International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 05 | May 2019 www.irjet.net p-ISSN: 2395-0072

Recommendation in E-Commerce using Collaborative Filtering

Daniya Rukiya[1], Tahani Africa[2], Shweta Nayak[3], Sayed Nooreain[4]

[1,2,3]*Anjuman Institute of Technology and Management, Bhatkal*
[4]*(Professor of Computer Science and Engineering, Dept. of Computer Science, AITM, Bhatkal)*

Abstract - *Collaborative filtering (CF) algorithms have been widely used to build recommender systems since they have distinguishing capability of sharing collective wisdoms and experiences. Recommendation algorithms typically work by suggesting items that are similar to the ones that a user likes, or items that similar users like. Most of the items in the recommendation list are already familiar to users. To address this issue, this paper performs clustering of stocks based on the transaction history of the product purchases and then divides the products into high, medium, low selling stock using K-Means clustering algorithm. Furthermore customers are divided into high, medium and low loyal customer using RFM (Recency, Frequency, Monetary) classification.*

Key Words: **Collaborative, Recommender, K-Means Clustering, RFM Classification, Recommender System, Collaborative Filtering.**

1. INTRODUCTION

To realize services that provide serendipity, this paper assesses the surprise of each user when presented recommendations. We propose a recommendation algorithm that focuses on the search time that, in the absence of any recommendation, each user would need to find a desirable and novel item by himself. Moreover, the degree of user's surprise is proportional to the estimated search time, we consider both innovators' preferences and trends for identifying items with long estimated search times. To predict which items the target user is likely to purchase in the near future, the candidate items, RFM algorithm weights each item that innovators have purchased and that reflect one or more current trends; it then lists them in order of decreasing weight. Moreover, normal users can hardly discover these items by themselves due to the limited time spent on online shopping. As a result, it is necessary to develop a recommender system that can discover popular items and less popular items. Since new items may have extremely short time-to-live, e.g., some garments appeared in newly released movies, it is necessary for the recommender system to be real-time, i.e., react rapidly. Besides, many of the less popular items in the long tail are extremely special which means they may only serve the interests of a small group of users. To fetch the attention from users and to discover their personalized needs, it is necessary to introduce serendipity into recommender systems..

1.1 Problem Statement

Recommendation in e-commerce has an urgent problem for users to choose demanded and interested items immediately and completely. This application will recommend the products to the customers by classifying the customers into high loyal customer, medium loyal customer and low loyal customer. Once they are classified, it then recommends the product based on loyalty. The products are categorized into high selling stock, medium selling stock and low selling stock. Whereas, high loyal customers are recommended medium selling stock and medium loyal customers are recommended low selling stock to gain the revenue in their environment.

1.2 Objectives

Recommenders system are the desire to improve users satisfaction and to increase economic success of the platform. In Ecommerce a recommender may either recommend a top recommendation based on the best price performance ratio for the customer but it may also show the products that are likely to lead to the highest revenue for the business.

2. METHODOLOGY

2.1 Architecture Overview

This architecture is mainly classified into 3 important modules - clustering of stocks, customer classification and products recommendation. The algorithms and methods used for these modules are as follows:

1. The user profile is created based on the activities performed by the user in the session.
2. Clustering Technique using Supervised K Means Machine Learning algorithm which is responsible for classification of stocks for the merchant.
3. RFM Classification algorithm which is responsible for classification of customers based on Low, Medium and High Loyal which will see the recency of transactions performed by the user.
4. Frequency of Purchase and monetary which is how much money customer will spend so that customers can be classified as Low Loyal, Medium Loyal and High Loyal.

5. Graph based page rank algorithm which is the best page where the Stock can be pushed to the customer which will depend on recursive weight function.

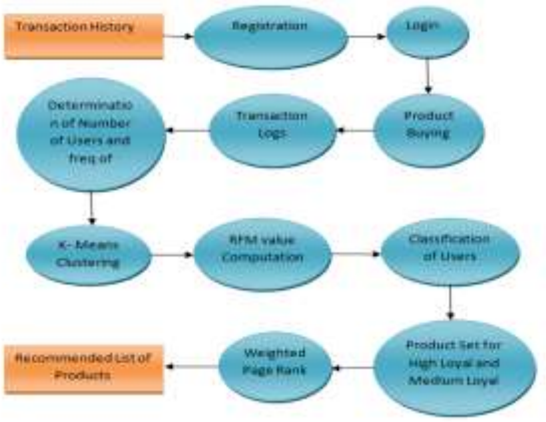


Fig -1: Architecture Overview

2.2 K-Means Clustering

The following are the steps used by K-Means Algorithm in order to classify the sales into high selling, medium selling and low selling stock.

1) Obtain the frequency set of products
2) Obtain the number of days set of the products
3) Plot the above 2 attributes.
4) Pick random 3 centers from the above plot.
5) Compute the distance between each data point to centers.
6) Assign the data point to a center which is having the lowest distance.
7) Repeat the process for all data points
8) Perform the computation for a set of iterations until optimization is achieved.

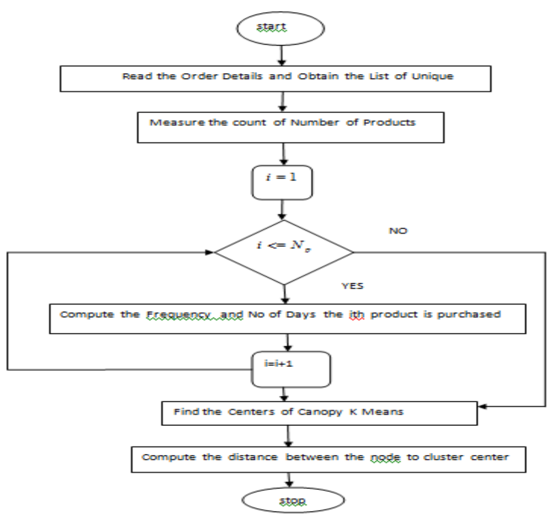


Fig -2: K-Means Algorithm Flowchart

2.3 RFM Classification

The following are the steps used by RFM Classification in order to classify the customers into high, medium and low loyal customer .

1. Find the unique customers from the order details.
2. Obtain the value of frequency for each of the customers.
3. Obtain the total money spend across all transactions by customer to calculate monetary.
4. Obtain the difference between current date and last date of purchase to get recency for each of the customers.
5. Sort the recency in descending order, Frequency in ascending order and monetary in ascending order and assign the weight values.
6. Combine the RFM value weights.
7. Sort the RFM in descending order.
8. Measure the number of rows of RFM matrix.
9. The users who are falling between 1 to C/3 are HIGH LOYAL
10. The users who are falling between C/3 +1 to C/2 are MEDIUM LOYAL
11. The users who are falling between C/2+1 to C are LOW LOYAL

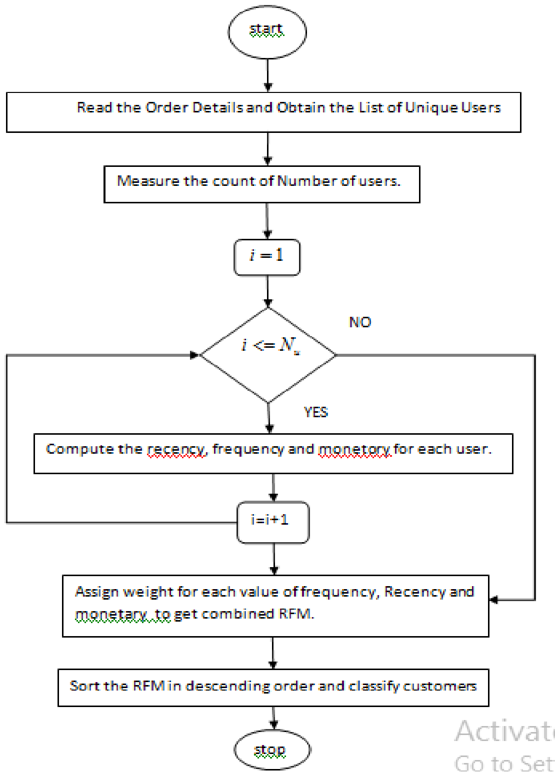


Fig -3: RFM Algorithm Flowchart

2.4 Weighted Page Rank Algorithm

This algorithm is divided into five main modules and they are Connectivity, Purity, UserId Page Frequency, Recursive Weight Computation and Weight Page per User.

2.4.1 Connectivity

1. Find the Number of Transactions Performed.
2. For each page find the Number of Products .
3. Find the transaction count for each products on a page and then add to get total transactions done on a page.
4. The connectivity of a page is defined as below,
$$connectvity = \frac{Number\ of\ transactions\ on\ Specific\ Page\ Product}{Total\ no\ of\ transactions}$$
5. Repeat the process for all pages.

2.4.2 Purity

1. Find the list of unique pages
2. For each of the page obtain the list of unique product ids
3. Only obtain those products on which transaction has been performed
4. For each of the product get the transaction count and obtain the summation of all transactions of product
5. The purity can be defined as below,

$$purity = \frac{Number\ of\ Products\ on\ which\ txn\ are\ performed}{Txn\ count\ on\ specific\ category}$$

2.4.3 User ID Page Frequency

1. Find the number of unique users except admin.
2. For each of the users compute page name, user id and frequency and construct the matrix.
3. On each page how many products purchased by specific user.

2.4.4 Recursive Weight Computation

1. Find the different category of products
2. For each category obtain the list of products
3. For each category obtain the transaction count across different products of a category
4. Perform the summation of transactions count to get total transaction of a category
5. Find the distinct amounts of transactions and get the matrix of amount and count of amount.
➢ Where, amount is the transaction cost and count of amount is number of times it appears in order log.

2.4.5 Weight Page per User

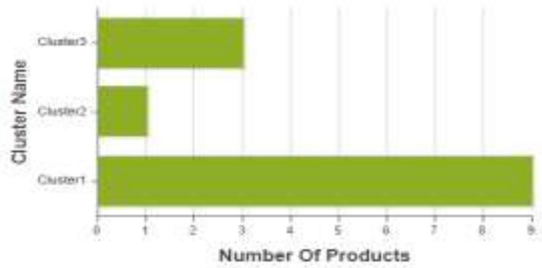
1. Obtain the Recursive Weight matrix
2. Obtain the distinct user names from UserId Page Matrix
3. For each of the pages obtain the following
   a) Connectivity
   b) Purity
   c) Recursive Weight
4. For each of the users – Obtain the number of times page is visited by the user
5. The total weight is computed using the following formula

$$Total\ Weight = N_{time\ visit} + purity + recursive\ weight + connectivity$$

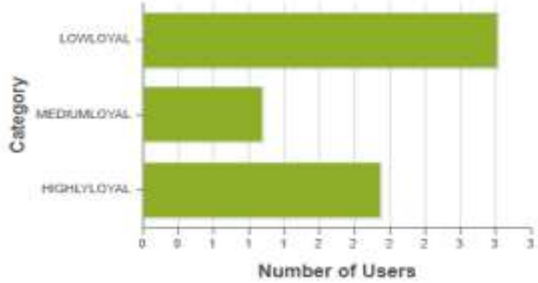
3. RESULT AND ANALYSIS

A recommender system built using a standard collaborative filtering method predicts preferences based only on the number of items purchased by the customers.

While this is useful and has proven to be largely successful, it may not consistently give high quality recommendations to the customers. Hence, in the present study, a list of items are first classified based on stocks as high selling stock, medium selling stock and low selling stock using K-Means clustering algorithm which is shown in **Graph -1** which then classifies customers based on their loyalty i.e. high loyal customer, medium loyal customer and low loyal customer using RFM classification algorithm as shown in **Graph -2**, it also calculates the amount of money spent by the customer on their transactions and also measures the days gap between their transactions which is generated using the standard collaborative filtering method.



Graph -1: K-Means Analysis Graph



Graph -2: RFM Analysis Graph

4. LIMITATIONS

Since our recommender engine is primarily based on collaborative filtering techniques, it suffers from the following problems:

1. **New User Problem**: When a new user is introduced, recommendations cannot be produced for them since the person may not have rated any products. This limitation could however be addressed using content-based filtering using parameters such as age, occupation, gender etc.,

2. **New Item Problem**: If an new item has not been rated, it will not participate in the recommendation list generation.

3. **Homogeneity of contents:** The in-house code is based on a single-content analysis which for the Movie Lens dataset is its genre. Because of the homogeneous nature of the contents, the present code is not amenable to using multiple contents.

4. **Lack of experimental validation**: A rigorous validation of the proposed algorithm is possible only through real-user experiments and the quality of the ratings obtained from these experiments. Such a validation will further improve the quality of the serendipitous items by identifying the bounds on the difference in the ratings to be used in our computational algorithm.

5. CONCLUSIONS

Recommendation algorithms provide an effective form of targeted marketing by creating a personalized shopping experience for each customer. The ecommerce system is using the entire modern way of shopping in the consumer market. There are lot of well established companies and new startups which provide the ecommerce services. Lot of products coming into the market and are sold online with good strategies.

The Product Buying is performed by the user and the log of product buying is maintained in the format of order details and order log. For large retailers, a good recommendation algorithm is scalable over very large customer bases and product catalogs, requires only sub second processing time to generate online recommendations, is able to react immediately to changes in a user's data, and makes compelling recommendations for all users regardless of the number of purchases and ratings. Unlike other algorithms, collaborative filtering is able to meet this challenge.

REFERENCES

[1] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," IEEE Internet Comput., vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[2] S. Deng et al., "A recommendation system to facilitate business process modeling," IEEE Trans. Cybern., vol. 47, no. 6, pp. 1380–1394, Jun. 2017.

[3] S. Pyo, E. Kim, and M. Kim, "LDA-based unified topic modeling for similar TV user grouping and TV program recommendation," IEEE Trans. Cybern., vol. 45, no. 8, pp. 1476–1490, Aug. 2015.

[4] Y. Zhu et al., "What to do next: Modeling user behaviors by time-LSTM," in Proc. 26th Int. Joint Conf. Artif. Intell., 2017, pp. 3602–3608.

[5] E. H.-C. Lu, W.-C. Lee, and V. S.-M. Tseng, "A framework for personal mobile commerce pattern mining and prediction," IEEE Trans. Knowl. Data Eng., vol. 24, no. 5, pp. 769–782, May 2012.

[6] M. Mao, J. Lu, G. Zhang, and J. Zhang, "Multirelational social recommendations via multigraph ranking," IEEE Trans. Cybern., vol. 47, no. 12, pp. 4049–4061, Dec. 2017.

[7] J. Bu et al., "Improving collaborative recommendation via user-item subgroups," IEEE Trans. Knowl. Data Eng., vol. 28, no. 9, pp. 2363–2375, Sep. 2016.

[8] Y. Zhu et al., "Heterogeneous hypergraph embedding for document recommendation," Neurocomputing, vol. 216, pp. 150–162, Dec. 2016.

[9] R. K. Merton et al., "The Matthew effect in science," Science, vol. 159, no. 3810, pp. 56–63, 1968.

[10] M. A. Hitt and C. Anderson, The Long Tail: Why the Future of Business Is Selling Less of More. New York, NY, USA: Hyperion, 2007.

[11] Y. C. Zhang, D. Ó. Séaghdha, D. Quercia, and T. Jambor, "Auralist: Introducing serendipity into music recommendation," in Proc. 5th Int. Conf. Web Search Web Data Min., 2012, pp. 13–22.

[12] M. Schedl and D. Hauger, "Tailoring music recommendations to user by considering diversity, mainstreaminess, and novelty," in Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, ACM, 2015, pp. 947–950.

[13] N. Kawamae, "Serendipitous recommendations via innovators," in Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, ACM, 2010, pp. 218–225.

[14] D. Cosley, S. Lawrence, and D. M. Pennock, "REFEREE: An open framework for practical testing of recommender systems using research index," in Proc. 28th Int. Conf. Very Large Data Bases, 2002, pp. 35–46.

[15] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in Proc. 14th Int. Conf. World Wide Web, ACM, 2005, pp. 22–32.

[16] H. Yu, Y. Wang, Y. Fan, S. Meng, and R. Huang, "Accuracy is not enough: Serendipity.

BIOGRAPHIES

DANIYA RUKIYA
Dept. of Computer Science Engineering,
AITM, Bhatkal.

TAHANI AFRICA
Dept. of Computer Science Engineering,
AITM, Bhatkal.

SHWETA NAYAK
Dept. of Computer Science Engineering,
AITM, Bhatkal.