**IR**

**ASSIGNMENT-2**

**Team members-**

1.Ayush Pandey (2015B3A70517H)

2.Chitti Ankith Reddy (2015B3A70534H)

3.KSSV Rama Krishna (2015B3A70610H)

4.Raj Nandwani (2015B3A70809H)

**Language: Python**

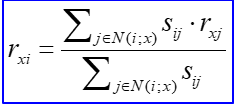
**Data Set:** The data set considered contains around 1 million ratings from 6040 users across around 4000 movies.

The **list of packages** used in code is-

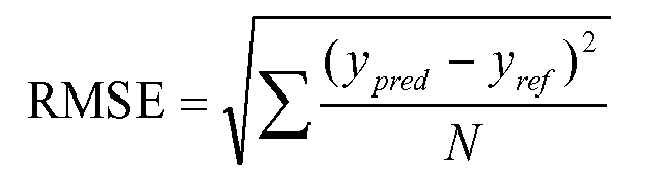
1. **Timit** and **Time** are used to find the running time of ratings prediction part in code.
2. **Random** is used to find random elements from a list and for choosing the rows and columns in CUR.
3. **Numpy** package is used for array and matrix manipulation.
4. **Multiprocessing** module is used for creating multiple processes, using lock variables for critical section and using shared memory.
5. **Xlrd** is used for reading excel files
6. **Openpyxl** is used for writing to excel files.
7. **Scipy** is used for handling sparse matrix
8. **Sklearn** is used for train-test splitting of dataset.

**Formulas:**

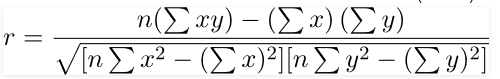
1. **Collaborative Filtering : User-user approach has been followed with similary score calculated based on the below formula:**



1. **Root Mean Square Error:**



1. **Spearman Rank Correlation: Formula used for calculating correlation between two sets X and Y is:**



**Assumptions & Considerations-**

1. While performing the SVD decomposition of the intersection matrix in CUR, the values in Sigma which are less than 0.1 have not been included in the pseudoinverse matrix.
2. In calculating precision at top k for SVD and CUR, only those movies are considered which were originally rated higher than 3 by the users.
3. For calculating precision in CF, the threshold rating for recommendation and relevancy is taken to be 3.5
4. 1000 rows and columns were randomly selected in CUR Decomposition
5. Minimum 10 users were used as neighbors for estimating the rating in CF.
6. In Calculating RMSE and Spearman Rank Correlation in SVD and CUR, the entire original user-movie matrix is compared with the low rank approximation obtained in both the methods.
7. Ranks haven’t been considered in calculating correlation owing to the huge amount of time taken by the written function.
8. **To handle generous and strict raters, before the similarity score is calculated, we average out the ratings across users.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Recommender System Technique** | **Root Mean Square Error (RMSE)** | **Precision on top K** | **Spearman Rank Correlation** | **Time taken for prediction** |
| Collaborative 0.96 0.813 0.494 580s (For  200,000 ratings) | | | | |
| Collaborative  along with  Baseline 0.90 0.81 0.57 580s | | | | |
| approach | | | | |
| SVD 1.0854 0.031 0.0045 520s | | | | |
| SVD with 90%  retained energy 1.07761 0.03 0.00465 511s | | | | |
| CUR 2.145 0.2 0. 157 210s | | | | |
| CUR with 90%  retained energy 0.781 0.333 0.610 223s | | | | |
|  | | | | |

* Collaborative Filtering gives precise estimates of the ratings
* As expected CUR takes much less time to compute compared to SVD.
* SVD requires a huge amount of RAM to store and manipulate the matrices.
* CUR might give biased results in some cases as the same columns/rows may be samples again.