**How to Cite**:
Srinadh, V. (2022). Evaluation of Apriori, FP growth and Eclat association rule mining algorithms. *International Journal of Health Sciences*, *6*(S2), 7475–7485.
https://doi.org/10.53730/ijhs.v6nS2.6729

# Evaluation of Apriori, FP growth and Eclat association rule miningalgorithms

**Dr. V Srinadh**
Computer Science & Engineering, GMR Institute of Technology, Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India
Corresponding author email: srinadh.v@gmrit.edu.in

***Abstract*---**Association rule mining means to discover the guidelines which empower us to anticipate the event of a particular thing dependent on the events of different things in the exchange. Incessant thing set mining prompts the disclosure of affiliations and connections among things in enormous value-based or social informational collections. With monstrous measures of information constantly being gathered and put away, numerous enterprises are becoming keen on mining such examples from their data sets. The disclosure of intriguing connection connections among immense measures of deal records can help in numerous business dynamic cycles, for example, inventory configuration, cross-advertising, and client shopping conduct examination. In this we assess diverse sort of calculations like Apriori, FP Growth and Eclat calculation for affiliation decide mining that deals with regular thing sets. Affiliation rule mining between various things in enormous scope data set is a significant information mining issue. We assess these calculations by considering different elements like number of exchanges, least help, memory utilization and execution time. Assessment of calculations is created dependent on exploratory information, which give end.

***Keywords*---**Apriori algorithm, FP growth algorithm, Eclat algorithm.

## Introduction

As of late the spans of information bases have expanded quickly. At the point when huge amount of information is continually acquired and put away in data sets, a few enterprises are becoming worried in mining affiliation rules from their data sets. To find the covered information from these information affiliation rule mining can be applied in any application. The mining of affiliation rules is perhaps the most famous issues of every one of these. Mining of affiliation rules from regular example from gigantic assortment of information is an incredible new innovation with extraordinary potential to help organizations center around

the main data in their information stockrooms and gives direction in dynamic cycles like cross showcasing, market bushel examination, advancement collection and so on A thorough examination, overview and investigation of different methodologies in presence for incessant thing set extraction with productive considerations have been introduced in this undertaking. This framework utilizes just customary calculation which utilizes expansiveness first hunt strategy and it plays out various outputs for creating competitor set and we need to look through related information things while buying items. The calculation gets ended when the incessant thing sets can't be expanded further. Consequently execution time is more burned through in creating applicants without fail, it additionally needs more hunt space and computational expense is excessively high.

In this paper we address Apriori, Fpgrowth and Eclat calculations for affiliation decide mining that chips away at successive thing sets. Affiliation rule mining between various things in huge scope information base is a significant information mining issue. These days, there are numerous calculations accessible for affiliation rule mining. In this we are assessing a portion of the calculation by considering different variables like number of exchanges, least help, memory use and execution time. Assessment of calculations are created dependent on exploratory information, which gives end.

**Literature Survey**

As of late, the significance of data set mining is developing at a very high speed because of the expanding utilization of registering for different applications. Perhaps the main information mining issues is mining affiliation rules. Affiliation rule mining is a methodology that is intended to discover successive examples, connections, affiliations or causal designs from informational collections found in different sorts of data sets, for example, social data sets, value-based data sets and different types of information vaults. An affiliation rule has two sections, a precursor and an ensuing. A predecessor is a thing found in the information. An ensuing is a thing that is found in blend with the predecessor. In information mining, affiliation rules are helpful for dissecting and anticipating client practices. Thus, this paper, presents the investigation of different affiliation rule mining and afterward examine about the past explores which are related with the affiliation rule mining. In addition, this paper gives the outline of the different affiliation rule mining calculations, including Apriori, Eclat, FP-development calculations and its correlations. The principle objective of this exploration is to learn about various affiliation rule mining calculations. At last, examinations are made in affiliation rule mining as far as benefits, bad marks, results and information sets[1].

Information is significant property for everybody. Huge measure of information is accessible on the planet. There are different archives to store the information into information distribution centers, data sets, data storehouse and so on This huge measure of information needs to measure with the goal that we can get helpful data. Information mining is a strategy to handle information, select it, incorporate it and recover some valuable data. Information mining is a scientific apparatus which permits clients to investigations information, classes it and rundowns the connections among the information. It finds the valuable data from huge number of social information bases. Information mining can play out these different

exercises utilizing its strategy like bunching, arrangement, expectation, affiliation learning and so on This paper presents an outline of affiliation rule mining calculations. Calculations are examined with appropriate model and looked at dependent on some exhibition factors like exactness, information support, execution speed etc[2].

The help is for the most part higher when the traditional Apriori calculation is utilized as mining information dependent on affiliation rules, on the off chance that the backings little low, excess continuous thing set and repetitive guidelines are delivered enormous, so the nearby compelling affiliation rules has a bigger certainty and a more modest help can't be mined out, which is the lethal imperfections of the old style Apriori calculation. As per the imperfections, the adequacy of neighborhood rules is demonstrated from the outset, in the mean time, two sorts of the amendment calculations are given: the one is Apriori-con calculation dependent on certainty and the other is Apriori calculation dependent on order which is additionally separated into three sorts, Apriori class-int calculation dependent on premium arrangement, apriori-classpre calculation dependent on gauge characterization and apriori-classclr calculation dependent on grouping. The rightness of the hypothesis is demonstrated in the article and the compelling of the revision calculations is displayed by cases[3].

Presently a day's Data mining has a great deal of internet business applications. The key issue is the way to discover valuable covered up designs for better business applications in the retail area. For the arrangement of these issues, The Apriori calculation is quite possibly the most well known information digging approach for discovering successive thing sets from an exchange dataset and infer affiliation rules. Rules are the found information from the information base. Discovering successive thing set (thing sets with recurrence bigger than or equivalent to a client indicated least help) isn't insignificant in light of its combinatorial blast. When regular thing sets are gotten, it is clear to create affiliation rules with certainty bigger than or equivalent to a client indicated least certainty. The paper representing Apriori calculation on mimicked data set and discovers the affiliation rules on various certainty value[4].

Discovering continuous thing sets is computationally the most costly advance in affiliation rules mining, and the greater part of the exploration consideration has been centered around it. With the perception that help assumes a significant part in continuous thing mining, in this paper, a guess on help tally is demonstrated and upgrades of customary Eclat calculation are introduced. The new Bi-Eclat calculation arranged on help: Items sort in dropping request as per the frequencies in exchange reserve while thing sets utilize rising request of help during help tally. Contrasted and conventional Eclat calculation, the consequences of examinations show that the Bi-Eclat calculation acquires better execution on a few public information bases given. Besides, the Bi-Eclat calculation is applied in investigating blend standards of solutions for Hepatitis B in Traditional Chinese Medicine, which shows its proficiency and viability in commonsense value [5].

**Methodology**

There are various techniques are proposed for generating frequent item sets so that association rules are mined efficiently. The approaches of generating frequent item sets aredivided into basic three techniques.

- Horizontal layout-based data mining techniques: Apriori algorithm
- Vertical layout-based data mining techniques: Eclat algorithm

**Apriori Algorithm**

This is the most classical and important algorithm for mining frequent item sets. Apriori isused to find all frequent item sets in a given database DB. The key idea of Apriori algorithmis to make multiple passes over the database. Apriori algorithm fairly depends on the Apriori property which states that "All non-empty item sets of a frequent itemset must be frequent". It also described the anti-monotonic property which says if the system cannot pass the minimum support test, all its supersets will fail to pass the test Apriori algorithm follows two phases:

- Generate Phase: In this phase candidate (k+1)-itemset is generated using k- itemset, this phase creates Ck candidate set.
- Prune Phase: In this phase candidate set is pruned to generate large frequent itemset using "minimum support" as the pruning parameter. This phase creates Lk large itemset.
- These disadvantages can be minimized by applying techniques to:
- Reduce passes of transaction database scans
- Shrink number of candidates
- Facilitate support counting of candidates

Apriori is a classic algorithm for frequent item set mining and association rule learning over transactional databases. Apriori algorithm is since the algorithm uses prior knowledge of frequent itemset properties. This technique uses the property that any subset of a large itemset must be a large itemset. Apriori generates the candidate item sets by joining the large item sets of the previous pass and deleting those subsets which are small in the previous pass without considering the transactions in the database.

An association rule is valid if its confidence and support are greater than or equal to corresponding threshold values. Apriori employs an iterative approach known as a level-wise search, where k-item sets are used to explore (k+1)-item sets. First, the set of frequent 1-itemsets is found. This set is denoted L1. L1 is used to find L2, the frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k item sets can be found. The finding of each Lk requiresone full scan of the database.

- Join Step: $C_k$ is generated by joining $L_{k-1}$ with itself.
- Prune Step: Any(k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset.

- Pseudo-code: C k: Candidate itemset of size k
  Lk: Frequent itemset of size kL1= {frequent items};
  for (k=1; L k! =Ø; k++)do begin
  C k+1 =candidates generated from L k;
  for each transaction t in database do
  increment the count of all candidates in C k-1 that are contained in tLk+1 = candidates in C k+1 with min_support
  end return Uk Lk

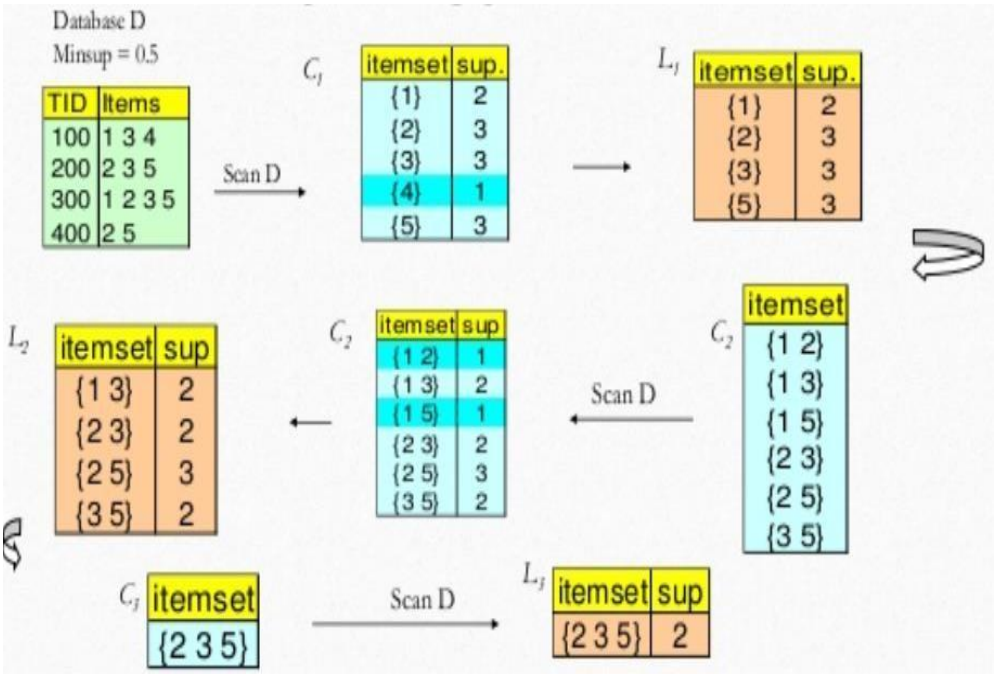


Fig: 3.1 Example of FP Apriori Algorithm

**FP Growth**

This is another important frequent pattern mining method, which generates frequent itemset without candidate generation. It uses tree-based structure. The problem of Apriorialgorithm was dealt with, by introducing a novel, compact data structure, called frequent pattern tree, or FP-tree then based on this structure an FP-tree-based pattern fragment growth method was developed.

It constructs conditional frequent pattern tree and conditional pattern base from database which satisfy the minimum support. FP-growth traces the set of concurrent items.

FP tree is constructed in two passes:

Pass 1: Scan data and count support for each item

- Discard infrequent items

- Sort frequent items in descending order based on their support

Pass 2: Reads one transaction at a time and maps it to the tree

- Fixed order is used so that path can be shared
- Pointers are maintained between nodes containing same items
- Frequent items are extracted from the list It suffers from certain disadvantages:
- Fp tree may not fit in main memory
- Execution time is large due to complex compact data structure.

**Eclat Algorithm**

ECLAT algorithm is a depth first search-based algorithm. It uses a vertical database layout i.e. instead of explicitly listing all transactions; each item is stored together with its cover (also called titlist) and uses the intersection-based approach to compute the support of an itemset. It requires less space than Apriori if item sets are small. It is suitable for small datasets and requires less time for frequent pattern generation than Apriori.

There are many regulated information mining strategies in which affiliation rule mining is one among them. ECLAT is a profundity first pursuit calculation utilizing set crossing point. It is a normally rich calculation reasonable for both successive just as equal execution with region upgrading properties. The ECLAT calculation is utilized to perform itemset mining. Itemset mining let us find continuous examples in information. This sort of example is called affiliation manages and is utilized in numerous application spaces. The fundamental thought for the ECLAT calculation is use Tidset crossing points to process the help of an applicant itemset staying away from the age of subsets that doesn't exist in the prefix tree. ECLAT, an information mining technique that was originally created for market bin examination. Successive thing set mining targets discovering formalities in the shopping conduct of the clients of grocery stores, mail-request organizations and online shops. It attempts to distinguish sets of items that are much of the time purchased together. Once recognized, such arrangements of related items might be utilized to advance the association of the offered items on the racks of a grocery store or the pages of a mail-request list or web shop, may give hints which items may advantageously be packaged, or may permit to propose different items to clients. In any case, incessant thing set digging might be utilized for a lot more extensive assortment of errands, what share that one is keen on discovering consistencies between (ostensible) factors in a given informational index. ECLAT depends on two fundamental advances: applicant age and pruning. In the competitor age step, every k-itemset applicant is created from two regular $k - 1 - itemsets$ and afterward its help is tallied, assuming its help is lower than the limit, it will be disposed of, else it is incessant thing sets and used to produce $k + 1 - itemsets$. Since ECLAT utilizes the upward design, checking support is paltry. Applicant age is surely a hunt in the inquiry tree. The thing base with 1-itemset is a comparability class with the prefix $\{\}$ and this identicalness class is equivalent to the underlying exchange information base

in vertical format. ECLAT doesn't completely take advantage of the descending conclusion property because of its profundity first inquiry.

**Algorithm**

- Get Tid list for each item (DB scan)
- Tid list of {a} is exactly the list of transactions containing {a}
- Intersect Tid list of {a} with the Tid lists of all other items, resulting in Tid lists of {a, b},
  {a, c},{a, d},={a}-conditional database (if {a} removed)
- Repeat from 1 on {a}-conditional database
- Repeat for all other items

| TID | Items |
|---|---|
| 1 | Bread,Butter,Jam |
| 2 | Butter,Coke |
| 3 | Butter,Milk |
| 4 | Bread,Butter,Coke |
| 5 | Bread,Milk |
| 6 | Butter,Milk |
| 7 | Bread,Milk |
| 8 | Bread,Butter,Milk,Jam |
| 9 | Bread,Butter,Milk |

| Item Set | TID set |
|---|---|
| Bread | 1,4,5,7,8,9 |
| Butter | 1,2,3,4,6,8,9 |
| Milk | 3,5,6,7,8,9 |
| Coke | 2,4 |
| Jam | 1,8 |

Frequent 1-itemsets

min_sup=2

| Item Set | TID Set |
|---|---|
| Bread | 1,4,5,7,8,9 |
| Butter | 1,2,3,4,6,8,9 |
| Milk | 3,5,6,7,8,9 |
| Coke | 2,4 |
| Jam | 1,8 |

Frequent 2-itemsets

| Item Set | TID set |
|---|---|
| {Bread,Butter} | 1,4,8,9 |
| {Bread,Milk} | 5,7,8,9 |
| {Bread,Coke} | 4 |
| {Bread,Jam} | 1,8 |
| {Butter,Milk} | 3,6,8,9 |
| {Butter,Coke} | 2,4 |
| {Butter,Jam} | 1,8 |
| {Milk,Jam} | 8 |

Frequent 3-itemsets

| Item Set | TID Set |
|---|---|
| {Bread,Butter,Milk} | 8,9 |
| {Bread,Butter,Jam} | 1,8 |

Fig: 3.2: Example of ECLAT

**Experimental Results**

**Apriori Result**

| Dataset Names | Transactions count from database | Frequent itemsets count | Maximum memory usage(mb) | Total time(ms) |
|---|---|---|---|---|
| chess | 3196 | 25060 | 1.24 | 17875 |
| Mushroom | 8416 | 505 | 18.94 | 1016 |
| Pumsb | 180 | 11271 | 1.92 | 593 |
| Retail | 685 | 2 | 2.56 | 94 |

Table 4.1: Results of Apriori Algorithm on different data sets

**Fp-Growth Results**

| Dataset Names | Transactions count from database | Frequent itemsets count | Maximum memory usage(mb) | Total time(ms) |
|---|---|---|---|---|
| chess | 3196 | 25060 | 9.82 | 547 |
| Mushroom | 8416 | 505 | 8.01 | 379 |
| Pumsb | 180 | 11271 | 8.00 | 297 |
| Retail | 685 | 2 | 2.89 | 156 |

Table 4.2: Results of FP Growth Algorithm on different data sets

**Eclat Result**

| Dataset Names | Transactions count from database | Frequent itemsets count | Maximum memory usage(mb) | Total time(ms) |
|---|---|---|---|---|
| chess | 3196 | 25060 | 534.75 | 1815 |
| Mushroom | 8416 | 505 | 51.88 | 328 |
| Pumsb | 180 | 11271 | 16.97 | 218 |
| Retail | 685 | 2 | 15.66 | 31 |

Table 4.3: Results of ECLAT Algorithm on different data sets

**Comparison**

Here we have taken mushroom dataset with different minimum support counts and appliedthe associative mining algorithms.

| Min_Sup | Apriori | | FP-Growth | | Eclat | |
|---|---|---|---|---|---|---|
| | Time(ms) | Memory(mb) | Time(ms) | Memory(mb) | Time(ms) | Memory(mb) |
| 0.3 | 2599 | 26.84 | 589 | 4.21 | 471 | 146.81 |
| 0.4 | 1356 | 26.84 | 430 | 2.17 | 220 | 75.81 |
| 0.5 | 280 | 26.84 | 233 | 1.98 | 145 | 43.81 |

Table 4.4: Comparison of various FP Mining Methods on the same datasetwith different Min_Sup

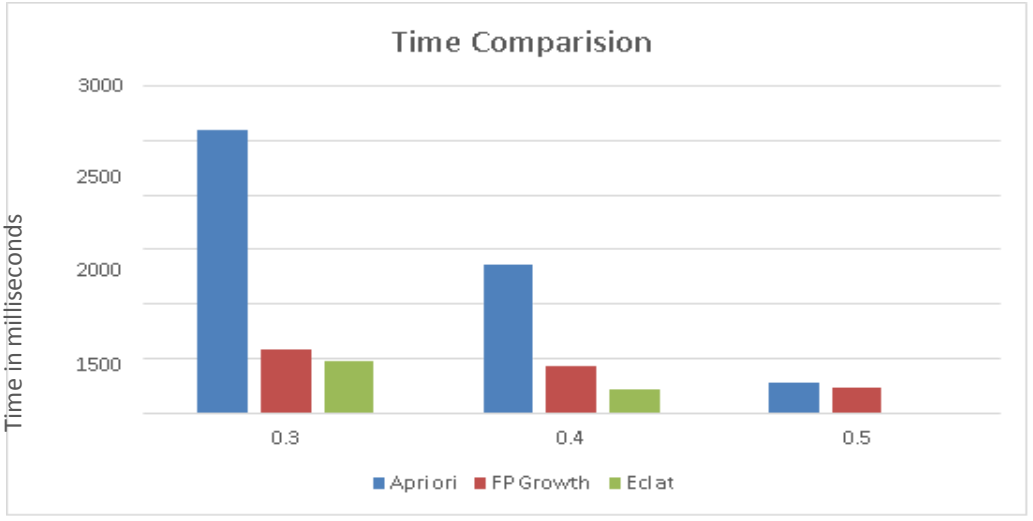


Figure 4.1: Graph Showing the Time taken(in ms) by mushroom dataset to generatefrequent patterns with different support counts

| DataSet Names | Transactions count from database | Frequent itemsets count | | | Maximum memory usage(mb) | | | Total time(ms) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Apriori | FP Growth | Eclat | Aprori | FP Growth | Eclat | Aprori | FP Growth | Eclat |
| Chess | 3196 | 25060 | 25060 | 25060 | 1.24 | 9.82 | 534.75 | 1787 | 547 | 1815 |
| Mushroom | 8416 | 505 | 505 | 505 | 18.94 | 8.01 | 51.88 | 1016 | 379 | 328 |
| Pumsb | 180 | 11271 | 11271 | 11271 | 1.92 | 8.00 | 16.97 | 593 | 297 | 218 |
| Retail | 685 | 2 | 2 | 2 | 2.56 | 2.89 | 15.66 | 94 | 156 | 31 |

Table 4.5: Comparison of various FP Mining Methods on the different dataset with same Min_Sup

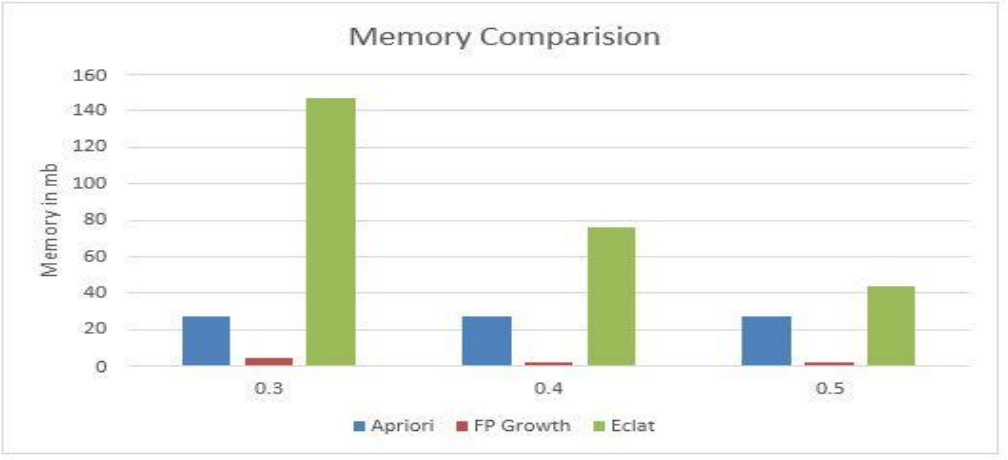


Figure 4.2: Graph Showing the Memory used(in mb) by mushroom dataset to generate frequent patterns with different support counts

In the table below, we have taken different datasets and the algorithms are applied by maintaining support count as 0.4 and the different parameters are taken down.

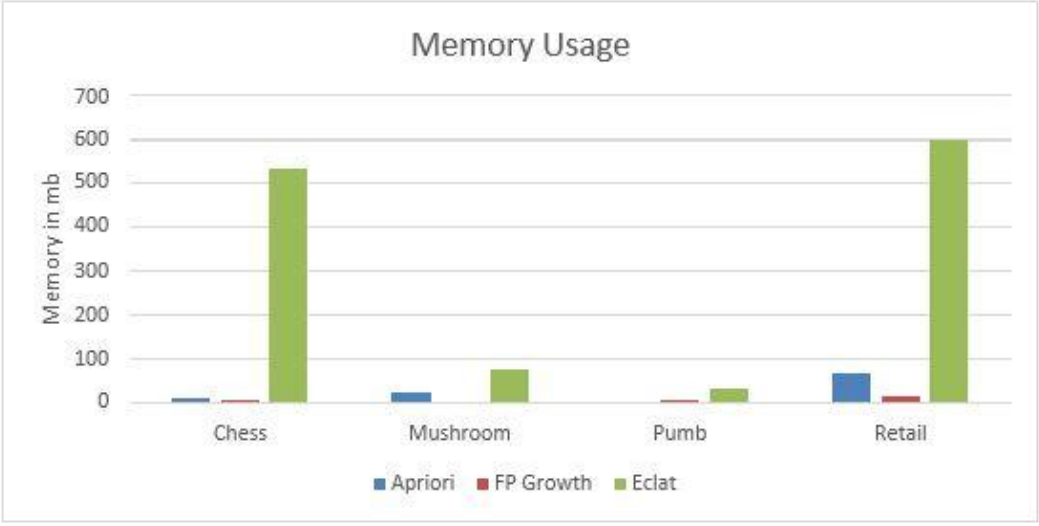


Figure 4.3: Graph Showing the Memory used (in mb) by different dataset to generate frequent patterns with same support count

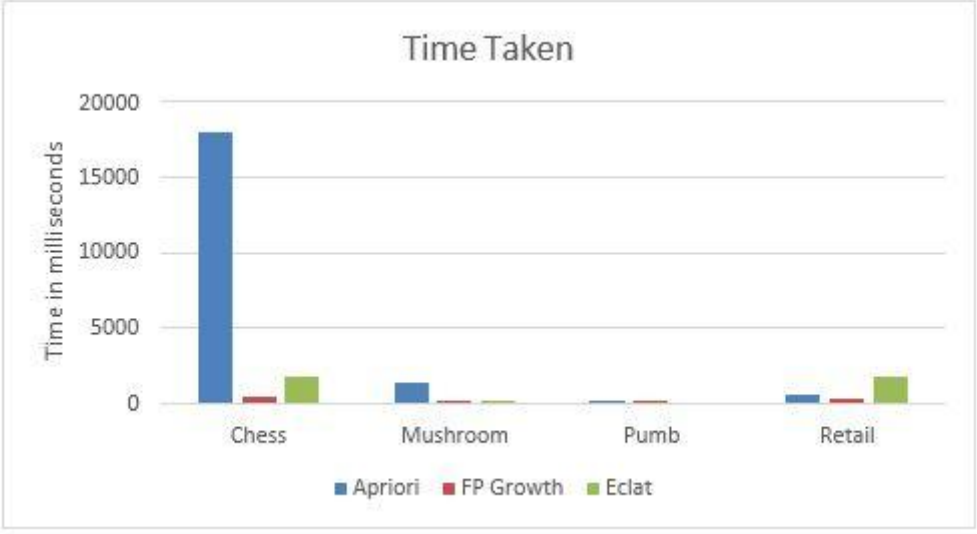


Figure 4.4: Graph Showing the Time taken (in ms) by different dataset togenerate frequent patterns with same support count

**Conclusion**

Frequent pattern mining is an important task in association rule mining. It has been found useful in many applications like market basket analysis, financial

forecasting etc. In classical algorithm like Apriori and Fpgrowth we use horizontal data form and here it consumes more time as we must scan the database many numbers of times. So, performance of technique depends on input data and available resources. Whereas, In Eclat repeated database scan is eliminated and it consumes less time and we can conclude that Eclat is better than Apriori and Fpgrowth. Here we just consider time as factor. If we consider other factor other than time the result may be varying from factor to factor.

**References**


1. P. Dhana Lakshmi, R. Porkodi "A Survey on Different Association Rule Mining Algorithms in Data Mining" ISSN 2321-5992; Volume 5, Issue 10, October 2017.
2. Trupti A. Kumbhare, Prof. Santosh V. "An Overview of Association Rule Mining Algorithms" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1), 2014, 927-930.
3. Zhang Chun-sheng, Liyan "Extension of Local Association Rules Mining Algorithm Based on Apriori Algorithm" 978-1-4799-3279-5 /14/$31.00 ©2014 IEEE
4. jugendra Dongre, Gend Lal Prajapati, S. V. Tokekar The Role of Apriori Algorithm for Finding the Association Rules in Data Mining 978-1-4799-2900-9/14/$31.00 ©2014IEEE
5. Xiaomei Yu, Hong Improvement of Eclat Algorithm Based on Support in Frequent Itemset Mining JOURNAL OF COMPUTERS, VOL. 9, NO. 9, SEPTEMBER 2014
6. Savi Gupta, Roopal Mamtora, "A Survey on Association Rule Mining in Market Basket Analysis", International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 4, Number 4 (2014), pp. 409-414.
7. Fournier-Viger, P., Gomariz, Gueniche, T. A. Soltani, A., Wu., C., Tseng, V. S. (2014). SPMF: A Java Open-Source Pattern Mining Library. Journal of Machine Learning Research (JMLR), 15: 3389-3393.
8. Varsha Mashoria, Anju Singh, "Literature Survey on Various Frequent Pattern Mining Algorithm", IOSR Journal of Engineering (IOSRJEN), e-ISSN: 2250-3021, p-ISSN: 2278- 8719, Vol. 3, Issue 1 (Jan. 2013), ||V1|| PP 58-64.
9. Endu Duneja, A.K. Sachan,"A Survey on Frequent Itemset Mining with Association Rules", International Journal of Computer Applications (0975 – 8887), Volume 46– No.23, May 2012.
10. Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", Data Mining and Knowledge Discovery, 8, 53–87, 2004, Kluwer Academic Publishers.