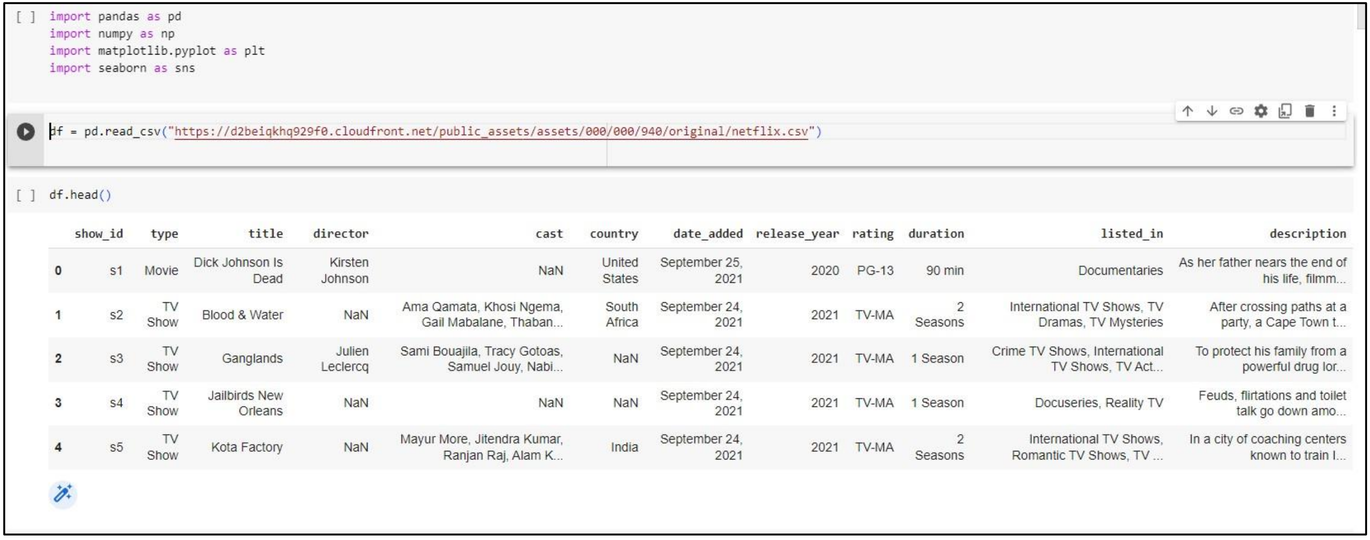
Name: Raghu

Business case: Netflix

Date: 26/May/2023

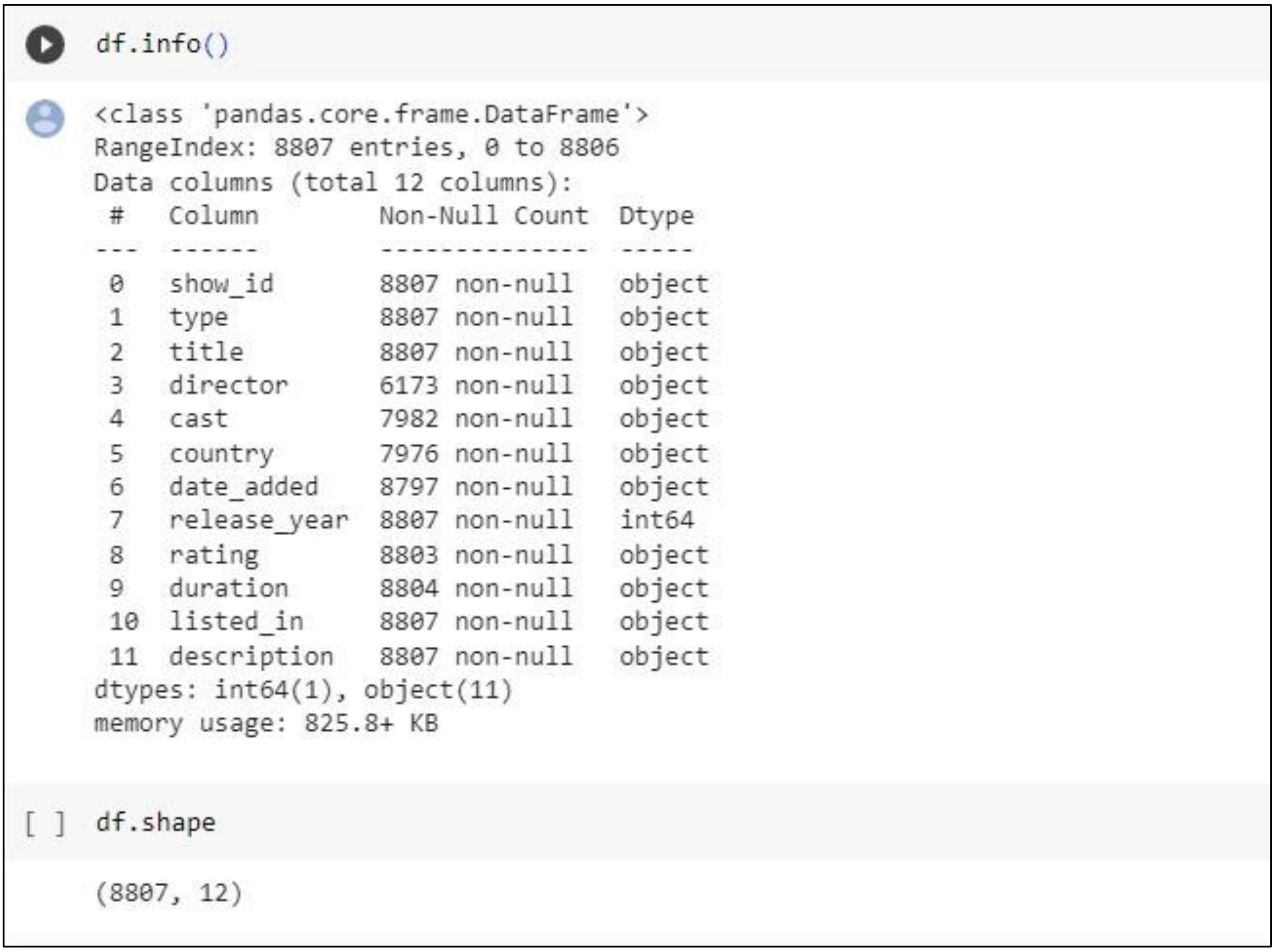


```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
```

```
df = pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv")
```

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

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...



```
df.info()
```

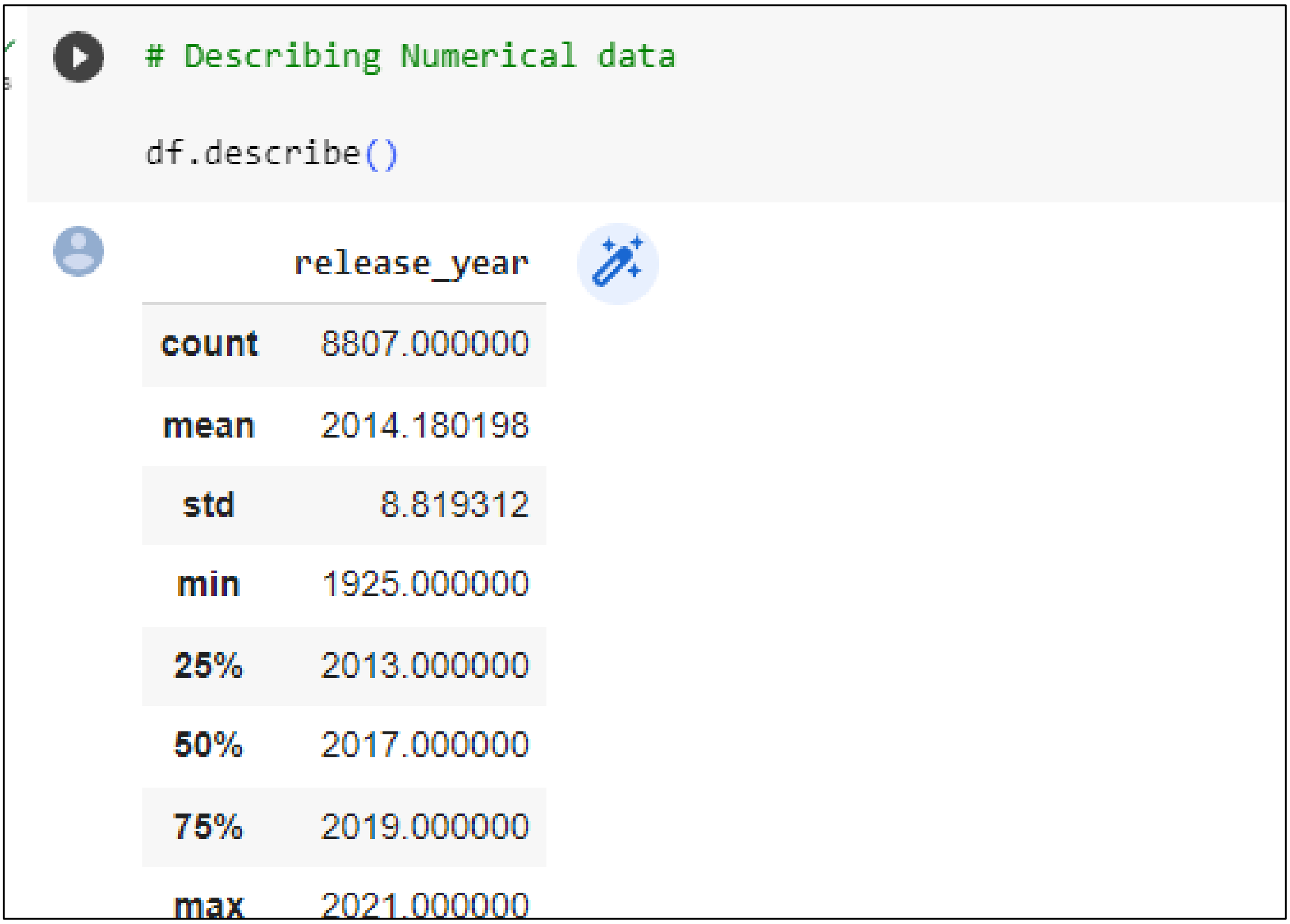
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64 
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

```
[ ] df.shape
```

```
(8807, 12)
```

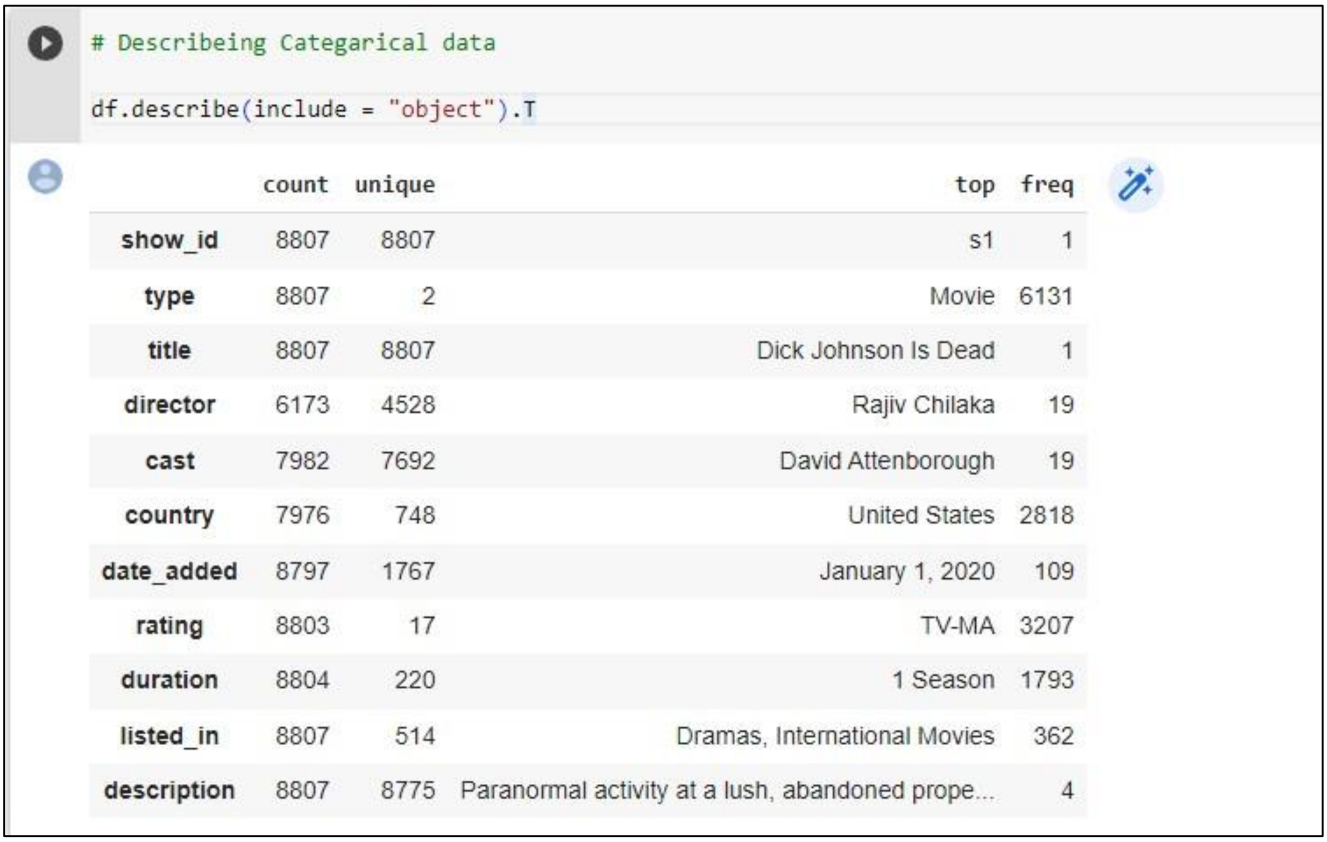
Name: Raghu
Business case: Netflix

Date: 26/May/2023



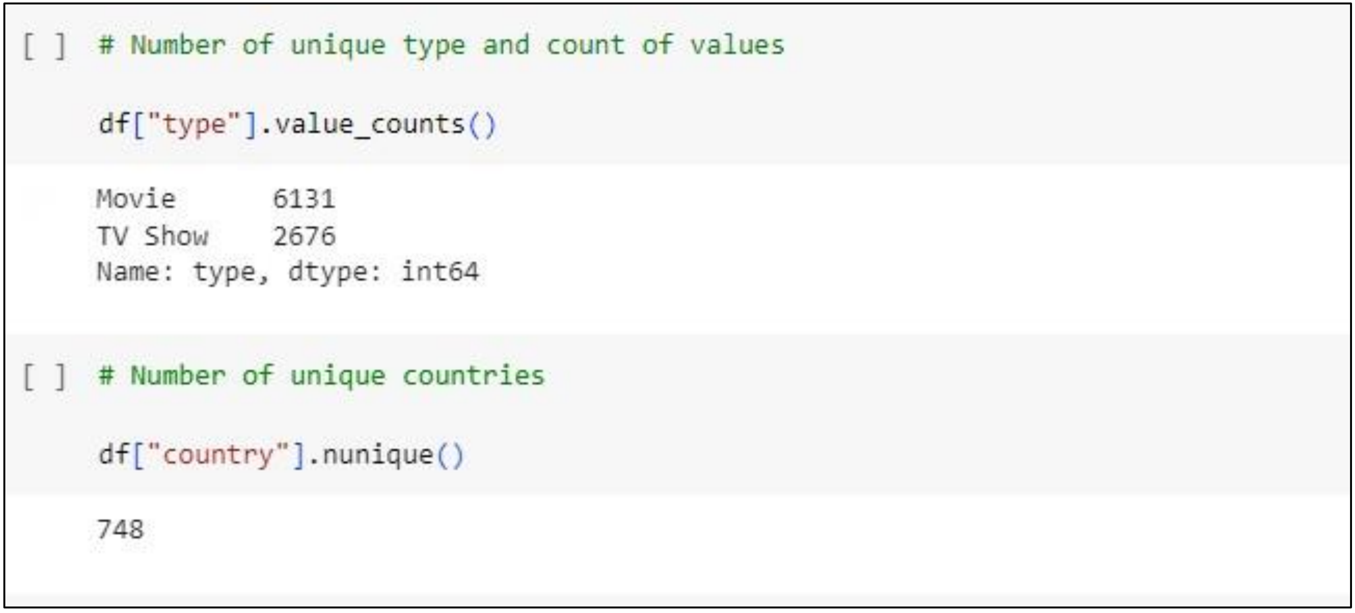
```
# Describing Numerical data

df.describe()
```

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000



```
# Describeing Categarical data

df.describe(include = "object").T
```

	count	unique	top	freq
show_id	8807	8807	s1	1
type	8807	2	Movie	6131
title	8807	8807	Dick Johnson Is Dead	1
director	6173	4528	Rajiv Chilaka	19
cast	7982	7692	David Attenborough	19
country	7976	748	United States	2818
date_added	8797	1767	January 1, 2020	109
rating	8803	17	TV-MA	3207
duration	8804	220	1 Season	1793
listed_in	8807	514	Dramas, International Movies	362
description	8807	8775	Paranormal activity at a lush, abandoned prope...	4

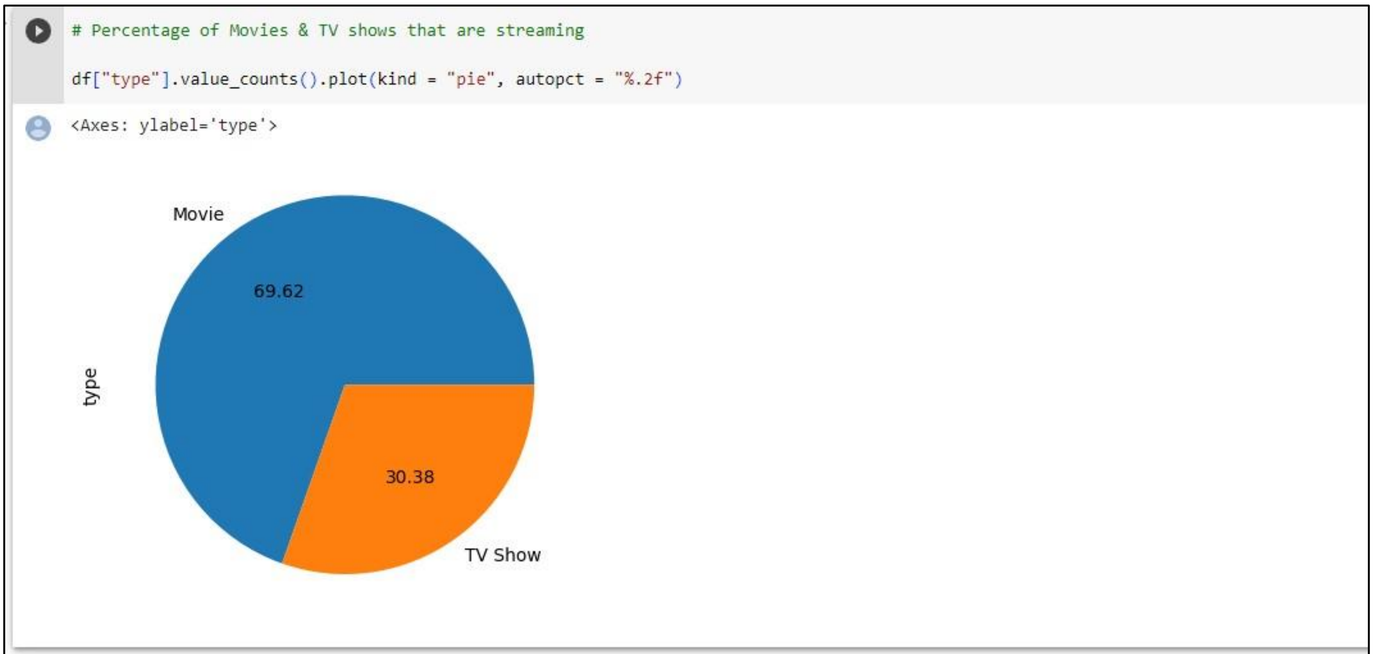


```
[ ]  # Number of unique type and count of values

     df["type"].value_counts()
```

```
Movie      6131
TV Show    2676
Name: type, dtype: int64
```

```
[ ]  # Number of unique countries

     df["country"].nunique()
```

```
748
```

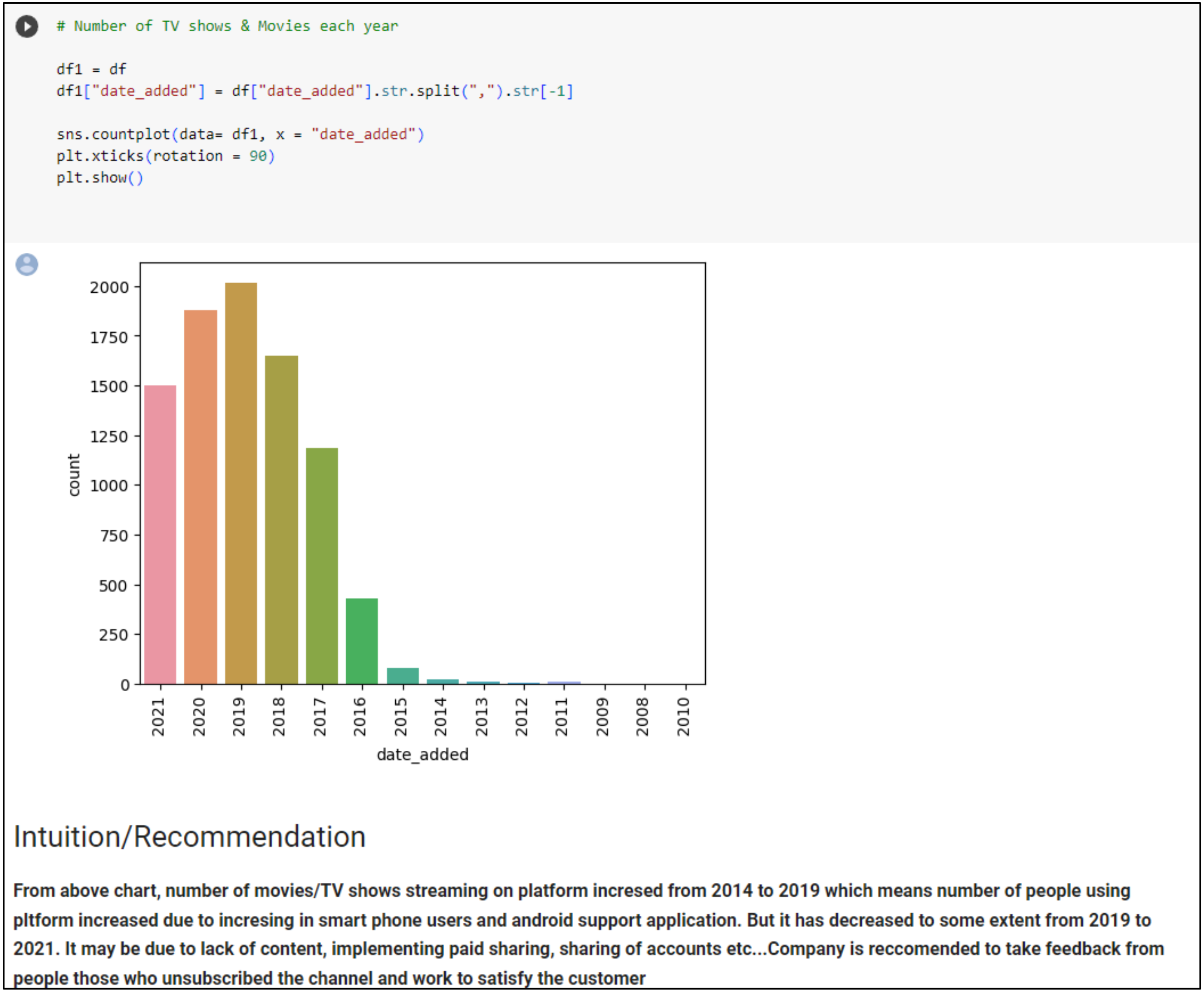


```
# Percentage of Movies & TV shows that are streaming

df["type"].value_counts().plot(kind = "pie", autopct = "%.2f")
```

```
<Axes: ylabel='type'>
```

Intuition/Recommendation:

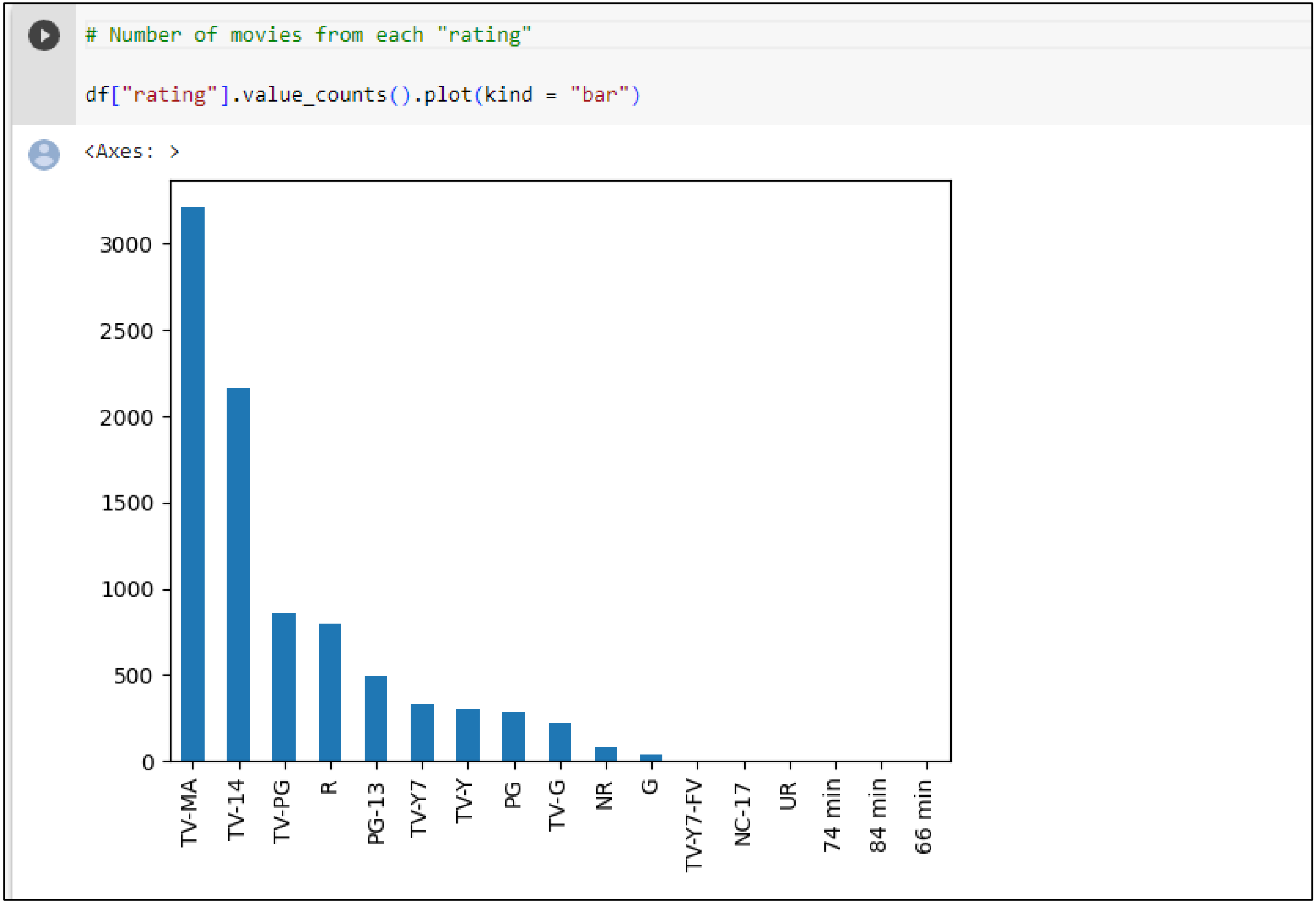
From the above pie chart, 69.62 % motion pictures are movies & 30.38 % of motion picturea are TV shows. So people more prefererable to watch movies compared to TV shows



```
# Number of TV shows & Movies each year

df1 = df
df1["date_added"] = df["date_added"].str.split(",").str[-1]

sns.countplot(data= df1, x = "date_added")
plt.xticks(rotation = 90)
plt.show()
```

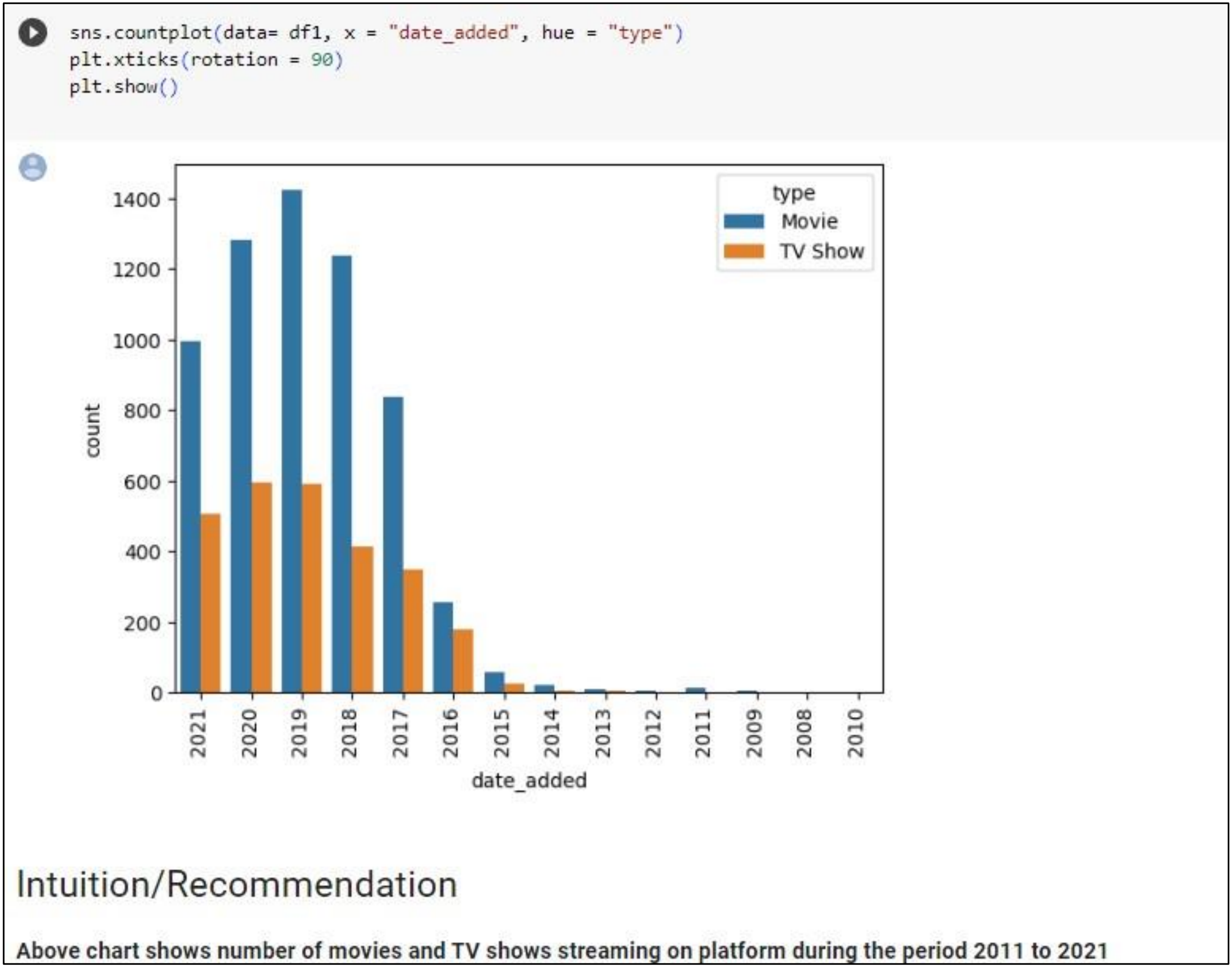
Intuition/Recommendation

From above chart, number of movies/TV shows streaming on platform incresed from 2014 to 2019 which means number of people using pltform increased due to incresing in smart phone users and android support application. But it has decreased to some extent from 2019 to 2021. It may be due to lack of content, implementing paid sharing, sharing of accounts etc...Company is reccomended to take feedback from people those who unsubscribed the channel and work to satisfy the customer



```
# Number of movies from each "rating"

df["rating"].value_counts().plot(kind = "bar")
```

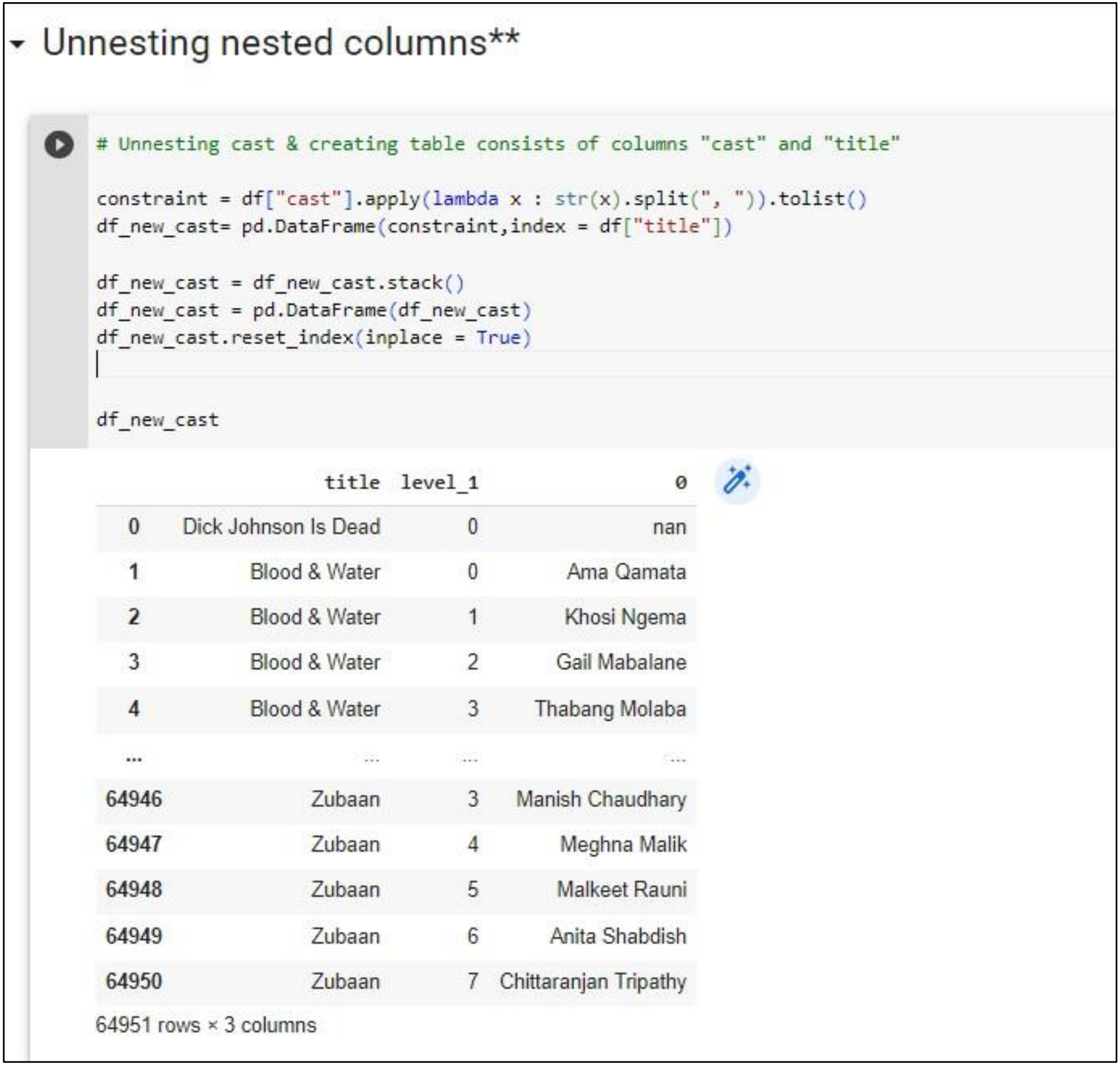
```
<Axes: >
```



```
sns.countplot(data= df1, x = "date_added", hue = "type")
plt.xticks(rotation = 90)
plt.show()
```

Intuition/Recommendation

Above chart shows number of movies and TV shows streaming on platform during the period 2011 to 2021

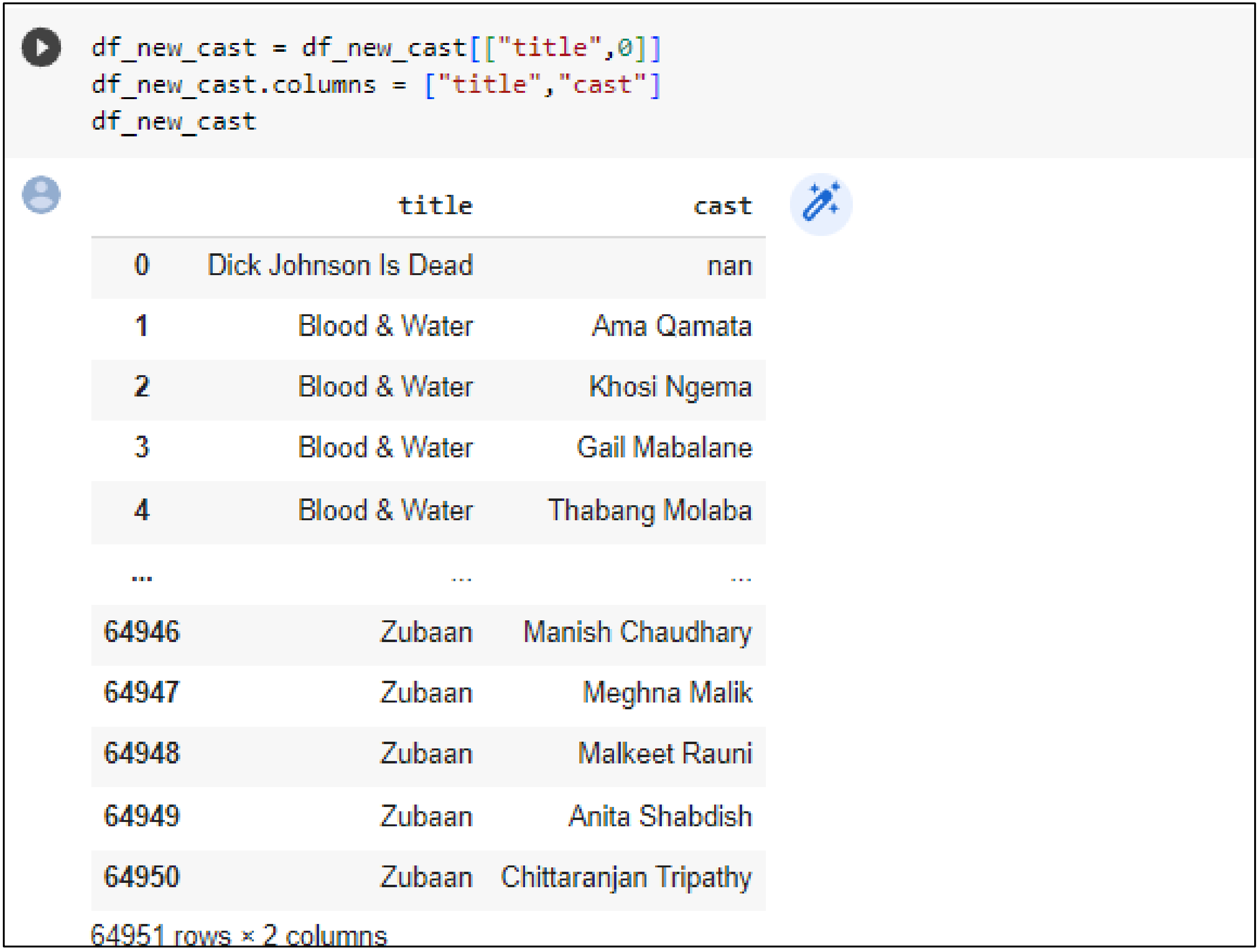


Unnesting nested columns**

```
# Unnesting cast & creating table consists of columns "cast" and "title"

constraint = df["cast"].apply(lambda x : str(x).split(", ")).tolist()
df_new_cast= pd.DataFrame(constraint,index = df["title"])

df_new_cast = df_new_cast.stack()
df_new_cast = pd.DataFrame(df_new_cast)
df_new_cast.reset_index(inplace = True)


df_new_cast
```

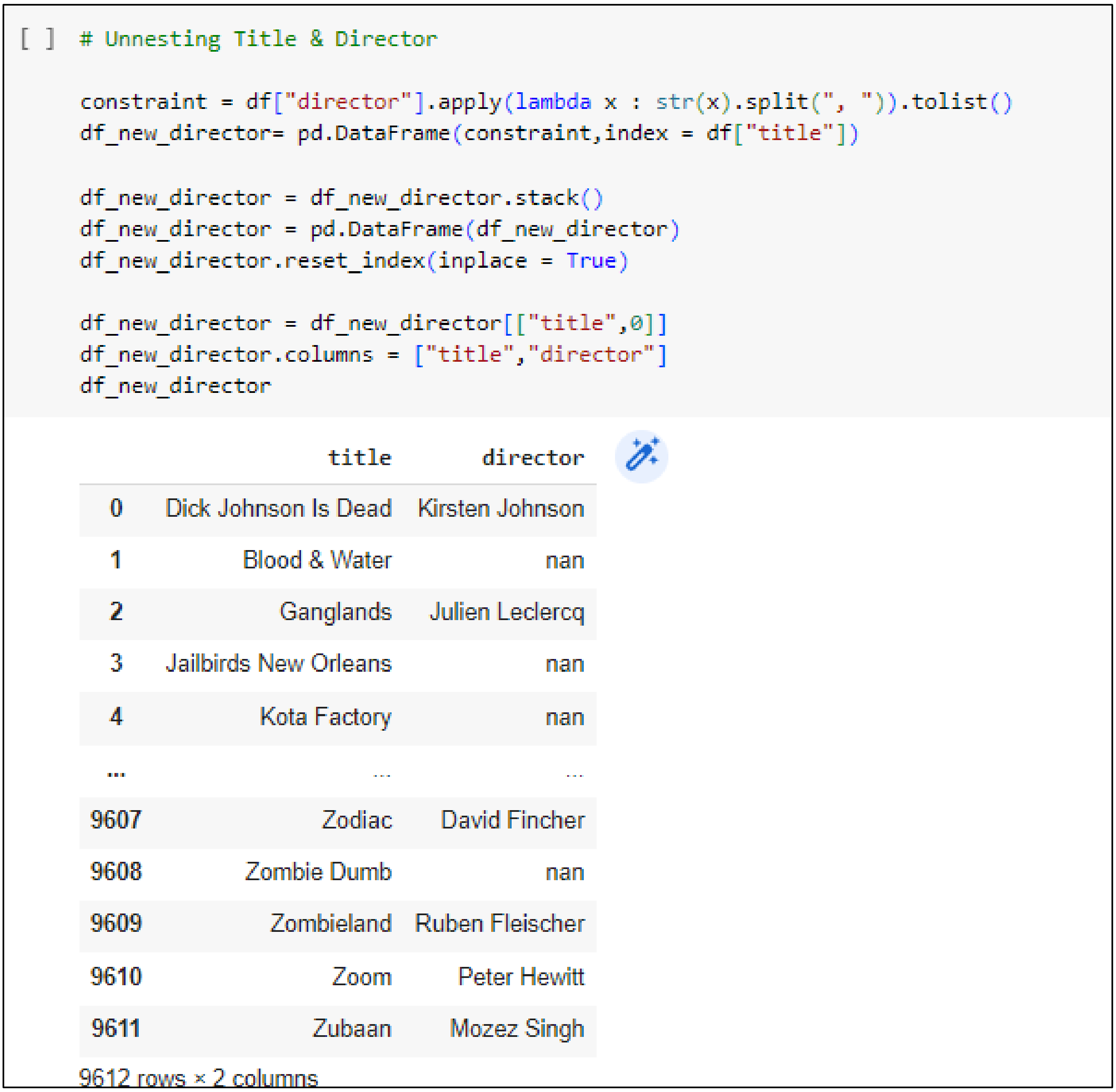
	title	level_1	0
0	Dick Johnson Is Dead	0	nan
1	Blood & Water	0	Ama Qamata
2	Blood & Water	1	Khosi Ngema
3	Blood & Water	2	Gail Mabalane
4	Blood & Water	3	Thabang Molaba
...	...	...	...
64946	Zubaan	3	Manish Chaudhary
64947	Zubaan	4	Meghna Malik
64948	Zubaan	5	Malkeet Rauni
64949	Zubaan	6	Anita Shabdish
64950	Zubaan	7	Chittaranjan Tripathy

64951 rows × 3 columns



```
df_new_cast = df_new_cast[["title",0]]
df_new_cast.columns = ["title","cast"]
df_new_cast
```

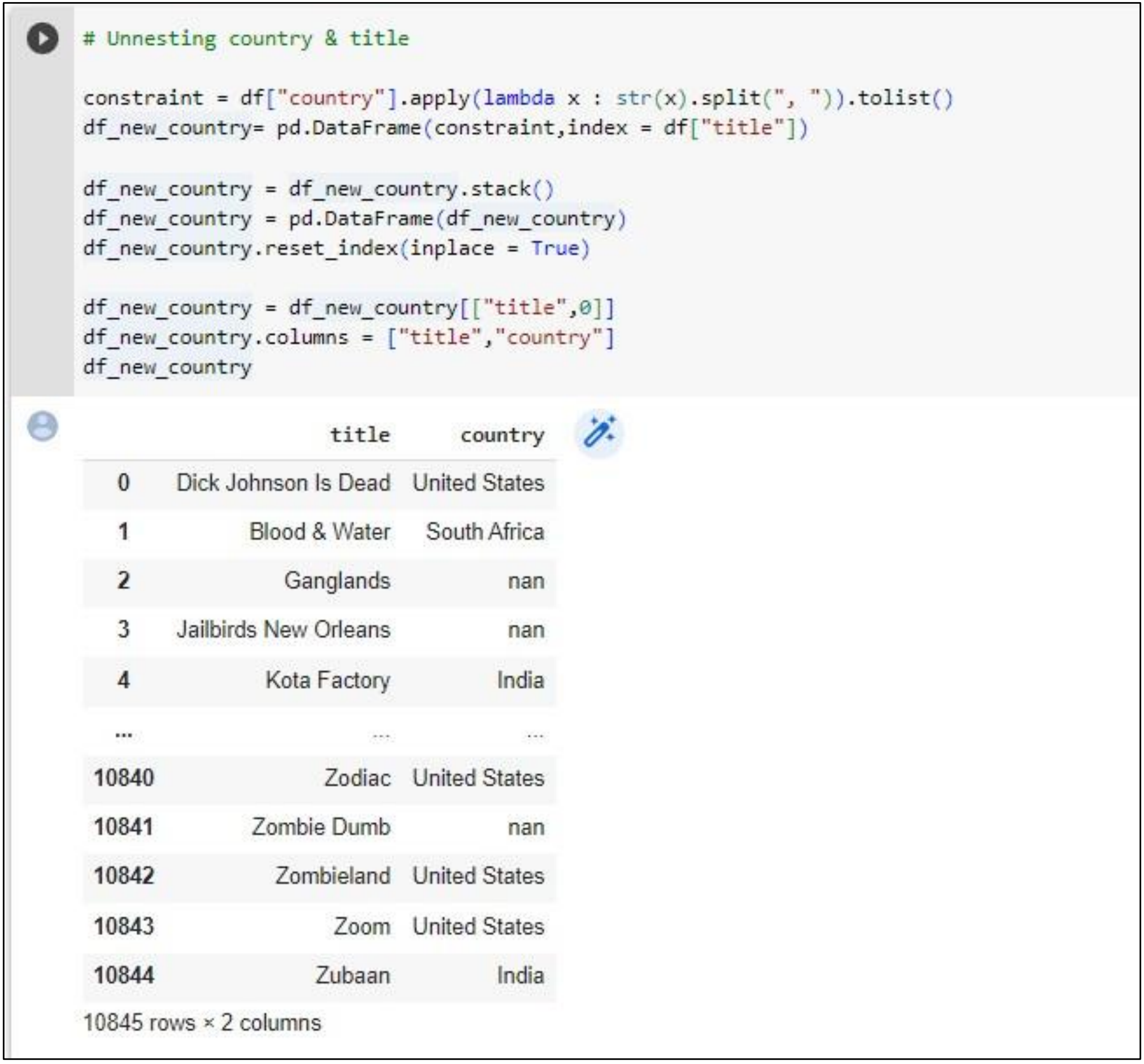
```
                       title                    cast
0         Dick Johnson Is Dead                   nan
1                Blood & Water            Ama Qamata
2                Blood & Water           Khosi Ngema
3                Blood & Water        Gail Mabalane
4                Blood & Water        Thabang Molaba
...                      ...                     ...
64946                 Zubaan     Manish Chaudhary
64947                 Zubaan          Meghna Malik
64948                 Zubaan         Malkeet Rauni
64949                 Zubaan        Anita Shabdish
64950                 Zubaan  Chittaranjan Tripathy
64951 rows × 2 columns
```



```
[ ] # Unnesting Title & Director

    constraint = df["director"].apply(lambda x : str(x).split(", ")).tolist()
    df_new_director= pd.DataFrame(constraint,index = df["title"])

    df_new_director = df_new_director.stack()
    df_new_director = pd.DataFrame(df_new_director)
    df_new_director.reset_index(inplace = True)

    df_new_director = df_new_director[["title",0]]
    df_new_director.columns = ["title","director"]
    df_new_director
```

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	nan
4	Kota Factory	nan
...	...	...
9607	Zodiac	David Fincher
9608	Zombie Dumb	nan
9609	Zombieland	Ruben Fleischer
9610	Zoom	Peter Hewitt
9611	Zubaan	Mozez Singh

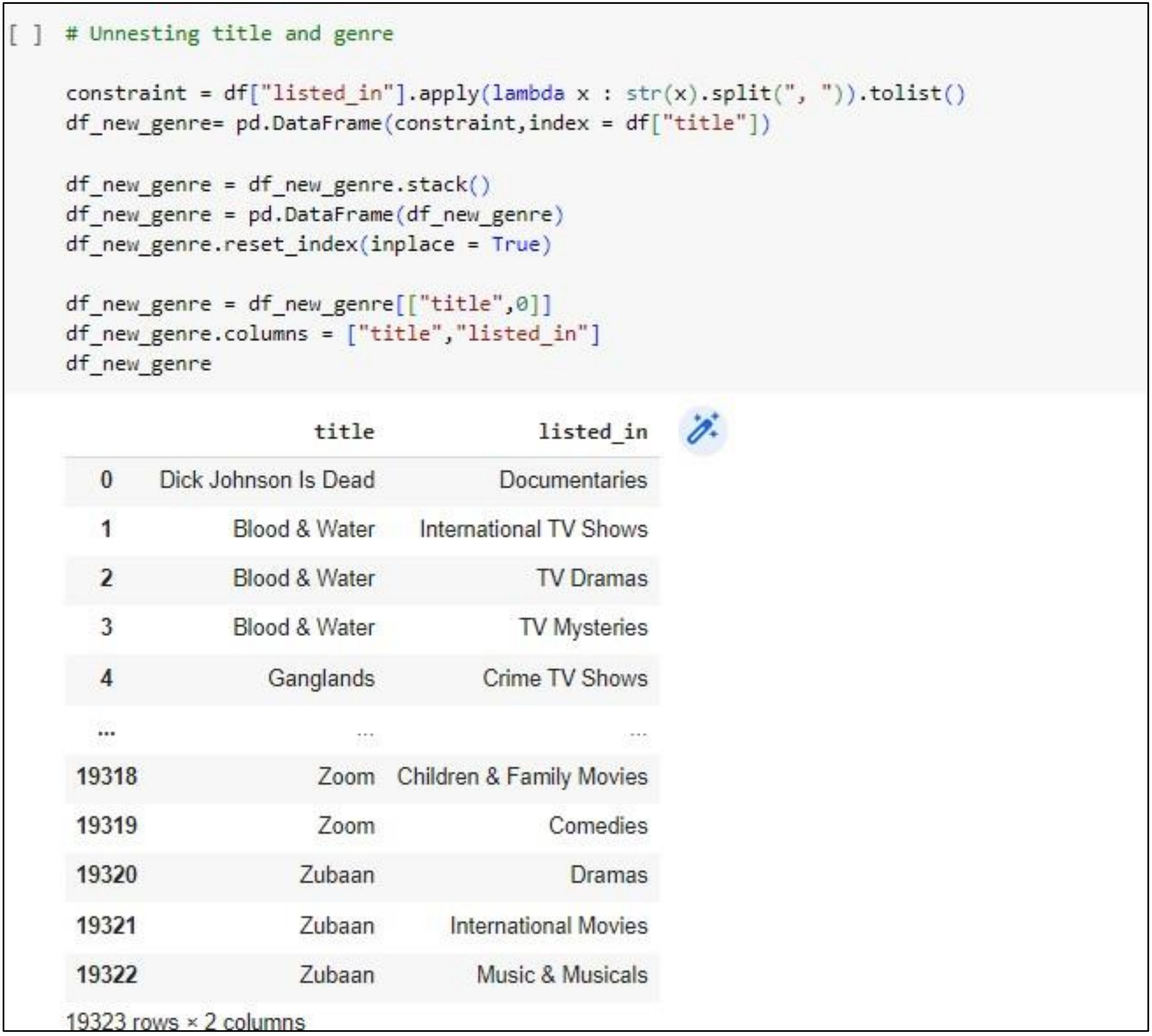
9612 rows × 2 columns



```
# Unnesting country & title

constraint = df["country"].apply(lambda x : str(x).split(", ")).tolist()
df_new_country= pd.DataFrame(constraint,index = df["title"])

df_new_country = df_new_country.stack()
df_new_country = pd.DataFrame(df_new_country)
df_new_country.reset_index(inplace = True)

df_new_country = df_new_country[["title",0]]
df_new_country.columns = ["title","country"]
df_new_country
```

	title	country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	nan
3	Jailbirds New Orleans	nan
4	Kota Factory	India
...	...	...
10840	Zodiac	United States
10841	Zombie Dumb	nan
10842	Zombieland	United States
10843	Zoom	United States
10844	Zubaan	India

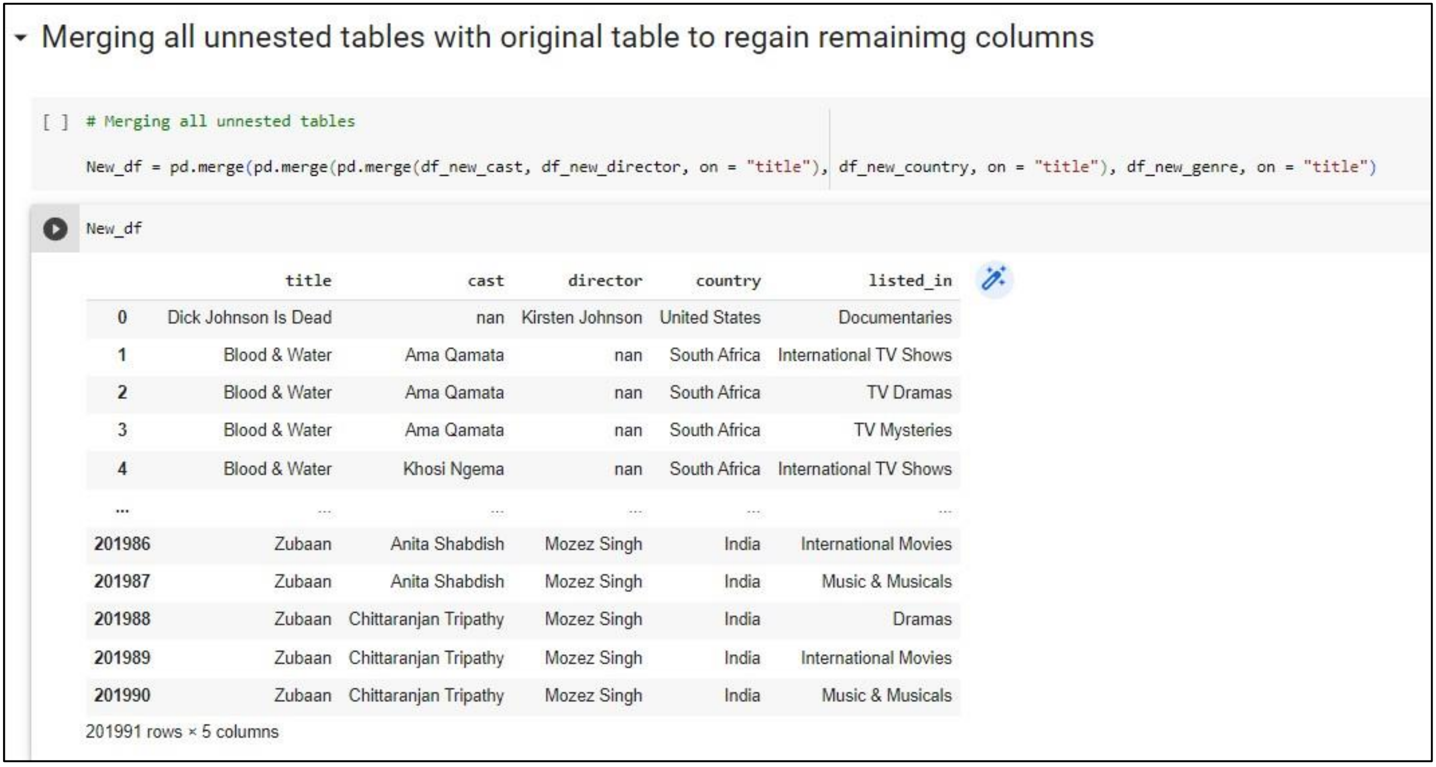
10845 rows × 2 columns



```
# Unnesting title and genre

constraint = df["listed_in"].apply(lambda x : str(x).split(", ")).tolist()
df_new_genre= pd.DataFrame(constraint,index = df["title"])

df_new_genre = df_new_genre.stack()
df_new_genre = pd.DataFrame(df_new_genre)
df_new_genre.reset_index(inplace = True)

df_new_genre = df_new_genre[["title",0]]
df_new_genre.columns = ["title","listed_in"]
df_new_genre
```

	title	listed_in
0	Dick Johnson Is Dead	Documentaries
1	Blood & Water	International TV Shows
2	Blood & Water	TV Dramas
3	Blood & Water	TV Mysteries
4	Ganglands	Crime TV Shows
...	...	...
19318	Zoom	Children & Family Movies
19319	Zoom	Comedies
19320	Zubaan	Dramas
19321	Zubaan	International Movies
19322	Zubaan	Music & Musicals

19323 rows × 2 columns



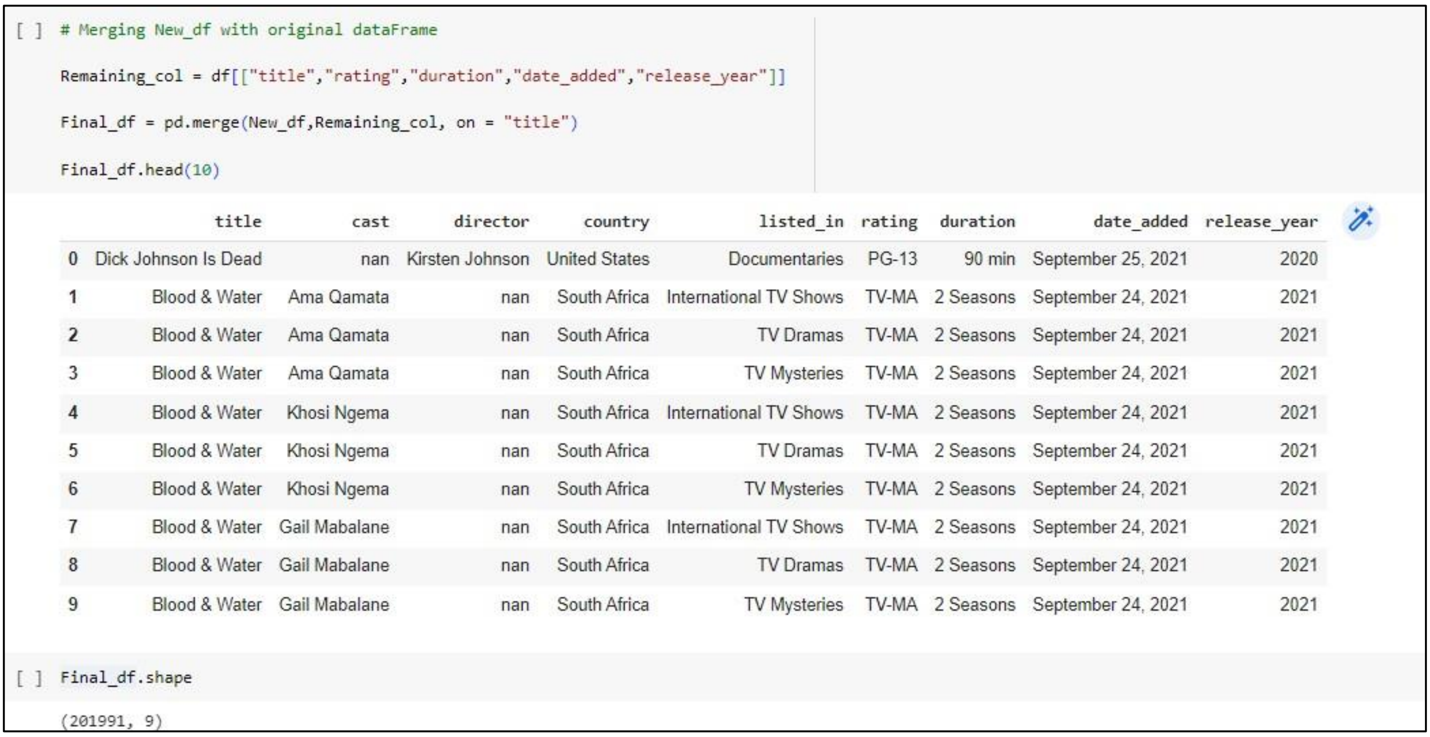
Merging all unnested tables with original table to regain remainimg columns

```
[ ] # Merging all unnested tables

    New_df = pd.merge(pd.merge(pd.merge(df_new_cast, df_new_director, on = "title"), df_new_country, on = "title"), df_new_genre, on = "title")
```

```
New_df
```

	title	cast	director	country	listed_in
0	Dick Johnson Is Dead	nan	Kirsten Johnson	United States	Documentaries
1	Blood & Water	Ama Qamata	nan	South Africa	International TV Shows
2	Blood & Water	Ama Qamata	nan	South Africa	TV Dramas
3	Blood & Water	Ama Qamata	nan	South Africa	TV Mysteries
4	Blood & Water	Khosi Ngema	nan	South Africa	International TV Shows
...	...	...	...	...	...
201986	Zubaan	Anita Shabdish	Mozez Singh	India	International Movies
201987	Zubaan	Anita Shabdish	Mozez Singh	India	Music & Musicals
201988	Zubaan	Chittaranjan Tripathy	Mozez Singh	India	Dramas
201989	Zubaan	Chittaranjan Tripathy	Mozez Singh	India	International Movies
201990	Zubaan	Chittaranjan Tripathy	Mozez Singh	India	Music & Musicals

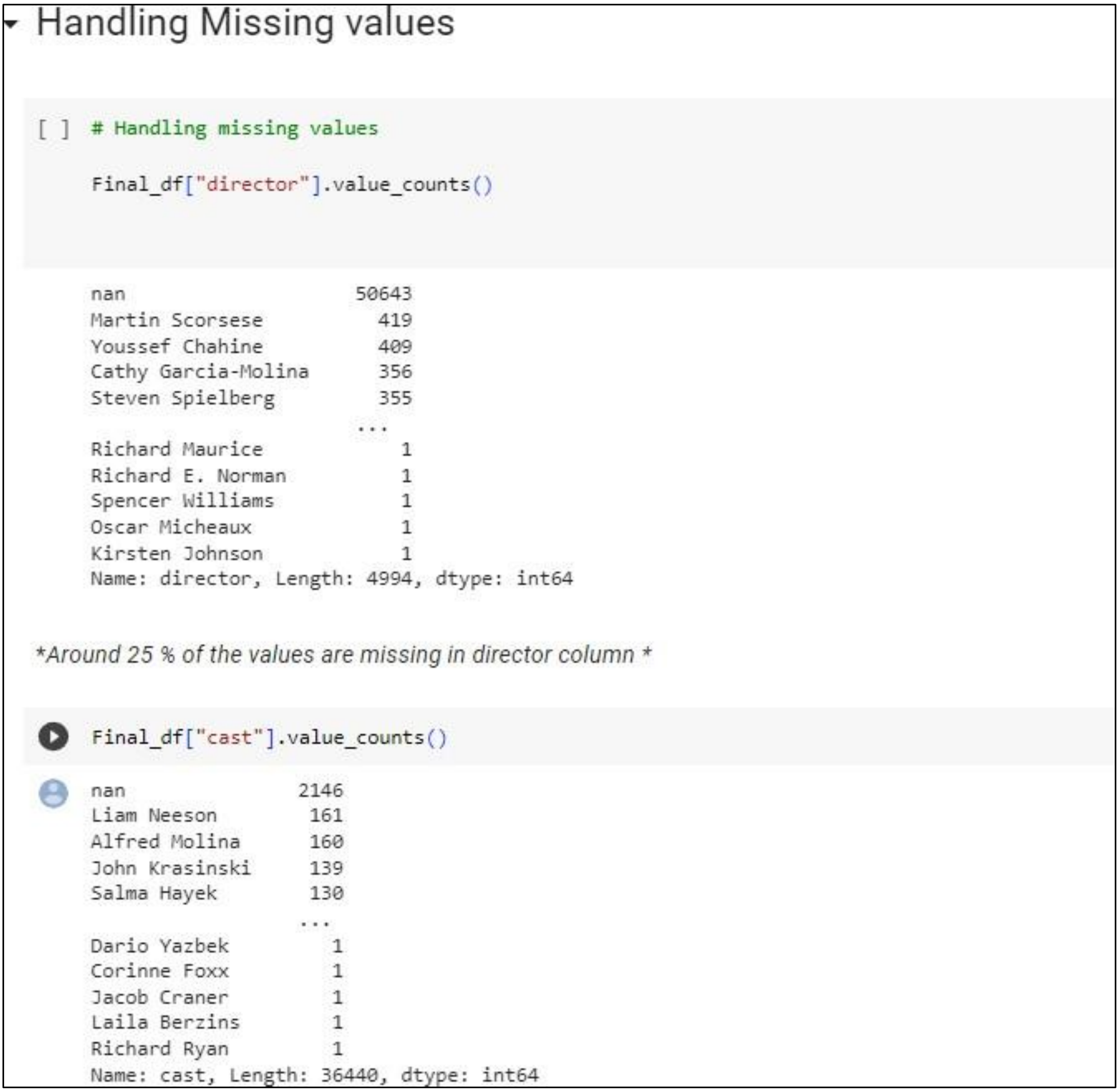
201991 rows × 5 columns



```
[ ] # Merging New_df with original dataFrame

    Remaining_col = df[["title","rating","duration","date_added","release_year"]]

    Final_df = pd.merge(New_df,Remaining_col, on = "title")

    Final_df.head(10)
```

	title	cast	director	country	listed_in	rating	duration	date_added	release_year
0	Dick Johnson Is Dead	nan	Kirsten Johnson	United States	Documentaries	PG-13	90 min	September 25, 2021	2020
1	Blood & Water	Ama Qamata	nan	South Africa	International TV Shows	TV-MA	2 Seasons	September 24, 2021	2021
2	Blood & Water	Ama Qamata	nan	South Africa	TV Dramas	TV-MA	2 Seasons	September 24, 2021	2021
3	Blood & Water	Ama Qamata	nan	South Africa	TV Mysteries	TV-MA	2 Seasons	September 24, 2021	2021
4	Blood & Water	Khosi Ngema	nan	South Africa	International TV Shows	TV-MA	2 Seasons	September 24, 2021	2021
5	Blood & Water	Khosi Ngema	nan	South Africa	TV Dramas	TV-MA	2 Seasons	September 24, 2021	2021
6	Blood & Water	Khosi Ngema	nan	South Africa	TV Mysteries	TV-MA	2 Seasons	September 24, 2021	2021
7	Blood & Water	Gail Mabalane	nan	South Africa	International TV Shows	TV-MA	2 Seasons	September 24, 2021	2021
8	Blood & Water	Gail Mabalane	nan	South Africa	TV Dramas	TV-MA	2 Seasons	September 24, 2021	2021
9	Blood & Water	Gail Mabalane	nan	South Africa	TV Mysteries	TV-MA	2 Seasons	September 24, 2021	2021

```
[ ] Final_df.shape

    (201991, 9)
```



Handling Missing values

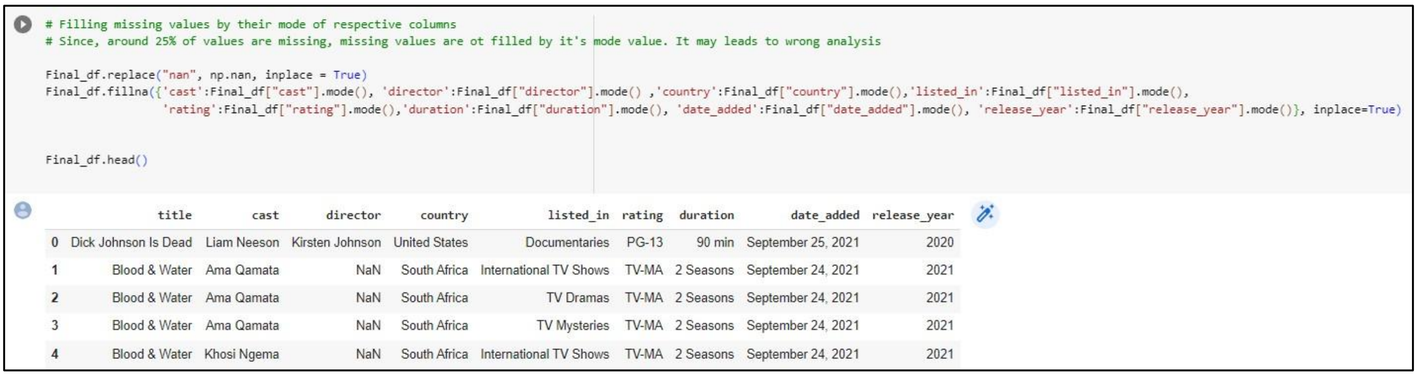
```
[ ] # Handling missing values

    Final_df["director"].value_counts()
```

```
nan                    50643
Martin Scorsese          419
Youssef Chahine          409
Cathy Garcia-Molina      356
Steven Spielberg         355
                        ...
Richard Maurice            1
Richard E. Norman          1
Spencer Williams           1
Oscar Micheaux             1
Kirsten Johnson            1
Name: director, Length: 4994, dtype: int64
```

Around 25 % of the values are missing in director column

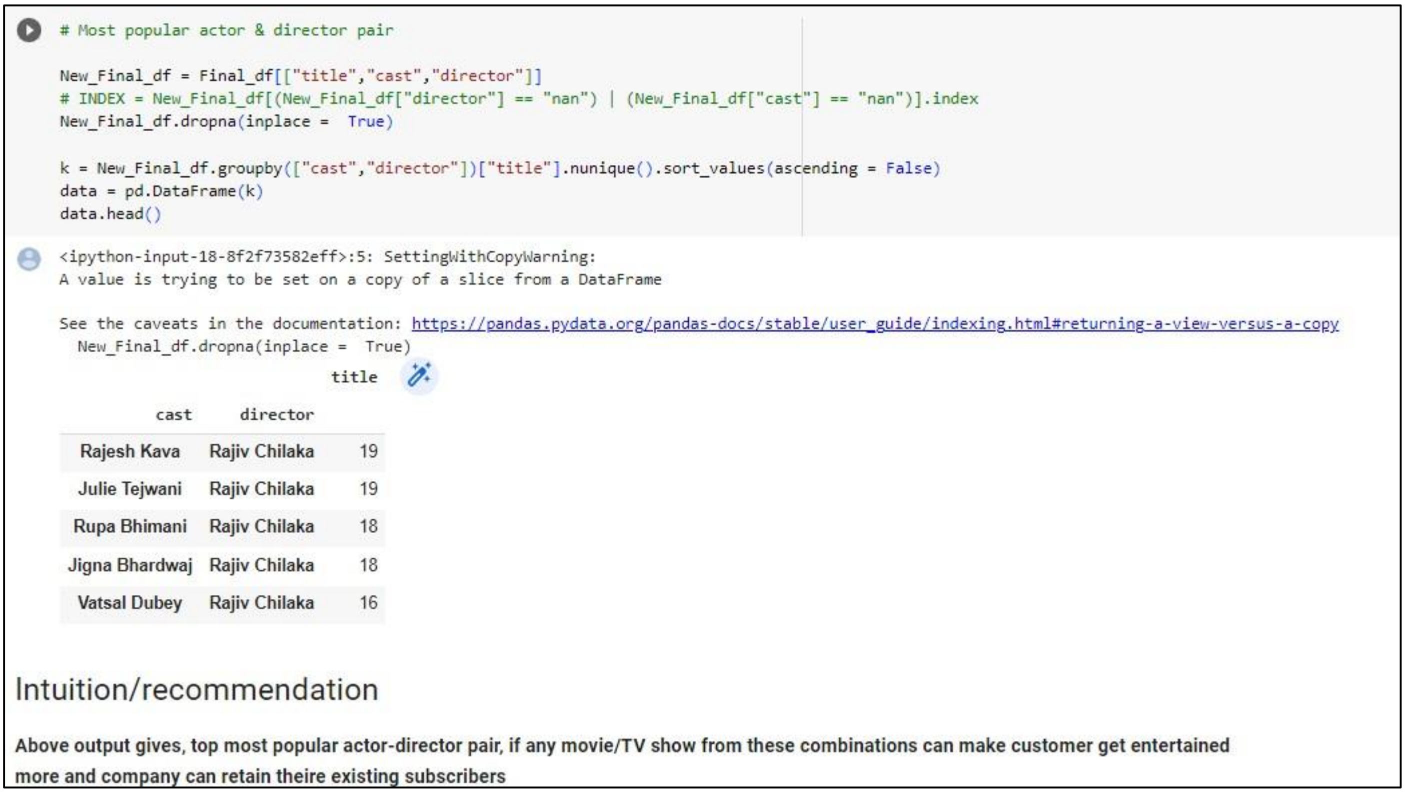
```
Final_df["cast"].value_counts()
```

```
nan               2146
Liam Neeson        161
Alfred Molina      160
John Krasinski     139
Salma Hayek        130
                  ...
Dario Yazbek         1
Corinne Foxx         1
Jacob Craner         1
Laila Berzins        1
Richard Ryan         1
Name: cast, Length: 36440, dtype: int64
```



```
# Filling missing values by their mode of respective columns
# Since, around 25% of values are missing, missing values are ot filled by it's mode value. It may leads to wrong analysis

Final_df.replace("nan", np.nan, inplace = True)
Final_df.fillna({'cast':Final_df["cast"].mode(), 'director':Final_df["director"].mode() ,'country':Final_df["country"].mode(),'listed_in':Final_df["listed_in"].mode(),
                'rating':Final_df["rating"].mode(),'duration':Final_df["duration"].mode(), 'date_added':Final_df["date_added"].mode(), 'release_year':Final_df["release_year"].mode()}, inplace=True)


Final_df.head()
```

	title	cast	director	country	listed_in	rating	duration	date_added	release_year
0	Dick Johnson Is Dead	Liam Neeson	Kirsten Johnson	United States	Documentaries	PG-13	90 min	September 25, 2021	2020
1	Blood & Water	Ama Qamata	NaN	South Africa	International TV Shows	TV-MA	2 Seasons	September 24, 2021	2021
2	Blood & Water	Ama Qamata	NaN	South Africa	TV Dramas	TV-MA	2 Seasons	September 24, 2021	2021
3	Blood & Water	Ama Qamata	NaN	South Africa	TV Mysteries	TV-MA	2 Seasons	September 24, 2021	2021
4	Blood & Water	Khosi Ngema	NaN	South Africa	International TV Shows	TV-MA	2 Seasons	September 24, 2021	2021



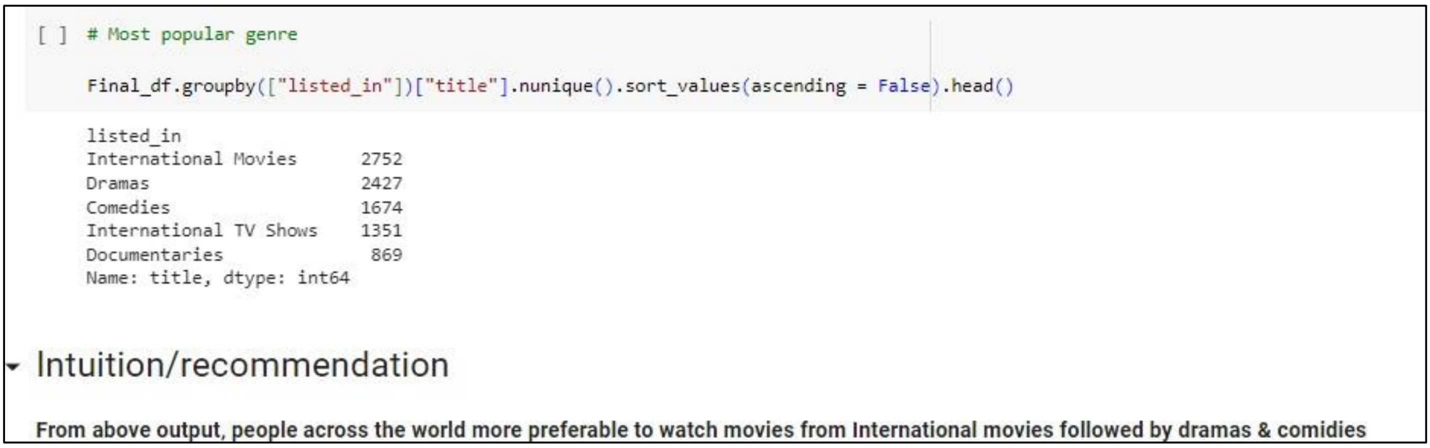
```
# Most popular actor & director pair

New_Final_df = Final_df[["title","cast","director"]]
# INDEX = New_Final_df[(New_Final_df["director"] == "nan") | (New_Final_df["cast"] == "nan")].index
New_Final_df.dropna(inplace =  True)

k = New_Final_df.groupby(["cast","director"])["title"].nunique().sort_values(ascending = False)
data = pd.DataFrame(k)
data.head()
```

```
<ipython-input-18-8f2f73582eff>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  New_Final_df.dropna(inplace =  True)
```

		title
cast	director	
Rajesh Kava	Rajiv Chilaka	19
Julie Tejwani	Rajiv Chilaka	19
Rupa Bhimani	Rajiv Chilaka	18
Jigna Bhardwaj	Rajiv Chilaka	18
Vatsal Dubey	Rajiv Chilaka	16

Intuition/recommendation

Above output gives, top most popular actor-director pair, if any movie/TV show from these combinations can make customer get entertained more and company can retain theire existing subscribers



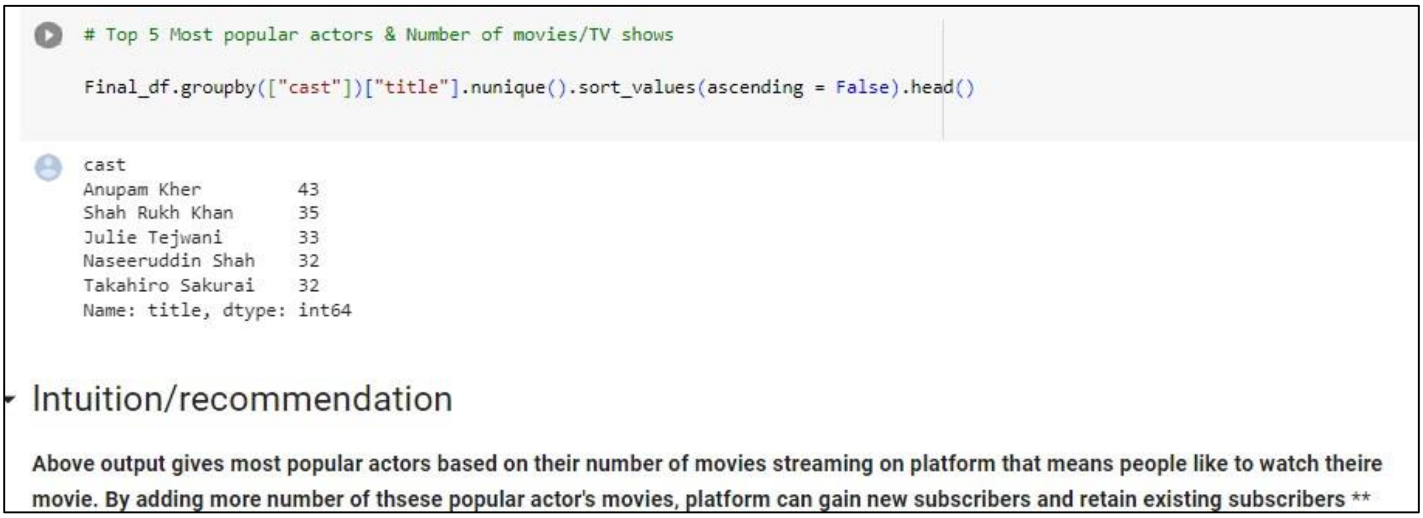
```
[ ] # Most popular genre

    Final_df.groupby(["listed_in"])["title"].nunique().sort_values(ascending = False).head()
```

```
listed_in
International Movies      2752
Dramas                    2427
Comedies                  1674
International TV Shows    1351
Documentaries              869
Name: title, dtype: int64
```

Intuition/recommendation

From above output, people across the world more preferable to watch movies from International movies followed by dramas & comidies

Name: Raghu
Business case: Netflix **Date: 26/May/2023**

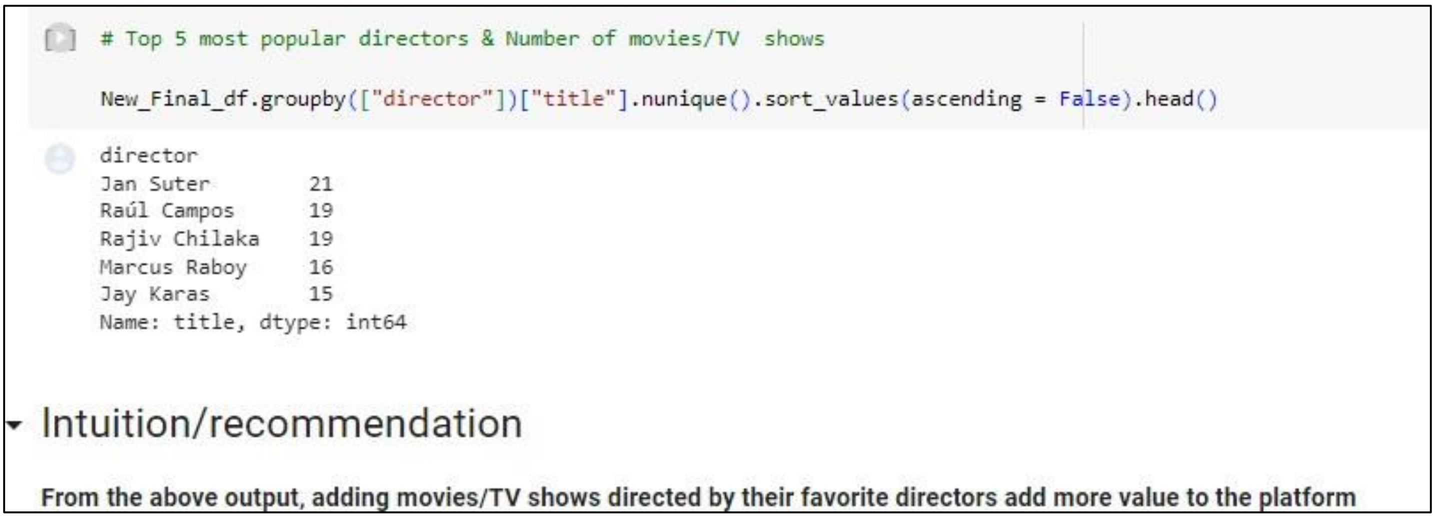


```
# Top 5 Most popular actors & Number of movies/TV shows

Final_df.groupby(["cast"])["title"].nunique().sort_values(ascending = False).head()
```

```
cast
Anupam Kher         43
Shah Rukh Khan      35
Julie Tejwani       33
Naseeruddin Shah    32
Takahiro Sakurai    32
Name: title, dtype: int64
```

Intuition/recommendation

Above output gives most popular actors based on their number of movies streaming on platform that means people like to watch theire movie. By adding more number of thsese popular actor's movies, platform can gain new subscribers and retain existing subscribers **

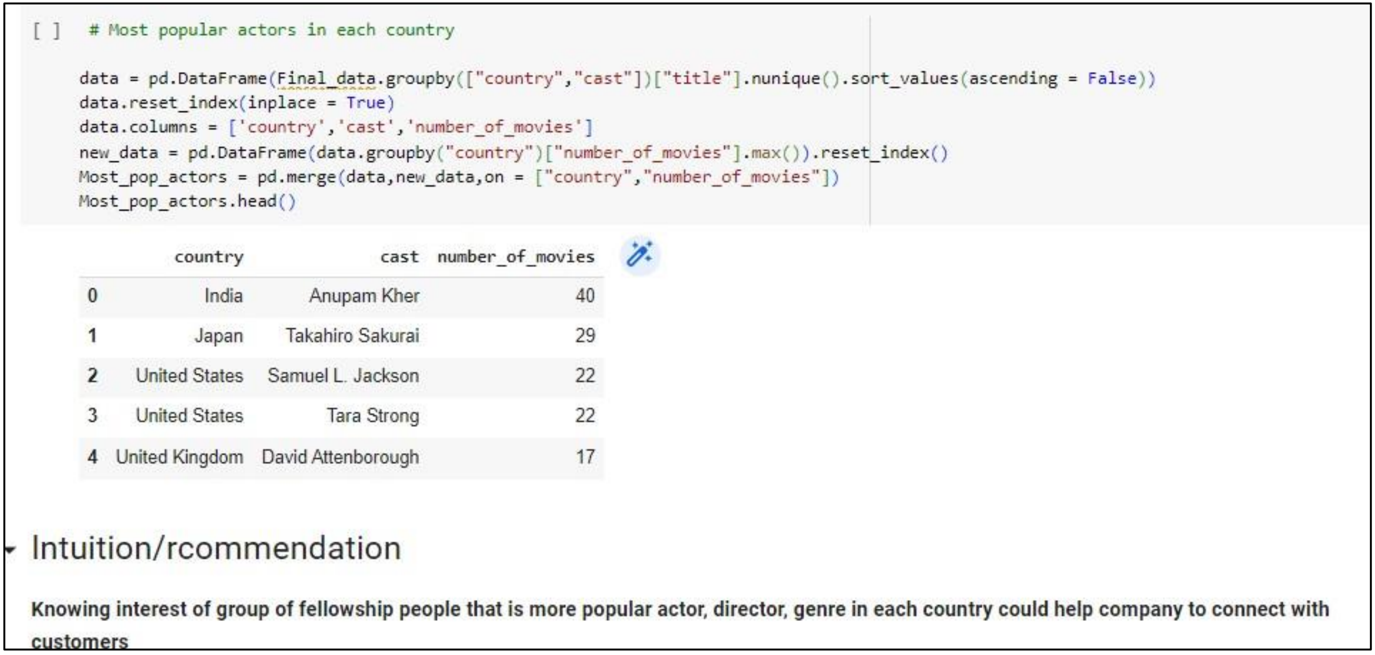


```
# Top 5 most popular directors & Number of movies/TV  shows

New_Final_df.groupby(["director"])["title"].nunique().sort_values(ascending = False).head()
```

```
director
Jan Suter         21
Raúl Campos       19
Rajiv Chilaka     19
Marcus Raboy      16
Jay Karas         15
Name: title, dtype: int64
```

Intuition/recommendation

From the above output, adding movies/TV shows directed by their favorite directors add more value to the platform

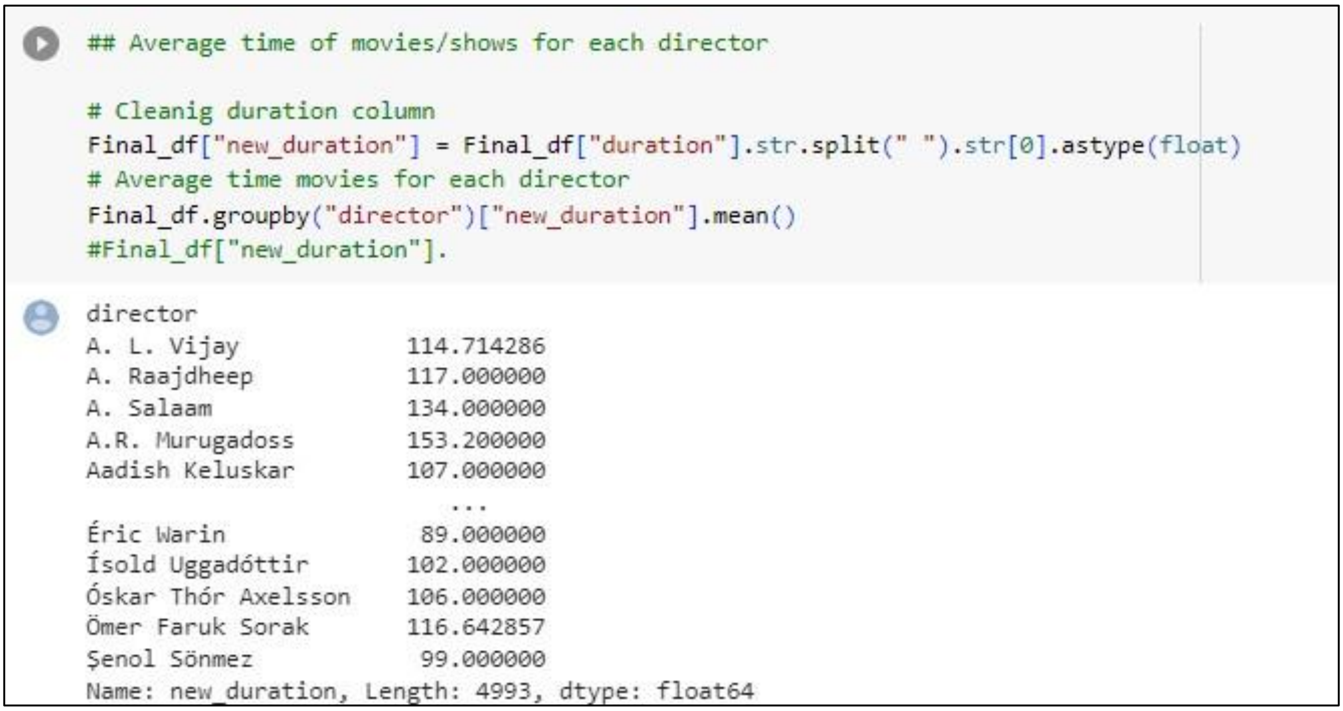


```
[ ]  # Most popular actors in each country

     data = pd.DataFrame(Final_data.groupby(["country","cast"])["title"].nunique().sort_values(ascending = False))
     data.reset_index(inplace = True)
     data.columns = ['country','cast','number_of_movies']
     new_data = pd.DataFrame(data.groupby("country")["number_of_movies"].max()).reset_index()
     Most_pop_actors = pd.merge(data,new_data,on = ["country","number_of_movies"])
     Most_pop_actors.head()
```

	country	cast	number_of_movies
0	India	Anupam Kher	40
1	Japan	Takahiro Sakurai	29
2	United States	Samuel L. Jackson	22
3	United States	Tara Strong	22
4	United Kingdom	David Attenborough	17

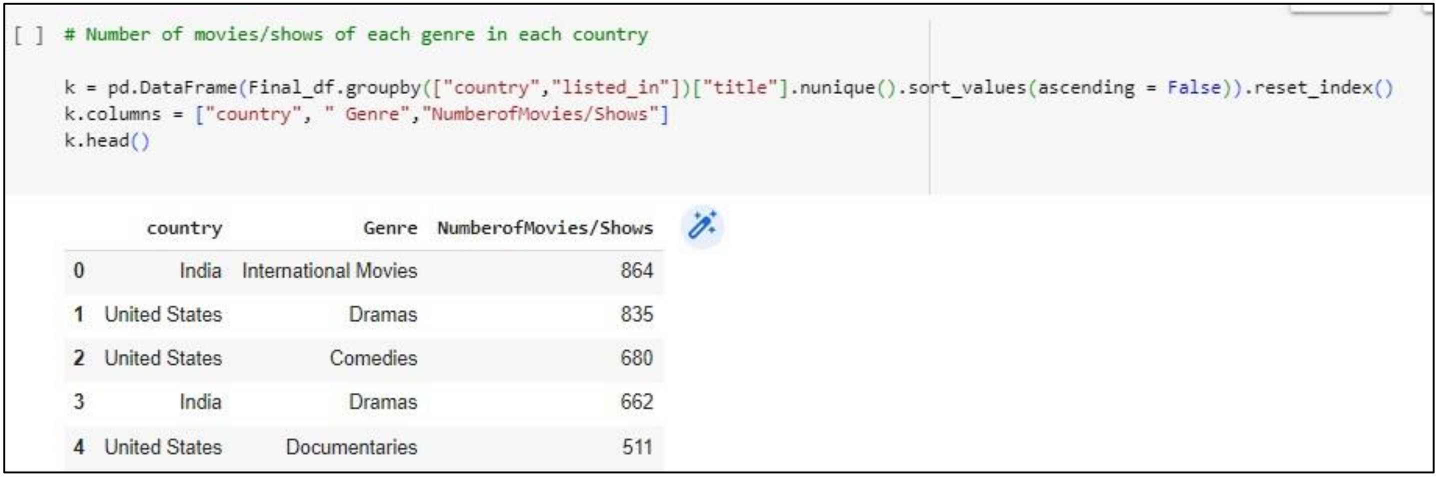
Intuition/rcommendation

Knowing interest of group of fellowship people that is more popular actor, director, genre in each country could help company to connect with customers



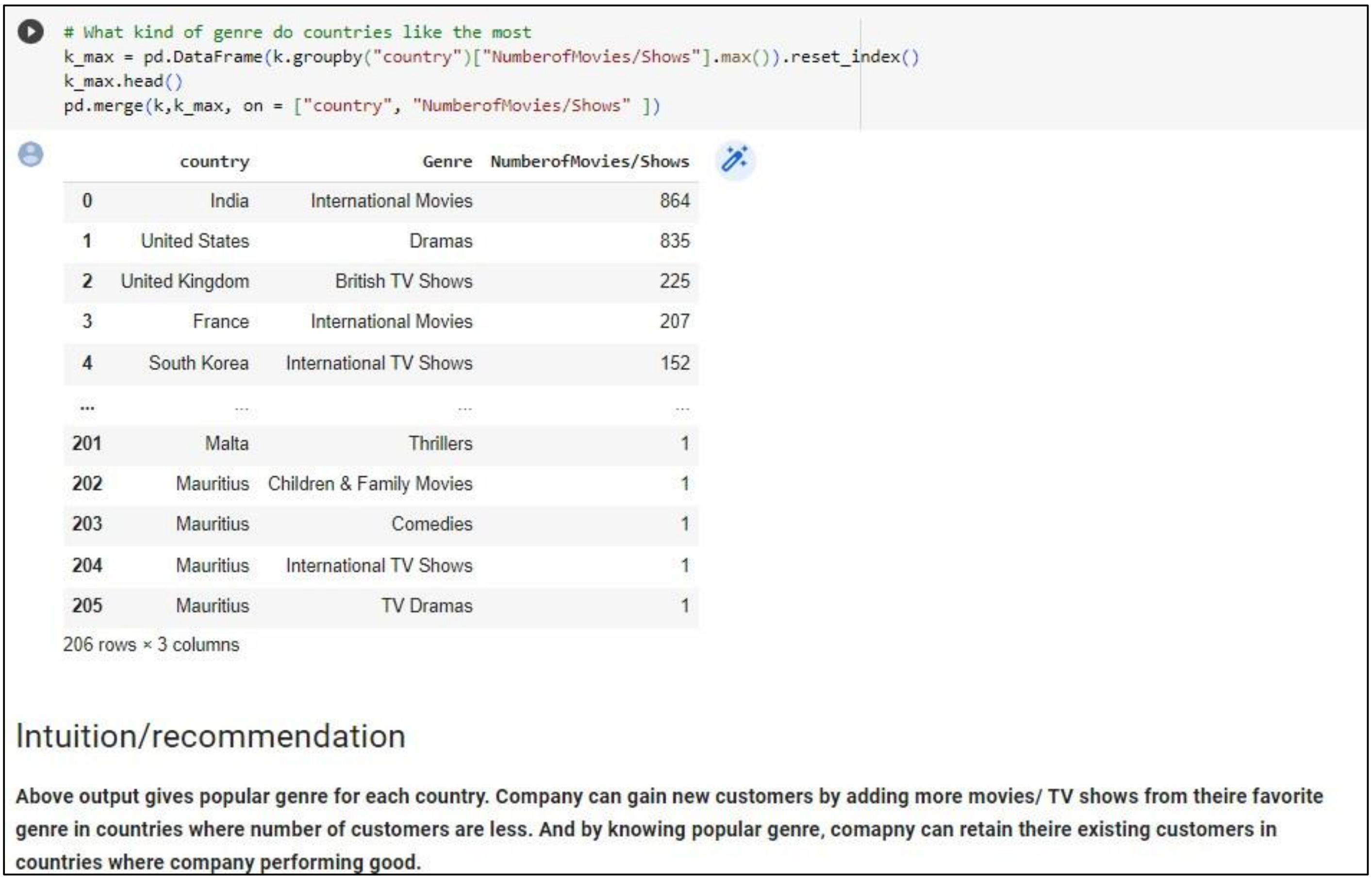
```
## Average time of movies/shows for each director

# Cleanig duration column
Final_df["new_duration"] = Final_df["duration"].str.split(" ").str[0].astype(float)
# Average time movies for each director
Final_df.groupby("director")["new_duration"].mean()
#Final_df["new_duration"].
```

```
director
A. L. Vijay             114.714286
A. Raajdheep            117.000000
A. Salaam               134.000000
A.R. Murugadoss         153.200000
Aadish Keluskar         107.000000
                           ...
Éric Warin               89.000000
Ísold Uggadóttir        102.000000
Óskar Thór Axelsson     106.000000
Ömer Faruk Sorak        116.642857
Şenol Sönmez             99.000000
Name: new_duration, Length: 4993, dtype: float64
```



```
[ ] # Number of movies/shows of each genre in each country

    k = pd.DataFrame(Final_df.groupby(["country","listed_in"])["title"].nunique().sort_values(ascending = False)).reset_index()
    k.columns = ["country", " Genre","NumberofMovies/Shows"]
    k.head()
```

	country	Genre	NumberofMovies/Shows
0	India	International Movies	864
1	United States	Dramas	835
2	United States	Comedies	680
3	India	Dramas	662
4	United States	Documentaries	511



```
# What kind of genre do countries like the most
k_max = pd.DataFrame(k.groupby("country")["NumberofMovies/Shows"].max()).reset_index()
k_max.head()
pd.merge(k,k_max, on = ["country", "NumberofMovies/Shows" ])
```

	country	Genre	NumberofMovies/Shows
0	India	International Movies	864
1	United States	Dramas	835
2	United Kingdom	British TV Shows	225
3	France	International Movies	207
4	South Korea	International TV Shows	152
...	...	...	...
201	Malta	Thrillers	1
202	Mauritius	Children & Family Movies	1
203	Mauritius	Comedies	1
204	Mauritius	International TV Shows	1
205	Mauritius	TV Dramas	1

206 rows × 3 columns

Intuition/recommendation

Above output gives popular genre for each country. Company can gain new customers by adding more movies/ TV shows from theire favorite genre in countries where number of customers are less. And by knowing popular genre, comapny can retain theire existing customers in countries where company performing good.