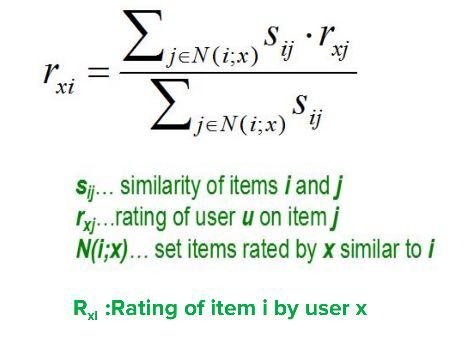
**Various Techniques for Implementing the Recommender System**

🡪Collaborative Filtering

Collaborative filtering filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future.

**Formulation:**

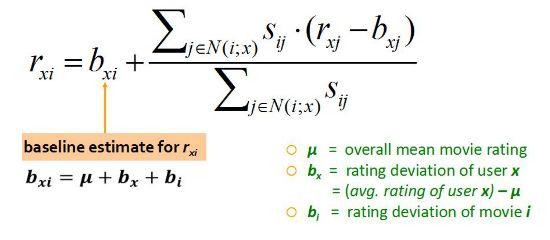
****

🡪Collaborative Filtering with Baseline Approach

The Baseline approach is used to take care of the cold start problem. The baseline is the avg rating + deviation of user + deviation of the movie. The CF gives the deviation from the baseline.

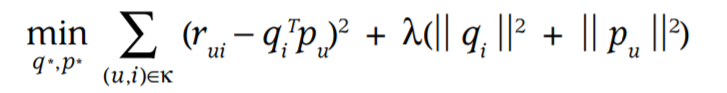
We solve the problem of strict and generous raters by using the centered cosine similarity.

**Formulation**​**:**

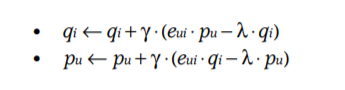
****

🡪Latent Factor Model

Latent Factor Model is another approach for a low-rank decomposition that takes into account hidden or latent factors of users and items to predict the corresponding ratings. It is based upon stochastically minimizing the error in observed and predicted ratings. The error is given by:



The update rule used for stochastic gradient descent is:



In this manner an m X n utility matrix is converted into multiplication of two lower rank matrices m X k and k X n where k is the number of latent factors taken.

**Results**

Users: 610

Movies: 10,000

Number of ratings: 80,000

No of Neighbours taken: 5

No of Latent Factors: 15

**The algorithms were evaluated based on 3 factors:**

1. Root Mean Square Error
2. Mean Average Error
3. CPU Time Taken for Prediction

|  |  |  |  |
| --- | --- | --- | --- |
| **Recommender** | **RMSE** | **Mean Average Error** | **CPU time taken for** |
| **System** |  |  | **Prediction (in seconds)** |
| **Technique** |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Collaborative** |  | 1.6503079 | 25.341563 |
| **Filtering** | 1.9496499 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Collaborative** |  |  |  |
| **Filtering with** | 1.824160021 | 1.531614406 | 21.2300754 |
| **Baseline** |  |  |  |
| **Approach** |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | 3.030202661685128e-14 | 3.624732420451774e-15 | 10.753729820251465 |
| **SVD with 100%** |  |  |  |
| **energy** |  |  |  |
|  |  |  |  |
| **SVD with 90%** | 2.8798104206150226e-14 | 3.0827985428533065e-15 | 9.122525930404663 |
| **energy** |  |  |  |
|  |  |  |  |
| **CUR with 100%** | 1.4349944288166214 | 0.9273532764330717 | 9. 236923456192017 |
| **energy** |  |  |  |
|  |  |  |  |
| **CUR with 90%** | 0.9719021316466407 | 0.5893627598269802 | 8.79685640335083 |
| **energy** |  |  |  |
|  |  |  |  |
| **Latent Factor Model** | 0.9200011 | 0.7437974 | 1.04838252067565918 |
|  |  |  |  |
|  |  |  |  |

**Packages Used**

Here are the following python packages used:

1. Numpy
2. math
3. pandas
4. time
5. csv
6. scipy