NETFLIX CASE STUDY

Problem Statement

Analyse 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 required Python Libraries -

In [466…
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading the dataset -

In [467…
```
df = pd.read_csv('/Users/bose/Desktop/netflix.csv')
```

Analysing Basic Metrics

In [468…
```
df.columns
```

Out[468]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_
added',
       'release_year', 'rating', 'duration', 'listed_in', 'description']
,
      dtype='object')
```

In [469…
```
df.head()
```

Out[469]:

	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

Shape of data -

In [470…
```
df.shape
```

Out[470]:
```
(8807, 12)
```

Datatype of columns -

In [471…
```
df.dtypes
```

Out[471]:
```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

In [472…
```
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
```

Statistical Summary

In [473…
```
df.describe()
```

Out[473]:

	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

In [474…
```
df.describe(include='object')
```

Out[474]:

	show_id	type	title	director	cast	country	date_added	rating	
count	8807	8807	8807	6173	7982	7976	8797	8803	
unique	8807	2	8807	4528	7692	748	1767	17	
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	TV-MA	
freq	1	6131	1	19	19	2818	109	3207	

Missing Values -

In [475…
```
# No of missing values in each column
df.isna().sum().sort_values(ascending = False)
```

Out[475]:
```
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
```

In [476…
```
# Convert the datatype of 'date_added' column from object to datetime64
df["date_added"] = pd.to_datetime(df["date_added"])
```

In [477…
```
# Replacing missing values in 'director' column with 'No Director'
df['director'].replace(np.NaN, 'No Director', inplace=True)
```

In [478…
```
# Replacing missing values in 'country' column with 'No Country'
df['country'].replace(np.NaN, 'No Country', inplace=True)
```

In [479…
```
# Replacing missing values in 'cast' column with 'No Cast'
df['cast'].replace(np.NaN, 'No Cast', inplace=True)
```

In [480…
```
# No of missing values in each column
df.isna().sum().sort_values(ascending = False)
```

Out[480]:
```
date_added      10
rating           4
duration         3
show_id          0
type             0
title            0
director         0
cast             0
country          0
release_year     0
listed_in        0
description      0
dtype: int64
```

In [481…
```
# Dropping rows with low number of missing values
df.dropna(inplace=True)
```

In [482…
```
# No of missing values in each column after above operations
df.isna().sum().sort_values(ascending = False)
```

Out[482]:
```
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
```

Unnesting of data in columns 'director', 'cast', 'country'

In [483…
```
# Unnesting the 'cast' column
cast = df['cast'].str.split(',', expand=True).stack()
cast = cast.reset_index(level=1, drop=True).to_frame('cast')
cast['show_id'] = df['show_id']
```

In [484…
```
# Unnesting the 'director' column
director = df['director'].str.split(',', expand=True).stack()
director = director.reset_index(level=1, drop=True).to_frame('director')
director['show_id'] = df['show_id']
```

In [485…
```
# Unnesting the 'country' column
country = df['country'].str.split(',', expand=True).stack()
country = country.reset_index(level=1, drop=True).to_frame('country')
country['show_id'] = df['show_id']
```

Non-Graphical Analysis

In [486…
```
df['type'].value_counts()
```

Out[486]:
```
Movie      6126
TV Show    2664
Name: type, dtype: int64
```

*Netflix has a collection of **6126 Movies** and **2664 TV Shows***

In [487…
```
country['country'].nunique()
```

Out[487]:
```
198
```

*Netflix has content from over **198 different countries***

In [488…
```
# Top 5 Directors
director['director'].value_counts()[1:6]
```

Out[488]:
```
Rajiv Chilaka     22
Raúl Campos       18
 Jan Suter        18
Marcus Raboy      16
Suhas Kadav       16
Name: director, dtype: int64
```

In [489…
```
# Top 5 Actors
cast['cast'].value_counts()[1:6]
```

Out[489]:
```
 Anupam Kher          39
 Rupa Bhimani         31
 Takahiro Sakurai     30
 Julie Tejwani        28
 Om Puri              27
Name: cast, dtype: int64
```

In [490…
```
df['rating'].nunique()
```

Out[490]:
```
14
```

*Netflix has **14 unique ratings** given to thier content. They are -*

In [491…
```
df['rating'].unique()
```

Out[491]:
```
array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
       'TV-G', 'G', 'NC-17', 'NR', 'TV-Y7-FV', 'UR'], dtype=object)
```

In [492…
```
# Unnesting the 'listed_in' column
genre = df['listed_in'].str.split(',', expand=True).stack()
genre = genre.reset_index(level=1, drop=True).to_frame('listed_in')
genre['show_id'] = df['show_id']
```

In [493…
```
genre['listed_in'].nunique()
```

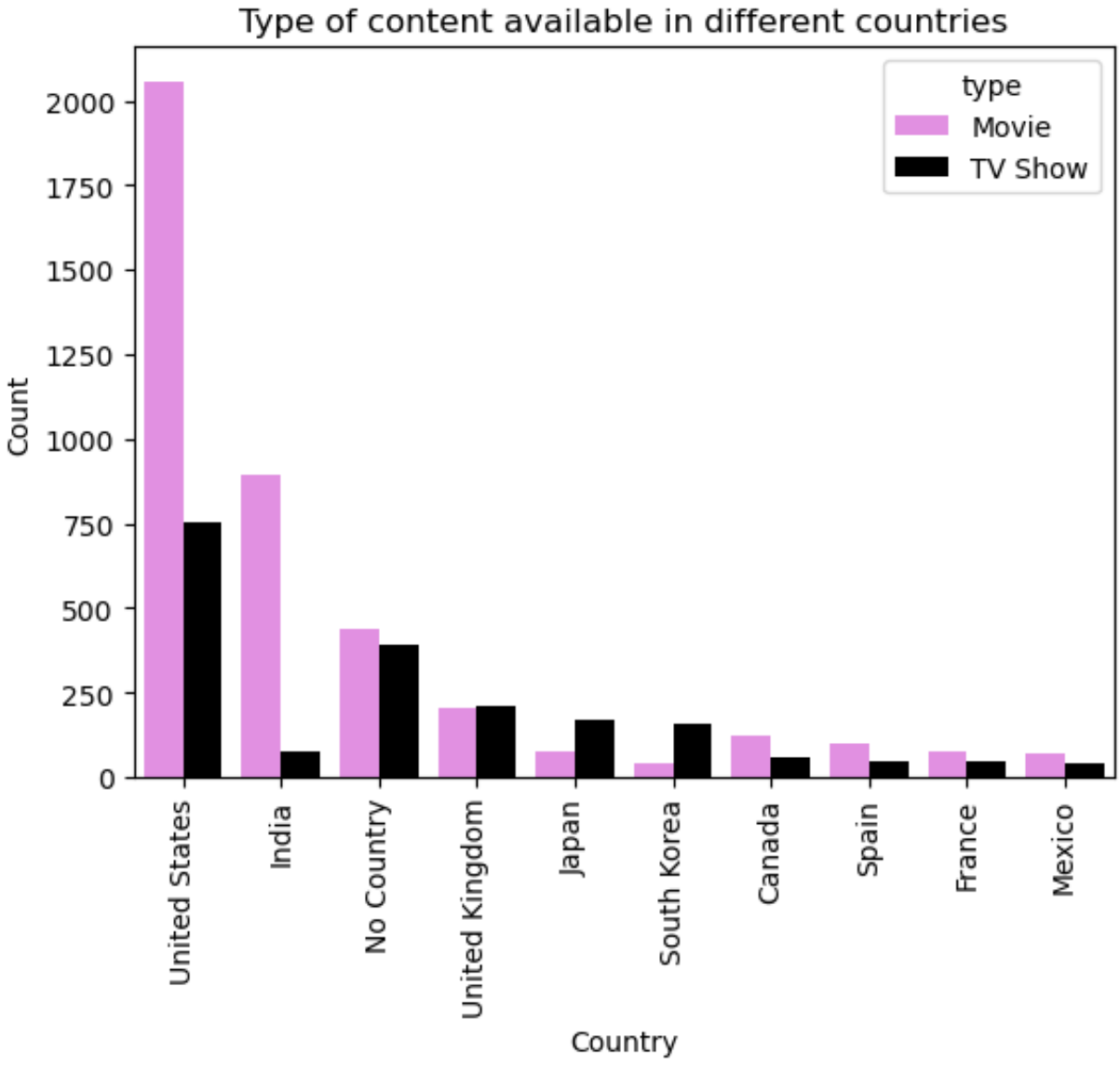
Out[493]:
```
73
```

*Netflix has content form **73 unique genre**. The top Genres are*

In [494…
```
genre['listed_in'].value_counts()
```

Out[494]:
```
 International Movies          2624
Dramas                         1599
Comedies                       1210
Action & Adventure              859
Documentaries                   829
                               ...
Romantic Movies                   3
Spanish-Language TV Shows         2
LGBTQ Movies                      1
TV Sci-Fi & Fantasy               1
Sports Movies                     1
Name: listed_in, Length: 73, dtype: int64
```

In [612…
```
sns.countplot(x= 'country', order= df['country'].value_counts().index[:10
plt.title('Type of content available in different countries')
plt.xticks(rotation = 90)
plt.xlabel('Country')
plt.ylabel('Count')
plt.show()
```



Insights -

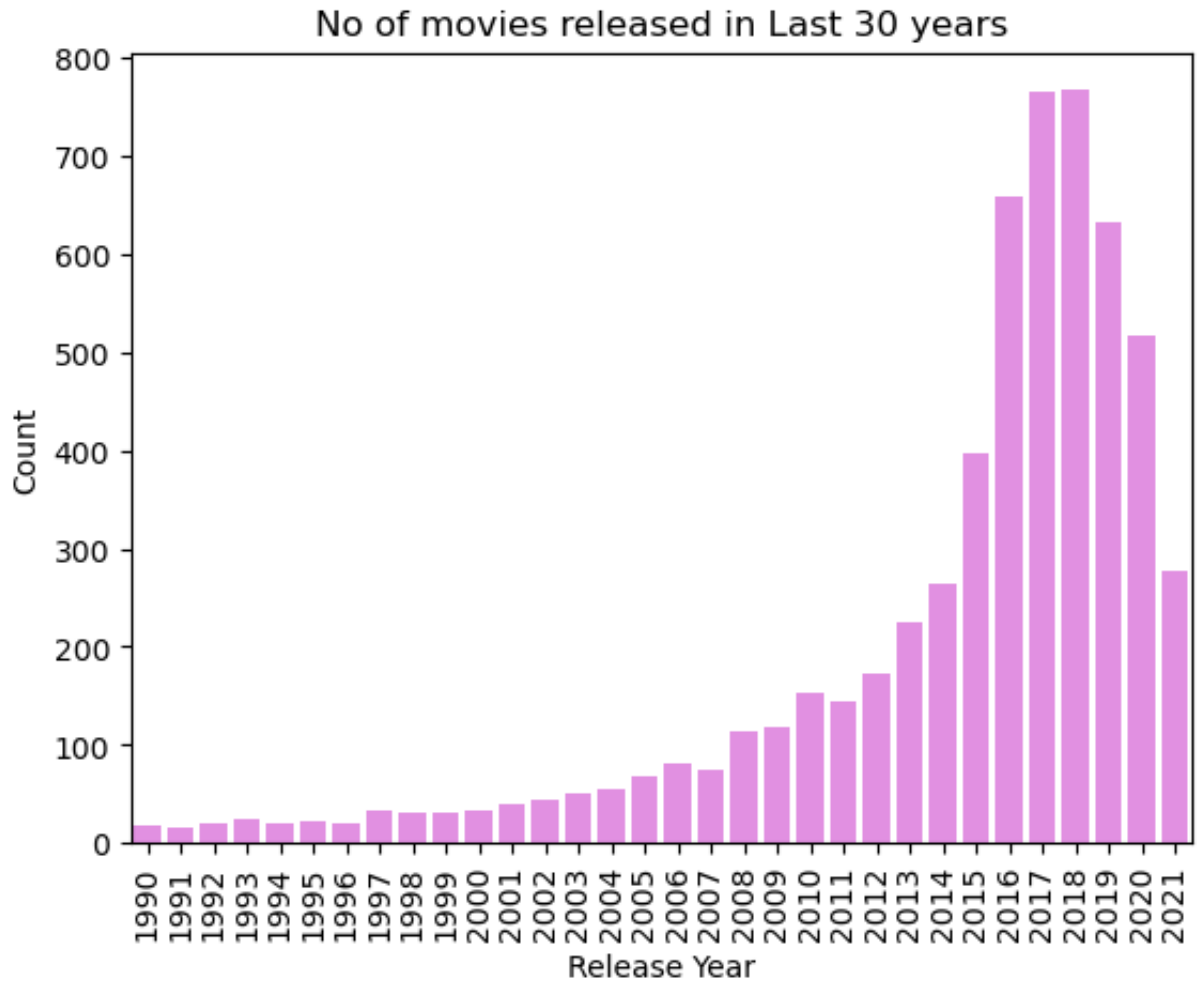
1. **Movies** are more popular in countries like **United States and India**
2. Whereas in countries like **Japan and South Korea**, **TV Shows** are more popular
3. Looking at the complete graph we can say that **Movies are more Popular than TV Shows**

Recommendations -

1. Netflix should **add more Movies for the global audience as they are more popular than TV Shows**
2. Whereas when coming to **specific countries like Japan and South Korea, they need to change their strategy and include more TV Shows than movies** in their platform

In [613…

```
a = df[(df['type'] == 'Movie')&(df['release_year']>= 1990)]
sns.countplot(x= 'release_year', data=a, color='violet')
plt.title('No of movies released in Last 30 years')
plt.xticks(rotation=90)
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.show()
```



Insights -

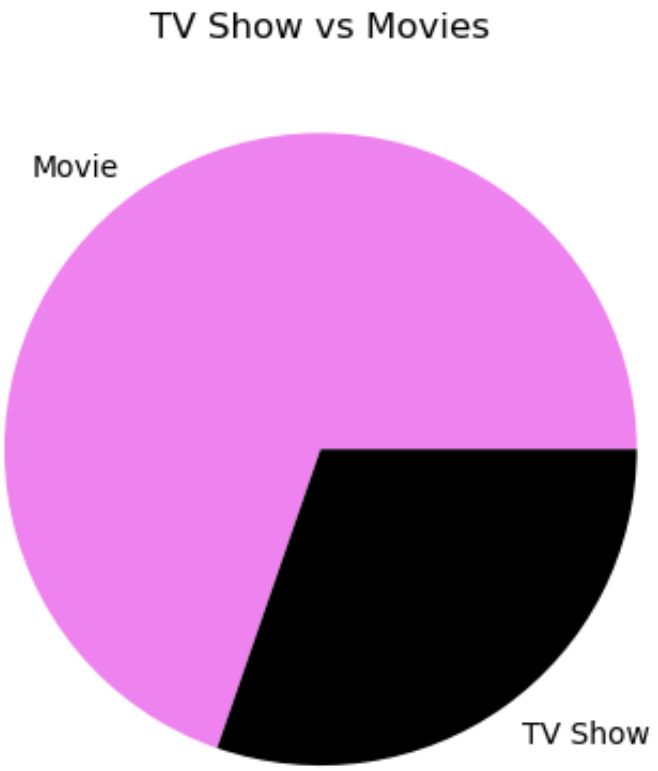
1. The **No of movies released per year has been steadily increasing in the past 30 years**
2. Altough it has **gone down a bit after 2018**. This might be due to Covid-19 outbreak
3. Maximum no of movies was relesed in the year 2018
4. Minimum no of movies was released in the year 1991

Recommendation -

1. Netflix should keep on adding more movies to their portfolio
2. This will keep their growth trajectory intact and reduce customer churn

In [639…

```
d = df['type'].value_counts()
colors = ['violet','black']
plt.pie(d.values, labels=d.index, colors=colors)
plt.title('TV Show vs Movies')
plt.show()
```



Insights -

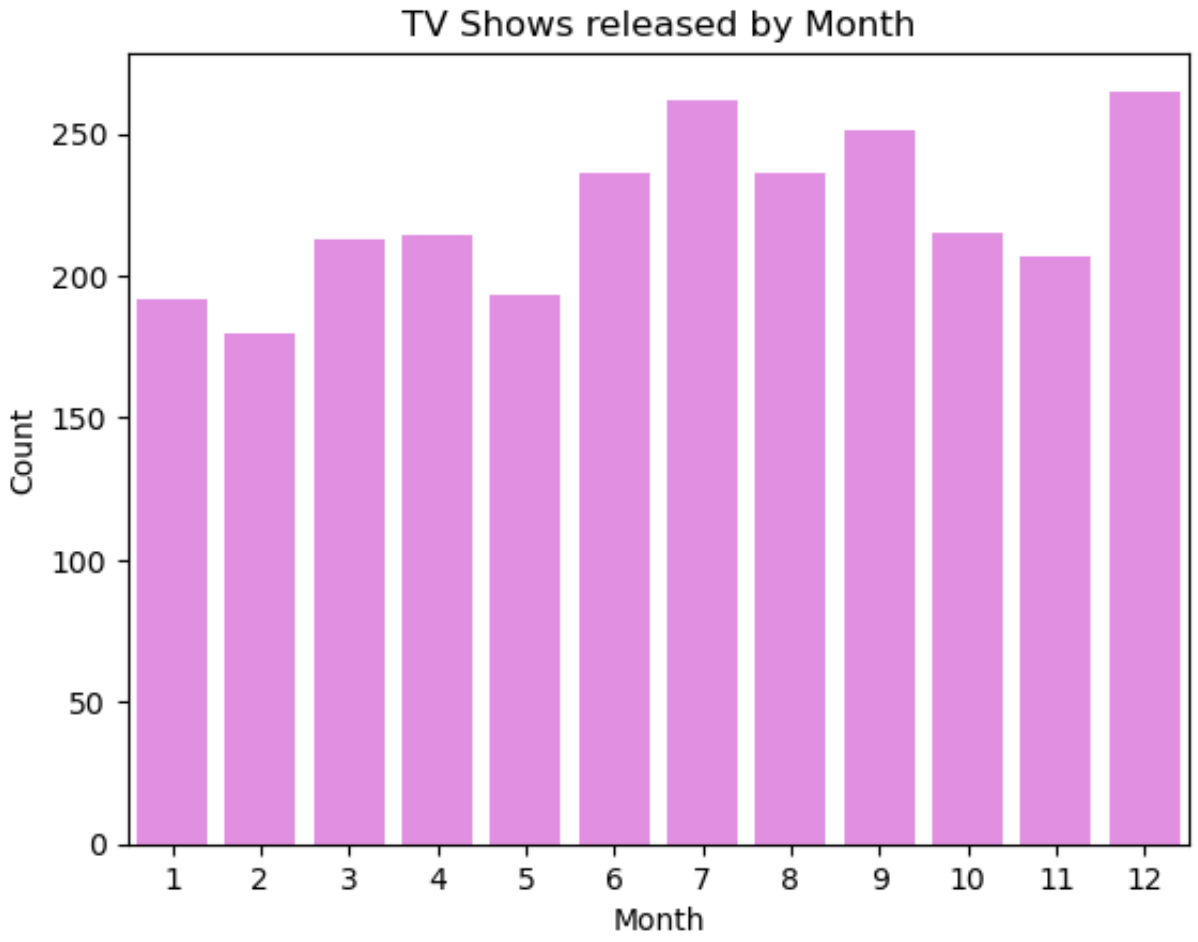
1. Netflix has more movies compared to TV Shows
2. Movies account for about 70% of the total content available on Netflix
3. Whereas TV Show accounts only for only 30%
4. There is opportunity for growth in TV Shows for Netflix

Recommendation -

1. Netflix can add more content to the TV Shows section
2. They can add hit shows from other franchises just to increase their customer base
3. And then build upon it by producing more quality content in thier own banner

In [638…
```
tv_show['month'] = tv_show['date_added'].dt.month
```

In [610…
```
plt.title('TV Shows released by Month')
sns.barplot(x=tv_show['month'].value_counts().index,y=tv_show['month'].va
plt.xlabel('Month')
plt.ylabel('Count')
plt.show()
```

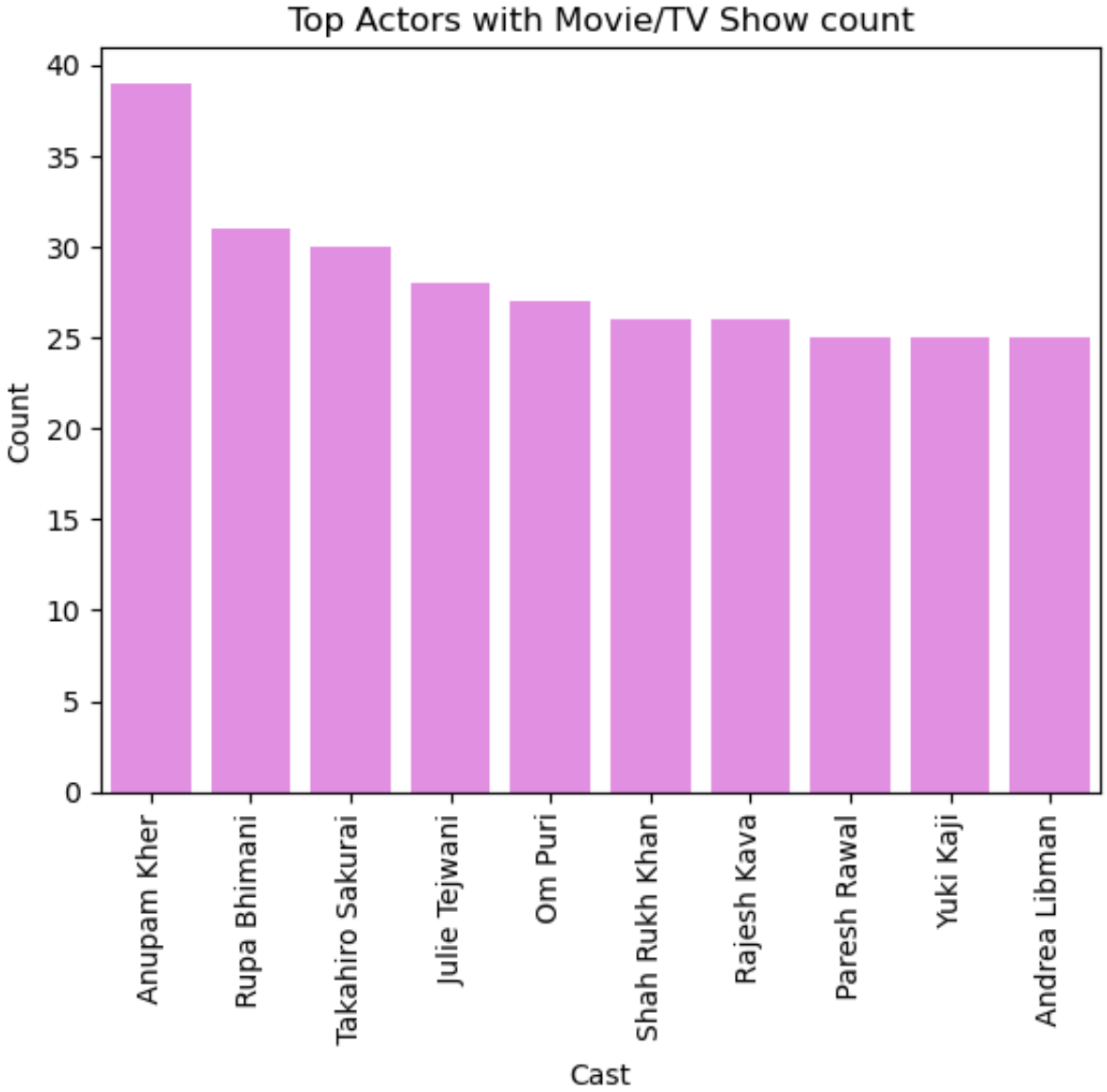


Insights -

1. **July** and **December** are the best months to release new TV Shows
2. These are the months netflix adds most TV Shows so people will be expecting new releases

Recommendation - 1.Netflix can increase the no of TV Shows released during the months of January, February and May as they are below the average no of TV Shows released through the year

In [615…
```
sns.countplot(x= 'cast', order= cast['cast'].value_counts(ascending=False
plt.title('Top Actors with Movie/TV Show count')
plt.xticks(rotation = 90)
plt.xlabel('Cast')
plt.ylabel('Count')
plt.show()
```

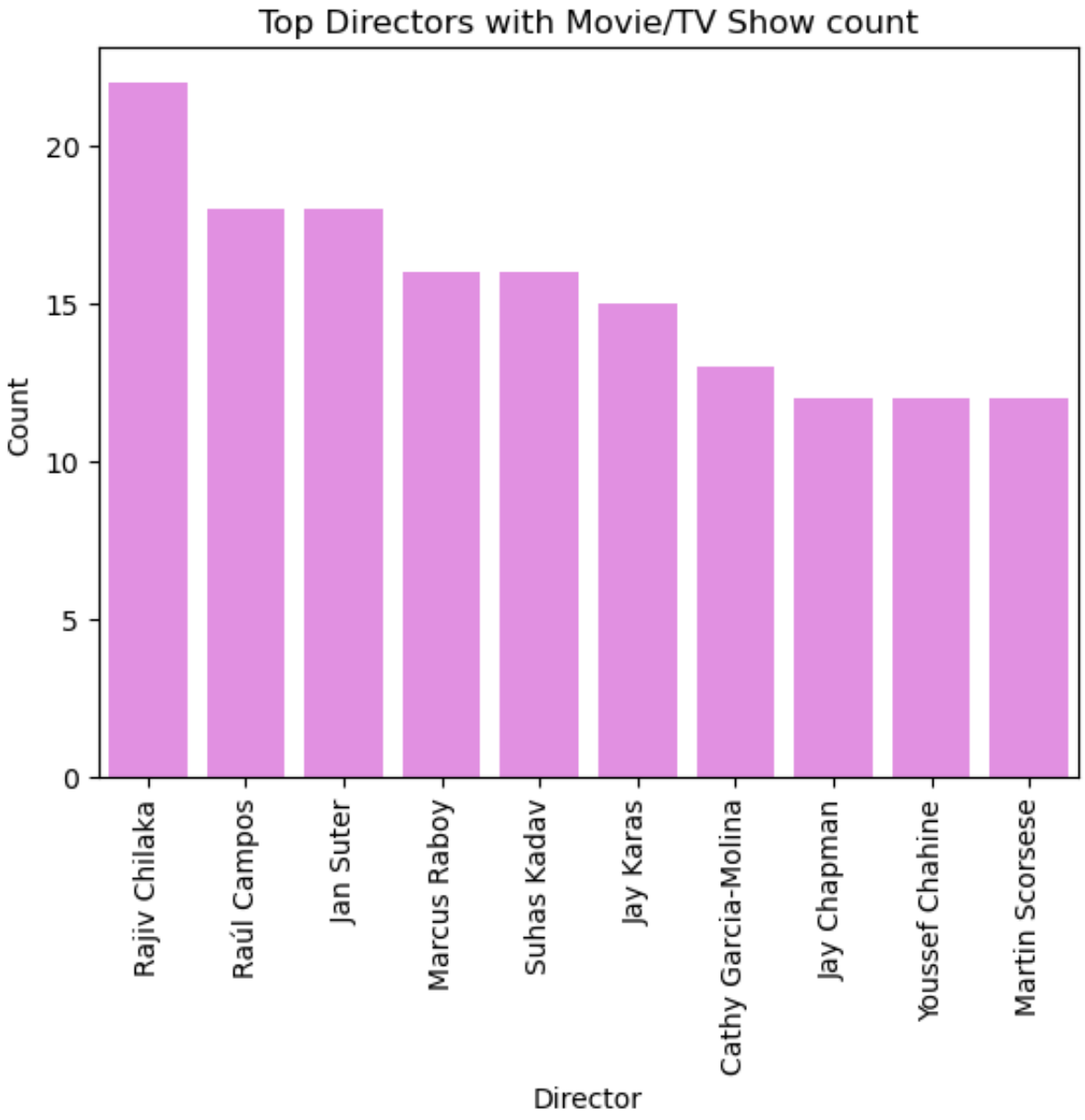


Insights -

1. **Anupam Kher** is the actor with most no of Movies/TV Shows
2. He has done **39** Movies/TV Shows

In [616…

```
sns.countplot(x= 'director', order= director['director'].value_counts(asc
plt.title('Top Directors with Movie/TV Show count')
plt.xticks(rotation = 90)
plt.xlabel('Director')
plt.ylabel('Count')
plt.show()
```

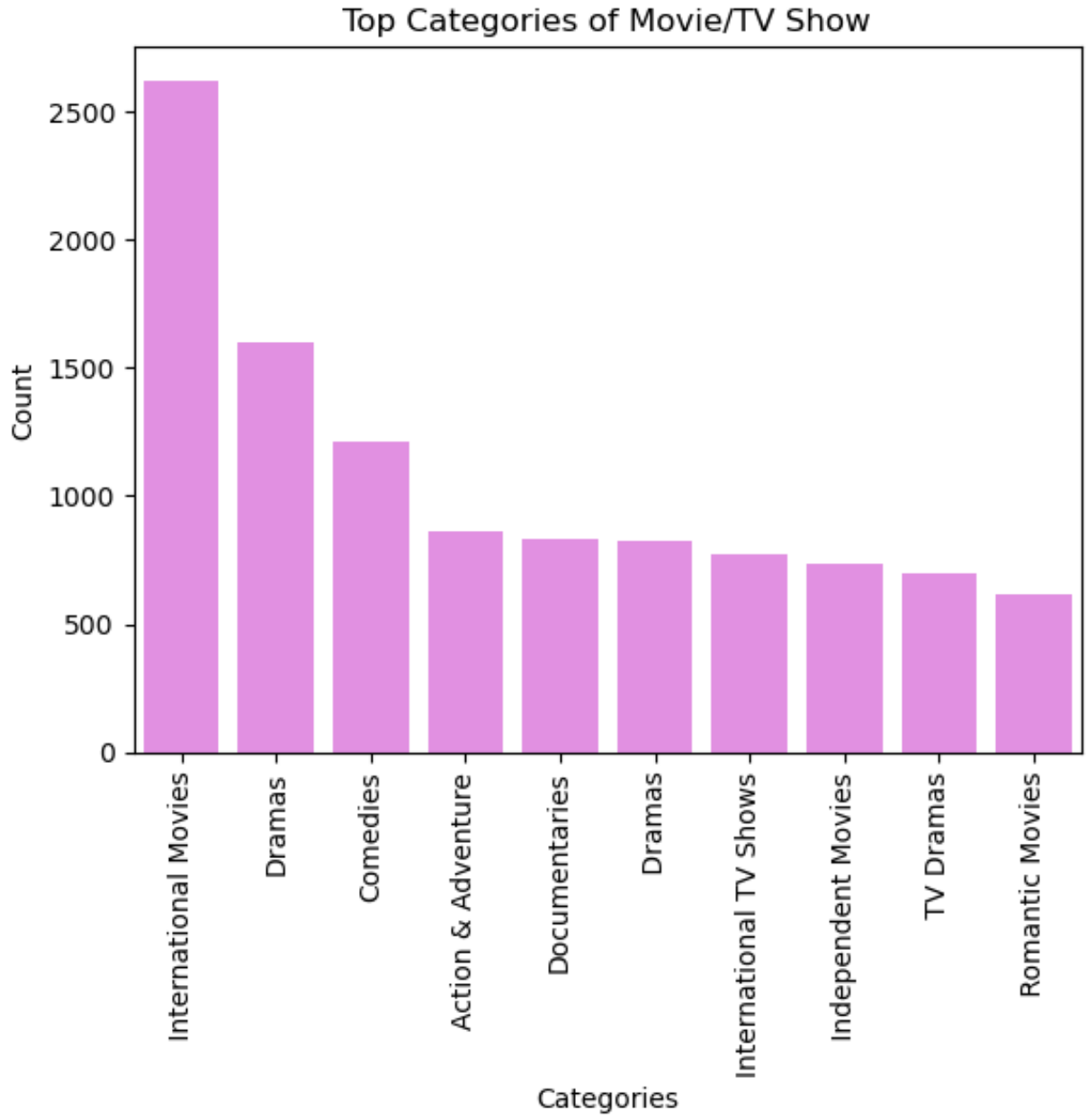


Insights -

1. **Rajiv Chilaka** is the director with most no of Movies/TV Shows
2. He has directed **22** Movies/TV Shows

In [617…

```
sns.countplot(x= 'listed_in', order= genre['listed_in'].value_counts(asce
plt.title('Top Categories of Movie/TV Show')
plt.xticks(rotation = 90)
plt.xlabel('Categories')
plt.ylabel('Count')
plt.show()
```



Insights -

1. **International Movies** is the category in which netflix has most Movies/TV Show
2. There are **2624** movies coming under Internation Movies Category

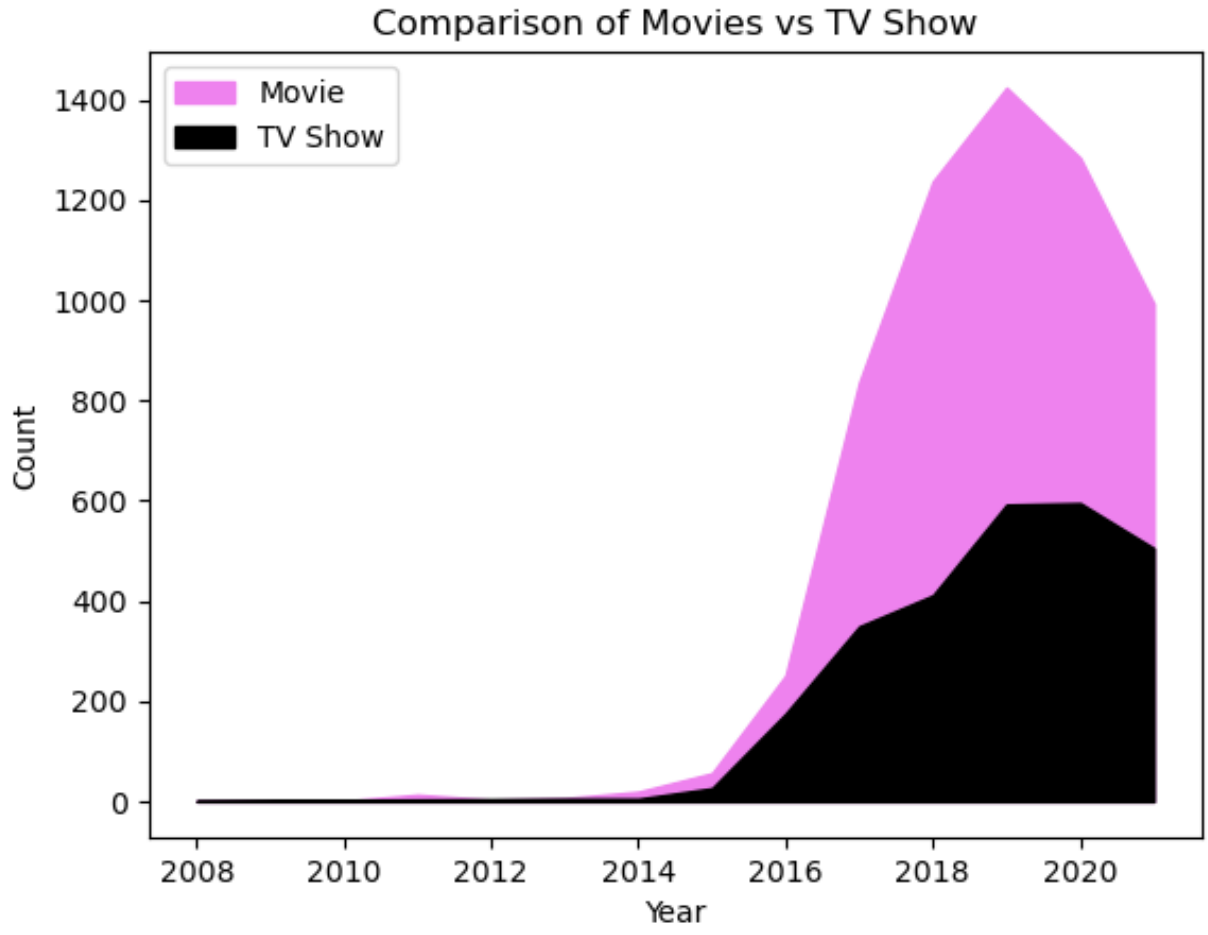
Recommendation -

1. Netflix can **add more dubbed/subbed versions of content in regional languages** to increase the popularity of categories like Comdey, Drama ,Action, etc
2. They can **add more regional content which will increase user engagement**

In [611…

```
movies = df[df['type']=='Movie']
tv_show = df[df['type']=='TV Show']
x_movies = movies['date_added'].dt.year.value_counts().sort_index().index
y_movies = movies['date_added'].dt.year.value_counts().sort_index().value
x_tvshow = tv_show['date_added'].dt.year.value_counts().sort_index().inde
y_tvshow = tv_show['date_added'].dt.year.value_counts().sort_index().valu
plt.title("Comparison of Movies vs TV Show")

plt.fill_between(x_movies, y_movies, color='violet')
plt.fill_between(x_tvshow, y_tvshow, color='black')
plt.xlabel('Year')
plt.ylabel('Count')
plt.legend(['Movie','TV Show'], loc='upper left')
plt.show()
```



Insights -

1. Netflix has been **adding more Movies as compared to TV Shows**
2. From 2008 to 2014 Netflix had similar no of TV shows compared to Movies
3. After 2014 there has been a **shift in trend and netflix started adding more Movies compared to TV Shows**

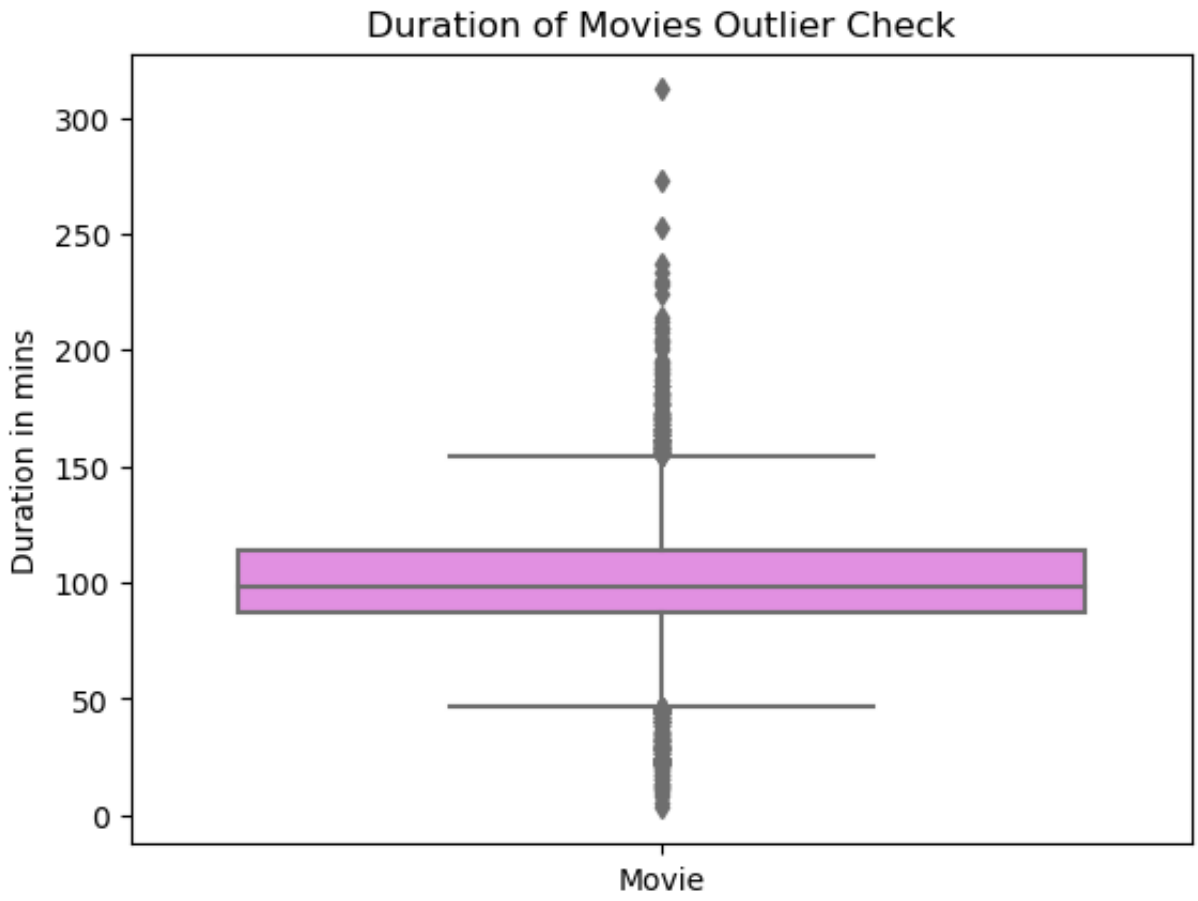
Recommendation -

1. Netflix should try to improve the user engangement for their TV Shows
2. More advertising can be done to increase the popularity of TV Shows

In [603…
```
movies['duration_int'] = movies['duration'].str.extract('(\d+)', expand=F
```

In [606…
```
sns.boxplot(data=movies,x='type',y='duration_int', color='violet')
plt.title('Duration of Movies Outlier Check')
plt.xlabel('')
plt.ylabel('Duration in mins')
```

Out[606]:
```
Text(0, 0.5, 'Duration in mins')
```



Insights -

1. Most movies have a runtime between 90 and 120 mins
2. Minimum value is 50 mins
3. Maximum value is 150 mins
4. Outliers of this category have a runtime above 150 mins

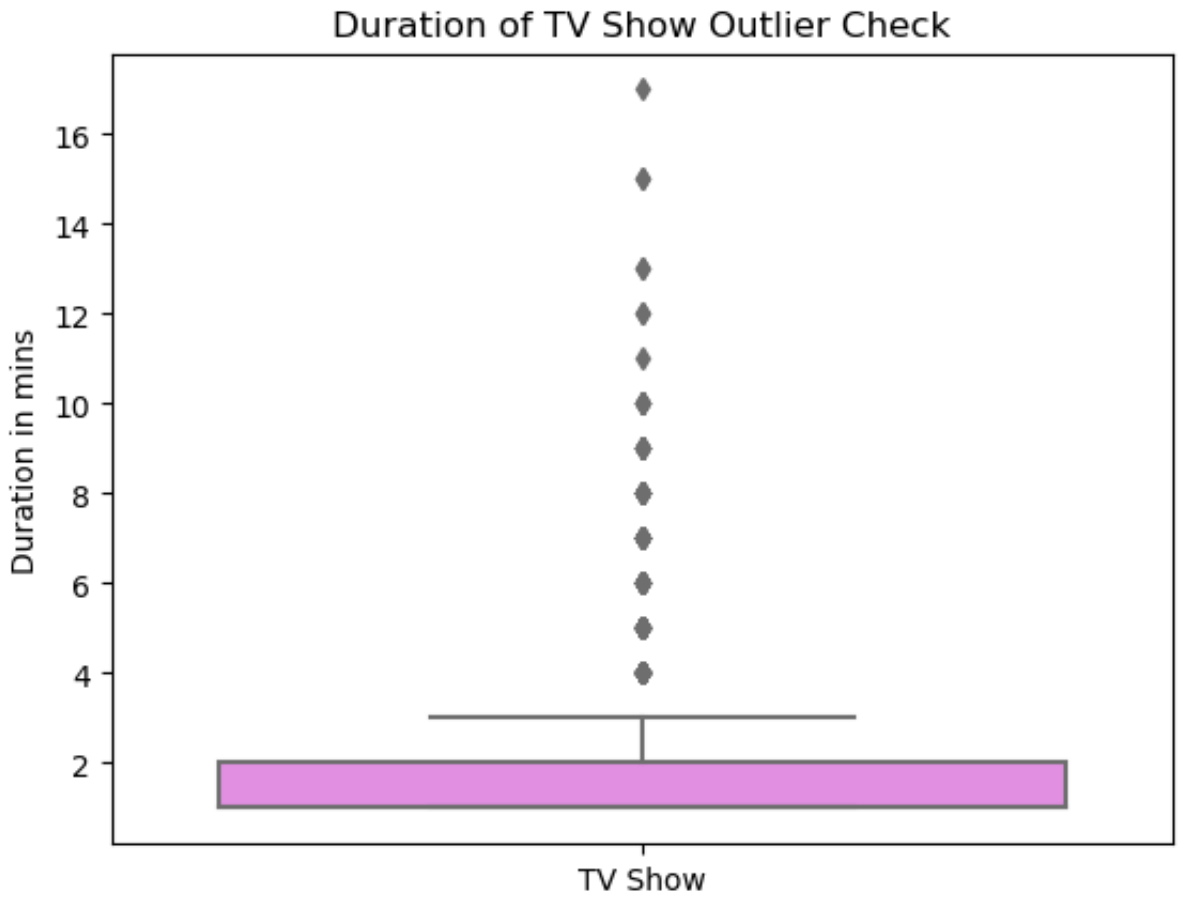
Recommendation -

1. Netflix should focus on including more movie with runtime between 90 and 120 mins
2. Reduce no of movies having runtime above 150 mins

In [602…
```
tv_show['duration_int'] = tv_show['duration'].str.extract('(\d+)', expand
```

In [607…
```
sns.boxplot(data=tv_show,x='type',y='duration_int',color='violet')
plt.title('Duration of TV Show Outlier Check')
plt.xlabel('')
plt.ylabel('Duration in mins')
```

Out[607]:
```
Text(0, 0.5, 'Duration in mins')
```



Insights -

1. Almost all of the series in netflix have only 1 or 2 seasons
2. Minimum no of seasons is 1
3. Maximum no of seasons is 3
4. Outlier in this case are TV Shows having more than 3 seasons

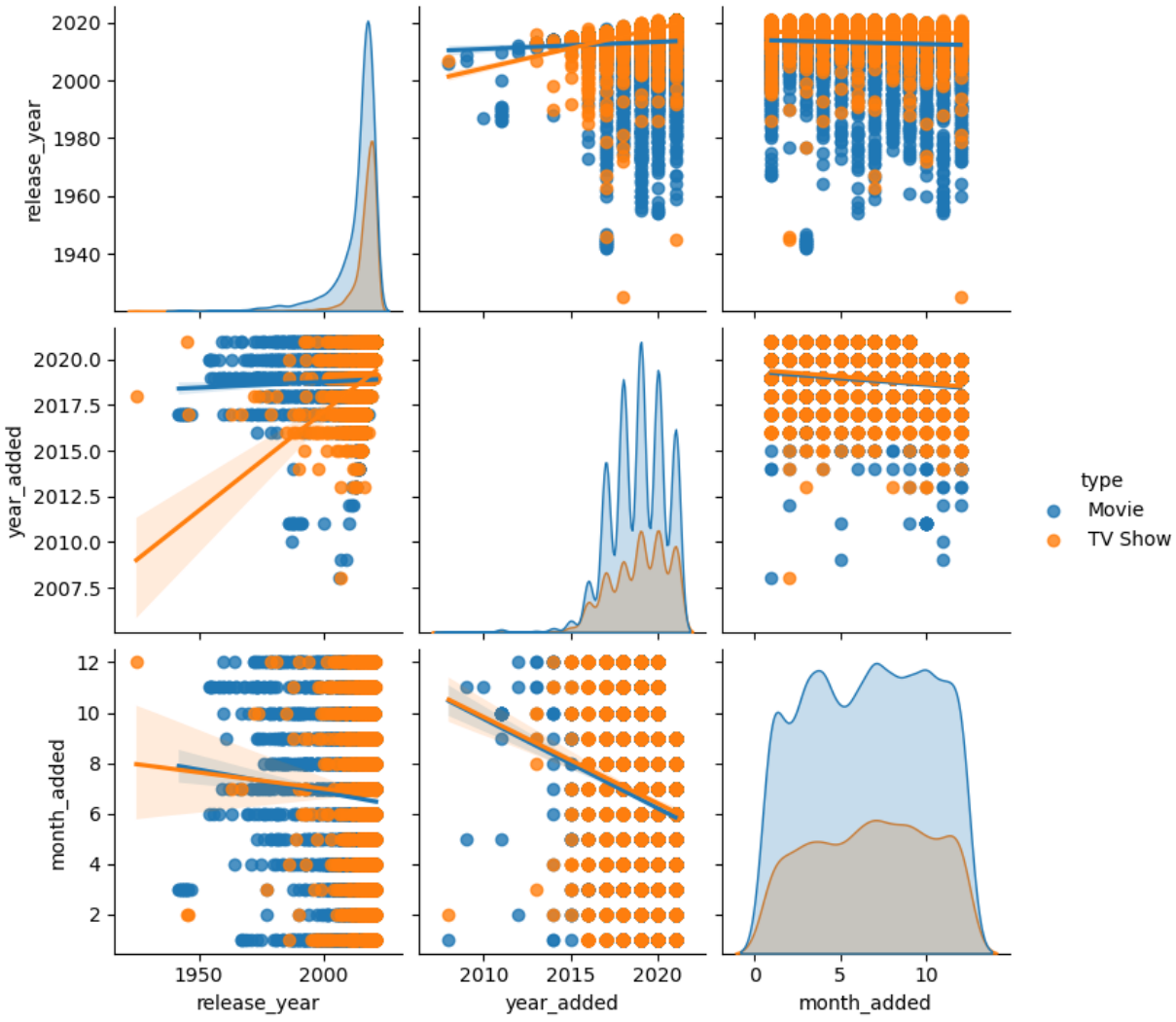
Recommendation -

1. Netflix should add more series having only 1 or 2 seasons
2. Reduce no of TV Shows which has more than 3 seasons

In [511…
```
df['year_added'] = df['date_added'].dt.year
```

In [513…
```
df['month_added'] = df['date_added'].dt.month
```

In [609…
```
sns.pairplot(data=df, hue='type', kind = 'reg')
plt.show()
```



Insights -

1. From the above pairplot we can observe that netflix has been giving more importance to movies than TV Shows
2. Through the years the amount of content getting added to Netflix has increased exponentially

Recommendation -

1. Add more no of TV Shows to the platform
2. Spend more on marketing/advertising on TV Shows

Conclusion

- Netflix already has a good market in countries like USA and India. Now they should try to increase thier user base in other Aisan countries like Japan, South Korea, and Europen countries like Spain and France. They can do so by adding more regional content and including more Movies/TV Shows with the countries regional stars.

- Netflix should add more content in regional languages and other genres like comedy, drama , action to bring in more subscribers to the platform.

- Netflix already has good user engagement when it comes to Movies. They need to do the same with TV Shows going forward. For that they need to spend extra in advertising for their TV Shows like ads on Youtube, Social Media, promoting the content through Influencers.

- Most of the content available in Netflix are new and recent releases. They can add some Old Classic Movies & TV Shows to their portfolio which will be an attraction to people, and therby bring in new subscribers to thier platform. At the same time it will also satisfy the old subscribers.

- There was a dip in the no of movies after 2018. Netflix needs to avoid such conditions in the future inorder to reduce customer churn.

In [ ]: