IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 8, May 2023

Movie Recommendation System Using Machine Learning

D. Raghu Vamshi[1], P. Vamshi[2], K. Hemanth Kumar[3], Mrs. Yellamma[4], B. Vasundara Devi[5]

Department of Computer Science and Engineering[1,2,3,4,5]
Sreenidhi Institute of Science and Technology, Hyderabad, India

Abstract: *A movie recommendation system or a movie recommender system, is an ML- based approach to filtering or predicting the users film preferences based on their past choices and behaviour. It's an advanced filtration mechanism that predicts the possible movie choices of the concerned user and their preferences towards a domain-specific item aka a movie. A recommendation system usually filters the given data using various methodologies such as collaborative filtering and the use of clustering algorithms, to suggest relevant ones to the user. A recommendation system such as this is used in day-to-day life in many areas like movies, books, music, news, shopping. The primary goal of a movie recommendation system is to filter and predict only those movies that a corresponding user is most likely to want to watch.*

Keywords*:* Recommendation System, Sentiment Analysis, Web Scraping

I. INTRODUCTION

Movie recommendation systems are a type of machine learning application that suggest movies to users based on their preferences and historical viewing behaviour. These systems use data on a user's past movie choices, as well as data on the characteristics of different movies, to create personalized recommendations that are tailored to the user's interests.

There are two main types of movie recommendation systems: content-based and collaborative filtering. Content-based systems recommend movies that are similar in terms of their attributes, such as genre, actors, and plot. Collaborative filtering systems recommend movies based on the preferences of other users who have similar viewing histories.

To build a movie recommendation system, machine learning algorithms are trained on large datasets of movie ratings and user preferences. The algorithms use this data to learn patterns and relationships between movies and users, which they can then use to make predictions about which movies a user is likely to enjoy.

Movie recommendation systems have become increasingly popular in recent years, as the amount of available movie content has grown exponentially. By using machine learning algorithms to suggest movies to users, these systems can help users discover new movies that they might not have otherwise found, while also helping movie studios and streaming services increase engagement and revenue.

II. EXISTING SYSTEM

Before existence the recommendation system, individuals would physically choose movies to watch from movie libraries. They either had to read the user's reviews and based on the review they would select a movie or had to randomly select a movie. This procedure isn't feasible, as there is an enormous number of spectators with a unique preference for movies. Hence many recommendation systems have been developed over the past decade. These systems use different approaches like a collaborative approach, a content-based approach.

Content-based approach is limited to a single user, where the user's past history and ratings are used for providing recommendations. There are a number of methodologies introduced to implement this recommendation system which includes various fields of Data Mining, Clustering and Bayesian Network methodology.

III. PROPOSED SYSTEM

Proposed recommendation system uses a combination of collaborative filtering and content-based approach to generate movie recommendations.

DOI: 10.48175/IJARSCT-10288

A Similarity value is calculated for implementing the collaborative filtering technique to select a set of users similar to the current user. Also, the users that are selected must be trustworthy. Hence a method is created to increase or decrease the significance of a particular user or item.

The proposed system uses ratings from other similar users and the current users own past history to make a suitable recommendation

Advantages

- Can make out – of – the box recommendations
- It is more dynamic i.e., can adjust to change in people's likes and dislike

IV. SYSTEM ARCHITECHTURE

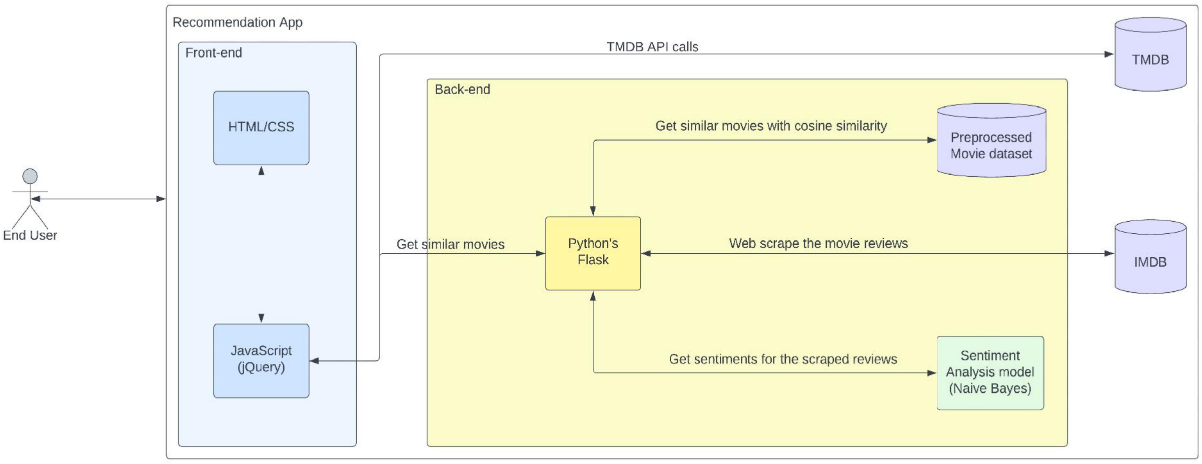
Recommendation App
Front-end
HTML/CSS
JavaScript (jQuery)
End User
TMDB API calls
TMDB
Back-end
Get similar movies with cosine similarity
Preprocessed Movie dataset
Get similar movies
Python's Flask
Web scrape the movie reviews
IMDB
Get sentiments for the scraped reviews
Sentiment Analysis model (Naive Bayes)

Figure 1: System Architecture

The above diagram is the system architecture for the movie recommendation system. The movie recommendation system uses sentiment analysis follows a client-server architecture, where the client is the user-facing application and the server is the backend that performs the recommendation and sentiment analysis tasks. Naïve Bayes classifier is used to implement the Sentiment analysis module of the system.

The client application is a web application that allows users to search for movies and receive personalized recommendations based on their previous viewing history and sentiment analysis of their reviews or ratings. The user's interactions with the client application are sent to the server for processing. Python's Flask framework is used to deploy the client user interface.HTML, CSS and JavaScript are used to create the UI/UX of the system in combination with Flask. The reviews which are used in sentiment analysis are obtained from IMDB.

The server is responsible for storing the movie data and user profiles, analysing the sentiment of user reviews or ratings using natural language processing (NLP) techniques, and generating movie recommendations based on the sentiment analysis and user preferences. The server also uses machine learning algorithms to improve the recommendation accuracy over time

The system architecture typically includes a database for storing the movie data, a recommendation engine for generating personalized recommendations, an NLP module for performing sentiment analysis, and a machine learning module for improving the recommendation accuracy.

In addition to the client-server architecture, a movie recommendation system also includes an external API (TMDB) for retrieving movie data and reviews, as well as cloud-based platforms for scalability and performance optimization.

The System also includes a database containing a pre-processed model which contains the similarity score of each movie respective of the other. The dataset used to create this model contains the meta-data of the movie such as cast, run time, genre, and type of movie (Live action, Animation).

DOI: 10.48175/IJARSCT-10288

V. UML DIAGRAMS

Use Case Diagram:

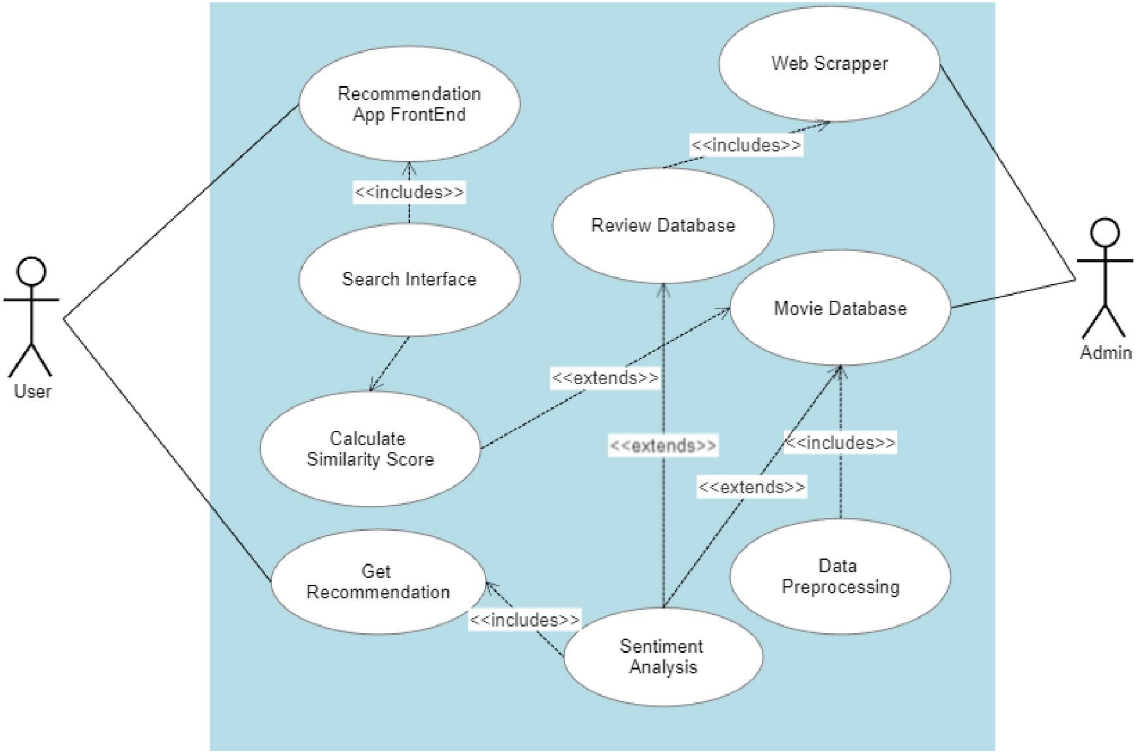

Figure 2: Use Case Diagram

Activity Diagram:

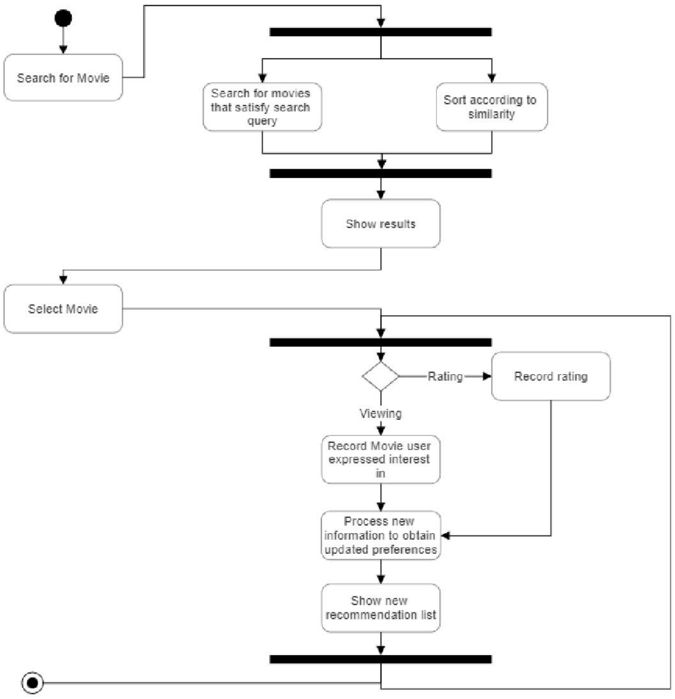
Search for Movie
Search for movies that satisfy search query
Sort according to similarity
Show results
Select Movie
Rating
Record rating
Viewing
Record Movie user expressed interest in
Process new information to obtain updated preferences
Show new recommendation list

Figure 3: Activity Diagram

DOI: 10.48175/IJARSCT-10288

VI. METHODS USED IN RECOMMENDATION

Naïve Bayes Classifier

Naive Bayes sentiment analysis is a popular method in Natural Language Processing (NLP), which leverages the principles of Bayes' theorem and statistical assumptions of independence. It's a probabilistic machine learning algorithm that's commonly used for text classification tasks, including sentiment analysis.

Sentiment analysis, also known as opinion mining, is a subfield of NLP that involves determining the emotional tone behind words to understand the attitudes, sentiments, and emotions expressed in a given body of text. This can be particularly useful for understanding customer reviews, social media comments, and other forms of user-generated content.

The Naive Bayes classifier is an excellent choice for such tasks due to its simplicity, efficiency, and effectiveness, particularly in dealing with large feature spaces, which is common in text data.

The term "naive" in Naive Bayes comes from the assumption that the features in the data are mutually independent, which means that the presence of a particular feature does not affect the presence of another. This assumption is naive because it is rarely true in real-world situations. However, despite this simplification, the Naive Bayes classifier performs remarkably well in many scenarios, especially in text classification tasks like sentiment analysis.

When applied to sentiment analysis, the Naive Bayes algorithm treats text as a 'bag of words'. This means that the text is transformed into a set of features where each feature represents a word. The order of the words is not considered in this model.

The algorithm works by calculating the probabilities of each class (like positive or negative sentiment) given the features (words), and then predicting the class with the highest probability. These probabilities are estimated based on the frequencies of the words in the training data.

In a movie review dataset, the algorithm might learn that reviews containing words like 'excellent' and 'outstanding' are often labeled as positive, while reviews containing words like 'boring' and 'tedious' are often labeled as negative. Then, when given a new review to classify, it can calculate the probabilities of the review being positive or negative based on the words it contains, and predict the class with the highest probability.

Despite its simplicity, Naive Bayes can often outperform more sophisticated machine learning algorithms, especially when dealing with high-dimensional, sparse data that is common in text classification tasks. Furthermore, it's highly scalable, making it suitable for large datasets, and it's relatively easy to understand and implement. However, its performance can be affected if the assumption of feature independence is violated, or if the model encounters words in the test data that it didn't see in the training data (a problem known as zero frequency), although there are techniques to handle these issues.

VII. CONCLUSION

In conclusion, movie recommendation systems have emerged as a powerful tool in the entertainment industry, utilizing machine learning algorithms to provide personalized movie suggestions to users based on their preferences and viewing history. These systems not only enhance the user experience by helping them discover new and engaging content but also offer valuable insights to movie studios and streaming services regarding audience preferences and trends.

Our movie recommendation system uses a combination of content-based and collaborative filtering approaches, leveraging a Similarity Score and sentiment analysis from user reviews to generate accurate and relevant recommendations. By utilizing data from The Movie Database (TMDB) and employing Python's Flask Framework for the front-end interface, our system offers a comprehensive solution for both users and industry stakeholders.

As the entertainment landscape continues to evolve and expand, movie recommendation systems will play an increasingly important role in connecting users with content that resonates with their interests. By continually refining the algorithms and incorporating new data sources, these systems can drive user engagement and satisfaction, ultimately contributing to the success and growth of the movie industry.

DOI: 10.48175/IJARSCT-10288

REFERENCES

[1]. N. K. Rao, N. P. Challa, S. S. Chakravarthi and R. Ranjana,"Movie Recommendation System using Machine Learning," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, pp. 711-716, doi: 10.1109/ICIRCA54612.2022.9985512.

[2]. Furtado, F., & Singh, A. (2020). Movie recommendation system using machine learning. International journal of research in industrial engineering, 9(1), 84-98.

[3]. Choudhury, SasmitaSubhadarsinee, Sachi Nandan Mohanty, and Alok Kumar Jagadev. "Multimodal trust based recommender system with machine learning approaches for movie recommendation." International Journal of Information Technology 13 (2021): 475-482.

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