

ENHANCING MOVIE RECOMMENDATIONS THROUGH ARTIFICIAL INTELLIGENCE AND PREDICTIVE ANALYTICS

^{*1}Sidra Mushtaq, ²Shagufta Munir, ³Basit Bashir, ⁴Sana Parveen,
⁵Muhammad Nadeem

¹Department of Computer Science, National University of Modern languages Sub-Campus,
Lahore, Pakistan

²Department of Computer Science, National University of Modern languages Sub-Campus,
Lahore, Pakistan

³Department of Computer Science, University of Engineering and Technology, Lahore, Pakistan

⁴Department of Information Technology, Government College University Hyderabad Pakistan

⁵Department of Computer Science, The Islamia University of Bahawalpur, Bahawalpur, Pakistan

[*¹sidra.mushtaq@numl.edu.pk](mailto:sidra.mushtaq@numl.edu.pk), [*²shagufta.munir@numl.edu.pk](mailto:shagufta.munir@numl.edu.pk), [*³basitbashir42@gmail.com](mailto:basitbashir42@gmail.com)

[*⁴sanahaq6677@gmail.com](mailto:sanahaq6677@gmail.com), [*⁵nb64852@gmail.com](mailto:nb64852@gmail.com)

DOI:-

Keywords:

Artificial Intelligence,
Analytics, Movie
Recommendations, Support
vector machine

Article History

Received: 15 May, 2026

Accepted: 18 June, 2026

Published: 20 June, 2026

Copyright @Author

Corresponding Author: *

Sidra Mushtaq

Abstract

Recommendation movie systems primarily aim to offer customers useful product suggestions by relying solely on past interactions. Recommender systems stand out as particularly useful in businesses due to their application of machine learning technologies. This form of recommendation filtering is used to attempt to predict a user's selection. With the help of data, it forecasts, aims, and even identifies what the consumers' needs are from an ever-growing assortment of options. Multiple markers such as a user's search history, their age and background, what they have bought previously, and a lot more, can help locate the users. It helps users locate products and services which are unavailable or difficult for them to find. People now find it difficult to locate and sort through their preferred content due to the deluge of information. This issue has been addressed by recommendation systems (RSs) however, conventional Appen recommendation systems, such as content-based and collaborative filtering, have serious issues with data scalability, data scarcity, and the cold-start problem, all of which call for sophisticated solutions. Data sparsity and a failure to consider the variety of recommended outcomes are two issues with traditional recommendation systems. While the second experiment extended predictions to 4800 movies and produced a SVM 96% accuracy as compared to others.

1. Introduction

Movie Recommendation systems are now widely employed in many different industries, including movies, music, books, videos, apparel, restaurants, food, locations, and many more. They have also grown increasingly stylish. Finding stuff that would be exciting to a person is the main goal of a recommendation system. Besides, it comes with some attributes for creating individual lists of fascinating and useful items for every user or individual.

There are an array of applications for recommendation systems. These are now utilized in most of the internet platforms and have become popular these past few years. These platforms contain a wide range of content, including movies, music, books, and videos; friends and stories on social media; products on e-commerce websites; people on dating and professional websites; and Google search results. With the evolution of big data and machine learning, recommendation systems have increasingly become more complex over time. In the future, recommendation systems could be able to handle large volumes of data and use advanced algorithms such as deep learning and reinforcement learning to improve the quality of their recommendations and make them more relevant to the user of the data [1]. The fact that implicit data has been used in these systems has prompted an increase in the number of research efforts on cognitive-based recommendation systems which rely on users' personalities or behaviours to make recommendations. One advantage of this is that it enables the recommendation systems to adjust to user changes rapidly [2]. Stepwise is the most common method in the film or any recommender system (RS), in which users rate a few objects and the system predicts the user's rating for an object which has not yet been rated. They can help make decisions by creating lists of items, resources and data that are tailored to the preferences and previous experiences of consumers. Moreover, RSs use several technologies to filter out satisfactory results, reducing search time and providing the information the consumers are looking for quickly. In the past few years, the use of RSs has grown exponentially in many different fields such as social networking, entertainment, e-commerce, scientific research, news, healthcare, tourism, and education. Personalized recommendation systems use AI, data mining, and other Internet-related

technologies to extract information from users' behaviour, and offer products that they will likely be very interested in. YouTube, Netflix, for instance, use RSs algorithms to offer customized video and content recommendations. Most of these suggestions are based on content analytics, user behaviour trends, and social connections. To recommend items based on the user's interests and the history of what they heard from friends, the systems utilize similarities between friends' relationships and listening records. In this way, users will be more likely to come back to the platform since they always get new and interesting content tailored to their interests. In a similar vein, RS algorithms are used by social networking and content-sharing businesses such as Facebook, Instagram, and LinkedIn to increase user engagement. Our study model will also look at how to utilise the vast number of movies available online to recommend movies that would be enjoyed by the viewer, to improve efficiency and user experience as a whole. Users can find superior material to watch more easily as a result. In line with the continuous developments in streaming services and digital entrepreneurship, the study offers a personalized movie recommendation model powered by AI that incorporates sentiment and content analysis. It addresses the problems of 'information overload' by integrating AI technology, enhancing the digital entertainment experience for both viewers and service providers. These days, recommendation algorithms are widely used in a variety of fields, including education and entertainment. Once upon a time, consumers were forced to pick and choose what books to read, what music to listen to, what movies to watch, and much more. There are more than 15 million movies in commercial movie libraries, and too many to watch individually. A lot of people are overwhelmed by the amount of movies to browse through. Consequently, both movie service providers' and consumers' enthusiasm depend on an effective recommendation system. As e-commerce and the Internet have evolved, the recommendation machine has become popular. This paper's electronic commerce recommendation system namely the collaborative filtering technique is investigated in the same manner as used in personalized movie recommendation systems [7]. To use both explicit and implicit user data for item suggestion, it is

necessary to understand the user's preferences by applying a set of different data mining techniques. Consequently, ongoing research has been carried out on how to get better insights for the recommendations based on already selected items, how to get input from the users for the recommendations, user correlation analysis, etc. [8-13]. A resilient distributed dataset (RDD) is used by its functions for programming. However, parallel program execution may be a plus. Spark provides a number of libraries and tools for handling and scaling data, including Scala, Python, and R [14]. Similar tastes are recommended with items. Finally, the hybrid strategy improves the quality of its recommendations by incorporating collaborative techniques, content-based filtering and other techniques [15].

2. Literature Review

Kumar et al. [16] developed MOVREC, a movie recommendation system based on collaborative filtering techniques. This type of filtering utilizes

Table 1: Movie Recommendation Systems literature Review

data from an entire user base in order to produce recommendations. De Campos et al. [17] also performed an evaluation of both the conventional recommendation methods. Since both of these methods have some delays, he suggested another system that is a blend of Bayesian network and collaborative method. Clustering was suggested by Kuźlewska [18] as a way to address the suggestions. The centroid-based solution and memory-based clustering techniques were examined. As a result, precise recommendations were produced. In their paper, Sharma and Maan [19] investigated a variety of recommendation techniques, including content-based, collaborative, and hybrid recommendations. It also outlines the benefits and limitations of these methods. An inductive learning algorithm was set forth by Li and Yamada [20]. In this particular instance, a tree which displays user recommendations has been built. Table 1 explores some of the main contributions made to the recommendation system.

Cited	Observations	Limitations
[21]	Reviewed content-based, collaborative filtering, and hybrid recommendation techniques. Found hybrid methods generally outperform standalone approaches in recommendation accuracy.	Review-focused; lacks experimental validation and evaluation on modern deep-learning architectures.
[22]	Identified collaborative filtering and hybrid systems as dominant approaches in streaming platforms. Highlighted growing need for personalization in OTT services.	Limited discussion of emerging AI techniques such as transformers and LLM-based recommenders.
[23]	Demonstrated that recommendation diversity and user control significantly affect user satisfaction and exploration behavior.	Focused on user experience rather than improving recommendation algorithm accuracy.
[24]	Proposed persona-based user modeling combined with NMF and DNNs. Improved personalization and addressed cold-start issues in real-time recommendations.	Increased computational complexity and dependency on accurate persona construction.

[25]	Combined deep reinforcement learning with collaborative filtering to improve adaptive recommendations and user engagement.	Training is computationally expensive and requires large-scale interaction data.
[26]	Enhanced recommendation quality by incorporating sentiment analysis from user reviews and ratings.	Performance depends heavily on review quality and language-processing accuracy.
[27]	Neural Collaborative Filtering (NCF) effectively captured complex user-item interactions and improved recommendation accuracy over traditional collaborative filtering.	Still suffers from data sparsity and requires substantial training resources.
[28]	Found that higher offline accuracy does not necessarily lead to higher user satisfaction. Diversity, transparency, and trust significantly influence user experience.	Study focused on user perception; results may vary across datasets and domains.
[29]	Utilized deep learning, graph techniques, and text mining to model dynamic user preferences, resulting in improved personalization.	High computational requirements and limited interpretability of learned representations.
[30]	Introduced a GNN-based model incorporating seasonality and graph topology, improving recommendation effectiveness.	Graph neural networks require large datasets and have high computational overhead.
[31]	Compared multiple recommendation algorithms and showed deep-learning-based methods generally achieve better predictive performance than classical approaches.	Comparative study only; did not propose a novel recommendation framework.
[32]	Combined LSTM-based sequential modeling with LLM semantic reasoning, achieving improved context-aware recommendations.	LLM integration increases computational cost and deployment complexity.
[33]	Applied Neural Collaborative Filtering with embedding layers and negative sampling, achieving strong recommendation accuracy (Hit Rate \approx 0.89).	Evaluated primarily on MovieLens dataset, limiting real-world generalizability.

3. Proposed Methodology

This study used a hybrid methodology that combines cosine similarity and text-to-number conversion. The cosine similarity technique is a recommendation method based on the comparison and prediction of user preferences and behaviors, which recommends and predicts highly rated movies for users. Alternating Least Square (ALS) algorithm was applied to enhance the methodology. The latent attributes that capture the underlying relationships and patterns in the dataset can be extracted more easily thanks to this factorization algorithm. Alternating Least Squares is a method that can be used to identify the latent factors that affect user preferences and movie ratings, based on the matrix factorization method. This helps the system to generate personalized recommendations for users and to accurately predict their needs.

3.1 The Suggested Model

The proposed model is hybrid movie recommendation system which utilizes Cosine similarity and Text to Number conversion. Hybrid recommendation models use several strategies to achieve better accuracy in the movie

recommendation. In the present study, a model-based procedure was employed and consisted of two major algorithms: Alternating Least Squares algorithm (ALS) and matrix factorisation. A large, diverse library of films will be part of the proposed framework, which will be a wide-ranging, comprehensive set of data. The data set will include language, release years, genres and other relevant attributes to reflect from the data. It was carefully selected and arranged to ensure that a variety of films from diverse sources were included. The data set will be thoroughly compiled and come with movies of all genres including documentaries, science fiction, romance, action, adventure, comedy, thriller, etc. It will cover both well-known and obscure films, enabling thorough and thorough coverage of the film industry. Furthermore, the database will feature a diverse range of languages for the films, ensuring that recommendations can be made to users who have different language preferences and reaching a global audience. Thanks to this multilingual feature, the framework will be able to suggest movies to users of different cultures, depending on their language.

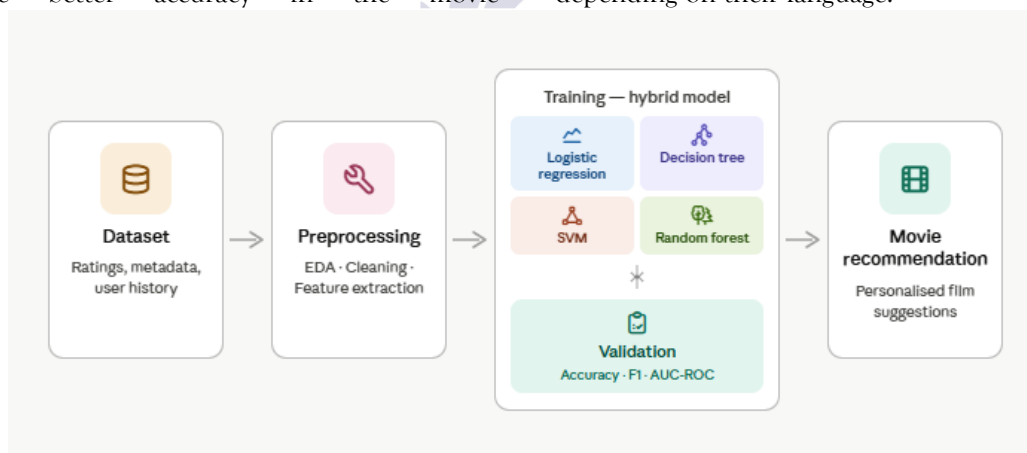


Figure 1. Workflow of the Proposed Movie Recommender System

The movie recommendation system can be explained through a typical pipeline, as shown in Figure 1: it begins with a raw Dataset, moves on to Preprocessing (data explorative analysis, data cleaning and feature extraction), and ends with usable features. These features are then input into a modeling phase to train and validate a Hybrid Model (training and validation loop), which proposes a mix of approaches to enhance the accuracy. The process results in an output "Movie Recommendation".

3.2 Dataset

For the experiments, the suggested model will be applied to the popular and widely-used "TMDB 5000 Movie Dataset" from the Kaggle dataset. One of the popular datasets used for analysis and assessment of recommendation systems is the TMDB 5000 Movie Dataset. It is a useful resource for researching user preferences and making recommendations as it contains user ratings and community user data. The TMDB 5000 Movie Dataset provides a comprehensive list of user ratings, giving valuable insights into people's perceptions and ratings of movies. The variety of

genres has a large number of users that have evaluated the films they have watched, which is a great advantage for a thorough analysis of user preferences and behavior. A thorough collection of 4,800 films makes up the Movie dataset used in this suggested model. The users of the dataset were randomly chosen, so that the users covered the broad spectrum of movie tastes. The recommendation system will be tested with the 4,800 movies to verify the effectiveness and

4. Experiment and Results

Table 1: *Logistic Regression*

Class	Precision	Recall	F1_score	Support
Negative	0.96	0.93	0.94	2475
Positive	0.94	0.96	0.95	2525
Accuracy	0.9476			

The table 1 shows the metrics of a Logistic Regression model used to make binary classification, including the accuracy of the model to classify between the negative and positive classes. The precision of positive predictions is 0.94 for the positive class and 0.86 for the negative class. Recall means the model's ability to pick out actual positive instances; the positive class in this

Table 2: *Decision Tree*

Class	Precision	Recall	F1_score	Support
Negative	0.80	0.81	0.81	2475
Positive	0.81	0.80	0.81	2525
Accuracy	0.8076			

The performance parameters of a Decision Tree classifier for predicting two classes Negative and Positive is presented in the table 2. The Precision value is 0.80, Recall value is 0.81, and F1_score value is 0.81 with a support of 2475 instances for the Negative class. In the Positive class, precision

Table 3: *Random Forest*

Class	Precision	Recall	F1_score	Support
Negative	0.92	0.94	0.93	2475
Positive	0.94	0.92	0.93	2525
Accuracy	0.9310			

The table 3 shows the performance of a Random Forest model trained to classify two classes: Negative and Positive. It contains important measures like Precision, Recall, F1_score, and Support (number of instances in each class). In Negative class the model gives Precision value, 0.92, Recall value, 0.94, F1_score value, 0.93 and

reliability of the model. You should be able to think of 5 films that you think are related to the movie you wrote. This low threshold ensures that enough recommendations for movies are provided so that users can make an accurate assessment of their preferences. Each of the datasets participants has a unique user ID, which enables individual user analysis and customized generation of recommendations.

case had a higher recall of 0.86 than the negative class had of 0.83. The F1 score, which is a balance between precision and recall, is 0.94 for negative class and 0.85 for positive class. The model has a total accuracy of 0.9476, meaning across the whole dataset (2,475 instances for the negative class and 2,525 instances of the positive class), the model has a fair degree of accuracy.

is 0.81, recall is 0.70 and F1_score is 0.81, with a support of 2525 instances. The overall accuracy of the model is 0.8076, which means that out of all the results that were tested, 70.76% were correct. This indicates that the Decision Tree has a relatively even performance between classes.

number of instances, 2475. On the other hand, for the Positive class, it had a higher precision of 0.94 and a lower recall of 0.92, and also an F1_score of 0.93 with 2,525 instances. For the overall model (Accuracy), the result is good with 0.8310 for both classes. Bookmark message Copy message Export.

Table 4: *SVM*

Class	Precision	Recall	F1_score	Support
Negative	0.98	0.95	0.96	2475
Positive	0.95	0.99	0.97	2525
Accuracy	0.9668			

The table 4 shows the results of a binary Support Vector Machine (SVM) model that classifies negative and positive classes. Precision is the accuracy of positive predictions with 0.88 for negative predictions and 0.85 for positive predictions. Recall is the model's ability to find all the relevant instances; it has a score of 0.85 for negative and 0.89 for positive classes. The F1 Score; a compromise between precision and recall,

is 0.86 for negative and 0.87 for positive, indicating a good balance between the two. The model is able to predict the correct value almost 86.68% of the time, with an overall accuracy of 0.8668. The support values indicate the number of instances that are present in the dataset for each class with 2525 positives and 2475 negatives, giving information about the class distribution of the dataset.

Table 5: *KNN*

Class	Precision	Recall	F1_score	Support
Negative	0.64	0.49	0.55	2475
Positive	0.63	0.68	0.70	2525
Accuracy	0.6354			

In this table (5), the evaluation metrics for a K-Nearest Neighbors (KNN) classification model are presented, which is based on two classes: Negative and Positive. Precision refers to the accuracy of the Positive predictions with 64% for Negative cases and 63% for Positive cases. Recall is a measure of the model's performance on the real positive cases

(68%) and real negative cases (49%). The scores of the F1 for Negative and Positive are 0.55 and 0.70 respectively which shows a moderate performance. The overall accuracy of the model is around 63.54%, indicating its performance for both of the classes. The number of real instances for each class is: Negative = 2475; Positive = 2525.

Table 6: *XGBoost*

Class	Precision	Recall	F1_score	Support
Negative	0.96	0.90	0.93	2475
Positive	0.91	0.97	0.94	2525
Accuracy	0.9350			

The following table 6 is the performance metrics of a predictive model with the technique XGBoost, measured in two classes, Negative class and Positive class. The model had a precision of 0.96, a recall of 0.90, and an F1 score of 0.93 and a support of 2475 instances for the Negative class. The Positive class, on the other hand, had a precision of 0.91, a recall of 0.97 and an F1 score

of 0.94, while the number of instances supported was 2525. The accuracy of the overall model is 0.9350, which suggests that the model is capable of making good predictions for both classes. These metrics demonstrate the ability of the model to accurately predict the classes with a balance between precision and recall of the two classes.

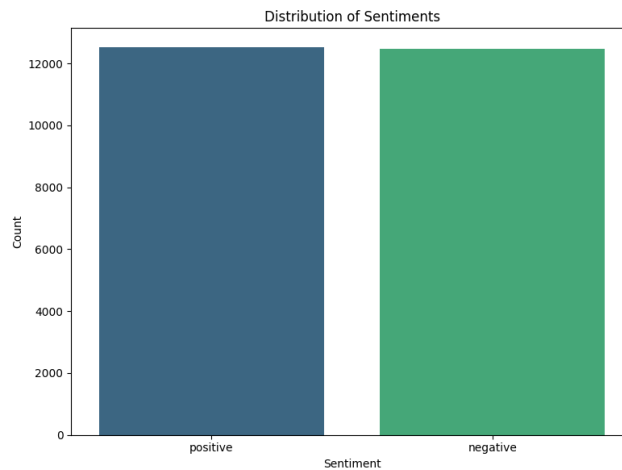


Figure 2. The bar chart for sentiments

The bar chart below shows the distribution of the sentiments, which are classified as 'positive' and 'negative'. The number of both positive and negative sentiments around each of them are in about equal numbers, with more positive than negative, and there are more than 12,000 examples of each. The two bars on the chart

suggest a balanced distribution of positive and negative responses, representing a relatively equal sample of responses. This indicates that the emotions contained in the data are mostly neutral with a significant amount of positive and negative emotions.

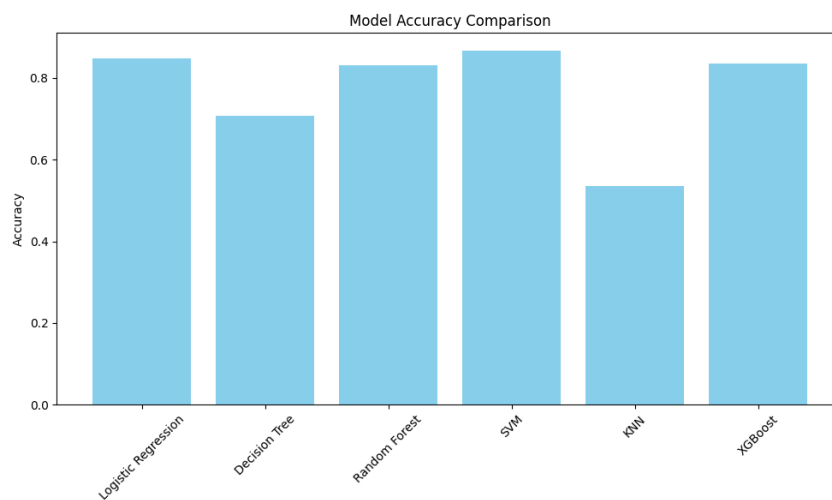


Figure 3. Bar Chart Accuracy with Machine Learning Models

The bar chart shows the performance of different machine learning models in terms of their accuracy, such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and XGBoost. The values of accuracy are ranging from under 0.5 to more than 0.8, Logistic Regression is

having the highest accuracy followed by SVM and Random Forest. However, Decision Tree and KNN have less performance. The graphical representation also clearly brings out the differences in the accuracy of the model, which helps to choose the most accurate model for predictive tasks.

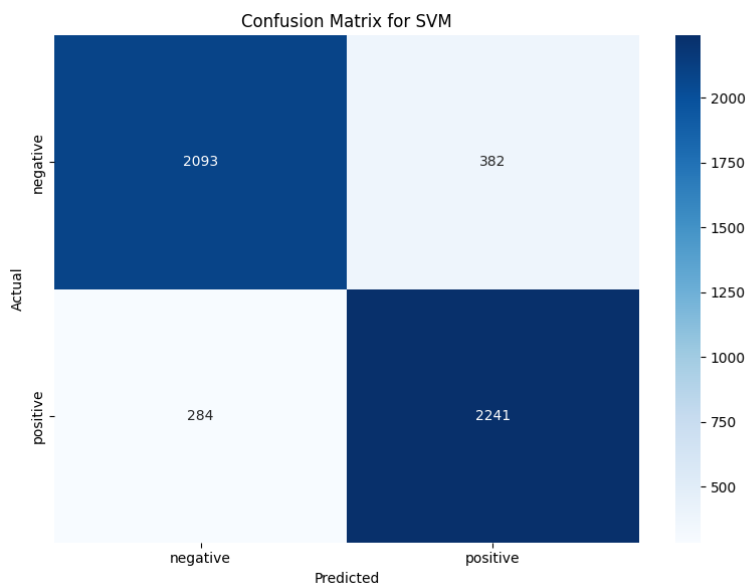


Figure 4. Confusion Matrix

This confusion matrix represents the accuracy of a Support Vector Machine (SVM) classification model for a binary classification problem. There are four different quadrants: True Negatives (2093): Predicted negatives that were correctly identified. False Positives (382): False alarms when the actual object was negative. (False Negatives = 284): False predictions of negative instances which were actually positive. True Positives (2241): The true positive cases that were correctly predicted. The high numbers in the true negative and true positive rows show that the model works well in determining the negative case and positive case, respectively, while the low numbers in the false positive and false negative rows suggest that the model can be further tuned or analyzed for those cases.

Performance Metrics:

$$\text{PRECISION} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{RECALL} = \frac{TP}{TP+FN} \quad (2)$$

$$\text{F1_Score} = \frac{2 \times TP}{TP+FN+FP} \quad (3)$$

5. Conclusion:

One of the most crucial parts affecting the modern digital experience is the recommendation system. The recommendation mechanism has proven to be extremely useful for online movie streaming companies, as it helps them understand user preferences, viewing history, and behavioral

patterns. With the overwhelming amount of content in the world, recommender systems have become crucial in assisting consumers to find content that is relevant and tailored to their interests but which they cannot discover by using searching manually. Recommender system is not only useful to the service providers but also helpful for customers, and it is also practical. They are able to retrieve information that corresponds to a user's preferences, lessening the cognitive load of the user when making decisions, saving time and enhancing the fun. These tools can assist platforms to boost the number of users, their retention and engagement, and, ultimately, their income. These solutions have the potential to generate huge user engagement, retention, and ultimately revenue on platforms. On this point, recommender systems have helped society in many aspects, providing an extensive choice to customers and helping the digital entertainment industry grow. There are many different ways and algorithms to construct effective recommender systems, such as collaborative, content-based and more sophisticated machine-learning approaches. In this study, it is found that Support Vector Machine (SVM) model is the best model as compared with the conventional models with the highest accuracy of 96%. The results indicate that SVM is an effective and reliable method for movie recommendation tasks and it has a good ability to classify and predict user choices. Last but not least, the importance of the intelligent recommender systems will remain to increase as the competition between streaming platforms continues from the

consumers' point of view. The results of this study further support the benefits of using SVM and other machine learning techniques to create a precise and effective recommendation system, which may be used in subsequent studies to improve prediction accuracy and user happiness. I am a connoisseur of the movies.

References

- [1] L. Terveen and W. Hill, "Beyond recommender systems: Helping people help each other," *HCI in the New Millennium*, vol. 1, no. 2001, pp. 487–1509, 2001.
- [2] Beheshti, A.; Yakhchi, S.; Mousaeirad, S.; Ghafari, S.M.; Goluguri, S.R.; Edrisi, M.A. Towards Cognitive Recommender Systems. *Algorithms* 2020, 13, 176.
- [3] Khadse, V.; Basha, A.P.S.; Iyengar, N.; Caytiles, R. Recommendation Engine for Predicting Best Rated Movies. *Int. J. Adv. Sci. Techn.* 2018, 110, 65–76.
- [4] Fayyaz, Z.; Ebrahimian, M.; Nawara, D.; Ibrahim, A.; Kashef, R. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *Appl. Sci.* 2020, 10, 7748.
- [5] M. Balabanovic, Y. Shoham, Fab: content-based, collaborative recommendation, *Communications of the ACM* 40 (3) (1997) 66–72.
- [6] Kumar, M., Yadav, D. K., Singh, A., & Gupta, V. K. (2015). A movie recommender system: Movrec. *International journal of computer applications*, 124(3), 7-11.
- [7] Peng, Xiao, Shao Liangshan, and Li Xiuran. "Improved Collaborative Filtering Algorithm in the Research and Application of Personalized Movie.
- [8] Abbasi-Moud, Z.; Vahdat-Nejad, H.; Sadri, J. Tourism Recommendation System Based on Semantic Clustering and Sentiment Analysis. *Expert Syst. Appl.* 2021, 167, 114324.
- [9]. Liu, F.; Lee, H.J. Use of Social Network Information to Enhance Collaborative Filtering Performance. *Expert Syst. Appl.* 2010, 37, 4772–4778.
- [10]. Yousif, M., Abbas, A., Hasan, Z., Ali, D., & Sarfraz, M. (2021). Smart Village Health System IoT to Envisage Chronical Disease Using Artificial Neural Network. *Internet Things Cloud Comput.*, 9(4), 27.
- [11]. Amato, F.; Moscato, V.; Picariello, A.; Piccialli, F. SOS: A Multimedia Recommender System for Online Social Networks. *Future Gener. Comput. Syst.* 2019, 93, 914–923.
- [12]. Capdevila, J.; Arias, M.; Arratia, A. GeoSRS: A Hybrid Social Recommender System for Geolocated Data. *Inf. Syst.* 2016, 57, 111–128.
- [13] Tarus, J.K.; Niu, Z.; Yousif, A. A Hybrid Knowledge-Based Recommender System for e-Learning Based on Ontology and Sequential Pattern Mining. *Future Gener. Comput. Syst.* 2017, 72, 37–48.
- [14]. Bazai, S.U.; Jang-Jaccard, J. In-Memory Data Anonymization Using Scalable and High Performance RDD Design. *Electronics* 2020, 9, 1732.
- [15]. Shambour, Q.Y.; Hussein, A.H.; Kharma, Q.M.; Abualhaj, M.M. Effective Hybrid Content-Based Collaborative Filtering Approach for Requirements Engineering. *Comput. Syst. Sci. Eng.* 2022, 40, 113–125.
- [16]. Kumar, M., Yadav, D. K., Singh, A., & Gupta, V. K. (2015). A movie recommender system: Movrec. *International journal of computer applications*, 124(3), 7-11.
- [17]. De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Combining content-based and collaborative recommendations: a hybrid approach based on Bayesian networks. *International journal of approximate reasoning*, 51(7), 785-799.
- [18]. Kuzelewska, U. (2014). Clustering algorithms in hybrid recommender system on movielens data.

- Studies in logic, grammar and rhetoric, 37(1), 125-139.
- [19]. Sharma, M., & Mann, S. (2013). A survey of recommender systems: approaches and limitations. *International journal of innovations in engineering and technology*, 2(2), 8-14.
- [20]. Li, P., & Yamada, S. (2004, December). A movie recommender system based on inductive learning. *IEEE conference on cybernetics and intelligent systems* (pp. 318-323). Singapore, Singapore: IEEE.
- [21]. Ariyanto, Y., & Widiyaningtyas, T. (2024). A systematic review of movie recommender systems. *ITEGAM-JETIA*, 10(47), 34-41.
- [22]. Wibisono, C. H., Purwanti, E., & Effendy, F. (2023, January). A systematic literature review of movie recommender systems for movie streaming service. In *AIP Conference Proceedings* (Vol. 2554, No. 1, p. 040005). AIP Publishing LLC.
- [23]. Sun, R., Akella, A., Kong, R., Zhou, M., & Konstan, J. A. (2024). Interactive content diversity and user exploration in online movie recommenders: A field experiment. *International Journal of Human-Computer Interaction*, 40(22), 7233-7247.
- [24]. Lee, H. C., Kim, Y. S., & Kim, S. W. (2024). Real-time movie recommendation: Integrating persona-based user modeling with NMF and deep neural networks. *Applied Sciences*, 14(3), 1014.
- [25]. 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.
- [26]. Sarhan, A. M., Ayman, H., Wagdi, M., Ali, B., Adel, A., & Osama, R. (2024). Integrating machine learning and sentiment analysis in movie recommendation systems. *Journal of Electrical Systems and Information Technology*, 11(1), 53.
- [27]. LI, Z. (2024). NCF-BASED MOVIE RECOMMENDATION SYSTEM. *APPLIED AND COMPUTATIONAL ENGINEERING Учредители: EWA Publishing*, 104(1), 72-77.
- [28]. Sun, R., Wu, X., Akella, A., Kong, R., Knijnenburg, B., & Konstan, J. A. (2024). What Are We Optimizing For? A Human-centric Evaluation of Deep Learning-based Movie Recommenders. *arXiv preprint arXiv:2401.11632*.
- [29]. Gao, Y., Zheng, H., & Cui, H. (2025). User preference modeling for movie recommendations based on deep learning. *Scientific Reports*, 15(1), 16592.
- [30]. Özden, C., & Özcan, A. (2025). Seasonality-Aware, Positional, and Topological-Guided GNN (SPT-GNN) for Movie Recommendation. *Computational Intelligence*, 41(5), e70148.
- [31]. Walek, B., & Sládek, O. (2025). Comparison of Selected Algorithms in Movie Recommender System. *Applied Sciences*, 15(17), 9518.
- [32]. Li, Y., & Grasman, R. (2025). DUALRec: A Hybrid Sequential and Language Model Framework for Context-Aware Movie Recommendation. *arXiv preprint arXiv:2507.13957*.
- [33]. Liu, C. (2025). Movie recommendation based on deep neural networks. *Procedia Computer Science*, 266, 32-40.