netflix

April 16, 2025

#NETFLIX DATA ANALYSIS

Analyst: ASHWINI POTNURU

Email: potnoruashwini@gmail.com

LinkedIn: www.linkedin.com/in/ashwini-potnuru

1 TABLE CONTENT

##ABOUT NETFLIX

Netflix is an American subscription video on-demand over-the-top streaming service. The service primarily distributes original and acquired films and television shows from various genres, and it is available internationally in multiple languages.Launched in 2007, nearly a decade after Netflix, Inc. began its pioneering DVD-by-mail movie rental service, Netflix is the most-subscribed video on demand streaming media service, with 301.6 million paid memberships in more than 190 countries as of 2025

2 PROBLEM STATEMENT

The primary object of the data is to Analyze 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

#**OBJECTIVE** The objective of this project is to analyze Netflix's content library to uncover insights about genre distribution, release trends, and country-wise content availability. It aims to identify patterns in content addition over time and highlight top-performing genres or directors. The analysis also explores relationships between ratings, content type, and release year. This project helps understand user content preferences and guides data-driven recommendations.

3 DATASET OVERVIEW

The dataset provided to you consists of a list of all the TV shows/movies available on Netflix: * Show_id: Unique ID for every Movie / Tv Show * Type: Identifier - A Movie or TV Show * Title: Title of the Movie / Tv Show * Director: Director of the Movie * Cast: Actors involved in the movie/show * Country: Country where the movie/show was produced * Date_added: Date it was added on Netflix * Release_year: Actual Release year of the movie/show * Rating: TV Rating of the movie/show * Duration: Total Duration - in minutes or number of seasons * Listed_in: Genre * Description: The summary description

4 IMPORTING LIBRARIES

[49]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
```

[50]:
```
pip install wordcloud matplotlib
```

```
Requirement already satisfied: wordcloud in /usr/local/lib/python3.11/dist-
packages (1.9.4)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-
packages (3.10.0)
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.11/dist-
packages (from wordcloud) (2.0.2)
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages
(from wordcloud) (11.1.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-
packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in
```

```
/usr/local/lib/python3.11/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-
packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
```

5 LOADING THE DATA SET

[51]:
```
Netflix=pd.read_csv('netflix.csv')
```

6 BASIC ANALYSIS OF DATA

[52]:
```
Netflix.head()
```

[52]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm…
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t…
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act…	To protect his family from a powerful drug lor…
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo…
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV …	In a city of coaching centers known to train I…

[53]:
```
Netflix.tail()
```

[53]:
```
     show_id     type        title         director  \
8802   s8803    Movie       Zodiac    David Fincher
8803   s8804  TV Show  Zombie Dumb              NaN
8804   s8805    Movie   Zombieland  Ruben Fleischer
8805   s8806    Movie         Zoom     Peter Hewitt
8806   s8807    Movie       Zubaan      Mozez Singh

                                                cast        country  \
8802  Mark Ruffalo, Jake Gyllenhaal, Robert Downey J…  United States
8803                                               NaN            NaN
8804  Jesse Eisenberg, Woody Harrelson, Emma Stone, …  United States
8805  Tim Allen, Courteney Cox, Chevy Chase, Kate Ma…  United States
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan…          India

            date_added  release_year rating   duration  \
8802  November 20, 2019          2007      R    158 min
8803       July 1, 2019          2018  TV-Y7  2 Seasons
8804   November 1, 2019          2009      R     88 min
8805   January 11, 2020          2006     PG     88 min
8806      March 2, 2019          2015  TV-14    111 min

                                      listed_in  \
8802                 Cult Movies, Dramas, Thrillers
8803         Kids' TV, Korean TV Shows, TV Comedies
8804                        Comedies, Horror Movies
8805             Children & Family Movies, Comedies
8806  Dramas, International Movies, Music & Musicals

                                         description
8802  A political cartoonist, a crime reporter and a…
8803  While living alone in a spooky town, a young g…
8804  Looking to survive in a world taken over by zo…
8805  Dragged from civilian life, a former superhero…
8806  A scrappy but poor boy worms his way into a ty…
```

[54]:
```
Netflix.shape
```

[54]:
```
(8807, 12)
```

[55]:
```
Netflix.columns
```

[55]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

[56]:
```
Netflix.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64 
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

[57]:
```
Netflix.describe()
```

[57]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

INSIGHTS

- Data set consists of 8807 rows and 12 columns
- The movie releasing years is in a range from minimum 1925 to maximum 2021

7 DATA CLEANING

- Check for duplicate records
- check for null values or errors
- Replace null values with appropriate values
- Remove records that are irrelevant to the analysis

[58]:
```
Netflix[Netflix.duplicated()]
```

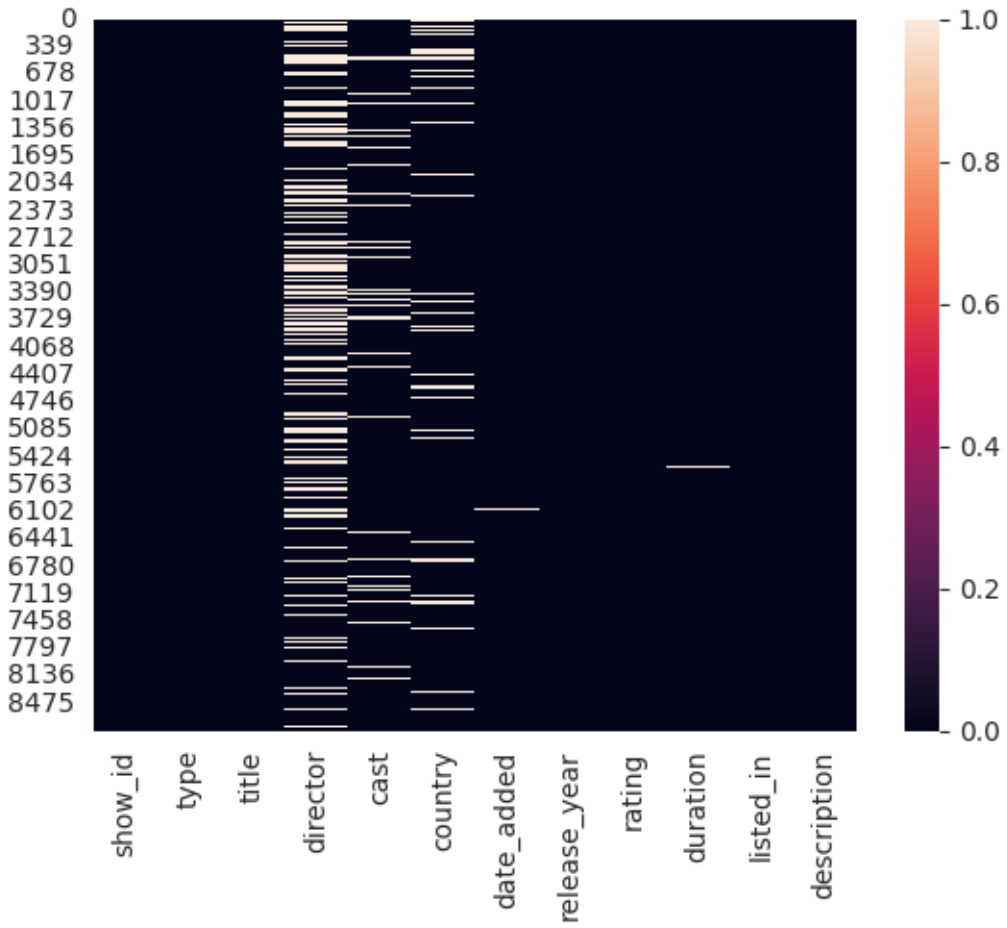
[58]:
```
Empty DataFrame
Columns: [show_id, type, title, director, cast, country, date_added,
release_year, rating, duration, listed_in, description]
Index: []
```

[59]:
```
Netflix.isnull().sum()
```

[59]:
```
show_id           0
type              0
title             0
director       2634
cast            825
country         831
date_added       10
release_year      0
rating            4
duration          3
listed_in         0
description       0
dtype: int64
```

[60]:
```
sns.heatmap(Netflix.isnull()) # showing null values
```

[60]:
```
<Axes: >
```



[61]:
```
Netflix['date_added']=pd.to_datetime(Netflix['date_added'],errors='coerce')
```

[62]:
```
Netflix['date_added'].fillna(Netflix['date_added'].mode()[0],inplace=True)
```

```
<ipython-input-62-372e60abc36b>:1: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace
method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  Netflix['date_added'].fillna(Netflix['date_added'].mode()[0],inplace=True)
```

[63]:
```
# Extracted month,month_name,day  information from the 'date_added' column
Netflix['month'] = Netflix['date_added'].dt.month
Netflix['month_name'] = Netflix['date_added'].dt.month_name()
Netflix['day'] = Netflix['date_added'].dt.day_name()
Netflix['month_name'].value_counts()
```

[63]:
```
month_name
January      825
July         819
December     797
September    765
April        759
October      755
August       749
March        734
June         724
November     697
May          626
February     557
Name: count, dtype: int64
```

[64]:
```
Netflix['country'].fillna(Netflix['country'].mode()[0],inplace=True)
```

```
<ipython-input-64-15d022924408>:1: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace
method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  Netflix['country'].fillna(Netflix['country'].mode()[0],inplace=True)
```

[65]:
```
Netflix['duration'].fillna('120mins',inplace=True)
```

```
<ipython-input-65-93f320e2c6cc>:1: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace
method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
```

```
instead, to perform the operation inplace on the original object.


  Netflix['duration'].fillna('120mins',inplace=True)
```

[66]:
```
Netflix[['director','cast']]=Netflix[['director','cast']].fillna('Unknown')
```

[67]:
```
# covert to category type
Netflix['type']=Netflix['type'].astype('category')
Netflix['rating']=Netflix['rating'].astype('category')
Netflix['cast']=Netflix['cast'].astype('object')
Netflix['listed_in']=Netflix['listed_in'].astype('object')
```

[68]:
```
if 'Unknown' not in Netflix['rating'].cat.categories:
  Netflix['rating']=Netflix['rating'].cat.add_categories(['Unknown']) # it adds␣
 ↪unknown as a valid category
Netflix['rating'].fillna('Unknown',inplace=True)
```

```
<ipython-input-68-0beeaae2e962>:3: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace
method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  Netflix['rating'].fillna('Unknown',inplace=True)
```

[69]:
```
Netflix.isna().sum()
```

[69]:
```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
month           0
```

```
month_name      0
day             0
dtype: int64
```

[70]:
```
#Netflix.to_csv('cleaned_data.csv',index=False)
```

[71]:
```
# from google.colab import files
# files.download('cleaned_data.csv')
```

8 NON GRAPHICAL ANALYSIS

[72]:
```
# Show titles of all movies that were released in India only

Netflix[(Netflix['type']=='Movie') & (Netflix['country']=='India')]['title']
```

[72]:
```
24                      Jeans
105          Angamaly Diaries
114                   Anjaam
116                   Dhanak
118                  Gurgaon
                ...
8772    Yamla Pagla Deewana 2
8773      Yanda Kartavya Aahe
8798                Zed Plus
8799                   Zenda
8806                  Zubaan
Name: title, Length: 893, dtype: object
```

[73]:
```
# How many unique tv show and movies

Movies=Netflix[Netflix['type']=='Movie']['title'].nunique()
TV_Shows=Netflix[Netflix['type']=='TV Show']['title'].nunique()
print(f'Movies:{Movies}')
print(f'TV_Shows:{TV_Shows}')
```

```
Movies:6131
TV_Shows:2676
```

INSIGHTS : There are 6131 unique movies and 2676 TV Shows on Netflix

RECOMMENDATION : Consider producing more TV Shows inorder to balance the content as movies currently dominate

[74]:
```
# who are the top 5 directors ?
Top_directors=Netflix['director'].value_counts()
Top_directors.head()
```

[74]:
```
director
Unknown                   2634
Rajiv Chilaka               19
Raúl Campos, Jan Suter      18
Suhas Kadav                 16
Marcus Raboy                16
Name: count, dtype: int64
```

[75]:
```
Top_Director=Netflix[Netflix['director']!='Unknown']['director'].value_counts()
Top_Director
```

[75]:
```
director
Rajiv Chilaka               19
Raúl Campos, Jan Suter      18
Suhas Kadav                 16
Marcus Raboy                16
Jay Karas                   14
                            ..
James Brown                  1
Ivona Juka                   1
Mu Chu                       1
Chandra Prakash Dwivedi      1
Majid Al Ansari              1
Name: count, Length: 4528, dtype: int64
```

INSIGHTS :

- Rajiv Chilaka is the most famous director with 19 movies followed by Raul Campos and Jan Suter with 18 movies
- Less directed movies director is Majid AI Ansari with only a single movie

RECOMMENDATIONS :

- Consider deeper collaborations with top directors like Rajiv Chilaka who consitently produce content
- Explore promoting content from the well known directors to reach universally

[76]:
```
# Which year has the highest number of releases?

Netflix['release_year'].value_counts().nlargest(1)
```

[76]:
```
release_year
2018    1147
Name: count, dtype: int64
```

[77]:
```
# How many movies and TV shows were released in each decade ?

Netflix['decade']=(Netflix['release_year']//10)*10
decade_count=Netflix.groupby(['decade','type'])['type'].count()
```

```
print(decade_count)
```

```
decade  type
1920    Movie         0
        TV Show       1
1940    Movie        13
        TV Show       2
1950    Movie        11
        TV Show       0
1960    Movie        23
        TV Show       2
1970    Movie        66
        TV Show       4
1980    Movie       122
        TV Show       7
1990    Movie       241
        TV Show      33
2000    Movie       677
        TV Show     133
2010    Movie      4184
        TV Show    1743
2020    Movie       794
        TV Show     751
Name: type, dtype: int64
```

```
<ipython-input-77-d776aade07bc>:4: FutureWarning: The default of observed=False
is deprecated and will be changed to True in a future version of pandas. Pass
observed=False to retain current behavior or observed=True to adopt the future
default and silence this warning.
  decade_count=Netflix.groupby(['decade','type'])['type'].count()
```

INSIGHTS :

- Content production grew dramatically from 1920s(1 title) to 2010s(5927 titles)
- The 2010s dominate with 4184 movies and 1743 TV Shows

RECOMMENDATION :

- Notice which decades has a boom in movies or tv shows
- If users show interest in older content, Netflix might restore more relevant content

[78]:
```
# Lucky day for the movie releases
Netflix['day_added']=Netflix['date_added'].dt.day_name()
Netflix['day_added'].value_counts()
```

[78]:
```
day_added
Friday       2476
Thursday     1387
Wednesday    1374
Tuesday      1182
```

```
Monday        845
Saturday      803
Sunday        740
Name: count, dtype: int64
```

INSIGHTS :

- Friday is the most popular day for the new releases
- weekend days(saturday and sunday) has a fewest releases

RECOMMENDATIONS :

- Maintain strong Friday release schedule but consider testing thursday release for weekend anticipation
- Experiment with surprise weekend drops for certain content types
- consider weekend picks sections for content added on Fridays

[79]:
```
# Understanding what content is available in different countries

Content=Netflix.groupby('country')[['listed_in','country']].value_counts().
 ↪sort_values(ascending=False)
Content.head(15)
```

[79]:
```
country        listed_in
United States  Documentaries                                265
               Stand-Up Comedy                              240
               Children & Family Movies                     150
               Kids' TV                                     121
India          Comedies, Dramas, International Movies       120
               Dramas, International Movies                 118
United States  Children & Family Movies, Comedies           110
India          Dramas, Independent Movies, International Movies  108
United States  Dramas                                       90
               Comedies                                      88
               Reality TV                                    85
               Docuseries                                    77
Japan          Anime Series, International TV Shows          75
United States  Dramas, Independent Movies                   74
               Documentaries, Music & Musicals               67
Name: count, dtype: int64
```

INSIGHTS :

- Netflix's catalog heavily favors U.S. and Indian content
- United States leads in the content production especially in Documentaries(265) and stand-Up Comedy(240)
- India is the second largest contributor specializing in Comedies and Dramas(120)
- Japan stands out in anime series and tv shows(75)
- limited representation of genres like Sci-Fi,Horror and Thrillers in top listings

RECOMMENDATIONS :

- Increase Genre diversity in under represented categories like Horror and Thriller
- Boost regional content from high potential markets like k-dramas from South Korea
- Compare with competitor platforms to identify gaps
- Track viewer engagement to prioritize high-performing genres

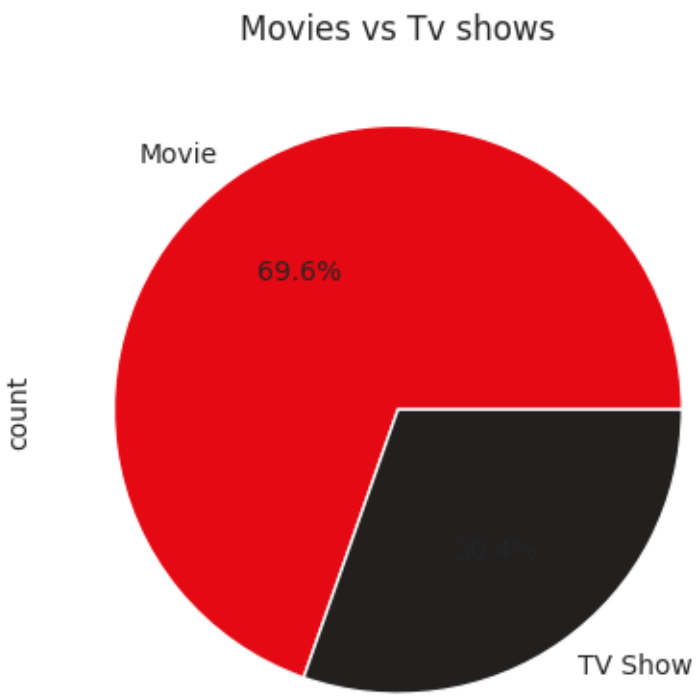
9 VISUAL ANALYSIS

[80]:
```
# Movies vs Tv shows

Netflix['type'].value_counts()
```

[80]:
```
type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

[81]:
```
Netflix['type'].value_counts().plot(kind='pie',autopct='%1.
  ↪1f%%',colors=['#E50914','#221F1F'])
plt.title('Movies vs Tv shows')
```

[81]:
```
Text(0.5, 1.0, 'Movies vs Tv shows')
```



INSIGHTS : * Movies dominate TV Shows in the content type * It is observed that movies percentage ratio is 69.6 % while TV Shows is 30.4%

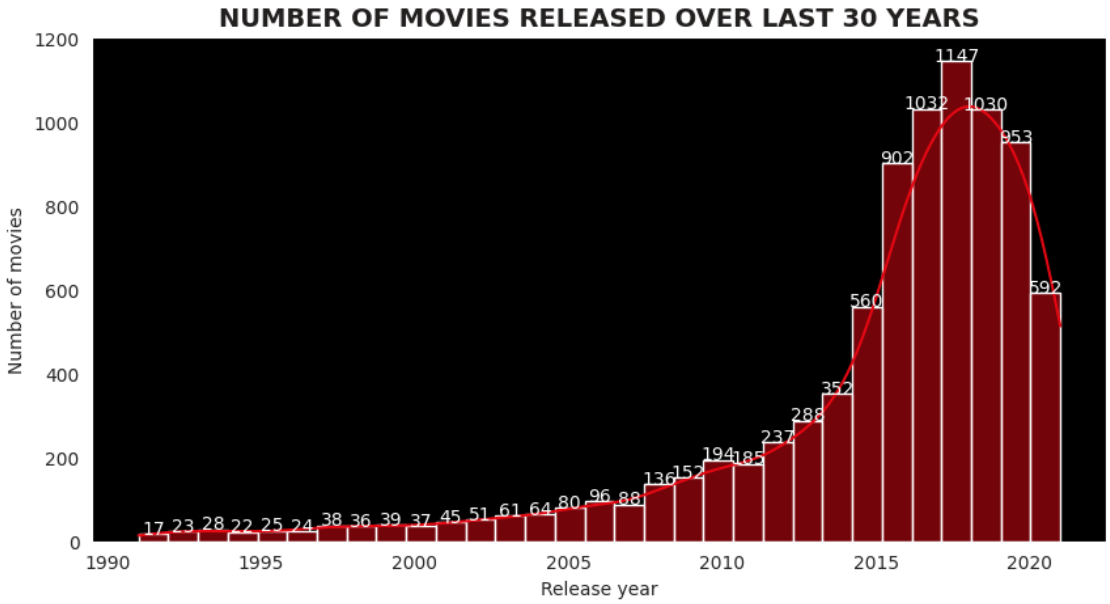
RECOMMENDATIONS : * Consider producing more tv shows inorder to balance the movies content * Collaborate with more directors to Produce TV Shows

[82]:
```
# How has the number of movies released per year changed over the last 20-30␣
 ↪years ?
current_year=2021
Last_30years=Netflix[(Netflix['release_year']>=(current_year-30)) &␣
 ↪(Netflix['release_year']<=current_year)]
Movies_per_year=Last_30years.
 ↪groupby('release_year')[['release_year']['type'=='Movies']].value_counts()
Movies_per_year
```

[82]:
```
release_year
1991      17
1992      23
1993      28
1994      22
1995      25
1996      24
1997      38
1998      36
1999      39
2000      37
2001      45
2002      51
2003      61
2004      64
2005      80
2006      96
2007      88
2008     136
2009     152
2010     194
2011     185
2012     237
2013     288
2014     352
2015     560
2016     902
2017    1032
2018    1147
2019    1030
2020     953
```

```
2021     592
Name: count, dtype: int64
```

[83]:
```
plt.figure(figsize=(10,5))
ax=sns.
 ↪histplot(data=Last_30years,x='release_year',kde=True,bins=31,color='#E50914')
plt.gca().set_facecolor('black')
plt.xlabel('Release year')
plt.ylabel('Number of movies')
plt.title('NUMBER OF MOVIES RELEASED OVER LAST 30␣
 ↪YEARS',fontsize=14,fontweight='bold',color='#221F1F')
plt.grid(False)
for bar in ax.patches:
   height=bar.get_height()
   ax.text(bar.get_x()+bar.get_width()/2.
 ↪,height,str(height),ha='center',color='white')
```



INSIGHTS :

- The Histogram shows a dramatic increase in movie releases peaking at 1147 in the most recent years
- Content releases have grown steadily from 2010s

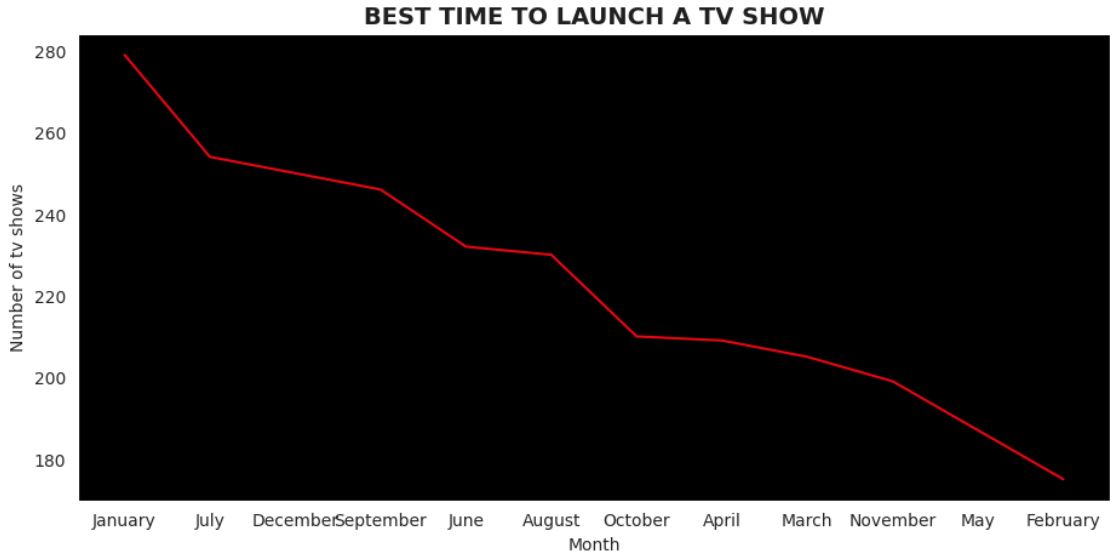
RECOMMENDATIONS :

- Focus on the most popular types of content and release more content as they keep viewers coming back to the platform over time
- Continue investing in digital-first releases and collaborations with online creators to stay ahead in the streaming industry

[84]:
```
# what is the best time to launch a tv show ?

Netflix['month']=Netflix['date_added'].dt.month_name()
TV_SHOWS=Netflix[Netflix['type']== 'TV Show']['month_name'].value_counts()
TV_SHOWS
```

[84]:
```
month_name
January      279
July         254
December     250
September    246
June         232
August       230
October      210
April        209
March        205
November     199
May          187
February     175
Name: count, dtype: int64
```

[85]:
```
plt.figure(figsize=(11,5))
ax=sns.lineplot(x=TV_SHOWS.index,y=TV_SHOWS.values,color='#E50914')
plt.ylabel('Number of tv shows')
plt.xlabel('Month')
plt.title('BEST TIME TO LAUNCH A TV␣
 ↪SHOW',fontsize=14,fontweight='bold',color='#221F1F')
plt.gca().set_facecolor('black')
plt.grid(False)
```



INSIGHTS :

- Based on my analysis January is the most peak month for TV Show launches
- Months from June to September also shows strong performance

RECOMMENDATIONS : * Launch more TV Shows during this period * Reserve flagship originals for peak months
* Use slower months for niche content or International releases * Analyze by Genre to see if certain shows perform better in specific months

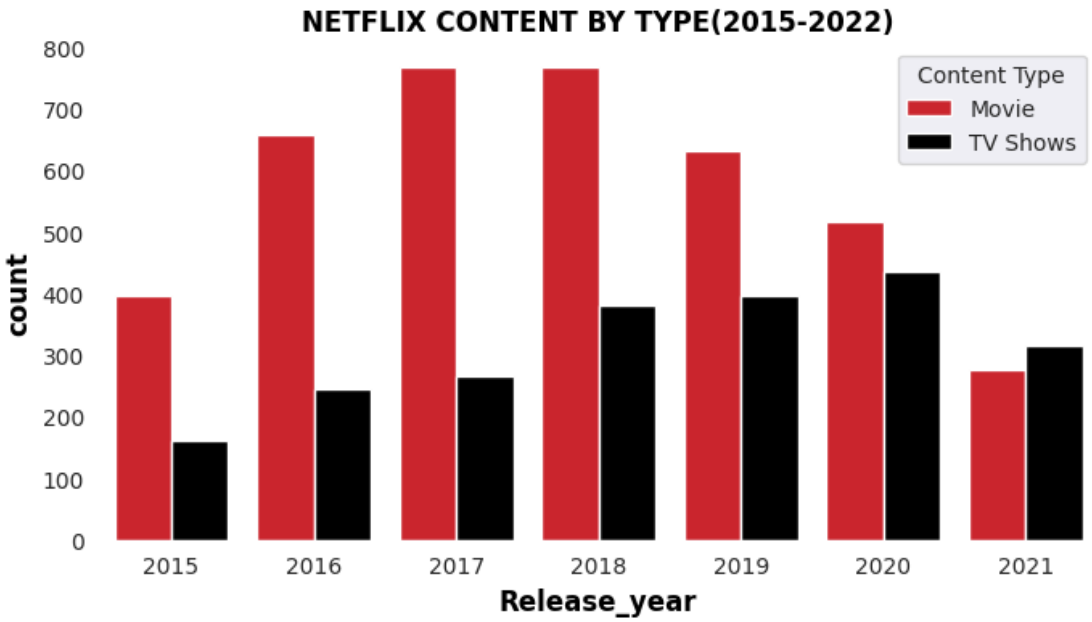
[86]:
```
# Does Netflix has more focus on TV Shows than Movies in recent years
Netflix.groupby('release_year')['type'].value_counts()
recent_year=Netflix[(Netflix['release_year']>=2015)&(Netflix['release_year']<=2022)]
Type_counts_=recent_year.
 ↪groupby(['release_year','type'])[['release_year','type']].value_counts()
Type_counts_
```

```
<ipython-input-86-13ef69e9d415>:4: FutureWarning: The default of observed=False
is deprecated and will be changed to True in a future version of pandas. Pass
observed=False to retain current behavior or observed=True to adopt the future
default and silence this warning.
  Type_counts_=recent_year.groupby(['release_year','type'])[['release_year','typ
e']].value_counts()
```

[86]:
```
release_year  type
2015          Movie      398
              TV Show    162
2016          Movie      658
              TV Show    244
2017          Movie      767
              TV Show    265
2018          Movie      767
              TV Show    380
2019          Movie      633
              TV Show    397
2020          Movie      517
              TV Show    436
2021          Movie      277
              TV Show    315
Name: count, dtype: int64
```

[87]:
```
plt.figure(figsize=(8,4))
ax=sns.
 ↪countplot(data=recent_year,x='release_year',hue='type',palette=['#E50914','black'])
plt.title('NETFLIX CONTENT BY␣
 ↪TYPE(2015-2022)',fontsize=12,fontweight='bold',color='black')
plt.gca().set_facecolor('white')
```

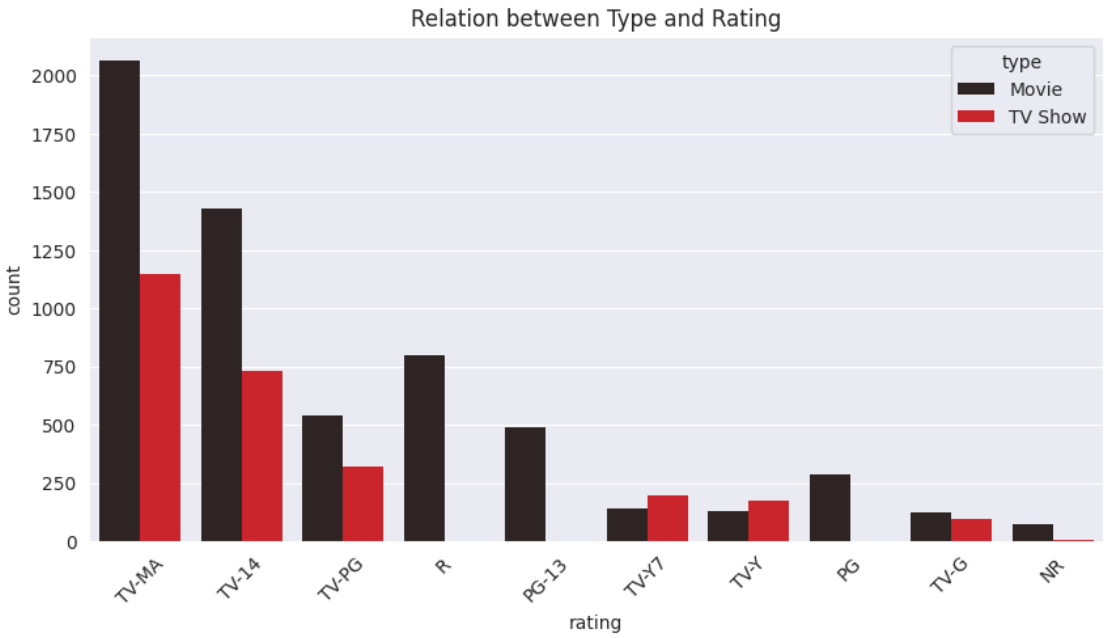
```
plt.xlabel('Release_year',fontsize=12,fontweight='bold',color='black')
plt.ylabel('count',fontsize=12,fontweight='bold',color='black')
ax.legend(title='Content Type',labels=['Movie','TV Shows'])
plt.show()
```



[88]:
```
# TV Shows and their ratings

Tv_show_ratings = Netflix['rating'].value_counts().nlargest(10).index
# # Filter the DataFrame for the top 10 ratings
Top_10_tv_shows= Netflix[Netflix['rating'].isin(Tv_show_ratings)]

plt.figure(figsize=(10,5))
ax = sns.countplot(x='rating', data=Top_10_tv_shows, hue='type',␣
 ↪color='#E50914',
                   order=Tv_show_ratings, dodge=True, edgecolor='none')
plt.title('Relation between Type and Rating')
plt.xticks(rotation=45)
sns.set_style('darkgrid')
plt.xticks(rotation=45)
sns.set_style('darkgrid')
```

```
<ipython-input-88-8eb746a5d2c4>:8: FutureWarning:

Setting a gradient palette using color= is deprecated and will be removed in
v0.14.0. Set `palette='dark:#E50914'` for the same effect.
```

```
  ax = sns.countplot(x='rating', data=Top_10_tv_shows, hue='type',
color='#E50914',
```



INSIGHTS :

- TV-MA is the most common viewer rating on Netflix, indicating a strong presence of content suitable for mature audiences, followed by 'TV-14' and 'TV-PG'

RECOMMENDATIONS :

- Continue producing mature content that differentiates Netflix
- Invest more in TV-14 rated shows for younger demographics
- Strengthen TV-Y/TV-Y7 offerings to attract family subscribers
- consider saperate kids profiles or content areas

[89]:
```
# Country with highest number of Movies ?

Country=Netflix[Netflix['type']=='Movie']['country'].value_counts()
Country.head(10)
```

[89]:
```
country
United States     2498
India              893
United Kingdom     206
Canada             122
Spain               97
Egypt               92
Nigeria             86
```

```
Indonesia            77
Japan                76
Turkey               76
Name: count, dtype: int64
```

[90]:
```
from wordcloud import WordCloud
```

[91]:
```
wordcloud = WordCloud(
    width=20000,
    height=10000,
    background_color='black',
    colormap='Reds').generate_from_frequencies(Country)
plt.imshow(wordcloud)
plt.axis('off')
plt.title('COUNTRIES WITH HIGHEST NUMBER OF MOVIES',fontsize=14)
```

[91]:
```
Text(0.5, 1.0, 'COUNTRIES WITH HIGHEST NUMBER OF MOVIES')
```

INSIGHTS :

- United States overwhelmingly leads with 2498 movies followed by India with 893 movies
- Countries likeUnited Kingdom and Canada also contribut a larger number of movies

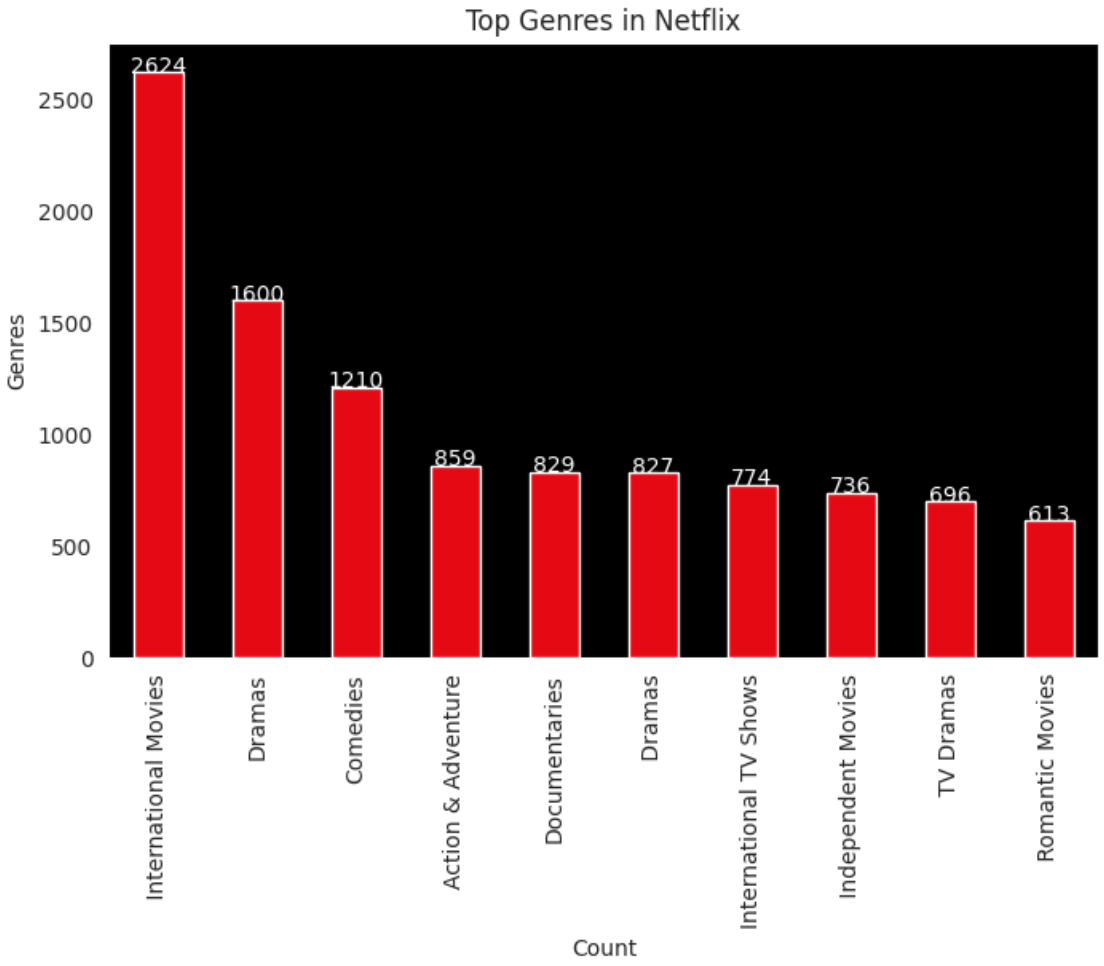
RECOMMENDATIONS :

- Continue partnering with U.S production studios while also promoting leser known American films to diversify the content
- Invest in acquiring or producing content from the under represented regions to expand global reach
- Offer multi language dubs or subtitles to improve accessibility

[92]:
```
# Top Genres in Netflix
Genres=Netflix['listed_in'].str.split(',',expand=True).stack().value_counts()
top_genres=Genres.head(10)
top_genres
```

[92]:
```
 International Movies     2624
Dramas                    1600
Comedies                  1210
Action & Adventure         859
Documentaries              829
 Dramas                    827
International TV Shows     774
 Independent Movies        736
 TV Dramas                 696
 Romantic Movies           613
Name: count, dtype: int64
```

[93]:
```
plt.figure(figsize=(8,5))
ax=top_genres.plot(kind='bar',color='#E50914')
plt.title('Top Genres in Netflix')
plt.xlabel('Count')
plt.ylabel('Genres')
plt.gca().set_facecolor('black')
plt.grid(False)
for bar in ax.patches:
   height=bar.get_height()
   ax.text(bar.get_x()+bar.get_width()/2.
 ↪,height,str(height),ha='center',color='white')
```



INSIGHTS :

- International Movies stands tops in the Genres with a total count of 2624
- Dramas and comedies Genre also produced a gretaer content

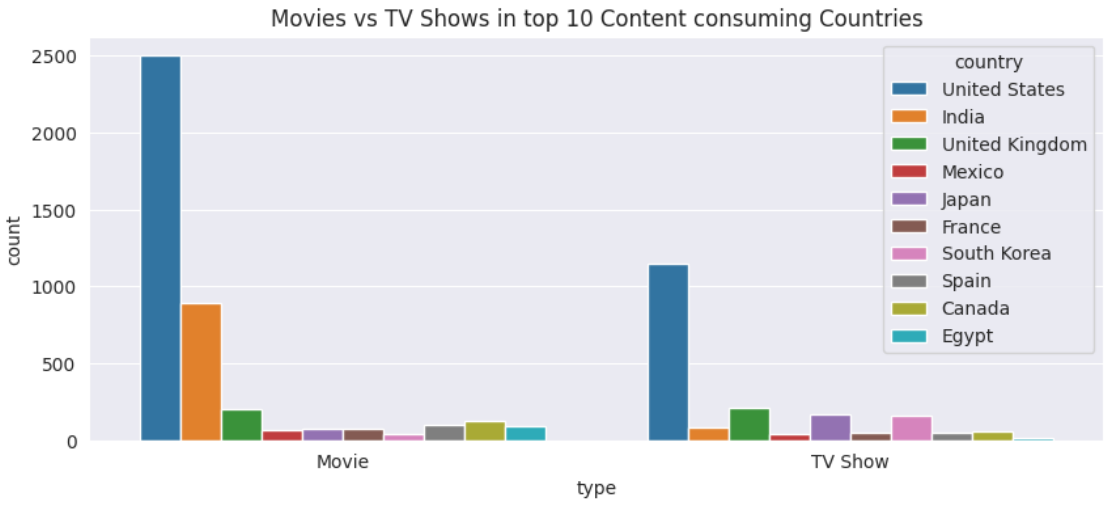
RECOMMENDATIONS :

- Since International Movies are highly popular,Netflix should continue investing in diverse sub-genres
- For Documentaries focus more on trending topics

[94]:
```
# Top Content Movies/TV Shows

Director_movies=Netflix[(Netflix['type']=='Movie') & (Netflix['director']!
  ↪='Unknown')][['director','type']].value_counts()
Director_movies
```

[94]:
```
director                  type
Rajiv Chilaka             Movie    19
Raúl Campos, Jan Suter    Movie    18
Suhas Kadav               Movie    16
Marcus Raboy              Movie    15
Jay Karas                 Movie    14
                                   ..
Jude Okwudiafor Johnson   Movie     1
Jude Weng                 Movie     1
Julia Hart                Movie     1
Julia Knowles             Movie     1
Juan Antin                Movie     1
Name: count, Length: 4354, dtype: int64
```

[95]:
```
plt.figure(figsize=(10,4))
top_movies=Netflix[Netflix['country'] != 'Unknown'].
 ↪groupby('country')['show_id'].nunique().sort_values(ascending=False).
 ↪head(10).index
type_count=Netflix['type'].value_counts().index
top_country_content=Netflix[(Netflix['country'].isin(top_movies)) &␣
 ↪(Netflix['type'].isin(type_count))]
sns.countplot(data=top_country_content,x='type',hue='country')
plt.title('Movies vs TV Shows in top 10 Content consuming Countries')
plt.show()
```



INSIGHT :

The above displays the distribution of Movies and tv shows for top 10 content consuming contries overall.

RECOMMENDATION :

Understanding the content consumption patterns in these top countries is crucial for our content strategy. We should analyze further to identify trends and preferences specific to each country, allowing us to tailor our content offerings accordingly. By aligning our content strategy with the preferences of audiences in these key markets, we can effectively engage viewers, enhance user satisfaction, and drive growth for our platform.

10 OVERALL INSIGHTS AND RECOMMENDATIONS

###**INSIGHTS** 1. There are 6131 unique movies and 2676 TV Shows on Netflix1. 2. Rajiv Chilaka is the most famous director with 19 movies followed by Raul Campos and Jan Suter with 18 movies 3. Friday is the most popular day for the new releases weekend days(saturday and sunday) has a fewest releases 4. United States leads in the content production especially in Documentaries(265) and stand-Up Comedy(240) 5. India is the second largest contributor specializing in Comedies and Dramas(120) 6. It is observed that movies percentage ratio is 69.6 % while TV Shows is 30.4% 7. TV-MA is the most common viewer rating on Netflix, indicating a strong presence of content suitable for mature audiences, followed by 'TV-14' and 'TV-PG' 8. United States overwhelmingly leads with 2498 movies followed by India with 893 movies

###**RECOMMENDATIONS:**

1. Consider producing more TV Shows inorder to balance the content as movies currently dominate 2.Maintain strong Friday release schedule but consider testing thursday release for weekend anticipation
2. Track viewer engagement to prioritize high-performing genres
3. Consider producing more tv shows inorder to balance the movies content
4. Continue investing in digital-first releases and collaborations with online creators to stay ahead in the streaming industry
5. Continue producing mature content that differentiates Netflix
6. consider saperate kids profiles or content areas
7. Continue partnering with U.S production studios while also promoting leser known American films to diversify the content
8. Invest in acquiring or producing content from the under represented regions to expand global reach
9. Offer multi language dubs or subtitles to improve accessibility