

**Netflix Dataset Analysis**

**Table of Contents**

1. Introduction
2. Exploratory Data Analysis
3. Result and Findings
4. Dashboard
5. Conclusion

**Introduction**

This dataset provides a comprehensive listing of all the movies and TV shows available on Netflix. The dataset includes various details for each title, which can be useful for data analysis and insights into Netflix's content library. Key features of the dataset include:

1. **Title**: The name of the movie or TV show.
2. **Type**: Specifies whether the listing is a movie or a TV show.
3. **Director**: The director of the movie or TV show.
4. **Cast**: The main actors and actresses involved in the movie or TV show.
5. **Country**: The country where the movie or TV show was produced.
6. **Date Added**: The date when the movie or TV show was added to Netflix.
7. **Release Year**: The year the movie or TV show was originally released.
8. **Rating**: The rating given to the movie or TV show (e.g., PG, TV-MA).
9. **Duration**: The duration of the movie in minutes or the number of seasons for TV shows.
10. **Genres**: The genres the movie or TV show belongs to.
11. **Description**: A brief description or synopsis of the movie or TV show.

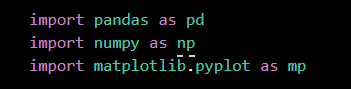
As of mid-2021, Netflix had over 200 million subscribers globally. As of the most recent data in 2024, Netflix's subscriber count has increased to approximately 238 million subscribers worldwide. This continued growth reflects Netflix's ongoing expansion and popularity as a leading streaming service.

**Exploratory data Analysis**

I used Python, Pandas and Matlplotlib to perform an exploratory data analysis (EDA) on the Netflix dataset, which provides insights into various aspects of the data, such as duration, ratings, and potential outliers. Below is a step-by-step explanation of the procedures followed:

1. **Importing Necessary Libraries:**

* I start by importing essential libraries: `pandas` for data manipulation ]and `matplotlib.pyplot` for visualization.



2. **Loading the Dataset:**

* The dataset is loaded into a pandas DataFrame using the `pd.read\_csv` function, specifying the file path and encoding as 'utf-8'.

df=pd.read\_csv(r"C:\Users\PMLS\Downloads\netflix1.csv",encoding='utf-8')

3. **Initial Data Inspection:**

* The first few rows of the dataset are displayed using `df.head()`, providing a preliminary view of the data structure.

df.head()

* The `df.info()` method is used to get a concise summary of the DataFrame, including the data types and non-null values for each column.

df.info()

* Statistical summaries of numerical columns are obtained using `df.describe()`, giving insights into central tendency, dispersion, and distribution shape.

df.describe()

4**. Data Cleaning and Transformation:**

* The `duration` column is split into two separate columns: `duration` and `duration\_Unit`. This is achieved by splitting the string values in the `duration` column on spaces and extracting the respective parts.

df['duration\_Unit']=df['duration'].str.split(" ").str[1]

df['duration']=df['duration'].str.split(" ").str[0]

* The original `duration` column is then converted to a numeric type using `pd.to\_numeric` with `errors='coerce'` to handle any non-numeric values gracefully, converting them to NaNs if necessary.

df['duration'] = pd.to\_numeric(df['duration'], errors='coerce')

* The transformed DataFrame is inspected again using `df.head()` and `df.info()` to ensure the changes were correctly applied.

df.info()

df.head()

5. **Checking for Duplicates:**

* The number of duplicate rows in the dataset is counted using `df.duplicated().sum()`, ensuring data integrity by identifying and potentially removing redundant entries.

df.duplicated().sum()

6. **Data Type Verification:**

* The data types of the DataFrame columns are verified using `df.dtypes` to ensure that numerical operations can be performed correctly on the `duration` column.

df.dtype

7. **Data Visualization:**

* A histogram of the `duration` column is created using `mp.hist(df['duration'])` to visualize the distribution of content durations. The x-axis is labeled "Duration" for clarity.

mp.hist(df['duration'])

mp.xlabel("Duration")

mp.show()

* A scatter plot is generated to explore the relationship between `rating` and `duration`. The plot is displayed using `display(mp.scatter(df['rating'], df['duration']))` with a large figure size for better visibility.

mp.figure(figsize=(10, 20))

display(mp.scatter(df['rating'],df['duration']))

mp.show()

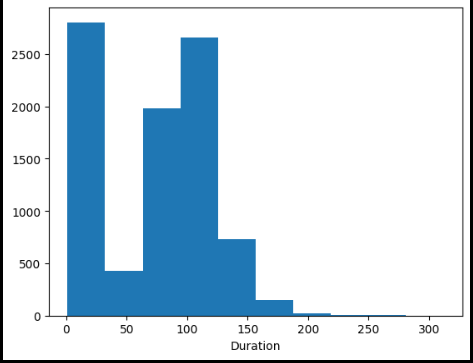
* A box plot of the `duration` column is plotted using `mp.boxplot(df['duration'])` to identify potential outliers and understand the spread of duration values.

mp.boxplot(df['duration'])

mp.show()

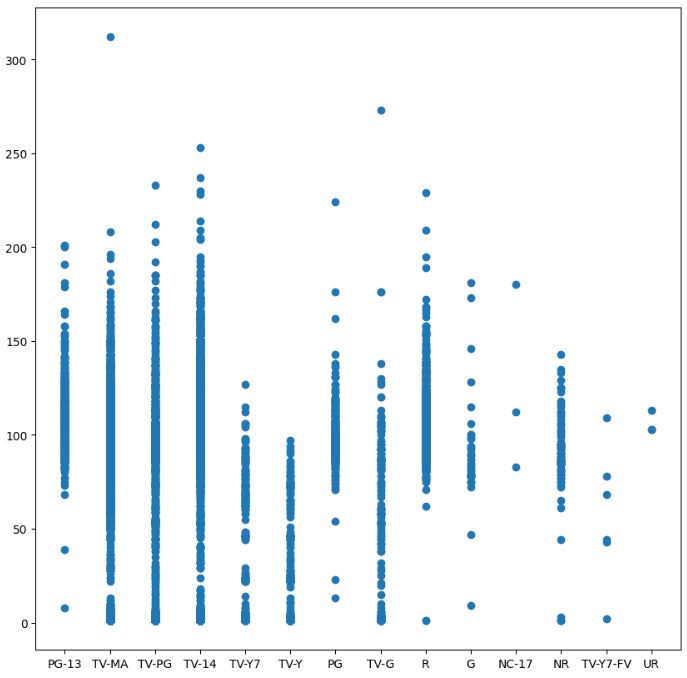
8. **Visualizing the Histogram:**

* The histogram of the `duration` column shows the frequency distribution of different content durations, helping to identify common length ranges for movies and TV shows on Netflix.



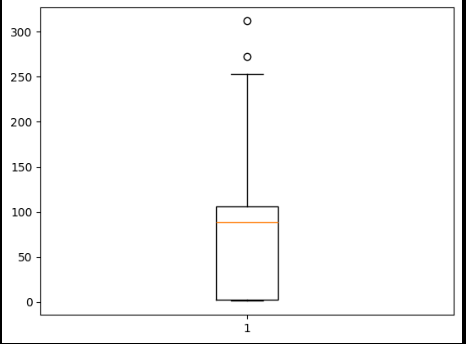
9**. Scatter Plot Analysis:**

* The scatter plot of `rating` versus `duration` provides insights into how content duration varies across different rating categories. It helps in understanding if certain ratings are associated with longer or shorter content



10. **Box Plot Analysis:**

* The box plot of the `duration` column highlights the median, quartiles, and potential outliers. This visual representation aids in identifying unusual duration values that deviate significantly from the typical range.



These steps collectively provide a comprehensive exploratory data analysis of the Netflix dataset, offering valuable insights into its structure, distribution, and potential anomalies. This EDA serves as a foundation for further in-depth analysis and data-driven decision-making.

# 

**Results and Findings**

The analysis was conducted using SQL. Data was cleaned through Python and its libraries, and later Power BI was used mainly for visualization purposes by creating charts. Meanwhile, the main analysis was conducted using SQL. Different queries and commands were used during the analysis, namely SELECT, TOP 10, COUNT, WHERE ,GROUP BY etc.

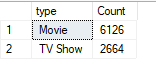
1. **Total Movie and Show Count.**

create view TotalCount\_Movie\_TVShows

as

select type,count(type)as [Count] from netflixTB group by type

select \* from TotalCount\_Movie\_TVShows



This query creates a view that counts the number of entries for each type (either 'Movie' or 'TV Show') in the netflixTB table. The result will show two rows, one for 'Movie' and one for 'TV Show', with their respective counts.

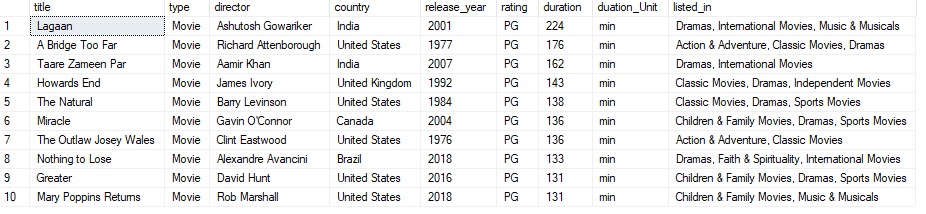
1. **Top 10 Parent-Guided Movies**

create view Top10\_ParentGuided\_Movies

as

select top(10) title,type,director,country,release\_year,rating,duration,duation\_Unit,listed\_in from netflixTB where rating like 'PG%' order by duration DESC

select \* from Top10\_ParentGuided\_Movies



This query creates a view of the top 10 parent-guided movies (those with ratings starting with 'PG') ordered by duration in descending order. The result will display the longest parent-guided movies.

1. **Average Release Year and Max/Min Duration by Type**

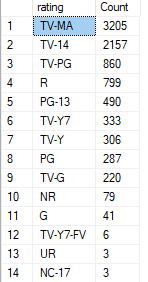
select type,avg(release\_year) as AVG\_ReleaseYear,max(duration) as MAX\_Duration ,min(duration) as MIN\_Duration from netflixTB group by type



This query calculates the average release year, maximum duration, and minimum duration for each type (Movie or TV Show). The result will show these statistics for each type.

1. **Rating Count**

select rating,Count(rating) as Count from netflixTB group by rating order by count(rating) DESC



This query counts the number of entries for each rating and orders the results by the count in descending order. The result will show the most common ratings first.

1. **Top 10 Adult TV Shows**

select top(10) title,type,country,release\_year,rating,duration,duation\_Unit,listed\_in from netflixTB where rating in ('TV-MA','R','NC-17') and duation\_Unit like 'Season%'

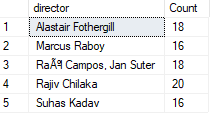
order by duration DESC



This query selects the top 10 adult TV shows (with ratings 'TV-MA', 'R', or 'NC-17') that are organized by the number of seasons, ordered by duration in descending order. The result will display the longest adult TV shows by season duration.

1. **Movies/Shows by Director with 15 to 20 Titles**

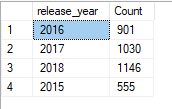
select distinct director,count(title) as Count from NetflixTB group by director having count(\*) between 15 and 20



This query identifies directors who have directed between 15 and 20 titles in the netflixTB table. The result will show each qualifying director and the number of titles they have directed.

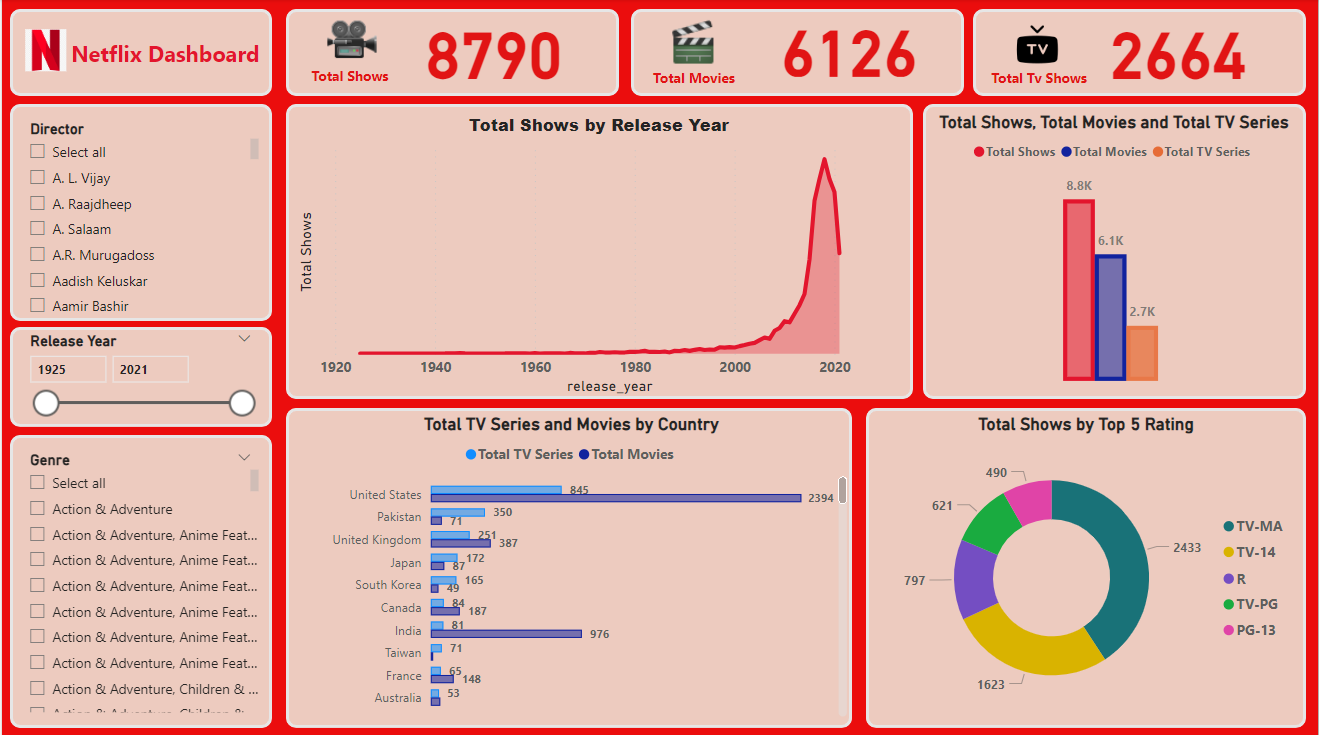
1. **Releases by Year between 2015 and 2018**

select release\_year,count(\*) as Count from netflixTB where release\_year between '2015' and '2018' group by release\_year having count(\*)>500



This query counts the number of releases per year between 2015 and 2018 where the count of releases is greater than 500. The result will display the years within the specified range that meet the criteria.

**Dashboard**



**Conclusion**

In this capstone project, we conducted an extensive exploratory data analysis (EDA) of the Netflix dataset, leveraging SQL queries and Python programming to extract meaningful insights. Our analysis began with a comprehensive understanding of the dataset's structure and continued with cleaning and transforming the data to enhance its usability.

Through various SQL queries, we identified critical patterns, such as the distribution of content types, top parent-guided movies, and adult TV shows. We examined the average release year and duration metrics for different content types and assessed the prevalence of various ratings. Additionally, our visualizations, including histograms, scatter plots, and box plots, provided a deeper understanding of content duration trends and potential outliers.

Our findings revealed the diverse landscape of Netflix's offerings, highlighting key metrics that can inform content strategy and user recommendations. By identifying directors with significant contributions and analyzing release trends over recent years, we gained valuable insights into content production and viewer preferences.

This analysis not only showcased our ability to handle and interpret large datasets but also demonstrated the power of combining SQL and Python for robust data analysis. The methodologies and insights derived from this project lay a strong foundation for future data-driven decision-making and strategic planning in the entertainment industry.