Matrix Factorization

As before, we'll start by importing the MovieLens 100K data set into a pandas DataFrame:

In [1]:
```
import pandas as pd

r_cols = ['user_id', 'movie_id', 'rating']
ratings = pd.read_csv('/Users/Zia/Google Drive/Bootcamp/Bootcamp Notes/\
Day 6 Recommendation Systems Notes/ml-100k/u.data', sep='\t', names=r_cols, usecols=range(3))
ratings.head()
```

Out[1]:

	user_id	movie_id	rating
0	0	50	5
1	0	172	5
2	0	133	1
3	196	242	3
4	186	302	3

In [2]:
```
m_cols = ['movie_id', 'title']
movies = pd.read_csv('/Users/Zia/Google Drive/Bootcamp/Bootcamp Notes/\
Day 6 Recommendation Systems Notes/ml-100k/u.item', sep='|', names=m_cols, usecols=range(2))
movies.head()
```

Out[2]:

	movie_id	title
0	1	Toy Story (1995)
1	2	GoldenEye (1995)
2	3	Four Rooms (1995)
3	4	Get Shorty (1995)
4	5	Copycat (1995)

In [3]:
```
ratings = pd.merge(movies, ratings)

ratings.head()
```

Out[3]:

	movie_id	title	user_id	rating
0	1	Toy Story (1995)	308	4
1	1	Toy Story (1995)	287	5
2	1	Toy Story (1995)	148	4
3	1	Toy Story (1995)	280	4
4	1	Toy Story (1995)	66	3

Now we'll pivot this table to construct a nice matrix of users and the movies they rated. NaN indicates missing data, or movies that a given user did not watch:

In [4]:
```
userRatings = ratings.pivot_table(index=['user_id'],columns=['title'],values='rating').fillna(0)
userRatings.head(20)
```

Out[4]:

title	'Til There Was You (1997)	1-900 (1994)	101 Dalmatians (1996)	12 Angry Men (1957)	187 (1997)	2 Days in the Valley (1996)	20,000 Leagues Under the Sea (1954)	2001: A Space Odyssey (1968)	3 Ninjas: High Noon At Mega Mountain (1998)	39 Steps, The (1935)	...	Yankee Zulu (1994)	Year of the Horse (1997)	You So Crazy (1994)	Young Frankenstein (1974)	Young Guns (1988)	Young Guns II (1990)	Young Poisoner's Handbook, The (1995)	Zeus and Roxa (199
user_id																			
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	2.0	5.0	0.0	0.0	3.0	4.0	0.0	0.0	...	0.0	0.0	0.0	5.0	3.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	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	2.0	0.0	0.0	0.0	0.0	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	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	2.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	...	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	4.0	0.0	0.0	0.0	5.0	0.0	0.0	...	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	4.0	0.0	0.0	5.0	5.0	0.0	4.0	...	0.0	0.0	0.0	5.0	3.0	0.0	3.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	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	0.0	0.0
10	0.0	0.0	0.0	5.0	0.0	0.0	0.0	5.0	0.0	4.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	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	4.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	2.0	4.0	0.0	0.0	2.0	5.0	1.0	4.0	...	0.0	2.0	0.0	5.0	3.0	0.0	1.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	5.0	0.0	0.0	0.0	4.0	0.0	0.0	...	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	3.0	0.0	0.0	0.0	3.0	0.0	0.0	...	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0

title	'Til There Was You (1997)	1-900 (1994)	101 Dalmatians (1996)	12 Angry Men (1957)	187 (1997)	2 Days in the Valley (1996)	20,000 Leagues Under the Sea (1954)	2001: A Space Odyssey (1968)	3 Ninjas: High Noon At Mega Mountain (1998)	39 Steps, The (1935)	...	Yankee Zulu (1994)	Year of the Horse (1997)	You So Crazy (1994)	Young Frankenstein (1974)	Young Guns (1988)	Young Guns II (1990)	Young Poisoner's Handbook, The (1995)	Zeus and Roxa (199
user_id																			
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

20 rows × 1664 columns

The next thing I need to do is de-mean the data (normalize by each users mean) and convert it from a dataframe to a numpy array.

In [20]:
```
print userRatings.shape
R = userRatings.as_matrix()
R
```

```
(944, 1664)
```

Out[20]:
```
array([[ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  2., ...,  0.,  4.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       ...,
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.]])
```

In [9]:
```
import numpy as np
user_ratings_mean = np.mean(R, axis = 1)
R_demeaned = R - user_ratings_mean.reshape(-1, 1)
R_demeaned
```

Out[9]:
```
array([[-0.00661058, -0.00661058, -0.00661058, ..., -0.00661058,
        -0.00661058, -0.00661058],
       [-0.58713942, -0.58713942,  1.41286058, ..., -0.58713942,
         3.41286058, -0.58713942],
       [-0.13581731, -0.13581731, -0.13581731, ..., -0.13581731,
        -0.13581731, -0.13581731],
       ...,
       [-0.05348558, -0.05348558, -0.05348558, ..., -0.05348558,
        -0.05348558, -0.05348558],
       [-0.19951923, -0.19951923, -0.19951923, ..., -0.19951923,
        -0.19951923, -0.19951923],
       [-0.34435096, -0.34435096, -0.34435096, ..., -0.34435096,
        -0.34435096, -0.34435096]])
```

Singular Value Decomposition

Numpy and Scipy have functions to do the singular value decomposition. We are going to use the Scipy function "svds" because it let's us choose the number of latent factors.

In [71]:
```
from scipy.sparse.linalg import svds
U, sigma, Vt = svds(R_demeaned, k = 50)
```

In [72]:
```
print len(U)
U
```

```
944
```

Out[72]:
```
array([[ 0.00858269,  0.01819776,  0.00915981, ...,  0.00066321,
        -0.00109957, -0.00265393],
       [-0.11564801,  0.03902162, -0.15329682, ...,  0.00555086,
         0.00523634, -0.06639073],
       [-0.02448783,  0.02722874, -0.00599876, ..., -0.05367411,
        -0.0459581 , -0.01313997],
       ...,
       [-0.01099894, -0.0065808 , -0.00123115, ..., -0.00761161,
        -0.02557046, -0.00824527],
       [-0.04418247, -0.02014794, -0.08744268, ..., -0.02411724,
         0.00750883, -0.02514059],
       [ 0.00897965,  0.00293514, -0.06995652, ...,  0.05699286,
        -0.01351556, -0.0447376 ]])
```

In [73]:
```
print len(sigma)
sigma
```

```
50
```

Out[73]:
```
array([  59.03747832,   59.34521924,   59.42916579,   59.74979572,
         60.32680146,   60.53231162,   60.87902146,   61.17678496,
         61.58699341,   61.9461599 ,   62.11163147,   62.45454861,
         62.69007186,   63.40094328,   63.84922013,   63.92349754,
         64.75151148,   64.83621046,   65.25994307,   66.25321926,
         66.83304469,   67.1682798 ,   67.58272569,   68.5465025 ,
         68.77482557,   69.55076687,   70.33307245,   72.25432761,
         72.71257308,   73.40705137,   75.04371729,   76.14199947,
         77.78356255,   78.34822642,   80.92514917,   83.3955813 ,
         83.8873092 ,   89.76803319,   92.2861299 ,   93.69857077,
         99.57567204,  106.48564158,  111.38678301,  126.01487807,
        138.27471724,  157.66562902,  158.77018762,  217.19384938,
        244.12084822,  521.27489713])
```

In [74]:
```
print Vt.shape
Vt
```

```
(50, 1664)
```

Out[74]:
```
array([[ 0.00227682,  0.00714813, -0.02842254, ..., -0.0013931 ,
        -0.02320351,  0.00082233],
       [-0.01090519,  0.00141279, -0.02567577, ..., -0.00154176,
         0.00998342, -0.00323196],
       [-0.00033571, -0.00088197,  0.00138266, ...,  0.00109754,
        -0.00348906,  0.00126878],
       ...,
       [ 0.00166576,  0.00031573,  0.02334028, ...,  0.0017573 ,
         0.00405857,  0.00071104],
       [-0.00096128,  0.00283342, -0.0225526 , ...,  0.00022061,
        -0.00168208,  0.00183455],
       [ 0.01544219,  0.0165472 , -0.00621039, ...,  0.01667376,
         0.01466418,  0.01697938]])
```

Done. The function returns exactly what I detailed earlier in this post, except that the Σ returned is just the values instead of a diagonal matrix. This is useful, but since I'm going to leverage matrix multiplication to get predictions I'll convert it to the diagonal matrix form.

In [75]:
```
sigma = np.diag(sigma)
sigma
```

Out[75]:
```
array([[  59.03747832,    0.        ,    0.        , ...,    0.        ,
           0.        ,    0.        ],
       [   0.        ,   59.34521924,    0.        , ...,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,   59.42916579, ...,    0.        ,
           0.        ,    0.        ],
       ...,
       [   0.        ,    0.        ,    0.        , ...,  217.19384938,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        , ...,    0.        ,
         244.12084822,    0.        ],
       [   0.        ,    0.        ,    0.        , ...,    0.        ,
           0.        ,  521.27489713]])
```

Making Predictions from the Decomposed Matrices

We now have everything I need to make movie ratings predictions for every user. I can do it all at once by following the math and matrix multiply U, Σ, and V^{T} back to get the rank $\mathrm{k} = 5$ approximation of R.

We also need to add the user means back to get the actual star ratings prediction.

In [120]:
```
all_user_predicted_ratings = np.dot(np.dot(U, sigma), Vt) + user_ratings_mean.reshape(-1, 1)
```

In [117]:
```
user_ratings_mean.shape
```

Out[117]:
```
(944,)
```

In [116]:
```
len(user_ratings_mean)
```

Out[116]:
```
944
```

In [111]:
```
len(user_ratings_mean.reshape(-1,1))
```

Out[111]:
```
944
```

In [118]:
```
user_ratings_mean.reshape(-1,1).shape
```

Out[118]:
```
(944, 1)
```

In [77]:
```
df = pd.DataFrame(all_user_predicted_ratings, columns=userRatings.columns)
df.head()
```

Out[77]:

title	'Til There Was You (1997)	1-900 (1994)	101 Dalmatians (1996)	12 Angry Men (1957)	187 (1997)	2 Days in the Valley (1996)	20,000 Leagues Under the Sea (1954)	2001: A Space Odyssey (1968)	3 Ninjas: High Noon At Mega Mountain (1998)	39 Steps, The (1935)	...	Yankee Zulu (1994)	Year of the Horse (1997)	You So Crazy (1994)	Young Frankenstein (1974)	
0	-0.031335	0.000821	-0.034204	-0.021158	0.001784	-0.153189	-0.083192	-0.004130	0.000049	-0.004028	...	0.003289	-0.000358	0.004787	-0.081861	0.0
1	-0.072076	-0.060237	0.598500	2.506863	-0.078158	0.186711	1.490075	5.766853	-0.019980	0.164837	...	-0.073835	-0.193848	-0.026537	3.436720	0.5
2	0.073818	0.000705	0.628568	0.443968	0.203148	0.347435	-0.250397	-1.393918	0.005043	-0.105983	...	-0.024160	0.021793	0.009585	-0.150768	-0.
3	-0.017711	0.015344	-0.023134	0.225587	0.651014	-0.161835	-0.158951	-0.141943	0.008309	-0.008342	...	-0.014991	0.104109	-0.009795	0.083725	0.1
4	-0.038605	0.005756	-0.102273	-0.138900	0.369944	-0.261871	0.051986	-0.200241	0.010192	-0.089849	...	0.024312	0.078282	0.016897	0.132723	0.2

5 rows × 1664 columns

In [96]:
```
userRatings.head()
```

Out[96]:

title	'Til There Was You (1997)	1-900 (1994)	101 Dalmatians (1996)	12 Angry Men (1957)	187 (1997)	2 Days in the Valley (1996)	20,000 Leagues Under the Sea (1954)	2001: A Space Odyssey (1968)	3 Ninjas: High Noon At Mega Mountain (1998)	39 Steps, The (1935)	...	Yankee Zulu (1994)	Year of the Horse (1997)	You So Crazy (1994)	Young Frankenstein (1974)	Young Guns (1988)	Young Guns II (1990)	Young Poisoner's Handbook, The (1995)	Zeus and Rox (199
user_id																			
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	2.0	5.0	0.0	0.0	3.0	4.0	0.0	0.0	...	0.0	0.0	0.0	5.0	3.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	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	2.0	0.0	0.0	0.0	0.0	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	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 1664 columns

In [97]:
```
similarmovies = df.loc[836].sort_values(ascending=False)
```

In [98]:
```
similarmovies
```

```
Out[98]: title
         Full Monty, The (1997)                     3.405390
         L.A. Confidential (1997)                   3.300887
         English Patient, The (1996)                3.000774
         Usual Suspects, The (1995)                 2.635652
         One Flew Over the Cuckoo's Nest (1975)     2.601655
         Apocalypse Now (1979)                      2.516532
         Pulp Fiction (1994)                        2.469998
         Rear Window (1954)                         2.390104
         Godfather: Part II, The (1974)             2.240454
         Contact (1997)                             2.222886
         Schindler's List (1993)                    2.216296
         Silence of the Lambs, The (1991)           2.161237
         Psycho (1960)                              2.115748
         Fugitive, The (1993)                       1.905062
         To Kill a Mockingbird (1962)               1.855826
         Chinatown (1974)                           1.786434
         Shawshank Redemption, The (1994)           1.744023
         Vertigo (1958)                             1.715048
         Raiders of the Lost Ark (1981)             1.698812
         Ulee's Gold (1997)                         1.686966
         Babe (1995)                                1.684000
         Godfather, The (1972)                      1.643696
         Scream (1996)                              1.618214
         Empire Strikes Back, The (1980)            1.578883
         Taxi Driver (1976)                         1.567272
         Citizen Kane (1941)                        1.550469
         Princess Bride, The (1987)                 1.516424
         Raising Arizona (1987)                     1.500312
         Brazil (1985)                              1.484685
         Chasing Amy (1997)                         1.418686
                                                      ...
         Pinocchio (1940)                          -0.341060
         2 Days in the Valley (1996)               -0.341290
         Hercules (1997)                           -0.372682
         Star Trek: The Motion Picture (1979)      -0.378680
         Bram Stoker's Dracula (1992)              -0.379561
         Circle of Friends (1995)                  -0.386116
         Benny & Joon (1993)                       -0.394437
         Maverick (1994)                           -0.397027
         Donnie Brasco (1997)                      -0.401247
         Snow White and the Seven Dwarfs (1937)    -0.404122
         Beauty and the Beast (1991)               -0.412347
```

```
Jurassic Park (1993)                        -0.414852
Frighteners, The (1996)                     -0.428587
Pump Up the Volume (1990)                   -0.472337
Con Air (1997)                              -0.485526
Heavy Metal (1981)                          -0.502605
Men in Black (1997)                         -0.516610
Die Hard (1988)                             -0.530522
Restoration (1995)                          -0.553535
Primal Fear (1996)                          -0.575033
Hunchback of Notre Dame, The (1996)         -0.580394
Aladdin (1992)                              -0.596069
Rock, The (1996)                            -0.655514
Bound (1996)                                -0.657587
Ransom (1996)                               -0.658074
Fifth Element, The (1997)                   -0.696236
Face/Off (1997)                             -0.715200
Jerry Maguire (1996)                        -0.726648
Toy Story (1995)                            -0.884202
Leaving Las Vegas (1995)                    -0.987624
Name: 836, Length: 1664, dtype: float64
```

In [99]:

```
myRatings = userRatings.loc[836].sort_values(ascending=False).replace(0.0, np.nan).dropna()
```

In [100]:
```
myRatings
```

```
Out[100]: title
          Rosewood (1997)                               5.0
          Godfather: Part II, The (1974)                5.0
          Chinatown (1974)                              5.0
          Laura (1944)                                  5.0
          Psycho (1960)                                 5.0
          Raging Bull (1980)                            5.0
          Raiders of the Lost Ark (1981)                5.0
          Cinema Paradiso (1988)                        5.0
          Arsenic and Old Lace (1944)                   5.0
          Full Monty, The (1997)                        5.0
          Cool Hand Luke (1967)                         5.0
          Rear Window (1954)                            5.0
          Manchurian Candidate, The (1962)              5.0
          One Flew Over the Cuckoo's Nest (1975)        5.0
          Usual Suspects, The (1995)                    5.0
          L.A. Confidential (1997)                      5.0
          Return of the Pink Panther, The (1974)        5.0
          Being There (1979)                            5.0
          Schindler's List (1993)                       5.0
          Apocalypse Now (1979)                         5.0
          Blade Runner (1982)                           4.0
          Streetcar Named Desire, A (1951)              4.0
          Indiana Jones and the Last Crusade (1989)     4.0
          Lost Highway (1997)                           4.0
          Jean de Florette (1986)                       4.0
          When Harry Met Sally... (1989)                4.0
          Contact (1997)                                4.0
          It's a Wonderful Life (1946)                  4.0
          Raising Arizona (1987)                        4.0
          Day the Earth Stood Still, The (1951)         4.0
          Koyaanisqatsi (1983)                          4.0
          Shine (1996)                                  4.0
          Soul Food (1997)                              4.0
          Pulp Fiction (1994)                           4.0
          Clerks (1994)                                 3.0
          Cop Land (1997)                               3.0
          Amistad (1997)                                3.0
          Seven Years in Tibet (1997)                   3.0
          English Patient, The (1996)                   3.0
          Chasing Amy (1997)                            3.0
          Citizen Kane (1941)                           3.0
          Sweet Hereafter, The (1997)                   3.0
```

```
Kundun (1997)                                  2.0
Crooklyn (1994)                                2.0
Event Horizon (1997)                           2.0
Mary Poppins (1964)                            2.0
Murder at 1600 (1997)                          2.0
Scream (1996)                                  1.0
Evita (1996)                                   1.0
She's So Lovely (1997)                         1.0
Name: 836, dtype: float64
```

In [101]:

```
myRatings.index
```

Out[101]:

```
Index([u'Rosewood (1997)', u'Godfather: Part II, The (1974)',
       u'Chinatown (1974)', u'Laura (1944)', u'Psycho (1960)',
       u'Raging Bull (1980)', u'Raiders of the Lost Ark (1981)',
       u'Cinema Paradiso (1988)', u'Arsenic and Old Lace (1944)',
       u'Full Monty, The (1997)', u'Cool Hand Luke (1967)',
       u'Rear Window (1954)', u'Manchurian Candidate, The (1962)',
       u'One Flew Over the Cuckoo's Nest (1975)',
       u'Usual Suspects, The (1995)', u'L.A. Confidential (1997)',
       u'Return of the Pink Panther, The (1974)', u'Being There (1979)',
       u'Schindler's List (1993)', u'Apocalypse Now (1979)',
       u'Blade Runner (1982)', u'Streetcar Named Desire, A (1951)',
       u'Indiana Jones and the Last Crusade (1989)', u'Lost Highway (1997)',
       u'Jean de Florette (1986)', u'When Harry Met Sally... (1989)',
       u'Contact (1997)', u'It's a Wonderful Life (1946)',
       u'Raising Arizona (1987)', u'Day the Earth Stood Still, The (1951)',
       u'Koyaanisqatsi (1983)', u'Shine (1996)', u'Soul Food (1997)',
       u'Pulp Fiction (1994)', u'Clerks (1994)', u'Cop Land (1997)',
       u'Amistad (1997)', u'Seven Years in Tibet (1997)',
       u'English Patient, The (1996)', u'Chasing Amy (1997)',
       u'Citizen Kane (1941)', u'Sweet Hereafter, The (1997)',
       u'Kundun (1997)', u'Crooklyn (1994)', u'Event Horizon (1997)',
       u'Mary Poppins (1964)', u'Murder at 1600 (1997)', u'Scream (1996)',
       u'Evita (1996)', u'She's So Lovely (1997)'],
      dtype='object', name=u'title')
```

```
In [102]: myRatings.values
```

```
Out[102]: array([ 5.,  5.,  5.,  5.,  5.,  5.,  5.,  5.,  5.,  5.,  5.,  5.,  5.,
                  5.,  5.,  5.,  5.,  5.,  5.,  5.,  4.,  4.,  4.,  4.,  4.,  4.,
                  4.,  4.,  4.,  4.,  4.,  4.,  4.,  4.,  3.,  3.,  3.,  3.,  3.,
                  3.,  3.,  3.,  2.,  2.,  2.,  2.,  2.,  1.,  1.,  1.])
```

This is starting to look like something useful! Note that some of the same movies came up more than once, because they were similar to more than one movie I rated. We'll use groupby() to add together the scores from movies that show up more than once, so they'll count more:

```
In [103]: similarmovies.index
```

```
Out[103]: Index([u'Full Monty, The (1997)', u'L.A. Confidential (1997)',
                 u'English Patient, The (1996)', u'Usual Suspects, The (1995)',
                 u'One Flew Over the Cuckoo's Nest (1975)', u'Apocalypse Now (1979)',
                 u'Pulp Fiction (1994)', u'Rear Window (1954)',
                 u'Godfather: Part II, The (1974)', u'Contact (1997)',
                 ...
                 u'Hunchback of Notre Dame, The (1996)', u'Aladdin (1992)',
                 u'Rock, The (1996)', u'Bound (1996)', u'Ransom (1996)',
                 u'Fifth Element, The (1997)', u'Face/Off (1997)',
                 u'Jerry Maguire (1996)', u'Toy Story (1995)',
                 u'Leaving Las Vegas (1995)'],
               dtype='object', name=u'title', length=1664)
```

The last thing we have to do is filter out movies I've already rated, as recommending a movie I've already watched isn't helpful:

In [104]:
```
predictions = similarmovies.drop(myRatings.index)
predictions.head(10)
```

Out[104]:
```
title
Silence of the Lambs, The (1991)    2.161237
Fugitive, The (1993)                1.905062
To Kill a Mockingbird (1962)        1.855826
Shawshank Redemption, The (1994)    1.744023
Vertigo (1958)                      1.715048
Ulee's Gold (1997)                  1.686966
Babe (1995)                         1.684000
Godfather, The (1972)               1.643696
Empire Strikes Back, The (1980)     1.578883
Taxi Driver (1976)                  1.567272
Name: 836, dtype: float64
```

There we have it!