Movie Recommendation Engine with Sentiment Analysis Using AJAX Request

[1]Prof. Satish Chadokar, [2]Naman Jain, [3]Ayush Thakre

[1-3]Dept. of Computer Science & Engineering,
SHRI BALAJI INSTITUTE OF TECHNOLOGY AND MANAGEMENT, BETUL, INDIA

***Abstract*: As Artificial Intelligence and Machine Learning have grown at a rapid pace in recent years, so has the amount of data on the internet. As a result, consumers find it difficult to select the precise information they desire, and learners find it difficult to suggest users exactly what they require. Here, recommendation systems come into play to point consumers in the direction of the content based on their preferences. This study aims to explain the creation and implementation of Movie Recommendation Systems in the Context of Recommendation of Movies and TV Shows on Online Streaming Platforms.**

Movie proposal in Web climate is fundamentally significant for Internet clients. It completes thorough accumulation of client's inclinations, surveys, and feelings to help them find appropriate motion pictures advantageously. Recommendation systems (RSs) have garnered immense interest for applications in e-commerce and digital media. Recommendation System is a smart system that offers pertinent information regarding the decisions a user has made. Collaborative filtering and content-based filtering are two of its useful techniques. this paper is aimed to explain making and implementation of Movie Recommendation Systems Using Machine Learning Algorithms, Sentiment Analysis and Cosine Similarity.

***Index Terms*: Recommendation system, Content-Based filtering, Sentiment Analysis, Cosine Similarity.**

Published in IJIRMPS (E-ISSN: 2349-7300), Volume 11, Issue 1, January-February 2023

I. INTRODUCTION

Recommendation systems are mostly used to assist consumers in receiving results that are tailored to their interests. By using a machine learning algorithm that has a target based on the viewer search, recommendation systems can also be utilized as a filtering strategy to isolate the best result from a set of anticipated results. Web-based models are required to provide consumers with movie recommendations. As Movies can be categorized based on genres like drama, comedy, action, drama, animation, and thrillers. Another technique to categorize movies is to use metadata such as actors, year of release, language, or director. By using the user's prior search terms and viewing history, the vast majority of online video-streaming services today provide a variety of related television shows and movies to the user.

These movie recommendation systems assist users in finding the films or television shows of their choice, saving them time when making viewing decisions. The major objective while developing a movie recommendation system is to make it trustworthy and effective in order to give people choices that are specifically tailored to their needs. Basically, there are two categories of recommendation systems: content-based filtering (CBF) and collaborative filtering (CF) [1].

As humans, we have a tendency to make judgements based on facts and data that we already have stored in our brain from searching the web, and this behavior gives rise to the notion of Collaborative Filtering. When two users' ratings are similar, they are regarded to be like-minded. While in the instance of content-based filtering, suggestions for results are made based on how contextually related two items are. In the modern world, the internet has become an essential component of daily life. Users also struggle with the issue of too much knowledge being available. Many applications, such as online shopping portals, netflix, amazon, have a recommendation system as their core feature. This paper focuses on movie recommendation system because movies are a significant source of leisure and entertainment in our lives. So, we propose a movie recommendation system by combining movie reviews datasets from various sources, such as Kaggle, where we take 5000+ different movies dataset till 2017 and Movie metadata dataset, as well as data from Wikipedia of movies from 2018 to 2022 and their reviews for sentiment analysis from TMDB website using TMDB API.

II. RELATED WORKS

Recommendation Systems are regarded as one of the most effective knowledge management engines, assisting us in filtering out irrelevant material and providing focused data based on feedback from previous data and similar data from user searches. Many recommendation systems have been presented to date, using various methodologies for computing such as Content-based filtering, Collaborative filtering, and hybrid models for recommendation. Sentiment Analysis is also used to increase the effectiveness of recommendations.

A. Efficient Bayesian Hierarchical User Modeling in Recommendation Systems

Providing users with individualised recommendations has been regarded as a serious challenge in the IR community since the 1970s. The techniques used to address this issue are classified into two types: content-based filtering and collaborative filtering. The scenario of content-based filtering is that a recommendation system screens a record stream and sends reports to the corresponding user that match a user profile. This study advances content-based recommendation research by improving the skill and feasibility of Bayesian hierarchical linear models, which have a strong theoretical background and great empirical performance on recommendation tasks [10][11]. This work makes no attempt to compare and contrast content-based and collaborative filtering or to handle the issue appropriately. According to this research, one compliments the other, and content-based filtering is especially useful for processing new reports/items with little or no user comments.

Due to the scarcity of information in IR applications, the generally employed EM method gradually merges data. This research proposes a novel quick learning method dubbed "Modified EM" for mastering a wide range of user profiles. This work uses a Bayesian hierarchical modelling technique to solve the cold start problem. Several experts have shown that this technique successfully balances shared and user-explicit data.

As a result, each user's initially inadequate performance is reduced. The EM algorithm is a popular boundary learning approach due to its ease of use and assembly assurance. However, a content-based recommendation system frequently works with reports in a very high-dimensional space, with each record represented by a relatively small vector. A comprehensive evaluation of the Expectation maximisation (EM) algorithm in this case reveals that, because to the limiting condition of the information elements, the EM method meets gradually. The "Modified EM algorithm," which is a better learning algorithm, is created by modifying the normal EM algorithm. Rather than calculating the mathematical answer for all of the user profile boundaries, For some element measurements, we infer the informative arrangement of the boundaries and employ the scientific arrangement rather than the mathematical arrangement determined at the E venture for those boundaries at the M advance. This drastically decreases processing at a single EM focus while also speeding up the learning algorithm. The Bayesian hierarchical modelling approach is becoming a prominent user profile learning approach due to its hypothetically argued capacity to help one user through information flow from diverse users via hyperpriors. This work investigated the drawbacks of the well-known EM-based learning methodology for Bayesian hierarchical linear models, as well as an improved learning method known as the Modified EM algorithm.

B. Content-Based Filtering for Movie Recommendations

The similarity of the products is what drives the recommendation algorithm. The general concept is that it is highly likely or recognized that if you enjoy one thing, you will also enjoy a related product. When a product's property or attributes can be easily ascertained, it works well. A content-based recommendation system for movies may make a movie recommendation based on information explicitly provided by the user, after which a user profile is created. This profile is also used to generate suggestions, which get better with time.

"Concepts of Term Frequency and Inverse Document Frequency are utilized for filtering systems and information retrieval" in content-based systems.[5] These words are mostly used to assess the significance of any movie. The frequency or count of times a word appears in a document is referred to as term frequency. In contrast, the total set of documents is represented by inverse document frequency. For example, let us say we type "The result of English Premier League" into a search engine. It is fairly certain that for any other search query, the word "The" will appear more frequently than "English premier league." As a result, "English premier league" is particularly significant in relation to "The." In these situations, weighting based on word frequency and inverse document frequency is utilized to assess importance [5].

III. Sentiment Analysis

Sentiment analysis is a subfield of natural language processing (NLP) that aims to automatically identify and extract subjective information from text data. It involves classifying texts or parts of texts as either positive, negative, or neutral with respect to a certain topic or subject. Sentiment analysis can be used to help businesses and organizations understand the sentiment of social media posts, customer reviews, and other forms of online or written communication. It can also be used to monitor brand reputation and customer satisfaction. There are various techniques and approaches for performing sentiment analysis, including rule-based methods, machine learning-based methods, and hybrid approaches that combine both. Some common challenges in sentiment analysis include dealing with irony, sarcasm, and negation, as well as handling subjectivity and cultural differences in language use. Sentiment analysis is the process of interpreting, processing, summarising, and reasoning emotional text. This approach was used in to calculate the polarity and trust of review sentences. Previously, Pang et al. classified the emotional polarity of movie commentary text into positive and negative using part of speech (POS), N-gram grammar (n-gram), and maximum entropy (ME). Turney investigated the polarity of text emotion using unsupervised learning in machine learning. The word pair was extracted from the feedback using tags, and the emotional polarity of the text was then determined using the Pointwise Mutual Knowledge and Information Retrieval (PMI-IR) method by comparing the similarity between terms in the text and words in the corpus[2].

The authors of suggested using the sentiment analysis model Valence Aware Dictionary and Sentiment Reasoner (VADER). Lexical features were coupled with five broad rules that combine grammatical and syntactic standards for expressing and emphasising emotion strength. Positive information is presumptively expected to have a positive impact, whereas negative information is presumptively expected to have a negative impact. Based on this discovery, some research conducted user reviews using sentiment analysis and determined the polarity of the outcomes. After that, users were guided to the movies with the most encouraging content[6][7].

IV. PROPOSED METHODOLOGY

A content-based recommender system is proposed in this paper whose results are improved using sentiment analysis, cosine similarity score and AJAX.

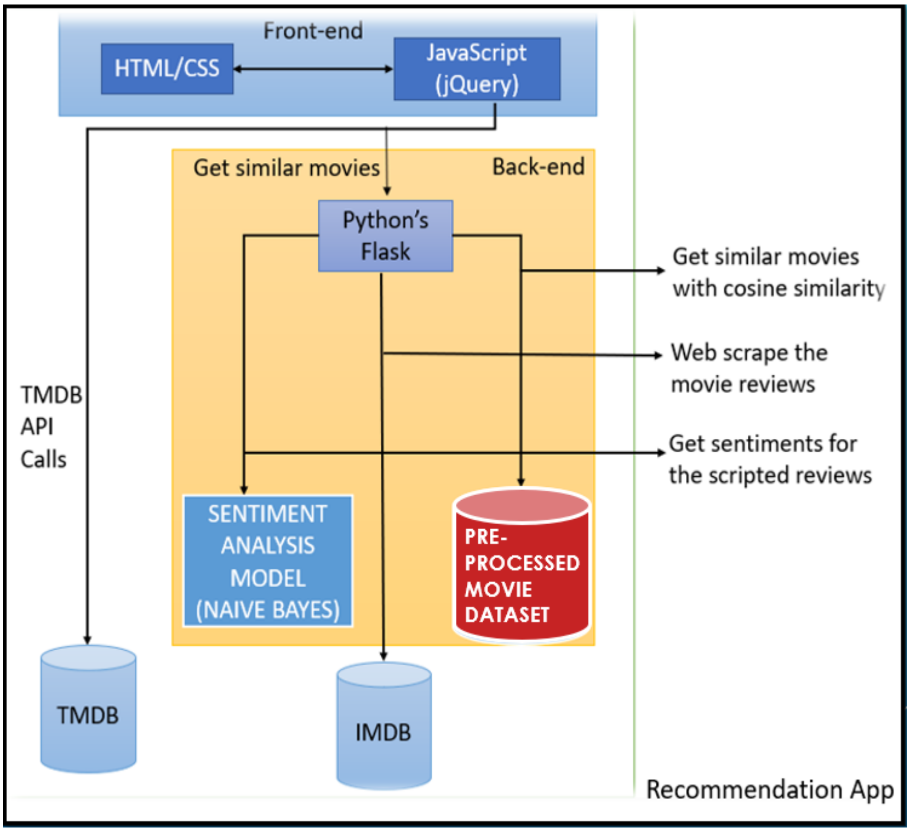

Fig 2. Proposed Movie recommendation system.

IV.1 Data Collection

The API is used to retrieve movie tricks (title, category, duration, rating, banner, and so on) from TMDB. Kaggle is the world's largest data science community, with strong tools that enable users to explore and share data sets, as well as test and construct models in a web-based data-science credit marketplace. Credit.csv from Kaggle gives information about the reaction to various movies and character reviews. The recommender system will be built using two databases: the first is a movie dataset of the top 5000 movies till 2017 and the movies metadata dataset, both of which are available on Kaggle, and the second is a Wikipedia database containing a list of movies from 2018 to 2022. TMDB (The Movie Database) uses an API to retrieve the movie's finer details, such as title, class, runtime, rating, banner, and so on. **The Movie Database (TMDB) (themoviedb.org)**

IV.2 Data Pre-processing

After the data is imported and visualized, we pre-process the data for training and testing purposes. Collected data must be labeled or tested with a feature extractor. Certain data features are selected in the Data Frame. Movie_metadata.csv contains information such as Movie title, director name, actor name, title, category, duration, rating, etc. on line 5043 and 28 columns. Credit.csv file downloaded from Movie Database, Kaggle contains 45476 rows and 8 columns.

IV.3 Web scraping from Wikipedia using Python

Web removal is a copying technique where specific data is gathered and copied from the internet, typically posted to a local website or spreadsheet, for subsequent retrieval or analysis. Downloading and uninstalling a web page's component is required to remove it. The lack of data on Kaggle between 2018 and 2022 provides a justification for removing the characteristics of films and the characters from those films from Wikipedia.

IV.4 Data Processing

After generating the datasets, we prepare the datasets using the "pandas" & "NumPy" library in python to create different data frames. The datasets will be divided into different groups and frames that is required for the next step, that is, transferring the data to the frontend using Flask framework (A python web framework). The dataset will be combined as a multiple list as a dictionary which can be passed to the html file so that it can be processed easily and the order of information will be preserved.
KEY:VALUE pair where KEY= [Movie, Cast, title, category, duration, rating, and so on]

IV.5 Sentiment Analysis (Training & Prediction)

In this approach, we want to use NLP (Natural Language Processing) for sentiment analysis. We read the feedback from the.txt file into an ipynb file. To obtain user reviews from the IMDB website, we use web scraping. The text is processed using NLTK (Natural Language Tool Kit) and the TFIDF (Term Frequency - Inverse Document Frequency) vectorizer. It aids in determining the significance of a word in a collection/corpus text, because this tells us which terms appear the most in the text [8].

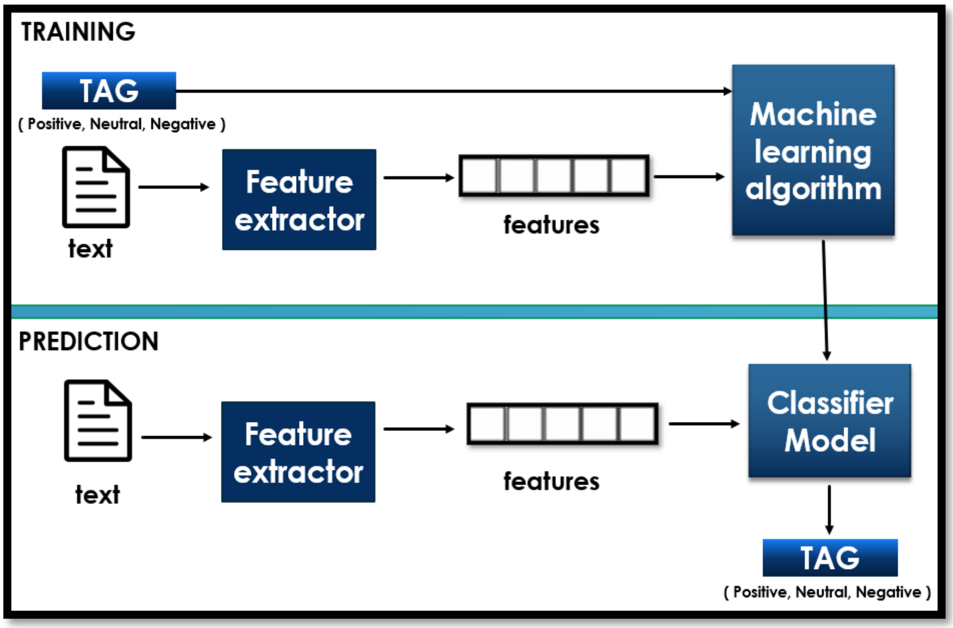

Fig 3. Workflow of Sentiment Analysis

IV.6 Naive Bayes

Naive Bayes is a popular algorithm for classifying text. Although it is simple, it often performs as well as much more complicated solutions. A naive Bayes' classifier works by figuring out the probability of different attributes of the data being associated with a certain class [4]. This is based on Bayes' theorem.

The Theorem is: -

$$P(A|B) = P(B|A), P(A)P(B)P(A|B) = P(B|A), P(A)P(B).$$

This states "the probability of A given that B is true equals the probability of B given that A is true times the probability of A being true, divided by the probability of B being true."

IV.7 Similarity Scores:

How does it determine which item is most comparable to the item liked (or selected in our case) by the user**?** Here comes the similarity scores. It is a numerical value ranges between zero to one which helps to determine how much two items are similar to each other on a scale of zero to one. This similarity score is obtained measuring the similarity between the text details of both of the items. So, similarity score is the measure of similarity between given text details of two items. This can be done by cosine-similarity.

IV.8 Cosine Similarity

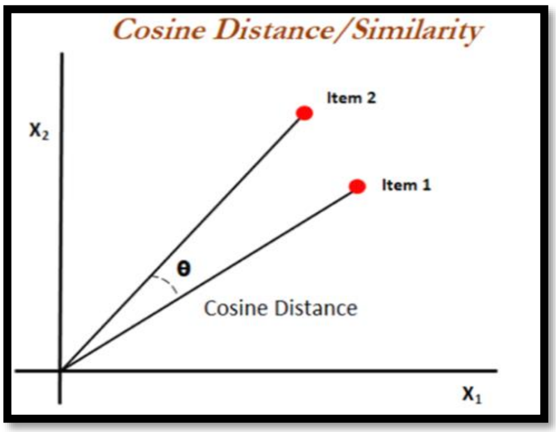

Fig 4. Cosine Distance/Similarity

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

IV.9 AJAX Request:

The request method will send the client a search in the form of a query.

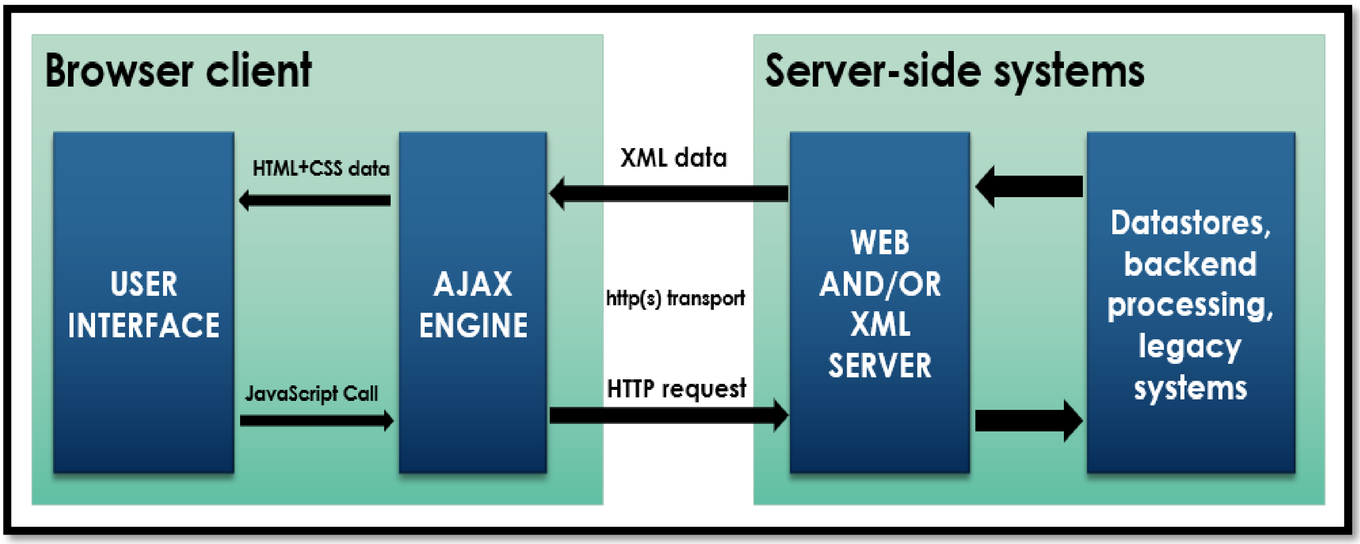

Fig 5. AJAX working

AJAX is stand for Asynchronous JavaScript and XML. AJAX requests are requests made by AJAX applications. Typically, it is an HTTP request done by (browser-resident) JavaScript that encodes the request and/or response data with XML. Ajax is a set of web development methodologies that use a variety of web technologies on the client side to generate smaller web applications. Web applications can submit data to a server and retrieve it in sync thanks to Ajax. Python Data can be displayed on the client side using Ajax in the form of Request and Feedback [9].

V. RESULTS

V.1 Landing Page:

This is the project's home page. The input field is a form that accepts a string of movie data and sends it to the back-end for pre-processing from the database. AJAX is then used to return the results to the client side.

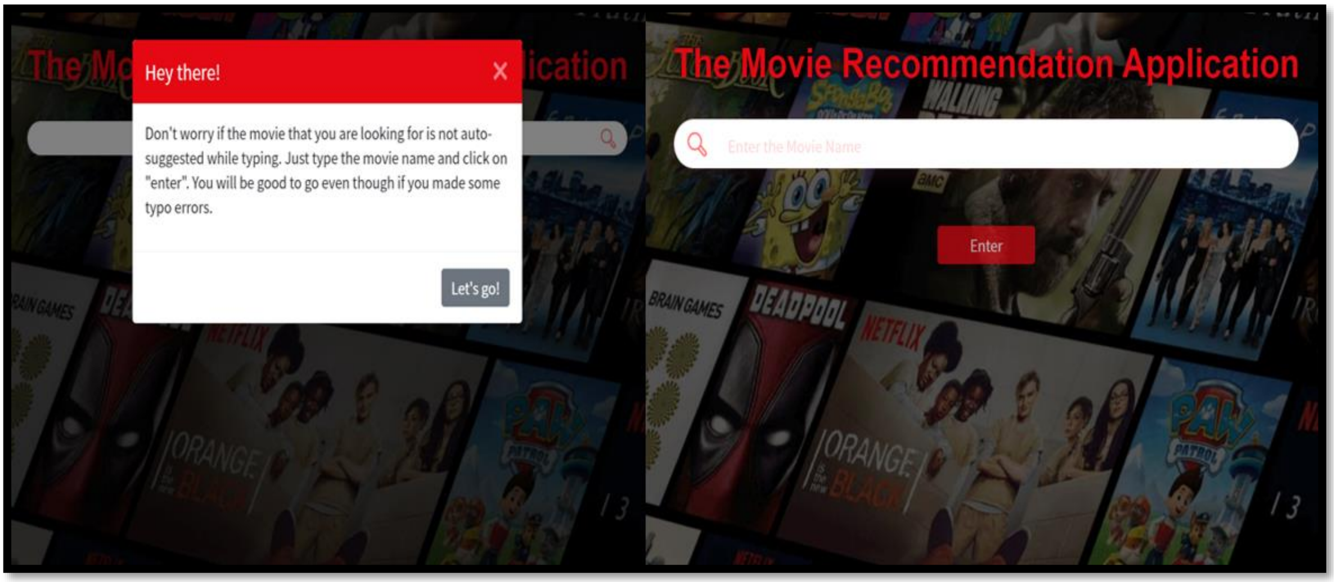

Fig 6. Landing Page.

V.2 Search result:

The output generated by the Search engine will be the movie name entered by the user. This is the result of our project, where in the string gets processed at the back-end and the details of movie along with genre, overview, runtime, rating, status, top cast, name, year of realized etc. are displayed at the client side.

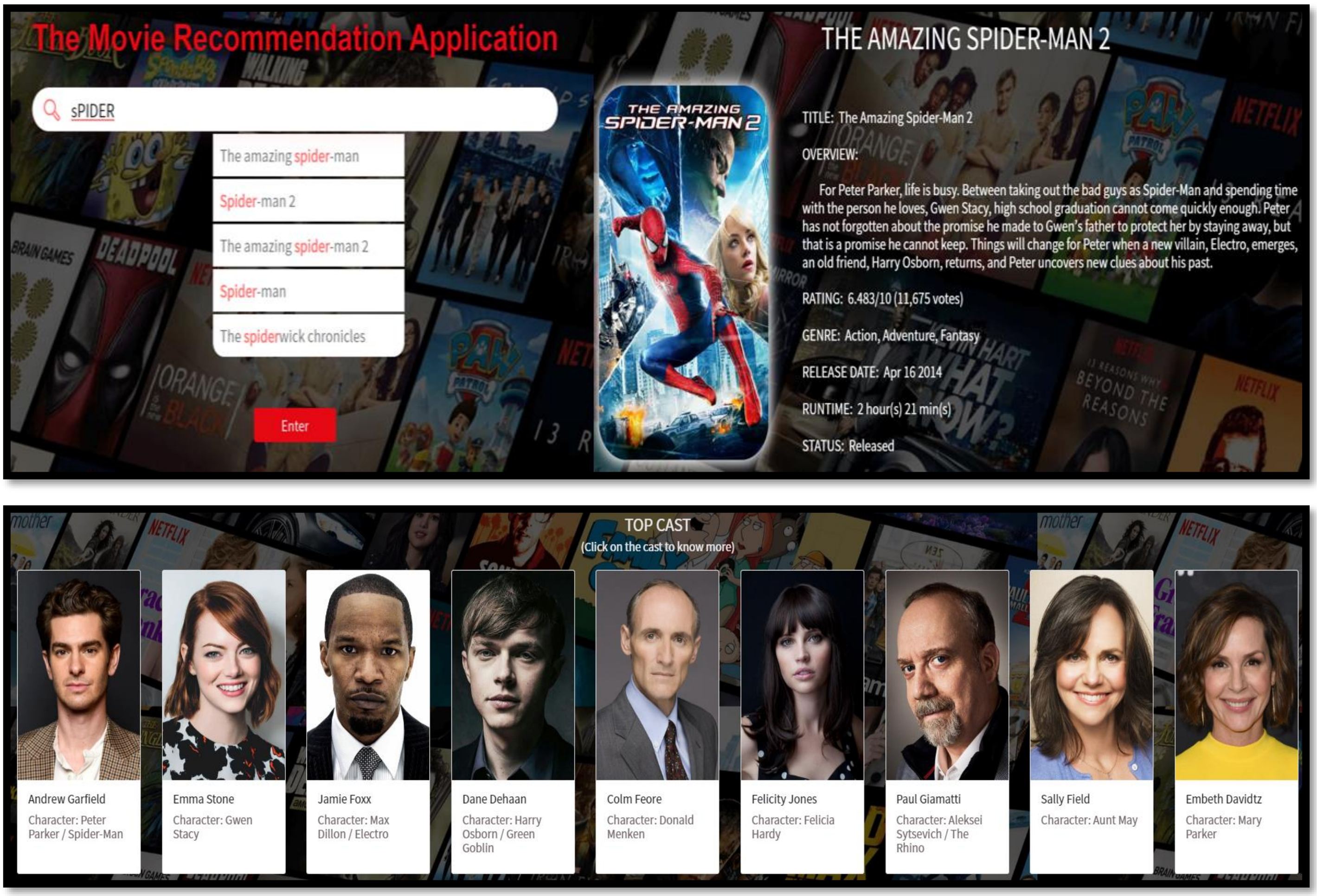

Fig 7. Search Result.

V.3 Web Scraping:

The cast portion of a movie uses Web scraping to display the cast's information. This is the outcome of the web scraping phase, in which the information about each actor who appeared in the film was obtained from Wikipedia.

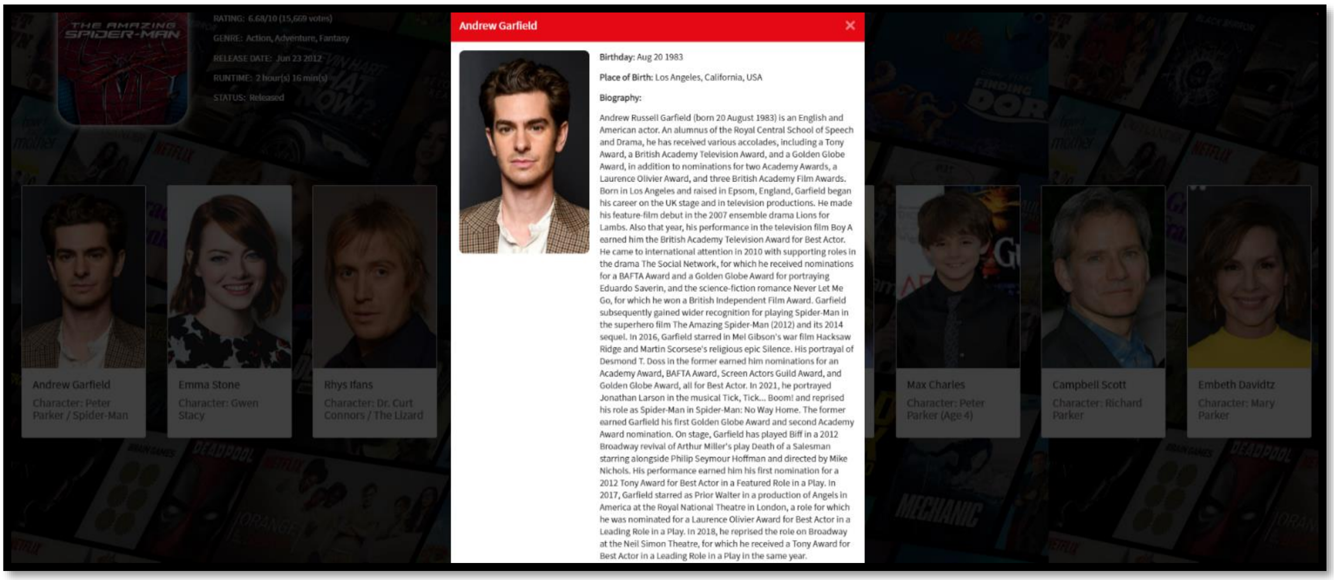

Fig 8. Web Scraping.

V.4 Sentiment analysis:

This is the project's sentiment analysis section, where user reviews are displayed and classified as positive, negative, or neutral using sentiment analysis.

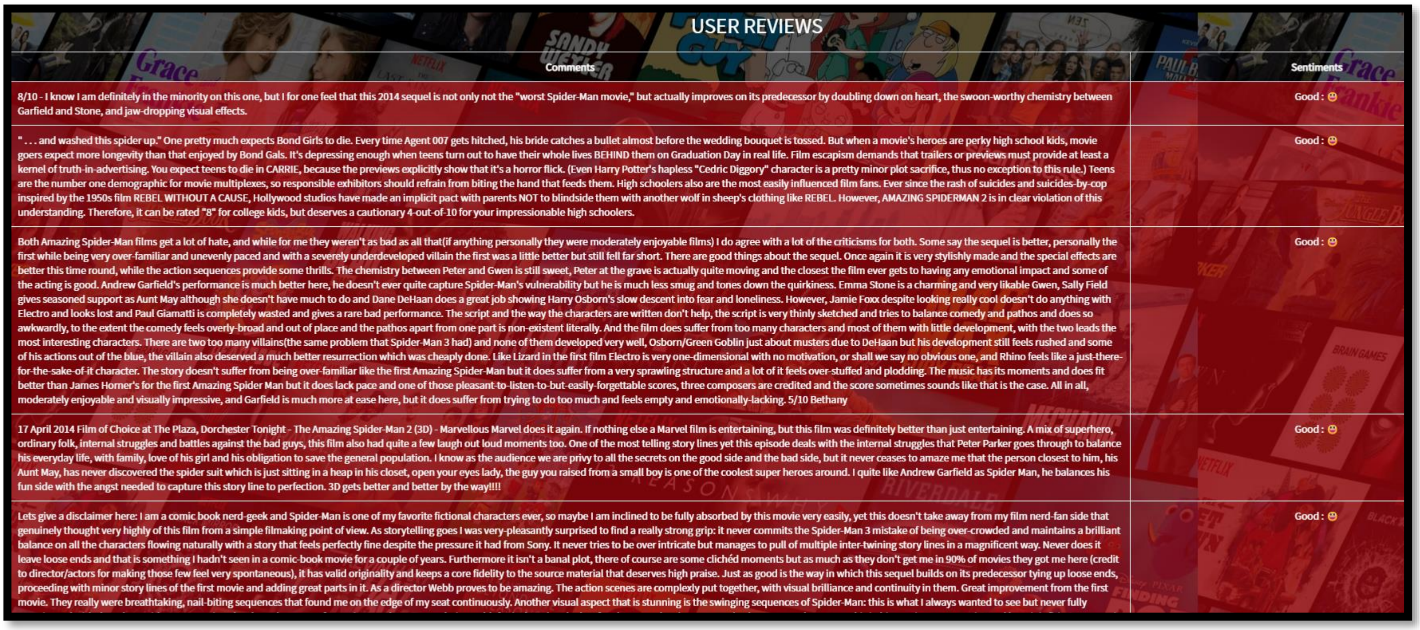

Fig 9. Sentiment Analysis.

V.5 Movie recommendation:

Sentiment analysis is used to recommend movies based on the genre, the same actors, the directors, etc. The movies that received the highest rating through content-based filtering are shown. This is what content-based filtering produced.

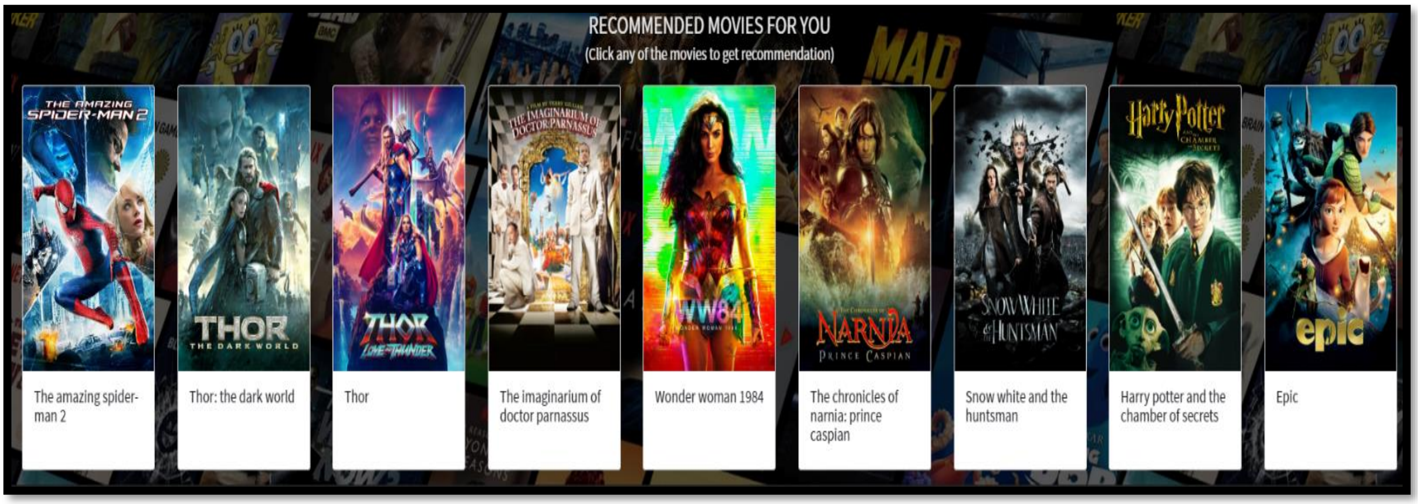

Fig 10. Movie Recommendation.

VII. Conclusion

The usage of recommendation systems can be a fantastic way to filter information and give users only the pertinent information. In this research, we present a recommender system that makes use of the TMDB data set and sentiment analysis. Metadata and a social network are essential factors utilized to recommend movies. The main use of Sentiment Analysis in the proposed model was to observe reviews of users for a particular movie. Sentiment analysis is helpful in gathering data on audience responses to a particular movie, and these responses are helpful in making more recommendations. we improve the suggestion process by gathering more pertinent data regarding user opinions, preferences, etc. It is possible to accomplish this and use all the facts to create a more potent recommendation system with the help of emerging technology. For our tryouts, we solely took into account films released up until 2022 using a static database. This structure can be studied in a fluid setting that frequently includes fresh films.
The project's accuracy is approximately 98%. Since the project depends on the updated database, it is challenging to obtain all the information for all updated movies from the database, which results in a 2% accuracy reduction.

VIII. ACKNOWLEDGMENT

We would like to thank our Project Guide Prof. Satish Chadokar for providing useful information to help us with our survey and many other useful discussions. We thank our HOD Prof. Pravin Malviya and Project Coordinator Prof. Vinay Sahu for their valuable advice during the course of this work. Lastly, we express our deepest gratitude to our Principal Dr. Paresh J. Shah for providing us with an environment to complete our survey successfully.

REFERENCES

[1] Raghav Mehta, Shikha Gupta, "Movie Recommendation Systems using Sentiment Analysis and Cosine Similarity," vol. 7(01), pp. 16-22, 2021.

[2] Aarushi Sharma, Kandyala Lekhana, Keerthana M, Keshav Anil Rathi, Dr. Sumithra Devi K A, "Movie recommendation system with sentiment analysis using AJAX," Vol. 03, Issue. 04, 2021.

[3] Sharma, Aarushi, Kandyala Lekhana, M. Keerthana, Keshav Anil Rathi, and Sumithra Devi KA. "MOVIE RECOMMENDATION USING SENTIMENT ANALYSIS AND AJAX."

[4] Dubey, A., and R. Ranjan. "Movie Recommendation System Using Naive Bayes Algorithm with Collaborative Filtering." International Journal of Science and Research 9, no. 7 (2018): 2018-2020.

[5] Le, J. "The 4 recommendation engines that can predict your movie tastes." (2018).

[6] Li, Hui, Jiangtao Cui, Bingqing Shen, and Jianfeng Ma. "An intelligent movie recommendation system through group-level sentiment analysis in microblogs." Neurocomputing 210 (2016): 164-173.

[7] B. Shen, and J. Ma, "An intelligent movie recommendation system through group-level sentiment analysis in microblogs," Neurocomputing, vol. 210, pp. 164–173, 2016.

[8] Ravi, Kumar, and Vadlamani Ravi. "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-based systems 89 (2015)

[9] Clayton J Hutto and Eric Gilbert. Vader: A simple rule-based model for analyzing sentiment in social media text. The Eighth International Conference on Weblogs and social media, held in 2014, is a collection of papers presented at the conference.

[10] P. Zigoris and Y. Zhang are the authors of this paper. User profiling using Bayesian adaptive methods with explicit and implicit input. Presented at the 2006 Conference on Information and Knowledge Management (27)

[11] Yu, K., Tresp, V., and Yu, S. Information filtering using a nonparametric hierarchical Bayesian framework. Pages 353–360 in SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and Development in Information Retrieval. The ACM Press published this book in 2004. (25)

[12] **Kaggle: Your Machine Learning and Data Science Community**

[13] **The Movie Database (TMDB) (themoviedb.org)**

[14] **https://datasets.imdbws.com/**