EE 232E Project 4:IMDb Mining

Group Member:
Fangyao Liu (204945018)
Chaojie Feng (505025111)
Haitao Wang (504294402)
Xiao PENG (005033608)

Part 1. Actor/Actress network

Given actor_movies.txt and actress_movies.txt, we are asked to create the network about all actors and actress, whose weights are determined by common movies divided by total movies that this actor/ actress that participated in.

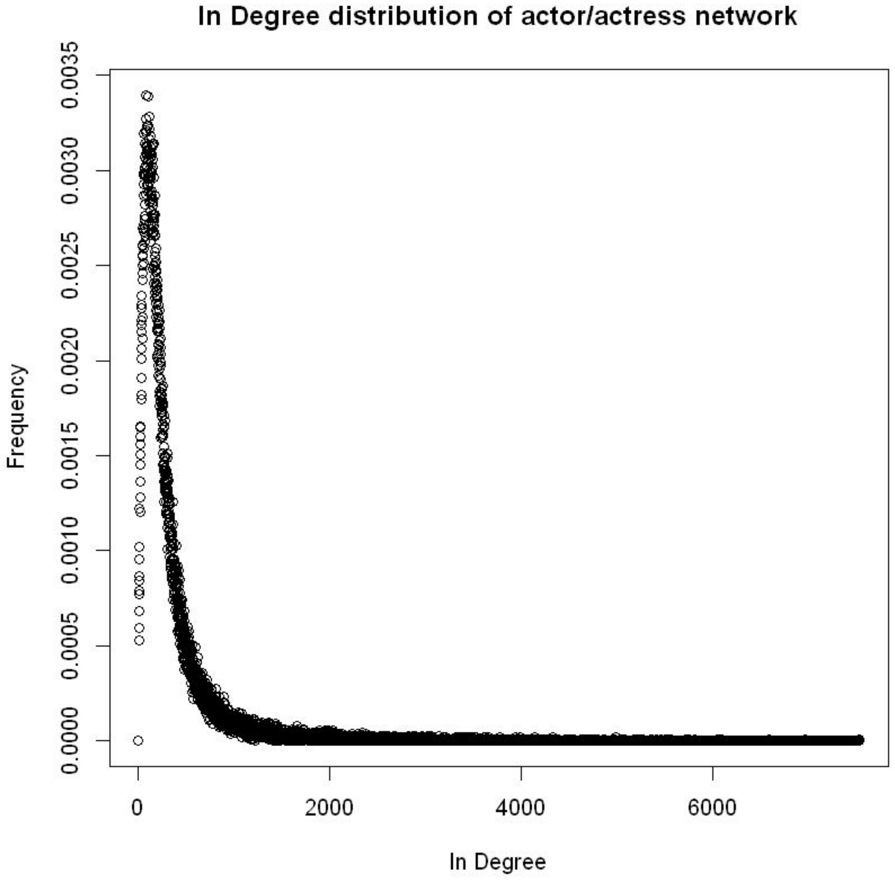
Before we discover the network, we need to clean the data. The steps include:

1. merging two files and then remove actor/actress who acted in less than 10 movies
2. Clean and standardize movie names

Question 1: After perform filtering on the data, we find out:

The total number of actors and actresses: 113132
The total number of unique movies: 468467

Question 2: Then we create a directed edge-list of different combination of actors/actresses. Directed weight from actor/actress i to actor/actress j is computed by common movies of i and j then divided by total movies acted by actor/actress i. After we get the edge-list, we load the file into R system and plot the in-degree distribution.



As we can see in the in-degree distribution plot, higher in-degree, less the frequency. This plot matches the reality. Because in-degree represents the times that an actor/actress has cooperated with others and ones that cooperated with many actors are minorities, The majorities are the ones that only cooperated with an intermediate number of actors/actresses.

Question 3: The algorithm takes input actor name and then transfer it to Node ID number. It searches for the output actor with highest weight that the input node is connected with.

The pairing between actors and actress can be tabulated below:

Actor Input	Actor Output	Weight
Tom Cruise	Nicole Kidman	0.1746
Emma Watson (II)	Daniel Radcliffe	0.52
George Clooney	Matt Damon	0.1194
Tom Hanks	Tim Allen	0.1021
Dwayne Johnson (I)	Steve Austin (IV)	0.2051
Johnny Depp	Helena Bonham Carter	0.0816
Will Smith (I)	Darrel Forester	0.1224
Meryl Streep	Robert De Niro	0.0618
Leonardo DiCaprio	Martin Scorsese	0.102
Brad Pitt	George Clooney	0.0986

From our search about each “best pair”, we find out that our results make sense. Take highest weight and lowest weight pair: “Emma Watson (II) & Daniel Radcliffe” and “Meryl Streep & Robert De Niro” as examples, we google these two pairs and results show that:

1 Emma Watson and Daniel Radcliffe acted as hero and heroine in Harry Potter Series. This series is a success and after so many cooperation, they must be a good pair.
2 Meryl Streep & Robert De Niro although has the lowest weight among these ten pairs, but our search also indicated that these two are very good friends and have deep emotion to each other, which implies they are a good pair

And she was greeted at the podium by fellow legendary actor Robert De Niro who planted a big kiss on The Post star's lips. The two are old pals, who famously costarred in 1978 Vietnam drama the Deer Hunter.

Scroll down for video

Question 4: We are using google pagerank algorithm to find the top 10 actor/actress in the network. The result is tabulated as:

Rank	ID	Name	In Degree	Movies	Score
1	85734	Bess Flowers	7537	828	0.00024
2	65947	Fred Tatasciore	3954	355	0.00020
3	27643	Sam Harris (II)	6960	647	0.00019
4	6539	Steve Blum (IX)	3316	373	0.00019
5	45415	Harold Miller (I)	6587	561	0.00017
6	32130	Ron Jeremy	2905	637	0.00016
7	52784	Lee Phelps (I)	5563	647	0.00016
8	40351	Yuri Lowenthal	2662	318	0.00015
9	18112	Robin Atkin Downes	2948	267	0.00015
10	49651	Frank O'Connor (I)	5502	623	0.00014

Table 4. Top actors/actresses found by PageRank algorithm

From table 4, we find that none of the actors/actresses names are familiar and none of them is in the previous section. This is surprising first but actually makes sense. Because there are two factors that will influence the PageRank Value of a certain node "a":

1. Number of nodes that links into "a"
2. The PageRank value of these nodes that links into "a"

So actors/actresses who cooperated with more actors/actresses are more likely to higher PageRank value, rather than the famous ones. For example, Bess Flowers was considered as the most prolific extra in Hollywood, but she didn't play major roles in movies. Notice that Fred

Tatasciore has less in-degree than Sam Harris (II) but still has a higher PageRank value. This is because the second factor: the nodes that link into Fred Tatasciore have higher value than the nodes that link into Sam Harris (II).

Question 5: Pagerank scores, number of movies, in-degree of actors/actresses listed in the previous section will be presented below:

Actor/Actress	Pagerank score	Number of movies	In degree
Tom Cruise	3.972248e-05	63	1651
Emma Watson (II)	1.761577e-05	25	453
George Clooney	4.000827e-05	67	1573
Tom Hanks	5.148528e-05	80	2064
Dwayne Johnson (I)	4.199428e-05	78	1357
Johnny Depp	5.379456e-05	98	2144
Will Smith (I)	3.201524e-05	49	1319
Meryl Streep	3.962374e-05	97	1594
Leonardo DiCaprio	3.166586e-05	49	1301
Brad Pitt	4.298439e-05	71	1739

Compared to top 10 pagerank actor/actress, these famous stars have acted in much less movies and therefore leads to a relatively low pagerank value.

2.1 Undirected movie network creation

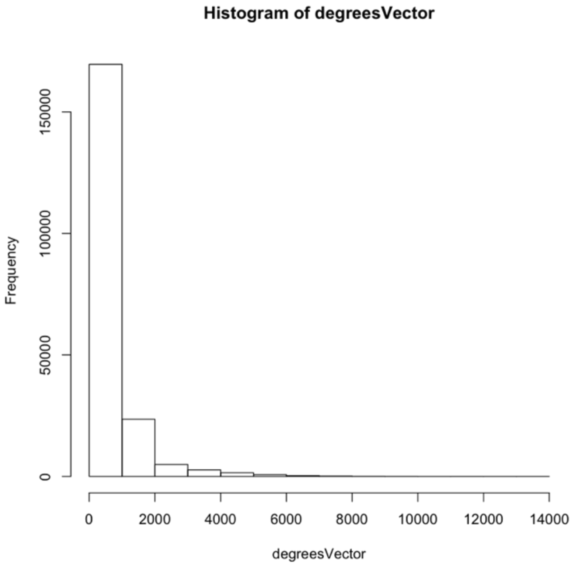
Question 6: In this question, we are supposed to create a weighted undirected movie network, whose weight of the edges are given by the equation

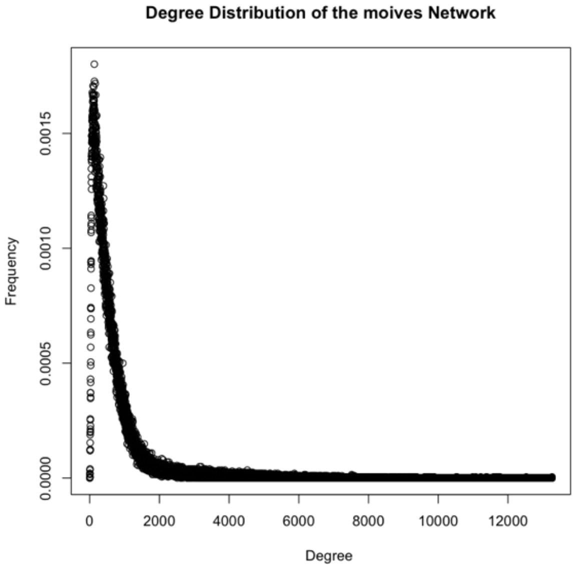
$$w_{i\to j} = \frac{|A_i \cap A_j|}{|A_i \cup A_j|}$$

Thus, the first step to implement this is to pick the movies under the requirement that it should have more than 5 performers in it. Then we need to construct a dictionary containing all filtered movies and the corresponding actors and actresses in these movies. The process is very time-consuming. After that, we calculate the weight by applying the equation shown above. Since the above operations are implemented in Python, and additionally, the following steps must be achieved with igraph package in R, we write the results stored in the dictionary data structure into a ".txt" file for the following questions. These whole processing takes us about two days to get the results.

The remaining steps are simply to create the weighted network based on the data we have obtained and to plot the degree distribution of the graph.

The plots of the degree distribution are shown below. The degree distribution shows that the most vertices have degrees below 1000. The whole curve has an exponent-function-like shape.

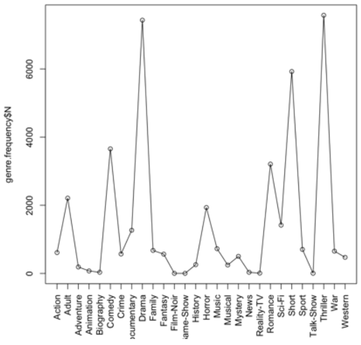


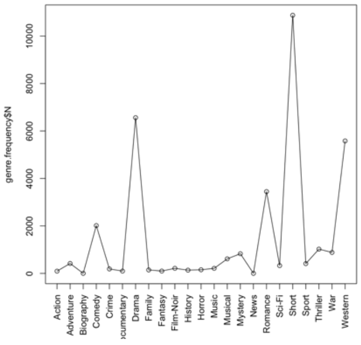


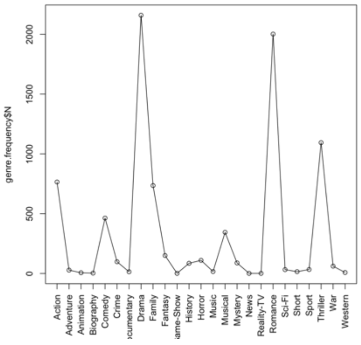
2.2 Communities in the movie network

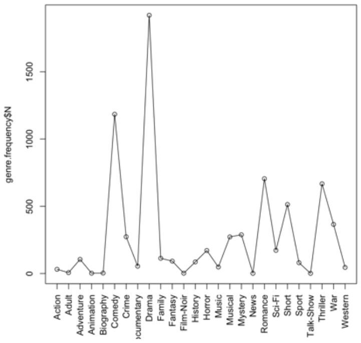
Question 7: Apply the fast greedy algorithm in R to the graph and we can get the community information of this graph. Totally, we can get 31 communities in the graph, and randomly pick 10 communities to analyze their genre distributions.

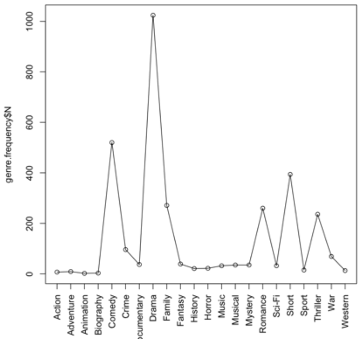
To get the genre distribution of each community, we have to read the “movie_genre.txt” file and find all the corresponding genres of the movies in the community.

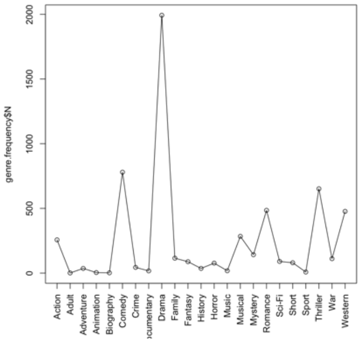

genre.frequency$N
0
2000
4000
6000
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
ocumentary
Drama
Family
Fantasy
Film-Noir
ame-Show
History
Horror
Music
Musical
Mystery
News
Reality-TV
Romance
Sci-Fi
Short
Sport
Talk-Show
Thriller
War
Western

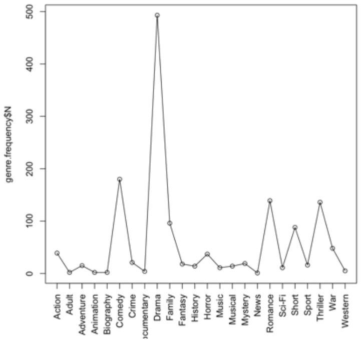

genre.frequency$N
0
2000
4000
6000
8000
10000
Action
Adventure
Biography
Comedy
Crime
ocumentary
Drama
Family
Fantasy
Film-Noir
History
Horror
Music
Musical
Mystery
News
Romance
Sci-Fi
Short
Sport
Thriller
War
Western

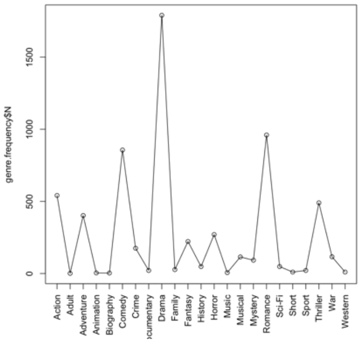

genre.frequency$N
0
500
1000
1500
2000
Action
Adventure
Animation
Biography
Comedy
Crime
ocumentary
Drama
Family
Fantasy
ame-Show
History
Horror
Music
Musical
Mystery
News
Reality-TV
Romance
Sci-Fi
Short
Sport
Thriller
War
Western

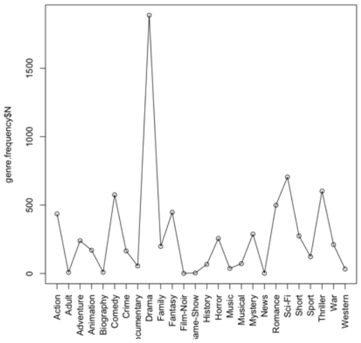

genre.frequency$N
0
500
1000
1500
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
ocumentary
Drama
Family
Fantasy
Film-Noir
History
Horror
Music
Musical
Mystery
News
Romance
Sci-Fi
Short
Sport
Talk-Show
Thriller
War
Western

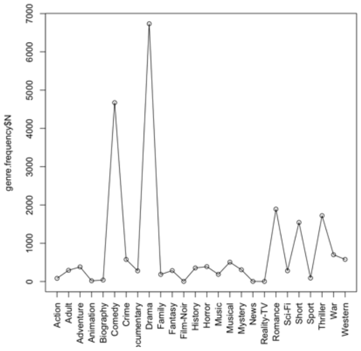

genre.frequency$N
0
200
400
600
800
1000
Action
Adventure
Animation
Biography
Comedy
Crime
ocumentary
Drama
Family
Fantasy
History
Horror
Music
Musical
Mystery
Romance
Sci-Fi
Short
Sport
Thriller
War
Western


genre.frequency$N
0
500
1000
1500
2000
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
ocumentary
Drama
Family
Fantasy
History
Horror
Music
Musical
Mystery
Romance
Sci-Fi
Short
Sport
Thriller
War
Western


genre.frequency$N
0
100
200
300
400
500
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
ocumentary
Drama
Family
Fantasy
History
Horror
Music
Musical
Mystery
News
Romance
Sci-Fi
Short
Sport
Thriller
War
Western


genre.frequency$N
0
500
1000
1500
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
ocumentary
Drama
Family
Fantasy
History
Horror
Music
Musical
Mystery
Romance
Sci-Fi
Short
Sport
Thriller
War
Western


genre.frequency$N
0
500
1000
1500
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
ocumentary
Drama
Family
Fantasy
Film-Noir
ame-Show
History
Horror
Music
Musical
Mystery
News
Romance
Sci-Fi
Short
Sport
Thriller
War
Western


genre.frequency$N
0
1000
2000
3000
4000
5000
6000
7000
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
ocumentary
Drama
Family
Fantasy
Film-Noir
History
Horror
Music
Musical
Mystery
News
Reality-TV
Romance
Sci-Fi
Short
Sport
Thriller
War
Western

Question 8:

After obtaining the distribution results, we can easily identify the dominant movie genre in the community by observation. For all the 31 communities, based simply on frequency counts, the dominant movie genres are recorded in the following table.

1	Thriller	17	Drama
2	Short	18	Drama
3	Drama	19	Drama
4	Drama	20	Romance
5	Drama	21	Drama
6	Drama	22	Drama
7	Drama	23	Thriller
8	Drama	24	Drama
9	Drama	25	Short
10	Drama	26	Adult
11	Drama	27	Short
12	Drama	28	Musical
13	Drama	29	Short
14	Drama	30	Drama
15	Drama	31	Drama
16	Short		

Based on this table, we can come up with the frequency for every genre being the dominant genre for a single community. We list it below.

Dominant Genre	**Amount**
Adult	1
Drama	21
Musical	1
Romance	1
Short	5
Thriller	2

We can determine the most dominant genre as "Drama" for all the communities, since it is considered as the dominant genre for 21 communities, which is more than other genres can be.

In last part of the question, we define the dominant genre for a certain community simply based on the frequency. But in this part of the question, we come up with a new concept for “Modified Score” as the definition for the dominant genre.

$$Modified\ Score = ln(c(i)) \times \frac{p(i)}{q(i)}$$

Applying this measurement, we can get the new results with similar steps.

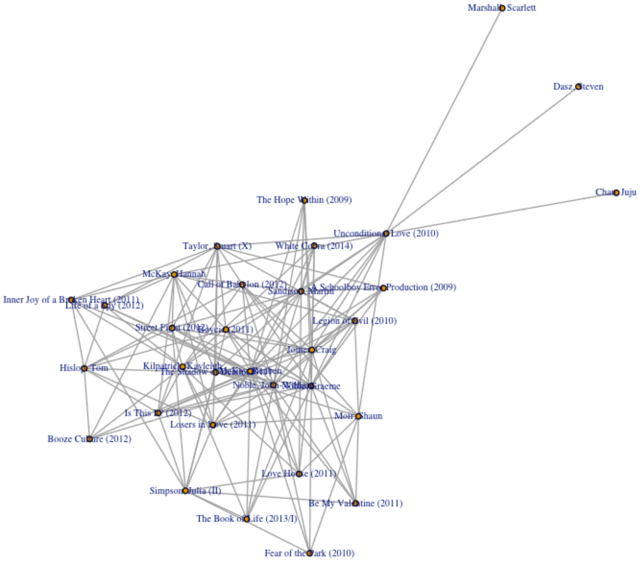
1	Thriller	17	Adventure
2	Film-Noir	18	Crime
3	Romance	19	Action
4	Crime	20	Romance
5	Comedy	21	Romance
6	Sci-Fi	22	Romance
7	Adventure	23	Thriller
8	Family	24	Drama
9	Western	25	Short
10	Family	26	Adult
11	War	27	Romance
12	War	28	Musical
13	Comedy	29	Short
14	Action	30	Drama
15	Fantasy	31	Drama
16	Short		

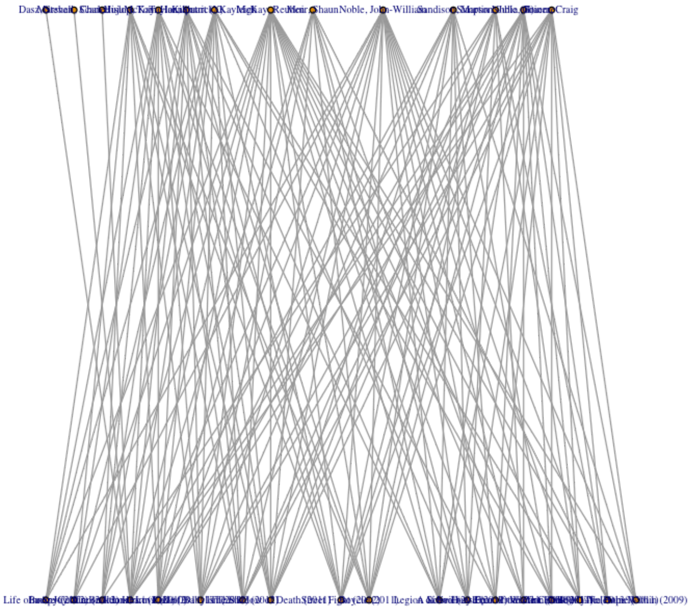
Dominant Genre	**Amount**
Action	2
Adult	1
Adventure	2
Comedy	2
Crime	2
Drama	3

Family	2
Fantasy	1
Film-Noir	1
Musical	1
Romance	5
Sci-Fi	1
Short	3
Thriller	2
War	2
Western	1

Comparing the results of 8(a) and 8(b), we can find that in 8(b) different communities has different dominant movies. In fact, the results show the improvement which is brought by the new “modified score” concept. The “modified score” takes the fraction of genre in the entire data set into consideration, which can compensate the inaccuracies during finding the dominant genre, due to the fact that the fraction of genre in the entire data set may vary. Putting it in another way, drama movies take the most significant part in all movies, while the film-noir movies very few. This makes it less possible for the film-noir movie to be a dominant movie of a community, than it is for the drama movie.

Among the 31 communities, we randomly pick one community whose size is between 10 and 20. We can look deeply into the behavior of this community. We’ve already created the dictionary that contains all the movies and the performers acting in every movie. Using these movies we can create a bipartite graph.


Marshall, Scarlett
Dasz, Steven
Chan, Juju
The Hope Within (2009)
Unconditional Love (2010)
Taylor, Stuart (X)
White Cobra (2014)
McKay, Hannah
Call of Babylon (2012)
Sandison, Martin
A Schoolboy Error Production (2009)
Inner Joy of a Broken Heart (2011)
Life of a Spy (2012)
Legion of Evil (2010)
Street Fight (2012)
Boycie (2011)
Joiner, Craig
Hislop, Tom
Kilpatrick, Kayleigh
The Shadow of Death
McKay, Reuben
Noble, John-William
Wilkie, Graeme
Is This It? (2012)
Losers in Love (2011)
Moir, Shaun
Booze Culture (2012)
Love House (2011)
Simpson, Julia (II)
Be My Valentine (2011)
The Book of Life (2013/I)
Fear of the Dark (2010)


Moir, Shaun
Noble, John-William
Craig
Life of a Spy (2012)
Legion of Evil (2010)
The Hope Within (2009)

According to the graph, we can define the actors having most degrees as the most important actors.

McKay, Reuben	18
Noble, John-William	17
Noble, Graeme	14
Joiner, Craig	13
Kilpatrick, Kayleigh	12
Sandison, Martin	11
Taylor, Stuart (X)	11
McKay, Hannah	11
Simpson, Julia (II)	10
Moir, Shaun	9
Hislop, Tom	9
Dasz, Steven	1
Chan, Juju	1
Marshall, Scarlett	1

The three most important actors are Reuben McKay, John-William Noble, Graeme Noble. In this community, the dominant genre, according to 8(a), is “short”, and that in 8(b) is “Romance”. Recalling how we get this community, we want to analyze the relationship between these important actors and the formation of this community. Firstly, these three actors have performed in many “Romance” movies. That’s why the frequencies of these three actors are highest. In other words, they contribute significantly to the construction of edge weights when we create the weighted undirected network graph. What’ more important, these three actors often show up together in the “Romance” movies. This can explain why the movies they acted in are clustered into the same community.

To articulate our conclusion, we look for the evidence in the “actor_movies.txt” file to support us. In the file, actor Reuben McKay has acted in 20 movies, while in 14 movies, 70% of all, Reuben McKay works with John-William Noble and Graeme Noble. The situation is the same for John-William Noble and Graeme Noble. So these three actors we extract from this community, are actually a combination that performs in many “Romance” movies.

2.3 Neighborhood Analysis of movies

We are looking for the pattern and relation between the three target movies and their neighbors. The three movies we are interested in are listed below:

Batman v Superman: Dawn of Justice (2016); *Rating: 6.6*

Mission: Impossible - Rogue Nation (2015); *Rating: 7.4*
Minions (2015); *Rating: 6.4*

Question 9

We first search neighbors of the target movies in the constructed graph. The score distributions among the neighbors of the target movies are shown in Figure 2.3.1. The average scores of the neighbor movies for the target movies are shown in Table 2.3.1. The average rating of the movies in the neighborhood is not similar to the rating of the movie whose neighbors have been extracted. Especially, there is a relatively big difference (around 1.15) between the rating of the movie "*Mission: Impossible - Rogue Nation (2015)*" and average rating of movies in its neighbor. From Figure 2.3.1, we find that the ratings of the neighboring movies are densely distributed around the actual scores of the target movies, but the means of the neighboring movies indeed deviate from the actual scores.

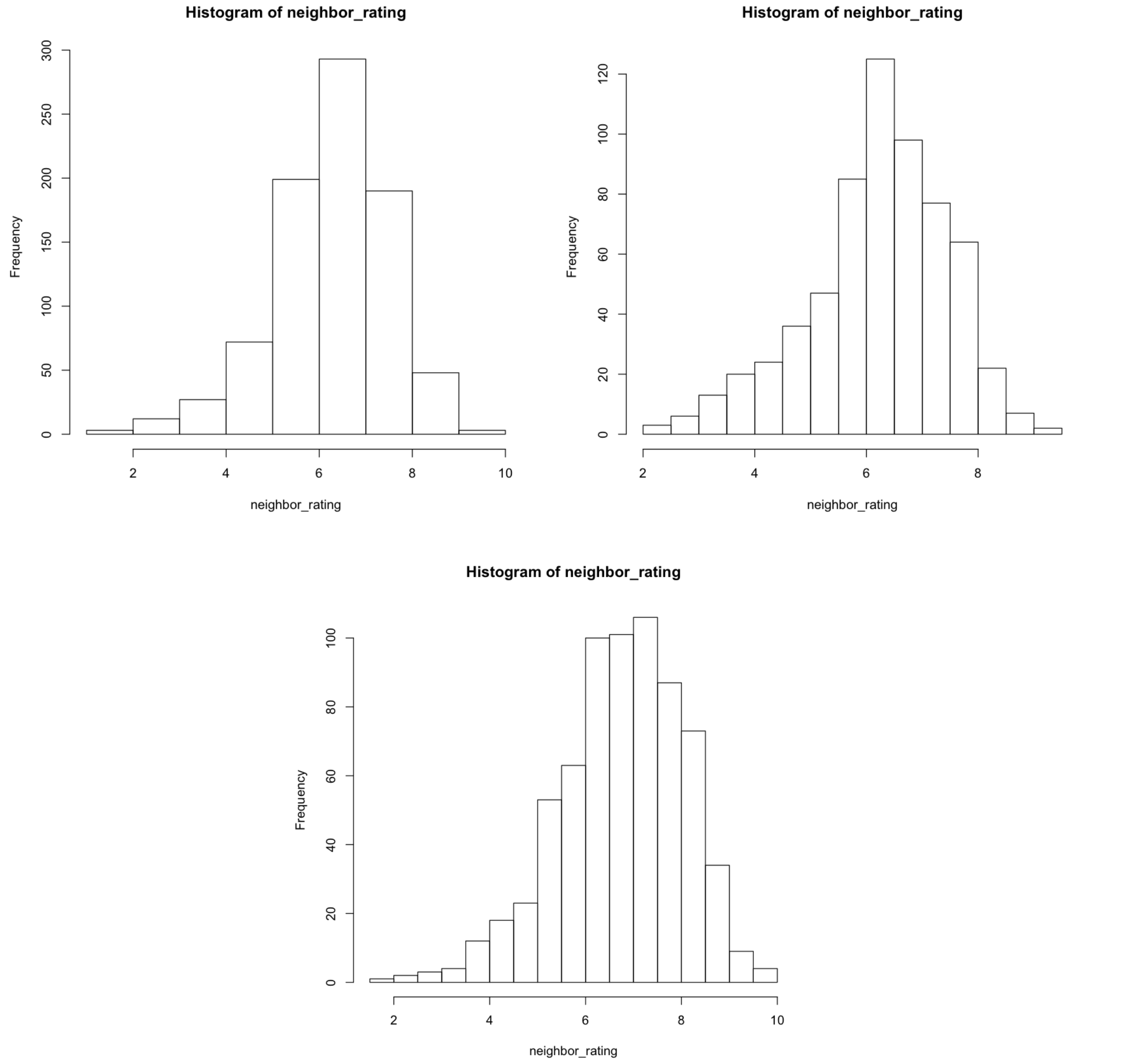


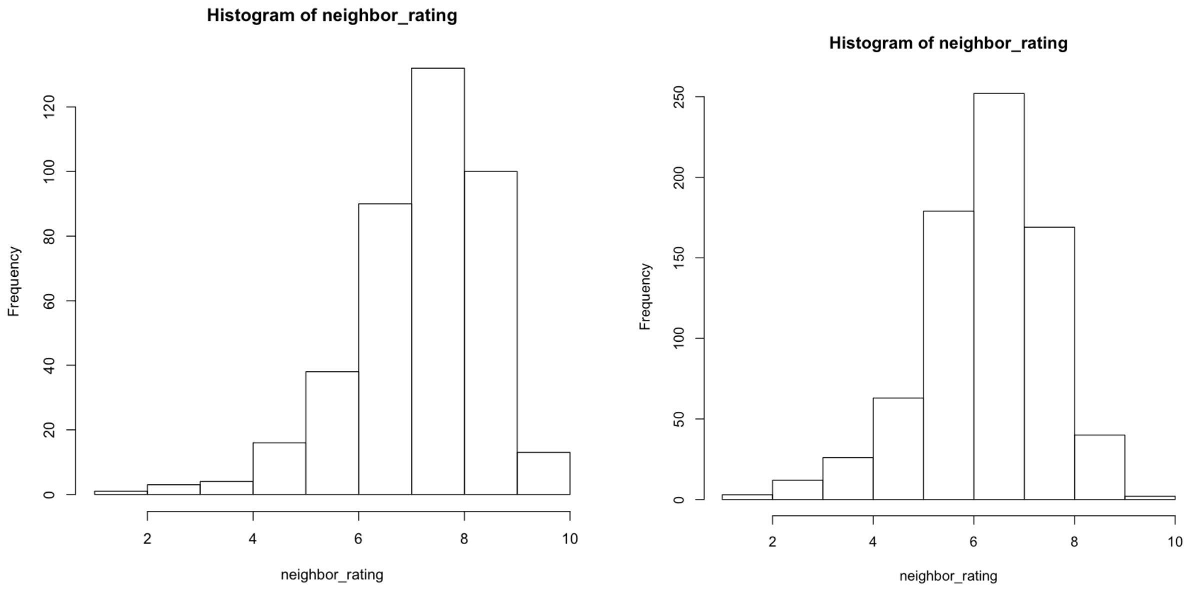
Figure 2.3.1: Score distributions of neighboring movies for the target movies. The left top one is for "Batman"; the right top one is for "Mission"; the middle bottom one is for "Minion".

Target Movies	*Batman v Superman: Dawn of Justice (2016)*	*Mission: Impossible - Rogue Nation (2015)*	*Minions (2015)*
Average rate	6.330778	6.248372	6.795087

Table 2.3.1: Average rates of the neighbor movies of the target movies.

Question 10

Due to the bad performance of making prediction of the target movies by averaging the ratings of the neighboring movies of the target movies, we want to explore a better approach to find the relation between the rating of the target movies and the ratings of their neighboring movies. We can restrict the neighborhood in the same community. In other words, we only consider the neighboring movies in the same community as the target movie. So, first, we need to find the communities, which the target movies belong to. The, we can disregard the movies in neighbor which are not in the same communities as the target movies. The rating distributions of the restricted neighboring movies are illustrated in Figure 2.3.2. The average ratings of the restricted neighboring movies for the three target movies are listed in Table 2.3.2. There is not a better match between the average rating of the movies in the restricted neighborhood and the rating of the movie whose neighbors have been extracted. The average score, instead, deviate more from the true scores for the target movies. The reason may be that the target movies, though having strong connection with the neighboring movies in the same communities, are not at the average level within community.



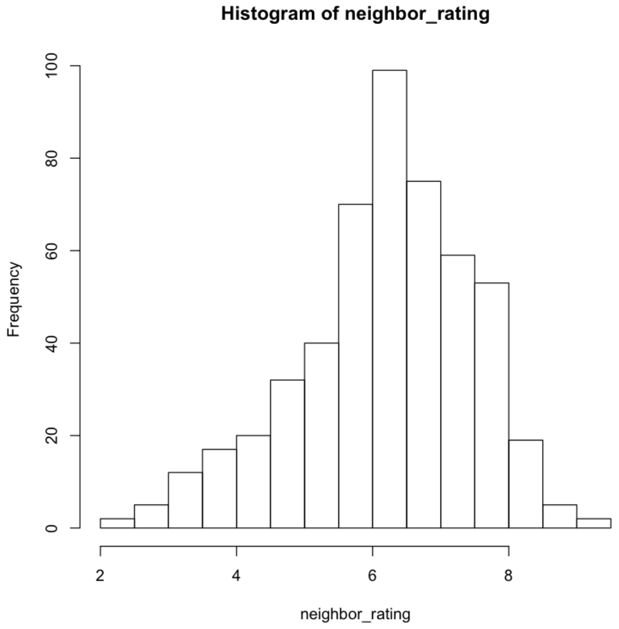


Figure 2.3.2: Score distributions of the restricted neighboring movies for the target movies. The left top one is for "Batman"; the right top one is for "Mission"; the middle bottom one is for "Minion".

Target Movies	*Batman v Superman: Dawn of Justice (2016)*	*Mission: Impossible - Rogue Nation (2015)*	*Minions (2015)*
Average rate	6.302815	6.227451	7.228463

Table 2.3.2: Average rates of the restricted neighboring movies of the target movies.

Question 11

For each neighboring movie, we extract the corresponding weight of the edge between the target movie and it. Then, we sort the weight values in descending manner, and extract the first 5 movies out of the list. The results are shown in Table 2.3.3. And, the corresponding communities which these top movies belong to are listed in Table 2.3.4.

Target movie	*Batman v Superman: Dawn of Justice (2016)*	*Mission: Impossible - Rogue Nation (2015)*	*Minions (2015)*
1	*Eloise (2015)*	*Man of Steel (2013)*	*The Lorax (2012)*
2	*The Justice League Part One (2017)*	*Phantom (2015)*	*Inside Out (2015)*
3	*Into the Storm (2014)*	*Breaking the Bank (2014)*	*Despicable Me 2 (2013)*
4	*Love and Honor (2013)*	*Suffragette (2015)*	*Up (2009)*

5	*Man of Steel (2013)*	*Now You See Me: The Second Act (2016)*	*Surf's Up (2007)*

Table 2.3.3: Top 5 neighboring movies for the target movies

Target movie	*Batman v Superman: Dawn of Justice (2016)*	*Community*	*Mission: Impossible - Rogue Nation (2015)*	*Community*	*Minions (2015)*	*Community*
1	*Eloise (2015)*	*1*	*Fan (2015)*	*1*	*The Lorax (2012)*	*1*
2	*The Justice League Part One (2017)*	*1*	*Phantom (2015)*	*1*	*Inside Out (2015)*	*1*
3	*Into the Storm (2014)*	*1*	*Breaking the Bank (2014)*	*1*	*Despicable Me 2 (2013)*	*1*
4	*Love and Honor (2013)*	*1*	*Suffragette (2015)*	*1*	*Up (2009)*	*1*
5	*Man of Steel (2013)*	*1*	*Now You See Me: The Second Act (2016)*	*1*	*Surf's Up (2007)*	*1*

Table 2.3.4: Top 5 neighboring movies and corresponding community numbers

Part 2.4 Movie network

Question 12: In order to predict the score of a certain movie, several features of the movie must be extracted. Considering the quality of a movie mainly depends on its director, actors, genres and scripts, we will take advantage of the data we have to predict movie ratings. Actors' or directors' ratings can be calculated by averaging the ratings of movies that he directed or played in. Genre can be considered as an offset. For example, generally, comedy will have a higher rating than thriller. Above all, we created the following features:

1. Director score, as x1
2. Top 10 actor score, as x2
3. Average actor score, as x3
4. The variance of all actors, as x4
5. Movie genre, as b

These five variables are merged in an array as global features. Movie genre will be treated as one-hot coding.

The regression model can be represented as r = Linear_Regression (movie_rating, feature). Here we performed a 10-fold cross validation, the result is presented below:

K	Training RMSE	Testing RMSE
1	0.835	0.527
2	0.833	0.535
3	0.833	0.548
4	0.835	0.533
5	0.834	0.536
6	0.835	0.535
7	0.834	0.532
8	0.835	0.535
9	0.836	0.525

However, the training error and testing error seems to have a 0.3 difference. We assume these are caused by data size. The data have a linear trend, however they are dispersed. If we include more samples, the set will have a more dispersed distribution. That's the reason why training errors are larger than testing error. In order to prove our hypothesis, we perform 2-fold cross validation and get the following result:

K	Training RMSE	Testing RMSE
1	0.767	0.759
2	0.758	0.767

As we can see here, when training set have almost the same number of samples as testing set, they will have a close RMSE. This proves our hypothesis. And the error of our trained model should be around 0.76 considering the whole data set.

Then we predict the movies

Name	Batman v Superman: Dawn of Justice (2016)	Mission: Impossible - Rogue Nation (2015)	Minions (2015)
real	6.6	7.4	6.4

predict	7.22	7.26	7.48

Calculating the RMSE of these three movie and we will get 0.72. This matches the expected error of our model.

Question 13

In this part, we make prediction of rating based on bipartite graph, in which movies and performers are two connected clusters. Within each cluster, there is no edge between any two members. In other words, movies are not connected with each other, while actors are not connected with each other either. So, we want to make prediction on rating of specific movie by utilizing the actors/actresses in the movie. After constructing the bipartite graph, we need to assign the weight for each edge. One way we come up with is to assign score for each movie actor/actress based on the given ratings of movies which he/she performs. Basically, we apply the following equation to determine the rating of each actor/actress:

$$m\ (average\ rating\ of\ movies\ of\ an\ actor\ or\ actress) = \frac{\sum rating\ of\ movie\ i}{total\ number\ of\ movies}$$

$$t(maximum\ rating\ of\ movie) = max\ (ratings\ of\ movies)$$

$$rate\ of\ an\ actor\ or\ actress = 0.2t + 0.8m$$

The reason why we consider the maximum rating of movies is that the rating of an actor/actress may be pulled down by some extremely low rated movie, we need to implement a simple way to counterpart this effect. The best way is to sort all movies and discard the outliers, but it is time consuming. So, we think up a comparably easy way to solve the problem. Also, the maximum rating might be far off the mean a value. Therefore, assigning a proper weight to the maximum rating(t) is very important. We choose a value between 0 and 0.5 for the weight of t.

In test time, we randomly choose 2000 rated movies and make prediction based on the method above. The RMSE score for the test set is 1.0716. Then, we make prediction on the target movies. The predicted ratings are shown as below:

	Predicted rating	Actual rating
Batman v Superman: Dawn of Justice (2016)	6.6	6.551188
Mission: Impossible - Rogue Nation (2015)	7.4	6.839698
Mission: Impossible - Rogue Nation (2015)	6.4	7.255890
RMSE score	0.591289	

Table 2.4.13: Rating prediction on the target movies and corresponding RMSE score.

This method is very naïve. First, we don't have information of what role a specific actor/actress is performing in the movie. So, it is hard to determine how much we should pay attention to the rating of actor/actress when we are calculating the rating of the movie. Second, the rating of each actor/actress may not be solid. Here, we assume each actor/actress contributes equally to the movie. Thus, we just average the ratings of movies where an actor/actress performs. Based on the two major disadvantages, the bipartite graph method is very limited. It is very like to make a bad prediction. Therefore, it is reasonable that this rating mechanism is worse than that in Question 12. The RMSE score for test set is worse, however, the rmse score for the three target movies is lower than that in Question 12. This might be that the second mechanism fits the target moveis better by accident. So, RMSE score for this kind of small test set (only containing three target movies) cannot tell much about the performance of the mechanism.