Group Id:66

Project-II

Abstract: A book recommendation system used in real world

Recommendation Technique:

<u>Collaborative filtering:</u>

Typically, the workflow of a collaborative filtering system is:

1. A user expresses his or her preferences by rating items (e.g. books, movies or CDs) of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.
2. The system matches this user's ratings against other users' and finds the people with most "similar" tastes.
3. With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item)

1) User Based Collaborative System:

steps:

1. Look for users who share the same rating patterns with the active user (the user whom the prediction is for).
2. Use the ratings from those like-minded users found in step 1 to calculate a prediction for the active user

2) Item Based Collaborative System:

Steps:

1.Build an item-item matrix determining relationships between pairs of items

2.Infer the tastes of the current user by examining the matrix and matching that user's data

Disadvantage of User based collaborative filtering:

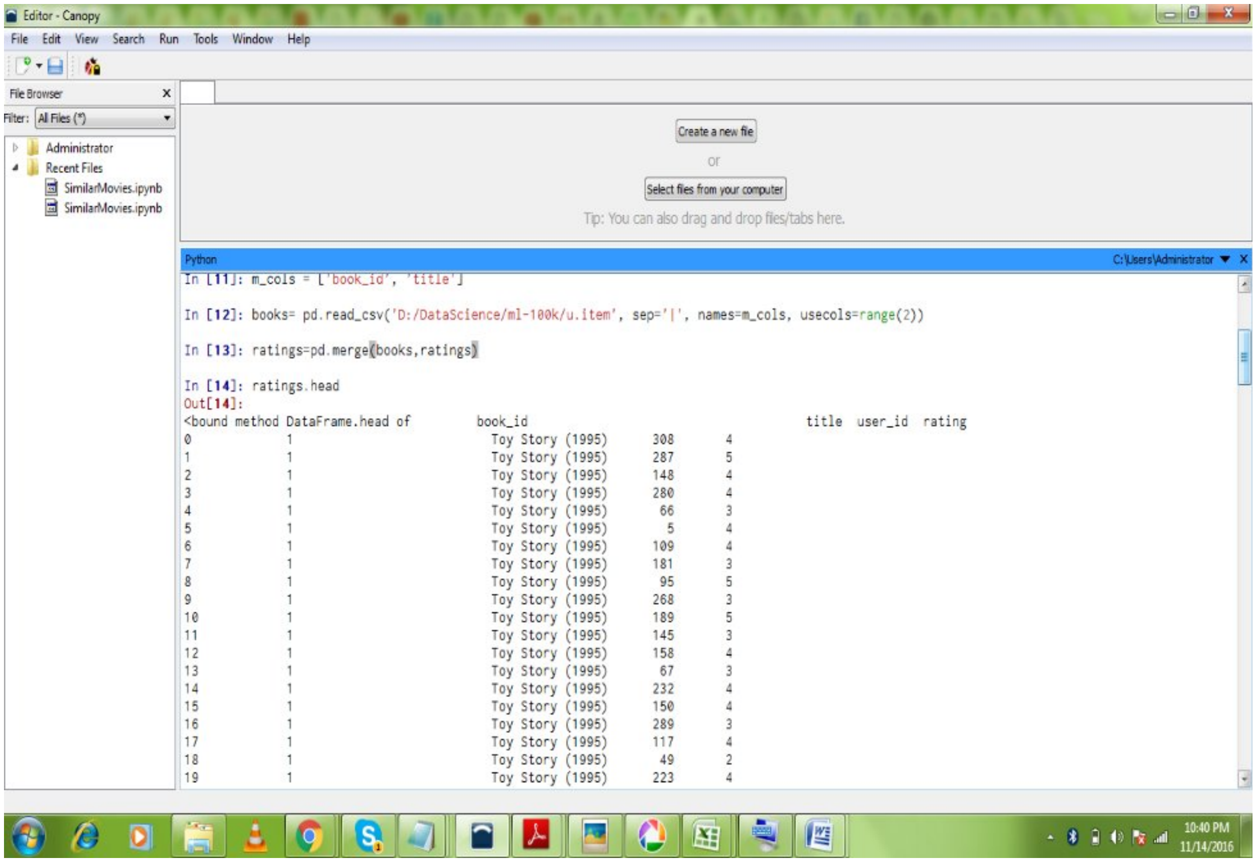
- People are fickle,taste changes
- Many more people than things(scalability)

Group Id:66

Item Based Collaborative System Implementation

```
import pandas as pd

r_cols = ['user_id', 'book_id', 'rating']

ratings = pd.read_csv('D:/DataScience/ml-100k/u.data', sep='\t',
names=r_cols, usecols=range(3))

m_cols = ['book_id', 'title']

books= pd.read_csv('D:/DataScience/ml-100k/u.item', sep='|',
names=m_cols, usecols=range(2))

books.head

ratings=pd.merge(books,ratings)

ratings.head
```

Group Id:66



- pivot_table function on a DataFrame will construct a user / book rating matrix

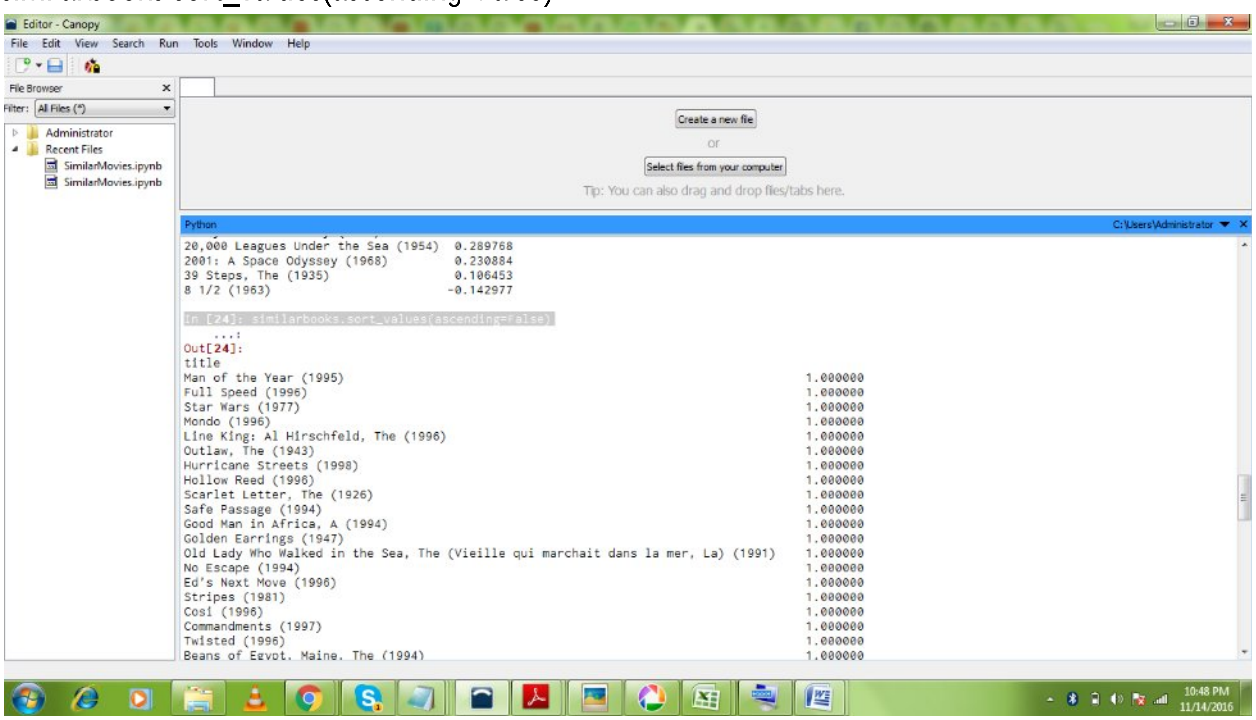
bookRatings = ratings.pivot_table(index=['user_id'],columns=['title'],values='rating')

starWarsRatings = bookRatings['Star Wars (1977)']

starWarsRatings.head()
Out[17]:
user_id
0 5.0
1 5.0
2 5.0
3 NaN
4 5.0
Name: Star Wars (1977), dtype: float64

- Pandas' corrwith function makes it really easy to compute the pairwise correlation of Star Wars' vector of user rating with every other books

```
similarbooks = bookRatings.corrwith(starWarsRatings)

similarbooks = similarbooks.dropna()

df = pd.DataFrame(similarbooks)

df.head(10)
Out[23]:
                                        0
title
'Til There Was You (1997)           0.872872
1-900 (1994)                       -0.645497
101 Dalmatians (1996)               0.211132
12 Angry Men (1957)                 0.184289
187 (1997)                          0.027398
2 Days in the Valley (1996)         0.066654
20,000 Leagues Under the Sea (1954)  0.289768
2001: A Space Odyssey (1968)        0.230884
39 Steps, The (1935)                0.106453
8 1/2 (1963)                       -0.142977

similarbooks.sort_values(ascending=False)
```



- Here,results are probably getting messed up by books that have only been viewed by a handful of people who also happened to like Star Wars. So we need to get rid of books that were only watched by a few people

```
import numpy as np

bookStats = ratings.groupby('title').agg({'rating': [np.size, np.mean]})

bookStats.head()
```

rating		
	size	**mean**
title		
'Til There Was You (1997)	9	2.333333
1-900 (1994)	5	2.600000
101 Dalmatians (1996)	109	2.908257
12 Angry Men (1957)	125	4.344000
187 (1997)	41	3.024390

- Getting rid of any books rated by fewer than 100 people, and check the top-rated ones that are left:

```
popularbooks = bookStats['rating']['size'] >= 100
bookStats[popularbooks].sort_values([('rating', 'mean')], ascending=False)[:15]
```

Out[9]:

Group Id:66

	rating	
	size	**mean**
title		
Close Shave, A (1995)	112	4.491071
Schindler's List (1993)	298	4.466443
Wrong Trousers, The (1993)	118	4.466102
Casablanca (1942)	243	4.456790
Shawshank Redemption, The (1994)	283	4.445230
Rear Window (1954)	209	4.387560
Usual Suspects, The (1995)	267	4.385768
Star Wars (1977)	584	4.359589
12 Angry Men (1957)	125	4.344000

	rating	
	size	mean
title		
Citizen Kane (1941)	198	4.292929
To Kill a Mockingbird (1962)	219	4.292237
One Flew Over the Cuckoo's Nest (1975)	264	4.291667
Silence of the Lambs, The (1991)	390	4.289744
North by Northwest (1959)	179	4.284916
Godfather, The (1972)	413	4.283293

100 might still be too low, but these results are pretty good

- joining data with our original set of similar books to Star Wars:

```
df = bookStats[popularbooks].join(pd.DataFrame(similarbooks, columns=['similarity']))
df.head()
```

Out[11]:

Group Id:66

	(rating, size)	(rating, mean)	similarity
title			
101 Dalmatians (1996)	109	2.908257	0.211132
12 Angry Men (1957)	125	4.344000	0.184289
2001: A Space Odyssey (1968)	259	3.969112	0.230884
Absolute	12	3.3700	0.08544

	(rating, size)	(rating, mean)	similarity
title			
Power (1997)	7	79	0
Abyss, The (1989)	151	3.589404	0.2037709

- sort these new results by similarity score.

```
df.sort_values(['similarity'], ascending=False)[:15]
```

Out[12]:

	(rating, size)	(rating, mean)	similarity

title			
Star Wars (1977)	584	4.359589	1.000000
Empire Strikes Back, The (1980)	368	4.206522	0.748353
Return of the Jedi (1983)	507	4.007890	0.672556
Raiders of the Lost Ark (1981)	420	4.252381	0.536117
Austin Powers: International Man	130	3.246154	0.377433

	(rating, size)	**(rating, mean)**	**similarity**
title			
of Mystery (1997)			
Sting, The (1973)	241	4.058091	0.367538
Indiana Jones and the Last Crusade (1989)	331	3.930514	0.350107
Pinocchio (1940	10	3.6732	0.34786

Group Id:66

	(rating,size)	(rating,mean)	similarity
title			
)	1	67	8
Frighteners, The (1996)	115	3.23 47 83	0.3 32 72 9
L.A. Confidential (1997)	297	4.16 16 16	0.3 19 06 5
Wag the Dog (1997)	137	3.51 09 49	0.3 18 64 5
Dumbo	12	3.49	0.3 17

Group Id:66

	(rating, size)	(rating, mean)	similarity
title			
(1941)	3	5935	656
Bridge on the River Kwai, The (1957)	165	4.175758	0.316580
Philadelphia Story, The (1940)	104	4.115385	0.314272
Miracle on 34th Street	101	3.722772	0.310921

Group Id:66

	(rating, size)	(rating, mean)	similarity
title			
(1994)			

Conclusion:

Ideally we'd filter out the similar books with Star Wars

References:

http://files.grouplens.org/papers/www10_sarwar.pdf

http://www10.org/cdrom/papers/519/

http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/itembased.html