Proceedings of the Fourth International Conference on Inventive Systems and Control (ICISC 2020)
IEEE Xplore Part Number: CFP20J06-ART; ISBN: 978-1-7281-2813-9

# Development of Product Recommendation Engine By Collaborative Filtering and Association Rule Mining Using Machine Learning Algorithms

Abhiraj Biswas, Kaza Sai Vineeth, Ayush Jain, Mohana
Department of Electronics &Telecommunication Engineering
RV College of Engineering, Bengaluru- 560059 and affiliated to
Visvesvaraya Technological University, Belagavi, Karnataka, India

*Abstract*— **Recommendation engines are a subclass of information filtering system that seeks to predict the 'rating' or 'preference' that user would give to an item. It finds information designs in the informational index by learning customers decisions and produces the results that co-identifies with their requirements. Real time examples like Amazon, have been using a recommendation engine for suggesting the goods or products that customers might also like. As the database used in this paper consists of large amount of data, it becomes a difficult and cumbersome process to provide viable choice of products for all the customers. The need of state of art recommendation engine is a necessity in real world e-commerce platforms to solve the issue and fulfil the customers' needs. There are numerous ways such as collaborative filtering, content-based filtering, hybrid filtering, etc. to build a recommendation system. This paper developed a product recommendation engine that uses collaborative filtering approach, which finds similarity between items bought by the customers with other customers, purchase pattern, and association rule mining framework. The recommendations were generated in order to facilitate 'cross-sell' across various items. The collaborative filtering (CF) approach produced the top 10 recommendations for each user. The association rule mining produced the rules based on minimum support (at 0.001) and minimum confidence (at 0.8) values. These values produced around 40,000 viable rules. It can be inferred that selection of metrics and the computation speed is important for quality recommendations.**

***Keywords—Association rule mining, Item-based CF, Collaborative filtering, User-based CF, Recommendation engine.***

## I. INTRODUCTION

The measure of data in this world is expanding more rapidly than our capacity to process it. Users regularly experience challenges in finding the substance they need rapidly, because of the vast measure of data accessible on the web. More often the user endeavors to look for help from other people who had recently had similar requirements for those things or pick up those things that are nearer to what they are searching for and this occasionally brought about serious data overload issue. If one can prescribe two or three things to clients contingent upon their prerequisites and interests, it will make a beneficial outcome on the client experience and lead to more visits [5]. In this way, organizations nowadays are building splendid and vigilant recommendation engines by looking into on the past transaction of their customers. A recommendation engine channels the information with the assistance of various calculations and prescribes the most applicable things to clients. It catches the past conduct of a client and relying upon that, prescribes items which the prone to purchase. Recommendation engines help organizations in executing balanced marketing methodologies, depending on client purchase history to uncover client inclinations and recognize items that clients may purchase. Traditionally, there are two methods to construct a recommender system: Collaborative filtering and content-based recommendation. In the principal technique, the recommender works with information that the client gives, either unequivocally (evaluating) or verifiably (tapping on a connection). In view of that information, a client profile is created, which is then used to make proposals to the client [3]. As the client gives more sources of information or takes activities on the suggestions, the engine turns out to be increasingly precise. In the second approach, recommendation is done based on users' past behavior. Based on different similarity measures the top similar items are selected and recommended to the desired user. This paper develops a product recommender engine which can generate quality recommendations for the customers. The methods that will be used with respect to the project are distance metric based collaborative filtering, association rule mining and matrix factorization based collaborative filtering.

An extensive overview on collaborative filtering recommendation techniques has been discussed in [1]. The concept of user-based CF and item-based CF is discussed in this paper, which serves as the building blocks for this paper. A vivid use of collaborative filtering in product recommendations is discussed in [2]. The different goals of a product recommendation engine are extensively explained and were taken as a reference to build the required model. The phases and steps involved in building a recommendation system are explained in [3]. The importance of feedback in information collection phase to build a user profile for existing and new users is highlighted and is taken as reference in this paper. A survey on the techniques involved in a collaborative filtering process and its various advantages in a variety of fields has been discussed in [4]. Different metric evaluations like mean absolute error and mean square error, which are used to measure the performance of the system, are explained. An extensive outlook on the use of correlation to find the similarity factor is given in [5]. This concept considered as a reference in order to generate the item similarity factor so that similar items can be recommended to the target user with the help of a suitable model. The concept of market basket analysis is very essential in determining the viable associations between the items. The concept of market basket analysis used for generation of frequent items sets is explained in [6]. The associations among items can be measured with the help of support, confidence and lift metrics. The important concepts such as lift, confidence and support are elaborated in paper [7].

## II. Fundamentals Recommendation Engine

A recommendation system or engine is a new age technology that recommends products to customers, by predicting the rating or preference a customer would give to a product. The feedbacks can be in the form of usual ratings that a customer can give to a product. If the ratings are not implicitly found alongside the products, then the customer's purchase history like the number of times the customer has bought or viewed the product can be taken as an explicit rating [3]. Recommendation frameworks have a few distinct employments. The most well-known use for a recommendation framework is positioning items by how much a client might want them. On the off chance that a client is perusing or looking for items, we need to demonstrate to them the items they might want most first in the list.

Recommendation frameworks can likewise be utilized to discover how comparative diverse items are to one another. In the event that items are fundamentally the same as one another, they may speak to similar clients. The bigger the quantity of appraised things that are accessible for a client, the simpler it is to make strong forecasts about the future behavior of the client [4]. Item comparability is particularly helpful in situations where lot about a specific client is not required. One can suggest comparative items, regardless of whether the client hasn't entered any of their own item audits yet. One can likewise utilize recommendation frameworks to make sense of if two unique clients are like one another. On the off chance that two clients have comparative inclinations for items, one can expect they have comparative interests.

Recommender systems area at the end of the day used by vendors to build for their benefit in revenue. By suggesting cautiously chosen things to clients, recommender systems convey pertinent things to the consideration of clients. In order to accomplish more extensive business-driven objective of expanding revenue, the regular operational and specialized objectives of recommender systems are applicability, novelty and expanding recommendation variety.

### *A. Collaborative Filtering Based Recommendation Engine*

The overwhelming measure of information requires systems for proficient data filtering. Collaborative filtering is a strategy for making automatic forecasts about the interests of a client by gathering inclinations or preferences' data from numerous clients. There are two variants of collaborative filtering: user-based collaborative filtering and item-based collaborative filtering. The standard strategy for collaborative filtering is known as nearest neighborhood calculation.

It has a $n \times m$ lattice of evaluations, with client $u_i$, $I = 1, \ldots n$ and thing $p_j$, $j=1, \ldots m$. Presently one needs to foresee the rating $r_{ij}$ if target client one didn't watch/rate a thing j. The procedure is to ascertain the similitudes between target client and every single other client, select the top X comparable clients, and take the weighted normal of appraisals from these X clients with similarities as weights [9] [10].

*User Based - Collaborative Filtering (UBCF)*

This type of collaborative filtering is a powerful method for recommending helpful items to clients by taking into consideration that a customer will probably incline toward the things favored by other similar customers. UBCF requires the certain rating scores of items, which are provided by users to calculate similarities between users. A user item matrix is constructed wherein all the items bought by a particular user is stored and filled in with ratings which are implicit or explicit. These ratings provide the information about how much the user likes the item. These ratings metric should always be normalized so as to scale down the values to a common scale [8] [11].

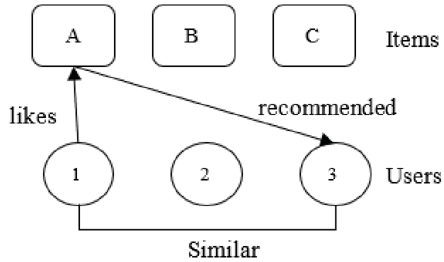


Fig. 1. User based collaborative filtering.

Many similarity measures and distance metrics can be used to find the similarities between users [5]. These metrics help in preparing the similarity matrix between the users which stores the similarity index between users. For instance, as found in, figure. 1, user 1 and user 3 have fundamentally the same as inclination conduct. If user 1 prefers Item A, UBCF can prescribe item A to user 3 due to the shared likeness that is existing between these two clients.

*Item Based Collaborative Filtering (IBCF)*

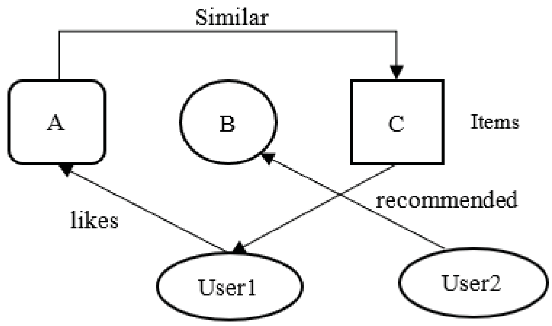


Fig. 2. Item based collaborative filtering

This method, predict items by considering the similarities between the items that each user has bought or visited. Based on the similarities between items and the purchase behavior of a user, IBCF can recommend quality products that the target user will prefer [1]. Similar techniques are used in this type of collaborative filtering for generating the similarity matrix, but here the similarity values will be for a pair of items. For example, in figure 2, there is a list of Items mentioned. User 1 likes Item A and purchases it. As Item A and Item C are like each other, IBCF system will recommend Item C to User A. The similarity index value between Item B and Item A is less than the value between Item A and Item C. This is the reason that Item C is taken as a more efficient and quality recommendation than Item B for User1.

### *B. Association Rule Mining Based Recommendation Engine*

Association rules helps to show the probability of associations between data things inside considerable instructive records in various types of databases. Association rule mining, at a fundamental measurement, incorporates the use of AI models to analyze data for models, or co-occasion, in a database. It recognizes visit if association, which are called affiliation rules. An association rule has two sections: a precursor (if) and an ensuing (then) [7]. A precursor is a thing found inside the data. An ensuing is a thing found in blend with the precursor. Association rules analysis is a strategy to reveal how things are related to one another. There are three basic approaches to quantify affiliation.

Measure 1: Support. This says how noticeable an item set is, as assessed by the degree of trades in which an item set appears.

Measure 2: Confidence. This says how likely thing Y is procured when thing X is purchased, imparted as {X - > Y}.

Measure 3: Lift. This says how likely thing Y is acquired when thing X is purchased, while controlling for how predominant thing Y is. [7]-[14].

## III. Design And Methodology Of Recommendation Engine

Recommender systems are basic for web-based organizations that offer a huge choice of items. Amazon, spotify, instagram, and netflix all utilization recommender frameworks to enable their online clients to comprehend the substantial volume of individual things – books, films, hardware, whatever – found in their substance indexes. Those recommender frameworks offer some benefit to clients by understanding an individual client's conduct and afterward prescribing to them things, they may discover helpful [7].

Association rules are mined from a lot of "exchanges". For community recommendation, we as a rule have clients' appraisals of articles. Instructions to change over evaluations to “exchanges" is controlled by which sort of affiliations and what number of dimensions of affiliations we need to find. Right off the bat, we are keen on anticipating if a client might want a thing. Thus, we map the evaluations for a thing into two classifications: like and abhorrence as indicated by whether the rating for the thing is more noteworthy than or not exactly, some edge esteem.

### *A. Item Based Collaborative Filtering*

Prior collaborative filtering frameworks dependent on rating comparability between clients (known as client collective separating) had a few issues:

- Frameworks performed inadequately when they had numerous items however relatively couple of evaluations.
- Figuring likenesses between all sets of clients was costly.
- Client profiles changed rapidly and the whole framework display must be recomputed

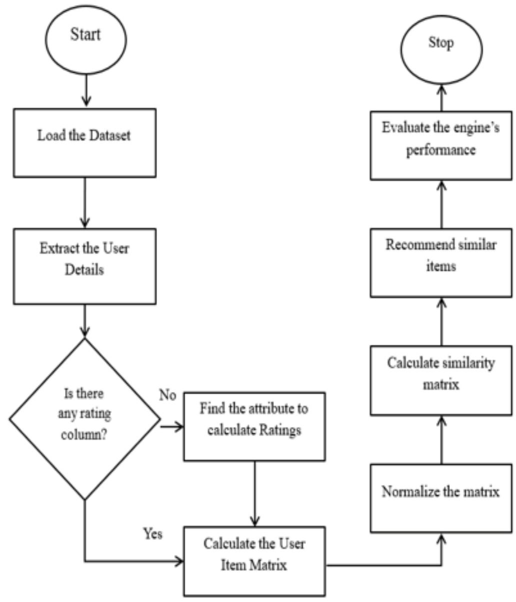


Fig. 3. Flowchart of Item based Collaborative Filtering

Item-item models settle these issues in frameworks that have large number of clients than items. Item-item models use rating conveyances per item, not per client. With a greater number of clients than items, every item will have a larger number of evaluations than every client, so an item's normal rating more often does not change rapidly. This prompts increasingly stable rating dispersions in the model. At the point when clients expend and afterward rate an item, that item's comparable items are picked from the current framework model and added to the client's proposals. This similarity measures can take numerous structures, for example, similarity between ratings or cosine of those rating vectors. As in client frameworks, comparability capacities can utilize standardized evaluations. Second, the framework executes a recommendation stage. It utilizes the comparative items to a client's as of now appraised items to produce a rundown of suggestions.

As shown in figure. 3, at the beginning the user item matrix is constructed with the help of number of times an item was bought by a user. After that, the similarity between the target user and the rest of the users is calculated. The select a subset of the users (neighborhood) according to their similarity with the target user. Finally find the top neighbors from the similarity matrix and generate appropriate recommendations for the target user. The similarity values between things are estimated by watching every one of the clients who have appraised both the things.

*Cosine similarity*

$$\text{sim(i, j)=cos(i, j)=}\frac{(i.j)}{|(|i|)| * |(|j|)|} \quad (1)$$

*Correlation-based similarity*

$$\text{sim (i, j)} = \frac{\sum_{u\epsilon U} Ru,i-\bar{R}i)(Ru,j-\bar{R}j)}{\sqrt{\sum_{u\epsilon U}(Ru,i-\bar{R}i)2}\sqrt{\sum_{u\epsilon U}(Ru,j-\bar{R}j)2}} \quad (2)$$

*Min-max normalization*

$$\left(\frac{v-\text{minA}}{\text{maxA}-\text{minA}}\right)(\text{new\_max}_A\text{-new\_min}_A)\text{+new\_min}_A \quad (3)$$

*Z-score normalization*

$$\frac{d-mean(P)}{std(P)} \quad (4)$$

## B. *Association rule mining*

Market basket analysis is a technique dependent on the hypothesis that if one purchases a specific group of items, it is more (or less) liable to purchase another group of items. In order to choose intriguing guidelines from the arrangement of every conceivable standard, limitations on different proportions of centrality and intrigue are utilized. The best-realized requirements are least limits on help and certainty. This enables the retailer to generate rules that are in line with the purchases of the consumer or clients. This also helps in forming baskets of items, which enables recommending cluster of items to the consumer.

Let X, Y be the item sets, $X \Rightarrow Y$ is an association rule and T is the set of transactions. The important factor that let us decide the important rules out of all the rules are:

*Support*

$$\text{supp(X)} = \frac{|\{t\epsilon\ T: X\epsilon t\}|}{|T|} \quad (5)$$

*Confidence*

$$\text{conf}(X \Rightarrow Y) = \frac{\text{supp(X U Y)}}{\text{supp(X)}} \quad (6)$$

*Lift:* The lift indicates how independent the items are with respect to each other when bought together.

$$\text{lift}(X \Rightarrow Y) = \frac{\text{supp(X U Y)}}{\text{supp(X)*supp(Y)}} \quad (7)$$

fig. 4 gives a brief outlook of the association rule-mining algorithm, which starts with obtaining every item in the transaction and calculating support for every item. The minimum support (min_supp) is determined to prune the item sets, which doesn't fulfill the conditions. The items which are more than the min_supp are grouped as the frequent item sets. The confidence of each item in every frequent item sets is calculated and checked with the minimum confidence (min_conf) value.

Following the association rule mining principle, all items whose confidence does not satisfy the condition, the item sets, and the items are pruned from the rules. The item sets which satisfies the condition make it to the frequent item sets [7].

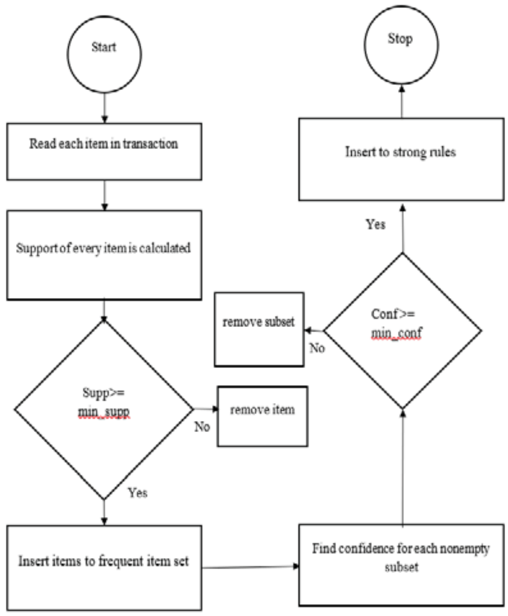


Fig. 4. Flowchart of Association Rule Mining

## IV. Results And Discussion

The user-based recommendations which used UBCF algorithm and the item-based recommendations which used IBCF algorithm are summed up in order to build a simple and an efficient product recommender engine. The results of this recommendation engine is shown and discussed in this section. The User-Item (UI) matrix that has been processed from the database is depicted in figure. 5. The columns of the UI matrix represent different users and the rows of the UI matrix represents different items present in the database. The UI matrix is filled with the count of how many times each user bought different items of the database. For example, user 4 bought item 6 '25' times, so it is filled with '25', whereas the same user bought item 5 '456' times and so it is filled with '456'.

The UI matrix depicts a relation between item and user and the fill-in value shows how strong is the item liked by the user. The Ui matrix is usually filled with rating values, but due to the lack of ratings column in the dataset, a separate attribute was a 'likeness' value. Another important factor that lies in the UI matrix is the normalization of the matrix. This matrix consists of a normalized value which serves as a similarity score. As discussed above, there are many kinds of normalization and it is done to scale down the ratings value to a nominal value between 0 and 1. This gives a general scale to compute the similarity score between the users. This count can be treated as the liking factor of each user for a particular item. Each user has a unique range of likeness for different items. In order to get the liking factor of all the users into a common range we use min-max normalization technique. Finally, the normalized User-Item matrix is shown in figure. 6. Using this UI matrix, we can generate the similarities between any two users using the cosine similarity algorithm.

| | User1 | User2 | User3 | User4 | User5 | User6 | User7 | User8 | User9 | User10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Item1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Item3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Item5 | 0 | 0 | 0 | 456 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item6 | 0 | 0 | 0 | 25 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item8 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Item9 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Item15 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Item16 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

Fig. 5. Item matrix user based (quantity of items purchased)

| | User1 | User2 | User3 | User4 | User5 | User6 | User7 | User8 | User9 | User10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Item1 | 0.002193 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00219 |
| Item3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item4 | 0 | 0 | 0.002193 | 0 | 0 | 0 | 0 | 0 | 0.002193 | 0 |
| Item5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item6 | 0 | 0 | 0 | 0.054825 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item8 | 0 | 0 | 0 | 0 | 0 | 0 | 0.002193 | 0 | 0 | 0 |
| Item9 | 0 | 0.010965 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Item14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.002193 | 0 | 0 |
| Item15 | 0 | 0 | 0 | 0 | 0.002193 | 0.002193 | 0 | 0 | 0 | 0 |

Fig. 6. Normalized User Item matrix

The Item-Item similarity (IIS) matrix that has been processed from the database is depicted in figure. 7. Both the columns and rows of the IIS matrix represent different items present in the database. The IIS matrix is filled with the similarity between itself and every unique item of the database. The similarity score between two items is calculated using the cosine and string similarity algorithm which is explained above. The similarity measures a give a sense of how the users are related to each other. The more the user-user score the more similar are the users. This gives a very important insight that the products can be recommended to each other. The diagonal elements of the IIS matrix are '1' as they represent the same item and the similarity score that they achieve is '1'. The long form of the tables helps in obtaining the knowledge that there are various items which have had already been bought by a particular user, and the engine must select only those items which are similar to the items and nit the ones which are already been bought by the user.

The various associations between items is calculate based on the confidence values and lift scores. The parameter specification was set at values of min_sup=0.001 and min_confidence=0.8. The total number of rules came around 49122. The analysis, which can be done by referring figure. 8. are:

- All the customers who bought 'Wobbly Chicken' also went for 'Metal'.
- All the customers who bought 'Wrap' also bought 'Billboard Fonts Design'.

| | 382100 | 382101 | 382102 | 382103 | 382104 | 382105 | 731004 | 731008 | 731009 | 731012 |
|---|---|---|---|---|---|---|---|---|---|---|
| 382100 | 1 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.00000000 | 0 | 0 | 0.00000000 | 0.00000000 |
| 382101 | 0 | 1.00000000 | 0 | 0.06039673 | 0.0000000 | 0.00000000 | 0 | 0 | 0.00000000 | 0.00000000 |
| 382102 | 0 | 0.00000000 | 1 | 0.00000000 | 0.0000000 | 0.00000000 | 0 | 0 | 0.00000000 | 0.00000000 |
| 382103 | 0 | 0.06039673 | 0 | 1.00000000 | 0.0000000 | 0.21960191 | 0 | 0 | 0.00000000 | 0.00000000 |
| 382104 | 0 | 0.00000000 | 0 | 0.00000000 | 1.0000000 | 0.22492022 | 0 | 0 | 0.00000000 | 0.00000000 |
| 382105 | 0 | 0.00000000 | 0 | 0.21960191 | 0.2249202 | 1.00000000 | 0 | 0 | 0.00000000 | 0.00000000 |
| 731004 | 0 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.00000000 | 1 | 0 | 0.00000000 | 0.00000000 |
| 731008 | 0 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.00000000 | 0 | 1 | 0.00000000 | 0.00000000 |
| 731009 | 0 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.00000000 | 0 | 0 | 1.00000000 | 0.00000000 |
| 731012 | 0 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.00000000 | 0 | 0 | 0.00000000 | 1.00000000 |
| 731015 | 0 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.00000000 | 0 | 0 | 0.00000000 | 0.08860695 |
| 738120 | 0 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.00000000 | 0 | 0 | 0.00000000 | 0.00000000 |
| 738125 | 0 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.00000000 | 0 | 0 | 0.00000000 | 0.00000000 |
| 738128 | 0 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.00000000 | 0 | 0 | 0.00000000 | 0.00000000 |
| 738150 | 0 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.00000000 | 0 | 0 | 0.00000000 | 0.00000000 |
| 738155 | 0 | 0.00000000 | 0 | 0.00000000 | 0.0000000 | 0.04965363 | 0 | 0 | 0.03641846 | 0.00000000 |

Fig. 7. Item-Item similarity matrix

Sometimes it becomes important to check the influence of an item on several other items. The rules generated in R has two parts in it i.e. lhs and rhs. The lhs describe the items for which we are checking the associations and the rhs define the items that are closely associated with those items. figure. 9. shows all the items that are closely relate with 'Metal' so that customers who made a transaction for those items can be recommended 'Metal' items. It also shows that the transactions which involve 'Metal' can be recommended 'Decorative' based on the strong confidence and lift score.

```
> inspect(association.rules[1:10])
     lhs                             rhs                              support     confidence  lift      count
[1]  {WOBBLY CHICKEN}             => {DECORATION}                     0.001261773 1.0000000   443.8200  28
[2]  {WOBBLY CHICKEN}             => {METAL}                          0.001261773 1.0000000   443.8200  28
[3]  {DECOUPAGE}                  => {GREETING CARD}                  0.001036456 1.0000000   389.3158  23
[4]  {BILLBOARD FONTS DESIGN}     => {WRAP}                           0.001306836 1.0000000   715.8387  29
[5]  {WRAP}                       => {BILLBOARD FONTS DESIGN}         0.001306836 0.9354839   715.8387  29
[6]  {ENAMEL PINK TEA CONTAINER}  => {ENAMEL PINK COFFEE CONTAINER}   0.001396963 0.8157895   385.1741  31
[7]  {WOBBLY RABBIT}              => {DECORATION}                     0.001532153 1.0000000   443.8200  34
[8]  {WOBBLY RABBIT}              => {METAL}                          0.001532153 1.0000000   443.8200  34
[9]  {ART LIGHTS}                 => {FUNK MONKEY}                    0.001712406 1.0000000   583.9737  38
[10] {FUNK MONKEY}                => {ART LIGHTS}                     0.001712406 1.0000000   583.9737  38
```

Fig. 8. Illustration of 10 rules based on the lift score

```
> inspect(metal.association.rules)
    lhs                              rhs     support     confidence lift   count
[1] {WOBBLY CHICKEN}              => {METAL} 0.001261773 1          443.82 28
[2] {WOBBLY RABBIT}               => {METAL} 0.001532153 1          443.82 34
[3] {DECORATION}                  => {METAL} 0.002253166 1          443.82 50
[4] {DECORATION,WOBBLY CHICKEN}   => {METAL} 0.001261773 1          443.82 28
[5] {DECORATION,WOBBLY RABBIT}    => {METAL} 0.001532153 1          443.82 34
```

```
> inspect(metal.association.rules)
    lhs        rhs          support     confidence lift   count
[1] {METAL} => {DECORATION} 0.002253166 1          443.82 50
>
```

Fig. 9. Associations generated for an item

## V. CONCLUSIONS AND FUTURE SCOPE

The top online retail companies in the world requires a prolific and efficient recommendation engine for the cause of profits. Different e-commerce websites use different algorithms to cater to the needs of the engines. The collaborative filtering and association rule mining approach helped in increasing the cross-sell of products. This facilitated customers to go through a catalogue of products, providing alternative choices. For the purpose of similarity scores, algorithms such as cosine similarity, jaccard similarity and pearson correlation used for the generation of efficient similarity scores. All the algorithms are used to generate user-based recommendations and item-based recommendations.

The code used for the development of the algorithm can be made faster, simpler and less complex upon doing research and trying to code in other programming languages. The ideology used for the recommendation depends on various attributes such as ratings, frequency of items bought, net amount spent, are present in the database and more innovative ideas such as matrix factorization, in order to make good and efficient recommendations which can help the cater provide accurate recommendations. The next big advancement that is being implemented in ecommerce platforms is the value aware recommendation system, which combines online commercial and miniaturized scale financial aspects into giving customized recommendations.

## References

[1] BardulSarwar et.al., "*Item-based Collaborative Filtering Recommendation Algorithms",* in proceedings of the *10th International Conference on World Wide Web*, Hong Kong, pp. 285-295, 2001.

[2] Ya-Yueh Shih et all., "*Product Recommendation Approaches: Collaborative Filtering via Customer Lifetime Value and Customer Demands*", *Journal of Expert Systems*, vol.35(1), pp. 350-360, 2008.

[3] F.O. Isinkaye et.al., *"Recommendation Systems: Principles, Methods and evaluation", Journal of Egyptian Informatics*, vol. 16(3), pp. 261-273, August 2015.

[4] Masupha Lerato et.al., *"A Survey of Recommender System Feedback Techniques, Comparison and Evaluation Metrics",* in *proceedings of IEEE International Conference on Computing, Communication and Security, Mauritius*, pp.1-4, December 2015.

[5] Charu C Agarwal et.al., *"Recommender Systems: The Textbook", International Publishing, Springer, Switzerland*, 1st edition, pp. 139-166, 2016.

[6] Yehuda Koren et.al., *"Matrix Factorization Techniques for Recommender Systems", IEEE transactions on Computer Society*, pp. 42-50, 2009.

[7] Lamis Hassanieh et al., *"Similarity measures for collaborative filtering recommender systems"*, proceedings of *IEEE Middle East and North Africa Communications Conference,* Lebanon, pp.1-5, April 2018.

[8] Wenming Ma et.al., *"Normalizing Item-Based Collaborative Filter Using Context-Aware Scaled Baseline Predictor", Journal of Mathematical Problems in Engineering, China*, vol.7(1), pp.1-9, April 2017.

[9] Krishna C V et.al., "A review of Artificial Intelligence methods for data science and data analytics: Applications and Research Challenges" *2nd International Conference on I-SMAC (IoT in social, mobile, Analytics and cloud), 2018.*

[10] Abhiraj Biswas et.al.,"Survey on Edge Computing - Key Technology in Retail Industry" *International Conference on Intelligent Computing and Control Systems (ICICCS 2019).*

[11] R. Sharma et al., *"Collaborative Filtering-based recommender system: Approaches and research challenges",* proceedings of 3rd conference on Computational Intelligence and Communication Technology, Ghaziabad, India, pp. 1-6, February 2017.

[12] Raj, Jennifer S. "A COMPREHENSIVE SURVEY ON THE COMPUTATIONAL INTELLIGENCE TECHNIQUES AND ITS APPLICATIONS." *Journal of ISMAC* 1, no. 03 (2019): 147-159.

[13] Charanjeet Kaur et.al, "*Association Rule Mining using Apriori Algorithm: A Survey", International Journal of Advanced Research in Computer Engineering & Technology*, vol. 2(2), pp. 2081-2084, June 2013.

[14] Andrej Trnka et.al., *"Market Basket Analysis with Data Mining Methods",* in the proceedings of *1st International Conference on Networking and Information Technology*, pp. 1-20, 2010.