Recommendation system

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
# Load dataset
movies_df=pd.read_csv("C:/Users/Ashraf/Documents/Datafiles/Movie.csv")
```

In [3]:

```
movies_df[0:5]
```

Out[3]:

	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

In [4]:

```
# number of unique user in the dataset
len(movies_df.userId.unique())
```

Out[4]:

```
4081
```

In [5]:

```
len(movies_df.movie.unique())
```

Out[5]:

```
10
```

In [6]:

```
user_movies_df = movies_df.pivot(index='userId',
                                 columns='movie',
                                 values='rating').reset_index(drop=True)
```

In [7]:

```
user_movies_df
```

Out[7]:

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	NaN	NaN	NaN	NaN	3.5	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	NaN
3	NaN	4.0	NaN	3.0	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	3.0	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
4076	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4077	3.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	NaN
4078	NaN	3.0	4.0	5.0	NaN	3.0	1.0	NaN	4.0	NaN
4079	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	NaN
4080	NaN	NaN	NaN	NaN	4.0	4.0	NaN	NaN	4.5	NaN

4081 rows × 10 columns

In [8]:

```
user_movies_df.index = movies_df.userId.unique()
```

In [9]:

```
user_movies_df
```

Out[9]:

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	NaN	NaN	NaN	NaN	3.5	NaN	NaN	NaN	NaN	NaN
6	NaN	NaN	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	NaN
10	NaN	4.0	NaN	3.0	NaN	NaN	NaN	NaN	NaN	NaN
11	NaN	NaN	NaN	NaN	3.0	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
7044	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7070	3.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	NaN
7080	NaN	3.0	4.0	5.0	NaN	3.0	1.0	NaN	4.0	NaN
7087	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	NaN
7105	NaN	NaN	NaN	NaN	4.0	4.0	NaN	NaN	4.5	NaN

4081 rows × 10 columns

In [10]:

```
#Impute those NaNs with 0 values
user_movies_df.fillna(0, inplace=True)
```

In [11]:

```
user_movies_df
```

Out[11]:

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 [12]:

```
#Calculating Cosine Similarity between Users
from sklearn.metrics import pairwise_distances
from scipy.spatial.distance import cosine, correlation
```

In [13]:

```
user_sim = 1 - pairwise_distances( user_movies_df.values,metric='cosine')
```

In [14]:

```
user_sim
```

Out[14]:

```
array([[1.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.55337157],
       [0.        , 1.        , 0.        , ..., 0.45883147, 0.        ,
        0.        ],
       [0.        , 0.        , 1.        , ..., 0.45883147, 1.        ,
        0.62254302],
       ...,
       [0.        , 0.45883147, 0.45883147, ..., 1.        , 0.45883147,
        0.47607054],
       [0.        , 0.        , 1.        , ..., 0.45883147, 1.        ,
        0.62254302],
       [0.55337157, 0.        , 0.62254302, ..., 0.47607054, 0.62254302,
        1.        ]])
```

In [15]:

```
#Store the results in a dataframe
user_sim_df = pd.DataFrame(user_sim)
```

In [16]:

```
#Set the index and column names to user ids
user_sim_df.index = movies_df.userId.unique()
user_sim_df.columns = movies_df.userId.unique()
```

In [17]:

```
user_sim_df.iloc[0:5, 0:5]
```

Out[17]:

	3	6	8	10	11
3	1.0	0.0	0.0	0.0	1.0
6	0.0	1.0	0.0	0.0	0.0
8	0.0	0.0	1.0	0.0	0.0
10	0.0	0.0	0.0	1.0	0.0
11	1.0	0.0	0.0	0.0	1.0

In [18]:

```
np.fill_diagonal(user_sim, 0)
user_sim_df.iloc[0:5, 0:5]
```

Out[18]:

	3	6	8	10	11
3	0.0	0.0	0.0	0.0	1.0
6	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0
11	1.0	0.0	0.0	0.0	0.0

In [19]:

```
#Most Similar Users
user_sim_df.idxmax(axis=1)[0:5]
```

Out[19]:

```
3       11
6      168
8       16
10    4047
11       3
dtype: int64
```

In [20]:

```
movies_df[(movies_df['userId']==6) | (movies_df['userId']==168)]
```

Out[20]:

	userId	movie	rating
1	6	Toy Story (1995)	5.0
60	168	Toy Story (1995)	4.5
3725	6	Grumpier Old Men (1995)	3.0
6464	6	Sabrina (1995)	5.0

In [21]:

```
user_1=movies_df[movies_df['userId']==6]
```

In [22]:

```
user_2=movies_df[movies_df['userId']==11]
```

In [23]:

```
user_2.movie
```

Out[23]:

```
4       Toy Story (1995)
7446    GoldenEye (1995)
Name: movie, dtype: object
```

In [24]:

```
user_1.movie
```

Out[24]:

```
1              Toy Story (1995)
3725    Grumpier Old Men (1995)
6464               Sabrina (1995)
Name: movie, dtype: object
```

In [25]:

```
pd.merge(user_1,user_2,on='movie',how='outer')
```

Out[25]:

	userId_x	movie	rating_x	userId_y	rating_y
0	6.0	Toy Story (1995)	5.0	11.0	4.5
1	6.0	Grumpier Old Men (1995)	3.0	NaN	NaN
2	6.0	Sabrina (1995)	5.0	NaN	NaN
3	NaN	GoldenEye (1995)	NaN	11.0	2.5

In [ ]:

```

```