WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Utilizing Machine Learning to Deploy a Recommendation System with Content-Based Recommenders

Syed Kaif Ulla[1], Mohammed Fakrudin Shahid[2], Ms. Jeshma Nishita Dsouza[3]

[1,2] UG Student, [3]Guidance
St. Joseph's University, Bangalore University

ABSTRACT

In today's digital world, there is a lot to discover and look at on the Internet, but not everything is to our liking. We may receive feeds of films, news, clothes, and other content that we do not like or are interested in. This leaves the customer disinterested in the application and unwilling to use it again. In this session, you'll need to write code that, at a beginner's level, can recognize good patterns in your customer's behavior and recommend articles that are most relevant to your customer's interests. This allows the consumer experience to be satisfying, resulting in positive evaluations and popularity.

YouTube is home to many of the recommendation systems we encounter in our daily lives. When we notice a lot of news regarding GK and current issues, they serve matching videos to various subscribers. can help you gain popularity through application ratings while boosting user experience. This recommendation system guideline can help you achieve the best results for your application's profitability while also bringing your team closer together.

Nowadays, people watch movies to unwind from their hectic life. However, watching movies is difficult since it takes a long time to select movies from the vast database of movies available in the globe, and it is a process. The procedure was tough and time-consuming for movie pickers due to the big data collection, but we were able to accomplish it with the help of a movie recommendation system that uses content-based filtering, in which this recommendation system suggests movies to viewers based on what they watch. In this article, I demonstrate how to create a recommendation system utilizing content-based filtering and a Heroku implementation. Typically, recommender systems are created utilizing either content-based or collaborative filtering. Heroku enables easy application deployment

I. INTRODUCTION

Today, recommendation systems are critical in places such as YouTube, Netflix, Amazon Prime, Instagram, and Facebook, where the system suggests an item to the user based on the ratings or preferences assigned to the item. The technology even predicts the ratings or preferences assigned to an item. When you search for something on the website, this recommendation system assists you by suggesting goods you might appreciate. Assume you looked for a cell phone today on Amazon or Flipkart. When you open the Amazon or Flipkart app again, the system will show you some cell phones connected to your previous search. This is similar to how Amazon Prime and Netflix propose movies based on their viewing history and ratings.Every action you take on these websites is monitored by a system that subsequently recommends movies to you depending on your tastes. As far as we know, there are three basic types of filtering techniques: content-based filtering, collaborative filtering, and hybrid filters. For this project, I used the content-based filtering technique and the popular TMDB movie dataset. To develop the website, I used common Python libraries such as pickle, sklearn, NumPy, pandas, requests, and AST, as well as Streamlit. The primary purpose of this project is to develop a model that allows a system to easily recommend movies to a user based on their prior behavior.

Today, recommendation algorithms are widely utilized and have found applications in a range of industries, including e-commerce, retail, finance, and entertainment. These systems collect and automatically analyse user data in order to generate tailored suggestions for each user. Content-based filtering (CBF), collaborative filtering (CF), and hybrid filtering are the three most used strategies for developing recommendation systems. CBF is a way for evaluating each item's content and recommending further things with similar traits. By comparing the likenesses of the people and the items, CF addresses some of CBF's flaws and provides recommendations. It generates a recommendation based on the user's previous preferences and the preferences of other users who share those preferences

Natural language processing (NLP) is a technology that allows you to extract information from text and classify statements, words, and documents as positive or negative. Knowing the author's perspective and indicating the user experience are both quite beneficial. Opinion mining uses data mining concepts to extract and categorise thoughts published in various online forums or platforms. This allows us to better understand how the user feels about a specific topic.

The study's objectives are to deal with huge amounts of data and filter important information, to recommend related films depending on user preferences, and to conduct sentiment analysis on reviews of the chosen movie.

II. LITERATURE REVIEW

Sang-Min Choi et al. investigated the sparsity problem, the cold-start problem, and other limitations of the collaborative filtering approach. To address this issue, the authors offered a patch based on category data. The authors proposed a genre-based movie recommendation system. The writers say that the freshly created content contains category information. New content may still be recommended based on category or genre information, even if it lacks sufficient ratings or views. The proposed solution is unbiased in terms of both recently published, rarely watched, and highly rated content. As a result, a hybrid technique for movie selection was proposed, allowing for the recommendation of even new films. Both content-based and collaborative filtering have limitations, although they can be useful in some situations, according to the authors. As a result, the authors developed a hybrid technique that incorporates both contextual and content-based features. In the film "More," filtering and collaborative filtering are used as solutions. System of proposals Pure collaborative filtering does not use the Pearson correlation coefficient. Used. Instead, a new recipe was used. This formula, however, includes a "division by zero" error. This mistake occurs.

Because of this, the authors ignored users who gave the films the same rating. In the case of p, we created a "Weighted KM-Slope-VU" collaborative filtering technique for movie selection. The authors used K-means clustering to divide users into groups that were comparable.users discussed three sorts of recommender systems: simple, content-based, and customized. Finally, a collaborative filtering-based hybrid recommender system was proposed as a possible solution. The K-means clustering technique was proposed as a solution by System of Recommendations. To discriminate between similar users, the authors used clustering. The authors later created a neural network to provide recommendations for each cluster.

The suggested system consists of several phases, including data preparation, principal component analysis, clustering, and classification. Data Preprocessing for Building and Applying Neural Networks When determining the consumption ratio, we took into account users' opinions, preferences, and feedback.

Following the clustering stage, the authors utilized a neural network to replicate user ratings for unwatched movies and make predictions.Finally, recommendations are made based on anticipated high ratings. Gaurav Arora and colleagues developed a movie recommendation algorithm based on user similarity. The study paper's broad reach is based on a hybrid methodology that employs collaborative filtering and context-based filtering, but neither the parameters used nor the specifics of how it was implemented internally have been made public. Subramanya Swamy et al. proposed a tailored solution for movie suggestion.

The Euclidean distance measure was used as a collaborative filtering strategy to find which users were most similar. It is determined which user has the smallest Euclidean distance value. Lastly, movie suggestions are based on The highest rating given by a user.

Furthermore, the authors suggest that the guidance evolves with time. that the system improves performance as the user's preferences change over time. Harper et al. thoroughly discussed The Movie Lens. Recommendation systems can help reduce information overload.

The authors discussed a number of themes, including scalability, cold start concerns, and data scarcity. The authors analyzed over 15 research publications on movie recommendation systems. After reviewing all of these publications, the authors discovered that many writers used collaborative filtering instead of content-based filtering. In addition, they discovered that many authors used a hybrid method.

Even though recommendation systems have been extensively studied, there is always room for improvement. The disadvantages that currently exist. Niharika Immanent and colleagues developed a hybrid recommendation technique that uses both content-based and collaborative filtering approaches in a hierarchical manner to provide consumers with personalized movie recommendations. The most unique aspect of this research work is how the authors recommended movies using an appropriate order of photographs that described the movie's story structure. This leads to improved graphics.

The author also explored content-based techniques, hybrid recommender systems, collaborative filtering systems, gene-correlated recommender systems, and other methodologies. The suggested strategy consists of four major parts. Social networking sites such as Facebook are first used to determine a user's interests.

The movie reviews must then be analyzed, and recommendations must be made.

III. METHODOLOGY USED

This section discusses the approaches used to conduct the study and implement the algorithms. The study employed a dataset to train the cosine similarity model, which is used to recommend movies. Then, a new dataset was used to train Nave Bayes and the Support Vector Machine Classifier for Sentiment Analysis. The movie recommendation model is now used to predict related movies using a movie's name as an input. The movie's reviews are scraped from the IMDB website and fed into the Sentiment Analysis model, which classifies them as either favorable or negative.

Following preprocessing of the dataset, the required features must be combined into a single feature. Later, we'll need to convert that feature's text into vectors..

We must then determine how comparable the vectors are. Finally, seek guidance based on the system architecture shown below. I began by gathering data from the TMDB movie dataset, from which I collected two datasets: the movies and credits databases, and then imported the dataset onto the terminal I was working on to display the data that was accessible. Following data preparation to isolate critical information from irrelevant data, I exhibited the dataset to see which attributes were useful for making suggestions. Throughout the procedure, I experienced challenges with missing data values and data imbalances. After that I converted the movie data to vectors using Scikit-Learn's count vectorizer tool.This tool converts the given text into vectors based on the count of each word in the whole text, excluding stop words. The model is built with a count vectorizer and cosine-similarity to discover the angle differences between two vectors, create a similarity matrix, and locate similarities between the videos. A proposal technique is then developed to recommend a movie to the user in this manner.

i. Dataset

The study used three datasets. The first is for sentiment analysis, and the other two are for movie recommendations. "tmdb 5000 movies.csv," "tbmd 5000 credits.csv," and "reviews.txt" files are used for recommendation and sentiment analysis, respectively. The two datasets used for movie suggestion are then joined to form a single data set, with the columns "movie id," "title," and "tags" retained.

The reviews data collection includes only two columns: "reviews" and "comments." Positive comments are numbered one, while critical remarks are numbered zero. 2975 negative comments outnumber 3943 positive comments.

ii. Data Pre-Processing

After combining the two datasets, just the relevant columns, such as "movie id" and "title," remained. "Overview," "Genres," "Keywords," "cast," and "crew" are kept, but the rest of the dataset is removed. The columns for "genres," "keywords," "cast," and "crew" were then refined with Abstract Syntax trees. These columns have also been grouped under the title "tags." The column "tags" is then tokenized with the count vectorizer. Tokenizing means dividing up sentences into individual words. This concludes the pre-processing for the movie recommendation. Natural Language Tool Kit (NLTK) is required for pre-processing sentiment analysis data. Python programs can be constructed utilizing the NLTK, a leading platform for working with human language data..

This is a standard Python module for computational linguistics and natural language processing. This library is used to download stop words. The most common words in any language are stop words. These terms are: "a," "an," "the," "if," and "or." Because they contain little significant information, they are used in text mining and natural language processing (NLP) to remove such words. The Comments column is then tokenized using the Tfid-fVectoriser.

The datasets' data pre-processing is now completed. The dataset for movie recommendations, which contains the "movie id," "title," and "tags," is displayed in Figure 8. The dataset for sentiment analysis is displayed in Figure 9. It has two columns, one for "comments" and another for "reviews."

The Cosine Similarity algorithm is implemented using Python's sklearn module. After asking the user to select one, the algorithm suggests five additional movies that are similar to the one they entered.

Vectors are utilized as data objects in cosine similarity, and the similarity is calculated using a product space. The resemblance increases as the distance decreases and decreases as it grows. Regardless of size, the cosine similarity measure can be used to calculate how similar two data objects are.

iii. Sentiment analysis with machine learning

SVC and NB, two algorithms that have been used, were previously discussed. The data is separated into two sets: testing and training, with sizes of 0.20 and 0.80, respectively. Next, the two models are fitted. To increase accuracy, both models have their hyperparameters adjusted.

SVC is a supervised technique in machine learning that can be used for classification as well as regression. The data points are classified by identifying a hyperplane in an N-dimensional space. If there are just two input features, the hyperplane is a line; if there are three input characteristics, the hyperplane is a two-dimensional surface. Radial Basis Function. The kernel was used in the model, which is a type of Non-linear SVM. It is referred to as the RBF kernel. In metric space, squared distance is measured using Euclidean distance. This method creates non-linear hyperplanes.

The proposed method makes use of the multinomial NB model, which predicts the content of texts such as emails and newspaper articles. For a given sample, the likelihood of each badge is calculated, and the badge with the highest probability is output. This algorithm was an excellent choice for sentiment analysis of movie reviews because it is commonly used for natural language processing and text data analysis.

IV. CONCLUSION

This essay is largely divided into two parts. One of them focuses on sentiment analysis, while the other is a movie recommendation system that employs content-based recommendations. The paper thoroughly investigates both systems and draws some important findings. The Movie Recommendation System uses the Cosine Similarity algorithm to recommend the best films that are relevant to the movie the user submitted based on a variety of characteristics such as the movie's genre, overview, cast, and ratings. Even after several tests, Cosine Similarity has produced respectable findings and has been quite consistent in terms of film recommendations.

First, I cleaned up the data I downloaded from TMDB. The data was submitted to exploratory data analysis, which included extracting interesting insights, removing missing values, and preparing the data for use in training our model. When a visitor visits our website and types the title of a movie into the search field, they will obtain several autosuggestions related to that film.

In this study, sentiment analysis is also important. It simply attempts to categorize the evaluations as good or unfavourable. For this, two algorithms have been used. The first is NB, whereas the second is SVC. Because there is so much variance in reviews, it is critical to choose the best algorithm for classification. This is the primary justification for using two algorithms to classify reviews. Finally, the experimental results show that SVM has a modest accuracy edge over NB.

Here are some of the study's findings that have been mentioned:

1. Improving Sentiment Analysis Accuracy to better identify ironic or sarcastic judgments.

2. Analysis of reviews in languages other than English sentiment.

3. Movie recommendations tailored to user preferences (cast, genre, year of release, etc.).

Despite its precision, the technology has limitations. One of them is that the system will not propose movies if the user-entered movie is not included in the dataset or if the user does not enter the movie's name exactly as it appears in the dataset. Another disadvantage of emotive analysis is the language barrier that arises. Only reviews posted in English so far will be assessed. The Sentimental Analysis wrongly labels reviews that are sarcastic or acerbic.

REFERENCES

[1] Sang-Min Choi, Sang-Ki Ko, and Yo-Sub Han. "A genre correlation-based movie recommendation algorithm." Expert Systems and Applications, 39.9 (2012), pp. 8079-8085.

[2] George Lekakos and Petros Caravelas, "A Hybrid Approach to Movie Recommendation." Tools for multimedia and applications 36.1 (2008), pp. 55-70 Das,

[3] Debashis, Laxman Sahoo, and Sujoy Datta are the names of three people. "A recommendation system survey." International Magazine160.7 of Computer Applications (2017).

[4] Jiang, Zhang, and colleagues. "Personalized real-time movie recommendation system: Practical prototype and evaluation." 180-191 in Tsinghua Science andTechnology 25.2 (2019).

[5] S. Rajarajeswari and colleagues. "Movie Recommendation System." Computing, Information, and Emerging Research Applications and communication 329-340, Springer Singapore, 2019.

[6] [Ahmed, Muyeed, Mir Tahsin Imtiaz, and Raiyan Khan are the members "Clustering and recommendation system for movies a network for pattern recognition"IEEE 8th Annual Meeting 2018

[7] M. A. Hossain and M. N. Uddin, "A Neural Engine for Movie Recommendation System," in 2018 4th International Conference on Electrical Engineering and

[8] Information and Communication Technology (iCEEiCT), pp. 443-448, doi: 10.1109/CEEICT.2018.8628128.

[9] Dr. Yogesh Kumar Sharma, Monika D.Rokade (2020). Detection of Malicious Network Packet Activity utilizing Deep Learning 29(9s), 2324 - 2331, International Journal of Advanced Science and Technology.

[10] M. A. Hossain and M. N. Uddin, "A Neural Engine for Movie Recommendation System," in 2018 4th International Conference on Electrical Engineering and

Information and Communication Technology (iCEEiCT), pp. 443-448, doi: 10.1109/CEEICT.2018.8628128.

[11] Monika D.Rokade and Dr. Yogesh kumar Sharma, "Deep and machine learning approaches for anomaly-based classification.""Detection of intrusions in imbalanced network traffic," IOSR Journal of Engineering (IOSR JEN), ISSN (e): 2250-3021, ISSN (p): 2278-8719

[12] Dr. Yogesh Kumar Sharma, Monika D.Rokade"MLIDS: A Machine Learning-Based Intrusion Detection Approach"2021 International Conference on Emerging Smart Computing and Datasets for Real-Time Networks IEEE Informatics (ESCI)

[13] Dr. Yogesh Kumar Sharma, Monika D.Rokade (2020). Detection of Malicious Network Packet Activity utilizing Deep Learning 29(9s), 2324 - 2331, International Journal of Advanced Science and Technology.

[14] N. Nassar, A. Jafar, Y. Rahhal, A novel deep multi-criteria collaborative filtering model for recommendation system, Knowl. Based Syst. 187 (2020) 104811 .

i. Beheshti, S. Yakhchi, S. Mousaeirad, S.M. Ghafari, S.R. Goluguri, M.A. Edrisi, Towards cognitive recommender systems, Algorithms 13 (8) (2020) 176 .

[15] S. Sharma, V. Rana, M. Malhotra, Automatic recommendation system based on hy-brid filtering algorithm, Educ. Inf. Technol. 27 (2021) 1–16 .

[16] S.R.S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, B. Venkatesh, Content-based movie recommendation system using genre correlation, in: Smart Intelligent Computing and Applications, Springer, Singapore, 2019, pp. 391–397 .

[17] . M. Yasen, S. Tedmori, Movies reviews sentiment analysis and classification, in: Proceedings of the IEEE Jordan International Joint Conference on Elec-trical Engineering and Information Technology, JEEIT, 2019, pp. 860–865, doi: 10.1109/JEEIT.2019.8717422

[18] N. Rajput, S. Chauhan, Analysis of various sentiment analysis techniques, Int. J. Comput. Sci. Mob. Comput. 8 (2) (2019) 75–79 .

[19] Z. Shaukat, A.A. Zulfiqar, C. Xiao, M. Azeem, T. Mahmood, Sentiment analysis on IMDB using lexicon and neural networks, SN Appl. Sci. 2 (2) (2020) 1–10 .

[20] T. Widiyaningtyas, I. Hidayah, T.B. Adji, User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system, J. Big Data 8 (2021) 52 .

[21] R.H. Singh, S. Maurya, T. Tripathi, T. Narula, G. Srivastav, Movie recommendation system using cosine similarity and KNN, Int. J. Eng. Adv. Technol. (IJEAT) 9 (5) (2020) 2–3 ISSN: 2249 –8958VolumeIssueJune .

[22] S. Kumar, K. De, P.P. Roy, Movie recommendation system using sentiment analysis from microblogging data, IEEE Trans. Comput. Soc. Syst. 7 (4) (2020) 915–923 .

i. Rahman, M.S. Hossen, Sentiment analysis on movie review data using machine learning approach, in: Proceedings of the International Conference on Bangla Speech and Language Processing (ICBSLP), IEEE, 2019, pp. 1–4 .

[23] S. Uddin, A. Khan, M.E. Hossain, M.A. Moni, Comparing different supervised ma-chine learning algorithms for disease prediction, BMC Med. Inf. Decis. Mak. 19 (1) (2019) 1–16 .

[24] S. Ghosh, A. Dasgupta, A. Swetapadma, A study on support vector machine based linear and non-linear pattern classification, in: Proceedings of the International Con-ference on Intelligent Sustainable Systems (ICISS), IEEE, 2019, pp. 24–28 .

[25] K. Dashtipour, M. Gogate, A. Adeel, H. Larijani, A. Hussain, Sentiment analysis of Persian movie reviews using deep learning, Entropy 23 (5) (2021) 596 .

[26] S. Soubraylu, R. Rajalakshmi, Hybrid convolutional bidirectional recurrent neural network based sentiment analysis on movie reviews, Comput. Intell. 37 (2) (2021) 735–757 .