supplier Relationships:

Strengthen relationships with content suppliers and negotiate better terms to expedite the addition of movies. This could include exclusivity deals or priority access to new releases.