RESEARCH ARTICLE OPEN ACCESS

# Flick Counsel System Using Machine Learning Approach

Harshada Choudhar[1],Kanchan Gaikwad[2],Shweta Gaikwad[3],Prachi Jaigude[4]

[1]( UG Students, Computer Engineering, SVPM's College of Engineering Malegaon(bk), Baramati)
Email :harshadachoudhar2001@gmail.com,kanchangaikwad32@gmail.com,gaikwadshweta595@gmail.com, prachi.jaigude2001@gmail.com.

Abstract—In this hustling world, entertainment is a necessity for each one of us to refresh our mood and energy. For watching favourable movies online we can utilize movie recommendation system. Our goal is to make the movie discovery process. Recommendation system will continuously analyze individual user's movie preferences and recommend custom movie recommendations. This is purely a movie recommendation service in that it offers a list of movie suggestions based on previous user ratings. This is designed not to search for movies but to discover them through our recommendation process. Our system will allow users to rate movies they have seen. Then the data is analyzed, and recommended to the user. The core of our project, recommendation system, is based on a collaborative filtering algorithm . We have filtered and tuned the parameters around this algorithm by comparing our predicted ratings against actual ratings using sample and out-of-sample techniques as well as inspected live user feedback..

Keywords: Machine Learning, CollaborativeFiltering, Recommendation System, Cosine Similarity algorithm.

## 1. INTRODUCTION

Movie recommendation systems have become increasingly popular in recent years as they provide users with personalized movie recommendations based on their preferences. With the vast number of movies available, users can feel overwhelmed when choosing what to watch. A recommendation system can help users discover new movies that they may not have found on their own. These systems use various algorithms and techniques to analyze user behavior, such as past movie ratings and viewing history, to generate recommendations that are tailored to each individual user. In this project, we aim to build a movie recommendation system using the collaborative filtering algorithm and integrate it with a Flask application to enable API calls and make it accessible on the web.

The movie recommendation system we are building is based on the collaborative filtering algorithm. Collaborative filtering is a widely used technique in recommendation systems and works by finding similarities between users based on their movie ratings and using these similarities to make recommendations. This approach is effective as it relies on the collective preferences of similar users rather than just the individual user's preferences.

The system will be integrated with a Flask application to allow API calls and make it accessible on the web. Flask is a popular web framework used to create APIs and web applications in Python. Integrating the movie recommendation system with Flask will enable users to access the system through a web interface,making it easier to use and more widely available.

There are many existing movie recommendation systems available online, such as Netflix, IMDb, and Amazon Prime. These systems use various algorithms and techniques to provide personalized recommendations to users. However, many of these systems have limitations, such as requiring a subscription, lack of transparency in their recommendation algorithms, and limited availability of movies. By building our own movie recommendation system and integrating it with a Flask application, we aim to create a more accessible and transparent recommendation system for users to discover new movies based on their interests.

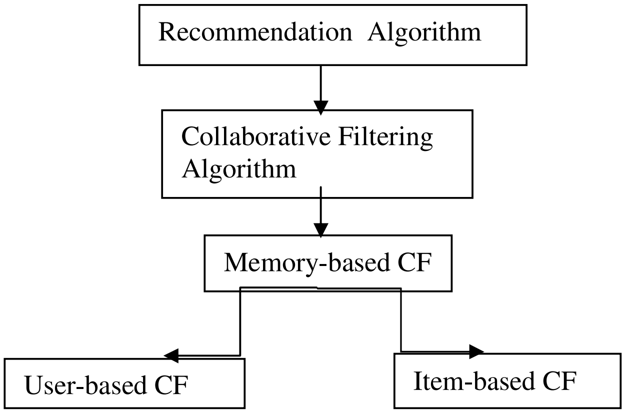
Overall, the movie recommendation system we are building aims to provide accurate recommendations to users while being transparent and accessible. The system uses collaborative filtering algorithm, integrated with a Flask application to enable API calls and make it accessible on the web. The system is designed to be scalable, user-friendly, and able to handle a large number of users and movies.

## 2.OBJECTIVE

In this paper to design a recommendation system that will be able to recommend the movie using ML(Machine Learning Algorithms).

1.To increase user satisfaction
2. Increased user-engagement and loyalty to website.
3. SciPy increases revenue for business.
4. Helps the service provider to deliver their services to the right user.

## 3.RELATED WORK



Existing recommendation algorithm can be divided into three kinds such as content-based, collaborative filtering and hybrid. Among these recommendation algorithms, collaborative filtering is the most popular technique, based on the core assumption that users who have expressed similar interests in the past will share common interests in the future. Collaborative filtering methods can be model-based or memory –based. Model-based algorithm first construct a model to represent user behavior and, therefore, to predict their ratings. The parameters of the model are estimated using the data from the rating matrix. Memory-based approaches can be categorized as user-based or item-based. User-based collaborative filtering algorithms generate recommendations based on the preference of similar users. In contrast to user-based CF, item-based CF approaches recommend items on the basis of information about other items that a user has previously rated.

## 4.LITERATURE SURVEY

Before building our movie recommendation system, we conducted a literature survey to understand the existing research in the field of movie recommendation systems. Our survey revealed that there are many different algorithms and techniques used in movie recommendation systems, with collaborative filtering being one of the most popular.

Collaborative filtering is a widely used technique in movie recommendation systems, as it relies on the collective preferences of similar users to generate recommendations. One of the earliest works in collaborative filtering is the Netflix Prize, which was a competition launched by Netflix in 2006 to improve their movie recommendation algorithm. The winning team used collaborative filtering and matrix factorization techniques to improve the accuracy of the algorithm.

In addition to collaborative filtering, other techniques used in movie recommendation systems include content-based filtering, hybrid filtering, and deep learning-based techniques. Content-based filtering uses features of the movies, such as genre, actors, and directors, to generate recommendations. Hybrid filtering combines multiple techniques, such as collaborative filtering and content-based filtering, to generate more accurate recommendations. Deep learning-based techniques use neural networks to learn complex relationships between users and movies.

Our literature survey also revealed that there are many challenges in building movie recommendation systems, such as the cold-start problem, sparsity problem, and scalability problem. The cold-start problem occurs when a new user or movie enters the system and there is not enough data to generate accurate recommendations. The sparsity problem occurs when users have rated only a small number of movies, making it difficult to find similarities between users. The scalability problem occurs when the system needs to handle a large number of users and movies.

To overcome these challenges, researchers have proposed various solutions, such as incorporating external data sources, using matrix factorization techniques, and improving the user interface to encourage more user interactions. Our literature survey helped us to understand the existing research in the field of movie recommendation systems and informed the design of our system.

## 5.EXISTING SYSTEM

There are many existing movie recommendation systems available online. These systems use various algorithms and techniques to provide personalized recommendations to users.
Along with the expansion of domain comes information overload and difficulty in extraction of data, to overcome this problem the recommendation system plays a vital role. Our system gives fast and coherent suggestions while there is problem in finding the desired content from the ever-increasing millions of contents every year. Our proposed system improve accuracy and performance of a regular filtering technique.

## 6.PROPOSED SYSTEM

### A. Data Collection

There are several datasets available to build a movie recommendation system. But for this project , we are going to use a IMDB dataset that contains columns like title , genre , description , director , actors , year , runtime(Minutes), rating , votes , revenue(millions) , meta score , etc.

### B. Data Preprocessing

Data preprocessing 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.
The dataset contains one CSV file and it contains various columns which can be useful for the further analysis of data.

### C. Data Analysis

We are done selection of analytical techniques such as collaborative filtering algorithm and cosine similarity algorithm.

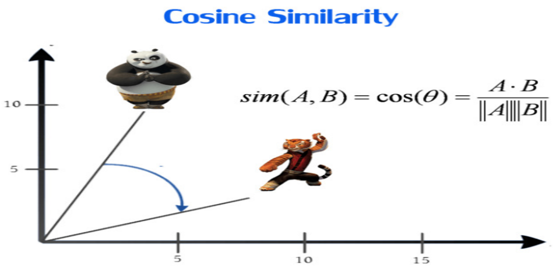
### D. Collaborative Filtering

Collaborative filtering approaches build a model from the user's past behavior (i.e. items purchased or searched by the user) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that users may have an interest in.

1] **User Based**: Here, we look for the users who have rated various items in the same way and then find the rating of the missing item with the help of these users.

2] **Item Based**: Here, we explore the relationship between the pair of items (the user who bought Y, also bought Z).We find the missing rating with the help of the ratings given to the other items by the user.

### E. Cosine Similarity

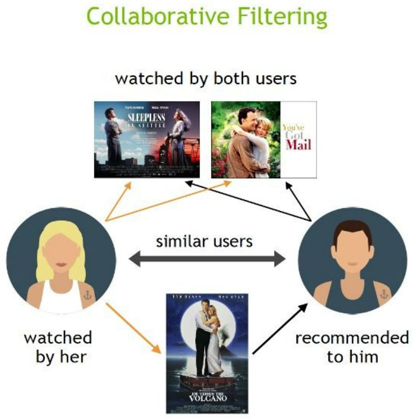


To calculate the distance in between the target movie and the movies in the dataset, cosine similarity is used. It measures the similarity in between two documents irrespective of how different they are in size and calculates the cosine angle between two vectors.

## 7.MODULES

### I. Collaborative Filtering Algorithm:

The collaborative filtering algorithm is the core of the recommendation engine. It works by finding similarities between users based on their movie ratings and using these similarities to make recommendations. There are several variations of collaborative filtering, including user-based and item-based collaborative filtering, which differ in how they calculate similarities between users and items. The algorithm is trained on a dataset of movie ratings to generate personalized movie recommendationseach user.



### I.Flask Application:

The Flask application is a web framework used to create the API for the movie recommendation system. The Flask application runs on a server and listens for API requests from clients. It routes these requests to the appropriate function in the movie recommendation system and returns a response to the client. The Flask application is used to enable API calls, allowing the system to be accessed on the web.

### II. Database:

The database module is used to store user details and improve future recommendations. When a user rates a movie or inputs their movie preferences, this information is stored in the sqlite3 database. The system uses this information to improve privacy for the user. The database also enables the system to handle a large number of users and feedback.

### III. User Interface:

The user interface module is used to create a user-friendly interface for the movie recommendation system. This module takes input from the user, such as genre, actors, and directors, and displays personalized movie recommendations based on the input. The user interface is designed to be easy to use for non-technical users.

Overall, these modules work together to create a movie recommendation system accurate, scalable, and user-friendly. The collaborative filtering algorithm generates personalized movie recommendations for each user, which are accessed through the Flask application API. The database that is module stores user preferences and improves future recommendations, and the user interface module provides a user-friendly interface for users to interact with the system.

## 9.CONCLUSION

In this paper we recommends the movies using IMDB dataset. We have done exploratory data analysis on data also done data prepossessing so that we can train our model on it. Now whenever a user visits our websites and gives his choice for movie name then it gets auto-suggestions related to movie.

Our project works only for Hollywood movies but we can extend it so that it can work for any movie, we just need the required data.

## 10.FUTURE SCOPE

a) Can develop a web application.
b) Can include the collaborative based filtering for top most results.
c) Can work on real time data with supercomputers.

## 11.REFERENCS

[1] Ramin Ebrahim Nikhil ; Hadi Moradi ; Mohammad Amin Sadeghi, Movie Recommender System Based on Percentage of View, 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), 2019.https://ieeexplore.ieee.org/abstract/document/8734976

[2] Ching-Seh (Mike), WU Dept. of Computer Science, San Jose State University, San Jose, CA, USA, 2018.

[3] Roettgers, 2014. "Netflix spends $150 million on content recommendations every year".

[4]Koran, Y., Bell, R., & Kolinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

[5]Sarwar, B., Karpas, G., Konstanz, J., & Riedel, J. (2001). Item-based collaborative filtering recommendation algorithms. In

Proceedings of the 10th international conference on World Wide Web (pp. 285-295).

[6]Dominicus, G., & Pushilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749.

[7]Hu, Y., Koran, Y., & Kolinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In 2008 Eighth IEEE International Conference on Data Mining (pp. 263-272).

[8]Bell, R. M., Koran, Y., & Kolinsky, C. (2010). All together now: A perspective on the Netflix Prize. Chance, 23(1), 24-29.

[9]Lops, P., Gemas, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Recommender Systems Handbook (pp. 73-105). Springer, Boston, MA.

[10]Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet computing, 7(1), 76-80.

[11]Shani, G., & Gunawardena, A. (2011). Evaluating recommendation systems. In Recommender Systems Handbook (pp. 257-297). Springer, Boston, MA.