OPEN ACCESS
ISSN 2460-9056
http://socj.telkomuniversity.ac.id/ijoict/

Intl. Journal on ICT
Vol. 8, No. 2, Dec 2022. pp. 11-21
doi: 10.21108/ijoict.v8i2

Recommender System Based on Matrixs Factorization on Twitter Using Random Forest (Case Study: Movies on Netflix)

Bagas Teguh Imani [1*], Erwin Budi Setiawan [2]

[1,2]*School of Computing, Telkom University*
Bandung, Indonesia

[*] bagasteguhimani@student.telkomuniversity.ac.id, erwinbudisetiawan@telkomuniversity.ac.id

Abstract

In this day and age, there is a lot of entertainment that can be done, one of which is watching movies using the Netflix platform. When you want to watch, sometimes users can be confused about which movies to watch according to their tastes and interests, which requires a solution, namely by using a recommendation system. The recommendation system is a system that emerged as a solution to provide information by learning data from users with previously stored data items. One of the recommendation system techniques is Collaborative Filtering. By using Collaborative Filtering, this study will focus on using two Matrix Factorization-based methods, which are Non-Negative Matrix Factorization and Probabilistic Matrix Factorization, to try to solve the gap in the data. This study will use the Random Forest algorithm to improve the results of good predictions. A recommendation system based on Matrix Factorization on Twitter will be made using Random Forest in a case study of films on Netflix. The experimental results have shown that the use of the system gets a Mean Absolute Error (MAE) value of 0.7641 to 0.8496 and a Root mean squared error (RMSE) of 1.0359 to 1.1935.

.

Keywords: Matrix Factorization, Random Forest, Recommender System, Twitter.

I. Introduction

Social media has emerged as one of the most widely used platforms by people. As a result, various activities are often shared by social media users to express their points of view [1]. Twitter is one of the most widely used social media platforms for social media users and allows users to create short "Tweet" messages with a limit of 140 characters. Apart from sharing their interests, opinions, and knowledge, users on Twitter can also search for the latest news and reviews, such as movies [2]. In this era, many people like to watch movies, and many platforms provide movie viewing services. Netflix is one such popular platform that is a subscription-based streaming and production company. Netflix enables its subscribers to watch movies of various genres that interest them. With several titles and genres available on the Netflix platform, it can become confusing for users to select the desired shows. Therefore, a solution referred to as a "recommendation system" (RS) is needed.

RS is a system that focuses on filtering user information to overcome the problem of excessive information, whereby it will predict unknown items for the user [3]. The recommendation system has been widely used by well-known companies such as Amazon.com, which allows users to get recommendations for books to buy;

Received on 01 Aug 2022. Revised on 12 Aug 2022. Accepted and Published on 28 Dec 2022.

Netflix, which provides viewing recommendations to users; and YouTube, which also provides viewing recommendations to its users. In addition, there are many other examples of companies in the e-commerce sector and other fields that have used it [4][5][6]. The recommendation system has many techniques, one of which is collaborative filtering (CF). CF is a machine learning method that is used to identify a relationship in data and CF is often used in recommendations to identify the similarities between user and item data. In CF, there are often problems with data, such as data that contains a lot of blank data (data sparsity), so an algorithm is needed to minimize these problems. So this research will use the Random Forest algorithm and the CF Matrix Factorization technique to solve the problem and produce a good rating prediction.

In this study, the Random Forest (RF) algorithm will be used to develop a Netflix movie recommendation system on Twitter based on Matrix Factorization (MF). It is believed that RF is capable of improving the results of good recommendations. In addition, it works with incomplete data attributes and can handle large samples [7]. MF is also used due to its ability to enhance prediction accuracy in measured datasets. Furthermore, it can also be used to remove dimensions from the item's space and retrieve latent or hidden relationships between items in the data set [8]. Through the utilization of MF, the research will utilize the use of MF methods, namely Probabilistic Matrix Factorization (PMF) and Non-Negative Matrix Factorization (NNMF), which will be compared with the evaluation results.

The purpose of this study is to focus on how to implement a recommendation system by using MF with Random Forest. Both MF methods will be tried to predict the gap in the data rating between users and movies. After that, we will compare the results of both methods. The best methods will be applied to random forest to try to increase the results and will be expected to provide a low prediction error rate and can provide accurate predictions to the user.

The rest of the journal is organized as follows. Section two discusses issues related to the topic of research. Whereas the third section describes our research method for making a recommendation system, Section IV provides the experiment results and discussion, followed by the conclusion in the last section.

II. LITERATURE REVIEW

This study was based on several existing research references with links to the method and the intended topic. The purpose of the reference is to have references and limitations on the methods used in this study.

Twitter is a social media platform where short messages, or tweets, are shared among a large number of users through a very simple messaging mechanism. Twitter users can easily get access to information and many people can interact with them [12]. A recommendation system is a system that estimates users' preferences on items and recommends items that they might like. Recommendation models are usually classified into three categories, namely Collaborative Filtering, Content-Based, and Hybrid [14]. Matrix Factorization (MF) is a recommendation system technique that belongs to the Model Based-Collaborative Filtering category. Matrix Factorization characterizes items and users inferred from the item rating pattern. High correspondence between items and user factors leads to recommendations. This technique also offers a lot of flexibility for modeling various real-life situations [5].

The Probabilistic Matrix Factorization (PMF) is one of the MF methods that removes bias parameters from users and items to calculate predictions of interaction [19]. PMF applies a probabilistic approach to matrix factorization. PMF will estimate the noise between the target value and the predicted evaluation. The Non-Negative Matrix Factorization (NNMF) method works by decomposing a large matrix into 2 smaller matrices [19]. This method works much like regular MF, but the user and item matrices are always positive. The NNMF method will look for the approximate factorization of a rating matrix R, which is decomposed into a non-negative latent factor matrix U and V (4) [15]. Random Forest is one of the most accurate model algorithms and works efficiently on large datasets. Random Forest can effectively predict missing data accurately, even in situations where part of the data is missing and without preprocessing [16].

An evaluation is performed to determine the accuracy of the results of the experiments performed. The results of the system performance evaluation process will be used to measure the performance of the algorithm being tested. Making measurements is done using evaluation metrics [17]. In this study, evaluation metrics will be used to measure mean absolute error (MAE) and root mean square error (RMSE). These two measurements will calculate the average error or error in the prediction by comparing the actual rating value and the product rating predicted by the user [18].

In the paper [7], an experiment was conducted on how to effectively distribute predictive models using Matrix Factorization (MF) and Random Forest (RF) for a big data recommendation system. The results showed the use of these two methods gave good results compared to other methods by calculating evaluation metrics using MAE and RMSE [10], which means combining MF and RF will improve the accuracy of recommendations and reduce data sparsity problems.

This study uses NNMF and PMF methods as MF techniques in trying to predict empty ratings and combine them with Random Forest. To our knowledge, the combination of PMF and NNMF with Random Forest has not been explored further.

III. RESEARCH METHOD

A. System Design

Overall, the system that will be built is a recommendation system to recommend movies to users. The Recommender processes developed in this study consist of several steps including Crawling Data, Preprocessing Data, Matrix Factorization, Random Forest, Evaluation, and Rating Prediction results. Fig. 1 is an overview of the flow of the system created in this study.

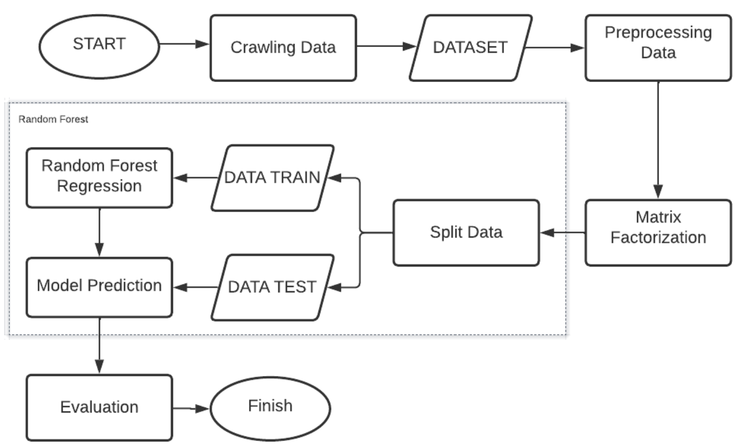

Fig. 1. Recommender System Matrix Factorization with Random Forest Flowchart

B. Crawling Data

At the beginning of the process, we crawled data from several Twitter users. Crawling is the process of retrieving data from a data source that will be used as a data set. In this study, the data will be taken from Twitter. The results of the crawl were user reviews of movie titles on the Netflix platform datasets that we obtained on the Kaggle platform. After getting user review data, we try to choose the best reviews from users for each movie.

C. Preprocessing Data

After we get the data from the crawling results, we have to do some preprocessing of the data to obtain a better quality of the data so that it can become usable data for the next step. This study uses some steps of preprocessing, which consist of Text Processing, Polarity, Transformation, and Labeling.

1) Text Processing

Text Processing is the practice of cleaning and preparing text data. After getting the review data from the crawling results, there are many sentences that use the "@, []" symbol and other symbols, or even other special characters. So we have to try to clean it from the data that has been obtained. We used several text preprocessing techniques like removing mentions, hashtags, special characters, HTTP symbols, and multiple whitespaces until we got the clean review data.

2) Polarity

Polarity is the process of changing a given sentence by considering the number of positive or negative terms in the sentence. As an illustration, if the sentence "I like this movie" has the word "like," it will be considered a positive term. But if there is the sentence "This movie is so disgusting" with the word "disgusting", it will be considered negative terms [9]. Polarity is useful for some conditions, and it does a good job of structuring data sets. In this study, we apply polarity by using library help from TextBlob [22]. It is a library for processing textual data. It gives an API for performing natural language processing (NLP) tasks like speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more [20]. After that, we do text processing on the results of the review. We try to apply polarity and the result of the review will be a number between "1" and "-1".

3) Transformation

Transformation is the process of changing the format, structure, or values of data. In this study, we only use transformation by only changing values of data or data normalization. In this step, we try to transform the polarity result by normalizing it into a rating with a value between "0" and "5".

4) Labeling

Labeling is the process of identifying raw data and adding one or more meaningful and informative labels to provide context so that a machine learning model can learn from it. After we do Data Transformation, we create a new label for our data called "Rating" That value is the result of the data transformation in the previous step.

After doing several stages of preprocessing the data above, we form the data into a data matrix between the users and the title of the movies (item) which values the rating results.

D. Matrix Factorization

In this study, we use Matrix Factorization to predict gaps in rating data. One of the points of interest of Matrix Factorization is that it permits the combination of extra data. When explicit feedback isn't accessible, the system can gather users' inclinations utilizing verifiable feedback, which in a roundabout way reflects suppositions by observing users' behavior [5]. This study focus on using 2 methods of Matrix Factorization, 1) Non-negative Matrix Factorization and 2) Probabilistic Matrix Factorization, which will later try to fill in data gaps that still have a lot of blank data (0 ratings).

1) Non-negative Matrix Factorization

The non-negative Matrix Factorization (NNMF) method works by decomposing a large matrix into 2 smaller matrices [19]. This method works like a regular MF, but the user and item matrices are always positive. The NNMF method will find the factorization of a rating matrix R which is decomposed into a matrix of non-negative latent factors U and V that can be seen in formula (1) [15]. In this study, we apply NNMF by using library help from Sklearn Decomposition [23]. We apply NNMF latent-factor or n_component valued "3" which means will take 3 number of components features from datasets.

$$R \approx UV^T \quad (1)$$

2) Probabilistic Matrix Factorization

The Probabilistic Matrix Factorization (PMF) works by removing bias parameters from users and items to calculate predictions of interaction [19]. The PMF applies a probabilistic approach to matrix factorization. PMF will estimate the noise between the target value and evaluate the prediction following the Gaussian Distribution. This distribution can be seen in formula (2).

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{n}\prod_{j=1}^{m}[N(r_{i,j}|u_i v_j^T,\sigma^2)]^{l_{i,j}} \quad (2)$$

In this formula, N(x|μ,) is a Gaussian distribution. with mean μ and variance, and which will be worth 1 if user i rate item j and is worth 0 otherwise. As we can see, this distribution is a spherical Gaussian with the following parameters $u_i v_j^T$ equals Mean and σ^2 equals Variance [15]. In this study, we apply PMF by using library help from the Surprise Library for Singular Value Decomposition (SVD) algorithm [24]. The SVD algorithm from Surprise is equivalent to Probabilistic Matrix Factorization and can be achieved by setting the "biased" parameter to "False".

E. Random Forest

After trying to apply the two matrix factorization methods, the next step is to apply a random forest. A Random Forest contains a Decision Tree, which is a combination of individual training data. A Random Forest is a combination of each good tree, which is then combined into one model.

The random forest that we use is the Random Forest Regressor. A random forest regressor is formed from a collection of regression decision trees and its output is a weighted average of the estimate from each decision tree in Fig. 2 [11]. Before applying it, we split the data first by dividing it into 20% for the data test and 80% for the data train. We implement Random Forest by using the open-source sci-kit-learn python toolkit. The random forest regressor method has two stages. Firstly, during the training stage, the forest learns a mapping between the features of the data and the parameters to be estimated. Secondly, during the testing stage, the forest uses mapping to estimate parameters from previously unseen data [11].

After applying the Random Forest Regressor, we try to tune hyperparameters for maximum performance. We use GridSearchCV and RandomizeSearchCV with the help of the sklearn library [23]. It tries all combinations of hyperparameters given in and calculates model performance for each combination by using K-fold cross-validation until we get the best parameter and apply it again to Random Forest. The result we choose is the one that produces the best evaluation metrics and rating prediction.

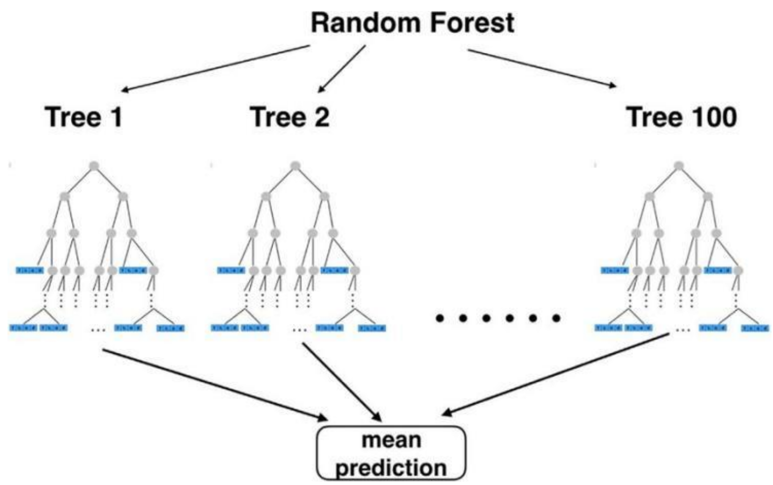

Fig. 2. Random Forest Regressor [11]

F. Evaluation

We build performance measures of prediction values by calculating prediction error rates with evaluation metrics named Mean Absolute Error (MAE) with formula (3) and Root Mean Square Error (RMSE) with formula (4). The use of both metrics is because both metrics are particularly suited for data cases that calculate normal distribution or calculate probabilities [18].

$$\text{MAE} = \frac{\sum_{u \in U, i \in I} \left| r_{u,i} - \hat{r}_{u,i} \right|}{n} \tag{3}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{u \in U, i \in I} (r_{u,i} - \hat{r}_{u,i})^2}{n}} \tag{4}$$

As we can see in formula (3), n is the sample size, $\sum_{u \in U, i \in I}$ is a symbol meaning sum, and $\left| r_{u,i} - \hat{r}_{u,i} \right|$ is the observed value for the observation, and formula (4) shows the square root of the MAE. The smaller the MAE and RMSE values, the better the resulting performance. Otherwise, if the MAE and RMSE values are high, then the resulting performance will be bad. The MAE and RMSE values range between 0 and 5 because the smallest difference between the rating prediction and the actual rating is 0, while the biggest difference is 5 [13].

G. Model Prediction

After implementing random forest regression, we can get the rating prediction results for new users who are watching a movie, which will also calculate the similarity between movies in the data. The similarity we got with the help of the sklearn library is called pairwise cosine similarity [23] by calculating the similarity of the movie datasets after applying MF. A new user will get recommendations based on rating predictions as a result of this study.

IV. RESULTS AND DISCUSSION

The objective of the experiment was to find out the evaluation metrics of Matrix Factorization with the Random Forest method using the Netflix movie datasets. The results of the experiment will compare MAE and RMSE from the Non-Negative Matrix Factorization (NNMF) method and the Probabilistic Matrix Factorization

(PMF) method. After we compare, the best result of the two methods will be taken, and we will try to use a random forest. The experiment was varied by using 50, 100, 150, 200, 250, and 300 "n_estimators" known as the number of trees in the Random Forest.

A. Data

We used a dataset containing 89.764 tweets toward 643 Kaggle dataset movies taken from 2005 to 2021 and took out 35 Twitter users that often review movies [21]. After that, we choose the best user reviews from each movie's title. The best review is a user's Twitter review that really reviews a movie title because, in the process of crawling, we find a lot of nonsense reviews. Table I provides an example of a user reviewing tweet datasets. Then, we apply polarity to make user reviews a number between -1 and 1 and transform them so that they become a rating between 0 and 5 to produce datasets as shown in Table II.

TABLE I
EXAMPLE OF USERS TWEET ON MOVIES

User	**Movies Title**	**Tweets**
User 1	*Tanda Tanya*	*Wow!"Sang Pencerah", "Rumah Dara" dan "Tanda Tanya (?)" masuk Top Ratings!!!*
User 2	*The Conjuring*	*The Conjuring, film terbaru yg masuk Top 100 film terbaik dan favorit saya sepanjang masa.*

TABLE II
POLARITY RESULT

Tweets	**Polarity**	**Rating**
Wow!"Sang Pencerah", "Rumah Dara" dan "Tanda Tanya (?)" masuk Top Ratings!!!	0.3	3.25
The Conjuring, film terbaru yg masuk Top 100 film terbaik dan favorit saya sepanjang masa.	0.625	4.06

After doing all the preprocessing stages, we create a 2D-sized matrix that connects users and movie titles, containing the rating results from the previous stage. Table III shows an example of the results of making a matrix that has a size of 643 movies and 35 users. As a note, not all users review a movie, so there is a lot of empty data (we consider it a value of "0").

TABLE III
USERS AND MOVIES MATRIX

Movies Title	**User 1**	**User 2**
Tanda Tanya	3.25	3
The Conjuring	0	4.06
The Book of Henry	0	0
Drive	1	0

B. Experiment Result

Tables IV-V show the experiment results using Non-Negative Matrix Factorization (NNMF) and Probabilistic Matrix Factorization (PMF), respectively. After applying both Matrix Factorization methods, it can fill more than 70% of the gaps in the datasets. More details can be seen in Table IV below. And it shows that Non-Negative Matrix Factorization has better performance than Probabilistic Matrix Factorization, which can leave only 15% data sparsity in the datasets.

TABLE IV
SPARSITY AFTER IMPLEMENTING MATRIX FACTORIZATION

	Amount of "0"	Sparsity (%)
Original Datasets	16.915	75
Non-Negative Matrix Factorization (NNMF)	2839	15
Probabilistic Matrix Factorization (PMF)	4206	20

In Table V we try to evaluate the application of both methods, Non-Negative Matrix Factorization (NNMF) and Probabilistic Matrix Factorization (PMF), by looking at the values of RMSE and MAE. We evaluate with a K-fold that is valued from 1 until 5 with the help of the Surprise library. Then we got an evaluation for NNMF, where RMSE was valued at 0.9717 until 0.9845 and MAE was valued at 0.8330 until 0.8408. For PMF, we got RMSE valued at 1.1758 until 1.1974 and MAE valued at 1.0884 until 1.1012.

TABLE V
EVALUATION OF BOTH MATRIX FACTORIZATION METHODS

	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
	RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
NNMF	0.9771	0.8346	0.9717	0.8330	0.9739	0.8349	0.9723	0.8351	0.9845	0.8408
PMF	1.1974	1.1012	1.1758	1.0884	1.1810	1.0932	1.1953	1.1005	1.1962	1.1004

From the evaluation of both Matrix Factorization methods, we can conclude that NNMF has better performance than PMF because it gets lower RMSE and MAE values. After we know that the NNMF method is better than PMF, we try to apply the NNMF method with Random Forest Regressor. The Non-Negative Matrix Factorization (NNMF) methods on the Random Forest Regressor (NNMF-RF) can be seen in Fig. 3 below.

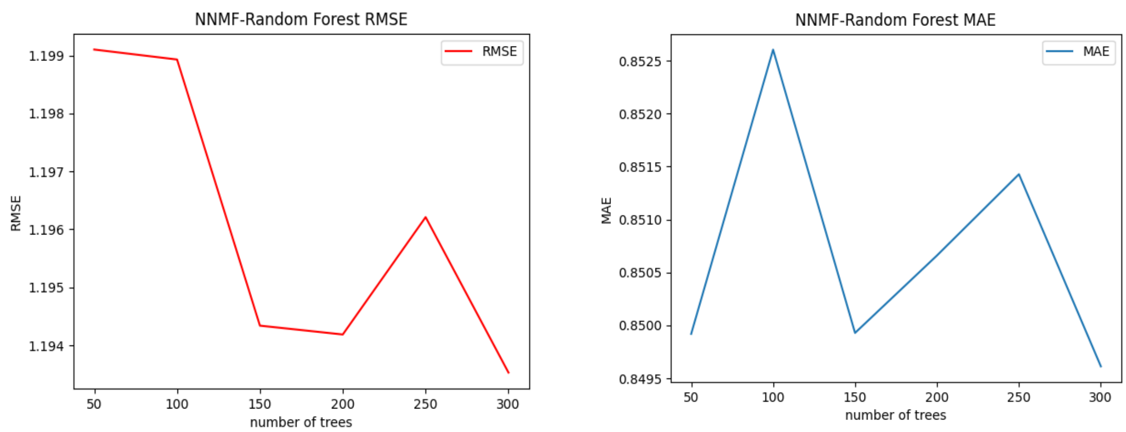

Fig. 3. NNMF-RF Evaluation Result

From Fig. 3, we can conclude that the performance of NNMF on the Random Forest Regressor gets the best average error when the n_estimator (number of trees) value is 300, which produces a RMSE equal to 1.1935 with an MAE equal to 0.8496. We try to apply hyperparameter tuning to random forests using GridSearchCV and RandomizeSearchCV. With the best parameters, we got GridSearchCV to give n_estimator value of 150 and a resulting RMSE of 1.0359 with MAE equal to 0.7641. While using RandomizeCV, the best parameter we got to give an n_estimator value of 200 and a resulting RMSE of 1.0777 with MAE equal to 0.7847. Table VI shows a comparison of the evaluation of NNMF-RF results.

TABLE VI
EVALUATION OF BOTH MATRIX FACTORIZATION METHODS

Method	**Random Forest**		**Random Forest GridSearchCV**		**Random Forest RandomizeSearchCV**	
	RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
NNMF-RF	1.1935	0.8496	1.0359	0.7641	1.0777	0.7847

After we apply Random Forest, we are trying to give rating predictions to new users who have watched a movie. As an example, a new user who is currently looking at a movie named "Words Bubble Up Like Soda Pop" will get a rating prediction with similarities to other movies as shown in Table VII below.

TABLE VII
RATING PREDICTION RESULT

Similarity	**Rating Prediction**	**Movie Title**
0.91	3.5	Silencer
0.85	0.7	Five Feet Apart
0.80	2	The Spiderwick Chronicles
0.62	0.5	The Hurricane Heist
0.3	0	G-Force

V. CONCLUSION

InIn this study, we have described our approach to the recommender system on the Netflix movie database from user reviews on Twitter. To give a prediction on the user, we apply one of the model-based collaborative filtering methods using Matrix Factorization, which focuses on Negative Matrix Factorization (NNMF) and Probabilistic Matrix Factorization (PMF) methods. Then we evaluate both methods by looking at the values of RMSE and MAE. We find that NNMF has a better result with lower RMSE and MAE than PMF, where RMSE is valued at 0.9717 and MAE is valued at 0.8330. After that, we apply NNMF (because it's the best method we have) with Random Forest to improve the results and evaluate them based on the average error (RMSE and MAE). The results show that NNMF-RF has an RMSE value of equal to 1.1935 with an MAE equal to 0.8496. After using hyperparameter tuning on Random Forest, it can improve the result of the error rate. But, after we apply random forest, it shows that using NNMF already gives the best results.

Therefore, future research can improve the performance of the recommender system by using larger datasets and experimenting with different Matrix Factorization Methods such as Hybrid MF, Deep Learning MF, etc., and combining them with Random Forest Regressor or any other regressor method for the better performance of the recommender system.

REFERENCES

[1] Das, Debashis & Chidananda, Himadri & Sahoo, Laxman. (2018). Personalized Movie Recommendation System Using Twitter Data. 10.1007/978-981-10-7871-2_33.
[2] AlSomaikhi, Noura & Alzamil, Zakarya. (2020). Twitter Users' Classification Based on Interest: Case Study on Arabic Tweets. International Journal of Information Retrieval Research. 10. 1-12. 10.4018/IJIRR.2020010101.
[3] Ortega, Fernando & Hernando, Antonio & Bobadilla, Jesus & Kang, Jeon-Hyung. (2016). Recommending Items to Group of Users using Matrix Factorization based Collaborative Filtering. Information Sciences. 345. 10.1016/j.ins.2016.01.083.
[4] Tan, Huiyi & Guo, Junfei & Li, Yong. (2008). E-learning Recommendation System. 430-433. 10.1109/CSSE.2008.305.
[5] Koren, Yehuda & Bell, Robert & Volinsky, Chris. (2009). Matrix factorization techniques for recommender systems. Computer. 42. 30-37.
[6] Nagarnaik, Paritosh & Thomas, A.. (2015). Survey on recommendation system methods. 2nd International Conference on Electronics and Communication Systems, ICECS 2015. 1603-1608. 10.1109/ECS.2015.7124857.
[7] Li, Xiang & Wang, Zhijian & Wang, Liuyang & Hu, Ronglin & Zhu, Quanyin. (2018). A Multi-Dimensional Context-Aware Recommendation Approach Based on Improved Random Forest Algorithm. IEEE Access. 6. 1-1. 10.1109/ACCESS.2018.2865436.
[8] Ilhami, Mirza & Suharjito, Suharjito. (2014). Film Recommendation Systems using Matrix Factorization and Collaborative Filtering. 2014 International Conference on Information Technology Systems and Innovation, ICITSI 2014 - Proceedings. 10.1109/ICITSI.2014.7048228.
[9] Gopi, Arepalli & Jyothi, R. & Narayana, V.Lakshman & Sandeep, K.. (2020). Classification of tweets data based on polarity using improved RBF kernel of SVM. International Journal of Information Technology. 1-16. 10.1007/s41870-019-00409-4.
[10] Hammou, Badr & Ait Lahcen, Ayoub & Mouline, Salma. (2019). An Effective Distributed Predictive Model with Matrix Factorization and Random forest for Big Data Recommendation systems. Expert Systems with Applications. 137. 10.1016/j.eswa.2019.06.046.
[11] Nedjati-Gilani, Gemma & Schneider, Torben & Hall, Matt & Cawley, Niamh & Hill, Ioana & Ciccarelli, Olga & Drobnjak, Ivana & Gandini Wheeler-Kingshott, Claudia & Alexander, Daniel. (2017). Machine learning based compartment models with permeability for white matter microstructure imaging. NeuroImage. 150. 10.1016/j.neuroimage.2017.02.013.
[12] Kywe, Su Mon & Lim, Ee-Peng & Zhu, Feida. (2012). A Survey of Recommender Systems in Twitter. SocInfo. 7710. 420-433. 10.1007/978-3-642-35386-4_31.
[13] Fakhri, Alif & Baizal, Abdurahman & Setiawan, Erwin. (2019). Restaurant Recommender System Using User-Based Collaborative Filtering Approach: A Case Study at Bandung Raya Region. Journal of Physics: Conference Series. 1192. 012023. 10.1088/1742-6596/1192/1/012023.
[14] Sun, Yueming & Zhang, Yi. (2018). Conversational Recommender System. DOI: https://doi.org/10.1145/3209978.3210002.
[15] Shiraishi, Yuya & Takama, Yasufumi. (2017). Proposal on matrix-based collaborative filtering using personal values. 55-60. 10.1109/IWCIA.2017.8203561.
[16] Ajesh, A & Nair, Jayashree & Jijin, P. (2016). A random forest approach for rating-based recommender system. 1293-1297. 10.1109/ICACCI.2016.7732225.
[17] Gunawardana, Asela & Shani, Guy. (2015). Evaluating recommender systems. 10.1007/978-1-4899-7637-6_8.
[18] Chai, Tianfeng & Draxler, R.. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?. Geosci. Model Dev.. 7. 10.5194/gmdd-7-1525-2014.
[19] Yoshua, I., & Bunyamin, H. (2021). Pengimplementasian Sistem Rekomendasi Musik Dengan Metode Collaborative Filtering. *Jurnal STRATEGI-Jurnal Maranatha*, *3*(1), 1-16.

[20] Kaur, Chhinder & Sharma, Anand. (2020). Twitter Sentiment Analysis on Coronavirus using Textblob.

[21] Mohamed Bakrey. (2021). Netflix_tv_Showing Analyiss, Version 6. Retrieved February, 2022 from https://www.kaggle.com/code/mohamedbakrey/netflix-tv-showing-analyiss/data.

[22] Loria, S. (2018). textblob Documentation. Release 0.15, 2.

[23] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

[24] Hug, N., (2020). Surprise: A Python library for recommender systems. Journal of Open Source Software, 5(52), 2174, https://doi.org/10.21105/joss.02174.