Sarhan *et al.*
Journal of Electrical Systems and Inf Technol (2024) 11:53
https://doi.org/10.1186/s43067-024-00177-7

Journal of Electrical Systems and Information Technology

RESEARCH **Open Access**

Integrating machine learning and sentiment analysis in movie recommendation systems

Amany M. Sarhan[1*], Hager Ayman[2], Mariam Wagdi[2], Bassant Ali[2], Aliaa Adel[2] and Rahf Osama[2]

*Correspondence: amany_sarhan@f-eng.tanta.edu.eg

[1] Department of Computer and Control Engineering, Tanta University, Tanta, Egypt
[2] Department of Artificial Intelligence, Faculty of Artificial Intelligence, Delta University, Gamassa, Egypt

Abstract

The fast growth of the film business, along with an ever-increasing number of movie options, has highlighted the need for better recommendation algorithms. This study investigates the application of sentiment analysis in a movie recommendation system with the goal of improving the user experience. The importance of this sector stems from its ability to bridge the gap between user interests and the vast number of cinematic products, addressing individual emotional states and preferences. Researchers choose to generate movie recommendations based on the sentiments conveyed by viewers' reviews of the movies. Sentiment-based movie recommendation system research employs techniques such as natural language processing and hybrid models with the goal of increasing user satisfaction. To this purpose, the integration of advanced machine learning algorithms such as cosine similarity, support vector machine, and Naive Bayes improves recommendation systems with sentiment analysis. Cosine similarity improves movie recommendations by recognizing minor user preferences, while support vector machines and Naive Bayes enhance sentiment analysis by offering a nuanced interpretation of textual attitudes. Through trials, the proposed system employs two public datasets for sentiment analysis, namely the TMDB5k dataset and the Reviews dataset, and makes predictions (positive, negative, or neutral) based on the content of the review through conducting sentiment analysis on the text using the Viscous Accretion Disk Evolution Resource (VADER) approach. The findings, based on users' feedback, are more accurate and informative regarding movie quality, where SVM accuracy is 99.28% and Naïve Bayes accuracy is 96.60% when used with VADER sentiment analysis.

Keywords: Recommendation systems, Movie recommendation, Sentiment analysis, Cosine similarity, Support vector machine, Machine learning

Introduction

Recommender systems enable consumers to reach intelligent choices by collecting statistics on their preferences in a range of domains, including products and services. These systems allow users to select the best option from a variety of possibilities. One of the most important applications of recommender systems, particularly for visitors and travelers, is the selection of a restaurant from a large number of unknown options. However, obtaining user preferences is a difficult problem with these systems. Traditional methods, such as questionnaires or user-provided rates to restaurants, are incapable of dynamically extracting their preferences. Online comments on websites and social networks are now seen to be a rich source of implicit information. In this sense, the user's dietary preferences may be deduced from the processing of these remarks and the analysis of the data [1–4].

Previous sentiment analysis-based recommendation systems examine people's opinions based on subject's criteria (such as content idea or media quality in movies). The most crucial consideration when selecting a movie is the quality of the idea, while others can see the quality of actors is better. Previous recommender systems retrieve movies names using broad approaches such as term-frequency (TF) or bag of words. Repetitive patterns, on the one hand, show the user's preferences and, on the other hand, represent elements in which the user is uninterested, and which are retrieved solely via repetition. As a result, suggestion accuracy is low in these systems.

People are turning more and more to the Internet for movie viewing instead of traditional televisions, owing to the advancement of Internet technology. Online movies are getting popular, and consumers are seeking better tools to propose appropriate movies to view. Figure 1 shows the movies industry in both advertising section and box office revenue in 2020 (reported by SNL Kagan [5]), where the y-axis represents the revenue in Billion Dollars.

Figure 2 shows movies investment prediction in USA up to 2030 [6] where the y-axis represents the predicted movies investment in Billion Dollars. Both figures indicate the importance of using recommender system to guide the viewers to similar movies to increase the companies' revenue.

The purpose of a recommender system is to deliver individualized information to users, which enhances the user experience and boosts corporate profit. It is now a very popular issue in both industry and academia as one of the usual applications of machine learning driven by the real world [7]. In many online E-commerce platforms, recommender systems have become a critical tool for finding users' hidden interests and preferences, creating pleasurable user experiences, and driving incremental income. Modern recommender systems have gained enormous success and unsurpassed performance in recent years, thanks to very expressive machine learning algorithms.

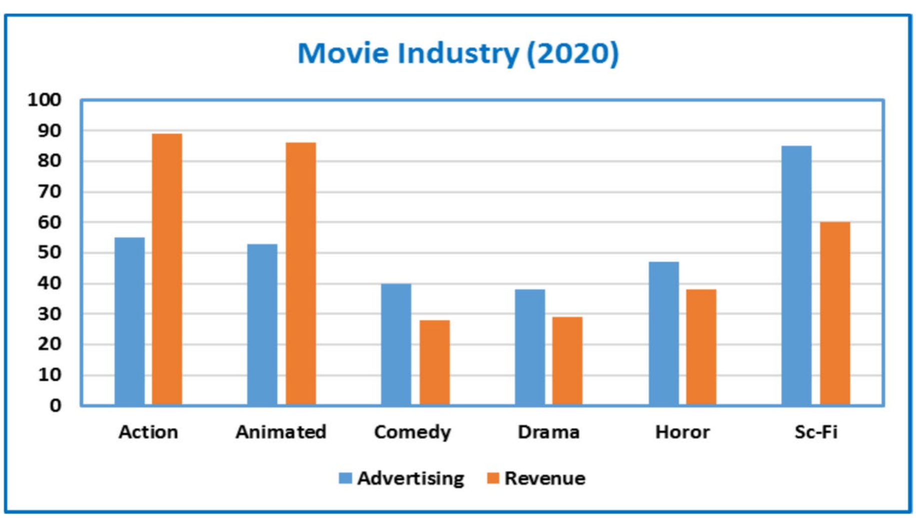

Fig. 1 Movie industry statistic in 2020 (By: SNL Kagan, 2020)

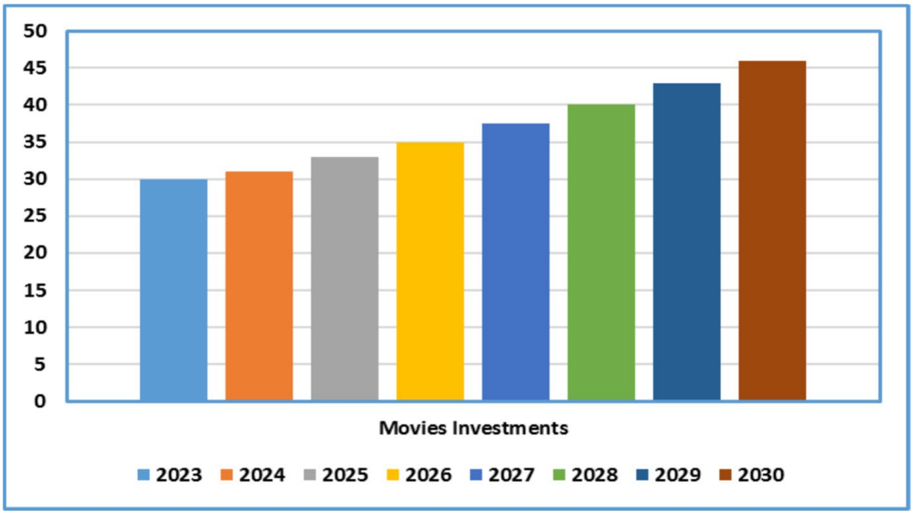

Fig. 2 Movies investment prediction up to 2030 in USA

Some studies have explored the effectiveness of collaborative filtering in movie recommendation systems, emphasizing the importance of user-item interactions [8, 9]. Content-based filtering has been used to recommend movies based on features like genre, director, or actors [10]. Sentiment analysis has been employed to analyze movie reviews and understand audience opinions [11]. Hybrid models have been proposed to combine the strengths of both collaborative and content-based filtering for improved recommendation accuracy [12, 13].

Sentiment analysis has evolved across distinct types starting at polarity-based analysis, where the early models were utilizing techniques like bag-of-words and machine learning classifiers, classified text into positive, negative, or neutral sentiments. Aspect-based analysis models were developed to extract nuanced opinions by focusing on specific aspects or features within a sentence or document, beyond overall sentiment. The researchers are using modern approaches such as natural language processing to develop these concepts. They are also using advanced machine learning approaches to increase the accuracy and value of movie suggestions for users, such as cosine similarity, support vector machines, and Naive Bayes [14].

In this article, we will present a movie recommender system that uses sentiment analysis to comprehend more about what people like to offer more relevant recommendations. To accomplish this, we enhance the extraction of user preferences using sentiment analysis. The motivation behind creating a sentiment-based movie recommendation system lies in enhancing user satisfaction and engagement. By analyzing sentiments in user reviews, the system aims to capture not only explicit preferences, but also the emotional nuances users express toward movies. This personalized approach enables the system to recommend films that resonate with users on an emotional level, fostering a deeper connection with the content. Such a system not only optimizes user experience by delivering more relevant recommendations, but also contributes to a more enjoyable and tailored entertainment journey, ultimately increasing user retention and satisfaction in the ever-expanding landscape of digital content.

The main contributions of this research are:

1. This research pioneers a unique framework amalgamating sentiment analysis and a hybrid recommendation system for movies.
2. A model predicts movie ratings by scrutinizing sentiment in accompanying comments.
3. We have proposed a new way of calculating the comprehensive sentiment of a movie.
4. The paper also unveils a content-based filtering system using cosine similarity for recommendation, incorporating support vector machines and Naive Bayes for sentiment analysis.

Related work

In this section, we aim to discuss the basics of a recommender system and its types followed by the goal of sentiment analysis and how can a recommender benefit from sentiment analysis. The previous work in movie recommendation systems is then analyzed and listed and the shortcoming of the previous work to build upon in this paper.

Recommender system

Recommender systems are vital for enhancing user experience, boosting engagement, and increasing revenue by offering personalized suggestions. They come in various types, including collaborative filtering, recommend items based on the preferences and behaviors of users with similar tastes [9, 15–18]. It can be user-based or item-based collaborative filtering, and content-based filtering recommends items by analyzing the content and characteristics of items, matching them with a user's preferences derived from their past behavior, and hybrid systems that combine approaches for improved accuracy [2, 19, 20]. Applications span diverse sectors such as e-commerce, streaming services, social media, news aggregation, job platforms, and travel, where recommender systems help users discover tailored content and make informed choices. The significance lies in reducing information overload, providing relevant recommendations, and adapting to evolving user preferences. Continuous advancements in machine learning contribute to the ongoing refinement and effectiveness of recommender systems [3, 9]. Recommender systems are classified into three major categories as shown in Fig. 3. We will discuss each of them in some details.

Collaborative filtering recommender systems

Collaborative filtering (CF) is a common method of recommending things to others. It operates on the premise that if a customer liked similar items to someone else in the past, he will probably like similar things in the future as well. CF systems provide suggestions based on persons with similar likes rather than only looking at the specifics of what the customer enjoyed [17, 21]. As chosen in Fig. 4, it creates a matrix of items that each customer likes, then considers what others who are similar to a specific customer have previously admired and proposes items to the customer based on what these similar people liked. There are two main ways CF does this; memory-based algorithms, where the algorithm looks at everyone's preferences in the whole chart to find similar users or

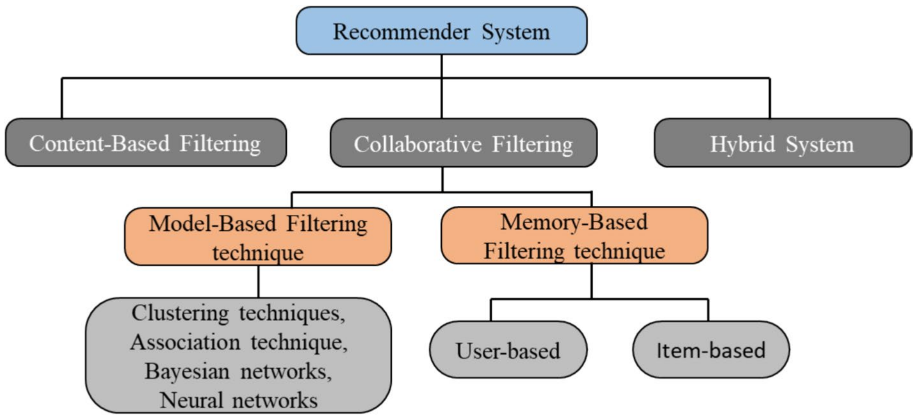

Fig. 3 Classification of recommender systems

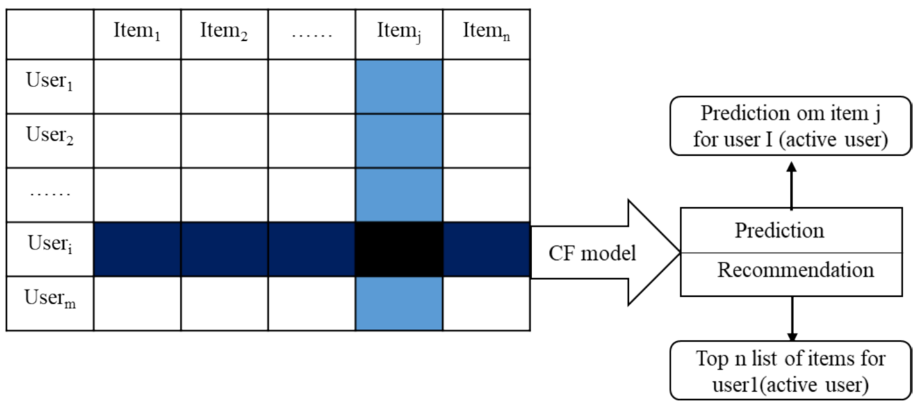

Fig. 4 An overview of the CF process

items. Once they find these similar ones, they use their past likes to suggest things to you. It could be based on what people like you enjoyed before (user-based) or what similar items are there to the one you're looking at (item-based) [1, 15].

Model-based algorithms use more advanced methods like machine learning to build a system that understands patterns in the data. This system then predicts what you might like based on those patterns. Examples include using Bayesian models, clustering models, decision trees, and other techniques.

Content-based recommender systems

Content-based recommender systems suggest entities based on the details of those entities and what users like, as shown in Fig. 5. They aim to recommend items related to what a user liked in previous sessions. For example, if a user likes a website with words like "stack," "queue," and "sorting," the system would suggest pages about data structures and algorithms. These systems work well when suggesting brand-new items. Even if there are no ratings for the new item yet, the system can use information about the item to suggest it to the right users. For instance, if a user enjoyed movies like "The Terminator" and "The Matrix," they might get a suggestion for a new science fiction movie [21].

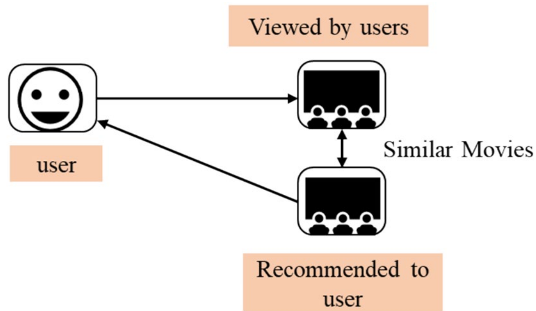

Fig. 5 Content-based recommender's process

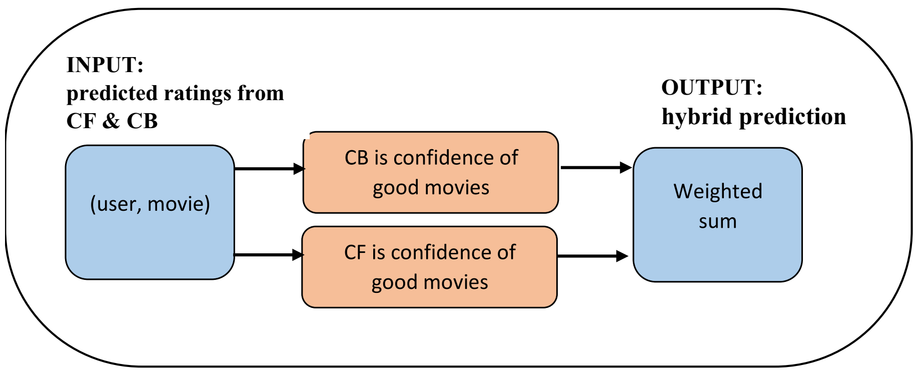

Fig. 6 Hybrid recommender process

Hybrid recommender systems

Collaborative filtering (CF) and content-based (CB) recommenders each have their good and not-so-good sides. But hybrid recommender systems do something different. They mix CF and content-based methods to gather the benefits of both approaches and escape the shortcomings [10, 21–23]. There are different ways to mix these methods, like:

- *Weighted*: Combining scores from different methods to give one recommendation.
- *Switching*: Picking recommendations from one method or the other based on what's happening.
- *Mixed*: Showing recommendations from both methods simultaneously.
- *Cascade*: Refining recommendations from one method with the help of the other.
- *Feature combination*: Mixing features from both methods and using them in one process.
- *Feature augmentation*: Using the recommendation from one method as input for the other.

In this sense, hybrid recommender systems blend collaborative filtering and content-based approaches in clever ways to give better and more diverse recommendations, as shown in Fig. 6.

Sentiment analysis

Sentiment analysis is a branch of natural language processing (NLP) which is essential for businesses to understand and respond to customer opinions, market trends, and public sentiment. Its types include polarity-based analysis, emotion-based analysis, and aspect-based analysis, providing nuanced insights. Applications are found in social media monitoring, customer feedback analysis to political and financial analysis. By analyzing sentiments on platforms like Twitter and customer reviews, businesses can improve products, enhance customer experiences, and make informed decisions. Sentiment analysis is particularly valuable in market research, allowing organizations to gain insights into consumer preferences. Its role in political analysis and financial markets helps anticipate trends and reactions. Overall, sentiment analysis is a versatile tool that aids decision-makers in extracting valuable insights from textual data across diverse industries [24, 25]. To apply sentiment analysis to your business, it is crucial to understand its various types as shown in Fig. 7.

Fine-grained sentiment analysis decides if opinions are positive or negative. It can be really simple, like thumbs up or thumbs down, or more detailed, like rating something from super positive to super negative. This is so close to our movie review. In emotion detection sentiment analysis, we figure out the feelings in the text, like happiness or anger. It uses a mix of word dictionaries and computer learning approaches. Aspect-based sentiment analysis deeply explores specific parts of a product or service, like how people feel about the camera quality of a phone. Finally, intent-based sentiment analysis focuses on understanding the goal behind a message, especially in customer support. It helps to make things smoother and faster. Knowing these types helps businesses use sentiment analysis to understand if people are happy or not, what emotions they are showing, what they like or dislike about products, and why they are reaching out for support.

A study in [26] describes a unique, context-aware, deep learning-powered Persian sentiment analysis method. The suggested deep learning-driven automated feature engineering technique categorizes Persian movie reviews as favorable or negative. Two deep learning methods, convolutional neural networks (CNN) and long-short-term memory (LSTM), were used and compared with previously suggested manual-feature-engineering-driven and SVM-based methodology. The simulation results show that LSTM outperformed the multilayer perceptron (MLP), autoencoder, support vector machine (SVM), logistic regression, and CNN algorithms.

Recommender with sentiment analysis

Integrating sentiment analysis into recommender systems brings forth a multifaceted dimension to user interaction. The system's ability to discern emotional responses from

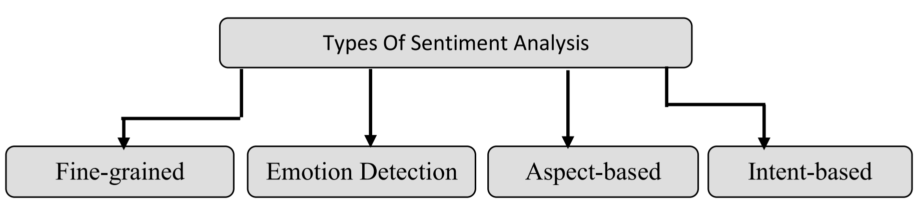

Fig. 7 Types of sentiment analysis

reviews enables it to adapt to evolving user sentiments over time. As users engage with the recommended content, the recommender system continues to refine its understanding of their emotional preferences, creating a dynamic and evolving user profile.

Furthermore, the incorporation of advanced sentiment analysis techniques such as machine learning approaches contributes to the system's accuracy in deciphering nuanced sentiments expressed in user reviews. The utilization of these sophisticated algorithms allows the recommender system to not only identify positive or negative sentiments, but also grasp subtle variations within the emotional spectrum, leading to more precise and tailored recommendations.

In [27], an approach to recommendation that combines sentiment analysis and collaborative filtering was offered. The approach is implemented in the adaptive recommender system architecture, which comprises feature extraction techniques and deep learning-based sentiment analysis. The findings of an empirical investigation using two prominent datasets reveal that integrating deep learning-based sentiment analysis and collaborative filtering approaches considerably improves the effectiveness of the recommender system. The architecture is divided into two components, one for producing sentiment models and the other for providing suggestions to a certain user based on the models previously developed. The data from the reviews were preprocessed before being utilized to run and train a sentiment-based hybrid deep learning-BASED model. For rating the yielded prediction, a user-based collaborative filtering approach is integrated with sentiment-based algorithms.

In [28], a recommendation model was proposed that uses a hybrid recommendation model (HRM) as well as a hybrid sentiment analysis to improve the accuracy and correctness of any recommender system. Using an HRM, we first build a preliminary suggestion list in the suggested technique. The HRM based on sentiment analysis is utilized to build the final recommendation list, which beats the standard models across several assessment criteria. Because each of the collaborative and content-based filtering approaches may correct the weaknesses of the other, the recommender system is accurate and steady. On the one hand, collaborative filtering can prevent the content-based filtering approach from losing its customization; on the other hand, content-based filtering can compensate for the collaborative filtering method's scalability flaw.

Previous work in movie recommendation systems

To solve the limitation mentioned above, this research discusses the advantages and disadvantages of recommendation algorithms based on content-based and collaborative algorithms and gives optimization suggestions simultaneously. Furthermore, we synthesized the ideas of the above traditional recommendation algorithms, constructed a recommendation model of movies which is based on the CNN deep learning algorithm, and applied it to the simple movie system which we designed [29]. The model adopts the calculation method of cosine similarity, tests it through the MovieLens/ml-1 m dataset to calculate its loss and accuracy, and provides the results in the end. Additionally, a method for addressing the cold-start problem of users, products, and systems is described. Despite the model's fast training, its accuracy in predicting user ratings is low, which may be due to the loss function's irrational design.

In [30], a movie recommender system is presented as an autonomous machine learning-based technology that selects movies from large movie databases such as Netflix and Amazon depending on user preferences. Partitional weighted co-clustering for movie recommender system is the primary subject of this work. The major goal of this study piece is to fine-tune the parameters of user and movie neighborhoods by changing the values for the row clusters number and column clusters number co-clustering parameters. Test results acquired from the MovieLens database reveal that the suggested technique can provide more accurate tailored suggestions for the movie by the order of 7.91% when compared to existing methods.

In [31], a collaborative filtering approach for movie suggestions is introduced that includes temporal effects to address the temporal and dynamic impacts of user-item interaction. To demonstrate the importance of the proposed method, the approach was tested on MovieLens dataset, where the results show that the method outperforms a state-of-the-art model by 1.35% and 1.28% on ML-100 K and 1 M datasets, respectively.

The researchers in [32] employed sentiment analysis of YouTube trailer comments to estimate probable movie ratings. Following that, they generated tailored lists of future movies depending on the user's preference. Combined with existing movie data, a recommendation system was built for new films. This study utilizes publicly accessible data from The Movie Database (TMDb) as well as a dataset of new movies developed by randomly picking one hundred movies published on Netflix between 2020 and 2021. The method was tested on a dataset of Netflix movies and shown to successfully predict movie ratings and recommend movies.

The research in [11] focused on improving movie recommendation systems by combining sentiment analysis, with the goal of detecting and extracting personal information from textual data. Traditional fully connected neural networks (FNNs) are often employed for sentiment analysis, but their ability to capture long-term relationships is limited. Long short-term memory (LSTM) models, conversely, are recognized for their capacity to absorb context and meaning in text. To give reliable and timely movie choices based on user preferences, reviews, and emotions, the proposed movie recommendation system blends sentiment analysis, collaborative filtering, and content-based approaches. They used the MovieLens dataset, and their algorithm outperformed other previous work.

In [33], sentiment analysis has been presented as a more accurate approach of extracting emotional information from films. They offer a movie recommendation system in this study that integrates sentiment analysis, collaborative filtering, and content-based approaches. Their approach is designed to provide mobile users accurate and fast suggestions based on their preferences, reviews, and emotions. For this investigation, the MovieLens dataset was used. To prepare the dataset, they cleaned and preprocessed it, then eliminated any duplicate ratings, as well as any movies or individuals with a low number of ratings. In addition, feature engineering was used to extract useful characteristics from the raw data, such as movie genres, user demographics, and movie release dates.

In [34], the system employs content-based filtering, utilizing cosine similarity, to suggest ten movie recommendations based on user preferences. It also provides detailed information about the searched movie, including ratings, release date, cast, and genres.

Additionally, the system performs sentiment analysis on movie reviews, categorizing them as 'Good' or 'Bad.' The methodology involves data collection, cleaning, API integration, sentiment analysis using NLTK, and recommendation using cosine similarity. The system is implemented in Python and achieves a 98.77% accuracy in sentiment analysis.

In [35], a K-mean clustering technique is used to cluster users, and then deep neural network (DNN) is used to assign new users to the desired cluster to select the categories that are most suitable for users. A collaborative recommender system (CRS) is presented relying on the hybrid similarity criteria, which calculates similarities based on a threshold between the new user and the users in the selected category. They employed a modified Friendlink algorithm to determine the similarity of those connected via the link.

To improve recommendation accuracy, the model in [14] incorporates both cosine similarity and sentiment analysis. Cosine similarity, a common measure for item similarity, is used to assess the similarity of two movies, helping the finding of material that closely matches a user's preferences. Sentiment analysis is also used to examine the emotional tone of movie reviews, which aids in assessing the overall positivity or negativity of a review and hence influences the film's overall rating. This system performs sentiment analysis on the reviews of the movie chosen using machine learning in order to enhance the user experience. Two machine learning algorithms are used: Naïve Bayes (NB) classifier and support vector machine (SVM) classifier. Machine learning's automation of sentiment analysis leads to the system's capacity to autonomously label reviews as positive or negative based on learned data. This paper focuses on the creation of a content-based movie recommendation system, offering a model that grows over time. When different systems using content-based approaches are compared, the findings become increasingly interesting. Tmdb_5000 movies dataset was used combined with reviews and credits to train the models.

In [36], a content-based recommendation system for movies is provided that is augmented by a rating algorithm that takes into account the visual similarity of the material as well as the sentiment of user evaluations. A pre-trained visual geometry group (VGG) network is employed for feature extraction from key frames of the movie trailer for visual similarity. Following that, the similarity is computed based on the Euclidean distance between the distribution of frames in the test and reference movies. A publicly accessible IMDb dataset is used for sentiment analysis and choose the model with the greatest combination of accuracy and F1-score before computing the percentage of favorable reviews in the movie.

In [37], a movie recommender system was proposed utilizing a hybrid approach, combining content-based and collaborative filtering methods. It leverages continuous analysis of user behavior for real-time adaptation. The system employs a gradient descent algorithm on the MovieLens dataset, extracting latent features to predict movie ratings. Results indicate that the item-by-item-based approach outperforms, taking 170 s to build the model and 3 s to predict 100,021 ratings, achieving efficient and accurate recommendations.

The work in [16] focuses on implementing a movie recommendation system that employs several machine learning techniques, specifically comparing collaborative filtering and content filtering. They use the TMDB dataset, including movie details, and apply algorithms like alternating least square (ALS), K-nearest neighbors (KNN), single-valued decomposition) (SVD), co-clustering, and cosine similarity. Results indicate that the content-based model with cosine similarity outperforms others, achieving lower root-mean-squared error (RMSE) and mean squared error (MSE) values. The paper contributes by proposing a comprehensive approach to movie recommendation, emphasizing the effectiveness of content filtering with cosine similarity.

In [38], five machine learning classifiers were applied to preprocessed data comprising feature vectors to categorize the movie reviews data on IMDb movies dataset: multinomial Naïve Bayes, SVM, decision tree, Bernoulli Nave Bayes, and maximum entropy. The proposed system used weighted score fusion to improve the recommendations. They measured correlation measures between sentiment and movie ratings.

In [39], when compared to content-based filtering, KNN algorithms and collaborative filtering are employed to increase the accuracy of the results. This method is based on cosine similarity utilizing k-nearest neighbor with the assistance of a collaborative filtering strategy, overcoming the limitations of content-based filtering. Although Euclidean distance is preferable, cosine similarity is employed since the precision of cosine angle and movie equidistance are almost same. They used MovieLens as a dataset.

In [40], the research work is dedicated to creating a movie recommendation system that utilizes both cosine similarity and sentiment analysis. Cosine similarity is a measure of how similar two objects are to one another. An assessment of the emotions stated in a movie review may decide if the review is outstanding or unfavorable, and hence the overall rating for the film. However, their work is only a notion, with no experiments or outcomes.

In [41], a similarity method is described known as user profile correlation-based similarity (UPCSim), which allows additional user behavioral data to impact the recommendation's accuracy. It computes similarity weights based on the user's rating and behavior value, and it classifies the user's preferences using the K-nearest neighbors method. While this method reduces the mean absolute error (1.64%) and the root-mean-square error (1.4%). It takes longer time to compute. A list of previous work in the filed is given in the review given in [47].

To conclude, the previous studies are summarized in Table 1, where we can see that the need to improve the quality of the recommender systems employed in the field of Movies, specifically using TMDB (5 K) dataset. The idea of sentiment analysis is not utilized in an efficient way; therefore, we introduce the current work to enable the integration of machine learning with sentiment analysis.

Proposed recommendation system with sentiment analysis

Constructing a sophisticated movie recommendation system demands a seamless integration of diverse elements. Singular reliance on recommendations or sentiment analysis proves inadequate in delivering a comprehensive and enriched user experience. Thus, an imperative is established for a holistic approach that effortlessly intertwines recommendation algorithms with sentiment analysis [12]. Sentiment analysis enhances the user experience, providing more accurate and timely movie suggestions compared to traditional recommendation systems. In the domain of movie suggestions, the recommendation system serves as the foundational framework, harnessing advanced algorithms to

Table 1 Pervious related methods in movie recommendation system

Paper	Dataset	Classifier	Accuracy
He [29]	Movie Lens 1 M dataset	CNN, cosine similarity	21.07%
Singh et al. [33]	MovieLens	Sentiment analysis + collaborative filtering and content-based	98.89%
Airen and Agrawal [30]	MovieLens	Partitional weighted co-clustering for + fine-tune the parameters of user and movie neighborhoods	7.91% when compared to existing methods
Behera and Nain [31]	MovieLens	Collaborative filtering + temporal and dynamic impacts of user-item interaction	Increased by 1.35% and 1.28% on ML-100 K and 1 M datasets
Pawar [34]	TMDB (5 K), IMDb API	Cosine similarity	98.77%
Pavitha et al. [14]	TMDB (5 K)	Naïve Bayes, support vector machine, conventional neural network	SVM = 98.6% NB = 97.33%
Kalkar and Chawan [16]	TMDB	KNN, SVD, ALS, co-clustering, cosine similarity	97.8%
Mehta and Kamdar [36]	IMDb	Content-based + VGG for feature selection + Euclidean distance	90%
Jamnekar and S.U. Bohra [37]	MovieLens	Gradient descent	95%
Vahidi et al. [35]	MovieLens	DNN + CRS + Friendlink	91%
Gupta et al. [39]	MovieLens	KNN + Collaborative + cosine similarity	Mean absolute error (MAE) = 0.248

scrutinize user preferences and historical data. Yet, the cruse of user satisfaction surpasses mere suggestions, encompassing an intricate understanding of the emotional facets intertwined with movie-watching.

Introducing sentiment analysis as a pivotal component adds a layer of depth to the recommendation process. By delving into the sentiments expressed in user reviews and interactions, the system acquires profound insights into the emotional resonances associated with a movie. This nuanced comprehension not only hones the recommendations, but also aligns them with the user's prevailing mood and preferences. Essentially, the amalgamation of recommendation algorithms and sentiment analysis begets a dynamic synergy. This synergy propels the recommendation system to unprecedented heights, where user satisfaction transcends content matching and extends into emotional resonance. Users are not merely presented with movies that align with their preferences but are tailored to suit their current emotional state. By augmenting the movie recommendation system with sentiment analysis, we not only furnish users with films they are likely to relish, but also factor in the emotional context, ensuring a more personalized and immersive cinematic experience. This harmonious fusion transforms the recommendation process into a journey that caters not only to the intellect, but also resonates with the heart, culminating in a genuinely holistic and user-centric movie-watching experience.

Furthermore, the incorporation of advanced sentiment analysis techniques such as support vector machines and Naive Bayes contributes to the system's accuracy in deciphering nuanced sentiments expressed in user reviews. The utilization of these sophisticated algorithms allows the recommender system to not only identify positive or negative sentiments, but also grasp subtle variations within the emotional spectrum, leading to more precise and tailored recommendations.

Additionally, the dual-layered approach involving cosine similarity, and sentiment analysis enhances the robustness of the recommender system. By combining these methodologies, the system leverages both collaborative filtering based on user preferences and sentiment-based filtering, providing a comprehensive and holistic recommendation strategy.

In conclusion, the integration of sentiment analysis into the movie recommender system not only elevates the accuracy of recommendations, but also transforms the user experience into a more emotionally intelligent and empathetic journey. This innovative approach reflects the evolving landscape of personalized content recommendation, catering to the diverse and nuanced preferences of users in the ever-expanding realm of entertainment.

Methodology

Figure 8 presents the proposed movie recommendation system general architecture. Two datasets have been used for study, one for movie recommendation (TMDB 5 K/10 K) and one for sentiment analysis (reviews.txt). Preparing the sentiment analysis dataset for training a machine learning model is very crucial. Data preprocessing is an important step before the development of any machine learning model. Before passing text into SVM, it is necessary to clean the dataset and remove any noise in the dataset.

The preprocessing for sentiment analysis requires natural language processing as follows:

- *Text cleaning*: Remove any irrelevant information, such as HTML tags, special characters, or non-alphabetic characters.

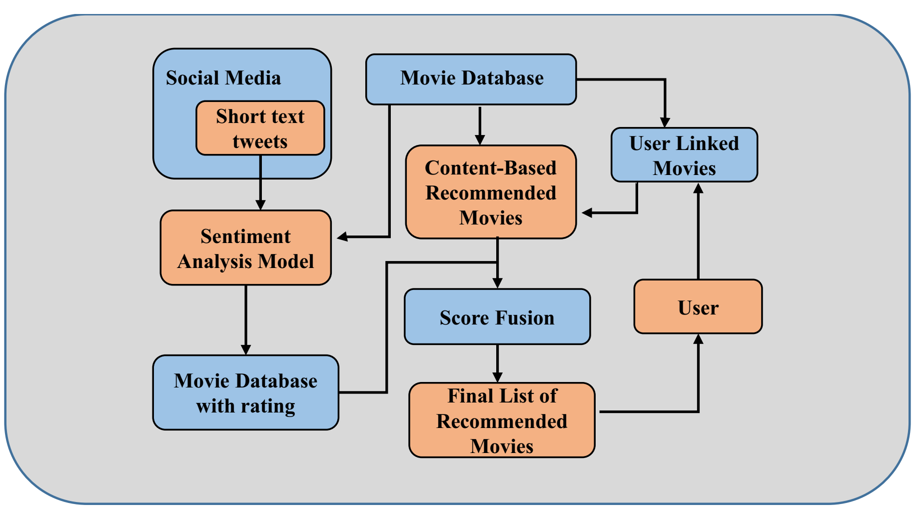

Fig. 8 Proposed movie recommendation system architecture

- *Lowercasing*: Convert the entire text to lowercase to maintain uniformity and simplify subsequent analysis.
- *Tokenization*: Break down the text into individual words or tokens. This step is crucial for further analysis as it allows you to work with individual words.
- *Stop words removal*: Remove common words (stop Words) that do not carry much meaning (e.g., "the," "and," "is"). This helps reduce the dimensionality of the data and focuses on more meaningful words.
- *Sentiment Labeling* Assign sentiment labels to the reviews (e.g., positive, negative, neutral). This step is crucial for training a sentiment analysis model.

Proposed movie recommender systems based on machine learning and sentiment analysis framework

Figure 9 depicts the proposed framework of movie recommendation system. The main component of the framework is the sentiment analysis which effectively classifies comments into positive, negative, or neutral categories. The polarity algorithm within this analyzer assigns a sentiment polarity score to a given piece of text. The scores represent the intensity of positive, neutral, and negative sentiments within the text. SVM and Naive Bayes machine learning models are employed for the learning step of the recommender system as proved in previous studies to attain high accuracy.

Preprocessing steps ensure text cleanliness and improve the model's predictive capabilities. This sentiment analysis offers reliable and accurate classification of comments, aiding in understanding sentiment patterns within the dataset. The combination of SVM

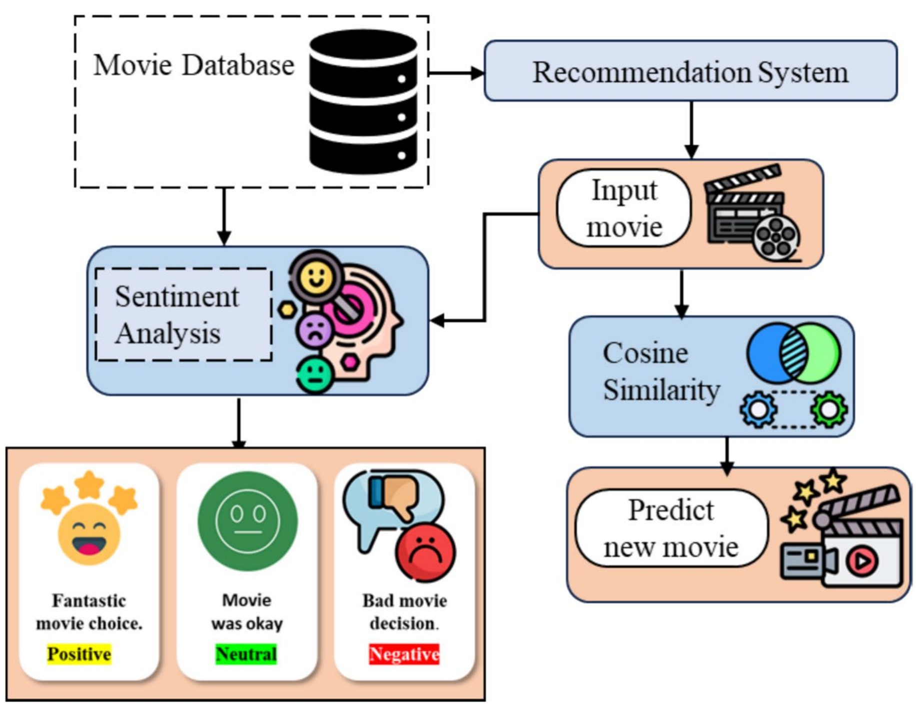

Fig. 9 Framework of movie recommendation system

and NB algorithms ensures a comprehensive analysis, contributing to informed decision-making in scenarios like reviews, customer feedback, and opinion mining.

Naive Bayes algorithm

A Bayes' theorem-based approach is applied. According to the Naive Bayes algorithm, the presence of one characteristic in a group has no influence on the availability of other features in that class. This framework is simple to build and is very useful when dealing with large datasets. Because of its simplicity, Naive Bayes is known to perform better even with the most powerful classification algorithms. Computation details of Naïve Bayes are given in Eq. (1) [14, 38].

$$P(c|x) = \frac{P(x|c).P(c)}{P(x)} \tag{1}$$

where P(c|x) is the posterior probability, $P(x|c)$ is the likelihood of x over c, $P(c)$ is the class prior probability, and $P(x)$ is the predictor prior probability. $P(c|x) = P(x_1|c).P(x_2|c)\ldots\ldots.P(x_m|c).P(c)$

Algorithm

Step 1: Input: movie recommendation system input features

Step 2: Assign training and testing dataset for movie recommendation system

Step 3: Generate Output: Classification on movie recommendation system

Step 4: Apply Function: Naïve Bayes (Input features I)

Step 5: Go over the training data.

Step 6: For each class, compute the mean and standard deviation of the predictor variables.

Step 7: In each class, use the G gauss density formula to compute the probability of p_i.

Step 8: Estimate the likelihood for each class until all predictor variables (p_1, p_2,...p_n) have been calculated.

Step 9: Determine the probability for each class.

Step 10: Maximize your chances.

Step 11: Return Classification outcomes of movie recommendation system

Cosine similarity

Cosine similarity, a mathematical concept employed in various fields, particularly shines in the realm of text analysis. This similarity metric operates within the framework of an inner product space, where vectors represent the numerical representations of entities under consideration. Its fundamental principle revolves around measuring the cosine of the angle formed between two vectors, providing a reliable indicator of their alignment or similarity. In practical terms, cosine similarity excels at gauging the likeness of vectors, especially in the context of document similarity analysis. Each document, treated as a vector in a high-dimensional space, can be characterized by the frequency or presence of specific terms, forming a unique vector representation. Cosine similarity then becomes a valuable tool to assess how closely these vectors align in this multi-dimensional space [14, 34].

A document can be depicted through numerous attributes, each capturing the occurrence of a specific word or an expression within the document. Consequently, each

Table 2 Document vector or term-frequency vector

Document	Team	Coach	Hockey	Baseball	Soccer	Penalty	Score	Win	Loss	Season
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

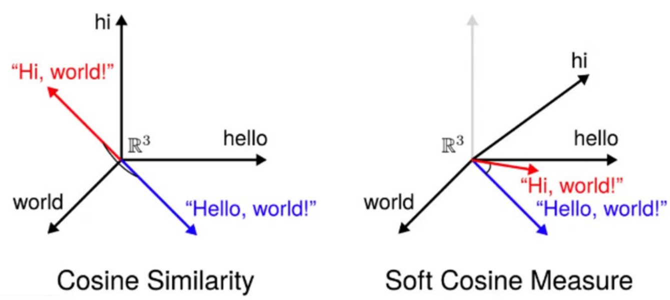

Fig. 10 Cosine similarity computation [14, 34]

document transforms into an entity characterized by a term-frequency vector. For instance, as demonstrated in Table 2, Document1 is composed of five occurrences of the term "team" and three instances of "hockey." Notably, the term "coach" is entirely absent from the document, denoted by a count value of 0. This data configuration can exhibit significant asymmetry. In cosine similarity, vectors serve as the representations of data objects within datasets. When defined in a product space, similarity is computed by examining the cosine of the angle between these vectors.

The smaller the distance, the greater the similarity, while a larger distance implies lower similarity. Cosine similarity functions as a metric for gauging the similarity of data objects, irrespective of their sizes (Fig. 10). From a mathematical perspective, it is the cosine of the angle formed by two vectors projected within a multi-dimensional space [14, 34]:

$$\mathbf{Cos}\boldsymbol{\theta} = \frac{\vec{\boldsymbol{a}}\,,\,\vec{\boldsymbol{b}}}{\parallel \vec{\boldsymbol{a}} \parallel \; \parallel \vec{\boldsymbol{b}} \parallel} = \frac{\sum_1^n \boldsymbol{a_i b_i}}{\sqrt{\sum_1^n \boldsymbol{a}_i^2}\sqrt{\sum_1^n \boldsymbol{b}_i^2}} \tag{2}$$

where Cos θ is the cosine similarity between vectors $\vec{a}$ and $\vec{b}$, and a_i and b_i are the elements of the vectors a and b, respectively.

Support vector machine

Classifying data is a common machine learning task where given data points belong to two classes. The objective is to determine the class of a new data point. In support vector machines, data points are treated as p-dimensional vectors, and the goal is to separate them with a (p1)-dimensional hyperplane, forming a linear classifier (Fig. 11). The optimal hyperplane, known as the maximum-margin hyperplane, is chosen to maximize the distance to the nearest data point on each side [38].

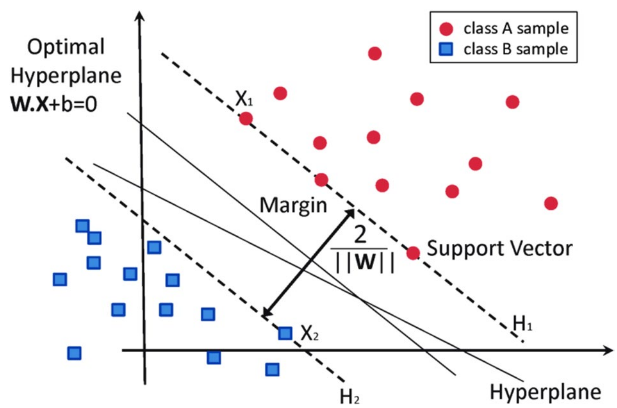

Fig. 11 Support vector machine [38]

The VADER sentiment intensity analyzer

The Viscous Accretion Disk Evolution Resource (VADER) sentiment intensity analyzer [42] calculates the sentiment scores as a combination of lexical and grammatical features present in the input text. The core idea behind VADER is to use a pre-built sentiment lexicon (a dictionary of words with associated sentiment scores) and a set of rules to analyze the sentiment of a given piece of text. Lexicon-based approach VADER uses a pre-built lexicon of words where each word is assigned a polarity score (positive, negative, or neutral).

The lexicon also includes sentiment modifiers that can adjust the sentiment intensity of the words they modify scoring rules. VADER employs a set of rules to handle sentiment intensifiers, negations, and other linguistic nuances that affect sentiment. For example, "very good" might be interpreted as more positive than "good." The individual word scores are combined to calculate the compound score (referred to as aggregation to compound score), which is a normalized and weighted sum. The compound score takes into account the overall sentiment of the text. To make a decision logic for positive, neutral, and negative, the compound score is then used to categorize the overall sentiment into positive, neutral, or negative based on a threshold. The threshold values (e.g., 0.05 and -0.05 in the previous example) are somewhat arbitrary and can be adjusted based on the desired level of sensitivity [42]. Figure 12 shows the steps of the proposed sentence intensity approach.

Linear combination for merged sentiment score

The sentiment value is combined with the recommender's findings (expressed as cosine similarity). The merged sentiment score (S_{merged}) is calculated by combining the separate sentiment scores as follows:

$$S_{\text{merged}} = \alpha \cdot S_{\text{s}} + \beta \cdot S_{\text{cosine}} \quad (3)$$

where α and β are weighting coefficients that determine the relative influence of each component, and $\alpha + \beta = 1$ to ensure that the weights are normalized. We used $\alpha = 0.3$ and $\beta = 0.7$.

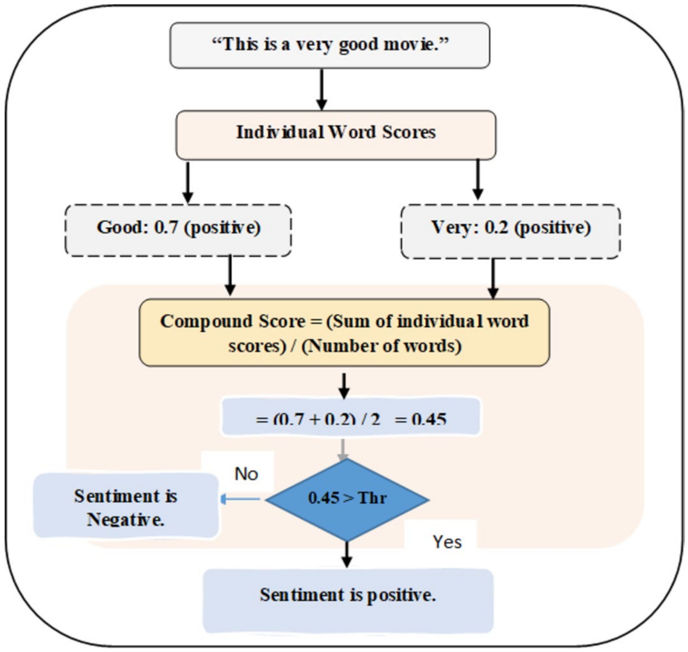

Fig. 12 Proposed sentence intensity approach

The choice of α and β is crucial and should be carefully determined to balance the impact of VADER and cosine similarity in the final sentiment analysis. The coefficients are often chosen based on experimentation and tuning to achieve the best performance.

- If α is higher, it means more emphasis is given to the sentiment score from VADER.
- If β is higher, it means more emphasis is given to the sentiment score from cosine similarity.

The linear combination of the two elements has several benefits as follows:

1. Compensating for limitations: VADER may excel in capturing sentiment nuances in social media language but might struggle with certain contexts. On the other hand, cosine similarity might be more robust in capturing general sentiment trends but less sensitive to language nuances [42]. By combining the two methods, the approach aims to compensate for each other's limitations.
2. Enhanced accuracy: The idea is that the combination of lexicon-based sentiment analysis (VADER) and vector-based similarity (cosine similarity) might enhance the overall accuracy of sentiment analysis by considering different aspects of the text [42].
3. Customization: The weighting coefficients allow for customization based on the specific needs and characteristics of the text data being analyzed.

Comments	scores	compound	comp_score
the da vinci code book is just awesome	{'neg': 0.0, 'neu': 0.631, 'pos': 0.369, 'comp...	0.6249	pos
this was the first clive cussler ive ever read...	{'neg': 0.0, 'neu': 0.871, 'pos': 0.129, 'comp...	0.5023	pos
i liked the da vinci code a lot	{'neg': 0.0, 'neu': 0.641, 'pos': 0.359, 'comp...	0.4215	pos
brokeback mountain was boring	{'neg': 0.434, 'neu': 0.566, 'pos': 0.0, 'comp...	-0.3182	neg
so brokeback mountain was really depressing	{'neg': 0.367, 'neu': 0.633, 'pos': 0.0, 'comp...	-0.4391	neg

Fig. 13 Dataset details

	movie_id	title	overview	genres	keywords	cast	
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"cre "52fe48009251416c750a
1	285	Pirates of the Caribbean: At World's	Captain Barbossa, long believed to be	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island,	[{"cast_id": 4, "character": "Captain	[{"cre "52fe4232c3a36847f800

Fig. 14 Sample of the TMDB dataset

Experimental results

In this section, we show the experiment results of the proposed movie recommendation system based on sentiment analysis with machine learning. The model was implemented using Python 3.11 ('base': conda) involving the NLTK framework running Juypter [43] and Colab [44].

Used databases

We used two different databases which has different environment and challenges: TMDB, Reviews. TMDB data contain two files: tmdb_5000_credits.csv, and tmdb_5000_movies.csv. It contains 24 columns, as shown in Fig. 13. A sample of the dataset is given in Fig. 14. Reviews dataset has two columns, namely comments and reviews, as shown in Fig. 15. The sentiment labels are distributed as in Fig. 16 where y-axis is the sentiment count in each of the three labels: positive, negative and neutral.

As a preprocessing stage, we add columns reviews for the movies dataset. The positive comments have been labeled as 1 and the negative ones have been labeled as 0. Implementation of the combined function of the merged sentiment score is straight forward, and we calculate the VADER sentiment score of the review text utilizing the VADER sentiment analyzer to obtain a sentiment score. Then, we normalize the

	Reviews	Comments
0	1	The Da Vinci Code book is just awesome.
1	1	this was the first clive cussler i've ever rea...
2	1	i liked the Da Vinci Code a lot.
3	1	i liked the Da Vinci Code a lot.
4	1	I liked the Da Vinci Code but it ultimatly did...

Fig. 15 Reviews dataset

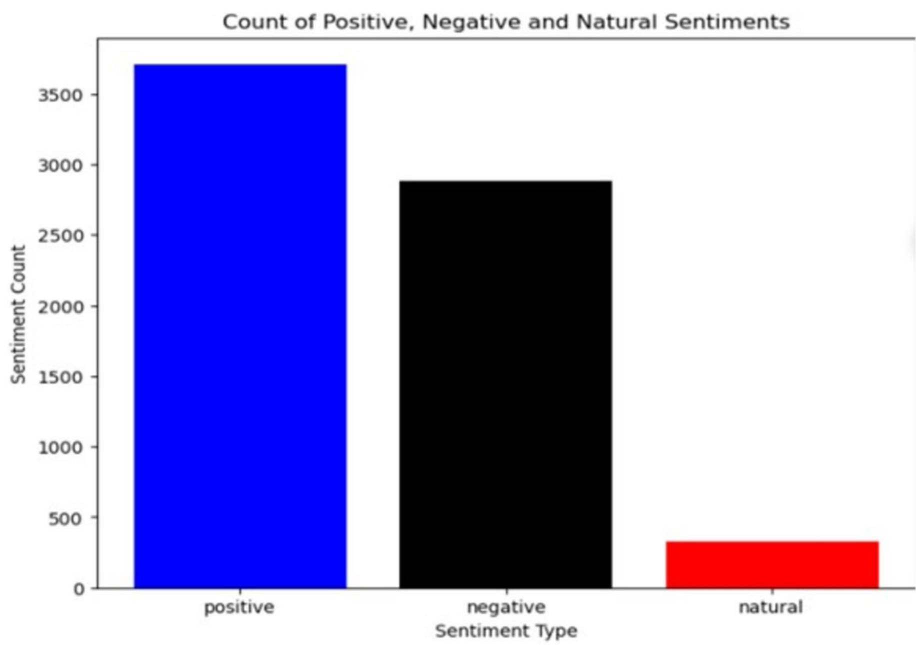

Fig. 16 Sentiment labeling count

recommender result (i.e., cosine similarity results) to a range between 0 and 1. Finally, we compute the merged sentiment score using Eq. (3) after defining the formula weighting coefficients (α and β) to find the optimal combination. Many values of α and β have been tried, until we reached the best performance.

Evaluation metrics

To evaluate our model, we have used various performance metrics such as accuracy, F1 score, precision, and recall which are calculated as follows [45, 46]:

$$\text{Recall (Sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

$$\text{Precision } = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

Table 3 Performance results of manually annotated

	Algorithm	Precision	Recall	F1-score
Macro-avg	Naïve Bayes	0.92	0.87	0.89
	Support vector machine	0.97	0.98	0.98
Weighted avg	Naïve Bayes	0.96	0.97	0.96
	Support vector machine	0.99	0.99	0.99

$$\text{Accuracy} = \frac{\text{Correct prediction}}{\text{Total prediction}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (6)$$

$$\text{F}_1\text{\_score} = 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

where TN is the true negative cases in which the actual classification is negative and the predicted classification is negative, TP is the true positive cases in which the actual classification is positive and the predicted classification is positive, FN is the false negative cases in which the actual classification is positive and the predicted classification is negative, and FP is the false positive cases in which the actual classification is negative and the predicted classification is positive. Both FP and FP indicate the errors in classification.

Macro-average is also computed for precision, recall and as F1-score using [45]:

$$\text{Macro Avg} = \frac{\text{Metric\_Class\_1} + \text{Metric\_Class\_2} + \cdots + \text{Metric\_Class\_N}}{N} \quad (8)$$

where $\text{Metric\_Class}_\text{i}$ is the precision, and recall or F1-score metric for class i and N is the number of classes,

while weighted average is also computed for precision, recall and as F1-score using [45]:

$$\text{Weighted Avg} = \frac{\text{Metric\_Class\_1} \times \text{Count\_Class\_1} + \text{Metric\_Class\_2} \times \text{Count\_Class\_2} + \cdots + \text{Metric\_Class\_N} \times \text{Count\_Class\_N}}{N} \quad (9)$$

where $\text{Metric\_Class}_\text{i}$ is the precision, recall or F1-score metric for class i, $\text{Count\_Class}_\text{i}$ is the count of samples in class i, and N is the number of classes.

Experimental results and comparisons

In our innovative movie recommendation system, we leverage cosine similarity to enhance the accuracy and relevance of our recommendations. For the training of the review dataset, we employ state-of-the-art machine learning models, specifically support vector machine (SVM) and Naive Bayes. We compare the results of a manual annotated sentiment review against the results of automated sentiment analyzed review using VADER approach in Tables 3 and 4.

We found out that the automated sentiment analyzed review exhibits higher performance, SVM 99.28% and Naïve Bayes 96.60%, than those who have been manually annotated: SVM 98.34% and Naïve Bayes 96.97%. This robust combination of cosine similarity and advanced machine learning algorithms ensures a high-quality and effective movie recommendation experience.

Table 4 Performance results of automated sentiment analyzed

	Algorithm	Precision	Recall	F1-score
Macro avg	Naïve Bayes	0.91	0.95	0.93
	Support vector machine	0.99	0.97	0.98
Weighted avg	Naïve Bayes	0.97	0.97	0.97
	Support vector machine	0.98	0.98	0.98

Figure 17 gives the confusion matrix of the proposed recommender. As seen from the figure, the number of false negative (FN) and false positive (FP) is very small compared to the true positive and true negative. This ensures the reliability of the recommender system.

Figure 18 shows a sample of the recommender output for one of the movies: "Spectre." Five recommended movies are given according to the rating and the feedback by the viewers of such movies. Table 5 gives a comparison with previous studies, which shows the superiority of our approach results over these studies.

Challenges and future work

The recommendation system stands out as the paramount method for information filtering, particularly in managing vast datasets and establishing meaningful connections with users. This study focuses on movie recommendations for upcoming releases, employing sentiment analysis on user-generated content from social media platforms. Specifically, we extract comments from Netflix's official YouTube channel, assess the overall sentiment, and predict the rating for unreleased movies.

Our proposed model combines data from the TMDb dataset, encompassing both existing and upcoming movies. This hybrid recommender system generates a curated list of preferred upcoming movies. Sentiment analysis is executed using VADER and TextBlob approaches, predicting both VADER and TextBlob ratings. While these methods generally yield accurate results, discrepancies were observed for movies numbered 50, 55, and 60 compared to IMDB ratings.

This experiment focused on English-language YouTube comments from Netflix's official channel. Future studies could explore other social media platforms like Twitter, incorporate cross-lingual comments, and consider additional keywords such as "Lol" and "Omg," as well as emojis.

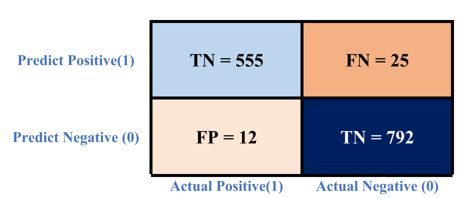

Fig. 17 Confusion matrix

```
print("Enter a movie: ", end=" ")
movie=input()
print('----------------------')
print('Recommended Movies Are:')
recommend(movie)
```

```
Enter a movie:  Spectre
----------------------
Recommended Movies Are:
['Quantum of Solace',
 'Never Say Never Again',
 'Skyfall',
 'Thunderball',
 'From Russia with Love']
```

Fig. 18 Output sample

Conclusion

This paper is structured into two main sections, with a focus on both movie recommendation systems and sentiment analysis. The detailed examination of these systems has yielded significant conclusions. In the movie recommendation system, the utilization of the cosine similarity algorithm has proven effective in suggesting movies related to the user's input. Factors considered for recommendations include the movie's genre, overview, cast, and ratings. Through multiple tests, cosine similarity consistently provided accurate movie suggestions.

The study also emphasizes the pivotal role of sentiment analysis, aimed at classifying reviews as positive or negative. Two algorithms, Naive Bayes (NB) and support vector classification (SVC), were employed to determine the most effective classification method. The diverse nature of reviews prompted the use of two algorithms to ascertain the optimal choice. Experimental results indicate a slightly superior accuracy of the SVM algorithm over NB.

Several prospects for future enhancements are outlined:

Table 5 Comparison with previous models

Algorithm	Accuracy
Proposed recommender with manual sentiment analysis	SVM 98.34% Naïve Bayes 96.97
Proposed recommender with VADER sentiment analysis	SVM 99.28% Naïve Bayes 96.60%
Cosine similarity [34]	98.77%
CNN, cosine similarity [29]	21.07%
NB, SVM, CNN [14]	98.6%
DNN + CRS + Friendlink [35]	91%
Content-based + VGG for feature selection + Euclidean distance [36]	90%
Gradient descent [37]	95%

- Increasing the accuracy of sentiment analysis for improved classification, particularly for sarcastic or ironic reviews.
- Expanding sentiment analysis to reviews in languages other than English.
- Enhancing movie recommendations based on users' preferences, considering factors such as cast, genre, and year of release.
- Despite the system's high accuracy, it does face limitations. If a user-entered movie is absent from the dataset or if the user input does not match the dataset's format, the system may fail to recommend movies. Additionally, a linguistic barrier exists in sentiment analysis, restricting the system to English reviews at present. Addressing these limitations could further enhance the system's overall effectiveness.

Author contributions

AS, HA, BA and MW invovled in conceptualization of this study, adding the basic ideas writing and editing the manuscript. BA, AA and RO invovled in formal analysis, methodology, software, validation, and writing original draft preparation. AS, HA, MW, BA, AA and RO invovled in investigation, methodology, and validation.

Funding

No funding is received.

Availability of data and materials

The dataset used in this paper is publicly available online at: https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata, https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.

Declarations

Competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Received: 6 August 2024 Accepted: 10 November 2024
Published online: 22 November 2024

References

1. Alyari F, Navimipour NJ (2018) Recommender systems: a systematic review of the state-of-the-art literature and suggestions for future research. Kybernetes 47:985–1017
2. Caro-Martinez M, Jimenez-Diaz G, Recio-Garcia JA (2018) A theoretical model of explanations in recommender systems. In: Proceedings of the ICCBR, Stockholm, Sweden pp 9–12
3. Ambikesh G, Rao SS, Chandrasekaran K (2024) A grasshopper optimization algorithm-based movie recommender system. Multimed Tools Appl 83(18):54189–54210
4. Yuri A, Widiyaningtyas T (2024) A systematic review of movie recommender systems. ITEGAM-JETIA 10(47):34–41
5. Movie industry statistic in 2020, SNL Kagan, 2020. https://www.spglobal.com/market-intelligence/en/news-insights/research/2021-box-office-rebounds-to-nearly-double-2020
6. Movies Investment Prediction up to 2030 in USA. https://www.researchandmarkets.com/report/united-states-film-motion-picture-market?srsltid=AfmBOoqjxgzVy_w0plVoEEFaPxT0i8-qKecIea9lg5NjjTNZ1Y3poxlH
7. Tufail S, Riggs H, Tariq M, Sarwat AI (2023) Advancements and challenges in machine learning: a comprehensive review of models, libraries, applications, and algorithms. Electronics MDPI 12(8):1789
8. Nesmaoui R, Louhichi M, Lazaar M (2023) A collaborative filtering movies recommendation system based on graph neural network. Proced Comput Sci 220:456–461
9. Yao Z (2023) Review of movie recommender systems based on deep learning. SHS Web Conf 159:02010
10. Sharma S, Rana V, Malhotra M (2022) Automatic recommendation system based on hybrid filtering algorithm. Education and Information Technologies, Springer, pp 1–16
11. Garg A, Vats S, Jaiswal G, Sharma A (2021) Analytical approach for sentiment analysis of movie reviews using CNN and LSTM. International conference on artificial intelligence and speech technology. Springer, Cham, pp 99–115
12 Siles I, Espinoza-Rojas J, Naranjo A, Tristán MF (2019) The mutual domestication of users and algorithmic recommendations on Netflix. Commun Cult Crit 12(4):499–518
13. Deshmukh N, Yadav L, Deharkar A (2024) Movie Recommendation System with Sentiment Analysis. Int Res J Modernization Eng Technol and Sci 6(5)
14. Pavitha N, Pungliya V, Raut A, Bhonsle R, Purohit A, Patel A, Shashidhar R (2022) Movie recommendation and sentiment analysis using machine learning. Global Trans Proc 3(1):279–284

15 Nassar N, Jafar A, Rahhal Y (2020) A novel deep multi-criteria collaborative filtering model for recommendation system. Knowl Based Syst 187:104811
16 Kalkar SD, Chawan PM (2022) Recommendation system using machine learning techniques. Int Res J Eng Technol (IRJET) 9:9
17. Huang Z, Lu X, Duan H (2011) Context-aware recommendation using rough set model and collaborative filtering. Artif Intell Rev 35:85–99
18 Seyam TA, Pathak A (2024) AgriScan: Next.js powered cross-platform solution for automated plant disease diagnosis and crop health management. J Electr Syst Inf Technol 11:45
19 Gemmell J, Schimoler T, Mobasher B, Burke R (2012) Resource recommendation in social annotation systems: a linearweighted hybrid approach. J Comput Syst Sci 78:1160–1174
20. Yi N (2017) Implementation of movie recommender system based on graph database. School of computer science communication, PhD. Thesis University of China Beijing, China
21. Elahi M, Ricci F, Rubens N (2016) A survey of active learning in collaborative filtering recommender systems. Comput Sci Rev 20:29–50
22. Mohanraj V, Chandrasekaran M, Senthilkumar J, Arumugam S, Suresh Y (2012) Ontology driven bee's foraging approach based self-adaptive online recommendation system. J Syst Softw 85:2439–2450
23. Hsu CC, Chen HC, Huang KK, Huang YM (2012) A personalized auxiliary material recommendation system based on learning style on facebook applying an artificial bee colony algorithm. Comput Math Appl 64:1506–1513
24 Beheshti A, Yakhchi S, Mousaeirad S, Ghafari SM, Goluguri SR, Edrisi MA (2020) Towards cognitive recommender systems. Algorithms MDPI 13:8
25. Antai R (2016) A new hybrid approach to sentiment classification PhD thesis, University of Essex
26 Dashtipour K, Gogate M, Adeel A, Larijani H, Hussain A (2021) Sentiment analysis of Persian movie reviews using deep learning. Entropy MDPI 23:5
27 Dang CN, Moreno-García MN, Prieta FD (2021) An approach to integrating sentiment analysis into recommender systems. Sensors MDPI 21:16
28. Karn AL, Karna RK, Kondamudi BR, Bagale G, Pustokhin DA, Pustokhina IV, Sengan S (2023) Customer centric hybrid recommendation system for E-Commerce applications by integrating hybrid sentiment analysis. Electron Commer Res 23(1):279–314
29. He H, Shang Z, Wu M, Zhang Y (2023) Movie recommendation system based on traditional recommendation algorithm and CNN model. Highlights Sci Eng Technol 34:255–261
30. Airen S, Agrawal J (2023) Movie recommender system using parameter tuning of user and movie neighbourhood via co-clustering. Proced Comput Sci 218:1176–1183
31. Behera G, Nain N (2023) Collaborative filtering with temporal features for movie recommendation system. Proced Comput Sci 218:1366–1373
32 Sahu S, Kumar R, MohdShafi P, Shafi J, Kim SK, Ijaz MF (2022) A hybrid recommendation system of upcoming movies using sentiment analysis of Youtube trailer reviews. Mathematics MDPI 10:9
33. Singh P, Srivastava G, Singh S, Kumar S (2023) Intelligent movie recommender framework based on content-based & collaborative filtering assisted with sentiment analysis. Int J Adv Res Comput Sci 14:3
34 Pawar S, Patne P, Ratanghayra P, Dadhich S, Jaswal S (2022) Movies recommendation system using cosine similarity. Int J Innov Sci Res Technol 7:4
35. Farashah MV, Etebarian A, Azmi R, Dastjerd RE (2021) A hybrid recommender system based on link prediction for movie baskets analysis. J Big Data 8:32
36. Mehta I, Kamdar A (2022) Movie recommendation system using composite ranking. arXiv preprint arXiv:2212.00139
37. Jamnekar MV, Bohra SU (2021) A hybrid approach for movie recommendation based on user behavior. Int J Sci Res Sci Technol 8:543–550
38. Kumar S, De K, Roy PP (2020) Movie recommendation system using sentiment analysis from microblogging data. IEEE Trans Comput Soc Syst 7(4):915–923
39. Gupta M, Thakkar A, Gupta V, Rathore DP (2020) Movie recommender system using collaborative filtering. international conference on electronics and sustainable communication systems (ICESC). IEEE, pp 415–420
40 Marappan R, Bhaskaran S (2022) Movie recommendation system modeling using machine learning. Int J Math Eng Biol Appl Comput 1:12–16
41 Widiyaningtyas T, Hidayah I, Adji TB (2021) User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system. J Big Data 8:53
42. Clayton H, Gilbert E (2014) VADER: A parsimonious rule-based model for sentiment analysis of social media text. Proc Int AAAI Conf Web Soc Med 8(1):216–225
43. Juypter. https://jupyter.org/
44. Colab. https://colab.research.google.com/
45. Sarhan A, Abdel-Rahem R, Darwish B, Abou-Attia A, Sneed A, Hatem S, Badran A, Ramadan M (2024) Egyptian car plate recognition based on YOLOv8, Easy-OCR, and CNN. J Electr Syst Inf Technol 11(1):32
46. Reyad M, Sarhan AM, Arafa M (2024) Architecture optimization for hybrid deep residual networks in liver tumor segmentation using a GA. Int J Comput Intell Syst 17(1):1–22
47. Modi P, Kumar A, Kapoor B (2023) Filmview: a review paper on movie recommendation systems. Iconic Res Eng J 6(12).

Publisher's Note