e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com

# MACHINE LEARNING BASED MOVIE RECOMMENDER APLICATION WITH DEPLOYMENT

**S. Yogesh*[1], S. Ganesh Kumar*[2]**

*[1]Student, Department of Data Science & Business system, SRM Institute of science and technology, Kattankulathur

*[2]Professor, Department of Data Science & Business system, SRM Institute of science and technology, Kattankulathur

DOI : https://www.doi.org/10.56726/IRJMETS37481

## ABSTRACT

Nowadays, People watch movies to unwind from their hectic lives. However, Due to the vast selection of films that are available worldwide, choosing a movie to watch is not a simple task. This system combines collaborative filtering with content-based filtering. wherein this system of recommendations suggests a film to the user depending on the material which identified by user's hidden pattern furthermore based on a user's watching pattern similarity with another. In this article, Machine learning based recommender application system would be constructed using Hybrid filtering then would be deployed using Heroku. In this system, Machine learning based recommender application system with Heroku Deployment will be constructed using a both filtering based on content and collaborative filtering approach. Heroku Deployment makes application deployment simple and better user experience.

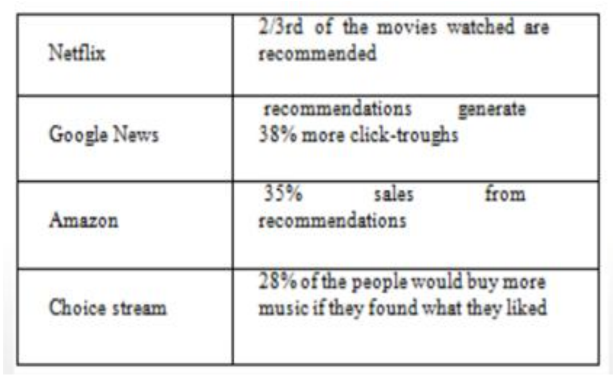
**Keywords—** Collaborative filtering & content-based filtering, visualization, UI interaction

## I. INTRODUCTION

Nowadays, Recommender application systems play a significant part in all the digital platforms, where the recommender application recommends with the help of the user's credits for an item, Netflix, and Amazon Prime use this application to suggest movies. No matter what you do on these websites, a system monitors user activity and proposes things or products that consumers are very likely to interact with.Artificial intelligence-based algorithms used in recommender application sift through all available options to create a customized lists that are interesting and pertinent to a certain user. The major goal of this project is to create a machine learning model that allows the system to easily propose movies to the users depending on their activity and multi users rating pattern similarity.

### A. PURPOSE

This research paper discusses movie recommender application system and the reasoning for going with movie recommender application, problems with old movie recommender application, and a suggestion for a machine learning associated system for movie recommender applications. Many well-known movies suggesting application datasets are already accessible on Kaggle and other platforms. Some of the well-known sources are Movie lens, the TMDB Movie Dataset, and the Netflix dataset. Movie suggestion is used by websites such as Netflix, Amazon Prime, and others increasing the client experience eventually maximizing income or earnings shown in fig.1.1.



| | |
|---|---|
| Netflix | 2/3rd of the movies watched are recommended |
| Google News | recommendations generate 38% more click-troughs |
| Amazon | 35% sales from recommendations |
| Choice stream | 28% of the people would buy more music if they found what they liked |

Fig. 1.1 shows purpose of recommender application.

## B. PROBLEM STATEMENT

The main problem statement of this proposed recommender application system is to build a machine learning based movie recommendation application with deployment based on better movie lens dataset. Data sparsity indicates that the data is widely dispersed and contains null or missing values. Scalability indicates that predicting a large number of ratings items is challenging. Gray sheep denotes difficulties with time and memory. It also requires a significant amount of previously collected data so that the user can generate pertinent recommendations. The problem with new users or items is another name for it. The likelihood of new users getting positive recommendations on new products is low because existing users don't rate or purchase products. Because of the enormous volume of information that is produced every day, scalability issues arise. To calculate suggestions and how rapidly a recommender application can provide a suggestion, a lot of computing power is frequently required. Each active user must have evaluated a select few items from a sizable database. This keeps the remaining goods from getting overlooked by the other users. This results in data scarcity.

## C. STUDY OBJECTIVE

This research paper's primary goal is to build a machine learning based movie recommender application with deployment based on better movie lens dataset. The suggested approach entails building a hybrid recommender application system that blends collaborative filtering with content-based filtering for improved accuracy and better recommendations. It involves Heroku deployment for better user experience. It has better algorithm for better suggestions. It provides more recommendations for users due to its hybrid filtering.

## D. DATASET DESCRIPTION

Dataset used is collected from movie lens site. It has 27800000 ratings and 1200000 tag applications spread among 60000 films. These records were created by 29000 users between the year 1995 and 2019 shown in fig. 1.2. For inclusion, users were selected in random fashion. All the users who were selected to have given at least one movie a rating. Each user is represented by an id, with no other information provided.

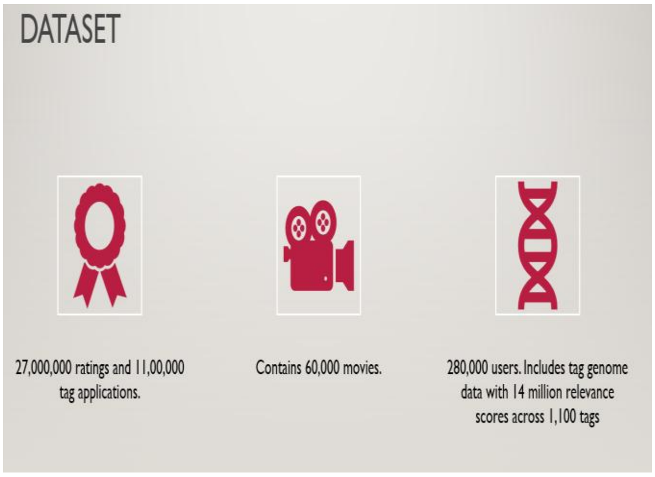

DATASET
27,000,000 ratings and 11,00,000 tag applications.
Contains 60,000 movies.
280,000 users. Includes tag genome data with 14 million relevance scores across 1,100 tags

**Fig. 1.2** shows dataset description.

1) User Ids: The inclusion of users of Movie Lens was done in random manner. Their identification has been hidden. The user ids are consistent between tags and csv ratings.
2) Movies' Ids: Rating file, tags file, movies' id file, and links file all have consistent movies' ids (in other words, throughout these four data files, the same id relates to the same movie.)
3) Ratings Data File Structure: The rating file contains all the ratings. After the header row, this file's lines list users' ratings for individual movies in the following order: users' Id, movies' Id, credits. credits are given in half-star increments on a 5-point scale (0 credits - 5 credits)
4) Tags File Structure: The tags file contains all tags. After the header row, this file's lines list users' ratings for individual movies in the following order: users' Id, movies' Id, credits, timestamp. The order of the lines in this file is user Id first, followed by movie Id inside user. Users generate tags, which are metadata for

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com

movies. Typically, each tag consists of simply one. Each user decides what a particular tag means, is worth, and serves.

5) Movies File Structure: The movies file contains all the movies information. After the header row, this file's lines are formatted as follows, with each line representing a different movie: movie ID, name, and genres. Genres are a pipe-separated list and are selected from the following shown in fig.1.3.

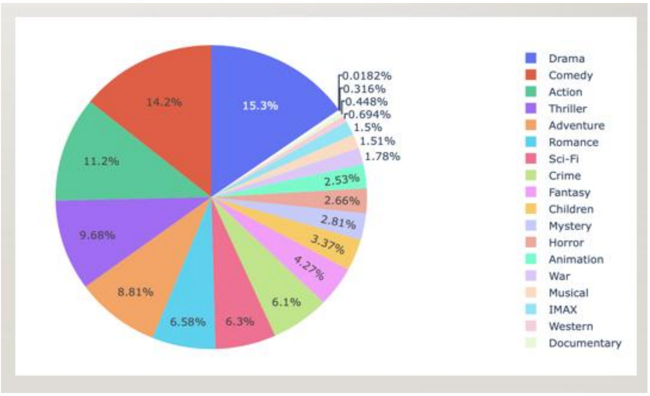


**Fig. 1.3** shows total genres in dataset.

6) Scores File Structure: An updated version of the scores can be found in this data collection. The relevance scores for movie tags are stored in a data structure called the tag genome. A value for each tag in the genome is allocated to each video in the structure, which is a dense matrix. The score was calculated utilizing user-generated content, such as tags, ratings, and textual reviews, using a machine learning technique.

## METHODS

**There are 4 methods of recommender application system. They are:**

7) Popularity based model: It is a type of recommendation system that bases its decisions on what is currently popular or popular in general. These algorithms hunt for goods or movies that are in high demand or are well-liked by customers and immediately suggest them. For instance, the system will learn that a product is the most popular if the majority of users consistently buy it. As a result, it will recommend that product to newly registered users, increasing the likelihood that they will do the same.

8) Filtering Based on Content : It is an alternative recommendation system that functions on the same general principle of content. When a user is watching a movie, the system will hunt for more films with comparable content or that belong to the same genre. When comparing related content shown in fig.1.4, a number of crucial factors are used to calculate similarity. In addition to previous activities or explicit comments, it makes recommendations based on an item's attributes that are similar to the user's favorite items.

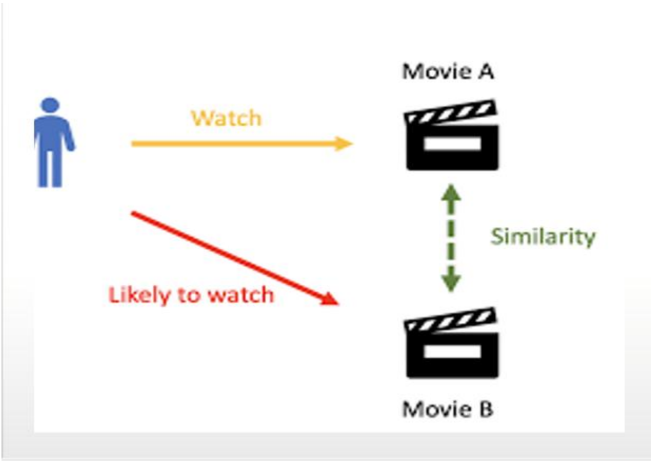


**Fig 1.4** shows filtering based on content.

9) Collaborative filtering: Based on the preferences of a comparable user B, collaborative filtering algorithms can recommend an item to user A. Furthermore, the embeddings can be learned automatically without the

need for feature engineering shown in fig.1.5. Collaborative Filtering seeks out what comparable users want and provides suggestions to group users into groups of comparable sorts and suggest each user based on the preferences of its group.

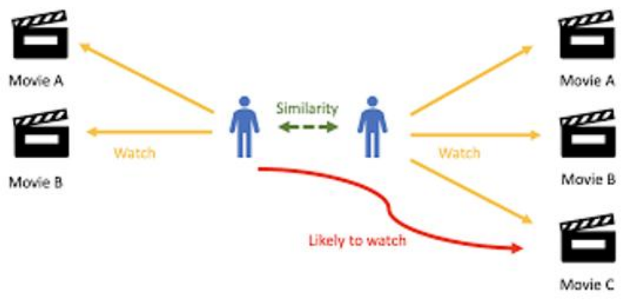


Fig 1.5 shows that collaborative filtering.

10) Hybrid Filtering: Utilizing a hybrid technique of filtering in this work, the movie recommender application system is implemented and shown in fig.1.6. This approach improves system performance by addressing the flaws in each individual algorithm.

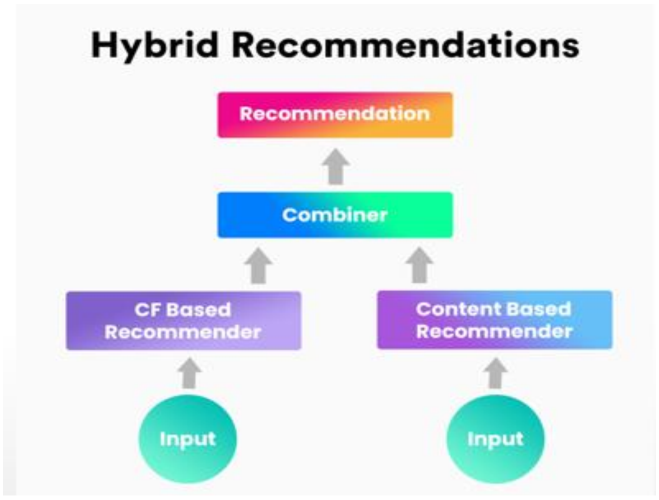


**Fig.1.6** shows hybrid filtering.

## II. RELATED WORK

More number of recommender systems were introduced before. Some of the related systems are:

Sang-Min Choi and colleagues et al. [1] addressed the limitations of the collaborative filtering approach. The authors provided a category-based alternative information to get around this issue. The authors presented a method for choosing movies that is based on. The writers claim that the recently created content includes the category information. Regarding highly regarded content that is frequently seen and new content that is not generally consumed, the suggested answer is neutral. As a result, even brand-new movies can be suggested by the recommendation system.Debashis Das et al. [3] presented several recommendations application types and general facts about each. This research consisted of a survey on recommendation engines. Both personalized and generic recommendation systems were covered by the authors. The advantages and disadvantages of various recommendation systems were also covered by the authors.According to Yeole Madhavi B.1, Rokade Monika D.2, Khatal Sunil S [10]. The author put forth and created a method for creating movie recommendation application based on filtering based on content.Chen and Aickeli [11], The authors presented a concept in which collaborative filtering technology is combined with artificial immune system technology.Basilico and Hofmann [12], The authors suggested a model in which all training data, including historical user-item ratings and properties of objects or users, is integrated into a single framework to learn a prediction function.Md. Akter

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com

Hossain et al. [15] proposed the neural engine-based recommender system, or NERS. An efficient interaction between two datasets was meticulously carried out by the authors. The results of the authors' system, they added, are better than those of other systems since they combined generic and behavior-based datasets. The authors employed three different estimators to contrast their strategy with existing systems.

## III. PROPOSED WORK

Firstly, the dataset is collected from movie lens dataset and the movies, ratings, tags, genome-scores, genome-tags and links dataset are loaded. Once the dataset loading phase is complete, preprocessing the dataset is necessary to separate valuable data from extraneous data. Later, text from that feature need to be transformed into vectors. There are 2 vectors transformation library used. They are count vectorizer, td-idf vectorizer which are available in the scikit-learn library. Using those converted vectors we have perform vector similarity. Depending on the system design given below, users receive recommendations based on hidden pattern and users' s similarity. Problems including missing data values and data imbalance circumstances may arise throughout this operation. Visualization of the dataset is essential to determine which features are useful for recommendation.In order to create a similarity matrix and determine how similar the videos are. The smaller the distance the maximum is the similarity and vice versa. Then based on cosine similarity content-based filtering is built. Next step is to build collaborative filtering using cosine distance with adjusted vectors called "Centered cosine". Both models are combined using Heroku deployment with the help of stream-lit and pickle. User Interface can be adjusted for better user experience.

### A. Architectural flow

The architectural flow of the proposed system follows Dataset collection, data preprocessing, model building, NLTK usage, Hybrid filtering using content-based and collaborative based and then deployed using Heroku shown in fig.3.1

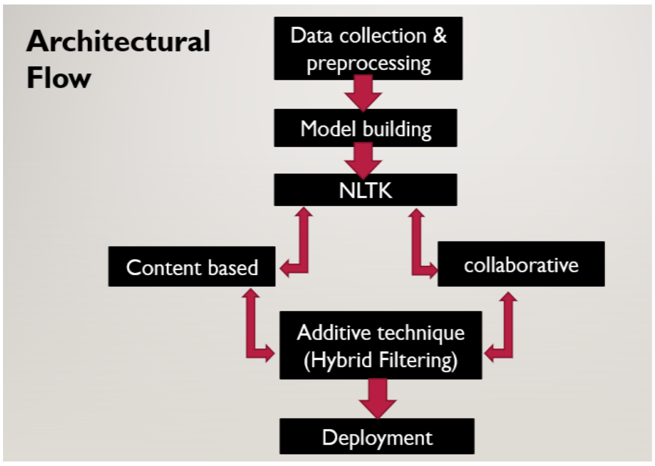


**Fig.3.1** shows architectural flow of the proposed system.

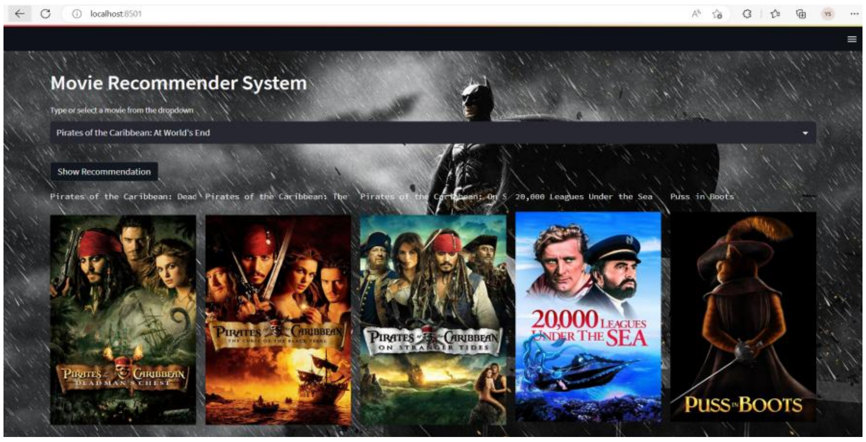

localhost:8501
Movie Recommender System
Type or select a movie from the dropdown
Pirates of the Caribbean: At World's End
Show Recommendation
Pirates of the Caribbean: Dead
Pirates of the Caribbean: The
Pirates of the Caribbean: On S
20,000 Leagues Under the Sea
Puss in Boots

**Fig.3.2** shows recommender application system.

## IV. CONCLUSION

In this proposed system, Movie Recommender application has constructed for movie recommendations to users. It enables a user to choose from a list of attributes and then recommends movies to him based on content-based criteria. and collaborative based filtering. Finally results from hybrid model are deployed for better user experience.

## V. 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,Debashis, Laxman Sahoo, and Sujoy Datta are the names of three people. "A recommendation system survey." International Magazine160.7 of ComputerApplications (2017).

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

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

[5] 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.

[6] M. A. Hossain and M. N. Uddin, "A Neural Engine for Movie Recommendation System," in 2018 4th International Conference on Electrical Engineering andInformation and Communication Technology (iCEEiCT), pp. 443-448, doi: 10.1109/CEEICT.2018.8628128.

[7] 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.

[8] 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.

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

[10] 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).

[11] N. Immaneni, I. Padmanaban, B. Ramasubramanian and R. Sridhar, "A meta-level hybridization approach to personalized movie recommendation," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 2193-2200, doi: 10.1109/ICACCI.2017.8126171.