Business Case - Netflix - Data Exploration and Visualisation

Problem Statement -

Analyze the data and generate insights that could help Netflix in deciding which type of shows/movies to produce and how they can grow the business in different countries.

Importing Python Libraries necessary while carrying out data exploration & visualisation -

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

Upload & read csv file in pandas dataframe -

```
netflix = pd.read_csv("netflix.csv", sep = ",", encoding = "ISO-8859-1")
```

Inspecting Dataset & Analyzing Different Metrics -

```
netflix.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas,	NaN	September 24, 2021	2021	TV-MA	1 Season

```
netflix.tail()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	durati
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 r
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	Seasc
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 r
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 r
					Vicky					

Observations on -

1) shape of data

2) data types

3) Statistical summary

```
netflix.shape
```

```
(8807, 12)
```

```
netflix.columns
```

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

```
netflix.size
```

```
105684
```

```
netflix.dtypes
```

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

```
netflix.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
```

```
netflix.describe()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-f4e7d0b9f151> in <cell line: 1>()
----> 1 netflix.describe().round()

NameError: name 'netflix' is not defined
```

SEARCH STACK OVERFLOW

```
netflix.describe(include = object)
```

	show_id	type	title	director	cast	country	date_added	rating	duration	listed_i
count	8807	8807	8807	6173	7982	7976	8797	8803	8804	880
unique	8807	2	8807	4528	7692	748	1767	17	220	51
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	TV-MA	1 Season	Drama Internation Movie

```
# convert the data type from object to datetime64
netflix["date_added"] = pd.to_datetime(netflix["date_added"])
```

Data Cleaning (Optional Treatment) -

```
Check for Missing values & Duplicates.
```

```
# Null counts
netflix.isnull().sum().sort_values(ascending = False)
```

```
director        2634
country          831
cast             825
date_added        10
rating             4
duration           3
show_id            0
type               0
title              0
release_year       0
listed_in          0
description        0
dtype: int64
```

```
# Null values percentage
round(100 * (netflix.isnull().sum() / len(netflix.index)),2).sort_values(ascending = False)
```

```
director        29.91
country          9.44
cast             9.37
date_added       0.11
rating           0.05
duration         0.03
show_id          0.00
type             0.00
title            0.00
release_year     0.00
listed_in        0.00
description      0.00
dtype: float64
```

```
# Drop low percentage null values
netflix = netflix[~pd.isnull(netflix["rating"])]
```

```
netflix = netflix[~pd.isnull(netflix["duration"])]
```

```
netflix = netflix[~pd.isnull(netflix["date_added"])]
```

```
# Replace the null values for country, cast & director
netflix["country"].replace(np.NaN, "No Country", inplace = True)
```

```
netflix["cast"].replace(np.NaN, "No Cast", inplace = True)
```

```
netflix["director"].replace(np.NaN, "No Director", inplace = True)
```

```
# Check for Null counts again
netflix.isnull().sum().sort_values(ascending = False)
```

```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

Non Graphical Analysis -

```
netflix.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No Cast	United States	2021-09-25	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	No Director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	No Country	2021-09-24	2021	TV-MA	1 Season
		TV	Jailbirds	No		No				

```
# Unnesting of director columns & Fetching top 5 Directors -

filtered_directors = pd.DataFrame()

filtered_directors = netflix['director'].str.split(',',expand=True).stack()

filtered_directors = filtered_directors.to_frame()

filtered_directors.columns = ['Director']

directors = filtered_directors.groupby(['Director']).size().reset_index(name = 'Total Content')

directors = directors[directors.Director != 'No Director']

directors = directors.sort_values(['Total Content'],ascending = False)


directors.head(5)
```

	Director	Total Content
4019	Rajiv Chilaka	22
261	Jan Suter	18
4066	RaÃºl Campos	18
4650	Suhas Kadav	16
3233	Marcus Raboy	16

```
# Unnesting of cast columns & Fetching top 5 actors -

filtered_cast = pd.DataFrame()

filtered_cast = netflix['cast'].str.split(',',expand=True).stack()

filtered_cast = filtered_cast.to_frame()

filtered_cast.columns = ['Actor']

actors = filtered_cast.groupby(['Actor']).size().reset_index(name = 'Total Content')

actors = actors[actors.Actor != 'No Cast']

actors = actors.sort_values(['Total Content'],ascending=False)


actors.head(5)
```

	Actor	Total Content
2605	Anupam Kher	39
26903	Rupa Bhimani	31
30263	Takahiro Sakurai	30
15518	Julie Tejwani	28
23591	Om Puri	27

```
# Unnesting of country columns & Fetching top 5 actors -

filtered_country = pd.DataFrame()

filtered_country = netflix['country'].str.split(',',expand=True).stack()

filtered_country = filtered_country.to_frame()

filtered_country.columns = ['Countries']

countries = filtered_country.groupby(['Countries']).size().reset_index(name = 'Total Content')

countries = countries[countries.Countries != 'No Country']

countries = countries.sort_values(['Total Content'],ascending=False)


countries.head(5)
```

	Countries	Total Content
192	United States	3202
141	India	1008
191	United Kingdom	627
106	United States	479
122	Canada	271

```
# Movies & TV_Shows -
netflix["type"].value_counts()
```

```
Movie      6126
TV Show    2664
Name: type, dtype: int64
```

```
# Ratings -
netflix["rating"].head(5)
```

```
0    PG-13
1    TV-MA
2    TV-MA
3    TV-MA
4    TV-MA
Name: rating, dtype: object
```

```
# Year wise count -
netflix["release_year"].value_counts().reset_index().head(10)
```

	index	release_year
0	2018	1146
1	2017	1030
2	2019	1030
3	2020	953
4	2016	901
5	2021	592
6	2015	555
7	2014	352

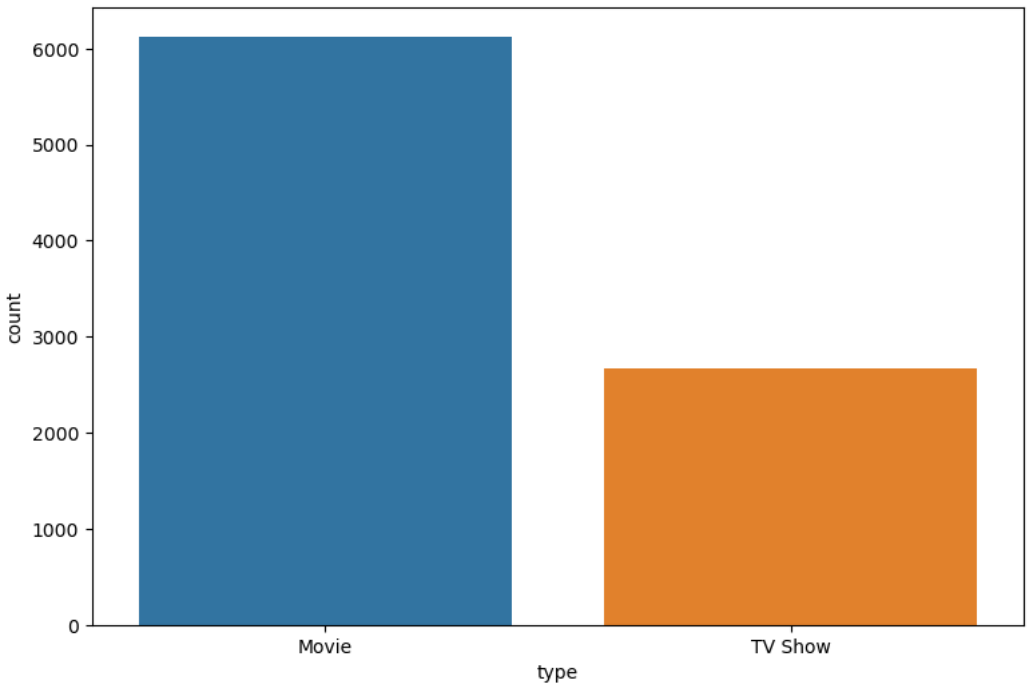
9	2012	236

```
# Listed_in (Genres) -
netflix["listed_in"].value_counts().head(10)
```

```
Dramas, International Movies                        362
Documentaries                                       359
Stand-Up Comedy                                     334
Comedies, Dramas, International Movies              274
Dramas, Independent Movies, International Movies    252
Kids' TV                                            219
Children & Family Movies                            215
Children & Family Movies, Comedies                  201
Documentaries, International Movies                 186
Dramas, International Movies, Romantic Movies       180
Name: listed_in, dtype: int64
```

Visual Analysis -

```
# Count plots - Movies & TV Shows
plt.figure(figsize= (9, 6))
sns.countplot(x = 'type', data = netflix)
plt.show()
```



Insights -

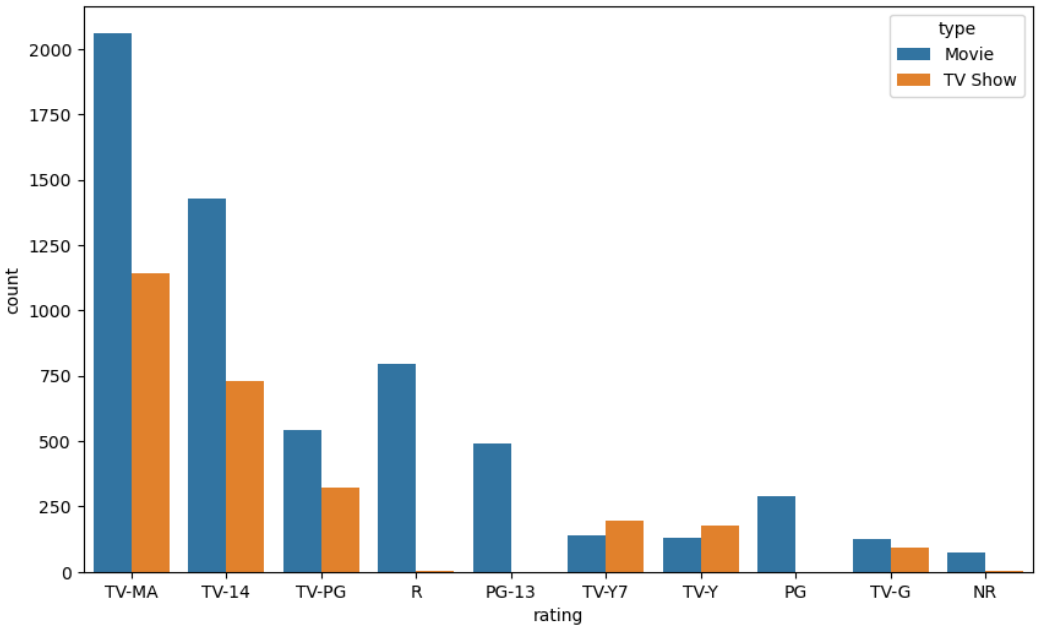
1) Netflix offers two primary categories of content: movies and TV shows.

2) Netflix has a greater quantity of movies compared to TV shows in its library.

3) The total number of distinct entertainment titles available on Netflix is 8807.

4) Among these titles, 6131 are movies, while the remaining are TV shows.

```
# Ratings -
plt.figure(figsize= (10, 6))
sns.countplot(x = "rating", order = netflix["rating"].value_counts().index[:10] ,data = netflix, hue = "type")
plt.show()
```



Insights -

1) Netflix utilizes a total of 14 different ratings for both movies and TV shows.

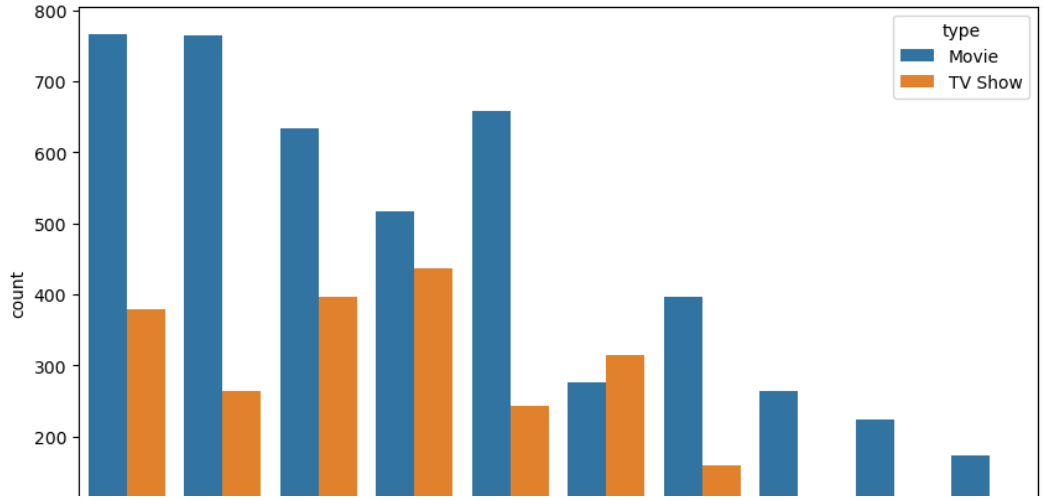
2) Out of these 14 ratings, six are specifically assigned to TV shows: TV-14, TV-G, TV-MA, TV-PG, TV-Y, TV-Y7.

3) More than 2000 movies on Netflix have been given the TV-MA rating.

4) Similarly, over 1100 TV shows on Netflix have received the TV-MA rating exclusively.

5) A small number of movies are categorized under ratings such as G, NC-17, TV-Y7, TV-Y7-FV, and UR.

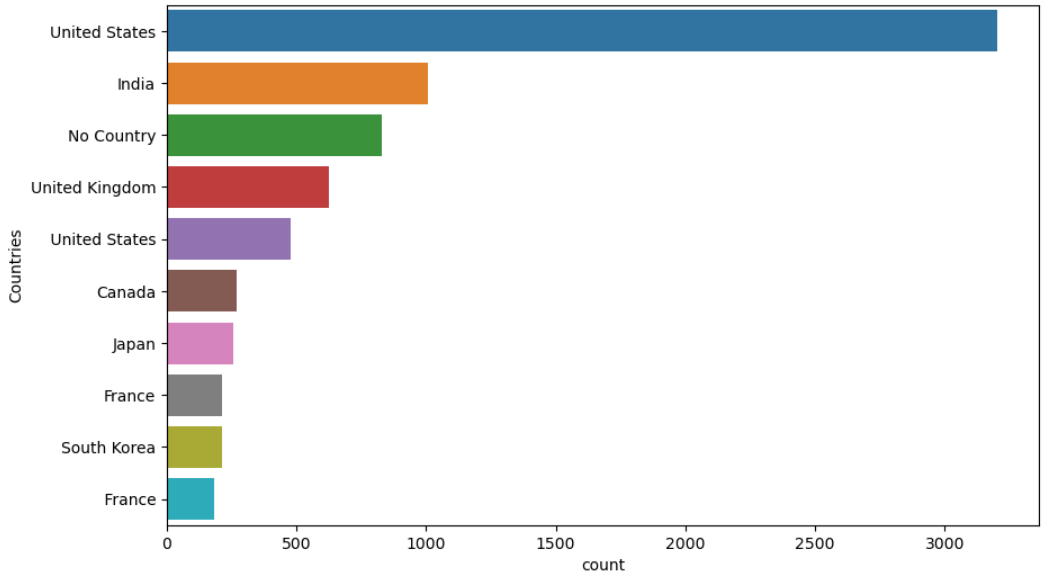
```
# Year wise -
plt.figure(figsize= (10, 6))
sns.countplot(x = "release_year", order = netflix["release_year"].value_counts().index[:10] ,data = netflix, hue = "type")
plt.show()
```



Insights -

1) The highest number of TV shows were added to Netflix's collection after 2013.
2) The initial inclusion of movies in Netflix was relatively slow, but it accelerated significantly after 2014.
3) The majority of movies available on Netflix were released between 2010 and 2020.
4) The years 2017 and 2018 witnessed the highest number of movie releases on Netflix.

```
# country wise -
plt.figure(figsize= (10, 6))
sns.countplot(y = "Countries", order = filtered_country["Countries"].value_counts().index[:10], data = filtered_country)
plt.show()
```

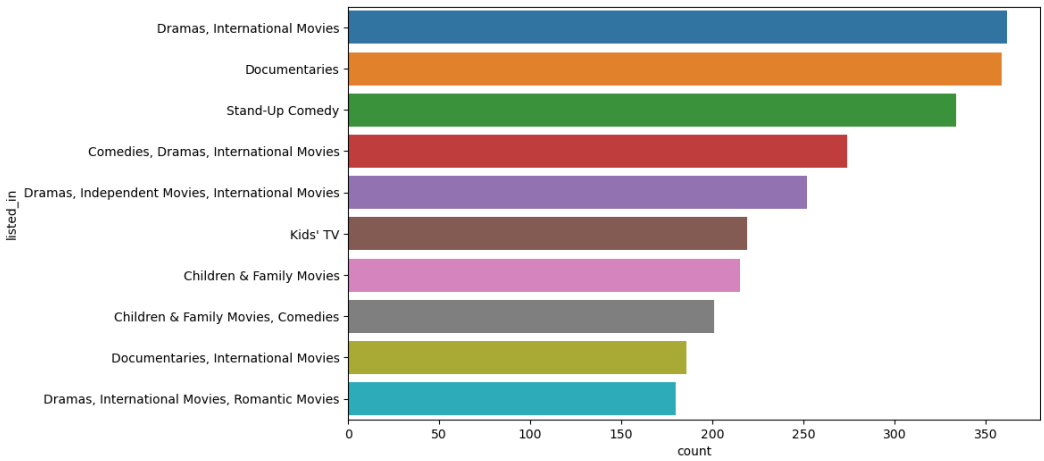


Insights -

1) Most number of movies & TV shows are produced by United States ,
followed by India (2nd most number of movies on Netflix)

```
# listed_in (Genres) -
plt.figure(figsize= (10, 6))
sns.countplot(y = "listed_in", order = netflix["listed_in"].value_counts().index[:10] ,data = netflix)
```

```
plt.show()
```

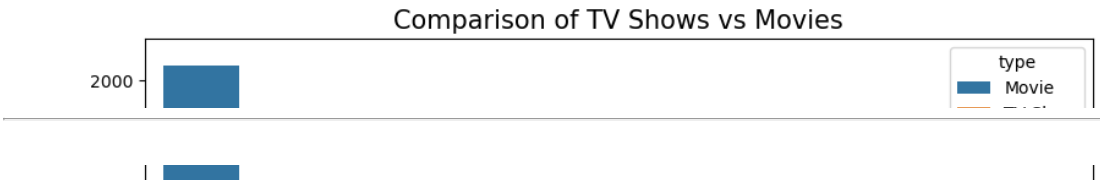


Insights -

1) International movies & dramas are the most popular Genres on Netflix.

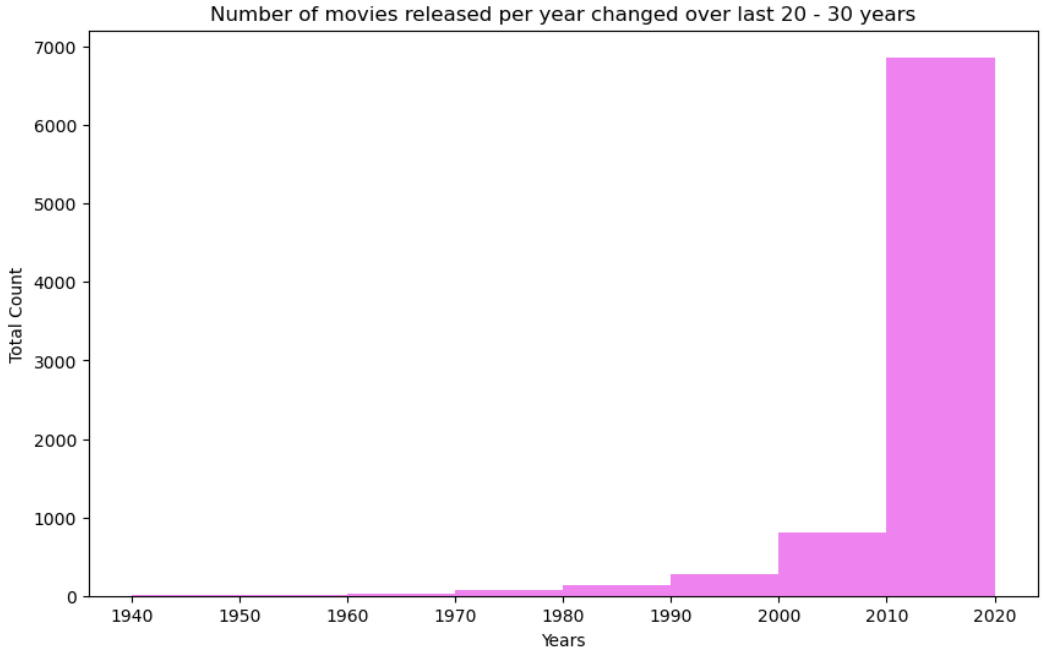
Comparison of TV shows vs Movies (OR) Content available in different countries -

```
plt.figure(figsize= (10, 6))
sns.countplot(x = "country", order = netflix["country"].value_counts().index[:5], hue = "type", data = netflix)
plt.title("Comparison of TV Shows vs Movies", fontsize = 15)
plt.xticks(rotation = 90)
plt.xlabel("Country", fontsize = 12)
plt.ylabel("Count", fontsize = 12)
plt.show()
```



Number of movies released per year changed over last 20 - 30 years -

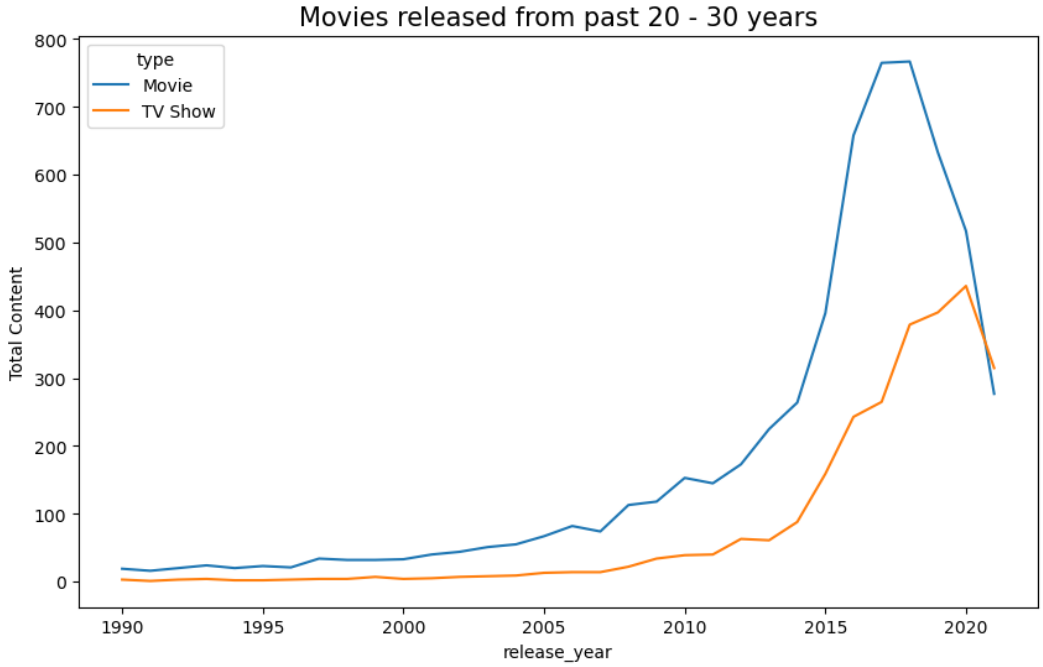
```
# Histogram -
plt.figure(figsize=(10,6))

plt.title('Number of movies released per year changed over last 20 - 30 years')

plt.xlabel('Years')

plt.ylabel('Total Count')

plt.hist(netflix.release_year, bins = np.arange(1940, 2025, 10), color = 'violet')

plt.show()
```



Insights -

```
1) Since the start of OTT platforms, after 2010, there is drastic increase in count of movies
compared to past 20 - 30 years span.
2) Maximum Movies are released in between 2010 to 2020.
3) Minimum Movies are released in between 1950 to 1960.
4) More than 6500 Movies were released in 2010 to 2020 that's why we can say that there was increased in
employment in between 2010 to 2020 in Film Industry.
```

```
#Lineplot Approach -
plt.figure(figsize= (10, 6))
df1 = netflix[['type','release_year']]

df2 = df1.groupby(['release_year','type']).size().reset_index(name='Total Content')

df2 = df2[df2['release_year'] >= 1990]

sns.lineplot(data = df2, x="release_year", y="Total Content", hue = "type")

plt.title("Movies released from past 20 - 30 years", fontsize = 15)

plt.show()
```

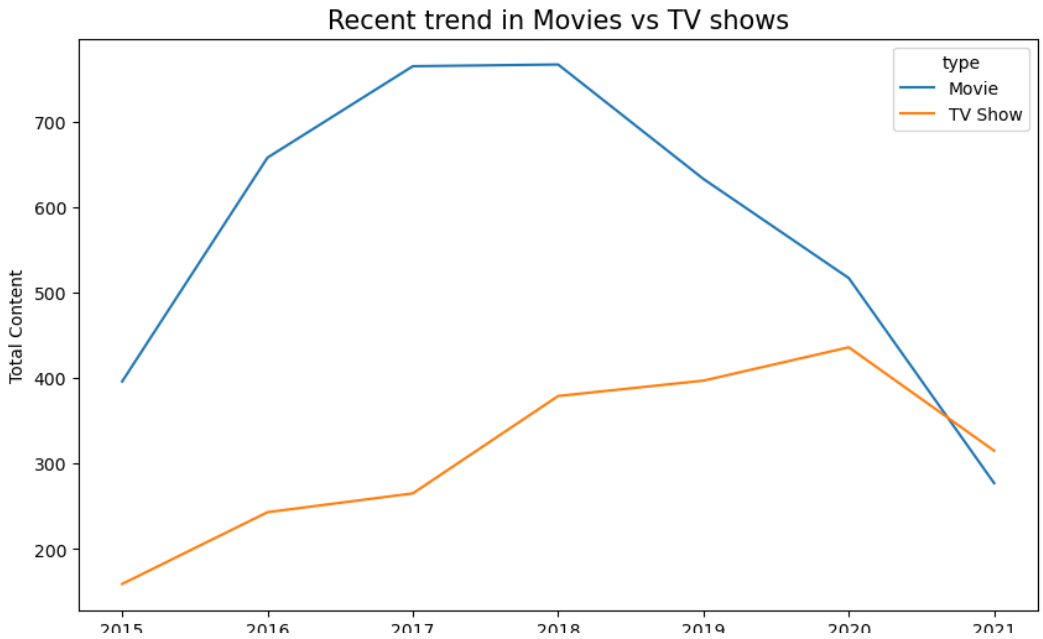


Recent trend in movies vs TV shows -

```
df1 = netflix[['type','release_year']]

df2 = df1.groupby(['release_year','type']).size().reset_index(name='Total Content')

df2 = df2[df2['release_year'] >= 2015]

df2
```

	release_year	type	Total Content
105	2015	Movie	396
106	2015	TV Show	159
107	2016	Movie	658
108	2016	TV Show	243
109	2017	Movie	765
110	2017	TV Show	265
111	2018	Movie	767
112	2018	TV Show	379
113	2019	Movie	633
114	2019	TV Show	397
115	2020	Movie	517
116	2020	TV Show	436
117	2021	Movie	277
118	2021	TV Show	315

```
plt.figure(figsize= (10, 6))
sns.lineplot(data = df2, x="release_year", y="Total Content", hue = "type")
plt.title("Recent trend in Movies vs TV shows", fontsize = 15)
plt.show()
```



Insights -

1) Movies line plot got hump in between 2017 to 2018, which means that the count was at the peak.
2) for TV shows the count was increased after 2015 itself but with lesser slope as compared to movies.
After 2020, the count fall down suddenly.

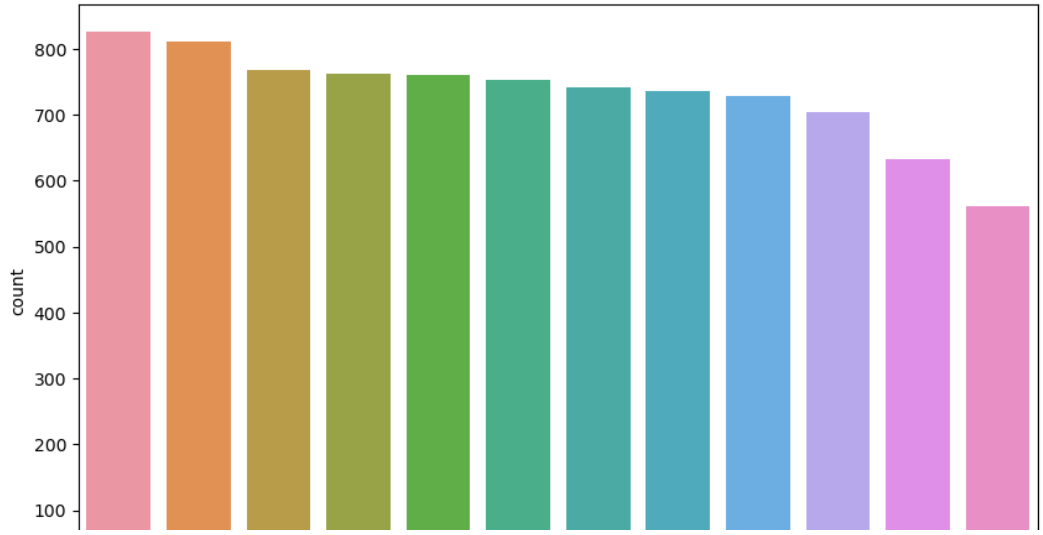
Best time to launch TV Show or Movie -

```
netflix['Month'] = netflix['date_added'].dt.month
```

```
netflix["Month"].value_counts().reset_index()
```

	index	Month
0	7	827
1	12	812
2	9	769
3	4	763
4	10	760
5	8	754
6	3	741
7	1	737
8	6	728
9	11	705
10	5	632
11	2	562

```
plt.figure(figsize= (10, 6))
sns.countplot(x = "Month", order = netflix["Month"].value_counts().index[:12], data = netflix)
plt.show()
```



Insights -

1) From the above bar plot, It is clear that the month of july has highest count of movies / TV shows followed by December month.

Gap between release date and date added -

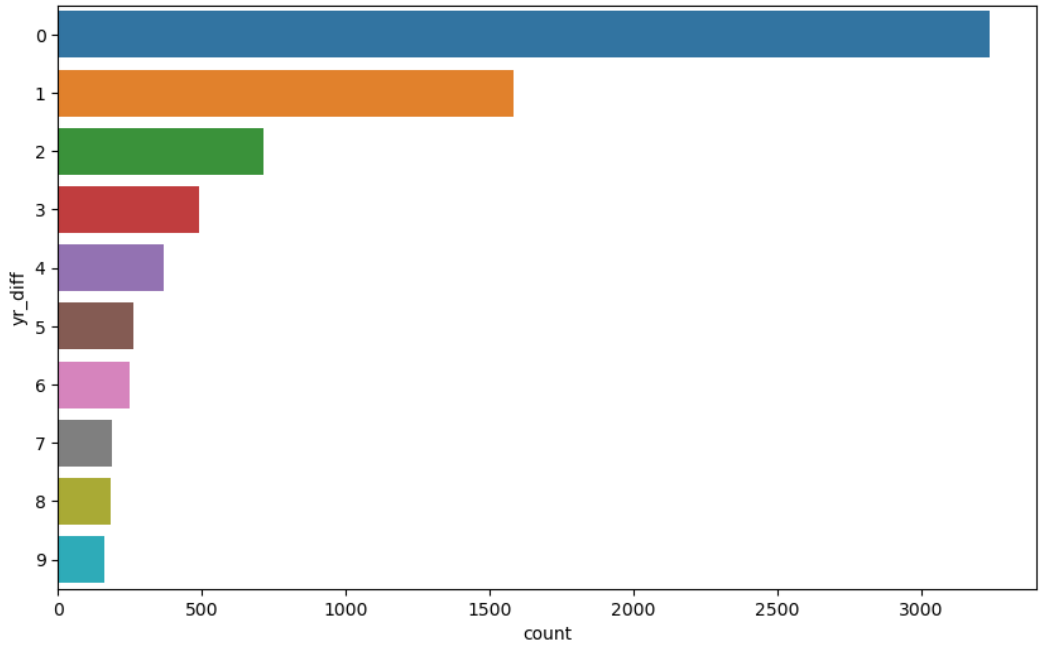
```
netflix["yr_diff"] = netflix["ad_year"] - netflix["release_year"]
```

```
netflix["ad_year"] = netflix["date_added"].dt.year
```

```
netflix["yr_diff"] = netflix["ad_year"] - netflix["release_year"]
```

```
plt.figure(figsize= (10, 6))
sns.countplot(y = "yr_diff", order = netflix["yr_diff"].value_counts().index[:10], data = netflix)
```

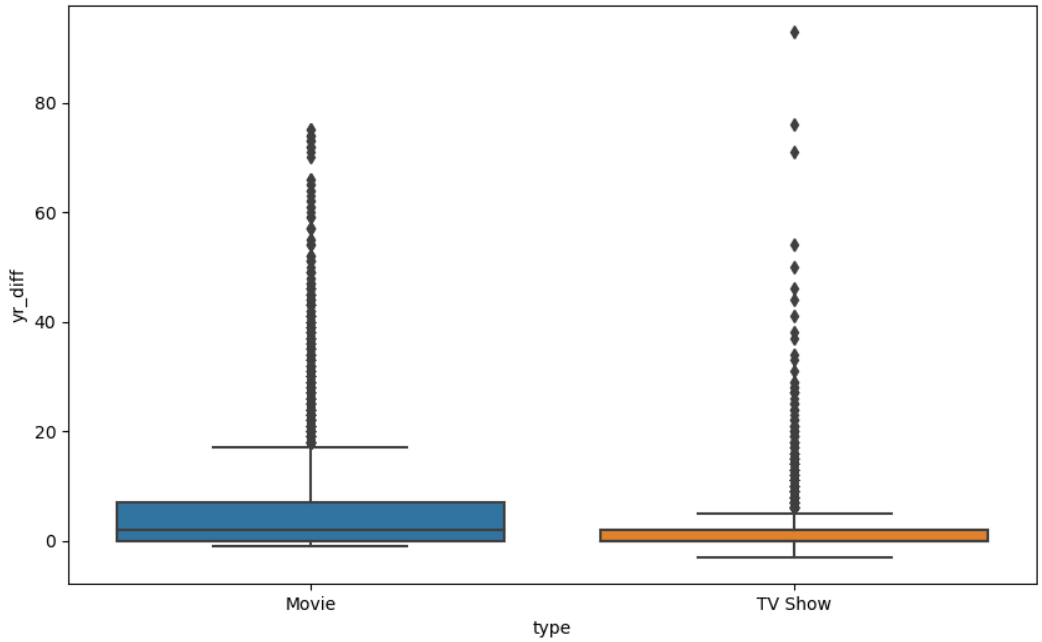
```
<Axes: xlabel='count', ylabel='yr_diff'>
```




```
# Box plot -
plt.figure(figsize= (10, 6))
```

```
sns.boxplot(x = "type", y = "yr_diff", data = netflix)
```

```
<Axes: xlabel='type', ylabel='yr_diff'>
```



Insights -

1) From above box plot, median value signifies that there are movies / TV shows having nearly 0 years difference in release_date & added_date.
2) For 1 year Difference = count is greater than 1500.

Correlation -

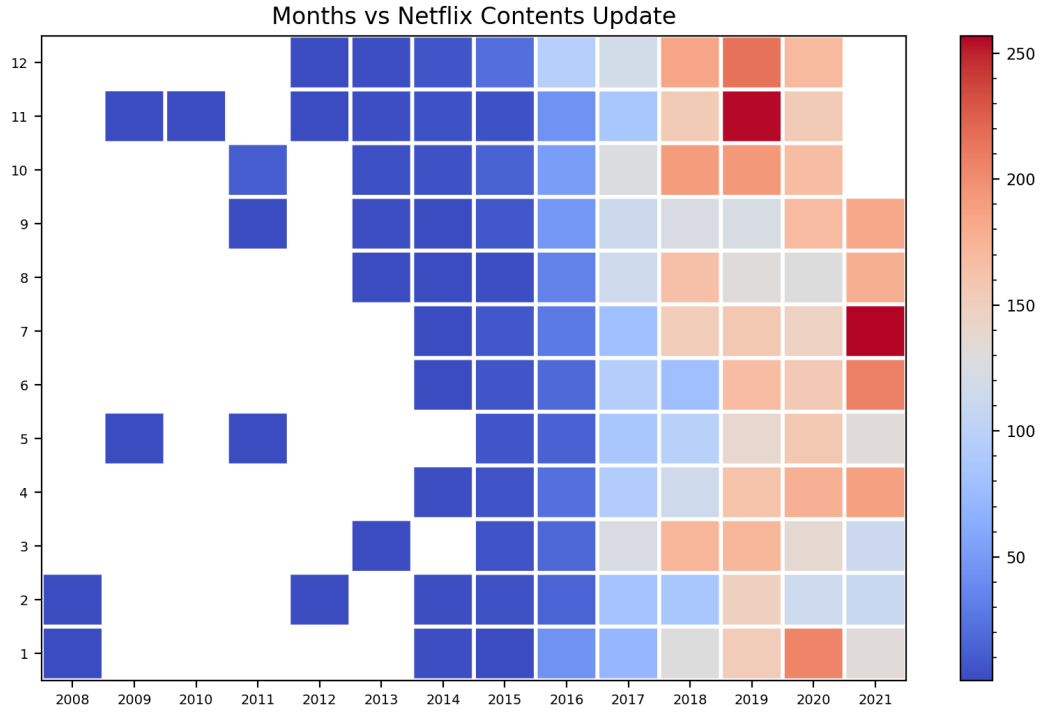
```
netflix.corr()
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_4744\1972714546.py:1: FutureWarning: The default value of nume
  netflix.corr()
```

	release_year	Month	ad_year	yr_diff
release_year	1.000000	-0.039031	0.111624	-0.984049
Month	-0.039031	1.000000	-0.160650	0.010429
ad_year	0.111624	-0.160650	1.000000	0.066943
yr_diff	-0.984049	0.010429	0.066943	1.000000

```
new_df = netflix.groupby("ad_year")["Month"].value_counts().unstack().T
```

```
plt.figure(figsize = (10,6), dpi = 200)
plt.pcolor(new_df, cmap = "coolwarm", edgecolors =  "white", linewidths = 2)
plt.xticks(np.arange(0.5, len(new_df.columns), 1), new_df.columns, fontsize = 7)
plt.yticks(np.arange(0.5, len(new_df.index), 1), new_df.index, fontsize = 7)
plt.title("Months vs Netflix Contents Update", fontsize = 12)
cbar = plt.colorbar()

cbar.ax.tick_params(labelsize = 8)
cbar.ax.minorticks_on()
plt.show()
```



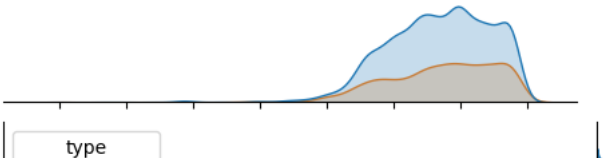
Insights -

1) The above heatmap shows the relationship between Months & netflix content update.
2) Here Dec-2019 & July-2021 have the highest monthly content updates.

Joint Plot -

```
plt.figure(figsize= (12, 8))
sns.jointplot( x = "date_added", y = "release_year", data = netflix, hue = "type")
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



Insights -

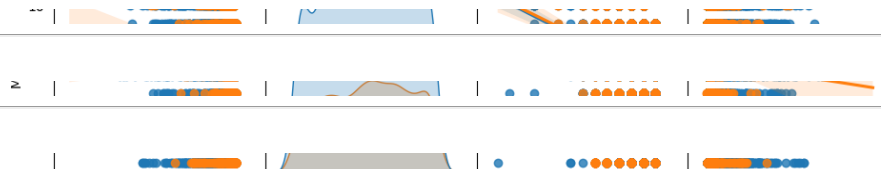
1) The relation between date_added & release_year is shown from above joint plot.
2) Here, after 2018, there is not much difference in release_year & date_added
because in graph we can see lot of points are concentrated in the region of 2018-2022.
3) Movies are started releasing after 1940.
4) According to the data given Movies and TV Shows are started listed in Netflix after 2008.
5) Maximum TV Shows are listed in Netflix after 2013.
6) In starting, listing of Movies are very slow in Netflix. It has rapidly increased after 2014.

Pair Plot -

```
sns.pairplot(data = netflix, hue = "type", kind = "reg")
plt.show()
```

Insights -

1) From above pair plot, we get summary of upward and downward trend of various continuous variables.
2) year difference is inversely proportional to release_year which makes sense as it tells that
now a days if any movie releases, it will broadcast on any OTT platform within months.
3) Positive correlation can be seen for release_year & ad_year.



Summary of Project on Netflix Dataset:

1) The goal of this exploratory data analysis (EDA) project was to gain insights and understanding from the Netflix dataset.

2) The project involved data preparation and cleaning, exploratory analysis and visualization, asking
and answering questions about the data, and summarizing the inferences.

Data Preparation & Cleaning:

1) Initial data inspection was performed to understand the structure and content of the dataset.

2) Missing values were handled by either dropping rows/columns or imputing values based on the context.

3) Data cleaning tasks such as handling duplicates and transforming data types were carried out.

Exploratory Analysis & Visualization:

1) Relevant features in the dataset were identified for analysis.

2) Categorical variables were explored by counting the occurrences of each category.

3) Visualizations using matplotlib and seaborn were created to gain insights into the data,
such as histograms, bar charts, and box plots.

Business Insights -

1) Movies constitute approximately 69.6% of Netflix's content, whereas TV shows make up the remaining 30.4%.

2) By seeing this data, the demand of Netflix has increased after 2014 only.

3) With more than 6500 movies released between 2010 and 2020,
this period saw a notable increase in employment opportunities within the film industry.

4) Netflix's growth is evident from the data, showcasing their marketing strategies to enter new global markets.
According to Business Insider, Netflix had approximately 158 million subscribers worldwide,

with 60 million in the US and nearly 98 million internationally.

5) Initially, Netflix's subscribers were mainly from the US,
but their decision to expand internationally played a major role in their success.

6) Content selection is influenced by popular markets, leading to the addition of numerous international movies
and TV shows during Netflix's global expansion.

Recommendations -

1) As we can see that the business is at peak in countries like USA and India,
so netflix should also target asian countries like Japan, Russia as well as European countries like France & UK
to increase their viewership.

2) As per the comparison of TV shows & movies data, netflix should also concentrate on producing TV shows
so that people who love to watch TV shows would come back to netflix platform.

3) The content on Netflix which are decreasing at the end of 2020. Netflix should produce more & more content
so that in the race OTT platform, they will secure their position with billions of subscribers in the future.

4) At the end of every movie or TV shows, Netflix should take feedback from each & every customer,
so that it will help them to produce relevant content which their subscribers want.

✓ 0s completed at 21:15