*Mini Review*

# Recommender System for Movielens Datasets using an Item-based Collaborative Filtering in Python

**Raja Marappan** *ID

School of Computing, SASTRA Deemed University, Thanjavur, India

*Correspondence: Raja Marappan (professor.m.raja@gmail.com)

**Abstract:** Everyone likes movies irrespective of color, gender, age, location, and race. The most important thing is how the users are getting our unique combinations of choices concerning the preferences of the movies. This article focuses on the creation of a movie recommendation system using item-based collaborative filtering.

**Keywords:** Movie Recommendation, Recommendation System, Movies Preferences, Collaborative Filtering

## 1. Introduction

The recommendation system is a program that filters the information to predict the preference or rating of the item. In the case of movies, the system filters and predicts the movies that the users would prefer based on their preferences [1,2].

## 2. Movie Recommendation System

The movie recommendation system recommends the top movies using item-based collaborative filtering (CF) [1,2]. The dataset applied is the Movielens small dataset as shown in Table 1. The movie dataset has movie id and genre columns. The rating dataset has the columns: user id, movie id, and rating as shown in Table 2 [3-7].

The steps in the proposed model using Python are as follows:

1. Import the required libraries.
2. Specify the CSV file path and import the dataset.
3. Search the files using the command: dataframe.head() to print the dataset rows.
4. Construct a new data frame with user id and movie id columns.
5. Remove the movies with minimal ratings. Assume that at least 15 votes per movie and the user should vote for at least 50 movies.
6. Apply the function csr_matrix to reduce the sparsity.
7. Apply the KNN method to calculate the cosine similarity measure.
8. Find similar movies and sort them out.
9. Identify the top movies.

**Table 1. Movielens datasets – ML-100K and ML-1M**

|                           | ML-100K  | ML-1M     |
|---------------------------|----------|-----------|
| Number of users           | 943      | 6,040     |
| Number of movies          | 1,682    | 3,952     |
| Number of ratings         | 10,0000  | 1,000,209 |
| Number of all genres      | 19       | 18        |
| Average number of genres  | 1.7      | 1.6       |
| Rating scales             | 1-5      | 1-5       |

**Table 2. Rating dataset**

|   | userId | movieId | rating | timestamp  |
|---|--------|---------|--------|------------|
| 0 | 1      | 110     | 1.0    | 1425941529 |
| 1 | 1      | 147     | 4.5    | 1425942435 |
| 2 | 1      | 858     | 5.0    | 1425941523 |
| 3 | 1      | 1221    | 5.0    | 1425941546 |
| 4 | 1      | 1246    | 5.0    | 1425941556 |

### 3. Conclusions & Future Work

This article models the creation of a movie recommendation system using an item-based CF. in the future other soft computing approaches can be integrated to create a better recommendation of movies [8-12].

### References

[1] Dataset: https://grouplens.org/datasets/movielens/

[2] Dataset: https://www.themoviedb.org/documentation/api

[3] Harper, F.M.; Konstan, J.A. The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* 2015, *5*, 19:1–19:19, doi: 10. 1145/2827872.

[4] Marappan, R., & 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

[5] Raja Marappan, S Bhaskaran. Datasets Finders and Best Public Datasets for Machine Learning and Data Science Applications. COJ Rob Artificial Intel. 2(1). COJRA. 000530. 2022.

[6] Bhaskaran, S.; Marappan, R.; Santhi, B. 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

[7] Bhaskaran, S.; Marappan, R.; Santhi, B. 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

[8] Marappan, R., Sethumadhavan, G. Solving Graph Coloring Problem Using Divide and Conquer-Based Turbulent Particle Swarm Optimization. Arab J Sci Eng (2021). https://doi.org/10.1007/s13369-021-06323-x

[9] Marappan, R.; Sethumadhavan, G. 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

[10] Marappan, R., Sethumadhavan, G. Solution to Graph Coloring Using Genetic and Tabu Search Procedures. Arab J Sci Eng 43, 525–542 (2018). https://doi.org/10.1007/s13369-017-2686-9

[11] R. Marappan and G. Sethumadhavan, "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.

[12] G. Sethumadhavan and R. Marappan, "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.