Netflix Case Study

Analyse the data and provide insights that will assist Netflix in selecting what sort of shows/movies to make and how to expand the business in different countries.

1. Defining Problem Statement and Analysing basic metrics

In [1]:

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

In [2]:

```
df=pd.read_csv("C:/Users/Seamovation Labs/Downloads/Netflix-business-case.csv")
```

In [3]:

```
df.head()
```

Out[3]:

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

In [4]:

```
df
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021
...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006

	show_id	type	title	director	cast	country	date_added	release_year
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015

8807 rows × 12 columns

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 [5]:

```
df.columns #checking column names
```

Out[5]:

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

In [6]:

```
df.shape #total rows and column
```

Out[6]:

```
(8807, 12)
```

Data type of all attribute

In [7]:

```
df.dtypes #cheking the datatypes
```

Out[7]:

```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

In [8]:

```
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 [9]:

```
df.describe(include = "object").T
```

Out[9]:

	count	unique	top	freq
show_id	8807	8807	s1	1
type	8807	2	Movie	6131
title	8807	8807	Dick Johnson Is Dead	1
director	6173	4528	Rajiv Chilaka	19
cast	7982	7692	David Attenborough	19
country	7976	748	United States	2818
date_added	8797	1767	January 1, 2020	109
rating	8803	17	TV-MA	3207
duration	8804	220	1 Season	1793
listed_in	8807	514	Dramas, International Movies	362
description	8807	8775	Paranormal activity at a lush, abandoned prope...	4

Missing Value Dectection

In [10]:

```
print('\nColumns with missing value:')
print(df.isnull().any())
```

```

Columns with missing value:
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 [11]:

```
df.T.apply(lambda x: x.isnull().sum(), axis = 1) #checking null value counts
```

Out[11]:

```
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 [12]:

```
df.isnull().sum().sum()
```

Out[12]:

```
4307
```

As a primarily observation the dataset contains

There are total dataset are 8807 out of 4307 are missing and here are list below:-

- Director = 2634
- cast = 825

- country = 831
- date added = 10
- rating = 4
- duration = 3

In [13]:

```
df.isnull().sum()/len(df)*100 #null value with percentage
```

Out[13]:

```
show_id          0.000000
type             0.000000
title            0.000000
director        29.908028
cast             9.367549
country          9.435676
date_added       0.113546
release_year     0.000000
rating           0.045418
duration         0.034064
listed_in        0.000000
description      0.000000
dtype: float64
```

Highest amount of missing data is for director(30%), cast(9%) and country(9%). Filling missing values for each...

Filling missing value for rating

In [14]:

```
df['rating'].isna().sum()
```

Out[14]:

```
4
```

In [15]:

```
df[df['rating'].isna()]
```

Out[15]:

	show_id	type	title	director	cast	country	date_added	release_yea
5989	s5990	Movie	13TH: A Conversation with Oprah Winfrey & Ava ...	NaN	Oprah Winfrey, Ava DuVernay	NaN	January 26, 2017	201
6827	s6828	TV Show	Gargantia on the Verdurous Planet	NaN	Kaito Ishikawa, Hisako Kanemoto, Ai Kayano, Ka...	Japan	December 1, 2016	201
7312	s7313	TV Show	Little Lunch	NaN	Flynn Curry, Olivia Deeble, Madison Lu, Oisín ...	Australia	February 1, 2018	201
7537	s7538	Movie	My Honor Was Loyalty	Alessandro Pepe	Leone Frisa, Paolo Vaccarino, Francesco Miglio...	Italy	March 1, 2017	201

In [16]:

```
df['rating'].unique()
```

Out[16]:

```
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)
```

In [17]:

```
df['rating'].fillna("NR", inplace = True) #Replace raing Nan to NR
```

In [18]:

```
df['rating'].unique()
```

Out[18]:

```
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',
       'TV-Y7-FV', 'UR'], dtype=object)
```

The status of missing ratings data has been changed to 'NR' (Not Rated). Incorrect data will be replaced in the next step.

Filing missing value for duration

In [19]:

```
df['rating'].isna().sum()
```

Out[19]:

```
0
```

In [20]:

```
df[df['duration'].isna()]
```

Out[20]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	d
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017	74 min	
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	84 min	
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015	66 min	

The missing duration is available in rating, so need to place it from rating to duration

In [21]:

```
#copying the duration from rating column to duration column
df.loc[5541,'duration'] = df.loc[5541,'rating']
df.loc[5794,'duration'] = df.loc[5794,'rating']
df.loc[5813,'duration'] = df.loc[5813,'rating']
```

In [22]:

```
df['duration'].isna().sum()
```

Out[22]:

```
0
```

In [23]:

```
#replacing these values to "NR" in the rating column
df['rating'].replace('74 min','NR', inplace = True)
df['rating'].replace('84 min','NR', inplace = True)
df['rating'].replace('66 min','NR', inplace = True)
```

In [24]:

```
df['rating'].unique()
```

Out[24]:

```
array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
       'TV-G', 'G', 'NC-17', 'NR', 'TV-Y7-FV', 'UR'], dtype=object)
```

Filling missing values for country

In [26]:

```
df['country'].isna().sum()
```

Out[26]:

```
831
```

In [27]:

```
#filling missing country values with most frequent country
df['country'] = df['country'].fillna(df['country'].mode()[0])
```

In [28]:

```
df['country'].isna().sum()
```

Out[28]:

```
0
```

Dropping missing values for date_added

In [30]:

```
df['date_added'].isna().sum()
```

Out[30]:

```
10
```

In [31]:

```
df.dropna(subset = ['date_added'], inplace = True)
```

In [32]:

```
df['date_added'].isna().sum()
```

Out[32]:

```
0
```

Spliting the date to day, month and year

In [33]:

```
df['date_added'] = pd.to_datetime(df['date_added'])
```

In [36]:

```
#Separating the date,month and year in new column in the dataframe
df['day'] = df['date_added'].dt.day.astype(int)
df['month'] = df['date_added'].dt.month.astype(int)
df['year'] = df['date_added'].dt.year.astype(int)
```

In [37]:

```
df.head()
```

Out[37]:

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

In [38]:

```
df.drop('added_day',1,inplace = True)
df.drop('added_month',1,inplace = True)
df.drop('added_year',1,inplace = True)
```

```
C:\Users\Seamovation Labs\AppData\Local\Temp\ipykernel_11448\3505506712.p
y:1: FutureWarning: In a future version of pandas all arguments of DataFra
me.drop except for the argument 'labels' will be keyword-only.
  df.drop('added_day',1,inplace = True)
C:\Users\Seamovation Labs\AppData\Local\Temp\ipykernel_11448\3505506712.p
y:2: FutureWarning: In a future version of pandas all arguments of DataFra
me.drop except for the argument 'labels' will be keyword-only.
  df.drop('added_month',1,inplace = True)
C:\Users\Seamovation Labs\AppData\Local\Temp\ipykernel_11448\3505506712.p
y:3: FutureWarning: In a future version of pandas all arguments of DataFra
me.drop except for the argument 'labels' will be keyword-only.
  df.drop('added_year',1,inplace = True)
```

In [39]:

```
df.head()
```

Out[39]:

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

Creating a new DF to store title and each cast

In [40]:

```
constraint = df['cast'].apply(lambda x: str(x).split(',')).tolist()
df_new = pd.DataFrame(constraint, index=df['title'])
df_new = df_new.stack()
df_new = pd.DataFrame(df_new)
```

In [41]:

```
df_new.head()
```

Out[41]:

		0
title		
Dick Johnson Is Dead	**0**	nan
Blood & Water	**0**	Ama Qamata
	1	Khosi Ngema
	2	Gail Mabalane
	3	Thabang Molaba

In [42]:

```
df_new.T
```

Out[42]:

title	**Dick Johnson Is Dead**								**Blood**
	0	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	nan	Ama Qamata	Khosi Ngema	Gail Mabalane	Thabang Molaba	Dillon Windvogel	Natasha Thahane	Arno Greeff	Xolile Tshabalala

1 rows × 64882 columns

Visual / Data Analysis

Univariate - A single variable was used in the analysis. We won't get into the arithmetic underlying these concepts right now; instead, let's look at them in graph form.

In [43]:

```
df.describe()
```

Out[43]:

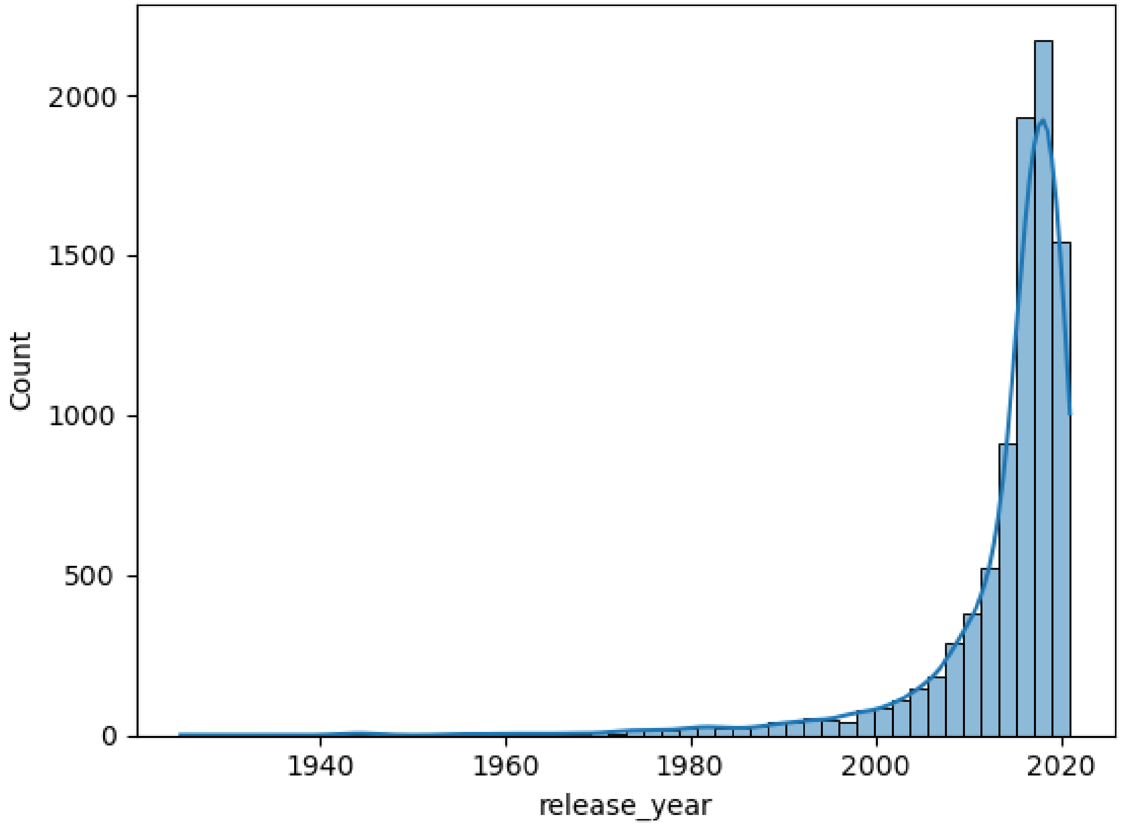
	release_year	day	month	year
count	8797.000000	8797.000000	8797.000000	8797.000000
mean	2014.183472	12.497329	6.654996	2018.871888
std	8.822191	9.887551	3.436554	1.574243
min	1925.000000	1.000000	1.000000	2008.000000
25%	2013.000000	1.000000	4.000000	2018.000000
50%	2017.000000	13.000000	7.000000	2019.000000
75%	2019.000000	20.000000	10.000000	2020.000000
max	2021.000000	31.000000	12.000000	2021.000000

In [44]:

```
#Boxplot
sns.histplot(df['release_year'], bins = 50, kde = True)
```

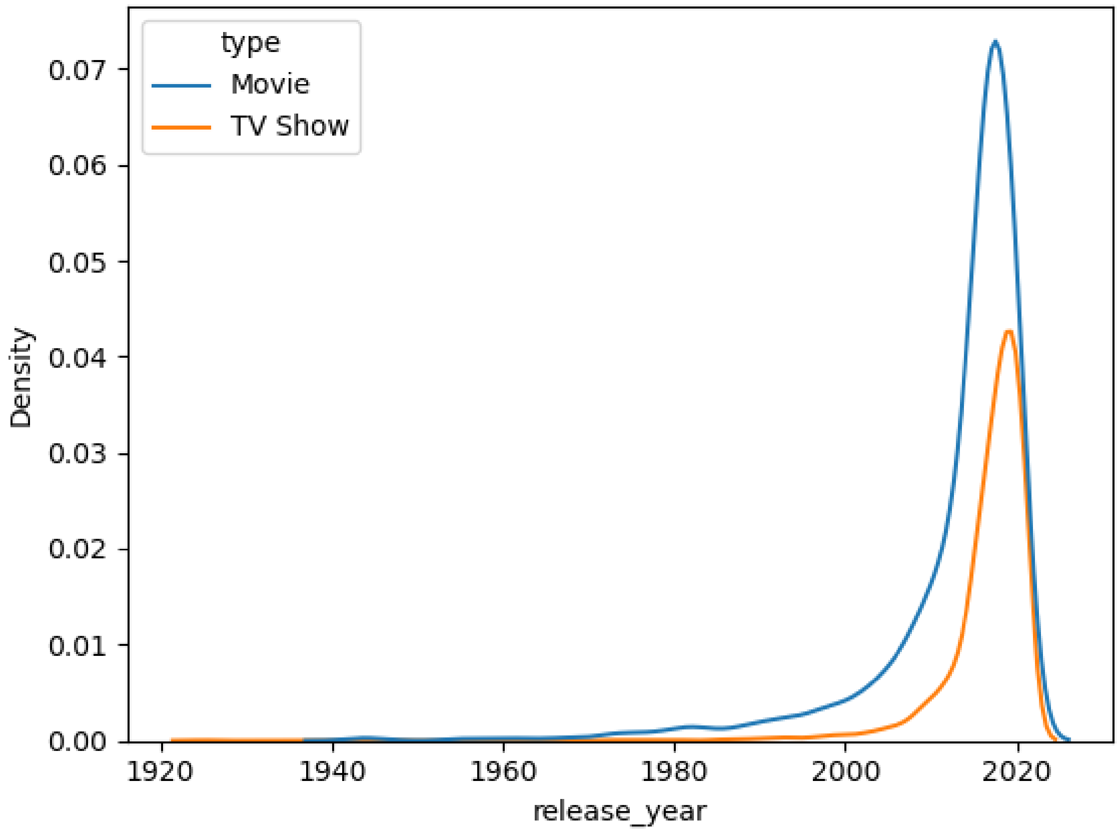
Out[44]:

```
<Axes: xlabel='release_year', ylabel='Count'>
```



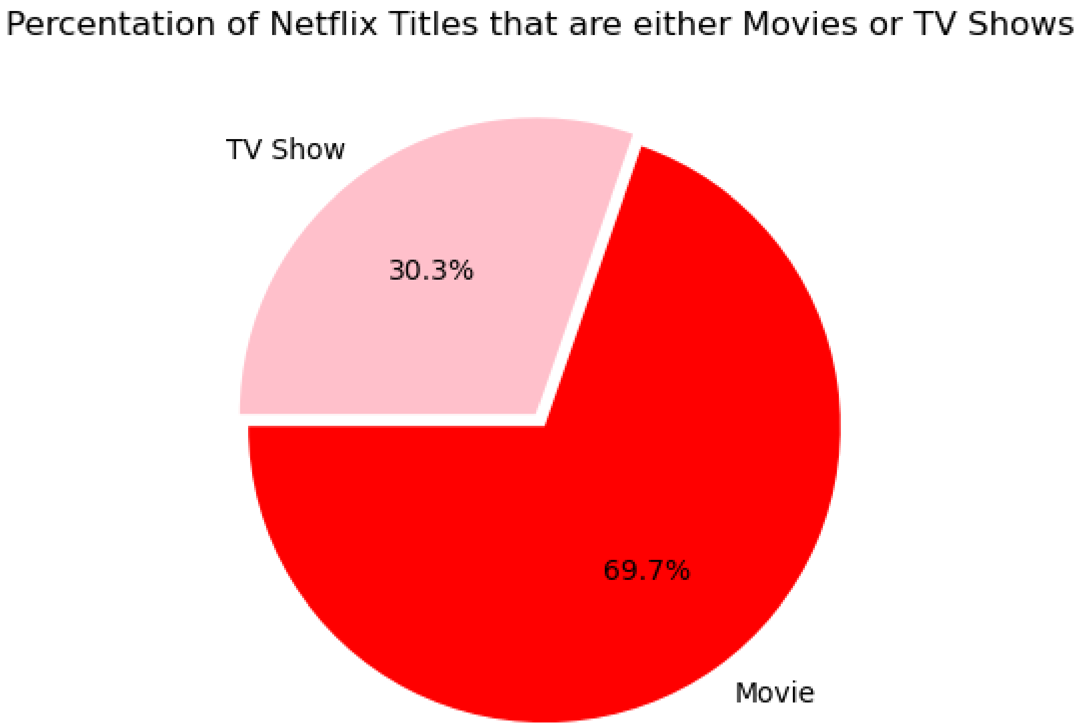
In [45]:

```
sns.kdeplot(data = df, x = 'release_year', hue = 'type')
plt.show()
```



In [46]:

```
#Distplot
plt.title("Percentation of Netflix Titles that are either Movies or TV Shows")
g=plt.pie(df.type.value_counts(),explode=(0.025,0.025),
labels=df.type.value_counts().index, colors=['red','pink'],autopct='%1.1f%%',
startangle=180)
```



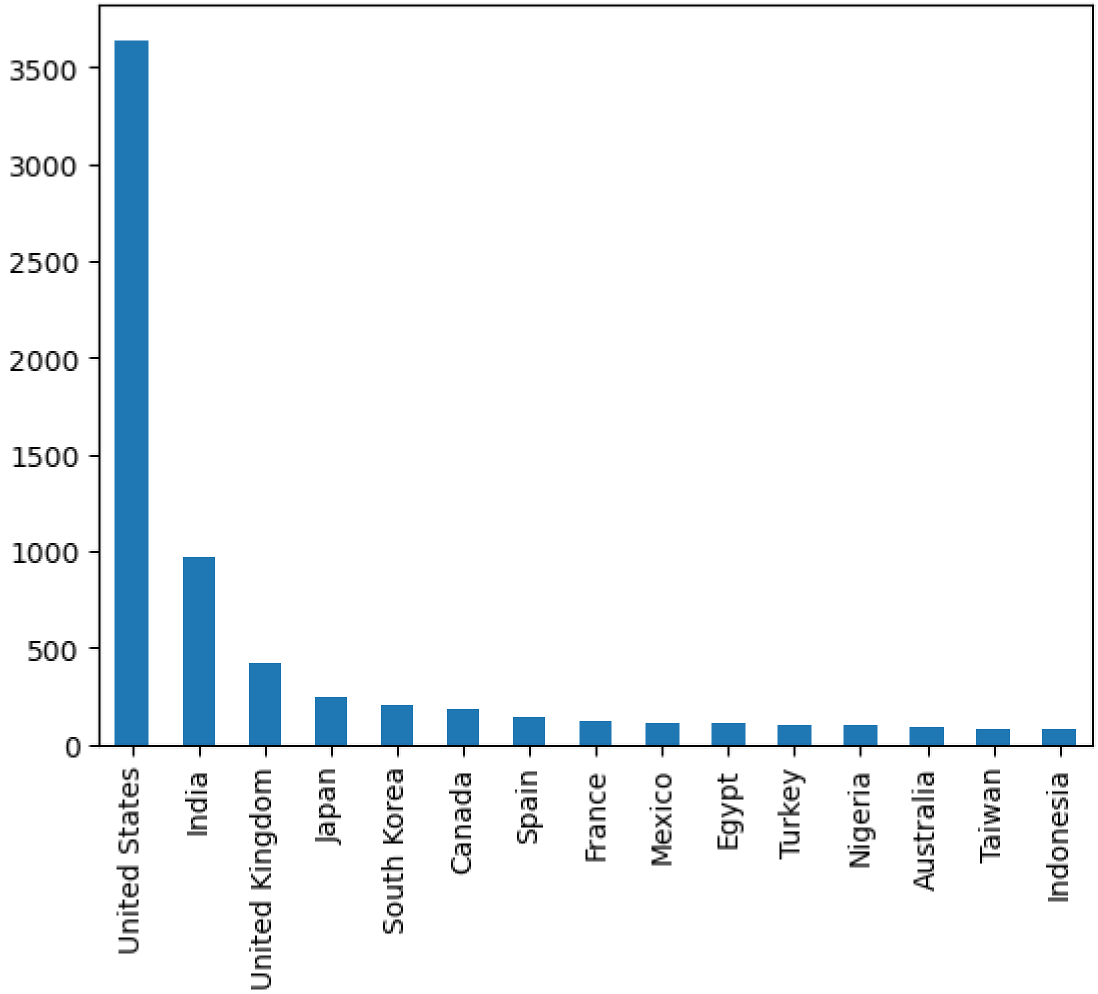
There are 69.6% for movie and 30.4% for TV Shows

In [48]:

```
#Shows which country content the releases the most
df['country'].value_counts().head(15).plot(kind = 'bar')
```

Out[48]:

```
<Axes: >
```



United States has the highest number of releases, India follows as the second

In [63]:

```
df['director'].value_counts().head(15)
```

Out[63]:

```
Rajiv Chilaka             19
Raúl Campos, Jan Suter    18
Marcus Raboy              16
Suhas Kadav               16
Jay Karas                 14
Cathy Garcia-Molina       13
Martin Scorsese           12
Youssef Chahine           12
Jay Chapman               12
Steven Spielberg          11
Don Michael Paul          10
David Dhawan               9
Yılmaz Erdoğan             8
Lance Bangs                8
Kunle Afolayan             8
Name: director, dtype: int64
```

The top contributor is an Indian Director Rajiv Chilaka

In [64]:

```
# Getting more infor on Rajiv Chilaka
df.loc[df['director']=='Rajiv Chilaka'].groupby('listed_in').count()
```

Out[64]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dura
listed_in										
Children & Family Movies	18	18	18	18	16	18	18	18	18	
Children & Family Movies, Sports Movies	1	1	1	1	1	1	1	1	1	

Second top contributor is Raúl Campos and Jan Suter

In [65]:

```
# Getting more infor on Raúl Campos, Jan Suter
df.loc[df['director']=='Raúl Campos, Jan Suter'].groupby('listed_in').count()
```

Out[65]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dura
listed_in										
Stand-Up Comedy	18	18	18	18	18	18	18	18	18	

Third top contributor is Marcus Raboy

In [67]:

```
#Getting more info on Marcus Raboy(3rd top contributor)
df.loc[df['director']=='Marcus Raboy'].groupby('listed_in').count()
```

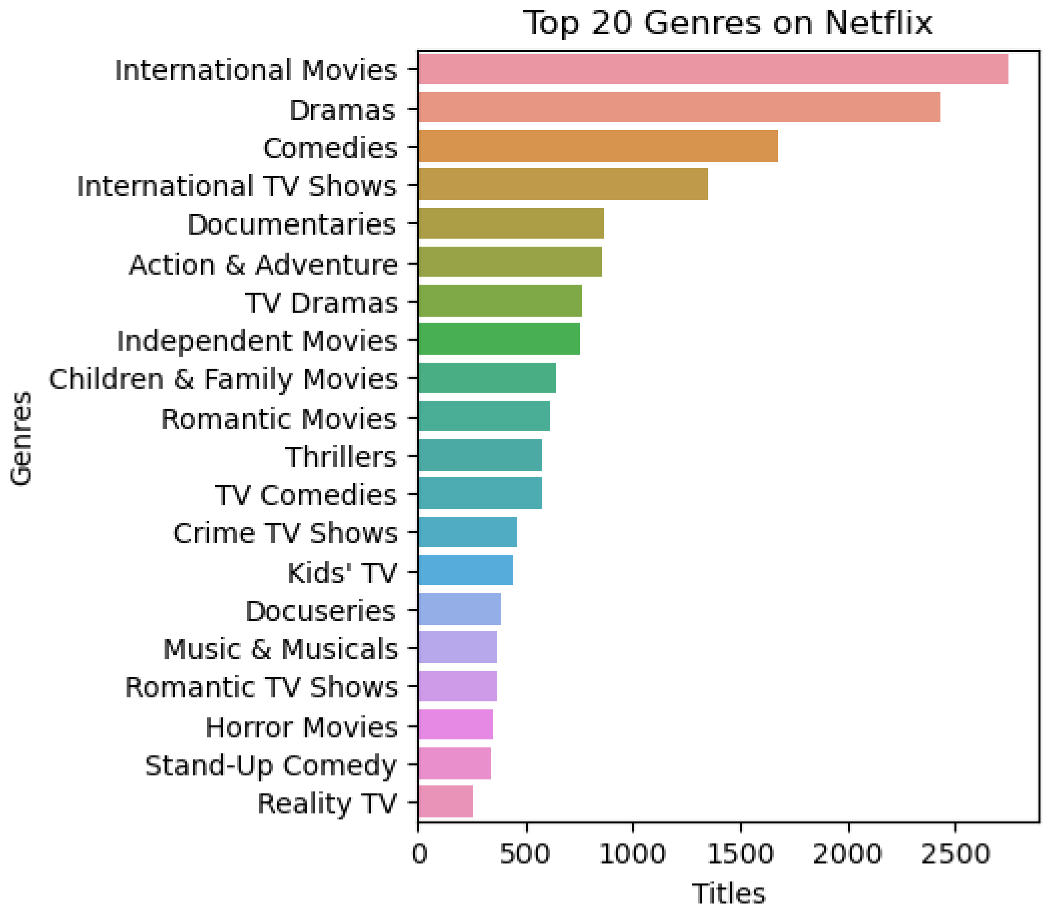
Out[67]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	du
listed_in										
Stand-Up Comedy	15	15	15	15	15	15	15	15	15	
Stand-Up Comedy & Talk Shows, TV Comedies	1	1	1	1	1	1	1	1	1	

Clearly, Kids Entertainment and Comedy Programs seems to be very popular

In [75]:

```
filtered_genres = df.set_index('title').listed_in.str.split(', ',
expand=True).stack().reset_index(level=1, drop=True);
plt.figure(figsize=(4,5))
g = sns.countplot(y = filtered_genres,
order=filtered_genres.value_counts().index[:20])
plt.title('Top 20 Genres on Netflix')
plt.xlabel('Titles')
plt.ylabel('Genres')
plt.show()
```



However, the highest number of content are international movies and dramas, even though comedy

FINAL RECOMMENDATIONS

- Netflix has to focus on TV Shows also because there are people who will like to see tv shows rather than movies
- By approaching the top director we can plan some more movies/tv shows in order to increase the popularity
- Not only reaching top director we can also see the director with less no of movies and having high rating as there may be some financial issues or anything so inorder to get good content netflix can reach to them and netflix can produce the movie and give the director a chance.
- We have seen most no of international movies genre so need to give priority to other geners like hooro,comedy..etc
- Over 69% of the netflix catalog are movies - movies seem to be trending

- Data shows that over 2000 new content is uploaded on the 1st of every month, and over 600 during mid month. Hence the recommended day to upload new content is the first of every month
- Some movies can be released directly into ott which has some positive talk which may help in improving subscriptions
- Advertisement in the country which has very less movies released should be increased and attract people of that country by making their native TV Shows

In [ ]: