



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 4, April 2023

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com



Movie Recommendation Engine Using Collaborative Filtering

Kishor Sonawane, Khushi Tiwari, Vedantika Wadkar, Rohit Prasad, Varsha Gosavi

UG Student, Dept. of C.S., SKNCOE, Savitribai Phule Pune University, Pune, India

UG Student, Dept. of C.S., SKNCOE, Savitribai Phule Pune University, Pune, India

UG Student, Dept. of C.S., SKNCOE, Savitribai Phule Pune University, Pune, India

UG Student, Dept. of C.S., SKNCOE, Savitribai Phule Pune University, Pune, India

Assistant Professor, Dept. of C.S., SKNCOE, Savitribai Phule Pune University, Pune, India

ABSTRACT: Movie recommendation engines are an essential tool for helping users discover movies that align with their preferences and interests. A movie recommendation system is a software tool that predicts a user's level of interest in a movie and recommends the most relevant movies from a vast array of options. The goal of a movie recommendation system is to provide users with personalized movie suggestions based on their viewing history and preferences. To determine these recommendations, movie recommendation systems employ machine learning and information retrieval techniques.

Collaborative recommendation services, also known as collaborative filtering, recommend movies based on the ratings and behavior of similar users. These services do not require any additional information about the movie itself. The recommendation system can employ algorithms based on user behavior and movie characteristics, as well as model and memory-based approaches.

KEYWORDS: Text detection, Inpainting, Morphological operations, Connected component labelling.

I. INTRODUCTION

Movie recommendation engines are an essential tool for helping users discover movies that align with their preferences and interests. Collaborative filtering is a widely used approach for building recommendation engines that leverage the behavior and preferences of similar users to generate recommendations. In this implementation paper, we present a movie recommendation engine that utilizes collaborative filtering to provide personalized and accurate movie recommendations.

We collected a dataset of movie ratings from a large group of users and developed a collaborative filtering algorithm that leverages this data to predict which movies a user might enjoy based on the ratings of similar users. We evaluated our engine using various metrics, including accuracy, precision, recall, and F1-score, and compared its performance to other state-of-the-art methods.

The rest of this paper is structured as follows: in the next section, we provide a comprehensive review of the existing literature on collaborative filtering for movie recommendation engines and identify the gaps and challenges in the field. Then, we describe our methodology for building the movie recommendation engine using collaborative filtering, including the data collection process, feature selection, algorithm selection, and evaluation strategy. In the following section, we present the results of our engine and compare its performance to other methods. We then discuss the implications of our findings and suggest future directions for research. Finally, we conclude by summarizing the main contributions of our implementation paper and highlighting the significance of our collaborative filtering approach.

II. RELATED WORK

Several research studies have proposed various models and techniques to enhance the performance of movie recommendation systems. Mukesh Kumar Kharita [1] introduced a similarity model to predict unknown movie ratings and recommend movies based on the user's past ratings and likings. This approach is particularly useful when



the underlying metadata of the data is unknown. Dharmendra Pathak [2] proposed a hybrid model that combined collaborative filtering and content-based filtering, thereby overcoming the limitations of both approaches. However, this model required semi-labelled data to start with. Ching-Seh (Mike) Wu [3] presented a novel recommendation model that uses a similarity matrix to train without any co-relation equation, making it faster than other models. Bagher Rahimpour Cami [4] et al. employed a Bayesian nonparametric framework to capture the temporal preferences of a user, providing significant performance benefits compared to other approaches. Noor Ifada [5] et al. analyzed and compared the performance of collaborative filtering and hybrid-based approaches for generating movie recommendations. Jochen Nessel [6] proposed a content-based filtering approach that recommends movies based on the genres, runtime, artists involved, and other hard-coded factors. However, this method requires labelled data to start with, which is not always efficient. Yajie Hu, Ziqi Wang [7] et al. aimed at finding stars or films that are similar in performance style, genre, and film style to the queried stars or films, recommending them to users. Sajal Halder [8] suggested a system that recommends a set of movies based on the user's interest or the popularity of the movies. Despite the numerous proposed movie recommendation systems, most of them struggle to recommend movies efficiently to either existing or new users.

III. METHODOLOGY

The methodology involves analyzing user-item interaction data to identify similarities between users and recommend items that similar users have preferred in the past. In the case of movie recommendation, the methodology involves finding similar users who have similar movie preferences and recommending movies that these similar users have enjoyed. Collaborative filtering can be implemented using two approaches - memory-based and model-based - with each approach having its own advantages and disadvantages.

While memory-based methods rely on the entire dataset and require high computational resources, model-based methods use statistical techniques to build models that can provide better recommendations with less computational requirements. However, model-based methods require a well-defined training set and may not work well with sparse data. One used in this project is user-based memory based collaborative filtering system. User-based memory-based collaborative filtering makes recommendations by finding users who have similar ratings to a target user and then recommending items that those similar users have rated highly. The similarity between users is calculated using metrics such as Pearson correlation, cosine similarity, or Euclidean distance

IV. MATHEMATICAL MODEL

Mathematical model for the Pearson correlation coefficient used in the following collaborative filtering. Pearson's correlation coefficient (also known as Pearson's R) is a statistical measure that quantifies the linear relationship between two variables. In the context of collaborative filtering, it is used to determine the similarity between two users or two items based on their ratings.

Let R be a matrix of ratings where $R(i,j)$ represents the rating of user i for item j . Let $S(i)$ be the set of items for which user i has provided a rating, and $S(j)$ be the set of items for which user j has provided a rating. Let μ be the mean rating across all users and items, and let $\sigma(i)$ and $\sigma(j)$ be the standard deviation of the ratings provided by user i and user j , respectively.

The Pearson correlation coefficient between users i and j can be calculated as follows:

$$r(i, j) = \frac{\sum_{k \in S(i) \cap S(j)} (R(i,k) - \mu(i))(R(j,k) - \mu(j))}{\sigma(i)\sigma(j)|S(i) \cap S(j)|}$$

Here, the sum is taken over all items k for which both users i and j have provided a rating. The denominator normalizes the correlation coefficient to account for differences in the scale of ratings and the sparsity of the data.



This correlation coefficient can be used to predict the rating of user i for an item j by taking a weighted average of the ratings provided by other users who are similar to user i , where the weights are given by the corresponding correlation coefficients.

V. IMPLEMENTATION DETAILS

The implementation of the collaborative filtering movie recommendation engine was carried out using Python programming language. The Flask web framework was utilized to create a user interface for the engine, while the Pandas library was employed to handle data and perform computations. The dataset of movie ratings provided by users was used as the data source for the engine. The data was loaded into a Pandas data frame for further processing. Prior to running the collaborative filtering algorithm, data pre-processing was performed to remove any missing values and to convert the data into a suitable format for the algorithm.

The collaborative filtering algorithm was implemented using the Pearson correlation coefficient as the similarity measure between users. This algorithm enabled the engine to suggest movies to users based on the ratings of other users who had similar tastes. A Flask application was developed to provide a user-friendly interface for the movie recommendation engine. The user could input their movie ratings, and the engine would generate a list of movie recommendations based on their input.

The application was deployed on a local server for testing purposes. The implementation of the collaborative filtering movie recommendation engine using Flask and Pandas libraries proved to be successful. The engine was able to provide accurate movie recommendations based on the user's input. The use of collaborative filtering algorithm made the engine more efficient in generating recommendations, as it used the opinions of other users with similar tastes. Overall, this implementation of the movie recommendation engine provides a promising solution to the problem of personalized movie recommendations.

VI. CONCLUSION

It is widely acknowledged that personalized content will play a significant role in the future, with providers and customers collaborating to determine the best form and extent of personalization. This will determine which methods are effective and which are not suitable for certain types of content.

Recommender systems have become ubiquitous and are used to recommend a wide range of products, services, and information. The field of recommender systems is supported by a research community that focuses on developing creative interaction ideas, powerful algorithms, and rigorous experiments. However, there are still many challenges in the field, particularly in the interface between research and commercial practice.

Collaborative filtering is a method that analyzes user behavior patterns to predict individual interests without requiring explicit user input. The method assumes that users who have similar preferences for similar products will agree on other products. This approach requires collecting information about the behavior and preferences of as many users as possible to make accurate predictions. However, a significant challenge with collaborative filtering is that it has a high latency for new users and new items entering the system.

Therefore, collaborative filters are considered a type of artificial intelligence and a learning system, and there is a need for future research to address these challenges.

REFERENCES

- [1] M. K. Kharita, A. Kumar and P. Singh, "Item-Based Collaborative Filtering in Movie Recommendation in Real time," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), 2018, pp. 340-342, doi: 10.1109/ICSCCC.2018.8703362.
- [2] D. Pathak, S. Matharia and C. N. S. Murthy, "ORBIT: Hybrid movie recommendation engine," 2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), 2013, pp. 19-24, doi: 10.1109/ICE-CCN.2013.6528589.



- [3] C. -S. M. Wu, D. Garg and U. Bhandary, "Movie Recommendation System Using Collaborative Filtering," 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), 2018, pp. 11-15, doi: 10.1109/ICSESS.2018.8663822.
- [4] B. R. Cami and M. R. M. Brujerdi, "Using Hybrid Trust Model for Handling Inaccurate Resource," 2009 International Conference on Availability, Reliability and Security, 2009, pp. 630- 634, doi: 10.1109/ARES.2009.164.
- [5] N. Ifada, T. F. Rahman and M. K. Sophan, "Comparing Collaborative Filtering and Hybrid based Approaches for Movie Recommendation," 2020 6th Information Technology International Seminar (ITIS), 2020, pp. 219-223, doi: 10.1109/ITIS50118.2020.9321014.
- [6] J. Nessel and B. Cimpa, "The MovieOracle - Content Based Movie Recommendations," 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 2011, pp. 361-364, doi: 10.1109/WI-IAT.2011.236.
- [7] Y. Hu, Z. Wang, W. Wu, J. Guo and M. Zhang, "Recommendation for Movies and Stars Using YAGO and IMDB," 2010 12th International Asia-Pacific Web Conference, 2010, pp. 123-129, doi: 10.1109/APWeb.2010.51.
- [8] Halder, S., Sarkar, A. M. J., & Lee, Y.-K. (2012). Movie Recommendation System Based on Movie Swarm. 2012 Second International Conference on Cloud and Green Computing. doi:10.1109/cgc.2012.121

BIOGRAPHY

Kishor Sonawane is a B.E student in the Computer Engineering Department, at ShrimatiKashibaiNavale College of Engineering, SPPU. He is in the final year of his Bachelor's in Computer Engineering and will pass out in 2023. His research interests are Machine Learning, Data Science, Data Structures and Algorithms, and Web development.

Vedantika Wadkar is a B.E student in the Computer Engineering Department, at ShrimatiKashibaiNavale College of Engineering, SPPU. She is in the final year of his Bachelor's in Computer Engineering and will pass out in 2023. Her research interests are Design and analysis of Algorithms, Data Analytics, DBMS, Front End Web development.

Khushi Tiwari is a B.E student in the Computer Engineering Department, at ShrimatiKashibaiNavale College of Engineering, SPPU. She is in the final year of his Bachelor's in Computer Engineering and will pass out in 2023. Her research interests are Web 3.0, Blockchain Technology, DBMS, Full stack Web development, etc.

Rohit Prasad is a B.E student in the Computer Engineering Department, at ShrimatiKashibaiNavale College of Engineering, SPPU. He is in the final year of his Bachelor's in Computer Engineering and will pass out in 2023. His research interests are Machine Learning, Natural Language Processing, Artificial Intelligence, Algorithms, Data science, etc.



Impact Factor: 8.423



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details