movies.dat

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)|0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0

2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)|0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0

3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms%20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0

4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty%20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0

5|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Copycat%20(1995)|0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|0|1|0|0

test.dat

1 2 3 100

1 4 3 100

2 1 2 100

2 2 3 100

2 3 2 100

2 4 4 100

3 2 4 100

3 3 2 100

3 4 5 100

4 3 4 100

4 4 3 100

4 5 4 100

5 1 3 100

5 2 3 100

5 4 5 100

5 5 5 100

result

List of movies are:

{ '1': 'Toy Story (1995)',

'2': 'GoldenEye (1995)',

'3': 'Four Rooms (1995)',

'4': 'Get Shorty (1995)',

'5': 'Copycat (1995)'}

The preference list is:

{ '1': {'GoldenEye (1995)': 3.0,

'Get Shorty (1995)': 3.0},

'2': { 'Four Rooms (1995)': 2.0,

'Get Shorty (1995)': 4.0,

'GoldenEye (1995)': 3.0,

'Toy Story (1995)': 2.0},

'3': { 'Four Rooms (1995)': 2.0,

'Get Shorty (1995)': 5.0,

'GoldenEye (1995)': 4.0},

'4': { 'Copycat (1995)': 4.0,

'Four Rooms (1995)': 4.0,

'Get Shorty (1995)': 3.0},

'5': { 'Copycat (1995)': 5.0,

'Get Shorty (1995)': 5.0,

'GoldenEye (1995)': 3.0,

'Toy Story (1995)': 3.0}}

==============================================

Compute pair-wise similarity, similarity between users "1" and "2"

similarity between user 1 and user 2

prefs["1"]: {'GoldenEye (1995)': 3.0,

'Get Shorty (1995)': 3.0}

prefs["2"]: { 'Four Rooms (1995)': 2.0,

'Get Shorty (1995)': 4.0,

'GoldenEye (1995)': 3.0,

'Toy Story (1995)': 2.0}

Shared items, si: ['GoldenEye (1995)', 'Get Shorty (1995)']

distance = sqrt( (prefs["1"][si[0]]-prefs["2"][si[0]])^2 +

(prefs["2"][si[1]]-prefs["2"][si[1]]^2))

= sqrt( (prefs["1"]['GoldenEye'] - prefs["2"]['GoldenEye'])^2 +

(prefs["2"]['Get Shorty'] - prefs["2"]['Get Shorty']^2))

= sqrt ((3.0-3.0)^2 + (3.0-4.0)^2) = 1.0

similarity = 1/(1+distance) = 0.5

==============================================

similarity between user 3 and guest(1): 0.3090

Four Rooms (1995) : 2, 0.618(weighted)

similarity between user 2 and guest(1): 0.5

rating contribution:

Four Rooms (1995) : 2, 1.0(weighted)

Toy Story (1995) : 2, 1.0(weighted)

similarity between user 5 and guest(1): 0.3333

rating contribution:

Copycat (1995) 5, 1.66666(weighted)

Toy Story (1995) 3, 1.0(weighted)

similarity between user 4 and guest(1): 1.0

Copycat (1995) 4, 5.6666(weighted)

Four Rooms (1995)4, 5.618(weighted)

Final rating:

[(4.25, 'Copycat (1995)'),

(3.1055728090000843, 'Four Rooms (1995)'),

(2.4000000000000004, 'Toy Story (1995)')]

user ratings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Toy Story | Golden Eye | Four Rooms | Get Shorty | Copycat |
| “1” |  | 3 |  | 3 |  |
| “2” | 2 | 3 | 2 | 4 |  |
| “3” |  | 4 | 2 | 5 |  |
| “4” |  |  | 4 | 3 | 4 |
| “5” | 3 | 3 |  | 5 | 5 |

Rating computations, only interested in 3 movies:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Similarity | Toy Story | Four Rooms | Copycat |
| “2” | 0.5 | 1.0 | 1.0 |  |
| “3” | 0.309 |  | 0.618 |  |
| “4” | 1.0 |  | 4.0 | 4.0 |
| “5” | 0.33 | 0.99 |  | 1.66 |
| Subtotal |  | 1.99 | 5.618 | 5.66 |
| Weight sum |  | 0.83 | 1.809 | 1.33 |
| Rating |  | 2.4 | 3.1055 | 4.25 |

===================================================================

**Evaluation process**

For each user in the test data

For each movie rated by this user

Remove the rating of that movie for the user

Compute a rating for the movie

Add it to the list of movie ratings

Compute the pearson correlation of the estimated ratings with the actual ratings

Compute the average pearson correlation of all the users in the test data

Use the first 100 users as the test set. (also remove them from the training set)

Example: Test set: “1”,“2”; Training set: “3”,”4”,”5”

* For user “1”,
  + Compute rating for ‘Golden Eye’🡪 record rating=[4.3]
  + Compute rating for ‘Get Shorty’ 🡪 record rating=[4.3, 2.5]
  + R1=Correlation between [4.3, 2.5] and [3, 3]
* For user “2”
  + Compute rating for ‘Toy Story’🡪 record rating=[3]
  + Compute rating for ‘Golden Eye’🡪 record rating=[3, 4]
  + Compute rating for ‘Four Rooms’🡪 record rating=[3, 4, 2.5]
  + Compute rating for ‘Get Shorty’ 🡪 record rating=[3, 4, 2.5, 3.2]
  + R2=Correlation between [3, 4, 2.5, 3.2] and [2, 3, 2, 4]
* Compute the average of R1 and R2