International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Video Recommender System

Vishwas Kant[1], Govind Sharma[2], Rishabh Verma[3], Manik Wazir[4], Ranjeeta Krishan[5]

[1]*Student, Department of Computer Science & Engineering, Bhilai Institute of Technology Raipur, Chhattisgarh, India*
[2]*Student, Department of Computer Science & Engineering, Bhilai Institute of Technology Raipur, Chhattisgarh, India*
[3]*Student, Department of Computer Science & Engineering, Bhilai Institute of Technology Raipur, Chhattisgarh, India*
[4]*Student, Department of Computer Science & Engineering, Bhilai Institute of Technology Raipur, Chhattisgarh, India*
[5]*Faculty, Department of Computer Science & Engineering, Bhilai Institute of Technology Raipur, Chhattisgarh, India*

ABSTRACT

Recommender System is a tool helping users find content and overcome information overload. It predicts the interests of users and makes recommendations according to the interest model of users. For watching favorable videos online we can utilize video recommendation systems, which are more reliable since searching for preferred videos will require more and more time which one cannot afford to waste. In this paper, to improve the quality of a video recommendation system, a Hybrid approach is by combining content-based filtering, watching time filtering, and location-based filtering using a single text-to-vector conversion technique and a single technique to find the similarity between the vectors. The hybrid approach helps to get the advantages of both approaches as well as tries to eliminate the drawbacks of both methods.

Keywords: *- Video Recommendations, Text to vector, Vector similarity, Video Watching Percentage filtering, Location-based filtering, Hybrid approach.*

1. INTRODUCTION

Recommender systems have proven to be a successful technique for dealing with the ever-increasing volume of internet information. So, the job of a recommender system is to suggest the most relevant items to the user. Recommender systems are used for providing personalized recommendations based on the user profile and previous behavior. Recommendation systems are used on YouTube for video recommendations, Amazon and Flipkart for product recommendations, Netflix and Amazon Prime for movie recommendations, and so on. Whatever you do on such websites, there is a system that sees your behavior and then ultimately suggests things with which you are highly likely to engage. This research paper deals with video recommendations and the logic behind video recommendation systems, traditional video recommendation systems, issues related to traditional video recommendation systems, and a proposed solution for personalized video recommendation systems. A lot of famous video recommendation-related datasets are already available on Kaggle and other websites. Websites like Netflix, Amazon Prime, Instagram, YouTube, etc. use video recommendations to increase their revenue or profits by ultimately improving the user experience. To filter the data, we need some filtering techniques. There are different types of filtering techniques or video recommendation algorithms upon which a recommendation system can be based upon.

Major filtering techniques or video recommendation algorithms are as follows:

1. Content-Based Filtering

2. Watching Percentage-Based Filtering.

3. User Location-Based Filtering

2. LITERATURE REVIEW

V. Subramaniyaswamy, et. al. [1] have proposed a solution for personalized movie recommendations which uses collaborative filtering techniques. Euclidean distance metric has been used to find the most similar user. The user with the least value of Euclidean distance is found. Finally, movie recommendation is based on what that particular user has best rated. The authors have even claimed that the recommendations are varied as per the time so that the system performs better with the changing taste of the user with time.

D.K. Yadav, et. al. [2] have proposed a personalized movie recommendation solution using a collaborative filtering approach. Collaborative filtering makes use of information provided by the user. That information is analyzed and a movie is recommended to the users which are arranged with the movie with the highest rating first. The system also has a provision for the user to select attributes on which he wants the movie to be recommended. Luis M Capos et al.

Gaurav Arora, et. al. [3] have proposed a movie recommendation solution based on users' similarities. The research paper is very general in the sense that the authors have not mentioned the internal working details. In the Methodology section, the authors have mentioned City Block Distance and Euclidean Distance but have not mentioned anything about cosine similarity or other techniques. The authors stated that the recommendation system is based on a hybrid approach using context-based filtering and collaborative filtering but neither they have stated the parameters used, not they have stated about the internal working details.

Costin-Gabriel Chiru, et. al. [4] proposed Movie Recommender, a system that uses the information known about the user to provide movie recommendations. This system attempts to solve the problem of unique recommendations which results from ignoring the data specific to the user. The psychological profile of the user, their watching history, and the data involving movie scores from other websites are collected. They are based on aggregate similarity calculation. The system is a hybrid model which uses both content-based filtering and collaborative filtering.

According to R. Lavanya, et. al. [5], in order to tackle the information explosion problem, recommendation systems are helpful. The authors mentioned the problems of data sparsity, cold start problems, scalability, etc. The authors have done a literature review of nearly 15 research papers related to the movie recommendation system. After reviewing all these papers, they observed that most of the authors have used collaborative filtering rather than content-based filtering. Also, the authors noticed that a lot of authors have used a hybrid-based approach. Even though a lot of research has been done on recommendation systems, there is always a scope for doing more in order to solve the existing drawbacks.

Sang-Min Choi, et. al. [6] mentioned the shortcomings of collaborative filtering approaches like the sparsity problem or the cold-start problem. To avoid this issue, the authors have proposed a solution to use category information. The authors have proposed a movie recommendation system that is based on genre correlations. The authors stated that the category information is present for the newly created content. Thus, even if the new content does not have enough ratings or enough views, still it can pop up in the recommendations list with the help of category or genre information. The proposed solution is unbiased over the highly rated most watched content and new content which is not watched a lot. Hence, even a new movie can be recommended by the recommendation system.

Urszula Kuzelewska et. al. [7] proposed clustering as a way to deal with recommender systems. Two methods of computing cluster representatives were presented and evaluated. Centroid-based solution and memory-based collaborative filtering methods were used as a basis for comparing the effectiveness of the proposed two methods. The result was a significant increase in the accuracy of the generated recommendations when compared to just the centroid-based method.

Md. Akter Hossain, et. al. [8] proposed NERS which is an acronym for neural engine-based recommender system. The authors have done a successful interaction between the 2 datasets carefully. Moreover, the authors stated that the results of their system are better than the existing systems because they have incorporated the usage of a general dataset as well as a behavior-based dataset in their system. The authors have used 3 different estimators to evaluate their system against the existing systems.

Debashis Das, et. al. [9] wrote about the different types of recommendation systems and their general information. This was a survey paper on recommendation systems. The authors mentioned personalized recommendation systems as well as non-personalized systems. User-based collaborative filtering and item-based collaborative filtering were explained with a very good example. The authors have also mentioned the merits and demerits of different recommendation systems.

Harpreet Kaur et. al. [10] have proposed a solution for personalized movie recommendations in which the system uses a mixture of content and a collaborative filtering algorithm. While recommending both the context of the movies, the user-user relationship and also the user-item relationship are considered important in the recommendation system.

3. PROPOSED METHODOLOGY

We need to perform preprocessing on the dataset and combine the relevant features into a single feature. Later, we need to convert the text from that particular feature into vectors. Later, we need to find the similarity between the vectors. Finally, get the recommendations as per the system architecture mentioned below.

Architecture

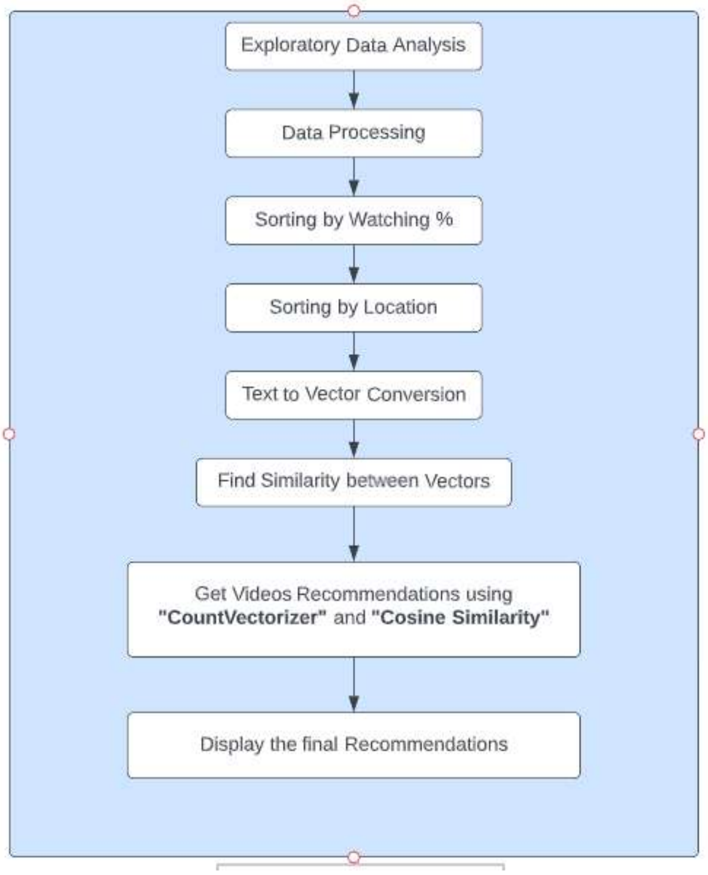

Fig. 1. System Architecture

Dataset

The ‘Dataset’ is taken into consideration for movie recommendation purposes in this research work. The dataset is composed of 2 CSV files - ‘content.csv’ and ‘relation.csv’.

The ‘content.csv’ dataset is nothing but it is all about content information, consisting of the following attributes:

- ‘content_type’: It indicates whether the content is series, sports, or movies.
- ‘language’: It indicates the languages of content.
- ‘genre’: It indicates the genres of the movie like Action, Documentary, etc.
- ‘duration’: It indicates the duration of the content.
- ‘release_date’: It indicates the release date of the content.
- ‘rating’: It indicates the rating of video content from 1-10.
- ‘episode_count’: It is nothing but the number of episodes of content related to the series if the content is not related to the series then the episode number will be zero.
- ‘season_count’: It is nothing but the number of seasons of content related to the series if the content is not related to the series then the season number will be zero.
- ‘content_id’: It indicates content ID.

1	content_id	content_type	language	genre	duration	release_date	rating	episode_count	season_count
2	cont_475_19_32	series	english	drama	4980000	01-07-2018	10	32	19
3	cont_2185_15_21	series	english	drama	3000000	29-03-2016	4	21	15
4	cont_4857_13_28	series	tamil	comedy	3120000	06-03-2006	8	28	13
5	cont_3340_1_5	sports	hindi	cricket	9900000	10-01-2009	0	5	1
6	cont_1664_10_29	series	hindi	action	3660000	25-05-2020	2	29	10
7	cont_51_1_37	series	hindi	comedy	3060000	04-02-2002	10	37	1
8	cont_2208_1_24	series	marathi	drama	3600000	09-07-2011	7	24	1
9	cont_2679_5_17	series	hindi	comedy	3240000	03-05-1994	7	17	5
10	cont_4790_4_19	series	marathi	drama	3060000	18-12-2020	1	19	4
11	cont_1437_5_25	series	hindi	comedy	2880000	27-05-2002	1	25	5
12	cont_3065_11_20	series	malayalam	drama	4320000	15-06-2004	5	20	11
13	cont_3053_5_1	series	hindi	drama	3300000	14-11-2017	5	1	5
14	cont_2478_11_9	series	tamil	drama	5100000	22-03-2017	8	9	11
15	cont_673_10_48	series	punjabi	comedy	3600000	24-06-2016	2	48	10
16	cont_3375_1_22	series	english	action	3180000	26-06-2001	5	22	1
17	cont_3383_9_6	series	tamil	horror	2940000	20-10-2011	2	6	9
18	cont_1690_1_6	sports	hindi	cricket	4200000	10-12-2005	0	6	1
19	cont_3315_7_2	series	hindi	documentary	3420000	29-06-2018	9	2	7
20	cont_2853_5_22	series	english	drama	3840000	15-02-2013	3	22	5
21	cont_43_13_8	series	marathi	drama	2820000	08-05-2018	2	8	13
22	cont_3923_1_7	series	kannada	action	3720000	27-03-2016	1	7	1
23	cont_4084_3_1	series	english	drama	5220000	04-04-1997	8	1	3
24	cont_1937_6_52	series	hindi	action	3300000	25-07-2017	7	52	6

Fig. 2. Dataset "content.csv"

The 'relation.csv' dataset is nothing but it is all about user activities or history, consisting of the following attributes:

- 'user_id': It indicates the user ID of a particular user. We can relate this to the email ID of a user.
- 'content_id': It indicates content ID. Here the content ID is used to map the 'content.csv' dataset that holds the information of the content.
- 'date': It indicates the date. It is nothing but a date on which the user watches the particular content. It uses the "DD-MM-YYYY" format.
- 'durata': It indicates the runtime of content. Runtime means the actual length of the content in a millisecond.
- 'start_time': It indicates the start time of watching content.
- 'end_time': It indicates the end time of watching content.
- 'location': It indicates the location of the user. It contains the area pin code, city, district, and state.

	user_id	content_id	durata	date	start_time	end_time	location
0	user_13110@domain.com	cont_3943_2_23	273600	26-09-2020	17:44:02	18:32:02	203001 Bulandshahr_Kutchery Bulandshahr Uttar_...
1	user_18909@domain.com	cont_1981_6_18	352800	03-12-2020	17:24:17	17:51:17	670101 Thalassery_Bazar Kannur Kerala
2	user_14564@domain.com	cont_2227_9_21	327600	07-06-2020	17:48:51	17:54:51	274506 Payasi Deoria Uttar_Pradesh
3	user_15761@domain.com	cont_3943_8_34	342000	02-10-2020	09:42:32	10:44:32	736121 Panchkolguri Jalpaiguri West_Bengal
4	user_16705@domain.com	cont_1369_7_18	306000	24-07-2020	12:20:20	13:09:20	686652 Mary_Land Kottayam Kerala
...	...	...	...	...	...	...	...
13839	user_14842@domain.com	cont_2223_1_8	342000	28-10-2020	06:10:12	07:24:12	841206 Basahi Saran Bihar
13840	user_18909@domain.com	cont_748_10_2	334800	11-08-2021	19:28:39	19:52:39	670101 Thalassery_Bazar Kannur Kerala
13841	user_16705@domain.com	cont_4833_10_13	396000	27-06-2019	00:28:47	00:37:47	686652 Mary_Land Kottayam Kerala
13842	user_15761@domain.com	cont_3577_21_11	482400	14-01-2020	11:12:52	11:58:52	736121 Panchkolguri Jalpaiguri West_Bengal
13843	user_11184@domain.com	cont_3187_4_20	338400	30-08-2018	14:42:01	15:30:01	396060 Kanbhai_Bo Navsari Gujarat

13844 rows × 7 columns

Fig. 3. Database "realation.csv"

Preprocessing steps include removing stopwords and special characters, combining the content type, language, genre, duration, release date, rating, and location into a single string, removing punctuation marks, converting text into lowercase, etc.

	content_id	tag
0	cont_3943_2_23	series english drama 4980000 01-07-2018 10
1	cont_1981_6_18	series english drama 3000000 29-03-2016 4
2	cont_2227_9_21	series tamil comedy 3120000 06-03-2006 8
3	cont_3943_8_34	sports hindi cricket 9900000 10-01-2009 0
4	cont_1369_7_18	series hindi action 3660000 25-05-2020 2

Fig. 4. Properties of a video are combined into a single feature titled 'tag'

The 'tag' attribute needs to be further processed by using some algorithms.

Algorithm

Content-Based Filtering:

We can use CountVectorizer or TfidfVectorizer or Glove or Word2Vec to create vectors from the text. After converting the text into vectors, we need to find the similarity between the vectors. Cosine Similarity or sigmoid_kernel or some other technique can be used to find the similarity between the vectors.

Content-based Recommendation using CountVectorizer and Cosine Similarity, In this case, we will use CountVectorizer to create vectors from the preprocessed text mentioned in the 'tag' attribute. After getting the vectors, we will find the similarity between the vectors using Cosine Similarity.

After applying the algorithm data will sort for each user based on two factors:

1. Firstly we subtract the end_time column from the start_time column to get the actual watch time. After getting the actual watch time then we calculate the percentage of it concerning the actual length of a particular video and sort videos by their watching percentage in ascending order.
2. Secondly, we compare the locations of each user and then we sort the videos of each user based on their locations.

4. RESULT AND ANALYSIS

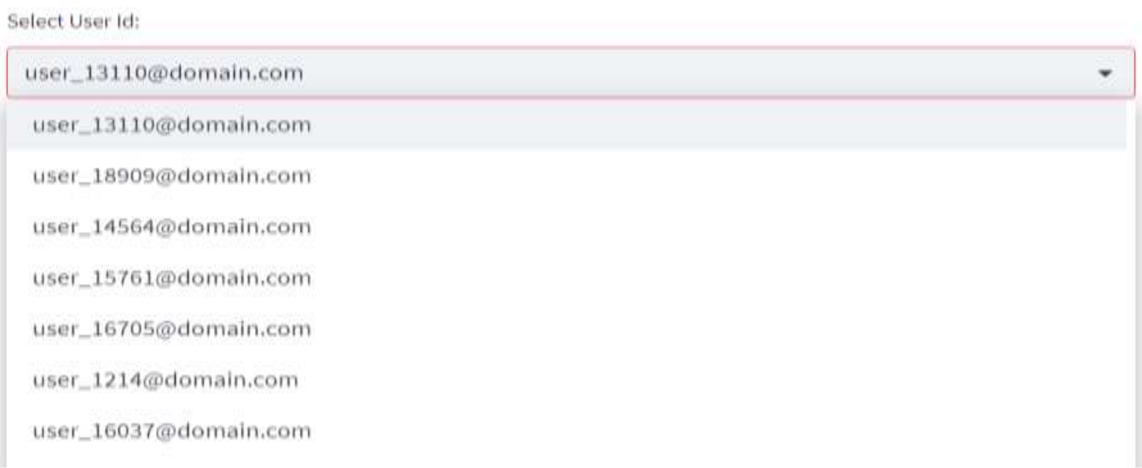

Fig. 5. Taking User ID as an Input

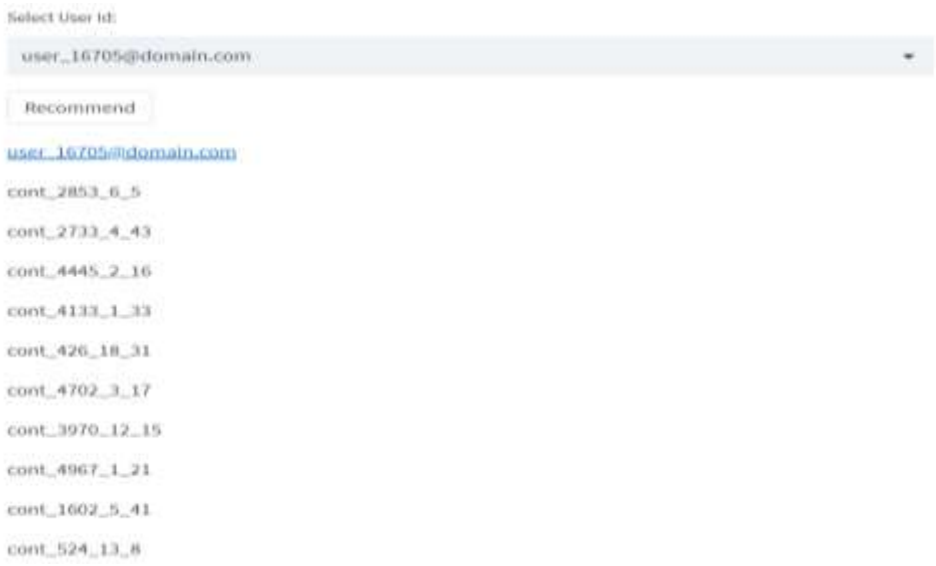

Fig. 6. Recommendations form of content ID

5. CONCLUSION

In this paper, we have introduced Video Recommender System. We can see from the results that the after sorting the dataset based on video-watching percentage and user location, the final recommendations are better than the individual recommendations of the algorithm mentioned in this research work. Hence, it is always better to manipulate the results of different algorithms to get the final result which has the advantages of the individual algorithms. Our approach can be further extended to other domains to recommend songs, venues, news, books, tourism, e-commerce sites, etc.

6. REFERENCES

[1]. Subramaniyaswamy, V., et al. "A personalized movie recommendation system based on collaborative filtering." International Journal of High-Performance Computing and Networking 10.1-2 (2017): 54-63.

[2]. Manoj Kumar, D.KYadav, Ankur Singh, Vijay Kr. Gupta," A Movie Recommender System: MOVREC" International Journal of Computer Applications (0975 – 8887) Volume 124 – No.3, August 2015.

[3]. Arora, Gaurav, et al. "Movie recommendation system based on users' similarity." International Journal of Computer Science and Mobile Computing 3.4 (2014): 765-770.

[4]. Costin-Gabriel Chiru, Vladimir-Nicolae Dinu, Ctlina Preda, Matei Macri; "Movie Recommender System Using the User's Psychological Profile" in IEEE International Conference on ICCP, 2015.

[5]. R. Lavanya, U. Singh, and V. Tyagi, "A Comprehensive Survey on Movie Recommendation Systems," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 532-536, DOI: 10.1109/ICAIS50930.2021.9395759.

[6]. Choi, Sang-Min, Sang-Ki Ko, and Yo-Sub Han. "A movie recommendation algorithm based on genre correlations." Expert Systems with Applications 39.9 (2012): 8079-8085.

[7]. Urszula Kużelewska; "Clustering Algorithms in Hybrid Recommender System on MovieLens Data", Studies in Logic, Grammar, and Rhetoric, 2014.

[8]. M. A. Hossain and M. N. Uddin, "A Neural Engine for Movie Recommendation System," 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (IEEE ICT), 2018, pp. 443-448, DOI: 10.1109/CEEICT.2018.8628128.

[9]. Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on the recommendation system." International Journal of Computer Applications 160.7 (2017).

[10]. Harper, F. Maxwell, and Joseph A. Konstan. "The movie lens datasets: History and context." Acm transactions on interactive intelligent systems (tiis) 5.4 (2015): 1-19.

[11]. Zhang, Jiang, et al. "Personalized real-time movie recommendation system: Practical prototype and evaluation." Tsinghua Science and Technology 25.2 (2019): 180-191.

[12]. Monika D.Rokade, Dr.Yogesh Kumar Sharma "MLIDS: A Machine Learning Approach for Intrusion Detection for Real-Time Network Dataset", 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), IEEE

[13]. N. Immaneni, I. Padmanaban, B. Ramasubramanian, and R. Sridhar, "A meta-level hybridization approach to personalized movie recommendation," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 2193-2200, DOI: 10.1109/ICACCI.2017.8126171