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

In [2]:
```
movies_df=pd.read_csv("movies.csv", usecols=['movieId','title'], dtype={'movieId': 'int32', 'title': 'str'})
rating_df=pd.read_csv("ratings.csv", usecols=['userId', 'movieId', 'rating'], dtype={'userID':'int32', 'movieId':'int32', 'rating':'float32'})
```

In [3]:
```
movies_df
```

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)
...	...	...
9737	193581	Black Butler: Book of the Atlantic (2017)
9738	193583	No Game No Life: Zero (2017)
9739	193585	Flint (2017)
9740	193587	Bungo Stray Dogs: Dead Apple (2018)
9741	193609	Andrew Dice Clay: Dice Rules (1991)

9742 rows × 2 columns

In [4]:
```
rating_df
```

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
...	...	...	...
100831	610	166534	4.0
100832	610	168248	5.0
100833	610	168250	5.0
100834	610	168252	5.0
100835	610	170875	3.0

100836 rows × 3 columns

In [5]:
```
df=pd.merge(movies_df,rating_df,on='movieId')
```

In [6]:
```
df
```

Out[6]:

	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 [7]:
```
# combine_movie_rating=df.dropna(how='any', axis=0)
combine_movie_rating = df.dropna(axis = 0, subset = ['title'])
movieratingcount=pd.DataFrame(combine_movie_rating.groupby('title')['rating'].count())
```

In [8]:
```
combine_movie_rating.shape
```

Out[8]:
```
(100836, 4)
```

In [9]:
```
movieratingcount
```

Out[9]:

	rating
title	
'71 (2014)	1
'Hellboy': The Seeds of Creation (2004)	1
'Round Midnight (1986)	2
'Salem's Lot (2004)	1
'Til There Was You (1997)	2
...	...
eXistenZ (1999)	22
xXx (2002)	24
xXx: State of the Union (2005)	5
¡Three Amigos! (1986)	26
À nous la liberté (Freedom for Us) (1931)	1

9719 rows × 1 columns

In [10]:
```
rating_with_totalRatingCount =pd.merge(combine_movie_rating, movieratingcount, on='title')
```

In [11]:
```
rating_with_totalRatingCount.rename(columns={'rating_y':'totalRatingCount', 'rating_x':'rating'}, inplace= True)
```

In [12]:
```
rating_with_totalRatingCount.head()
```

Out[12]:

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

In [13]:
```
rating_with_totalRatingCount.shape
```

Out[13]:
```
(100836, 5)
```

In [14]:
```
popularity_threshold=50
rating_popular_movie=rating_with_totalRatingCount.query('totalRatingCount >= @popularity_threshold')
rating_popular_movie.head()
```

Out[14]:

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

In [15]:
```
movie_features_df=rating_popular_movie.pivot_table(index='title', columns='userId', values='rating').fillna(0)
movie_features_df.head()
```

Out[15]:

userId	1	2	3	4	5	6	7	8	9	10	...	601	602	603	604	605	606	607	608	609	610
title																					
10 Things I Hate About You (1999)	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	5.0	0.0	0.0	0.0	0.0	0.0
12 Angry Men (1957)	0.0	0.0	0.0	5.0	0.0	0.0	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
2001: A Space Odyssey (1968)	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	...	0.0	0.0	5.0	0.0	0.0	5.0	0.0	3.0	0.0	4.5
28 Days Later (2002)	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	3.5	0.0	5.0
300 (2007)	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	3.0	0.0	0.0	5.0	0.0	4.0

5 rows × 606 columns

In [16]:
```
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[16]:
```
NearestNeighbors(algorithm='brute', metric='cosine')
```

In [17]:
```
movie_features_df.shape
```

Out[17]:
```
(450, 606)
```

In [18]:
```
query_index=np.random.choice(movie_features_df.shape[0])
print(query_index)
distances, indices= model_knn.kneighbors(movie_features_df.iloc[query_index,:].values.reshape(1, -1), n_neighbors = 6)
```

```
273
```

In [19]:
```
movie_features_df.head()
```

Out[19]:

userId	1	2	3	4	5	6	7	8	9	10	...	601	602	603	604	605	606	607	608	609	610
title																					
10 Things I Hate About You (1999)	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	5.0	0.0	0.0	0.0	0.0	0.0
12 Angry Men (1957)	0.0	0.0	0.0	5.0	0.0	0.0	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
2001: A Space Odyssey (1968)	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	...	0.0	0.0	5.0	0.0	0.0	5.0	0.0	3.0	0.0	4.5
28 Days Later (2002)	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	3.5	0.0	5.0
300 (2007)	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	3.0	0.0	0.0	5.0	0.0	4.0

5 rows × 606 columns

In [20]:
```
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 Mission: Impossible (1996):

1: Independence Day (a.k.a. ID4) (1996), with distance of 0.32264602184295654:
2: Jurassic Park (1993), with distance of 0.407723605632782:
3: Twister (1996), with distance of 0.41159355640411377:
4: Rock, The (1996), with distance of 0.43483167886673401:
5: GoldenEye (1995), with distance of 0.44934725761413574:
```