

Ain Shams University

Faculty of Engineering

Computer and Systems Engineering Department

CSE 412: Selected Topics in Computer Engineering

4th Year CSE – 2nd Semester 2019/2020

TERM PROJECT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
   
   
Team Names:

* Reham Essam El-Tagoury
* Sara Abdelkhalek Mohamed
* Shrouk Ragheb Abd-Elazeem
* Abdullah Galal Mohamed
* Abdulrahman Hatem
* Mariam Khaled

Abstract:

In this project we apply Apriori algorithm to find all the possible association rules for user-defined values of support and confidence. Additionally, we compute the lift and leverage for each rule to prioritize the rules in the given dataset. The dataset itself represents the customer data for an insurance company; it has 12 attributes with 5822 records.

The dataset and the description of its attributes are available at <http://www.liacs.nl/~putten/library/cc2000/>.   //drive link \*our Data only\*.

Table of Contents:

|  |  |
| --- | --- |
| Content | Page Number |
| Purpose |  |
| List of definitions |  |
| Overview |  |
| Beneficiaries |  |
| Project Aims and Objectives |  |
| Detailed Project Description |  |
| Project Phases |  |
| System Architecture |  |
| Development Environment |  |
| Testing Cases and Results |  |
| Conclusion |  |
| Use Earned Value method to assess the progress in your project |  |

1. Introduction:

1.1 Purpose:

The purpose of the project aims to present how data mining is useful in the insurance company system, apply Apriori algorithm on a database containing a lot of transactions, mining frequent itemset and relevant association rules to determine relations among variables in large databases.

* 1. List of Definitions:

|  |  |
| --- | --- |
| Term | Definition |
| ATWP | Actual time of work performed. |
| ETWP | Estimated time of work performed. |
| ETWS | Estimated time of work scheduled. |
| TPI | Time performance index. |
| TV | Time variance. |
| SPI | Schedule performance index. |
| SV | Schedule variance. |
| System Architecture | Comma-separated values file format. |
| CSV | Comma-separated values file format. |
| Itemset | A collection of one or more items. |
| Lift | Measures how many times more often X and Y occur together than expected if they were statistically independent. |
| Leverage | It is a similar notion to lift but instead of a ratio it is the difference. |
|  |  |
| Support | Fraction of transactions that contain an itemset. |
| Confidence | The % of transactions that contain X, which also contain Y. |

* 1. Overview:

This document presents the main informations and details about big data project which his purpose mentioned above, the document presents that starting from the purpose and description of the project, followed by some terminologies that we used, beneficiary areas of this program, project aims and the five objectives of it, detailed description, our three phases that we worked through it and system architecture diagram, then the development environment and six test cases to ensure work, finished by conclusion of this work and earned value analysis on the time that we spent in this project.

# 2. Beneficiaries:

This program is useful for the insurance company. It will help employees to sell more insurance policies by one button click. The employee will enter minimum confidence, and minimum support and the program will produce all association rules. This will help employees to predict who would be interested in buying insurance policy and give and explain why. This procedure will save effort and a lot of time.  
This program can also be useful in other fields by applying simple modification. Let’s look at some areas where this program will help quite a lot:

Market Basket Analysis:

This is the most typical example of association mining. Data is collected using barcode scanners in most supermarkets. This database, known as the “market basket” database, consists of a large number of records on past transactions. A single record lists all the items bought by a customer in one sale. Knowing which groups are inclined towards which set of items gives these shops the freedom to adjust the store layout and the store catalog to place the optimally concerning one another.

Medical Diagnosis:

Association rules in medical diagnosis can be useful for assisting physicians for curing patients. Diagnosis is not an easy process and has a scope of errors which may result in unreliable end-results. Using relational association rule mining, we can identify the probability of the occurrence of illness concerning various factors and symptoms. Further, using learning techniques, this interface can be extended by adding new symptoms and defining relationships between the new signs and the corresponding diseases.

Census Data:

Every government has tones of census data. This data can be used to plan efficient public services (education, health, transport) as well as help public businesses (for setting up new factories, shopping malls, and even marketing particular products). This application of association rule mining and data mining has immense potential in supporting sound public policy and bringing forth an efficient functioning of a democratic society.

3. Project Aims and Objectives:

The aim of using Apriori is to present how data mining is useful in the insurance company system by finding frequent itemsets and association between different itemsets, that is, association rule.

**Objectives:**

Our objective is to find:

1-Support

Support is a so-called frequency constraint. Its main feature is that it possesses the property of down-ward closure which means that all sub sets of a frequent set (support > min. support threshold) are also frequent. This property is used to prune the search space (usually a tree of item sets with increasing size) in level-wise algorithms (e.g., the APRIORI algorithm).

sup (X -> Y) = sup (Y -> X) = P (X and Y)

2-Confidence

Confidence is not down-ward closed and was developed together with support (the so-called support-confidence framework). While support is used to prune the search space and only leave potentially interesting rules, confidence is used in a second step to filter rules that exceed a min. confidence threshold.

conf (X -> Y) = P (Y | X) = P (X and Y)/P(X) = sup (X -> Y)/sup(X)

3-Lift

Lift measures how many times more often X and Y occur together than expected if they were statistically independent.

Lift (X -> Y) = lift(Y -> X) = P(X and Y)/(P(X)P(Y)) = conf (X -> Y)/sup(Y) = conf (Y -> X)/sup(X)

4-Leverage

Leverage measures the difference of X and Y appearing together in the data set and what would be expected if X and Y where statistically dependent. The rational in a sales setting is to find out how many more units (items X and Y together) are sold than expected from the independent sells.

leverage (X -> Y) = P (X and Y) - (P (X)P (Y))

5- Association rules:

The association rule discovery is the derivation of if-then-rules based on the itemsets. It is specifically designed for mining over transactions in databases.

4. Detailed Project Description:

In this project we apply Apriori algorithm to find all the association rules in the given dataset. The dataset itself represents the customer data for an insurance company (TIC); it has 86 attributes -from which we have chosen 12 attributes (1-2-3-4-5-6-7-8-9-47-48-49) with 5822 records. In the source code, the attributes are numbered (1-2-3-4-5-6-7-8-9-10-11-12) respectively .Our task is to find all the possible association rules for user-defined values of support and confidence. Additionally, we compute the lift and leverage for each rule to prioritize the rules.

This data set used in the CoIL 2000 Challenge contains information on customers of an insurance company. The data consists of 86 variables and includes product usage data and socio-demographic data derived from zip area codes. The data was collected to answer the following question: Can you predict who would be interested in buying a caravan insurance policy and give an explanation why? The 12 attributes we chose to perform our algorithm on are: "PPERSAUT Contribution car policies", "PBESAUT Contribution delivery van policies", "PMOTSCO Contribution motorcycle/scooter policies", "MOSTYPE Customer Subtype", "MAANTHUI Number of houses", "MGEMOMV Avg size household", "MGEMLEEF Avg age", "MOSHOOFD Customer main type", "MGODRK Roman catholic", "MGODPR Protestant", "MGODOV Other religion" and "MGODGE No religion".

Our program basically does the following: read the specified attributes from the dataset csv file, apply Apriori algorithm based on the user-defined values of minimum support and confidence and print the output for the user which is all possible rules with also the lift and leverage values calculated for each rule. The implementation of Apriori algorithm consists of the following: Find the unique values of each column and consider it an independent item, find the frequent items and eliminating the items whose support is less than the specified minimum support and extending them to larger item sets as long as those item sets appear sufficiently often in the database, generate all possible combinations of the frequent item sets to generate all possible rules and finally eliminate the rules whose confidence is less than the specified minimum confidence.

5. Project Phases:

Learning Phase

In this phase we learned all about the Apriori algorithm and the detailed steps to solve a problem with it, collected all the rules that we need like lift and leverage rules and how to calculate the confidence of a rule. We solved a problem as a practice for us to make sure that we completely understand all the steps before starting to code.

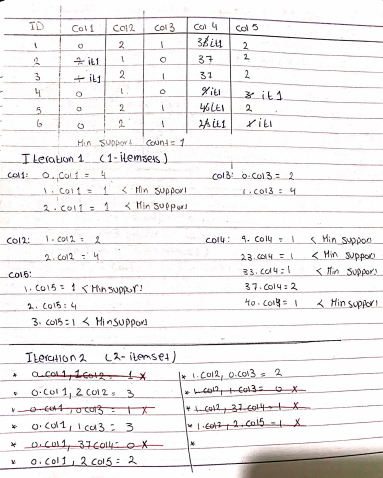


Fig (1): The example that we solved as practice

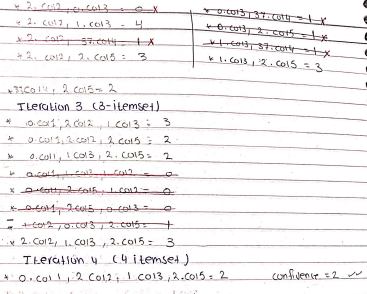


Fig (2): The example that we solved as practice

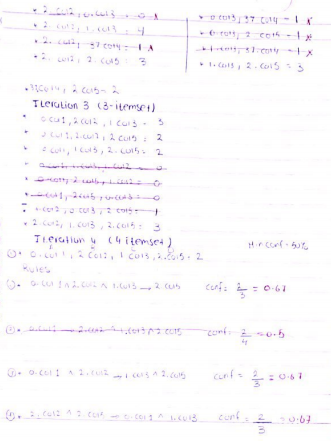


Fig (3): The example that we solved as practice

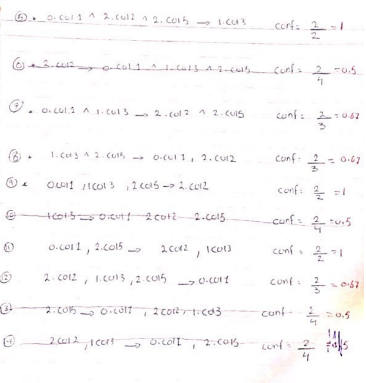


Fig (4): The example that we solved as practice

Implementation phase

In this phase, we developed a python script that takes as input: The dataset, minimum support and minimum confidence and generate as output: All possible association rules and the values of lift and leverage for each rule.

Final phase

In this phase, we finalized our work by documenting our project in this report to demonstrate our goals and the steps we did to achieve it. Furthermore, we prepared a video containing a brief presentation of our project and a demo of our program running on some test cases and illustrating the output.

# 6. System Architecture:

# A screenshot of a cell phone Description automatically generated

# 7. Development Environment:

## -Windows 10 Operating System.

## -Sublime Text:

Written by a Google engineer with a dream for a better text editor, Sublime Text is an extremely popular code editor. Supported on all platforms, Sublime Text has built-in support for Python code editing and a rich set of extensions (called packages) that extend the syntax and editing features.

Installing additional Python packages can be tricky: all Sublime Text packages are written in Python itself, and installing community packages often requires you to execute Python scripts directly in Sublime Text.

**Pros:** Sublime Text has a great following in the community. As a code editor, alone, Sublime Text is fast, small, and well supported.

**Cons:** Sublime Text is not free, although you can use the evaluation version for an indefinite period of time. Installing extensions can be tricky, and there’s no direct support for executing or debugging code from within the editor.

8. Testing Cases and Results:

We will apply 6 test cases.

Test case 1:

* Input:
* Min-support = 0.1
* Min-confidence = 0.9
* Output:
* 13 association rules
* Lift and leverage for each rule



Fig (5): Output of test case1.

A picture containing sitting, table, large, group

Description automatically generated

Fig (6): Output of test case1.

Test Case 2:

* Input:
* Min-support = 0.2
* Min-confidence = 0.7
* Output:
* 6 association rules
* A picture containing sitting, light, table, lit

  Description automatically generatedLift and leverage for each rule

Fig (7): Output of test case 2.

Test Case 3:

* Input :
* Min-support = 0.3
* Min-confidence = 0.7
* Output:
* 7 association rules
* Lift and leverage for each rule

A picture containing sitting, table, light

Description automatically generated

Fig (8): Output of test case 3.

Test Case 4:

* Input:
* Min-support = 0.5
* Min-confidence = 0.5
* Output:
* 2 association rules
* Lift and leverage for each rule

Fig.(9): Output of test case 4.

Test Case 5:

* Input:
* Min-support = 0.5
* Min-confidence = 0.7
* Output:
* 1 association rule
* Lift and leverage for each rule

Fig (10): Output of test case 5.

Test case 6:

* Input:
* Min-support = 0.9
* Min-confidence = 0.3
* Output:
* 0 association rule



Fig (11): Output of test case 6.

9. Conclusion:

The conclusion to this work is that Apriori algorithm is applied on the transactional database. By using measures of apriori algorithm, frequent itemsets can be generated from the database. Apriori algorithm is associated with certain limitations of large database scans. Advantage of apriori is its easy implementation. Association rule mining efficiency can be improved by using attributes which will give the valuable information to the employee as well as the business. Association rule mining has a wide range of applicability in many areas.

10. Use Earned Value method to assess the progress in your project:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activity | Predecessor | Duration (Days) | Hours / Day | Total time (h) |
| Learning Phase | - | 2 | 3 | 6 |
| Implementation Phase | Learning Phase | 8 | 2 | 16 |
| Final Phase | Implementation Phase | 3 | 2 | 6 |

|  |  |  |
| --- | --- | --- |
| Field report at end of day 10 | | |
| Activity | Actual % complete | Actual Time |
| Learning Phase | 100 | 6 |
| Implementation Phase | 75 | 15 |
| Final Phase | 0 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | ATWP | ETWP | ETWS | TPI | TV | SPI | SV |
| Learning Phase | 6 | 6 | 6 | - | - | - | - |
| Implementation Phase | 15 | 12 | 16 | - | - | - | - |
| Final Phase | 0 | 0 | 0 | - | - | - | - |
| Total | 21 | 18 | 22 | 0.86 | - 3 | 0.81 | - 4 |

* Every work time unit is an average of total team members work time in this unit
* We planned to make EVA on day 10
* We have 2 days delayed
* TPI = ETWP / ATWP
* TV = ETWP - ATWP
* SPI = ETWP / ETWS
* SV = ETWP - ETWS