BENCHMARKING FOR RECOMMENDER SYSTEM (MFRISE)

Mahesh Mali
Computer Engineering Department, SVKMs NMIMS, Mukesh Patel School of Technology Management and Engineering, Mumbai, (India).
E-mail: maheshmalisir@gmail.com

Dhirendra Mishra
Computer Engineering Department, SVKMs NMIMS, Mukesh Patel School of Technology Management and Engineering, Mumbai, (India).

M. Vijayalaxmi
Computer Engineering Department, V.E.S. College of Engineering, Mumbai University, (India).

Reception: 05/11/2022 **Acceptance**: 20/11/2022 **Publication**: 29/12/2022

Suggested citation:

Mali, M., Mishra, D., y Vijayalaxmi, M. (2022). Benchmarking for Recommender System (MFRISE). *3C TIC. Cuadernos de desarrollo aplicados a las TIC, 11*(2), 146-156. https://doi.org/10.17993/3ctic.2022.112.146-156

https://doi.org/10.17993/3ctic.2022.112.146-156

ABSTRACT

The advent of the internet age offers overwhelming choices of movies and shows to viewers which create need of comprehensive Recommendation Systems (RS). Recommendation System will suggest best content to viewers based on their choice using the methods of Information Retrieval, Data Mining and Machine Learning algorithms. The novel Multifaceted Recommendation System Engine (MF-RISE) algorithm proposed in this paper will help the users to get personalized movie recommendations based on multi-clustering approach using user cluster and Movie cluster along with their interaction effect. This will add value to our existing parameters like user ratings and reviews.

In real-world scenarios, recommenders have many non-functional requirements of technical nature. Evaluation of Multifaceted Recommendation System Engine must take these issues into account in order to produce good recommendations. The paper will show various technical evaluation parameters like RMSE, MAE and timings, which can be used to measure accuracy and speed of Recommender system. The benchmarking results also helpful for new recommendation algorithms.

The paper has used MovieLens dataset for purpose of experimentation. The studied evaluation methods consider both quantitative and qualitative aspects of algorithm with many evaluation parameters like mean squared error (MSE), root mean squared error (RMSE), Test Time and Fit Time are calculated for each popular recommender algorithm (NMF, SVD, SVD++, SlopeOne, Co-Clustering) implementation. The study identifies the gaps and challenges faced by each above recommender algorithm. This study will also help researchers to propose new recommendation algorithms by overcoming identified research gaps and challenges of existing algorithms.

KEYWORDS

Comparing recommender system, bench-marking recommendation system algorithms, comparing recommendation algorithms, challenges of various recommendation algorithms, Performance evaluation of Recommendation algorithms.

https://doi.org/10.17993/3ctic.2022.112.146-156

1. INTRODUCTION

Availability of internet and global resources has increased number of availability of movies and shows which can be viewed by users. Recommendation Systems are tools used to give movie recommendations to the end-users based on their likes or likes of the similar users [1]. Recommender systems are good for both service providers as well as users. They reduce the time to find and selecting correct item on internet. A recommender system is an information filtering system which recommends the best movies to the user by considering some similarity between users or movies or user ratings for movies. The existing types of recommendation systems algorithms are Collaborative Filtering (CF) and Content-Based Filtering (CB).

Multiple well-known recommendation algorithms based on above categories are already proposed, KDD algorithm, SVD algorithms, SlopeOne and Co-Clustering algorithm. In this paper we have implemented and analyzed their comparative performance, as it can be used for benchmarking performance of our proposed multifaceted recommender system (MFRISE). This paper also explained the challenges and limitations of each algorithm. Such, challenges can be used to improve performance quality recommendations by modifying algorithm.

2. ARCHITECTURE OF MULTIFACETED RECOMMENDER SYSTEM (MFRISE)

The general recommendation system algorithm will use the mathematical function to suggest recommendations based on past similarity between users and movies [3]. The algorithm must be able to measure the usefulness of movie to user. In order to get good recommendations, we need lot of implicit and explicit data. Data coming from user ratings is acts as explicit data. Implicit data fetched from social data and watch history. The MFRISE is hybrid recommender system introduced by our paper which is used for Improvement Recommendations with help of identifying similar movies using content based (CB) filtering and perform multi-clustering and find the community impact on recommendations using text analytics,

Step 1 : Data Prepossessing

Step 2 : Similarity based recommendations

Step 3 : Clustering using Items similarity

Step 4 : Clustering using User similarity

Step 5 : Find Social impact on items

Step 6 : Multi-cluster & interactions

Step 7 : Ranking recommendations to user

Step 8 : Validation and testing

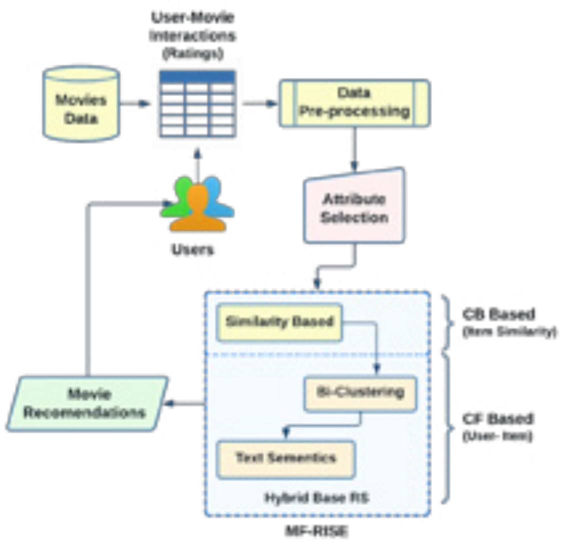

Fig. 2. MS-RISE Proposed Architecture.

https://doi.org/10.17993/3ctic.2022.112.146-156

The detailed Implementation of Multifaceted Recommendation System Engine (MF-RISE) is included in upcoming paper on our proposed work. The proposal of method and

experimentation on benchmarking algorithms is proposed in this paper.

2.1 RECOMMENDER SYSTEM (RS) EVALUATION METHODS

The evaluation of recommendation system algorithm is not as easy as evaluation of any other machine learning algorithms, as the recommendation output for each user is different than other user [3]. This is main reason for which we cannot simply divide dataset into training and testing data. The methods used for evaluation RSs are,

a) Train - Test Split [6]: In RS algorithms it is not possible to take separate Training data set and testing data set, since the training data used to fit algorithm and test data set is used for evaluating RS algorithm. But, the user in training data may not be available in test data so, it is difficult to use separate test data. We have used masking method, rating values for some users are masked and then rating are predicted using algorithm then we can compare these ratings for checking accuracy.

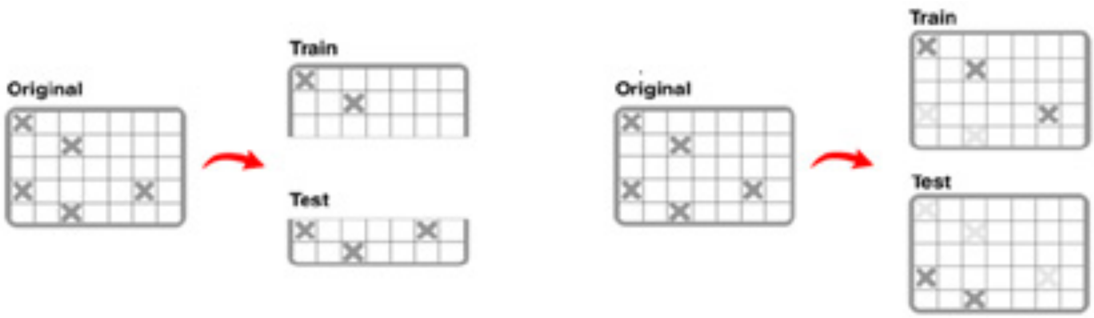

Fig. 3. Train-Test Split (masking).

b) K-Fold Cross Validation Method [7]: Cross-validation is a statistical method used to estimate the performance of RS algorithms or any machine learning algorithm.

- Dataset divided in k Groups
- Accept that group as Test Data
- Take remaining groups as Training Data
- Fit Model on Training Data
- Evaluate on Test Data
- Calculate Evaluation Score

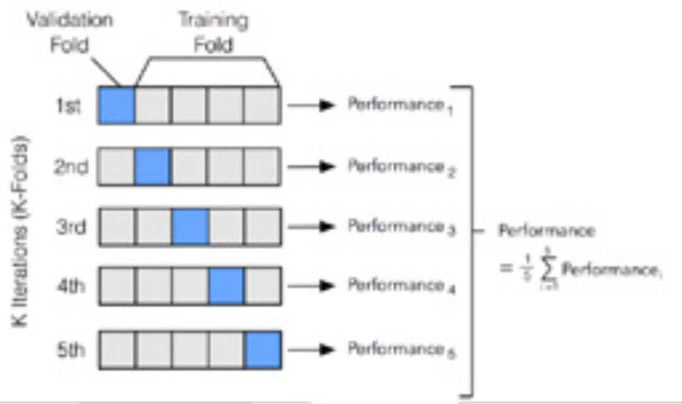

Fig. 4. K-Fold Cross Validation Concept.

3. BENCHMARKING RECOMMENDER SYSTEM (RS) ALGORITHMS

We usually categorize recommendation engine algorithms as collaborative filtering models and content-based models. In this paper, we are going to study and discuss few advantages and drawbacks of some popular recommender algorithms to compare their performances based on various evaluation

https://doi.org/10.17993/3ctic.2022.112.146-156

metrics. This paper will set a benchmark for our proposed implementation of MF-RISE with below popular RS algorithms,

a) Similarity Based Algorithm

- Baseline algorithm

b) Neighborhood Algorithm

- K-Nearest Neighbors Algorithm (KNN)

c) Hybrid Methods

- Co-Clustering
- Slope-One

d) Matrix Factorization Method

- Single value Decomposition (SVD)
- Advanced SVD (SVD++)
- Negative Matrix Factorization (NMF)

4. EVALUATION METRICS FOR RECOMMENDER SYSTEMS

The most important thing for RS is to evaluate the performance of algorithm. The traditional algorithm evaluation metrics used to measure errors may not be effective for recommendation algorithms, as there are different recommendations for each user and no recommendations can be same for even same user. We need to take help of various traditional and modern methods for validating recommendation results.

A. Accuracy Metrics

Recommendation accuracy will measure difference between recommender's estimated ratings and actual user ratings.

a) Mean Absolute Error(MAE): [5] Absolute Error is the amount of error in prediction and actual rating.

$$\text{AbsoluteError} = |r_{ui} - \hat{r}_{ui}|$$

$$r_{ui} = \text{Rating from the proposed algorithm}$$
$$\hat{r}_{ui} = \text{Actual user Rating}$$

The mean value of absolute errors can be given as Mean Absolute Error(MAE),

$$\text{MAE} = \frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}|$$

b) Mean Square Error(MSE): [5] The measure of the average of the squares of the errors is called as Mean Square Error(MSE). MSE is not as small as MAE. MSE can be calculated as,

$$\text{MSE} = \frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2$$

c) Root Mean Square Error(RMSE): [5] The Root Mean Squared Error(RMSE) is better in terms of performance when dealing with larger error values. RMSE is more useful when lower residual values

https://doi.org/10.17993/3ctic.2022.112.146-156

are preferred. MSE is highly biased for higher values. Therefore, RMSE is more preferred accuracy measure.

B. Classification Metrics: Recommendation accuracy can also be measured by traditional precision and recall metrics. Recommended items has a high interaction value i.e. number of ratings, can be considered as most accurate predictions.

i) Precision: Precision is the ratio of true positives (number of relevant results) and total positives recommended items.

ii) Recall: A Recall is essentially the ratio of number of relevant items that are recommended to all relevant items.

C. Ranking Metrics [4]: Recommendation accuracy can also measure by Top-N results given by RS algorithm.

i) Hit Rate: The Hit occurs, if a user rated one of the top-10 recommended movie. So, first we find the Top 10 movie recommendations. then, we find movies rated by user. If user rates a movie which is already recommended, we consider that as one hit. Finally, ratio of Number of hits and total recommended movies is Hit Ratio.

ii) Miss Ratio: The Miss occurs, if a user rated movie not present in the top-10 recommended movie. If user rates a movie which is not recommended, we consider that as Miss. Finally, ratio of Number of Misses and total recommended movies is Miss Ratio.

D. Execution Time Metrics

Recommendation algorithm speed can be one of the important metrics, as we are dealing with very large set of data. The time required for algorithm to calculate the recommendation from input dataset is used as execution time. The time required for fitting algorithm is Fit Time and the time taken to run it on test data is Test time.

5. EXPERIMENTAL SETUP

We build experiments based on MovieLens datasets provided by Group Lens [11]. MovieLens datasets contain user ratings for multiple movies. The dataset contains 2113 users, 10197 movies and 855598 user ratings including tag assignments. The datasets contain only users that have rated at least 20 movies. They have conventional ratings which is preferred when predicting ratings. Since the Root Mean Squared Error(RMSE) and Mean Squared Error(MAE) values are depended on the rating scale, the results will be more comparable. We have used 5-Fold cross validation method for selecting training and testing dataset more effectively. The performance after each fold is analyzed and decided to work on 5 Folds for memory and time optimization. After 5-Folds, the performance is not improved considerably hence, decided to work with 5-Fold method [7]. The comparison of MovieLens datasets,

Table I. Comparisons of Datasets.

Dataset	HetRec [6]	ml latest [6]	100k [6]
Movies	10197	9742	1682
Users	2113	610	943
Ratings	855598	100836	1000000

6. PERFORMANCE EVALUATION

A. Baseline Algorithms [12]

The similarity based algorithms are content based algorithms used for predicting a random rating based on the distribution, this algorithm assumes user ratings are normally distributed. The prediction is generated from a normal distribution N (μ, σ^2) where μ and σ are estimated from the training data using Maximum Likelihood Estimation [3], If user u is unknown, then the bias b_u is assumed to be zero. The same applies for item i with b_i.

$$\hat{\mu} = \frac{1}{|R_{train}|} \sum_{r_{ui} \in R_{train}} r_{ui}$$

$$\hat{\sigma} = \sqrt{\sum_{r_{ui} \in R_{train}} \frac{(r_{ui} - \hat{\mu})^2}{|R_{train}|}}$$

Predicted Rating is,

$$\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i$$

The best part of algorithm is simple implementation and useful for comparing algorithm accuracy. The points to improve is need more personalized predictions and less execution time for complex predictions. This algorithm also faces the problem of cold start for novice system users.

B. Matrix factorization Algorithm [16]

The Single Value Decomposition (SVD) is a Matrix factorization algorithm popularized by Simon Funk during the Netflix Prize. This is equivalent to Probabilistic Matrix Factorization algorithm. It Constructs a matrix with the row of users and columns of items and the elements are given by the users' ratings The singular value decomposition [15] is a method of decomposing a matrix into three other matrices.

$$A = U\ S\ V^T$$

Where A= m x n utility matrix

U= m x r rating singular matrix

The prediction r_{ui} is set as,

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

If user u is unknown, then the bias b_u and the factors p_u are assumed to be zero.

The SVD is good for with few datasets and it can improve performance on many algorithms. It majorly uses the Principal component analysis(PCA) which is useful for dimensional reduction.

C. SVD++ Algorithm [20]

The Single Value Decomposition (SVD++) is extension of SVD algorithm, with considering implicit ratings This is equivalent to Probabilistic Matrix Factorization algorithm. It Constructs a matrix with the row of users and columns of items and the elements are given by the users' ratings.

The prediction r_{ui} is set as,

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$$

https://doi.org/10.17993/3ctic.2022.112.146-156

Where, the y_j terms are a new set of item factors that capture implicit ratings. Here, an implicit rating describes the fact that a user u rated an item j, regardless of the rating value. If user u is unknown, then the bias b_u and the factors p_u are assumed to be zero.

D. Non-Negative Matrix Factorization(NMF) [22]

The Non-Negative Matrix Factorization(NMF) is equivalent to Non negative Matrix Factorization algorithm. It Constructs a matrix with the row of users and columns of items and the elements are given by the users' ratings.

The NMF algorithm can improves performance on many algorithms. NMF based methods used in for solving problems in computer vision. The computational complexity of CF based algorithm is very high and it results in many missing ratings. Some major improvements are required to achieve high computational efficiency and prediction accuracy.

E. Co-Clustering Algorithm [25]

A Co-Clustering is based on collaborative filtering algorithm. This approach is based on simultaneous clustering of users and movies (items) for efficient CF based algorithm.

In Co-clustering method, every users and movies are assigned some clusters C_u, C_i, and some co-clusters C_{ui} The prediction r_{ui} is set as,

$$r_{ui} = C_{ui} + (\mu_u - C_u) + (\mu_i - C_i)$$

Where, If the user is unknown, the prediction is $r_{ui} = \mu_i$, If the item is unknown, the prediction is $r_{ui} = \mu_u$, If both are unknown, the prediction is $r_{ui} = \mu$

The co-clustering algorithm has good control over learning and can consider multiple dimensions of data. but, needs more execution time in few cases for some critical recommendation. The cold start issue become major issue in this algorithm.

F. Slope One Algorithms [27]

Slope One algorithm is based on the movie-user rating matrix based on the linear model y=xb+c. Where, parameter y is the rating of the predicted target user on the target movie, parameter x is the rating of the target user on the reference movie, and parameter b is the deviation value of the user's score of different movies. Slope One algorithm calculates enter of the evaluation of excessive user ratings mean the score difference between the movies, and then at the time of target users recommend, uses the linear relationship, estimate the prediction score of the movies y according to the target user's score of project x and the deviation value b , that is, generate the prediction by using the deviation value of all users among different movies. Slope One algorithm is simple in calculation and having good performance. It can handle cold start issue well by predicting ratings. But, the fit time will be higher as compare to other algorithms.

7. RESULTS

All experiments are run on a Desktop with Intel Core i5 8th gen (CPU@2.30GHz) and 8GB RAM, all data stored on solid state Memory (SSD) for faster access and optimum performance. In this paper, we present the various evaluation parameter like average RMSE, MAE and total execution time of various algorithms (used in study) with a 5-fold cross-validation procedure.

Table II. Evaluation Metrics for Benchmark Algorithms.

Parameter	*RMSE*	*MAE*	*FIT TIME*	*TEST TIME*
BaseLine	1.52	1.2224	0.9	0.13
KNN	0.9793	0.7736	0.41	2.32
SVD	0.9379	0.7394	3.97	0.13
CO-CLUSTERING	0.9654	0.7555	1.41	0.09
SLOPEONE	0.9434	0.74527	0.55	1.71

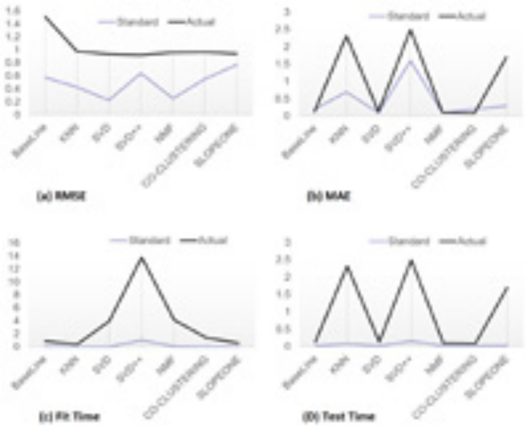

Fig. 15. Benchmark Algorithms Performance Analysis.

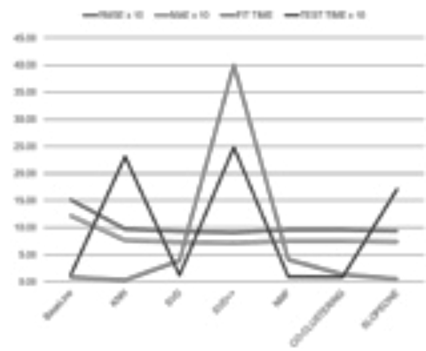

Fig. 17. Comparative Performance Analysis.

8. CONCLUSION AND FUTURE WORK

In this paper, we present a real-world benchmark for our new recommendation system algorithm. The prediction accuracy of a recommender system is dependent on various parameters. In study, we have seen all algorithms are optimized for the MovieLens dataset. SVD, NMF and co-clustering algorithm performs better on the larger dataset than the other collaborative filtering algorithm. To obtain more detailed results, testing algorithms on datasets with more similar properties can be performed.

We have deployed many important RS algorithms to study their performance comparisons, which was ubiquitous and crucial in recommendation scenarios. After comparing all algorithms, we found that SVD++ algorithm need highest Fit Time due to complexity of calculations. There is lower RMSE calculations in all algorithms except Baseline algorithm. Overall execution time of SVD algorithm and co-clustering algorithm is very lower. So, we are planning to can plan to use SVD, NMF and Co-Clustering algorithms for efficient implementation of movies recommendation process.

We can conclude that the SVD, NMF and Co-Clustering algorithm is seemingly more accurate than other the Item-based collaborative filtering algorithm for larger datasets.

REFERENCES

[1] MALI, MAHESH, DHIRENDRA S. MISHRA, AND M. VIJAYALAXMI. "Multifaceted recommender systems methods: A review." Journal of Statistics and Management Systems 23.2 (2020): 349-361.

[2] MEHTA Y., SINGHANIA A., TYAGI A., SHRIVASTAVA P., MALI M. (2020) A Comparative Study of Recommender Systems. In: Kumar A., Paprzycki M., Gunjan V. (eds) ICDSMLA 2019. Lecture Notes in Electrical Engineering, vol 601. Springer, Singapore.

https://doi.org/10.17993/3ctic.2022.112.146-156

[3] F.O. ISINKAYE, Y.O. FOLAJIMI, B.A. OJOKOH, Recommendation systems: Principles, methods and evaluation, Egyptian Informatics Journal, Volume 16, Issue 3, 2015, Pages 261-273, ISSN 1110-8665,
https://doi.org/10.1016/j.eij.2015.06.005.

[4] PU, PEARL, LI CHEN, AND RONG HU. "Evaluating recommender systems from the user's perspective: survey of the state of the art." User Modeling and User-Adapted Interaction 22.4 (2012): 317-355.

[5] CREMONESI, PAOLO, ET AL. "An evaluation methodology for collaborative recommender systems." 2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution. IEEE, 2008.

[6] CANAMARES, ROCIO, PABLO CASTELLS, AND ALISTAIR MOFFAT. "Offline evaluation options for recommender systems." Information Retrieval Journal 23.4 (2020): 387-410.

[7] MORENO-TORRES, JOSE GARCIA, JOSE A. SAEZ, AND FRANCISCO HERRERA. "Study on the impact of partition-induced dataset shift on k-fold cross validation." IEEE Transactions on Neural Networks and Learning Systems 23.8 (2012): 1304-1312.

[8] FAYYAZ, ZESHAN, ET AL. "Recommendation systems: Algorithms, challenges, metrics, and business opportunities." applied sciences 10.21 (2020): 7748.

[9] SHARMA, RITU, DINESH GOPALANI, AND YOGESH MEENA. "Collaborative filtering-based recommender system: Approaches and research challenges." 2017 3rd international conference on computational intelligence and communication technology (CICT). IEEE, 2017.

[10] HUG, NICOLAS. "Surprise: A Python library for recommender systems." Journal of Open Source Software 5.52 (2020): 2174.

[11] F. MAXWELL HARPER AND JOSEPH A. KONSTAN. 2015. The MovieLens Datasets: History and Context. ACM Trans. Interact. Intell. Syst. 5, 4, Article 19 (January 2016), 19 pages. https://doi.org/10.1145/2827872

[12] RENDLE, STEFFEN, LI ZHANG, AND YEHUDA KOREN. "On the difficulty of evaluating baselines: A study on recommender systems." arXiv preprintarXiv:1905.01395 (2019).

[13] WANG, KAI, ET AL. "Rl4rs: A real-world benchmark for reinforcement learning based recommender system." arXiv preprint arXiv:2110.11073 (2021).

[14] AHUJA, RISHABH, ARUN SOLANKI, AND ANAND NAYYAR. "Movie recommender system using k-means clustering and k-nearest neighbor." 2019 9th International Conference on Cloud Computing, Data Science and Engineering (Confluence). IEEE, 2019.

[15] WANG, JIANFANG, ET AL. "A collaborative filtering algorithm based on svd and trust factor." 2019 international conference on computer, network, communication and information systems (CNCI 2019). Atlantis Press, 2019.

[16] MEHTA, RACHANA, AND KEYUR RANA. "A review on matrix factorization techniques in recommender systems." 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA).
IEEE, 2017.

[17] RICCI, FRANCESCO, LIOR ROKACH, AND BRACHA SHAPIRA. "Recommender systems: introduction and challenges." Recommender systems handbook. Springer, Boston, MA, 2015. 1-34.

[18] ZHANG, WEIWEI, ET AL. "Recommendation system in social networks with topical attention and probabilistic matrix factorization." PloS one 14.10 (2019): e0223967.

[19] VOZALIS, MANOLIS G., AND KONSTANTINOS G. MARGARITIS. "A recommender system using principal component analysis." Published in 11th panhellenic conference in informatics. 2007.

[20] AL SABAAWI, A., KARACAN, H. AND YENICE, Y. (2021). Two Models Based on Social Relations and SVD++ Method for Recommendation System. International Association of Online Engineering. Retrieved May 9, 2022
from https://www.learntechlib.org/p/218689/.

[21] GUAN, XIN, CHANG-TSUN LI, AND YU GUAN. "Matrix factorization with rating completion: An enhanced SVD model for collaborative filtering recommender systems." IEEE access 5 (2017): 27668-27678.
[22] LUO, XIN, ET AL. "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems." IEEE Transactions on Industrial Informatics 10.2 (2014): 1273-1284.
[23] ZHANG, SHENG, ET AL. "Learning from incomplete ratings using nonnegative matrix factorization." Proceedings of the 2006 SIAM international conference on data mining. Society for Industrial and Applied
Mathematics, 2006.
[24] KUMAR, RAJEEV, B. K. VERMA, AND SHYAM SUNDER RASTOGI. "Social popularity based SVD++ recommender system." International Journal of Computer Applications 87.14 (2014).
[25] FENG, LIANG, QIANCHUAN ZHAO, AND CANGQI ZHOU. "Improving performances of Top-N recommendations with co-clustering method." Expert Systems with Applications 143 (2020): 113078.
[26] LI, MAN, LUOSHENG WEN, AND FEIYU CHEN. "A novel Collaborative Filtering recommendation approach based on Soft Co-Clustering." Physica A: Statistical Mechanics and its Applications 561 (2021): 125140.
[27] WANG, QING-XIAN, ET AL. "Incremental Slope-one recommenders." Neuro-computing 272 (2018): 606-618.
[28] SONG, YUE TING, AND SHENG WU. "Slope one recommendation algorithm based on user clustering and scoring preferences." Procedia Computer Science 166 (2020): 539-545.
[29] SALAM PATROUS, ZIAD, AND SAFIR NAJAFI. "Evaluating prediction accuracy for collaborative filtering algorithms in recommender systems." (2016).

https://doi.org/10.17993/3ctic.2022.112.146-156