```
from google.colab import files
upload=files.upload()
```

Choose Files movies.csv

- **movies.csv**(application/vnd.ms-excel) - 494431 bytes, last modified: 12/21/2021 - 100% done

```
Saving movies.csv to movies (1).csv
```

```
import pandas as pd
```

```
movies = pd.read_csv('movies.csv',usecols=['movieId','title'])
movies.head()
```

| | 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) |

```
from google.colab import files
upload=files.upload()
```

Choose Files ratings.csv

- **ratings.csv**(application/vnd.ms-excel) - 2483723 bytes, last modified: 12/21/2021 - 100% done

Saving...

```
ratings = pd.read_csv('ratings.csv',usecols=['userId','movieId','rating'])
ratings.head()
```

| | 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 |

```
movies.shape
```

```
(9742, 2)
```

```
ratings.shape
```

```
(100836, 3)
```

```
movies_users=ratings.pivot(index='movieId',columns='userId',values='rating').fillna(0)
movies_users.head()
```

| userId | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| movieId | | | | | | | | | | | | | | | | | |
| 1 | 4.0 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 4.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.5 | 0.0 | 4. |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 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. |
| 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 | 3.0 | 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 | 0.0 | 0.0 | 0.0 | 0. |

5 rows × 610 columns

```
from scipy.sparse import csr_matrix

mat_movies=csr_matrix(movies_users.values)
```

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



NearestNeighbors

model=NearestNeighbors(metric='cosine',algorithm='brute',n_neighbors=20)

model.fit(mat_movies)

```
from sklearn.neighbors import NearestNeighbors
model=NearestNeighbors(metric='cosine',algorithm='brute',n_neighbors=20)
model.fit(mat_movies)
```

```
NearestNeighbors(algorithm='brute', metric='cosine', n_neighbors=20)
```

```
!pip install fuzzywuzzy
```

```
Collecting fuzzywuzzy
  Downloading fuzzywuzzy-0.18.0-py2.py3-none-any.whl (18 kB)
Installing collected packages: fuzzywuzzy
Successfully installed fuzzywuzzy-0.18.0
```

```
from fuzzywuzzy import process
```

```
/usr/local/lib/python3.7/dist-packages/fuzzywuzzy/fuzz.py:11: UserWarning: Using slow
  warnings.warn('Using slow pure-python SequenceMatcher. Install python-Levenshtein t
```

```
def recommender(movie_name,data,n):
   idx= process.extractOne(movie_name, movies['title'])[2]
   print("Movies Selected:", movies[ 'title' ][idx], 'Index: ',idx)
   print('Searching for recommondation....')
   distance, indices= model.kneighbors (data[idx],n_neighbors=n)

   for i in indices :
    print (movies['title'][i].where(i!=idx))
```

```
recommender('toy story',mat_movies,10)
```

```
Movies Selected: Toy Story (1995) Index:  0
Searching for recommondation....
0                                                    NaN
2353                                 'night Mother (1986)
418                                  Jurassic Park (1993)
615                  Independence Day (a.k.a. ID4) (1996)
224             Star Wars: Episode IV - A New Hope (1977)
314                                   Forrest Gump (1994)
322                                 Lion King, The (1994)
910     Once Upon a Time in the West (C'era una volta ...
546                            Mission: Impossible (1996)
                                              Diva (1981)
```

Saving...

✓ 5s completed at 10:05 PM

Saving...