In [1]:
```
import pandas as pd
from scipy import sparse
```

In [7]:
```
# The dataset consisting of the ratings
ratings = pd.read_csv('ratings.csv')
ratings.head(5)
```

Out[7]:

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

In [8]:
```
# The dataset consisting of information of the movie
movies = pd.read_csv('movies.csv')
movies.head(5)
```

Out[8]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

In [9]:
```
# Lets merge both the datasets
ratings = pd.merge(movies,ratings).drop(['genres','timestamp'],axis=1)
print(ratings.shape)
ratings.head()
```

```
(100836, 4)
```

Out[9]:

	movieId	title	userId	rating
0	1	Toy Story (1995)	1	4.0
1	1	Toy Story (1995)	5	4.0
2	1	Toy Story (1995)	7	4.5
3	1	Toy Story (1995)	15	2.5
4	1	Toy Story (1995)	17	4.5

In [12]:
```
# Lets create a dataset consisting of user-ratings for different movies
userRatings = ratings.pivot_table(index=['userId'],columns=['title'],values='rati
print("Before: ",userRatings.shape)
userRatings.head(5)
```

Before: (610, 9719)

Out[12]:

title	'71 (2014)	'Hellboy': The Seeds of Creation (2004)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'Tis the Season for Love (2015)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	*batteries not included (1987)
userId										
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 9719 columns

In [13]:
```
# Removing the movies for which less than 10 users had given ratings
userRatings = userRatings.dropna(thresh=10, axis=1).fillna(0,axis=1)
print("After: ",userRatings.shape)
userRatings.head(5)
```

After: (610, 2269)

Out[13]:

title	'burbs, The (1989)	(500) Days of Summer (2009)	10 Cloverfield Lane (2016)	10 Things I Hate About You (1999)	10,000 BC (2008)	101 Dalmatians (1996)	101 Dalmatians (One Hundred and One Dalmatians) (1961)	12 Angry Men (1957)	12 Years a Slave (2013)	1: Hou (201
userId										
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0

5 rows × 2269 columns

In [14]:
```python
# Finding the (pearson) correlation matrix between the movies
corrMatrix = userRatings.corr(method='pearson')
corrMatrix.head(100)
```

Out[14]:

title	'burbs, The (1989)	(500) Days of Summer (2009)	10 Cloverfield Lane (2016)	10 Things I Hate About You (1999)	10,000 BC (2008)	101 Dalmatians (1996)	101 Dalmatians (One Hundred and One Dalmatians) (1961)	12 An M (19
title								
'burbs, The (1989)	1.000000	0.063117	-0.023768	0.143482	0.011998	0.087931	0.224052	0.0342
(500) Days of Summer (2009)	0.063117	1.000000	0.142471	0.273989	0.193960	0.148903	0.142141	0.1597
10 Cloverfield Lane (2016)	-0.023768	0.142471	1.000000	-0.005799	0.112396	0.006139	-0.016835	0.0317
10 Things I Hate About You (1999)	0.143482	0.273989	-0.005799	1.000000	0.244670	0.223481	0.211473	0.0117
10,000 BC (2008)	0.011998	0.193960	0.112396	0.244670	1.000000	0.234459	0.119132	0.0591
...	...	...	...	...	...	...	...	
Almost Famous (2000)	0.099554	0.209549	0.032088	0.296727	0.134434	0.118628	0.242958	0.0791
Along Came Polly (2004)	0.027287	0.282426	0.113213	0.193085	0.162678	0.180259	0.112928	0.1217
Along Came a Spider (2001)	0.064762	-0.003205	0.016372	0.085365	-0.018241	0.080388	0.094016	-0.0166
Amadeus (1984)	0.136013	0.084829	-0.055707	0.105783	-0.008620	0.055704	0.121697	0.2442
Amazing Spider-Man, The (2012)	0.083419	0.224961	0.149903	0.103802	0.278253	0.096137	0.152795	0.0705

100 rows × 2269 columns

In [16]:
```python
# Function that returns the similar movies using the correlation matrix
def get_similar(movie_name,rating):
    similar_ratings = corrMatrix[movie_name]*(rating-2.5)
    similar_ratings = similar_ratings.sort_values(ascending=False)
    #print(type(similar_ratings))
    return similar_ratings
```

In [17]:
```python
# Finding the similar movies based on the movies we have provided
romantic_lover = [("(500) Days of Summer (2009)",5),("Alice in Wonderland (2010)'
similar_movies = pd.DataFrame()
for movie,rating in romantic_lover:
    similar_movies = similar_movies.append(get_similar(movie,rating),ignore_inde>

similar_movies.head(10)
```

Out[17]:

	'burbs, The (1989)	(500) Days of Summer (2009)	10 Cloverfield Lane (2016)	10 Things I Hate About You (1999)	10,000 BC (2008)	101 Dalmatians (1996)	101 Dalmatians (One Hundred and One Dalmatians) (1961)	12 Angry Men (1957)	12 Y a S (2
0	0.157792	2.500000	0.356179	0.684973	0.484900	0.372257	0.355353	0.399389	0.33
1	-0.016276	0.203998	0.126834	0.113241	0.092218	0.085790	0.072825	0.097794	0.08
2	-0.304722	-0.062634	-0.214700	-0.118754	-0.037059	-0.063992	-0.170195	-0.280090	-0.01
3	-0.102988	-0.056808	-0.049655	-0.042987	-0.021729	-0.055422	-0.051115	-0.097954	-0.06

4 rows × 2269 columns

In [18]:
```
# Printing the list of top 20 similar movies
similar_movies.sum().sort_values(ascending=False).head(20)
```

Out[18]:
```
(500) Days of Summer (2009)                      2.584556
Alice in Wonderland (2010)                       1.395229
Silver Linings Playbook (2012)                   1.254800
Yes Man (2008)                                   1.116264
Adventureland (2009)                             1.112235
Marley & Me (2008)                               1.108381
About Time (2013)                                1.102192
Crazy, Stupid, Love. (2011)                      1.088757
50/50 (2011)                                     1.086517
Help, The (2011)                                 1.075963
Up in the Air (2009)                             1.053037
Holiday, The (2006)                              1.034470
Friends with Benefits (2011)                     1.030875
Notebook, The (2004)                             1.025880
Easy A (2010)                                    1.015771
Secret Life of Walter Mitty, The (2013)          0.997979
Perks of Being a Wallflower, The (2012)          0.967425
Toy Story 3 (2010)                               0.963276
Ugly Truth, The (2009)                           0.959079
Harry Potter and the Half-Blood Prince (2009)    0.954180
dtype: float64
```

In [19]:
```
# Checkiing another exmaple.
action_lover = [("Amazing Spider-Man, The (2012)",5),("Mission: Impossible III (2
similar_movies = pd.DataFrame()
for movie,rating in action_lover:
    similar_movies = similar_movies.append(get_similar(movie,rating),ignore_index

# Seems our recommender system is working pretty well. We are getting all movies
similar_movies.head(10)
similar_movies.sum().sort_values(ascending=False).head(20)
```

Out[19]:
```
Amazing Spider-Man, The (2012)                            3.233134
Mission: Impossible III (2006)                            2.874798
2 Fast 2 Furious (Fast and the Furious 2, The) (2003)     2.701477
Over the Hedge (2006)                                     2.229721
Crank (2006)                                              2.176259
Mission: Impossible - Ghost Protocol (2011)               2.159666
Hancock (2008)                                            2.156098
The Amazing Spider-Man 2 (2014)                           2.153677
Hellboy (2004)                                            2.137518
Snakes on a Plane (2006)                                  2.137396
Jumper (2008)                                             2.129716
Chronicles of Riddick, The (2004)                         2.121689
Tron: Legacy (2010)                                       2.111843
Fantastic Four (2005)                                     2.083022
X-Men: The Last Stand (2006)                              2.077530
Wreck-It Ralph (2012)                                     2.067907
Kung Fu Hustle (Gong fu) (2004)                           2.067457
Godzilla (2014)                                           2.061653
Incredible Hulk, The (2008)                               2.050104
Quantum of Solace (2008)                                  2.016189
dtype: float64
```