A

Project Report on

**Present and Future Trends of Netflix OTT Platform**

Submitted in partial fulfillment of completion of the course

Advanced Diploma in IT, Networking and Cloud

Submitted by:

**Himanshu**

**Geetanjali Raheja**

**Neeraj Kumar**

**Dewesh**

Under Guidance of:

**Ms. Jiljil Mathew (IBM/Kyndryl)**

**Mr. Deepneel Majumdar (Edunet)**

|  |  |  |
| --- | --- | --- |
| IBM-Logo - Chicago Innovation | DGT introduces high end diploma courses - digitalLEARNING Magazine | Edunet Foundation-Delhi- CSR Organization profile |

Year 2022

Abstract

Acknowledgement

Team Composition and Workload Division

Table of Contents

1. Introduction to Problem
2. Literature Review
3. Proposed Solution
4. Requirements

4.1 Technology Stack

4.2 Hardware

4.3 Software

4.4 Deployment Environment

5. User Requirements

6. Design Documentation

7. Implementation Details

8. Testing

9. Deployment

10. Future Scope

11. Conclusion

Appendix A Project Code

Appendix B Screenshot of Project

References

**Abstract**

Netflix is a platform that provides their users a way to stream movies without the need of renting movies from the store and some of the newly released movies are available there. Netflix is considered to be one of the best in providing online streaming as they contain a vast amount of films and television series, including those that are produced by themselves. It's subscribers keep on increasing as the year progresses. At the end of 2019, Netflix had 167 millions subscribers and by the first quarter of 2020, that number increases up to 182 millions subscribers. As a result from Covid-19, most prefer to avoid public places such as cinema thus increasing the amount of subscribers for Netflix mainly as they are considered one of the main contenders in online streaming companies. Directors and movie's producer are attracted to secure an agreement with Netflix to showcase their movies or television series on the platform. This yields advantages for both parties as both of them are able to enjoy the benefit gained from it.

**Acknowledgement**

We are really grateful for this project opportunity and would sincerely thank Jiljil Mam (IBM Mentor) and Deepneel sir (Edunet Mentor) for trusting me and my team mates with this project. They are proved to be a guiding light in my entire journey and their valuable insight has helped me and my team mates improve and make this project a success.

Besides, we would like to thank Shreya & Radhika ma’am who helped us by giving us valuable guidance and suggestions helped in various phases of the completion of this project. We will always be thankful to you in this regard.

Also, I would thank my project team mates who supported me academically as well as emotionally to complete this project without any obstacles.

**Team Composition and Workload Division**

**DATA ANALYTICS TEAM 07**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name** | **Workload** |
| **1** | Himanshu | Data Visualization in Jupyter Notebook, Dashboard In Microsoft Power BI |
| **2** | Geetanjali Raheja | Data Preparation in Jupyter Notebook, Dashboard In Microsoft Power BI |
| **3** | Neeraj Kumar | Data Collection, Documentation, Deployment |
| **4** | Dewesh | Documentation |

**Table of Contents**

1. **Introduction to Problem**

There are an abundance of movies being released nowadays from big companies and even from individuals. This causes competition to be developed as viewers have many choices on which movies or TV shows they want to watch. Directors had to challenge themselves in making a movie genre that is suitable for everyone and also making it a sensation. Most directors are unsure on what type of movies that can attract a large audience as there are no available platforms for them to look at. Without a proper insight, they may face a financial loss if the movie receives little to none viewers.

1. **Literature Review**

With the advent of technology, the need for consumption of data has increased tremendously. Every single activity of our life has become interlinked with data. As the famous British Mathematician – Clive Humby – once said, “Data is the new oil.”

As interesting as it may sound, the fact that there is a constant need for intelligent solutions – with data as inputs – cannot be neglected. Due to these changes, we have witnessed a paradigm shift in almost every single field around the world. One such field is the entertainment industry that had a rapid transposition as the industry soon began to adopt virtual methods for releasing its content. The Over-The-Top (OTT) as a means to provide and deliver content has gained huge prominence around the world. One of the key players in the game being Netflix, has changed the way people subject themselves to entertainment. The from-home-all-in-one package type subscription services that these OTT platforms has to offer had customers glued to them quite literally. All these behavioral information about large base of customers is essentially a valuable data. Thus, our project focuses on deriving crucial insights from the publicly available Netflix dataset (obtained from Kaggle). We have also made use of Geographical Data Set and Netflix Title Reviews Data Set.

1. **Proposed Solution**

The proposed solution is we are giving a best analysis on the acquired data, so Netfilx can understand which type of Movies and Shows are liked by people. This analysis will make it easy to understand where to invest more money and time for increasing their streaming.

1. **Requirements**
   1. **Technology Stack**
2. Python v3.9.4

* NumPy Libraries
* Pandas Libraries
* Matplotlib Libraries
  1. **Hardware**

1. 1 GB Ram or above
2. 2 GB Hard Disk or above
3. Intel i3 3rd generation or above
4. Processor bit size(32 bit/64 bit or above)
   1. **Software**
5. Operating System(Windows/Linux/Macbook/Unix/Kaios)
6. Jupyter Notebook
7. Microsoft Power BI
   1. **Deployment Environment**
8. Microsoft Power BI
9. **User Requirements**
   1. Mobile (Android or IOS) or System (Windows or MacOS or Linux)
   2. Browser (Chrome, Mozilla Firefox and Microsoft Edge etc.)
10. **Design Documentation**
11. **Data Collection**

As our problem statement is based on Netflix OTT platform analysis, we found the data from Kaggle website. The dataset contains 15321 rows of records and 21 columns of attributes. It contains two types of attributes which are string and integer. It refers to the movie's title along with the director of the movie, the time published, year published, the rating of the movie, type of show, description of the movie and how long it is running for. The oldest movie available in this dataset is from the year 2015 and the latest is from 2021.

**Data Description:-**

|  |  |  |
| --- | --- | --- |
| **No** | **Attributes** | **Description** |
| 1 | show\_id | As id that differentiate each type of movie and tv show available |
| 2 | Title | The title of the movie or tv show |
| 3 | Genre | Category will be based on where the majority of the movie or tv show. |
| 4 | Languages | The language that the movie in ….. |
| 5 | Series or Movie | The type of the show whether it is movie or tv show |
| 6 | Country Availability | In which countries available movie or tv show |
| 7 | Runtime | The total runtime for each movie or tv show |
| 8 | Director | The individual that is responsible for making the movie or tv show and supervise the actors and crews |
| 9 | Writer | A film writer is a person who writes scripts for movie or tv show. |
| 10 | Actors | Who performed in that movie or tv show |
| 11 | View Rating | The rating of movie or tv show based on age- appropriateness |
| 12 | IMDb Votes | IMDb registered users can vote for movie or tv show |
| 13 | IMDb Score | IMDB score that allows users to rate movie or tv show on a scale of one to ten |
| 14 | Rotten Tomatoes Score | Rotten Tomatoes score measures the percentage as the reviews of a given for movie or tv show |
| 15 | Metacritic Score | Metacritic score is considered the rating of a film. |
| 16 | Awards Received | Awards refer to receive for your outstanding performance or your efforts for movie or tv show. |
| 17 | Awards Nominated For | nominated for award but for the same category multiple movie or tv show get selected but only one can win it. |
| 18 | Boxoffice | The box office a measure of how popular and financially successful a movie or tv show |
| 19 | Release Date | The date on which the movie or tv show is released to be publically |
| 20 | Netflix Release Date | The date that movie or tv show is added on Netflix |
| 21 | Production House | production house is that organization which gets a movie or tv show made. |

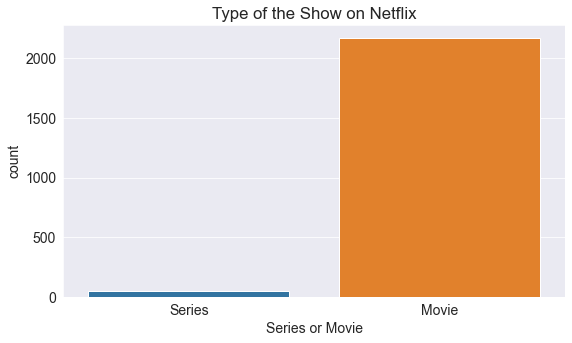
**Data Dictionary:-**

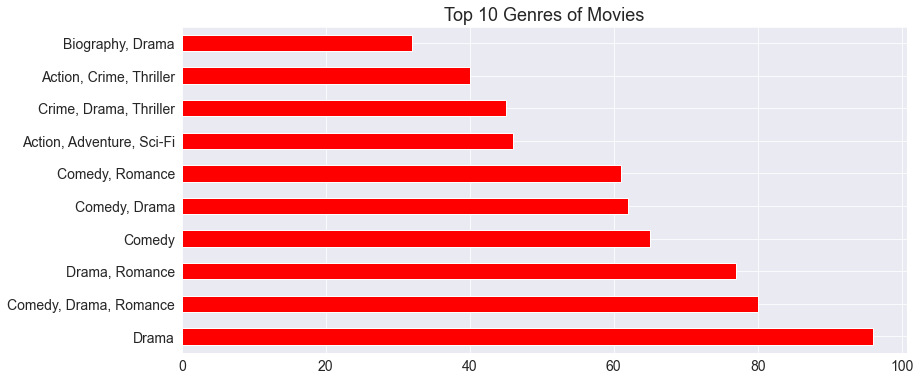
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Attributes** | **Data Format** | **Type** |
| 1 | show\_id | - | Numeric |
| 2 | Title | - | Alphabetic |
| 3 | Genre | - | Alphabetic |
| 4 | Languages | - | Alphabetic |
| 5 | Series or Movie | - | Alphabetic |
| 6 | Country Availability | - | Alphabetic |
| 7 | Runtime | - | Numeric |
| 8 | Director | - | Alphabetic |
| 9 | Writer | - | Alphabetic |
| 10 | Actors | - | Alphabetic |
| 11 | View Rating | - | Numeric |
| 12 | IMDb Votes | - | Numeric |
| 13 | IMDb Score | - | Numeric |
| 14 | Rotten Tomatoes Score | - | Numeric |
| 15 | Metacritic Score | - | Numeric |
| 16 | Awards Received | - | Numeric |
| 17 | Awards Nominated For | - | Numeric |
| 18 | Boxoffice | - | Numeric |
| 19 | Release Date | MM/DD/YYYY | Date |
| 20 | Netflix Release Date | MM/DD/YYYY | Date |
| 21 | Production House | - | Numeric |

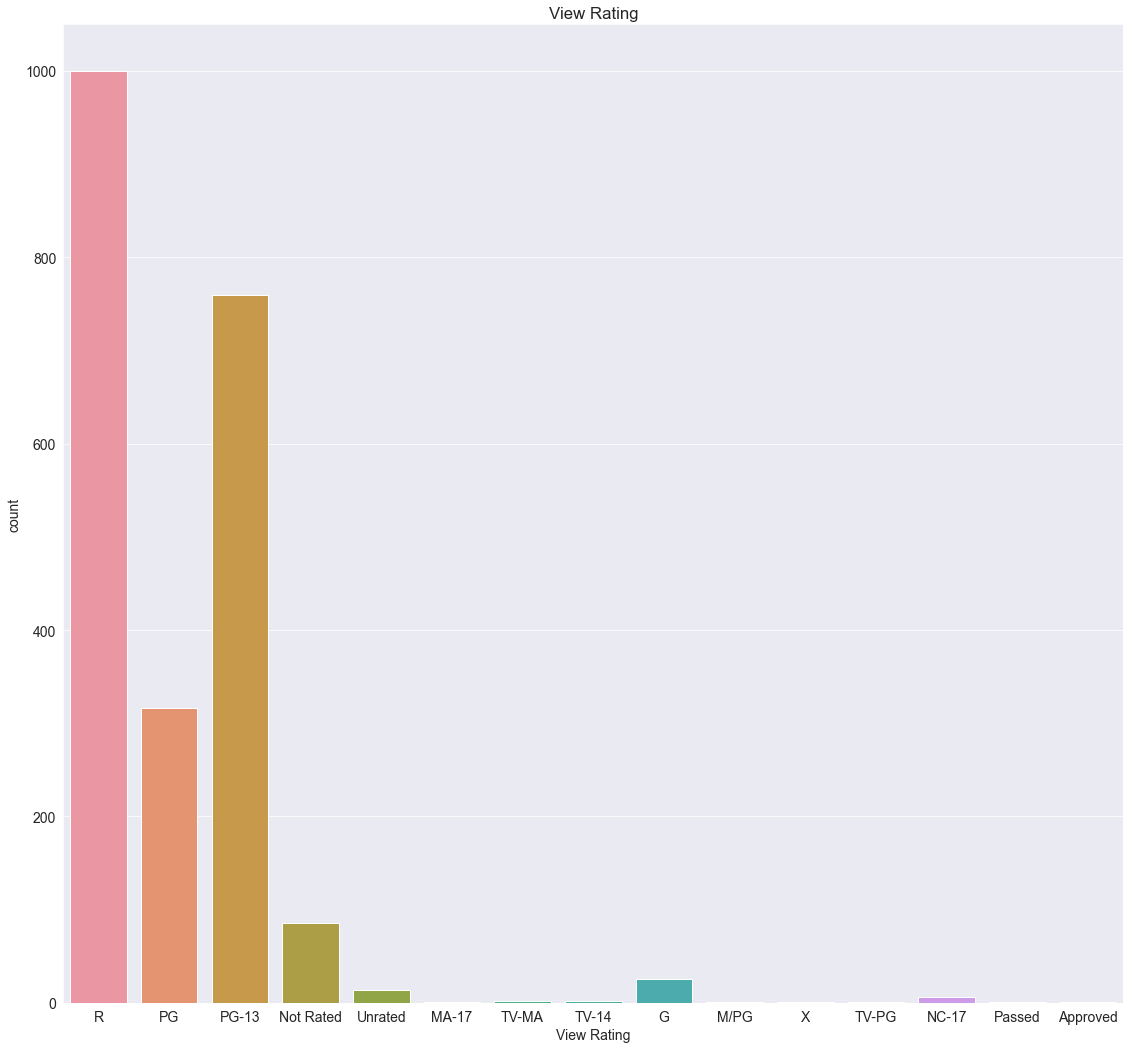
1. **Data Pre-processing & Cleaning**
2. Finding NaN values
3. Removing rows containing excessive NaN values:- 13104 rows from the dataset.
4. Validated the data:-

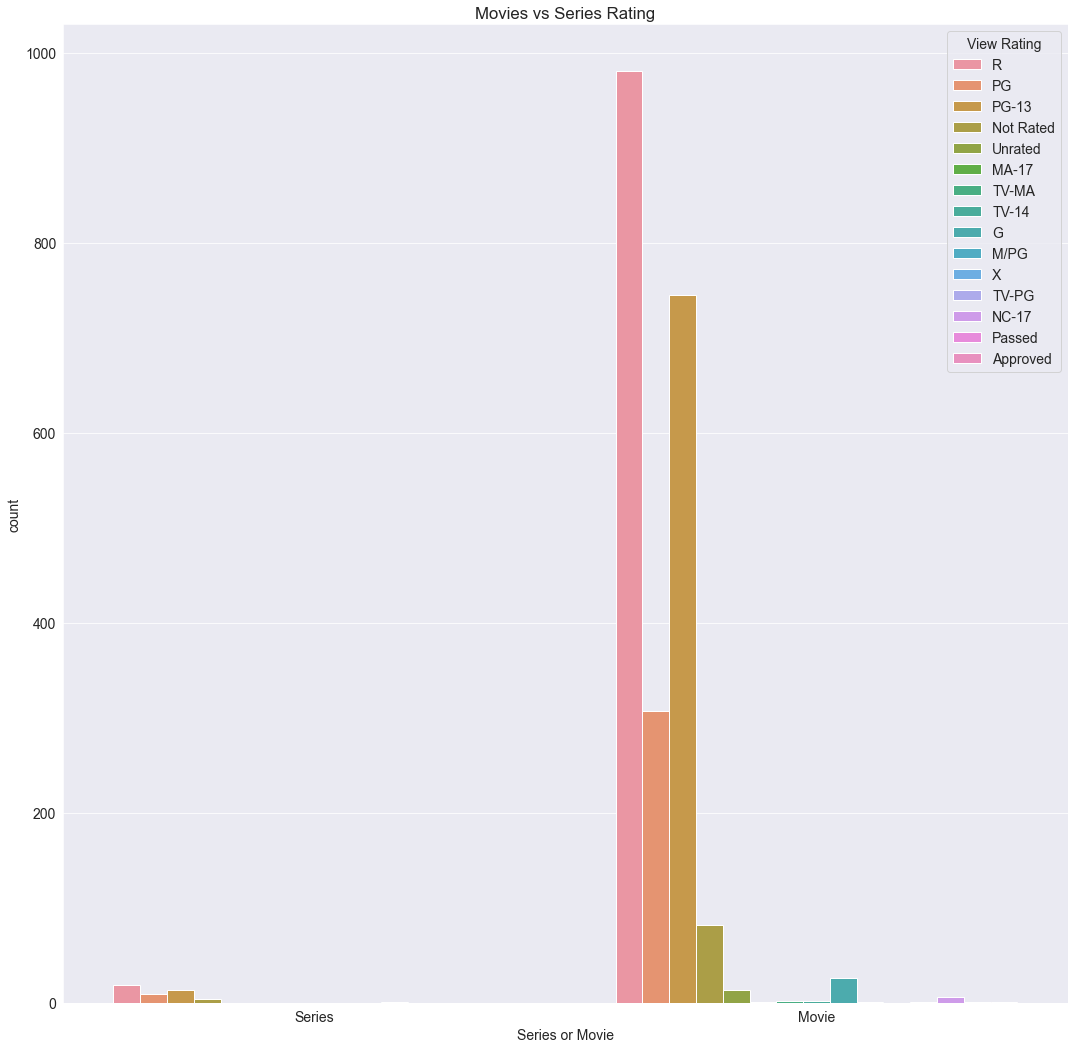
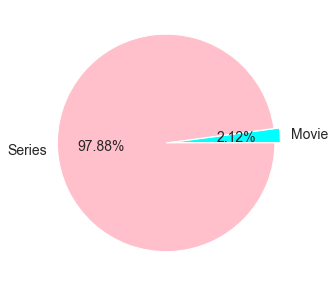
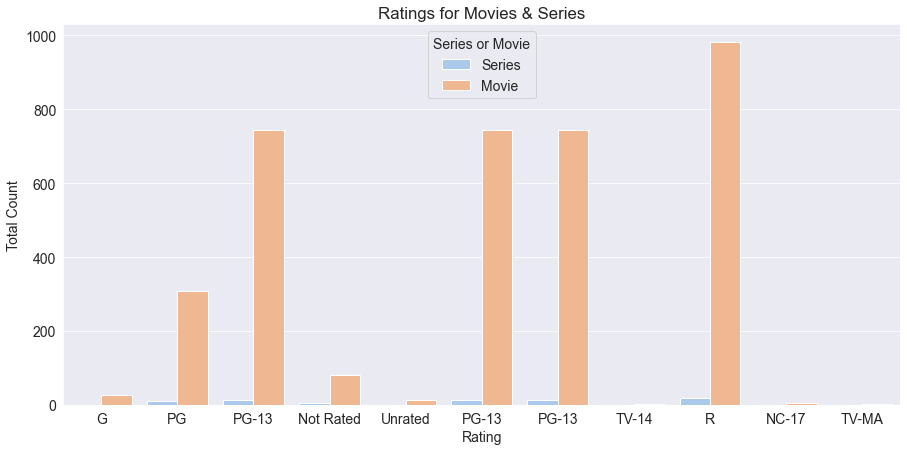
|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **Data Type** | **Previous Data Format** | **Converted Data Format** |
| Boxoffice | Numeric | Integer | Currency |

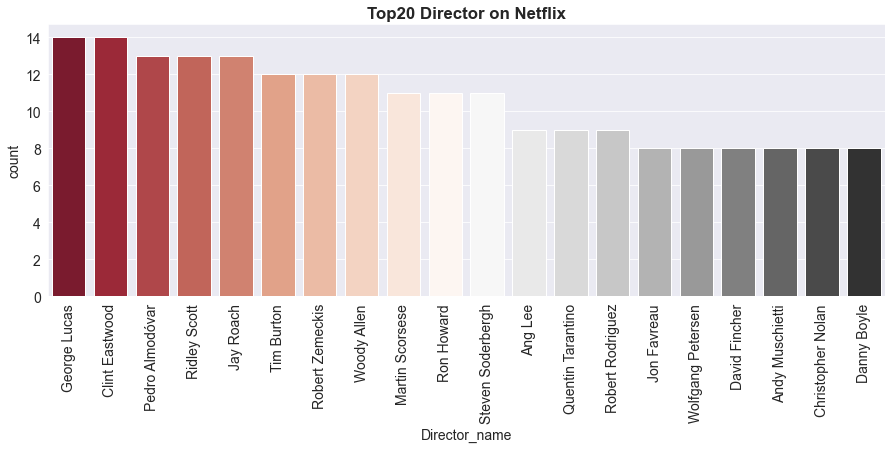
1. **Data Analysis & Visualization in Jupyter Notebook**
   * The difference in types of shows available was analysed

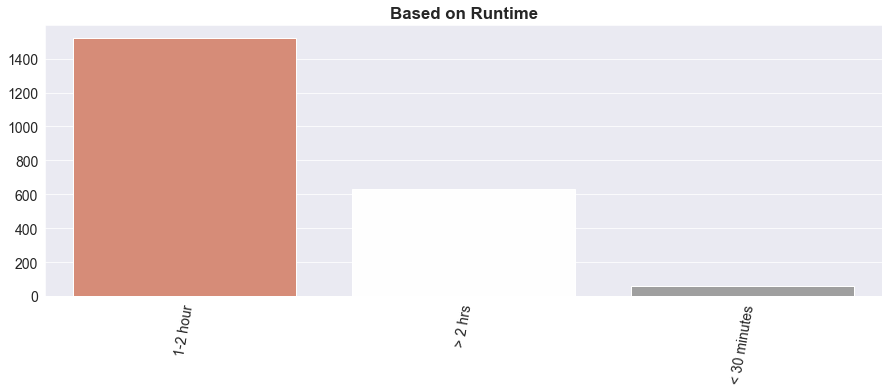


* + We will now look at what Genres of movie account for the most on Netflix using a bar graph.
* The audience prefers R and PG-13 shows more and the least preferred rating shows are MA-17.  Most of the content watched by the audience is for a not mature audience. The R rating is a type of rating given by the TV parental guidelines to a television program.

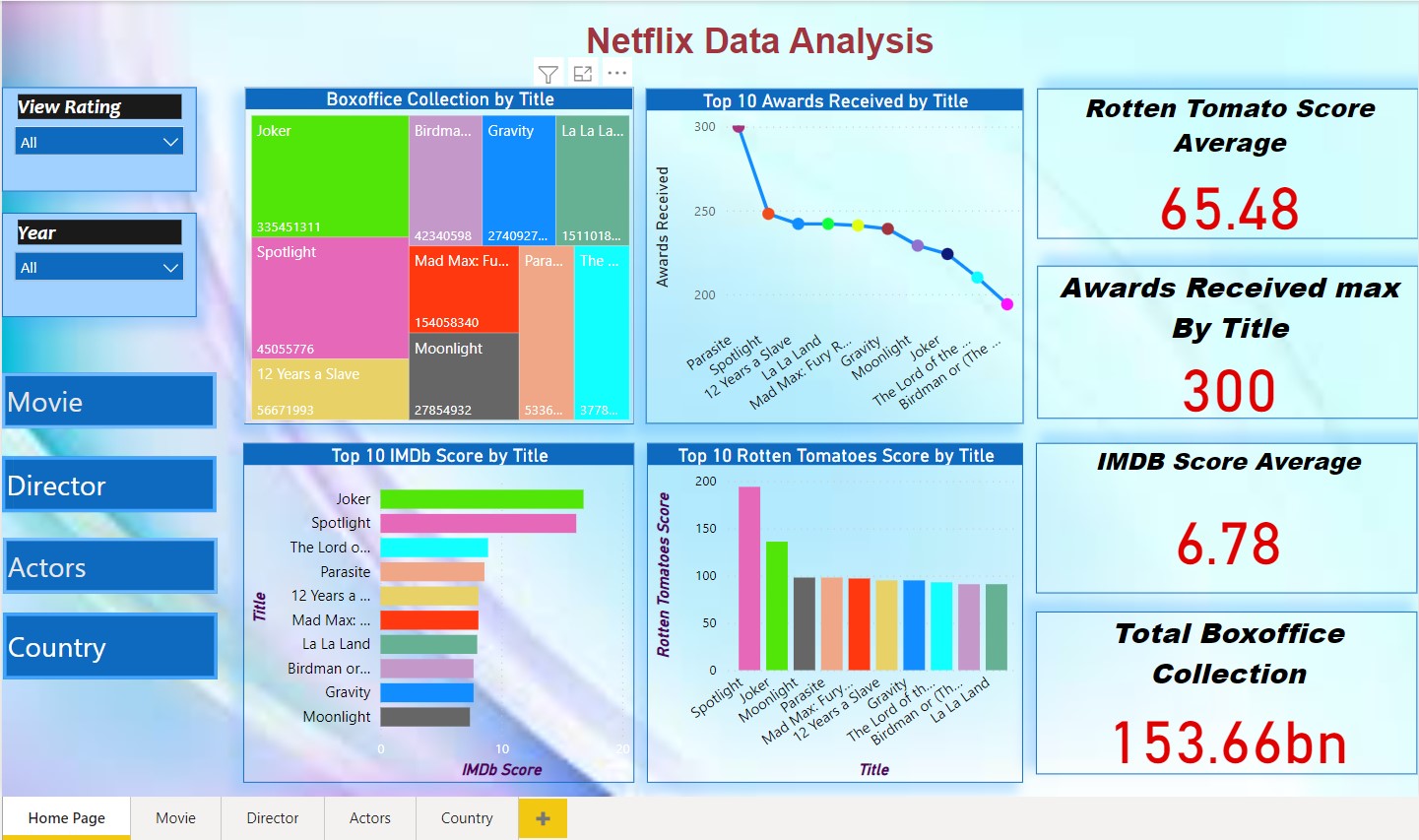
The second largest type of rating watched by the audience is PG-13 which is inappropriate for children younger than age 14. The conclusion is drawn here is most of the audience is of not the mature age.

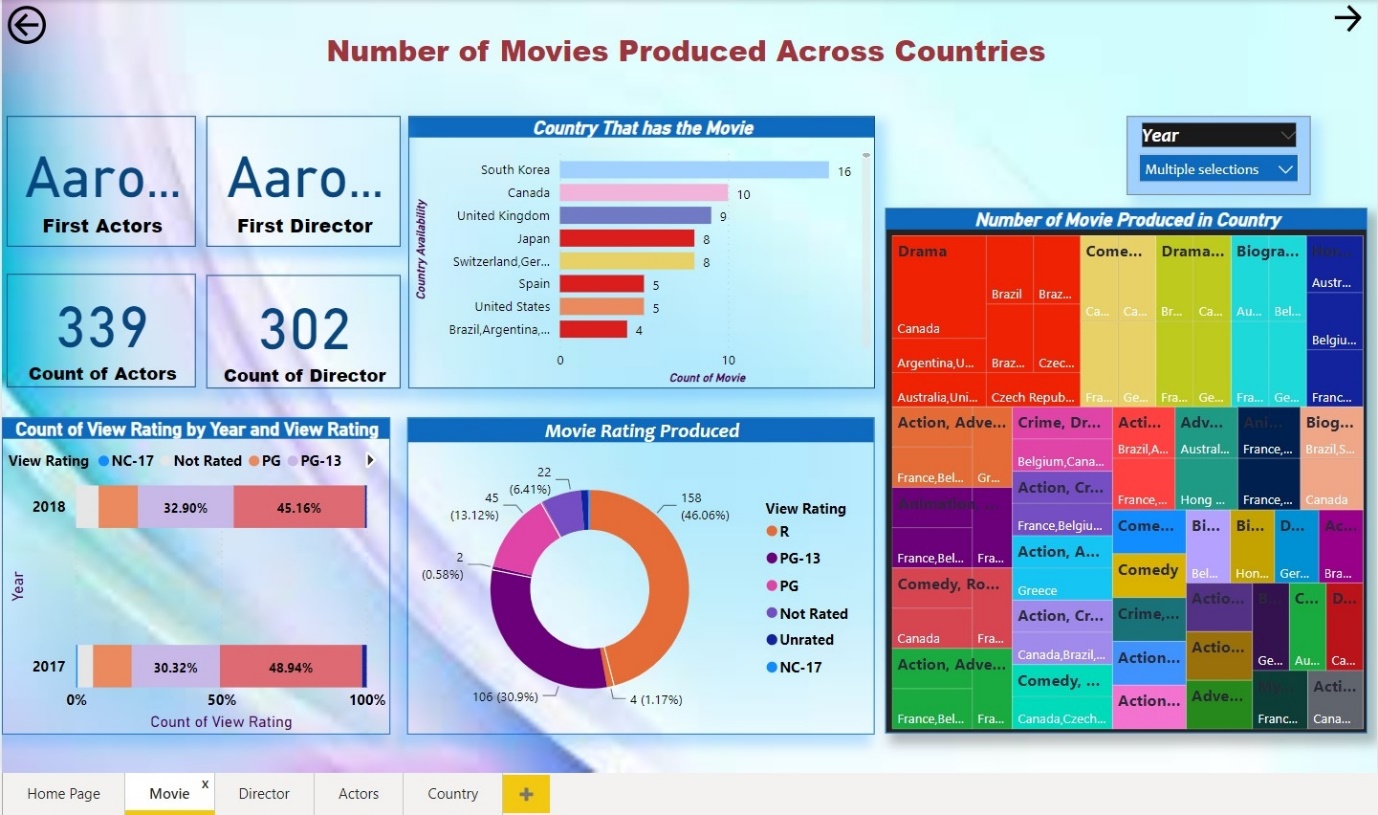
* Movies vs Series Rating
* Well, audience prefers Movies over TV Show as 97.88% audience like Movies.
* Most Popular Title
* Highest rating TV shows or Movies
* 
* Top 20 Director on Netflix

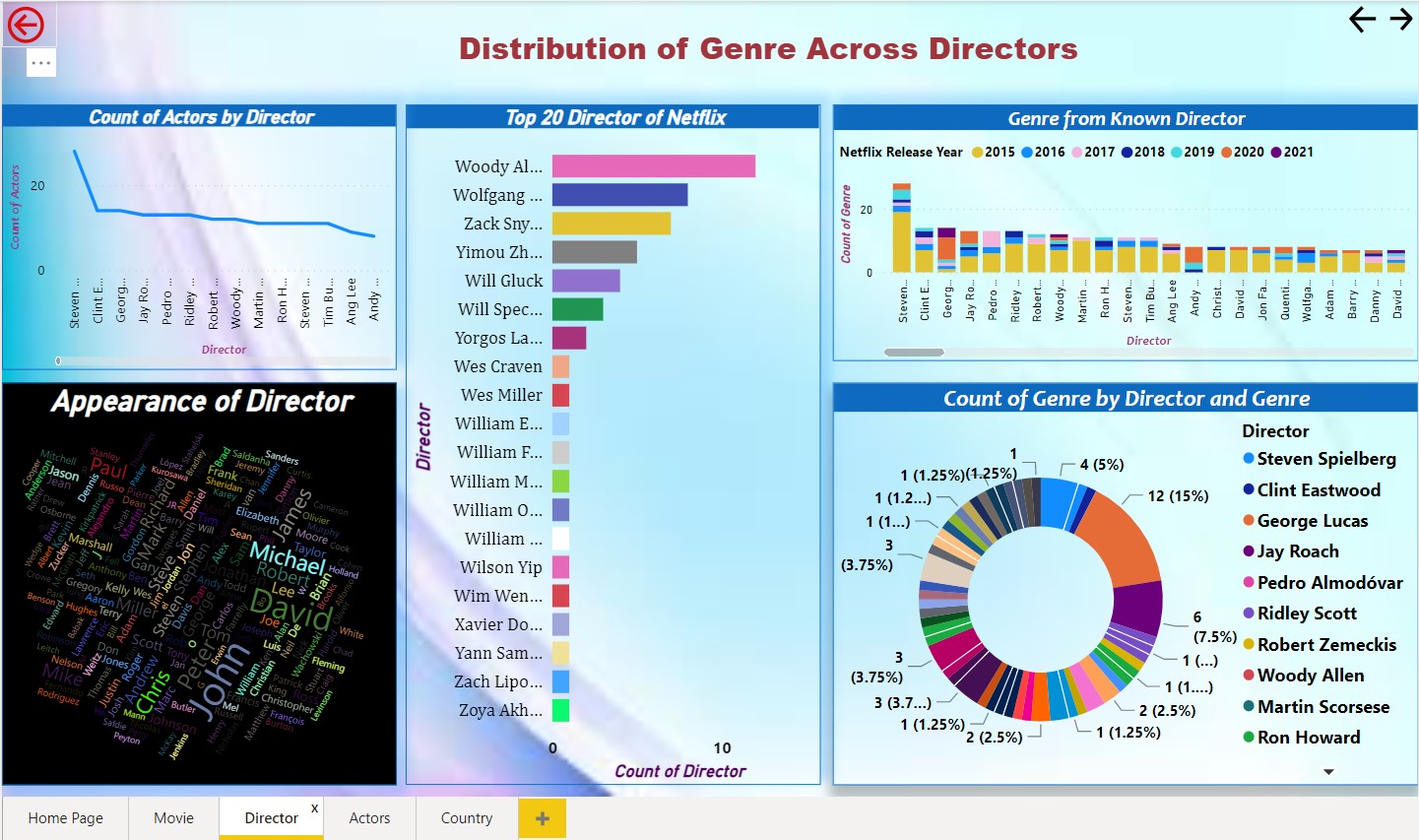


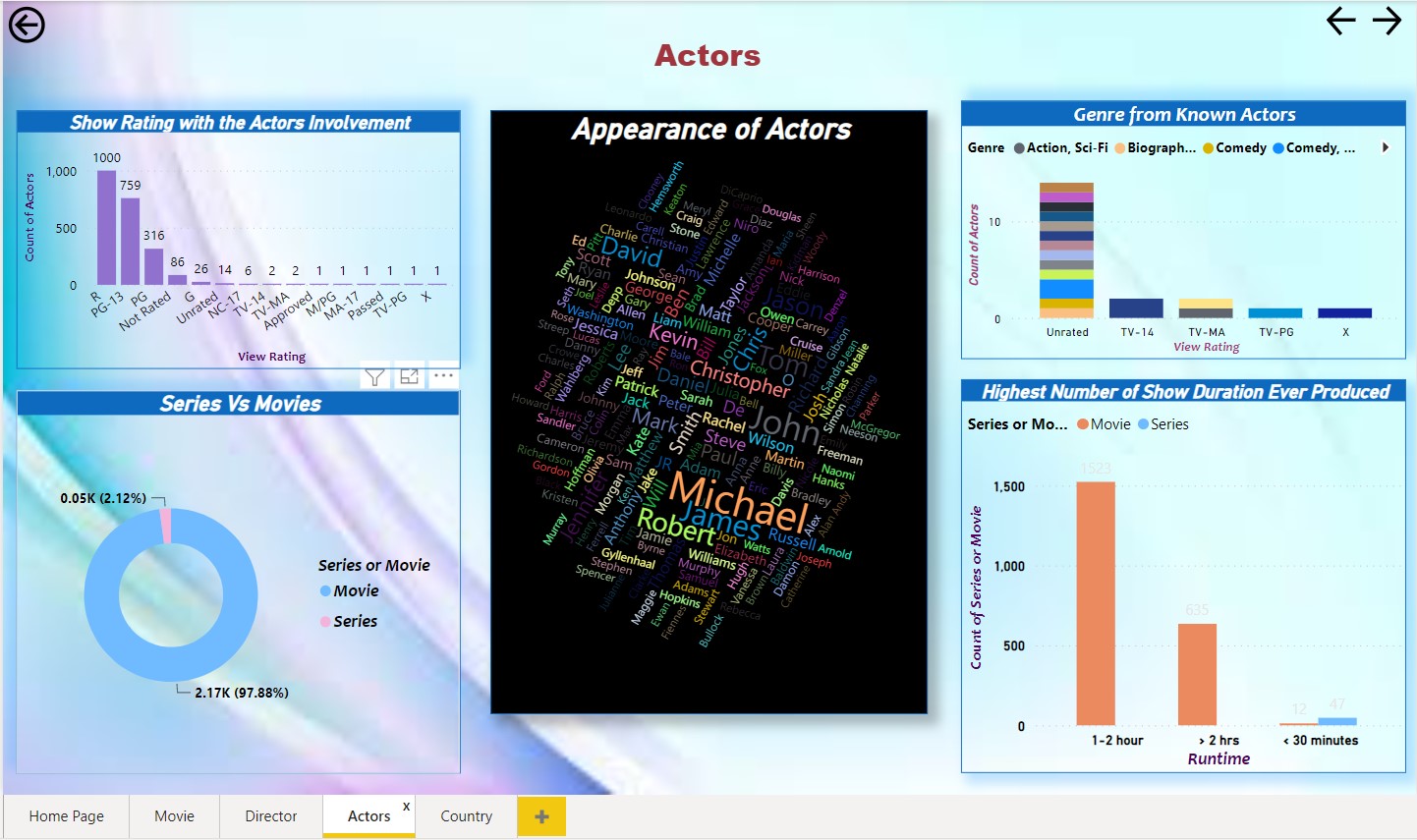
* Based on Runtime Movie or TV Show

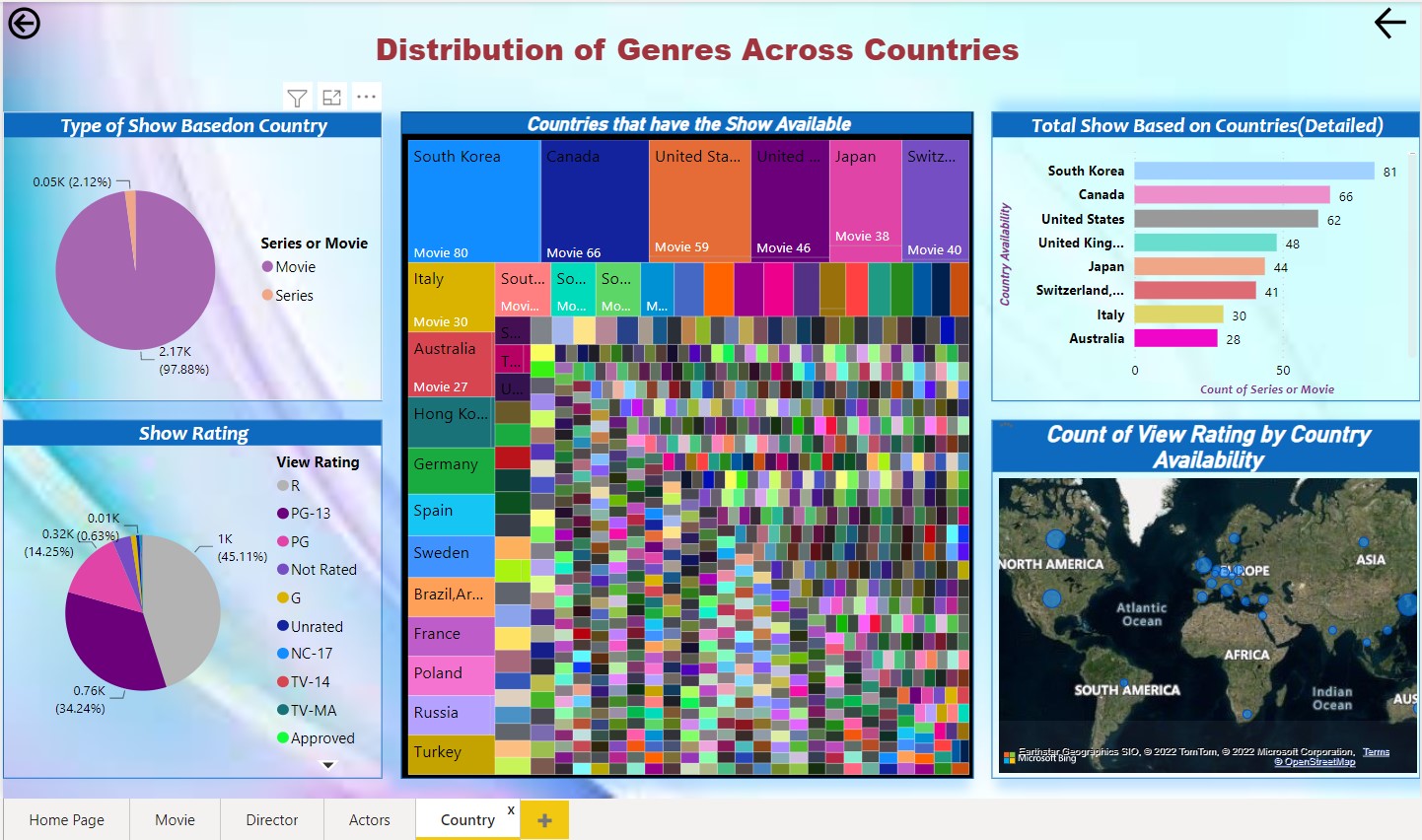
1. **Dashboard in Microsoft Power BI**

****

****

****

****

****

1. **Implementation Details**

**Jupyter Notebook:-** Firstly, we have imported all required libraries. Then we are reading the CSV file using the method read\_csv, also clean and dropping the duplicate records by modifying actual data. After that, we are visualize the data in charts formats in Jupyter Notebook.

**Microsoft Power BI:-** In Microsoft Power BI, we import the dataset and then we clean data. After that, we import charts and create dashboard.

1. **Testing**

As for testing we asked many of our friends to use and give review.

And receives many of the positive responses.

1. **Deployment**

**Jupyter Notebook:-**  Jupyter Notebook project, we deployed on Github.

**Microsoft Power BI:-** We are deployed this project on Microsoft Power BI.

1. **Future Scope**

To perform prediction analysis on the dataset with respect to IMDB rating or Box Office collection to check which genre of movie or series will performed better.

1. **Conclusion**

* Top IMDB score movie or tv show is Joker movie.
* Top Rotten Tomato score movie or tv show is spotlight movie.
* Parasite movie is winning 300 award.
* South Korea produce 81 movies.

Instructions:

1. Font- Arial
2. Main /Title Heading- 16 (bold, center aligned)
3. Heading-14 (bold)
4. Sub heading-12 (bold)
5. Normal text-12
6. Text Alignments- Justified
7. Image/Screenshot/Table Alignments- Center
8. Caption below Images/screenshot/table - Centre, Font size 8
9. References to be numbered in square box like [1] ….
10. Any code to be attached as appendix at the end like Appendix A, Appendix B …
11. Screenshots of project can also be attached as appendix

**Literature Review**

**Reviewing Literature**

**A literature review** in a thesis is a critical review of literature relevant to your field of study. It’s not a summary of the whole field, or a haphazard regurgitation of everything you’ve read in the field but an organized and critical discussion that lets your reader see what you’ve ‘made’ of the literature relevant to the topic of your thesis and your research question. It functions to contextualize your research within that research field by identifying where there are gaps in previous research that your own research will help to fill. To see how a literature review can be organized so that it provides a critical review of the field relevant to the thesis topic and shows how previous research both informs and provides a rationale for the suggested research, see the annotated discipline examples.

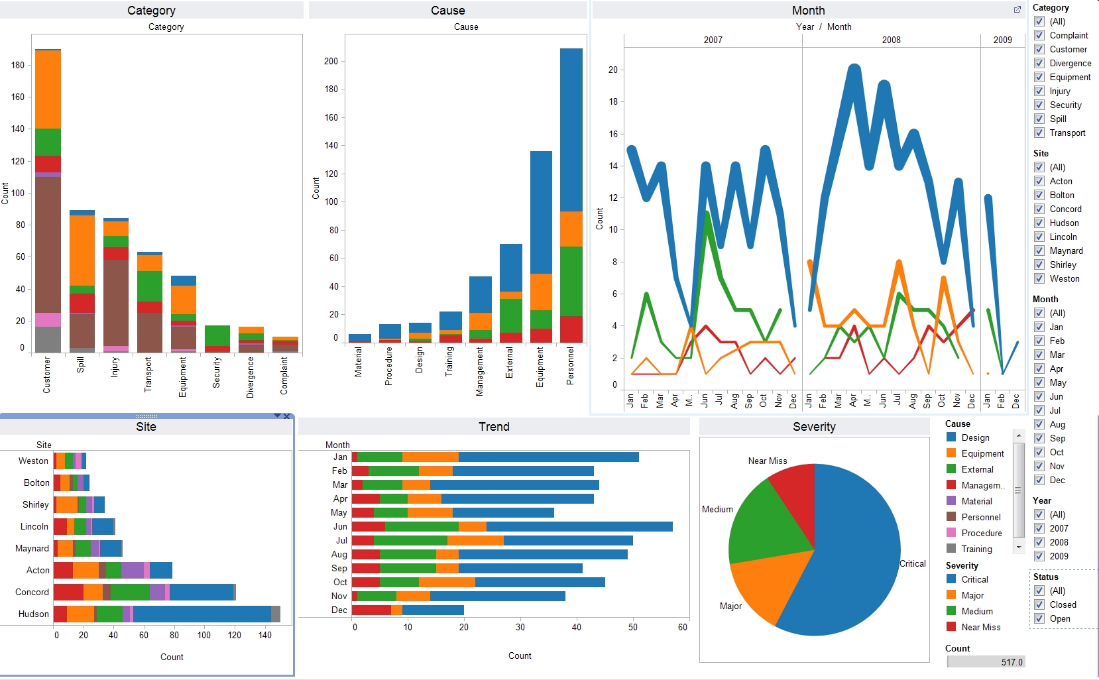


Image:1 Data visualization of electronics manufacturing market Reference: [data\_visualization\_python.jpg (1658×1026) (wp.com)](https://i1.wp.com/www.simplifiedpython.net/wp-content/uploads/2018/07/data_visualization_python.jpg?resize=1658%2C1026&ssl=1)

**Appendix A**

**Project Code**

#importing necessary libraries

import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

import plotly.graph\_objects as go

from termcolor import colored

#from plotly.offline import init\_notebook\_mode, iplo

import warnings

warnings.filterwarnings("ignore")

# read dataset in jupyter notebook

dataset1 = pd.read\_csv("dataset1.csv")

dataset1["Netflix Release Date"]= pd.to\_datetime(dataset1["Netflix Release Date"])

#dataset information

dataset1.info()

dataset1["Netflix Release Date"].dt.year

dataset1.isnull().sum()

dataset1

dataset1.duplicated().sum()

# to count how many type of shows

dataset1['Series or Movie'].value\_counts()

#count listed\_in shows

dataset1['Genre'].value\_counts()

#import libs for data visulaisation

import seaborn as sns

import matplotlib

import matplotlib.pyplot as plt

%matplotlib inline

sns.set\_style('darkgrid')

matplotlib.rcParams['font.size'] = 14

matplotlib.rcParams['figure.figsize'] = (9, 5)

matplotlib.rcParams['figure.facecolor'] = '#00000000'

#Firstly the difference in types of shows available was analysed

sns.countplot(x='Series or Movie', data=dataset1)

plt.title("Type of the Show on Netflix");

#We will now look at what Genres of movie account for the most on Netflix using a bar graph.

plt.figure(figsize=(13,6))

dataset1[dataset1["Series or Movie"]=="Movie"]["Genre"].value\_counts()[:10].plot(kind="barh",color="red")

plt.title("Top 10 Genres of Movies",size=18);

#We will now look at what Genres of movie account for the most on Netflix using a bar graph.

plt.figure(figsize=(13,6))

dataset1[dataset1["Series or Movie"]=="Series"]["Genre"].value\_counts()[:10].plot(kind="barh",color="black")

plt.title("Top 10 Genres of Series",size=18);

#Types of shows by ratings

plt.figure(figsize=(19,18))

sns.countplot(x='View Rating',data=dataset1)

plt.title("View Rating");

plt.figure(figsize=(18,18))

sns.countplot(x='Series or Movie',hue='View Rating',data=dataset1)

plt.title("Movies vs Series Rating");

dataset1['Comedy']=dataset1['Genre'].str.contains('Comedy')

dataset1['Comedy']=dataset1['Comedy'].replace({True:'Comedy',False:'NaN',})

dataset1['Crime']=dataset1['Genre'].str.contains('Crime')

dataset1['Crime']=dataset1['Crime'].replace({True:'Crime',False:'NaN',})

dataset1['Drama']=dataset1['Genre'].str.contains('Drama')

dataset1['Drama']=dataset1['Drama'].replace({True:'Drama',False:'NaN',})

dataset1['Fantasy']=dataset1['Genre'].str.contains('Drama')

dataset1['Fantasy']=dataset1['Fantasy'].replace({True:'Fantasy',False:'NaN',})

dataset1['Horror']=dataset1['Genre'].str.contains('Horror')

dataset1['Horror']=dataset1['Horror'].replace({True:'Horror',False:'NaN',})

dataset1['Romance']=dataset1['Genre'].str.contains('Romance')

dataset1['Romance']=dataset1['Romance'].replace({True:'Romance',False:'NaN',})

dataset1['Thriller']=dataset1['Genre'].str.contains('Thriller')

dataset1['Thriller']=dataset1['Thriller'].replace({True:'Thriller',False:'NaN',})

dataset1['Action']=dataset1['Genre'].str.contains('Action')

dataset1['Action']=dataset1['Action'].replace({True:'Action',False:'NaN',})

dataset1['Adventure']=dataset1['Genre'].str.contains('Adventure')

dataset1['Adventure']=dataset1['Adventure'].replace({True:'Adventure',False:'NaN',})

dataset1['Fantasy']=dataset1['Genre'].str.contains('Fantasy')

dataset1['Fantasy']=dataset1['Fantasy'].replace({True:'Fantasy',False:'NaN',})

dataset1['Sci-Fi']=dataset1['Genre'].str.contains('Sci-Fi')

dataset1['Sci-Fi']=dataset1['Sci-Fi'].replace({True:'Sci-Fi',False:'NaN',})

dataset1['Mystery']=dataset1['Genre'].str.contains('Mystery')

dataset1['Mystery']=dataset1['Mystery'].replace({True:'Mystery',False:'NaN',})

dataset1['Biography']=dataset1['Genre'].str.contains('Biography')

dataset1['Biography']=dataset1['Biography'].replace({True:'Biography',False:'NaN',})

dataset1['History']=dataset1['Genre'].str.contains('History')

dataset1['History']=dataset1['History'].replace({True:'History',False:'NaN',})

plt.figure(figsize=(10,5))

plt.pie(dataset1['Series or Movie'].value\_counts().sort\_values(),labels=dataset1['Series or Movie'].value\_counts().index,explode=[0.05,0],

autopct='%1.2f%%',colors=['cyan','pink'])

plt.show()

from wordcloud import WordCloud, ImageColorGenerator

text = " ".join(str(each) for each in dataset1.Title)

#Create and generate a word cloud image:

wordcloud = WordCloud(max\_words=200, background\_color="gray").generate(text)

plt.figure(figsize=(10,6))

plt.figure(figsize=(15,10))

#Display the generated image:

plt.imshow(wordcloud, interpolation='Bilinear')

plt.title('Most Popular Title',fontsize = 30)

plt.axis("off")

plt.show()

order = ['G','PG','PG-13','Not Rated','Unrated','PG-13','PG-13','TV-14','R','NC-17','TV-MA']

plt.figure(figsize=(15,7))

g = sns.countplot(dataset1["View Rating"], hue= dataset1["Series or Movie"] , order=order, palette="pastel");

plt.title("Ratings for Movies & Series")

plt.xlabel("Rating")

plt.ylabel("Total Count")

plt.show()

fig, ax = plt.subplots(1,2, figsize=(19, 5))

g1 = sns.countplot(dataset1["View Rating"], order=order,palette="Set2", ax=ax[0]);

g1.set\_title("Ratings for Movies") g1.set\_xlabel("Rating") g1.set\_ylabel("Total Count")

g2 = sns.countplot(dataset1["View Rating"], order=order,palette="Set2", ax=ax[1]);

g2.set(yticks=np.arange(0,1600,200)) g2.set\_title("Ratings for Series") g2.set\_xlabel("Rating")

g2.set\_ylabel("Total Count")

fig.show()

dataset1['Director\_name'] = dataset1['Director'].apply(lambda x : x.replace(' ,',',').replace(', ',',').split(','))

Director\_count = [] for i in dataset1['Director\_name']: Director\_count += i

Director\_dict = dict((i, Director\_count.count(i)) for i in Director\_count)

dataset1\_Director\_count = pd.DataFrame(Director\_dict.values(),Director\_dict.keys()).reset\_index().sort\_values(0,ascending=False).rename(

columns = {'index' : 'Director\_name', 0 : 'count'}).iloc[1:21]

plt.figure(figsize=(15,5))

sns.barplot(x='Director\_name',y='count',data=dataset1\_Director\_count,palette="RdGy")

plt.title("Top20 Director on Netflix",fontweight="bold")

plt.xticks(rotation=90)

plt.show()

dataset1\_series1 = dataset1[dataset1["Series or Movie"] == "Series"]

dataset1\_movies1 = dataset1[dataset1["Series or Movie"] == "Movie"]

dataset1\_content = dataset1['Netflix Release Year'].value\_counts().reset\_index().rename(columns = {

'Netflix Release Year' : 'count', 'index' : 'Netflix Release Year'}).sort\_values('Netflix Release Year')

dataset1\_content['percent'] = dataset1\_content['count'].apply(lambda x : 100\*x/sum(dataset1\_content['count']))

dataset1\_series1 = dataset1\_series1['Netflix Release Year'].value\_counts().reset\_index().rename(columns = {

'Netflix Release Year' : 'count', 'index' : 'Netflix Release Year'}).sort\_values('Netflix Release Year')

dataset1\_series1['percent'] = dataset1\_series1['count'].apply(lambda x : 100\*x/sum(dataset1\_series1['count']))

dataset1\_movies1 = dataset1\_movies1['Netflix Release Year'].value\_counts().reset\_index().rename(columns = {

'Netflix Release Year' : 'count', 'index' : 'Netflix Release Year'}).sort\_values('Netflix Release Year')

dataset1\_movies1['percent'] = dataset1\_movies1['count'].apply(lambda x : 100\*x/sum(dataset1\_movies1['count']))

t1 = go.Scatter(x=dataset1\_movies1['Netflix Release Year'], y=dataset1\_movies1["count"], name="Movies", marker=dict(color="#a678de"))

t2 = go.Scatter(x=dataset1\_series1['Netflix Release Year'], y=dataset1\_series1["count"], name="Series", marker=dict(color="#6ad49b"))

t3 = go.Scatter(x=dataset1\_content['Netflix Release Year'], y=dataset1\_content["count"], name="Total Contents", marker=dict(color="brown"))

data = [t1, t2, t3]

layout = go.Layout(title="Content added over the years", legend=dict(x=0.1, y=1.1, orientation="h"))

fig = go.Figure(data, layout=layout)

fig.show()

import plotly.graph\_objects as go

from plotly.offline import init\_notebook\_mode, iplot

filtered\_countries = dataset1.set\_index('Title')["Country Availability"].str.split(', ', expand=True).stack().reset\_index(level=1, drop=True);

filtered\_countries = filtered\_countries[filtered\_countries != 'Country Unavailable']

iplot([go.Choropleth(

locationmode='country names',

locations=filtered\_countries,

z=filtered\_countries.value\_counts()

)])

plt.figure(figsize=(15,5))

sns.barplot(x = dataset1["Runtime"].value\_counts().head(10).index,

y = dataset1["Runtime"].value\_counts().head(10).values,palette="RdGy")

plt.xticks(rotation=80)

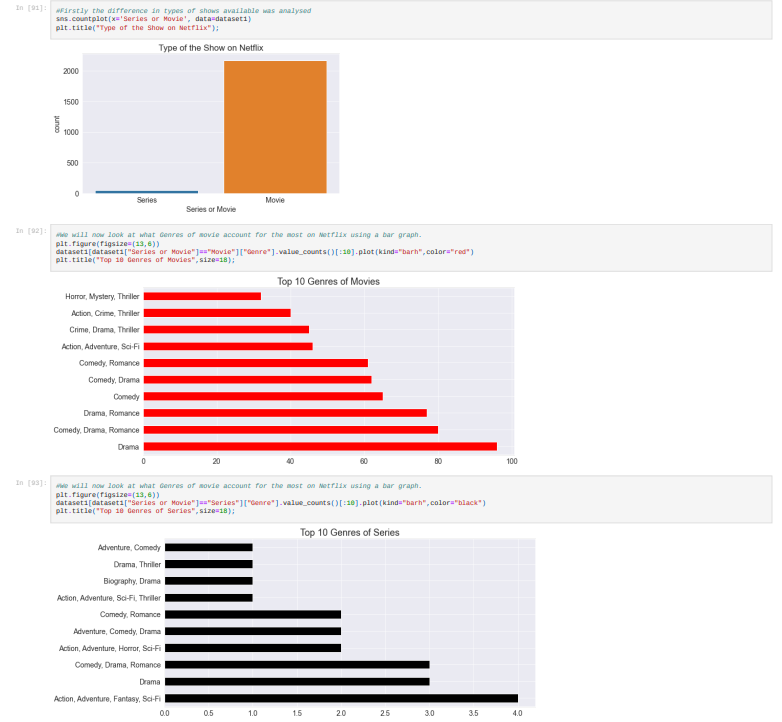
plt.title("Based on Runtime",fontweight="bold")

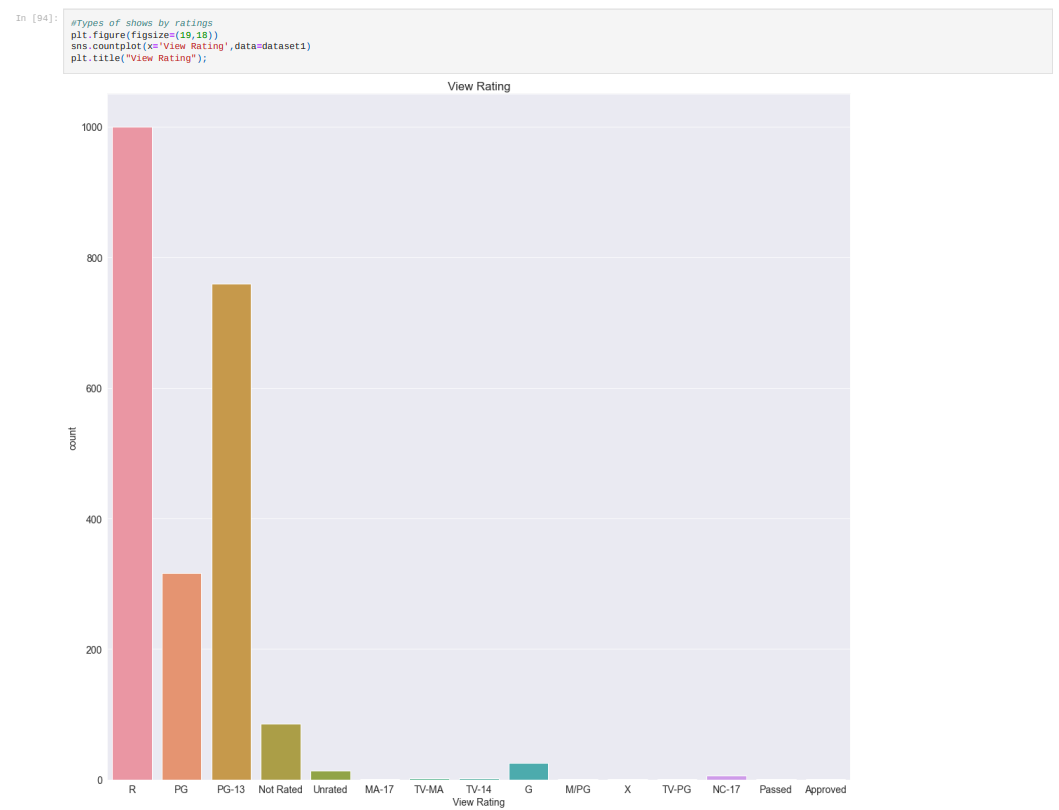
plt.show()

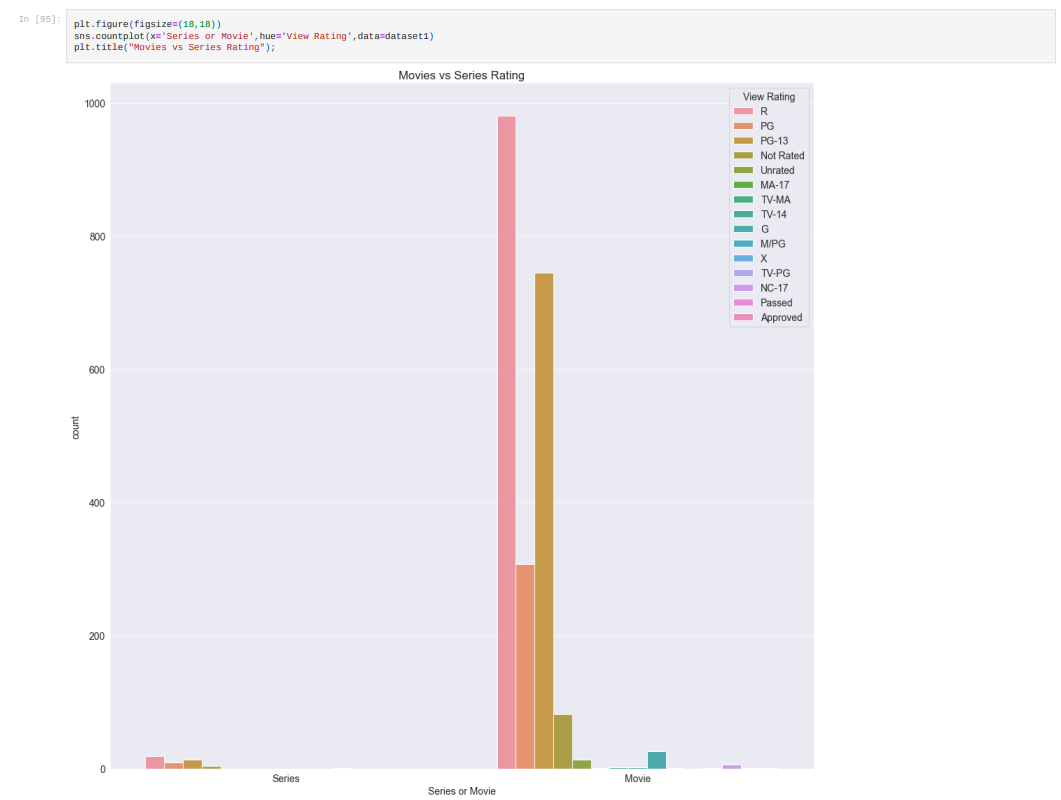
**Appendix B**

**Screenshot of Project**

1. **Screenshot’s of Jupyter Notebook**

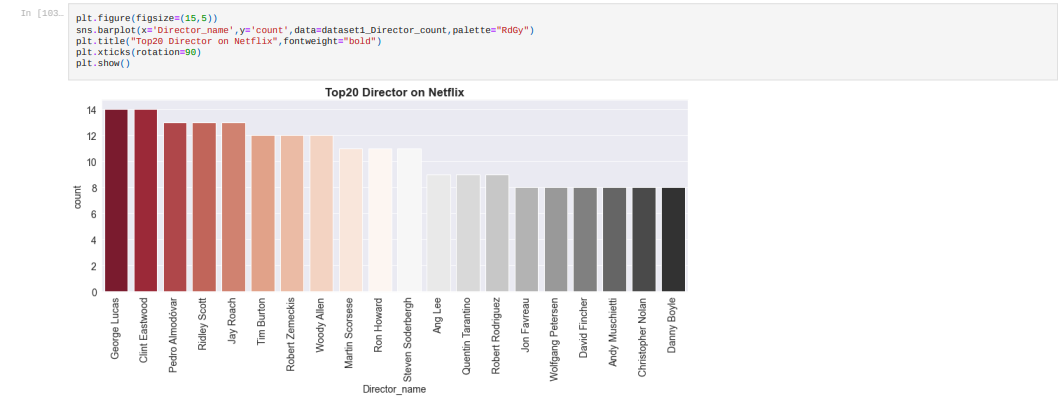


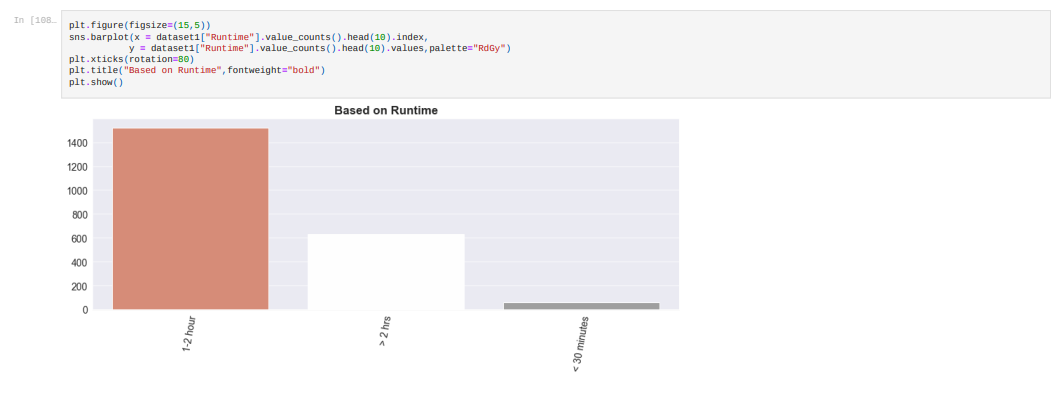




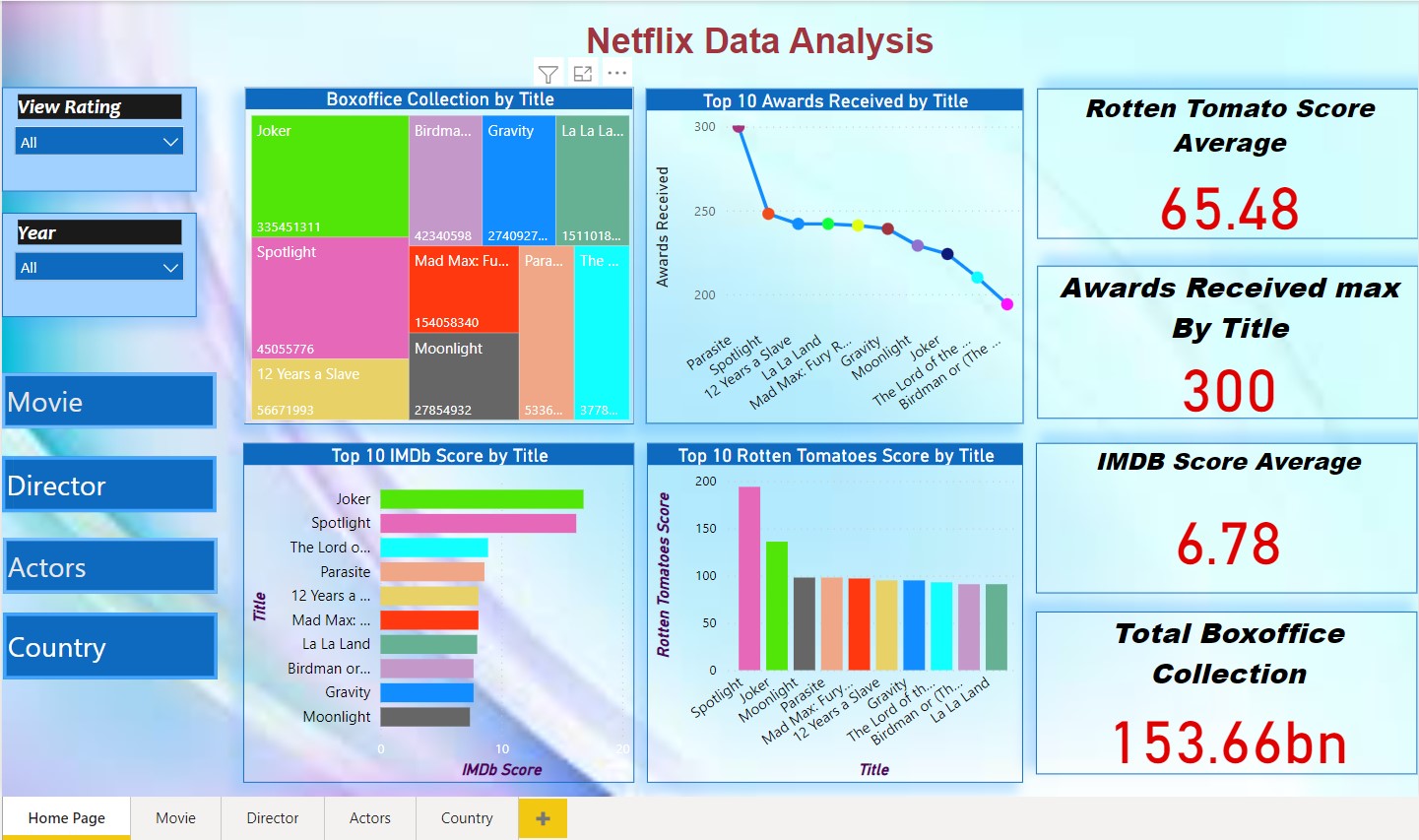


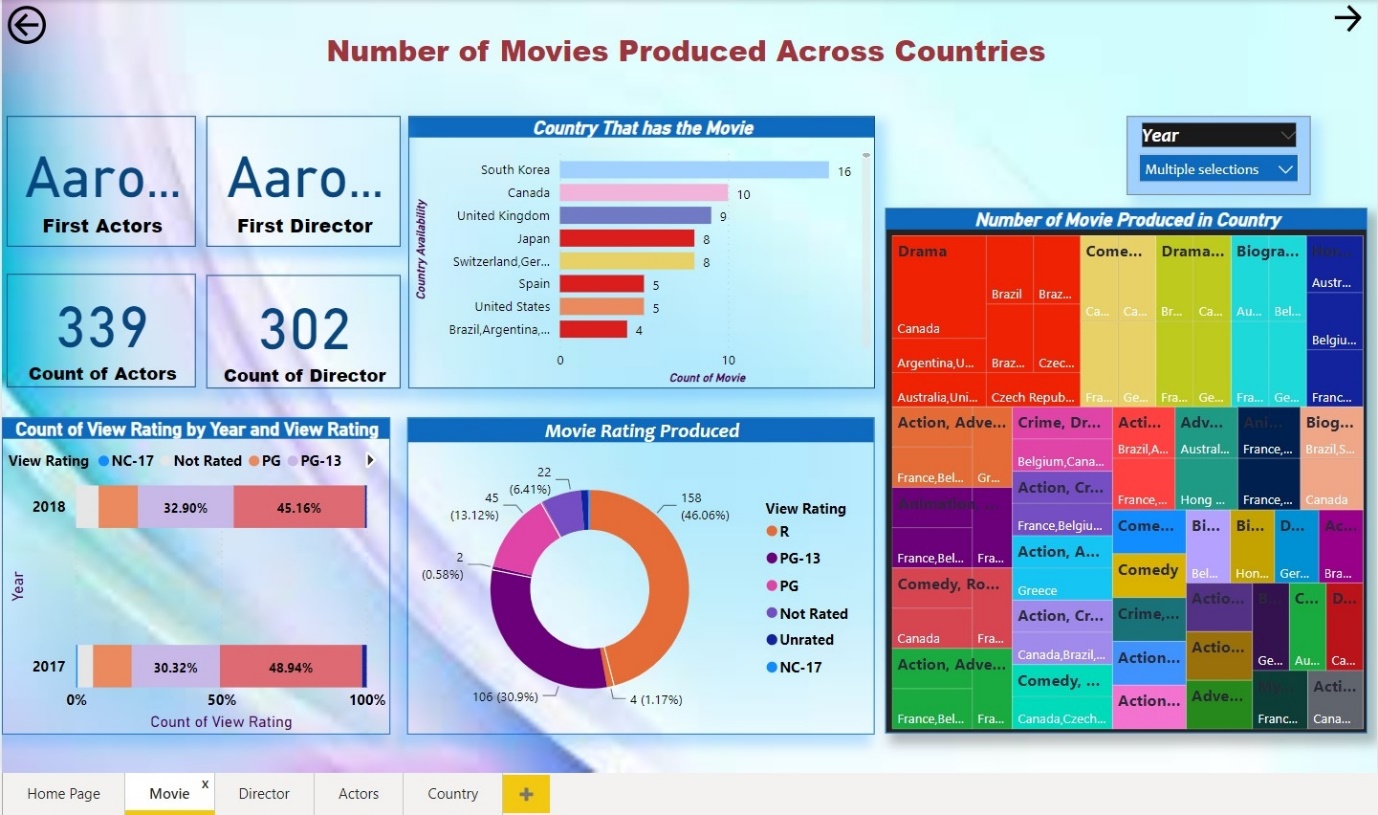


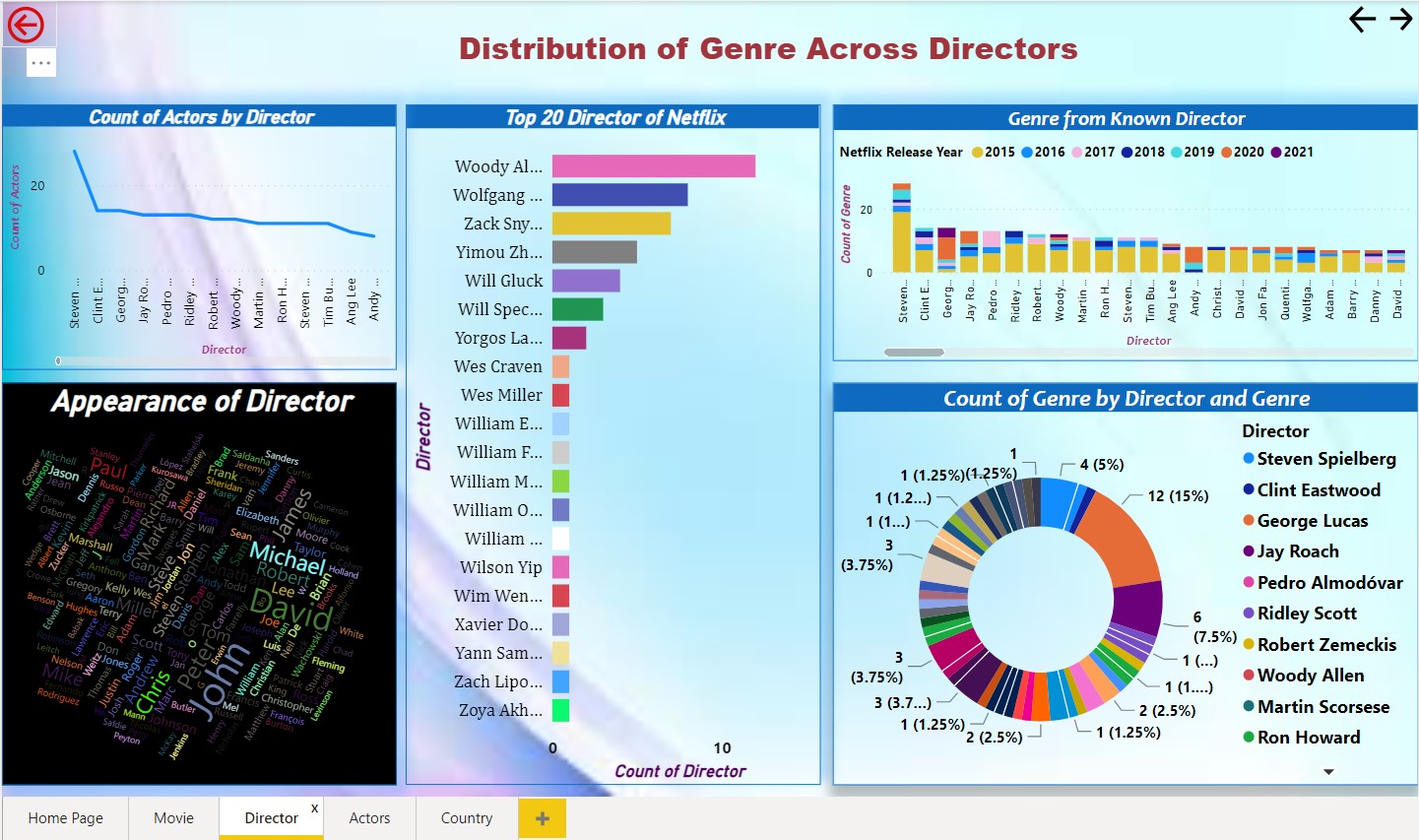


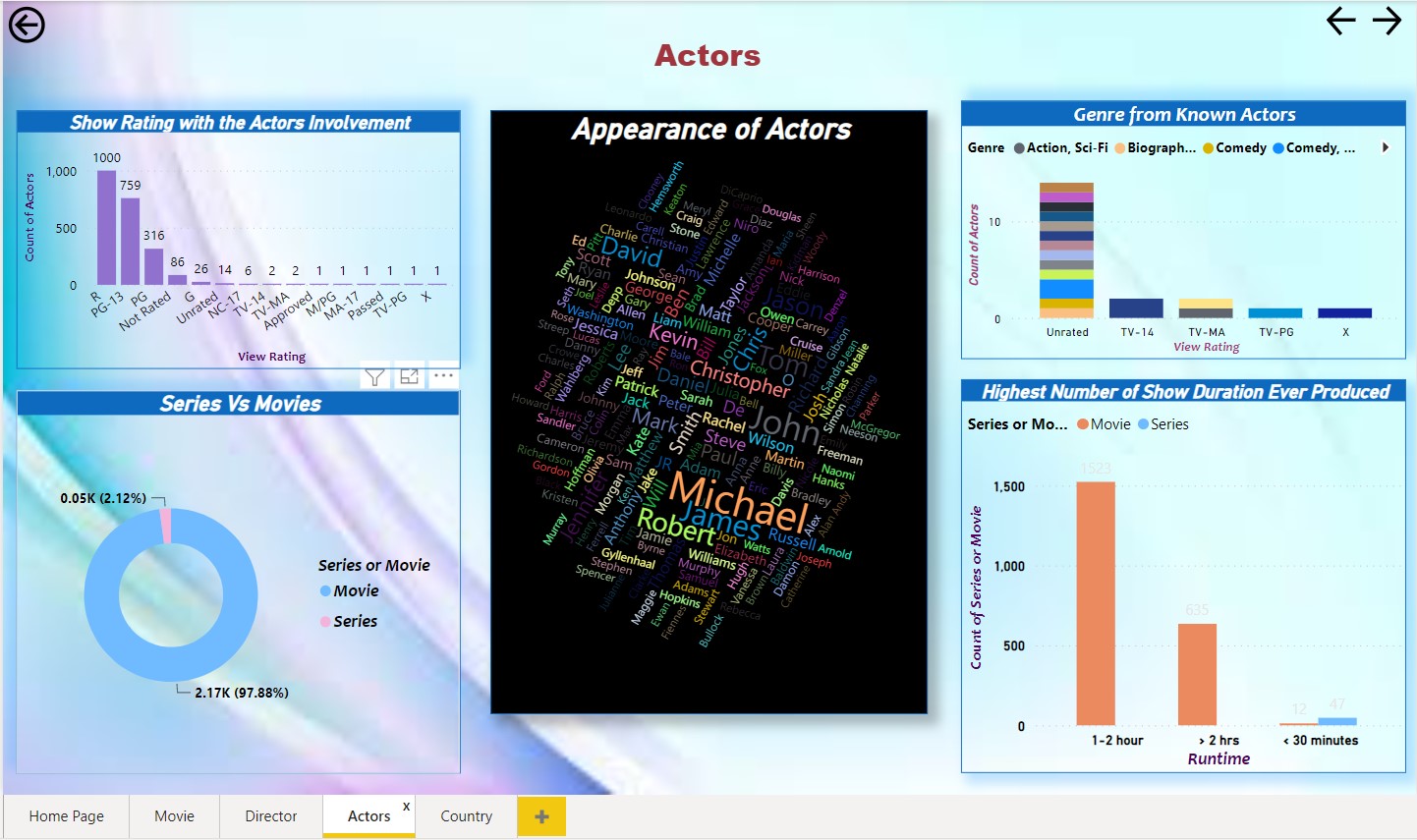


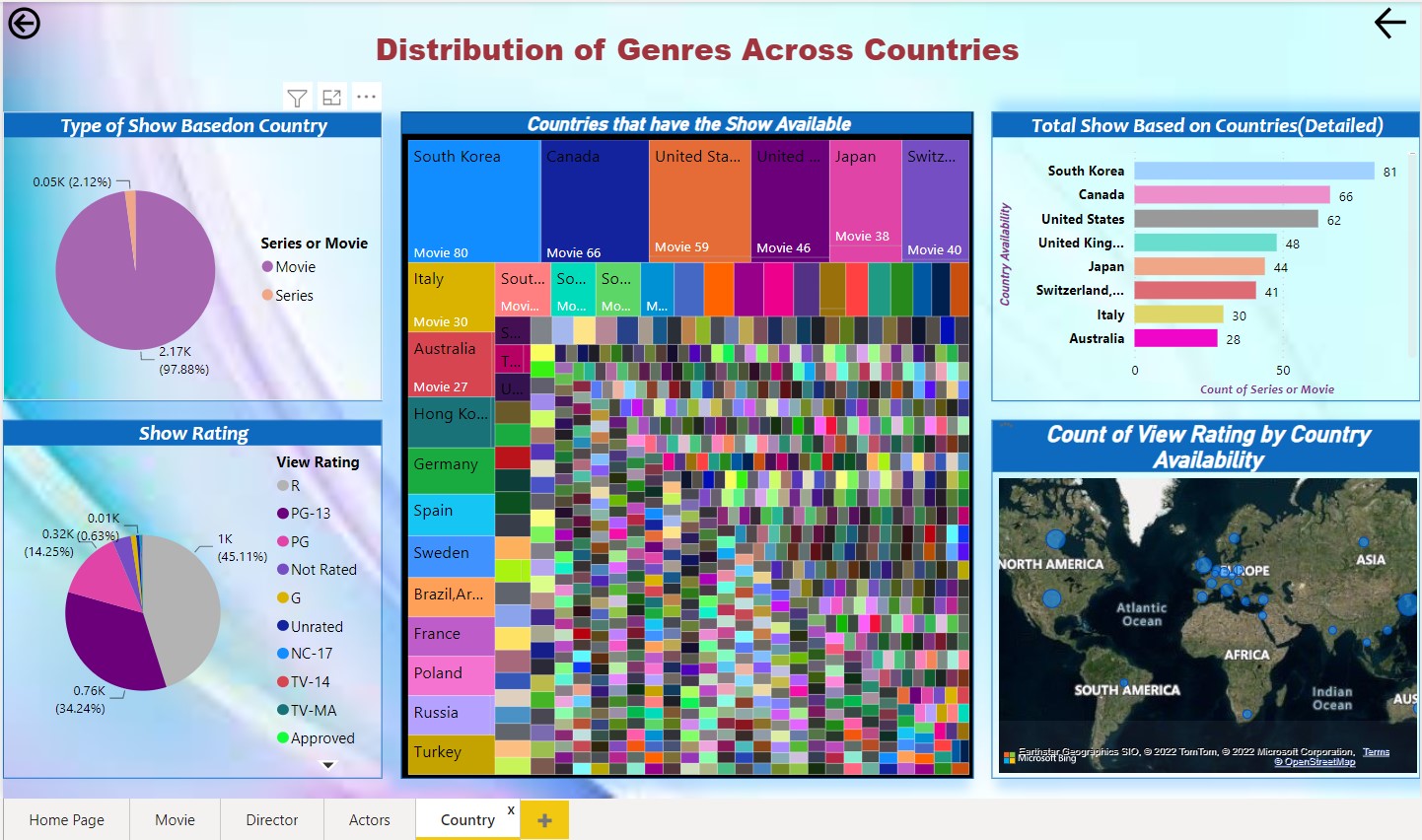
1. **Screenshot’s of Microsoft Power BI Dashboard**

****

****

****

****

****

**References**

[Latest Netflix data with 26+ joined attributes | Kaggle](https://www.kaggle.com/ashishgup/netflix-rotten-tomatoes-metacritic-imdb)