

# Movie Recommendation System

Dr.R.KavithaM.Santhosh R.Selvachandran

*Professor, Department of IT, Velammal College of Engineering and Technology,  
Madurai*

*UG Students, Department of IT, Velammal College of Engineering and Technology  
Madurai*

Submitted: 01-08-2021

Revised: 10-08-2021

Accepted: 15-08-2021

**ABSTRACT:** Recommendation is an ideology that works as choice-based system for the end users. Users are recommended with their favorite movies based on history of other watched movies or based on the category of the movies. These types of recommendations are becoming popular because of their ability to think and react as human brain. For this purpose, deep learning or artificial intelligence comes into picture. It is the ability to think as a human brain as give the output best suited to the

## I. INTRODUCTION

Recommendation systems are highly popular in the field of entertainment like movies, songs, clothes, accessories, food, restaurants etc. The increasing popularity is due to the accuracy in prediction. Even before a person can imagine about the next similar movie or song, it is recommended under the section of “you may also like”. The prediction of movies or songs of user interest has boosted the gain of companies and profit of such predicting systems. Predictions are generally done on the basis of human interests, behavior, personality etc. For any recommendation system to work properly it should know its audience. Some of such highly popular recommendation websites are Netflix, YouTube, Amazon, Flip kart, Hulu etc. These systems have taken user understanding and prediction to a great level. Recommendations are generally classified in two types – Collaborative filtering and Content based filtering.

## II. CONTENT-BASED FILTERING

In content-based filtering, liking of only one particular user is taken into consideration. The formerly liked items by the user are collected and based on the similarity between these items new items are recommended. Some websites also using hybrid filtering, which combines the collaborative filtering and the content-based filtering. Netflix and Amazon Prime is an example of such a hybrid system. In this paper, content-based filtering is applied using Python and tensorflow as backend. The recommendations done in this paper are made

end users liking. This paper focuses on implementing the recommendation system of movies using deep learning with neural network model using the activation function of SoftMax to give an experience to users as friendly recommendation. Moreover, this paper focuses on different scenarios of recommendation like the recommendation based on history, genre of the movie etc.

**Keywords :** Movie Recommendation system using deep learning that considers a favorite movie of the user and recommends similar ten movies based on keywords, cast, director, genre and popularity of the movie. It also recommends the top ten movies based on the genre (or category) of the movie which uses popularity as the major aspect to recommend the movies.

The details of the movies (title, genre, runtime, rating, poster, etc) are fetched using an API by TMDB, <https://www.themoviedb.org/documentation/api>, and using the IMDB id of the movie in the API, I did web scraping to get the reviews given by the user in the IMDB site using beautifulsoup4 and performed sentiment analysis on those reviews.

## III. PROBLEM DEFINITION

There are lot of alternatives available on internet when user wants to search for something of his interest. But to choose from such a wide area is very puzzling and frustrating as it is not probable that the user will acquire the information which he wants. Now-a-days there is a need of recommendation system that can think as a human brain. Just like Netflix where even the user is not aware of his favorite movies or series, Netflix suggests the user with what should be watched next. Netflix uses the hybrid method (content-based + collaborative filtering).

This project recommends the user with movies of his choice using the deep learning neural network model and content based filtering. This system specifically focuses on the recommendation of movies. The user will enter his favorite movie and

based on the similarity between that movie and other movies, user will be suggested top ten movies. Similarity is checked on the basis of parameters such as popularity, genre, cast, director, keywords. It uses DNN model which is trained on the basis of the mentioned parameters and one hot encoding is performed on those parameters. The concept of model training reduces the time required for ediction and ultimately provides parallel processing.

#### IV. THE CINEMA MOVIE

We have developed a similar application called "The Movie Cinema" which supports all language movies. But the only thing that differs from this application is that I've used the TMDB's recommendation engine in "The Movie Cinema". The recommendation part developed by me in this application doesn't support for multi-language movies as it consumes 200% of RAM (even after

deploying it to Heroku) for generating Count Vectorizer matrix for all the 700,000+ movies in the TMDB.

#### V. HOW COSINE SIMILARITY WORKS?

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

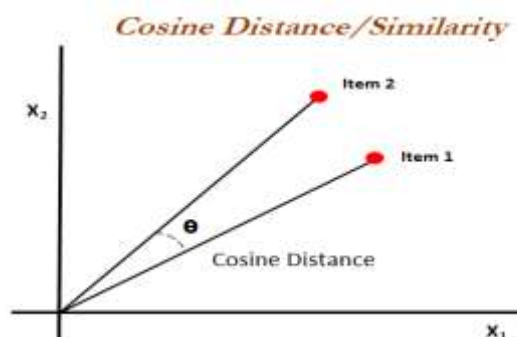


Fig.1 Cosine Distance/Similarity Diagram

#### VI. IMPLEMENTATION

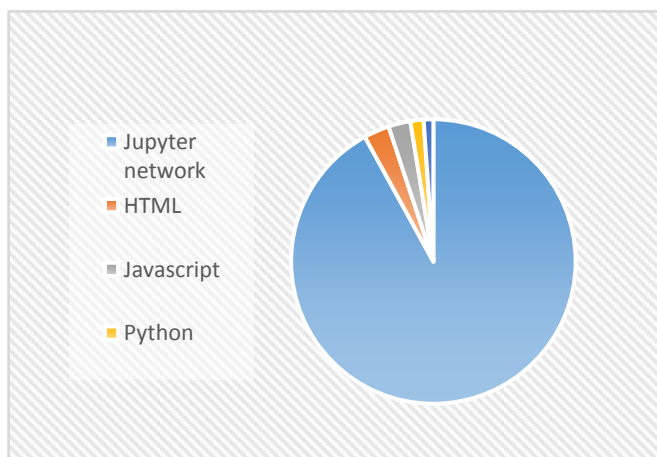
At first the dataset of almost 40,000 Hollywood movies consisting of release date, popularity, cast, director, genre, title and many more columns is processed to remove all the ambiguities and impurities present in the data. Data processing occurs in order for the data to be transmitted should be clean and error-free. After the data processing and data cleaning is done, one more column known as "combined features" is added to the existing dataset. In this column all the data for four columns, namely, genre, title, cast, director is combined. Later one hot encoding is performed on this column and a "bag of words" is formed which contains either 1 or 0. 1 resembling the presence while 0 resembling the absence. A DNN model is created with output activation function as softmax and the bag of words and one result in predicted by the model. The model is trained with 150 epochs and 7342 steps.

After the processing is completed on the dataset, input from the user is accepted. User is prompted to enter his favorite movie. When the user enters the title of his favorite movie corresponding row from the combined features column is searched and again one hot encoding is performed where categorical variables are assigned binary representation. This calls the DNN trained model and a list of result is given as output by the model which specifies the probabilities of similarities with all the movies. By arranging the top 10 similar movies the title of those top 10 movies is generated as the output and shown to the user.

The movies are not only generated based on the combined features after user enters the title of the movie, but the movies are also generated based on the popularity aspect of the movie. When an user enters only the genre of the movie (say comedy) then based on the popularity top 10 movies..

## VII. LANGUAGE USED

Fig.2 Pie Chart of Languages Used



## VIII. ARCHITECTURE

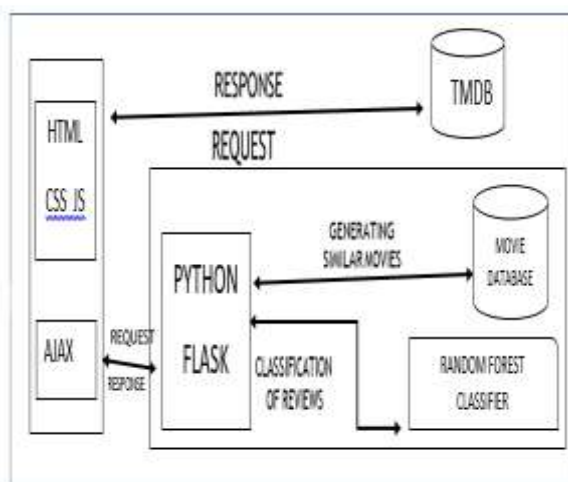


Fig.3 Architecture Diagram

## IX. USER INTERFACE

User interface is the middleware through which users communicate with the system. For the proposed framework user interface is created through a Flask framework of Python. It gives the ability to associate HTML, CSS and JavaScript in the Python environment. A flask website can be personalized as per users need and requirements. For this project, a simple UI has been maintained with a text box hinting the user to enter his favorite movie.

While entering the favorite movie user is also provided with some auto suggestions emerging from the database using JavaScript. This reduces the efforts and time of the user to enter the entire

movie name also it becomes easy for the backend code to directly fetch the movie title. Recommendation based on genre of the movie gives the user a broader view of movies if he is not sure about any particular movie. These all things are provided in the user interface which is created using the Flask framework.

## X. RESULTS AND ANALYSIS DATASET

The dataset used for this project consists of 500 Hollywood movie records consisting of the attributes such as ID, Title, Release\_Date, Director, Cast, Popularity, Keywords, Genre and many more

attributes. The attributes used for this project are popularity, genre, title, director and keywords.

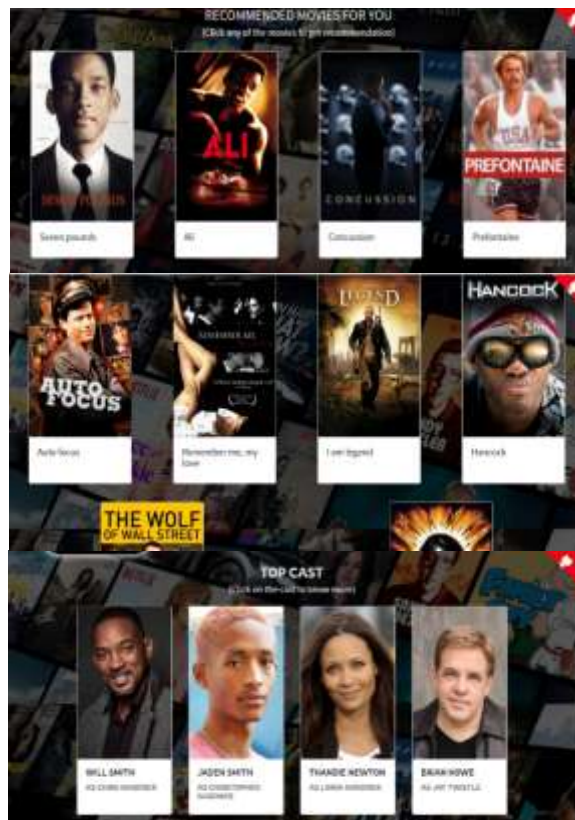
10.1 ACCURACY BASED ON USER HISTORY: For recommending the movies based on history of movies by user is based on the DNN model training. The accuracy at the end when the model is trained is 91% with a loss of 9%. For model training attributes such as keywords, title, genre, cast and director are used.

10.2 ACCURACY BASED ON GENRE : The user enters the type of the movie that are to be suggested. The type of the movies can be of any kind, such as, Comedy, Action, Sci-Fi, Adventure etc. This recommendation is done on the basis of popularity attribute from the dataset. The accuracy for this is of 94% with a loss of 6% and a Mean Square Error of 0.034.

Table 1. Table shows the summarized content of the accuracy, loss and MSE

	Based On	% Success	% Loss	Mean Square Error
	History of movies	91%	9%	0.018
	Genre	94%	6%	0.0034

### XI. SCREENSHOTS



## XII. CONCLUSION

The most important thing that any user wishes, is to have the information that he truly desires, to be available at any given time. If the user has to search for some information by devoting his valuable time, it becomes tedious. To avoid this recommendation systems were built. This recommendation system completely focuses on recommending best top 10 movies to the user based on his choice thus saving his time.

## REFERENCES

- [1]. Shreya Agarwal, CSE ME, Indore, India. An Improved Approach for Movie Recommendation System. 5 ITM Web of Conferences 32, 03008 (2020) <https://doi.org/10.1051/itmconf/20203203008> ICACC-2020
- [2]. Ching-Seh (Mike) Wu Dept. of Computer Science San Jose State University, Deepti Garg Dept. of Computer Science San Jose State University, Unnathi Bhandary Dept. of Computer Science San Jose State University.
- [3]. Tianqi Zhou, Lina Chen, Jian Shen. Movie Recommendation System Employing the User-based CF in Cloud Computing
- [4]. Jeffrey Lund Computer Science Department Brigham Young University Provo, Utah 84602, USA jefbund@byu.edu Yiu-Kai Ng Computer Science Department Brigham Young University Provo, Utah 84602, USA ng@compsci.byu.edu Movie Recommendations Using the Deep Learning Approach
- [5]. Libo Zhang 1, Tiejian Luo2, Fei Zhang2, And Yanjun Wu1 1 Institute of Software, Chinese Academy of Sciences, Beijing 100190, China. 2. University of Chinese Academy of Sciences, Beijing 101408, China. A Recommendation Model Based on Deep Neural Network
- [6]. KEMA: Master's Thesis at Vionlabs Supervisor : Chang Gao Examiner: Mihhail Matskin. Content-based Recommender System for Movie Website