TEAM Return Zero

1. Introduction:

1.1. Overview:

Our project is an interactive **OTT Analysis Dashboard** which will provide useful insights to the users due to which the customers can easily decide which OTT platform is best suited for their requirements.

1.2. Purpose:

The main objective of this challenge is to investigate different OTT platform datasets and analyse the utilization of them. The insights could be ideal for the actual OTT platform to realise where they really stand in the competitive market.

2. Literature survey:

2.1 Existing problems:

For example, let us assume that the User wants to know which OTT platform contains the highest number of English movies for his/her subscription. But the user does not know which OTT application suits this criterion the best.

2.2 Proposed solutions:

Our Dashboard provides insights like:

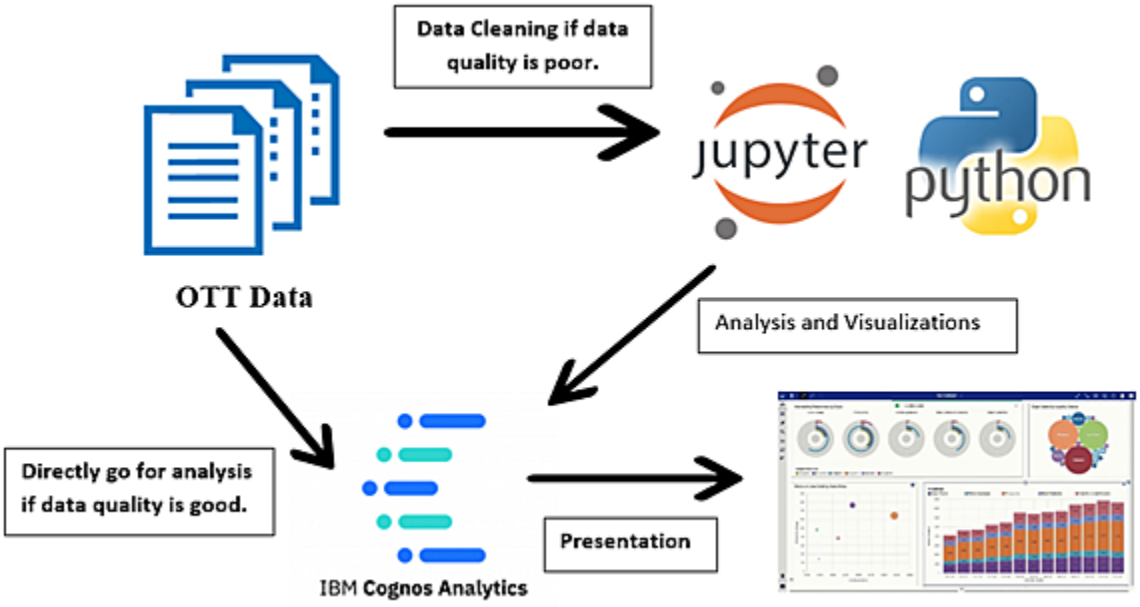
a. **Number of movies released per year in different OTT platforms.**
b. **Top films based on IMDB ratings in different OTT platforms.**
c. **Forecasting number of movies which could be releasing in the years to come.**
d. **Analysing how the running time of movies have changed over the years in different languages.**
e. **Number of movies released based on the age category in each OTT application.**
f. **Various genre of films released in different OTT applications.**
g. **Number of films released in different languages in different OTT apps.**
h. **Quantity vs Quality of each OTT platforms, etc.**

Just as an example, lets us assume that the user wants to the no. of movies released in various languages. He/she notices this section in the dashboard. Then the user considers the

analysis of all OTT applications which is showcased in the dashboard and gets to know that Netflix is the optimal application for this particular requirement. Therefore, the user opts for a Netflix for subscription.

3. Theoretical Analysis:

3.1. Block diagram:



3.2. Hardware/ Software requirements:

Any device which can connect to internet can access the dashboard, provided the user has the required IBM permissions. A browser is recommended.

4. Experimental Investigations:

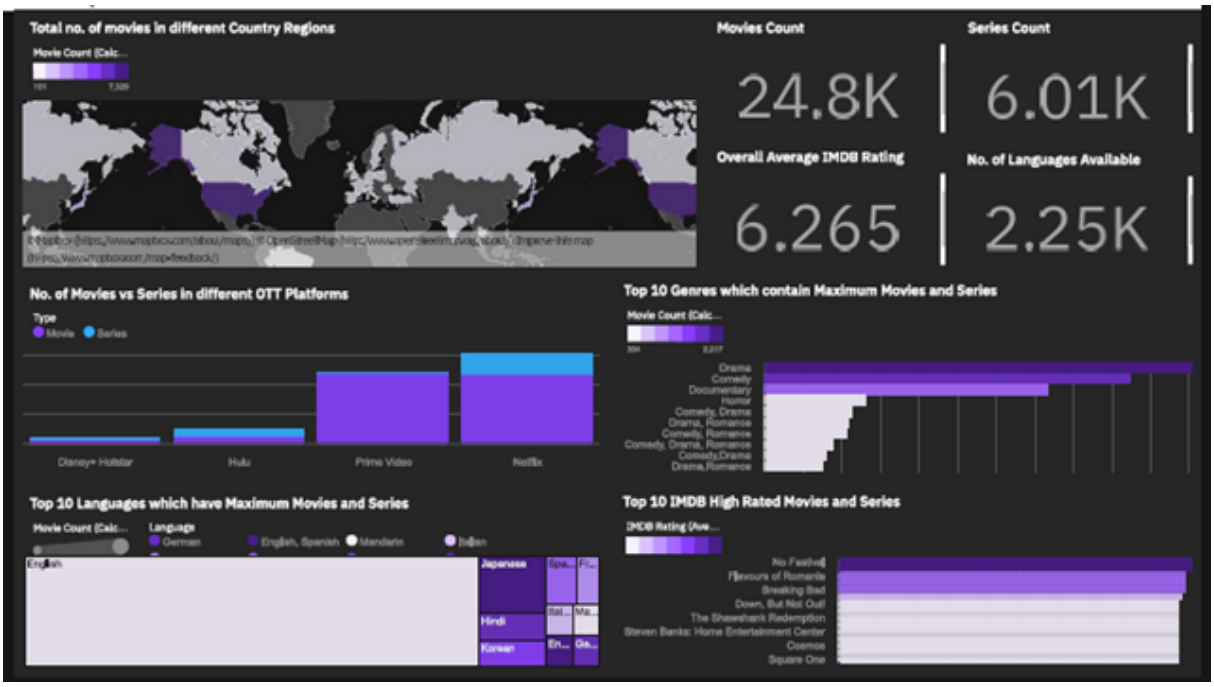
We tried to gather data from various sources, so that we could use the most of the resources available in net. The idea to scrap data raised, but as we could not find sources with all information, the idea was dropped.

5. Workflow:

The work flow is simple. The link will take the user to IBM's website. After validating the credentials of user, the dashboard will be displayed. The dashboard has 3 tabs.

6. Result:


Quantity vs Quality of different OTT
OTT
Disney+ Hotstar
Hulu
Netflix
Prime Video
Quality (Calculated)
Movie Count
Movies Count
24.8K
Series Count
6.01K
Overall Average IMDB Rating
6.265
No. of Languages Available
2.25K
Average Running Time of Movies and Series in each Year
Type
Movie
Series
No. of Movies and Series Released in each Year.
No. of Movies and Series in different Age Categories
NC-17
R
PG
G
TV-PG
TV-Y
Approved


Total no. of movies in different Country Regions
Movies Count
24.8K
Series Count
6.01K
Overall Average IMDB Rating
6.265
No. of Languages Available
2.25K
No. of Movies vs Series in different OTT Platforms
Disney+ Hotstar
Hulu
Prime Video
Netflix
Top 10 Genres which contain Maximum Movies and Series
Drama
Comedy
Documentary
Horror
Comedy, Drama
Drama, Romance
Comedy, Romance
Comedy, Drama, Romance
Comedy,Drama
Drama,Romance
Top 10 Languages which have Maximum Movies and Series
Language
English
Japanese
Hindi
Korean
Top 10 IMDB High Rated Movies and Series
No Festival
Flavours of Romantic
Breaking Bad
Down, But Not Out!
The Shawshank Redemption
Steven Banks: Home Entertainment Center
Cosmos
Square One

7. Advantages and Disadvantages:

Advantages:

The main advantages of this dashboard are unique quantity vs quality comparison which will help users to choose the best OTT platforms. Also, the data was collected and pre-processed from various sources, so as to utilize as much data as we could for this analysis. The movie and series trends with respect to durations, number of releases in each year, genre, languages were analysed.

Disadvantages:

The whole data was not available for all the OTT platforms, so these insights can be considered as an analysis done on sample data.

8. Applications:

This analysis will help the users to choose the best OTT platform for themselves. Also, it will help the various OTT platform to scale where they stand among the other OTT platforms. The application areas include the above fields but not limited to the above fields.

9.Conclusion:

Country wise movie count, number of movies in each genre and languages, some top IMDB rated movies, quality vs quantity comparison, average run time of movies in each year,

number of movie releases in each year are the insights gained from the dashboard. The number of movie releases in the upcoming years were forecasted.

10. Future Scope:

With more data, the accuracy of insights from analytics can be improved .

11. Bibliography:

Appendix:

Code:

```
#!/usr/bin/env python
 # coding: utf-8
 # ## IMPORTING LIBRARIES
 # In[1]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# In[74]:
netflix_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\New
Dataset\netflix-rotten-tomatoes-metacritic-imdb.csv")
 netflix_df.head()
# In[75]:
netflix_df.columns
# In[76]:
netflix_df.info()
# In[77]:
netflix_df.describe()
# ## Data Cleaning
 # ### 1. Removing Unwanted Columns
 # In[78]:
unwanted_cols = ['Tags',
       'Hidden Gem Score','Director',
       'Writer', 'Actors',
       'Rotten Tomatoes Score', 'Metacritic Score', 'Awards Received',
       'Awards Nominated For', 'Boxoffice',
    'Production House', 'Netflix Link', 'IMDb Link',
       'Summary', 'IMDb Votes', 'Image', 'Poster', 'TMDb Trailer',
       'Trailer Site']
 netflix_df.drop(columns=unwanted_cols,inplace=True)
# In[79]:
netflix_df.head()
# ### 2. Handling Missing Values
 # In[80]:
netflix_df.isnull().sum()
```

```
# In[81]:
 # 1. Handle missing values in Languages column by replacing with most freq value
freq_lang = netflix_df.Languages.value_counts().index[0]
 netflix_df['Languages'].fillna(freq_lang,inplace=True)
# In[82]:
 # 2. Handle missing values in IMDB score col by replacing it with median of the IMDB col
imdb_median = netflix_df['IMDb Score'].median()
 netflix_df['IMDb Score'].fillna(imdb_median,inplace=True)
# In[83]:
 # 3. Handling missing values in Release Date col by replacing with the value 'Not
Available'
netflix_df['Release Date'].fillna('23 Sep 1677',inplace=True)
# In[84]:
 # 4. Handling missing values in Genre col by replacing with the value 'Not Available'
netflix_df['Genre'].fillna('Not Available',inplace=True)
# In[85]:
 # 5. Handling missing values in Country Availability by dropping the rows using dropna()
netflix_df.dropna(subset=['Country Availability'],inplace=True)
# In[86]:
 # 6. Handling missing values in Runtime by dropping the rows using dropna()
netflix_df.dropna(subset=['Runtime'],inplace=True)
# In[87]:
 # 7. Handling missing values in View Rating by replacing with the value 'Not Rated'
netflix_df['View Rating'].fillna('Not Rated',inplace=True)
# In[88]:
netflix_df.isnull().sum()
# ## 5. Checking for Duplicate Values
# In[89]:
duplicate_val = netflix_df.duplicated()
# In[90]:
netflix_df[duplicate_val]
# In[91]:
netflix_df.drop_duplicates(keep='first',inplace=True)
# In[92]:
netflix_df[duplicate_val]
# In[93]:
netflix_df
# ## Formatting the Date columns - Release Date and Netflix Release Date
 # In[94]:
netflix_df['Netflix Release Date'] = pd.to_datetime(netflix_df['Netflix Release Date'])
# In[95]:
netflix_df.dtypes
# In[96]:
netflix_df=netflix_df.reset_index()
# In[97]:
netflix_df['Release Date']
# In[98]:
from datetime import datetime
```

```
def date_format(inp):
    inp = datetime.strptime(inp,"%d %b %Y")
    inp = str(inp).split(' ')[0]
    return inp
# In[99]:
for i in range(0,len(netflix_df['Release Date'])):
    netflix_df['Release Date'][i]=date_format(netflix_df['Release Date'][i])
# In[100]:
netflix_df['Release Date'].value_counts()
# In[101]:
netflix_df['Release Date'] = pd.to_datetime(netflix_df['Release Date'])
# In[102]:
netflix_df.dtypes
# In[103]:
netflix_df['Movie Released Year'] = ' '
# In[104]:
for i in range(0,len(netflix_df['Release Date'])):
    netflix_df['Movie Released Year'][i] = netflix_df['Release Date'][i].year
# ### Formatting the values of Runtime Column
 # In[105]:
netflix_df['Runtime'].value_counts()
# In[106]:
netflix_df['Runtime'].replace({'1-2 hour':120,'< 30 minutes':30,'> 2 hrs':180,'30-60
mins':60},inplace=True)
# In[107]:
netflix_df['Runtime'].value_counts()
# ## Formatting the values of View Rating Column
 # In[108]:
netflix_df['View Rating'].replace({'NOT RATED':'Not
Rated','UNRATED':'Unrated'},inplace=True)
# In[109]:
netflix_df['View Rating'].value_counts()
# ### Creating OTT column
 # In[111]:
netflix_df['OTT'] = 'Netflix'
# ## Saving the Netflix Cleaned File
 # In[112]:
netflix_df.to_csv('netflix cleaned data.csv')
# ## 2. Disney Plus Data
 # In[150]:
hotstar_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\New
Dataset\disney_plus_shows.csv")
 hotstar_df.head()
# In[151]:
hotstar_df.columns
# In[152]:
hotstar_df.info()
# In[153]:
```

```
hotstar_df.describe()

 # In[154]:
unwanted_cols = ['imdb_id', 'plot','director', 'writer', 'actors',
       'awards', 'metascore',
       'imdb_votes']
 hotstar_df.drop(columns=unwanted_cols,inplace=True)
# In[155]:
hotstar_df.head()
# ### 3. Handling Missing Values
 # In[156]:
hotstar_df.isnull().sum()
# In[157]:
 ## 1. Hamdling missing values in title column by removing the rows using dropna
hotstar_df.dropna(subset=['title'],inplace=True)
# In[158]:
hotstar_df.isnull().sum()
# In[159]:
 # 2. Handling missing values in Genre col by replacing with the value 'Not Available'
hotstar_df['genre'].fillna('Not Available',inplace=True)
# In[160]:
 # 3. Handle missing values in Languages column by replacing with most freq value
freq_lang = hotstar_df.language.value_counts().index[0]
 hotstar_df['language'].fillna(freq_lang,inplace=True)
# In[161]:
 # 4. Handle missing values in IMDB rating col by replacing it with median of the IMDB col
imdb_rat_median = hotstar_df['imdb_rating'].median()
 hotstar_df['imdb_rating'].fillna(imdb_rat_median,inplace=True)
# In[162]:
hotstar_df.isnull().sum()
# In[163]:
 # 5. Handling missing values in country column by replacing it with most freq col
country_freq = hotstar_df['country'].value_counts().index[0]
 hotstar_df.country.fillna(country_freq,inplace=True)
# In[164]:
 # 6. Handle missing values in runtime col by replacing it with median of the runtime col
hotstar_df['runtime']=hotstar_df['runtime'].str.replace('min','')
# In[165]:
hotstar_df['runtime']=hotstar_df['runtime'].str.replace('h','')
# In[166]:
 # convert runtime col from string to numeric datatype
hotstar_df['runtime']=pd.to_numeric(hotstar_df['runtime'])
 hotstar_df['runtime'].dtype
# In[167]:
runtime_median = hotstar_df['runtime'].median()
 hotstar_df['runtime'].fillna(runtime_median,inplace=True)
# In[168]:
hotstar_df.isnull().sum()
```

```
# In[169]:
 # 7. Handling missing values rated col by replacing with the value 'Not Rated'
hotstar_df['rated'].fillna('Not Rated',inplace=True)
# In[170]:
hotstar_df.isnull().sum()
# In[171]:
 # 8. Handling missing values in release date col by replacing with the date '01 Jan 1111'
hotstar_df['released_at'].fillna('23 Sep 1677',inplace=True)
# In[172]:
hotstar_df.isnull().sum()
# ## 5. Checking for Duplicate Values

 # In[173]:
duplicate_values = hotstar_df.duplicated()
 hotstar_df[duplicate_values]
# ## Formatting the Date columns - Release Date and Hotstar Release Date
 # In[174]:
hotstar_df.dtypes
# In[175]:
hotstar_df=hotstar_df.reset_index()
# In[176]:
from datetime import datetime
def date_format(inp):
    inp = datetime.strptime(inp,"%d %b %Y")
    inp = str(inp).split(' ')[0]
    return inp
# In[177]:
 # formatting released at column
for i in range(0,len(hotstar_df['released_at'])):
    hotstar_df['released_at'][i]=date_format(hotstar_df['released_at'][i])
# In[178]:
def date_format_2(inp):
    inp = datetime.strptime(inp,"%B %d, %Y")
    inp = str(inp).split(' ')[0]
    return inp
# In[179]:
for i in range(0,len(hotstar_df['added_at'])):
    hotstar_df['added_at'][i]=date_format_2(hotstar_df['added_at'][i])
# In[180]:
hotstar_df['released_at'] = pd.to_datetime(hotstar_df['released_at'])
# In[181]:
hotstar_df['added_at'] = pd.to_datetime(hotstar_df['added_at']
# In[182]:
hotstar_df.dtypes
# # Creating movie release year column from released_at column.
 # In[206]:
hotstar_df['Movie Released Year'] = ' '
# In[207]:
```

```
for i in range(0,len(hotstar_df['released_at'])):
    hotstar_df['Movie Released Year'][i] = hotstar_df['released_at'][i].year
# In[208]:
hotstar_df
# ## Formatting the values of Rated Column
 # In[199]:
hotstar_df['rated'].replace({'NOT RATED':'Not
Rated','UNRATED':'Unrated','PASSED':'Passed','APPROVED':'Approved'},inplace=True)
# In[200]:
hotstar_df['rated'].value_counts()
# ### Drop Year Column from the hotstar table
 # In[204]:
hotstar_df.drop(columns=['year'],inplace=True)
# In[209]:
hotstar_df
# ## Saving the Hotstar Cleaned File
 # In[210]:
hotstar_df.to_csv('hotstar cleaned data.csv')
# ## Amazon Prime Data
 # In[2]:
prime_mov_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\New
Dataset\OTT_final_data.csv")
 prime_mov_df
# In[3]:
prime_mov_df.OTT.value_counts()
# In[4]:
prime_mov_df2 = prime_mov_df[prime_mov_df['OTT']=='Prime Video'].copy()
 prime_mov_df2.head()
# In[8]:
prime_mov_df2.drop(columns=['Unnamed: 0'],inplace=True)
# In[10]:
prime_mov_df2.head()
# In[11]:
prime_mov_df2 = prime_mov_df2.reset_index()
# In[14]:
prime_mov_df2.drop(columns=['index'],inplace=True)
# In[15]:
prime_mov_df2.head()
# ## Add Type column to the table
 # In[16]:
prime_mov_df2['Type'] = 'Movie'
# In[17]:
prime_mov_df2.head()
# ## Format the values of the Age column
 # In[18]:
prime_mov_df2.Age.value_counts()
# In[28]:
prime_mov_df2['Age'].replace({'18+':'NC-17','7+':'PG','13+':'PG-13','all':'G','16+':'R'},i
```

```
nplace=True)
# In[29]:
prime_mov_df2.Age.value_counts()
# In[34]:
prime_mov_df2.to_csv('Prime cleaned data.csv')
# ## Amazon Prime Tv Shows Data
 # In[80]:
prime_tv_df = pd.read_csv(r'I:\IBM Hack Challenge - OTT Analysis\New Dataset\Prime TV
Shows Data set (1).csv',encoding="ISO-8859-1")
 prime_tv_df.head()
# In[81]:
prime_tv_df.info()
# In[82]:
prime_tv_df.describe()
# ## Data Cleaning - Amazon Prime Series Data
 # ### Removing Unwanted Columns
 # In[83]:
prime_tv_df.drop(columns=['S.no.','No of seasons available'],inplace=True)
# In[84]:
prime_tv_df.head()
# ### Handling Missing Values
 # In[85]:
 ## Handling missing values in IMDB rating column by replacing it with median of the IMDB
col
prime_tv_df['IMDb rating'].fillna(prime_tv_df['IMDb rating'].median(),inplace=True)
# In[86]:
prime_tv_df.isnull().sum()
# In[87]:
prime_tv_df.dropna(axis=0,inplace=True)
# In[88]:
prime_tv_df.isnull().sum()
# ### Format the values of Age of Viewers column
 # In[89]:
prime_tv_df['Age of viewers'].value_counts()
# In[90]:
prime_tv_df['Age of
viewers'].replace({'18+':'NC-17','7+':'PG','13+':'PG-13','All':'G','16+':'R'},inplace=Tru
e)
# In[91]:
prime_tv_df['Age of viewers'].value_counts()
# In[97]:
 ### Change the datatype of the year to int
prime_tv_df['Year of release']=prime_tv_df['Year of release'].astype('int')
# In[98]:
prime_tv_df
# In[102]:
prime_tv_df['Type']='Series'
# In[103]:
```

```
prime_tv_df['Language'].value_counts()
# In[104]:
prime_tv_df = prime_tv_df.reset_index()
# In[105]:
prime_tv_df['Country']=''
# In[106]:
for i in range(0,len(prime_tv_df)):
    if prime_tv_df['Language'][i]=='English':
        prime_tv_df['Country'][i] ='USA'
    elif prime_tv_df['Language'][i]=='Hindi' or prime_tv_df['Language'][i]=='Marathi' or
prime_tv_df['Language'][i]=='Telugu':
        prime_tv_df['Country'][i] ='India'
    elif prime_tv_df['Language'][i]=='Japanese':
        prime_tv_df['Country'][i] ='Japan'
    elif prime_tv_df['Language'][i]=='Italiano':
        prime_tv_df['Country'][i] ='Italy'
    elif prime_tv_df['Language'][i]=='Spanish':
        prime_tv_df['Country'][i] ='Spain'
    elif prime_tv_df['Language'][i]=='Deutsch':
        prime_tv_df['Country'][i] ='Germany'
    elif prime_tv_df['Language'][i]=='Nederlands':
        prime_tv_df['Country'][i] ='Netherland'
    elif prime_tv_df['Language'][i]=='Serbian':
        prime_tv_df['Country'][i] ='Serbia'
    elif prime_tv_df['Language'][i]=='French':
        prime_tv_df['Country'][i] ='France'
    elif prime_tv_df['Language'][i]=='Spanish':
        prime_tv_df['Country'][i] ='Spain'
    elif prime_tv_df['Language'][i]=='Hebrew':
        prime_tv_df['Country'][i] ='Israel'
    elif prime_tv_df['Language'][i]=='Suomi':
        prime_tv_df['Country'][i] ='Finland'
    elif prime_tv_df['Language'][i]=='Russian':
        prime_tv_df['Country'][i] ='Russia'
# In[108]:
prime_tv_df['Country'].value_counts()
# In[109]:
prime_tv_df.to_csv('prime tv cleaned data.csv')
# ## 4.HULU Movies Data
 # In[2]:
hulu_mov_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\New
Dataset\OTT_final_data.csv")
 hulu_mov_df
# In[3]:
hulu_mov_df2 = hulu_mov_df[hulu_mov_df.OTT=='Hulu'].copy()
# In[4]:
hulu_mov_df2
# In[5]:
```

```
hulu_mov_df2=hulu_mov_df2.reset_index()
# In[6]:
hulu_mov_df2.drop(columns=['Unnamed: 0'],inplace=True)
# In[11]:
hulu_mov_df2['Type'] = 'Movie'
# In[8]:
hulu_mov_df2['Age'].replace({'18+':'NC-17','7+':'PG','13+':'PG-13','all':'G','16+':'R'},in
place=True)
# In[9]:
hulu_mov_df2['Age'].value_counts()
# In[10]:
hulu_mov_df2.dtypes
# In[12]:
hulu_mov_df2
# In[13]:
hulu_mov_df2.drop(columns=['index'],inplace=True)
# In[15]:
hulu_mov_df2.to_csv('Hulu movies cleaned data.csv')
# ## Hulu Series Dataset
 # In[35]:
hulu_series_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\tv_shows.csv")
 hulu_series_df.head()
# In[36]:
hulu_series_df2=hulu_series_df[hulu_series_df['Hulu']==1]
# In[37]:
hulu_series_df2.info()
# In[38]:
len(hulu_series_df)
# In[39]:
len(hulu_series_df2)
# ## Data Cleaning - Hulu Series Data
 # ### 1. Removing Unwanted Columns
 # In[40]:
hulu_series_df2.columns
# In[41]:
hulu_series_df2.drop(columns=['Unnamed: 0', 'ID','Rotten Tomatoes',
       'Netflix', 'Hulu', 'Prime Video', 'Disney+', 'Type'],inplace=True)
# In[42]:
hulu_series_df2.head()
# In[43]:
hulu_series_df2 = hulu_series_df2.reset_index()
# ### 2. Handling Missing Values
 # In[44]:
hulu_series_df2.isnull().sum()
# In[45]:
 ## 1. Handling missing values in Age column
age_freq_value = hulu_series_df2.Age.value_counts().index[0]
 hulu_series_df2.Age.fillna(age_freq_value,inplace=True)
```

```
# In[46]:
hulu_series_df2.isnull().sum()
# In[61]:
 ## 2. Handling Missing values in IMDB column
hulu_series_df2.IMDb = hulu_series_df2.IMDb.str.replace('/10','')
 hulu_imdb_median = hulu_series_df2.IMDb.median()
 hulu_series_df2.IMDb.fillna(hulu_imdb_median,inplace=True)
# In[62]:
hulu_series_df2.IMDb.unique()
# In[63]:
hulu_series_df2.isnull().sum()
# ### Format the Age column
 # In[64]:
hulu_series_df2.Age.value_counts()
# In[66]:
hulu_series_df2['Age'].replace({'18+':'NC-17','7+':'PG','13+':'PG-13','all':'G','16+':'R'}
,inplace=True)
# In[67]:
hulu_series_df2.Age.value_counts()
# In[68]:
hulu_series_df2.head()
# In[69]:
hulu_series_df2['Type'] = 'Series'
hulu_series_df2['OTT'] = 'Hulu'
hulu_series_df2['Runtime'] = 60
hulu_series_df2['Country'] = 'Not Available'
hulu_series_df2['Language'] = 'Not Available'
hulu_series_df2['Genre'] = 'Not Available'
# In[70]:
hulu_series_df2.head()
# In[71]:
hulu_series_df2.drop(columns=['index'],inplace=True)
# In[72]:
hulu_series_df2.head()
# In[73]:
hulu_series_df2.to_csv('hulu series cleaned file.csv')
# ## Read all the cleaned datasets
 # In[113]:
netflix_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\Cleaned Dataset\netflix
cleaned data.csv")
 disney_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\Cleaned Dataset\hotstar
cleaned data.csv")
 prime_movie_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\Cleaned Dataset\Prime
cleaned data.csv")
 prime_series_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\Cleaned
Dataset\prime tv cleaned data.csv")
 hulu_movie_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\Cleaned Dataset\Hulu
movies cleaned data.csv")
```

```
 hulu_series_df = pd.read_csv(r"I:\IBM Hack Challenge - OTT Analysis\Cleaned Dataset\hulu
series cleaned file.csv")
# In[115]:
netflix_df.head()
# ## Combining all datasets into single file
 # In[116]:
netflix_df.columns
# In[132]:
netflix_df.columns = ['Unnamed: 0', 'Title', 'Year', 'IMDB Rating', 'Age Restriction',
       'Genre', 'Language', 'Runtime', 'Country', 'Type', 'Release Date',
       'Netflix Release Date', 'OTT']
# In[133]:
netflix_2_df = netflix_df[['Unnamed: 0', 'Title', 'Year', 'IMDB Rating', 'Age Restriction',
       'Genre', 'Language', 'Runtime', 'Country', 'Type', 'OTT']].copy()
 netflix_2_df.head()
# In[118]:
disney_df.columns
# In[119]:
disney2_df = disney_df[['Unnamed: 0', 'Title', 'Year', 'IMDB Rating', 'Age Restriction',
       'Genre', 'Language', 'Runtime', 'Country', 'Type','OTT']].copy()
# In[134]:
OTT_final_data_1 =
pd.concat([netflix_2_df,disney2_df,prime_movie_df,prime_series_df,hulu_movie_df,hulu_series
_df])
# In[135]:
OTT_final_data_1.head()
# In[138]:
len(OTT_final_data_1)
# In[142]:
netflix_df.columns = ['Unnamed: 0', 'Title', 'Year', 'IMDB Rating', 'Age Restriction',
       'Genre', 'Language', 'Runtime', 'Country', 'Type', 'Release Date',
       'OTT Release Date', 'OTT']
# In[141]:
disney_df.columns
# In[143]:
OTT_final_data_2 = pd.concat([netflix_df,disney_df])
 OTT_final_data_2.head()
# In[144]:
 ### Saving the cleaned file
OTT_final_data_1.to_csv('OTT_final_data_1.csv')
 OTT_final_data_2.to_csv('OTT_final_data_2.csv')
```