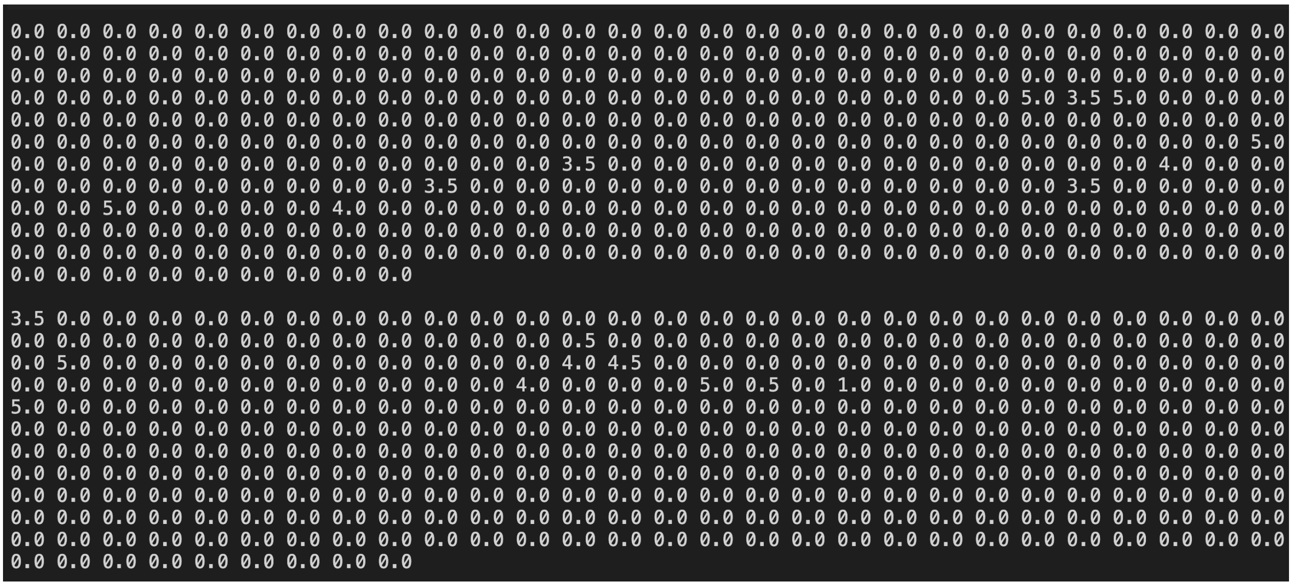
In this paper I have implemented a recommender system to recommend movies a user can watch using collaborative filtering. I used matrix factorisation to predict ratings for an unwatched movie and recommend the movies that are predicted to have the highest rating for that user.

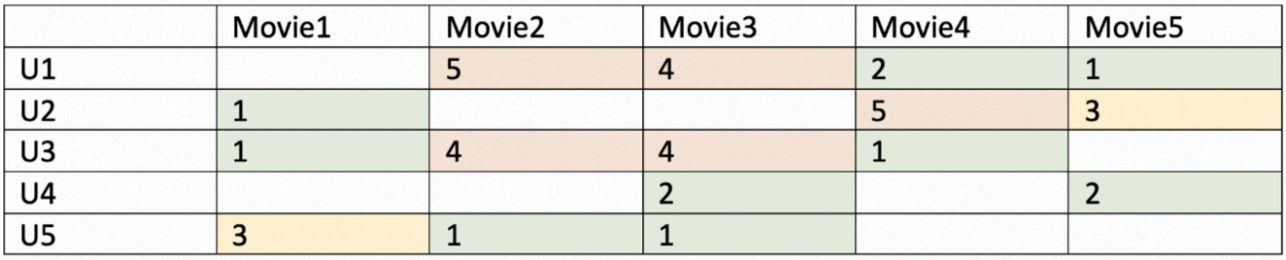
The procedure

- ***The DataSet -*** I have used a movies dat set from MovieLens. It has 25M movie ratings in total. There are a total of 62,000 movies rated by 162,000 users with each user rating between 5-10 movies. For my purpose and to make the training easier, I have taken ratings of a 100 users over 317 movies. Here are ratings of 2 users over all the 317 movies. The movies which user hasn't rated default to a 0 and as you can see, the matrix is very sparse.



```
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 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 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 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 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 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 0.0 0.0 0.0 0.0 5.0 3.5 5.0 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 0.0 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 0.0 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 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 5.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 3.5 0.0 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 0.0 0.0 0.0 0.0 0.0 0.0 0.0 3.5 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 3.5 0.0 0.0 0.0 0.0
0.0 0.0 5.0 0.0 0.0 0.0 0.0 4.0 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 0.0 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 0.0 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 0.0 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 0.0 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 0.0

3.5 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 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 0.0 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 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 4.0 4.5 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 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 4.0 0.0 0.0 0.0 5.0 0.5 0.0 1.0 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 0.0 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 0.0 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 0.0 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
0.0 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 0.0 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 0.0 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 0.0 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 0.0 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 0.0 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 0.0 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 0.0 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 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

- ***Rating Prediction -*** We have used collaborative filtering to predict the ratings of a user for an unrated movie. The predictions is performed based on the preference of other users who have rated similarly as our current user based on the common movies watched. We use matrix factorisation to perform this prediction.
- ***Matrix factorization -*** Matrix factorisation is a way to generate latent features when multiplying two different kinds of entities. We apply this to identify the relationship between users and items.With the input of user ratings over movies, we would like to recommend movies based on the predicted ratings for unrated or unwatched movies by the user.



	Movie1	Movie2	Movie3	Movie4	Movie5
U1		5	4	2	1
U2	1			5	3
U3	1	4	4	1	
U4			2		2
U5	3	1	1		

Two users give high ratings to a certain move when the movie is acted by their favourite actor and actress or the movie genre is an action one, etc. From the table above, we can find that the user1 and user3 both give high ratings to movie2 and movie3. By matrix factorisation, we are able to discover these latent features to predict a rating with similarity in user's preferences and interactions.

- ***The prediction*** - The ratings matrix *R* will have *n*×*m* rows and columns. The matrix *R* can be decomposed into two thin matrices *P* and *Q*. *P* will have *n*×*f* dimensions and *Q* will have *m*×*f* dimensions where *f* is the number of latent factors. The matrix *R* can be decomposed in such a way that the dot product of the matrix *P* and transposed *Q* will yield a matrix with *n*×*m* dimensions that closely approximates the original ratings matrix *R.*

The decomposition of the ratings matrix into the user matrix and the item matrix makes intuitive sense. The rating of a user-item combination, say Olivia to *Toy Story*, can be explained by Olivia's preference to comedy movies and whether the movie *Toy Story* is acclaimed highly in a comedic content scale or not. This generalization approach greatly reduces the size of the matrices.

$$\mathbf{R} \approx \mathbf{P} \times \mathbf{Q}^T = \hat{\mathbf{R}}$$

The rating of any user for an item can be found by the below dot product

$$\hat{r}_{ij} = p_i^T q_j = \sum_{k=1}^{k} p_{ik} q_{kj}$$

To find the matrices P and Q we minimise the following error. Here, K refers to the set of tuples with users and their actual ratings. We minimise the mean squared error w.r.t to the actual rating given by a user and the predicted rating for that item

$$\min \sum_{(u,i) \in K} (r_{ui} - \widehat{r_{ui}})^2$$

$$\min \sum_{(u,i) \in K} (r_{ui} - p_u q_i^T)^2$$

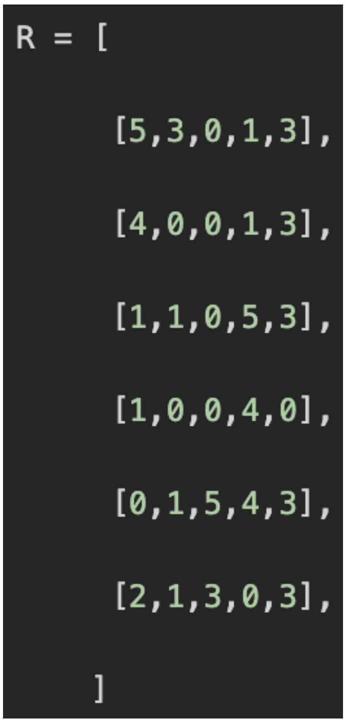
The Error is minimised using gradient descent. We find the gradients w.r.t parameters P and Q and update them.

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(q_{kj}) = -2e_{ij}q_{kj}$$

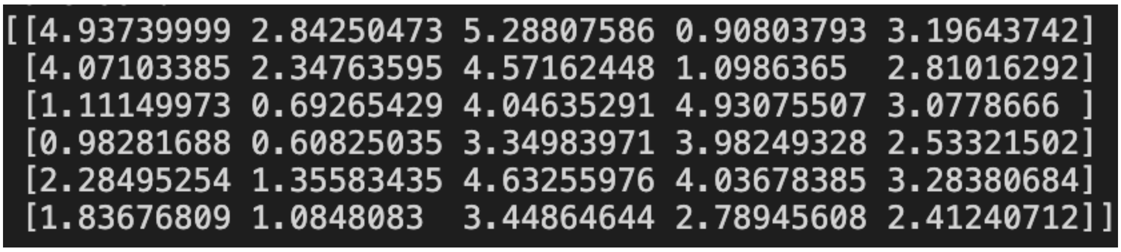
$$\frac{\partial}{\partial q_{ik}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(p_{ik}) = -2e_{ij}p_{ik}$$

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + 2\alpha e_{ij} q_{kj}$$
$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + 2\alpha e_{ij} p_{ik}$$

- ***An example*** **-** Consider the following ratings matrix with 6 users who have rated 5 movies. Upon performing matrix factorisation and printing the predicted ratings matrix, we get



```
R = [

     [5,3,0,1,3],

     [4,0,0,1,3],

     [1,1,0,5,3],

     [1,0,0,4,0],

     [0,1,5,4,3],

     [2,1,3,0,3],

    ]
```

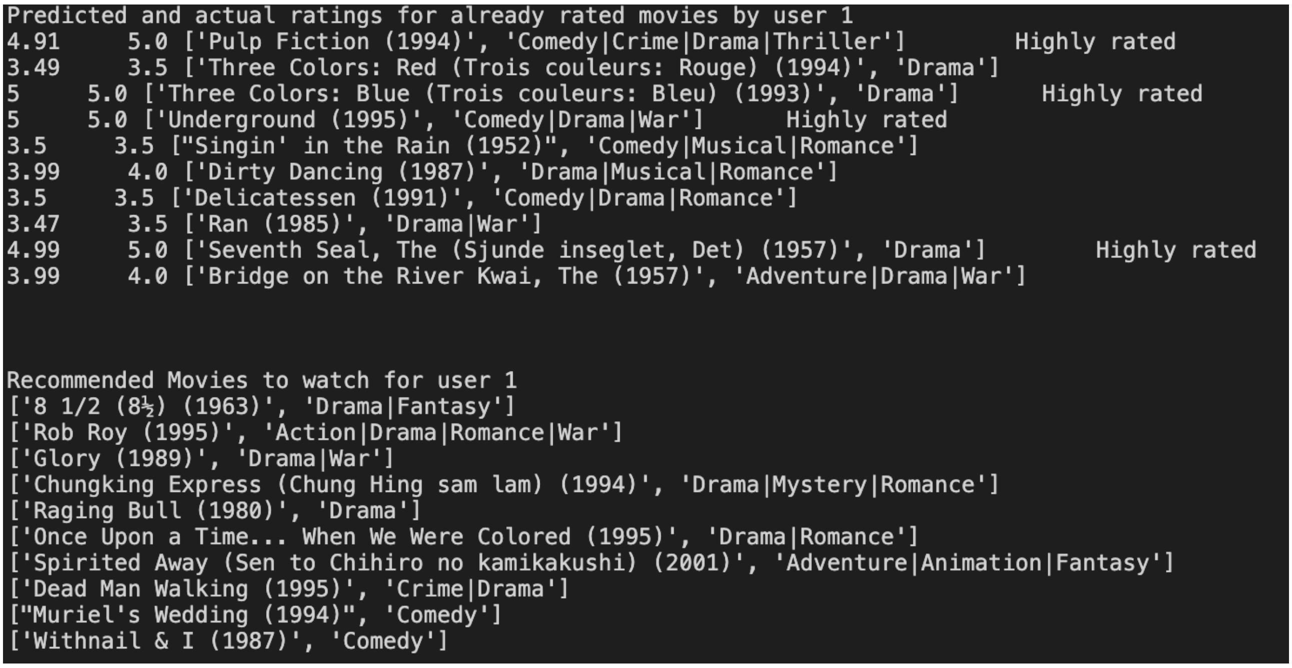


```
[[4.93739999 2.84250473 5.28807586 0.90803793 3.19643742]
 [4.07103385 2.34763595 4.57162448 1.0986365  2.81016292]
 [1.11149973 0.69265429 4.04635291 4.93075507 3.0778666 ]
 [0.98281688 0.60825035 3.34983971 3.98249328 2.53321502]
 [2.28495254 1.35583435 4.63255976 4.03678385 3.28380684]
 [1.83676809 1.0848083  3.44864644 2.78945608 2.41240712]]
```

The error seems to be minimum as the difference between actual ratings and predicted ratings can be seen to minimum. Any more training will make the model overfit the data and lead to bad generalisation.

Results

The predicted ratings for the actual ratings of the user have been displayed to showcase which movies have been highly rated by the user.
If the recommended movies are examined, we can see that most movies with the Drama genre and a release date in 1990's are on top of the list. This can be explained by the fact that the user likes movies that belong to the drama genre and released at sometime in the 1990's.



```
Predicted and actual ratings for already rated movies by user 1
4.91     5.0 ['Pulp Fiction (1994)', 'Comedy|Crime|Drama|Thriller']       Highly rated
3.49     3.5 ['Three Colors: Red (Trois couleurs: Rouge) (1994)', 'Drama']
5     5.0 ['Three Colors: Blue (Trois couleurs: Bleu) (1993)', 'Drama']      Highly rated
5     5.0 ['Underground (1995)', 'Comedy|Drama|War']      Highly rated
3.5     3.5 ["Singin' in the Rain (1952)", 'Comedy|Musical|Romance']
3.99     4.0 ['Dirty Dancing (1987)', 'Drama|Musical|Romance']
3.5     3.5 ['Delicatessen (1991)', 'Comedy|Drama|Romance']
3.47     3.5 ['Ran (1985)', 'Drama|War']
4.99     5.0 ['Seventh Seal, The (Sjunde inseglet, Det) (1957)', 'Drama']        Highly rated
3.99     4.0 ['Bridge on the River Kwai, The (1957)', 'Adventure|Drama|War']



Recommended Movies to watch for user 1
['8 1/2 (8½) (1963)', 'Drama|Fantasy']
['Rob Roy (1995)', 'Action|Drama|Romance|War']
['Glory (1989)', 'Drama|War']
['Chungking Express (Chung Hing sam lam) (1994)', 'Drama|Mystery|Romance']
['Raging Bull (1980)', 'Drama']
['Once Upon a Time... When We Were Colored (1995)', 'Drama|Romance']
['Spirited Away (Sen to Chihiro no kamikakushi) (2001)', 'Adventure|Animation|Fantasy']
['Dead Man Walking (1995)', 'Crime|Drama']
["Muriel's Wedding (1994)", 'Comedy']
['Withnail & I (1987)', 'Comedy']
```