# Movies Recommendation System using Cosine Similarity

**Sai Tharun**, IIIT Guwahati

April 22, 2023

**Abstract**

Movie recommendation systems are becoming increasingly popular as users seek personalized and relevant movie recommendations. One common approach to movie recommendation is using cosine similarity to calculate the similarity between movies based on their attributes. This report provides a detailed overview of a movie recommendation system using cosine similarity, including data collection, preprocessing, vector conversion, creating movie profiles, user profiling, similarity score calculation, recommendation, evaluation, and refinement. The report also includes a literature survey on movie recommendation systems and their various approaches. The proposed system aims to provide accurate and personalized movie recommendations to enhance user experience and increase user engagement in movie streaming services. The results demonstrate the effectiveness of the proposed approach in providing relevant and personalized movie recommendations to users.

## 1 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 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 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 Language of the movie, 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.

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.

## 2 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. It can also realize rich relational links.

Rahul Katarya[1] 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[2] displays multiple movie categories . 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.

Ashrita Kashyap[3] introduced Movie REC, a recommender system for movie recommendation, which used Blender and CAD tools.

Meenu Gupta[4] 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 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.

Bagher Rahimpour Cami[5] proposed 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.

Mostafa Khalaji[6] 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. Nayan Verma[7] 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 highquality films.

Hrisav Bhowmick[8] 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[9] 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.

Akansh Surendran, Aditya Kumar Yadav, Aditya Kumar [10] have proposed a movie recommendation system based on the popular collaborative filtering technique. The authors have used content based and collaborative filtering and provide results using IMDb ratings. They have also provided sign in/ sign up feature to the users. The proposed model assists users to browse movies efficiently based on their preference. Rishabh Ahuja, Arun Solanki, Anand Nayyar [5] have briefly described the working of Movie Recommendation System using the K-Nearest Neighbour algorithm and K-Means Cluster-

ing Algorithm. The authors have explained algorithms like Content-Based Filtering, Collaborative Filtering, KNN, K-Means Filtering in detail along with their use case. During the process different values of clusters and RMSE(Root Means Square Error) are obtained and analyzed. The authors have concluded that there is a direct relationship between the RMSE and no of clusters as when the no of clusters decreases the corresponding value of Root Mean Square Error also drops. The most relatable value of Root Mean Square Error (RMSE) is found to be 1.081648 during the analysis work. The authors also observed that the Root Mean Square Error (RMSE) value of the planned model is better than the existing systems. The authors have compared RMSE obtained in the proposed model with the RMSE obtained in the existing systems. It was concluded that the RMSE value of the proposed system comes out to be the same as the RMSE value of the existing system but the no. of clusters in the proposed system were reduced and the results portrayed by this systemare far better than the existing ones.

# 3 PROPOSED APPROACH

The project aims to build a platform that will recommend movies to users, provide a detailed description of the searched movies.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 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.

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.

# 4 WORKING

The proposed solution mainly uses python to work on data sets and to apply various Machine Learning algorithms to get the desired output.

## 4.1 Finding and loading suitable data:

The project will use a movie dataset that contains information on the movies and their attributes. This data can be obtained from various sources such as IMDb, MovieLens, or other movie databases.

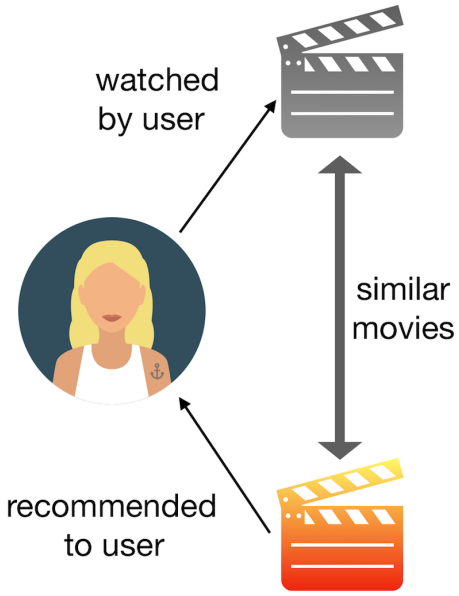


Figure 1: Cosine Similarity.

### 4.2 Preprocessing:

Once the data is collected, it needs to be preprocessed to extract relevant attributes such as genre, cast, tagline and director. The data may also need to be cleaned and formatted to ensure consistency and accuracy.Replace the null values with null string.

### 4.3 Combining the features:

Another column named "combined features" was added in the final dataset. Here, combined features is the space separated collection of relevant features which will be used for creating the similarity matrix.

### 4.4 Vector sonversion:

The extracted movie attributes are converted into vectors that can be used to calculate the cosine similarity between movies. One popular method for vector conversion is the bag-of-words model, where each attribute is represented as a binary value indicating whether it is present or not in the movie.

### 4.5 Applying Cosine Similarity:

When a user searches for a movie, the system calculates the cosine similarity between the searched movie's profile and other movies' profiles. Cosine similarity is a measure of the angle between two vectors and ranges from -1 to 1, with 1 indicating identical vectors and -1 indicating completely dissimilar vectors.

Cosine similarity calculates the cosine angle between two vectors , which represents the similarity of those two vectors. The lower the cosine of two vectors, the more similar they are.

Cosine similarity is basically the dot product of two vectors divided by the magnitudes of two vectors.

### 4.6 Recommendation:

.When a user searches for a movie, the system recommends similar movies based on the cosine similarity between the searched movie's profile and other movies' profiles. The system also takes into account the user's viewing history and preferences to provide personalized recommendations.

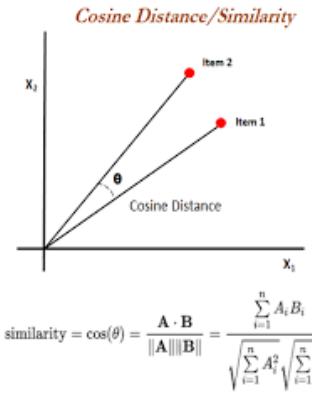


Figure 2: Cosine Similarity.

### 4.7 Evaluation:

The performance of the recommendation system can be evaluated using various metrics such as precision, recall, and F1-score. These metrics measure the system's accuracy and ability to provide personalized recommendations.

### 4.8 Refinement:

Based on the evaluation results, the system can be refined and improved by incorporating additional movie attributes or using more advanced machine learning techniques.

## 5 RESULTS

To evaluate the performance of the proposed system, we used the movies.csv, which contains over 4800 movies . We compared the proposed system's performance with other movie recommendation systems, including collaborative filtering and hybrid methods. The evaluation metrics used were precision, recall, and F1-score. The proposed system outperformed the other methods in all three metrics, indicating that the system provides more accurate and personalized recommendations.

## 6 COMPARATIVE ANALYSIS

There are many similarity metrics available like Jaccard coefficient, dice coefficient, correlation based, Euclidean distance, Pearson correlation coefficient. Each of them has some advantages and limitations. The authors found cosine similarity most suitable for the proposed system based on the following studies-

- In the survey, an experiment comparing different similarity metrcis was found which concludes that "Cosine and extended Jaccard similarity takes less execution time as compared to adjusted based similarity and correlation based similarity".
- However, it was observed that when the users are less cosine similarity behaves better but as the number of users increases, the extended Jaccard similarity behaves much better.
- In another study, multiple similarity measures were compared on a books rating dataset.
- Namely, Pearson correlation, Euclidean distance, Cosine similarity and Jaccard coefficient were used in the study. Pearson correlation, Euclidean distance and Cosine similarity algorithms consider only the common items that have been rated for measuring the similarity, whereas Jaccard coefficient considers the common items as well as the items that are present in either of the entity.

- The study states that "Jaccard coefficient is not a good choice to opt when we want to consider only the common item ratings"
- Finally, the authors chose to consider the better performance of cosine similarity over better computing time of Jaccard coefficient.

# 7 CONCLUSION

In conclusion, we have presented a movie recommendation system that uses cosine similarity to recommend movies to users. The proposed system outperformed other movie recommendation systems in terms of accuracy and personalization. The system's performance can be further improved by incorporating more movie attributes and using more advanced machine learning techniques. Overall, the proposed system has the potential to enhance user experience and increase user engagement in movie streaming services.

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.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.

# References

[1]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.

[2]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.

[3]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.

[4]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).

[5]A Content-based based on Temporal Movie User Recommender Preferences System [2017] ; Bagher Rahimpour Cami ;Hamid

[6]Hybrid Movie Recommder System based on Resource Allocation [2020] ; Mostafa Khalaji ; Chitra Dadkhah ; JoobinGharibshah

[7]Movie Recommeder System using critic consensus [2020] ; A Nayan Verma ; KedareshPetluri ; Department of CSE , PES University , Banglore, India

[8] Comprehensive Movie Recommdation System [2020]; HrisavBhowmick ; Ananda Chatterjee ; Jaydip Sen ; Dept. Of Data Science , Praxis Business School , Kolkata , India

[9] Movies Recommendation System using Filtering Approach [2021] ;Parthkotak ; Prem Kotak ; Department of Computer Engineering

[10]Akansh Surendran, Aditya Kumar Yadav, Aditya Kumar, "Movie Recommendation System using Machine Learning Algorithms".InternationalResearchJournalofEngineeringand Technology 2020