Amara Naas (anaas)
CS595 — Fall 2013

Assignment 7

11/14/2013

Problem 1

In my approach to solve this problem I used the data which available at [1] and used the *MovieLens 100k* as specified in assignment 7. The data include three data files which are *u.data*[1], *u.item*[1], and *u.user*[1]. These three files have an information about 100,000 ratings by 943 users on 1,682 movies, the 1,682 movies, and the users respectively. Also the data include the code which I modified to solve this question and other questions and *assignment8.py*[1]. "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"[2].

In order to find the five movies which have the highest average ratings I did the following:

- The code *recommendations.py* already imported the data from *u.item* and saved the movies' *title and Ids* in a dictionary called *movies*.

```
movies={}
for line in open('u.item'):
  (id,title)=line.split('|')[0:2]
  movies[id]=title
```

- And also created another dictionary which contains *user id, movie ids, and the user ratings for each of them* from *u.data* and called *prefs*.

```
prefs={}
for line in open('u.data'):
  (user,movieid,rating)=line.split('\t')[0:3]
  prefs.setdefault(user,{})
  prefs[user][movies[movieid]]=float(rating)
```

- I created another dictionary for all movies and their user rating and called it *rating_hi_avg*.

```
rating_hi_avg = {}
for user in prefs.keys():
     for key,value in prefs[user].iteritems():
            rating_hi_avg.setdefault(key,[])
            rating_hi_avg[key].append(value)
```

- I created another dictionary for all movies and their user average rating and called it *average*.

```
average = {}
for movie in rating_hi_avg.keys():
  average[movie] = mean(rating_hi_avg[movie])
```

[1]File uploaded to github

- I sorted all movies based on their average and reverse to get the highest average rating and print the highest ten movies.

```
sorted_x = sorted(average.iteritems(), key=operator.itemgetter(1))
sorted_x.reverse()
print (" ##### number [1] solution ##### ")
for (key,value) in sorted_x[0:10]:
 print key,' & ',value,' \\\\'
```

Table 1 show that there are ten ties movies that have the highest average ratings.

Movies	Ratings
A Great Day in Harlem (1994)	5.0
Prefontaine (1997)	5.0
Aiqing wansui (1994)	5.0
Star Kid (1997)	5.0
Marlene Dietrich: Shadow and Light (1996)	5.0
Entertaining Angels: The Dorothy Day Story (1996)	5.0
The Saint of Fort Washington (1993)	5.0
Someone Else's America (1995)	5.0
Santa with Muscles (1996)	5.0
They Made Me a Criminal (1939)	5.0

Table 1: The 10 movies which have the highest average ratings

Problem 2

For answering question two I just did the following:

- Created new dictionary called *lengthList* and find the length of *rating_hi_avg* from question one for all rating of each movies.

```
lengthList = {}
for movie in rating_hi_avg.keys():
  lengthList[movie] = len(rating_hi_avg[movie])
```

- I sorted all movies based on the length of their ratings and reverse to get the movies that received the most ratings.

```
sorted_x = sorted(lengthList.iteritems(), key=operator.itemgetter(1))
sorted_x.reverse()
print (" ##### number [2] solution ##### ")
```

```
for (key,value) in sorted_x[0:5]:
 print key,' & ',value,' \\\\'
```

Table 2 show the movies that received the most ratings.

Movies	Number of Ratings
Star Wars (1977)	583
Contact (1997)	509
Fargo (1996)	508
Return of the Jedi (1983)	507
Liar Liar (1997)	485

Table 2: The movies that received the most ratings

Problem 3

For answering question three I just did the following:

- Created new dictionary called *rating_women* and extract all female from *prefs* dictionary and save all movies that are rated by women.

```
rating_women = {}
for user in prefs.keys():
  if gender[user][1] == 'M':
    continue
  for key,value in prefs[user].iteritems():
            rating_women.setdefault(key,[])
            rating_women[key].append(value)
```

- I created another dictionary for all movies and their user average rating by women and called it *average*.

```
average = {}
for movie in rating_women.keys():
  average[movie] = mean(rating_women[movie])
```

- I sorted all movies based on their average and reverse to get the highest average rating and print the highest movies.

```
sorted_x = sorted(average.iteritems(), key=operator.itemgetter(1))
sorted_x.reverse()
print (" ##### number [3] solution ##### ")
```

```
for (key,value) in sorted_x[0:11]:
 print key,' & ',value,' \\\\'
```

Table 3 show the movies ties that were rated the highest on average by women.

Movies	Women's Ratings
The Visitors (Visiteurs, Les) (1993)	5.0
Prefontaine (1997)	5.0
Telling Lies in America (1997)	5.0
Foreign Correspondent (1940)	5.0
Faster Pussycat! Kill! Kill! (1965)	5.0
Year of the Horse (1997)	5.0
Mina Tannenbaum (1994)	5.0
Maya Lin: A Strong Clear Vision (1994)	5.0
Everest (1998)	5.0
Someone Else's America (1995)	5.0
Stripes (1981)	5.0

Table 3: The movies that were rated the highest on average by women

Problem 4

For answering question fore I just change the condition in question three as in the following line:

```
if gender[user][1] == 'F':
```

Table 4 show the movies ties that were rated the highest on average by men.

Problem 5

For answering question five I just invert the preference matrix to be user-centric in *prefs* dictionary instead of movie-centric by using *topMatches* function which give the most and least like match to *Top Gun*.

```
flip_prefs = transformPrefs(prefs)
print (" ##### number [5] solution ##### ")
print('Top match : ',topMatches(flip_prefs,'Top Gun (1986)')[0])
print('Least match : ',topMatches(flip_prefs,'Top Gun
    (1986)')[len(topMatches(flip_prefs,'Top Gun (1986)')) - 1 ])
```

Movies	Men's Ratings
Delta of Venus (1994)	5.0
A Great Day in Harlem (1994)	5.0
The Leading Man (1996)	5.0
Love Serenade (1996)	5.0
Prefontaine (1997)	5.0
Aiqing wansui (1994)	5.0
Little City (1998)	5.0
Star Kid (1997)	5.0
Marlene Dietrich: Shadow and Light (1996)	5.0
Entertaining Angels: The Dorothy Day Story (1996)	5.0
The Quiet Room (1996)	5.0
The Saint of Fort Washington (1993)	5.0
A Letter From Death Row (1998)	5.0
Santa with Muscles (1996)	5.0
They Made Me a Criminal (1939)	5.0

Table 4: The movies that were rated the highest on average by men

Table 5 show the movies that most and least like match to *Top Gun* based on the following Pearson's correlation:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \tag{1}$$

Movies	Pearson's r
Top match: Wild America (1997)	1.0
Least match: The Babysitter (1995)	-1.0

Table 5: The movies that most and least like match to *Top Gun*

Problem 6

In this question I created a dictionary called *userMostRating* and save all users and their ratings length than sorted them.

```
userMostRating = {}
for user in prefs.keys():
   userMostRating.setdefault(user,len(prefs[user]))

sorted_x = sorted(userMostRating.iteritems(),
    key=operator.itemgetter(1))
```

```
sorted_x.reverse()
print (" ##### number [6] solution ##### ")
for (key,value) in sorted_x[0:5]:
 print key,' &  ',value,' \\\\'
```

Table 6 show the five raters rated the most films.

Raters' Id	Number Of Films
405	736
655	678
13	632
450	538
276	516

Table 6: The five raters rated the most films

Problem 7

In this question I found that there is 806 ties agreed and I did the following to get the results:

Compute scores for everyone to everyone by calling *calculateSimilarUser* which call the *topMatches* function in turn and return the top fore similarity scores for everyone.

```
simi_users = calculateSimilarUser(prefs,4)
```

```
def calculateSimilarUser(prefs,n=5):
  # Create a dictionary of users showing which other users they
  # are most similar to.
  result={}
  c=0
  for user in prefs:
    # Status updates for large datasets
    c+=1
    if c%100==0: print "%d / %d" % (c,len(prefs))
    # Find the most similar items to this one
    scores=topMatches(prefs,user,n=n,similarity=sim_pearson)
    result[user]=scores
  return result
```

- Define a cumulative difference score after sorting, and report the group of five that have the smallest score.

```
cumulative={}
for user in simi_users:
  cumulative.setdefault(user,0)

  for (key,value) in simi_users[user]:
    num+=key
    raters.append(value)

  cumulative[user]=(num,raters)
  print cumulative[user]

sorted_x = sorted(cumulative.iteritems(), key=operator.itemgetter(1))
sorted_x.reverse()
print (" ##### number [7] solution ##### ")
for (key,value) in sorted_x[0:5]:
 print key,' &  ',value,' \\\\'
```

Table 7 show the five raters most agreed with each other.

1st User ID	other users ID	cumulative difference score
675	99, 908, 906, 886	0
628	98, 97, 932, 912	0
440	98, 939, 911, 908	0
170	98, 939, 861, 777	0
600	98, 936, 879, 874	0

Table 7: The five raters most agreed with each other

Problem 8

In this question I did the same thing as question seven, the only change was:

- Reporting the group of five that have the highest score.

```
sorted_x = sorted(cumulative.iteritems(), key=operator.itemgetter(1))
print (" ##### number [6] solution ##### ")
for (key,value) in sorted_x[0:5]:
 print key,' &  ',value,' \\\\'
```

Table 8 show the five raters most disagreed with each other.

1st User ID	other users ID	cumulative difference score
13	46, 876, 701, 397	2.641
655	384, 895, 816, 762	2.675
796	205, 333, 143, 812	2.823
130	511, 369, 172, 415	2.828
551	691, 132, 641, 675	2.854

Table 8: The five raters most disagreed with each other

Problem 9

In this question I did the following steps:

- I created two dictionaries one for men over forty and the other for men under forty *rating_m_up* and *rating_m_down*.
- I than iterate over *prefs* dictionary by key *user* and use the condition statement to find female user over forty and under forty, add all of their rating, and save them in the dictionaries.

```
rating_m_up={}
rating_m_down={}
for user in prefs.keys():
  if gender[user][0] < '40' and gender[user][1] == 'M':
    for key,value in prefs[user].iteritems():
              rating_m_down.setdefault(key,[])
              rating_m_down[key].append(value)
  elif gender[user][0] > '40' and gender[user][1] == 'M':
    for key,value in prefs[user].iteritems():
          rating_m_up.setdefault(key,[])
          rating_m_up[key].append(value)
  else:
       continue
```

- I created another two dictionaries for all movies and their user average rating by men over and under forty and called it *average_m_down* and *average_m_up*.

```
average_m_down = {}
for movie in rating_m_down.keys():
  average_m_down[movie] = mean(rating_m_down[movie])


average_m_up = {}
for movie in rating_m_up.keys():
  average_m_up[movie] = mean(rating_m_up[movie])
```

- I sorted all movies based on their average and reverse to get the highest average rating and print the highest movies.

```
sorted_md = sorted(average_m_down.iteritems(),
    key=operator.itemgetter(1))
sorted_md.reverse()


sorted_mu = sorted(average_m_up.iteritems(),
    key=operator.itemgetter(1))
sorted_mu.reverse()
print (" ##### number [9 B over ] solution ##### ")
for (key,value) in sorted_mu[0:30]:
 print key,' &  ',value,' \\\\'
```

Table 9 show the movies that were rated highest on average by men under 40.

Movies by men under 40	ratings
Delta of Venus (1994)	5.0
Santa with Muscles (1996)	5.0
Crossfire (1947)	5.0
Leading Man, The (1996)	5.0
Love Serenade (1996)	5.0
Prefontaine (1997)	5.0
Aiqing wansui (1994)	5.0
Love in the Afternoon (1957)	5.0
Star Kid (1997)	5.0
Angel Baby (1995)	5.0
Maya Lin: A Strong Clear Vision (1994)	5.0
Entertaining Angels: The Dorothy Day Story (1996)	5.0
Magic Hour, The (1998)	5.0
Quiet Room, The (1996)	5.0
Saint of Fort Washington, The (1993)	5.0
Perfect Candidate, A (1996)	5.0
Letter From Death Row, A (1998)	5.0

Table 9: The movie that was rated highest on average by men under 40

Table 10 show the movies that were rated highest on average by men over 40.

Problem 10

In this question I did the same thing as in question nine with only change to the condition statement as following:

Movies by men over 40	ratings
Hearts and Minds (1996)	5.0
Faithful (1996)	5.0
Marlene Dietrich: Shadow and Light (1996)	5.0
Strawberry and Chocolate (Fresa y chocolate) (1993)	5.0
Late Bloomers (1996)	5.0
Solo (1996)	5.0
Grateful Dead (1995)	5.0
Prefontaine (1997)	5.0
Rendezvous in Paris (Rendez-vous de Paris, Les) (1995)	5.0
World of Apu, The (Apur Sansar) (1959)	5.0
Aparajito (1956)	5.0
Ace Ventura: When Nature Calls (1995)	5.0
Star Kid (1997)	5.0
Two or Three Things I Know About Her (1966)	5.0
Poison Ivy II (1995)	5.0
Double Happiness (1994)	5.0
Little City (1998)	5.0
Boxing Helena (1993)	5.0
Spice World (1997)	5.0
They Made Me a Criminal (1939)	5.0
Great Day in Harlem, A (1994)	5.0
Little Princess, The (1939)	5.0
Unstrung Heroes (1995)	5.0
Leading Man, The (1996)	5.0
Indian Summer (1996)	5.0

Table 10: The movie that was rated highest on average by men over 40

```
if gender[user][0] < '40' and gender[user][1] == 'F':
  for key,value in prefs[user].iteritems():
            rating_w_down.setdefault(key,[])
            rating_w_down[key].append(value)
elif gender[user][0] > '40' and gender[user][1] == 'F':
```

Table 11 show the movies that were rated highest on average by women under 40.
Table 12 show the movies that were rated highest on average by women over 40.

Movies by women under 40	ratings
Backbeat (1993)	5.0
Prefontaine (1997)	5.0
Telling Lies in America (1997)	5.0
Faster Pussycat! Kill! Kill! (1965)	5.0
Year of the Horse (1997)	5.0
Mina Tannenbaum (1994)	5.0
Maya Lin: A Strong Clear Vision (1994)	5.0
Nico Icon (1995)	5.0
The Umbrellas of Cherbourg, (Parapluies de Cherbourg, Les) (1964)	5.0
Everest (1998)	5.0
Heaven's Prisoners (1996)	5.0
The Wedding Gift, (1994)	5.0
The Horseman on the Roof, (Hussard sur le toit, Le) (1995)	5.0
Grace of My Heart (1996)	5.0
Someone Else's America (1995)	5.0
Don't Be a Menace to South Central While Drinking Your Juice in the Hood (1996)	5.0
Stripes (1981)	5.0

Table 11: The movie that was rated highest on average by women under 40

Refrences

[1] `http://www.grouplens.org/node/73`.

[2] `http://www.cs.odu.edu/~mln/teaching/cs595-f13/?method=getElement&elem~assignments~a8~a8.txt`.

Movies by women over 40	ratings
Shallow Grave (1994)	5.0
Great Dictator, The (1940)	5.0
Visitors, The (Visiteurs, Les) (1993)	5.0
Shall We Dance? (1937)	5.0
In the Bleak Midwinter (1995)	5.0
Funny Face (1957)	5.0
Ma vie en rose (My Life in Pink) (1997)	5.0
Swept from the Sea (1997)	5.0
Best Men (1997)	5.0
Foreign Correspondent (1940)	5.0
Tombstone (1993)	5.0
Wrong Trousers, The (1993)	5.0
Top Hat (1935)	5.0
Quest, The (1996)	5.0
Balto (1995)	5.0
Angel Baby (1995)	5.0
Band Wagon, The (1953)	5.0
Letter From Death Row, A (1998)	5.0
Mina Tannenbaum (1994)	5.0
Mary Shelley's Frankenstein (1994)	5.0
Gold Diggers: The Secret of Bear Mountain (1995)	5.0
Nightmare Before Christmas, The (1993)	5.0
Grand Day Out, A (1992)	5.0
Bride of Frankenstein (1935)	5.0
Pocahontas (1995)	5.0
Safe (1995)	5.0

Table 12: The movie that was rated highest on average by women over 40