ISSN No:-2456-2165

Movies Recommendation System using Cosine Similarity

Shubham Pawar, Pritesh Patne, Priya Ratanghayra, Simran Dadhich, Shree Jaswal
IT dept., SFIT, Mumbai, India

Abstract:- A Recommendation System is a filtering program whose primary goal is to predict the "rating" or "preference" of a user towards a domain-specific item. In our case, this domain-specific item is a movie. Hence the main focus of our recommendation system is to provide a total of ten movie recommendations to users who searched for a movie that they like. These results are based on similar traits/demographics of the movie that has been searched. Content based filtering is a technique that is used to recommend movies. Apart from providing recommendations the system also provides information about the searched movie. The additional details include the movie rating, its release date, cast, and genres. The system also provides additional information about the cast. To help the user save time on reading reviews the system also performs sentiment analysis on the movie's reviews, grading them into two categories which are 'Good' and 'Bad'.

I. INTRODUCTION

A recommendation system is a type of suggesting system which makes suggestions based on the user's liking. These systems can be applied to various data. These systems can retrieve and filter data based on users preferences to give suggestions or recommendations in the upcoming period.

To watch a movie the first step is to select a movie that matches the user's liking. Users often waste a lot of time selecting a movie to watch. Here comes the need for a recommendation system. It can recommend popular movies based on their rating, but what makes the system useful is its ability to recommend movies based on users' liking and preferences. The purpose of this system is to search for content that would be interesting to an individual.

Since the number of users and the movies are increasing day by day, computing the recommended movies list in asingle node machine takes a very large time. When we deal with huge volumes of data coming from various sources and in a variety of formats as we see in the case of movies where there is a huge amount of data to be computed and then recommended to a user, it involves many aspects that have to be taken into consideration while recommending movies to the user.

Our recommending system uses cosine similarity which is a type of content-based filtering method to recommend similar movies to the user. Additional information about the searched movie will also be provided. The additional information includes a Movie Poster, an Overview of the movie, a Rating of the movie, Genres, the Run time of the movie, and its status which can either be released or unreleased.

This system will also provide the user with sentiment analysis on the reviews of the movie.

These functions of this system will prove to be very useful to the user and consequently save a lot of time, which the user can invest in actually watching the movie he/she likes.

II. LITERATURE REVIEW

By using graph databases, we can construct a data model, it is simpler and more expressive to organize data than to use it. No SQL database or traditional relational database. Ningning Yi can model and manage data applications in a simple and intuitive manner, and it can also make data units smaller and more standard[1]. It can also realize rich relational links.

Ashrita Kashyap1 et al. introduced Movie REC, a recommender system for movie recommendation, which used Blender and CAD tools[2].

Meenu Gupta et al. used KNN algorithms and collaborative filtering in order to increase the accuracy of results as compared to content-based filtering[3]. A collaborative filtering technique combines cosine similarity with the knearestneighbor approach, which alleviates many of the drawbacks associated with content-based filtering. However, it cannot handle fresh items since it hasn't seen them during training.

Rahul Katarya et al. [4] use a hybrid cluster and optimization approach to improve movie prediction accuracy. Such a hybrid approach has been used to overcome the limitations of typical content-based and collaborative recommendation systems. For clustering, k-means algorithm is applied and for optimization, cuckoo search optimization is implemented.

The Android application developed by Nimish Kapoor et al. displays multiple movie categories [5]. Users can add ratings, reviews, create a favorite list of movies, and watch movie trailers. The application's main purpose is to rate movies based on the SVM model used to categorize the ratings into positive and negative emotions.

Bagher Rahimpour Cami et al. propose a content-based movie recommendation system that predicts movie preferences based on temporal user preferences[6]. In the proposed method, the content attributes of rated movies (for each user) are incorporated into a Dirichlet Process Mixture Model to infer user preferences and provide a proper recommendation list.

ISSN No:-2456-2165

Mostafa Khalaji et al. designed a system that combines collaborative filtering and content-based filtering to solve the cold-start problem for new items[7]. HMRS-RA would reduce the cold start problem for new movies by considering contextual information such as genre. Utilizing clustering to reduce the dimensionality of the data, the proposed method solves the scalability problem.

Our lives are greatly improved by movie recommendation systems because they reduce the amount of time and effort required to determine the value of the film. Nayan Verma et al. have used methods like swarm-based collaborative filtering, KNN with S-BERT, and universal sentence encoder[8]. This paper also includes how you can handle challenges to systems. The results of the experiment indicate that the system is effective at predicting highquality films.

Hrisav Bhowmick et al. [9] have implemented eight different methods for recommending movies. An example of a genre-based recommendation technique was that movies associated with a particular genre were checked first, then based on the scores, recommended. In genre based recommendation, however, there remains a high chance that the recommended movies may not be liked by the target user since the recommendation is based on only genre, not user profile similarity. Using the Pearson Correlation Coefficient Based recommended system the similarity between users can be easily determined, but it is a long formula-based method that requires a lot of computation and memory.

Data Collection is a technique that Parth Kotak et al. developed for filtering a data base of movies[10]. It collects ratings from the user and then pre-processes it. The next step is to clean the data, then train the machine learning model, and finally generate predictions. The user enters a movie name and the year in the search bar, and the program recommends four movies based on the likability and user ratings of each movie in that particular year. With a better data set, the model becomes more accurate.

III. METHODOLOGY

The project aims to build a platform that will recommend movies to users, provide a detailed description of the searched movies and perform sentiment analysis on the movie reviews. The information provided will surely cut down the time spent in selecting a movie to watch.

The main purpose to develop a movies recommendation system is to provide users with recommendations that are not based on popularity or purely rating but based on the movies that the user likes. This will lead to a highlypersonalized recommendation, which will increase the accuracy of the recommendation system. The sentiment analysis and the additional information of the searched movie will help the user make an informed decision while selecting a movie. The user won't have to surf the internet for finding a movie that he/she likes as all the information needed will be provided on a single platform. The user won't have to rely on friends for a movie suggestion as the recommendation system will provide the user with the top ten movies that are most similar to the searched movie.

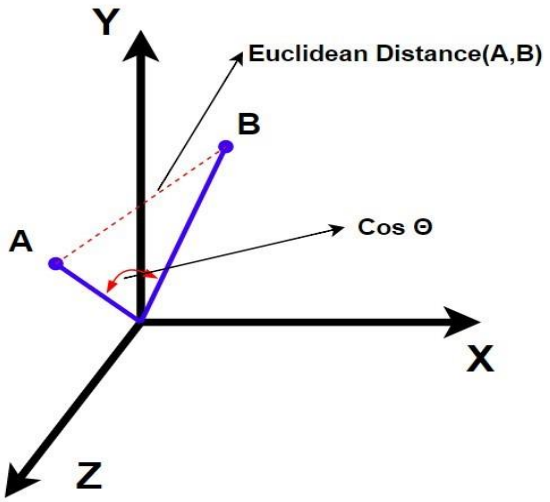

Fig. 1: Cosine Similarity

The movies are recommended based on a simple algorithm called Cosine Similarity. Cosine similarity is a measure used to determine the similarity between two items[14]. Mathematically it can be determined as the cosine angle between two vectors in a three-dimensional plane. We can also check the Euclidean distance between the two vectors to determine how different or similar they are from each other. In our case, one of the vectors is the movie that is searched and the rest of the movies in the database are checked as the second vector. The top ten movies which have the least Euclidean distance corresponding to the searched movie are shown as recommendations.

Cosine Similarity is a type of Content-based filtering approach. It is one of the most popular techniques used in recommendation systems. The attributes of a thing are termed as "content". Based on these attributes we are able to classify whether the two things are similar or not. The attributes can be words specified in the database such as genre, cast names, director names, description, and so on. If the attributes match or have a high similarity then the two movies can be classified as similar movies. The intuition behind this sort of recommendation system is that if a user liked a particular movie or show, he/she might like a movie or a show similar to it.

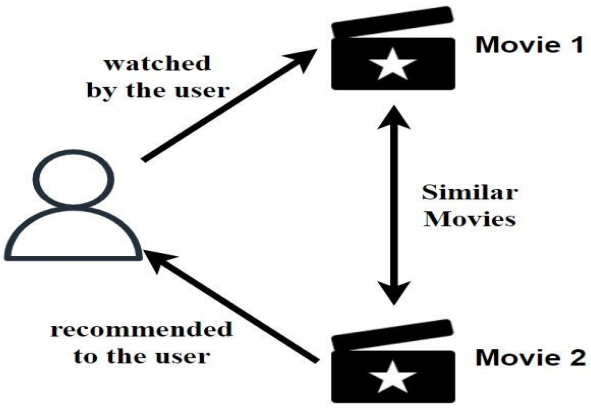

Fig. 2. Content Based Filtering

ISSN No:-2456-2165

IV. WORKING

The proposed solution mainly uses python to work on data sets and to apply various Machine Learning algorithms to get the desired output. AJAX, HTML, and JSON are also used to create dynamic web pages and easy-to-understand GUI for a better user experience.

The Implementation mainly consists of six steps. They are as follows:

- **Step 1:Finding and loading suitable data**
  Appropriate data sets are shortlisted and downloaded from Kaggle[12]. To keep the data up to date, the data for the previous three years is fetched from Wikipedia. The Reviews of various movies are fetched for IMDb (Internet Movie Database) to perform Sentiment Analysis[13]. Additionally TMDB (The Movie Database) API is used to fetch other data and images cast and movie posters[14].
- **Step 2:Data Cleaning**
  The data sets were taken from Kaggle and the data fetched from Wikipedia, both are processed in Jupyter Notebooks to clean the data. The cleaned data is then loaded into the main CSV file which will be used whenever the data needs to be accessed.
- **Step 3:Creating an API key**
  API stands for "Application Programming Interface." An API is a software intermediary that allows two applications to talk to each other.
  In other words, an API is a messenger that delivers your request to the provider that you are requesting it from and then delivers the response back to you. In our case the provider is TMDB. TMDB has a huge collection of movies data, from which the system can fetch the information that it needs. To use TMDB API, an API key has to be generated after creating an account on TMDB.
- **Step 4:Performing Sentiment Analysis**
  NLTK (Natural Language Toolkit) library is imported in python to perform various functions on the reviews data. NLTK corpus is imported to go through all kinds of Natural Language data sets. The Tfidf vectorizer will tokenize the data, learn the vocabulary and inverse document frequency weightings. Multinomial Naive Bayes algorithm is used for classification and analysis of the data which is then divided into training and testing data in the ratio of 4:1.
- **Step 5: Creating files for Web pages**
  For the system to be useful and easy to use, the GUI mustbe good. This would help the user to communicate with the software. Two main HTML pages are created, one for when the user searches a movie and then another for when the user gets all the details of the movie, gets a sentiment analysis of the movie reviews, and gets recommended the top ten movies. AJAX is used to get data from the server, and JSON is a data format that is used to send data to the server.
- **Step6: Applying ML Algorithms for Recommendation**
  Count Vectorizer function is applied on the main data set to form a count matrix. The Count Vectorizer function is used to transform the data into a vector based on the frequency(count) of each word that occurs in the data set. Then Cosine Similarity is performed on those vectors to find the Euclidean Distance to recommend the top ten movies which are most similar to the searched movie.

V. RESULTS

The Movies recommendation system that is created is very user-friendly and easy to use. When the website is loaded, the user can view a screen in which he can enter a movie name to get its detail and recommendations. When the movie name is typed, a suggest drop-down list appears to perform auto-complete. When a movie is selected, the enter button gets enabled.

After clicking on the enter button a new page is loaded, which is made up of four sections. Movie Details, Top Cast, Sentiment Analysis on movie reviews, and the top ten recommendations.

After training and testing the data in the ratio of 4:1, Multinomial Naive Bayes algorithm is used to perform Sentiment Analysis. The accuracy of Sentiment Analysis results in 98.77167 percent.

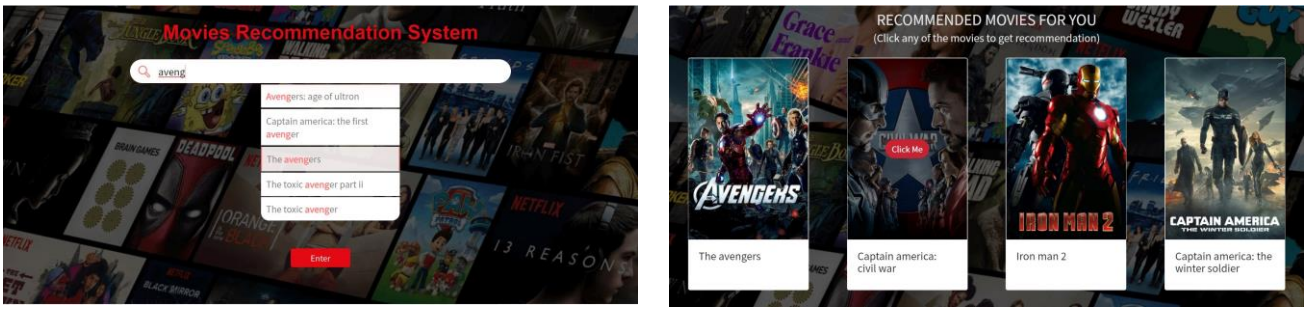

Fig. 3: Home Page

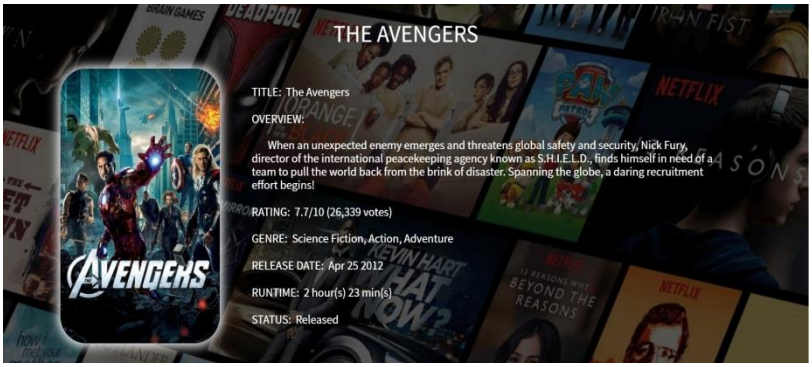

Fig. 4: Movie Details

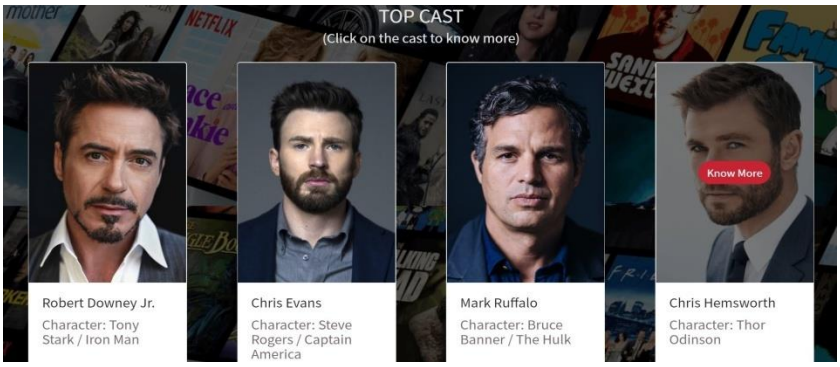

Fig. 5: Top Cast

Fig. 6: Recommendations

VI. CONCLUSION

When the user searches for a movie that he/she has already watched the Movies Recommendation System will recommend the top ten movies that are most similar to the searched movie. Moreover, the system will show additional details of the movie and provide sentiment analysis on the reviews of that movie. All these features will save user's time which otherwise would have been wasted on finding a movie that he/she may or may not like.

Every month several movies are released, the movies database only gets bigger and bigger. This would help the system to provide a more accurate recommendation to the user and in turn increase customer satisfaction.

REFERENCES

[1.] Implementation of Movie Recommender System Based on Graph Database [2017] ;Ningning Yi ; School of Computer Science Communication University of China Beijing, China
[2.] Movie Recommender System: MOVREC using Machine Learning Techniques (2020) Ashrita Kashyap1 , Sunita. B2 ,Sneh Srivastava3 , Aishwarya. PH4 , Anup Jung Shah5 Department of Computer Science Engineering SAIT, Bengaluru, Karnataka, India.
[3.] Movie Recommender System Using Collaborative Filtering; Meenu Gupta; Aditya Thakkar ; Aashish ; Vishal Gupta ; Dhruv Pratap Singh Rathore Department of Computer Science Engineering Chandigarh University, Punjab (2020).

[4.] An effective collaborative movie recommender system with cuckoo search[2017] ; Rahul Katarya ; Om Prakash Verma ; Department of Computer Science Engineering, Delhi Technological University, Delhi, India.
[5.] Movie Recommendation System Using NLP Tools [2020] Nimish Kapoor; Saurav Vishal; Krishnaveni K S; Department of Computer Science and Engineering, Amrita School of Engineering, Amritapuri, Amrita Vishwa Vidyapeetham, Amrita University, India.
[6.] A Content-based based on Temporal Movie User Recommender Preferences System [2017] ; Bagher Rahimpour Cami ;Hamid
[7.] Hassanpour ;HodaMashayekhi Faculty of Computer Engineering IT Shahrood University of Technology Shahrood, Iran
[8.] Hybrid Movie Recommder System based on Resource Allocation [2020] ; Mostafa Khalaji ; Chitra Dadkhah ; JoobinGharibshah
[9.] ; Faculty of Computer Engineering , K.N. Toosi University of Technology , Tehran ,Iran
[10.] Movie Recommeder System using critic consensus [2020] ; A Nayan Verma ; KedareshPetluri ; Department of CSE , PES University , Banglore, India
[11.] Comprehensive Movie Recommdation System [2020]; HrisavBhowmick ; Ananda Chatterjee ; Jaydip Sen ; Dept. Of Data Science , Praxis Business School , Kolkata , India
[12.] Movies Recommendation System using Filtering Approach [2021] ;Parthkotak ; Prem Kotak ; Department of Computer Engineering
[13.] , Vidyalankar Institute of Technology , Mumbai , India
[14.] https://www.kaggle.com/rounakbanik/the-movies-dataset
[15.] https://www.imdb.com
[16.] https://www.themoviedb.org/login
[17.] https://www.machinelearningplus.com/nlp/cosine-similarity