Movie Recommendation System Using Machine Learning

Aayush Khanna and Kartik Arya

September 30, 2023

MOVIE RECOMMENDATION SYSTEM BASED ON MACHINE LEARNING

Aayush Khanna
Computer Science Engineering
Chandigarh University
Gharuan, Punjab, India
aayushkhanna0201@gmail.com

Kartik
Computer Science Engineering
Chandigarh University
Gharuan, Punjab, India
kartikarya1345@gmail.com

Abstract

The exponential growth of digital content has led to an overwhelming abundance of movies and TV shows, making it increasingly challenging for viewers to discover content that aligns with their preferences. To address this issue, the Movie Recommendation System based on Machine Learning has emerged as a promising solution. This abstract provides an overview of such a system.

In this research paper we are going to develop a model to recommend movies based on item-based filtering and not on user based. Most of the movie recommendations used by organizations are using user based collaborative filtering but this limits us to recommend movies when we get a sufficient data of user to find their behavior.

Item based filtering usually maps the item with one another using some criterion or model and here we will use vectorization for that. This model will map all the vectors which is here referred as movies on a single point and will work on that to find angle between them and the most similar movies will be the one with smallest angle between them.

Key components of the system include data collection and preprocessing, feature engineering, and model training. Item-based collaborative filtering, are used to establish relationships between users and movies based on their interactions. Content-based filtering analyzes movie attributes like genre, actors, summary, and user preferences to create content-based recommendations. Hybrid methods combine collaborative and content-based filtering to enhance recommendation quality.

In conclusion, the Movie Recommendation System Based on Machine Learning leverages advanced algorithms to provide users with personalized and engaging movie recommendations, addressing the challenge of content discovery in the digital age. This system not only enhances user satisfaction but also benefits content providers by increasing user engagement and retention.

Keywords: Movie Recommendation, recommendation system, natural language processing, vectorization, collaborative filtering, personalized recommendation, user preferences.

I. INTRODUCTION

In an era characterized by the exponential growth of digital media and entertainment consumption, the importance of personalized movie recommendation systems cannot be overstated. The vast array of available movies, coupled with diverse viewer preferences, has made it increasingly challenging for individuals to discover content that resonates with their tastes and interests. As a result, movie recommendation systems have emerged as a pivotal solution to help users navigate this overwhelming cinematic landscape. The pursuit of enhancing movie recommendation systems has become a central focus of research in recent years, fueled by the growing demand for more accurate and effective content recommendations. These systems, powered by artificial intelligence and machine learning algorithms, play a crucial role in shaping the way audiences engage with movies and discover new cinematic gems. They not only contribute to user satisfaction by facilitating content discovery but also offer invaluable insights into consumer behavior and preferences for content providers and streaming platforms. This research paper delves into the realm of movie recommendation systems, aiming to present a comprehensive overview of the state of the art, recent advancements, challenges, and potential future directions in this exciting field. By addressing these facets, we hope to contribute to the ongoing discourse surrounding personalized content recommendation, ultimately fostering a more engaging and satisfying movie-watching experience for viewers worldwide.

II. LITERATURE SURVEY

(i) Existing System:

Existing recommendation algorithms can be divided into four kinds: content based, knowledge-based, Collaborative Filtering, and hybrid. Among these recommendation algorithms, CF is the most popular technique, based on the core assumption that users who have expressed similar interests in the past will share common interests in the future [1].

CF methods can be model-based or memory-based. Model-based algorithms first construct a model to represent user behavior and, therefore, to predict their ratings. The parameters of the model are estimated using the data from the rating matrix. There are many model-based approaches: Principal Component Analysis (PCA) is based on algebra; Bayes methods are based on statistics.

Reference [2] User-based collaborative filtering algorithms generate recommendations based on the preference of similar users. In contrast to user-based CF [4], item-based CF approaches recommend items on the basis of information about other items that a user has previously rated. The recommended items for the given user are ranked by the similarities between each candidate item and other items that the user has rated.

Reference [5] designs a movie recommendation system using data clustering and computational intelligence, designing an algorithm featuring K-means clustering and cuckoo search optimization, and evaluating the recommendation performance on the Movie Lens dataset. Based on this assumption, we designed a CF algorithm, KM-Slope-VU, in which, according to users' profile attributes, K-means is utilized to partition users into several clusters, and then each cluster produces an opinion leader by calculating the average rating of the items [1].

We also intuit that the historical data of user evaluations of items are naturally correlated to their tastes, and therefore should be utilized to cluster users. A system recommends different movies to users. Since this system is based on a collaborative approach, it will give progressively explicit outcomes contrasted with different systems that are based on the content-based approach. Content-based recommendation systems are constrained to people, these systems don't prescribe things out of the box. These systems work on individual users' ratings, hence limiting your choice to explore more. [3]

(ii) Proposed System:

Since the existing system have developed many features it's a needed time to develop a new model based on approach of merging all of these approaches and finding the best one to be the best model for movie recommendation system. As everyone are aware of OTT platforms, we can see that they use recommendation system and recommend the best movies for their customers to get the best viewership but for customers it is not always the best thing.

Most of the times the movies customers want to watch are not limited to one platform and one movie from a platform may not have the best recommendation movie on the same platform and that movie will not be recommended. So, there is a need of an independent platform with movies from all OTT platforms as well as others in cinemas only to recommend. Along this the proposed system should also try using different algorithms as one of them which is not even used till now is vectorization. This method places all the movies on vectors on the basis of similar words in the title and summary as well as the actors name and director's name.

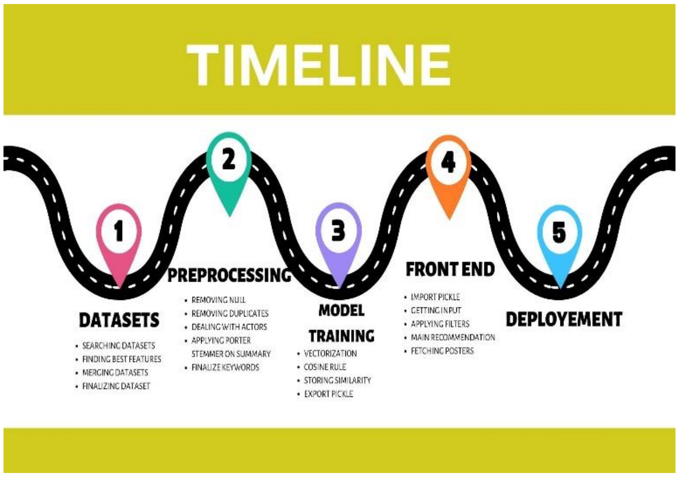
After placing the movies on the vector, we can use another method called cosine similarity to find the angle between movies in vectors so that the movie with smallest angle will be the most similar movie and must be recommended. After the technique also may be the accuracy will increase but there are well designed systems already for the recommendation systems so for new functionality, we will add some extra filters systems called personalized filtering so that the user can filter the movies based on time of release, actor, language, etc.

Along with this the main thing that we have to improve is the scalability, as most of the movie recommendation systems are based on datasets with Hollywood movies. We have to add some Bollywood movies also in that system to make Indians audience also attracted towards it.

III. METHODOLOGY

As far as discussed in the previous section under proposed system about the new technique must be used here. Now we will discuss in detail about this technique and how will it work for our project and domain to recommend best movies. As like any other project we first start with getting the datasets after which followed by preprocessing the data and here, we will make some keywords which we can use to recommend movies based on them which we are going to make using the data available. After preprocessing comes our model training step which the definitely most important as well as the most unique and highlighted feature of this study and project. We are going to train the model using vectorization and then going to use cosine rule to measure the similarity in terms of cosine angle about which we are going to discuss here in a while. After getting the data lied on the vector which different angles, we will use that to find best movie for our new data by finding nearest vector. After whole part we are also going to make a front end to interact with user and display output with their pictorial representation of movies.

This whole overview of our methodology is summarized here in this flowchart.



1. Datasets

According to this flowchart our first step is going to be getting the dataset which is a crucial step to get better results because getting a good dataset or getting a dataset with less wrong or missing values will result in better recommendation always.

First of all, we are going to search of some datasets on websites which provide globally free datasets like Kaggle and others. After getting datasets we are going to include some Bollywood movies also because most datasets do not include Bollywood movies so we are trying here to get them too. After finalizing some datasets, we will go through all of them and find the best features and variables that are going to help us with the recommendation process. If we get same variables in more than one dataset then we will go with the one with less mistakes or wrong or missing values.

There can be a situation also where we may get stuck with different variables on different datasets but want to include all, we are going to merge datasets if possible. We will merge them on the basis of some common variable which can be movie in our case and with this we will finalize our dataset with the best features.

In this model we have taken some datasets from Kaggle and gone through them to find the best features and finalized a dataset. Our dataset is going to include features such as movie title, imdb id, movie summary, actors, genres, year of release, imdb rating, story, tagline, release date.

2. Preprocessing

After getting the dataset we are going to preprocess it to find out whether the features we have finalized are really going to add some meaning or not. So first of all, we are going for irrelevant features and found as date of release as the date doesn't matter but the year does which is already a feature so we are going to drop date of release as a feature. Another thing we are going to see is the number of null values so by using some python function we first find which features are not good for prediction and we find that tagline is a feature which can add a little meaning to our prediction only and is getting a lot of null vales so we decide to drop this column too. After that a similar feature we saw is with story as that column is too much correlated with summary and most of the entries have summary just a part of story and also story are getting some null values so we decide to drop it for now but this can be included later if needed.

As after null values we will handle some duplicate values, which we did not get so much so just by using some python functions we drop those duplicated values and as result we got some unique entries without null or duplicate values.

Under the preprocessing, our first part which was data cleaning is done and this is the time to do more processing to the data to get the data in a form to use for recommendation. So first we are going to actor's column and look up on that to know that every actor doesn't add value to the movie and the audience just use some actors to watch another similar movie of them so we tried to find the best number of actors which we should use for recommendation and tried some by using manual power to get a number which we thought is the best as 5.

After actor's column we tried to get other columns and we are going to make a column named keywords where we will include all the keywords which are necessary for recommendation and we will include keywords from the existing columns such as summary, genres, actors and title. For this we have to process these words as we can't directly use them to make keywords. As u know some similar words do not make any difference in meaning but are just used differently because of grammar. For example, let's take some words "plays", "played", "playing" and all of them have them have the same meaning but are used at different cases and this will create a new word and will treat everyone differently and doesn't count it as similar. So, we are going to use here a technique called stemmer porter which is a function inside library named natural language toolkit and comes under natural language processing and it will convert all the words to their base words and as the example given "plays", "played", "playing "will all be converted to base word as "play" and thus all will be treated same. After applying this stemmer porter to the major column like summary, we can move forward to include all words and make another column named keywords. We should also keep in mind that names such as actors name and surnames should be treated as one word by removing the space between them so that processing will not be done based on names or surnames only.

3. Model Training

So next step after the preprocessing is model training which is the most important and highlighted step of our research and project. Until now we have made our data ready for training by creating keywords of the data.

Now the first step under model training will be vectorization which is a function under scikit learn library which usually contains many algorithms and models for machine learning. This vectorization is an approach which apply all the vectors on a scale of 180 degrees from a same point. On that point it will depart many vectors with an algorithm by placing the most similar vectors closest and next to each other. In our case the vectors are termed as movies and the most similar movies are plotted on the vectors according to those keywords.

Now the main problem is to how to find out that which vector is actually nearest to the required one so for that we use another technique called cosine similarity which comes under another library of scikit learn called metrics, this technique can find the angle between any two vectors in our case and convert it into cosine form. It finds angle between all the vectors her and create a data frame where every movie can be mapped to each other and an angle would be there to showcase the similarity between them.

Now that we have found the movies that are best related to each other our task is to just fetch the movies from the data frame and this is not a huge task with just one thing in the mind which is that we have to exclude the movie at the top in similarity ranking as obviously the movie with lowest angle with another and closely related is the movie itself so have to exclude that movie and starting fetching the movie from the second best correlated and display from that only.

4. Frontend

So, our model is developed until now and is totally ready to be deployed for use because it already started recommending the movies based on an input movie but before that for user interaction to give user a model where they don't have to think about the backend and how the model was created and they can just input a movie that they want to be used to recommend other movies.

First of all, we will create an input tab to input the movie name with giving suggestions to user to neglect any spelling mistakes and can just output the movies by running the model based on that movie.

Up to this our model can work and be available for users but to create better environment we will add some extra functionality by applying features to user filters on the movies while recommending. For example, we can add a filter to recommend movies latest to some year only, also we can add a feature to recommend movie with imdb rating above a level only which the user can set. And this is the reason why we didn't drop these columns before so that these can used to filter data. Another feature can be sorting on the basis of imdb rating which will not only be optional by user interaction but can be default also.

After all these methods we are going to add posters for which we have not dropped imdb id before and also poster path which can be used to fetch the poster from the source and display with the movie name when the user asks for recommendation.

IV. CONCLUSION

This research has come to a result that how the system actually works on this technique which is not generally used by organizations which actually have a reason. Organizations like Netflix, Amazon prime video and Disney Hotstar does not use these techniques for recommending movies and instead they use user based collaborative filtering which recommends movies based on similar users. For example, if user 1 and user 2 usually watch similar movies so any movie that user 1 watch will be recommended to user 2. But this beholds a drawback as it requires a lot of pre used data to match users and only an organization can do this because they already store users' data.

But now let's take a case where we are not an organization and just want to recommend movie basis of another movie because we can't track user behavior so this case need another solution based on making similarity on movies instead on users. For this case our solution is one of the best one as we are also creating an independent website for it which does not track user choice like these streaming platforms.

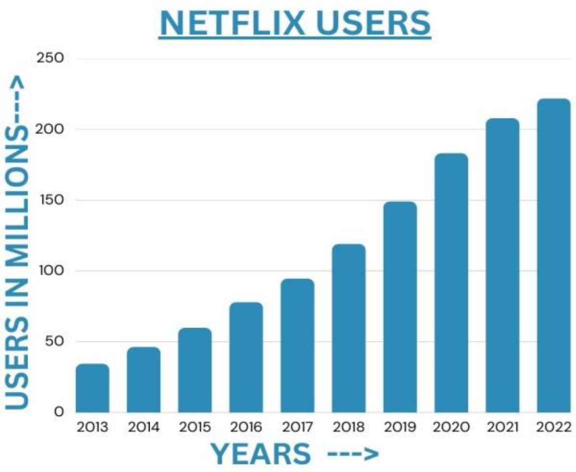
And if we talk about another solutions, we all know k means and knn are already being used here but they are not too efficient while handling categorical data and also, they need number of neighbors which is k to be defined before working.

Our system has successfully derived some results which were phenomenal for our case and users got a great experience using it.

V. FUTURE USE

As much as discussed till now we have understood that the system is going to recommend the best results in case of an independent platform. If we talk about future results, it is not going to be decreasing as we have seen in past years how the use of ott platforms have been increased just because they don't their users go easily and recommend some other movies related to that which attracts the audience which indirectly result in increasing nature of our project also.

Let us explain the process behind that, as we know users usually watch a movie on some ott or cinema and also download it from third party sites but users are not recommended movies in case of last two. User always want to watch another movie just like that if he/she likes that movie and usually search the same on google but they do not find anything positive but if this project is deployed with some Bollywood movies they can use for recommendations and will remember the function forever and just like some other websites it can be popular and used worldwide.



In this picture above we can see that Netflix users have been increasing every year and that too after covid have increased very frequently and we have seen that it is correlated to our model as well.

VI. REFERENCES

[1] J. Zhang, Y. Wang, Z. Yuan and Q. Jin, "Personalized real-time movie recommendation system: Practical prototype and evaluation," in Tsinghua Science and Technology, vol. 25, no. 2, pp. 180-191, April 2020, doi: 10.26599/TST.2018.9010118.

[2] Zan Wang, Xue Yu, Nan Feng, Zhenhua Wang, an improved collaborative movie recommendation system using computational intelligence, Journal of Visual Languages & Computing, Volume 25, Issue 6, 2014, Pages 667-675, ISSN 1045-926X, https://doi.org/10.1016/j.jvlc.2014.09.011

[3] Furtado, F., & Singh, A. (2020). Movie recommendation system using machine learning. International Journal of Research in Industrial Engineering, 9(1), 84-98. doi: 10.22105/riej.2020.226178.1128

[4] Arora, G., Kumar, A., Devre, G.S. and Ghumare, A., 2014. Movie recommendation system based on users' similarity. International journal of computer science and mobile computing, 3(4), pp.765-770.

[5] M. Deshpande and G. Karypis, Item-based top-N recommendation algorithms, ACM Trans. Inf. Syst., vol. 22, no. 1, pp. 143–177, 200

[6] C. -S. M. Wu, D. Garg and U. Bhandary, "Movie Recommendation System Using Collaborative Filtering," 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2018, pp. 11-15, doi: 10.1109/ICSESS.2018.8663822.

[7] Kumar M, Yadav DK, Singh A, Gupta VK. A movie recommender system: Movrec. International journal of computer applications. 2015 Jan 1;124(3).