Project 2 Report

EE 232E
June 9, 2017

Gourav Khadge
Eun Sun Lee
Yusi Ou

I. Introduction

IMDB is a movie rating website which provides user rating information about both movies and actors. In this project, we created and explored networks constructed from IMDB ratings for 246372 movies and 243999 actors. We explored the relationships between movie and actor ratings and popularity through pagerank, community finding and genre finding. The features extracted in this study were used to predict ratings for three movies that do not appear in the network through multiple-parameter linear regression.

II. Problem 1

The provided datafiles were pre-processed and parsed in python to clean the data and extract Actor and Movie networks. First “actor_movies.txt” and “actress_movies.txt” were read into python line by line, and split into tokens delimited by double tabs. Any tokens that were pure whitespace were ignored. We then removed from the dataset any actors who performed in less than 5 movies. Then we used a regular expression to extract the movie title from the movie string. This was necessary because many movies in the files had flair associated with the actor such as “Water Lords (2013) (as David Jones)”. To remove the flair, we took advantage of the fact that almost every movie string ended with a year in parentheses. (2013) in the previous example. The characters after the year were typically specific to the particular actor and could be ignored. Some movies had a more complex date such as “A Mother's Love (2011/II)” which include “/II” to indicate that it’s the second part of a movie released in the same year. Other movies had unknown years represented as question marks such as “The Ghost Experiment 3D (????)”. The regular expression used to extract the movies incorporated all these possibilities for movie dates that delimit the end of the movie title and was given by the expression: `'^.+\(([0-9][0-9][0-9][0-9]|\?\?\?\?).*?\)'`. Several movies did not end in any expression of year of release. In general, it was found that these movie titles were rare, and most likely the result of a data corruption in the file unrelated to our processing. For these movie strings, the regex extraction would fail. In these cases, the movie string was just stripped of trailing whitespace and treated as a title anyway.

The data was directly read into dictionaries where the keys represented actors and the value associated with each key was an array of movies the actor had been in. A corresponding movie dictionary was created by inverting this dictionary so the keys were movies and the value associated with each key was an array of actors that were in each movie.

For the purposes of reading the network files into R, all movie titles and actors names were stripped of all whitespace, since the read_graph function in R reads in whitespace delimited edgelist files.

III. Problem 2

The processed data from the previous problem was used to construct the Actor network edge weights in Python, which were used to construct the graph to be imported and processed in R. The weights were calculated by looping through all actor and movie combinations using the following method. This edge weight assignment is a measure of the number of shared movies between two actors. The same process was used to calculate edge weights for the Movie network, which measures the number of shared actors between two movies in an analogous way.

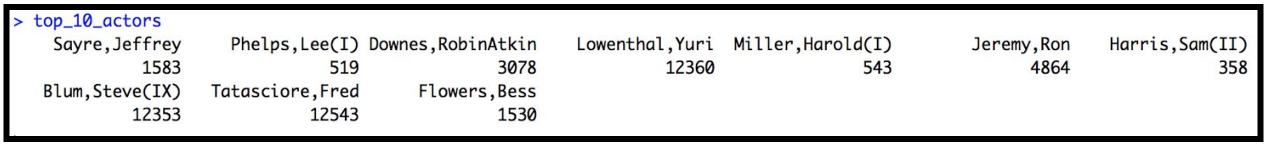
V = **all actors/actressess in list**

S_i = { $m|i \in V$, m **is a movie in which** i **has acted**}

$E = \{(i,j)|i,j \in V, S_i \cap S_j \neq \emptyset\}$ **and for each directed Edge** $i \rightarrow j$**, a weight is assigned as** $\frac{|S_i \cap S_j|}{|S_i|}$.

IV. Problem 3

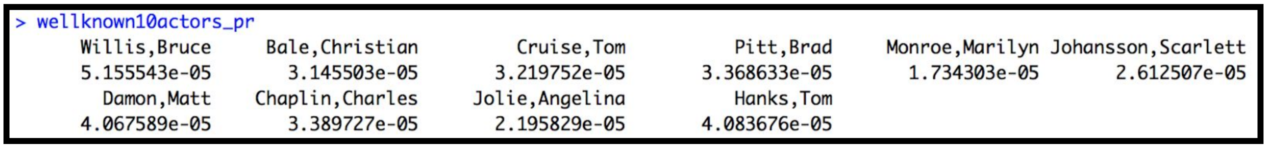
Pagerank algorithm is run on the actor/actress network and the top 10 actors with high pageranks are given in Figure 3.1. However, I do not recognize any of their names. Searching for the movies these actors played, Robin Atkin Downes was a voice actor in many famous movies such as Batman series or Iron Man movies. Steve Blum is another voice actor played in many famous movies. Ron Jeremy was ranked No.1 in AVN "50 Top Porn Stars of all Time."



```
> top_10_actors
    Sayre,Jeffrey      Phelps,Lee(I) Downes,RobinAtkin     Lowenthal,Yuri  Miller,Harold(I)         Jeremy,Ron     Harris,Sam(II)
             1583                519              3078              12360               543               4864                358
  Blum,Steve(IX)   Tatasciore,Fred      Flowers,Bess
            12353             12543              1530
```

Figure 3.1 Top 10 high pagerank actors

Below is the list of actors who we consider famous and their pageranks. Their pageranks are surprisingly much lower than the actors in Figure 3.1.



```
> wellknown10actors_pr
      Willis,Bruce     Bale,Christian         Cruise,Tom          Pitt,Brad    Monroe,Marilyn Johansson,Scarlett
      5.155543e-05       3.145503e-05       3.219752e-05       3.368633e-05      1.734303e-05       2.612507e-05
        Damon,Matt    Chaplin,Charles    Jolie,Angelina          Hanks,Tom
      4.067589e-05       3.389727e-05       2.195829e-05       4.083676e-05
```

Figure 3.2 Pagerank of 10 movie celebrities

Comparing Figure 3.1 and Figure 3.2, the voice actors have high pagerank because the number of movies they played is much higher than that of other actors. Moreover, the actors in top 10 pageranks seem to be the ones from each field. Considering how pagerank works, it makes sense the higher pagerank is given to the best actor in each field. For example, Ron Jeremy has high pagerank not just because he is a famous porn actor. In my opinion, Brad Pitt is more popular than him to general public. However, within his field, or maybe, within the movies of adult genre, he is outstandingly famous compared to other porn stars.

V. Problem 4

Re-arranging the data in python, we reconstruct the graph using movies as nodes. All movie nodes with less than 5 actors are purged from the network. We weighted the connections based on the Jaccard index of the actor sets between any two movie nodes. Since the Jaccard index is not directional, the resulting graph is undirected. This constructed graph is exported to a tab separated format for analysis in R

VI. Problem 5

Communities were detected using the Fast Greedy Newman method in R. Through this 63 communities were found of various sizes. Each community was analyzed to see if they contained large numbers of a particular genre. For each community, if more than 20% of the movies were of a certain genre, that genre was documented as the genre of the entire community. If there were multiple genres that broached the 20% threshold, they were all added as joint labels. If no genre reached this threshold, the community was unlabeled. In Table 5.1, the genre labels are attached as two element structures. The first element is the genre, and the second element is the proportion of movies in that community that are of that genre. The most genre labels any community had was three. No community of substantial size (>500) was dominated (>50%) by any genre. Community 7 had the highest amount of domination by a single genre, having 40.1% of its movies being in the “Adult” genre. We were not able to extract significant meaning out of these genre labels besides the intuition that these communities were had many of the same actors in various movies, which tended to act in movies of the same genre, but not strictly so.

Table 5.1 Community Genre labels

Community	Size	1st Genre	2nd Genre	3rd Genre
1	7009	['Drama', 0.26123555428734485]		
2	9822	['Drama', 0.23365913255956017]		
3	9141			
4	64629	['Short', 0.2510018722245432]		
5	5615	['Drama', 0.3285841495992876]		
6	6315	['Drama', 0.2804433887569279]		

7	7120	['Adult', 0.4007022471910112]		
8	30441	['Drama', 0.24230478630793995]		
9	7970	['Drama', 0.365495608531995]		
10	6882	['Drama', 0.2972972972972973]		
11	8161	['Drama', 0.25143977453743416]		
12	2392	['Drama', 0.3290133779264214]	['Comedy', 0.2763377926421405]	
13	16650	['Drama', 0.21543543543543545]		
14	36944	['Short', 0.3208098744045041]		
15	1238	['Drama', 0.3602584814216478]	['Romance', 0.21405492730210016]	
16	1836	['Drama', 0.38235294117647056]		
17	10590	['Drama', 0.253918791312559]		
18	912	['Drama', 0.3081140350877193]		
19	1089	['Drama', 0.2800734618916437]		
20	4452	['Drama', 0.2722371967654987]		
21	874	['Romance', 0.21967963386727687]		
22	3987	['Drama', 0.31677953348382243]	['Romance', 0.2761474793077502]	
23	883			
24	14	['Thriller', 0.7857142857142857]		
25	20	['Short', 0.5]		
26	6	['Short', 1.0]		
27	24	['Drama', 0.375]		
28	6	['Thriller', 0.6666666666666666]	['Sci-Fi', 0.3333333333333333]	
29	140	['Drama', 0.5]		

30	471			
31	2	['Short', 1.0]		
32	10	['Short', 0.9]		
33	499	['Drama', 0.5470941883767535]		
34	23	['Short', 1.0]		
35	7	['Horror', 0.42857142857142855]	['Thriller', 0.42857142857142855]	
36	6	['Horror', 0.8333333333333334]		
37	5	['Horror', 0.6]		
38	27			
39	7	['Comedy', 0.42857142857142855]	['Short', 0.2857142857142857]	
40	6	['Short', 0.6666666666666666]	['Action', 0.3333333333333333]	
41	8	['Romance', 0.625]	['Comedy', 0.25]	
42	8	['Documentary', 1.0]		
43	7	['Short', 1.0]		
44	8	['Short', 1.0]		
45	12	['Short', 0.5833333333333334]		
46	7	['Short', 0.8571428571428571]		
47	10	['Short', 0.9]		
48	7	['Short', 1.0]		
49	6	['Short', 1.0]		
50	9	['Short', 1.0]		
51	8	['Thriller', 0.375]	['Drama', 0.25]	
52	7	['Short', 0.42857142857142855]	['Sci-Fi', 0.2857142857142857]	

53	7	['Adventure', 0.7142857142857143]	['Thriller', 0.2857142857142857]	
54	5	['Short', 1.0]		
55	5	['Short', 0.6]		
56	5	['Short', 0.8]		
57	5	['Short', 0.6]		
58	5	['Short', 0.4]	['Crime', 0.4]	
59	4	['Comedy', 0.5]	['Thriller', 0.25]	['Short', 0.25]
60	6	['Short', 1.0]		
61	3	['Short', 1.0]		
62	3	['Short', 1.0]		
63	2	['Short', 1.0]		

VII. Problem 6

The following movies were added to the network and their neighbors were sorted by edge weight. The top five highest weighted neighbor movies are collected in the tables below. The community of each neighbor movie was found using the communities found with the fast greedy algorithm. All but one of the top neighbors of all three movies was found to be in Community 4.

Batman v Superman: Dawn of Justice (2016)

Nearest Neighbor	Eloise (2015)	Into the Storm (2014)	The Justice League Part One (2017)	Grain (2015)	Man of Steel (2013)
Community	4	4	4	8	4

Table 6.1 Nearest neighbors of Batman v Superman: Dawn of Justice based on edge weight.

Mission: Impossible - Rogue Nation (2015)

Nearest Neighbor	Fan (2015)	Phantom (2015)	Breaking the Bank (2014)	The Program (2015/II)	The Rise of the Krays

					(2015)
Community	4	4	4	4	4

Table 6.2 Nearest neighbors of Mission: Impossible - Rogue Nation based on edge weight.

Minions (2015)

Nearest Neighbor	The Lorax (2012)	Inside Out (2015)	Up (2009)	Despicable Me 2 (2013)	Surf's Up (2007)
Community	4	4	4	4	4

Table 6.3 Nearest neighbors of Minions based on edge weight.

VIII. Problem 7

The list of sorted neighbors from Problem 6 and their associated ratings from the rating list data was used to derive a rating predictor for the three movies. The rating list does not contain movies after the year 2015, and therefore some top neighbor ratings were blank (NA values). In these cases, the next top neighbor with a valid rating was used in order to calculate the mean rating. The ratings below are calculated by using the mean and weighted means of the top 20 neighbors and all neighbors of each movie. The weights used are the edge weights between the core movie and each respective neighbor.

	IMDB Rating	Mean (20 top neighbors)	Weighted Mean (20 top neighbors)	Mean (all neighbors)	Weighted Mean (all neighbors)
Batman v Superman: Dawn of Justice (2016)	6.7	6.68 (0.03% error)	6.59 (1.60% error)	6.36 (5.10% error)	6.36 (5.10% error)
Mission: Impossible - Rogue Nation (2015)	7.4	7.74 (4.59% error)	7.84 (5.95% error)	6.16 (16.8% error)	7.84 (5.95% error)

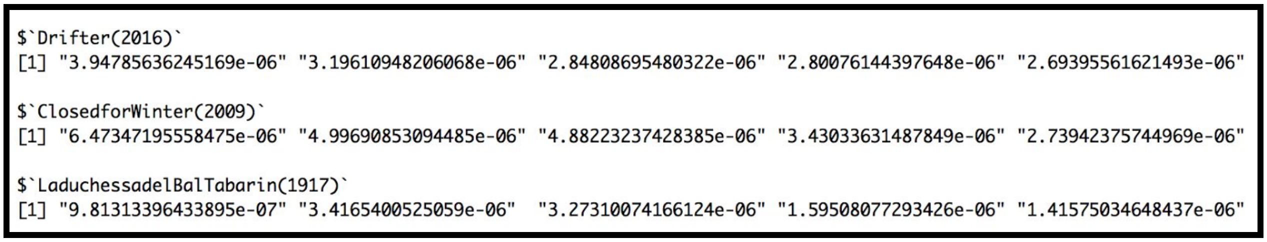
Minions (2015)	6.4	7.24 (13.1% error)	7.26 (13.4% error)	6.80 (6.25% error)	6.92 (8.13% error)

Table 7.1 Predicted ratings for three new movies using four functions.

IX. Problem 8

Similar to problem 7, the rating of the three movies are predicted using linear regression. The first feature is top 5 pageranks of the actors in each movie. The second feature is 101 boolean values of top 100 directors list. Then the third feature is the genre of each movie.

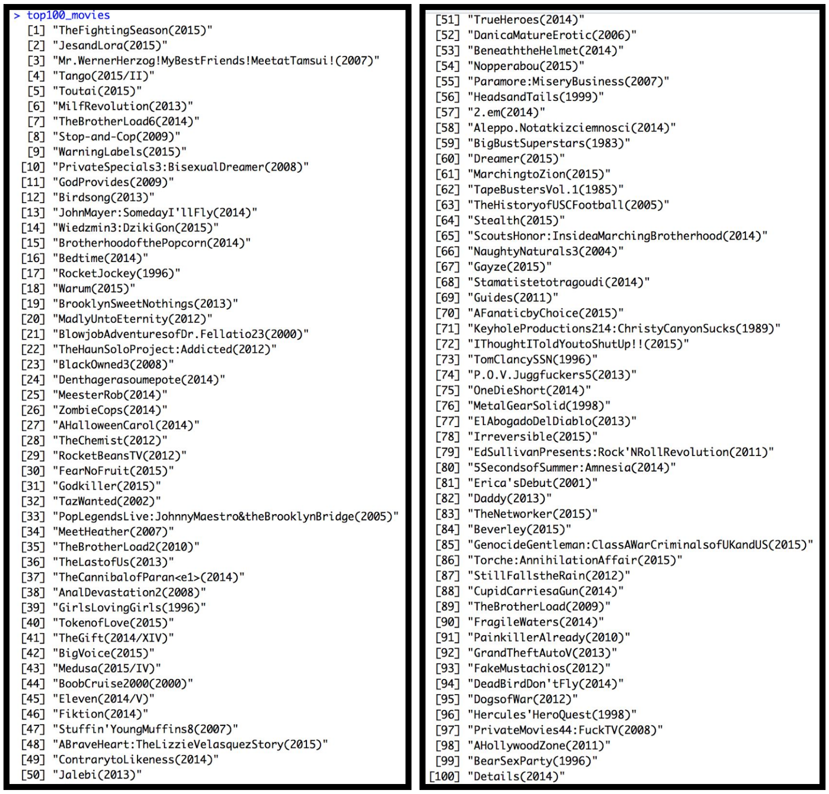
From problem 3, pagerank algorithm is run on the actor/actress network. Using the pagerank of each actor and actress found, the pagerank of all actors in each movie can be found. Then, top 5 pageranks are sorted to create the top 5 pagerank feature as shown below in Figure 8.1.



```
$`Drifter(2016)`
[1] "3.94785636245169e-06" "3.19610948206068e-06" "2.84808695480322e-06" "2.80076144397648e-06" "2.69395561621493e-06"

$`ClosedforWinter(2009)`
[1] "6.47347195558475e-06" "4.99690853094485e-06" "4.88223237428385e-06" "3.43033631487849e-06" "2.73942375744969e-06"

$`LaduchessadelBalTabarin(1917)`
[1] "9.81313396433895e-07" "3.4165400525059e-06"  "3.27310074166124e-06" "1.59508077293426e-06" "1.41575034648437e-06"
```

Figure 8.1 Top 5 pageranks of actors in each movie

Then, the next feature is 101 boolean vector which tells if the director is one of the top100 directors or not. The list of top 100 directors can be found either from “IMDb top 250” or movie_rating.txt file. Figure 8.2 is the list of top rated 100 movies extracted from movie_rating.txt file.



```
> top100_movies
  [1] "TheFightingSeason(2015)"
  [2] "JesandLora(2015)"
  [3] "Mr.WernerHerzog!MyBestFriends!MeetatTamsui!(2007)"
  [4] "Tango(2015/II)"
  [5] "Toutai(2015)"
  [6] "MilfRevolution(2013)"
  [7] "TheBrotherLoad6(2014)"
  [8] "Stop-and-Cop(2009)"
  [9] "WarningLabels(2015)"
 [10] "PrivateSpecials3:BisexualDreamer(2008)"
 [11] "GodProvides(2009)"
 [12] "Birdsong(2013)"
 [13] "JohnMayer:SomedayI'llFly(2014)"
 [14] "Wiedzmin3:DzikiGon(2015)"
 [15] "BrotherhoodofthePopcorn(2014)"
 [16] "Bedtime(2014)"
 [17] "RocketJockey(1996)"
 [18] "Warum(2015)"
 [19] "BrooklynSweetNothings(2013)"
 [20] "MadlyUntoEternity(2012)"
 [21] "BlowjobAdventuresofDr.Fellatio23(2000)"
 [22] "TheHaunSoloProject:Addicted(2012)"
 [23] "BlackOwned3(2008)"
 [24] "Denthagerasoumepote(2014)"
 [25] "MeesterRob(2014)"
 [26] "ZombieCops(2014)"
 [27] "AHalloweenCarol(2014)"
 [28] "TheChemist(2012)"
 [29] "RocketBeansTV(2012)"
 [30] "FearNoFruit(2015)"
 [31] "Godkiller(2015)"
 [32] "TazWanted(2002)"
 [33] "PopLegendsLive:JohnnyMaestro&theBrooklynBridge(2005)"
 [34] "MeetHeather(2007)"
 [35] "TheBrotherLoad2(2010)"
 [36] "TheLastofUs(2013)"
 [37] "TheCannibalofParan<e1>(2014)"
 [38] "AnalDevastation2(2008)"
 [39] "GirlsLovingGirls(1996)"
 [40] "TokenofLove(2015)"
 [41] "TheGift(2014/XIV)"
 [42] "BigVoice(2015)"
 [43] "Medusa(2015/IV)"
 [44] "BoobCruise2000(2000)"
 [45] "Eleven(2014/V)"
 [46] "Fiktion(2014)"
 [47] "Stuffin'YoungMuffins8(2007)"
 [48] "ABraveHeart:TheLizzieVelasquezStory(2015)"
 [49] "ContrarytoLikeness(2014)"
 [50] "Jalebi(2013)"
```

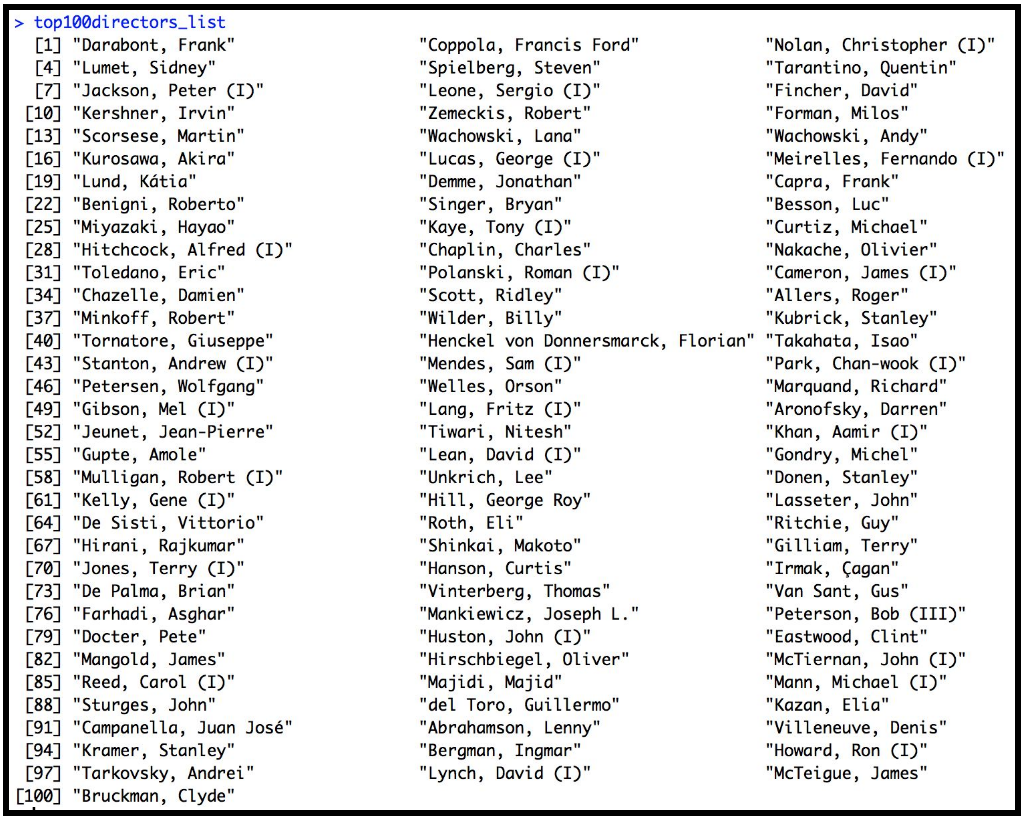
```
 [51] "TrueHeroes(2014)"
 [52] "DanicaMatureErotic(2006)"
 [53] "BeneaththeHelmet(2014)"
 [54] "Nopperabou(2015)"
 [55] "Paramore:MiseryBusiness(2007)"
 [56] "HeadsandTails(1999)"
 [57] "2.em(2014)"
 [58] "Aleppo.Notatkizciemnosci(2014)"
 [59] "BigBustSuperstars(1983)"
 [60] "Dreamer(2015)"
 [61] "MarchingtoZion(2015)"
 [62] "TapeBustersVol.1(1985)"
 [63] "TheHistoryofUSCFootball(2005)"
 [64] "Stealth(2015)"
 [65] "ScoutsHonor:InsideaMarchingBrotherhood(2014)"
 [66] "NaughtyNaturals3(2004)"
 [67] "Gayze(2015)"
 [68] "Stamatistetotragoudi(2014)"
 [69] "Guides(2011)"
 [70] "AFanaticbyChoice(2015)"
 [71] "KeyholeProductions214:ChristyCanyonSucks(1989)"
 [72] "IThoughtIToldYoutoShutUp!!(2015)"
 [73] "TomClancySSN(1996)"
 [74] "P.O.V.Juggfuckers5(2013)"
 [75] "OneDieShort(2014)"
 [76] "MetalGearSolid(1998)"
 [77] "ElAbogadoDelDiablo(2013)"
 [78] "Irreversible(2015)"
 [79] "EdSullivanPresents:Rock'NRollRevolution(2011)"
 [80] "5SecondsofSummer:Amnesia(2014)"
 [81] "Erica'sDebut(2001)"
 [82] "Daddy(2013)"
 [83] "TheNetworker(2015)"
 [84] "Beverley(2015)"
 [85] "GenocideGentleman:ClassAWarCriminalsofUKandUS(2015)"
 [86] "Torche:AnnihilationAffair(2015)"
 [87] "StillFallstheRain(2012)"
 [88] "CupidCarriesaGun(2014)"
 [89] "TheBrotherLoad(2009)"
 [90] "FragileWaters(2014)"
 [91] "PainkillerAlready(2010)"
 [92] "GrandTheftAutoV(2013)"
 [93] "FakeMustachios(2012)"
 [94] "DeadBirdDon'tFly(2014)"
 [95] "DogsofWar(2012)"
 [96] "Hercules'HeroQuest(1998)"
 [97] "PrivateMovies44:FuckTV(2008)"
 [98] "AHollywoodZone(2011)"
 [99] "BearSexParty(1996)"
[100] "Details(2014)"
```

Figure 8.2 Top 100 movies created from movie_rating.txt

Surprisingly, most of the movies are not those well-known masterpieces. For example, movie “The fighting season (2015)” has the rating of 9.9 from the movie_rating.txt file. From IMDb movie website below, the rating of “The fighting season (2015)” is 8.1.

http://www.imdb.com/title/tt2699466/

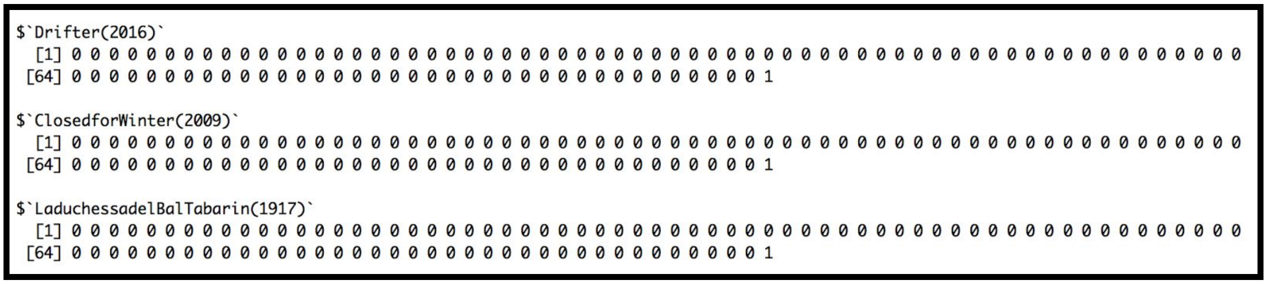
It is still high. However, the number of raters are only 17 users. This comes to the conclusion that relying on “IMDb top 250” would be more accurate in terms of extracting the directors who make highly reviewed movies. Thus, top 100 directors were extracted from “IMDb top 250” as shown in Figure 8.3.



```
> top100directors_list
  [1] "Darabont, Frank"                "Coppola, Francis Ford"             "Nolan, Christopher (I)"
  [4] "Lumet, Sidney"                  "Spielberg, Steven"                 "Tarantino, Quentin"
  [7] "Jackson, Peter (I)"             "Leone, Sergio (I)"                 "Fincher, David"
 [10] "Kershner, Irvin"                "Zemeckis, Robert"                  "Forman, Milos"
 [13] "Scorsese, Martin"               "Wachowski, Lana"                   "Wachowski, Andy"
 [16] "Kurosawa, Akira"                "Lucas, George (I)"                 "Meirelles, Fernando (I)"
 [19] "Lund, Kátia"                    "Demme, Jonathan"                   "Capra, Frank"
 [22] "Benigni, Roberto"               "Singer, Bryan"                     "Besson, Luc"
 [25] "Miyazaki, Hayao"                "Kaye, Tony (I)"                    "Curtiz, Michael"
 [28] "Hitchcock, Alfred (I)"          "Chaplin, Charles"                  "Nakache, Olivier"
 [31] "Toledano, Eric"                 "Polanski, Roman (I)"               "Cameron, James (I)"
 [34] "Chazelle, Damien"               "Scott, Ridley"                     "Allers, Roger"
 [37] "Minkoff, Robert"                "Wilder, Billy"                     "Kubrick, Stanley"
 [40] "Tornatore, Giuseppe"            "Henckel von Donnersmarck, Florian" "Takahata, Isao"
 [43] "Stanton, Andrew (I)"            "Mendes, Sam (I)"                   "Park, Chan-wook (I)"
 [46] "Petersen, Wolfgang"             "Welles, Orson"                     "Marquand, Richard"
 [49] "Gibson, Mel (I)"                "Lang, Fritz (I)"                   "Aronofsky, Darren"
 [52] "Jeunet, Jean-Pierre"            "Tiwari, Nitesh"                    "Khan, Aamir (I)"
 [55] "Gupte, Amole"                   "Lean, David (I)"                   "Gondry, Michel"
 [58] "Mulligan, Robert (I)"           "Unkrich, Lee"                      "Donen, Stanley"
 [61] "Kelly, Gene (I)"                "Hill, George Roy"                  "Lasseter, John"
 [64] "De Sisti, Vittorio"             "Roth, Eli"                         "Ritchie, Guy"
 [67] "Hirani, Rajkumar"               "Shinkai, Makoto"                   "Gilliam, Terry"
 [70] "Jones, Terry (I)"               "Hanson, Curtis"                    "Irmak, Çagan"
 [73] "De Palma, Brian"                "Vinterberg, Thomas"                "Van Sant, Gus"
 [76] "Farhadi, Asghar"                "Mankiewicz, Joseph L."             "Peterson, Bob (III)"
 [79] "Docter, Pete"                   "Huston, John (I)"                  "Eastwood, Clint"
 [82] "Mangold, James"                 "Hirschbiegel, Oliver"              "McTiernan, John (I)"
 [85] "Reed, Carol (I)"                "Majidi, Majid"                     "Mann, Michael (I)"
 [88] "Sturges, John"                  "del Toro, Guillermo"               "Kazan, Elia"
 [91] "Campanella, Juan José"          "Abrahamson, Lenny"                 "Villeneuve, Denis"
 [94] "Kramer, Stanley"                "Bergman, Ingmar"                   "Howard, Ron (I)"
 [97] "Tarkovsky, Andrei"              "Lynch, David (I)"                  "McTeigue, James"
[100] "Bruckman, Clyde"
```

Figure 8.3 Top 100 directors list created from "IMDb top 250"

Moreover, within top 100 movies, there are movies made by the same directors. The top 100 directors thus were extracted from the movies beyond top 100. Additionally, if some movies were co-directed, they were considered as two separate directors in the list. Since we are creating 101 boolean values, the order of these directors in the list did not add any weight to the linear regression. Finally, from the top 100 directors list, a feature as shown in Figure 8.4 was created.

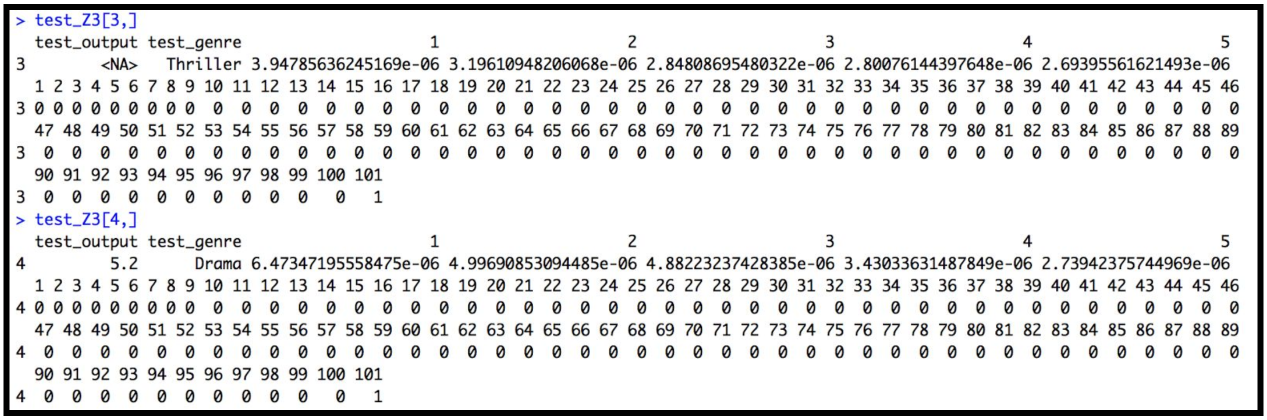


```
$`Drifter(2016)`
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [64] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

$`ClosedforWinter(2009)`
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [64] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

$`LaduchessadelBalTabarin(1917)`
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [64] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

Figure 8.4 101 boolean value of each movie created from top 100 directors

Lastly, as mentioned above, genre feature is created. This feature contains the genre of each movie.

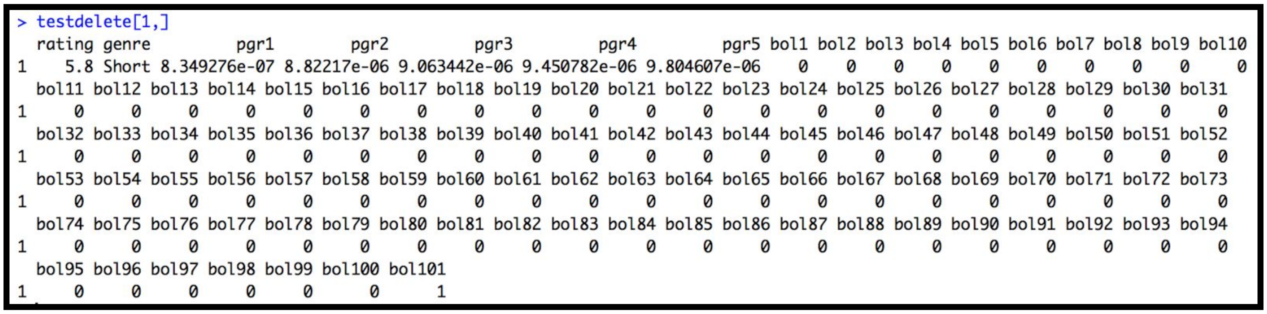
After collecting data of all the features, these features were combined in a matrix. However, there are some movies that did not have ratings as shown in Figure 8.5. These movies were taken out from the data since there is no rating to train linear regression with.



```
> test_Z3[3,]
  test_output test_genre                    1                    2                    3                    4                    5
3        <NA>   Thriller 3.94785636245169e-06 3.19610948206068e-06 2.84808695480322e-06 2.80076144397648e-06 2.69395561621493e-06
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
3 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  90 91 92 93 94 95 96 97 98 99 100 101
3  0  0  0  0  0  0  0  0  0  0   0   1
> test_Z3[4,]
  test_output test_genre                    1                    2                    3                    4                    5
4         5.2      Drama 6.47347195558475e-06 4.99690853094485e-06 4.88223237428385e-06 3.43033631487849e-06 2.73942375744969e-06
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
4 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  90 91 92 93 94 95 96 97 98 99 100 101
4  0  0  0  0  0  0  0  0  0  0   0   1
```

Figure 8.5 Example of moives that do not have ratings

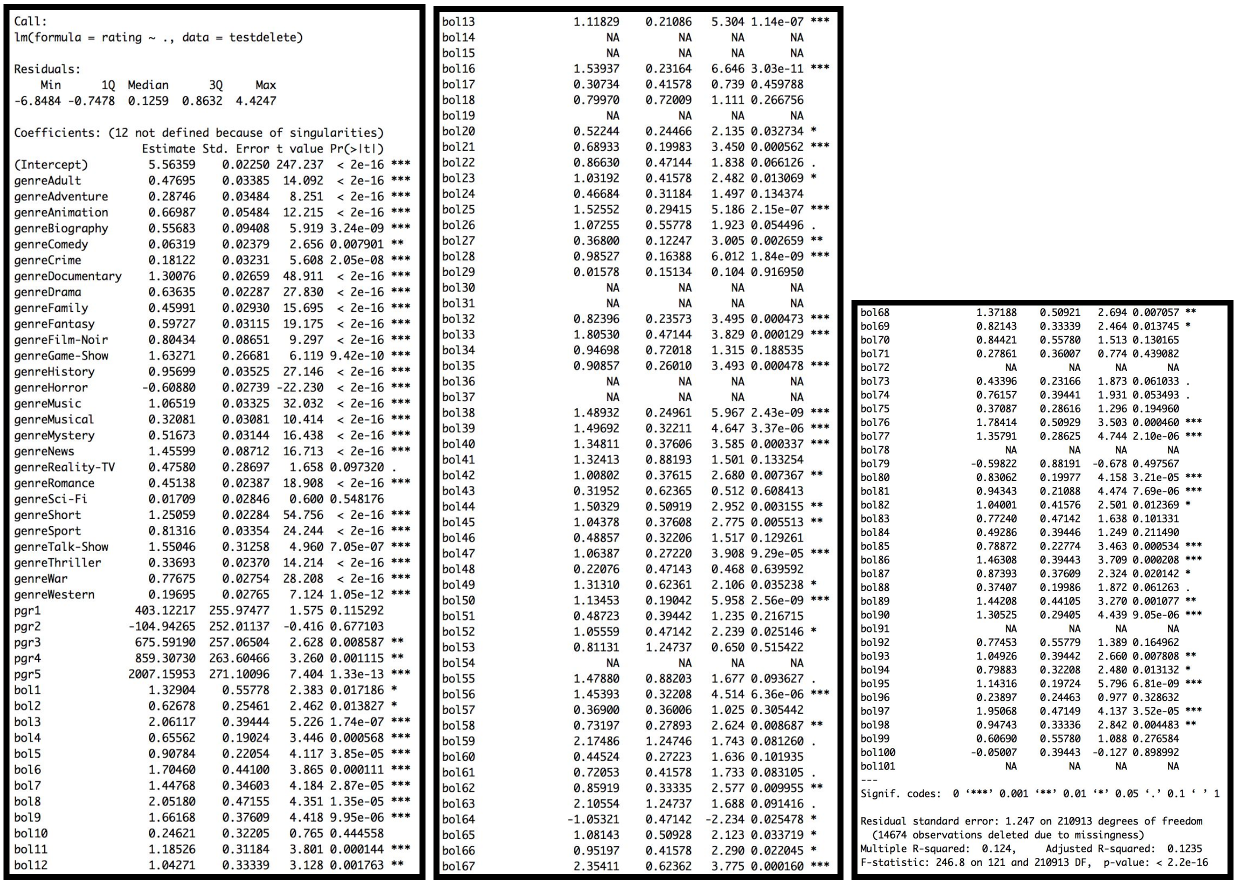
Thus final matrix has a form as shown in Figure 8.6 which has rating in the first column, genre, top 5 pageranks, each element of 101 boolean vectors as follow.



```
> testdelete[1,]
  rating genre         pgr1        pgr2         pgr3         pgr4         pgr5 bol1 bol2 bol3 bol4 bol5 bol6 bol7 bol8 bol9 bol10
1    5.8 Short 8.349276e-07 8.82217e-06 9.063442e-06 9.450782e-06 9.804607e-06    0    0    0    0    0    0    0    0    0     0
  bol11 bol12 bol13 bol14 bol15 bol16 bol17 bol18 bol19 bol20 bol21 bol22 bol23 bol24 bol25 bol26 bol27 bol28 bol29 bol30 bol31
1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
  bol32 bol33 bol34 bol35 bol36 bol37 bol38 bol39 bol40 bol41 bol42 bol43 bol44 bol45 bol46 bol47 bol48 bol49 bol50 bol51 bol52
1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
  bol53 bol54 bol55 bol56 bol57 bol58 bol59 bol60 bol61 bol62 bol63 bol64 bol65 bol66 bol67 bol68 bol69 bol70 bol71 bol72 bol73
1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
  bol74 bol75 bol76 bol77 bol78 bol79 bol80 bol81 bol82 bol83 bol84 bol85 bol86 bol87 bol88 bol89 bol90 bol91 bol92 bol93 bol94
1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
  bol95 bol96 bol97 bol98 bol99 bol100 bol101
1     0     0     0     0     0      0      1
```

Figure 8.6 Format of final matrix of all features combined.

To create a linear regression model with multiple data, we use lm function as following: *model = lm(rating~ ., data= testdelete)*. The model’s summary is shown in Figure 8.7. For bol101, for example, the coefficient in a regression is NA because it is linearly related to the other variables.



```
Call:
lm(formula = rating ~ ., data = testdelete)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.8484 -0.7478  0.1259  0.8632  4.4247 

Coefficients: (12 not defined because of singularities)
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)         5.56359    0.02250 247.237  < 2e-16 ***
genreAdult          0.47695    0.03385  14.092  < 2e-16 ***
genreAdventure      0.28746    0.03484   8.251  < 2e-16 ***
genreAnimation      0.66987    0.05484  12.215  < 2e-16 ***
genreBiography      0.55683    0.09408   5.919 3.24e-09 ***
genreComedy         0.06319    0.02379   2.656 0.007901 ** 
genreCrime          0.18122    0.03231   5.608 2.05e-08 ***
genreDocumentary    1.30076    0.02659  48.911  < 2e-16 ***
genreDrama          0.63635    0.02287  27.830  < 2e-16 ***
genreFamily         0.45991    0.02930  15.695  < 2e-16 ***
genreFantasy        0.59727    0.03115  19.175  < 2e-16 ***
genreFilm-Noir      0.80434    0.08651   9.297  < 2e-16 ***
genreGame-Show      1.63271    0.26681   6.119 9.42e-10 ***
genreHistory        0.95699    0.03525  27.146  < 2e-16 ***
genreHorror        -0.60880    0.02739 -22.230  < 2e-16 ***
genreMusic          1.06519    0.03325  32.032  < 2e-16 ***
genreMusical        0.32081    0.03081  10.414  < 2e-16 ***
genreMystery        0.51673    0.03144  16.438  < 2e-16 ***
genreNews           1.45599    0.08712  16.713  < 2e-16 ***
genreReality-TV     0.47580    0.28697   1.658 0.097320 .  
genreRomance        0.45138    0.02387  18.908  < 2e-16 ***
genreSci-Fi         0.01709    0.02846   0.600 0.548176    
genreShort          1.25059    0.02284  54.756  < 2e-16 ***
genreSport          0.81316    0.03354  24.244  < 2e-16 ***
genreTalk-Show      1.55046    0.31258   4.960 7.05e-07 ***
genreThriller       0.33693    0.02370  14.214  < 2e-16 ***
genreWar            0.77675    0.02754  28.208  < 2e-16 ***
genreWestern        0.19695    0.02765   7.124 1.05e-12 ***
pgr1              403.12217  255.97477   1.575 0.115292    
pgr2             -104.94265  252.01137  -0.416 0.677103    
pgr3              675.59190  257.06504   2.628 0.008587 ** 
pgr4              859.30730  263.60466   3.260 0.001115 ** 
pgr5             2007.15953  271.10096   7.404 1.33e-13 ***
bol1                1.32904    0.55778   2.383 0.017186 *  
bol2                0.62678    0.25461   2.462 0.013827 *  
bol3                2.06117    0.39444   5.226 1.74e-07 ***
bol4                0.65562    0.19024   3.446 0.000568 ***
bol5                0.90784    0.22054   4.117 3.85e-05 ***
bol6                1.70460    0.44100   3.865 0.000111 ***
bol7                1.44768    0.34603   4.184 2.87e-05 ***
bol8                2.05180    0.47155   4.351 1.35e-05 ***
bol9                1.66168    0.37609   4.418 9.95e-06 ***
bol10               0.24621    0.32205   0.765 0.444558    
bol11               1.18526    0.31184   3.801 0.000144 ***
bol12               1.04271    0.33339   3.128 0.001763 ** 
bol13               1.11829    0.21086   5.304 1.14e-07 ***
bol14                    NA         NA      NA       NA    
bol15                    NA         NA      NA       NA    
bol16               1.53937    0.23164   6.646 3.03e-11 ***
bol17               0.30734    0.41578   0.739 0.459788    
bol18               0.79970    0.72009   1.111 0.266756    
bol19                    NA         NA      NA       NA    
bol20               0.52244    0.24466   2.135 0.032734 *  
bol21               0.68933    0.19983   3.450 0.000562 ***
bol22               0.86630    0.47144   1.838 0.066126 .  
bol23               1.03192    0.41578   2.482 0.013069 *  
bol24               0.46684    0.31184   1.497 0.134374    
bol25               1.52552    0.29415   5.186 2.15e-07 ***
bol26               1.07255    0.55778   1.923 0.054496 .  
bol27               0.36800    0.12247   3.005 0.002659 ** 
bol28               0.98527    0.16388   6.012 1.84e-09 ***
bol29               0.01578    0.15134   0.104 0.916950    
bol30                    NA         NA      NA       NA    
bol31                    NA         NA      NA       NA    
bol32               0.82396    0.23573   3.495 0.000473 ***
bol33               1.80530    0.47144   3.829 0.000129 ***
bol34               0.94698    0.72018   1.315 0.188535    
bol35               0.90857    0.26010   3.493 0.000478 ***
bol36                    NA         NA      NA       NA    
bol37                    NA         NA      NA       NA    
bol38               1.48932    0.24961   5.967 2.43e-09 ***
bol39               1.49692    0.32211   4.647 3.37e-06 ***
bol40               1.34811    0.37606   3.585 0.000337 ***
bol41               1.32413    0.88193   1.501 0.133254    
bol42               1.00802    0.37615   2.680 0.007367 ** 
bol43               0.31952    0.62365   0.512 0.608413    
bol44               1.50329    0.50919   2.952 0.003155 ** 
bol45               1.04378    0.37608   2.775 0.005513 ** 
bol46               0.48857    0.32206   1.517 0.129261    
bol47               1.06387    0.27220   3.908 9.29e-05 ***
bol48               0.22076    0.47143   0.468 0.639592    
bol49               1.31310    0.62361   2.106 0.035238 *  
bol50               1.13453    0.19042   5.958 2.56e-09 ***
bol51               0.48723    0.39442   1.235 0.216715    
bol52               1.05559    0.47142   2.239 0.025146 *  
bol53               0.81131    1.24737   0.650 0.515422    
bol54                    NA         NA      NA       NA    
bol55               1.47880    0.88203   1.677 0.093627 .  
bol56               1.45393    0.32208   4.514 6.36e-06 ***
bol57               0.36900    0.36006   1.025 0.305442    
bol58               0.73197    0.27893   2.624 0.008687 ** 
bol59               2.17486    1.24746   1.743 0.081260 .  
bol60               0.44524    0.27223   1.636 0.101935    
bol61               0.72053    0.41578   1.733 0.083105 .  
bol62               0.85919    0.33335   2.577 0.009955 ** 
bol63               2.10554    1.24737   1.688 0.091416 .  
bol64              -1.05321    0.47142  -2.234 0.025478 *  
bol65               1.08143    0.50928   2.123 0.033719 *  
bol66               0.95197    0.41578   2.290 0.022045 *  
bol67               2.35411    0.62362   3.775 0.000160 ***
bol68               1.37188    0.50921   2.694 0.007057 ** 
bol69               0.82143    0.33339   2.464 0.013745 *  
bol70               0.84421    0.55780   1.513 0.130165    
bol71               0.27861    0.36007   0.774 0.439082    
bol72                    NA         NA      NA       NA    
bol73               0.43396    0.23166   1.873 0.061033 .  
bol74               0.76157    0.39441   1.931 0.053493 .  
bol75               0.37087    0.28616   1.296 0.194960    
bol76               1.78414    0.50929   3.503 0.000460 ***
bol77               1.35791    0.28625   4.744 2.10e-06 ***
bol78                    NA         NA      NA       NA    
bol79              -0.59822    0.88191  -0.678 0.497567    
bol80               0.83062    0.19977   4.158 3.21e-05 ***
bol81               0.94343    0.21088   4.474 7.69e-06 ***
bol82               1.04001    0.41576   2.501 0.012369 *  
bol83               0.77240    0.47142   1.638 0.101331    
bol84               0.49286    0.39446   1.249 0.211490    
bol85               0.78872    0.22774   3.463 0.000534 ***
bol86               1.46308    0.39443   3.709 0.000208 ***
bol87               0.87393    0.37609   2.324 0.020142 *  
bol88               0.37407    0.19986   1.872 0.061263 .  
bol89               1.44208    0.44105   3.270 0.001077 ** 
bol90               1.30525    0.29405   4.439 9.05e-06 ***
bol91                    NA         NA      NA       NA    
bol92               0.77453    0.55779   1.389 0.164962    
bol93               1.04926    0.39442   2.660 0.007808 ** 
bol94               0.79883    0.32208   2.480 0.013132 *  
bol95               1.14316    0.19724   5.796 6.81e-09 ***
bol96               0.23897    0.24463   0.977 0.328632    
bol97               1.95068    0.47149   4.137 3.52e-05 ***
bol98               0.94743    0.33336   2.842 0.004483 ** 
bol99               0.60690    0.55780   1.088 0.276584    
bol100             -0.05007    0.39443  -0.127 0.898992    
bol101                   NA         NA      NA       NA    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.247 on 210913 degrees of freedom
  (14674 observations deleted due to missingness)
Multiple R-squared:  0.124,	Adjusted R-squared:  0.1235 
F-statistic: 246.8 on 121 and 210913 DF,  p-value: < 2.2e-16
```

Figure 8.7 Summary of linear regression model

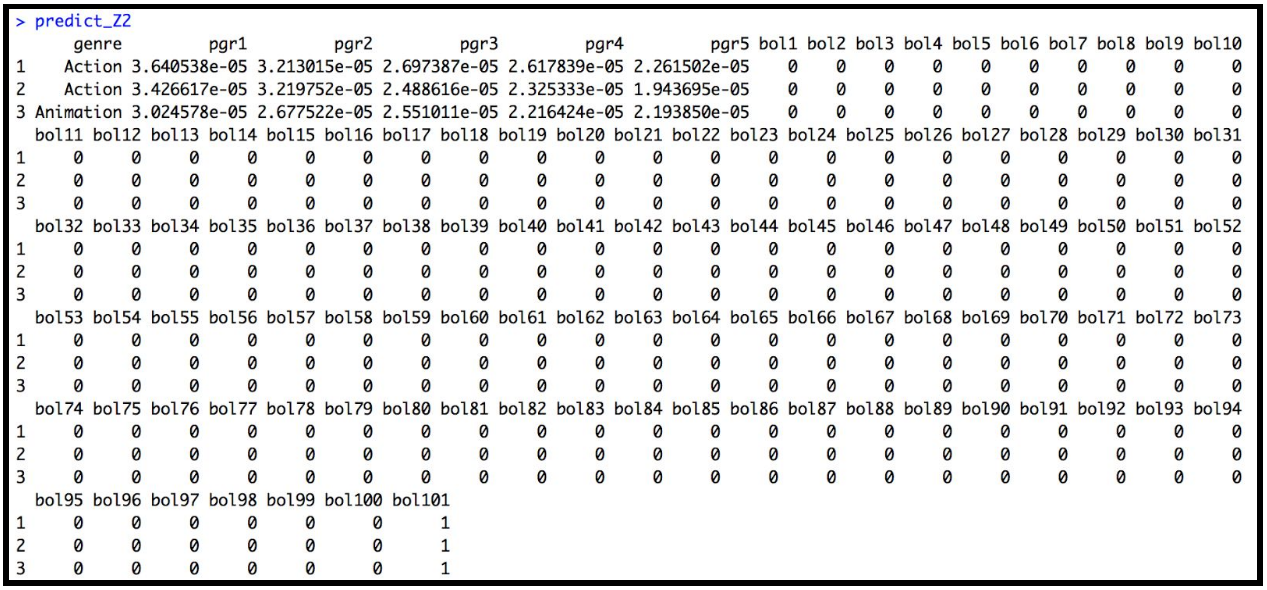
In order to predict, we need to have the same information of the three movies. The data in Table 8.1 is found from each movie's IMDb page.

<table>
<tr><td colspan="2">Batman v Superman: Dawn of Justice (2016)</td></tr>
<tr><td>rating</td><td>6.7</td></tr>
<tr><td>director</td><td>Zack Snyder</td></tr>
<tr><td>Cast</td><td>Ben Affleck, Henry Cavill, Amy Adams, Jesse Eisenberg, Diane Lane, Laurence Fishburne, Jeremy Irons, Holly Hunter,
Gal Gadot, Scoot McNairy, Callan Mulvey</td></tr>
<tr><td colspan="2">Mission: Impossible - Rogue Nation (2015)</td></tr>
<tr><td>rating</td><td>7.4</td></tr>
<tr><td>director</td><td>Christopher Mcquarrie</td></tr>
<tr><td>Cast</td><td>Tom Cruise, Rebecca Ferguson, Jeremy Renner
Simon Pegg, Ving Rhames, Sean Harris, Simon Mcburney, Jingchu Zhang, Tom Hollander</td></tr>
</table>

Minions (2015)	
rating	6.4
director	Kyle Balda, Pierre Coffin
cast	Sandra Bullock, Jon Hamm, Michael Keaton, Allison Janney, Steve Coogan, Jennifer Saunders, Geoffrey Rush, Steve Carell, Pierre Coffin

Table 8.1 Information of the three movies from IMDb page

From the information in Table 8.1, features of the three movies were created as shown in Figure 8.8.



```
> predict_Z2
      genre         pgr1         pgr2         pgr3         pgr4         pgr5 bol1 bol2 bol3 bol4 bol5 bol6 bol7 bol8 bol9 bol10
1    Action 3.640538e-05 3.213015e-05 2.697387e-05 2.617839e-05 2.261502e-05    0    0    0    0    0    0    0    0    0     0
2    Action 3.426617e-05 3.219752e-05 2.488616e-05 2.325333e-05 1.943695e-05    0    0    0    0    0    0    0    0    0     0
3 Animation 3.024578e-05 2.677522e-05 2.551011e-05 2.216424e-05 2.193850e-05    0    0    0    0    0    0    0    0    0     0
  bol11 bol12 bol13 bol14 bol15 bol16 bol17 bol18 bol19 bol20 bol21 bol22 bol23 bol24 bol25 bol26 bol27 bol28 bol29 bol30 bol31
1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
3     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
  bol32 bol33 bol34 bol35 bol36 bol37 bol38 bol39 bol40 bol41 bol42 bol43 bol44 bol45 bol46 bol47 bol48 bol49 bol50 bol51 bol52
1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
3     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
  bol53 bol54 bol55 bol56 bol57 bol58 bol59 bol60 bol61 bol62 bol63 bol64 bol65 bol66 bol67 bol68 bol69 bol70 bol71 bol72 bol73
1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
3     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
  bol74 bol75 bol76 bol77 bol78 bol79 bol80 bol81 bol82 bol83 bol84 bol85 bol86 bol87 bol88 bol89 bol90 bol91 bol92 bol93 bol94
1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
3     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
  bol95 bol96 bol97 bol98 bol99 bol100 bol101
1     0     0     0     0     0      0      1
2     0     0     0     0     0      0      1
3     0     0     0     0     0      0      1
```

Figure 8.8 Features of the three movies in the same format as the features of other training set movies

The result predicted is thus shown below in Table 8.2.

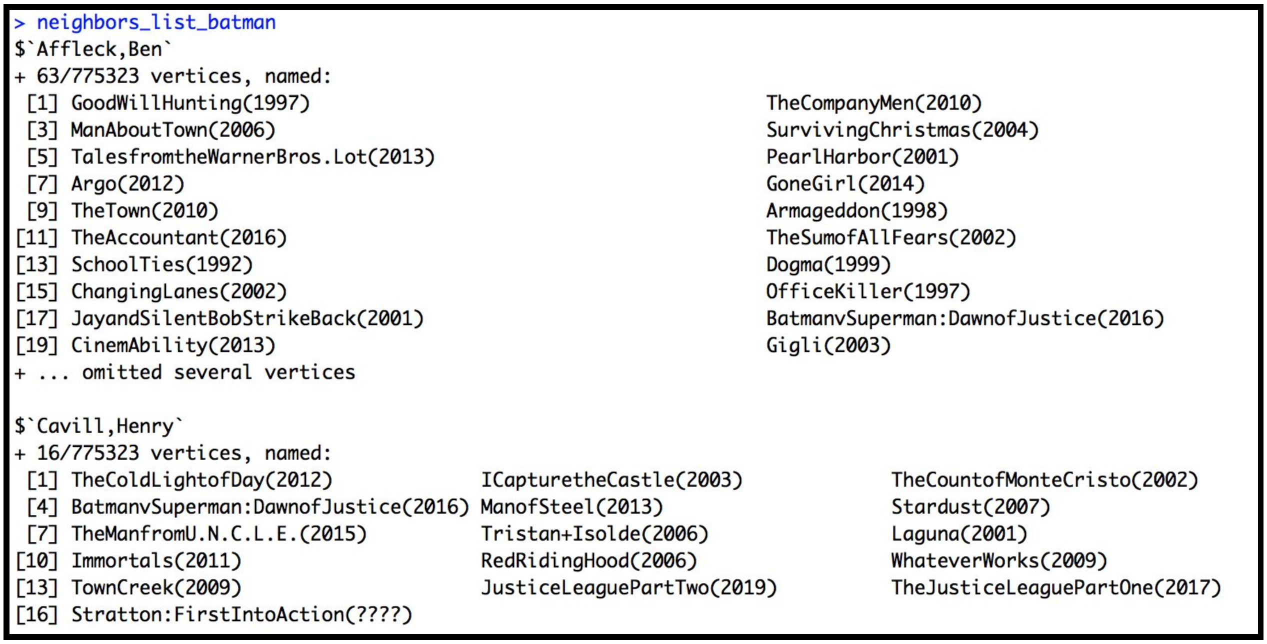
Batman v Superman: Dawn of Justice (2016)	5.66
Mission: Impossible - Rogue Nation (2015)	5.64
Minions (2015)	6.32

Table 8.2 Predicted rating of the three movies

The results are not as accurate as expected but the reasons can be assumed that the features are time-dependent values. Movies in IMdb 250 are continuously changing and thus the popularity of actors. Moreover, the rating of movies are not directly proportional to its popularity. Some movies are seen by only few people but still have higher rating. Some movies with new genre, though the movie is well-known, still have lower ratings than the movies in adult genre. However, the summary of linear regression was partially verified. For example, the slope of news genre are lower than slope of animation genre. The average rating of animation movies were much higher than the average rating of news genre. The linear regression performed properly from the data given. However, in order to predict rating of the three movies, there are much more information and even enormous information may not be enough to explain the complexity behind how the users rate movies.

X. Problem 9

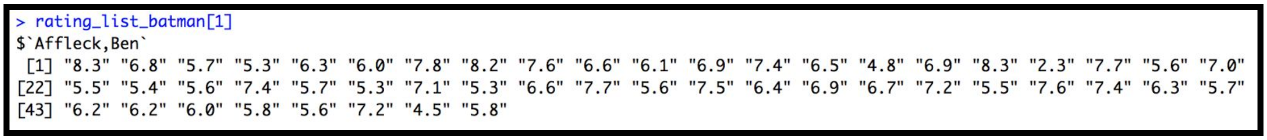
First the bipartite graph of actors network and movie network was constructed with python code. The graph is undirected and unweighted graph. The vertices of one part represent actors/actresses. The other part of vertices represent movies. The actor and actress vertices are connected to all the movies each actor/actress has played in. Thus, using neighbors function, it is possible to get all the movies each actor and actress has played in. For example, neighbors list of batman is shown in Figure 9.1. It shows partial neighbor vertices of two actors in Batman movie.



```
> neighbors_list_batman
$`Affleck,Ben`
+ 63/775323 vertices, named:
 [1] GoodWillHunting(1997)                              TheCompanyMen(2010)
 [3] ManAboutTown(2006)                                 SurvivingChristmas(2004)
 [5] TalesfromtheWarnerBros.Lot(2013)                   PearlHarbor(2001)
 [7] Argo(2012)                                         GoneGirl(2014)
 [9] TheTown(2010)                                      Armageddon(1998)
[11] TheAccountant(2016)                                TheSumofAllFears(2002)
[13] SchoolTies(1992)                                   Dogma(1999)
[15] ChangingLanes(2002)                                OfficeKiller(1997)
[17] JayandSilentBobStrikeBack(2001)                    BatmanvSuperman:DawnofJustice(2016)
[19] CinemAbility(2013)                                 Gigli(2003)
+ ... omitted several vertices

$`Cavill,Henry`
+ 16/775323 vertices, named:
 [1] TheColdLightofDay(2012)           ICapturetheCastle(2003)          TheCountofMonteCristo(2002)
 [4] BatmanvSuperman:DawnofJustice(2016) ManofSteel(2013)               Stardust(2007)
 [7] TheManfromU.N.C.L.E.(2015)        Tristan+Isolde(2006)             Laguna(2001)
[10] Immortals(2011)                   RedRidingHood(2006)              WhateverWorks(2009)
[13] TownCreek(2009)                   JusticeLeaguePartTwo(2019)       TheJusticeLeaguePartOne(2017)
[16] Stratton:FirstIntoAction(????)
```

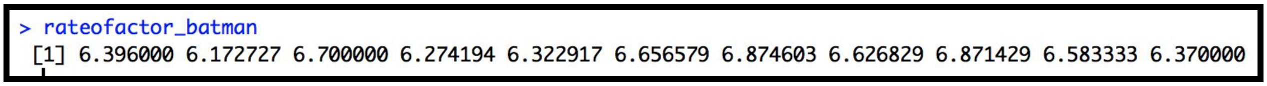
Figure 9.1 Part of neighbors of actor and actress in Batman v Superman: Dawn of Justice (2016)

Then, the new list is created with the rate of the movies that each actor or actress has played in. The score tagged to each actor and actress was calculated by taking the average of the rating of movies he or she has played in as shown in Figure 9.2. Finally, the ratings of the new movies, “Batman v Superman: Dawn of Justice (2016)”, “Mission: Impossible - Rogue Nation (2015)” and “Minions (2015)”, are predicted by taking average of the scores tag to the actor and actress as shown in Figure 9.3.



```
> rating_list_batman[1]
$`Affleck,Ben`
 [1] "8.3" "6.8" "5.7" "5.3" "6.3" "6.0" "7.8" "8.2" "7.6" "6.6" "6.1" "6.9" "7.4" "6.5" "4.8" "6.9" "8.3" "2.3" "7.7" "5.6" "7.0"
[22] "5.5" "5.4" "5.6" "7.4" "5.7" "5.3" "7.1" "5.3" "6.6" "7.7" "5.6" "7.5" "6.4" "6.9" "6.7" "7.2" "5.5" "7.6" "7.4" "6.3" "5.7"
[43] "6.2" "6.2" "6.0" "5.8" "5.6" "7.2" "4.5" "5.8"
```

Figure 9.2 Rating of movies the actor played in



```
> rateofactor_batman
 [1] 6.396000 6.172727 6.700000 6.274194 6.322917 6.656579 6.874603 6.626829 6.871429 6.583333 6.370000
```

Figure 9.3 Score of actors who played in Batman v Superman: Dawn of Justice (2016)

Therefore, the values in Table 9.1 are the predicted ratings of the three new movies using bipartite graph.

Batman v Superman: Dawn of Justice (2016)	6.531692
Mission: Impossible - Rogue Nation (2015)	6.327938
Minions (2015)	6.327938

Table 9.1 Predicted rating of new movies

XI. Conclusion

The data did not reveal any strong correlations with the movie rating, but there were some interesting ones. In general, it was found that a movie's rating generally *decreases* as the pagerank of its actors increases. This might highlight a peculiarity in measuring the importance of an actor to the movie by the actor's pagerank. This was already a concern highlighted in part 3. Pagerank did not accurately identify main characters in movies which might be more predictive of the movie's rating. Instead pagerank tended to value the side characters more which seemed to play in many movies and not have important roles. It's possible that actors with higher pagerank are cheaper actors that appear in many films, and their negative correlation to movie rating might indicate that the movie budget was lower and cheaper actors were used as a cost-savings measure.

There were also some movies which were given high weight in their connection to each other despite not sharing any main characters. It was found that their high weight was due only to a large share of background actors that happened to be the same. This likely throws off any algorithms that relies on using the Jaccard index or a related index to weight the connection between two movies.

Future work might involve working to find features that can identify main characters in movies, and better systems of calculating weights between different nodes to weight main characters differently from side characters and background characters.

The technique in part 9 performs better than the technique in number 8. In part 9, all the movies were correctly identified as being higher rated movies than the average movie, whereas in part 8, the algorithm has trouble identifying that the first two movies outperform the average.