In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [71]:
```
df=pd.read_csv('/Users/praneetcb/Documents/netflix.csv')
```

In [10]:
```
#Analyze the data and generate insights that could help Netflix in decidi
#and how they can grow the business in different countries
```

1. Problem Statement: The problem that Netflix is facing is to determine which type of shows/movies to have and how they can grow their business in different countries. Netflix needs to analyze the data and generate insights to make informed decisions about the type of content to produce and how to expand their business globally.

Basic Metrics: To analyze the data and generate insights, We should first look at some basic metrics that will give the idea of their current performance.

a. Genre Preference of viewers

b. Top Countries that high highest viewers and Popular genre in those countries.

c. Viewing Time: We should track the amount of time subscribers spend watching their shows/movies to understand which shows are popular and which ones are not.

d. Ratings: We should track the ratings of their shows/movies to see which ones are well-received by their audience and which ones are not.

e. Cast/Directors : Top Directors and actors who appeared more number of time and their view time.

d. Comparison of tv shows vs. movies

2. Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary

In [18]:
```
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 [26]:
```
df.shape
```

Out[26]:
```
(8807, 12)
```

In [19]:
```
df.describe()
```

Out[19]:

	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

In [ ]:
```
#In Data-type we see that column --> date_added is in string/object value
```

In [201…
```
df['date_added']=pd.to_datetime(df['date_added'], format='%Y%m%d')
```

In [190…
```
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   datetime64[ns]
 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: datetime64[ns](1), int64(1), object(10)
memory usage: 825.8+ KB
```

In [ ]:
```
#Missing Value Detection
```

In [24]:
```
#Total missing values in whole Dataframe
df.isna().sum().sum()
```

Out[24]:
```
4307
```

In [22]:
```
#columns that have missing values
df.isna().sum()
```

Out[22]:
```
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 [ ]:
```
# Columns with director, cast and country lets keep them with Null values
```

In [222…
```
# Column date_added has 10 Missing value lets fill those with mean value
# Note: Also checked that the value added timestamp is not lesser than re
```

In [216…
```
x=df['date_added'].mean()
```

In [223…
```
mean_date=pd.Timestamp.date(x)
```

In [232…
```
df.loc[df['date_added'].isnull()]
df.fillna({'date_added': mean_date}, inplace=True)
```

In [ ]:
```
# Missing value from the column ---> Rating
```

In [234…
```
df[df['rating'].isna()]
```

Out[234]:

	show_id	type	title	director	cast	country	date_added	releas
5989	s5990	Movie	13TH: A Conversation with Oprah Winfrey & Ava ...	NaN	Oprah Winfrey, Ava DuVernay	NaN	2017-01-26	
6827	s6828	TV Show	Gargantia on the Verdurous Planet	NaN	Kaito Ishikawa, Hisako Kanemoto, Ai Kayano, Ka...	Japan	2016-12-01	
7312	s7313	TV Show	Little Lunch	NaN	Flynn Curry, Olivia Deeble, Madison Lu, Oisín ...	Australia	2018-02-01	
7537	s7538	Movie	My Honor Was Loyalty	Alessandro Pepe	Leone Frisa, Paolo Vaccarino, Francesco Miglio...	Italy	2017-03-01	

In [272…
```
# We can fill these values with mode value of Column - Rating as it highe
```

In [287…
```
mode=df['rating'].mode()[0]
```

In [288…
```
df['rating'].fillna(mode, inplace=True)
```

In [290…
```
df[df['show_id']=='s6828'] # As sample-example: We can see that missing v
```

Out[290]:

	show_id	type	title	director	cast	country	date_added	release_yea
6827	s6828	TV Show	Gargantia on the Verdurous Planet	NaN	Kaito Ishikawa, Hisako Kanemoto, Ai Kayano, Ka...	Japan	2016-12-01	201

In [295…
```
# Next column with missing value is duration lets check whats is been mis
```

In [294…
```
df[df['duration'].isna()]
```

Out[294]:

	show_id	type	title	director	cast	country	date_added	release_year	rat
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	2017-04-04	2017	
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	2016-09-16	2010	
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	2016-08-15	2015	

In [ ]:
```
# We can see that mismatch-entry is been done as the rating column has th
# and these values has to be placed in column --> duration
```

In [298…
```
df.loc[df['director']=='Louis C.K.','duration']=df['rating']
```

In [299…
```
df.loc[df['director']=='Louis C.K.'].head()
```

Out[299]:

	show_id	type	title	director	cast	country	date_added	release_year	rat
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	2017-04-04	2017	
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	2016-09-16	2010	
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	2016-08-15	2015	

In [ ]:
```
# Lets fill the rating column with mode value of that column
```

In [300…
```
df.loc[df['director']=='Louis C.K.','rating']=mode
```

In [301…
```
df.loc[df['director']=='Louis C.K.'].head()
```

Out[301]:

	show_id	type	title	director	cast	country	date_added	release_year	rat
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	2017-04-04	2017	
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	2016-09-16	2010	
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	2016-08-15	2015	

In [302…
```
df.isna().sum()
```

Out[302]:
```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added         0
release_year       0
rating             0
duration           0
listed_in          0
description        0
dtype: int64
```

In [ ]:
```
# Task of replacing missing values in the columns --> rating, date_added,
```

3. Non-Graphical Analysis: Value counts and unique attributes (Statistical summary)

In [303…
```
df['type'].value_counts() #Classification of content by type> -- Tv_Show
```

Out[303]:
```
Movie      6131
TV Show    2676
Name: type, dtype: int64
```

In [304…
```
df['country'].value_counts() #Number of content by country wise > -- Tv_S
```

Out[304]:
```
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
Name: country, Length: 748, dtype: int64
```

In [305…
```
df.groupby('type').count()
```

Out[305]:

type	show_id	title	director	cast	country	date_added	release_year	rating	durat
Movie	6131	6131	5943	5656	5691	6131	6131	6131	6
TV Show	2676	2676	230	2326	2285	2676	2676	2676	2

In [306…
```
df['release_year'].value_counts() #Number of movies and TV_shows by Year-
```

Out[306]:
```
2018    1147
2017    1032
2019    1030
2020     953
2016     902
        ...
1959       1
1925       1
1961       1
1947       1
1966       1
Name: release_year, Length: 74, dtype: int64
```

In [43]:
```
df['type'].unique()
```

Out[43]:
```
array(['Movie', 'TV Show'], dtype=object)
```

In [307…
```
df['director'].nunique() #Number of directors in the dataframe
```

Out[307]:
```
4528
```

In [309…
```
df['cast'].nunique()
```

Out[309]:
```
7692
```

In [311…
```
df['country'].nunique()
```

Out[311]:
```
748
```

In [49]:
```
df['release_year'].unique() #DataFrame that has released_years
```

Out[49]:
```
array([2020, 2021, 1993, 2018, 1996, 1998, 1997, 2010, 2013, 2017, 1975,
       1978, 1983, 1987, 2012, 2001, 2014, 2002, 2003, 2004, 2011, 2008,
       2009, 2007, 2005, 2006, 1994, 2015, 2019, 2016, 1982, 1989, 1990,
       1991, 1999, 1986, 1992, 1984, 1980, 1961, 2000, 1995, 1985, 1976,
       1959, 1988, 1981, 1972, 1964, 1945, 1954, 1979, 1958, 1956, 1963,
       1970, 1973, 1925, 1974, 1960, 1966, 1971, 1962, 1969, 1977, 1967,
       1968, 1965, 1946, 1942, 1955, 1944, 1947, 1943])
```

In [312…
```
df['release_year'].max() #Maximum year value
```

Out[312]:
```
2021
```

In [314…
```
df['release_year'].min() #Minimum year value
```

Out[314]:
```
1925
```

In [319…
```
df['date_added'].value_counts()
```

Out[319]:
```
2020-01-01    110
2019-11-01     91
2018-03-01     75
2019-12-31     74
2018-10-01     71
              ...
2017-02-21      1
2017-02-07      1
2017-01-29      1
2017-01-25      1
2020-01-11      1
Name: date_added, Length: 1714, dtype: int64
```

In [320…
```
df['listed_in'].value_counts()
```

Out[320]:
```
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
Name: listed_in, Length: 514, dtype: int64
```

In [321…
```
df['rating'].value_counts()
```

Out[321]:
```
TV-MA       3214
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
NC-17          3
UR             3
Name: rating, dtype: int64
```

In [58]:
```
df['director'].value_counts()
```

Out[58]:
```
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
Name: director, Length: 4528, dtype: int64
```

In [316…
```
df['cast'].value_counts()
```

Out[316]:
```
David Attenborough
19
Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava,
Mousam, Swapnil
14
Samuel West
10
Jeff Dunham
7
David Spade, London Hughes, Fortune Feimster
6


..
Michael Peña, Diego Luna, Tenoch Huerta, Joaquin Cosio, José María Yazpi
k, Matt Letscher, Alyssa Diaz
1
Nick Lachey, Vanessa Lachey
1
Takeru Sato, Kasumi Arimura, Haru, Kentaro Sakaguchi, Takayuki Yamada, K
endo Kobayashi, Ken Yasuda, Arata Furuta, Suzuki Matsuo, Koichi Yamadera
, Arata Iura, Chikako Kaku, Kotaro Yoshida     1
Toyin Abraham, Sambasa Nzeribe, Chioma Chukwuka Akpotha, Chioma Omeruah,
Chiwetalu Agu, Dele Odule, Femi Adebayo, Bayray McNwizu, Biodun Stephen
1
Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghn
a Malik, Malkeet Rauni, Anita Shabdish, Chittaranjan Tripathy
1
Name: cast, Length: 7692, dtype: int64
```

In [55]:
```
df['duration'].value_counts()
```

Out[55]:
```
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
```

4. Visual Analysis - Univariate, Bivariate after pre-processing of the data

In [315…
```
df.head()
```

Out[315]:

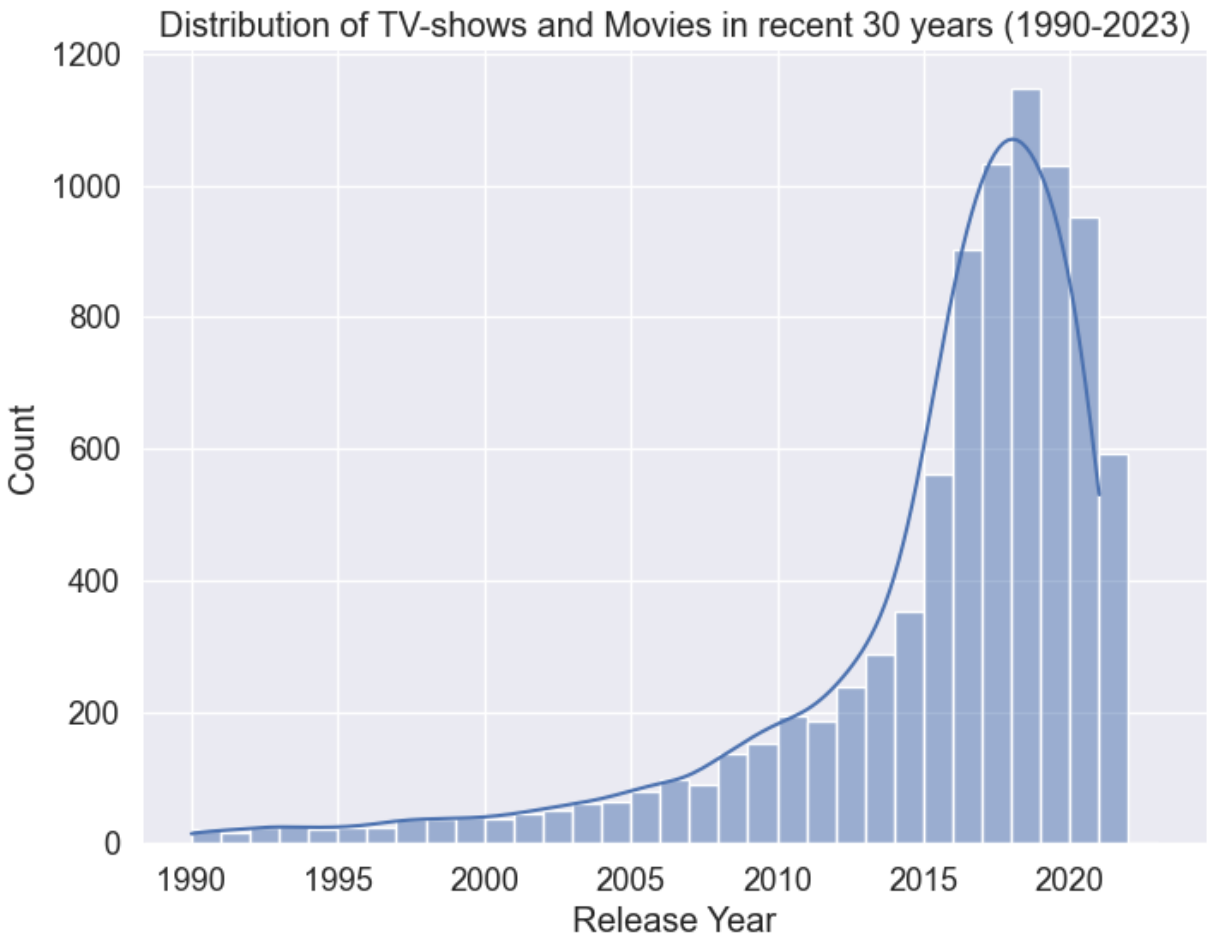
	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021

In [417…
```
release_years = df[df['release_year'].between(1990, 2023)]

# Define the bin range for the release year distplot
binrange = range(1990, 2024, 1)

# Create a distplot using Seaborn
sns.set_style('darkgrid')
fig, ax = plt.subplots(figsize=(8,6))
sns.histplot(release_years, x='release_year', kde=True, bins=binrange, ax
ax.set_xlabel('Release Year')
ax.set_ylabel('Count')
ax.set_title('Distribution of TV-shows and Movies in recent 30 years (199
sns.despine(left=True, bottom=True)

plt.show()
```



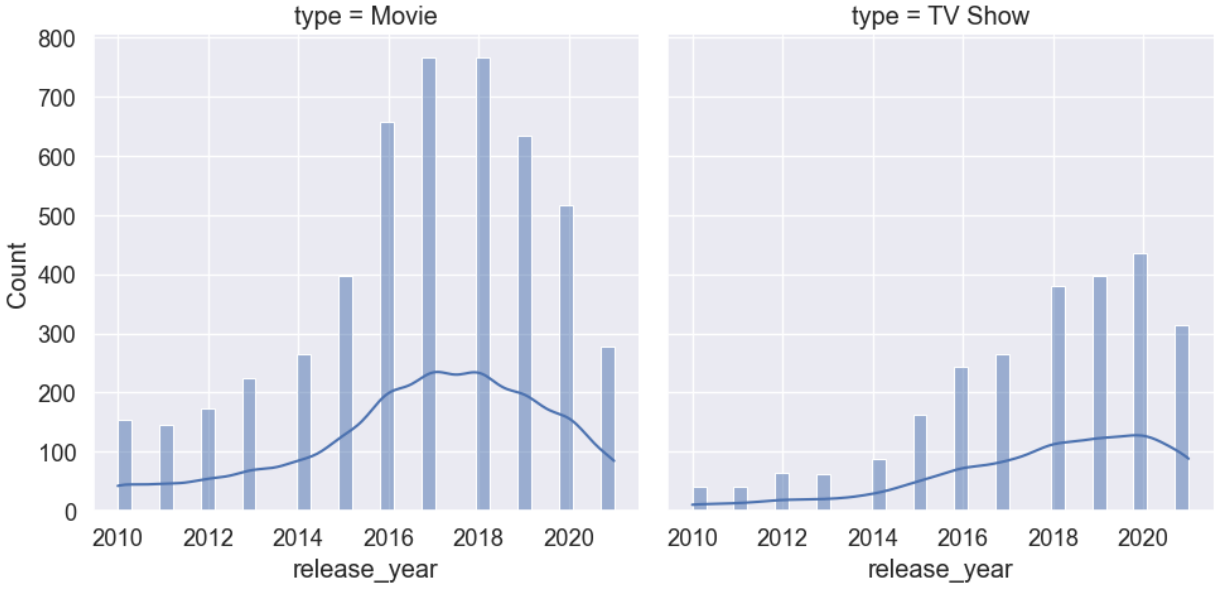
Comparison b/w TV-shows and Movies in recent years

In [583…
```
def comp(df):
    return df[df['release_year'].between(2010,2023)]
```

In [584…
```
selected_years = comp(df)
```

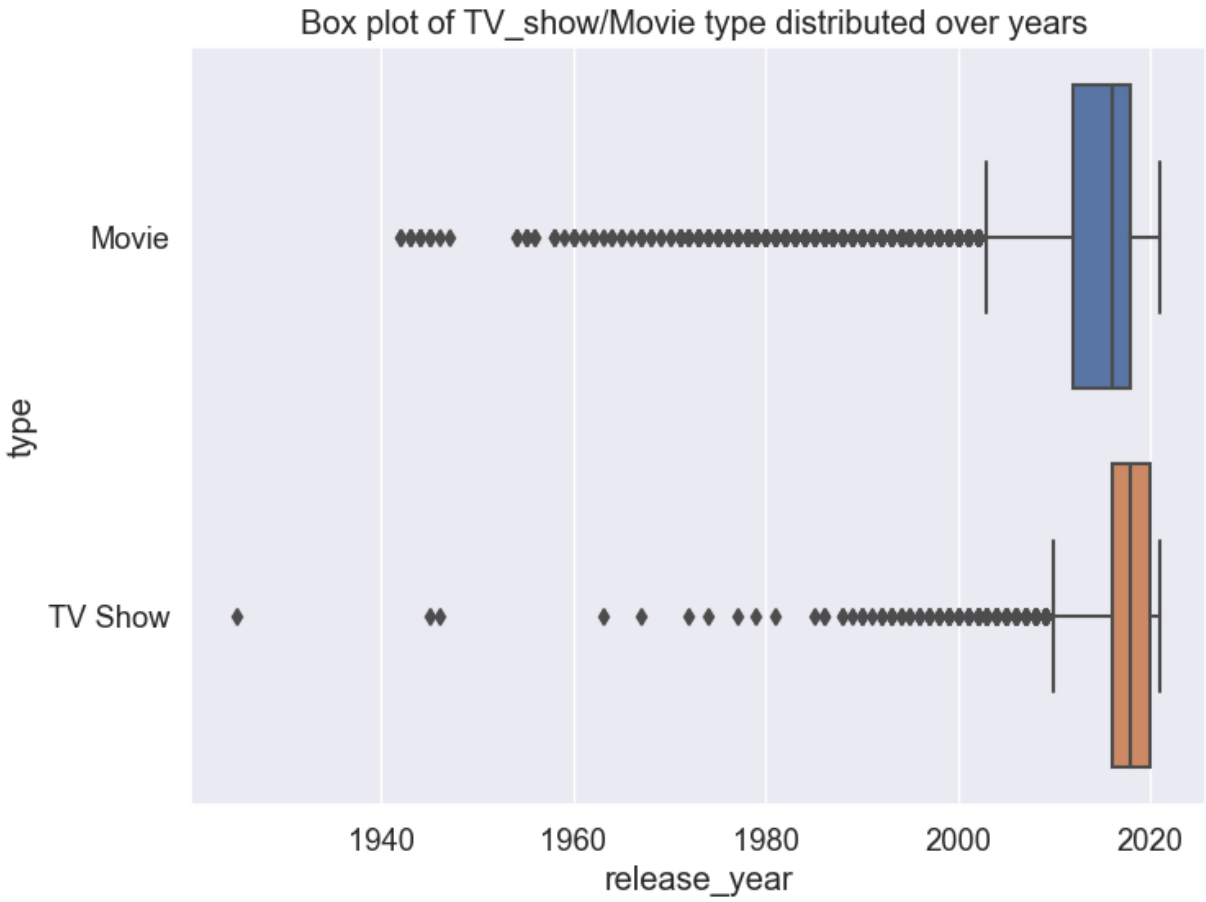
In [887…
```
plt.figure (figsize=(8,6))
sns.displot(data=selected_years, x="release_year", col="type", kde=True)
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



In [575…
```
TV_genre_x=selected_years[selected_years['type']=='TV Show']
Movie_genre_z=selected_years[selected_years['type']=='Movie']
```

In [884…
```
sns.boxplot(x=df['release_year'], y=df['type'],data=df)
plt.title('Box plot of TV_show/Movie type distributed over years ')
plt.show()
```



In [ ]:

In [ ]:
```
## Pre-processing involves unnesting of the data in columns like Actor, D
```

In [109…
```
df1=df.copy() #Making copy dataframe to extract the nested values in the
```

Top 20 actors who contribuited more number of TV_shows/movies in Netflix

In [110…
```
df1['actor']=df1['cast'].str.split(', ')
```

In [111…
```
df1=df1.explode(['actor'], ignore_index=False)
```

In [112…
```
df1.head() # We have new column at last named 'actor' where the nested va
```

Out[112]:

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

In [165...

```
top_actors=df1['actor'].value_counts()[0:20] #Top 20 actors/cast who cont
```

In [170...

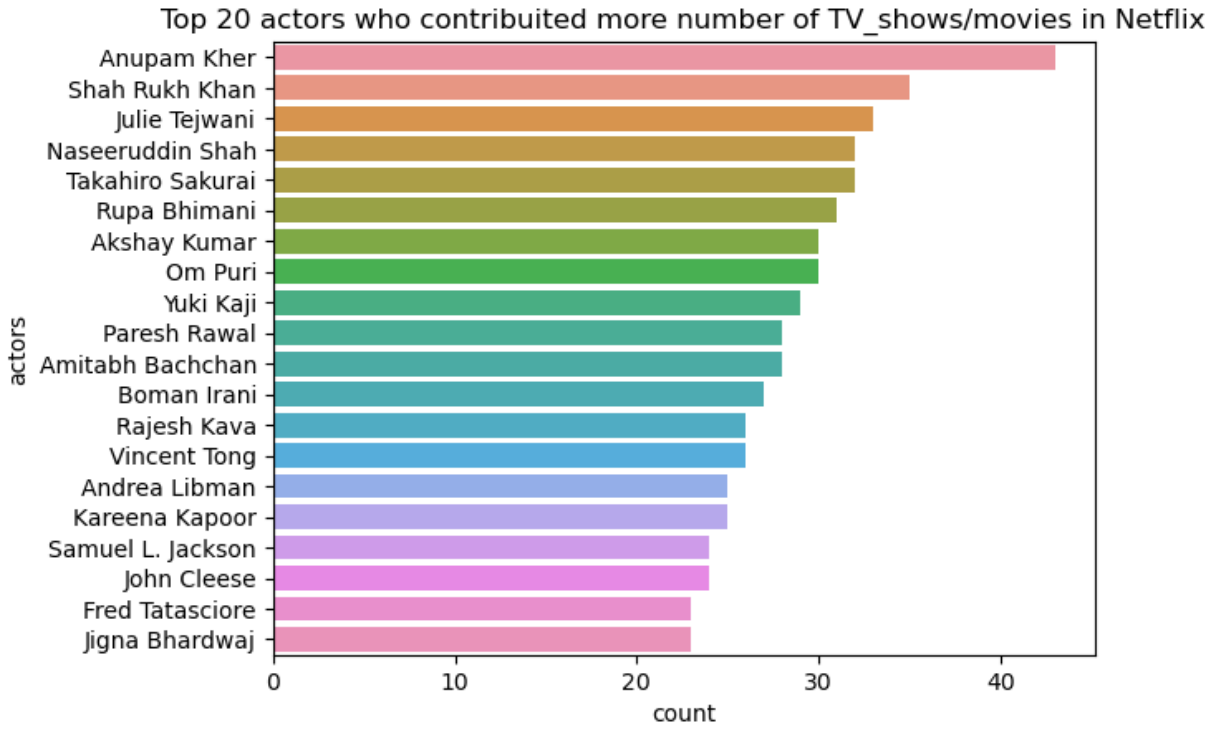
```
top_actors=top_actors.reset_index()
```

In [171...

```
top_actors.columns=['actors','count']
```

In [357...

```
sns.barplot(data=top_actors, y='actors', x='count')
plt.title('Top 20 actors who contribuited more number of TV_shows/movies
plt.show()
```



Top 10 Director-Cast Pairs with the Highest Number of Titles

In [326…
```
df1 = df1.assign(director=df1.director.str.split(', ')).explode('director
```

In [338…
```
director_actor_pairs = df1.groupby(['director', 'actor'])['title'].count(
```

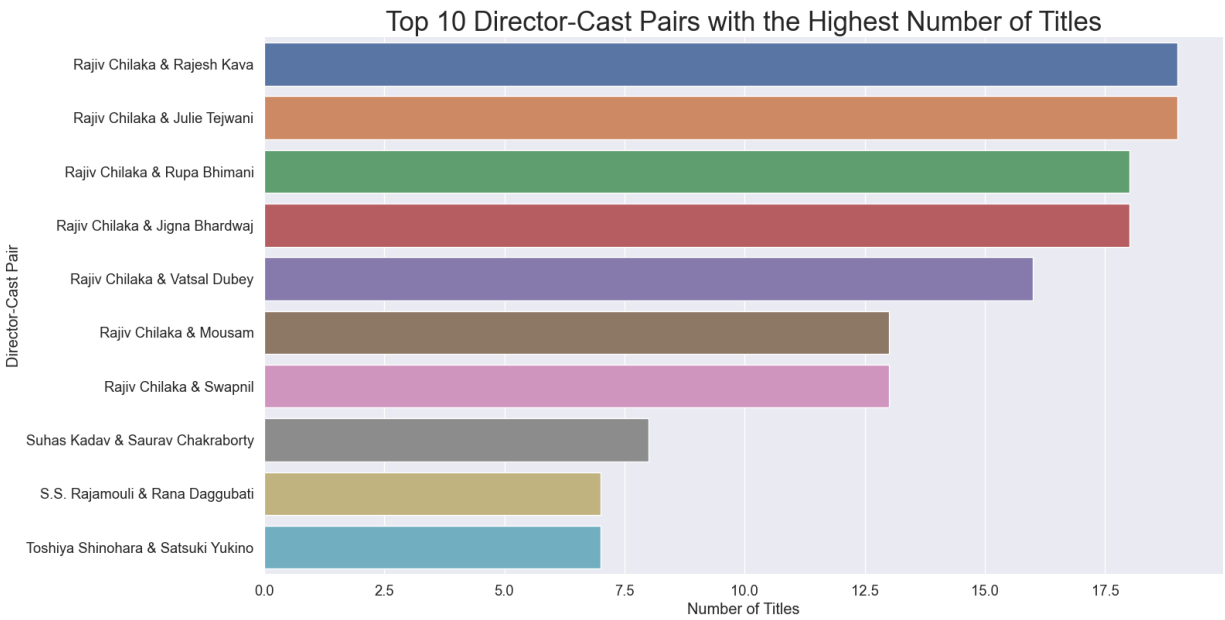
In [341…
```
top_pairs = director_actor_pairs.sort_values(ascending=False)[:10] #Top 1
```

In [342…
```
top_pairs
```

Out[342]:
```
director           actor
Rajiv Chilaka      Rajesh Kava          19
                   Julie Tejwani        19
                   Rupa Bhimani         18
                   Jigna Bhardwaj       18
                   Vatsal Dubey         16
                   Mousam               13
                   Swapnil              13
Suhas Kadav        Saurav Chakraborty    8
S.S. Rajamouli     Rana Daggubati        7
Toshiya Shinohara  Satsuki Yukino        7
Name: title, dtype: int64
```

In [376…
```
sns.set_style('darkgrid')
fig, ax = plt.subplots(figsize=(16,9))
sns.barplot(x=top_pairs.values, y=top_pairs.index.map(lambda x: f'{x[0]}
ax.set_xlabel('Number of Titles')
ax.set_ylabel('Director-Cast Pair')
ax.set_title('Top 10 Director-Cast Pairs with the Highest Number of Title
sns.despine(left=True, bottom=True)

plt.show()
```



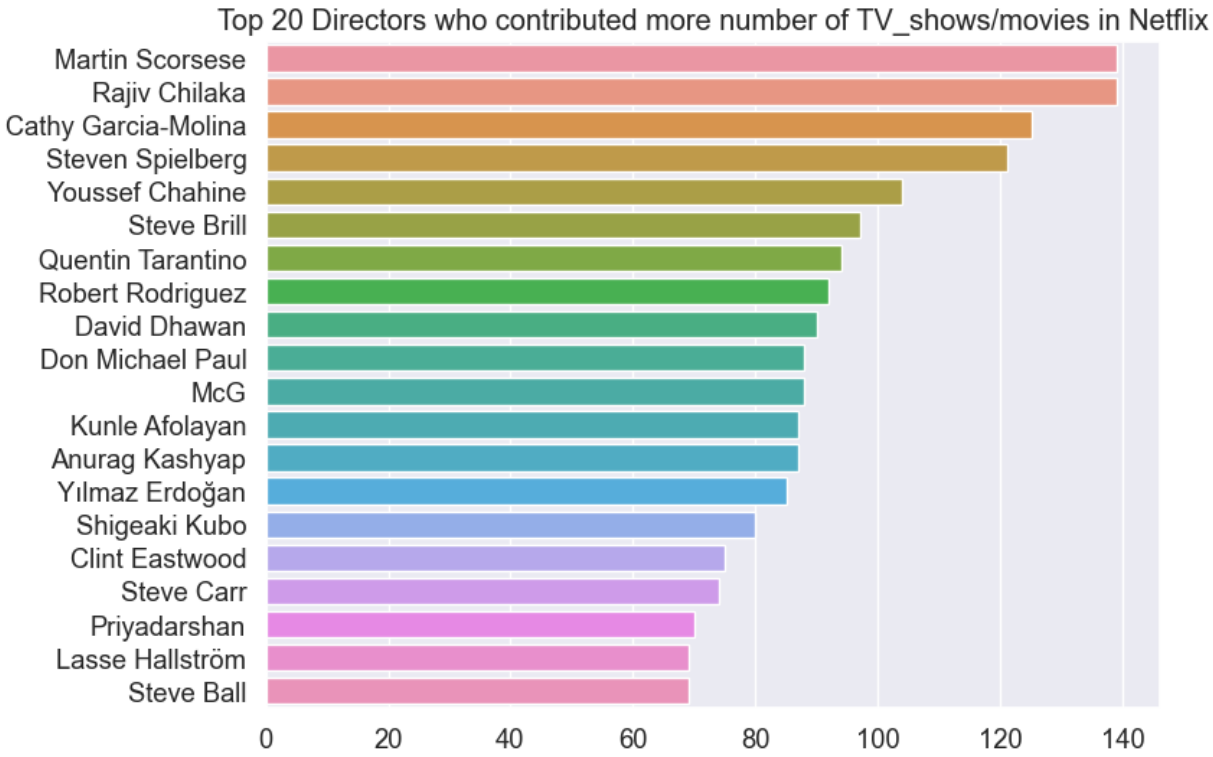
Top 20 Directors who contribuited more number of TV_shows/movies in Netflix

In [379…
```
top_directors=df1['director'].value_counts()[:20]
```

In [387…
```
top_directors #Top Directors who has produced highest number of content i
```

Out[387]:
```
Martin Scorsese         139
Rajiv Chilaka           139
Cathy Garcia-Molina     125
Steven Spielberg        121
Youssef Chahine         104
Steve Brill              97
Quentin Tarantino        94
Robert Rodriguez         92
David Dhawan             90
Don Michael Paul         88
McG                      88
Kunle Afolayan           87
Anurag Kashyap           87
Yılmaz Erdoğan           85
Shigeaki Kubo            80
Clint Eastwood           75
Steve Carr               74
Priyadarshan             70
Lasse Hallström          69
Steve Ball               69
Name: director, dtype: int64
```

In [892…
```
sns.barplot(x=top_directors.values, y=top_directors.index)
plt.title('Top 20 Directors who contributed more number of TV_shows/movie
plt.show()
```



Top 20 Countries which contribuited highest content

In [600…
```
df_country=df.copy() #making a copy to unnest column country and explodin
```

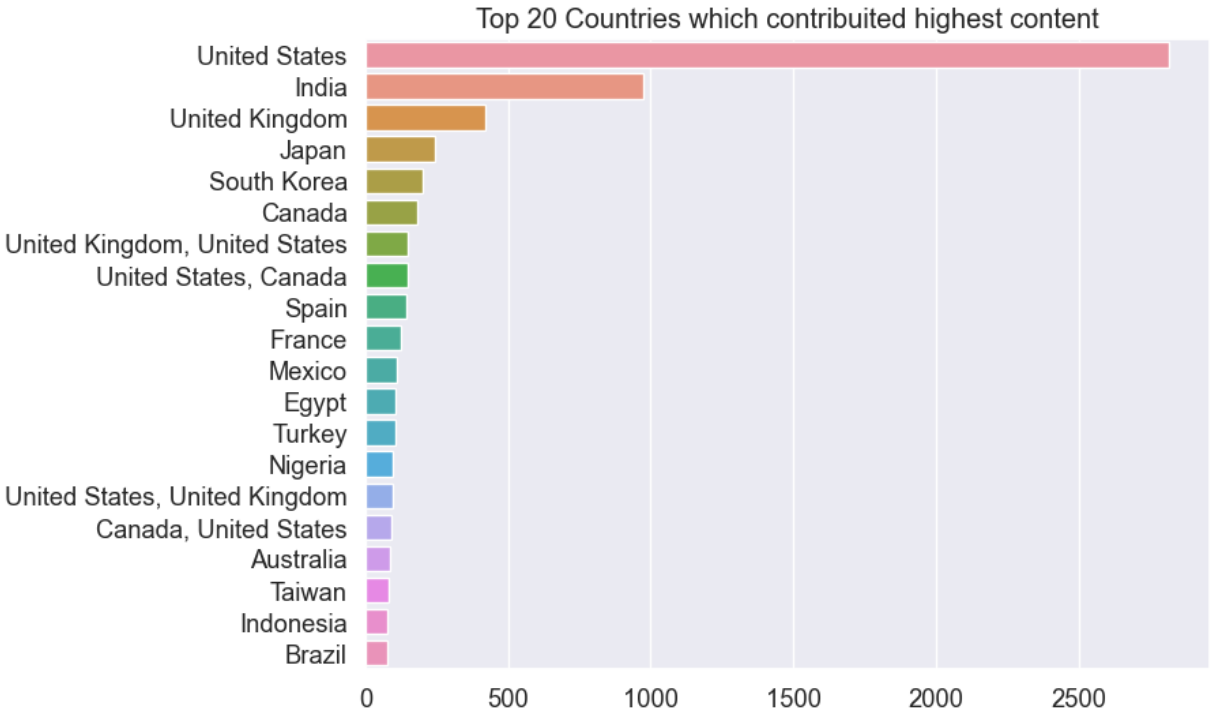
In [601…
```
df_country = df_country.assign(director=df_country.country.str.split(', '
```

In [602…
```
df_country['country'].value_counts() # Highest number of countries that h
```

Out[602]:
```
United States     2818
India              972
United Kingdom     419
Japan              245
South Korea        199
                  ...
Namibia              1
Poland,              1
Zimbabwe             1
Mozambique           1
Georgia              1
Name: country, Length: 748, dtype: int64
```

In [603…
```
top_countries=df_country['country'].value_counts()[:20] # Top 20 countrie
```

In [606…
```
sns.barplot(x=top_countries.values, y=top_countries.index)
plt.title('Top 20 Countries which contribuited highest content')
plt.show()
```



Analysis of Top 20 Genre which contribuited highest content

In [607…
```
df_genre=df.copy()
```

In [608…
```
df_genre['genre']=df_genre['listed_in'].str.split(', ')
```

In [609…
```
df_genre=df_genre.explode(['genre'], ignore_index=False)
```

In [611…
```
df_genre.head()
```

Out[611]:

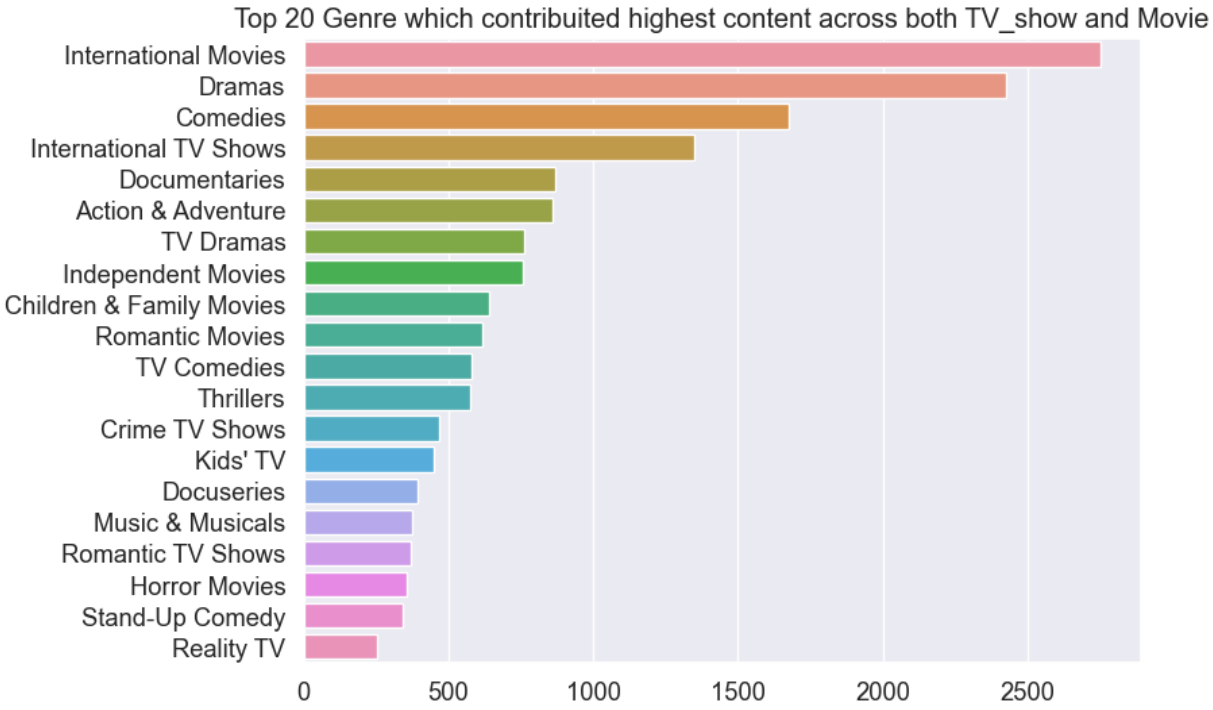
	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi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

In [614...

```
top_genre=df_genre['genre'].value_counts()[0:20]
```

In [898...

```
sns.barplot(x=top_genre.values, y=top_genre.index)
plt.title('Top 20 Genre which contribuited highest content across both TV
plt.show()
```



Most Popular Genres of TV Shows and Movies on Netflix

In [751...
```
tv_mov_genres=df_genre.groupby(['type'])['genre'].value_counts() #Groupby
```

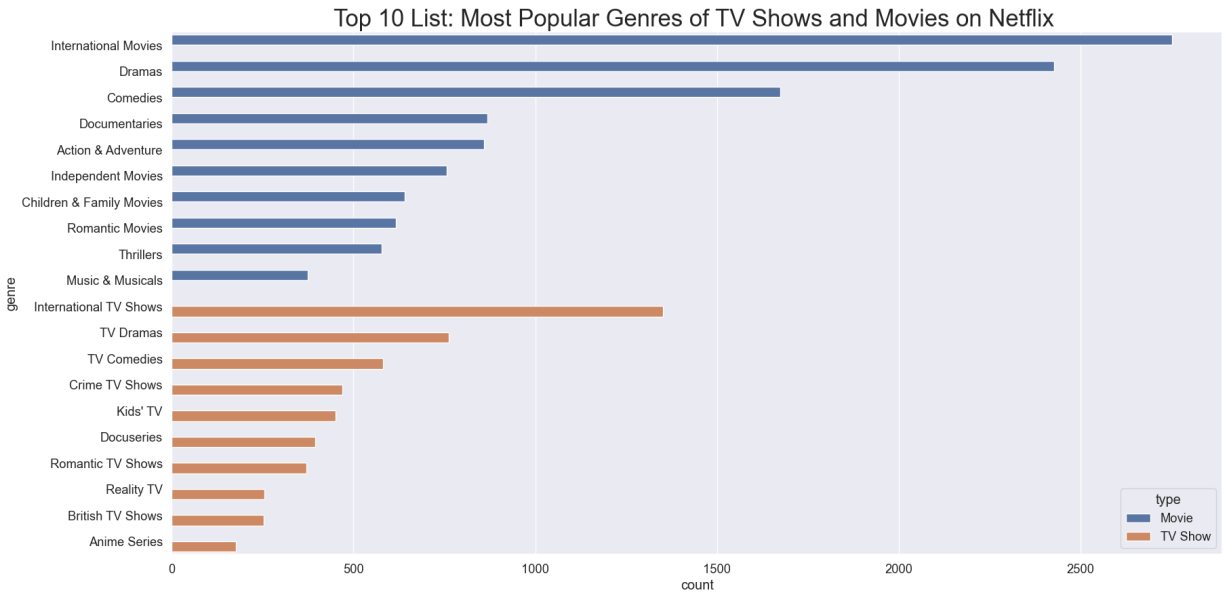
In [752...
```
tv_mov_genres=tv_mov_genres.to_frame()
```

In [753...
```
tv_mov_genres.columns=['count']
```

In [754...
```
tv_mov_genres=tv_mov_genres.reset_index(level=[1])
```

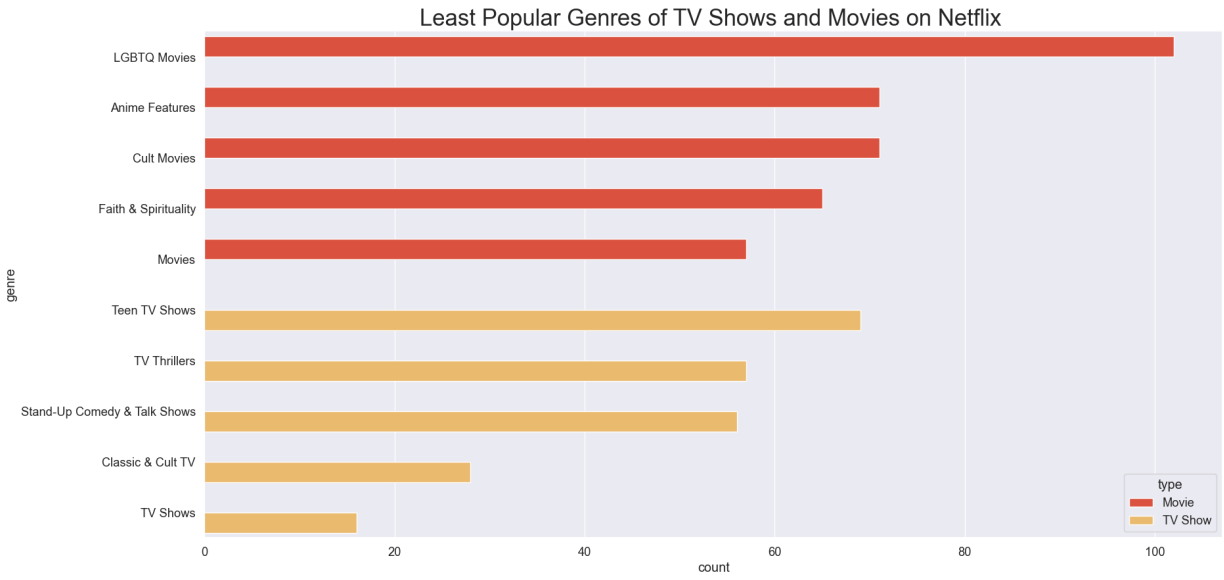
In [755...
```
tv_mov_genres_top=tv_mov_genres.groupby('type').head(10) #Taking top 10 G
```

In [757...
```
plt.figure(figsize=(20,10))
sns.barplot(y=tv_mov_genres_top['genre'], x=tv_mov_genres_top['count'], h
plt.title('Top 10 List: Most Popular Genres of TV Shows and Movies on Net
plt.show()
```



In [760…
```
tv_mov_genres_low=tv_mov_genres.groupby('type').tail(5) #Least genre wher
```

In [788…
```
plt.figure(figsize=(20,10))
sns.set_style('darkgrid')
sns.barplot(y=tv_mov_genres_low['genre'], x=tv_mov_genres_low['count'], h
plt.title('Least Popular Genres of TV Shows and Movies on Netflix ', font
plt.show()
```

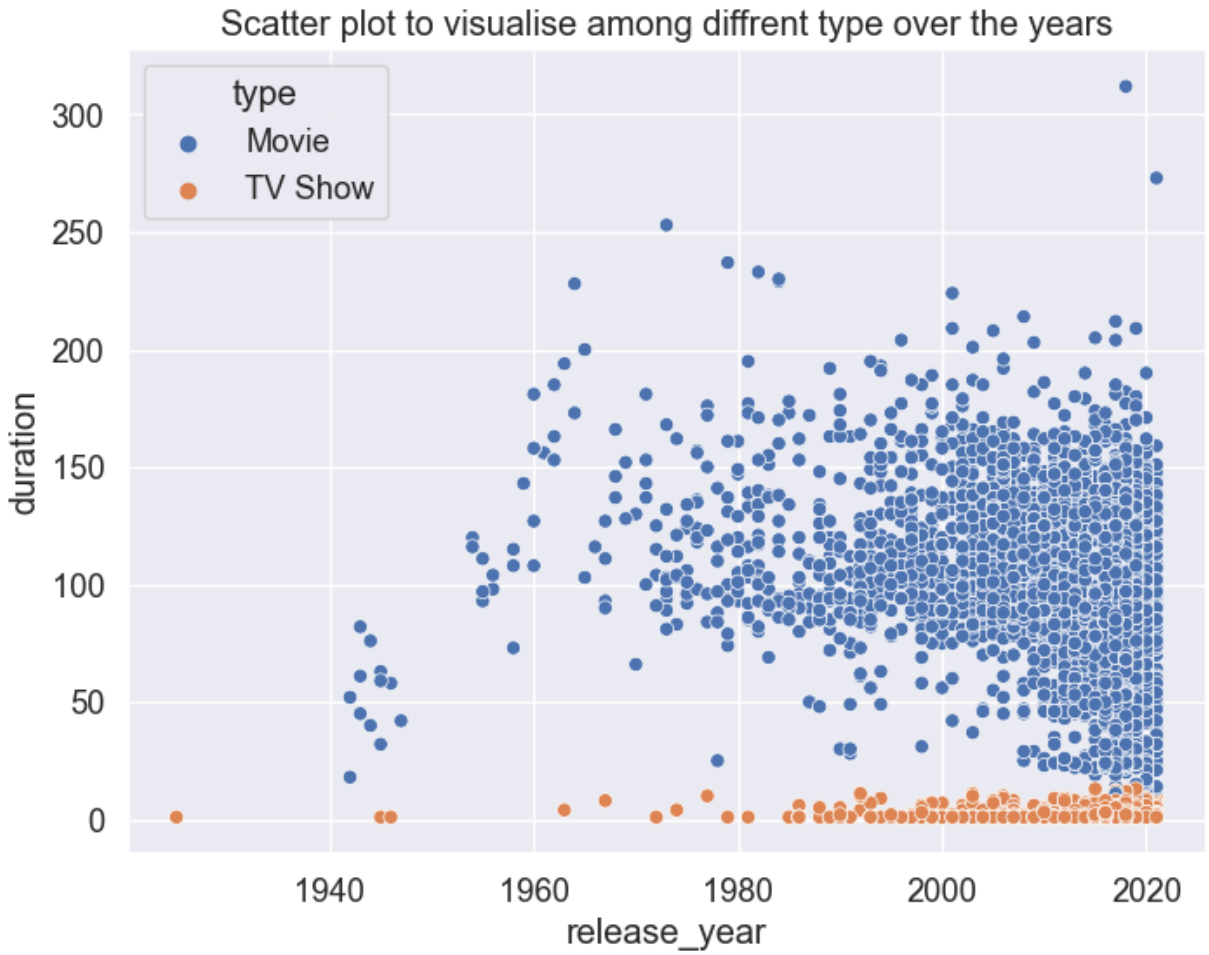


In [ ]:
```
```

Graphical anlaysis of how duration in Movies and TV-shows is distributed

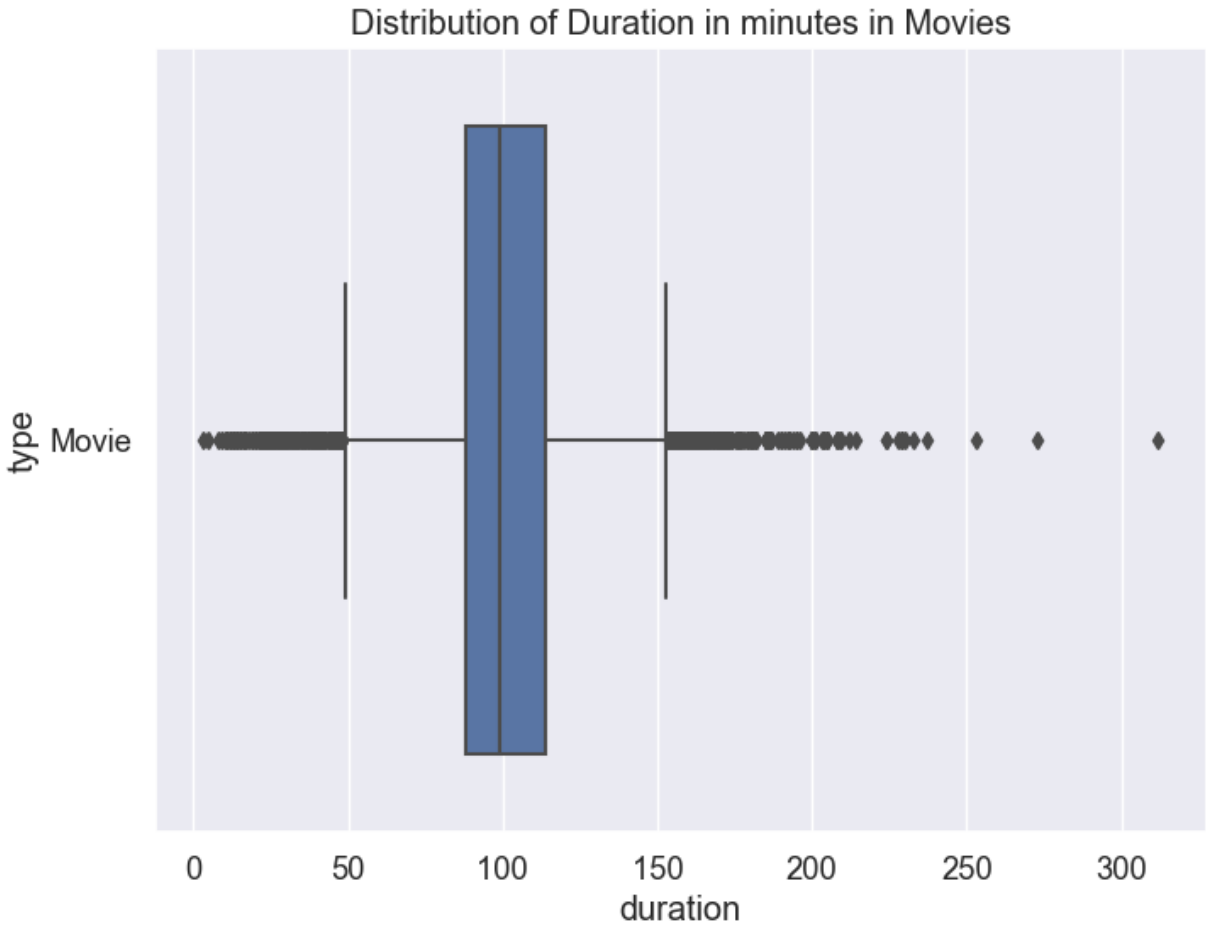
In [828…
```
df_country['duration'] = df_country['duration'].str.extract('(\d+)', expa
```

In [881…
```
sns.scatterplot(data=df_country, x='release_year', y='duration', hue='typ
plt.title('Scatter plot to visualise among diffrent type over the years')
plt.show()
```



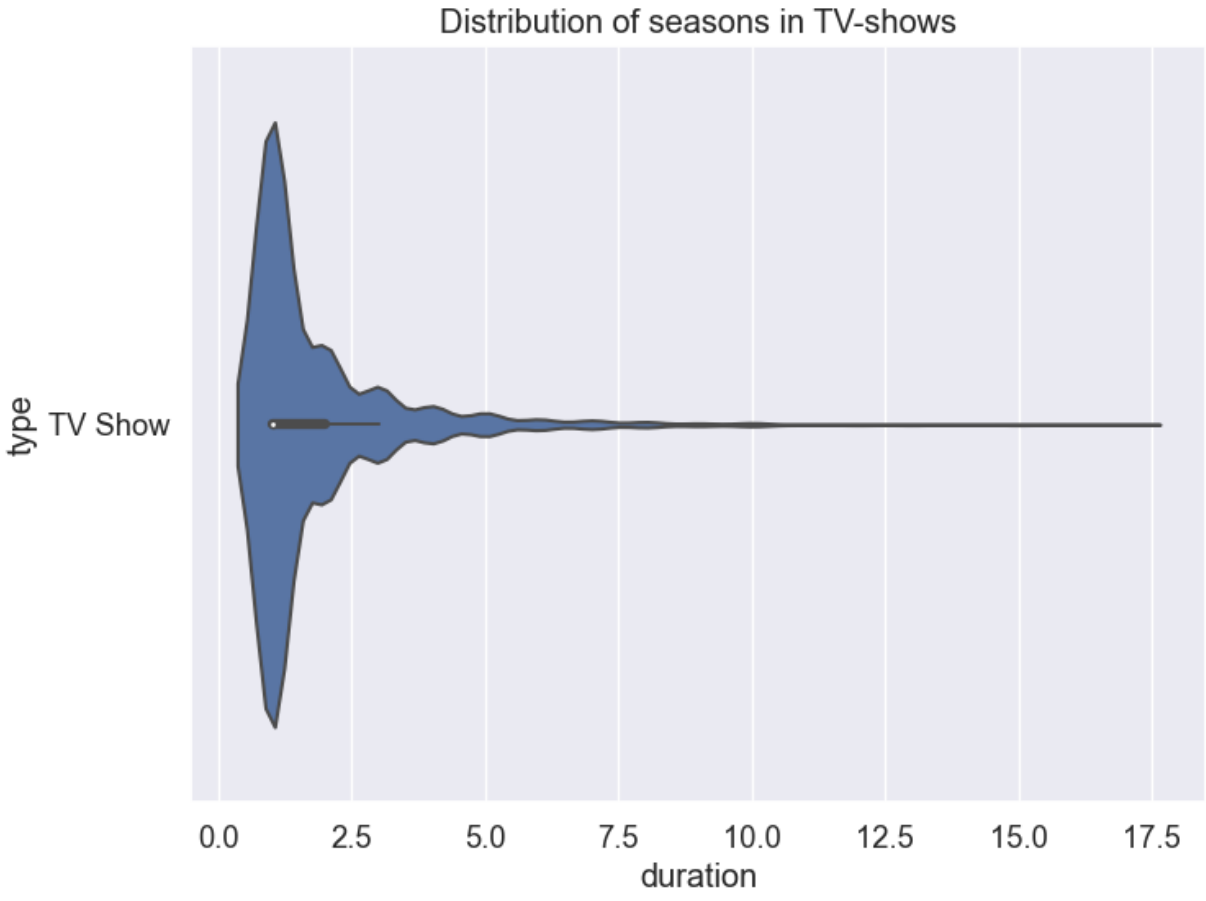
In [861…
```
TV_genre=df_country[df_country['type']=='TV Show']
Movie_genre=df_country[df_country['type']=='Movie'] #Distribution of dura
```

In [895…
```
sns.boxplot(data=Movie_genre, y='type', x='duration')
plt.title('Distribution of Duration in minutes in Movies')
plt.show()
```



In [897…

```
sns.violinplot(data=TV_genre, y='type', x='duration')
plt.title('Distribution of seasons in TV-shows')
plt.show()
```



In [ ]:

Business Insights:

Based on the analysis of the Netflix dataset, we can draw the following business insights:

1. Content Distribution: The analysis shows that 70% of the content on Netflix is Movies and 30% is TV shows. This indicates that the majority of the user base on Netflix prefers to watch movies over TV shows. Netflix should focus on producing more original movies to cater to the audience's preference.

3. Geographical Targeting: The analysis shows that most of the content is produced by the United States, followed by India, UK, Japan, and South Korea. This suggests that Netflix should focus on these regions to acquire more content and increase its user base.

4. Growth Strategy: The analysis shows that in 2018, Netflix released a lot more content than other years, and the growth has happened in recent years. This indicates that Netflix's strategy to produce more original content is paying off, and the company should continue to focus on producing more original content to drive user growth.

5. Popular Genres: The analysis shows that International Movies and International TV shows are the most popular genres on Netflix. Additionally, Dramas, Comedies, Documentaries, and Action and Adventure are also popular across both movies and TV shows. Netflix should produce more content in these genres to cater to the preferences of its users.

6. Least Popular Genres: The analysis shows that LGBTQ Movies, Anime Features, Cult Movies, Faith and Spirituality in Movies, Teen TV shows, TV thrillers, Talk shows, classic, and cult TV are the least popular genres on Netflix. Netflix should limit producing content in these genres and focus on more popular genres to attract and retain its users.

7.Content Duration: The analysis shows that most movies range from 90 minutes to 120 minutes, and most TV shows have 1 season to 3 seasons. Netflix should produce more

content that fits these time durations to meet the user preferences.

8. Top Actors and Directors: The analysis shows that Anupam kher, Shah rukh khan, Julie Tejwani, Naseerudin shah, and Takahiro sakurai are the top actors, and Martin Scorsese, Rajiv Chilaka, Cathy Garcia-Molina, Steven Spielberg, and Youssef Chahine are the top directors that have the highest number of TV shows/Movies on Netflix. Netflix can collaborate with these actors and directors to produce more original content and attract a larger audience base.

9. TV Shows vs Movies: The analysis shows that there are more TV shows than movies in recent years. This suggests that Netflix is focusing more on producing original TV shows to cater to the preferences of its user base.

10. Based on these insights, Netflix should continue to focus on producing more original content, particularly in popular genres, for its audience. The company should also collaborate with popular actors and directors to produce more original content and attract a larger user base. Finally, the company should focus more on producing original TV shows to cater to the preferences of its user base, but also produce more movies to balance its content distribution.

Business Recommendation:

Increase the production of TV shows as they are more in demand in recent years.

Focus on producing content in countries where there is a growing demand, such as India, UK, Japan, and South Korea.

Continue to focus on popular genres such as International TV-shows and Movies, Dramas, Comedies, Documentaries, and Action and Adventure.

Consider working with top actors and top directors to increase view time.

Consider producing more seasons for TV shows to keep audiences engaged.

Increase the production of content in international markets to cater to the growing demand for International content.

Continue to produce content that resonates with audiences and cater to their preferences, to increase customer loyalty and retention.

Consider producing content in multiple languages to cater to a more diverse audience.

Collaborate with local artists and filmmakers to produce region-specific content that resonates with local audiences.

Invest in personalization algorithms to help customers find content that suits their interests.

In [ ]: