JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-20, No.-2, February (2025) pp 100 - 114 ISSN (Print) 0973-8975

COMPARATIVE ANALYSIS OF REGRESSION AND COLLABORATIVE FILTERING MODELS FOR RECOMMENDATION SYSTEMS: AN EMPIRICAL STUDY

Nisha Bali[1], Kulvinder Singh[2], Sanjeev Dhawan[3]

[1,2,3] Department of Computer Science Engineering, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, India

Email: [1]Kherwal5855@gmail.com, [2]ksingh2015@kuk.ac.in, [3]sdhawan2015@kuk.ac.in

Corresponding Author: **Nisha Bali**

https://doi.org/10.26782/jmcms.2025.02.00009

(Received: November 29, 2024; Revised: January 19, 2025; Accepted: February 06, 2025)

Abstract

Recommendation systems are very crucial for enhancing the utility of a given product and or service for a specific user in different fields. This paper focuses on the comparison of various filtering techniques in order to determine their effectiveness in identifying the preferences of the user. The paper also looks into basic methods like user-based collaborative filtering and item-based collaborative filtering, which uses the item's attributes. Also, the paper assesses the subsequent methods such as linear regression, ridge regression, Lasso regression, random forest regression, and XGBoost regression. From the performance evaluation metrics, the researchers reach RMSE and MAE to compare the effectiveness of the proposed methods and reveal their weaknesses. This paper aims to evaluate the performances of the above filtering approaches to gain an understanding of the extent to which these methods improve the recommendations' accuracy and contribute to the literature by providing recommendations on filtering models suitable for various recommendation tasks.

Keywords: Collaborative filtering, Recommendation system, user-based,item-based, Model Comparison.

I. Introduction

In the context of the modern digital environment, users are exposed to mass amounts of data and options. Recommender systems are recommender systems that have emerged as indispensable instruments to facilitate individualizing of users' experience from content that matches their preferences. These systems are extensively applied in practice in fields like e-shopping, social networking, and entertainment and can help clients find something like a movie, an item, or friends [XIV].

Generally, recommendation systems can be divided into three categories: collaborative filters, content-based filters, and hybrid filters [XX]. Collaborative filtering operates from the patterns of users and items and seeks to come up with recommendations. This may be achieved using user-user methods whereby like users are grouped or item-item where similar items to the one a user has indicated liking are provided. On the other hand, content-based filtering uses properties of items such as movie genres directors', or actors' recommended items that may be interesting to a given user.

Fig. 1 illustrates collaborative filtering, where recommendations are made based on similar users' preferences. In this system, users interacting with the same items are identified as having similar tastes. For instance, if one user reads a new item, the system will recommend it to another similar user. This method helps provide personalized suggestions by leveraging the collective behavior of users with shared interests.

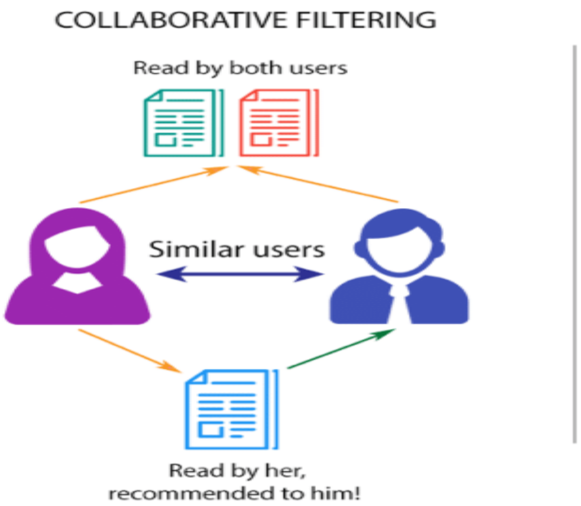

Fig. 1. Collaborative filtering [XXVII]

Collaborative filtering is one of the recommendation system techniques by which the preference of any particular user is forecasted with the preference data of other like-minded users or the same particular item [XXII]. It operates in two main forms, including two main filtering methods: user base and item base. User-based collaborative filtering finds similar users and suggests things they like. In contrast, item-based Collaborative filtering finds other things liked by users who also liked the current item. It employs sizeable matrices of users and items; it applies k-NN or matrix factorization, depending on the problem. Included in Netflix and Amazon, for instance, collaborative filtering aims at assisting separate users and taking advantage of the overall tendencies that are characteristic of a given community [V, IX].
The primary objectives of this study are:

- To examine the efficacy of various filtering techniques
- To discuss the basic types of the traditional companion methods, such as the user and item-based strategies on the CF.
- To analyze some of the multiple linear regression methodologies, including Linear, Ridge, Lasso, Random Forest Regression, and XGBoost Regression.

Altogether, this paper provides an experimental comparison of various CF approaches using the MovieLens dataset and tries to shed light on the real-world performance of the described techniques. Using fundamental guiding review measures like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), the result demonstrates the anticipated method's benefits and flaws. Such findings help decide on suitable models for various recommendation use cases, thus contributing to the area of personalized content recommendation.

II. Literature Review

Recommendation systems are used to filter large volumes of data when a large number of choices are available. There is extensive use of collaborative filtering (CF), although it has inherent problems associated with data sparsity and cold start. These challenges are overcome by Natarajan et al. [XIX] by combining Linked Open Data to matrix factorization results, making the data handling easier and decreasing the sparsity of the data. Zhou et al. XVII] present FMLP-Rec, another all-MLP model that eliminates noise and has better performance than Transformer structures. This study has presented Efficient Heterogeneous Collaborative Filtering by Chen et al. [II]. As an enhancement of HCF, this model does not apply any negative sampling, and it enhances both the accuracy and the training rate. Sharma et al. [XXXI] present one of the hybrid filtering algorithms based on collaborative and content-based filtering approaches to avoid the cold start and scalability problems. DeepICF was introduced by Xue et al. [XIII], which proposed a deep learning model that learns the complex item dependencies for improving the MovieLens and Pinterest datasets. Neural collaborative filtering with knowledge graph embeddings is exploited by Shokrzadeh et al. [XXIII], improving the recall, precision, and F1 score. Huang et al. [XXI] deal with the problem of computational complexity in deep neural networks by using Broad Learning Systems at the same time, achieving high accuracy. Al-Ghobari et al. [XXV] proposed LAPTA, which incorporates a user's preference along with their GPS to provide customized travel directions. To overcome this issue, Nguyen et al. [III] reconstruct the CF by incorporating cognitive similarity measures that increase the level of accuracy and reliability. Peng et al. [XXVI] propose the application of deep reinforcement learning for improving CF regarding the data sparsity and cold start problems, where the DDPG algorithm and SVD demonstrate high effectiveness. Another study done by Fang et al. [XXIV] proposed CF-DNNF. In this model, the performance of prediction is enhanced by the use of deep neural networks than traditional methods. Widiyaningtyas et al. [VII] propose UPCSim, incorporating genre and user profile data to enhance recommendation accuracy. Dang et al. [XV] integrate sentiment analysis into CF, improving recommendation reliability. Alhijawi et al. [XXVIII] introduce INH-BP, a predictive model that uses genetic algorithms to handle cold start and sparsity. K and Srikantaiah [IV] compare similarity metrics to enhance movie recommendation systems. Aljunid and Huchaiah [XVI] developed ΓUICF, a hybrid model addressing sparsity and scalability. Iwendi et al. [XXXIII] present a pointer-based item-to-item CF system with machine learning, improving accuracy and scalability. Nassar et al. [I] combine deep neural networks with matrix factorization for multi-criteria CF, achieving superior results. Kim et al. [X] use emotional information to enhance recommendations, demonstrating improved accuracy. Lastly, Mu and Wu [XII] offer a multimodal movie recommendation system integrating deep learning and multimodal data, addressing cold-start issues and enhancing prediction accuracy. These studies collectively advance recommendation systems through deep learning, sentiment analysis, user profiles, and multimodal data integration.

Despite the advancements in recommendation systems highlighted by various studies, several research gaps remain. The literature lacks a comprehensive comparison of traditional regression models and their effectiveness in recommendation tasks. Existing studies predominantly emphasize deep learning and hybrid methods, overlooking the potential benefits of simpler, more interpretable models. This will make simpler models and reduce complexity.

Moreover, while techniques like FMLP-Rec, DeepICF, and knowledge graph embeddings significantly improve accuracy and efficiency, they often come with increased computational complexity and resource requirements. This raises the need to evaluate the trade-offs between model complexity and performance in practical applications.

III. Methodology

This research aims to comprehensively evaluate the effectiveness of various recommendation filtering methods using the MovieLens dataset. The methodology includes data preparation, implementation of filtering techniques, regression models, and evaluation metrics to assess performance. This section outlines each step in detail.

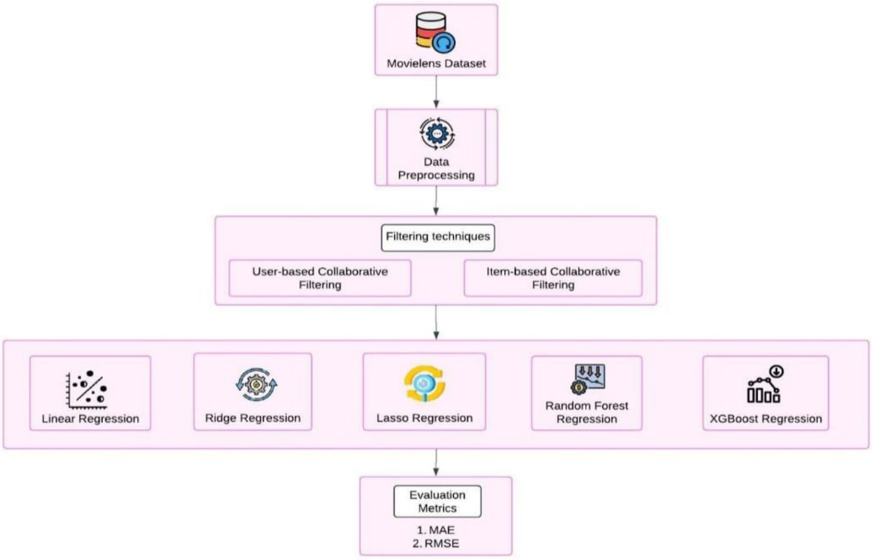

Fig. 2. System block diagram

Fig 2 presents a workflow for evaluating recommendation system models using the MovieLens dataset. Preprocessing is the first step to ensure that the data is clean and ready for analysis. Applied two Filtering techniques that is User-based Collaborative filtering and Item-based Collaborative filtering. Next, the preprocessed data is fed into various regressor models like Linear, Ridge, Lasso, Random Forest (RF), and XGBoost, which predict user ratings. To evaluate the performance of these models, Mean Absolute Error (MAE) is used. In parallel, a Collaborative Filtering approach is employed, utilizing both user-based and item-based filtering techniques. The accuracy of these models is assessed using the Root Mean Squared Error (RMSE). This workflow allows a comparative analysis of traditional regressor models and collaborative filtering for building a robust recommendation system.

IV. Data Preparation

The datasets used in this study are the MovieLens ratings, movies, and user datasets. These datasets contain:

- **Rating Dataset**: Information on user ratings for movies.
- **Movies Dataset**: Details about movies, including titles and genres.
- **Users Dataset**: Demographic information of users.

The data preparation process involved the following steps:

- **Loading Data**: The datasets were loaded using the pandas' library in Python.
- **Merging Datasets**: The ratings, movies, and user datasets were merged based on relevant keys (e.g., MovieID and UserID).
- **Data Cleaning**: Redundant columns were removed, and missing values were handled appropriately.

- **Filtering Data**: Movies with only one rating were filtered out to ensure that the analysis focuses on movies with sufficient user feedback.
- **Train-Test Split**: A ratio of 80-20 was used to divide the cleaned and merged dataset into training and testing sets.

V. Filtering Techniques

V.i. Collaborative Filtering

Based on patterns of interaction between users and items, collaborative filtering can predict user preferences. Collaboration filtering was implemented in two different ways:

• **User-Based Collaborative Filtering:** This method connects users with preferences that are similar to those of a particular user and suggests that items similar users have favored [XXX]. A surprise library was utilized to implement this method, and the algorithm used here was the KNNBasic. The same user comparison was based on Euclidean. A possible similarity was between 0 and 1[VI].

• **Item-Based Collaborative Filtering**: It suggests other items liked by the target user – this makes use of the cross-similarity of items that a user has given high ratings. Surprise library's KNNBasic algorithm was also used in the tests. Item similarity was computed with cosine similarity [XVIII, XI].

VI. Regression Models

To further enhance prediction accuracy, several regression models were applied to the recommendation task:

Linear Regression

This basic model estimates the ratings using attributes belonging to users and movies where it is assumed that such attributes are linearly related to the ratings. Linear regression implies a linear relationship between the target (rating) and the features (user and movie attribute X):

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

Where, y is the predicted rating, X_1, X_2,…, X_n are the features (user and movie attributes), β_0 is the intercept, β_1, β_2,…,β_n are the model coefficients and ϵ is the error term.

Ridge Regression

Linear regression is an improvement of Ridge Regression, which has features further improved by L2 regularization. This procedure reduces multicollinearity by adding a penalty to large coefficients, thus promoting the model's capability in generalization. The cost function becomes:

$$Cost\ Function = \sum_{i=1}^{m} (y_i - (\beta_0 + \sum_{j=1}^{n} \beta_j X_{ij}))^2 + \lambda \sum_{j=1}^{n} \beta_j^2$$

where, $\boldsymbol{\lambda}$ is the regularization parameter that controls the strength of the penalty and the penalty term $\lambda \sum_{j=1}^{n} \beta_j^2$ prevents coefficients from becoming too large.

Lasso Regression

Also, in a similar manner to ridge regression, Lasso Regression avails L1 regularization. This type of regularization is often beneficial for a feature selection process since the regularization term can drive some of the coefficients to zero, thus producing a sparser model. The cost function is:

$$Cost\ Function = \sum_{i=1}^{m} \left(y_i - \left(\beta_0 + \sum_{j=1}^{n} \beta_j X_{ij}\right)\right)^2 + \lambda \sum_{j=1}^{n} |\beta_j|$$

Where, the penalty term $\lambda \sum_{j=1}^{n} |\beta_j|$ promotes sparsity in the model.

Random Forest Regressor

This method of learning creates several decision trees and then uses their results to provide improved and accurate predictions. Specifically, it is very good at modeling nonlinear patterns that may exist within the input data. Where each tree T_k provides a predicted rating and the final prediction is the average of the outputs from all the tress:

$$\hat{y} = \frac{1}{K} \sum_{k=1}^{K} T_k(X)$$

where, K is the number of trees and $T_k(X)$ is the prediction from the K-th trees.

XGBoost Regressor

The XGBoost algorithm builds a dynamic ensemble of weak learners (decision trees) according to a stage-by-stage approach. It optimizes the following objective function, which includes both a loss function and a regularization term:

$$Objective = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Where, $L(y_i, \hat{y}_i)$ is the loss function, $\Omega(f_k)$ is the regularization term, which penalizes model complexity to prevent overfitting and f_k is the weak k-th learner.

The Regularization term $\Omega(f_k)$ often takes the form:

$$\Omega(f_k) = \gamma T + \frac{1}{2} \boldsymbol{\lambda} ||w||^2$$

Where, T is the number of leaves in the tree, w represents the leaf weights and λ and γ are regularization parameters.

VII. Evaluation Metrics

The following evaluation metrics were employed to assess the performance of the filtering methods and regression models: The mean absolute error and root mean square error were used to evaluate the model. The formulas are

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2} \quad (1)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y_i}| \quad (2)$$

Algorithm 1 describes a movie rating prediction and recommendation system that involves various machine-learning techniques. It starts by loading user, movie, and rating datasets, merging and filtering them to prepare the data. Afterward, it trains a number of regression models (Linear, Ridge, Lasso, Random Forest, and XGBoost) to predict movie ratings. Additionally, the algorithm uses KNN to implement collaborative filtering for personalized recommendations (both user- and item-based). Additionally, cosine similarity is used to recommend movies based on genre similarity. The system returns top movie recommendations based on user input, combining regression predictions and similarity-based filtering for accurate suggestions.

Algorithm 1: Collaborative Filtering and Regression-Based Recommendation system

---

Algorithmic Input Parameters:

- Number of iterations (θ_iter)
- Train-test split ratio (λ_split)
- Performance evaluation metrics (RMSE, MAE)
- Collaborative filtering methods (User_CF, Item_CF)
- Regression models (Linear, Ridge, Lasso, Random Forest, XGBoost)

Input Data Parameters:

- User-item interaction matrix (U)
- Item attributes matrix (A)
- User preference data (P)
- Ratings data (R)

Output:

- Best collaborative filtering method (lowest RMSE/MAE)
- Best regression model (lowest RMSE/MAE)

Step-by-step Algorithm:

1. Initialize:
- Load the user-item interaction matrix U and item attributes matrix A.
- Set the number of iterations (θ_iter) and split ratio (λ_split) for train-test data splitting.

2. Main Iterative Process:
- While k < θ_iter, repeat:

3. Collaborative filtering Evaluation:
- For each filtering method (User_CF, Item_CF):

- Apply the filtering method to predict user ratings based on training data.
- Evaluate the predicted ratings on the test data using RMSE and MAE.

4. Regression Models Evaluation:

- For each regression model (Linear, Ridge, Lasso, Random Forest, XGBoost):
- Train the model using the training data.
- Predict user ratings using the trained model on the test data.
- Evaluate the model performance using RMSE and MAE.

5. Identify Best Performing Models:

- Compare the RMSE and MAE scores for both collaborative filtering methods and regression models.
- Identify the best-performing collaborative filtering method (lowest RMSE/MAE).
- Identify the best-performing regression model (lowest RMSE/MAE).

6. Update Iteration:

- Increment the iteration counter and continue.

7. Return:

- After completing all iterations, return the best filtering method and the best regression model based on performance metrics.

VII. Result and Analysis

The implementation of the methods and models was carried out using Python, leveraging libraries such as pandas, sci-kit-learn, Surprise, and XGBoost. The results obtained for comparative analysis are as follows

Table 1: Comparison of the Regressor model

Models	MAE
Linear Regressor	0.9302
Ridge regressor	0.9302
Lasso regressor	0.9340
Random Forest regressor	0.9020
XGBoost regressor	0.8620

The performance of various regression models was evaluated and compared using the MAE metric.

As illustrated in Table 2, the Linear Regressor and Ridge Regressor both achieved an identical MAE of 0.9302, indicating similar performance in predicting movie ratings. The Lasso Regressor, incorporating L1 regularization for feature selection, showed a slightly higher MAE of 0.9340, suggesting a marginally less accurate prediction compared to the previous models. The Random Forest Regressor, an ensemble learning method, demonstrated an improved MAE of 0.9020. The XGBoost Regressor outperformed all other models with the lowest MAE of 0.8620, showcasing its advanced gradient boosting approach's robustness in achieving more accurate predictions.

Fig 3 illustrates the Mean Absolute Error (MAE) scores for various regression models. The x-axis represents the different models, including Linear Regressor, Ridge Regressor, Lasso Regressor, Random Forest Regressor, and XGBoost Regressor, while the y-axis represents the corresponding MAE scores. From the graph, it is evident that the XGBoost Regressor performs the best, achieving the lowest MAE score, indicating higher prediction accuracy compared to the other models. On the other hand, the Lasso Regressor has the highest MAE, followed by the Ridge Regressor and Linear Regressor, which exhibit similar performances. Forest Regressor also shows a significant improvement over the Lasso, Ridge, and Linear models but is outperformed by XGBoost. Overall, this graph suggests that XGBoost is the most effective regressor in this evaluation, providing the most accurate predictions with the lowest error.

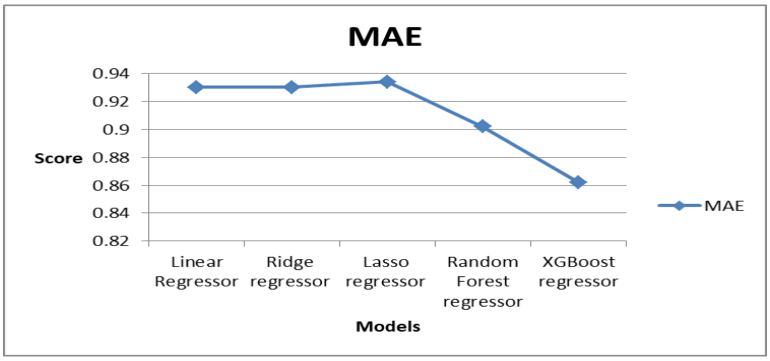

Fig. 3. MAE scores of the regressor model

Table 2 : Comparison of the Regressor model

Models	RMSE
User-based CF	0.9321
Item-based CF	0.9123

Table 2 shows the RMSE test of both users and items CF using KNN. With the User-Based Collaborative Filtering, the RMSE achieved was 0. 9321, which clearly shows the predictive efficiency attained by utilizing the users' resemblance. As a result, the RMSE of Item-Based Collaborative Filtering was slightly lower than that of User-Based Collaborative Filtering at 0. 9123, which shows that item similarity makes predictions more precise when based on recommendations. These results suggest that the choice of the specific method of collaborative filtering has to be made depending on the characteristics of the data set and the type of users' interactions with items.

Fig 4 compares the RMSE of two models: User-based Collaborative KNN Filtering and Item-based Collaborative Filtering. On the y-axis, there is the RMSE score, which measures prediction accuracy, with lower values indicative of better performance. The chart shows that Item-based Collaborative Filtering achieves a lower RMSE score, around 0.91, indicating better prediction accuracy compared to

User-based Collaborative KNN Filtering, which has a higher RMSE score, of approximately 0.93. This suggests that the item-based approach provides more accurate recommendations in this evaluation, making it more effective at reducing prediction errors.

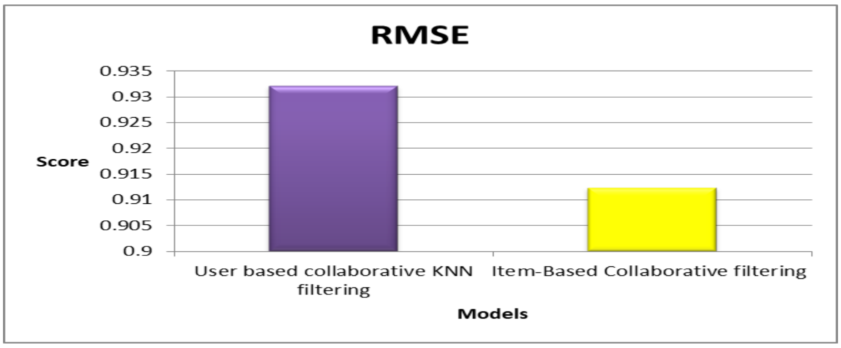

Fig. 4. RMSE Score of Collaborative Filtering Models

In summary, the findings of MAE and RMSE on various models and approaches are also instructive when assessing their forecasting performance. Thus, the improved result of the XGBoost Regressor in the aspect of AME proves that it has the potential to improve the recommendation system. Thus, the comparison of user-based and item-based CF methods is also useful for practical advice on the directions for improvement of CF methods.

Table 4: Comparative analysis Of Models with an existing model

Model	RMSE	MAE
KNN-w-Baseline[28]	1.009	-
UTV [31]	1.13	-
KNN[32]	-	0.887
Restricted Boltzmann Machines[33]	-	1.056
Best model (Item Based CF)	**0.9123**	
Best Model(XGBoost)		**0.8620**

Results emphasize that the best Item-Based Collaborative Filtering model gave an RMSE of 0. 9123. On the contrary, the proposed solution XGBoost Regressor model achieved an overall MAE of 0. 8620, which highlighted that the presented approaches performed significantly better than the conventional methods. A gradient-boosting approach is highly efficient in addressing multivariate data because of its multivariate nature.Also, the user-based and item-based collaborative filtering approaches are compared to show that the latter brings slightly better RMSE results (0. 9123 vs. 0. 9321). In other words, using similarities between items improves recommendations. These points help the practitioners to choose the optimal algorithms for varied settings and thus contribute to enhancing the creation of accurate and efficient recommender systems. This work also demonstrates how there is a need to incorporate more than one stability measure, MAE and RMSE in this case, to get an overall view of the performance of the model that can help enhance the overall satisfaction of the users in recommendation services.

VIII. Conclusion and Future Scope

The purpose of this paper is to provide a quantitative analysis comparing the effectiveness of collaborative filtering and advanced regression models in recommendation systems. The findings show that the XGBoost Regressor comes out clearly on top concerning the accuracy of prediction, which makes it a reliable system for recommendation. Item-based-kNNs also perform better than user-based-kNNs, once again pointing to the idea that using similarities between items is a very effective strategy. For future works, it is possible to investigate the reinforcement of both methods and use a combination of collaborative recommendation systems and content-based recommendation systems to gather the merits of both. Incorporating a deep learning approach with other contextual features like users' mood or temporal data will only advance the accuracy and specificity of the recommendations. As for these models, the assessment of larger and various data sets will also be important in determining their expansibility and efficacy in different fields. This will continue to contribute to the enhancement of better and more effective recommendation systems specialized for the actual end-users.

Conflicts of Interest

The authors declare that they have no conflict of interest exists.

References

I. Al-Ghobari, M., Muneer, A., & Fati, S. M. (2021). Location-Aware Personalized Traveler Recommender System (LAPTA) Using Collaborative Filtering KNN. Computers, Materials & Continua/Computers, Materials & Continua (Print), 69(2), 1553–1570. 10.32604/cmc.2021.016348

II. Alhijawi, B., Al-Naymat, G., Obeid, N., & Awajan, A. (2021). Novel predictive model to improve the accuracy of collaborative filtering recommender systems. Information Systems, 96, 101670. 10.1016/j.is.2020.101670

III. Alcacer, A., Epifanio, I., Valero, J., & Ballester, A. (2021). Combining Classification and User-Based Collaborative Filtering for Matching Footwear Size. Mathematics, 9(7), 771. 10.3390/math9070771

IV. Aljunid, M. F., & Huchaiah, M. D. (2021). An efficient hybrid recommendation model based on collaborative filtering recommender systems. CAAI Transactions on Intelligence Technology, 6(4), 480–492. 10.1049/cit2.12048

V. Anwar, T., & Uma, V. (2021). Comparative study of recommender system approaches and movie recommendation using collaborative filtering. International Journal of Systems Assurance Engineering and Management, 12(3), 426–436. 10.1007/s13198-021-01087-x

VI. Chen, C., Zhang, M., Zhang, Y., Ma, W., Liu, Y., & Ma, S. (2020). Efficient Heterogeneous Collaborative Filtering without Negative Sampling for Recommendation. Proceedings of the AAAI Conference on Artificial Intelligence, 34(01), 19–26. 10.1609/aaai.v34i01.5329

VII. Dang, C. N., Moreno-García, M. N., & De La Prieta, F. (2021). An Approach to Integrating Sentiment Analysis into Recommender Systems. Sensors, 21(16), 5666. 10.3390/s21165666

VIII. Fang, J., Li, B., & Gao, M. (2020). Collaborative filtering recommendation algorithm based on deep neural network fusion. International Journal of Sensor Networks, 34(2), 71. 10.1504/ijsnet.2020.110460

IX. Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., & Kashef, R. (2020). Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities. Applied Sciences, 10(21), 7748. 10.3390/app10217748

X. Forouzandeh, S., Berahmand, K., & Rostami, M. (2020). Presentation of a recommender system with ensemble learning and graph embedding: a case on MovieLens. Multimedia Tools and Applications, 80(5), 7805–7832. 10.1007/s11042-020-09949-5

XI. Huang, L., Guan, C. R., Huang, Z. W., Gao, Y., Wang, C. D., & Chen, C. L. P. (2024). Broad Recommender System: An Efficient Nonlinear Collaborative Filtering Approach. IEEE Transactions on Emerging Topics in Computational Intelligence, 1–15. 10.1109/tetci.2024.3378599

XII. Iwendi, C., Ibeke, E., Eggoni, H., Velagala, S., & Srivastava, G. (2021). Pointer-Based Item-to-Item Collaborative Filtering Recommendation System Using a Machine Learning Model. International Journal of Information Technology & Decision Making, 21(01), 463–484. 10.1142/s0219622021500619

XIII. K, R. C., & Srikantaiah, K. (2021). Similarity Based Collaborative Filtering Model for Movie Recommendation Systems. 10.1109/iciccs51141.2021.9432354

XIV. Kim, T. Y., Ko, H., Kim, S. H., & Kim, H. D. (2021). Modeling of Recommendation System Based on Emotional Information and Collaborative Filtering. Sensors, 21(6), 1997. 10.3390/s21061997

XV. Mohamed, M. H., Khafagy, M. H., & Ibrahim, M. H. (2019). Recommender Systems Challenges and Solutions Survey. https://doi.org/10.1109/itce.2019.8646645

XVI. Mu, Y., & Wu, Y. (2023). Multimodal Movie Recommendation System Using Deep Learning. Mathematics, 11(4), 895. 10.3390/math11040895

XVII. Nassar, N., Jafar, A., & Rahhal, Y. (2020). Multi-criteria collaborative filtering recommender by fusing deep neural network and matrix factorization. Journal of Big Data, 7(1). https://doi.org/10.1186/s40537-020-00309-6

XVIII. Natarajan, S., Vairavasundaram, S., Natarajan, S., & Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data. Expert Systems With Applications, 149, 113248. 10.1016/j.eswa.2020.113248

XIX. Nguyen, L. V., Hong, M. S., Jung, J. J., & Sohn, B. S. (2020). Cognitive Similarity-Based Collaborative Filtering Recommendation System. Applied Sciences, 10(12), 4183. 10.3390/app10124183

XX. Nguyen, L. V., Vo, Q. T., & Nguyen, T. H. (2023). Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services. Big Data and Cognitive Computing, 7(2), 106. 10.3390/bdcc7020106

XXI. Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., & Fragopoulou, P. (2022). Collaborative filtering recommender systems taxonomy. Knowledge and Information Systems, 64(1), 35–74. 10.1007/s10115-021-01628-7

XXII. Peng, S., Siet, S., Ilkhomjon, S., Kim, D. Y., & Park, D. S. (2024). Integration of Deep Reinforcement Learning with Collaborative Filtering for Movie Recommendation Systems. Applied Sciences, 14(3), 1155. 10.3390/app14031155

XXIII. R. K., & S, M. J. A. (2021). A Hybrid Deep Collaborative Filtering Approach for Recommender Systems. Research Square (Research Square). 10.21203/rs.3.rs-651522/v1

XXIV. Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, *9*(1). 10.1186/s40537-022-00592-5

XXV. Sharma, S., Rana, V., & Malhotra, M. (2021). *Automatic recommendation system based on hybrid filtering algorithm. Education and Information Technologies.* 10.1007/s10639-021-10643-8

XXVI. Shokrzadeh, Z., Feizi-Derakhshi, M. R., Balafar, M. A., & Mohasefi, J. B. (2024). Knowledge graph-based recommendation system enhanced by neural collaborative filtering and knowledge graph embedding. *Ain Shams Engineering Journal/Ain Shams Engineering Journal*, *15*(1), 102263. 10.1016/j.asej.2023.102263

XXVII. Singh, P. K., Sinha, M., Das, S., & Choudhury, P. (2020). Enhancing recommendation accuracy of item-based collaborative filtering using Bhattacharyya coefficient and most similar item. *Applied Intelligence*, *50*(12), 4708–4731. 10.1007/s10489-020-01775-4

XXVIII. Thakker, U., Patel, R., & Shah, M. (2021). A comprehensive analysis on movie recommendation system employing collaborative filtering. *Multimedia Tools and Applications*, *80*(19), 28647–28672. 10.1007/s11042-021-10965-2

XXIX. Widiyaningtyas, T., Hidayah, I., & Adji, T. B. (2021). User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system. *Journal of Big Data*, *8*(1). 10.1186/s40537-021-00425-x

XXX. Xue, F., He, X., Wang, X., Xu, J., Liu, K., & Hong, R. (2019). Deep Item-based Collaborative Filtering for Top-N Recommendation. *ACM Transactions on Office Information Systems*, *37*(3), 1–25. 10.1145/3314578

XXXI. Yalcin, E., & Bilge, A. (2024). A novel target item-based similarity function in privacy-preserving collaborative filtering. *The Journal of Supercomputing*. 10.1007/s11227-024-06221-7

XXXII. Yu. S., Guo, M., Chen, X., Qiu, J., & Sun, J. (2023). Personalized Movie Recommendations Based on a Multi-Feature Attention Mechanism with Neural Networks. *Mathematics*, *11*(6), 1355. 10.3390/math11061355

XXXIII. Zhou, K., Yu, H., Zhao, W. X., & Wen, J. R. (2022). Filter-enhanced MLP is All You Need for Sequential Recommendation. *Proceedings of the ACM Web Conference 2022*. 10.1145/3485447.3512111