Frequent Pattern Generation in Association Rule Mining Using Apriori and FP Tree Algorithm

[1]Divya Makwana,[2]Krunal Panchal
[1]M.E Student,[2]Assitant Professor
Dept. of Computer Engineering,
L.J College of Eng. & Tech., Ahmedabad, India

***Abstract*—- Web mining can be comprehensively characterized as disclosure and investigation of valuable data from the World Wide Web. Web Usage Mining can be depicted as the revelation and investigation of client availability design, amid the mining of log documents and related information from a specific Web webpage, with a specific end goal to acknowledge and better serve the requirements of Web-based applications. Web utilization mining itself can be arranged further contingent upon the sort of use information considered they are web server, application server and application level information. This Research work concentrates on web utilize mining and particularly monitors running over the web usage cases of locales from the server log records. The holding of memory and time use is thought about by methods for Apriori calculation and enhanced Frequent Pattern Tree calculation.**

Keyword- Web usage mining, Apriori algorithm, improved Frequent Pattern Tree algorithm, Web log mining

I.INTRODUCTION

Web Mining

The number of Internet applications has grown and continue to grow significantly, affecting the lives of people in various aspects of their life including education, health, business and etc. The convenience and flexibility of services offered by web applications are the contributing factors why web applications are fast gaining popularity. In the process, web applications almost invariably churn out huge data containing user transactions and activity logs of user operations. Web usage mining is the process of finding out what users are looking for on the internet. Few users might be looking at only documented data, whereas some others might be interested in multimedia data.

- ➢ Web Server Data: The client logs are assembled by the Web server. Usually facts and figures include Internet Protocol address, sheet quotation and get access to time.
- ➢ Application Server Data: financial submission servers have significant characteristics to endow e-commerce submissions to be built on peak of them with tiny effort. A key feature is the proficiency to pathway diverse kinds of enterprise events and logs them in application server logs.

FP TREE

FP tree is a solid data architecture that retained important, absolutely vital and quantitative information considering common patterns [2].It comprises of one root marked as "root", a set of piece prefix sub-trees as the child of the root, and a frequent-item header chart. The size of the FP-trees bounded by the overall occurrences of the frequent items in the database.
The height of the tree is bound by the maximal number of frequent items in a transaction.

Advantages:

- It uses Compact data structure.
- It eliminates repeated database scan.
- It is faster than Apriori algorithm.
- It reduces the total number of candidate item sets by producing a compressed version of the database in terms of an FP tree.

Disadvantages:

- It takes more time for recursive calls.
- It is good only when user access paths are common.
- It utilizes more memory.

Apriori:

The Apriori Algorithm is an influential algorithm for mining frequent item sets for Boolean association rules [3].
Find the frequent item sets: the sets of items that have minimum support A subset of a frequent item set must also be a frequent item set i.e., if {AB} is a frequent item set, both {A} and {B} should be frequent item set. Iteratively find frequent item sets with cardinality from 1 to K (k-item set). Use the frequent item sets to generate association rules.

Advantages:

- It is very easy and simple algorithm.
- Its implementation is easy.

Disadvantages:

- It does multiple scan over the database to generate candidate set
- The number of database passes are equal to the max length of frequent item set.

Proposed System:

Proposed Model

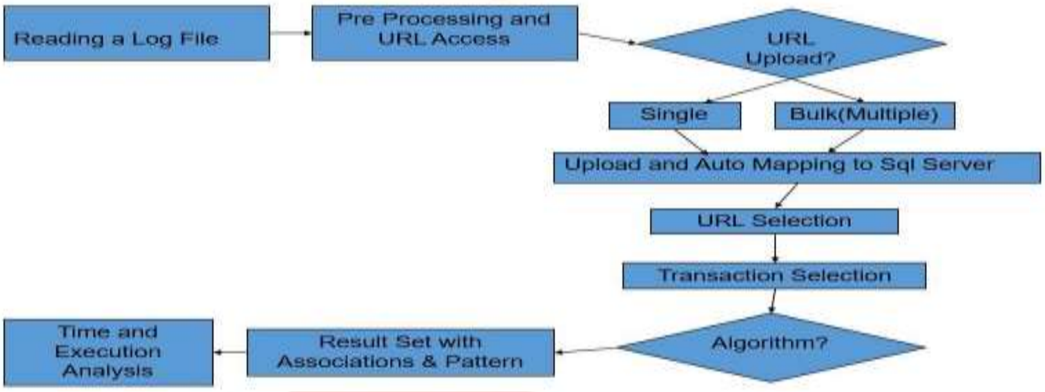

Reading a Log File
Pre Processing and URL Access
URL Upload?
Single
Bulk(Multiple)
Upload and Auto Mapping to Sql Server
URL Selection
Transaction Selection
Algorithm?
Result Set with Associations & Pattern
Time and Execution Analysis

Figure 1: Proposed Model

1. Start
2. Reading a log-file.
3. Pre-processing and URL Access.
4. URL Upload.
5. Single upload or Bulk (File) Upload.
6. Upload and Auto Mapping to SQL server.
7. URL Selection.
8. Transaction Selection.
9. Algorithm apply.
10. Result set with Associations and Pattern.
11. Time and Execution analysis.
12. End

Proposed Flowchart:

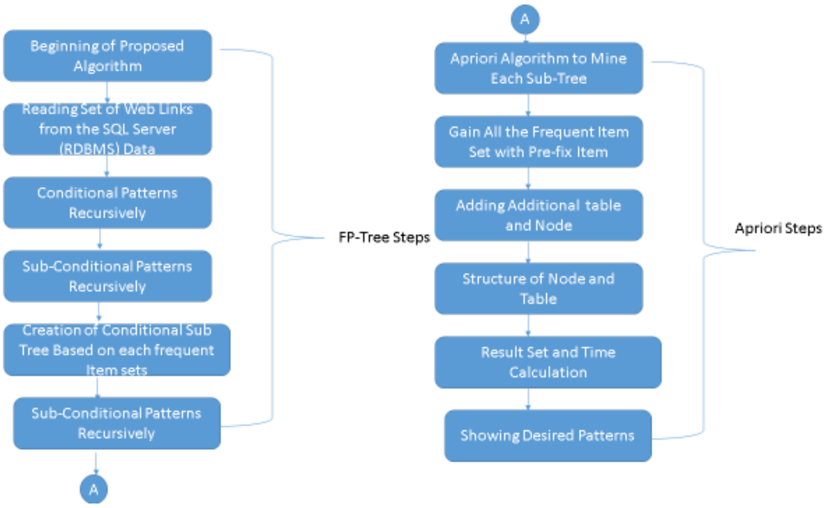

Beginning of Proposed Algorithm
Reading Set of Web Links from the SQL Server (RDBMS) Data
Conditional Patterns Recursively
Sub-Conditional Patterns Recursively
Creation of Conditional Sub Tree Based on each frequent Item sets
Sub-Conditional Patterns Recursively
A
FP-Tree Steps
A
Apriori Algorithm to Mine Each Sub-Tree
Gain All the Frequent Item Set with Pre-fix Item
Adding Additional table and Node
Structure of Node and Table
Result Set and Time Calculation
Showing Desired Patterns
Apriori Steps

Figure 2: Proposed flow

Proposed Algorithm:

Proposed Algorithm combines the Apriori algorithm and FP-tree structure. The advantage of proposed algorithm is that it doesn't need to generate conditional pattern bases and sub- conditional pattern tree recursively. And the proposed results of the experiments.

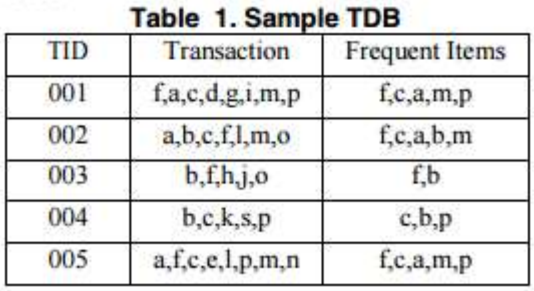
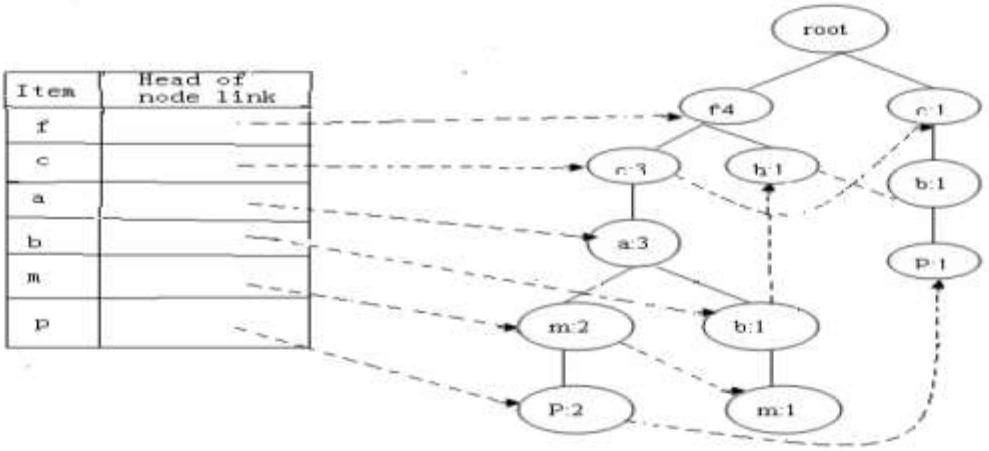


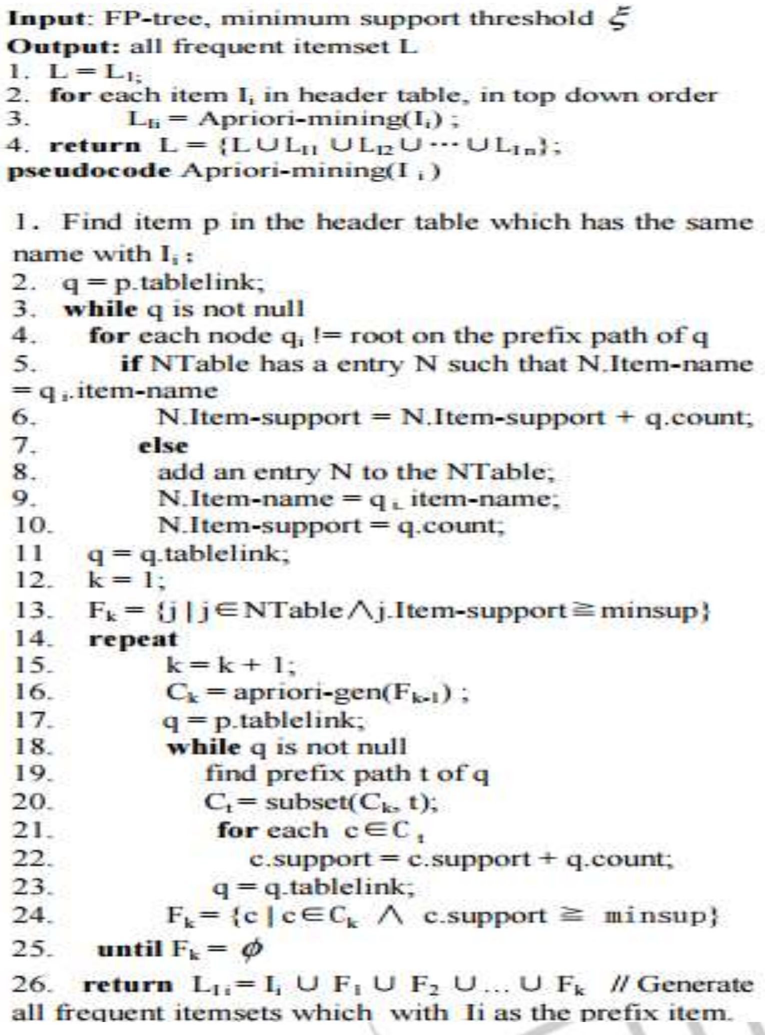
Table 1. Sample TDB

TID	Transaction	Frequent Items
001	f,a,c,d,g,i,m,p	f,c,a,m,p
002	a,b,c,f,l,m,o	f,c,a,b,m
003	b,f,h,j,o	f,b
004	b,c,k,s,p	c,b,p
005	a,f,c,e,l,p,m,n	f,c,a,m,p

(Transaction Table to understand proposed algorithm)



We consider using the Apriori method to mining the frequent item sets basing on the FP-tree, the divide-and-conquer strategy is still adopted by mining process. That is to say, the Compressed FP-tree is partitioned off a set of conditional sub tree, each of the conditional sub tree associated with a frequent item. If there are n 1-frequent items Ii(i=1,2,….n), then the FP-tree can be divided into n conditional subtree FPTi (i=1,2,….n) , and FPTi is the conditional subtree associating with frequent item Ii .Then use the apriori algorithm to mine each conditional subtree, and gain all the frequent itemsets with the first prefix item Ii. The proposed algorithm includes two steps, the first step is to construct the FPtree as FP-growth does, the second step is to use of the apriori algorithm to mine the FP-tree. On the second step, it is needed to add an additional node Table, named NTable, each entry in the NTable has two fields: Item-name, and Item-support. Item-name: the name of the node appears in the FPTi, Item-support: the number of the node appear with Ii the pseudo code of the proposed algorithm is described below.



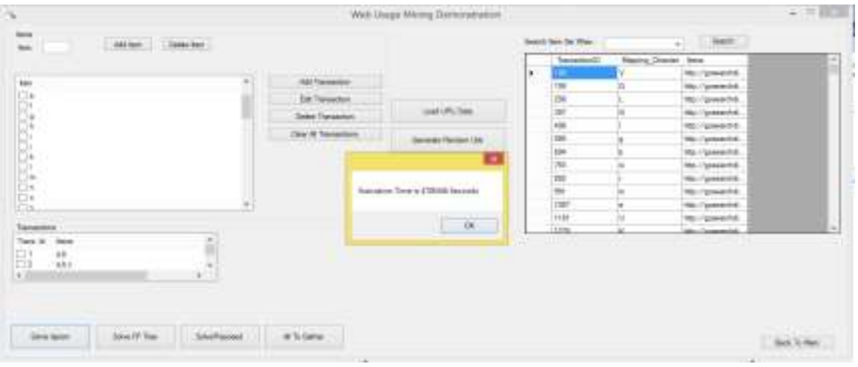
```
Input: FP-tree, minimum support threshold $\xi$
Output: all frequent itemset L
L = $L_1$;
for each item $I_i$ in header table, in top down order
        $L_{Ii}$ = Apriori-mining($I_i$) ;
return L = {$L \cup L_{I1} \cup L_{I2} \cup \cdots \cup L_{In}$};
pseudocode Apriori-mining($I_i$)

 Find item p in the header table which has the same
name with $I_i$ ;
 q = p.tablelink;
 while q is not null
   for each node $q_i$ != root on the prefix path of q
     if NTable has a entry N such that N.Item-name
= $q_i$.item-name
        N.Item-support = N.Item-support + q.count;
     else
        add an entry N to the NTable;
        N.Item-name = $q_i$ item-name;
        N.Item-support = q.count;
   q = q.tablelink;
   k = 1;
   $F_k$ = {j | j $\in$ NTable $\wedge$ j.Item-support $\geqq$ minsup}
   repeat
       k = k + 1;
       $C_k$ = apriori-gen($F_{k-1}$) ;
       q = p.tablelink;
       while q is not null
          find prefix path t of q
          $C_t$ = subset($C_k$, t);
           for each c $\in C_t$
              c.support = c.support + q.count;
         q = q.tablelink;
       $F_k$ = {c | c $\in C_k$ $\wedge$ c.support $\geqq$ minsup}
   until $F_k$ = $\phi$
 return $L_{1i}$ = $I_i \cup F_1 \cup F_2 \cup \ldots \cup F_k$   // Generate
all frequent itemsets which with Ii as the prefix item.
```

To explain the algorithm, we use an example with the transaction database showed in Table 1. The FP tree of this database is showed in Figure 1. The mining process begins from the top of the header table, and moves toward the bottom. For the f-node, it has only one prefix path and the prefix path has no other node except root-node, so there is no frequent item set has the first prefix item with f. For the next c-node in the header table, it has two prefix paths {f}, and {root}, f.support = c.count = 3 = minsup, there is no other request item in c's prefix, so we gain the frequent itemset {cf } with support 3, the process of mining the Frequent itemsets with the first prefix item c terminates. Next for the a-node, it has one prefix path:cf, and c.support = f.support = a.count = 3, so item c and item f are 1-frequent item, generate 2-candicate cf by joining c and f, then generate two-subset of a's prefix path, here just is cf, and (cf).support=a.count=3=minsup, so itemset cf is frequent, finally ,we get the frequent itemsets {ac:3,af:3,acf:3}. The remaining mining processes are similar.

Screenshot:

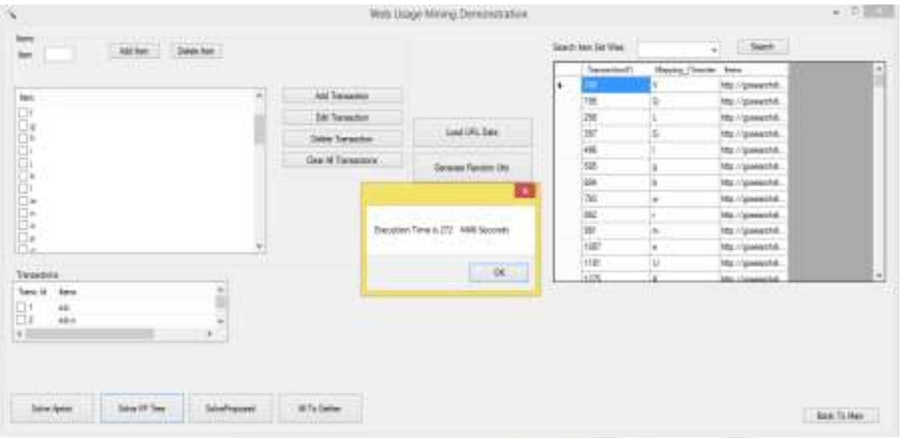
Execution time by Apriori algorithm



Execution Time is [illegible] Seconds

Figure 3 Apply Apriori algorithm

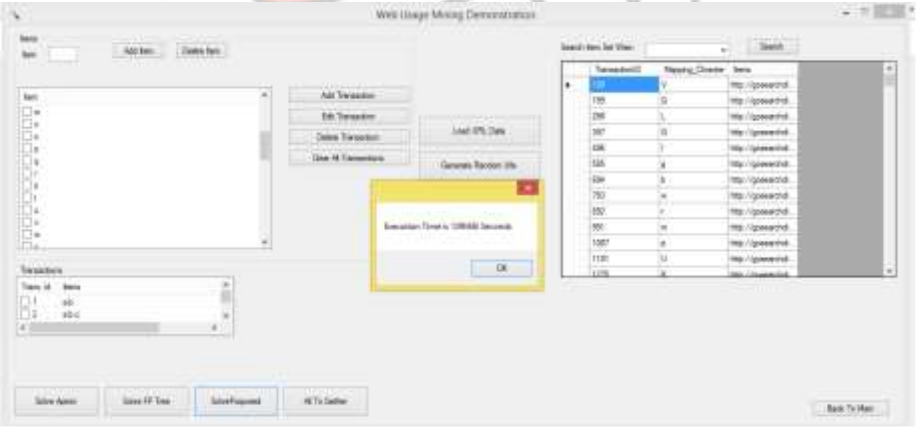
Execution time with FP-Tree algorithm



Execution Time is 272 [illegible] Seconds

Figure 4 Apply FP Tree algorithm

Execution of Proposed Algorithm

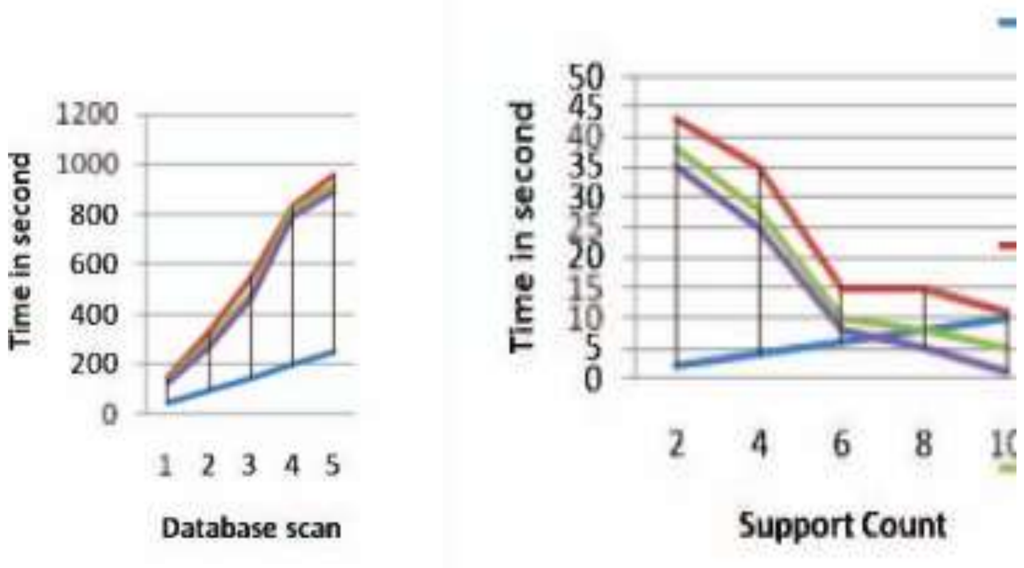


Execution Time is [illegible] Seconds

Figure 5 Apply Proposed algorithm

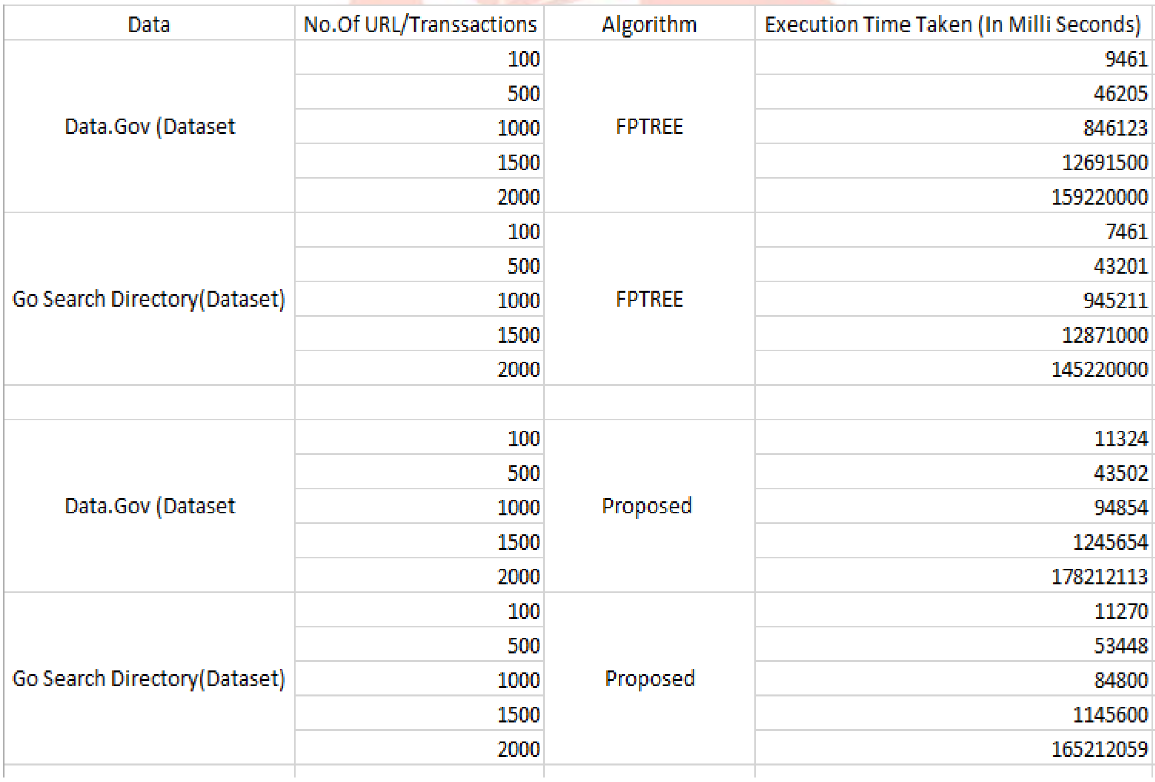
Results and experiments

Comparison Matrix and Performance Evaluation



Red- FP Tree Green and Purple – Apriori Blue – Proposed Algorithm

Results of Proposed System



Data	No.Of URL/Transsactions	Algorithm	Execution Time Taken (In Milli Seconds)
Data.Gov (Dataset	100	FPTREE	9461
	500		46205
	1000		846123
	1500		12691500
	2000		159220000
Go Search Directory(Dataset)	100	FPTREE	7461
	500		43201
	1000		945211
	1500		12871000
	2000		145220000
Data.Gov (Dataset	100	Proposed	11324
	500		43502
	1000		94854
	1500		1245654
	2000		178212113
Go Search Directory(Dataset)	100	Proposed	11270
	500		53448
	1000		84800
	1500		1145600
	2000		165212059

Conclusion:

The principle downside of Apriori calculation is that the hopeful set creation is expensive, particularly if an expansive number of examples or potentially long examples exist. The Fundamental downside of FP-development calculation is the unstable amount of does not have a decent applicant era technique. Future research can consolidate FP-Tree with Apriori competitor era strategy to tackle the weaknesses of both Apriori and FP-development. In future the calculation can be reached out to web content mining, web structure mining, and so forth. The work can likewise be reached out to concentrate data from picture records.

References:

1) Muhammad Asif, Jamil Ahmed“Analysis of Effectiveness of Apriori and Frequent Pattern Tree Algorithm in Software Engineering Data Mining” in IEEE 2015, DOI: 10.1007/978-981-10-0251-9_38 ,pp-574-578

2) Ashika Gupta, Rakhi arora, Ranjana sikarwar”Web Usage Mining Using Apriori Algorithm and Improved Frequent Pattern Tree Algorithm in Association Rule”in IEEE 2015, DOI: 10.1109/INDICON.2015.7443677,pp-353-357

3) Nandita Agrawal, Anand Jawdekar ”User-Based Approach For Finding Various Results In Web Usage Mining“ in IEEE 2015 , DOI:10.1109/ICCCNT.2015.7395171, pp-29-33

4) Hong-Yi Chang, Yih-Jou Tzang,Zih-Huan Hong “A Hybrid Algorithm for Frequent Pattern Mining Using MapReduce Framework” in IEEE 2015, DOI: 10.1007/978-981-10-0251-9_38,pp-20-22

5) Avadh Kishor Singh, Ajeet Kumar, Ashish K. Maurya ”Association Rule Mining for Web Usage Data to Improve Websites” in IEEE 2014, DOI: 10.1007/968-983-10-0251-9_38, pp-942-944

6) K .S .K .D. Association Rules Mining: A Recent Overview, GTS International Tran on Computer Science, Vol.65 (1), 2006), DOI: 10.1007/978-981-10-0251-9_38 pp. 403-412.

7) A R "Fast Algorithms for Mining Association Rules", Sep12-15 1994, Chile, 487-99, pdf, 1-55860- 153-9.

8) Mannila H, "Efficient algorithms for discovering association rules mining." conference Knowledge Discovery in Databases (SIGKDD). DOI:10.1109/ICCCNT.2015.7395171,pp.1-7,13-15 July 2015.

9) Han, J., Kamber, M., Data Mining Concepts and Techniques, Morgan Kaufmann Publisher, 2001

10) Heikki, Mannila. 1996. Data mining: machine learning, statistics, and databases, IEEE.

11) Jiawei Han, Micheline Kamber, “Data Mining Concepts and Techniques” Elsevier Publication.

12) Tan, P. N., M. St., V. Kumar, "Introduction to web Mining", Addison Wesley, 2013, 769pp.

13) Charu C. Aggarwa” Data Mining” Addison-Pearson 2012 ISBN 976-81-317-0688-6

15) https://en.wikipedia.org/wiki/C_Sharp_(programming_language)(Accessed:4/12/2016)at1:27am

16) https://en.wikipedia.org/wiki/Microsoft_Visual_Studio(Accessed:21/10/2016)at 2:17 pm

17) https://en.wikipedia.org/wiki/Web_mining (Accessed:4/12/2016)at 1:26 am

18) https://en.wikipedia.org/wiki/Data_mining(Accessed:4/12/2016)at1:26 pm