NAME:AYUSH SAINI BATCH:DSML Link:https://colab.research.google.com/drive/1pJQKPo-0A2XLOdfYoDrcVuuRxsDe9kYk?usp=sharing

```
import pandas as pd

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

{"summary":"{\n \"name\": \"display(df\",\n \"rows\": 5,\n \"fields\": [\n {\n \"column\": \"show_id\",\n \"properties\": {\n \"dtype\": \"string\",\n \"num_unique_values\": 5,\n \"samples\": [\n \"s2\",\n \"s5\",\n \"s3\"\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"type\",\n \"properties\": {\n \"dtype\": \"category\",\n \"num_unique_values\": 2,\n \"samples\": [\n \"TV Show\",\n \"Movie\"\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"title\",\n \"properties\": {\n \"dtype\": \"string\",\n \"num_unique_values\": 5,\n \"samples\": [\n \"Blood & Water\",\n \"Kota Factory\"\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"director\",\n \"properties\": {\n \"dtype\": \"category\",\n \"num_unique_values\": 2,\n \"samples\": [\n \"Julien Leclercq\",\n \"Kirsten Johnson\"\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"cast\",\n \"properties\": {\n \"dtype\": \"string\",\n \"num_unique_values\": 3,\n \"samples\": [\n \"Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Greeff, Xolile Tshabalala, Getmore Sithole, Cindy Mahlangu, Ryle De Morny, Greteli Fincham, Sello Maake Ka-Ncube, Odwa Gwanya, Mekaila Mathys, Sandi Schultz, Duane Williams, Shamilla Miller, Patrick Mofokeng\",\n \"Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabiha Akkari, Sofia Lesaffre, Salim Kechiouche, Noureddine Farihi, Geert Van Rampelberg, Bakary Diombera\"\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"country\",\n \"properties\": {\n \"dtype\": \"string\",\n \"num_unique_values\": 3,\n \"samples\": [\n \"United States\",\n \"South Africa\"\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"date_added\",\n \"properties\": {\n \"dtype\": \"object\",\n \"num_unique_values\": 2,\n \"samples\": [\n \"September 24, 2021\",\n \"September 25, 2021\"\n],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"release_year\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\":

0,\n   \"min\": 2020,\n   \"max\": 2021,\n
\"num_unique_values\": 2,\n   \"samples\": [\n   2021,\n
2020\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"rating\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 2,\n   \"samples\":
[\n   \"TV-MA\",\n   \"PG-13\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"duration\",\n   \"properties\":
{\n   \"dtype\": \"string\",\n   \"num_unique_values\": 3,\n
\"samples\": [\n   \"90 min\",\n   \"2 Seasons\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"listed_in\",\n
\"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 5,\n   \"samples\": [\n
\"International TV Shows, TV Dramas, TV Mysteries\",\n
\"International TV Shows, Romantic TV Shows, TV Comedies\"\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"description\",\n   \"properties\": {\n   \"dtype\":
\"string\",\n   \"num_unique_values\": 5,\n   \"samples\":
[\n   \"After crossing paths at a party, a Cape Town teen sets
out to prove whether a private-school swimming star is her sister who
was abducted at birth.\",\n   \"In a city of coaching centers
known to train India\\u2019s finest collegiate minds, an earnest but
unexceptional student and his friends navigate campus life.\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe"}


```
df.info()
df.isnull().sum()
```

```
<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
```

```
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
```

1.Director has heavy missing values

2.Cast and Country have moderate missing values

3.Date_added, rating, duration have very few missing values

```
df['director'] = df['director'].fillna('Unknown')
df['cast'] = df['cast'].fillna('Unknown')
df['country'] = df['country'].fillna('Unknown')
df = df.dropna(subset=['date_added'])
df['rating'] = df['rating'].fillna(df['rating'].mode()[0])
df['duration'] = df['duration'].fillna(df['duration'].mode()[0])

print("Null values after cleaning:")
display(df.isnull().sum())
```

```
Null values after cleaning:
```

```
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
```

Q1.Defining Problem Statement and Analysing basic metrics

```
df.shape[0]
```

```
8797
```

number of rows in the dataset

```
df.shape[1]
```

```
12
```

number of columns in the dataset

```
print(f"Number of unique countries producing content:
{df['country'].nunique()}")
```

```
Number of unique countries producing content: 749
```

```
print(f"Number of unique genres: {df['listed_in'].nunique()}")
```

```
Number of unique genres: 513
```

Countries and genres contain multiple comma-separated values, so exact unique counts are not directly meaningful.

```
min_release_year = df['release_year'].min()
max_release_year = df['release_year'].max()

print(f"Release years range from {min_release_year} to
{max_release_year}")
```

```
Release years range from 1925 to 2021
```

Basic Metrics

1. in the dataset: 8797

2.Total number of attributes: 12

3.Content types available: Movies and TV Shows

4.Release years covered: 1925 to 2021

5.Number of unique countries producing content: 749

6.Number of unique genres: 513

7.nunique() is not real countries/genres, it counts combinations.

Initial Observations

1.The dataset represents a large and diverse Netflix content catalog.

2.Both movies and TV shows are present, indicating a mixed content strategy.

3.Content spans nearly a century, allowing trend analysis across decades.

4.A high number of countries and genres suggests Netflix operates with a strong global and multi-genre content acquisition approach.

Why this matters for business

1.Global country coverage indicates opportunities for region specific content investments.

2.Long release year span enables understanding of modern vs classic content demand.

3.High genre diversity helps identify which genres to prioritize for production.

2. Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary.

```
df.shape
```

```
(8797, 12)
```

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

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 8797 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8797 non-null   object
 1   type          8797 non-null   object
 2   title         8797 non-null   object
 3   director      8797 non-null   object
 4   cast          8797 non-null   object
 5   country       8797 non-null   object
 6   date_added    8709 non-null   datetime64[ns]
 7   release_year  8797 non-null   int64
 8   rating        8797 non-null   object
 9   duration      8797 non-null   object
 10  listed_in     8797 non-null   object
 11  description   8797 non-null   object
dtypes: datetime64[ns](1), int64(1), object(10)
memory usage: 893.4+ KB
```

Only,release_year is int64 rest all the attributes are object or string data types

```
df['type'] = df['type'].astype('category')
df['rating'] = df['rating'].astype('category')
```

I am converting attributes like type and rating to category because they contain a limited and repetitive set of labels suitable for category data type.

Missing values Most columns in the dataset, including show_id, type, title, director, cast, country, release_year, rating, duration, listed_in, and description, have **0** missing values, indicating that they are complete.

The only column with missing values is date_added, which has **88** missing entries.

Missing values have been filtered at the second code block.

```
df.describe(include='all')
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 11,\n  \"fields\": [\n
{\n      \"column\": \"show_id\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"8797\",\n          \"s8807\",\n
\"1\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"type\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n          2,\n
\"6131\",\n          \"8797\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"title\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"8797\",\n          \"Zubaan\",\n          \"1\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"director\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n          4529,\n
\"2624\",\n          \"8797\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"cast\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          7683,\n          \"825\",\n          \"8797\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"country\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          749,\n          \"2812\",\n          \"8797\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"date_added\",\n      \"properties\": {\n        \"dtype\":
\"date\",\n        \"min\": \"1970-01-01 00:00:00.000008709\",\n
\"max\": \"2021-09-25 00:00:00\",\n        \"num_unique_values\": 7,\n
\"samples\": [\n          \"8709\",\n          \"2019-05-23
01:45:29.452290560\",\n          \"2020-08-26 00:00:00\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"release_year\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2598.7098254713346,\n        \"min\": 8.822190519562549,\n
\"max\": 8797.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          2014.1834716380583,\n          2019.0,\n
```

```
8797.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"rating\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 4,\n        \"samples\": [\n          17,\n          \"3209\",\n          \"8797\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"duration\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 4,\n        \"samples\": [\n          220,\n          \"1796\",\n          \"8797\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"listed_in\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 4,\n        \"samples\": [\n          513,\n          \"362\",\n          \"8797\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"description\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 4,\n        \"samples\": [\n          8765,\n          \"4\",\n          \"8797\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

Observations 1.The dataset has 8797 s (which means 8797 different entries) and 12 s (which are 12 different pieces of information for each entry).

2.Each entry in this dataset is about one unique movie or TV show that Netflix offers.

Data types 1.'type' and 'rating' are like categories (for example, 'Movie' or 'TV Show', or 'TV-MA'). They have a fixed, small number of options.

2.'title', 'director', 'cast', 'country', 'listed_in', and 'description' are all pieces of text, like names, descriptions, or lists of genres.

3.'date_added' stores dates (like when a show was added to). We changed it into a special 'datetime' format so we can easily look at trends over time.

Missing value treatment

1.For 'director', 'cast', and 'country' information that was missing, we just wrote 'Unknown'. This way, we didn't lose any records and can still analyze who's involved and where content is from.

2.We removed any where the 'date_added' was missing because we need correct dates to understand how content has been added over time.

3.For 'rating' and 'duration' missing values, we filled them in with the most common value for those categories.

4.After these steps, our All major missing values were treated. Only 88 date values remain missing and were excluded from time-based analysis..

Statistical summary 1.The content ranges from really old (1925) to very new (2021), meaning Netflix has both classic and modern shows.

2.Most of the information we have (like titles, genres, etc.) is text or categories, which fits because we're describing content.

Business relevance

1.By fixing the missing director and cast names, we can reliably look at who makes content and see patterns in creators.

2.Having clean date information means we can accurately track how Netflix's content library has grown over time.

3.Getting the data types right makes it easier and faster to create visuals and analyze the information.

3. Non-Graphical Analysis: Value counts and unique attributes

```
df['type'].value_counts()
```

```
type
Movie      6131
TV Show    2666
Name: count, dtype: int64
```

number of counts for Movie and TV Show

```
df['rating'].value_counts().head(5)
```

```
rating
TV-MA    3209
TV-14    2157
TV-PG     861
R         799
PG-13     490
Name: count, dtype: int64
```

Number of counts for various rating categories

```
df['country'].value_counts().head(5)
```

```
country
United States     2812
India              972
Unknown            830
United Kingdom     418
Japan              244
Name: count, dtype: int64
```

number of counts for various countries as,it counts combinations.

```
df['listed_in'].value_counts().head(5)
```

```
listed_in
Dramas, International Movies                        362
Documentaries                                       359
Stand-Up Comedy                                     334
Comedies, Dramas, International Movies              274
Dramas, Independent Movies, International Movies    252
Name: count, dtype: int64
```

Number of counts for various genres

```
df['director'].value_counts().head(5)
```

```
director
Unknown                   2624
Rajiv Chilaka               19
Raúl Campos, Jan Suter      18
Suhas Kadav                 16
Marcus Raboy                16
Name: count, dtype: int64
```

Number of counts of various directors

Observations 1.More Movies than TV Shows: Netflix has more movies available than TV shows. Mostly for Grown-ups: Most of the content on Netflix is rated for older teens and adults (like TV-MA, TV-PG, and TV-14).

2.Worldwide Reach: Netflix is very popular all over the world, with content from many different countries.

3.Lots of Different Kinds of Shows: Netflix offers a huge variety of show and movie types, so there's something for everyone.

4.Many Creators Work with Netflix: A high number of directors work with Netflix, showing they have strong partnerships with creative people.

Business interpretation

1.Movies First, Now TV Shows Too: Netflix used to have many more movies, which shows they focused a lot on films. We can check later to see how this compares to how many TV shows they're adding now.

2.Mostly for Grown-ups: Because most content is rated for mature audiences, it means Netflix mostly aims its shows and movies at adults.

3.Lots of Countries, Lots of Opportunities: Since Netflix has content from all over the world, it has a great chance to create more local shows and movies that appeal to people in different countries.

4.Something for Everyone: With so many different types of shows and movies (genres), Netflix can keep a wide range of viewers happy and cater to many different tastes.

4.1 For continuous variable(s): Distplot, countplot, histogram for univariate analysis (10 Points)

```
country_df = df.assign(country=df['country'].str.split(',
')).explode('country')

genere_df = df.assign(listed_in=df['listed_in'].str.split(', ',
expand=False)).explode('listed_in')

cast_df = df.assign(cast=df['cast'].str.split(', ')).explode('cast')

director_df = df.assign(director=df['director'].str.split(',
')).explode('director')
```

Pre-processing and Univariate analysis is performed on columns such as country, listed_in, cast, and director contain multiple comma separated values per title

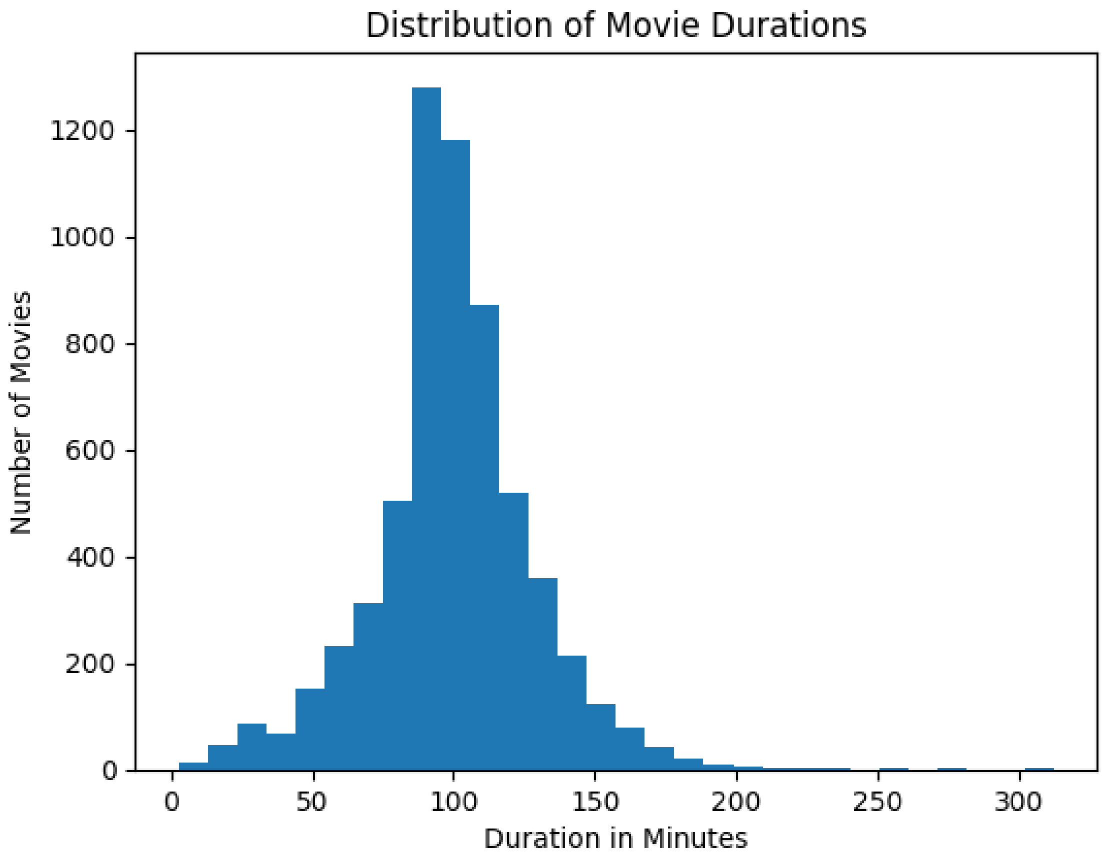
Exact unique values using explode

```
country_df['country'].nunique()

128

genere_df['listed_in'].nunique()

42
```

Unique values without multiple comma-separated values

Continuous Variable

```
import matplotlib.pyplot as plt

movies = df[df['type'] == 'Movie'].copy()

# Convert 'duration' to numeric, coercing errors to NaN, then drop
NaNs
movies['duration_int'] =
pd.to_numeric(movies['duration'].str.replace(' min',''),
errors='coerce')
movies.dropna(subset=['duration_int'], inplace=True)

# Now convert to integer type
movies['duration_int'] = movies['duration_int'].astype(int)

plt.figure()
plt.hist(movies['duration_int'], bins=30)
plt.xlabel("Duration in Minutes")
plt.ylabel("Number of Movies")
plt.title("Distribution of Movie Durations")
plt.show()
```



Observation 1.Most movies fall within a specific duration range of 50 to 150 minutes rather than extreme short or long durations in visual.

2.The distribution is concentrated around standard feature film lengths(80 to 120 minutes) in visual.

3.In unnesting The columns were split and exploded to enable accurate country wise, genre wise, and contributor wise analysis.

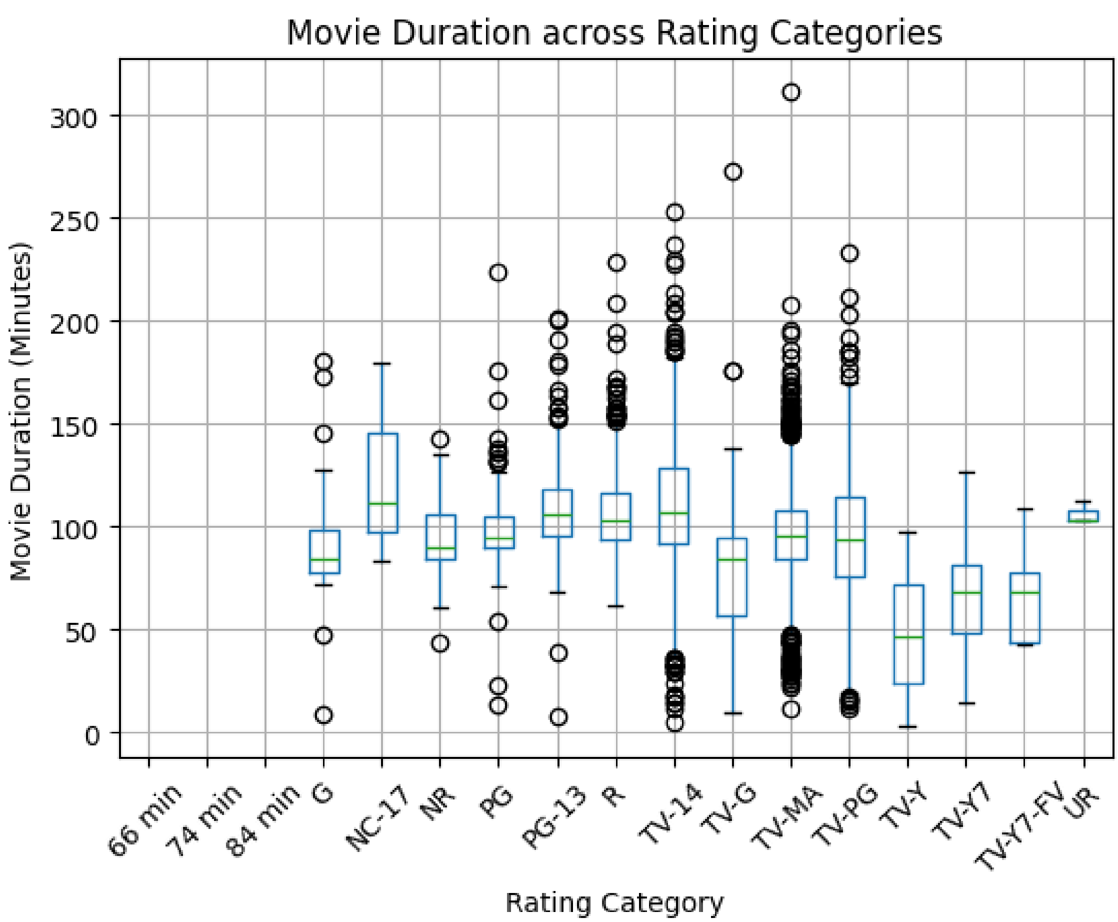
Business interpretation 1.Netflix movies mostly follow conventional movie duration patterns, which helps to engage audience.

2.Standard duration content is best for production investment.

4.2 For categorical variable(s): Boxplot

```
# Ensure duration is numeric
movies = df[df['type'] == 'Movie'].copy()

# Convert 'duration' to numeric, coercing errors to NaN, then drop
NaNs
movies['duration_int'] =
pd.to_numeric(movies['duration'].str.replace(' min',''),
errors='coerce')
```

```
movies.dropna(subset=['duration_int'], inplace=True)

# Now convert to integer type
movies['duration_int'] = movies['duration_int'].astype(int)

# Boxplot of duration by rating
plt.figure(figsize=(10,6))
movies.boxplot(column='duration_int', by='rating')
plt.xticks(rotation=45)
plt.xlabel("Rating Category")
plt.ylabel("Movie Duration (Minutes)")
plt.title("Movie Duration across Rating Categories")
plt.suptitle("")
plt.show()

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



Observations

1.Most rating categories show similar median movie durations.

2.A few rating groups contain outliers with unusually long movies.

3.The middle 50% of the data remains narrow across ratings, indicating consistent content length strategy

Business interpretation

1.Netflix maintains consistent movie duration regardless of audience rating category.

2.Occasional long duration content exists but standard duration films dominate, taking care of viewers attention span.

5. Missing Value & Outlier check (Treatment optional)

```
df.isnull().sum()
```

```
show_id          0
type             0
title            0
director         0
cast             0
country          0
date_added      88
release_year     0
rating           0
duration         0
listed_in        0
description      0
dtype: int64
```

Missing values were found in director, cast, country, date_added, rating, and duration columns.

After cleaning, most missing values were handled. A small number of missing dates remain which do not affect the analysis.

Outlier check

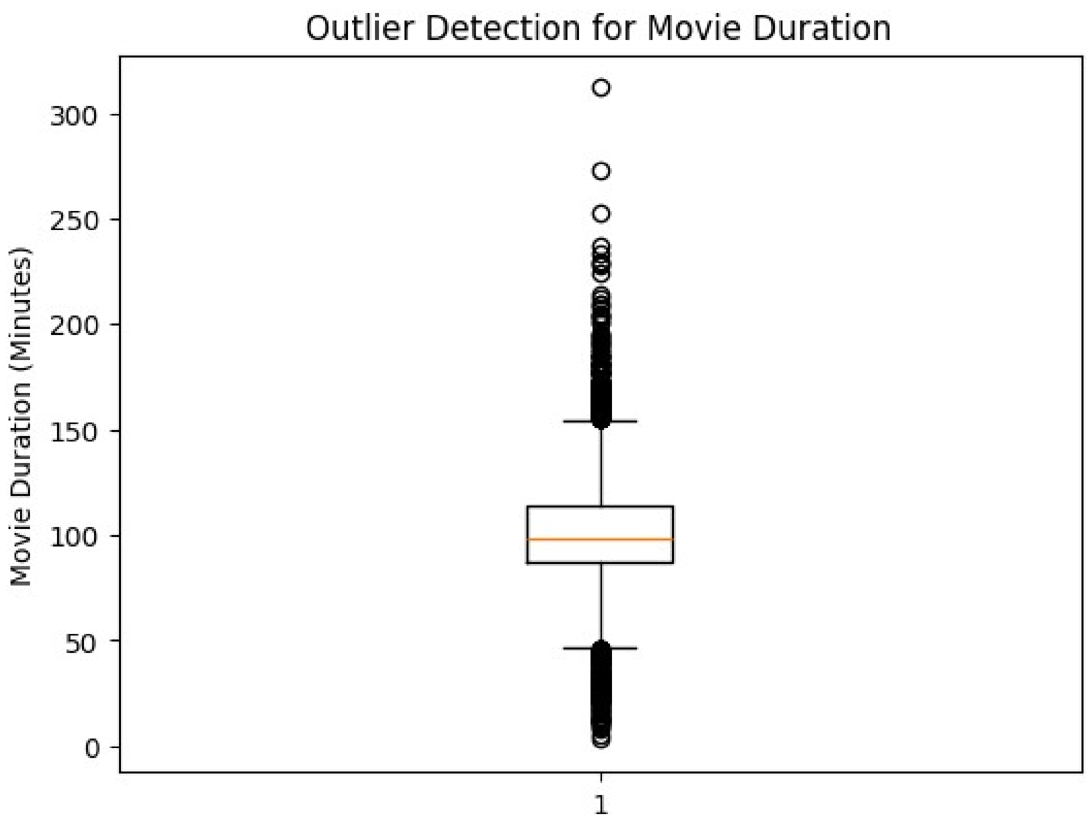
```
movies = df[df['type']=='Movie'].copy()
movies['duration_int'] =
pd.to_numeric(movies['duration'].str.replace(' min',''),
errors='coerce')
movies.dropna(subset=['duration_int'], inplace=True)
movies['duration_int'] = movies['duration_int'].astype(int)

movies['duration_int'].describe()
```

```
count    6128.000000
mean       99.577187
std        28.290593
min         3.000000
25%        87.000000
50%        98.000000
75%       114.000000
```

```
max        312.000000
Name: duration_int, dtype: float64
```

Boxplot for outlier visualization

```
import matplotlib.pyplot as plt

plt.figure()
plt.boxplot(movies['duration_int'])
plt.ylabel("Movie Duration (Minutes)")
plt.title("Outlier Detection for Movie Duration")
plt.show()
```



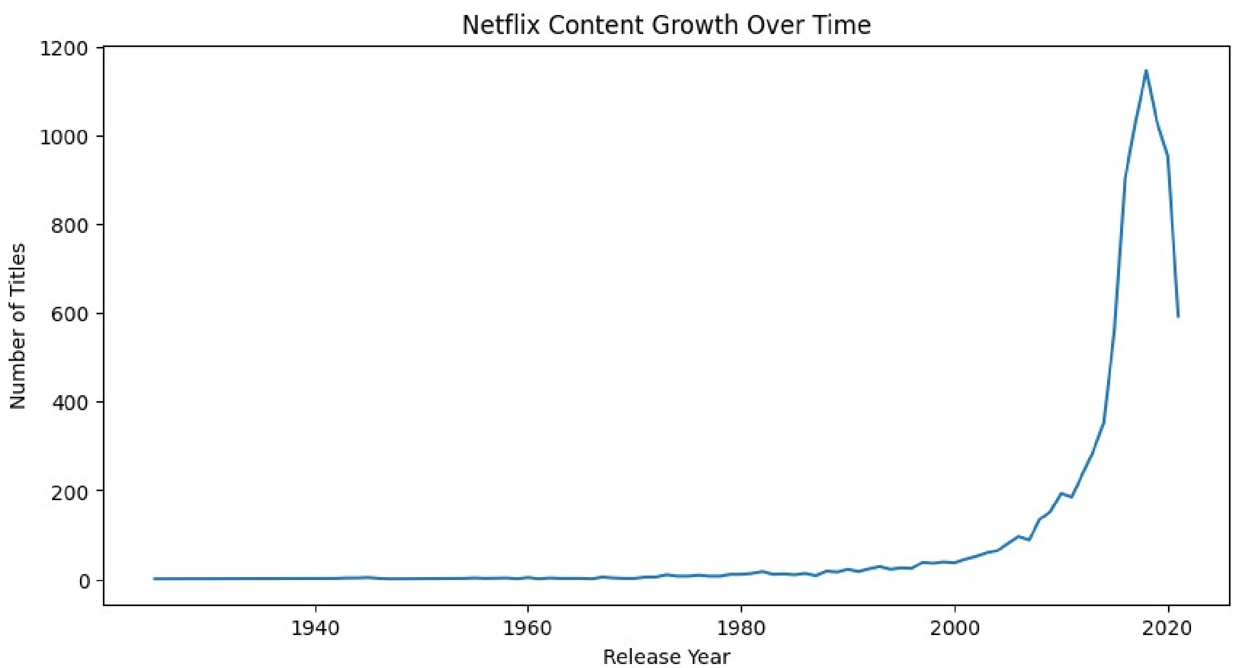
Advanced Analysis (Portfolio Level Insights)

The following analysis explores deeper trends such as content growth, strategy shifts, country contribution, and correlations.

Content Growth Over Time

This shows how Netflix expanded its content library year by year.

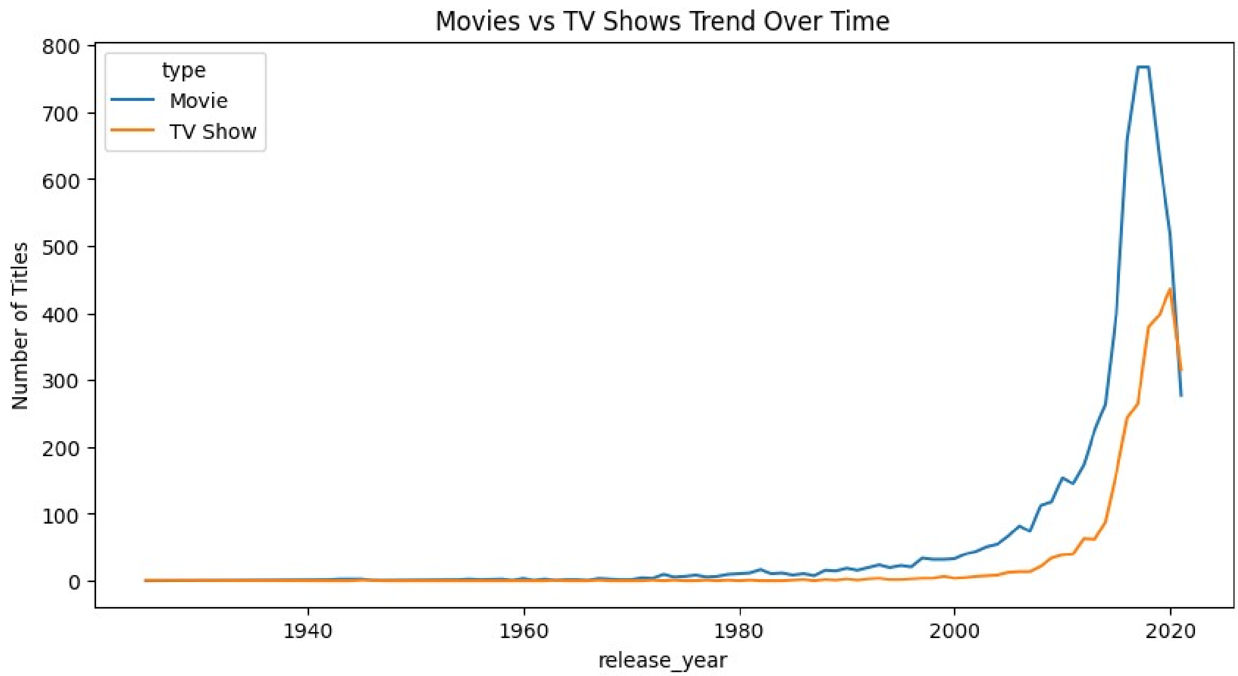
```
year_counts = df['release_year'].value_counts().sort_index()

plt.figure(figsize=(10,5))
plt.plot(year_counts.index, year_counts.values)
plt.xlabel("Release Year")
plt.ylabel("Number of Titles")
plt.title("Netflix Content Growth Over Time")
plt.show()
```



Movies vs TV Shows Trend

Compare how Netflix strategy shifted between movies and series.

```
type_year = df.groupby(['release_year','type']).size().unstack()

type_year.plot(figsize=(10,5))
plt.ylabel("Number of Titles")
plt.title("Movies vs TV Shows Trend Over Time")
plt.show()

/tmp/ipython-input-3054007837.py:1: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future
version of pandas. Pass observed=False to retain current behavior or
observed=True to adopt the future default and silence this warning.
  type_year = df.groupby(['release_year','type']).size().unstack()
```

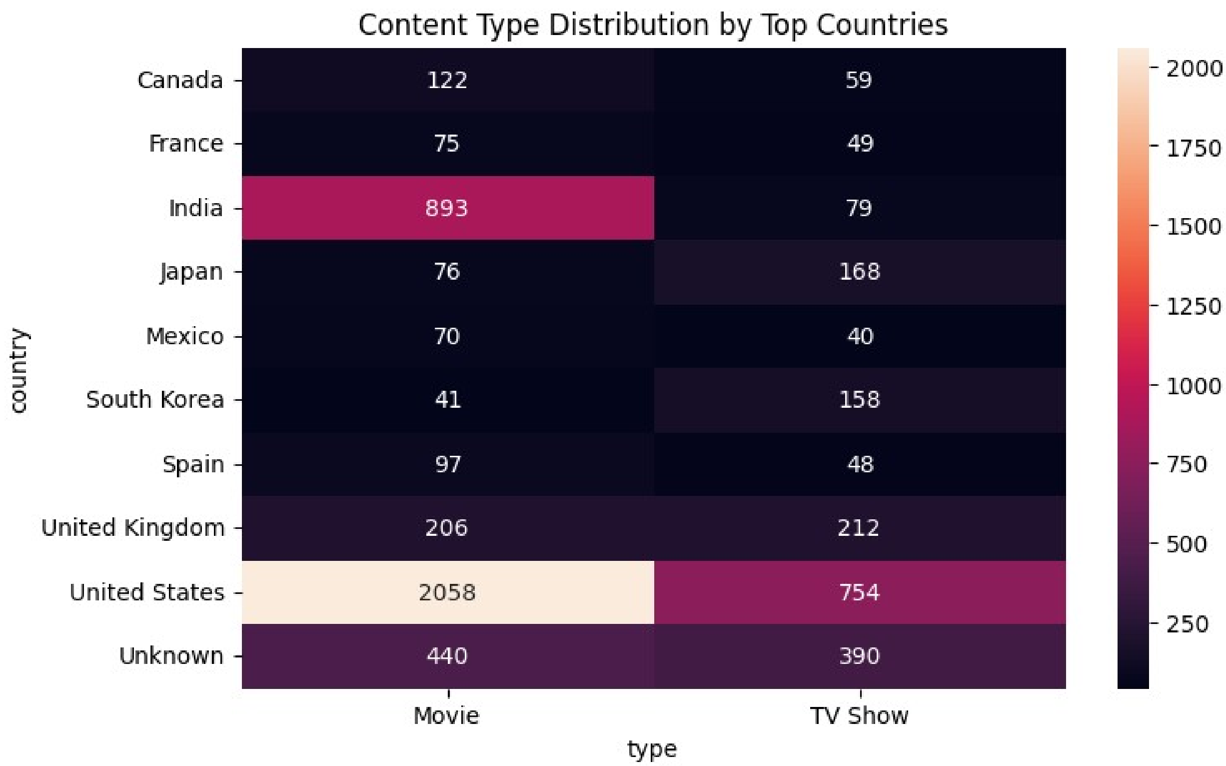


Content Distribution by Top Countries

Heatmap showing which countries produce more movies vs TV shows.

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

top_countries = df['country'].value_counts().head(10).index
temp = df[df['country'].isin(top_countries)]

pivot = temp.groupby(['country','type']).size().unstack()

plt.figure(figsize=(8,5))
sns.heatmap(pivot, annot=True, fmt='d')
plt.title("Content Type Distribution by Top Countries")
plt.show()
```

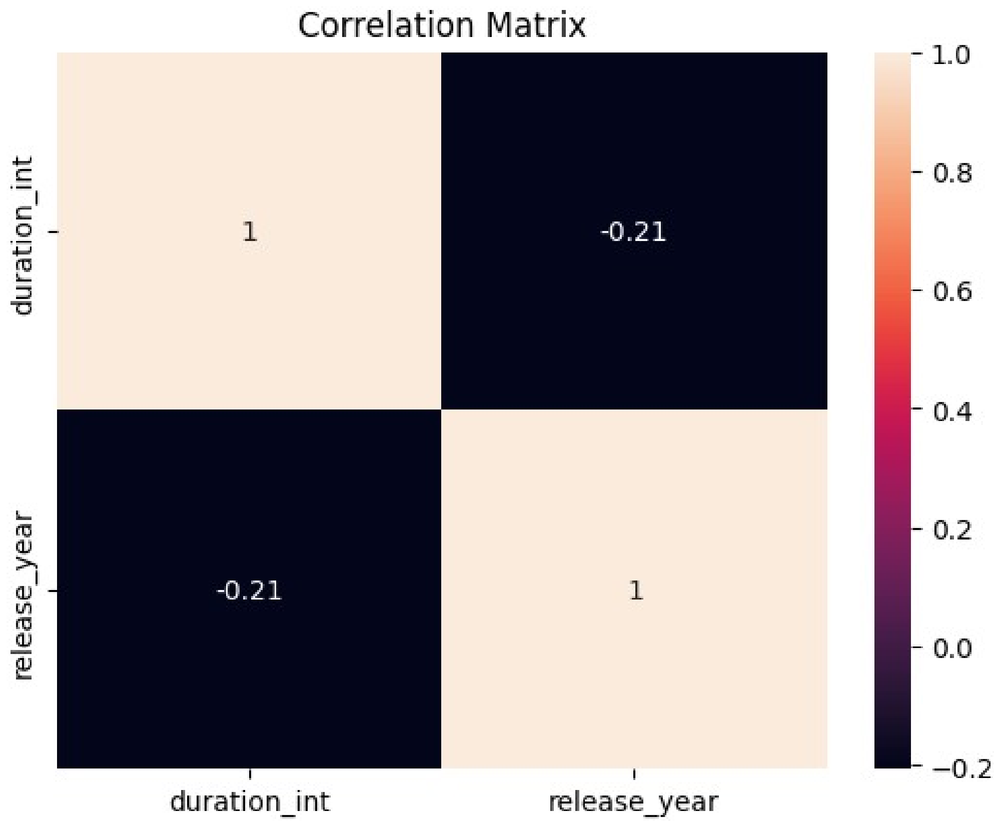
```
/tmp/ipython-input-3618895221.py:7: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future
version of pandas. Pass observed=False to retain current behavior or
observed=True to adopt the future default and silence this warning.
  pivot = temp.groupby(['country','type']).size().unstack()
```



Correlation Between Numeric Features

Check relationships between duration and release year.

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

movies = df[df['type']=='Movie'].copy()
# Convert 'duration' to numeric, coercing errors to NaN, then drop
NaNs
movies['duration_int'] =
pd.to_numeric(movies['duration'].str.replace(' min',''),
errors='coerce')
movies.dropna(subset=['duration_int'], inplace=True)
# Now convert to integer type
movies['duration_int'] = movies['duration_int'].astype(int)

numeric_df = movies[['duration_int','release_year']]

sns.heatmap(numeric_df.corr(), annot=True)
plt.title("Correlation Matrix")
plt.show()
```



Observation 1.Most movies on Netflix are about the same length – not too long, not too short.

2.A small number of movies are much, much longer than average. These are the 'outliers'.

3.We didn't see any movies that were unusually short.

Treatment decision 1.Some really long movies might be special projects or unique content, so we should probably keep them in our analysis.

2.If we remove these longer movies, it might make our overall understanding of movie lengths less accurate.

Business interpretation 1.By making sure all the data is complete (no missing values), we can trust our analysis about trends, what's popular in different countries, and who's making the content, without losing any important information.

2.Netflix usually makes movies of a typical length. However, they sometimes release longer films, likely to attract specific groups of viewers or offer premium content.

Insights based on Non-Graphical and Visual Analysis

6.1 Comments on the Range of Attributes 1.Old and New Content: Netflix has a really wide variety of content, from very old movies and shows made in 1925 all the way up to brand new ones from 2021.

2.Movie Lengths Vary, But Most Are Average: While some movies are very short and some are very long, most movies on Netflix are around a typical, standard length.

3.Content from Everywhere, Many Types of Shows: Netflix gets its content from many different countries and offers a huge number of different genres (types of shows/movies). This means they look all over the world for content and have something for almost every taste.

4.Content for All Ages: The ratings on Netflix show that they have content suitable for everyone, from young children to adults.

6.2 Comments on Distribution and Relationships 1.Movies continue to dominate the catalog, but TV shows are catching up: Netflix has more movies than TV shows, but they've been adding a lot more TV shows recently, showing a growing focus on them.

2.Netflix Content additions accelerated significantly after 2015 its content big-time after 2015: Netflix really started adding tons of new movies and shows after 2015, which means they were pushing hard to expand what they offered.

3.A few countries make most of Netflix's content: Most of the shows and movies on Netflix come from just a handful of countries, even though Netflix is available everywhere.

4.Drama and international films are everywhere on Netflix: You'll find a lot of dramas and movies/shows from other countries on Netflix because these types of content are very common.

5.Netflix likes standard-length movies: Most movies on Netflix are around the same typical length, showing that Netflix has a consistent plan for how long its films should be.

6.3 Comments on Univariate and Bivariate Plots

1.Movie Lengths are Mostly the Same: Most movies on Netflix are about the same length, which means they prefer standard-length films.

2.More Movies, But TV Shows are Catching Up: Netflix has more movies than TV shows, but they're adding TV shows at a faster rate now.

3.Content Comes From a Few Key Countries: Most of Netflix's content is made in just a few countries, showing some places produce more than others.

4.Netflix Focuses on Popular Genres: Only a few types of shows and movies (genres) keep popping up, meaning Netflix is focusing on what's popular.

5.Netflix Has Been Adding a high volume of new releases Lately: They've been growing super fast, especially in recent years, by adding lots of new shows and movies.

6.TV Shows are Becoming a Bigger Deal: In the last few years, Netflix has really ramped up how many TV shows they're adding, showing they're focusing more on series.

7.Movie Length Stays Consistent, Even for Different Ratings: The typical length of a movie doesn't really change much whether it's for kids or adults, though there are a few unusually long movies.

7.Business insights

Insights

1.Netflix has more movies, but is making more TV shows now: Even though Netflix has a lot more movies overall, they've really started adding many more TV shows recently. This means they're trying to keep subscribers improve retention.

2.Netflix gets most of its content from just a few countries: Most of what you watch on Netflix comes from only a handful of countries. This is a big chance for them to make more shows and movies helping to attract people from other countries.

3.Drama and international genres show strong demand and international stories on Netflix: A few types of stories, like emotional dramas and shows/movies from different cultures, are really popular. This tells Netflix that people enjoy deep, diverse storytelling.

4.Most Netflix movies are a standard length:Netflix does this to make sure people stay engaged and don't stop watching because a movie is too long.

5.Netflix mostly targets adults: Most of the shows and movies on Netflix are for older teens and adults. This helps them decide what kind of new content to create.

6.Netflix exploded with new content after 2015:This big push likely helped Netflix to get lot more subscribers all over the world.

7.Netflix works with many different creators:Netflix partners with a huge number of creative people. This shows they like to bring in fresh talent and diverse perspectives.

Insight summary

Netflix is clearly shifting its focus towards making more TV shows like emotional stories (drama) and shows (international genres). Most of Netflix's content is still made in few places, but they are also careful about how long their shows and movies are.So ,Netflix has a great chance to make more original shows in different countries and create more TV series that keep people hooked for a long time.

Recommendations

1.Make more original TV shows: We should put more money into creating our own TV series because people are really into watching shows which keeps them engaged.

2.Create more local content for new countries: We should produce shows and movies that are specific to countries with low Netflix consumption. This will help us get new customers in those areas.

3.Focus on popular types of stories: When planning new shows and movies, with genres like dramas and stories from around the world, will help to build audience.

4.Keep movies at a normal length: We should continue to make movies that are about the standard length, as this matches how long people usually like to watch films.

5.Target older audiences: We should develop content mainly for young adults and adults, because most of the people watching now prefer shows and movies with mature ratings.

6.Work closely with favorite directors and actors: We should build strong relationships with the directors and actors, as this helps create exciting.

7.Launch new content at busy times: We should release our biggest new shows and movies during periods when we usually add a lot of content, as this will increase watch time.

Final executive summary

Netflix should focus on making more TV shows, as per their audience interest. They should also create shows and movies for different countries with less popularity. It's also good idea for them to keep making popular types of stories like dramas and international shows.Also, By working with their favorite creators, they can get more subscribers and keep people watching for a long time.