Gurukul International Multidisciplinary Research Journal (GIMRJ)*with* International Impact Factor 8.249
Peer Reviewed Journal
https://doi.org/10.69758/RJNJ1514

e-ISSN No. 2394-8426
Special Issue On Advanced Computational Techniques: Emerging Trends from Postgraduate Studies
Issue–I(VI), Volume–XII

Movie Recommendation System With Sentiment Analysis Rating

[1]Bhagyashri K.Baghele,[2]Prof. Prerna Dangra
[1]PG Student, [2]Assistant Professor
Department of Computer science,
G H Raisoni Amravati University Nagpur, India

Received on: *11 April ,2024* ***Revised on:*** *26 May ,2024,* ***Published on:*** *01 June ,2024*

Abstract: People's desires, trends, and interests change in tandem with how the world is changing. People like to see movies based on their interests, and this is also true in the movie industry. There are a lot of web-based movie service providers now, and they aim to keep their members entertained in order to grow their clientele and notoriety. The service provider could suggest movies that customers would enjoy in order to increase business by giving them an excuse to view more entertaining movies. Customers are likely to renew the web-based movie service provider application on a regular basis if this is done. The aim of this project is to develop a machine learning-driven movie recommendation system that can suggest films to users based on their ratings and areas of interest. In order to do this, collaborative filtering is utilized to calculate features based on user and movie information, and content-based filtering is used to recommend movies based on movie-movie similarity. To increase performance, the suggested system makes advantage of the novel ensemble learning algorithm, XGBoost.

Keywords- **Natural Language Processing (NLP), Machine Learning (ML), Machine Learning, Recommendation System, Content based filtering, Collaborative filtering, sentiment analysis.**

1. INTRODUCTION :

A recommendation system is a type of information filtering system that makes suggestions based on user preferences and makes an effort to predict the interests of the user . Recommendation systems have a wide range of applications. These have grown in popularity, and the majority of the websites we use today employ them. These systems frequently have the ability to gather data regarding user preferences, which they can then utilize to enhance their recommendations going forward. Generally speaking, recommendation systems are made to assist consumers in locating and choosing products, which could be movies, books, or dining establishments that are accessible online or through other digital information sources like Netflix, Hotstar, Amazon, etc. A movie recommendation system suggests movies based on the user's wants and preferences, presenting a limited selection of movies that best suit the user's needs based on the user's information and item specifics. There are essentially two methods used by recommendation systems . While the second one is based on collaborative filtering, the first one is based on content-based filtering. Content-based filtering systems often examine the information's content and identify patterns in it. When a user uses a content-based filtering movie recommendation system, the system will first examine all of the movies the user has viewed in the past and evaluate their content before suggesting a new film to them. After that, it suggests movies with material that is comparable to the user's. Conversely, the collaborative filtering is determined by the ratings provided by various users. It operates on the premise that two users with similar tastes are those who provide the same rating to certain films. Thus, it suggests movies to users depending on how well-liked they are by other users who are similar to them. The primary goal of the suggested system is to give users with movie recommendations based on both their personal tastes and other users' ratings left after viewing a certain film. This is accomplished through the use of collaborative filtering, which generates different features using user and movie ratings, and content-based filtering, which predicts a list of comparable movies. The system's accuracy is increased with the usage of the XGBoost algorithm.

2. RELATED WORK:

Recommender systems and information retrieval have conducted a great deal of research on movie recommendation systems. Sentiment analysis approaches have gained popularity in recent years as a way to improve these systems and capture the emotional side of user preferences. This section examines the body of research and literature in the fields of sentiment analysis and movie recommendation systems.

Conventional movie recommendation systems have mostly used hybrid techniques, content-based filtering, or collaborative filtering. In order to provide recommendations based on similarities between users or objects, collaborative filtering techniques examine user-item interactions. Conversely, content-based filtering techniques match things to users' tastes by suggesting them based on their characteristics and attributes. Hybrid strategies combine content-based and collaborative methods to best utilize each one's advantages.

A wide range of topics related to movie recommendation systems have been examined in numerous research works, such as algorithmic methods, assessment criteria, and user modeling approaches. While some research has addressed scalability and efficiency difficulties in large-scale systems, others have concentrated on increasing recommendation diversity and accuracy. Furthermore, studies have looked into how user demographics, temporal dynamics, and social influence affect recommendation performance in context.

Though traditional recommendation algorithms have advanced, people' emotional reactions to movies are still frequently ignored by these systems. Although users' genre preferences and movie ratings may be reliably predicted by them, their ability to capture the subtle emotional reactions that impact users' overall happiness and enjoyment may be compromised.

Sentiment identification and extraction from textual data is the primary goal of sentiment analysis, sometimes referred to as opinion mining, which is a branch of natural language processing (NLP). Determining the polarity (positive, negative, or neutral) of opinions expressed in text—such as evaluations of products, posts on social media, or comments on movies—is the aim of sentiment analysis.

Many sentiment analysis techniques have been created by researchers; these include lexicon-based approaches, machine learning techniques, and deep learning techniques. Lexicon-based techniques use dictionaries or prepared sentiment lexicons to give words and sentences a sentiment score. Based on labeled training data, machine learning models, such recurrent neural networks (RNNs) and support vector machines (SVMs), learn to categorize text. By identifying intricate patterns in textual input, deep learning architectures like recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have demonstrated encouraging results in sentiment analysis tasks.

Sentiment analysis has been used in many different fields, such as customer feedback analysis, social media analysis, and product recommendation. Sentiment analysis can offer insightful information about viewers' emotional reactions to movies in the context of movie recommendation systems, allowing for more engaging and personalized suggestions. Although sentiment analysis and movie recommendation systems have benefited much from previous research, there is still a need to integrate these two fields to improve user happiness and suggestion accuracy. The next sections of this study report on experiments conducted to evaluate the effectiveness of a novel framework for incorporating sentiment analysis ratings into movie recommendation systems.

3. METHODOLOGY

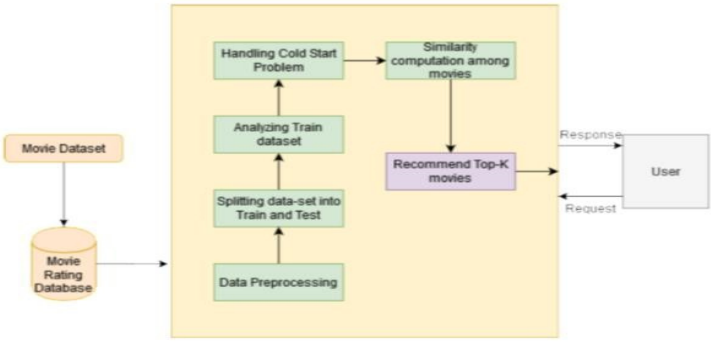

Figure1. Flow Diagram

The flow-diagram of our suggested strategy for movie recommendations is displayed in Fig. 1. We must first obtain the movie dataset. There are five files in our dataset: four of them include customer rating data, and the fifth file contains the title and ID of the movie. After that, we cleaned and preprocessed the data in accordance with the analysis's findings. We discovered throughout analysis that our dataset had a Cold Start issue, which is addressed appropriately. Based on current user ratings and movie ratings, a list of new features is derived to determine how similar the movies are to each other. The technology suggests movies to users based on their previous viewing choices whenever they watch a new film. The several processes in the suggested Movie Recommendation method are covered in this section.

A. Data Gathering

The Netflix Movie Recommendation competition dataset was obtained from Kaggle.com and utilized in the suggested work . The dataset consists of five files in total. Of which four files provide the CustomerID, Rating, Date, and MovieIDs of every film the customers have viewed. The movie ID, title, and year are contained in the fifth file.

The MovieIDs in the dataset range sequentially from 1 to 17770, and the CustomerIDs, with gaps, range from 1 to 2649429. There are 480189 distinct clients. There are 100480507 consumer movie ratings in the dataset. Movies are rated on a scale from 1 to 5. The five files are combined into a single CSV file, which is then kept in the database. Once we had all the information, we began to analyze the dataset and looked at the distribution of scores.

B. Data Preprocessing

Null or empty data entries in CustomerID, Rating, Date, and MovieIDs are found in the data preprocessing step. There are no empty data values in the dataset that is provided. If there are duplicate values, they are also eliminated from the dataset. We have not found any duplicate records in our dataset.

C. Splitting data into Train and Test sets

The dataset should initially be split into training and test datasets in order to construct the machine learning model. Using the training data set, the machine learning model discovers patterns and makes predictions about future responses. The algorithm's efficiency is then calculated based on the anticipated outcome. The test dataset is used to verify the algorithm's ability to predict new responses more accurately based on the learning it completed during the training phase. The training and testing portions of the dataset are split 80:20.

Total Customers	Total Movies	Date range	Ratings Range

380889	15670	2000-2005	1-5

TABLE 1. DESCRIPTION OF DATASET

After splitting dataset into 80% for training and 20% for testing, the description of training dataset and testing dataset are shown in tables 2 & 3.

Total Customers	Total Movies	Total No. of Ratings
405341	32425	9884405

TABLE 2. DESCRIPTION OF TRAIN DATASET

Total Customers	Total Movies	Total No. of Ratings
348712	24757	19896102

TABLE 3. DESCRIPTION OF TEST DATASET

D. Analyzing the training dataset

Figure 2 displays the distribution of movie ratings in the training dataset. The chart shows that the majority of people who rated movies gave ratings of 4, 3, and 5, while a small percentage of users offered ratings of 1 and 2.

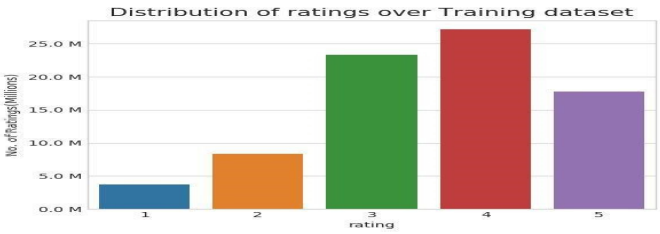

Figure 2. Distribution of rating over Training dataset

The training dataset's Cumulative Distribution Function (CDF) and Probability Density Function (PDF) were also shown. Figure 3 reveals that the majority of users who watched movies only rated a few of them, with the peak sign indicating that PDF, or the number of ratings per user, is present. Ninety percent of consumers only provided a few ratings, as shown in the CDF plot.

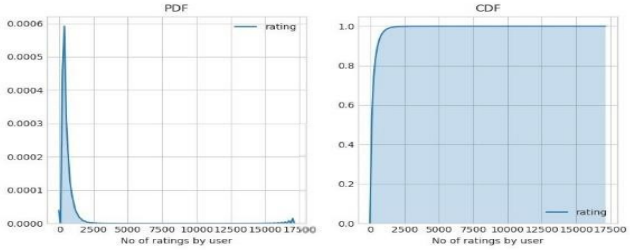

Figure 3. PDF & CDF of rating in Train dataset

E. Cold start problem with Users and Movies

Every time a new user registers or a new video is uploaded to the dataset, the cold start issue arises. For the simple reason that neither the new user nor any other user would have rated any movies. In order to solve the cold start issue, the rating for each new user and movie in the dataset is changed to 0.

F. Finding similar movies

The Cosine Similarity is used to determine how similar two movies are to each other. The dot product of one movie vector and another movie vector is calculated using cosine similarity. The outcome shows how similar these films are to one another. A pair of vectors A and B are considered comparable if their dot product and magnitude product ratios are equal. Its mathematical definition is provided by equation .

$\text{Cos}(\theta) = \|AA.B\|\|B\| = \sum n\ similay\ AiBni\ Bi2$ (1)

If two movies' vectors are identical, their similarity will be 1, and if they are orthogonal, it will be 0. Stated otherwise, the similarity is a limited integer between 0 and 1 that indicates the degree of resemblance between the two vectors. We calculated the movie-movie similarity matrix. Figure 4 is how the similarity matrix appears. Because every movie is a mirror image of itself, all the numbers on the diagonal are 1. Because the similarity between A and B is equal to that between B and A, the matrix is also symmetrical.

□ 1 0.22 1 0.15 □ 0.07□ □
□ 0.34 0.34 □ 0.18□ □
□0.22 □ 1 □ 0.26
□0.15 0.18 □ □□ □
□ 0.26 1 □
□ □0.07

Figure 4: Movie-movie similarity matrix

G. Root Mean Square Error Performance Metric

A common statistic to gauge the model's error rate is root mean square error (RMSE). It is predicated on the discrepancy between actual and model-predicted values. To calculate RMSE, use equation (2).

H. Generating new features using ratings

H. Generating new features using ratings

In order to enable the XGBoost algorithm to predict user ratings in the test data, five additional features were generated from the training dataset using user and movie rating data.

•Here is a list of the five new features.

• Gavg: The overall global average rating
• The top five similar users who rated the same movie are sur1, sur2, sur3, sur4, and sur5.
• The top 5 similar movies as assessed by a user are smr1, smr2, smr3, smr4, and smr5.
• UAvg: The average user rating
• MAvg: The mean score assigned to a film

I. Empirical filtering with the XGBoost algorithm

A content-based recommendation system can only make recommendations for films that are similar to a specific film. It is unable to identify preferences or make suggestions. A collaborative filtering system makes recommendations for products based on user preferences. Here, the system will calculate the most comparable users for the movie using cosine similarity. For instance, the algorithm will suggest user1 with

movie2 if user1 has rated movie1 and user2 has rated both movies. It works similarly to suggesting movies to consumers based on comparable users who have already seen the film.

J. XGBoost

It is commonly known that XGBoost (Extreme Gradient Boosting) outperforms other machine learning algorithms in terms of results The gradient boosting framework is the foundation of XGBoost, a class of boosting algorithms.

The ensemble principle is the foundation of the boosting approach. It creates a single predictive model by integrating many machine learning approaches. Generally speaking, the algorithm is employed to increase accuracy. Because of these benefits, the XGBoost algorithm is presently used in a lot of machine learning projects. In general, it outperforms other machine learning methods such as the KNN algorithm and linear regression. It can take use of parallelism since it can run on multi-core systems.

4. RESULTS AND DISCUSSIONS

We used Python 3 Jupyter Notebook to conduct experiments on movie datasets in order to assess the performance of the suggested solution. The suggested system has a 64-bit version of Windows 10 operating system and an Intel Core i7-5600U @ 2.60GHz CPU with 16.00 GB of RAM. Based on the movies a user has watched, the suggested system will suggest the Top-10 related movies for that user. The Root Mean Square Error (RMSE) is also reduced by using the 13 characteristics of the XGBoost algorithm.

For instance, when a user enters the movie ID 1 for the film "Dinosaur Planet," the suggested recommender system displays a list of the top 10 comparable films, as illustrated in Figure 5. Based on user preferences, we can see from the figure how similar the movies are.

movie_id	year_of_release	title
694	2000.0	When Dinosaurs Roamed America
5302	2003.0	Chased by Dinosaurs: Three Walking with Dinosa...
1084	2001.0	Walking with Prehistoric Beasts
13586	2001.0	Allosaurus: A Walking with Dinosaurs Special
1173	1999.0	Walking with Dinosaurs
4181	2003.0	Walking with Cavemen
8800	2003.0	Prehistoric America: A Journey Through the Ice...
10656	2003.0	Before We Ruled the Earth: Mastering the Beasts
15648	2002.0	National Geographic: Dinosaur Hunters: Secrets...
10257	2002.0	Prehistoric Planet: The Complete Dino Dynasty_

Figure 4. Output of Recommendation system

Movie ID	Movie Title	Ratings from Customer	Total Similar Movies
59	The Rise and Fall of ECW	752	19323
555	Sherlock Holmes and the Spider Woman	877	13294
2043	The Winds of War	1067	17320

TABLE 1. SAMPLE OUTPUTS

V. CONCLUSION

We have put in place a collaborative filtering and content-based filtering suggestion system. By adding 0 ratings, the dataset's cold start issue is resolved. The suggested system made advantage of thirteen features, including user and movie data as well as content-based and collaborative filtering to forecast the top ten movies based on user interests. The XGBoost algorithm is used to increase the system's performance. The RMSE value that we determined was 1.076. Our next tasks include putting the deep learning algorithm-based recommendation system into practice and assessing how the accuracy of the system has improved.

VI. REFERENCES

1) Clustering algorithms in hybrid recommender system using movielens data: Kuzelewska, U. (2014) Studies in logic, language and rhetoric, vol. 37, no. 1, pp. 125-139.
2) Geetha, G., Safa, M., Fancy, C., and Saranya, D. [2]. "A hybrid approach for recommender system using content-based filtering and collaborative filtering" was published in Journal of Physics: Conference Series, volume 1000, issue 1, page 012101, 2018.
3) De Campos, L.M., Rueda-Morales, M.A., Huete, J.F., and Fernández-Luna, J.M. (2010). The International Journal of Approximate Reasoning, vol. 51, no. 7, pp. 785799, presents a hybrid technique based on Bayesian networks for combining content-based and collaborative recommendations.
4) "A Movie Recommender System: MOVREC," Manoj Kumar, D.K. Yadav, Ankur Singh, and Vijay Kr. Gupta; International Journal of Computer Applications, vol. 124, no. 3, pp. 7-11, 2015.
5) The article "Movie Recommender System using Collaborative Filtering" was published in the International Journal on Future Revolution in Computer Science & Communication Engineering in 2018. It was authored by Nupur Kalra, Deepak Yadav, and Gourav Bathla.
6) Jaydeep Gheewala and Sonali R. Gandhi, "A survey on recommendation system with collaborative filtering using big data," in Proceedings of the 2017 IEEE International Conference on Innovative Mechanisms for Industry Applications, pp. 457–460.
7) "MOVIEMENDER A Movie Recommender System," by Rupali Hande, Ajinkya Gutti, Kevin Shah, Jeet Gandhi, and Vrushal Kamtikar, International Journal of Engineering Sciences & Research Technology, pp. 469–473.
8) Carlos Guestrin, Tianqi Chen, and others [9]. Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016. "Xgboost: A Scalable Tree Boosting System."