ISSN 0146-4116, Automatic Control and Computer Sciences, 2024, Vol. 58, No. 5, pp. 491–505.

An Ensemble Learning Hybrid Recommendation System Using Content-Based, Collaborative Filtering, Supervised Learning and Boosting Algorithms

Kulvinder Singh[a], *, Sanjeev Dhawan[a], **, and Nisha Bali[a], ***

[a] *Department of Computer Science Engineering, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, Haryana, India*

**e-mail: ksingh2015@kuk.ac.in*

***e-mail: sdhawan2015@kuk.ac.in*

****e-mail: Kherwal5855@gmail.com*

Received October 2, 2023; revised February 14, 2024; accepted February 27, 2024

Abstract—The evolution of recommendation systems has revolutionized user experiences by providing personalized recommendations. Although conventional systems such as collaborative and content-based filtering are reliable, they still suffer from inherent limitations. We introduce a hybrid recommendation system that combines content-based filtering using TF-IDF and cosine similarity with collaborative filtering and SVD to address these challenges. We bolster our model through supervised machine learning algorithms like decision trees (DT), random forests (RF), and support vector regression (SVR). To amplify predictive prowess, boosting algorithms including CatBoost and XGBoost are harnessed. Our experiments are performed on the benchmark dataset MovieLens 1M, which highlights the superiority of our hybrid method over more traditional alternatives with SVR being the best-performing algorithm consistently. Our hybrid model achieved an MSLE score of 2.3 and an RMSLE score of 1.5, making SVR consistently the best-performing algorithm in the recommendation system. This combination demonstrates the potential of collaborative-content hybrids supported by cutting-edge machine-learning techniques to reshape the field of recommendation systems.

Keywords: recommendation systems, hybrid, content-based filtering, collaborative filtering, decision tree, random forest, support vector regressor, CatBoost, XGBoost

DOI: 10.3103/S0146411624700615

1. INTRODUCTION

With the rapid expansion of digital material, providing precise and individualized recommendations is essential. In several industries, including social media, online shopping, and entertainment, recommendation systems have become crucial tools. A recommendation system's (RSs) primary objective is to offer consumers pertinent and personalized recommendations based on their preferences, behavior, and item attributes. RSs are critical in improving user experiences and engagement. RSs may dramatically improve user happiness and retention rates by providing personalized recommendations. When users are presented with relevant and fascinating information, they are more likely to engage with a platform for extended periods, resulting in higher user loyalty and platform success [1]. Furthermore, RSs can promote serendipitous discoveries, bringing users fresh and fascinating things or information they might not have encountered otherwise. This improves user pleasure and generates a feeling of surprise and discovery, encouraging users to explore further inside the platform or application. In light of the massive amount of data and the increasing rise of online platforms, recommendation systems have become necessary for consumers, content producers, and companies. They assist consumers in navigating the enormous information landscape and contribute to company success and profitability by increasing user engagement and boosting sales [2]. As technology advances and data availability grows, the need for effective RSs will only expand, making them a critical component of current information retrieval and personalization systems. Traditional recommendation systems are categorized into content-based and collaborative filtering methods. To find trends and provide recommendations based on the views of comparable users, collaborative filtering [3] takes advantage of user-item interactions. Content-based filtering examines the qualities and features of products recommending favouritism based preferences of users.

Hybrid recommendation systems have drawn much interest because they can handle these restrictions. Multiple recommendation strategies are combined in hybrid systems to use each approach's advantages and compensate for its shortcomings. In this research, we offer a unique hybrid recommendation system that combines cosine similarity with collaborative filtering and singular value decomposition (SVD) [4] and content-based filtering using the term frequency-inverse document frequency (TF-IDF) [5]. The hybrid model is made by combining several algorithms like decision tree (DT), random forest (RF), and support vector regressor (SVR). To further leverage the credibility of our proposed system, CatBoost, and XGBoost are implemented for boosting. We evaluate the performance of our hybrid recommendation system using the substantial user ratings and movie information included in the popular MovieLens dataset [6]. The effectiveness of our hybrid strategy over more established ones is shown by the results of our experiments, underscoring its ability to deliver precise and individualized movie recommendations.

This methodology overcomes the shortcomings of individual approaches by combining several ensemble learning algorithms, boosting algorithms, collaborative filtering, and content-based filtering, thus raising user satisfaction.

Paper contribution: This paper presents an innovative approach to improve recommendation systems by implementing hybrid models that incorporate various filtering strategies, including collaborative filtering, content-based filtering, and boosting algorithms. Moreover, it thoroughly analyzes the proposed hybrid system's performance on the MovieLens dataset and discusses potential future research directions to advance recommendation quality.

Paper objective: Our study aims to enhance recommendation systems by using contextual data, a hybrid approach, and accurate and tailored recommendations. We strive to create a reliable system that outperforms current methods, enhances crucial metrics, and streamlines the training process by using an ensemble of supervised machine learning algorithms leveraging the results. This research also aims to consider user context and preferences to improve the accuracy and relevance of recommendations.

Paper organization: Section II examines the achievements and limitations of earlier research in deep learning and machine learning based recommendation systems. Section VI describes the methods and techniques used in our hybrid recommendation system. Section VII summarizes the experimental setting utilized to measure the effectiveness of the hybrid recommendation system. The results of the experiments conducted on the MovieLens 1M Dataset to confirm the model's performance are presented in Section VIII. In Section IX, a comparative analysis of the model's performance is established to highlight its significance. Finally, Section X presents the conclusion of the paper, including future directions for this research initiative.

2. BACKGROUND WORK

This section explores previous research in recommendation systems, their contributions, and limitations, as tabulated in Table 1.

A. Deep Learning Methods Used in Recommendation Systems

Over time, deep learning approaches have significantly improved recommendation systems by providing cutting-edge ways for identifying intricate patterns and producing precise recommendations. The neural network and matrix factorization deep learning techniques are often used in recommendation systems [7]. Moreover, Dau and Salim [8] explored various kinds of neural networks used in recommendation systems and their limitations.

Neural networks have successfully modeled intricate user-item interactions and nonlinear correlations. The performance of CNNs in context-based recommendation systems has been explored [9] using convolutional neural networks and multilayer perceptrons. MLPs' ability to learn high-level representations of user preferences and item properties makes accurate predictions and tailored recommendations possible. CNNs, on the other hand, are particularly advantageous for recommendation tasks requiring sequences, such as clickstream or session-based recommendations, since they are well-suited for collecting spatial and temporal patterns in sequential data [10] explains why deep learning models need to consistently provide fruitful results in recommendation systems. Another popular deep learning technique in recommendation systems called matrix factorization seeks for the user-item interaction matrix to be broken down into low-dimensional latent components as deciphered in [11, 12]. Meaningful representations of users and objects can be analyzed by using methods like singular value decomposition (SVD) and autoencoders. In particular, autoencoders have demonstrated promise in reconstructing the input data to develop rich representations [13], which may be used to provide customized recommendations based on

Table 1. Analysis of various methods used in RSs, contributions, and limitations

Method	Paper	Contribution	Limitation	Conclusion
Deep learning	[7, 8]	Comparative analysis of deep learning methods used in RSs over the years	No proper analysis of the drawbacks mentioned	Semantic analysis score [7]: 80%; Auto-encoders perform better than CNN [8]
Deep learning	[9]	Performance scrutiny of CNNs in RSs and feature selection	Lack of a thorough understanding of the issues raised	Deep learning performs better than fuzzy DT
Deep learning	[10]	Explores the complexity of neural networks-based model performance	Lack of critical analysis of alternate solutions	Establishment of DL as a model approach
Deep learning	[11, 12, 14]	Surveys the use of matrix factorization in RSs.	No comparative analysis was shown	Matrix factorization is a trustable approach
Context-based filtering	[15]	Analysis of the use of CBF on structured data	Need for proper data manipulation techniques	RMSE score: 56%;
Collaborative filtering	[18, 29, 31]	Survey of CF frameworks, limitations, and future scopes	Directions explored in future scope are poorly analyzed	The use of CF in unstructured data is not recommended
Demographic RSs	[21]	Exploration of how the demographics of a user manipulated recommendations	Limitations were mentioned, but no thorough analysis was provided	MAE score: 326.7%
Knowledge-based RSs	[22, 23]	Survey on KBRS	Lack of a thorough understanding of the issues raised	KBRS's pros and cons established

learned user and object embeddings [14]. Deep learning techniques utilize neural networks and matrix factorization to identify patterns, model user preferences, and provide tailored recommendations for consumers.

B. Machine Learning Models Used in Recommendation Systems

(1) *Content-Based Filtering RSs (CBF):* By comparing the user profile and item description, CBRS uses CBF to propose products [15]. The recommendation system thus suggests products that are comparable to previous records of user preferences. The similarity of items is determined based on their attributes concludes that if a user provides positive reviews of a comedy film, the algorithm can learn to suggest this genre to the user in the future.

(2) *Collaborative Filtering RSs:* Several research works have explored collaborative filtering in RSs like Ramesh et al. [16], Kharroubi et al. [17], Ekstrand [18], and Bobadilla [19]. They critically surveyed the methodology, its limitations, and future research directions. The idea of item-based collaborative filtering, which uses similarity between things to produce recommendations, was first described in a study by Burke [20]. Their research showed how this strategy can increase recommendation accuracy. Their study's dependence on explicit user evaluations might be scarce and skewed.

(3) *Demographic RSs:* Based on user demographics like age, education, employment, location, etc., demographic operations are based. Clustering algorithms group target customers into specific categories based on demographic data. However, the same selection of things will be recommended if the user's demographic characteristics do not change over time. They might thereby overlook a fresh and valuable recommendation. The accuracy of RS can be increased by knowing a user's demographics [21].

(4) *Knowledge-Based RSs:* Knowledge-based RSs provide recommendations based on connections and similar preference probability between the user and goods. Case-based reasoning in knowledge-based RSs divides the user's requirements into multiple instances based on various criteria and provides recommendations that closely match the user's expected selection [22]. Constraint-based RS is a different kind of

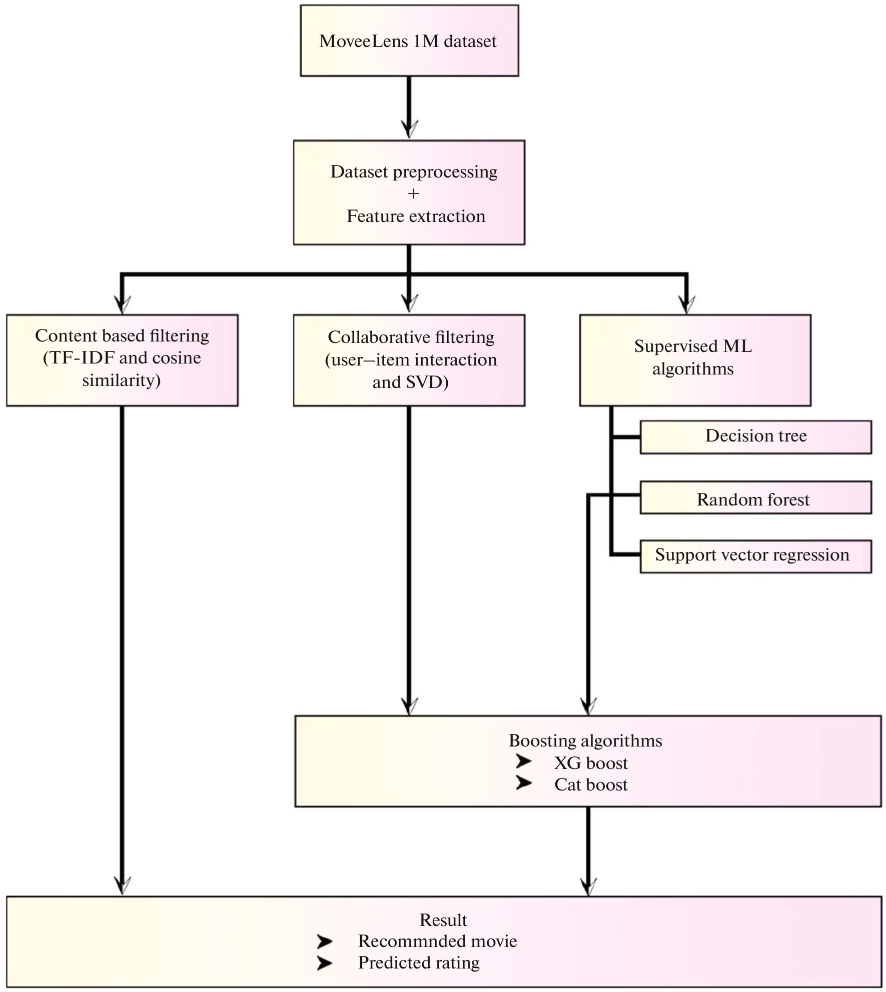

Fig. 1. Proposed model.

KBRS that adapts to the user's preferences and only suggests products that fit those preferences [23]. During the nonavailability of an easily accessible user preference item, a set of close equivalent items is suggested. By using semantic web technologies, a user's preference knowledge base with a range of viewpoints is built using ontologies.

(5) *Hybrid RSs:* Hybrid recommendation systems combine several models' characteristics to achieve an admirable result. Burke [20], Ghazanfar and Prugel-Bennett [24], and Çano and Morisio [25] analyzed different hybrid approaches used in RSs over time. Li and Kim [26] studied clustering-based hybrid system on the MovieLens dataset, while Zhang et al. [27], and Fan et al. [28] had combined random forest and *K*-nearest neighbor (KNN) in its hybrid model. Figure 1 shows the block diagram of our proposed approach.

Creating data collection and augmentation methods, using contextual data, and experimenting using alternate recommendation systems to overcome these constraints is a fruit-providing stratagem [30]. By overcoming these restrictions, recommendation systems can develop more reliable, accurate, and flexible to accommodate a variety of user demands.

3. METHODOLOGY

This section outlines the methodology and algorithms used in our hybrid recommendation system.

A. Dataset

The MovieLens 1M dataset [6], a frequently used benchmark dataset in recommendation systems, served as the basis for our experiments. This article details the ratings and free-text tagging activities of

Table 2. Description of files contained in MovieLens 1M dataset

Files	Content
ratings.dat	UserID, MovieID, rating, timestamp
users.dat	UserID, gender, age, occupation, zip-code

MovieLens, a movie recommendation service. The 1000209 anonymous ratings of around 3900 movies provided by 6040 users of MovieLens [6] from year 2000 to 2003. The data file used in this research paper contains 20000263 anonymous ratings from 138493 users who rated 27278 films from 1995 to 2015. It offers a wide range of user-item interactions, allowing a thorough assessment initiative for our hybrid recommendation system. Table 2 describes the files contained in the MovieLens 1M dataset.

B. Collaborative Filtering with SVD

Collaborative filtering approaches are applied to identify the underlying patterns in the user-item interaction matrix. SVD is deterministically used in [32} to break down the matrix into latent components. The data's dimensionality is reduced through SVD, revealing hidden relationships between users and things. The missing ratings are forecasted and recommendations are made to the users by using low-rank approximations of the original user-item matrix derived using SVD [33]. The SVD-based forecasts are one of the sources of recommendations in the collaborative filtering part of the hybrid RSs.

$$R = USV^T.$$

Here, R is the user–item interaction matrix, U is the user matrix, S is the diagonal singular value matrix, V is the item matrix.

Using matrix factorization, CF partitions the user-item interaction matrix R into three separate matrices: U (representing user preferences), S (a diagonal matrix holding singular values), and V^T (representing item attributes). With the help of this decomposition, recommendation algorithms can identify hidden patterns in user-item interactions, resulting in precise and individualized recommendations. Its capacity to forecast missing values in the interaction matrix presents an important strategy for creating effective and efficient recommendation systems.

C. Content-Based Filtering with TF-IDF and Cosine Similarity

Movie qualities like genre, director, and cast focus on including content-based filtering in the hybrid method. The presence or absence of characteristics is represented as a vector using the term frequency-inverse document frequency (TF-IDF) weighting. Terms that are common inside a movie but uncommon throughout the full movie collection are given greater weight by TF-IDF [34]. Based on its characteristic profile, the weighting system effectively represents the distinctiveness of each film. Cosine similarity is used to compare movies based on their attribute vectors. Users can find films with comparable qualities using TF-IDF and cosine similarity [35].

D. Use of SVM Classifier

We incorporate the SVM classifier into our hybrid technique to improve the recommendation process. The content-based characteristics retrieved using TF-IDF train the SVM classifier [36]. SVMs have a reputation for handling high-dimensional data and capturing intricate patterns. More precise predictions and recommendations are provided with the help of the SVM classifier, which learns decision boundaries between movies based on their content characteristics. SVM classifier's prowess in handling highly dimensional content-based characteristics strengthens the overall recommendation procedure [37].

$$mk + z = 0.$$

Here, m is the weight perpendicular to the hyper plane, k is the feature vector, and z is the biased term.

Analyzing the feature vectors of the data points makes it possible to distinguish between several classes of data points. A good outcome denotes belonging to one class, whereas a negative outcome denotes belonging to the opposite class. This representation serves as the foundation for accurately categorizing and suggesting objects, given the bias index value of z.

E. Ensemble Learning Algorithms

(1) *Decision Tree* (*DT*): Decision tree is a common supervised machine learning technique for classification and regression applications. Each internal node in the decision tree corresponds to a decision based on a feature, while each leaf node corresponds to a class label or a projected value. It recursively splits the data into subsets depending on the most critical attributes [38].

(2) *Random Forest* (*RF*): Random forest is widely used for classification, regression, and machine learning tasks. It can analyze user-item interactions, past preferences, and demographic information to create several decision trees, each of which forecasts the likelihood of a user's choice for a specific item since RF can manage enormous and diverse datasets [39]. RF displayed improved performance compared to DT. RF's ensemble technique successfully decreased overfitting and increased predicted accuracy by combining many decision trees, yielding a noticeably lower MAPE score of 99.6. As a result, there is now a better balance between collecting complex patterns and successfully generalizing to fresh data.

(3) *Support Vector Regression* (*SVR*): SVR accomplishes nonlinear regression by translating the data into a higher-dimensional feature space and applying the SVM principles. SVR may be used in recommendation systems to forecast continuous or numerical ratings, such as consumer preferences or product rankings. In this situation, SVR builds a model that can forecast ratings for objects that users still need to give a rating by learning from prior user-item interaction data. In our ensemble learning model, SVR performed relatively better than the other algorithms. The model's success is potentially ascribed to the application of matrix factorization techniques, which capture complex user-item interactions and produce extremely exact recommendations with minimal error. The minimization function is:

Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum\left(\varepsilon_i + \varepsilon_i^*\right),$$

$$y_i - wx_i - b \le \varepsilon_i,$$

$$y_i - wx_i - b \ge -\varepsilon_i,$$

where w represents the weight vector and b is the biased term, Epsilon and Epsilon' are the slack variables for the positive and negative deviations, C is the regularization parameter.

SVR's competent success lies in determining the combinations w and b that meet the stated constraints and minimize the objective function. The regularization parameter C controls the trade-off between getting a lower training error and tolerating more deviations.

F. Boosting Algorithms

(1) *XGBoost:* XGBoost has been used in recommendation systems over time to provide consumers with personalized and pertinent recommendations. The technique is particularly suited for collaborative filtering tasks where user-item interactions are frequently represented as sparse matrices due to its capacity to manage sparse data and capture complicated feature relationships. XGBoost leverages a group of decision trees to reduce prediction mistakes and applies boosting techniques to add weak learners iteratively. The aim of recommendation systems is a binary value that represents user preference (for example, clicked or not clicked), and the working algorithm uses user-item attributes and previous interaction data as input [40]. Algorithm 1 presents the parallel prefix sum algorithm for the XGBoost algorithm. The system creates item ratings for each user based on prior user-item interactions and feature representations and then uses those scores to propose the top-ranked things to users.

Algorithm 1: XGBoost Binary Classification
Require *dataset(x,y)*
Define *num_r, l_rate, and depth*
Initialize Prediction
initial_pred = log (mean(y) / (1 - mean(y)))
    Training Loop
    Compute gradient and hessian of loss function
     for *i=1 to i=num_r*
        grad_i = -∂L(y_i, pred_i) / ∂pred_i
        hess_i = ∂^2*L(y_i, pred_i) / ∂pred_i*^{2}
     end for

Prediction
Declare *tree*
*updated_pred = pred + l_rate * tree*
Binary Classification
for *i=1 to i=num_r*
final_pred = 1 / (1 + exp(-pred_i))
end for
mean(y) = (Σ y_i) / n
log(odds) = log(P(y=1) / P(y=0))
Prob(y=1) = 1/(1+e^(-log(odds)))

(2) *CatBoost:* CatBoost is a potent gradient-boosting framework that can effectively handle categorical characteristics, making it a good fit for recommendation systems that frequently interact with high-cardinality categorical data. CatBoost reduces the requirement for human feature engineering and preprocessing while delivering improved performance using a novel method that uses the characteristics of categorical features. CatBoost may be used in the context of recommendation systems to create reliable models that accurately forecast user-item interactions, such as user preferences or item ratings. Large and sparse categorical datasets can be handled, missing values can be handled automatically, and intricate connections between categorical variables can be captured intuitively. A significant tool for real-time or high-throughput recommendation systems [41], CatBoost's capacity for handling large-scale information and its quick training skills enable it to deliver accurate and individualized recommendations.

G. Evaluation Metrics

Standard evaluation metrics like root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), mean square logarithmic error (MSLE), and root mean square logarithmic error (RMSLE) are used to evaluate the effectiveness of our hybrid recommendation system.

4. EXPERIMENTAL SECTION

This section outlines the experimental setup for evaluating hybrid recommendation system performance. The programming framework, libraries, dataset, preprocessing steps, and parameter settings employed in our experiments are explained further.

A. Dataset Preprocessing

The MovieLens Dataset is thoroughly cleaned and preprocessed.

(1) Data cleaning: To ensure data consistency, we handle missing values and eliminate duplicates.

(2) Data transformation: The process of transforming category information into analytically sound numerical representations, such as genres, is done in this step.

(3) Data splitting: This is the process of dividing a dataset into two parts: one for training

and one for testing. This helps assess the performance of the recommendation system.

B. Ensemble Approach

To provide a robust recommendation system, we use an ensemble technique to maximize the use of the individual capabilities of DT, RF, and SVR. The ensemble model exploits the complementary strengths of these machine learning algorithms, resulting in a more reliable and precise recommendation system. Random forests excel at reducing overfitting and enhancing prediction stability; support vector regression is skilled at handling complex, high-dimensional data and minimizing prediction errors. Decision trees offer interpretability and can capture non-linear relationships in the data. The ensemble model benefits from the distinctive qualities of each approach and mitigates their specific limits through this collection of algorithms. This is done by extracting complex patterns and insights from the data enabling us to provide customers with recommendations that are more accurate and unique to them.

Table 3. Evaluation metrics table

Model	Before boosting					After boosting									
						XG boost					Cat boost				
	RMSE	MAE	MAPE	MSLE	RMSLE	RMSE	MAE	MAPE	MSLE	RMSLE	RMSE	MAE	MAPE	MSLE	RMSLE
SVD	2367.4	1935.5	206.8	2.3	1.52	3481.3	3021.2	99.6	39.4	6.2	3481.3	3021.2	99.6	39.4	6.2
DT	3481.3	3021.2	206.8	39.9	6.3	3481.3	3021.2	99.6	39.4	6.2	3481.3	3021.2	99.6	39.4	6.2
RF	3481.3	3021.2	99.6	39.6	6.2	3481.3	3021.2	99.6	39.4	6.2	3481.3	3021.2	99.6	39.4	6.2
SVR	2367.4	1935.5	206.8	2.3	1.5	3481.3	3021.2	99.6	39.4	6.2	3481.3	3021.2	99.6	39.4	6.2

5. RESULT ANALYSIS

The graphical and numerical results obtained from our hybrid model are discussed in detail in this section. The scores of the evaluated metrics on our hybrid model are tabulated in Table 3. The values and distribution of movies in the dataset are graphically illustrated in Fig. 2. Figure 3 represents the heat map demonstrating the correlations of the various features given in the dataset. Figure 4 represents the distribution of movie ratings in the MovieLens dataset. The graph shows the frequency or number of movies that received a particular rating. It also provides insights into the popularity and distribution of ratings among the movies in the dataset.

The projected ratings of the recommendations per movie produced by our hybrid recommendation system are shown in Fig. 5. The sample input taken is "Jumanji (1995)" whose results are plotted in Fig. 6. The movie and anticipated rating for each point on the graph correspond to a different film. The distribution and range of the expected ratings across the suggested films are analyzed using this visualization. The text

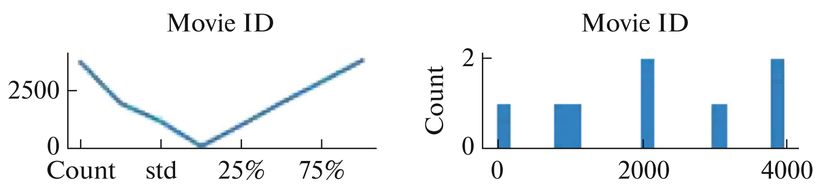

Fig. 2. Movies distribution in dataset.

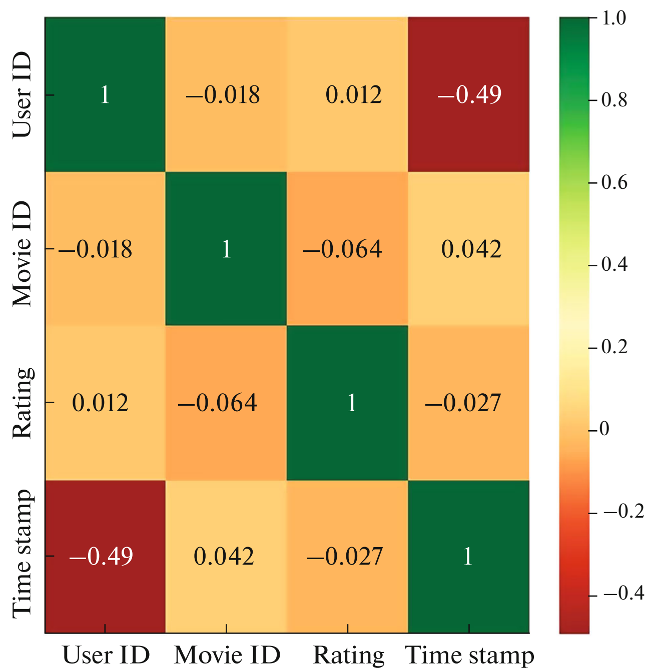

Fig. 3. Heat map.

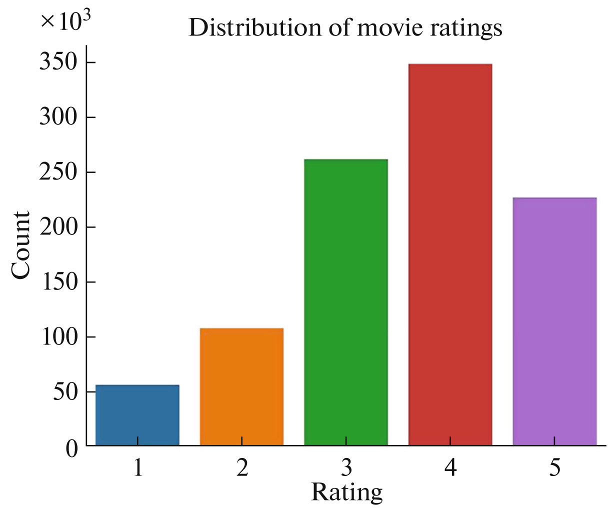

Fig. 4. Distribution of movie ratings.

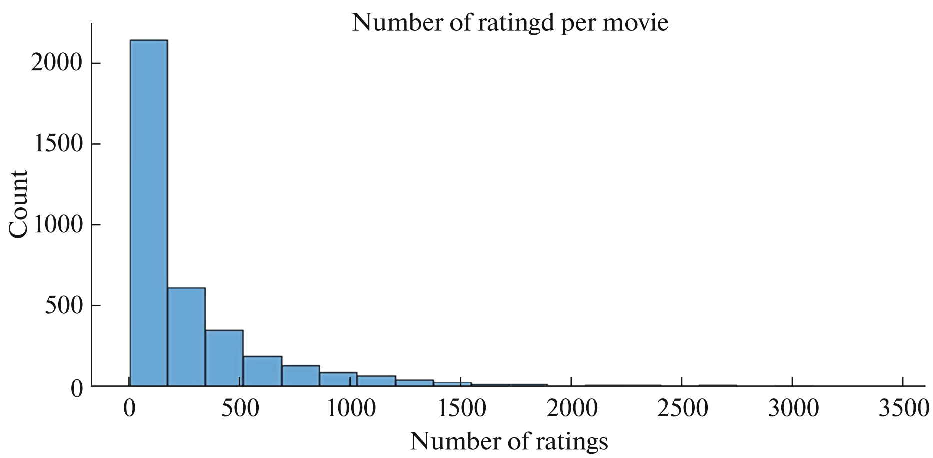

Fig. 5. Number of ratings per movie.

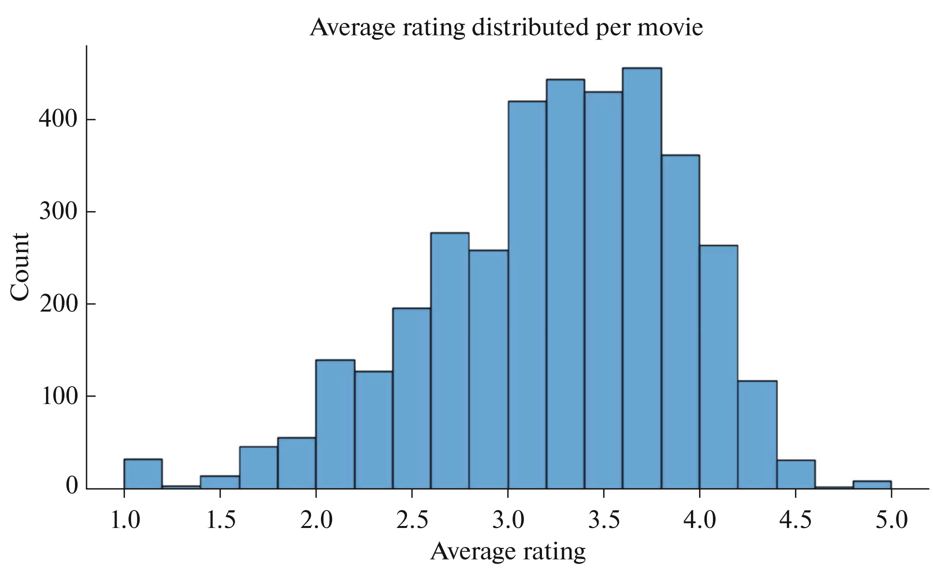

Fig. 6. Average rating distributed per movie.

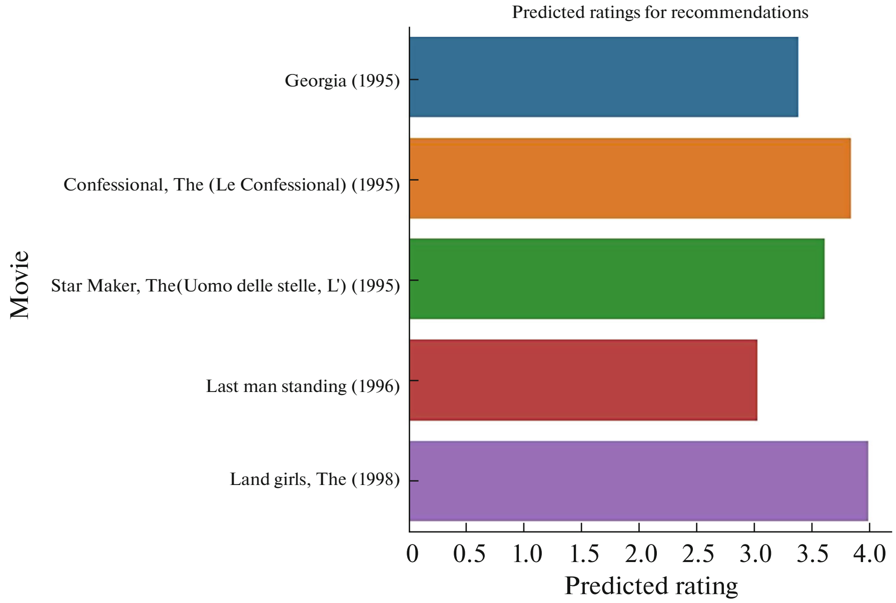

Fig. 7. Predicted ratings for recommendations.

provides information on the reliability and quality of recommendations and allows for evaluation of the recommendation system's ability to predict ratings.

The number of ratings each film in the dataset has accumulated is depicted in Fig. 7. It offers details on the distribution of user interactions with films and the levels of popularity or engagement of certain aspects. Figure 8 spots trends such as films with many ratings, which indicate a high level of user interest or popularity, and films with fewer ratings, which indicate a lower level of user involvement. The graph reveals an analysis of whether the reviews for films were largely favorable or negative and whether there was a tendency for reviews to tilt higher or lower on average. It assists in identifying films that have received favorable or unfavorable user reviews, reflecting their perceived value or appeal. Table 4 shows the movies recommended by our hybrid model and their predicted ratings on a custom prompt.

The top 10 films in the dataset with the most ratings are shown in Fig. 8. The height of each bar in the graph, which corresponds to a particular movie, indicates how many ratings it received. The most frequently rated films' popularity and user involvement are revealed by this visualization. It makes it possible to determine which films received a sizable number of ratings, suggesting their vast viewing or interest. By concentrating on the top 10 films, it draws emphasis to the particular films that received the most comments and attention from users.

Figure 9 shows how our proposed system is performing. MAPE, MSLE, and RMSLE scores of SVD, DT, RF, and SVR are plotted both before and after applying XGBoost and Cat Boost. Analysis can be made that our hybrid model shows better performance after boosting. Table IV shows the scores of the evaluated metrics over our hybrid model, both before boosting and after boosting. It also highlights a comparison between the performance of both XGBoost and Cat Boost. It shows how the predictions of the recommendation system are leveraged post-application of the boosting algorithms.

Table 4. Model performance on custom data input

Custom Input	Tom and Huck (1995)
Number of Recommendations	4
Recommended Movie	**Predicted Rating**
The Amazing Panda Adventure (1995)	3.6260408308302465
Casper (1995)	3.792875762757196
Far From Home: The Adventures of Yellow Dog (1995)	4.225775777568836
The Secret Adventures of Tom Thumb (1993)	3.967961573272767

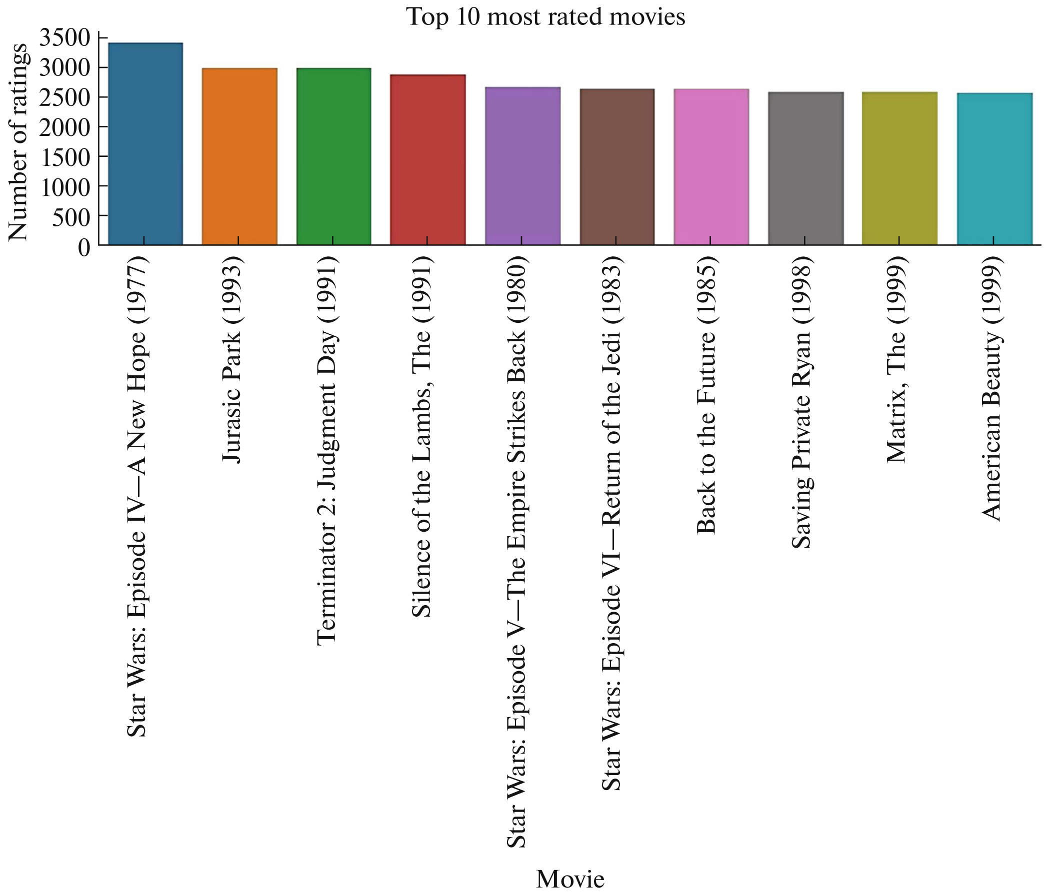

Fig. 8. Graph of most rated movies VS ratings.

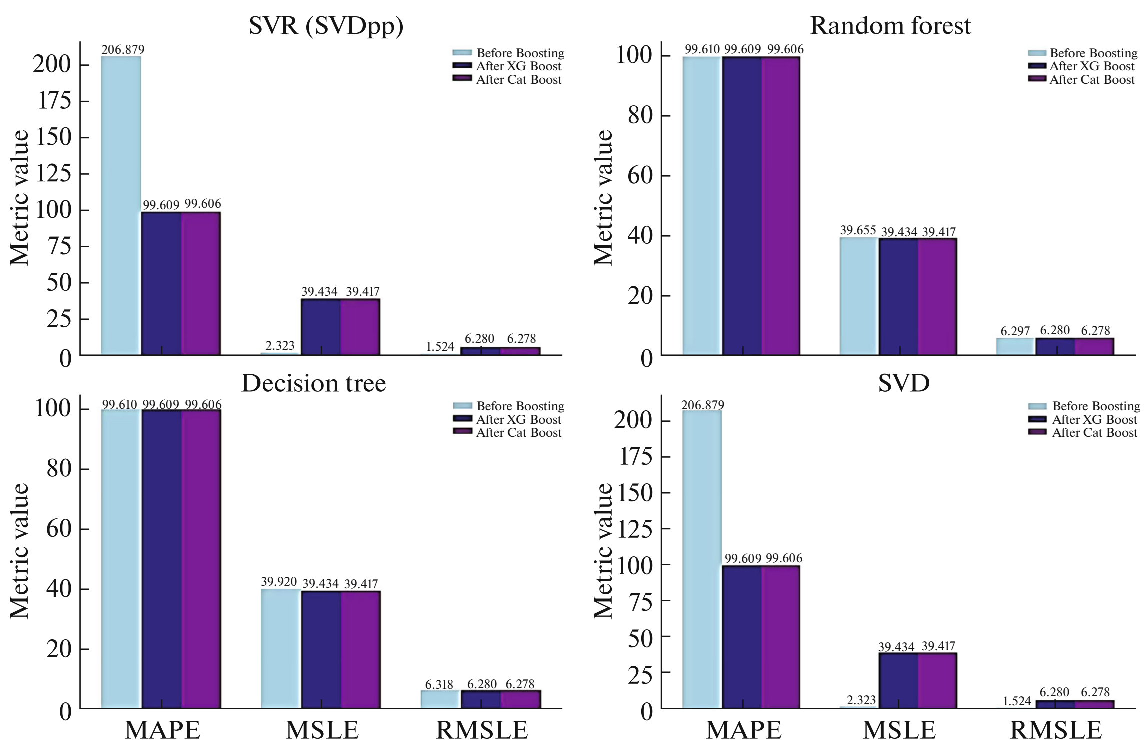

Fig. 9. Model comparison before and after boosting.

Table 5. Comparative analysis

Paper	Model used	Better performing algorithm	Metrics	Score
[20]	Supervised learning Hybrid model	NB	MSLE	3.8451
Our model	Hybrid	SVD (before boosting)	MSLE	2.323360
[42]	Autoencoder based collaborative filtering	U-AutoRec	RMSLE	1.831
Our model	Hybrid	SVD (before boosting)	RMSLE	1.524257
[43]	CF-NADE and RBM	U-CF-NADE-S	RMSE	2437.9
Our model	Hybrid	SVD (before boosting)	RMSE	2367.480008

Greater precision is shown by a lower RMSE number, which denotes that, on average, the projected ratings are more in line with the actual ratings. Our hybrid approach generates recommendations with reasonably accurate projected ratings, as indicated by the RMSE values. A lower MAE number denotes higher prediction accuracy, similar to RMSE. Our hybrid recommendation system demonstrates remarkably little error in predicting ratings, which highlights its capacity to produce recommendations that closely match the preferences of the user. The MAPE scores in Table 3 show that the anticipated ratings of our hybrid system are, on average, a certain percentage of the actual ratings. In terms of properly calculating the logarithmic variations between expected and actual ratings, the MSLE score shows good performance. Our hybrid recommendation system exhibits reasonably accurate predictions of ratings on the logarithmic scale, with the evaluated RMSLE scores, as shown in Table 3.

6. COMPARATIVE ANALYSIS

In this section, we present a comparative analysis of the performance of the hybrid ensemble model used by our research and the models used in papers [20, 42]. Table 5 presents this comparison analysis where all the papers used the MovieLens 1M Dataset. We intricately present a comparative analysis of the algorithms and methods used in these papers and conclude that our model performs consistently well.

7. CONCLUSIONS

On the MovieLens dataset, our hybrid recommendation system, which combines collaborative filtering with SVD, content-based filtering with TF-IDF and cosine similarity, an ensemble learning framework containing DT, RF, SVR algorithms, and XGBoost and CatBoost for boosting, has shown encouraging results. This novel hybrid architecture proposed in this research proves to bypass all threshold experimental parameters and satisfies as one of the strongest frameworks to be implemented in recommendation systems. By combining several approaches, we may make use of each one's advantages while minimizing the drawbacks of each methodology alone. Overall, our hybrid approach offers a thorough and efficient method of producing individualized movie recommendations. Low RMSE and MAE values illustrate the success of our hybrid model at producing individualized movie recommendations.

To enhance the quality of recommendations, researchers can focus on integrating deep learning techniques, contextual information, and social network analysis. Obtaining feedback from users and conducting user studies can provide valuable insights into the system's usability and user satisfaction. This ensures that the models reach their maximum potential and deliver extremely precise, personalized recommendations across various domains by following these approaches.

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. Shah, K., Salunke, A., Dongare, S., and Antala, K., Recommender systems: An overview of different approaches to recommendations, *2017 Int. Conf. on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 2018, IEEE, 2018, pp. 1–4.
https://doi.org/10.1109/iciiecs.2017.8276172
2. Mansur, F., Patel, V., and Patel, M., A review on recommender systems, *2017 Int. Conf. on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 2018, IEEE, 2018, pp. 1–6.
https://doi.org/10.1109/iciiecs.2017.8276182
3. Zhang, R., Liu, Q.-D., Chun-Gui, Wei, J.-X., and Huiyi-Ma, Collaborative filtering for recommender systems, *2014 Second Int. Conf. on Advanced Cloud and Big Data*, Huangshan, China, 2014, IEEE, 2014, pp. 301–308.
https://doi.org/10.1109/cbd.2014.47
4. Rahmah, A., Santoso, H.B., and Hasibuan, Z.A., Exploring technology-enhanced learning key terms using TF-IDF weighting, *2019 Fourth Int. Conf. on Informatics and Computing (ICIC)*, Semarang, Indonesia, 2019, IEEE, 2019, pp. 1–4.
https://doi.org/10.1109/icic47613.2019.8985776
5. Jiang, Sh., Li, J., and Zhou, W., An application of SVD++ method in collaborative filtering, *2020 17th Int. Computer Conf. on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China, 2020, IEEE, 2020, pp. 192–197.
https://doi.org/10.1109/iccwamtip51612.2020.9317347
6. Harper, F.M. and Konstan, J.A., The movielens datasets, *ACM Trans. Interact. Intell. Syst.*, 2016, vol. 5, no. 4, p. 19.
https://doi.org/10.1145/2827872
7. Gündoğan, E. and Kaya, M., A novel hybrid paper recommendation system using deep learning, *Scientometrics*, 2022, vol. 127, no. 7, pp. 3837–3855.
https://doi.org/10.1007/s11192-022-04420-8
8. Da'u, A. and Salim, N., Recommendation system based on deep learning methods: a systematic review and new directions, *Artif. Intell. Rev.*, 2020, vol. 53, no. 4, pp. 2709–2748.
https://doi.org/10.1007/s10462-019-09744-1
9. Rosewelt, L.A. and Renjit, J.A., A content recommendation system for effective e-learning using embedded feature selection and fuzzy DT based CNN, *J. Intell. Fuzzy Syst*, 2020, vol. 39, no. 1, pp. 795–808.
https://doi.org/10.3233/JIFS-191721
10. Jannach, D., de Souza P. Moreira, G., and Oldridge, E., Why are deep learning models not consistently winning recommender systems competitions yet?, *Proc. Recommender Systems Challenge 2020*, New York: Association for Computing Machinery, 2020, pp. 44–49.
https://doi.org/10.1145/3415959.3416001
11. Koren, Ye., Bell, R., and Volinsky, Ch., Matrix factorization techniques for recommender systems, *Computer*, 2009, vol. 42, no. 8, pp. 30–37.
https://doi.org/10.1109/mc.2009.263
12. Yu, H.-F., Hsieh, C.-J., Si, S., and Dhillon, I.S., Parallel matrix factorization for recommender systems, *Knowl. Inf. Syst.*, 2014, vol. 41, no. 3, pp. 793–819.
https://doi.org/10.1007/s10115-013-0682-2
13. Zhang, G., Liu, Ya., and Jin, X., A survey of autoencoder-based recommender systems, *Front. Comput. Sci.*, 2020, vol. 14, no. 2, pp. 430–450.
https://doi.org/10.1007/s11704-018-8052-6
14. Xue, H.-J., Dai, X., Zhang, J., Huang, Sh., and Chen, J., Deep matrix factorization models for recommender systems, *Proc. Twenty-Sixth Int. Joint Conf. on Artificial Intelligence*, Int. Joint Conf. on Artificial Intelligence Organization, 2017, pp. 3203–3209.
https://doi.org/10.24963/ijcai.2017/447
15. Son, J. and Kim, S.B., Content-based filtering for recommendation systems using multiattribute networks, *Expert Syst. Appl.*, 2017, vol. 89, pp. 404–412.
https://doi.org/10.1016/j.eswa.2017.08.008
16. Ramesh, A., Anusha, J., and Tauro, C.J.M., A novel, generalized recommender system for social media using the collaborative-filtering technique, *ACM SIGSOFT Softw. Eng. Notes*, 2014, vol. 39, no. 3, pp. 1–4.
https://doi.org/10.1145/2597716.2597721
17. Kharroubi, S., Dahmani, Yo., and Nouali, O., Improving collaborative recommendation based on item weight link prediction, *Turk. J. Electr. Eng. Comput. Sci.*, 2022, vol. 30, no. 1, pp. 109–126.
https://doi.org/10.3906/elk-2008-26

18. Ekstrand, M.D., Collaborative filtering recommender systems, *Found. Trends Human–Computer Interact.*, 2011, vol. 4, no. 2, pp. 81–173.
https://doi.org/10.1561/1100000009
19. Bobadilla, J., Hernando, A., Ortega, F., and Bernal, J., A framework for collaborative filtering recommender systems, *Expert Syst. Appl.*, 2011, vol. 38, no. 12, pp. 14609–14623.
https://doi.org/10.1016/j.eswa.2011.05.021
20. Burke, R., Hybrid recommender systems: Survey and experiments, *User Model. User-Adapted Interact.*, 2002, vol. 12, no. 4, pp. 331–370.
https://doi.org/10.1023/A:1021240730564
21. Wang, Yu., Chan, S.C.-F., and Ngai, G., Applicability of demographic recommender system to tourist attractions: A case study on trip advisor, *2012 IEEE/WIC/ACM Int. Conf.s on Web Intelligence and Intelligent Agent Technology*, Macau, China, 2012, IEEE, 2012, pp. 97–101.
https://doi.org/10.1109/wi-iat.2012.133
22. Tarus, J.K., Niu, Zh., and Mustafa, G., Knowledge-based recommendation: A review of ontology-based recommender systems for e-learning, *Artif. Intell. Rev.*, 2018, vol. 50, no. 1, pp. 21–48.
https://doi.org/10.1007/s10462-017-9539-5
23. Martinez, L., Barranco, M.J., Perez, L.G., and Espinilla, M., A knowledge based recommender system with multigranular linguistic information, *Int. J. Comput. Intell. Syst.*, 2008, vol. 1, no. 3, p. 225.
https://doi.org/10.2991/ijcis.2008.1.3.4
24. Ghazanfar, M.A. and Prugel-Bennett, A., A scalable, accurate hybrid recommender system, *2010 Third Int. Conf. on Knowledge Discovery and Data Mining*, Phuket, Thailand, 2010, IEEE, 2010, pp. 94–98.
https://doi.org/10.1109/wkdd.2010.117
25. Çano, E. and Morisio, M., Hybrid recommender systems: A systematic literature review, *Intell. Data Anal.*, 2017, vol. 21, no. 6, pp. 1487–1524.
https://doi.org/10.3233/ida-163209
26. Li, Q. and Kim, B.M., Clustering approach for hybrid recommender system, *Proc. IEEE/WIC Int. Conf. on Web Intelligence (WI 2003)*, Halifax, Canada, 2003, IEEE, 2003, pp. 33–38.
https://doi.org/10.1109/WI.2003.1241167
27. Zhang, H.-R., Min, F., He, X., and Xu, Y.-Y., A hybrid recommender system based on user-recommender interaction, *Math. Probl. Eng.*, 2015, vol. 2015, p. 145636.
https://doi.org/10.1155/2015/145636
28. Fan, H., Wu, K., Parvin, H., Beigi, A., and Pho, K.-H., A hybrid recommender system using KNN and clustering, *Int. J. Inf. Technol. Decision Making*, 2021, vol. 20, no. 2, pp. 553–596.
https://doi.org/10.1142/s021962202150005x
29. Cacheda, F., Carneiro, V., Fernández, D., and Formoso, V., Comparison of collaborative filtering algorithms, *ACM Trans. Web*, 2011, vol. 5, no. 1, p. 2.
https://doi.org/10.1145/1921591.1921593
30. Da Silva, F.L., Slodkowski, B.K., da Silva, K.K.A., and Cazella, S.C., A systematic literature review on educational recommender systems for teaching and learning: Research trends, limitations and opportunities, *Educ. Inf. Technol.*, 2023, vol. 28, no. 3, pp. 3289–3328.
https://doi.org/10.1007/s10639-022-11341-9
31. Jalili, M., Ahmadian, S., Izadi, M., Moradi, P., and Salehi, M., Evaluating collaborative filtering recommender algorithms: A survey, *IEEE Access*, 2018, vol. 6, pp. 74003–74024.
https://doi.org/10.1109/access.2018.2883742
32. Polat, H. and Du, W., SVD-based collaborative filtering with privacy, *Proc. 2005 ACM Symp. on Applied Computing*, Santa Fe, Mexico, 2005, New York: Association for Computing Machinery, 2005, pp. 791–795.
https://doi.org/10.1145/1066677.1066860
33. Zhang, Sh., Wang, W., Ford, J., Makedon, F., and Pearlman, J., Using singular value decomposition approximation for collaborative filtering, *Seventh IEEE Int. Conf. on E-Commerce Technology (CEC'05)*, Munich, 2005, IEEE, 2005, pp. 257–264.
https://doi.org/10.1109/ICECT.2005.102
34. Tata, S. and Patel, J.M., Estimating the selectivity of *tf-idf* based cosine similarity predicates, *ACM SIGMOD Rec.*, 2007, vol. 36, no. 2, pp. 7–12.
https://doi.org/10.1145/1328854.1328855
35. Yunanda, G., Nurjanah, D., and Meliana, S., Recommendation system from Microsoft News data using TF-IDF and cosine similarity methods, *Building Inf., Technol. Sci.*, 2022, vol. 4, no. 1, pp. 277–284.
https://doi.org/10.47065/bits.v4i1.1670
36. Xu, J.A. and Araki, K., A SVM-based personal recommendation system for TV programs, *2006 12th Int. Multi-Media Modelling Conf.*, Beijing, 2006, IEEE, 2006, pp. 401–404.
https://doi.org/10.1109/MMMC.2006.1651358

37. Mantovani, R.G., Rossi, A.L.D., Alcobaça, E., Vanschoren, J., and de Carvalho, A.C.P.L.F., A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers, *Inf. Sci.*, 2019, vol. 501, pp. 193–221.
https://doi.org/10.1016/j.ins.2019.06.005
38. Linda, S. and Bharadwaj, K.K., A decision tree based context-aware recommender system, *Intelligent Human Computer Interaction*, Tiwary, U., Ed., Lecture Notes in Computer Science, vol. 11278, Cham: Springer, 2018, pp. 293–305.
https://doi.org/10.1007/978-3-030-04021-5_27
39. Zhang, H.-R. and Min, F., Three-way recommender systems based on random forests, *Knowledge-Based Syst.*, 2016, vol. 91, pp. 275–286.
https://doi.org/10.1016/j.knosys.2015.06.019
40. Xu, L., Liu, J., and Gu, Yu., A recommendation system based on extreme gradient boosting classifier, *2018 10th Int. Conf. on Modelling, Identification and Control (ICMIC)*, Guiyang, China, 2018, IEEE, 2018, pp. 1–5.
https://doi.org/10.1109/icmic.2018.8529885
41. Hancock, J.T. and Khoshgoftaar, T.M., CatBoost for big data: An interdisciplinary review, *J. Big Data*, 2020, vol. 7, no. 1, p. 94.
https://doi.org/10.1186/s40537-020-00369-8
42. Sedhain, S., Menony, A.K., Sannery, S., and Xie, L., AutoRec: Autoencoders meet collaborative filtering, *WWW'2015 Companion: Proc. 24th Int. Conf. World Wide Web*, Florence, 2015, New York: Association for Computing Machinery, 2015, pp. 111–112.
https://doi.org/10.1145/2740908.2742726
43. Zheng, Y., Tang, B., Ding, W., and Zhou, H., A neural autoregressive approach to collaborative filtering, *33rd Int. Conf. Mach. Learn. ICML 2016*, 2016, vol. 2, pp. 1179–1190. https://proceedings.mlr.press/v48/zheng16.html.

Publisher's Note. Allerton Press remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
AI tools may have been used in the translation or editing of this article.