Case Study Presentation

Subject - Complex Network Analysis

Presented by - Ramya Mercy Rajan

Roll number - AA.SC.P2MCA2107434

1. Choose appropriate dataset of your choice from the following sites. However, students are fre choose any other relevant dataset for complex networks case study.

I have selected dataset from Kaggle.com https://www.kaggle.com/code/vikassingh1996/netflix movies-and-shows-plotly-recommender-sys/data

Data contains Rows-5837, Columns-12

Netflix is a subscription-based streaming service that allows our members to watch TV shows a movies on an internet-connected device.

About this Dataset: Netflix is one of the most popular media and video streaming platforms. Th have over 8000 movies or tv shows available on their platform, as of mid-2021, they have over 200M Subscribers globally. This tabular dataset consists of listings of all the movies and tv show available on Netflix, along with details such as - cast, directors, ratings, release year, duration, e

In this dataset I have selected two features Source as"duration" and Target as "title" for the Network Analysis visualization purpose in order to know the series with more running time , degree of centralities, item which has most of scores compared to others.

The highest seen is 15 Seasons

Season simply means the collection of certain Episodes broadcasted within a certain time period in one season or year.

Q2) Create Complex Network from the chosen dataset and visualize it properly. If thedataset is too huge, then subset of data can be visualized.

I'm using the original dataset for network analysis. I have selected Title which is the name of Series or Movies and Duration to check the running durations of each movies or series. New "Data" is created.

In [9]:

```
df = pd.read_csv("netflix.csv")
```

In [10]:

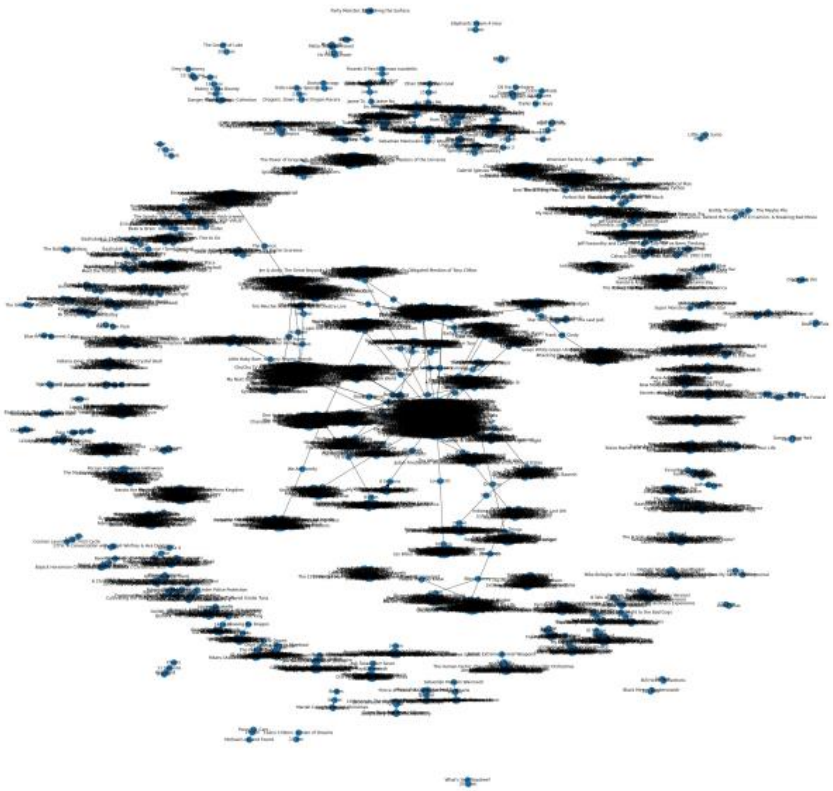
```
Data = nx.from_pandas_edgelist(df,source="duration",target="title")
```

In [11]:

```
nx.info(Data)
```

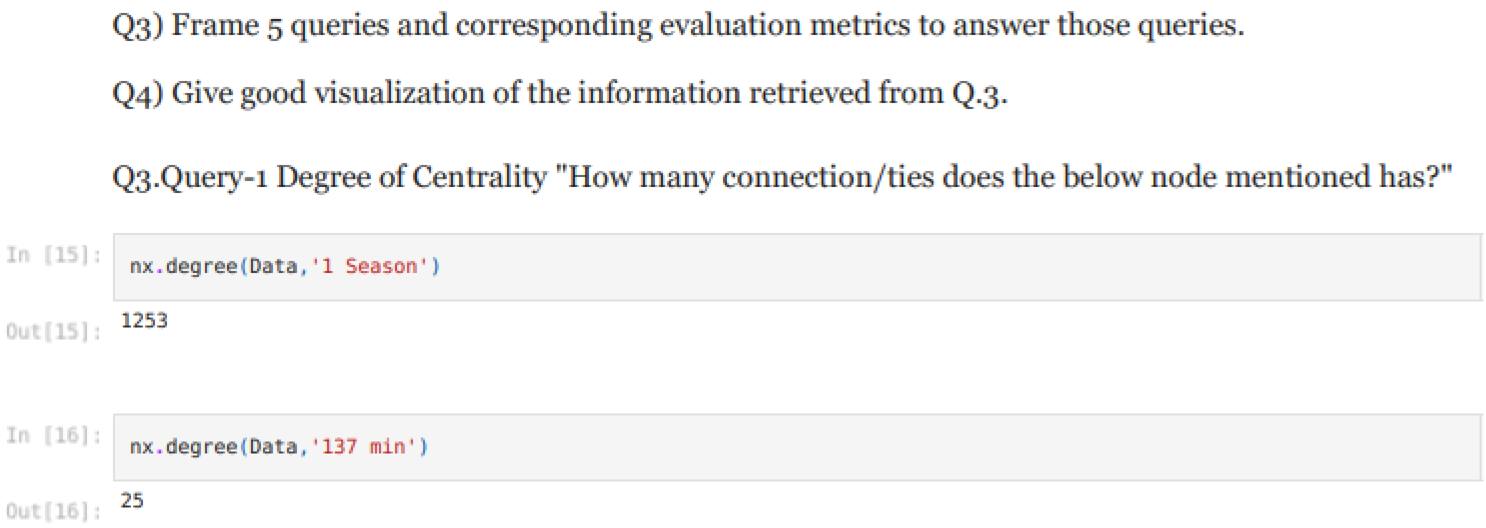
Out[11]:

```
'Graph with 5974 nodes and 5827 edges'
```



Visualization of Nodes and its connection

Query 1 – How many connections of movies does the duration “1 Season” have? Degree of Centrality



Q3) Frame 5 queries and corresponding evaluation metrics to answer those queries.

Q4) Give good visualization of the information retrieved from Q.3.

Q3.Query-1 Degree of Centrality "How many connection/ties does the below node mentioned has?"

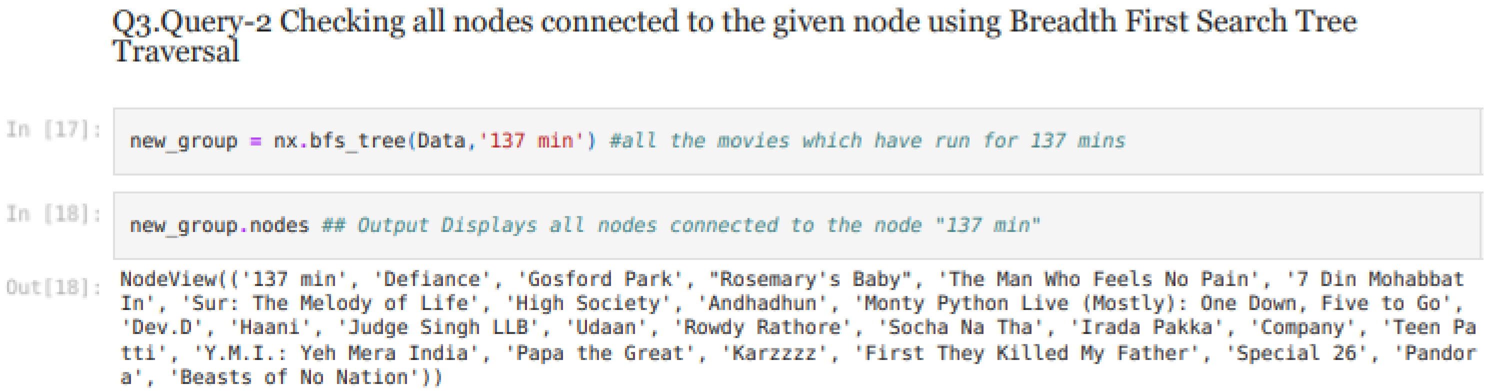
```
In [15]: nx.degree(Data,'1 Season')

Out[15]: 1253
```

```
In [16]: nx.degree(Data,'137 min')

Out[16]: 25
```

Query 2 – How many movies are connected with shortest path to 137 minutes duration? BFS



Q3.Query-2 Checking all nodes connected to the given node using Breadth First Search Tree Traversal

In [17]:

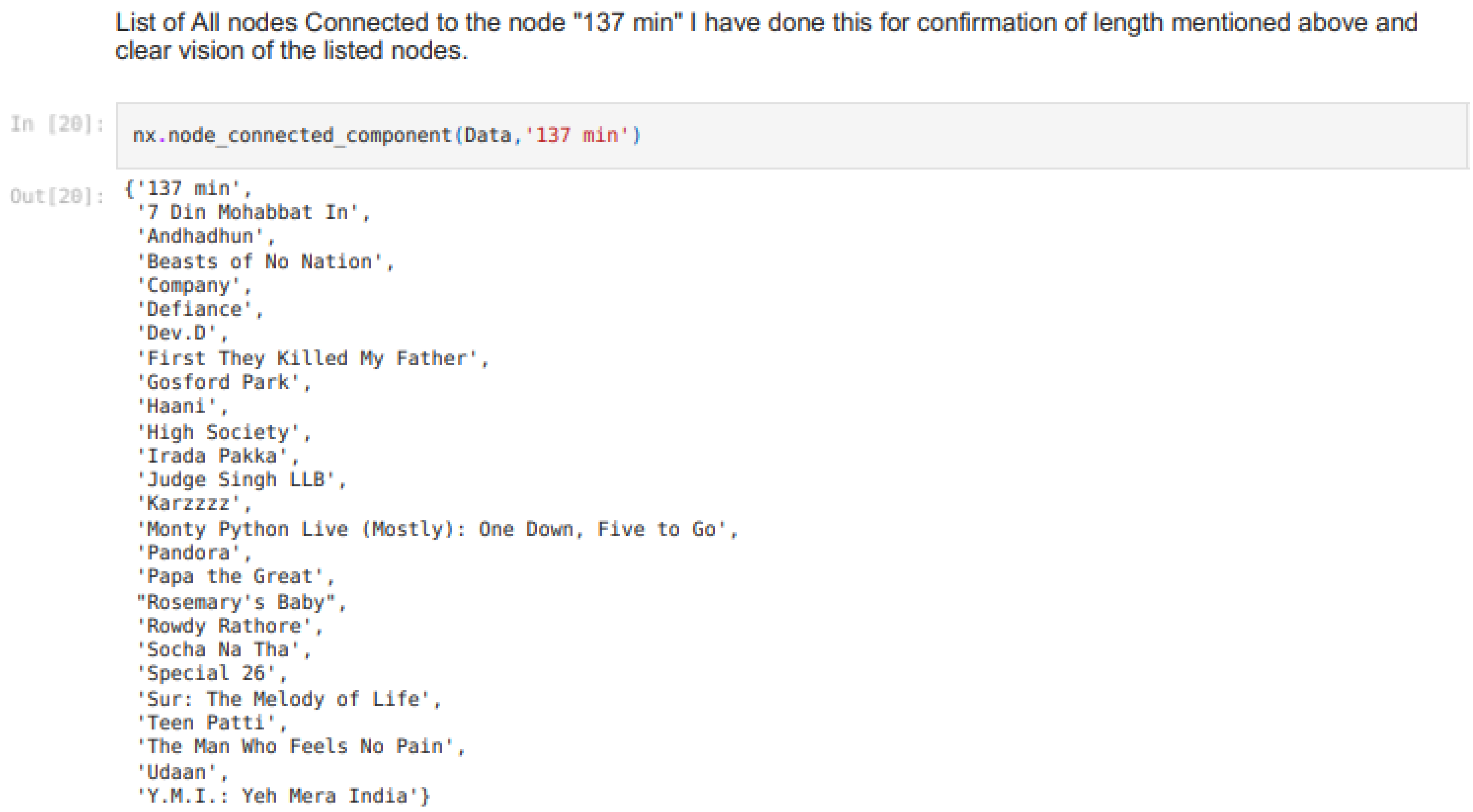
```
new_group = nx.bfs_tree(Data,'137 min') #all the movies which have run for 137 mins
```

In [18]:

```
new_group.nodes ## Output Displays all nodes connected to the node "137 min"
```

Out[18]:

```
NodeView(('137 min', 'Defiance', 'Gosford Park', "Rosemary's Baby", 'The Man Who Feels No Pain', '7 Din Mohabbat
In', 'Sur: The Melody of Life', 'High Society', 'Andhadhun', 'Monty Python Live (Mostly): One Down, Five to Go',
'Dev.D', 'Haani', 'Judge Singh LLB', 'Udaan', 'Rowdy Rathore', 'Socha Na Tha', 'Irada Pakka', 'Company', 'Teen Pa
tti', 'Y.M.I.: Yeh Mera India', 'Papa the Great', 'Karzzzz', 'First They Killed My Father', 'Special 26', 'Pandor
a', 'Beasts of No Nation'))
```

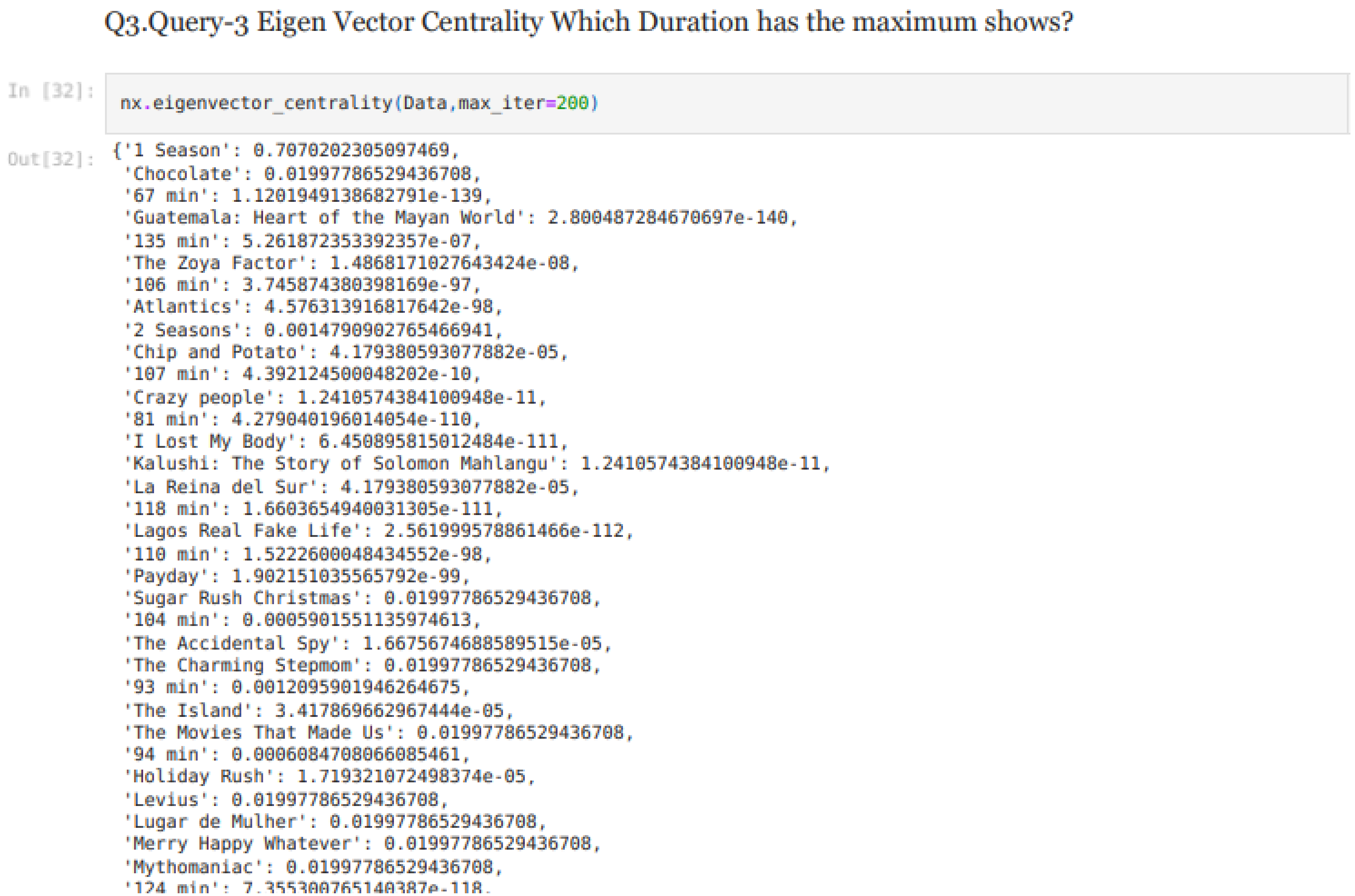


List of All nodes Connected to the node "137 min" I have done this for confirmation of length mentioned above and clear vision of the listed nodes.

```
In [20]: nx.node_connected_component(Data,'137 min')
```

```
Out[20]: {'137 min',
          '7 Din Mohabbat In',
          'Andhadhun',
          'Beasts of No Nation',
          'Company',
          'Defiance',
          'Dev.D',
          'First They Killed My Father',
          'Gosford Park',
          'Haani',
          'High Society',
          'Irada Pakka',
          'Judge Singh LLB',
          'Karzzzz',
          'Monty Python Live (Mostly): One Down, Five to Go',
          'Pandora',
          'Papa the Great',
          "Rosemary's Baby",
          'Rowdy Rathore',
          'Socha Na Tha',
          'Special 26',
          'Sur: The Melody of Life',
          'Teen Patti',
          'The Man Who Feels No Pain',
          'Udaan',
          'Y.M.I.: Yeh Mera India'}
```

Query 3- Which duration have the maximum shows? Eigen Vector



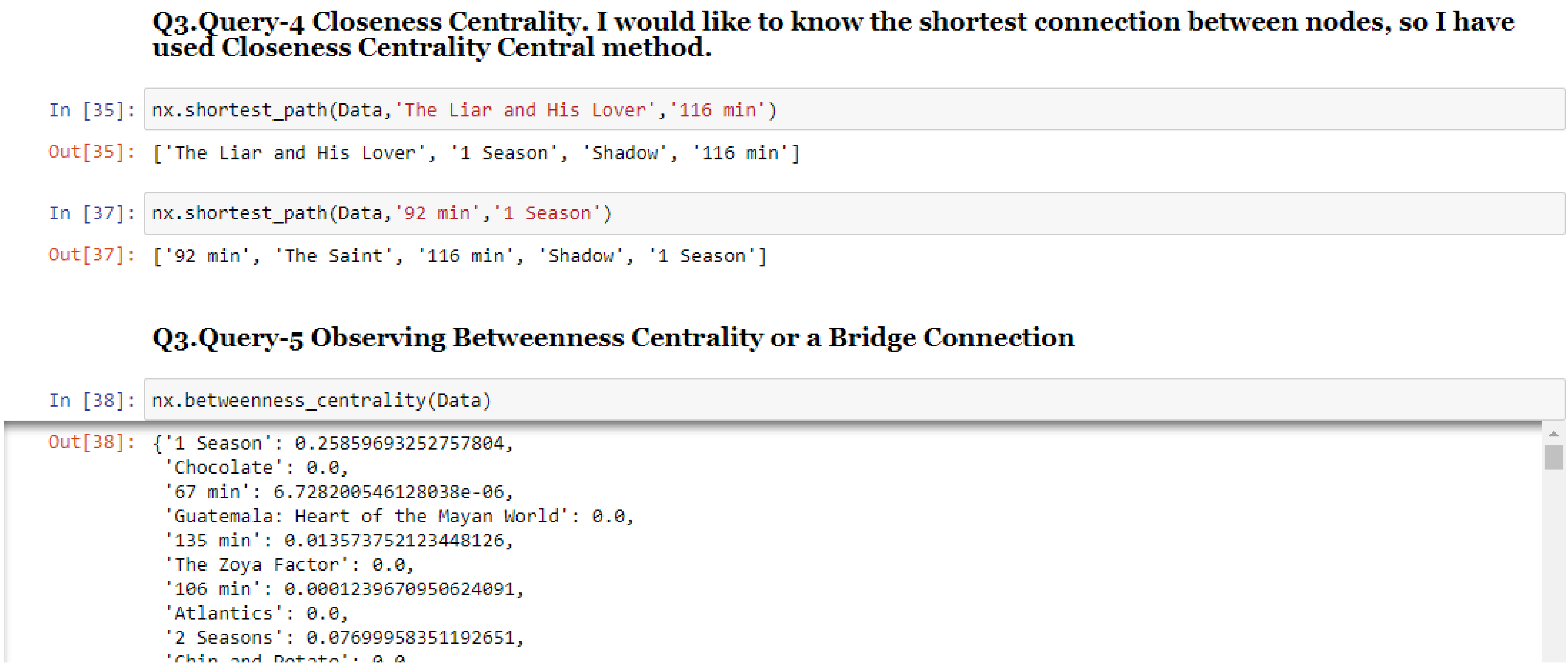
Q3.Query-3 Eigen Vector Centrality Which Duration has the maximum shows?

```
In [32]:  nx.eigenvector_centrality(Data,max_iter=200)

Out[32]: {'1 Season': 0.7070202305097469,
          'Chocolate': 0.01997786529436708,
          '67 min': 1.1201949138682791e-139,
          'Guatemala: Heart of the Mayan World': 2.800487284670697e-140,
          '135 min': 5.261872353392357e-07,
          'The Zoya Factor': 1.4868171027643424e-08,
          '106 min': 3.745874380398169e-97,
          'Atlantics': 4.576313916817642e-98,
          '2 Seasons': 0.0014790902765466941,
          'Chip and Potato': 4.179380593077882e-05,
          '107 min': 4.392124500048202e-10,
          'Crazy people': 1.2410574384100948e-11,
          '81 min': 4.279040196014054e-110,
          'I Lost My Body': 6.450895815012484e-111,
          'Kalushi: The Story of Solomon Mahlangu': 1.2410574384100948e-11,
          'La Reina del Sur': 4.179380593077882e-05,
          '118 min': 1.6603654940031305e-111,
          'Lagos Real Fake Life': 2.561999578861466e-112,
          '110 min': 1.5222600048434552e-98,
          'Payday': 1.902151035565792e-99,
          'Sugar Rush Christmas': 0.01997786529436708,
          '104 min': 0.0005901551135974613,
          'The Accidental Spy': 1.6675674688589515e-05,
          'The Charming Stepmom': 0.01997786529436708,
          '93 min': 0.0012095901946264675,
          'The Island': 3.417869662967444e-05,
          'The Movies That Made Us': 0.01997786529436708,
          '94 min': 0.0006084708066085461,
          'Holiday Rush': 1.719321072498374e-05,
          'Levius': 0.01997786529436708,
          'Lugar de Mulher': 0.01997786529436708,
          'Merry Happy Whatever': 0.01997786529436708,
          'Mythomaniac': 0.01997786529436708,
          '124 min': 7.355300765140387e-118,
```

Query 4- What is the shortest connection between a particular movie and a particular duration? Closeness Centrality.

Query 5- What is the betweenness Centrality of one particular duration, for example Season 15 comes in between how many movies?



Q3.Query-4 Closeness Centrality. I would like to know the shortest connection between nodes, so I have used Closeness Centrality Central method.

```
In [35]: nx.shortest_path(Data,'The Liar and His Lover','116 min')

Out[35]: ['The Liar and His Lover', '1 Season', 'Shadow', '116 min']
```

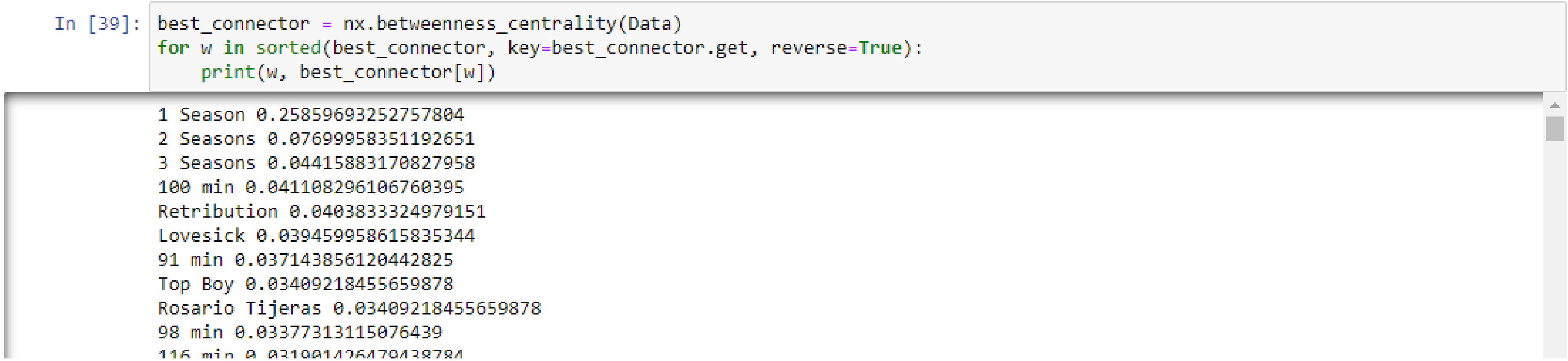
```
In [37]: nx.shortest_path(Data,'92 min','1 Season')

Out[37]: ['92 min', 'The Saint', '116 min', 'Shadow', '1 Season']
```

Q3.Query-5 Observing Betweenness Centrality or a Bridge Connection

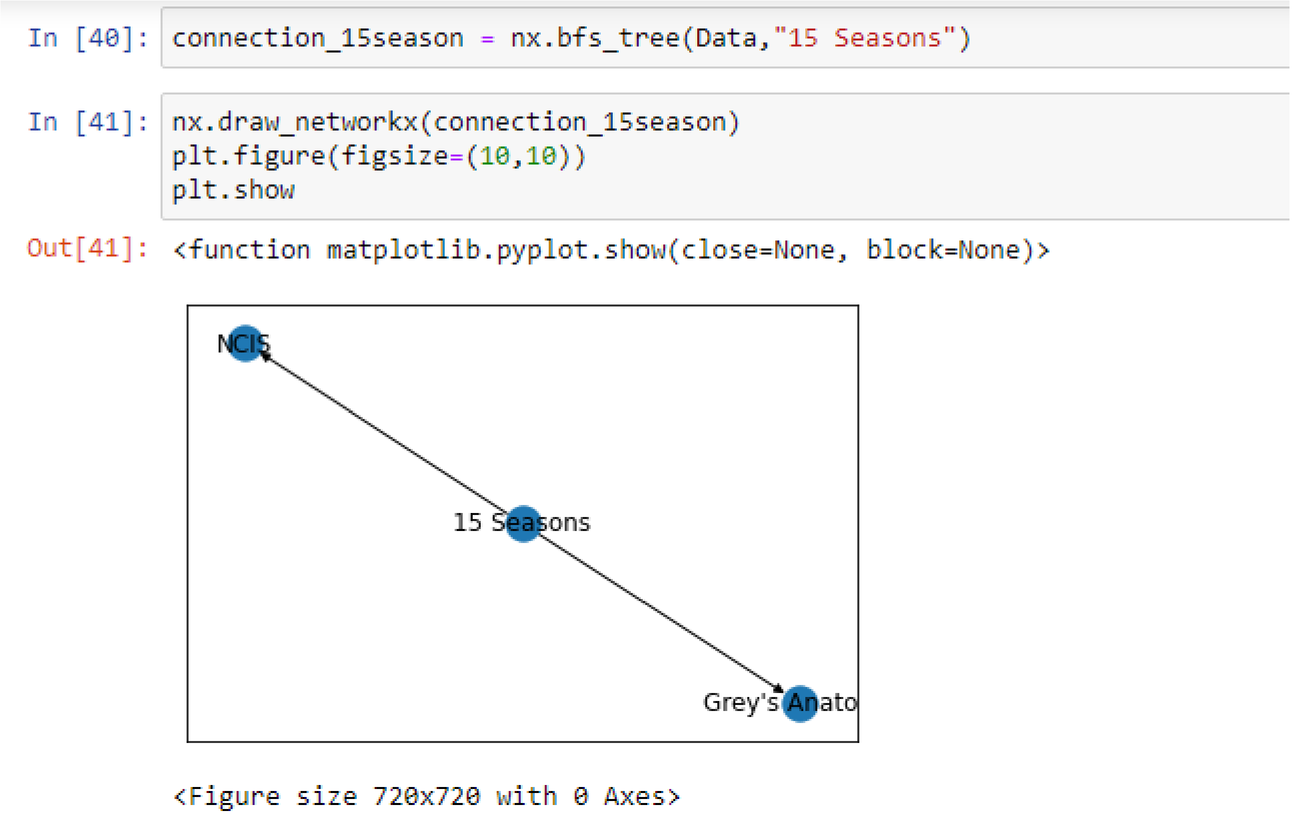
```
In [38]: nx.betweenness_centrality(Data)

Out[38]: {'1 Season': 0.25859693252757804,
          'Chocolate': 0.0,
          '67 min': 6.728200546128038e-06,
          'Guatemala: Heart of the Mayan World': 0.0,
          '135 min': 0.013573752123448126,
          'The Zoya Factor': 0.0,
          '106 min': 0.0001239670950624091,
          'Atlantics': 0.0,
          '2 Seasons': 0.07699958351192651,
          'Chip and Potato': 0.0
```



```
In [39]: best_connector = nx.betweenness_centrality(Data)
         for w in sorted(best_connector, key=best_connector.get, reverse=True):
             print(w, best_connector[w])
```

```
1 Season 0.25859693252757804
2 Seasons 0.07699958351192651
3 Seasons 0.04415883170827958
100 min 0.041108296106760395
Retribution 0.0403833324979151
Lovesick 0.039459958615835344
91 min 0.037143856120442825
Top Boy 0.03409218455659878
Rosario Tijeras 0.03409218455659878
98 min 0.03377313115076439
116 min 0.031901426479438784
```

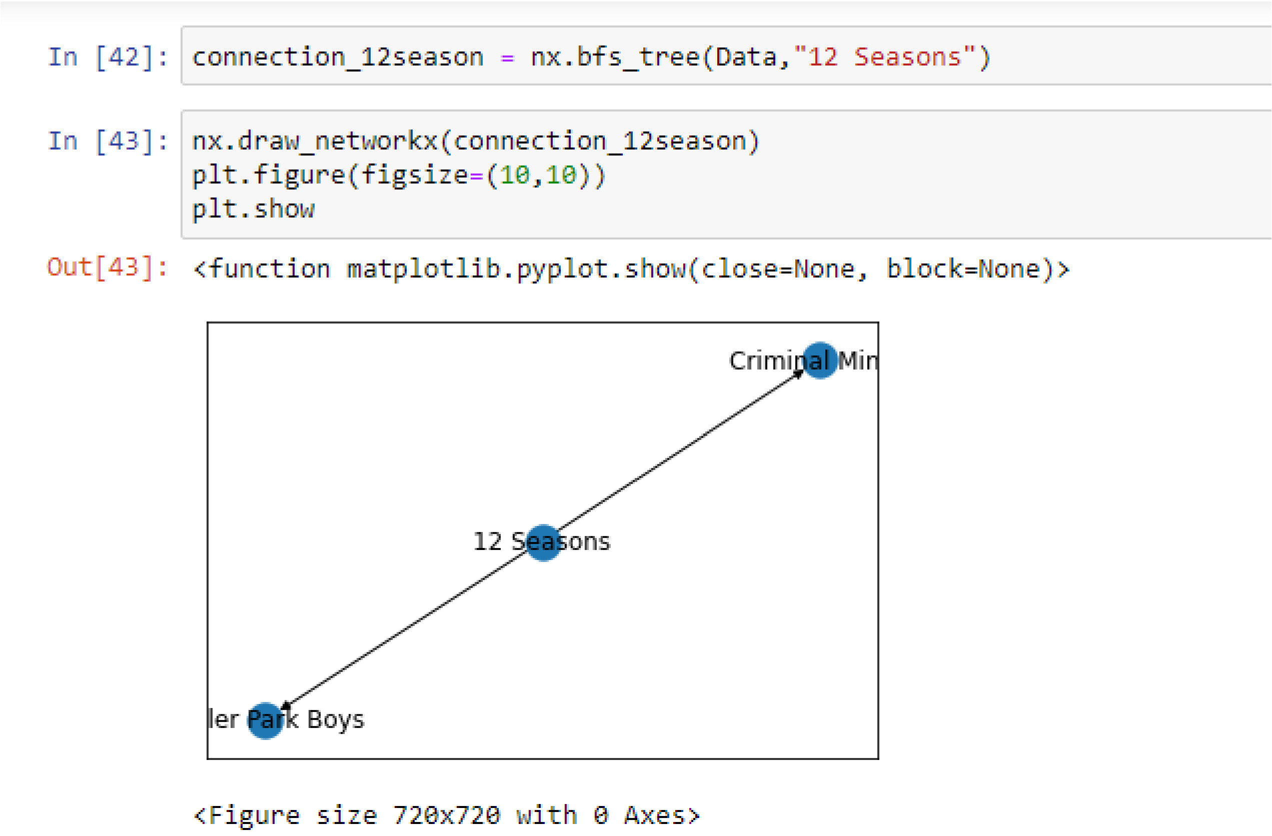


```
In [40]: connection_15season = nx.bfs_tree(Data,"15 Seasons")
```

```
In [41]: nx.draw_networkx(connection_15season)
         plt.figure(figsize=(10,10))
         plt.show
```

```
Out[41]: <function matplotlib.pyplot.show(close=None, block=None)>
```

```
<Figure size 720x720 with 0 Axes>
```



```
In [42]: connection_12season = nx.bfs_tree(Data,"12 Seasons")
```

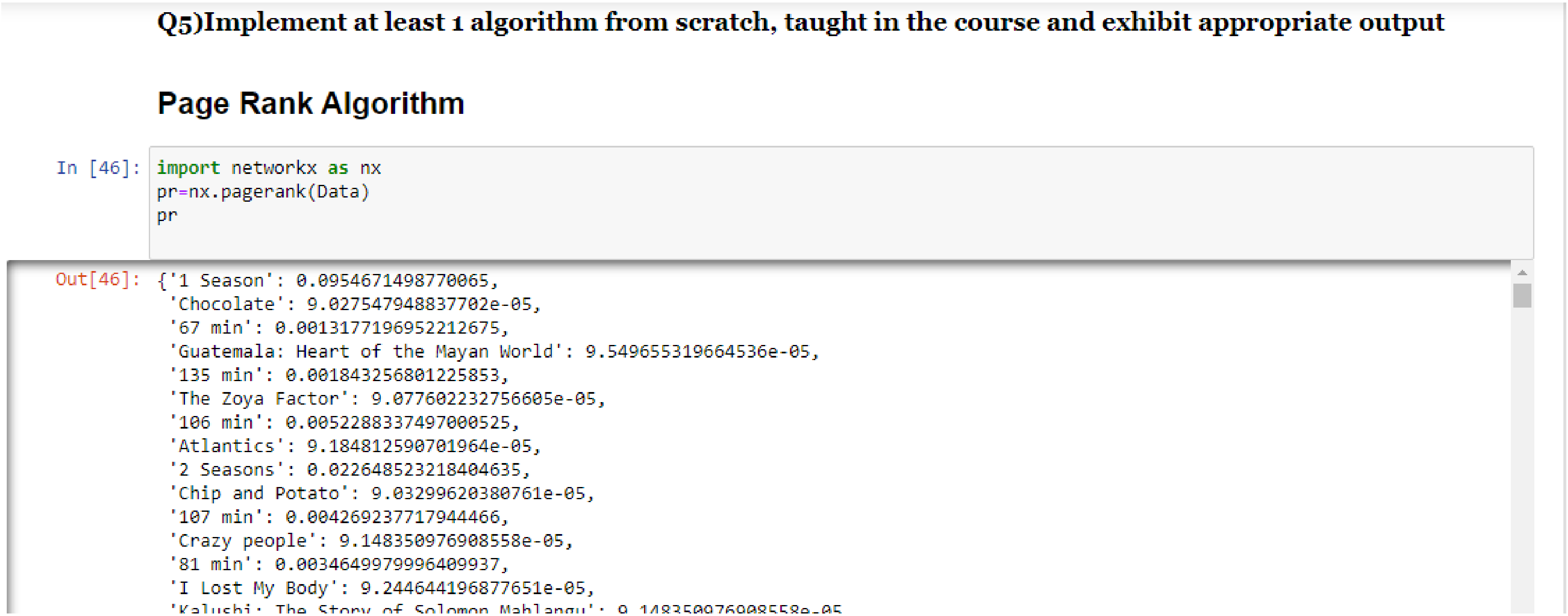
```
In [43]: nx.draw_networkx(connection_12season)
         plt.figure(figsize=(10,10))
         plt.show
```

```
Out[43]: <function matplotlib.pyplot.show(close=None, block=None)>
```

```
<Figure size 720x720 with 0 Axes>
```

What algorithm is suitable for this dataset?

Page Rank algorithm can suggest me the best duration with many movies have run. Most movies of Netflix have run for 1 Season compared to all other duration. Therefore, Netflix can consider those types of movies more as its trending and demanding to audience.

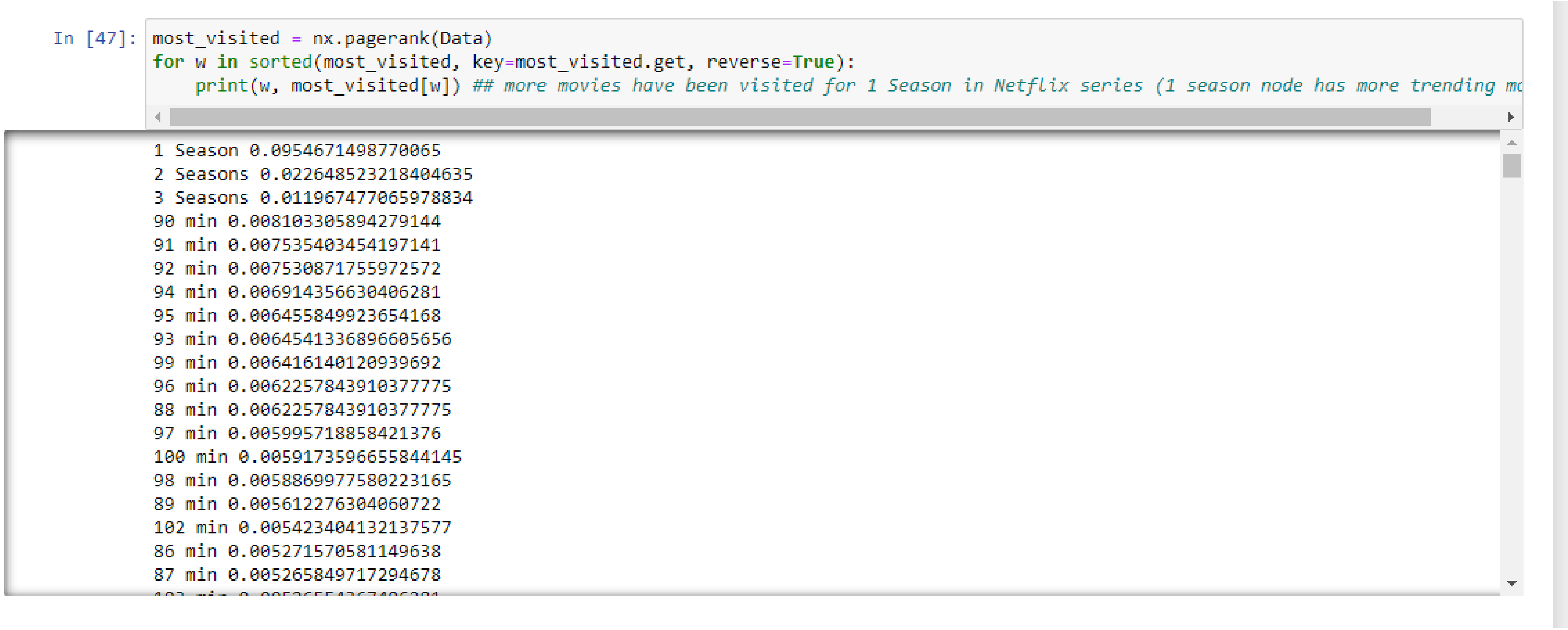


Q5)Implement at least 1 algorithm from scratch, taught in the course and exhibit appropriate output

Page Rank Algorithm

```
In [46]: import networkx as nx
         pr=nx.pagerank(Data)
         pr
```

```
Out[46]: {'1 Season': 0.0954671498770065,
          'Chocolate': 9.027547948837702e-05,
          '67 min': 0.0013177196952212675,
          'Guatemala: Heart of the Mayan World': 9.549655319664536e-05,
          '135 min': 0.001843256801225853,
          'The Zoya Factor': 9.077602232756605e-05,
          '106 min': 0.005228833749700525,
          'Atlantics': 9.184812590701964e-05,
          '2 Seasons': 0.022648523218404635,
          'Chip and Potato': 9.03299620380761e-05,
          '107 min': 0.004269237717944466,
          'Crazy people': 9.148350976908558e-05,
          '81 min': 0.0034649979996409937,
          'I Lost My Body': 9.244644196877651e-05,
          'Kalushi: The Story of Solomon Mahlangu': 9.148350976908558e-05
```



```
In [47]: most_visited = nx.pagerank(Data)
         for w in sorted(most_visited, key=most_visited.get, reverse=True):
             print(w, most_visited[w]) ## more movies have been visited for 1 Season in Netflix series (1 season node has more trending mo
```

```
1 Season 0.0954671498770065
2 Seasons 0.022648523218404635
3 Seasons 0.011967477065978834
90 min 0.008103305894279144
91 min 0.007535403454197141
92 min 0.007530871755972572
94 min 0.006914356630406281
95 min 0.006455849923654168
93 min 0.0064541336896605656
99 min 0.006416140120939692
96 min 0.0062257843910377775
88 min 0.0062257843910377775
97 min 0.005995718858421376
100 min 0.0059173596655844145
98 min 0.0058869977580223165
89 min 0.005612276304060722
102 min 0.0054234041321375 77
86 min 0.005271570581149638
87 min 0.005265849717294678
```

I have noticed there have been more movies played for 1 Season. I would like to know what type of movies was played on 1 Season.

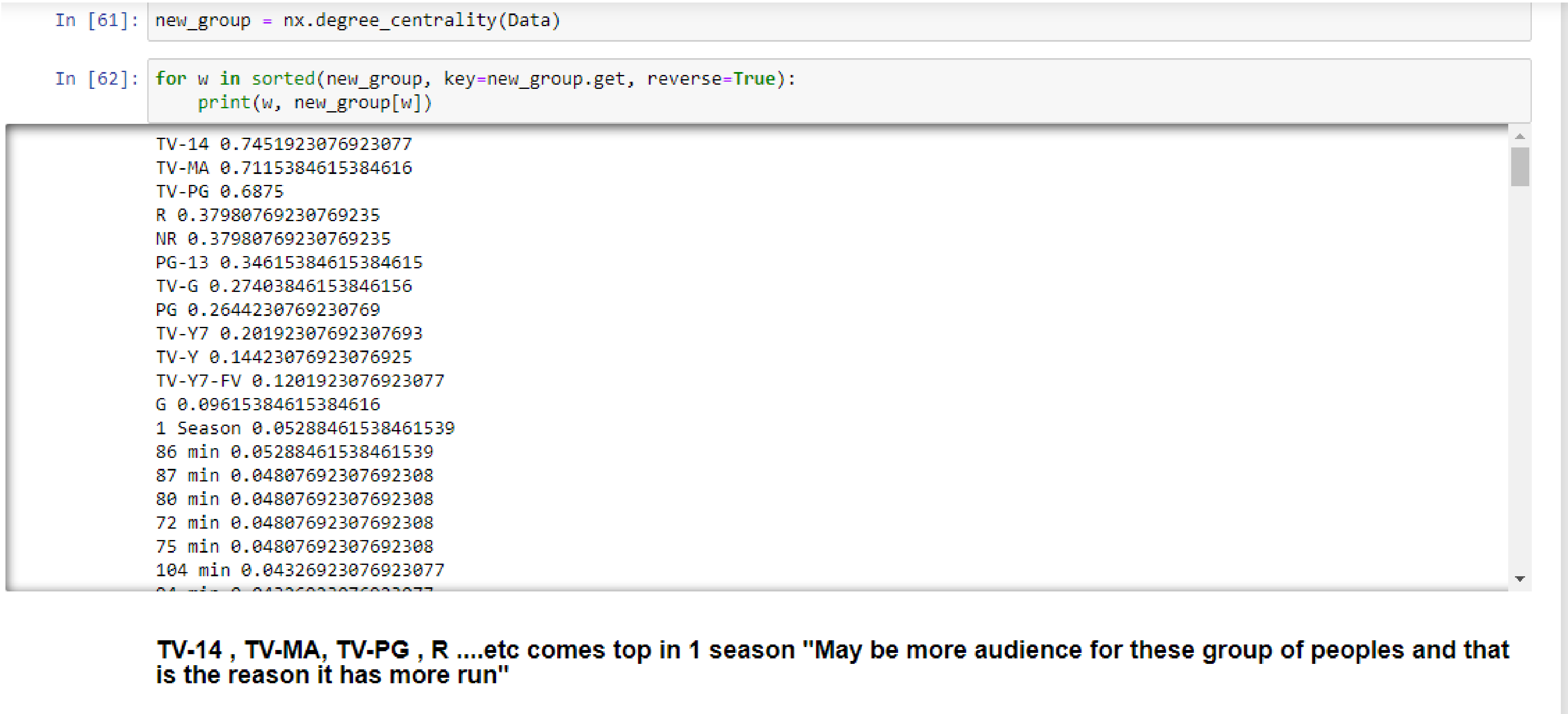
```
In [48]: df1 = pd.read_csv("netflix.csv")
```

```
In [49]: Data = nx.from_pandas_edgelist(df,source="duration",target="rating")
```

```
In [50]: new_group = nx.bfs_tree(Data,'1 Season')
```

```
In [51]: new_group.nodes
```

```
Out[51]: NodeView(('1 Season', 'TV-14', 'TV-PG', 'TV-MA', 'TV-Y7-FV', 'TV-Y7', 'TV-G', 'TV-Y', 'R', 'NR', 'G', nan, '135 min', '106 mi
         n', '107 min', '2 Seasons', '118 min', '104 min', '93 min', '134 min', '121 min', '56 min', '119 min', '84 min', '105 min', '10
         0 min', '46 min', '3 Seasons', '78 min', '4 Seasons', '108 min', '55 min', '116 min', '95 min', '103 min', '97 min', '89 min',
         '126 min', '14 min', '94 min', '92 min', '87 min', '99 min', '112 min', '127 min', '62 min', '5 Seasons', '101 min', '110 min',
         '86 min', '82 min', '133 min', '146 min', '8 Seasons', '98 min', '111 min', '6 Seasons', '80 min', '124 min', '76 min', '157 mi
```



```
In [61]: new_group = nx.degree_centrality(Data)
```

```
In [62]: for w in sorted(new_group, key=new_group.get, reverse=True):
             print(w, new_group[w])
```

```
TV-14 0.7451923076923077
TV-MA 0.7115384615384616
TV-PG 0.6875
R 0.37980769230769235
NR 0.37980769230769235
PG-13 0.34615384615384615
TV-G 0.27403846153846156
PG 0.2644230769230769
TV-Y7 0.20192307692307693
TV-Y 0.14423076923076925
TV-Y7-FV 0.1201923076923077
G 0.09615384615384616
1 Season 0.05288461538461539
86 min 0.05288461538461539
87 min 0.04807692307692308
80 min 0.04807692307692308
72 min 0.04807692307692308
75 min 0.04807692307692308
104 min 0.04326923076923077
```

TV-14 , TV-MA, TV-PG , R ....etc comes top in 1 season "May be more audience for these group of peoples and that is the reason it has more run"

Therefore I would suggest Netflix to select types of shows based on more run and more demanding for Audience like Season 1 to Season 15 preferably.

Thankyou!!