# NetFlix: Business Case Study

## Business Insights and Recommendations based on Netflix data analysis are as follows-

Netflix, Inc. is a technology and media services provider as well as a production company based in Los Gatos, California, USA. Founded in 1997 by Reed Hastings and Marc Randolph in Scotts Valley, California, Netflix primarily operates a subscription-based streaming service. This service allows users to stream a wide variety of films and television series online, including content produced by Netflix itself.

## Business Problem

1. **Defining problem statement and analysing basic metrics**
    a. Importing libraries we needs

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

    b. Loading the data set
    **netflix = pd.read_csv('netflix.csv')**
    **netflix.head()**

```python
netflix = pd.read_csv('netflix.csv')
netflix.head()
```

| | 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... |

    c. Let's check the first 5 rows of Netflix data using this code - **netflix.head()**

```python
netflix.head()
```


| | 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... |

# NetFlix: Business Case Study

```
netflix.tail()
```

| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8802 | s8803 | Movie | Zodiac | David Fincher | Mark Ruffalo, Jake Gyllenhaal, Robert Downey J... | United States | November 20, 2019 | 2007 | R | 158 min | Cult Movies, Dramas, Thrillers | A political cartoonist, a crime reporter and a... |
| 8803 | s8804 | TV Show | Zombie Dumb | NaN | NaN | NaN | July 1, 2019 | 2018 | TV-Y7 | 2 Seasons | Kids' TV, Korean TV Shows, TV Comedies | While living alone in a spooky town, a young g... |
| 8804 | s8805 | Movie | Zombieland | Ruben Fleischer | Jesse Eisenberg, Woody Harrelson, Emma Stone, ... | United States | November 1, 2019 | 2009 | R | 88 min | Comedies, Horror Movies | Looking to survive in a world taken over by zo... |
| 8805 | s8806 | Movie | Zoom | Peter Hewitt | Tim Allen, Courteney Cox, Chevy Chase, Kate Ma... | United States | January 11, 2020 | 2006 | PG | 88 min | Children & Family Movies, Comedies | Dragged from civilian life, a former superhero... |
| 8806 | s8807 | Movie | Zubaan | Mozez Singh | Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan... | India | March 2, 2019 | 2015 | TV-14 | 111 min | Dramas, International Movies, Music & Musicals | A scrappy but poor boy worms his way into a ty... |

The dataset contains over 8807 titles, 12 descriptions. Afer a quick view of the data frames, it looks like a typical movie/TVshows data frame without ranges. We can also see that there are NaN values in some columns.

2. **Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, and statistical summary.**
   a. To get all attributes netflix_df.columns

```
netflix.columns

Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

   b. The shape of data

```
netflix.ndim

2
```

Data types of all attributes could be found using this code –

   a. Netflix.info()

```
netflix.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
```

# NetFlix: Business Case Study

## Statistical summary before data cleaning:

```
netflix.describe()
```

| | release_year |
|---|---|
| count | 8807.000000 |
| mean | 2014.180198 |
| std | 8.819312 |
| min | 1925.000000 |
| 25% | 2013.000000 |
| 50% | 2017.000000 |
| 75% | 2019.000000 |
| max | 2021.000000 |

## Missing value detection data profiling and cleaning

The process of identifying incorrect a partial one inaccurate, irrelevant, or missing data and then altering, updating, or deleting it when required is known as data cleaning. One of the fundamental components of data science is considered to be data cleansing.

print('\nColumns with missing value:')

print(netlix.isnull().any())

```
print('\nColumns with missing value:')
print(netflix.isnull().any())
```

```
Columns with missing value:
show_id          False
type             False
title            False
director          True
cast              True
country           True
date_added        True
release_year     False
rating            True
duration          True
listed_in        False
description      False
dtype: bool
```

From the info, we know that there are 8807 entries and 12 columns to work with for this EDA. There are few columns that contain null values, “director,” “cast,” “country,” “date_added,” “ratng.”

netflix.T.apply(lambda x: x.isnull().sum(), axis = 1)

```
netflix.T.apply(lambda x: x.isnull().sum(), axis = 1)

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
```

netflix_df.isnull().sum().sum()

```
netflix.isnull().sum().sum()

4307
```

There are a total of 4307 null values across the entire dataset with 2634 missing points under "director", 825 under "cast", 831 under "country", 11 under "date_added", 4 under "rating" and 3 under "duration ". We will have to handle all null data points before we can dive into EDA and modelling.

## Imputation is a treatment method for missing value by filling it in using certain techniques.

Can use **mean**, **mode**, or use **predictive modelling**. In this case study, we will discuss the use of the fillna function from Pandas for this imputation. Drop rows containing missing values. Can use the dropna function from Pandas.

netflix.director.fillna("No Director", inplace=True)

netflix.cast.fillna("No Cast", inplace=True)

netflix.country.fillna("Country Unavailable", inplace=True)

netflix.dropna(subset=["date_added", "rating"], inplace=True)

# NetFlix: Business Case Study

## Checking missing value

netflix.isnull().any()

```
netflix.isnull().any()

show_id         False
type            False
title           False
director        False
cast            False
country         False
date_added      False
release_year    False
rating          False
duration         True
listed_in       False
description     False
dtype: bool
```

For missing values, the easiest way to get rid of them would be to delete the rows with the missing data. However, this wouldn't be beneficial to our EDA since there is a loss of information.

Since "director", "cast", and "country" contain the majority of null values, we chose to treat each missing value is unavailable. The other two label "date_added”,” duration” and "rating" contain an insignificant portion of the data so it drops from the dataset. Finally, we can see that there are no more missing values in the data frame.

## Statistical summary after data cleaning:

netflix.describe()

```
netflix.describe()
```

| | release_year |
|---|---|
| count | 8807.000000 |
| mean | 2014.180198 |
| std | 8.819312 |
| min | 1925.000000 |
| 25% | 2013.000000 |
| 50% | 2017.000000 |
| 75% | 2019.000000 |
| max | 2021.000000 |

## 3. Non-graphical analysis

Non-Graphical Analysis involves calculating the summary statistics, without using pictorial or graphical representations. There are 3 main functions that Pandas library provide us, and I will be discussing about them. Those functions are:

1. info()

# NetFlix: Business Case Study

2. isna().sum() or isnull().sum()

3. describe()

## Checking the data using .head()

```
netflix.head()
```

| | 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 l... |

1. .info(): mainly indicates the number of features, non-null count, and data type of each features. Additionally, it also shows the number of features in present in each data type(s). This helps us to determine how many numerical and categorical features we have.

```
netflix.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
```

2. Read the description of the data

```
netflix.describe()
```

| | release_year |
|---|---|
| count | 8807.000000 |
| mean | 2014.180198 |
| std | 8.819312 |
| min | 1925.000000 |
| 25% | 2013.000000 |
| 50% | 2017.000000 |
| 75% | 2019.000000 |
| max | 2021.000000 |

3. isna().sum() or isnull().sum()
   netflix.T.apply(lambda x: x.isnull().sum(), axis = 1)

```
netflix.T.apply(lambda x: x.isnull().sum(), axis = 1)

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
```

## 4. Exploratory analysis and visualization

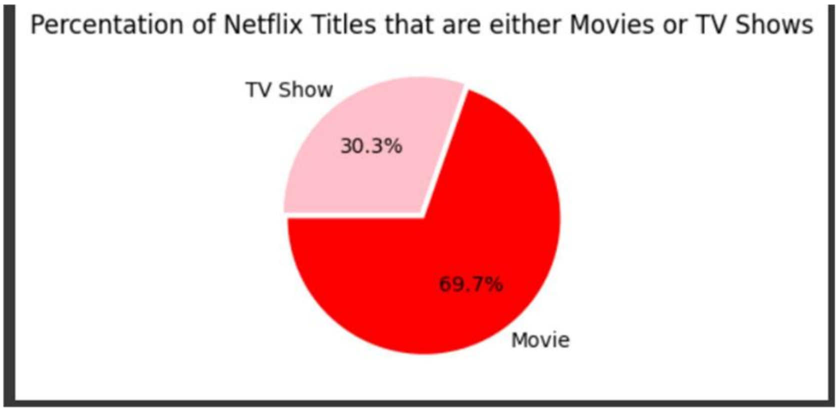
**Visual Analysis - Univariate, Bivariate after pre-processing of the data**

a. **Univariate analysis:** Analysis done based only on one variable. We are not going to focus on math behind these concepts, for now, let's see what these are in graphs. (please have some basic idea on these concepts if you don't get them by seeing graphs).

**A➔ Pie Plot –**

1. **Netflix content by type:**
   Analysis of entire Netflix dataset consisting of both movies and shows.
   Let’s compare the total number of movies and shows in this dataset to know which one is the majority.

   plt.figure(figsize=(6,3))
   plt.title("Percentation of Netflix Titles that are either Movies or TV Shows")
   g=plt.pie(netflix_df.type.value_counts(),explode=(0.025,0.025),
   labels=netflix_df.type.value_counts().index, colors=['red','pink'],autopct='%1.1f%%',
   startangle=180) plt.show()



# NetFlix: Business Case Study

There are far more movie titles (69.7%) that TV shows titles (30.3%) in terms of title.

2. **Amount of content as a function of time: Distplot**
   We will explore the amount of content Netflix has added throughout the previous years. Since we are interested in when Netflix added the title onto their platform, we will add a “year_added” column to show the date from the “date_added” columns.

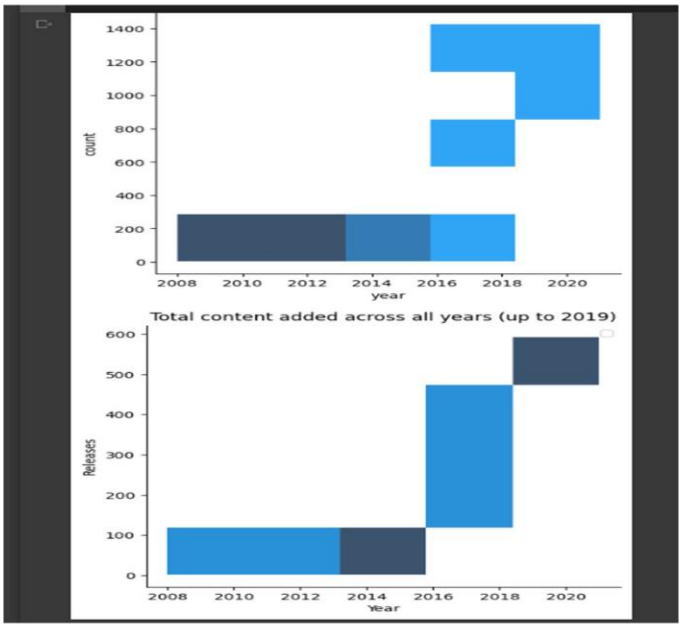
```
netflix["year_added"] = pd.to_date□me(ne□lix_df.date_added).dt.year
netflix_movies["year_added"] = pd.to_datetime(netflix_movies.date_added).dt.year
netflix_shows["year_added"] = pd.to_datetime(netflix_shows.date_added).dt.year
netflix_year =
netflix.year_added.value_counts().to_frame().reset_index().rename(columns={"inde
x": "year",
"year_added":"count"})
netflix_year= netflix_year[netflix_year_df.year != 2020]
print(netflix_year)
```

| | year | count |
|---|---|---|
| 0 | 2019 | 2016 |
| 2 | 2018 | 1648 |
| 3 | 2021 | 1498 |
| 4 | 2017 | 1185 |
| 5 | 2016 | 426 |
| 6 | 2015 | 82 |
| 7 | 2014 | 24 |
| 8 | 2011 | 13 |
| 9 | 2013 | 11 |
| 10 | 2012 | 3 |
| 11 | 2009 | 2 |
| 12 | 2008 | 2 |
| 13 | 2010 | 1 |

```
shows_year_df=netflix_shows_df.year_added.value_counts().to_frame().reset_index().rena
me(columns={"index": "year", "year_added":"count"})
shows_year_df = shows_year_df[shows_year_df != 2020]
shows_year_df
```

| | year | count |
|---|---|---|
| 0 | NaN | 595 |
| 1 | 2019.0 | 592 |
| 2 | 2021.0 | 505 |
| 3 | 2018.0 | 411 |
| 4 | 2017.0 | 349 |
| 5 | 2016.0 | 175 |
| 6 | 2015.0 | 26 |
| 7 | 2014.0 | 5 |
| 8 | 2013.0 | 5 |
| 9 | 2008.0 | 1 |

```
fig, ax = plt.subplots(figsize=(7, 5))
sns.displot(data=netflix_year_df, x='year', y='count')
sns.displot(data=movies_year_df, x='year', y='count')
sns.displot (data=shows_year_df, x='year', y='count')
ax.set_xticks(np.arange(2008, 2020, 1))
plt.title("Total content added across all years (up to 2019)")
plt.legend(['Total','Movie','TV Show'])
plt.ylabel("Releases")
plt.xlabel("Year")
```




Based on the timeline above, we can conclude that the popular streaming platform started gaining traction after 2013. Since then, the amount of content added has been increasing significantly.

The growth in the number of movies on Netflix is much higher than that on TV shows. About 1,300 new movies were added in both 2018 and 2019.
Besides, we can know that Netflix has increasingly focused on movies rather than TV shows in recent years.

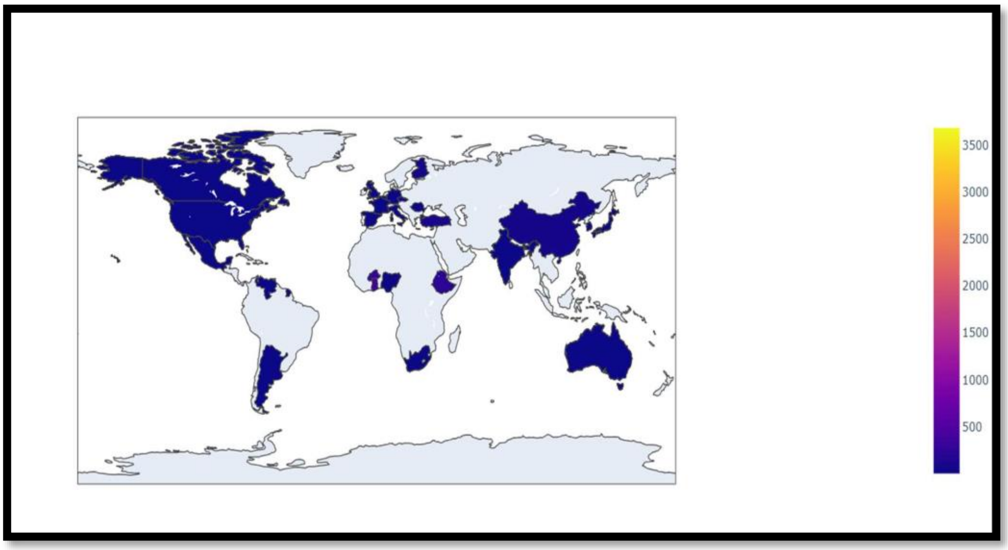
3. **Exploring the countries contribution with the most content of Netflix**
Next is exploring the countries by the amount of the produces content of Netflix. We need to separate all countries within a film before analysing it, then removing titles with no countries available.

```
import plotly.graph_objects as go
from plotly.offline import init_notebook_mode, iplot
```

We need to separate all countries within a film before analyzing it, then removing titles with no countries available.

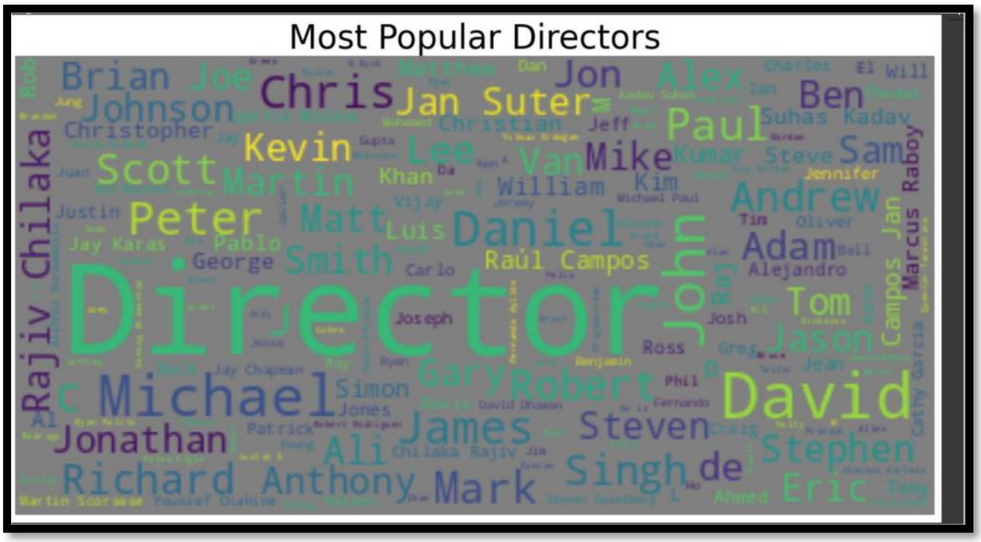
```
filtered_countries = netflix_df.set_index('title').country.str.split(', ',
expand=True).stack().reset_index(level=1, drop=True);
filtered_countries = filtered_countries[filtered_countries != 'Country Unavailable']
iplot([go.Choropleth(
locationmode='country names',
```

```
locations=filtered_countries,
z=filtered_countries.value_counts()
)])
```




4. **Top directors on Netflix**: To know the most popular director, we can visualize it.

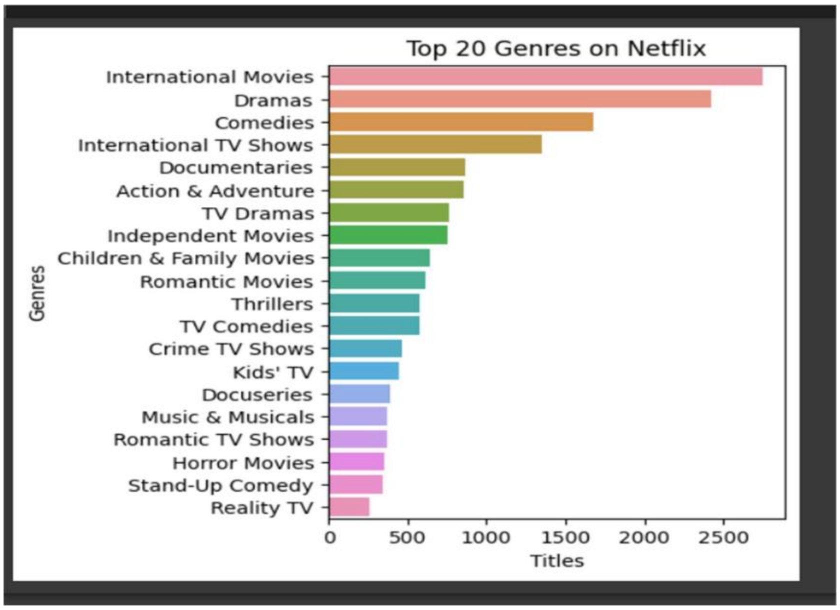
```
from wordcloud import WordCloud, ImageColorGenerator
text = " ".join(str(each) for each in netflix_df.director)
# Create and generate a word cloud image:
wordcloud = WordCloud(max_words=200, background_color="gray").generate(text)
plt.figure(figsize=(10,6))
plt.figure(figsize=(15,10))
# Display the generated image:
plt.imshow(wordcloud, interpolation='Bilinear')
plt.title('Most Popular Directors',fontsize = 30)
plt.axis("off")
plt.show()
```




The most popular director on Netflix, with the most titles, is mainly international.

5. **Top 20 Genres on Netflix: Count Plot**

```
filtered_genres = netflix_df.set_index('title').listed_in.str.split(', ',
expand=True).stack().reset_index(level=1, drop=True);
plt.figure(figsize=(4,5))
g = sns.countplot(y = filtered_genres,
order=filtered_genres.value_counts().index[:20])
plt.title('Top 20 Genres on Netflix')
plt.xlabel('Titles')
plt.ylabel('Genres')
plt.show()
```




From the graph, we know that International Movies take the first place, followed by dramas and comedies.

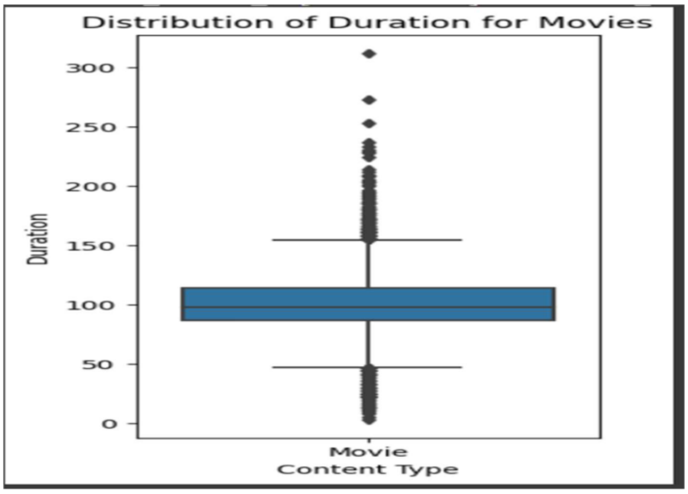
**b. Bivariate analysis:** Bi means two and variate means variable, so here there are two variables. The analysis is related to cause and the relationship between the two variables. There are three types of bivariate analysis.

**A➔** Bivariate Analysis of two Numerical Variables (Numerical-Numerical)

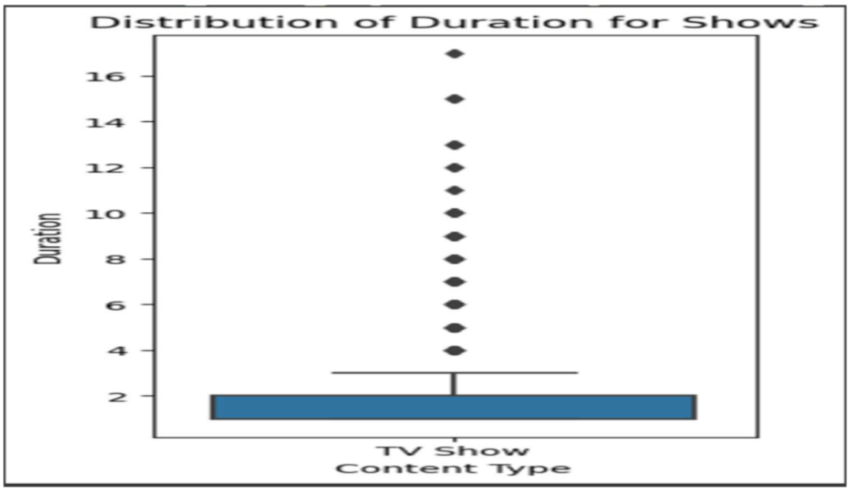
### 4.2. For categorical variable(s): Boxplot

Duration Distribution for Movies and TV Shows: Analysing the duration distribution for movies and TV shows allows us to understand the typical length of content available on Netflix. We can create box plots to visualize these distributions and identify outliers or standard durations.

```
netflix_movies_df = netflix_df[netflix_df.type.str.contains("Movie")]
netflix_movies_df['duration'] = netflix_movies_df['duration'].str.extract('(\d+)',
expand=False).astype(int)

# Creating a boxplot for movie duration
plt.figure(figsize=(10, 6))
sns.boxplot(data=netflix_movies_df, x='type', y='duration')
plt.xlabel('Content Type')
plt.ylabel('Duration')
plt.title('Distribution of Duration for Movies')
plt.show()
```




```
netflix_shows_df = netflix_df[netflix_df.type.str.contains("TV Show")]
netflix_shows_df['duration'] = netflix_shows_df['duration'].str.extract('(\d+)',
expand=False).astype(int)

# Creating a boxplot for movie duration
plt.figure(figsize=(3, 6))
sns.boxplot(data=netflix_shows_df, x='type', y='duration')
plt.xlabel('Content Type')
plt.ylabel('Duration')
plt.title('Distribution of Duration for Shows')
plt.show()
```

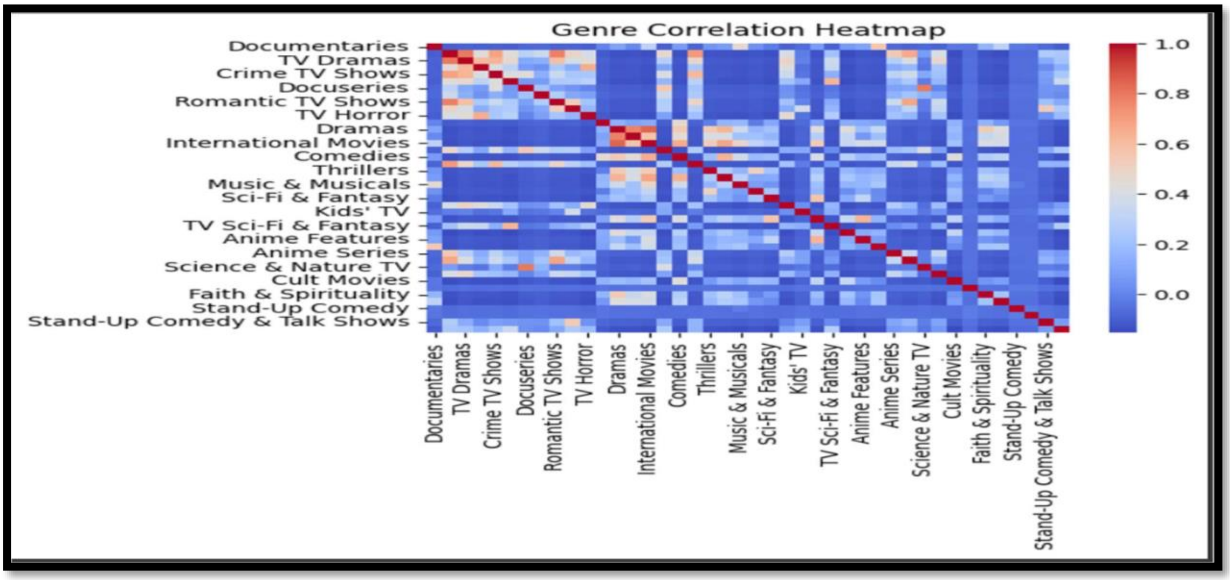



Analysing the movie box plot, we can see that most movies fall within a reasonable duration range, with few outliers exceedingly approximately 2.5 hours. This suggests that most movies on Netflix are designed to fit within a standard viewing time.

For TV shows, the box plot reveals that most shows have one to four seasons, with very few outliers having longer durations. This aligns with the earlier trends, indicating that Netflix focuses on shorter series formats.

### 4.3. For correlation: Heatmaps, Pairplots

**Genre Correlation Heatmap:** Genres play a significant role in categorizing and organizing content on Netflix. Analysing the correlation between genres can reveal interesting relationships between different types of content. We create a genre data DataFrame to investigate genre correlation and fill it with zeros. By iterating over each row in the original DataFrame, we update the genre data DataFrame based on the listed genres. We then create a correlation matrix using this genre data and visualize it as a heatmap.

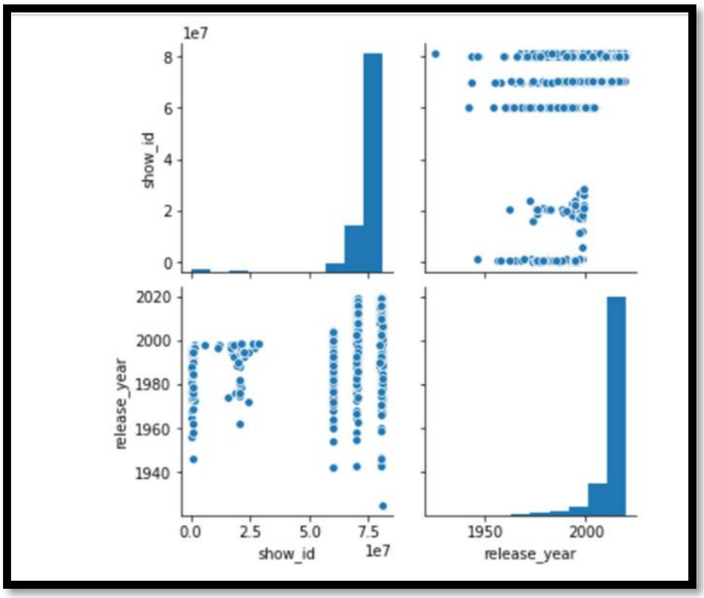



The heatmap demonstrates the correlation between different genres. By analysing the heatmap, we can identify strong positive correlations between specific genres, such as TV Dramas and International TV Shows, Romantic TV Shows, and International TV Shows.

## Pairplots

A pairplot plot a pairwise relationships in a dataset. The pairplot function creates a grid of Axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column.

sns.pairplot(nf_df);



# NetFlix: Business Case Study

5. **Missing value and outlier check (Treatment optional)**

**What is an outliner?**

In a random sampling from a population, an outlier is defined as an observation that deviates abnormally from the standard data. In simple words, an outlier is used to define those data values which are far away from the general values in a dataset. An outlier can be broken down into out-of-line data.

For example, let us consider a row of data [10,15,22,330,30,45,60]. In this dataset, we can easily conclude that 330 is way off from the rest of the values in the dataset, thus 330 is an outlier.

It was easy to figure out the outlier in such a small dataset, but when the dataset is huge, we need various methods to determine whether a certain value is an outlier or necessary information.

**Why do we need to treat outliers?**

Outliers can lead to vague or misleading predictions while using machine learning models. Specific models like linear regression, logistic regression, and support vector machines are susceptible to outliers. Outliers decrease the mathematical power of these models, and thus the output of the models becomes unreliable. However, outliers are highly subjective to the dataset. Some outliers may portray extreme changes in the data as well.

Visual Detection

**Box plots** are a simple way to visualize data through quantiles and detect outliers. IQR(Interquartile Range) is the basic mathematics behind boxplots. The top and bottom whiskers can be understood as the boundaries of data, and any data lying outside it will be an outlier.

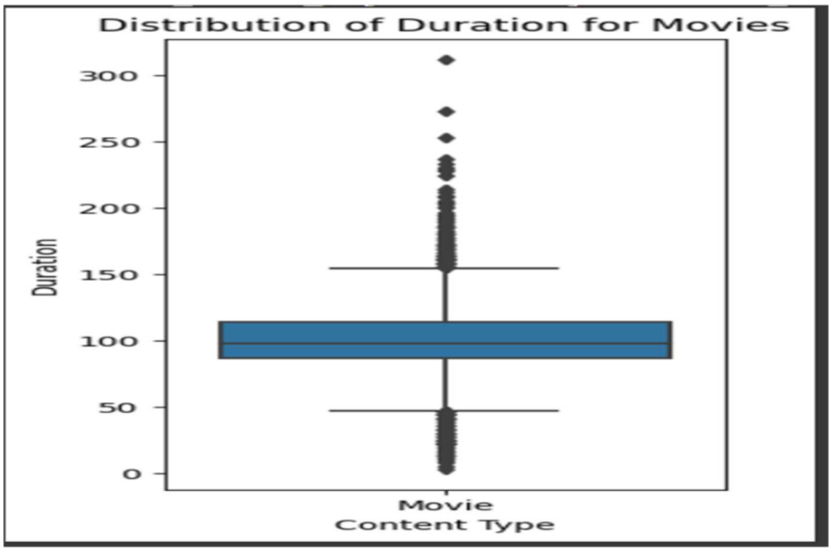
## For categorical variable(s): Boxplot

Duration distribution for Movies and TV Shows

Analysing the duration distribution for movies and TV shows allows us to understand the typical length of content available on Netflix. We can create box plots to visualize these distributions and identify outliers or standard durations.

```
netflix_movies_df = netflix_df[netflix_df.type.str.contains("Movie")]

netflix_movies_df['duration'] = netflix_movies_df['duration'].str.extract('(\d+)',

expand=False).astype(int)

============================================================

# Creating a boxplot for movie duration

plt.figure(figsize=(10, 6))

sns.boxplot(data=netflix_movies_df, x='type', y='duration')

plt.xlabel('Content Type')

plt.ylabel('Duration')

plt.title('Distribution of Duration for Movies')
```

plt.show()




netflix_shows_df = netflix_df[netflix_df.type.str.contains("TV Show")]

netflix_shows_df['duration'] = netflix_shows_df['duration'].str.extract('(\d+)',

expand=False).astype(int)

---

# Creating a boxplot for movie duration

plt.figure(figsize=(3, 6))

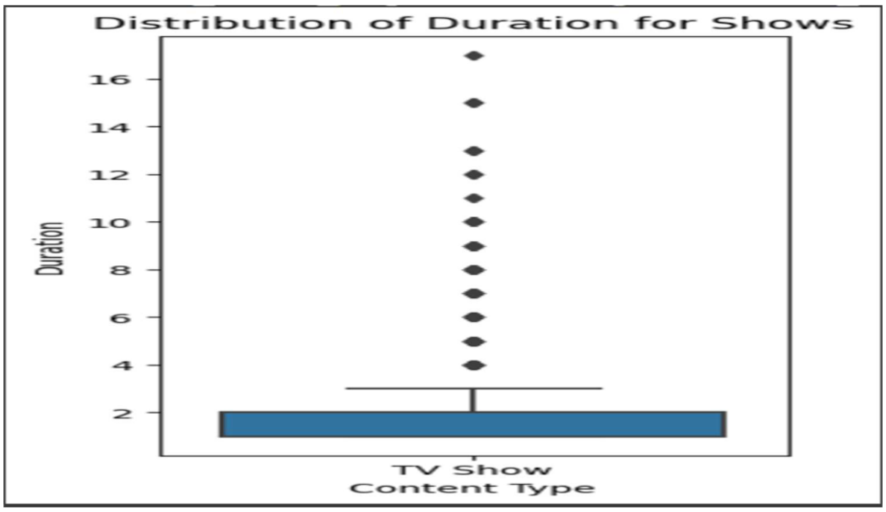
sns.boxplot(data=netflix_shows_df, x='type', y='duration')

plt.xlabel('Content Type')

plt.ylabel('Duration')

plt.title('Distribution of Duration for Shows')

plt.show()



Analysing the movie box plot, we can see that most movies fall within a reasonable duration range, with few outliers exceedingly approximately 2.5 hours. This suggests that most movies on Netflix are designed to fit within a standard viewing time.

For TV shows, the box plot reveals that most shows have one to four seasons, with very few outliers having longer durations. This aligns with the earlier trends, indicating that Netflix focuses on shorter series formats.

What are Missing values?

In a dataset, we often see the presence of empty cells, rows, and columns, also referred to as Missing values. They make the dataset inconsistent and unable to work on. Many machine learning algorithms return an error if parsed with a dataset containing null values. Detecting and treating missing values is essential while analyzing and formulating data for any purpose.

**Detecting missing values**

There are several ways to detect missing values in Python. isnull() function is widely used for the same purpose.

**dataframe.isnull().values.any()** allows us to find whether we have any null values in the dataframe.

print('\nColumns with missing value:')

print(ne�lix_df.isnull().any())

```
print('\nColumns with missing value:')
print(netflix_df.isnull().any())

Columns with missing value:
show_id          False
type             False
title            False
director          True
```

From the info, we know that there are 8807 entries and 12 columns to work with for this EDA. There are a few columns that contain null values, “director,” “cast,” “country,” “date_added,” “ra�ng.”

**dataframe.isnull().sum()** this function displays the total number of null values in each column.

netflix_df.T.apply(lambda x: x.isnull().sum(), axis = 1)

```
netflix.T.apply(lambda x: x.isnull().sum(), axis = 1)

show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        3
listed_in       0
description     0
year_added      0
dtype: int64
```

# NetFlix: Business Case Study

netflix_df.isnull().sum().sum()

4307

There are a total of 4307 null values across the entire dataset with 2634 missing points under "director", 825 under "cast", 831 under "country", 11 under "date_added", 4 under "rating" and 3 under “duration ”. We will have to handle all null data points before we can dive into EDA and modelling.

### Remedies to the outliers and missing values

Imputation is a treatment method for missing value by filling it in using certain techniques. Can use mean, mode, or use predictive modelling. In this case study, we will discuss the use of the fillna function from Pandas for this imputation. Drop rows containing missing values. Can use the dropna function from Pandas.

netflix_df.director.fillna("No Director", inplace=True)

netflix_df.cast.fillna("No Cast", inplace=True)

netflix_df.country.fillna("Country Unavailable", inplace=True)

netflix_df.dropna(subset=["date_added", "rating"], inplace=True)

```
netflix_df.isnull().any()

show_id          False
type             False
title            False
director         False
cast             False
country          False
date_added       False
release_year     False
rating           False
duration         False
listed_in        False
description      False
dtype: bool
```

For missing values, the easiest way to get rid of them would be to delete the rows with the missing data. However, this wouldn't be beneficial to our EDA since the is a loss of information. Since "director", "cast", and "country" contain the majority of null values, we chose to treat each missing value is unavailable.

The other two label "date_added”,” duration” and "rating" contain an insignificant portion of the data so it drops from the dataset. Finally, we can see that there are no more missing values in the data frame.

## Business insights:

1. Quantity: Our analysis revealed that Netflix had added more movies than TV shows, aligning with the expectation that movies dominate their content library.
2. Content Addition: July emerged as the month when Netflix adds the most content, closely followed by December, indicating a strategic approach to content release.
3. Genre Correlation: Strong positive associations were observed between various genres, such as TV dramas and international TV shows, romantic and international TV shows, and independent movies and dramas. These correlations provide insights into viewer preferences and content interconnections.

4. Movie Lengths: The analysis of movie durations indicated a peak around the 1960s, followed by a stabilization around 100 minutes, highlighting a trend in movie lengths over time.
5. TV Show Episodes: Most TV shows on Netflix have one season, suggesting a preference for shorter series among viewers.
6. Common Themes: Words like love, life, family, and adventure were frequently found in titles and descriptions, capturing recurring themes in Netflix content.
7. Rating Distribution: The distribution of ratings over the years offers insights into the evolving content landscape and audience reception.
8. Data-Driven Insights: Our data analysis journey showcased the power of data in unravelling the mysteries of Netflix's content landscape, providing valuable insights for viewers and content creators.
9. Continued Relevance: As the streaming industry evolves, understanding these patterns and trends becomes increasingly essential for navigating the dynamic landscape of Netflix and its vast library.
10. Happy Streaming: We hope this blog has been an enlightening and entertaining journey into the world of Netflix, and we encourage you to explore the captivating stories within its ever-changing content offerings. Let the data guide your streaming adventures.

## Recommendations

1. Netflix has to focus on TV Shows also because there are people who will like to see tv shows rather than movies
2. By approaching the top director we can plan some more movies/tv shows in order to increase the popularity
3. Not only reaching top director we can also see the director with less no of movies and having high rating as there may be some financial
4. Issues or anything so in order to get good content netflix can reach to them and netflix can produce the movie and give the director a chance.
5. We have seen most no of international movies genre so need to give priority to other genres like horror, comedy..etc.
6. In TV Shows we may focus on thriller genre which will be helpfull for having more no of seasons
7. Most of the movies released on ott platform is in a year 2019 so we need to go on increasing this value in order to attract people by showing that getting subscription is useful as netflix is releasing more movies per year.
8. Mainly the release on ott platform should focus on the festival holidays, year-end and weekends which is to be mainly focussed.
9. Some movies can be released directly into ott which has some positive talk which may help in improving subscriptions Should focus on a actor who has immense following and make use of it by doing a TV Shows or web series.
10. Advertisement in the country which has very less movies released should be increased and attract people of that country by making their native TV Shows

## Additional Considerations:

1. Regional differences: Analyse data by region to understand specific preferences and tailor content accordingly.
2. Content quality: Ensure quality remains a top priority alongside catering to popular genres.
3. Competition: Monitor competitor offerings and adjust strategies based on market trends.