International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

Optimizing Business Revenue by Visualisation of Demanding Product's Sales data and Deriving Association Rules among the Products using Data Mining

Prof. Pradeep N. Fale[1], Narendra Moundekar*[2], Ritesh Saudagar[3], Prajwal Kamdi[4], Mrunali Rode[5], Janvi Borkar[6]
[1]Assistant Professor, Department of Information Technology, Priyadarshini College of Engineering, Nagpur, Maharashtra, India
[2-6]Department of Information Technology, Priyadarshini College of Engineering, Nagpur, Maharashtra, India

ABSTRACT

Article Info
Volume 8, Issue 2
Page Number : 134-142

Publication Issue :
March-April-2022

Article History
Accepted: 20 March 2022
Published: 30 March 2022

Because of the fierce competition in the market, everyone is busy with getting the maximum attention of people. For that producer must have products which satisfies the needs of customers. Huge scale research is going in this field. In such situations, customer requirements are very important. The value of a production plan can be modeled as a function that reflects the communication of the company with different agents, for example, customers and competitors. The issue concentrated in this system is to recognize the production plan with the maximum utility for a company, where expected number of the customers for the chosen products assesses the utility of a production plan in the plan. The solution is achieved using Apriory Algorithm in Data Mining.

Keywords: Production Plan Modeling, Apriori Algorithm, Data Mining

I. INTRODUCTION

In Today's era, the competition in the business is skyrocketing as the number of stores isincreasing day by day. Due to this competition, the retail traders should take all the necessary steps to increase the revenue of their business. Manually keeping track of all the revenue related data is almost impossible as the number of products in the store is growing exponentially. Consider a supermarket with a large collection of items. Typical business decisions that the management of the supermarket has to make includes, what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so-called basket data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club.

Several organizations have collected massive amounts of such data. These data sets are usually stored on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information.

This project introduces the problem of mining" a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence, and presents an efficient algorithm for this purpose. An example of such an association rule is the statement that 90% of transactions that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule. We Propose Apriori algorithm for finding the k-least products, which is also important for production plan.

II. AIM & OBJECTIVES

In market analysis, mining most demanding itemsets from a transaction database refers to the discovery of the itemsets, which frequently appear together in the transactions. However, the unit profits and purchased quantities of items are not considered in the framework of frequent itemset mining. Hence, it cannot satisfy the requirement of the user who is interested in discovering the itemsets with high sales profits. In view of this, utility mining emerges as an important topic in data mining for discovering the itemsets with high utility like profits.

The Objective of the proposed application is as follows:

- Developing a User Interface to select the Dataset.
- Calculating Confidence ratio.
- Determining k most demanding products with the highest expected number of the total customers.

III. RELATED WORK

In this area we exhibit a brief audit of the distinctive calculations, methods, ideas and methodologies that have been characterized in different research diaries and distributions. Agrawal, R., Imielinski, T., Swami, A. [1] proposed Frequent itemset mining calculation that uses the Apriori standard. Standard technique depends on Support-Confidence Model. Bolster measure is utilized. An anti-monotone property is utilized to diminish the inquiry space. It produces frequent itemsets and discovers affiliation governs between things in the database. It doesn't recognize the utility of an itemset [1]. Yao, H., Hamilton, H.J., Buzz, C.J. [2] proposed a system for high utility itemset mining. They sum up past work on itemset share measure [2]. This distinguishes two sorts of utilities for things, transaction utility and outside utility. They distinguished and dissected the issue of utility mining. Alongside the utility bound property and the bolster bound property. They characterized the numerical model of utility mining in view of these properties. The utility bound property of any itemset gives an upper bound on the utility estimation of any itemset. This utility bound property can be utilized as a heuristic measure for pruning itemsets as early stages that are not anticipated that would qualify as high utility itemsets [2]. Yao, H., Hamilton, H.J., Buzz, C.J. [3] proposed a calculation named Umining and another heuristic based calculation UminingH to discover high utility itemsets. They apply pruning techniques in view of the scientific properties of utility limitations. Calculations are more productive than any past utility based mining calculation. Liu, Y., Liao, W.K., Choudhary A. [4] proposed a two stage calculation to mine high utility itemsets. They utilized a transaction weighted utility (TWU) measure to prune the inquiry space. The calculations in light of the hopeful era and-test approach. The proposed calculation experiences poor execution when mining thick datasets and long examples much like the Apriori [1]. It requires least database examines, substantially less memory space and less computational cost. It can without much of a stretch handle extensive databases. Erwin, A., Gopalan, R.P., N.R. Achuthan [5] proposed an effective CTU-Mine Algorithm in view of Pattern Growth approach. They present a reduced information structure called as Compressed Transaction Utility tree (CTU-tree) for utility mining, and another calculation called CTU-Mine for mining high utility itemsets. They indicate CTU-Mine works more effectively than Two Phase for thick datasets and long example datasets. In the event that the limits are high, then Two Phase runs moderately quick contrasted with CTU-Mine, however when the utility limit gets to be lower, CTUMine beats Two-phase. Erwin, A., Gopalan, R.P., N.R. Achuthan [7] proposed a proficient calculation called CTU-PRO for utility mining utilizing the example development approach. They proposed another minimized information representation named Compressed Utility Pattern tree (CUP-tree) which develops the CFP-tree of [11] for utility mining. TWU measure is utilized for pruning the inquiry space yet it keeps away from a rescan of the database. They demonstrate CTU-PRO works more effectively than Two-phase and CTU-Mine on thick information sets. Proposed calculation is likewise more proficient on scanty datasets at low bolster thresholds. TWU measure is an overestimation of potential high utility itemsets, in this way requiring more memory space and more calculation when contrasted with the example development calculations. Erwin, R.P. Gopalan, and N.R. Achuthan [14] proposed a calculation called CTU-PROL for mining high utility itemsets from vast datasets. They utilized the example development approach [6]. The calculation first finds the extensive TWU things in the transaction database and if the dataset is little, it makes information structure called Compressed Utility Pattern Tree (CUP-Tree) for mining high utility itemsets. On the off chance that the information sets are too huge to be in any way held in fundamental memory, the calculation makes subdivisions utilizing parallel projections that can be along these lines mined autonomously. For every subdivision, a CUP-Tree is utilized to mine the total arrangement of high utility itemsets. The counter monotone property of TWU is utilized for pruning the hunt space of subdivisions in CTU-PROL, yet not at all like Two-phase of Liu et al. [4], CTU-PROL calculation keeps away from a rescan of the database to decide the real utility of high TWU itemsets. The execution of calculation is looked at against the Two-phase calculation in [4] furthermore with CTU-Mine in [5]. The outcomes demonstrate that CTU-PROL beats previous calculations on both scanty and thick datasets at most bolster levels for long and short examples.

In the second database examine, the calculation discovers all the two component transaction-weighted usage itemsets and it brings about three component transactions weighted use itemsets. The disadvantage of this calculation is that it experiences level astute hopeful era and test philosophy [18].

J Hu et al built up a calculation for frequent thing set mining that distinguish high utility thing mixes. The objective of this calculation is to discover sections of information, characterized through blends of a few things (rules), which fulfil certain conditions as a gathering and boost a predefined target work. The high utility example mining issue considered is not the same as previous methodologies, as itbehaviours control disclosure concerning singular traits and additionally regarding the general standard for the mined set, endeavouring to discover gatherings of such examples that together adds to the most to a predefined target work [19].

Y-C. Li, J-S. Yeh and C-C. Chang proposed a disengaged thing disposing of technique (IIDS). In this paper, they found high utility itemsets furthermore diminished the quantity of hopefuls in each database examine. They recovered productive high utility itemsets utilizing the mining calculation called FUM and DCG+. In this system they demonstrated a superior execution than all the past high utility example mining procedure. Nevertheless, their calculations still endure with the issue of level astute era and test issue of Apriori and it require various database filters [20].

IV. RESEARCH METHODOLOGY

In data mining, Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation).

The whole point of the algorithm (and data mining, in general) is to extract useful information from large amounts of data. For example, the information that a customer who purchases a keyboard also tends to buy a mouse at the same time is acquired from the association rule below:

Support: The percentage of task-relevant data transactions for which the pattern is true.

The formula for support with respect to two items i.e. for example keyboard to mouse will be:

$$\text{Support}=\frac{\text{No. of transacti ons containing both Keyboard and Mouse}}{\text{No. of total transacti ons}}$$

Confidence: The measure of certainty or trustworthiness associated with each discovered pattern. The formula for the confidence for example keyboard to mouse will be:

$$\text{Confidence}=\frac{\text{No. of transacti ons containing both Keyboard and Mouse}}{\text{No. of transacti ons containing (Keyboard)}}$$

The algorithm aims to find the rules, which satisfy both a minimum support threshold and a minimum confidence threshold (Strong Rules).

Item: article in the basket.

Itemset: a group of items purchased together in a single transaction.

Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps (DNA sequencing). Each transaction is seen as a set of items (an itemset). Given a threshold *C*, the Apriori algorithm identifies the item sets, which are subsets of at least *C* transactions in the database. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length ***k*** from item sets of length ***k-1***. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent ***k***-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The pseudo code for the algorithm is given below for a transaction database *T*, and a support threshold of ϵ. Usual set theoretic notation is employed; though note that *T* is a multiset. C_k is the candidate set for level ***k***. At each step, the algorithm is assumed to generate the candidate sets from the large item sets of the preceding level, heeding the downward closure lemma. ***Count[c]*** accesses a field of the data structure that represents candidate set *C*, which is initially assumed to be zero. Many details are omitted below, usually the most important part of the implementation is the data structure used for storing the candidate sets, and counting their frequencies.

Apriori(T, ϵ)
  $L_1 \leftarrow \{\text{large } 1 - \text{itemsets}\}$
  $k \leftarrow 2$
  while $L_{k-1} \neq \emptyset$
    $C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a\} - \{c \mid \{s \mid s \subseteq c \wedge |s| = k-1\} \nsubseteq L_{k-1}\}$
    for transactions $t \in T$
      $C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}$
      for candidates $c \in C_t$
        $count[c] \leftarrow count[c] + 1$
    $L_k \leftarrow \{c \mid c \in C_k \wedge \ count[c] \geq \epsilon\}$
    $k \leftarrow k + 1$
  return $\bigcup_k L_k$

Assume that a large supermarket tracks sales data by stock-keeping unit (SKU) for each item: each item, such as "butter" or "bread", is identified by a numerical SKU. The supermarket has a database of transactions where each transaction is a set of SKUs that were bought together.

V. SYSTEM DESIGN

A: Data Flow Diagram

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

DFD Level-0

Fig.1 2DFD Level-0

DFD Level-1

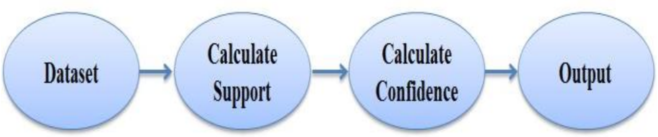

Fig.2 DFD Level-1

DFD Level-2

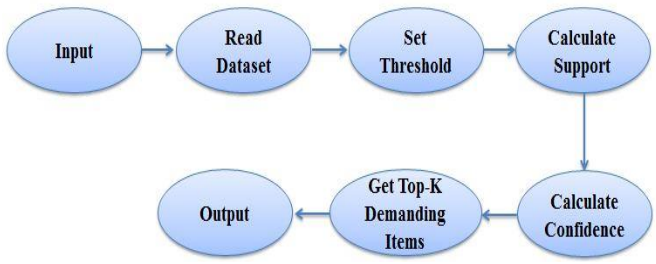

Fig.3 DFD Level-2

B: Use Case Diagram:

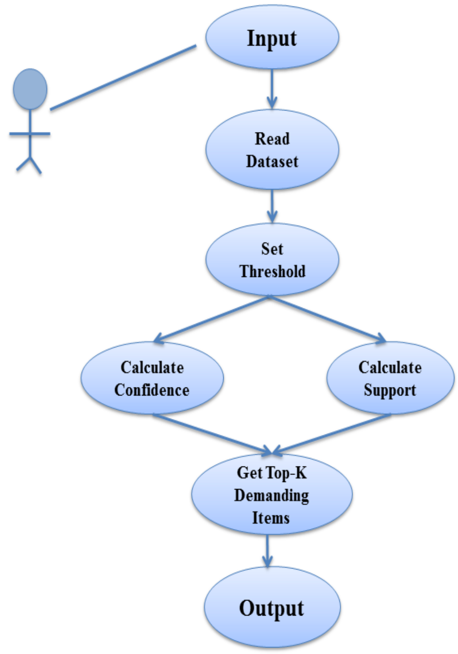

Fig.4 Use Case Diagram

VI. RESULTS AND DISCUSSIONS

A: RESULT

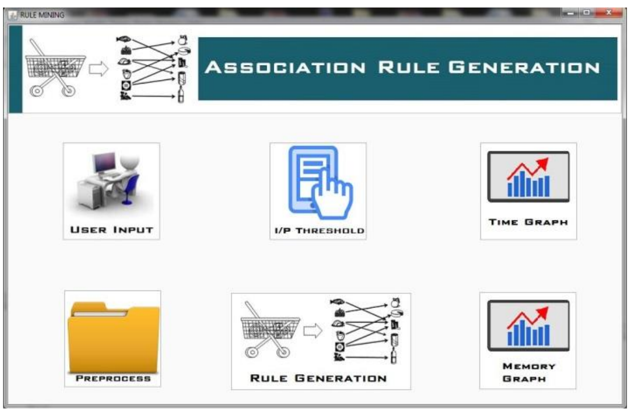

Fig5. Home Screen

Fig 6. Select Dataset

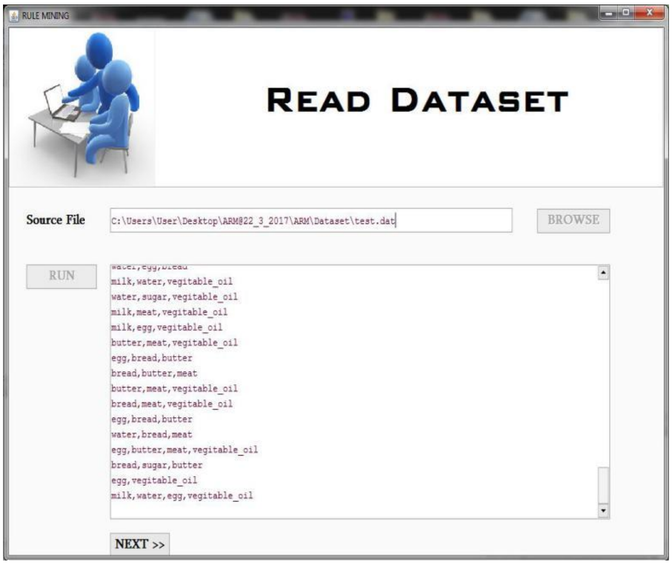

Fig 7. Read Dataset

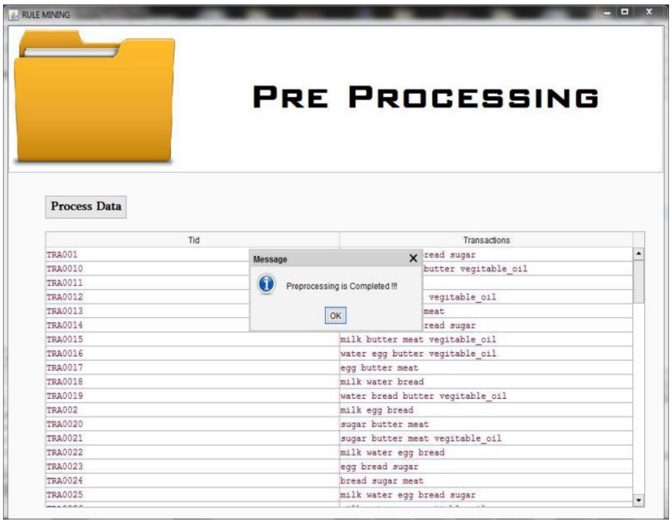

Fig 8. Data Pre-Processing

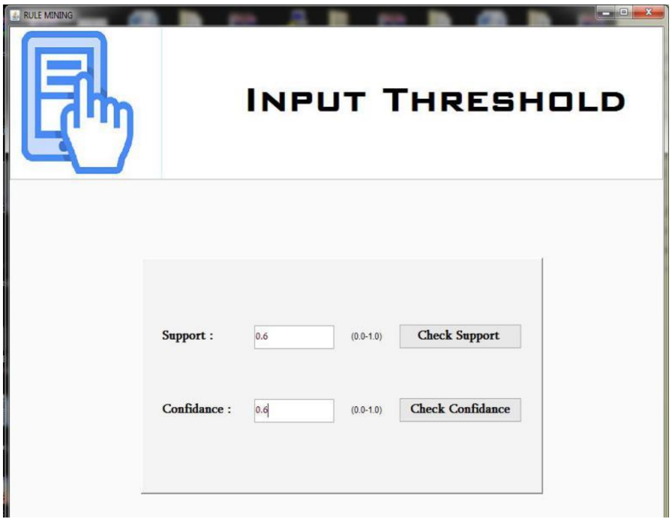

Fig 9. Input Thresholds

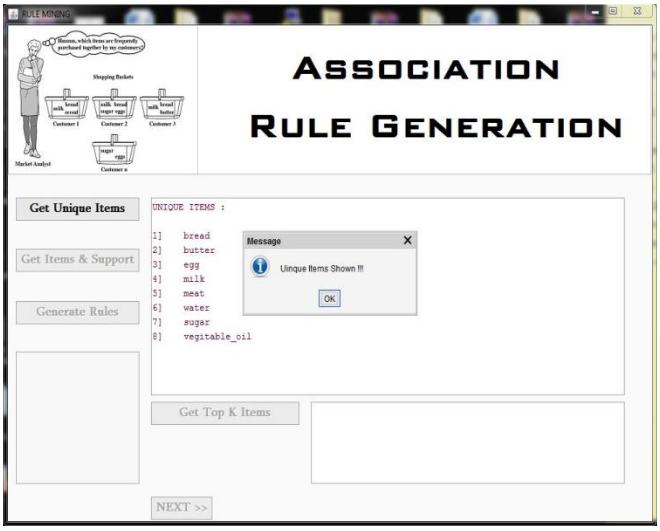

Fig 10. Unique Item Generation

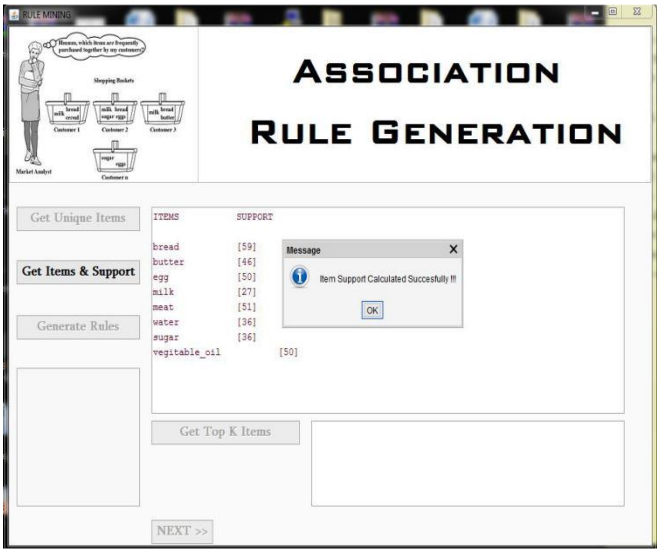

Fig 11. Calculate Support

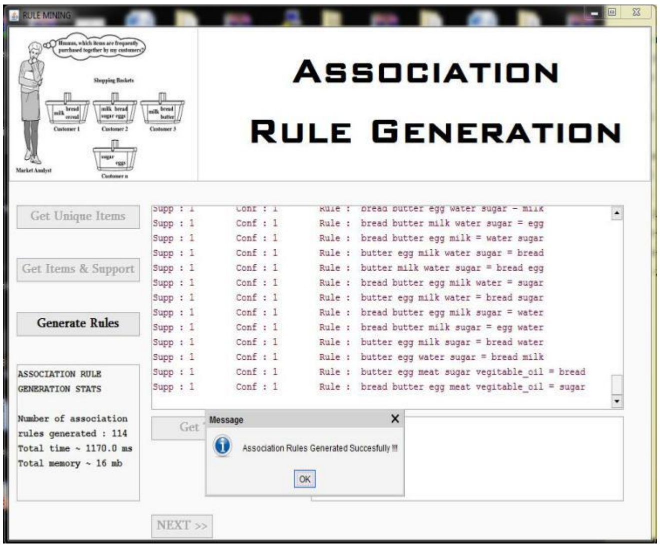

Fig 12. Generate Association Rule

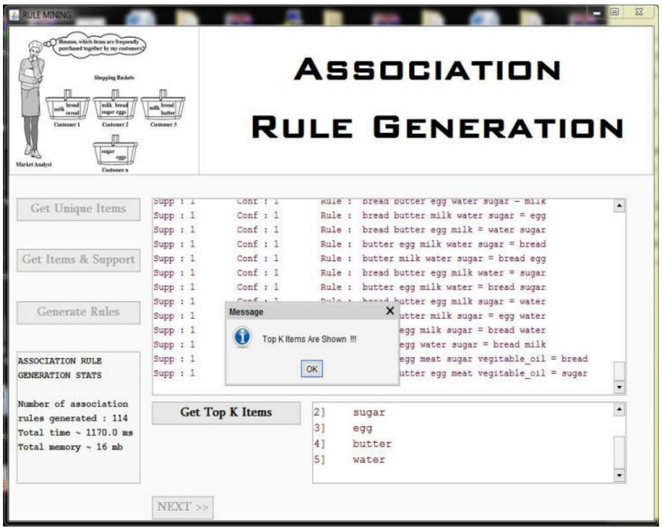

Fig 13. Generate Top-K Items

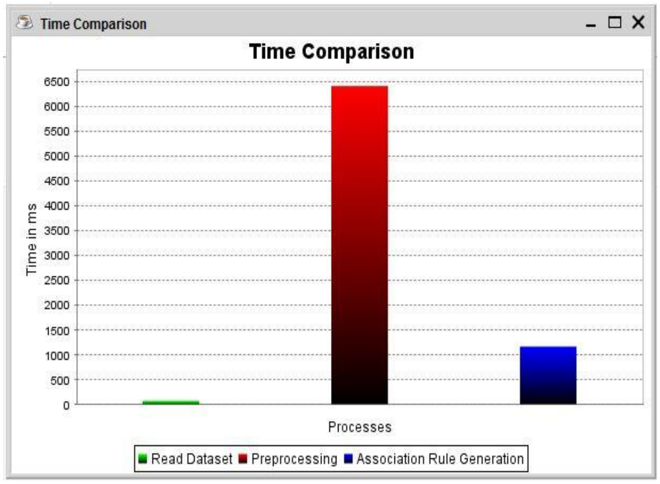

Fig 14. Time Comparison

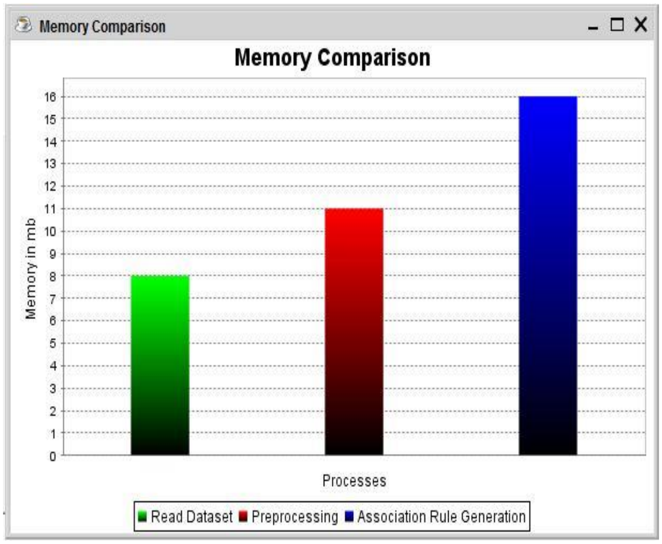

Fig 15. Memory Comparison

VII. CONCLUSION

We implemented Apriori algorithm for association rule mining so as to get the top k demanding items from the transactional dataset. The Apriori principle can reduce the number of itemsets we need to examine. Put simply, the Apriori principle states that if an itemset is infrequent, then all its subsets must also be infrequent. This means that if {beer} was found to be infrequent, we can expect {beer, pizza} to be equally or even more infrequent. So in consolidating the list of popular itemsets, we need not consider {beer, pizza}, nor any other itemset configuration that contains beer.Results show the implementation work and the results generated in terms of Time & Memory Consumption.

VIII. REFERENCES

[1]. Agrawal, R., Imielinski, T., Swami, A., "Mining Association Rules between Sets of Items in Large Database", In: ACM SIGMOD International Conference on Management of Data (1993).

[2]. Yao, H., Hamilton, H.J., Buzz, C. J., "A Foundational Approach to Mining Itemset Utilities from Databases", In: 4th SIAM

International Conference on Data Mining, Florida USA (2004).

[3]. Yao, H., Hamilton, H.J., "Mining itemset utilities from transaction databases", Data & Knowledge Engineering 59(3), 603–626 (2006).

[4]. Liu, Y., Liao, W.K., Choudhary, A., "A Fast High Utility Itemsets Mining Algorithm", In: 1st Workshop on Utility-Based Data Mining, Chicago Illinois (2005).

[5]. Erwin, A., Gopalan, R.P., N.R. Achuthan, "CTUMine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach", In: IEEE CIT 2007.Aizu Wakamatsu, Japan (2007).

[6]. Han, J., Wang, J., Yin, Y., "Mining frequent patterns without candidate generation", In: ACM SIGMOD International Conference on Management of Data (2000).

[7]. Erwin, A., Gopalan, R.P., Achuthan, N.R, "A Bottom-Up Projection Based Algorithm for Mining High Utility Itemsets", In: International Workshop on Integrating AI and Data Mining. Gold Coast, Australia (2007).

[8]. CUCIS. Center for Ultra-scale Computing and Information Security, North-western University.

[9]. Yao, H., Hamilton, H.J., Geng, L., "A Unified Framework for Utility Based Measures for Mining Itemsets", In: ACM SIGKDD 2nd Workshop on Utility-Based Data Mining (2006).

[10]. Pei, J., "Pattern Growth Methods for Frequent Pattern Mining", Simon Fraser University (2002).S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.

[11]. Sucahyo, Y.G., Gopalan, R.P., CT-PRO: "A Bottom- Up Non Recursive Frequent Itemset Mining Algorithm Using Compressed FP-Tree Data Structure", In: IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI), Brighton UK (2004).

[12]. G. Salton, Automatic Text Processing, Addison-Wesley Publishing, 1989.

[13]. J. Pei, J. Han, L.V.S. Lakshmanan, "Pushing convertible constraints in frequent itemset mining", Data Mining and Knowledge Discovery 8 (3) (2004) 227–252.

[14]. Bin Chen, Peter Hass, Peter Scheuermann, "A New Two-Phase Sampling Based Algorithm for Discovering Association Rules", SIGKDD '02 Edmonton, Albetra, Canada © 2002 ACM 1 58113 567 X/02/2007.

[15]. Ming-Yen lin, Tzer-Fu Tu, Sue-Chen Hsueh, "High utility pattern mining using the maximal itemset property and, lexicographic tree structures", Information Science 215(2012) 1-14.

[16]. Sudip Bhattacharya, DeeptyDubey, "High utility itemset mining, International Journal of Emerging Technology and advanced Engineering", ISSN 2250-2459, Volume 2, issue 8, August 2012.

[17]. Y.Liu, W.K. Liao and A. Choudhary, ―A two phase algorithm for fast discovery of high utility itemset‖, Cheng, D. and Liu. H. PAKDD, LNCS. PP: 689-695, 2005.

[18]. J.Hu, A. Mojsilovic, ―High utility pattern mining: A method for discovery of high utility itemssets‖, in: pattern recognition. PP: 3317-3324, 2007.

[19]. Y.-C. Li,j,-s. Yeh, and C.-C. Chang, ―Isolated Items Discarding Strategy for Discovering High Utility Itemsets,‖ Data and Knowledge engg., pp: 198-217, 2008.

[20]. Liu Jian-Ping, Wang Ying Fan-Ding, ‖Incremental Mining algorithm Pre-FP in Association Rule Based on FP-tree‖, Networking and Distributed Computing, International Conference, pp: 199-203, 2010.

[21]. A. Erwin, R.P. Gopalan, and N.R. Achuthan, "Efficient Mining of High Utility Itemsets from

Large Datasets", T. Washio et al. (Eds.): PAKDD 2008, LNAI 5012, pp. 554–561, 2008. © Springer- Verlag Berlin Heidelberg 2008.

Cite this article as :

Prof. Pradeep N. Fale , Narendra Moundekar, Ritesh Saudagar, Prajwal Kamdi, Mrunali Rode, Janvi Borkar , "Optimizing Business Revenue by Visualisation of Demanding Product’s Sales data and Deriving Association Rules among the Products using Data Mining", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 2, pp. 134-142, March-April 2022.
Journal URL : https://ijsrcseit.com/CSEIT228224