INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD
ISSN: 2455-0620 [Impact Factor: 6.719]
Monthly, Peer-Reviewed, Refereed, Indexed Journal with IC Value : 86.87
Volume - 8, Issue - 9, September - 2022 Publication Date: 10/09/2022

DOIs:10.2015/IJIRMF/202209002 --:-- Research Article

Hipster Movie Recommender System Based On Personalized Hybrid Filtering Algorithm

[1]M. Vidya Jaykumar, [2]Dr. Shubhangi Sapkal, [3]Veena Jayakumar Menoki

[1]Mtech Student, Department of Computer Science and Engineering, Government College of Engineering, Aurangabad, Maharashtra, India
[2]Associate Professor, Department of Computer Science and Engineering, Government College of Engineering, Aurangabad, Maharahtra
[3]Mtech Student, Department of Computer Science and Engineering, Government College of Engineering, Aurangabad, Maharashtra, India
Email - [1]vidya.menoki@gmail.com.com, [2]me18f05f007@geca.ac.in, [3]veenajayakumar.95@gmail.com

Abstract: *A Hipster recommender system proposed in this paper is a movie recommender system. The objective of our system is to recommend those movies that are mostly unseen but have very good reviews thus the name- Hipster. Unlike other mentioned recommender systems that uses collaborative filtering or content based filtering or a combination of them and some other algorithms, this paper creates movie recommendations based on a hybrid algorithm that uses certain features of content based algorithm and a personalized popularity based algorithm. The paper also makes comparisons of proposed system and existing system based on attributes such as accuracy and variations of using different metrics. Thus we create a specialized movie recommender system that recommends only great movies of the genre that the user prefers and that which remains widely unknown, like hidden gems.*

Key Words: *Movie, Recommendation, Hybrid, Popularity, Context-based.*

1. INTRODUCTION:

Online on-demand video streaming platforms like Netflix, Prime, Disney Hotstar are popular now-a-days. They not only have a vast database of movies, documentaries, TV shows, sitcoms etc. but also make use of recommender systems that suggests or recommends movies to a user that are most probable to be watched next by them. It always comes as a shock when you watch a good movie and discover that it is an old one. There are lots of movies that remain unknown despite them having great ratings. These maybe due to the fact that the movie wasn't a crowd-pleaser or have a popular cast or even due the sensitivity of the content not appealing to the people. The proposed system discovers such movies first based on the genre that the user suggests through input movie then compares the ratings and number of views of those movies to create top ten unpopular movies that have really good rating belonging to the same genre.

In order to make these personalized recommendations, the paper uses content based filtering algorithm- which models a training data to give movies similar to the given genre. This data can now be arranged as per their popularity by using the number of ratings they received and then only those movies having a rating of or above the threshold value is selected and recommended. The proposed system achieved to build a movie recommender that could deep dive into the pits of filmography and pull out some dusty diamond of a movie that would appease even the mustiest of critics.
There are several methods that are used to construct recommender systems including: collaborative filtering, content based, popularity based, knowledge-based, and hybrid recommender systems [1, 4].

In the collaborative filtering approach the user profile consists of users' feedback (e.g. ratings) and the neighborhood measure is used to provide recommendations [2, 3, 4]. This approach recommends items that are similar to the ones that the user or neighbors have preferred in the past [4]. These approaches use a similarity measure [5, 6] to recommend the items that are similar to the ones the user has previously selected [4].Commonly liked movies are generally considered as popular. Those movies that have a wide range of user ratings but have high number of number of voters can also be considered popular. Other attributes like plot, genre, cast, etc matters in the movie being popular and is also used to find a similar popular movie. Context aware recommender systems, employ contextual information such as time, location, and social data to make recommendations [7,4]. With the user requirements or resource specifications gathered in user profile, the knowledge based approaches use referencing/ case-based reasoning to find resources that fulfill the user needs [4, 8, 9]. The hybrid approaches combine the above mentioned approaches and use a rich user profile that consists of several components such as ratings, temporal/spatial, social info [4, 9, 10].

In this paper, a content-based algorithm is used along with a customized popularity based algorithm to develop a personalized movie recommender system. This is due to the content based filtering being too user- specific while in this paper, only certain user attributes are taken into account like the average rating of a movie, the number of votes for a movie as well as the movie genre and the title. The threshold used in calculating the constraint is found using the fact that ratings for 95 percentile of popular movies remain unchanged. So we take into account those movies that have been voted by at least 25 percentage of the average number of voters. Here the data set contains the 1000 movies from IMDB website. This table, in fact, could gather 250 observations (the movies) and 38 columns. However, we want the movie recommender to be based only on the genre, average ratings and number of votes, so these are the only columns we considered in the modeling. The plot column can also be taken into account but the recommender changes its recommendations if the movie genre and plot are both considered instead of just the movie genre or just the plot. Data set can also be taken from dependable source like MovieLens, IMDB etc. But since the proposed Hipster recommender system has specific requirements the data set was scrapped from the IMDB website itself.

2. EXISTING WORK:

In today's day and age, many of the big tech companies out there use a Recommender System for their sales and profit. It can be found anywhere from Amazon (product recommendations) to YouTube (video recommendations) to Facebook (friend recommendations). The ability to recommend relevant products or services to users can be a huge boost for a company, which is why it's so common to find this technique employed in so many sites. The pioneers in creating algorithms for recommendation systems and using them to serve their customers better in a personalized manner are:

a) Group Lens: Group Lens helped in developing initial recommender systems by pioneering collaborative filtering model. It also provided many data-sets to train models including Movie Lens and Book Lens that any researcher can use as a resource.

b) Amazon: Amazon implemented commercial recommender systems that majorly uses collaborative filtering or a hybrid of collaborative and content-based filtering. They also implemented a lot of computational improvements over its competitors.

c) Netflix: Netflix which is a popular steaming platform pioneered Latent Factor/ Matrix Factorization models in their recommendation systems.

d) Google-Youtube: Youtube uses a hybrid Recommendation System for recommending video contents. All their systems use Deep Learning based algorithms. They use a user's history to grab the data they need.

3. PROPOSED SYSTEM:

Rankings for things, like movies and music, often do not tell the public's taste, rather than the taste of a small number of buyers or users because social influence plays a big role in determining what is popular and what is not through an information flow [11, 12]. This is a hindrance to those users whose choices differ from the majority population. The conventionally used content-based filtering doesn't take user feelings into consideration so that alone cannot be used. Using a customized popularity based algorithm along with a content- based algorithm helps to achieve a recommender system that recommends movies that are not popular but have great user ratings (above threshold average ratings).

Data Collection and Segregation:

There are several ready-to-use movie datasets out there. The most well-known are probably the IMDB datasets. Internal, explicit data from users through scraping could also be used to help recommend movies. This data set contains around 1000 recently released movies. The columns contained in it are the movie title, genre, average ratings, and number of votes. Aside from these, there are other attributes like movie plot, cast, release year, reviews etc. but they are not necessary to determine which movie to recommend in the proposed system hence, can be scraped but is not used. The genre column also contains multiple entries for example action, adventure and horror for a particular movie. This means that a movie can belong to multiple genre which is further required in modeling the data. Our goal is to obtain only one column for each movie that contains all the characteristics together, in order to perform the vectorization. For that we need to do some cleaning and filtering first.

Data Cleaning and Filtering:

Now all the unnecessary columns if scraped can be dropped here. Since we only need 4 out of the 20 columns from IMDB here, others can be dropped. After scraping data (1000) movies, the access to the genres that the movie belonged to, how many votes a movie had received, and the movie's average rating on IMDB was received .The fewest

votes an individual movie had received were 8, the most votes an individual movie had received was 1, 53,693.The average number of votes a movie had received were 4,049. The hipster recommender system uses movies below 25 percentage of the average number of votes, since nothing much changes for the movies coming under 95 percentile of all movies. The rest have a wide margin of reception. So, the system, considers them as well as some mildly popular movies. This gives an initial threshold of approximately 19,213 votes. The proposed system also decided to look at movies whose average unweighted rating on IMDB was the mean or above. In the proposed system, an average rating at or above 6.5 is considered. Here, the genre column that contains multiple values is not normalized since they are further converted to bag of keywords in the next step.

A feature within the nltk package that allows extracting the key words from a text [13], and even assigns scores to each word is used in the proposed system. The Rake function is used to extract key words from the genre column, so we considered all the multi valued entries in the genre column and converted them into a bag of words form every movie. In order to do this, the function is applied to each row under the Genre column and assigned the list of key words to a new column.

Data Modelling:

In order to detect similarities between movies, vectorization is required. The hipster recommender system uses Count Vectorizer rather than Tf-Idf Vectorizer for a reason, a simple frequency counter for each word in bag of keywords column is needed. Tf-Idf tends to give less importance to the words that are more present in the entire corpus or column which is not wanted for the Hipster recommender system, because every word is important to detect similarity. The output matrix gives a value between 0-1 (0 being least similar and 1 being the most) on the similarity of movies in columns to the movies in the rows. Due to this the matrix is a symmetric matrix since every movie is equally similar to the other movie and all the movies are completely similar to itself therefore the value 1 is diagonal.

Once the matrix containing the count for each word is ready, the cosine similarity function can be applied to identify those movies that are similar to the input movie and the top ten movies that have the highest ratings is returned. It is based on the Cosine similarity calculation metric (formula 1) that calculates the angle between the vectors assigned to the two movies. It assigns a 1 to the most similar and a -1 to the most dissimilar movie pair.

$$\cos(\theta) = \frac{a.b}{||a||||b||} \quad (1)$$

Now that the similar movies are received, select only those that have an average ratings above 6.5 and the number of votes above 19, 213 which is the threshold calculated by –

Threshold = average number of votes X 0.25 (2)

In case there is a clash between two or more movies that have the same average ratings then the movies are arranged in the ascending order of its popularity i.e its number of ratings.

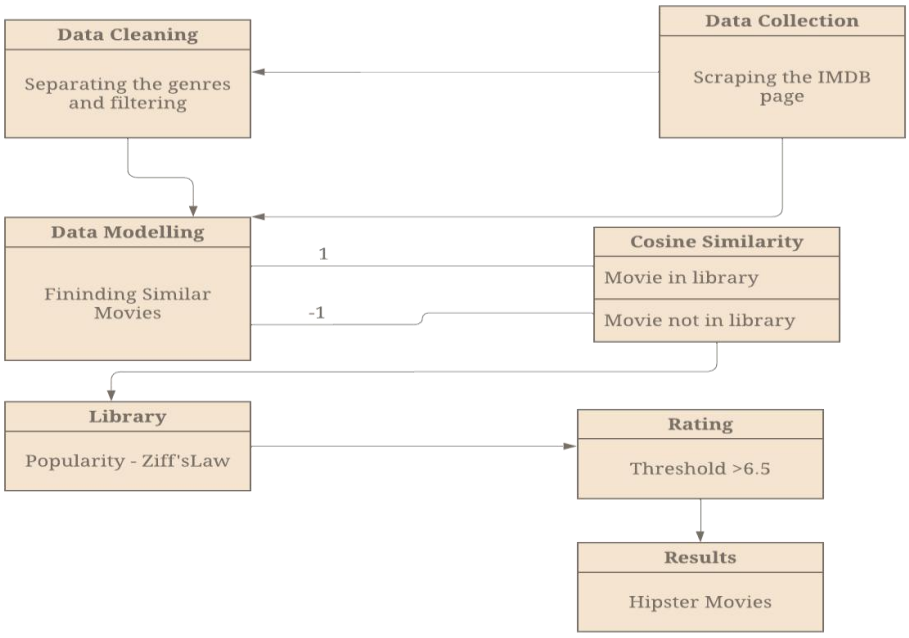

Figure 1: System Architecture of the proposed system

4. RESULT AND RESULT EVALUATION:

The proposed Hipster movie recommender system was used on a data set of 1000 Hindi movies recently released obtained after web crawling and scraping the IMDB website. The user input movie genre is then separated and filtered to get a bag of keywords and converted into a similarity matrix. On comparing the similarity between the input movie and the similar movie, only the movies having an average rating of 6.5 or above is ranked in the descending order of the number of votes received equal to or more than 19,213 votes. The top 10 output movies received on entering the movie "Masaan" is given in the table below-

Movie Name	Number of Ratings	IMDB Rating
96	16685	9.6
Black Friday	18254	8.5
Rangasthalam 1985	17972	8.4
Bangalore Days	15609	8.3
Section 375	10540	8.1
Thappad	13087	6.9
The Breadwinner	18961	7.7
Shaadi Mein Zaroor Aana	12145	7.6
Raat Akeli Hai	11589	7.3
Khuda Hafiz	12596	7.3

Table 1: Hipster Recommendation System Results

This paper has been evaluated using effective as well as practical evaluation metrics and uses a real data set. When compared with existing systems, for example, a movie recommendation system that uses AdaBoost classifier offers a precision of 0.93 using the RMSE results whereas the proposed recommendation machine shows a precision of 0.98. This is evidenced by the poll taken by a diverse group of people. Out of 10 of these people 9 of them suggested one of the movies shown in the table 1 when asked which movie they would watch next after watching the movie "Masaan". Every one of these users were asked to choose 5 movies each after watching the movie 'Masaan' from 2 lists- one containing the top 10 recommendations as shown in table 1 and another list containing the top 10 recommendations from an existing movie recommender system. The same users were again made to choose from 2 lists each from the existing recommender system and the proposed recommender system but this time having top 100 recommendations. The movies that majority of the users preferred to watch were unsurprisingly from the Hipster recommender system. A mean of these user preferences were then calculated in order to get the RMSE calculations. But saying that, taste in movies changes from person to person and demographic to demographic. Therefore, no matter the precision calculation of the movie recommendation system, the audience of the proposed system belong to a specific demography. More importantly, this system is tested on a small data set and no valid statistical evaluation is made [14].

5. CONCLUSION:

In the paper, a model based hybrid system is used by combining popularity and context based algorithms. This brings out true artistry forward and not just movies that have been watched again and again no matter its creativity by the proposed movie recommendation system. It is important to separate the movies first according to the user input and then apply the functions as this reduces the iteration of unnecessary movies whose genres are not preferred by the viewer. The average rating of 6.5 or above is focused on as those are the kind of movies that needs to be brought into the attention of the viewers.

We live in a world where content is consumed dramatically fast and there is shortcoming of good content daily only because they are not accessible among the pile of well favoured movies. Its a known fact that people tend to watch popular movies even though they have bad ratings. As the paper focuses on the "quality and not quantity", it achieves its target successfully. The accuracy of the proposed system was confirmed when a poll was taken on over 10 users over a wide range of age groups, interests, occupation etc. In the future, the use of more attributes that favors the user specific rating is to be implemented to better the performance.

REFERENCES:

1. Charu C Aggarwal (2016). *An introduction to recommender systems. In Recommender Systems,* (pp 1–28). Springer.
2. Nan Zheng and Qiudan Li (2011). A recommender system based on tag and time information for social tagging 124 systems. *Expert Systems with Applications, 38(4):4575–4587.*

3. Hao Wu, Yijian Pei, Bo Li, Zongzhan Kang, Xiaoxin Liu, and Hao Li (2015). Item recommendation in collaborative tagging systems via heuristic data fusion. *Knowledge-Based Systems, 75:124–140.*
4. B. R. Cami, H. Hassanpour and H. Mashayekhi (2017). *A content-based movie recommender system based on temporal user preferences, 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)* (pp. 121-125), doi: 10.1109/ICSPIS.2017.8311601.
5. Michael J Pazzani and Daniel Billsus (2007). *Content-based recommendation systems. In The adaptive web*,(pp 325–341). Springer.
6. Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro (2011). *Content-based recommender systems: State of the art and trends*. *In Recommender systems handbook* (pp 73–105). Springer.
7. C. Palmisano, A. Tuzhilin and M. Gorgoglione (2008), *Using Context to Improve Predictive Modeling of Customers in Personalization Applications, in IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 11*, (pp. 1535-1549), doi: 10.1109/TKDE.2008.110.
8. Walter Carrer-Neto, Mar´ia Luisa Hern´andez-Alcaraz, Rafael Valencia-Garc´ia, and Francisco Garc´ia-S´anchez (2012). *Social knowledge-based recommender system. Application to the movies domain. Expert Systems with application*s, 39(12):10990–11000.
9. Luis Omar Colombo-Mendoza, Rafael Valencia-Garc´ia, Alejandro Rodr´iguez-Gonz´alez, Giner Alor-Hern´andez, and Jos´e Javier Samper-Zapater (2015). *Recommetz: A context aware knowledge-based mobile recommender system for movie showtime*s. *Expert Systems with Applications*, 42(3):1202–1222.
10. Shouxian Wei, Xiaolin Zheng, Deren Chen, and Chaochao Chen (2016). A hybrid approach for movie recommendation via tags and ratings. Electronic Commerce Research and Applications, 18:83–94.
11. Shardanand, U., and Maes, P. (1995). *Social Information Filtering: Algorithms for Automating' Word of Mouth'.* In Proc. of CHI '95.
12. Kumar, Rajeev & Verma, Brijesh & Rastogi, Shyam (2014). *Social Popularity based SVD++ Recommender System. International Journal of Computer Applications.* 87. 10.5120/15279-4033.
13. Rose, Stuart & Engel, Dave & Cramer, Nick & Cowley, Wendy (2010). *Automatic Keyword Extraction from Individual Documents*. 10.1002/9780470689646.ch1.
14. Sait Can Yücebaş (2019). *MovieANN: A Hybrid Approach to Movie Recommender Systems Using Multi Layer Artificial Neural Networks*.Çanakkale Onsekiz Mart University Journal of Graduate School of Natural and Applied Sciences :5,2, 214-232