CS 535: Assignment #6

Due on 11:59pm March 21, 2018

Alexander C. Nwala 16:20

David Sinclair

March 16, 2018

Contents

1 Problem 1

The goal of this project is to use the basic recommendation principles we have learned for user-collected data. You will modify the code given to you which performs movie recommendations from the MovieLense data sets.

The MovieLense data sets were collected by the GroupLens Research Project at the University of Minnesota during the seven-month period from September 19th, 1997 through April 22nd, 1998. We are using the "100k dataset"; available for download from:
http://grouplens.org/datasets/movielens/100k/

There are three files which we will use:

1. u.data: 100,000 ratings by 943 users on 1,682 movies. Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered. This is a tab separated list of

user id — item id — rating — timestamp

The time stamps are unix seconds since 1/1/1970 UTC.

Example:

196 242 3 881250949
186 302 3 891717742
22 377 1 878887116
244 51 2 880606923
166 346 1 886397596
298 474 4 884182806
115 265 2 881171488

2. u.item: Information about the 1,682 movies. This is a tab separated list of

movie id — movie title — release date — video release date — IMDb URL — unknown — Action — Adventure — Animation —Children's — Comedy — Crime — Documentary — Drama — Fantasy — Film-Noir — Horror — Musical — Mystery — Romance — Sci-Fi — Thriller — War — Western —

The last 19 fields are the genres, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once. The movie ids are the ones used in the u.data data set.

Example:

161—Top Gun (1986)—01Jan1986——http://us.imdb.com/M/titleexact?Top%20Gun%20(1986)—0—1—0—0—0—0—0—0—0—
162—On Golden Pond (1981)—01Jan1981——http://us.imdb.com/M/title-exact?On%20Golden%20Pond%20(1981)—0—0—0—
163—Return of the Pink Panther, The (1974)—01Jan1974——http://us.imdb.com/M/titleexact?Return%20of%20the%20Pink%
0—0—0—0—0

3. u.user: Demographic information about the users. This is a tab separated list of:

user id — age — gender — occupation — zip code

The user ids are the ones used in the u.data data set.

Example:

1—24—M—technician—85711
2—53—F—other—94043
3—23—M—writer—32067
4—24—M—technician—43537

5—33—F—other—15213

The code for reading from the u.data and u.item files and creating recommendations is described in the book Programming Collective Intelligence. Feel free to modify the PCI code to answer the following questions.
(Questions 10 Points)

1.1 Question 1

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".

1.2 Answer 1

I created a program called 01_defuse.py. The program compared the user information to my own using euclidean Distance. Below is a sample of the code. I converted the gender to values of 1 for male and 2 for female, and the occupations I listed alphabetically and numbered them also. My numbers were 48,1,1 for age, gender Male = 1, and administrator = 1. This I used has P.

```
def euclideanDistance(P, Q):
   if( len(P) != len(Q) ):
      return -1
   sumOfSquares = 0
   for i in range(0, len(P)):
      sumOfSquares += (P[i] - Q[i]) * (P[i] - Q[i])
   return math.sqrt(sumOfSquares)

P = [48,1,1] #My age + gender + occupation
```

The file it output was called dist_users.txt. Sorting using 02_sorting.py The following numbers appeared. User Id and Euclidean Distance from me.

409 0.0
455 0.0
72 1.0

I then got the movie rating for the above users and created 3 files. Using pthon program 03_movie.py.

```
   for line in csv.reader(tsv, delimiter="\t"):
      rate = int(line[2])
      movie = int(line[1])
      user = line[0]
      if user == '409':
         print(user, movie, rate,file=open('409movie.txt','a'))
      if user == '455':
         print(user, movie, rate,file=open('455movie.txt','a'))
      if user == '72':
         print(user, movie, rate,file=open('072movie.txt','a'))
```

Program 04a_sortmovie.py did file cleanup. Program 05a_moviename.py took the ratings from each user and the movie titles can created a list per user for the rate of 1 and 5. The files can be seen in 072movierank1.tsv, 072movierank5.tsv, 409movierank1.tsv, 409movierank5.tsv, 455movierank1.tsv, 455movierank5.tsv.

Table 1. User 072 Bottom 3 rated films

Rated	Movie Title
1	Twelve Monkeys (1995)
1	Empire Strikes Back, The (1980)
1	Return of the Jedi (1983)

Table 2. User 072 Top 3 rated films

Rated	Movie Title
5	Dead Man Walking (1995)
5	Usual Suspects, The (1995)
5	Mr. Holland's Opus (1995)

Table 3. User 409 Bottom 3 rated films

Rated	Movie Title
1	Evil Dead II (1987)
1	Mimic (1997)
1	Kull the Conqueror (1997)

Table 4. User 409 Top 3 rated films

Rated	Movie Title
5	Postino, Il (1994)
5	Star Wars (1977)
5	Three Colors: Red (1994)

Table 5. User 455 Bottom rated film

Rated	Movie Title
1	Natural Born Killers (1994)

Table 6. User 455 Top 3 rated films

Rated	Movie Title
5	Star Wars (1977)
5	Pulp Fiction (1994)
5	Jurassic Park (1993)

I mostly identify with user 455, except I since he only had 1 movie that he rated has a 1 it was a little difficult. I did not select 409 because the some of the movies he liked I had not seen and some that he ranked has 1 I liked.

2 Problem 2

2.1 Question 2

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

2.2 Answer 2

To get the correlation and the negative correlation I used program 06_correlat.py. The program came mainly from the https://github.com/arthure/ProgrammingCollectiveIntelligence/blob/master/chapter2/recommendations.py program supplied during class.

Table 7. Correlation

Correlation	User id
0.7419985160044517	736
0.749777150492778	531
0.840343067198297	841
0.8660254037844378	341
0.866025403784439	729

Table 8. Negative Correlation

Negative Correlation	User id
-0.004087554618781602	230
-0.004880895105478244	382
-0.006123257613033035	551
-0.006435560762482478	724
-0.007895428709972493	152
-0.008537347209531861	337

3 Problem 3

3.1 Question 3

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).

3.2 Answer 3

To get the recommendations for the films I used program 08_ratings.py. The program came mainly from the https://github.com/arthure/ProgrammingCollectiveIntelligence/blob/master/chapter2/recommendations.py program supplied during class. The file I had it create was called 455recommend.csv which listed all movies not seen be user 455 and assigned the rating number to them.

Table 9. Top 5 Recommendations

Recommendation Rateing	Movie Title
5.000000000000001	Prefontaine (1997)
5.0	They Made Me a Criminal (1939)
5.0	Star Kid (1997)
5.0	Someone Else's America (1995)
5.0	Santa with Muscles (1996)

Table 10. Bottom 5 Recommendations

Recommendation Rateing	Movie Title
1.0	August (1996)
1.0	Amityville: Dollhouse (1996)
1.0	Amityville: A New Generation (1993)
1.0	Amityville 1992: It's About Time (1992)
1.0	3 Ninjas: High Noon At Mega Mountain (1998)

4 Problem 4

4.1 Question 4

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?

4.2 Answer 4

For this program I used 09_recommen.py that was created by Greg Reda in 2013. The code was found on http://www.gregreda.com/2013/10/26/usingpandasonthemovielensdataset/ and was helpful and interesting.
The films that it chose I have not seen. So I am not sure. I may decide to rent a couple and see if it is correct.

Table 11. Top 5 Recommendations

Movie Title	rating size	mean
They Made Me a Criminal (1939)	1	5.0
Marlene Dietrich: Shadow and Light (1996)	1	5.0
Saint of Fort Washington, The (1993)	2	5.0
Someone Else's America (1995)	1	5.0
Star Kid (1997)	3	5.0

Table 12. Bottom 5 Recommendations

Movie Title	rating size	mean
Eye of Vichy, The (Oeil de Vichy, L') (1993)	1	1.0
Butterfly Kiss (1995)	1	1.0
Daens (1992)	1	1.0
JLG/JLG - autoportrait de décembre (1994)	1	1.0
Touki Bouki (Journey of the Hyena) (1973)	1	1.0

5 Problem 5

5.1 Question 5

Extra credit

(3 points)

5. Rank the 1,682 movies according to the 1997/1998 MovieLense data. Now rank the same 1,682 movies according to todays (March 2018) IMDB data (break ties based on # of users, for example: 7.2 with 10,000 raters 7.2 with 9,000 raters).

Draw a graph, where each dot is a film (i.e., 1,682 dots). The xaxis is the MovieLense ranking and the y-axis is today's IMDB ranking.

5.2 Answer 5