```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
import seaborn as sns
```

```
! gdown '1Egl_DGub6Y75G0sOm562oJt6pN_oL3uzqtWg365Ui5Q'
```

```
Downloading...
From (original): https://drive.google.com/uc?id=1Egl_DGub6Y75G0sOm562oJt6pN_oL3uzqtWg365Ui5Q
From (redirected): https://docs.google.com/spreadsheets/d/1Egl_DGub6Y75G0sOm562oJt6pN_oL3uzqtWg365Ui5Q/export?format=xlsx
To: /content/netflix.xlsx
1.74MB [00:00, 127MB/s]
```

```
data = pd.read_excel('/content/netflix.xlsx')
```

```
data
```

	show_id	type	title	director	cast	country	date_added	release_ye
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	20
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	20
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	20
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	20
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	20
...	...	...	...	...	...	...	...	
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert	United States	2019-11-20	20

```
data.shape
```

```
(8807, 12)
```

Insights :-

There are 8807 rows of data containing 12 different attributes of data stored in columns.These are as follows :- show_id, type, title, director, cast, country, date_added, release_year, rating, duration, listed_in, description.

Below we have checked the total number of NaN values in mentioned particular columns.

```
data['show_id'].isna().value_counts()
```

```
show_id
False    8807
Name: count, dtype: int64
```

```
# Unique values of show_id
data['show_id'].nunique()
```

```
8807
```

Insights :- There are no missing elements in the column 'show_id' and all the elements are unique.

```
data['type'].isna().value_counts()
```

```
type
False    8807
Name: count, dtype: int64
```

```
# Unique values of type
data['type'].nunique()
```

```
2
```

Insights :- There are no missing elements in the column 'type' with only 2 elements are present in the column:- Movie and TV show.

```
data['title'].isna().value_counts()
```

```
title
False    8807
Name: count, dtype: int64
```

```
# Unique values of title
data['title'].nunique()
```

```
8804
```

Insights :- There are no missing elements in the column 'title' and 8804 elements are unique.

```
data['director'].isna().value_counts()
```

```
director
False    6173
True     2634
Name: count, dtype: int64
```

```
# Unique values of director
data['director'].nunique()
```

```
4528
```

Insights :- There are 2634 missing elements in the column 'director' and 4528 elements are unique.

```
data['cast'].isna().value_counts()
```

```
cast
False    7982
True      825
Name: count, dtype: int64
```

```
# Unique values of cast
data['cast'].nunique()
```

```
7692
```

Insights :- There are 825 missing elements in the column 'cast' and 7692 elements are unique.

```
data['country'].isna().value_counts()
```

```
country
False    7976
True      831
Name: count, dtype: int64
```

```
# Unique values of cast
data['country'].nunique()
```

```
748
```

Insights :- There are 831 missing elements in the column 'country' and 748 elements are unique.

```
data['date_added'].isna().value_counts()
```

```
date_added
False    8797
True       10
Name: count, dtype: int64
```

```
# Unique values of date_added
data['date_added'].nunique()
```

```
1714
```

Insights :- There are 10 missing elements in the column 'date_added' and 1714 elements are unique.

```
data['release_year'].isna().value_counts()
```

```
release_year
False    8807
Name: count, dtype: int64
```

```
# Unique values of release_year
data['release_year'].nunique()
```

```
74
```

Insights :- There are no missing elements in the column 'release_year' and 74 elements are unique.

```
data['rating'].isna().value_counts()
```

```
rating
False    8803
True        4
Name: count, dtype: int64
```

```
# Unique values of rating
data['rating'].nunique()
```

```
17
```

Insights :- There are 4 missing elements in the column 'rating' and 17 elements are unique.

```
data['duration'].isna().value_counts()
```

```
duration
False    8804
True        3
Name: count, dtype: int64
```

```
# Unique values of duration
data['duration'].nunique()
```

```
220
```

Insights :- There are 3 missing elements in the column 'duration' and 220 elements are unique.

```
data['listed_in'].isna().value_counts()
```

```
listed_in
False    8807
Name: count, dtype: int64
```

```
# Unique values of listed_in
data['listed_in'].nunique()
```

```
514
```

Insights :- There are no missing elements in the column 'listed_in' and 514 elements are unique.

```
data['description'].isna().value_counts()
```

```
description
False    8807
Name: count, dtype: int64
```

Insights :- There are no missing elements in the column 'description'.

Individual rows for director,cast and country

```
# Filling the NaN values of the required columns
data.cast.fillna('NA',inplace = True)
data.director.fillna('NA',inplace = True)
data.country.fillna('NA',inplace = True)
data.date_added.fillna('NA',inplace = True)
data.release_year.fillna('NA',inplace = True)
data.rating.fillna('NA',inplace = True)
data.duration.fillna('0 ',inplace = True)

# Coverting the date_added column in the datetime format
data['date_added'] = pd.to_datetime(data['date_added'], errors='coerce')
```

```
# creation of individual rows for directors,cast and country for granular view
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['genre'] = data['listed_in'].str.split(', ')
data = data.explode('listed_in')
```

```
# Creating new columns from the durations column first for movies
movie_data = data[data['type'] == 'Movie']
movie_data = movie_data.reset_index(drop=True)


data['movie_mins'] = movie_data['duration'].apply(lambda x: int(x.split(' ')[0]) if isinstance(x, str) and x != 'NA' else '0 ')

# Similar approach for TV show data
tv_show_data = data[data['type'] == 'TV Show']
tv_show_data = tv_show_data.reset_index(drop=True)


data['number_of_seasons'] = tv_show_data['duration'].apply(lambda x: int(x.split(' ')[0]) if isinstance(x, str) and x != 'NA' else '0 '
```

```
data
```

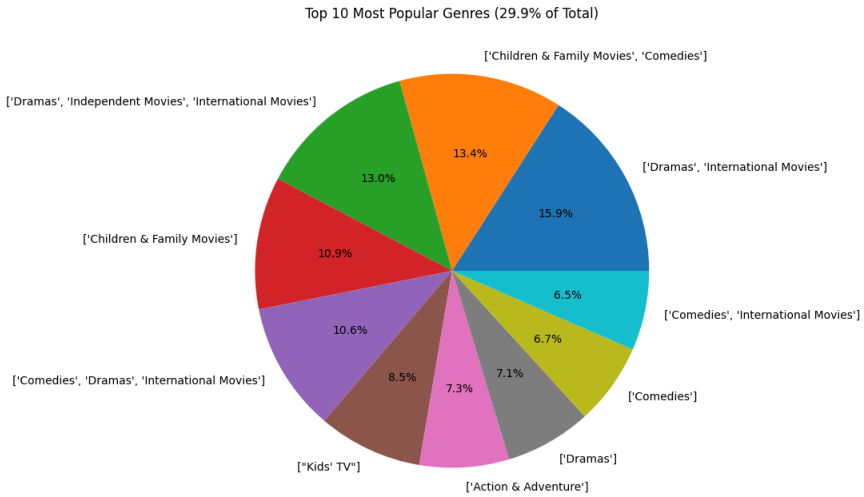
	show_id	type	title	director	cast	country	date_added	release_y
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NA	United States	2021-09-25	2
1	s2	TV Show	Blood & Water	NA	Ama Qamata	South Africa	2021-09-24	2
1	s2	TV Show	Blood & Water	NA	Khosi Ngema	South Africa	2021-09-24	2
1	s2	TV Show	Blood & Water	NA	Gail Mabalane	South Africa	2021-09-24	2
1	s2	TV Show	Blood & Water	NA	Thabang Molaba	South Africa	2021-09-24	2
...	...	...	...	...	...	...	...	
8806	s8807	Movie	Zubaan	Mozez Singh	Manish Chaudhary	India	2019-03-02	2
8806	s8807	Movie	Zubaan	Mozez Singh	Meghna Malik	India	2019-03-02	2
8806	s8807	Movie	Zubaan	Mozez Singh	Malkeet Rauni	India	2019-03-02	2
8806	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	2019-03-02	2

```
top_genres = data['genre'].value_counts()
top_genres
```

```
genre
[Dramas, International Movies]                              4255
[Children & Family Movies, Comedies]                        3578
[Dramas, Independent Movies, International Movies]          3465
[Children & Family Movies]                                  2912
[Comedies, Dramas, International Movies]                    2841
                                                            ...
[Crime TV Shows, International TV Shows, Reality TV]           1
[Docuseries, Reality TV, Teen TV Shows]                        1
[Reality TV, Science & Nature TV, TV Action & Adventure]       1
[Docuseries, Science & Nature TV, TV Comedies]                 1
[British TV Shows, Docuseries, Reality TV]                     1
Name: count, Length: 514, dtype: int64
```

Insights :- We can see that these content with Top genres are mostly being added on the platform, this might be because of the popularity of the content amongst consumers.

```
# Exhibit 1 :-

# Plotting a pie chart for top 10 genre and their respective overall share

# Get the top 10 most popular genres
top_genres = data['genre'].value_counts().head(10)

# Create a new DataFrame for the top 10 genres
top_genres_df = pd.DataFrame({
    'Genre': top_genres.index,
    'Count': top_genres.values
})

# Calculate the overall share of the top 10 genres
total_share = top_genres_df['Count'].sum() / data['genre'].value_counts().sum() * 100

# Create a pie chart
plt.figure(figsize=(12, 8))
plt.pie(top_genres_df['Count'], labels=top_genres_df['Genre'], autopct="%1.1f%%")
plt.title('Top 10 Most Popular Genres ({}% of Total)'.format(round(total_share, 1)))
plt.show()
```



Insights :- Dramas,International movies holds the most share of the content in Nextflix 15.9 % , followed by Children and family movies, Comedies with 13.4% and followed by Dramas,Independent movies, International movies with 13.0% share and so on.

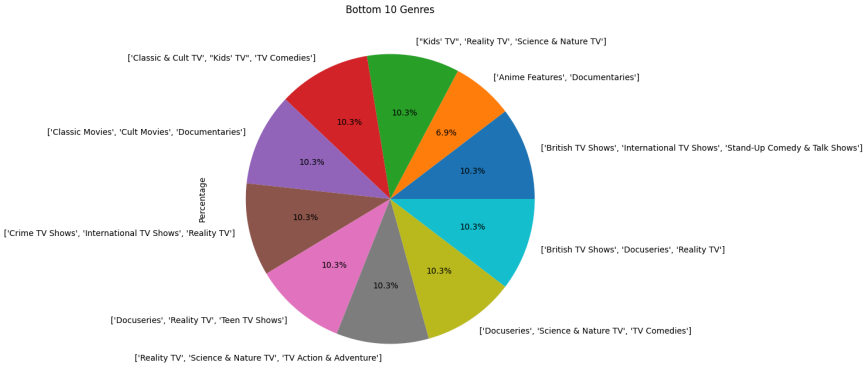
Recommendations :- These genres are most present on the platform, we need to promote these content on Netflix as it will mostly be watched on the platform.

```
bottom_genres = data['genre'].value_counts().tail(10)
bottom_genres
```

```
genre
[British TV Shows, International TV Shows, Stand-Up Comedy & Talk Shows]    1
[Anime Features, Documentaries]                                             1
[Kids' TV, Reality TV, Science & Nature TV]                                 1
[Classic & Cult TV, Kids' TV, TV Comedies]                                  1
[Classic Movies, Cult Movies, Documentaries]                                1
[Crime TV Shows, International TV Shows, Reality TV]                        1
[Docuseries, Reality TV, Teen TV Shows]                                     1
[Reality TV, Science & Nature TV, TV Action & Adventure]                    1
[Docuseries, Science & Nature TV, TV Comedies]                              1
[British TV Shows, Docuseries, Reality TV]                                  1
Name: count, dtype: int64
```

Insights :- We can see that these content with bottom genres are mostly being added only once on the platform, this might be because of the unpopularity of the content amongst consumers.

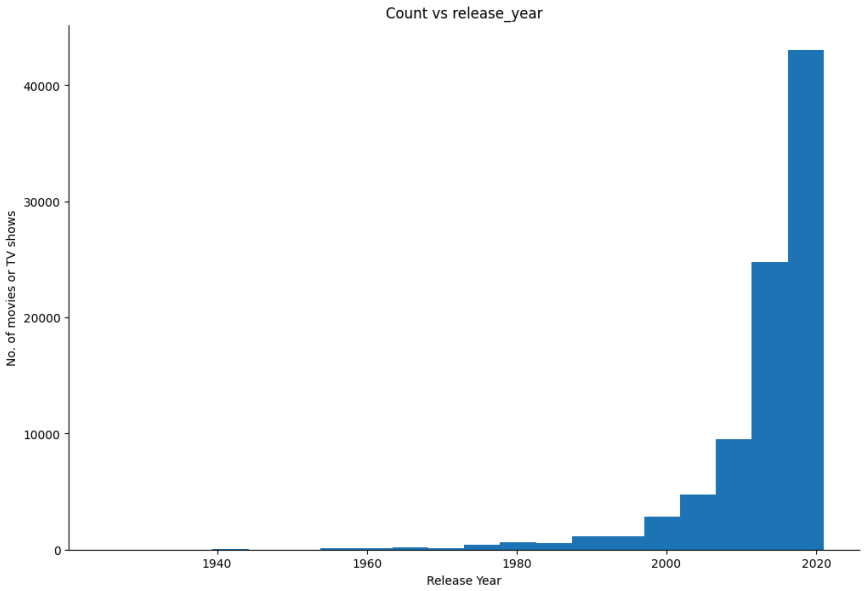
```
# Exhibit 2 :-

# Plotting a pie chart for bottom 10 genre and their respective overall share

# Calculating the total genres
total_genres = data['genre'].explode().nunique()

# Get the bottom 10 genres
bottom_genres = data['genre'].value_counts().tail(10)

# Calculate the percentage of each genre
bottom_genres_df = bottom_genres.to_frame().reset_index()
bottom_genres_df['Percentage'] = [len(genre) / total_genres * 100 for genre in bottom_genres_df['genre']]

# Create a figure and axes
fig, ax = plt.subplots(figsize=(12, 8))

# Plot the pie chart
bottom_genres_df['Percentage'].plot.pie(ax=ax, autopct='%1.1f%%', labels=bottom_genres_df['genre'])

# Set the title and labels
ax.set_title('Bottom 10 Genres')

# Display the plot
plt.show()
```



Insights :- Anime features, Documentries accounts to 6.9% of the total bottom gemres, followed by British TV Shows, International TV Shows, Stand-Up Comedy & Talk Shows with 10.3% and so on.

Recommendation :- These mentioned categories are least present on the platform and most of them are only used once. We can say that people are not interested in these genres and we can ignore this type of content as there will be less watching on this content.

```
# Exhibit 3 :-

# Count of movies or TV shows against release_year.

plt.figure(figsize=(12,8))
data['release_year'].plot(kind='hist', bins=20, title='Count vs release_year')
plt.gca().spines[['top', 'right',]].set_visible(False)
plt.ylabel('No. of movies or TV shows')
plt.xlabel('Release Year')
plt.show()
```



Insights :- We can see that the count of the content whether it may be Movies or TV shows, it has increased over the years. We can see two spikes. First one around 2010, it might be because of the popularity of internet wass growing and re-built model of the netflix has helped, second one was during 2019 after the lockdown, the streaming of the content has drastically increased during it.

Recommendations :- We need to check the increase more content of the genres which are liked by the consumers and also have to create some experimental genres to check whether the consumers will like it or not. In this way we can seggregate the content and increase the streaming on platform.

```
data.dtypes
```

```
show_id                    object
type                       object
title                      object
director                   object
cast                       object
country                    object
date_added          datetime64[ns]
release_year                 int64
rating                     object
duration                   object
listed_in                  object
description                object
genre                      object
movie_mins                  int64
number_of_seasons           int64
dtype: object
```

Insights :- We have the data in 3 data types :-

1. Object
2. datetime
3. int

```
data['country'].value_counts().index
```

```
Index(['United States', 'India', 'United Kingdom', 'NA', 'Canada', 'Japan',
       'France', 'Spain', 'Germany', 'South Korea',
       ...
       'Nicaragua', 'Vatican City', 'Kazakhstan', 'Sri Lanka', 'Afghanistan',
       'Mongolia', 'Armenia', 'Panama', 'Uganda', 'Palestine'],
      dtype='object', name='country', length=128)
```

Insights :- These are the list of the countries.

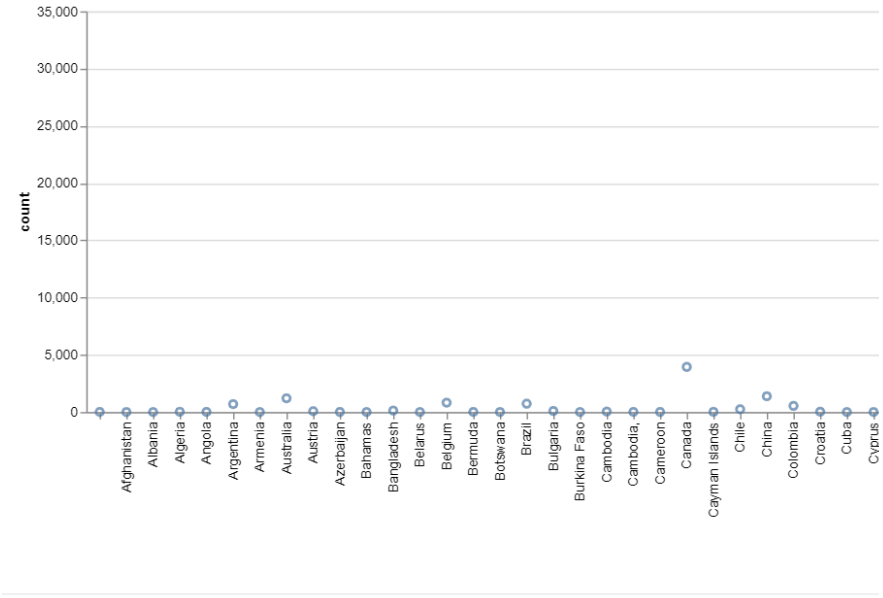
```
grouped_country =data.groupby('country')['genre'].agg(['count']).reset_index()
grouped_country
```

	country	count
0		12
1	Afghanistan	1
2	Albania	4
3	Algeria	29
4	Angola	16
...	...	...
123	Vatican City	1
124	Venezuela	12
125	Vietnam	50
126	West Germany	43
127	Zimbabwe	15

128 rows × 2 columns

Insights :- This is the Series with the count of the genres present in every country.

```
# Exhibit 4 :-

# This scatter plot represent the count of different genre in every country

import altair as alt
chart = alt.Chart(grouped_country).mark_point().encode(x='country', y='count')
chart
```

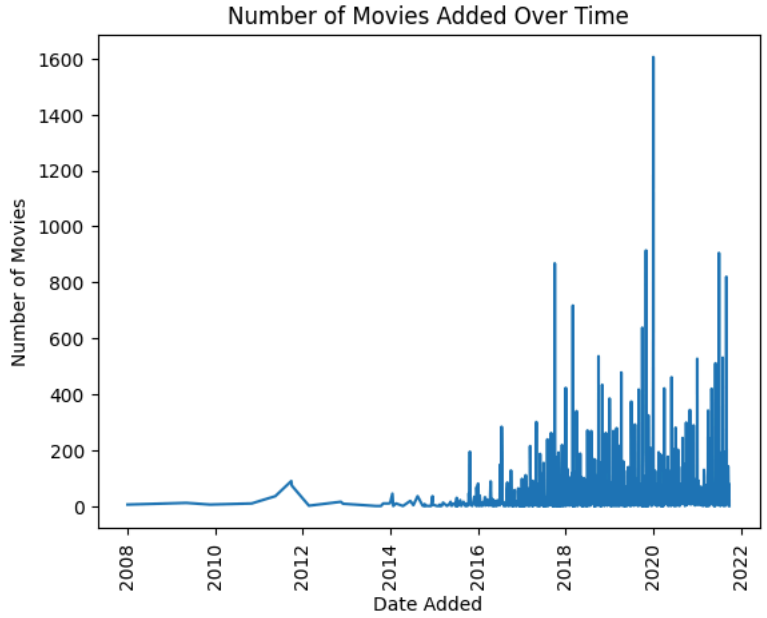


Insights :- United states has most number of genres of the content across TV shows and movies and are followed by India, United Kingdom, NA,Japan, Canada, etc.

Recommendation :- We need to increase the genres accross the Countries like India, Canada and Japan. And also we need to try to increase the content with the genres which are local to the countries in those particular countries in order to increase the business.

```
# Filter movies only
movies = data[data["type"] == "Movie"]

# Group movies by date_added and count
movie_count_by_date = movies.groupby("date_added").size()
```

```
# Exhibit 5 :-

# Lineplot to show the number of movies added per year
plt.plot(movie_count_by_date.index, movie_count_by_date.values)
plt.xlabel("Date Added")
plt.ylabel("Number of Movies")
plt.title("Number of Movies Added Over Time")
plt.xticks(rotation="vertical")
plt.figure(figsize=(12,8))
plt.show()
```



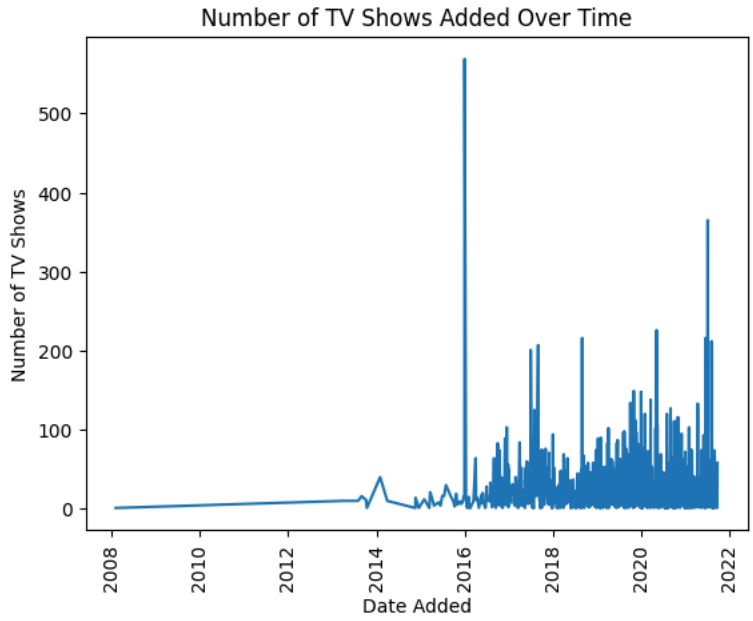

```
<Figure size 1200x800 with 0 Axes>
```

Insights :- We can see that there is a huge spike around the 2019-2020 period because of the Covid-19, the streaming of the Movies were on the rise and hence there was a increasing demand on the new movies in this time frame.

Recommendations :- We need to increase the genres and increase more of the local genres contents, so that the consumers can stream the movies.

```
# Filter TV shows only
TV_show = data[data["type"] == "TV Show"]

# Group movies by date_added and count
TV_show_count_by_date = TV_show.groupby("date_added").size()
```

```
# Exhibit 6 :-

# Lineplot to show the number of TV Shows added per year
plt.plot(TV_show_count_by_date.index, TV_show_count_by_date.values)
plt.xlabel("Date Added")
plt.ylabel("Number of TV Shows")
plt.title("Number of TV Shows Added Over Time")
plt.xticks(rotation="vertical")
plt.figure(figsize=(12,8))
plt.show()
```




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

Insights :- We see a huge spike around 2016, it might be because of the Netflix started creating Netflix originals TV shows, which were the center of attraction as it was more of decriptive thing other than Movies and they were adding more newer genres other than the repitative genres in the movies.

Recommendations:- We need to create more Netflix originals and we need to add the TV shows in the cycle of time across the year as the consumer should get apt time to complete the TV series.

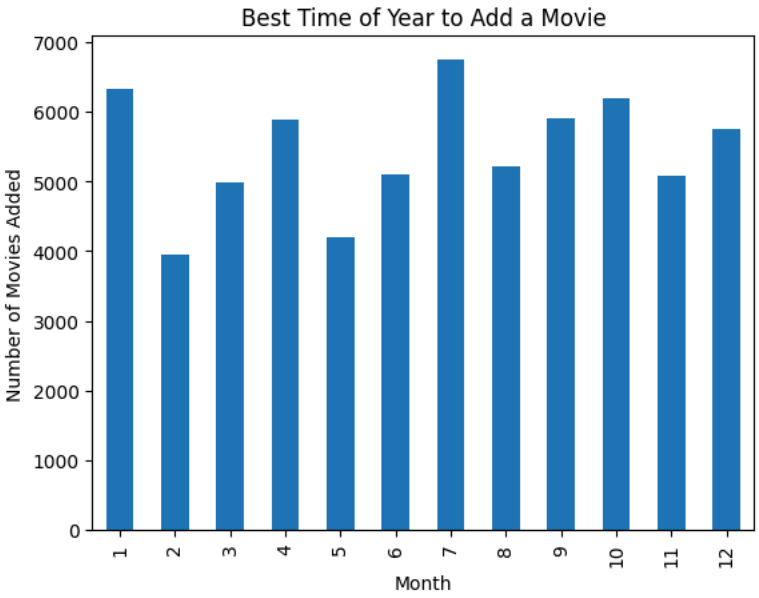
```
data[data['type'] == 'Movie'][['title','date_added']]
```

	title	date_added
0	Dick Johnson Is Dead	2021-09-25
6	My Little Pony: A New Generation	2021-09-24
6	My Little Pony: A New Generation	2021-09-24
6	My Little Pony: A New Generation	2021-09-24
6	My Little Pony: A New Generation	2021-09-24
...	...	...
8806	Zubaan	2019-03-02
8806	Zubaan	2019-03-02
8806	Zubaan	2019-03-02
8806	Zubaan	2019-03-02
8806	Zubaan	2019-03-02

65346 rows × 2 columns

```
movies_by_month = data[data['type'] == 'Movie'][['title','date_added']]
movies_by_month['month'] = pd.to_datetime(movies_by_month['date_added']).dt.month
movies_by_month_count = movies_by_month.groupby('month')['title'].count()
```

```
# Exhibit 7 :-

# Plotting a graph to check what is the best time of the year to add a Movie
movies_by_month_count.plot(kind='bar')
plt.xlabel('Month')
plt.ylabel('Number of Movies Added')
plt.title('Best Time of Year to Add a Movie')
plt.show()
```

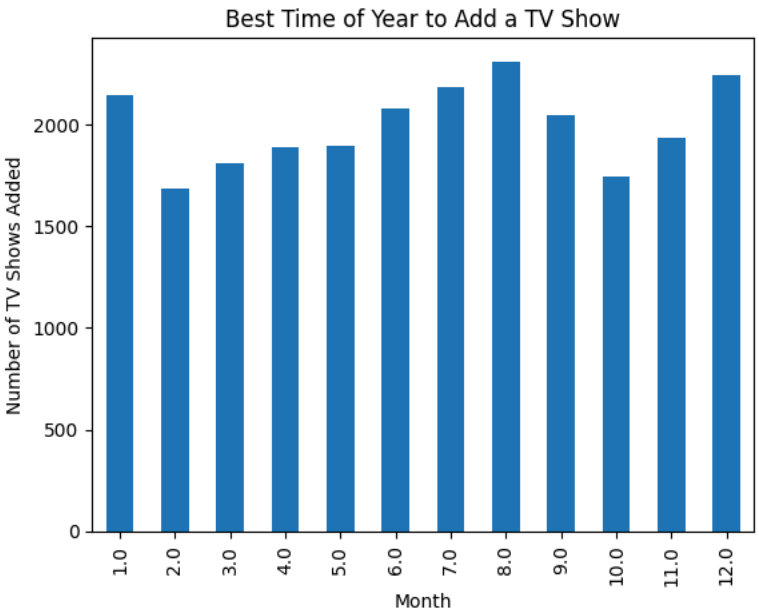


Insights :- We can see that nearly every month have similar count of TV shows added. February and May have the least and september has the highest.

Recommendation :- We can increase the number of TV shows added during the festive seasons and holiday seasons.

```
tv_shows_by_month = data[data['type'] == 'TV Show'][['title','date_added']]
tv_shows_by_month['month'] = pd.to_datetime(tv_shows_by_month['date_added']).dt.month
tv_shows_by_month_count = tv_shows_by_month.groupby('month')['title'].count()
```

```
# Exhibit 8 :-

# Plotting a graph to check what is the best time of the year to add a Movie

tv_shows_by_month_count.plot(kind='bar')
plt.xlabel('Month')
plt.ylabel('Number of TV Shows Added')
plt.title('Best Time of Year to Add a TV Show')
plt.show()
```

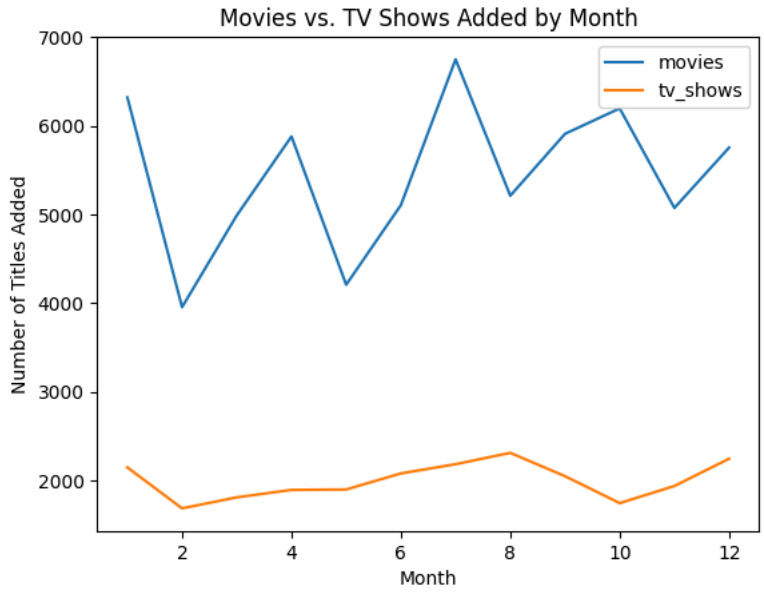


Insights :- We can see that nearly every month have similar count of TV shows added.

Recommendation :- We can increase the number of TV shows added during the festive seasons and holiday seasons.

```
movies_by_month = data[data['type'] == 'Movie'][['title','date_added']]
movies_by_month['month'] = pd.to_datetime(movies_by_month['date_added']).dt.month
movies_by_month_count = movies_by_month.groupby('month')['title'].count()

tv_shows_by_month = data[data['type'] == 'TV Show'][['title','date_added']]
tv_shows_by_month['month'] = pd.to_datetime(tv_shows_by_month['date_added']).dt.month
tv_shows_by_month_count = tv_shows_by_month.groupby('month')['title'].count()


# Combined Data
combined_data = pd.DataFrame({
    'month': movies_by_month_count.index,
    'movies': movies_by_month_count.values,
    'tv_shows': tv_shows_by_month_count.values
})
```

```
# Exhibit 9 :-

# Plotting line chart to check the trend of the TV shows vs Movies
combined_data.plot(x='month', y=['movies', 'tv_shows'], kind='line')
plt.xlabel('Month')
plt.ylabel('Number of Titles Added')
plt.title('Movies vs. TV Shows Added by Month')
plt.show()
```



Insights :- It is clear that consumers are streaming more Movies than that of the TV shows. They are most likely to invest in a Movie that than of the TV show, it might be because of the lengthy TV shows and time involvement.

Recommendations :- We need add more of the Movies than that of the TV shows.

```
!pip install google.colab
import google.colab
from google.colab import output
google.colab.files.save('Netflix_Business_case.pdf')
```