OPEN ACCESS
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 9, Issue. 1, April 2024. pp. 19-30
doi:10.34818/indojc.2024.9.1.908

Music Recommendation System Using Alternating Least Squares Method

Muhammad Rafi Irfansyah #1, Dade Nurjanah #2, Hani Nurrahmi #3

 # School of Computing, Telkom University
Jl. Telekomunikasi No. 1 Ters. Buah Batu Bandung 40257 Indonesia
[1] krafime@student.telkomuniversity.ac.id
[2] dadenurjanah@telkomuniversity.ac.id
[3] haninurrahmi@telkomuniversity.ac.id

Abstract

Music is not just entertainment, but it also has a positive impact on psychological well-being. The music landscape is generally dominated by millennials, especially in Indonesia. Music recommendation systems are becoming an important factor in offering songs that match users' preferences. Collaborative Filtering (CF), particularly the Alternating Least Squares (ALS) method, has become a popular solution for data sparsity problems in user-item interactions. Using the Precision@K metric, ALS provides the best results at a 50:50 data split ratio, 0.30225 for the Last FM dataset and 0.19742 for the Taste Profile dataset.

Further analysis shows that ALS is more effective on datasets with balanced data distributions, such as Last FM, than on datasets with noisier characteristics, such as Taste Profile. The main conclusion is that ALS is suitable for use on datasets with balanced data distributions and can provide more optimal recommendations. For further development, handling sparsity data on Taste Profile needs to be improved to improve the performance of the recommendation model. This illustrates the importance of adapting the model to the unique characteristics of each dataset to achieve more accurate music recommendations.

Keywords: alternating least squares, collaborative filtering, sparse, recommendation system, music

Abstrak

Musik bukan sekadar hiburan, melainkan juga memiliki dampak positif pada kesejahteraan psikologis. Lingkungan musik umumnya didominasi oleh kalangan milenial, terutama di Indonesia. Sistem rekomendasi musik menjadi faktor penting untuk menawarkan lagu yang sesuai dengan selera pengguna. Collaborative Filtering (CF), khususnya metode Alternating Least Squares (ALS), telah menjadi solusi populer untuk data pada masalah sparsity dalam interaksi pengguna-item. Dengan menggunakan metrik Precision@K, ALS memberikan hasil terbaik pada proporsi pembagian data 50:50, yaitu 0.30225 untuk dataset Last FM dan 0.19742 untuk dataset Taste Profile.

Analisis lebih lanjut menunjukkan bahwa ALS lebih efektif pada dataset yang memiliki distribusi data seimbang, seperti pada Last FM, daripada dataset dengan karakteristik yang lebih bising (noise), seperti Taste Profile. Kesimpulan utama adalah ALS cocok untuk digunakan pada dataset yang memiliki distribusi data yang seimbang dan dapat memberikan rekomendasi yang lebih optimal. Sementara untuk pengembangan lebih lanjut, penanganan data sparsity pada Taste Profile perlu ditingkatkan untuk meningkatkan performa model rekomendasi. Ini menggambarkan pentingnya penyesuaian model dengan karakteristik unik dari setiap dataset untuk mencapai rekomendasi musik yang lebih akurat.

Kata Kunci: alternating least squares, collaborative filtering, sparse, sistem rekomendasi, musik

Received on January, 2024. Accepted on April, 2024

I. Introduction

Music has a significant role in fulfilling a person's entertainment needs. Listening to music can not only create a positive mood, but also help reduce stress and prevent anxiety and depression disorders [1]. In addition, music has a positive impact on brain and nervous system performance by increasing brain concentration and cognitive abilities [2].

Statistics from the Global Consumer Survey in October 2019 show that music streaming users in Indonesia are dominated by millennials. As much as 45 percent are from the age of 25-34 years, followed by the age of 18-24 years by 28.6 percent, and the age of 35-44 years by 26.4 percent [3]. In this context, music recommendation systems play an important role in providing the best song choices from various genres to listeners [4]. The increasing use of these recommendation systems can make it easier for listeners to choose songs according to their preferences.

In their study titled "Of Spiky SVDs and Music Recommendation," Darius Afchar, Romain Hennequin, and Vincent Guigue explore the domain of music recommendation systems, focusing on the occurrence of spiky formations within recommendation dataset embeddings [15]. They examine matrix factorization techniques, particularly truncated Singular Value Decomposition (SVD), commonly used in collaborative filtering for music recommendation. Despite challenges in selecting the basis of similarity, they emphasize SVD's consistent performance and theoretical depth. Notably, they observe distinct spikes within embedding vectors, suggesting a self-organizing structure that may influence item retrieval and similarity calculations. This research provides insights into recommendation dataset structures, relevant for enhancing collaborative filtering algorithms. Meanwhile, Collaborative Filtering (CF), although preferred for its personalized recommendations, faces challenges like sparsity in user-item interaction data [5]. This limitation underscores the importance of studies like Afchar et al.'s, which offer potential solutions to improve CF-based recommendation systems, thereby enhancing recommendation accuracy and user experience.

The right solution to overcome the sparsity constraint is with the Alternating Least Squares (ALS) method in the Collaborative Filtering paradigm that becomes an effective choice [6]. ALS as a matrix factorization method is considered to be more efficient and scalable [7]. This method separates the rating matrix into two smaller matrices, the user matrix (U) and the item matrix (V). With an iterative approach, ALS alternately updates the user matrix and item matrix to minimize the square error between the observed ratings and the predicted ratings.

The implementation of ALS is able to improve the quality of recommendations by overcoming the sparsity problem. ALS models user preferences and item representations in latent space, resulting in more personalized and relevant recommendations. ALS can also handle data that only shows user interest, without requiring data that shows the level of user satisfaction.

The objective of this study is to enhance the user listening experience in the context of the growing music streaming user base in Indonesia, particularly among millennials. This will be achieved through the implementation of the ALS method as a music recommendation system, leveraging implicit datasets such as Last FM and Taste Profile [8,9]. By focusing on addressing the sparsity issue inherent in these datasets, the study aims to demonstrate the efficacy of ALS in providing more accurate and personalized music recommendations tailored to user preferences, thereby contributing to the improvement of music streaming services at scale.

II. Literature Review

A. Recommendation Systems

Recommendation systems are systems that use artificial intelligence techniques to provide suggestions for specific items to users based on their preferences, needs, or interests [10]. Recommendation systems can be applied in various fields, such as e-commerce, education, entertainment, healthcare, and others. Recommendation systems can improve the quality of the user experience, expand user choice, and increase user loyalty.

B. *Collaborative Filtering*

Collaborative filtering is a recommendation method based on the simple idea that users are more likely to prefer items that are recommended by friends with similar preferences [11]. The Collaborative Filtering process can be seen in Figure 1. In this context, the algorithm predicts a user's preference for an item by finding a group of other users who have similar preferences to the user to be recommended. This method is called Collaborative Filtering, which uses user rating data to generate recommendations. This method assumes that users who give high ratings to a particular item will also give high ratings to other items that are liked by users who are similar to them. Therefore, the predicted preference or rating for an item can be estimated from the ratin gs given by a group of similar users.

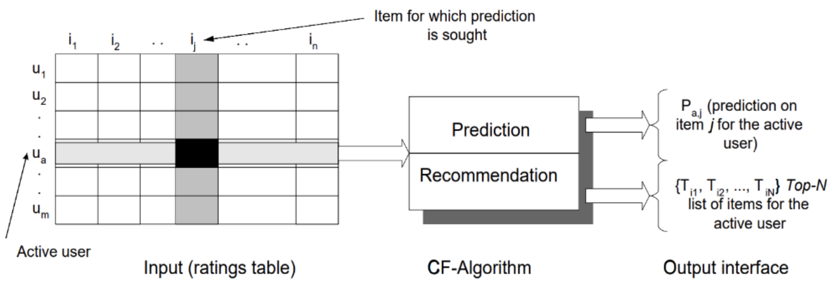

Fig. 1. Collaborative Filtering Process.

C. *Alternating Least Squares in Recommendation Systems*

The Alternating Least Squares (ALS) method is one of the matrix factorization approaches that is applied to recommendation systems, especially when user-item rating data has sparse characteristics [12]. In the development of recommendation systems, ALS is used to overcome the sparsity problem that arises when most of the entries in the user-item rating matrix do not have values. ALS factors the rating matrix into two smaller matrices, namely the user matrix (U) and the item matrix (V), where each matrix is represented as a vector of dimension k. A visualization of matrix factorization can be seen in Figure 2.

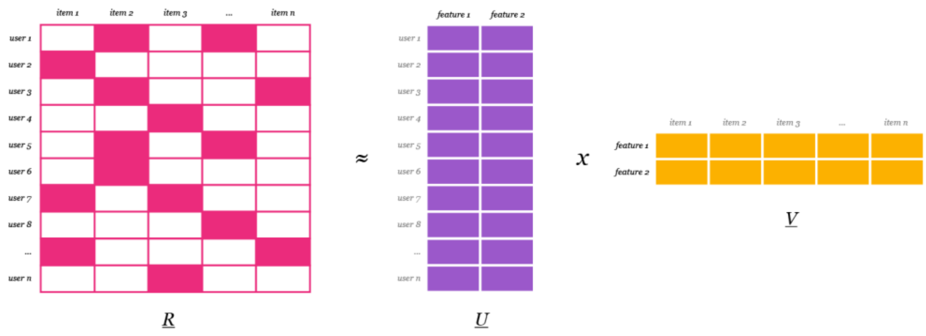

Fig. 2. ALS rating matrix factorization process.

The main process of ALS involves alternating iterations between updating the user matrix and the item matrix. When one set of vectors is fixed, the others are optimized by minimizing the square error between the observed ratings and the predicted ratings. This procedure is repeated until convergence or a specified iteration limit is reached.

The application of ALS in recommendation systems helps to generate more personalized and relevant recommendations by modeling user preferences and item representations in latent space. ALS is also often applied in the context of product or content recommendations in various platforms, such as Netflix. This platform uses ALS to recommend movies that are in line with the user's taste, based on the user's interaction data with the items.

D. Data Sparsity

Data sparsity is a problem that arises due to insufficient or very rare information. The issue of data sparsity is often encountered in various internet platforms and application systems with a large number of users, projects, or services, but very little relevant rating or historical information [13]. In the context of recommendations, this situation's data causes the system to be unable to depict user preferences or accurately calculate their interest similarities, leading to a decrease in data quality. Therefore, the accuracy of recommendation systems built with sparse data will decrease.

$$Sparsity\ (\%) = 100 \times \left(1 - \frac{The\ number\ of\ interacting\ domains}{The\ number\ of\ possible\ interactions\ in\ the\ matrix}\right) \quad (1)$$

In Formula (1), the sparsity percentage reflects the proportion of interactions that do not occur in the matrix. The higher the sparsity level, the less relevant information about user preferences is available. Therefore, careful handling of data sparsity is crucial to improving the quality and accuracy of the recommendation system.

III. Research Method

The first step in building a music recommendation system using the Alternating Least Squares (ALS) method is to prepare a music dataset that contains information about users, items, and interactions between users and items. In this study, the implicit Last FM and Taste Profile datasets were used. The application of ALS to these two datasets was carried out separately, but with the same development flow.

The music dataset is then processed by deleting irrelevant data and normalizing the data. After the data preprocessing process is complete, the dataset is divided into two parts, namely training data and test data. This data division is grouped into four data proportions. The training data is used to train the recommendation system model with ALS, while the test data is used to test the performance of the model. The ALS model is then tested using the Precision@K metric. Finally, further model evaluation is carried out to measure how accurate the built recommendation system is. The flow diagram can be seen in Figure 4.

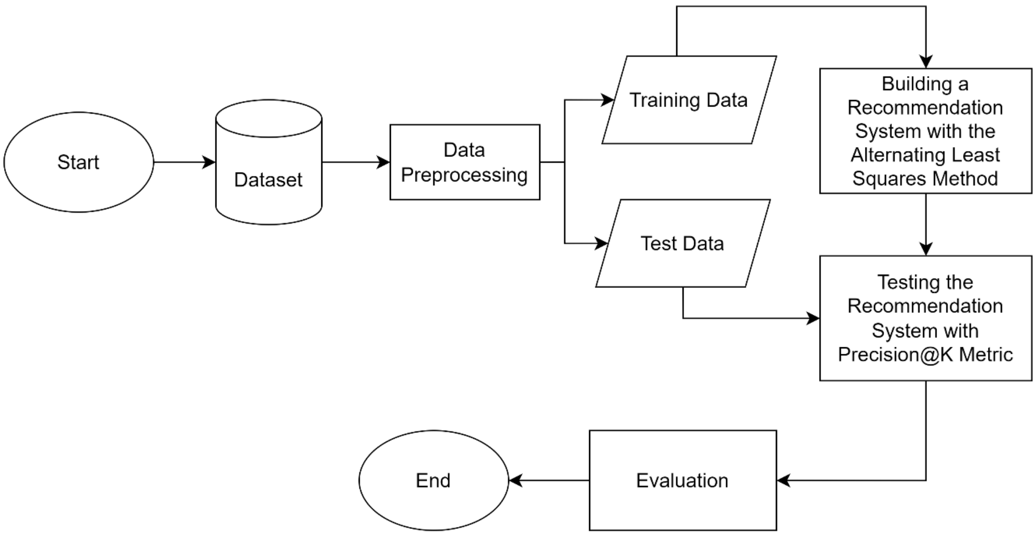

Fig. 4. System development flow with the ALS method.

A. Dataset Description

In this study, two main datasets are utilized, namely Last FM [8] and Taste Profile [9], both of which are music datasets. The Last FM dataset contains information about users, artists, and interactions, while Taste Profile includes data about users, tracks, and interactions. Although both provide information about music, these datasets are not related to each other. Statistics of both datasets are documented in Table I.

TABLE I
STATISTICS OF BOTH DATASETS

Dataset	Users	Items	Interactions	Sparsity
Last FM	358.868	292.385	17.535.605	99.9833%
Taste Profile	1.019.318	384.546	48.373.586	99.9877%

The portion used to train the model is the interaction part, which includes interaction data between users and items. The sparsity level reaches a high level, approaching 100%, indicating that most cells in the interaction matrix are empty. However, the filled (non-zero) portion of the matrix becomes the main focus in training the recommendation system model. Table II below represents sample data used to train the model.

TABLE II
SAMPLE DATA USED TO TRAIN THE MODEL

Dataset	Sample Data		
	UsersID	**ItemID**	**Interactions**
Last FM	0	19370	229.0
	0	19606	288.0
	358867	274818	39.0
Taste Profile	0	26214	2.0
	1	94412	1.0
	1019317	189466	56.0

As seen in Table II, the Last FM dataset has a more balanced distribution of the number of interactions compared to the Taste Profile dataset, where the Taste Profile dataset has many interactions with a value of 1. This differing distribution can affect the training process of the recommendation system model. Additionally, these two datasets do not have user overlap, meaning there are no same users between the Last FM and Taste Profile datasets. This is caused by the data structure used, where the UserID here aligns with the array of rows in the sparse matrix. Therefore, each UserID is considered unique in the context of that dataset. The presence or absence of user overlap can affect the recommendation system model's ability to generalize user preferences from various groups and determine appropriate recommendation policies.

B. Data Preprocessing

During the data preprocessing stage, the Taste Profile dataset requires additional effort because the 'track' column contains arrays containing track IDs, artists, albums, and track names. To obtain the required information, namely artist names and song titles, this array needs to be separated. The result of this separation is stored in the 'track' variable, which then becomes one of the features in the recommendation system.

After preparing the dataset with the obtained track information, the next step is to perform word weighting using the Best Match 25 (BM25) method. This method calculates word weights in documents based on frequency of occurrence and document length, using parameters K1 and B to control the impact of frequency and document length [14]. The purpose of this word weighting is to measure document relevance to a query, giving higher weights to words that rarely appear in the document collection but frequently appear in certain documents.

Once the BM25 weighting process is completed, the dataset is divided into two parts, namely training data and test data. The proportions of using training data and test data can be varied; in this study, the proportions used are 80:20, 70:30, 60:40, and 50:50. Training data is used to train the recommendation system model, while test data is used to evaluate the model's performance. However, the data splitting method used differs from the usual method, using masking to hide some values from the user-item interaction matrix. These hidden values are considered as zero or non-existent, and the model is asked to predict these missing values. This way, the research can measure the effectiveness of the model in handling data sparsity and making predictions on elements not observed in the dataset.

C. Building a Recommendation System with the Alternating Least Squares Method

The recommendation system was initially built by initializing the ALS Model. In this stage, the ALS model was created using the AlternatingLeastSquares module from the implicit library. The ALS model was defined with several key parameters, such as factors, regularization, alpha, and iterations. Factors indicate the number of latent space dimensions representing users and items, while regularization prevents overfitting. Alpha controls the scalar factor that strengthens the positive contribution of data, and iterations determine the number of cycles the model undergoes in the optimization process. This study used factors of 50, regularization of 0.1, alpha of 1.0, and 15 iterations. Additionally, this research enabled the calculation of loss at each iteration, which is the value of the error function minimized by the model. The formula for loss calculation can be seen in (2).

$$loss = \sum_{u} \sum_{i} C_{ui}(P_{ui} - X_u Y_i)^2 + \lambda(\|X_u\|^2 + \|Y_i\|^2) \tag{2}$$

The next step is the creation of the Epoch Callback, where the epoch_callback function is designed to record key information at each training iteration. The recorded information includes the epoch number, time taken, and loss value. This function serves as a callback called at the end of each iteration, aiming to help monitor the progress of the ALS model training and provide a deeper understanding of changes in loss values during the training process.

During the training phase, the ALS model was trained using the training data train_plays, which contains the number of user-item interaction rows that were previously separated. This is crucial for measuring time and loss during the training process to understand how efficient and accurate the model is in learning the data. Time indicates the duration taken by the model to perform one iteration, influenced by factors such as data size, the number of threads, and linear algebra operations used. Meanwhile, loss indicates the value of the error function that the model aims to minimize, measuring the difference between user preferences for items and preferences predicted by the model. These preferences are measured by confidence values, indicating the strength of the relationship between users and items. To train the model, the fit_model function is executed, using threads for parallel computations, which can enhance computational efficiency. The training process is monitored through the epoch_callback function, recording loss values at each iteration.

During the training process, the ALS model successfully achieved convergence with a reasonably efficient running time. The analysis of training results is depicted in Table III, showing training time and verified loss values for various data proportions.

TABLE III
RESULTS OF THE MODEL TRAINING

Dataset	Data Proportion	Running Time	Loss
Last FM	80:20	4 minutes 19 seconds	0.00305
	70:30	3 minutes 37 seconds	0.00278
	60:40	3 minutes 37 seconds	0.00249
	50:50	3 minutes 33 seconds	0.00218
Taste Profile	80:20	15 minutes 13 seconds	0.000284
	70:30	14 minutes 54 seconds	0.000284
	60:40	13 minutes 58 seconds	0.000284
	50:50	12 minutes 29 seconds	0.000284

The analysis in Table III indicates that the ALS model is capable of achieving convergence with training time efficiency ranging from 3 to 14 minutes, depending on the data proportion. These results reflect the adaptability of the ALS model to the characteristics of various datasets. In the Last FM dataset, the 80:20 data proportion requires a training time of 4 minutes 49 seconds with a loss value of 0.00305, while the 50:50 data proportion shows a shorter training time of 2 minutes 36 seconds with a loss value of 0.00218. A similar situation is observed in the Taste Profile dataset, where the 80:20 data proportion exhibits a loss value of 0.000284. The varied training times highlight the efficiency of the ALS model, while the low loss values indicate the model's ability to estimate user preferences well, especially in data lacking explicit ratings.

From these results, it can be analyzed that the ALS model is not only efficient in training with relatively short times but also capable of providing accurate predictions with small loss values. This signifies the model's ability to handle data sparsity issues effectively, making it a good choice for recommendation systems with incomplete data. The visualization of the decrease in loss values at each iteration during training can be observed in Figures 5 and 6.

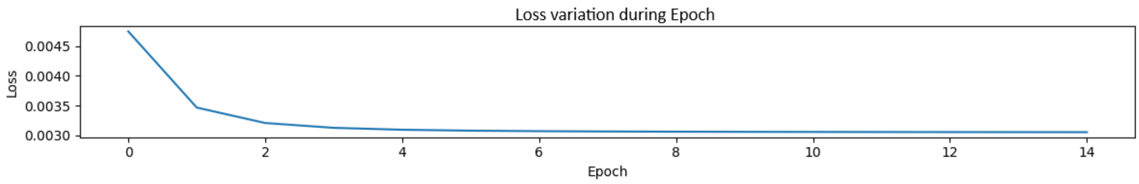

Fig. 5. Graph of loss variation during training with the Last FM dataset with an 80:20 data proportion.

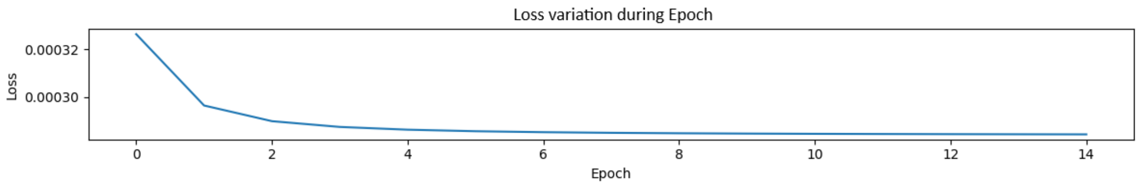

Fig. 6. Graph of loss variation during training with the Taste Profile dataset with an 80:20 data proportion.

D. Testing the Recommendation System with Precision@K Metric

Testing was conducted using the Precision@K metric, which measures the proportion of recommended and relevant items in a recommendation list of size k. This testing was applied to all data separation proportions. The metric works by comparing the top k items recommended by the model with the top k items desired by the user. If the recommended item is in the list of desired items, it is considered relevant. Precision@K is calculated using the following formula:

$$Precision@K = \frac{Number\ of\ relevan\ items\ in\ the\ top\ k\ recommendations}{k} \tag{3}$$

With the Precision@K metric, the quality of recommendations improves as the value approaches 1. This value is calculated from the average proportion of relevant items in the top 10 recommended items for all tested users (k = 10). The choice of k is based on default parameters commonly used in recommendation system evaluations.

E. Advanced Testing

This research tested the performance of the ALS model in providing music recommendations that align with user preferences, even when user data may be limited or incomplete. Two randomly unrelated users were chosen as the main subjects or target users. The recommendation system generated the top 5 lists of recommended artists and scores for each target user. These scores measure the proximity between the user vector and the recommended item vector. The recommendation results were then evaluated in two ways. First, by comparing them with the test data, which contains the actual preferences of the target users, to assess how well the recommendations align with the desires of the target users. Second, by calculating the average scores from the recommendation results issued by the system for each data separation proportion, i.e., 80:20, 70:30, 60:40, and 50:50, to determine which data proportion yields the best recommendation results for the target users. This aims to examine the influence of training and testing data proportions on the performance of the recommendation system.

IV. Results And Discussion

The results of testing using the Precision@K metric with various data proportion divisions can be observed in Table IV.

TABLE IV
TEST RESULTS USING PRECISION@K METRIC

Dataset	Data Proportion	Running Time	Precision@K
Last FM	80:20	11 minutes 21 seconds	0.16955
	70:30	11 minutes 43 seconds	0.21011
	60:40	10 minutes 21 seconds	0.25894
	50:50	11 minutes 12 seconds	0.30225
Taste Profile	80:20	39 minutes 53 seconds	0.13122
	70:30	39 minutes 37 seconds	0.15477
	60:40	39 minutes 42 seconds	0.17682
	50:50	40 minutes 30 seconds	0.19742

The Precision@K values are obtained from the average proportion of relevant items in the top K items. The K value used in the study is 10. From Table IV, it is evident that the model's performance, measured by the Precision@K metric, varies depending on the dataset type and data proportion division. The recorded computation time also provides an overview of the model's efficiency in processing data.

Furthermore, for further evaluation, the research was conducted on two random users, UserID 76 and UserID 332215, in the Last FM dataset. Meanwhile, in the Taste Profile dataset, the study focused on UserID 30 and UserID 200. However, in this specific context, further analysis was directed towards the Last FM dataset, considering UserID 76. The recommendation results from various data proportions were then compared with the test data containing actual preferences. The comparison results can be seen in Table V.

TABLE V
COMPARISON RESULTS OF RECOMMENDATION RELEVANCE WITH TEST DATA FOR USERID 76

Last FM Dataset							
80:20		70:30		60:40		50:50	
Artists	**Listened**	**Artists**	**Listened**	**Artists**	**Listened**	**Artists**	**Listened**
metallica	Yes	system of a down	Yes	metallica	Yes	system of a down	Yes
guns n' roses	No	rammstein	Yes	system of a down	Yes	apocalyptica	Yes
guns n roses	Yes	red hot chili peppers	Yes	iron maiden	Yes	nightwish	Yes
ac/dc	No	iron maiden	Yes	guns n' roses	No	tenacious d	No
the offspring	No	guns n' roses	No	apocalyptica	Yes	cradle of filth	No

Then, a comparison of the average scores from the recommendation results at each data proportion was also conducted to evaluate the model's performance on UserID 76 in the Last FM dataset. The scores generated here are from the Top-K Recommendation for ALS, measured as scores indicating how close the user vector is to the recommended item vector. The Top-K recommendation for ALS is the process of selecting the top K items with the highest scores for each user and considering them as the most relevant items to the user's preferences.

TABLE VI
COMPARISON OF AVERAGE SCORES BASED ON DATA PROPORTION DIVISIONS FOR USERID 76

Last FM Dataset							
80:20		70:30		60:40		50:50	
Artists	**Score**	**Artists**	**Score**	**Artists**	**Score**	**Artists**	**Score**
metallica	1.180427	system of a down	1.068468	metallica	1.266361	system of a down	1.069707
guns n' roses	1.060235	rammstein	1.036252	system of a down	1.117492	apocalyptica	0.922644
guns n roses	1.003362	red hot chili peppers	1.012558	iron maiden	1.112368	nightwish	0.916550
ac/dc	0.950043	iron maiden	1.000992	guns n' roses	1.074234	tenacious d	0.861920
the offspring	0.902609	guns n' roses	0.981075	apocalyptica	1.029525	cradle of filth	0.855694
Average							
1.0193352		1.019869		1.119996		0.925303	

Based on Table VI, the 60:40 data proportion shows the highest average score, namely 1.119996, for UserID 76 in the Last FM dataset. This indicates that this data proportion is the most optimal in dividing the training and testing data, as this score is influenced by several relevant items for UserID 76 found in the training data. Conversely, other data proportions yield lower average scores, indicating that the model has not achieved the desired level of accuracy in providing recommendations for UserID 76.

The overall evaluation of the recommendation model testing results was conducted through the Precision@K metric on two main datasets, namely Last FM and Taste Profile. Table IV provides an overview of the variation in model performance influenced by dataset type and data proportion division. In the Last FM dataset, it was found that the 50:50 data proportion achieved the highest Precision@K value, indicating that a balanced distribution between training and testing data has a positive impact on model accuracy. The same result is observed in the Taste Profile dataset.

To evaluate the model's performance more deeply, this research focused on UserID 76 in the Last FM dataset. Table V illustrates that the 80:20 data proportion resulted in the lowest number of relevant items, indicating that the model is less effective in learning patterns in the testing data with that data proportion. Furthermore, the average scores in Table VI confirm that the 60:40 data proportion provides the highest score, highlighting the importance of selecting an optimal data proportion division to enhance model performance.

However, it should be noted that in the Taste Profile dataset, noisy characteristics make the model struggle to provide relevant recommendations. This mismatch is caused by a larger number of users and items, as well as the high proportion of 1 values in the dataset, which tends to influence the model's recommendations. This indicates that, regardless of success in the Last FM dataset, each dataset poses unique challenges that require different approaches in developing a recommendation model.

V. Conclusion

The results of testing the recommendation model with the ALS method on the Last FM and Taste Profile datasets provide in-depth insights. The proportion of data division plays a crucial role in the model's performance, with the 60:40 data proportion showing the highest performance with an average score of 1.119996 for UserID 76 in the Last FM dataset using the Top-K Recommendation approach. This score represents the closeness between the user's preferences and the recommended items. This indicates that this data proportion is optimal in dividing the training and testing data, and the proper selection of data proportion can enhance the model's accuracy.

In the Last FM dataset, the 50:50 data proportion achieves the highest Precision@K value, indicating that a balanced distribution between training and testing data can improve the model's accuracy. This finding reinforces the conclusion that ALS is suitable for use with datasets that have a balanced distribution between training and testing. However, further analysis on the Taste Profile dataset reveals challenges faced by the model due to noisy characteristics and the high proportion of 1 values in the dataset. Therefore, ALS is more suitable for use with datasets like Last FM that have a more balanced data distribution and are less influenced by dominating values.

Testing on UserID 76 on Last FM highlights the importance of proper data proportion division for specific users. The 80:20 data proportion results in the lowest number of relevant items, while the 60:40 data proportion provides the highest score, indicating that different data proportions can yield significant results depending on specific user preferences.

In further research development, addressing data sparsity in the Taste Profile dataset needs improvement to enhance the performance of the recommendation model. This analysis supports the view that ALS can deliver optimal results in datasets with a balanced data distribution, such as Last FM, and emphasizes the importance of adapting the model to the unique characteristics of each dataset to produce more accurate recommendations.

Acknowledgment

I would like to express my heartfelt gratitude to my academic advisor for their guidance and support throughout the completion of this thesis. Their insights and feedback have been invaluable in shaping the direction of my research. Additionally, I extend my appreciation to the Telkom University for providing the necessary resources. I am also thankful for the encouragement and understanding from my family and friends, whose support has been a constant source of motivation. Overall, I am thankful to all those who have contributed to this academic journey in various ways.

REFERENCES

[1] A. Harismi, "Manfaat Mendengarkan Musik yang Tidak Disangka-Sangka," [Online]. Available: https://www.sehatq.com/artikel/mendengarkan-musik-dan-manfaatnya-bagi-kondisi-emosional. [Accessed 16 May 2023].
[2] Manfaat.co.id, "14 Manfaat Mendengarkan Musik Instrumental bagi Kesehatan," [Online]. Available: https://manfaat.co.id/manfaat-mendengarkan-musik-instrumental. [Accessed 16 May 2023].
[3] W. F. I. Putri, "Tren Mendengarkan Musik di Indonesia Berubah, Streaming Makin Populer," [Online]. Available: https://www.medcom.id/hiburan/musik/3NO9XMWk-tren-mendengarkan-musik-di-indonesia-berubah-streaming-makin-populer. [Accessed 16 May 2023].
[4] M. F. Mahardika, "Mengenal Industri Musik dan Produksi Dibaliknya," [Online]. Available: https://www.its.ac.id/news/2021/10/18/mengenal-industri-musik-dan-produksi-dibaliknya/. [Accessed 16 May 2023].
[5] A. Fareed, S. Hassan, S. B. Belhaouari, and Z. Halim, "A collaborative filtering recommendation framework utilizing social networks," Machine Learning with Applications, vol. 14, p. 100495, 2023. DOI: 10.1016/j.mlwa.2023.100495.
[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, pp. 30–37, 2009. DOI: 10.1109/MC.2009.263.
[7] J. Chen et al., "Efficient and Portable ALS Matrix Factorization for Recommender Systems," in 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017, pp. 409–418. DOI: 10.1109/IPDPSW.2017.91.
[8] Last.fm, "Last.fm API," [Online]. Available: https://www.last.fm/api. [Accessed 11 September 2023].
[9] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, "The MSD," in Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011).
[10] C. C. Aggarwal, Recommender Systems: The Textbook, 1st ed. Springer Cham, 2016. DOI: 10.1007/978-3-319-29659-3.
[11] S. Martin, J. I. Sihotang, and B. Jonathan, "Mesin Rekomendasi menggunakan Algoritma Alternating Least Square (ALS) pada Goodreads," Jurnal CoreIT, vol. 6, no. 2, p. 79, 2020. DOI: 10.24014/coreit.v6i2.11578.
[12] S. Gosh et al., "Recommendation System for E-commerce Using Alternating Least Squares (ALS) on Apache Spark," in P. Vasant, I. Zelinka, and G. W. Weber (Eds.), Intelligent Computing and Optimization, Springer International Publishing, 2021, pp. 880–893. DOI: 10.1007/978-3-030-68154-8_75.
[13] Z. Zhang, T. Peng, and K. Shen, "Overview of Collaborative Filtering Recommendation Algorithms," IOP Conference Series: Earth and Environmental Science, vol. 440, no. 2, p. 022063, 2020. DOI: 10.1088/1755-1315/440/2/022063.
[14] "Pengelompokan Hasil Pencarian Skripsi Berbahasa Indonesia Menggunakan Metode DBSCAN dengan Pembobotan BM25," Jurnal Teknologi Informasi Dan Ilmu Komputer, vol. 10, no. 4, pp. 781-790, 2023. DOI: 10.25126/jtiik.20241046899.
[15] D. Afchar, R. Hennequin, and V. Guigue, "Of Spiky SVDs and Music Recommendation," ACM, New York, NY, USA, 2023, pp. 1-10.