Kadali Devi Sindhuja, 2025, 13:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Movie Recommendation System

Assistant Professor Akhilesh kumar singh, Naman kumar Maurya, Sejal Verma
Galgotias University

Abstract- Recommendation System used to predict and suggest movies based on user preferences.The technique applied here tries to predict user preferences using an information filtering technique that improves the user experience through timely and pertinent recommendations. In particular, movie suggestions are vital for enhancing interpersonal relationships since they may provide users with entertainment options based on their tastes or the current popularity of films.

Data filtering systems often use these to help people locate content that meet their needs by going through large databases and making recommendations on what to buy or watch. These filtering systems, at times referred to as recommender systems, recommendation engines, or platforms, are designed to predict how a user might rank or favor an item. They are mainly used in the business sector.

The primary purpose of this project is to produce a content- based model for film recommendations that involve cosine similarity and vectorization to provide the consumer with general recommendations regarding the popularity of the films.

Keywords: Movie recommendation, vectorization, cosine similarity, content-based recommendation system, data saturation, and data extraction.

I. INTRODUCTION

In today's digital era, people often feel overwhelmed by the vast amount of information available online. Whether searching for lodging, researching investments, or choosing entertainment, the sheer volume of data can make decision- making difficult. To address this challenge, many companies have implemented recommendation systems that personalize user experiences by suggesting relevant content.

Recommendation systems have been widely researched and successfully applied across various domains, including e- commerce, music, and movies. Platforms like Amazon, MovieLens, and Netflix rely on these algorithms to enhance user engagement and drive business growth. In the entertainment industry, movie recommendation systems play a crucial role in helping users discover content tailored to their preferences.

This paper explores the development of a movie recommendation system, discussing various techniques such as collaborative filtering, content-based filtering, and hybrid approaches. By analyzing their effectiveness, we aim to contribute to advancements in recommendation systems and improve user experience in movie selection.

Netflix	2/3rd of the movies watched are recommended
Google News	recommendations generate 38% moreclick-troughs

Amazon	35% sales from recommendations
ChoiceStream	28% of the people would buy moremusic if they found what they liked

Table1. Companies benefit through recommendation system

In order to consider the preferences of users, recommender systems provide recommendations that are most likely to be accepted by users. Users may choose to provide feedback during their interactions or later sessions, which can be either open or implicit.

To make more personalized recommendations in subsequent interactions, the recommender system tracks user behaviors and feedback in its database.

These systems have become an indispensable part of websites run by major online retailers such as Amazon.com and Snapdeal, as well as services offering online movie rentals like Netflix, due to their potential financial returns.

High-quality, personalized suggestions greatly enhance the user experience. Online recommendation systems are increasingly used in a wide range of applications to provide users with personally relevant information.

Generally, recommender systems can be divided into two main types:

1. Collaborative filtering approach
2. Content-based filtering method

- **Collaborative filtering**

Collaborative filtering suggests items based on the similarity between users and their preferences. The system recommends products that have been favored by people with similar tastes.

Some advantages of collaborative filtering include:

- Content-Independent – It relies on user connections and preferences rather than the content of the items.
- Real Quality Measurement – Since users directly rate the products, this technique provides an accurate measure of quality.
- Serendipitous Recommendations – The system can offer unexpected yet relevant recommendations, as they are based on user similarities rather than item similarities.

- **Content-based filtering**

Content-based filtering relies on user preference profiles and item descriptions. In this approach, items are described using key attributes, and user profiles reflect their preferences. The algorithm recommends products that resemble those a user has favored in the past by analyzing previously rated items.

Various content-based filtering techniques have been proposed in research, and hybrid recommender systems often incorporate these approaches. For example, an early study by Eyjólfsson et al. introduced MOVIEGEN, a movie recommendation system. While it had certain benefits, it required users to answer a set of questions, making the process time-consuming, less personalized, and somewhat stressful.

To overcome these limitations, we propose an improved movie recommendation system that primarily relies on user input. In this system, users can select specific attributes such as actor, director, genre, year, and rating. Based on past interactions, the system predicts and suggests movies that align with their preferences.

II. PROBLEM STATEMENT

The reason behind this project is that we lose our quality time in search of movies, so we try to design a movie recommendation system that helps people in finding movies of their interest.

Objectives

1. To design a movie recommending system based on machine learning.

2. The ultimate objective of the movie recommendation system is to suggest the users their most probable programs and reduce their searching time.

Methodology

Vectorization is the process of converting data such as text or images into numerical vectors in machine learning. Vectorization helps transform user behaviors like viewing history or movie attributes such as genres, plot descriptions, ratings, etc., into a numerical representation that can be studied and compared in the context of a movie recommendation system.

Important Steps in Movie Recommendation Vectorization:

1. **Text Vectorization:** If your data includes text (e.g., movie descriptions, genres, or user reviews), you can vectorize this information to use it in your recommendation algorithm.
2. **2Bag of Words (BOW):** The number of occurrences of a word in a document, such as a movie synopsis, with no regard for the order in which the words appear.
3. **Feature Vectorization:** Movie attributes like genres, ratings, release year, etc., are usually categorical or numerical. You can create vectors to represent these attributes.

- One hot encoding can represent categorical variables like genres.
- Normalization or Standardization may be applied to numerical features like ratings or runtime.

User Behavior Vectorization: User interaction data (e.g., watch history, ratings) can also be vectorized.

- **Collaborative Filtering:** This method creates user and movie vectors based on user preferences (e.g., user A watched movies X, Y, and Z) and finds patterns in user behavior to make recommendations.
- **Matrix Factorization:** Reduces the dimensionality of the user-item matrix (e.g., users vs. movies) and finds latent factors that explain patterns in the data, making the recommendation more efficient.

$$R = U\Sigma V^T$$

Steps to Use Vectorization in a Movie Recommendation System:

Data Collection:

Collect movie metadata (e.g., genres, descriptions, cast) and user interaction data (e.g., watch history, ratings).

Text Vectorization:

If using movie descriptions or reviews, apply TF- IDF or Word2Vec to convert text into numerical vectors.

Feature Vectorization:

- Convert categorical features like genres using
- one-hot encoding or binary encoding.
- Normalize numerical features like movie runtime, release year, and ratings.

User-Movie Matrix:

Create a user-item matrix, which contains rows for users, columns for movies, and each cell filled with the user's rating or interaction with a special movie.

Similarity Calculation:

- Use cosine similarity or other distance measures between movie vectors or between user vectors to recommend similar movies.
- For example, if user A liked movie X, you could recommend movies similar to X by finding movies whose vectors are closest to X's vector.

Recommendation Generation:

- Based on the vectors, compute the similarity scores between users and movies or between movies themselves.
- Recommend the top 5 movies with the highest similarity scores for each user.

III. MODEL ARCHITECTURE

This flow diagram visually summarizes how a movie recommendation system works, from the user's input to delivering movie suggestions, using data and preprocessing.

Downloading and Loading Data Description:

- This step involves acquiring a dataset from Kaggle, which contains movies and relevant attributes like movie_id, title, and possibly a description or other metadata (like genres, keywords, etc.).
- The data is generally stored in formats like CSV, which can be loaded using Python's Pandas library.

Process:

- You search for a movie dataset on Kaggle, download it, and load it into a Pandas Data Frame for processing.
- Data like movie id, title, and tags (i.e., descriptive words that summarize the movie) are extracted or created by combining columns such as genres, overviews, and keywords.

Preprocessing the Data Description:

- Preprocessing involves cleaning and preparing the dataset for analysis. In this case, it mainly involves constructing the "tags" that will represent each movie.
- A "tag" is a combination of descriptive features that can help distinguish one movie from another. This could include genres, overview, keywords, and other text data.

Process:

- Clean the text data by removing unnecessary characters (such as punctuation or symbols).
- Lowercase all text to ensure uniformity during text processing.
- If necessary, concatenate different features (like genres and overviews) into a single text column named "tags." The goal is to have a single textual representation for each movie.

movie_id	title	tags
19995	Avatar	["In, the, 22nd, century., a, paraplegic, Marine...
285	Pirates of the Caribbean: At World's End	["Captain, Barbossa, long, believed, to, be..."]
206647	Spectre	["A, cryptic, message, from, Bond's, past, send...
49026	The Dark Knight Rises	["Following, the, death, of, District, Attorney..."]
49529	John Carter	["John, Carter, is, a, war-weary, former, mili..."]
...	...	...
9367	El Mariachi	["El, Mariachi, just, wants, to, play, his, gui..."]
72766	Newlyweds	["A, newlywed, couple's, honeymoon, is, upend
231617	Signed, Sealed, Delivered	["Signed, Sealed, Delivered", introduces, a..."]
126186	Shanghai Calling	["When, ambitious, New, York, attorney, Sam, i
25975	My Date with Drew	["Ever, since, the, second, grade, when, he, fi..."]

Vectorizing the Tags Description:

- Vectorization is the process of transforming text into a numerical from to be processed and analyzed by a computer.
- This example uses the TF-IDF (Term Frequency- Inverse Document Frequency) approach. When the textual data is transformed into a matrix using TF-IDF, each movie is represented by a vector of integers that indicate how important the words (tags) in that film are in relation to other movies in the dataset.

Process:

• TF (Term Frequency): This measures how often a word (or tag) appears in the tags of a movie. The more a word is used, the more its value.

$$\mathrm{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

• IDF This increases the weights of less frequent an d more unique words found in fewer film, thus im parting importance and reduces the weight of common words that appear in many films.

$$\mathrm{IDF}(t) = \log\left(\frac{N}{\text{Number of documents containing term } t}\right)$$

• The final output is a TF-IDF matrix, where each row represents a movie and each column represents a movie and each column represents a word (Tag).

The entries in the matrix represent the term’s importance for that particular movie.

The final output is a TF-IDF matrix, where each row represents a movie and each column represents a word (Tag). The entries in the matrix represent the term’s importance for the particular movie.

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t)$$

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

Cosine Similarity for Similarity Calculation Description:

- We should compare films to each other after vectorizing the tags. Cosine similarity is helpful in this case. A metric called cosine similarity uses the angle between two vectors to calculate how similar they are.
- In an n-dimensional space, it computes the cosine of the angle between two vectors. Two vectors are said to be identical if their cosine similarity is 1, and entirely distinct if it is 0.

Process:

- Calculate the cosine similarity of the vectors in the TF-IDF matrix's rows..
- We compute for each movie a similarity to all other movies. Each entry in the resulting similarit y matrix is a measure of the similarity between t wo movies.
- When two films have a greater similarity score, it means that shared tags of the movies are closer.

Sorting and Finding Top Recommendations Description:

- Once you have the similarity scores for each movie, the goal is to find the most similar movies to a given one. This involves sorting the similarity scores for each movie and picking the top N movies (usually 5).
- These recommendations are based on the movies that share the highest similarity in terms of their tags.

The TF-IDF vectors of two films are A and BBB. A·BA · BA·B: Dot product of the vectors.
These are the vector magnitudes (norms): ‖A‖ and ‖B‖.

Process:

- For a given movie (e.g., The Matrix), retrieve the corresponding row in the similarity matrix, which contains its similarity scores with all other movies.
- Sort these scores in descending order to rank the movies based on how similar they are to the target movie.
- Return the top N (e.g., 5) movies with the highest similarity scores, excluding the original movie itself.

IV. DISCUSSION

The evidence in this evaluation study underlines the importance of recommendation systems to the motion picture industry.

Diverse tactics, which include collaborative filtering, content-based fully filtering, and hybrid methods, had been examined and in comparison. It goes without saying that combining multiple approaches, such as deep learning algorithms, can improve the efficiency and general performance of movie recommendation systems.

The overview specifically stresses the fact that dynamic components are essential parts of advise structure. Standard approaches lack capturing dynamic and time-evolving aspects of person-item interactions often. Still, the integration of temporal effects into collaborative filtering methods has been rather promising in enhancing the accuracy of advise.

Besides, incorporating presenting elements and societal impact into recommendation structures has produced excellent outcomes. Utilizing social traits and behaviors enhances the first-class of pointers by providing insightful information about users' decisions. Adding reasons to the suggestion process increases users' comprehension and confidence in the device's judgments.

V. CONCLUSION

In conclusion, this project emphasizes the growing importance of recommendation systems, particularly in the realm of movies. By leveraging a content-based filtering approach, combined with vectorization and cosine similarity, the model effectively streamlines the movie selection process, providing users with personalized recommendations based on movie popularity and attributes. The use of vectorization techniques like TF-IDF enables the system to process large datasets efficiently, offering relevant suggestions while minimizing users' search time.

The project has successfully demonstrated that content- based recommendation systems can provide meaningful insights by analyzing textual and categorical data from movies. While collaborative filtering and hybrid methods also offer significant benefits, this content-based approach shows promise in delivering precise recommendations tailored to user preferences. Moreover, integrating dynamic user behavior and contextual data can further enhance the recommendation process, making it more adaptive to changing preferences over time.

Future work could involve extending the model by incorporating user feedback, temporal factors, and social influences, thereby improving the system's accuracy and making the movie recommendation process even more user- friendly.

REFERENCES

1. Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." International Journal of Computer Applications 160.7 (2017).
2. Rajarajeswari, S., et al. "Movie Recommendation System." Emerging Research in Computing, Information, Communication and Applications. Springer, Singapore, 2019. 329-340.
3. Ahmed, Muyeed, Mir Tahsin Imtiaz, and Raiyan Khan. "Movie recommendation system using clustering and pattern recognition network." 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2018.
4. Arora, Gaurav, et al. "Movie recommendation system based on users' similarity." International Journal of Computer Science and Mobile Computing 3.4 (2014): 765-770.
5. Subramaniyaswamy, V., et al. "A personalised movie recommendation system based on collaborative filtering." International Journal of High Performance Computing and Networking 10.1-2 (2017): 54-63.
6. R. Lavanya, U. Singh and V. Tyagi, "A Comprehensive Survey on Movie Recommendation Systems," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 532-536, doi: 10.1109/ICAIS50930.2021.9395759.
7. Nisha Sharma and Mala Dutta, "Movie Recommendation Systems: A Brief Overview", July 2020.
8. Gaurav Srivastav, Sargam Maurya, Tanisha Tripathi, Tushar Narula, "Movie Recommendation System Using Cosine Similarity and KNN", June 2020.
9. Aishwarya P, Anup Jung Shah, and Ashrita Kashyap, "A Movie Recommender System: MOVREC using Machine Learning Techniques", 2020.
10. Kumar, M., Yadav, D. K., Singh, A., & Gupta, V. K. (2015). A movie recommender system: Movrec. International journal of computer applications, 124(3), 7-11
11. R. E. Nakhli, H. Moradi, and M. A. Sadeghi, "Movie Recommender System Based on

Percentage of View," In 2019 5th Conference on Knowledge-Based Engineering and Innovation (KBEI), pp. 656-660, IEEE.

12. G. Wang, "Survey of personalized recommendation system," Computer Engineering & Applications, 2012.
13. Peng, Xiao, Shao Liangshan, and Li Xiuran. "Improved Collaborative Filtering Algorithm in the Research and Application of Personalized Movie.
14. Movie Recommendation System Using Collaborative filtering, 978-1-5386-65657118/2018 IEEE.
15. Sharma, M., & Mann, S. (2013). A survey of recommender systems: approaches and limitations. International journal of innovations in engineering and technology, 2(2), 8-14. [11] Shreya Agrawal, Pooja Jain, —An Improved Approach for Movie Recommendation System, || International conference on I-SMAC (I-SMAC 2017).
16. https://en.wikipedia.org/wiki/Recommende r_system
17. https://towardsdatascience.com/calculate-similarity- the-most-relevant-metrics-in-a-nutshell9a43564f533e
18. https://www.researchgate.net/publication/1 1563559_C omparative_methods_for_the_analysis_of_c ont