Netflix Business Case Study

Submitted by :

B. Raju Naik

Downloading the dataset using !gdown function for the analysis.

In [2]:
```
!gdown 'https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv'
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
To: /content/netflix.csv
100% 3.40M/3.40M [00:00<00:00, 28.6MB/s]
```

Read the downloaded data using pd.read_csv() function.

Basic Analysis

In [3]:
```
import pandas as pd
netflix_df = pd.read_csv('netflix.csv')
netflix_df.head()
```

Out[3]:

	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...

In [4]:
```
netflix_df.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
```

In [5]:
```
netflix_df.isnull().sum()
```

Out[5]:
```
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
```

In [6]:
```
netflix_df.shape
```

Out[6]:
```
(8807, 12)
```

1. Un-nesting the columns

a. Un-nest the columns those have cells with multiple comma separated values by creating multiple rows

a. Unnesting cast column

INPUT:

In [7]:
```
import pandas as pd
#Splitting of the comma separated elements into a list
netflix_df['cast'] = netflix_df['cast'].str.split(',')
#Exploding the list elements into separate rows
unnested_netflix_df = netflix_df.explode('cast', ignore_index=True)
unnested_netflix_df
```

Out[7]:

	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	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	s2	TV Show	Blood & Water	NaN	Khosi Ngema	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...
3	s2	TV Show	Blood & Water	NaN	Gail Mabalane	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...
4	s2	TV Show	Blood & Water	NaN	Thabang Molaba	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...
...	...	...	...	...	...	...	...	...	...	...	...	...
64946	s8807	Movie	Zubaan	Mozez Singh	Manish Chaudhary	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...
64947	s8807	Movie	Zubaan	Mozez Singh	Meghna Malik	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
												his way into a ty...
64948	s8807	Movie	Zubaan	Mozez Singh	Malkeet Rauni	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...
64949	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...
64950	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

64951 rows × 12 columns

In [8]:
```
unnested_netflix_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64951 entries, 0 to 64950
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       64951 non-null  object
 1   type          64951 non-null  object
 2   title         64951 non-null  object
 3   director      45938 non-null  object
 4   cast          64126 non-null  object
 5   country       59898 non-null  object
 6   date_added    64882 non-null  object
 7   release_year  64951 non-null  int64 
 8   rating        64913 non-null  object
 9   duration      64948 non-null  object
 10  listed_in     64951 non-null  object
 11  description   64951 non-null  object
dtypes: int64(1), object(11)
memory usage: 5.9+ MB
```

In [9]:
```
(unnested_netflix_df.isna().sum()/len(netflix_df))*100
```

Out[9]:
```
show_id          0.000000
type             0.000000
title            0.000000
director       215.885091
cast             9.367549
country         57.374815
date_added       0.783468
release_year     0.000000
rating           0.431475
duration         0.034064
listed_in        0.000000
description      0.000000
dtype: float64
```

2. Handling null values

a. For categorical variables with null values, update those rows as unknown_column_name. Example : Replace missing value with Unknown Actor for missing value in Actors column.

b. Replace with 0 for continuous variables having null values.

INPUT:

In [10]:
```
import pandas as pd
unnested_netflix_df['director'] = unnested_netflix_df['director'].fillna('Unknown_Director')
unnested_netflix_df['country'] = unnested_netflix_df['country'].fillna('Unknown_Country')
unnested_netflix_df['cast'] = unnested_netflix_df['cast'].fillna('Unknown_Cast')
unnested_netflix_df['date_added'] = unnested_netflix_df['date_added'].fillna('Unknown_date_added')
unnested_netflix_df['rating'] = unnested_netflix_df['rating'].fillna('Unknown_rating')
unnested_netflix_df['duration'] = unnested_netflix_df['duration'].fillna('Unknown_duration')
unnested_netflix_df
```

Out[10]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descrip
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown_Cast	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As father n the er his film
1	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	A cros paths par Cape T
2	s2	TV Show	Blood & Water	Unknown_Director	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	A cros paths par Cape T
3	s2	TV Show	Blood & Water	Unknown_Director	Gail Mabalane	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	A cros paths par Cape T
4	s2	TV Show	Blood & Water	Unknown_Director	Thabang Molaba	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	A cros paths par Cape T
...	...	...	...	...	...	...	...	...	...	...	...	
64946	s8807	Movie	Zubaan	Mozez Singh	Manish Chaudhary	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scra but boy wo his way a
64947	s8807	Movie	Zubaan	Mozez Singh	Meghna Malik	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scra but boy wo

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descrip
												his way a
64948	s8807	Movie	Zubaan	Mozez Singh	Malkeet Rauni	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scra but boy wo his way a
64949	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scra but boy wo his way a
64950	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scra but boy wo his way a

64951 rows × 12 columns

In [11]:
```
unnested_netflix_df.isna().sum()
```

Out[11]:
```
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
dtype: int64
```

In [12]:
```
unnested_netflix_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64951 entries, 0 to 64950
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       64951 non-null  object
 1   type          64951 non-null  object
 2   title         64951 non-null  object
 3   director      64951 non-null  object
 4   cast          64951 non-null  object
 5   country       64951 non-null  object
 6   date_added    64951 non-null  object
 7   release_year  64951 non-null  int64 
 8   rating        64951 non-null  object
 9   duration      64951 non-null  object
 10  listed_in     64951 non-null  object
 11  description   64951 non-null  object
dtypes: int64(1), object(11)
memory usage: 5.9+ MB
```

Unnesting using country column

In [13]:
```
unnested_netflix_df['country'] = unnested_netflix_df['country'].str.split(',')
unnested_netflix_df = unnested_netflix_df.explode('country', ignore_index=True)
unnested_netflix_df
```

Out[13]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descript
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown_Cast	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As father ne the en his film
1	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	A cross paths part Cape T
2	s2	TV Show	Blood & Water	Unknown_Director	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	A cross paths part Cape T
3	s2	TV Show	Blood & Water	Unknown_Director	Gail Mabalane	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	A cross paths part Cape T
4	s2	TV Show	Blood & Water	Unknown_Director	Thabang Molaba	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	A cross paths part Cape T
...	...	...	...	...	...	...	...	...	...	...	...	
81736	s8807	Movie	Zubaan	Mozez Singh	Manish Chaudhary	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scra but p boy wo his way a
81737	s8807	Movie	Zubaan	Mozez Singh	Meghna Malik	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scra but p boy wo

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descript
												his way a
81738	s8807	Movie	Zubaan	Mozez Singh	Malkeet Rauni	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scra but p boy wo his way a
81739	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scra but p boy wo his way a
81740	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scra but p boy wo his way a

81741 rows × 12 columns

In [14]:
```
unnested_netflix_df.shape
```

Out[14]:
```
(81741, 12)
```

In [15]:
```
unnested_netflix_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 81741 entries, 0 to 81740
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       81741 non-null  object
 1   type          81741 non-null  object
 2   title         81741 non-null  object
 3   director      81741 non-null  object
 4   cast          81741 non-null  object
 5   country       81741 non-null  object
 6   date_added    81741 non-null  object
 7   release_year  81741 non-null  int64 
 8   rating        81741 non-null  object
 9   duration      81741 non-null  object
 10  listed_in     81741 non-null  object
 11  description   81741 non-null  object
dtypes: int64(1), object(11)
memory usage: 7.5+ MB
```

In [16]:
```
unnested_netflix_df.isna().sum()
```

Out[16]:
```
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
dtype: int64
```

Unnesting using listed_in columns

In [17]:
```
unnested_netflix_df['listed_in'] = unnested_netflix_df['listed_in'].str.split(',')
unnested_netflix_df = unnested_netflix_df.explode('listed_in', ignore_index=True)
unnested_netflix_df
```

Out[17]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descri
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown_Cast	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	A father the e hi fil
1	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	cro path pa Cape
2	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Dramas	cro path pa Cape
3	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Mysteries	cro path pa Cape
4	s2	TV Show	Blood & Water	Unknown_Director	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	cro path pa Cape
...	...	...	...	...	...	...	...	...	...	...	...	
186394	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min	International Movies	A sc but boy w his wa
186395	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min	Music & Musicals	A sc but boy w

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descri
												his wa
186396	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min	Dramas	A scı but boy w his wa
186397	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min	International Movies	A scı but boy w his wa
186398	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min	Music & Musicals	A scı but boy w his wa

186399 rows × 12 columns

In [18]:
```
unnested_netflix_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 186399 entries, 0 to 186398
Data columns (total 12 columns):
 #   Column        Non-Null Count   Dtype 
---  ------        --------------   ----- 
 0   show_id       186399 non-null  object
 1   type          186399 non-null  object
 2   title         186399 non-null  object
 3   director      186399 non-null  object
 4   cast          186399 non-null  object
 5   country       186399 non-null  object
 6   date_added    186399 non-null  object
 7   release_year  186399 non-null  int64 
 8   rating        186399 non-null  object
 9   duration      186399 non-null  object
 10  listed_in     186399 non-null  object
 11  description   186399 non-null  object
dtypes: int64(1), object(11)
memory usage: 17.1+ MB
```

In [ ]:
```
unnested_netflix_df.isna().sum()
```

Out[ ]:
```
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
dtype: int64
```

Data Exploration

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

a. For Non-graphical Analysis:

INPUT:

In [ ]:
```
import pandas as pd
from IPython.display import display

Categorical_columns = ['show_id','type','title','director','cast','country','date_added','release_year','rating','dura

for column in Categorical_columns:
  print(f"\n\nValue counts for column '{column}' : ")
  df = pd.DataFrame(netflix_df[column].value_counts().reset_index(name = 'count'))
  display(df)
```

```
Value counts for column 'show_id' :
```

	index	count
0	s1	1
1	s5875	1
2	s5869	1
3	s5870	1
4	s5871	1
...	...	...
8802	s2931	1
8803	s2930	1
8804	s2929	1
8805	s2928	1
8806	s8807	1

8807 rows × 2 columns

```
Value counts for column 'type' :
```

	index	count
0	Movie	6131
1	TV Show	2676

```
Value counts for column 'title' :
```

	index	count
0	Dick Johnson Is Dead	1
1	Ip Man 2	1
2	Hannibal Buress: Comedy Camisado	1
3	Turbo FAST	1
4	Masha's Tales	1
...	...	...
8802	Love for Sale 2	1
8803	ROAD TO ROMA	1
8804	Good Time	1
8805	Captain Underpants Epic Choice-o-Rama	1
8806	Zubaan	1

8807 rows × 2 columns

```
Value counts for column 'director' :
```

	index	count
0	Rajiv Chilaka	19
1	Raúl Campos, Jan Suter	18
2	Marcus Raboy	16
3	Suhas Kadav	16
4	Jay Karas	14
...	...	...
4523	Raymie Muzquiz, Stu Livingston	1
4524	Joe Menendez	1
4525	Eric Bross	1
4526	Will Eisenberg	1
4527	Mozez Singh	1

4528 rows × 2 columns

```
Value counts for column 'cast' :
```

	index	count
0	[David Attenborough]	19
1	[Vatsal Dubey, Julie Tejwani, Rupa Bhimani, ...	14
2	[Samuel West]	10
3	[Jeff Dunham]	7
4	[David Spade, London Hughes, Fortune Feimster]	6
...	...	...
7687	[Michael Peña, Diego Luna, Tenoch Huerta, J...	1
7688	[Nick Lachey, Vanessa Lachey]	1
7689	[Takeru Sato, Kasumi Arimura, Haru, Kentaro...	1
7690	[Toyin Abraham, Sambasa Nzeribe, Chioma Chuk...	1
7691	[Vicky Kaushal, Sarah-Jane Dias, Raaghav Cha...	1

7692 rows × 2 columns

```
Value counts for column 'country' :
```

	index	count
0	United States	2818
1	India	972
2	United Kingdom	419
3	Japan	245
4	South Korea	199
...	...	...
743	Romania, Bulgaria, Hungary	1
744	Uruguay, Guatemala	1
745	France, Senegal, Belgium	1
746	Mexico, United States, Spain, Colombia	1
747	United Arab Emirates, Jordan	1

748 rows × 2 columns

```
Value counts for column 'date_added' :
```

	index	count
0	January 1, 2020	109
1	November 1, 2019	89
2	March 1, 2018	75
3	December 31, 2019	74
4	October 1, 2018	71
...	...	...
1762	December 4, 2016	1
1763	November 21, 2016	1
1764	November 19, 2016	1
1765	November 17, 2016	1
1766	January 11, 2020	1

1767 rows × 2 columns

```
Value counts for column 'release_year' :
```

	index	count
0	2018	1147
1	2017	1032
2	2019	1030
3	2020	953
4	2016	902
...	...	...
69	1959	1
70	1925	1
71	1961	1
72	1947	1
73	1966	1

74 rows × 2 columns

```
Value counts for column 'rating' :
```

	index	count
0	TV-MA	3207
1	TV-14	2160
2	TV-PG	863
3	R	799
4	PG-13	490
5	TV-Y7	334
6	TV-Y	307
7	PG	287
8	TV-G	220
9	NR	80
10	G	41
11	TV-Y7-FV	6
12	NC-17	3
13	UR	3
14	74 min	1
15	84 min	1
16	66 min	1

```
Value counts for column 'duration' :
```

	index	count
0	1 Season	1793
1	2 Seasons	425
2	3 Seasons	199
3	90 min	152
4	94 min	146
...	...	...
215	16 min	1
216	186 min	1
217	193 min	1
218	189 min	1
219	191 min	1

220 rows × 2 columns

```
Value counts for column 'listed_in' :
```

	index	count
0	Dramas, International Movies	362
1	Documentaries	359
2	Stand-Up Comedy	334
3	Comedies, Dramas, International Movies	274
4	Dramas, Independent Movies, International Movies	252
...	...	...
509	Kids' TV, TV Action & Adventure, TV Dramas	1
510	TV Comedies, TV Dramas, TV Horror	1
511	Children & Family Movies, Comedies, LGBTQ Movies	1
512	Kids' TV, Spanish-Language TV Shows, Teen TV S...	1
513	Cult Movies, Dramas, Thrillers	1

514 rows × 2 columns

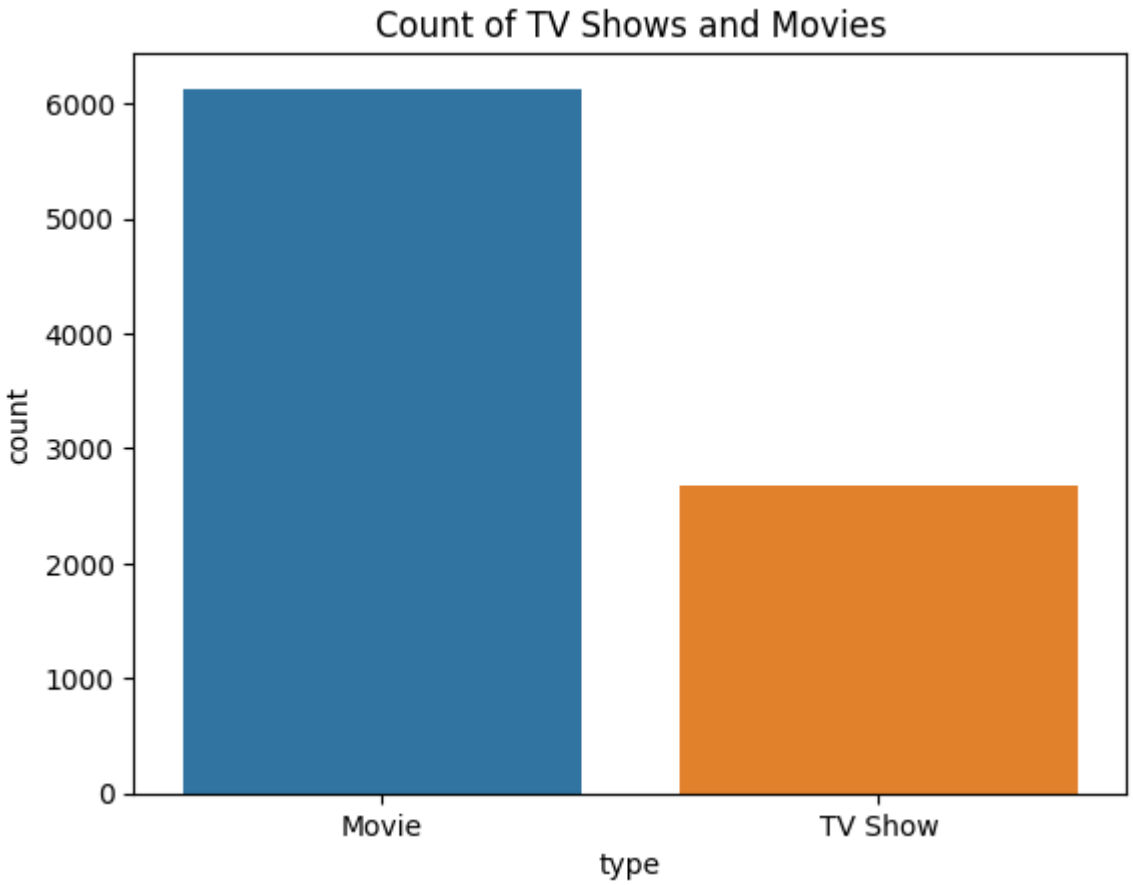
```
Value counts for column 'description' :
```

	index	count
0	Paranormal activity at a lush, abandoned prope...	4
1	Challenged to compose 100 songs before he can ...	3
2	A surly septuagenarian gets another chance at ...	3
3	Multiple women report their husbands as missin...	3
4	Secrets bubble to the surface after a sensual ...	2
...	...	...
8770	Sent away to evade an arranged marriage, a 14-...	1
8771	When his partner in crime goes missing, a smal...	1
8772	During 1962's Cuban missile crisis, a troubled...	1
8773	A teen's discovery of a vintage Polaroid camer...	1
8774	A scrappy but poor boy worms his way into a ty...	1

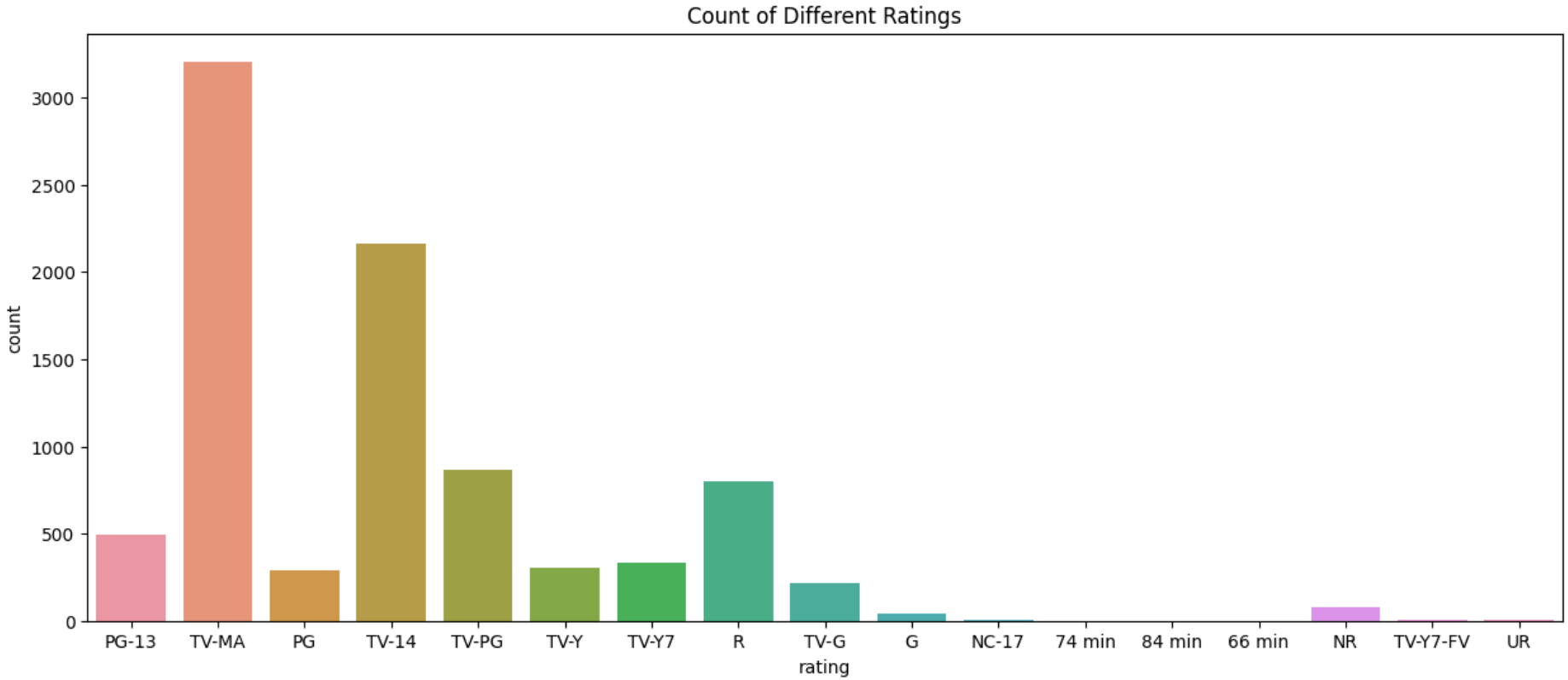
8775 rows × 2 columns

b. For graphical analysis:

In [20]:
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.countplot(x='type', data=netflix_df)
plt.title('Count of TV Shows and Movies')
plt.savefig('count_plot.png')
plt.show()
```



In [22]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(15, 6))
sns.countplot(x='rating', data=netflix_df)
plt.title('Count of Different Ratings')
plt.savefig('count_plot.png')
plt.show()
```



Insights:

1. There are 6131 Movies and 2676 Tv shows are present.
2. There are 4528 unique directors. Rajiv chilaka has directed the most.
3. In the netflix data after unnesting the cast column we found that, there are 64951 actors.
4. We observed that this data frame contains 16789 countries in which netflix has released its TV shows and Movies.
5. There are 16 different kinds of ratings provided by netflix in which TV-MA has the highest rating.
6. Netflix has wide variety of genres i.e, nearly 514 genres.

2. Comparison of Tv shows vs. movies.

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

Input

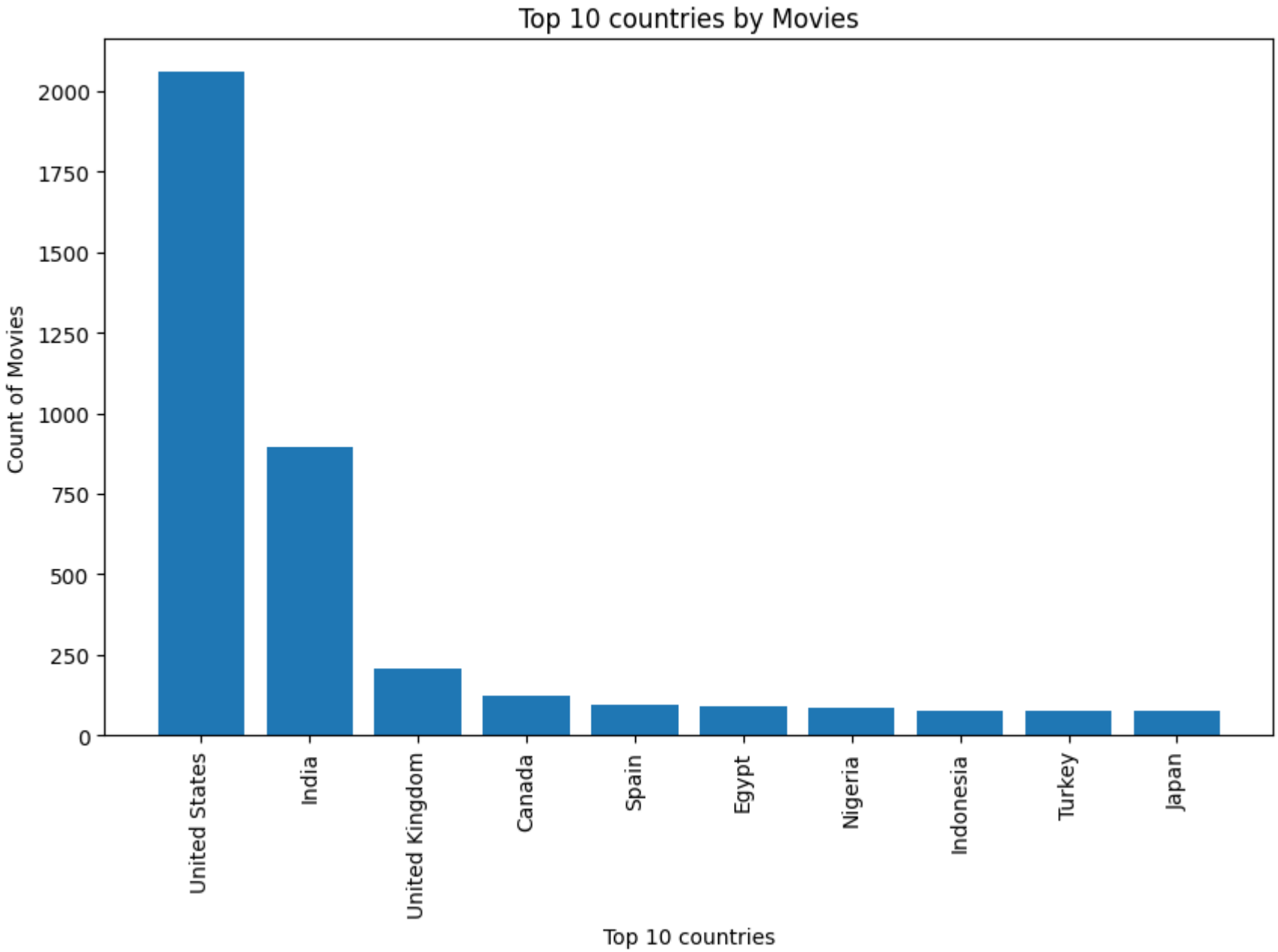
In [24]:
```
import pandas as pd

netflix_df = pd.read_csv('netflix.csv')

movies_df = netflix_df[netflix_df['type'] == 'Movie']


country_movies_count = movies_df.groupby('country')['title'].count().reset_index(name = 'Movie_titles_count')


top_10_countries = country_movies_count.sort_values(by = 'Movie_titles_count', ascending = False,).head(10)

top_10_countries
```

Out[24]:

	country	Movie_titles_count
525	United States	2058
218	India	893
440	United Kingdom	206
50	Canada	122
384	Spain	97
128	Egypt	92
319	Nigeria	86
238	Indonesia	77
428	Turkey	76
278	Japan	76

Plotting of above data

In [25]:
```
plt.figure(figsize=(10,6))
plt.bar(x=top_10_countries['country'],height=top_10_countries['Movie_titles_count'])
plt.xticks(rotation=90)
plt.xlabel('Top 10 countries')
plt.ylabel('Count of Movies')
plt.title('Top 10 countries by Movies')
plt.savefig('count_plot.png')
plt.show()
```



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

INPUT:

In [26]:
```
import pandas as pd

netflix_df = pd.read_csv('netflix.csv')

TVshows_df = netflix_df[netflix_df['type'] == 'TV Show']

country_TVshows_count = TVshows_df.groupby('country')['title'].count().reset_index(name = 'TV_Show_titles_count')


top_10_countries_Tvshows = country_TVshows_count.sort_values(by = 'TV_Show_titles_count', ascending = False,).head(10)

top_10_countries_Tvshows
```

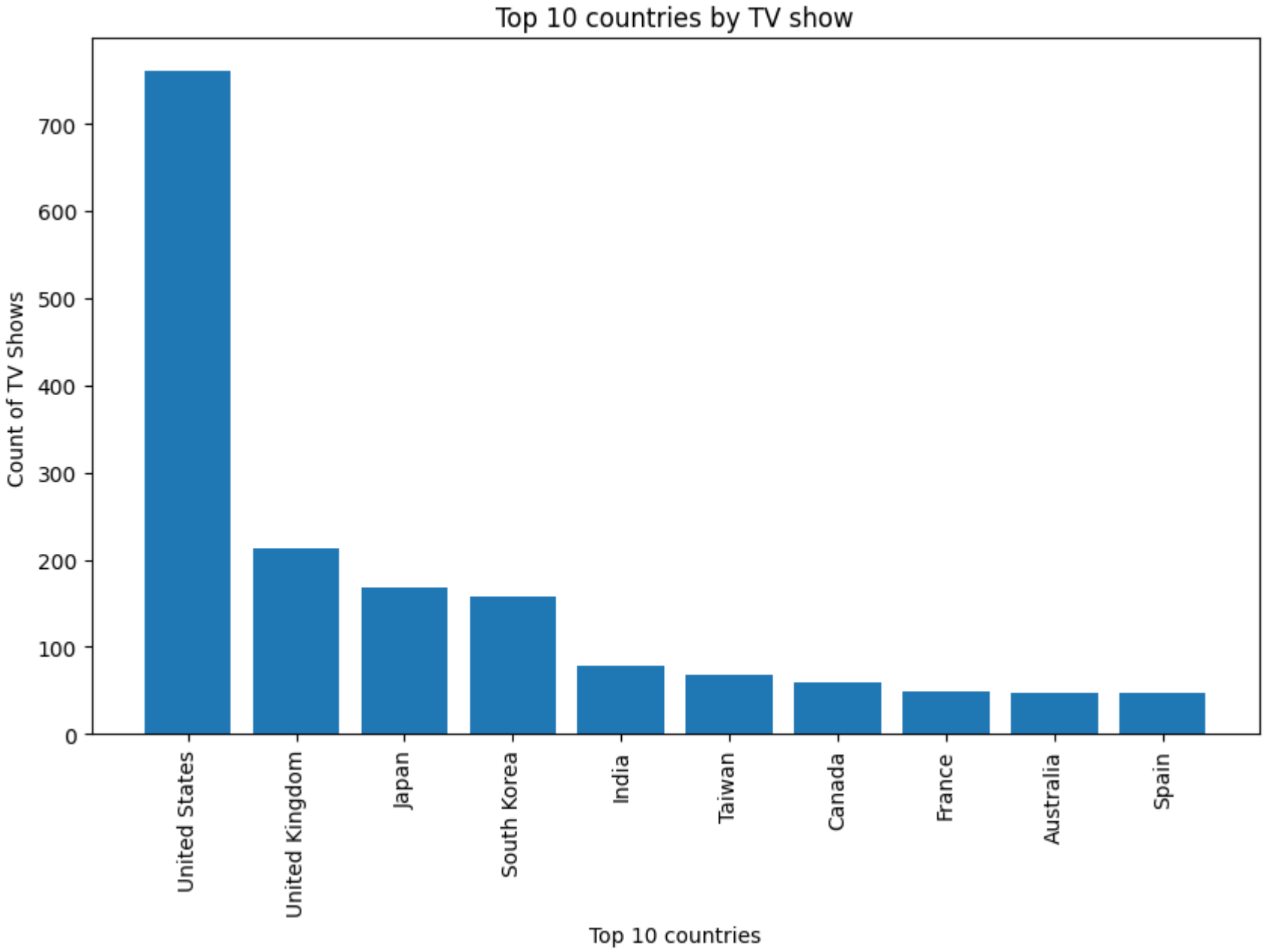
Out[26]:

	country	TV_Show_titles_count
160	United States	760
140	United Kingdom	213
83	Japan	169
120	South Korea	158
66	India	79
132	Taiwan	68
17	Canada	59
47	France	49
4	Australia	48
125	Spain	48

Plotting of above data

In [27]:
```
plt.figure(figsize=(10,6))
plt.bar(x=top_10_countries_Tvshows['country'],height=top_10_countries_Tvshows['TV_Show_titles_count'])
plt.xticks(rotation=90)
plt.xlabel('Top 10 countries')
plt.ylabel('Count of TV Shows')
plt.title('Top 10 countries by TV show')
```

```
plt.savefig('count_plot.png')
plt.show()
```



Insights:

1. The **Top 10 countries by Movies** Barchart reveals that United States, India and UK are the top 3 countries where most of the movies are produced.
2. The **Top 10 countries by TV show** Barchart reveals that United States, United Kingdom and Japan are the top 3 countries where most of the TV shows are produced.
3. India produces more number of Movies when compared to TV shows.

Recommendations:

In the least popular countries, the Netflix has to release the Tv shows and Movies in their regional languages in order to get more subscibers and it should also reduce subscription tariffs which is affordable by lower income countries also.

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

a. Find which is the best week to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies

INPUT:

In [28]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

netflix_df = pd.read_csv('netflix.csv')

netflix_df['date_added'] = pd.to_datetime(netflix_df['date_added'], errors='coerce')



netflix_df['week_added'] = netflix_df['date_added'].dt.isocalendar().week.astype('Int64')



max_week_tvshows = netflix_df[netflix_df['type'] == 'TV Show'].groupby(by = 'week_added').count()
```

```
max_week_movies = netflix_df[netflix_df['type'] == 'Movie'].groupby(by = 'week_added').count()


max_title_tvshows = max_week_tvshows['title'].idxmax()
max_title_movie = max_week_movies['title'].idxmax()

max_title_tvshows_plot = max_week_tvshows['title'].sort_values(ascending=False).iloc[:10]
max_title_movie_plot = max_week_movies['title'].sort_values(ascending=False).iloc[:10]



print("Week with the maximum number of TV shows added:",max_title_tvshows,'th Week', end = "")
print('\n')
print("Week with the maximum number of Movies added:",max_title_movie,'st Week', end = '')
```

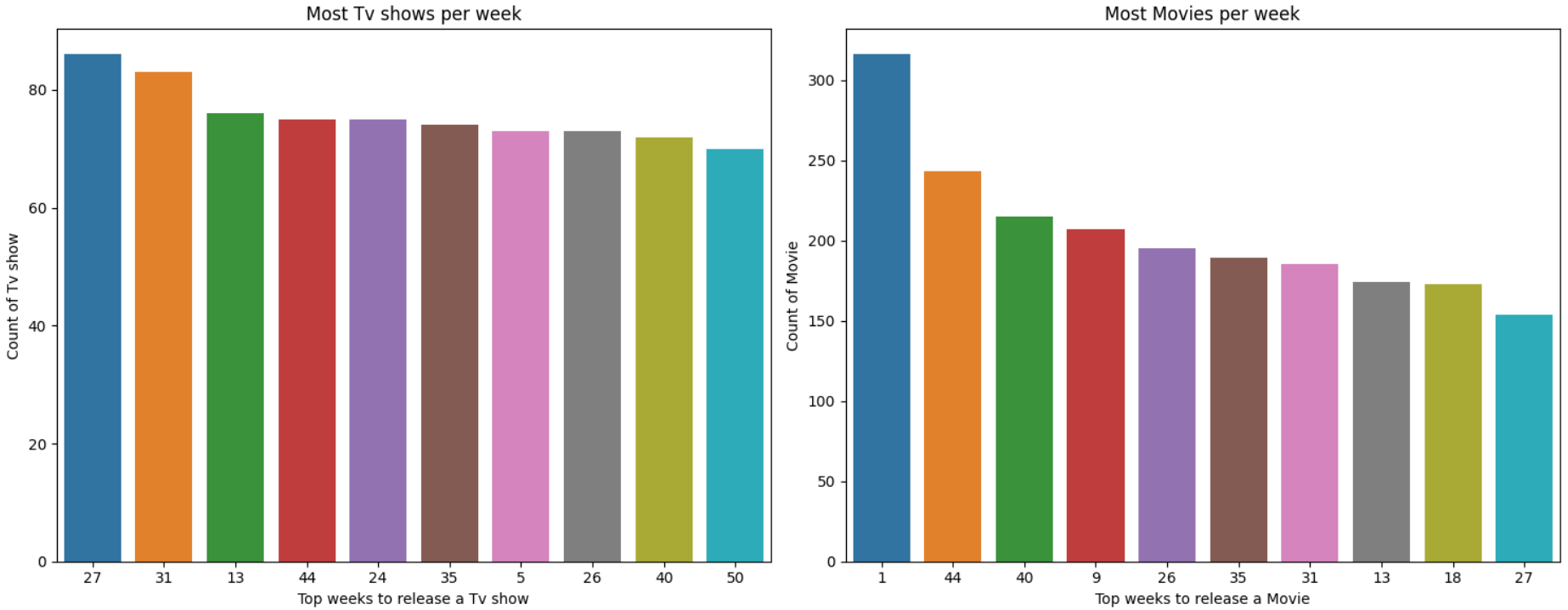
```
Week with the maximum number of TV shows added: 27 th Week

Week with the maximum number of Movies added: 1 st Week
```

Plotting of above data

In [33]:

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


fig, axes = plt.subplots(1, 2, figsize=(15, 6))


sns.barplot(x=max_title_tvshows_plot.index, y=max_title_tvshows_plot.values, order=max_title_tvshows_plot.index, ax=ax
axes[0].set_xlabel('Top weeks to release a Tv show')
axes[0].set_ylabel('Count of Tv show')
axes[0].set_title('Most Tv shows per week')


sns.barplot(x=max_title_movie_plot.index, y=max_title_movie_plot.values, order=max_title_movie_plot.index, ax=axes[1])
axes[1].set_xlabel('Top weeks to release a Movie')
axes[1].set_ylabel('Count of Movie')
axes[1].set_title('Most Movies per week')

plt.savefig('count_plot.png')

plt.tight_layout()
```

```
plt.show()
```



Insights:

1. Based on the provided data and chart, it can be discerned that the optimal time to release a TV show is during the 27th week of the year, while the most favorable week for releasing a movie appears to be the 1st week of the year.

Recommendation:

This data provides valuable insights for viewers looking to stay ahead of upcoming releases, particularly during these highlighted weeks. Netflix strategically chooses these periods to unveil compelling TV shows, aiming to maximize viewership and generate substantial profits.

b. Find which is the best month to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies

INPUT

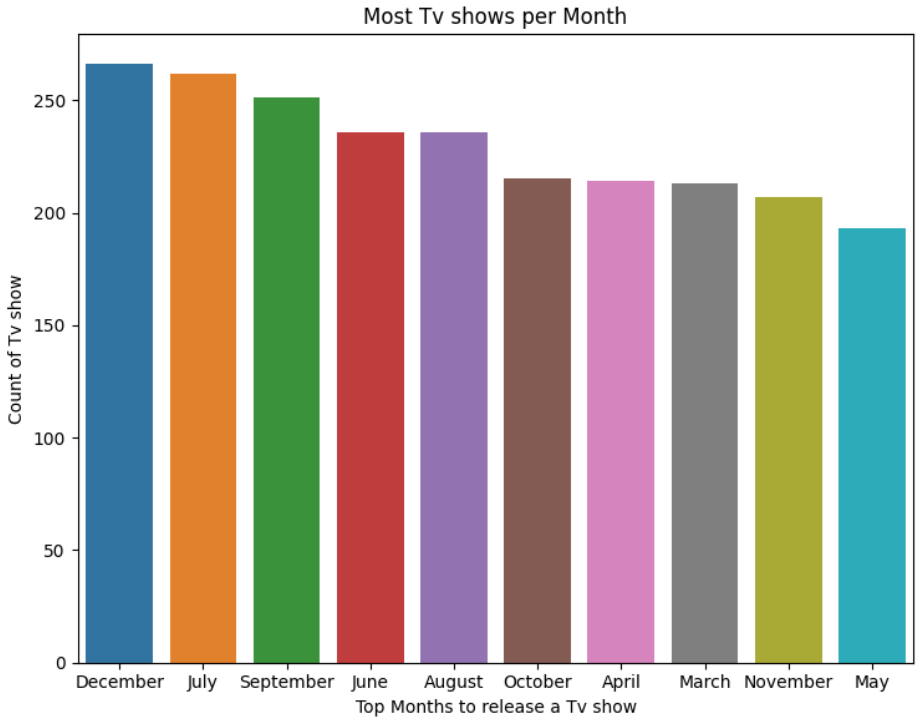
In [34]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import calendar

netflix_df = pd.read_csv('netflix.csv')

netflix_df['date_added'] = pd.to_datetime(netflix_df['date_added'])
netflix_df['month_added'] = netflix_df['date_added'].dt.month.astype('Int64')

netflix_df_tvshow = netflix_df[netflix_df['type'] == 'TV Show'].groupby(by='month_added').count()
netflix_df_movie = netflix_df[netflix_df['type'] == 'Movie'].groupby(by='month_added').count()

max_month_tvshow = calendar.month_name[netflix_df_tvshow['title'].idxmax()]
max_month_movie = calendar.month_name[netflix_df_movie['title'].idxmax()]

max_month_tvshow_plot = netflix_df_tvshow['title'].sort_values(ascending=False).iloc[:10]
max_month_movie_plot = netflix_df_movie['title'].sort_values(ascending=False).iloc[:10]

print("Month with the maximum number of TV shows added:", max_month_tvshow, end="")
print('\n')
print("Month with the maximum number of Movies added:", max_month_movie, end="")
```

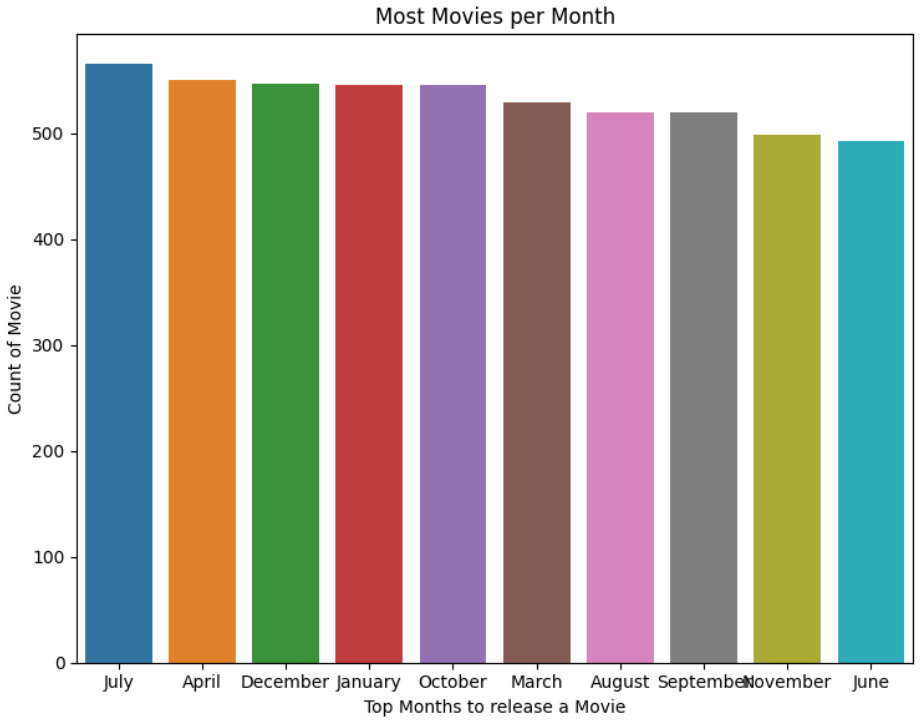
```
Month with the maximum number of TV shows added: December

Month with the maximum number of Movies added: July
```

Plotting of above data

In [35]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

fig, axes = plt.subplots(1, 2, figsize=(15, 6))

# Convert month numbers to month names for plotting
month_names = [calendar.month_name[i] for i in max_month_tvshow_plot.index]

# Plot TV shows subplot
sns.barplot(x=month_names, y=max_month_tvshow_plot.values, order=month_names, ax=axes[0])
axes[0].set_xlabel('Top Months to release a Tv show')
axes[0].set_ylabel('Count of Tv show')
axes[0].set_title('Most Tv shows per Month')
```

```
# Convert month numbers to month names for plotting
month_names = [calendar.month_name[i] for i in max_month_movie_plot.index]

# Plot Movies subplot
sns.barplot(x=month_names, y=max_month_movie_plot.values, order=month_names, ax=axes[1])
axes[1].set_xlabel('Top Months to release a Movie')
axes[1].set_ylabel('Count of Movie')
axes[1].set_title('Most Movies per Month')

plt.savefig('count_plot.png')

plt.tight_layout()
plt.show()
```





Insights:

1. July stands out as the month during which Netflix predominantly expands its movie content library, while December takes the spotlight as the prime month for the release of new TV shows. This observation hints at a deliberate and strategic approach to content release,

possibly aligning with seasonal patterns or audience preferences throughout the year.

Recommendations:

This dataset offers valuable insights for viewers seeking to stay informed about upcoming releases, especially during these highlighted months. Netflix strategically selects these timeframes to unveil captivating TV shows, with the goal of optimizing viewership and generating significant engagement and revenue.

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

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

INPUT:

In [36]:
```
import pandas as pd

netflix_df = pd.read_csv('netflix.csv')

netflix_df['cast'] = netflix_df['cast'].str.split(',')


netflix_df = netflix_df.explode('cast', ignore_index=True)

top10_tvshow = netflix_df[netflix_df['type']
                          == 'TV Show'].groupby(by = 'cast').size().reset_index(name = 'title').sort_values(by = 'tit

top10_tvshow_df = pd.DataFrame(top10_tvshow)

top10_movies = netflix_df[netflix_df['type']
                          == 'Movie'].groupby(by = 'cast').size().reset_index(name = 'title').sort_values(by = 'title
top10_movies_df = pd.DataFrame(top10_movies)

print("THE TOP 10 ACTORS APPEARED IN MOST TV shows :")
display(top10_tvshow_df)

print("\nTHE TOP 10 ACTORS APPEARED IN MOST MOVIES :")
display(top10_movies_df)
```

THE TOP 10 ACTORS APPEARED IN MOST TV shows :

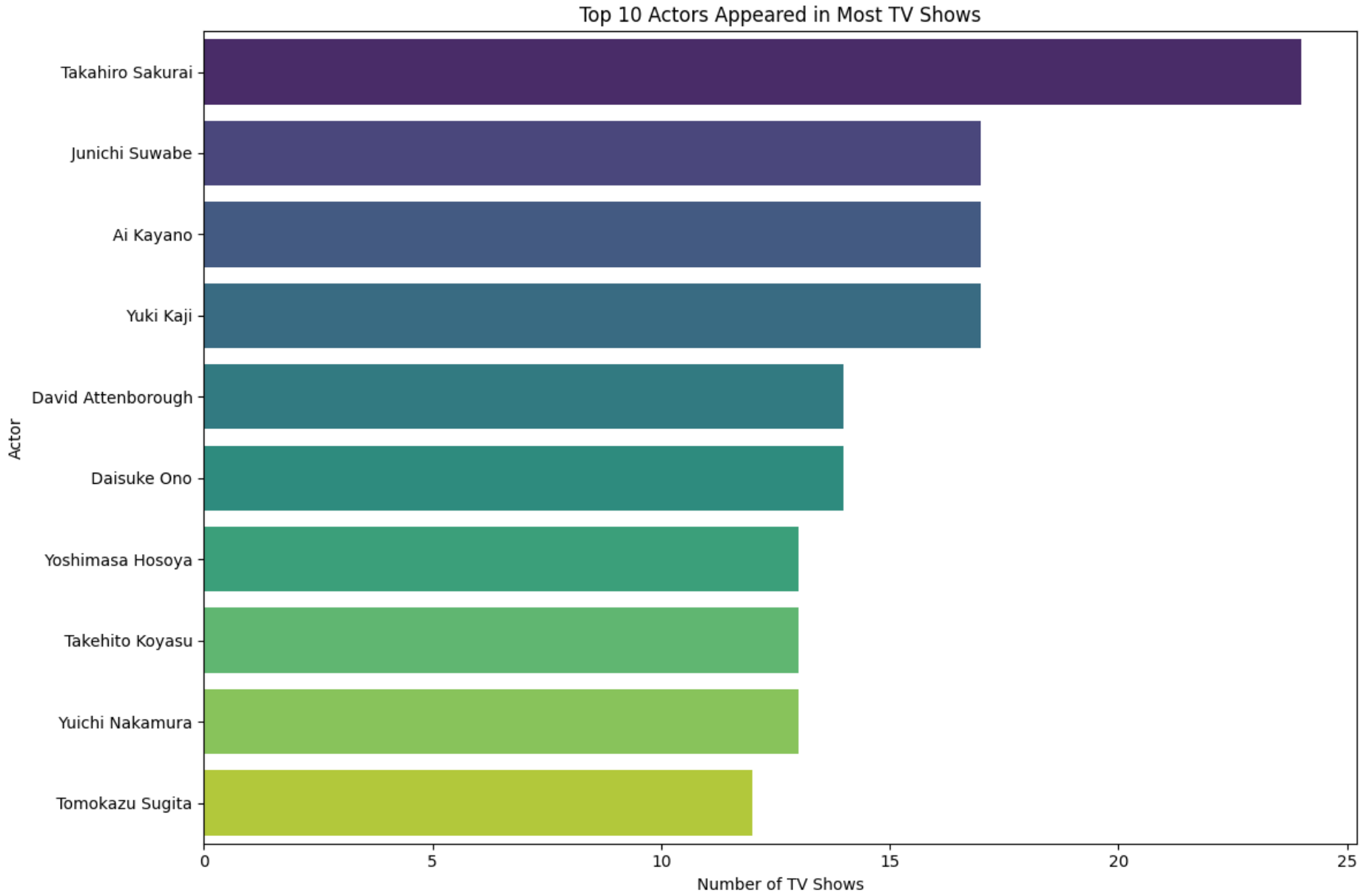
	cast	title
11974	Takahiro Sakurai	24
6136	Junichi Suwabe	17
222	Ai Kayano	17
13204	Yuki Kaji	17
13923	David Attenborough	14
2573	Daisuke Ono	14
13125	Yoshimasa Hosoya	13
11989	Takehito Koyasu	13
13188	Yuichi Nakamura	13
12410	Tomokazu Sugita	12

THE TOP 10 ACTORS APPEARED IN MOST MOVIES :

	cast	title
1946	Anupam Kher	38
16781	Om Puri	27
19235	Rupa Bhimani	27
27291	Shah Rukh Khan	26
17025	Paresh Rawal	25
3109	Boman Irani	25
11219	Julie Tejwani	24
24247	Akshay Kumar	23
18089	Rajesh Kava	21
15977	Naseeruddin Shah	20

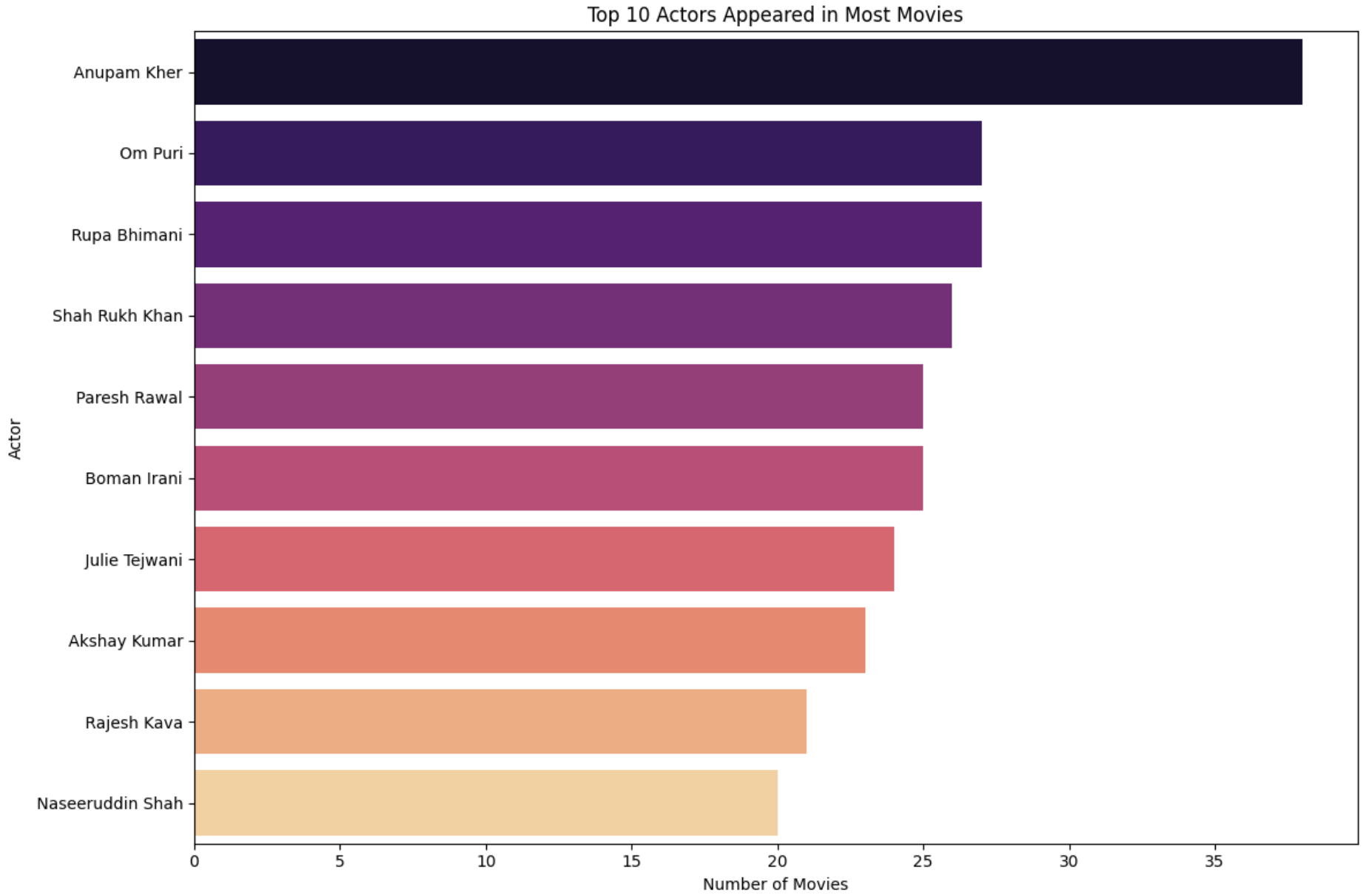
Plotting of above data

In [42]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 8))
sns.barplot(x='title', y='cast', data=top10_tvshow_df, palette='viridis')
plt.xlabel('Number of TV Shows')
plt.ylabel('Actor')
plt.title('Top 10 Actors Appeared in Most TV Shows')

plt.savefig('count_plot.png')

plt.tight_layout()
plt.show()
```



In [43]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Plotting the top 10 actors in movies
```

```
plt.figure(figsize=(12, 8))
sns.barplot(x='title', y='cast', data=top10_movies_df, palette='magma')
plt.xlabel('Number of Movies')
plt.ylabel('Actor')
plt.title('Top 10 Actors Appeared in Most Movies')

plt.savefig('count_plot.png')

plt.tight_layout()
plt.show()
```


Top 10 Actors Appeared in Most Movies
Anupam Kher
Om Puri
Rupa Bhimani
Shah Rukh Khan
Paresh Rawal
Boman Irani
Julie Tejwani
Akshay Kumar
Rajesh Kava
Naseeruddin Shah
Actor
0
5
10
15
20
25
30
35
Number of Movies

Insights:

1. Anupam Kher, Om Puri, Rupa Bhimani, Shah Rukh Khan (SRK), and Paresh Rawal stand out as the top 5 actors with the most substantial presence in Netflix movies, collectively securing the top positions. Notably, these accomplished Indian actors hold the record for the highest number of appearances, solidifying their prominence in the realm of Netflix filmography.

2. Japanese actors such as Takahiro Sakurai, Junichi Suwave, Ai Kayano, Yuki Kaji, and others listed have contributed significantly to TV shows on Netflix, indicating a notable presence of Japanese talent in the platform's content.

Recommendations:

1. **Diverse Indian Talents:** As there is a rich pool of Indian talent contributing significantly to the platform. Netflix may consider further collaborations and productions featuring these actors to cater to the diverse tastes of its global audience.

2. **Japanese Influence in TV Shows:** Netflix could explore expanding its Japanese TV show offerings or developing more original content with Japanese actors to capitalize on this audience interest.

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

INPUT:

In [44]:
```
import pandas as pd
from IPython.display import display

netflix_df = pd.read_csv('netflix.csv')

Top10_tv_show_data = netflix_df[netflix_df['type'] == 'TV Show'].groupby(by = 'director').size().reset_index(name = 't

Top10_tv_show_df = pd.DataFrame(Top10_tv_show_data)

Top10_movies_data = netflix_df[netflix_df['type'] == 'Movie'].groupby(by = 'director').size().reset_index(name = 'tit

Top10_movies_df = pd.DataFrame(Top10_movies_data)

print("THE TOP 10 DIRECTORS APPEARED IN MOST TV shows")
display(Top10_tv_show_df)

print("\nTHE TOP 10 DIRECTORS APPEARED IN MOST MOVIES")
display(Top10_movies_df)
```

```
THE TOP 10 DIRECTORS APPEARED IN MOST TV shows
```

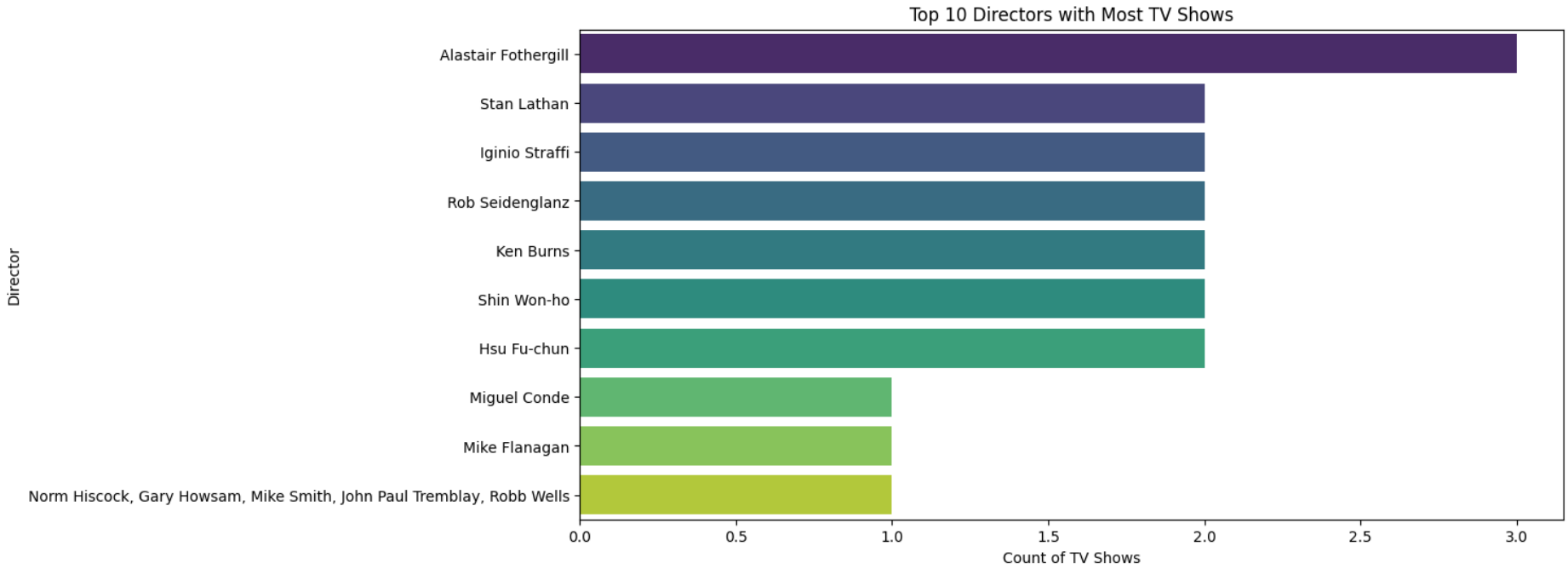
	director	title
5	Alastair Fothergill	3
194	Stan Lathan	2
74	Iginio Straffi	2
177	Rob Seidenglanz	2
108	Ken Burns	2
187	Shin Won-ho	2
72	Hsu Fu-chun	2
144	Miguel Conde	1
145	Mike Flanagan	1
151	Norm Hiscock, Gary Howsam, Mike Smith, John Pa...	1

THE TOP 10 DIRECTORS APPEARED IN MOST MOVIES

	director	title
3252	Rajiv Chilaka	19
3303	Raúl Campos, Jan Suter	18
3885	Suhas Kadav	16
2492	Marcus Raboy	15
1716	Jay Karas	14
661	Cathy Garcia-Molina	13
2562	Martin Scorsese	12
1713	Jay Chapman	12
4306	Youssef Chahine	12
3862	Steven Spielberg	11

In [47]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

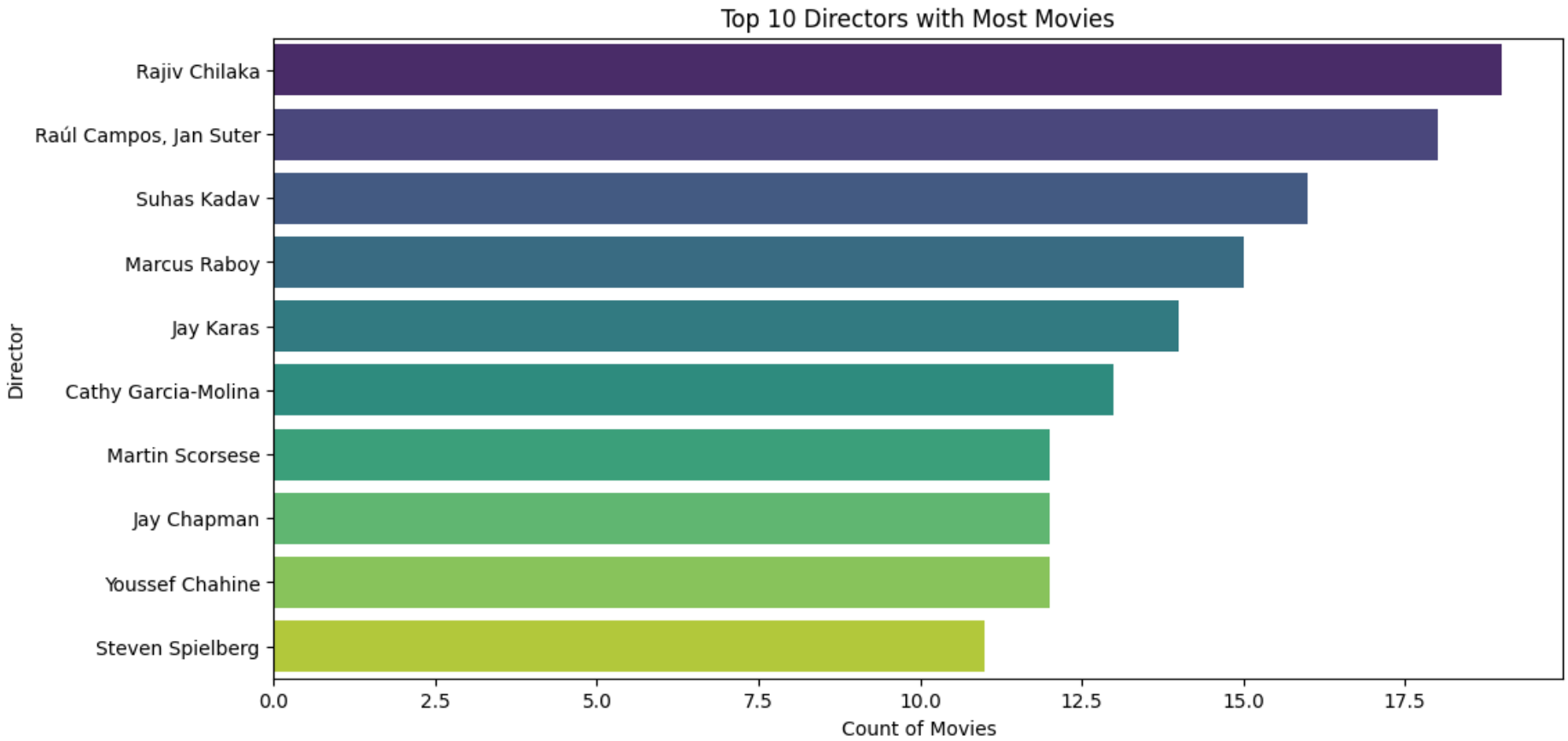
```
from IPython.display import display

# Plot Top TV Show Directors
plt.figure(figsize=(12, 6))
sns.barplot(x='title', y='director', data=Top10_tv_show_data, palette='viridis')
plt.xlabel('Count of TV Shows')
plt.ylabel('Director')

plt.savefig('count_plot.png')

plt.title('Top 10 Directors with Most TV Shows')
plt.show()
```



In [48]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from IPython.display import display

# Plot Top Movie Directors
plt.figure(figsize=(12, 6))
sns.barplot(x='title', y='director', data=Top10_movies_data, palette='viridis')
plt.xlabel('Count of Movies')
```

```
plt.ylabel('Director')
plt.title('Top 10 Directors with Most Movies')

plt.savefig('count_plot.png')

plt.show()
```



Insights:

1. The bar chart representing the Top 10 Directors with the Highest Movie Count reveals that Rajiv Chilaka, Raul Campos, Suhas Kadav, Marcus Raboy and Jay Karas stands out as the top 5 directors with the most prolific contribution to the Netflix Movie library, showcasing a remarkable body of work in terms of directed content.

2. The bar chart representing the Top 10 Directors with the Highest TV show Count reveals that Alastair Fothergill, Stan Lathan, Iginio Straffi, Rob Seidenglanz and Ken Burns stands out as the top 5 directors with the most prolific contribution to the Netflix TV show library, showcasing a remarkable body of work in terms of directed content.

Recommendations:

1. Explore opportunities to showcase diverse and engaging movie content under the direction of these top directors. Promote these directors' works to emphasize the richness and variety they bring to the Netflix movie library.

2. Invest in projects directed by these top TV show directors, recognizing their significant impact on the diversity and quality of Netflix's TV show offerings. Strategically market TV shows directed by these individuals to highlight their substantial contributions to the Netflix TV show library.

5. Which genre movies are more popular or produced more.

INPUT:

In [49]:
```
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt

netflix_df = pd.read_csv('netflix.csv')

netflix_df['listed_in'] = netflix_df['listed_in'].str.split(',')


netflix_df = netflix_df.explode('listed_in', ignore_index=True)


join_text = " ".join(netflix_df['listed_in'])

wordcloud = WordCloud(width=2000, height=1000, background_color='black',  min_font_size = 10).generate(join_text)


plt.figure(figsize=(15, 13))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
```

```
plt.savefig('wordcloud.png')
plt.show()
```

Insights:

Analyzing the word cloud for genres, prominent terms such as "International Movies," "TV/Movies Comedies," "Dramas," "Action," and "Romantic" emerge frequently. This suggests a rich diversity of content, encompassing romantic, action, comedy, and drama genres within Netflix's extensive library.

6. Find After how many days the movie will be added to Netflix after the release of the movie (you can consider the recent past data)

Difference in Days

INPUT:

In [ ]:
```
import pandas as pd
import numpy as np

netflix_df = pd.read_csv('netflix.csv')

netflix_df['date_added'] = netflix_df['date_added'].fillna(0)

netflix_df['date_added_year'] = pd.to_datetime(netflix_df['date_added'], errors='coerce')

netflix_df['release_year'] = pd.to_datetime(netflix_df['release_year'],format='%Y')



netflix_df['days_to_add'] = netflix_df['date_added_year'] - netflix_df['release_year']
display(netflix_df.iloc[:,[12,7,13]].sort_values('days_to_add', ascending = False).reset_index())



mode_days_to_add = netflix_df['days_to_add'].mode()
print('\nMode of difference in days : \n',mode_days_to_add)
```

	index	date_added_year	release_year	days_to_add
0	4250	2018-12-30	1925-01-01	34331 days
1	1331	2021-02-09	1945-01-01	27798 days
2	7790	2017-03-31	1942-01-01	27483 days
3	8205	2017-03-31	1942-01-01	27483 days
4	8739	2017-03-31	1943-01-01	27118 days
...	...	...	...	...
8802	6066	1970-01-01	2013-01-01	-15706 days
8803	7847	1970-01-01	2015-01-01	-16436 days
8804	8182	1970-01-01	2015-01-01	-16436 days
8805	7406	1970-01-01	2016-01-01	-16801 days
8806	6174	1970-01-01	2018-01-01	-17532 days

8807 rows × 4 columns

```
Mode of difference in days :
 0   334 days
Name: days_to_add, dtype: timedelta64[ns]
```

Insights:

After a movie or TV show is initially released, it typically takes approximately 334 days for it to be added to the Netflix platform. This duration represents the time lapse between the original release date and the date when the content becomes available for streaming on Netflix. The 334-day period is an average or typical timeframe, suggesting that there is a consistent pattern in the delay between the initial release and the content's availability on the Netflix platform. This information provides insight into the distribution and timing of when new content becomes accessible to Netflix subscribers following its original release.

Difference in years

In [ ]:
```
import pandas as pd
import numpy as np
```

```
netflix_df = pd.read_csv('netflix.csv')

netflix_df['date_added_year'] = pd.to_datetime(netflix_df['date_added']).dt.year

netflix_df['date_added_year'] = netflix_df['date_added_year'].replace(np.nan, 0).astype(int)

netflix_df['years_to_add'] = netflix_df['date_added_year'] - netflix_df['release_year']
display(netflix_df.iloc[:,[7,12,13]].sort_values('years_to_add', ascending = False))

mode_years_to_add = netflix_df['years_to_add'].mode()[0]
print('\nMode of difference in years : ',mode_years_to_add)
```

	release_year	date_added_year	years_to_add
4250	1925	2018	93
1331	1945	2021	76
8205	1942	2017	75
7790	1942	2017	75
8660	1943	2017	74
...	...	...	...
6066	2013	0	-2013
7847	2015	0	-2015
8182	2015	0	-2015
7406	2016	0	-2016
6174	2018	0	-2018

8807 rows × 3 columns

```
Mode of difference in years :  0
```

Insights:

"Upon analyzing the data, it becomes evident that a significant portion of the movies is added to Netflix within approximately one year of their release. This finding aligns with our earlier investigation into the difference in days between the date a movie is added to Netflix and its

original release date. The majority of movies exhibit a pattern of being made available on the streaming platform within a relatively short time frame of around one year, reflecting a consistent trend in the Netflix content acquisition process."

In [50]:
```
!jupyter nbconvert --to html /content/Netflix_Case_Studyy.ipynb
```

```
[NbConvertApp] Converting notebook /content/Netflix_Case_Studyy.ipynb to html
[NbConvertApp] Writing 2067076 bytes to /content/Netflix_Case_Studyy.html
```