MARKET BASKET ANALYSIS FOR RECOMMENDER SYSTEMS USING APRIORI ALGORITHM

Anirudh Reddy Kaveti(axk190081), Gautam Shreedhar Bhat(gxs160730), Kushagra Dar(kxd180025), Rohit Sreekumar(rxs176030)

The University of Texas at Dallas, Richardson, TX-75080, USA

ABSTRACT

Data mining has many e-Commerce applications. It is vital to find the important hidden patterns for better business applications in the e-commerce retail sector. In order to analyze these secluded business patterns, we use the apriori algorithm. The apriori algorithm is one of the most popular data mining approach to derive association rules by finding frequent itemsets from a transaction dataset. Finding frequent itemset is significant because of its combinatorial explosion. Once frequent itemsets are obtained, association rules with larger confidence can be generated. In this work, we developed a recommender system using apriori algorithm that benefits retailors and customers. The project is implemented using Python. We make use of Kaggle dataset to generate the association rules and the developed recommender system is tested on the same data.

Index Terms — Data Mining, retailor, customer, apriori algorithm, association rules, support, confidence, lift, itemsets.

1. INTRODUCTION

With increased competition now a days, every retailer either an online ecommerce site or a local super market are looking for better marketing strategies and investing a lot in customer retention [1]. These retailers stores enormous amount of customer information and their purchases in the form of transactions. This data can be used by retail industries to target their customers and make their business more profitable. Manually going through the data and using traditional data visualizations are labour intensive and tedious. Data mining is that field which helps in extracting the hidden patterns or predictive information from large information. Data mining algorithms can be used to predict future trends and customer behaviours [2-4].

One such algorithm that can be used to find customer buying habits or consumer behaviour is Apriori Algorithm. The name apriori algorithm as it uses ' prior ' knowledge of the properties set by frequent items. The algorithm was introduced by Rakesh Agrawal and Ramakrishnan Srikant in 1994. Apriori algorithm is a classic data mining algorithm which is used for mining frequent itemsets and relevant association rules. It is designed to operate on a database that includes several purchases, such as products bought in a store by customers. It is very important for efficient market basket analysis and it helps customers to more easily buy their items, which increases market sales [4]. This algorithm is used in several other industries like retail, entertainment, insurance. This algorithm has unique healthcare utility as it can help detect adverse drug reactions (ADRs) by generating association rules that show the combination of drugs and patient characteristics that may contribute to ADRs [5].

In this project, we took a real world dataset and extracted required knowledge. Applying apriori algorithm on this knowledge, we were able to find patterns in customer buying habits. These patterns can be used by online retailers to recommend their products to new customers and also help offline stores or supermarkets to arrange the products in such a way that customer gets all the products he wish to buy at one place, which in turn helps to make profitable business.

The remainder of this paper is organized as follows. In section 2, we discuss some of the previous and related work which use apriori algorithm. Section 3 describes the apriori algorithm. Section 4 describes the implemented algorithm. Analysis over method and experimental results are presented in section 5. Conclusion is in section 6.

2. RELATED WORK

Association rule mining works on two majors steps. The first one is to find all itemsets with ample supports. The second step is to generate association rules by combining these frequent itemsets [6 - 9]. The availability of the minimum support threshold and minimum confidence threshold values are assumed in the conventional association rules mining [10, 11]. However, these parameters are hard to be set without specific knowledge. Often users have difficulties in setting the support threshold to obtain the required results. Setting the support threshold too large sometimes result in producing a fewer number of rules or even no rules to conclude. Selection of smaller thresholds produce too many results for the users. This thereby increases the computational time and also for screening these rules. Therefore, algorithms need to be built to generate minimum support and minimum values of confidence depending on the datasets in the databases.

Researchers have developed association rule mining without support threshold [12]. However, there is normally another restriction such as similarity or confidence pruning. We note that, the coincidental itemset problem had not been directly considered by these aforementioned works. Some researches take the coincidental itemset problem into account, and an additional measure is proposed [13] that improves the support confidence framework. In our work, we make use of conventional technique.

3. APRIORI ALGORITHM

This section discusses the apriori algorithm with an example and fundamental definitions described in apriori algorithm.

3.1. DEFNITIONS

Association Rules:
Association Rules are patterns or cause- effect relations that are present in huge datasets, which indicate the probability of relation between two or more items.

These are denoted as $A \rightarrow B$, where A denotes an itemset or group of products whose presence indicate the presence of items in itemset B.
Example: If temperatures go down, sales of winter clothing increase.

Support:
Support indicates how many times a product appeared in the whole dataset or percentage of an association rule out of all the rules.

$$Support\ (A \rightarrow B) = \frac{(Number\ of\ transactions\ containing\ both\ A\ and\ B)}{(Total\ number\ of\ transactions\ in\ the\ database)} \quad (1)$$

Confidence:
$Confidence\ (A \rightarrow B)$ is the percentage of transactions which contain both A and B out of all the transactions containing A. Confidence also indicates how likely a pattern or association rule identified is going to happen.

$$Confidence\ (A \rightarrow B) = \frac{(Number\ of\ transactions\ containing\ both\ A\ and\ B)}{(Total\ number\ of\ transactions\ containing\ A)} \quad (2)$$

Lift:
Lift indicates how likely an item B is purchased when an items in itemset A are purchased and also taking into consideration the support of item B. This helps to identify those relations which exist only because both items A and B are frequently.

- A lift of 1 indicates there is no relation between A and B.
- A lift of greater than 1 indicate more probability of B being purchased when A is bought.
- A lift of less than 1 indicate less probability of B being purchased when A is bought.

$$Lift\ (A \rightarrow B) = \frac{No.of\ transactions\ containing\ A\ and\ B}{\{Total\ no.of\ transactions\ containing\ A \times \ Total\ No.of\ transactions\ containing\ B\}} \quad (3)$$

Conviction:
Conviction indicates the interest in the association rule. Its range is 0 to Infinity.
- Conviction of 1indicates no relation between the itemsets in the association rule.
- Higher conviction indicates higher interest in the rule.

$$Conviction\ (A \rightarrow B) = \frac{1 - support\ (B)}{1 - Confidence\ (A,B)} \quad (4)$$

3.2. APRIORI ALGORITHM

In this section, we briefly discuss the working of the apriori algorithm. The steps of finding the association rules is explained with an example.
Firstly, the minimum support and minimum confidence value is defined. The support of all single candidate itemsets is calculated by going through the entire database. These itemsets are denoted as C_1. After the calculation of C_1, all those itemsets with support less than minimum support value is removed to form new itemsets which are denoted by L_1. Later, for each itemset in L_1, combine it with every other itemset to form C_2. In order to generate L_2, all itemsets with support greater than minimum support value are selected. The steps are repeated L_k = Φ. Rules in L_{k-1} are the final association rules.
Apriori Algorithm is explained below with an example. Assume there are 9 transactions and the products $p1, p2, p3,$

Table 1: The set of orders and the products purchased in each order

Transactions/Order	Products Purchased
100	*p1,p2,p5*
101	*p3*
102	*p2,p4*
103	*p1,p3,p5*
104	*p4,p1*
105	*p4,p5*
106	*p1,p3,p5*
107	*p3,p2,p5*
108	*p4,p5*

$p4, p5$ are purchased individually or in combination as shown in the *Table 1*. For this data, let us assume minimum support count = 2, minimum confidence percentage = 60%.

Step - 1:
Using equation (1), calculate support of each item in the transaction table. This is called C_1 (Candidate Set)

Table 2: The support values for all the products

Item	Support
p1	4
p2	3
p3	4
p4	4
p5	6

Since all the products are having support value greater than minimum support value, we consider each product to derive association rules. Therefore, $L_1 = C_1$.

Step - 2:
By joining L_1, we generate Candidate set(C_2). Since there are 5 products we get 10 itemsets.

Table 3: Support values for the combination of itemsets

Item set	Support
{p1,p2}	1
{p1,p3}	2
{p1,p4}	1
{p1,p5}	3
{p2,p3}	1
{p2,p4}	1
{p2,p5}	2
{p3,p4}	0
{p3,p5}	3
{p4,p5}	2

Now we remove those itemsets with support less than minimum support count and also those itemsets whose subsets are not frequent itemsets are also removed.

Table 4: The support values which are above the minimum threshold. (L_2)

Item set	Support
{p1,p3}	2
{p1,p5}	3
{p2,p5}	2
{p3,p5}	3
{p4,p5}	2

The above table is called L_2
We have obtained frequent itemsets. Now we generate association rules from it and calculate confidence of each rule. Just for example, we consider the first two itemsets. i.e., $\{p1, p3\}$ and $\{p1, p5\}$.
The confidence values for each combination of the itemsets are calculated. For example, let us calculate for two combinations i.e., $\{p1, p2\}$ and $\{p1, p3\}$. This yields four confidence values.

$\{p1\} \rightarrow \{p3\}$:

$$Confidence = \frac{\text{Support(p1,p3)}}{\text{support(p1)}} = \frac{2}{4} \times 100 = 50\%$$

$\{p3\} \rightarrow \{p1\}$:

$$Confidence = \frac{\text{Support(p1,p3)}}{\text{support(p3)}} = \frac{2}{4} \times 100 = 50\%$$

$\{p1\} \rightarrow \{p5\}$:

$$Confidence = \frac{\text{Support(p1,p5)}}{\text{support(p1)}} = \frac{3}{4} \times 100 = 75\%$$

$\{p5\} \rightarrow \{p1\}$:

$$Confidence = \frac{\text{Support(p1,p5)}}{\text{support(p5)}} = \frac{3}{6} \times 100 = 50\%$$

Since the minimum confidence percentage is set to be 60%, we consider the third rule as a strong association rule. Therefore, if customers buy a product {p1}, 75% of times they also buy product {p5}.

4. ALGORITHM IMPLEMENTATION

The apriori algorithm is implemented in a similar way as explained in the Section 3. The support vales for the itemsets are calculated for the *Kaggle data.* Steps 1 and 2 are repeated to generate the following tables. The number of frequent itemsets are limited to two in order to generate the association rules. Using these association rules, the confidence values are calculated. All The association rules that are greater than the minimum confidence threshold displayed to the customer/ user. The flowchart of the implemented algorithm is shown in Figure 1.
In this project, we also consider the customer perspective while shopping. For this, simple questions/ commands are displayed on the screen as user interface. Firstly, the association rules are generated for the particular database using a set support threshold and confidence threshold. This acts as the back end of the system. The user is then asked to choose the first item. Based on the selected item, the recommender system provides a list of association rules i.e. the list of items that the customer can select based on the first item which was selected. Again, based on the second item which was selected, a new list of items are displayed on the users' screen. We have to note that, the new list just depends on the previous selection and not the combination of items one and two. This is because we consider making the association rules using only two itemsets. These steps are repeated until the user wishes to continue the shopping. At any stage of the shopping, the user can quit shopping. This can be done by pressing *'Q'*. Once the user quits, the shopping is completes and a *thank you* message is displayed. The user interface flowchart is shown in Figure 2.

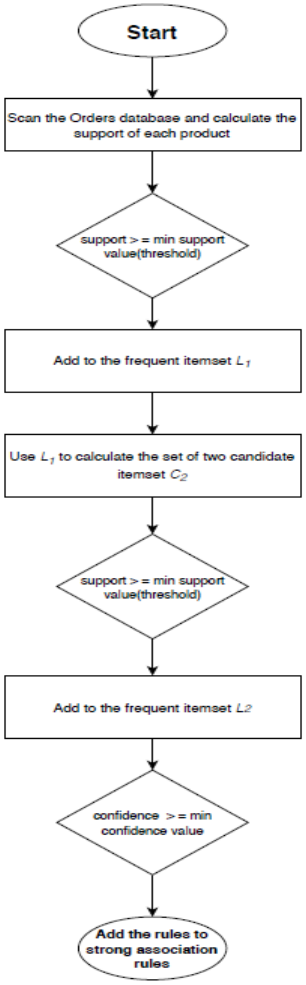

Fig. 1. Flowchart of the implemented algorithm

5. ANALYSIS AND RESULTS

In this section, the analysis and the results of the apriori algorithm used for the implemented recommended system is discussed.

The results of the apriori algorithm implemented for the Kaggle data that contains the transaction orders of the groceries is shown below. The results show the number of association rules that that we get by varying the support and the confidence values. The association rules here can also be seen as the number of items that can be recommended for the user. The Figure 3 shows the number of association rules for a set of minimum support threshold values with constant confidence. The confidence value is constant at 70%. From the figure, we can see that as the support threshold value increases, the number of association rules decrease. This is expected as the threshold of support increases, it is obvious to see less number of frequent itemsets combinations in the transactions. The Figure 4 shows the number of association rules for particular confidence threshold values. The support value is set to be constant at 1%. From the Figure 4, as

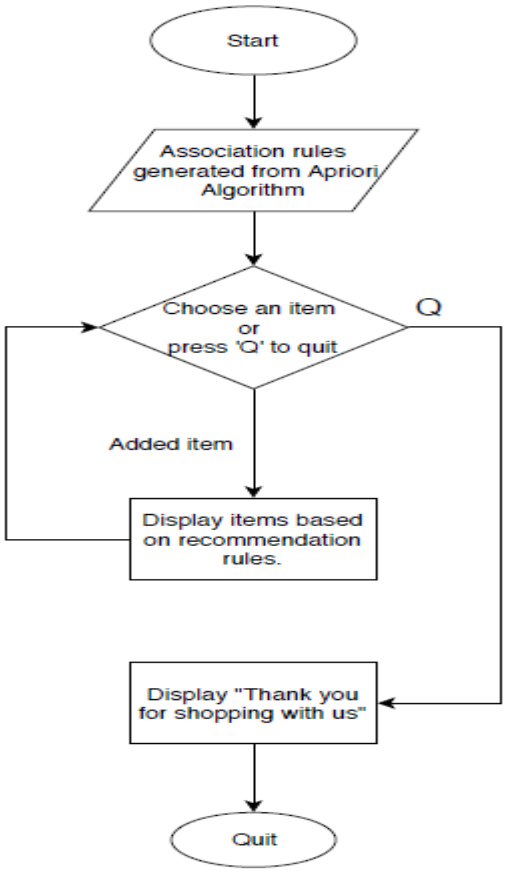

Fig. 2. Flowchart of the implemented User interface

expected, we can see that as the confidence threshold value increases, the number of association rules decrease. This is because, the association rules will have to meet higher confidence threshold values. Therefore, from figures 3 and 4, we can infer that the apriori algorithm is implemented properly.

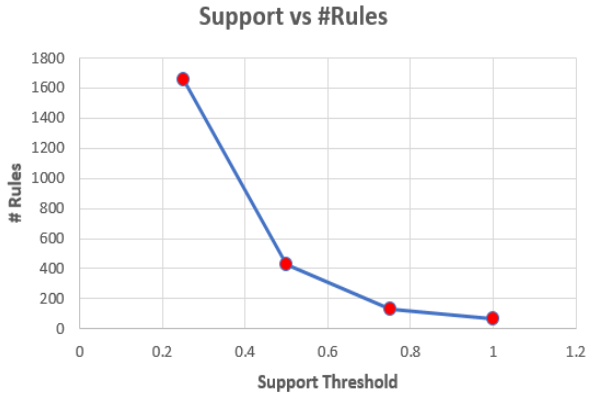

Fig. 3. The graph showing the support vs number of association rules implemented algorithm

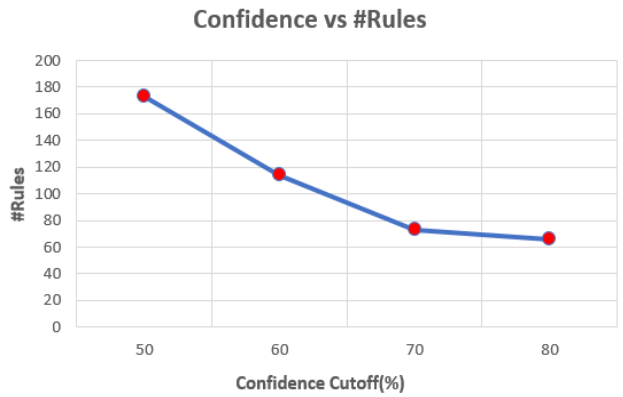

Fig. 4. The graph showing the confidence thresholds vs number of association rules implemented algorithm

Table 5: The numerical values of the graph shown in figure 3

Support Threshold	Number of Rules
0.25	1660
0.5	426
0.75	128
1	64

Table 6: The numerical values of the graph shown in figure 4

Confidence Threshold	Number of Rules
50	173
60	114
70	73
80	66

6. CONCLUSION

Recommendation system using apriori algorithm fetches critical information from the data for any business from its transaction-oriented database of customers. This system benefits the customers to find products they want to buy from the site. Conversely, they also help retailors by generating more sales and increasing their revenue. The method is provides association rules quickly making it computationally efficient. The method is implemented and the steps are shown graphically. The results of the implemented algorithm are analyzed.

7. ACKNOWLEDGMENT

We would like to thank Dr. Anurag Nagar for all his support and for teaching the fundamental topics of machine learning. We would also like to University of Texas at Dallas for all the support.

8. REFERENCES

[1] 'The 6 biggest challenges retailer Face today", www.onStepRetail.com. retrieved on June 20 II

[2] Berry, M. J. A. and Linoff, G. Data mining techniques for marketing, sales and customer support, USA: John Wiley and Sons,1997

[3] Fayyad, U. M; Piatetsky-Shapiro, G.; Smyth, P.; and Uthurusamy, R.1996. Advances in Knowledge Discovery and Data Mining. Menlo Park, Calif.: AAAI Press.

[4] Jiawei Han and Micheline Kamber (2006), Data Mining Concepts and Techniques, published by Morgan Kauffman, 2nd ed.

[5] digital vidya 2019. https://www.digitalvidya.com/blog/apriori-algorithms-in-data-mining/

[6] 2019.https://www.hackerearth.com/blog/developers/beginners-tutorial-apriori-algorithm-data-mining-r-implementation/

[7] Mahgoub, Hany, and D. Rösner. "Mining association rules from unstructured documents." In Proc. 3rd Int. Conf. on Knowledge Mining, ICKM, Prague, Czech Republic, pp. 167-172. 2006.

[8] Kannan S, Bhaskaran R. Association rule pruning based on interestingness measures with clustering. arXiv preprint arXiv:0912.1822. 2009 Dec 9.

[9] Ashrafi MZ, Taniar D, Smith K. A new approach of eliminating redundant association rules. InInternational Conference on Database and Expert Systems Applications 2004 Aug 30 (pp. 465-474). Springer, Berlin, Heidelberg.

[10] Tang P, Turkia MP. Parallelizing Frequent Itemset Mining with FP-Trees. InComputers and Their Applications 2006 Mar 23 (pp. 30-35).

[11] Ashrafi MZ, Taniar D, Smith K. Redundant association rules reduction techniques. International Journal of Business Intelligence and Data Mining. 2007 Jan 1;2(1):29-63.

[12] Dimitrijević M, Bošnjak Z, Subotica S. Discovering interesting association rules in the web log usage data. Interdisciplinary Journal of Information, Knowledge, and Management. 2010 Jan 1;5:191-207.

[13] Cheung YL, Fu AW. Mining frequent itemsets without support threshold: with and without item constraints. IEEE Transactions on Knowledge and Data Engineering. 2004 Jul 26;16(9):1052-69.