Netflix casestudy

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
df=pd.read_csv('netflix.csv')
```

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

Out[ ]:

	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	PG
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	

In [ ]:
```
df.tail(1)
```

Out[ ]:

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

In [ ]:
```
```

Q1. Defining Problem Statement and Analysing basic metrics

What we want to do with the given dataset

1. Analyze the data to generate insights on the types of shows and movies Netflix should produce.
2. Identify strategies for growing Netflix's business in different countries.

3. Provide recommendations for tailoring content based on regional preferences and demand.

In [ ]:
```
df.head(3)
# I can see many missing data
#Duration is not correct some are given as Seasons
```

Out[ ]:

	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

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

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

In [ ]:
```
df.info()
#Below is the info like total rows=0 to 8806 and cloumns 12
```

```
<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 [ ]:
```
df.nunique()
# we can observe the the unique values present in each column
# Main thing to observe is their are 2 "TYPES" those are movies and TV shows
```

Out[ ]:

	0
show_id	8807
type	2
title	8807
director	4528
cast	7692
country	748
date_added	1767
release_year	74
rating	17
duration	220
listed_in	514
description	8775

dtype: int64

In [ ]:
```
df[df.duplicated()]
#we can observe as a whole their are no duplicates
#But if we take each column we can find duplicate
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	durati

In [ ]:
```
#null values
df.isnull().sum()
# what i observe is max null values are in director column
''''''''''
 we have to make sure that director names are their because many of the audience
 might like the director if they scearch for movies on the base of director they

 ''''''''''
```

Out[ ]:

	0
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

After basic analisis what i observe is

1. Their are 8807 entries ans 12 columns 2.Their are null vaues in Director,rating,cast,country,date added columns

which can cause a negetive side like : if we take situations were directors are missing

- Many viewers choose to watch content based on their preference for a particular director.
- If director names are missing, viewers may have a negative experience when searching for content.
- Properly displaying director names helps to meet the expectations of audiences who value directors when selecting movies or shows.

This can be case for cast

Q2. 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 [ ]:
```
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 [ ]:
```
df.nunique()
```

Out[ ]:

	0
show_id	8807
type	2
title	8807
director	4528
cast	7692
country	748
date_added	1767
release_year	74
rating	17
duration	220
listed_in	514
description	8775

dtype: int64

In [ ]:
```
df.nunique().index
```

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

In [ ]:
```
#we can see that their are missing values below for columns.
df[df['director'].isnull()]
df[df['cast'].isnull()]
df[df['country'].isnull()]
df[df['date_added'].isnull()]
```

```
df[df['rating'].isnull()]
df[df['duration'].isnull()]
```

Out[ ]:

	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	April 4, 2017	2017	
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015	

In [ ]:
```
# Iam changing the type column from object to category
df['type']=df['type'].astype('category')
```

In [ ]:
```
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   category
 2   title         8807 non-null   object  
 3   director      6173 non-null   object  
 4   cast          7982 non-null   object  
 5   country       7976 non-null   object  
 6   date_added    8797 non-null   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: category(1), int64(1), object(10)
memory usage: 765.7+ KB
```

In [ ]:
```
# i wolud like to mention Missing insted of NaN
df.fillna('Missing',inplace=True)
```

In [ ]:
```
# Different type of statistical for numerical ,categorical columns
df['show_id'].describe()
df['type'].describe()
df['title'].describe()
df['director'].describe()
df['cast'].describe()
df['country'].describe()
df['date_added'].describe()
df['rating'].describe()
df['duration'].describe()
```

```
df['listed_in'].describe()
df['description'].describe()
df['release_year'].describe()
```

Out[ ]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

Let us take some statistical info to analyse further

In [ ]:
```
#Type their are two and on the top is movies with 6131
df['type'].describe()
```

Out[ ]:

	type
count	8807
unique	2
top	Movie
freq	6131

dtype: object

In [ ]:
```
#director their are 4528 directors and Rajiv Chilaka has directed more number or
df['director'].describe()
```

Out[ ]:

	director
count	6173
unique	4528
top	Rajiv Chilaka
freq	19

dtype: object

In [ ]:
```
#Countries their are 748 unique countries where US is on the top movies and TV.s
df['country'].describe()
```

Out[ ]:

	country
count	7976
unique	748
top	United States
freq	2818

dtype: object

In [ ]:
```
#max united states
df['country'].value_counts(ascending=False)
```

Out[ ]:

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

748 rows × 1 columns

dtype: int64

In [ ]:
```
#Duration as we can see it is writen season 1
df['duration'].describe()
df[df['duration']=='1 Season']
df[df['title']=='Blood & Water']
df['duration'].unique()
'''''
we can see that their are many unique values for duration but in this column the
to be changed as we have to give the correct duration so to that the viewer can
whach this

'''''
```

Out[ ]:
```
array(['90 min', '2 Seasons', '1 Season', '91 min', '125 min',
       '9 Seasons', '104 min', '127 min', '4 Seasons', '67 min', '94 min',
       '5 Seasons', '161 min', '61 min', '166 min', '147 min', '103 min',
       '97 min', '106 min', '111 min', '3 Seasons', '110 min', '105 min',
       '96 min', '124 min', '116 min', '98 min', '23 min', '115 min',
       '122 min', '99 min', '88 min', '100 min', '6 Seasons', '102 min',
       '93 min', '95 min', '85 min', '83 min', '113 min', '13 min',
       '182 min', '48 min', '145 min', '87 min', '92 min', '80 min',
       '117 min', '128 min', '119 min', '143 min', '114 min', '118 min',
       '108 min', '63 min', '121 min', '142 min', '154 min', '120 min',
       '82 min', '109 min', '101 min', '86 min', '229 min', '76 min',
       '89 min', '156 min', '112 min', '107 min', '129 min', '135 min',
       '136 min', '165 min', '150 min', '133 min', '70 min', '84 min',
       '140 min', '78 min', '7 Seasons', '64 min', '59 min', '139 min',
       '69 min', '148 min', '189 min', '141 min', '130 min', '138 min',
       '81 min', '132 min', '10 Seasons', '123 min', '65 min', '68 min',
       '66 min', '62 min', '74 min', '131 min', '39 min', '46 min',
       '38 min', '8 Seasons', '17 Seasons', '126 min', '155 min',
       '159 min', '137 min', '12 min', '273 min', '36 min', '34 min',
       '77 min', '60 min', '49 min', '58 min', '72 min', '204 min',
       '212 min', '25 min', '73 min', '29 min', '47 min', '32 min',
       '35 min', '71 min', '149 min', '33 min', '15 min', '54 min',
       '224 min', '162 min', '37 min', '75 min', '79 min', '55 min',
       '158 min', '164 min', '173 min', '181 min', '185 min', '21 min',
       '24 min', '51 min', '151 min', '42 min', '22 min', '134 min',
       '177 min', '13 Seasons', '52 min', '14 min', '53 min', '8 min',
       '57 min', '28 min', '50 min', '9 min', '26 min', '45 min',
       '171 min', '27 min', '44 min', '146 min', '20 min', '157 min',
       '17 min', '203 min', '41 min', '30 min', '194 min', '15 Seasons',
       '233 min', '237 min', '230 min', '195 min', '253 min', '152 min',
       '190 min', '160 min', '208 min', '180 min', '144 min', '5 min',
       '174 min', '170 min', '192 min', '209 min', '187 min', '172 min',
       '16 min', '186 min', '11 min', '193 min', '176 min', '56 min',
       '169 min', '40 min', '10 min', '3 min', '168 min', '312 min',
       '153 min', '214 min', '31 min', '163 min', '19 min', '12 Seasons',
       nan, '179 min', '11 Seasons', '43 min', '200 min', '196 min',
       '167 min', '178 min', '228 min', '18 min', '205 min', '201 min',
       '191 min'], dtype=object)
```

In [ ]:
```
#WE can see the count of relesedate ,mean etc
df['release_year'].describe()
```

Out[ ]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

dtype: float64

In this analysis i have observed is :

1. Their are duplicates present in the data that is why count is not maching the unique.
2. most of the movies are realised and producde in US.
3. we can see that their are many unique values for duration but in this column the value 1 season is highest frequency this needs to be changed as we have to give the correct duration so to that the viewer can have a idea how much time that the he/she can take out to whach this.
4. I have changed the type column as category.

Q3. Non-Graphical Analysis: Value counts and unique attributes

I have done some analysis above that comes under this .

In [ ]:
```
df.head(1)
```

Out[ ]:

	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

In [ ]:
```
# the below lines tell me the categorical data to understand the distribution of
df['show_id'].value_counts()
df['type'].value_counts()
df['title'].value_counts()
df['director'].value_counts()
df['cast'].value_counts()
df['country'].value_counts()
df['date_added'].value_counts()
```

```
df['rating'].value_counts()
df['duration'].value_counts()
df['listed_in'].value_counts()
df['description'].value_counts()
df['release_year'].value_counts()
```

Out[ ]:

release_year	count
2018	1147
2017	1032
2019	1030
2020	953
2016	902
...	...
1959	1
1925	1
1961	1
1947	1
1966	1

74 rows × 1 columns

dtype: int64

In [ ]:
```
#Year where max number of movies release 2018
df['release_year'].value_counts(ascending=False)
```

Out[ ]:

release_year	count
2018	1147
2017	1032
2019	1030
2020	953
2016	902
...	...
1959	1
1925	1
1961	1
1947	1
1966	1

74 rows × 1 columns

dtype: int64

In [ ]:
```
df['release_year'].describe()
#We can see the metrics below for release_year
```

Out[ ]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

dtype: float64

In [ ]:
```
#Maximum nunber of movies directed is Rajiv Chilaka
df['director'].value_counts(ascending=False)
```

Out[ ]:

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

4528 rows × 1 columns

dtype: int64

In [ ]:
```
df['show_id'].unique()
df['type'].unique()
df['title'].unique()
df['director'].unique()
df['cast'].unique()
df['country'].unique()
df['date_added'].unique()
df['rating'].unique()
df['duration'].unique()
df['listed_in'].unique()
df['description'].unique()
df['release_year'].unique()
```

Out[ ]:
```
array(['As her father nears the end of his life, filmmaker Kirsten Johnson stag
es his death in inventive and comical ways to help them both face the inevitabl
e.',
       'After crossing paths at a party, a Cape Town teen sets out to prove whe
ther a private-school swimming star is her sister who was abducted at birth.',
       'To protect his family from a powerful drug lord, skilled thief Mehdi an
d his expert team of robbers are pulled into a violent and deadly turf war.',
       ...,
       'Looking to survive in a world taken over by zombies, a dorky college st
udent teams with an urban roughneck and a pair of grifter sisters.',
       'Dragged from civilian life, a former superhero must train a new crop of
youthful saviors when the military preps for an attack by a familiar villain.',
       "A scrappy but poor boy worms his way into a tycoon's dysfunctional fami
ly, while facing his fear of music and the truth about his past."],
      dtype=object)
```

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

Out[ ]:
```
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 [ ]:
```
df['director'].unique()
```

Out[ ]:
```
array(['Kirsten Johnson', nan, 'Julien Leclercq', ..., 'Majid Al Ansari',
       'Peter Hewitt', 'Mozez Singh'], dtype=object)
```

In [ ]:
```
df['date_added'].unique()
```

Out[ ]:
```
array(['September 25, 2021', 'September 24, 2021', 'September 23, 2021',
       ..., 'December 6, 2018', 'March 9, 2016', 'January 11, 2020'],
      dtype=object)
```

In [ ]:
```
df['duration'].unique()
```

Out[ ]:
```
array(['90 min', '2 Seasons', '1 Season', '91 min', '125 min',
       '9 Seasons', '104 min', '127 min', '4 Seasons', '67 min', '94 min',
       '5 Seasons', '161 min', '61 min', '166 min', '147 min', '103 min',
       '97 min', '106 min', '111 min', '3 Seasons', '110 min', '105 min',
       '96 min', '124 min', '116 min', '98 min', '23 min', '115 min',
       '122 min', '99 min', '88 min', '100 min', '6 Seasons', '102 min',
       '93 min', '95 min', '85 min', '83 min', '113 min', '13 min',
       '182 min', '48 min', '145 min', '87 min', '92 min', '80 min',
       '117 min', '128 min', '119 min', '143 min', '114 min', '118 min',
       '108 min', '63 min', '121 min', '142 min', '154 min', '120 min',
       '82 min', '109 min', '101 min', '86 min', '229 min', '76 min',
       '89 min', '156 min', '112 min', '107 min', '129 min', '135 min',
       '136 min', '165 min', '150 min', '133 min', '70 min', '84 min',
       '140 min', '78 min', '7 Seasons', '64 min', '59 min', '139 min',
       '69 min', '148 min', '189 min', '141 min', '130 min', '138 min',
       '81 min', '132 min', '10 Seasons', '123 min', '65 min', '68 min',
       '66 min', '62 min', '74 min', '131 min', '39 min', '46 min',
       '38 min', '8 Seasons', '17 Seasons', '126 min', '155 min',
       '159 min', '137 min', '12 min', '273 min', '36 min', '34 min',
       '77 min', '60 min', '49 min', '58 min', '72 min', '204 min',
       '212 min', '25 min', '73 min', '29 min', '47 min', '32 min',
       '35 min', '71 min', '149 min', '33 min', '15 min', '54 min',
       '224 min', '162 min', '37 min', '75 min', '79 min', '55 min',
       '158 min', '164 min', '173 min', '181 min', '185 min', '21 min',
       '24 min', '51 min', '151 min', '42 min', '22 min', '134 min',
       '177 min', '13 Seasons', '52 min', '14 min', '53 min', '8 min',
       '57 min', '28 min', '50 min', '9 min', '26 min', '45 min',
       '171 min', '27 min', '44 min', '146 min', '20 min', '157 min',
       '17 min', '203 min', '41 min', '30 min', '194 min', '15 Seasons',
       '233 min', '237 min', '230 min', '195 min', '253 min', '152 min',
       '190 min', '160 min', '208 min', '180 min', '144 min', '5 min',
       '174 min', '170 min', '192 min', '209 min', '187 min', '172 min',
       '16 min', '186 min', '11 min', '193 min', '176 min', '56 min',
       '169 min', '40 min', '10 min', '3 min', '168 min', '312 min',
       '153 min', '214 min', '31 min', '163 min', '19 min', '12 Seasons',
       nan, '179 min', '11 Seasons', '43 min', '200 min', '196 min',
       '167 min', '178 min', '228 min', '18 min', '205 min', '201 min',
       '191 min'], dtype=object)
```

In [ ]:
```
df.groupby('type')['title'].value_counts()
df['type'].value_counts()
```

Out[ ]:

type	count
Movie	6131
TV Show	2676

dtype: int64

In [ ]:
```
#max number of shows
df[df['type']=='TV Show']['director'].value_counts(ascending=False)
```

Out[ ]:

director	count
Alastair Fothergill	3
Rob Seidenglanz	2
Hsu Fu-chun	2
Iginio Straffi	2
Shin Won-ho	2
...	...
Juliana Vicente	1
Chang Chin-jung, Chen Rong-hui	1
Thierry Demaizière, Alban Teurlai	1
Manolo Caro	1
Michael Cumming	1

222 rows × 1 columns

dtype: int64

In [ ]:
```
# max number of movies
df[df['type']=='Movie']['director'].value_counts(ascending=False)
```

Out[ ]:

director	count
Rajiv Chilaka	19
Raúl Campos, Jan Suter	18
Suhas Kadav	16
Marcus Raboy	15
Jay Karas	14
...	...
Dennis Rovira van Boekholt	1
Naoto Amazutsumi	1
Jenny Gage	1
Kaila York	1
Mozez Singh	1

4354 rows × 1 columns

dtype: int64

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

Out[ ]:

	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

In [6]:
```
'''''
We can see that the listed_in is type of contnet we can make into diffrent catig
'''''
df['listed_in']=df['listed_in'].apply(lambda x:x.split(',')).explode(',')
```

In [ ]:
```
'''''
Below are top 5 in the count in the listed_in
'''''
df['listed_in'].value_counts().head()
```

Out[ ]:

listed_in	count
International Movies	1162
Dramas	665
Comedies	540
International TV Shows	410
TV Dramas	374

dtype: int64

In [ ]:
```
data_movies=df[df['type']=='Movie']
data_TVshows=df[df['type']=='TV Show']
```

In [ ]:
```
data_movies['listed_in'].value_counts()
''''
Below i can observe that their are some tv dramas and also all the names with TV
''''
```

Out[ ]:

listed_in	count
International Movies	809
Dramas	468
Comedies	378
TV Dramas	285
International TV Shows	282
...	...
Documentaries	4
Cult Movies	2
Music & Musicals	1
Classic & Cult TV	1
Stand-Up Comedy	1

68 rows × 1 columns

dtype: int64

In [ ]:
```
data_TVshows['listed_in'].value_counts()
```

Out[ ]:

listed_in	count
International Movies	353
Dramas	197
Comedies	162
International TV Shows	128
Dramas	128
...	...
Spanish-Language TV Shows	1
Classic & Cult TV	1
TV Shows	1
Romantic Movies	1
LGBTQ Movies	1

67 rows × 1 columns

dtype: int64

What i observe in the analysis is :

1. While doing unique i have found some data is the cloumn which is appropriate to the column which needs to be changed or have to replace with some of the other values like missing etc.
2. Maximum of each column like max number of movies and shows are released in 2018 and max number movies is directed by RajivChilaka and the max number show is directed by Alastair Fothergill.
3. In both movies and shows International Movies are comming first because perticular country people are intrested in other country films.

In [ ]:

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

Note: Pre-processing involves unnesting of the data in columns like Actor, Director, Country

4.1 For continuous variable(s): Distplot, countplot, histogram for univariate analysis

4.2 For categorical variable(s): Boxplot

4.3 For correlation: Heatmaps, Pairplots

In [3]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
df.tail(1)
```

Out[ ]:

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

Univariate analysis

1. Numerical(date_added,release_year,duration)
2. Ctogarical(type)

In [ ]:
```
df['date_added'].value_counts(ascending=False)
# January 1, 2020 is more added count on total
```

Out[ ]:

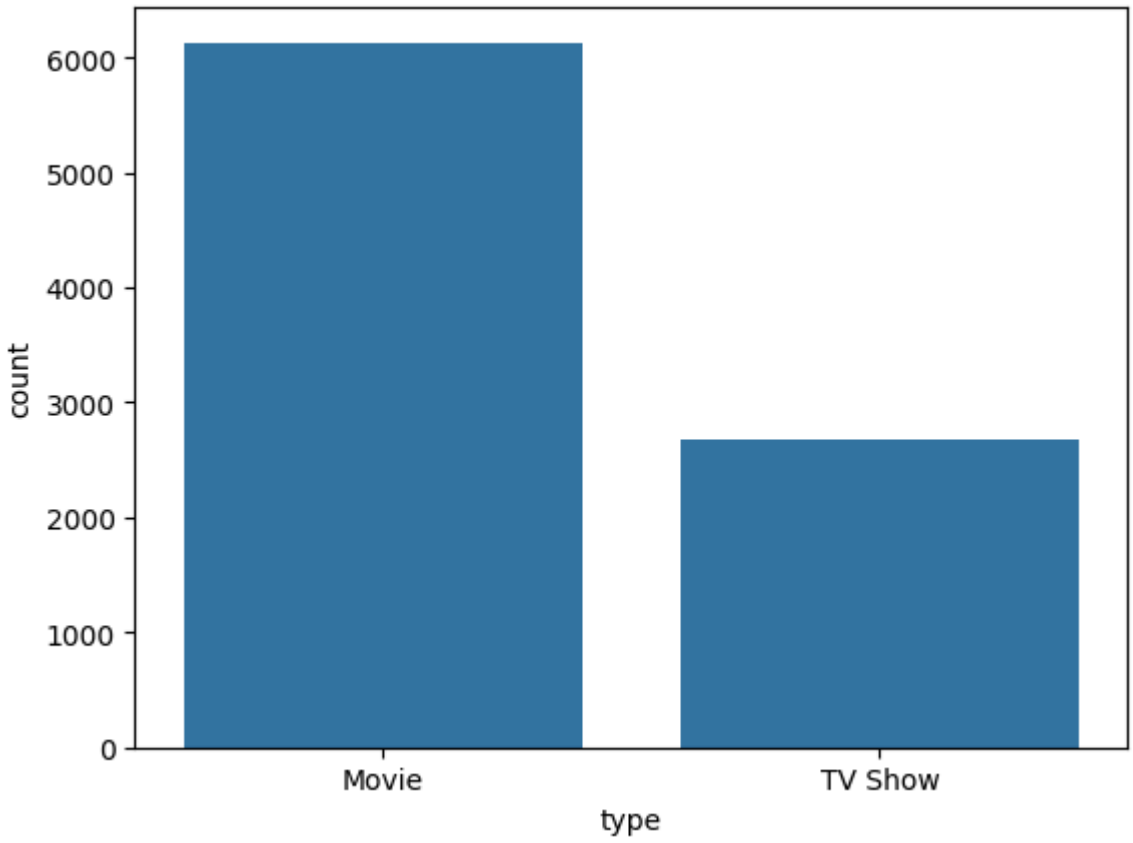
date_added	count
January 1, 2020	109
November 1, 2019	89
March 1, 2018	75
December 31, 2019	74
October 1, 2018	71
...	...
December 4, 2016	1
November 21, 2016	1
November 19, 2016	1
November 17, 2016	1
January 11, 2020	1

1767 rows × 1 columns

dtype: int64

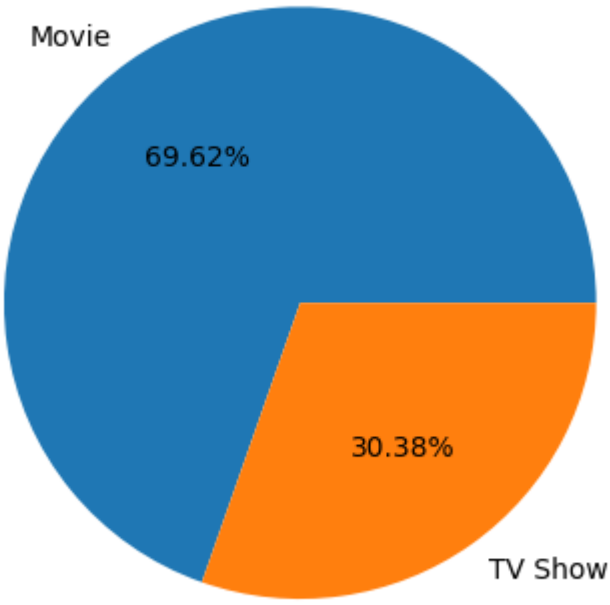
In [ ]:
```
sns.countplot(x='type', data=df, order=df['type'].value_counts(ascending=False).
```

Out[ ]:
```
<Axes: xlabel='type', ylabel='count'>
```



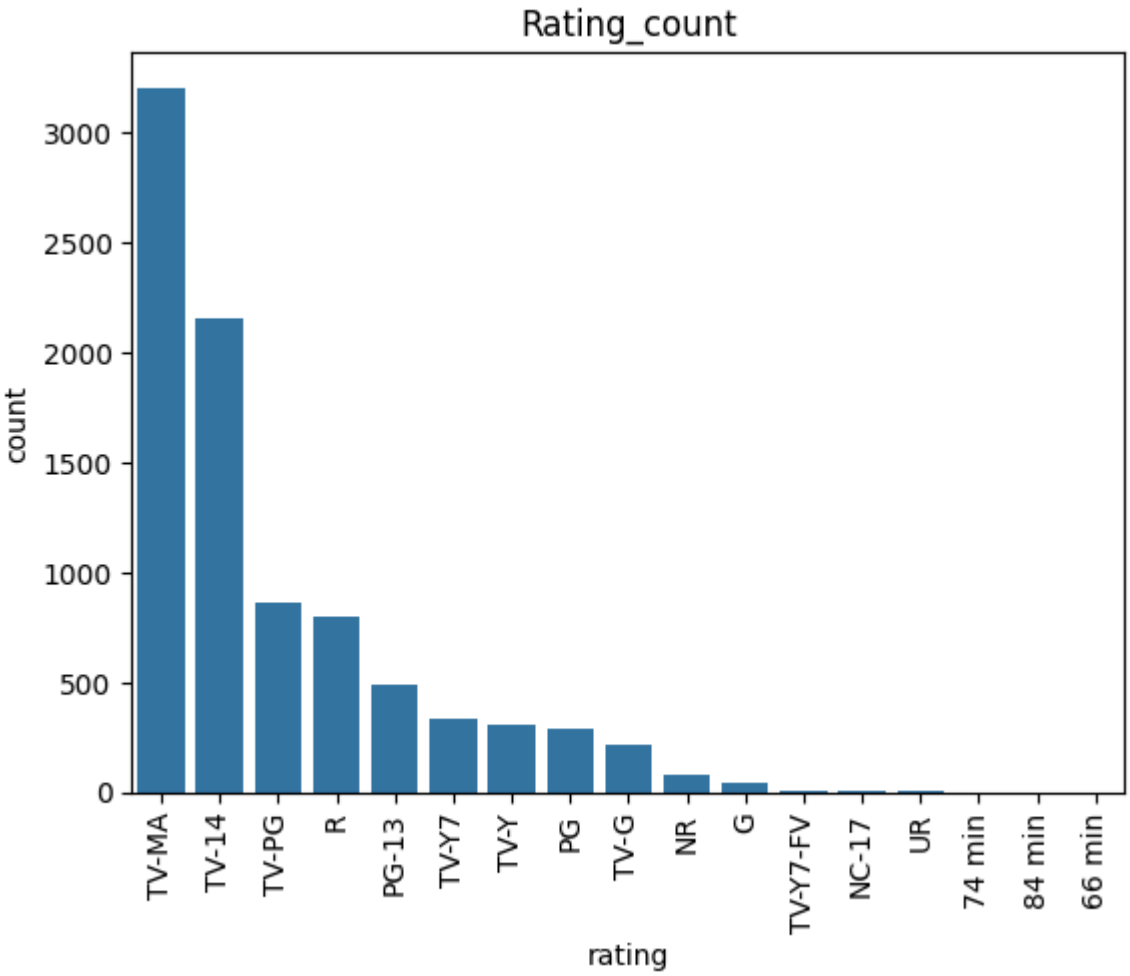
In [ ]:
```
#pie chart on types
data=df['type'].value_counts()
plt.pie(data, labels=data.index, autopct='%.2f%%' )
plt.show()
''''
Here we can see the percentage in a pie chart where movies and TV.Shows
we can see that movie percentage are higher

''''
```

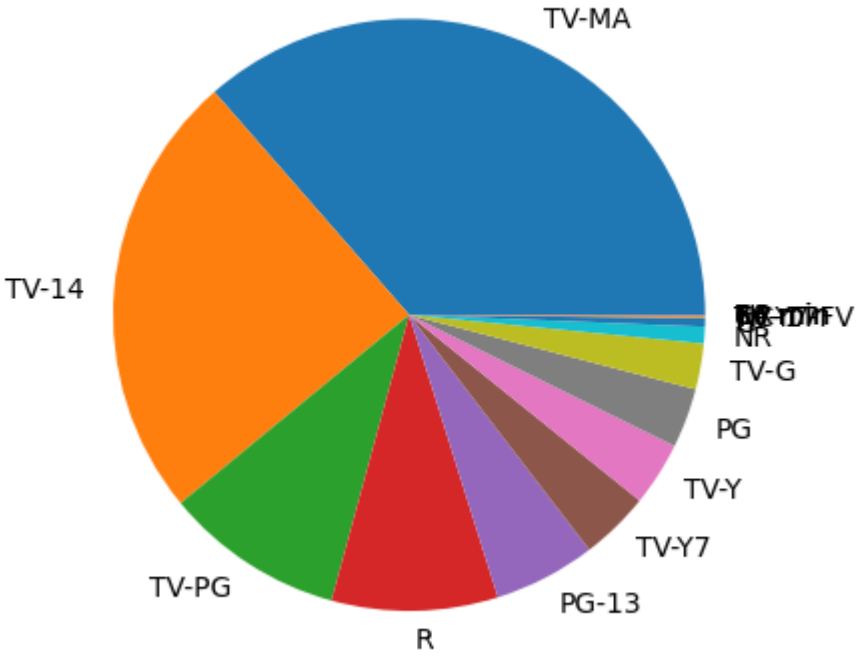


In [ ]:
```
sns.countplot(x='rating', data=df, order=df['rating'].value_counts(ascending=Fal
plt.xticks(rotation=90)
plt.title('Rating_count')
plt.show()

'''''
Max number of movies are rated as TV-MA
'''''
```



In [ ]:
```
deep=df['rating'].value_counts()
plt.pie(deep,labels=df['rating'].value_counts().index)
plt.show()
```



In [ ]:
```
data_m=data_movies['listed_in'].value_counts().head(5)
```

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

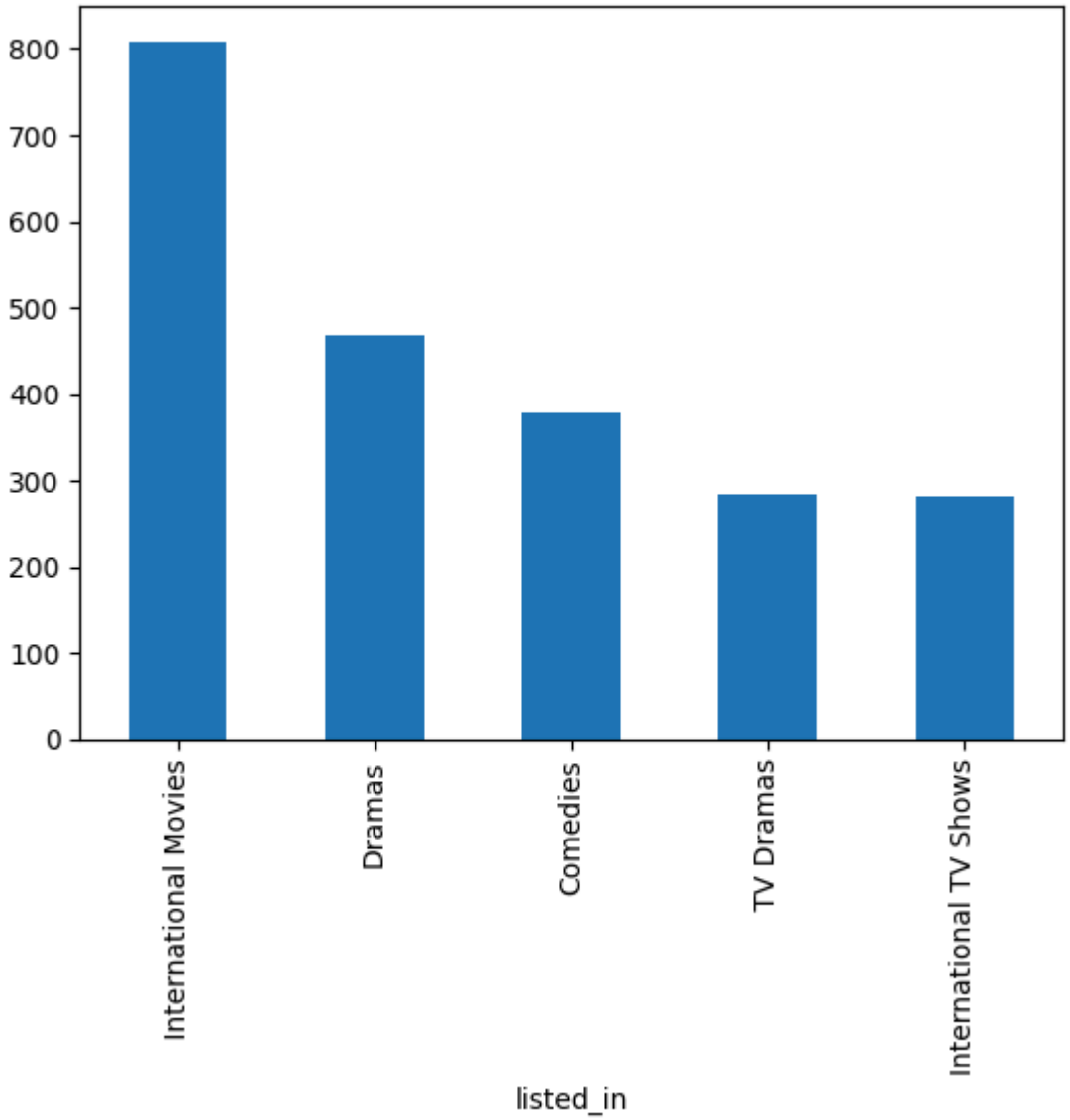
Out[ ]:

listed_in	count
International Movies	809
Dramas	468
Comedies	378
TV Dramas	285
International TV Shows	282

dtype: int64

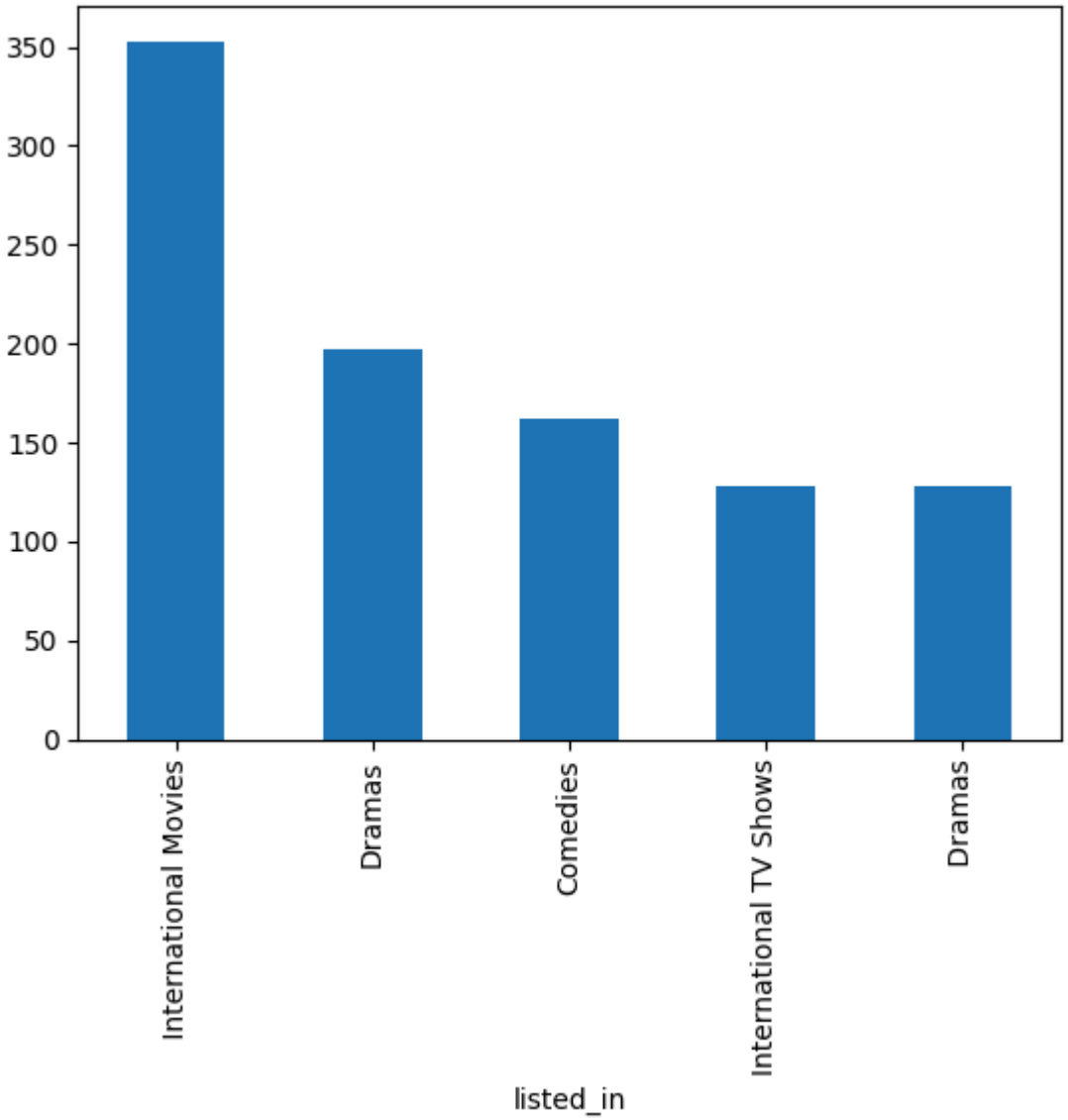
In [ ]:
```
data_movies['listed_in'].value_counts().head(5).plot(kind='bar')
'''
We can observe that in movies International Movies can be recomended
'''
```

Out[ ]:
```
<Axes: xlabel='listed_in'>
```



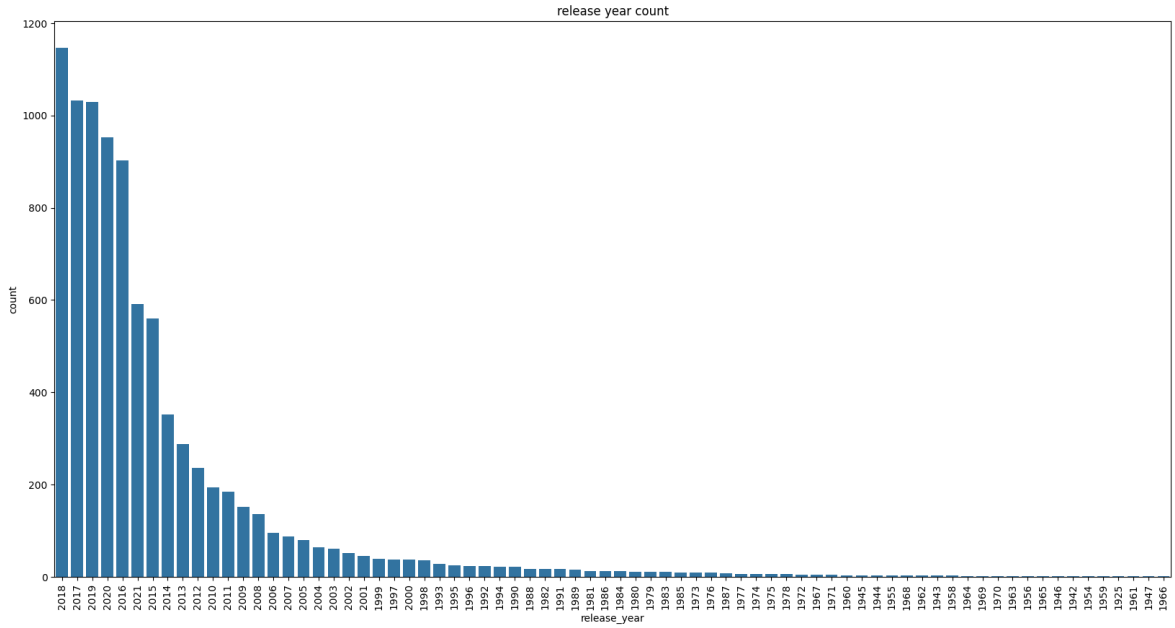
In [ ]:
```
data_TVshows['listed_in'].value_counts().head(5).plot(kind='bar')
''''
We can observe that in movies International Movies can be recomended for shows a
''''
```

Out[ ]: <Axes: xlabel='listed_in'>



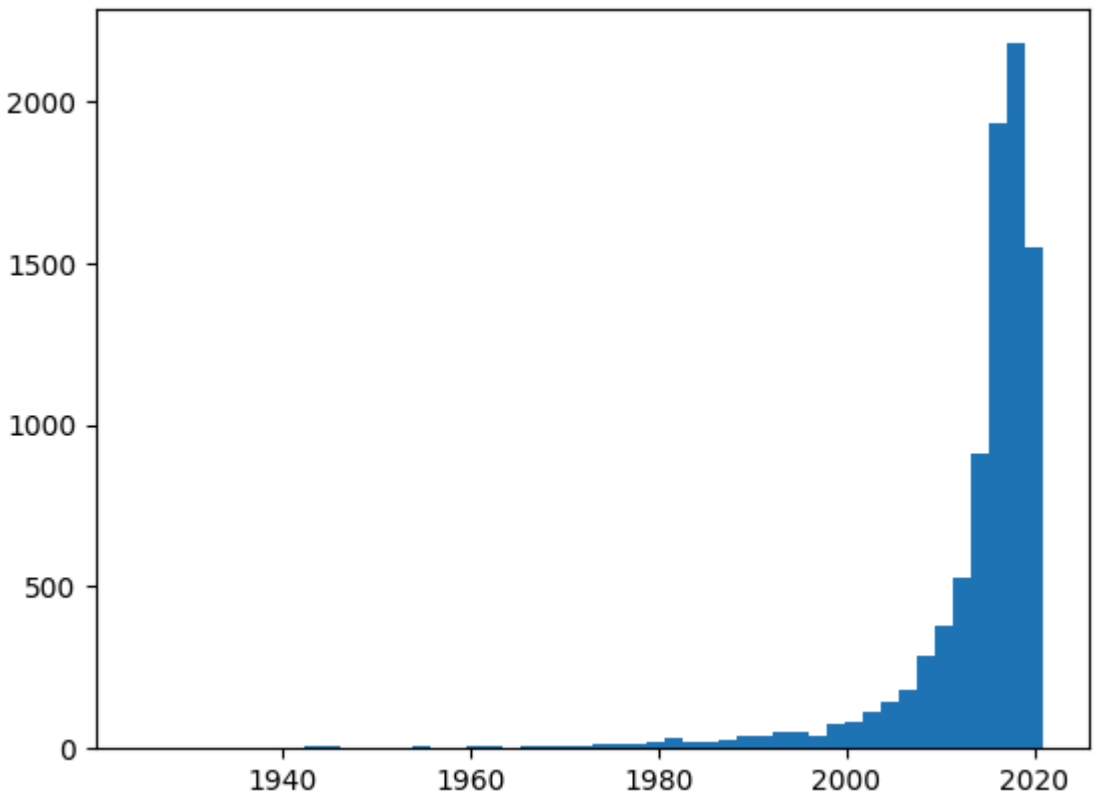
In [ ]:
```
plt.figure(figsize=(20,10))
sns.countplot(x='release_year', data=df, order=df['release_year'].value_counts(a
plt.xticks(rotation=90)
plt.title('release year count')
plt.show()

'''''
As we can see 2018 max number of movies is produced
'''''
```



In [ ]:
```
#matplot
plt.hist(df['release_year'],bins=50)
plt.show()

'''''
We can see the histogram view as below
'''''
```



In [ ]:
```
df.tail(1)
```

Out[ ]:

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

For categorical variable(s): Boxplot

In [ ]:
```
data=df[df['type']=='Movie']
```

In [ ]:
```
data.tail(1)
```

Out[ ]:

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

In [ ]:
```
data['release_year'].value_counts(ascending=False).head(5)
#max number of movies relased are in 2017
```

Out[ ]:

release_year	count
2017	767
2018	767
2016	658
2019	633
2020	517

dtype: int64

In [ ]:
```
data_TV_shows=df[df['type']=='TV Show']
```

In [ ]:
```
data_TV_shows.tail(1)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7

In [ ]:
```
data_TV_shows['release_year'].value_counts(ascending=False).head(5)
#Max numberof TV_shows realsed in year 2020
```

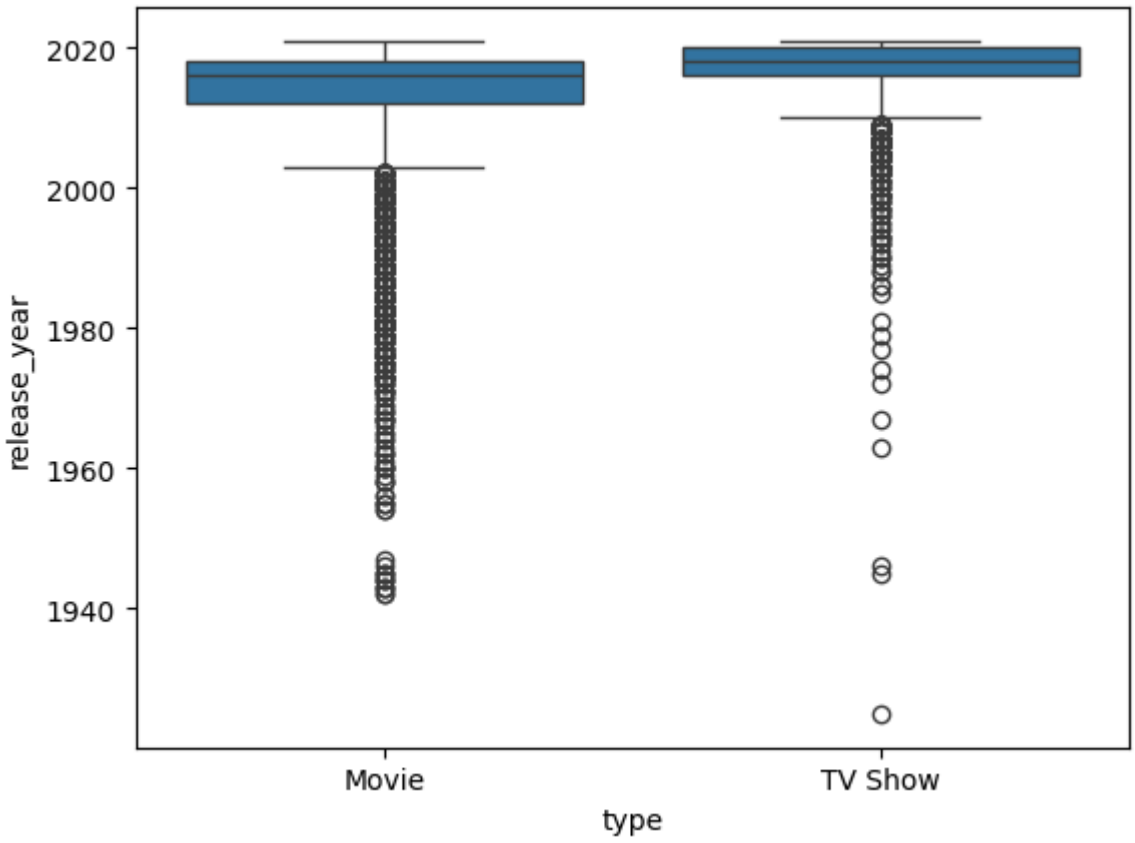
Out[ ]:

release_year	count
2020	436
2019	397
2018	380
2021	315
2017	265

dtype: int64

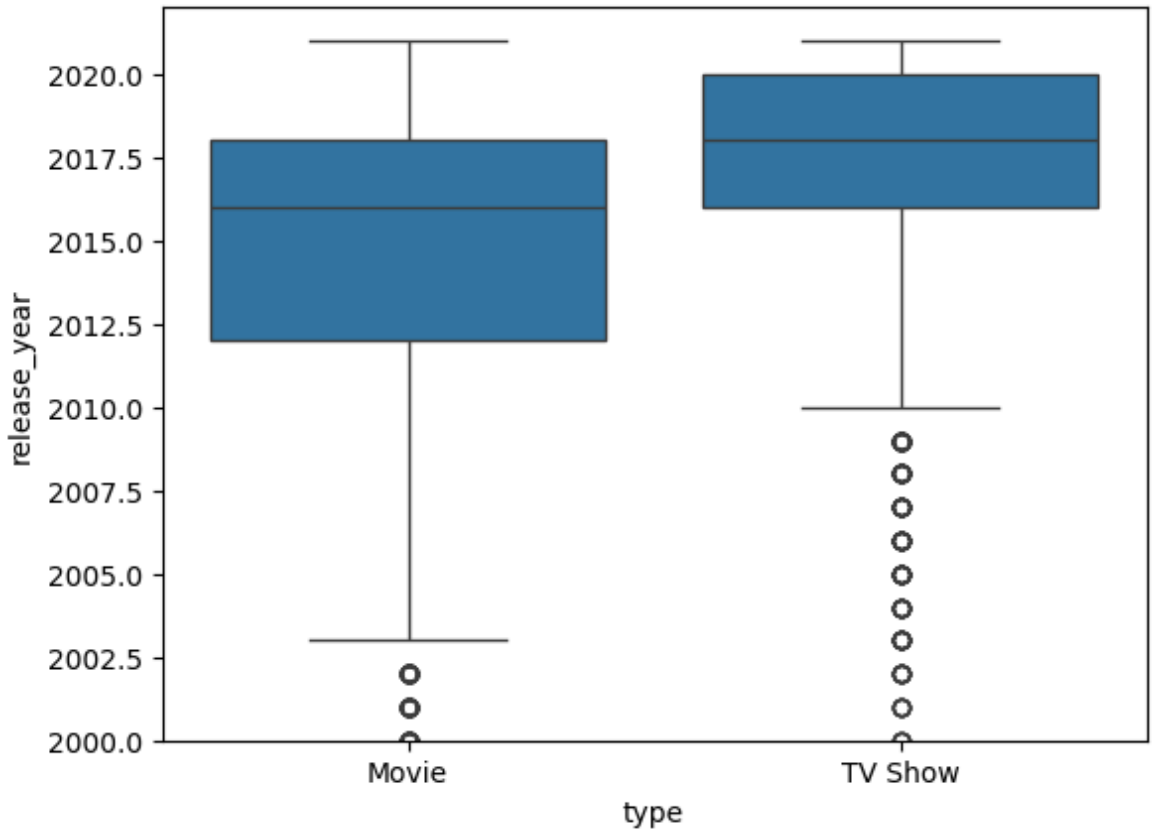
In [ ]:
```
sns.boxplot(x='type',y='release_year',data=df)
```

Out[ ]:
```
<Axes: xlabel='type', ylabel='release_year'>
```



As we can see more outliers in the above we can set range form (2000, 2022)

In [ ]:
```
'''''
As far as i can analyse the data taking form 2000 is better
'''''
sns.boxplot(x='type',y='release_year',data=df)
plt.ylim(2000, 2022)
plt.show()
```


2020.0
2017.5
2015.0
2012.5
2010.0
2007.5
2005.0
2002.5
2000.0
release_year
Movie
TV Show
type

In [ ]:
```
df
```

Out[ ]:

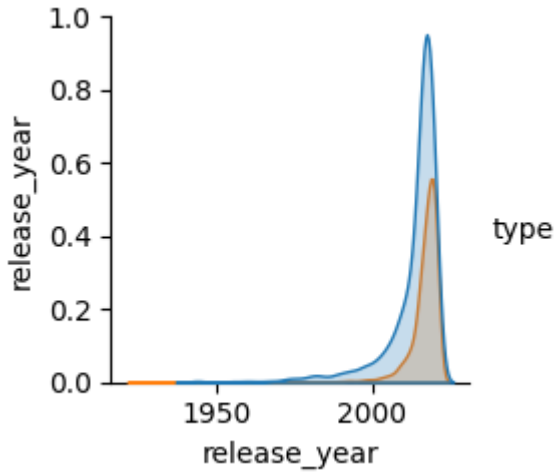
	show_id	type	title	director	cast	country	date_added	release_y
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2
2	s3	TV Show	Ganglands	Julien Leclercq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

	show_id	type	title	director	cast	country	date_added	release_y
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2

8807 rows × 12 columns

In [ ]:
```
sns.pairplot(data=df,hue='type')
```

Out[ ]:
```
<seaborn.axisgrid.PairGrid at 0x781870ecc890>
```



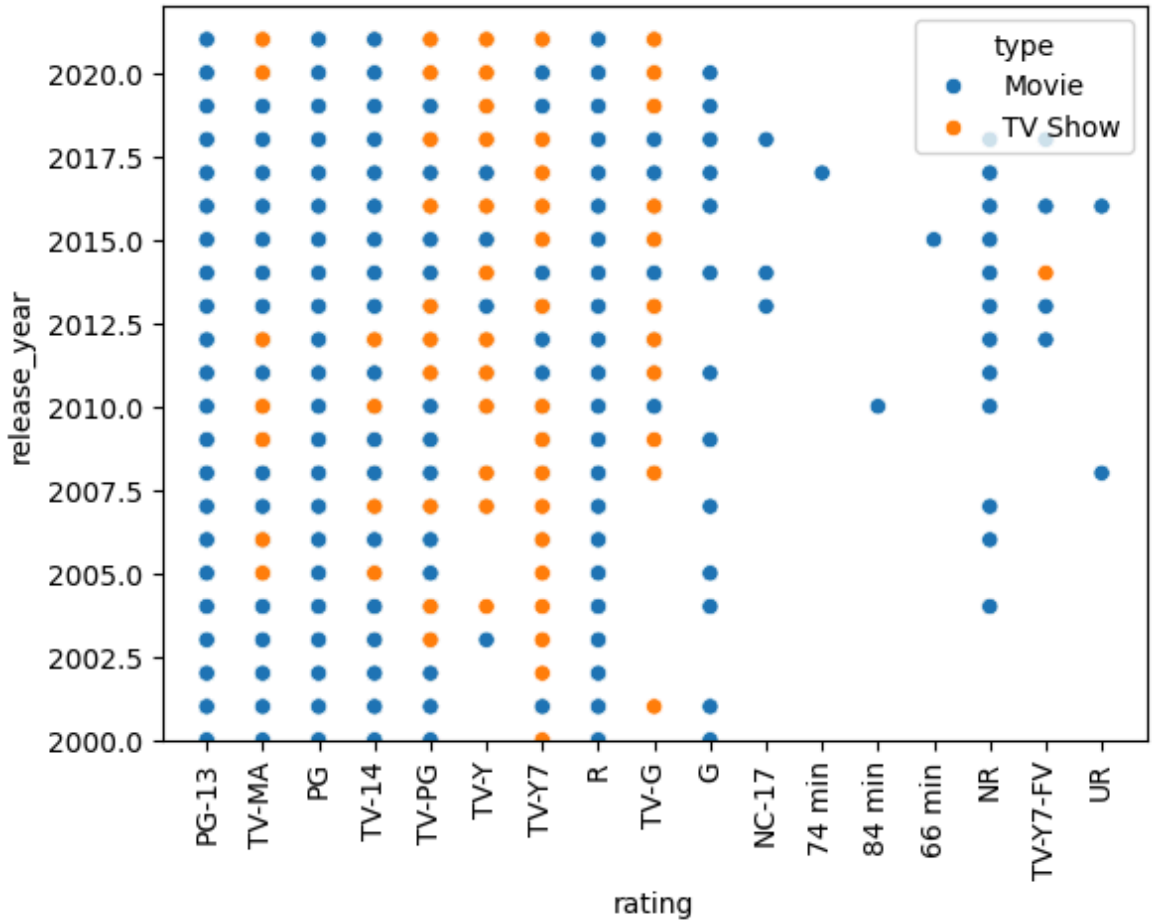
What i observe in the analysis is : 1.The percentage of movies is higher compared to TV shows, indicating that movies are more prominent in the dataset. 2.This suggests that a significant portion of the movies in the dataset are intended for mature audiences, with TV-MA being the dominant rating. 3.There seems to be a preference or high potential for international movies to be recommended, which applies to both movies and TV shows in the dataset. 4.marked the peak in movie production, and there has been a decline in movie production in the following years.

In [7]:
```
df.head(2)
```

Out[7]:

	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	

In [12]:
```
sns.scatterplot(x='rating',y='release_year',hue='type',data=df)
plt.xticks(rotation=90)
plt.ylim(2000, 2022)
plt.show()
```



5. Missing Value & Outlier check

I have done the missing Value analisys previously also here also i will do the analysis

In [ ]:
```
df.isnull().sum()
''''
We can see their are dfferent null values
''''
```

Out[ ]:

	0
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 [ ]:
```
#I want to fill the null value with missing data so that it will be easy for fur
df.fillna('Missing_data')
```

Out[ ]:

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

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

8807 rows × 12 columns

Listed_in should be more refined

Their are many null values in the columns which can be captured and worked upon the retreval form other sources for example director and cast can be retriving for third party sources so that the user will able to find the director and actors

In [ ]:

6. Insights based on Non-Graphical and Visual Analysis (10 Points)

6.1 Comments on the range of attributes

6.2 Comments on the distribution of the variables and relationship between them

6.3 Comments for each univariate and bivariate plot

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

Out[ ]:

	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	PG
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	

In [ ]:
```
df['release_year'].describe()
''''
We can see the distributions of release_year below
'''
```

Out[ ]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

dtype: float64

In [ ]:
```
data_movies['duration'].describe()
''''
the recomended duration of time to realse is 90min
'''
```

Out[ ]:

	duration
count	6128
unique	205
top	90 min
freq	152

dtype: object

In [ ]:
```
data_movies['country'].describe()
''''
Best country to produce is in United States
'''
```

Out[ ]:

	country
count	5691
unique	651
top	United States
freq	2058

dtype: object

In [ ]:
```
data_TVshows['country'].describe()
```

Out[ ]:

	country
count	2285
unique	196
top	United States
freq	760

dtype: object

I have given the commemnts for univariate and bivariant analisys in the above cells

7. Business Insights

1. Missing values in Director, Cast, Country, and Date Added can negatively impact user experience.
2. The "Duration" column shows 1 season as the most frequent value, which doesn't provide users with clear information about the content's length .
3. Duplicate entries lead to discrepancies between count and unique records in the dataset.
4. A significant number of movies and shows were produced in 2018, followed by a decline in production in subsequent years.
5. International movies are highly popular both in movies and TV shows, indicating a strong preference for global content.
6. The dataset shows that movies are more numerous than TV shows, suggesting a stronger preference for movies.
7. TV-MA is the dominant rating, suggesting a mature audience preference for movies.
8. Missing information (e.g., Director, Cast) can be filled using third-party sources like IMDB, enhancing content metadata.

In [ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

8.Recommendations

1. Fill missing Director and Cast data by pulling from third-party sources or using a default "missing" value.
2. Address missing Country data for better content discoverability in different markets.
3. Standardize duration data by showing actual runtime (hours/minutes for movies) or episode counts for TV shows. This helps users assess time commitment and enhances user satisfaction.

4. Remove duplicates and clean the dataset regularly to ensure data consistency.This improves search accuracy and enhances content recommendations.
5. Revisit reasons behind the decline in production and adjust strategy accordingly.
6. Invest in and promote international content due to its high viewership.
7. Focus on promoting more movies while balancing TV shows to maintain long-term engagement
8. Create region-based categories, collaborate with international production houses, and target marketing efforts in specific regions
9. Integrate third-party APIs (e.g., IMDB) to automatically fill missing Director, Cast, and Country data.
10. Expand international content offerings and highlight it in recommendations.