IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Volume 3, PP 31-35

Teenhub - Recommendation System

Pranav Jain[1], Kedar Kale[2], Ganesh Choudhary[3], Paresh Aher[4], Deepali Maste[5]
[1]*Computer Engineering, Atharva College of Engineering, India*
[2]*Computer Engineering, Atharva College of Engineering, India*
[3]*Computer Engineering, Atharva College of Engineering, India*
[4]*Computer Engineering, Atharva College of Engineering, India*
[5]*Assistant Professor, Computer Engineering, Atharva College of Engineering, India*

Abstract:*The concept of TeenHub falls into two domains: Machine Learning and Advanced Web Application. In TeenHub, the aim is to visualize the Machine Learning algorithms on the web applications by using smart data visualization tools, algorithms and techniques and not limiting them to just run onto terminals. It will be an all in one package for entertainment news and ratings.*
Keywords: *Machine Learning, Collaborative Filtering, KNN, Recommendation System, TF-IDF*

I. Introduction

On the Internet, where the number of choices is overwhelming, there is need to filter, prioritize and efficiently deliver relevant information in order to alleviate the problem of information overload, which has created a potential problem to many Internet users. Recommender systems solve this problem by searching through large volume of dynamically generated information to provide users with personalized content and services.

The explosive growth in the amount of available digital information and the number of visitors to the Internet have created a potential challenge of information overload which hinders timely access to items of interest on the Internet [2]. Almost 90% data on the internet has been generated in the last two years. Information retrieval systems, such as Google, and Altavista have partially solved this problem but prioritization and personalization (where a system maps available content to user's interests and preferences) of information were absent. This has increased the demand for recommender systems more than ever before.

TeenHub is an advanced Web Application which uses Machine Learning Algorithms. It is a recommendation engine for movies, songs, news and games. It also houses a custom search engine which makes navigation a lot easier. It uses TF-IDF [4] algorithm for the search engine and KNN algorithm for the recommendation system of the application. TeenHub promises to show only relevant data to the users with the help of these algorithms. To make sure that the models which will be created for each user remain accurate TeenHub will rely on more than 100,000 real ratings made by humans and not randomized ratings. This also makes sure that the overall accuracy of the system also remains high.

II. Literature Review

A number of techniques have been proposed to solve the problem of filtering and recommendation. They are explained briefly as follows,

A. Content-based filtering:

In content-based recommender systems, the goal remains the same that is to recommend items, but the approach is quite different than other types of recommender systems. Here as Marko Balabanovic, YoavShoham have explained, the system tries to recommend items which are similar to the items which were liked by the given user in the past [7]. Past activity is heavily used in content-based recommender systems. As per the example given by Adomavicius G., Tuzhilin A., in a movie recommender system application content-based recommender system will try to comprehend the similarities (particular actors, genres, producers, directors, movie general idea, location, etc.) among the movies that a user has rated highly in the past [8]. Then, using the similarities a list of movies is generated. This list contains movies which have a very high degree of resemblance with the movies that user liked. This list is then sorted according to their similarity. There might be a chance that some movies which are already rated by the user are also on the list, so these movies are removed from the list. This is the list that content-based recommender system would recommend. The content in content-based recommender systems can be the features of items. It generally works by extracting data from this content and hence content-based systems are designed mostly to recommend text-based items. These text-based items are represented as documents and to know the importance of these documents, we assign it a weight based on the keywords the documents have. A well-known method called term-frequency / inverse document frequency

(TF-IDF) is used to measure this weight. Term-frequency of a document is known as the number of times a term appears in a document. There are some words in the documents which don't have much meaning and are generally common to all documents, for example words like, 'the', 'a', and many more, so words like these are removed while calculating the weight of the document. This process is known as Inverse document frequency. This is the reason why we see Inverse document frequency in combination with term-frequency. Examples of content based recommender systems are InfoFinder, NewsWeeder, Fab. As mentioned in the paper by Adomavicius G. and Tuzhilin A.,Fab, a web page recommender uses 100 most important words to describe a web page[8]. Similarly, the Syskill&Webert system uses 128 most important words to do the same job.

B. Collaborative filtering:

Collaborative filtering predicts items for particular user based on items rated by other users. It makes prediction about the interests of user by collecting preference from many users. As written and explained by Terveen Loren, Hill Will,these systems focus on algorithms for matching people based on their opinion (likes or dislikes, preferences, ratings etc.) and weighting the interests of people with similar taste to produce a recommendation for the information seeker [9].

The underlying assumption of the collaborative filtering approach is that if a user X has same point of view as a user Y on a problem (issue), then X is more likely to have Y's point of view on different problem (issue). For example, in movie recommendation application, in order to recommend movies to user A, collaborative filtering system tries to find out users who have similar tastes in movies with user A. Then, only the movie that are most liked by other users which are similar to user A would be recommended

Types of Collaborative filtering:

1. User-based collaborative filtering:

In user-based collaborative filtering it compare user's opinion (likes or dislikes) for an item and find out similarity between users. Similarity between users is decided based on their overlapping opinion (likes or dislikes) for an item.For example, two users A and B like X and Y movies and now user A also likes Z movie, then according to User-based collaborative filtering user B will also like Z movie.

2. Item-based collaborative filtering:

Unlike User-based collaborative filtering in Item-based collaborative filtering similarity between items is evaluated based on user's opinion (likes or dislikes) for an item.For example, user A likes X, Y and Z movies, another user B likes X and Z movies and dislikes Y movies, based on users opinion (likes or dislikes) there is similarity between X and Z movies. So when C user like X movie, then according to Item-based collaborative filtering user C will also like Z.

C. Hybrid systems:

Recently hybrid recommendation systems are being developed which involve combination of both content based and collaborative filtering. The drawbacks of using purely content based or collaborative filtering are as follows. In content based filtering user preferences are matched with item description to give recommendations, if the user profile is new and have a low amount of user preferences then the quality of recommendations will be poor. This particular problem is referenced to as "cold start". In collaborative filtering recommendations are calculated based on users similar to the user, people who have rated items similar to the user are used to predict what other items the user will rate highly. For example, if user A rates items x and y as 8 and 9 respectively and user B rates items x, y and z as 8, 9 and 9 respectively it can be assumed that user B is similar to user A and items rated highly by B will also be rated highly by A, hence item z is presented as the recommendation to user A.

The problem with this approach is that it calculates the correlation between users and not the items themselves. So if user B rates some items highly which are unrelated to items previously rated then the recommendation is highly skewed. In such case if the number of users is low then the probability of unrelated recommendation is high. The dilemma while choosing a suitable technique is whether to choose a user correlation based approach or content correlation based approach.

The above dilemma can be dealt with by implementing a combination of techniques to minimize their individual drawbacks. As mentioned by Adomavicius G. and Tuzhilin A., the techniques can be unified into a single model or can be implemented separately and their results can be weighed to achieve most appropriate answer[8]. Netflix is a good example of the use of hybrid recommender systems. The website makes recommendations by comparing the watching and searching habits of similar users (i.e., collaborative filtering) as well as by offering movies that share characteristics with films that a user has rated highly (content-based filtering).

III. Comparative Study

For creating recommendation engines, we had the option of creating the engine based on content-based technique, user based collaborative filtering and item based collaborative filtering.

In content-based recommendation user interests are collected from the user and items similar to those interests are recommended to the user [6]. Also, the items similar to the ones the user has rated highly are recommended to the user, the disadvantage of this approach is that the user is not presented with any new items from other categories and the recommendation feels unnatural, for example if the user has selected only sports as his interest in an e-commerce website then only sports goods will be recommended and items from other categories which fall in the neighborhood of the item will be neglected. If the user purchases a football from the sports category he may be interested in purchasing some shoes, socks and jerseys too, but since recommendation is strictly restricted to the sports category and the other items belong to shoes and fashion category the natural items to recommend are ignored.

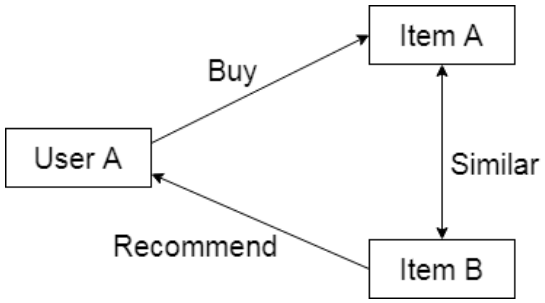

Figure 1: Working of content-based recommendation system

In user based collaborative filtering correlation between existing users is calculated to find the most similar user to the current user then the unpurchased items are recommended to the user [7]. The drawback of this approach is that for a site housing millions of items each user may have purchased only up to a few hundreds of those items and similar users are very sparse and since the users are very few in number the recommendations generated from a sparse dataset has a good probability of being inaccurate.

Item based collaborative filtering generates correlation between items based on ratings by same users [5]. The advantage of this approach is that since the items are correlated there is no strict filtering based on categories and it also eliminates the drawbacks of user based collaborative filtering.

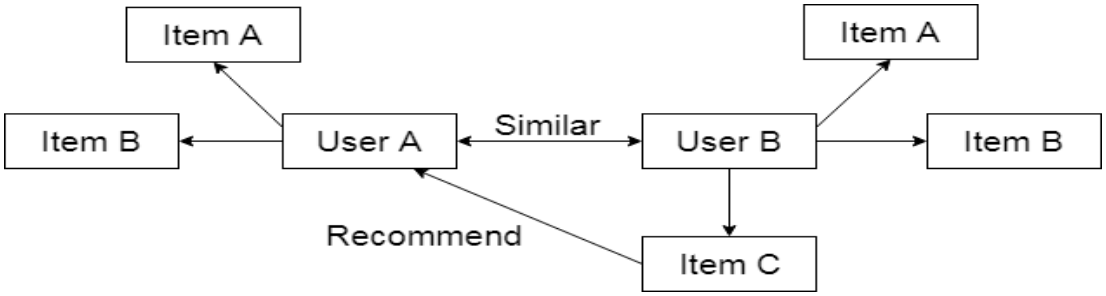

Figure 2: Working of collaborative filtering recommendation system

Hence the concept of TeenHub will adopt the item based collaborative filtering approach to provide the most appropriate recommendations.

IV. Proposed System

The proposed system has three main components:

Similarities among items: In this step, correlation between selected item rating and all other items ratings in database is evaluated. Item with highest correlation value is recommended to user.

Classification: This step applied at start of application. In this step classification of movies, games and songs based on different genre. For each possible genre there is panel which display movies, games and songs related to genre.

Search Engine: This step considerstf-idf weight, which is composed by two terms, the first computes the normalized Term Frequency (TF), aka. The number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed

as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

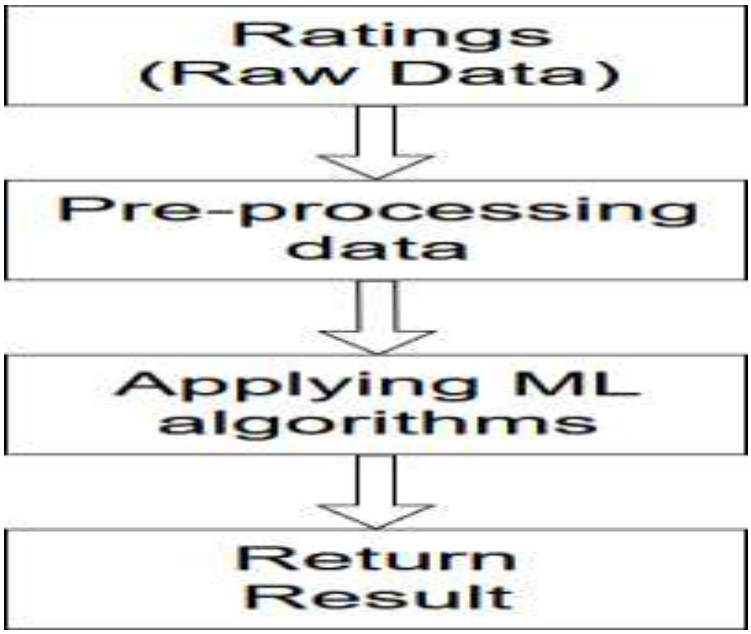

Figure 3: Overview of a Recommendation System

Here, the aim is to provide a one place solution for all entertainment needs of users of all age group. The objectives are:

1. To implement recommendation engine for movies, songs, games and news.
2. To implement a custom search engine.
3. To design a responsive design for the web application.
4. To implement a dashboard for the users to view their activities.
5. TeenHub has various entertainment fields such as movies, songs, games and news which help to user to get all kind of information on the same platform.
6. It uses user previous activity such as rating, history to recommend new movies, games, or songs and it makes user-friendly environment.
7. TeenHub make easy for the user to access all field just by single step registration.
8. In order to select best games or movie, it also provide weekly, monthly and yearly ranking of games and songs, based on user rating in particular genre.
9. TeenHub also includes news related to games and movies and weekly popular movie to help user for getting best choice.
10. It allows user to rate movie, review, rate game in any appropriate genre and accordingly system recommend movies, games or songs to him/her.

V. Item Based Collaborative Filtering

First, the system executes a model-building stage by finding the similarity between all pairs of items, such as correlation between ratings. Second, the system executes a recommendation stage. It uses the most similar items to a user's already-rated items to generate a list of recommendations [5]. This form of recommendation is analogous to "people who rate item X highly, like you, also tend to rate item Y highly, and you haven't rated item Y yet, so you should try it".

The user ratings are compared across movies, and ratings of other users to generate a correlation matrix. Each row of this matrix represents a user and each column represents s a movie. Each cell contains a number which corresponds to the relation between different user ratings and movies. Using this number, we determine whether a user will like a particular movie or not. Our custom algorithm gives rewards to similar movie ratings which further helps in giving recommendations.

VI. K Nearest Neighbors

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression [10]. In both cases, the input consists of the k closest training examples in the feature space.

After a user has expanded a movie to view its information, similar movies to the current movie are displayed to the user using KNN. KNN determines which movies to display by classifying movies based on their genres and ratings.

VII. Conclusion

The users will have one common login for all the 4 parts. The users will get a dashboard to see their own data and activities on interactive charts. The user can login as a guest or make an account for self. The recommendation engine will recommend the user movies based on his previous ratings for movies. Also, when the user browses through movies, songs or games, KNN algorithm will make sure to display the user with suggestions similar to the current item. A news app is also included to keep the user updated with the latest of news in the entertainment industry.

The custom search engine will be based on TF-IDF algorithm to navigate through the application. The movie recommendation engine will be based on KNN algorithm and will trained on a dataset of not less than 1,00,000 real ratings.

Although, TeenHub provides facility for user to get various entertainment information on single platform and due to recommendation based system it helps user to get right content and it also provide ranking of movies and games which help user to select proper movie or game and it has daily news feature which keep user updated.

References

Journal Papers:

[1]. DietmarJannach and Simon Fischer, Recommendation-based Modeling Support for Data Mining Processes, Germany and Simon Fischer Rapid Miner GmbH, Germany, *Proceedings of the 8th ACM Conference on Recommender system*, pages 337-340, October 2014.

[2]. Ali Heydarzadegan, YaserNemati, Mohsen Moradi, Evaluation of Machine Learning Algorithms in Artificial Intelligence, *IJCSMC, Vol. 4, Issue. 5, May 2015*, pg.278 – 286

[3]. Tian Xia, Yanmei Chai, An Improvement to TF-IDF: Term Distribution based Term Weight Algorithm, *JOURNAL OF SOFTWARE, VOL. 6, NO. 3, MARCH 2011*

[4]. Ramos, Juan. (2003). Using TF-IDF to determine word relevance in document queries.

[5]. G. Linden; B. Smith; J. York, Amazon.com recommendations: item-to-item collaborative filtering, *IEEE Internet Computing (Volume: 7, Issue: 1, Jan/Feb 2003)*, pages 76-80

[6]. Michael J. Pazzani, Daniel Billsus. "Content-Based Recommendation Systems". *Springer-Verlag Berlin Heidelberg 2007. doi: 10.1007/978-3-540-72079-9_10*

[7]. Marko Balabanovic, YoavShoham. "Content-Based, Collaborative Recommendation". *Communications of the ACM, Volume 40 Issue 3, March 1997, Pages 66-72. doi: 10.1145/245108.245124*

[8]. Adomavicius, G.; Tuzhilin, A. (June 2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". *IEEE Transactions on Knowledge and Data Engineering. 17 (6): 734-749. doi:10.1109/TKDE.2005.99*

[9]. Terveen, Loren*;* Hill, Will (2001). "Beyond Recommender Systems: Helping People Help Each Other" *(PDF). Addison-Wesley. p. 6*. Retrieved 16 January2012.

[10]. Shweta Taneja, Charu Gupta, Kratika Goyal, DharnaGureja. (February 2014). "An Enhanced K-Nearest Neighbor Algorithm Using Information Gain and Clustering". *Advanced Computing & Communication Technologies (ACCT). doi: 10.1109/ACCT.2014.22*