

Dissertation Submitted for the partial fulfillment of the **M.Sc. as a part of M.Sc. (Integrated) Five Years Program AIML/Data Science** degree to the Department of AIML & Data Science.

M.Sc. Project Dissertation

Movie Recommender System

submitted to



By

CHARCHIL SINGH
Semester-VIII

M.Sc. (Integrated) Five Years Program AIML/Data Science

Department of AIML & Data Science.
School of Emerging Science and Technology
Gujarat University

April 2023

DECLARATION

This is to certify that the research work reported in this dissertation entitled **Movie Recommender System** for the partial fulfilment of M.Sc. as a part of M.Sc. (Integrated) in Artificial Intelligence and Machine Learning/Data Science degree is the result of investigation done by myself.

Place: Ahmedabad

Date: 23/04/2023

Name of Student

Charchil Singh

ACKNOWLEDGEMENT

On this occasion of submitting my project work, I would like to thank all the people who have made this possible. The role of the Department of AIML, Gujarat University in shaping my professional capability is very important.

I thank Dr. Ravi Gor, coordinator of the Department of AIML, Data Science & Actuarial Science, for all his support.

Rashmi Madam my mentor has always guided me on the right path and has been a motivating source for innovative project work. This work would have not been possible without her guidance, support, and encouragement. With the help of her guidance, I magnificently overcame the difficulties during my work and learned at each step.

I am heartily and loving thankful to my parents and my friends for their sacrifice, the support & and love which encouraged me to efficiently overcome the difficulties in my pursuit of the project and who always helped me in my work.

Last but not the least, I would not be able to complete the work without the blessings of almighty and would like to thank God for giving me strength and patience to carry out my work with full dedication.

-CHARCHIL SINGH

Index

Sr. No	Content	Page No.
1	Abstract & Key Words	
2	Introduction	
3	Literature review	
4	Methodology	
5	Working	
6	Results	
7	Conclusion	

Chapter 1

Abstract & Key Words

Movie recommendation system proposed whose primary objective is to suggest a recommended list through singular value decomposition collaborative filtering and cosine similarity. The present work improves these approaches by taking the movies' content information into account during the item similarity calculations. The proposed approach recommends the top n recommendation list of movies to users on user's interest preferences that were not already rated. Graphically shows the percentage of already viewed movies by user and movies recommended to User. Now a day's recommendation system has changed the fashion of looking the items of our interest. OTT Movie Application Recommendation for mobile users is crucial. It performs a complete aggregation of user preferences, reviews, and emotions to help you make suitable movies. It needs every precision and timeliness; however, this can be info filtering approach that's accustomed predict the preference of that user. Recommender System may be a system that seeks to predict or filter preferences in keeping with the user's selections. The very common purpose where recommender system is applied are OTT platforms, search engines, articles, music, videos etc. during this work we tend to propose a Collaborative approach-based Movie Recommendation system. It is supported collaborative filtering approach that creates use of the knowledge provided by users, analyses them so recommends the flick that's best suited to the user at that point.

Chapter 2

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 user's 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 a single 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 must 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 watching the movie he/she likes.

Chapter 3

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. We can model and manage data applications in a simple and intuitive manner, and it can also make data units smaller and more standard. It can also realize rich relational links. We introduced Movie REC, a recommender system for movie recommendation, which used Blender and CAD tools. Now we used KNN algorithms and collaborative filtering in order to increase the accuracy of results as compared to content-based filtering. A collaborative filtering technique combines cosine similarity with the knearest-neighbor 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. We 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. propose a content-based movie recommendation system that predicts movie preferences based on temporal user preferences. 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. We. designed a system that combines collaborative filtering and content-based filtering to solve the cold-start problem for new items. 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. We have used methods like swarm-based collaborative filtering, KNN with S-BERT, and universal sentence encoder. This paper also includes how you can handle challenges to systems. The results of the experiment indicate that the system is effective at predicting high quality films. We 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. 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.

Chapter 4

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 highly personalized 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.

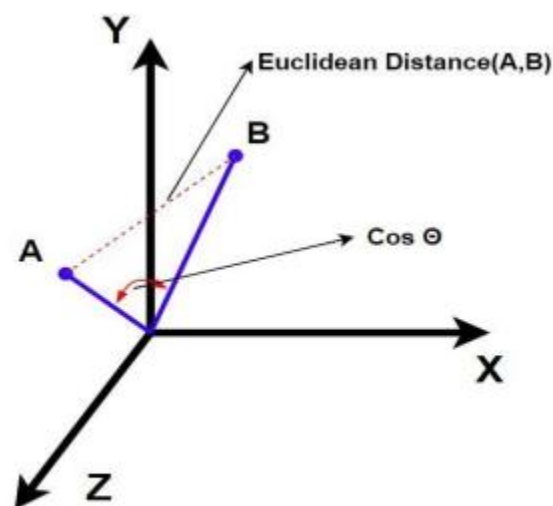


Fig. 1: Cosine Similarity

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. 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.

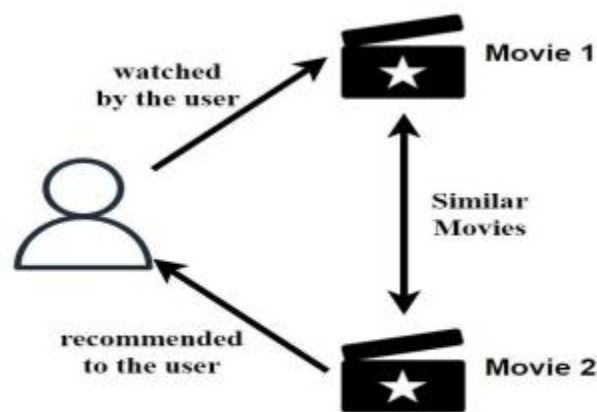


Fig. 2. Content Based Filtering

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 can 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.

$$\text{CosSim}(x, y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}}$$

Fig: To implement Cosine similarity we take an example of 2 movies of different genre adventure and comedy

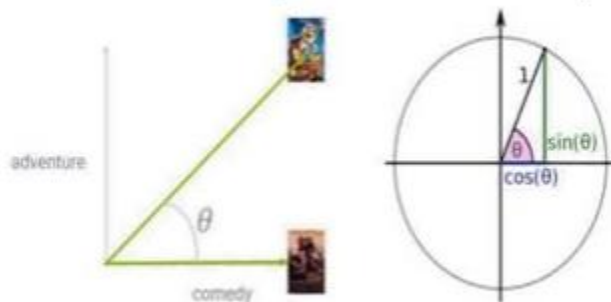


Fig: Cosine similarity

The angle theta between the two movies will determine the similarity between the two movies. The theta ranges from 0- 1. If the value of the theta is near 1 then it is most similar and if it's near to 0 then it is least similar. The movie will be recommended if it is close to 1 otherwise there would be no similarity between them. It will recommend the best movies to the user according to the Cosine similarity. After the cosine similarity, we have used a normalised popular score through which we get our function of computing distance. Then by using the KNN functionality, we have found the nearest neighbour which will be recommended to the user.

Chapter 5

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. 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. Additionally, TMDB (The Movie Database) API is used to fetch other data and images cast and movie posters.
- 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 must 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 must be 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 like the searched movie.

Chapter 6

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.

Chapter 7

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.

Chapter 8

References

<https://www.kaggle.com/rounakbanik/the-moviesdataset>
<https://www.imdb.com> [16.] <https://www.themoviedb.org/login>
<https://www.machinelearningplus.com/nlp/cosinesimilarity>
https://www.youtube.com/live/A_78fGgQMjM?feature=share