

# Business Case Study: Popular OTT Streaming Platform

## EDA Using Python

### Context:

This particular business case focuses on the operations of a popular media and video streaming platform, that have over 10000 movies or tv shows available with more than 200 million subscribers globally. This case study aims to analyze the dataset consisting of listings of all the movies and tv shows available on the platform, along with cast, directors, ratings, release year, duration, etc to provide data driven insights and actionable business recommendations to help business decide which type of shows/movies to produce and how they can grow the business in different countries.

This case study report contains the solutions to the problem statements (using Python queries), sample output of the queries, followed by insights and recommendations. As part of the confidentiality agreement, the name of the streaming platform, the actual dataset and problem statements are not included in this report.

[Google Colab Notebook-Python File](#) - This Python project involves exploratory data analysis (EDA) of a dataset from this streaming platform.

\*\*\*\*\*

1. Import the dataset in pandas and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

Shape of the dataset and Data type of all columns –

Code:

```
data = pd.read_csv("platform.csv")
```

There are a total of 8807 rows and 12 columns. The following columns – show\_id, type, title, release\_year, listed\_in and description have non-null entries. Other columns have null entries in the range of 3- 260. Except release\_year, rest of the columns have datatype as object. Release\_year has integer datatype.

```
[137] data.shape
(8807, 12)

data.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 unique values in each column

```
for i in data.columns:
    print (f"{i}: {data[i].nunique()}")
```

```
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
```

Columns such as director, cast, country, listed\_in (genre) have multiple values within each cell which are comma separated. Unnest these columns by splitting values into multiple rows

```
data["cast"] = data["cast"].str.split(", ")
data = data.explode("cast")
data["director"] = data["director"].str.split(", ")
data = data.explode("director")
data["country"] = data["country"].str.split(", ")
data = data.explode("country")
data["listed_in"] = data["listed_in"].str.split(", ")
data = data.explode("listed_in")
```

Before:

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...

After Unnesting:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Dramas
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Mysteries
1	s2	TV Show	Blood & Water	NaN	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows

Now, the number of rows has increased to 2,01,991

```
data.shape
```

(201991, 12)

Checking count and percentage of null values in each column

```
print(data.isnull().sum())
print((data.isnull().sum() / len(data)) * 100)
```

show_id	0	show_id	0.000000
type	0	type	0.000000
title	0	title	0.000000
director	2634	director	29.908028
cast	825	cast	9.367549
country	831	country	9.435676
date_added	10	date_added	0.113546
release_year	0	release_year	0.000000
rating	4	rating	0.045418
duration	3	duration	0.034064
listed_in	0	listed_in	0.000000
description	0	description	0.000000
dtype: int64		dtype: float64	

## Handling null values in country column

- Impute the country column on the basis of directors whose other movie titles had countries given.
- Check the mode of country for the director and impute in place of nulls the corresponding mode
- For directors where country column is null and cannot fetch a mode value, use 'cast' as the base column and repeat the same process

```
cd = data.groupby('director')['country'].agg(lambda x: x.mode().iloc[0] if not
x.mode().empty else None).reset_index()
cd.columns = ['director', 'mode_country']
data = data.merge(cd, how = "left", on = "director")
data["country"] = data["country"].fillna(data["mode_country"])
data.drop(columns=['mode_country'], inplace = True)

cc = data.groupby('cast')['country'].agg(lambda x: x.mode().iloc[0] if not
x.mode().empty else None).reset_index()
cc.columns = ['cast', 'mode_country']
data = data.merge(cc, how = "left", on = "cast")
data["country"] = data["country"].fillna(data["mode_country"])
data.drop(columns=['mode_country'], inplace = True)
```

Handling remaining null values - For other categorical variables with null values, update those rows as `unknown_column_name`.

```
data["director"].fillna("Unknown", inplace = True)
data["cast"].fillna("Unknown", inplace = True)
data["country"].fillna("Unknown", inplace = True)
data["duration"].fillna(0, inplace = True)
```

Now, while checking percentage of null values, we can almost all null values have been taken care of

```
print((data.isnull().sum() / len(data)) * 100)
```

```
show_id      0.000000
type         0.000000
title        0.000000
director     0.000000
cast         0.000000
country      0.000000
date_added   0.078221
release_year 0.000000
rating       0.033170
duration     0.000000
listed_in    0.000000
description  0.000000
dtype: float64
```

To move further, convert the `date_added` column in the dataset to a datetime formatted column and extract year, month, week from it

```
data['date_added'] = data['date_added'].str.strip() #to remove extra spaces
# Format the datetime to DD-MM-YYYY
data['release_date'] = pd.to_datetime(data['date_added']).dt.strftime('%Y-%m-%d')
data['release_date'] = pd.to_datetime(data['release_date'])
data["year"] = data['release_date'].dt.year.astype('Int64')
data["month"] = data['release_date'].dt.month_name()
data["month_num"] = data['release_date'].dt.month
data['week_Added'] = data['release_date'].dt.isocalendar().week
```

date_added	release_year	rating	duration	listed_in	description	release_date	year	month	month_num	week_Added
September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	2021-09-25	2021	September	9.0	38
September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...	2021-09-24	2021	September	9.0	38
September 24, 2021	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...	2021-09-24	2021	September	9.0	38

Four new columns have been added now for month, month\_number, week and year.

The duration column has 'mins' or 'season' keyword added. To be able to perform numerical analysis on duration column, create a new column to add the numerical duration

```
data["duration_modified"] = data["duration"].str.split().str[0].astype(float)
```

duration	listed_in	description	release_date	year	month	month_num	week_Added	duration_modified
90 min	Documentaries	As her father nears the end of his life, filmm...	2021-09-25	2021	September	9.0	38	90.0
2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...	2021-09-24	2021	September	9.0	38	2.0
2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...	2021-09-24	2021	September	9.0	38	2.0

Value Count of Categorical Column

```
data.groupby("type")["title"].nunique()
data.groupby("country")["title"].nunique().sort_values(ascending=False)[:5]
data.groupby("listed_in")["title"].nunique().sort_values(ascending=False)[:5]
```

country	count
United States	3854
India	1158
United Kingdom	842
Unknown	571
Canada	484

type	count
Movie	6131
TV Show	2676

genre	count
International Movies	2752
Dramas	2427
Comedies	1674
International TV Shows	1351
Documentaries	869

## Insights:

- Movies are popular than TV shows -We have 70:30 ratio of Movies and TV Shows
- US, India, United Kingdom, Canada and France are the leading countries in Content Creation, with US constituting 52% of all content.
- Of all the genres, International Movies (25%), Dramas (22%) and Comedies (15%) are the most popular

## Univariate Analysis - Distribution of Categorical Variables

```
plt.figure(figsize = (11,7)).suptitle("Distribution of Categorical Variables by  
number of movies and TV shows")

plt.subplot(2,3,1)
plt.bar(data.groupby("type")["title"].nunique().index,  
data.groupby("type")["title"].nunique(), color = "#894585")
plt.title("Distribution of Movies and TV Shows", fontsize = 10)

plt.subplot(2,3,2)
plt.bar(data.groupby("country")["title"].nunique().sort_values(ascending=False)[:10]  
.index, data.groupby("country")["title"].nunique().  
sort_values(ascending=False)[:10], color = "#69d84f")
plt.xticks(rotation = 75, fontsize=8)
plt.title("Distribution of Top 10 Countries", fontsize = 10)

plt.subplot(2,3,3)
plt.bar(data.groupby("director")["title"].nunique().sort_values(ascending=False)[1:  
11].index, data.groupby("director")["title"].nunique().  
sort_values(ascending=False)[1:11], color = "#b2996e")
plt.xticks(rotation = 75, fontsize=8)
plt.title("Distribution of Top 10 Directors", fontsize = 10)

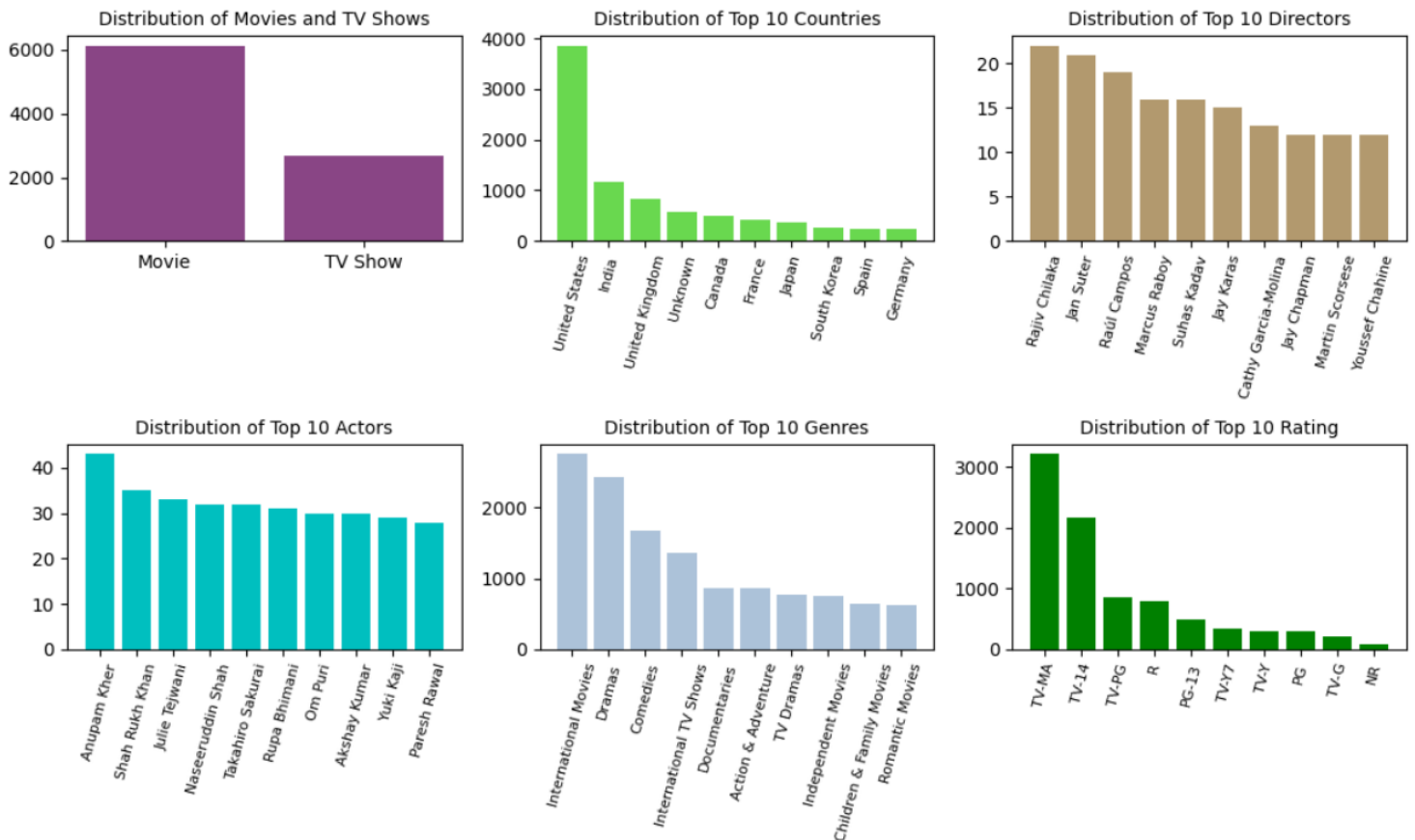
plt.subplot(2,3,4)
plt.bar(data.groupby("cast")["title"].nunique().sort_values(ascending=False)[1:11].  
index, data.groupby("cast")["title"].nunique().  
sort_values(ascending=False)[1:11], color = "c")
plt.xticks(rotation = 75, fontsize=8)
plt.title("Distribution of Top 10 Actors", fontsize = 10)

plt.subplot(2,3,5)
plt.bar(data.groupby("listed_in")["title"].nunique().sort_values(ascending=False)[:  
10].index, data.groupby("listed_in")["title"].nunique().  
sort_values(ascending=False)[:10], color = "#acc2d9")
plt.xticks(rotation = 75, fontsize=8)
plt.title("Distribution of Top 10 Genres", fontsize = 10)

plt.subplot(2,3,6)
plt.bar(data.groupby("rating")["title"].nunique().sort_values(ascending=False)[:10]  
.index, data.groupby("rating")["title"].nunique().  
sort_values(ascending=False)[:10], color = "g")
plt.xticks(rotation = 75, fontsize=8)
plt.title("Distribution of Top 10 Rating", fontsize = 10)

plt.tight_layout()
```

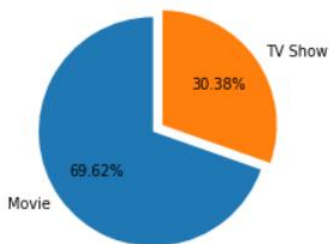
## Distribution of Categorical Variables by number of movies and TV shows



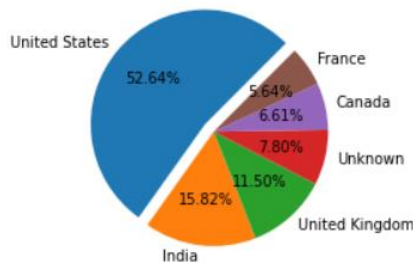
### Insights:

- Anupam Kher, SRK, Julie Tejewani, Naseeruddin Shah, and Takahiro Sakurai hold the top spots in the Most Watched content category.
- Rajiv Chilaka, Jan Suter, Raul Campos, and Suhas Kadav are among the most popular directors.
- The majority of highly rated content is aimed at mature audiences, including R-rated content, material not suitable for viewers under 14, and content requiring parental guidance.

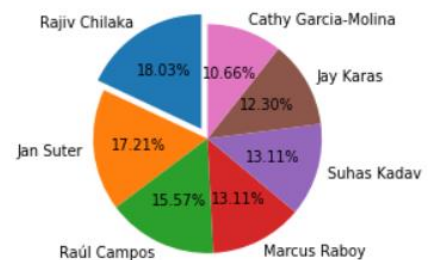
Distribution of Movies and TV Shows



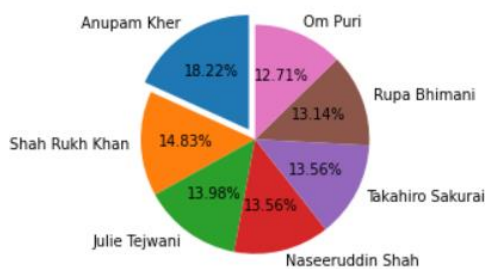
Distribution of Top Countries



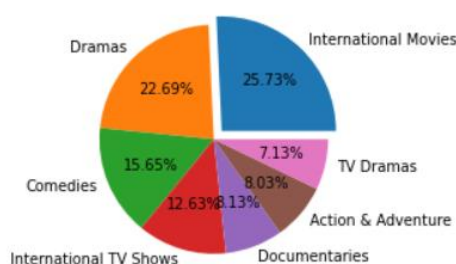
Distribution of Top Directors



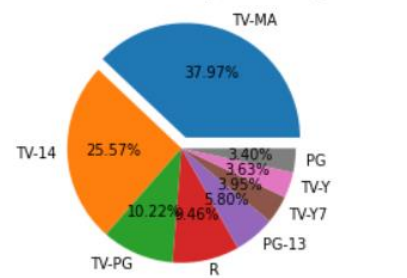
Distribution of Top Actors



Distribution of Top Genres



Distribution of Top 10 Rating



## Comparison of TV Shows vs Movies

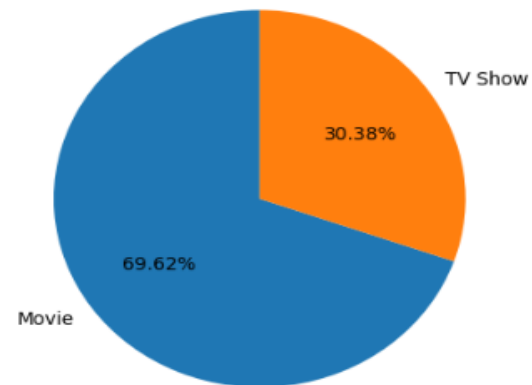
- Number of TV shows and Movies added to the platform so far and their percentage split

Code:

```
type_count = data.groupby("type")["title"].count()
```

Code Output:

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



Percentage Split using Pie Chart = `plt.pie(type_count, labels = type_count.index, startangle = 90, autopct = "%.2f%%")`

Insights: There are 6,131 Movies and 2,676 TV shows added. Movies constitute of 69.6% of total content available.

- Get the following year range available in the dataset:
  - Release\_year range of all tv shows and movies
  - release date range of all tv shows and movies

Code:

- `data.groupby("type").agg(min_year = ("release_year", "min"), max_year = ("release_year", "max"))`
- `data.groupby("type").agg(min_year = ("release_date", "min"), max_year = ("release_date", "max"))`

Output

(i)

	min_year	max_year
type		
Movie	1942	2021
TV Show	1925	2021

(ii)

	min_year	max_year
type		
Movie	2008-01-01	2021-09-25
TV Show	2008-02-04	2021-09-24

Insights: All movies were released between 1942 and 2021 and TV shows between 1925 and 2021. They were released to the platform between 2008 and 2021. The platform started releasing Movies and TV shows almost around the same time. Still, Movies constitute of 69.6% of total content available.

- Does this streaming platform have more focus on TV Shows than movies in recent years?

```
data.groupby("year")["type"].value_counts().sort_index(ascending=False)[:10]
```



Insights: In the last 5 years, from 2021-2017, there were more movies added to the platform than TV shows. In 2021, there were 993 movies and 505 TV shows added, means 488 more movies than TV shows. In 2020, there were 689 more movies than TV shows added. In 2019, it was 832 more movies than TV shows. The number of movies added to the platform has drastically reduced from 2019 to 2021 – from 1424 to 993. But so has number of TV shows – from 592 in 2019 to 505 in 2021. Hence, we can say the platform still focuses more on movies than TV shows.

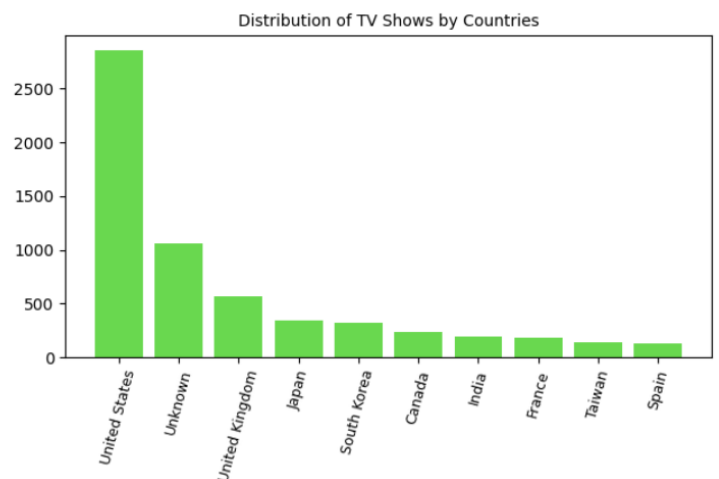
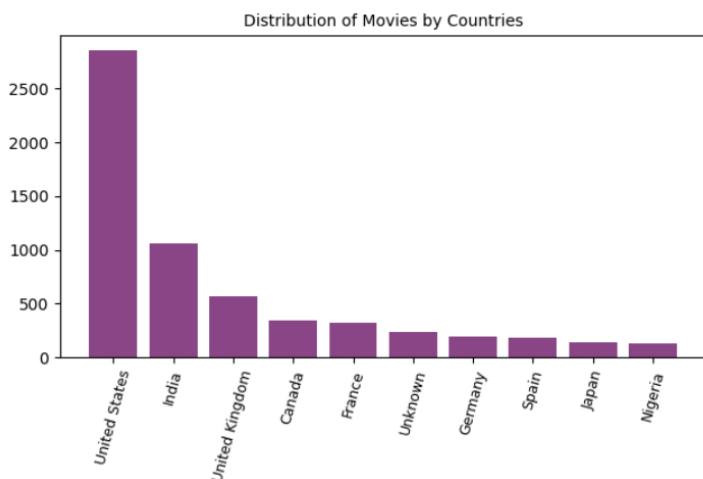
year	type	
2021.0	TV Show	505
	Movie	993
2020.0	TV Show	595
	Movie	1284
2019.0	TV Show	592
	Movie	1424
2018.0	TV Show	412
	Movie	1237
2017.0	TV Show	349
	Movie	839
Name: count, dtype: int64		

### Distribution of movies and TV shows by Countries

```
plt.subplot(2,3,1)
plt.bar(data[data["type"] ==
"Movie"].groupby("country")["title"].unique().sort_values(ascending=False).head(10)
.index, data[data["type"] ==
"Movie"].groupby("country")["title"].unique().sort_values(ascending=False).head(10)
), color = "#894585")
plt.xticks(rotation = 75, fontsize=8)
plt.title("Distribution of Movies by Countries", fontsize = 10)

plt.subplot(2,3,2)
plt.bar(data[data["type"] == "TV
Show"].groupby("country")["title"].unique().sort_values(ascending=False).head(10).
index, data[data["type"] ==
"Movie"].groupby("country")["title"].unique().sort_values(ascending=False).head(10)
), color = "#69d84f")
plt.xticks(rotation = 75, fontsize=8)
plt.title("Distribution of TV Shows by Countries", fontsize = 10)
```

### Distribution of movies and TV shows by Countries



### Insights:

- The United States is the largest producer of both movies and TV shows. It has more than double the number of productions compared to the next leading country in both categories.
- India ranks second in movie production, significantly ahead of other countries like the United Kingdom and Canada. However, India is much less prominent in the production of TV shows.



- Countries like the United Kingdom, Japan, and South Korea also have notable representation in TV show production.

What time of the year is the best time to launch a TV show?

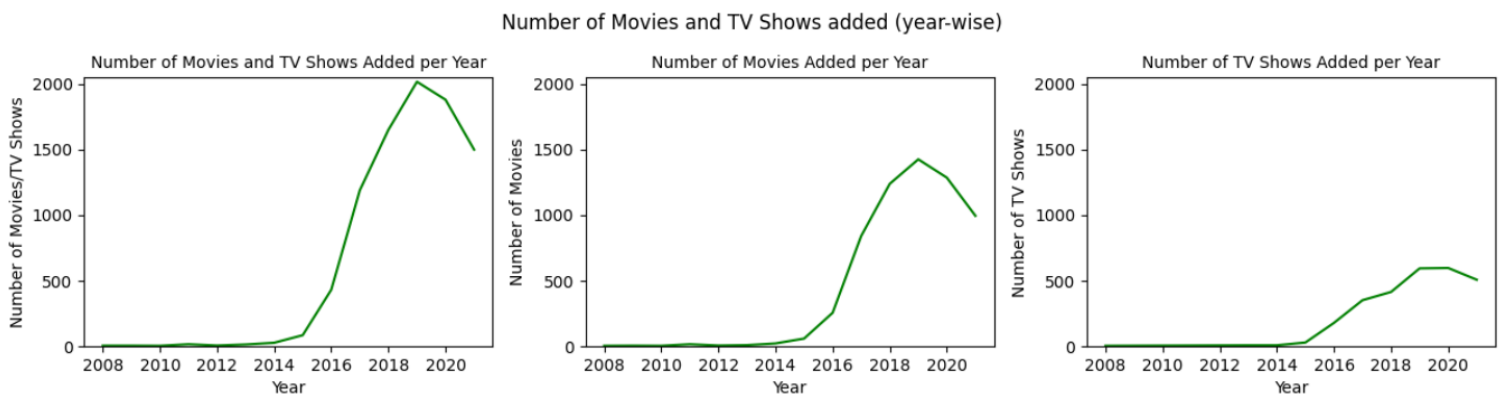
Let's get year-wise, month-wise and week-wise analysis of this

Year-wise:

```
plt.subplot(2,3,1)
data_year = data.groupby("year")["title"].nunique().reset_index()
sns.lineplot(data = data_year, x = "year", y = "title", color = 'g')
plt.xlabel("Year")
plt.ylabel("Number of Movies/TV Shows")
plt.title("Number of Movies and TV Shows Added per Year", fontsize = 10)
plt.ylim(0, 2050)

plt.subplot(2,3,2)
data_movies = data[data["type"] == "Movie"].groupby("year")["title"].nunique().reset_index()
# data_year = data.groupby("year")["title"].nunique().reset_index()
sns.lineplot(data = data_movies, x = "year", y = "title", color = 'g')
plt.xlabel("Year")
plt.ylabel("Number of Movies")
plt.title("Number of Movies Added per Year", fontsize = 10)
plt.ylim(0, 2050)

plt.subplot(2,3,3)
data_shows = data[data["type"] == "TV Show"].groupby("year")["title"].nunique().reset_index()
# data_year = data.groupby("year")["title"].nunique().reset_index()
sns.lineplot(data = data_shows, x = "year", y = "title", color = 'g')
plt.xlabel("Year")
plt.ylabel("Number of TV Shows")
plt.title("Number of TV Shows Added per Year", fontsize = 10)
plt.ylim(0, 2050)
```



Insights:

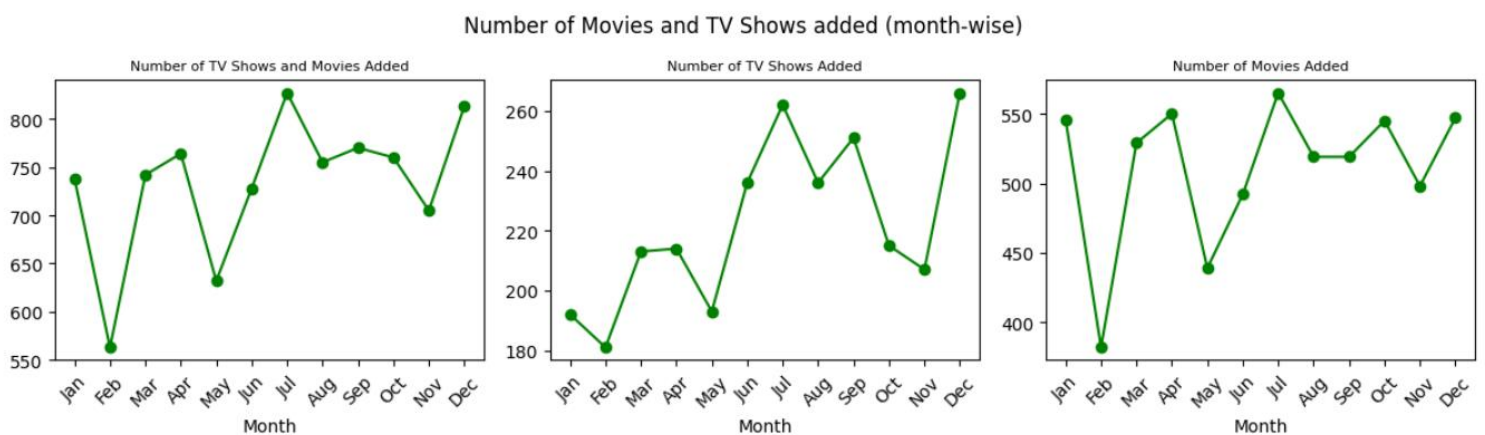
- There was a sharp increase in the number of movies and TV shows added from 2015 to 2019, indicating a period of rapid content expansion.
- The highest number of movies and TV shows were added in 2018-2019, marking the peak of content production.
- Decline After 2019: There is a noticeable decline in the number of movies and TV shows added after 2019, likely influenced by external factors such as the COVID-19 pandemic.

## Month-wise

```
plt.subplot(2,3,1)
monthly = data.groupby("month_num")["title"].nunique()
plt.plot(monthly.index, monthly.values, marker = 'o', color = 'g')
plt.xlabel('Month')
plt.title('Number of TV Shows and Movies Added', fontsize = 8)
plt.xticks(rotation = 45, ticks=range(1, 13), labels=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])

plt.subplot(2,3,2)
monthly_S = data[data["type"] == "TV Show"].groupby("month_num")["title"].nunique()
plt.plot(monthly_S.index, monthly_S.values, marker = 'o', color = 'g')
plt.xlabel('Month')
plt.title('Number of TV Shows Added', fontsize = 8)
plt.xticks(rotation = 45, ticks=range(1, 13), labels=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])

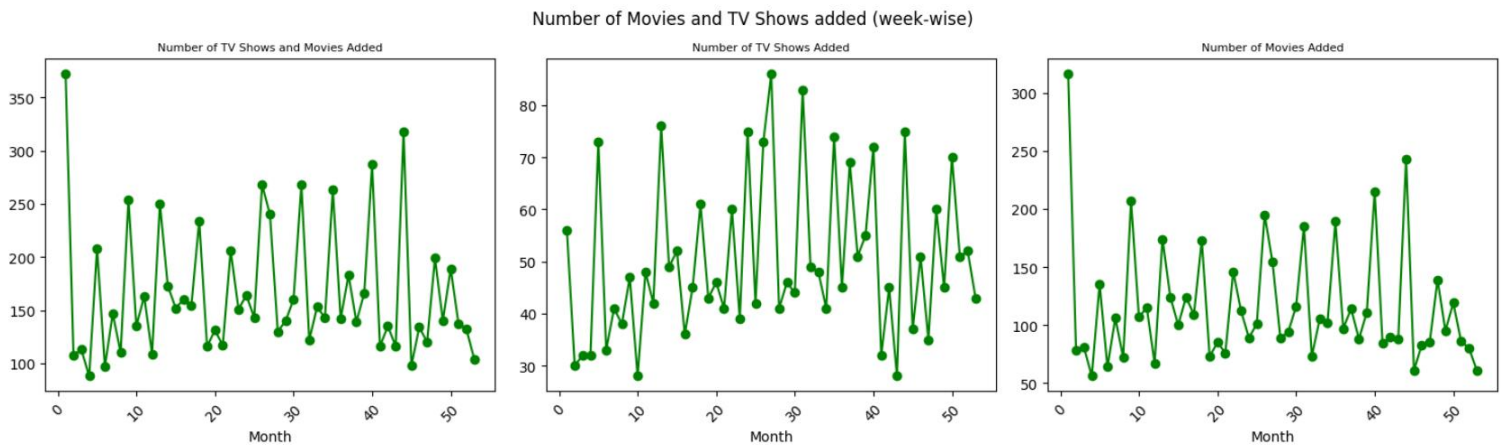
plt.subplot(2,3,3)
monthly_M = data[data["type"] == "Movie"].groupby("month_num")["title"].nunique()
plt.plot(monthly_M.index, monthly_M.values, marker = 'o', color = 'g')
plt.xlabel('Month')
plt.title('Number of Movies Added', fontsize = 8)
plt.xticks(rotation = 45, ticks=range(1, 13), labels=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
```



### Insights:

- Most of the content is added in the month of December, January (first and last months) and July
- December and January coincide with the holiday season, including Christmas and New Year's celebrations. During this period, people typically have more free time, are on vacation. This leads to an increase in streaming activity as people look for entertainment options. The platform may also aim to boost its year-end performance metrics by releasing a significant amount of content in December.
- July is part of the summer vacation period for many people, including school and university students. This time off leads to increased leisure time and higher demand for entertainment.

## Week-wise



### Insights:

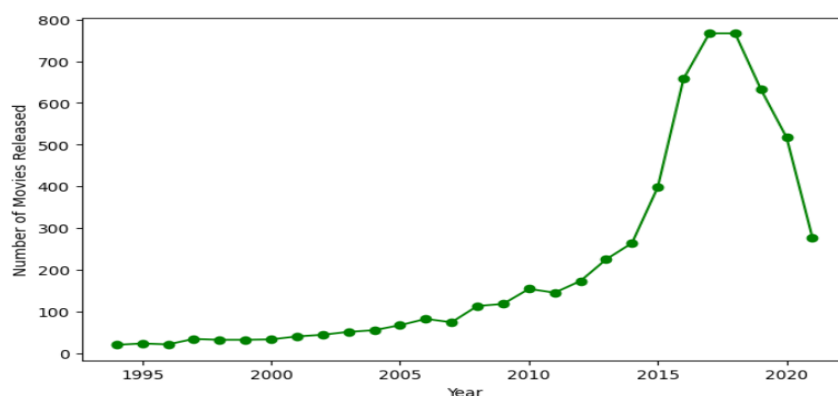
- Most of the Content is added in the first week of the year, reinstating what we observed for first and last months of the year

How has the number of movies released per year changed over the last 20-30 years?

```
from datetime import datetime
current_year = datetime.now().year
threshold_year = current_year - 30

last_30_years = data[data['release_year'] >= threshold_year]
yearly_trend = last_30_years[last_30_years["type"] ==
"Movie"].groupby("release_year")["title"].count()

plt.figure(figsize=(8, 5))
plt.plot(yearly_trend.index, yearly_trend.values, marker='o', color =
'g')
plt.xlabel('Year')
plt.ylabel('Number of Movies Released')
plt.show()
```



Insights: There was a slow but steady increase in the number of movies released per year from 1995 to around 2010. There is a significant and rapid increase in the number of movies released per year from 2010 to around 2018. Technological advancements and the rise of streaming platforms like this one, which started producing and acquiring a large volume of content contributed to this trend. The sharp drop in the number of movies released in the last few years

since 2019 likely reflects the impact of the COVID-19, which caused production halts, and delays.

### Analysis of actors/directors of different types of shows/movies

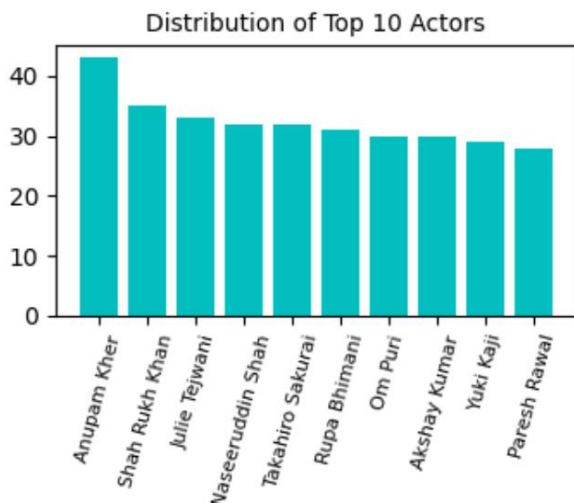
```
actor_director = data[(data["cast"] != "Unknown") & (data["director"] != "Unknown")]
actor_director.groupby(["cast", "director"])["title"].nunique().sort_values(ascending=False)
```

		title
cast	director	
Rajesh Kava	Rajiv Chilaka	19
Julie Tejjwani	Rajiv Chilaka	19
Rupa Bhimani	Rajiv Chilaka	18
Jigna Bhardwaj	Rajiv Chilaka	18
Vatsal Dubey	Rajiv Chilaka	16
...	...	...
Hari Kondabolu	Bobcat Goldthwait	1
Hari Shivdasani	Raj Kapoor	1
Hariata Moriarty	Ainsley Gardiner	1
	Briar Grace-Smith	1
Şopê Dirisû	Remi Weekes	1

Insights: The Top cast - director combination is Rajesh Kava and Rajiv Chilaka, majorly for the Indian cartoon series - Chhota Bheem

### Distribution of Top 10 Actors

```
data.groupby("cast")["title"].nunique().sort_values(ascending=False)[1:11]
```



Insights: Top 5 casts are Anupam Kher, Shah Rukh Khan, Julie Tejjwani, Naseeruddin Shah and Rakahiro Sakurai – majorly dominated by Indian actors. suggesting this platform's recognition of their cultural influence and the appeal of their content to global audiences. Their frequent appearances may reflect high viewer engagement and interest in their work, prompting the platform to feature more of their productions to cater to audience preference.

## Analysis of Duration

```
data["duration"] = data["duration"].str.split().str[0].astype(float)
data.groupby("type")["duration"].mean()
```

Insights: Movies have an average duration of 99 mins and TV shows have an average number of 1.7 seasons

```
type
Movie      99.577187
TV Show    1.764948
Name: duration, dtype: float64
```

## Outlier Checks:

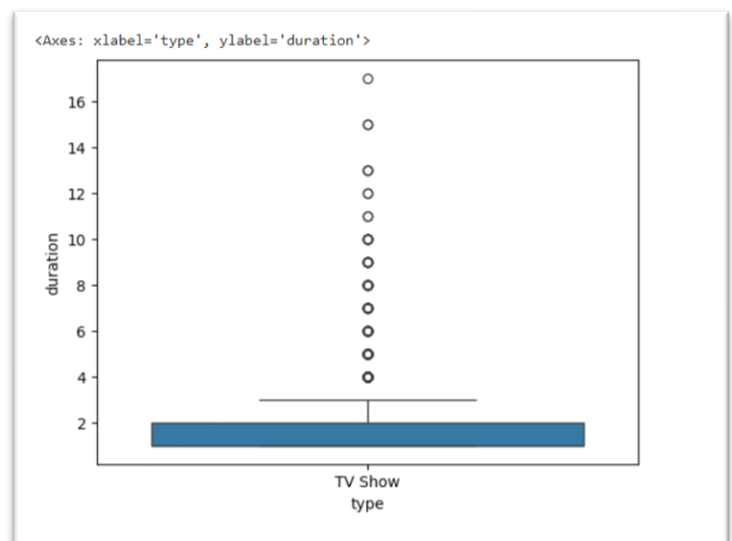
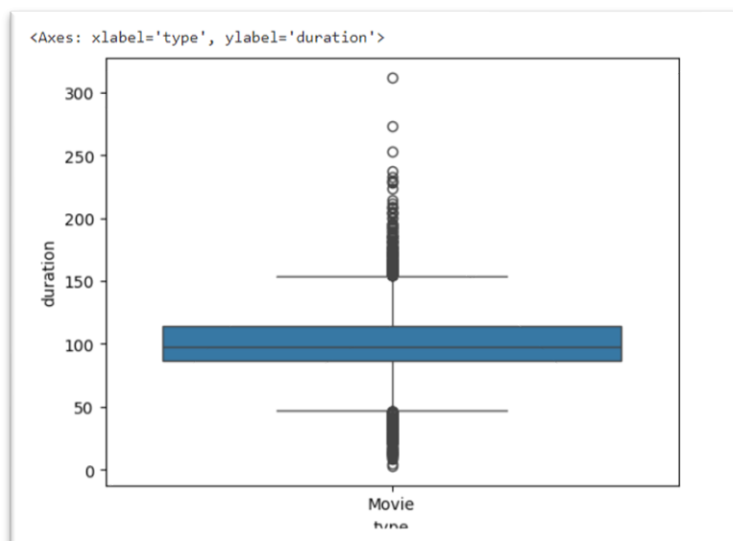
- Average duration (in mins) for movies and average number of seasons for TV shows and create a boxplot for outlier checks

### (i) Movie\_plot:

```
Movie_plot = data[data["type"] == "Movie"]
sns.boxplot(data = Movie_plot, x = "type", y = "duration")
```

### (ii) TV Show\_plot:

```
TVSHOW_plot = data[data["type"] == "TV Show"]
sns.boxplot(data = TVSHOW_plot, x = "type", y = "duration")
```



Insights: The median TV show duration is 2 seasons, indicating that most TV shows have relatively short runs. There are numerous outliers in TV show durations. These outliers represent TV shows with durations significantly longer than the typical range. The outliers extend up to 16 seasons. These long-running shows could be popular series with multiple seasons, possibly due to high viewership and demand. For the platform, understanding the typical duration and the presence of outliers can help in content planning. It can inform decisions about the expected lifecycle of new shows and the allocation of resources for long-running series.

The median duration of movies is approximately 100 minutes. The middle 50% of movie durations range from about 80 to 120 minutes.

## Distribution of movies and TV shows by Duration

```
duration_shows = data[data["type"] == "TV
Show"].groupby("duration")["title"].nunique().reset_index().sort_values(by =
"title", ascending=False)[:15]
duration_movies = data[data["type"] ==
"Movie"].groupby("duration")["title"].nunique().reset_index().sort_values(by =
"title", ascending=False)[:15]

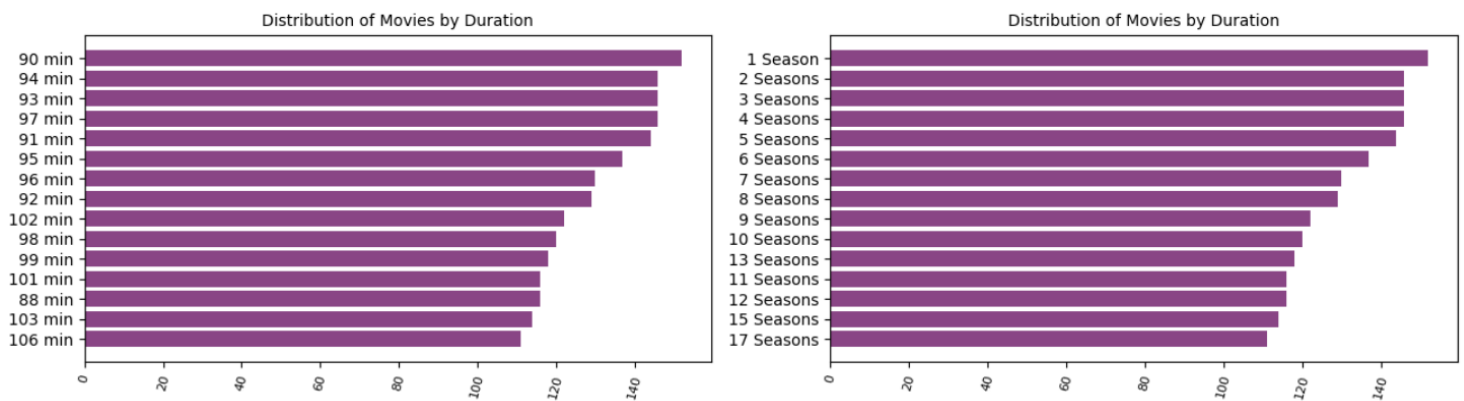
plt.figure(figsize = (13,7)).suptitle("Distribution of movies and TV shows by
Duration")

plt.subplot(2,3,1)
plt.barh(duration_movies[::-1]["duration"], duration_movies[::-1]["title"], color =
"#894585")
plt.xticks(rotation = 75, fontsize=8)
plt.title("Distribution of Movies by Duration", fontsize = 10)

plt.subplot(2,3,2)
plt.barh(duration_shows[::-1]["duration"], duration_shows[::-1]["title"], color =
"#894585")
plt.xticks(rotation = 75, fontsize=8)
plt.title("Distribution of Movies by Duration", fontsize = 10)

plt.tight_layout()
```

Distribution of movies and TV shows by Duration



### Insights:

- For TV shows, single-season series are the most common, and the number of shows decreases as the number of seasons increases. When it comes to movies, those with a duration of 80-100 minutes are the most popular.

## Business Recommendations:

1. Given the popularity of movies (70:30 ratio with TV shows) and the platform's focus on movies, allocate more resources to acquiring and promoting a diverse range of films. Highlight movies of different durations (particularly 80-100 minutes, which are the most popular) and international content to cater to a broad audience base. It offers a one-time, shorter engagement, which can appeal to viewers looking for a quick entertainment fix. Given the potential for long-term viewer engagement, the platform should focus on producing high-quality TV shows (even if the number of shows are limited) that have the potential to become long-running series. This can build a dedicated audience base.
2. With actors like Anupam Kher and Shah Rukh Khan topping the list of casts, there's a clear interest in Indian cinema. Promote their existing content and secure more of their upcoming releases, as they are likely to drive higher viewer engagement.
3. Directors such as **Rajiv Chilaka**, **Jan Suter**, **Raul Campos**, and **Suhas Kadav** are behind highly popular content. Collaborate with these directors for exclusive releases, and emphasize their work in platform-wide promotions to attract their fan base.
4. As most highly rated content is aimed at **Mature Audiences** (R-rated, parental guidance, and not intended for viewers under 14), expand the portfolio of similar content. Consider producing or acquiring more mature-themed dramas, thrillers, and documentaries to capture this segment further.
5. Given the increase in streaming activity during **December-January** (holiday season) and **July** (summer vacation), plan major content releases, exclusive premieres, and promotional campaigns around these periods to maximize viewership and subscription sign-ups.
6. Since India is a significant producer of movies but has less prominence in TV show production, invest in creating more local, high-quality TV series aimed at Indian audiences. This could help diversify the platform's offerings and tap into the potential growth of the TV show market in India.
7. Since single-season TV shows are the most common, create more content that aligns with this format. Single-season series are easier for new users to commit to, and by focusing on this, the platform can cater to binge-watchers looking for fresh, complete storylines.