```
import pandas as pd
import numpy as np
import plotly.express as px
from textblob import TextBlob
```

Netflix Content Analysis – Exploratory Data Analysis (EDA)

Overview In this notebook, I explore the Netflix dataset to uncover meaningful patterns and insights related to content type, production trends, ratings, contributors, and sentiments. These insights help in understanding the dataset and guide the feature engineering and modeling process in later steps.

```
df=pd.read_csv('netflix_titles.csv (1).zip',encoding='latin1')
```

df.shape

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8809 entries, 0 to 8808
Data columns (total 26 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   show_id       8809 non-null   object 
 1   type          8809 non-null   object 
 2   title         8809 non-null   object 
 3   director      6175 non-null   object 
 4   cast          7984 non-null   object 
 5   country       7978 non-null   object 
 6   date_added    8799 non-null   object 
 7   release_year  8809 non-null   int64  
 8   rating        8805 non-null   object 
 9   duration      8806 non-null   object 
 10  listed_in     8809 non-null   object 
 11  description   8809 non-null   object 
 12  Unnamed: 12   0 non-null      float64
 13  Unnamed: 13   0 non-null      float64
 14  Unnamed: 14   0 non-null      float64
 15  Unnamed: 15   0 non-null      float64
 16  Unnamed: 16   0 non-null      float64
 17  Unnamed: 17   0 non-null      float64
 18  Unnamed: 18   0 non-null      float64
 19  Unnamed: 19   0 non-null      float64
 20  Unnamed: 20   0 non-null      float64
 21  Unnamed: 21   0 non-null      float64
 22  Unnamed: 22   0 non-null      float64
 23  Unnamed: 23   0 non-null      float64
 24  Unnamed: 24   0 non-null      float64
 25  Unnamed: 25   0 non-null      float64
dtypes: float64(14), int64(1), object(11)
memory usage: 1.7+ MB
```

Data Cleaning

The dataset contained missing values and inconsistent data across several columns. To prepare it for analysis, I:

Filled missing values with descriptive placeholders such as:

director: “No director specified”

cast: “No cast specified”

country: “Unknown”

rating: “Not rated”

duration: “Not available”

date_added: “Unknown”

- Checked for and removed any duplicate rows to ensure data integrity.
- Converted the date_added column to datetime format using pd.to_datetime for easier analysis of time-based trends.
- Dropped irrelevant or overly incomplete columns that would not add value to the analysis.

These steps helped ensure the dataset was clean, structured, and ready for exploration and modeling.

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
df['director']=df['director'].fillna('Director not specified')
df['country'] = df['country'].fillna('Unknown')
df['date_added'] = df['date_added'].fillna('Unknown')
df['rating'] = df['rating'].fillna('Not Rated')
df['duration'] = df['duration'].fillna('Not Available')
df['cast'] = df['cast'].fillna('No cast specified')
```

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

```
df = df.drop(columns=[col for col in df.columns if 'Unnamed' in col])
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descriptio
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No cast specified	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her fathe nears th end of h life, filmm
1	s2	TV Show	Blood & Water	Director not specified	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	Aft crossin paths at party, a Cap Town t
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protec his fami from powerf drug lor
3	s4	TV Show	Jailbirds New Orleans	Director not specified	No cast specified	Unknown	2021-09-24	2021	TV-MA	1 Season	Docuseries, Reality TV	Feud flirtation and toil talk go dow amo
4	s5	TV Show	Kota Factory	Director not specified	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city c coachin center known t train I

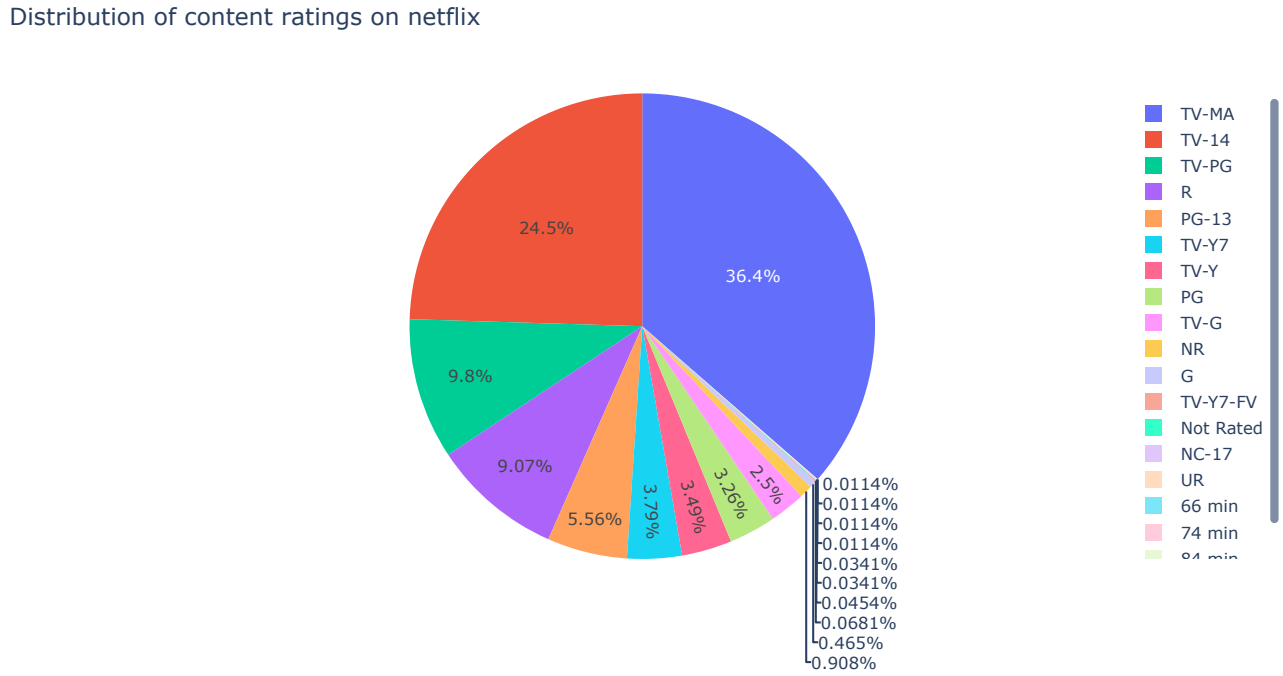
Next steps: Generate code with df | View recommended plots | New interactive sheet

```
x = df.groupby(['rating']).size().reset_index(name='counts')
```

Distribution of Content Ratings on Netflix (Pie Chart)

Here, I analyze how Netflix's content is distributed across different ratings (e.g., TV-MA, PG, etc.). The pie chart helps understand audience targeting and content guidelines across various age groups.

```
piechart=px.pie(x,values='counts',names='rating',title= 'Distribution of content ratings on netflix')
piechart.show()
```




```
directors_list= pd.DataFrame()
print(directors_list)
```

```
Empty DataFrame
Columns: []
Index: []
```

```
directors_list=df['director'].str.split(',',expand=True).stack()
print(directors_list)
```

```
0     0           Kirsten Johnson
1     0    Director not specified
2     0           Julien Leclercq
3     0    Director not specified
4     0    Director not specified
                   ...
8804  0           Ruben Fleischer
8805  0              Peter Hewitt
8806  0               Mozez Singh
8807  0              Yeon Sang-ho
8808  0              Susanne Bier
Length: 9614, dtype: object
```

```
directors_list=directors_list.to_frame()
print(directors_list)
```

```
                            0
0    0         Kirsten Johnson
1    0  Director not specified
2    0         Julien Leclercq
3    0  Director not specified
4    0  Director not specified
...                        ...
8804 0         Ruben Fleischer
8805 0            Peter Hewitt
8806 0             Mozez Singh
8807 0            Yeon Sang-ho
8808 0            Susanne Bier

[9614 rows x 1 columns]
```

```
directors_list.columns=['Director']
print(directors_list)
```

```
                     Director
0    0         Kirsten Johnson
1    0  Director not specified
2    0         Julien Leclercq
3    0  Director not specified
4    0  Director not specified
```

```
...                       ...
8804 0         Ruben Fleischer
8805 0            Peter Hewitt
8806 0             Mozez Singh
8807 0            Yeon Sang-ho
8808 0            Susanne Bier

[9614 rows x 1 columns]
```

Double-click (or enter) to edit

```
directors=directors_list.groupby(['Director']).size().reset_index(name='Total Count')
print(directors)
```

```
                      Director  Total Count
0                Aaron Moorhead            2
1                   Aaron Woolf            1
2      Abbas Alibhai Burmawalla            1
3               Abdullah Al Noor            1
4           Abhinav Shiv Tiwari            1
...                         ...          ...
5117               Ã agan Irmak            1
5118         Ã sold UggadÃ³ttir            1
5119      Ã skar ThÃ³r Axelsson            1
5120          Ã mer Faruk Sorak            2
5121             Å enol SÃ¶nmez            2

[5122 rows x 2 columns]
```

```
directors=directors[directors.Director!='Director not specified']
print(directors)
```

```
                      Director  Total Count
0                Aaron Moorhead            2
1                   Aaron Woolf            1
2      Abbas Alibhai Burmawalla            1
3               Abdullah Al Noor            1
4           Abhinav Shiv Tiwari            1
...                         ...          ...
5117               Ã agan Irmak            1
5118         Ã sold UggadÃ³ttir            1
5119      Ã skar ThÃ³r Axelsson            1
5120          Ã mer Faruk Sorak            2
5121             Å enol SÃ¶nmez            2

[5121 rows x 2 columns]
```

```
directors=directors.sort_values(by=['Total Count'])
print(directors)
```

```
            Director  Total Count
3189  Mandeep Kumar            1
3199  Manish Tiwary            1
3198   Manish Saini            1
3196   Manish Gupta            1
3195  Manika Sharma            1
...             ...          ...
3236   Marcus Raboy           16
4652    Suhas Kadav           16
261       Jan Suter           18
4068   RaÃºl Campos           18
4021  Rajiv Chilaka           22

[5121 rows x 2 columns]
```

```
directors=directors.sort_values(by=['Total Count'],ascending=False)
print(directors)
```

```
                    Director  Total Count
4021           Rajiv Chilaka           22
4068            RaÃºl Campos           18
261                Jan Suter           18
4652             Suhas Kadav           16
3236            Marcus Raboy           16
...                      ...          ...
3914           Phil Sgriccia            1
3916        Philip Barantini            1
3917  Philip Einstein Lipski            1
3884              Peter Lord            1
3851         Pavel Kostomarov            1

[5121 rows x 2 columns]
```

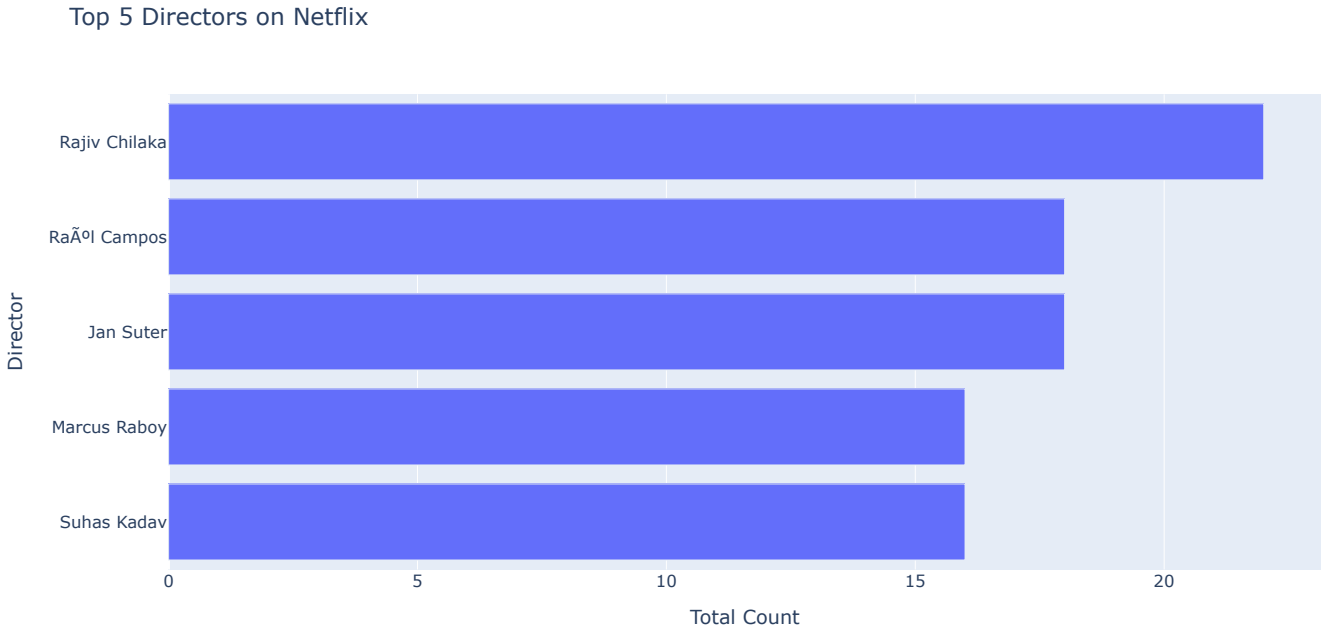
Top 5 Directors on Netflix (Bar Chart)

This visualization shows the directors with the most content on Netflix. It helps us identify key contributors and popular creators influencing Netflix's content library.

```
Top5Directors =directors.head()
print(Top5Directors)
```

```
            Director  Total Count
4021  Rajiv Chilaka           22
4068   RaÃºl Campos           18
261       Jan Suter           18
4652     Suhas Kadav           16
3236   Marcus Raboy           16
```

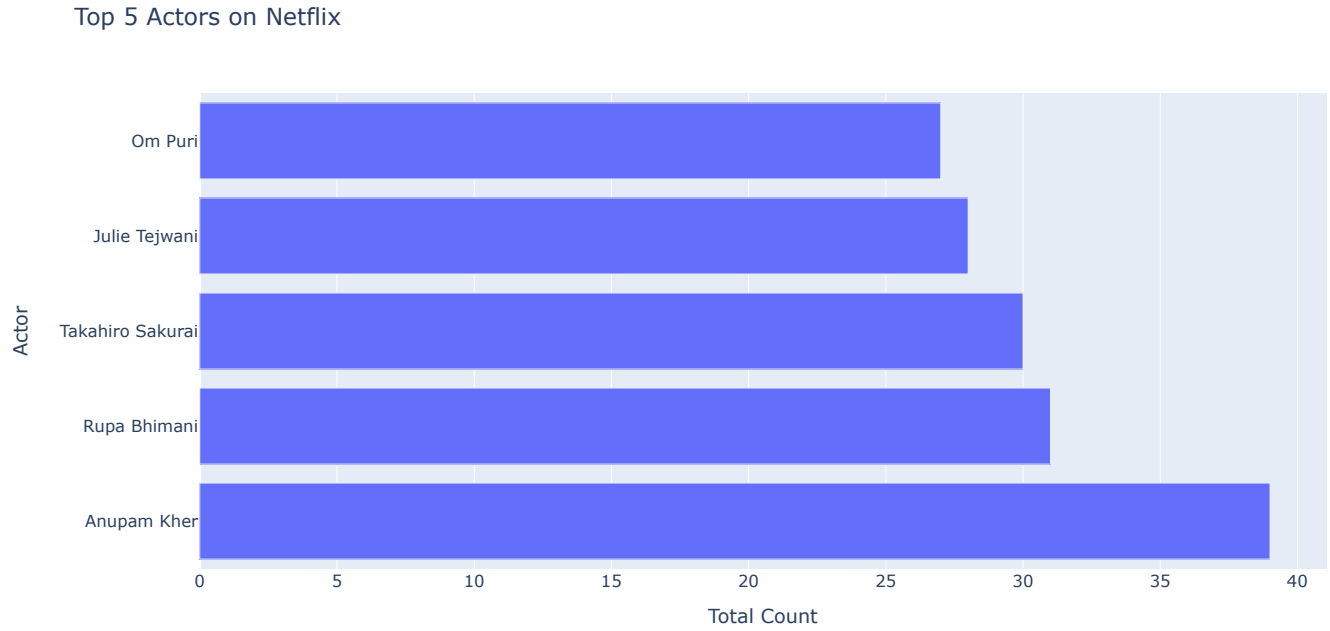
```
Top5Directors =Top5Directors.sort_values(by='Total Count')
Barchart=px.bar(Top5Directors,x='Total Count',y='Director',title='Top 5 Directors on Netflix')
Barchart.show()
```



Top 5 Actors on Netflix (Bar Chart)

Similar to directors, this chart highlights the actors who appear most frequently in Netflix content, offering insights into star power and casting trends.

```
cast_df=pd.DataFrame()
castdf=df['cast'].str.split(',',expand=True).stack()
castdf=castdf.to_frame()
castdf.columns=['Actor']
actors=castdf.groupby(['Actor']).size().reset_index(name='Total Count')
actors=actors[actors.Actor!='No cast specified']
actors=actors.sort_values(by=['Total Count'],ascending=False)
Top5Actors=actors.head()
TopActors=Top5Actors.sort_values(by=['Total Count'])
Barchart2=px.bar(Top5Actors,x= 'Total Count',y='Actor',title='Top 5 Actors on Netflix')
Barchart2.show()
```



Trend of Content Production Over the Years (Line Chart using px.line)

I explore how Netflix's content library has evolved over time by plotting yearly trends in content production. This helps identify growth patterns, peaks, and strategic expansion periods.

```
df1=df[['type', 'release_year']]
df1=df1.rename(columns={"release_year": "Release Year","type":"Type"})
df2=df1.groupby(['Release Year','Type']).size().reset_index(name='Total Count')
```

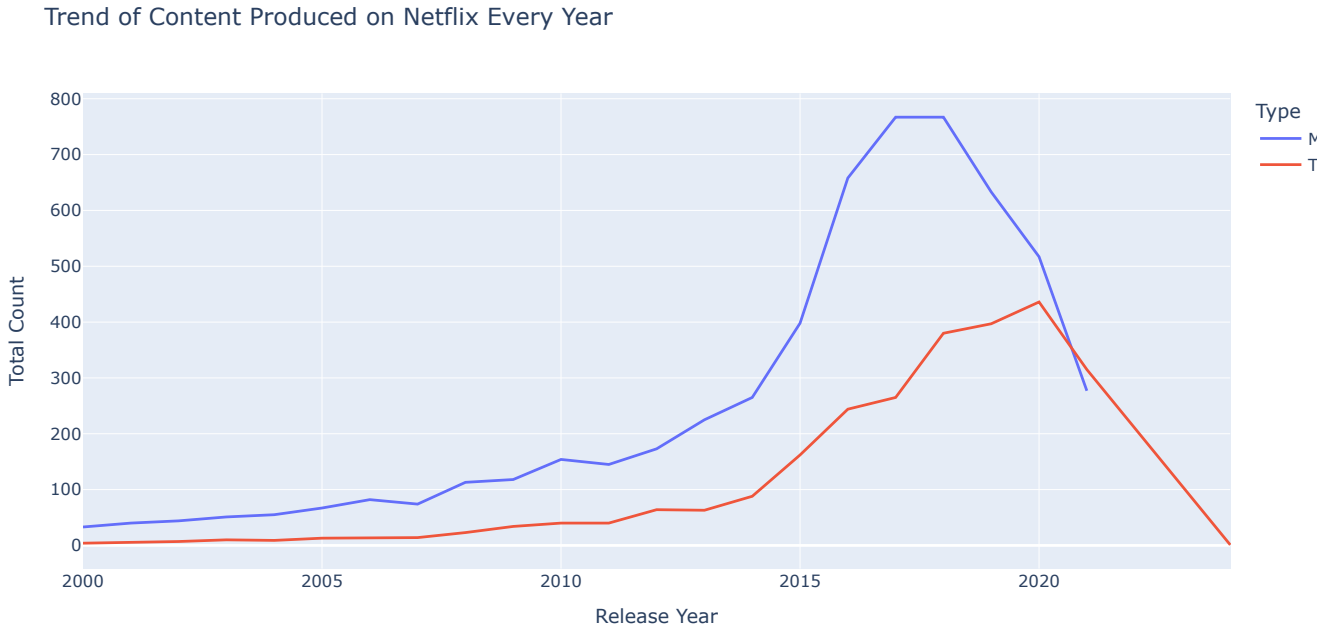
```
print(df2)
```

```
     Release Year     Type  Total Count
0            1925  TV Show            1
1            1942    Movie            2
2            1943    Movie            3
3            1944    Movie            3
4            1945    Movie            3
..            ...      ...          ...
115          2020    Movie          517
116          2020  TV Show          436
117          2021    Movie          277
118          2021  TV Show          315
119          2024  TV Show            1

[120 rows x 3 columns]
```

Double-click (or enter) to edit

```
df2=df2[df2["Release Year"]>=2000]
graph=px.line(df2,x= 'Release Year' ,y='Total Count',color='Type',title='Trend of Content Produced on Netflix Every Year')
graph.show()
```

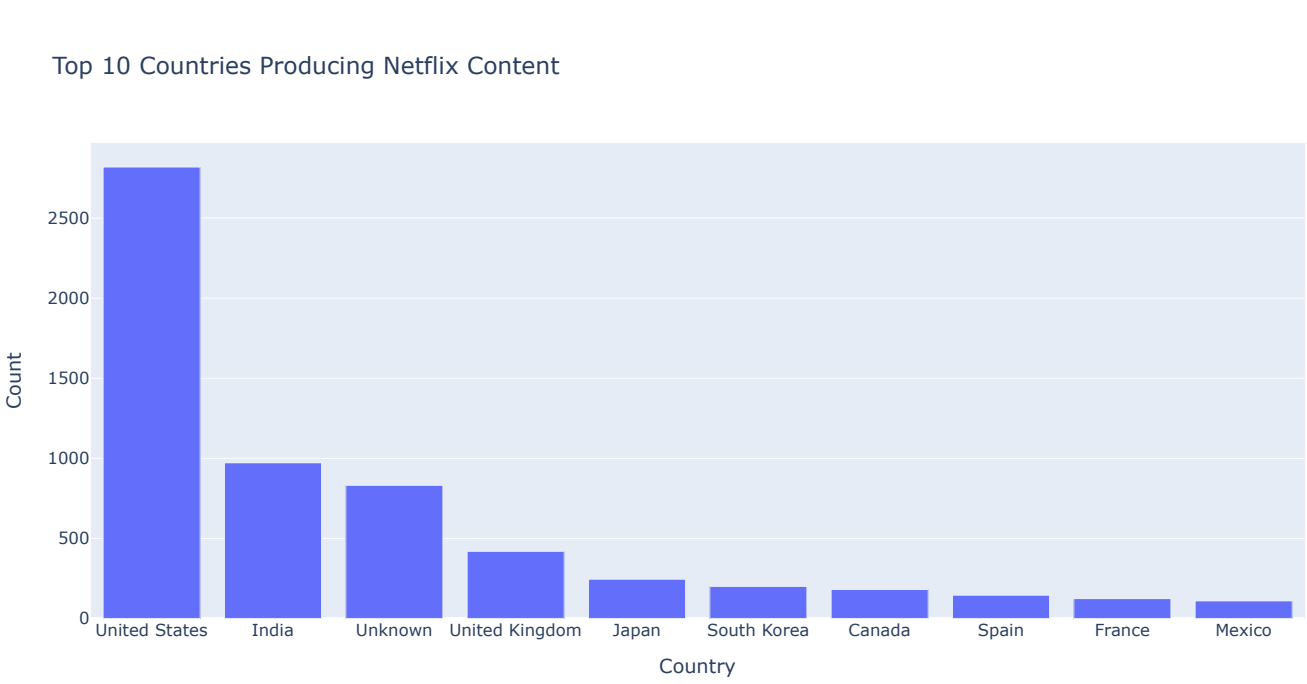


Top 10 Countries Producing Netflix Content (Bar Chart)

This section explores which countries contribute the most content to Netflix. The bar chart helps us understand geographic trends and regional focus areas.

```
top_countries = df['country'].value_counts().head(10).reset_index()
top_countries.columns = ['Country', 'Count']
```

```
countries = px.bar(top_countries, x='Country', y='Count', title="Top 10 Countries Producing Netflix Content")
countries.show()
```

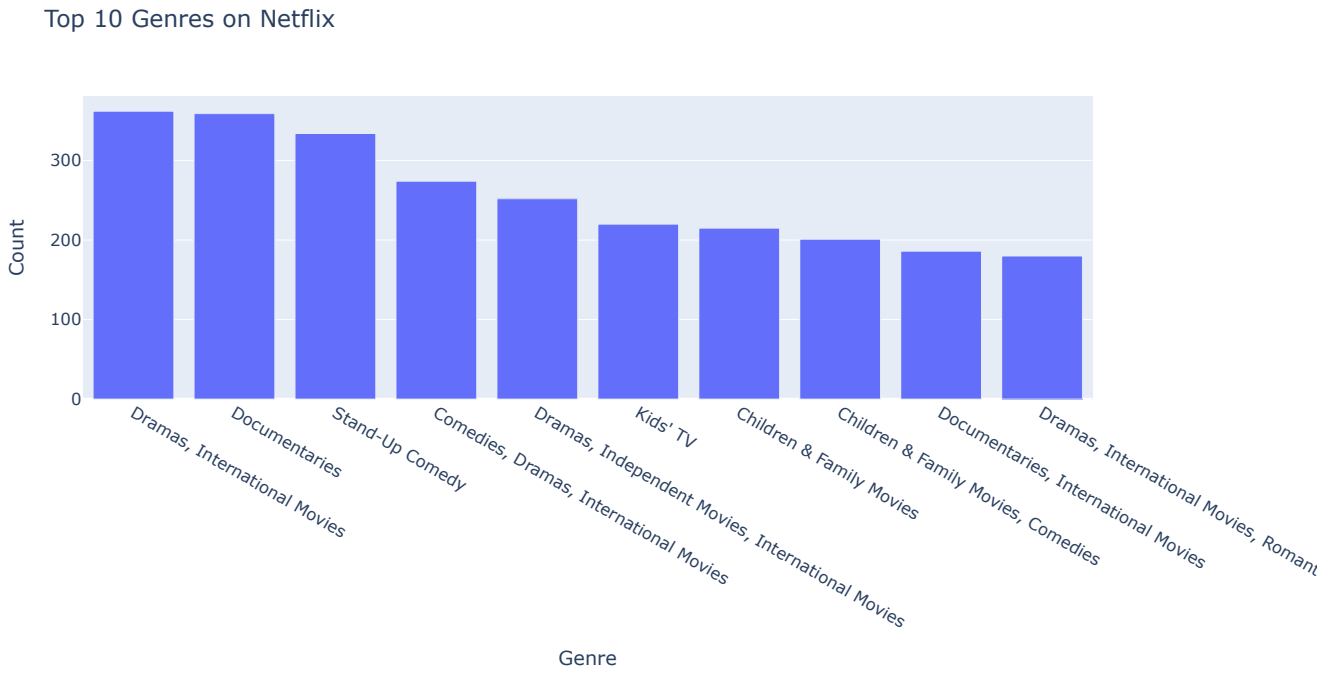


Top 10 Genres on Netflix (Bar Chart)

This visualization shows the most popular genres available on Netflix. Understanding genre distribution helps in identifying content preferences and user engagement areas.

```
top_genres= df['listed_in'].value_counts().head(10).reset_index()
top_genres.columns=['Genre','Count']


genres=px.bar(top_genres,x='Genre',y= 'Count',title= 'Top 10 Genres on Netflix')
genres.show()
```

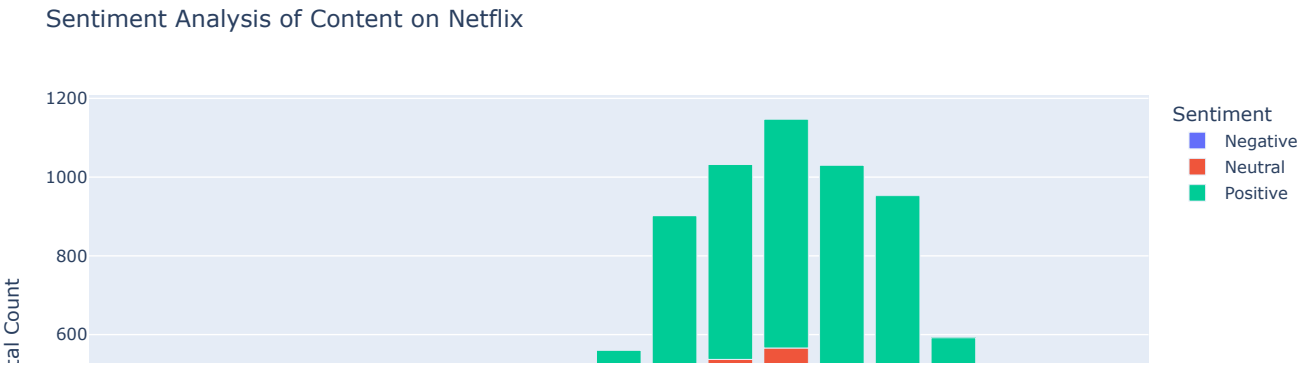


Sentiment Analysis of Netflix Content (Bar Chart using px.bar)

Using sentiment analysis on content descriptions, I analyze whether content has a more positive, neutral, or negative sentiment. This gives insights into audience appeal and marketing tone used across different titles.

```
df3=df[['release_year' ,'description']]
df3=df3.rename(columns={'release_year':'Release Year','description':'Description'})
for index,row in df3.iterrows():
  d=row ['Description']
  testimonial=TextBlob(d)
  p=testimonial.sentiment.polarity
  if p==0:
    sent='Neutral'
  elif p>0:
      sent='Positive'
  else:
        sent='Negative'
  df3.loc[[index,2],'Sentiment']=sent

df3=df3.groupby(['Release Year','Sentiment']).size().reset_index(name='Total Count')
df3=df3[df3['Release Year']>2005]
bargraph=px.bar(df3,x='Release Year',y='Total Count',color='Sentiment',title='Sentiment Analysis of Content on Netflix')
bargraph.show()
```


Sentiment Analysis of Content on Netflix
1200
1000
800
600
Sentiment
Negative
Neutral
Positive