In [33]:
```python
#importing libraries
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
from scipy.sparse import csr_matrix
```

In [34]:
```python
#load rating into a dataframe
df_ratings = pd.read_csv('ratings.csv')
df_ratings.head()
```

Out[34]:

	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 [35]:
```python
#load movies into a dataframe
df_movies = pd.read_csv('movies.csv')
df_movies.head()
```

Out[35]:

	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 [36]:
```python
# pivot ratings dataframe
movie_ratings = df_ratings.pivot(index='userId', columns='movieId', values='rating')
movie_ratings.head()
```

Out[36]:

movieId	1	2	3	4	5	6	7	8	9	10	...	193565	193567	193571	193573	193579	193581	193583	193585	193587	193609
userId																					
1	4.0	NaN	4.0	NaN	NaN	4.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 9724 columns

In [37]:
```python
# fill nan values with zero
movie_ratings = movie_ratings.fillna(0)
movie_ratings.head()
```

Out[37]:

movieId	1	2	3	4	5	6	7	8	9	10	...	193565	193567	193571	193573	193579	193581	193583	193585	193587	193609
userId																					
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	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	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	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	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	0.0	0.0	0.0	0.0

5 rows × 9724 columns

In [38]:
```python
# Convert to matrix
ratings_matrix = csr_matrix(movie_ratings.values)
ratings_matrix
```

Out[38]:
```
<610x9724 sparse matrix of type '<class 'numpy.float64'>'
	with 100836 stored elements in Compressed Sparse Row format>
```

In [39]:
```
# cosine similarity
cosine = cosine_similarity(ratings_matrix.T)
cosine
```

Out[39]:
```
array([[1.        , 0.41056206, 0.2969169 , ..., 0.        , 0.        ,
        0.        ],
       [0.41056206, 1.        , 0.28243799, ..., 0.        , 0.        ,
        0.        ],
       [0.2969169 , 0.28243799, 1.        , ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.        , 0.        , 0.        , ..., 1.        , 1.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 1.        , 1.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        1.        ]])
```

In [53]:
```
#function to recommend a movie

def recommend_movies(movie):
    # movie index
    movie_index = df_movies[df_movies['title'] == movie].index[0]
    similarity_scores = list(enumerate(cosine[movie_index]))
    similarity_scores = sorted(similarity_scores, key=lambda x: x[1], reverse=True)

    #returning top ten movies
    similarity_scores = similarity_scores[1:11]
    movie_indices = [i[0] for i in similarity_scores]
    movie_list = []
    movie_titles = df_movies['title'].iloc[movie_indices]
    movie_list.append(movie_titles)

    #could not figure out how to add scores right next to the movie. Tried to return a df and would get errors.
    return movie_list, similarity_scores
```

In [60]:
```
# running the function
movie_choice = 'Jumanji (1995)'
recommended = recommend_movies(movie_choice)
print(f'Top ten movie recommendations for:', movie_choice)
print(*recommended)
```

```
Top ten movie recommendations for: Jumanji (1995)
[322                     Lion King, The (1994)
436                    Mrs. Doubtfire (1993)
325                         Mask, The (1994)
418                     Jurassic Park (1993)
504                        Home Alone (1990)
483    Nightmare Before Christmas, The (1993)
506                           Aladdin (1992)
512              Beauty and the Beast (1991)
18      Ace Ventura: When Nature Calls (1995)
276                    Santa Clause, The (1994)
Name: title, dtype: object] [(322, 0.5884377258584126), (436, 0.5498181061555002), (325, 0.5449810767978693), (418, 0.538045566
9772967), (504, 0.524876420608931), (483, 0.5181613195590729), (506, 0.515619976850775), (512, 0.507457989132598), (18, 0.49756
026413689786), (276, 0.4973675079070731)]
```

Using the small MovieLens data set, create a recommender system that allows users to input a movie they like (in the data set) and recommends ten other movies for them to watch. In your write-up, clearly explain the recommender system process and all steps performed. If you are using a method found online, be sure to reference the source.
You can use R or Python to complete this assignment. Submit your code and output to the submission link.

the recommender system that I used was cosine similarity which is used to find the consine distances between all of the data points. a cosine similarity of 1 means that they're exactly alike while 0 means that they're nothing alike. afer adding both csv files to separate dataframes, I pivoted the ratings dataframe and turned it into a matrix. Then I found the cosine similarity for that dataframe. Lastly, I created a function that would find the top ten similar movies based on the consine similarity and then print them out. I'm not sure if I agree with this recommender system. I personally love Jumangi but do not enjoy several of the other movies on the top ten list. Maybe a different recommender system would be more accurate for movies. Though, this does remind of the outlandish movie recommendations that netflix gives me so this might be what they use, or something similar to it.