ECE 232E Project 4
IMDB DATA MINING

Cyrus Tabatabai-Yazdi (405029242)

Zongheng Ma (905027293)

Akshay Shetty (905028886)

Bakari Hassan (705035029)

4 June 2018

Table of Contents

1 Actor/Actress Network

For the first part, our goal was to explore the properties of actors and actresses by constructing a directed network in order to capture and analyze relationships.

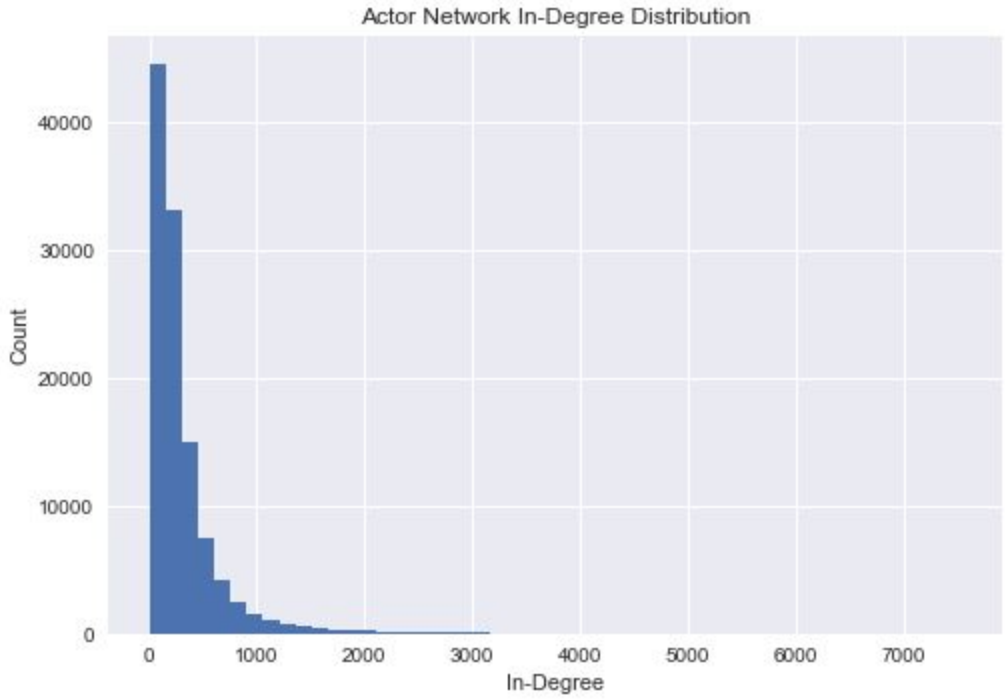
Question 1

The first question asked to preprocess the two text files: actor_movies.txt and actress_movies.txt. The approach we took was to use a regular expression to remove anything that appeared in parentheses. This is because, as stated in the project statement, there may be instances of the same movie with different qualifiers in the parentheses such as Movie X (voice) or Movie X (as uncredited). Not accounting for these discrepancies would lead to an overcount of the number of movies. However, we kept the year which appears in parentheses because there are numerous movies with the same title but different years. For example, there are classic movies and their remakes like *Beauty and the Beast (1991)* and *Beauty and the Beast (2017)*. These are two different movies and we adjusted our regular expression to account for this. In addition, close examination of the dataset revealed another quirk. In IMDB notation, (VG) stands for video game. We noticed that in the dataset, there were a few instances of objects with the same name and year but one was a regular movie while the other was a video game. For example, in the dataset, there is *Toy Story 2 (1999)* as well as *Toy Story 2 (1999) (VG)*. Technically, they should be counted as different "movies" and our code also accounted for this. In addition, when going through the dataset, we ignored any actor/actress with less than 10 movies as required by counting the number of "\t\t"s appearing in the line. However, during the processing we noticed that for some actors/actresses, there was a movie followed by the same movie with the {{SUSPENDED}} qualifier. Online research indicated that this meant shooting of the movie was suspended, but the actor/actress still receives credit for it. We should not count a suspended movie and the movie itself twice. For example, take a look at actor **Cambridge, Richard** on line **284165** in actor_movies.txt. He has participated in ten movies so he should be included in the actor network. However, upon closer examination, within that line you notice *Idol of Evil (2009)* followed directly by *Idol of Evil (2009) {{SUSPENDED}}*. I think it is fair to assume these are the same movies. It is possible they started shooting at one point and then suspended before resuming. Our code accounts for this and we remove any actors/actresses whose movies fall to less than 10 when we look for this anomaly. It should be noted that we created two data structures in this step in order to help us further along in the project. The first was a dictionary from actors/actresses to movies and the second was a dictionary from movies to actors/actresses. The final number of actors and movies we calculated are shown below:

# of Unique Actors	113098
# of Unique Movie	477055

Question 2

To create the actor network, we first made an edgelist file with three columns per line: *SOURCE TO WEIGHT*. The *SOURCE* is the source vertex of an edge, *TO* is the destination vertex, and *WEIGHT* is the weight of the edge as required. Our first approach was brute force where we looked at every possible pairs of actors in both directions. However, we realized this was not a good approach and used the dictionaries we created in the previous step. For each source actor, we only looked at the actors who had participated in a common movie with source actor. This greatly sped up computation and the edgelist was computed in 35 minutes and the actual graph was created in Python-IGraph in 3 minutes. Below is the in-degree distribution for the actor network



As you can see, most actors have in-degrees in the range of of 0-1000 with very few actors with in-degrees over 1000. This is probably due to the fact that an actor is not likely to participate with such a large range of different actors. In addition, it seems to follow the power law distribution.

Question 3

For this question, we had to create an algorithm to find the preferred actor of ten famous contemporary actors:

The algorithm we devised was to look at all the outgoing edges for each of the ten actors and choose the edge with the highest weight, and return the name of the actor at the endpoint of the highest weight edge found. The results are:

Actor/Actress Name	Preferred Actor to Work With	Edge Weight
Tom Cruise	Nicole Kidman	0.1746
Emma Watson (II)	Daniel Radcliffe	0.5200
George Clooney	Matt Damon	0.1194
Tom Hanks	Tim Allen (I)	0.1125
Dwayne Johnson (I)	Steve Austin (IV)	0.2051
Johnny Depp	Helena Bonham Carter	0.0816
Will Smith (I)	Darrell Foster	0.1224
Meryl Streep	Kevin Kline (I)	0.0619
Leonardo DiCaprio	Martin Scorsese	0.1020
Brad Pitt	George Clooney	0.0986

All of these pairings make sense. Let's go through them one by one to see why. For Tom Cruise,he was married to Nicole Kidman at one point before divorcing and they did participate in several movies together in the 90's as well as first decade of 2000. For Emma Watson (II), the connection to Daniel Radcliffe is obvious. They are both the stars of the *Harry Potter* movie franchise and they featured in all 8 of the franchise's movies. George Clooney and Matt Damon played in the *Ocean's* franchise together. Tom Hanks and Tim Allen were stars in the *Toy Story* franchise. Dwayne Johnson and Steve Austin were both WWE stars and have done several movies together as well. Johnny Depp and Helena Bonham Carter have done several movies together including *Charlie and the Chocolate Factory (2007)* and *Alice in Wonderland (2010).* In addition, Tim Burton, a popular director, often casts Depp and Bonham Carter together, and coincidentally or not, Tim Burton and Helena Bonham Carter were once married. Meryl Streep and Kevin Kline are also well known collaborators and are around the same age. The Leonardo DiCaprio and Martin Scorsese was surprising at first due to Martin Scorsese being mainly a director. However, DiCaprio has done many films directed by Scorsese and Scorsese is known for making cameos either through voice or an extra in his movies. The Brad Pitt and George Clooney pairing is obvious as well due to the *Ocean's* franchise and a couple other movies. It seems franchises are one of the keys to having a high score as some of the pairings are due mainly to franchises with numerous movies.

Question 4

For this question, we had to run pagerank on our actor network and find the actors with the top 10 highest pageranks. We used i-graph's pagerank() function to calculate the scores.

Actor	PageRank	# of Movies	In-Degree
1. Eric Roberts(I)	0.000108	298	3548
2. Ron Jeremy	0.000102	637	2895
3. Danny Trejo	0.0000984	241	3493
4. Bess Flowers	0.0000947	828	7534
5. Adolf Hitler	0.0000897	379	3838
6. Samuel L. Jackson	0.0000892	159	3406
7. Sam Harris (II)	0.0000867	600	6954
8. Richard Riehle	0.0000847	197	3012
9. Robert De Niro	0.0000823	138	3293
10. Harold Miller (I)	0.0000819	561	6581

This top 10 list does not have any actor/actress listed in the previous section. We believe this is due to a couple reasons. One reason is that all the actors above have done an incredible amount of movies. Doing more movies means creating more connections to different actors, leading to the high in-degree and a potentially higher pagerank, and, as you will see in the next section, the well-known actors have done many movies but not as much as the ones above. There are still a couple famous names above like Eric Roberts, Samuel L. Jackson, Robert De Niro, and Danny Trejo who have all had prolific careers and it is not surprising to see them above. One surprising name we noticed was Adolf Hitler. Looking at the dataset, it seems that Hitler was credited in a lot of documentaries due to footage of him being used and as one of the most infamous historical figures of all time, it should come to no surprise many documentaries and movies have been made about him.

Question 5

The pageranks, number of movies, and in-degrees of the 10 actors are shown below:

Actor	PageRank	# of Movies	In-Degree

Tom Cruise	0.0000431	63	1368
Emma Watson (II)	0.0000129	25	423
George Clooney	0.0000419	67	1567
Tom Hanks	0.0000517	80	2038
Dwayne Johnson (I)	0.00003806	78	1347
Johnny Depp	0.0000555	98	2111
Will Smith (I)	0.0000344	49	1285
Meryl Streep	0.0000414	97	1580
Leonardo DiCaprio	0.00003501	49	1296
Brad Pitt	0.0000462	71	1736

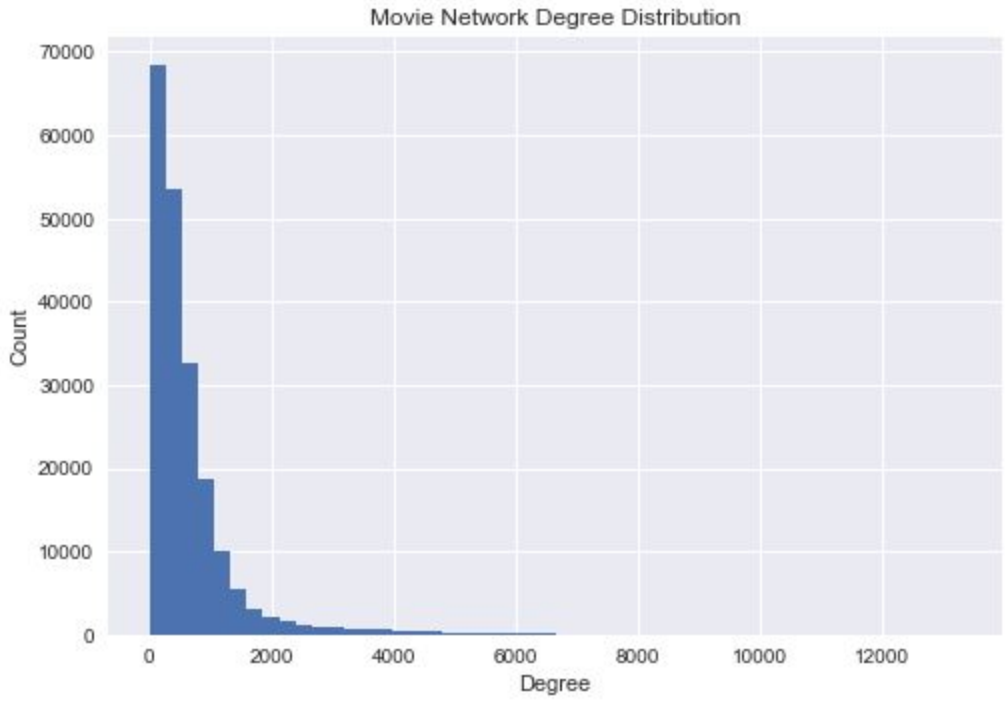
As you can see, despite having done many movies and having high in-degrees, they still pale in comparison to the actors with the highest page ranks.

2 Movie Network

For this part, our goal was to create an undirected movie network to analyze the relationships between various movies

Question 6

To create the weighted, undirected movie network, we followed a similar approach to the actor network. We first created an edgelist file in the form: *SOURCE TO EDGE*. The *source* and *to* do not have as much meaning as in the actor network since the network is undirected. To prevent looking at all possible movie pairings, we used the movies to actor dictionary we had created to look only the the movies that shared a common actor with each movie. The degree distribution is shown below

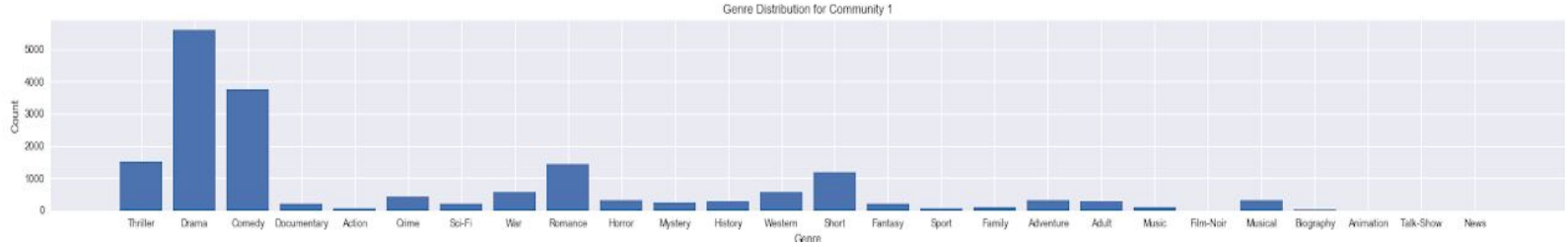


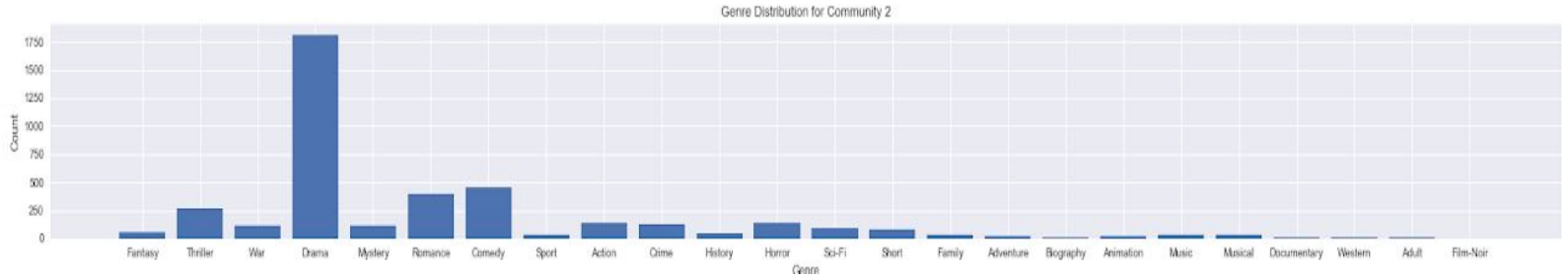
The degree distribution follows a similar pattern to the actor network with many movies with low degrees and less with high degrees. This makes intuitive sense as the connections between movies are predicated on having common actors and it is unlikely for a movie to share a common actor with over 1000 movies. It also seems to follow the power law distribution.

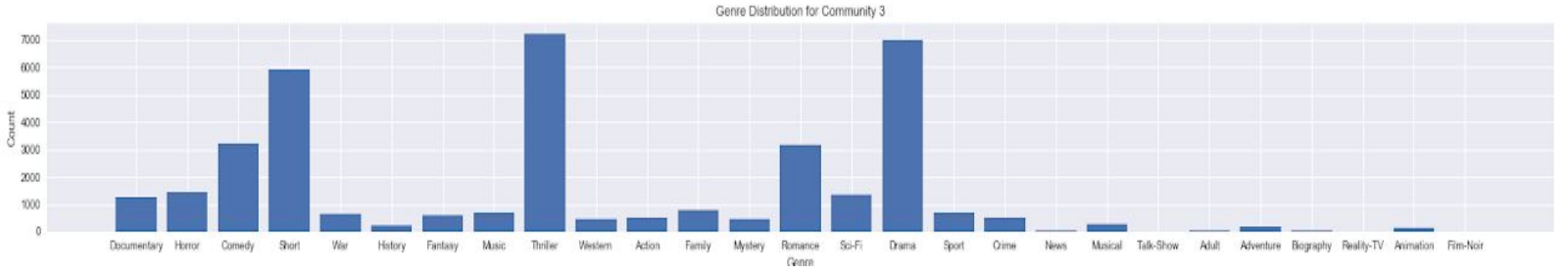
Question 7

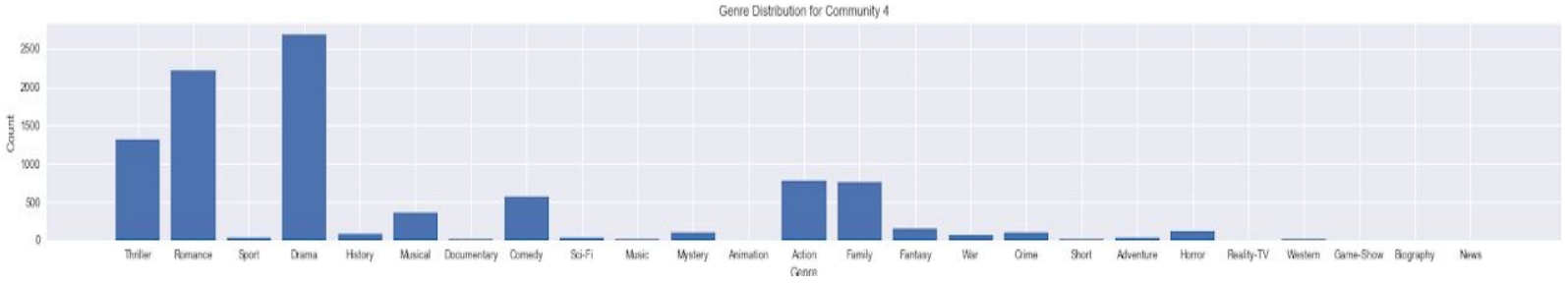
This question was tricky. We first tried generating the community structure in Python and it took 48.35 hours exactly. Based on our understanding of R in our previous projects, we tried generating the community structure in R by creating the movie network using the edgelist we

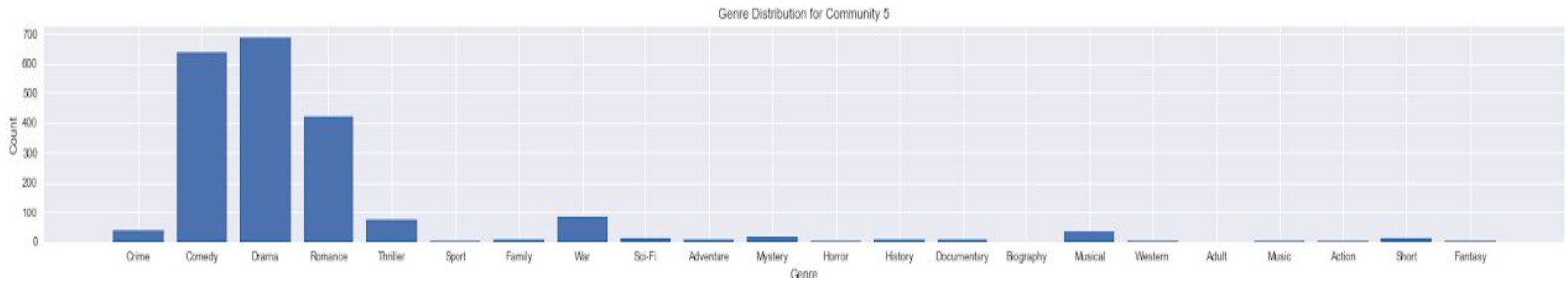
created in Python and using the fastgreedy.community() method. It took 3.42 hours in R and we had 31 communities versus 15 communities in Python.We decided to go with the community structure found in R. Genre distributions for the first 10 communities are below.

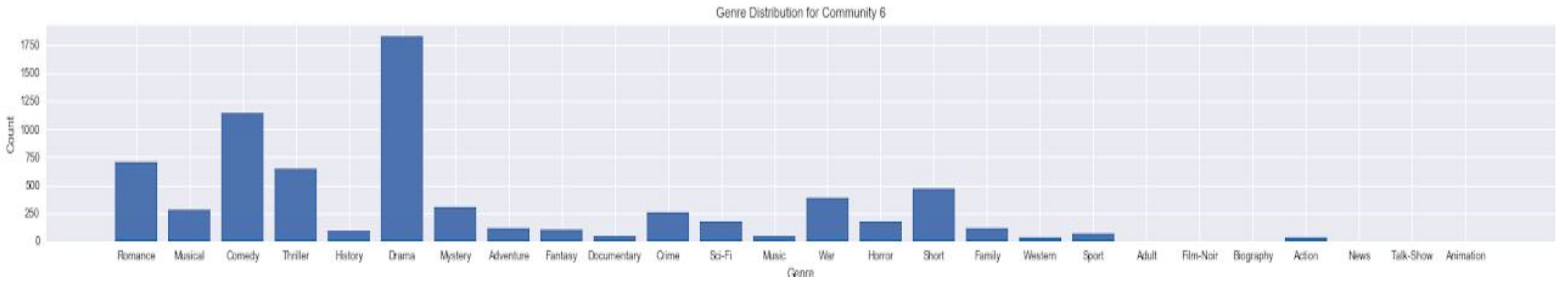


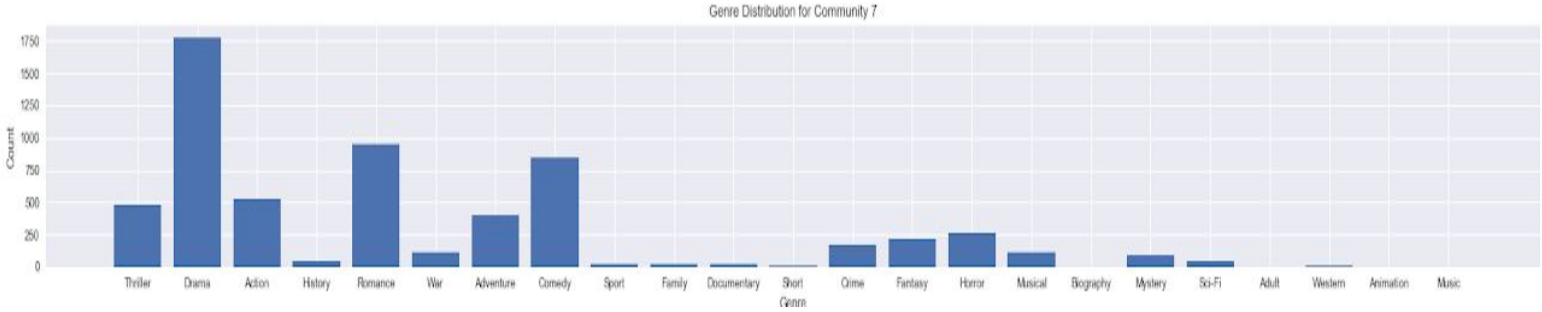


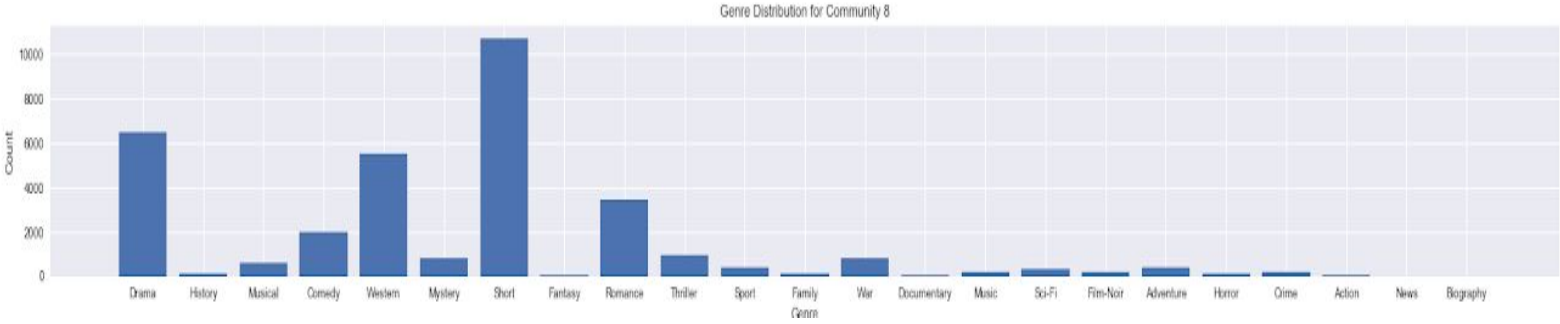


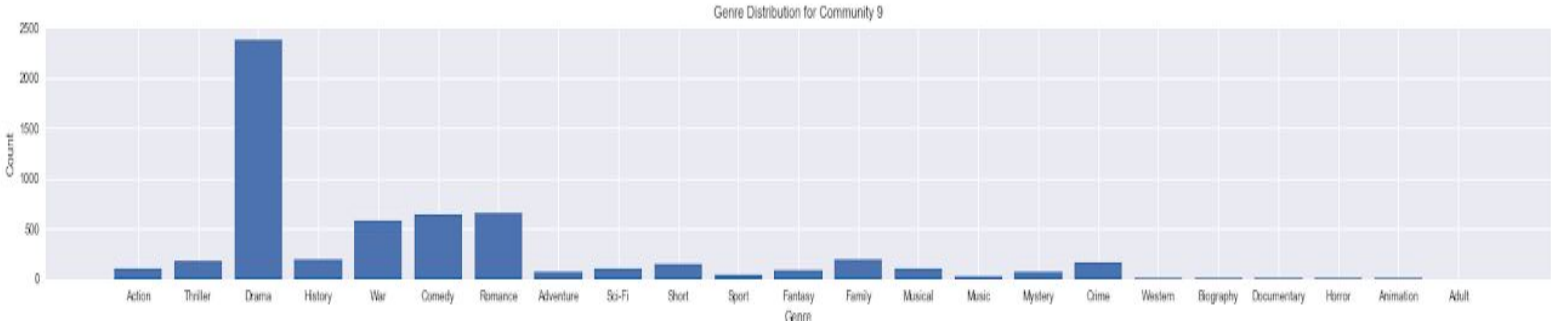


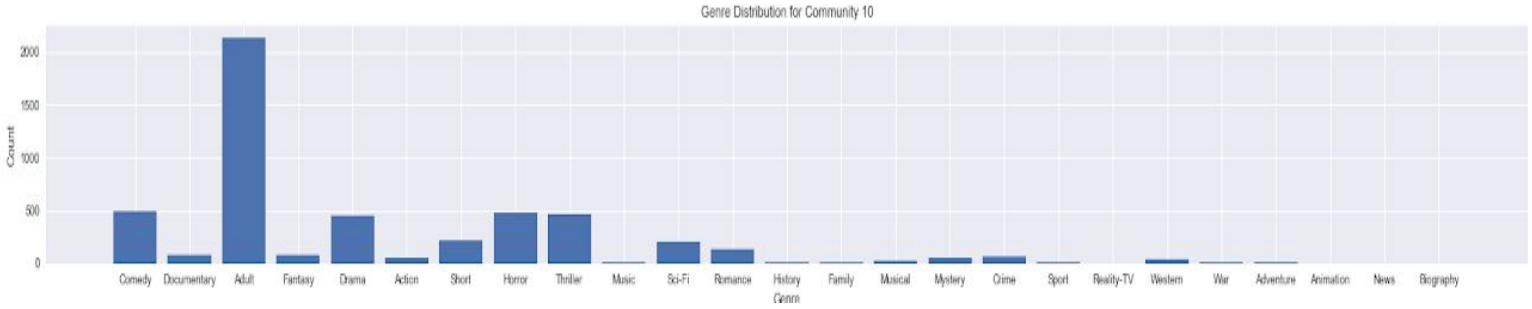

Genre Distribution for Community 5
Count
Genre
Crime
Comedy
Drama
Romance
Thriller
Sport
Family
War
Sci-Fi
Adventure
Mystery
Horror
History
Documentary
Biography
Musical
Western
Adult
Music
Action
Short
Fantasy


Genre Distribution for Community 6
Count
Genre
Romance
Musical
Comedy
Thriller
History
Drama
Mystery
Adventure
Fantasy
Documentary
Crime
Sci-Fi
Music
War
Horror
Short
Family
Western
Sport
Adult
Film-Noir
Biography
Action
News
Talk-Show
Animation


Genre Distribution for Community 7
Count
Genre
Thriller
Drama
Action
History
Romance
War
Adventure
Comedy
Sport
Family
Documentary
Short
Crime
Fantasy
Horror
Musical
Biography
Mystery
Sci-Fi
Adult
Western
Animation
Music


Genre Distribution for Community 8
Count
Genre
Drama
History
Musical
Comedy
Western
Mystery
Short
Fantasy
Romance
Thriller
Sport
Family
War
Documentary
Music
Sci-Fi
Film-Noir
Adventure
Horror
Crime
Action
News
Biography


Genre Distribution for Community 9
Count
Genre
Action
Thriller
Drama
History
War
Comedy
Romance
Adventure
Sci-Fi
Short
Sport
Fantasy
Family
Musical
Music
Mystery
Crime
Western
Biography
Documentary
Horror
Animation
Adult



Question 8(a)

Below, we mark the most frequent genre in each community as well as the exact number of occurrences of the genre.

Community Number	Most Frequent Genre	Count
1	Drama	5609
2	Thriller	1814
3	Drama	7238
4	Drama	2690
5	Drama	690
6	Drama	1831
7	Drama	1779
8	Short	10723
9	Drama	2383
10	Adult	2138
11	Drama	1744
12	Drama	1418
13	Drama	446
14	Drama	311
15	Drama	297
16	Drama	565

17	Drama	987
18	Drama	1028
19	Drama	1617
20	Drama	720
21	Drama	103
22	Romance	165
23	Thriller	8
24	Drama	238
25	Short	12
26	Adult	8
27	Short	9
28	Musical	73
29	Short	80
30	Drama	8
31	Short	8

As you can see, the most dominant genre across communities tends to be drama. 20 of the 31 communities had drama as their most frequent genre. This is not surprising as dramas do tend to be one of the most popular genres for movies. Since movies in this dataset can have more than one genre, many movies have drama elements along with other styles.

Question 8(b)

For this question, we wanted to identify the most frequent genre in a different way. Specifically, we assign each genre in the community a score of $ln(c(i)) * (p(i) / q(i))$. These modified scores would account for the popularity of a genre across the entire dataset to find the relative importance of a genre within a community. This is very similar to the TF-IDF approach.

Community Number	Most Frequent Genre	Count

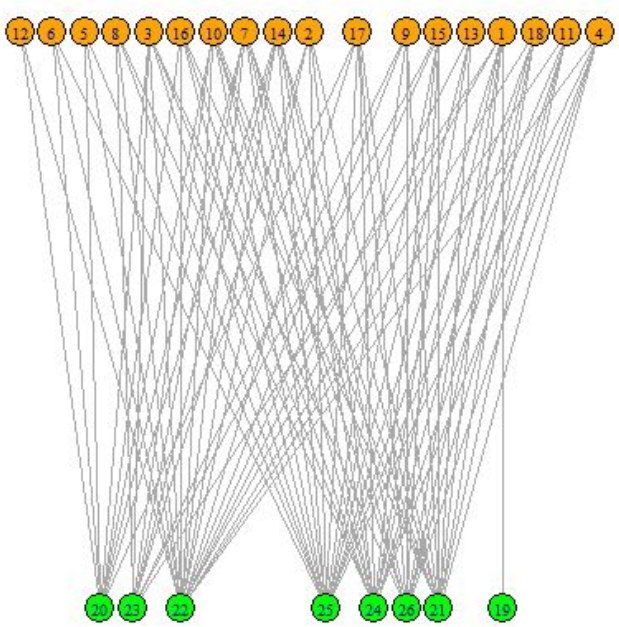
1	Comedy	12.115
2	Drama	7.215
3	Documentary	22.66
4	Family	27.847
5	Comedy	22.057
6	Mystery	18.265
7	Adventure	34.227
8	Western	38.4464
9	War	24.935
10	Adult	245.033
11	Comedy	7.749
12	Western	12.566
13	Family	19.373
14	History	8.594
15	Romance	11.682
16	War	37.801
17	Musical	14.432
18	Family	17.351
19	Action	53.088
20	Adventure	32.298
21	Action	8.024
22	Romance	14.793
23	Thriller	16.136
24	Action	18.635
25	Short	17.772
26	Adult	120.619

27	Short	11.131
28	Musical	52.772
29	Short	43.849
30	Sport	8.373
31	Short	5.45

As you can see, only one of the communities, community 2, has drama as its most frequent genre based on the modified scores. Less common genres like shorts, war, and western become more dominant. We believe this is due to the fact that these genres are less common in the entire dataset but appear in a greater proportion inside the community than their true distribution. This means those genres are relatively more important within the community.

Question 8(c)

We chose community 27 which had 18 movies for this question.



Mapping:

Movies	Actors
[1] Unconditional Love (2010)	[19] Steven Dasz
[2] Be My Valentine (2011)	[20] Tom Hislop
[3] Losers in Love (2011)	[21] Craig Joiner
[4] The Hope Within (2009)	[22] Reuben McKay
[5] Life of a Spy (2012)	[23] Shaun Moir
[6] Booze Culture (2012)	[24] Graeme Noble
[7] Call of Babylon (2012)	[25] John-William Noble
[8] Fear of the Park (2010)	[26] Martin Sadison
[9] Love House (2011)	
[10] Boycie (2011)	
[11] White Cobra (2014)	
[12] Inner Joy of a Broken Heart (2011)	
[13] The Book of Life (2013/I)	
[14] The Shadow of Death (2011)"	
[15] Legion of Evil (2010)	
[16] Is This It? (2012)	
[17] Street Fight (2012)	
[18] A Schoolboy Error Production (2009)	

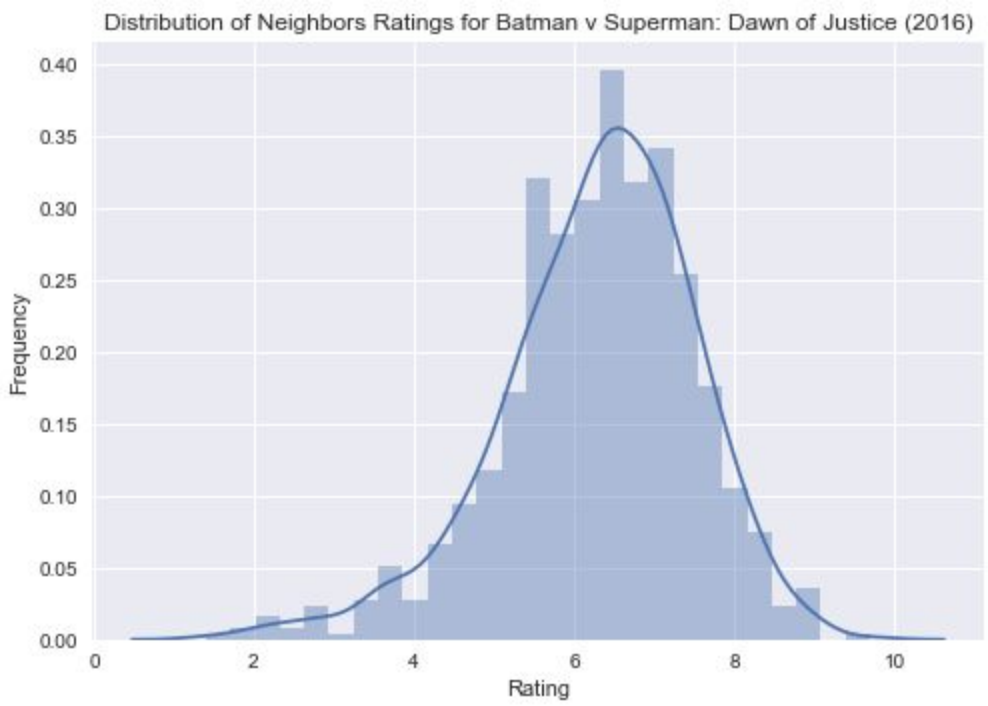
We found the top three actors in this network using the pagerank scores after running pagerank on the bipartite network. The top three actors are :

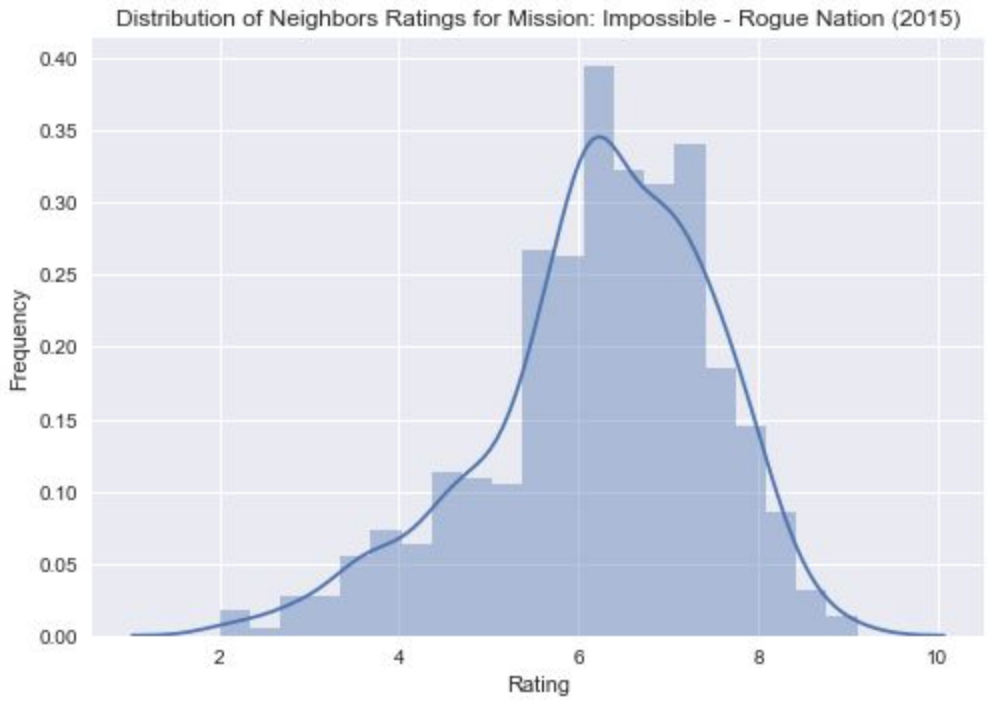
Actor	PageRank	Movies/Degree
Reuben McKay	0.093456	18
John-William Noble	0.087103	17
Graeme Noble	0.07144	14

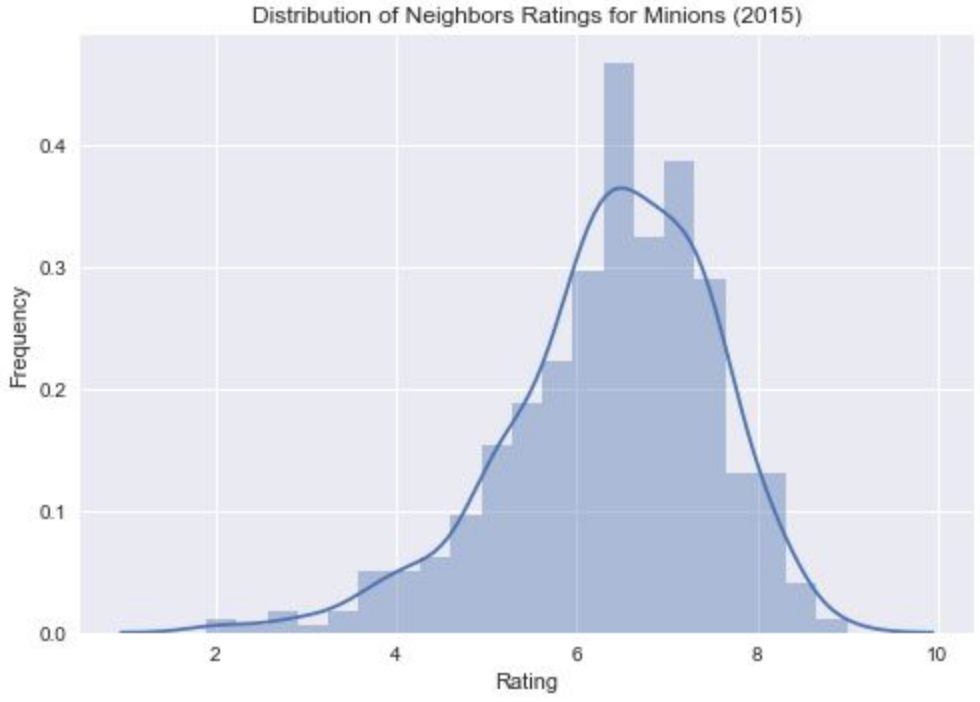
As you can see, all three of these actors participated in the vast majority, if not all, of the movies in the community. Thus, it makes sense that these three actors would form the core of the community. The most frequent genre in community 27 was short with a count of 9 and short with a score of 3.21 using modified scores. This makes sense as looking at the IMDB pages for these actors , the majority of the movies these three actors played in are shorts.

Question 9

Below are the the distributions of the ratings of the neighbors of the three movies, Batman v Superman: Dawn of Justice, Mission Impossible: Rogue Nation (2015), and Minions (2015)




Distribution of Neighbors Ratings for Mission: Impossible - Rogue Nation (2015)
Frequency
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
2
4
6
8
10
Rating


Distribution of Neighbors Ratings for Minions (2015)
Frequency
0.4
0.3
0.2
0.1
0.0
2
4
6
8
10
Rating

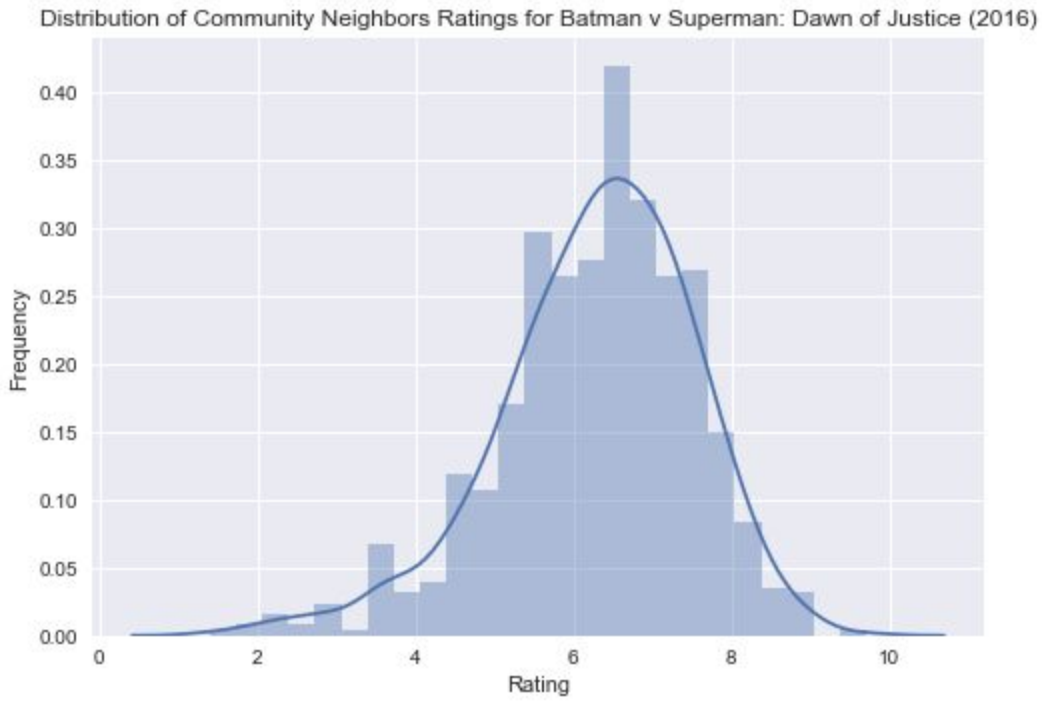
The below table compares the actual ratings of the three movies to the average ratings of their respective neighbors

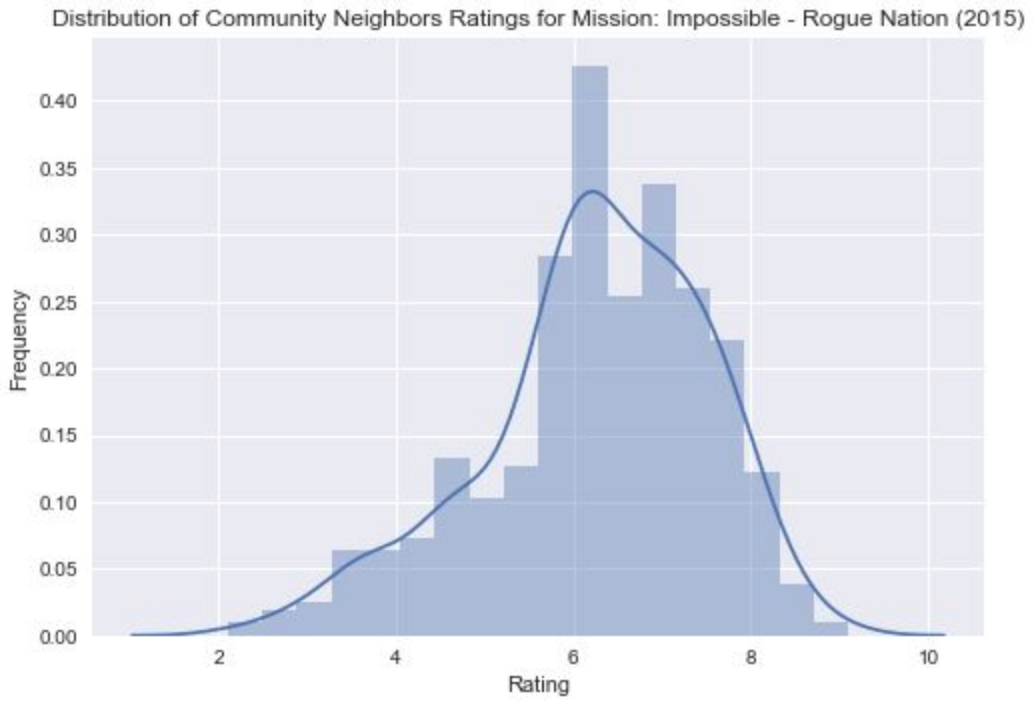
Movie	Actual Rating	Average Rating of Neighbors
Batman v Superman: Dawn of Justice	6.6	6.296
Mission Impossible: Rogue Nation	7.4	6.232
Minions	6.4	6.384

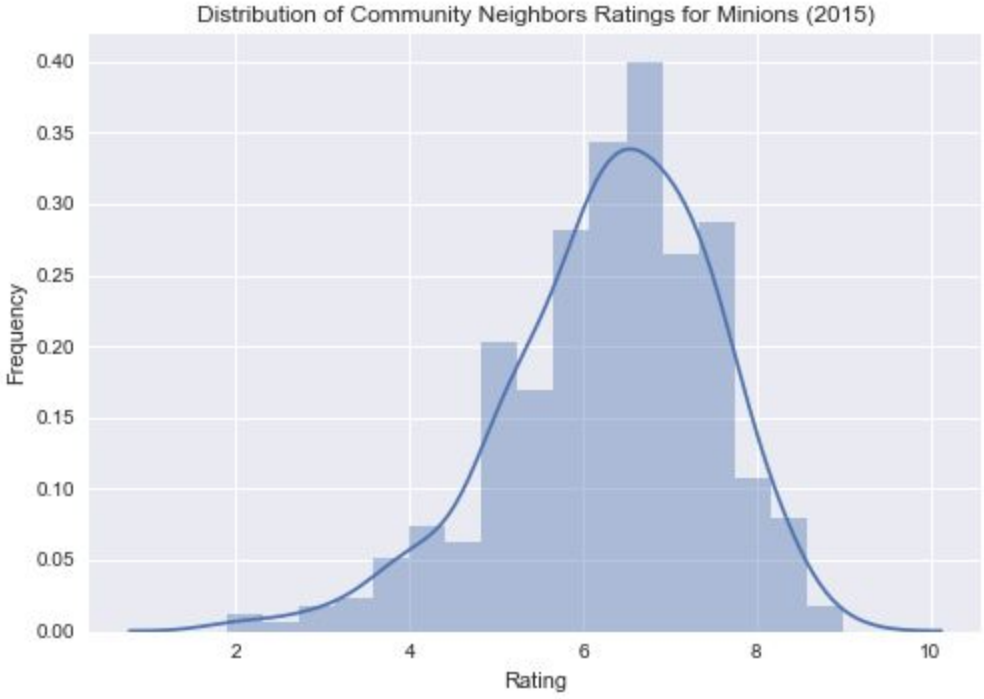
All the average ratings seem fairly close, particularly for Minions (2015). The biggest discrepancy is for Mission Impossible: Rogue Nation where the difference between the actual rating and average rating of the neighbors is around 1.2.

Question 10

For this question, instead of looking at all the neighbors to find the average rating, we only considered neighbors in the same community as each movie.




Distribution of Community Neighbors Ratings for Mission: Impossible - Rogue Nation (2015)
Frequency
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
2
4
6
8
10
Rating


Distribution of Community Neighbors Ratings for Minions (2015)
Frequency
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
2
4
6
8
10
Rating

Movie	Actual Rating	Average Rating of Neighbors
Batman v Superman: Dawn of Justice	6.6	6.293
Mission Impossible: Rogue Nation	7.4	6.269
Minions	6.4	6.331

When restricted to neighbors in the same community, the average ratings were pretty similar the difference between the actual and average rating improving for Mission Impossible: Rogue Nation when restricted to community neighbors. However, the difference did get worse for Minions slightly. Movies in the same community may share a core group of actors but adding more movies that may not be in the community can add a dampening factor to the effect of a few core actors. It is common in deep learning to add noise to a model to prevent overfitting and it is analogous to this case where we want to mitigate the effect the performances of a few actors might have across different movies by taking into account the ratings of movies which may not be as similar.

Question 11

For this question, we found the top five neighbors for the three movies using edge weights. In other words, we wanted to find the movies who shared a lot of common actors with the three movies.

Top Five Neighbors for Batman v Superman with Community Membership

Movie	Community Membership	Edge Weight
Eloise (2015)	3	0.113
The Justice League Part One (2017)	3	0.076
Into the Storm (2014)	3	0.073
Love and Honor (2013)	3	0.061
Man of Steel (2013)	3	0.059

Top Five Neighbors for Mission Impossible - Rogue Nation (2015)

Movie	Community Membership	Edge Weight
Fan (2015)	4	0.159

Phantom (2015)	4	0.146
Breaking the Bank (2014)	3	0.1028
Suffragette (2014)	3	0.1022
Now You See Me: The Second Act (2016)	3	0.1011

Top Five Neighbors for Minions (2015):

Movie	Community Membership	Edge Weight
The Lorax (2012)	3	0.278
Inside Out (2015)	3	0.250
Up (2009)	3	0.250
Surf's Up (2007)	3	0.231
Despicable Me 2 (2013)	3	0.225

Each of the three movies are also in community 3. Thus, it should come as no surprise that most of the top five neighbors are in the same community as each movie except for a couple for Mission Impossible. The two movies not in the same community, Fan and Phantom, are Indian movies, so though they have common actors with Mission Impossible: Rogue Nation, it makes intuitive sense those movies would belong to a community that includes more Indian movies so the possibility of common actors is higher. In addition, all of these movies are mainstream ones so it makes sense they pretty much all belong to the same community as many of the high-budget flicks tend to cast similar people.

Question 12

The five features we used to train a regression model were the page ranks of the top 5 actors in each movie. We felt like these were good features as a high page rank denotes a person with a lot of connections and high reputation who might be likely to play in higher end movies. In addition, all the movies have at least 5 actors, so it made sense to use this approach.

We used the Linear Regression model from scikit-learn's linear_model package and trained on all movies that have available ratings in the movie_to_rating.txt file, splitting it into a training set of 90% and 10% test. We also used a Random Forest regression model to compare the results.

Linear Regression Model

Data	RMSE
Training Set	1.1928
Test Set	1.2279
Three Movies	0.8135

Random Forest Model

Data	RMSE
Training Set	1.0641
Test Set	1.1926
Three Movies	0.9791

The random forest performed better than the linear regression model. We expected this as the random forest is an ensemble model and ensemble models tend to perform better due to the fact that the various models are identically and independently distributed, thus reducing the variance and preventing overfitting.

Linear Regression Model Predictions

Movie	Actual Rating	Predicted Rating
Batman v Superman: Dawn of Justice	6.6	6.0981
Mission Impossible: Rogue Nation	7.4	6.1209
Minions	6.4	6.2575

Random Forest Model Predictions

Movie	Actual Rating	Predicted Rating
Batman v Superman: Dawn of Justice	6.6	6.1062
Mission Impossible: Rogue Nation	7.4	6.5252

Minions	6.4	6.5959

Question 13

We created the bipartite graph using the aforementioned edgelist approach and had a dictionary that mapped each actor to a weight which we decided would be the average rating of all the movies with available ratings that the actor has played in. We consider this is a good metric because a high average rating would indicate that an actor has played in a lot of highly rated movies. So, a movie with many high weight actors would probably be a highly rated movie. Likewise, a low average rating for an actor would mean the actor has played in many low rated movies so a movie with actors with low scores would probably be a low rated movie. To make the actual prediction, we took the both average of all the weights of the actors connected to the target movie in the bipartite graph as well as the average of the top 5 actors

Using All Actor Scores

Movie	Actual Rating	Predicted Rating
Batman v Superman: Dawn of Justice	6.6	6.3098
Mission Impossible: Rogue Nation	7.4	6.4336
Minions	6.4	6.5533

Using Top 5 Scores Actors

Movie	Actual Rating	Predicted Rating
Batman v Superman: Dawn of Justice	6.6	7.884
Mission Impossible: Rogue Nation	7.4	7.777
Mininos	6.4	7.213

Since we are not training a model, our “training set” would be all the movies we in the bipartite graph. We compared the predicted rating to the actual rating, calculating the RMSE

Using All Actor Scores

Set	RMSE
Movies in Graph	1.0217

Three movies	0.5892

Using Top 5 Actor Scores

Set	RMSE
Movies in Graph	1.065
Three movies	1.566

.
Using all the actor scores who participated in a movie rather than only the top 5 provided a better estimate for the three movies although the difference in RMSE for all the movies is negligible. . We believe this is because using the top 5 actor scores provides an overly optimistic assessment as more than a few actors lead to the success of a movie.

We believe this approach is better because even though pagerank is a good metric, a high pagerank typically corresponds to having played in a lot of movies. Having played in a lot of movies is not necessarily the best indicator of quality as an actor could have played in hundreds of B-movies. Thus, we feel like taking a weighted average of actors' ratings' is a better heuristic for predicting ratings as can be seen by the lower RMSE.