Ain Shams University
Faculty of Engineering
Computer and Systems Department
CSE 412: Selected Topics in Computer Engineering 4th Year CSE – 2nd Semester
2019/2020

Apriori Algorithm to find all the association rules in the given dataset

Group: 30

Students:

Name	ID	Section
محمد رضا جوده عوض	**1501207**	**3**
احمد اشرف محمود محمد	**1500049**	**1**
احمد شريف محمود جمال الدين	**1500104**	**1**
سهى علاء	**12E0075**	**2**
محمود عبد النبي عبدالحفيظ	**1501357**	**3**
محمد خالد عواد متولي	**1501199**	**3**

Abstract:

This report describes the implementation of Apriori algorithm to find all the association rules in the given dataset. The dataset itself represents the customer data for an insurance company; it has 12 attributes with 5822 records. This implementation aims to find all the possible association rules for user-defined values of support and confidence, in addition to computing the lift and leverage for each rule. The output of the algorithm is generated into a text file containing all valid association rules along with their support, confidence, lift and leverage. The algorithm was implemented from scratch using python programming language and sublime pycharm. This implementation was done by a group of 6 senior students at the faculty of engineering, Ain shams university, department of computer and systems engineering as a term project for a big data course. This report is considered as one of the deliverables for the project alongside the source code, a demo representation of how the code implements the algorithm and presentation slides.

Table of Contents:

1. Introduction

1.1. Purpose

The purpose of this report is to describe the term project of the Apriori algorithm implementation, clarify the project aims, objectives and phases.

1.2. List of definitions

• **Apriori Algorithm:** It is an algorithm for frequent item set mining and association rule learning over relational databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

• **Support:** It is an indication of how frequently the items appear in the data. It is measured by the proportion of transactions in which an itemset appears.

• **Confidence:** It explains how likely Y is purchased when X is purchased. It defines association between two items. This is measured by the proportion of transactions with item X, in which item Y also appears.

• **Lift:** This says how likely item Y is purchased when item X is purchased, while controlling for how popular item Y is. A lift value greater than 1 means that item Y is likely to be bought if item X is bought, while a value less than 1 means that item Y is unlikely to be bought if item X is bought.

• **Leverage:** It computes the difference between the observed frequency of X and Y appearing together and the frequency that would be expected 5

if X and Y were independent. A leverage value of 0 indicates independence.

Association rules: if-then statements that help to show the probability of relationships between data items within large data sets in various types of databases.

• **Cross tabular data format:** It is a method to quantitatively analyze the relationship between multiple variables. Cross tabulation groups variables to understand the correlation between different variables. It also shows how correlations change from one variable grouping to another. It is usually used in statistical analysis to find patterns, trends, and probabilities within raw data.

• **Actual time:** is the total time taken for the actual work completed.

• **Planned value:** the approved duration to get the task completed.

• **Earned value:** the work completed to date. It shows you the value that the project has produced if it were terminated today.

• **SPI:** Schedule performance index. It shows how you are progressing compared to the planned project schedule. It is a measure of schedule efficiency, expressed as the ratio of earned value to planned value.

• **SV:** Schedule variance. It helps you complete on time. You are ahead of schedule if the Schedule Variance is positive, behind schedule if the Schedule Variance is negative, on schedule if the Schedule Variance is zero.

1.3. Overview

The report starts by stating the project beneficiaries, then describing the project aims and objectives. The report then gives a detailed description for the project, its architecture and development environment, also providing some test cases and their result. Finally, it provides the project conclusion and uses the earned value method to assess the progress in the project.

2. Beneficiaries

The Apriori algorithm is used for data mining and generation for association rules which can be very helpful to various beneficiaries like:

2.1. Insurance Companies

• Predict which customers are potentially interested in an insurance policy.
• Describe the actual or potential customers; and possibly explain why these customers buy a policy.

2.2. Retail Shops

• Determine which retail items are purchased together
• Describe the potential customers for certain retail items

2.3. Web Services

• Determine a set of links clicked on by one user in a single session
• Filtering the web advertisements that appears to the user according to his /her interests

3. Project Aims and Objectives

3.1. Project Aims

This project aims to calculate the Apriori algorithm and generate all valid association rules to predict which customers are potentially interested in an insurance policy and to

describe the actual or potential customers; and possibly explain why these customers buy a policy.

3.2. Project Objectives

• Calculate Apriori algorithm to generate valid association rules based on user-defined minimum support and minimum confidence
• Calculate and output the support for each valid association rule
• Calculate and output the confidence for each valid association rule
• Calculate and output the lift for each valid association rule
• Calculate and output the leverage for each valid association rule

4. Detailed Project Description

The project was developed by using Python with the help of some basic libraries. It is important to mention that NO ready libraries to calculate either the support, confidence, lift, leverage, or association rules were used, as requested in the project requirements.

The project algorithm is executed by using 6 functions (which will be discussed later in the system architecture):
• Assigning_Values
• Slicing_rows
• Retrieve_L1_Support
• Retrieve_L2_Support
• Retrieve_L3_Support
• Retrieve_L4_Support
• Retrieve_L5_Support
• Retrieve_L6_Support
• Retrieve_L7_Support
• Retrieve_L8_Support
• Retrieve_L9_Support
• Retrieve_L10_Support

• Retrieve_L11_Support
• Retrieve_L12_Support
• Calculate_All

The program firstly prompts the user to enter the minimum support and keeps prompting him/her until they enter a value. Then it prompts him/her to enter the minimum confidence and keeps prompting until they enter a value.

The program then calls “Slicing_rows” function to extract the specified attributes on which the association rules will be generated from the while data found in the “Final.csv” file. These 12 attributes for our team are from index 37 till index 48, and calculate all support of all levels for the data and compare them to the user-defined minimum support, If the value is above minimum support then the program calls ”Assigning_Values” function to assign this value.

Theses item sets are then passed along with the minimum support to “Calculate_All” function to calculate all the possible rules for every and each item set and compare their confidence with the user-defined minimum confidence. Then it calculates the lift and leverage for the valid rules. Finally, it returns the valid rules with their support, confidence, lift, and the index of the attribute in the right-hand side of the rule.

5. Project Phases

5.1. Data Extraction and Inputs Validation:

Working on extracting the 12 attributes from “Final_csv” and adding their names to them according to the data description file, also working on validating the user inputs.

5.2. Support Calculation:

Working on calculating the support for all available item sets in the data and comparing them to the user-defined minimum support and pruning the item sets that have support less than the minimum support

5.3. Confidence, Lift, and Leverage Calculations:

Working on calculating the confidence for all different rules that can be generated from each item set that has support more than or equal the minimum support, then compare them to the user-defined minimum confidence and pruning the rules that have confidence less than the minimum confidence. Then for each item set with valid rules calculate both the lift and the leverage.

5.4. Outputting the Rules:

Working on outputting all the previously calculated valid rules in an understandable format , also printing the support, confidence, lift and leverage for each rule.

6. System Architecture

This Apriori algorithm is implemented in python by implementing 8 main functions:

• Slicing_rows(A,B,C,D,E,F,G,H,I,J,K,M,input_length)

Which takes the data rows of “Final_csv” and the minimum support which generates the possible combinations of all data rows ,calculates their support ,and raises the Flag of the attribute when its value above minimum support

• Assigning_Values(Row_Slice, [Flag_A, Flag_B ,…..Flag_M],input_length,counter)

which takes the Flags of attribute that handled by the pervious function and if the flag = 1 it records it in a dictionary and returns this dictionary.

• Retrieve_L1_Support(Fin_out , strr)

This function is used to retrieve level 1 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the attribute ,‘strr’ ,by which it access the list and get the support value.

• Retrieve_L2_Support(Fin_out , strr1, strr2)

This function is used to retrieve level 2 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the two attributes (strr1, strr2) sby which it access the list and get the support value.

• Retrieve_L3_Support(Fin_out , strr1, strr2, strr3)

This function is used to retrieve level 3 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the three attributes (strr1, strr2, strr3) by which it access the list and get the support value.

• Retrieve_L4_Support(Fin_out , strr1, strr2, strr3, strr4)

This function is used to retrieve level 4 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the four attributes (strr1, strr2, strr3, strr4) by which it access the list and get the support value.

• Retrieve_L5_Support(Fin_out , strr1, strr2, strr3, strr4, strr5)

This function is used to retrieve level 5 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the five attributes (strr1, strr2, strr3, strr4, strr5) by which it access the list and get the support value.

• Retrieve_L6_Support(Fin_out , strr1, strr2, strr3, strr4, strr5, strr6)

This function is used to retrieve level 6 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the six attributes (strr1, strr2, strr3, strr4, strr5, strr6) by which it access the list and get the support value.

• Retrieve_L7_Support(Fin_out , strr1, strr2, strr3, strr4, strr5, strr6, strr7)

This function is used to retrieve level 6 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the seven attributes by which it access the list and get the support value.

• Retrieve_L8_Support(Fin_out , strr1, strr2, strr3, strr4, strr5, strr6, strr7, strr8)

This function is used to retrieve level 6 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the eight attributes by which it access the list and get the support value.

• Retrieve_L9_Support(Fin_out , strr1, strr2, strr3, strr4, strr5, strr6, strr7, strr8, strr9)

This function is used to retrieve level 6 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the nine attributes by which it access the list and get the support value.

• Retrieve_L10_Support(Fin_out , strr1, strr2, strr3, strr4, strr5, strr6, strr7, strr8, strr9, strr10)

This function is used to retrieve level 6 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the ten attributes by which it access the list and get the support value.

• Retrieve_L11_Support(Fin_out , strr1, strr2, strr3, strr4, strr5, strr6, strr7, strr8, strr9, strr10, strr11)

This function is used to retrieve level 6 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the eleven attributes by which it access the list and get the support value.

• Retrieve_L12_Support(Fin_out , strr1, strr2, strr3, strr4, strr5, strr6, strr7, strr8, strr9, strr10, strr11, strr12)

This function is used to retrieve level 6 support from the list of tuples ‘Fin_out’ , its arguments are the support list and the twelve attributes by which it access the list and get the support value.

• Calculate_All(Fin_out)

Function which takes the output from any Retrieve_L{n}_Support function and remove the rules that below minimum confidence and calculate the lift and leverage for each rule that its confidence and support are above the minimum confidence and support ,and prints the right rules in the terminal window.

7. Development Environment

The Apriori algorithm was implemented using Python, which is a free software environment for statistical computing and graphics.

The project was completely developed and debugged by using Pycharm integrated development environment for Python language.

8. Testing Cases and Results

Min support = 90%
Min confidence = = 90% (there is no rule to handle it)

```
C:\Users\MOHAMED\Desktop\BigData>python Support_3.py
Please enter min support in percentage : 90
Please enter min confidence in percentage : 90
you have entered Min support = 90.0 %, and minConfidence = 90.0 %
There is 0 Association Rules generated !
```

Min support = 50%
Min confidence = 50%

```
C:\Users\MOHAMED\Desktop\BigData>python Support_3.py
Please enter min support in percentage : 50
Please enter min confidence in percentage : 50
you have entered Min support = 50.0 %, and minConfidence = 50.0 %
('MINK7512_0',) , support = 0.558
--------------------------------------------------------------------------
--------------------------
('MINK123M_0',) , support = 0.842
--------------------------------------------------------------------------
--------------------------
('PWAPART_0',) , support = 0.598
--------------------------------------------------------------------------
--------------------------
('PPERSAUT_0',) , support = 0.598
--------------------------------------------------------------------------
--------------------------
('PBESAUT_1',) , support = 0.518
--------------------------------------------------------------------------
--------------------------
There is 5 Association Rules generated !
```

Min support = 30%
Min confidence =30%

```
C:\Users\MOHAMED\Desktop\BigData>python Support_3.py
Please enter min support in percentage : 30
Please enter min confidence in percentage : 30
you have entered Min support = 30.0 %, and minConfidence = 30.0 %
--------------------------
('PWABEDR_0',) , support = 0.489
--------------------------------------------------------------------------------
--------------------------
('PWABEDR_6',) , support = 0.398
--------------------------------------------------------------------------------
--------------------------
('PWALAND_0',) , support = 0.458
--------------------------------------------------------------------------------
--------------------------
('PPERSAUT_0',) , support = 0.598
--------------------------------------------------------------------------------
--------------------------
('PPERSAUT_1',) , support = 0.401
--------------------------------------------------------------------------------
--------------------------
('PBESAUT_1',) , support = 0.518
--------------------------------------------------------------------------------
--------------------------
('PBESAUT_0',) , support = 0.458
--------------------------------------------------------------------------------
--------------------------
There is 13 Association Rules generated !
```

Min support = 10%
Min confidence = 10%

```
C:\Users\MOHAMED\Desktop\BigData>python Support_3.py
Please enter min support in percentage : 10
Please enter min confidence in percentage : 10
you have entered Min support = 10.0 %, and minConfidence = 10.0 %
--------------------------------------------------------------------------
--------------------------
('PPERSAUT_0',) , support = 0.598
--------------------------------------------------------------------------
--------------------------
('PPERSAUT_1',) , support = 0.401
--------------------------------------------------------------------------
--------------------------
('PBESAUT_1',) , support = 0.518
--------------------------------------------------------------------------
--------------------------
('PBESAUT_0',) , support = 0.458
--------------------------------------------------------------------------
--------------------------
There is 40 Association Rules generated !
```

Min support = 1%
Min confidence = 1%

```
C:\Users\MOHAMED\Desktop\BigData>python Support_3.py
Please enter min support in percentage : 1
Please enter min confidence in percentage : 1
you have entered Min support = 1.0 %, and minConfidence = 1.0 %
-
PBESAUT_1 ==============>PWAPART_2 support=0.011 ,Confidence=0.03 , lift = 0.058,leverage=-0.179
------------------------------------------------------------------------------------------------
-
PWAPART_0 ==============>PBESAUT_0 support=0.014 ,Confidence=0.023 , lift = 0.051,leverage=-0.26
------------------------------------------------------------------------------------------------
-
PBESAUT_0 ==============>PWAPART_0 support=0.014 ,Confidence=0.023 , lift = 0.051,leverage=-0.26
------------------------------------------------------------------------------------------------
-
PWALAND_0 ==============>PBESAUT_0 support=0.016 ,Confidence=0.035 , lift = 0.076,leverage=-0.194
------------------------------------------------------------------------------------------------
-
PBESAUT_0 ==============>PWALAND_0 support=0.016 ,Confidence=0.035 , lift = 0.076,leverage=-0.194
------------------------------------------------------------------------------------------------
-
There is 84 Association Rules generated !
```

Min Support = 0.1%
Min confidence = 0.1%

```
C:\Users\MOHAMED\Desktop\BigData>python Support_3.py
Please enter min support in percentage : 0.1
Please enter min confidence in percentage : 0.1
you have entered Min support = 0.1 %, and minConfidence = 0.1 %
```

```
MINK7512_6 ==============>PWALAND_4,PPERSAUT_1 support=0.002 ,Confidence=0.007 , lift = 28.986,leverage=0.002
-----------------------------------------------------------------------------------------------
MINK7512_6 ==============>PWALAND_0,PBESAUT_0 support=0.002 ,Confidence=0.007 , lift = 64.516,leverage=0.002
-----------------------------------------------------------------------------------------------
MINK7512_0 ==============>PWALAND_0,PBESAUT_0 support=0.003 ,Confidence=0.007 , lift = 0.173,leverage=-0.014
-----------------------------------------------------------------------------------------------
MINK7512_0 ==============>PWALAND_3,PBESAUT_1 support=0.002 ,Confidence=0.007 , lift = 0.276,leverage=-0.005
-----------------------------------------------------------------------------------------------
MINK7512_6 ==============>PWALAND_3,PBESAUT_1 support=0.001 ,Confidence=0.007 , lift = 76.923,leverage=0.001
-----------------------------------------------------------------------------------------------
MINK7512_6 ==============>PWALAND_4,PBESAUT_1 support=0.002 ,Confidence=0.007 , lift = 142.857,leverage=0.002
-----------------------------------------------------------------------------------------------
MINK7512_0 ==============>PPERSAUT_0,PBESAUT_1 support=0.002 ,Confidence=0.007 , lift = 0.256,leverage=-0.006
-----------------------------------------------------------------------------------------------
MINK7512_6 ==============>PPERSAUT_1,PBESAUT_1 support=0.003 ,Confidence=0.007 , lift = 120.0,leverage=0.003
-----------------------------------------------------------------------------------------------
MINK7512_6 ==============>PPERSAUT_0,PBESAUT_0 support=0.002 ,Confidence=0.007 , lift = 71.429,leverage=0.002
-----------------------------------------------------------------------------------------------
MINK7512_0 ==============>PPERSAUT_0,PBESAUT_0 support=0.003 ,Confidence=0.007 , lift = 0.192,leverage=-0.013
-----------------------------------------------------------------------------------------------
MINK7512_0 ==============>PPERSAUT_1,PBESAUT_1 support=0.002 ,Confidence=0.007 , lift = 0.143,leverage=-0.012
-----------------------------------------------------------------------------------------------
There is 3779 Association Rules generated !
```

9. Conclusion

We can conclude from the project that data mining, even though it takes a lot of processing power, is considered an essential part for the success of many organizations and services. As for our case in this project, the association rules we generate help in predicting which customers are potentially interested in an insurance policy, in describing the actual or potential customers and in explaining why these customers buy a policy. All of which are critical information for any successful insurance company. Additionally, each association rule has its own support and confidence, and the more their values approach 100%, the more the rule is thought to be a fact. So, the user of the system can define his/her own minimum support and minimum confidence based on his/her preferences and act according to the association rules generated.