In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
movies_df = pd.read_csv('D://sistem cerdas//recom//movies.csv',usecols=['movieId','title
'],dtype={'movieId': 'int32', 'title': 'str'})
rating_df=pd.read_csv('D://sistem cerdas//recom//ratings.csv',usecols=['userId', 'movieId
', 'rating'],
dtype={'userId': 'int32', 'movieId': 'int32', 'rating': 'float32'})
```

In [3]:

```
movies_df.head()
```

Out[3]:

	movieId	title
0	1	Toy Story (1995)
1	2	Jumanji (1995)
2	3	Grumpier Old Men (1995)
3	4	Waiting to Exhale (1995)
4	5	Father of the Bride Part II (1995)

In [4]:

```
rating_df.head()
```

Out[4]:

	userId	movieId	rating
0	1	1	4.0
1	1	3	4.0
2	1	6	4.0
3	1	47	5.0
4	1	50	5.0

In [5]:

```
df = pd.merge(rating_df,movies_df,on='movieId')
df.head()
```

Out[5]:

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

In [6]:

```
combine_movie_rating = df.dropna(axis = 0, subset = ['title'])
```

```
movie_ratingCount = (combine_movie_rating.
     groupby(by = ['title'])['rating'].
     count().
     reset_index().
     rename(columns = {'rating': 'totalRatingCount'})
     [['title', 'totalRatingCount']]
    )
movie_ratingCount.head()
```

Out[6]:

	title	totalRatingCount
0	'71 (2014)	1
1	'Hellboy': The Seeds of Creation (2004)	1
2	'Round Midnight (1986)	2
3	'Salem's Lot (2004)	1
4	'Til There Was You (1997)	2

In [7]:

```
rating_with_totalRatingCount = combine_movie_rating.merge(movie_ratingCount, left_on = '
title', right_on = 'title', how = 'left')
rating_with_totalRatingCount.head()
```

Out[7]:

	userId	movieId	rating	title	totalRatingCount
0	1	1	4.0	Toy Story (1995)	215
1	5	1	4.0	Toy Story (1995)	215
2	7	1	4.5	Toy Story (1995)	215
3	15	1	2.5	Toy Story (1995)	215
4	17	1	4.5	Toy Story (1995)	215

In [8]:

```
pd.set_option('display.float_format', lambda x: '%.3f' % x)
print(movie_ratingCount['totalRatingCount'].describe())
```

```
count   9719.000
mean      10.375
std       22.406
min        1.000
25%        1.000
50%        3.000
75%        9.000
max      329.000
Name: totalRatingCount, dtype: float64
```

In [9]:

```
popularity_threshold = 50
rating_popular_movie= rating_with_totalRatingCount.query('totalRatingCount >= @popularity
_threshold')
rating_popular_movie.head()
```

Out[9]:

	userId	movieId	rating	title	totalRatingCount
0	1	1	4.000	Toy Story (1995)	215
1	5	1	4.000	Toy Story (1995)	215
2	7	1	4.500	Toy Story (1995)	215
3	15	1	2.500	Toy Story (1995)	215

3	15	1	2.500	Toy Story (1995)	215
	userId	movieId	rating	title	totalRatingCount
4	17	1	4.500	Toy Story (1995)	215

In [10]:

```
rating_popular_movie.shape
```

Out[10]:

```
(41362, 5)
```

In [11]:

```
movie_features_df=rating_popular_movie.pivot_table(index='title',columns='userId',values
='rating').fillna(0)
movie_features_df.head()
```

Out[11]:

userId	1	2	3	4	5	6	7	8	9	10	...	601	602	603	604	605	606	607
title																		
10 Things I Hate About You (1999)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	3.000	0.000	5.000	0.000	0.000
12 Angry Men (1957)	0.000	0.000	0.000	5.000	0.000	0.000	0.000	0.000	0.000	0.000	...	5.000	0.000	0.000	0.000	0.000	0.000	0.000
2001: A Space Odyssey (1968)	0.000	0.000	0.000	0.000	0.000	0.000	4.000	0.000	0.000	0.000	...	0.000	0.000	5.000	0.000	0.000	5.000	0.000
28 Days Later (2002)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	0.000	0.000	0.000	0.000	0.000
300 (2007)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.000	...	0.000	0.000	0.000	0.000	3.000	0.000	0.000

5 rows × 606 columns

In [12]:

```
from scipy.sparse import csr_matrix

movie_features_df_matrix = csr_matrix(movie_features_df.values)

from sklearn.neighbors import NearestNeighbors


model_knn = NearestNeighbors(metric = 'cosine', algorithm = 'brute')
model_knn.fit(movie_features_df_matrix)
```

Out[12]:

```
NearestNeighbors(algorithm='brute', metric='cosine')
```

In [13]:

```
movie_features_df.shape
```

Out[13]:

```
(450, 606)
```

In [14]:

In [14]:

```
query_index = np.random.choice(movie_features_df.shape[0])
print(query_index)
query_index =2
```

```
350
```

In [15]:

```
distances, indices = model_knn.kneighbors(movie_features_df.iloc[query_index,:].values.reshape(1, -1), n_neighbors = 6)
```

In [16]:

```
movie_features_df.head()
```

Out[16]:

userId	1	2	3	4	5	6	7	8	9	10	...	601	602	603	604	605	606	607
title																		
10 Things I Hate About You (1999)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	3.000	0.000	5.000	0.000	0.000
12 Angry Men (1957)	0.000	0.000	0.000	5.000	0.000	0.000	0.000	0.000	0.000	0.000	...	5.000	0.000	0.000	0.000	0.000	0.000	0.000
2001: A Space Odyssey (1968)	0.000	0.000	0.000	0.000	0.000	0.000	4.000	0.000	0.000	0.000	...	0.000	0.000	5.000	0.000	0.000	5.000	0.000
28 Days Later (2002)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	0.000	0.000	0.000	0.000	0.000
300 (2007)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.000	...	0.000	0.000	0.000	0.000	3.000	0.000	0.000

5 rows × 606 columns

In [17]:

```
for i in range(0, len(distances.flatten())):
    if i == 0:
        print('Recommendations for {0}:\n'.format(movie_features_df.index[query_index]))
    else:
        print('{0}: {1}, with distance of {2}:'.format(i, movie_features_df.index[indices.flatten()[i]], distances.flatten()[i]))
```

```
Recommendations for 2001: A Space Odyssey (1968):

1: Blade Runner (1982), with distance of 0.32926440238952637:
2: Alien (1979), with distance of 0.43005305528640747:
3: Apocalypse Now (1979), with distance of 0.4308894872665405:
4: Aliens (1986), with distance of 0.4363347887992859:
5: Clockwork Orange, A (1971), with distance of 0.43840235471725464:
```

In [ ]: