IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

CONTENT RECOMMENDATION SYSTEMS AND APPROACHES USING HYBRID FILTERING

[1] Kusumalatha Karre, [2]Sai Charan, [3]C Rohit, [4]Hari Krishnan

[1]Assistant Professor, [2] Student, [3] Student, [4] Student
[1]ECM Department,
[1]Sreenidhi Institute of Science and Technology
Ghatkesar, Hyderabad, Telengana

Abstract: Even though many people love watching movies, their choice of watching movies isn't the same for every person, differs based on their various preferences like genre, actors, directors etc. When we take all this into account, it's astoundingly difficult to generalize a movie and say that everyone would like it. To tackle this, we use Recommendation systems such as hybrid filtering. A movie recommendation system using hybrid filtering combines content-based and collaborative filtering methods to suggest movies to users. The content-based method uses movie features such as genre, plot, and actors to find similar movies, while the collaborative filtering method utilizes user behavior such as watching history, ratings, and user similarity to make recommendations. By combining these two methods, the system can provide more accurate and diverse movie suggestions to users.

Keywords: Content Based Filtering, Collaborative Filtering, Hybrid Filtering.

INTRODUCTION

The world we live in today is quite different and unique in many ways compared to our ancestors even a hundred years ago. The boredom caused by repeatedly performing tasks so as to move ahead on the hierarchy. These social hierarchies were built to dish out just enough satisfaction and resources to keep the individual wanting for more. When a student comes back from college or an employee returns back from work, the only thing they look forward to is Escapism. As the word suggests, Escapism is the urge to distract the mind from the realities and modern trauma [1].

As the world has evolved, Escapism too has gotten better. During the old ages plays or skits would come to town and attract villagers with their fancy costumes and dresses. Over the ages this whole space has evolved with plays becoming a staple form of entertainment, over the last hundred odd years plays and stories have used technology to adapt themselves and reach out to a wider audience. The movies we see today are nothing more than an evolved form of entertainment of what the Middle Ages had. That being said, Movies are an excellent form of Escapism. With newer advancements in digital and media technologies, A good three-hour long movie has enough entertainment to keep you occupied and make you forget about the life you have to return to once the end credits roll. The emphasis here is on a good movie, and this differs from person to person. Some might like action; some may prefer dramas or romance. Ultimately everyone has a unique taste and catering to everyone's unique taste is exactly the goal of many in the movie and media business. So here lies the problem statement, how do you cater to a demand for the appropriate movie for every user, without making the whole system too personalized or focused for the said user. This is our main thought behind this project. To effectively build a system which learns to understand the users, their liking and disliking's, their no-go zones

LITERATURE SURVEY

Recommendation systems is a subclass of information filtering systems which can predict ratings and/or preferences a user could give to an item. Recommender systems are of various types, among them today we mainly employ three or four systems. The main classifications of recommender systems are Content filtering, Collaborative filtering, Knowledge based and Hybrid approaches.
Content filtering is a method used for building recommendation systems. It recommends items on the basis of description of an item and user's interest [2]. Content filtering is a very popular and useful filtering method which provides recommendations to a large set of users. It is used widely by large companies in their recommendation system workflows.
Collaborative filtering is the process of filtering and evaluating items based on opinions of other people [3]. This system is reliant on a set of users who rank and list items based on their likings. The collaborative process then tries to understand the growing positive or negative trends in this set of data and tries to recommend an item deems is the most similar.

Hybrid approach is the process of combining two or more types of recommendation strategies to benefit their complementary advantages [4]. As Hybrid approach is a mix of two or more recommendation strategies, the system built will be entirely unique to the demands of that particular problem statement.

The recommendation systems we use today are a refined and modern approach to the systems built earlier in the 90's. With the advent of latest technologies such as faster storage and more computing, computer scientists were able to realize many concepts. The first ideas of recommender systems were built on the ideas of classification and hence content-based filtering came to be. Towards the late 90's researchers developed an alternative to content-based filtering which was more user centric and was able to classify users. Advancements in factoring algorithms lead to the creation of Matrix Factorization methods which cut down on redundancy and made the whole process a lot more efficient [5]. Today the game has shifted towards personalization and the whole approach to recommendation systems are pivoted to cater to this personalization demand. Hybrid approaches have emerged as a clear winner, Hybrid approaches allow us to access the best of two approaches, this allows for more user personalization [4]. Hybrid approaches are still an evolving field and research is still being done to optimize various metrics, the most recent significant research in this field is the development of Context Aware Recommendation Systems [6].

EXISTING SYSTEM

Recommendations are an essential tool today to reach out to a larger audience and carter to ever growing personalized needs of every user. The type of recommendation system might change in accordance with the requirements at that given time.

Content Based Recommendation Systems: Content based recommendation system is a system which identifies and item features to recommend other items consisting of similar features based on the likes, dislikes of a user [2]. These systems use various techniques to analyze the features of the items, such as keywords, metadata, and user-generated content, to create a profile of each item. The system first builds a profile of the user's preferences based on their interactions with items. This profile is then compared to the content features of all available items in the system, and the items that most closely match the user's profile are recommended. It also incorporates specific feedback provided by the user. Usually, recommendation systems take time to gauge user's inclinations, this is called the Cold Start problem and this can be resolved by using content-based recommendation systems. The accuracy of these systems depends on the quality of the content-based similarity measure used to match users with items.

Content filtering utilized a Latent Dirichlet Allocation (LDA) algorithm. In the context of content-based filtering, LDA can be used to represent items as mixtures of latent topics and to compute similarity between items based on their topic distributions [7].

Collaborative Recommendation Systems: In Collaborative recommendation systems the recommendation is done by the behavior of the user and behavior of similar users [3]. The recommendation to a large extent is based on the similarity of one user to another. These systems analyze the behavior of many different users and attempt to find patterns or similarities between them. Once these patterns are identified, the system can make recommendations based on the behavior of users with similar patterns. In user-based systems, the system finds users with similar behavior to the target user and recommends items that those similar users have liked. As discussed above it's not just about a singular user's inclinations, but rather the whole set of users, and the accuracy and potency of the recommendations are based on the size of the pre-existing user base as well as effectiveness of data collection, data extraction techniques. Some inherent advantages that collaborative recommendation systems have are, breaking down the simplistic notion of a Black and White user, and realizing the true diversity users and their choices possess. Higher degree of personalization compared to other systems.

Collaborative filtering uses an algorithm known as the Alternating Least Squares (ALS). It is a Matrix factorization technique commonly used in collaborative filtering. The basic idea behind ALS is to represent items as vectors in a latent feature space, where the dot product of a user vector and an item vector represents the predicted rating for that user-item pair. [8].

PROPOSED SYSTEM

The proposed system is known as the Hybrid Recommendation System, this approach is called hybrid as it involves both the content based and collaborative based filtering methods combined into one [4]. The algorithm we use includes a mix of both content and collaborative filtering methods to help achieve hybrid filtering. The initial movie list is generated by a content-based recommendation approach.

Content Based Recommendation: It is the type of filtering which uses suggests movies based on similarity as compared with other movies, this works by identifying item features and comparing them [2]. The movie which has been given as input by the user is analyzed and the description of the movies are requested. The description of the movie is broken down to remove any grammar or connecting words which might impact the accuracy of the final output. These specified descriptive words are then used to build a similarity table which consist of similarity values of each iterable as compared to the original input query, thereafter the movies are ranked based on their similarity scores from highest to lowest. This is the first half of the process.

Collaborative Filtering: Collaborative filtering works on the principle of, if user A and B like a few common movies, then there is high chance that a movie liked by user B would also be liked by user A [3]. In our project the second half of the process consists of taking the same movies list generated by the first half of the process and input them to a collaborative filtering system to further personalize the output. In the collaborative filtering system, the movie list is then compared with a pre-existing user rating database of movies and using a threshold value, movies are filtered out. If there are no existing data to work with, i.e., a new user, then the system works by using the initial movie list generated by the content-based filtering system as the output.

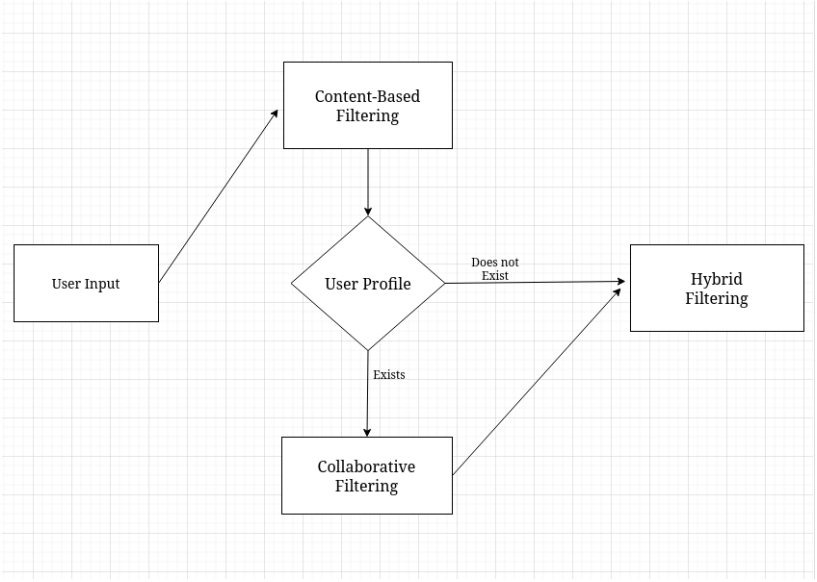

Fig 1. Flowchart of proposed system

The final output of the algorithm is five movies which are an accurate recommendation of the movie input by the user. These recommendations are also personalized to some extent which means the history of users' movie likings are also weighed in before making the final recommendation. Furthermore, explicit requests by users are also taken into consideration and the output reflects this. To set a benchmark for our recommendations we have also implemented a content-based recommender system which gives results based on the content filtering process. This is done to set a benchmark for our hybrid recommendations.

SYSTEM DESIGN

Our current system consists of various parts which perform specific functions, we can broadly classify them as the Frontend, Backend and Database. The Frontend consists of all the visual and peripheral parts of the application, it is also tasked with collecting and displaying the information to the user. The Backend consists of running the main algorithms which are responsible for generating the output for the given user input, the Backend is hosted as an API to reduce the bulkiness of the application. There are also databases which are responsible for storing data according to their pre-defined parameters. The database is further divided into 3 segments, movieMetadata, movieRatings, linksSmall. These are called tables to simplify the process, each table has predefined parameters for the data to be entered. movieMetadata consist of all the auxiliary information related to a particular movie, the movieRatings table consist of ratings for a particular movieId, mainly the linksSmall consist of tmdbID which is useful for certain processes of the algorithm. The workflow of the application is as follows: As the user opens the app, the user is prompted with a Login/Signup page. Once the login and authentication is over, the user can view a large variety of movies or search for the movie he or she prefers. The search request of the user is parsed by the frontend and the word sequence string is extracted. This information along with other crucial data is then relayed to the API using a POST request, the function of the API consists of two parameters ‘userId’ and ‘movieId’. The API accesses the Database and pulls the required information and generates a list of outputs.

This is the final list of recommendations. All this while the Frontend is listening for a response for its POST request and when it receives the said list, it begins to parse through the information and pick them out one at a time. The Frontend also makes calls to the Database to get the relevant information regarding the movie description, movie poster etc. All this is then readied to be displayed to the user as the final output of recommendation to the given input. This whole sequence of actions takes place in under a second, right from user request to final outputs. Technologies involved play a key role in this. Frontend is built using Flutter, as it is lightweight and less cumbersome. Backend is written in python and hosted using OracleCloud. The database is hosted on MongoDb for easier integration with the python script.

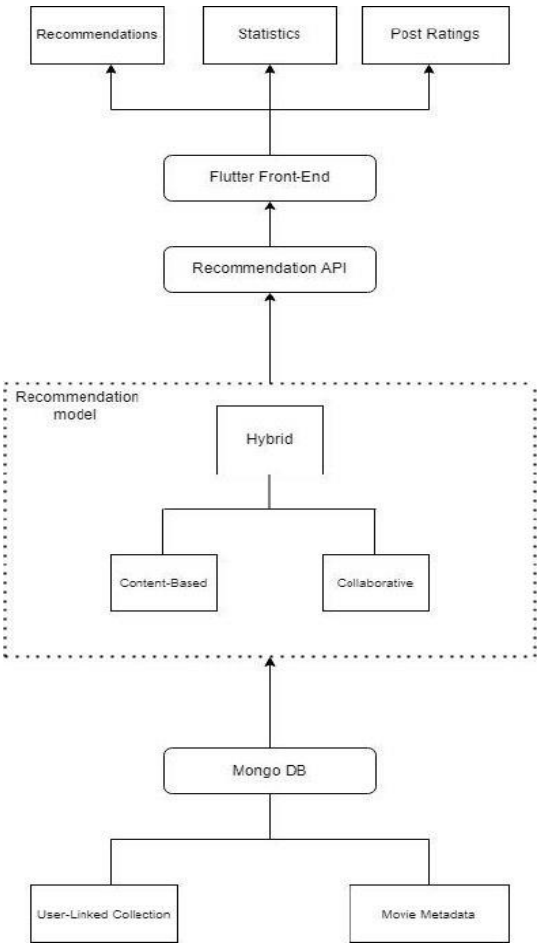

Fig 2 Detailed System Architecture

ALGORITHMS EMPLOYED

The algorithms used in the project are indicative of our needs and usability. As the Hybrid system we employed is a convolved mix of content and collaborative approaches, the algorithms used in it also are different. The two main algorithms used are.

Cosine Similarity: Cosine similarity is nothing but the similarity of two vectors, it is calculated by measuring the cosine angle between the two vectors [9], indicating the overlap or similarity of the two vectors. Ranges for similarity range from -1 to +1. With -1 indicating perfect dissimilarity and +1 indicating perfect similarity. First the frequency of each term in the two documents is determined. Then, the dot product of the two vectors is calculated, which is the sum of the products of the corresponding elements in the vectors. Next, the magnitude of each vector is calculated, which is the square root of the sum of the squares of the elements in the vector. Finally, the cosine similarity score is determined by dividing the dot product of the two vectors by the product of their magnitudes. In our project we have employed a cosine similarity algorithm which compares the descriptive words of two movies to generate a similarity score.

Singular Value Decomposition: Singular Value decomposition is mainly a matrix factorization technique that influences user item interactions [10]. At its core, SVD is a matrix factorization method that decomposes a matrix into three components: the left singular vectors, the right singular vectors, and the singular values. The singular values represent the magnitude of the importance of each singular vector in the original matrix, while the left and right singular vectors represent the basis vectors for the row and column spaces of the matrix, respectively. One key advantage of SVD is its ability to find an underlying structure within a dataset, making it particularly useful for uncovering patterns in large, high-dimensional datasets. It is best suited while using working with large datasets. We used Singular Value Decomposition as the main algorithm for implementing collaborative filtering. The mathematical formula of SVD is [10].

$$R = U * S * V^T$$

CONCLUSION AND FUTUTURE SCOPE

The conclusion and main takeaways of this project are the efficiency and usability of recommender systems. Especially the hybrid approach, to build a more personalized and accurate result. In conclusion, recommendations and the systems that power these recommendations are highly useful and needed in modern tech and media businesses. On the matter of future scope of such a project, there can be many additions and improvements. Splitting the future scope of our project into long- and short-term outlooks can help us prioritize our needs and resources efficiently. In the short term we could implement easy-to-rollout features which help increase user engagement, such as Data Visualization, History of Movies, Live Links on how to access the movies. In the long term we could look towards expanding the app to newer content and genres, introduction of podcasts and other media, and not explicitly restricted to movies. We could also work on bigger features for the app such as a community space for users to chat and discuss movies and help facilitate building of user led communities. These would require significant investment of time, resources and technical know-how. But all of the above expansions would be based on the core fundamentals of the app which is a user centric recommendation system.

ACKNOWLEDGMENT

A special thanks to Mrs Kusumalatha Karre for helping us in executing our ideas and guiding us throughout the way. Also, thanks to the entire ECM Department.

REFERENCES

1. Anxiety and Escapism, "https://www.counselling-directory.org.uk/memberarticles/anxiety-and-escapism-a-post-traumatic-stress-disorder"
2. Charilaos Zisopoulos, Savvas Karagiannidis, Georgios Demirtsoglou, Stefanos Antaris, "Content-Based Recommendation Systems" https://www.researchgate.net/publication/236895069_Content-Based_Recommendation_Systems
3. Ben Schafer, Dan Frankowski, Ben J, "Collaborative Filtering Recommender Systems" https://www.researchgate.net/publication/200121027_Collaborative_Filtering_Recommender_Systems
4. Erion Çano, Maurizio Morisio, "Hybrid Recommender Systems: A Systematic Literature Review" https://www.researchgate.net/publication/321281871_Hybrid_Recommender_Systems_A_Systematic_Literature_Review
5. Yehuda Koren, Robert Bell, Chris Volinsky, "Matrix Factorization Techniques for Recommender Systems" https://ieeexplore.ieee.org/document/5197422
6. Amit Livne, Eliad Shem Tov, Adir Solomon, Achiya Elyasaf, Bracha Shapira, Lior Rokach, "Evolving Context-Aware Recommender Systems With Users in Mind" https://arxiv.org/abs/2007.15409
7. Simon Leijon, "Recommender System Using Online Latent Dirichlet Allocation And Wikipedia", https://www.diva-portal.org/smash/get/diva2:1676187/FULLTEXT02
8. Xiangnan He, Hanwang Zhang, Min-Yen Kan, Tat-Seng Chua, "Fast Matrix Factorization for Online Recommendation with Implicit Feedback", https://arxiv.org/pdf/1708.05024.pdf
9. Cosine Similarity, "https://www.sciencedirect.com/topics/computer-science/cosinesimilarity#:~:text=Cosine%20similarity%20measures%20the%20similarity,document%20similarity%20in%20text%20analysis".
10. Singular value decomposition, "https://en.wikipedia.org/wiki/Singular_value_decomposition#:~:text=In%20linear%20algebra%2C%20the%20singular,related%20to%20the%20polar%20decomposition."