Nenflix Data Analysis

➢ What does this mean in simple terms?

Imagine Netflix is trying to figure out what kind of content to buy or make next year to beat its rivals like Amazon Prime and Disney+.

This project's job is to look at Netflix's history (content from 2008 to 2021) and answer three main questions about their content trends:

1. **Have they been focusing more on Movies or TV Shows over time?**
    - (Objective: Analyze the distribution of Movies vs. TV Shows over the years.)
2. **Which movie and TV show categories (genres) are the most popular, and how has that popularity changed?**
    - (Objective: Identify the most common genres and how their popularity has changed.)
3. **Which countries contribute the most content to Netflix's library?**
    - (Objective: Compare country-wise contributions to Netflix's catalog.)

➢ Why is this problem important?

Understanding these trends is **crucial for making smart business decisions**.

- By knowing what their audience has been watching (popular genres and balance of Movies vs. TV Shows), Netflix can **refine its strategy** for content acquisition and production.
- It helps them identify if they have any **gaps** or **underrepresented categories** that could be filled.
- It ensures they are catering to their **diverse international audiences** and staying competitive globally.

➢ Installing and starting Process :

1. STEP 1

```
➢ # --- STEP 1: SETUP AND DATA LOADING ---
➢ import pandas as pd
➢ import numpy as np
➢ import matplotlib.pyplot as plt
➢ import seaborn as sns
➢
➢ # Set visualization style
➢ sns.set_style("whitegrid")
➢
➢ # Define the file name (Updated to match your file)
➢ FILE_NAME = 'Netflix Dataset.csv'
➢
➢ # Load the dataset
```

```
try:
    df = pd.read_csv(FILE_NAME)
    print(f"Dataset '{FILE_NAME}' loaded successfully.")
except FileNotFoundError:
    print(f"Error: The file '{FILE_NAME}' was not found. Please upload it.")
    exit()

# Define the correct column names for the analysis
TYPE_COL = 'Category'     # 'Movie' or 'TV Show'
DATE_COL = 'Release_Date' # Used to extract the year
GENRE_COL = 'Type'        # Genre/Type column
COUNTRY_COL = 'Country'   # Country column

print(f"Data loaded with {df.shape[0]} records.")
```

: SETUP AND DATA LOADING ---, prepares the Python environment and loads the Netflix dataset, setting the stage for all subsequent analysis.

1. Library Imports and Setup

Code	Explanation
import pandas as pd	Imports the pandas library, which is essential for working with data tables (DataFrames). The alias pd is the standard convention.
import numpy as np	Imports the NumPy library, used for advanced mathematical and array operations.
import matplotlib.pyplot as plt	Imports the primary plotting interface from Matplotlib for creating static, interactive, and animated visualizations.
import seaborn as sns	Imports the Seaborn library, which is built on Matplotlib and provides a high-level interface for drawing attractive statistical graphics.
sns.set_style("whitegrid")	Sets the visual theme for all Seaborn/Matplotlib plots to use a white background with gray grid lines, improving readability.

2. File and Data Loading

Code	Explanation
FILE_NAME = 'Netflix Dataset.csv'	Defines the name of the file to be loaded.
try: df = pd.read_csv(FILE_NAME)	Attempts to read the CSV file and load its contents into a variable named df (for DataFrame).
except FileNotFoundError: ...	This try...except block handles errors: if the file is not found, the program prints an error message instead of crashing.
print(f"Data loaded with {df.shape[0]} records.")	Prints a confirmation that the data was loaded and specifies the total number of rows (records) in the dataset.

3. Column Definitions

These lines define variables to hold the specific column names used in the dataset. This practice (aliasing) makes the code easier to read and maintain, as you only have to update the column name definitions once if the dataset file changes.

- TYPE_COL = 'Category': The column distinguishing between **'Movie'** and **'TV Show'**.
- DATE_COL = 'Release_Date': The column used to extract the **release year**.
- GENRE_COL = 'Type': The column containing **genres** of the content.
- COUNTRY_COL = 'Country': The column containing the **country of origin**.

1. STEP 2 :

```
# --- STEP 3: ANALYSIS - OBJECTIVE 1: MOVIES VS. TV SHOWS OVER TIME ---
print("\n--- 1. Content Distribution (Movies vs. TV Shows) Over Time ---")

# Filter for content released from 2008 onwards
df_trend = df[df[YEAR_COL] >= 2008].copy()

# Group by release year and category (Movie/TV Show)
content_by_year = df_trend.groupby([YEAR_COL, TYPE_COL]).size().unstack(fill_value=0)

# Plotting
```

```
plt.figure(figsize=(12, 6))
content_by_year.plot(kind='line', stacked=False, ax=plt.gca())
plt.title(f'Content Added to Netflix by Type Over the Years')
plt.xlabel('Release Year')
plt.ylabel('Number of Titles')
plt.legend(title='Category')
plt.savefig('content_distribution_over_years.png')
plt.show()
```

CODE EXPLANATION :

Code Segment	Purpose
print("\n--- 1. Content Distribution... ---")	A descriptive heading printed to the console to mark the start of this analysis step.
Data Preparation	
df_trend = df[df[YEAR_COL] >= 2008].copy()	Filters the Dataset. It creates a new DataFrame (df_trend) containing only the records released from 2008 onwards. This focuses the analysis on the recent strategic period mentioned in the problem statement.
content_by_year = df_trend.groupby([YEAR_COL, TYPE_COL]).size().unstack(fill_value=0)	Aggregates the Data. This is the core transformation: .groupby([YEAR_COL, TYPE_COL]) groups the data by Release Year and Category (Movie or TV Show). .size() counts the number of titles within each year-category group. .unstack(fill_value=0) pivots the table so that Movie and TV Show become separate columns, with the release year as the index. Any year where a content type had

Code Segment	Purpose
	zero entries is explicitly filled with 0.
Plotting	
plt.figure(figsize=(12, 6))	Sets the size of the visualization (12 inches wide, 6 inches high) for better clarity.
content_by_year.plot(kind='line', stacked=False, ax=plt.gca())	Generates the Plot. It uses the transformed data (content_by_year) to create a line chart. The stacked=False parameter ensures the Movie and TV Show counts are plotted as separate lines, allowing for a direct comparison of their trends.
plt.title(...), plt.xlabel(...), plt.ylabel(...)	Sets the chart title and axis labels to ensure the plot is clear and informative.
plt.legend(title='Category')	Adds a legend to distinguish between the lines for Movies and TV Shows.
plt.savefig('content_distribution_over_years.png')	Saves the generated plot to a file, which is a required project outcome.
plt.show()	Displays the plot (as shown in the visual output).

OUT PUT :

This line chart, titled "Content Added to Netflix by Type Over the Years", illustrates the evolution of Netflix's content catalog from 2008 to 2021 by tracking the annual number of Movies and TV Shows released.

The chart is crucial for Objective 1 of your project, which is to analyze the distribution of Movies vs. TV Shows over the years[1].

Key Trends Observed

The visualization clearly shows a dramatic shift in Netflix's content acquisition and release strategy, particularly starting around 2014-2015.

1. Pre-2015: Low Activity (2008–2014)

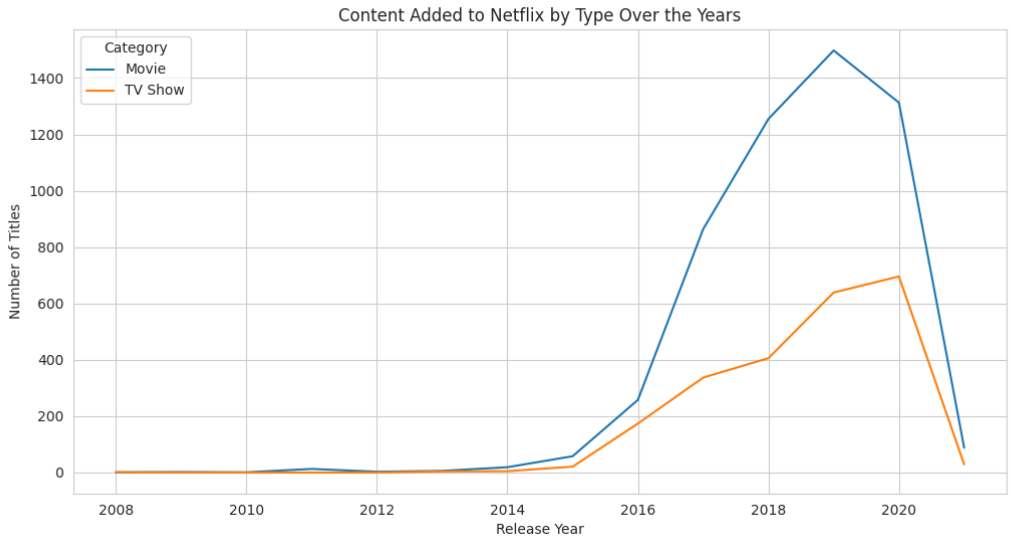
- Content addition was minimal and relatively flat across both categories .
- The lines for both Movies (blue) and TV Shows (orange) hover near the zero mark, indicating very few new releases added to the platform during these years.

2. Rapid Growth Phase (2015–2019)

- Movies Dominate Growth: Starting around 2015, there is a sharp, exponential increase in the number of Movie titles added annually. The Movie line consistently remains significantly higher than the TV Show line.
- TV Show Steady Increase: The number of TV Shows also increases, but at a more gradual and sustained pace compared to the explosive growth in Movies.
- Peak Content Addition: The peak addition year for both content types, but especially for Movies, occurred around 2019, with over 1,400 Movie titles and approximately 650 TV Show titles added.

3. Sharp Decline (2020–2021)

- Both content types show a steep drop in the number of added titles in the final years of the dataset (2020 and 2021).
- In 2021, the number of new releases dropped to the lowest point since the pre-growth period (around 2014), indicating a massive reduction in the volume of content added in the last year of the available data.



STEP 3: ANALYSIS - OBJECTIVE 2: TOP GENRES ---

```
print("\n--- 2. Genre Popularity Analysis ---")

# Calculate top 10 genres using the helper function
genre_counts = split_and_count(df, GENRE_COL)
top_10_genres = genre_counts.head(10)

# Plotting
plt.figure(figsize=(10, 6))
sns.barplot(x=top_10_genres.index, y=top_10_genres.values, palette="viridis")
plt.title(f'Top 10 Most Common Genres')
plt.xlabel('Genre')
plt.ylabel('Count of Titles')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.savefig('top_10_genres.png')
plt.show()

print(f"Top 5 Genres: {top_10_genres.head(5).index.tolist()}")
```

CODE EXPLANATION :

Code Segment	Purpose
`print("\n--- 2. Genre Popularity Analysis ---")`	A descriptive heading printed to the console to mark the start of this analysis step.
Data Aggregation	
`genre_counts = split_and_count(df, GENRE_COL)`	**Calculates Genre Counts.** This uses the custom split_and_count helper function (defined in **STEP 2** but not shown here) to process the GENRE_COL (which holds comma-separated genres). It splits the strings, counts the occurrences of each individual genre, and returns a Series of counts.
`top_10_genres = genre_counts.head(10)`	**Identifies the Top 10.** It takes the top 10 genres from the genre_counts Series (which is already sorted by count in descending order).

Code Segment	Purpose
Plotting	
plt.figure(figsize=(10, 6))	Sets the size of the visualization (10 inches wide, 6 inches high).
sns.barplot(x=top_10_genres.index, y=top_10_genres.values, ...)	**Generates the Bar Chart.** It creates a bar plot using Seaborn: - x axis: The genre names (the index of top_10_genres). - y axis: The count of titles for each genre (the values of top_10_genres). - palette="viridis": Sets the color scheme for the bars.
plt.title(...), plt.xlabel(...), plt.ylabel(...)	Sets the chart title and axis labels.
plt.xticks(rotation=45, ha='right')	Rotates the genre labels on the X-axis by 45 degrees to prevent overlap and improve readability.
plt.tight_layout()	Automatically adjusts subplot parameters to give a tight layout, preventing labels from being cut off.
plt.savefig('top_10_genres.png')	Saves the generated bar chart to a file.
plt.show()	Displays the plot (as shown in the visual output).
print(f"Top 5 Genres: {...}")	Prints the names of the top 5 most common genres directly to the console for a quick summary.

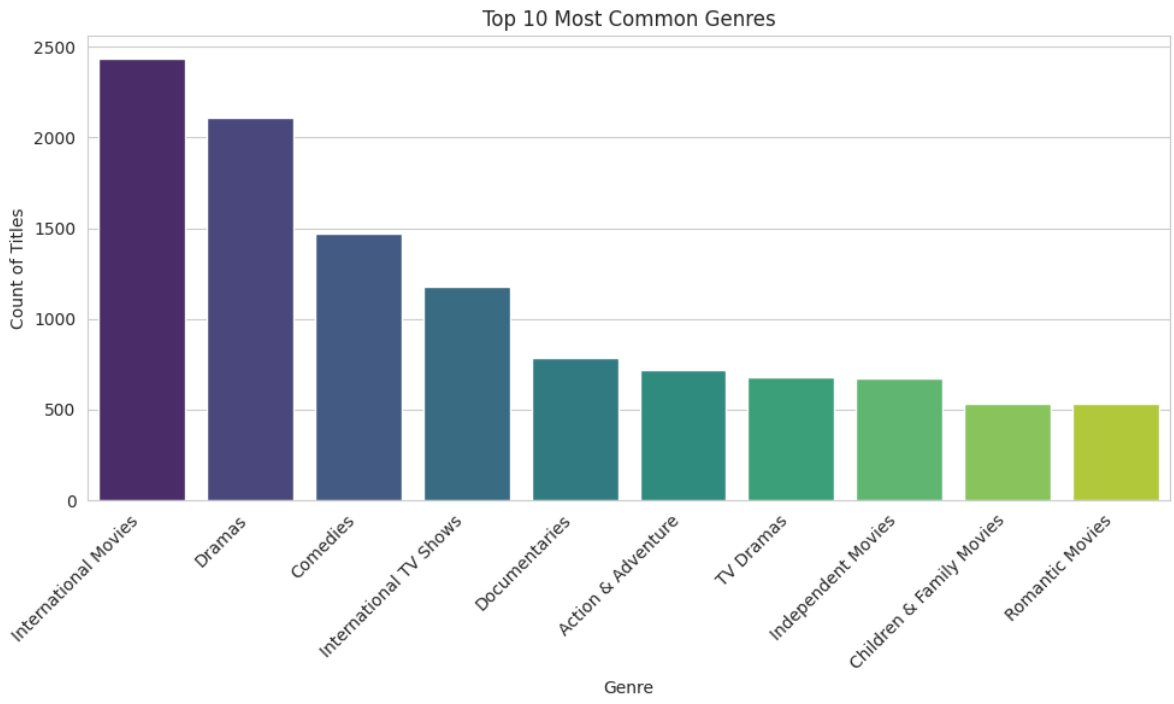
OUT PUT :

This bar chart, titled **"Top 10 Most Common Genres"**, visually represents the frequency of the 10 most popular content genres in the Netflix dataset. This analysis fulfills **Objective 2** of the project: "Identify the most common genres".

The vertical axis shows the **Count of Titles**, and the horizontal axis lists the **Genre** categories. Since many titles on Netflix have multiple associated genres, the total count in this chart is higher than the total number of unique titles in the dataset.

Key Observations and Insights

1. **Dominance of International Content and Drama:** The top two genres are significantly more numerous than the rest, highlighting Netflix's focus on non-US markets and dramatic storytelling.
    - **International Movies** is the most common category, with nearly **2,500 titles**.
    - **Dramas** is the second most common, with over **2,000 titles**.
2. **Core Content Pillars:** The top five genres represent the core of Netflix's content strategy:
    - **Comedies** (around 1,500 titles) and **International TV Shows** (around 1,200 titles) are the third and fourth most common.
    - **Documentaries** hold the fifth spot, with over 750 titles, showing a strong presence in non-fiction content.
3. **Action and Niche Categories:**
    - Genres like **Action & Adventure**, **TV Dramas**, and **Independent Movies** are consistently represented, ranging between 650 and 750 titles.
    - **Children & Family Movies** and **Romantic Movies** round out the top 10, each with over 500 titles.



STEP 04 :

```
# --- STEP 5: ANALYSIS - OBJECTIVE 3: COUNTRY CONTRIBUTIONS ---

print("\n--- 3. Country-wise Contribution Analysis ---")

# Calculate country contributions using the helper function

country_counts = split_and_count(df, COUNTRY_COL)
```

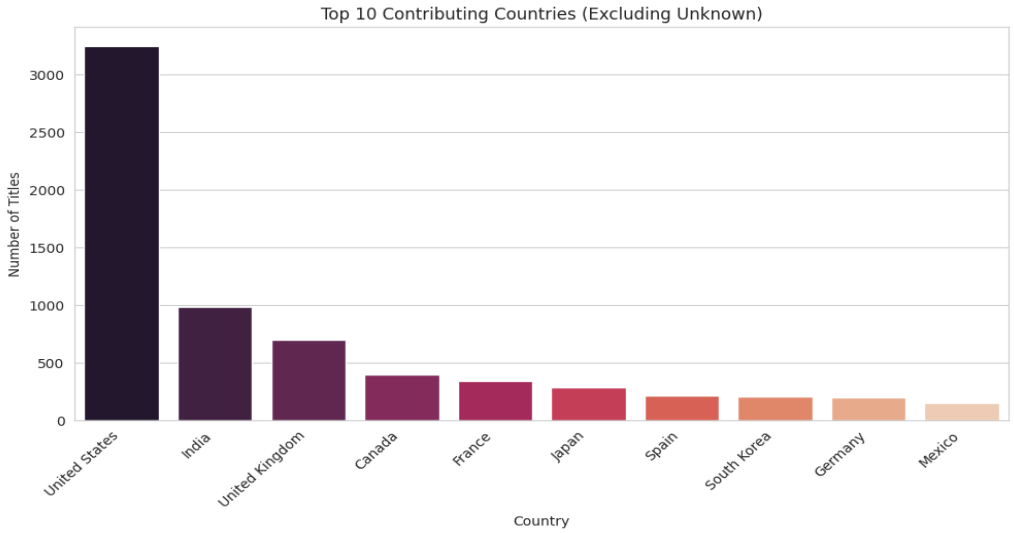
```
# Filter out 'Unknown' and get the top 10
top_10_countries = country_counts.drop(labels=['Unknown'], errors='ignore').head(10)
# Plotting
plt.figure(figsize=(10, 6))
sns.barplot(x=top_10_countries.index, y=top_10_countries.values, palette="rocket")
plt.title(f'Top 10 Contributing Countries (Excluding Unknown)')
plt.xlabel('Country')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.savefig('top_10_countries.png')
plt.show()
print(f"The country with the highest contribution is: {top_10_countries.index[0] if not top_10_countries.empty else 'N/A'}.")
```

EXPLANATION :

Code Segment	**Purpose**
print("\n--- 3. Country-wise Contribution... ---")	A descriptive heading printed to the console to mark the start of this analysis step.
Data Aggregation and Filtering	
country_counts = split_and_count(df, COUNTRY_COL)	**Calculates Country Counts.** This uses the custom split_and_count helper function to process the COUNTRY_COL (which may contain multiple, comma-separated countries for a single title). It splits the strings and counts the occurrences of each individual country.
top_10_countries = country_counts.drop(labels=['Unknown'], errors='ignore').head(10)	**Filters and Selects Top 10.** .drop(labels=['Unknown'], errors='ignore'): Excludes the category 'Unknown' from the counts, as this is a placeholder for missing data and doesn't represent a true country

Code Segment	Purpose
	contribution. .head(10): Selects the top 10 countries with the highest title counts (since country_counts is sorted by value in descending order).
Plotting	
plt.figure(figsize=(10, 6))	Sets the size of the visualization (10 inches wide, 6 inches high).
sns.barplot(x=top_10_countries.index, y=top_10_countries.values, ...)	**Generates the Bar Chart.** It creates a bar plot using Seaborn, displaying the top 10 countries on the **X-axis** and their **Title Counts** on the **Y-axis**.
plt.title(...), plt.xlabel(...), plt.ylabel(...)	Sets the chart title and axis labels to ensure the plot is clear and informative.
plt.xticks(rotation=45, ha='right')	Rotates the country names on the X-axis by 45 degrees to prevent overlap.
plt.tight_layout()	Adjusts the plot to ensure all elements, especially the rotated labels, fit neatly within the figure area.
plt.savefig('top_10_countries.png')	Saves the generated bar chart to a file.
plt.show()	Displays the plot (as shown in the visual output).
print(f"The country with the highest contribution is: {...}")	Prints the name of the top-contributing country to the console for a quick, explicit summary.

OUTPUT :



This bar chart, titled **"Top 10 Contributing Countries (Excluding Unknown)"**, fulfills **Objective 3** of your project: "Compare country-wise contributions to Netflix's catalog."
It visualizes which countries have contributed the largest number of titles (Movies and TV Shows) to the Netflix library, demonstrating insights into its global content representation.

Key Observations and Insights

1. Overwhelming U.S. Dominance

- The **United States** is by far the largest contributor, with over **3,000 titles**.
- Its contribution is more than **three times** that of the next highest country, indicating that the U.S. remains the core market and primary source of content for Netflix.

2. Emerging International Hubs

- **India** is the second-highest contributor with approximately **1,000 titles**. This highlights the strategic importance and significant volume of content acquired from the Indian market (often Bollywood and regional cinema).
- The **United Kingdom** is close behind India with over **700 titles**.

3. Broad Global Reach

- The remaining countries in the top 10 (**Canada, France, Japan, Spain, South Korea, Germany, and Mexico**) show that Netflix maintains a broadly diversified global catalog.
- **South Korea** and **Japan**'s presence in the top 10 reflects the platform's investment in Asian content (K-dramas, anime, etc.), which has seen huge international growth.
- The counts for these remaining countries are relatively similar, clustering between **200 and 500 titles**, suggesting a significant, but not dominant, level of localized content production and acquisition across major global markets.