International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER SSRN

IJIEMR Transactions, online available on 12th Dec 2022. Link :http://www.ijiemr.org/downloads.php?vol=Volume-11&issue=Issue 12

DOI: 10.48047/IJIEMR/V11/ISSUE 12/58

Title **APPLICATION OF ARTIFICIAL INTELLIGENCE RECOMMENDATION ENGINE FOR MOVIES**

Volume 11, ISSUE 12, Pages: 455-464

Paper Authors
M. Mounika, J. Varshini Reddy, A. Shashi Preetam, N. Shusheel, M. Hemsai Reddy

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

APPLICATION OF ARTIFICIAL INTELLIGENCE RECOMMENDATION ENGINE FOR MOVIES

M. Mounika[1], J. Varshini Reddy[2], A. Shashi Preetam[2], N. Shusheel[2], M. Hemsai Reddy[2]

[1]Professor, [2]UG Student, [1,2]Department of Information Technology

[1,2]Malla Reddy Engineering College and Management Sciences, Medchal, Telangana

ABSTRACT

Providing a useful suggestion of products to online users to increase their consumption on websites is the goal of many companies nowadays. People usually select or purchase a new product based on some friend's recommendations, comparison of similar products or feedbacks from other users. To do all these tasks automatically, a recommender system must be implemented. The recommender systems are tools that provide suggestions that best suit the client's needs, even when they are not aware of it. That offers of personalized content are based on past behavior and it hooks the customer to keep coming back to the website. In this paper, a movie recommendation mechanism within MovieLens will be built. The dataset that was used here consists of over 17K movies and 500K+ customers.

Keywords: Recommendation system, Movie recommendation, MovieLens dataset.

1. INTRODUCTION

Recommendation systems help users find and select items (e.g., books, movies, restaurants) from the huge number available on the web or in other electronic information sources. Given a large set of items and a description of the user's needs, they present to the user a small set of the items that are well suited to the description. Similarly, a movie recommendation system provides a level of comfort and personalization that helps the user interact better with the system and watch movies that cater to his needs. Providing this level of comfort to the user is primary motivation in opting for movie recommendation system as my Project. The chief purpose of our system is to recommend movies to its users based on their viewing history and ratings that they provide. The system will also recommend various E-commerce companies to publicize their products to specific customers based on the genre of movies they like. Personalized recommendation engines help millions of people narrow the universe of potential films to fit their unique tastes. Collaborative filtering and content-based filtering are the prime approaches to provide recommendation to users. Both are best applicable in specific scenarios because of their respective ups and downs. In this paper we have proposed a mixed approach such that both the algorithms complement each other thereby improving performance and accuracy of the of our system.

The main types of recommender algorithms are explained below, and we will choose the best one for implementation.

- Popularity-Based Recommendation System
- Collaborative Filtering
- Content-based filtering
- Hybrid Filtering

2. LITERATURE SURVEY

Ahuja et al. [1] used the various tools and techniques to build recommender systems. Various algorithms such as K-Means Clustering, KNN, Collaborative Filtering, Content-Based Filtering have been described in detail. Further, after studying different types of machine learning algorithms, there is a clear picture of where to apply which algorithm in different areas of industries such as recommender systems, e-commerce, etc. Then there is an illustration of how implementations and working of the proposed system are used for the implementation of the movie recommender system. Various building blocks of the proposed system such as Architecture, Process Flow, Pseudo Code, Implementation and Working of the System is described in detail.

Awan et al. [2] implemented a movie recommendation system based on a collaborative filtering approach using the alternating least squared (ALS) model to predict the best-rated movies. This proposed system used the last search data of a user regarding movie category and references this to instruct the recommender engine, thereby making a list of predictions for top ratings. The proposed study used a model-based approach of matrix factorization, the ALS algorithm along with a collaborative filtering technique, which solved the cold start, sparse, and scalability problems. This work performed experimental analysis and successfully obtained minimum root mean squared errors (oRMSEs) of 0.8959 to 0.97613, approximately. Moreover, this proposed movie recommendation system showed an accuracy of 97% and predicted the top 1000 ratings for movies.

Walek et al. [3] proposed a monolithic hybrid recommender system called Predictory, which combined a recommender module composed of a collaborative filtering system (using the SVD algorithm), a content-based system, and a fuzzy expert system. The proposed system served to recommend suitable movies. The system worked with favorite and unpopular genres of the user, while the final list of recommended movies is determined using a fuzzy expert system, which evaluates the importance of the movies. The expert system works with several parameters – average movie rating, number of ratings, and the level of similarity between already rated movies.

Tahmasebi et al. [4] introduced a hybrid social recommender system utilizing a deep autoencoder network. The proposed approach employed collaborative and content-based filtering, as well as users' social influence. The social influence of each user is calculated based on his/her social characteristics and behaviors on Twitter. For the evaluation purpose, the required datasets have been collected from MovieTweetings and Open Movie Database. The evaluation results showed that the accuracy and effectiveness of the proposed approach have been improved compared to the other state-of-the-art methods.

Wang et al. [5] proposed a movie recommendation framework based on a hybrid recommendation model and sentiment analysis on Spark platform to improve the accuracy and timeliness of mobile movie recommender system. In the proposed approach, first used a hybrid

recommendation method to generate a preliminary recommendation list. Then sentiment analysis is employed to optimize the list. Finally, the hybrid recommender system with sentiment analysis is implemented on Spark platform. The hybrid recommendation model with sentiment analysis outperforms the traditional models in terms of various evaluation criteria. This proposed method makes it convenient and fast for users to obtain useful movie suggestions.

Yassine et al. [6] proposed a new intelligent recommender system that combines collaborative filtering (CF) with the popular unsupervised machine learning algorithm K-means clustering. Also, this paper used certain user demographic attributes such as the gender and age to create segmented user profiles, when items (movies) are clustered by genre attributes using K-means and users are classified based on the preference of items and the genres they prefer to watch. To recommend items to an active user, Collaborative Filtering approach then is applied to the cluster where the user belongs. Following the experimentation for well-known movies, this work shown that the proposed system satisfies the predictability of the CF algorithm in GroupLens. In addition, this proposed system improved the performance and time response speed of the traditional collaborative Filtering technique and the Content-Based technique too.

Kumar et al. [7] proposed a hybrid RS for the movies that leverage the best of concepts used from CF and CBF along with sentiment analysis of tweets from microblogging sites. The purpose to use movie tweets is to understand the current trends, public sentiment, and user response of the movie. Experiments conducted on the public database have yielded promising results.

Reddy et al. [8] built the recommendation system on the type of genres that the user might prefer to watch. The approach adopted to do so is content-based filtering using genre correlation. The dataset used for the system is Movie Lens dataset. The data analysis tool used is R.

Keshava et al. [9] aimed to recommend applicable objects to a consumer-based totally on ancient data. If a movie is rated excessive by means of a consumer who also watched the movie you are watching now, it's miles possibly to show up inside the recommendations. The films with the highest overall scores are likely to be enjoyed by way of nearly everyone. The algorithm which does all these features is called CineMatch. For personal users, it also learns from the conduct of the person to higher expect a movie the consumer is anticipated to be fascinated in.

Mahata et al. [10] proposed the Intelligent movie recommender system that combines the concept of Human-Computer Interaction and Machine Learning. The proposed system is a subclass of information filtering system that captures facial feature points as well as emotions of a viewer and suggests them movies accordingly. It recommends movies best suited for users as per their age and gender and as per the genres they prefer to watch. The recommended movie list is created by the cumulative effect of ratings and reviews given by previous users. A neural network is trained to detect genres of movies like horror, comedy based on the emotions of the user watching the trailer.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

3. PROPOSED SYSTEM

3.1 Dataset Description

Index(['adult', 'belongs_to_collection', 'budget', 'genres', 'homepage', 'id',
'imdb_id', 'original_language', 'original_title', 'overview',
'popularity', 'poster_path', 'production_companies',
'production_countries', 'release_date', 'revenue', 'runtime',
'spoken_languages', 'status', 'tagline', 'title', 'video',
'vote_average', 'vote_count'],
dtype='object')

Features

adult: Indicates if the movie is X-Rated or Adult.

belongs_to_collection: A stringified dictionary that gives information on the movie series the particular film belongs to.

budget: The budget of the movie in dollars.

genres: A stringified list of dictionaries that list out all the genres associated with the movie.

homepage: The Official Homepage of the move.

id: The ID of the move.

imdb_id: The IMDB ID of the movie.

original_language: The language in which the movie was originally shot in.

original_title: The original title of the movie.

overview: A brief blurb of the movie.

popularity: The Popularity Score assigned by TMDB.

poster_path: The URL of the poster image.

production_companies: A stringified list of production companies involved with the making of the movie.

production_countries: A stringified list of countries where the movie was shot/produced in.

release_date: Theatrical Release Date of the movie.

revenue: The total revenue of the movie in dollars.

runtime: The runtime of the movie in minutes.

spoken_languages: A stringified list of spoken languages in the film.

status: The status of the movie (Released, To Be Released, Announced, etc.)

tagline: The tagline of the movie.

title: The Official Title of the movie.

video: Indicates if there is a video present of the movie with TMDB.

vote_average: The average rating of the movie.

vote_count: The number of votes by users, as counted by TMDB.

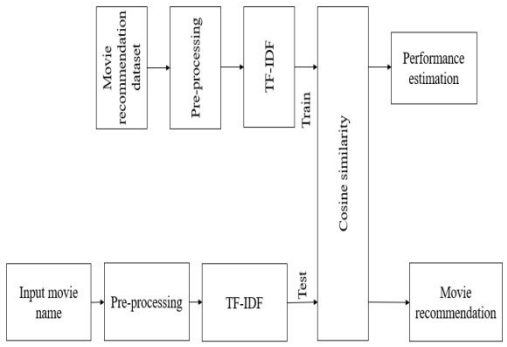


Fig. 1: Block diagram of proposed system.

3.2 Movie Recommendation Dataset

The content-based movie recommendation system used tmdb_5000_movies and tmdb_5000_credits datasets, which are publicly available. The movie hit prediction and target audience prediction module make use of the IMDb rating dataset. Use of two different databases (TMDb and IMDb) creates synchronization problems since they use two separate movies ID. The proposed

system uses the link small dataset to merge two databases. The tmdb_5000_movie data set is consistent with 4803 movie data. It contains 20 movie attributes. That is its accountants the movie with published year from 1916 to 2017. From the 20 attributes, we have chosen only 4 attributes. Selected attributes are keywords, overview, tagline and genre. Attributes like budget, revenue, and release year are very much dependent on time, and since we are considering movie of more than a hundred years of span, these attributes could not be selected. Other attributes like runtime, spoken language, original title, homepage is also irrelevant to the proposed system.

3.3 Pre-processing

Data Pre-processing in Machine learning

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

Why do we need Data Pre-processing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

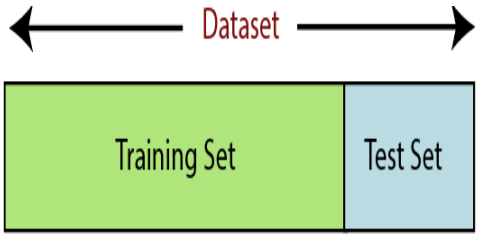
- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

3.3.1 Splitting the Dataset into the Training set and Test set

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model.

Supposeif we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:



Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

3.4 TF-IDF Feature extraction

TF-IDF which stands for Term Frequency – Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document. Let's take an example, we have a string or Bag of Words (BOW) and we have to extract information from it, then we can use this approach.

The tf-idf value increases in proportion to the number of times a word appears in the document but is often offset by the frequency of the word in the corpus, which helps to adjust with respect to the fact that some words appear more frequently in general. TF-IDF use two statistical methods, first is Term Frequency and the other is Inverse Document Frequency. Term frequency refers to the total number of times a given term t appears in the document doc against (per) the total number of all words in the document and The inverse document frequency measure of how much information the word provides. It measures the weight of a given word in the entire document. IDF show how common or rare a given word is across all documents. TF-IDF can be computed as tf * idf

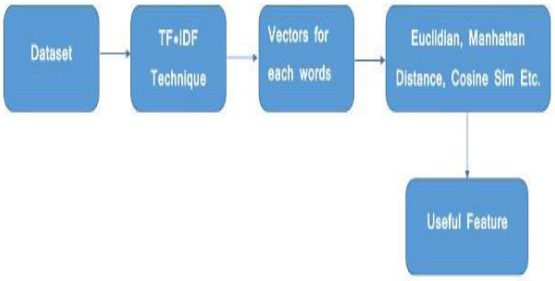


Fig. 2: TF-IDF block diagram.

TF-IDF do not convert directly raw data into useful features. Firstly, it converts raw strings or dataset into vectors and each word has its own vector. Then we'll use a particular technique for retrieving the feature like Cosine Similarity which works on vectors, etc.

Terminology:

t — term (word)

d — document (set of words)

N — count of corpus

corpus — the total document set

Term Frequency (TF): Suppose we have a set of English text documents and wish to rank which document is most relevant to the query, "Data Science is awesome!" A simple way to start out is by eliminating documents that do not contain all three words "Data" is", "Science", and "awesome", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document; the number of times a term occurs in a document is called its term frequency. The weight of a term that occurs in a document is simply proportional to the term frequency.

$$tf(t,d) = count\ of\ t\ in\ d \ / number\ of\ words\ in\ d$$

Document Frequency: This measures the importance of document in whole set of corpus, this is very similar to TF. The only difference is that TF is frequency counter for a term t in document d, whereas DF is the count of occurrences of term t in the document set N. In other words, DF is the number of documents in which the word is present. We consider one occurrence if the term consists in the document at least once, we do not need to know the number of times the term is present.

$$df(t) = occurrence\ of\ t\ in\ documents$$

Inverse Document Frequency (IDF): While computing TF, all terms are

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing IDF, an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely. **The** IDF is the inverse of the document frequency which measures the informativeness of term t. When we calculate IDF, it will be very low for the most occurring words such as stop words (because stop words such as "is" is present in almost all of the documents, and N/df will give a very low value to that word). This finally gives what we want, a relative weightage.

$$idf(t) = N/df$$

Now there are few other problems with the IDF, in case of a large corpus,say 100,000,000 , the IDF value explodes , to avoid the effect we take the log of idf . During the query time, when a word which is not in vocab occurs, the df will be 0. As we cannot divide by 0, we smoothen the value by adding 1 to the denominator.

$$idf(t) = log(N/(df + 1))$$

The TF-IDF now is at the right measure to evaluate how important a word is to a document in a collection or corpus. Here are many different variations of TF-IDF but for now let us concentrate on this basic version.

$$tf - idf(t,d) = tf(t,d) * log(N/(df + 1))$$

Implementing TF-IDF: To make TF-IDF from scratch in python, let's imagine those two sentences from different document:

first_sentence: "Data Science is the sexiest job of the 21st century".

second_sentence: "machine learning is the key for data science".

3.5 Cosine Similarity

In Data Mining, similarity measure refers to distance with dimensions representing features of the data object, in a dataset. If this distance is less, there will be a high degree of similarity, but when the distance is large, there will be a low degree of similarity.

3.6 Advantages of proposed system

- The cosine similarity is beneficial because even if the two similar data objects are far apart by the Euclidean distance because of the size, they could still have a smaller angle between them. Smaller the angle, higher the similarity.
- When plotted on a multi-dimensional space, the cosine similarity captures the orientation (the angle) of the data objects and not the magnitude.

4. RESULTS AND DISCUSSION

Data index

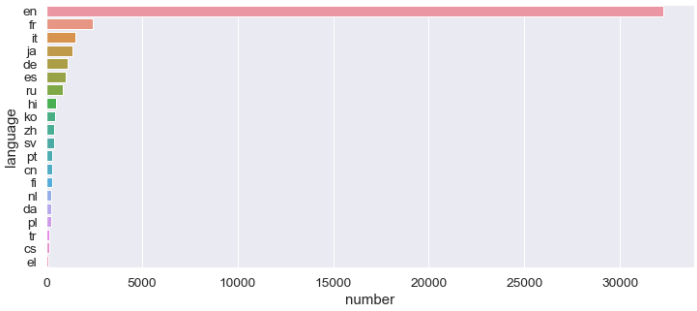
```
Index(['adult', 'belongs_to_collection', 'budget', 'genres', 'homepage', 'id',
       'imdb_id', 'original_language', 'original_title', 'overview',
       'popularity', 'poster_path', 'production_companies',
       'production_countries', 'release_date', 'revenue', 'runtime',
       'spoken_languages', 'status', 'tagline', 'title', 'video',
       'vote_average', 'vote_count'],
      dtype='object')
```

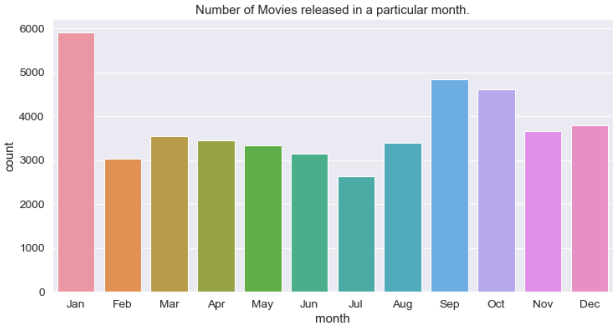
Features

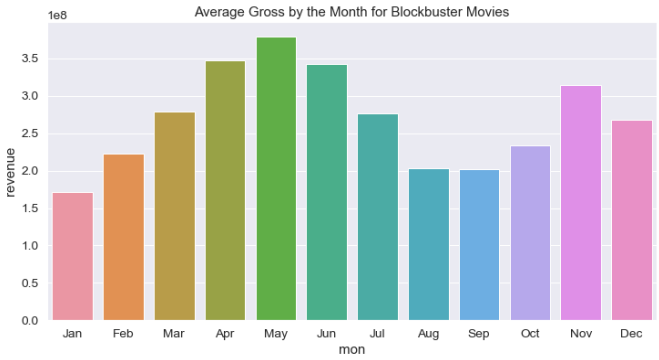
- **adult:** Indicates if the movie is X-Rated or Adult.
- **belongs_to_collection:** A stringified dictionary that gives information on the movie series the particular film belongs to.
- **budget:** The budget of the movie in dollars.

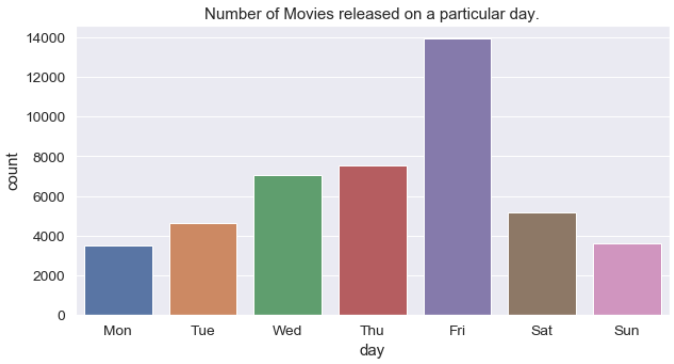
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

- **genres:** A stringified list of dictionaries that list out all the genres associated with the movie.
- **homepage:** The Official Homepage of the move.
- **id:** The ID of the move.
- **imdb_id:** The IMDB ID of the movie.
- **original_language:** The language in which the movie was originally shot in.
- **original_title:** The original title of the movie.
- **overview:** A brief blurb of the movie.
- **popularity:** The Popularity Score assigned by TMDB.
- **poster_path:** The URL of the poster image.
- **production_companies:** A stringified list of production companies involved with the making of the movie.
- **production_countries:** A stringified list of countries where the movie was shot/produced in.
- **release_date:** Theatrical Release Date of the movie.
- **revenue:** The total revenue of the movie in dollars.
- **runtime:** The runtime of the movie in minutes.
- **spoken_languages:** A stringified list of spoken languages in the film.
- **status:** The status of the movie (Released, To Be Released, Announced, etc.)
- **tagline:** The tagline of the movie.
- **title:** The Official Title of the movie.
- **video:** Indicates if there is a video present of the movie with TMDB.
- **vote_average:** The average rating of the movie.
- **vote_count:** The number of votes by users, as counted by TMDB.

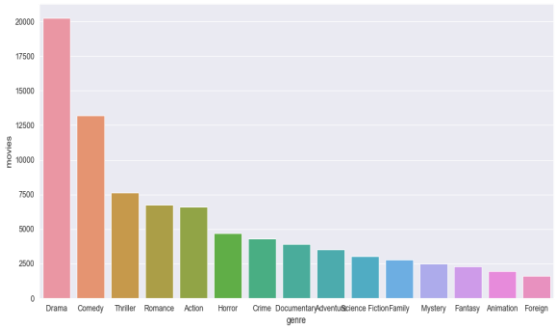








International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org



5. CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

In this work, we propose the fully Content-based Movie Recommender system. The proposed system only uses the content data of movie as the training dataset. Furthermore, the proposed method takes advantages of the Cosine similarity model to extract features from the content of movies and transform the textual content data info feature vectors which can keep linear relationship semantically. We conduct the evaluation with dataset.

5.2 Future scope

Future scope includes that try to use more metadata as input feature and use different ratings to weight the dataset instead of IMDb ratings. It highlights the filtering criteria in the recommender systems, algorithms implemented in movie recommender systems, the performance measurement criteria, the challenges in implementation, and recommendations for future research. Some of the most popular machine learning algorithms used in movie recommender systems such as K-means clustering, principal component analysis, and self-organizing maps with principal component analysis are discussed in detail. Special emphasis is given to research works performed using metaheuristic-based recommendation systems. The research aims to bring to light the advances made in developing the movie recommender systems, and what needs to be performed to reduce the current challenges in implementing the feasible solutions.

REFERENCES

[1] R. Ahuja, A. Solanki and A. Nayyar, "Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019, pp. 263-268, doi: 10.1109/CONFLUENCE.2019.8776969.

[2] Awan MJ, Khan RA, Nobanee H, Yasin A, Anwar SM, Naseem U, Singh VP. A Recommendation Engine for Predicting Movie Ratings Using a Big Data Approach. Electronics. 2021; 10(10):1215. https://doi.org/10.3390/electronics10101215.

[3] B. Walek, V. Fojtik, “A hybrid recommender system for recommending relevant movies using an expert system”, Expert Systems with Applications, Volume 158, 2020, 113452, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113452.

[4] Tahmasebi, H., Ravanmehr, R. & Mohamadrezaei, R. Social movie recommender system based on deep autoencoder network using Twitter data. Neural Comput & Applic 33, 1607–1623 (2021). https://doi.org/10.1007/s00521-020-05085-1.

[5] Y. Wang, M. Wang, W. Xu, "A Sentiment-Enhanced Hybrid Recommender System for Movie Recommendation: A Big Data Analytics Framework", Wireless Communications and Mobile Computing, vol. 2018, Article ID 8263704, 9 pages, 2018. https://doi.org/10.1155/2018/8263704.

[6] A. Yassine, L. Mohamed, M. Al Achhab, "Intelligent recommender system based on unsupervised machine learning and demographic attributes", Simulation Modelling Practice and Theory, Volume 107, 2021, 102198, ISSN 1569-190X, https://doi.org/10.1016/j.simpat.2020.102198.

[7] S. Kumar, K. De and P. P. Roy, "Movie Recommendation System Using Sentiment Analysis from Microblogging Data," in IEEE Transactions on Computational Social Systems, vol. 7, no. 4, pp. 915-923, Aug. 2020, doi: 10.1109/TCSS.2020.2993585.

[8] Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., Venkatesh, B. (2019). Content-Based Movie Recommendation System Using Genre Correlation. In: Satapathy, S., Bhateja, V., Das, S. (eds) Smart Intelligent Computing and Applications. Smart Innovation, Systems and Technologies, vol 105. Springer, Singapore. https://doi.org/10.1007/978-981-13-1927-3_42

[9] Keshava, M. & Reddy, P. & Srinivasulu, S. & Naik, B.. (2020). Machine Learning Model for Movie Recommendation System. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS040741.

[10] Mahata, A., Saini, N., Saharawat, S., Tiwari, R. (2017). Intelligent Movie Recommender System Using Machine Learning. In: Basu, A., Das, S., Horain, P., Bhattacharya, S. (eds) Intelligent Human Computer Interaction. IHCI 2016. Lecture Notes in Computer Science (), vol 10127. Springer, Cham. https://doi.org/10.1007/978-3-319-52503-7_8.