

# Movie Recommender Model Using Machine Learning Approaches

Raja Marappan\*, S. Bhaskaran

School of Computing, SASTRA Deemed University, Thanjavur, India.

**How to cite this paper:** Raja Marappan, S. Bhaskaran. (2022). Movie Recommender Model Using Machine Learning Approaches. *The Educational Review, USA*, 6(7), 317-319.  
DOI: 10.26855/er.2022.07.006

**Received:** June 20, 2022  
**Accepted:** July 18, 2022  
**Published:** August 2, 2022

**Corresponding author:** Raja Marappan, School of Computing, SASTRA Deemed University, Thanjavur, India.  
**Email:** raja\_csmath@cse.sastra.edu

## Abstract

There are different suggestions or information filtering systems developed to solve real-world problems. The recommendation systems are performing the role of information filtering in different scenarios. To provide a better recommendation, different soft computing strategies such as machine learning and evolutionary computing are applied. The recommendation systems fulfill the requirements of the users on time. Concerning organizations, the company likes to keep their users long on the platforms to maximize the profit. Better recommendations are expected to generate positive feedback for both users and organizations. One of the most widely used real-world applications is the movies in which the users are expecting better information filtering. The movie recommender system is expected to predict the preferred items of the user based on the similarity ratings of other people. This article focuses on developing the movie recommendation model using machine learning approaches—the count vectorizer and nearest neighbors approaches.

## Keywords

Recommender system, movie recommender, machine learning, similarity ratings

## 1. Introduction

In general, the recommendation systems fulfill the requirements of the users on time. Concerning organizations, the company like to keep their users long on the platforms to maximize the profit (Harper et al, 2015). Hence, better recommendations are expected to generate positive feedback for both users and organizations (Marappan et al, 2022). The kinds of recommendation systems are sketched in Figure 1. The high correlation in recommendation is shown in Figure 2 and the content-based filtering is depicted in Figure 3 [3-4] (Raja Marappan et al., 2022).

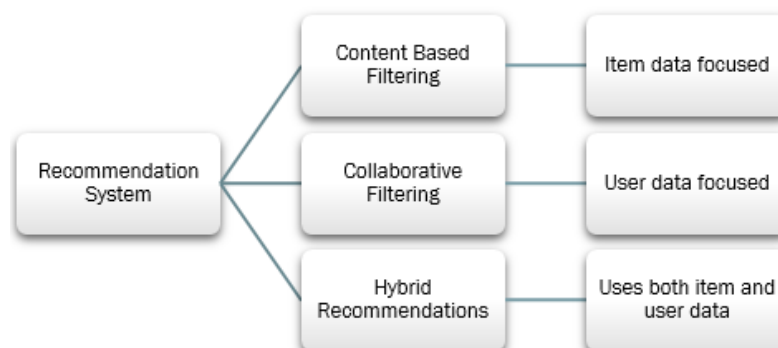


Figure 1. Types of recommendation systems.

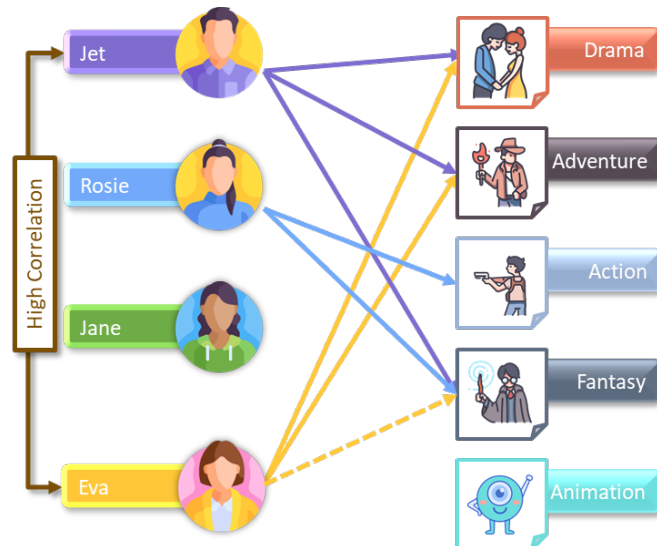


Figure 2. The high correlation in recommendation.

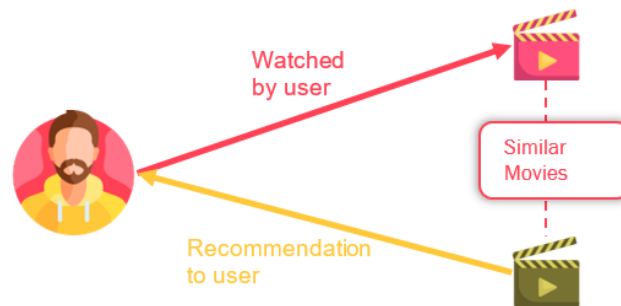


Figure 3. Content-based filtering.

## 2. Movie Recommendation using ML

The ML approach for movie recommendation uses the TMDB API dataset (Bhaskaran et al, 2020). The structure of the movie dataset is shown in Table 1. The dataset is taken from Dataset: <https://www.themoviedb.org/documentation/api>.

Table 1. Movie dataset

**Title:** Movie Title.  
**Overview:** Abstract of the Movie.  
**Popularity:** Movie popularity rating as per TMDB.  
**Vote\_average:** Votes average out of 10.  
**Vote\_count:** Number of votes from the users.  
**Release\_date:** Date of release of the movie.  
**Keywords:** Keywords for the movie by TMDB in the list.  
**Genres:** Movie Genres in the list.  
**Cast:** Cast of the movie on the list.  
**Crew:** Crew of the movie in the list.

The model development to recommend the movies using the count vectorizer and nearest neighbors approaches involves the following steps:

- 1) Read the movie dataset.
- 2) Perform the data cleaning operation and remove the unnamed columns.
- 3) Change the new datatypes for the columns.
- 4) Define a function to analyze the data.
- 5) Construct a chart to identify the popular genres.
- 6) Construct a chat to identify top genres based on a specific requirement.
- 7) Determine the ratings based on vote measures and popularity.

- 8) Define a text for the count vectorizer.
- 9) Evaluate the cosine similarity scores of all movies.
- 10) Sort the top 15 movies based on their popularity.
- 11) Apply nearest neighbors, create genre columns and train the model.
- 12) Define a function to recommend the top 15 movies.
- 13) Compare the results of the proposed approaches.

### 3. Conclusions & Future Work

In this research, the movie recommendation model is developed using a count vectorizer and nearest neighbors approaches. In the future, the proposed recommendation model is extended using different soft computing models (R. Marappan et al., 2013; G. Sethumadhavan et al., 2013; Marappan et al., 2018-2021).

### References

- Bhaskaran, S., Marappan, R., Santhi, B. (2020). Design and Comparative Analysis of New Personalized Recommender Algorithms with Specific Features for Large Scale Datasets. *Mathematics*, 2020, 8, 1106. <https://doi.org/10.3390/math8071106>.
- Bhaskaran, S., Marappan, R., Santhi, B. (2021). Design and Analysis of a Cluster-Based Intelligent Hybrid Recommendation System for E-Learning Applications. *Mathematics*, 2021, 9, 197. <https://doi.org/10.3390/math9020197>.
- G. Sethumadhavan and R. Marappan. (2013). A genetic algorithm for graph coloring using single parent conflict gene crossover and mutation with conflict gene removal procedure. *2013 IEEE International Conference on Computational Intelligence and Computing Research*, 2013, pp. 1-6. doi: 10.1109/ICCIC.2013.6724190.
- Harper, F. M. and Konstan, J. A. (2015). The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 2015, 5, 19:1-19:19. doi:10.1145/2827872.
- Marappan, R. and Bhaskaran, S. (2022). Movie Recommendation System Modeling Using Machine Learning. *International Journal of Mathematical, Engineering, Biological and Applied Computing*, 2022, 1(1), 12-16. DOI: 10.31586/ijmebac.2022.291.
- Marappan, R. and Sethumadhavan, G. (2018). Solution to Graph Coloring Using Genetic and Tabu Search Procedures. *Arab J SciEng.*, 43, 525-542 (2018). <https://doi.org/10.1007/s13369-017-2686-9>.
- Marappan, R. and Sethumadhavan, G. (2020). Complexity Analysis and Stochastic Convergence of Some Well-known Evolutionary Operators for Solving Graph Coloring Problem. *Mathematics*, 2020, 8, 303. <https://doi.org/10.3390/math8030303>.
- Marappan, R. and Sethumadhavan, G. (2021). Solving Graph Coloring Problem Using Divide and Conquer-Based Turbulent Particle Swarm Optimization. *Arab J SciEng.* (2021). <https://doi.org/10.1007/s13369-021-06323-x>.
- R. Marappan and G. Sethumadhavan. (2013). A New Genetic Algorithm for Graph Coloring. *2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation*, 2013, pp. 49-54. doi: 10.1109/CIMSim.2013.17.
- Raja Marappan and S. Bhaskaran. (2022). Datasets Finders and Best Public Datasets for Machine Learning and Data Science Applications. *COJ Rob Artificial Intel.*, 2(1). COJRA. 000530. 2022.