Kingdom of Saudi Arabia

Ministry of Education

Qassim University

College of Computer

Information Technology Department



المملكة العربية السعودية

وزارة التعليم

جامعة القصيم

كلية الحاسب

قسم تقنية المعلومات

An Enhanced System for Association

Rules Construction for Data Mining Applications

*Students:*

|  |  |
| --- | --- |
| Manar Abdulrahman Al-Salamah | 321200260 |
| Maha Mohammed Al-Qumea | 321200762 |

*Supervisor:*

Dr. Chafia Kara-Mohamed

Phase I/371

*P project report submitted in partial fulfillment of the requirements   
for B.Sc. degree in Information Technology.*

*Qassim-Saudi Arabia*

*1437/1438 (2016/2017)*

**Table of Contents**

Table of Contents --------------------------------------------------------------------------ii

List of Figures -----------------------------------------------------------------------------iv

List of Tables -------------------------------------------------------------------------------v

Certificate ----------------------------------------------------------------------------------vi

Dedication ---------------------------------------------------------------------------------vii

Acknowledgement ----------------------------------------------------------------------viii Abstract ------------------------------------------------------------------------------------ix

Chapter One

INTRODUCTION AND OBJECTIVES

1.1 Introduction ------------------------------------------------------------------------------1

1.2 Problem Specification and Motivation -----------------------------------------------1

1.3 Goals and Objectives -------------------------------------------------------------------2

1.4 Study Scope ------------------------------------------------------------------------------2

1.5 Study Plan and Schedule ---------------------------------------------------------------3

1.5.1 Project Plan -----------------------------------------------------------------------3

1.5.2 Project Time Table --------------------------------------------------------------4

1.6 Organizing of the Chapters ------------------------------------------------------------4

Chapter Two

Literature REVEIW

2.1 What is Data Mining? ------------------------------------------------------------------5

2.1.1 Data Mining Application--------------------------------------------------------5

2.1.2 Data Mining Process ------------------------------------------------------------6

2.1.3 Data Mining Environments -----------------------------------------------------7

2.1.3.1 Rapid Miner----------------------------------------------------------------7

2.1.3.2 WEKA----------------------------------------------------------------------8

2.1.3.2 Orange----------------------------------------------------------------------9

2.2 Data Mining Algorithms ---------------------------------------------------------------9

2.2.1 Classification --------------------------------------------------------------------9

2.2.2 Clustering ------------------------------------------------------------------------10

2.2.3 Outlier Analysis -----------------------------------------------------------------10

2.2.4 Association Rules AR ----------------------------------------------------------11

2.3 General AR Algorithms----------------------------------------------------------------11

2.3.1 Process ----------------------------------------------------------------------------12

2.3.2 Some important AR Algorithm ------------------------------------------------12

2.3.3 FP-Growth ------------------------------------------------------------------------13

2.3.3.1 Definition -----------------------------------------------------------------13

2.4 Related Works (Association Rule) ---------------------------------------------------17

2.4.1 Applying FP-Growth in WEKA------------------------------------------------17

2.4.1.1 FP-Growth Compared to other Algorithms ---------------------------20

2.4.1.2 Some Improvements on FP-Growth Algorithms ---------------------21

2.4.1.3 Implementation -----------------------------------------------------------23

2.5 Conclusion-------------------------------------------------------------------------------------23

Chapter Three

METHODOLOGY

3.1 Introduction ----------------------------------------------------------------------------------24

3.2 Design of Project Structure ----------------------------------------------------------------24

3.3 Data and Information Gathering ----------------------------------------------------------25

3.4 The Interface Design -----------------------------------------------------------------------25

3.5 Gathering Project Data Set ----------------------------------------------------------------28

3.6 Class Diagram of The Algorithm ---------------------------------------------------------28

3.7 The Project Process -------------------------------------------------------------------------29

3.8 What is Web Logs? -------------------------------------------------------------------------29

Chapter FOUR

IMPLIMNTAION AND TESTING

4.1 Introduction ------------------------------------------------------------------------------32

4.2 IMPLIMANTATION-------------------------------------------------------------------33

4.2.1 FPGrowthApplication ----------------------------------------------------------33

4.2.2 FPTree -----------------------------------------------------------------------------34

4.2.3 HeaserTableComarator ----------------------------------------------------------34

4.2.4 FPGrwoth -------------------------------------------------------------------------34

4.2.5 AlgoAgrawalFaster94------------------------------------------------------------36

4.2.6 ArraysAlgos -----------------------------------------------------------------------38

4.2.7AssociationRules\_FPGrowth\_saveToFile-------------------------------------38

4.2.8 Welcoming Interface -------------------------------------------------------------39

4.3 Data Sets ---------------------------------------------------------------------------------40

4.3.1 Type of Data Set -----------------------------------------------------------------40

Chapter Five

RESULTS AND DISCUSSION

5.1 Test the Code with Small and New Data Set ----------------------------------------41

5.1.1 The Frequent ----------------------------------------------------------------------41

5.1.2 The Rule ---------------------------------------------------------------------------43

5.1.3 All the Result ----------------------------------------------------------------------44

5.2 Test the code with Anonymous Microsoft Web Data Set-------------------46

5.2.1 The Frequent ----------------------------------------------------------------------46

5.2.2 The Rule ---------------------------------------------------------------------------47

5.2.3 All the Result ---------------------------------------------------------------------49

5.3 The Analyses of the Result------------------------------------------------------51

Conclusion-----------------------------------------------------------------------------53

References --54

Appendix --57

**List of Figures**

Figure 1.1: Project Plan 3

Figure 1.2: Project Time Table 4

Figure 2.1 The process of data mining 7

Figure 2.2 RapidMiner Software 8

Figure 2.3: WEKA Software 8

Figure 2.4: Orange Software 9

Figure 2.5: Header Table 17

Figure 2.6: Data set 18

Figure 2.7: WEKA Interface 18

Figure 2.8: WEKA’s List of Functions 19

Figure 2.9: Association’s Algorithms 19

Figure 2.10: Result Using FP-Growth 20

Figure 2.11: Implementation---------------------------------------------------------23

Figure 3.1: Designing Process 25

Figure 3.2: Project Interface 26

Figure 3.3: File in Menu Bar 26

Figure 3.4: Window Options 27

Figure 3.5: Edit Options 27

Figure 3.6: FP-Growth Class Diagram 28

Figure 3.7: Process of the Project 29

Figure 4.3: latest Project Interface--------------------------------------------------40

Figure 5.1: Result of frequent with threshold 2------------------------------------41

Figure 5.2: Result of frequent with threshold 4------------------------------------42

Figure 5.3: Result of rule with default threshold 2 and with confidence 0.2---43

Figure 5.4: Result of rule with threshold 4 and with confidence 0.4------------44

Figure 5.5: Result of all with default threshold and with confidence 0.2-------45

Figure 5.6: Result of all with threshold 4 and with confidence 0.4--------------45

Figure 5.7: Result of frequent with threshold 10-----------------------------------46

Figure 5.8: Result of frequent with threshold 20-----------------------------------46

Figure 5.9: Result of frequent with threshold 50-------------------------------------47

Figure 5.10: Result of rule with frequent default threshold 2 and with confidence 0.1-------------------------------------------------------------------------------47

Figure 5.11: Result of rule with default threshold 4 and with confidence 0.2----48

Figure 5.12: Result of rule with default threshold 50 and with confidence 0.5---48

Figure 5.13: Result of all with default threshold 10 and with confidence 0.1-----49

Figure 5.14: Result of all with default threshold 20 and with confidence 0.4-----50

Figure 5.15: Result of all with default threshold 50 and with confidence 0.5-----50

**List of Tables**

Table 2.1: Transactional Data 15

Table 5.1: Description of a small part of data set -------------------------------------51

**Certificate**

It is certified that project report has been prepared and written under my direct supervision and guidance. The project report is approved for submission for its evaluation.

*Dr. Chafia Kara-Mohamed*

**Dedication**

Thanks to Allah first and for most and then offer my sincere thanks to our parents who have been our constant source of inspiration. They have given us the drive and discipline to tackle any task with enthusiasm and determination. Without their love and support this project would not have been made possible.

Also, special thanks and appreciation to our supervisor: Dr. Chafia Kara-Mohamed who has always supported us throughout the process. As well as to each of those who gives us assistance to complete this part of our project, and certainly to all our work for them, to all the students of Information Technology Department and to the scientific edifice distinctive Qassim University.

*Manar Al-Salamah*

*Maha Al-Qumea*

**Acknowledgement**

First and foremost, Glory and Praise to Almighty *Allah* for sustaining us in all our endeavors – throughout our lives.

We would like to thank our Dean Waleed Al-Battah, our Vice-Dean for Students Affairs Dr. Suliman Al-‘Amro, our Vice-Dean for Educational Planning Dr. Ajlan Al-Ajlan and the Head of the IT Department Dr. Saleh Albahli for the facilities available at the College. Our thanks are also extended to our supervisors Dr. Chafia Kara-Mohamed and Prof. Aboubekeur Hamdi-Cherif, for the efforts they provided for us during the supervision of this Project. Their guidance helped us all the time during our graduate studies and while undertaking this Project. Our thanks also go to the examiners who will have to evaluate the present manuscript and to all our instructors in the College. We thank the administrative staff for the help they provided during these years at the University. Obviously, our heartfelt thanks go to our families for the support they provided us through our entire lives.

Finally, we would like to thank everyone who contributed to the successful completion of this Project

*Manar Al-Salamah*

*Maha Al-Qumea*

Abstract

When studying the users’ behaviors of our University site, we can help the administration in enhancing the different services offered by adapting the content or the structure or the interface of the university portal to satisfy these users. We can get these behaviors by finding the most frequent visited pages.

The emergence of huge data in the digital world has motivated the scientists to mine it in the aim of discovering the hidden knowledge inside it. This is the main objective of what we call Data Mining. Given that knowledge can be present in different forms in this data, and given the variety of data repositories, many algorithms are developed for this purpose.

One of the most important approach in data mining is finding the frequent patterns in a dataset. One of the most important algorithms developed for this purpose is FP-Growth.

In our project, we intend to discover the frequent patterns in the data provided by the web servers in our University. In other words, we will implement the FP-Growth algorithm to mine a real website.

Chapter One

INTRODUCTION AND OBJECTIVES

## 1.1 Introduction

Nowadays, the web represents an important source of information. The data is characterized by its sparsity. The huge amount of data in different type and Forms in the world has motivated the scientists to mine it in the aim of discovering the hidden knowledge inside it. This is the main aim of extracting data, especially through the vast amount of useful and non-useful data and the different forms of data and the diversity of the data warehouse. The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration. In this project our intention is to use Association Rule algorithm to detect patterns repeated in the data and finding out which data is duplicates in these data. To ease the search of important data and repetitive abundance for the user. We choose the Web mining to take data from it to be the data which we will use to extract patterns repeated in it, and we will follow algorithms FP- growth in this procedure.

## 1.2 Problem Specification and Motivation

Our problem is to extract patterns repeated in a data collected from web server log files. Mining these files by applying FP-Growth algorithm. Then the results are analyzed to get the lessons. After the analysis of these data and knowledge we have then some tips directed to the owner of the data set. The owner of the data set can focus on these patterns refined in terms of development and it might be useful to improve the website. Discovering repetitive patterns can help to know what the site’s visitors are interested in and what are not.

**This project is focusing on solving the following problems:**

1. Take the data set from a web site and apply the FP-Growth algorithm to it, and extract the frequent pattern from it.
2. Analyze the frequent patterns extracted from the FP-Growth.
3. Give the owner of those data set some tips to improve the web site.

**Motivation is the key of success; here is what motivates us to work on this****project:**

1. Huge amount of data sets, and data mining with more frequent pattern is accumulated, so we are drowning in data but we are starving for knowledge.
2. Support the responsible people who need a new knowledge.
3. The Knowledge gained helps responsible of the data in the planning of the development of the website.

## 1.3 Goals and Objectives

The main objective of this project is using the data mining techniques and tools for data set from the web to get the frequent visited pages by the users of the site.

The project aims to:

1. Apply the FP-growth over a dataset, and extract the frequent patterns from it.
2. Analyze the patterns found and give the site administration the results of this analysis.
3. Provide suitable knowledge to save time and effort and to assist in finding the frequent patterns to the owner of the web site.

## 1.4 Study Scope

The project scope is essentially limited by the data mining and it includes data mining tools and techniques, data sets, data analysis and use Association Rule (AR) algorithm to find the frequent patterns. In our project also, we have to study the web server log files as a source of our data.

## 1.5 Study Plan and Schedule

The project plan is a statement of how and when a project’s objectives are to be achieved by showing the project tasks and the corresponding period for each task as presented in the contents of Figure (1.1).

### 1.5.1 Project Plan

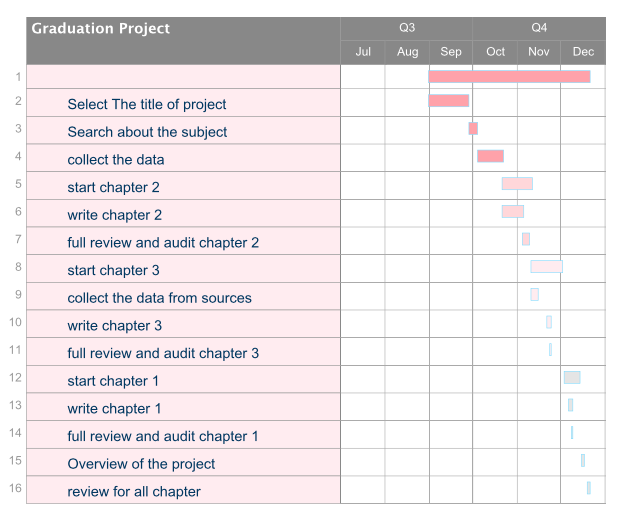
The project plan gives the steps and tasks of the project and the required time intervals estimated in working days. 

Figure 1.1: Project Plan.

### 1.5.2 Project Time Table

The project time table and project plan has been arranged conforming to the following schedule:

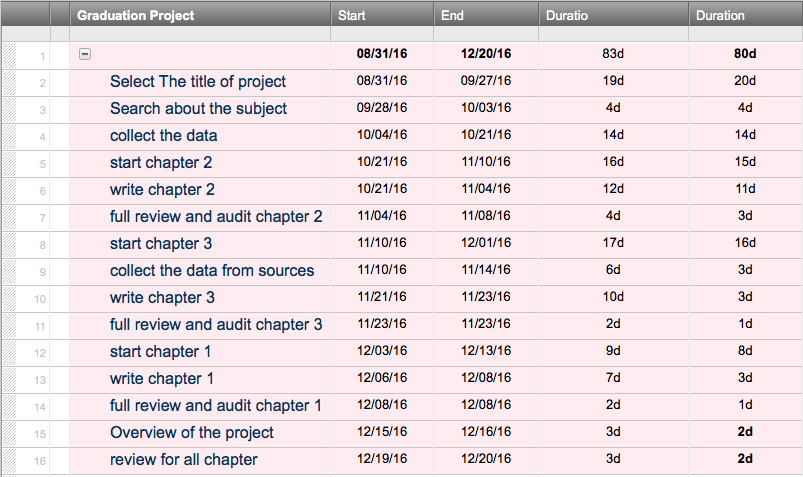


Figure 1.2: Project Time table.

## 1.6 Organizing of the Chapters

The project document is organized in three chapters; the first chapter explains the introduction and objectives. The second chapter is an explanation of data mining applications, techniques, tools, functions, with special focus on FP-Growth algorithm. Chapter three concentrates on project methodology consisting in data gathering and design of the GUI of the system and different class diagrams of project process.

# Chapter Two

LITRATURE REVIEW

## 2.1 What is Data Mining?

Data mining was defined in [1] by “data mining refers to extracting or “mining” knowledge from large amounts of data”. Other terms carry a similar or slightly different meaning to data mining, such as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging”. Second definition is given in [2] by “Data mining is the practice of automatically searching large stores of data to discover patterns and trends that go beyond simple analysis”.

Data mining should be applicable to any kind of data repository, as well as to transient data, such as data streams. Data repositories will include relational databases, data warehouses, transactional databases, advanced database systems, flat files, data streams, and the World Wide Web.

### 2.1.1 Data Mining Applications

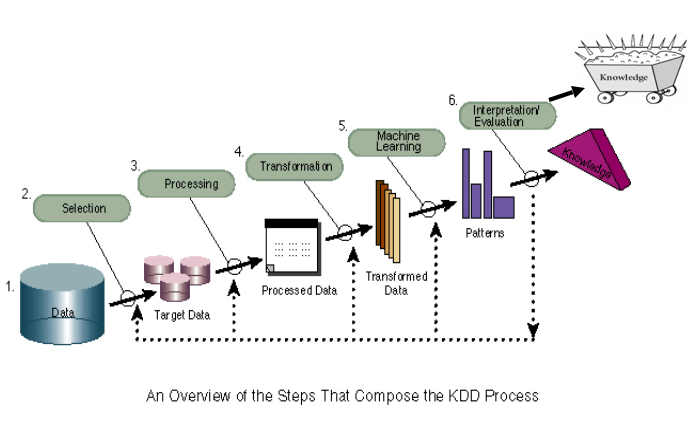
Data Mining is used in many fields worldwide such as [3]:

* Financial Data Analysis: The financial data in banking and financial industry is generally reliable and of high quality which facilitates systematic data analysis and data mining.
* Retail Industry: Data Mining has its great application in Retail Industry because it collects large amount of data from on sales, customer purchasing history, goods transportation, consumption, and services.
* Telecommunication Industry: Today the telecommunication industry is one of the most emerging industries providing various services such as fax, pager, cellular phone, internet messenger, images, e-mail, web data transmission, etc.
* Biological Data Analysis: In recent times, we have seen a tremendous growth in the field of biology such as genomics, proteomics, functional Genomics and biomedical research.
* Intrusion Detection: Intrusion refers to any kind of action that threatens integrity, confidentiality, or the availability of network resources.

### 2.1.2 Data Mining process

Data mining as simply an essential step in the process of knowledge discovery. Knowledge discovery as a process consists [1] of an iterative sequence of the following steps:

* + Data cleaning (to remove noise and inconsistent data)
  + Data integration (where multiple data sources may be combined)
  + Data selection (where data relevant to the analysis task are retrieved from the database)
  + Data transformation (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance)
  + Data mining (an essential process where intelligent methods are applied in order to extract data patterns)
  + Pattern evaluation (to identify the truly interesting patterns representing knowledge based on some interestingness measures
  + Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user



[1] Figure 2.1 The process of data mining

### 2.1.3 Data Mining Environments

The development and application of data mining algorithms requires the use of powerful software tools. The most commonly used tools are: WEKA, Rapid Miner, and Orange. These are open sources software applications developed for data mining.

#### 2.1.3.1 RapidMiner

Written in the Java Programming language, this tool offers advanced analytics through template-based frameworks. Users hardly have to write any code. Offered as a service, rather than a piece of local software, this tool holds top position on the list of data mining tools.

In addition to data mining, RapidMiner also provides functionality like data preprocessing and visualization, predictive analytics and statistical modeling, evaluation, and deployment. What makes it even more powerful is that it provides learning schemes, models and algorithms from WEKA and R scripts [4].

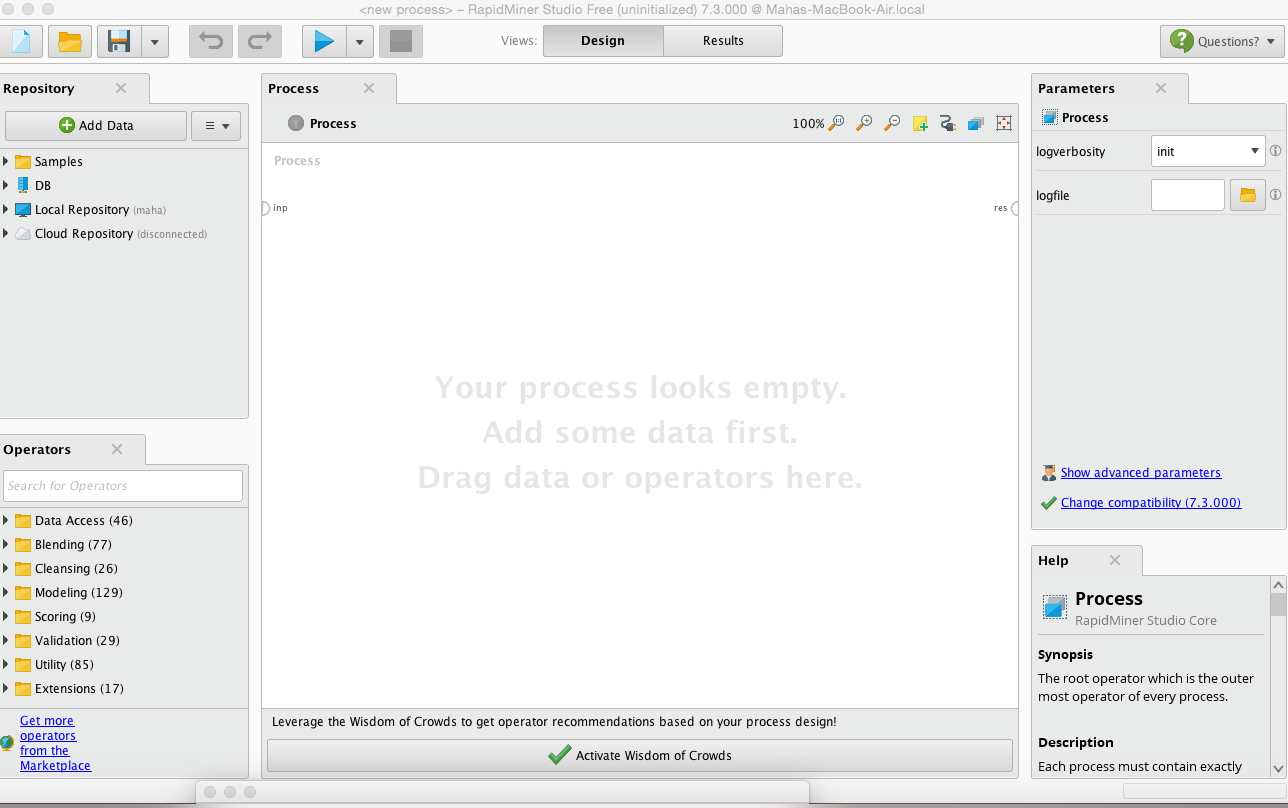


Figure 2.2 RapidMiner Software

#### 2.1.3.2 WEKA

The original non-Java version of WEKA primarily was developed for analyzing data from the agricultural domain. With the Java-based version, the tool is very sophisticated and used in many different applications including visualization and algorithms for data analysis and predictive modeling. It is free under the GNU General Public License, which is a big plus compared to RapidMiner, because users can customize it [4].

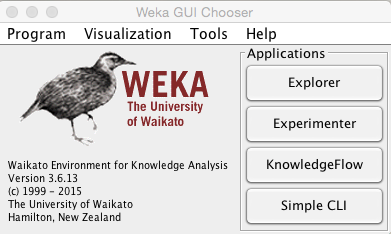


Figure 2.3 WEKA software

#### 2.1.3.3 Orange

A [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) and data mining package (written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language))). It has a visual programming front-end for explorative [data analysis](https://en.wikipedia.org/wiki/Data_analysis) and [visualization](https://en.wikipedia.org/wiki/Information_visualization), and can also be used as a Python library. The program is maintained and developed by the Bioinformatics Laboratory of the Faculty of Computer and Information Science at University of Ljubljana [5].

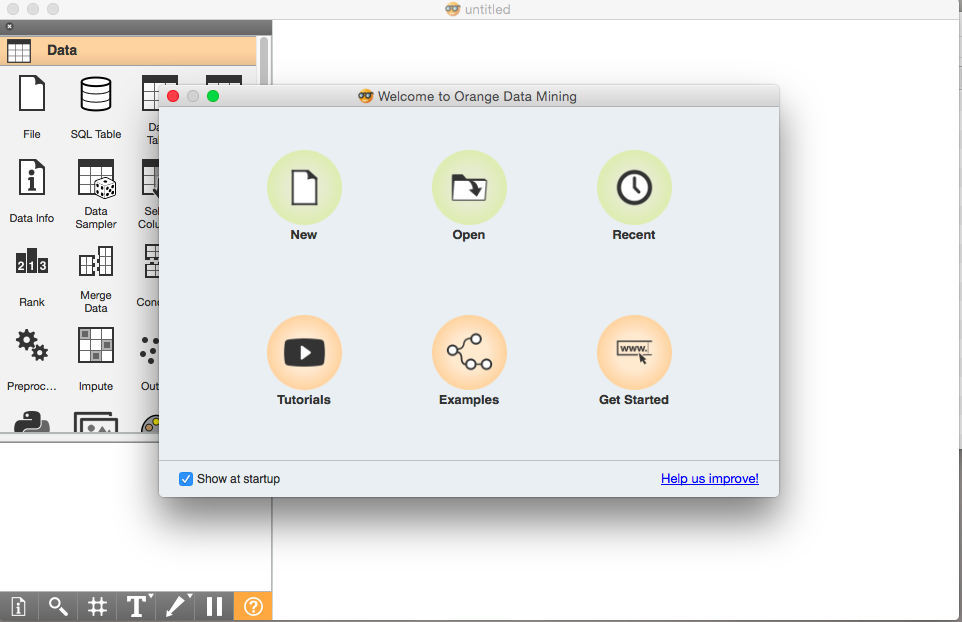


Figure 2.4: Orange Software

## 2.2 Data Mining Algorithms

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general, data mining tasks can be classified into two categories: descriptive and predictive. Descriptive mining tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make predictions [1].Below, are some of data mining functions:

### 2.2.1 Classification

Classification is a model assigning an object to a certain class based on its similarity to previous examples of other objects. It can be done with reference to original data or based on a model of that data. In other words, data classifies (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data. Typical applications of classification are: credit approval, target marketing, medical diagnosis, and treatment effectiveness analysis.

A two-step process of classification

1. Model construction: describing a set of predetermined classes. This model is represented as classification rules, decision trees, or mathematical formula and using set of samples called training set.
2. Model usage: for classifying future or unknown objects. The known label of test sample is compared with the classified result from the model. Then, the percentage of these samples which are correctly classified called an accuracy rate. [6].

### 2.2.2 Clustering

Clustering is a data mining technique used to place data elements into related groups without advance knowledge of the group definitions. Popular clustering techniques include k-means clustering and expectation maximization (EM) clustering [7]. Clustering is an unsupervised learning that finds natural grouping of instances given un-labeled data. Clustering, in other words, is defined as “ the process of grouping physical or abstract objects into classes of similar objects”.

### 2.2.3 Outlier Analysis

An outlier is a data point which is significantly different or inconsistent from the mining data set. Many data mining methods discard outliers as noise or exceptions. The outliers may be of particular interest, such as in the case of fraud detection, where outliers may indicate fraudulent activity. Thus, outlier detection and analysis is an interesting data mining task, referred to as outlier mining or outlier analysis.

Two basic types of procedures for detecting outliers

1. Block procedures: In this case, either the entire suspect objects are treated as outliers or all of them are accepted as consistent
2. Consecutive (or sequential) procedures: An example of such a procedure is the inside out procedure

### 2.2.4 Association Rules (AR)

Association analysis is useful for discovering interesting relationships hidden in large data sets. It aims to extract correlations, frequent patterns, associations, or casual structures among sets of items in the transaction databases or other data repositories. These relationships are not based on inherent properties of the data themselves, but rather based on co-occurrence of the data items. Association rules are often used by retail stores to analyze market basket transactions. The discovered association rules can be used by management to increase the effectiveness and reduce the cost associated with advertising, marketing, inventory, and stock location on the floor. Association rules are also used for other applications such as prediction of failure in telecommunications networks by identifying what events occur before a failure [8].

The problem is usually decomposed into two sub problems. One is to find those item sets whose occurrences exceed a predefined threshold in the database; those item sets are called frequent or large item sets. The second problem is to generate association rules from those large item sets with the constraints of minimal confidence [9].

## 2.3 General AR Algorithms

Association rule mining is to ﬁnd out association rules that satisfy the predeﬁned minimum support and conﬁdence from a given database [10]. The problem is usually decomposed into two sub-problems. One is to ﬁnd those item sets whose occurrences exceed a predeﬁned threshold in the database, those item sets are called frequent or large item sets. The second problem is to generate association rules from those large item sets with the constraints of minimal conﬁdence.

### 2.3.1 Process

Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Association rule generation is usually split up into two separate steps:

1. A minimum support threshold is applied to find all *frequent item-sets* in a database.
2. A minimum confidence constraint is applied to these frequent item-sets in order to form rules.

### 2.3.2 Some Important AR Algorithms [11]

In the first pass, the support of each individual item is counted and also the large ones are detected. In each succeeding pass, the large item-sets detected in the preceding pass are employed to generate new item-sets called candidate item-sets.

The support of each candidate item-sets id counted and the large ones are determined. This process is repeated until no new large item-sets are found the various mining techniques that are enlisted below:

* 1. AIS: For calculation of association rule mining in [10] suggested the AIS algorithm. The algorithm strives to enhance the quality of databases in association with the required functionality in order to undertake decision support queries.
  2. Apriori: This algorithm is employed for learning association rule and frequent item set mining. It differs from the rest in the sense that very few candidates set of item sets is produced for each database scan.
  3. Apriori hybrid: In Apriori and Apriori TID, the same candidate generation procedure is used in all the passes of the database that results in counting of the same item sets. But this is not necessary to be followed.
  4. FP-Growth: FP-growth algorithm is employed to overcome the drawbacks of Apriori algorithm. FP-tree must be created in FP-growth. This algorithm overcomes the drawbacks of an Apriori by not generating the candidate set and taking few passes over the database.

### 2.3.3 FP-Growth [12]

One of the currently fastest and most popular algorithms for frequent item set mining is the FP-growth algorithm. It is based on a preﬁx tree representation of the given database of transactions (called an FP-tree), which can save considerable amounts of memory for storing the transactions. The basic idea of the FP-growth algorithm can be described as a recursive elimination scheme.

#### 2.3.3.1 Definition

It is an efficient method wherein the mining is done by an extended prefix-tree structure on a complete set of frequent patterns by patterns fragment growth. The tree structure stores the compressed information about frequent patterns [13].

A frequent-pattern tree (or FP-tree in short) is a tree structure deﬁned[1] below.

* 1. It consists of one root labeled as “null”, a set of item-preﬁx subtrees as the children of the root, and a frequent-item-header table.
  2. Each node in the item-preﬁx subtree consists of three ﬁelds: item-name, count, and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name, or null if there is none.
  3. Each entry in the frequent-item-header table consists of two ﬁelds, (1) item-name and (2) head of node-link (a pointer pointing to the ﬁrst node in the FP-tree carrying the item-name).

**Algorithm: FP growth.** Mine frequent item sets using an FP-tree by pattern fragment growth.

**Input:**

* D, a transaction database;
* *min\_sup*, the minimum support count threshold.

**Output:** The complete set of frequent patterns.

**Method:**

1. The FP-tree is constructed in the following steps:

(a) Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the list of frequent items.

(b) Create the root of an FP-tree, and label it as “null.” For each transaction *Trans* in D do the following.

* Select and sort the frequent items in Trans according to the order of L.
* Let the sorted frequent item list in *Trans* be [p| P], where p is the first element and P is the remaining list.
* Call ***insert\_tree*** ([p| P], T), which is performed as follows.

If T has a child N such that N. *item -name* = p. *item-name*, then increment N ’s count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same *item-name* via the node-link structure. If P is nonempty, call ***insert\_tree*** (P, N) recursively.

2. The *FP\_tree* is mined by calling FP growth (*FP\_tree, null*), which is implemented as follows. procedure FP growth (*Tree*, α)

1. **if** Tree contains a single path P **then**
2. **for each** combination (denoted as β) of the nodes in the path P
3. generate pattern β ∪ α with *support\_count = minimum support count* of nodes in β;
4. **else** for eachin the header of *Tree* {
5. generate pattern β = ∪ α with *support\_count*=.support count;
6. construct β’s conditional pattern base and then β’s conditional *FP\_tree* ;
7. **if**  ̸= ∅ then
8. call FP growth (, β);}

Example: Let the transaction database, DB, be (the first two columns of) Table 1 and the minimum support threshold be 3.

Table 2.1 Transactional Data

|  |  |  |
| --- | --- | --- |
| TID | Item Bought | (Ordered) Frequent Items |
| 100 | *f, a, c, d, g, i, m, p* | *f, c, a, m, p* |
| 200 | *a, b, c, f, l, m, o* | *f, c, a, b, m* |
| 300 | *b, f, h, j, o* | *f, b* |
| 400 | *b, c, k, s, p* | *c, b, p* |
| 500 | *a, f, c, e, l, p, m, n* | *f, c, a, m, p* |

A compact data structure can be designed based on the following observations:

1. Since only the frequent items will play a role in the frequent pattern mining, it is necessary to perform one scan of DB to identify the set of frequent items (with *frequency count* obtained as a by-product).
2. If we store the set of frequent items (i.e., notice that the ordering is unimportant) of each transaction in some compact structure, it may avoid repeated scanning of DB.
3. If multiple transactions share an identical frequent item set, they can be merged into one with the number of occurrences registered as *count*. It is easy to check whether two sets are identical if all the frequent items in

different transactions are sorted according to a fixed order.

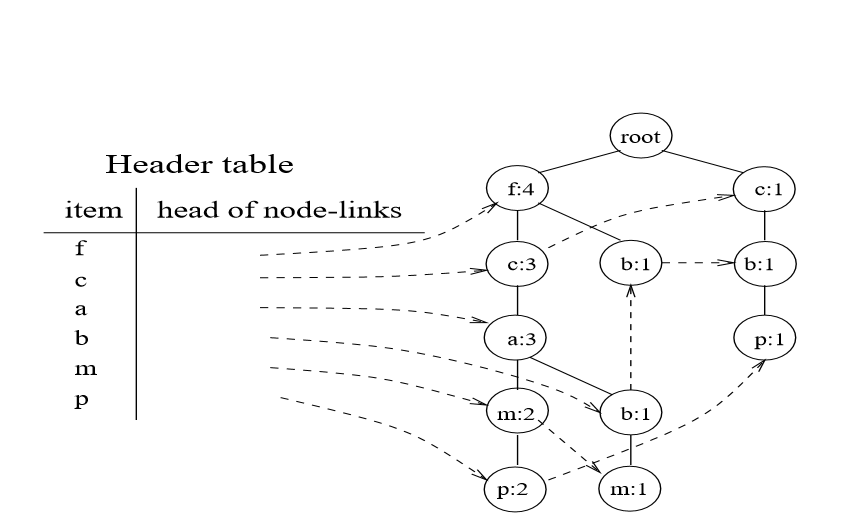
1. If two transactions share a common preﬁx, according to some sorted order of frequent items, the shared parts can be merged using one preﬁx structure as long as the count is registered properly. If the frequent items are sorted in their frequency descending order, there are better chances that more preﬁx strings can be shared.

With the above observations, one may construct a frequent-pattern tree as follows. First, a scan of DB derives a list of frequent items. The set, {(f: 4), (c: 4), (a: 3), (b: 3), (m: 3), (p: 3)} where the number after each item indicates the support. Elements in the set are ordered in frequency descending order. This ordering is important since each path of a tree will follow this order. For convenience of later discussions, the frequent items in each transaction are listed in this ordering in the rightmost column of Table 1.

Second, the root of a tree is created and labeled with “*null*”. The FP-tree is constructed as follows by scanning the transaction database DB the second time.

1. The scan of the ﬁrst transaction leads to the construction of the ﬁrst branch of the tree: {*(f:1), (c:1), (a:1), (m:1), (p:1)*}*.* Notice that the frequent items in the transaction are listed according to the order in the list of frequent items.
2. For the second transaction, since its (ordered) frequent item list {f, c, a, b, m} shares a common preﬁx {f, c, a} with the existing path {f, c, a, m, p}, the count of each node along the preﬁx is incremented by 1, and one new node (*b:1*) is created and linked as a child of (*a:2*) and another new node (*m:1*) is created and linked as the child of (*b:1*).
3. For the third transaction, since its frequent item list {f, b} shares only the node{f}with the *f*-preﬁx subtree, *f’*s count is incremented by 1, and a new node (*b:1*) is created and linked as a child of (*f:3*).
4. The scan of the fourth transaction leads to the construction of the second branch of the tree, {(c:1), (b:1), (p:1)}.
5. For the last transaction, since its frequent item list *(f, c, a, m, p)* is identical to the ﬁrst one, the path is shared with the count of each node along the path incremented by 1.

To facilitate tree traversal, an item header table is built in which each item points to its ﬁrst occurrence in the tree via a node-link. Nodes with the same item-name are linked in sequence via such *node-links*. After scanning all the transactions, the tree, together with the associated node-links, are shown in the ﬁgure down below.



[14] Figure 2.5: Header Table

## 2.4 Related Works (Association Rule)

Many systems were developed using FP-Growth. In this section we will see how to use this algorithm in WEKA environment.

### 2.4.1 Applying FP-Growth in WEKA

First, we choose dataset. In this example, we choose the data with 10 attributes from a0 to a9. Each one of these components has binary values true or false.

These are the data that we choose to mine using FP-Growth:

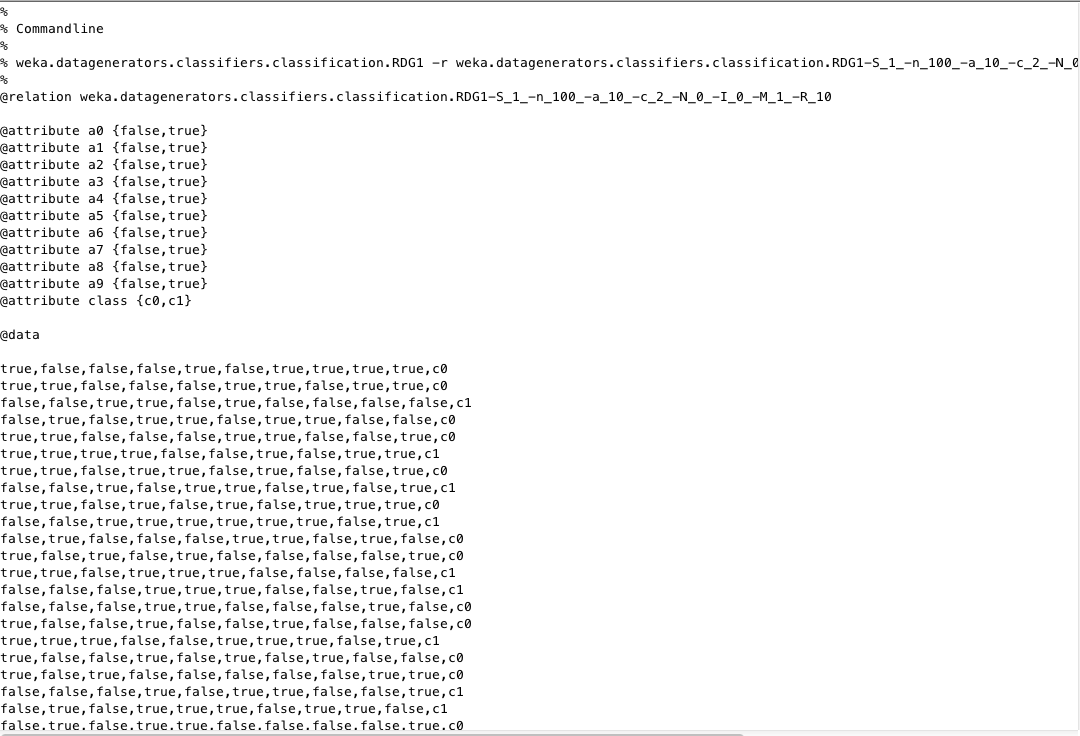


Figure 2.6: Data set

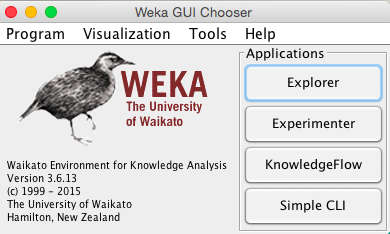


Figure 2.7: WEKA Interface [1]

After choosing explorer as an application, functions of datamining will appear to determine what function we will use.

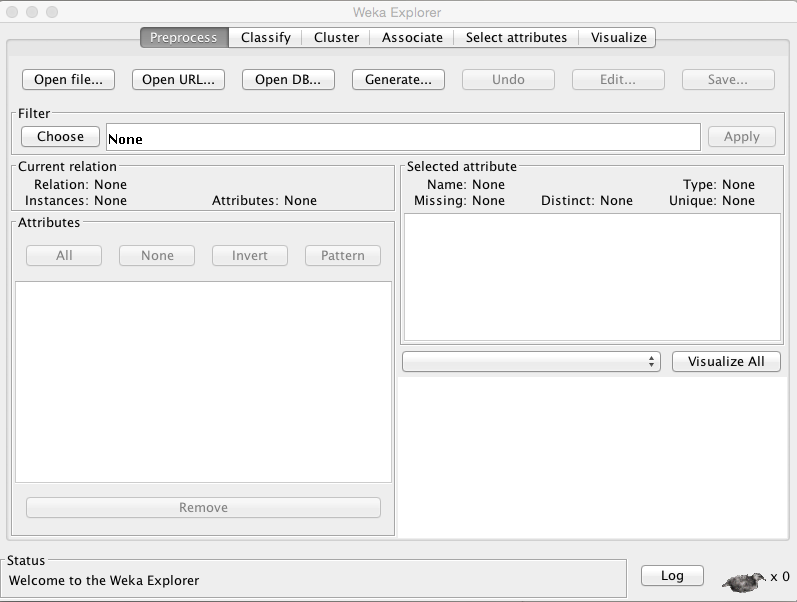


Figure 2.8: WEKA’s List of Functions

After clicking the associate, we choose FP-Growth algorithm as the figure below shows

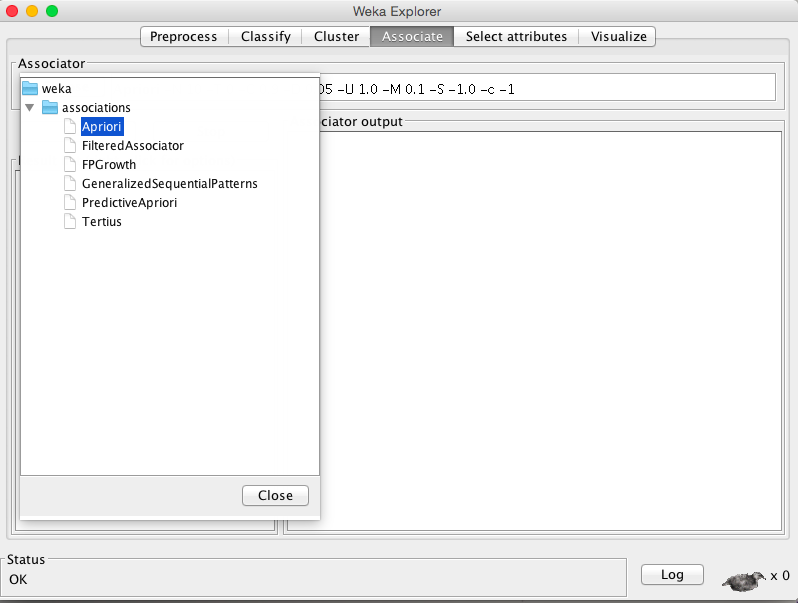


Figure 2.9: Association’s Algorithms

We will explain the first row from the top 10 result are shown in the screen. In line 1 the value true in a7 and the class c1 has 19 attributes, then the class a5 has 19 true value.

the (CONF) is Confidence of a rule indicates its accuracy.

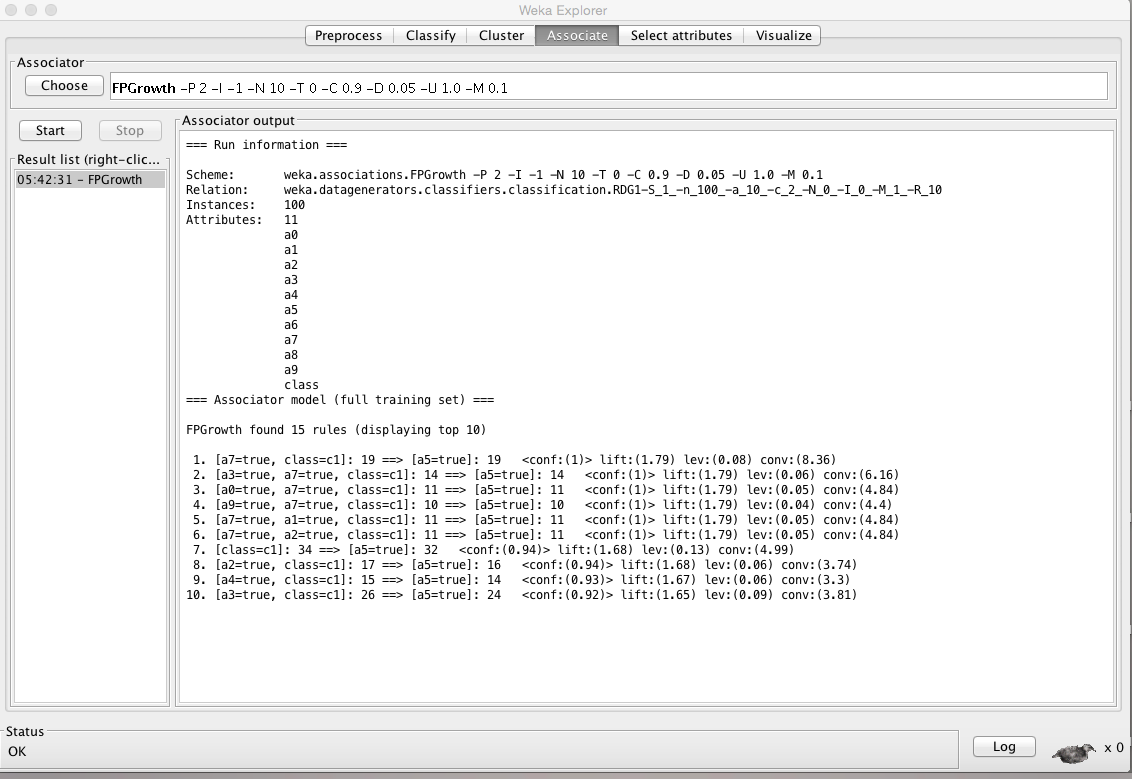
It is defined as Conf(X ⇒ Y) = supp(X ∪ Y) / supp(X) and it is equal  1 , Lift measures how many times more often X and Y occur together than expected if they are statistically independent and it is equal  1.79, Leverage measures the difference of X and Y appearing together in the data set and what would be expected if X and Y were statistically dependent and it is equal 0.08, and Conviction it is  equal 8.36 and all the rules it has same with different value.

Figure 2.10: Result Using FP-Growth

#### 2.4.1.1 FP-Growth Compared to other Algorithms

The FP Growth algorithm is more efficient than other algorithms for the following reasons:

1. Divide and Conquer: The mining data is decomposed into sub-datasets according to the frequent patterns identified. It leads to more focused search of smaller databases.
2. There is no candidate generation. Thus, no candidate test is required.
3. No repeated scans of the whole database.

#### 2.4.1.2 Some Improvements on FP-Growth Algorithm

Even that FP-Growth, when first developed in 2000, was outperforming the existing algorithms in the field; many improvements are done for FP-Growth in the aim to deal with some bottleneck problems faced during the use of the algorithm. In our literature review, we have scanned some recent papers in this direction. In the following section we present sample of these papers according to their chronological aspect.

2004:  An improving a new efficient QFP-growth algorithm [15]

In [15], authors propose a new efficient algorithm QFP-growth not only heirs all the advantages in FP-growth method, but also avoids its bottleneck in generating a huge number of conditional FP-trees. By Using the technology of temporary root, QFP-growth reduces the processing time and memory space for mining frequent item sets significantly.  Testing the performance also shows that the QFP-growth method is efficient and scalable for mining large databases or data warehouses. Moreover, the algorithm generates frequent item sets in order so that the result can be used expediently.

2006: Aggregative chain [16]

In this paper [16], an improvement of the architecture of the FP tree is proposed. The new FP tree is a one-way tree and only the pointers that point its parent at each node are kept. Route information of different nodes in a same item are compressed into aggregative chains so that the frequent patterns will be produced in aggregative chains without generating node links and conditional pattern bases.

2008: Experimental analysis shows outperformance of FP-growth al on Apriori regarding the process of mining the association rules [17]

In this paper, the authors have made many experiments showing that FP-growth algorithm resolves two neck-bottle problems of traditional Apriori algorithm and has more efficiency than original one. This improvement showed higher mining efficiency in execution time, memory usage and CPU utilization [17].

2011: mining all possible frequent item set without generating the conditional FP tree.

In [18], A.B.M. Rezbaul Islam and Tae-Sun Chung proposed a new and improved FP tree with a table and a new algorithm for mining association rules. This algorithm mines all possible frequent item set without generating the conditional FP tree. It also provides the frequency of frequent items, which is used to estimate the desired association rules. It reduces the total step of the procedure and also takes less memory.

2015: Painting Growth [19]

Here also, Yi Zeng, Shiqun Yin, Jiangyue Liu, and Miao Zhang made a research paper on improved FP-Growth algorithm in association rules mining [19]. The efficiency is not high in FP-Growth that because the scanning of database is done twice. Their research put forward two improved algorithms—Painting-Growth algorithm and N Painting-Growth algorithm—which use two-item permutation sets to dig. Both algorithms scan database only once to obtain the results of mining.

2015: Spark framework for FP-Growth [20]

The search operation as the major time-consuming operation has a higher complexity also the mining efficiency of dataset will be reduced or even failed. These problems are solved by [20] distributed SPFP algorithm based on Spark framework and improved FP-growth algorithm. The results of tests show that, SPFP has high efficiency, cluster, and Flexibility.

#### 2.4.1.3 Implementation

[13] Figure 2.11: Implementation

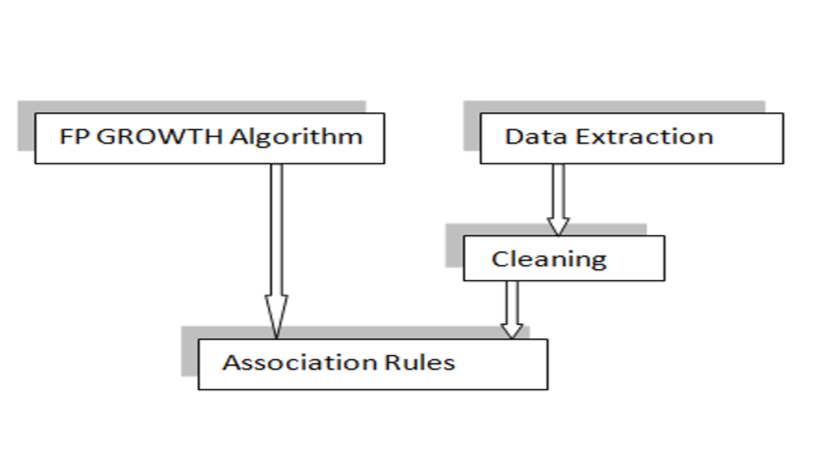


Figure 2.11 shows the implementation process. The implementation starts with the user feedback dataset obtained online and comprised of a range of attributes. This dataset is then cleaned by rectifying and resolving the missing and incorrect values. Then the available FP Growth algorithm is applied on the clean dataset which results in formation of association rules required for analysis.

## 2.5 Conclusion

In this chapter, we covered the theoretical basics of our project among others data mining, its related functions, processes of data and examples of programs used to mine data. And we talked, in detail, about FP-Growth algorithm which is our adopted algorithm for the implementation. At the end of chapter, we have presented some improvements done on FP-growth algorithm.

# Chapter Three

METHODOLOGY

## 3.1 Introduction

This chapter will be concentrated on the project analysis and design process, starting with data and information which are essential for effective operations and decision making at all levels of life aspects. Project structure design also will be explained in this chapter.

## 3.2 Design of Project Structure

Project outline consists of a sequence of steps as shown in fig 3.1 This figure includes all the project KDD processes. It starts from data collection until getting required knowledge.

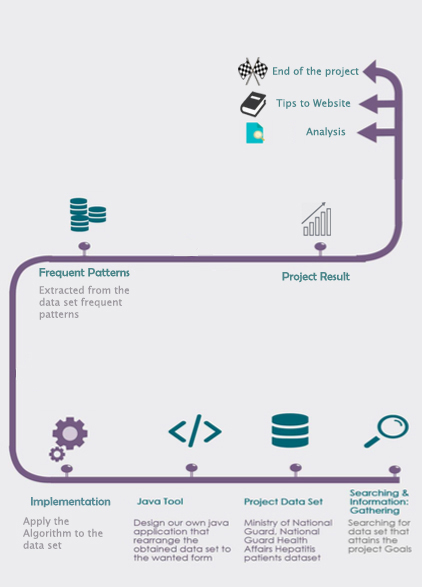


Figure 3.1: Designing Process

## 3.3 Data and information Gathering

The first step in the project structure is collecting historical datasets from various sources. It provides a strong base for the project.

## 3.4 The Interface Design

The main interface when the project is executed. There are 3 ways of showing the result: The tree, the analysis or All.

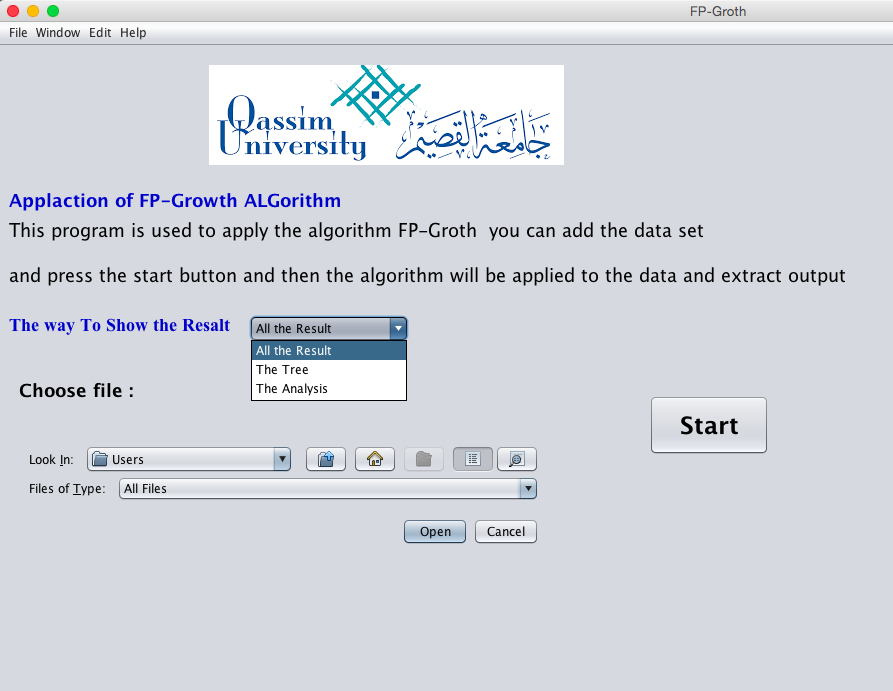


Figure 3.2: Project Interface

The menu bar has 4 contents. The File has New for a new data-set to enter also the Print and Exit from the project.



Figure 3.3: File in Menu Bar

The window has 3 options Minimize, Maximize and Zoom.



Figure 3.4: Window Options

In Edit you can Copy or Cut the result.



Figure 3.5: Edit Options

## 3.5 Gathering Project Data Set

We will apply our project in web-mining. We are requiring the data from the Deanship of Information Technology in our university and we are waiting for the response and agreement for us to use the data.

## Draft Class Diagram of the Algorithm

We propose here a draft for the class diagram that will be implemented in the second phase of the project. Therefore, it can be modified during the phase two implementation.

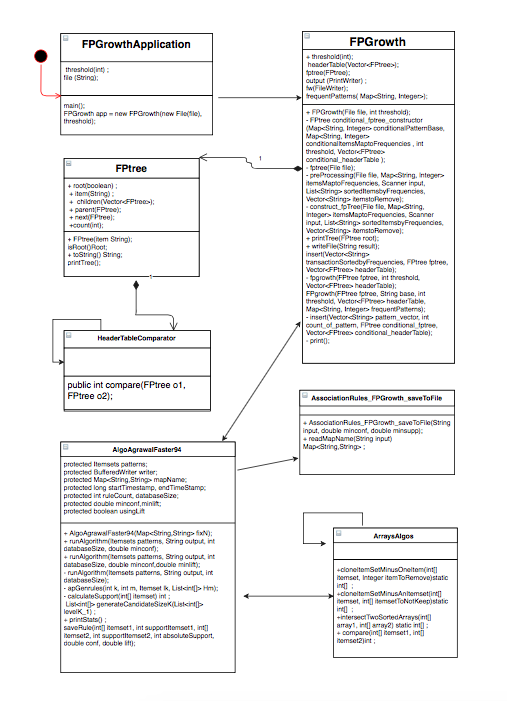
****

Figure 3.6: Proposed FP-Growth Class Diagram

## 3.7 Project Process advise

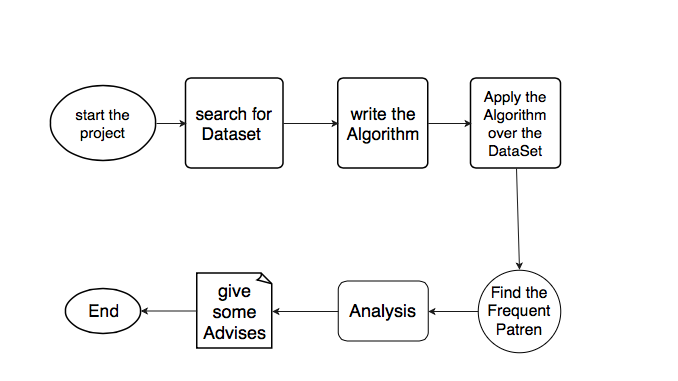


Figure 3.7: Process of the Project

## 3.8 What is Web Logs?

A log, in a computing context, is the automatically produced and time-stamped documentation of events relevant to a particular system. Web log file is log file automatically created and maintained by a web server. Every "hit" to the Web site, including each view of a HTML document, image, or other object, is logged.

[Format](http://publib.boulder.ibm.com/tividd/td/ITWSA/ITWSA_info45/en_US/HTML/guide/c-logs.html#ncsa):

The raw web log file format is essentially one line of text for each hit to the web site. This contains information about who was visiting the site, where they came from, and exactly what they were doing on the web site.

Web Site Analyzer can use the information in HTTP, FTP, and other server log files to [analyze a site](http://publib.boulder.ibm.com/tividd/td/ITWSA/ITWSA_info45/en_US/HTML/guide/c-analsite.html).

The log file formats that Web Site Analyzer can analyze include the following:

* NCSA (Common or Access, Combined, and Separate or 3-Log)
* W3C Extended (used by Microsoft IIS 4.0 and 5.0)
* SunTM ONE Web Server (iPlanet)
* IBM Tivoli Access Manager WebSEAL
* WebSphere Application Server Logs
* FTP Logs

Example on NCSA Log Formats

The NCSA log formats are based on NCSA httpd, and are widely accepted as standard among HTTP server vendors.

Web Site Analyzer supports the following options:

* [Common (access log)](http://publib.boulder.ibm.com/tividd/td/ITWSA/ITWSA_info45/en_US/HTML/guide/c-logs.html#common)
* [Combined](http://publib.boulder.ibm.com/tividd/td/ITWSA/ITWSA_info45/en_US/HTML/guide/c-logs.html#combined)
* [Separate with Date (three log)](http://publib.boulder.ibm.com/tividd/td/ITWSA/ITWSA_info45/en_US/HTML/guide/c-logs.html#separate)

Switching among options is specific to each HTTP server implementation.

The fields in the Common log file format are:   
  
host rfc931 username date:time request statuscode bytes   
  
The following example shows these fields populated with values in a common log file record:

125.125.125.125 - dsmith [10/Oct/1999:21:15:05 +0500] "GET /index.html HTTP/1.0" 200 1043

host (125.125.125.125 in the example)   
The IP address or host/subdomain name of the HTTP client that made the HTTP resource request.  
  
rfc931 ("-" in the example)   
The identifier used to identify the client making the HTTP request. If no value is present, a "-" is substituted.  
  
username (dsmith in the example)   
The username, (or user ID) used by the client for authentication. If no value is present, a "-" is substituted.  
  
date:time timezone ([10/Oct/1999:21:15:05 +0500] in the example)   
The date and time stamp of the HTTP request.  
  
The fields in the date/time field are:   
  
[dd/MMM/yyyy:hh:mm:ss +-hhmm]   
  
where the fields are defined as follow:  
  
dd is the day of the month  
MMM is the month  
yyy is the year  
:hh is the hour  
:mm is the minute  
:ss is the seconds  
+-hhmm is the time zone  
  
  
In practice, the day is typically logged in two-digit format even for single-digit days. For example, the second day of the month would be represented as 02. However, some HTTP servers do log a single digit day as a single digit. When parsing log records, you should be aware of both possible day representations.  
  
request ("GET /index.html HTTP/1.0" in the example)   
The HTTP request. The request field contains three pieces of information. The main piece is the requested resource (index.html). The request field also contains the HTTP method (GET) and the HTTP protocol version (1.0).  
  
statuscode (200 in the example)   
The status is the numeric code indicating the success or failure of the HTTP request.  
  
bytes (1043 in the example)   
The bytes field is a numeric field containing the number of bytes of data transferred as part of the HTTP request, not including the HTTP header.

# Chapter Four

Implementation AND TESTING

## 4.1 Introduction

Now we are reaching the phase of implementing and testing the project and the researches which what we have finished in the first three chapters. We will use java language to write the code of our project. We are taking the WEKA FP-Growth algorithm code and try to improve it. Weka uses arff” attribute relation file format” which is an ASCI text file describes a list of instances sharing a set of attributes as a format to its datasets [21]. arff files contains 2 parts header and data. Data sets must be in arff. In order-to use a dataset in WEKA it must reformat the attribute into a binary instead of numeric. Below some examples of datasets in WEKA airlines passengers and contact lenses.

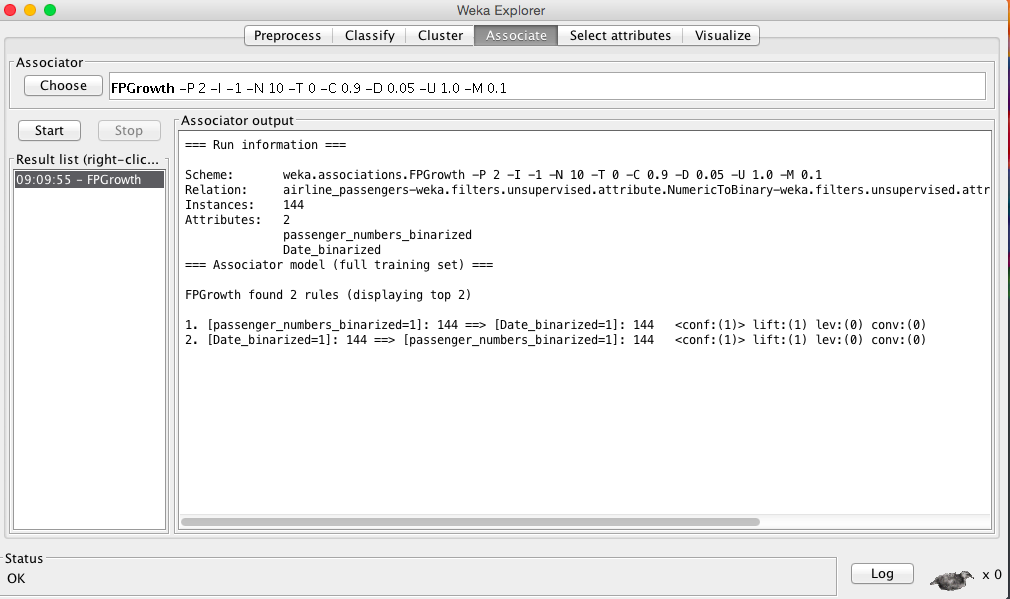


Figure4.1: WEKA Airline Passengers Data Set

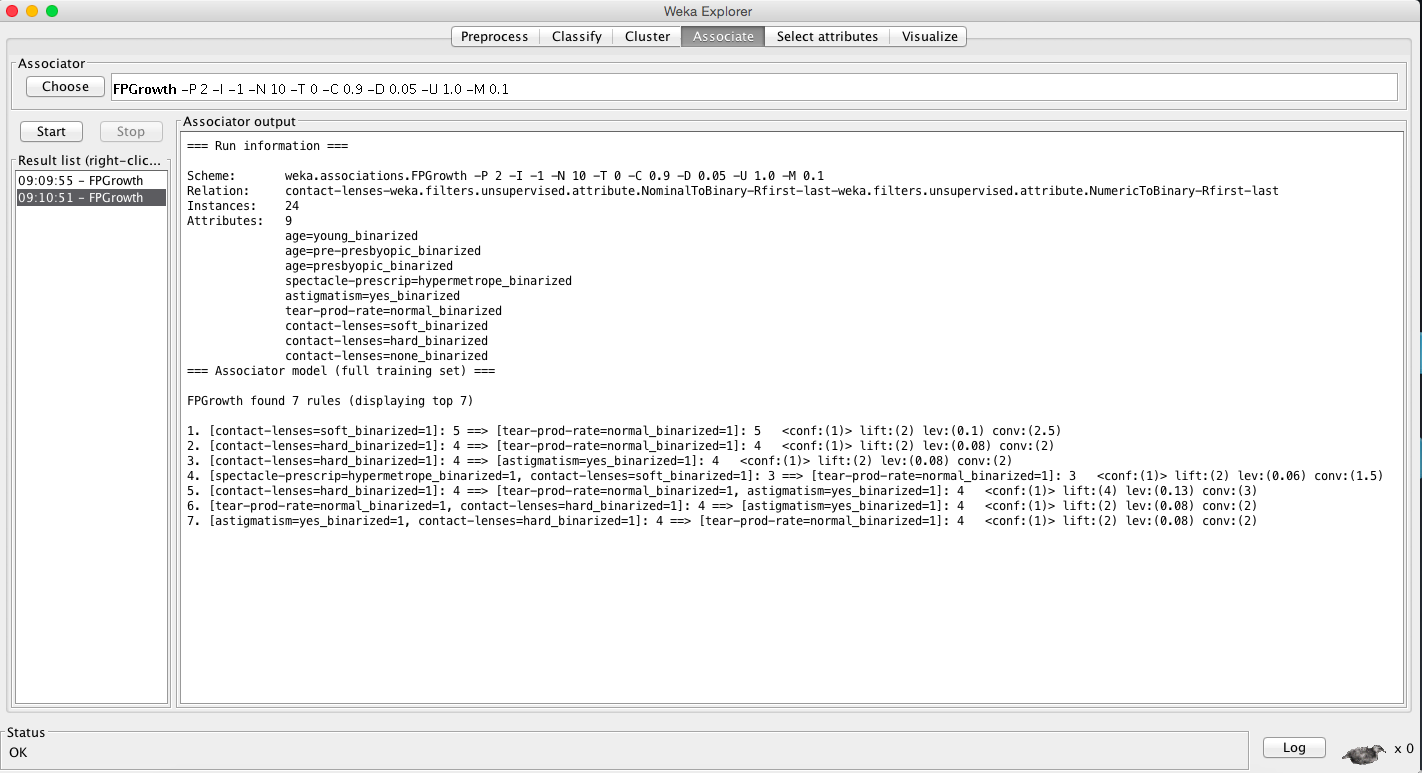


Figure 4.2: WEKA Contact Lenses Data Set

## 4.2 Implementation

For creating our project, we created several classes. Four of them are for creating fp-growth algorithm FPGrowthApplication, FPTree, HeaderTableComarator. And the other three for the interface.

### 4.2.1 FPGrowthApplication

public class FPGrowthApplication

In this class, we have the main method, but before the main we defined an integer value of threshold and this gives us the value of repetition that we want. We also defined a file of data-set for testing the algorithm. The last thing is calculating the time it takes for the program to finish processing.

### 4.2.2 FPTree

In the beginning, we declared the tree variables. The constructor then has the item from the data-set. boolean isRoot() is a method to return the root. public String toString() convert the tree items to string values to print in the method public void printTree().

### 4.2.3 HeaderTableComarator

class HeaderTableComparator implements Comparator<FPtree>

This class for comparing the total frequency of elements to order items from largest to smallest.

### 4.2.4 FPGrowth

public FPGrowth(File file, int threshold)

Contractor of this class receive the path of data set and the value of repetition.

private **FPtree conditional\_fptree\_constructor(Map<String, Integer> conditionalPatternBase, Map<String, Integer> conditionalItemsMaptoFrequencies,** int **threshold, Vector<FPtree> conditional\_headerTable)**

In this function it is receive map type has <String, Integer> word as a string and his frequent ) and threshold value. Its work is to construct FP-Tree and removing non-frequents and making a vector instead of string. The tree root is null and Arrange the elements in the tree and call the insert function to add them to the tree then return the FP-Tree.

privatevoid **fptree (File file)**

a method to read the data sets and look for repeated element then sort it and remove the item less than the value of repetition(threshold).

privatevoid **preprocessing (File, Map<String, Integer> itemsMaptoFrequencies, Scanner input, List<String> sortedItemsbyFrequencies, Vector<String> itemstoRemove)**

this method print all items in a file with their frequencies then print the list of sorted items then remove item with frequencies less than the threshold.

privatevoid **construct\_fpTree (File, Map<String, Integer> itemsMaptoFrequencies, Scanner input, List<String> sortedItemsbyFrequencies, Vector<String> itemstoRemove)**

print the elements of the header. Take each line from the transactions and order them from the largest to the smallest.

publicvoid **printTree (FPtree root)**

function to print the tree for each group of items

void insert(Vector<String> transactionSortedbyFrequencies, FPtree, Vector<FPtree> headerTable)

if the tree is empty return no thing, and if it’s find the item it’s count will be increased by one. If the item is not in tree it will add it and make it’s count to one. Then item will be deleted from transactionSortedbyFrequencies.

privatevoid **print ()**

to print frequentPattern after finish from tree.

### 4.2.5 AlgoAgrawalFaster94

first define patterns from Itemsets class then define writer object to write the output file if the user wish to write to a file then define string to keep the name not is number from input file then define some object to statistics like startTimestamp (last execution start time) endTimeStamp (last execution end time) ruleCount (number of rules generated) databaseSize (number of transactions in database) after that define the parameters minconf, minlift, and usingLift

public AlgoAgrawalFaster94(Map<String, String> fixN)

in this class, we just assigning the mapName equal to fixN

public void runAlgorithm (Itemsets patterns, String output, int databaseSize, double minconf) throws IOException

this class to assigning minconf equal the minconf the user inters and put minlift = 0 and put usingLift = false because we do not use this in the project after that call runAlgorithm and give this parameter patterns, output, and databaseSize.

privatevoid **runAlgorithm (Itemsets patterns, String output,** int **databaseSize)** throws **IOException**

first check if the user want to keep the result into memory or user want to save the result to a file then record the time when the algorithm starts then initializing variable to count the number of rules found (ruleCount = 0) then save itemsets in a member variable

we have for loop to order the item in the data set after finish this for it is end of sorting the item, after that we can start generating the rules for each frequent itemset of size >=2 that we will name “lk" and for each itemset "itemsetSize1" of size 1 that is member of lk Now we will calculate the support and confidence

before define support from integer type that equal calculateSupport(itemset\_Lk\_minus\_hm\_P\_1)

then convert this integer to double define as supportAsDouble then apply the law of conf = lk. getAbsoluteSupport () / supportAsDouble

then check if the conf lees than the minconf the user inters continuous after finish save the rule and close the file if we saved the result to a file and record the end time of the algorithm execution.

private void apGenrules (int k, int m, Itemset lk, List<int []> Hm) throws IOException

The ApGenRules as described in p.14 of the paper by Agrawal. (see the Agrawal paper for more details). k the size of the first itemset used to generate rules

m the recursive depth of the call to this method (first time 1, then 2...). lk the itemset that is used to generate rules. Hm a set of itemsets that can be used with lk to generate rules. IOException exception if error while writing output file.

private int calculateSupport (int [] itemset)

this method receives itemset and return the support of the itemset, we first get the list of patterns having the same size as “itemset" we perform a binary search to find the position of itemset in this list.

In this class define first equal =0 and last = the last item and the middle equal the middle item. After that define that comparaison if the comparaison > 0 put first = middle + 1 else comparaison < 0 put last = middle - 1.

protected **List<**int **[]> generateCandidateSizeK (List<**int **[]> levelK\_1)**

This method levelK\_1 a set of itemsets of size k-1 and return a set of candidates

first create a variable to store candidates.

publicvoid **printStats ()**

Print statistics about the algorithm execution to System.out. print Number of association rules generated and Total time.

protected void saveRule (int [] itemset1, int supportItemset1, int [] itemset2, int supportItemset2,

int **absoluteSupport,** double **conf,** double **lift)** throws **IOException**

This method for save rule in the file and print it in the interface.

### 4.2.6 ArraysAlgos

public static int [] cloneItemSetMinusOneItem (int [] itemset, Integer itemToRemove)

Take the item set and check if not remove put it in a new item set.

public static int [] cloneItemSetMinusAnItemset (int [] itemset, int [] itemsetToNotKeep)

Make a copy from the items and check if the set from the item that excluded.

public static int [] intersectTwoSortedArrays (int [] array1, int [] array2)

This method performs the intersection of two sorted arrays of integers and return a new sorted array, the first param the first array, b the second array, and return the resulting sorted array, return the subrange of the new array that is full.

public static Comparator<int []> *comparatorItemsetSameSize* = new Comparator<int []> ()

A Comparator for comparing two itemsets having the same size using the lexical order.

public int compare (int [] itemset1, int [] itemset2)

Compare two itemsets and return -1,0 and 1 if the second itemset is larger, equal, or smaller than the first itemset according to the lexical order.

### 4.2.7 AssociationRules\_FPGrowth\_saveToFile

public **AssociationRules\_FPGrowth\_saveToFile (String input,** double **minconf,** double **minsupp)**

First define the rule file then call FPGrowth to Apply the fp-growth algorithm to find frequent itemsets then call AlgoAgrawalFaster94 to Generating all rules from the set of frequent itemsets.

public Map<String, String> readMapName (String input) throws IOException

Checking the data set and read it to see if it is in a correct format (not having \* & ^ % $ # @ ! > < ? ” : )

### 4.2.8 Welcoming Interface

In Phase 1 the interface design was this figure 3.2 :

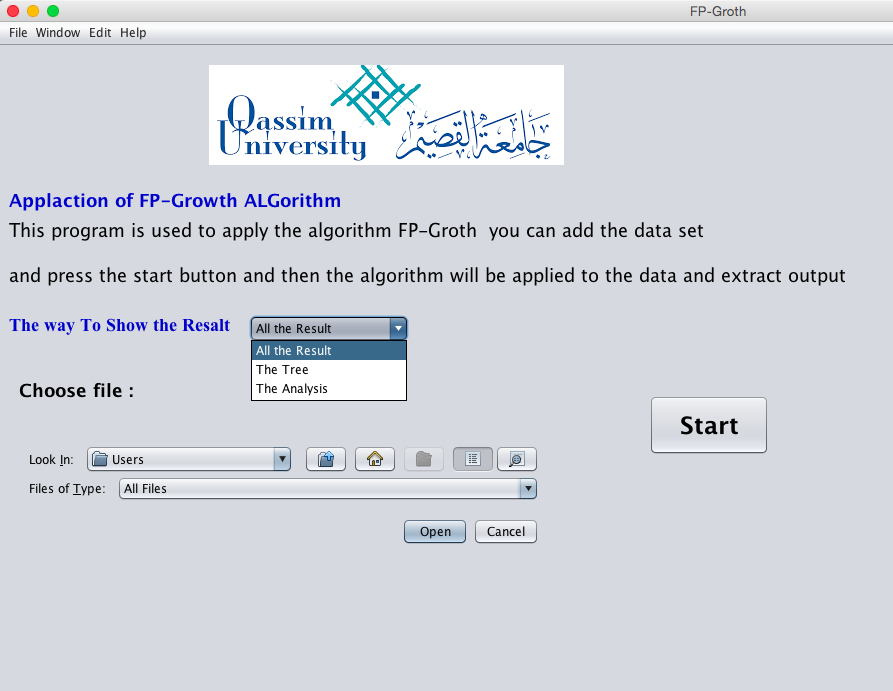


Figure 3.2: Project Interface

But now we changed the interface design for several reasons:

1. Easier to use.
2. More Organized to the user.
3. Walk step by step with the user and not show all the results and inputs in one page.

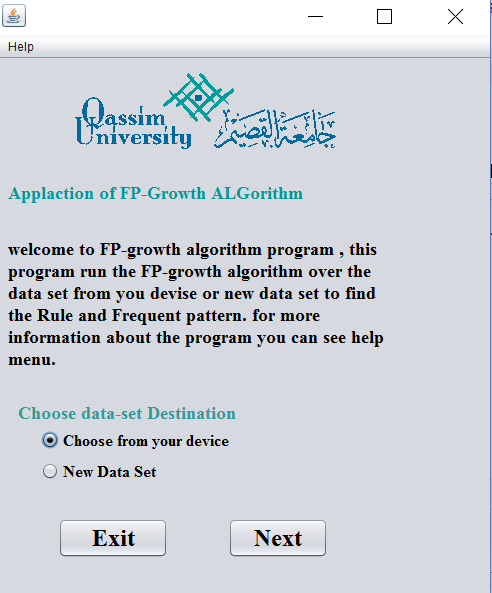


Figure 4.3: latest Project Interface

## 4.3 Data Sets

### 4.3.1 Type of Data Set

After looking for a data set like the data we decided to use in the first phase we found a data set from UCI (University of California, Irvine) site Machine Learning repository we found the anonymous Microsoft Web Data Set, that is created the data by sampling and processing the www.microsoft.com logs. The data records the use of www.microsoft.com by 38000 anonymous, Randomly-selected users. For each user, the data lists all the areas of the web site (Vroots) that user visited in a one week timeframe. the Number of Attributes: 294.

# Chapter Five

RESULTS AND DISCUSSION

## 5.1 Test the Code with Small and New Data set:

### 5.1.1 The Frequent

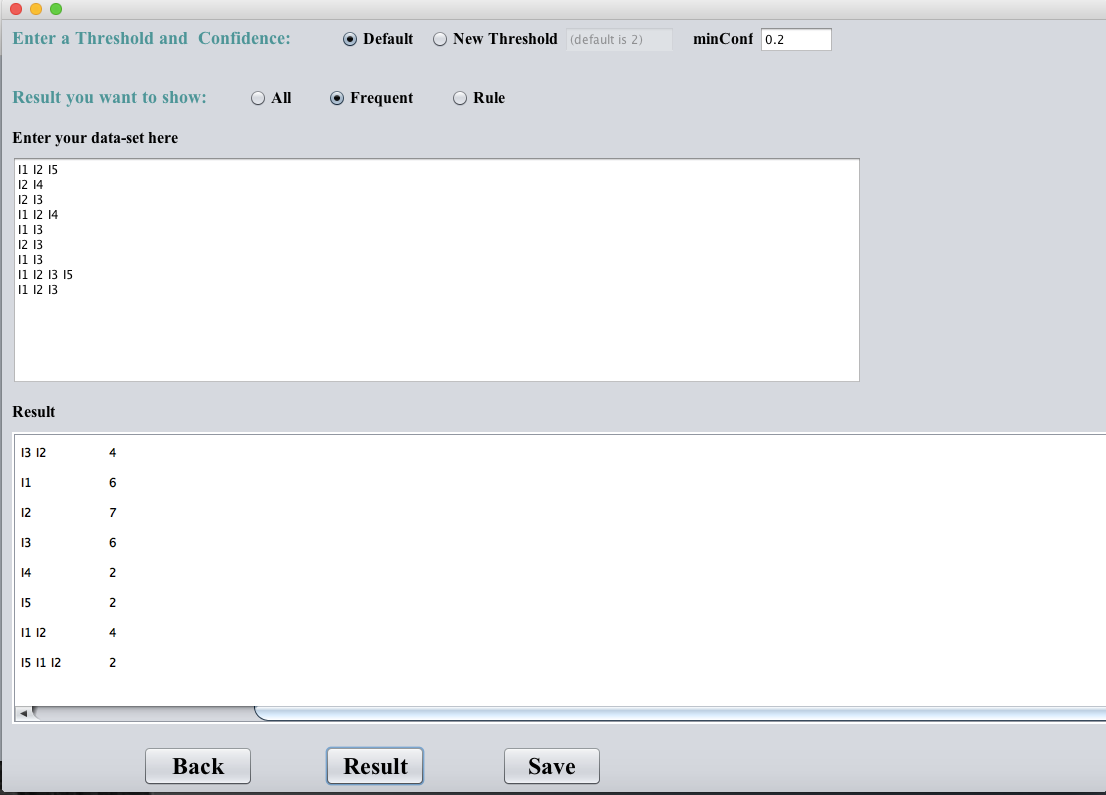
Showing with default Threshold (=2):

Figure 5.1: Result of frequent with threshold 2

Another example with new Threshold =4:

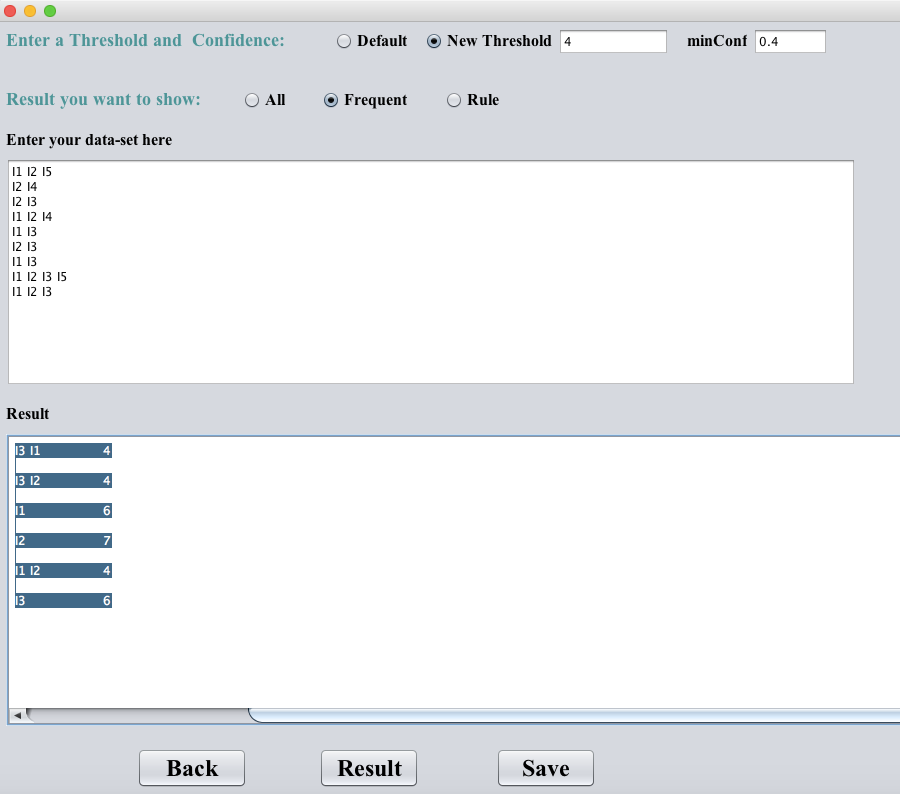


Figure 5.2: Result of frequent with threshold 4

After seeing the result of many threshold, we find that the number of items in the data from i1 to i5 and the number of frequent change from Threshold to other Threshold.

In the option of frequent the algorithms take each element with the rest of the elements and then take two elements with the rest of the elements and thus find the number of recurrence of this group created by only one element or of many elements. Of course, except for elements whose frequency is less than Threshold.

### 5.1.2 The Rule

This option takes the minimum confidence and extract the rule by applying the formula of confidence. And the rule is expression of the algorithms, it takes each element with the remaining elements and each element with the remaining elements are called rule and then the formula of the confidence.

Here in the first example number of rules equal to 49, but the second example the number of rule equals 35.

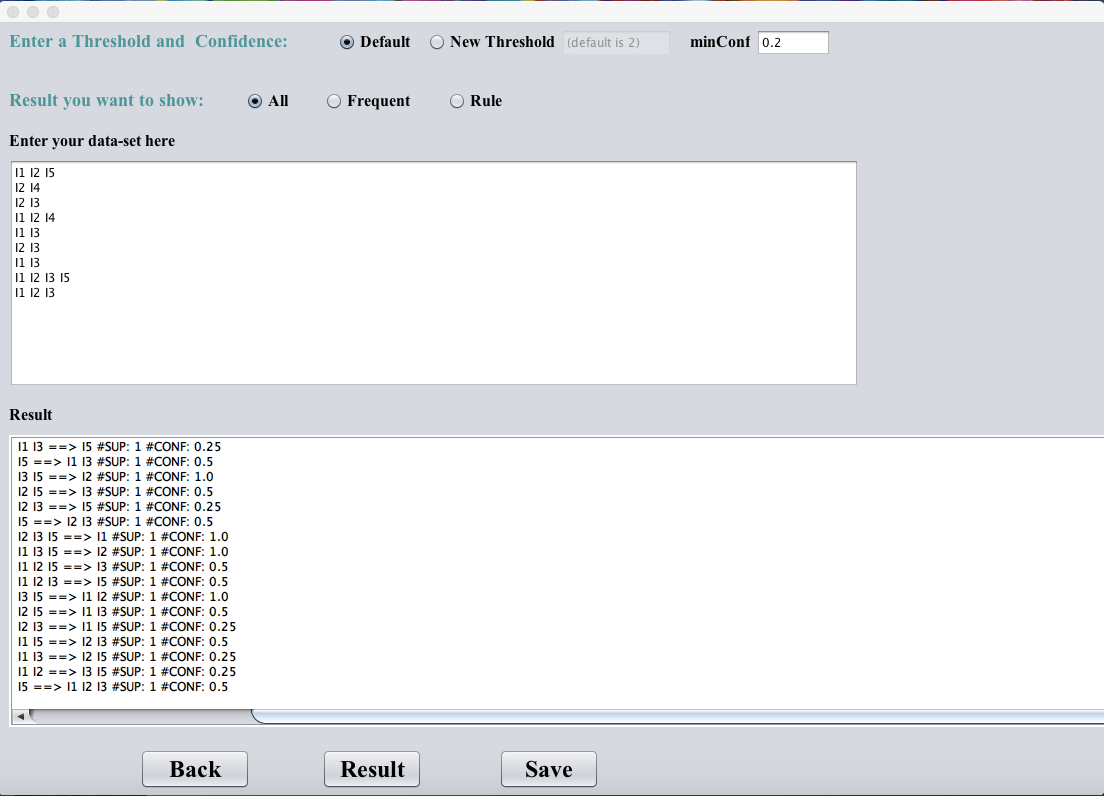
with default Threshold =2 and with confidence =0.2

Figure 5.3: Result of rule with default threshold 2 and with confidence 0.2

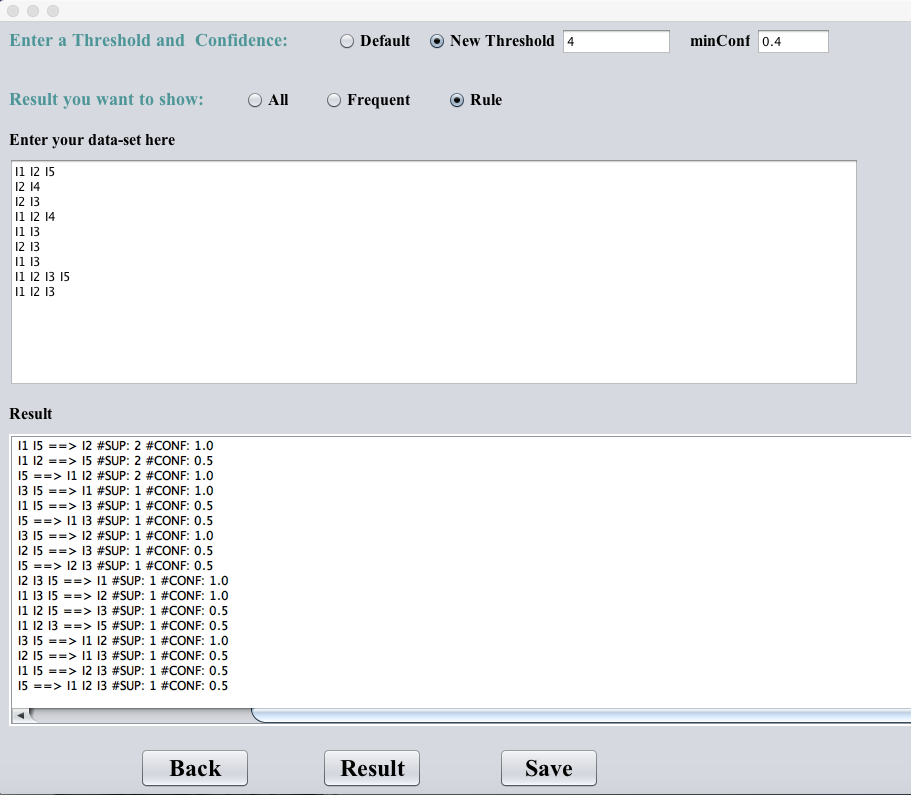
with new Threshold =4 and with confidence =0.4:

Figure 5.4: Result of rule with threshold 4 and with confidence 0.4

### 5.1.3 All the Results

This option to understand and display the steps of the algorithms, it calculates the frequency of each element of the data set and then order the items from the largest to the smallest, and then displays the items more than Threshold, then displays the remaining elements and then tree elements.

With default Threshold and min confidence =0.2

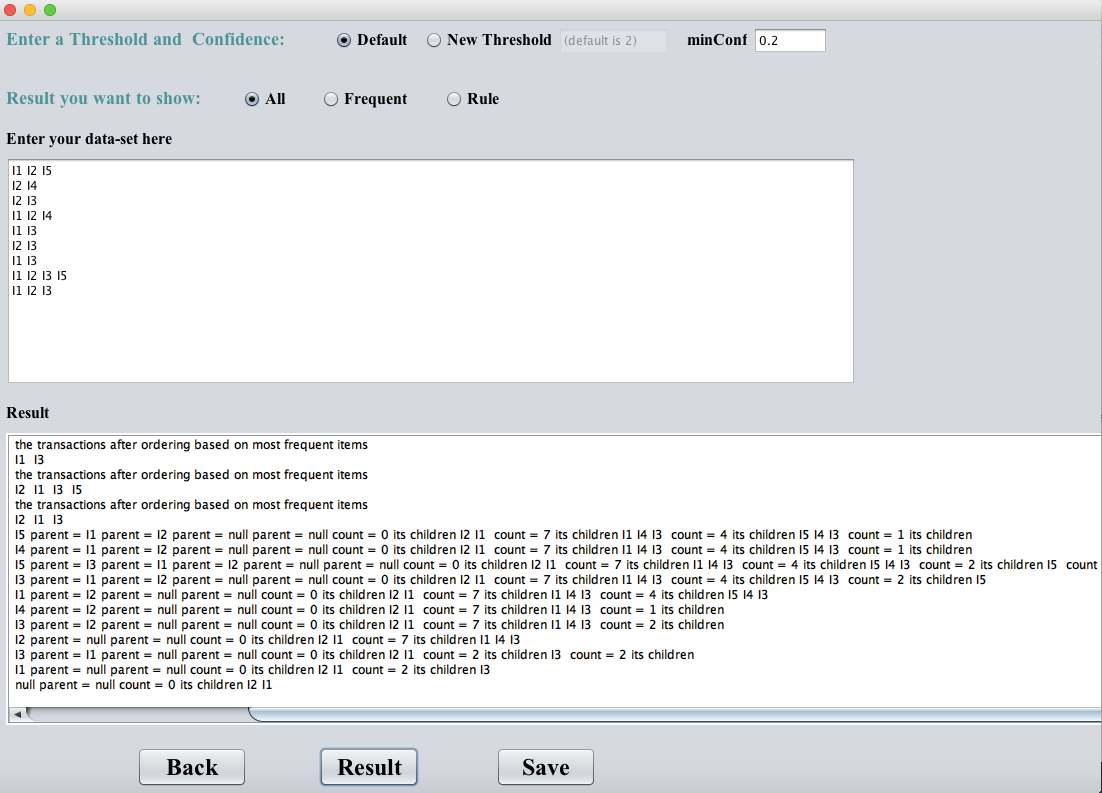


Figure 5.5: Result of all with default threshold and with confidence 0.2

With Threshold =4 and min confidence =0.4

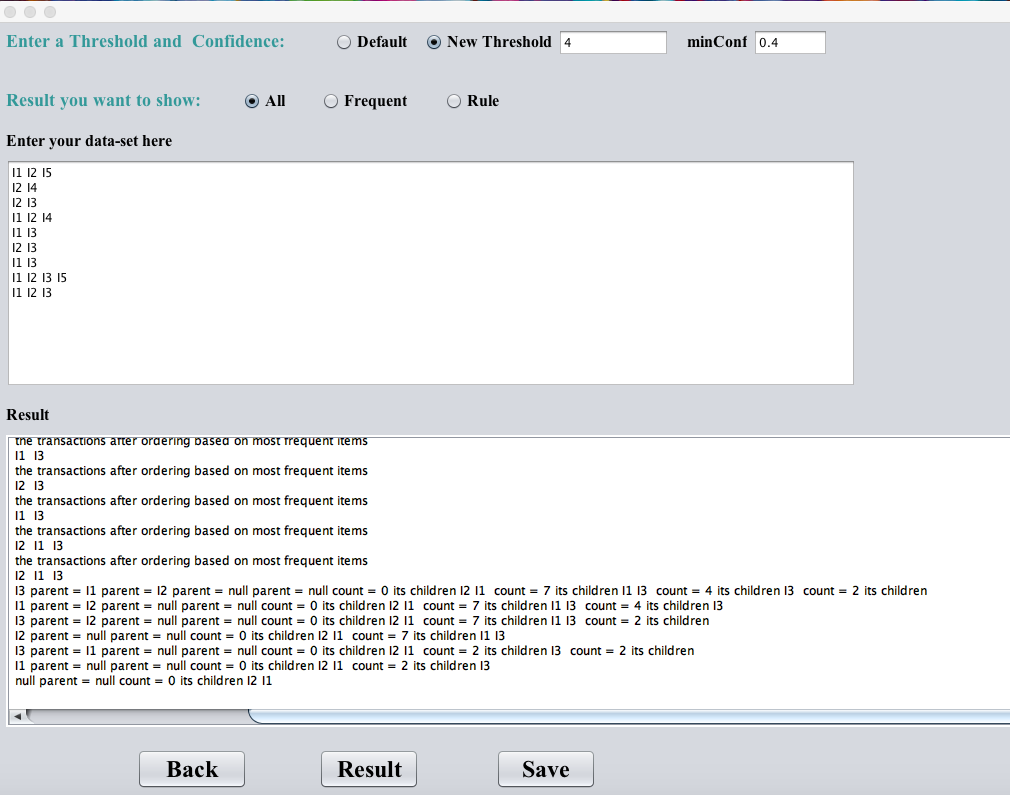


Figure 5.6: Result of all with threshold 4 and with confidence 0.4

## 5.2 Test the code with Anonymous Microsoft Web Data Set

### 5.2.1 The Frequent

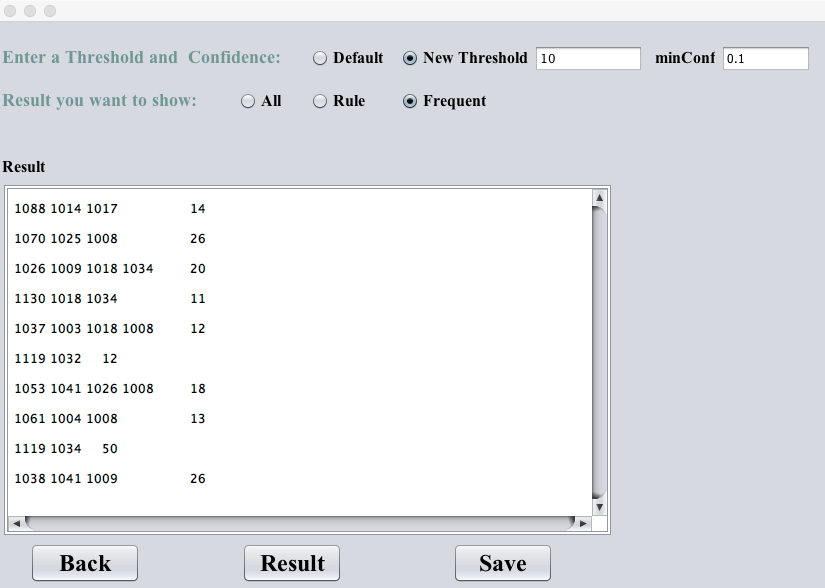
Applying our data set with Threshold = 10:

Figure 5.7: Result of frequent with threshold 10

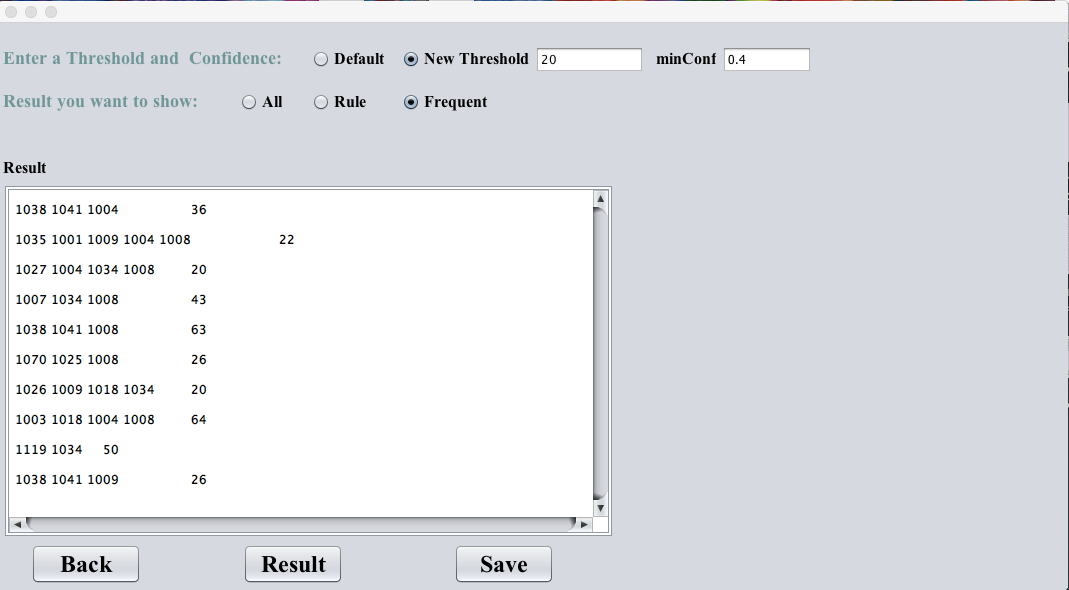
with Threshold = 20:

Figure 5.8: Result of frequent with threshold 20

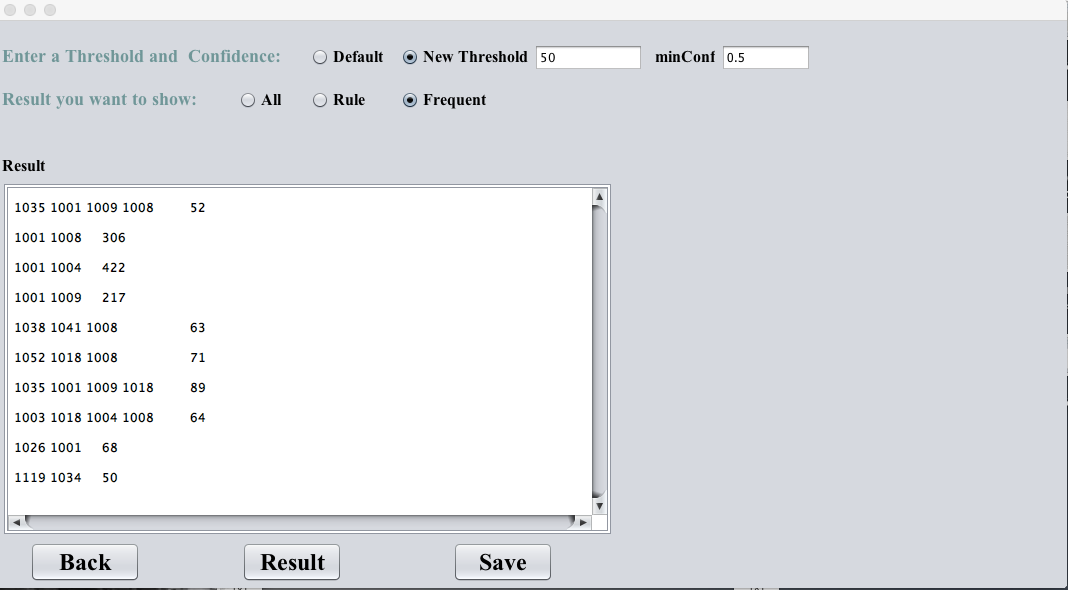
with Threshold = 50:

Figure 5.9: Result of frequent with threshold 50

### 5.2.2 The Rule

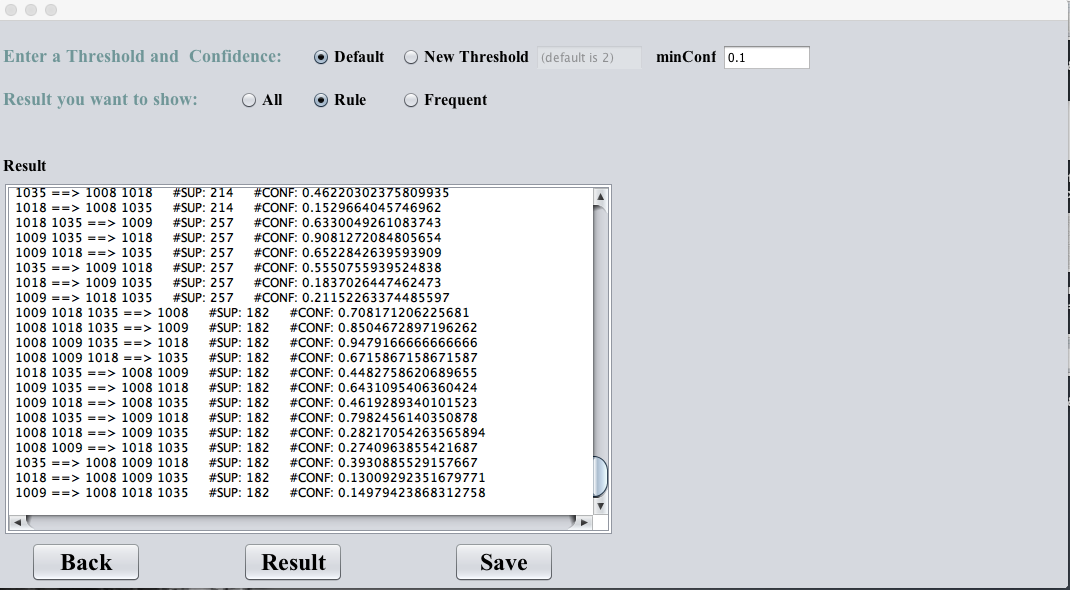
with Threshold =2 and with confines =0.1:

Figure 5.10: Result of rule with frequent default threshold 2 and with confidence 0.1

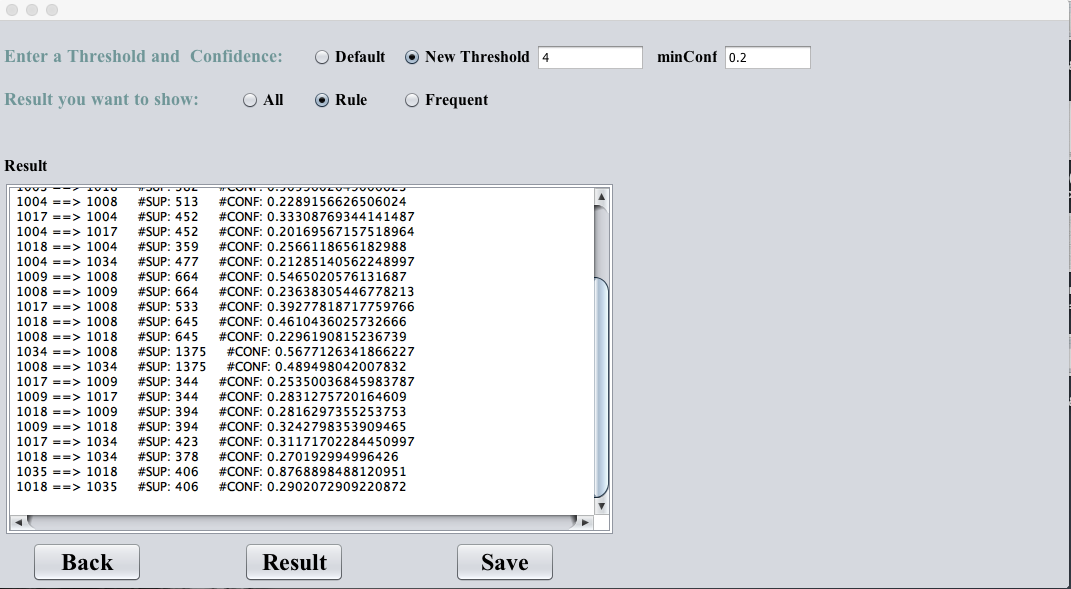
with Threshold = 4 and with confines =0.2:

Figure 5.11: Result of rule with default threshold 4 and with confidence 0.2

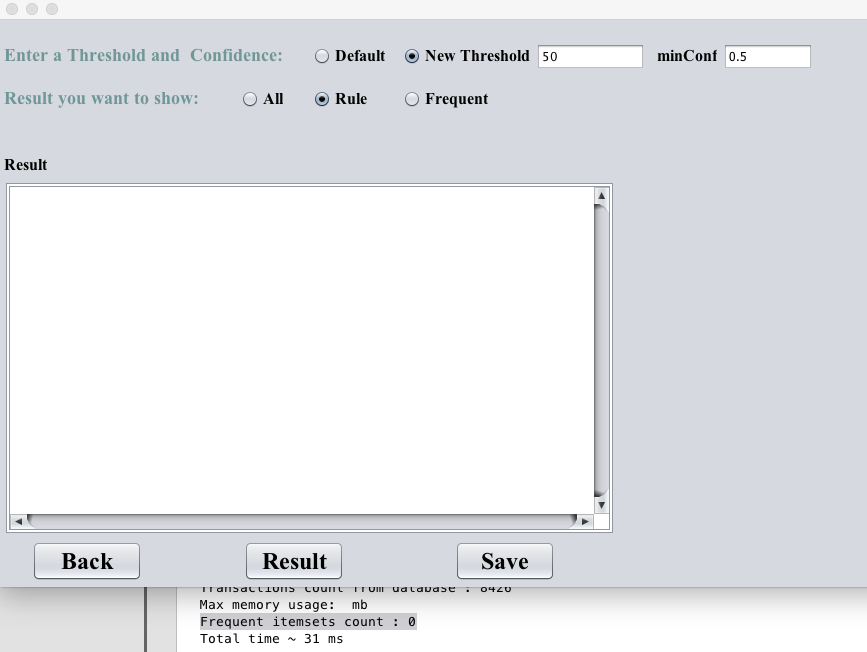
with Threshold (=50) and with confines =0.5. This gave us no rules with these numbers.

Figure 5.12: Result of rule with default threshold 50 and with confidence 0.5

### 5.2.3 All the Results

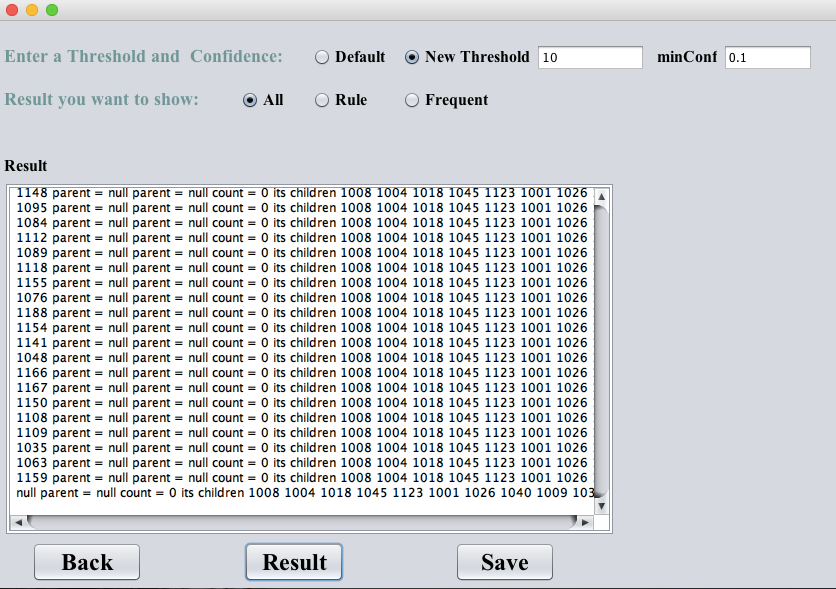
with new Threshold =10 and with confines =0.1:

Figure 5.13: Result of all with default threshold 10 and with confidence 0.1

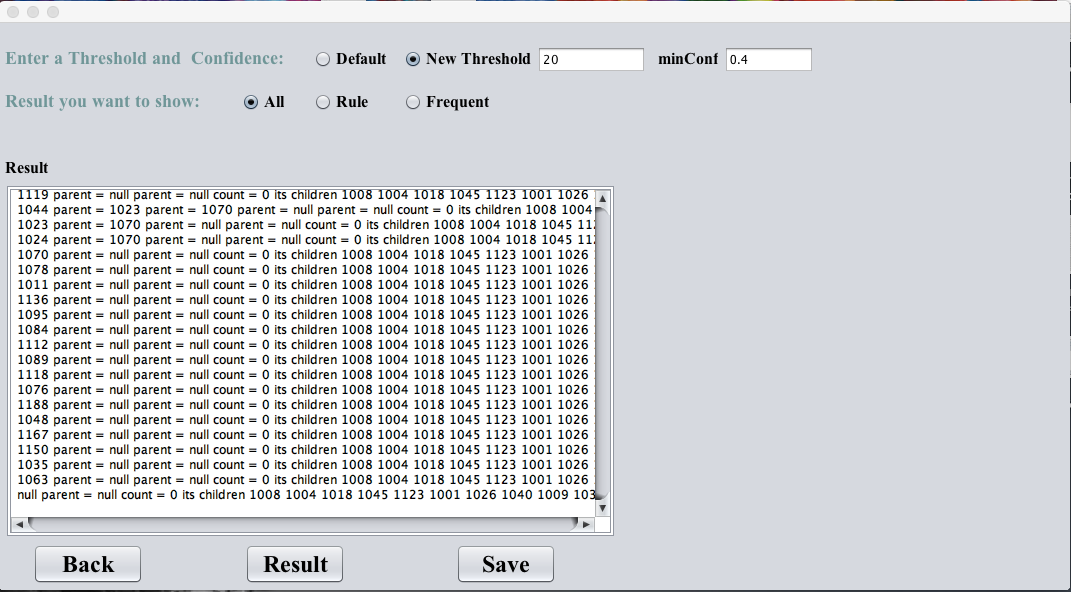
with new Threshold =20 and with confines =0.4:

Figure 5.14: Result of all with default threshold 20 and with confidence 0.4

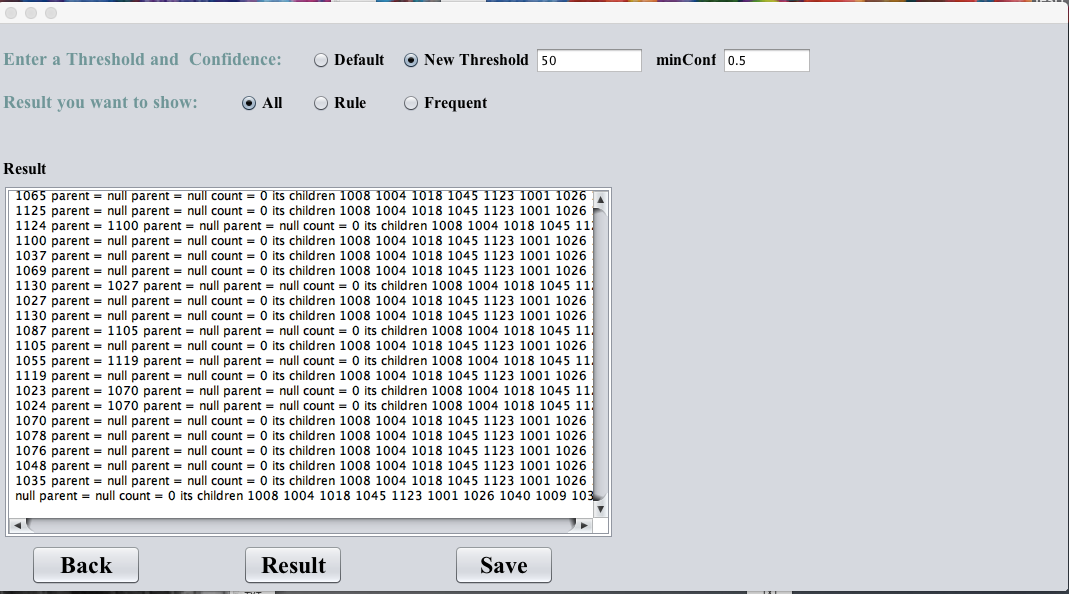
with new Threshold (= 50) and with confines =0.5:

Figure 5.15: Result of all with default threshold 50 and with confidence 0.5

## 5.3 The Analyses of the result

After applying the algorithm, we take the top 25 frequent and analyze it.

|  |  |  |
| --- | --- | --- |
| **The Frequent item** | **Number of Frequency** | **Description of the Dataset** |
| 1008 | 2809 | "Free Downloads","/msdownload" |
| 1034 | 2422 | "Internet Explorer","/ie" |
| 1004 | 2241 | "Microsoft.com Search","/search" |
| 1018 | 1399 | "isapi","/isapi" |
| 1034 1008 | 1375 | "Internet Explorer”,"/ie" then "Free Downloads","/msdownload" |
| 1017 | 1357 | "Products ","/products" |
| 1009 | 1215 | "Windows Family of OSs","/windows" |
| 1001 | 1154 | "regwiz","/regwiz" |
| 1026 | 837 | "Internet Site Construction for Developers","/sitebuilder" |
| 1003 | 755 | "Knowledge Base","/kb" |
| 1009 1008 | 664 | "Windows Family of OSs”,"/windows" then "Free Downloads","/msdownload" |
| 1018 1008 | 645 | “isapi","/isapi" then "Free Downloads","/msdownload" |
| 1017 1008 | 533 | "Products “,"/products" then "Free Downloads","/msdownload" |
| 1004 1008 | 513 | "Microsoft.com Search","/search" then "Free Downloads”,"/msdownload" |
| 1025 | 509 | "Web Site Builder's Gallery","/gallery" |
| 1001 1018 | 503 | “regwiz","/regwiz" then "isapi","/isapi" |
| 1004 1034 | 477 | "Microsoft.com Search”,"/search" then "Internet Explorer","/ie" |
| 1035 | 463 | "Windows95 Support","/windowssupport" |
| 1017 1004 | 452 | "Products “,”/products" then "Microsoft.com Search","/search" |
| 1003 1001 | 451 | "Knowledge Base”,"/kb" then "regwiz","/regwiz" |
| 1017 1034 | 423 | "Products ","/products" then "Internet Explorer","/ie" |
| 1001 1004 | 422 | “regwiz”,"/regwiz" then "Microsoft.com Search","/search" |
| 1041 | 416 | "Developer Workshop","/workshop" |
| 1040 | 409 | "MS Office Info","/office" |
| 1035 1018 | 406 | "Windows95 Support”,"/windowssupport" then "isapi","/isapi" |

Table 5.1: Description of a small part of data set

Conclusion

In this project, our objective is to help our Information Technology Deanship to analyze the University site from the users’ aspect. To achieve this goal, we want to apply FP-Growth algorithm as an association rule algorithm on the log files of the server in order to find the most frequent using web pages. Analyzing these patterns will give directions to the administration in the improvement of the site.

In the first phase of our project, we have studied some literature in the field of data mining in general and in association rules in particular.

More focus is given to the FP-Growth algorithms and different improvements done on it to avoid some problems found during its real implementation.

At the end, we have given a vision of our system by designing its class diagram and setting the main outputs for it.

References

1. Han, J. and M. Kamber, “Data Mining Concepts and Techniques”, Second Edition, MORGAN KAUFMANN, 2006.

[2] Oracle Help Center, “Data Mining Concepts”,

<https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/process> .

[3] Tutorials Points, “Data Mining Applications”,

<https://www.tutorialspoint.com/data_mining/dm_applications_trends.htm>.

[4] Goopta, C. “Six of the Best Open Source Data Mining Tools”, 2014.

<http://thenewstack.io/six-of-the-best-open-source-data-mining-tools>

[5] Wikipedia, “Orange (software)”, 2016.

<https://en.wikipedia.org/wiki/Orange_(software)>

[6] Beriman, L, J. Friedman, R. Olshen, and C. Stone, “Classification and Regression Trees”, Wadsworth International Group, 1984.

[7] Chapple, M. Database Expert, 2014.

<http://databases.about.com/od/datamining/g/clustering.htm>

[8] Zhao Q. and S.S. Bhowmick, "Association Rule Mining", Nanyang Technological University, Singapore, 2003.

<https://www.lri.fr/~antoine/Courses/Master-ISI/Regle-association.pdf>

[9] Siyamand Ayubia, Maybin K. Muyebab, Ahmad Baraania, John Keanec, "An algorithm to mine general association rules from tabular data", ScienceDirect Information Sciences, Volume 179, Issue 20, 29 September 2009, Pages 3520–3539.

[10] Agrawal, R., T. Imieliński, A. Swami, "Mining association rules between sets of items in large databases", SIGMOD '93 Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data.

[11] T. Bharathi,P.Krishnakumari , “A Comparative Analysis on Efficiency of Contemporary Association Rule Mining Algorithm”, RVS Arts and Science College-INDIA, Jan. 22 – 23, 2016,

<http://ieeexplore.ieee.org.ezproxy.qu.edu.sa/stamp/stamp.jsp?arnumber=7586406&tag=1>.

[12] Borgelt, C. “An Implementation of the FP-growth Algorithm”, Otto-von-Guericke-University of Magdeburg Universit¨atsplatz 2, 39106 Magdeburg, Germany, <http://www.borgelt.net/papers/fpgrowth.pdf>

[13] [Sidhu, Shivam](http://search.proquest.com.ezproxy.qu.edu.sa/indexinglinkhandler/sng/au/Sidhu,+Shivam/$N?accountid=30906); [Meena, Upendra Kumar](http://search.proquest.com.ezproxy.qu.edu.sa/indexinglinkhandler/sng/au/Meena,+Upendra+Kumar/$N?accountid=30906); [Nawani, Aditya](http://search.proquest.com.ezproxy.qu.edu.sa/indexinglinkhandler/sng/au/Nawani,+Aditya/$N?accountid=30906); [Gupta, Himanshu](http://search.proquest.com.ezproxy.qu.edu.sa/indexinglinkhandler/sng/au/Gupta,+Himanshu/$N?accountid=30906); [Thakur, Narina](http://search.proquest.com.ezproxy.qu.edu.sa/indexinglinkhandler/sng/au/Thakur,+Narina/$N?accountid=30906), “FP Growth Algorithm Implementation”, Foundation of Computer Science, United States, 2014,

<http://search.proquest.com.ezproxy.qu.edu.sa/docview/1529134556/abstract/D059D5B651644612PQ/1?accountid=30906>

[14] J. P. Jiawei Han, and Yiwen Yi, "Mining Frequent Patterns without Candidate Generation," Simon Fraser University2000, Available: <http://www.cs.bme.hu/~marti/adatbanya/fpgrowth.pdf>.

[15] Q. Yong, L. Yong-Jie, and X. Qing-Song, "An improved algorithm of mining from FP-tree," vol. 3, pp. 1665-1670 vol.3: IEEE.

[16] J. M. Y. P. J. Huiyan, "Research and Application on Web Information Retrieval Based on Improved FP-Growth Algorithm*,* vol. 11, no. 5, pp. 1065-1068, 2006.

[17] Z. Wei, L. Hongzhi, and Z. Na, "Research on the FP Growth Algorithm about Association Rule Mining," vol. 1, pp. 315-318: IEEE.

[18] A. B. M. R. Islam and T.-S. Chung, "An Improved Frequent Pattern Tree Based Association Rule Mining Technique," pp. 1-8.

[19] Y. Zeng, S. Yin, J. Liu, and M. Zhang, "Research of Improved FP-Growth Algorithm in Association Rules Mining," *Scientific Programming,* vol. 2015, pp. 1-6, 2015.

[20] L. Deng and Y. Lou, "Improvement and Research of FP-Growth Algorithm Based on Distributed Spark," pp. 105-108: IEEE.

[21] Available: <http://www.cs.waikato.ac.nz/ml/weka/arff.html>

Appendix

|  |  |
| --- | --- |
| **Means \ concept** | **ID** |
| "regwiz","/regwiz" | 1000 |
| "Support Desktop","/support" | 1001 |
| "End User Produced View","/athome" | 1002 |
| "Knowledge Base","/kb" | 1003 |
| "Microsoft.com Search","/search" | 1004 |
| "Norway","/norge" | 1005 |
| "misc","/misc" | 1006 |
| "International IE content","/ie\_intl" | 1007 |
| "Free Downloads","/msdownload" | 1008 |
| "Windows Family of OSs","/windows" | 1009 |
| "Visual Basic","/vbasic" | 1010 |
| "MS Office Development","/officedev" | 1011 |
| "Outlook Development","/outlookdev" | 1012 |
| "Visual Basic Support","/vbasicsupport" | 1013 |
| "Office Free Stuff","/officefreestuff" | 1014 |
| "Excel","/msexcel" | 1015 |
| "MS Excel","/excel" | 1016 |
| "Products ","/products" | 1017 |
| "isapi","/isapi" | 1018 |
| "MS PowerPoint","/mspowerpoint" | 1019 |
| "Developer Network","/msdn" | 1020 |
| "Visual C","/visualc" | 1021 |
| "Typography Site","/truetype" | 1022 |
| "Spain","/spain" | 1023 |
| "Internet Information Server","/iis" | 1024 |
| "Web Site Builder's Gallery","/gallery" | 1025 |
| "Internet Site Construction for Developers","/sitebuilder" | 1026 |
| "Internet Development","/intdev" | 1027 |
| "OLE Development","/oledev" | 1028 |
| "Clip Gallery Live","/clipgallerylive" | 1029 |
| "Windows NT Server","/ntserver" | 1030 |
| "MS Office","/msoffice" | 1031 |
| "Games","/games" | 1032 |
| "MS Store Logo Merchandise","/logostore" | 1033 |
| "Internet Explorer","/ie" | 1034 |
| "Windows95 Support","/windowssupport" | 1035 |
| "Corporate Desktop Evaluation","/organizations" | 1036 |
| "Windows 95","/windows95" | 1037 |
| "SiteBuilder Network Membership","/sbnmember" | 1038 |
| "Internet Service Providers","/isp" | 1039 |
| "MS Office Info","/office" | 1040 |
| "Developer Workshop","/workshop" | 1041 |
| "Visual Studio","/vstudio" | 1042 |
| "Connecting Small Business","/smallbiz" | 1043 |
| "Developer Media Development","/mediadev" | 1044 |
| "NetMeeting","/netmeeting" | 1045 |
| "IE Support","/iesupport" | 1046 |
| "MS Publisher","/publisher" | 1048 |
| "Support Network Program Information","/supportnet" | 1049 |
| "Macintosh Office","/macoffice" | 1050 |
| "MS Schedule+ News","/scheduleplus" | 1051 |
| "MS Word News","/word" | 1052 |
| "Jakarta","/visualj" | 1053 |
| "Exchange","/exchange" | 1054 |
| "MSHome Kids Stuff","/kids" | 1055 |
| "sports","/sports" | 1056 |
| "MS PowerPoint News","/powerpoint" | 1057 |
| "SP Referral (ART)","/referral" | 1058 |
| "Sweden","/sverige" | 1059 |
| "MS Word","/msword" | 1060 |
| "promo","/promo" | 1061 |
| "MS Access News","/msaccess" | 1062 |
| "Intranet Strategy","/intranet" | 1063 |
| "MS Site Builder Workshop","/activeplatform" | 1064 |
| "Java Strategy and Info","/java" | 1065 |
| "Music Producer","/musicproducer" | 1066 |
| "FrontPage","/frontpage" | 1067 |
| "VBScript Development","/vbscript" | 1068 |
| "Windows CE","/windowsce" | 1069 |
| "ActiveX Technology Development","/activex" | 1070 |
| "N. American Automap","/automap" | 1071 |
| "Visual InterDev","/vinterdev" | 1072 |
| "Taiwan","/taiwan" | 1073 |
| "Windows NT Workstation","/ntworkstation" | 1074 |
| "Job Openings","/jobs" | 1075 |
| "NT Workstation Support","/ntwkssupport" | 1076 |
| "MS Office Support","/msofficesupport" | 1077 |
| "NT Server Support","/ntserversupport" | 1078 |
| "Australia","/australia" | 1079 |
| "Brazil","/brasil" | 1080 |
| "Access Development","/accessdev" | 1081 |
| "MS Access","/access" | 1082 |
| "MS Access Support","/msaccesssupport" | 1083 |
| "UK","/uk" | 1084 |
| "Exchange Support","/exchangesupport" | 1085 |
| "OEM","/oem" | 1086 |
| "MS Proxy Server","/proxy" | 1087 |
| "OutLook","/outlook" | 1088 |
| "Office Reference","/officereference" | 1089 |
| "Games Support","/gamessupport" | 1090 |
| "Windows Hardware Development","/hwdev" | 1091 |
| "Visual FoxPro","/vfoxpro" | 1092 |
| "VBA Development","/vba" | 1093 |
| "Microsoft Home","/mshome" | 1094 |
| "Product Catalog","/catalog" | 1095 |
| "Microsoft Press","/mspress" | 1096 |
| "Latin America Region","/latam" | 1097 |
| "For Developers Only","/devonly" | 1098 |
| "Executive Computing","/cio" | 1099 |
| "MS in Education","/education" | 1100 |
| "Microsoft OLE DB","/oledb" | 1101 |
| "Microsoft Home Essentials","/homeessentials" | 1102 |
| "MS Works","/works" | 1103 |
| "Hong Kong","/hk" | 1104 |
| "France","/france" | 1105 |
| "Czech Republic","/cze" | 1106 |
| "Slovakia","/slovakia" | 1107 |
| "MS TeamManager","/teammanager" | 1108 |
| "TechNet (World Wide Web Edition)","/technet" | 1109 |
| "Mastering Series","/mastering" | 1110 |
| "Visual Source Safe","/ssafe" | 1111 |
| "Canada","/canada" | 1112 |
| "Internet Security Framework","/security" | 1113 |
| "Service Advantage","/servad" | 1114 |
| "Hungary","/hun" | 1115 |
| "Switzerland","/switzerland" | 1116 |
| "Sidewinder","/sidewinder" | 1117 |
| "SQL Server","/sql" | 1118 |
| "Corporation Information","/corpinfo" | 1119 |
| "Switching from Competitive Products","/switch" | 1120 |
| "Microsoft Magazine","/magazine" | 1121 |
| "Microsoft User Group Program","/mindshare" | 1122 |
| "Germany","/germany" | 1123 |
| "Industry Marketing Information (Vertical)","/industry" | 1124 |
| "ImageComposer","/imagecomposer" | 1125 |
| "Media Manager","/mediamanager" | 1126 |
| "NetShow","/netshow" | 1127 |
| "MS Solutions Framework","/msf" | 1128 |
| "ActiveX Data Objects","/ado" | 1129 |
| "IT Technical Information","/syspro" | 1130 |
| "MS Money Information","/moneyzone" | 1131 |
| "MS Money Support","/msmoneysupport" | 1132 |
| "FrontPage Support","/frontpagesupport" | 1133 |
| "BackOffice","/backoffice" | 1134 |
| "MS Word Support","/mswordsupport" | 1135 |
| "WorldWide Offices - US Districts","/usa" | 1136 |
| "About Microsoft ","/mscorp" | 1137 |
| "Developer Magazine","/mind" | 1138 |
| "MS in K-12 Education","/k-12" | 1139 |
| "Netherlands (Holland)","/netherlands" | 1140 |
| "Europe","/europe" | 1141 |
| "South Africa","/southafrica" | 1142 |
| "Site Builder Workshop","/workshoop" | 1143 |
| "For Developers Only News","/devnews" | 1144 |
| "Visual Fox Pro Support","/vfoxprosupport" | 1145 |
| "Microsoft Solution Providers","/msp" | 1146 |
| "Microsoft Financial Forum","/msft" | 1147 |
| "Channel Resources","/channel\_resources" | 1148 |
| "Advanced Data Connector","/adc" | 1149 |
| "Internet Information Server News","/infoserv" | 1150 |
| "MS PowerPoint Support","/mspowerpointsupport" | 1151 |
| "Russia","/rus" | 1152 |
| "Venezuela","/venezuela" | 1153 |
| "MS Project","/project" | 1154 |
| "Sidewalk","/sidewalk" | 1155 |
| "Powered by BackOffice","/powered" | 1156 |
| "Windows 32 bit developer","/win32dev" | 1157 |
| "Interactive Media Technologies","/imedia" | 1158 |
| "Transaction Server","/transaction" | 1159 |
| "Visual C Support","/visualcsupport" | 1160 |
| "Works Support","/workssupport" | 1161 |
| "IIS Support","/infoservsupport" | 1162 |
| "Open Type","/opentype" | 1163 |
| "Systems Management Server","/smsmgmt" | 1164 |
| "Poland","/poland" | 1165 |
| "Mexico","/mexico" | 1166 |
| "Windows Hardware Testing","/hwtest" | 1167 |
| "Sales Information (infobase)","/salesinfo" | 1168 |
| "MS Project","/msproject" | 1169 |
| "Microsoft Mail","/mail" | 1170 |
| "MS Merchant","/merchant" | 1171 |
| "Belgium","/belgium" | 1172 |
| "Microsoft OnLine Institute","/moli" | 1173 |
| "New Zealand","/nz" | 1174 |
| "MS Project Support","/msprojectsupport" | 1175 |
| "Java Script Development","/jscript" | 1176 |
| "Master Marketing Calendar","/events" | 1177 |
| "msdownload.","/msdownload." | 1178 |
| "Colombia","/colombia" | 1179 |
| "Slovenija","/slovenija" | 1180 |
| "Kids Support","/kidssupport" | 1181 |
| "Fortran","/fortran" | 1182 |
| "Italy","/italy" | 1183 |
| "MS Excel Support","/msexcelsupport" | 1184 |
| "SNA Server","/sna" | 1185 |
| "Job Listings for Pre-Grads","/college" | 1186 |
| "ODBC Development","/odbc" | 1187 |
| "Korea","/korea" | 1188 |
| "Internet News","/internet" | 1189 |
| "Repository","/repository" | 1190 |
| "Management","/management" | 1191 |
| "Visual J++ Support","/visualjsupport" | 1192 |
| "Office Developer Support","/offdevsupport" | 1193 |
| "China","/china" | 1194 |
| "Portugal","/portugal" | 1195 |
| "ie40","/ie40" | 1196 |
| "SQL Support","/sqlsupport" | 1197 |
| "Picture It","/pictureit" | 1198 |
| "feedback","/feedback" | 1199 |
| "Benelux Region","/benelux" | 1200 |
| "MS Hardware","/hardware" | 1201 |
| "Advanced Technology","/advtech" | 1202 |
| "Denmark","/danmark" | 1203 |
| "MS Schedule+","/msscheduleplus" | 1204 |
| "Hardware Supprt","/hardwaresupport" | 1205 |
| "Volume Purchasing Options","/select" | 1206 |
| "Internet Control Pack ","/icp" | 1207 |
| "Israel","/israel" | 1208 |
| "Turkey","/turkey" | 1209 |
| "SNA Support","/snasupport" | 1210 |
| "SMSMGT Support","/smsmgmtsupport" | 1211 |
| "World Wide Offices","/worldwide" | 1212 |
| "Corporate Customers","/corporate\_solutions" | 1213 |
| "MS Financial Services","/finserv" | 1214 |
| "For Developers Only Info","/developer" | 1215 |
| "Virtual Reality Markup Language","/vrml" | 1216 |
| "Ireland","/ireland" | 1217 |
| "MS Publisher Support","/publishersupport" | 1218 |
| "Corporate Advertising Content","/ads" | 1219 |
| "Mac Office Support","/macofficesupport" | 1220 |
| "Microsoft TV Program Information","/mstv" | 1221 |
| "MS Office News","/msofc" | 1222 |
| "Finland","/finland" | 1223 |
| "Authorized Technical Education Center Program","/atec" | 1224 |
| "Anti-Piracy Information","/piracy" | 1225 |
| "MS Schedule+ Support","/msschedplussupport" | 1226 |
| "Argentina","/argentina" | 1227 |
| "Visual Test","/vtest" | 1228 |
| "Uruguay","/uruguay" | 1229 |
| "Mail Support","/mailsupport" | 1230 |
| "Windows NT Developer Support","/win32devsupport" | 1231 |
| "SiteBuilder Network Specs & Standards","/standards" | 1232 |
| "vbscripts","/vbscripts" | 1233 |
| "Office Free Stuff News","/off97cat" | 1234 |
| "MS Training Evaluation","/onlineeval" | 1235 |
| "Developing for Global Markets","/globaldev" | 1236 |
| "Developer Days","/devdays" | 1237 |
| "Excel Development","/exceldev" | 1238 |
| "Microsoft Consulting","/msconsult" | 1239 |
| "Thailand","/thailand" | 1240 |
| "India","/india" | 1241 |
| "MS Garden","/msgarden" | 1242 |
| "MS Usability Group","/usability" | 1243 |
| "Developer Newswire","/devwire" | 1244 |
| "Open Financial Connectivity","/ofc" | 1245 |
| "Developer Media Games","/gamesdev" | 1246 |
| "Wine Guide","/wineguide" | 1247 |
| "Softimage ","/softimage" | 1248 |
| "Fortran Support","/fortransupport" | 1249 |
| "Middle East","/middleeast" | 1250 |
| "Reference Support","/referencesupport" | 1251 |
| "Community Affairs","/giving" | 1252 |
| "MS Word Development","/worddev" | 1253 |
| "ie3","/ie3" | 1254 |
| "Message Queue Server","/msmq" | 1255 |
| "Solutions in Action","/sia" | 1256 |
| "Professional Developers Series","/devvideos" | 1257 |
| "Peru","/peru" | 1258 |
| "controls","/controls" | 1259 |
| "Exchange Trial","/trial" | 1260 |
| "MS's Complete Do It Yourself Guide","/diyguide" | 1261 |
| "Chile","/chile" | 1262 |
| "Educational Services & Programs","/services" | 1263 |
| "MS Partner Web","/se\_partners" | 1264 |
| "Source Safe Support","/ssafesupport" | 1265 |
| "Licenses and Piracy","/licenses" | 1266 |
| "Caribbean","/caribbean" | 1267 |
| "javascript","/javascript" | 1268 |
| "Customer Guides","/business" | 1269 |
| "developr","/developr" | 1270 |
| "mdsn","/mdsn" | 1271 |
| "softlib","/softlib" | 1272 |
| "mdn","/mdn" | 1273 |
| "Professional Developer Conference","/pdc" | 1274 |
| "security.","/security." | 1275 |
| "Visual Test Support","/vtestsupport" | 1276 |
| "NetShow for PowerPoint","/stream" | 1277 |
| "MS in Higer Education","/hed" | 1278 |
| "Multimedia Golf","/msgolf" | 1279 |
| "MS Interactive Music Control","/music" | 1280 |
| "IntelliMouse","/intellimouse" | 1281 |
| "home","/home" | 1282 |
| "Cinemainia","/cinemania" | 1283 |
| "partner","/partner" | 1284 |
| "International AutoRoute","/autoroute" | 1287 |
| "library","/library" | 1288 |
| "Master Chef Product Information","/masterchef" | 1289 |
| "Activate the Internet Conference","/devmovies" | 1290 |
| "news","/news" | 1291 |
| "MS North Africa","/northafrica" | 1292 |
| "Encarta","/encarta" | 1293 |
| "Bookshelf","/bookshelf" | 1294 |
| "Training","/train\_cert" | 1295 |
| "Central America","/centroam" | 1297 |