Business Case: Netflix - Data Exploration and Visualisation

```
#importing different libaries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

#Loading of dataset
df =
pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/asset
s/000/000/940/original/netflix.csv")

df
```

```
     show_id     type                  title         director  \
0         s1    Movie   Dick Johnson Is Dead  Kirsten Johnson
1         s2  TV Show          Blood & Water              NaN
2         s3  TV Show              Ganglands  Julien Leclercq
3         s4  TV Show  Jailbirds New Orleans              NaN
4         s5  TV Show           Kota Factory              NaN
...      ...      ...                    ...              ...
8802   s8803    Movie                 Zodiac    David Fincher
8803   s8804  TV Show            Zombie Dumb              NaN
8804   s8805    Movie             Zombieland  Ruben Fleischer
8805   s8806    Movie                   Zoom     Peter Hewitt
8806   s8807    Movie                 Zubaan     Mozez Singh

                                                  cast        country
\
0                                                  NaN  United States

1     Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...   South Africa

2     Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...            NaN

3                                                  NaN            NaN

4     Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...          India

...                                                ...            ...

8802  Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...  United States

8803                                               NaN            NaN

8804  Jesse Eisenberg, Woody Harrelson, Emma Stone, ...  United States
```

```
8805  Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...  United States
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...          India

              date_added  release_year rating   duration  \
0     September 25, 2021          2020  PG-13     90 min
1     September 24, 2021          2021  TV-MA  2 Seasons
2     September 24, 2021          2021  TV-MA   1 Season
3     September 24, 2021          2021  TV-MA   1 Season
4     September 24, 2021          2021  TV-MA  2 Seasons
...                  ...           ...    ...        ...
8802   November 20, 2019          2007      R    158 min
8803        July 1, 2019          2018  TV-Y7  2 Seasons
8804    November 1, 2019          2009      R     88 min
8805    January 11, 2020          2006     PG     88 min
8806       March 2, 2019          2015  TV-14    111 min

                                              listed_in  \
0                                         Documentaries
1       International TV Shows, TV Dramas, TV Mysteries
2     Crime TV Shows, International TV Shows, TV Act...
3                                Docuseries, Reality TV
4     International TV Shows, Romantic TV Shows, TV ...
...                                                 ...
8802                     Cult Movies, Dramas, Thrillers
8803             Kids' TV, Korean TV Shows, TV Comedies
8804                             Comedies, Horror Movies
8805                Children & Family Movies, Comedies
8806     Dramas, International Movies, Music & Musicals

                                            description
0     As her father nears the end of his life, filmm...
1     After crossing paths at a party, a Cape Town t...
2     To protect his family from a powerful drug lor...
3     Feuds, flirtations and toilet talk go down amo...
4     In a city of coaching centers known to train I...
...                                                 ...
8802  A political cartoonist, a crime reporter and a...
8803  While living alone in a spooky town, a young g...
8804  Looking to survive in a world taken over by zo...
8805  Dragged from civilian life, a former superhero...
8806  A scrappy but poor boy worms his way into a ty...

[8807 rows x 12 columns]
```

```
df.duplicated().sum()
```

```
0
```

#Insights

1. No duplicates in the data set
2. Columns 'Director','Cast' contains missing value or missing data. We need to change missing values to "No Data" so that our data stays accurate and our analysis remains fair
3. Duration of TV shows arev shown in seoenns and movies in minutes.

Columns information:

Show_id: Unique ID for every Movie / Tv Show

Type: Identifier - A Movie or TV Show

Title: Title of the Movie / Tv Show

Director: Director of the Movie

Cast: Actors involved in the movie/show

Country: Country where the movie/show was produced

Date_added: Date it was added on Netflix

Release_year: Actual Release year of the movie/show

Rating: TV Rating of the movie/show

Duration: Total Duration - in minutes or number of seasons

Listed_in italicized text: Genre

Description: The summary description

```
df.shape
```

```
(8807, 12)
```

This Dataset having 8807 rows and 12 columns.

#Handling missing values

```
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
```

It can be seen that the columns 'director', 'cast' , 'country have numerous missing values in which column 'director' have the most number of missing values.

```
# Replacing missing values in the 'director' column with 'No Data'
df['director'].replace(np.nan, 'No Data', inplace=True)

# Replacing missing values in the 'cast' column with 'No Data'
df['cast'].replace(np.nan, 'No Data', inplace=True)
```

For the ‘director’ and ‘cast’ columns, replacing missing values with ‘No Data’ to maintain data integrity and avoid any bias in the analysis.

```
# Filling missing values in the 'country' column with the mode value
df['country'] = df['country'].fillna(df['country'].mode()[0])
```

Using the mode (the value that appears most often) to fill in any missing values in the 'country' column. This helps maintain consistency in the data and reduces the amount of data that is not usable.

```
# Finding the mode rating for movies and TV shows
movie_rating = df.loc[df['type'] == 'Movie', 'rating'].mode()[0]
tv_rating = df.loc[df['type'] == 'TV Show', 'rating'].mode()[0]

# Filling missing rating values based on the type of content
df['rating'] = df.apply(lambda x: movie_rating if x['type'] == 'Movie'
and pd.isna(x['rating'])
                        else tv_rating if x['type'] == 'TV Show' and
pd.isna(x['rating'])
                        else x['rating'], axis=1)
```

For the ‘rating’ column, fill in missing values based on the ‘type’ of the show. We assign the mode of ‘rating’ for movies and TV shows separately.

```
# Finding the mode duration for movies and TV shows
movie_duration_mode = df.loc[df['type'] == 'Movie', 'duration'].mode()
[0]
```

```
tv_duration_mode = df.loc[df['type'] == 'TV Show', 'duration'].mode()
[0]

# Filling missing duration values based on the type of content
df['duration'] = df.apply(lambda x: movie_duration_mode if x['type']
== 'Movie'
                          and pd.isna(x['duration'])
                          else tv_duration_mode if x['type'] == 'TV
Show'
                          and pd.isna(x['duration'])
                          else x['duration'], axis=1)
```

For the 'duration' column, fill in missing values based on the 'type' of the show. We assign the mode of 'duration' for movies and TV shows separately.

```
# Dropping rows with missing values
df.dropna(inplace=True)
```

droping any remaining rows with missing values to ensure a clean dataset for analysis.

```
# Converting the 'date_added' column to datetime format
df["date_added"] = pd.to_datetime(df['date_added'])
```

converting the 'date_added' column to datetime format using pd.to_datetime() to enable further analysis based on date-related attributes.

```
# Extracting month, month name, and year from the 'date_added' column
df['month_added'] = df['date_added'].dt.month
df['month_name_added'] = df['date_added'].dt.month_name()
df['year_added'] = df['date_added'].dt.year
```

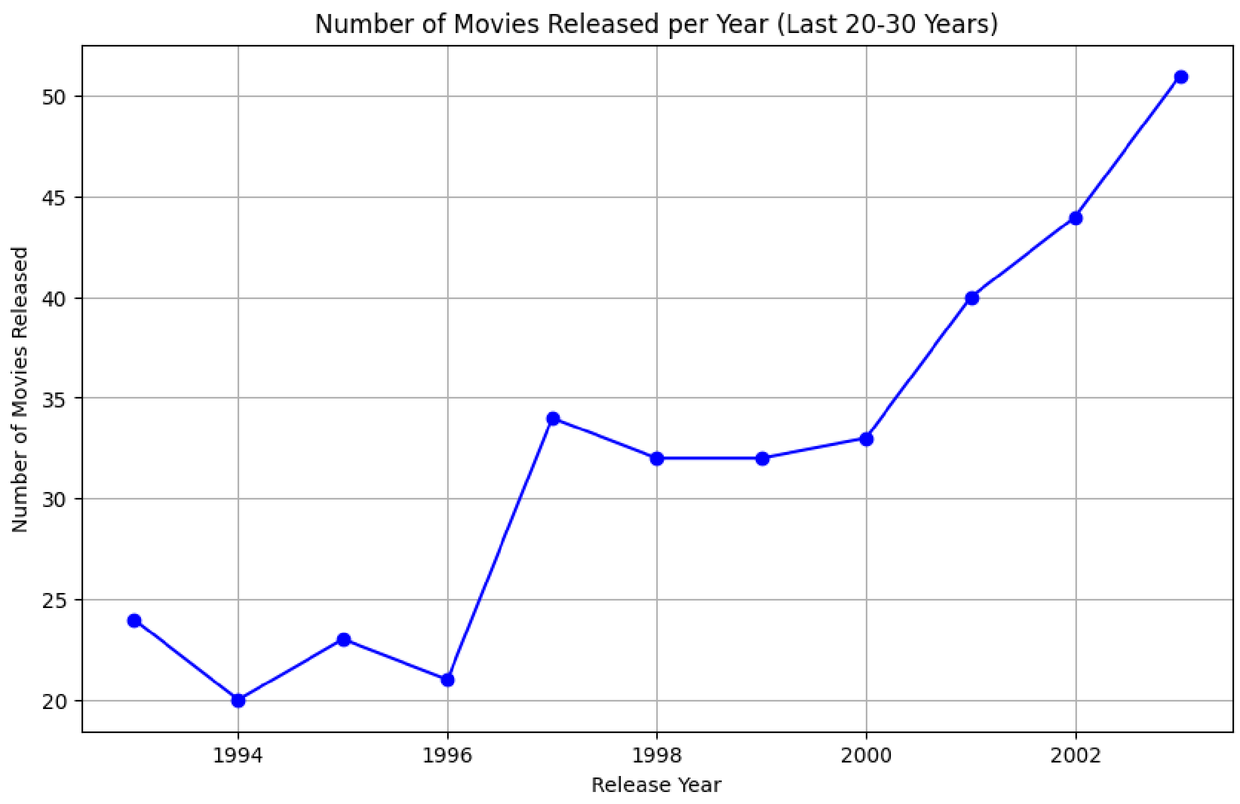
#Additional Data Transformations Extracting additional attributes from the 'date_added' column to enhance our analysis capabilities. We remove the month and year values to analyze trends based on these temporal aspects.

```
# Splitting and expanding the 'cast' column
df_cast = df['cast'].str.split(',', expand=True).stack()
df_cast = df_cast.reset_index(level=1, drop=True).to_frame('cast')
df_cast['show_id'] = df['show_id']

# Splitting and expanding the 'country' column
df_country = df['country'].str.split(',', expand=True).stack()
df_country = df_country.reset_index(level=1,
drop=True).to_frame('country')
df_country['show_id'] = df['show_id']

# Splitting and expanding the 'listed_in' column
df_listed_in = df['listed_in'].str.split(',', expand=True).stack()
df_listed_in = df_listed_in.reset_index(level=1,
drop=True).to_frame('listed_in')
df_listed_in['show_id'] = df['show_id']

# Splitting and expanding the 'director' column
df_director = df['director'].str.split(',', expand=True).stack()
df_director = df_director.reset_index(level=1,
drop=True).to_frame('director')
df_director['show_id'] = df['show_id']
```

#Data Transformation: Cast, Country, Listed In, and Director For a better analysis of categorical attributes, we create separate dataframes for them. This makes it easier to explore and analyze them at a more relaxed pace.

#Q1) How has the number of movies released per year changed over the last 20-30 years?

```
# Filter out the rows containing TV shows
movies_data = df[df['type'] == 'Movie']

# Group the data by release year and count the number of movies in
each year
movies_per_year =
movies_data['release_year'].value_counts().sort_index()

# Filter the data for the last 20-30 years
current_year = pd.Timestamp.now().year
years_to_consider = range(current_year - 30, current_year - 20 + 1)
movies_per_year =
movies_per_year[movies_per_year.index.isin(years_to_consider)]

# Plot the number of movies released per year
plt.figure(figsize=(10, 6))
plt.plot(movies_per_year.index, movies_per_year.values, marker='o',
linestyle='-', color='b')
plt.xlabel('Release Year')
plt.ylabel('Number of Movies Released')
plt.title('Number of Movies Released per Year (Last 20-30 Years)')
plt.grid(True)
plt.show()
```

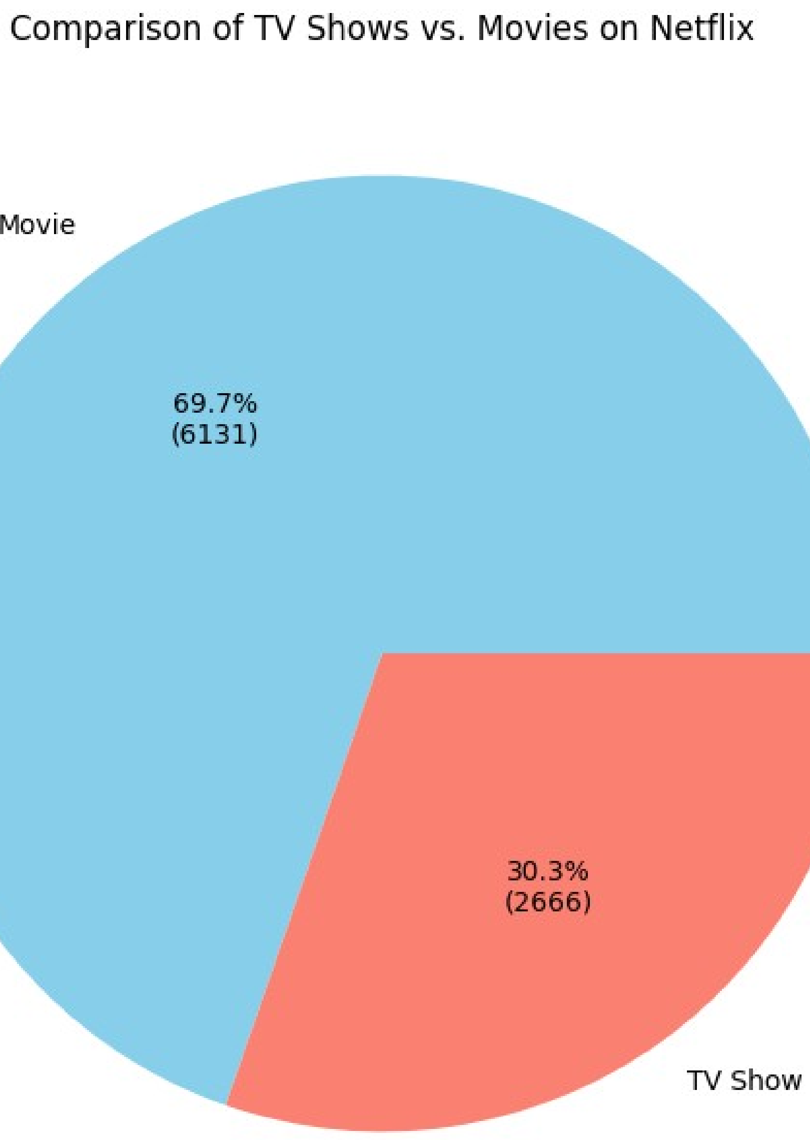


Insight

- Between the years 1993 and 1996, the annual count of movie releases remained within the range of 20 to 25.
- However, a notable spike occurred in 1997, where the number of movie releases surged substantially, ranging from 31 to 34.
- From the year 2000 onward, a consistent and prominent rise in the yearly count of movie releases can be observed.

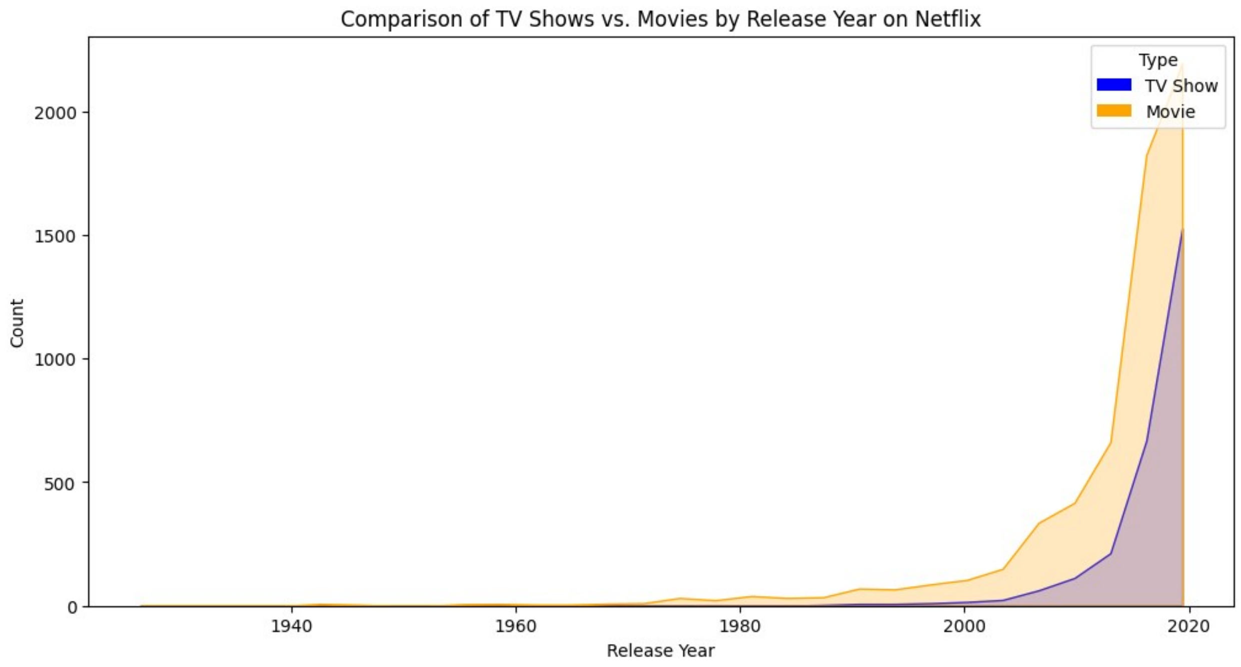
##Q2). Comparison of tv shows vs. movies.

```
# Compare TV shows vs. movies count
show_counts = df['type'].value_counts()

# Calculate the count of TV shows and movies
show_counts = df['type'].value_counts()

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(show_counts, labels=show_counts.index, autopct=lambda p:
'{:.1f}%\n({:.0f})'.format(p, p * sum(show_counts) / 100),
colors=['skyblue', 'salmon'])
plt.title('Comparison of TV Shows vs. Movies on Netflix')
plt.show()
```



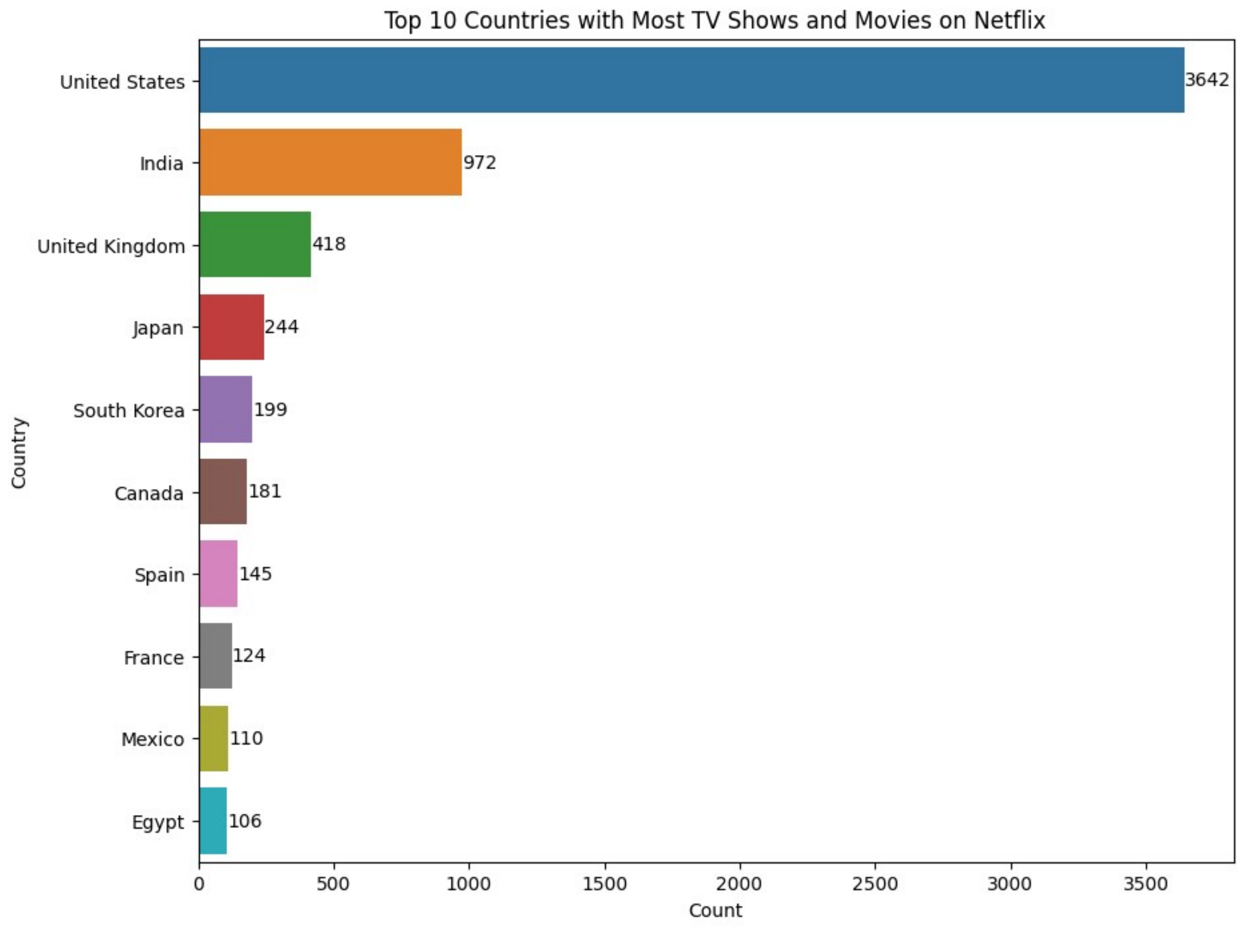
```
# Compare TV shows vs. movies by release year

import matplotlib.pyplot as plt
from matplotlib.patches import Patch



palette = {'TV Show': 'blue', 'Movie': 'orange'}
```

```python
plt.figure(figsize=(12, 6))

# Create an "ax" object to work with
ax = sns.histplot(data=df, x='release_year', hue='type',
element='poly', bins=30, palette=palette)


# Set title and labels
plt.title('Comparison of TV Shows vs. Movies by Release Year on
Netflix')
plt.xlabel('Release Year')
plt.ylabel('Count')

# Adjust x-axis and y-axis label font size
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)

# Manually add a legend
legend_elements = [Patch(facecolor=palette[label], label=label) for
label in palette]
ax.legend(handles=legend_elements, title='Type', fontsize=10)

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



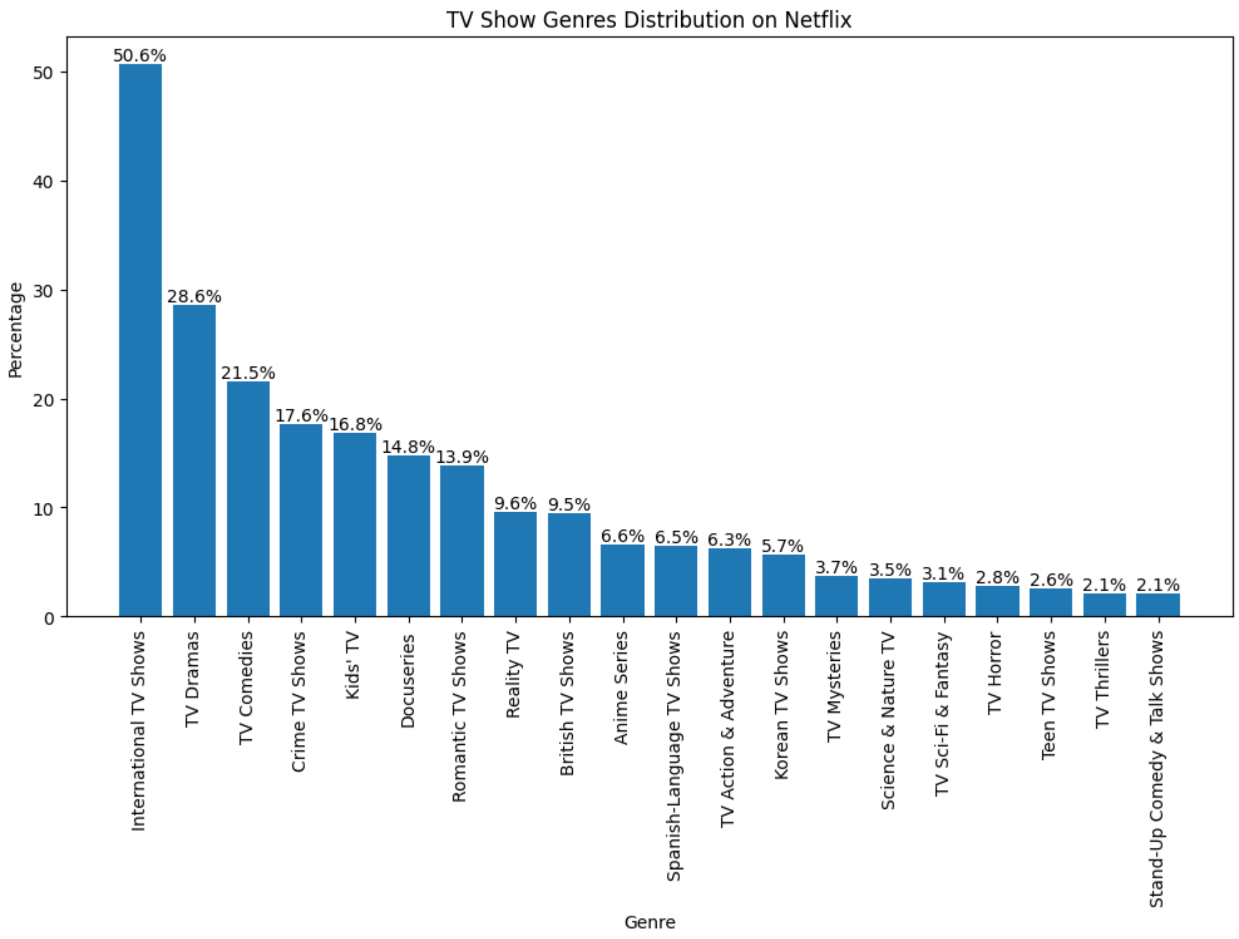
```python
# Compare TV shows vs. movies by country
top_countries = df['country'].value_counts().head(10)
```

```python
plt.figure(figsize=(10, 8))
ax = sns.barplot(x=top_countries.values, y=top_countries.index)
plt.title('Top 10 Countries with Most TV Shows and Movies on Netflix')
plt.xlabel('Count')
plt.ylabel('Country')

# Add count values on the bars
for p in ax.patches:
    ax.annotate(str(int(p.get_width())), (p.get_width(), p.get_y() +
p.get_height() / 2.),
                ha='left', va='center', fontsize=10, color='black')

plt.show()
```

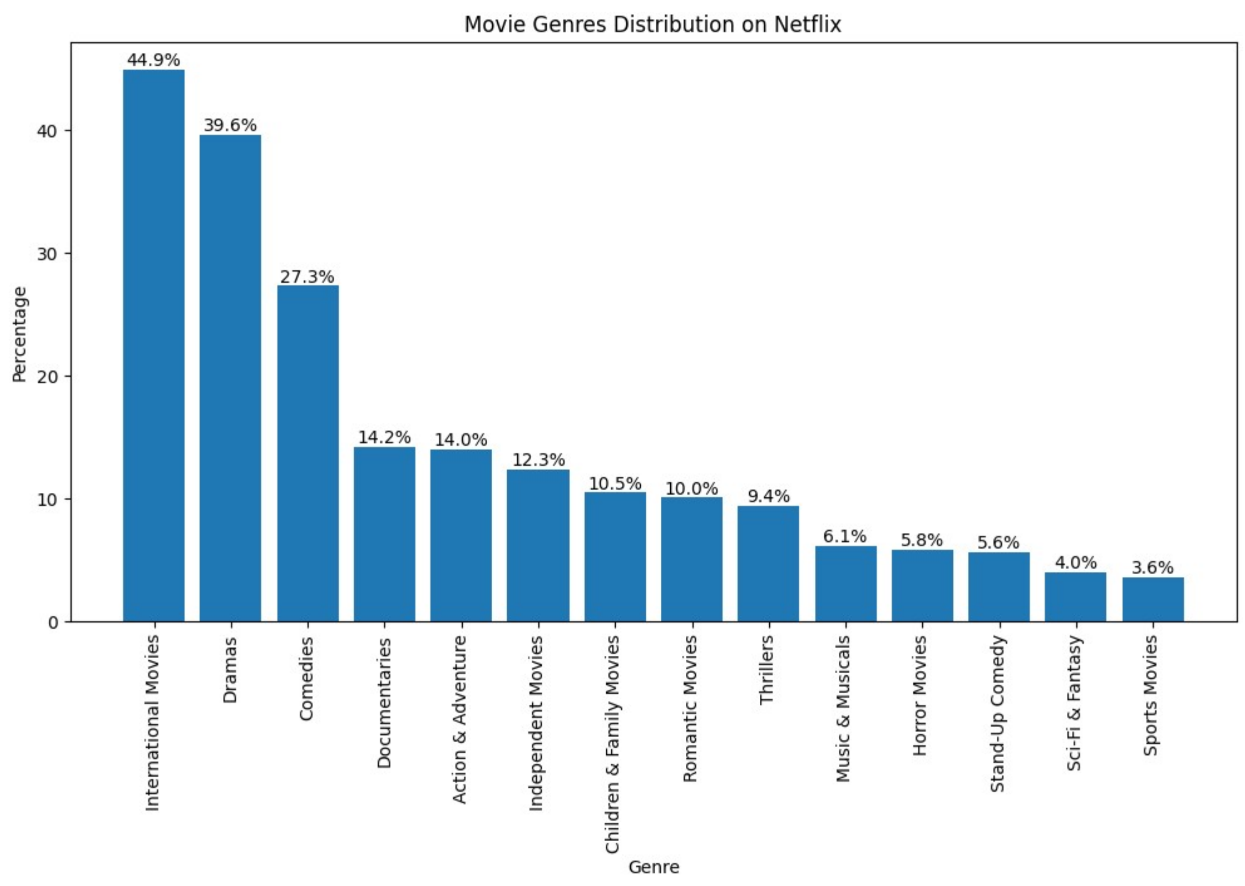


```python
# Extract unique genres
unique_genres = df['listed_in'].str.split(', ').explode().unique()

# Separate the data for TV shows and movies
tv_shows = df[df['type'] == 'TV Show']
movies = df[df['type'] == 'Movie']
```

```
# Calculate genre counts for TV shows
tv_show_genre_counts = tv_shows['listed_in'].str.split(',
').explode().value_counts()

# Calculate genre counts for movies
movie_genre_counts = movies['listed_in'].str.split(',
').explode().value_counts()

# Calculate total counts for percentage calculation
total_tv_shows = len(tv_shows)
total_movies = len(movies)

# Set a threshold for minimum bin percentage to show labels
threshold_percentage = 2.0  # You can adjust this threshold value as
needed

# Calculate percentages for TV show genres
tv_show_percentages = (tv_show_genre_counts / total_tv_shows) * 100

# Calculate percentages for movie genres
movie_percentages = (movie_genre_counts / total_movies) * 100

# Filter small bins and their labels based on threshold percentage
tv_show_filtered = tv_show_percentages[tv_show_percentages >=
threshold_percentage]
movie_filtered = movie_percentages[movie_percentages >=
threshold_percentage]

# Plot histogram for TV show genres
plt.figure(figsize=(12, 6))
plt.bar(tv_show_filtered.index, tv_show_filtered.values)
plt.title('TV Show Genres Distribution on Netflix')
plt.xlabel('Genre')
plt.ylabel('Percentage')

# Annotate plot with labels for filtered bins
for genre, percentage in tv_show_filtered.items():
    plt.annotate(f'{percentage:.1f}%', (genre, percentage),
ha='center', va='bottom')

plt.xticks(rotation=90)
plt.show()
```



```
# Plot histogram for movie genres
plt.figure(figsize=(12, 6))
plt.bar(movie_filtered.index, movie_filtered.values)
plt.title('Movie Genres Distribution on Netflix')
plt.xlabel('Genre')
plt.ylabel('Percentage')

# Annotate plot with labels for filtered bins
for genre, percentage in movie_filtered.items():
    plt.annotate(f'{percentage:.1f}%', (genre, percentage),
ha='center', va='bottom')

plt.xticks(rotation=90)
plt.show()
```



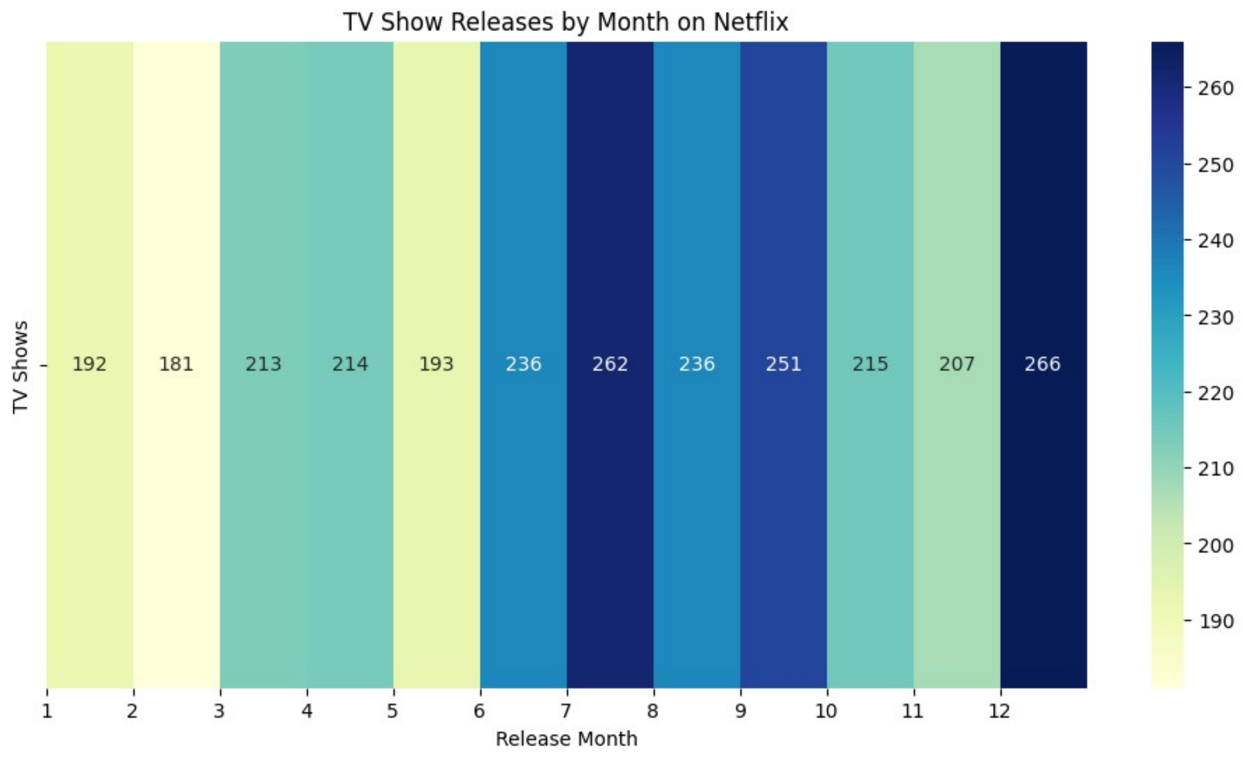
Insights

- Netflix offers a diverse collection of content. In the total of 8797 contents 69.7% (6131) are movies and 30.3%(2666) are TV shows.
- Comparing the two, the number of available Movies significantly surpasses that of TV Shows on Netflix.
- Starting from 2010, there has been a remarkable surge in the release of both Movies and TV Shows on Netflix, indicating the platform's soaring popularity.
- The United States stands out as the leading country of origin for Netflix's TV Shows and Movies. India and the United Kingdom follow closely in the second and third spots.
- When analyzing genres, a striking resemblance emerges between TV Shows and Movies. Both categories predominantly feature genres such as International, Dramas, and Comedies. Notably, 50% of TV Shows and 44% of Movies fall under the International genre, underscoring Netflix's global viewership.

Q3). What is the best time to launch a TV show?

```
# Convert Date_added to datetime format
df['date_added'] = pd.to_datetime(df['date_added'])

# Extract release month from date_added
df['release_month'] = df['date_added'].dt.month
```

```
# Filter TV Shows
tv_shows = df[df['type'] == 'TV Show']

# Count TV Shows by release month
release_month_counts =
tv_shows['release_month'].value_counts().sort_index()

# Create a heatmap
plt.figure(figsize=(12, 6))
sns.heatmap(release_month_counts.values.reshape(1, -1), cmap='YlGnBu',
annot=True, fmt='d', yticklabels=['TV Shows'])
plt.title('TV Show Releases by Month on Netflix')
plt.xlabel('Release Month')
plt.ylabel('')
plt.xticks(range(12), labels=[str(i + 1) for i in range(12)])  #
Adjust month labels
plt.show()
```



Insight

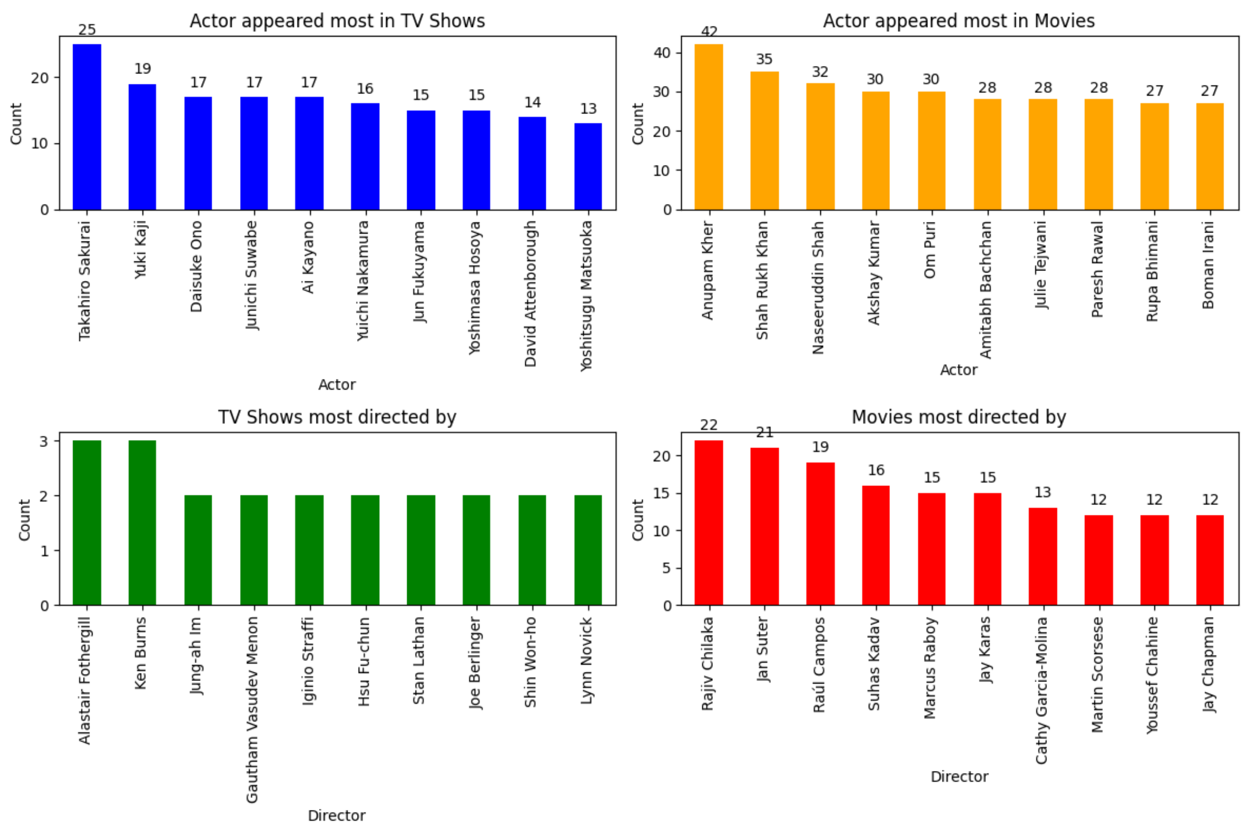
- The optimal time to launch a TV show is during the month of December, coinciding with the global holiday season. This period allows for a larger potential audience due to annual vacations around the world.
- Additionally, it's worth noting that a significant number of TV shows are released between June and September, inclusive. This trend can be attributed to the summer

vacation period in the United States, a major source of origin for many Netflix movies and TV shows. If the primary target audience for the TV show is predominantly in the U.S., the release window of June to September could be strategically chosen to align with the American summer vacation. However, if the intended audience is more diverse or global, December emerges as an optimal month for the TV show release due to the broader appeal of holiday-related leisure time.

Q4. Analysis of actors/directors of different types of shows/movies.

```
# Function to analyze actors/directors
def analyze_actors_directors(data, column, show_type):
    # Filter data by show type
    filtered_data = data[data['type'] == show_type]

    # Exclude rows with 'No Data' in the specified column
    filtered_data = filtered_data[filtered_data[column] != 'No Data']

    # Split the column values by comma and create a list
    split_values = filtered_data[column].str.split(',
').dropna().tolist()

    # Flatten the list of lists
    flattened_values = [item for sublist in split_values for item in
sublist]

    # Create a frequency count of actors/directors
    freq_count = pd.Series(flattened_values).value_counts()

    return freq_count

# Analysis of actors for TV Shows
tv_show_actors = analyze_actors_directors(df, 'cast', 'TV Show')
tv_show_actors = tv_show_actors.head(10)

# Analysis of actors for Movies
movie_actors = analyze_actors_directors(df, 'cast', 'Movie')
movie_actors = movie_actors.head(10)

# Analysis of directors for TV Shows
tv_show_directors = analyze_actors_directors(df, 'director', 'TV
Show')
tv_show_directors = tv_show_directors.head(10)

# Analysis of directors for Movies
movie_directors = analyze_actors_directors(df, 'director', 'Movie')
movie_directors = movie_directors.head(10)
```

```python
# Create bar charts
plt.figure(figsize=(12, 8))

plt.subplot(2, 2, 1)
tv_show_actors.plot(kind='bar', color='blue')
plt.title('Actor appeared most in TV Shows')
plt.xlabel('Actor')
plt.ylabel('Count')
for i, v in enumerate(tv_show_actors):
    plt.text(i, v + 1, str(v), ha='center', va='bottom', fontsize=10)

plt.subplot(2, 2, 2)
movie_actors.plot(kind='bar', color='orange')
plt.title('Actor appeared most in Movies')
plt.xlabel('Actor')
plt.ylabel('Count')
for i, v in enumerate(movie_actors):
    plt.text(i, v + 1, str(v), ha='center', va='bottom', fontsize=10)

plt.subplot(2, 2, 3)
tv_show_directors.plot(kind='bar', color='green')
plt.title('TV Shows most directed by')
plt.xlabel('Director')
plt.ylabel('Count')
#for i, v in enumerate(tv_show_directors):
    #plt.text(i, v + 1, str(v), ha='center', va='bottom', fontsize=10)

plt.subplot(2, 2, 4)
movie_directors.plot(kind='bar', color='red')
plt.title('Movies most directed by')
plt.xlabel('Director')
plt.ylabel('Count')
for i, v in enumerate(movie_directors):
    plt.text(i, v + 1, str(v), ha='center', va='bottom', fontsize=10)

plt.tight_layout()
plt.show()
```



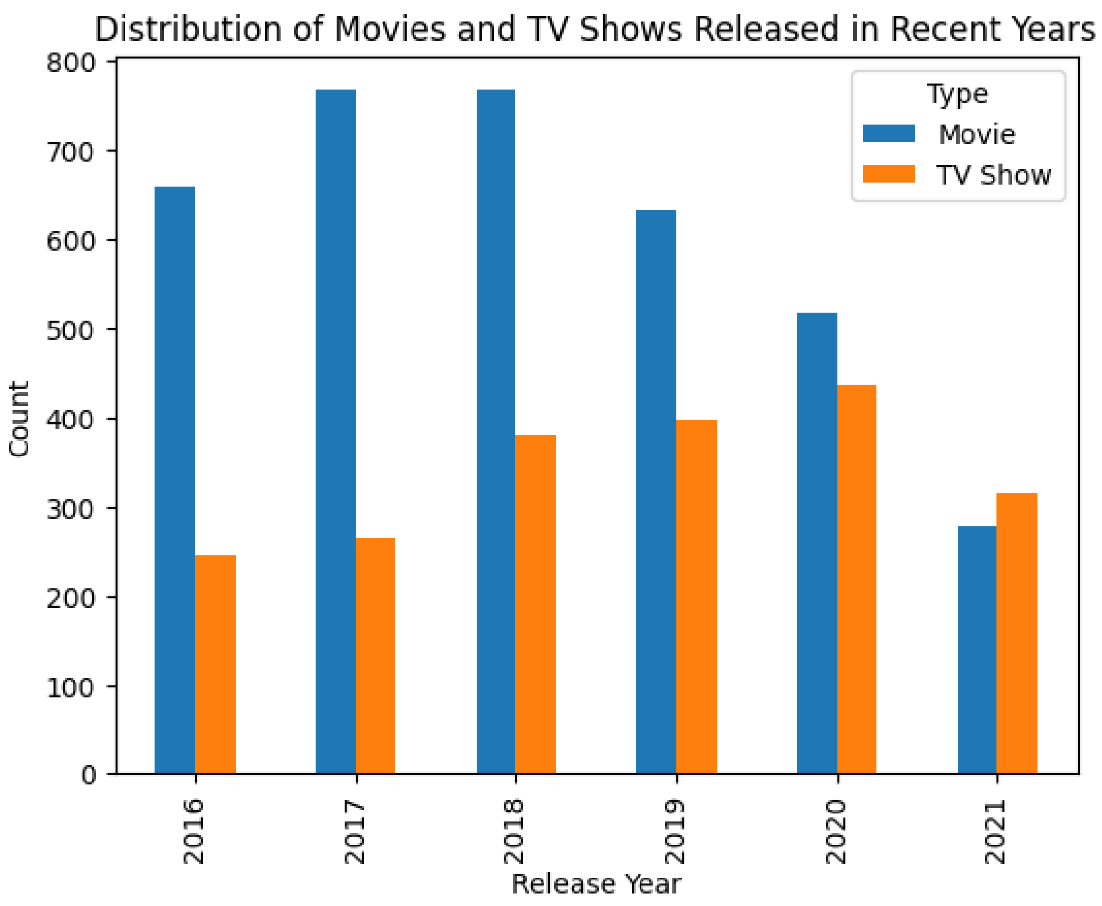
Insight

- Among the top Actors, Takahiro Sakurai has appeared 3 TV shows, and Anupam Kher has appeared 22 film
- Among the top directors, Alastair Fotherghill has directed 3 TV shows, and Rajiv Chilaka has directed 22 films

##Q5). Does Netflix has more focus on TV Shows than movies in recent years?

```
recent_years = range(2016,2024)  # Change the range as needed
recent_df = df[df['release_year'].isin(recent_years)]

# Count movies and TV shows by release year and type
counts = recent_df.groupby(['release_year', 'type'])
['show_id'].count().unstack()

# Plot dual bar charts
plt.figure(figsize=(10, 6))
counts.plot(kind='bar', stacked=False)
plt.title('Distribution of Movies and TV Shows Released in Recent
Years')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.legend(title='Type')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



Insight

- It has been observed that in recent years, the number of TV shows being released on Netflix has been gradually increasing, while the number of movies being released has shown a gradual decrease. In the year 2021, for the first time in the history of Netflix, the total count of TV shows exceeded the total count of movies released in that year. This indicates a shift in focus for Netflix towards producing more TV shows than movies.

Q6). Understanding what content is available in different countries

```
country = df["country"].apply(lambda x: str(x).split(", ")).tolist()
#exploding the country column
df_country = pd.DataFrame(country, index = df["title"])
df_country = df_country.stack()
df_country = df_country.reset_index()
df_country.drop(columns = "level_1" , inplace = True)
df_country.columns = ["title" , "country"]
```

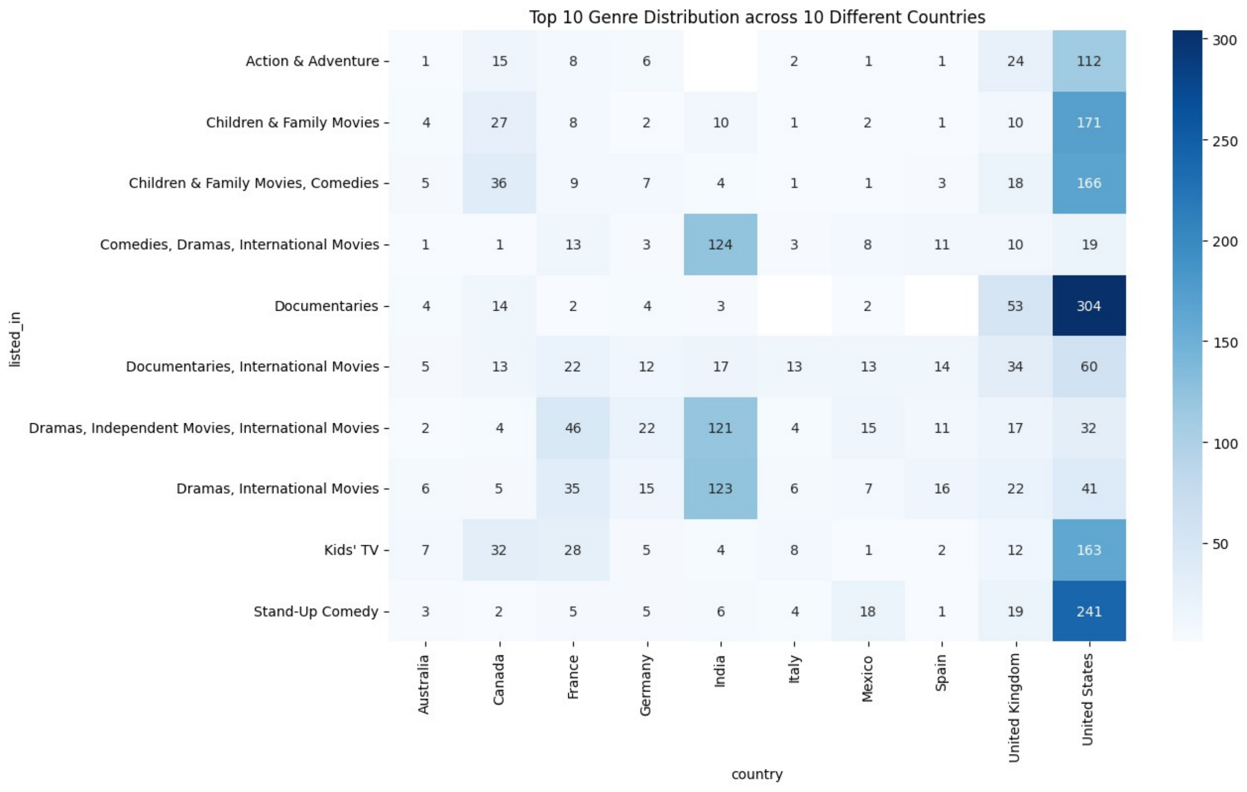
```
listed_in = df["listed_in"].apply(lambda x: str(x).split(",
")).tolist()
df_genre = pd.DataFrame(listed_in, index = df["title"])

df_trend_country = df.merge(df_country , on = "title")
df_trend_country.drop(columns = "country_x" , inplace = True)
df_trend_country.rename(columns = {"country_y":"country"}, inplace =
True)

temp = df_trend_country['country'].value_counts()[:11].reset_index()
temp.rename(columns = {'index':'country', 'country':'count'},
inplace=True)
country_list = temp['country'].tolist()
df_top10country =
df_trend_country.loc[df_trend_country['country'].isin(country_list)]
df_top10country = df_top10country.loc[df_top10country["country"]!
="Unknown"] #dropping of rows whose value is unknown.

genre_country_df= df_trend_country.merge(df_genre , on= "title")
genre_country_df.head(5)
```

```
  show_id     type                     title         director  \
0      s1    Movie   Dick Johnson Is Dead  Kirsten Johnson
1      s2  TV Show           Blood & Water          No Data
2      s3  TV Show               Ganglands  Julien Leclercq
3      s4  TV Show   Jailbirds New Orleans          No Data
4      s5  TV Show            Kota Factory          No Data

                                                cast
date_added  \
0                                            No Data  September 25,
2021
1  Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...  September 24,
2021
2  Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...  September 24,
2021
3                                            No Data  September 24,
2021
4  Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...  September 24,
2021

   release_year rating   duration  \
0          2020  PG-13     90 min
1          2021  TV-MA  2 Seasons
2          2021  TV-MA   1 Season
3          2021  TV-MA   1 Season
4          2021  TV-MA  2 Seasons

                                          listed_in  \
0                                     Documentaries
```

```
1    International TV Shows, TV Dramas, TV Mysteries
2  Crime TV Shows, International TV Shows, TV Act...
3                           Docuseries, Reality TV
4  International TV Shows, Romantic TV Shows, TV ...

                                         description        country  \
0  As her father nears the end of his life, filmm...  United States
1  After crossing paths at a party, a Cape Town t...   South Africa
2  To protect his family from a powerful drug lor...  United States
3  Feuds, flirtations and toilet talk go down amo...  United States
4  In a city of coaching centers known to train I...          India

                        0                        1
2
0          Documentaries                     None
None
1  International TV Shows               TV Dramas           TV
Mysteries
2         Crime TV Shows  International TV Shows  TV Action &
Adventure
3             Docuseries             Reality TV
None
4  International TV Shows       Romantic TV Shows            TV
Comedies

temp_genre = genre_country_df['listed_in'].value_counts()
[:10].reset_index()
temp_genre.rename(columns={'index': 'genre', 'listed_in': 'count'},
inplace=True)
genre_list = temp_genre['genre'].tolist()
df_top10_genre =
genre_country_df.loc[genre_country_df['listed_in'].isin(genre_list)]
df_top10_genre.head()

  show_id   type                             title  \
0      s1  Movie              Dick Johnson Is Dead
6      s7  Movie  My Little Pony: A New Generation
7      s8  Movie                           Sankofa
8      s8  Movie                           Sankofa
9      s8  Movie                           Sankofa

                        director  \
0                Kirsten Johnson
6  Robert Cullen, José Luis Ucha
7                   Haile Gerima
8                   Haile Gerima
9                   Haile Gerima

                                                cast
date_added  \
0                                              No Data  September 25,
2021
6  Vanessa Hudgens, Kimiko Glenn, James Marsden, ...  September 24,
2021
7  Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...  September 24,
2021
8  Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...  September 24,
2021
9  Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...  September 24,
2021

   release_year rating duration  \
0          2020  PG-13   90 min
6          2021     PG   91 min
7          1993  TV-MA  125 min
8          1993  TV-MA  125 min
9          1993  TV-MA  125 min

                                           listed_in  \
0                                      Documentaries
6                           Children & Family Movies
7  Dramas, Independent Movies, International Movies
8  Dramas, Independent Movies, International Movies
9  Dramas, Independent Movies, International Movies

                                         description        country  \
0  As her father nears the end of his life, filmm...  United States
6  Equestria's divided. But a bright-eyed hero be...  United States
7  On a photo shoot in Ghana, an American model s...  United States
8  On a photo shoot in Ghana, an American model s...          Ghana
9  On a photo shoot in Ghana, an American model s...   Burkina Faso

                          0                    1                      2

0             Documentaries                 None                   None

6  Children & Family Movies                 None                   None

7                    Dramas   Independent Movies   International Movies

8                    Dramas   Independent Movies   International Movies

9                    Dramas   Independent Movies   International Movies
```

```python
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 8))
sns.heatmap(heat_genre_final, annot=True, cmap="Blues", fmt=".0f")
```

```
plt.title("Top 10 Genre Distribution across 10 Different Countries")
plt.show()
```



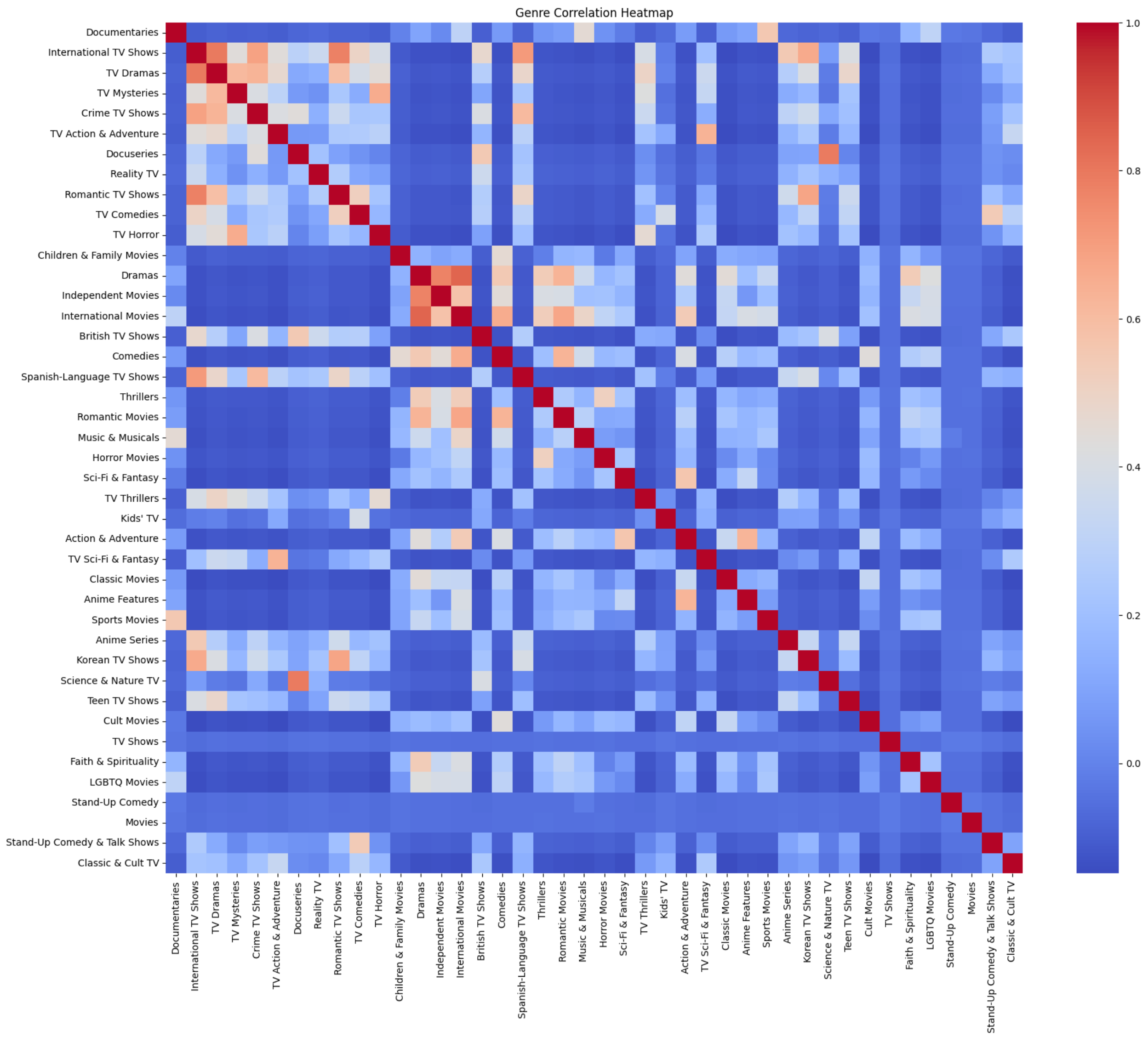
listed_in	Australia	Canada	France	Germany	India	Italy	Mexico	Spain	United Kingdom	United States
Action & Adventure	1	15	8	6		2	1	1	24	112
Children & Family Movies	4	27	8	2	10	1	2	1	10	171
Children & Family Movies, Comedies	5	36	9	7	4	1	1	3	18	166
Comedies, Dramas, International Movies	1	1	13	3	124	3	8	11	10	19
Documentaries	4	14	2	4	3		2		53	304
Documentaries, International Movies	5	13	22	12	17	13	13	14	34	60
Dramas, Independent Movies, International Movies	2	4	46	22	121	4	15	11	17	32
Dramas, International Movies	6	5	35	15	123	6	7	16	22	41
Kids' TV	7	32	28	5	4	8	1	2	12	163
Stand-Up Comedy	3	2	5	5	6	4	18	1	19	241

Insight

- Most of the contents are available in US
- In India dramas, comedies are more popular.

```
# Extracting unique genres from the 'listed_in' column
genres = df['listed_in'].str.split(', ', expand=True).stack().unique()

# Create a new DataFrame to store the genre data
genre_data = pd.DataFrame(index=genres, columns=genres, dtype=float)

# Fill the genre data DataFrame with zeros
genre_data.fillna(0, inplace=True)

# Iterate over each row in the original DataFrame and update the genre
data DataFrame
for _, row in df.iterrows():
    listed_in = row['listed_in'].split(', ')
    for genre1 in listed_in:
        for genre2 in listed_in:
            genre_data.at[genre1, genre2] += 1

# Create a correlation matrix using the genre data
```

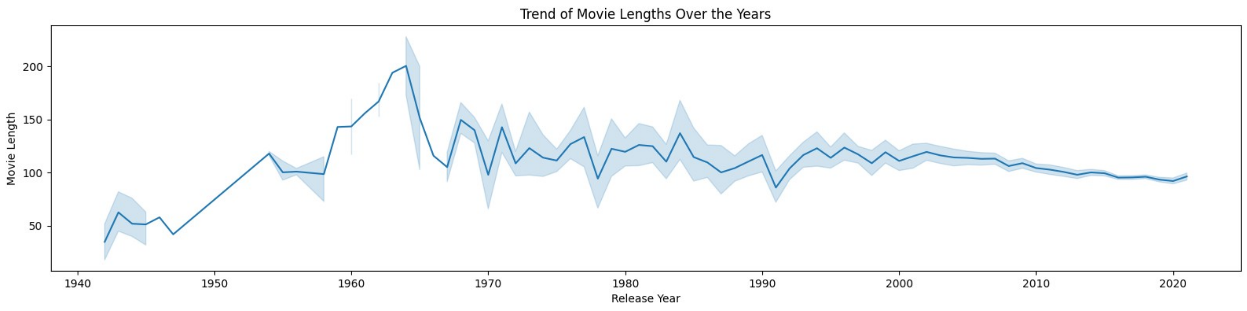
```
correlation_matrix = genre_data.corr()

# Create the heatmap
plt.figure(figsize=(20, 16))
sns.heatmap(correlation_matrix, annot=False, cmap='coolwarm')

# Customize the plot
plt.title('Genre Correlation Heatmap')
plt.xticks(rotation=90)
plt.yticks(rotation=0)

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

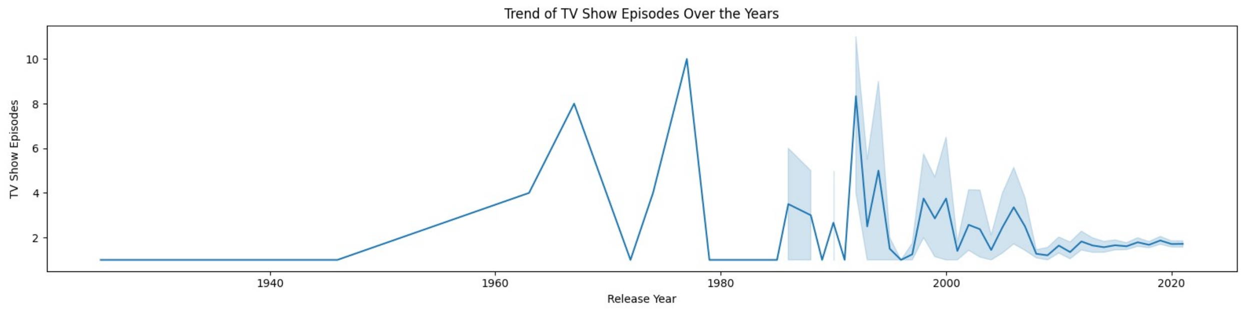


Insight

- The heatmap showcases the correlation between different genres. Analyzing the heatmap reveals pronounced positive correlations between specific genres, such as TV Dramas and International TV Shows, as well as Romantic TV Shows and International TV Shows.

```
df_movies = df[df['type'] == 'Movie']
df_tv_shows = df[df['type'] == 'TV Show']
# Extract the movie lengths and TV show episodes from the 'duration'
column
movie_lengths = df_movies['duration'].str.extract('(\d+)',
expand=False).astype(int)
tv_show_episodes = df_tv_shows['duration'].str.extract('(\d+)',
expand=False).astype(int)

# Create line plots for movie lengths and TV show episodes
plt.figure(figsize=(16, 8))

plt.subplot(2, 1, 1)
sns.lineplot(data=df_movies, x='release_year', y=movie_lengths)
plt.xlabel('Release Year')
plt.ylabel('Movie Length')
plt.title('Trend of Movie Lengths Over the Years')

plt.subplot(2, 1, 2)
sns.lineplot(data=df_tv_shows, x='release_year', y=tv_show_episodes)
plt.xlabel('Release Year')
plt.ylabel('TV Show Episodes')
plt.title('Trend of TV Show Episodes Over the Years')

# Adjust the layout and spacing
plt.tight_layout()

# Show the plots
plt.show()
```





Insight

- In recent years Netflix is prefering TV Shows with less than Two seasons.
- Since 2000's the duration of movie is around 100 - 120 minutes.

Conclusion

- Quantity : Netflix had more movies than TV Shows in thier Library.
- Between 1993 and 1996, movie releases numbered 20 to 25 annually. In 1997, a notable surge brought releases to 31 to 34. Post-2000, a consistent and strong growth in yearly movie counts occurred.
- Netflix boasts a diverse content collection with 8797 titles, comprising 69.7% movies and 30.3% TV shows. Movies outnumber TV shows. Post-2010, a significant surge in releases reflects Netflix's rising popularity. The US leads top countries, followed by India and the UK. Genre analysis shows both TV shows and movies focus on International, Dramas, and Comedies, highlighting Netflix's global viewership.
- The ideal TV show launch time is December, aligning with the worldwide holiday season and increased vacationing audience. Notably, a significant TV show influx occurs from June to September, driven by the U.S. summer break, suitable if the target audience is mainly American. For broader or global viewership, December's broader holiday allure remains strategically optimal.
- Recent trends reveal a rise in Netflix TV show releases and a decline in movies. In 2021, a historic shift occurred as TV show launches surpassed movies for the first time, highlighting Netflix's evolving emphasis on TV content production.
- The majority of content is accessible in the US. In India, dramas and comedies hold higher popularity levels.
- Anupam Kher is one of the Top Actor and Rajiv Chilaka is the top director in Netflix Contents. This signifies Netflix's targeted approach towards the Indian audience.

Recommendation

- Based on the analysis of these points, it is recommended that Netflix continues to diversify its content library, with a strategic focus on increasing its TV show offerings. The historical trend of rising TV show releases and the milestone in 2021 where TV shows surpassed movies underline the growing demand for TV content. To cater to their global viewership, Netflix should maintain a balance between genres like International, Dramas, and Comedies, which resonate with a wide audience.
- Considering the popularity of December releases aligning with the holiday season, Netflix should strategically plan the launch of their original TV shows during this period to capture a larger audience during peak leisure time. Additionally, for the American audience, capitalizing on the surge in TV show releases from June to September, corresponding with the US summer vacation, could be advantageous.
- As India exhibits a preference for dramas and comedies, Netflix should continue curating and producing content in these genres to cater to the preferences of the Indian audience. However, the company should also be mindful of the potential for diversification in India's content preferences over time.
- Overall, maintaining a balanced content mix, strategic release timing, and adapting content strategies to specific regional preferences will help Netflix continue its growth and global dominance in the streaming industry.
- Acknowledging Anupam Kher's stature as a top actor and Rajiv Chilaka's prominence as a leading director within Netflix's content offerings, it's astute to capitalize on their influence for strategic collaborations. Incorporating acclaimed Indian artists and directors could effectively amplify Netflix's appeal to the Indian audience, fostering greater engagement and viewership. By curating exclusive content that resonates with Indian sensibilities and culture, the platform can significantly bolster its presence in the Indian market. This approach aligns with Netflix's aim to tap into the vast potential of the Indian audience and ultimately drive substantial business growth.