

projectncp

July 10, 2023

1 NETFLIX CASE STUDY PROJECT

ABHISHEK SHETE

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

```
[57]: data = pd.read_csv('original_netflix.csv')
data
```

```
[57]:
```

	show_id	type	title	director \
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson
1	s2	TV Show	Blood & Water	NaN
2	s3	TV Show	Ganglands	Julien Leclercq
3	s4	TV Show	Jailbirds New Orleans	NaN
4	s5	TV Show	Kota Factory	NaN
...
8802	s8803	Movie	Zodiac	David Fincher
8803	s8804	TV Show	Zombie Dumb	NaN
8804	s8805	Movie	Zombieland	Ruben Fleischer
8805	s8806	Movie	Zoom	Peter Hewitt
8806	s8807	Movie	Zubaan	Mozez Singh

	cast	country \
0	NaN	United States
1	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa
2	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN
3	NaN	NaN
4	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India
...
8802	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States
8803	NaN	NaN
8804	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States
8805	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States
8806	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India

	date_added	release_year	rating	duration	\
0	September 25, 2021	2020	PG-13	90 min	
1	September 24, 2021	2021	TV-MA	2 Seasons	
2	September 24, 2021	2021	TV-MA	1 Season	
3	September 24, 2021	2021	TV-MA	1 Season	
4	September 24, 2021	2021	TV-MA	2 Seasons	
...	
8802	November 20, 2019	2007	R	158 min	
8803	July 1, 2019	2018	TV-Y7	2 Seasons	
8804	November 1, 2019	2009	R	88 min	
8805	January 11, 2020	2006	PG	88 min	
8806	March 2, 2019	2015	TV-14	111 min	

	listed_in	\
0	Documentaries	
1	International TV Shows, TV Dramas, TV Mysteries	
2	Crime TV Shows, International TV Shows, TV Act...	
3	Docuseries, Reality TV	
4	International TV Shows, Romantic TV Shows, TV ...	
...	...	
8802	Cult Movies, Dramas, Thrillers	
8803	Kids' TV, Korean TV Shows, TV Comedies	
8804	Comedies, Horror Movies	
8805	Children & Family Movies, Comedies	
8806	Dramas, International Movies, Music & Musicals	

	description
0	As her father nears the end of his life, filmm...
1	After crossing paths at a party, a Cape Town t...
2	To protect his family from a powerful drug lor...
3	Feuds, flirtations and toilet talk go down amo...
4	In a city of coaching centers known to train I...
...	...
8802	A political cartoonist, a crime reporter and a...
8803	While living alone in a spooky town, a young g...
8804	Looking to survive in a world taken over by zo...
8805	Dragged from civilian life, a former superhero...
8806	A scrappy but poor boy worms his way into a ty...

[8807 rows x 12 columns]

```
[26]: data.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

```

```
[27]: data.shape
```

```
[27]: (8807, 12)
```

```
[ ]:
```

```
[ ]:
```

```
[28]: data.columns
```

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

```
[29]: data['rating'].unique()
```

```
[29]: array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
         'TV-G', 'G', 'NC-17', '74 min', '84 min', '66 min', 'NR', nan,
         'TV-Y7-FV', 'UR'], dtype=object)
```

above we observed unique rating

```
[30]: data.describe()
```

```
[30]:      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
```

below we have described content released between year 1925 and 2021.

```
[31]: data['release_year'].describe()
```

```
[31]: count      8807.000000
      mean       2014.180198
      std         8.819312
      min       1925.000000
      25%       2013.000000
      50%       2017.000000
      75%       2019.000000
      max       2021.000000
      Name: release_year, dtype: float64
```

```
[32]: data.isnull().sum()
```

```
[32]: 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
```

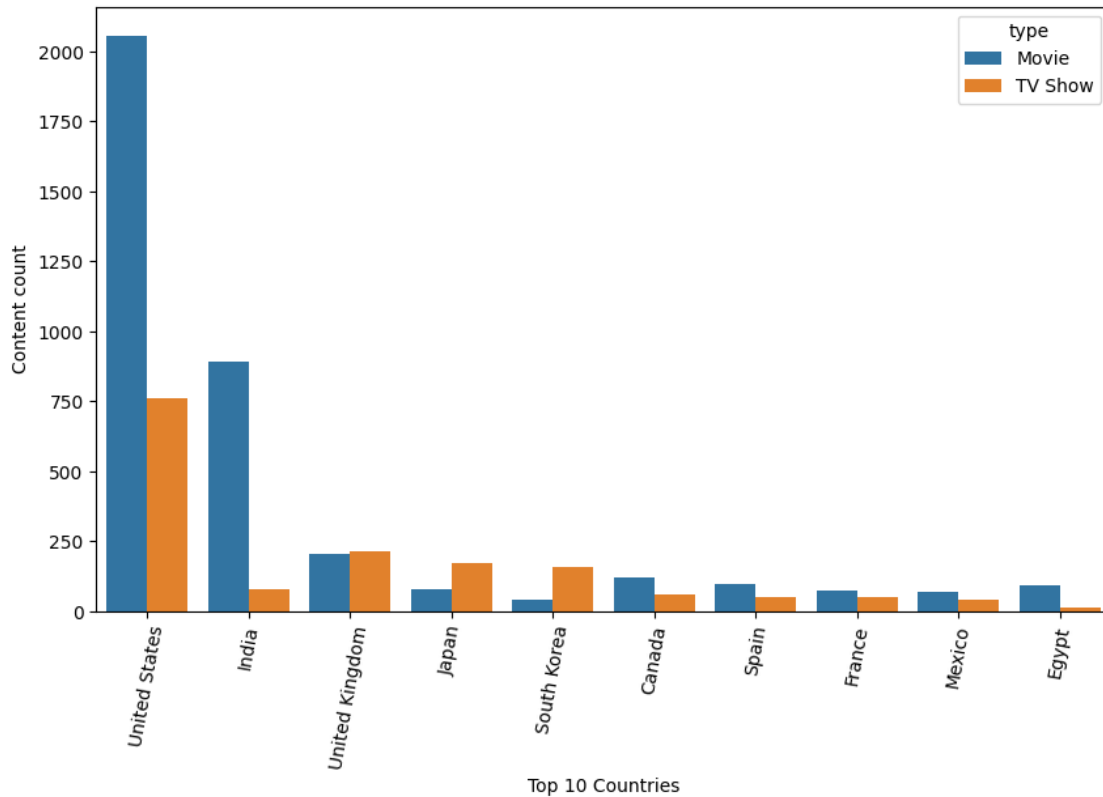
```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[33]: plt.figure(figsize = [10,6])
      ax = sns.countplot(data=data,x='country', order = data['country'].
      ↪value_counts().head(10).index, hue = 'type')
      plt.xticks(rotation =80)
      ax.set(xlabel='Top 10 Countries', ylabel='Content count')
```

```
[33]: [Text(0.5, 0, 'Top 10 Countries'), Text(0, 0.5, 'Content count')]
```



[]:

Above we can see the top 10 countries which based on the content count

1. From above bar graph that shows the United states having highest content count followed by India despite Egypt has lowest content count.
2. Only Japan and South Korea has more interest in TV shows than movies
3. european countries like spain and france having equal TV show content count.

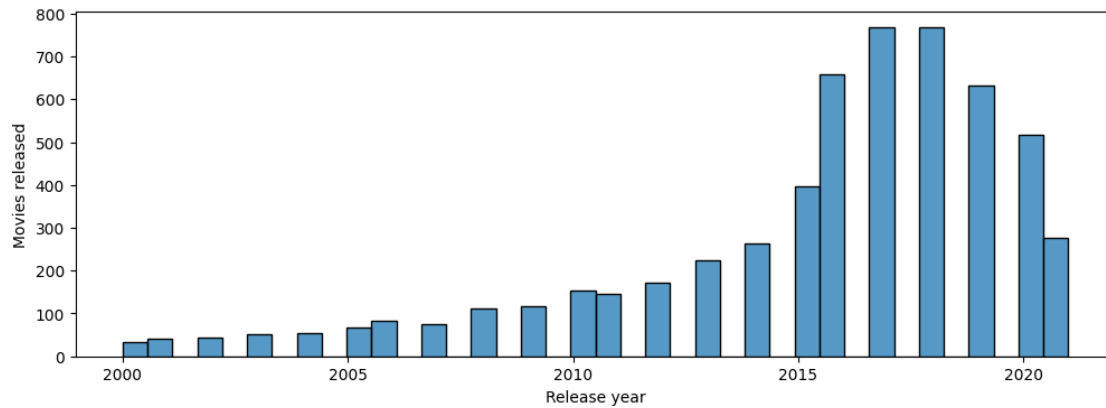
[]:

[]:

[]:

```
[34]: data['country'].isna().sum()
after2000 = data[(data['release_year'] >= 2000) & (data['type'] == 'Movie')]
plt.figure(figsize = (12,4))
ax= sns.histplot(data = after2000, x= 'release_year')
ax.set(xlabel='Release year', ylabel='Movies released')
```

```
[34]: [Text(0.5, 0, 'Release year'), Text(0, 0.5, 'Movies released')]
```



Above we can see movies released between year 2000 to 2020.

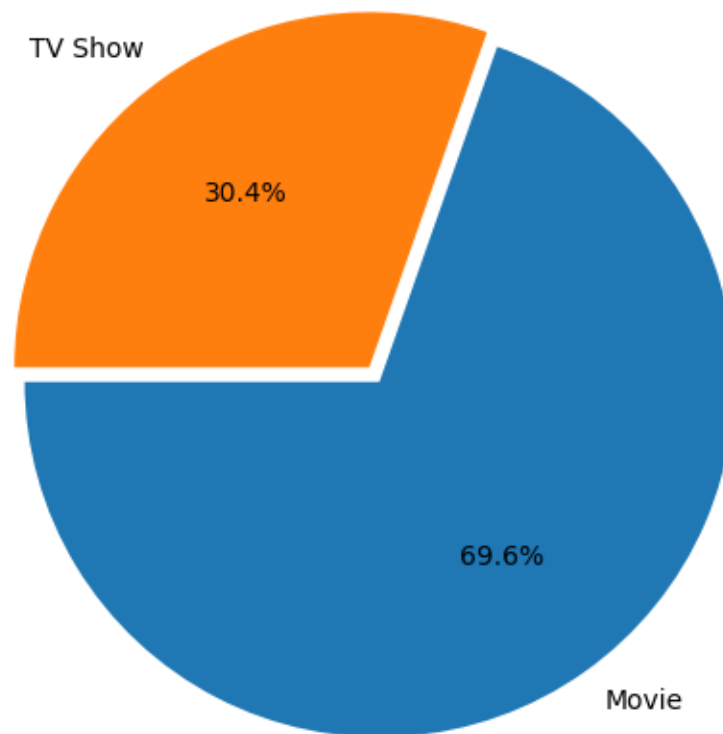
```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[35]: plt.figure(figsize=(12,6))
plt.title("Percontation of Netflix Titles that are either Movies or TV Shows")
g = plt.pie(data.type.value_counts(),explode=(0.025,0.025), labels=data.type.
↪value_counts().index,autopct='%1.1f%%', startangle=180)
plt.show()
```

Percentation of Netflix Titles that are either Movies or TV Shows



TV shows contributed for the 30% whereas Movies contributed to around 70% of total content on Netflix.

[]:

[]:

[]:

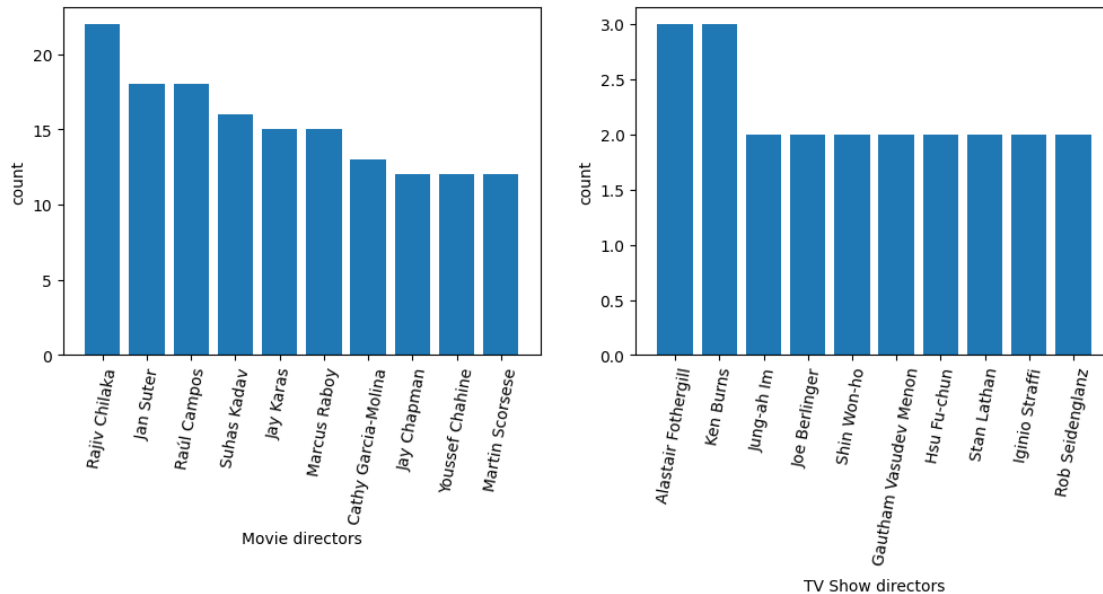
```
[36]: plt.figure(figsize = (12,4))

movie_directors = data.loc[data['type'] == 'Movie']['director'].str.
    ↪split(',',expand=True).stack()
plt.subplot(1, 2, 1)
plt.bar(movie_directors.value_counts().head(10).index, movie_directors.
    ↪value_counts().head(10).values)
plt.xticks(rotation = 80)
```

```
plt.xlabel("Movie directors")
plt.ylabel("count")

tvshow_directors = data.loc[data['type'] == 'TV Show']['director'].str.
    ↪split(',',expand=True).stack()
tvshow_directors.value_counts()
plt.subplot(1, 2, 2)
plt.bar(tvshow_directors.value_counts().head(10).index, tvshow_directors.
    ↪value_counts().head(10).values)
plt.xticks(rotation = 80)
plt.xlabel("TV Show directors")
plt.ylabel("count")
```

[36]: Text(0, 0.5, 'count')



Above are the top 10 directors for Movies and TV Shows.

[]:

[]:

```
[37]: plt.figure(figsize = (12,4))

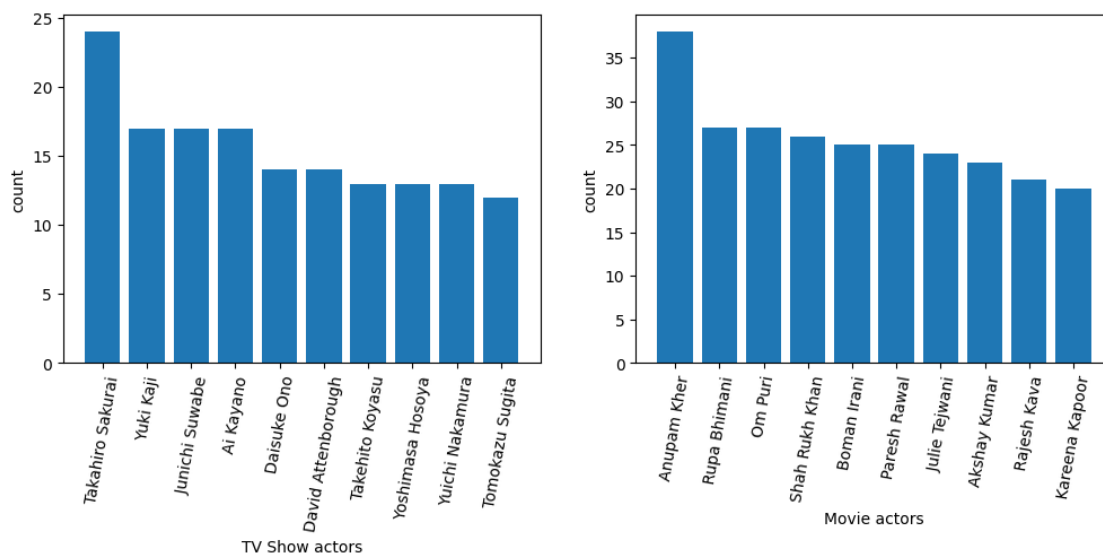
tvshow_actors = data.loc[data['type'] == 'TV Show']['cast'].str.
    ↪split(',',expand=True).stack()
tvshow_actors.value_counts()
plt.subplot(1,2,1)
```



```
plt.bar(tvshow_actors.value_counts().head(10).index, tvshow_actors.
    ↪value_counts().head(10).values)
plt.xticks(rotation =80)
plt.xlabel("TV Show actors")
plt.ylabel("count")

movie_actors = data.loc[data['type'] == 'Movie']['cast'].str.
    ↪split(',',expand=True).stack()
movie_actors.value_counts()
plt.subplot(1,2,2)
plt.bar(movie_actors.value_counts().head(10).index, movie_actors.value_counts().
    ↪head(10).values)
plt.xticks(rotation =80)
plt.xlabel("Movie actors")
plt.ylabel("count")
```

[37]: Text(0, 0.5, 'count')



Above are the top 10 actors for Movies and TV Shows.

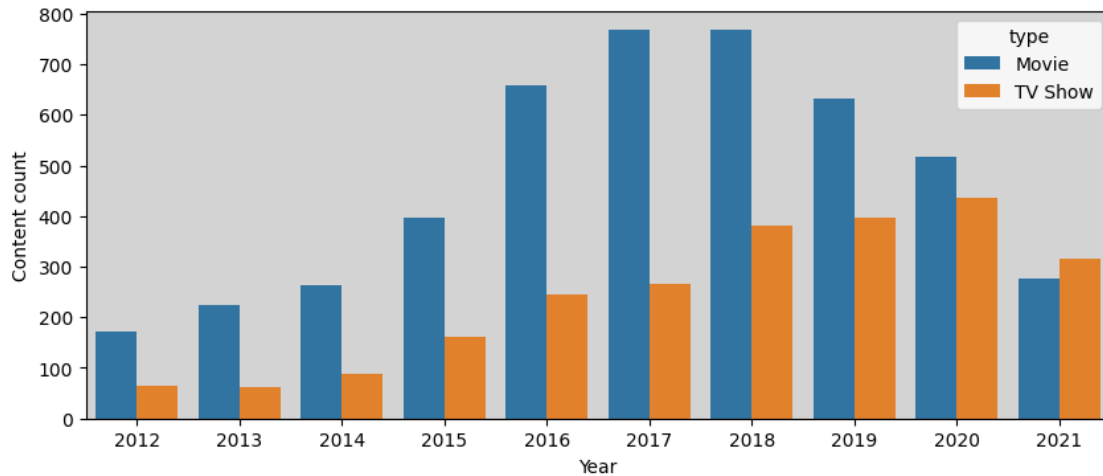
[]:

[]:

```
[38]: release_year_analysis = data.loc[data['release_year'] >=2012]
plt.figure(figsize = [10,4])
ax = sns.countplot(release_year_analysis, x='release_year', hue='type')
ax.set_facecolor("lightgrey")
```

```
ax.set(xlabel='Year', ylabel='Content count')
```

```
[38]: [Text(0.5, 0, 'Year'), Text(0, 0.5, 'Content count')]
```



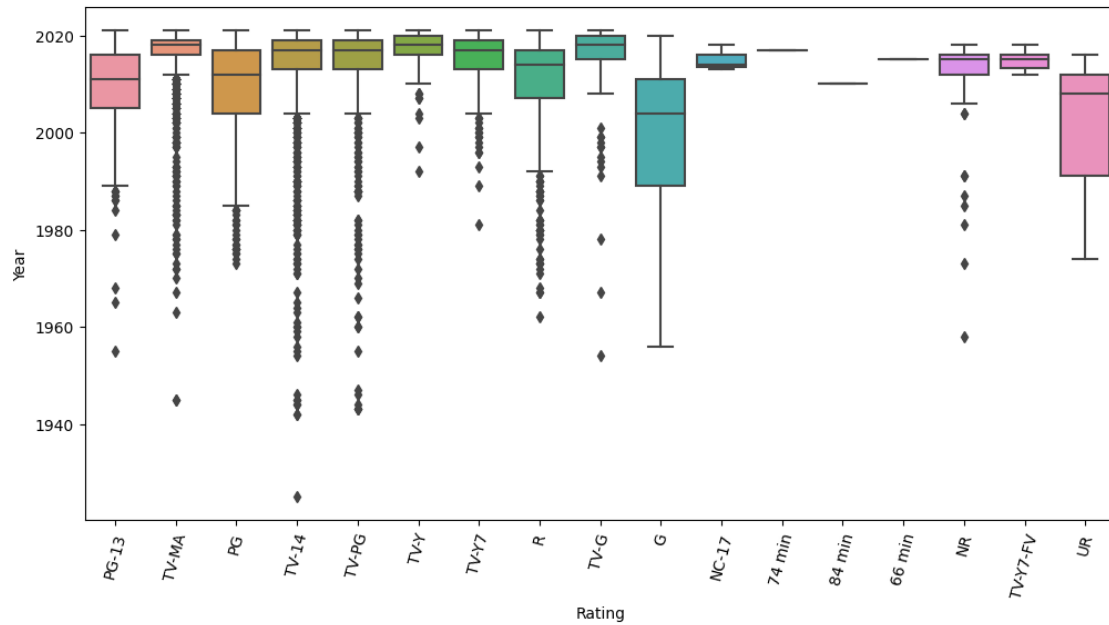
1. basically we observed that movies having highest content count over the last decade.
2. we saw a steady incling in movies until the year 2018 whereas slowly downfall till the year 2021
3. TV shows having very less content count in the starting period if time.

```
[ ]:
```

```
[ ]:
```

```
[56]: plt.figure(figsize = [12,6])
ax = sns.boxplot(x = data['rating'], y = data['release_year'])
plt.xticks(rotation =75)
plt.xlabel("Rating")
plt.ylabel("Year")
```

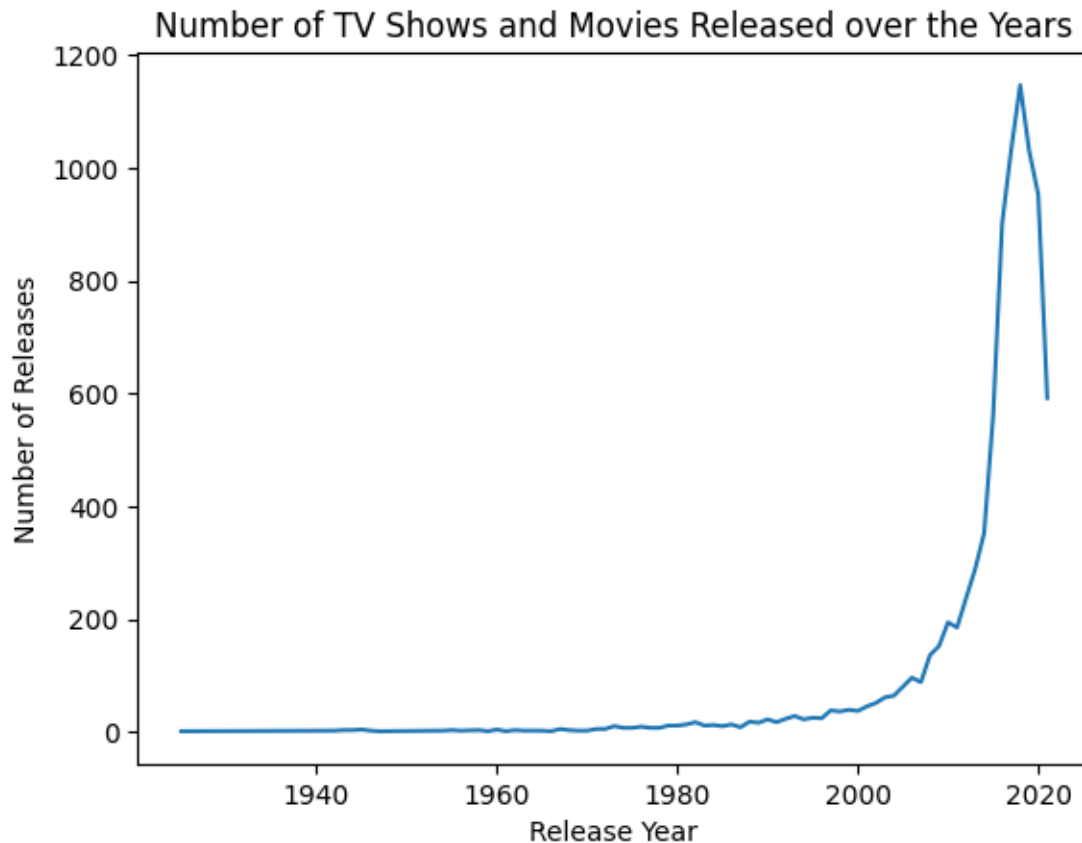
```
[56]: Text(0, 0.5, 'Year')
```



[]:

[]:

```
[54]: release_counts = data['release_year'].value_counts().sort_index()
plt.plot(release_counts.index, release_counts.values)
plt.xlabel('Release Year')
plt.ylabel('Number of Releases')
plt.title('Number of TV Shows and Movies Released over the Years')
plt.show()
```



above are the number of movies and TV shows released over the years

[]:

[]:

[]:

2 BUSINESS INSIGHTS

1. We can conclude that people prefer more regional content. Korean language content is most popular in South Korea, Spanish language content is popular among Mexico and Spain, British English content is more popular in United Kingdom.
2. India is the second highest content produced after USA.
3. Most of the movies directed by Rajiv Chilaka.
4. Anupam Kher is actor in most of the Movies.
5. International Movies have a highest count in India.
6. Ken Burns and Alastair Fothergill were the directed most TV Shows.

3 RECOMMENDATIONS

1. These findings can inform further analyses or decision-making processes.
2. Netflix should provide perks for customers who residing in countries having more number of content count. For e.g. top countries like USA, India, Japan.
3. Watch time in TV Shows is observed more as compared to Movies in recent years, so we should focus more on rating wise content for TV Shows than movies.