UE19CS203

Online Streaming Services

Statistics for Data Science

Pihoo Verma (PES2UG19CS282)-E

Priyansh Jain(PES2UG19CS303)-E

Ria Singh(PES2UG19CS326)-E

Rishika Satheesh(PES2UG19CS330)-E

Abstraction

We are using the dataset "Movies on Netflix, Prime Video, Hulu, and Disney+" . This dataset is an amalgamation of a comprehensive list of movies available on various streaming platforms and IMDB dataset. The dataset contains 16,744 values and 17 columns. The major columns being of Age, Language, content variety based on genres, and runtime. We will analyze the genres, ratings, languages, streaming platforms etc for our result.

Introduction

During the unprecedented quarantine we found ourselves at home, with an unexpected vacation from our busy day to day lives. Even with work from home, we were less tired and up to enjoy our spare time. So most of us resorted to binge watching and catching up on movies and shows that we have only been adding to our watchlist since aeons.

Even though the Indian audience has been gradually migrating from the traditional television services to online streaming platforms over the last few years, the online media market saw a boom in streaming services like Netflix, Hulu, Disney+, Hotstar etc. due the pandemic. Thus, baffling the consumers on which subscriptions are better suited to their preferences and which movies they should watch next, the need for recommendations. Thus we aimed to analyze a dataset of the movies and shows present on various streaming platforms.

Dataset

Our dataset was accessed from [Kaggle](), a subsidiary of Google LLC, the largest data science community and is under CC0: Public Domain, thus available for us to use for our learning purposes. We chose this dataset because it has an adequate amount of unique values numbering upto 16.7k and 17 columns, thus allowing us a liberty to venture in different aspects.

Importing Python Libraries

- For Mathematical Functions On The Data
  - import numpy as np
- Access And Manipulate The Data frame
  - import pandas as pd
- Visualization
  - import matplotlib.pyplot as plt
  - import seaborn as sns
  - from plotly.offline import iplot
  - import cufflinks as cf
    - cf.go_offline()
- Data Overview
  - from pandas_profiling import ProfileReport
  - import re
  - import plotly.graph_objects as go
    - fig=go.Figure()

```python
import pandas as pd
import numpy as np

# for visualization
import matplotlib.pyplot as plt
import seaborn as sns
from plotly.offline import iplot
import cufflinks as cf
cf.go_offline()

# for data overview
from pandas_profiling import ProfileReport
import plotly.graph_objects as go
fig = go.Figure()
import re
```

Importing CSV file

pd.read_csv() function from pandas library is used to import

[6]:
```
movie_data = pd.read_csv("/kaggle/input/movies-on-netflix-prime-video-hulu-and-disney/MoviesOnStreamingPlatforms_updated.csv
```

[7]:
```
movie_data.head()
```

Out[7]:

	Unnamed: 0	ID	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	Type	Directors	Genres	Country	Lang
0	0	1	Inception	2010	13+	8.8	87%	1	0	0	0	0	Christopher Nolan	Action,Adventure,Sci-Fi,Thriller	United States,United Kingdom	English,Japanese,Fi
1	1	2	The Matrix	1999	18+	8.7	87%	1	0	0	0	0	Lana Wachowski,Lilly Wachowski	Action,Sci-Fi	United States	Er
2	2	3	Avengers: Infinity War	2018	13+	8.5	84%	1	0	0	0	0	Anthony Russo,Joe Russo	Action,Adventure,Sci-Fi	United States	Er
3	3	4	Back to the Future	1985	7+	8.5	96%	1	0	0	0	0	Robert Zemeckis	Adventure,Comedy,Sci-Fi	United States	Er
4	4	5	The Good, the Bad and the Ugly	1966	18+	8.8	97%	1	0	1	0	0	Sergio Leone	Western	Italy,Spain,West Germany	I

Preprocessing

Dataset Overview: We have 16744 unique values in our dataset and 17 columns. Now looking at the column data information present in out dataset:

```
movie_data.shape
```

Out[16]
```
(16744, 17)
```

[18]:
```
movie_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16744 entries, 0 to 16743
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Unnamed: 0       16744 non-null  int64
 1   ID               16744 non-null  int64
 2   Title            16744 non-null  object
 3   Year             16744 non-null  int64
 4   Age              7354 non-null   object
 5   IMDb             16173 non-null  float64
 6   Rotten Tomatoes  5158 non-null   object
 7   Netflix          16744 non-null  int64
 8   Hulu             16744 non-null  int64
 9   Prime Video      16744 non-null  int64
 10  Disney+          16744 non-null  int64
 11  Type             16744 non-null  int64
 12  Directors        16018 non-null  object
 13  Genres           16469 non-null  object
 14  Country          16309 non-null  object
 15  Language         16145 non-null  object
 16  Runtime          16152 non-null  float64
dtypes: float64(2), int64(8), object(7)
memory usage: 2.2+ MB
```

Data Cleaning: First of all we will drop countries and Directors column as we do not need them for our current case study.

```
df=pd.DataFrame(movie_data)
df.drop(['Directors','Country'], axis=1, inplace= True)
df.head()
```

Out[164]:

	Unnamed: 0	ID	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	Type	Genres	Language	Runtime	Rotten_Tomatoe
0	0	1	Inception	2010	13+	8.8	87%	netflix	0	0	0	0	Action,Adventure,Sci-Fi,Thriller	English,Japanese,French	148.0	
1	1	2	The Matrix	1999	18+	8.7	87%	netflix	0	0	0	0	Action,Sci-Fi	English	136.0	
2	2	3	Avengers: Infinity War	2018	13+	8.5	84%	netflix	0	0	0	0	Action,Adventure,Sci-Fi	English	149.0	
3	3	4	Back to the Future	1985	7+	8.5	96%	netflix	0	0	0	0	Adventure,Comedy,Sci-Fi	English	116.0	
4	4	5	The Good, the Bad and the Ugly	1966	18+	8.8	97%	netflix	0	prime	0	0	Western	Italian	161.0	

Now we will detect and observe the null values present in the dataset and deal with them appropriately.

[12]:

```
movie_data.isnull().sum()
```

Out[12]

```
Unnamed: 0            0
ID                    0
Title                 0
Year                  0
Age                9390
IMDb                571
Rotten Tomatoes   11586
Netflix               0
Hulu                  0
Prime Video           0
Disney+               0
Type                  0
Directors           726
Genres              275
Country             435
Language            599
Runtime             592
dtype: int64
```

+ Code + Markdown

```
Age_Null = round(movie_data['Age'].isnull().sum()/len(movie_data['Age']) * 100 , 2)
Genres_Null = round(movie_data['Genres'].isnull().sum()/len(movie_data['Genres']) * 100 , 2)
Directors_Null = round(movie_data['Directors'].isnull().sum()/len(movie_data['Directors']) * 100 , 2)
Runtime_Null = round(movie_data['Runtime'].isnull().sum()/len(movie_data['Runtime']) * 100 , 2)
Language_Null = round(movie_data['Language'].isnull().sum()/len(movie_data['Language']) * 100 , 2)
Country_Null = round(movie_data['Country'].isnull().sum()/len(movie_data['Country']) * 100 , 2)
Rotten_Null = round(movie_data['Rotten Tomatoes'].isnull().sum()/len(movie_data['Rotten Tomatoes']) * 100 , 2)
```

+ Code + Markdown

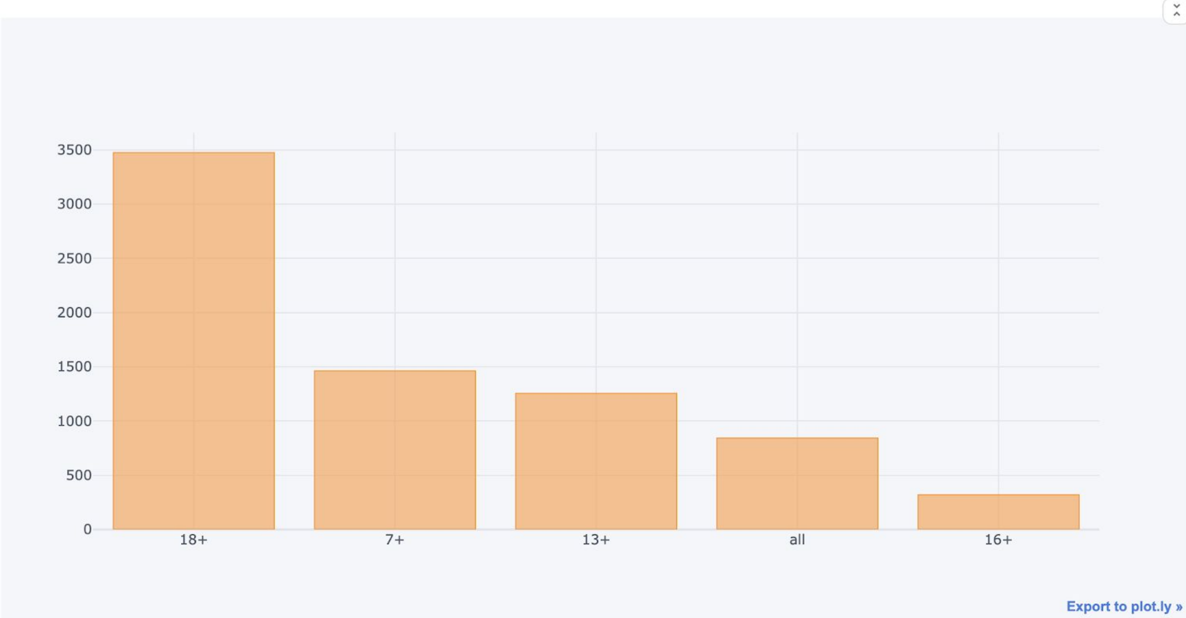
[123]:

```
print("Age Null: {}%".format(Age_Null))
print("Genres_Null: {}%".format(Genres_Null))
print("Directors_Null: {}%".format(Directors_Null))
print("Runtime_Null : {}%".format(Runtime_Null))
print("Language_Null: {}%".format(Language_Null))
print("Country_Null: {}%".format(Country_Null))
print("Rotten_Null: {}%".format(Rotten_Null))
```

```
Age Null: 56.08%
Genres_Null: 1.64%
Directors_Null: 4.34%
Runtime_Null : 3.54%
Language_Null: 3.58%
Country_Null: 2.6%
Rotten_Null: 69.19%
```

A visual representation of the motion picture content rating system.

```
movie_data['Age'].value_counts().iplot('bar')
```



These are the following values which have a missing age rating.

[24]:
```
movie_data[movie_data['Age'].isnull()].shape
```

Out[24]:
```
(9390, 17)
```

```
movie_data[movie_data['Age'].isnull()]
```

Out[25]:

	Unnamed: 0	ID	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	Type	Directors	Genres	Country	Language	Runtime
32	32	33	Train to Busan	2016	NaN	7.5	94%	1	0	1	0	0	Sang-ho Yeon	Action,Horror,Thriller	South Korea	Korean,Hawaiian	118.0
57	57	58	A Silent Voice	2016	NaN	8.2	94%	1	0	0	0	0	Naoko Yamada	Animation,Drama,Family,Romance	Japan	Japanese,Japanese Sign Language	130.0
89	89	90	The Dawn Wall	2018	NaN	8.1	100%	1	0	0	0	0	Josh Lowell,Peter Mortimer	Documentary,Biography,Sport	Austria,United States	English	100.0
110	110	111	Black Mirror: Bandersnatch	2018	NaN	7.2	92%	1	0	0	0	0	David Slade	Drama,Mystery,Sci-Fi,Thriller	United States,United Kingdom	English	90.0
123	123	124	Neon Genesis Evangelion: The End of Evangelion	1997	NaN	8.1	88%	1	0	0	0	0	Hideaki Anno,Kazuya Tsurumaki	Animation,Action,Drama,Fantasy,Sci-Fi	Japan	Japanese	87.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
16736	16736	16737	Winged Seduction: Birds of Paradise	2012	NaN	6.5	NaN	0	0	0	1	0	Molly Hermann	Documentary	United States	English	NaN
16739	16739	16740	The Ghosts of Buxley Hall	1980	NaN	6.2	NaN	0	0	0	1	0	Bruce Bilson	Comedy,Family,Fantasy,Horror	United States	English	120.0
16741	16741	16742	Sharks of Lost Island	2013	NaN	5.7	NaN	0	0	0	1	0	Neil Gelinas	Documentary	United States	English	NaN
16742	16742	16743	Man Among Cheetahs	2017	NaN	6.6	NaN	0	0	0	1	0	Richard Slater-Jones	Documentary	United States	English	NaN
16743	16743	16744	In Beaver Valley	1950	NaN	NaN	NaN	0	0	0	1	0	James Algar	Documentary,Short,Family	United States	English	32.0

9390 rows × 17 columns

Exploratory Data Analysis

In EDA, we will analyze the datasets to summarize their main characteristics of the values.

Rotten Tomatoes Ratings of movies present

Overviewing the highest rotten tomato rating movies on the platforms included in our dataset. The Tomatometer score represents the percentage of professional critic reviews that are positive for a given film or television show.

```
movie_data['Rotten Tomatoes'].value_counts()
```

Out[16]:
```
100%    407
80%     162
50%     136
83%     131
67%     126
       ...
7%       10
5%       10
4%        9
3%        4
2%        4
Name: Rotten Tomatoes, Length: 99, dtype: int64
```

For better graphical visualization, we will round up the values:

[18]:

```
#rounding off for histogram representation
import re

def convert_str_to_int(val):
    new_val =  re.sub('%','',val)
    return(int(new_val))

def round_fix(data):
    data_str = str(data).strip()
    if data_str != 'nan':
        data = convert_str_to_int(data_str)
        if data in range(0,11):
#             print(data)
            return '10'
        if data in range(11,21):
            return '20'
        if data in range(21,31):
            return '30'
        if data in range(31,41):
            return '40'
        if data in range(41,51):
            return '50'
        if data in range(51,61):
            return '60'
        if data in range(61,71):
            return '70'
        if data in range(71,81):
            return '80'
        if data in range(81,91):
            return '90'

        if data in range(91,101):
            return '100'


movie_data['Rotten_Tomatoes_overview'] = movie_data['Rotten Tomatoes'].apply(round_fix)
```

+ Code + Markdown

```
movie_data['Rotten_Tomatoes_overview'].value_counts().iplot(kind='bar', bins=20, xTitle = 'Rotten Tomatoes Ratings', yTitle='Number of movies',
```

Observation: In Rotten Tomatoes, the highest possible rating is 100%. Also most of the movies present on online streaming services lie in the upper grade and have positive reviews, thus meaning the content of these platforms is adept.

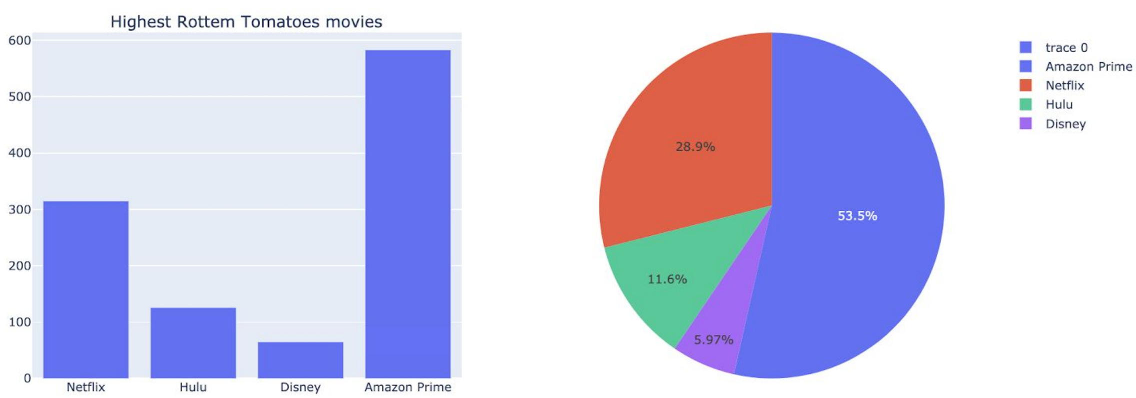
Top Rotten Tomatoes Movies on Online Platforms

Let us compare which platform holds a better compilation of movies, i.e. on which platform the frequency of high Rotten Tomatoes Rating movie is greater.

```python
netflix_count = movie_data[movie_data['Rotten_Tomatoes_overview'] == '100']['Netflix'].sum()
Hulu_count = movie_data[movie_data['Rotten_Tomatoes_overview'] == '100']['Hulu'].sum()
Disney_count = movie_data[movie_data['Rotten_Tomatoes_overview'] == '100']['Disney+'].sum()
prime_count = movie_data[movie_data['Rotten_Tomatoes_overview'] == '100']['Prime Video'].sum()

indexes = ['Netflix', 'Hulu', 'Disney', 'Amazon Prime']
values = [netflix_count, Hulu_count, Disney_count,prime_count]
for x,y in zip(indexes,values):
    print(x,y)
```

```
Netflix 315
Hulu 126
Disney 65
Amazon Prime 583
```

```python
from plotly.subplots import make_subplots

fig = make_subplots(
rows=1,cols=2, subplot_titles=["Highest Rottem Tomatoes movies"],
specs=[[{'type':'bar'},{'type':'pie'}]])

fig.add_trace(go.Bar(x=indexes, y=values), row=1,col=1)
fig.add_trace(go.Pie(labels=indexes, values=values), row=1,col=2)
```



Observation: Amazon Prime holds more high rating movies, followed by Netflix, Hulu and Disney in mentioned order only.

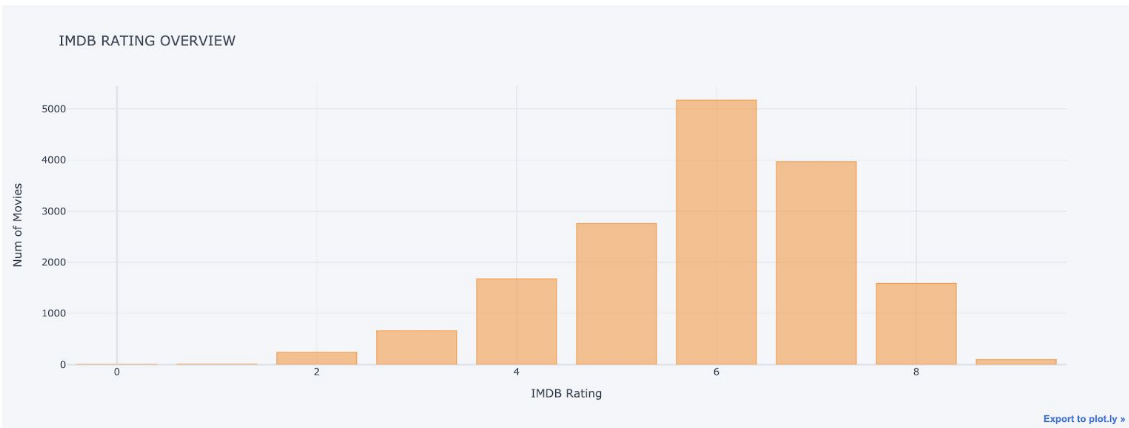
IMDB Ratings

Besides Rotten Tomatoes, IMDB ratings are also widely considered by the critics and the consumers.

[97]:
```
#rounding off
def round_val(data):
    if str(data) != 'nan':
        return round(data)

movie_data['IMDB_group'] = movie_data['IMDb'].apply(round_val)
values = movie_data['IMDB_group'].value_counts().sort_index(ascending=False).tolist()
index = movie_data['IMDB_group'].value_counts().sort_index(ascending=False).index
for x,y in zip(index,values):
    print(x,y)
```

```
9.0 97
8.0 1590
7.0 3965
6.0 5173
5.0 2758
4.0 1678
3.0 660
2.0 241
1.0 7
0.0 4
```

[134]:
```
movie_data['IMDB_group'].value_counts().iplot('bar', xTitle='IMDB Rating', yTitle='Num of Movies', title='IMDB RATING OVERVIEW')
```



Observation: IMDB is stricter in its ratings, the amount of absolute 10 is very low because it is reserved for exceptional cinematic masterpieces, also in IMDB the good movies lie above the ratings of 6.5 and 7.

Top IMDB Movies on Online Platforms

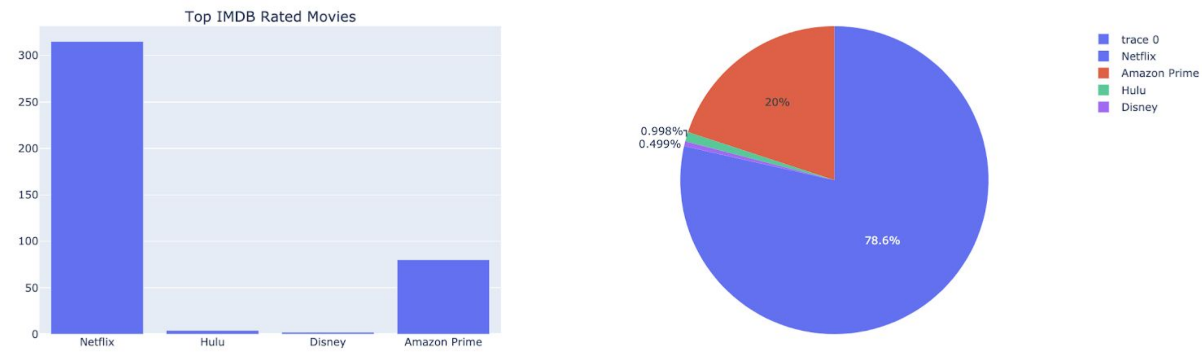
Similar to Rotten Tomatoes, let us compare which platform holds a better compilation of movies, i.e. which platform has more high IMDB rating movies.

[290]:
```
Netflix_count = movie_data[movie_data['IMDB_group'] == 9]['Netflix'].sum()
Hulu_count = movie_data[movie_data['IMDB_group'] == 9]['Hulu'].sum()
Disney_count = movie_data[movie_data['IMDB_group'] == 9]['Disney+'].sum()
prime_count = movie_data[movie_data['IMDB_group'] == 9]['Prime Video'].sum()

indexes = ['Netflix', 'Hulu', 'Disney', 'Amazon Prime']
values = [netflix_count, Hulu_count, Disney_count,prime_count]
for x,y in zip(indexes,values):
    print(x,y)
```

```
Netflix 315
Hulu 4
Disney 2
Amazon Prime 80
```

[342]:

```
fig = make_subplots(
rows=1,cols=2, subplot_titles=["Top IMDB Rated Movies"],
specs=[[{'type':'bar'},{'type':'pie'}]])

fig.add_trace(go.Bar(x=indexes, y=values), row=1,col=1)
fig.add_trace(go.Pie(labels=indexes, values=values), row=1,col=2)
```



Observation: We can see that Netflix has the highest number of top IMDB movies, followed by Amazon Prime Video.

Recommended Movies

Taking the IMDB ratings in account, these are the top recommendations

[163]:

```
movie_data
```

Out[163]:

Jnnamed: 0	ID	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	Type	Directors	Genres	Country	Language	Runtime	Rotten_
0	1	Inception	2010	13+	8.8	87%	netflix	0	0	0	0	Christopher Nolan	Action,Adventure,Sci-Fi,Thriller	United States,United Kingdom	English,Japanese,French	148.0	
1	2	The Matrix	1999	18+	8.7	87%	netflix	0	0	0	0	Lana Wachowski,Lilly Wachowski	Action,Sci-Fi	United States	English	136.0	
2	3	Avengers: Infinity War	2018	13+	8.5	84%	netflix	0	0	0	0	Anthony Russo,Joe Russo	Action,Adventure,Sci-Fi	United States	English	149.0	
3	4	Back to the Future	1985	7+	8.5	96%	netflix	0	0	0	0	Robert Zemeckis	Adventure,Comedy,Sci-Fi	United States	English	116.0	
4	5	The Good, the Bad and the Ugly	1966	18+	8.8	97%	netflix	0	prime	0	0	Sergio Leone	Western	Italy,Spain,West Germany	Italian	161.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
16739	16740	The Ghosts of Buxley Hall	1980	NaN	6.2	NaN	0	0	0	disney	0	Bruce Bilson	Comedy,Family,Fantasy,Horror	United States	English	120.0	
16740	16741	The Poof Point	2001	7+	4.7	NaN	0	0	0	disney	0	Neal Israel	Comedy,Family,Sci-Fi	United States	English	90.0	
16741	16742	Sharks of Lost Island	2013	NaN	5.7	NaN	0	0	0	disney	0	Neil Gelinas	Documentary	United States	English	NaN	
16742	16743	Man Among Cheetahs	2017	NaN	6.6	NaN	0	0	0	disney	0	Richard Slater-Jones	Documentary	United States	English	NaN	
16743	16744	In Beaver Valley	1950	NaN	NaN	NaN	0	0	0	disney	0	James Algar	Documentary,Short,Family	United States	English	32.0	

ws × 19 columns

Movie Genre Frequency

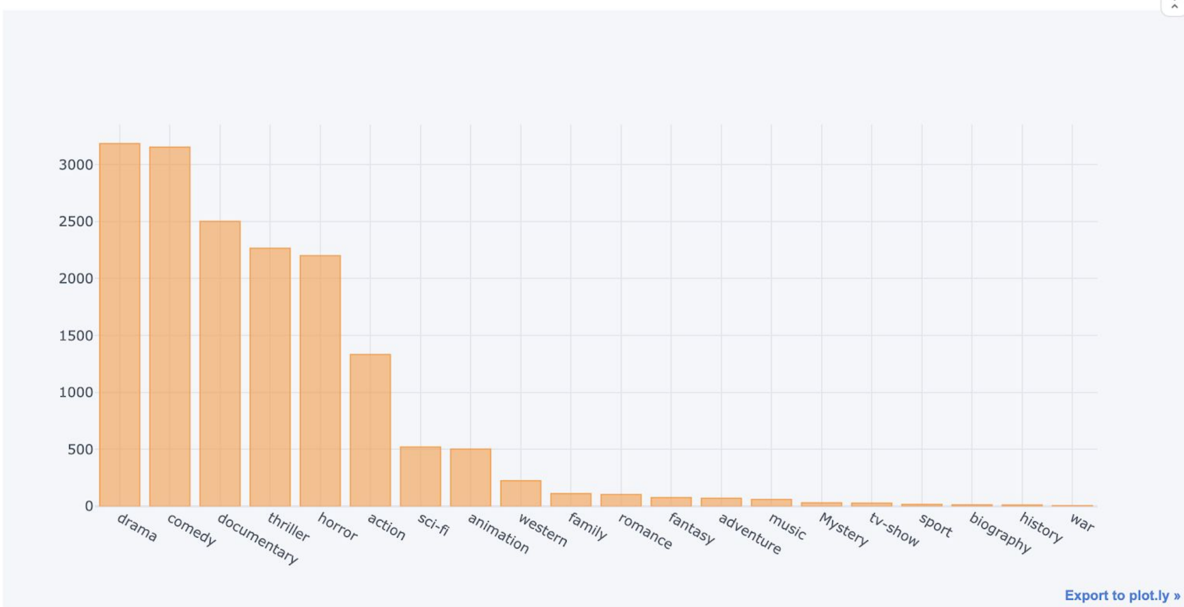
Calculating by priority rule, we will assign the major genre of the film and then calculate the frequency of subsequent genres.

[165]:
```
def check_thriller(data):
#     printdata)
    if str(data).strip() != 'nan':
#         print(data)
        if 'horror' in data.lower():
            return 'horror'
        elif 'thriller' in data.lower():
            return 'thriller'
        elif 'sci-fi' in data.lower():
            return 'sci-fi'
        elif 'documentary' in data.lower():
            return 'documentary'
        elif 'action' in data.lower():
            return 'action'
        elif 'animation' in data.lower():
            return 'animation'
        elif 'comedy' in data.lower():
            return 'comedy'
        elif 'western' in data.lower():
            return  'western'
        elif 'drama' in data.lower():
            return 'drama'
        elif 'fantasy' in data.lower():
            return 'fantasy'
        elif 'romance' in data.lower():
            return 'romance'
        elif 'music' in data.lower():
            return 'music'
        elif 'adventure' in data.lower():
            return 'adventure'
        elif 'sport' in data.lower():
            return 'sport'
        elif 'reality-tv' in data.lower() or 'talk-show' in data.lower() or 'game-show' in data.lower():
            return 'tv-show'
        elif 'history' in data.lower():
            return 'history'
        elif 'family' in data.lower():
            return 'family'
        elif 'biography' in data.lower():
            return 'biography'
        elif 'biography' in data.lower():
            return 'biography'
        elif 'mystery' in data.lower():
            return 'Mystery'
        elif 'war' in data.lower():
            return 'war'

movie_data['mov_type'] = movie_data['Genres'].apply(check_thriller)
```

+ Code   + Markdown

Plotting the data graphically, for better visual representation.

```
movie_data['mov_type'].value_counts().iplot('bar')
```



Top Movies and online platforms

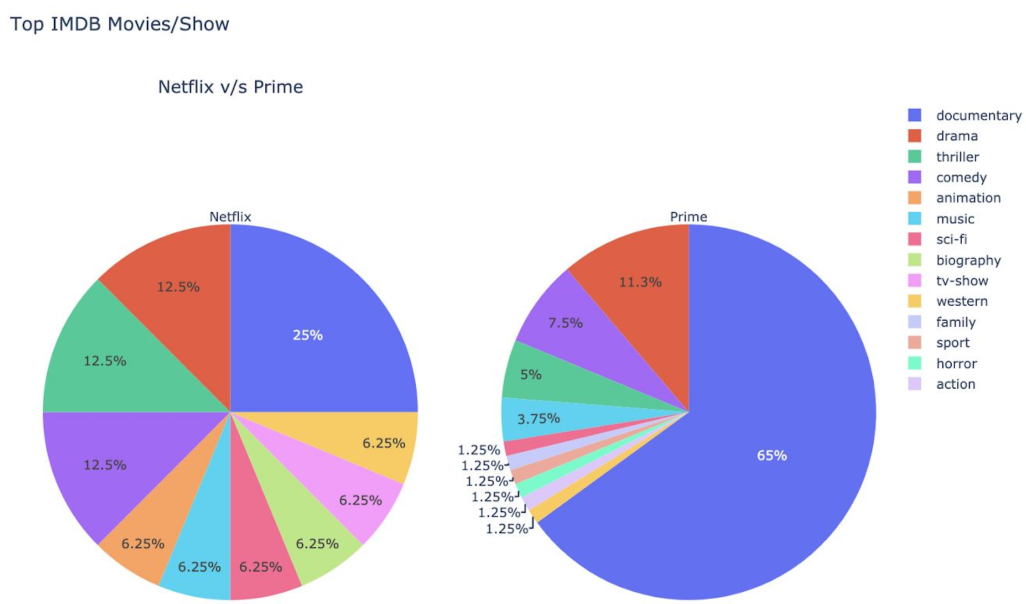
Let us calculate genre frequency on different platforms (here indexes are for the genres and the values are for the frequency).

```
[1066]: top_movie = movie_data[movie_data['IMDB_group'] == 9]
```

```
[1067]: net_index = top_movie[top_movie['Netflix']=='netflix']['mov_type'].value_counts().index.tolist()
net_val = (top_movie[top_movie['Netflix']=='netflix']['mov_type'].value_counts().values.tolist())

prime_index = top_movie[top_movie['Prime Video']=='prime']['mov_type'].value_counts().index.tolist()
prime_val = (top_movie[top_movie['Prime Video']=='prime']['mov_type'].value_counts().values.tolist())

disney_index = top_movie[top_movie['Disney+']=='disney']['mov_type'].value_counts().index.tolist()
disney_val = (top_movie[top_movie['Disney+']=='disney']['mov_type'].value_counts().values.tolist())

hulu_index = top_movie[top_movie['Hulu']=='hulu']['mov_type'].value_counts().index.tolist()
hulu_val = (top_movie[top_movie['Hulu']=='hulu']['mov_type'].value_counts().values.tolist())
```

Comparing two services, Netflix and Amazon Prime we observe that Netflix holds a wider variety of cinematic palette as compared to Amazon Prime Video where documentaries seem to heavily dominate, followed by drama.

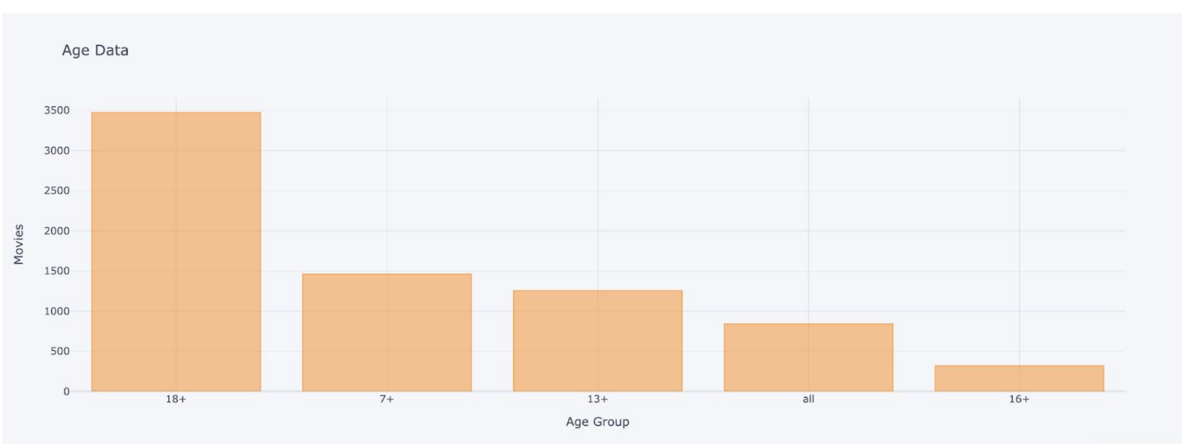
```
fig = make_subplots(
rows=1,cols=2, subplot_titles=["Netflix v/s Prime"],
specs=[[{'type':'pie'},{'type':'pie'}]])

fig.add_trace(go.Pie(labels=net_index, values=net_val, title='Netflix'), row=1,col=1)
fig.add_trace(go.Pie(labels=prime_index, values=prime_val, title='Prime'), row=1,col=2)
fig.update_layout(height=800, width=1000, title_text='Top IMDB Movies/Show')
```



Age appropriate content availability

Here we will observe the supply of the various age appropriate content on these platforms.

```
[276]: movie_data['Age'].value_counts().iplot('bar', xTitle='Age Group', yTitle='Movies',title='Age Data')
```



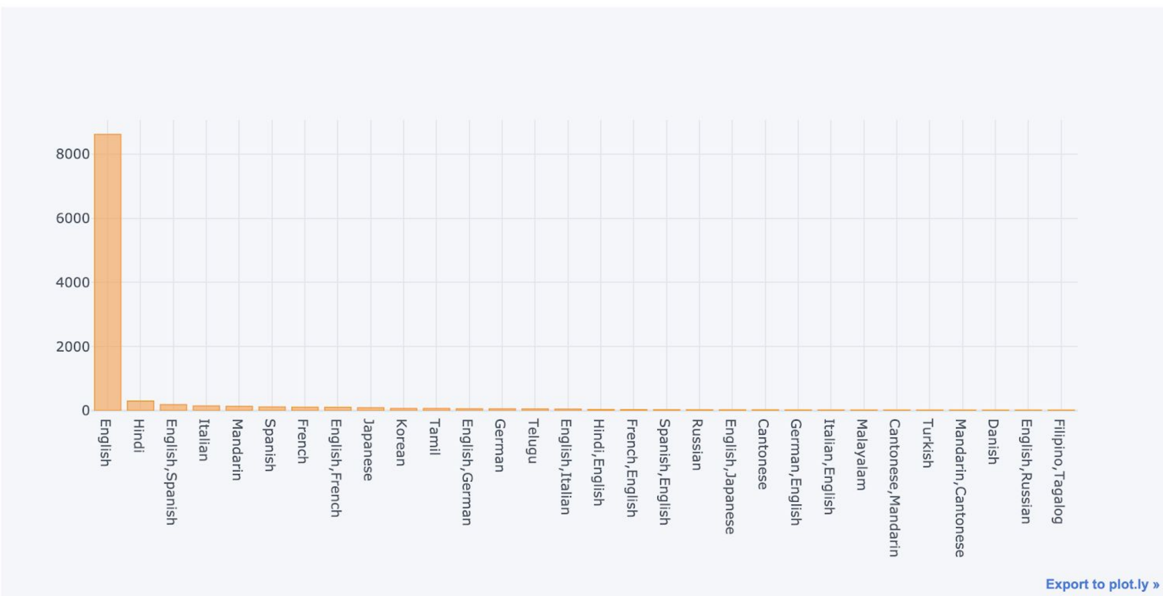
Observation: Most of the content is for 18 yr + users, while there does appear to be content for young kids and toddlers.

Amazon Prime Video Overview

For case study, let us take one of the services and inspect various aspects.

1. Language: From the graphical representation Amazon Prime Video has a vast collection of English Language and very little of others. This is mostly due to the fact that English is a global language and has the most speakers so most motion pictures have been dubbed into English, to accommodate the masses.

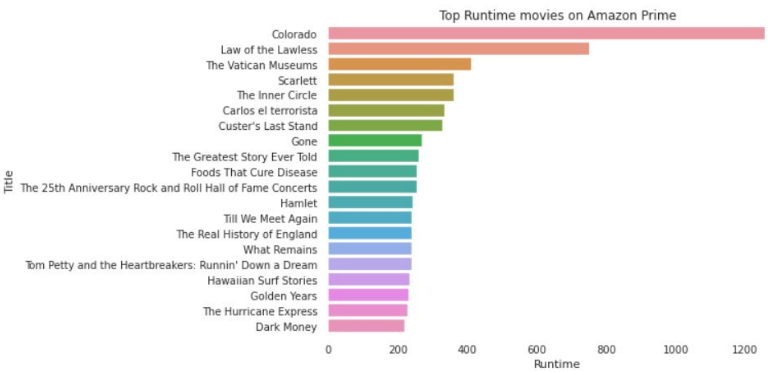
```
[1072]: prime_movies['Language'].value_counts()[:30].iplot('bar')
```



2. Runtime: We can observe that “Colorado” seems to have the highest runtime, followed by “Law of the Lawless”.

```
[120]:
# Runtime
runtime_net = prime_movies.sort_values(by='Runtime',ascending=False).head(20)
runtime_net

sns.barplot(data=runtime_net, y='Title',x='Runtime')
plt.title('Top Runtime movies on Amazon Prime')
```

```
Out[120]: Text(0.5, 1.0, 'Top Runtime movies on Amazon Prime')
```



3. Movies with 8+ IMDB rating: There are 324 movies on Amazon Prime with IMDB 8+ rating.

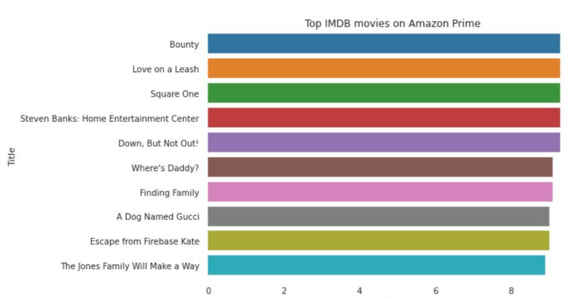
```
[121]:
net_movies = prime_movies[prime_movies['IMDb'] > 8]
net_movies_count = net_movies.shape[0]
print("Movies with IMDB 8+ Rating in Amazon Prime: {}".format(net_movies_count))
```

```
Movies with IMDB 8+ Rating in Amazon Prime: 324
```

4. Top IMDB movies: We will sort the movies present in the Amazon Prime Video collection based on their IMDB ratings. “Bounty” comes in first, succeeded by “Love on a Leash” with a very minute difference.

```
[122]:
net_movies = net_movies.sort_values(by='IMDb', ascending=False).head(10)
sns.barplot(data=net_movies, y='Title',x='IMDb')
plt.title('Top IMDB movies on Amazon Prime')
```

```
Out[122]: Text(0.5, 1.0, 'Top IMDB movies on Amazon Prime')
```



5. Genres with IMDB ratings: The top 10 genres with the highest IMDB ratings are as follow-

```
movie_type = prime_movies['mov_type'].value_counts().index.tolist()
print(movie_type[:10])
```

```
['drama', 'comedy', 'horror', 'documentary', 'thriller', 'action', 'sci-fi', 'animation', 'western', 'romance']
```

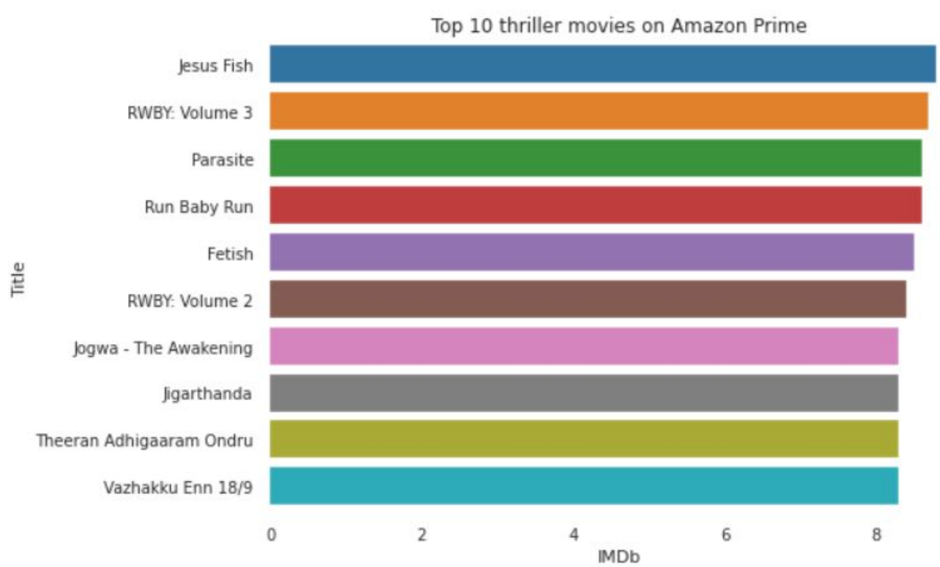
Thriller Movies: Jesus Fish, RWY: Volume 3, Parasite, and Fetish are a few among the top thriller movies on Amazon Prime with a good IMDB rating.

```
def movie_plot(movie) :
    print(movie)
    thriller_net = prime_movies[prime_movies['mov_type'] == movie]
    net_movies = thriller_net.sort_values(by='IMDb', ascending=False).head(10)
    sns.barplot(data=net_movies, y='Title',x='IMDb')
    plt.title('Top 10 {} movies on Amazon Prime'.format(movie))
```

[125]:

```
movie_plot('thriller')
```

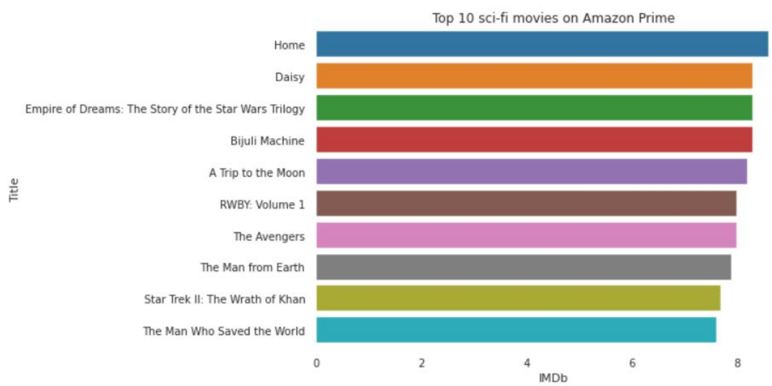
```
thriller
```



Sci-fi Movies: Home and Daisy are the top 2 sci-fi movies with good IMDB ratings.

```
movie_plot('sci-fi')
```

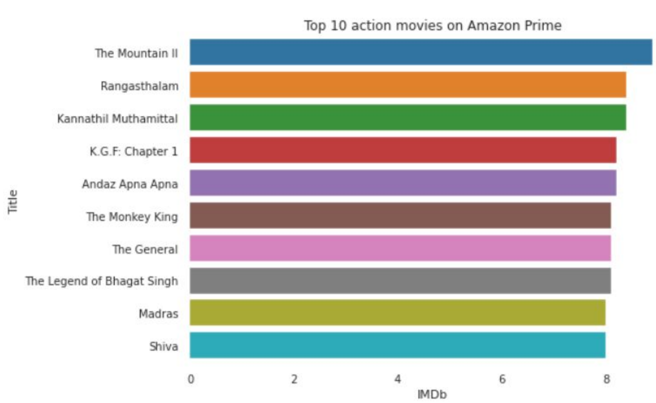
```
sci-fi
```



Action Movies: The Mountain, and Rangasthan are the top 2 action movies with good IMDB ratings.

```
movie_plot('action')
```

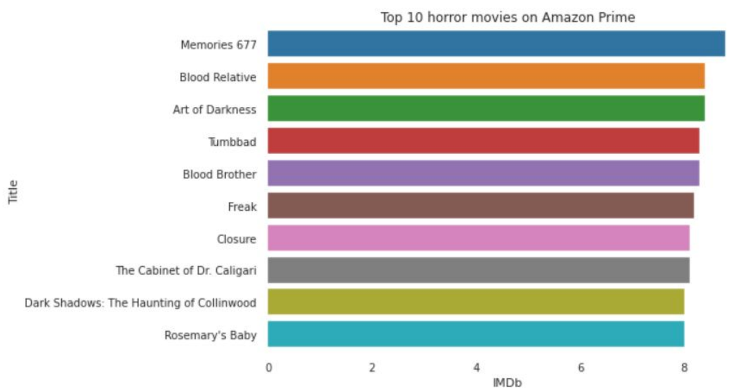
```
action
```



Horror Movies: Memories 667, Blood Relative and Art of Darkness are most recommended horror movies, with good IMDB ratings, to watch on Amazon Prime.

```
movie_plot('horror')
```

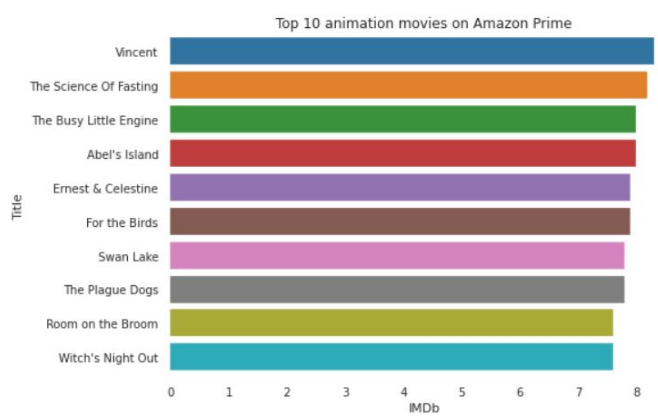
```
horror
```



Animation: Vincent, The Science of Fasting and The Busy Little Engine are top 3 animated movies on Amazon Prime.

```
movie_plot('animation')
```

```
animation
```



Hypothesis Testing

Using Inferential, Descriptive, and Exploratory analysis, we performed some research on the population sample.

[11]:
```
points=movie_data['IMDb']
mu=points.mean()
sigma=points.std(ddof=0)
print("mu: ", mu, ", sigma:", sigma)
```

```
mu:  5.902751499412594 , sigma: 1.3478257911683673
```

H0: The null hypothesis, denoted by ***H*0**, is usually the hypothesis that sample observations result purely from chance. The sample is from the MoviesOnStreamingPlatforms, x_bar = μ.

HA: The alternative hypothesis, denoted by ***H*1 *or Ha***, is the hypothesis that's ample observations are influenced by some non-random cause. The sample is not from the MoviesOnStreamingPlatforms, x_bar != (not equal) μ.

```
z_critical = 1.96 # alpha level of 0.05 and two-tailed test
x_bar = 8.66
N = 5
SE = sigma/np.sqrt(N)
z_stat = (x_bar - mu)/SE
print(z_stat)
```

```
4.57432638444195
```

Since z_stat is greater than z_critical we reject the null hypothesis and accept the alternative. Statistically, we say the sample mean is different from the population mean and thus the sample is drawn not from the population.

But what if the sample size was larger? Let's redo the calculation with N=300.

```
[13]:
N = 300;
SE = sigma/np.sqrt(N)
z_stat = (x_bar - mu)/SE
print(z_stat)
```

```
35.43257981412172
```

Here too we get a value of z_stat which is greater than z_critical. **Hence we accept Alternate Hypothesis.**

Correlation

Data Correlation is a way to understand the relationship between multiple variables and attributes in your dataset. Using Correlation, you can get some insights such as: One or multiple attributes depend on another attribute or a cause for another attribute. An effect score closer to 0 translates to there being no relationship.A score closer to 1 and -1 results in a positive and negative correlation respectively.

```
[120]:
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.pyplot import pie,axis,show
%matplotlib inline
```

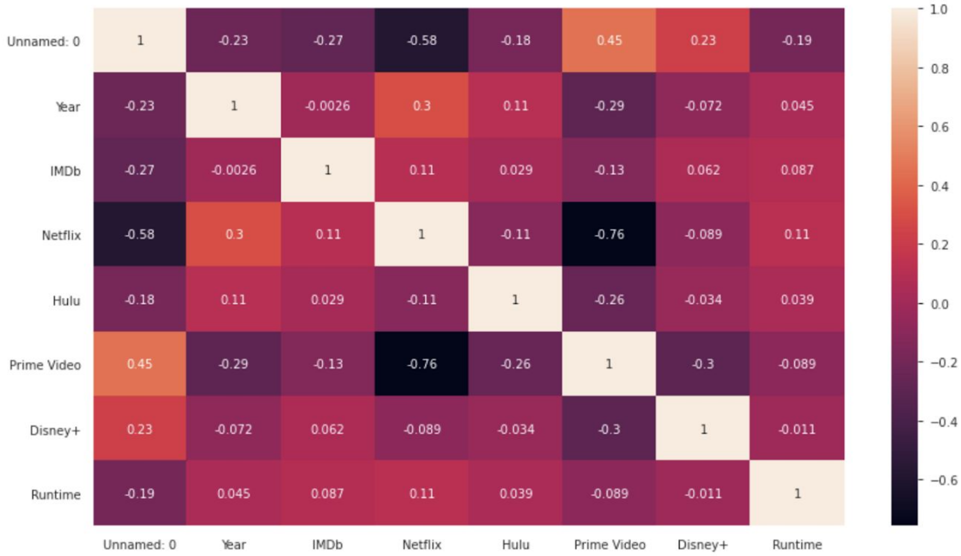
```
movie_data = pd.read_csv("/kaggle/input/movies-on-netflix-prime-video-hulu-and-disney/MoviesOnStreamingPlatforms_updated.csv")
df_corr = movie_data.corr()
fields = ['Type', 'ID']
# drop rows
df_corr.drop(fields, inplace=True)
# drop cols
df_corr.drop(fields, axis=1, inplace=True)
```

```
[128]:
corr = movie_data.corr(method='kendall')
plt.figure(figsize=(14,8))
sns.heatmap(corr,annot=True)
plt.savefig("correlation.png")
movie_data.columns
```

Heatmap of the dataset showing relation between the relevant columns.

Out[128

```
Index(['Unnamed: 0', 'Title', 'Year', 'Age', 'IMDb', 'Rotten Tomatoes',
       'Netflix', 'Hulu', 'Prime Video', 'Disney+', 'Directors', 'Genres',
       'Country', 'Language', 'Runtime'],
      dtype='object')
```



+ Code + Markdown

Result and Discussion

By all our observations we can conclude the following:

1. If you prefer critic reviews and Documentary series, Amazon Prime Video holds a better collection.
2. While if you are a connoisseur of cinema and prefer a variety in motion pictures, you should definitely check out Netflix.
3. Netflix's collection is mostly curated to user preferred movies rather than professional critics.
4. English Movies have the biggest set, as it is an internationally spoken language.
5. Most Online streaming services have 18+ motion pictures content rating, so it is advisable for the consumers to adhere to it.