Machine Learning Based Recommender System for Research and Courses

Tazkia Damayantia), Vinna Rahmayanti Setyaning Nastitib), Yufis Azharc), and Zamah Sarid)

Department of Engineering, Universitas Muhammadiyah Malang, Malang, Indonesia

d) Corresponding author: [zamahsari@umm.ac.id](mailto:zamahsari@umm.ac.id)

a) [tazkiadamayanti@webmail.umm.ac.id](mailto:tazkiadamayanti@webmail.umm.ac.id)

b) [vinastiti@umm.ac.id](mailto:vinastiti@umm.ac.id)

c) [yufis@umm.ac.id](mailto:yufis@umm.ac.id)

**Abstract.**  MOOCs have become a popular trend in online learning, fostering the growth of platforms like edX, Udemy, and Coursera. Despite the flexibility MOOCs offer, many learners need help to complete courses, affecting user productivity. To address this, Educational Recommender Systems (ERSs) have been employed to recommend courses based on user interaction history. Collaborative Filtering (CF) is widely used in recommendation systems, with traditional techniques like User and Item Collaborative Filtering focusing on user or item similarities. However, Matrix Factorization (MF) maps users and items into latent spaces for modeling interactions. This study explores the performance of Neural Collaborative Filtering (NCF) using neural networks to learn complex user-item similarity functions. Using the Coursera dataset from Kaggle, The NCF model utilized MF and NN embeddings to capture linear and non-linear interactions, respectively. The results showed that the method achieved an RMSE of 0.1838 on the validation data. These results demonstrate the potential of using NCF in online course recommendation systems.

**Keywords:** Courses, Neural Collaborative Filtering, Recommender System

# INTRODUCTION

Massive Open Online Courses (MOOCs) have become a popular online learning trend in the last decade [1]. The trend of online learning has led to the growth of various learning platforms such as edX, Udemy, Coursera, and others. The popularity of MOOCs in the Technology-Enhanced Learning (TEL) field as a flexible educational platform had provided more opportunities for a global audience to learn [2]. Although learning through MOOCs offers great flexibility, many learners still struggle to complete these courses, affecting user productivity [3]. While the number of MOOCs enrollees is very high, the completion rate is below 10% [4]. To attract user interest, Educational Recommender Systems (ERs) provide course or learning material recommendations based on users’ interaction history with course items [2].

The recommender system not only assists users in identifying the most suitable products based on their preferences but also enhances the efficiency of discovering products within a shorter timeframe [5]. The most widely used Collaborative Filtering (CF) technique in RS is primarily due to their domain-independent [6]. Therefore, CF techniques are more accurate compared to Content-Based (CB) approaches [7]. These techniques work by collecting feedback from users who rate various items and then using similarities in the rating behavior of several users to determine how to recommend an item [8]. Meanwhile, Matrix Factorization (MF) maps users and items into latent space to model user-item interactions [9]. However, RS based on matrix factorization techniques still face the problems of sparse rating data, cold start and scalability of the algorithm, as many missing values in the user-item interaction matrix, leading to challenges in accurately predicting user preferences [10] [11].

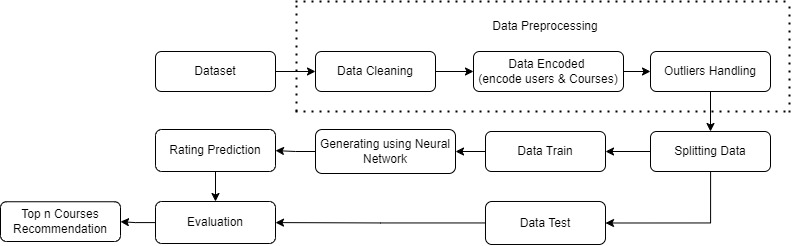
To address the limitations of MF, the use of Neural Collaborative Filtering (NCF) is proposed. NCF utilizes neural networks, particularly Multilayer Perceptron (MLP) to learn more complex similarity functions between users and items. This allows NCF to provide more accurate recommendations without requiring additional information beyond user-item interaction data. Additionally, the more a user interacts with an item, the more recommendations are generated. Qalbyassalam et al. applied the NCF method to a skincare recommendation system, achieving an RMSE of 0.8033 for explicit rating and 0.4931 for implicit rating. This study focuses on using implicit ratings based on sentiment scores from user text reviews to enhance the performance of the skincare recommendation system compared to using explicit ratings [12].

Previous research, “Learner comments-based recommendation system”, conducted by Hazar et al. in 2022, developed a Recommender System (RS) for courses using the Coursera dataset from Kaggle with a content-based method based on user comments to recommend learning videos. This RS utilizes sentiment analysis to address the issue of limited data availability and predict new ratings based on user reviews [13]. Using the same dataset [14], a hybrid filtering approach was also developed for the RS by incorporating student ratings and reviews to tackle the problems of sparsity and cold start in recommendation systems. Both studies used CNN algorithms to predict ratings and cosine similarity to measure the performance of the models used.

This study focuses on the performance of Neural Collaborative Filtering (NCF) on online course for recommending courses based on the rating history of course items provided by users. The use of course ratings represents user interest in those courses. RMSE is employed as an evaluation metric to measure the performance of the model. It is expected that by implementing the recommendation system, users will be able to more efficiently identify courses of interest that match their preferences.

# METHODS

The recommender system for courses is implemented through several key stages. **FIGURE 1** illustrates the proposed stages of the research.



**FIGURE 1**. Research Methods

The research stage started with the collection of rating data and course data. Afterwards, data cleaning was performed by removing null values, duplicates, removing unused attributes, and filtering out courses that have more than 500 reviews. The cleaned data was merged with course data based on course id, and then the categorical data (users and courses) was encoded into numeric. Furthermore, the data was prepared for model training by dividing it into training and validation sets. The recommendation model was developed using Neural network and evaluated by plotting the loss and RMSE for training and validation. After that, rating prediction is performed using the trained model to determine the highest-rated courses which are recommended to the user. Evaluation of the model results is done by comparing the generated recommendations with the user's rating history, and in the final stage, the recommended courses are displayed to the user.

In this research, the Coursera dataset used to create a recommender system comes from Kaggle, the same dataset used in the research article by Hazar et al [13]. The data preprocessing stages include:

* Data cleaning

At this stage, data cleaning involves addressing missing values and duplicate data. Missing values (null or NaN) are first identified, and common methods to handle them include deleting rows or columns with missing values if their number is insignificant relative to the total data, or imputing missing values using strategies such as the mean, median, or mode of the respective column. This is crucial as null values in reviews or incomplete course data can interfere with analysis. Next, duplicate data, which consists of rows with identical entries across all columns, is identified. If duplicates are found, they are removed to prevent them from affecting the analysis results. Ensuring the removal of duplicate data is vital for maintaining consistency and accuracy.

* Data encoded

Before the encoding stage, merging data involved combining course and rating datasets based on common identifiers, such as course\_id. After that encoding categorical data such as reviewer names and course IDs are converted into a numerical form that can be processed by the model. Commonly used approaches are label encoding where each category is represented by a unique numerical value, or one-hot encoding where each category is converted into a binary vector. These encodings allow the use of categorical data in the process of training and evaluating models with TensorFlow.

* Outlier Handling

For outliers Processing, handle outliers based on the final input module data. The presence of outliers will undoubtedly have a negative impact on the data. Isolation Forest is a machine learning algorithm used to detect and eliminate outliers. To use Isolation Forest, import the necessary libraries, such as the IsolationForest class and other data processing libraries from scikit-learn. After handling missing values and converting data types, create an IsolationForest object for anomaly detection and set relevant parameters such as n\_estimators and contamination. Sort or select a threshold based on the scores to determine which samples are considered outliers. Then, handle their outliers data accordingly [15].

In the next stage, data splitting is performed with a ratio of 90:10 for training and validation data. The neural network was model developed using the Tensorflow library, which involves creating a model consisting of input, embedding, and output layers. The model utilizes a Neural Collaborative Filtering (NCF) approach with Multi-Layer Perceptron to learn the similarity function of user behavior based on simple one-hot encodings of users and items [16]. The embedding layers transform sparse vectors into dense vectors for users and items, which can be considered as latent feature vectors. These vectors are then fed into the MLP to obtain prediction scores. Each layer in the MLP models potential user-item interaction structures, with the final hidden layer predicting the end score. The prediction model for user responses to items can be expressed with the following formula [17].

(1)

Explanation:

U : Latent Matrix ( I x K)

V : Latent Matrix (J x K)

Thefa (ƒ) : Parameter model

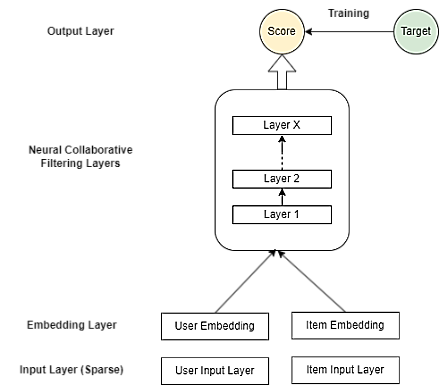
∈ : Vektor one-hot encoding from user

∈ : Vektor one-hot encoding from item

: Model parameters of the interaction function

This research utilizes a methodology that makes use of item IDs (courses) and user IDs (learners) as input features. The embedding layers above the input layer are fully connected layers that transform sparse representations into dense vectors. The item and user embeddings are then sent into a multi-layer neural architecture, which maps the latent vectors to predict ratings. The model is designed to identify latent structures in item-user interactions, by modifying the NFC layer. The target score is the predicted rating generated by the model based on the interaction between the user and item embeddings [9].

NCF in course recommendation systems models learner-course interactions through neural networks, predicting the interaction based on the embedding representation of the user and the course, along with the associated biases. To address the cold start problem for new items or users, the NCF model uses embeddings to represent users and items as dense vectors in a latent space, which are then processed by a multi-layer neural network to predict interactions. By integrating item-user embedding representations with matrix factorization, the model effectively tackles the challenges of data sparsity and cold start problems. **FIGURE 2** below shows the workflow of Neural Collaborative Filtering.



**FIGURE 2**. Flowchart Neural Collaborative Filtering

After training, the model performance is evaluated using test data with RMSE (Root Mean Squared Error). This technique is used to evaluate the accuracy of recommendation systems based on rating data to find low prediction errors. The principle of this technique is based on using the predicted rating and the actual rating, then calculating the average error of the test set using the equation, where 𝑃 is the predicted rating and 𝑅 is the actual rating, and producing a final score [8].

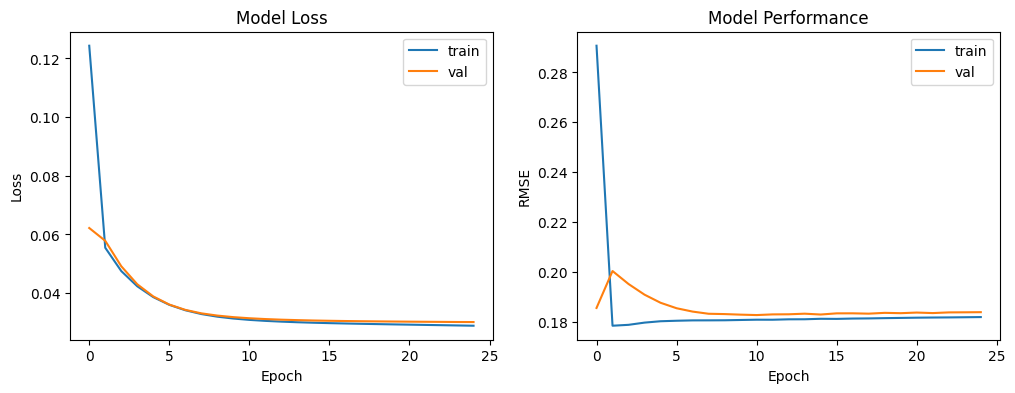
# RESULTS AND DISCUSSION

In this section, the result of the research conducted focuses on the performance of Neural Collaborative Filtering. The usage of the Coursera dataset in Kaggle, with data used 1.45 million reviews uploaded by users. There are two datasets Coursera\_courses.csv consisting of 623 courses on Coursera and Coursera\_reviews.csv consisting of 1.4 million reviews with ratings. The data preprocessing involves cleaning by removing duplicates and entries with NaN/Null values. This step is very important to keep the data consistent and ensure there is no duplication of data. After that, merging data involved combining course and rating datasets based on course\_id. Encoding is then performed on the course\_id and user\_id attributes to convert them into numeric forms that the model can process. Through threshold screening for outliers handling, we get 488554 normal data and 4930 abnormal data, so these 4930 variables need to be eliminated. The preprocessed Coursera dataset can be seen in **TABLE 1**.

**TABLE 1**. Final data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Course\_url | Course\_id | Reviewers | Rating | User\_id |
| Gamification | https://www.coursera.org/learn/gamification | 99 | Valery L | 5.0 | 87097 |
| Version Control with Git | <https://www.coursera.org/learn/version-control-with-git> | 292 | Brayan Y M C | 5.0 | 246863 |
| Econometrics: Methods and Applications | <https://www.coursera.org/learn/erasmus-econometrics> | 286 | Mrinal | 5.0 | 42197 |
| Customer Analytics | <https://www.coursera.org/learn/wharton-customer-analytics> | 50 | Xinmiao Z | 5.0 | 133760 |
| The Technology of Music Production | https://www.coursera.org/learn/technology-of-music-production | 332 | Suryadeep D | 5.0 | 91252 |

The Neural Collaborative Filtering (NCF) model used employs two inputs, user and item. Training with a dataset that has been divided into 439698 training data and 48856 validation data. NCF combines Matrix Factorization (MF) and Neural Networks (NN) to capture the non-linear interaction between users and items in recommender systems. In the given model, two types of embedding are used: MF embedding and NN embedding. MF embeddings capture linear interactions between users and items, while NN embeddings capture more complex non-linear interactions. The training process for the Neural Collaborative Filtering model was performed with 25 epochs and a batch size of 1024. EarlyStopping was used to help prevent overfitting by stopping training at the point where the model had achieved its best performance on the validation data. The results of the model training can be seen in **FIGURE 3**.



**FIGURE 3**. Evaluation Model

Based on **FIGURE 3**, we can analyze the performance of the NCF model in predicting user ratings for course items. The model with input (x) index user and product index using explicit rate as a target (y) gets good result with an RMSE on the validation set reached 0.1838, suggests that the model is accurate in predicting user ratings, making it a reliable choice for the recommender system. Although a slight increase in validation RMSE after epoch 4 hints at potential overfitting, the consistent performance suggests that EarlyStopping was effective. The combination of MF and NN embeddings allowed the model to capture both simple and complex user-item interactions, contributing to its success. NCF model's training and validation performance, as evaluated, suggests that it is well-suited for applications in recommendation systems, particularly in scenarios where both linear and non-linear interactions play a significant role in user behavior prediction.

# CONCLUSIONS

Based on the research, it can be concluded that the performance of Neural Collaborative Filtering (NCF) on Coursera online course data, using user rating history, achieves an RMSE of 0. 1838 on the validation data in a 90:10 training-validation scenario. Implementing NCF in an online course recommendation system allows users to receive course recommendations that better match their interests and preferences. By integrating item-user embedding with matrix factorization, the proposed model effectively mitigates issues related to data sparsity and the cold start problem. As a result, users can more efficiently identify courses of interest, potentially leading to higher course completion rates and increased user productivity. Future improvements could focus on further tuning the model architecture or employing additional regularization techniques to refine performance further.

# Acknowledgments

This research is supported by Universitas Muhammadiyah Malang. The authors also express their gratitude to the Informatics Laboratory of UMM for their invaluable guidance and support throughout this study.

# References

1. M. J. Gomez, M. Calderón, V. Sánchez, F. J. G. Clemente, and J. A. Ruipérez-Valiente, "Large scale analysis of open MOOC reviews to support learners’ course selection," *Expert Systems with Applications,* vol. 210, p. 118400, 2022.
2. R. Alatrash, M. A. Chatti, Q. U. Ain, Y. Fang, S. Joarder, and C. Siepmann, "ConceptGCN: Knowledge concept recommendation in MOOCs based on knowledge graph convolutional networks and SBERT," *Computers and Education: Artificial Intelligence,* vol. 6, p. 100193, 2024.
3. J.-W. Tzeng, N.-F. Huang, A.-C. Chuang, T.-W. Huang, and H.-Y. Chang, "Massive open online course recommendation system based on a reinforcement learning algorithm," *Neural Computing and Applications,* pp. 1-12, 2023.
4. S. K. Narayanasamy and A. Elçi, "An effective prediction model for online course dropout rate," *International Journal of Distance Education Technologies (IJDET),* vol. 18, no. 4, pp. 94-110, 2020.
5. K. K. Jena *et al.*, "E-learning course recommender system using collaborative filtering models," *Electronics,* vol. 12, no. 1, p. 157, 2022.
6. F. Fkih, "Enhancing item-based collaborative filtering by users’ similarities injection and low-quality data handling," *Data & Knowledge Engineering,* vol. 144, p. 102126, 2023.
7. G. Behera and N. Nain, "DeepNNMF: deep nonlinear non-negative matrix factorization to address sparsity problem of collaborative recommender system," *International journal of information technology,* vol. 14, no. 7, pp. 3637-3645, 2022.
8. Y. Afoudi, M. Lazaar, and M. Al Achhab, "Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network," *Simulation Modelling Practice and Theory,* vol. 113, p. 102375, 2021.
9. P. Paul and R. P. Singh, "Sentiment Rating Prediction using Neural Collaborative Filtering," in *2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, 2022, vol. 7, pp. 148-153: IEEE.
10. R. Lara-Cabrera, Á. González-Prieto, and F. Ortega, "Deep matrix factorization approach for collaborative filtering recommender systems," *Applied Sciences,* vol. 10, no. 14, p. 4926, 2020.
11. J. Wang and L. Liu, "A multi-attention deep neural network model base on embedding and matrix factorization for recommendation," *International Journal of Cognitive Computing in Engineering,* vol. 1, pp. 70-77, 2020.
12. C. Qalbyassalam, R. F. Rachmadi, and A. Kurniawan, "Skincare recommender system using neural collaborative filtering with implicit rating," pp. 272-277: IEEE.
13. M. J. Hazar, M. Zrigui, and M. Maraoui, "Learner comments-based recommendation system," *Procedia Computer Science,* vol. 207, pp. 2000-2012, 2022.
14. M. J. Hazar, S. Zrigui, M. Maraoui, M. Zrigui, and H. Nicolas, "A Recommendation System Involving a Hybrid Approach of Student Review and Rating for an Educational Video," *Journal of the Brazilian Computer Society,* vol. 29, no. 1, pp. 73-85, 2023.
15. Y. Yang *et al.*, "Prediction analysis of TBI 24-h survival outcome based on machine learning," *Heliyon,* vol. 10, no. 9, 2024.
16. H. Liu, "Implementation and Effectiveness Evaluation of Four Common Algorithms of Recommendation Systems-User Collaboration Filter, Item-based Collaborative Filtering, Matrix Factorization and Neural Collaborative Filtering," pp. 224-227: IEEE.
17. H. Morise, K. Atarashi, S. Oyama, and M. Kurihara, "Neural collaborative filtering with multicriteria evaluation data," *Applied Soft Computing,* vol. 119, p. 108548, 2022.