

# MoView Engine : An Open Source Movie Recommender

Vallari Manavi<sup>1</sup>, Anjali Diwate<sup>2</sup>, Priyanka Korade<sup>3</sup>, and Mrs. Anita Senathi<sup>1\*</sup>

<sup>1</sup> Ramrao Adik Institute of Technology Nerul, Information Technology Department, India

<sup>2</sup> Ramrao Adik Institute of Technology Nerul, Information Technology Department, India

<sup>3</sup> Ramrao Adik Institute of Technology Nerul, Information Technology Department, India

<sup>1\*</sup> Ramrao Adik Institute of Technology Nerul, Information Technology Department, India

**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 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.

## 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. In collaborative filtering user with similar interests are found out from a large group of users. Then the user is recommended with items based on the items liked or used by similar users. 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 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.

## II. 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) with deep learning. 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 prediction and ultimately provides parallel processing.

## III. RELATED WORK

[1] An improved approach for Movie Recommendation System includes the hybrid version of traditional methods. The system includes both Content based filtering and Collaborative filtering method. Hybrid method is specially used for improving quality of the recommendation system. Improved version of recommendation uses Genetic Methodology for implementation and Support Vector Machine as a classifier. Hybrid approach helps to get advantages from both the methods as well as removes the drawbacks of both methods. Due to combination of the two methodologies, the proposed approach shows an

improvement in accuracy, scalability and quality of recommendation.

[2] Recommendation using Collaborative Filtering is the main approach of the above system. Collaborative filtering systems analyses the user's behavior and preferences and predict what they would like based on similarity with other users. The system is implementation of Collaborative Filtering algorithm using Apache Mahout. Apache Mahout is Big Data component which provides machine learning libraries. Matplotlib is one of the Python libraries used for machine learning. The entire system is implemented using the combination of Collaborative filtering and Apache Mahout.

[3] Movie Recommendation system is based on the Collaborative Filtering method. The paper focuses on how to design reliable and highly accurate algorithm for movie recommendation. Java Language on a Ubuntu System is specially used for recommendation. The system also uses MapReduce Framework of Big Data, which can easily handle large data sets. By using MapReduce framework, the result shows that the system can achieve high efficiency and reliability in case of large data sets.

[4] In this paper, they have proposed a deep learning approach based on auto encoders to produce a collaborative filtering system which predicts movie ratings for a user based on a large database of ratings from other users. Using the Movie Lens dataset, they have predicted users' ratings on new movies, thereby enabling movie recommendations using a neural network model which performs well in terms of root mean squared error for collaborative filtering. Further the system uses regularization to reduce recommendation errors.

[5] A model combining a collaborative filtering recommendation algorithm using deep learning technology is proposed in two parts. First, the model uses a feature representation method based on a quadric polynomial regression model, which obtains the latent features more accurately by improving upon the traditional matrix factorization algorithm. Then, these latent features are regarded as the input data of the deep neural network model, which is the second part of the proposed model and is used to predict the rating scores. Finally, by comparing with other recommendation algorithms on three public datasets, it is verified that the recommendation. Performance can be effectively improved by this model.

[6] This paper introduces content-based recommender system for the movie website of VionLabs. There are a lot of features extracted from the movie by analyzing the text information, they are diversity and unique, which is also the difference from other recommender systems. They used these features to construct movie model and calculate similarity to recommend movies

and introduced a new approach for setting weight of features, which improves the representative of movies

## VI. PROPOSED WORK

### A. Overview of our Approach

The proposed framework is an online website. The system initially contains a movies dataset consisting of 40,000 records. When user enters a movie of his choice one hot encoding is performed on the parameters of the movie such as *cast*, *genre*, *director*, *keywords*. The given movie by the user is parsed through the neural network model and the model is trained with all the words containing the parameters of the movies.

Based on similarity between the parameters of the movies, top 10 movies are recommended to the user. Probability of similarity between the movies is calculated using the softmax activation function. Similarity matrix is calculated in descending order to generate a list of top 10 similar movies. User can also enter the category of the movie he is interested in, for example, comedy movies. Then based on the *popularity* of the comedy movies the system produces top ten movies as a result. You can view the movie details by clicking on the movie name you wish to watch from the recommended list.

The system uses tensorflow as backend and flask framework to generate a user-friendly website. Tensorflow is used in order to support neural network. Along with tensorflow library, NLTK library is also used for natural language processing of the movie parameters. Natural language processing is required in order to find out the similarity percentage between all the movies. Tokenization of these parameters is done using NLTK to perform one hot encoding. Based on the result of one hot encoding a bag of words is made which contains the result which specifies whether the word is present (1) or absent (0). This bag of words is then given as input to the neural network model to train itself. The output of the model is the similarity probability.

The figure 1 shows a workflow model of the proposed recommendation system. The figure 2 shows how the deep learning model works internally with input layers, hidden layers and output layer.

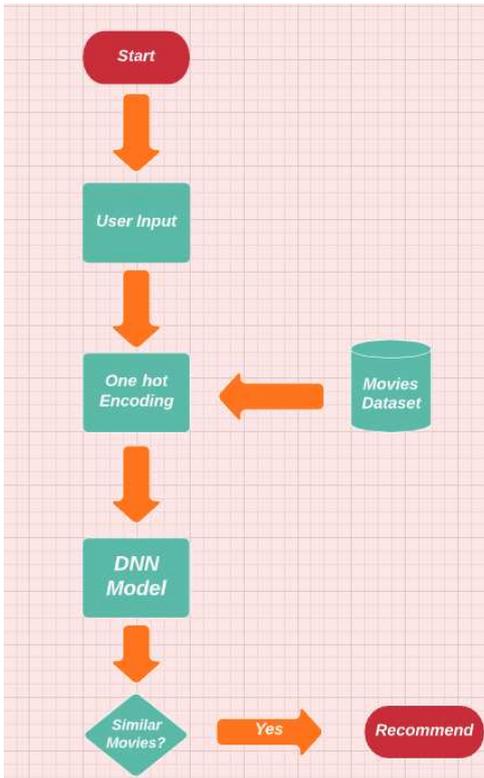
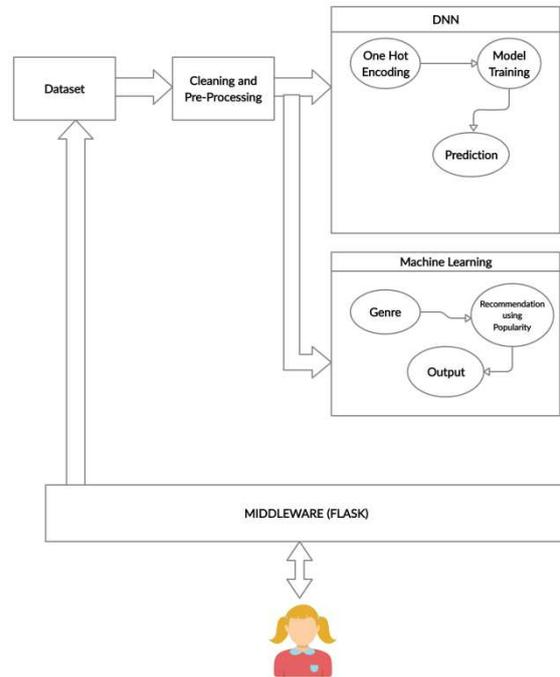


Fig 1. Workflow Chart

B. System Architecture



C. 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

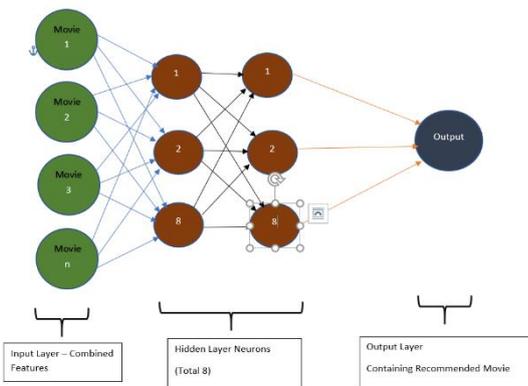


Fig 2. Neural Network Model

movie (say comedy) then based on the popularity top 10 movies are generated.

For the implementation of system these basic requirements are needed to be satisfied.

1. Software Requirements
  - a. Python 3.7.2 – Used for backend as high level language.
  - b. An IDE that supports Python. (For this project PyCharm is used).
  - c. Tensorflow – Package in Python used for deep learning.
  - d. Flask – A Python framework used for hosting the website.
2. Hardware Requirements
  - a. System with minimum 6 GB RAM.
  - b. I5 processor or above.
  - c. System type – 32 bit operating system and above.
  - d. Hard disk with minimum of 250 GB.

#### D. Deep Learning Algorithm

The deep learning model used for this system is the basic DNN algorithm. Deep learning is all about the use of Neural Networks. Neural networks is based on the neuron system of the human brain. It analyses the input deeply and finally provides an output.

There are many activation functions and types of neural network models. For this system the basics are used. A simple DNN model with 8 nodes in the hidden layer and an activation function of SoftMax. This SoftMax activation function is used only while calculating the output or we can say the result matrix. At first the DNN model is trained using the existing dataset to generate the result. Then the input of the user is applied over the trained DNN model to generate a result matrix. This result matrix is a list in python which contains the percentage of similarity between the entered movie and the movies present in the database. This percentage of similarity is calculated by the SoftMax function.

The figure shows the general working of the DNN model and the SoftMax activation func

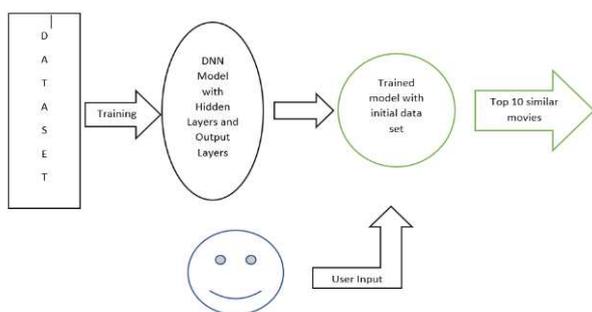


Fig 3. Implementation Analysis

#### E. 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.

### V. RESULTS AND ANALYSIS DATASET

The dataset used for this project consists of 40,000 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.

#### 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.

#### 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



Fig 4. Flask UI

The figure shows the basic user interface created using the Flask framework which acts as a combiner for Machine Learning and HTML. Flask is easily manageable and can be developed according to the user needs. Flask is a direct framework provided by Python.

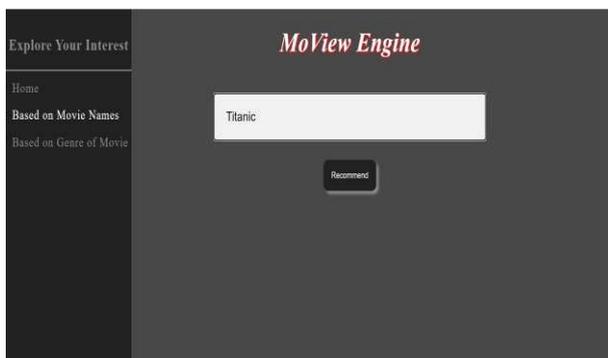


Fig 5. Input by user-1 (based on movie name)



Fig 6. Top ten recommended movies

The figure specifies the input given by the user. Input -1 shows the movie name entered by the user. This name is given as input to the DNN model for fetching the closely related movies based on Neural Networks. The corresponding attributes of the entered movie are fetched and then passed to the trained DNN model. The next figure shows the top ten recommended movies similar to the input.

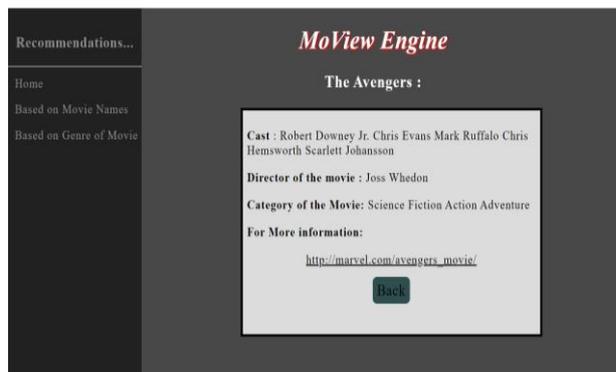


Fig 7. Details of the movie

Above figure shows the details of the movie such as cast, director of the movie, category of the movie and URL for more information selected by the user from the recommended list.

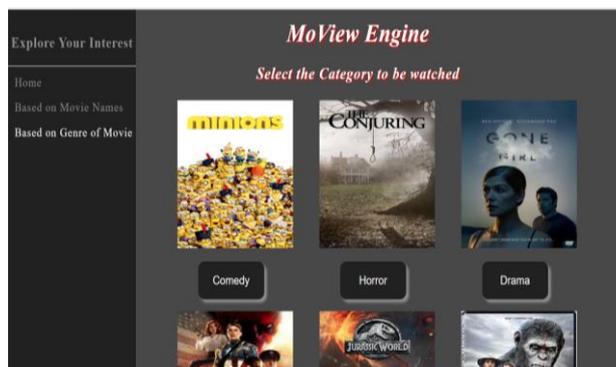


Fig 8. Genre of movie

The figure shows the genre of the movie (in this case comedy). All the comedy movies based on their popularity are suggested to the end user. User can enter any genre of the movie and then the output is generated.

## VI. 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.

## VII. REFERENCES

- [1] Shreya Agarwal, CSE ME, Indore, India. An Improved Approach for Movie Recommendation System.

[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.

Movie Recommendation System Using Collaborative Filtering

[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 [jeffbund@byu.edu](mailto:jeffbund@byu.edu)

Yiu-Kai Ng Computer Science Department Brigham Young University Provo, Utah 84602, USA [ng@compsci.byu.edu](mailto:ng@compsci.byu.edu)

Movie Recommendations Using the Deep Learning Approach

[5] Libo Zhang<sup>1</sup>, Tiejian Luo<sup>2</sup>, Fei Zhang<sup>2</sup>,  
And Yanjun Wu<sup>1</sup>

<sup>1</sup> Institute of Software, Chinese Academy of Sciences, Beijing 100190, China.

<sup>2</sup> University of Chinese Academy of Sciences, Beijing 101408, China. A Recommendation Model Based on Deep Neural Network

[6] KE MA: Master's Thesis at VionLabs Supervisor: Chang Gao Examiner: Mihhail Matskin. Content-based Recommender System for Movie Website