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

In [2]:
```
df=pd.read_csv("f:/dataset/recommend/movies_collaborative.csv")
df
```

Out[2]:

	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
...	...	...	...	...
100831	193581	Black Butler: Book of the Atlantic (2017)	184	4.0
100832	193583	No Game No Life: Zero (2017)	184	3.5
100833	193585	Flint (2017)	184	3.5
100834	193587	Bungo Stray Dogs: Dead Apple (2018)	184	3.5
100835	193609	Andrew Dice Clay: Dice Rules (1991)	331	4.0

100836 rows × 4 columns

In [3]:
```
user_movie=df.pivot_table(index='userId',columns='movieId',values='rating')
```

In [4]:
```
user_movie
```

Out[4]:

movieId	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**...**	**193565**	**193567**	**193571**	**193573**	**193**
userId																
1	4.0	NaN	4.0	NaN	NaN	4.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	N
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	N
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	N
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	N
5	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
606	2.5	NaN	NaN	NaN	NaN	NaN	2.5	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	N
607	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	N
608	2.5	2.0	2.0	NaN	NaN	NaN	NaN	NaN	NaN	4.0	...	NaN	NaN	NaN	NaN	N
609	3.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	...	NaN	NaN	NaN	NaN	N
610	5.0	NaN	NaN	NaN	NaN	5.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	N

610 rows × 9724 columns

In [5]:
```
user_movie.fillna(0.0,inplace=True)
```

In [6]:
```
user_movie
```

Out[6]:

movieId userId	1	2	3	4	5	6	7	8	9	10	...	193565	193567	193571	193573	193579	193581	1
1	4.0	0.0	4.0	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	
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	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	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	
5	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	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
606	2.5	0.0	0.0	0.0	0.0	0.0	2.5	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
607	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	
608	2.5	2.0	2.0	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	
609	3.0	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	
610	5.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	

610 rows × 9724 columns

In [7]:
```
from sklearn.neighbors import NearestNeighbors
```

In [9]:
```
model=NearestNeighbors(metric='cosine')
X=user_movie.values
model.fit(X)
```

Out[9]:
```
▾ NearestNeighbors
NearestNeighbors(metric='cosine')
```

In [10]:
```
model.kneighbors([X[0]],n_neighbors=5)
```

Out[10]:
```
(array([[1.11022302e-16, 6.42592290e-01, 6.48438482e-01, 6.54872948e-01,
         6.54965721e-01]]),
 array([[  0, 265, 312, 367,  56]], dtype=int64))
```

In [19]:
```
sim_user_movies=df[df.userId==266].movieId.values
sim_user_movies
```

Out[19]:
```
array([   1,    6,   16,   17,   21,   24,   32,   39,   45,   50,   64,
         69,   95,  110,  132,  135,  145,  198,  223,  235,  260,  288,
        296,  316,  353,  356,  367,  368,  377,  440,  441,  457,  474,
        476,  480,  493,  514,  541,  551,  555,  589,  592,  608,  694,
        733,  778,  785,  919,  924, 1036, 1042, 1060, 1079, 1080, 1089,
       1090, 1093, 1097, 1127, 1136, 1171, 1196, 1197, 1198, 1199, 1200,
       1206, 1208, 1210, 1213, 1215, 1220, 1222, 1240, 1242, 1259, 1261,
       1265, 1266, 1270, 1275, 1285, 1288, 1291, 1297, 1298, 1356, 1374,
       1375, 1376, 1394, 1405, 1466, 1476, 1485, 1500, 1517, 1527, 1573,
       1580, 1597, 1610, 1614, 1617, 1641, 1663, 1722, 1732, 1772, 1784,
       1866, 1883, 1884, 1911, 1912, 1923, 1968, 2000, 2028, 2064, 2094,
       2097, 2105, 2115, 2167, 2174, 2194, 2195, 2249, 2253, 2278, 2288,
       2300, 2302, 2321, 2353, 2355, 2384, 2387, 2395, 2396, 2428, 2471,
       2490, 2539, 2560, 2571, 2580, 2599, 2605, 2683, 2700, 2702, 2706,
       2710, 2712, 2716, 2770, 2791, 2797, 2804, 2812, 2863, 2890, 2916,
       2918, 2947, 2959, 2968, 2985, 3016, 3020, 3039, 3052, 3060, 3108,
       3113, 3142, 3156, 3265], dtype=int64)
```

In [20]:
```
target_user_movies=df[df.userId==1].movieId.values
```

In [21]:
```
target_user_movies
```

Out[21]:
```
array([   1,    3,    6,   47,   50,   70,  101,  110,  151,  157,  163,
        216,  223,  231,  235,  260,  296,  316,  333,  349,  356,  362,
        367,  423,  441,  457,  480,  500,  527,  543,  552,  553,  590,
        592,  593,  596,  608,  648,  661,  673,  733,  736,  780,  804,
        919,  923,  940,  943,  954, 1009, 1023, 1024, 1025, 1029, 1030,
       1031, 1032, 1042, 1049, 1060, 1073, 1080, 1089, 1090, 1092, 1097,
       1127, 1136, 1196, 1197, 1198, 1206, 1208, 1210, 1213, 1214, 1219,
       1220, 1222, 1224, 1226, 1240, 1256, 1258, 1265, 1270, 1275, 1278,
       1282, 1291, 1298, 1348, 1377, 1396, 1408, 1445, 1473, 1500, 1517,
       1552, 1573, 1580, 1587, 1617, 1620, 1625, 1644, 1676, 1732, 1777,
       1793, 1804, 1805, 1920, 1927, 1954, 1967, 2000, 2005, 2012, 2018,
       2028, 2033, 2046, 2048, 2054, 2058, 2078, 2090, 2093, 2094, 2096,
       2099, 2105, 2115, 2116, 2137, 2139, 2141, 2143, 2161, 2174, 2193,
       2253, 2268, 2273, 2291, 2329, 2338, 2353, 2366, 2387, 2389, 2395,
       2406, 2414, 2427, 2450, 2459, 2470, 2478, 2492, 2502, 2528, 2529,
       2542, 2571, 2580, 2596, 2616, 2617, 2628, 2640, 2641, 2644, 2648,
       2654, 2657, 2692, 2700, 2716, 2761, 2797, 2826, 2858, 2872, 2899,
       2916, 2944, 2947, 2948, 2949, 2959, 2985, 2987, 2991, 2993, 2997,
       3033, 3034, 3052, 3053, 3062, 3147, 3168, 3176, 3243, 3247, 3253,
       3273, 3386, 3439, 3440, 3441, 3448, 3450, 3479, 3489, 3527, 3578,
       3617, 3639, 3671, 3702, 3703, 3729, 3740, 3744, 3793, 3809, 4006,
       5060], dtype=int64)
```

In [22]:
```
sim_user_movies
```

Out[22]:
```
array([   1,    6,   16,   17,   21,   24,   32,   39,   45,   50,   64,
         69,   95,  110,  132,  135,  145,  198,  223,  235,  260,  288,
        296,  316,  353,  356,  367,  368,  377,  440,  441,  457,  474,
        476,  480,  493,  514,  541,  551,  555,  589,  592,  608,  694,
        733,  778,  785,  919,  924, 1036, 1042, 1060, 1079, 1080, 1089,
       1090, 1093, 1097, 1127, 1136, 1171, 1196, 1197, 1198, 1199, 1200,
       1206, 1208, 1210, 1213, 1215, 1220, 1222, 1240, 1242, 1259, 1261,
       1265, 1266, 1270, 1275, 1285, 1288, 1291, 1297, 1298, 1356, 1374,
       1375, 1376, 1394, 1405, 1466, 1476, 1485, 1500, 1517, 1527, 1573,
       1580, 1597, 1610, 1614, 1617, 1641, 1663, 1722, 1732, 1772, 1784,
       1866, 1883, 1884, 1911, 1912, 1923, 1968, 2000, 2028, 2064, 2094,
       2097, 2105, 2115, 2167, 2174, 2194, 2195, 2249, 2253, 2278, 2288,
       2300, 2302, 2321, 2353, 2355, 2384, 2387, 2395, 2396, 2428, 2471,
       2490, 2539, 2560, 2571, 2580, 2599, 2605, 2683, 2700, 2702, 2706,
       2710, 2712, 2716, 2770, 2791, 2797, 2804, 2812, 2863, 2890, 2916,
       2918, 2947, 2959, 2968, 2985, 3016, 3020, 3039, 3052, 3060, 3108,
       3113, 3142, 3156, 3265], dtype=int64)
```

In [26]:
```
s1={10,20,30}
s2={30,40,50}
print(s1 | s2)
print(s1 & s2)
print(s1-s2)
print(s2-s1)
```

```
{50, 20, 40, 10, 30}
{30}
{10, 20}
{40, 50}
```

In [28]:
```
recommend_movie=set(sim_user_movies)-set(target_user_movies)
```

In [31]:
```
df_recmd_movies=df[(df.userId==266) & (df.movieId.isin(recommend_movie))]
```

In [35]:
```
df_recmd_movies.sort_values('rating',ascending=False).head()
```

Out[35]:

	movieId	title	userId	rating
903	16	Casino (1995)	266	5.0
18769	778	Trainspotting (1996)	266	5.0
22888	1079	Fish Called Wanda, A (1988)	266	5.0
25215	1199	Brazil (1985)	266	5.0
25309	1200	Aliens (1986)	266	5.0

In [56]:

```
df=pd.read_csv("f:/dataset/recommend/movies_collaborative.csv")
user_movie=df.pivot_table(index='userId',columns='movieId',values='rating')
user_movie.fillna(0.0,inplace=True)
model=NearestNeighbors(metric='cosine')
X=user_movie.values
model.fit(X)

uid=int(input("enter userId:"))
final_result_df=pd.DataFrame()
title=[]
rating=[]
if uid>=1 and uid<=610:
    sim_score,idx=model.kneighbors([X[uid-1]],n_neighbors=2)
    sim_user_movies=df[df.userId==idx[0][1]+1].movieId.values
    target_user_movies=df[df.userId==uid].movieId.values
    recommend_movie=set(sim_user_movies)-set(target_user_movies)
    df_recmd_movies=df[(df.userId==idx[0][1]+1) & (df.movieId.isin(recommend_movie))]
    result_df=df_recmd_movies.sort_values('rating',ascending=False).head()
    print("Similar User:",idx[0][1]+1)
    print("---------Top 5 Recommended Movies----------")

    for t,r in zip(result_df.title,result_df.rating):
        title.append(t)
        rating.append(r)
else:
    print("user does not exist")

final_result_df['title']=title
final_result_df['rating']=rating
final_result_df
```

```
user does not exist
```

Out[56]:

title	rating

In [ ]: