**Assignment Seven**

Jacob Berlin

CS432 – Spring 2016

1. Find 3 users who are closest to you in terms of age, gender, and occupation. For each of those 3 users:

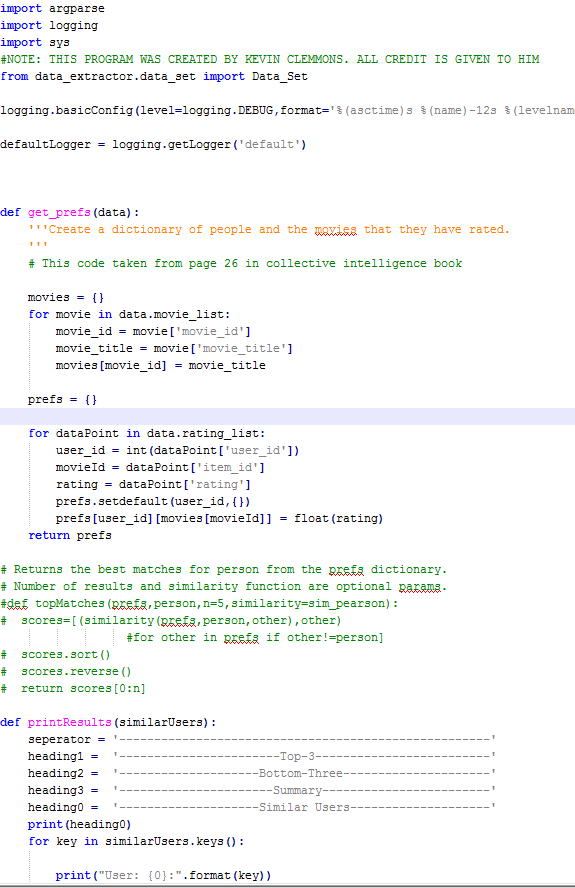
- what are their top 3 favorite films?

- bottom 3 least favorite films?

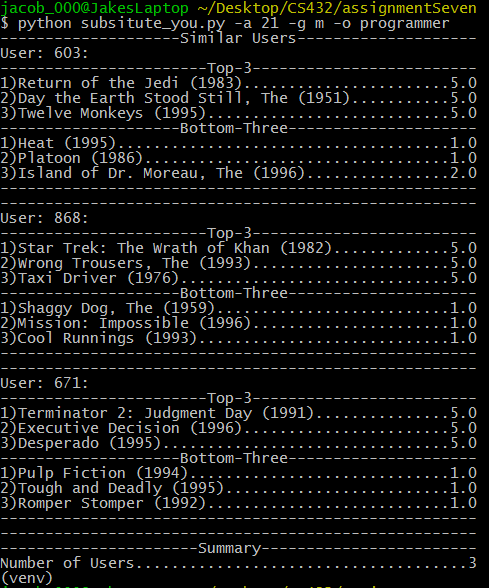
Based on the movie values in those 6 tables (3 users X (favorite + least)), choose a user that you feel is most like you. Feel free to note any outliers (e.g., "I mostly identify with user 123, except I did not like “Ghost'' at ll"). This user is the "substitute you".

For questions 1, 2, 3, and 4 I used and referenced ‘recommendations.py’.

To start this assignment I downloaded the grouping of the 100k movie reviews off of <http://grouplens.org/datasets/movielens/100k/> and sorted all of the data into their respective folder under ‘./Data/MovieLens’.



This problem required us to take the files ‘u.user’, ‘u.data’, and ‘u.item’ which housed the data for 943 different users, 1700 different movies, and 100,000 individual ratings to link together three users most like ourselves based on the ratings that each had given six different movies, three of their favorite and three of their least. Given the preliminary information that I am a 21 year old male programmer who enjoys mainly science fiction or horror movies, my search was narrowed immensely. After modifying a program created by my classmate Kevin Clemmons named ‘substitute\_you.py’ (partially shown above) which took in the parameters of the user’s age, gender, and occupation to generate the users who were most like myself, I came up with the data needed.



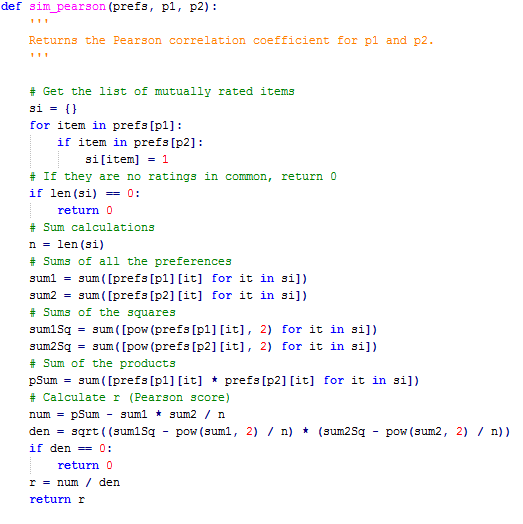
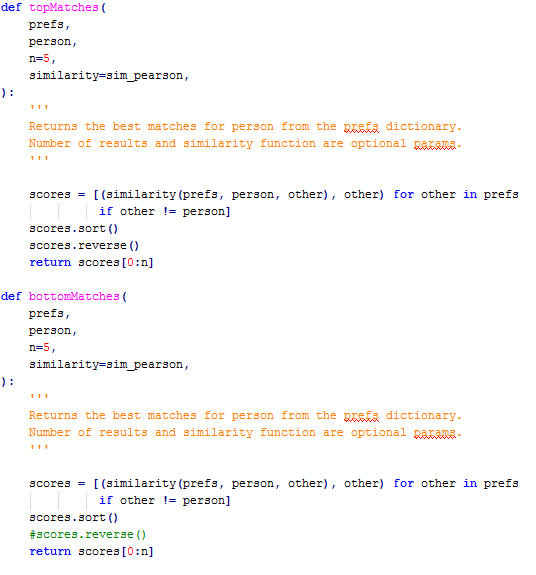
(5.0 for top rated, 1.0 for bottom rated)

User 603:   
Top: Return of the Jedi (1983), The Day the Earth Stood Still (1951), Twelve Monkeys (1995)  
Bottom: Heat (1995), Platoon (1986), The Island of Dr. Moreau (1996)  
User 868:  
Top: Star Trek: The Wrath of Khan (1982), The Wrong Trousers (1993), Taxi Driver (1976)  
Bottom: The Shaggy Dog (1959), Mission: Impossible (1996), Cool Runnings (1993)  
User 671:  
Top: Terminator 2: Judgement Day (1991), Executive Decision (1996), Desperado (1995)  
Bottom: Pulp Fiction (1994), Tough and Deadly (1995), Rough Stomper (1992)

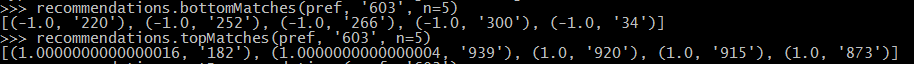
From the program I received a choice of three different people from the ‘u.users’ file who were all 21 year old male programmers. Now, specifically because User 671 rated ‘Pulp Fiction (1944)’ as their least favorite movie of all time, I immediately crossed him off of my list of potential substitutes. User 868 has a strong lineup of favorite movies with ‘Star Trek: The Wrath of Khan (1982)’ being the most notable, however User 603’s top rated movie, ‘Return of the Jedi (1983),’ is my favorite movie of all time…thus making this an easy choice. Although ‘Twelve Monkeys (1995)’ would not be close to one of my top movies, I generally agree with most of 603’s choices, therefore User 603 will be my ‘substitute me.’

2. Which 5 users are most correlated to the substitute you? Which 5 users are least correlated (i.e., negative correlation)?

To calculate which users are most and least correlated to my substitute me I used the sim\_pearson correlation score calculator in ‘recommendations.py’.



With list of the users and the items each user has rated the sim\_pearson is able to calculate a correlation score based off of who rated movies similarly to my user. The ‘topMatches’ and ‘bottomMatches’ functions were called to actually get the list of the greatest and least correlated users. A score of 1.0 signifies that the correlated user is extremely similar to substitute me, a score of -1.0 means that the user rates movies exactly the opposite of how substitute me would.



Most correlation to 603: Value: u.user information:

User 182 1.0 182|36|M|programmer|33884

User 939 1.0 939|26|F|student|33319

User 920 1.0 920|30|F|artist|90008

User 915 1.0 915|50|M|entertainment|60614

User 873 1.0 873|48|F|administrator|33763

Least correlation to 603: Value: u.user information:

User 220 -1.0 220|30|M|librarian|78205

User 252 -1.0 252|42|M|engineer|07733

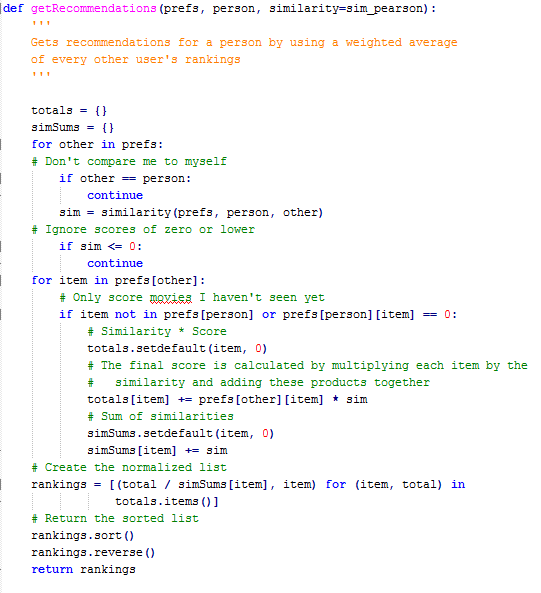
User 266 -1.0 266|62|F|administrator|78756

User 300 -1.0 300|26|F|programmer|55106

User 34 -1.0 34|38|F|administrator|42141

3. Compute ratings for all the films that the substitute you hasn't seen. Provide a list of the top 5 recommendations for films that the substitute you should see. Provide a list of the bottom 5 recommendations (i.e., films the substitute you is almost certain to hate).

In the file ‘recommendations.py’ there is a function called ‘getRecommendations’ that will take the list of rankings for the movies that substitute me has made and generate ratings based off of those statistics on movies that substitute me has not seen depending on what other users like me would rate them (all based off of the correlation score).



I ran the function and received the following output.



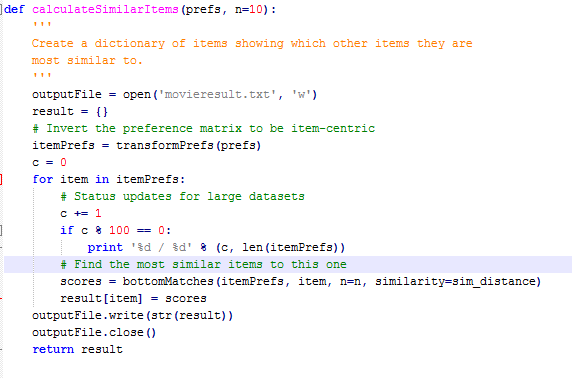
(5.0 rating for movies that substitute me should see, 1.0 for movies that substitute me would hate)  
Movies that substitute me should see: (all 5.0 ratings)  
1. The Wrong Trousers (1993)   
2. When We Were Kings (1996)  
3. Toy Story (1995)  
4. Tombstone (1993)  
5. A Time to Kill (1996)

Movies that substitute me would hate: (all 1.0 ratings)  
1. Air Bud (1997)  
2. Anaconda (1997)  
3. Bean (1997)  
4. Beauty and the Beast (1991)  
5. Boogie Nights (1997)

4. Choose your (the real you, not the substitute you) favorite and least favorite film from the data. For each film, generate a list of the top 5 most correlated and bottom 5 least correlated films. Based on your knowledge of the resulting films, do you agree with the results? In other words, do you personally like / dislike the resulting films?

To start this problem I picked out my personal favorite movie from the list, ‘Return of the Jedi (1983)’ and my least favorite movie, ‘Batman & Robin (1997)’. I don’t think that I would need to explain myself for either of them…especially the latter.

In ‘recommendations.py’ there is a function called ‘calculateSimilarItems’ that takes the entire MovieLens list and gives the n amount of most correlated and n amount of least correlated films for each movie. I ran the function twice to get the most correlated first and the least correlated after. All of the compilation data is stored in ‘movieresult.txt’.



First compilation:



Second compilation:



Finally, with all of the data in the text file, I then took out all of the information pertaining to both of my chosen movies and came out with these results: (1.0 for most correlated, 0 for least correlated)

'Return of the Jedi (1983)': Most correlated

[(1.0, 'klaka (Cold Fever) (1994)'),

(1.0, "Wooden Man's Bride, The (Wu Kui) (1994)"),

(1.0, 'Witness (1985)'),

(1.0, 'Wings of Courage (1995)'),

(1.0, 'Wedding Gift, The (1994)')],

'Return of the Jedi (1983)': Least correlated

[(0, 'All Things Fair (1996)'),

(0, 'August (1996)'),

(0, 'B. Monkey (1998)'),

(0, 'Big Bang Theory, The (1994)'),

(0, 'Bird of Prey (1996)')]

'Batman & Robin (1997)': Most correlated

[(1.0, 'Van, The (1996)'),

(1.0, 'Vampire in Brooklyn (1995)'),

(1.0, 'Two if by Sea (1996)'),

(1.0, 'Twin Town (1997)'),

(1.0, 'Turning, The (1992)')],

'Batman & Robin (1997)': Least correlated

[(0, '3 Ninjas: High Noon At Mega Mountain (1998)'),

(0, '8 Seconds (1994)'),

(0, 'Afterglow (1997)'),

(0, 'All Things Fair (1996)'),

(0, 'American Buffalo (1996)')],

Based on the results that I received for ‘Return of the Jedi (1983)’ and ‘Batman & Robin (1997)’, I do not agree with the results on the sole grounds that I do not think that I have heard of any single one of the movies listed under either the most correlated or least correlated movies. I think it has something to do with my age being only 21.

Recommendations.py used and referenced from:

“https://github.com/Jberl002/Programming-Collective-Intelligence/blob/master/chapter2/recommendations.py”