TEST Engineering & Management

*January-February 2020*
*ISSN: 0193-4120 Page No. 12663 - 12672*

# Performance Comparison of Data Science Algorithms for Finding Association Rules

**Dr.J.Chenni Kumaran,**Department of Information Technology, Panimalar Institute of Technology, Chennai, Tamilnadu, INDIA
**Dr.V.Suresh Kumar,**Principal, Ammini College of Engineering, Palakkad, Kerala,INDIA.
**Mr.G.DineshKumar,**Full Stack Developer, Strakin Technologies, Bangalore, Karnataka, INDIA.

***Article Info***
***Volume 82***
***Page Number: 12663 - 12672***
***Publication Issue:***
***January-February 2020***

***Article History***
***Article Received****: 18 May 2019*
***Revised****: 14 July 2019*
***Accepted****: 22 December 2019*
***Publication****: 24 February 2020*

***Abstract:***
One of the foremost difficult areas of knowledge science is a way to effectively realize the frequent itemsets and association rules among the itemsets from large data sets. Many algorithms are there for locating a frequent pattern from large knowledge sets. Apriori algorithm is that the most traditional algorithm for mining association rule and finding frequent patterns from immense knowledge sets. In Apriori algorithm, massive numbers of candidate itemsets are generated, increase in records within the info leads to too several input/output outlay and it leads to multiple scanning of database. As a result execution time is hyperbolic. During this Paper, in conjunction with Apriori algorithm, Eclat algorithm additionally used for distinctive and projected the frequent itemsets and association rules among the info sets from massive info in a good manner. Eclat algorithm uses vertical info format. There's no compelled to scan the info to seek out the support count. Execution time to seek out the frequent itemsets and association rules between the info things is attenuated that the performance of the algorithm is hyperbolic. Performance of Apriori and Eclat algorithms are evaluated victimization execution time.

***Keywords:*** *Apriori algorithm , Eclat algorithm, Frequent Pattern Mining, Association Rule Mining, Data Importing, Preprocessing, Eclat Association Rule, join, prune, itemsets.*

## I. INTRODUCTION

In recent years, knowledge Science is that the most beneficial and wide used method for the exploration and analysis of huge amount of knowledge to amass valid, novel, doubtless helpful and intelligent patterns hidden in knowledge within the areas like finance, banking, retail sales, production, population study, employment, observance of human or machines etc., have ways in which to record renowned data however cannot tackle the uncertainties of the longer term because of lack of tools to use this known information.

The tremendous growth in information and databases has led to the requirement for brand spanking new techniques and tools which will show intelligence rework data into helpful data and knowledge. In business, designing and estimates of future values are vital. The commodities trade wants prediction for foretelling of provide, sales, and demand for production designing, sales, selling and monetary choices.

With this increase in web of things and connected device, users are currently accessing such a lot of information and in conjunction with it a rise got to manage and perceive data. Information Science applications are totally different from developing normal applications. Rather than writing that solves the particular issues, the info Science algorithms are able to absorb the data then engineered their own logic supported that data. The scope of this paper is to seek out the frequent patterns and association rules between the info things in most ordinarily used is Apriori algorithm. In Apriori algorithm, massive numbers of candidate itemsets are generated, increase in records within the info leads to too several input/output defrayment and it leads to multiple scanning of database [7]. As a result

January-February 2020
ISSN: 0193-4120 Page No. 12663 - 12672

execution time is exaggerated and performance is shrunken. In conjunction with Apriori algorithm, Eclat algorithm are conjointly used for locating frequent itemsets and association rules among the info sets from massive info in an efficient manner [9]. Performance of Apriori algorithm and Eclat algorithm are evaluated victimization execution time and analysis done.

*(A)Frequent Pattern Mining*

Frequent patterns are patterns that seem oft-times in an exceedingly information set [8]. Finding frequent patterns plays an important role in mining associations, correlations, and lots of alternative attention-grabbing relationships among information. As an example, a group of things, like paste and brush, which seem oft-times along in an exceedingly dealing information set, could be a frequent itemsets. Consider, the progressively fierce competition is offered within the field of retail trade. Every and each company has quite more than 5000 stores, 1,000,000 transactions each hour, 10,000 and additional merchandise are on the market and serve quite one million customers, to create enhancements, and obtain profit within the business, some techniques need to be used.

*(B)Association Rule Mining*

Association rule mining may be a technique which will be wont to realize rules associative between combination merchandise [2]. As an example, however possible it's for a client to shop for product A and products B at identical time. The approach is by exploitation major functions of knowledge science to work out the frequent patterns and association rules between the itemsets.

## II. RELATED WORK

Using Eclat algorithm, the frequent patterns and association rules between the info things are known [1]. Eclat rule is one in every of the info science algorithm. Knowledge Science is associate knowledge domain field that includes computing, arithmetic, statistics and domain data. knowledge sscience may be a method to try to do the analysis like frequent pattern mining, Association rule mining, classification, clustering, regression etc. to seek out the intelligent patterns hidden within the large quantity of data-sets [2].

Using information Science, a lot of correct results are obtained. Eclat algorithmic rule is quicker compared to Apriori algorithm. It uses vertical information format. There's no compelled to scan the information to search out the support count. Execution time to search out the frequent itemsets and association rules between the info things are reduced that is the performance of the algorithmic rule is exaggerated [3].

## III. PROPOSED METHODOLOGY

The design diagram is outlined with the flow of information, that is refined and used for locating frequent patterns and association rule between the itemsets. The below figure Fig. 1 shows the design diagram of this paper.

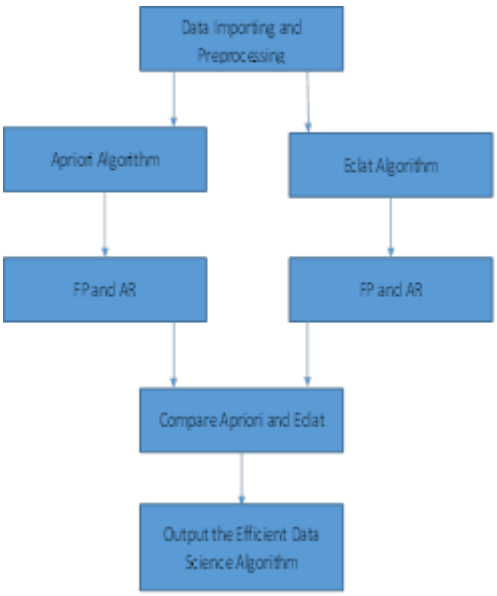



Fig. 1. Design Diagram

In information commerce, information ought to be loaded in to the R atmosphere for analysis. In information pre-processing, the collected data ought to be born-again into intelligible format. Standardization is that the technique employed to rework the varied format of knowledge into the common format and min-max technique is used for

January-February 2020
ISSN: 0193-4120 Page No. 12663 - 12672

normalization of data values. It's not necessary to carry all the attributes for doing the analysis, we are able to hold solely the attributes that affects the analysis. The missing values issues ought to be solved by straightforward applied mathematics techniques. The pre-processed information is given as an input to Apriori algorithm and Eclat algorithm. These algorithms generate the frequent patterns and association between the itemsets. The output is pictured by a graph. Finally, the performance of Apriori and Eclat algorithms are matched on the premise of execution time.

### *A. Data Importing and Preprocessing*

Data is obtainable in any file format like .txt, .csv, .xlsx, .spss etc. Information ought to be loaded in to R atmosphere for analysis. Once information is extracted from the file it ought to be keeping in a data frame. Information pre-processing is that the data processing technique that involves remodeling data into perceivable format. The fresh data is extremely at risk of noise, missing values, and inconsistency. Real world data is commonly incomplete, inconsistent, and is probably going to contain errors.

Data pre-processing is that the proved technique for breakdown such problems. In order to improve the quality of the data consequently, the mining results of raw data is preprocessed so the efficiency process improved. It is not necessary to hold all the attributes for doing the analysis; we can hold only the attributes which is affecting the analysis. The missing values downside need to be resolved by easy applied math techniques.

Data standardization is that the method by that similar information is collected in varied formats is reworked to a standard format that enhances the comparison process, permits for cooperative analysis and enormous scale analytics. Formula used for standardization is X-mean(X) / Standard Deviation(X).

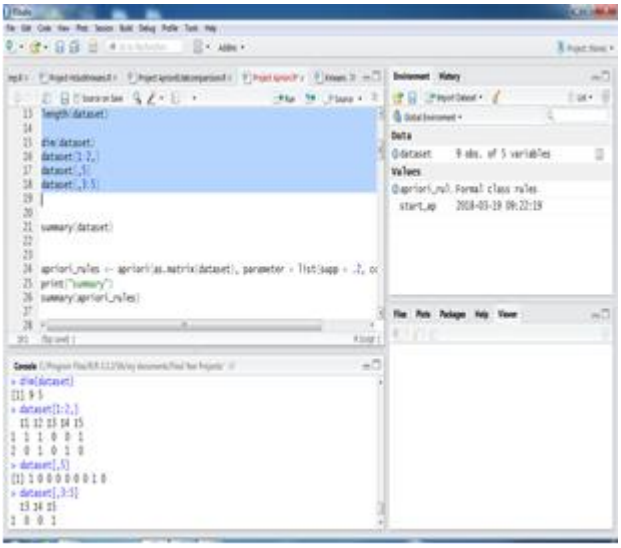


Fig. 2. Screen Shot Of Data Importing And Data Pre-Processing

The above Figure Fig. 2 shows the screen shot of data importing and data pre-processing work.

### *B. APRIORI ALGORITHM*

Apriori is an algorithm proposed by R.Agrawal and R.Srikant in 1994 for digging continuous itemsets for boolean association rules [1]. The name of the algorithmic rule relies on the actual fact that the algorithm uses previous information of frequent itemsets properties. Apriori algorithm uses associate unvaried approach referred to as level-wise search, wherever k-itemsets are accustomed get (k+1) - itemsets. Initially a set of frequent 1-itemsets is found by scanning the database for the occurrences of count of each item and collecting those items that satisfies the minimum support count.

The ensuing set L1 is employed to search out L2, the set of frequent 2-itemsets that is employed to search out L3, and so on, till no additional frequent k-itemsets may be found. Each Lk needs one full scan of the info to seek out itemsets and to boost the potency of the level-wise generation of frequent itemsets, a crucial property referred to as the Apriori property used for reducing the search area [6].

Apriori property: All non-empty subsets of a frequent item set should even be frequent.

There are two-steps to seek out the frequent itemsets:

January-February 2020
ISSN: 0193-4120 Page No. 12663 - 12672

a) The join step: To find Lk a set of candidate k-itemsets Ck, is generated by joining Lk-1 with itself [5].

b) The prune step - A scan of the database to determine the count of each candidate in Ck result in the determination of Lk, all candidates having a count not less than the minimum support count are frequent and therefore belong to Lk. To reduce the size of Ck, the Apriori property is used as any (K-1)-itemsets that is not frequent cannot be a subset of a frequent k-itemsets [5].

If any (K-1)-subset of a candidate k-itemsets is not in Lk-1, then the candidate cannot be frequent either and so can be removed from Ck. Once we have retrieved all frequent itemsets in the database, it generates the association rule satisfied the confidence threshold (min_conf) from frequent itemsets. DB is a database of transactions taken as input and output the frequent itemsets available in the database.

The below Figure Fig. 3. Shows the steps of Apriori Algorithm.

```
L1 = find_frequent 1-itemsets (DB);
for (k = 2; Lk-1 ? ¢; k++)
{
Ck = apriori_cangen(Lk-1);
for each transaction t € DB
{
Ct = subset(Ck,t);
for each candidate c € Ct
c.count++;
}
Lk = {c € Ck[c.count >= min_sup]}
}
return L = UkLk;

procedure apriori_cangen(Lk-1 : frequent(k-1)-
itemsetss)
for each itemsets l1 € Lk-1
for each itemsets l2 € Lk-1
if (l1[1] = l2[1]^ l1[2] = l2[2]) ^...^ (l1[k-2] =
l2[k-2]^ l1[k-1] = l2[k-1])
then {
c= Join l1 and l2;
if has_infrequent_subset(c, Lk-1) then
delete c;
else add c to Ck;
}
return Ck;

procedure has_infrequent_subset(c; Lk-1);
for each .(k-1)-subset s of c
if s∈Lk-1 then
return TRUE;
return FALSE;
```

Fig. 3. Apriori Algorithm

Initially Apriori finds the frequent 1 itemsets, L1 then Lk-1 is used to generate candidates Ck to find Lk for k>= 2. The apriori_cangen procedure generates the candidate itemsets using join procedure by joining Lk-1 with Lk-1 then pruning applies the Apriori property to eliminate the itemsets that is having subset which is not frequent.

January-February 2020
ISSN: 0193-4120 Page No. 12663 - 12672

Once all of the candidates have been generated, the database is scanned, the count for each of these candidates is accumulated and all the candidates satisfying the minimum support count form the set of frequent itemsets, L. Association rules from the frequent itemsets have to be generated [4]. The test for infrequent subsets is shown in procedure has_infrequent_subset.

Table 1.1 Transaction Database for Apriori Algorithm

| TID | List of Items |
|---|---|
| T1 | I1, I2, I4 |
| T2 | I2, I4 |
| T3 | I3, I4 |
| T4 | I1, I2, I5 |
| T5 | I1, I4 |
| T6 | I1, I3, I5 |
| T7 | I1, I2 ,I3, I5 |

The transaction databases D with seven transactions are shown in Table 1.1. The transaction T1 contains the itemsets of I1, I2, I3 and transaction T2 contains the itemsets of I2, I4. In this way all the seven transactions have its own itemsets.

Apriori algorithm is used for the generation of frequent 1 itemsets from the databases D is shown in the figure Fig. 4. In the first iteration of the algorithm, the entire available item in the database is a member of the candidate 1-itemsets, C1.

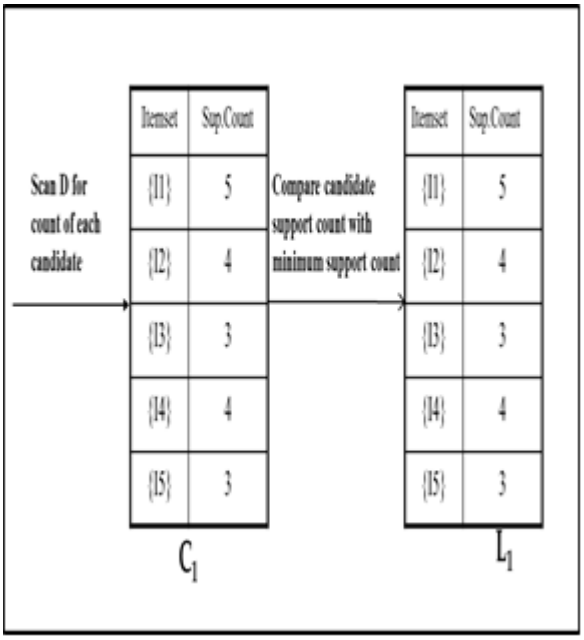
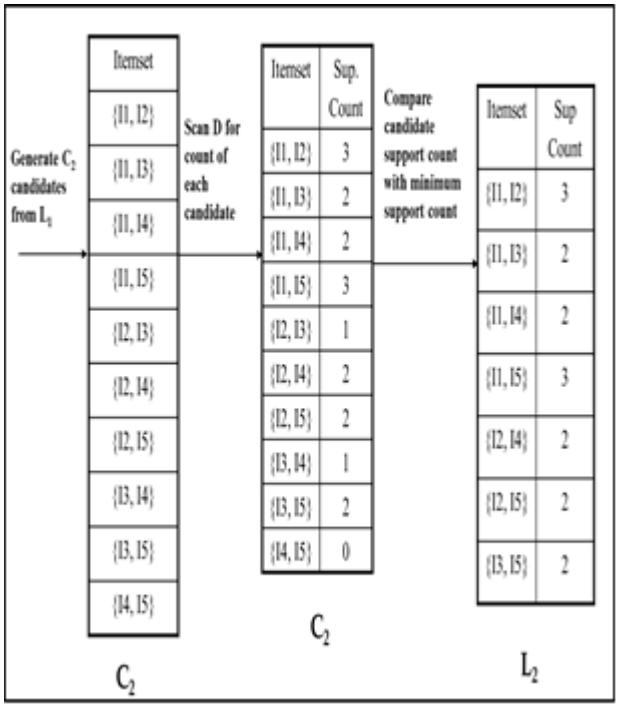



Fig. 4. Generation of Frequent 1- Itemsets in Apriori Algorithm

The algorithm counts the number of occurrences of each item by scanning all the transactions. The minimum support count is 2, frequent 1-itemsets, L1, can be determined by all of the candidates in C1 satisfy minimum support. Figure Fig. 5 shows how Apriori algorithm is used for the generation of frequent 2 itemsets from the databases D.

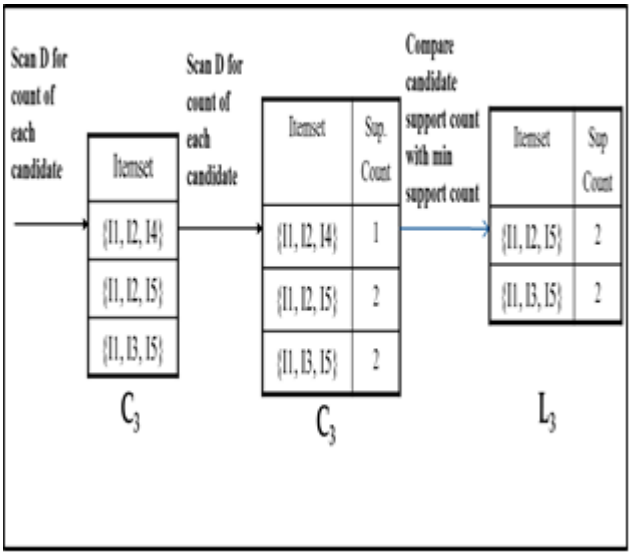
Fig. 5. Generation Frequent 2- Itemsets in Apriori Algorithm

January-February 2020
ISSN: 0193-4120 Page No. 12663 - 12672



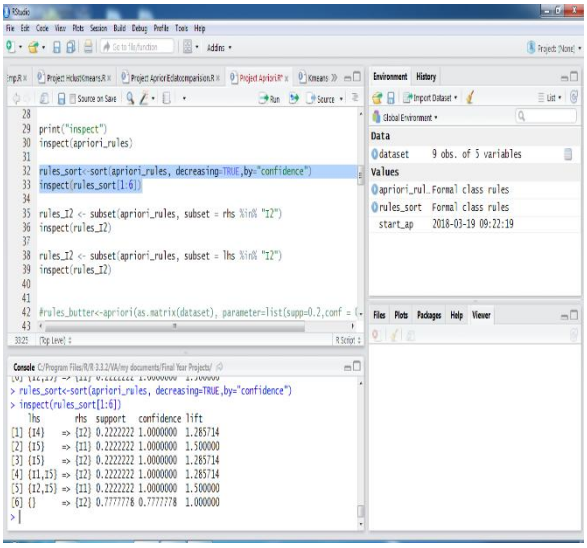

The candidate 2 itemsets are generated by joining L1 with L1 and during pruning no candidates are removed from C2 because each subset of the candidates set is also frequent. Database is scanned for generating the support count of each candidate itemsets in C2. The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate 2-itemsets in C2 having minimum support.

Fig. 6. Generation of Frequent 3- Itemsets in Apriori Algorithm




Frequent 3 itemsets from database D is generated using Apriori algorithm is shown in Figure Fig. 6. The candidate 3 itemsets are generated by joining L2 with L2 and apply pruning operation on that candidate 3 itemsets. The transactions in D are scanned to determine L3, consisting of those candidate3-itemsets in C3 having minimum support. L3 is having 2 sets of 3 itemsets which satisfies the minimum support count are {I1,I2,I5} and {I1,I3,I5}.

Fig. 7. Screen Shot Of Apriori Algorithm execution output



The above Figure Fig. 7. Shows the execution output of Apriori Algorithm.

Frequent 4 itemsets from database D is generated by joining L3 with L3 to obtain candidate 4 itemsets and results in {I1, I2, I3, I5} and it is pruned because its subset is not frequent. Thus C4 = ¢ and algorithm terminates. Let I = $\{i_1,i_2...i_m\}$ be a set of items. Let D the task-relevant data, be a set of database transactions where each transaction T is a set of items such that $T \subseteq I$.

January-February 2020
ISSN: 0193-4120 Page No. 12663 - 12672

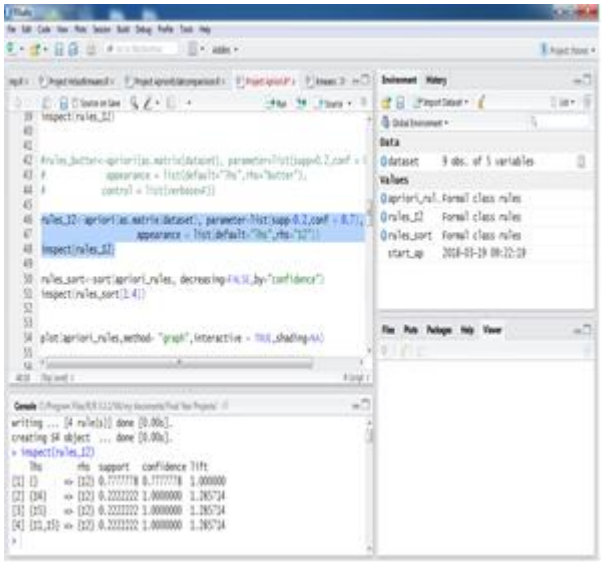


Fig. 8. Screen Shot of the Sorting the Association Rule

The on top of Figure Fig. 8. Shows the screen shot of the sorting the Association Rule .

An association rule is an implication of the form A=>B, where A⊂I, B⊂I and A∩B=Φ. The rule A=>B holds in the transaction set D with support s, where s is the percentage of transactions in D that contain AUB. This is taken to be the probability, P(AUB). The rule A=>B has confidence c in the transaction set D, where c is the percentage of transactions in D containing A that also contain B. This is taken to be the conditional probability, P(A|B).

The support of an itemsets is the percentage of transactions in the DB in which the itemsets appears.

A =>B

Supp(A => B) = P(A U B)

Supp(A U B) = No. of tuples containing both A and B /

Total no. of tuples

Confidence is defined as the measure of certainty or trustworthiness associated with each discovered Pattern. This signifies the purchase of item B, whenever item A is purchased.

A =>B

Conf(A => B) = P(B | A)

Conf(A => B) = No. of tuples containing both A and B /

No. of tuples containing A

Rules that satisfy both a minimum support threshold (min_sup) and a minimum confidence threshold (min_conf) are called strong rules. The strong association rule generated for the table1 transaction is:-

I2 ∧ I5 => I1

I3 ∧ I5 => I1

I1 ∧ I3 => I5

The above association rule specifies that in a sales environment, the customer who is getting I2 and I5 will also get I1 then the customer who is buying I3 and I5 will also buy I1 and the customer who is purchasing I1 and I3 will also purchase I5. In this logic, Apriori algorithm will generates frequent patterns and association rules among the dataitems.

The below Figure Fig.9 shows the Graph of Apriori Association Rule.

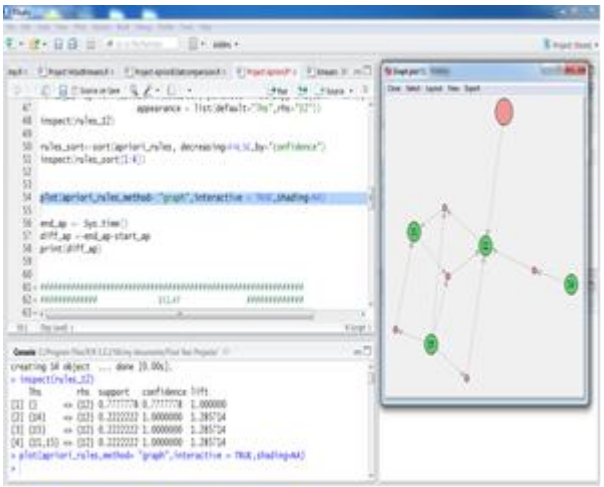


Fig. 9. Screen Shot of Apriori Association Rule Graph

### *C. Eclat Algorithm*

The Eclat algorithm is used to perform itemsets mining. ECLAT is Equivalence Class Transformation. Itemsets mining, finding the frequent patterns in data like if a consumer buys milk, he also buys bread. This type of pattern is called association rules and is used in many application domains [10].

January-February 2020
ISSN: 0193-4120 Page No. 12663 - 12672

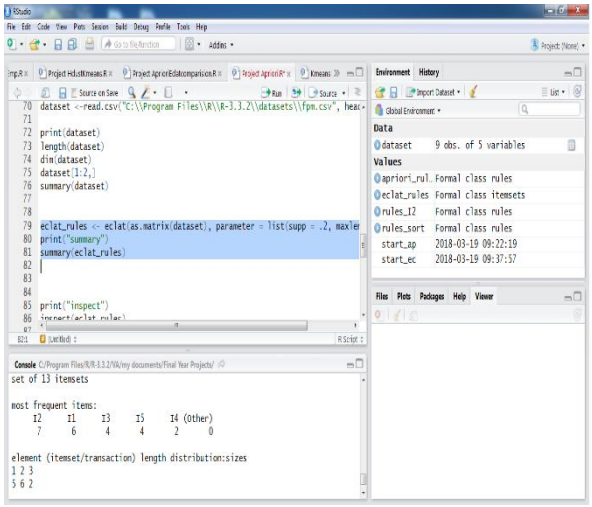


Fig. 10. Screen shot of Eclat Algorithm execution output

The on top of Figure Fig. 10. Showed the execution output of Eclat Algorithm.

The basic idea for the Eclat algorithm is use tidset intersections to compute the support of a candidate itemsets avoiding the generation of subsets that does not exist in the prefix tree.

It uses a vertical database layout i.e. instead of explicitly listing all transactions: each item is stored together with its cover (also called tidlist). It uses the intersection based approach to compute the support of an itemsets.

The underneath Figure Fig. 11. Showed the output of Eclat Association Rule.

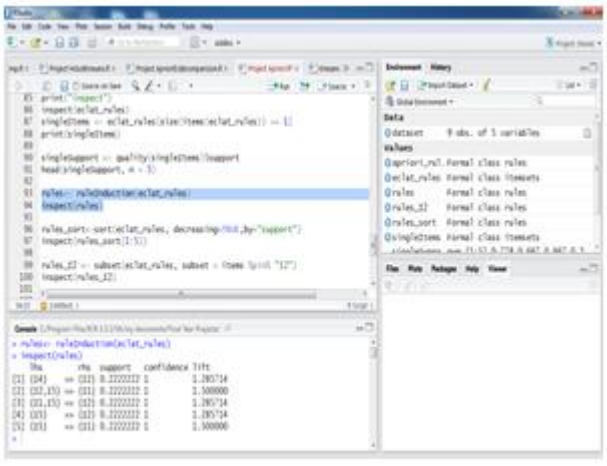


***Fig. 11. Screen shot of Eclat Association Rule execution output***

***Algorithm***

1. Get TID list for each item from DB

2. TID list of {a} is exactly the list of transactions containing {a}

3. Intersect the TID list of {a} with TID list of all other items, resulting in

2-itemsets – {a,b},{a,c}….

4. Then form 3 –itemsets – {a,b,c},….

The above process is repeated until no Transaction is available

The support and confidence of the itemsets calculation is same as Apriori algorithm. The finding of association rule in Eclat is same as Apriori algorithm.

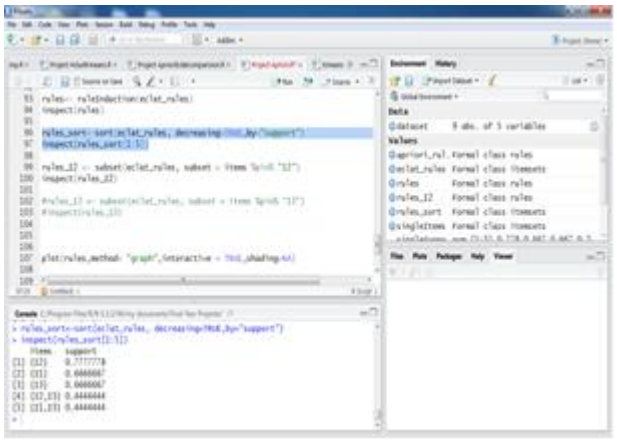


Fig. 12. Screen Shot Of Sorting Association Rules Execution Output

The higher than figure Fig. 12. Shows the Screen Shot Of Sorting Association Rules Execution Output.

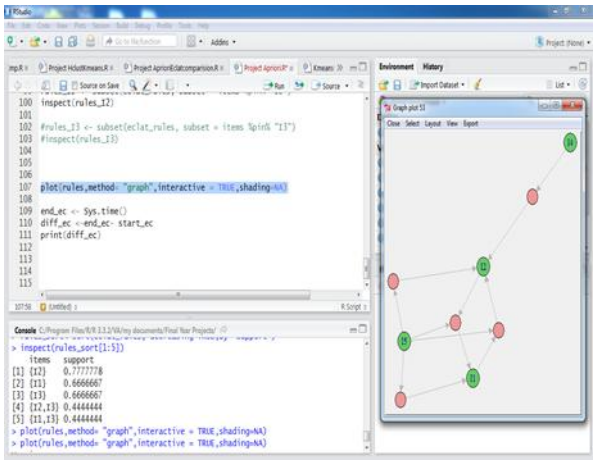


Fig. 13. Screen Shot Of Eclat Association Rule Graph Output

Now, the Eclat Association Rule Graph output showed in Figure Fig. 13.

January-February 2020
ISSN: 0193-4120 Page No. 12663 - 12672

*D. Comparison Of Data Science Algorithms*

The Table 1.2 shows the performance of Apriori and Eclat algorithms are evaluated using execution time. The total number of records taken for evaluation is 10000, 50000 and 100000.

For 10000 records, Apriori algorithm takes 4.7 seconds and Eclat algorithm takes only 3.5 seconds. For 50000 records, Apriori algorithm takes 5.9 seconds and Eclat algorithm takes only 4.4 seconds. For 100000 records, Apriori algorithm takes 7.4 seconds and Eclat algorithm takes only 4.7 seconds.

The Eclat algorithm has taken less execution time for finding the frequent itemsets and association rule among the items when compared to Apriori algorithm. The performance of the algorithms has shown in graphical representation in the Figure Fig. 6. It shows that the Eclat algorithm has a better performance over Apriori algorithm.

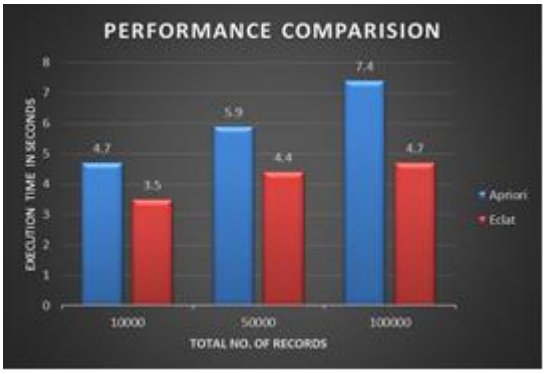



Fig. 14. Performance graph of Eclat Algorithm

## IV. CONCLUSION

The higher than Figure Fig. 14 showed the performance graph of Eclat Algorithm.

It addresses the algorithms used for finding of frequent patterns and association rule from huge amount of datasets [1]. Initially Apriori algorithm is applied to find the frequent patterns and association rule, then Eclat algorithm is used. In Eclat, it is not necessary to scan the entire database, for finding the support count. The massive experimental work was performed for evaluating and comparing Apriori and Eclat algorithms. The execution time taken by Eclat algorithm for calculating frequent patterns and association rule is very less compared to Apriori algorithm. It concludes, that Eclat algorithm consistently performs better and faster than Apriori algorithm.

Table 1.2 Performance evaluation of Apriori and Eclat Algorithms

| **Sl. No.** | **Algorithms** | **Total No. of Records** | | |
|---|---|---|---|---|
| | | *10000* | *50000* | *100000* |
| 1 | APRIORI | 4.7 seconds | 5.9 seconds | 7.4 seconds |
| 2 | ECLAT | 3.5 seconds | 4.4 seconds | 4.7 seconds |

REFERENCES

1. Jeff Heaton, “Comparing dataset characteristics that favour the Apriori and FP-Growth frequent itemset mining algorithms”, IEEE SoutheastConf 2016.
2. WildanBudiawan Zulfikar, Agung Wahana, WisnuUriawan, Nur Lukman. “Implementation of Association rules with Apriori Algorithm for Increasing the quality of Promotion”, Proceedings of the IEEE International Conference on Cyber and It service Mangement, 2016, PP: 26-27.
3. Cao Xiaojun, “Mining accurate top-K frequent closed itemset from data streams”. Proceedings of IEEE International Conference on Computer Science and Electronics Engineering. 2012; PP:180–184.
4. Hong Guo., and Ya Zhou. “An Algorithm for Mining Association Rules Based on Improved Genetic Algorithm and its Application”, Proceedings of IEEE International Conference on Genetic and Evolutionary Computing, 2009, pp.117-120.
5. Sun D, Teng S, Zhang W. “An algorithm to improve the effectiveness of Apriori”. Proceedings of IEEE International Conference on Cognitive Informatics; 2007.PP: 385–390.
6. Lei Ji., Baowen Zhang., and Jianhua Li. “A New Improvement on Apriori Algorithm”, Proceedings of IEEE International Conference on

*January-February 2020*
*ISSN: 0193-4120 Page No. 12663 - 12672*

Computational Intelligence, Vol.1, 2006, pp.840-844.

7. Mohammed Al-Maolegi, Bassam Arkok, “An Improved Apriori Algorithm For Association Rules”, International Journal on Natural Language Computing (IJNLC) Vol. 3, No.1, February 2014,pp.21-29.
8. Reddy Sangeetha DM et al, "Apriori Algorithm and its Applications in The Retail Industry for Analyzing Customer Interests", International Journal of Engineering Technology Science and Research, Volume 2, Issue 3, March 2015, pp. 46-51.
9. Manjit kaur, Urvashi Grag, "ECLAT Algorithm for Frequent Itemsets Generation", International Journal of Computer Systems, Vol. 1, Issue 3, 2014, pp. 82-84.
10. Yuzi Dou, Xiwei Fei, Rui Zhu, Tianzhu Gao, Yanbing Wu and Lei Ma, "Application of Improved Eclat Algorithm in Students' Evaluation of Teaching", MATEC Web of Conferences 228, 01017 (2018) CAS 2018, pp.1-5.