**Data analysis of Netflix in various contests**

Netflix is one of the largest providers of online streaming services. It collects a huge amount of data because it has a very large subscriber base. In this article, I’m going to introduce you to a data science project on Netflix data analysis with Python.

**Netflix Data Analysis**

We can analyze a lot of data and models from Netflix because this platform has consistently focused on changing business needs by shifting its business model from on-demand DVD movie rental and now focusing a lot about the production of their original shows.

In this article, I’ll take a look at some very important models of Netflix data to understand what’s best for their business. Some of the most important tasks that we can analyze from Netflix data are:

1. understand what content is available
2. understand the similarities between the content
3. understand the network between actors and directors
4. what exactly Netflix is focusing on
5. and sentiment analysis of content available on Netflix.

**Netflix Data Analysis with Python**

The dataset I use for the Netflix data analytics task consists of TV shows and movies streamed on Netflix as of 2019. The dataset is provided by [Flixable](https://flixable.com/) which is an engine of third-party research available on Netflix.

I’ll start this Netflix data analysis task with Python by importing the dataset and all the Python libraries needed for this task:

|  |  |
| --- | --- |
|  | import numpy as np # linear algebra |
|  | import pandas as pd # for data preparation |
|  | import plotly.express as px # for data visualization |
|  | from textblob import TextBlob # for sentiment analysis |
|  |  |
|  | dff=pd.read\_csv('netflix\_titles.csv') |
|  | dff.shape |

(6234, 12)

So the data consists of 6234 rows and 12 columns, now let’s look at the column names:

1

dff.columns

Index(['show\_id', 'type', 'title', 'director', 'cast', 'country', 'date\_added',

'release\_year', 'rating', 'duration', 'listed\_in', 'description'],

dtype='object')

#### **Distribution of Content:**

To begin the task of analyzing Netflix data, I’ll start by looking at the distribution of content ratings on Netflix:

1

z = dff.groupby(['rating']).size().reset\_index(name='counts')

2

pieChart = px.pie(z, values='counts', names='rating',

3

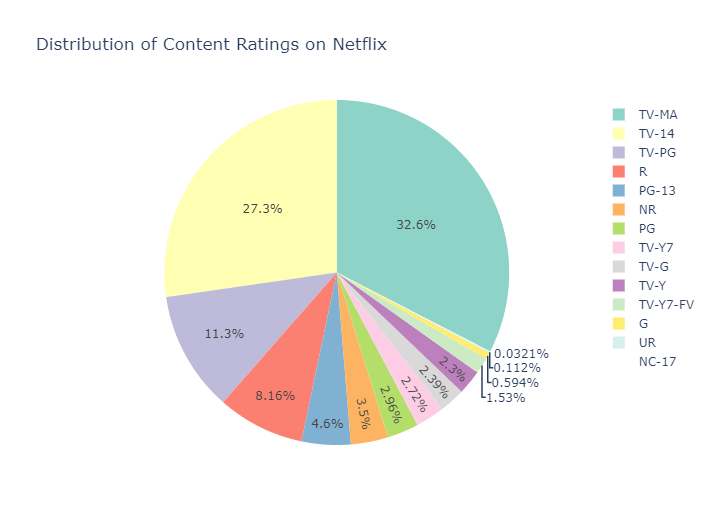
title='Distribution of Content Ratings on Netflix',

4

color\_discrete\_sequence=px.colors.qualitative.Set3)

5

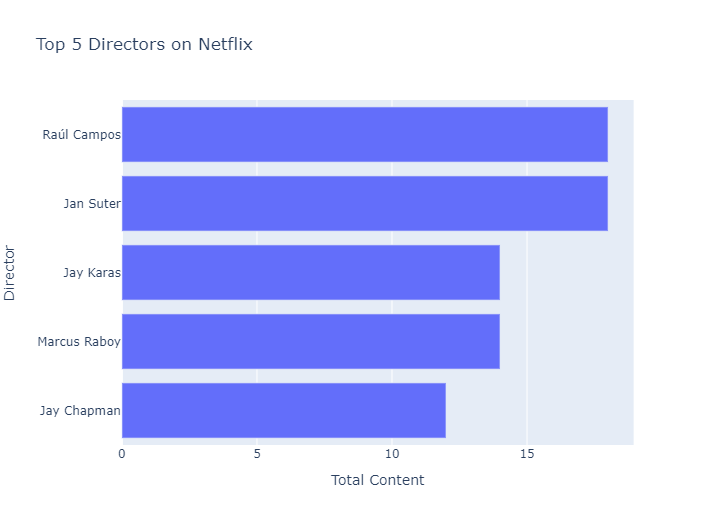
pieChart.show()



The graph above shows that the majority of content on Netflix is categorized as “TV-MA”, which means that most of the content available on Netflix is intended for viewing by mature and adult audiences.

Now let’s see the top 5 successful directors on this platform:

|  |  |
| --- | --- |
|  | dff['director']=dff['director'].fillna('No Director Specified') |
|  | filtered\_directors=pd.DataFrame() |
|  | filtered\_directors=dff['director'].str.split(',',expand=True).stack() |
|  | filtered\_directors=filtered\_directors.to\_frame() |
|  | filtered\_directors.columns=['Director'] |
|  | directors=filtered\_directors.groupby(['Director']).size().reset\_index(name='Total Content') |
|  | directors=directors[directors.Director !='No Director Specified'] |
|  | directors=directors.sort\_values(by=['Total Content'],ascending=False) |
|  | directorsTop5=directors.head() |
|  | directorsTop5=directorsTop5.sort\_values(by=['Total Content']) |
|  | fig1=px.bar(directorsTop5,x='Total Content',y='Director',title='Top 5 Directors on Netflix') |
|  | fig1.show() |

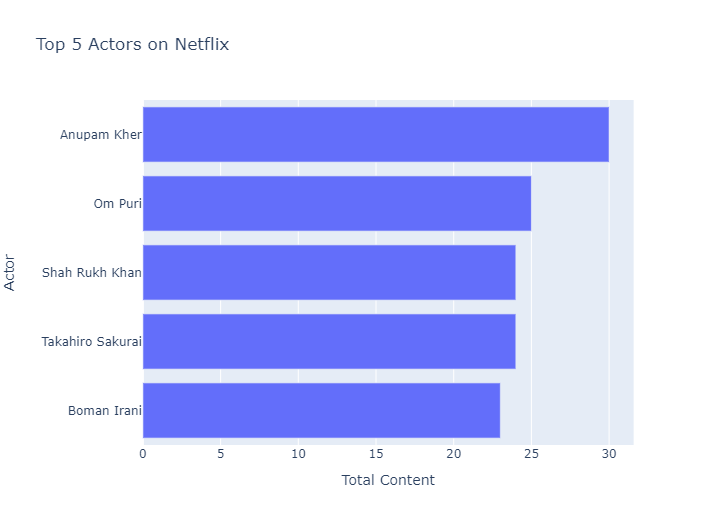


From the above graph it is derived that the top 5 directors on this platform are:

1. Raul Campos
2. Jan Suter
3. Jay Karas
4. Marcus Raboy
5. Jay Chapman

Now let’s have a look at the top 5 successful actors on this platform:

|  |  |
| --- | --- |
|  | dff['cast']=dff['cast'].fillna('No Cast Specified') |
|  | filtered\_cast=pd.DataFrame() |
|  | filtered\_cast=dff['cast'].str.split(',',expand=True).stack() |
|  | filtered\_cast=filtered\_cast.to\_frame() |
|  | filtered\_cast.columns=['Actor'] |
|  | actors=filtered\_cast.groupby(['Actor']).size().reset\_index(name='Total Content') |
|  | actors=actors[actors.Actor !='No Cast Specified'] |
|  | actors=actors.sort\_values(by=['Total Content'],ascending=False) |
|  | actorsTop5=actors.head() |
|  | actorsTop5=actorsTop5.sort\_values(by=['Total Content']) |
|  | fig2=px.bar(actorsTop5,x='Total Content',y='Actor', title='Top 5 Actors on Netflix') |
|  | fig2.show() |



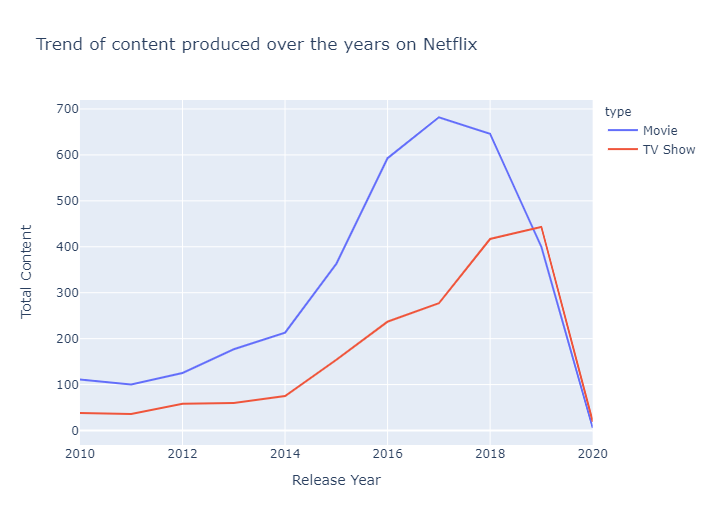
From the above plot, it is derived that the top 5 actors on Netflix are:

1. Anupam Kher
2. Om Puri
3. Shah Rukh Khan
4. Takahira Sakurai
5. Boman Irani

### **Analyzing Content on Netflix:**

The next thing to analyze from this data is the trend of production over the years on Netflix:

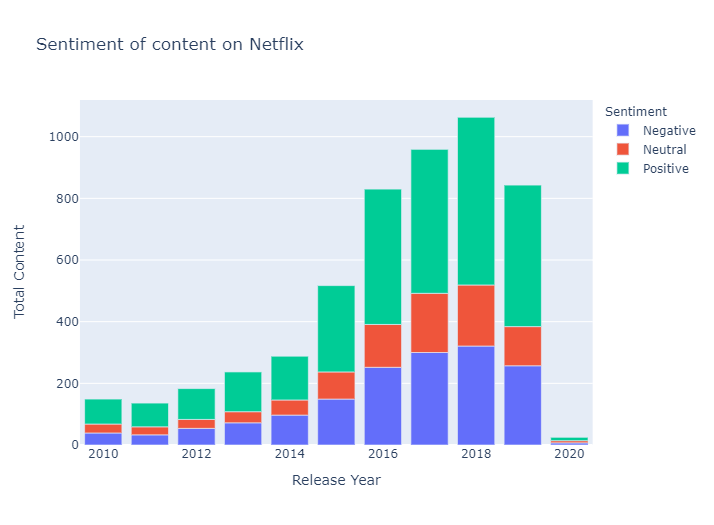
|  |  |
| --- | --- |
|  | df1=dff[['type','release\_year']] |
|  | df1=df1.rename(columns={"release\_year": "Release Year"}) |
|  | df2=df1.groupby(['Release Year','type']).size().reset\_index(name='Total Content') |
|  | df2=df2[df2['Release Year']>=2010] |
|  | fig3 = px.line(df2, x="Release Year", y="Total Content", color='type',title='Trend of content produced over the years on Netflix') |
|  | fig3.show() |



The above line graph shows that there has been a decline in the production of the content for both movies and other shows since 2018.

At last, to conclude our analysis, I will analyze the sentiment of content on Netflix:

|  |  |
| --- | --- |
|  | dfx=dff[['release\_year','description']] |
|  | dfx=dfx.rename(columns={'release\_year':'Release Year'}) |
|  | for index,row in dfx.iterrows(): |
|  | z=row['description'] |
|  | testimonial=TextBlob(z) |
|  | p=testimonial.sentiment.polarity |
|  | if p==0: |
|  | sent='Neutral' |
|  | elif p>0: |
|  | sent='Positive' |
|  | else: |
|  | sent='Negative' |
|  | dfx.loc[[index,2],'Sentiment']=sent |
|  |  |
|  |  |
|  | dfx=dfx.groupby(['Release Year','Sentiment']).size().reset\_index(name='Total Content') |
|  |  |
|  | dfx=dfx[dfx['Release Year']>=2010] |
|  | fig4 = px.bar(dfx, x="Release Year", y="Total Content", color="Sentiment", title="Sentiment of content on Netflix") |
|  | fig4.show() |



So the above graph shows that the overall positive content is always greater than the neutral and negative content combined.