Netflix Case Study

Netflix is an US based online streaming plaform which streams both movies as well as web series

Objectives

- Increase revenue by attracting more subscribers
- Acquire pre created popular content or produce relevant ones through distinguised creators
- Recommend relevant content to subscribers based their preference for better user experience

Risk

- In case Netflix is unable to lure in content creators or acquire popular existing contents, it might end up having an outdated and uninteresting library
- Lack of intersting content could lead to loss of both existing and potential new subscribers
- Could result in loss of revenues and become unsustainable in the long run

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import chi2_contingency
```

Loading Dataset

In [33]:

```
master_dataset = pd.read_csv('C:/Users/Arijit/Scaler/Projects/Netflix/netflix.csv')
```

Verifying the start and the end of the dataset to identify the validity of the data

```
In [3]: master_dataset.head() # Looks clean
```

Out[3]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

In [4]:
```
master_dataset.tail() # Looks clean
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A political cartoonist, a crime reporter and a...
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	2 Seasons	Kids' TV, Korean TV Shows, TV Comedies	While living alone in a spooky town, a young g...
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 min	Comedies, Horror Movies	Looking to survive in a world taken over by zo...
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 min	Children & Family Movies, Comedies	Dragged from civilian life, a former superhero...
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

Inference :

- The data looks clean based on the above findings

Data Summary

In [ ]:

```
master_dataset.info() # 8807 records, 1 int column rest object
```

Observation :

- The dataset has 12 columns and 8807 records
- Release year is numerical column. The rest of the columns are categorical in nature

In [5]:

```
master_dataset.describe(include='all')
```

Out[5]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
count	8807	8807	8807	6173	7982	7976	8797	8807.000000	8803	8804	8807	8807
unique	8807	2	8807	4528	7692	748	1767	NaN	17	220	514	8775
top	s143	Movie	Fuga	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	NaN	TV-MA	1 Season	Dramas, International Movies	Paranormal activity at a lush, abandoned prope...
freq	1	6131	1	19	19	2818	109	NaN	3207	1793	362	4
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2014.180198	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	8.819312	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1925.000000	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2013.000000	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2017.000000	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2019.000000	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2021.000000	NaN	NaN	NaN	NaN

Observation :

- show_id is the unique identifier per record
- type has 2 unique values i.e. Movie and TV Show
- title represents content name and has unique values for all records

- content from 4528 different directors are available
- data for around 7692 unique combinations of star cast is present
- content is currenly streamed in several countries
- content available from year 1925 till recently suggesting tremendous coverage
- data also has the runtime per content, censorship ratings and a summary of the content

Quality check :

In [6]:

```
print((master_dataset.isna().sum() / len(master_dataset)) * 100) # almost 30% NaN values in director column
```

```
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
```

In [7]:

```
print(len(master_dataset.loc[master_dataset.isnull().values.any(axis=1),:])) # 3475 records with aleast 1 NaN v
print((len(master_dataset.loc[master_dataset.isnull().values.any(axis=1),:]) / len(master_dataset)) * 100) # al
```

```
3475
39.45724991484047
```

Observation :

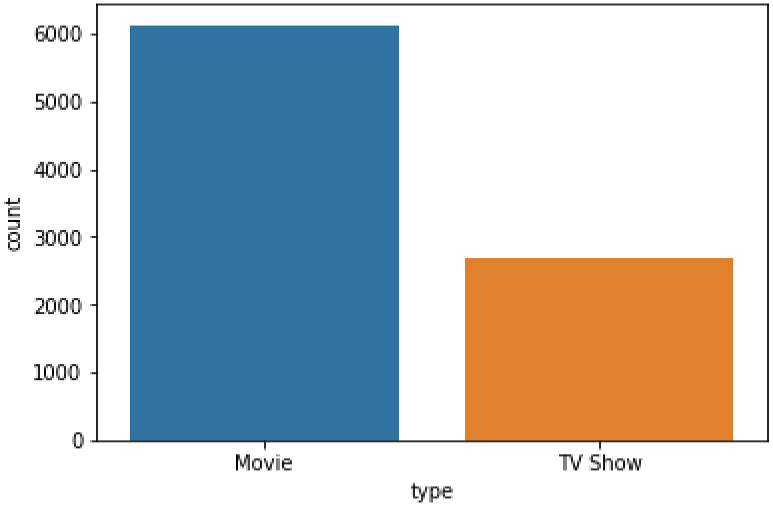
- Around 30% of director values are missing
- More than 9% missing values for both cast and country
- Less than 1% data missing for date_added, rating and duration
- More than 39% records have at least one or more missing details

Univariate Analysis

In [8]:

```
print(master_dataset['type'].value_counts())
sns.countplot(data=master_dataset['type'], x = master_dataset['type'].index)
plt.show()
```

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



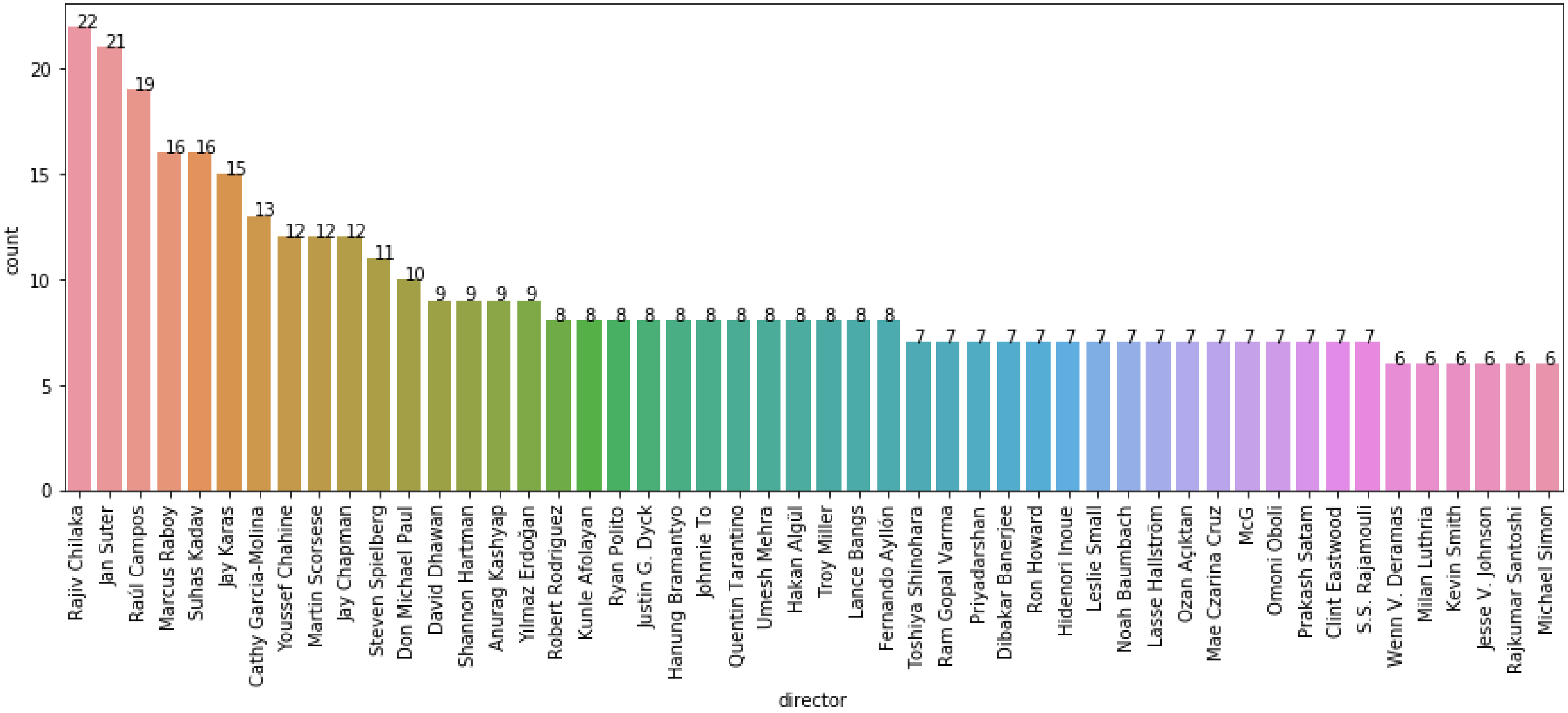
Observation :

- 2 types of content available
- Around 6131 movies and 2676 TV Shows

In [9]:

```
master_dataset_dir_exp = master_dataset.copy()
master_dataset_dir_exp['director'].fillna('NA', inplace=True)
master_dataset_dir_exp['director'] = [list(map(lambda x : x.strip(), val))  for val in master_dataset_dir_exp['
master_dataset_dir_exp = master_dataset_dir_exp.explode('director')
print(master_dataset_dir_exp['director'].value_counts())

# Directors with 3 or lesser content
count_val = master_dataset_dir_exp['director'].value_counts()
print(count_val[count_val.values <= 3].index.to_series().count())

dir = master_dataset_dir_exp['director'].value_counts().drop(index='NA').head(50).index
data = master_dataset_dir_exp.loc[master_dataset_dir_exp['director'].isin(dir)]
plt.figure(figsize=(15,5))
plot = sns.countplot(data=data, x = 'director', order = dir)
plot.set_xticklabels(labels=dir, rotation=90)
for p in plot.patches:
   plot.annotate('{:}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
plt.show()
```

```
NA                          2634
Rajiv Chilaka                 22
Jan Suter                     21
Raúl Campos                   19
Marcus Raboy                  16
                            ...
Li Pei-Chuan                   1
Will McCormack                 1
Jorge Hernandez Aldana         1
Nathaniel Warsh                1
Nottapon Boonprakob            1
Name: director, Length: 4994, dtype: int64
4798
```



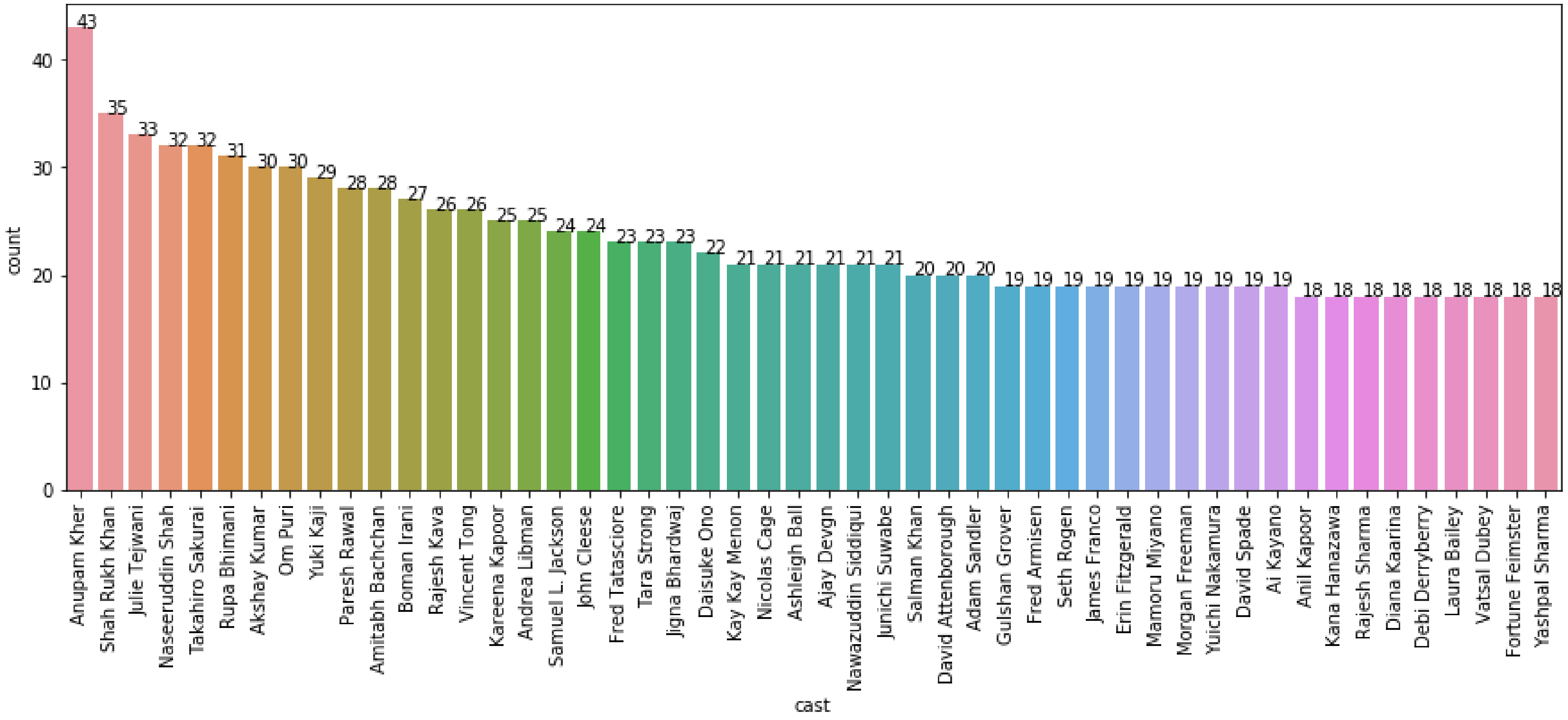
Observation :

- Contents of 4994 directors are available
- Rajiv Chilaka has the maximum number of contents available i.e. 22 closely followed by Jan Suter with 21 and Raul Campos with 19
- Famous Hollywood directors like Martin Scorsese and Steven Spielberg have 12 and 11 contents available respectively
- Famous Indian directors like Anurag Kashyap and SS Rajamouli have 9 and 7 contents available respectively
- Around 4798 directors have 3 or lesser content available which is more than 96% percent. This looks like an area of concern

In [10]:

```
master_dataset_cast_exp = master_dataset.copy()
master_dataset_cast_exp['cast'].fillna('NA', inplace=True)
master_dataset_cast_exp['cast'] = [list(map(lambda x : x.strip(), val))  for val in master_dataset_cast_exp['cas
master_dataset_cast_exp = master_dataset_cast_exp.explode('cast')
print(master_dataset_cast_exp['cast'].value_counts())

# Actors with 3 or lesser content
count_val = master_dataset_cast_exp['cast'].value_counts()
print(count_val[count_val.values <= 3].index.to_series().count())

cst = master_dataset_cast_exp['cast'].value_counts().drop('NA').head(50).index
plt.figure(figsize=(15,5))
plot = sns.countplot(data=master_dataset_cast_exp.loc[master_dataset_cast_exp['cast'].isin(cst)], x = 'cast', or
plot.set_xticklabels(labels=cst, rotation=90)
for p in plot.patches:
   plot.annotate('{:}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
plt.show()
```

```
NA                        825
Anupam Kher                43
Shah Rukh Khan             35
Julie Tejwani              33
Naseeruddin Shah           32
                         ...
Brad Kalilimoku             1
J. Don Ferguson             1
Skylan Brooks               1
Ana Martín                  1
Richard Collins-Moore       1
Name: cast, Length: 36440, dtype: int64
33193
```



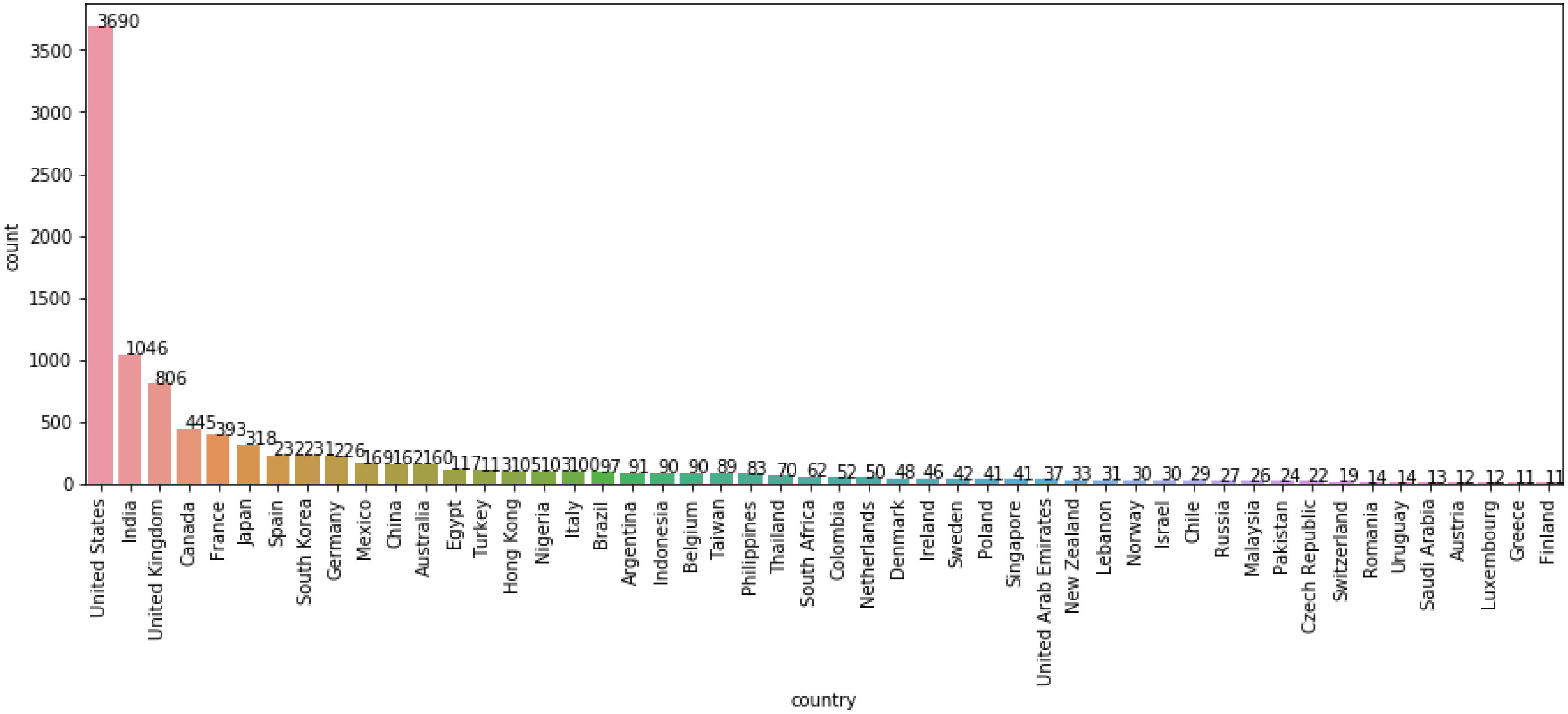
Observation :

- Contents of 36440 actors are available
- Anupam Kher has the maximum number of contents available i.e. 43 closely followed by famous Indian actor Shah Rukh Khan with 35 and Julie Tejwani with 33
- Famous Hollywood actors like Samuel Jacson and Nicolas Cage have 24 and 21 contents available respectively
- Famous Indian actors like Amitabh Bachchan and Akshay Kumar have 28 and 30 contents available respectively
- Around 33193 directors have 3 or lesser content available which is more than 91% percent. This again looks like an area of concern

In [11]:

```
master_dataset_cnt_exp = master_dataset.copy()
master_dataset_cnt_exp['country'].fillna('NA', inplace=True)
master_dataset_cnt_exp['country'] = [list(map(lambda x : x.strip(), val))  for val in master_dataset_cnt_exp['co
master_dataset_cnt_exp = master_dataset_cnt_exp.explode('country')
print(master_dataset_cnt_exp['country'].value_counts())


# Countries with 3 or lesser content
count_val = master_dataset_cnt_exp['country'].value_counts()
print(count_val[count_val.values <= 3].index.to_series().count())

cnt = master_dataset_cnt_exp['country'].value_counts().drop('NA').head(50).index
plt.figure(figsize=(15,5))
plot = sns.countplot(data=master_dataset_cnt_exp.loc[master_dataset_cnt_exp['country'].isin(cnt)], x = 'country'
plot.set_xticklabels(labels=cnt, rotation=90)

for p in plot.patches:
   plot.annotate('{:}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
plt.show()
```

```
United States     3690
India             1046
NA                 831
United Kingdom     806
Canada             445
                  ... 
Azerbaijan           1
Mongolia             1
Jamaica              1
Latvia               1
Angola               1
Name: country, Length: 124, dtype: int64
53
```



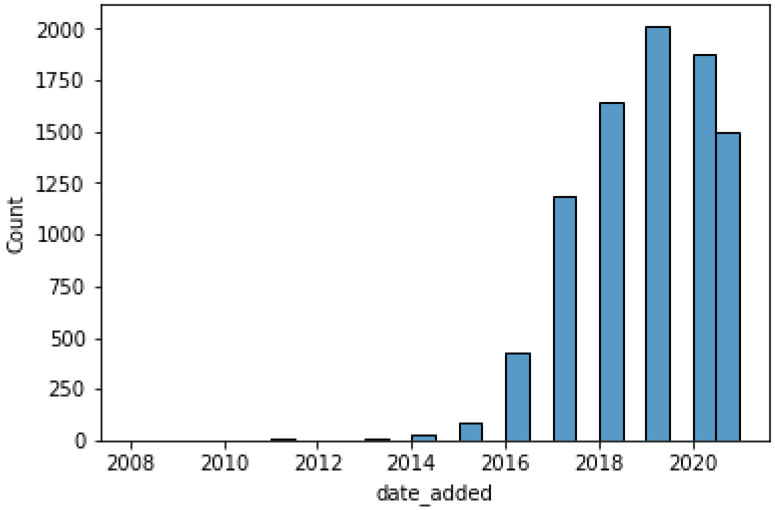
Observation :

- Contents are available in 124 countries
- US has maximum amount of contents available at 3690
- India has the second most amount of content i.e. 1046 which is almost 28% lesser than US. Since India has a large population, increasing relevant content can attract high volumes of subscribers
- Around 53 countries have 3 or lesser content available which is more than 42% percent. This can definitely be improved

In [12]:

```
added_year = pd.to_datetime(master_dataset['date_added'])
ys = pd.DatetimeIndex(added_year).year
ys = ys.dropna()
print(ys.value_counts().sort_index(ascending=False))

plot = sns.histplot(data = ys, binwidth=.5, bins=ys.value_counts().count())

plt.show()
```

```
2021.0    1498
2020.0    1879
2019.0    2016
2018.0    1649
2017.0    1188
2016.0     429
2015.0      82
2014.0      24
2013.0      11
2012.0       3
2011.0      13
2010.0       1
2009.0       2
2008.0       2
Name: date_added, dtype: int64
```



Observation :

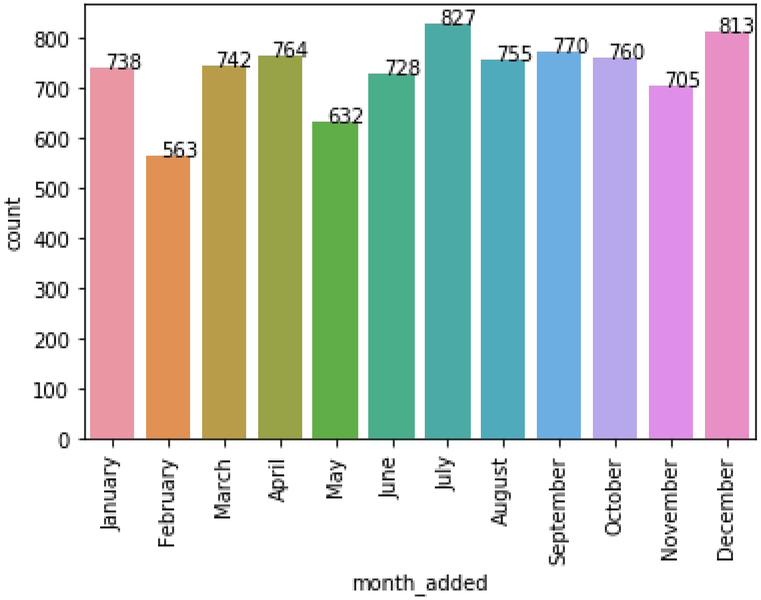
- Till date the maximum content was made available in the year 2019 i.e. 2016
- Despite the years 2020 and 2021 being affected due o pandemic, considerable amount of content have bee added i.e. 1879 and 1498 respectively
- There has been a whopping 750% rise in amount of content being made available since 2008

In [13]:
```
from functools import cmp_to_key
month_list=['January','February','March','April','May','June','July','August','September','October','November','
def comparator(a, b):
  a_index = month_list.index(a)
  b_index = month_list.index(b)
  if a_index>b_index:
      return 1
  elif a_index<b_index:
      return -1
  return 0
```

In [14]:

```
ms = pd.Series(pd.DatetimeIndex(added_year).month_name().to_series().values)
columns = master_dataset.columns.to_list()
columns.append('month_added')
master_dataset = pd.concat([master_dataset, ms], ignore_index=True, axis=1)
master_dataset.columns = columns
print(master_dataset['month_added'].value_counts())


ms_ord = sorted(master_dataset['month_added'].value_counts().index, key=cmp_to_key(comparator))
plot = sns.countplot(data=master_dataset, x='month_added', order=ms_ord)
plot.set_xticklabels(labels=ms_ord, rotation=90)

for p in plot.patches:
   plot.annotate('{:}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
plt.show()
```

```
July         827
December     813
September    770
April        764
October      760
August       755
March        742
January      738
June         728
November     705
May          632
February     563
Name: month_added, dtype: int64
```



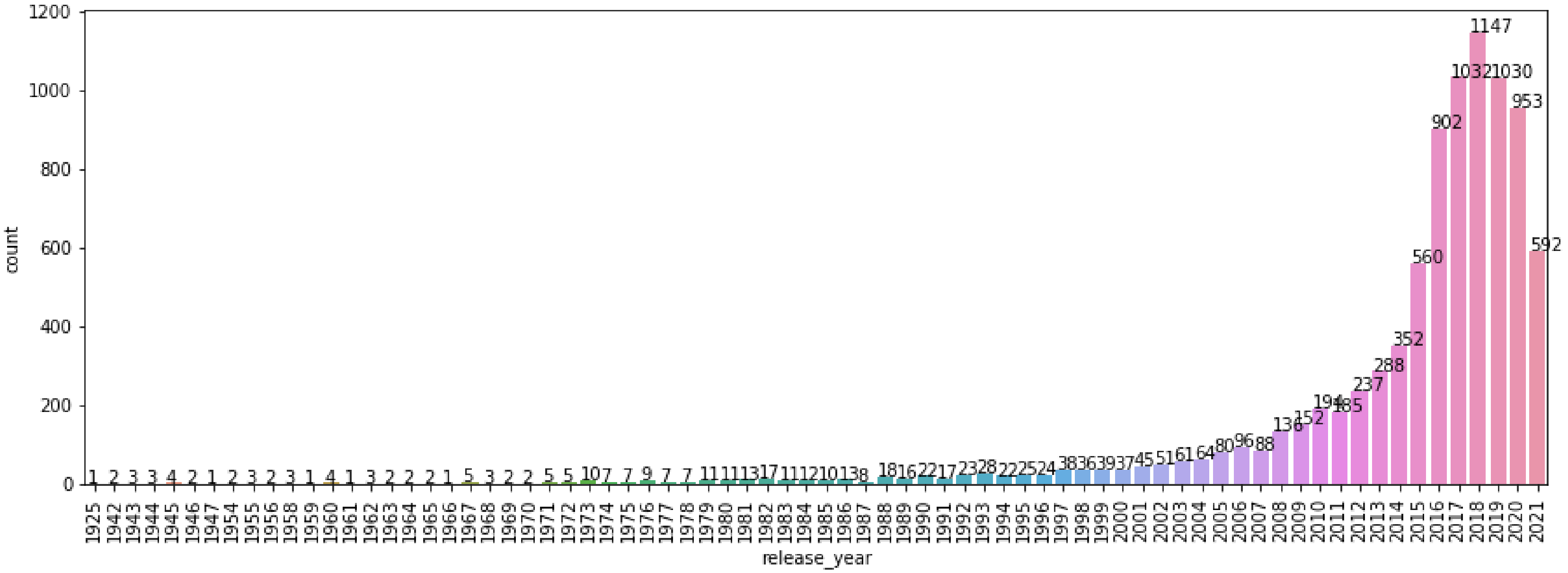
Observation :

- The maximum amount of contents have been added in the month of July i.e. 827
- December being a festive month, also has seen an addition of 813 contents
- Barring February and May, the distribution of contents across different months have been decent

In [15]:

```python
print(master_dataset['release_year'].value_counts())

ry_ord = master_dataset['release_year'].value_counts().sort_index(ascending=True).index
plt.figure(figsize=(15,5))
plot = sns.countplot(data = master_dataset, x='release_year', order=ry_ord)
plot.set_xticklabels(labels=ry_ord, rotation=90)

for p in plot.patches:
   plot.annotate('{:}'.format(p.get_height()), (p.get_x(), p.get_height()+0.01))
plt.show()
```

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



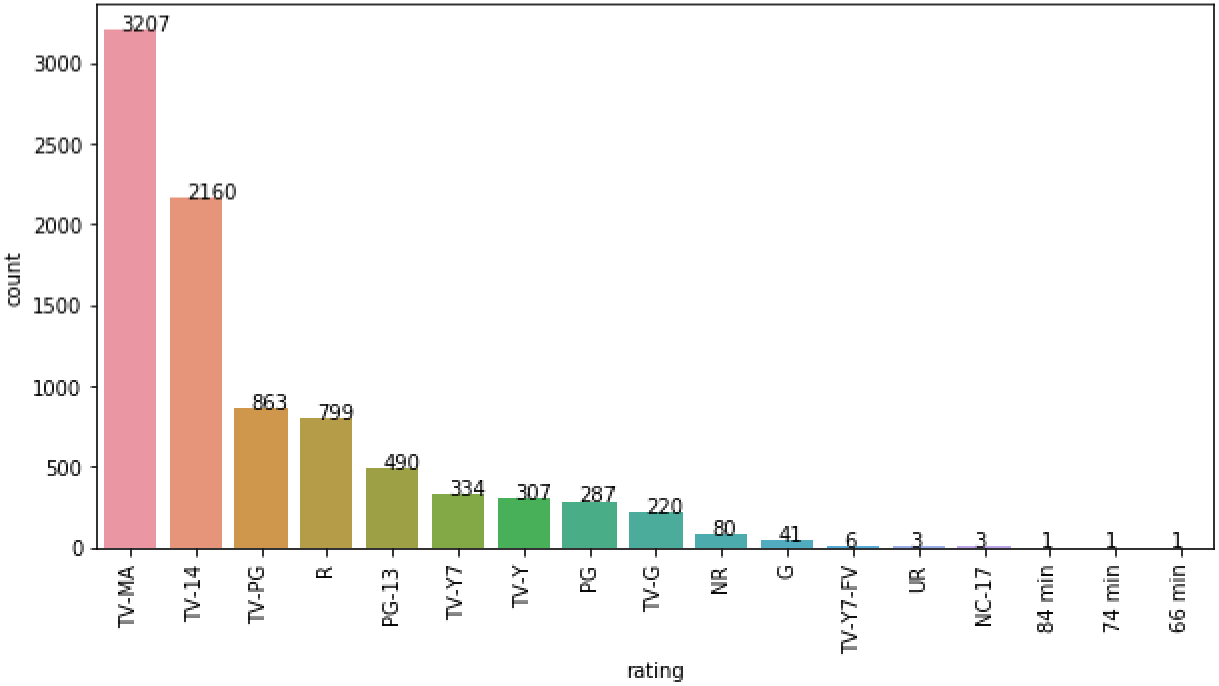
Observation :

- Year 2018 had the maximum amount of releases i.e. 1147
- A sharp decline is observed in 2021 which could also be attributed to pandemic

In [16]:

```
rts = master_dataset['rating']
print(rts.value_counts())

ra_ord = master_dataset['rating'].value_counts().index

plt.figure(figsize=(10,5))
plot = sns.countplot(data = master_dataset, x = 'rating', order=ra_ord)
plot.set_xticklabels(labels=ra_ord, rotation=90)

for p in plot.patches:
   plot.annotate('{:}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
plt.show()
```

```
TV-MA       3207
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
UR             3
NC-17          3
84 min         1
74 min         1
66 min         1
Name: rating, dtype: int64
```



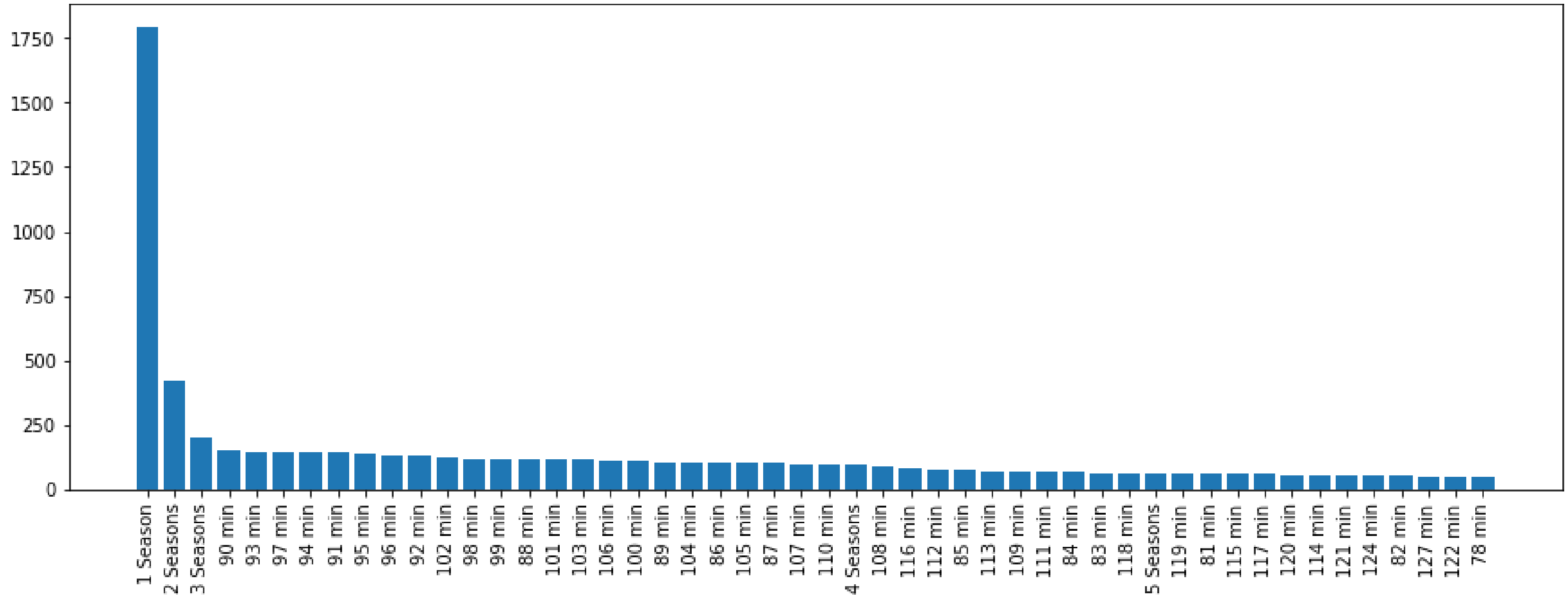
Observation :

- The maximum amount of content, 3207, is present in TV-MA rating category i.e. for mature audiences
- Majority of the content is available either for mature audiences or with parental guidance. More kid friendly content can be made available

In [17]:

```
ds = master_dataset['duration']
ds = ds.fillna('NA')
print(ds.value_counts())
#sns.histplot(data = ds, bins=10)


plt.figure(figsize=(15,5))
plt.bar(ds.value_counts().index.to_series().head(50), ds.value_counts().head(50))
plt.xticks(rotation=90)
plt.show()
```

```
1 Season      1793
2 Seasons      425
3 Seasons      199
90 min         152
93 min         146
              ...
194 min          1
43 min           1
17 Seasons       1
212 min          1
230 min          1
Name: duration, Length: 221, dtype: int64
```



Observation :

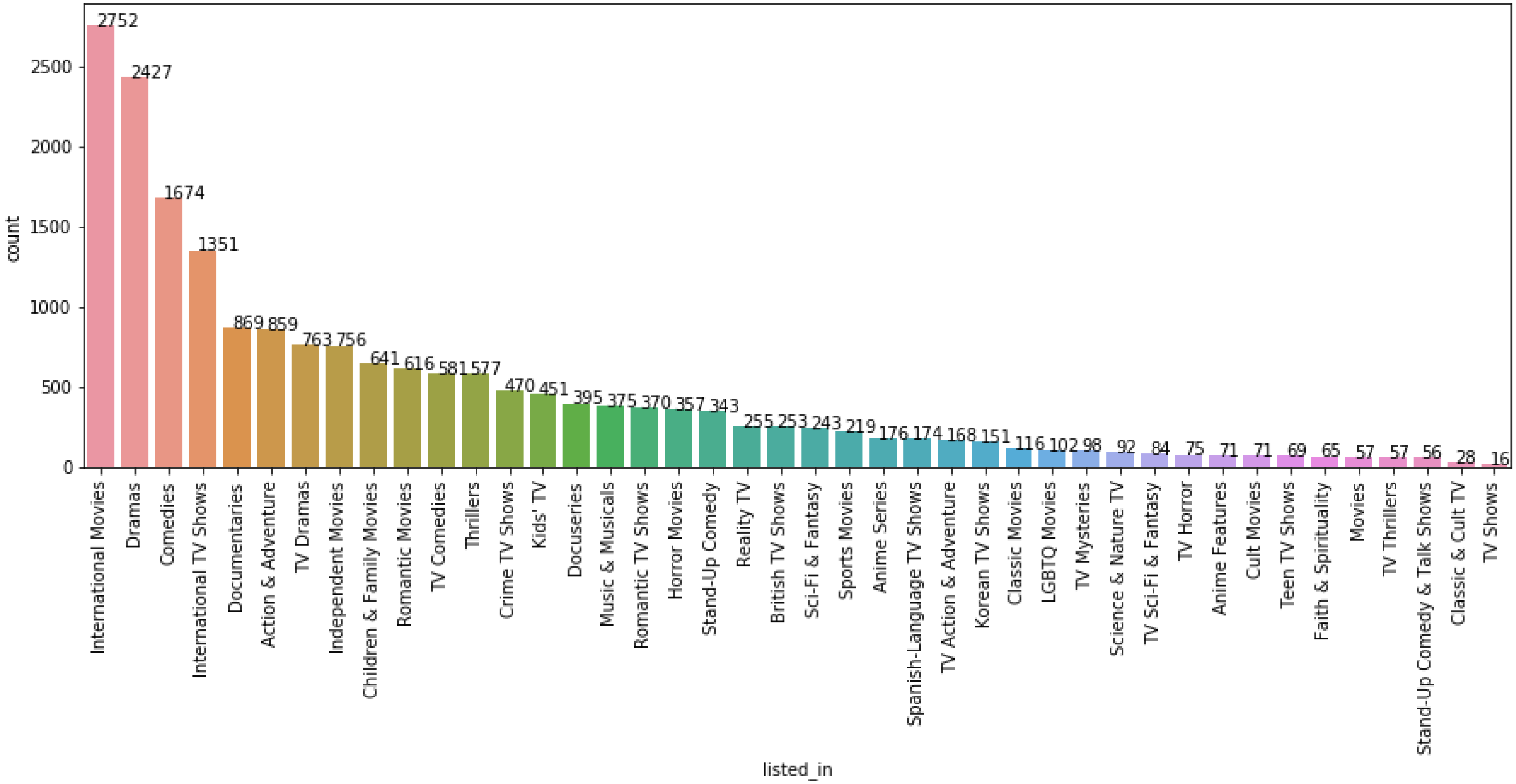
- Around 1793 TV Shows have only Season 1 available.
- Based on the popularity of the content, further seasons can be planned to attract subscribers

In [18]:

```
master_dataset_gnr_exp = master_dataset.copy()
master_dataset_gnr_exp['listed_in'].fillna('NA', inplace=True)
master_dataset_gnr_exp['listed_in'] = [list(map(lambda x : x.strip(), val))  for val in master_dataset_gnr_exp['
master_dataset_gnr_exp = master_dataset_gnr_exp.explode('listed_in')
print(master_dataset_gnr_exp['listed_in'].value_counts())


li_ord = master_dataset_gnr_exp['listed_in'].value_counts().index

plt.figure(figsize=(15,5))
plot = sns.countplot(data = master_dataset_gnr_exp, x = 'listed_in', order=li_ord)
plot.set_xticklabels(labels=li_ord, rotation=90)

for p in plot.patches:
   plot.annotate('{:}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
plt.show()
```

```
International Movies            2752
Dramas                          2427
Comedies                        1674
International TV Shows          1351
Documentaries                    869
Action & Adventure               859
TV Dramas                        763
Independent Movies               756
Children & Family Movies         641
Romantic Movies                  616
TV Comedies                      581
Thrillers                        577
Crime TV Shows                   470
Kids' TV                         451
Docuseries                       395
Music & Musicals                 375
Romantic TV Shows                370
Horror Movies                    357
Stand-Up Comedy                  343
Reality TV                       255
```

```
British TV Shows                253
Sci-Fi & Fantasy                243
Sports Movies                   219
Anime Series                    176
Spanish-Language TV Shows       174
TV Action & Adventure           168
Korean TV Shows                 151
Classic Movies                  116
LGBTQ Movies                    102
TV Mysteries                     98
Science & Nature TV              92
TV Sci-Fi & Fantasy              84
TV Horror                        75
Anime Features                   71
Cult Movies                      71
Teen TV Shows                    69
Faith & Spirituality             65
Movies                           57
TV Thrillers                     57
Stand-Up Comedy & Talk Shows     56
Classic & Cult TV                28
TV Shows                         16
Name: listed_in, dtype: int64
```



Observation :

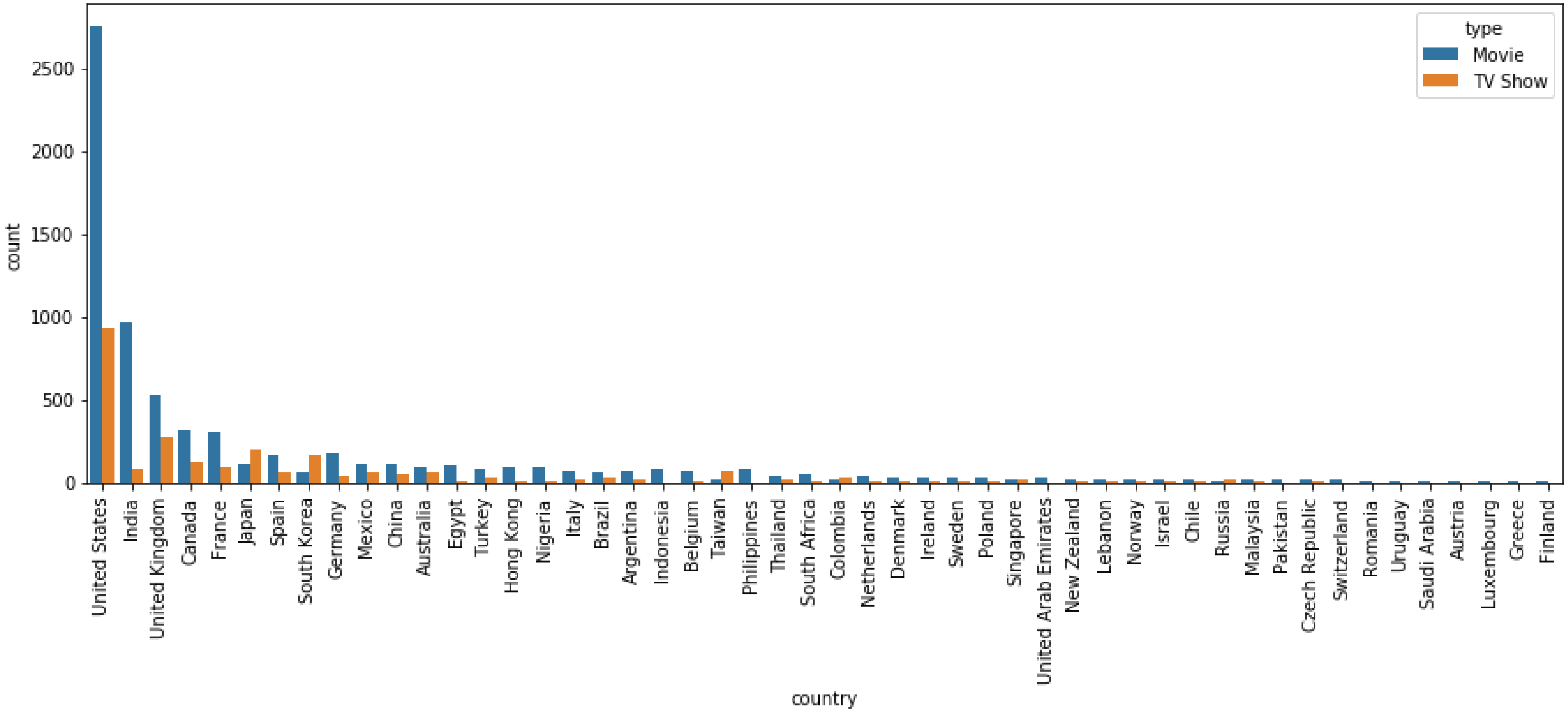
- Around 2752 International movies are available closely followed by 2427 dramas
- Animation and Horror genres can be eplored more as they have only 176 and 75 such contents respectively

Bivariate Analysis

In [19]:

```
print(pd.crosstab(master_dataset['country'], master_dataset['type'], margins=True, margins_name='Total').sort_va


cnt = master_dataset_cnt_exp['country'].value_counts().drop('NA').head(50).index
plt.figure(figsize=(15,5))
plot = sns.countplot(data=master_dataset_cnt_exp.loc[master_dataset_cnt_exp['country'].isin(cnt)], x = 'country'
plot.set_xticklabels(labels=cnt, rotation=90)
plt.show()
```

```
type                            Movie  TV Show  Total
country
Total                            5691     2285   7976
United States                    2058      760   2818
India                             893       79    972
United Kingdom                    206      213    419
Japan                              76      169    245
South Korea                        41      158    199
Canada                            122       59    181
Spain                              97       48    145
France                             75       49    124
Mexico                             70       40    110
Egypt                              92       14    106
Turkey                             76       29    105
Nigeria                            86        9     95
Australia                          39       48     87
Taiwan                             13       68     81
Indonesia                          77        2     79
Brazil                             50       27     77
United Kingdom, United States      63       12     75
Philippines                        73        2     75
United States, Canada              51       22     73
```



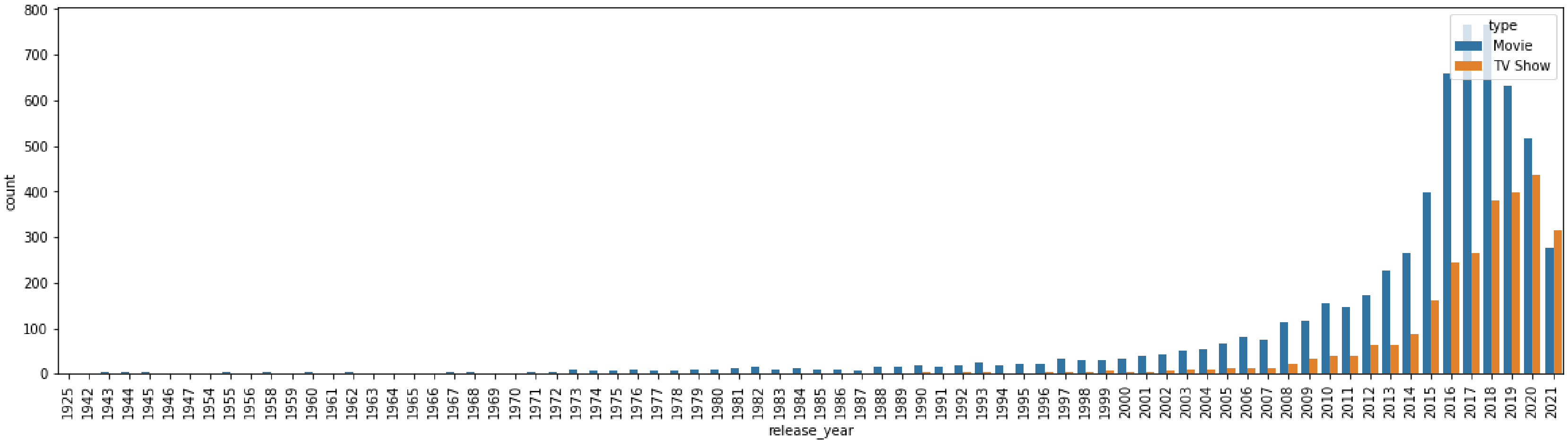
Observation :

- US leads the way with 2058 movies and 760 TV Shows
- India has 893 movies available but has only 79 TV Shows. This certainly needs improvement as the number of footfalls in theatres have decreased considerably since the pandemic outbreak.
- Uk have more than 200 contents for both movies and TV Shows
- Isreal being one of the top quality content creators does not find place in the top 30 countries with most content. Again another area that can be looked at.

In [20]:

```
print(pd.crosstab(master_dataset['release_year'], master_dataset['type'], margins=True, margins_name='Total').sc

ry_ord = master_dataset['release_year'].value_counts().sort_index(ascending=True).index
plt.figure(figsize=(20,5))
plot = sns.countplot(data = master_dataset, x='release_year', order=ry_ord, hue='type')
plot.set_xticklabels(labels=ry_ord, rotation=90)
plt.show()
```

```
type          Movie  TV Show  Total
release_year
Total          6131     2676   8807
2018            767      380   1147
2017            767      265   1032
2019            633      397   1030
2020            517      436    953
2016            658      244    902
2021            277      315    592
2015            398      162    560
2014            264       88    352
2013            225       63    288
2012            173       64    237
2010            154       40    194
2011            145       40    185
2009            118       34    152
2008            113       23    136
2006             82       14     96
2007             74       14     88
2005             67       13     80
2004             55        9     64
2003             51       10     61
```

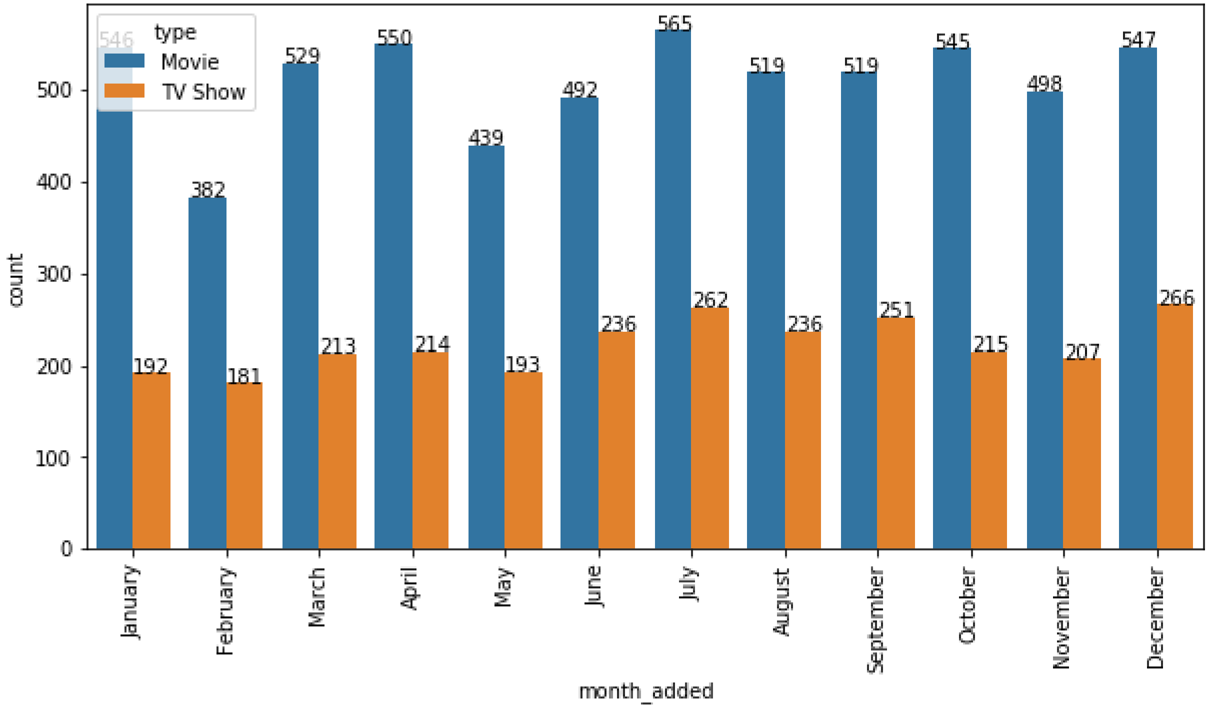


Observation :

- 2018 saw the maximum content being produced with 767 Movies and 380 TV Shows
- 2021 saw a sharp decline most likely due to the pandemic outbreak
- The entertainment industry has come a long way from releasing only 1 film in 1925 to nearly 1000 contents per year in recent times

In [21]:

```
ms_ord = sorted(master_dataset['month_added'].value_counts().index, key=cmp_to_key(comparator))
plt.figure(figsize=(10,5))
plot = sns.countplot(data=master_dataset, x='month_added', order=ms_ord, hue='type')
plot.set_xticklabels(labels=ms_ord, rotation=90)
for p in plot.patches:
   plot.annotate('{:}'.format(p.get_height()), (p.get_x(), p.get_height()+0.01))
plt.show()
```

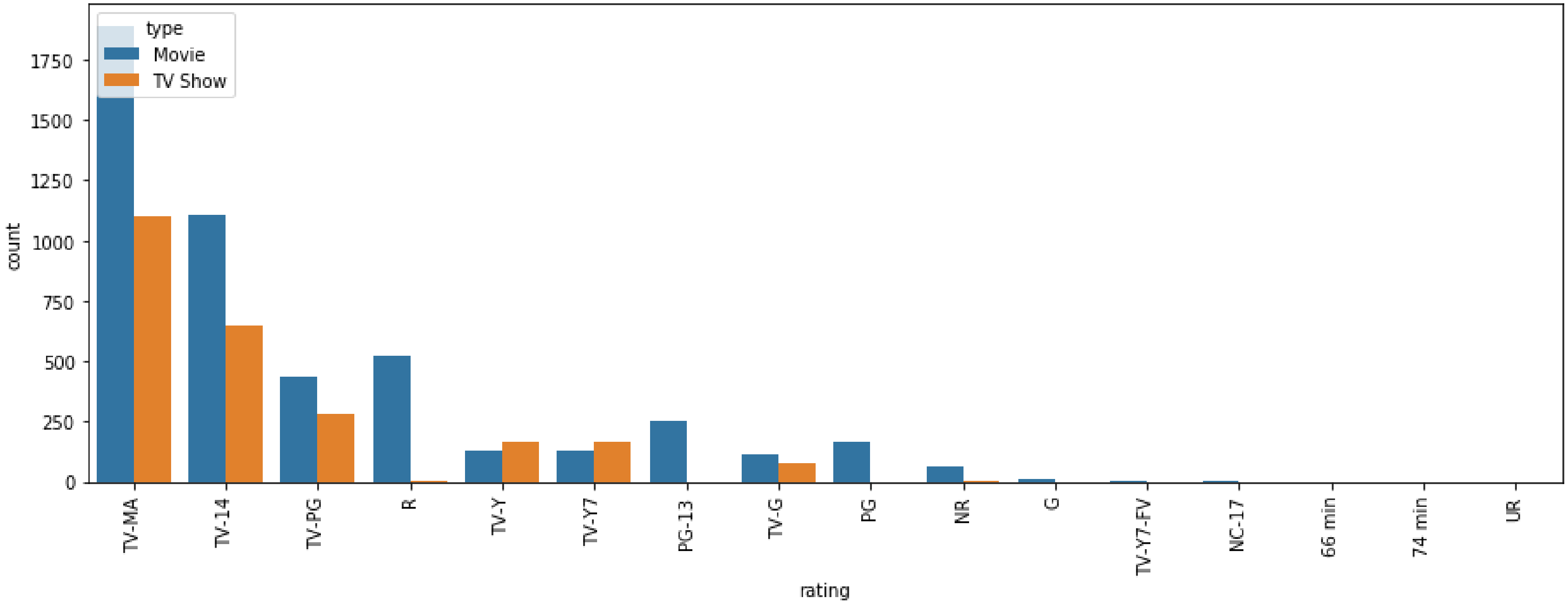


Observation :

- July and December months get majority of the release
- February and May are the least producers

In [22]:

```
# Ratings added in last 10 years
last_10_year_ratings = master_dataset.loc[master_dataset['release_year'] > 2010]

ra_ord = last_10_year_ratings['rating'].value_counts().index

plt.figure(figsize=(15,5))
plot = sns.countplot(data = last_10_year_ratings, x = 'rating', order=ra_ord, hue='type')
plot.set_xticklabels(labels=ra_ord, rotation=90)

plt.show()
```

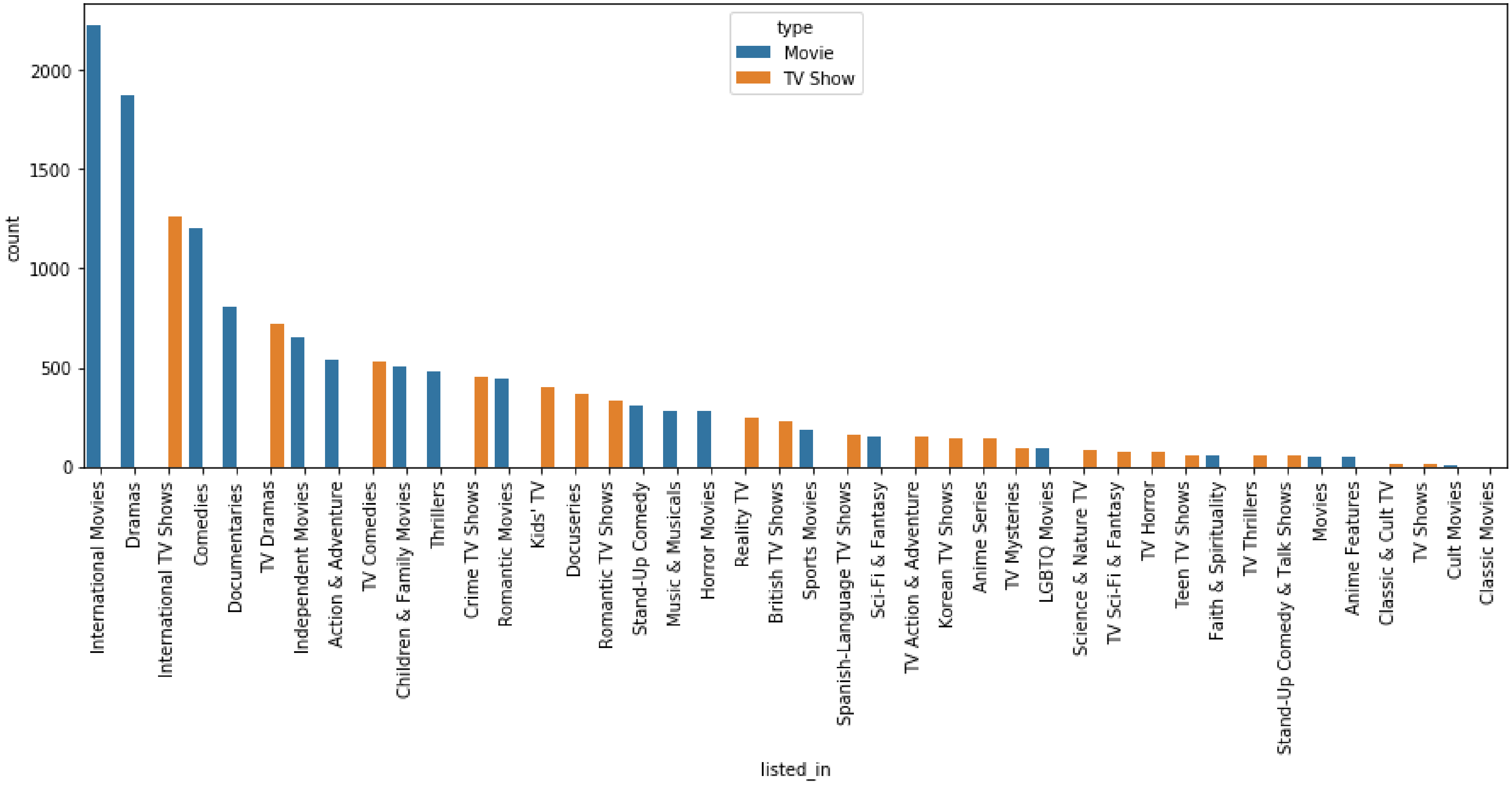


Observation :

- Last 10 years saw an incredibly high amount of content being generated with TV-MA rating
- The amount of movies for kids have decreased considerably and can be an area of focus

In [23]:

```
# Genre added in last 10 years
last_10_year_genre = master_dataset_gnr_exp.loc[master_dataset_gnr_exp['release_year'] > 2010]

li_ord = last_10_year_genre['listed_in'].value_counts().index

plt.figure(figsize=(15,5))
plot = sns.countplot(data = last_10_year_genre, x = 'listed_in', order=li_ord, hue='type')
plot.set_xticklabels(labels=li_ord, rotation=90)
plt.show()
```



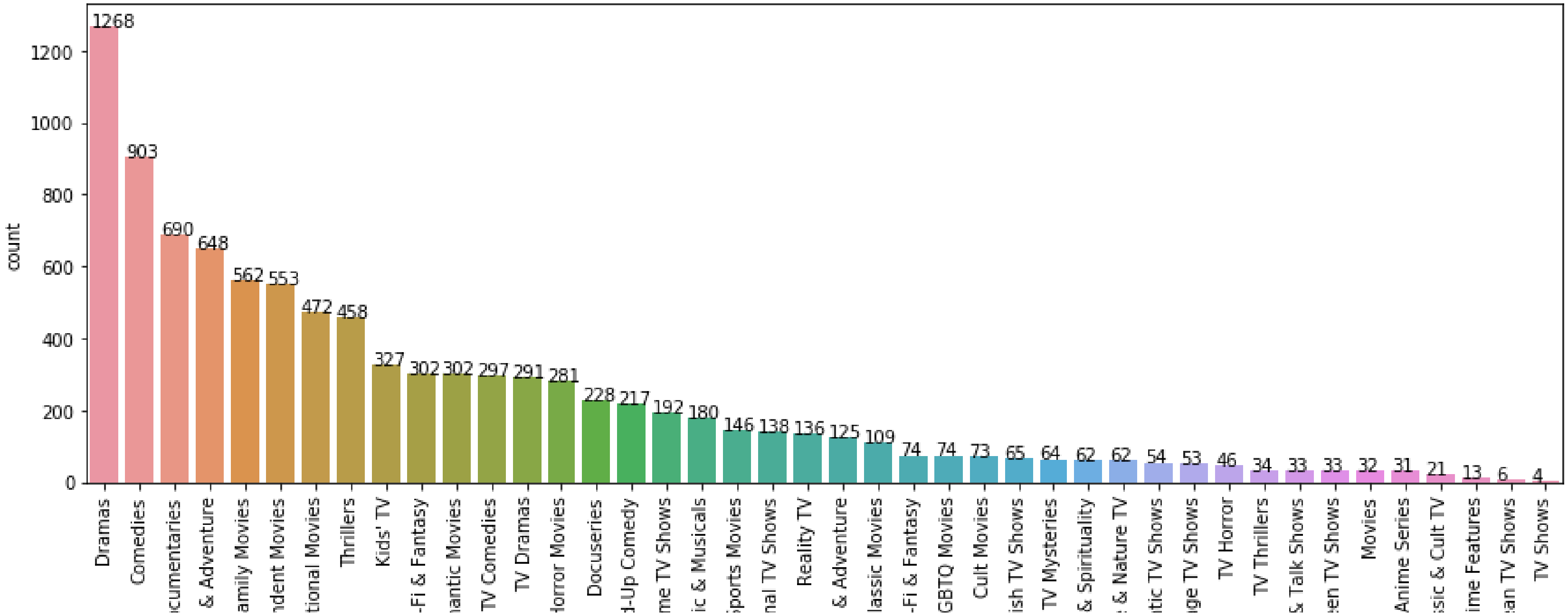
Observation :

- International Movies leads the way in terms of the number of content in the past 10 years
- Animation and Horror genres have been less explored
- The number of sports related content are also considerably lesser in number

In [24]:

```
cnt_listed_exp = master_dataset_cnt_exp.copy()
cnt_listed_exp['listed_in'] = [list(map(lambda x : x.strip(), val))  for val in cnt_listed_exp['listed_in'].str
cnt_listed_exp = cnt_listed_exp.explode('listed_in')
```

In [25]:
```
cnt_list = master_dataset_cnt_exp['country'].value_counts().drop('NA').head(4).index

for cnt_val in cnt_list:
  temp = cnt_listed_exp.loc[cnt_listed_exp['country'] == cnt_val]
  cnt_listed_in = temp['listed_in'].value_counts().index
  plt.figure(figsize=(15,5))
  plot = sns.countplot(data=temp.loc[temp['listed_in'].isin(cnt_listed_in)], x = 'listed_in', order=cnt_listed_i
  plot.set_xticklabels(labels=cnt_listed_in, rotation=90)
  plt.xlabel(f'listed_in for {cnt_val}')
  for p in plot.patches:
    plot.annotate('{:}'.format(p.get_height()), (p.get_x(), p.get_height()+0.05))
  plt.show()
```

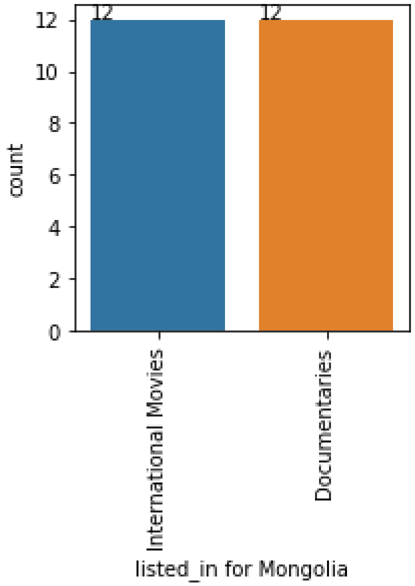


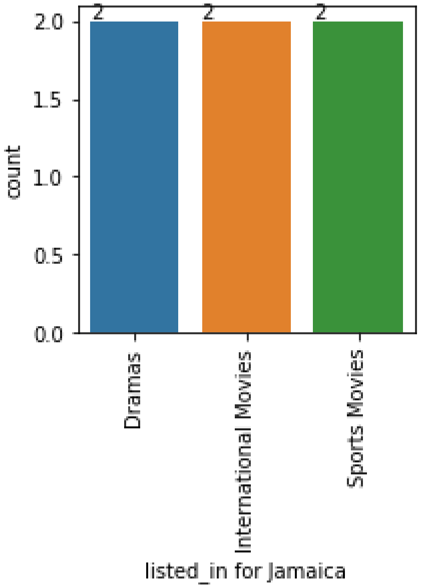
Observation :

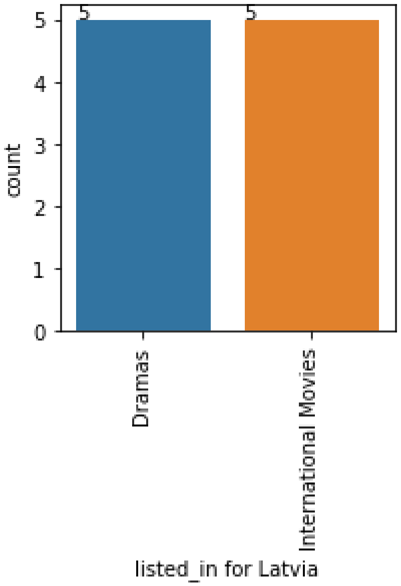
- All the top 4 countries have a different mix of content available
- Drama, Comedies and Documentaries are the top 3 genres prevalent in US
- International movies, Drama and Comedies are the top 3 genres in India
- British TV shows, Drama and International movies are the top 3 genres in UK
- Comedies, Drama and Children & Family contents are the top 3 genres in Canada

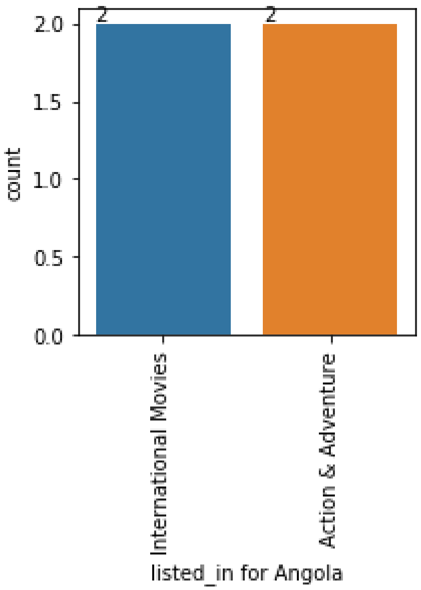
- The amount of Animation content is extremely low in both US and Canada
- Animation content is completely unavailable in India and UK

In [26]:
```
master_dataset_cnt_exp['country'].value_counts().drop('NA')

cnt_list = master_dataset_cnt_exp['country'].value_counts().drop('NA').tail(4).index

for cnt_val in cnt_list:
  temp = cnt_listed_exp.loc[cnt_listed_exp['country'] == cnt_val]
  cnt_listed_in = temp['listed_in'].value_counts().index
  plt.figure(figsize=(3,3))
  plot = sns.countplot(data=temp.loc[temp['listed_in'].isin(cnt_listed_in)], x = 'listed_in', order=cnt_listed_i
  plot.set_xticklabels(labels=cnt_listed_in, rotation=90)
  plt.xlabel(f'listed_in for {cnt_val}')
  for p in plot.patches:
    plot.annotate('{:}'.format(p.get_height()), (p.get_x(), p.get_height()+0.02))
  plt.show()
```




2
2
2
2.0
1.5
1.0
0.5
0.0
count
Dramas
International Movies
Sports Movies
listed_in for Jamaica


5
5
5
4
3
2
1
0
count
Dramas
International Movies
listed_in for Latvia



Observation :

- Armenia, Mongolia, Bahamas and Montenegro comprises of the bottom 4 countries with least content
- There is definitely scope for increasing the content to attract subscribers and increase the potentially untapped markets

Correlation

Methodology :

- Pearson's ChiSquared test have been performed to understand if the variables are correlated with each other

- An alpha of 0.05 has been used to compare the test statistic generated from the chisquared test. Values lesser than alpha would infer variables being correlated with 95% confidence

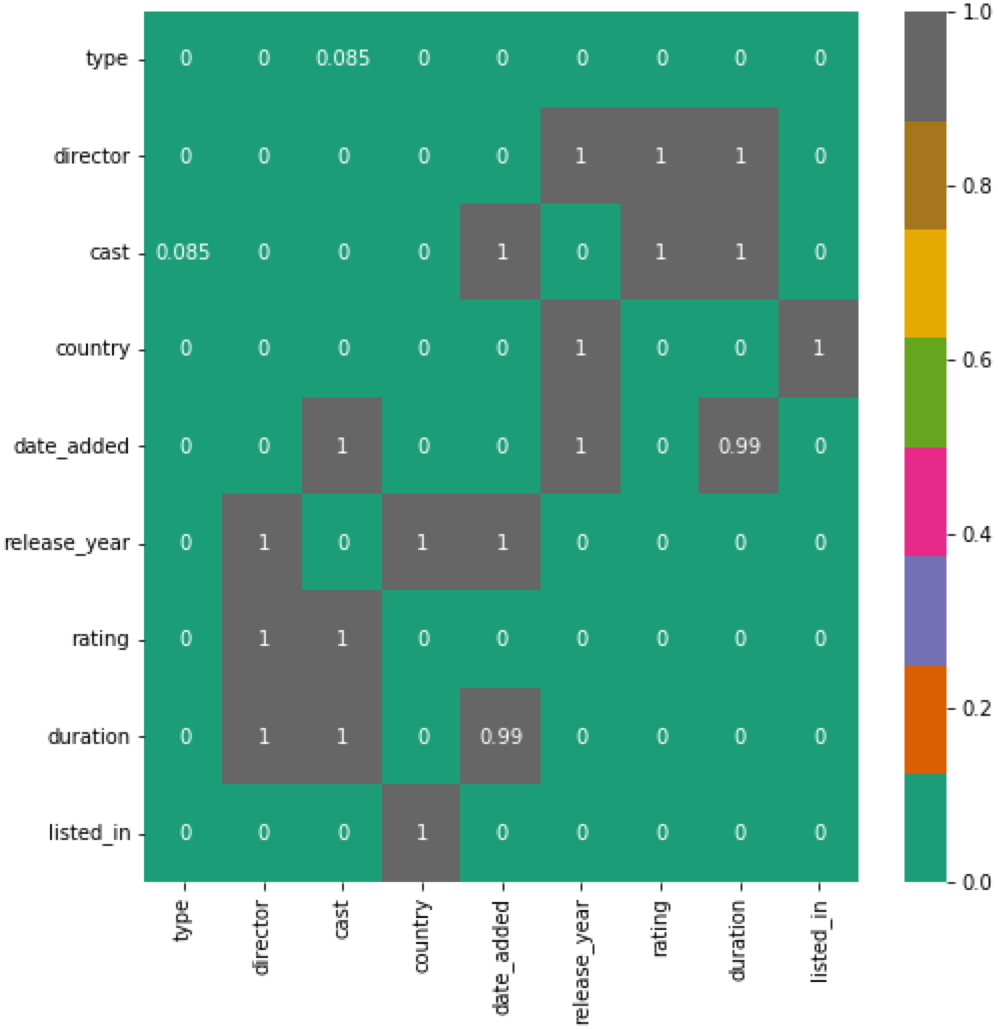
In [34]:

```python
correlation_master = master_dataset.drop(['show_id', 'title', 'description'], axis=1)
columns = correlation_master.columns
columns

col_length = len(columns)
correlation_df = pd.DataFrame(columns=columns, index=columns, dtype=object)
for col_num in range(col_length):

  correlation_df.iloc[col_num, col_num] = 0

  for next_col_num in range(col_num + 1, col_length):
    cross_tab_temp = pd.crosstab(correlation_master.iloc[:,col_num], correlation_master.iloc[:,next_col_num])
    stat, p, dof, expected = chi2_contingency(cross_tab_temp)
    test_stat = round(p, 5)
    correlation_df.iloc[col_num, next_col_num] = test_stat
    correlation_df.iloc[next_col_num, col_num] = test_stat

correlation_df = correlation_df.astype(float)
correlation_df
```

Out[34]:

	type	director	cast	country	date_added	release_year	rating	duration	listed_in
type	0.00000	0.0	0.08522	0.0	0.00000	0.0	0.0	0.00000	0.0
director	0.00000	0.0	0.00000	0.0	0.00000	1.0	1.0	1.00000	0.0
cast	0.08522	0.0	0.00000	0.0	1.00000	0.0	1.0	1.00000	0.0
country	0.00000	0.0	0.00000	0.0	0.00000	1.0	0.0	0.00000	1.0
date_added	0.00000	0.0	1.00000	0.0	0.00000	1.0	0.0	0.98637	0.0
release_year	0.00000	1.0	0.00000	1.0	1.00000	0.0	0.0	0.00000	0.0
rating	0.00000	1.0	1.00000	0.0	0.00000	0.0	0.0	0.00000	0.0
duration	0.00000	1.0	1.00000	0.0	0.98637	0.0	0.0	0.00000	0.0
listed_in	0.00000	0.0	0.00000	1.0	0.00000	0.0	0.0	0.00000	0.0

In [35]:
```
plt.figure(figsize=(8, 8))
sns.heatmap(correlation_df, annot=True, cmap='Dark2')
plt.show()
```



	type	director	cast	country	date_added	release_year	rating	duration	listed_in
type	0	0	0.085	0	0	0	0	0	0
director	0	0	0	0	0	1	1	1	0
cast	0.085	0	0	0	1	0	1	1	0
country	0	0	0	0	0	1	0	0	1
date_added	0	0	1	0	0	1	0	0.99	0
release_year	0	1	0	1	1	0	0	0	0
rating	0	1	1	0	0	0	0	0	0
duration	0	1	1	0	0.99	0	0	0	0
listed_in	0	0	0	1	0	0	0	0	0

Observation :

- Variables haing values less than 0.05 are considered correlated with 95% confidence

- type seems to be correlated with all the variables barring cast and month_added
- director seems to be correlated with all the variables barring release_year, rating and duration
- cast seems to be correlated with director, country, release_year, listed_in and month_added
- country seems to correlated with everything except release_year and listed_in
- date_added does not seem to correlated with cast and duraion. Surprisingly it is not correlated with release_year which could be due to the fact that Netflix keeps adding previously released contents
- release_year does not seem to be correlated with country, director and date_added
- rating does not seem to be correlated with director and cast
- duration does not seem to be correlated with cast, director and date_added
- listed_in does not seem to be correlated with country
- month_added does not seem to be correlated with type

Handling Missing Values

Approach

- In general, mode i.e. the most popular value in the column is used to impute missing values in a categorical column
- In this case, considering the most popular director and replacing the same might not make sense as there are different geographies inolved
- Based on the correlation table and some domain knowledge, we have utilised 3 columns namely type, country and listed_in to find out the common values and then replace it within the group

In [29]:
```
director_grouped = master_dataset.groupby(['type', 'country', 'listed_in'])['director'].apply(pd.Series.mode).re
director_grouped.drop('level_3', axis = 1, inplace=True)

master_dataset = pd.merge(master_dataset, director_grouped, on=['type', 'country', 'listed_in'])
master_dataset.loc[master_dataset['director'].isna(),'director'] = master_dataset.loc[master_dataset['director']
print(master_dataset['director'].isna().sum())
```

```
0
```

Approach

- Based on the correlation table and some domain knowledge, we have utilised 2 columns namely country and listed_in to find out the common values and then replace it within the group
- Since the cast comprises or several actors, we have considered the top 5 values within the group

In [30]:

```
cast_grouped = master_dataset.groupby(['country', 'listed_in'])['cast'].agg(lambda x : ','.join(x.value_counts()

master_dataset = pd.merge(master_dataset, cast_grouped, on=['country', 'listed_in'])
master_dataset.loc[master_dataset['cast'].isna(),'cast'] = master_dataset.loc[master_dataset['cast'].isna(),'pop
print(master_dataset['cast'].isna().sum())
```

```
0
```

Approach

- Based on the correlation table and some domain knowledge, we have utilised 3 columns namely type, country and listed_in to find out the common values and then replace it within the group

In [31]:

```
rating_grouped = master_dataset.groupby(['type', 'country', 'listed_in'])['rating'].apply(pd.Series.mode).reset_
rating_grouped.drop('level_3', axis = 1, inplace=True)

master_dataset = pd.merge(master_dataset, rating_grouped, on=['type', 'country', 'listed_in'])
master_dataset.loc[master_dataset['rating'].isna(),'rating'] = master_dataset.loc[master_dataset['rating'].isna(
print(master_dataset['rating'].isna().sum())
```

```
0
```

Recommendations

- **Focus on Indian market**

  India being a growing market in the web space, there can be an increase in digital content especially TV shows.

  Recent study suggests that theatrical footfalls have drastically reduced since the pandemic outbreak and there has been an increase of digital consumption

Emergence of new players like Amazon play, Hotstar etc could become competitors in case the content volume and quantity is not taken care of

Reduction of subscription charge can also become a factor in the long run

- **Animation as a content is extremely low in numbers**

  In most of the countries. Recent studies suggest an increment in the consumption of Japanese animation content across geographies

  Such contents can be made available in all countries either with subtitles or regional dubbing

  Potential to increase young subscribers

  This will also increase content with less restrictive ratings as we have seen majority of the content are either for mature audiences or for restrictve watching

- **Increase of market share in untapped geographies**

  Around 53 countries, i.e. 42%, have only 3 or lesser content available

  Potentially these are huge untapped markets

  Most of the countries are either developing or third world countries hence high subscription cost might be an issue

  An experimental idea would be is to keep minimum subscription charge and increase customer base

  The library can be increased by using the pre existing popular contents which would not substantially increase the cost and hence can be sustainable