An Efficient Recommendation System using Random Forest in Machine Learning

[1]KASIMALLA BHAVANI, [2]Dr. K.VENKATA RAO

[1] M. Tech Scholar, [2]PROFESSOR,
Department of Computer Science & Systems Engineering,
Andhra University, Visakhapatnam, Andhra Pradesh, India.

ABSTRACT: Machine learning is a scientific study of algorithms which is used to learn from data without any human interaction. Recommendation system or Recommender System comes under Machine learning method which has become one of the essential systems in e-commerce websites. Many techniques are proposed to efficiently capture the opinion of the users and to provide recommendations accurately. Recommendation System that seeks to predict the preference (rating) a user would give to an item. They are primarily used in commercial application. Collaborative filtering is one such successful method to provide recommendation to the users. In this paper, we demonstrate Collaborative filtering methods, which are classified as memory-based and model-based. Memory based algorithms: User based collaborative filtering (UBCF) and Item based collaborative filtering (IBCF). Model based algorithms: K-Means and Random Forest Classification. Random forest predicts recommendations based on users preferences while targeting users interest and current trends. We evaluated the result with the help of the well-known MovieLens dataset, the result show that random forest approach is more reliable than other algorithms in terms of RMSE value.

Keywords: Recommendation System, Machine learning, Collaborative Filtering, Random Forest, RMSE.

I. INTRODUCTION

Recommendation System (RS) is one type of Information Filtering System(IFS). Recommendation System that seeks to predict the rating(or preference) a user would give to an item. Recommendation Systems are used in a different areas, and are most commonly identified as playlist generators for video, Audio, Service (like Netflex,YouTube etc.,) product recommenders for available services such as Amazon, social Media. RS can be classified into three major techniques are used for processing input data and formulating the prediction: collaborative filtering (CF), content-based filtering (CBF)[7][9], Hybrid filtering(combination of CF and CBF).

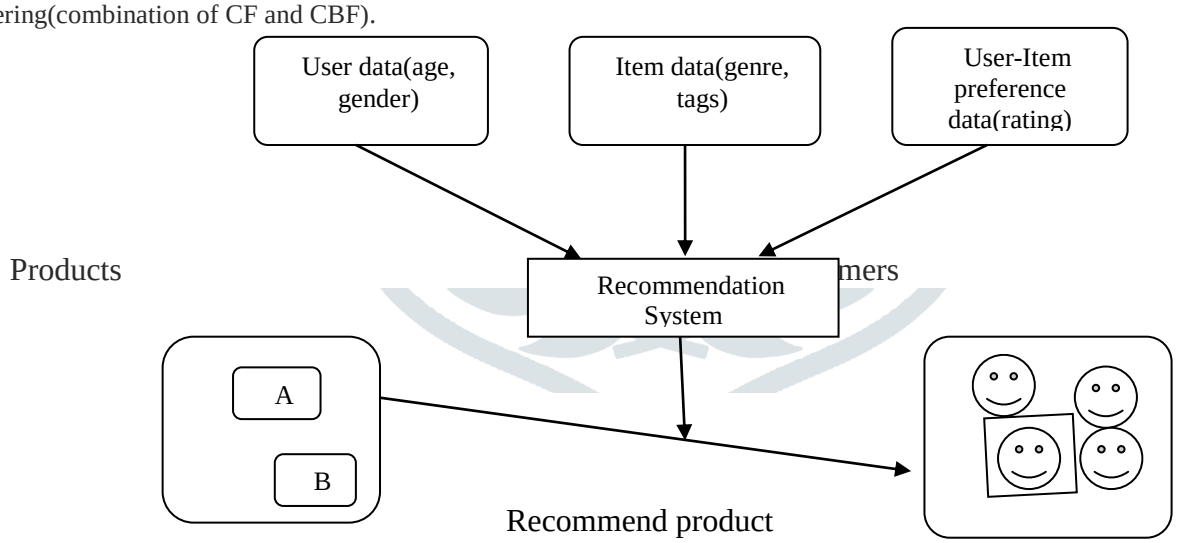

Figure 1: Recommendation Process

I.I COLLABORATIVE FILTERING:

In Collaborative Filtering, user previously purchased or selected items and/or Numerical ratings given to those items, as well as similar decisions made by other users. This model predicts items or items ratings that the user may have interest in. The CF system generates recommendations using only information about rating profiles for different users or items. On the other hand, CBF requires that items are described by features, and it utilizes pre-tagged characteristics of an item in order to recommend additional items with similar properties. In contrast to content-based filtering, collaborative filtering is applicable to any type of content, while it can also capture concepts that are hard to represent, such as quality. Additionally, Collaborative filtering has been acknowledged as the most successful and most widely implemented recommendation technique. For these reasons, we will focus on the collaborative filtering strategy in this paper[10].

Collaborative filtering technique can be distinguished into two major classes: model-based and memory-based. In Memory-based approach, assumes that the user and item previously ratings in database must be present in the system memory while recommendation algorithm running. The similarities between different users (or items) are calculated by searching the user and item database and then aggregating the interest of neighbors as recommendations [8]. In Model-based approach, CF models are developed using machine learning algorithms to predict user's rating of unrated items.

a. Memory-Based Collaborative Filtering:- Memory-Based method contains user based collaborative filtering(UBCF) and item based collaborative filtering(IBCF). Memory based technique rely heavily on simple similarity measures (Cosine Similarity and Pearson Correlation) to match similar users or items together. Memory- Based method faces some drawbacks like cold-start and scalability problem.

b. Model-Based Collaborative Filtering:- Model-Based method contains guessing how much a user will like an item that they did not encounter before. For that more machine learning algorithms to train to the vector of items for a specific user then they can predict the users preference (rating) for new item that has been added to the system. It rectifies the problem faced by the memory-based technique. Model-Based algorithms are typically faster than Memory-Based ones at query time and they are more suitable for cold-start recommendation. K-Means Clustering, Random Forest algorithms comes under model-based collaborative filtering. In this paper presents the efficient recommendation system using random forest in machine learning.

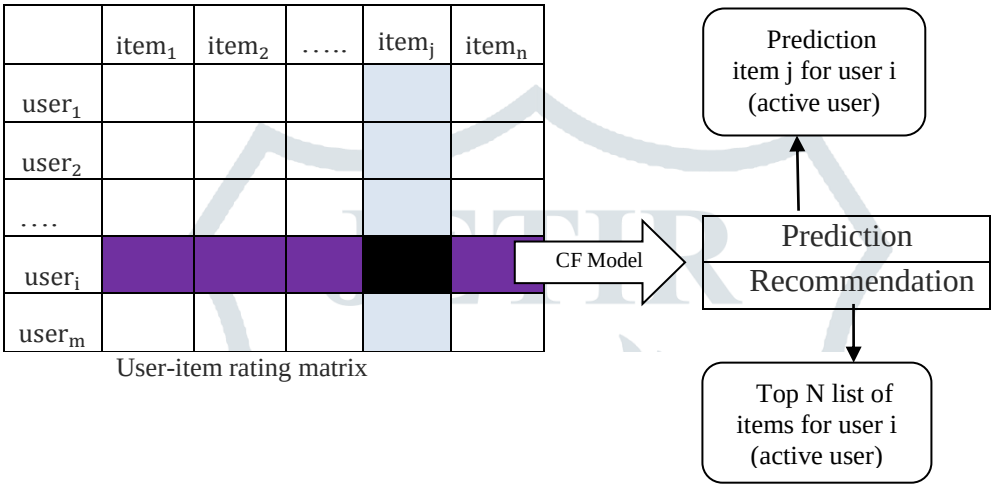

Figure 2: Collaborative Filtering Process

The rest of this paper is organized as follows. We provide recommendation system using collaborative filtering in Section 2, and propose our methods in section 3. In Section 4, we show the dataset details. In section 5, we provides experimental results of the proposed method. Finally, we reach a conclusion in Section 6.

II. RELATED WORKS

In this section presents the related works relevant to using MovieLens dataset for implementing recommendation system algorithms:-K-Means clustering, collaborative filtering based on user clustering and item, content based filtering, Hybrid approaches and popularity based approaches.

[1] Shivani Sharma, the Author proposed, "A Recommender System Based on Improvised K- Means Clustering Algorithm ", this paper shows how the change in selection of centroids improves the quality of recommendations and also decreases the execution time. The improvised approach has been authenticated with extensive set of experiments based on MovieLens dataset. These experiments proved that the improvised approach of RS provides better quality clusters, less execution time than existing algorithm and improves the accuracy of recommendations.

[2] SongJie Gong, the Author proposed, "A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item", this paper try to solve the problems of scalability and sparsity in the collaborative filtering, for that the system uses a personalized recommendation approach joins the user clustering technology and item clustering technology and finally use the item cluster collaborative filtering to produce the recommendation.

[3] N Lakshmipathi Anantha and Bhanu Prakash Bhattula, Authors proposed," A Review on Recommendation System using Rating Dataset",this paper contains recommendation systems are being used in each field like Ecommerce or M-commerce, social networking, research articles, music and travel. Collaborative filtering, content based filtering, Hybrid approaches and popularity based approaches are used in Recommendation Systems. Using R tool analyzed how many similarity of users and similarly movies are available in the dataset is calculated, how recommendations are generating and finally done evaluation part on developed recommender engine.

III. METHODOLOGY:

The main goal of this paper is to develop an recommendation system based on users ratings (preferences) and to estimate the system using evaluation techniques. In this, perform analysis on MovieLens data and to recommend the new or untried movies to users. We explore the different techniques IBCF and UBCF with Cosine similarity, IBCF and UBCF with Pearson Correlation, K-Means clustering, Random Forest and compare all techniques for efficient result for recommendation system. To obtain our research goal, we are using random forest algorithm in order to reduce the root mean squared error value.

I. Data Preprocessing :- In RS, we are using Movielens dataset, which contains 1682 observations and 23 variables and sparse as well, there may be users that might have hardly rated any movies (watched or may not be watched) and many a movies which may not be rated to a good extent. To maintain a good baseline on which recommendations could be made. we will take into consideration those users who have rated at least 20 movies and those movies that are rated at least 50 users. To extracting data the comprises of at least 20 preferences(ratings) for user and 50 ratings.

II. Algorithms:

i.User-based Collaborative Filtering(UBCF)

UBCF algorithm has a mainly focus on filling an user-item matrix and recommending based on the users more similar to the active user. In UBCF algorithm, First let's prepare the data for validation and using k-fold to validate the model, forming train and test dataset for achieves the result accurately. UBCF is applied with Similarity functions (Cosine Similarity & Pearson Correlation) to identify 25 neighboring users with similar characteristics and base recommendations on that basis, after that apply test model to recommend top 5 movies to each of user, then try to find latest predicted movies for each user as most recommend.

ii.Item-based Collaborative Filtering (IBCF)

IBCF algorithm fills a Item-Item matrix , and recommends based on similar items. First, let's prepare the data for validation and using k-fold to validate the model, forming train and test dataset for achieves the result accurately. IBCF is applied with Similarity functions(Cosine Similarity & Pearson Correlation) to identify 25 neighboring items with similar genre profile and base recommendations on that basis. Next apply the model to test. Lets suggested (recommend) top 5 movies to each of user. Finally we try to obtain the latest among those predicted for each user as most recommend.

Cosine Similarity : In Cosine similarity function, two items/users are thought of as two vectors in the m dimensional user-space/item-space. The similarity between them is measured by computing the cosine of the angle between these two vectors[4].

Formally, in the m and n ratings matrix similarity between items i and j, denoted by sim(i,j) given by

$$\text{Sim(i,j)= cos(A,B)}= \frac{A*B}{||A||*||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad \text{... (Eq.1)}$$

Pearson Correlation Coefficient (PCC): It finds the linear relationship between two vectors**.** The coefficient value ranges from −1 to +1, tracing both positive and negative correlations. A typical measure of similarity is the Pearson correlation coefficient, which is applied on the items rated in common by two users[4],

$$W(i,j) = \frac{\sum_k (R_{i,k}-\overline{R_i})(R_{j,k}-\overline{R_j})}{\sqrt{\sum_k ((R_{i,k}-\overline{R_i}))^2 \sum_k ((R_{j,k}-\overline{R_j}))^2}} \quad \text{... (Eq.2)}$$

Table 1. RMSE Values for UBCF & IBCF

1.	IBCF_Cosine	1.443
2.	IBCF_Pearson	1.243
3.	UBCF_Cosine	0.984
4.	UBCF_Pearson	0.978

From the above table , Model comparison based on varying recommendations UBCF with Pearson Correlation distance is the better model.

iii.K-Means Clustering

K-Means is an unsupervised learning algorithm, which is used for arranging set of data points in groups called clusters. Initially, centroids are chosen at random and are assigned to each cluster and proceeding the step and hence finding the initial mean value. The system assumes that the users belonging to the same cluster have similar ratings. Clustering is completed based on the preferences of users. Each product has been rated by the users. Based on the ratings labels are generated. K-Means algorithm performance is evaluated through root mean squared error[1].We obtain an RMSE value for K-Means is 2.079428.

iv.Random Forest

The proposed Random Forest algorithm aims to develop an efficient technique based on recommendation approach is compared with the existing K-Means and as well as the similarity methods i.e. Cosine Similarity and Pearson Correlation.

Random forest is most accurate 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. 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 generate an output class we take the mode of that classes as the output of random forest. In the method proposed, random forest 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 system. The system will label the users into its respective classes which has been learned during the training phase[6].

Table 2: Random Forest result

Mtry	RMSE	RSquare	MAE
1	0.287363852	09551108	0.1821579137
2	0.099003067	0.9925811	0.047627630
3	0.033208108	0.9989825	0.0119907029
4	0.003584302	0.9999813	0.0008099305

From the table, it shows that the random forest algorithm RMSE,RSquare,MAE values.

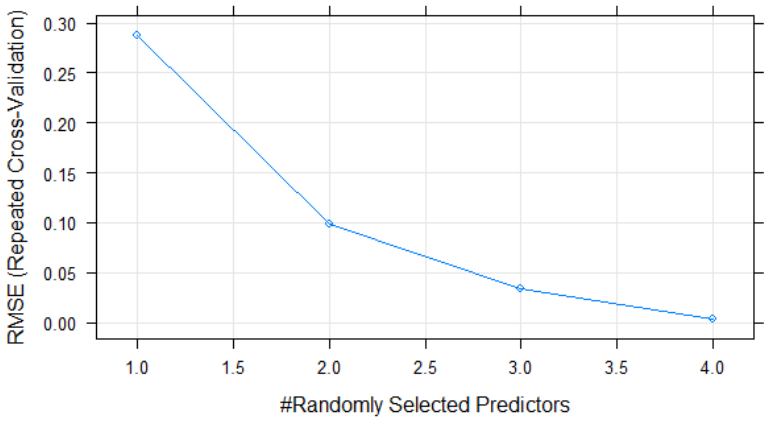

Figure 3: Random Forest Result

IV. DATASET

We accomplished with an efficient MovieLens dataset which is widely used in recommender systems. The MovieLens dataset use the version with 943 users and 1682 movies. The original Rate relation contains the movies rating with 5 measures. Each and every users have watched at least one movie, and the dataset consists of approximately 1,00,000 movies ratings. The dataset is divided into two parts: training set and testing test. Training set contains the trained matrix and testing set contains the actual rating of any user with respect to any item.

V. EXPERIMENTAL RESULTS

I. Performance metric: Root-mean-square error (RMSE)

The root-mean-square error (RMSE) is good accuracy measure to calculate prediction error rate. It evaluates the residuals between value of actual rating and predicted rating of an item by a user. RMSE is an aid to collect the magnitudes of the errors in predictions for various times into a single measure of predictive power.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(\text{P}_{\text{ru}} - \text{A}_{\text{ru}})^2} \quad \text{... (Eq.3)}$$

Where n is all products rating in test set, P_{ru} is the rating prediction of a product for user(u) by the RS, and A_{ru} is actual rating of the same product. The RMSE Error rate score is evaluated only for the movies which have been rated by the users.

II. Performance comparison

The proposed Random Forest method in this paper has been compared with similarity techniques(those are Cosine Similarity with UBCF and IBCF, Pearson Correlation Coefficient with UBCF and IBCF) and K-Means Clustering techniques are compared with each other based on RMSE metric and the same dataset used in all the methods. Through the method proposed in this paper there is an improvement in reducing an RMSE Error rate in Random Forest algorithm approaches which is most efficient amongst other Recommendation Systems (RS).

Table 3: Algorithms RMSE values

S.No.	Algorithms	RMSE
1.	IBCF_Cosine	1.443
2.	IBCF_Pearson	1.243
3.	UBCF_Cosine	0.984
4.	UBCF_Pearson	0.978
5.	K-Means	2.079
6.	Random Forest	0.287

From the table, it shows that the random forest had less rmse when compared to other algorithms where the item-based collaborative filtering with cosine similarity(IBCF_cosine) technique had more rmse value.

VI. CONCLUSION

In this paper, we proposed a random forest approach to model-based recommendation system. By comparing several recommendation approaches, how many similarity of users and similarly movies are available in the dataset is calculated, how different recommendations are generating and finally done evaluation part on recommendation system, which provides an efficient recommendation system based on effective random forest algorithm with minimum RMSE value.

VII. REFERENCES

[1] Shivani Sharma, "A Recommender System Based on Improvised K- Means Clustering Algorithm ", July 2018 ,International Journal of Research in Advent Technology, Vol.6, No.7, E-ISSN: 2321-9637.

[2] Gong, S. (2010): A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering. Journal of Software, vol. 5, pp. 245-252.

[3] N Lakshmipathi Anantha and Bhanu Prakash Bhattula, "A Review on recommendation system using rating dataset",2017, Volume 116 No. 5, 133-138.

[4] Xie, F.; Chen, Z.; Shang, J.; Huang, W.; Li, J. (2015): Item Similarity Learning Methods for Collaborative Filtering Recommender Systems. IEEE 29th International Conference on Advanced Information Networking and Applications, pp. 896-903.

[5] Raval, U. R.; Jani, C. (2015): Implementing and Improvisation of K-Means Clustering. International Journal of Computer Science and Mobile Computing, vol. 4, pp. 72-76.

[6] Ajesh A[1], Jayashree Nair[1], Jijin PS[1], "A Random Forest Approach for Rating–based Recommender System", 978-1-5090-2029- 4/16/$31.00 @2016 IEEE

[7] P. Resnick and H. R. Varian, "Recommender systems," Communications of the ACM, 1997, vol. 40, no. 3, pp. 56–58.

[8] Wang, Y.; Deng, J.; Gao, J.; Zhang, P. (2017): "A Hybrid User Similarity Model for Collaborative Filtering. Information Sciences".

[9] Isinkaye, F. O.; Folajimi, Y. O.; Ojokoh, B. A. (2015): "Recommendation Systems: Principles, Methods and Evaluation". Egyptian Informatics Journal, pp. 261273.

[10] Jia Rongfei[1]; Jin Maozhong[2];Liu Chao[3], " A New Clustering Method For Collaborative Filtering", 978-1-4244-7578-0/$26.00 @2010 IEEE.