In [7]:
```
import pandas as pd
import pyxlsb
import numpy as np
```

In [8]:
```
dataframe = pd.read_excel("stc TV Data Set_T3.xlsb",index_col=0)
```

In [9]:
```
dataframe.shape
```

Out[9]:
```
(1048575, 5)
```

In [10]:
```
dataframe.head()
```

Out[10]:

	user_id_maped	program_name	rating	date_	program_genre
0	26138	100 treets	1	2017-05-27	Drama
1	7946	Moana	1	2017-05-21	Animation
2	7418	The Mermaid Princess	1	2017-08-10	Animation
3	19307	The Mermaid Princess	2	2017-07-26	Animation
4	15860	Churchill	2	2017-07-07	Biography

In [11]:
```
dataframe.describe()
```

Out[11]:

	user_id_maped	rating
count	1.048575e+06	1.048575e+06
mean	1.709266e+04	2.497283e+00
std	1.003513e+04	1.119837e+00
min	1.000000e+00	1.000000e+00
25%	8.253000e+03	1.000000e+00
50%	1.714900e+04	2.000000e+00
75%	2.566500e+04	3.000000e+00
max	3.428000e+04	4.000000e+00

In [12]:
```
dataframe.isnull().any()
```

Out[12]:
```
user_id_maped    False
program_name     False
rating           False
date_            False
program_genre    False
dtype: bool
```

In [13]:
```
import matplotlib.pyplot as plt
import plotly
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [14]:
```
movie_features_df=dataframe.pivot_table(index='program_name',columns='user_id_maped',values='rating').fillna(0)
movie_features_df.head()
```

Out[14]:

user_id_maped	1	5	9	11	15	17	20	26	28	30	...	34259	34261	34263	34265	34267	34269	34271	34273	34277	34280
program_name																					
#FollowFriday	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10 Days in a Madhouse	0.0	0.0	0.0	0.0	1.5	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
100 treets	0.0	0.0	0.0	1.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
101 Dalmatians	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
102 Dalmatians	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 11578 columns

In [15]:
```
from scipy.sparse import csr_matrix
from sklearn.neighbors import NearestNeighbors
movie_features_df_matrix = csr_matrix(movie_features_df.values)
model_knn = NearestNeighbors(metric = 'cosine', algorithm = 'brute')
model_knn.fit(movie_features_df_matrix)
```

Out[15]:
```
NearestNeighbors(algorithm='brute', metric='cosine')
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [16]:
```
movie_features_df_selection= movie_features_df.reset_index()
movie_features_df_selection[['program_name']]
dataframe[['program_name']].head(10)
```

Out[16]:

	program_name
0	100 treets
1	Moana
2	The Mermaid Princess
3	The Mermaid Princess
4	Churchill
5	Beavis And Butt-Head Do America
6	The Mermaid Princess
7	Coco

8	Kidnap
9	The Accountant

In [ ]:

In [17]:
```
program_name='Moana'
recomendations=5

query_index = np.random.choice(movie_features_df.shape[0])
distances, indices = model_knn.kneighbors(movie_features_df.
                    iloc[movie_features_df_selection.index[movie_features_df_selection['program_name'] == program_name].tolist()
n_neighbors = recomendations+1)

for i in range(0, len(distances.flatten())):
    if i == 0:
        print('Recommendations for {0}:\n'.format(movie_features_df.index[movie_features_df_selection.index[movie_features_df_sel
    else:
        print('{0}: {1}, with distance of {2}:'.format(i, movie_features_df.index[indices.flatten()[i]], distances.flatten()[i])
```

```
Recommendations for Moana:

1: Trolls, with distance of 0.42764217010640215:
2: Surf's Up : WaveMania, with distance of 0.4705763355181768:
3: The Mermaid Princess, with distance of 0.5066377099343184:
4: The Boss Baby, with distance of 0.551442834662541:
5: The Jetsons & WWE: Robo-WrestleMania!, with distance of 0.5610577907608365:
```

In [ ]: