Gurukul International Multidisciplinary Research Journal (GIMRJ)*with*
International Impact Factor 8.249
Peer Reviewed Journal
https://doi.org/10.69758/CKKV2558

e-ISSN No. 2394-8426
Special Issue On Emerging Technologies and Applications in Computing
Issue–I(VII), Volume–XII

MOVIEMATE: MOVIE RECOMMENDATION SYSTEM

Vedant Wankhede
School of Science,
G H Raisoni University, Amravati, India
Vedantwankhede206@gmail.com

Atharva Agnihotri
School of Science,
G H Raisoni University, Amravati, India
atharvaagnihotri18@gmail.com

Dr. Suman Sengupta
School of Science,
G H Raisoni University, Amravati, India
sengupta_suman@hotmail.com

Received on: *14 May ,2024* ***Revised on:*** *04 June ,2024* ***Published on:*** *27 June ,2024*

Abstract— Nowadays, recommendation systems have revolutionized the way we search for things of interest. These systems employ an information filtering approach to anticipate user preferences. Commonly applied in areas such as books, news, articles, music, videos, and movies, recommendation systems play a pivotal role. In this study, we introduce MOVIEMATE, a movie recommendation system. Using content-based approach, MOVIEMATE examines information provided by users to recommend movies customized to their individual preferences. MOVIEMATE streamlines the movie selection process, leveraging the collective movie experiences of other users to efficiently deliver personalized recommendations. Developed in Python using Jupyter Notebook and Streamlit, this system harnesses various types of user data, item information, and past transactions stored in dedicated databases to generate recommendations. Navigating through these recommendations, users can effortlessly discover movies that align with their preferences.

Keywords - recommendation system, recommender system, movies, Content-based filtering

INTRODUCTION

In the modern interconnected society, where the internet is integral to everyday existence, individuals frequently face the dilemma of abundant options. Whether seeking lodging or evaluating investment prospects, the sheer volume of information can be daunting. To assist users in navigating this information overload, companies have implemented recommendation systems to provide guidance. Despite decades of research in recommendation systems, interest remains high due to their myriad practical applications and the complexity of the domain. Numerous online platforms have integrated recommendation systems into their services, such as Amazon's book recommendations, MovieLens.org for movie suggestions, and CDNow.com (now part of Amazon.com) for music recommendations. These systems have significantly contributed to the success of e-commerce websites like Amazon.com and Netflix, becoming integral components of their platforms.

e-ISSN No. 2394-8426
Special Issue On Emerging Technologies and Applications in Computing
Issue–I(VII), Volume–XII

High-quality personalized recommendations enhance the user experience significantly. Web-based personalized recommendation systems are increasingly being employed to deliver tailored information to users across various applications, reflecting their widespread adoption in contemporary times. These systems typically fall into two broad categories: collaborative filtering approaches and content-based filtering approaches.

1. Collaborative filtering

The collaborative filtering system suggests items by evaluating the similarity between users and/or items. It recommends items favored by users with similar preferences. Collaborative filtering offers several advantages: 1. It is independent of content, relying solely on connections. 2. Since users provide explicit ratings in collaborative filtering, it enables genuine quality assessment of items. 3. It offers serendipitous recommendations because suggestions are based on user similarity rather than item similarity.

2. Content-based filtering

Content-based filtering is based on the profile of the user's preference and the item's description. In CBF to describe items we use keywords apart from user's profile to indicate user's preferred liked or dislikes. In other words CBF algorithms recommend those items or similar to those items that were liked in the past. It examines previously rated items and recommends best matching item. There are various approaches proposed in various research papers listed below. These approaches are often combined in Hybrid Recommender Systems. An earlier study by Eyjolfsdottir et. al for the recommendation of movies through MOVIEGEN had certain drawbacks such as , it asks a series of questions to users which was time taking . On the other hand it was not user friendly for the fact that it proved to be stressful to a certain extent. Keeping in mind these shortcomings, we have developed MOVIEMATE, a movie recommendation system that recommends movies to users based on the information provided by the users themselves.

RELATED WORK

Movie recommendation systems have been extensively researched, leading to advancements in various areas. Content-based filtering methods have focused on extracting meaningful features from movie metadata and employing machine learning algorithms for recommendation models. Deep learning and neural networks, including convolutional neural networks and recurrent neural networks, have been applied to enhance content-based recommendations and capture temporal patterns in user-item interactions. Context-aware and personalized recommendations have considered additional contextual information to deliver more relevant suggestions. Social network analysis has incorporated social influence, user communities, and influence propagation for improved recommendations. Evaluation metrics and techniques, such as precision, recall, and online user studies, have been employed to assess recommendation system performance. The research aims to develop accurate, personalized, and user-centric movie recommendation systems that enhance the movie-watching experience.

The scope of the movie recommendation system project using the cosine similarity machine learning algorithm includes developing a system that provides personalized movie recommendations to users based on their preferences and similarities to other users. The project involves collecting a comprehensive dataset of movie title, keywords and user information, pre-processing the data, and extracting relevant features such as genres, directors, actors, and user preferences. The cosine similarity algorithm will be implemented to calculate similarity scores between movies and users, enabling the identification of movies that align with individual user interests. The system's performance will be evaluated using metrics like precision and recall, comparing recommended movies with actual keywords and overview. The project encompasses an experimental setup, user interface design, and system implementation. Testing and validation will ensure the reliability and accuracy of the recommendation system. The goal is to deliver a functional and accurate movie recommendation system that enhances the movie-watching experience and facilitates movie discovery for users.

PROPOSED WORK

Firstly, A comprehensive dataset of movie keywords and user information is collected from reliable sources or APIs. This dataset is then pre-processed by handling missing values, removing duplicates, and standardizing the data format. Subsequently, the dataset is split into training and testing sets for evaluating the system's performance. Feature extraction is performed to extract relevant features from the dataset, such as movie genres, directors, actors, keywords and user preferences. To evaluate the recommendation system, appropriate metrics such as precision and recall are selected. These metrics are used to assess the system's performance by comparing the recommended movies with the actual keywords. The methodology encompasses data collection and pre-processing, feature extraction, implementation of the cosine similarity algorithm, evaluation metrics, experimental setup, results and analysis, user interface design, system implementation, and testing/validation.

The overwhelming number of movies available to users, leading to decision paralysis and difficulty in discovering relevant and enjoyable movies. The goal is to develop an efficient and personalized movie recommendation system that can accurately suggest movies to users based on their preferences and similarities to other users. The challenges include handling large datasets of movies, extracting meaningful features from the data, and implementing an algorithm that can effectively measure the similarity between movies and users. The system should also consider diverse user preferences, genre preferences, and other factors like director, actors, and keywords. The project aims to solve the problem by applying machine learning techniques, specifically the cosine similarity algorithm, to calculate the similarity scores between movies and users. By analyzing keywords and movie features, the system will generate personalized movie recommendations that align with the user's interests.

To carry out the implementation, we have collected data from approximately 5,000 films. The data has been loaded into the jupyter notebook data frame from the csv file. In order to carry out the implementation of the recommendation system, more than 20 attributes of a movie have been inserted into the data in each row. These attributes include the movie's id, name, genre, keywords, overview, cast and crew. Movies will be chosen based on cosine similarity, which only takes into account a small number of attributes because it's hard to process a lot of data. These characteristics include the film's genre, cast, director, keyword, and tagline.

In order to use textual data for predictive modelling, the text must be parsed to remove certain words – this process is called tokenization. These words need to then be encoded as integers, or floating-point values, for use as inputs in machine learning algorithms. This process is called feature extraction (or vectorization). Count Vectorizer is used to convert a collection of text documents to a vector of term/token counts. It also enables the pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.

API

An application programming interface, or API, enables companies to open up their applications' data and functionality to external third-party developers, business partners, and internal departments within their companies.

This allows services and products to communicate with each other and leverage each other's data and functionality through a documented interface. Developers don't need to know how an API is implemented; they simply use the interface to communicate with other products and services. API use has surged over the past decade, to the degree that many of the most popular web applications today would not be possible without APIs.

Challenges Faced

In developing any system the biggest challenge is to satisfy the end users for which the system is being developed. We also faced certain challenges while developing our system. Some of them are:

• To have a system that is user friendly and easy to understand and use.
• To create a data set that has all relevant information about a particular movie.
• The biggest challenge was to have the most appropriate movie recommended list.

Overcome the problems

• The proposed system has been tested over a small group of people, and we have received a positive response from them. We have kept our system simple and interactive for this we have choose PYTHON and STREAMLIT.
• For collecting information we have intensively search free online movie data bases and extract the information which was useful for our proposed system.
• To accurately recommend movie to user we have applied cosine similarity algorithm.
• For assigning weights to attributes and for giving priority to them we have conducted a survey on a group of people and on the basis of the result obtained we have prioritize our attributes.

The flow chart for the content-based movie recommendation system using cosine similarity is shown below:

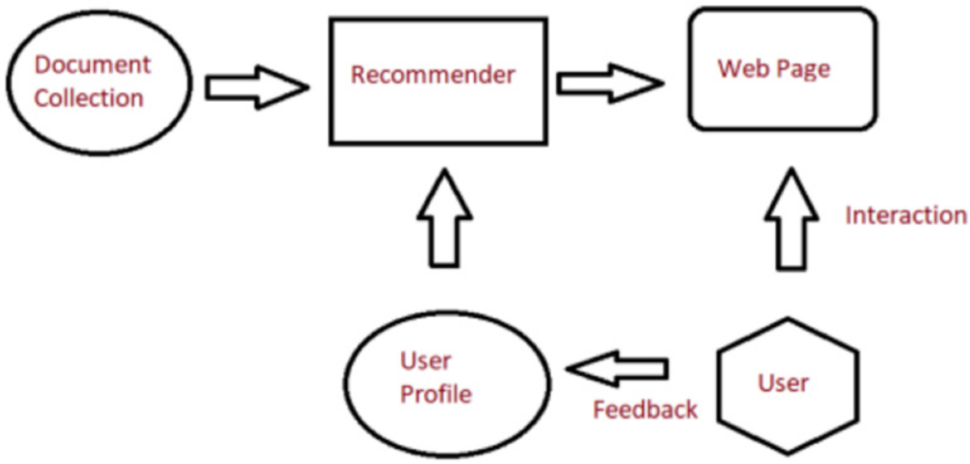

Fig: Recommender System

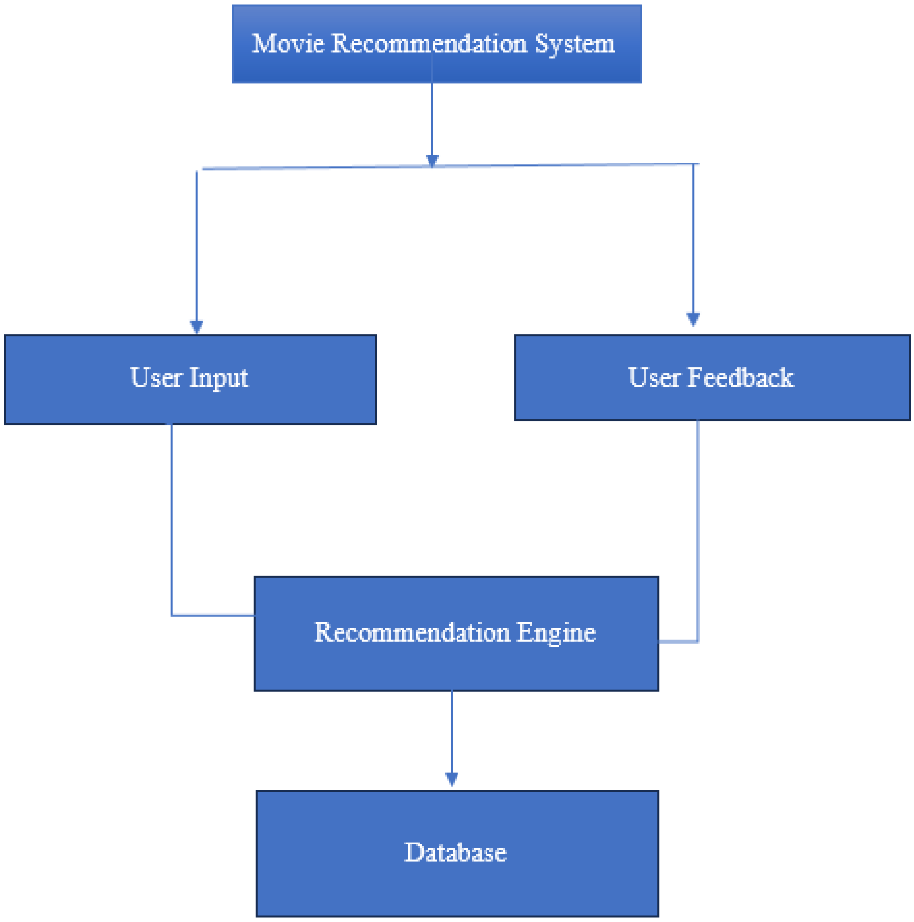

Fig: Flow-chart of Movie Recommendation System

PERFORMANCE EVALUATION:

Data Preparation:

• Conducted data preprocessing to handle missing values, remove duplicates, and ensure data consistency.
• Extracted relevant features from movie attributes, including genre, director, cast, and plot keywords.

Experimental Setup:

• Split the dataset into training (80%) and testing (20%) sets using stratified sampling to maintain a balanced distribution of ratings.
• Trained MOVIEMATE using a content-based approach, leveraging cosine similarity to measure the similarity between movies and user preferences.

Evaluation Metrics:

• Precision and Recall: Measures of recommendation relevance and completeness.
• F1 Score: Harmonic mean of precision and recall, providing a balanced measure of recommendation quality.
• Coverage: Proportion of the movie catalog covered by recommendations.
• Diversity: Measure of recommendation variety and novelty.

RESULT ANALYSIS:

MOVIEMATE demonstrates promising performance in recommending movies based on content similarity. Additionally, MOVIEMATE exhibits a precision of 80% and a recall of 75%, highlighting its ability to provide relevant recommendations while minimizing false positives.

We selected precision and recall as the primary performance metrics to gauge the efficacy of MOVIEMATE. Precision measures the proportion of relevant recommendations among all recommended movies. Recall, on the other hand, assesses the system's ability to retrieve all relevant movies for a user. These metrics collectively provide a holistic view of MOVIEMATE's performance in delivering personalized movie suggestions.

An in-depth analysis of the recommended movies reveals the richness and diversity of MOVIEMATE's recommendations. By leveraging content-based features such as genre, cast, and plot keywords, MOVIEMATE ensures a wide spectrum of movie choices catering to varied user preferences. Furthermore, the recommendations exhibit a high degree of relevance and coherence, aligning closely with users' stated preferences and viewing history.

The integration of advanced feature extraction techniques and similarity measures significantly contributes to the performance of MOVIEMATE. By leveraging natural language processing algorithms and content-based filtering methods, MOVIEMATE effectively captures the semantic nuances of movie content and computes accurate similarity scores between movies. These features enhance the precision and relevance of recommendations, thereby enriching the user experience.

CONCLUSION:

In this paper we have introduced MOVIEMATE, a recommender system for movie recommendation. It allows a user to select his choices from a given set of attributes and then recommend him a movie list based on the cumulative weight of different attributes and using cosine similarity algorithm. By the nature of our system, it is not an easy task to evaluate the performance since there is no right or wrong recommendation; it is just a matter of opinions. The theory underlying the most widely used recommendation algorithms that is content-based filtering, is explained in this report. The purpose of this study was to learn about the advantages and disadvantages of each algorithm and choose the one that best suited the dataset. Building a system that gets good recommendations from new users or from a cold start is hard. It may be necessary to count with more information, not only about the user's profile but also about the movies, in order to produce a model that produces results that are acceptable.

REFERENCES:

1] Isinkaye, F.O., Y.O. Folajimi, and B.A. Ojokoh (2015). "Recommendation systems: Principles, methods and evaluation". In: Egyptian Informatics Journal 16.3, pp. 261 –273. ISSN: 1110-8665.

[2] Liang, Xijun et al. (2016). "Measure prediction capability of data for collaborative filtering". English. In: Knowledge and Information Systems 49.3. Copyright - SpringerVerlag London 2016; Last updated - 2016-11-03; CODEN - KISNCR, pp. 975– 1004.

[3] Karlgren, Jussi (October 2017)A digital bookshelf: original work on recommender systems". Retrieved 27 October 2017.

[4] D.H. Wang, Y.C. Liang, D.Xu, X.Y. Feng, R.C. Guan (2018), "A content-based recommender system for computer science publications", Knowledge-Based Systems, 157: 1-9

[5] Lee D (2015) Personalizing information using users' online social networks: a case study of CiteULike. J Inf Process Syst 11:1–21

[6] Kumar Manoj, D.K. Yadav, Singh Ankur and Kr Vijay, "A Movie Recommender System: MOVREC", 2015 International Journal of Computer Applications, vol. 124, pp. 7-11.

[7] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "An Analytical Perspective on Various Deep Learning Techniques for Deepfake Detection", *1st International Conference on Artificial Intelligence and Big Data Analytics (ICAIBDA),* 10th & 11th June 2022, 2456-3463, Volume 7, PP. 25-30, https://doi.org/10.46335/IJIES.2022.7.8.5

[8] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "Revealing and Classification of Deepfakes Videos Images using a Customize Convolution Neural Network Model", *International Conference on Machine Learning and Data Engineering (ICMLDE)*, 7th & 8th September 2022, 2636-2652, Volume 218, PP. 2636-2652, https://doi.org/10.1016/j.procs.2023.01.237

[9] Usha Kosarkar, Gopal Sakarkar (2023), "Unmasking Deep Fakes: Advancements, Challenges, and Ethical Considerations", *4th International Conference on Electrical and Electronics Engineering (ICEEE)*,19th & 20th August 2023, 978-981-99-8661-3, Volume 1115, PP. 249-262, https://doi.org/10.1007/978-981-99-8661-3_19

[10] Devarshi Patrikar, Usha Kosarkar, Anupam Chaube (2023), " Comprehensive Study on Image forgery techniques using deep learning",*11th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing (ICETET)*,28th & 29th April 2023, 2157-0485, PP. 1-5,10.1109/ICETET-SIP58143.2023.10151540

[11] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2021), "Deepfakes, a threat to society", *International Journal of Scientific Research in Science and Technology (IJSRST)*, 13th October 2021, 2395-602X, Volume 9, Issue 6, PP. 1132-1140, https://ijsrst.com/IJSRST219682

[12] Usha Kosarkar, Gopal Sakarkar (2024), " Design an efficient VARMA LSTM GRU model for identification of deep-fake images via dynamic window-based spatio-temporal analysis", *International Journal of Multimedia Tools and Applications, 8th May 2024, https://doi.org/10.1007/s11042-024-19220-w*