Proceedings of the Third International Conference on Inventive Research in Computing Applications (ICIRCA-2021)
IEEE Xplore Part Number: CFP21N67-ART; ISBN: 978-0-7381-4627-0

Movie Recommendation System using Cosine Similarity with Sentiment Analysis

Harsh Khatter[1] **Nishtha Goel**[2] **Naina Gupta**[3] **Muskan Gulati**[4]
[1]Assistant Professor, Department of Computer Science, KIET Group of Institutions, Delhi-NCR, Ghaziabad
[2,3,4]Student, Department of Computer Science and Engineering
[2,3,4]ABES Engineering College, Ghaziabad affiliated to Dr. APJ AKTU Lucknow
harsh.khatter@kiet.edu[1]
nishtha.17bcs1036@abes.ac.in[2]
naina.17bcs1205@abes.ac.in[3]
muskan.17bcs1082@abes.ac.in[4]

2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA) | 978-1-6654-3877-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIRCA51532.2021.9544794

ABSTRACT: Multimedia is considered as one of the best sources of entertainment. People of all age groups love to watch movies. Movie Recommender System is essential in our social lives as it enhances the field of entertainment. The proposed system on Movie Recommendation System caters the requirements of the user. The major aim is to provide crisp relevant content to the end-users out of semi-structured content on the internet. The main purpose is to generate accurate, efficient and personalized recommendations to the user. Various building blocks of the paper like Introduction, Literature Survey, Proposed System, Implementation & Result, Comparative Analysis, Conclusion and Future Work are discussed in detail. The proposed machine learning model is trained, tested, and a sentiment classifier is generated which classify the sentiments as a good or a bad sentiment. The recommender system is generated by applying Cosine similarity and making API Calls. As a result, the live working of the system generates accurate and personalized recommendations along with the analysis of sentiments for the end users. It is also concluded that Cosine Similarity provides better and efficient results for a recommender system.

Keywords: Cosine similarity, Information Retrieval, Machine Learning, Movie Recommendation System, Personalized search, Recommendation System, Sentiment Analysis.

I. INTRODUCTION

With the increase of World Wide Web and high-speed internet, multimedia is turned out as one of the best sources of entertainment. People are keenly interested in watching movies. However, there are thousands of movies available and it becomes extremely difficult for people to choose a suitable movie. So, Movie Recommendation Systems solves this problem by providing accurate suggestions based on people interests. It also recommends movies that people are generally not aware of and thus saves a lot of time. For ex: A person who is a fan of Avengers Series and has already watched "The Avengers", "Avengers: Infinity War" and "Avengers: Age of Ultron". So, the recommendation system would suggest "Avengers: The End Game.

A recommender system is that system which generates recommendations based on the user's interests and needs. It basically recommends those items to the user that seems to fulfils his/her needs and also analyses the sentiments on the reviews given by the user for that movie. As shown in Fig 1. Movie recommendation systems provide a user with the movie suggestions that are more likely to be watched by him using some means like, the users past behaviour, or user's profile etc. It generates personalized recommendations based on user's likes, ratings and dislikes.

The major goal is to provide related content to the users out of relevant and irrelevant collection of items to the users of Movie Recommendation System. The authors aim to deliver a Movie Recommendation System which caters all the requirements of the user. Due to high-speed internet and various platforms releasing movies of different genres, people of different generations binge watch the latest movies and shows. All these addicted users desire to watch movies of their interests. Movie Recommendation System caters to all the needs of the users such that the user feels comfortable and is able to interact with the system. So that's what the proposed system does it recommends the movie according to user's interest.

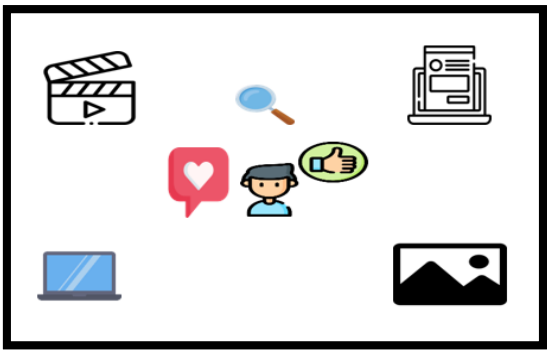

Fig 1. Personalized Recommendations

Nowadays people are more active on social media, online shopping websites, Netflix, amazon prime and so on. People also share their thought about a particular product by writing the reviews or comments about it. While online shopping or online watching movies people generally go through the reviews of the item which is the main source of someone's sentiments [1]. Sentiment analysis helps in making business decision. Sentiment analysis is used to highlight the week and good point of the product so that one can make better business decision.

As we have discussed earlier user depends on other to suggest some content of their personal choice and interest which they can binge watch in their free time. So the authors are building this system to remove this kind of dependency on others because it is difficult for both the sides to involve in such an activity which wastes time and result will also not be that much accurate as it is given by a recommender system.

Hence, with the help of user's preferences and interest a better structure with accurate recommendations is provided to the users.

II. LITERATURE REVIEW

Recommender systems can be utilized in many contexts to generate recommendations to the users that might interest them. Recommendation systems were first developed by Tapestry project in 1992[5]. The Tapestry project (*first* commercial *recommender system*) introduced the term "collaborative filtering" Existing movie recommendation systems are mostly built using the content based and collaborative filtering approach. They show the recommended movies to the user on the basis of user's priorities using collaborative filtering. It is a technique that is used to filter those items which a user might like on the ground of reaction by similar users [6]. It works by finding smaller group of users from the larger set of people. While content-based Filtering i.e., another approach uses an item features in order to recommend other items with similar features or properties. Except these two approaches there are also some other approaches proposed in the past which is discussed below.

Harsh Khatter, Anil Kumar Ahlawat [1] have proposed a model based on content curation algorithms which are based on personalized web searching. They suggested a new technique which is to use clustering technique for recommendations along with association rule mining. The system proposed by them automatically provides the recommendation to the user as soon as he/she logged in. The expected outcome of this model is the top N relevant recommendations to the user based on the user's interest. M. Chenna Keshava,S. Srinivasulu, P. Narendra Reddy and B. Dinesh Naik[2] focussed on providing recommendations to the user based on the user's ancient data. The authors also committed to provide accurate recommendations and suggest much more content to an individual. They have used the concept of CineMatch. They increased CineMatch Algorithms by about 10% with the use of some Collaborative Filtering Techniques. The authors have used various ML algorithms like XGBoost for featuring the data, Surprise Baselineonly Model for training and testing data, Surprise KNN Baseline Model for updating the data by using the features obtained from Surprise Baselineonly Model, Matrix Factorization SVD for using next to update the data at every particular instance of time, Matrix Factorization SVDpp for updating the model with the obtained features. They have concluded that the best model was Matrix Factorization SVDpp which provided the RMSE Value of 1.0675. V.R.Azhaguramyaa, Hemanshu P Thakker, Murali Manohar K S, and Mithun K[3] have discussed the RNN algorithm for product recommendation. The authors have used Recommendation system as a filtering system which is used to predict the product that the user would like to brought or purchase. The proposed system recommends the movie according to the Comments and ratings provided by the user.

Akansh Surendran, Aditya Kumar Yadav, Aditya Kumar [4] 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 Clustering 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 system are far better than the existing ones.

Nirav Raval, Vijayshri Khedkar[6] authors have discussed the Collaborative filtering based approach for movie recommender system. The Recommender system uses the neural network model technique to train the system or model which helps to learn the user-item interaction. The main objective of their system is to recommend movie to the user based on the user's previous behavior and also based on the similar decision made by some other users. P. Stakhiyevich and Z. Huang[7] have focussed on the impact of strong user profiles on the movie recommendation results. Vector space model(VSM) based model is used to construct the user profile. They have concluded that user profiles based on data with additional features are significantly more effective in personal recommendations than user profiles based on data with fewer features. Gayatri Khanvilkar, Prof. Deepali Vora [8] have discussed the Random forest algorithm for sentiment analysis for product recommendation. The authors have also considered sentiment of the user along with user's profile for recommendation of the product. The main objective of the system is to provide recommendation based on sentiment analysis. The proposed system recommends the product according to the user's profile and sentiments analysis. The authors also observed that random forest is not only effective for accuracy but also for robustness.

Nalmpantis, Orestis and Tjortjis[9] described a method which aims to incorporate the personality of a user into a movie recommendation system as "50/50 Recommender". The authors have claimed hypothetically that if personality is incorporated into movie recommendation Systems, it will increase its performance 10 folds. So, the main motion revolves around the examination of incorporating personality into the Recommendation Systems. In order to personalize the Movie Recommendation System, the authors have introduced a concept of combining a personality test with the Collaborative Filtering Technique. They have combined an existing Movie Recommender System with the Big Five Personality Test. The first main contribution of the proposed system is combining the Personality with KNN Flow Chart. The authors have concluded that personality plays a major role in a recommender system and improved the recommendation quality. The results portray that there are 3.6% more users which use 50/50 Recommender System. The 50/50 recommender system overpowers the performance of KNN Recommender system and thus it is widely accepted by the users. The authors have concluded that the 50/50 Recommender System shows exceptional results as it provides better personalized recommendations than the existing ones. Vellaichamy, Vimala and Kalimuthu[10] proposed a hybrid recommender system that combines Fuzzy C Means clustering (FCM) technique with the bat optimization method to reduce scalability issues. The authors have implemented it in two phases – FCM and BA. Experiment results show that the proposed algorithm can produce better recommendation results as compared to other models in terms of measures like Mean Absolute Error (MAE), precision and recall. The proposed system also reduces scalability and data sparsity problems which occur in traditional recommender systems.

Bei-Bei CUI [11] has generated a prototype of Movie Recommendation System combined it with the existing needs of the recommendation systems by the extensive use of KNN Algorithms and Collaborative Filtering Algorithms. The author has also considered the JAVAEE system Database model to implement the proposed model. The author aims to automatically generate personalized recommendations to the user by developing a prototype by the help of KNN Algorithm. The author has used KNN Filtering Collaborative Algorithm. The collaborative Filtering Algorithm is combined with KNN Algorithm. KNN is used to select neighbours. Cosine Similarity and Pearson Correlation Similarity is used in the paper. Rupali Hande , Ajinkya Gutti , Kevin Shah , Jeet Gandhi , Vrushal Kamtikar [12] have discussed the hybrid algorithm for movie recommender system which is called as content-boosted collaborative filtering system. The main objective of their system is to recommend movie to the user's based on the user ratings that they have provided and their earlier history. The authors have applied collaborative filtering to the pseudo user-rating matrix which helps to make recommendations to the users. Saikat Bagchi [13] has laid out an analysis of performance of different similarity measures (provided by Apache Mahout) which are used in collaborative filtering. The author has compared and analysed similarity measures like Euclidean distance, city-block, uncentered cosine, pearson correlation, spearman correlation, Tanimoto coefficient, Log likelihood.

Bela Gipp, Joran Beel, Christian Hentschel [14] have laid out the first Hybrid based approach first recommender system which is known as Scienstein. They have generated an extremely powerful alternative of the existing engines. The authors have combined Citation Analysis, source analysis, Author Analysis, Implicit Analysis, and Explicit Analysis. They have used some never used methods like Distance Similarity Index(DSI) and In-Text impact Factor. They basically focussed on combining Content Based and Collaborative Based Filtering. Also, they have offered extremely user-friendly GUI to the users in order to handle complex situations. Bo Pang, Lillian Lee, Shivakumar Vaithyanathan [15] have examined many machine learning techniques for sentiment classification problems. The authors have compared efficiency of 3 techniques - Naive

Bayes, maximum entropy classification, and SVM. They have concluded that Naive Bayes model is found to be optimal in cases with highly dependent features. Maximum entropy classification model is supposed to perform well in NLP applications. To prevent overfitting, parameter training was performed. In SVM, the aim is to find a hyperplane whichh best classifies the two classes. Joachim's (1999) SVM light package is used for training and testing model. It was found that all the three models achieve 90% or more accuracy in topic-based classification. Also, they have concluded that sentiment classification is more difficult than topic categorisation.

Limitation and Issues of Existing Models:

Most of the existing Models do not show any kind of review to the users. The existing models do not display all the information related to the particular movie. Most of the existing models do not consider Sentiment Analysis. The existing models don't contain any past history of the users for providing the recommendation. The recommendations provided by the existing models are not personalized and does not cater to all the requirements of the user. Some of the existing models use very limited datasets. We have considered all the limitations in the existing models and try to work on these issues so as to provide the best movie recommendation system.

III. PROPOSED SYSTEM

According to Fig 2, In the proposed system the end users interact with the system UI and enter any movie name which they want to search. The searched movie's related information is obtained by fetching the TMDB API. The API call provides all the information for that particular movie and the reviews are collected from the dataset. Then, sentiment analysis is done on the collected reviews to determine whether the sentiments are good or bad. After sentiment analysis, all the movie details and the ratings are passed to the recommendation model. This model aims to provide recommendations to the users as per their interests. The recommendation model uses the concept of Cosine Similarity. Finally, the generated predictions are provided to the users in the form of recommended movies. Thus, the system is able to provide accurate, efficient and personalized recommendations to the users.

The proposed system has two main features – review sentiment analysis and recommending similar movies to the end user. In this paper, the focus is on the recommendation System. The recommendation model uses similarity scores to recommend a similar movie. Similarity score helps to determine how much two items are similar to each other on a scale of zero to one. Cosine similarity is a famous metric to calculate similarity between two vectors. In this model, each vector represents a movie's features and they have an angle θ between them. To calculate the similarity between two movies, all the features of those two movies are combined into a feature vector. The cosine similarity will measure the similarity between these two vectors which is a measurement of how similar the two movies are.

Table 1. Comparison table of different approaches

S.No.	YEAR	TITLE	METHODOLOGY USED	GAPS IN WORK
1	2020	Analysis of Content Curation Algorithms on Personalized Web Searching	Clustering technique with association rule mining	1) It doesn't not show any kind of review to the user related to any item. 2) It doesn't provide rating of any item so user will not be able to take quick decision whether he/she should consider that item or not.
2	2020	Machine Learning Model for Movie Recommendation System	Cinematch Algorithm, KNN Algorithm, Collaborative Based Filtering Algorithm	1) The main limitation of the proposed model is that the hyper parameters of the XGBoost Model do not improve the RMSE Value. 2) The cloud resources are not used. 3) The size of users and the movie date is very limited.
3	2020	Smart Product Recommender System using Machine Learning	Recurrent Neural Network Algorithm	It doesn't not consider user's past history for providing the recommendation.
4	2020	Movie Recommendation System using Machine Learning Algorithm	Collaborative Filtering and Content Based Filtering	Less accurate than most of the modern hybrid systems

5	2019	Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor	Implemented using K-Means Clustering, K-Nearest Neighbor Algorithms considering the RMSE Values.	1) This paper doesn't consider Sentimental Analysis. 2) The data sets were limited.
6	2019	Collaborative Filtering Based Movie Recommendation System	Neural Network Model	It doesn't consider any side feature of an item or movie such as country, age
7	2019	An experimental study of building user profiles for movie recommender system	Vector Space Model	Feature importance is not considered

IV. IMPLEMENTATION AND RESULT

The steps to follow in the proposed model are mentioned below:

Step – 1: Data collection - The movies dataset is collected from Kaggle and Wikipedia. The data from Kaggle is for movies up till year 2016 and dataset for movies after 2016 was taken from Wikipedia. Then, both datasets are combined in a common format to get the final dataset.

Step – 2: Data Analysis - The original dataset from Kaggle consisted 28 features in total including movie title, director name, director's Facebook link, actors' name, language, etc. A lot of the features are irrelevant for the proposed model and do not really contribute to the end result.

Dataset was finally reduced down to following 7 columns - movie_title, director_name, genres, actor_1_name, actor_2_name, actor_3_name, title_year.

Step – 3 : Combining the features – Another column named "combine_info" was added in the final dataset. Here, combine_info feature is the space separated collection of other features except the movie_title which will be used for creating the similarity matrix.

Step – 4: Applying Cosine Similarity - While there a number of similarity metrics which can be used in a recommendation system, the authors chose to use cosine similarity.

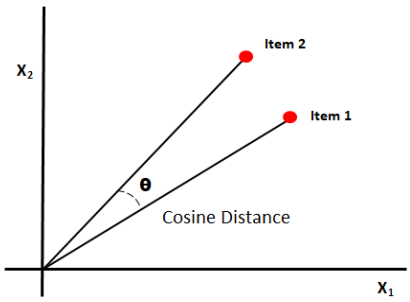

Fig 3. Cosine Distance/Similarity

Cosine similarity calculates the cosine angle between two vectors as shown in Fig 3, 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.

$$similarity = \cos(\Theta) = \frac{A.B}{|A||B|}$$

- **A.B** = dot product of the vectors 'x' and 'y'.
- ||**x**|| and ||**y**|| = length of the two vectors 'x' and 'y' respectively.

It ranges from 0 to 1 with 0 being the lowest (the least similar) and 1 being highest (the most similar).

Cosine similarity matrix is created using cosine_similarity() function present in python's "sklearn.metrics.pairwise" package. The scores are calculated over the **combined_info** feature of our dataset which includes director's name, actors' name, genres, and title_year of the movies.

Step – 5: API Calls – Movie details like overview, duration, genre, release date, poster is scraped using IMDB API to provide details of the movie to users.

Step – 6: Movie Recommendation - When the system has to recommend movies similar to the given movie, it reads the cosine similarity matrix row for that movie and checks the similarity score of this movie with other movies. Then, the movies having higher score are recommended to the user.

V. COMPARATIVE ANALYSIS

As discussed, 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 the adjusted based similarity and correlation based similarity"[17]

- 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[16].
- 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"[16].
- Finally, the authors chose to consider the better performance of cosine similarity over better computing time of Jaccard coefficient.

VI. CONCLUSION

Proposed recommendation system with sentiment analysis is very useful for the personalised recommendation and also helps in making business decision. Sentiment analysis really helps to highlight the week and good point of the product so, that we can make better business decision. Here, Cosine similarity is used over many other available similarities for the recommendation system because it has better computing time and efficiency than others. Another advantage of cosine similarity is that it can still give smaller angle between two similar objects even if they are far apart by the Euclidean distance.

This approach can be used as a base for other recommender systems which recommends songs, books, news, videos etc. It can also be incorporated into various ecommerce websites. Also, the system can further be improved such that the user can also provide rating and comment on the movies.

REFERENCES

[1] Khatter, Harsh and Kumar Ahlawat, Anil, "Analysis of Content Curation Algorithms on Personalized Web Searching" (March 29, 2020). Proceedings of the International Conference on Innovative Computing & Communications (ICICC) 2020, http://dx.doi.org/10.2139/ssrn.3563374

[2] M. Chenna Keshava, S. Srinivasulu, P. Narendra Reddy, B. Dinesh Naik, "Machine Learning Model for Movie Recommendation System," International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181 IJERTV9IS040741 Vol. 9 Issue 04, April-2020, DOI: 10.17577/IJERTV9IS040741

[3] Prof.V.R.Azhaguramyaa, Hemanshu P Thakker, Murali Manohar K S, Mithun K, "Smart Product Recommender System using Machine Learning," International Journal of Advanced Science and Technology ,Vol. 29, No. 9s, (2020).

[4] Akansh Surendran, Aditya Kumar Yadav, Aditya Kumar, "Movie Recommendation System using Machine Learning Algorithms". International Research Journal of Engineering and Technology 2020

[5] R. Ahuja, A. Solanki and A. Nayyar, "Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor," *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2019, pp. 263-268, doi: 10.1109/CONFLUENCE.2019.8776969.

[6] Nirav Raval, Vijayshri Khedkar, "Collaborative Filtering Based Movie Recommendation System," International Journal of Scientific & Technology Research, Vol 8, Issue 12, Dec 2019.

[7] P. Stakhiyevich and Z. Huang, "An Experimental Study of Building User Profiles for Movie Recommender System," *2019 International Conference on High Performance Computing and Communications;* Zhangjiajie, China, 2019, pp. 2559-2565, doi: 10.1109/HPCC/SmartCity/DSS.2019.00358.

[8] Gayatri Khanvilkar, Prof. Deepali Vora, "Sentiment Analysis for Product Recommendation Using Random Forest," International Journal of Engineering & Technology 2018.

[9] *Nalmpantis, Orestis. "The 50/50 recommender: personality in movie recommender systems." Published in* Engineering Applications of Neural Networks, Springer International Publishing *2017.*

[10] Vellaichamy, Vimala & Kalimuthu, Vivekanandan. (2017). "Hybrid Collaborative Movie Recommender System Using Clustering and Bat Optimization". International Journal of Intelligent Engineering and Systems. 10. 38-47. 10.22266/ijies2017.1031.05.

[11] Bei-Bei CUI School of Software Engineering, Beijing University of Technology, Beijing, China, "Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm". January 2017, ITM Web of Conferences 12(8):04008

[12] Rupali Hande , Ajinkya Gutti , Kevin Shah , Jeet Gandhi , Vrushal Kamtikar, "Moviemender- A Movie Recommender System". International Journal Of Engineering Sciences & Research Technology 2016.

[13] Saikat Bagchi (2015), "Performance and Quality Assessment of Similarity Measures in Collaborative Filtering Using Mahout", 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

[14] Bela Gipp, Joran Beel, Christian Hentschel, "Scienstein: A Research Paper Recommender System", in *Proceedings of the International Conference on Emerging Trends in Computing (ICETiC'09)*, 2009

[15] Bo Pang, Lillian Lee, Shivakumar Vaithyanathan (2002), "Thumbs up? Sentiment Classification using Machine Learning Techniques". Proc. 2002 Conf. on Empirical Methods in Natural Language Processing (EMNLP)

[16] Mr. Sridhar Dilip Sondur, Mr. Amit P Chigadani, Dr. Shantharam Nayak "Similarity Measures for Recommender Systems: A Comparative Study", May 2016, published in Journal for Research.

[17] Madhuri Angel Baxla (2014), "Comparative study of similarity measures for item based top n recommendation", NIT, Rourkela.