1. Import Necessasary Libraries

In [9]:

```
import pandas as pd
```

2. Import Data

In [8]:

```
movies_data = pd.read_csv('Movie.csv')
movies_data.head()
```

Out[8]:

	userId	movie	rating
0	3	Toy Story (1995)	4.0
1	6	Toy Story (1995)	5.0
2	8	Toy Story (1995)	4.0
3	10	Toy Story (1995)	4.0
4	11	Toy Story (1995)	4.5

3. Data Understanding

In [3]:

```
movies_data.shape
```

Out[3]:

```
(8992, 3)
```

In [4]:

```
movies_data.userId.nunique()
```

Out[4]:

```
4081
```

In [10]:

```
movies_data.movie.nunique()
```

Out[10]:

```
10
```

In [14]:

```
print(movies_data.movie.unique())
```

```
['Toy Story (1995)' 'Jumanji (1995)' 'Grumpier Old Men (1995)'
 'Waiting to Exhale (1995)' 'Father of the Bride Part II (1995)'
 'Heat (1995)' 'Sabrina (1995)' 'Tom and Huck (1995)'
 'Sudden Death (1995)' 'GoldenEye (1995)']
```

4. Data preparation

In [17]:

```
pd.pivot_table(data = movies_data,values='rating', index='userId',columns='movie')
```

Out[17]:

movie	Father of the Bride Part II (1995)	GoldenEye (1995)	Grumpier Old Men (1995)	Heat (1995)	Jumanji (1995)	Sabrina (1995)	Sudden Death (1995)	Tom and Huck (1995)	Toy Story (1995)	Waiting to Exhale (1995
userId										
1	NaN	NaN	NaN	NaN	3.5	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	NaN
4	NaN	4.0	NaN	3.0	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	3.0	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
7115	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7116	3.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	NaN
7117	NaN	3.0	4.0	5.0	NaN	3.0	1.0	NaN	4.0	NaN
7119	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	NaN
7120	NaN	NaN	NaN	NaN	4.0	4.0	NaN	NaN	4.5	NaN

4081 rows × 10 columns

In [20]:

```
movie_recomm = pd.pivot_table(data = movies_data,values='rating', index='userId',columns='m
movie_recomm
```

Out[20]:

movie	Father of the Bride Part II (1995)	GoldenEye (1995)	Grumpier Old Men (1995)	Heat (1995)	Jumanji (1995)	Sabrina (1995)	Sudden Death (1995)	Tom and Huck (1995)	Toy Story (1995)	Waiting to Exhale (1995
userId										
1	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	4.0	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	4.0	0.0
4	0.0	4.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	..
7115	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7116	3.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
7117	0.0	3.0	4.0	5.0	0.0	3.0	1.0	0.0	4.0	0.0
7119	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0
7120	0.0	0.0	0.0	0.0	4.0	4.0	0.0	0.0	4.5	0.0

4081 rows × 10 columns

In [22]:

```
movie_recomm.head(40)
```

Out[22]:

movie	Father of the Bride Part II (1995)	GoldenEye (1995)	Grumpier Old Men (1995)	Heat (1995)	Jumanji (1995)	Sabrina (1995)	Sudden Death (1995)	Tom and Huck (1995)	Toy Story (1995)	Waiting to Exhale (1995
userId										
1	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	4.0	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	4.0	0.0
4	0.0	4.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	3.0	0.0	0.0	5.0	0.0	0.0	5.0	0.0
7	0.0	0.0	3.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0
8	0.0	4.0	5.0	3.0	0.0	0.0	0.0	0.0	4.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
11	0.0	2.5	0.0	0.0	0.0	0.0	0.0	0.0	4.5	0.0
12	2.0	0.0	3.0	3.0	0.0	3.0	0.0	0.0	4.0	0.0
13	0.0	3.0	0.0	0.0	3.0	0.0	0.0	0.0	4.0	0.0
14	0.0	0.0	0.0	0.0	0.0	3.5	0.0	0.0	4.5	0.0
15	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	3.0	0.0
17	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	4.0	5.0	0.0	5.0	0.0	0.0	5.0	0.0
21	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	3.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
24	2.0	3.0	0.0	4.0	0.0	3.0	0.0	0.0	4.0	0.0
26	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0	3.0	0.0	4.0	0.0	0.0	0.0
30	2.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0
32	0.0	3.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
34	0.0	5.0	0.0	0.0	3.0	3.0	0.0	0.0	5.0	0.0
38	0.0	0.0	3.0	0.0	0.0	5.0	0.0	3.0	0.0	0.0
39	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	5.0	0.0
41	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
46	3.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0

movie	Father of the Bride Part II (1995)	GoldenEye (1995)	Grumpier Old Men (1995)	Heat (1995)	Jumanji (1995)	Sabrina (1995)	Sudden Death (1995)	Tom and Huck (1995)	Toy Story (1995)	Waiting to Exhale (1995
userId										
47	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
48	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
53	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
54	3.0	4.0	0.0	3.0	3.0	0.0	0.0	0.0	4.0	0.0
58	0.0	0.0	0.0	4.5	0.0	0.0	0.0	0.0	5.0	0.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.5	0.0
61	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
66	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
67	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

In [24]:

```
movie_recomm.reset_index()
```

Out[24]:

movie	userId	Father of the Bride Part II (1995)	GoldenEye (1995)	Grumpier Old Men (1995)	Heat (1995)	Jumanji (1995)	Sabrina (1995)	Sudden Death (1995)	Tom and Huck (1995)	Toy Story (1995)
0	1	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0
1	2	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
2	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0
3	4	0.0	4.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
4	5	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...
4076	7115	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4077	7116	3.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0
4078	7117	0.0	3.0	4.0	5.0	0.0	3.0	1.0	0.0	4.0
4079	7119	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0
4080	7120	0.0	0.0	0.0	0.0	4.0	4.0	0.0	0.0	4.5

4081 rows × 11 columns

In [26]:

```
movie_recomm = movie_recomm.reset_index( drop = True)
movie_recomm
```

Out[26]:

movie	Father of the Bride Part II (1995)	GoldenEye (1995)	Grumpier Old Men (1995)	Heat (1995)	Jumanji (1995)	Sabrina (1995)	Sudden Death (1995)	Tom and Huck (1995)	Toy Story (1995)	Waiting to Exhale (1995)
0	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	4.0	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	4.0	0.0
3	0.0	4.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...
4076	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4077	3.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
4078	0.0	3.0	4.0	5.0	0.0	3.0	1.0	0.0	4.0	0.0
4079	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0
4080	0.0	0.0	0.0	0.0	4.0	4.0	0.0	0.0	4.5	0.0

4081 rows × 10 columns

In [29]:

```
movie_recomm.index = movies_data['userId'].unique()
```

In [32]:

```
movie_recomm
```

Out[32]:

movie	Father of the Bride Part II (1995)	GoldenEye (1995)	Grumpier Old Men (1995)	Heat (1995)	Jumanji (1995)	Sabrina (1995)	Sudden Death (1995)	Tom and Huck (1995)	Toy Story (1995)	Waiting to Exhale (1995)
3	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
10	0.0	4.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...
7044	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7070	3.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
7080	0.0	3.0	4.0	5.0	0.0	3.0	1.0	0.0	4.0	0.0
7087	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0
7105	0.0	0.0	0.0	0.0	4.0	4.0	0.0	0.0	4.5	0.0

4081 rows × 10 columns

In [33]:

```
movie_recomm.shape
```

Out[33]:

```
(4081, 10)
```

Model Building and Finding out the similar user for recommendations

1.Using Correlation Matrix for UBCF

In [38]:

```
movie_transpose = movie_recomm.T
movie_transpose
```

Out[38]:

movie	3	6	8	10	11	12	13	14	16	19	...	6975	6979	6993	7030	7031	7(
Father of the Bride Part II (1995)	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	
GoldenEye (1995)	0.0	0.0	0.0	4.0	0.0	0.0	0.0	4.0	0.0	2.5	...	2.0	0.0	0.0	0.0	3.0	
Grumpier Old Men (1995)	0.0	4.0	0.0	0.0	0.0	3.0	3.0	5.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
Heat (1995)	0.0	0.0	0.0	3.0	0.0	0.0	0.0	3.0	0.0	0.0	...	0.0	0.0	0.0	0.0	3.0	
Jumanji (1995)	3.5	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	2.0	4.0	0.0	
Sabrina (1995)	0.0	0.0	0.0	0.0	0.0	5.0	3.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
Sudden Death (1995)	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	
Tom and Huck (1995)	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	
Toy Story (1995)	0.0	0.0	4.0	0.0	0.0	5.0	0.0	4.0	4.0	4.5	...	0.0	4.0	0.0	4.0	0.0	
Waiting to Exhale (1995)	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	

10 rows × 4081 columns

In [40]:

```
movie_corr = movie_transpose.corr().round(2)
movie_corr
```

Out[40]:

	3	6	8	10	11	12	13	14	16	19	...	6975	6979	6993	7(
3	1.00	-0.11	-0.11	-0.16	1.00	-0.21	-0.17	-0.27	-0.11	-0.16	...	-0.11	-0.11	1.00	0
6	-0.11	1.00	-0.11	-0.16	-0.11	0.28	0.67	0.56	-0.11	-0.16	...	-0.11	-0.11	-0.11	-0
8	-0.11	-0.11	1.00	-0.16	-0.11	0.60	-0.17	0.40	1.00	0.86	...	-0.11	1.00	-0.11	0
10	-0.16	-0.16	-0.16	1.00	-0.16	-0.31	-0.25	0.48	-0.16	0.24	...	0.78	-0.16	-0.16	-0
11	1.00	-0.11	-0.11	-0.16	1.00	-0.21	-0.17	-0.27	-0.11	-0.16	...	-0.11	-0.11	1.00	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7044	-0.11	-0.11	-0.11	-0.16	-0.11	-0.21	-0.17	-0.27	-0.11	-0.16	...	-0.11	-0.11	-0.11	-0
7070	-0.17	-0.17	0.72	-0.25	-0.17	0.33	-0.25	0.13	0.72	0.58	...	-0.17	0.72	-0.17	0
7080	-0.35	0.35	0.35	0.48	-0.35	0.54	0.40	0.81	0.35	0.41	...	0.18	0.35	-0.35	0
7087	-0.11	-0.11	1.00	-0.16	-0.11	0.60	-0.17	0.40	1.00	0.86	...	-0.11	1.00	-0.11	0
7105	0.48	-0.22	0.57	-0.32	0.48	0.67	0.20	-0.05	0.57	0.41	...	-0.22	0.57	0.48	0

4081 rows × 4081 columns

In [41]:

```
#pd.set_option('max_columns',None)
```

2. Using pairwise distances

In [42]:

```
from sklearn.metrics import pairwise_distances
#from scipy.spatial.distance import correlation,cosine
```

In [53]:

```
movie_recomm
```

Out[53]:

movie	Father of the Bride Part II (1995)	GoldenEye (1995)	Grumpier Old Men (1995)	Heat (1995)	Jumanji (1995)	Sabrina (1995)	Sudden Death (1995)	Tom and Huck (1995)	Toy Story (1995)	Waiting to Exhale (1995)
3	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
10	0.0	4.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...
7044	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7070	3.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
7080	0.0	3.0	4.0	5.0	0.0	3.0	1.0	0.0	4.0	0.0
7087	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0
7105	0.0	0.0	0.0	0.0	4.0	4.0	0.0	0.0	4.5	0.0

4081 rows × 10 columns

In [44]:

```
movie_recomm.values
```

Out[44]:

```
array([[0. , 0. , 0. , ..., 0. , 0. , 0. ],
       [0. , 0. , 4. , ..., 0. , 0. , 0. ],
       [0. , 0. , 0. , ..., 0. , 4. , 0. ],
       ...,
       [0. , 3. , 4. , ..., 0. , 4. , 0. ],
       [0. , 0. , 0. , ..., 0. , 5. , 0. ],
       [0. , 0. , 0. , ..., 0. , 4.5, 0. ]])
```

In [46]:

```
user_to_user = pairwise_distances(X =movie_recomm.values, metric='euclidean')
user_to_user
```

Out[46]:

```
array([[0.        , 5.31507291, 5.31507291, ..., 9.39414711, 6.10327781,
        6.04152299],
       [5.31507291, 0.        , 5.65685425, ..., 7.74596669, 6.40312424,
        8.26135582],
       [5.31507291, 5.65685425, 0.        , ..., 7.74596669, 1.        ,
        5.67890835],
       ...,
       [9.39414711, 7.74596669, 7.74596669, ..., 0.        , 7.81024968,
        8.26135582],
       [6.10327781, 6.40312424, 1.        , ..., 7.81024968, 0.        ,
        5.67890835],
       [6.04152299, 8.26135582, 5.67890835, ..., 8.26135582, 5.67890835,
        0.        ]])
```

In [48]:

```
movie_recomm.index
```

Out[48]:

```
Int64Index([   3,    6,    8,   10,   11,   12,   13,   14,   16,   19,
            ...
            6975, 6979, 6993, 7030, 7031, 7044, 7070, 7080, 7087, 7105],
           dtype='int64', length=4081)
```

In [52]:

```
user_user_df = pd.DataFrame(data = user_to_user, columns = movie_recomm.index,index =movie_
user_user_df
```

Out[52]:

	3	6	8	10	11	12	13	14	16
3	0.000000	5.315073	5.315073	6.103278	0.500000	8.440972	5.500000	8.845903	5.315073
6	5.315073	0.000000	5.656854	6.403124	5.000000	7.141428	3.162278	6.480741	5.656854
8	5.315073	5.656854	0.000000	6.403124	5.000000	5.916080	5.830952	7.071068	0.000000
10	6.103278	6.403124	6.403124	0.000000	5.830952	9.165151	6.557439	6.403124	6.403124
11	0.500000	5.000000	5.000000	5.830952	0.000000	8.246211	5.196152	8.660254	5.000000
...	...	...	...	...	...	...	...	...	..
7044	5.315073	5.656854	5.656854	6.403124	5.000000	8.660254	5.830952	9.055385	5.656854
7070	6.363961	6.652067	3.500000	7.297260	6.103278	6.873864	6.800735	7.889867	3.500000
7080	9.394147	7.745967	7.745967	6.855655	9.219544	6.403124	7.211103	4.000000	7.745967
7087	6.103278	6.403124	1.000000	7.071068	5.830952	5.830952	6.557439	7.141428	1.000000
7105	6.041523	8.261356	5.678908	8.789198	6.103278	5.123475	6.800735	9.069179	5.678908

4081 rows × 4081 columns

In [61]:

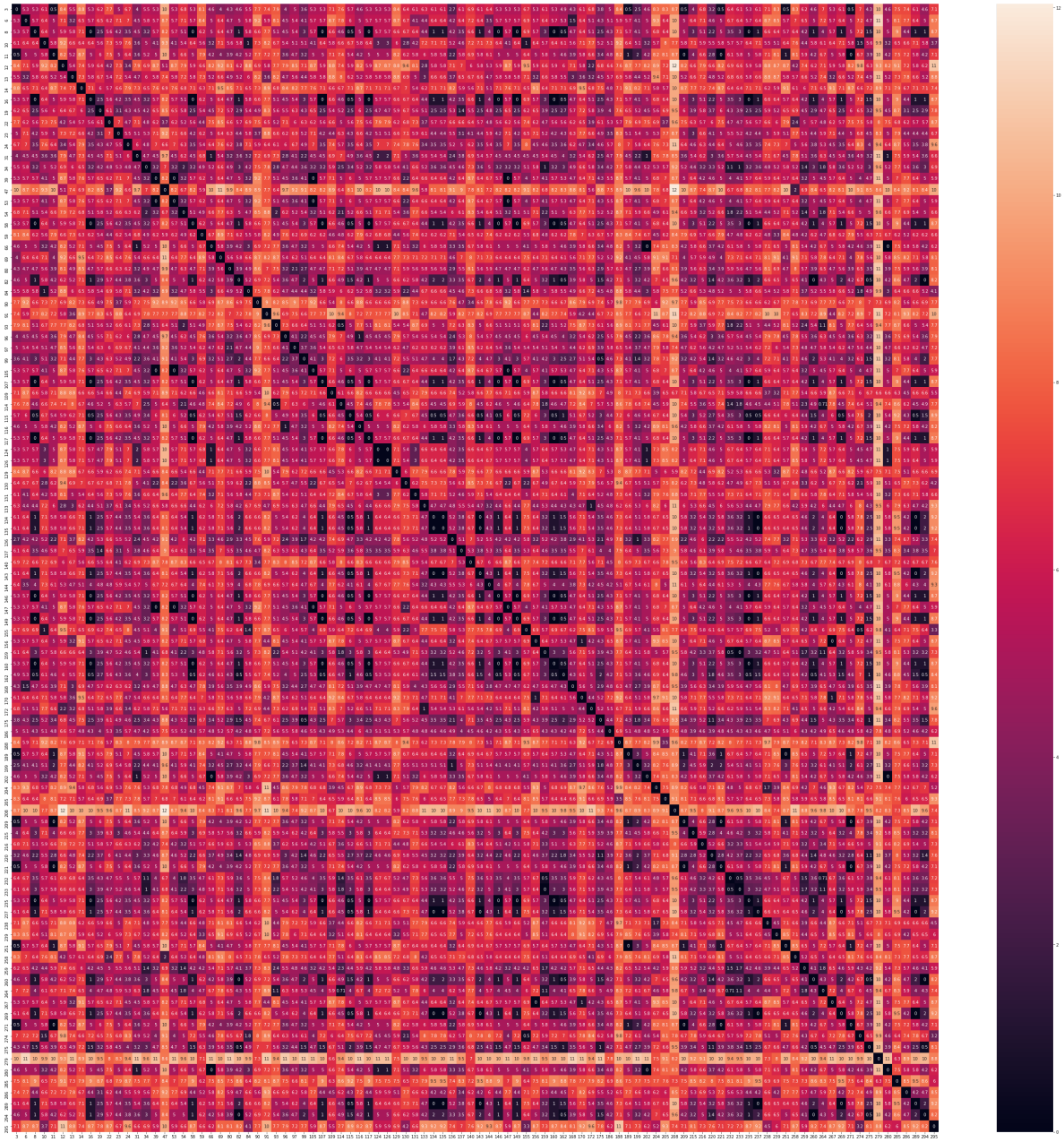
```
first_100_users = user_user_df.iloc[:100,:100]
first_100_users
```

Out[61]:

12	13	14	16	19	...	271	274	275	279	28
972	5.500000	8.845903	5.315073	6.224950	...	0.500000	6.964194	4.301163	10.307764	4.60977
428	3.162278	6.480741	5.656854	6.519202	...	5.000000	7.228416	4.716991	11.357817	5.00000
080	5.830952	7.071068	0.000000	2.549510	...	5.000000	7.228416	1.500000	10.246951	5.00000
151	6.557439	6.403124	6.403124	5.612486	...	5.830952	1.500000	5.590170	9.899495	3.16227
211	5.196152	8.660254	5.000000	5.958188	...	0.000000	6.726812	3.905125	10.392305	4.24264
...	...	...	...	...	...	...	...	...	...	
385	7.280110	7.937254	9.000000	8.746428	...	7.483315	6.422616	8.440972	6.324555	7.48331
103	7.810250	6.708204	4.358899	3.082207	...	7.211103	7.433034	4.924429	8.944272	5.83095
952	6.557439	7.141428	1.000000	2.549510	...	5.830952	7.826238	2.500000	10.198039	5.83095
414	5.196152	7.141428	1.000000	2.915476	...	4.242641	6.726812	0.500000	10.392305	4.24264
780	8.774964	7.416198	8.660254	7.778175	...	7.071068	3.201562	8.077747	8.831761	6.16441

In [64]:

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(50,50))
sns.heatmap(data = first_100_users, annot=True)
plt.show()
```



3. Lets use Cosine similarity for UBCF

In [67]:

```
user_user_cosine = pairwise_distances(X = movie_recomm.values,metric='cosine')
user_user_cosine
```

Out[67]:

```
array([[0.        , 1.        , 1.        , ..., 1.        , 1.        ,
        0.44662843],
       [1.        , 0.        , 1.        , ..., 0.54116853, 1.        ,
        1.        ],
       [1.        , 1.        , 0.        , ..., 0.54116853, 0.        ,
        0.37745698],
       ...,
       [1.        , 0.54116853, 0.54116853, ..., 0.        , 0.54116853,
        0.52392946],
       [1.        , 1.        , 0.        , ..., 0.54116853, 0.        ,
        0.37745698],
       [0.44662843, 1.        , 0.37745698, ..., 0.52392946, 0.37745698,
        0.        ]])
```

In [70]:

```
user_user_cf = pd.DataFrame(data=user_user_cosine,columns = movie_recomm.index,index=movie_
user_user_cf
```

Out[70]:

	3	6	8	10	11	12	13	14	16
3	0.000000	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000
6	1.000000	0.000000	1.000000	1.000000	1.000000	0.609433	0.292893	0.384543	1.000000
8	1.000000	1.000000	0.000000	1.000000	1.000000	0.349055	1.000000	0.507634	0.000000
10	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	0.384543	1.000000
11	0.000000	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000
...	...	...	...	...	...	...	...	...	..
7044	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
7070	1.000000	1.000000	0.247423	1.000000	1.000000	0.510114	1.000000	0.629457	0.247423
7080	1.000000	0.541169	0.541169	0.380578	1.000000	0.298116	0.432225	0.110468	0.541169
7087	1.000000	1.000000	0.000000	1.000000	1.000000	0.349055	1.000000	0.507634	0.000000
7105	0.446628	1.000000	0.377457	1.000000	0.446628	0.234545	0.608707	0.693481	0.377457

4081 rows × 4081 columns

In [110]:

```
import numpy as np
#np.fill_diagonal(a = user_user_cf, vol = 0)
np.fill_diagonal(user_user_cf.to_numpy(),0)
```

In [111]:

```
user_to_user_top_30 = user_user_cf.iloc[:30,:30]
user_to_user_top_30
```

Out[111]:

	3	6	8	10	11	12	13	14	16
3	0.000000	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000
6	1.000000	0.000000	1.000000	1.000000	1.000000	0.609433	0.292893	0.384543	1.000000
8	1.000000	1.000000	0.000000	1.000000	1.000000	0.349055	1.000000	0.507634	0.000000
10	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	0.384543	1.000000
11	0.000000	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000
12	1.000000	0.609433	0.349055	1.000000	1.000000	0.000000	0.263540	0.439120	0.349055
13	1.000000	0.292893	1.000000	1.000000	1.000000	0.263540	0.000000	0.564806	1.000000
14	1.000000	0.384543	0.507634	0.384543	1.000000	0.439120	0.564806	0.000000	0.507634
16	1.000000	1.000000	0.000000	1.000000	1.000000	0.349055	1.000000	0.507634	0.000000
19	1.000000	1.000000	0.125843	0.611486	1.000000	0.430972	1.000000	0.330481	0.125843
22	1.000000	0.562405	0.416540	0.737443	1.000000	0.164440	0.381147	0.281810	0.416540
23	0.485504	1.000000	0.314006	0.588403	0.485504	0.553456	1.000000	0.408920	0.314006
24	1.000000	1.000000	0.210648	1.000000	1.000000	0.086534	0.565878	0.611350	0.210648
31	1.000000	1.000000	1.000000	0.200000	1.000000	1.000000	1.000000	0.507634	1.000000
34	1.000000	1.000000	0.292893	0.575736	1.000000	0.539713	1.000000	0.390728	0.292893
39	1.000000	1.000000	1.000000	0.400000	1.000000	1.000000	1.000000	0.630726	1.000000
47	1.000000	0.580686	0.475858	0.685515	1.000000	0.153855	0.332876	0.290308	0.475858
53	1.000000	1.000000	1.000000	0.400000	1.000000	1.000000	1.000000	0.630726	1.000000
54	1.000000	1.000000	0.485504	0.485504	1.000000	0.665092	1.000000	0.430030	0.485504
58	1.000000	1.000000	0.000000	1.000000	1.000000	0.349055	1.000000	0.507634	0.000000
59	1.000000	1.000000	0.455669	0.346803	1.000000	0.379924	0.711325	0.329975	0.455669
66	1.000000	1.000000	1.000000	0.200000	1.000000	1.000000	1.000000	0.507634	1.000000
69	0.400000	1.000000	1.000000	1.000000	0.400000	1.000000	1.000000	1.000000	1.000000
80	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
82	1.000000	1.000000	0.000000	1.000000	1.000000	0.349055	1.000000	0.507634	0.000000
84	1.000000	1.000000	1.000000	0.010051	1.000000	1.000000	1.000000	0.390728	1.000000
90	0.636197	1.000000	0.393661	0.514929	0.636197	0.368491	0.742752	0.402919	0.393661
91	1.000000	0.542504	1.000000	1.000000	1.000000	0.324977	0.137338	0.718431	1.000000
93	1.000000	1.000000	0.292893	0.575736	1.000000	0.539713	1.000000	0.390728	0.292893
96	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

30 rows × 30 columns

In [116]:

```
user_to_user_top_30.idxmax()
```

Out[116]:

```
3     6
6     3
8     3
10    3
11    6
12    3
13    3
14    3
16    3
19    3
22    3
23    6
24    3
31    3
34    3
39    3
47    3
53    3
54    3
58    3
59    3
66    3
69    6
80    3
82    3
84    3
90    6
91    3
93    3
96    3
dtype: int64
```

In [108]:

```
movies_data
```

Out[108]:

	userId	movie	rating
0	3	Toy Story (1995)	4.0
1	6	Toy Story (1995)	5.0
2	8	Toy Story (1995)	4.0
3	10	Toy Story (1995)	4.0
4	11	Toy Story (1995)	4.5
...	...	...	...
8987	7087	GoldenEye (1995)	3.0
8988	7088	GoldenEye (1995)	1.0
8989	7105	GoldenEye (1995)	2.0
8990	7113	GoldenEye (1995)	3.0
8991	7117	GoldenEye (1995)	3.0

8992 rows × 3 columns

In [119]:

```
movies_data[(movies_data['userId']==23) | (movies_data['userId']==6)]
```

Out[119]:

	userId	movie	rating
1	6	Toy Story (1995)	5.0
11	23	Toy Story (1995)	4.0
3725	6	Grumpier Old Men (1995)	3.0
6464	6	Sabrina (1995)	5.0

In [120]:

```
movies_data[(movies_data['userId']==3) | (movies_data['userId']==6)]
```

Out[120]:

	userId	movie	rating
0	3	Toy Story (1995)	4.0
1	6	Toy Story (1995)	5.0
3725	6	Grumpier Old Men (1995)	3.0
6464	6	Sabrina (1995)	5.0

In [121]:

```
movies_data[(movies_data['userId']==90) | (movies_data['userId']==6)]
```

Out[121]:

	userId	movie	rating
1	6	Toy Story (1995)	5.0
26	90	Toy Story (1995)	3.5
3725	6	Grumpier Old Men (1995)	3.0
6464	6	Sabrina (1995)	5.0
7459	90	GoldenEye (1995)	3.5

In [122]:

```
movies_data[(movies_data['userId']==96) | (movies_data['userId']==3)]
```

Out[122]:

	userId	movie	rating
0	3	Toy Story (1995)	4.0
29	96	Toy Story (1995)	3.5
3734	96	Grumpier Old Men (1995)	4.0

Across 4081 users

In [123]:

```
user_user_cf.idxmax()
```

Out[123]:

```
3       6
6       3
8       3
10      3
11      6
       ..
7044    3
7070    3
7080    3
7087    3
7105    6
Length: 4081, dtype: int64
```

In [ ]:

```

```