NETFLIX

Business Problem:

Analyse 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

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
stream="/content/drive/MyDrive/netflix.csv"
```

Reading Netflix file by using read_csv function of pandas and creating dataframe named as netflix

In [3]:
```
netflix=pd.read_csv(stream)
```

In [4]:
```
netflix
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Dc
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	T
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	T
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	I S
...	...	...	...	...	...	...	...	...	...	...	
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 min	
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	2 Seasons	
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 min	H
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 min	Fa
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14	111 min	M

8807 rows × 12 columns

Data Cleaning

In [6]:
```
data=netflix.copy()

#Here I created duplicate copy of original dataset
```

In [7]:
```
data.isnull().any()

#Found missing Values
```

Out[7]:
```
show_id         False
type            False
title           False
director         True
cast             True
country          True
date_added       True
release_year    False
rating           True
duration         True
listed_in       False
description     False
dtype: bool
```

In [8]:
```
data.isnull().sum()
```

Out[8]:
```
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 [9]:
```
data.dropna(subset=['rating', 'duration','date_added'],inplace=True)
```

Columns namely as date_added,rating,Duration which have very small number of missing values are 10,4,& 3 respectively,so we can directly drop it

In [10]:
```
data['director'].fillna("UNKNOWN",inplace=True)
data['cast'].fillna("UNKNOWN",inplace=True)
data['country'].fillna("UNKNOWN",inplace=True)
```

columns named as Director,cast,Country which have very large number of missing values 2634,825,831 respectively ,we can't drop it directly because it may cause loss of data so we can fill this value with unknow entry

In [11]:
```
data.isnull().any()
#finnaly no missing value found
```

Out[11]:
```
show_id         False
type            False
title           False
director        False
cast            False
country         False
date_added      False
release_year    False
rating          False
duration        False
listed_in       False
description     False
dtype: bool
```

Exploratory the data analysis

In [ ]:
```
netflix.shape
```

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

overall there are 8807 rows & 12 columns in the dataset

In [ ]:
```
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
```

It shows us total detail information i.e;total no of columns & rows ,names of the columns, their different datatypes present in the dataframe.it also shows number of nonenullvalues in each columns

In [ ]:
```
netflix.head()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	lis
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Docume
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	Intern TV Sho Dram My
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Cri S Intern TV Sho
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docu Rea
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	Intern TV S Roma Shows

It gives by default first five rows of the dataset

In [ ]:
```
netflix.tail()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 min	Cu
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	2 Seasons	K S
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 min	C
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 min	C
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14	111 min	Int

It gives by default last five rows of the dataset

In [ ]:
```
netflix.describe()
#it will perform detail aggregation operations
```

Out[ ]:

	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

describe()method shows the statistical summary of whole dataframe

In [ ]:
```
netflix.describe(include=object)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	rating	duration	listed_in	des
count	8807	8807	8807	6173	7982	7976	8797	8803	8804	8807	
unique	8807	2	8807	4528	7692	748	1767	17	220	514	
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	TV-MA	1 Season	Dramas, International Movies	Pa ac ab
freq	1	6131	1	19	19	2818	109	3207	1793	362	

It provide basic summary statistics for the object columns in your DataFrame, such as count, unique values, top value, and frequency of the top value.

In [ ]:
```
np.any(netflix.duplicated())
#checking duplicate values
```

Out[ ]:
```
False
```

it states that there is no duplicate values present in netflix dataset

In [ ]:
```
df=netflix.copy()
```

In [ ]:
```
df.head()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	lis
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Docume
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	Intern TV Sho Dram My
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Cri S Intern TV Sho
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docu Rea
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	Intern TV S Roma Shows

In [ ]:
```
df["date_added"]=pd.to_datetime(df.date_added)
df["year"]=df.date_added.dt.year
```

In [ ]:
```
df.head(2)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	liste
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Documenta
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Internati TV Shows Dramas Myste

Here we changed the input data(date_added) into datetime format objects and we created new column-year

In [ ]:
```
df.type=df.type.astype("category")
```

In [ ]:
```
df.rating=df.rating.astype("category")
```

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       8807 non-null   object        
 1   type          8807 non-null   category      
 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   datetime64[ns]
 7   release_year  8807 non-null   int64         
 8   rating        8803 non-null   category      
 9   duration      8804 non-null   object        
 10  listed_in     8807 non-null   object        
 11  description   8807 non-null   object        
 12  year          8797 non-null   float64       
dtypes: category(2), datetime64[ns](1), float64(1), int64(1), object(8)
memory usage: 775.0+ KB
```

changing data type of "type" and "rating" columns into category

Earliest released Movies

In [ ]:
```
df["release_year"].min()
```

Out[ ]:
```
1925
```

In [ ]:
```
df.query("release_year==1925")[["title","release_year","type"]]
```

Out[ ]:

	title	release_year	type
4250	Pioneers: First Women Filmmakers*	1925	TV Show

earliest tvshow is released in year-1925

In [ ]:
```
pd.to_datetime(df.date_added).dt.year.min()
```

Out[ ]:
```
2008.0
```

In [ ]:
```
df.query("year==2008")[["title","year","type"]]
```

Out[ ]:

	title	year	type
5957	To and From New York	2008.0	Movie
6611	Dinner for Five	2008.0	TV Show

earliest movie is released in the year -2008

value_counts for few columns

In [ ]:
```
df["type"].value_counts()
```

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

In [ ]:
```
df["title"].value_counts()
```

Out[ ]:
```
Dick Johnson Is Dead                     1
Ip Man 2                                 1
Hannibal Buress: Comedy Camisado         1
Turbo FAST                               1
Masha's Tales                            1
                                        ..
Love for Sale 2                          1
ROAD TO ROMA                             1
Good Time                                1
Captain Underpants Epic Choice-o-Rama    1
Zubaan                                   1
Name: title, Length: 8807, dtype: int64
```

In [ ]:
```
pd.DataFrame(df["director"].value_counts())
```

Out[ ]:

	director
Rajiv Chilaka	19
Raúl Campos, Jan Suter	18
Marcus Raboy	16
Suhas Kadav	16
Jay Karas	14
...	...
Raymie Muzquiz, Stu Livingston	1
Joe Menendez	1
Eric Bross	1
Will Eisenberg	1
Mozez Singh	1

4528 rows × 1 columns

In [ ]:
```
pd.DataFrame(df["country"].value_counts())
```

Out[ ]:

	country
United States	2818
India	972
United Kingdom	419
Japan	245
South Korea	199
...	...
Romania, Bulgaria, Hungary	1
Uruguay, Guatemala	1
France, Senegal, Belgium	1
Mexico, United States, Spain, Colombia	1
United Arab Emirates, Jordan	1

748 rows × 1 columns

In [ ]:
```
pd.DataFrame(df["listed_in"].value_counts())
```

Out[ ]:

	listed_in
Dramas, International Movies	362
Documentaries	359
Stand-Up Comedy	334
Comedies, Dramas, International Movies	274
Dramas, Independent Movies, International Movies	252
...	...
Kids' TV, TV Action & Adventure, TV Dramas	1
TV Comedies, TV Dramas, TV Horror	1
Children & Family Movies, Comedies, LGBTQ Movies	1
Kids' TV, Spanish-Language TV Shows, Teen TV Shows	1
Cult Movies, Dramas, Thrillers	1

514 rows × 1 columns

In [ ]:
```
df.nunique()
#it represents no of unique values
```

Out[ ]:
```
show_id         8807
type               2
title           8807
director        4528
cast            7692
country          748
date_added      1714
release_year      74
rating            17
duration         220
listed_in        514
description     8775
year              14
dtype: int64
```

Number of unique values present in each columns of the dataframe

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

Out[ ]:
```
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
year              10
dtype: int64
```

Number of Null Values in each columns

In [ ]:
```
null_values=100*(df.isna().sum())/len(df.index)
d.DataFrame(null_values).reset_index()
```

```
null_values.columns=["Column's Name","Null Percentage"]
```

In [ ]:
```
null_values
```

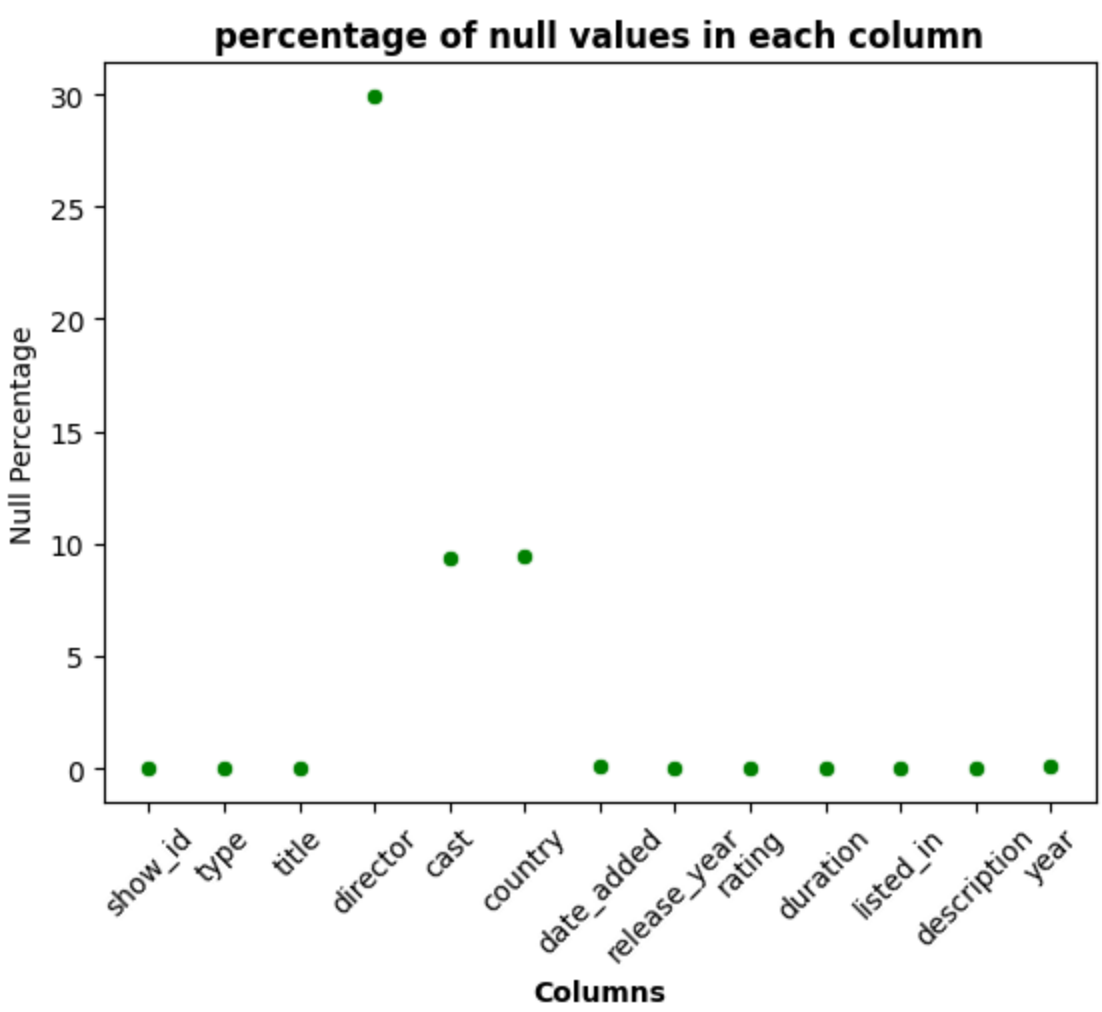
Out[ ]:

	Column's Name	Null Percentage
0	show_id	0.000000
1	type	0.000000
2	title	0.000000
3	director	29.908028
4	cast	9.367549
5	country	9.435676
6	date_added	0.113546
7	release_year	0.000000
8	rating	0.045418
9	duration	0.034064
10	listed_in	0.000000
11	description	0.000000
12	year	0.113546

perecntage of null values in each columns and the above values shows that nearly 30% of data in directors column is missing

Scatterplot showing percentage of Null values in each Columns

In [ ]:
```
sns.scatterplot(x="Column's Name",y="Null Percentage",data=null_values,color="green")
plt.xticks(rotation=45,fontsize=10),
plt.title("percentage of null values in each column",weight="bold"),
plt.xlabel("Columns",fontsize=10,weight="bold")
plt.show()
#scatter plot
```



the percentage of missing values is very high in director column and deleting this would make loss of data

Analysis on Release year and Types

In [ ]:
```
df.query("year.isna()")
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	
6066	s6067	TV Show	A Young Doctor's Notebook and Other Stories	NaN	Daniel Radcliffe, Jon Hamm, Adam Godley, Chris...	United Kingdom	NaT	2013	TV-MA	2 Seasons	B S C TV
6174	s6175	TV Show	Anthony Bourdain: Parts Unknown	NaN	Anthony Bourdain	United States	NaT	2018	TV-PG	5 Seasons	Do
6795	s6796	TV Show	Frasier	NaN	Kelsey Grammer, Jane Leeves, David Hyde Pierce...	United States	NaT	2003	TV-PG	11 Seasons	Cu C
6806	s6807	TV Show	Friends	NaN	Jennifer Aniston, Courteney Cox, Lisa Kudrow, ...	United States	NaT	2003	TV-14	10 Seasons	Cu C
6901	s6902	TV Show	Gunslinger Girl	NaN	Yuuka Nanri, Kanako Mitsuhashi, Eri Sendai, Am...	Japan	NaT	2008	TV-14	2 Seasons	
7196	s7197	TV Show	Kikoriki	NaN	Igor Dmitriev	NaN	NaT	2010	TV-Y	2 Seasons	
7254	s7255	TV Show	La Familia P. Luche	NaN	Eugenio Derbez, Consuelo Duval, Luis Manuel Áv...	United States	NaT	2012	TV-14	3 Seasons	Inte TV L
7406	s7407	TV Show	Maron	NaN	Marc Maron, Judd Hirsch, Josh Brener, Nora Zeh...	United States	NaT	2016	TV-MA	4 Seasons	C
7847	s7848	TV Show	Red vs. Blue	NaN	Burnie Burns, Jason Saldaña, Gustavo Sorola, G...	United States	NaT	2015	NR	13 Seasons	TV A C TV
8182	s8183	TV Show	The Adventures of Figaro Pho	NaN	Luke Jurevicius, Craig Behenna, Charlotte Haml...	Australia	NaT	2015	TV-Y7	2 Seasons	Kid C

Missing values in date_added column and year column where data correspoding to release year 2003, 2008, 2010, 2012, 2013, 2015, 2016, 2018

In [ ]:
```
Tempy=df[["release_year","year","type"]]
Tempy["diff"]=Tempy.loc[:,"year"]-Tempy.loc[:,"release_year"]
```

```
<ipython-input-34-f2549cb0f08c>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  Tempy["diff"]=Tempy.loc[:,"year"]-Tempy.loc[:,"release_year"]
```

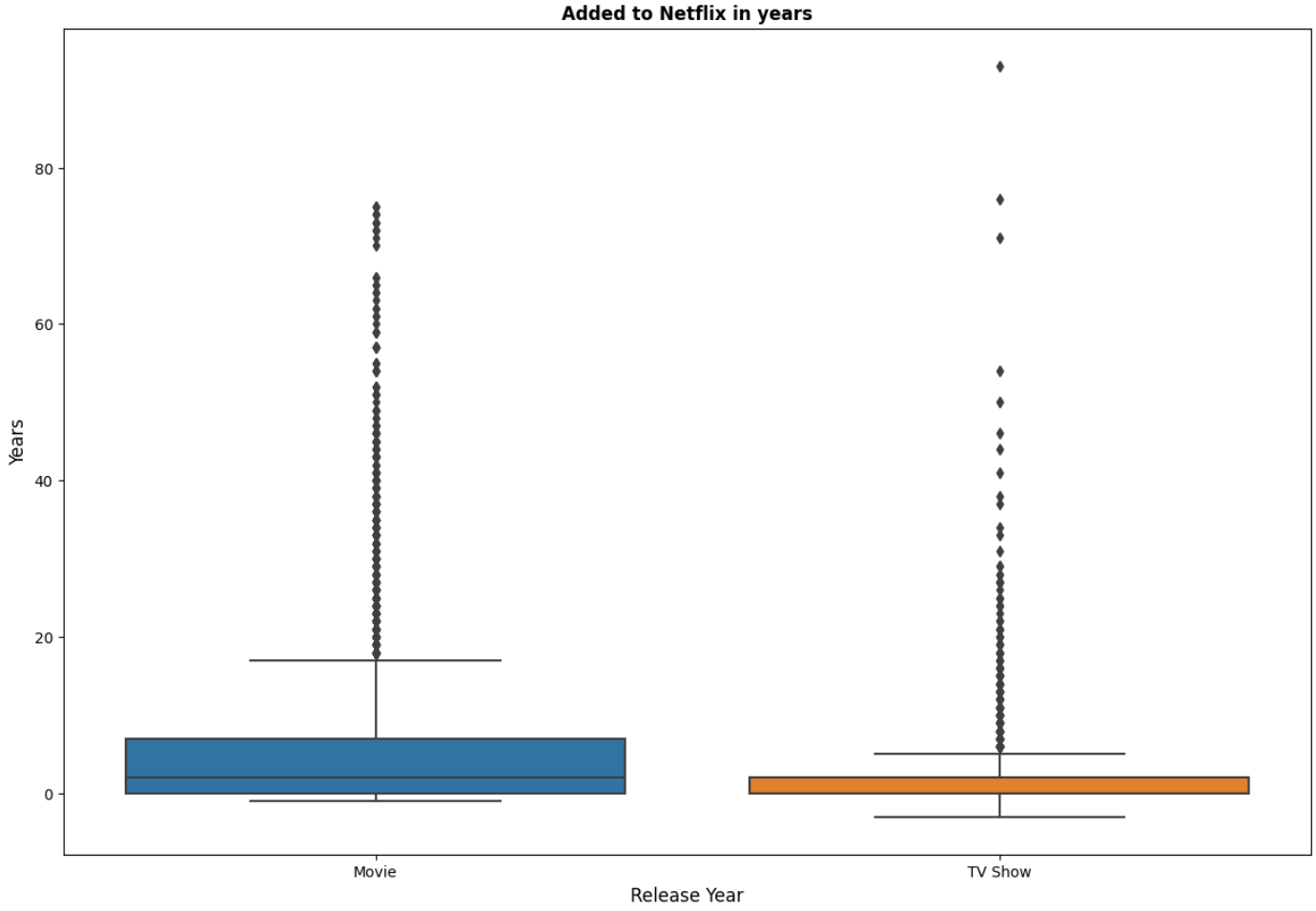
In [ ]:
```
Tempy
```

Out[ ]:

	release_year	year	type	diff
0	2020	2021.0	Movie	1.0
1	2021	2021.0	TV Show	0.0
2	2021	2021.0	TV Show	0.0
3	2021	2021.0	TV Show	0.0
4	2021	2021.0	TV Show	0.0
...	...	...	...	...
8802	2007	2019.0	Movie	12.0
8803	2018	2019.0	TV Show	1.0
8804	2009	2019.0	Movie	10.0
8805	2006	2020.0	Movie	14.0
8806	2015	2019.0	Movie	4.0

8807 rows × 4 columns

In [ ]:
```
fig=plt.figure(figsize=(15,10))
sns.boxplot(x="type",y="diff",data=Tempy)
plt.xlabel("Release Year",fontsize=12)
plt.ylabel("Years",fontsize=12)
plt.title("Added to Netflix in years",fontsize=12,weight="bold")
plt.show()
#boxplot
```



-from Above;Nearly 50% of movies added to to platform within 1 to 3 Years of release, Whereas Nearly 75% of Movies added to Netflix before 10 years.

-In case of TV Shows, Nearly 75% of shows are added before 3 to 4 years.

In [ ]:
```
df_copie=df.copy()
```

Movies and TV shows released per year

In [ ]:
```
df_copie.head(2)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	liste
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Documenta
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Internati TV Shows Dramas Myste

percentage of Movies & TVshow

In [ ]:
```
#Total numbers of movies and Tv shows
Total=len(df_copie.index)
```

In [ ]:
```
#Total number of movies
No_of_movie=len(df.query("type=='Movie'").index)
```

In [ ]:
```
#Total number of TV Shows
No_of_shows=len(df.query("type=='TV Show'").index)
```

In [ ]:
```
#Movie percentage
movie_per=100*No_of_movie/Total
f"{round(movie_per,2)}%"
```

Out[ ]:
```
'69.62%'
```

total movies percentage-69.62%

In [ ]:
```
#TV show Percentage
show_per=100*No_of_shows/Total
f"{round(show_per,2)}%"
```

Out[ ]:
```
'30.38%'
```

total Tv Shows percentage-30.38%

Percentage of increase of Movies & Tv Show Released from 1990 to 2021

In [ ]:
```
#percentage increase of Movies
movie_90=df_copie.query("type=='Movie'").query("release_year==1990")
total_90=len(movie_90)
total_90
```

Out[ ]:
```
19
```

In [ ]:
```
movie_21=df_copie.query("type=='Movie'").query("release_year==2021")
total_21=len(movie_21)
total_21
```

Out[ ]:
```
277
```

In [ ]:
```
#percentage increment
per_movie=100*(total_21-total_90)/total_90
round(per_movie,2)
```

Out[ ]:
```
1357.89
```

In [ ]:
```
show_90=df_copie.query("type=='TV Show'").query("release_year==1990")
Total_90=len(show_90)
Total_90
```

Out[ ]:
```
3
```

In [ ]:
```
show_21 = df_copie.query("type=='TV Show'").query('release_year==2021')
Total_21=len(show_21)
```

Out[ ]:
```
315
```

In [ ]:
```
#percentage increase of tv shows
per_show=100*(Total_21-Total_90)/Total_90
round(per_show,2)
```

Out[ ]:
```
10400.0
```

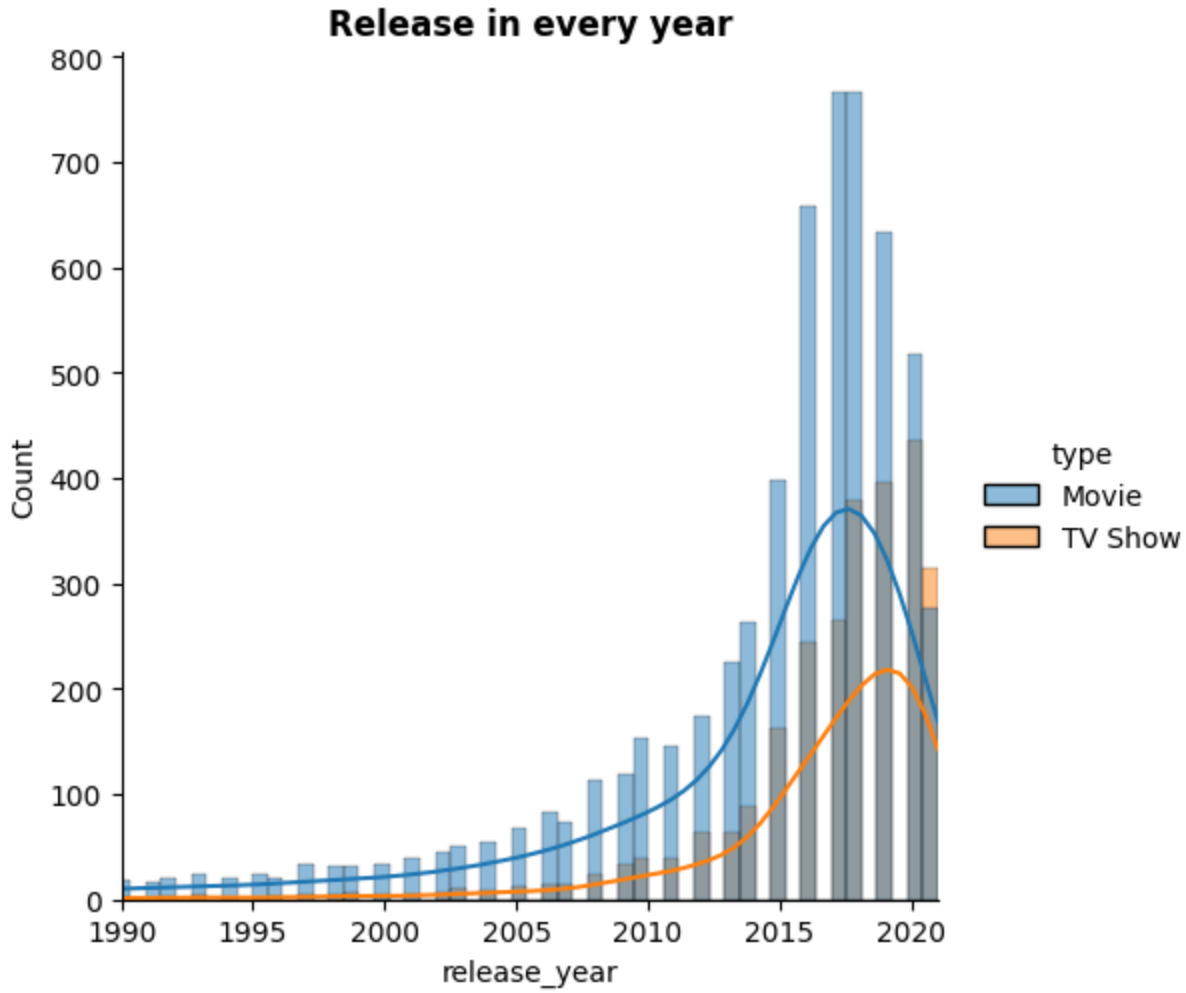
Perecentage wise increase of Movies->1357.89%

percentage wise increase of TV Shows->10400.0%

In [ ]:
```
year = df_copie.release_year.value_counts().index
val =df_copie.release_year.value_counts().values
```

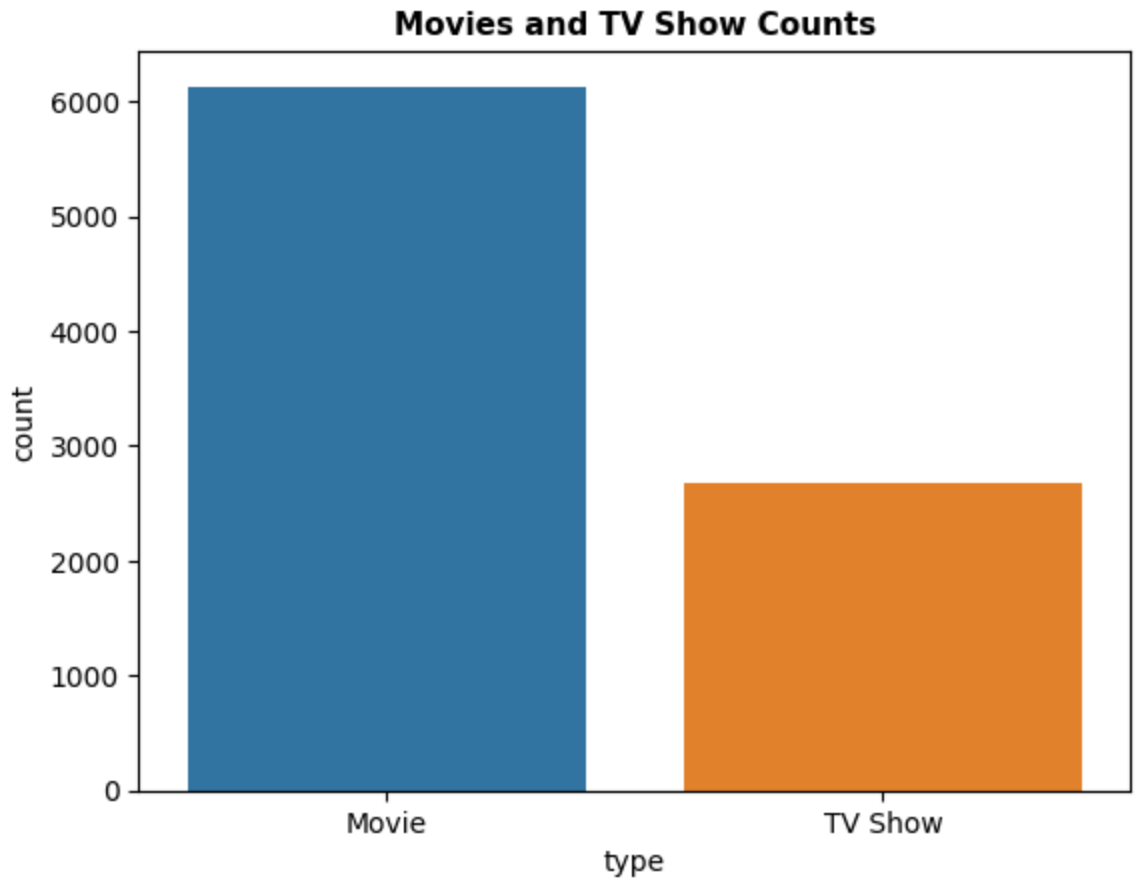
Univariate analysis using bar, countplot, displot and kdeplot between shows vs movies

In [ ]:
```
sns.displot(df_copie,x="release_year",hue="type",kde=True)
plt.xlim(1990,2021)
plt.title("Release in every year",fontsize=12,weight="bold")
plt.show()
#displot
```



In [ ]:
```
sns.countplot(x="type",data=df_copie)
plt.title("Movies and TV Show Counts",fontsize=11,weight="bold")
```

Out[ ]: Text(0.5, 1.0, 'Movies and TV Show Counts')

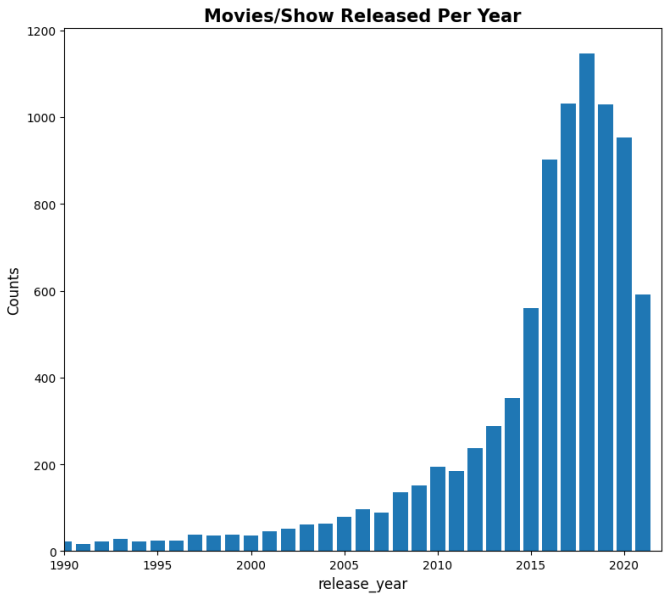


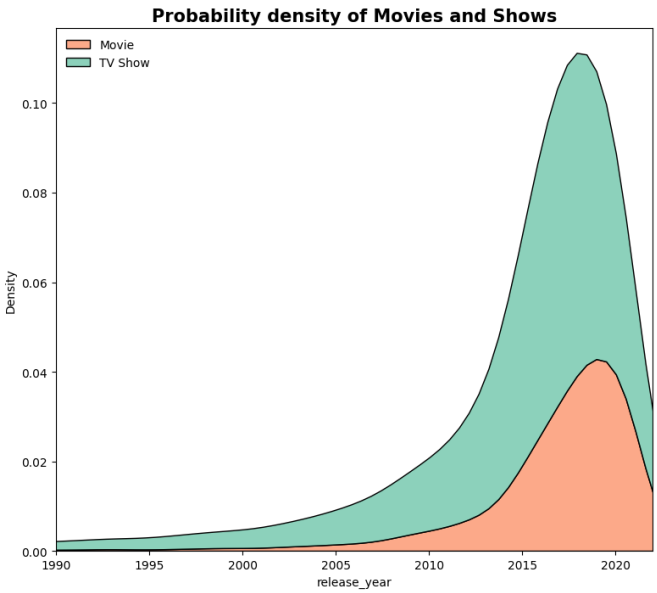
In [ ]:
```
plt.figure(figsize=(20,8))
plt.subplot(1,2,1)

plt.bar(year,val)
plt.xlim(1990,2022)
plt.xlabel('release_year',fontsize=12)
plt.ylabel('Counts',fontsize=12)
plt.title('Movies/Show Released Per Year',fontsize=15,weight='bold')
#barplot

plt.subplot(1,2,2)
sns.kdeplot(x='release_year',hue='type',data=df_copie,palette='Set2',multiple='stack')
plt.xlim(1990,2022)
plt.title('Probability density of Movies and Shows',fontsize=15,weight='bold')
plt.legend(['Movie','TV Show'],loc='upper left',frameon=False)
#Kde plot

plt.show()
```





insights

-Movies consitutes Major part of release i.e;nearly 70 percentage whereas TV Show constitue lower part of release i.e; 30 percentage

-although percentage increase of TV Shows is much higher than movies in past 30 years

Best Time To Launch a TV Show

In [ ]:
```
df_show = df_copie.query("type == 'TV Show'")
df_show.head(2)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Internatic TV Sho TV Dram Myste
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	Crime Sho Internatic TV Sho TV A

In [ ]:
```
df_show['Month'] = df_show['date_added'].dt.month_name()
df_show['Day'] = df_show['date_added'].dt.day_name()
```

```
<ipython-input-55-ae4be26065a5>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df_show['Month'] = df_show['date_added'].dt.month_name()
<ipython-input-55-ae4be26065a5>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df_show['Day'] = df_show['date_added'].dt.day_name()
```

In [ ]:
```
df_show.head(2)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Internatic TV Sho TV Dram Myste
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	Crime Sho Internatic TV Sho TV A

In [ ]:
```
df_m = df_show.groupby('Month')[['type']].count().reset_index()
df_m = df_m.rename(columns={'type':'counts'})
```

In [ ]:
```
df_m["Month"]=df_m["Month"].astype("category")
```

In [ ]:
```
df_m["Month"]=df_m["Month"].cat.set_categories(["January","February","March","April","Ma
```

In [ ]:
```
df_m.sort_values("counts",ascending=False)
df_m
```

Out[ ]:

	Month	counts
0	April	214
1	August	236
2	NaN	266
3	February	181
4	January	192
5	July	262
6	June	236
7	March	213
8	May	193
9	NaN	207
10	NaN	215
11	September	251

In [ ]:
```
df_d=df_show.groupby("Day")[["type"]].count().reset_index()
df_d=df_d.rename(columns={"type":"counts"})
```

In [ ]:
```
df_d['Day'] = df_d['Day'].astype('category')
```

In [ ]:
```
df_d['Day'] = df_d['Day'].cat.set_categories(['Sunday','Monday','Tuesday','Wednesday','T
```

In [ ]:
```
df_d.sort_values('counts',ascending=False)
```

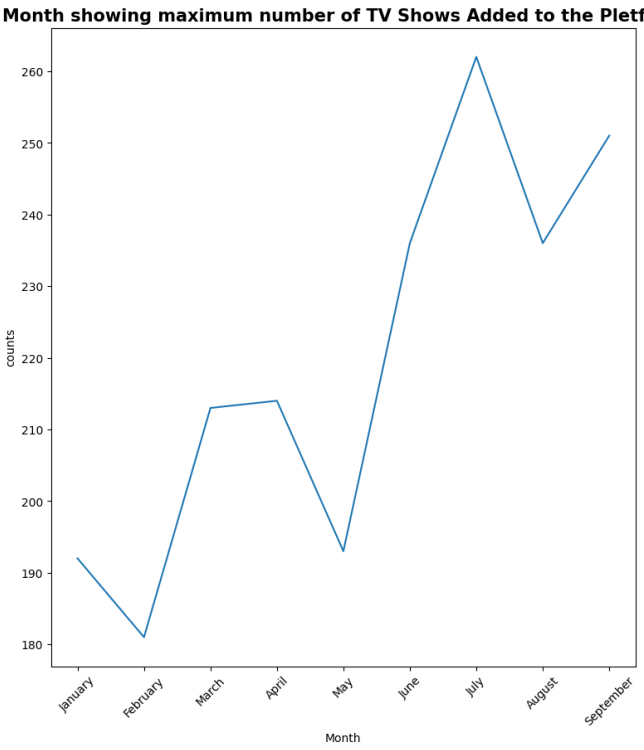
Out[ ]:

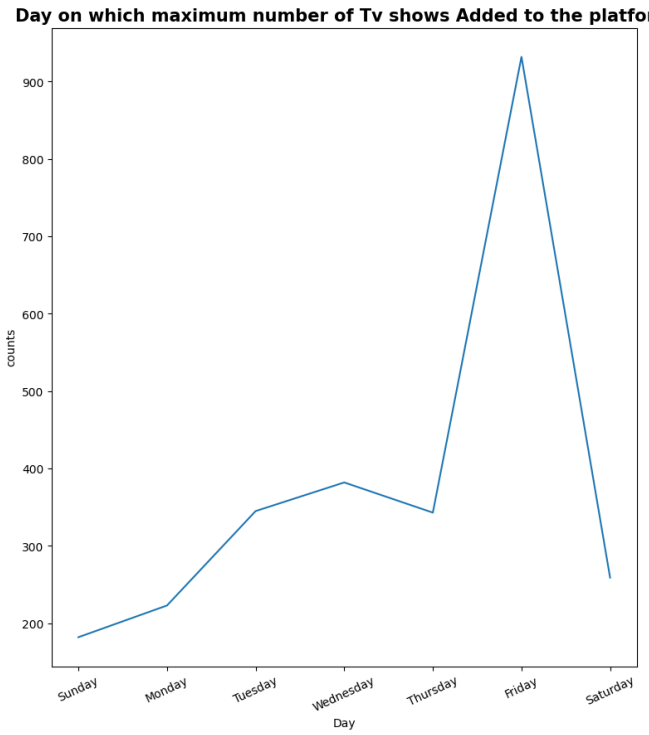
	Day	counts
0	Friday	932
6	Wednesday	382
5	Tuesday	345
4	Thursday	343
2	Saturday	259
1	Monday	223
3	Sunday	182

In [ ]:
```
plt.figure(figsize=(20,10))

plt.subplot(1,2,1)
sns.lineplot(x="Month",y="counts",data=df_m)
plt.xticks(rotation=45)
plt.title("Month showing maximum number of TV Shows Added to the Pletform",fontsize=15,w

plt.subplot(1,2,2)
sns.lineplot(x="Day",y="counts",data=df_d)
plt.xticks(rotation=25)
plt.title("Day on which maximum number of Tv shows Added to the platform",fontsize=15,we
plt.show()
```





Insights

Important month --> December

-This could be due to new year, or Christmas celebration as due to holiday users get enough time on these occassions. So this this month is very crucial to launch any show

Important day --> Friday

-This is important, as on weekend generally users have enough time to spent on watching their favorite show.

Analysis of Actors/Directors and Types of shows/Movies

In [ ]:
```
dir=df.copy()
```

In [ ]:
```
dir["director"]=dir["director"].str.split(",")
```

In [ ]:
```
drc=dir.explode("director")
```

Top 10 Directors who made highest movies or TV Shows

In [ ]:
```
drc.head()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	lis
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Docume
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Intern TV Sho Dram My
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	Cri S Intern TV Sho
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	1 Season	Docu Rea
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2 Seasons	Intern TV S Roma Shows

In [ ]:
```
Name=drc["director"].value_counts().index
Total=drc["director"].value_counts().values
```

In [ ]:
```
Top10=Name[:10]
Top10
```

Out[ ]:
```
Index(['Rajiv Chilaka', 'Raúl Campos', ' Jan Suter', 'Marcus Raboy',
       'Suhas Kadav', 'Jay Karas', 'Cathy Garcia-Molina', 'Martin Scorsese',
       'Jay Chapman', 'Youssef Chahine'],
      dtype='object')
```

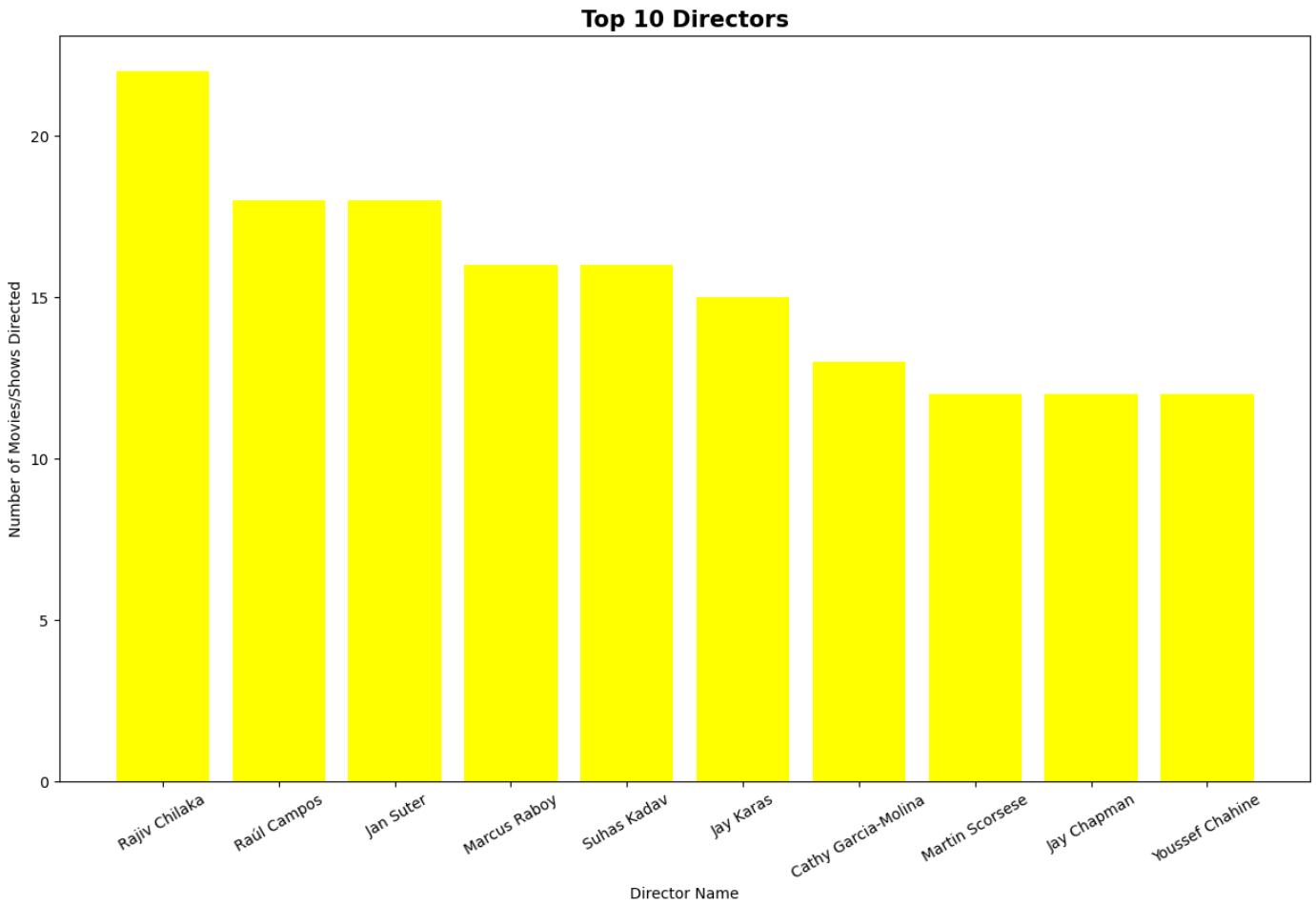
->Above names are Top 10 directors

In [ ]:
```
Total=Total[:10]
```

In [ ]:
```
plt.figure(figsize=(15,9))

plt.bar(Top10,Total,color="yellow")
plt.title('Top 10 Directors',fontsize=15,weight='bold')
plt.xticks(rotation = 30)
plt.xlabel('Director Name',fontsize=10)
plt.ylabel('Number of Movies/Shows Directed',fontsize=10)


plt.show()
```



->Rajiv Chilaka is the no:1 & famous director

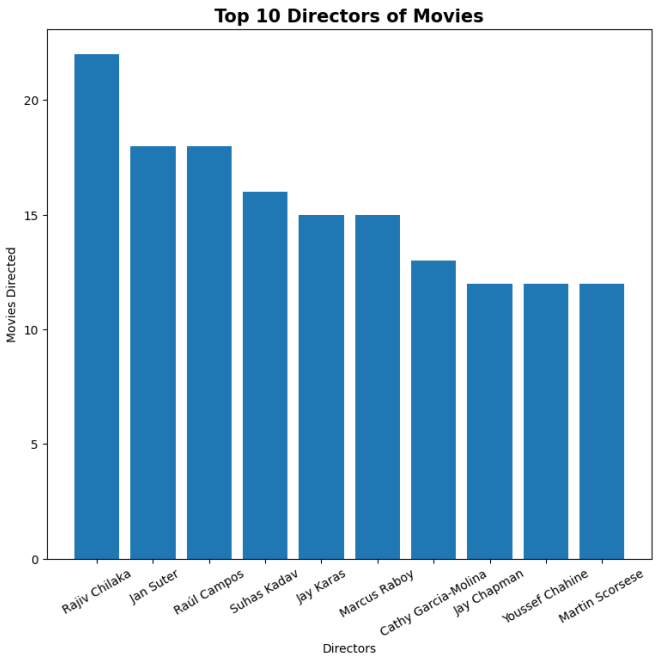
In [ ]:
```
#Movies Directed
dm=drc.query("type=='Movie'")["director"].value_counts()
Top_10_Movies=dm.index[:10]
val_m=dm.values[:10]
```

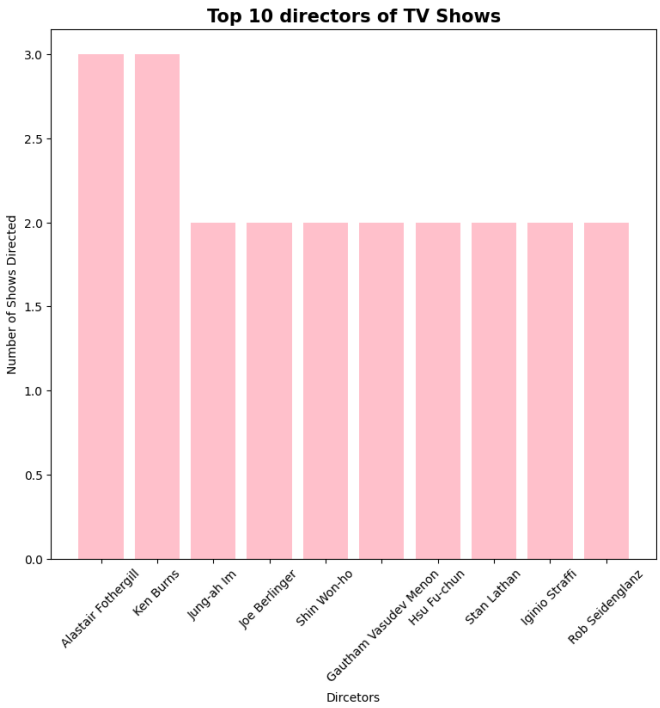
In [ ]:
```
#shows dircted
ds=drc.query("type=='TV Show'")["director"].value_counts()
Top=ds.index[:10]
val_s=ds.values[:10]
```

In [ ]:
```
plt.figure(figsize=(20,8))

#movies
plt.subplot(1,2,1)
plt.bar(Top_10_Movies,val_m)
plt.title("Top 10 Directors of Movies",fontsize=15,weight="bold")
plt.xticks(rotation=30)
plt.xlabel("Directors",fontsize=10)
plt.ylabel("Movies Directed",fontsize=10)

#shows
plt.subplot(1,2,2)
plt.bar(Top,val_s,color="pink")
plt.title("Top 10 directors of TV Shows",fontsize=15,weight="bold")
plt.xticks(rotation=45)
plt.xlabel("Dircetors",fontsize=10)
plt.ylabel("Number of Shows Directed",fontsize=10)
plt.show()
```





In [ ]:
```
drc['Directors'] = drc.groupby('show_id')[['director']].transform(lambda x:x.count())
```

In [ ]:
```
drc.head(4)
```

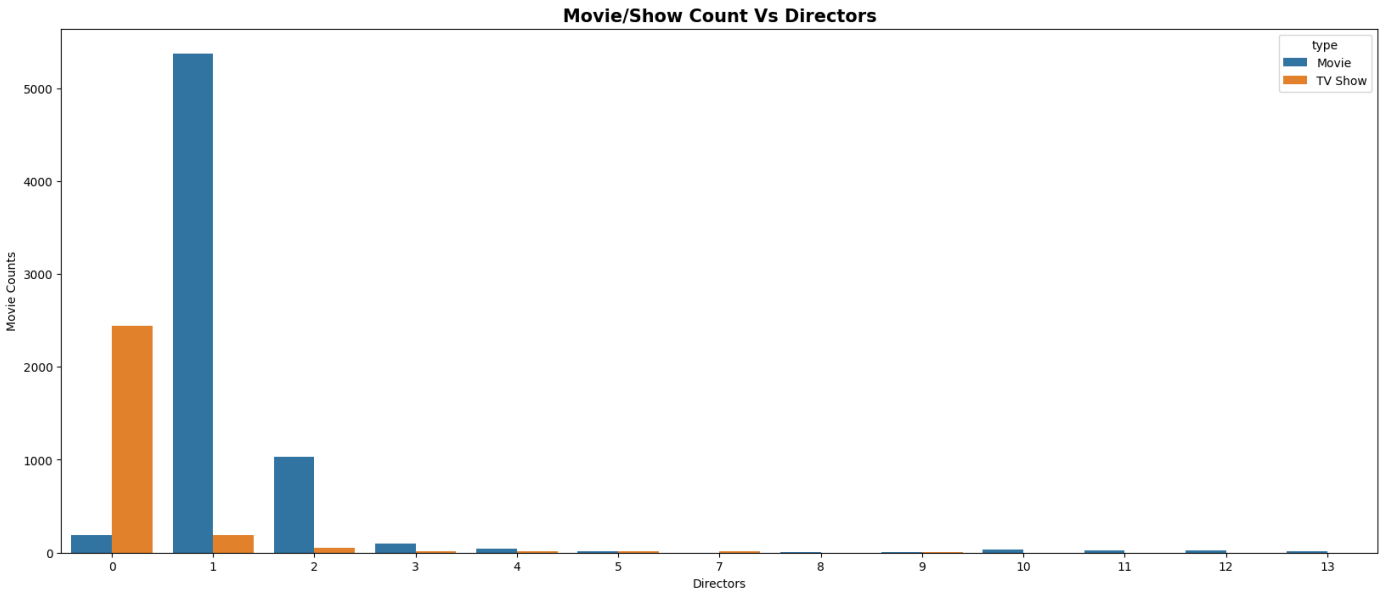
Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	lis
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Docume
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Intern TV Sho Dram My
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	Cri S Intern TV Sho
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	1 Season	Docu Rea

In [ ]:
```
plt.figure(figsize=(20,8))
sns.countplot(data=drc,x='Directors',hue='type',dodge=True)
plt.title('Movie/Show Count Vs Directors',fontsize=15,weight='bold')
plt.ylabel('Movie Counts',fontsize=10)

#countplot
```

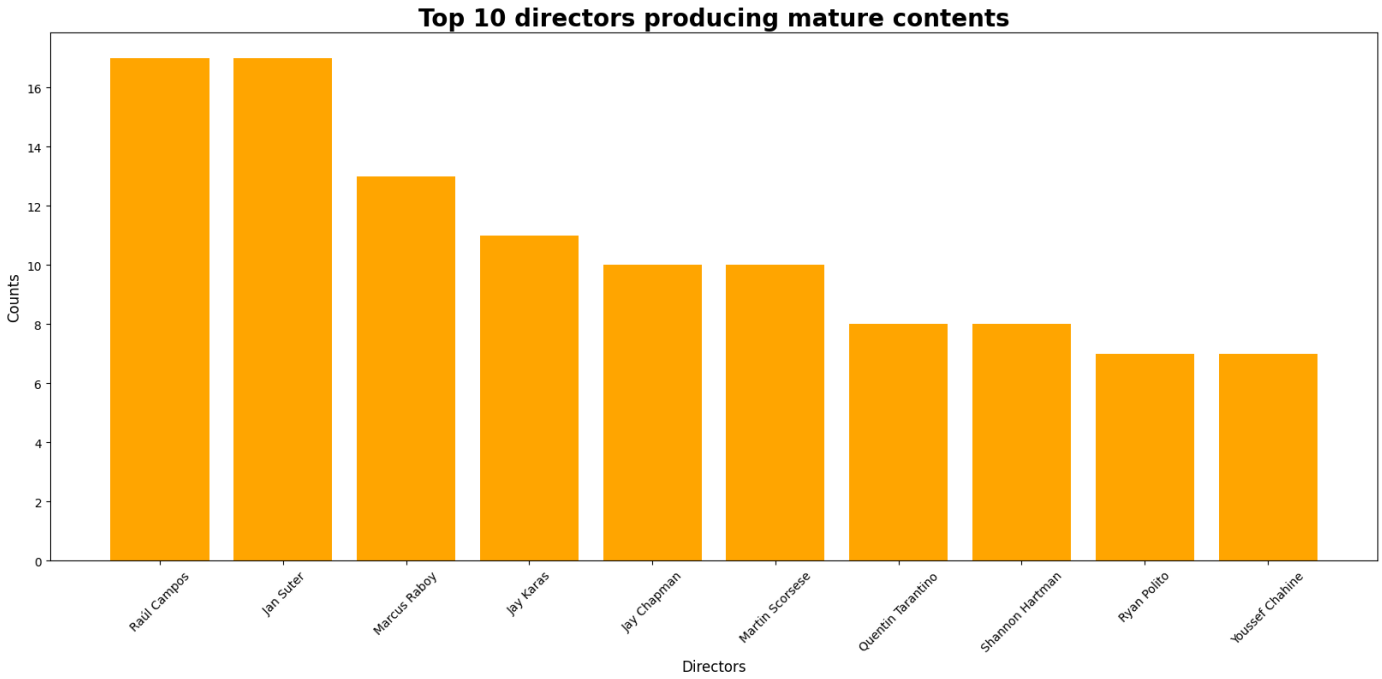
Out[ ]:
```
Text(0, 0.5, 'Movie Counts')
```



In [ ]:
```
Mature_movies = ['TV-MA','R','NC-17','G']
Adolescent = ['TV-14','TV-PG','PG-13','PG','TV-G','G']
Kids = ['TV-Y','TV-Y7-FV','G']
drc['rating_new'] = drc['rating'].apply(lambda x: 'Mature' if x in (Mature_movies) else
```

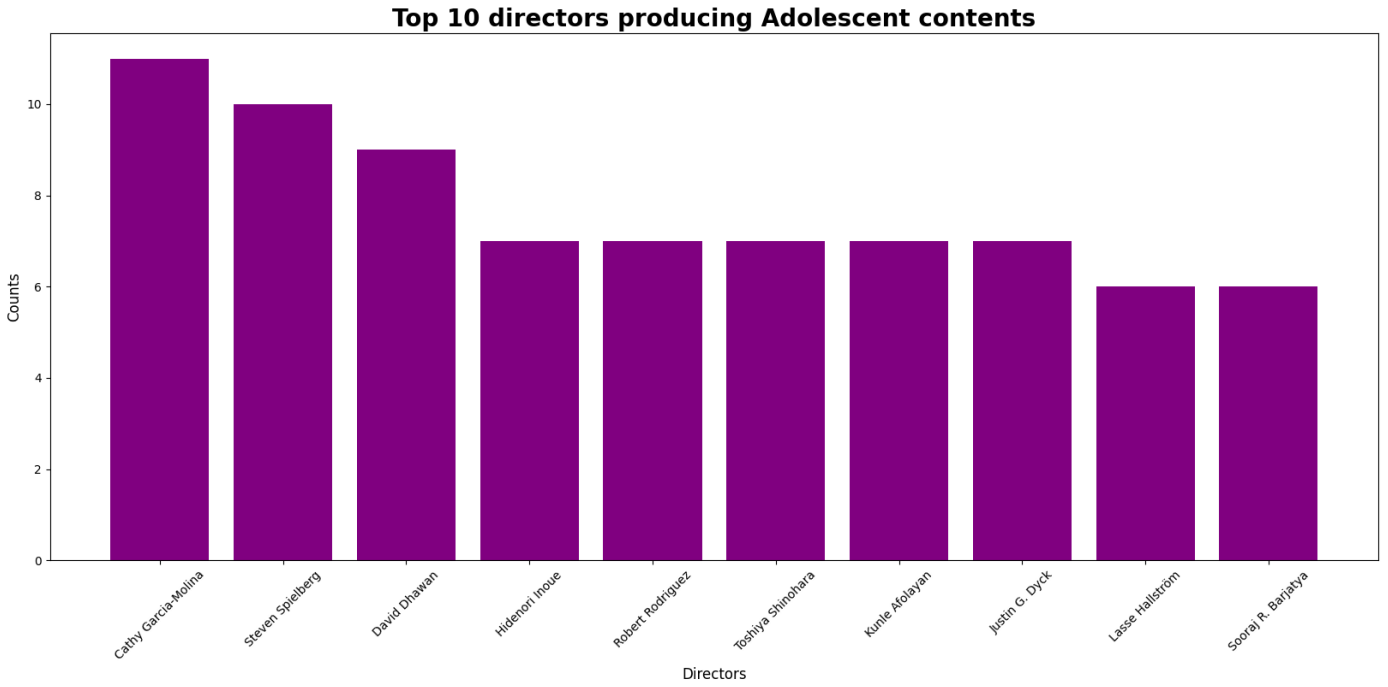
In [ ]:
```
d_m = drc.query("rating_new == 'Mature'")
dm = d_m['director'].value_counts()
mval = dm.values[:10]
mname = dm.index[:10]
```

In [ ]:
```
plt.figure(figsize=(20,8))
plt.bar(mname,mval,color='orange')
plt.xticks(rotation = 45)
plt.title('Top 10 directors producing mature contents',fontsize=20,weight='bold')
plt.xlabel('Directors',fontsize=12)
plt.ylabel('Counts',fontsize=12)
plt.show()
```



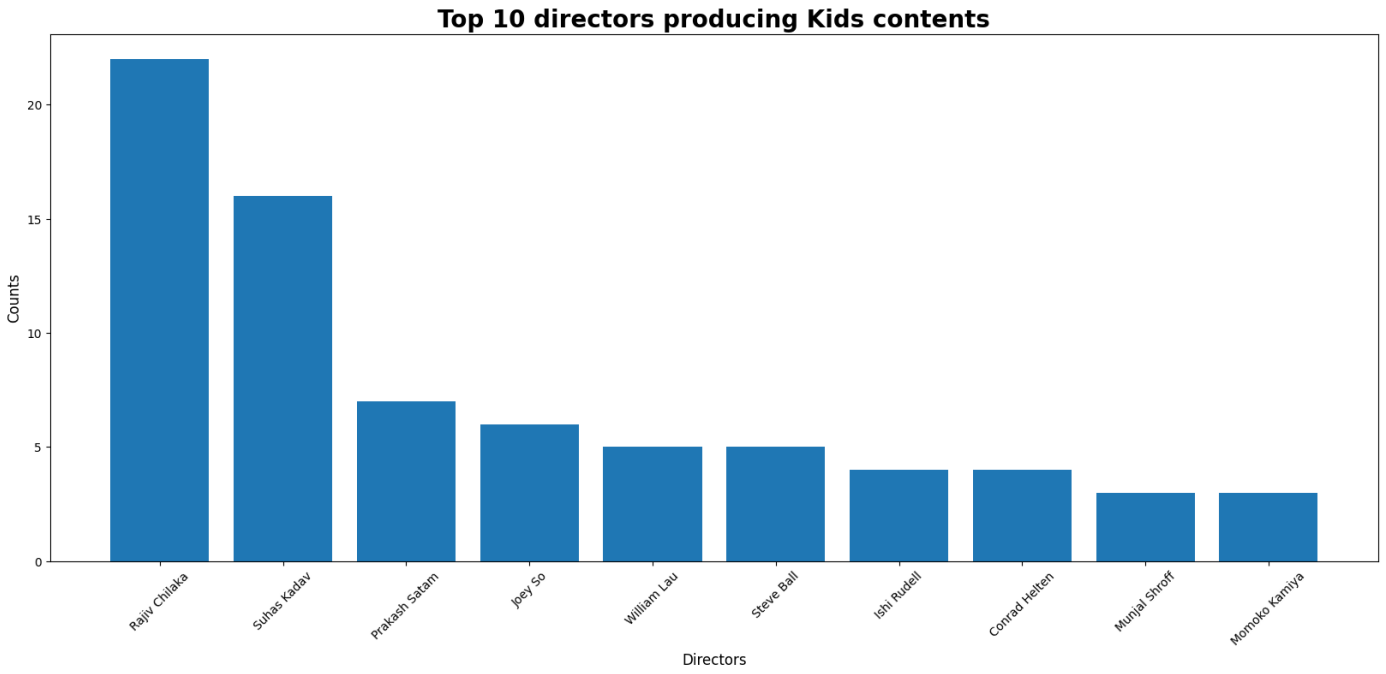
In [ ]:
```
d_a = drc.query("rating_new == 'Adolescent'")
da = d_a['director'].value_counts()
aval = da.values[:10]
aname = da.index[:10]
```

In [ ]:
```
plt.figure(figsize=(20,8))
plt.bar(aname,aval,color='purple')
plt.xticks(rotation = 45)
plt.title('Top 10 directors producing Adolescent contents',fontsize=20,weight='bold')
plt.xlabel('Directors',fontsize=12)
plt.ylabel('Counts',fontsize=12)
plt.show()
```



In [ ]:
```
d_k = drc.query("rating_new == 'Kids'")
dk = d_k['director'].value_counts()
kval = dk.values[:10]
kname = dk.index[:10]
```

In [ ]:
```
plt.figure(figsize=(20,8))
plt.bar(kname,kval)
plt.xticks(rotation = 45)
plt.title('Top 10 directors producing Kids contents',fontsize=20,weight='bold')
plt.xlabel('Directors',fontsize=12)
plt.ylabel('Counts',fontsize=12)
plt.show()
```



In [ ]:
```
drc.head(3)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	lis
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Docume
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Intern TV Sho Dram My
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	Cri S Intern TV Sho

In [ ]:
```
dr=drc.copy()
```

In [ ]:
```
dr["listed_in"]=dr["listed_in"].str.split(",")
```

In [ ]:
```
dr=dr.explode("listed_in")
```

In [ ]:
```
dr["listed_in"]=dr["listed_in"].apply(lambda x:x.lstrip())
```

In [ ]:
```
#Replacing the Repeated names and merging the category
dr['listed_in'] = dr['listed_in'].apply(lambda x:'Movies' if 'Movies' in x else 'Dramas'
```

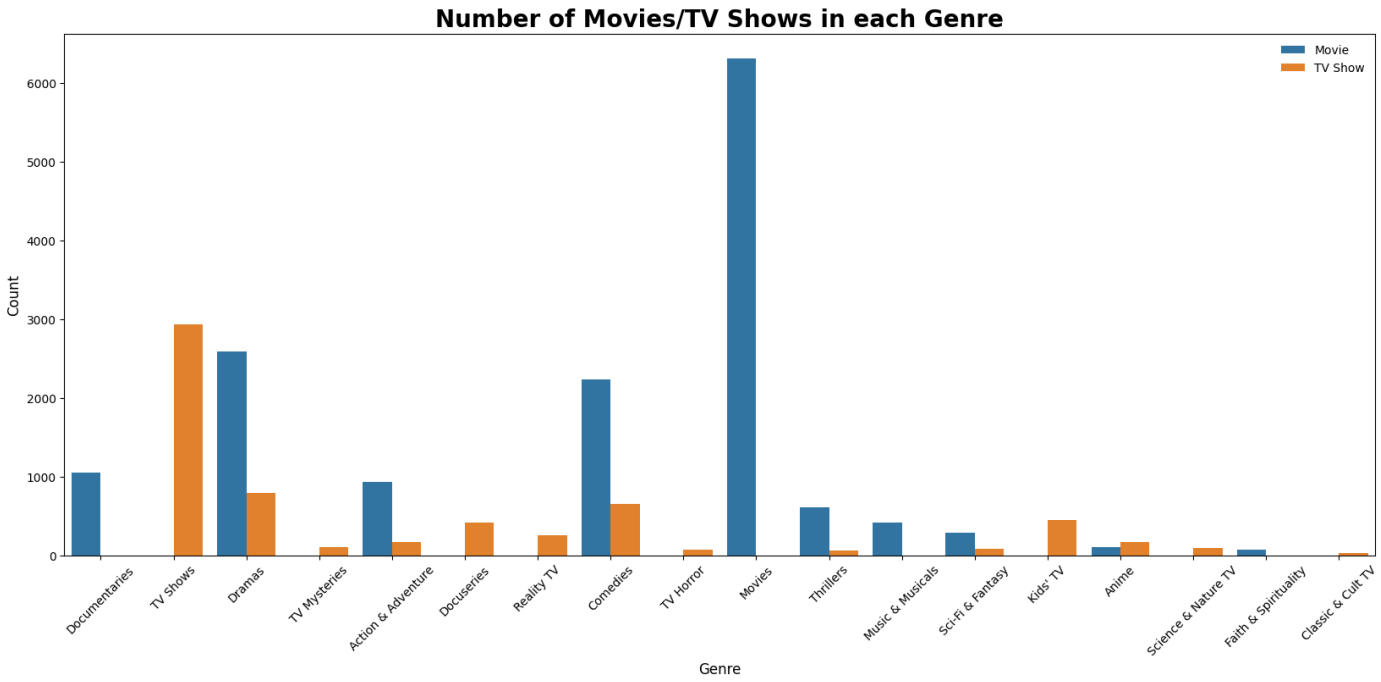
In [ ]:
```
dr.head()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	lis
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Docume
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	TV
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	D
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	TV My
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	TV

In [ ]:

```
plt.figure(figsize=(20,8))
sns.countplot(data=dr,x="listed_in",hue='type')
plt.legend(["Movie","TV Show"],loc="upper right",frameon=False)
plt.title("Number of Movies/TV Shows in each Genre",fontsize=20,weight="bold")
plt.xlabel("Genre",fontsize=12)
plt.ylabel("Count",fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



insights:

-Most famous directors ---> 'Rajiv Chilaka', 'Raúl Campos', ' Jan Suter', 'Marcus Raboy','Suhas Kadav

-comparitively Movies are most popular than TV Shows

Analysis on rating

-Mature movies --> ['TV-MA','R','NC-17','G']

-Adolescent --> ['TV-14','TV-PG','PG-13','PG','TV-G','G']

-Kids --> ['TV-Y','TV-Y7-FV','G']

-G --> all ages

-NR --> Not Rated

In [ ]:
```
dta=drc.copy()
```

creating a new column by replaciong Movies==1 and TV Shows==0

In [ ]:
```
dta["cat"]=dta["type"].apply(lambda x: 1 if x=="Movie" else 0)
```

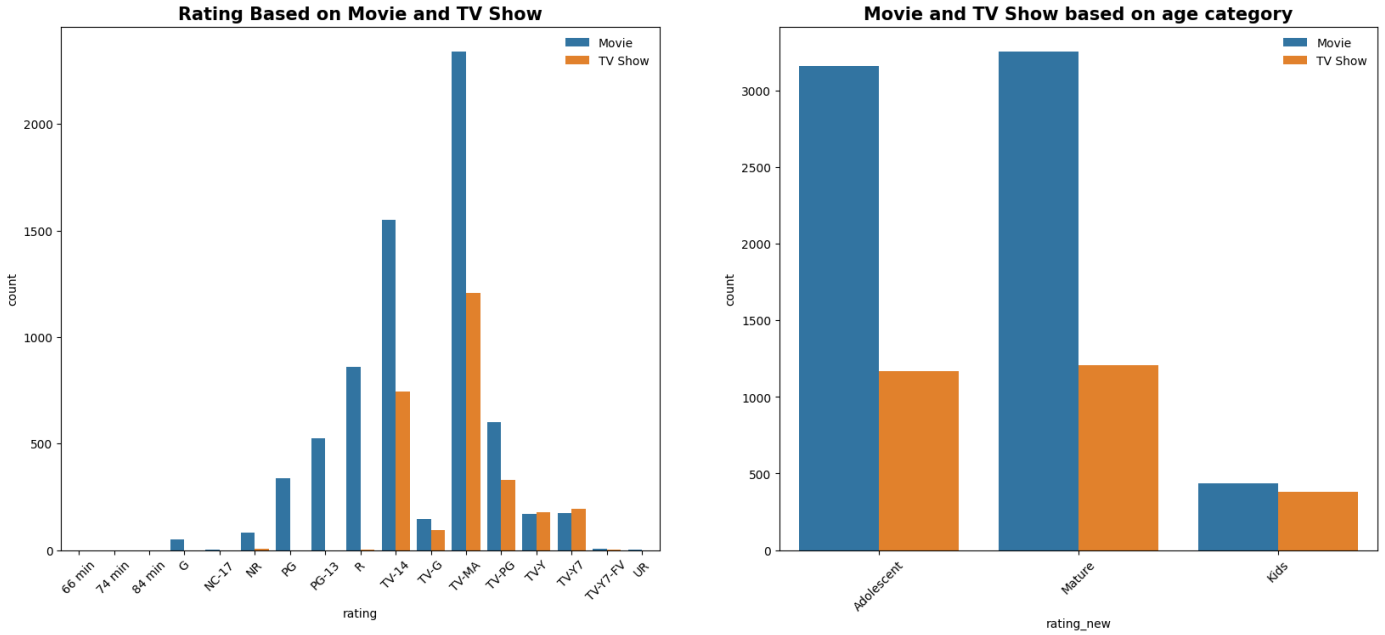
In [ ]:
```
d=dta.query("rating in ['66 min','74 min','84 min']")
dta.drop(index=d.index,inplace=True)
```

In [ ]:
```
dta.drop(["show_id","title","description"],axis=1,inplace=True)
```

In [ ]:
```
#Age category and type
Mature_movies=['TV-MA','R','NC-17','G']
Adolescent=['TV-14','TV-PG','PG-13','PG','TV-G','G']
Kids=['TV-Y','TV-Y7-FV','G']
dta['rating_new'] = dta['rating'].apply(lambda x: 'Mature' if x in (Mature_movies) else
```

In [ ]:
```
plt.figure(figsize=(20,8))
# Rating and type
plt.subplot(1,2,1)
sns.countplot(dta,x='rating',hue='type')
plt.legend(['Movie','TV Show'],loc='upper right',frameon=False)
plt.title('Rating Based on Movie and TV Show',fontsize=15,weight='bold')
plt.xticks(rotation=45)

# Age category and type
plt.subplot(1,2,2)
sns.countplot(dta,x='rating_new',hue='type')
plt.legend(['Movie','TV Show'],loc='upper right',frameon=False)
plt.title('Movie and TV Show based on age category',fontsize=15,weight='bold')
plt.xticks(rotation=45)
plt.show()
```



insights:

Most of the movies/shows are available on netflix is either of Mature or Adolescent is higher whereas kids content of shows/movies are lower

Analysis on Actors

In [ ]:
```
cst=df.copy()
```

In [ ]:
```
cst.drop(index=cst.query("cast.isna()").index,inplace=True)
```

In [ ]:
```
#splitting element
cst["cast"]=cst["cast"].str.split(",")

#Exploding cast column
cast=cst.explode("cast")
```

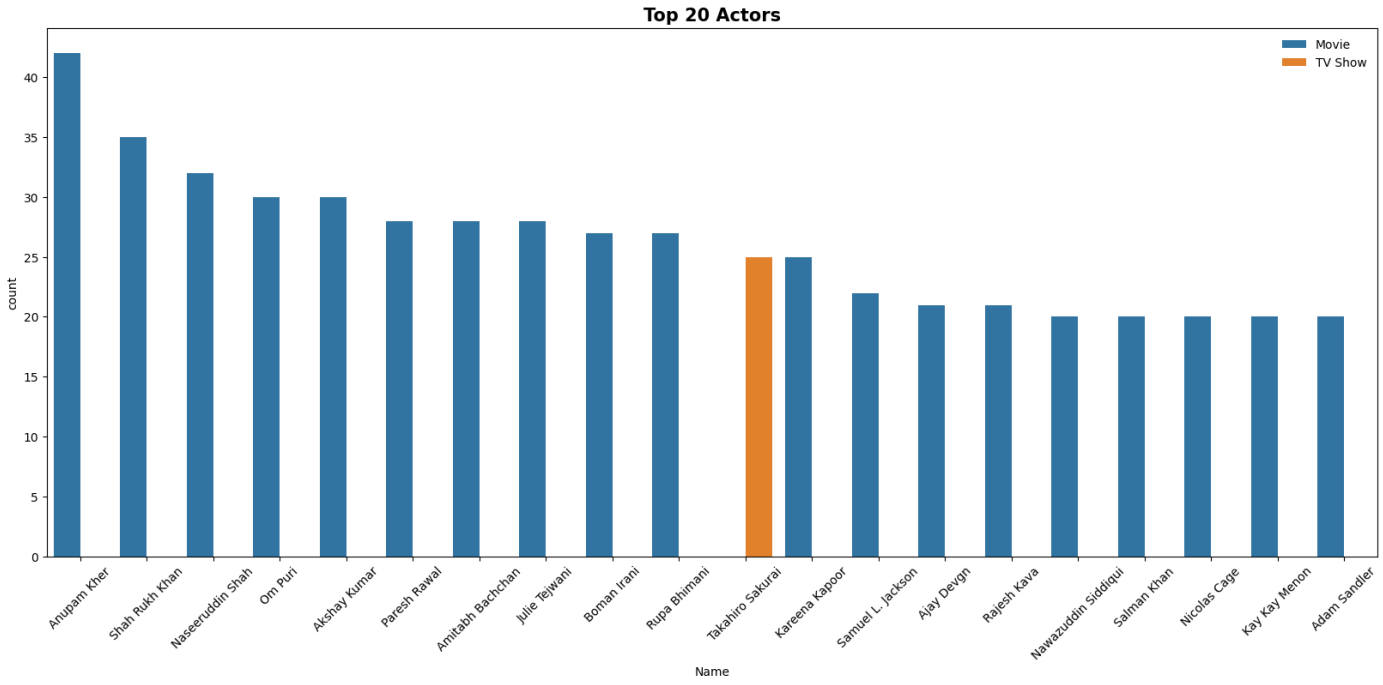
In [ ]:
```
cast["cast"]=cast["cast"].str.lstrip()
```

In [ ]:
```
cast['cast'].str.lstrip().value_counts().head(10).index
```

Out[ ]:
```
Index(['Anupam Kher', 'Shah Rukh Khan', 'Julie Tejwani', 'Naseeruddin Shah',
       'Takahiro Sakurai', 'Rupa Bhimani', 'Akshay Kumar', 'Om Puri',
       'Yuki Kaji', 'Paresh Rawal'],
      dtype='object')
```

In [ ]:
```
Top=pd.DataFrame(cast[["cast","type"]].value_counts()).reset_index()
Top.columns=["Name","type","count"]
Top=Top.head(20)
```

In [ ]:
```
plt.figure(figsize=(20,8))
sns.barplot(x="Name",y="count",data=Top,hue='type')
plt.title("Top 20 Actors",fontsize=15,weight="bold")
plt.legend(loc="upper right",frameon=False)
plt.xticks(rotation=45)
plt.show()
#barplot
```



Top 10 Actors Category(rating)wise

In [ ]:
```
Mature_movies = ['TV-MA','R','NC-17','G']
Adolescent = ['TV-14','TV-PG','PG-13','PG','TV-G','G']
Kids = ['TV-Y','TV-Y7-FV','G']
cast['rating_new'] = cast['rating'].apply(lambda x: 'Mature' if x in (Mature_movies) els
```

In [ ]:
```
mat = cast.query("rating_new=='Mature'")
mat[['cast','type']]
```

Out[ ]:

	cast	type
1	Ama Qamata	TV Show
1	Khosi Ngema	TV Show
1	Gail Mabalane	TV Show
1	Thabang Molaba	TV Show
1	Dillon Windvogel	TV Show
...	...	...
8804	Emma Stone	Movie
8804	Abigail Breslin	Movie
8804	Amber Heard	Movie
8804	Bill Murray	Movie
8804	Derek Graf	Movie

30573 rows × 2 columns

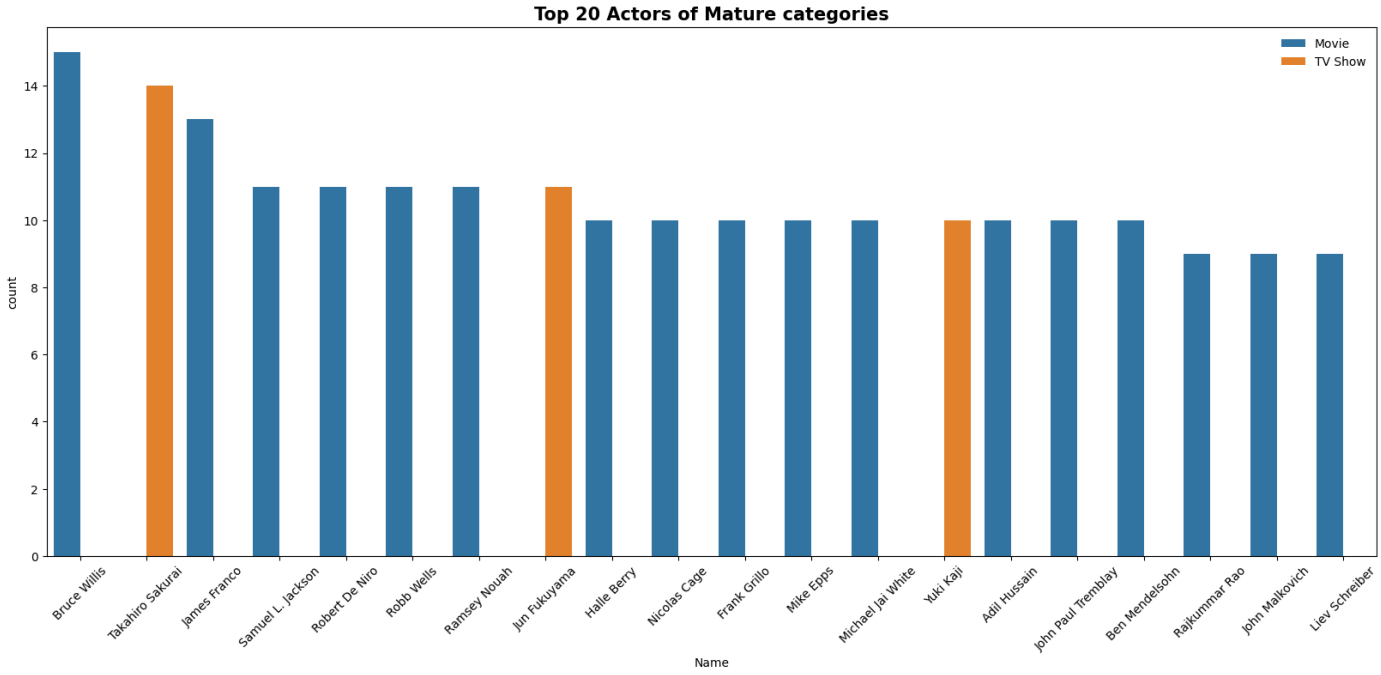
In [ ]:
```
mat = pd.DataFrame(mat[['cast','type']].value_counts()).reset_index()
mat.columns=['Name','type','count']
topm = mat.head(20)
```

In [ ]:
```
plt.figure(figsize=(20,8))

sns.barplot(x='Name',y='count',data=topm,hue='type')
plt.title('Top 20 Actors of Mature categories',fontsize=15,weight='bold')
plt.legend(loc='upper right',frameon=False)
plt.xticks(rotation=45)

plt.show()

#barplot
```



In [ ]:
```
ado = cast.query("rating_new=='Adolescent'")
```

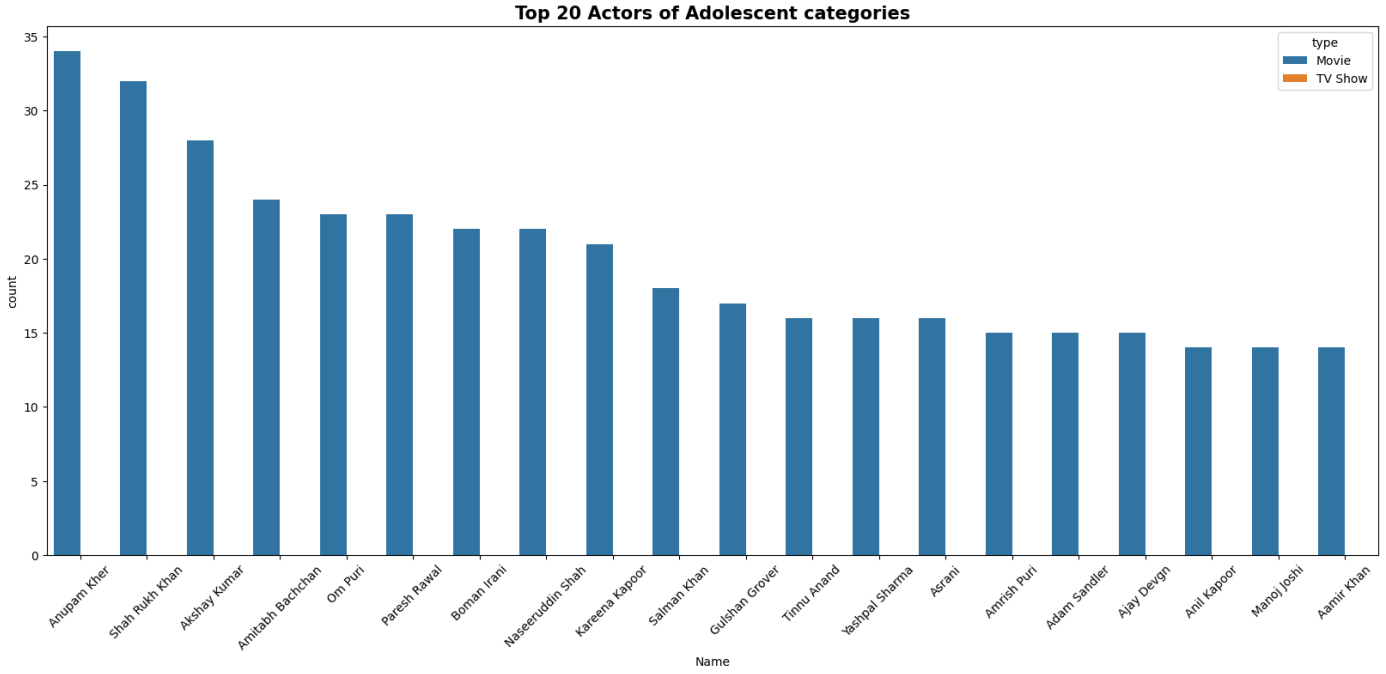
In [ ]:
```
ad = pd.DataFrame(ado[['cast','type']].value_counts()).reset_index()
ad.columns=['Name','type','count']
tops = ad.head(20)
```

In [ ]:
```
fig = plt.figure(figsize=(20,8))

sns.barplot(x='Name',y='count',data=tops,hue='type')
plt.title('Top 20 Actors of Adolescent categories',fontsize=15,weight='bold')
plt.xticks(rotation=45)

plt.show()

#barplot
```



In [ ]:
```
Kids = ['TV-Y','TV-Y7-FV','G']
```

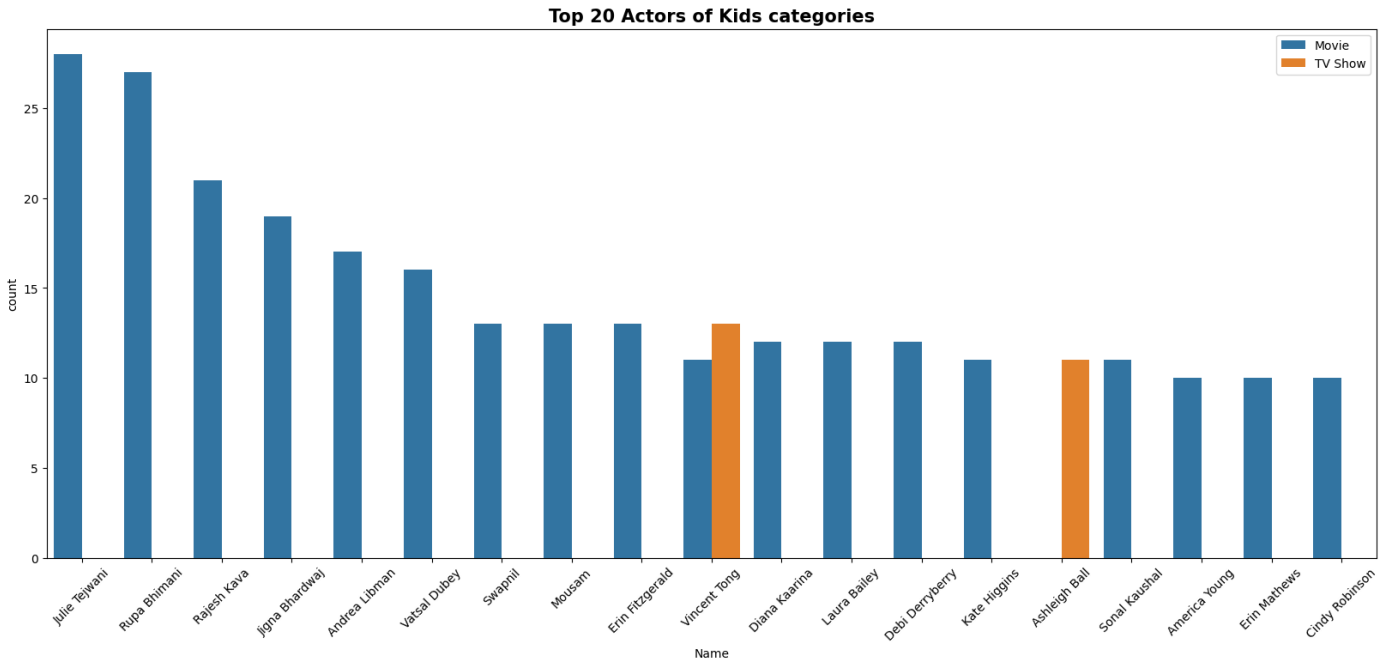
In [ ]:
```
kd = cast.query("rating_new=='Kids'")
```

In [ ]:
```
kd = pd.DataFrame(kd[['cast','type']].value_counts()).reset_index()
kd.columns=['Name','type','count']
topk = kd.head(20)
```

In [ ]:
```
plt.figure(figsize=(20,8))

sns.barplot(x='Name',y='count',data=topk,hue='type')
plt.title('Top 20 Actors of Kids categories',fontsize=15,weight='bold')
plt.legend(loc='upper right')
plt.xticks(rotation=45)

plt.show()
```



insights

Top 10 actors of all categories are --> ['Anupam Kher', 'Shah Rukh Khan', 'Julie Tejwani', 'Naseeruddin Shah', 'Takahiro Sakurai', 'Rupa Bhimani', 'Akshay Kumar', 'Om Puri', 'Yuki Kaji', 'Paresh Rawal']

Major Focus On Netflix

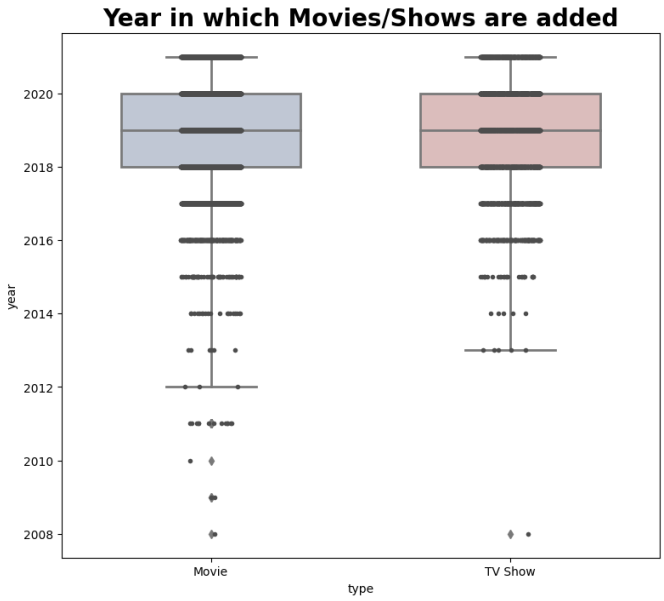
In [ ]:
```
df.info()
```

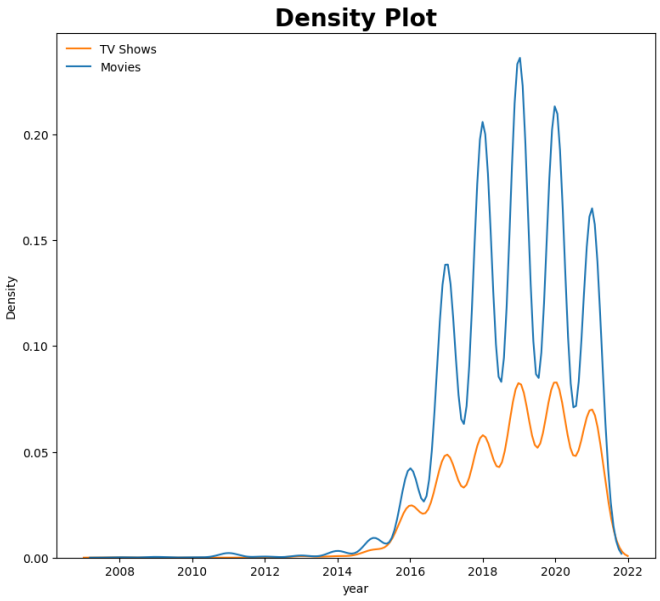
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       8807 non-null   object        
 1   type          8807 non-null   category      
 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   datetime64[ns]
 7   release_year  8807 non-null   int64         
 8   rating        8803 non-null   category      
 9   duration      8804 non-null   object        
 10  listed_in     8807 non-null   object        
 11  description   8807 non-null   object        
 12  year          8797 non-null   float64       
dtypes: category(2), datetime64[ns](1), float64(1), int64(1), object(8)
memory usage: 775.0+ KB
```

In [ ]:
```
plt.figure(figsize=(20,8))
#plot for year in which movies/shows added
plt.subplot(1,2,1)
sns.boxplot(data=df,x="type",y="year",linewidth=2,whis=3,width=0.6,palette="vlag")
plt.title("Year in which Movies/Shows are added",fontsize=20,weight="bold")
sns.stripplot(x="type",y="year",data=df,size=4,color=".3",linewidth=0)

#Density plot
```

```
plt.subplot(1,2,2)
sns.kdeplot(data=df,x="year",hue="type")
plt.title("Density Plot",fontsize=20,weight="bold")
plt.legend(["TV Shows","Movies"],loc="upper left",frameon=False)
plt.show()
```





Insights

Netflix started it's online operation in near around 2008, it took 10 years to upload 25% percent of content on it's platform, Whereas in just 1 year it added 25% of shows and movies. From above left plot we can see that median year is 2019, when most of the shows/movies added to the platform.

The pattern seems similar for both Movies and TV Show's addition to the platform, which means Netflix has focused on both the categories. Although, overall size of movies is much larger than TV Shows.

Country Wise Analysis

In [ ]:
```
dc=df.copy()
```

In [ ]:
```
#splitting elements by ","
dc["country"]=dc["country"].str.split(",")

#separating different countries with explode function
dc=dc.explode("country")

#dropping Nan Values
dc.dropna(subset=["country"],inplace=True)

#Removing left white space
dc["country"]=dc["country"].apply(lambda x:x.lstrip())
```

In [ ]:
```
dfc=dc.copy()
```

In [ ]:
```
dfc=dfc["country"].value_counts().head(10).index
```

In [ ]:
```
top10_countrys=dfc.to_list()
```

In [ ]:
```
top10_countrys
```

Out[ ]:
```
['United States',
 'India',
 'United Kingdom',
 'Canada',
 'France',
 'Japan',
 'Spain',
 'South Korea',
 'Germany',
 'Mexico']
```

In [ ]:
```
# Data of top 10 countries .i.e ['United States', 'India', 'United Kingdom', 'Canada', '
dfc=dc[dc["country"].isin(top10_countrys)]
```

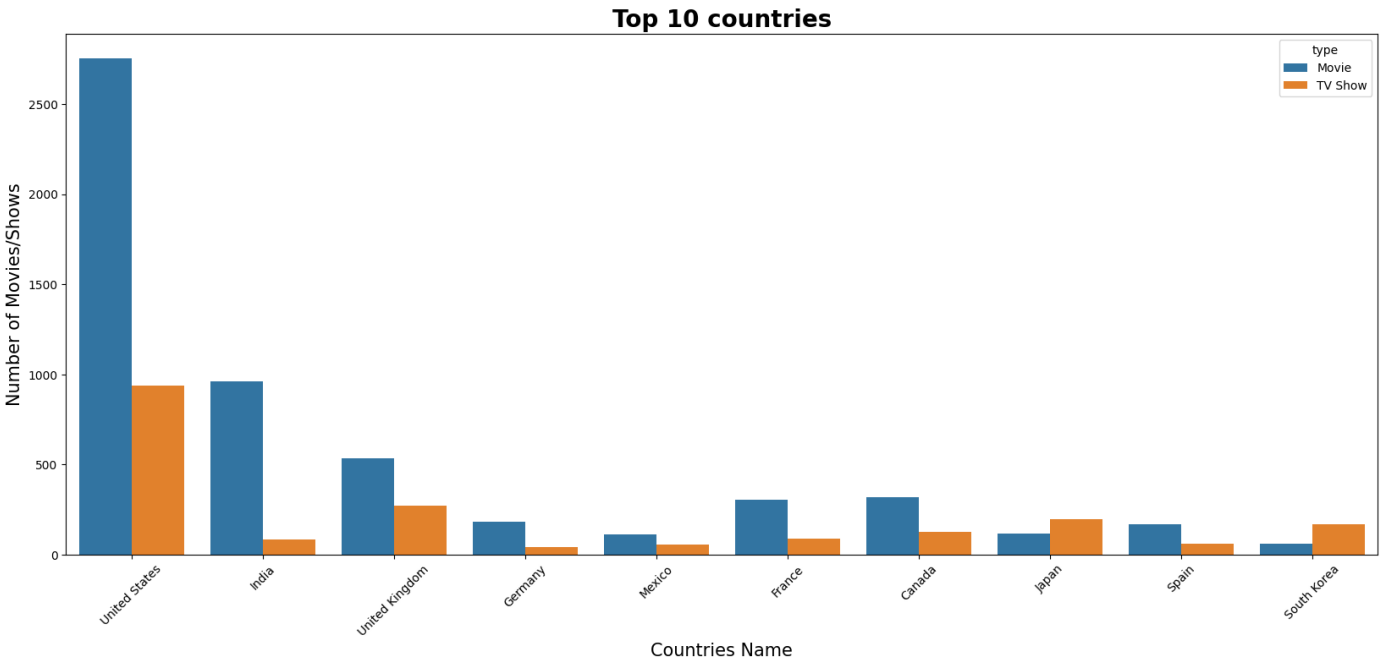
In [ ]:
```
dfc.head()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	lis
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Docume
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2 Seasons	Intern TV S Roma Shows
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States	2021-09-24	1993	TV-MA	125 min	D Indep M Intern I
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United Kingdom	2021-09-24	1993	TV-MA	125 min	D Indep M Intern I
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	Germany	2021-09-24	1993	TV-MA	125 min	D Indep M Intern I

In [ ]:
```
plt.figure(figsize=(20,8))
sns.countplot(data=dfc,x="country",hue="type")
plt.xticks(rotation=45)
plt.title("Top 10 countries",fontsize=20,weight="bold")
plt.xlabel("Countries Name",fontsize=15)
plt.ylabel("Number of Movies/Shows",fontsize=15)
plt.show()

#Countplot
```



Insights

-From above its clear that top 3 coutries of choice are United States, India and United Kingdom

In [ ]:
```
# Top 10 Directors and countries
dir = ['Rajiv Chilaka', 'Raúl Campos', 'Jan Suter', 'Marcus Raboy', 'Suhas Kadav', 'Jay
country = ['United States', 'India', 'United Kingdom', 'Canada', 'France', 'Japan', 'Spa
```

In [ ]:
```
# Top 10 Directors who belongs to top 10 countries
top_dc = dc[(dc['country'].isin(country)) & (dc['director'].isin(dir))]
```

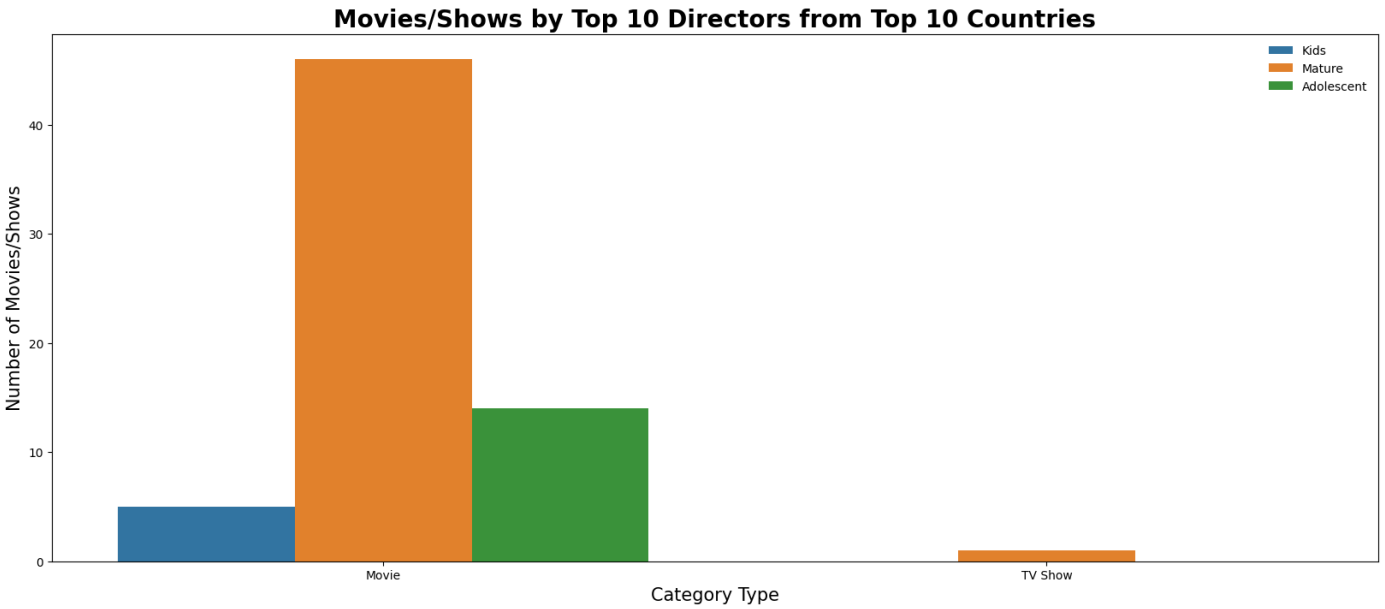
In [ ]:
```
# Categorising different ratings into 3 Groups
Mature_movies = ['TV-MA','R','NC-17','G']
Adolescent = ['TV-14','TV-PG','PG-13','PG','TV-G','G']
Kids = ['TV-Y','TV-Y7-FV','G']
top_dc['rating_new'] = top_dc['rating'].apply(lambda x: 'Mature' if x in (Mature_movies)
```

```
<ipython-input-153-9b5cc9962819>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  top_dc['rating_new'] = top_dc['rating'].apply(lambda x: 'Mature' if x in (Mature_movie
s) else 'Adolescent' if x in (Adolescent) else 'Kids')
```

In [ ]:
```
# Type of content produced in top 10 countries from top 10 directors
plt.figure(figsize=(20,8))

sns.countplot(data=top_dc,x='type',hue='rating_new')
plt.title('Movies/Shows by Top 10 Directors from Top 10 Countries',fontsize=20,weight='b
plt.xlabel('Category Type',fontsize = 15)
plt.ylabel('Number of Movies/Shows',fontsize = 15)
plt.legend(['Kids','Mature','Adolescent'],frameon=False)
plt.show()
```



Insights

-Most favourable content is Mature

In [ ]:
```
Actors = ['Anupam Kher', 'Shah Rukh Khan', 'Julie Tejwani', 'Naseeruddin Shah','Takahiro
```

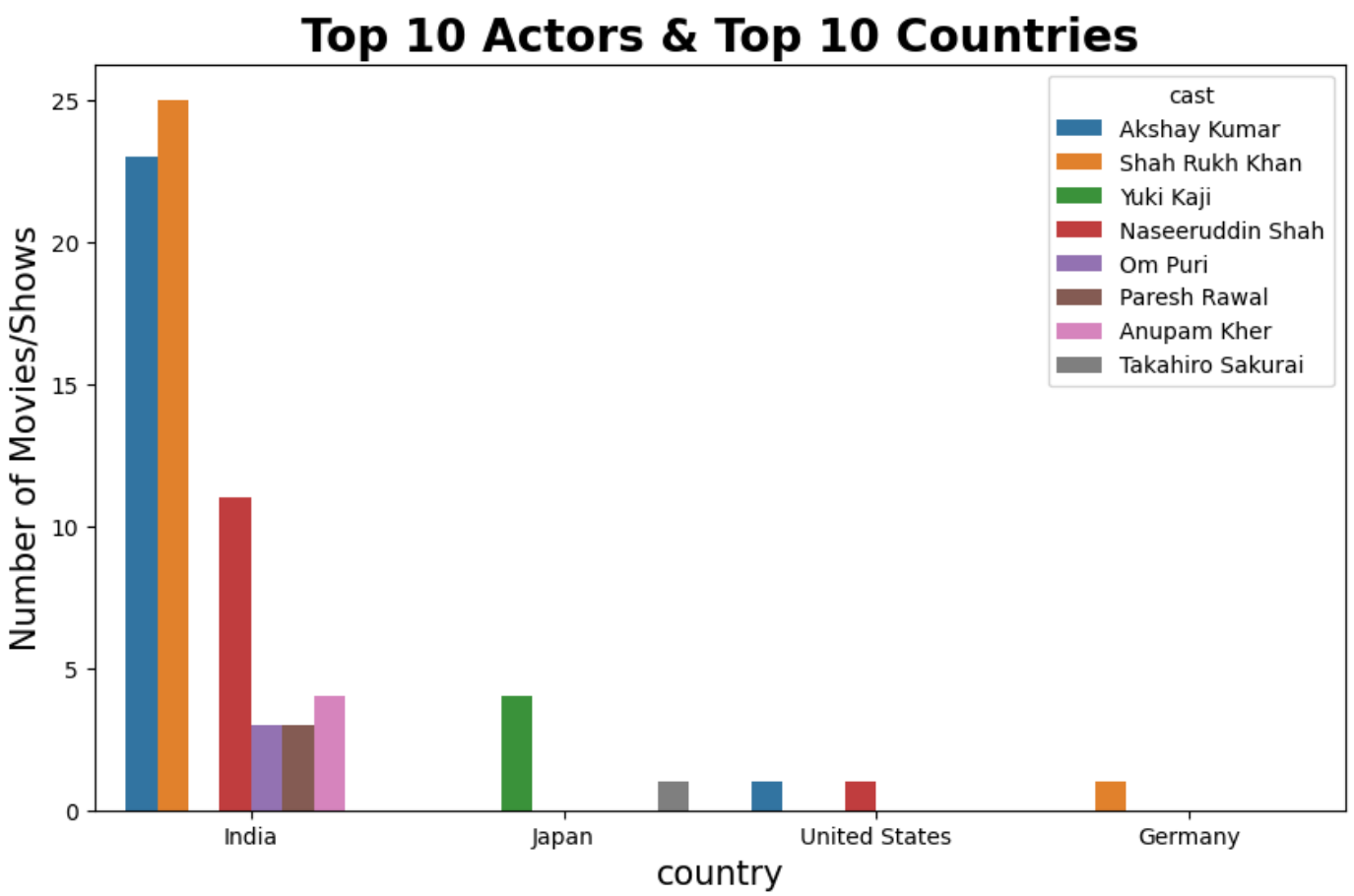
In [ ]:
```
# Splitting element
dc['cast'] = dc['cast'].str.split(',')

# Exploding cast column
dc = dc.explode('cast')
```

In [ ]:
```
dcc = dc[(dc['cast'].isin(Actors)) & (dc['country'].isin(country))]
```

In [ ]:
```
# Top 10 Actors from top 10 countries
plt.figure(figsize=(10,6))

sns.countplot(data=dcc,x='country',hue='cast')
plt.title('Top 10 Actors & Top 10 Countries',fontsize=20,weight='bold')
plt.xlabel('country',fontsize = 15)
plt.ylabel('Number of Movies/Shows',fontsize = 15)
# plt.legend(['Kids','Mature','Adolescent'],frameon=False)
plt.show()
```



In [ ]:
```
dc=df.copy()
```

In [ ]:
```
dc['country'] = dc['country'].str.split(',')

# Separating different countries with explode function
dc = dc.explode('country')

# Dropping Nan Values
dc.dropna(subset=['country'],inplace=True)

# Removing left white space
dc['country'] = dc['country'].apply(lambda x: x.lstrip())
```

In [ ]:
```
# Splitting elements by ','
dc['listed_in'] = dc['listed_in'].str.split(',')

# Separating different elements with explode function
dc = dc.explode('listed_in')

# Dropping Nan Values
dc.dropna(subset=['listed_in'],inplace=True)

# Removing left white space
dc['listed_in'] = dc['listed_in'].apply(lambda x: x.lstrip())
# Replacing the repeated names and merging the category
dc['listed_in'] = dc['listed_in'].apply(lambda x:'Movies' if 'Movies' in x else 'Dramas'
```

In [ ]:
```
dc = dc.query("country == ['United States','India','United Kingdom','Canada','Germany']"
```

In [ ]:
```
# Most favourable content
Country_content = pd.DataFrame(dc[['country','listed_in']].value_counts()).reset_index()
Country_content.columns = ['Country',"Content Type","Count"]
Country_content.head(10)
```

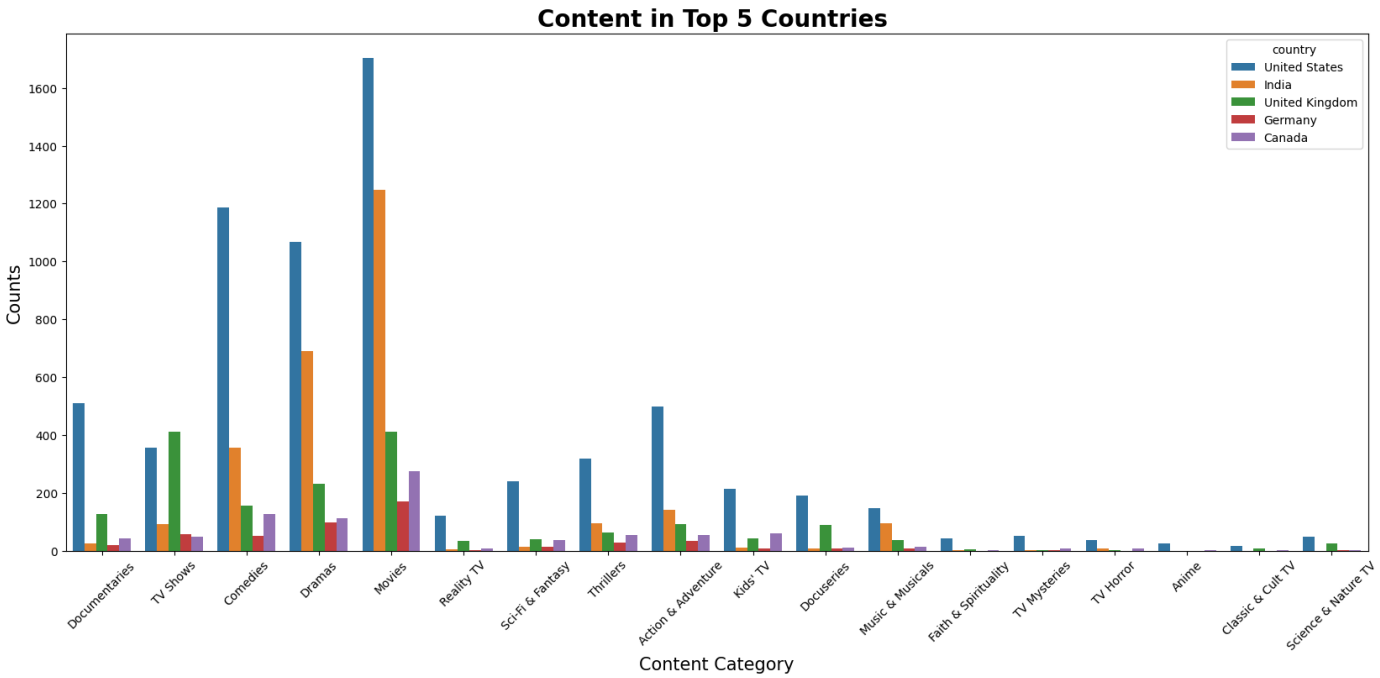
Out[ ]:

	Country	Content Type	Count
0	United States	Movies	1703
1	India	Movies	1247
2	United States	Comedies	1187
3	United States	Dramas	1067
4	India	Dramas	690
5	United States	Documentaries	512
6	United States	Action & Adventure	498
7	United Kingdom	TV Shows	413
8	United Kingdom	Movies	411
9	India	Comedies	358

In [ ]:
```
plt.figure(figsize=(20,8))

sns.countplot(data=dc,hue='country',x='listed_in')
plt.xticks(rotation = 45)
plt.title('Content in Top 5 Countries ',fontsize=20,weight='bold')
plt.xlabel('Content Category',fontsize = 15)
plt.ylabel('Counts',fontsize = 15)
```

Out[ ]:
```
Text(0, 0.5, 'Counts')
```



Insights

United States --> Movies, Comedies and Dramas are the top 3 content

India --> Movies, Dramas and Comedies are also for India

United Kingdom --> TV Shows, Movies and Dramas

Analysis of Top Content on the basis of Time

In [ ]:
```
df.head(2)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	liste
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Documenta
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Internati TV Shows Dramas Myste

In [ ]:
```
df.duration.value_counts()
```

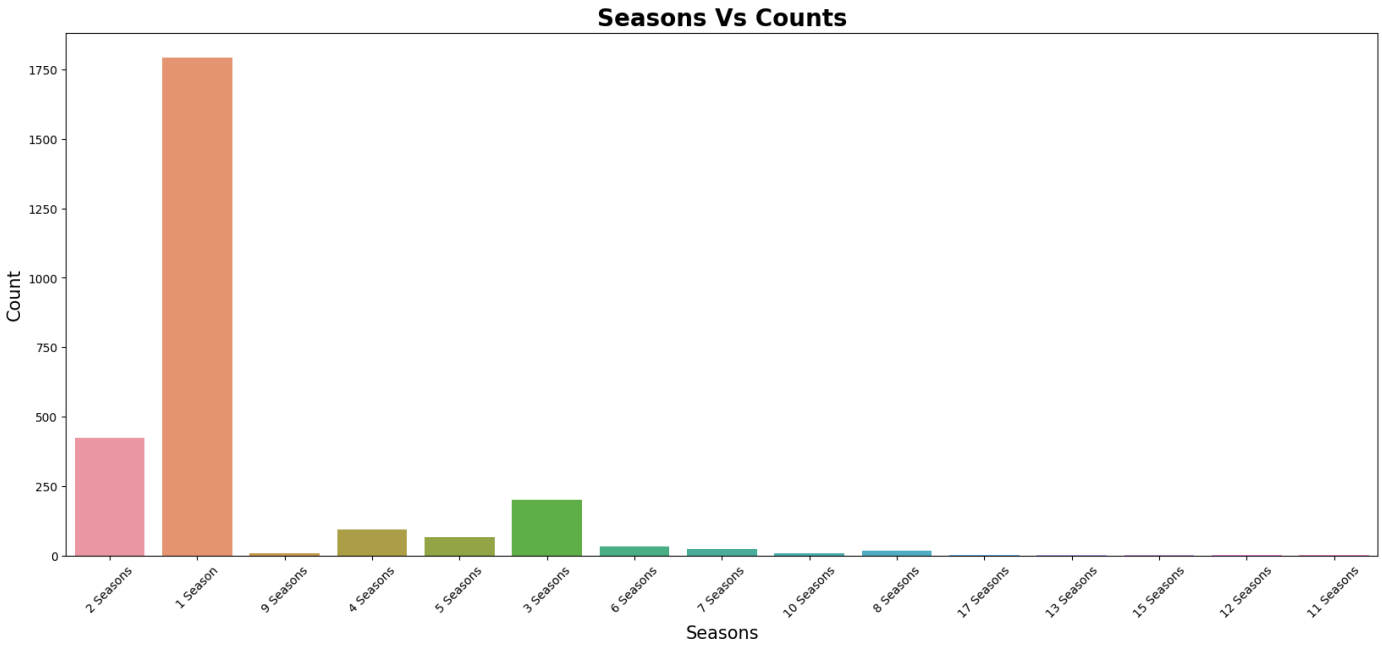
Out[ ]:
```
1 Season      1793
2 Seasons      425
3 Seasons      199
90 min         152
94 min         146
              ...
16 min           1
186 min          1
193 min          1
189 min          1
191 min          1
Name: duration, Length: 220, dtype: int64
```

TV Show seasons

In [ ]:
```
dl = df.dropna(subset=['duration'])
```

In [ ]:
```
dls = dl[dl['duration'].str.contains('Season')]
```

In [ ]:
```
plt.figure(figsize=(20,8))
sns.countplot(data=dls,x='duration')
plt.xticks(rotation = 45)
plt.title('Seasons Vs Counts',fontsize=20,weight='bold')
plt.xlabel('Seasons',fontsize=15)
plt.ylabel('Count',fontsize=15)
plt.show()
```



In [ ]:
```
dls[dls['duration'] == '17 Seasons']
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
548	s549	TV Show	Grey's Anatomy	NaN	Ellen Pompeo, Sandra Oh, Katherine Heigl, Just...	United States	2021-07-03	2020	TV-14	17 Seasons	Romantic TV Shows, TV Dramas

Insights

Maximum number of show availabe on netflix is of 1 Season, followed by 2 Seasons and 3 Seasons.

Show title with Name Grey's Anatomy have maximum number of seasons produced in United States.

In [ ]:
```
# Copying dataframe to new variable
dlx = dl.copy()
```

In [ ]:
```
# removing min from the movies duration to make it into integers
dlx['Movie'] = dlx['duration'].apply(lambda x: x.strip(' min') if 'min' in x else 0)
```

In [ ]:
```
# removing Seasons from TV Shows duration to make it into integers
dlx['Seasons'] = dlx['duration'].apply(lambda x: x.strip(' Seasons') if 'Seasons' in x e
```

In [ ]:
```
# converting objects into integer
dlx['Seasons'] = dlx['Seasons'].astype(int)
dlx['Movie'] = dlx['Movie'].astype(int)
```

In [ ]:
```
# Here 0 indicates number of TV shows, whereass all values greater than 0 indicated movi
dlx.Movie.value_counts().head(20)
```

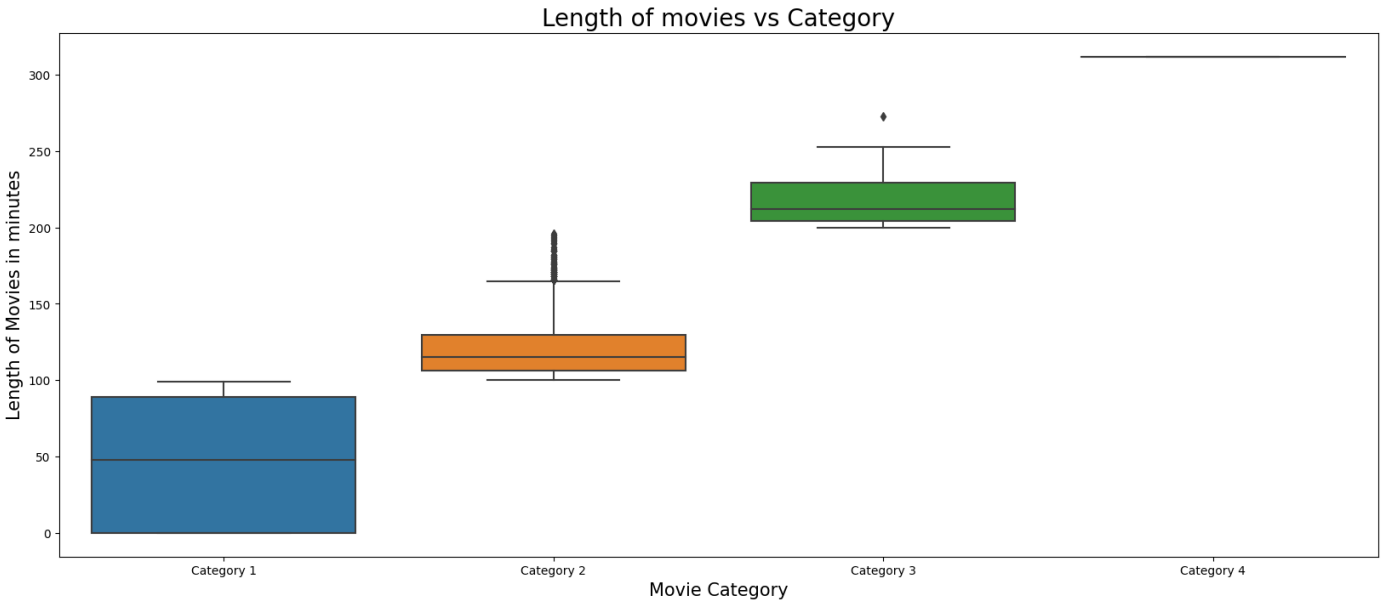
Out[ ]:
```
0      2676
90      152
97      146
94      146
93      146
91      144
95      137
96      130
92      129
102     122
98      120
99      118
101     116
88      116
103     114
106     111
100     108
89      106
104     104
86      103
Name: Movie, dtype: int64
```

In [ ]:
```
dlx['movie_cat'] = dlx['Movie'].apply(lambda x:'Category 4' if x >= 300 else 'Category 3
```

In [ ]:
```
dls = dlx[dlx['Movie'] > 100]
```

In [ ]:
```
plt.figure(figsize=(20,8))
sns.boxplot(data=dlx,x='movie_cat',y='Movie')
plt.xlabel('Movie Category',fontsize=15)
plt.ylabel('Length of Movies in minutes',fontsize=15)
plt.title('Length of movies vs Category',fontsize=20,weight=20)
```

Out[ ]:
```
Text(0.5, 1.0, 'Length of movies vs Category')
```



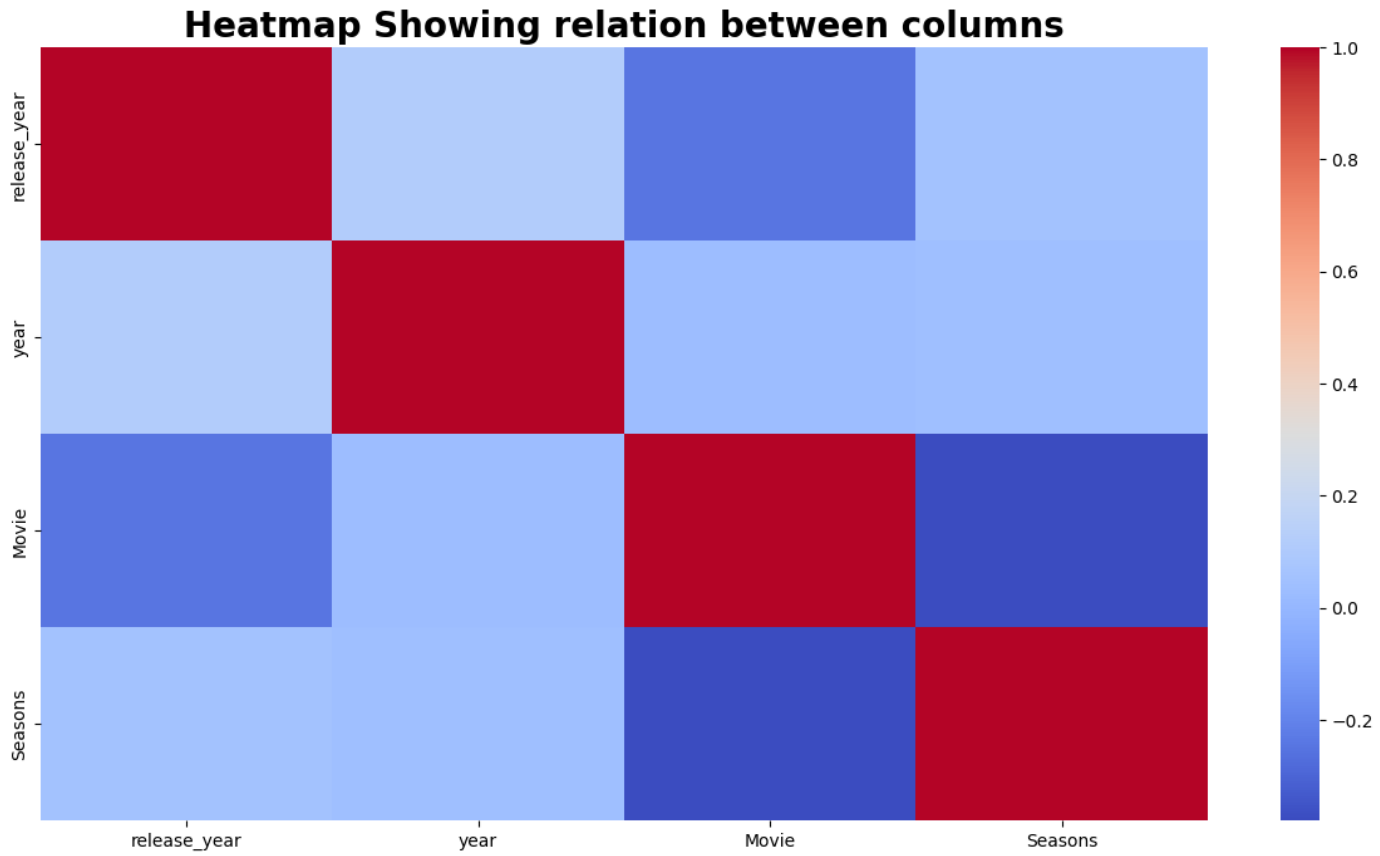
Insights

There are maximum movies with duration of 100 mintues and below

In [ ]:
```
a = dlx.corr(numeric_only=True)
```

In [ ]:
```
plt.figure(figsize=(15,8))
sns.heatmap(a,cmap='coolwarm')
plt.title('Heatmap Showing relation between columns',fontsize=20,weight='bold')
```

Out[ ]: Text(0.5, 1.0, 'Heatmap Showing relation between columns')



Insights

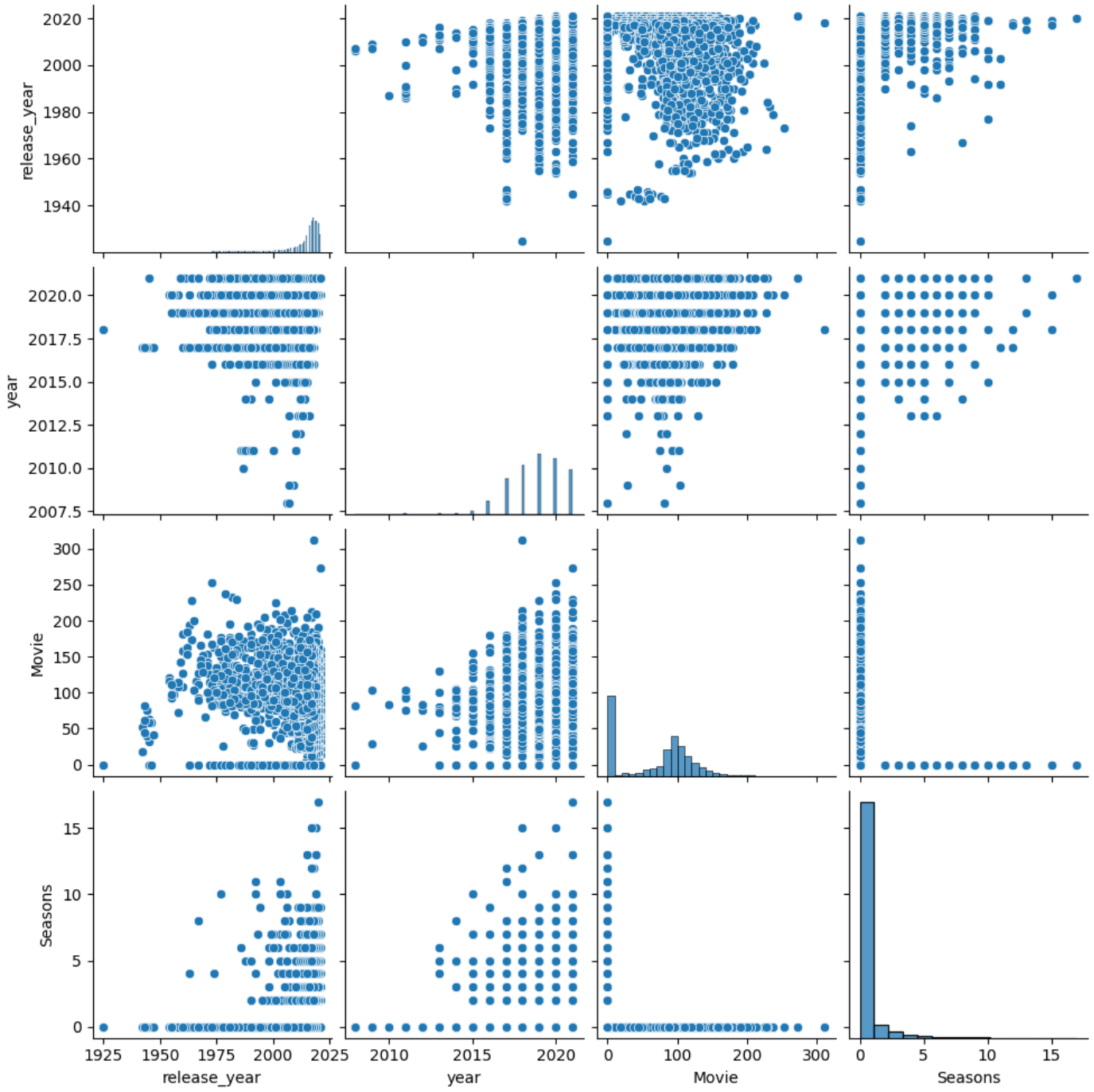
Season & Release Year ---> The value is very close to 0, So there is no relation between them

Movie & Release Year ---> The value is beyond -0.2 and very close to -0.3 which shows it has negative correlation between them.

year& Release Year ---> Again, 0 signifies no correlation

release_year with itself ---> Shows very high correlation, which is obvious

In [ ]:
```
# Pairplot between Movies, Release Year and Date added
sns.pairplot(data=dlx)
plt.show()
```



NETFLIX-Business Insights

1.**More than 50% of movies/shows added to the platform within 5 years of release**

-This shows, Netflix is doing best to cater the needs of user

2.The best time to launch a show is in December whereas, the best day is Friday.

-This could be due to holiday as users have enough amount of time to spend on their favorite shows.

3.Most famous directors --> Rajiv Chilaka, Raúl Campos, Jan Suter, Marcus Raboy,Suhas Kad

4.Contents on the platform --> Mature > Adolescent > Kids

-Most of the content belongs to Mature category

5.Most famous actors --> Anupam Kher, Shah Rukh Khan, Julie Tejwani, Naseeruddin Shah, Takahiro Sakurai

6.Most of the content is added in 2019

-Netflix is continuously adding movies, but the rate of that increased heavily in 2019

7.Most famous Countries United States, India, United Kingdom, Canada, France

-United States --> Movies, Comedies and Dramas are the top 3 content

-India --> Movies, Dramas and Comedies are also for India

-United Kingdom --> TV Shows, Movies and Dramas

8.Maximum number of show availabe on netflix is of 1 Season, followed by 2 Seasons and 3 Seasons.

-Show title with Name Grey's Anatomy have 17 Seasons which is produced in United States.

9.There are maximum of movies with duration of 100 mintues and below

-Movies having length near around 100 is performing good

Recommendations:

1. Time between Movie/TV Show release and uploaded to platform should be minimised.

2.It's good to launch movies in December and on Friday.

3.Consider adding movies/shows of actors and director who are performing best in the world.

4.Movies, Dramas and Comedies are top performer, it will be good if we add more of these.

5.Top 3 countries in overall terms is United States, India and United Kingdom. It would be good if Netflix focus on these countries. In addition, it could also increase it's reach in countries like Canada and France.