International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 03 | Mar 2021 www.irjet.net p-ISSN: 2395-0072

Content Recommendation System

Ayush Mishra[1], Malhaar Waghela[2], Madhavi Gavade[3], Asst Prof. Odilia Gonsalves[4]

[1,2,3]Student, Department of Information and Technology, Atharva College of Engineering, Mumbai, Maharashtra, India.

[4]Assistant Professor, Department of Information and Technology, Atharva College of Engineering, Mumbai, Maharashtra, India.

---***---

Abstract – *As we know that the main use of content recommendation system is to recommend movies and web series on the basis of the search that users made into the search box. The information is taken from the input and that is in the form of browsing data. This information reflects the prior usage of the product as well as the assigned rating. A recommendation system is a platform that provides its users with various contents based on their preferences and likes. It is basically implemented by using machine learning algorithms like KNN, Collaborative filtering and Content based similarity matrix.*

In COVID-19 PANDEMIC, OTT platforms are getting more attention from the public. Since the impact on theatres & cinemas is getting graver by the time. Creating a more peaceful and natural environment at home OTT platforms gives numerous enjoyable viewing content to the consumers. Telecommunications industries in India are trying to provide the bundle services of various OTT platforms like Netflix, Amazon Prime, Hotstar, Zee, etc. with more options choices become harder so creating a recommendation system is essential however Giants like Netflix & Amazon already have the well-devised system. However, Local OTT Platforms like Zee, Alt Balaji, Sony, etc. Here we are testing two model i.e Content-based Filtering & Collaborative Filtering. To evaluate our approach, we have utilised a cluster of data from Global & Local OTT Platforms to check which provides the better-recommed list.

***Key Words*: Sentimental Analysis, Recommendation, Collaborative Filtering, Content Based Similarity Matrix**

1. INTRODUCTION

As we know that from the last few decades, with the rise of some online platforms like Amazon, YouTube, Netflix, and many other such web services, recommender systems gain more and more place in our lives. From e-commerce to online advertisements recommender systems are used today and are unavoidable in our daily online journeys. Recommendation systems are very important in some of the industries as they can generate a huge amount of income if they are efficient or also be a way to stand out significantly from competitors. If we talk about the proof of how significance is recommender systems actually, then we can mention that a few years ago, Netflix organized challenges ("Netflix Prize") where the goal was to produce a recommender system that performs better than its algorithm with a prize of 1 million dollars to win. So It is very much clear that the impact of the recommender system is very large in all these fields.

The recommendation system is a topic which is growing and changing with a very rapid face according to the person's choice and likes. Here we will focus on the movie recommendation system, so we know how much we love watching movies and TV series. So after completing one movie or TV series the person might want to watch movies of the same actors or genres so for assuring this feature recommendation system comes into the picture. And because of this system customers gets satisfaction and subscription Provider like Netflix will also get profit, which leads to a win-win situation for both users and company While using large datasets with numerous attributes to distinguish the movies and TV Shows with many aspects like genres, plot, actors, region, cultural background, content provider, ratings, reviews, etc. While working with many aspects the KNN algorithm doesn't work well with high dimensional data because use with a large number of dimensions, it becomes difficult for the algorithm to calculate the distance in each dimension. KNN is sensitive to noise in the dataset. We intend to extend the entertainment industry to new heights where the viewers can choose the content according to his/her preferences without any barrier like content provider, language or availability, etc.

Problem Statement

Almost every online streaming sites like Netflix, Amazon contains recommendation system as per cast, genre, etc. but we are not satisfied sometimes because of some reasons because of companies like Netflix, Amazon show recommendation of only those movies which is present on their database. For example, we can see Avengers infinity war on Netflix but we won't get the recommendation of Avengers Endgame and Captain America but it doesn't mean that these movies are not there in Marvel Universe and the internet

To create a system that can recommend movies that users want to watch. The purpose of a recommendation system is to search for content that would be interesting to an individual. Moreover, it involves several factors to create personalized lists of useful and interesting content specific to each user/individual. It depends on what other people with similar traits/demographics are watching, and how likely you to watch those movies.

2. LITERATURE REVIEW

In this paper, authors Martin Szomszor and Harith Alanil have demonstrated that a movie recommendation system can be built purely on collaborative tagging. By using different types of tag-clouds that express a user's degree of interest, a prediction for a previously unrated movie can be made based on the similarity of its keywords. We have faith that our movie recommendation algorithms which we have used can be improved by combining them with more traditional content-based recommender strategies. Since, as we know the IMDb given extensive information about the actors, directors, and writers of movies, as well as demographic breakdowns of the ratings, a more detailed profile, can be constructed for each user. Also, our recommendation algorithms have not exploited any collaborative recommender techniques. Further research may show that rating tag clouds are a useful and more efficient way to find neighbors with similar tastes.

In this paper, Zan Wang and other co-authors have mentioned that the sparse of user-item rating matrix makes it hard to find real neighbors to form the final recommendation list. In our experiments, we compare the performances and some trends f the existing baseline CF movie recommendation systems with our approach, while the neighborhood size varies from 5-60 in an increment of 5. In the sparse data environment, the selection of "like-minded" neighborhoods based on common ratings is a vital action to generate high-quality movie recommendations. In our proposed approach, feature lection based on PCA was first performed on whole data space, and then the clusters were generated from relatively low dimension vector space transformed the first step. In this way, the original user space becomes much denser and reliable and used for neighborhood selection instead of searching in the w user space. Also, to result in the best neighborhood, we apply genetic algorithms to optimize the K-means process to cluster similar users. Based on the Movie lens dataset, the experimental evaluation of the proposed approach proved that it is capable of providing high prediction accuracy and more reliable movie recommendations for users' preferences compared to the listing lettering-based CFS.

As for the cold-start issue, the experiment also demonstrated that our proposed approach is capable of generating an effective estimation of movie ratings for new users via traditional movie recommendation systems.

In this paper, Neelima Sajja mentioned about the internal details and the query modalities of a movie recommender system. In particular, we have stressed the voting-based movie selection mechanism that uses stored user preferences for different movie dimensions and options for each dimension. The voting scheme used provides desirable guarantees about the nature of recommendation produced and is also robust to minor variation in specified preferences. There are several ways in which our movie recommendation system can be enhanced. Features that we are planning to add are the following:

Instance-based querying: They have designed a combination of instance-based and voting schemes by which the user can ask for a recommendation similar to a movie that he/she may have liked in the past. To provide this function, we have to store liked and disliked movies by the users in the past. Learning: We are evaluating the possibility of incorporating other learning schemes by which the system can effectively update the stored user preferences based on recommendations that were liked or disliked by the user. Explanation facility: We plan to implement an explanation facility by which the user can be given more details as to why a certain movie was recommended based on his/her preferences.

In this paper, Author Shinhyun Ahn and co-author Chung Kon-Shi explain how they utilized 5 types of cultural metadata about movies for a movie recommendation and groped the potential of cultural metadata. Then, User can be able to comments which represent a wide spectrum of implicit information, and genres metadata which are controlled and precise showed their high potential.
And they also explain about naive Bayes algorithm that we can use for the sentimental analysis part.

3. DESIGN METHODOLOGY

The Design Methodology of recommendation system is dependent on two methods which is **Content based** and **Collaborative filtering** method.

Content based

After calculating the weights of each feature for each user, we use them to predict the recommendation ratingsof the contents. We calculate the rating for each feature set separately using the weights obtained with the use of the training data. We use the following method to generate ratings:rk(u, i) = ∑j∈Fk,jwk(u, j).(5)In this equation, rk(u, i) represents the rating that user ugives movie iaccording to the feature set k.Fk,jare the features under feature set kthat are related to movie i.

Collaborative filtering

The basic idea behind collaborative filtering is that when we need to predict the rating for a certain user and a certain movie, even if the user has not seen the movie, we can check to see if other people with whom the user had similar ratings on past movies liked that movie or not and we can base the predictions on those neighbors' opinions. We formalize this basic idea as follows: Consider that we need to predict the rating of the user for movie i. Let Me be the set of users who have watched the movie within the training period. If Di=∅, then we can not produce a collaborative rating. Let Di,u ⊆Dibe the set of users who have also watched at least one common movie with you within the training period. Then we compute the collaborative rating of the user for the movie i.e. :

rc(u, i) = 1 C×∑ v∈Di,u c(u, v)×r(v, i).(In this equation, c(u, v)≥1 is the number of movies that both user and user have watched within the training period. r(v, i) is the implicit rating of the user for movie i. Cis a normalization constant, defined as C=∑ v∈Di,u c(u, v).

3.1 Implicit Rating Techniques

If there is explicit types of ratings (discrete rating) are available, then a recommendation system may use these ratings. On the other hand, for many systems, users do not want to provide such explicit feedback, and therefore implicit ratings need to be produced based on the user's past behavior. Most recommendation systems rely on explicit user feedback. In our system, users do not rate movies explicitly, so we calculate implicit ratings by using the viewing durations. Assume that the user watches the movie for t(u, i) minutes during the training period and times the total duration of the movie i. We define the normalized viewing duration (the implicit rating) of user for movie is:r(u, i) = t(u, i)ti.(1)This formulation is similar to that of. The difference in our formulation is that it can be greater than 1 if a user has watched content more than once.

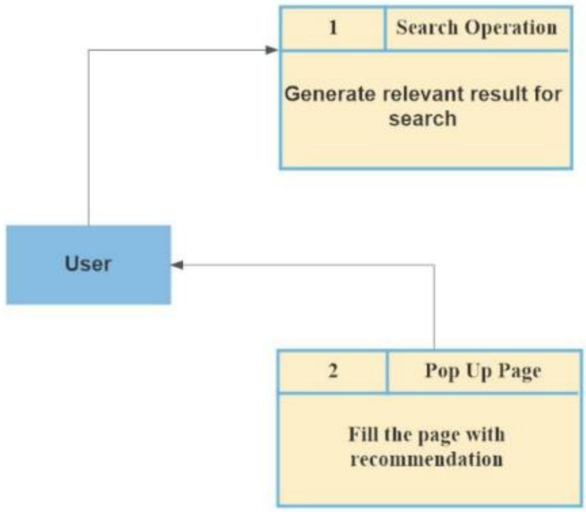

Fig 1: System Flowchart

3.2 Content Features

The movie actor, director, and genre features were used directly. The release year information was discretized into six different values as before 1980, the 1980s, 1990s, 2000s,2010–2011, and 2012. The movie synopsis documents will be used for getting the keywords of a web series and movies. First of all, the synopsis documents were processed using the Zemberek software. Zemberek is an open-source, platform-independent natural language processing software for Turkish and other Turkic languages. In the Turkish language, new words are formed by concatenating suffixes on top of the root. Zemberek finds the root(s) that can be the actual root of an observed word. After we obtain the roots of the words in the summary document, we need to use them to represent a movie. To extract the keyword features for movies, similar to, we use the tf-idf weights. The tf-idf weight keyword (term)jin movie is computed as:tf −idf (j, i) = tf (j, i)×log(N/nj).

4. RESULT OF IMPLEMENTATION

The system was built and implemented on Python 3.6.4. The testing was done on dynamic images for real-time movies recommendation. Application of Artificial Intelligence that is Sentimental Analysis is being used in our system. Following are the snapshots of the working system recommending movies.

4.1 Home Page

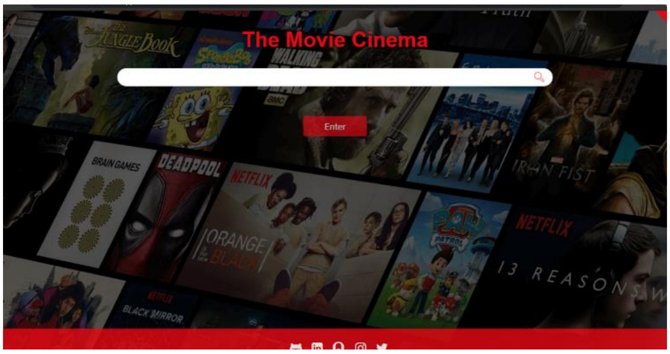

Fig 2: Home Page

4.2 Searched Movies

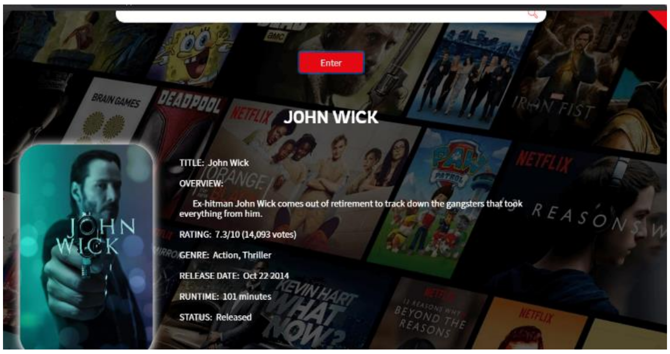

Fig 3: Searched Movies

4.3 Recommended Movies

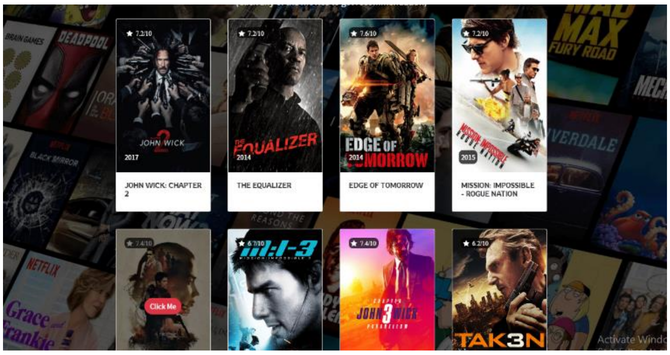

Fig 4: Recommended Movies

4.4 GUI of the Project

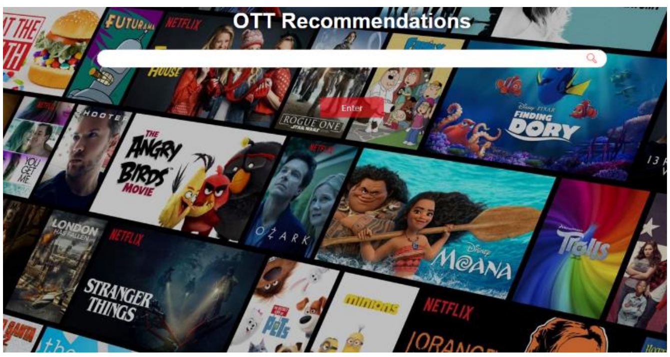

Fig 5: GUI

CONCLUSION

Our proposed system is effective in providing much more accurate and fast results. The system proves to be time-saving and requires less maintenance. This system is useful and easy to use and have very attractive user-interface so many users will surely give try to this. Many users like to watch regional languages movies, So they can watch that by getting a recommendation system.

ACKNOWLEDGEMENT

We express our deep gratitude to **Asst. Prof. Odilia Gonsalves**, our project guide for her exemplary guidance, monitoring and constant encouragement to push us beyond our limits throughout the course of this project. We would like to thank her for providing idealistic solutions and give fresh dimensions to the completion of this project. Regular virtual meetings and doubt solving sessions made it easy for us. We are overwhelmed to offer our gratitude to Head of the Information Technology department **Prof. Deepali Maste** for her moral support which motivated us throughout the hardships to complete the project on time in this pandemic. Special thanks to the principle of Atharva College of Engineering **Prof. Shrikant Kallurkar** and the entire administrative staff to provide us with well-furnished libraries along with necessary reference papers. Well- equipped IT labs and other infrastructure provided us with a perfect environment to deliver the best of us.

REFERENCES

[1] Choi, S.-M., Han, Y.-S.: A content recommendation system based on category correlations. The Fifth International Multi-Conference on Computing in the Global Information Technology, pp. 1257–1260 (2010)

[2] Huang Z, Chen, Zeng, “Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering,” ACM TRANS, 2004.

[3] R. D. Burke. Hybrid web recommender systems. In P. Brusilovsky, A. Kobsa, The Adaptive Web, Methods and Strategies of Web Personalization, Springer, 2007.

[4] Y. Koren. Factorization meets the neighbourhood: a multifaceted collaborative filtering model. 2008. ACM.

[5] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 2003.

[6] P. Melville, R. J. Mooney, and R. Nagarajan. Content boosted collaborative filtering for improved recommendations. 2002.ACM

[7] M. J. Pazzani and D. Bills. Content-based recommendation systems. May 2007.Springer

[8] Ningning Yi, Chunfang Li, Xin Fang: Design and Implementation of Movie Recommender System Based on Graph Database. IEEE.2017

[9] Dharmendra Pathak, Sandeep Matharia: Hybrid Recommendation Engine. IEEE(2013).

[10] Ramin Ebrahim Nakhil, Hadi Moradi: Movie Recommendation Systems based on percentage of views. IEEE(2019).