VideoStreamingPlatformsReport

June 1, 2023

1 DA CAPSTONE PROJECT-VIDEO STREAMING PLATFORMS REPORT

—Romana & Nirmala (Team memebers)

Objective of the project

The Capstone Project is an opportunity for bootcamp students to begin to pull together a numbe

Table of contents

2 1.Scenario

VIDEO STREAMING PLATFORMS REPORT

Task: Use the following datasets to generate a report to the general public on the video str

Disney+ Titles: CSV file (disney_plus_titles.csv)
Hulu Titles: CSV file (hulu_titles.csv)
Netflix Titles: CSV file (netflix_titles.csv)
Amazon Prime Titles: SQL table (amazon_prime_titles.sql)

3 2.Select a real-world dataset provided

Data set used:*

The datasets used for our case study are Disney.csv, Hulu.csv, Netflix.csv, and AmazonPrime.sq

About the Data:*

These datasets comprise a comprehensive list of movies and TV shows available on various strea

4 3.Prepare Phase

Data verififcation:

We have 3 CSV files and one SQL file with a lot of missing values. Let's load the datasets int

Perform any additional steps such as parsing dates, creating additional columns, merging multiple datasets, etc.

5 4.Process Phase

```
All processes from data cleaning, aggregation, analysis, to visualization, will be carried ou
```

6 5.Perform exploratory Analysis & Visualization

```
Load the dataset into a data frame using Panda explore the number of rows & columns, ranges of
Handle missing, incorrect and invalid data perform any additional steps (parsing dates, creati
```

7 6. Ask & answer questions about the data

- Ask at least 4 interesting questions about your dataset*

```
Answer the questions by computing the results using Numpy/Pandas or by plotting graphs using Ma
1.What is the overall distribution of movies and TV shows in the dataset? Predective model incl
2.Are there any regional or cultural factors that contribute to the success of movies from spe
3.Are there any dominant genres that significantly outweigh others in terms of popularity?
4.Are there any specific directors who are consistently successful in terms of the number of ti
```

8 7: Summarize your inferences & write a conclusion

```
Write a summary of what you've learned from the analysis.
Share ideas for future work on the same topic using other relevant datasets
Share links to resources you found useful during your analysis
```

9 8: Make a submission & share your work

```
Jupyter Notebook file (.ipynb) containing at least:
Four different questions that are explored and answered
At least two visualizations for each of the four questions
A summary for each question explaining what approach you took and what your conclusions were
PowerBI Dashboard (.pwib) containing at least three visualizations reporting on an aspect of th
```

---

package for converting notebook to pdf

[27]:
```
conda install nbconvert
```

```
Collecting package metadata (current_repodata.json): …working… done
Solving environment: …working… done

# All requested packages already installed.
```

```
Note: you may need to restart the kernel to use updated packages.


==> WARNING: A newer version of conda exists. <==
  current version: 23.3.1
  latest version: 23.5.0

Please update conda by running

    $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

     conda install conda=23.5.0
```

Import necessary packages

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import psycopg2
import seaborn as sns
import psycopg2
import nbconvert
```

[3]:
```
from nbconvert import HTMLExporter

fileExporter = HTMLExporter(template_name='classic')
(body, resources) = fileExporter.from_filename('VideoStreamingPlatformsReport.
 ↪ipynb')
```

10 Select a real-world dataset provided

We will upload the datasets that will help us to answer the business task given. We will focus on the following datasets:

Movies on Netflix, Prime Video, Hulu and Disney+ TV shows on Netflix, Prime Video, Hulu and Disney+

[4]:
```
# Specify the file path
file_path_disney = 'C:
 ↪\\Users\\paperspace\\Downloads\\da_project_data\\disney_plus_titles.csv'
file_path_hulu = 'C:
 ↪\\Users\\paperspace\\Downloads\\da_project_data\\hulu_titles.csv'
```

```
file_path_netflix = 'C:
 ↪\\Users\\paperspace\\Downloads\\da_project_data\\netflix_titles.csv'
# Read the CSV file into a DataFrame
disney_plus = pd.read_csv(file_path_disney)
hulu_title = pd.read_csv(file_path_hulu)
netflix_title =pd.read_csv(file_path_disney)
```

11 Read SQL files

[5]:
```
!pip install psycopg2
```

```
Requirement already satisfied: psycopg2 in
c:\users\paperspace\anaconda3\lib\site-packages (2.9.6)
```

[6]:
```
# Replace the placeholders with your PostgreSQL database credentials
connection = psycopg2.connect(
    host="localhost",
    database="amazon_prime",
    user="postgres",
    password="welcome123"
)
```

[7]:
```
cursor = connection.cursor()
```

[8]:
```
# Example: Execute a SELECT query
import pandas as pd
cursor.execute("SELECT * FROM amazon_prime_titles")

# Fetch the results
results = cursor.fetchall()

# Convert the results into a pandas DataFrame
amazon_prime_df = pd.DataFrame(results, columns=[desc[0] for desc in cursor.
 ↪description])

# Close the cursor and the database connection
cursor.close()
connection.close()

# Print the DataFrame
#print(amazon_prime_df)
```

12 3.Prepare Phase (Preview/Inpecting Data set)

[9]:
```
disney_plus.shape
```

[9]:
```
(1450, 12)
```

[10]:
```
hulu_title.shape
```

[10]:
```
(3073, 12)
```

[11]:
```
netflix_title.shape
```

[11]:
```
(1450, 12)
```

[12]:
```
amazon_prime_df.shape
```

[12]:
```
(9668, 12)
```

[13]:
```
disney_plus.head(4)
```

[13]:
```
  show_id   type                                              title  \
0      s1  Movie  Duck the Halls: A Mickey Mouse Christmas Special
1      s2  Movie                              Ernest Saves Christmas
2      s3  Movie                       Ice Age: A Mammoth Christmas
3      s4  Movie                         The Queen Family Singalong

                           director  \
0  Alonso Ramirez Ramos, Dave Wasson
1                       John Cherry
2                      Karen Disher
3                   Hamish Hamilton

                                               cast        country  \
0  Chris Diamantopoulos, Tony Anselmo, Tress MacN…            NaN
1           Jim Varney, Noelle Parker, Douglas Seale           NaN
2  Raymond Albert Romano, John Leguizamo, Denis L…  United States
3  Darren Criss, Adam Lambert, Derek Hough, Alexa…            NaN

          date_added  release_year rating duration                 listed_in  \
0  November 26, 2021          2016   TV-G   23 min          Animation, Family
1  November 26, 2021          1988     PG   91 min                    Comedy
2  November 26, 2021          2011   TV-G   23 min  Animation, Comedy, Family
3  November 26, 2021          2021  TV-PG   41 min                   Musical

                                      description
0   Join Mickey and the gang as they duck the halls!
1  Santa Claus passes his magic bag to a new St. …
2         Sid the Sloth is on Santa's naughty list.
```

```
3              This is real life, not just fantasy!
```

[14]:
```
hulu_title.head(4)
```

[14]:
```
  show_id   type                          title director  cast country  \
0      s1  Movie  Ricky Velez: Here's Everything      NaN   NaN     NaN
1      s2  Movie                   Silent Night      NaN   NaN     NaN
2      s3  Movie                   The Marksman      NaN   NaN     NaN
3      s4  Movie                           Gaia      NaN   NaN     NaN

         date_added  release_year rating duration               listed_in  \
0  October 24, 2021          2021  TV-MA      NaN        Comedy, Stand Up
1  October 23, 2021          2020    NaN   94 min  Crime, Drama, Thriller
2  October 23, 2021          2021  PG-13  108 min        Action, Thriller
3  October 22, 2021          2021      R   97 min                  Horror

                                        description
0  Comedian Ricky Velez bares it all with his ho…
1  Mark, a low end South London hitman recently r…
2  A hardened Arizona rancher tries to protect an…
3  A forest ranger and two survivalists with a cu…
```

[15]:
```
netflix_title.head(4)
```

[15]:
```
  show_id   type                                          title  \
0      s1  Movie  Duck the Halls: A Mickey Mouse Christmas Special
1      s2  Movie                           Ernest Saves Christmas
2      s3  Movie                    Ice Age: A Mammoth Christmas
3      s4  Movie                      The Queen Family Singalong

                         director  \
0  Alonso Ramirez Ramos, Dave Wasson
1                      John Cherry
2                    Karen Disher
3                  Hamish Hamilton

                                          cast        country  \
0  Chris Diamantopoulos, Tony Anselmo, Tress MacN…           NaN
1          Jim Varney, Noelle Parker, Douglas Seale          NaN
2  Raymond Albert Romano, John Leguizamo, Denis L…  United States
3  Darren Criss, Adam Lambert, Derek Hough, Alexa…           NaN

         date_added  release_year rating duration                 listed_in  \
0  November 26, 2021          2016   TV-G   23 min          Animation, Family
1  November 26, 2021          1988     PG   91 min                    Comedy
2  November 26, 2021          2011   TV-G   23 min  Animation, Comedy, Family
3  November 26, 2021          2021  TV-PG   41 min                   Musical
```

```
                                          description
0   Join Mickey and the gang as they duck the halls!
1  Santa Claus passes his magic bag to a new St. …
2          Sid the Sloth is on Santa's naughty list.
3               This is real life, not just fantasy!
```

[16]:
```
amazon_prime_df.head(4)
```

[16]:
```
  show_id   type                  title        director  \
0      s1  Movie   The Grand Seduction    Don McKellar
1      s2  Movie  Take Care Good Night    Girish Joshi
2      s3  Movie  Secrets of Deception     Josh Webber
3      s4  Movie    Pink: Staying True  Sonia Anderson

                                              cast        country  \
0     Brendan Gleeson, Taylor Kitsch, Gordon Pinsent         Canada
1   Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar          India
2  Tom Sizemore, Lorenzo Lamas, Robert LaSardo, R…  United States
3  Interviews with: Pink, Adele, Beyoncé, Britney…  United States

       date_added  release_year rating duration                listed_in  \
0  March 30, 2021          2014   None  113 min            Comedy, Drama
1  March 30, 2021          2018    13+  110 min     Drama, International
2  March 30, 2021          2017   None   74 min  Action, Drama, Suspense
3  March 30, 2021          2014   None   69 min              Documentary

                                          description
0  A small fishing village must procure a local d…
1  A Metro Family decides to fight a Cyber Crimin…
2  After a man discovers his wife is cheating on …
3  Pink breaks the mold once again, bringing her …
```

[61]:
```
disney_plus.dtypes  # data type of disney df
```

[61]:
```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
```

```
dtype: object
```

[65]:
```
hulu_title.dtypes
```

[65]:
```
show_id          object
type             object
title            object
director         object
cast            float64
country          object
date_added       object
release_year      int64
rating           object
duration         object
listed_in        object
description      object
dtype: object
```

[66]:
```
netflix_title.dtypes
```

[66]:
```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

[67]:
```
amazon_prime_df.dtypes
```

[67]:
```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
```

```
description     object
dtype: object
```

[17]:
```
amazon_prime_df.isnull().sum()
```

[17]:
```
show_id            0
type               0
title              0
director        2082
cast            1233
country         8996
date_added      9513
release_year       0
rating           337
duration           0
listed_in          0
description        0
dtype: int64
```

[18]:
```
disney_plus.isnull().sum()
```

[18]:
```
show_id           0
type              0
title             0
director        473
cast            190
country         219
date_added        3
release_year      0
rating            3
duration          0
listed_in         0
description       0
dtype: int64
```

[19]:
```
hulu_title.isnull().sum()
```

[19]:
```
show_id            0
type               0
title              0
director        3070
cast            3073
country         1453
date_added        28
release_year       0
rating           520
duration         479
```

```
listed_in          0
description        4
dtype: int64
```

[20]:
```
netflix_title.isnull().sum()
```

[20]:
```
show_id          0
type             0
title            0
director       473
cast           190
country        219
date_added       3
release_year     0
rating           3
duration         0
listed_in        0
description      0
dtype: int64
```

After reviewing the four datasets, we found some important points:

All datasets have unnecesarry first column (Show_id) All datasets share a similar column name which will help a lot in merging process

In hulu director,cast has mostly NaN and most of the country too have NaN values,Country column values are mostly missing in all csv

Duration colmun has to be fixed in all csv, values should be in min.(doubt)

80-90 % of country & date_added of Amazon_Prime_df 's data are missing so we need to impute it by mean of the column, rest all missing values from other colmuns does not contribute significantely so we will replac them with unknow

13 Is it okay to concatenate all the files or do seprate EDA of each csv?

If you have three CSV files from different streaming platforms (Disney, Hulu, Netflix and Amazon Prime) with the same column names and the same number of columns, it is generally fine to concatenate them into a single dataframe and perform exploratory data analysis (EDA) on the combined dataset.

Concatenating the data from multiple sources can be beneficial when you want to analyze the data collectively and look for patterns or insights across different streaming platforms. It allows you to have a more comprehensive view of the video streaming industry as a whole.

By concatenating the datasets, you can perform EDA on the combined data to gain insights and draw comparisons between the platforms. You can explore various aspects such as the distribution of content types, release years, ratings, durations, genres, and more across all platforms.

However, it's also important to consider the specific analysis objectives and the nature of the data. If you have specific questions or hypotheses that are platform-specific, you might want to perform separate EDA on each platform to understand their unique characteristics.

In summary, if the datasets have the same structure and you are interested in analyzing the video streaming industry as a whole, concatenating the datasets and conducting EDA on the combined data can provide a broader perspective.

Concatenating the dataframe:

[21]:
```
merged_df = pd.concat([disney_plus, hulu_title, netflix_title,amazon_prime_df])
# Optional: Reset the index of the concatenated DataFrame
concatenated_df = merged_df.reset_index(drop=True)
```

14 Inspecting merged data frame

[22]:
```
merged_df.shape    # from all cvs files  and sql file
```

[22]: (15641, 12)

[24]:
```
merged_df.head(10)
```

[24]:

	show_id	type	title	director	cast	country
0	s1	Movie	Duck the Halls: A Mickey Mouse Christmas Special	Alonso Ramirez Ramos, Dave Wasson	Chris Diamantopoulos, Tony Anselmo, Tress MacN…	NaN
1	s2	Movie	Ernest Saves Christmas	John Cherry		
2	s3	Movie	Ice Age: A Mammoth Christmas	Karen Disher		
3	s4	Movie	The Queen Family Singalong	Hamish Hamilton		
4	s5	TV Show	The Beatles: Get Back	NaN		
5	s6	Movie	Becoming Cousteau	Liz Garbus		
6	s7	TV Show	Hawkeye	NaN		
7	s8	TV Show	Port Protection Alaska	NaN		
8	s9	TV Show	Secrets of the Zoo: Tampa	NaN		
9	s10	Movie	A Muppets Christmas: Letters To Santa	Kirk R. Thatcher		

```
1           Jim Varney, Noelle Parker, Douglas Seale            NaN
2  Raymond Albert Romano, John Leguizamo, Denis L…  United States
3  Darren Criss, Adam Lambert, Derek Hough, Alexa…            NaN
4  John Lennon, Paul McCartney, George Harrison, …            NaN
5              Jacques Yves Cousteau, Vincent Cassel  United States
6  Jeremy Renner, Hailee Steinfeld, Vera Farmiga,…            NaN
7  Gary Muehlberger, Mary Miller, Curly Leach, Sa…  United States
8  Dr. Ray Ball, Dr. Lauren Smith, Chris Massaro,…  United States
9  Steve Whitmire, Dave Goelz, Bill Barretta, Eri…  United States

          date_added  release_year rating   duration  \
0  November 26, 2021          2016   TV-G     23 min
1  November 26, 2021          1988     PG     91 min
2  November 26, 2021          2011   TV-G     23 min
3  November 26, 2021          2021  TV-PG     41 min
4  November 25, 2021          2021    NaN   1 Season
5  November 24, 2021          2021  PG-13     94 min
6  November 24, 2021          2021  TV-14   1 Season
7  November 24, 2021          2015  TV-14  2 Seasons
8  November 24, 2021          2019  TV-PG  2 Seasons
9  November 19, 2021          2008      G     45 min

                             listed_in  \
0                    Animation, Family
1                              Comedy
2            Animation, Comedy, Family
3                             Musical
4        Docuseries, Historical, Music
5            Biographical, Documentary
6          Action-Adventure, Superhero
7        Docuseries, Reality, Survival
8  Animals & Nature, Docuseries, Family
9              Comedy, Family, Musical

                                        description
0   Join Mickey and the gang as they duck the halls!
1  Santa Claus passes his magic bag to a new St. …
2          Sid the Sloth is on Santa's naughty list.
3               This is real life, not just fantasy!
4  A three-part documentary from Peter Jackson ca…
5  An inside look at the legendary life of advent…
6  Clint Barton/Hawkeye must team up with skilled…
7  Residents of Port Protection must combat volat…
8  A day in the life at ZooTampa is anything but …
9  Celebrate the holiday season with all your fav…
```

Checking the data type of the new dataframe It has one col of integer type reset all are object type

[25]:
```
merged_df.dtypes
```

[25]:
```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

#Merged Data Profiling & Cleaning *counting* the number of cells with empty values in every column

Data Cleaning means the process of identifying incorrect, incomplete, inaccurate, irrelevant, or missing pieces of data and then modifying, replacing, or deleting them as needed. Data Cleansing is considered as the basic element of Data Science.

[26]:
```
print('\nColumns with missing value:')
print(merged_df.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      True
dtype: bool
```

From the info, we know that there are 15641 entries and 12 columns to work with for this EDA. There are a few columns that contain null values, “director,” “cast,” “country,” “date_added,” “rating,”“duration,”“description.”

[27]:
```
pd.isnull(merged_df).sum()
```

[27]:
```
show_id            0
type               0
```

```
title              0
director        6098
cast            4686
country        10887
date_added      9547
release_year       0
rating           863
duration         479
listed_in          0
description        4
dtype: int64
```

Director has more than 40% missing values, if required can be reaplced by unknow(merged_df.director = merged_df.director.fillna('Unknown'))

Country has more than 70% missin data

date_added can to be date time data type

[28]:
```
merged_df.isnull().sum().sum()
```

[28]:
```
32564
```

There are a total of 32564 null values across the entire dataset with 6098 missing points under "director" 4686 under "cast," 10887 under "country," 479 under "duration," 863 under "rating," and "4 under"description.' We will have to handle all null data points before we dive into EDA and modeling.

Imputation is a treatment method for missing value by filling it in using certain techniques. Can use mean, mode, or use predictive modeling. In this module, 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.

15 Process Phase

[29]:
```
merged_df.director.fillna("No Director", inplace=True)
merged_df.cast.fillna("No Cast", inplace=True)
merged_df.country.fillna("Country Unavailable", inplace=True)
merged_df.fillna("rating Unavailable", inplace = True)
merged_df.dropna(subset=["duration", "description"], inplace=True)
```

The easiest way to get rid of them would be to delete the rows with the missing data for missing values. However, this wouldn't be beneficial to our EDA since it 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 description 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.

[30]:
```
print(merged_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
```

[31]:
```
merged_df.columns
```

[31]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

[32]:
```
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 15641 entries, 0 to 9667
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       15641 non-null  object
 1   type          15641 non-null  object
 2   title         15641 non-null  object
 3   director      15641 non-null  object
 4   cast          15641 non-null  object
 5   country       15641 non-null  object
 6   date_added    15641 non-null  object
 7   release_year  15641 non-null  int64
 8   rating        15641 non-null  object
 9   duration      15641 non-null  object
 10  listed_in     15641 non-null  object
 11  description   15641 non-null  object
dtypes: int64(1), object(11)
memory usage: 1.6+ MB
```

[33]:
```
merged_df.corr()
```

```
C:\Users\paperspace\AppData\Local\Temp\1\ipykernel_6512\4191659586.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  merged_df.corr()
```

[33]:
```
              release_year
release_year           1.0
```

[34]:
```
merged_df.describe()
```

[34]:
```
       release_year
count  15641.000000
mean    2008.198645
std       18.481279
min     1920.000000
25%     2006.000000
50%     2015.000000
75%     2019.000000
max     2021.000000
```

Formatting

We will change the value in Type column from 0 into movie and 1 into tv shows.Which might help in data modelling later

change type: 1 = movie, 1 = show Method 1: latter can be used as target varible during model buliding merged_df.loc[merged_df[‘type’] == ‘Movie’, ‘Movie’] = 1 merged_df.loc[merged_df[‘type’] == ‘TV Show’, ‘TV Show’] = 0

16 Perform exploratory Analysis & Visualization

Before we ask questions about the dataset, it would help to explore these columns and better understand how representative the dataset is. This will help us finding any patterns or biases.

[35]:
```
print("Unique values in each column:\n", merged_df.nunique())
```

```
Unique values in each column:
 show_id          9668
type                2
title           13923
director         6293
cast             9113
country           266
date_added       1159
release_year      101
rating            103
duration          231
listed_in        1177
description     13899
dtype: int64
```

17 Perform EDA

[38]:
```
# Count the number of movies and TV shows
print(merged_df['type'].value_counts())

# Find the top 10 countries with the most shows
top_countries = merged_df['country'].value_counts().head(10)
print(top_countries)


# Explore the distribution of ratings
rating_counts = merged_df['rating'].value_counts()
print(rating_counts)

# Identify the most common genres
genres = merged_df['listed_in'].str.split(', ')
all_genres = [genre for sublist in genres for genre in sublist]
common_genres = pd.Series(all_genres).value_counts().head(10)
print( common_genres)
```

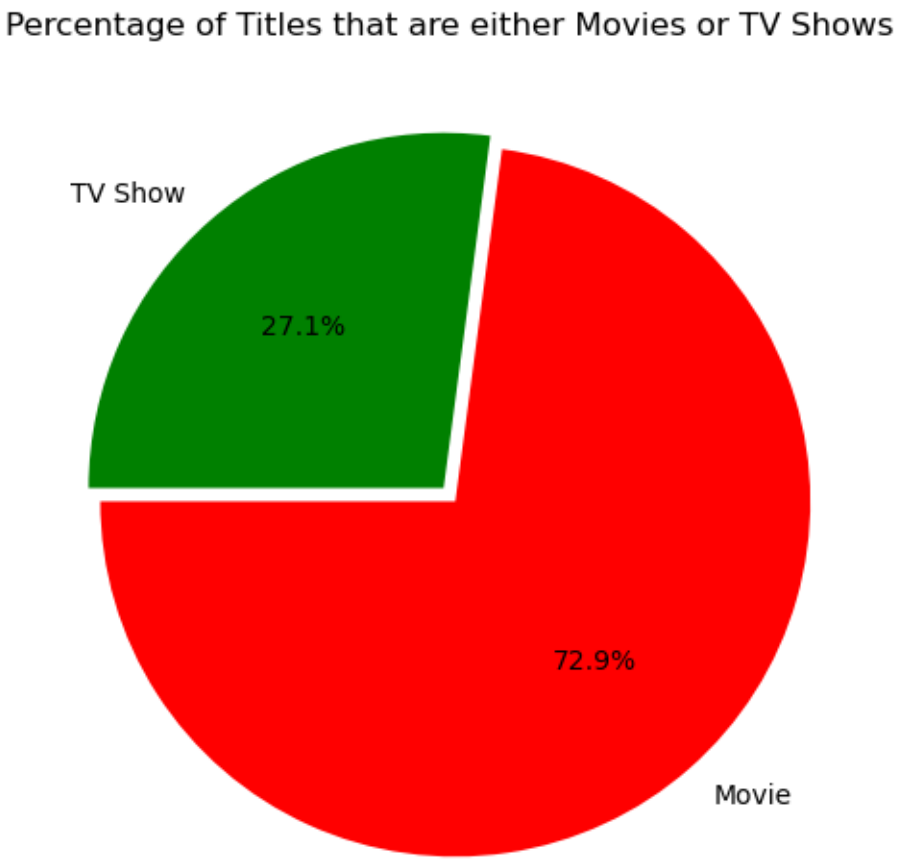
```
Movie      11402
TV Show     4239
Name: type, dtype: int64
Country Unavailable              10887
United States                     3184
Japan                              270
India                              233
United Kingdom                     187
United States, Canada               80
United Kingdom, United States       66
Canada                              62
United States, United Kingdom       61
Canada, United States               59
Name: country, dtype: int64
13+        2117
16+        1547
R          1355
ALL        1268
18+        1243
           …
157 min       1
28 min        1
45 min        1
5 min         1
67 min        1
Name: rating, Length: 103, dtype: int64
Drama               4862
Comedy              3818
```

```
Action              2212
Animation           1631
Kids                1529
Suspense            1501
Family              1469
Documentary         1341
Horror              1179
Special Interest     980
dtype: int64
```

17.1 1.Content By Type,What is the overall distribution of movies and TV shows in the dataset?

Analyze the entire dataset(Netflix,Hulu,Disney& Amazon Prime) 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.

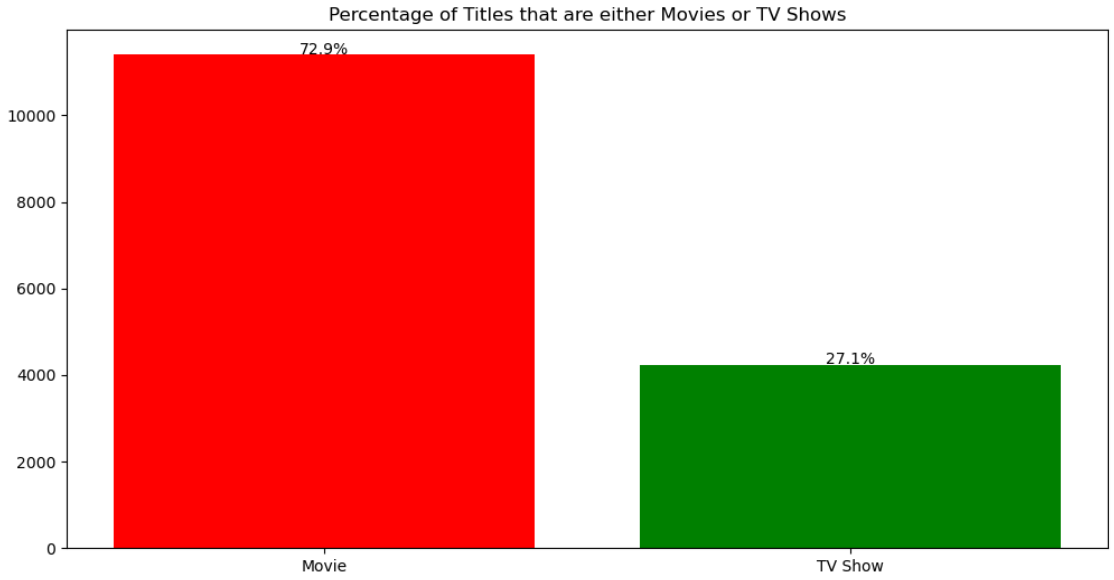
[40]:
```
plt.figure(figsize=(12, 6))
plt.title("Percentage of Titles that are either Movies or TV Shows")

# Calculate the value counts of the 'type' column in the merged dataset
type_counts = merged_df['type'].value_counts()

# Define the labels and colors for the pie chart
labels = type_counts.index
colors = ['red', 'green']

# Create the pie chart
plt.pie(type_counts, explode=(0.025, 0.025), labels=labels, colors=colors,␣
 ↪autopct='%1.1f%%', startangle=180)

plt.show()
```



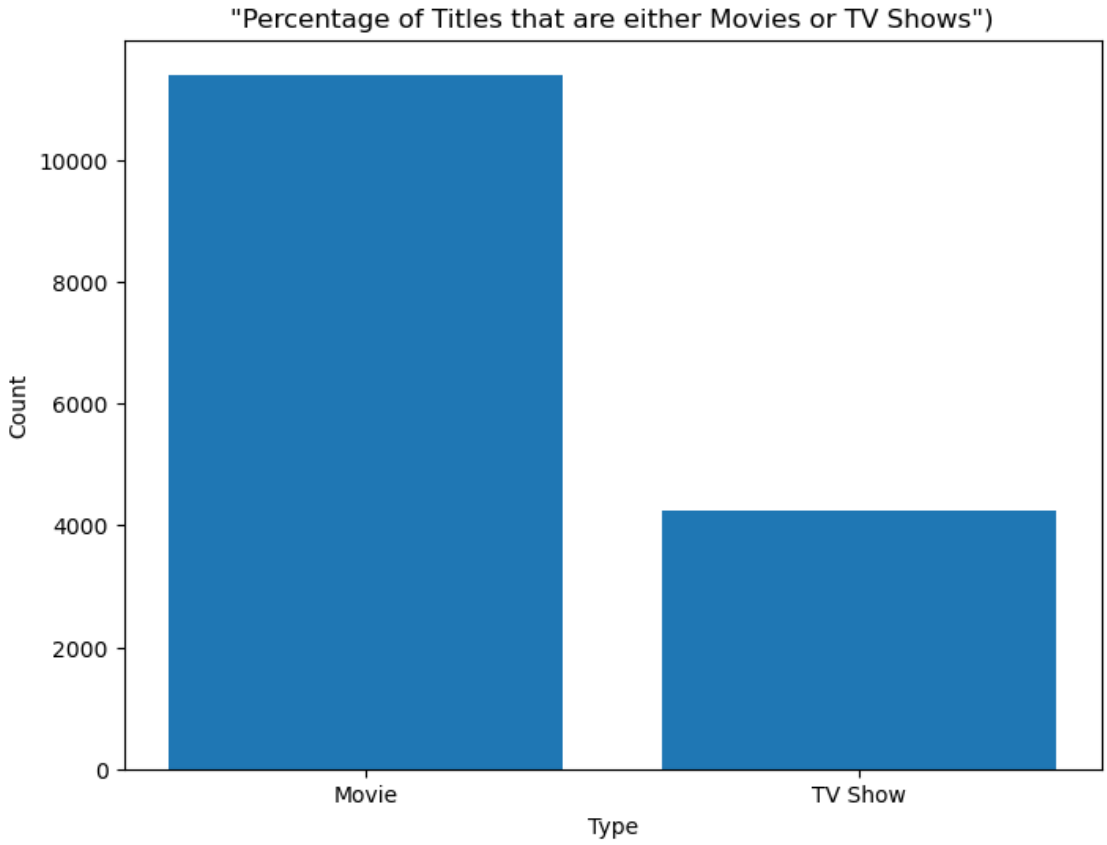
18 Bar chart representation - content by Type

[41]:
```
plt.figure(figsize=(12, 6))
plt.title("Percentage of Titles that are either Movies or TV Shows")

# Calculate the value counts of the 'type' column in the merged dataset
type_counts = merged_df['type'].value_counts()

# Define the labels and colors for the bar chart
labels = type_counts.index
colors = ['red', 'green']

# Create the bar chart
plt.bar(labels, type_counts, color=colors)

# Add percentage labels to the bars
```

```
total = type_counts.sum()
for i, count in enumerate(type_counts):
    percentage = count / total * 100
    plt.text(i, count + 10, f"{percentage:.1f}%", ha='center')

plt.show()
```



[43]:
```
# Create a bar plot for type counts
plt.figure(figsize=(8, 6))
plt.bar(type_counts.index, type_counts.values)
plt.xlabel('Type')
plt.ylabel('Count')
plt.title('"Percentage of Titles that are either Movies or TV Shows")')
plt.show()
```



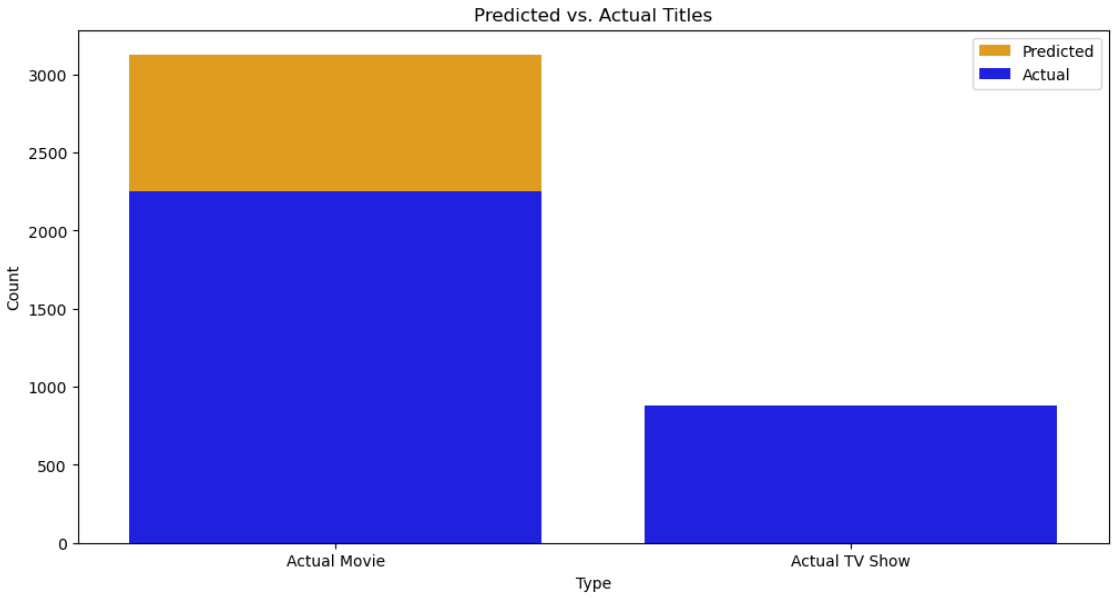
So, there are about8,590 ++ movies and almost 4,000 TV shows, with movies being the majority. There are far more movie titles (72.9%) that TV shows titles (27.1%) in terms of title.

19 Predective model for above Graph

[44]:
```
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.impute import SimpleImputer

# Create a new DataFrame for the prediction model
merged_df.loc[merged_df['type'] == 'Movie', 'Movie'] = 1
merged_df.loc[merged_df['type'] == 'TV Show', 'TV Show'] = 0

# Convert 'type' column to binary values: 1 for 'Movie' and 0 for 'TV Show'
merged_df['target'] = merged_df['type'].map({'Movie': 1, 'TV Show': 0})

# Split the data into features (X) and target variable (y)
X = merged_df[['Movie', 'TV Show']]
```

```
y = merged_df['target']

# Handle missing values in the feature matrix
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_imputed, y, test_size=0.
 ↪2, random_state=42)

# Create and train the logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: {:.2f}%".format(accuracy * 100))

# Create a bar plot of the predicted vs. actual values
plt.figure(figsize=(12, 6))
sns.barplot(x=['Predicted Movie', 'Predicted TV Show'], y=[(y_pred == 1).sum(),␣
 ↪(y_pred == 0).sum()], color='orange', label='Predicted')
sns.barplot(x=['Actual Movie', 'Actual TV Show'], y=[(y_test == 1).sum(),␣
 ↪(y_test == 0).sum()], color='blue', label='Actual')
plt.title("Predicted vs. Actual Titles")
plt.xlabel("Type")
plt.ylabel("Count")
plt.legend()
plt.show()
```

```
Accuracy: 71.94%
```



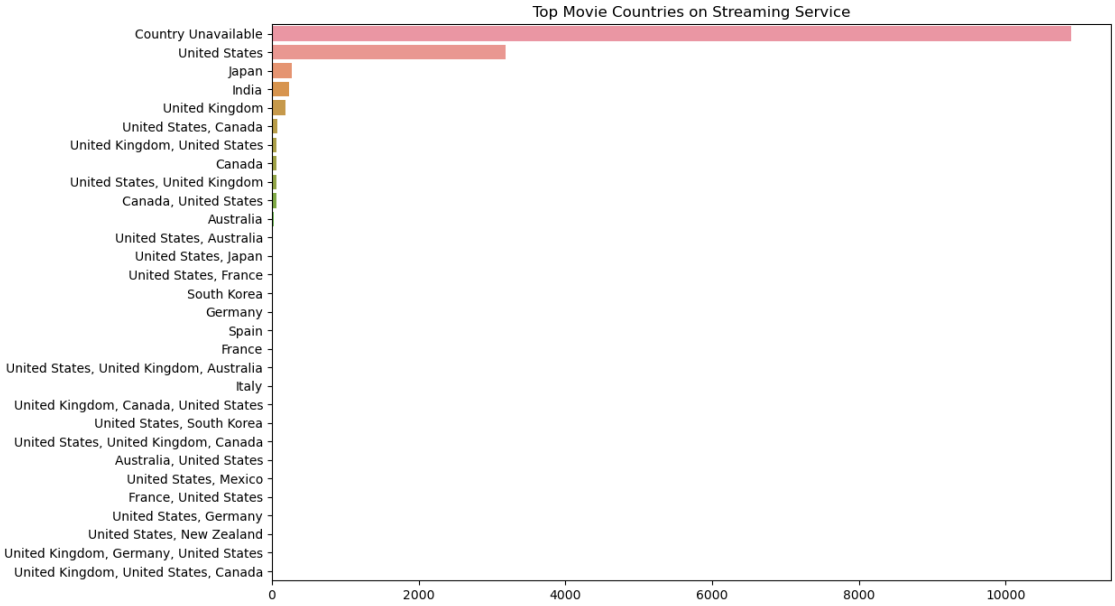
Logistic regression is a statistical algorithm used for binary classification problems. It is a type of regression analysis that is used to model the relationship between a set of input variables (also known as features or independent variables) and a binary target variable (also known as the dependent variable or the outcome).

The goal of logistic regression is to estimate the probability that an instance belongs to a certain class (e.g., whether an email is spam or not spam). It calculates a linear combination of the input variables and applies a non-linear function (called the logistic function or sigmoid function) to the result. The logistic function maps the linear combination to a value between 0 and 1, representing the probability of the instance belonging to the positive class.Logistic regression is a statistical algorithm used for binary classification problems. It is a type of regression analysis that is used to model the relationship between a set of input variables (also known as features or independent variables) and a binary target variable (also known as the dependent variable or the outcome).x

19.1 2.Counting and assigning the 30 top countires to check streaming services used. Any regional or cultural factors that contribute to the success of movies from specific countries

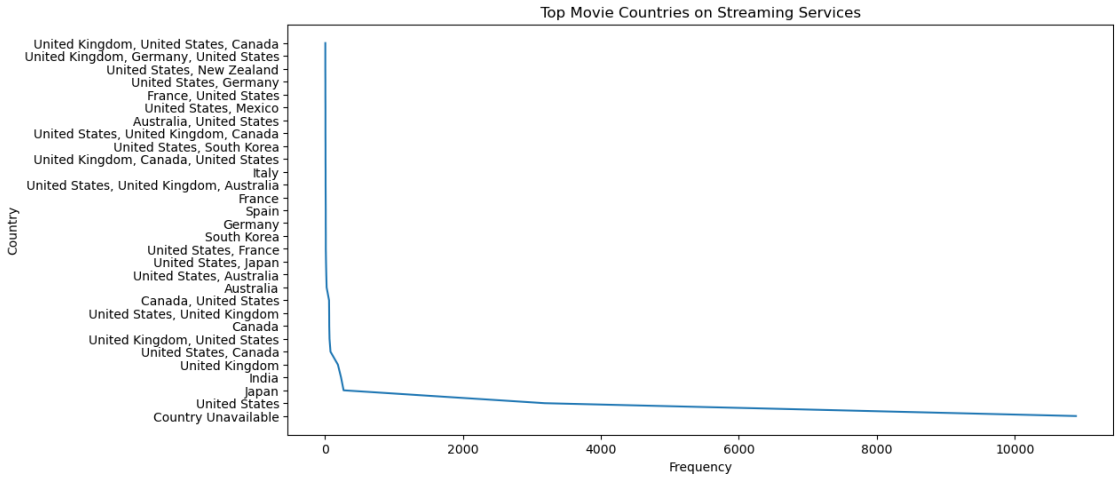
[45]:
```
country = merged_df.country.value_counts().head(30)
plt.figure(figsize=(12,8))
plt.title('Top Movie Countries on Streaming Service')
sns.barplot(x=country.values, y=country.index)
```

[45]:
```
<Axes: title={'center': 'Top Movie Countries on Streaming Service'}>
```



20 Plot Graph -Counting and assigning the 30 top countires to check streaming serices used.

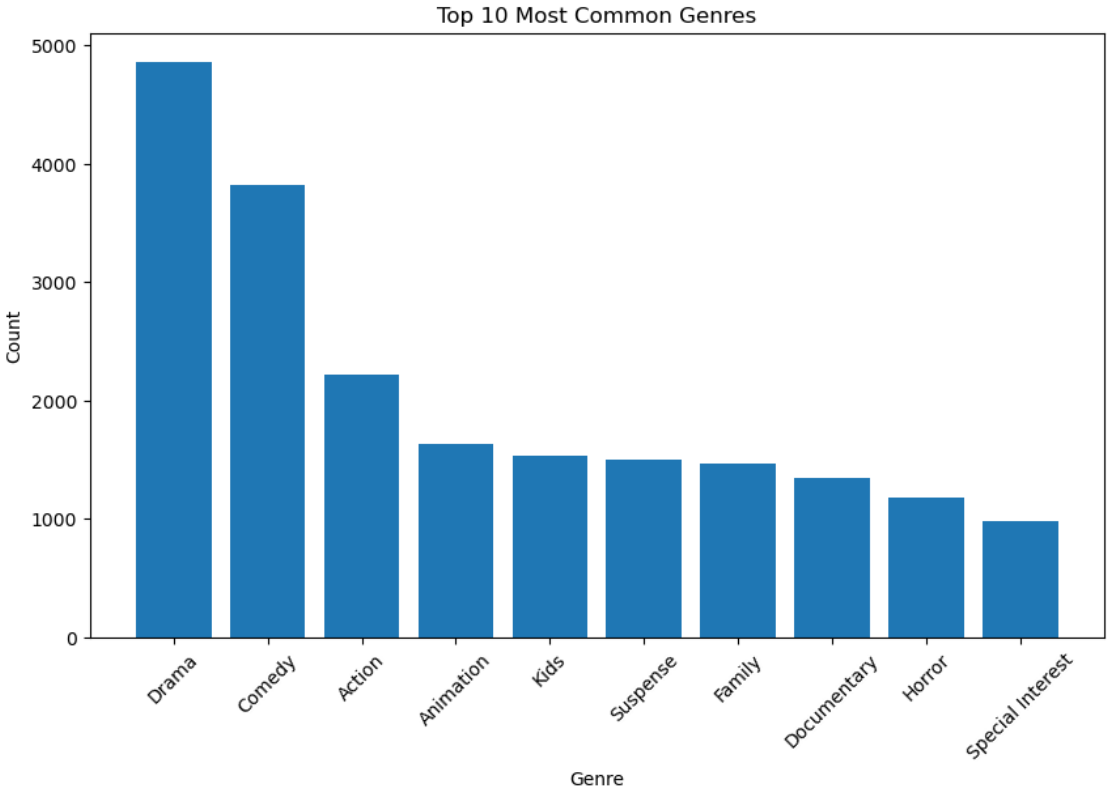
[47]:
```
import matplotlib.pyplot as plt

country = merged_df['country'].value_counts().head(30)

plt.figure(figsize=(12, 6))
plt.title('Top Movie Countries on Streaming Services')
plt.plot(country.values, country.index)
plt.xlabel('Frequency')
plt.ylabel('Country')

plt.show()
```



Clearly, the number is not very misrepresented. It is important to consider this as Netflix, Prime Video, Hulu and Disney+ are all american services.Asian countries also make use of different streaming services not popular in America

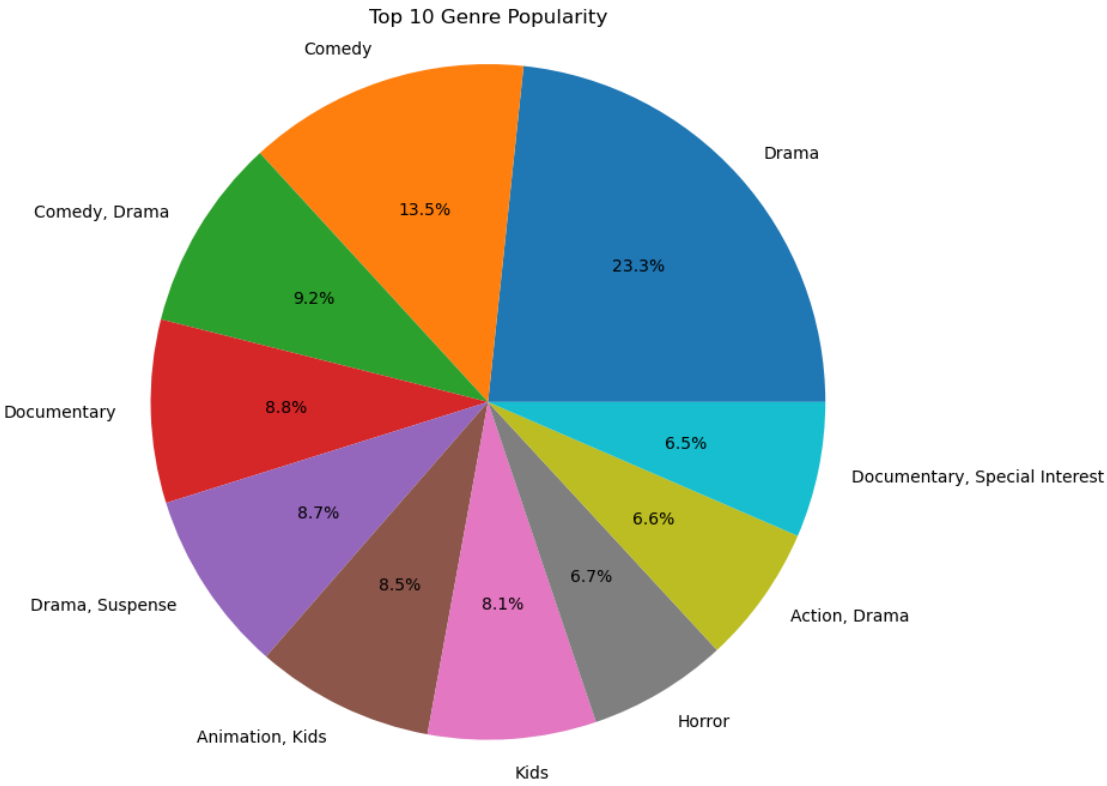
20.1 3. Top Genres ,any dominant genres that significantly outweigh others in terms of popularity

[48]:
```
# Create a bar plot for common genres
plt.figure(figsize=(10, 6))
plt.bar(common_genres.index, common_genres.values)
plt.xlabel('Genre')
plt.ylabel('Count')
plt.title('Top 10 Most Common Genres')
plt.xticks(rotation=45)
plt.show()
```



21 Pie Chart -Top Generes

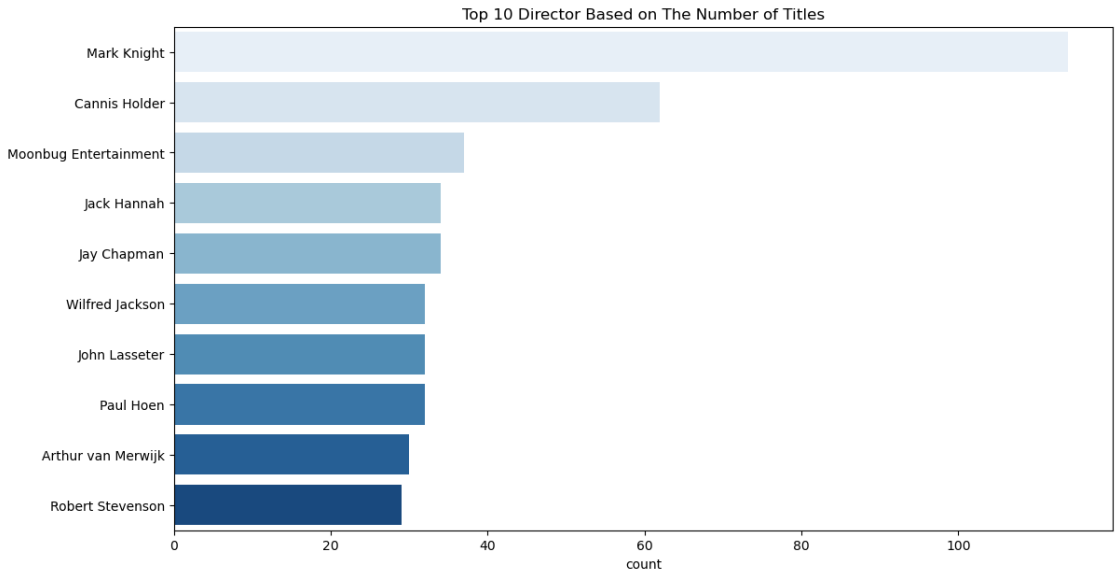
[49]:
```
all_data = merged_df
genre_counts = all_data['listed_in'].value_counts()
top_20_genres = genre_counts.head(10)

plt.figure(figsize=(8, 8))
plt.pie(top_20_genres, labels=top_20_genres.index, autopct='%1.1f%%')
plt.title('Top 10 Genre Popularity')
plt.axis('equal')
plt.show()
```



From the graph, we know that dramas and comedies takes the top row.

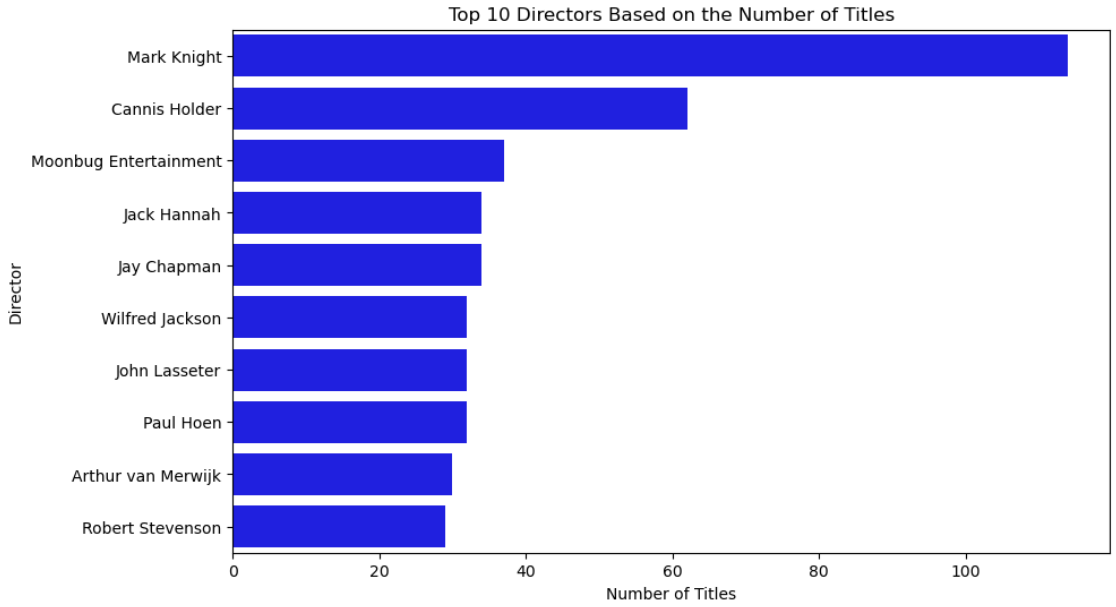
4. Top Directors ,any specific directors who are consistently successful in terms of the number of titles

[50]:
```
filtered_directors = merged_df[merged_df.director != 'No Director'].
 ↪set_index('title').director.str.split(', ', expand=True).stack().
 ↪reset_index(level=1, drop=True)
plt.figure(figsize=(13,7))
plt.title('Top 10 Director Based on The Number of Titles')
sns.countplot(y = filtered_directors, order=filtered_directors.value_counts().
 ↪index[:10], palette='Blues')
plt.show()
```



Barplot -Top Directors

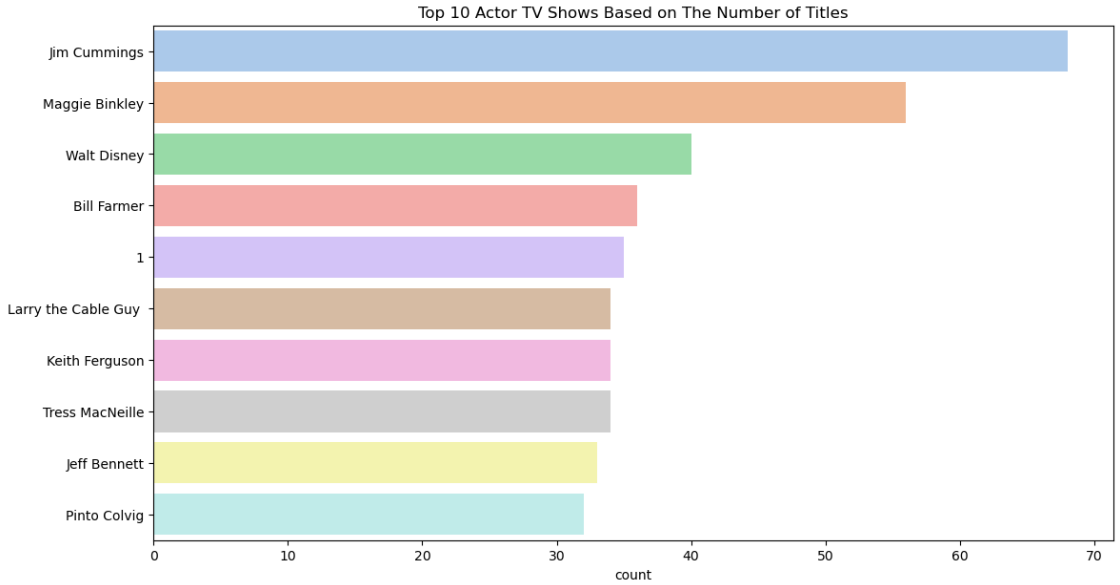
[51]:
```
filtered_directors = merged_df[merged_df.director != 'No Director'].
 ↪set_index('title').director.str.split(', ', expand=True).stack().
 ↪reset_index(level=1, drop=True)

plt.figure(figsize=(10, 6))
plt.title('Top 10 Directors Based on the Number of Titles')

# Calculate the director counts
director_counts = filtered_directors.value_counts().
 ↪sort_values(ascending=False)[:10]

# Create a horizontal bar chart using Seaborn
sns.barplot(x=director_counts, y=director_counts.index, color='blue')

plt.xlabel('Number of Titles')
plt.ylabel('Director')

plt.show()
```



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

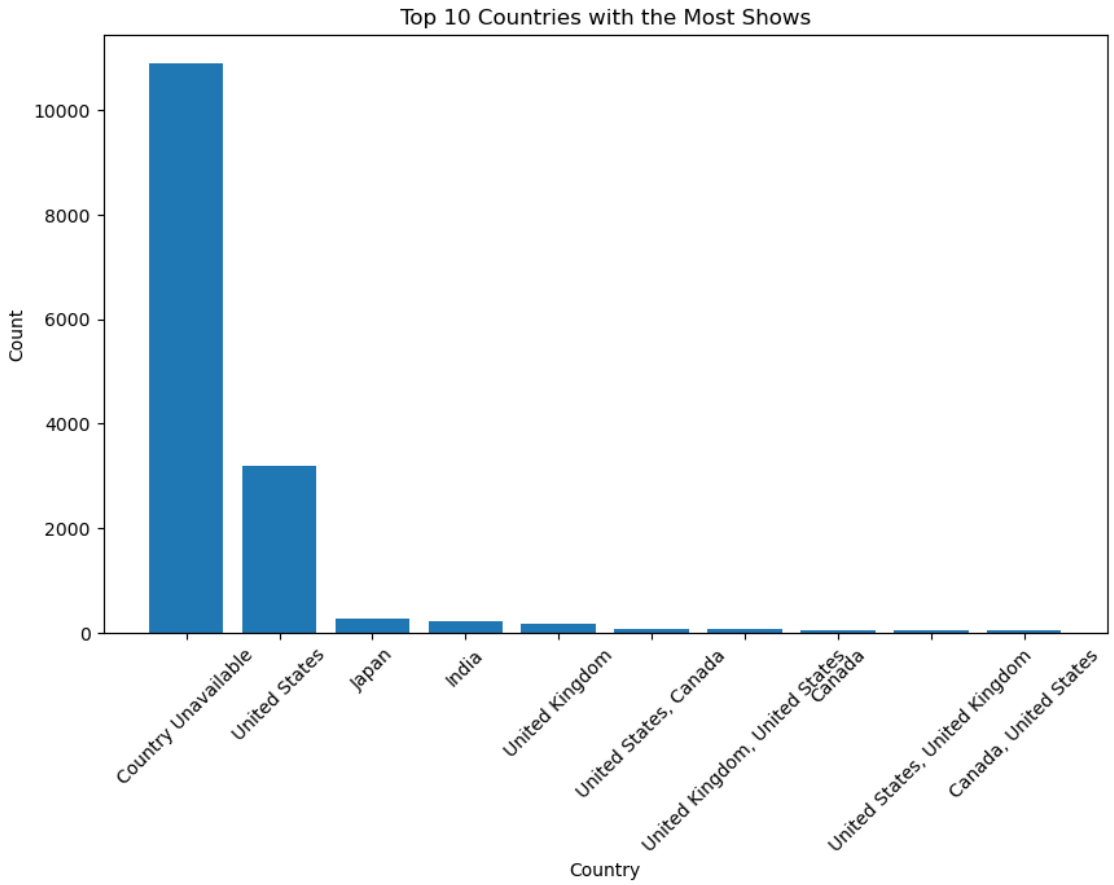
Top Actor on based on the number of titles

[52]:
```
filtered_cast_shows = merged_df[merged_df.cast != 'No Cast'].set_index('title').
  ↪cast.str.split(', ', expand=True).stack().reset_index(level=1, drop=True)
plt.figure(figsize=(13,7))
plt.title("Top 10 Actor TV Shows Based on The Number of Titles")
sns.countplot(y=filtered_cast_shows, order=filtered_cast_shows.value_counts().
  ↪index[:10], palette='pastel')
plt.show()
```



Jim Cummings is the best Actor based on TV Shows

[54]:
```
# Create a bar plot for top countries
plt.figure(figsize=(10, 6))
plt.bar(top_countries.index, top_countries.values)
plt.xlabel('Country')
plt.ylabel('Count')
plt.title('Top 10 Countries with the Most Shows')
plt.xticks(rotation=45)
plt.show()
```



United States has most of the shows.

22 Ask & answer questions about the data

Ask at least 4 interesting questions about your dataset

1.What is the overall distribution of movies and TV shows in the dataset?

There are about8,590 ++ movies and almost 4,000 TV shows, with movies being the majority. There are far more movie titles (72.9%) that TV shows titles (27.1%) in terms of title.

2.Are there any regional or cultural factors that contribute to the success of movies from specific countries?

It is important to consider this as Netflix, Prime Video, Hulu and Disney+ are all american services.Asian countries also make use of different streaming services not popular in America.Clearly, the number is not very misrepresented.

3.Are there any dominant genres that significantly outweigh others in terms of popularity?

From the graph, we know that dramas and comedies takes the top row.

4.Are there any specific directors who are consistently successful in terms of the number of titles they have directed?

The most popular director with the most titles, is mainly international.Mark Knight!

23 Summarize your inferences & write a conclusion

We have drawn many interesting inferences from the merged dataset to answer business questions. In conclusion, the dataset provides valuable information about the distribution of movies and TV shows, the influence of regional and cultural factors, the popularity of different genres, and the success of specific directors. This information can be further analyzed and utilized for business decisions such as content acquisition, production partnerships, and audience targeting strategies.

Future Improvements:

One possible future improvement could be conducting analysis and model building based on different service providers to understand the variations in user preferences and behavior across platforms.

References:

You can refer to the scikit-learn documentation for information on linear models: cikit-learn.org/stable/search.html?q=linear+model