NETFLIX CASE STUDY

READ DATA & IMPORT PACKAGES

In [451…

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

In [452…

```
df = pd.read_csv(r"C:\Users\netflix.csv")
```

In [453…

```
df.head(5)
```

Out[453]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	du
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	9
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	Se
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 S
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 S
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	Se

Basic data information

In [454…

```
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
```

Number of missing values in each column

In [455…

```
df.isna().sum()
```

Out[455]:

```
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 [456…

```
# There are 12 columns in the dataset , all of them are objects except release_yea
 # the columns director, cast, country, date_added ,rating and duration contains nu
```

DATA CLEANING & FILLING NULL VALUES

Fill the null values in country and director columns with 'Unknown' and 'UNKNOWN'respectively , fill the cast null values with 'not available' , remove the null values of date_added , rating and duration.

In [457…

```
df['country'].fillna('Unknown',inplace = True)
df.dropna(subset = ['date_added','duration','rating'], inplace = True)
df['director'].fillna('UNKNOWN',inplace=True)
df['cast'].fillna('not available',inplace = True)
```

Change the data type of date_added to date time frame ,extract month and year from the date_added column

In [458…

```
df['date_added'] = pd.to_datetime(df['date_added'])
df['dateadd_month'] = df['date_added'].dt.month.astype(int)
df['dateadd_year'] = df['date_added'].dt.year.astype(int)
```

Extract the duration values from the duration column by splitting the numerical values from object and then convert the column to integer type.

In [459…

```
df['duration'] = df['duration'].apply(lambda x: x.split(" ")[0])
```

In [460…

```
df['duration'] = df['duration'].astype(int)
```

Check for missing values and datatype

In [461…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8790 entries, 0 to 8806
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   show_id        8790 non-null   object
 1   type           8790 non-null   object
 2   title          8790 non-null   object
 3   director       8790 non-null   object
 4   cast           8790 non-null   object
 5   country        8790 non-null   object
 6   date_added     8790 non-null   datetime64[ns]
 7   release_year   8790 non-null   int64
 8   rating         8790 non-null   object
 9   duration       8790 non-null   int32
 10  listed_in      8790 non-null   object
 11  description    8790 non-null   object
 12  dateadd_month  8790 non-null   int32
 13  dateadd_year   8790 non-null   int32
dtypes: datetime64[ns](1), int32(3), int64(1), object(9)
memory usage: 927.1+ KB
```

In [462…

```
df.shape
```

Out[462]:

```
(8790, 14)
```

In [463…

```
# Statistical summary
```

In [464…

```
df[df['type']=='Movie'].describe()
```

Out[464]:

	release_year	duration	dateadd_month	dateadd_year
count	6126.000000	6126.000000	6126.000000	6126.000000
mean	2013.120144	99.584884	6.609370	2018.851126
std	9.681723	28.283225	3.452541	1.561173
min	1942.000000	3.000000	1.000000	2008.000000
25%	2012.000000	87.000000	4.000000	2018.000000
50%	2016.000000	98.000000	7.000000	2019.000000
75%	2018.000000	114.000000	10.000000	2020.000000
max	2021.000000	312.000000	12.000000	2021.000000

In [465…

```
df[df['type']=='TV Show'].describe()
```

Out[465]:

	release_year	duration	dateadd_month	dateadd_year
count	2664.000000	2664.000000	2664.000000	2664.000000
mean	2016.627628	1.751877	6.762763	2018.925300
std	5.735194	1.550622	3.396231	1.600804
min	1925.000000	1.000000	1.000000	2008.000000
25%	2016.000000	1.000000	4.000000	2018.000000
50%	2018.000000	1.000000	7.000000	2019.000000
75%	2020.000000	2.000000	10.000000	2020.000000
max	2021.000000	17.000000	12.000000	2021.000000

In [466…

```
#The above data is free of null values and have the appropriate type for columns.
```

In [467…

```
df.head(5)
```

Out[467]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	not available	United States	2021-09-25	2020	PG-13
1	s2	TV Show	Blood & Water	UNKNOWN	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...	Unknown	2021-09-24	2021	TV-MA
3	s4	TV Show	Jailbirds New Orleans	UNKNOWN	not available	Unknown	2021-09-24	2021	TV-MA
4	s5	TV Show	Kota Factory	UNKNOWN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA

In the above data , duration column has values for movie in minutes and duration for TV Show is in number of seasons

UNNESTING OF DATA PRESENT IN COLUMNS 'cast' , 'director' ,'country', 'listed_in' for exploratory analysis.

In [468…

```
# Unnesting the cast
```

In [469…

```
df['cast'].apply(lambda x: str(x).split(', ')).tolist()
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)
df_new.reset_index(inplace=True)
df_new=df_new[['title',0]]
df_new.columns=['title','cast']
```

In [470…

```
df_new.head(5)
```

Out[470]:

	title	cast
0	Dick Johnson Is Dead	not available
1	Blood & Water	Ama Qamata
2	Blood & Water	Khosi Ngema
3	Blood & Water	Gail Mabalane
4	Blood & Water	Thabang Molaba

In [471…

```
# Unnesting the director
```

In [472…

```
df['director'].apply(lambda x: str(x).split(', ')).tolist()
constraint=df['director'].apply(lambda x: str(x).split(', ')).tolist()
df1=pd.DataFrame(constraint,index=df['title'])
df1=df1.stack()
df1=pd.DataFrame(df1)
df1.reset_index(inplace=True)
df1=df1[['title',0]]
df1.columns=['title','director']
```

In [473…

```
df1.head(5)
```

Out[473]:

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	UNKNOWN
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	UNKNOWN
4	Kota Factory	UNKNOWN

In [474…

```
# Unnesting the country
```

In [475…

```
df['country'].apply(lambda x: str(x).split(', ')).tolist()
constraint=df['country'].apply(lambda x: str(x).split(', ')).tolist()
df2=pd.DataFrame(constraint,index=df['title'])
df2=df2.stack()
df2=pd.DataFrame(df2)
df2.reset_index(inplace=True)
df2=df2[['title',0]]
df2.columns=['title','country']
```

In [476…

```
df2.head(5)
```

Out[476]:

	title	country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	Unknown
3	Jailbirds New Orleans	Unknown
4	Kota Factory	India

In [477…

```
# Unnesting the genre
```

In [478…

```
df['listed_in'].apply(lambda x: str(x).split(', ')).tolist()
constraint=df['listed_in'].apply(lambda x: str(x).split(', ')).tolist()
df3=pd.DataFrame(constraint,index=df['title'])
df3=df3.stack()
df3=pd.DataFrame(df3)
df3.reset_index(inplace=True)
df3=df3[['title',0]]
df3.columns=['title','listed_in']
```

In [479…

```
df3.head(5)
```

Out[479]:

	title	listed_in
0	Dick Johnson Is Dead	Documentaries
1	Blood & Water	International TV Shows
2	Blood & Water	TV Dramas
3	Blood & Water	TV Mysteries
4	Ganglands	Crime TV Shows

In [480…

```
# merge all the unnested dataframes
```

In [481…

```
df4=pd.merge(df_new,df1,on = 'title')
df5=pd.merge(df4,df2,on = 'title')
df6=pd.merge(df5,df3,on='title')
```

In [482…

```
df6.head(5)
```

Out[482]:

	title	cast	director	country	listed_in
0	Dick Johnson Is Dead	not available	Kirsten Johnson	United States	Documentaries
1	Blood & Water	Ama Qamata	UNKNOWN	South Africa	International TV Shows
2	Blood & Water	Ama Qamata	UNKNOWN	South Africa	TV Dramas
3	Blood & Water	Ama Qamata	UNKNOWN	South Africa	TV Mysteries
4	Blood & Water	Khosi Ngema	UNKNOWN	South Africa	International TV Shows

In [483…

```
# Left join merged data with original dataframe on 'title'.
```

In [484…

```
net =df[['show_id','type','title','date_added','release_year','rating','duration',
```

In [485…

```
net.head(5)
```

Out[485]:

	show_id	type	title	date_added	release_year	rating	duration	description	dateadd_mor
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90	As her father nears the end of his life, filmm...	
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	
2	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	
3	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	
4	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	

In [486…

```
#Lets check the Null values and type of the dataframe and number of unique element
```

In [487…

```
net.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 201763 entries, 0 to 201762
Data columns (total 14 columns):
 #   Column         Non-Null Count   Dtype         
---  ------         --------------   -----         
 0   show_id        201763 non-null  object        
 1   type           201763 non-null  object        
 2   title          201763 non-null  object        
 3   date_added     201763 non-null  datetime64[ns]
 4   release_year   201763 non-null  int64         
 5   rating         201763 non-null  object        
 6   duration       201763 non-null  int32         
 7   description    201763 non-null  object        
 8   dateadd_month  201763 non-null  int32         
 9   dateadd_year   201763 non-null  int32         
 10  cast           201763 non-null  object        
 11  director       201763 non-null  object        
 12  country        201763 non-null  object        
 13  listed_in      201763 non-null  object        
dtypes: datetime64[ns](1), int32(3), int64(1), object(9)
memory usage: 20.8+ MB
```

In [488…

```
net.nunique()
```

Out[488]:

```
show_id           8790
type                 2
title             8790
date_added        1713
release_year        74
rating              14
duration           210
description       8758
dateadd_month       12
dateadd_year        14
cast             36393
director          4992
country            128
listed_in           42
dtype: int64
```

In [489…

```
net['type']=net['type'].astype('category')
```

In [490…

```
net.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 201763 entries, 0 to 201762
Data columns (total 14 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   show_id        201763 non-null  object
 1   type           201763 non-null  category
 2   title          201763 non-null  object
 3   date_added     201763 non-null  datetime64[ns]
 4   release_year   201763 non-null  int64
 5   rating         201763 non-null  object
 6   duration       201763 non-null  int32
 7   description    201763 non-null  object
 8   dateadd_month  201763 non-null  int32
 9   dateadd_year   201763 non-null  int32
 10  cast           201763 non-null  object
 11  director       201763 non-null  object
 12  country        201763 non-null  object
 13  listed_in      201763 non-null  object
dtypes: category(1), datetime64[ns](1), int32(3), int64(1), object(8)
memory usage: 19.4+ MB
```

EXPLORATORY DATA ANALYSIS

In [491…

```
# lets extract data from our unnested dataframe 'net' and find the distribution of
```

In [492…

```
data_type = net[['type','title']].drop_duplicates(keep='last')
```

In [493…

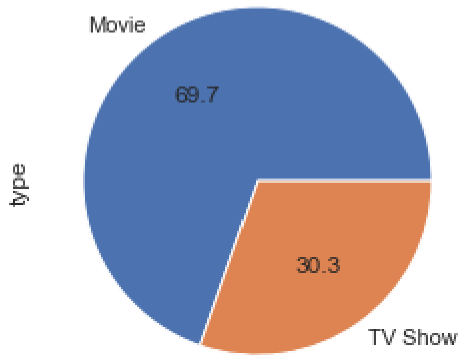
```
data_type['type'].value_counts(normalize=True)*100
```

Out[493]:

```
Movie      69.692833
TV Show    30.307167
Name: type, dtype: float64
```

In [494…

```
data_type['type'].value_counts().plot(kind='pie',autopct="%.1f")
plt.show()
```



As we can see from the pie chart above , majority of the content on netflix is movies.

Analysis by cast

In [495…

```
cast_frame = net[['title','type','cast']].drop_duplicates(keep = 'last')
```

```
In [496…  cast_frame['cast'].value_counts().sort_values(ascending=False).head(6)
```

```
Out[496]: not available       825
          Anupam Kher          43
          Shah Rukh Khan       35
          Julie Tejwani        33
          Takahiro Sakurai     32
          Naseeruddin Shah     32
          Name: cast, dtype: int64
```

Anupam kher , shah Rukh Khan, Julie Tejwani, Takahiro Sakurai, Naseeruddin Shah are the top actors with maximum no. of movies and TV shows.

```
In [497…  director_frame = net[['title','type','director']].drop_duplicates(keep = 'last')
```

```
In [498…  director_frame['director'].value_counts().sort_values(ascending=False).head(6)
```

```
Out[498]: UNKNOWN          2621
          Rajiv Chilaka      22
          Jan Suter          21
          Raúl Campos        19
          Marcus Raboy       16
          Suhas Kadav        16
          Name: director, dtype: int64
```

Raniv Chilaka and Jan Suter are the top directors with maximum number of movies and TV Show

```
In [499…  country_frame = net[['title','type','country']].drop_duplicates(keep = 'last')
```

```
In [500…  country_frame['title'].value_counts().head(5)
```

```
Out[500]: Barbecue                         12
          The Look of Silence              10
          The Professor and the Madman      8
          Shaun the Sheep                   8
          The Breadwinner                   7
          Name: title, dtype: int64
```

The movie 'Barbecue' is launched in maximum number of countries.

```
In [501…  country_frame['country'].value_counts().sort_values(ascending=False).head(6)
```

```
Out[501]: United States     3680
          India             1046
          Unknown            829
          United Kingdom     803
          Canada             445
          France             393
          Name: country, dtype: int64
```

The maximum number of TV Show and movies are launched in 'United States'.

```
In [502…  genre_frame = net[['title','type','listed_in']].drop_duplicates(keep = 'last')
```

```
In [503…  genre_frame['listed_in'].value_counts().sort_values(ascending=False).head(6)
```

Out[503]:

```
International Movies      2752
Dramas                    2426
Comedies                  1674
International TV Shows    1349
Documentaries              869
Action & Adventure         859
Name: listed_in, dtype: int64
```

Dramas are the most available genre on Netflix and the genre which is present the most on Netflix is 'dramas'.

In [504…

```
genre_frame['listed_in'].value_counts().sort_values(ascending=False).tail(6)
```

Out[504]:

```
Faith & Spirituality            65
TV Thrillers                    57
Stand-Up Comedy & Talk Shows    56
Movies                          53
Classic & Cult TV               26
TV Shows                        16
Name: listed_in, dtype: int64
```

Faith & Spirituality ,TV Thrillers and stand-Up Comedy & Talk Shows are the least no. of content available on Netflix.

In [505…

```
duration_frame = net[['title','type','duration']].drop_duplicates(keep = 'last')
```

In [506…

```
duration_frame[duration_frame['type']=='TV Show'].groupby('duration')['title'].cou
```

Out[506]:

```
duration
1     1791
2      421
3      198
4       94
5       64
6       33
7       23
8       17
9        9
10       6
Name: title, dtype: int64
```

As we can see most of the TV Show have 1 or 2 seasons .

In [507…

```
tv_shows=tv_shows=duration_frame[duration_frame['type']=='TV Show']
movies=duration_frame[duration_frame['type']=='Movie']
```

In [508…

```
movies['duration'].value_counts().head(20)
```

```
Out[508]: 90     152
          94     146
          93     146
          97     146
          91     144
          95     137
          96     130
          92     129
          102    122
          98     120
          99     118
          101    116
          88     116
          103    114
          106    111
          100    108
          89     106
          104    104
          86     103
          105    101
          Name: duration, dtype: int64
```

Most of the movies are in between 90-100 minutes of duration

```
In [509…  duration_frame.groupby('type')['duration'].mean()
```

```
Out[509]: type
          Movie      99.584884
          TV Show     1.751877
          Name: duration, dtype: float64
```

The average running type of movies is 99.5 minutes and avg number of seasons of a TV Show is 1.75 seasons.

```
In [510…  tv=df[df['type']=='TV Show']
          movie=df[df['type']=='Movie']
```

```
In [511…  tv['dateadd_year'].value_counts()
```

```
Out[511]: 2020    595
          2019    592
          2021    505
          2018    411
          2017    349
          2016    175
          2015     26
          2014      5
          2013      5
          2008      1
          Name: dateadd_year, dtype: int64
```

```
In [512…  tv['dateadd_month'].value_counts()
```

Out[512]:

```
12    265
7     262
9     251
8     236
6     236
10    215
4     214
3     213
11    207
5     193
1     192
2     180
Name: dateadd_month, dtype: int64
```

A large number of TV Shows are launched in the recent years and in the month from july to september i.e. summer holidays and also maximum no. of TV shows are launched in December i.e . during christmas holidays.

In [513…

```
x = net[['type','cast','title']].drop_duplicates(keep = 'last')
x[x['type']=='TV Show']['cast'].value_counts().head(5)
```

Out[513]:

```
not available       350
Takahiro Sakurai     25
Yuki Kaji            19
Junichi Suwabe       17
Daisuke Ono          17
Name: cast, dtype: int64
```

Takahiro Sakurai appeared in most number of TV Shows .

In [514…

```
x[x['type']=='Movie']['cast'].value_counts().head(5)
```

Out[514]:

```
not available       475
Anupam Kher          42
Shah Rukh Khan       35
Naseeruddin Shah     32
Akshay Kumar         30
Name: cast, dtype: int64
```

Anupam Kher appeared in most number of movies

In [515…

```
j=net[['type','director','title']]
j.drop_duplicates(keep='last')
j[j['type']=='TV Show']['director'].value_counts()
```

Out[515]:

```
UNKNOWN             49142
Noam Murro            189
Thomas Astruc         160
Houda Benyamina       104
Damien Chazelle       104
                    ...  
Rashida Jones           1
Sharon Grimberg         1
Garrett Bradley         1
Alex Gibney             1
Padraic McKinley        1
Name: director, Length: 300, dtype: int64
```

Noam Murro has directed the maximum number of TV Show.

In [516…

```
j[j['type']=='Movie']['director'].value_counts()
```

Out[516]:

```
UNKNOWN                1283
Martin Scorsese         419
Youssef Chahine         409
Cathy Garcia-Molina     356
Steven Spielberg        355
                       ... 
Mark Zwonitzer            1
Rudge Campos              1
David Salzberg            1
Christian Tureaud         1
Kirsten Johnson           1
Name: director, Length: 4776, dtype: int64
```

Martin Scorsese has directed the most number of movies.

In [517…

```
# find no. of ratings
```

In [518…

```
h= net[['title','country','rating']]
h.drop_duplicates(keep='last')
h['rating'].value_counts()
```

Out[518]:

```
TV-MA       73835
TV-14       43859
R           25860
PG-13       16246
TV-PG       14913
PG          10919
TV-Y7        6294
TV-Y         3664
TV-G         2779
NR           1543
G            1530
NC-17         149
TV-Y7-FV       86
UR             86
Name: rating, dtype: int64
```

Most of the content available on netflix is for mature audience and adult content for people above the age of 14

In [519…

```
h.groupby('rating')['country'].value_counts()
```

Out[519]:

```
rating    country       
G         United States     907
          United Kingdom    130
          Spain              74
          Ireland            65
          Germany            56
                           ... 
TV-Y7-FV  Denmark             2
          Unknown             2
UR        France             45
          United Kingdom     21
          United States      20
Name: country, Length: 526, dtype: int64
```

In [538…

```
# Type of content available in United States
```

In [537…

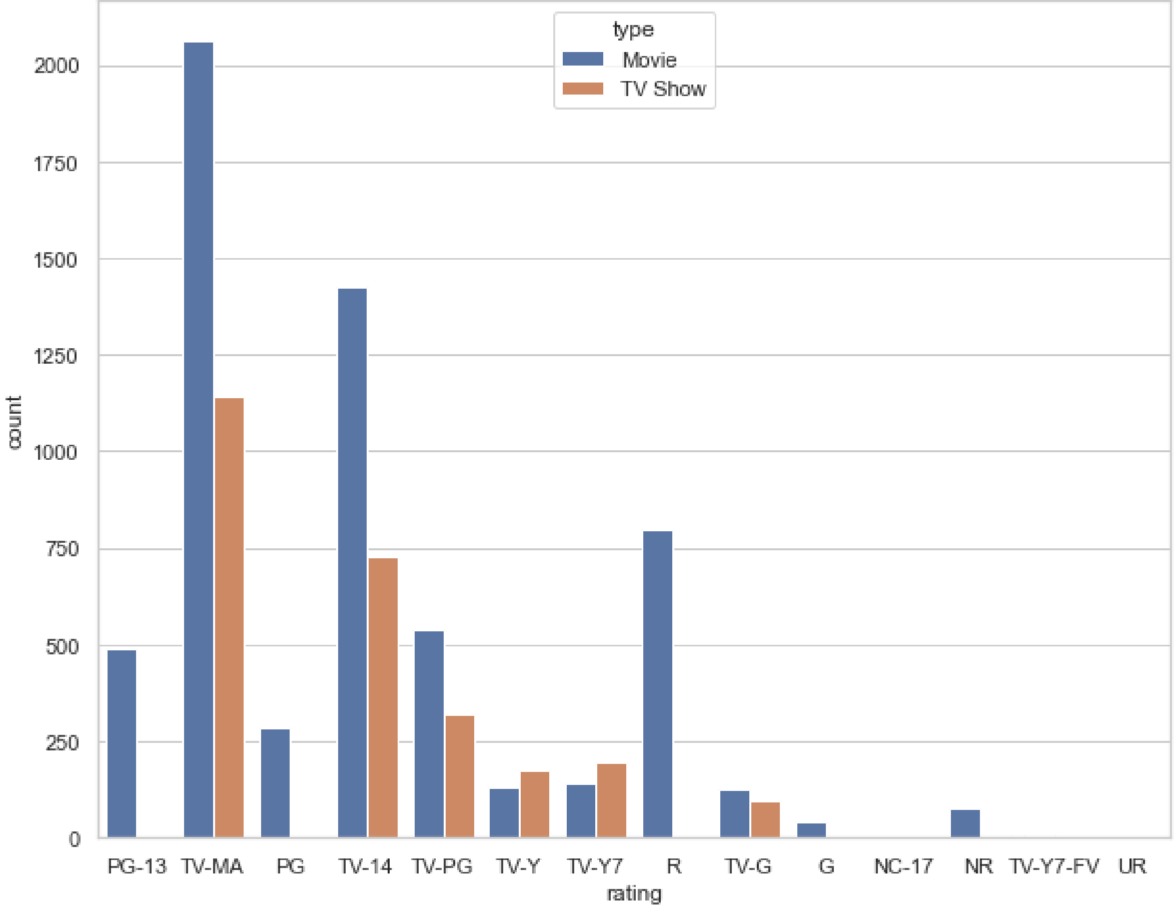
```
l = net[['country','rating','title']].drop_duplicates(keep='last')
u= l[l['country']=='United States']
u['rating'].value_counts()
```

Out[537]:

```
TV-MA       1099
R            660
TV-14        495
PG-13        433
TV-PG        302
PG           243
TV-Y7        147
TV-Y         127
TV-G          89
NR            42
G             39
TV-Y7-FV       2
NC-17          1
UR             1
Name: rating, dtype: int64
```

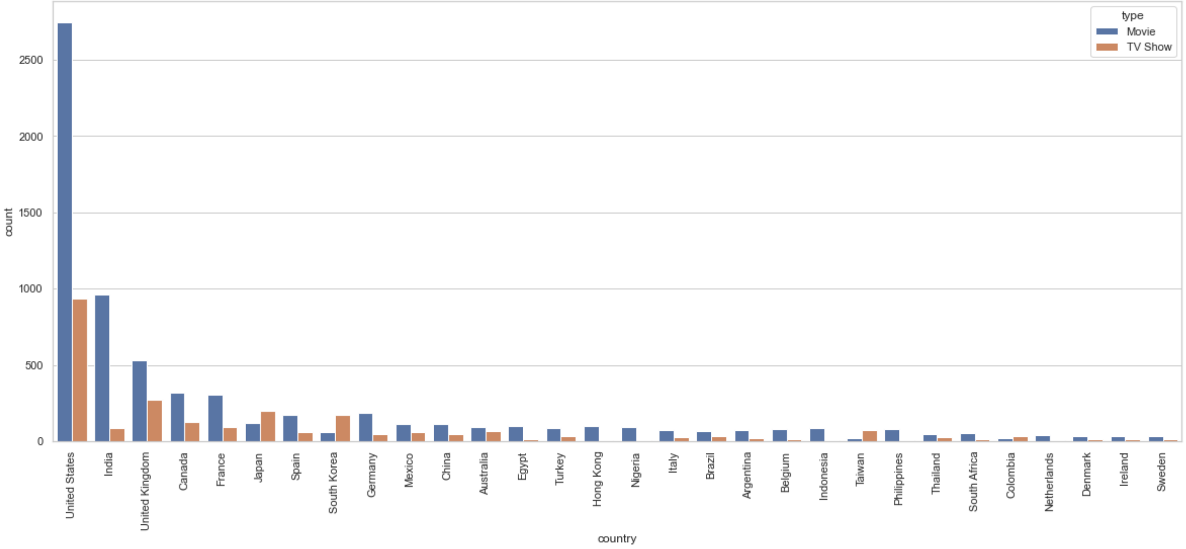
In [520…

```
q=net[['title','rating','type']].drop_duplicates(keep='last')
plt.figure(figsize=(10,8))
sns.countplot(x='rating', hue='type', data=q)
plt.show()
```



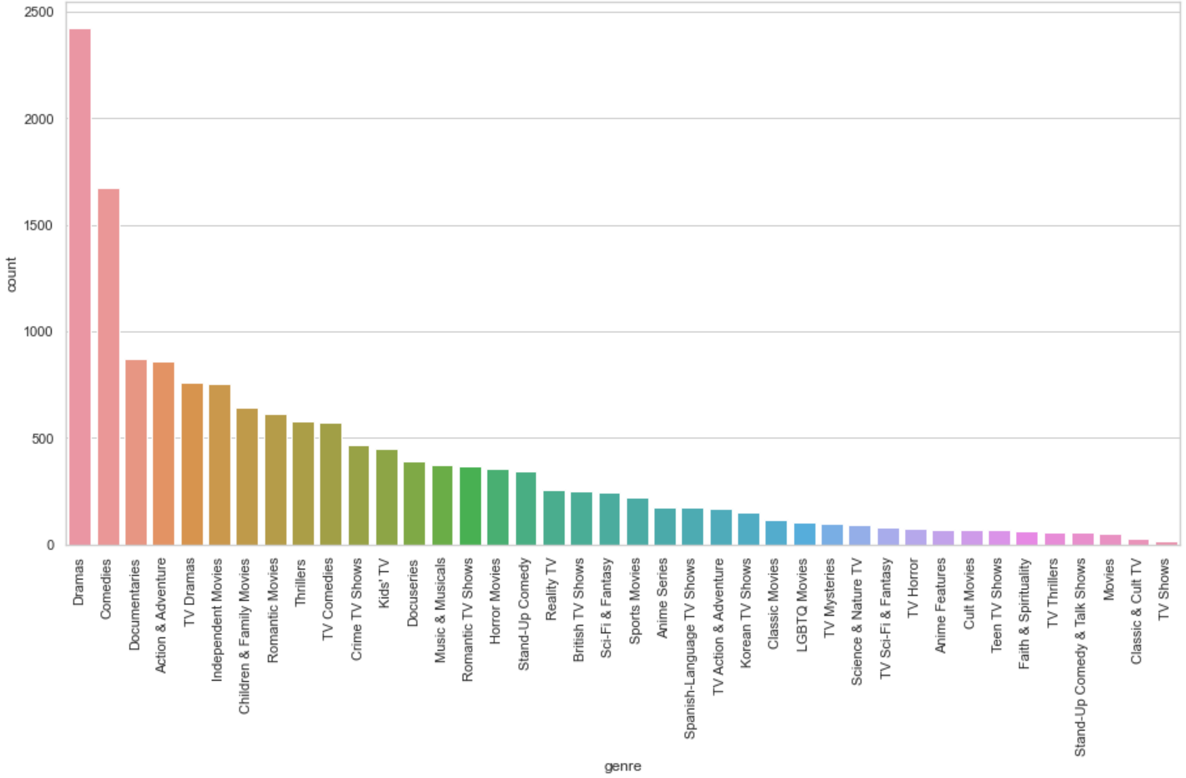
In [521…

```
r=net[['title','country','type']].drop_duplicates(keep='last')
m =r[r['country']!='Unknown']
plt.figure(figsize=(20,8))
sns.countplot(x='country', hue='type', data=m,order = m['country'].value_counts().
plt.xticks(rotation=90)
plt.show()
```



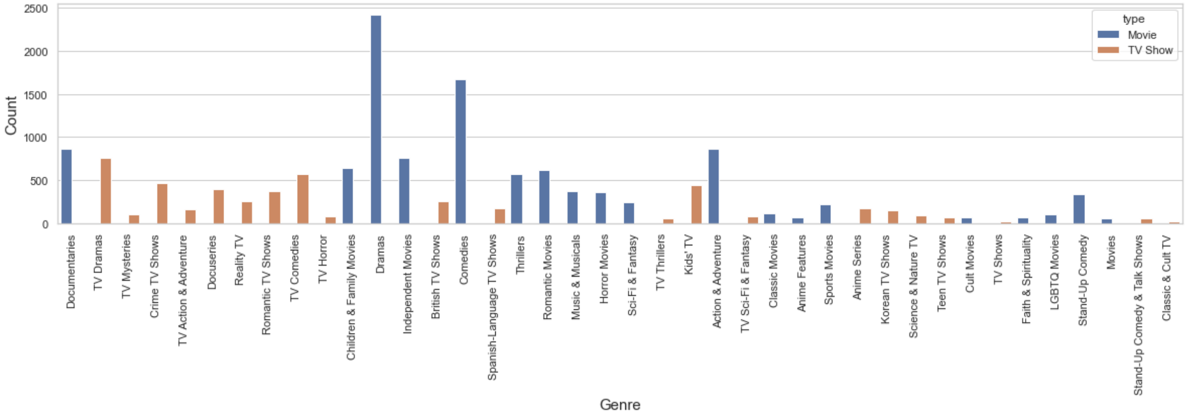
from the above graph , it is visible that countries Japan , South Korea ,Taiwan and columbia has higher proportion of Tv Show than movies .

```
In [522…
df0 = df3[(df3['listed_in']!='International Movies')&(df3['listed_in']!='Internati
plt.figure(figsize = (16,8))
sns.countplot(x = 'listed_in', data = df0, order = df0['listed_in'].value_counts()
sns.set(style="whitegrid")
plt.xlabel('genre')
plt.xticks(rotation=90)
plt.show()
```



Genre availability on Netflix.

```
In [523…
t= net[['type','listed_in','title']].drop_duplicates(keep="last")
d= t[(t['listed_in']!='International Movies')&(t['listed_in']!='International TV Sl
plt.figure(figsize = (20,4))
sns.countplot(x='listed_in', hue='type', data= d)
plt.xlabel('Genre',fontsize=15)
plt.ylabel('Count',fontsize=15)
plt.xticks(rotation=90)
plt.show()
```



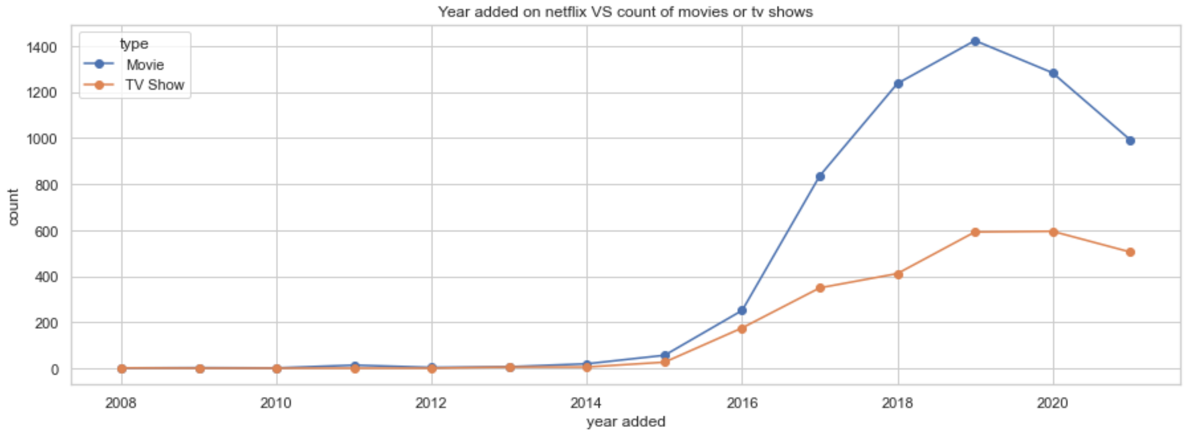
Both in TV shows and movies drama and comedy is the most available content on netflix.

In [524…

```
#  line chart to  find a relation no. of between movies or Tv show relesed VS year
```

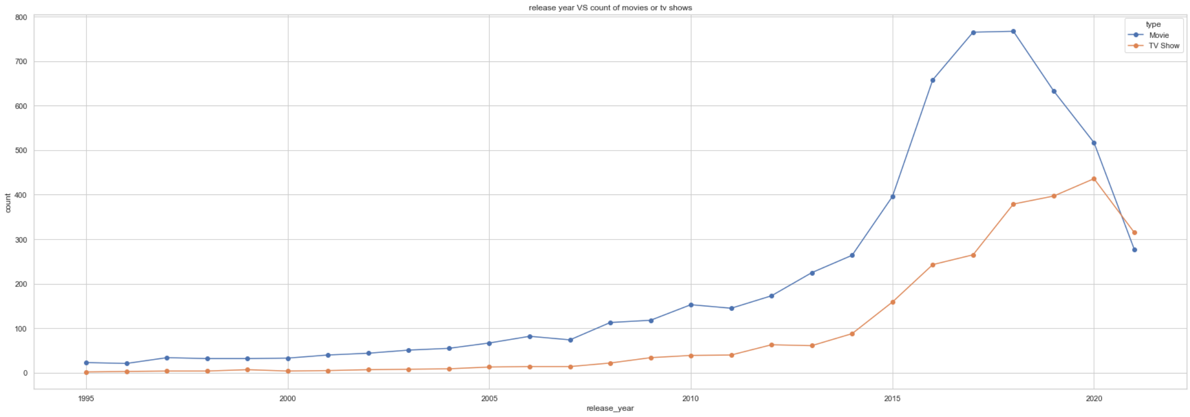
In [525…

```
a= net[['dateadd_year','title','type']].drop_duplicates(keep='last')
counts = a.groupby(['dateadd_year', 'type']).size().unstack()
counts.plot(kind = 'line', marker = 'o', figsize = (15, 5))
plt.xlabel('year added')
plt.ylabel('count')
plt.title('Year added on netflix VS count of movies or tv shows')
plt.show()
```



In [526…

```
ad= net[['release_year','title','type']].drop_duplicates(keep='last')
am=ad[ad['release_year']>=1995]
cnts = am.groupby(['release_year', 'type']).size().unstack()
cnts.plot(kind = 'line', marker = 'o', figsize = (30, 10))
plt.xlabel('release_year')
plt.ylabel('count')
plt.title('release year VS count of movies or tv shows')
plt.show()
```



Comment : The decline in no. of movies released is very sharp as compared to TV shows after 2020 , it becomes lower than TV shows on a cetain year after 2020

In [527…

```
h= net[['title','description']].drop_duplicates(keep='last')
text = ' '.join(description for description in h.description.dropna())
tct = text.lower()
tmk = tct.split()
g = pd.DataFrame(tmk)
g[0].value_counts().head(50)
```

Out[527]:

```
a                11592
the               8095
to                6432
and               6305
of                5260
in                4327
his               3341
with              2257
her               2076
an                1992
for               1781
on                1756
their             1667
when              1512
this              1389
from              1290
as                1222
is                1108
by                1004
after              992
he                 871
that               820
who                805
but                804
at                 738
young              717
into               712
new                693
–                  606
life               577
up                 573
they               539
two                495
she                473
family             454
man                446
out                418
woman              415
must               397
are                382
while              376
world              371
love               371
friends            366
about              352
him                345
find               335
one                328
documentary        313
finds              312
Name: 0, dtype: int64
```

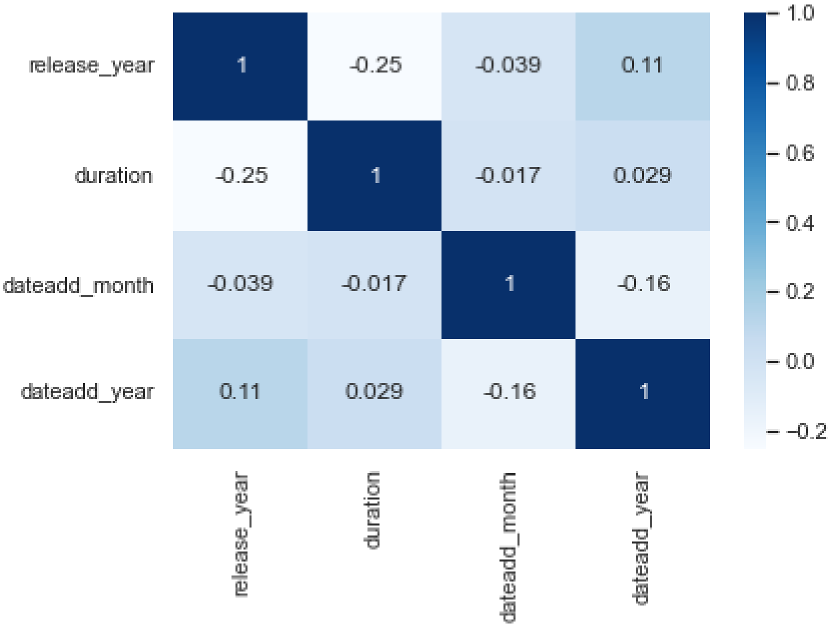
Most of the movies and TV shows released on Netflix contains words like young , love , friends , find , family i.e. positive content .

In [528…

```
sns.heatmap(df.corr(), cmap='Blues', annot=True)
```

Out[528]:

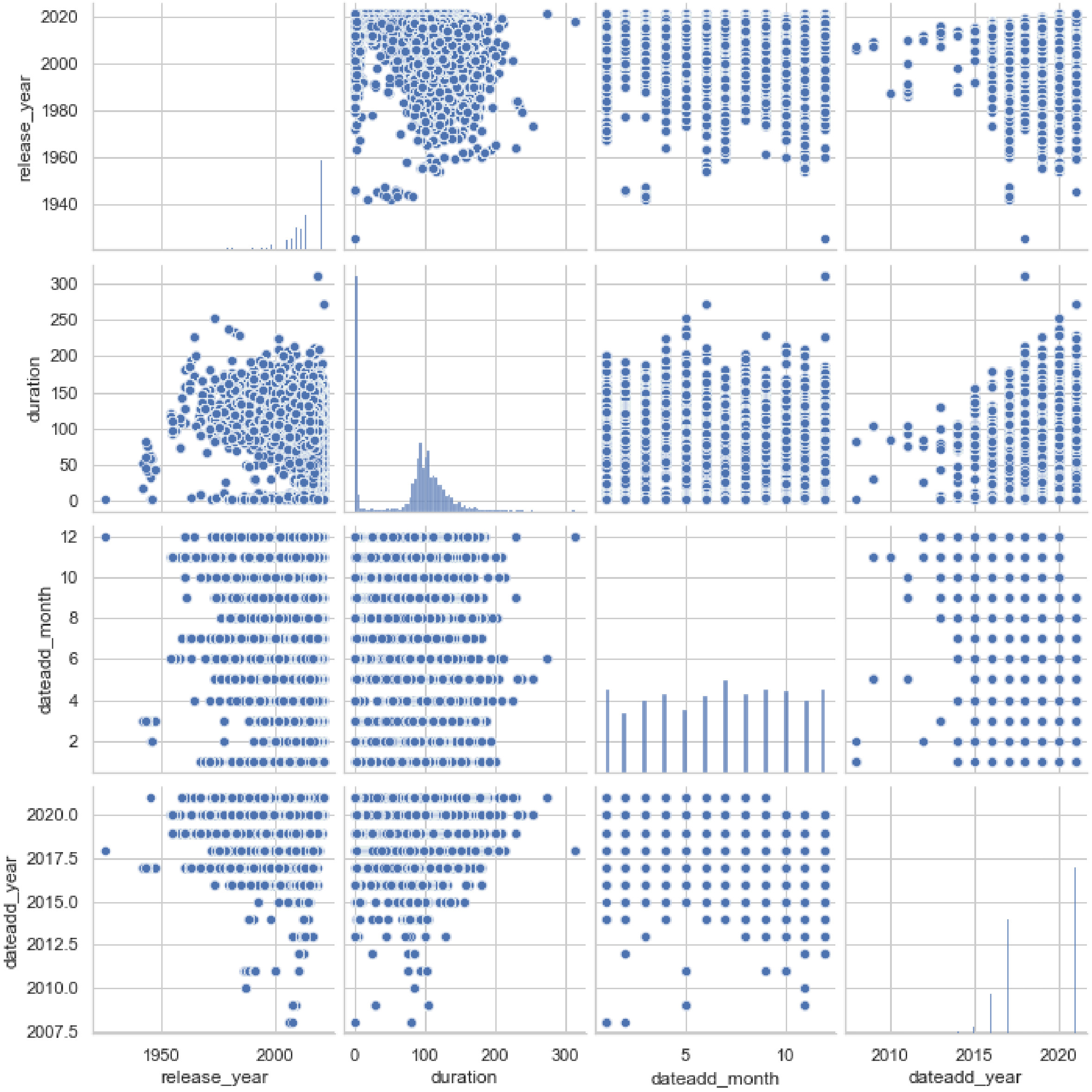
```
<AxesSubplot:>
```



As we can see the correlation between release year , netflix add year , netflix add month and duration is weak.

In [529…
```
sns.pairplot(net)
```

Out[529]:
```
<seaborn.axisgrid.PairGrid at 0x17b58618160>
```

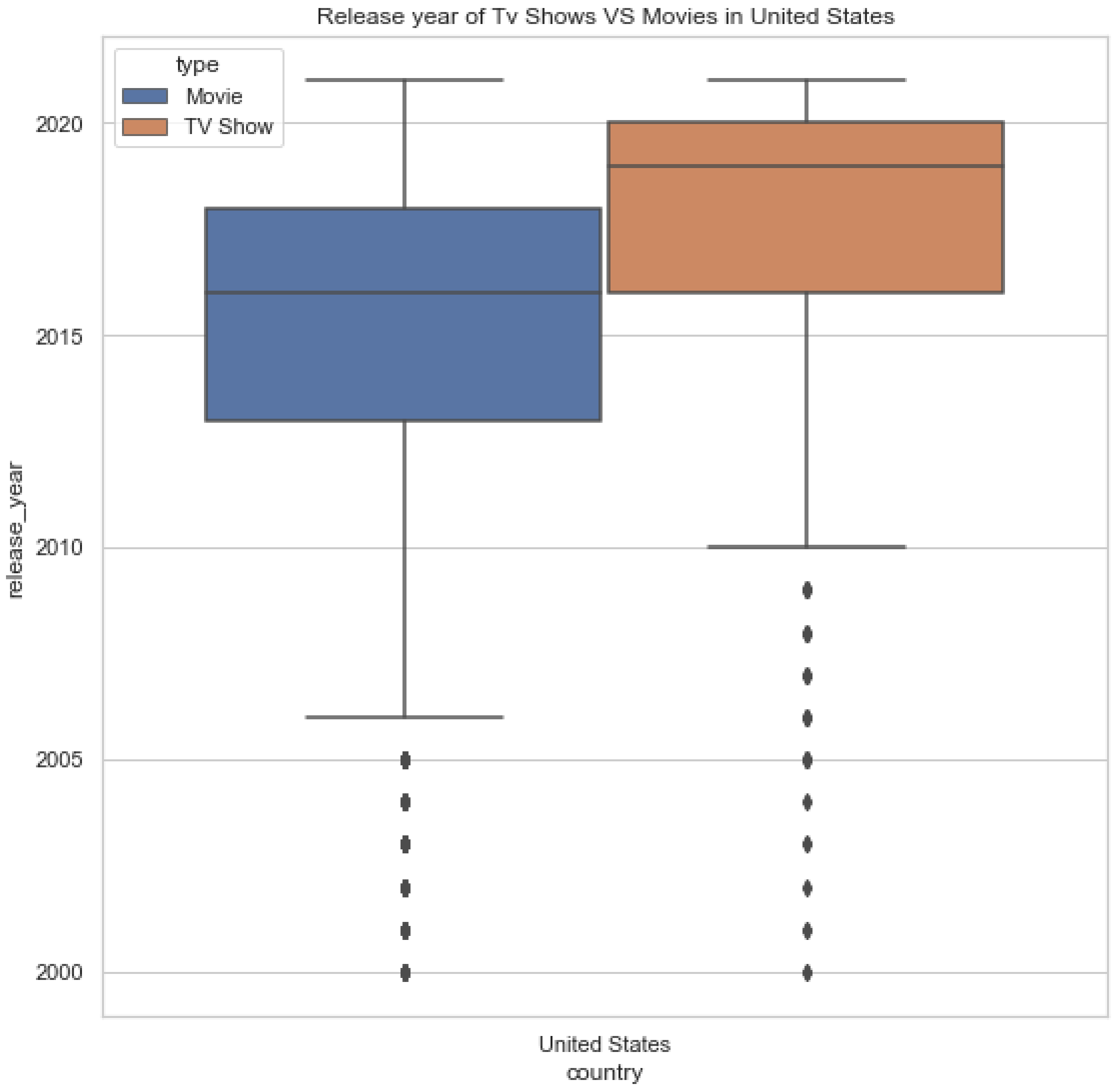


In [530…

```
# boxplot to compare the release dates of movies and TV shows
```

In [531…

```
y =net[['country','title','type','release_year']].drop_duplicates(keep = 'last')
z= y[(y['release_year']>=2000)&(y['country']=='United States')]
plt.figure(figsize = (9,9))
sns.boxplot(x='country', y='release_year', data=z, hue='type')
plt.title('Release year of Tv Shows VS Movies in United States')
plt.show()
```



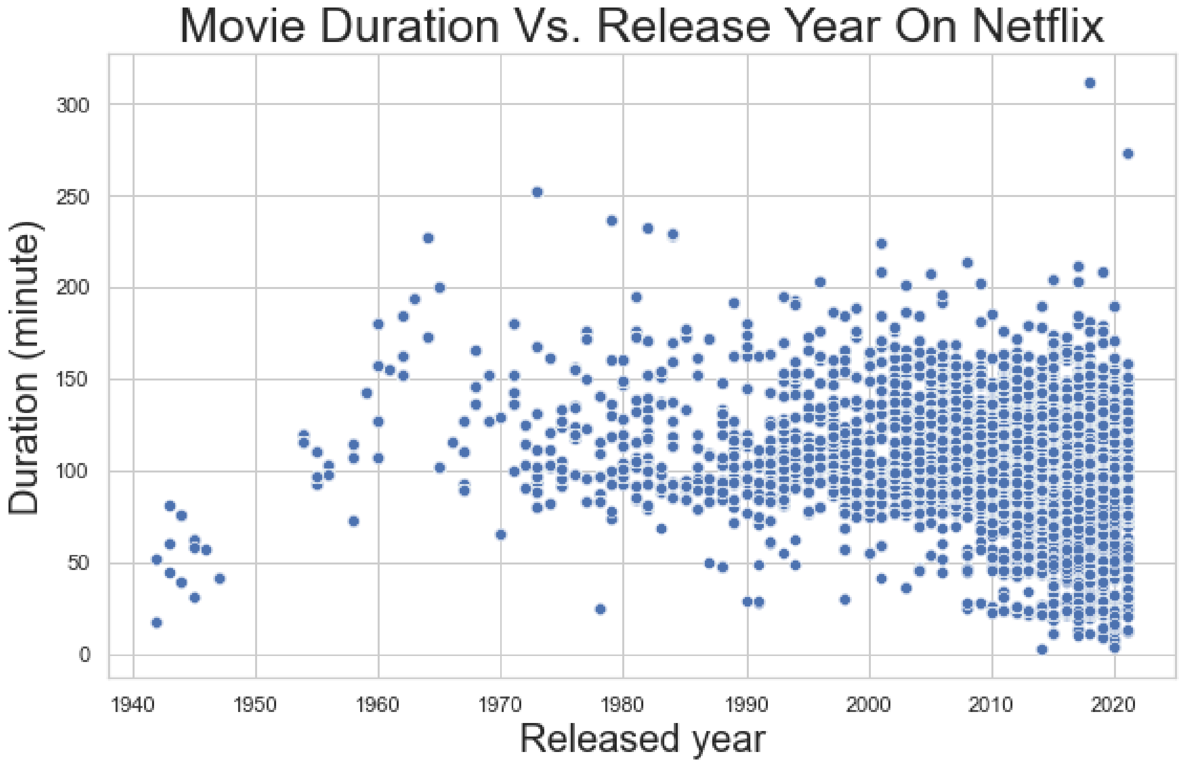
Comments : The movies have a wide range of release year but the median of TV shows is much higher than that of Movies showing that more TV shows are being released than movies in recent years in U.S.

In [532…

```
#TV Show and Movie duration VS release date : scatter plot
```

In [533…

```
p = net[['release_year','duration','type']]
t = p[p['type']=='Movie']
plt.figure(figsize = (10,6))
sns.scatterplot(x = "release_year", y = "duration",data = t)
plt.title("Movie Duration Vs. Release Year On Netflix", fontsize = 25)
plt.xlabel("Released year", fontsize = 20)
plt.ylabel("Duration (minute)", fontsize = 20)
plt.show()
```

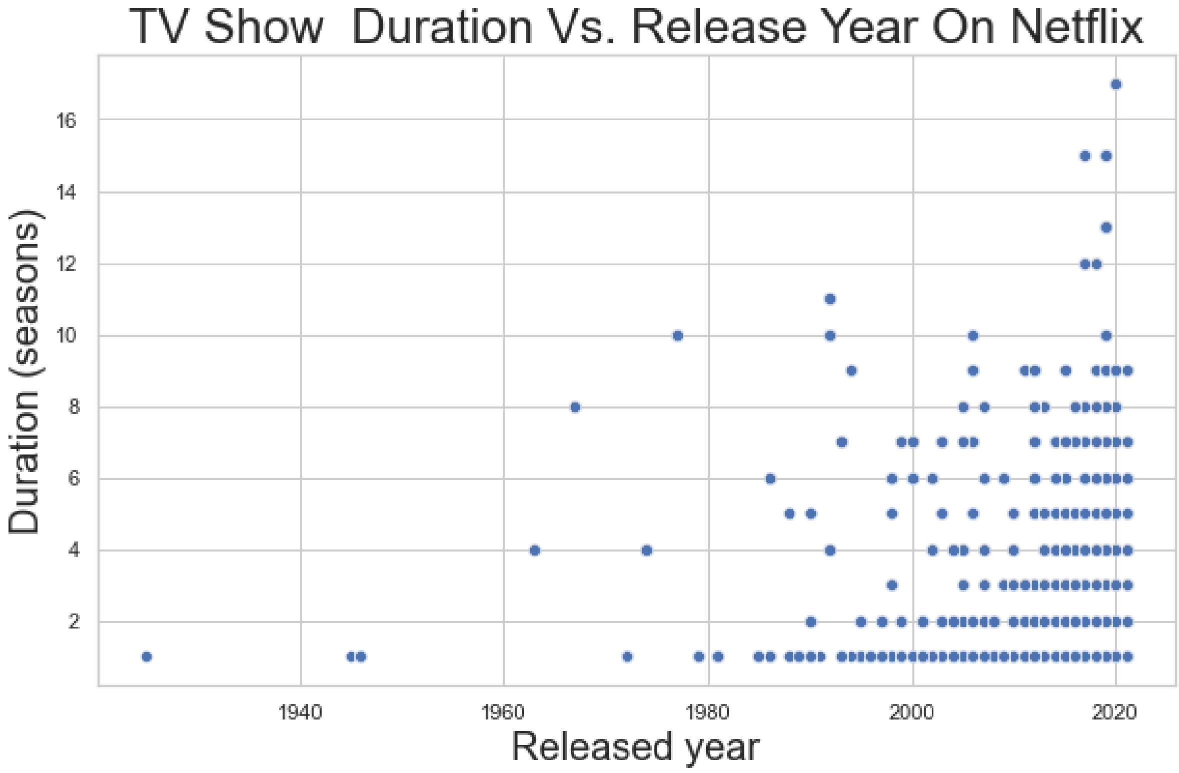


In [534…

```
# Comment :  From 2000 to 2020 , the no. of movies with duration less than 50 minu
```

In [535…

```
p = net[['release_year','duration','type']]
t = p[p['type']=='TV Show']
plt.figure(figsize = (10,6))
sns.scatterplot(x = "release_year", y = "duration",data = t)
plt.title("TV Show  Duration Vs. Release Year On Netflix", fontsize = 25)
plt.xlabel("Released year", fontsize = 20)
plt.ylabel("Duration (seasons)", fontsize = 20)
plt.show()
```



After the year 2000 , the number of seasons of TV shows released started to increase from 1 to 8 .

Business Insights

```
1. 70% of the content is movies and 30% is TV Shows.
2. From the bar graph analysis between type and countries , it is
inferred that the south Asian countries like Japan ,Taiwan ,South
Korea  has more number of TV Shows as compared to the movies ,and
the number of movies in countries like United States , India, UK
,Canada ,France,Spain  have more number of movies than TV Shows .
3. From the line chart between year added and number of TV Shows
or movies released  it can be seen that the number of movies added
on netflix increased till 2019 , after that it started declining
sharply, whereas the number of  TV shows added increased till 2020
and then declines.The decline in the no. of movies added is very
sharp as compared to the TV Shows.
4. From the line chart between release_year and number of TV Shows
or movies released , it can be seen that the number of movies
released on netflix increased till 2018 , after that it started
declining sharply, whereas the number of  TV shows released
increased till 2020  and then declines.The decline in the no. of
movies released is very sharp as compared to the TV Shows and at a
certain year the number of movies released is less than the number
of TV Show.
5. A large number of TV Shows are launched in the recent years
and in the month from july to september i.e. summer holidays and
also maximum no. of TV shows are launched in December i.e . during
christmas holidays.
6. Anupam kher , shah Rukh Khan, Julie Tejwani, Takahiro Sakurai,
Naseeruddin Shah  are the top  5 actors with maximum no. of movies
and TV shows.
7. Raniv Chilaka is the top director with maximum number of movies
and TV Show.
8. The movie 'Barbecue' is launched in maximum number of
countries.
9. The maximum number of TV Show and movies are launched in
'United States'.
10. 'Dramas' are the most available genre on Netflix and the genre
which is present the most on Netflix .
11. Faith & Spirituality ,TV Thrillers and stand-Up Comedy & Talk
Shows are  the least no. of genre available on Netflix.
12. The average running type of movies is 99.5 minutes and avg
number of seasons of a TV Show is 1.75 seasons.
13. Takahiro Sakurai appeared in most number of TV Shows .
14. Anupam Kher appeared in most number of movies.
15. Noam Murro has directed the maximum number of TV Shows.
16. Martin Scorsese has directed the most number of movies.
17. Most of the content available on netflix is for mature
audience and adult content for people above the age of 14.
18. Most of the movies and TV shows released on Netflix contains
words like young , love , friends , find , family i.e. positive
content
19. From 2000 to 2020 , the no. of movies with duration less than
50 minutes increased.
```

```
20. After the year 2000 , the number of seasons of TV shows
released started to increase from 1 to 8
```

RECOMMENDATIONS:

1. As the percentage of TV Shows on netflix is higher than movies ,so netflix should add more movies ,specifically in countries like japan , south Korea and Taiwan.
2. As the number of TV Shows being released has surpassed the movies released in the past year but the number of movies added is still higher than the TV Shows added so netflix should increase the rate at which it is adding the TV Shows.
3. Recently the number of movies with duration around 50 min are being released more and the number of seasons of TV series is also increasing so netflix should add movies with less duration and add TV Shows with no. of seasons between 1-8.
4. As netflix has a lot of positive content so it should add more negative content .
5. Netflix should add content under the UR,Tv-G ,G ,NR,NC-17 ratings , they should add more kids content.
6. They should add more content related to the genre - Faith & Spirituality ,TV Thrillers and stand-Up Comedy & Talk Shows.
7. They should add shows and movies with famous casts like Takahiro Sakurai, Anupam Kher and shah rukh khan.

In [ ]: