International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Efficient Hybrid Movie Recommendation System Framework Based on A Sequential Model

[1]Ravikumar R N, [2]Sanjay Jain, [3]Manash Sarkar

Submitted: 25/04/2023 **Revised:** 26/06/2023 **Accepted:** 05/07/2023

Abstract: A recommendation system is a system that offers suggestions to users, leveraging specific data such as books, movies, songs, and other relevant information. Movie recommendation algorithms utilize the attributes of previously enjoyed films to predict the preferences of users and suggest similar movies they might enjoy. Businesses that gather ample customer data and strive to deliver top-notch recommendations can greatly benefit from these recommendation systems. When creating a movie recommendation system, multiple elements such as genre, cast, and even the director of the movie are taken into account. This paper introduces a hybrid movie recommendation system that utilizes a combination of weighted average and min-max scaler to assess movie ratings and popularity. Moreover, TF-IDF is utilized for transforming the data into vectors, while cosine similarity is employed to gauge the resemblance among these vectors. The recommender system is built using the Movies dataset. The results show the top-K recommendation for users as well as the proposed system can provide a prediction of rating for a particular movie.

Keywords: *Recommendation system, sequential model, vectorizer, hybrid system.*

1. Introduction:

Amid the problem of ever-growing information overload, Recommender Systems (RS) are frequently employed to assist individuals or groups of individuals in locating the information they need. Over the past two decades, the volume of data has increased significantly due to the widespread use of online applications such as e-commerce, social networks, and multimedia streaming. With the vast amount of information available on websites, individuals often face difficulties in accessing the specific content they are looking for. The issue of information overload in knowledge engineering cannot be ignored, as users increasingly prioritize obtaining only relevant information.

Recommender Systems, which are mathematical models [1, 2], aim to provide customized recommendations to users by taking into account their individual preferences. Since the introduction of the Netflix Prize, there has been significant interest in recommender systems from both the business and academic communities [3]. When determining a user's recommendations, many variables are taken into account, including the user's interests, mood, tastes, and degree of resemblance to other users, to name a few [4]. According to the latest literature, the factors mentioned above are taken into account to enhance recommendation quality. There are three primary categories in which existing systems can be classified: Collaborative Filtering (CF), content-based filtering, and hybrid models [5].

In order to produce recommendations, knowledge-based recommendation systems combine product knowledge with underlying data or understanding of learner behaviour and interaction. Ontologies have proven to be effective in representing commonly accepted conceptualizations and accurately modelling essential knowledge related to learner behaviour, items, learner profile data, and other relevant factors. These ontologies are utilized across various platforms with different underlying structures. Hybrid recommendation systems, on the other hand, are referred to as recommender systems that leverage multiple algorithms in conjunction with one another [6, 7].

The movie recommendation system takes into account the user's actions during film watching indirectly. At the same time, the user's past ratings or history are directly used in the development of movie recommendation systems. The technique of employing the opinions of other users to filter or compute information is referred to as collaborative filtering [8, 9, 10]. Collaborative filtering encompasses the gathering of movie ratings or preferences from various users and subsequently providing movie recommendations to different users by considering their past preferences and interests.

The main objective of this paper is to present a fresh hybrid method for recommending items to users by

[1]*Department of Computer Science and Engineering, ASET, Amity University Rajasthan, Jaipur, India*
rnravikumar.cse@gmail.com
[2]*Department of Computer Science and Engineering, ASET, Amity University Rajasthan, Jaipur, India*
Jainsanjay17@yahoo.co.in
[3]*Department of Computer Science and Engineering, Atria Institute of Technology, Bangalore, India*
manashsarkar53@gmail.com

considering their interests. It has been suggested to use a hybrid approach for making movie recommendations, which combines weighted average and min-max scaler for popularity and rating of films. The process of vectorizing the information involves applying TF-IDF, and then quantifying the degree of similarity between the vectors using the cosine similarity function. The findings indicated that the top-K system for user recommendations and the suggested one for providing prediction rates for specific films.

The organization of the paper is as follows: Section 2 offers an overview of the literature survey, Section 3 presents an in-depth description of the proposed methodology, Section 4 demonstrates the results and discussion, and finally, Section 5 winds up the paper with a conclusion.

2. Literature Survey:

Over a period of time, different recommendation systems have emerged, employing collaborative filtering, content-based filtering, or a combination of both techniques.

The need to extract valuable information from large amounts of raw data to address business challenges has risen in parallel with the increasing urgency of corporate requirements. This trend is also applicable to digital recommendation systems, which have become widespread in consumer-focused sectors like books, music, clothing, movies, news articles, locations, and utilities. These systems gather user data to enhance future suggestions. Wu et al. [11] utilized Apache Mahout to implement two collaborative filtering algorithms for the development of a movie recommendation system in their study. Additionally, this research aims to delve deeper into the movie dataset through data analysis using Python's Matplotlib tools.

Ahuja et al. [12] created a movie recommender system by employing the K-Means Clustering and K-Nearest Neighbour algorithms. The team acquired the Movie Lens dataset from Kaggle and utilized Python as the programming language for implementation. The study aimed to introduce several principles of machine learning and recommendation systems. The construction of the recommender systems in this research involved the utilization of diverse tools and methods. Detailed explanations were provided for various methods, including K-Means Clustering, KNN, Collaborative Filtering, and Content-Based Filtering. Moreover, by conducting thorough research on various machine learning algorithms, valuable knowledge was acquired regarding the appropriate utilization of each algorithm in diverse fields, including recommender systems and e-commerce. The researchers then showcased the implementation and functionality of the proposed system for constructing the movie recommender system.

The goal of Gupta, et al [13] was to enhance a standard filtering technique's performance and accuracy. Content-based filtering stands as the most straightforward approach among various methods available for developing a recommendation system. The process included gathering input from users, comparing it with their previous activities and behaviour, and providing them with a selection of similar movies to consider. In order to improve the accuracy of the recommendations compared to content-based filtering, K-NN algorithms and collaborative filtering were utilized. This approach addressed the drawbacks of content-based filtering by incorporating collaborative filtering principles and employing the k-nearest neighbour technique based on cosine similarity. Despite a preference for Euclidean distance, cosine similarity was utilized due to its consistent accuracy in measuring the angle and equidistance between movies.

With the use of a model that integrates sentiment analysis and cosine similarity, Marappan, et al [14] focused on creating a movie recommendation system. Cosine similarity is a metric used to compare two objects' resemblance to one another. The degree to which a movie review is positive or negative—and, thus, its total score—can be assessed by looking at the emotions it expresses. Due to its ability to learn from training and evaluate data, the machine has the potential to automate the assessment of whether a review is positive or negative. Over a period of time, outcomes from comparing various systems using content-based methods will become clearer.

Sanwal et al. [15] introduced an innovative hybrid recommender system with improved performance. The system being suggested categorizes users into two primary segments: regular users and non-regular users. To enhance outcomes, a variety of machine learning and deep learning approaches are utilized within each segment. Regular users utilize techniques like decision trees, support vector regression, and random forests to achieve their goals. On the other hand, various deep learning techniques, collaborative filtering, and matrix factorization are used for non-regular users. Compared to the conventional ways, this strategy is more effective.

An innovative deep autoencoder network-based hybrid social recommender system was presented by Tahmasebi, et al [16]. The suggested strategy used social influence from users, collaborative filtering, and filtering based on content. The level of social influence attributed to each user is established by analyzing their Twitter behaviours and personal characteristics. MovieTweetings and the Open Movie Database have provided the necessary datasets for the evaluation.

When developing a movie recommendation system, factors like the film's genre, the actors featured in it, and even the director can be considered. One, more than one, or a mix of these attributes may be used by the systems to suggest films. Reddy et al. [17] considered the user's preferred genres when providing recommendations. They employed a content-based filtering approach that relies on genre correlation to accomplish this. Movie Lens is the dataset that the system uses.

In order to improve the user experience, recommendation systems provide quick and rational decisions. Singh and colleagues [18] presented an approach that offers users generalized recommendations by considering the film's popularity and/or genre. The implementation of the Content-Based Recommender System involved several deep-learning techniques. Moreover, this study provides a valuable understanding of the difficulties encountered by content-based recommendation systems and our efforts to address them.

Geetha, et al [19] proposed a film recommendation approach capable of suggesting movies to both new users and those already acquainted with the system. The method involves mining film databases to collect essential statistics such as popularity and aesthetics, which are crucial for generating recommendations. Our system enhances the accuracy of movie suggestions by employing a combination of content-based, collaborative, and hybrid filtering techniques, thereby leveraging the advantages of these two strategies.

Wang et al [20] introduced a movie recommendation framework implemented on the Spark platform. The framework incorporated a hybrid recommendation model and employed sentiment analysis techniques to enhance the precision and timeliness of mobile movie recommender systems. The suggested strategy starts by creating a preliminary list of recommendations using a hybrid recommendation mechanism. Once the list has been optimised, sentiment analysis is used. Last but not least, the Spark platform integrates a hybrid recommender system with sentiment analysis. The sentiment analysis-hybrid recommendation model outperforms the conventional models.

Predictory, proposed by Walek et al [21], is an advanced hybrid recommender system known for its monolithic architecture. The system utilizes a combination of different elements, such as a collaborative filtering system employing the SVD algorithm, a content-based system, and a fuzzy expert system, to generate movie recommendations. By utilizing these modules, Predictory considers the user's preferred and disliked genres and incorporates a fuzzy expert system to assess the significance of each film. It is worth mentioning that the noteworthy aspect of this research lies in the creation of a sophisticated hybrid system for selecting movies. The system's effectiveness was assessed using the MovieLens dataset and compared to traditional recommender systems.

Recommendation systems play a crucial role in websites and e-commerce applications, utilizing data clustering and computational intelligence to prioritize recommender systems. Baht et al. [22] presented a novel approach to developing a collaborative movie recommendation system called ACO-KM, which is a hybrid model merging K-means clustering and ant colony optimization. This model was specifically tailored to suit movie datasets. The proposed system's performance was analyzed and compared to existing works, demonstrating its ability to address scalability and efficiency challenges in recommendations, thereby reducing cold start problems.

Darban, et al [23] suggested a recommender system approach that utilizes a graph-based model for assessing similarities in user ratings, along with user demographic and location data. To enhance the feature set, an Autoencoder feature extraction technique is employed to extract new attributes from the combined data. This approach, known as GHRS, surpasses several existing recommendation algorithms in accuracy by incorporating the supplementary feature set for user clustering.

3. Proposed Methodology:

Recommender systems offer a solution to the issue of overwhelming information faced by users of websites where they can rate particular items. One of the most effective, practical, and well-known programmes for people to view films quickly is called the movie recommender system. Researchers made numerous attempts to find solutions to these problems. The paper introduces a movie recommendation system that utilizes a hybrid approach, depicted in Figure 1 of the document.

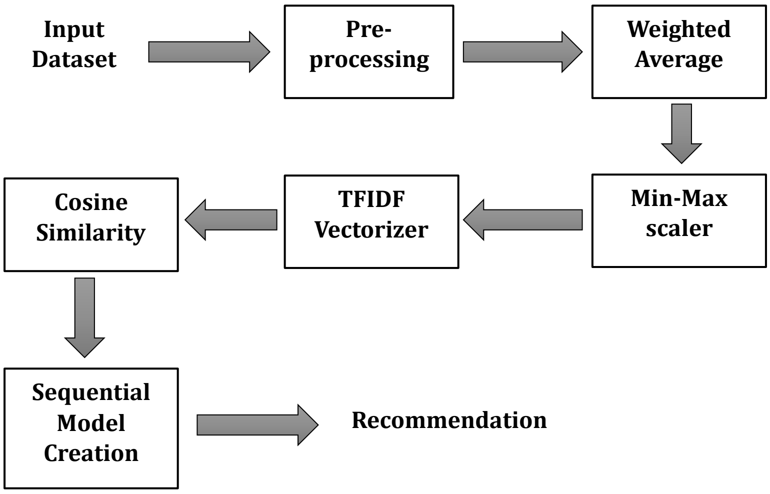


Fig 1: Block diagram of the proposed system

3.1 PRE-PROCESSING:

Pre-processing is the initial stage of the proposed recommender system. The dataset is taken from Kaggle, which needs to be cleaned before processing. Pre-processing is a method for organizing unstructured text to make it more understandable and represent the movie. Text processing is utilized to convert the movie data into a collection of terms identified within each film. The summary and names of the films are included in the movie data. Thus, the data cleaning process takes place, where the unwanted columns in the dataset have been removed initially. Then the datatype was changed for some of the columns. And somewhere the null values are replaced as required. And in some cases, the null values are dropped to clean the text data as well. In this way, the pre-processing takes place for further processing.

3.2 HYBRID TECHNIQUE:

The Hybrid approach leverages the advantages of both the Collaborative Filtering-based method and the Content-based approach [24], leading to an amplified benefit. In contrast to the Collaborative Filtering methodology, this method also calls for the utilisation of movie data and extra algorithms. Term weighting is a method that determines the importance of each term contained in each document and is frequently used in information retrieval research.

3.2.1 WEIGHTED AVERAGE:

The rating has been determined using a weighted average, which involves calculating the importance of each number in the dataset. Essentially, the weighted average considers the relative frequency or significance of specific factors within the data collection. A specified weight is multiplied by each value in the data set before the final computation is completed when calculating a weighted average.

Let x_i be the observations and w_i be the weights of the observations. The weighted average has been expressed as follows:

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

Where,

$\bar{x}$ = Weighted average

n = Number of terms to be averaged

w_i = Weights applied to x values

x_i = Data values to be averaged

The final average value is more descriptive because it takes into account how significant each observation is in relation to the others.

3.2.2 MIN-MAX SCALER:

The popularity-based recommendation system does away with the need-to-know other parameters like browsing history, preferences, movie star cast, genre, and other things. The star rating holds the utmost importance in the development of a scalable recommendation system. Thus, to obtain such popularity, the min-max scaler has been employed in the proposed recommender system. All feature values are converted to new values with the same scale (0-1) using the Min-Max Scaler. The equation below illustrates the Min-Max Scaler formula. Prior to the dataset being processed by Min-Max Scaler, the median of the features is used to fill in all features with null values for numeric data.

$$X_{std} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$$X_{scaled} = X_{std} \times (max - min) + min$$

Then the genre, overview, production, keywords and character crew also other details are added to the bag of words. Before converting words into vectors, the word scoring word frequency takes place. Words that appear frequently in the text start to predominate but may not provide the model with as much "informational content" as uncommon but potentially domain-specific words. There is some problem with word frequency scoring.

3.2.3 TF-IDF:

A possible approach involves adjusting the frequency of terms based on how often they occur in all documents, leading to decreased scores for commonly used words like "the" found in all publications. This scoring method is known as Term Frequency - Inverse Document Frequency (TF-IDF).

In the suggested system, TF-IDF is commonly employed as a part of content-based algorithmic recommendation systems. Inverse-Document-Frequency (IDF) and Term-Frequency (TF) are the two positions that make up this structure. Interests and preferences that may appear in user profiles are dealt with by TF. On the other hand, IDF centers its attention on the reciprocal of word frequency across the entirety of information furnished by the user profile. To generate a user recommendation using the information from their user profile, these two theories are merged.

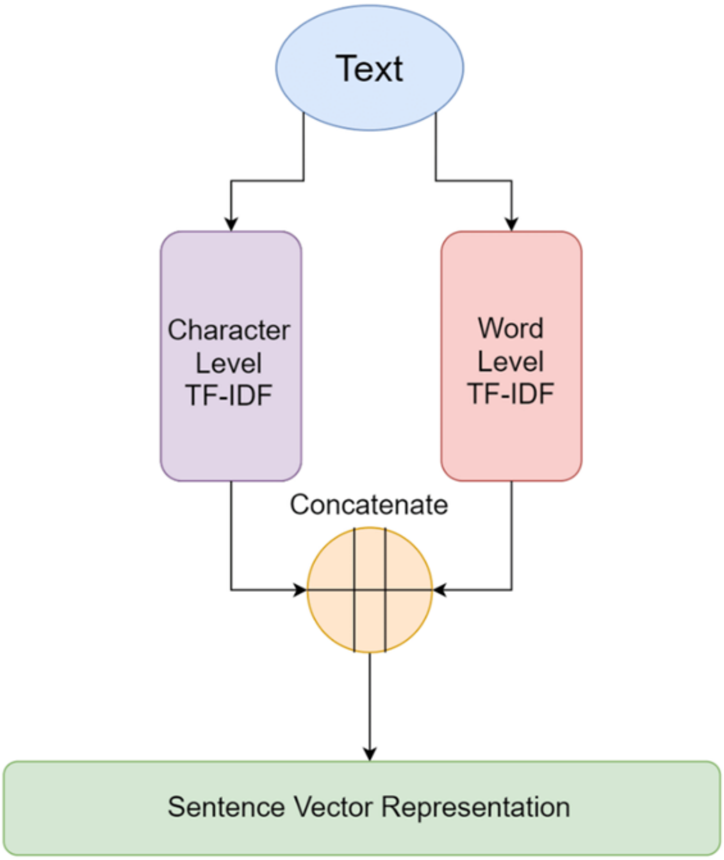


Fig 2: TF-IDF vectorization process

The user's preferences are suggested using the Term-Frequency Inverse-Document-Frequency (TF-IDF) algorithm. By applying the TF-IDF vectorization algorithm mentioned earlier, each attribute of the datasets is transformed into a vector. The cosine similarity method is used to calculate the similarity measure for each vector. Corresponding quantities are generated for films based on a user's request for a specific film and a specified number of suggestions. A confidence score indicating the similarity to the represented film is assigned to each comparable movie found in the descending order of similarity. The indexes of these films are collected according to the user's requested number of recommendations, resulting in a list of films displayed to the user. An interface allows the user to view the recommendations generated by the engine, which has been trained using training data to produce similarity metrics. The equations below were computed in Python as part of the backend process to determine TF-IDF.

The expression of the term frequency, also known as IF, is as follows:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Where n is the frequency of terms in the movie title.

IDF, i.e. inverse document frequency, demonstrates the inverse relationship between the number of words in a movie title and its mass, expressed as a reciprocal.

Therefore, the equation below can be used to determine the TF-IDF weight for the catchphrase in the record.

$$idf_j = \log\left[\frac{n}{df_j}\right]$$

$$TF\_IDF = (Frequency\ of\ words/Total\ words\ of\ sentence) \times (Total\ Documents/Documents\ containg\ the\ word)$$

3.2.4 COSINE SIMILARITY:

Cosine similarity calculates the cosine value between vectors to facilitate the comparison of similarity among attributes in a dataset. Any two pieces of text, including documents, sentences, paragraphs, and attributes, can be compared using the cosine similarity method. There are times when the vectors' similarity measurements result in unstable outcomes. The cosine similarity is graphically illustrated in Figure 3. Cosine similarity can be expressed as:

$$\cos\theta = \frac{A.B}{\|A\| \times \|B\|}$$

The similarity between vectors can be expressed based on vector similarity using cosine similarity.

$$\cos\theta = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}}$$

Where, A_i and B_i are components of vector A and B accordingly.

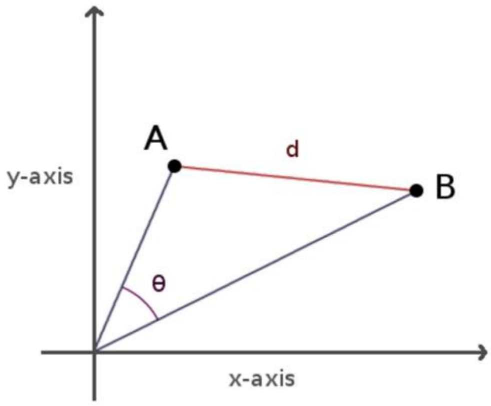


Fig 3: Cosine Similarity

3.3 SEQUENTIAL MODEL CREATION:

The Sequential API enables us to construct models layer by layer, making it suitable for a wide range of problems. By utilizing the Sequential model API, we can create deep learning models by instantiating the Sequential class and subsequently adding model layers to it. Users will generate a significant number of interaction behaviours in recommender systems over time. The sequential recommendation technique anticipates the user's subsequent interaction item by taking information from these behavioural sequences.

The system's objective is to forecast the user's next interaction with the item x_{n+1} for the series of items $(x_1, x_2, \ldots x_n)$ that they have previously interacted with in order to represent this challenge. The user's historical behaviours are crucial for modelling the user's interest in recommender systems. The model is trained using each user's behaviour as a sample in many widely used recommendation techniques, such as collaborative filtering [25].

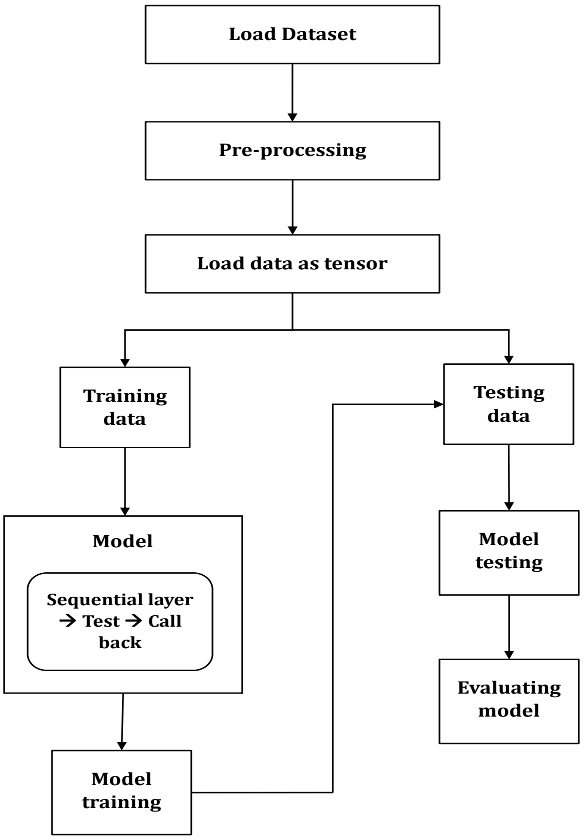


Fig 4: Flow chart of model creation

Sequential recommendation, in contrast, learns timestamp-aware sequential patterns from the user's past behaviour sequence to recommend the following item that the user would be interested in. The user's preference for a specific item is explicitly modelled by them. Sequential recommending poses two fundamental challenges.

Extracting as much valuable information as possible from the user's sequence of interests is crucial for enhancing the effectiveness of sequential recommendations. This entails understanding the user's diverse interests, such as their short-term, long-term, dynamic preferences, and other relevant types, to make accurate predictions about the next item that would capture their attention. The user's interest must first be collected from each sample, or each sequence, in order to anticipate the subsequent item. Simultaneously modelling the users' short-term, long-term, and dynamic interests is quite difficult, particularly as the sequence length grows. In order to improve representation learning, collaboration signals between distinct sequences must be captured in addition to modelling inside a sequence because items may occur in numerous sequences or users may have multiple sequences.

In the proposed sequential recommender model, the dataset was partitioned into a training dataset and a testing dataset. The sequential model has been created using the training dataset through tensor flow. Then the model has been evaluated with the training dataset, which provides the final movie recommendation for users.

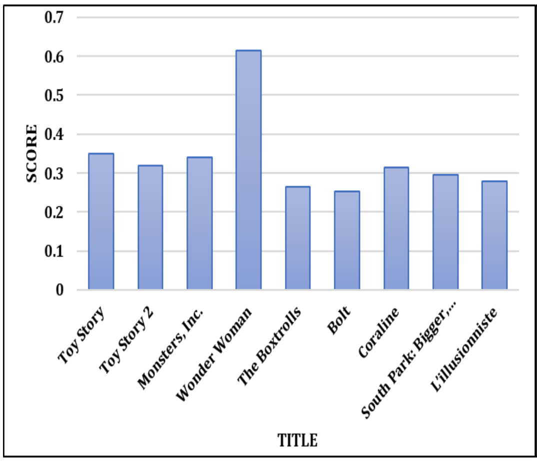
4. Results:

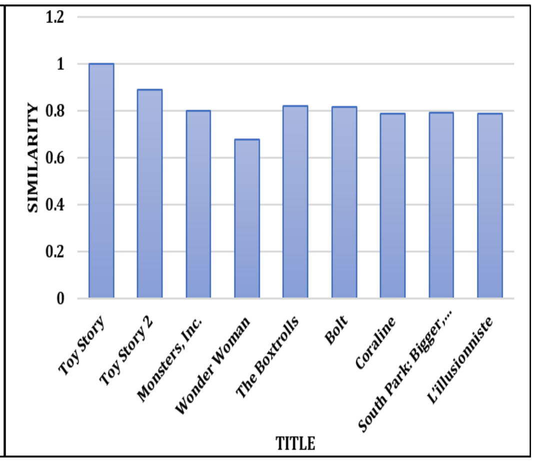
In this study, the Movie dataset was employed for both training and testing. The dataset comprises 10,004 movie reviews, covering 9,125 films and is gathered from a user base of 671 individuals. It incorporates a 5-point rating system, where 5 represents the highest possible score for a film. This dataset was created on October 17, 2016, utilizing data collected between September 9, 1995, and October 16, 2016, specifically for research purposes. Each user has given their opinion on at least 20 films. A user id is used to represent each person in the collection.

The created movie recommendation system is highly intuitive and effortlessly accessible for users. In this section, all the outputs and results of the above-mentioned models, processes and comparisons are given. Table 1 and Figure 3 show the scores and similarity scores for a few movies presented in the testing dataset.

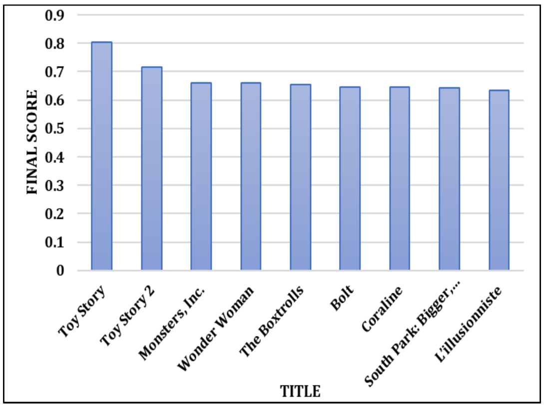
Table 1: Scoring parameters

Original Title	Score	Similarity	Final Score
Toy Story	0.348666	1.000000	0.804600
Toy Story 2	0.317960	0.888092	0.717053
Monsters, Inc.	0.340947	0.797036	0.660209
Wonder Woman	0.615227	0.678222	0.659323
The Boxtrolls	0.263451	0.820824	0.653612
Bolt	0.251638	0.815353	0.646239
Coraline	0.314661	0.788111	0.646076
South Park: Bigger, Longer & Uncut	0.295218	0.792142	0.643065
L'illusionniste	0.278807	0.787954	0.635210





(a) **(b)**



(c)

Fig 5: Scoring Parameters

The results have been shown for user 231. The hybrid movie recommendation system has recommended top-k movies. Here we have assigned k as 5. Thus, the recommender system recommends the top 5 recommended movies. For example, the top 5 recommendations for user 231 are presented in Figure 5. The ranking root mean square value obtained is 1.117 with a total loss of 986.974. Further, the accuracy for different k values as top-k categorial accuracy is presented in Figure 6 and graphically presented in figure 7. In addition, the proposed recommender system can predict the rate for a particular movie for the user. In Figure 8, the prediction rate for the Minion movie for user 231 is 3.0177.

Ranking RMSE : 1.117

top_1_categorical_accuracy : 0.001

top_5_categorical_accuracy : 0.004

top_10_categorical_accuracy : 0.009

top_50_categorical_accuracy : 0.036

top_100_categorical_accuracy : 0.071

Total loss : 986.974

Fig 6: Loss Parameters

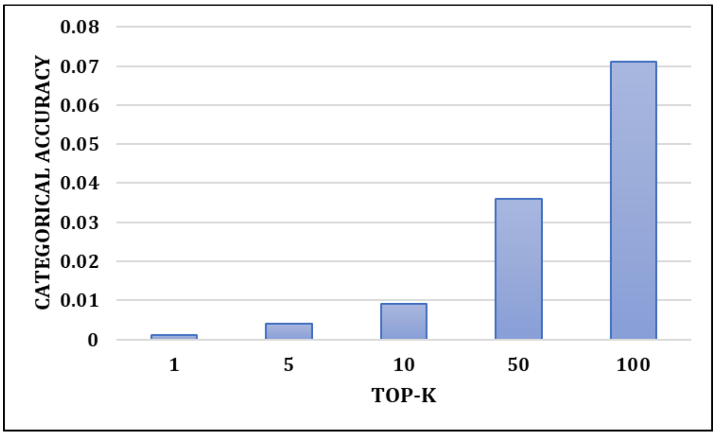


Fig 7: Top-K Categorical accuracy

Top 5 recommendations for user 231:

1. The Talented Mr.Ripley
2. Scarface
3. Scarface
4. Meet the Robinsons
5. Lonely Hearts

Predicted rate for Minions for user 231: 3.0177414417266846

Fig 8: Recommendation and Prediction Results

5. Conclusion:

The primary objective behind developing a film recommendation system is to provide consumers with suggestions tailored to their preferred movies, rather than relying solely on popularity or ratings. This personalized approach will significantly enhance the accuracy of the recommendation algorithm. In this paper, the hybrid movie recommendation system, which combines the weighted average and min-max scaler for movie rating and popularity, has been proposed. TF-IDF is applied to vectorize the information, after which the cosine similarity is computed to determine the similarity between the resulting vectors. Following the results, the top-5 recommendations for users as well as the proposed system can provide the prediction rate for specific movies. There is a lot of research being done on constructing recommender systems using deep learning and neural networks. It is discovered that systems created using these techniques attain high-performance accuracy.

References:

[1] Wang, Zhibo, Jilong Liao, Qing Cao, Hairong Qi, and Zhi Wang. "Friendbook: a semantic-based friend recommendation system for social networks." *IEEE Transactions on mobile computing* 14, no. 3 (2014): 538-551.

[2] Yuan, Nicholas Jing, Yu Zheng, Liuhang Zhang, and Xing Xie. "T-finder: A recommender system for finding passengers and vacant taxis." *IEEE Transactions on Knowledge and data engineering* 25, no. 10 (2012): 2390-2403.

[3] Zhang, Feng, Ti Gong, Victor E. Lee, Gansen Zhao, Chunming Rong, and Guangzhi Qu. "Fast algorithms to evaluate collaborative filtering recommender systems." *Knowledge-Based Systems* 96 (2016): 96-103.

[4] He, Jianming, and Wesley W. Chu. *A social network-based recommender system (SNRS)*. Springer US, 2010.

[5] Burke, Robin. "Hybrid recommender systems: Survey and experiments." *User modelling and user-adapted interaction* 12 (2002): 331-370.

[6] Tarus, John, Zhendong Niu, and Bakhti Khadidja. "E-learning recommender system based on collaborative filtering and ontology." *International Journal of Computer and Information Engineering* 11, no. 2 (2017): 256-261.

[7] Rabahallah, Kahina, Latifa Mahdaoui, and Faiçal Azouaou. "MOOCs Recommender System using Ontology and Memory-based Collaborative Filtering." In *ICEIS (1)*, pp. 635-641. 2018.

[8] Luo, Xin, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems." *IEEE Transactions on Industrial Informatics* 10, no. 2 (2014): 1273-1284.

[9] Badsha, Shahriar, Xun Yi, and Ibrahim Khalil. "A practical privacy-preserving recommender system." *Data Science and Engineering* 1 (2016): 161-177.

[10] Ullah, Farman, Ghulam Sarwar, Sung Chang Lee, Yun Kyung Park, Kyeong Deok Moon, and Jin Tae Kim. "Hybrid recommender system with temporal information." In *The International Conference on Information Network 2012*, pp. 421-425. IEEE, 2012.

[11] Wu, Ching-Seh Mike, Deepti Garg, and Unnathi Bhandary. "Movie recommendation system using collaborative filtering." In *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 11-15. IEEE, 2018.

[12] Ahuja, Rishabh, Arun Solanki, and Anand Nayyar. "Movie recommender system using k-means clustering and k-nearest neighbour." In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 263-268. IEEE, 2019.

[13] Gupta, Meenu, Aditya Thakkar, Vishal Gupta, and Dhruv Pratap Singh Rathore. "Movie recommender system using collaborative filtering." In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 415-420. IEEE, 2020.

[14] Marappan, Raja, and S. Bhaskaran. "Movie recommendation system modelling using machine learning." *International Journal of Mathematical, Engineering, Biological and Applied Computing* (2022): 12-16.

[15] Sanwal, Muhammad, and Cafer ÇALIŞKAN. "A hybrid movie recommender system and rating prediction model." *International Journal of Information Technology and Applied Sciences (IJITAS)* 3, no. 3 (2021): 161-168

[16] Tahmasebi, Hossein, Reza Ravanmehr, and Rezvan Mohamadrezaei. "Social movie recommender system based on deep autoencoder network using Twitter data." *Neural Computing and Applications* 33 (2021): 1607-1623.

[17] Reddy, S. R. S., Sravani Nalluri, Subramanyam Kunisetti, S. Ashok, and B. Venkatesh. "Content-based movie recommendation system using genre correlation." In *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 2*, pp. 391-397. Springer Singapore, 2019.

[18] Singh, Ramni Harbir, Sargam Maurya, Tanisha Tripathi, Tushar Narula, and Gaurav Srivastav. "Movie recommendation system using cosine similarity and KNN." *International Journal of Engineering and Advanced Technology* 9, no. 5 (2020): 556-559.

[19] Geetha, G., M. Safa, C. Fancy, and D. Saranya. "A hybrid approach using collaborative filtering and content-based filtering for recommender system." In *Journal of Physics: Conference Series*, vol. 1000, no. 1, p. 012101. IOP Publishing, 2018.

[20] Wang, Yibo, Mingming Wang, and Wei Xu. "A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework." *Wireless Communications and Mobile Computing* 2018 (2018).

[21] Walek, Bogdan, and Vladimir Fojtik. "A hybrid recommender system for recommending relevant movies using an expert system." *Expert Systems with Applications* 158 (2020): 113452.

[22] Bahl, Dushyant, Vaibhav Kain, Akshay Sharma, and Mugdha Sharma. "A novel hybrid approach towards movie recommender systems." *Journal of Statistics and Management Systems* 23, no. 6 (2020): 1049-1058.

[23] Darban, Zahra Zamanzadeh, and Mohammad Hadi Valipour. "GHRS: Graph-based hybrid recommendation system with application to movie recommendation." *Expert Systems with Applications* 200 (2022): 116850.

[24] Dos Santos, Henrique Lemos, Cristian Cechinel, Ricardo Matsumura Araujo, and Miguel-Ángel Sicilia. "Clustering learning objects for improving their recommendation via collaborative filtering algorithms." In *Metadata and Semantics Research: 9th Research Conference, MTSR 2015, Manchester, UK, September 9-11, 2015, Proceedings 9*, pp. 183-194. Springer International Publishing, 2015.

[25] He, Xiangnan, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. "Neural collaborative filtering." In *Proceedings of the 26th international conference on world wide web*, pp. 173-182. 2017.

[26] Samad, A. . (2022). Internet of Things Integrated with Blockchain and Artificial Intelligence in Healthcare System. Research Journal of Computer Systems and Engineering, 3(1), 01–06. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/34

[27] Reddy V, S. ., Madhav, V. ., M, B. ., Krishna A, A. ., A, K. ., Inthiyaz, S. ., & Ahammad, S. H. . (2023). Hybrid Autonomous Vehicle (Aerial and Grounded). International Journal on Recent and Innovation Trends in Computing and Communication, 11(1), 103–109. https://doi.org/10.17762/ijritcc.v11i1.6056