In [9]:
```
## COMP 8745 : Spring 2021
## Project: Recommedation System for Movie Ratings
## Due Date: April 23, 2021
## Team:
##   1. Lokesh Chandra Das (U00740183)
##   2. Navid Mohammad Imran(U00761100)
## Implementation Algorithm: Item-Item Collaborative Filtering


# libraries
import numpy as np
import pandas as pd
import sklearn
from sklearn.metrics.pairwise import cosine_similarity
import scipy
```

In [10]:
```
# load dataset. This is an example dataset
columns = ['userID','movieID','Rating'] # We have added header to make it simple
user_movie_dataset = pd.read_csv('exampledataset/data_test.txt', delimiter=',',
user_movie_dataset #  This is basically displaying the following table
```

Out[10]:

	userID	movieID	Rating
0	1	1	2
1	1	3	3
2	2	1	5
3	2	2	2
4	3	1	3
5	3	2	3
6	3	3	1
7	4	2	2
8	4	3	2

In [11]:
```
# load actual dataset: Working Dataset
# User large dataset. rating.txt
columns = ['userID','movieID','Rating'] # Please do not remove this column.

user_movie_dataset = pd.read_csv('netflix/ratings.txt', delimiter=',', names=col
user_movie_dataset.head() # display first 5 entries of the dataset. We use it to
```

Out[11]:

	userID	movieID	Rating
0	28	1392773	4.0
1	28	1990901	5.0
2	28	765331	3.0
3	28	1987434	4.0
4	28	2193455	4.0

In [18]:
```
# Step 1: make user-movie matrix
```

```
user_movie_mat = user_movie_dataset.pivot(index='userID', columns='movieID', val
user_movie_mat.head()
# Note: This is creating a sparse matrix with the cell values 'NaN' if a particu
```

Out[18]:

movieID	7	79	199	481	769	906	1310	1333	1427	1442	...	2648572	2648589	26
userID														
28	4.0	NaN	NaN	NaN	NaN	3.0	3.0	2.0	NaN	4.0	...	NaN	3.0	
48	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
305	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	...	NaN	NaN	
577	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
595	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	

5 rows × 28968 columns

In [23]:

```
# Python API 'cosine_similarity' does not work on undefined value. To get rid of
user_movie_mat.fillna(0, inplace=True)
user_movie_mat.head()
```

Out[23]:

movieID	7	79	199	481	769	906	1310	1333	1427	1442	...	2648572	2648589	2648'
userID														
28	4.0	0.0	0.0	0.0	0.0	3.0	3.0	2.0	0.0	4.0	...	0.0	3.0	
48	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
305	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	...	0.0	0.0	
577	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
595	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 × 28968 columns

In [22]:

```
user_movie_mat.shape
```

Out[22]:

```
(92, 28968)
```

In [13]:

```
# Step 2: Item-item similarity matrix using cosine similarity
similarity_mat = cosine_similarity(user_movie_mat.T, user_movie_mat.T)
similarity_mat
```

Out[13]:

```
array([[1.        , 0.43503212, 0.37994855, ..., 0.42237094, 0.3741104 ,
        0.55577001],
       [0.43503212, 1.        , 0.72314299, ..., 0.0951968 , 0.61113629,
        0.65693576],
       [0.37994855, 0.72314299, 1.        , ..., 0.        , 0.78132929,
        0.61589504],
       ...,
       [0.42237094, 0.0951968 , 0.        , ..., 1.        , 0.        ,
        0.11553664],
       [0.3741104 , 0.61113629, 0.78132929, ..., 0.        , 1.        ,
        0.5317937 ],
```

```
       [0.55577001, 0.65693576, 0.61589504, ..., 0.11553664, 0.5317937 ,
        1.        ]])
```

In [30]:

```
# Step 3: predict rating
def predict_rating(user_id, movie_id):
    df = pd.DataFrame(user_movie_mat)

    user_index = np.where(df.index.values==user_id)[0][0]
    movie_index = np.where(df.columns.values==movie_id)[0][0]
    # We are calculating predicted rating of an item by a particular user using
    rating = np.sum(np.dot(user_movie_mat.iloc[user_index, :],similarity_mat[:,
    # bounded rating between 1 and 5
    rating = 1.0 if rating < 1 else rating;
    rating = 5.0 if rating > 5 else rating;
    rating = "{:.2f}".format(rating)
    print ("Predicted rating by user {0} for movie {1} is {2}".format(user_id, m
```

In [31]:

```
predict_rating(28,7) # first parameter is userID and second parameter is movieID
```

```
Predicted rating by user 28 for movie 7 is 2.08
```

In [16]:

```
## Finally done
```