🎬Neflix Movies and TV Shows Analysis Project

📌Project Summary:

This project analyzes Netflix content using Python and data science techniques. The dataset contains information about movies and TV shows, including genres, ratings, cast, and release years. The goal is to extract insights on trends, top-rated content, genre popularity, and country-wise distribution. The project also includes interactive visualizations, recommendation systems, and sentiment analysis of movie descriptions.

🔍Insights from the Analysis:

1) **Rating Column Issue:**
   The rating column was missing, requiring data cleaning and re-adding before analysis.

2) **Hover Data Issue on Bar Graph:**
   Initially, the bar graph lacked hover data. The issue was resolved by adding hover_data=["title", "release_year", "rating"] to the plotly.express graph.

3) **Most Watched Movie:**
   The highest-rated movie was determined by sorting the dataset based on ratings.

4) **Most Famous Actor:**
   The most frequent actor was identified by counting appearances across multiple movies/shows.

5) **Top 10 Movies by Year:**
   For each release year, the top 10 movies were identified based on ratings.

6) **Trend Analysis of Netflix Content:**
   A line graph showed that Netflix has steadily increased the number of movies and TV shows over the years.

7) **Genre Popularity Over Time:**
   Top 10 genres were visualized to understand which categories gained popularity over different years.

8) **Top 10 Successful Directors:**
The highest-rated directors were identified and displayed in a horizontal bar chart.

9) **Netflix Originals vs. Non-Originals:**
A box plot compared ratings of Netflix Originals vs. Non-Originals, revealing whether Netflix-produced content performs better.

10) **Country-wise Netflix Preferences:**
A world map visualization showed the distribution of Netflix content across different countries.

11) **Sentiment Analysis of Movie Descriptions:**
Sentiment scores were derived from descriptions, and a histogram displayed the distribution of positive and negative sentiments.

12) **Movie Recommendations:**
A content-based recommendation system using TF-IDF and cosine similarity suggested similar movies based on a selected title.

📚 Libraries Used:

1. **pandas** - Data manipulation and preprocessing
2. **numpy** - Numerical computations
3. **matplotlib** - Data visualization (static plots)
4. **seaborn** - Enhanced visualization with statistical plotting
5. **plotly.express** - Interactive visualizations
6. **collections.Counter** - Counting occurrences of actors in multiple movies/shows
7. **textblob** - Sentiment analysis of movie descriptions
8. **sklearn.feature_extraction.text** - TF-IDF vectorization for content-based recommendations
9. **sklearn.metrics.pairwise** - Cosine similarity for recommendations

🛠️ Errors and solutions

1) Missing rating Column Issue

The rating column was missing or improperly removed, causing issues in your analysis. If the column was deleted or not loaded correctly, any operation using df['rating'] would throw a KeyError.
If it contained null (NaN) values, calculations like df['rating'].mean() would give unexpected results.

2) Hover Data on Bar Graph Missing (Release Year, Rating, Title)
When hovering over the bars in a graph, only some information (like count or rating) might be shown instead of release year, rating, and title.
This happens because the visualization function does not pass all required columns as hover data.
Matplotlib does not support interactive tooltips by default, so it needs plotly or seaborn.
Possible Causes:
The plotting library (Matplotlib/Seaborn) doesn't support hover effects natively.
The hover_data argument was missing in Plotly Express.
Data formatting issues prevented values from appearing correctly.

3)Data Type Conversion Issues Problem: Converting date_added to DateTime
If there are any invalid date formats, pd.to_datetime() will throw a ValueError.
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')

4)Problem: Extracting Numbers from duration
Some values in the duration column may contain text like 'Seasons' instead of minutes (for TV shows).
Extracting numbers using .astype(float) will throw a ValueError.
df['duration'] = df['duration'].str.extract('(\d+)').astype(float)
df['duration'].fillna(0, inplace=True) # Replace NaN values with 0

5) Text Overlapping in Plots
In bar plots with a large number of categories, labels will overlap.
plt.xticks(rotation=45, ha='right')