PROBLEM STATEMENT

Netflix has 222M subscribers and has a large content library, They want to know :

1. Which type of shows or movies/shows/genre/ratings to focus on
2. How to stratigically expand across different countries

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

df = pd.read_csv('/content/netflix.csv')
df.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...

```
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

```
df.shape
```

```
(8807, 12)
```

Data set consist of 8807 rows and 12 columns

```
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
```

Converting data types :

convert date added to datetime :

```
df["date_added"] = pd.to_datetime(df["date_added"], errors='coerce', format='mixed')
```

Convert categorical columns to category :

```
categorical_cols = ['type','country','rating','listed_in']
for col in categorical_cols:
  df[col]=df[col].astype('category')
```

Extract year and month from date added :

```
df['year_added']=df['date_added'].dt.year
df['month_added']=df['date_added'].dt.month
```

```
# Statistical summary :

print("/statistical summary (Numerical columns):")
print(df.describe())
```

```
/statistical summary (Numerical columns):
                          date_added  release_year   year_added  month_added
count                           8797   8807.000000  8797.000000  8797.000000
mean   2019-05-17 05:59:08.436967168   2014.180198  2018.871888     6.654996
min              2008-01-01 00:00:00   1925.000000  2008.000000     1.000000
25%              2018-04-06 00:00:00   2013.000000  2018.000000     4.000000
50%              2019-07-02 00:00:00   2017.000000  2019.000000     7.000000
75%              2020-08-19 00:00:00   2019.000000  2020.000000    10.000000
max              2021-09-25 00:00:00   2021.000000  2021.000000    12.000000
std                              NaN      8.819312     1.574243     3.436554
```

```
# categorical summary :

print("/statistical summary (Categorical columns):")
print(df.describe(include="category"))
```

```
/statistical summary (Categorical columns):
         type        country rating                     listed_in
count    8807           7976   8803                          8807
unique      2            748     17                           514
top     Movie  United States  TV-MA  Dramas, International Movies
freq     6131           2818   3207                           362
```

Non Graphical Analysis :

```
print("Movies vs TV shows :")
print(df['type'].value_counts())
```

```
Movies vs TV shows :
type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

```
# Number of unique values in each columns :

print("Unique values in each column:")
for col in df.columns:
  print(f"{col} : {df[col].nunique()}")
```

```
Unique values in each column:
show_id : 8807
type : 2
title : 8807
director : 4528
cast : 7692
country : 748
date_added : 1714
release_year : 74
rating : 17
duration : 220
listed_in : 514
description : 8775
year_added : 14
month_added : 12
```

```
# Top 10 countries producing contents :

print("Top 10 countries producing contents:")
print(df['country'].value_counts().head())
```

```
Top 10 countries producing contents:
country
United States     2818
India              972
United Kingdom     419
```

```
Japan              245
South Korea        199
Name: count, dtype: int64
```

```
# Top 10 directors :

print("Top 10 directors:")
print(df['director'].value_counts().head())
```

```
Top 10 directors:
director
Rajiv Chilaka             19
Raúl Campos, Jan Suter    18
Suhas Kadav               16
Marcus Raboy              16
Jay Karas                 14
Name: count, dtype: int64
```

```
# Top 10 actors :

print("Top 10 actors:")
print(df["cast"].value_counts().head())
```

```
Top 10 actors:
cast
David Attenborough                                                                                          19
Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava, Mousam, Swapnil                     14
Samuel West                                                                                                 10
Jeff Dunham                                                                                                  7
Michela Luci, Jamie Watson, Eric Peterson, Anna Claire Bartlam, Nicolas Aqui, Cory Doran, Julie Lemieux, Derek McGrath     6
Name: count, dtype: int64
```

```
# Splitting "cast" columns

actors = df['cast'].dropna().str.split(',')
actors = actors.explode()

print('Top 10 actors:')
actors.value_counts().head()
```

```
Top 10 actors:
                    count
       cast
  Anupam Kher          39
  Rupa Bhimani         31
Takahiro Sakurai       30
  Julie Tejwani        28
     Om Puri           27

dtype: int64
```

```
# Top Rated Movies:

print('Top Rated Movies:')
print(df['rating'].value_counts().head())
```

```
Top Rated Movies:
rating
TV-MA    3207
TV-14    2160
TV-PG     863
R         799
PG-13     490
Name: count, dtype: int64
```

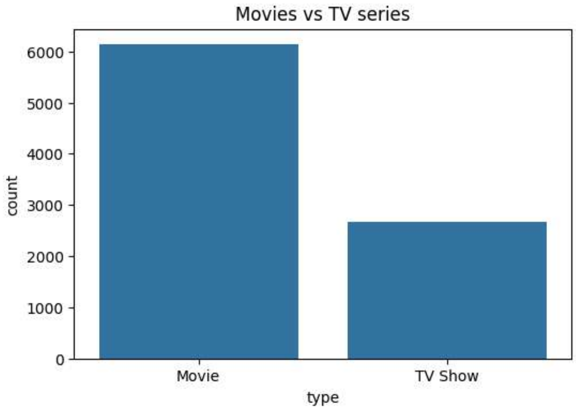
```
# Release per year :

print("Release per year:")
print(df['release_year'].value_counts().head())
```

```
Release per year:
release_year
2018    1147
2017    1032
2019    1030
2020     953
2016     902
Name: count, dtype: int64
```

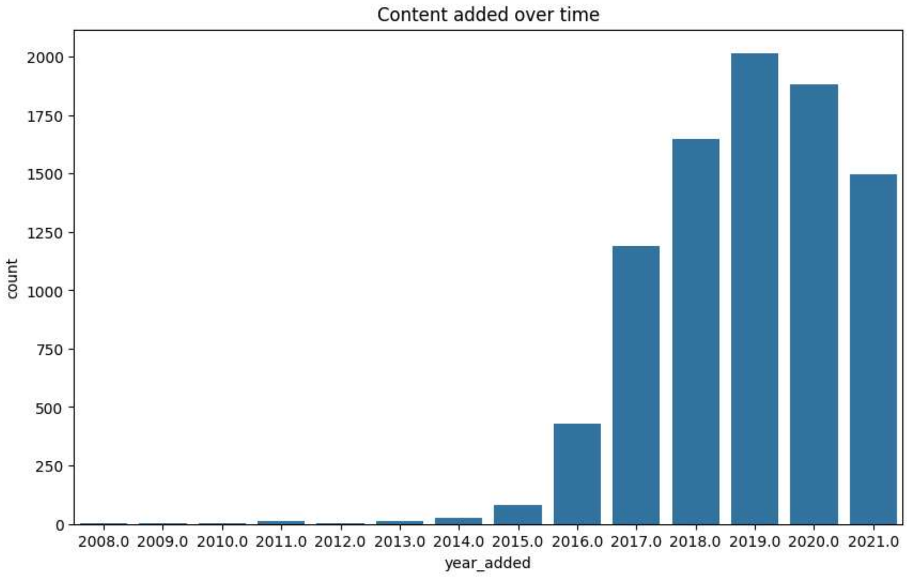
```
# Release year range :

print("Release year Range:")
print("Earlier Year:",df["release_year"].min())
print("later Year:",df["release_year"].max())
```

```
Release year Range:
Earlier Year: 1925
later Year: 2021
```

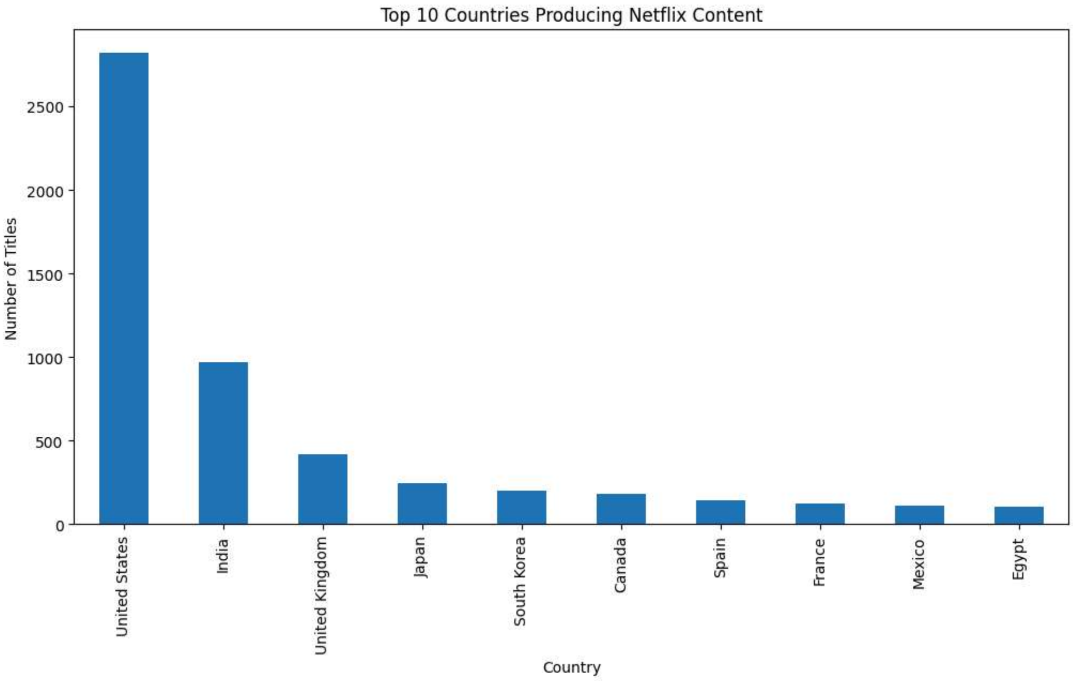
```
# Visuals:

# Line plot of releases per year :
# Movies vs TV series:

plt.figure(figsize=(6,4))
sns.countplot(x='type',data=df)
plt.title("Movies vs TV series")
plt.show()
```



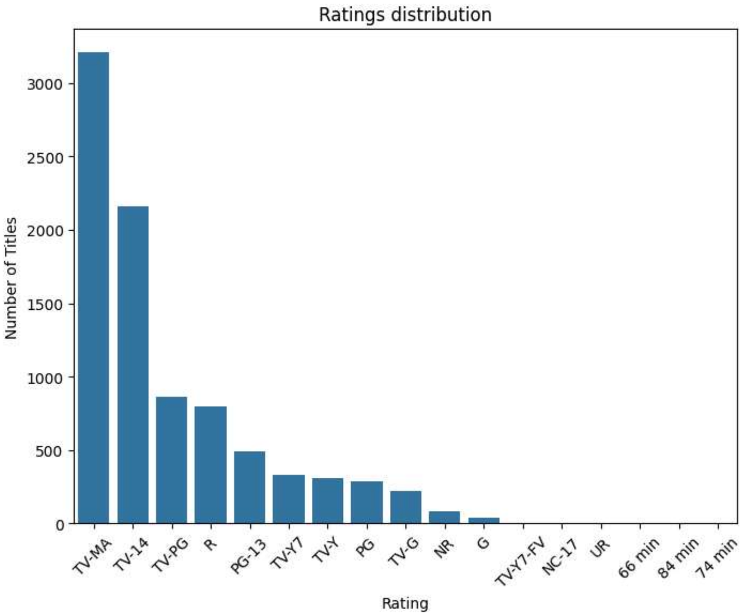
```
# Content added over time :

plt.figure(figsize=(10,6))
sns.countplot(x="year_added",data=df)
plt.title("Content added over time")
plt.show()
```



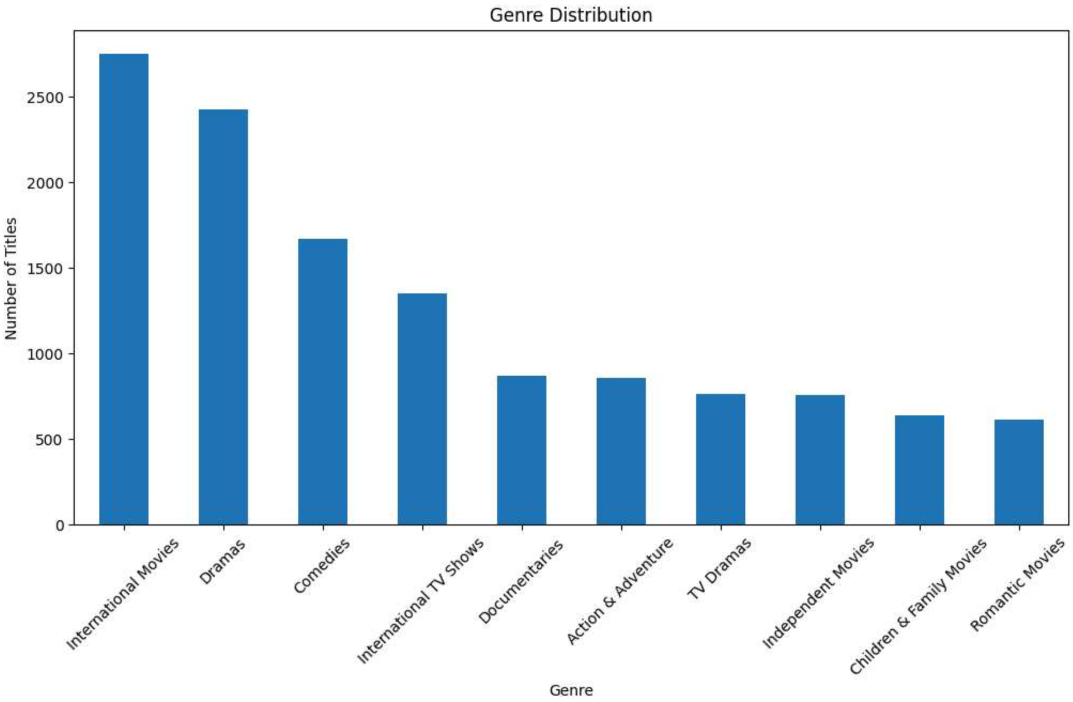
```
# Top 10 countries producing content:


plt.figure(figsize=(12,6))
df['country'].value_counts().head(10).plot(kind='bar')
plt.title("Top 10 Countries Producing Netflix Content")
plt.xlabel("Country")
plt.ylabel("Number of Titles")
plt.show()
```



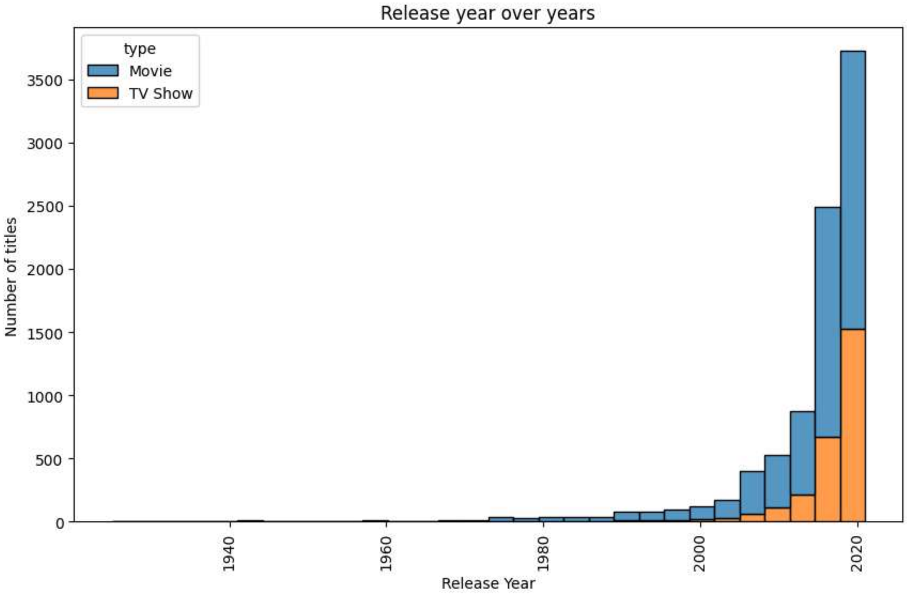
```
# Ratings distribution :

plt.figure(figsize=(8,6))
sns.countplot(x='rating',data=df,order=df['rating'].value_counts().index)
plt.title("Ratings distribution")
plt.xlabel("Rating")
plt.xticks(rotation=45)
plt.ylabel("Number of Titles")
plt.show()
```



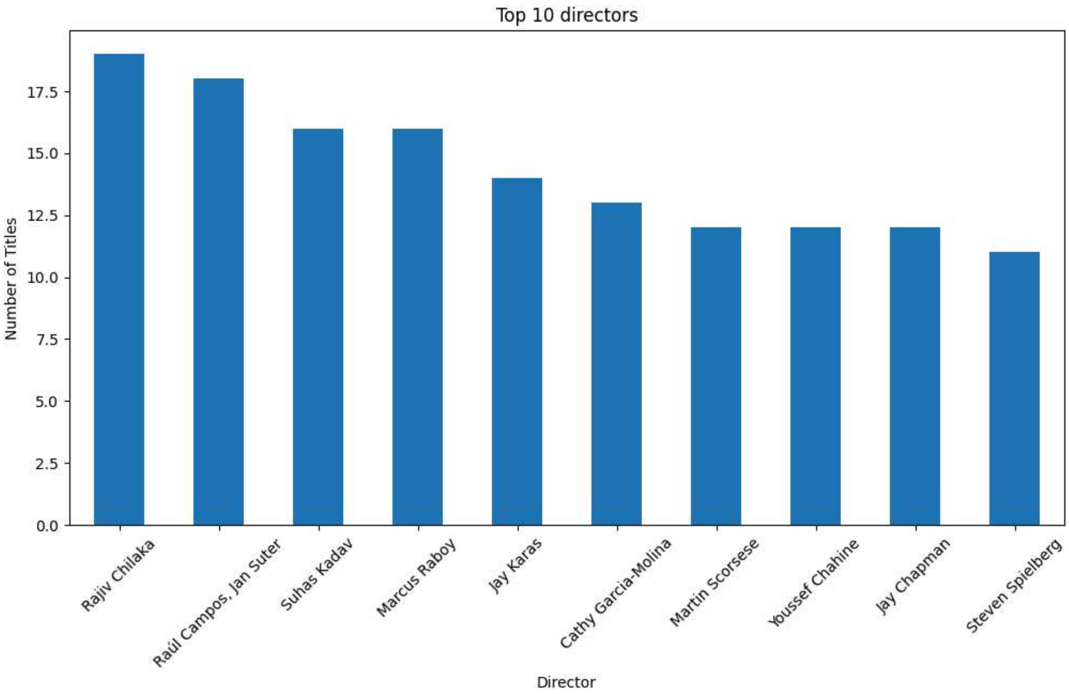
```
# Genre Distribution :

plt.figure(figsize=(12,6))
genres = df['listed_in'].str.split(', ').explode()
genres.value_counts().head(10).plot(kind='bar')
plt.title("Genre Distribution")
plt.xlabel("Genre")
plt.xticks(rotation=45)
plt.ylabel("Number of Titles")
plt.show()
```



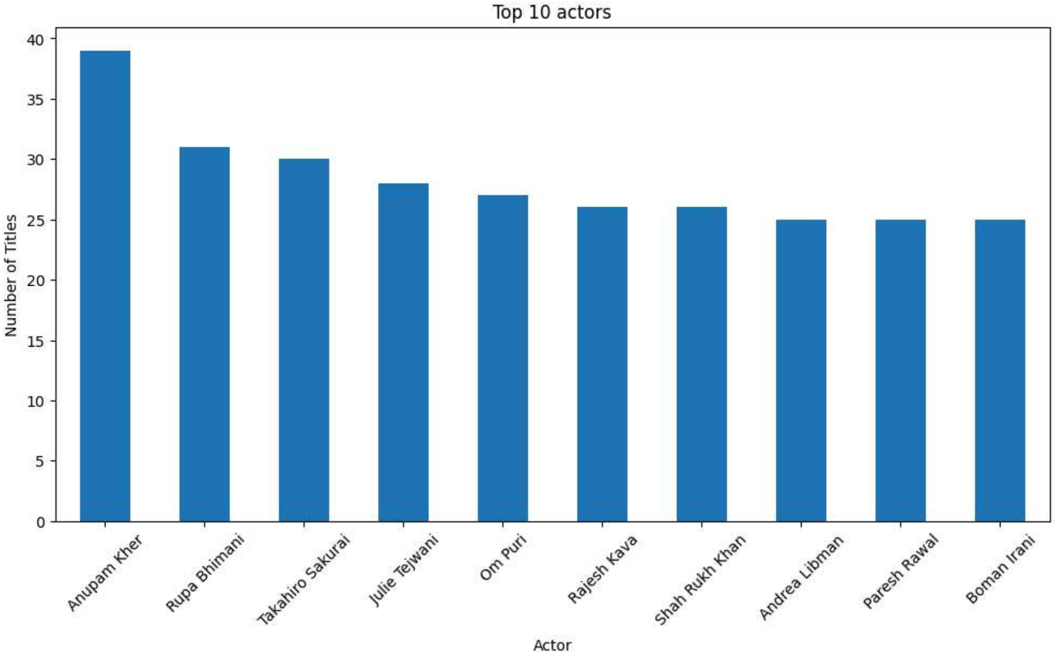
```
# Release year trend over years :

plt.figure(figsize=(10,6))
sns.histplot(data=df,x='release_year', hue='type',bins=30,multiple='stack')
plt.xlabel("Release Year")
plt.xticks(rotation=90)
plt.ylabel("Number of titles")
plt.title("Release year over years")
plt.show()
```



```
# Top 10 directors :

plt.figure(figsize=(12,6))
df['director'].value_counts().head(10).plot(kind='bar')
plt.title('Top 10 directors')
plt.xlabel('Director')
plt.xticks(rotation=45)
plt.ylabel('Number of Titles')
plt.show()
```



```
# Top 10 actors :

plt.figure(figsize=(12,6))
actors = df['cast'].str.split(',').explode()
actors.value_counts().head(10).plot(kind='bar')
plt.title('Top 10 actors')
plt.xlabel('Actor')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.show()
```



INSIGHTS

1. Netflix produces movies more than TV shows, total number of movies produced are 6131 and TV shows are 2676.
2. Top countries who produce most number of movies are USA and India with 2818 and 972 movies respectively.
3. Rajiv Chilaka and Raul Campos, Jan Suter are the top directors who has created most number of movies with 19 and 18 respectively.
4. Anupam Kher, Rupa Bhimani and Takahiro Sakurai are the actors who acted in most number of movies with 39,31,30 respectively.

5. TV-MA, TV-14 are the top rated categories with 3207,2160.
6. Most number 0f movies were released in 2018 and 2017 with 1147 and 1032 respectively.

RECOMMENDATIONS :

1. International movies are made the most so invest in regional language(India, Japan, Korea etc) movies as well for expansion of business.
2. Number of TV shows produced is very less when compared to movies so focus more in TV shows for customer retention, but invest in blockbuster movies for acquisition.
3. Children and family movies are produced very less so create more kids movies and TV shows to compete with Disney+ on Kids content.
4. Colab with regional directors and actors who are popular in demand.
5. Expand genres like documentaries and Action & Adventure which has decent demand.
6. Demand for Dramas and Comedies are high so invest more in those genres.