IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Movie Recommendation System

Ms. Tisha Sachin Shah
Department of Information Technology and Computer Science
SK Somaiya College, Somaiya Vidyavihar University
Mumbai, India

Dr. Swati Maurya
Department of Information Technology and Computer Science
School of Basics and Applied Sciences, Somaiya Vidyavihar University
Mumbai, India

ABSTRACT

This research paper demonstrates the implementation of advanced machine learning filters used for collaborative and content-based recommendations. The delivery of personalized recommendations depends on these approaches because they analyze patterns of user choices together with elements of movies. The exploration builds recommendation accuracy through a K-Nearest Neighbors (KNN) and Recurrent Neural Networks (RNN) amalgamation. The joint operation of KNN and RNN services achieves optimal performance through rapid item and user similarity evaluation from KNN and RNN's sequential data processing for tracking user taste changes.

Various filtering approaches that merge collaborative and content-based methods receive evaluation in this research through accuracy assessments and recommendation expansion evaluations. The collaborative filtering framework helps content-based filtering systems accomplish their collection process by performing item-item comparison operations. Initial recommendations for matching movies originate from built-in content features which include genre classifications and directorial credits combined with casting information. Hybrid recommendation models combine different recommendation methods in order to address collaborative systems' cold-start problems and content-based approaches' targeted application conditions.

For this research, the datasets used are the Netflix Prize dataset and MovieLens, which are famous for their huge and diverse movie data. The datasets offered in these provide a solid basis for training and testing the proposed model. The study also compares the coverage and data quality with the IMDb Top 1000 dataset. Therefore, the Netflix Prize dataset provides large user-movie interaction data; MovieLens provides detailed movie metadata and user ratings, thereby achieving a fair evaluation of the system.

The proposed hybrid approach produces various advantages to enhance user-specific recommendations and dynamic response capabilities as well as better prediction accuracy. The research notes that RNN component training requires extensive datasets while also accepting the implementation challenges of this hybrid system. Although challenging to implement the hybrid system demonstrates promising capabilities to deliver relevant movie suggestions at the appropriate times.

The recommendation system which combines KNN and RNN enables development in research for movie recommendation platforms. The suggestion generation process in the system uses algorithm-matched technology to link content-based approaches with collaborative filtering for individualized recommendations. The core principles described in this work can help industrial fields strengthen their hybrid recommendation system creation process.

Keywords— Movie recommendation, collaborative filtering, content-based filtering, RNN, KNN, Netflix Prize, MovieLens, IMDb.

I. INTRODUCTION

User recommendation engines provide tailored content to viewers during their exploration of extensive content collections which emerge from information overcrowding. Accurate user preference forecasting serves as the base of movie suggestions since it generates positive user satisfaction outcomes in recommendation systems. Standard recommendation technology joins item-content filtering for user-item feature identification with user-item collaborative filtering that analyzes user and item relationships. Both content-based recommendations and collaborative filtering present challenges for system scalability and user base expansion.

The proposed recommendation framework combines the content based and collaborative filtering methods. The system combines content technology and content analysis for resolving these system challenges. The system incorporates two methods to leverage their benefits and overcome their limitations. The system employs K-Nearest Neighbors (KNN) and Recurrent Neural Networks (RNN) to achieve outcomes from both models without their respective drawbacks. Recommendations must be made to establish superior solutions for movie recommendation systems. The investigative research analyzes multiple filtering strategies together with algorithms for its recommendation evaluation. The system obtains evaluation data from MovieLens and Netflix Prize and IMDb to measure its performance metrics. knowledge into the efficiency of different strategies.

II. LITERATURE SURVEY

In the paper "A systematic review of movie recommender systems", by Yuri Ariyanto and Triyana Widyaningtyas (2024), a systematic literature review (SLR) of movie recommender systems (MRS) is presented. At the same time, it looks at the most recent processing and the data technique, the recommendation algorithms and the assessment technique, which was comprehensive science about the development of the field.

The study which is titled “Recommendation System using Machine Learning Techniques” by Shailesh D. Kalkar and Prof. Pramila M. Chawan focuses on the main aim of a recommendation system that is to predict user interests and understand the user’s mental process. The system tries to deliver relevant recommendations based on users’ needs and interests.

In the paper titled "A Review of Movie Recommendation System: Limitations, Survey and Challenges" by Mahesh Goyani and Neha Chaurasiya published in 2020, the authors talk about different filtering systems, their uses, advantages and limitations. The system efficiency and user similarity are emphasized by them to be improved through hybrid recommendation techniques. The study also examines the performance of the RJMSD similarity measure for movie recommendations and suggests adding more attributes of the movies to improve accuracy.

The paper titled “A Brief Analysis of Collaborative and Content Based Filtering Algorithms used in Recommender Systems” was written by Sri Hari Nallamala, Usha Rani Bajjuri, Sarvani Anandarao, Dr. D. Durgaprasad and Dr. Pragaban Mishra. This research presents a summary of various techniques and approaches used in the collaborative filtering framework for recommendation systems. Collaborative filtering, hybrid filtering and content-based filtering are the methods discussed. One of the most effective techniques of generating personalized recommendations is collaborative filtering among these.

The paper 'Content based movie recommendation system' by N. Pradeep, K. K. Rao Mangalore, B. Rajpal, N. Prasad, and R. Shastri (2020) also published in 2020, presents a method to recommend movies based on user input and metadata like cast, crew, keywords and genres. The authors discuss the need for recommendation systems for making personalized suggestions and the difficulties of dealing with large amount of data. In the proposed approach, item features are evaluated using cosine similarity and relevant recommendations are provided to users.

III. PROBLEM DEFINITION

Although recommendation system developments have not solved their present-day difficulties:

- High sparsity in user-item interaction matrices presents challenges to discover major trends when performing collaborative filtering operations.
- The systems encounter cold start problems due to their existence of newly added products as well as users. Users who join a system with minimal available information cannot receive useful recommendations at their early stage.
- An increase in item number and user quantity negatively impacts recommendation system performance thus creating operational limitations in real-time scenarios.
- The right amount of relevant recommendations must be established to achieve customer satisfaction without sending too many suggestions to the same user.

A hybrid recommendation framework will be developed that integrates RNN with KNN algorithm to merge content-based and collaborative filtering approaches as a solution to the established challenges. The system achieves high recommendation accuracy together with operational stability through data obtained from IMDb and Netflix Prize and MovieLens.

IV. METHODOLOGY

4.1 Data Collection and Preprocessing

Research relies on three main data sources for its operations.

- MovieLens provides enough data for collaborative plus content-based filtering because it stores user ratings alongside detailed movie metadata.
- A large collection of user ratings within the Netflix Prize functions optimally for developing collaborative filtering models.
- The IMDb database enables comprehensive access to complete content about actors and crew members and movie genres needed for content-based filtering operations.

4.2 Filtering Techniques

- **Content-Based Filtering**: In order to determine movie similarity, we apply a TF-IDF approach to the metadata from the films (genres, actors, and directors) in the datasets. Content based filtering uses RNNs to analyze the sequential data and extract temporal patterns of the item metadata and user's interactions. In order to produce accurate item profiles, features such as cast, director, and genres are included.
- **Collaborative Filtering**: For example, in order to compare each user by comparing users with similar rating behaviors, we use user based collaborative filtering. Collaborative filtering use previous interaction data to find similarity of users or items using KNN. This approach helps in recording the latent preferences and user behavior patterns.

4.3 Algorithm

K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is a nonparametric, instance-based machine learning algorithm very commonly used for classification as well as regression tasks. Since it is simple and effective in finding similar entities based on proximity feature space, KNN was used in this study. Based on a simple principle that similar data points are likely close to each other, the algorithm is a good pattern recognition, recommendation systems and anomaly detection tool.

How KNN Functions:

- Training Phase: For training phase, KNN is different from the other traditional machine learning models: it does not require an explicit weight update and parameter tuning for the training phase. It is a lazy learning algorithm, which means it stores the entire training dataset and computes only at the time of prediction. However, KNN is highly adaptable but computationally expensive at inference.
- Prediction Phase: In Prediction Phase, when a new data point enters, KNN finds the "K" nearest data points (neighbors) from the stored dataset as per a chosen similarity metric. The new point is determined to be of the class or value based on the aggregation of its neighbors' characteristics.
- Distance Calculation: The new data point's distance from every other point in the dataset is calculated via the algorithm. Euclidean distance is the most frequently utilized distance measure.
- Voting or Averaging Mechanism: The algorithm classifies the new data point to the class that occurs most frequently among its nearest neighbors (majority voting). The predicted value is computed as the average (or weighted average) of the values of the nearest neighbors.

Advantages of KNN:

- It is simple to implement and easy to understand.
- Does not require explicit training.
- It is applicable for classification and regression tasks.
- It is effective in capturing complex decision boundaries with an appropriate value of K.

Limitations of KNN:

- It is computationally expensive for large datasets because the entire dataset needs to be stored and searched.
- The choice of *K* and the distance metric is sensitive to performance.
- It is prone to unprecedented curse of dimensionality where increasing number of feature space dimensions may result in inefficiencies.

Recurrent Neural Networks (RNN):

Recurrent Neural Networks (RNNs) are a specific sort of artificial neural network intended for handling sequential information by catching temporal associations in a set of information. Contrary to traditional feed forward networks, RNNs can remember a form of memory to model sequential patterns across the time. For this reason, they are especially useful for natural language processing, speech recognition, as well as time series forecasting.

Functioning of RNNs:

1. Processing Sequential Inputs:
   Instead of feeding the entire dataset all at once, the network is fed one time step of data at a time.
   The current state of the network is determined by combining information from the previous step with each new input.
2. Maintaining a Hidden State:
   Information is stored in a hidden state of RNNs, which acts as memory, that keeps track of past inputs.
   The network learns to capture temporal dependencies of the inputs by updating this hidden state dynamically as new inputs arrive.
3. Generating Output:
   RNNs have different output modes depending on the specific application: An output with each step (for example, language modelling) or only for the whole sequence (for example, sentiment analysis).
   The current input and accumulated knowledge from previous time steps affect the output.
4. Training the Network with Backpropagation Through Time (BPTT).
   After all time steps have been processed, the network calculates the error by comparing the predicted and actual outputs.
   Then this error is propagated backward through the network all the way back to previous time steps, adjusting the network's parameters to better predict future predictions.
5. Handling Long-Term Dependencies:
   As all basic RNNs have vanishing gradients problem that limits the ability to retain information in far back time steps, advanced variants such as Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) help by selectively storing and forgetting the information.

Advantages of RNNs:

- It is very efficient for sequentially capturing patterns in the time series data.
- Suitable for modeling user interactions over time.
- It is extensible by attention mechanisms for better results on tasks such as machine translation.

Challenges of RNNs:

- Computationally intensive due to sequential processing.
- Limited in their ability to learn long range dependencies due to prone to vanishing and exploding gradients.
- It is slow compared to feedforward networks.

4.4 Model Training and Evaluation

- **Training**: IMDb data is used to further develop the content-based component of the hybrid model, that is trained using the MovieLens and Netflix Prize datasets.
- **Evaluation Metrics**: F1-score, Precision, Recall, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are implemented as evaluation metrics to evaluate the performance of the system.
- **Cross-Validation**: Protect against over fitting and generalizability by using cross validated.

V. LIBRARIES USED

The data processing, numerical computations, machine learning and the deep learning all have common easy to use libraries in python. In this section you will get a brief idea what are some of the essential libraries and their usage in data analysis and model development.

1. **Data Handling and Numerical Computation**

- **Pandas (pandas as pd)**
  - It is used for the purpose of structured data manipulation and analysis.
  - It provides DataFrame, a flexible tabular data structure similar to SQL tables or spreadsheets.
  - It is efficient in handling large datasets, missing values and transformations.
- **NumPy (numpy as np)**
  - A fundamental library for numerical operations in Python.
  - It supports multi dimensional arrays, matrix operations and a large set of mathematical functions for scientific computing.
- **Regular Expressions (re library)**
  - It is a built-in Python module for pattern matching and text manipulation.
  - It is used very widely for text preprocessing, cleaning, and extracting meaningful patterns from raw text data.

2. **Data Visualization**

- **Matplotlib (matplotlib.pyplot as plt)**
  - It is a widely used plotting library that supports creating static, animated or interactive visualizations.
  - It supports line plots, scatter plots, bar charts, histograms, and so on.

3. **Text Processing and Feature Extraction**

- **TF-IDF Vectorization (TfidfVectorizer from sklearn.feature_extraction.text)**
  - It converts raw text data to numerical representations by computing Term Frequency-Inverse Document Frequency (TF-IDF).
  - It can transform textual data into a structured format for the use in machine learning model.

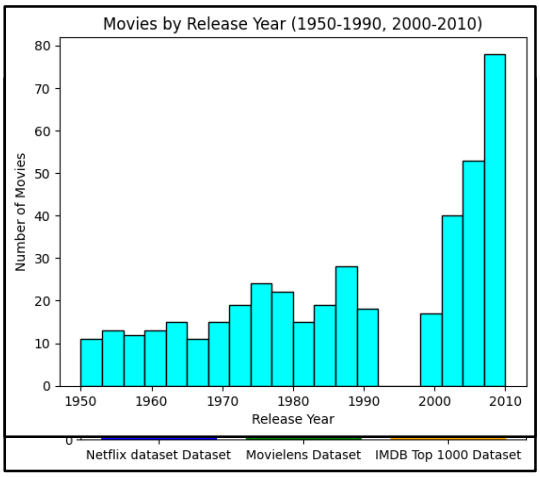
4. **Machine Learning Algorithms and Evaluation**

- **k-Nearest Neighbors (NearestNeighbors from sklearn.neighbors)**
  - KNN algorithm is implemented for both the supervised and unsupervised tasks including classification, clustering and recommendation systems.
  - It find similar objects based on distance metrics (Euclidean or cosine similarity etc.)
- **Train-Test Splitting (train_test_split from sklearn.model_selection)**
  - It splits the datasets into training and testing sets to make a robust evaluation of the machine learning model on the data that is also unseen.
- **Accuracy Score Evaluation for the Model (accuracy_score from sklearn.metrics)**
  - It compares predicted labels with actual labels and calculates the classification accuracy.

5. **Deep Learning and Neural Networks**

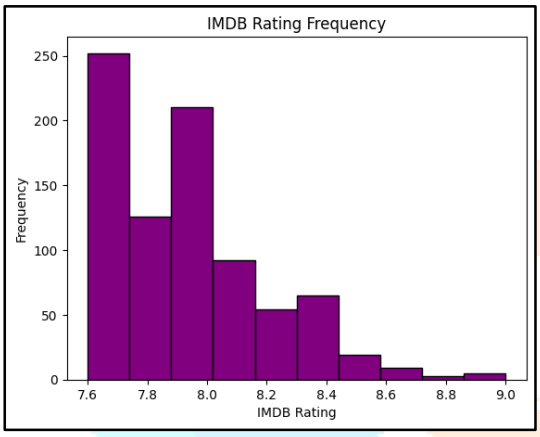
- **Sequential API (tensorflow.keras.models.Sequential)**
  - This is a simple module way to deploy deep learning models consisting a linear stack of layers.
- **Word Embeddings (tensorflow.keras.layers.Embedding)**
  - It is an essential part of Natural Language Processing (NLP) task, where it functions as a function to convert words or tokens into dense vector representations.
- **Long Short-Term Memory (LSTM) (tensorflow.keras.layers.LSTM)**
  - It is a type of recurrent neural network (RNN) layer that allows modeling to learn about long terms dependencies in sequential data including time series and text.
- **Fully Connected Layers (tensorflow.keras.layers.Dense)**
  - Dense (fully connected) layers are implemented, which are most commonly used in intermediary network layers and output layers for predictions.

Such libraries and functions are the base of data science, machine learning and deep learning applications. With the use of their effective use, they can be used to efficiently process data, train models and evaluate them in different domains, such as natural language processing, time series forecasting and recommendation systems.

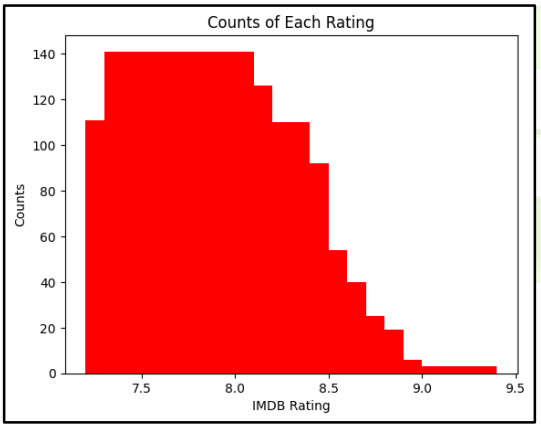


VI. IMPLEMENTATION

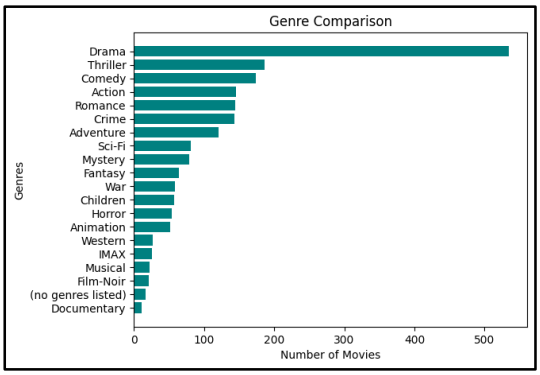
The bar chart compares the sizes of IMDB Top 1000, Netflix and MovieLens dataset. The comparison is made between the MovieLens dataset with over 90,000 films and the Netflix dataset with about 20,000 films. However, the IMDB Top 1000 dataset is the smallest with only 1,000 films. The graphic shows that the IMDB Top 1000 is a carefully curated list of the highest rated films, and the MovieLens dataset is the largest and most appropriate for large scale study. The color-coded bars make the image clearer.



The frequency distribution of the IMDB movie ratings in the dataset is illustrated by the histogram. 10 bins, ranging from 7.6 to 9.0, are used to group ratings. With the highest frequency seen close to 7.6, the bulk of films are scored in the lower range of this dataset, with the majority falling between 7.6 and 8.0. As ratings rise, the frequency gradually declines, with very few films receiving ratings higher than 8.6. This implies a lower prevalence of higher scores. As you can see, the IMDB rating distribution has one side very flat, having most films stick to the lower end of the scale, and then one side that rises up sharply, centered at the 10, representing movies with 10 stars on the IMDB.

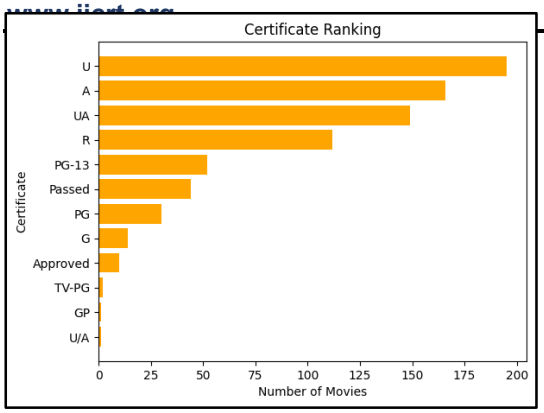


The distribution of IMDb ratings for a dataset is shown in the bar chart. The x axis represents IMDb ratings from roughly 7.5 to 9.5 and the y axis is the number of films that fulfill those ratings. The red bars indicate that, with counts peaking at about 8, the majority of the films in the sample have ratings that are clustered between 7.5 and 8.5. As ratings get closer to 9, the distribution significantly decreases, suggesting that there are fewer highly regarded films. This histogram suggests that the distribution of IMDb scores in the dataset is left skewed, meaning that mid to high ratings are more common than extremely high ratings.



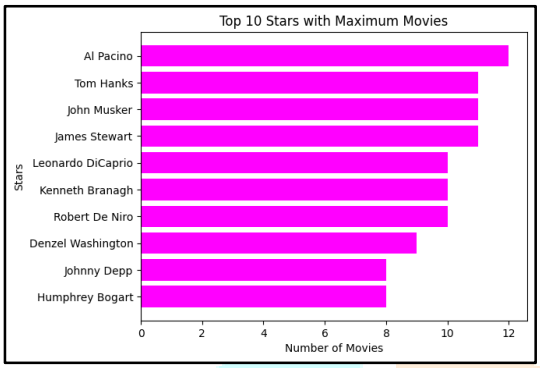
A horizontal bar chart is used to compare movie genre popularity of a dataset. The y axis is different genres while the x axis is the overall number of films related to that genre. "Drama" is the most popular genre, much more than the others, then there is "Thriller," "Comedy" and "Action," the least popular genres. The bars are arranged in descending order. Near the bottom are 'Documentary', 'Film-Noir' and 'Musical' which are fewer common genres. The chart shows that, unlike minor genres, which are not well represented in movies, dramatic and exciting genres are very well represented in the dataset.

The number of movies released from two time period: 1950–1990 and 2000–2010 are shown in the histogram. The release years are shown on the x axis and the number of movies on the y axis. The movie counts are divided into 20 bins and cyan colored bars indicate movie counts.



The chart shows a steady and moderate release rate from 1950 to 1990 with small peaks in the late 1970s and 1980s. From 2000 to 2010, movie production shows a sharp increase, reaching the peak in 2010 with the highest number of releases. This implies that the film industry grew significantly in the early 21st century as opposed to the previous decades.

The certificate ranking bar chart presents the distribution of movies according to their certification ratings. The number of movies for each certificate type is represented by the horizontal bars. The highest count is with the "U" certificate (suitable for all audiences), followed by "A" (adult audience) and "UA" (parental guidance). The certificates other than "R" (restricted), "PG-13," "PG," and "G" progressively have fewer movies. The least represented rare certifications are "Approved," "TV-PG," "GP," and "U/A." The chart is using orange bars and the y axis order is inverted for clarity, with the most frequent certificates at the top.



A bar chart, 'Top 10 Stars with Maximum Movies', lists the actors and celebrities with the most movies in the dataset. Each horizontal bar is the number of films, and each bar is a star. Al Pacino is the actor with the most films, followed by James Stewart, Tom Hanks and John Musker. Some well-known actors include Robert De Niro, Kenneth Branagh and Leonardo DiCaprio. Completing the top 10 are celebrities like Humphrey Bogart and Johnny Depp. Inverted y axis and magenta bars make the top stars with the most films more readable.

VII. BENEFITS

1. **Improved Accuracy:**
   The combined use of RNN and KNN recommends more accurately by using both temporal patterns and similarity metrics.
2. **Reduction of the Cold-Start Issue:**
   The system may generate recommendations for new users or items that require fewer interaction data through content-based filtering.
3. **Improved Scalability:**
   Since KNN can handle large data sets efficiently through optimized neighbor searches, adopting it for collaborative filtering assures scalability.
4. **Diverse Recommendations:**
   Hybrid approach establish a paradigm balance between diversity and accuracy that minimize over specialization and increases the satisfaction of users.
5. **Comprehensive Data Utilization:**
   The model is improved in terms of understanding user preferences and item features using a variety of datasets like MovieLens, Netflix Prize, and IMDb.
6. **Robustness:**
   The system becomes more resistant to data unpredictable nature and sparsity when multiple algorithms and filtering techniques are integrated.

VIII. LIMITATIONS

Although its advantages, the suggested approach has certain drawbacks:

1. **Computational Complexity:**
   Real-time recommendation capabilities might be affected by the rise in computational expenses that results from combining RNN and KNN.
2. **Data Dependency:**
   The completeness and quality of the datasets can make an important effect on the way the system performs. For inaccurate or skewed data, inaccurate recommendations might arise.
3. **Scalability Limits:**
   KNN can be extended to a very large extent, however, in case of extremely large datasets, more sophisticated techniques or dimensionality reduction approaches may be required.

4. **Cold-Start for New Items:**
   Adding completely new items with minimal metadata can still be challenging, regardless of how content-based screening takes care of new users.
5. **Parameter Tuning:**
   Sometimes it can invest a significant amount of time carefully modify the hybrid model's hyperparameters, such as the weighting factor in the fusion method.
6. **Limited Contextual Awareness:**
   Situational variables, user mood, or time of day may not be taken into account by the system in a way that will effectively make movie choices.

IX. CONCLUSION

In this research, the hybrid movie recommendation system combines the content-based and collaborative filtering techniques using such algorithms as RNN and KNN. The system integrates a number of datasets from MovieLens, Netflix Prize, and IMDb to make its recommendations more adaptable and accurate. Comparison evaluation demonstrates the superiority of the hybrid approach over the traditional methods of filtering and the way it can deal with the common problems such as the cold start problem and data sparsity.

Nevertheless, the system could be improved in terms of computational complexity and data requirements. In future research, more scalable, and more contextually understanding deep learning models, such as Transformer architectures can be included. Furthermore, having contextual data integrated with the real time data streams can finally result in more dynamic and nuanced recommendations that fit user's preferences better and set that.

Taking all things into consideration, the hybrid recommendation system has great potential in improving personalized content delivery and enhancing user experience in the ever-changing digital media environment.

REFERENCES

1. Shailesh D. Kalkar, Prof. Pramila M. Chawan. *Recommendation System using Machine Learning Techniques.* International Research Journal of Engineering and Technology (IRJET).
2. Demirtsoglou Georgios, Zisopoulos, Karagiannidis Savvas. *Content-Based Recommendation Systems.* Charilaos Zisopoulos.
3. Sri Hari Nallamala, Usha Rani Bajjuri, Sarvani Anandarao, Dr. D. DurgaPrasad, Dr. Pragnaban Mishra. *A Brief Analysis of Collaborative and Content Based Filtering Algorithms used in Recommender Systems.* Published under licence by IOP Publishing Ltd.
4. Yuri Ariyanto ,Triyanna Widiyaningtyas. *A systematic review of movie recommender systems.* Institute of Technology and Education Galileo da Amazônia, 2024.
5. M. Chenna Keshava , P. Narendra Reddy, S. Srinivasulu , B. Dinesh Naik. *Machine Learning Model for Movie Recommendation System.* International Journal of Engineering Research & Technology (IJERT).
6. Mahesh Goyani, Neha Chaurasiya. *A Review of Movie Recommendation System: Limitations, Survey and Challenges.* Electronic Letters on Computer Vision and Image Analysis 19(3) :18-37, 2020.
7. Antaris Stefanos, Demirtsoglou Georgios, Zisopoulos, Karagiannidis Savvas. *Content-Based Recommendation Systems.* Charilaos Zisopoulos
8. F. Furtado, A. Singh. *Movie recommendation system using machine learning*. International Journal of Research in Industrial Engineering.
9. Bhowmick, Hrisav Chatterjee, *Ananda Sen, Jaydip. Comprehensive Movie Recommendation System.*
10. Xavier Thomas. *Content-Based Personalized Recommender System Using Entity Embeddings*. 2020
11. N. Pradeep, K. K. Rao Mangalore, B. Rajpal, N. Prasad, R. Shastri. *Content based movie recommendation system*. Ayandegan Institute of Higher Education,, 2020.
12. Ruchika, Mayank Sharma, Syed Akhter Hossain. *Efficient Machine Learning Algorithms in Hybrid Filtering Based Recommendation System.* University of Tehran, 2023.
13. Javaji, Shashidhar, Reddy Sarode, Krutika. Hybrid Recommendation System using Graph Neural Network and BERT Embeddings.2023.
14. Yan, Yubing Moreau, Camille Wang, Zhuoyue Fan, Wenhan Fu, Chengqian. *Transforming Movie Recommendations with Advanced Machine Learning: A Study of NMF, SVD,and K-Means Clustering.*2024