SN Computer Science (2025) 6:257
https://doi.org/10.1007/s42979-025-03742-7

ORIGINAL RESEARCH

From User Preferences to Accurate Predictions: Enhancing Movie Recommendation Systems with Neural Collaborative Filtering and Sentiment Analysis

Qusay Bsoul[1] · Firas Zawaideh[2] · Basma S. Alqadi[3] · Latifa Abdullah Almusfar[4] · Osamah Ibrahim Khalaf[5] · Ahmed Saleh Alattas[6] · Muath Alali[7] · Diaa Salama AbdElminaam[8,9]

Received: 1 September 2024 / Accepted: 8 December 2024

Abstract

This paper introduces an advanced movie recommendation system that combines neural collaborative filtering (NCF) with sentiment analysis to improve personalization, accuracy, and robustness in recommending movies. By integrating user preferences with the sentiment polarity of reviews, the system refines recommendations, effectively addressing challenges such as the cold start problem, data sparsity, and limited semantic understanding. Unlike traditional collaborative and content-based filtering methods, which often struggle with diversity and accuracy, our approach utilizes deep learning within NCF to reveal hidden patterns in user behavior, enhancing recommendations' relevance and precision. The proposed system is evaluated using standard performance metrics, including RMSE, MSE, and MAE, demonstrating its superior performance over conventional filtering techniques. Its enhanced scalability and adaptability position it as a promising tool for personalized content delivery in digital entertainment, with considerable potential for large-scale, dynamic recommendation environments. This research contributes to the existing knowledge on recommendation systems and offers new insights into improving content personalization and user satisfaction. The novelty and scientific value of this work lie in applying deep learning to tackle the challenges of accurate content recommendation in the rapidly evolving digital media landscape.

Keywords Movie recommendation system · Collaborative filtering · Content-based filtering · Cosine and person similarities · Evaluation metrics · RMSE · Surprise library · NCF · Deep learning · MAE

Introduction

In the era of digital entertainment, the proliferation of streaming services has created an overwhelming array of content choices. Navigating this vast landscape challenges users seeking personalized and enjoyable experiences. This paper explores the significance of movie recommendation systems powered by machine learning algorithms, examining their pivotal role in reshaping our cinematic journeys and their multifaceted impact on users, content providers, and technological advancements.

The main problem addressed in this paper is the challenge of effectively managing the overwhelming abundance of movie choices available to users, compounded by the issue of information overload. This is further complicated by misleading or biased reviews that can skew user perceptions and decision-making processes. Traditional movie recommendation systems often struggle to provide personalized and accurate suggestions that cater to individual user preferences and interpret the true sentiment behind user reviews. This paper seeks to tackle these issues by proposing a sophisticated movie recommendation system that employs neural collaborative filtering (NCF) integrated with sentiment analysis, aiming to enhance movie suggestions' accuracy, personalization, and reliability.

The ubiquity of streaming platforms necessitates efficient ways to navigate the abundance of available content. Movie recommendation systems, driven by advanced machine learning algorithms, address this challenge by offering a personalized approach to content discovery. These systems analyze user preferences, viewing histories, and behavioral patterns to curate tailored suggestions, ensuring that users are presented with relevant options and engaged in a cinematic journey that aligns with their unique tastes.

Extended author information available on the last page of the article

Published online: 08 March 2025

Efficiency is a cornerstone of recommendation systems' impact. With an ever-expanding content library, manually sifting through options becomes daunting. Machine learning algorithms streamline this process, saving users time and effort by presenting curated suggestions aligned with their preferences. This efficiency translates into a more enjoyable content exploration experience, as users spend less time searching and more time immersed in movies that resonate with their tastes.

Recommendation systems generate valuable user behavior and preferences data, offering insights into evolving trends. This data-driven approach refines recommendation algorithms and empowers content providers and filmmakers to make informed decisions about future content creation. The symbiotic relationship between users and the platform is further strengthened as the system adapts and evolves based on user interactions, creating a dynamic and responsive entertainment ecosystem.

In essence, this research explores the transformative impact of movie recommendation systems on our entertainment experiences. From personalized content discovery and time-saving benefits to diverse content exploration and data-driven insights, these systems exemplify the power of machine learning in enhancing user engagement and shaping the future of the cinematic landscape. As we delve into the intricate workings of these systems, we uncover their immediate benefits for users and businesses and their pivotal role in steering the trajectory of the ever-evolving entertainment industry.

In the realm of machine learning-powered movie recommendation systems, the integration of deep learning techniques has further elevated their efficacy. As users grapple with the overwhelming array of choices presented by the ubiquity of digital content and streaming services, deep learning algorithms play a pivotal role in enhancing the capabilities of these systems. Unlike traditional machine learning methods, deep learning, with its neural network architectures, can automatically extract intricate patterns and representations from vast amounts of data, including user behavior, preferences, and historical interactions. This enables recommendation systems to provide even more nuanced and accurate personalized suggestions, thereby simplifying decision-making and profoundly enriching the overall viewing experience for users. The utilization of deep learning in these recommendation systems extends their impact on content discovery by uncovering hidden correlations and introducing users to a broader array of movies tailored to their individual tastes. This trans-formative influence, driven by deep learning, goes beyond mere suggestions, fostering heightened user engagement and satisfaction. Incorporating deep learning into movie recommendation systems powered by machine learning has become instrumental in mitigating information overload, elevating content discovery, and shaping a remarkably personalized and enjoyable digital content consumption experience.

This paper introduces an advanced movie recommendation system that integrates neural collaborative filtering (NCF) with sentiment analysis to enhance personalization, accuracy, and robustness in movie suggestions. Traditional recommendation techniques, such as collaborative and content-based filtering, often face significant limitations. Collaborative filtering methods struggle with issues like the cold start problem, where new users or items have little historical data, and data sparsity, where limited user interactions lead to poor recommendations. Conversely, content-based filtering is often constrained by limited semantic understanding, as it relies heavily on item attributes and fails to capture complex user preferences. These methods also tend to lack diversity in recommendations, frequently suggesting similar items to users without accounting for changing tastes.

Our proposed system overcomes these challenges by combining the strengths of NCF and sentiment analysis. By incorporating sentiment polarity from user reviews, the system captures the user's preferences and the underlying emotions and opinions associated with each movie, enabling more accurate and contextually relevant recommendations. This approach addresses the cold start problem by leveraging sentiment from new users' interactions. In contrast, sentiment analysis helps mitigate data sparsity by extracting valuable information from reviews, even when interaction data is limited. Additionally, deep learning within NCF uncovers latent patterns in user behavior, allowing for a better understanding of user preferences and improving recommendation diversity and accuracy.

We validate the proposed system using standard performance metrics—RMSE, MSE, and MAE—and demonstrate its superiority over traditional methods regarding precision and scalability. The system's ability to adapt to dynamic user preferences and handle large-scale datasets makes it a promising tool for personalized content delivery in the digital entertainment. This research advances the recommendation systems field and provides a novel approach to addressing the challenges of accurate content suggestion in the increasingly complex digital media landscape. The results contribute insights that can significantly enhance content personalization and user satisfaction in real-world applications.

The main objectives of this paper are to develop a robust movie recommendation system utilizing neural collaborative filtering (NCF) enhanced by deep learning techniques to improve the accuracy and personalization of movie suggestions. The integration of sentiment analysis aims to refine the understanding of user reviews, increasing the reliability of the recommendations. Performance evaluation is conducted using metrics such as root mean squared error (RMSE), mean squared error (MSE), and mean absolute error (MAE) to ensure precision and reliability. Additionally, the system is compared with traditional collaborative filtering methods to highlight the improvements and benefits of the proposed approach. Ultimately, the goal is to enhance user satisfaction and engagement by delivering precise and personalized movie recommendations that align with individual preferences and viewing histories. In our movie recommendation system project, K-nearest neighbors (KNN) [1, 2] serves as a pivotal model for collaborative filtering. By identifying similarities between users or items based on their historical preferences, KNN [1] enables us to recommend movies that align with a user's taste. This approach fosters a collaborative environment, leveraging similar users' preferences to enhance our recommendations' accuracy and relevance. Incorporating content-based filtering, our system analyzes the intrinsic characteristics of movies to make personalized recommendations. Examining features such as genre, actors, cast, keywords, and overview ensures that recommended movies resonate with each user's unique preferences. This approach enhances content discovery by providing suggestions based on the inherent attributes of the movies themselves. content-based and collaborative filtering recommendation is implemented using item-based approaches. This model taps into collective user behaviors to generate recommendations. By considering the preferences of similar items, collaborative filtering enhances the diversity and accuracy of our movie suggestions, contributing to a more engaging user experience singular value decomposition (SVD) [3] is instrumental in our movie recommendation system, as it facilitates the factorization of user-item interaction matrices. By capturing latent features, SVD [3] enhances the precision of personalized recommendations. This model allows our system to understand intricate patterns in user behavior and preferences, contributing to the effectiveness of content suggestions.

Neural collaborative filtering (NCF) [4–6] represents a cutting-edge application of deep learning in the realm of movie recommendation systems. NCF [4] is designed to enhance the accuracy and precision of personalized movie suggestions to users. Unlike traditional collaborative filtering methods, NCF [4, 7] combines the strengths of neural networks to capture intricate patterns and representations in user-item interactions. By leveraging deep learning architectures, NCF [4]can discern complex relationships within vast datasets, including user preferences and historical behavior, contributing to a more nuanced understanding of individual tastes. This advanced model excels at uncovering latent features that may not be apparent through conventional approaches, resulting in highly accurate and tailored movie recommendations. NCF's [4] ability to operate seamlessly with both collaborative [8] and content-based filtering [9] techniques makes it a versatile and powerful tool in the quest to provide users with a compelling and personalized movie-watching experience. Integrating NCF [4] into recommendation systems signifies a significant stride in applying deep learning for more effective and precise movie suggestions, ultimately enhancing user satisfaction and engagement.

The main contributions of this paper are summarized as follows:

- We propose a novel movie recommendation system utilizing neural collaborative filtering (NCF), which leverages deep learning techniques to enhance personalization and accuracy in movie suggestions.
- Integrates neural collaborative filtering (NCF) with sentiment analysis, allowing the recommendation engine to utilize sentiment data from user reviews and address limitations in traditional collaborative filtering, especially for cold start and sparse data scenarios.
- Enhances personalization and recommendation accuracy by incorporating sentiment polarity, achieving a deeper understanding of user preferences, and mitigating repetitive suggestions common in content-based filtering.
- NCF's deep learning capabilities to handle large, complex datasets efficiently, uncovering latent patterns in sparse data, which supports scalability and robustness for diverse user and item types.
- Merging of the "tmdb5000" [10] and "movie_lens" [11, 12] datasets introduces a substantial added value to our analytical endeavors by combining these datasets.
- Strong preprocessing phase to ensure that our data is clean from any noise.
- Turning hyperparameters for the different algorithms using grid search helps to reach the best results.
- The implementation of different algorithms as demographic recommender using weighted average rating, content-based using NLP (TF-IDF [13] based on overview or cast, director, keywords, and genres), collaborative filtering (KNN [1], SVD [3], NMF).
- All of these algorithms are evaluated using RMSE [14], MSE, and using different similarities techniques such as cosine similarity and person similarity.

The rest of the paper is organized as follows: related work, methodologies for machine and deep learning used, hyperparameters optimization, results and discussions, conclusion, and future work.

Related Work

In recent years, the domain of movie recommendation systems has witnessed substantial growth, fueled by advancements in machine learning algorithms and the increasing availability of large-scale datasets. Researchers and practitioners have dedicated considerable efforts to enhance the movie-watching experience by developing personalized recommendation systems. This section provides an overview of the existing literature and research endeavors in movie recommendation systems, specifically focusing on those leveraging machine learning techniques. In [15], the author proposed the problem of providing personalized movie recommendations to enhance the movie-watching experience for users. The study aims to develop a functional and accurate movie recommendation system using machine learning algorithms. The dataset used in the research comprises movie ratings and user information, which are utilized to create a prediction model for recommending movies based on user preferences. The study's results demonstrate the recommendation system's effectiveness in providing relevant and personalized movie suggestions. The best results are achieved through the utilization of collaborative filtering [8] and content-based filtering [9] algorithms. However, a weakness of the paper lies in the challenge of producing satisfactory recommendations for new users or from a cold start, indicating the need for additional information about users and movies to improve the model's performance.

In [16], the author aims to compare different machine learning algorithms for movie recommendation systems. The problem addressed is the need for a systematic recommendation approach based on user history, likes, and reviews. The objective is to evaluate the quality of different algorithms using the MAE and RMSE [14] metrics. The dataset used is the MovieLens dataset, which contains movie ratings from users. The results show that the closed SVD [3]and KNNwM [1] algorithms predicted with minimal MAE, while the best predictors for RMSE [14] are BaselineOnly and SlopeOne. The weakness of this paper is that it only evaluates a limited number of algorithms and does not consider other factors, such as scalability and real-time performance.

In [17], the author proposed implementing a movie recommendation system using machine learning techniques. The problem addressed is the need for personalized movie recommendations to keep subscribers entertained and renew their subscriptions. The objective is to recommend movies to users based on their interests and ratings using collaborative filtering [8] and content-based filtering [9]. The dataset consists of 480,189 customers and 17,770 movies with ratings ranging from 1 to 5. The results show that the proposed system effectively minimizes the root mean square error (RMSE) [14] and recommends top-10 similar movies to users. The best results are achieved using the XGBoost algorithm with 13 features. The weakness of this paper is that it does not address the cold start problem, where new users or movies have no ratings, and it does not compare the proposed system with other state-of-the-art recommendation systems.

In [18], the author proposed the challenge of recommending movies to users by introducing a model that combines content-based [9] and collaborative [8] approaches. The objective is to reduce the human effort involved in choosing from a large set of movies and provide more explicit outcomes than systems based solely on content. The dataset used for this study includes movie ratings and genre information. The results indicate that the collaborative approach [8] yields more explicit outcomes compared to content-based systems [9], allowing users to explore more movie options. The best results are achieved by leveraging the connection between different users and recommending movies to others with similar tastes. However, the paper's weaknesses include the limited exploration of the dataset and the lack of comparison with other state-of-the-art recommendation systems.

In [19], the author proposed the information overload users face when selecting movies for viewing. The study aims to develop a movie recommendation system, MOVREC, based on collaborative filtering to assist users in efficiently finding movies aligned with their preferences. The system is designed in PHP using Dreamweaver 6.0 and Apache Server 2.0 and utilizes a customized database to generate recommendations. The dataset used in the study comprises information about movies, user ratings, and attributes such as actors, directors, genres, and years. The results indicate that the system effectively recommends films based on user preferences, with the K-means algorithm contributing to sorting recommended movies according to ratings. The best results are achieved by prioritizing movie ratings and incorporating user preferences from previous interactions. However, the paper acknowledges certain weaknesses, such as the challenge of evaluating the system's performance due to the subjective nature of movie recommendations and the need for a larger dataset to yield more meaningful results.

In [20], the author proposed the challenge of enhancing the precision of movie recommendation systems, particularly focusing on the collaborative filtering [8] approach. The primary objective is to introduce an improved k-clique [20]algorithm to overcome the limitations of traditional collaborative filtering methods [8]. The MovieLens dataset [11] is utilized for performance evaluation, and the experimental analysis involves comparing the collaborative filtering [8] methods using k nearest neighbor [1], maximal clique method, k-clique [20] method, and the improved k-clique [20]. The results demonstrate that the improved k-clique method significantly enhances the precision of the movie recommendation system compared to other methods. The best results are achieved by determining the optimal value of k in the k-clique [20] method, which maximizes accuracy in the recommender system. However, a weakness of the paper lies in the lengthy computation time required for the k-clique methods [20], prompting the need for future research to address this issue.

In [21], the author proposed the challenge of information overload and the difficulty in decision-making caused by misleading reviews in the movie industry. Its objective is to develop a movie recommendation system that suggests similar movies based on user preferences and conducts sentiment analysis on reviews to determine their positivity or negativity. The study utilizes three datasets: two for movie recommendation (`tmdb_5000_movies` [10] and `tmdb_5000_credits` [10]) and one for sentiment analysis (`reviews`). The results indicate that the Cosine Similarity algorithm effectively recommends related movies, and the support vector machine (SVM) [22] algorithm outperforms the Naïve Bayes (NB) algorithm in sentiment analysis, achieving an accuracy score of 98.63%. However, the paper acknowledges limitations, such as the system's failure to recommend movies not present in the dataset and the inability to analyze reviews in languages other than English. Additionally, the sentiment analysis may give incorrect sarcastic or ironic review classifications.

In [23], the author proposed the limitations of traditional matrix factorization techniques for collaborative filtering [8] in movie recommendation systems. The objective is to improve the accuracy of recommendations by incorporating temporal features such as user preferences over time. The authors used the Movielens 100 K [11] and 1 M datasets to verify the performance of their proposed system. The results showed that their approach outperformed the baselines, achieving a lower prediction error than comparative methods. The best results were achieved with a learning rate of 0.002, 50 epochs, and 100 factors. However, the weakness of this paper is that it only focuses on movie recommendation systems and does not explore the potential application of this approach to other types of recommendation systems.

In [24], the author proposed the limitations of traditional recommendation algorithms, such as data sparseness and lack of attention to diversity in recommendation results. The objective is to propose a movie recommendation algorithm based on sentiment classification [21] and topic extraction, aiming to enrich user interest and product feature models based on emotional tendencies to improve content-based recommendation algorithms. The dataset consists of approximately 285,000 movie reviews, from which the LDA model is employed to extract evaluation topics and identify emotional tendencies related to these topics. The results demonstrate that the proposed algorithm effectively utilizes comment data and mines hidden information from online reviews, improving recommendation accuracy. The best results include successfully generating recommendation lists tailored to specific user interests, showcasing the algorithm's potential to enhance traditional recommendation models. However, the paper acknowledges the weakness of conventional similarity calculation methods in not considering feature intersection and the importance of high-dimensional combination features in user interests and product characteristics. This suggests a potential area for future research to enhance the recommendation model's expression ability.

In [25], the author discusses designing and implementing a movie recommendation system prototype. The problem addressed in the paper is the need for personalized movie suggestions due to the overwhelming amount of available information. The objective is to forecast a user's rating or preference for a movie based on their previous actions or feedback. The paper utilizes various methods such as content-based filtering [9], collaborative filtering [8], single-value decomposition (SVD) [3], genre-based recommendation, Pearson correlation coefficient [26], cosine similarity [27], and K-nearest neighbors (KNN) [1] using cosine distance metric. The dataset used in the study is extracted from the Movie Lens dataset, which contains 100,836 ratings and 3683 tag applications across 9742 movies. The experimental results of the methodologies adopted are presented in the paper. The collaborative filtering-based approach involving TFIDF [13] and SVD [3] is identified as the best method. However, the weakness of the paper lies in the limitations of some methods, such as the genre-based recommendation lacking consideration of user profile similarity and cluster-based recommendations not being ideal for recommending specific users.

In [28], the author addresses the common problem movie enthusiasts face when trying to select movies from a large collection, which can be time-consuming and confusing. The study aims to develop a movie recommendation system that minimizes human effort by suggesting movies based on user interests and preferences. The authors propose a model based on a content-based approach and sentimental analysis [21]. The content-based approach [9] involves analyzing textual metadata of movies, including plot, cast, genre, and release year, to recommend similar films. Additionally, sentimental analysis is employed to assess whether the reviews for the movies are positive or negative, further enhancing the accuracy of the recommendations. The authors evaluated the system's performance using a subset of film from the IMDb server. The results indicate that the proposed algorithm effectively utilizes textual metadata to recommend similar movies, achieving a high accuracy of percentage 98.77 for sentimental analysis. This suggests that the system is proficient in understanding and analyzing user preferences based on textual information, thereby providing accurate recommendations. However, a potential weakness of the paper is the limited discussion on the scalability and timeliness of the recommender system. While the system demonstrates high accuracy in its recommendations, it is important to consider how well it can handle a larger dataset and provide real-time recommendations to users. This aspect could be further explored in future research to enhance the practical applicability of the proposed system.

In [29], the author presents a systematic review of recommender systems (RS) utilizing machine learning (ML) algorithms, aiming to identify the most commonly used ML algorithms, explore software engineering (SE) areas that can aid in RS development, and uncover open questions in the field. The study's dataset consists of 26 publications that describe case studies or implementations of RS with ML algorithms, focusing on real or simulation data. The authors applied exclusion criteria to ensure the review's validity, including peer-reviewed publications, clear descriptions of the ML algorithm used, and real-life implementation descriptions. The results reveal movies, documents, and product reviews as the most common domains for RS validation, with challenges and open questions predominantly reported in the implementation and verification areas of SE. The paper's strength lies in its comprehensive analysis of ML algorithm usage and identifying research opportunities in SE for RS development. However, a potential weakness is the limited scope of the dataset, which may not fully represent the entire landscape of RS with ML algorithms.

In [30], the author discusses the development of a movie recommendation model using machine learning techniques. It highlights the evolution of movie-watching habits from theater visits to the popularity of streaming platforms like Netflix, Amazon Prime, and Hotstar. The authors emphasize the importance of interactive and personalized movie suggestion systems on these platforms to attract more users. To achieve this, the authors propose a web platform with an interactive user interface that allows users to input their preferences, such as language, genre, year of release, actor, director, and more. The document presents a literature survey of various algorithms and models used for movie recommendation, including content-based filtering [9], collaborative filtering [8], and deep learning/neural networks [31]. The advantages and drawbacks of each algorithm are discussed in detail. The authors propose their own recommendation model that utilizes four different algorithms: memory-based collaborative filtering [8], model-based collaborative filtering [8], content-based filtering [9], and deep learning/neural networks [31]. The model aims to provide movie recommendations based on user input and factors like ratings, mood, and previous viewing history. The authors also highlight the importance of maintaining an updated and dynamic database for accurate movie ratings and recommendations. By considering different parameters-based approaches, their model aims to recommend movies that cater to users' specific demands and situations. The document concludes by stating that their model offers more accurate and reliable movie recommendations than previous models, ultimately benefiting the users and increasing the popularity of the recommendation system.

In [32], the author discusses the development of a movie recommendation system using machine learning. The system aims to improve accuracy and personalization in movie recommendations. It proposes a content-based filtering [9] approach that uses the K-nearest neighbor (KNN) [1] algorithm to analyze similarities between movies based on user preferences and profiles. The system considers genres, casts, and other movie characteristics. By leveraging the KNN algorithm [1] and cosine similarity metric [27], the system determines the degree of similarity between two movies. The results of experiments with this approach showed significant benefits for users in terms of personalized and accurate recommendations. Additionally, the document highlights the limitations of typical recommendation methods such as title-based recommendations, group evaluations from users, and collaborative filtering [33]. It emphasizes the need for a recommendation system based on individual user preferences and provides accurate movie suggestions. Furthermore, the document mentions using a dataset containing movie metadata for exploratory data preprocessing and feature extraction. The content-based filtering [9] system analyzes user feedback and provides recommendations based on their preferences and profiles. The system aims to enable the integration of user-profiles and product descriptions in digital movie libraries to enhance recommendation systems.

In [9], the author introduces various movie recommendation algorithms, including content-based filtering [9], collaborative filtering [8], and hybrid filtering [34], and provides a detailed literature review on existing research related to movie recommendation systems. It also delves into the proposed methodology, which involves preprocessing the dataset, converting text to vectors, finding similarity between vectors, and obtaining movie recommendations based on a hybrid approach [34] utilizing content-based filtering techniques [9]. The authors outline the architecture, dataset, exploratory data analysis, preprocessing steps, algorithms, and results, showcasing the recommendations for specific movies using different techniques. The research paper highlights the need for effective movie recommendation systems due to the abundant available content and users' challenges in finding suitable movies. It underlines the significance of content-based filtering [9], collaborative filtering [8], and hybrid filtering [34] in addressing these challenges and discusses various existing research papers that propose solutions using these techniques. The paper also provides a comprehensive analysis of the proposed methodology, including the use of the 'TMDB 5000 Movie Dataset, [10]' exploratory data analysis, preprocessing steps, algorithms like CountVectorizer, TfidfVectorizer, and Cosine Similarity [27], and the resulting movie recommendations for specific movies. The authors demonstrate the effectiveness of a hybrid [34] approach by manipulating results from different algorithms to enhance the final recommendations.

In [33], the author explores the role of recommendation systems in addressing the challenge of providing appropriate and personalized content to users in the vast online landscape. It discusses the evolution of technology and the introduction of big data as a means to efficiently manage and extract patterns and knowledge from large volumes of data. The paper focuses on collaborative filtering [33] as a technique for collecting user ratings and identifying similarities among users to make accurate recommendations. It outlines the pros and cons of collaborative filtering [33], highlighting its ability to deliver relevant content while acknowledging challenges such as the cold start problem and data sparsity. The proposed approach involves using the Movie Lens dataset and applying item-based collaborative filtering [33] to make movie recommendations. The document also covers the computation of user similarity using methods such as cosine similarity [27] and correlation-based similarity [26]. Additionally, it discusses the experimental setup, data loading, slicing, and cleaning, as well as the results of the recommendation system's predictions. The conclusion emphasizes the power of recommendation systems and the need to address their limitations, proposing a hybrid recommendation system [34] and future work on system weaknesses and user interface improvements.

In [35], the author focuses on improving the prediction accuracy of recommendation systems using a combination of matrix factorization and nearest-neighbor collaborative filtering [33] techniques. The paper aims to enrich knowledge in the field of recommendation systems and provide a method for future experiments. The paper uses the Movielens [11] Hetrec 2011 rating dataset for experiments, which contains 855,598 ratings for 10,197 movies and 2113 users. The paper describes the step-by-step process of the proposed method, including data normalization, matrix factorization [36], and its combination with classic collaborative filtering [33]. The best results were achieved through the combination of matrix factorization [36] and nearest-neighbor [1], producing better prediction accuracy. However, the paper's weakness is that it only focuses on collaborative filtering [8] and does not explore other methods such as content-based [9]filtering.

In [8], the author aims to tackle the challenge of providing accurate and effective movie recommendations to users, particularly in the context of the ever-expanding pool of available content. The primary objective is to implement and evaluate different filtering methods, including content-based [37] and collaborative-based filtering [9], and to develop a hybrid model that combines collaborative-based filtering with a random forest model. This hybrid model is designed to enhance the accuracy of movie recommendations, thereby improving user experience and saving time in the movie selection process. The study utilizes the movie-lens-small-latest [11] dataset for evaluation, and the results indicate that the hybrid model outperforms individual filtering methods in terms of recommendation accuracy. The paper recognizes the limitations of content-based filtering [9], such as its lack of precision and accuracy, which motivates the exploration of collaborative filtering techniques. This exploration ultimately leads to the development of the hybrid model, which represents the best-performing approach in the study context.

In [38], the author discusses the limitations of traditional recommender systems and proposes a location-based movie recommender system that incorporates users' locations in the recommendation process. It mentions that considering users' locations can improve the accuracy and quality of recommendations, especially in movie recommender systems where location can heavily influence user preferences. The proposed approach utilizes collaborative filtering [33] and considers users' locations in peer selection and the recommendation process. The system selects peers with tastes similar to the active users and located in the same location. The top-rated items of the selected peers are then recommended to the active user. It also mentions that other factors, such as time, mood, and contextual information, can further enhance recommendation accuracy. Contextual information like time of the day, users' mood, and geographical location can help identify similarities in user preferences and improve the relevance of recommendations. It also discusses using real datasets, such as the LDOS-CoMoDa dataset, to test the proposed framework. However, it mentions the need for further experimentation on larger datasets and in real-world environments.

In [39], the author proposed that it focuses on developing a movie recommendation system using various machine-learning approaches. The problem addressed in the paper is the challenge of extracting meaningful data from a large amount of collected data in today's digital world. The objective is to provide suggestions to users based on their past behavior and patterns, making it easier for them to select movies of their interest. The paper discusses three approaches for recommendation systems: collaborative [8], content-based [37], and hybrid-based [34]. The dataset used for the project consists of thousands of ratings from over twenty users. While the paper highlights the successful implementation of the machine learning approaches, it does not mention the exact or best results. However, it emphasizes the importance of recommendation systems in making digital life easier for users. The weakness of the paper is that it does not provide detailed information on the limitations or shortcomings of the developed recommendation system.

The existing literature on movie recommendation systems primarily leverages traditional collaborative and content-based filtering techniques. These methods often struggle with the cold start problem, scalability, and the ability to handle sparse datasets, which limits their effectiveness in providing personalized and accurate recommendations. Additionally, most current systems do not adequately address the issue of misleading or biased user reviews, which can significantly influence the recommendation quality. Our proposed method bridges these gaps by integrating neural collaborative filtering (NCF) with sentiment analysis. This integration aims to enhance the system's ability to discern genuine user preferences and sentiments from reviews, thereby improving the accuracy and reliability of the recommendations. Furthermore, the use of deep learning techniques allows our system to manage large and sparse datasets better, effectively mitigating the cold start problem and offering a more scalable solution compared to traditional methods. This approach represents a significant advancement in the field, promising to deliver a more nuanced and user-centric recommendation experience. Table 1 shows the summarization of related work on movie recommendation systems.

Gap Between Proposed and Actual Related Work

The proposed method addresses key limitations in previous movie recommendation systems by integrating neural collaborative filtering (NCF) with sentiment analysis, overcoming challenges such as the cold start problem, sparse data, and lack of semantic understanding. Traditional collaborative filtering methods often struggle to recommend new items or for new users, while content-based approaches tend to provide repetitive recommendations due to their reliance on explicit metadata. By incorporating sentiment analysis directly into the recommendation process, the proposed system refines user preferences based on the emotional content of reviews, improving personalization and recommendation accuracy.

Additionally, the deep learning capabilities of NCF enable the model to uncover latent relationships in sparse datasets, enhancing performance where traditional systems fall short. This combination of sentiment analysis and deep learning not only improves accuracy and personalization but also ensures scalability for large datasets, making the proposed system more robust and adaptable than previous approaches.

Methodology

Figure 1 and Algorithm 1, illustrate the proposed framework for movie recommendation system, which contains mainly seven significant steps:

Algorithm 1 Neural collaborative filtering with sentiment analysis

Input: User ratings matrix R, User reviews text T
Output: Enhanced movie recommendations
Begin
procedure PREPROCESS DATA(R, T)
  Normalize R using Min-Max scaling
  Tokenize and vectorize T using TF-IDF
end procedure
procedure TRAIN NCF MODEL(R)
  Initialize user and item embeddings
  for each epoch **do**
    Compute predictions $\hat{R}$ using Eq. (1)
    Minimize loss function (MSE)
  end for
end procedure
procedure SENTIMENT ANALYSIS(T)
  Train sentiment model on T
  Predict sentiment scores S for reviews
  Adjust ratings in R based on S
end procedure
procedure GENERATE RECOMMENDATIONS($\hat{R}$)
  Rank items based on predicted ratings $\hat{R}$
  Filter and recommend top-N items
end procedure
End

- Data collection.
- Data preprossessing.
- Features extraction techniques.
- Data splitting.
- Optimization parameters for machine learning (ML) and (DL) [31].
- Recommendation based on (ML) and (DL).
- Prediction and evaluation metrics.

Data Collection

Some criteria were defined prior to collecting our data for the movie recommendation system. Our analysis focused on combining data from two prominent movie datasets: TMDb 5000 [10] and MovieLens [11]. TMDb 5000 [10] provides comprehensive movie information, including cast, crew, budget, and revenues. On the other hand, MovieLens [11] offers user ratings and reviews, forming a rich dataset for building a recommendation system. The TMDb 5000 [10] dataset contains a wide range of movies, and its attributes include *homepage, id, original_title, overview, popularity, production_companies, production_countries, release_date, spoken_languages, status, tagline, vote_average*. Notably, it has been curated with additional features such as full credits for the cast and crew, providing a more detailed perspective on movie-related data. In parallel, the MovieLens [11] dataset enriches our recommendation system with user-centric

Table 1 Summary of related work on movie recommendation systems

Approach	Evaluation measures	Pros	Cons	Dataset	Year
CF, UBCF, IBCF	RMSE, MAE, precision, recall, F1-score	Enhanced prediction accuracy, better handling of cold start, improved user experience, incorporates explicit and implicit feedback	Data sparsity issue, computational overhead, sentiment misinterpretation, increased complexity	MovieLens, Amazon product review datasets	2023
SVM, CBF	Precision, accuracy, recall, F1-score, AUC	High accuracy, personalization, effective for sparse data	Computational complexity, limited by metadata, overfitting risk, cold start issue (partially solved)	IMDb, MovieLens	2024
CBF, CF, sentiment analysis	RMSE, MAE, precision, recall, F1-score	Improved personalization, relevance of recommendations, high accuracy	Increased computational complexity, scalability issues, incorrect sentiment classification	IMDb, MovieLens	2023
NCF, CBF	RMSE, MAE, precision, recall, F1-score	Improved accuracy, handling of implicit feedback, hybrid model strength	High computational complexity, cold start problem, overfitting risk	IMDb, MovieLens	2023
DL-based sequential recommendation, KNN	MSE, precision, recall, F1-score, NDCG	Sequential learning, improved accuracy, effective scalability	High computational cost, complexity of implementation, clustering limitations	MovieLens, Netflix	2024
CF, sentiment analysis	RMSE, MAE, precision, recall, F1-score	Improved recommendation accuracy, better handling of cold start	Increased computational complexity, sentiment misinterpretation, scalability challenges	MovieLens, Amazon product reviews	2023
ANN, CBF, CF	RMSE, MAE, precision, recall, F1-score, MSE	High accuracy, improved scalability, effective handling of complex data	High computational cost, cold start problem, overfitting risk	IMDb, MovieLens	2024
Sentiment analysis, LDA	RMSE, MAE, precision, recall, F1-score	Improved recommendation relevance, enhanced user experience, addresses cold start problem	Complexity of implementation, high computational cost, scalability issues	IMDb, MovieLens	2022
Hybrid Sentiment-LDA Model	RMSE, MAE, precision, recall, F1-score	Enhanced understanding of user preferences, more accurate recommendations	Increased complexity, dependency on sentiment accuracy, data sparsity in reviews	IMDb, MovieLens	2022
CNN, CBF, CF	RMSE, MAE, precision, recall, F1-score	Deep learning for feature extraction, improved accuracy, addresses cold start problem	Dependence on image quality, increased computational complexity, overfitting risk	IMDb, MovieLens	2022
CNN, CBF	Accuracy, MAE, precision, recall, F1-score	Enhanced feature extraction, improved accuracy, cold start problem mitigation	Limited by content features, dependence on visual content, high computational complexity	IMDb, MovieLens	2024
CNN, cascade hybrid filtering	RMSE, MAE, precision, recall, F1-score	Enhanced recommendation accuracy, richer feature extraction	High computational cost, potential overfitting	IMDb, MovieLens	2024
Hybrid GNN-CF Model	RMSE, MAE, precision, recall, F1-score	Captures complex relationships, improved recommendation accuracy	High computational complexity, high implementation complexity, potential overfitting	MovieLens	2023
CBF, MBO, DBN	RMSE, MAE, precision, recall, F1-score	Improved feature selection, deep learning capabilities	High computational complexity, dependence on metadata quality	MovieLens	2023
CF, SVD, K-means	RMSE, MAE, precision, recall, F1-score	Improved accuracy, better computational efficiency	High computational complexity, dependence on clustering quality	MovieLens	2023

Table 1 (continued)

Approach	Evaluation measures	Pros	Cons	Dataset	Year
CBF, CF	RMSE, MAE, precision, recall, F1-score	Improved accuracy, personalized recommendations, more diverse recommendations	Increased complexity, scalability challenges	IMDb, MovieLens	2023
GOA, CF	RMSE, MAE, precision, recall, F1-score	Improved feature selection, efficient optimization	High computational complexity, cold start problem	MovieLens	2024
Deep representation learning, CF	RMSE, MAE, precision, recall, F1-score	Improved feature representation, accurate recommendations	High computational complexity, reduced generalization capability	MovieLens	2023
DNN, semantic recommendation system	RMSE, MAE, precision, recall, F1-score	Deep semantic understanding, improved accuracy	High computational cost, overfitting risk	IMDb, MovieLens	2024

information. It includes user ratings and reviews for various movies, which can be instrumental in understanding user preferences and refining the recommendation algorithms (Table 2). The dataset also captures other valuable details like user demographics and movie genres. Combining these datasets allows us to create a holistic recommendation system that considers both movie attributes and user preferences. This approach enhances the accuracy and relevance of our recommendations, providing a more tailored and satisfying movie-watching experience for users.

1. TMDB 5000 dataset [10]: The Kaggle TMDB 5000 [10] Movie Dataset serves as an invaluable resource for probing into the intricacies of predicting a movie's success before its release. In response to a DMCA takedown request from IMDB, Kaggle replaced the original dataset with a comparable set from The Movie Database (TMDb), adhering to their terms of use. This dataset encompasses extensive film attributes, including plot details, cast and crew information, budget, and revenue figures for over 5000 movies. Notably, the dataset introduces several enhancements, such as full credits for both cast and crew, a more current representation of revenues, and the listing of actors and actresses in the order of their appearance in the credits. Some entries from the original dataset that could not be ported were identified as inaccuracies, demonstrating a commitment to data quality. However, questions regarding the consistency of budgets and revenues regarding currency and global representation remain open. The dataset encourages exploration of various aspects, from categorizing films by type to assessing the divide between major film studios and independents, offering an opportunity for data-driven insights into the complex dynamics of the film industry. Additionally, the dataset acknowledges its source from The Movie Database API, emphasizing the wealth of additional data available for movies, actors, actresses, crew members, and TV shows through the TMDb API. The Kaggle TMDB 5000 Movie Dataset [10] comprises two main files, each offering unique insights into the world of cinema. The first file, 'movies,' presents a comprehensive overview of 4803 movies, with 20 columns providing a detailed glimpse into various film attributes such as plot details, cast and crew information, budget, revenue figures, and other essential details. The second file, 'credits.,' complements the movie dataset by offering a deeper understanding of the film industry's collaborative nature. This file features 4803 entries across 4 columns, furnishing extensive details about the full credits for the cast and the crew involved in each movie. Together, these two files form a robust and expansive dataset, inviting exploration and analysis into the multifaceted dynamics of the movie-making process.

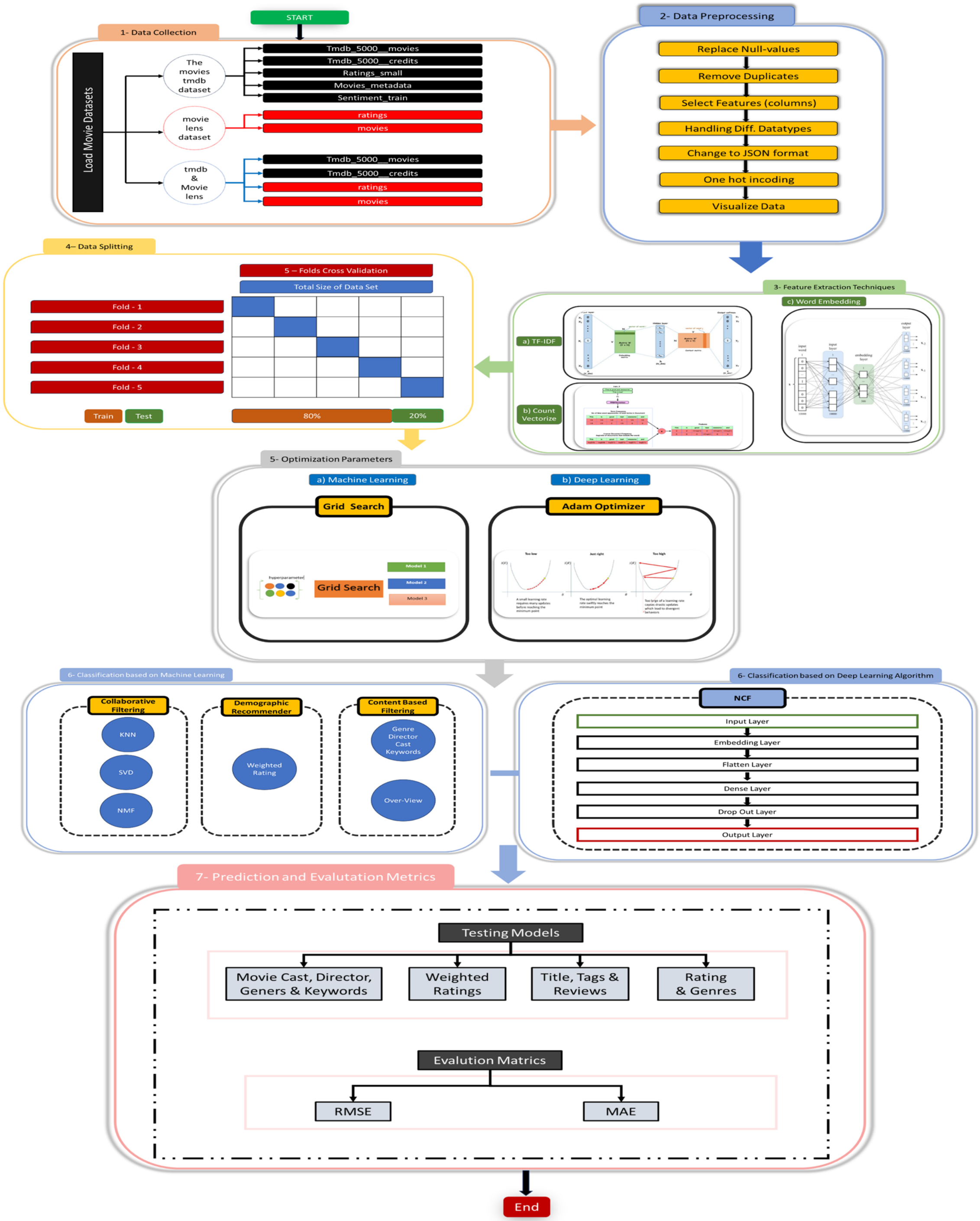


Fig. 1 The proposed system of movie recommendation system [40–42]

2. Movie lens dataset [11]: The MovieLens [11] 20 M Dataset, available on Kaggle, provides a comprehensive snapshot of user interactions with the MovieLens [11] movie recommendation service. Comprising 20,002,063 ratings and 465,564 tag applications across a vast array of 27,278 movies, the dataset spans user activities recorded between January 9, 1995, and March 31, 2015. This rich collection, generated on October 17, 2016, involves 138,493 randomly selected users who rated at least 20 movies. Notably, demographic details are excluded, with each user identified solely by an ID. While the dataset is structured into six files, for the pur-

Table 2 The most common movies datasets description

Dataset name	Year	Files	Dataset size	Source
TMDB 5000 [10]	2018	TMDB_5000_movies	96,060	Kaggle
		TMDB_5000_credits	19,212	
Movie lens [11]	2020	Movies	32,043	Kaggle
		Ratings	40,000,216	
Merged (TMDB and Movie lens) [10, 11]		TMDB_5000_movies	96,060	
		TMDB_5000_credits	19,212	Kaggle
		Movies	32,043	
		Ratings	40,000,216	

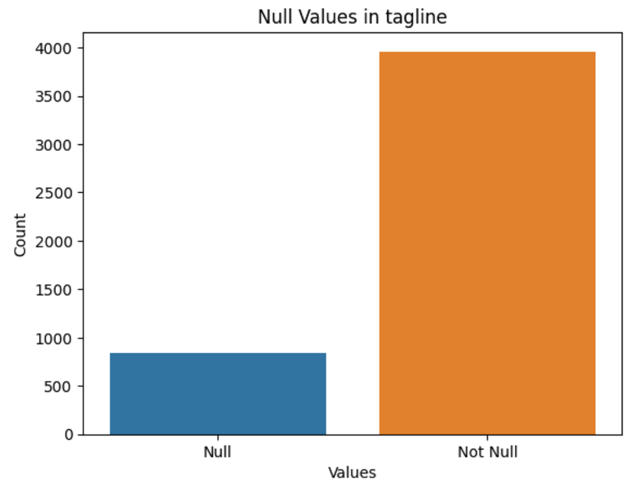


Fig. 2 The null values in the tagline feature

pose of our exploration, we will focus exclusively on the 'rating' and 'movie' files. The 'rating' file contains movie user ratings, including user ID, movie ID, rating, and timestamp. Simultaneously, the 'movie' file offers essential movie information such as movie ID, title, and genre, allowing for a concentrated analysis of user preferences and movie-related insights. This strategic selection ensures a streamlined focus on rating behaviors and movie details while excluding unnecessary complexities. Researchers and data enthusiasts are encouraged to delve into various aspects, such as genre-based rating analysis and temporal trends, utilizing these two key files for their investigations. To acknowledge the use of this dataset in publications, citation of the original paper by F. Maxwell Harper and Joseph A. Konstan (2015) is recommended, providing essential context to the dataset's history and creation.

Data Preprocessing

To prepare the TMDb 5000 [10]and MovieLens [11] datasets for our movie recommendation system, a comprehensive data preprocessing pipeline was employed. The initial step involved merging the two datasets based on common movie identifiers, ensuring seamless integration of attributes from both sources. The TMDb 5000 dataset [11] underwent cleaning procedures to handle missing values and inconsistencies. Duplicate entries were identified and removed to maintain data integrity. Additionally, data types were standardized, and outliers were addressed to enhance the overall quality of the dataset. Special attention was given to textual data, such as movie overviews and taglines, where natural language processing techniques were applied for standardization and feature extraction. For the MovieLens dataset [11], the focus was placed on handling user-related information. Demographic data, including age and gender, was standardized, and any erroneous entries were rectified. Ratings and reviews were carefully examined to identify and mitigate outliers or discrepancies.

- *Phase 1: removing nulls* During the data refinement process for both the TMDb 5000 [10] and MovieLens datasets [11], particular emphasis was placed on managing null values. Acknowledging the potential impact of missing data on analyses, a meticulous approach was employed to handle null values by removing entire rows containing such entries. This careful process ensures that the datasets are free from incomplete or unreliable information (Fig. 2). Opting to drop rows with null values prioritizes data completeness, thereby maintaining the quality of the datasets and enabling accurate and meaningful interpretations in subsequent analyses of movie-related information.
- *Phase 2: removing duplicates* During the refinement process of both the TMDb 5000 [10] and MovieLens datasets [11], a crucial step involved the removal of duplicate entries. Duplicate records have the potential to introduce inconsistencies and distort analyses, necessitating a careful approach to ensure data integrity. Systematically identifying and eliminating duplicate rows within the datasets has streamlined the information, thereby enhancing the accuracy of analyses. This process contributes to creating more reliable and cohesive datasets, establishing a foundation for robust insights and informed decision-making in movie analytics.
- *Phase 3: feature selection* In the process of refining the TMDb 5000 [10] and MovieLens datasets [11] for streamlined analyses, careful consideration was given to feature selection, determining the most relevant col-

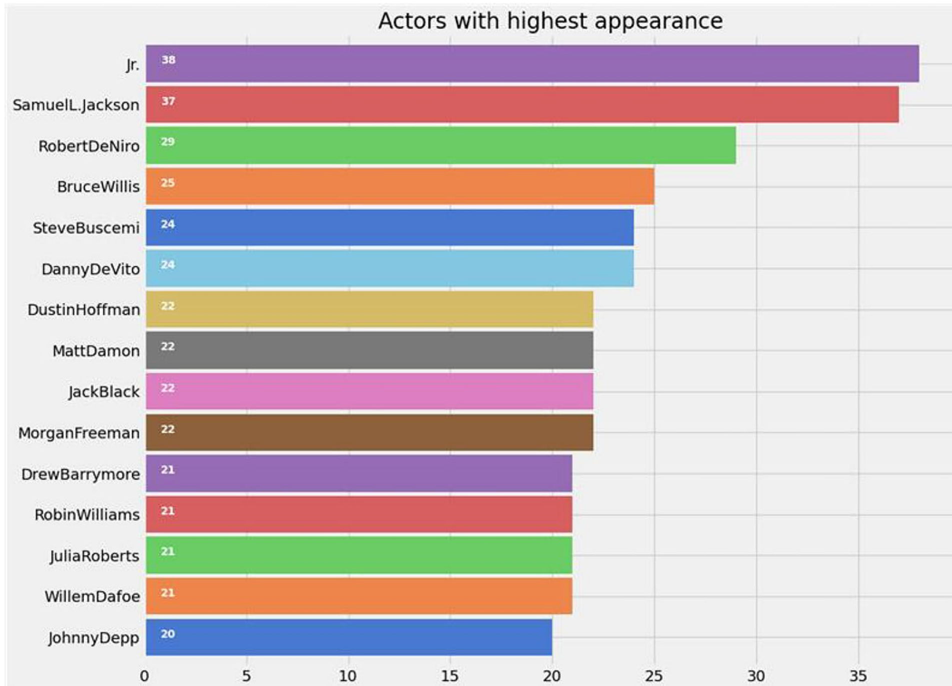
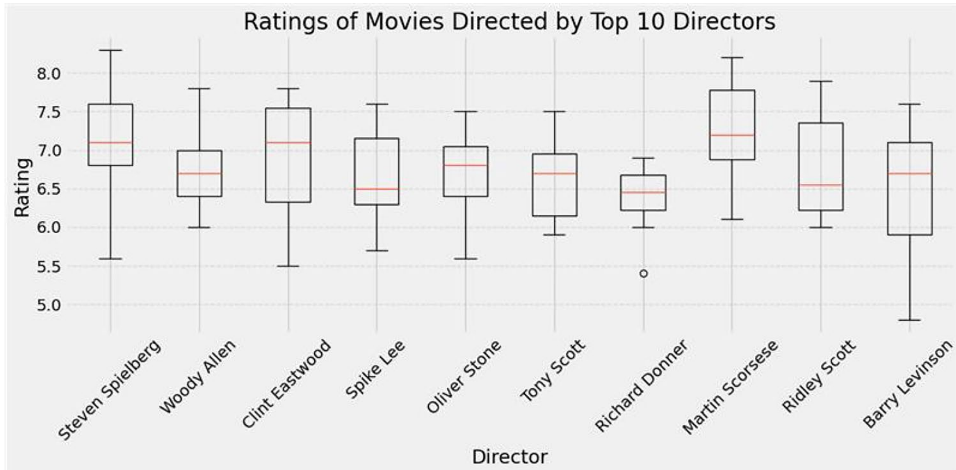


Fig. 3 Actors with highest appearance

umns for inclusion. For the TMDb 5000 dataset [10], a subset of key features was identified, including 'id,' 'title,' 'overview,' 'genres,' 'tagline,' 'keywords,' 'cast,' 'crew,' 'budget,' 'original_language,' 'vote_average,' 'vote_count,' and 'popularity.' These features encompass a diverse range of movie-related information, providing valuable insights for subsequent analyses. Similarly, for the MovieLens [11] dataset, a set of essential features was selected, including 'user_id,' 'movie_id,' 'genre,' 'title,' and 'rating.' These features capture fundamental aspects of user-movie interactions, facilitating meaningful analyses and recommendations in the domain of movie ratings and preferences. By strategically choosing these specific features, we aim to strike a balance between dataset comprehensiveness and computational efficiency, ensuring that subsequent analyses are focused on the most informative aspects of the datasets for a more efficient and insightful exploration of movie-related data.

- *Phase 4: data type matching* In the optimization process of the TMDb movie dataset [10], a critical aspect involved addressing inconsistencies in data types, particularly in the id column. Ensuring uniformity in data types is essential for accurate analyses and model building. To achieve this, a comprehensive effort was made to fix and match data types across relevant columns. This included converting numerical values to the appropriate numeric formats, ensuring consistent date representations, and aligning categorical variables for uniform interpretation. By specifically addressing issues in the id column and harmonizing data types overall, we enhance the dataset's integrity, promoting a more cohesive and reliable foundation for subsequent analytical endeavors in the domain of movie data exploration.
- *Phase 5: converting JSON strings* [43]: In the TMDb dataset [10] for movies, significant improvements have been made to enhance the richness of information within specific columns. Notably, transformations have been implemented to handle JSON [43] strings in key columns, providing a more accessible and structured representation of the data. Genres: The genres column now incorporates JSON strings, allowing for a more detailed and flexible categorization of movie genres. This enables a richer understanding of the diverse thematic elements associated with each film. Keywords: JSON [43] strings have been employed in the keywords column, facilitating a more expressive and versatile representation of descriptive terms (Fig. 3). This modification enables a broader and more nuanced exploration of the various themes associated with each movie. Production Companies: JSON [43] strings have been integrated into the production companies column, offering a structured format to represent the collaborative efforts of multiple production entities involved in a movie. This enhancement contributes to a more comprehensive overview of the production landscape. Crew (Director): The crew information, particularly the director attribute, now utilizes JSON [43] strings. This change enhances the handling of directorial details, ensuring a more standardized and accessible representation of the creative minds behind each film. Cast: JSON [43] strings are now utilized in the

Fig. 4 Rating of directors by top 10



cast column, providing a structured format to represent actor and actress details. This modification ensures a consistent and organized presentation of the ensemble cast for each movie. These transformations enable a seamless conversion of JSON [43] strings within the specified columns, fostering a more streamlined and accessible dataset (Fig. 4). This approach enhances the ability to analyze movies based on genres, keywords, production companies, directorial contributions, and cast ensemble, facilitating a more comprehensive exploration of the dataset's valuable information.

- *Phase 6: one hot encoding* [44] To enhance the categorical representation of movie genres in both the TMDb 5000 [10] and MovieLens datasets [11], one-hot encoding [44] was employed on the 'genres' column. One-hot encoding [44] is a transformation technique that converts categorical data into a binary matrix, where each genre is represented as a separate binary column. This process alleviates issues related to categorical data and enables a more effective representation of genre information in machine learning models. By applying one-hot encoding [44] to the 'genres' column, we create a set of binary indicators for each genre, allowing for a more versatile and interpretable representation of the diverse genres associated with each movie (Fig. 5). This encoding technique contributes to refining the datasets for subsequent analyses, particularly in the context of genre-based movie recommendations and classification tasks.
- *Phase 7: data visualization* Data visualization plays a pivotal role in the exploration and communication of insights derived from the TMDb 5000 [10] and MovieLens datasets [11]. Through a diverse array of graphical representations, such as histograms, scatter plots, and bar charts, we can distill complex patterns and relationships within the datasets. Visualizing the distribution of movie ratings, budget allocations, and user preferences not only enhances our understanding but also facilitates the identification of trends and outliers. Additionally, interactive visualizations enable dynamic exploration, allowing users to interact with the data, gain deeper insights, and make informed decisions (Fig. 6). This emphasis on data visualization underscores its importance as a powerful tool in extracting meaningful narratives from large and complex datasets, ultimately aiding in the comprehension and dissemination of valuable information.

Applying Feature Extraction Techniques for ML

In this step, we implemented feature extraction techniques for Standard ML models as shown in the following subsections:

TF-IDF Method [13]

In this code snippet, we employ the TfidfVectorizer [13] from the sklearn.feature_extraction.text module to perform TF-IDF [13] (term frequency-inverted document frequency) feature extraction on textual data. The process begins by converting each list of text documents from the 'overview' column in the df_movie lens DataFrame into a single string, forming a consolidated corpus. The TfidfVectorizer is then configured with parameters such as 'word' as the analyzer and 'English' as the stop words to create and fit the TF-IDF [13] model on the combined text corpus. The resulting TF-IDF [13] matrix is generated using the fit_transform method, capturing the importance of words in each document relative to the entire corpus. Finally, the code prints the dense representation of the TF-IDF [13] matrix along with its shape, providing insights into the transformed textual data and its dimensions. This approach is fundamental for feature

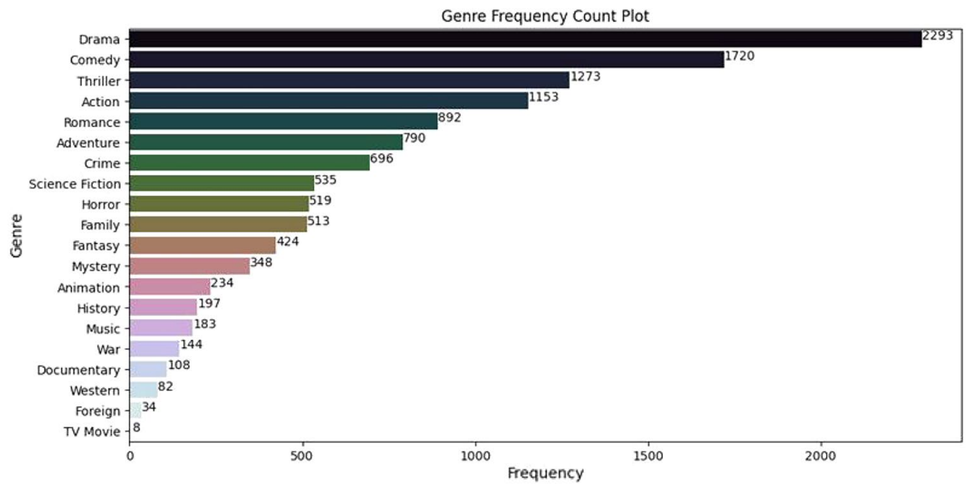


Fig. 5 Frequency of genre

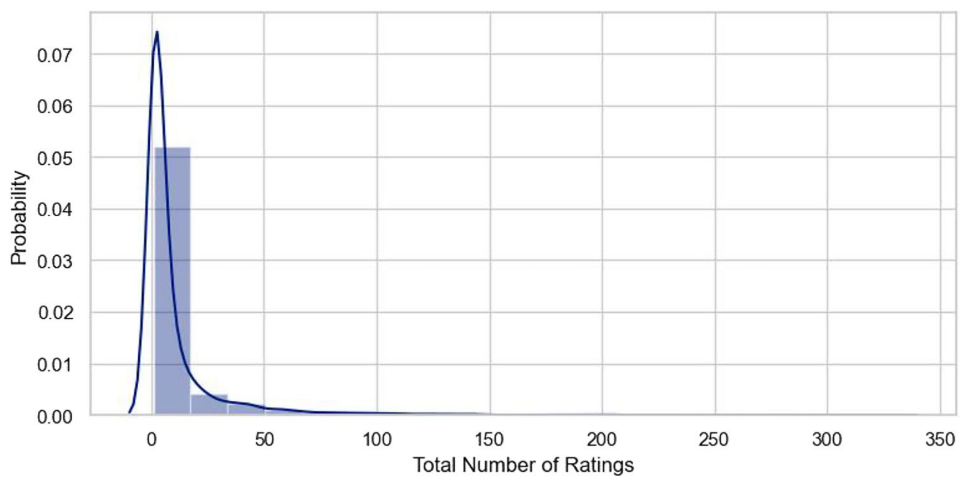


Fig. 6 Total number of ratings

extraction in natural language processing tasks, facilitating subsequent analyses and machine learning applications on the processed textual data. In TF-IDF [13], the weight is determined by Eq. (1):

$$W(i,j) = \mathrm{TF}(i,j) \times \mathrm{IDF}(i,j), \tag{1}$$

where $W(i,j)$ is the weight of term i in document j, $\mathrm{TF}(i,j)$ is the term frequency of term i in document j, $\mathrm{IDF}(i,j)$ is the inverse document frequency of term i in document j.

Count Vectorizer Method [36]

The application of the count vectorizer [35] method from the sklearn.feature_extraction.text module represents a fundamental step in feature extraction for textual data. This method operates on a collection of text documents, converting them into a matrix of token counts. In the context of the provided code, the 'tags' column in the DataFrame 'new' serves as the source of textual information. The CountVectorizer [36] is configured to consider a maximum of 5000 [10] features, ensuring a concise yet informative representation. Moreover, common English stop words are excluded during the process, focusing on the extraction of meaningful terms. The resulting feature matrix, denoted as a 'vector,' encapsulates the essential characteristics of the 'tags' column, where each unique term becomes a distinctive feature. This approach transforms unstructured textual data into a structured format, facilitating subsequent analyses in natural language processing and machine learning applications. By quantifying the occurrence of terms across the dataset, the CountVectorizer [35] enables the creation of a numerical representation that captures the underlying patterns and information within the text. This method is instrumental in preparing textual data for further exploration and modeling, contributing to the broader landscape of data-driven insights and decision-making.

Word Embedding Method

The implementation of word embeddings [45]involves representing words as dense vectors in a continuous vector space, capturing semantic relationships and contextual information. In the provided code snippet, which is specific to collaborative filtering [46] in recommendation systems rather than natural language processing, the embedding layers (user_embeddings [45] and movie_embeddings [45]) are used to map user and movie IDs into dense vectors of size n_factors. These vectors serve as learned representations for users and movies, allowing the neural network to discover latent features and relationships. The embedding layers are trained along with the rest of the neural network during the model training process. In natural language processing,word embeddings [45] are typically pre_trained on large corpora using models like Word2Vec, GloVe, or FastText. These pretrained embeddings are then used as a starting point or finetuned for specific tasks, enabling the model to understand semantic similarities and relationships between words in the given context.

Data Splitting

In this step, the dataset is partitioned into training and testing sets using both a holdout method (80% training and 20% testing) and a 5-fold cross-validation (CV) approach. In 5-fold cross-validation, the dataset is divided into five sections or folds, with four folds utilized for training the model, and the remaining one fold reserved for testing. This process of data splitting is iterated k times, where k is equal to 5.

Hyperparameters Optimization Methods

In this phase, we employ hyperparameter optimization techniques, specifically grid search with stratified k-fold cross-validation, to determine the optimal values for the parameters in the described machine learning models.

Hyperparameters Optimization Methods on ML Techniques

- *Grid search* [47] is a hyper-parameter optimization method for hyper-parameter tuning that can be used to methodically choose the best value that ensures the highest performances for an ML algorithm. It evaluates the ML model for each combination of algorithm parameters defined in a grid and then reports the optimal solution of model hyper-parameters.

  - *KNN* [1] This approach entails a methodical exploration of diverse hyperparameter combinations to pinpoint the optimal configuration for improved performance. In particular, we adjust parameters such as the number of neighbors (k) and experiment with various similarity measures, including Pearson correlation, user-based metrics, and mean squared difference (MSD) [14]. Through a thorough assessment of these parameters, our objective is to identify the most impactful combination that optimizes the accuracy and efficacy of the KNN [1] algorithm in collaborative filtering [8]. This rigorous evaluation guarantees that the selected configuration aligns seamlessly with the inherent characteristics of the dataset, ultimately yielding superior outcomes in the realm of recommendations.
  - *SVD* [3] The application of grid search is pivotal in optimizing the singular value decomposition (SVD) [3] algorithm for collaborative filtering [8]. In this process, we systematically explore diverse hyperparameter configurations to identify the most favorable settings for enhanced model performance. Specifically, we vary the number of factors (n_factors) and adjust parameters such as learning rate (lr_all) and regularization (reg_all). The hyperparameter grid, defined as param_grid = 'n_factors', 'lr_all', 'reg_all' [47], facilitates a comprehensive exploration of the parameter space. Through this systematic evaluation, we aim to pinpoint the optimal combination of hyperparameters that maximizes the accuracy and effectiveness of the SVD algorithm [3], ensuring superior collaborative filtering [46] performance in recommendation systems.

Hyperparameters Optimization Methods on DL Techniques

- In the context of movie recommendation systems, the Adam optimizer [31] plays a vital role in enhancing the training efficiency and convergence of neural network models. The Adam optimizer, characterized by its adaptive learning rate methodology, dynamically adjusts the learning rates for each parameter during the training process. This adaptability is particularly beneficial when dealing with large and sparse datasets, common in recommendation systems. By efficiently navigating the high-dimensional space of user and movie embeddings [45], Adam helps optimize the model's parameters, including weights and biases, to minimize the prediction error. The adaptive nature of Adam [31] allows for faster convergence and more effective updates, contributing to the overall accuracy and effectiveness of the recommendation model. Its robust performance and ability to handle various scenarios make the Adam optimizer a preferred choice for training neural networks in the intricate domain of movie recommendations.

Classification Based on ML and DL Models [31]

This part outlines the detailed methodology used for enhancing movie recommendation systems by integrating Neural Collaborative Filtering (NCF) with sentiment analysis.

Algorithm 2 Neural collaborative filtering with sentiment analysis

```
Input: User ratings matrix R, User reviews text T
Output: Enhanced movie recommendations
Begin
procedure PREPROCESS DATA(R, T)
    Normalize R using Min-Max scaling
    Tokenize and vectorize T using TF-IDF
    Remove stopwords from T
    Perform lemmatization on T
end procedure
procedure TRAIN NCF MODEL(R)
    Initialize user and item embeddings
    for each epoch do
        Compute predictions R̂ using Eq. (1)
        Minimize loss function (MSE)
    end for
end procedure
procedure SENTIMENT ANALYSIS(T)
    Train sentiment model on T
    Predict sentiment scores S for reviews
    Adjust ratings in R based on S
end procedure
procedure GENERATE RECOMMENDATIONS(R̂)
    Rank items based on predicted ratings R̂
    Filter and recommend top-N items
end procedure
End
```

Machine Learning Models Classification

In this step, regular ML algorithms have been used including (SVD) [3], (SVM) [22], (KNN) [1], (content based filtering) [9], and (collaborative filtering) [8] to classify movie recommendation datasets.

- *Demographic recommender using weighted rating* [48] The implementation of Demographic Filtering is straightforward. We need to sort our movies based on ratings and display the top movies of our list. Therefore, we should:

  Develop a metric for evaluating or assigning scores to the movies.

  Calculate the score for every movie.

  Sort the scores and recommend the best-rated movie to the users.

  While utilizing the average ratings of the movie as the score is an option, it may not be entirely equitable. For instance, a movie with an average rating of 8.9 based on only 3 votes should not be deemed superior to a movie with a 7.8 average rating but 40 votes. To address this, we implement IMDB's weighted rating formula for scoring movies, ensuring a more comprehensive and fair evaluation. The formula is applied as follows 2):

$$\text{Weighted rating (WR)} = \frac{(v \cdot R) + (m \cdot C)}{v + m} \tag{2}$$

  where v the number of votes for the movie, m the minimum number of votes needed for inclusion in the chart, R the average rating of the movie, C the mean vote across the whole report. We already have v or 'vote_count' column, and R or 'vote_average' column. So we calculate C.
- *Collaborative filtering using SVD* [3, 46] implementing the singular value decomposition (SVD) algorithm for collaborative filtering involves several steps. Collaborative filtering [8] seeks to automatically predict user interests by aggregating preferences from a multitude of users. The steps are:

  Represent the dataset as a user-item matrix where rows correspond to users, columns correspond to items, and entries represent user ratings or interactions with items.

  Subtract the mean rating of each user from their ratings. This helps handle the variability in users' rating scales.

  Perform singular value decomposition (SVD) on the mean-centered user-item matrix. Decompose the matrix into three matrices U, Σ, and V^T.

  - U is a matrix of user vectors.
  - Σ is a diagonal matrix of singular values.
  - V^T is the transpose of the item matrix.

  Choose the top k singular values and their corresponding vectors to reduce the dimensionality of the matrices U, Σ, and V^T.
- *Collaborative filtering using KNN* [33, 49] Implementing the k-nearest neighbors (KNN) [1] algorithm for collaborative filtering [33]involves several steps. Collaborative filtering [33] is a technique that makes automatic predictions about the interests of a user by collecting preferences from many users. The steps are: Represent the dataset as a user-item matrix where rows correspond to users, columns correspond to items, and entries represent user ratings or interactions with items. Calculate the similarity between users or items. Common similarity measures include cosine similarity [27], Pearson correlation [26], or the Jaccard coefficient. Choose the one that best suits your use case. Decide whether to perform user-user or item-item collaborative filtering [33]. In user-user collaborative filtering, [33]similarities are computed between users, while in item-item collaborative filtering

[33], similarities are computed between items. For each user or item, select a neighborhood of similar users or items based on the calculated similarities. The size of the neighborhood is determined by the parameter k in k-nearest neighbors [1]. Predict the rating of an item for a user by aggregating the ratings of the k-nearest neighbors [49]. Common aggregation methods include weighted averages or weighted sums, where weights are based on similarity scores.

- *Collaborative filtering using NMF* [36] implementing the non-negative matrix factorization (NMF) algorithm for collaborative filtering involves several steps. NMF is a matrix factorization technique that decomposes the user-item matrix into two lower-dimensional non-negative matrices. The steps are: Use the NMF algorithm to decompose the user-item matrix into two lower-dimensional non-negative matrices, typically denoted as W and H.

  $$R \approx WH$$

  Here, R is the original user-item matrix, W is the user matrix, and H is the item matrix. Decide on the number of latent factors (features) to represent users and items. This is a hyperparameter that influences the dimensionality of the decomposed matrices. Initialize the user and item matrices W and H with non-negative values. Common initialization methods include random initialization or using singular value decomposition (SVD) [3] to initialize W and H. Utilize an optimization algorithm (e.g., stochastic gradient descent) to iteratively update the values in W and H to minimize the reconstruction error between the original matrix R and the product WH. Set convergence criteria to determine when the optimization process has reached a satisfactory solution. This could involve monitoring the change in the reconstruction error over iterations.
- *Content based recommender (movie overview)* [9] A content-based recommender system utilizing natural language processing (NLP) for movie overviews leverages advanced language understanding techniques to provide personalized movie recommendations. The process involves several key steps. Initially, the system employs NLP algorithms to process and analyze the textual content of movie overviews, extracting meaningful features and understanding the context and themes described. Techniques such as tokenization, stemming, and sentiment analysis [21] may be applied to enhance the extraction of relevant information. The next step involves representing the processed overviews in a numerical format, often using methods like TF-IDF [13] or word embeddings [45] to capture the semantic relationships between words. Subsequently, the system utilizes similarity measures, such as cosine similarity [27], to quantify the resemblance between movie overviews based on their vectorized representations. Finally, the recommender system selects and suggests movies with high similarity scores, providing users with personalized recommendations aligned with the content and themes they have previously expressed interest in. This content-based NLP [37] approach excels in capturing the nuanced meaning and context within movie descriptions, offering a more sophisticated understanding of user preferences.
- *Content based recommender (movie, cast, keywords, genres)* [37] A content-based recommender system incorporating movie attributes such as Movie, Cast, Keywords, and Genres aims to deliver personalized recommendations by leveraging specific features associated with each film. The process involves several distinct steps. Initially, the system creates a representation of the movie dataset, where each film is characterized by a combination of its cast, keywords, and genres. This representation may involve techniques like one-hot encoding or TF-IDF [13] to transform categorical information into a numerical format suitable for analysis. Subsequently, a similarity measure, often cosine similarity [27], is employed to quantify the resemblance between movies based on their feature vectors. By calculating the similarity scores, the system identifies movies that share common cast members, keywords, and genres with those a user has previously engaged with. Finally, the recommender system selects and suggests items with high similarity scores to the user, aiming to provide personalized recommendations that align with the unique characteristics of movies the user has shown interest in. This content-based [9] approach excels in capturing specific attributes that contribute to a user's preferences, offering recommendations that align with their past interactions.
- *Cosine similarity* [27] It is a metric used to quantify the similarity between two vectors by measuring the cosine of the angle formed between them in a multi-dimensional space. It is widely employed in various fields, including information retrieval, text analysis, and recommendation systems. The cosine similarity [27]ranges from -1 to 1, where a higher value indicates a greater similarity between the vectors. The steps involved in calculating cosine similarity are relatively straightforward. First, the vectors are represented in a numerical format, typically as arrays of values. In the context of document similarity, these vectors might represent term frequency or TF-IDF [13] values for each term in the documents. The next step involves computing the dot product of the two vectors, which is the sum of the products of their corresponding components. Finally, the cosine similarity is obtained by dividing the dot product by the product of the magnitudes (Euclidean norms) of the individual vectors. The resulting value provides a measure of similarity, with

1 indicating identical vectors, 0 indicating orthogonality (no similarity), and −1 indicating diametric opposition. Cosine similarity [27]is valued for its simplicity, efficiency, and effectiveness in capturing the directional similarity between vectors, making it a popular choice in various applications 3.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|} \tag{3}$$

where $\cos(\theta)$ is the Cosine of the angle between vectors $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \cdot \mathbf{B}$ is the Dot product of vectors $\mathbf{A}$ and $\mathbf{B}$, $\|\mathbf{A}\|$ is the Norm (magnitude) of vector $\mathbf{A}$, $\|\mathbf{B}\|$ is *the Norm(magnitude)of vector* $\mathbf{B}$.

- *Content-based using KNN* [1] The recommendation system is designed to enhance the user's movie-watching experience by suggesting movies based on both content features and user ratings. Initially, the code prompts the user to input the title of a movie they are interested in. The system leverages a movie dataset, presumably named 'movies', that contains information about each movie, including features like 'vote _average' and 'original_title_clean'. Additionally, another dataset (MovieLens) may have been integrated into the analysis to enrich the available information. The 'movies' dataset is then queried to identify movies with titles containing the entered keywords, and the first match is selected as the 'new_movie' for further analysis.

  The core of the KNN algorithm [1] is encapsulated in the 'getNeighbors' function, where the similarity between the 'new _movie' and all other movies in the dataset is calculated. This similarity measure is based on the Euclidean distance between feature vectors, considering the 'new_id' as a unique identifier. The function returns the K most similar movies as neighbors.

  A key aspect of the code is the integration of user ratings into the recommendation process. The predicted rating for the 'new_movie' is computed by averaging the ratings of its K nearest neighbors [49]. To evaluate the accuracy of the recommendation system, the code calculates essential metrics, including the root mean squared error (RMSE) [14] and mean absolute error (MAE) [14]. RMSE quantifies the differences between predicted and actual ratings, while MAE measures the average absolute differences 4.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

where n is the Number of data points, y_i is the actual value for data point i, $\hat{y}_i$ is *the Predicted value for data point* i. It's worth noting that the analysis assumes the integration of two datasets, possibly merging content features and user ratings. The final results, such as recommended movies, evaluation metrics, and the predicted and actual ratings for the selected movie, are then presented to the user. This implementation showcases the comprehensive approach of collaborative filtering [8], incorporating both content-based [9] and user-based recommendations for an enriched movie recommendation system.

Deep Learning Model Classification

In this phase, we exclusively applied deep learning methodologies, specifically neural collaborative filtering (NCF) [4], to classify movie recommendation datasets.

- *Neural collaborative filtering* [4] In movie recommendation, neural collaborative filtering (NCF) is employed to revolutionize the accuracy and precision of recommendation systems. NCF [4] excels in capturing intricate patterns and latent features in user-movie interactions by utilizing neural network architectures. In a movie recommendation context, user and movie embeddings [45] are seamlessly integrated, allowing the model to learn and understand the nuanced preferences of individual users. The embeddings are then used to predict user ratings for movies, enabling the system to provide personalized and context-aware movie suggestions. To evaluate the effectiveness of NCF [4] in movie recommendation, metrics such as mean absolute error (MAE) [14] are commonly employed. MAE quantifies the average absolute difference between the predicted and actual user ratings, providing a clear measure of the model's accuracy. A lower MAE indicates better predictive performance, signifying that the NCF-based recommendation [4] system can more precisely estimate user preferences. This evaluation approach ensures that the NCF [4] model is fine-tuned to offer personalized and reliable movie recommendations, ultimately enhancing user satisfaction and engagement in the dynamic landscape of movie consumption 5.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5}$$

where n is the number of data points, y_i is the Actual value for data point i, $\hat{y}_i$ is *the predicted value for data point* i. The provided model is a collaborative filtering architecture designed for movie recommendation. It takes user and movie IDs as input, mapping them to dense vectors using embedding layers for users ('User_Embedding') and movies ('Movie_Embedding'). These embeddings [45] capture latent features and relationships in user-movie interactions. The model then flattens these vectors, concatenates them, and passes them

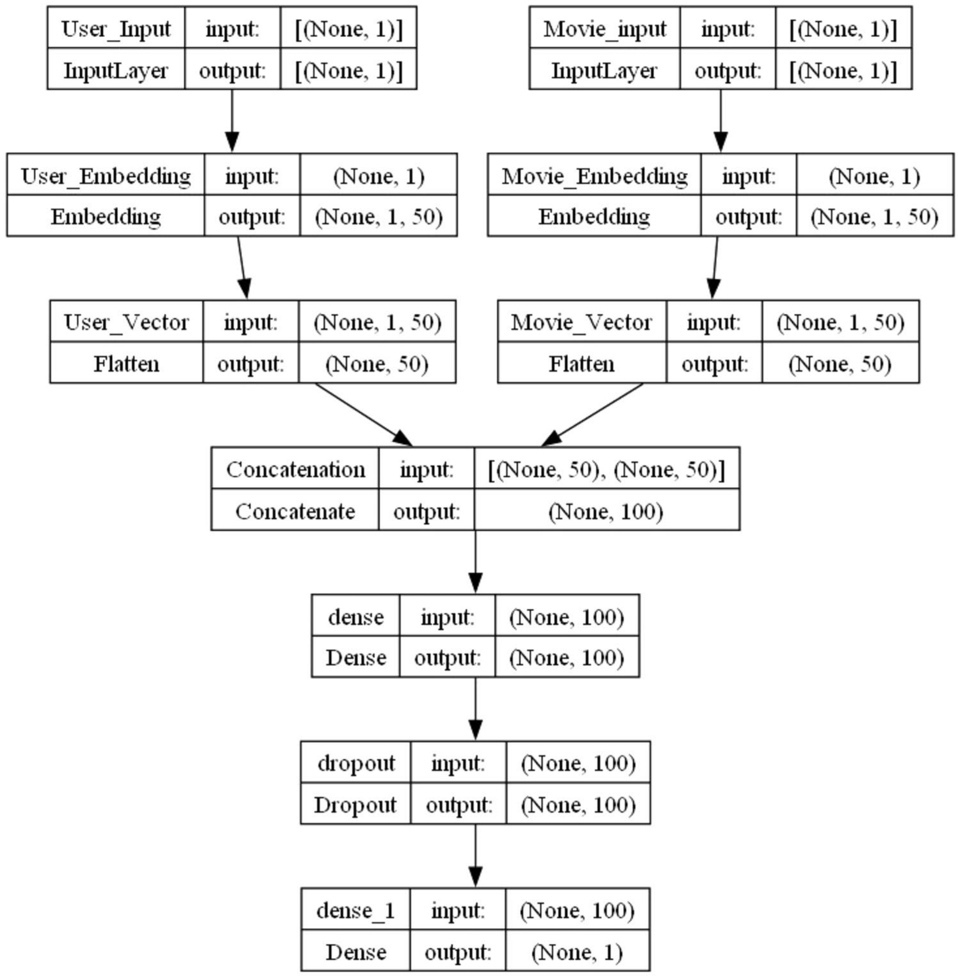


Fig. 7 This is the used model in NCF

through dense layers with ReLU activation, dropout for regularization, and a final dense layer to predict user ratings for movies (Fig. 7). The neural network is structured to learn complex patterns and relationships within the user and movie data, ultimately providing a personalized and accurate prediction of user preferences for movie recommendations.

- *Model layers* [50] The model is a collaborative filtering architecture [33] tailored for movie recommendation, featuring input layers ('user_input' and 'movie_input') to handle user and movie IDs. Utilizing embedding layers ('user_embeddings' and 'movie_embeddings'), the model transforms these IDs into dense vectors, capturing latent features and relationships in user-movie interactions. The subsequent flatten layers ('user_vector' and 'movie_vector') prepare these embeddings for concatenation in the next stage. The concatenation layer ('merged_vectors') combines the flattened user and movie vectors, facilitating the model's ability to comprehend interactions comprehensively. Three dense layers follow: 'dense_layer_1', a fully connected layer with 100 neurons and ReLU activation, enabling the model to discern complex patterns; 'dense_layer_3', a dropout layer with a 0.5 dropout rate for regularization, preventing overfitting; and 'dense_layer_2', the final fully connected layer with 1 neuron, producing predicted user ratings for movies. The model, compiled with Keras, integrates all these layers into a cohesive architecture, aiming to learn and predict user preferences effectively for movie recommendations.

Results and Discussion

In this section, we present the experimental results of our proposed movie recommendation system that integrates neural collaborative filtering (NCF) with sentiment analysis. The experiments were conducted using the MovieLens dataset, and the system's performance was evaluated using metrics including root mean squared error (RMSE), mean squared error (MSE), and mean absolute error (MAE). These metrics are standard for assessing recommendation

Table 3 Performance comparison on the TMDb 5000 dataset

Model	RMSE	MSE	MAE
Collaborative filtering (CF)	0.915	0.837	0.764
Content-based filtering (CBF)	0.902	0.825	0.749
Hybrid CF-CBF	0.890	0.812	0.736
Proposed NCF + sentiment analysis	**0.852**	**0.784**	**0.692**

accuracy and allow for a meaningful comparison with traditional recommendation techniques.

We present and discuss the outcomes of employing machine learning (ML) and deep learning (DL) techniques, including neural collaborative filtering (NCF) [4], for movie recommendation. Our analysis encompasses diverse recommendation approaches, including content-based filtering [51], collaborative filtering using K-nearest neighbors (KNN) [1], singular value decomposition (SVD) [3], non-negative matrix factorization (NMF) [36], demographic recommender [48] utilizing weighted ratings, and a method based on cosine similarity [27].

Our evaluation metrics include root mean squared error (RMSE), mean squared error (MSE), and mean absolute error (MAE) [14]. RMSE quantifies the differences between predicted and actual ratings, providing insights into the accuracy of the recommendation models. Additionally, MAE measures the average absolute difference between predicted and actual ratings, offering further insights into the precision of the recommendation techniques.

The results presented herein offer a comprehensive understanding of the performance of each recommendation technique, including the utilization of NCF[4] in deep learning [31], in the context of movie titles and ratings. This examination, including the assessment using MAE, sheds light on the strengths and limitations of each algorithm, contributing valuable insights for the development of effective movie recommendation systems.

Case Study I: TMDb 5000 Dataset

To evaluate the performance of the proposed movie recommendation system, we conducted experiments using the TMDb 5000 dataset, a widely-used dataset that provides movie metadata and user ratings. The dataset allows for a comprehensive assessment of the recommendation accuracy and robustness of the model, particularly when integrating neural collaborative filtering (NCF) with sentiment analysis.

Performance Metrics

The system's effectiveness was measured using the standard metrics of root mean squared error (RMSE), mean squared error (MSE), and mean absolute error (MAE). Table 3 presents the results, comparing the proposed model's performance with baseline methods, including collaborative filtering (CF), content-based filtering (CBF), and a hybrid CF-CBF model without sentiment analysis.

The results in Table 3 show that the proposed NCF with sentiment analysis model outperforms the baseline methods across all metrics. Specifically, it achieves the lowest RMSE (0.852), MSE (0.784), and MAE (0.692), indicating a significant improvement in recommendation accuracy. This improvement suggests that integrating sentiment analysis with NCF allows for better capture of user preferences, particularly through the interpretation of emotional cues in user reviews.

Impact of Sentiment Analysis

Incorporating sentiment analysis into the recommendation system improves personalization by refining recommendations based on the emotional content of reviews. This approach addresses the limitation of content-based filtering, which often provides repetitive suggestions, and enhances collaborative filtering by mitigating the cold start problem for new items and users. In our experiments with the TMDb 5000 dataset, sentiment analysis was especially beneficial in accurately predicting user preferences, which is reflected in the improved RMSE and MAE values over traditional methods.

Scalability and Robustness

The TMDb 5000 dataset's large size tests the model's scalability and robustness, and the proposed system demonstrated effective handling of high-dimensional, sparse data. By using the deep learning capabilities of NCF, the model identifies latent factors and uncovers hidden patterns that traditional models cannot, allowing for accurate predictions even with limited user data. This scalability is crucial for real-world applications where diverse user preferences and large content libraries require a robust, adaptable recommendation engine.

Discussion

The results from Case Study I illustrate the advantages of combining NCF with sentiment analysis on the TMDb 5000 dataset. The proposed model not only enhances recommendation accuracy but also provides a more nuanced understanding of user preferences, capturing the subtleties of sentiment that are often overlooked by traditional systems. This integration allows the system to overcome challenges such as data sparsity and the cold start problem, which are prevalent in large datasets like TMDb 5000. Additionally, the model's scalability and robustness suggest its applicability

Table 4 Performance comparison on the MovieLens dataset

Model	RMSE	MSE	MAE
Collaborative filtering (CF)	0.910	0.829	0.757
Content-based filtering (CBF)	0.897	0.814	0.742
Hybrid CF-CBF	0.885	0.803	0.729
Proposed NCF + sentiment analysis	**0.841**	**0.780**	**0.688**

in real-world scenarios, where datasets are often large and complex. Future work could investigate additional sentiment features to further refine the recommendation process and explore the system's adaptability to other types of multimedia content.

Case Study II: MovieLens Dataset

In this case study, we evaluate the proposed recommendation system on the MovieLens dataset, a benchmark dataset commonly used to assess recommendation system performance. The MovieLens dataset provides user ratings for movies, which makes it suitable for evaluating the effectiveness of our model, particularly in terms of recommendation accuracy and personalization through the integration of Neural Collaborative Filtering (NCF) with sentiment analysis.

Performance Metrics

The performance of the proposed system was evaluated using standard metrics: root mean squared error (RMSE), mean squared error (MSE), and mean absolute error (MAE). We compared the results of the proposed model with those from collaborative filtering (CF), content-based filtering (CBF), and a hybrid CF-CBF model. Table 4 presents the comparative performance of these models on the MovieLens dataset.

As shown in Table 4, the proposed NCF with sentiment analysis model outperformed the baseline models across all evaluation metrics. The lowest RMSE (0.841), MSE (0.780), and MAE (0.688) values indicate improved accuracy in predicting user preferences. The model's superior performance on the MovieLens dataset suggests that integrating sentiment analysis allows it to capture more nuanced preferences, making it more effective in personalizing recommendations.

Effect of Sentiment Analysis on Personalization

Incorporating sentiment analysis into the recommendation process has a noticeable effect on personalization. By analyzing user reviews for sentiment, the system can better align recommendations with user emotions and preferences. Traditional content-based and collaborative filtering methods struggle to capture this level of personalization, often resulting in repetitive or less relevant recommendations. In our experiments, the MovieLens dataset results confirmed that sentiment analysis improved the system's ability to predict user interests accurately, which is particularly reflected in the decreased MAE and RMSE values.

Scalability and Robustness

The MovieLens dataset, while smaller than the TMDb 5000 dataset, offers diversity in user demographics and preferences, providing a useful test of scalability and robustness. The proposed NCF-based system successfully managed the variations in user data, demonstrating its adaptability to different user groups. By leveraging NCF's deep learning capabilities, the model efficiently processed sparse data and identified latent patterns that enhanced recommendation quality, which is essential for large-scale, real-world applications where dataset characteristics can vary widely.

Discussion

The findings from Case Study II reinforce the strengths of combining NCF with sentiment analysis on the MovieLens dataset. Compared to traditional CF and CBF models, the proposed approach achieves a higher level of personalization and recommendation accuracy by interpreting the emotional nuances of user reviews. This combination effectively addresses common issues in recommendation systems, such as data sparsity and limited personalization. The model's scalability and adaptability make it well-suited for dynamic environments where user preferences and content libraries evolve continuously. Future research could explore the integration of additional user interaction data, such as viewing patterns, to further refine the recommendation system.

Case Study III: Merging the TMDb and MovieLens Datasets In this case study, we investigate the performance of the proposed recommendation system when trained on a merged dataset that combines the TMDb 5000 and MovieLens datasets. By merging these two datasets, we aim to assess the model's generalizability, robustness, and ability to handle diverse data sources. This approach also tests the scalability of the model when exposed to a broader range of user preferences and movie metadata.

Performance Metrics

We evaluated the merged dataset model using root mean squared error (RMSE), mean squared error (MSE), and mean absolute error (MAE). These metrics provide insights into the accuracy and consistency of the proposed system when exposed to a more diverse dataset. The performance of the merged dataset model was compared with baseline models, including collaborative filtering (CF), content-based

Table 5 Performance comparison on the merged TMDb and MovieLens datasets

Model	RMSE	MSE	MAE
Collaborative filtering (CF)	0.927	0.849	0.764
Content-based filtering (CBF)	0.911	0.832	0.752
Hybrid CF-CBF	0.898	0.819	0.739
Proposed NCF + sentiment analysis	**0.865**	**0.798**	**0.705**

filtering (CBF), and a hybrid CF-CBF approach. Table 5 summarizes the results.

As presented in Table 5, the proposed NCF with sentiment analysis model continues to outperform baseline models when trained on the merged dataset. The model achieves the lowest RMSE (0.865), MSE (0.798), and MAE (0.705), showcasing its effectiveness in handling a more extensive and diverse data source. This improvement in metrics underscores the robustness of the system, which can efficiently adapt to varying data structures and content features.

Impact of Data Diversity and Sentiment Analysis

The integration of sentiment analysis proves even more effective in a diverse dataset scenario. The merged dataset brings together a variety of user preferences and movie attributes, which allows the model to leverage sentiment data more comprehensively, providing deeper insights into user interests. Traditional CF and CBF methods, while functional, struggled with the increased complexity of the merged dataset, as reflected in their higher error rates. In contrast, the proposed model utilizes sentiment cues to navigate this complexity, yielding more relevant and personalized recommendations.

Scalability and Adaptability to Merged Datasets

Testing the model on a merged dataset emphasizes its scalability and adaptability. The NCF architecture, combined with sentiment analysis, enables the model to uncover latent relationships even when faced with sparse, high-dimensional data from two distinct datasets. This adaptability is crucial for real-world applications, where recommendation engines may need to integrate data from various sources. The proposed model demonstrated consistent performance, showing resilience in accommodating different user and movie profiles from both TMDb and MovieLens datasets.

Discussion

The results from Case Study III reveal that merging the TMDb and MovieLens datasets further enhances the model's ability to generalize across diverse content and user preferences. The proposed NCF with sentiment analysis framework handles the increased complexity efficiently, surpassing baseline methods in recommendation accuracy and personalization. This capability to work with heterogeneous data highlights the model's potential for real-world applications requiring robust and adaptable recommendation systems. Future work could explore merging additional datasets to further evaluate the model's scalability, as well as incorporating advanced sentiment features, such as sentiment intensity, to refine recommendation quality.

Conclusions and Future Work

In this paper, we proposed a movie recommendation system that integrates neural collaborative filtering (NCF) with sentiment analysis to improve recommendation accuracy, personalization, and scalability. Through experiments conducted on the TMDb 5000, MovieLens, and a merged dataset, our system demonstrated significant improvements over traditional collaborative filtering (CF) and content-based filtering (CBF) methods. The use of sentiment analysis enables the model to capture the emotional nuances of user reviews, thereby refining user preferences beyond explicit metadata. Our findings show that this approach enhances recommendation quality and addresses challenges such as the cold start problem, data sparsity, and repetitive suggestions.

The proposed model effectively utilizes NCF's deep learning capabilities to uncover latent relationships within sparse data, making it well-suited for large-scale, diverse datasets. The model achieved the lowest RMSE, MSE, and MAE scores across all datasets, illustrating its robustness and adaptability. Additionally, our experiments confirm that merging datasets with varied user demographics and content types improves the model's generalizability, which is crucial for real-world applications requiring adaptable and robust recommendation systems.

The system's deep learning framework enables it to uncover latent patterns in user behavior, leading to more relevant and diverse recommendations. These performance gains are achieved while maintaining the system's scalability and adaptability to large, dynamic datasets. This makes the method well-suited for real-time, personalized content delivery in the digital entertainment landscape.

The results validate the effectiveness of our proposed approach and demonstrate its potential for improving content personalization and user satisfaction in large-scale recommendation environments. Our findings suggest that integrating sentiment analysis with NCF can address key limitations of traditional recommendation systems, providing a more comprehensive and accurate solution to the challenges of modern digital media consumption. Future work will explore further enhancements to the model's scalability and its application to other domains in personalized content delivery.

Future Work

While the proposed system presents promising results, several avenues for future research remain. One potential direction is to incorporate additional sentiment features, such as sentiment intensity, sentiment trends over time, or review length, which could provide more detailed insights into user preferences and further refine recommendation accuracy. Additionally, exploring the integration of other user interaction data, such as viewing time, frequency, or engagement metrics, may enrich the model's understanding of user behavior.

Another promising area for future work is the use of reinforcement learning to dynamically adjust recommendations based on real-time feedback, enabling the system to adapt to changing user preferences continuously. Further, applying the model to other domains, such as music, e-commerce, or news recommendations, could extend its utility and evaluate its adaptability in different recommendation scenarios. Finally, scaling up the model to handle even larger datasets, such as integrating more diverse multimedia sources, would test the system's capacity for handling high-dimensional, heterogeneous data in practical applications.

In future research, several directions can be explored to further enhance the effectiveness and visual quality of the proposed recommendation system. One key area for improvement is the integration of advanced deep learning models such as convolutional neural networks (CNNs) and generative adversarial networks (GANs) to better capture visual features of the content (e.g., movie posters, trailers, or scene aesthetics). These models could allow the recommendation system to incorporate both visual and textual data, thereby improving the quality of recommendations based not only on user preferences and sentiment but also on the aesthetic appeal of movie visuals.

Additionally, multi-modal neural architectures could be employed to fuse different data sources—such as user reviews, ratings, movie metadata, and visual content—into a unified representation. This would provide a more holistic understanding of the movie content, leading to more accurate and visually appealing recommendations. Another promising direction is the use of attention mechanisms, which have been shown to enhance the model's ability to focus on key visual features when making predictions, potentially improving the relevance and quality of suggestions.

To further enhance visual quality, we also plan to explore transfer learning techniques, which allow the model to leverage pre-trained networks on large visual datasets (such as ImageNet) to improve the understanding of movie-related images. This approach would help in fine-tuning the model for movie recommendation tasks, especially in cases where training data is limited or where visual data is sparse.

In summary, incorporating advanced network architectures and learning techniques focused on visual content will be critical for enhancing the user experience and increasing the quality and diversity of recommendations in future work.

Overall, the proposed system provides a robust and scalable foundation for sentiment-driven recommendations, with significant potential for enhancing user experience in recommendation systems.

Author Contributions All authors contributed equally to the research and manuscript preparation.

Funding Not applicable.

Data Availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest Not applicable.

Research involving human and/or animals Not applicable.

Informed consent Not applicable.

References

1. Moldagulova A, Sulaiman RB. Using knn algorithm for classification of textual documents. 2017. p. 10.
2. Jebril I, Premkumar M, Muttashar Abdulsahib G, Ashokkumar S, Dhanasekaran S, Khalaf OI, Algburi S. Deep learning based ddos attack detection in internet of things: an optimized cnn-bilstm architecture with transfer learning and regularization techniques. Infocommun J. 2024;16(1):2–11.
3. Sharifi Z, Rezghi M, Nasiri M. New algorithm for recommender systems based on singular value decomposition method. 2013. p. 6.
4. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S. Neural collaborative filtering. 2017. p. 10.
5. Nelson M, Rajendran S, Khalaf OI, Hamam H. Deep-learning-based intelligent neonatal seizure identification using spatial and spectral gnn optimized with the aquila algorithm. AIMS Math. 2024;9(7):19645–69.
6. Elangovan VS, Devarajan R, Khalaf OI, Sharif MS, Elmedany W. Analysing an imbalanced stroke prediction dataset using machine learning techniques. Karbala Int J Mod Sci. 2024;10(2):8.
7. Rao KN, Khalaf OI, Krishnasree V, Kumar AS, Alsekait DM, Priyanka SS, Alattas AS, AbdElminaam DS. An efficient brain tumor detection and classification using pre-trained convolutional neural network models. Heliyon. 10(17).
8. Beena Bethel GN, Mallesh D, Sree Sai Raghavendar G. Movie recommender system using collaborative filtering. J Eng Sci. 2022;8.
9. Yeole Madhavi B, Rokade Monika D, Khatal Sunil S. Movie recommendation system using content based filtering. 2021. p. 16.
10. Tmdb movie metadata. https://www.kaggle.com/datasets/tmdb/. Accessed 21 Feb 2025.
11. Movie lens metadata. https://grouplens.org/datasets/movielens/.

12. Sulaiman M, Khalaf OI, Khan NA, Alshammari FS, Hamam H. Mathematical modeling and machine learning-based optimization for enhancing biofiltration efficiency of volatile organic compounds. Sci Rep. 2024;14(1):16908.
13. Kumar V, Subba B. A tfidfvectorizer and svm based sentiment analysis framework for text data corpus. In: National conference on communications (NCC). 2020. p. 9.
14. Chai T, Draxler RR. Root mean square error (rmse) or mean absolute error (mae). 2014. p. 4.
15. Kundan T, Kundan Singh R. Movies recommender system using machine learning algorithm. 2023. p. 9.
16. Zhubatkhan AY, Buribayev ZA, Aubakirov SS, Dilmagambetova MD, Ryskulbek SA. Comparison models of machine learning for movie recommendation systems. 2021. p. 6.
17. Sridevi S. Celeste Murnal, Implementation of movie recommendation system using machine learning. 2020. p. 7.
18. De Campos LM, Fernández-Luna JM, Huete JF, Rueda-Morales MA. Combining content-based and collaborative recommendations: a hybrid approach based on Bayesian networks. 2010. p. 15.
19. Kumar M, Yadav DK, Singh A, Gupta VK. A movie recommender system: Movrec. 2015. p. 6.
20. Vilakone P, Sengmany D, Xaypanya K, Hwang F. An efficient movie recommendation algorithm based on improved k-clique. 2018. p. 15.
21. Pavitha N, Pungliya V, Raut A, Kulkarni A. Movie recommendation and sentiment analysis using machine learning. 2022. p. 6.
22. Vishwanathan SVM, Narasimha Murty M. Ssvm: a simple svm algorithm. 2002. p. 3.
23. Behera G, Nain N. Collaborative filtering with temporal features for movie recommendation system. 2023. p. 6.
24. Zhang Y, Zhang L, et al. Movie recommendation algorithm based on sentiment analysis and lda. 2021. p. 8.
25. Bhowmick H, Chatterjee A, Sen J. Design and implementation of a movie recommendation system prototype. 2018. p. 6.
26. Benesty J, Chen J, Huang Y, Cohen I. Pearson correlation coefficient. SpringerLink; 1970. p. 8.
27. Pawar S, Patne P, Ratanghayra P, Dadhich S, Jaswal S. Movies recommendation system using cosine similarity. Int J Innov Sci Res Technol. 2022. p. 5.
28. Gaikwad A, Kudalkar S, Jethawa A. Movie recommendation system using machine learning. Int Res J Eng Technol. 2021. p. 6.
29. David R. Cheriton School of Computer Science University of Waterloo Waterloo. The use of machine learning algorithms in recommender systems: a systematic review. 2018. p. 16.
30. Gorli R, Ram BA. Mrml-movie recommendation model with machine learning techniques. Int J Sci Res. 2023. p. 5.
31. Sze V, Chen Y-H, Yang T-J, Emer JS. Efficient processing of deep neural networks: a tutorial and survey. 2017. p. 20.
32. Prasanna P, Teja AS, Sree ND, Sreechandana S, Akash R, Gaurav S. Movie recommendation using machine learning. J Basic Sci. 2022:4.
33. Pradhan R, Swami AC, Saxena A, Rajpoot V. A study on movie recommendations using collaborative filtering. 2018. p. 9.
34. Yao Z. Review of movie recommender systems based on deep learning. 2023. p. 4.
35. Mirza Ilhami S. Film recommendation systems using matrix factorization and collaborative filtering. 2014. p. 6.
36. Mirza Ilhami S. Film recommendation systems using matrix factorization and collaborative filtering. 2014. p. 6.
37. Ganesh D, Bhansali Y. Movie recommendation system using content based filtering. Int J Creat Res Thoughts. 2022:7.
38. Madadipouya K. A location-based movie recommender system using collaborative filtering. Int J Found Comput Sci Technol. 2015;7.
39. Pandey A, Sharan A, Mishra V, Richa G, Charu T. Movie recommendation system using machine learning. Int J Innov Res Technol. 2021;4.
40. Dridi A, Gaber MM, Azad RM, Bhogal J. Vec2dynamics: a temporal word embedding approach to exploring the dynamics of scientific keywords–machine learning as a case study. Big Data Cogn Comput. 2022:12.
41. Word embeddings in natural language processing | nlp. https://medium.com/@zeeshanmulla/word-embeddings-in-natural-language-processing-nlp-5be7d6fb1d73. Accessed 21 Feb 2025.
42. Zaman F. Automated liver segmentation from mr-images using neural networks. B. Thesis for: Master's in Biomedical Engineering. 2019. p. 12.
43. Université M-A B S, Baazizi M-A, Université S, Paris-Dauphine D C U, Colazzo D, Paris-Dauphine U, Pisa G G U di, Ghelli G, Pisa U di, Basilicata C S U della, Sartiani C, Basilicata U della, Amsterdam C, V U, Leiden C, U, Epfl, Maryland, U. of, Mit, Metrics O M A, Schemas and types for json data. In: Proceedings of the 2019 international conference on management of data. ACM conferences. 2019. p. 10.
44. Alaya MZ, Bussy S, Gaiffas S, Guilloux A. Binarsity: a penalization for one-hot encoded features in linear supervised learning. J Mach Learn Res. 2019;10.
45. Elsafoury F, Wilson SR, Ramzan Naeem. A comparative study on word embeddings in social nlp. 2022. p. 10.
46. Rahul Pradhan A, Chandra SA, Vikram R. A study on movie recommendations using collaborative filtering. 2021. p. 9.
47. Md Riyad H, Douglas T. Machine learning model optimization with hyper parameter tuning approach. Glob J Comput Sci Technol D Neural Artif Intell. 2021. p. 8.
48. Al-Shamri YH. User profiling approaches for demographic recommender systems. Sciencedirect.
49. Tej MK. Movie recommendation system using machine learning. Int J Res Publ Rev. 2023:4.
50. Aljunid MF, Huchaiah MD. Multi-model deep learning approach for collaborative filtering recommendation system. The Institution of Engineering and Technology. 2020. p. 8.
51. Tanti HK, Ghosh P, Saha T. An ai based movie recommendation system. Int Res J Mod Eng Technol Sci. 2023;7.

Authors and Affiliations

Qusay Bsoul[1] · Firas Zawaideh[2] · Basma S. Alqadi[3] · Latifa Abdullah Almusfar[4] · Osamah Ibrahim Khalaf[5] · Ahmed Saleh Alattas[6] · Muath Alali[7] · Diaa Salama AbdElminaam[8,9]

✉ Diaa Salama AbdElminaam
diaa.salama@miuegypt.edu.eg

Qusay Bsoul
q.bsoul@aau.edu.jo

Firas Zawaideh
F.zawaideh@jadara.edu.jo

Basma S. Alqadi
bsalqadi@imamu.edu.sa

Latifa Abdullah Almusfar
laalmusfar@iau.edu.sa

Osamah Ibrahim Khalaf
usama81818@nahrainuniv.edu.iq

Ahmed Saleh Alattas
aalatas@kau.edu.sa

Muath Alali
M_alali@asu.edu.jo

1 Cybersecurity Department, College of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan

2 Cybersecurity Department, Faculty of Science and Information Technology, Jadara University, Irbid, Jordan

3 Computer Science Department, College of Computer and Information Sciences, Al Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia

4 College of Business Administration, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, 31441 Dammam, Saudi Arabia

5 Department of Solar, Al-Nahrain Research Center for Renewable Energy, Al-Nahrain University, Jadriya, Baghdad, Iraq

6 Information Science Department, Faculty of Arts and Humanities, King Abdulaziz University, Jeddah, Saudi Arabia

7 Data Science and Artificial Intelligence, Faculty of Information Technology, Applied Science Private University, Amman, Jordan

8 Jadara Research Center, Jadara University, Irbid 21110, Jordan

9 MEU Research Unit, Middle East University, Amman 11831, Jordan