# Recommendation

In [21]:
```
import pandas as pd
from scipy.sparse import csr_matrix
from sklearn.neighbors import NearestNeighbors
from fuzzywuzzy import process
```

In [5]:
```
movies = pd.read_csv("./movies.csv", usecols=['movieId', 'title']);
movies.head()
```

Out[5]:

| | 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 [6]:
```
ratings = pd.read_csv("./ratings.csv", usecols=['userId', 'movieId', 'rating']);
ratings.head()
```

Out[6]:

| | 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 [7]:
```
movies.shape
```

Out[7]:
```
(9742, 2)
```

In [9]:
```
ratings.shape
```

Out[9]:
```
(100836, 3)
```

## Create movies_users matrix

In [11]:
```
ratings.pivot(index='movieId', columns='userId', values='rating')
```

Out[11]:

| userId | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... | 601 | 602 | 603 | 604 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **movieId** | | | | | | | | | | | | | | | |
| **1** | 4.0 | NaN | NaN | NaN | 4.0 | NaN | 4.5 | NaN | NaN | NaN | ... | 4.0 | NaN | 4.0 | 3.0 |
| **2** | NaN | NaN | NaN | NaN | NaN | 4.0 | NaN | 4.0 | NaN | NaN | ... | NaN | 4.0 | NaN | 5.0 |
| **3** | 4.0 | NaN | NaN | NaN | NaN | 5.0 | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **4** | NaN | NaN | NaN | NaN | NaN | 3.0 | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **5** | NaN | NaN | NaN | NaN | NaN | 5.0 | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | 3.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **193581** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **193583** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **193585** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **193587** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |
| **193609** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN |

9724 rows × 610 columns

In [14]:
```
movies_users = ratings.pivot(index='movieId', columns='userId', values='rating').fi
movies_users
```

Out[14]:

| userId | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... | 601 | 602 | 603 | 604 | 605 | 606 | 607 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **movieId** | | | | | | | | | | | | | | | | | | |
| **1** | 4.0 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 4.5 | 0.0 | 0.0 | 0.0 | ... | 4.0 | 0.0 | 4.0 | 3.0 | 4.0 | 2.5 | 4.0 |
| **2** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 4.0 | 0.0 | 0.0 | ... | 0.0 | 4.0 | 0.0 | 5.0 | 3.5 | 0.0 | 0.0 |
| **3** | 4.0 | 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 | 0.0 |
| **4** | 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 |
| **5** | 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 | 3.0 | 0.0 | 0.0 | 0.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **193581** | 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 |
| **193583** | 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 |
| **193585** | 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 |
| **193587** | 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 |
| **193609** | 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 |

9724 rows × 610 columns

In [16]:
```
mat_movies = csr_matrix(movies_users.values)
mat_movies
```

Out[16]:
```
<9724x610 sparse matrix of type '<class 'numpy.float64'>'
        with 100836 stored elements in Compressed Sparse Row format>
```

## Create Model

In [22]:
```
model = NearestNeighbors(metric='cosine', algorithm='brute', n_neighbors=20)
model.fit(mat_movies)
```

Out[22]:
```
NearestNeighbors(algorithm='brute', metric='cosine', n_neighbors=20)
```

## Item based recommandation

In [32]:
```
def recommender(movie_name, data, n):
    idx = process.extractOne(movie_name, movies['title'])[2]
    print('Movie Selected : ', movies['title'][idx], 'Index : ', idx)
    print("Searching for recommandation.............")
    distance, indices = model.kneighbors(data[idx], n_neighbors=n)
    # print(distance, indices)
    for i in indices:
        print(movies['title'][i].where(i!=idx))
```

In [34]:
```
recommender('iron man', mat_movies, 10)
```

```
Movie Selected :  Iron Man (2008) Index :  6743
Searching for recommandation.............
6743                                          NaN
7197                                Garage (2007)
7195                       Merry Madagascar (2009)
7354                            A-Team, The (2010)
6726                         Superhero Movie (2008)
7137                         Thirst (Bakjwi) (2009)
7026                                 Scorpio (1973)
7571                                 Win Win (2011)
3880                  Look Who's Talking Now (1993)
6388    After the Wedding (Efter brylluppet) (2006)
Name: title, dtype: object
```