An Efficient Movie Recommender Engine: Application of Artificial Intelligence

G. Vidyulatha[1], J. Nagarjun Naik[1], A. Yashwant Reddy[1]

[1]Department of CSE, Sree Dattha Institute of Engineering and Science, Hyderabad, India.

ABSTRACT

A recommendation system is a system that provides suggestions to users for certain resources like books, movies, songs, etc., based on some data set. Movie recommendation systems usually predict what movies a user will like based on the attributes present in previously liked movies. Such recommendation systems are beneficial for organizations that collect data from large amounts of customers and wish to effectively provide the best suggestions possible. A lot of factors can be considered while designing a movie recommendation system like the genre of the movie, actors present in it or even the director of the movie. The systems can recommend movies based on one or a combination of two or more attributes. In this paper, the recommendation system has been built on the type of genres that the user might prefer to watch. The approach adopted to do so is content-based filtering using genre correlation. The dataset used for the system is Movie Lens dataset.

Keywords: Recommender system, clustering, k-means clustering, content-based filtering, collaborative filtering, hybrid filtering.

I. INTRODUCTION

Recommender Systems has emerged as important factor for recent online shopping sites. As they make huge increase on the products sales, to find current trends, promotion opportunities for sellers. For users it helps to solve information overload, personalized recommendation, and finds new things in many online shopping sites. Recommender systems have been using Collaborative filtering as it0 s primal method. Profound work has been done to improve accuracy of recommender system. The necessity of recommender system is accuracy and speed to predict [1].

Recommendation systems or recommendation engines are a form of data filtering system that try to recommend products or any other information that are of interest by users. At starting, recommendation technology was relatively unrefined. It just recommended dissimilar products which other users had purchased, but the technology has become more sophisticated and is now an integral part of many online retailer0 s economic models. Complex algorithms have been used to analyze vast amount of data and determine what products the potential customers want to buy based on their shopping choices that a user makes. The user-based approach enables the user to personalize the recommendations based on their taste [2], hence the system will generate more apt user centric recommendation. Personalization enables user to overcome data overload and can make searching more efficient and less time consuming [3].

Many websites have implemented their version of recommendation system but lack accuracy to some extent. Categorical jump in recommendation have also been detected where some recommended products have no relation with the searched products, thus making the recommendations erroneous. This system now only focuses movies as a product and for the future works all other categories of products are planned to be included thus making system much diverse in terms of products accommodated and products recommended [4]. Below figure (Fig:1) shows the design of the system.

Users interact with the system through a web page. Recommendation engine stores user activity details to a database. Depending upon the user’s activity, the analyzer groups the users based on their ratings given to a product. Groups created will have similar mindset users with similar tastes. Generated recommendation is provided back to the users. Analyzer does periodic and instant updates to generate the recommendations. Recommendation database hold the recommendations generated by the analyzer. Recommendations can be directly given to the user from the data available from the database [7] [8].

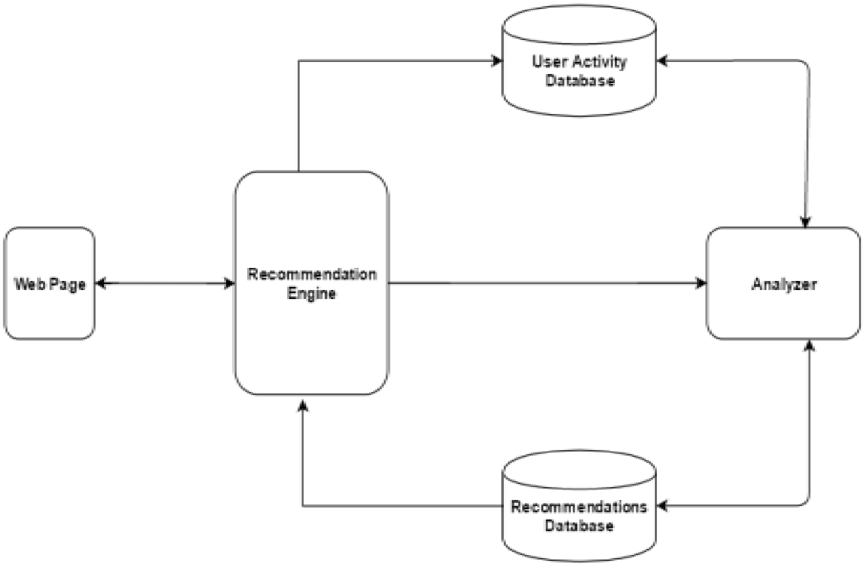


Fig. 1. System Design

II. RECOMMENDATION SYSTEM USING CONTENT-BASED FILTERING

The approach used for building the recommendation system is content-based filtering. As discussed earlier, content-based filtering analyses user's past behavior and recommends items like it based on the parameters considered. This aims at recommending movies to users based on similarity of genres. If a user has rated high for a certain movie, other movies containing similar genres are recommended by the system. The dataset used in for this purpose is subdivided into two sections.One section contains the list of movies along with the genres that they have been categorized under. The other part of the dataset contains a list of ratings of movies that have been rated by the user on a scale of 1–5, with 5 being the highest. First, a combined dataset of movies, genres and their ratings have to be constructed for correlating genres with the ratings. For the sake of simplicity, the ratings have been converted to binary values. If the rating given by a user is greater than 3, it receives a value of 1, otherwise it receives a value of −1. The genres are also segregated in a binary format, maintaining a consistent approach. Out of the set of 11 genres that are present in total, if a movie has a certain genre, it receives the value of 1. If the genre is not present in the movie, it receives a value of 0. The user profile matrix provides a combined effect of the genres and ratings by computing the dot product of the genre and the ratings matrix. Again, for the sake of consistency, a binary format is adopted. If the dot product is a negative value, 0 is assigned to it. For a positive value, 1 is assigned to it. After obtaining a dot product matrix of all the movies, a similarity measure is calculated by computing the least distance between the user under consideration and the others. The values which have the least deviation with respect to the current user's preferences are the ones that are recommended by the system. The algorithm adopted for building the recommendation system is given below:

Algorithm 1 : content-based filtering
Step 1: Construct a data frame of the genre dataset with movie ID as the rows and genres as columns separated by pipeline character.

Step 2: Make a list of all the genres that are available in the dataset.
Step 3: Iterate through the previously made genre data frame. If a genre is present in a movie, value of 1 is assigned to the genre matrix.
Step 4: Read the ratings sheet and construct a ratings matrix which assigns 1 for movies which has rating more than 3 and −1 for movies which has ratings less than or equal to 3.
Step 5: Calculate the dot product of the two matrices—genre matrix and ratings matrix. This is the result matrix Step 6. Convert the result matrix to a binary format. For a negative dot product value, assign 0, else assign a value of 1.
Step 7: Calculate the Euclidian distance between the current user and other users.
Step 8: Retain the rows which have the minimum distance. These are the recommended movies for the current user.

Offloading is a method of transferring resource-intensive application from portable device to remote server by considering different parameters. Offloading mechanisms involves three tasks before it get executed. They are partitioning, profiling, offloading decision.

Collaborative Filtering Algorithm

The users are clustered based on ratings given by them for each movie. The dataset has been centered on zero by subtracting the mean from individual rows. The resultant value of clustering will give clusters containing groups of users. The training set marked with the class labels is fed into the random forest classifier for learning. Test set is tested upon to get the predictions.

Random forest classifier

Random forest is most accurate ensemble classifier and works efficiently on huge dataset. It can effectively predict the missing data accurately, even in situations where large portions of data are missing and without pre-processing[12]. It combines bagging and random feature selection. Random forest contains decision trees that are combined individual learners. Random subset of training data is used to generate trees. The test rows are passed through the forest after the forest have been trained. Each tree generates an output class we take the mode of that classes as the output of random forest [13][17]. Classification of labels using random forest classifier has been done as mentioned in Algorithm 3. In the method proposed Random forest classifier is used to predict the labels of users. The dataset has been divided into train and test set.The training set is given as an input to the classifier. The classifier will label the users into its respective classes which has been learned upon during the training phase.

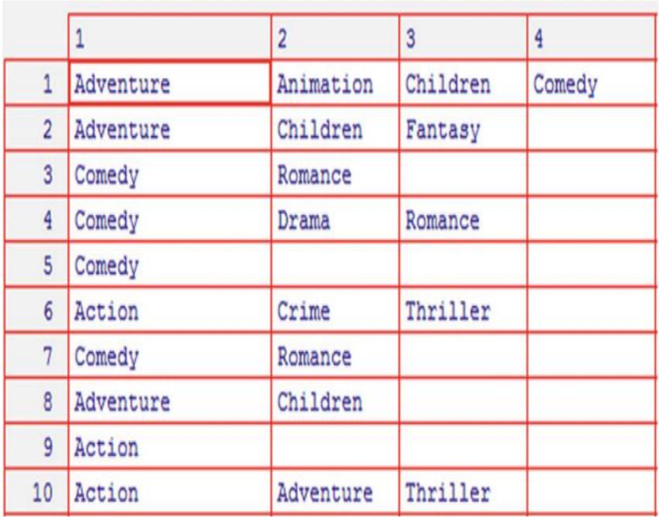
Algorithm 2: Random Forest Classifier

```
Input: Training data(Td)

Output: class label

1) To form t classifiers:

    for i = 1 to t do

            Select Tdi from the training data Td randomly Generate a root node
        called Rni with Tdi Invoke GenerateTree(Rni )

    end for
```

```
2) Generate Tree(Ni)
        if Ni consists only one class instance then
                return
        else
                Choose possible p% of the node(splitting features) in Ni randomly
                For splitting elect the feature called F with the more gain of information
                Build f child nodes of Ni , Ni1 ,..., Nif , here Fi possible values in can be in F that are Fi1 Fif
                        for j = 1 to Fi do
                                Replace Nij the contents of to TDj , here TDj is instances of Ni which is match to Fi Invoke GenerateTree(Ni)
                        end for
        end if
```

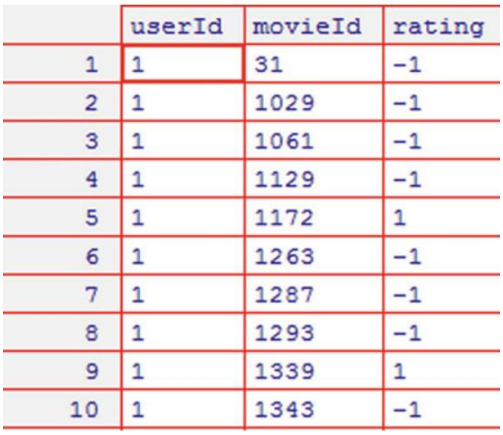
III. SIMULATION RESULTS

The genre matrix constructed with rows containing movies and genres separated by columns. There is a total of 11 genres in the dataset (Fig. 2). The ratings matrix for each user corresponding to the movie ID is converted to a binary format. Every user has rated one or more than one movie (Fig.3). Using the genres matrix and ratings matrix, the result matrix is computed which is the dot product of the previous two matrices. The result is further converted in a binary format in Fig. 4. If the value of the dot product is more than 0, 1 is assigned to that cell otherwise 0 is assigned.



	1	2	3	4
1	Adventure	Animation	Children	Comedy
2	Adventure	Children	Fantasy	
3	Comedy	Romance		
4	Comedy	Drama	Romance	
5	Comedy			
6	Action	Crime	Thriller	
7	Comedy	Romance		
8	Adventure	Children		
9	Action			
10	Action	Adventure	Thriller	

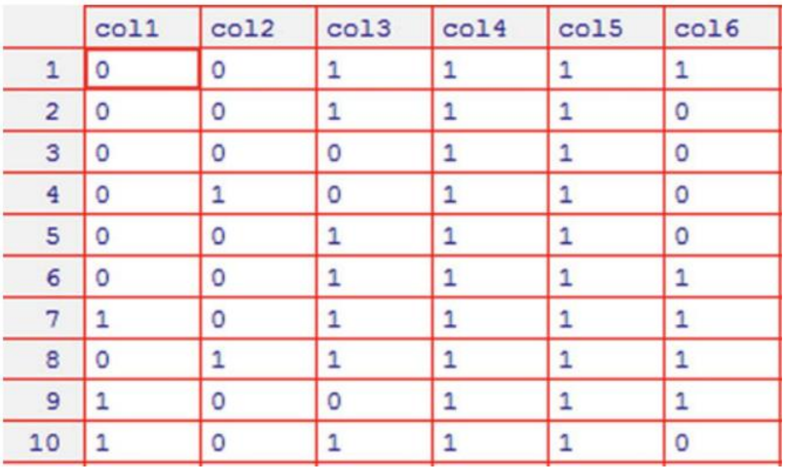
Fig. 2 Genre matrix



	userId	movieId	rating
1	1	31	-1
2	1	1029	-1
3	1	1061	-1
4	1	1129	-1
5	1	1172	1
6	1	1263	-1
7	1	1287	-1
8	1	1293	-1
9	1	1339	1
10	1	1343	-1

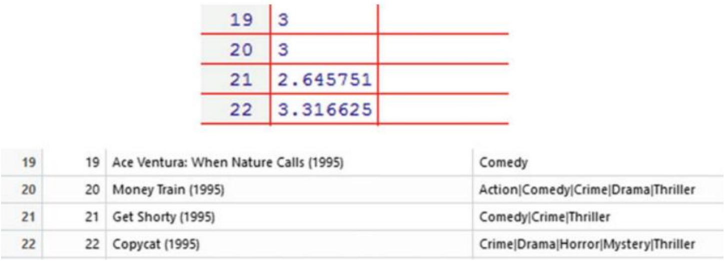
Fig. 3 Ratings matrix

After computing the result matrix, the Euclidean distances with respect to the other users are obtained and the ones having the minimum value is recommended as represented in Fig. 4. Figures 5 and 6 shows the output of the various movies that have been recommended to the users based on their previous behavioral patterns.



	col1	col2	col3	col4	col5	col6
1	0	0	1	1	1	1
2	0	0	1	1	1	0
3	0	0	0	1	1	0
4	0	1	0	1	1	0
5	0	0	1	1	1	0
6	0	0	1	1	1	1
7	1	0	1	1	1	1
8	0	1	1	1	1	1
9	1	0	0	1	1	1
10	1	0	1	1	1	0

Fig. 4 Result matrix



19	3	
20	3	
21	2.645751	
22	3.316625	

19	19	Ace Ventura: When Nature Calls (1995)	Comedy
20	20	Money Train (1995)	Action|Comedy|Crime|Drama|Thriller
21	21	Get Shorty (1995)	Comedy|Crime|Thriller
22	22	Copycat (1995)	Crime|Drama|Horror|Mystery|Thriller

Fig. 5 Euclidean distance

IV. CONCLUSION AND FUTURE WORK

The recommendation system implemented in this paper aims at providing movie recommendation based on the genres of the movies. If a user highly rates a movie of a genre, movies containing similar genres will be recommended to him. Recommendation systems are widely used in today’s era of Web 2.0 for searching for reliable and relevant information. While simple recommendation systems recommend users based on a few parameters, complex ones take many parameters into consideration.

By implementing machine learning in recommender systems, intel-Content-Based Movie Recommendation System Using Genre Correlation i-gent recommendations can be made for customers. Given the potential of such systems, they have a huge commercial value. Several MNCs have been exploiting the potential of recommendation system to lure customers into using their products. This also impacts greatly on the field of data mining and web mining. Mobile cloud computing (mcc) can save energy, improve application and experience of the users. All frameworks mentioned above have their own benefits and issues but still not up to level to address all issues related to security, energy, and user experience. Security issues are key problem in mcc, they need to be focused more compare to other issues.

REFERENCES

[1] Utkarsh Gupta and Dr Nagamma Patil, ”Recommender System Based on Hierarchical Clustering Algorithm Chameleon”. Dept of Information Technology, National Institute of Technology Karnataka, 2015.

[2] Al Mamunur Rashid, George Karypis, and John Riedl,” Learning Preferences of New Users in Recommender Systems: An Information Theoretic Approach”, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN-55455, 2008.

[3] Emmanouil Vozalis, Konstantinos G. Margaritis,”Analysis of Recommender Systems Algorithms”,University of Macedonia ,September, 2003.

[4] Daniar Asanov, ”Algorithms and Methods in Recommender Systems”, Berlin Institute of Technology Berlin, Germany, 2011.

[5] X. Su and T. M. Khoshgoftaar, ”A survey of collaborative filtering techniques”, Department of Computer Science and Engineering, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA, 3 August 2009.

[6] Hans BystrÃ ̈um ”Movie Recommendations from User Ratings”, Stanford University, 2013.

[7] S. S. Anand and B. Mobasher, ”Intelligent techniques for web personalization”, Proceedings of mm the 2nd Workshop on Intelligent Techniques in Web Personalization. 2003.

[8] Matteo Manca, Ludovico Boratto, Salvatore Carta,”Design and Architecture of a Friend Recommender System in the Social Bookmarking Domain”, Dipartimento di Matematica e Informatica Universit‘a di Cagliari via Ospedale 72, 09124 - Cagliari, Italy, August 27-29, 2014.