Best Actor Detection

Master of Science Project Report – Fall 2022

Dhruval Patel

Department of Computer Science

Committee Members

Dr. Murray Patterson (Advisor)

Dr. Zhipeng Cai

1. INTRODUCTION

Everyone loves watching movies these days. If we talk about what makes a movie successful, then we would think of a compelling story line, a well written script, talented editors, photographers, videographers, etc. People will also watch a movie if it is a particular genre but something that is even more important is the actor. Almost everyone even has a favorite actor and/or actress. Some people will watch a movie if it has their factor actor but no matter how the script is. For example, Dwayne Johnson, Will Smith, Tom Holland, Jennifer Lawrence, etc. are very popular actors and people will go watch a movie as long as one of these actors are playing a role in the movie. That being said, you may have seen your favorite actor in many movies and maybe that is why they are your favorite. But the real question is, are all the movies that you love, and your favorite actor connected somehow? Well in this project I will explore the connection between the movies and the actors using different graph theories and see if there are any connections.

2. MOTIVATION

There is a lot of varieties of information regarding my project's goal that I can utilize to answer the question. That being said, I will use two main factors: Actors and Movies. Actors will contain information like a list of movies that the particular actor has worked in, their gender, their height, their age, etc. Movies will contain information like its box office revenue that was generated, the language that it was filmed in, the country that the movie was released in, the genre of the movie, the release date, etc. These factors and the characteristics are just some ideas that I intend to have but it may vary depending upon the datasets. However, I will utilize this information to predict an actor who might make a successful movie. I will use centrality similarity measures (Degree centrality, Closeness centrality, Betweenness centrality) to discover this finding. Furthermore, I will find a popular pair of actors who will make a successful movie. For this, I will be using Jaccard Coefficient similarity measure. And last but not least, I will answer the following question at the end of my project: "Does working in a lot of movies make you the best actor or result in a successful movie?"

3. DATASETS

Initially, I started this project with only one dataset, but it only contained movies and the list of actors who worked in it which was not sufficient information to validate the findings of this project therefore I had to utilize a secondary dataset which contains more information.

3.1 DATASET 1

Like I mentioned above, this dataset contains movie and a list of actors who worked in the movie. The original un-processed data contains 1,288 movie information and there is a total of 21,143 actors' information. The statistic of this dataset is not a lot and therefore this is an another reason why it was very important to use a secondary dataset.

3.2 DATASET 2

The second dataset that I used in this project contained a lot of information but the main categories were Movie metadata, Character metadata and Plot summary. Movie Metadata contains Wikipedia movie ID, freebase movie ID, movie name, movie release date, movie box office revenue, movie runtime, movie languages, movie countries and movie genres. Character Metadata contains

Wikipedia, freebase movie ID, movie release date, character name, actor date of birth, actor gender, actor height, actor ethnicity, actor name, actor age at movie release, freebase character/actor map ID, freebase character ID and freebase actor ID. Plot summary contains a summary for each movie. As we can see that Movie metadata and Character metadata are the most important files as it contains a lot of valuable information. I basically used Wikipedia movie ID to keep both files connected as other attributes had missing values. This was an important step because I will be utilizing information from both files together. This dataset was also used to write scripts for validation purpose which is mentioned below in the report.

4. PRE-PROCESSING

The main part of the pre-processing step was to convert the dataset which is in a text file format to a graph format. I created two graphs for each dataset.

For dataset #1, I started by considering actors as nodes and edges as common movie between two nodes. In other words, if we have a movie and a list of actors who have worked in that movie then all the actors in the list get connected with each other forming a clique. Also, an actor can work in multiple movies which means that there can be multiple nodes which represent the same actor. In this step I made sure that all the nodes in my graphs were unique. However, my goal is to construct an undirected weighted graph which means that I need to have some attribute which represent the weights of the edges. Therefore, I used total common movies between two actors/nodes as their weight.

For dataset #2, I wrote a script called movie-actor.py where I extracted Wikipedia movie ID and a list of actors who worked in that movie from Character metadata file. The movie ID and actors were stored in a dictionary where Wikipedia movie ID is the key and the list of actors who worked in that movie are the value. After the creation of this dictionary, I wrote it onto movie-actors.tsv. After that, I opened this file in my notebook where I converted it in a graph format just like how I did with dataset #1. Started by considering actors as nodes and edges as common number of movies between two nodes.

After the formation of my graphs for both datasets, I achieved 21,143 nodes and 1,453,761 edges for dataset #1 and 2,082,113 nodes and 133,414 edges for dataset #2. These numbers are high and low at the same time depending upon the application that we want to use it for therefore I utilized a threshold which is mentioned below in this report.

5. ALGORITHMS

5.1 CENTRALITY

In graph theory, indicators of centrality assign numbers or rankings to nodes within a graph corresponding to their network position. They generally reflect a unit's prominence, in different substantive settings, this may be its structural power, status, prestige, or visibility. Studies often use network-based centrality measures in efforts to account for interunit differences in behavior or attitudes. I will be exploring three main centrality algorithms: Degree centrality, closeness centrality and betweenness centrality. These algorithms use graph theory to calculate the importance of any given node in a network. They cut through noisy data, revealing parts of the network that need attention – but they all work differently. Each measure has its own definition of 'importance', so you need to understand how they work to find the best one for your graph visualization applications.

5.1.1 DEGREE CENTRALITY

Degree Centrality is one of the easiest to calculate. The degree centrality of a node is simply its degree. In other words, it is the number of edges it has. The higher the degree or edges, the more central node it has. This can be an effective measure since many nodes with high degrees also have high centrality by other measures. The degree centrality of a vertex is a normalized value representing the number of edges touching a vertex. For a fracture in a network, it is an indicator of the number of fractures that intersect it. Vertices with low degree centrality will usually be on the periphery of the network or the low flow branches. For a vertex i, its degree centrality is given by

$$D(i) = \frac{1}{n-1}\sum_{j=1}^{n} A_{ij}$$

where A_{ij} is the ij-th element of the adjacency matrix A of the graph and n is the number of vertices in the graph. Example: We have an undirected graph (Figure 1) where we have total of 8 nodes. We can find degree centrality of each node using above equation and we get following result.

$$D(A) = \frac{1}{7} \approx 0.14$$
$$D(B) = \frac{2}{7} \approx 0.29$$
$$D(C) = \frac{2}{7} \approx 0.29$$
$$D(D) = \frac{3}{7} \approx 0.43$$
$$D(E) = \frac{4}{7} \approx 0.57$$
$$D(F) = \frac{2}{7} \approx 0.29$$
$$D(G) = \frac{3}{7} \approx 0.43$$
$$D(H) = \frac{3}{7} \approx 0.43$$

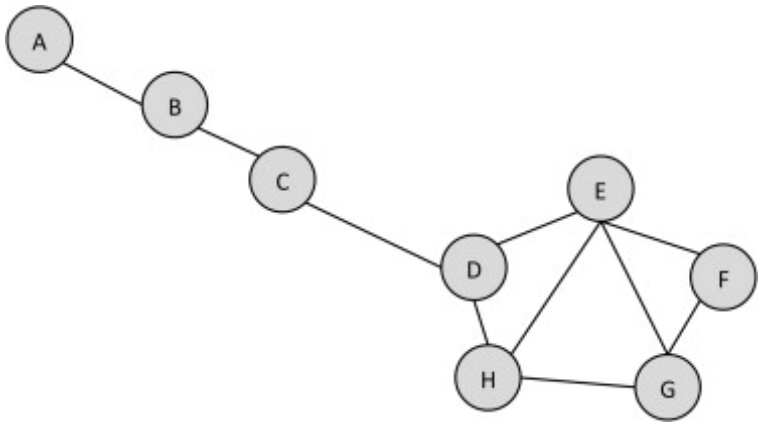

Figure 1

We can see that Node E has the highest normalized degree centrality value of 0.57 which tells us that Node E is a node that has a larger than average number of connections for the given graph.

5.1.2 CLOSENESS CENTRALITY

Closeness centrality indicates how close a node is to all other nodes in the network. It is calculated as the average of the shortest path length from the node to every other node in the network. Let's consider Figure 1 again. We can start computing the average shortest path length of node D. Table 1 shoes each node and the length of the shortest path from D.

Node	Shortest path from node D
A	3 (D-C-B-A)
B	2 (D-C-B)
C	1 (D-C)
E	1 (D-E)

F	2 (D-E-F)
G	2 (D-H-G)
H	1 (D-H)

Table 1

The average of these shortest path length is 1.71. We can repeat the exact same for node A.

Node	Shortest path from node A
B	1 (A-B)
C	2 (A-B-C)
D	3 (A-B-C-D)
E	4 (A-B-C-D-E)
F	5 (A-B-C-D-E-F)
G	5 (A-B-C-D-E-G)
H	4 (A-B-C-D-H)

Table 2

The average of these shortest path length is 3.43. We can do these for all other nodes and get different averages. However, we can visually see that node D might be the best optimal solution and we can also verify this by the average that we got from the Table 1 which is 1.71. Since our graph only has 8 total nodes, it is not too difficult to calculate and have a realistic comparable closeness centrality value but if we have a graph where the number of nodes is very high then we need to implement a different way of finding closeness centrality. We can do this by calculating the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph. We can represent this as

$$C(x) = \frac{1}{\sum_y d(x, y)}$$

Here, d(x, y) is the shortest distance between node x and node y. In other words, this is a normalized closeness centrality. We can apply this to figure D as well. So, if we calculate normalized closeness centrality of Node D from Figure 1, then we get $C(D) = \frac{1}{12} \approx 0.083$ which is the highest value compared to other nodes.

5.1.3 BETWEENNESS CENTRALITY

Betweenness centrality is a measure of centrality in a graph based on shortest paths. In other words, we can say that it is a widely used measure that captures a person's role in allowing information to pass from one part of the network to the other. For every pair of vertices in a connected graph, there exists at least one shortest path between the vertices such that either the number of edges that the path passes through (for unweighted graphs) or the sum of the weights of the edges (for weighted graphs) is minimized. The betweenness centrality for each vertex is the number of these shortest paths that pass through the vertex. Betweenness centrality finds wide application in network theory: it represents the degree of which nodes stand between each other. For example, in a telecommunications network, a node with higher betweenness centrality would have more control over the network, because more information will pass through that node. Betweenness centrality was devised as a general measure of centrality: it applies to a wide range of problems in network theory, including problems related to social networks, biology, transport, and scientific cooperation. We can calculate centrality of a node v in a graph as

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v. However, this expression will not be the best solution in all graphs because there is no specified range. So, if we have a graph with a very high number of nodes and edges then this expression will give us values that will not help us in discovering our findings. However, we can restrict the value to have it between 0 and 1 using normalized betweenness centrality. We can do that by

$$c(v) = \frac{p(v)}{(n-1)(n-2)/2}$$

where p(v) is the popularity of a node v (In other words, how many shortest paths passes through the node v) and n is a number of nodes in a given graph. Let's consider Figure 2 as an example. To find betweenness centrality of this graph, we will first list all the unique pairs where there exist a path. So, in this case we have following pairs: {(a,b), (a,c), (a,d), (b,c), (b,d), (c,d)}. Next step is to find shortest path for each pair.

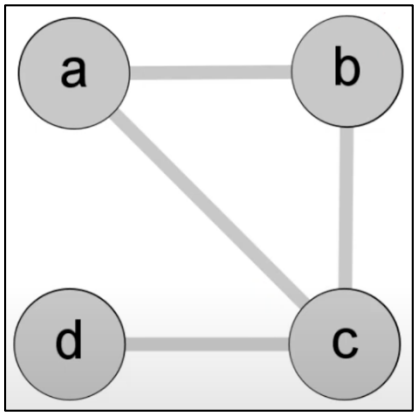

Figure 2

Pair		Shortest Path	a	b	c	d
a	b	(a,b)	0	0	0	0
a	c	(a,c)	0	0	0	0
a	d	(a,c,d)	0	0	1	0
b	c	(b,c)	0	0	0	0
b	d	(b,c,d)	0	0	1	0
c	d	(c,d)	0	0	0	0

Table 3

Table 3 represent shortest path for each pair and popularity of each node. Basically, we put 1 in a, b, c and d if a specific pair is passing through one of those nodes. Once we have all the values in the table, we sum it up. From Table 3, we can see that node a, b and d have sum of 0 and c have a sum of 2. Now we can calculate normalized betweenness centrality for each node and we get 0 for a, b and d. However, for c we get 2/3 which is greater than 0 so we can say that in this case node c has highest betweenness centrality.

5.2 JACCARD COEFFECIENT

The Jaccard Coefficient which is also known as Jaccard's Similarity Index compares members for two sets to see which members are shared and which are distinct. It's a measure of similarity for the two sets of data, with a range from 0% to 100%. The higher the percentage, the more similar the two populations. Although it's easy to interpret, it is extremely sensitive to small samples sizes and may give erroneous results, especially with very small samples or data sets with missing observations.

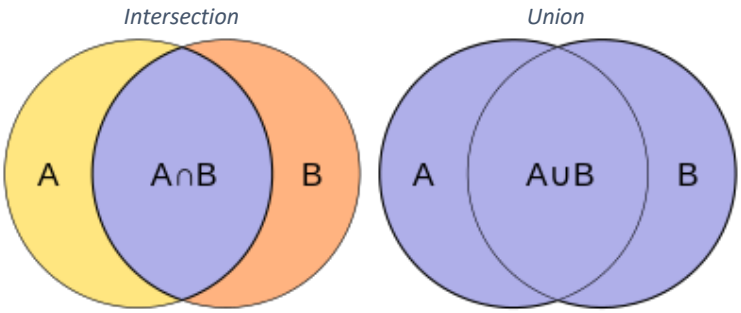

Figure 3

The formula to calculate Jaccard Coefficient is the number in both sets over the number in either set times 100. In other words, it is intersection over union like we can see in Figure 3. Let's assume that we have following sets. A = {0, 1, 2, 5, 6} and B = {0, 2, 3, 4, 5, 7, 9}. To calculate we will take the intersection/union which is

$$J(A \cap B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|0, 2, 5|}{|0, 1, 2, 3, 4, 5, 6, 7, 9|} = \frac{3}{9} \approx 0.33$$

6. THRESHOLD

The original dataset #1 contains 21,143 nodes and 1,453,761 edges and the original dataset #2 contains 2,082,113 nodes and 133,414 edges. These numbers can be good and bad at the same time. If we are wanting to plot the graph on this original data, then it is visually impossible because imagine having 21,142 nodes on the screen. All the nodes and edges will get overlapped which will basically mean nothing in terms to extracting information visually. Another reason why setting a threshold is important is because of computational time. Running centrality algorithms and Jaccard coefficient on graphs with less than 300 nodes is reasonable and can be ran within just few seconds, however if we have more than few thousand nodes then the computation time is few hours. Furthermore, the dataset has a lot of outliers. These outliers are basically unpopular actors who have worked only in a couple of movies or if that. Basically, running the algorithms with this outlier nodes is not informative to us therefore threshold is needed.

I created two thresholds for each dataset. One for visualization and one for computing algorithms. I utilized the threshold by extracting a subgraph from the original graph depending upon the criteria. The thresholds that I used are as follow:

- Dataset #1
  - Visualization: Edge weight > 5 and Node degree > 2
    - Subgraph: 125 edges and 51 nodes
  - Algorithm computation: Edge weight > 2 and Node degree > 2
    - Subgraph: 8,432 edges and 1,519 nodes
- Dataset #2
  - Visualization: Edge weight > 10 and Node degree > 10
    - Subgraph: 157 edges and 54 nodes
  - Algorithm computation: Edge weight > 2 and Node degree > 2
    - Subgraph: 10,552 edges and 2,742 nodes

As you can see that each dataset has different threshold but the subgraph that it created is similar. This was important because 50-55 nodes was the best spot for visualizing the graph and 1500-3000 nodes was the perfect number of nodes in correspond with the number of edges for computing the algorithms. Lower threshold resulted in longer computational time and higher threshold resulted in loss of important nodes and edges.

7. EXPERIMENTAL VALUES

I created a degree distribution plot before and after a threshold. Figure 4a represents the Dataset #1 and figure 4b represents Dataset #2. From figure 4a we can see that most of the nodes have a degree of 0-100, however, there is also a peek between 1250-1350 which can mean multiple things for example, a possibility of a very huge clique or just duplicates in the dataset but it is hard to justify that. If we look at overall graph, then we can also see the peek as an outlier since it is very distinguished from other degree frequency. Figure 4b, does not have any distinguished peek like figure 4a which means that this dataset might not have any duplicates or a very huge clique.

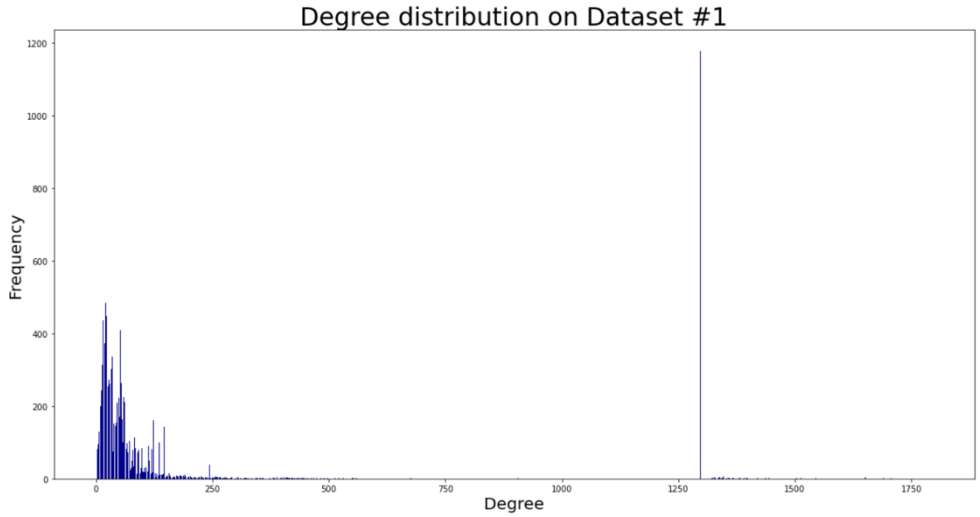

Figure 4a

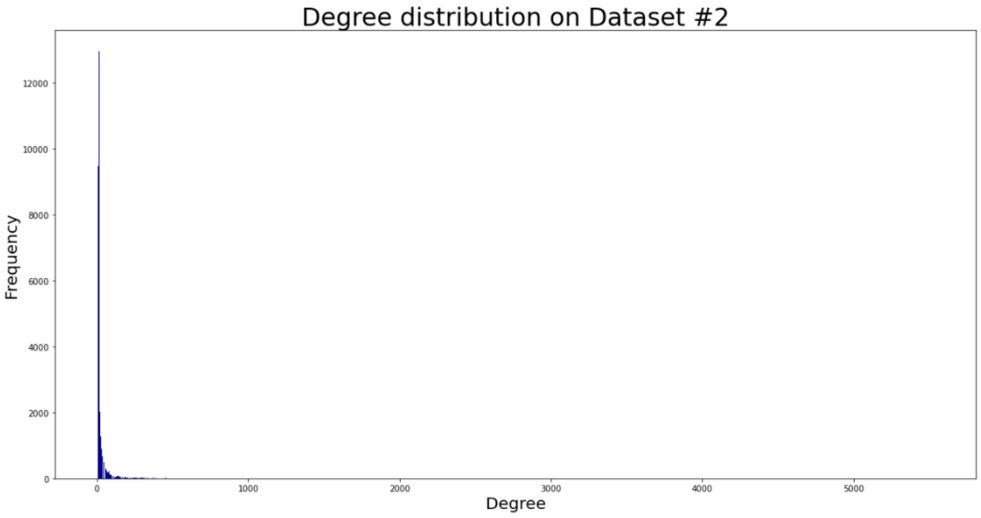

Figure 4b

Using the threshold that was specifically made for visualization, I plotted two main graphs. A degree distribution graph and a graph representation of our dataset. Figure 5a represents degree distribution for dataset #1. Here you can see that there are 0 nodes in the graph whose degree is 0 which means that there are no disconnected nodes in the subgraph. We can also see that the highest frequency is of degree 2 following with degree of 7. We can say the same for figure 5b which is the degree distribution for dataset 2. Here, we can see that there are at least 5 nodes that are completely disconnected which we can see in figure 7b. However, for degree distribution, we can also see that the frequency of higher degree is also high. For example, there are at least 9 nodes whose degree is 6 and at least 8 nodes whose degree is 7. Following are some statistics for each dataset after the threshold.

- Dataset #1:
    - Mean of Degree: 6.5
    - Standard Deviation Degree: 4.18
    - Mean of Frequency: 3.64
    - Standard Deviation of Frequency: 4.62
- Dataset #2
    - Mean of Degree: 5.5
    - Standard Deviation Degree: 3.61
    - Mean of Frequency: 4.5
    - Standard Deviation of Frequency: 2.65

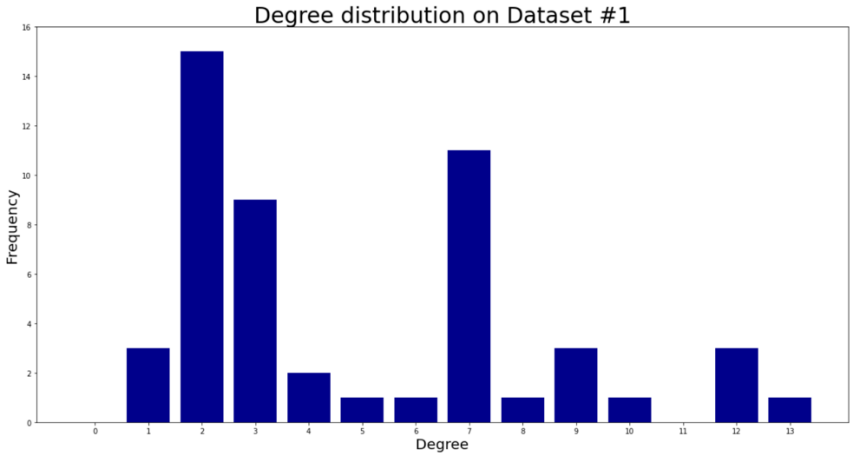

Figure 5a

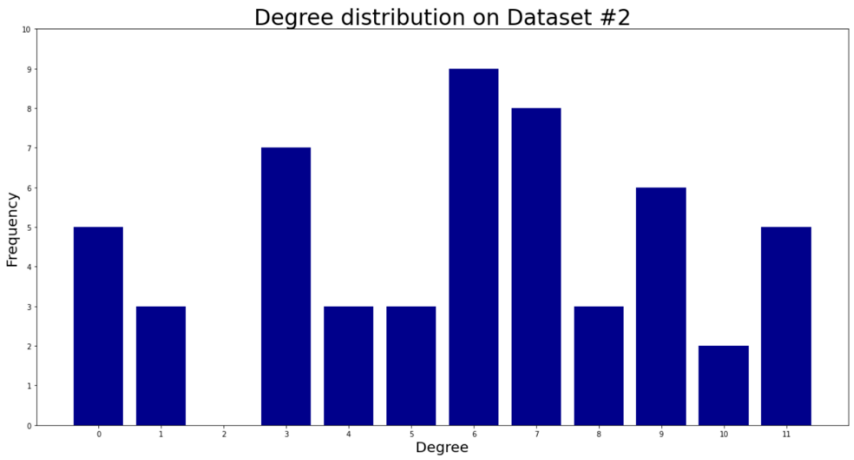

Figure 5b

Threshold: Edge weight > 10 and node degree > 10

	Degree	Degree Centrality	Closeness Centrality	Betweenness Centrality
Jagathi Sreekumar	11	0.207547	0.207547	0.002164
Nedumudi Venu	11	0.207547	0.207547	0.002164
Shakti Kapoor	11	0.207547	0.237080	0.009965
Asrani	11	0.207547	0.237080	0.014033
Prem Chopra	11	0.207547	0.237080	0.008483
Thilakan	10	0.188679	0.190252	0.000713
Mammootty	10	0.188679	0.190252	0.001611
Mohanlal	9	0.169811	0.175617	0.000402
Innocent Vincent	9	0.169811	0.175617	0.000402
Siddique	9	0.169811	0.175617	0.001279
Amitabh Bachchan	9	0.169811	0.218113	0.008175
Mukesh	9	0.169811	0.175617	0.000402
Mithun Chakraborty	9	0.169811	0.218113	0.005426
Dharmendra Deol	8	0.150943	0.201957	0.003698
Sreenivasan	8	0.150943	0.163073	0.000311
Prakash Raj	8	0.150943	0.157233	0.010643
Suresh Gopi	7	0.132075	0.152201	0.000311
Paresh Rawal	7	0.132075	0.201957	0.002721
Aruna Irani	7	0.132075	0.201957	0.001978
Brahmanandam	7	0.132075	0.145138	0.004233
Jayaram	7	0.132075	0.152201	0.000402
Anupam Kher	7	0.132075	0.188029	0.002501
Hema Malini	7	0.132075	0.188029	0.002334
Kader Khan	7	0.132075	0.201957	0.001963
M. S. Narayana	6	0.113208	0.117925	0.000000

Figure 6a

Threshold: Edge weight > 4 and node degree > 2

	Degree	Degree Centrality	Closeness Centrality	Betweenness Centrality
Jr.	127	0.046333	0.148586	0.177189
Mithun Chakraborty	115	0.041955	0.116797	0.007219
Shakti Kapoor	108	0.039402	0.118201	0.004430
Dharmendra Deol	106	0.038672	0.113250	0.006198
Amitabh Bachchan	103	0.037578	0.112193	0.004549
Prakash Raj	101	0.036848	0.112309	0.008109
Brahmanandam	97	0.035389	0.101078	0.004412
Nassar	95	0.034659	0.107509	0.003919
Amrish Puri	90	0.032835	0.116232	0.003387
Anupam Kher	87	0.031740	0.118525	0.006534
Mammootty	86	0.031375	0.095627	0.003200
Paresh Rawal	79	0.028822	0.110249	0.003453
Madan Puri	79	0.028822	0.097217	0.001240
Pran Sikhand	78	0.028457	0.103844	0.002732
Asrani	78	0.028457	0.111155	0.001950
Sanjay Dutt	78	0.028457	0.108643	0.002695
Prem Chopra	77	0.028092	0.107296	0.001277
Mohanlal	76	0.027727	0.106503	0.008224
Jagathi Sreekumar	74	0.026997	0.086662	0.000952
Frank Welker	71	0.025903	0.126867	0.037026
Kader Khan	68	0.024808	0.113488	0.001238
Moe Howard	68	0.024808	0.087478	0.023268
Larry Fine	67	0.024444	0.076454	0.001147
Aruna Irani	67	0.024444	0.104547	0.000929
Rajpal Yadav	63	0.022984	0.101744	0.002785

Figure 6b

Threshold: Edge weight > 5 and node degree > 2

	Degree	Degree Centrality	Closeness Centrality	Betweenness Centrality
Angel, Jack (I)	13	0.26	0.275238	0.027887
Farmer, Bill (I)	12	0.24	0.240833	0.016105
McGowan, Mickie	12	0.24	0.251304	0.006404
Lynn, Sherry (I)	12	0.24	0.251304	0.006404
Proctor, Phil	10	0.20	0.222308	0.002282
Bumpass, Rodger	9	0.18	0.206429	0.000350
Rabson, Jan	9	0.18	0.214074	0.001469
Bergen, Bob	9	0.18	0.206429	0.000350
Darling, Jennifer	8	0.16	0.206429	0.001197
Blaustein, Madeleine	7	0.14	0.140000	0.000000
Derryberry, Debi	7	0.14	0.192667	0.000000
Lillis, Rachael	7	0.14	0.140000	0.000000
Taylor, Veronica (I)	7	0.14	0.140000	0.000000
Gates, Ken (I)	7	0.14	0.140000	0.000000
Stuart, Eric (III)	7	0.14	0.140000	0.000000
Ratzenberger, John	7	0.14	0.199310	0.006122
Ootani, Ikue	7	0.14	0.140000	0.000000
Rogers, Kayzie	7	0.14	0.140000	0.000000
Jayne, Tara	7	0.14	0.140000	0.000000
Ranft, Joe	7	0.14	0.199310	0.006122
Pinney, Patrick	6	0.12	0.192667	0.011170
Sahara, Kenji	5	0.10	0.100000	0.002449
Tajima, Yoshifumi	4	0.08	0.083333	0.000816
Nakajima, Haruo	4	0.08	0.083333	0.000816
Suzuki, Kazuo (I)	3	0.06	0.071429	0.000000

Figure 6c

Threshold: Edge weight > 2 and node degree > 2

	Degree	Degree Centrality	Closeness Centrality	Betweenness Centrality
Welker, Frank	97	0.063900	0.256643	0.135131
Rooney, Mickey (I)	84	0.055336	0.237553	0.052265
Reynolds, Debbie (I)	73	0.048090	0.219408	0.025364
Kelly, Gene (I)	73	0.048090	0.211659	0.003266
Proctor, Phil	70	0.046113	0.219259	0.010948
Garland, Judy (I)	68	0.044796	0.205521	0.001676
Charisse, Cyd	67	0.044137	0.205477	0.001110
Astaire, Fred	67	0.044137	0.219408	0.010322
Miller, Ann (I)	66	0.043478	0.205434	0.001004
Williams, Esther (I)	66	0.043478	0.205434	0.001004
Powell, Eleanor (I)	66	0.043478	0.205434	0.001004
Lynn, Sherry (I)	66	0.043478	0.222628	0.012749
Crawford, Joan (I)	66	0.043478	0.205434	0.001004
Angel, Jack (I)	66	0.043478	0.230722	0.029475
McGowan, Mickie	64	0.042161	0.220304	0.005112
Grant, Cary	61	0.040184	0.205260	0.005618
O'Connor, Donald	61	0.040184	0.204913	0.001955
Farmer, Bill (I)	59	0.038867	0.219259	0.004732
Sinatra, Frank	58	0.038208	0.205390	0.003977
Mann, Danny (I)	57	0.037549	0.227799	0.044339
Derryberry, Debi	56	0.036891	0.219061	0.010231
Caron, Leslie	55	0.036232	0.198788	0.001470
Stewart, James (I)	55	0.036232	0.202140	0.000689
Gable, Clark	55	0.036232	0.199196	0.001472
Flynn, Joe (I)	53	0.034914	0.236396	0.016576

Figure 6d

I mentioned above in *Section 6 – Threshold* that I have two thresholds for each dataset, one for visualization and one for algorithm computation. However, I still ran the algorithms on both thresholds to see if the output will vary. Sure enough, it did. On the higher threshold, I received a lot of centrality values of 0 for both datasets as you can see in figure 6a and 6c. However, the lower threshold which was specifically is set for the computation of algorithms did return reasonable values without any 0s. From figure 6b, we can see that Frank Welker has the highest degree, closeness and betweenness centrality along with highest degree in the subgraph. This tells us that Frank Welker is the best actor in dataset 1. We can use the same concept in dataset 2 where Jr. has the highest centrality measures and degree making Jr. the best actor in dataset 2. However, does this really validate that this actor is really the best? To answer this, I created a script which is mentioned below.

Centrality algorithm does help in finding the best actor, but I wanted to also find pairs that can make a good movie. Therefore, I utilized Jaccard Coefficient and found the pairs with the highest value. You can see the top 25 best pairs of dataset 1 in figure 7a and best pairs of dataset 2 in figure 7b. The best

Threshold: Edge weight > 2 and node degree > 2

	Common Neighbor	Jaccard Coefficient
(Court, Alyson, Goy, Luba)	5	1.0
(Hayes, John (II), Brodhead, James)	3	1.0
(Matsushita, Hiromi (I), Furuta, Toshihiko)	2	1.0
(Medwin, Michael, Owens, John (I))	2	1.0
(Jenson, Roy (I), Morgan, Bob (I))	2	1.0
(Saijo, Yasuhiko, Saito, Noritake)	2	1.0
(Lloyd, Christopher (I), Carroll, Pat (I))	2	1.0
(Lloyd, Christopher (I), Stewart, Patrick (I))	2	1.0
(Ward, Dervis, Lemkow, Tutte)	2	1.0
(Stewart, Patrick (I), Carroll, Pat (I))	2	1.0
(Gable, Christopher, Fleet, Stanley)	2	1.0
(Playten, Alice, Erdman, Richard (I))	1	1.0
(Quick, Lee, Hollingshead, Megan)	10	0.9
(Henson, John (II), May, Ed)	6	0.9
(Barnett, Robbie (I), Plaskitt, Nigel)	6	0.9
(Barnett, Robbie (I), Clarke, Marcus (II))	6	0.9
(Lewis, Ted (II), Ortiz, Lisa)	10	0.8
(Quick, Lee, Lewis, Ted (II))	9	0.8
(Williams, Kerry (II), Hart, Stan (I))	9	0.8
(Williams, Kerry (II), Kay, Roger (II))	9	0.8
(Williams, Kerry (II), Ortiz, Lisa)	9	0.8
(Williams, Kerry (II), Green, Dan (III))	9	0.8
(Walmsley, Jon, Reitherman, Bruce)	8	0.8
(Muehl, Brian, Payne, Bob (I))	7	0.8
(Prell, Karen, Brill, Fran)	5	0.8

Figure 7a

Threshold: Edge weight > 4 and node degree > 2

	Common Neighbor	Jaccard Coefficient
(Harry Semels, Bess Flowers)	5	1.0
(Marion Ramsey, G. W. Bailey)	5	1.0
(Maryke Hendrikse, Janyse Jaud)	5	1.0
(Yûko Minaguchi, Bin Shimada)	5	1.0
(Duke York, Ethelreda Leopold)	4	1.0
(Monte Collins, Phyllis Crane)	4	1.0
(Monte Collins, James C. Morton)	4	1.0
(Peggy Cartwright, Winston and Weston Doty)	4	1.0
(Peter Hüttner, Michael Nyqvist)	4	1.0
(Phyllis Crane, James C. Morton)	4	1.0
(Edgar Dearing, Lyle Tayo)	3	1.0
(Edgar Dearing, Baldwin Cooke)	3	1.0
(Edgar Dearing, Jimmy Finlayson)	3	1.0
(Geneva Mitchell, William Irving)	3	1.0
(Geneva Mitchell, Dorothy Vernon)	3	1.0
(Geneva Mitchell, June Gittelson)	3	1.0
(Geneva Mitchell, Theodore Lorch)	3	1.0
(Geneva Mitchell, Fred Kelsey)	3	1.0
(Kichijiro Ueda, Kamatari Fujiwara)	3	1.0
(Rosina Lawrence, Leonard Landy)	3	1.0
(Margaret Dumont, Zeppo Marx)	3	1.0
(Ashleigh Ball, Nicole Oliver)	3	1.0
(Ashleigh Ball, Alessandro Juliani)	3	1.0
(Margie Liszt, Phil Arnold)	3	1.0
(Margie Liszt, Jean Willes)	3	1.0

Figure 7b

pairs of actors in dataset 1 is Alyson Court and Luba Goy. Here their Jaccard Coefficient is 1.0 which means that these actors really make a good pair. Same concept is applied on dataset 2 where best pairs of actors is Harry Semels and Bess Flowers. Based on the figure 7b, all pairs have Jaccard Coefficient of 1.0 which just means that all these pairs are perfectly compatible with each other. Also, these are the top 25 best pairs so if we investigate top 50 or top 100 pairs then we will find pairs where the JC value is not 1.0.

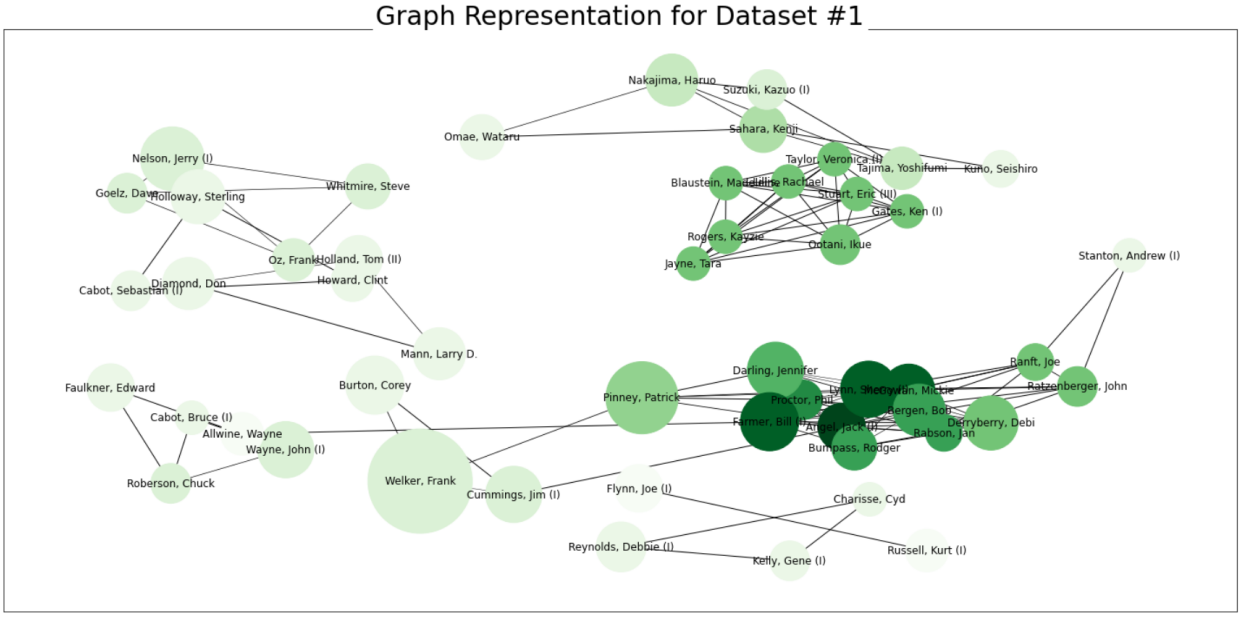

Figure 8a

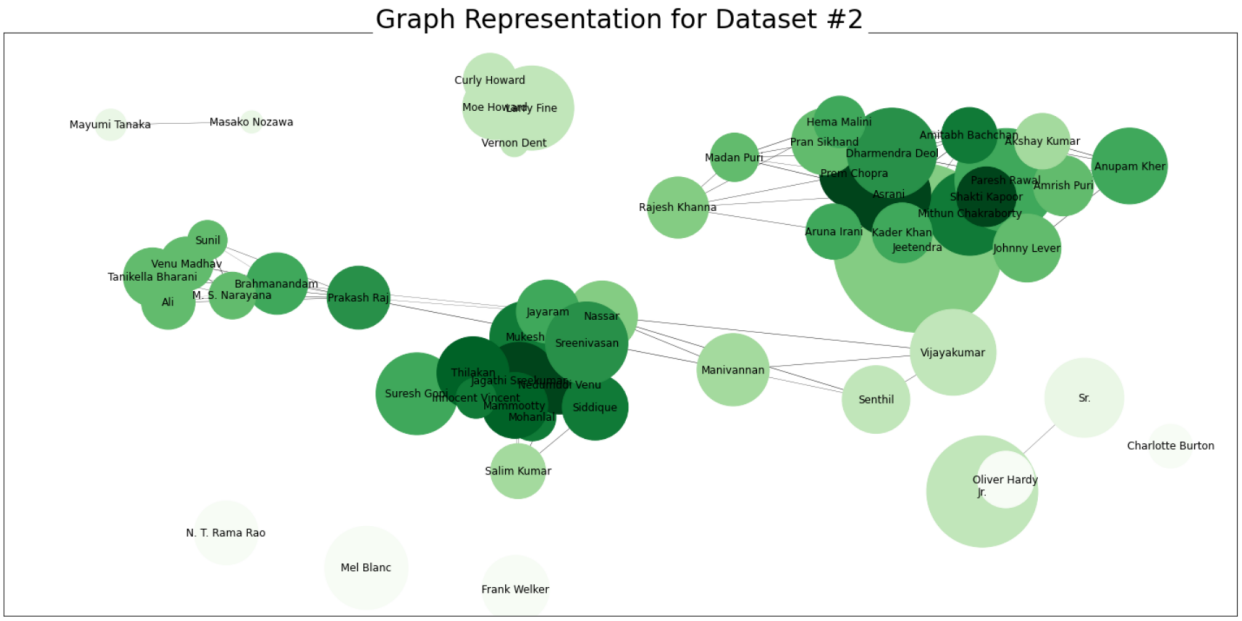

Figure 8b

Figure 8a and 8b are basically a graph representation of the subgraphs. The size of the node representants the number of movies that the actor has worked in, and the color of the node indicates the centrality value. Darker the node is means higher the centrality is. We can also see that there are at least 4 visible nodes in figure 8b that are completely disconnected. We can confirm this with figure 5b where we can see that there are 5 nodes with degree of 0.

8. SCRIPTS

The Centrality measures and Jaccard Coefficient similarity values were not enough to validate whether a particular actor is really successful or not therefore, I created two scripts which helped with the validation

8.1 ACTOR – GENDER

This script extracted actors and their genders from Character metadata file. The script basically created a tsv file which I utilized in the notebook for the evaluation. I discovered that there were 60,759 male actors and 35,657 female actors in my dataset 2. The main goal behind this script was to find whether a pair of male actors, a pair of female actors, or a pair of male and female actors work better. To do this, I used Jaccard Coefficient values of my dataset 2 and the gender information to find that there were 1,738,212 pairs of male actors, 330,313 pairs of female actors and 1,529,922 pairs of male and female actors.

The ratio of male to female actors is approximately 2 to 1 as we have half the size of female actors as male actors. That being said, if we want to check whether a male actor is better than a female actor then we can say that a number of pairs of male actors should be four time larger than a total number of pairs of female actors. If we consider male actors as m and female actors as n and since we are considering pairs in this case, we can represent it as $\frac{m*m}{n*n} = \frac{m^2}{n^2} = \frac{2^2}{1^2} = \frac{4}{1}$. This means that the ratio of pairs of male actors to pairs of female actors should be 4 to 1. However, if we multiply pairs of female

actors by 4 then we get 1,321,252 which tells us that the ratio in our case is 5 to 1. This tells us that male actors perform better than female actors.

8.2 MOVIE – REVENUE

This script extracted movies and the revenue that it generated from the Movie metadata file. Just like the actor-gender script, this script also created a tsv file which I utilized in my notebook for the evaluation. In my notebook, I extracted a dictionary where actors were the key and the list of movies that the actor has worked in is the value. I used this script to replace the movie with the revenue that it was able to generate. Once I had the dictionaries value replaced from movie to revenue, I summed it up. The final product was a dictionary where key is the actor, and the value is the revenue generated by this particular actor.

9. DATA VALIDATION

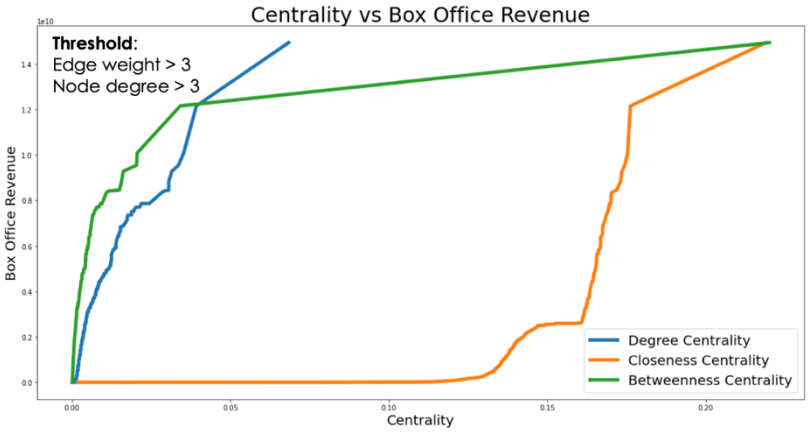

Figure 9

Can we justify that the centrality and Jaccard Coefficient reflects on the actor's success? Yes. In Figure 9 we can see that we have centrality values on x-axis and box office revenue on y-axis. We can see that there is a correlation between centrality and box office revenue which means that if an actors centrality measure is high then this actor is popular resulting in higher revenue generation for the movie that the actor will work in. From the Actor-Gender script, we can also say that we can use Jaccard Coefficient to find which pairs of actors work better.

SUMMARY

In this project, I utilized two datasets. Dataset 1 contains only actors and movies that the actor has worked in. Dataset 2 contains actor metadata and character metadata which consists of actors info such as their name, gender, ethnicity, etc and movie info such as movie name, revenue generated, language, etc. I used two similarity measures: Centrality (Degree, Closeness and Betweenness) and Jaccard Coefficient. For validation of my findings, I used box office revenue and the centrality measures and discovered a correlation which tells us that centrality value does help in detecting the best actor. Furthermore, I used Jaccard Coefficient to find that male actors tend to have a higher chance of making a movie successful.

A question that was mentioned in the motivation, "Does working in a lot of movies make you the best actor or result in a successful movie?". The answer to this question is NO. We can validate this from figure 8a. We can see that Jack (I) Angel has the highest centrality which we can also see in figure 6a. Although Frank Welker worked in a lot of movies, he still wasn't able to achieve the highest centrality which means that working in a lot of movies does not make you the best actor.

FUTURE WORK

The datasets that I used had few flaws. Dataset 1 was not big enough and Dataset 2 had a lot of missing values. It would be better to apply same concept on a dataset where there is more useful and less missing information. This will help in evaluating the findings. Also, this dataset contains actors that are either retired or passed away making the findings irrelevant. Therefore, a dataset with more info but with newer actors is needed. Although dataset 2 has a lot of missing values, different algorithms can be implemented on it. For example, Community detection: It is possible to find communities within a country, genres, languages, etc.

REFERENCES

[1] Bhasin, J. (2019) Graph analytics-introduction and concepts of centrality. Towards Data Science.

[2] Marsden, P.V., Golbeck, J. and Metcalf, L. Centrality measure, Centrality Measure - an overview.

[3] Hopkins, B., 2004. Kevin Bacon and Graph Theory.

[4] Heinold, B., 2019. A Simple Introduction to Graph Theory.

[5] Kimball, D. and Herdzik, E., n.d. Comparing IMDB Network of Actors to Random Graph Models. [online] Cs.rpi.edu.

[6] Betweenness Centrality (Centrality Measure) (2022) GeeksforGeeks.

[7] Bhasin, J. (2019) Graph analytics-introduction and concepts of centrality. Towards Data Science.

[8] CMU Movie Summary Corpus – Available at: http://www.cs.cmu.edu/~ark/personas/.