

netflix-data-analysis

November 28, 2023

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

```
[35]: !pip install TextBlob
```

```
Requirement already satisfied: TextBlob in /usr/local/lib/python3.10/dist-
packages (0.17.1)
Requirement already satisfied: nltk>=3.1 in /usr/local/lib/python3.10/dist-
packages (from TextBlob) (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages
(from nltk>=3.1->TextBlob) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages
(from nltk>=3.1->TextBlob) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.10/dist-packages (from nltk>=3.1->TextBlob) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from nltk>=3.1->TextBlob) (4.66.1)
```

```
[40]: import plotly.express as px
from textblob import TextBlob
```

```
[4]: df = pd.read_csv('netflix_titles.csv')
```

```
[5]: df.head()
```

```
[5]:      show_id      type      title \
0  81145628      Movie  Norm of the North: King Sized Adventure
1  80117401      Movie                Jandino: Whatever it Takes
2  70234439  TV Show                Transformers Prime
3  80058654  TV Show      Transformers: Robots in Disguise
4  80125979      Movie                #realityhigh
```

```
      director \
0  Richard Finn, Tim Maltby
1                NaN
2                NaN
```

```

3             NaN
4      Fernando Lebrija

```

```

                                cast \
0  Alan Marriott, Andrew Toth, Brian Dobson, Cole...
1                                Jandino Asporaat
2  Peter Cullen, Sumalee Montano, Frank Welker, J...
3  Will Friedle, Darren Criss, Constance Zimmer, ...
4  Nesta Cooper, Kate Walsh, John Michael Higgins...

```

```

                                country      date_added  release_year \
0  United States, India, South Korea, China  September 9, 2019      2019
1                                United Kingdom  September 9, 2016      2016
2                                United States  September 8, 2018      2013
3                                United States  September 8, 2018      2016
4                                United States  September 8, 2017      2017

```

```

      rating  duration      listed_in \
0      TV-PG    90 min  Children & Family Movies, Comedies
1      TV-MA    94 min                Stand-Up Comedy
2  TV-Y7-FV  1 Season                Kids' TV
3      TV-Y7  1 Season                Kids' TV
4      TV-14   99 min                Comedies

```

```

                                description
0  Before planning an awesome wedding for his gra...
1  Jandino Asporaat riffs on the challenges of ra...
2  With the help of three human allies, the Autob...
3  When a prison ship crash unleashes hundreds of...
4  When nerdy high schooler Dani finally attracts...

```

```
[6]: df.shape
```

```
[6]: (6234, 12)
```

```
[7]: df.columns
```

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

```
[8]: x = df.groupby(['rating']).size().reset_index(name='count')
      print(x)
```

```

      rating  count
0         G     37
1      NC-17      2

```

2	NR	218
3	PG	184
4	PG-13	286
5	R	508
6	TV-14	1698
7	TV-G	149
8	TV-MA	2027
9	TV-PG	701
10	TV-Y	143
11	TV-Y7	169
12	TV-Y7-FV	95
13	UR	7

```
[9]: pieChart = px.pie(x, values='count', names='rating', title = 'Distribution of_
    ↳content rating on the netflix')
    pieChart.show()
```

```
[10]: df['director'] = df['director'].fillna('Director not specified')
```

```
[11]: df.head()
```

```
[11]:
```

	show_id	type	title \
0	81145628	Movie	Norm of the North: King Sized Adventure
1	80117401	Movie	Jandino: Whatever it Takes
2	70234439	TV Show	Transformers Prime
3	80058654	TV Show	Transformers: Robots in Disguise
4	80125979	Movie	#realityhigh

	director \
0	Richard Finn, Tim Maltby
1	Director not specified
2	Director not specified
3	Director not specified
4	Fernando Lebrija

	cast \
0	Alan Marriott, Andrew Toth, Brian Dobson, Cole...
1	Jandino Asporaat
2	Peter Cullen, Sumalee Montano, Frank Welker, J...
3	Will Friedle, Darren Criss, Constance Zimmer, ...
4	Nesta Cooper, Kate Walsh, John Michael Higgins...

	country	date_added	release_year \
0	United States, India, South Korea, China	September 9, 2019	2019
1	United Kingdom	September 9, 2016	2016
2	United States	September 8, 2018	2013
3	United States	September 8, 2018	2016

	rating	duration	listed_in \
0	TV-PG	90 min	Children & Family Movies, Comedies
1	TV-MA	94 min	Stand-Up Comedy
2	TV-Y7-FV	1 Season	Kids' TV
3	TV-Y7	1 Season	Kids' TV
4	TV-14	99 min	Comedies

	description
0	Before planning an awesome wedding for his gra...
1	Jandino Asporaat riffs on the challenges of ra...
2	With the help of three human allies, the Autob...
3	When a prison ship crash unleashes hundreds of...
4	When nerdy high schooler Dani finally attracts...

```
[12]: director_list = pd.DataFrame()
      print(director_list)
```

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

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

```
[ ]: director_list = director_list.to_frame()
      print(director_list)
```

```
[15]: director_list.columns = ['Director']
```

```
[ ]: print(director_list)
```

```
[ ]: directors = director_list.groupby(['Director']).size().reset_index(name = '
      ↳Total Index')
      print(directors)
```

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

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

```
[20]: top5director = directors.head()
```

```
[ ]: top5director
```

```
[22]: top5director = top5director.sort_values(by=['Total Index'])
      barChart = px.bar(top5director,x='Total Index',y = 'Director',title = 'top 5_
      ↪Diretors')
      barChart.show()
```

Analyzing top 5 actors on the netflix

```
[23]: df['cast'].isnull().sum()
```

```
[23]: 570
```

```
[24]: df['cast'] = df['cast'].fillna('No cast specified')
```

```
[25]: cast_df = pd.DataFrame()
      cast_df =df['cast'].str.split(',',expand = True).stack()
      cast_df = cast_df.to_frame()
      cast_df.columns = ['Actor']
```

```
[26]: actors = cast_df.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()
      #top5actors.sort_values(by=['Total count'])
      barchart2 = px.bar(top5actors,x='Total count',y = 'Actor',title = 'Top 5 actor_
      ↪on the netflix')
      barchart2.show()
```

Analyzing on the content product on netflix based on years

```
[27]: df.columns
```

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

```
[28]: df1 = df[['type', 'release_year']]
      df1 = df1.rename(columns={'release_year': 'Release year', 'type': 'Type'})
```

```
[29]: df2 = df1.groupby(['Release year', 'Type']).size().reset_index(name = 'Total_
      ↪count')
```

```
[ ]: print(df2)
```

```
[31]: trend = px.line(df2,x = 'Release year',y = 'Total count',color='Type',title =_
      ↪'Trend of content product on netflix every year')
      trend.show()
```

```
[32]: df2 = df2[df2['Release year']>2000]
trend = px.line(df2,x = 'Release year',y = 'Total count',color='Type',title = 'Trend of content product on netflix every year')
trend.show()
```

Sentimental analysis of netflix content

```
[41]: 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()
```