CS 432: Homework Number Six

Due on March 31, 2019 at 4:20 PM

Alexander Nwala

Tim Bruce

Problem 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 all").
This user is the "substitute you".

Solution
At surface level, this problem seems very simple. As you work on it, development slows. It is not something you can do with recommendations.py which was provided. I was able to make the solution a lot simpler by making a class object which stores the user data for easier access.

Listing 1: User Class for Easy Data Access.

```
class User:
    """
    This class holds user data for organized, easy access.
    """
    user_id = None
    age = None
    gender = None
    occupation = ""
    zip_code = None
    ageGenderOccupationValue = None
    inputString = ""
    reviews = []

    def __init__(self, uid, age, gender, occupation, zip_code, input_string):
        self.user_id = int(uid)
        self.age = int(age)
        self.gender = gender
        self.occupation = occupation
        self.zip_code = zip_code
        self.inputString = input_string
```

Once the class is mostly populated, it is necessary to determine which three are closest to me in order to answer the question. To do this, I manually created a User object of myself, and assigned every user an aggregate score based on their age, gender, and occupation. I have implemented this by normalizing all of the values and then finding an average of the scores. This is a bad implementation, I should have set up a multi-dimensional vector and found the closest vector to me, and this may lead to bad results down the line. Once I have this aggregate score, and top three matches, the program does some basic data manipulation to get the movies from the text file and displayed them nicely here.

Image 1: The Top Three Matches to Me and Their Top Movies.

```
575|33|M|marketing|46032
User's top three movies:
Schindler's List (1993)
Rear Window (1954)
Monty Python and the Holy Grail (1974)

User's bottom three movies:
```

```
Liar Liar (1997)
Truth About Cats & Dogs, The (1996)
Some Folks Call It a Sling Blade (1993)



4|24|M|technician|43537
User's top three movies:
Blues Brothers 2000 (1998)
Wonderland (1997)
In & Out (1997)

User's bottom three movies:
Spawn (1997)
Mimic (1997)
Client, The (1994)



69|24|M|engineer|55337
User's top three movies:
Graduate, The (1967)
Scream (1996)
Empire Strikes Back, The (1980)

User's bottom three movies:
Devil's Own, The (1997)
Peacemaker, The (1997)
Saint, The (1997)
```

I ended up choosing user with ID 4 because I love the Blues Brothers, and I hate horror movies. I don't like Scream from user 69 and while I do adore the Holy Grail, and like Schindler's List, I have seen and enjoyed Liar Liar.

Problem 2

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

Solution
This problem was a lot of fun to answer. The problem itself is easy thanks to the provided recommendations.py. Just call sim_distance with the appropriate variables. I was able to solve it in about 10 minutes. However, I noticed that execution took a longish time, and maxed out a single core of my processor. So I went back and took some time to multithread it. The submitted version splits on the calculation for each other user, and uses the maximum number of cores on your machine. To solve the problem, just format the output nicely. I'm proud of this code, so I'm posting it here.

Listing 2: Multithreaded Code for Getting Similar Users to Your Substitute User.

```
def get_most_and_least_similar_users():
    """
    Calculates users that are similar to the substitute user from problem one. Early on, I noticed that the program
    took a while to execute, so I multi-threaded the sim_distance function to expedite execution.

    :return:
    Prints out data on the top five most similar and least similar users to the substitute user.
    """
    pool = mp.Pool(mp.cpu_count())
    prefs = r.loadMovieLens(os.getcwd()+"/ml-100k")
    similarity_data = [pool.apply(get_similarity_data, args=(prefs, other_id+1)) for other_id in range(943)]
    pool.close()
    similarity_data = sorted(similarity_data, key=lambda x: x[1])

    print("The most similar users to " + get_user(sub_id) + " (which is the substitute) are:")
    for i in range(5):
        u = get_user(similarity_data[len(similarity_data)-(i+1)][0])
        print(u)

    print("\nThe least similar users are: ")
    for i in range(5):
        u = get_user(similarity_data[i][0]).split("\n")[0]
        print(u)
```

Finally, there is the program's output, which is shown below.

```
tim@Wintertide-Daedalus:~/Documents/GitKraken/anwala.github.io/cs532-s19/assignments/A6$ ./A6.py
The most similar users to 4|24|M|technician|43537 (which is the substitute) are:
911|37|F|writer|53210
814|30|M|other|12345
794|32|M|educator|57197
793|22|M|student|85281
776|30|M|librarian|51157

The least similar users are:
67|17|M|student|60402
93|48|M|executive|23112
172|55|M|marketing|22207
196|49|M|writer|55105
208|43|M|engineer|01720
```

The similar users seem to skew younger and male, which is accurate to me, but they also tend to be in less technical positions. The least similar users skew older (for this data set) and tend to be in the more technical positions.

Problem 3

Compute ratings for all the films that the substitute you have not 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).

Solution This problem has a simple solution thanks to recommendations.py. The getRecommendations function takes the user ID as input, and outputs sorted potential matches. Beyond that there is just some text formatting to nicely output a solution to the problem.

```
The most recommended movies for 4|24|M|technician|43537 (which is the substitute) are:
They Made Me a Criminal (1939)
Star Kid (1997)
Someone Else's America (1995)
Sliding Doors (1998)
Saint of Fort Washington, The (1993)

The least recommended movies for the substitute are:
3 Ninjas: High Noon At Mega Mountain (1998)
Amityville 1992: It's About Time (1992)
Amityville 3-D (1983)
Amityville Curse, The (1990)
Amityville: A New Generation (1993)
```

These recommendations appear to be mostly drama films, with different twists such as Sci-fi, Comedy/Romance, and History, which fall into the categories of things I like. The least recommended movies are mostly from a single horror series which is a genre that I lothe, so it got that right too. Fun fact, Someone Else's America was released in Europe exactly nine days before I was born.

Problem 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?

Solution
This solution is also fairly easy using the calculateSimilarItems function, which does all of the heavy lifting of the solution for you, outputting all of the solutions for all of the films. After that is just basic data output stuff that I've done four times already. I attempted to speed up execution using multithreading, but all attempts produced results at least five times slower than single threading, if not four times more than that.

Because this answer seems really short, allow me to talk about my favorite movie on this list: The Blues Brothers. It's a great movie combining two of my favorite genres: Comedy and Musicals. Monty Python's The Holy Grail is a close second, but the Blues Brothers really knocks it out of the park. My least favorite movie of all time is Citizen Kane. I understand that it's a masterpiece of storytelling, but the rosebud twist at the end feels really meaningless to me, and it's just a bunch of people talking about a rich dude and his problems for the rest of the movie.
Here is the output from the program.

```
Five similar movies to my favorite movie Blues Brothers, The (1980) are:
 kldum klaka (Cold Fever) (1994)
Wonderland (1997)
Witness (1985)
```

Problem 4 continued on next page...

```
Wings of Courage (1995)
Wild Reeds (1994)

Five similar movies to my least favorite movie Citizen Kane (1941) are:
Two or Three Things I Know About Her (1966)
The Innocent (1994)
Spirits of the Dead (Tre passi nel delirio) (1968)
Spanish Prisoner, The (1997)
Someone Else's America (1995)
```

I cannot find the first movie, but I would hate wonderland, Witness, and probably Wild Reeds, so that recommendation probably doesn't work, and as I menitioned before, I would probably like Someone Else's America, so this recommendation system doesn't seem to work for me.