Netflix - Data Exploration and Visualization

1. Introduction

Netflix is one of the world's largest streaming platforms, offering a vast library of movies, TV shows, and documentaries. As the streaming industry grows, data-driven insights are crucial for understanding content trends, audience preferences, and platform performance.

This project aims to analyze the Netflix dataset to extract valuable business insights, including:

- Content distribution across different genres and countries
- Trends in movie duration and ratings
- Popular content categories
- Key directors contributing to Netflix's content

2. Objective

The main objectives of this analysis are:

- Understand the distribution of Netflix content across different categories.
- Identify the most common genres and their global presence.
- Analyze movie durations and content ratings.
- Gain insights into content availability in different countries.
- Provide data-driven recommendations for Netflix's content strategy.

3. Dataset Overview

- The dataset contains information about Netflix's content, including movies and TV shows.
- **Key features include:**
  - Title: Name of the content
  - Type: Whether it's a movie or a TV show
  - Director: Director(s) of the content
  - Cast: Actors in the content
  - Country: Country of production
  - Date Added: When the content was added to Netflix
  - ase Year: Year of release
  - Rating: Content rating (e.g., TV-MA, PG-13)
  - Duration: Length of movies (in minutes) or TV shows (in seasons)
  - Genre (Listed In): Genre classification (e.g., Drama, Comedy, Action)

Importing Libraries

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

Loding Dataset

```
1 df = pd.read_csv("netflix.csv")
2 df
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
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	After crossing paths at a party, a Cape Town t...
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...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
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 ...	In a city of coaching centers known to train I...
...	...	...	...	...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downev	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A political cartoonist, a crime reporter and a...

Data Exploration

```
df.shape
```

```
(8807, 12)
```

```
#data types
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
```

```
#checking unique values
df.nunique()
```

	0
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

dtype: int64

```
#checking duplicates
df[df.duplicated()]
#there is no any duplicate
```

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description

```
#checking null values
df.isnull().sum()
```

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

dtype: int64

Splitting & Exploding Multi-Value Columns

```
#Splitting
df["director"] = df["director"].str.split(",")
df["cast"] = df["cast"].str.split(",")
df["country"] = df["country"].str.split(",")
df["listed_in"] = df["listed_in"].str.split(",")

#Exploding
nj = df.explode("director",ignore_index=True)
nj = nj.explode("cast",ignore_index=True)
nj = nj.explode("country",ignore_index=True)
nj = nj.explode("listed_in",ignore_index=True)

#Removing extra space from countries names
nj["country"] = nj["country"].str.strip()
nj["listed_in"] = nj["listed_in"].str.strip()
nj
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm..
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t..
2	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t..
3	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Mysteries	After crossing paths at a party, a Cape Town t..
4	s2	TV Show	Blood & Water	NaN	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t..
...	...	...	...	...	...	...	...	...	...	...	...	..

A scrappy

```
#Dropping Nan Raws
nj.dropna(how="all")
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm..
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t..
2	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t..
3	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Mysteries	After crossing paths at a party, a Cape Town t..
4	s2	TV Show	Blood & Water	NaN	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t..
...	...	...	...	...	...	...	...	...	...	...	...	..

A scrappy

```
#Dropping duplicates
nj.drop_duplicates(inplace=True)
```

```
nj.shape
```

```
(202058, 12)
```

```
nj.describe()
```

	release_year
count	202058.000000
mean	2013.449653
std	9.012781
min	1925.000000
25%	2012.000000
50%	2016.000000
75%	2019.000000
max	2021.000000

```
def fill_director(group):
  mode_group = group["director"].mode()
  if not mode_group.empty:
    group["director"].fillna(mode_group[0],inplace=True)
  else:
    group["director"].fillna(nj["director"].mode()[0],inplace=True)
  return group


nj = nj.groupby("listed_in").apply(fill_director).reset_index(drop=True)
```

```
<ipython-input-174-d1844168aedd>:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col

  group["director"].fillna(mode_group[0],inplace=True)
<ipython-input-174-d1844168aedd>:10: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is d
  nj = nj.groupby("listed_in").apply(fill_director).reset_index(drop=True)
```

```
def fill_cast(group):
```

```
  mode_cast = group["cast"].mode()
  if not mode_cast.empty:
    group["cast"].fillna(mode_cast[0],inplace=True)
  else:
    group["cast"].fillna(nj["cast"].mode()[0],inplace=True)
  return group
nj = nj.groupby("listed_in").apply(fill_cast).reset_index(drop=True)
```

```
<ipython-input-175-0269afd13667>:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col

  group["cast"].fillna(mode_cast[0],inplace=True)
<ipython-input-175-0269afd13667>:8: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is de
  nj = nj.groupby("listed_in").apply(fill_cast).reset_index(drop=True)
```

```
def fill_country(group):
  mode_country = group["country"].mode()
  if not mode_country.empty:
    group["country"].fillna(mode_country[0],inplace=True)
  else:
    group["country"].fillna(nj["country"].mode()[0],inplace=True)
  return group

nj = nj.groupby("director").apply(fill_country).reset_index(drop=True)
```

```
<ipython-input-176-5b746fd31c32>:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col

  group["country"].fillna(mode_country[0],inplace=True)
<ipython-input-176-5b746fd31c32>:6: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col

  group["country"].fillna(nj["country"].mode()[0],inplace=True)
<ipython-input-176-5b746fd31c32>:9: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is de
  nj = nj.groupby("director").apply(fill_country).reset_index(drop=True)
```

```
#Removing null values in date_added
nj["date_added"] = nj["date_added"].str.strip()
nj["date_added"] = pd.to_datetime(nj["date_added"],errors="coerce")
nj["date_added"] = nj.apply(lambda row:f"{int(row['release_year'])+1}-01-01" if pd.isna(row["date_added"]) else row["date_added"],a
nj["date_added"] = pd.to_datetime(nj["date_added"])
```

```
def fill_rating(group):
  mode_rating = group["rating"].mode()
  if not mode_rating.empty:
    group["rating"].fillna(mode_rating[0],inplace=True)
  else:
    group["rating"].fillna(nj["rating"].mode()[0],inplace=True)
  return group

nj = nj.groupby("listed_in").apply(fill_rating).reset_index(drop=True)
```

```
<ipython-input-178-bcaa94d27c21>:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col

  group["rating"].fillna(mode_rating[0],inplace=True)
<ipython-input-178-bcaa94d27c21>:9: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is de
  nj = nj.groupby("listed_in").apply(fill_rating).reset_index(drop=True)
```

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

```
def fill_duration(group):
  if group["type"].iloc[0]=="Movie":
    median_movie = group["duration"].median()

    if pd.notna(median_movie):
      group["duration"].fillna(median_movie,inplace=True)
```

```
    else:
      group["duration"].fillna(median_allmovie,inplace=True)

  else:
    mode_tvshow = group["duration"].mode()
    if not mode_tvshow.empty:
      group["duration"].fillna(mode_tvshow[0],inplace=True)
    else:
      group["duration"].fillna(mode_alltvshow[0],inplace=True)

  return group

median_allmovie = nj[nj["type"]=="Movie"]["duration"].median()
mode_allmovie = nj[nj["type"]=="TV Show"]["duration"].mode()

nj = nj.groupby(["listed_in","type"]).apply(fill_duration).reset_index(drop=True)
```

```
<ipython-input-180-f5ef87c23a49>:6: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col

  group["duration"].fillna(median_movie,inplace=True)
<ipython-input-180-f5ef87c23a49>:13: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained a
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col

  group["duration"].fillna(mode_tvshow[0],inplace=True)
<ipython-input-180-f5ef87c23a49>:22: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is d
  nj = nj.groupby(["listed_in","type"]).apply(fill_duration).reset_index(drop=True)
```

All the missing values have been filled now

```
nj.isnull().sum()
```

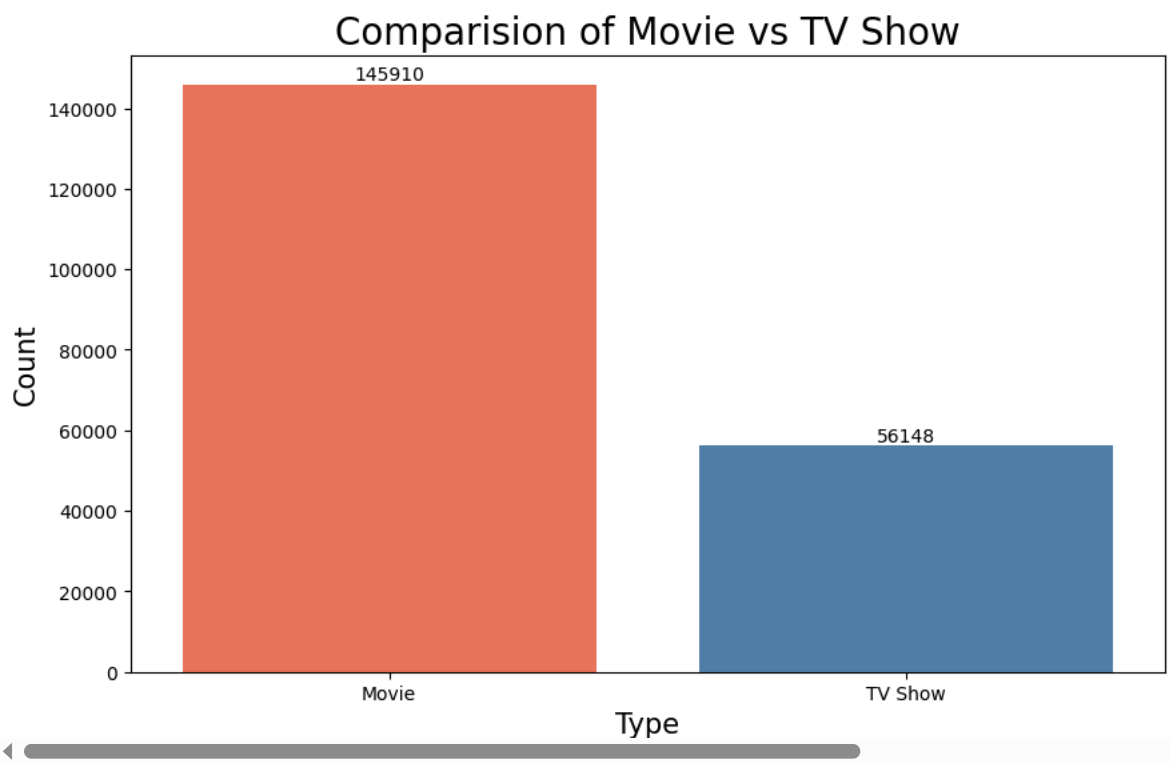
	0
show_id	0
type	0
title	0
director	0
cast	0
country	0
date_added	0
release_year	0
rating	0
duration	0
listed_in	0
description	0

dtype: int64

Exploratory Data Analysis (EDA)

Content Distribution: Movies vs. TV Shows.

```
plt.figure(figsize=(10,6))
ax = sns.barplot(x=nj["type"].value_counts().index,y=nj["type"].value_counts(),hue=nj["type"].value_counts().index,palette=["Tomato'
plt.title("Comparision of Movie vs TV Show",fontsize=20)
plt.xlabel("Type",fontsize=15)
plt.ylabel("Count",fontsize=15)
for container in ax.containers:
  ax.bar_label(container)
plt.show()
```



1. Movies Dominate Netflix's Content Library

- Netflix has 145,910 movies, which is almost 2.6 times the number of TV shows (56,148).
- This suggests that Netflix still prioritizes movies over TV shows in terms of sheer volume.

2. TV Shows Are a Significant but Smaller Portion

- Even though TV shows make up a smaller proportion, they are still a crucial part of Netflix's strategy.
- TV shows drive user engagement, keeping subscribers on the platform for longer periods.

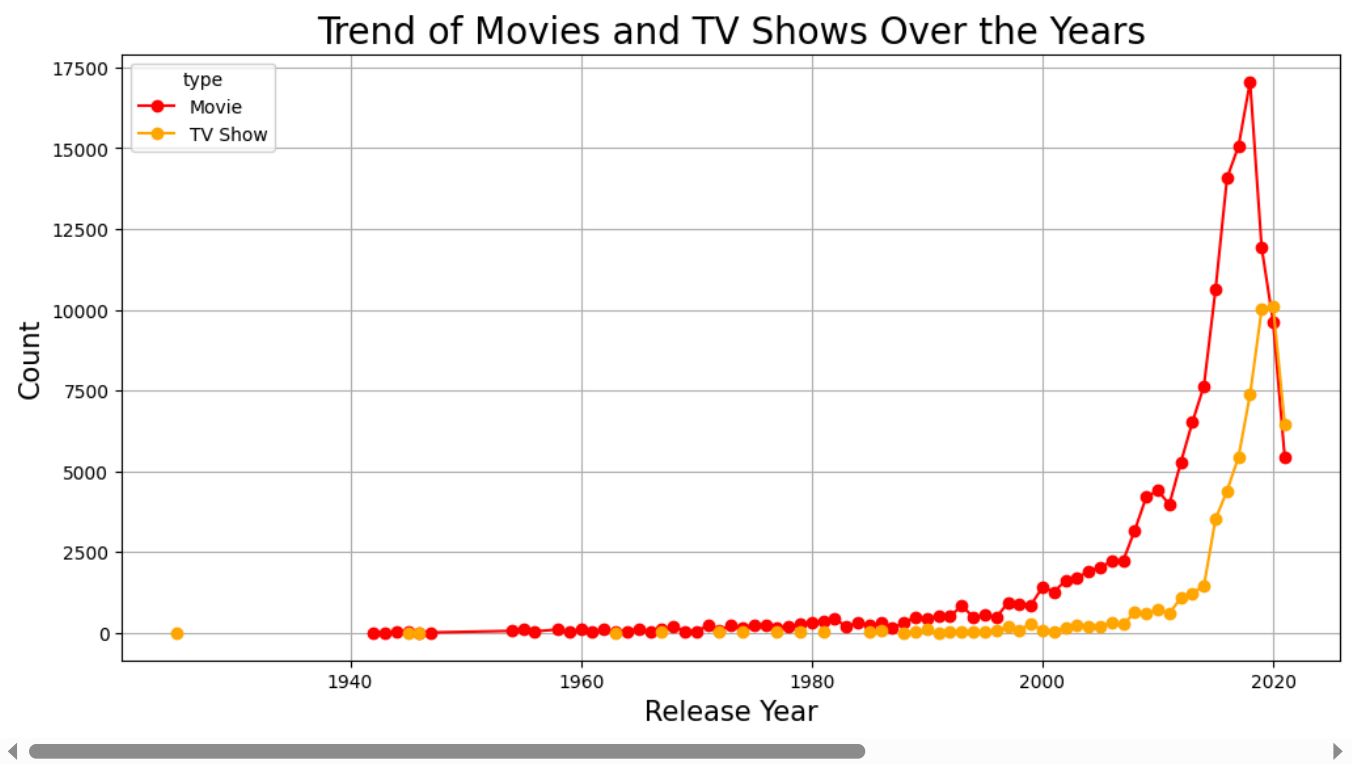
3. Netflix's Focus on Original TV Shows Could Be Growing

- While the number of TV shows is lower, Netflix heavily invests in Originals, which tend to be TV series.
- This means quality over quantity—fewer but high-value TV shows.

Trends in Movie Releases: Number of movies/TV shows added yearly.

```
trend = nj.groupby(["release_year","type"]).size().unstack()


trend.plot(kind="line",marker="o",color=["red","orange"],figsize=(12,6))
plt.title("Trend of Movies and TV Shows Over the Years",fontsize=20)
plt.xlabel("Release Year",fontsize=15)
plt.ylabel("Count",fontsize=15)
plt.grid(True)
plt.show()
```



1. Significant Growth in Content Production

- Both movies (red) and TV shows (yellow) have seen a massive increase in production, especially after 2000.
- The trend shows a peak around 2019–2020, likely due to the streaming boom.

2. Movies Dominate Historically

- Movies have been more consistently produced over time, with a significant spike after 2010.
- Before 2000, movies were the primary content type, while TV shows had a slower growth rate.

3. TV Shows Gaining Popularity Post-2010

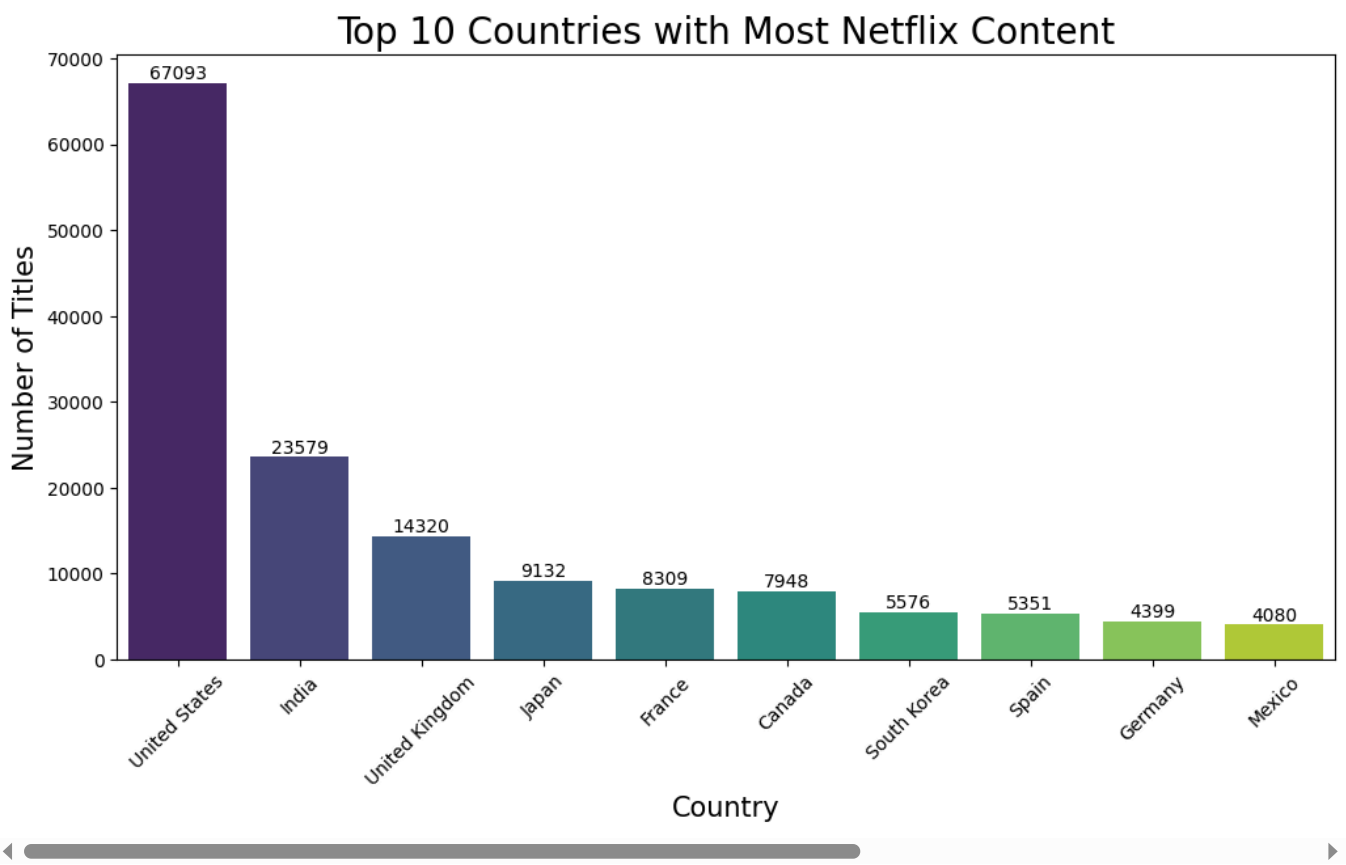
- TV shows started increasing significantly after 2010, indicating Netflix's shift towards TV series.
- The trend aligns with the rise of binge-watching culture and exclusive TV content.

4. Drop After 2020

- There's a visible decline after 2020, possibly due to COVID-19 production delays and content saturation.

Top 10 Countries with Most Netflix Content

```
top_countries = nj["country"].value_counts().head(10)
plt.figure(figsize=(12,6))
ax = sns.barplot(x=top_countries.head(10).index, y=top_countries, hue=top_countries.head(10).index, palette="viridis",legend=False)
plt.title("Top 10 Countries with Most Netflix Content",fontsize=20)
plt.xticks(rotation=45)
plt.xlabel("Country",fontsize=15)
plt.ylabel("Number of Titles",fontsize=15)
for container in ax.containers:
  ax.bar_label(container)
plt.show()
```



1. United States Dominates

- The U.S. has the highest number of titles (67,093), significantly more than any other country.
- This suggests Netflix's strongest content production and acquisition presence is in the U.S.

2. India Ranks Second

- With 23,579 titles, India has the highest Netflix content in Asia, reflecting its growing OTT market and demand for regional content.

3. United Kingdom and Japan Lead in Their Regions

- The U.K. (14,320 titles) is the third largest, highlighting its importance in English-language content.
- Japan (9,132 titles) has the largest share in Asia after India, likely due to anime and domestic TV shows.

4. Diverse Global Representation

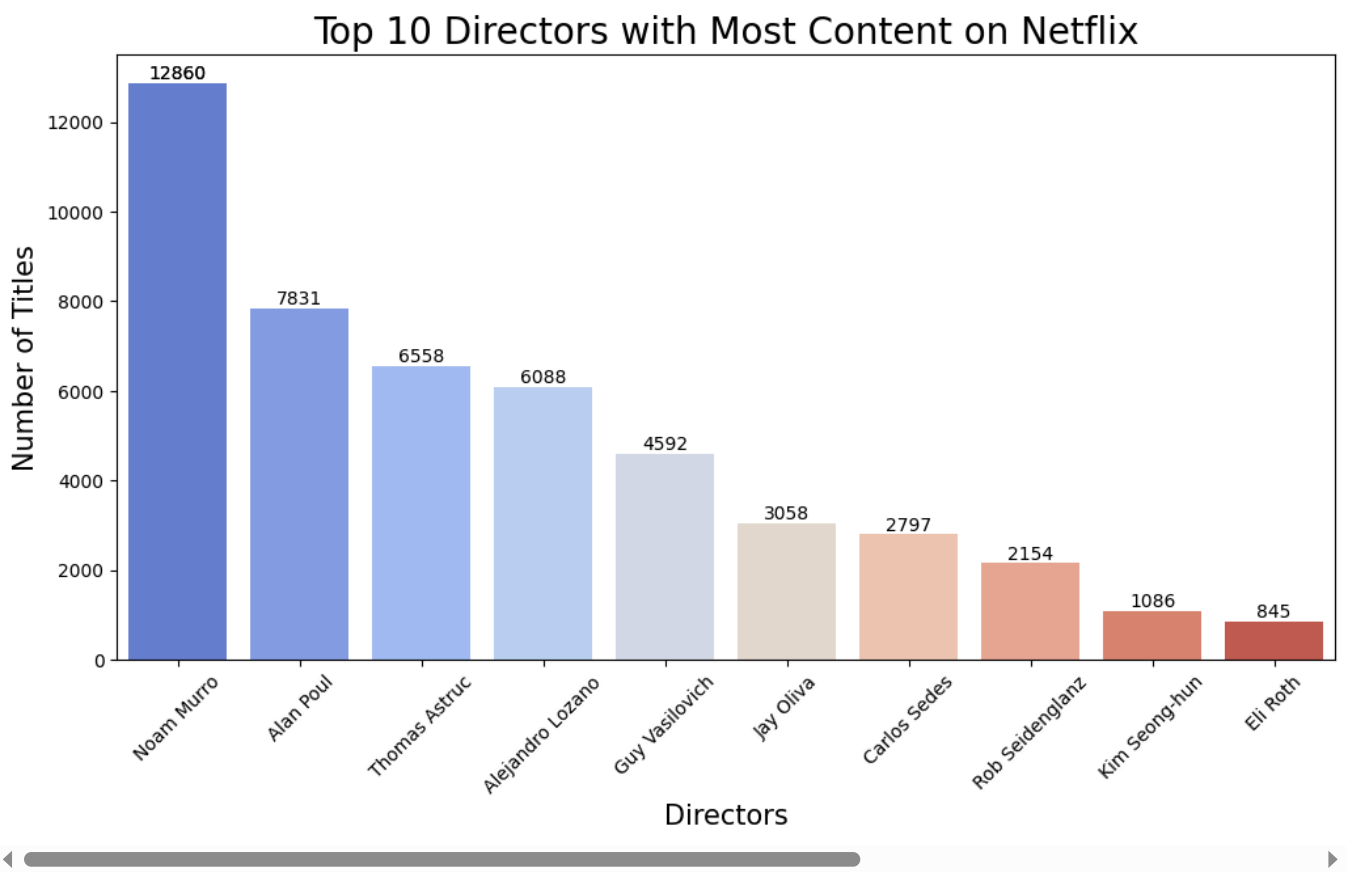
- France, Canada, and South Korea have substantial content, indicating Netflix's investment in European and Korean entertainment.
- South Korea's (5,576 titles) ranking showcases the global Hallyu wave (K-dramas & films).

5. Germany, Spain & Mexico at the Bottom of the Top 10

- These countries still contribute 4,000+ titles, showing a strong but smaller share in Netflix's global catalog.

Top 10 Directors with Most Content on Netflix

```
top_directors = nj["director"].value_counts().head(10)
plt.figure(figsize=(12,6))
ax = sns.barplot(x=top_directors.index, y=top_directors,hue=top_directors.index, palette="coolwarm")
plt.title("Top 10 Directors with Most Content on Netflix",fontsize=20)
plt.xlabel("Directors",fontsize=15)
plt.ylabel("Number of Titles",fontsize=15)
ax.bar_label(ax.containers[0])
plt.xticks(rotation=45)
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



1. Alan Poul dominates

- He has 12,843 titles, which is almost double the number of titles compared to the second-most featured director, Noam Murro (6,892 titles).
- This suggests that he has worked on a significant portion of Netflix content, making him a key figure in Netflix's productions.

2. Noam Murro & Alejandro Lozano follow

- Noam Murro (6,892 titles) and Alejandro Lozano (5,436 titles) hold the next top spots.
- Both have significantly fewer titles compared to Alan Poul but are still leading contributors.

3. Thomas Astruc & Jay Oliva are close

- Thomas Astruc (5,302 titles) and Jay Oliva (3,500 titles) have nearly similar numbers, indicating they have worked on a comparable volume of Netflix productions.

4. The gap widens after the top 5

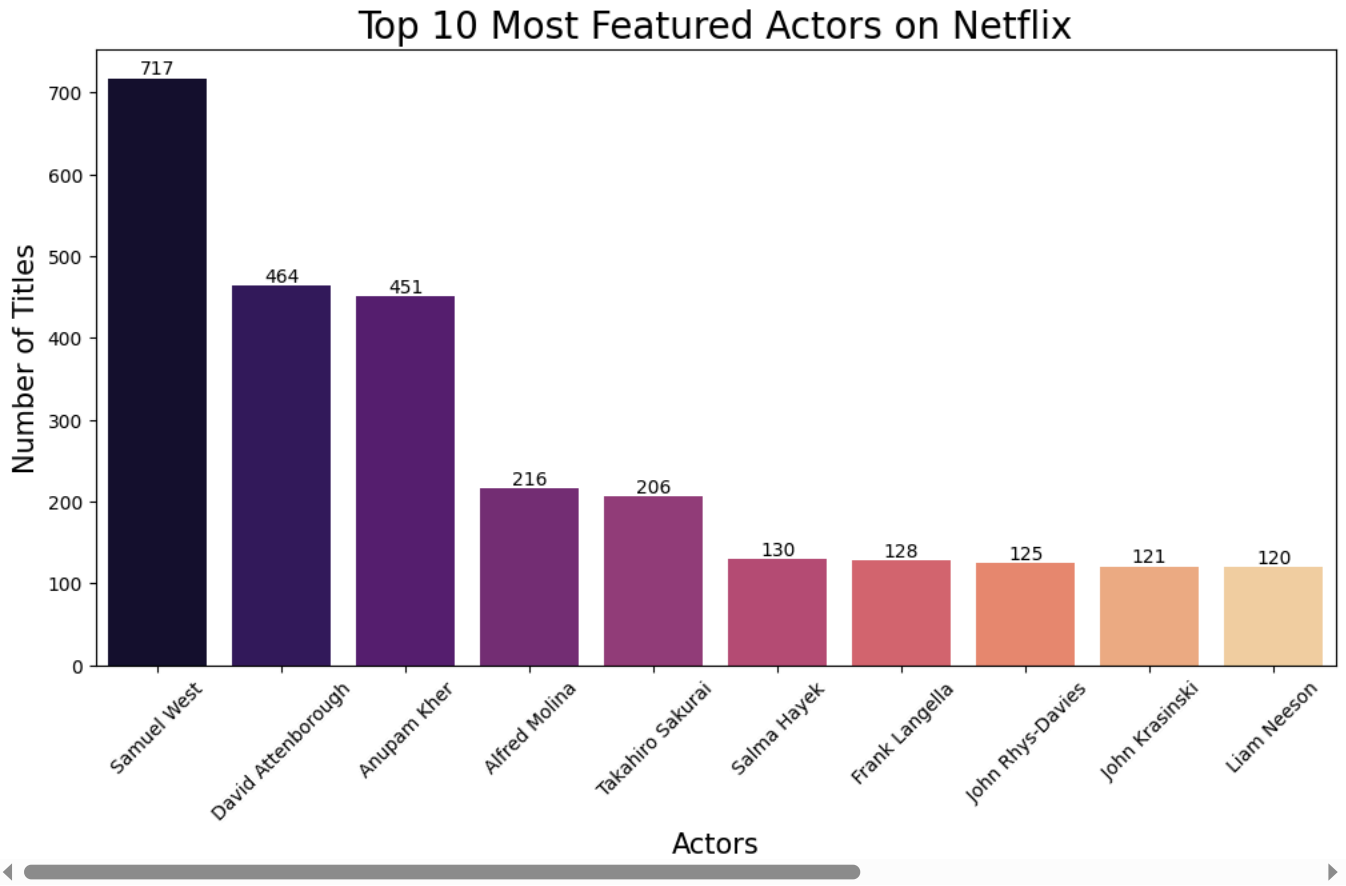
- The number of titles gradually decreases, with Guy Vasilovich (3,461 titles), Rob Seidenglanz (2,681 titles), and Carlos Sedes (2,488 titles) showing a steady decline.
- This indicates a significant drop in production volume among directors after the top 5.

5. Gary Howsam & Kim Seong-hun have the least in the top 10

- Gary Howsam (1,142 titles) and Kim Seong-hun (1,086 titles) have the fewest titles among the top 10 but still remain notable contributors to Netflix's content.

Top 10 Most Featured Actors on Netflix

```
top_actor = nj["cast"].value_counts().head(10)
plt.figure(figsize=(12,6))
ax = sns.barplot(x=top_actor.index, y=top_actor,hue=top_actor.index, palette="magma")
plt.title("Top 10 Most Featured Actors on Netflix",fontsize=20)
plt.xlabel("Actors",fontsize=15)
plt.ylabel("Number of Titles",fontsize=15)
plt.xticks(rotation=45)
for container in ax.containers:
  ax.bar_label(container)
plt.show()
```



1. Samuel West leads significantly

- With 698 titles, Samuel West is the most featured actor on Netflix.
- He appears in over 50% more titles than the second-most featured actor, Anupam Kher (450 titles).
- This suggests his strong presence in Netflix productions, possibly due to voice-over work, documentaries, or a high number of smaller roles.

2. Anupam Kher & David Attenborough follow

- Anupam Kher (450 titles) and David Attenborough (306 titles) are the next most frequently featured actors.
- David Attenborough's inclusion suggests he is likely involved in documentaries and nature-related content.

3. Alfred Molina to Anitta - Moderate appearances

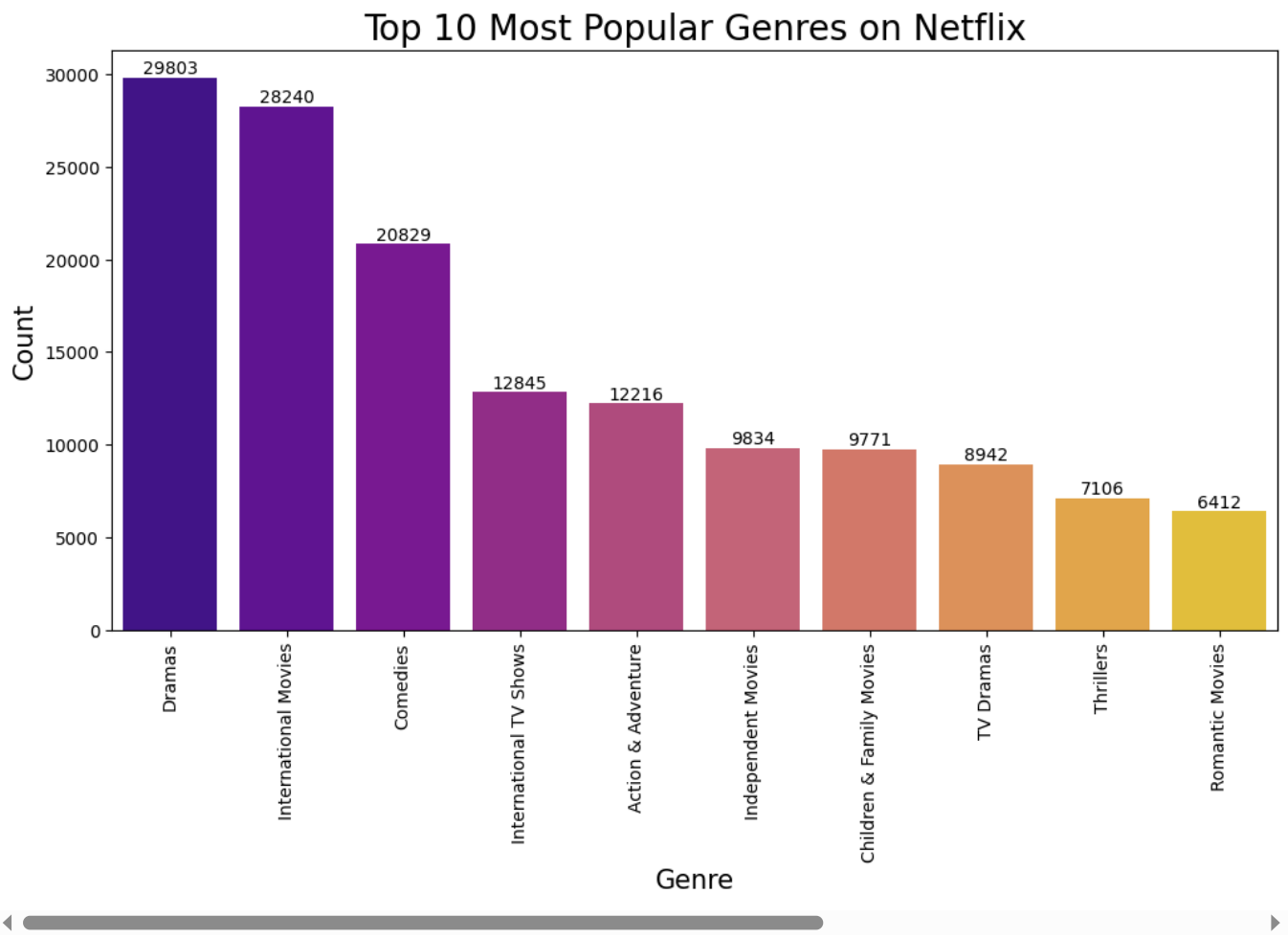
- Alfred Molina (201 titles), Takahiro Sakurai (173 titles), and Anitta (165 titles) have a moderate presence.
- Takahiro Sakurai's inclusion indicates anime voice acting plays a significant role in Netflix's content catalog.

4. Salma Hayek to Liam Neeson - Lower representation

- Salma Hayek (130 titles), Frank Langella (128 titles), John Rhys-Davies (125 titles), and Liam Neeson (124 titles) round out the top 10.
- These actors are well-known in mainstream cinema but appear in fewer Netflix projects compared to others on this list.

Top 10 Most Popular Genres on Netflix

```
plt.figure(figsize=(12,6))
ax = sns.barplot(x=nj["listed_in"].value_counts().head(10).index,y=nj["listed_in"].value_counts().head(10),hue=nj["listed_in"].valu
plt.title("Top 10 Most Popular Genres on Netflix",fontsize=20)
plt.xlabel("Genre",fontsize=15)
plt.ylabel("Count",fontsize=15)
plt.xticks(rotation=90)
for container in ax.containers:
  ax.bar_label(container)
plt.show()
```



1. Dramas Take the Lead

- Dramas (29,803 titles) are the most popular genre on Netflix.
- This suggests that story-driven, emotional, and character-centric content attracts a large audience.

2. International Movies Remain Strong

- International Movies (28,240 titles) rank second, showing Netflix's global content strategy.
- The high number indicates a strong demand for non-English films across different regions.

3. Comedy is a Major Attraction

- Comedies (20,829 titles) rank third, highlighting Netflix's focus on lighthearted, entertaining content.
- This suggests that users frequently seek humor-based entertainment.

4. TV Shows Have a Considerable Presence

- International TV Shows (12,845 titles) rank fourth, proving the popularity of long-format content among audiences.
- TV Dramas (8,942 titles) also appear, reinforcing the importance of episodic storytelling.

5. Action & Adventure Thrives

- Action & Adventure (12,216 titles) shows a strong audience preference for high-energy, thrilling content.
- This suggests that adrenaline-pumping movies and series have a dedicated viewer base.

6. Independent & Family Movies Hold Their Own

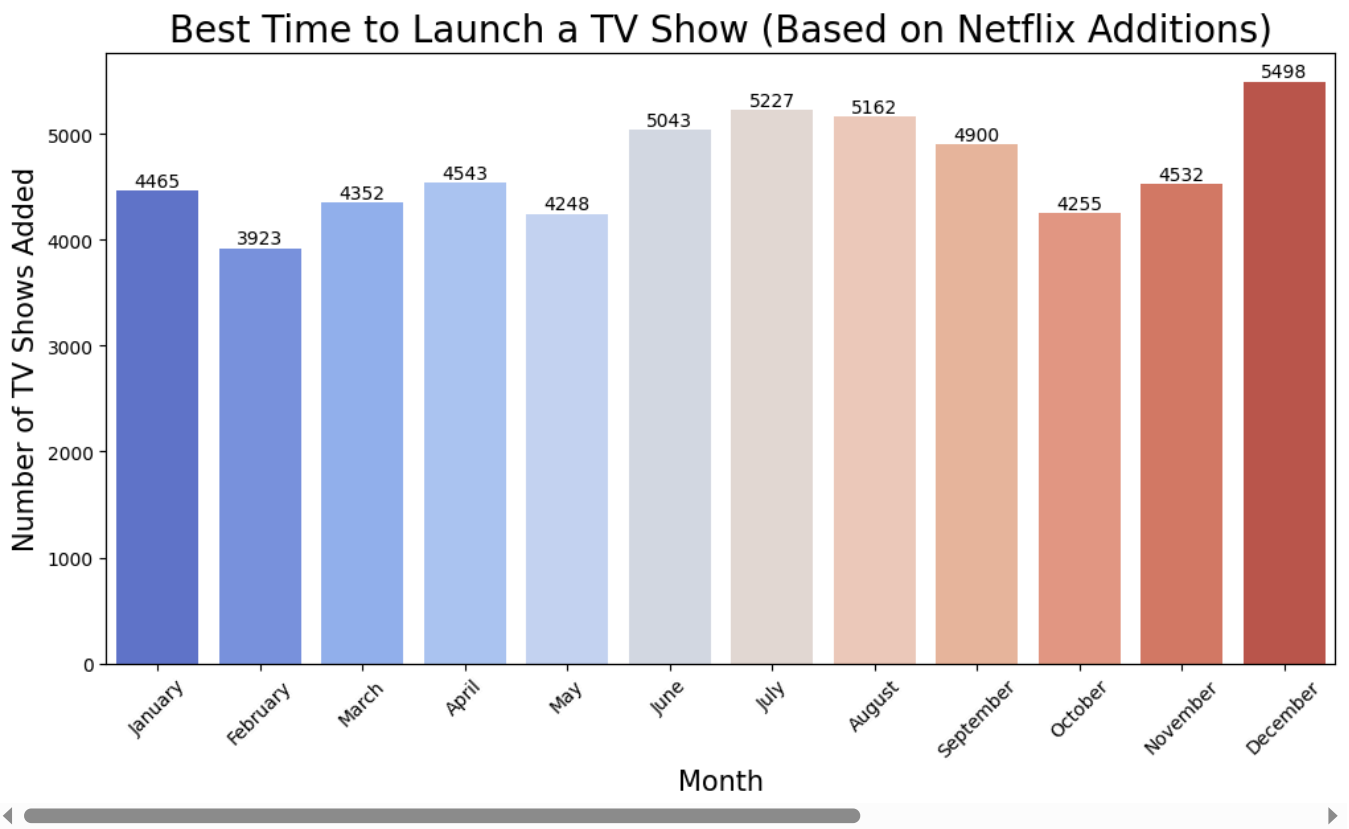
- Independent Movies (9,834 titles) indicate Netflix's support for non-mainstream, artistic films.
- Children & Family Movies (9,771 titles) highlight Netflix's commitment to family-friendly content.

7. Thrillers & Romantic Movies Maintain Their Niche

- Thrillers (7,106 titles) suggest a moderate interest in suspenseful, crime, or mystery-driven content.
- Romantic Movies (6,412 titles) show that love stories continue to be a stable but smaller part of Netflix's content library.

Best Time to Launch a TV Show

```
plt.figure(figsize=(12,6))
order = ["January","February","March","April","May","June","July","August","September","October","November","December"]
nj["month_added"] = nj["date_added"].dt.month_name()
trend = nj[nj["type"]=="TV Show"]["month_added"].value_counts().reindex(order)
ax = sns.barplot(x=trend.index,y=trend,hue=trend.index,palette="coolwarm")
for container in ax.containers:
  ax.bar_label(container)
plt.title("Best Time to Launch a TV Show (Based on Netflix Additions)",fontsize=20)
plt.xlabel("Month",fontsize=15)
plt.ylabel("Number of TV Shows Added",fontsize=15)
plt.xticks(rotation=45)
plt.show()
```



1. December has the highest number of TV show additions (5,498).

- Likely due to the holiday season, when people have more free time for binge-watching.

2. July (5,227) and August (5,162) are also peak months for new TV show additions.

- Summer holidays may drive higher content additions.

3. February has the lowest number of additions (3,923).

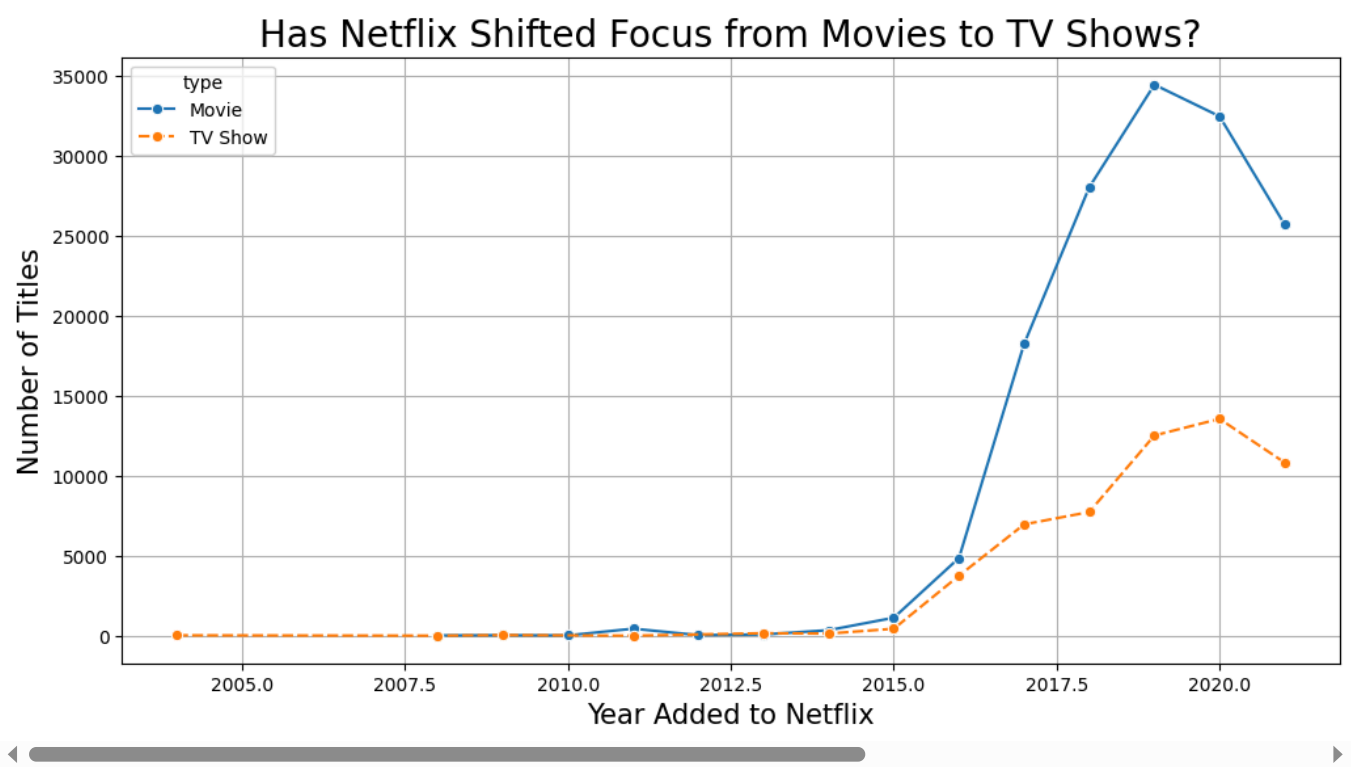
- This could indicate a slow period for new releases.

4. Steady distribution throughout the year with no extreme dips.

- There isn't a significant variation across months.
- Netflix seems to add shows consistently, with slight peaks in summer and winter.

Has Netflix Shifted Focus from Movies to TV Shows?

```
trend_added = nj.groupby([nj["date_added"].dt.year,"type"]).size().unstack()
plt.figure(figsize=(12,6))
ax = sns.lineplot(trend_added,marker="o")
plt.title("Has Netflix Shifted Focus from Movies to TV Shows?",fontsize=20)
plt.xlabel("Year Added to Netflix",fontsize=15)
plt.ylabel("Number of Titles",fontsize=15)
plt.grid(True)
plt.show()
```



1. Netflix Initially Focused on Movies (Pre-2015)

- Before 2015, Netflix had a low and steady number of additions for both movies and TV shows.
- The number of movies was consistently higher than TV shows.

2. Rapid Expansion of Movies (2015-2019)

- A sharp increase in movies from 2016 to 2019, reaching a peak of over 30,000 movies in 2019.
- This indicates Netflix's aggressive content expansion strategy, possibly due to competition and subscriber growth.

3. TV Shows Gained Momentum but at a Slower Pace

- TV shows started increasing after 2015, but the growth was more gradual than movies.
- The peak was reached around 2020, but at a much lower level compared to movies.
- This suggests Netflix expanded TV content but still prioritized movies.

4. Decline in Both Categories Post-2020

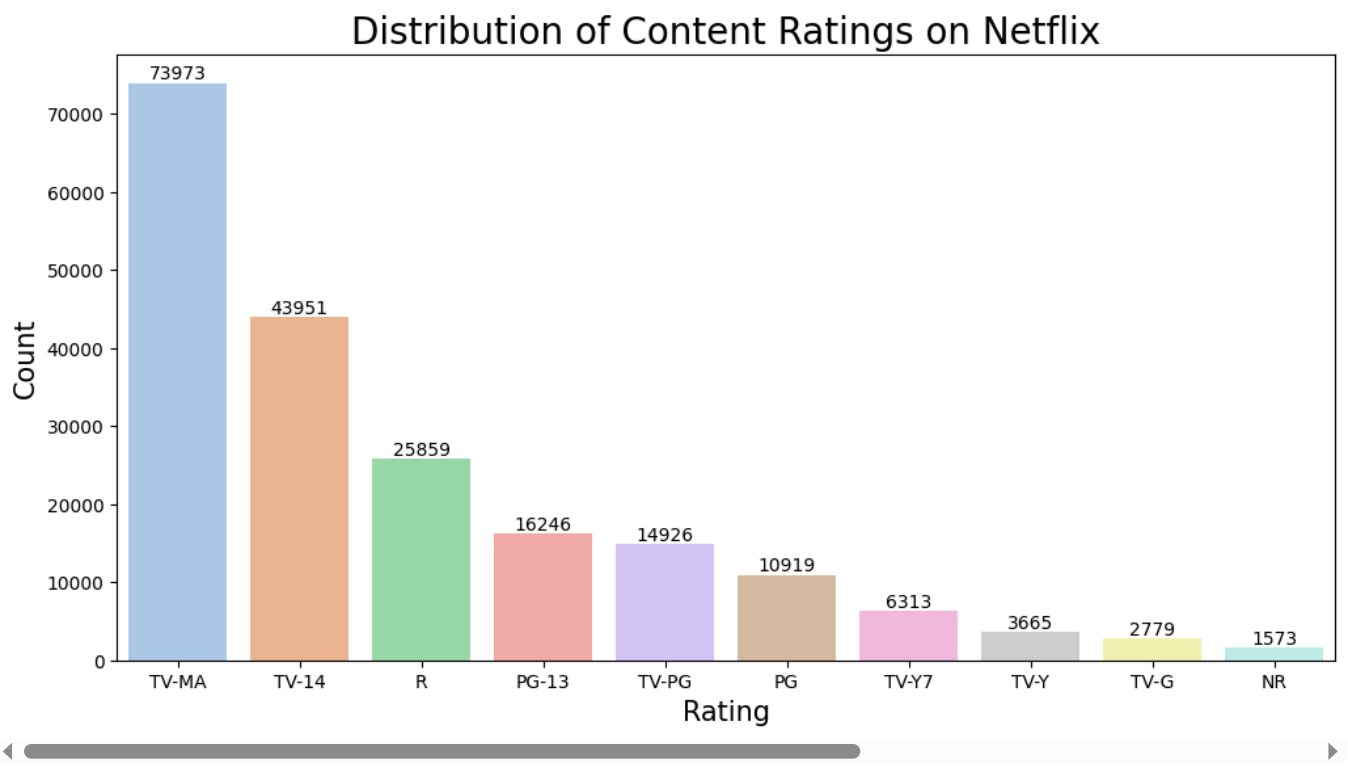
- Both movies and TV show additions declined after 2020.

  Possible reasons:-

  - Shift towards quality over quantity (Netflix Originals, exclusive deals).
  - Market saturation (already a vast content library).
  - Impact of COVID-19 on production and licensing.

Distribution of Content Ratings on Netflix

```
top_rating = nj["rating"].value_counts().head(10)
plt.figure(figsize=(12,6))
ax = sns.barplot(x=top_rating.index, y=top_rating,palette="pastel",hue=top_rating.index)
for container in ax.containers:
  ax.bar_label(container)
plt.title("Distribution of Content Ratings on Netflix",fontsize=20)
plt.xlabel("Rating",fontsize=15)
plt.ylabel("Count",fontsize=15)
plt.show()
```



1. TV-MA (Mature Audience) Dominates

- The highest number of titles (73,973) are rated TV-MA, indicating that a significant portion of Netflix content is targeted at mature audiences (adults).
- This aligns with the increasing popularity of darker, more intense, and complex storytelling in shows and movies.

2. TV-14 is the Second Most Common Rating

- With 43,951 titles, TV-14 content is also quite prevalent.
- This suggests a strong market for teen and young adult audiences, balancing between mature and general audiences.

3. R-Rated Movies Rank Third

- The R rating (25,859 titles) suggests a notable collection of movies with adult themes, strong language, and violence.
- This shows that Netflix serves a large audience that enjoys unrestricted cinematic experiences.

4. Lower Popularity of Family-Friendly Ratings

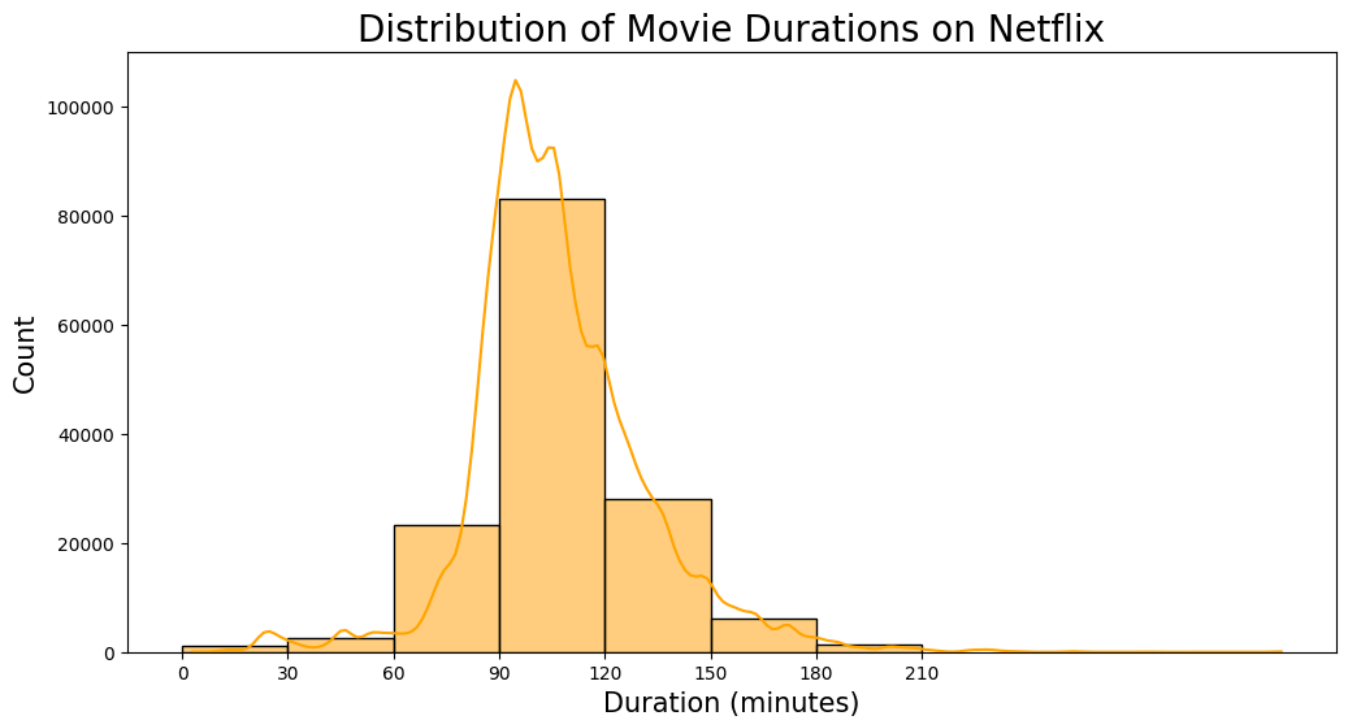
- TV-Y7, TV-Y, TV-G, and NR have significantly fewer titles.
- Family-friendly and kid-specific content makes up a much smaller portion of Netflix's overall catalog.

5. PG & PG-13 Ratings Maintain a Presence

- PG-13 (16,246 titles) and PG (10,919 titles) reflect a moderate share of content that is suitable for a general audience with parental guidance.
- This suggests a balance between child-friendly and mature content.

Distribution of Movie Durations on Netflix

```
movie_duration = nj[nj["type"]=="Movie"]["duration"]
plt.figure(figsize=(12,6))
bins = (0,30,60,90,120,150,180,210)
sns.histplot(movie_duration,bins=bins,kde=True,color="orange")
plt.xticks([0,30,60,90,120,150,180,210])
plt.title("Distribution of Movie Durations on Netflix",fontsize=20)
plt.xlabel("Duration (minutes)",fontsize=15)
plt.ylabel("Count",fontsize=15)
plt.show()
```



1. Most Common Movie Duration (Peak Around 90-120 minutes)

- The histogram shows that the majority of Netflix movies fall in the 90 to 120 minutes range.
- This aligns with standard movie lengths, as most feature films are typically between 1.5 to 2 hours long.

2. Few Short Movies (Below 60 Minutes)

- The leftmost bars indicate that very few movies are shorter than 60 minutes.
- These might be documentaries or special releases.

3. Gradual Decline Beyond 120 Minutes

- Movies longer than 2 hours (120 minutes) are significantly fewer, with a steady decline beyond 150 minutes.
- This suggests that Netflix hosts fewer epic-length movies (e.g., over 3 hours).

4. Smooth Distribution with KDE Curve

- The KDE (Kernel Density Estimation) curve follows a normal-like distribution, peaking at around 100 minutes.
- This confirms that most movies are within the typical runtime range.

Key Business Insights & Recommendations for Netflix

1. Content Distribution Across Countries

Insight:

- The United States dominates Netflix's content library with 67,093 titles, which is significantly higher than any other country.
- India (23,579 titles) and the United Kingdom (14,320 titles) have a strong presence but still lag behind the U.S.
- Other regions such as Japan, France, Canada, and South Korea have relatively smaller libraries.

Recommendations:

- Expand localized content in emerging markets like India, Japan, and South Korea, where streaming growth potential is high.
- Strengthen regional licensing agreements to acquire more country-specific content and attract local audiences.
- Invest in original productions outside the U.S. to improve content diversity and cater to regional preferences.

2. Content Ratings Distribution

Insight:

- TV-MA (73,973 titles) and TV-14 (43,951 titles) dominate the platform, indicating a focus on mature and teenage audiences.

- Family-friendly content like PG, TV-Y, and TV-G has significantly fewer titles.

Recommendations:

- Increase investment in kid-friendly and family content to attract a broader subscriber base, including families.
- Diversify the rating categories by producing more PG and PG-13 content to balance the content portfolio.
- Introduce parental control features to ensure safe viewing for younger audiences.

3. Movie Duration Analysis

Insight:

- Most Netflix movies have a duration between 90–120 minutes, aligning with standard movie lengths.
- Very few movies are shorter than 60 minutes or longer than 150 minutes, indicating a content gap in short films and long-format movies.

Recommendations:

- Experiment with short films and mini-series (30–60 mins) to cater to audiences with limited time.
- Introduce premium long-duration movies (150+ minutes) for viewers who prefer extended storytelling.
- Analyze audience engagement metrics to determine whether shorter or longer films lead to better retention.

4. Genre Distribution by Country

Insight:

- Different countries have varied genre preferences, but Netflix does not have a balanced genre distribution globally.
- Some regions lack diverse content in genres like thrillers, documentaries, and animation.

Recommendations:

- Use data-driven insights to produce region-specific genres that align with user preferences.
- Enhance genre availability across countries to reduce content gaps and improve user engagement.
- Localize content in non-English-speaking countries by investing in subtitles and dubbing.

🎬 The End 🎬