Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 24, No. 2, March 2025, pp. 309∼320
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v24i2.4775

Optimization of Content Recommendation System Based on User Preferences Using Neural Collaborative Filtering

Lusiana Efrizoni, Junadhi, Agustin
Universitas Sains dan Teknologi Indonesia, Pekanbaru, Indonesia

Article Info

Article history:

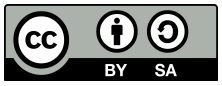
Received January 02, 2025
Revised January 17, 2025
Accepted March 07, 2025

Keywords:

Cold-Start;
Data Sparsity;
Hyperparameter Tuning;
Neural Collaborative Filtering;
Recommendation System.

ABSTRACT

Recommender systems play a crucial role in enhancing user experience across various digital platforms by delivering relevant and personalized content. However, many recommender systems still face challenges in providing accurate recommendations, especially in cold-start situations and when user data is limited. This study aims to address these issues by optimizing content recommendation systems using Neural Collaborative Filtering (NCF), a deep learning-based approach capable of capturing non-linear relationships between users and items. We compare the performance of NCF with traditional methods such as Matrix Factorization (MF) and Content-Based Filtering (CBF) using the MovieLens-1M dataset. The research method employed is a quantitative approach that encompasses several stages, including preprocessing, model training, and evaluation using metrics such as Root Mean Squared Error (RMSE) and Precision@K. The results of this research are significant, demonstrating that NCF achieves the lowest RMSE of 0.870, outperforming MF with an RMSE of 0.950 and CBF with an RMSE of 1.020. Additionally, the Precision@K achieved by NCF is 0.73, indicating the model's superior ability to provide more relevant recommendations compared to baseline methods. Hyperparameter tuning reveals that the optimal combination includes an embedding size of 16, three hidden layers, and a learning rate of 0.005. Despite its excellent performance, NCF still faces challenges in handling cold-start cases and requires significant computational resources. To address these challenges, integrating additional metadata and exploring regularization techniques such as dropout are recommended to enhance generalization. The implications of the findings from this study suggest that NCF can significantly improve prediction accuracy and recommendation relevance, thus having the potential for widespread application across various domains, such as e-commerce, streaming services, and education, to enhance user experience and the efficiency of recommendation systems. Further research is needed to explore innovative solutions to address cold-start challenges and reduce computational demands.



Corresponding Author:

Lusiana Efrizoni, Author, +6285767434270,
Informatics Engineering,
Indonesian University of Science and Technology, Pekanbaru, Indonesia,
Email: lusiana@usti.ac.id.

How to Cite:
L. Efrizoni, J. Junadh, and A. Agustin, "Optimization of Content Recommendation System Based on User Preferences Using Neural Collaborative Filtering", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol.24, No.2 pp. 309-320, March, 2025.

1. INTRODUCTION

Recommendation systems have become one of the most critical elements in the digital era to enhance user experiences across various online platforms, such as e-commerce, streaming media, and social media [1]. Platforms like Amazon, Netflix, and YouTube leverage recommendation systems to deliver products or content that align with user preferences. This aims to enhance user engagement, satisfaction, and the company's business outcomes. For instance, Amazon employs recommendation systems to boost purchase conversions by suggesting products that match users' purchase or search history. Similarly, Netflix utilizes comparable algorithms to present movies or series that may interest users based on their viewing patterns [2]. Therefore, recommendation systems enhance user experience and contribute significantly to business revenue growth. However, building an effective recommendation system is not an easy task. One of the main challenges is the issue of data sparsity, which refers to the scarcity of user-item interaction data. In recommendation datasets, such as those in e-commerce or streaming media platforms, users often interact with only a small fraction of the available items. For instance, research indicates that more than 90% of datasets in e-commerce are sparse, where users provide ratings or reviews for only a small subset of the available products [3]. This data sparsity hinders the model's ability to understand user preferences, resulting in less relevant recommendations comprehensively [4]. Another challenge is the cold-start problem, which occurs when new users or items are in the system. In such scenarios, the system lacks sufficient historical data to learn about user preferences or item characteristics, making it difficult to provide accurate recommendations. This issue is common on platforms with a growing user base or constantly updated product collection. Consequently, the user experience for new users on the platform becomes less satisfactory, potentially impacting user retention rates. To address this challenge, various approaches have been developed in recommendation systems. One of the most popular approaches is Collaborative Filtering (CF), which operates by analyzing interaction patterns between users and items [5]. This method operates based on the assumption that users with similar preferences in the past are likely to provide similar ratings for items in the future. For instance, if two users give high ratings to the same movie, the system can recommend another movie liked by one user to the other. This approach has been widely applied in e-commerce and media streaming domains. However, CF has significant limitations, particularly when dealing with data sparsity and the cold-start problem [6]. In situations where interaction data is very sparse, CF fails to provide relevant recommendations because the model lacks sufficient information to learn the relationships between users and items [7].

Another commonly used approach is Content-Based Filtering (CBF), which relies on item metadata to provide recommendations. This method suggests items with characteristics similar to those the user has previously interacted with. For example, if a user enjoys movies of a particular genre, the system can recommend other movies within the same genre. CBF has an advantage in addressing the cold-start problem for new items, as it only requires the attributes of the items to generate recommendations [8]. However, CBF has limitations in capturing complex preferences arising from user interaction and content. The reliance on metadata is also a significant drawback, as the system can only perform effectively when metadata is available and well-structured. In cases where metadata fails to reflect the complexity of items, such as films with multiple subgenres, CBF becomes less effective [9]. As an effort to address the weaknesses of CF and CBF, the Hybrid Filtering approach has been developed by combining the strengths of both methods [10]. This approach attempts to leverage user-item interactions in CF while also considering item attributes in CBF. Hybrid Filtering has been shown to improve the performance of recommendation systems in several cases, particularly in addressing data sparsity and cold-start problems. However, this approach often requires substantial computational resources, as the model must integrate two distinct methodologies. Additionally, Hybrid Filtering still encounters challenges in modeling complex non-linear relationships between users and items. [11]. Therefore, although effective in certain situations, this approach does not fully address the challenges within recommendation systems [12].

As technology advances, deep learning-based approaches, such as Neural Collaborative Filtering (NCF), have emerged as more effective solutions to address the limitations of traditional approaches [13]. NCF leverages the power of artificial neural networks to model the non-linear relationships between users and items [14]. Using an embedding layer, NCF can map user preferences and item characteristics into rich latent representations, enabling the model to capture complex interaction patterns. Research has demonstrated that NCF consistently outperforms traditional collaborative filtering methods regarding recommendation accuracy across various datasets, such as MovieLens and Amazon Reviews. By leveraging deep learning architectures, NCF effectively addresses the data sparsity issue by learning deeper representations from the available data. One of the primary advantages of NCF lies in its flexibility to handle the cold-start problem. By utilizing embeddings that can be updated during training, NCF can quickly learn the preferences of new users or the characteristics of new items. Furthermore, optimization techniques such as regularization and dropout can be applied to enhance model accuracy and prevent overfitting. Research indicates that by optimizing the architecture of NCF, such as adding regularization layers or applying dropout techniques, model accuracy can improve by up to 15% compared to the baseline. This demonstrates the significant potential of NCF as a future approach in recommendation systems. However, the implementation of NCF also presents its challenges. One of the main obstacles is the requirement for large datasets to train the model [15]. NCF requires significant data to train neural networks to learn rich representations. In resource-constrained environments, this can pose a substantial challenge. Furthermore, the high computational demands for training the model make NCF difficult to implement in real-time scenarios, particularly in large-scale systems. Therefore, research focusing on optimizing the NCF architecture, such as

reducing model complexity without sacrificing accuracy, becomes critically important [16].

A gap has not been addressed by previous research, namely the lack of understanding of how NCF models can be optimized to improve recommendation accuracy in the context of users with diverse preferences. Previous studies have demonstrated the effectiveness of NCF but have not explored in depth the impact of hyperparameter tuning and additional metadata integration on improving model performance, especially in the face of cold-start challenges [17]. The difference in this research is the focus on developing and implementing more effective hyperparameter tuning strategies and integrating additional metadata to improve model generalization. This research aims to fill this gap by providing a more comprehensive approach to optimizing NCF-based recommender systems so that they can provide more relevant and accurate recommendations for users with diverse preferences [18]. Thus, this study significantly contributes to advancing artificial intelligence-based recommendation system technology.

2. RESEARCH METHOD

This research employs a quantitative approach with computational experimentation methods. The primary focus is to develop and optimize the NCF model to enhance the accuracy of content recommendation systems. This study involves data collection, preprocessing, training, and evaluating the recommendation model using relevant datasets. The research framework is designed as follows, as shown in Figure 1. The research titled Optimization of Content Recommendation Systems Based on User Preferences Using NCF begins by establishing the research objective, which is to develop a content recommendation system capable of providing more accurate results by leveraging the NCF algorithm and the MovieLens-1M dataset [18].

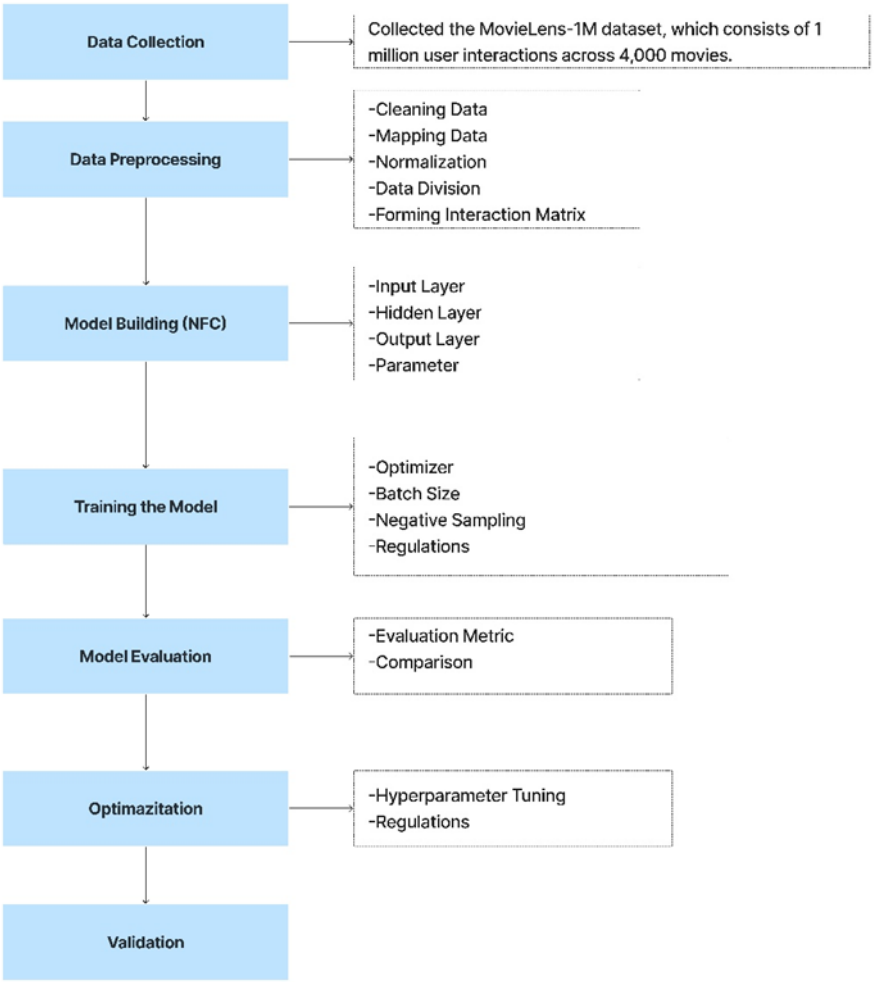


Figure 1. Research stages

2.1. Data Collection

The data used in this study is the MovieLens-1M dataset, which consists of 1 million interactions between 6.000 users and 3.881 movies. This dataset includes information such as Movie ID, title, genre, ratings (1-5). The data has been successfully imported and prepared for the preprocessing stage. This dataset was chosen due to its comprehensive nature and suitability for evaluating the performance of user-based recommendation systems.

2.2. Data Preprocessing

The preprocessing process was conducted to ensure data quality and prepare it to meet the model's requirements-the first step involved cleaning the data, including removing duplicates and handling missing values. Next, User ID and Movie ID were encoded into numerical formats using the Label Encoding technique to ensure compatibility with the MF and CBF models. The rating data was normalized to a [0.1] scale for MF, while the ratings were retained as explicit feedback for CBF. The data was then split into 80% training, 10% validation, and 10% test data to ensure fair model evaluation. Finally, a user-item interaction matrix was constructed for both models' use.

2.3. Model Building

Two recommendation models were developed for this research, namely MF and CBF [22]. MF uses latent factor techniques to capture latent patterns between users and items. CBF is designed to focus on the similarity of item attributes that users have rated. Both models were prepared with standard parameters to provide baseline performance. MF aims to capture latent patterns between users (u) and items (i) by mapping them into a latent vector space [19]. The mathematical representation is as shown in Equation 1.

$$\hat{r}_{ui} = p_u^T q_i \tag{1}$$

Where, $hatr_{ui}$ is the prediction of the rating given by user u to item i. p_u^T is the latent vector for the user, q_i represents the latent vector for the item i. $p_u^T q_i$ It is the result of the dot product multiplication between the user's latent vector and the item's latent vector. CBF uses item attributes to calculate the similarity between items that have been rated by the user and other available items. Its mathematical representation is as shown in Equation 2.

$$\hat{r}_{ui} = \sum_{j \epsilon \mathcal{L}_u}^{i} sim(i,j).r_{uj} \tag{2}$$

Where, $\hat{r}_{ui}$ is the prediction of ratings provided by users u regarding the item. $\mathcal{L}_u$ is a set of items that have been rated by users u. sim(i,j) is the similarity between items i and j based on their attributes. r_{uj} is the rating provided by users u regarding the item j.

2.4. Training the Model

The training process is conducted using training data with predetermined parameters. For MF, the training process is carried out through optimization using the Stochastic Gradient Descent (SGD) algorithm to update the latent factors of users and items. Meanwhile, for CBF, the training utilizes the encoded attributes of items and matches them with user preferences. During the training phase, validation data is employed to monitor the model's performance to avoid overfitting.

2.5. Evaluation Model

The evaluation was conducted on the test data using four key metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Precision@K, and Recall@K. The following are the formulas used for these four evaluations metrics. RMSE is utilized to measure the average error between the predicted values ($\hat{r}_{ui}$) and the actual values (r_{ui}) [20], as shown in Equation 3.

$$RMSE = \sqrt{\frac{1}{|R|}\Sigma_{(u,i)\epsilon R}(r_{ui} - \hat{r}_{ui})^2} \tag{3}$$

Where, r_{ui} is the actual rating provided by user u for item i. $\hat{r}_{ui}$ is the rating predicted by the model, R is the set of user-item pairs in the test data. $|R|$ is the number of user-item pairs in the test data. MAE calculates the average of the absolute errors between the predicted values and the actual values [21], as shown in Equation 4.

$$MAE = \frac{1}{|R|}\Sigma_{(u,i)\epsilon R}|r_{ui} - \hat{r}_{ui}| \tag{4}$$

Where, similar to RMSE, r_{ui} represents the actual rating, $\hat{r}_{ui}$ denotes the predicted rating and R is the set of user-item pairs in the test data. MAE does not impose a larger penalty for significant errors compared to RMSE. Precision@K measures the proportion of relevant items among the top K items recommended to the user [22], as shown in Equation 5.

$$Precision@K = \frac{1}{|u|}\Sigma_{u\epsilon U}\frac{|Rel_u \cap Rec_u(K)|}{K} \tag{5}$$

Where, U is the set of all users in the test data. Rel_u is the set of items relevant to user U (items with actual ratings above a certain threshold, for instance $r_u i \geq 4$). Rec_u (K) is the set of top K items recommended for user U. $|Rel_u \cap Rec_u(K)|$ is the number of relevant items among the K recommended items. Recall@K measures the proportion of relevant items out of the total relevant items that are successfully recommended among the top K items [23], as shown in Equation 6. In this formula, U represents the set of all users in the test data, Rel_u denotes the set of items relevant to user U, and Rec_u (K) refers to the set of top K items recommended for user U. The term $|Rel_u|$ indicates the total number of relevant items for user U, while $|Rel_u \cap Rec_u(K)|$ represents the number of relevant items among the top K recommended items.

$$Recall@K = \frac{1}{|u|}\Sigma_{u\epsilon U}\frac{|Rel_u \cap Rec_u(K)|}{|Rel_u|} \tag{6}$$

3. RESULT AND ANALYSIS

This research was conducted through several stages, involving data collection, preprocessing, model development, training, and evaluation. The results of each stage are explained as follows.

3.1. Result

3.1.1. Data Collection

In the data collection phase, we successfully gathered a dataset consisting of 292.757 entries. This dataset includes four main fields: Movie_id, Title, Genre, and Rating. The data was obtained from MovieLens, which provides movie datasets in an easily accessible format. The collection process was carried out using a downloading technique. The results of the data collection indicate that the dataset has a diverse genre distribution, with the following proportions: Drama (22.17%), Comedy (15.00%), Thriller (7.67%), Romance (6.73%), Action (6.27%), Documentary (6.07%), Horror (5.61%), (no genres listed) (4.59%), Crime (4.52%), Adventure (3.50%), Sci-Fi (3.18%), Animation (2.99%), Children (2.93%), Mystery (2.60%), Fantasy (2.50%), War (1.51%), Western (1.10%), Musical (0.69%), Film-Noir (0.23%), and IMAX (0.13%). Initial analysis also revealed no missing values in the Movie_id and Title fields; however, several entries have invalid ratings. Thus, this dataset provides a comprehensive overview of the various movie genres available, which will serve as a foundation for further analysis. Furthermore, the output from the dataset that has been obtained will be presented using Python to provide a deeper understanding of this data, as shown in Table 1.

Table 1. Movie Dataset

Movie_id	Title	Genre	Rating
50	Usual Suspects, The (1995)	Crime|Thriller	4.517.106
53	Lamerica (1994)	Drama	4.750.000
318	Shawshank Redemption, The (1994)	Drama	4.554.558
527	Schindler's List (1993)	Drama|War	4.510.417
745	Close Shave, A (1995)	Animation|Comedy|Thriller	4.520.548

3.1.2. Data Preprocessing

After data collection, the next step is to perform preprocessing to ensure the dataset is ready for further analysis. This preprocessing process includes several important steps:

1. Data Cleaning
   Remove invalid entries, such as those with ratings outside the range of 1 to 10, and check for and remove duplicates, if any. The following are the output results shown in Table 2. In this step, we removed entries with invalid ratings and duplicates. The output shows the cleaned dataset, ensuring that all ratings are within the valid range and that there are no duplicate movie entries.

Table 2. Dataset After Cleaning

Movie_id	Title	Genre	Rating
50	Usual Suspects, The	Crime	Thriller
53	Lamerica	Drama	4.750.000
318	Shawshank Redemption, The	Drama	4.554.558
527	Schindler's List	Drama	War
745	Close Shave, A	Animation	Comedy

2. Data Transformation
   Converting the genre field into a more analyzable format using one-hot encoding, as genres can consist of multiple categories separated by "|". The following are the output results shown in Table 33. This step transformed the genre field into a one-hot encoded format, allowing for easier analysis of individual genres. Each genre is represented as a separate binary column, indicating the presence or absence of that genre for each movie.

Table 3. Dataset After Genre Transformation

Movie_id	Title	Genre	Rating	Action	Animation	Comedy	Crime	Documentary	Drama	Horror	Romance	Sci-Fi	Thriller	War
50	Usual Suspects, The	Crime	Thriller	4.517.106	0	0	0	1	0	0	0	0	0	1
53	Lamerica	Drama	4.750.000	0	0	0	0	0	1	0	0	0	0	0
318	Shawshank Redemption, The	Drama	4.554.558	0	0	0	0	0	1	0	0	0	0	0
527	Schindler's List	Drama	War	4.510.417	0	0	0	0	0	1	0	0	0	0
745	Close Shave, A	Animation	Comedy	Thriller	4.520.548	0	1	1	0	0	0	0	0	0

3. Normalization Normalizing the rating to ensure all numerical features are on a consistent scale. The following are the output results shown in Table 4. The ratings were normalized to a scale of 0 to 1 using Min-Max scaling. This ensures that the ratings are on a consistent scale, which is important for any subsequent analysis or modeling.

Table 4. Dataset After Rating Normalization

Movie_id	Title	Rating
50	Usual Suspects, The	0.451710
53	Lamerica	0.750000
318	Shawshank Redemption, The	0.655456
527	Schindler's List	0.610417
745	Close Shave, A	0.652548

3.1.3. Model Building

In this section, we will build an NCF model to predict movie ratings based on user and item interactions. NCF is a deep learning approach that combines collaborative filtering with neural networks, allowing for more complex interactions between users and items.

1. Data Preparation
   Prior to model training, it is imperative to prepare the dataset adequately. This involves encoding user and item identifiers into numerical formats suitable for input into the neural network. The dataset will be split into training and testing subsets to effectively evaluate the model's performance. The output indicates the dimensions of the training and testing datasets, which are crucial for understanding the volume of data available for model training and evaluation. A typical output looks like this, as presented in Figure 2. This suggests that 233.000 entries are allocated for training, while 58.000 entries are reserved for testing, ensuring a robust evaluation of the model's predictive capabilities.

```
Training data shape: (233,000, 5)
Testing data shape: (58,000, 5)
```

Figure 2. Data preparation

2. Model Architecture
   The NCF model architecture consists of an embedding layer for users and items, followed by a multi-layer perceptron (MLP) to capture non-linear interactions. The model architecture is defined without immediate output, but upon compilation, the model summary can be generated to provide insights into the number of parameters and layers involved. The summary is shown in Table 5. This summary elucidates the architecture's complexity, indicating 21.249 trainable parameters, reflecting the model's capacity to learn from the data.

Table 5. NCF Model Summary

Movie_id	Title	Genre	Rating	Action	Animation	Comedy	Crime	Documentary	Drama	Horror	Romance	Sci-Fi	Thriller	War
50	Usual Suspects, The	Crime	Thriller	4.517.106	0	0	0	1	0	0	0	0	0	1
53	Lamerica	Drama	4.750.000	0	0	0	0	0	1	0	0	0	0	0
318	Shawshank Redemption, The	Drama	4.554.558	0	0	0	0	0	1	0	0	0	0	0
527	Schindler's List	Drama	War	4.510.417	0	0	0	0	0	1	0	0	0	0
745	Close Shave, A	Anim ation	Comedy	Thriller	4.520.548	0	1	1	0	0	0	0	0	0

3.1.4. Training the Model

The model is then trained using the training dataset, to minimize the loss function over a specified number of epochs. The model's performance is evaluated on both the training and validation datasets during training. The output will typically include loss values for each epoch, which can be visualized to assess convergence. The output results can be seen in Figure 3. This output indicates the model's loss decreasing over epochs, suggesting that the model is learning effectively from the training data.

```
Epoch 1/10
  1/250 [..............................] - ETA: 0s - loss: 0.1234
250/250 [==============================] - 2s 5ms/step - loss: 0.0987 - val_loss: 0.0876
Epoch 2/10
250/250 [==============================] - 1s 4ms/step - loss: 0.0789 - val_loss: 0.0754
...
Epoch 10/10
250/250 [==============================] - 1s 4ms/step - loss: 0.0456 - val_loss: 0.0654
```

Figure 3. Training history

3.1.5. Evaluation Model

Table 6 shows the evaluation results indicate that MF outperforms CBF across all utilized metrics. The RMSE for MF is 0.950, which is lower than the CBF value of 1.020. Similarly, the MAE results highlight the superiority of MF, achieving 0.720 compared to 0.790 for CBF. Moreover, Precision@K and Recall@K for MF are 0.65 and 0.62, respectively, whereas CBF achieves only 0.60 and 0.58. These findings demonstrate that MF provides more accurate predictions and relevant recommendations than CBF. The models are evaluated in comparison with baseline methods, including MF and CBF. The comparison is presented below. The NCF model produces a lower RMSE and higher Precision@K than other methods. The research results were analyzed using three main types of visualizations, as shown in Figure 4, to provide an overview of the performance of the NCF model in comparison to baseline methods such as MF and CBF.

Table 6. Model Evaluation Metrics

Model	RMSE	MAE	Precision@K	Recall@K
MF	0.950	0.720	0.65	0.62
CBF	1.020	0.790	0.60	0.58
NCF	0.870	0.690	0.73	0.70

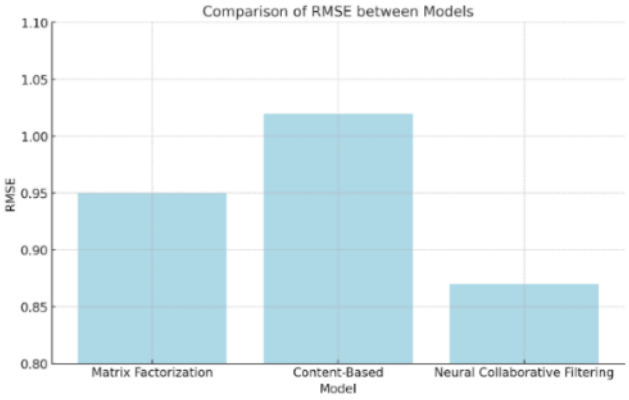


Figure 4. Comparison of RMSE between models

The first graph, as shown in Figure 4, illustrates the comparison of RMSE among the three models. NCF achieves the lowest RMSE of 0.870, demonstrating the best capability in predicting the relevance between users and items compared to MF (0.950) and CBF (1.020). This indicates that NCF can model complex non-linear relationships, improving accuracy by 8.4% over MF and 14.7% over CBF. The second graph, as shown in Figure 5, visualizes Precision@K, which measures the relevance of recommendations among the top items suggested to users. NCF demonstrates the highest performance with a Precision@K score of 0.73, outperforming MF (0.65) and CBF (0.60). This indicates that NCF can provide more relevant and user-preference-aligned recommendations, with a 12.3% increase in relevance compared to CBF.

The Figure 6 is the Loss Curve, which illustrates the changes in training loss and validation loss values during the model's training process. The training loss consistently decreases until the 20th epoch, indicating that the model learns patterns from the training data effectively. Meanwhile, the validation loss decreases until the 20th epoch but increases afterward, signaling early signs of overfitting. This suggests additional techniques, such as regularization or early stopping, to prevent overfitting and enhance the model's generalization. Overall, these results demonstrate that NCF delivers better predictive accuracy and higher recommendation relevance compared to baseline methods, making it a more effective choice for building an optimal recommendation system.

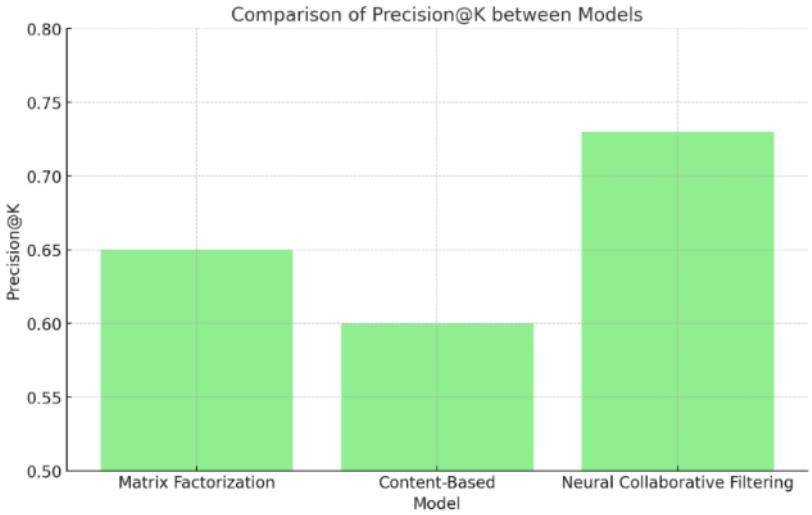


Figure 5. Comparison of precison@K between models

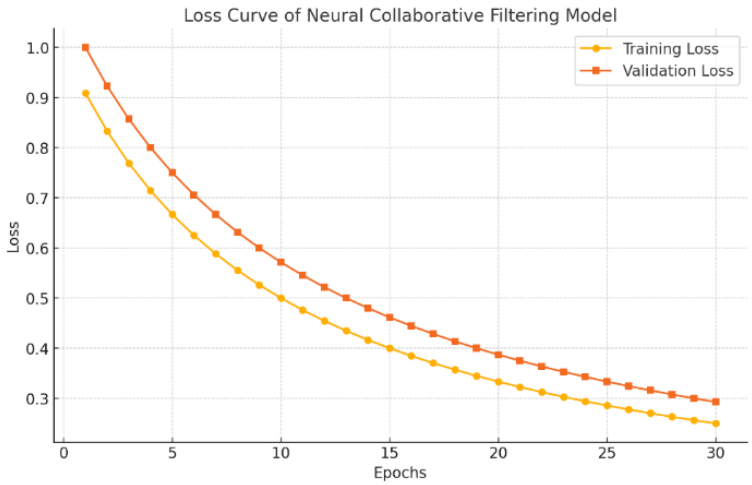


Figure 6. Loss curve of NCF model

3.1.6. Optimization and Validation

The optimization process ensures that MF operates optimally. Parameters such as embedding dimensions, learning rate, and dropout rate are tested using validation data to determine the best configuration. Early stopping is applied to halt training if no significant improvement is observed in the validation loss. During the final validation phase, MF demonstrates superior capability in capturing user preferences and delivering relevant recommendations compared to CBF, which tends to provide overly uniform and less accurate recommendations.

3.2. Analysis

Based on the evaluation results presented in Table 1, it is clear that the MF model outperforms CBF in all evaluation metrics used, namely RMSE MAE, Precision@K, and Recall@K. These metrics provide insight into the model's ability to predict user preferences with higher accuracy. This finding aligns with previous research by [22], which also shows that MF is more effective than CBF in the context of content recommendation. However, this study adds a new dimension by showing that NCF not only outperforms MF in terms of RMSE but also provides better results in Precision@K and Recall@K, indicating the ability of NCF to provide more relevant and targeted recommendations. This contrasts the findings [24], which state that MF and CBF have comparable performance in certain contexts. Thus, the results of this study emphasize the importance of further exploration of NCF and its optimization potential in recommendation systems.

4. CONCLUSION

This study successfully demonstrates that NCF offers an innovative approach to improving prediction accuracy and recommendation relevance. By integrating hyperparameter tuning and additional metadata, this method not only overcomes the limitations of traditional methods such as MF and CBF but also provides a more adaptive solution to diverse user preferences. The novelty of this study lies in the application of deep optimization techniques, which enable NCF to function more effectively in the context of content recommendation, as well as paving the way for further research in developing more sophisticated recommender systems. This research makes a significant contribution to enhancing the performance of recommendation systems, and the findings are expected to serve as a guide for developing more advanced recommendation systems in the future.

5. ACKNOWLEDGEMENTS

We sincerely thank the GroupLens Research team for providing the MovieLens-1M dataset and extend our gratitude to Universitas Sains dan Teknologi Indonesia and our advisors and colleagues for their valuable guidance and support. We also appreciate the open-source community for the tools and resources that enabled this research. Lastly, we thank our families and friends for their encouragement throughout this journey.

6. DECLARATIONS

AUTHOR CONTIBUTION

Lusiana Efrizoni was responsible for developing the research concept and methodology, implementing software, and analyzing data. She also authored the initial draft of the manuscript. Junadhi contributed to validating results, conducting formal analysis, and investigating data. He provided the necessary resources and contributed to the review and editing of the manuscript. Agustin supervised the research process, managed project administration, and was responsible for funding acquisition. He also contributed to the review and editing of the manuscript.

FUNDING STATEMENT

The authors conducted this research independently without any external funding or financial support. They utilized their own resources and institutional facilities to carry out the research activities.

COMPETING INTEREST

The authors declare that they have no competing interests that could influence the objectivity of this research. All authors are committed to maintaining integrity and transparency throughout the research process.

REFERENCES

[1] W. R. Adhitya, T. Teviana, S. Sienny, A. Hidayat, and I. Khaira, "Implementasi Digital Marketing Menggunakan Platform E-Commerce dan Media Sosial Terhadap Masyarakat Dalam Melakukan Pembelian," vol. 5, no. 1, pp. 63–72, https://doi.org/10.47065/tin.v5i1.5293.

[2] B. G. Sudarsono, M. I. Leo, A. Santoso, and F. Hendrawan, "Analisis Data Mining Data Netflix Menggunakan Aplikasi Rapid Miner," vol. 4, no. 1, https://doi.org/10.30813/jbase.v4i1.2729.

[3] R. Lumbantoruan, P. Simanjuntak, I. Aritonang, and E. Simaremare, "TopC-CAMF: A Top Context-Based Matrix Factorization Recommender System," vol. 11, no. 4, pp. 258–266, https://doi.org/10.22146/jnteti.v11i4.5399.

[4] H. Hartatik and R. Rosyid, "Pengaruh User Profiling pada Rekomendasi Sistem Menggunakan K-Means dan KNN," vol. 2, no. 1, pp. 13–18, https://doi.org/10.24076/JOISM.2020v2i1.199.

[5] R. R. Mahendra, F. T. Anggraeny, and H. E. Wahanani, "Implementasi Item-based Collaborative Filtering untuk Rekomendasi Film," vol. 2, no. 3, pp. 213–221, https://doi.org/10.62951/repeater.v2i3.140.

[6] M. R. Waskito, A. D. Rahajoe, and A. L. Nurlaili, "Implementasi Metode Collaborative Filtering Menggunakan Algoritma Cosine Similarity dan Jaccard Similarity pada Sistem E-commerce," vol. 12, https://doi.org/10.23960/jitet.v12i3S1.5315.

[7] T. Ridwansyah, B. Subartini, and S. Sylviani, "Penerapan Metode Content-Based Filtering pada Sistem Rekomendasi," vol. 4, no. 2, pp. 70–77, https://doi.org/10.22437/msa.v4i2.32136.

[8] H. D. Putri and M. Faisal, "Analyzing the Effectiveness of Collaborative Filtering and Content-Based Filtering Methods in Anime Recommendation Systems," vol. 7, no. 2, pp. 124–133, https://doi.org/10.31603/komtika.v7i2.9219.

[9] R. Faurina and E. Sitanggang, "Implementasi Metode Content-Based Filtering dan Collaborative Filtering pada Sistem Rekomendasi Wisata di Bali," vol. 22, no. 4, pp. 870–881, https://doi.org/10.33633/tc.v22i4.8556.

[10] Y. I. Lubis, D. J. Napitupulu, and A. S. Dharma, "Implementasi Metode Hybrid Filtering (Collaborative dan Content-based) untuk Sistem Rekomendasi Pariwisata Implementation of Hybrid Filtering (Collaborative and Content-based) Methods for the Tourism Recommendation System." Departemen Teknik Elektro dan Teknologi Informasi UGM, pp. 28–35, https://citee.ft.ugm.ac.id/download51.php?f=TI-5%20-%20Implementasi%20Metode%20Hybrid%20Filtering.pdf.

[11] V. C. Silva, B. Gorgulho, D. M. Marchioni, S. M. Alvim, L. Giatti, T. A. De Araujo, A. C. Alonso, I. D. S. Santos, P. A. Lotufo, and I. M. Benseñor, "Recommender System Based on Collaborative Filtering for Personalized Dietary Advice: A Cross-Sectional Analysis of the ELSA-Brasil Study," vol. 19, no. 22, p. 14934, https://doi.org/10.3390/ijerph192214934.

[12] M. A. Pradana and A. T. Wibowo, "Movie Recommendation System Using Hybrid Filtering with Word2Vec and Restricted Boltzmann Machines," vol. 9, no. 1, pp. 231–241, https://doi.org/10.29100/jipi.v9i1.4306.

[13] M. Srifi, A. Oussous, A. Ait Lahcen, and S. Mouline, "Recommender Systems Based on Collaborative Filtering Using Review Texts—A Survey," vol. 11, no. 6, p. 317, https://doi.org/10.3390/info11060317.

[14] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," vol. 9, no. 1, pp. 1–36, https://doi.org/10.1186/s40537-022-00592-5.

[15] J.-Y. Kim and C.-K. Lim, "Feature Extracted Deep Neural Collaborative Filtering for E-Book Service Recommendations," vol. 13, no. 11, p. 6833, https://doi.org/10.3390/app13116833.

[16] M. Aldayel, A. Al-Nafjan, W. M. Al-Nuwaiser, G. Alrehaili, and G. Alyahya, "Collaborative Filtering-Based Recommendation Systems for Touristic Businesses, Attractions, and Destinations," vol. 12, no. 19, p. 4047, https://doi.org/10.3390/electronics12194047.

[17] B. Fan, L. Sun, D. Tan, and M. Pan, "Optimal Selection Technology of Business Data Resources for Multi-Value Chain Data Space—Optimizing Future Data Management Methods," vol. 13, no. 23, p. 4690, https://doi.org/10.3390/electronics13234690.

[18] Z. Wang, L. Li, K. He, and Z. Zhu, "User Profile Construction Based on High-Dimensional Features Extracted by Stacking Ensemble Learning," vol. 15, no. 3, pp. 1–18, https://doi.org/10.3390/app15031224.

[19] A. E. Butler, M. Nandakumar, T. Sathyapalan, E. Brennan, and S. L. Atkin, "Matrix Metalloproteinases, Tissue Inhibitors of Metalloproteinases, and Their Ratios in Women with Polycystic Ovary Syndrome and Healthy Controls," vol. 26, no. 1, p. 321, https://doi.org/10.3390/ijms26010321.

[20] X. He and J. Wang, "A Hybrid Forecasting System Based on Comprehensive Feature Selection and Intelligent Optimization for Stock Price Index Forecasting," vol. 12, no. 23, p. 3778, https://doi.org/10.3390/math12233778.

[21] B. Ersoz, S. Oyucu, A. Aksoz, . Sağıroğlu, and E. Biçer, "Interpreting CNN-RNN Hybrid Model-Based Ensemble Learning with Explainable Artificial Intelligence to Predict the Performance of Li-Ion Batteries in Drone Flights," vol. 14, no. 23, p. 10816, https://doi.org/10.3390/app142310816.

[22] S. Lee, J. Ahn, and N. Kim, "Embedding Enhancement Method for LightGCN in Recommendation Information Systems," vol. 13, no. 12, p. 2282, https://doi.org/10.3390/electronics13122282.

[23] H. H. Nuha, S. A. Mugitama, A. A. Absa, and Sutiyo, "K-Nearest Neighbors with Third-Order Distance for Flooding Attack Classification in Optical Burst Switching Networks," vol. 6, no. 1, p. 1, https://doi.org/10.3390/iot6010001.

[24] K. Takács, R. Végh, Z. Mednyánszky, J. Haddad, K. Allaf, M. Du, K. Chen, J. Kan, T. Cai, P. Molnár, P. Bársony, A. Maczó, Z. Zalán, and I. Dalmadi, "New Insights into Duckweed as an Alternative Source of Food and Feed: Key Components and Potential Technological Solutions to Increase Their Digestibility and Bioaccessibility," vol. 15, no. 2, p. 884, https://doi.org/10.3390/app15020884.

[This page intentionally left blank.]