St. Francis Institute of Technology, Mumbai-400 103

Department Of Information Technology

A.Y. 2024-2025

Class: TE-ITA/B, Semester: VI

Subject: **Business Intelligence Lab**

Experiment – 8: To implement Apriori Association mining algorithm using open source tool WEKA and ORANGE

1. **Aim:** Implementation of Association in Data Mining (Apriori,FPM) in WEKA & Orange

0. **Objectives:** After study of this experiment, the students will be able to implement Apriori Algorithm in WEKA/Orange

0. **Outcomes:** After study of this experiment, the students will be able to

   CO 5: Design and Implement various frequent data mining techniques and formulate association rules on large data sets

0. **Prerequisite:** Introduction to algorithms of Associativity

0. **Requirements:** Personal Computer, Windows XP operating system/Windows 7, Internet Connection, Microsoft Word, WEKA tool, Orange tool.

0. **Theory:**
   a. Introduction to FPM
   a. Introduction to Apriori Algorithm

0. **Laboratory Exercise:** Implementation of Association Algorithm in WEKA & Orange and take printout of implementation along with coding and snapshot.

0. **Post-Experiments Exercise**
   a. **Questions:**
   - Solve numerical for Apriori algorithm
   - Simple CLI execution of Apriori algorithm in WEKA using the following command:
     java weka.associations.Apriori -N 100 -T 1 -C 1.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -I -t directory-path\bank-data-final.arff

   a. **Conclusion:**
   - Summary of Experiment
   - Importance of Experiment
   - Application of Experiment

0. **Reference:** Data Mining: Concept & Techniques, 3rd Edition, Jiawei Han, Micheline Kamber, Jian Pei, Elsevier.

Theory:

Introduction to FPM

What is FPM?

Frequent Pattern Mining is a data mining process that involves identifying recurring relationships, patterns, or associations within large datasets. These patterns may include frequently occurring items, sequences, or substructures. FPM is used to uncover hidden insights that can inform decision-making, guide marketing strategies, detect anomalies, or optimize business processes.

Key Concepts

- Support: This is the frequency or occurrence of a pattern in a dataset. It is often expressed as a percentage of the total number of transactions or data points. A higher support means that the pattern appears more frequently.
- Confidence: In the context of association rules (derived from frequent itemsets), confidence measures how often items in the rule's consequent appear in transactions that contain the rule's antecedent.
- Lift: Lift measures how much more often the items in the rule occur together than would be expected if they were statistically independent. It helps evaluate the strength of a rule.
- Itemset: A set or combination of items that appear together in a transaction or data record.
- Association Rules: These are implications of the form "if-then" that are generated from frequent itemsets. They express the likelihood of the occurrence of an item given the occurrence of another item.

Applications of FPM

- Market Basket Analysis: Retailers analyze customer transaction data to determine product associations. For instance, if customers frequently buy bread and butter together, promotions can be tailored accordingly.
- Web Usage Mining: Analyzing web logs to understand user behavior on websites.
- Bioinformatics: Identifying patterns in biological data such as genetic sequences.
- Fraud Detection: Recognizing unusual patterns that may indicate fraudulent activity.

Introduction to Apriori Algorithm

Overview

The Apriori Algorithm is one of the most popular algorithms used for mining frequent itemsets and generating association rules. It leverages a key principle: if an itemset is frequent, then all of its subsets must also be frequent. This idea is known as the Apriori property.

How Does Apriori Work?

- **Candidate Generation:** The algorithm begins by scanning the dataset to find all frequent 1-itemsets (i.e., individual items that meet a minimum support threshold). Next, it uses these to generate candidate 2-itemsets.
- **Pruning:** The candidate itemsets are pruned by eliminating those that do not meet the minimum support threshold. This step is crucial as it reduces the computational complexity by not considering itemsets that are unlikely to be frequent.
- **Iteration:** The process is repeated iteratively. Frequent k-itemsets are used to generate candidate (k+1)-itemsets until no further itemsets meet the minimum support criteria.
- **Association Rule Generation:** Once all the frequent itemsets have been identified, the algorithm can then generate association rules from these itemsets by computing confidence for each rule. Rules that meet the minimum confidence threshold are considered strong.

Advantages and Limitations

Advantages:

Simplicity: The algorithm is conceptually straightforward and easy to implement.

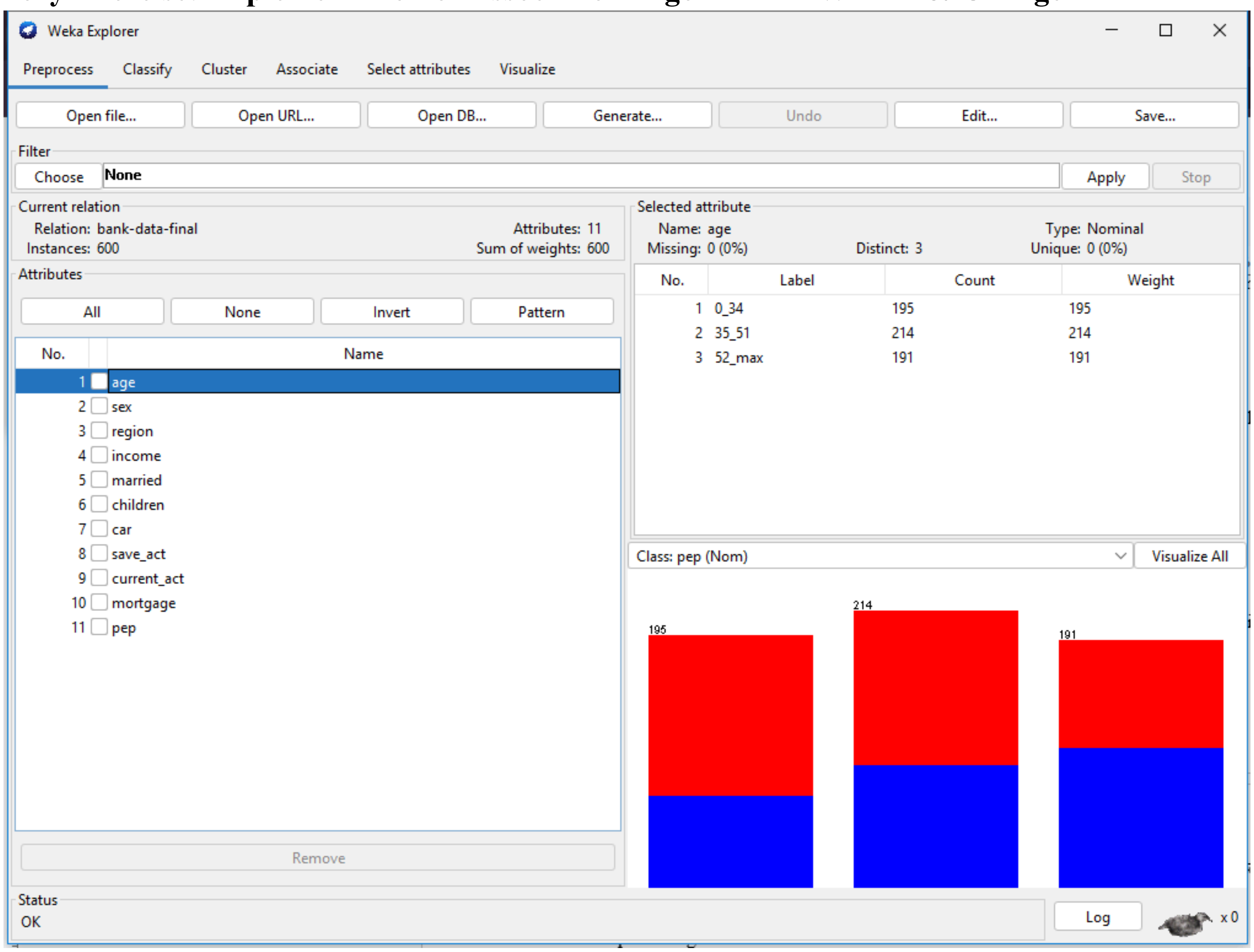
Efficiency (with proper thresholds): When minimum support is set appropriately, the algorithm can quickly reduce the number of candidate itemsets.

Limitations:
Multiple Scans: Apriori requires multiple scans of the dataset, which can be computationally expensive for very large databases.
Exponential Growth: The number of candidate itemsets can grow exponentially, particularly in datasets with many items and low minimum support thresholds.
Sparse Data Issues: In cases of highly sparse datasets, the algorithm may generate a very large number of candidates that barely meet the support criteria, affecting performance.

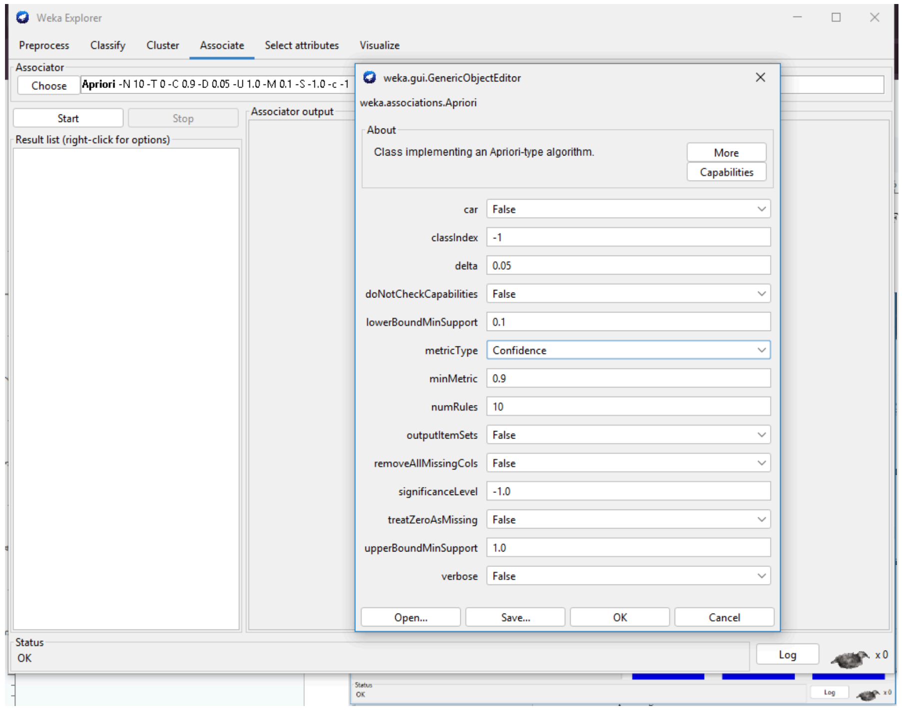
Practical Considerations
Parameter Tuning: Setting the appropriate minimum support and confidence levels is crucial for extracting meaningful patterns without overwhelming the process with too many insignificant rules.

Optimizations: Various optimizations and alternative algorithms (like FP-Growth) have been proposed to overcome some of the inefficiencies of Apriori, especially regarding multiple database scans.

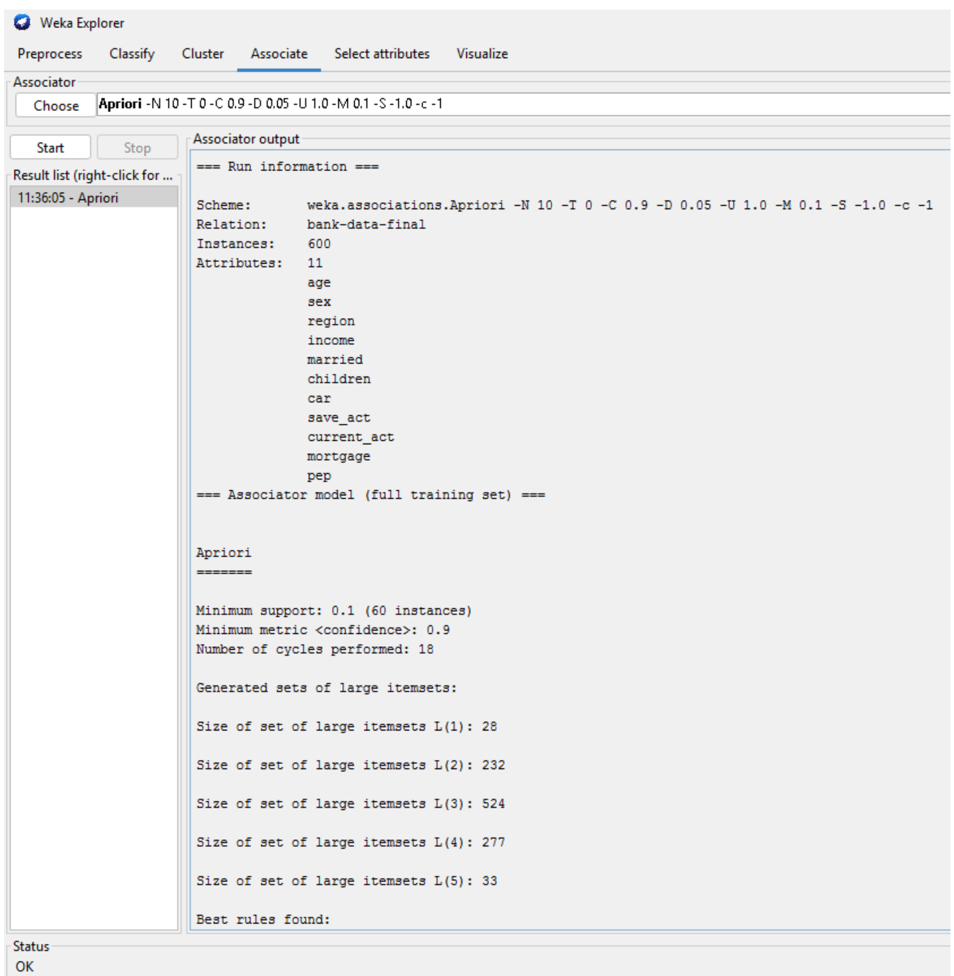
Laboratory Exercise: Implementation of Association Algorithm in WEKA & Orange



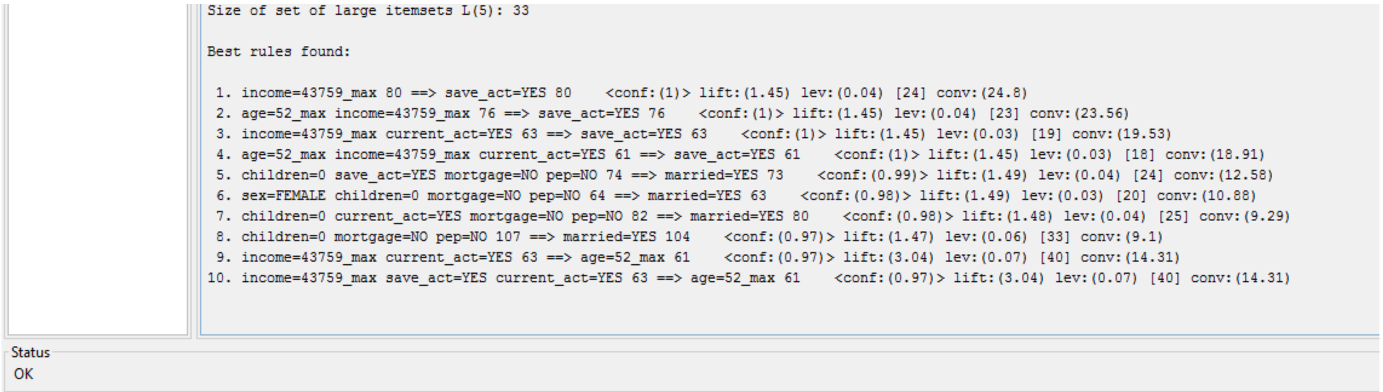
Uploading the dataset to WEKA



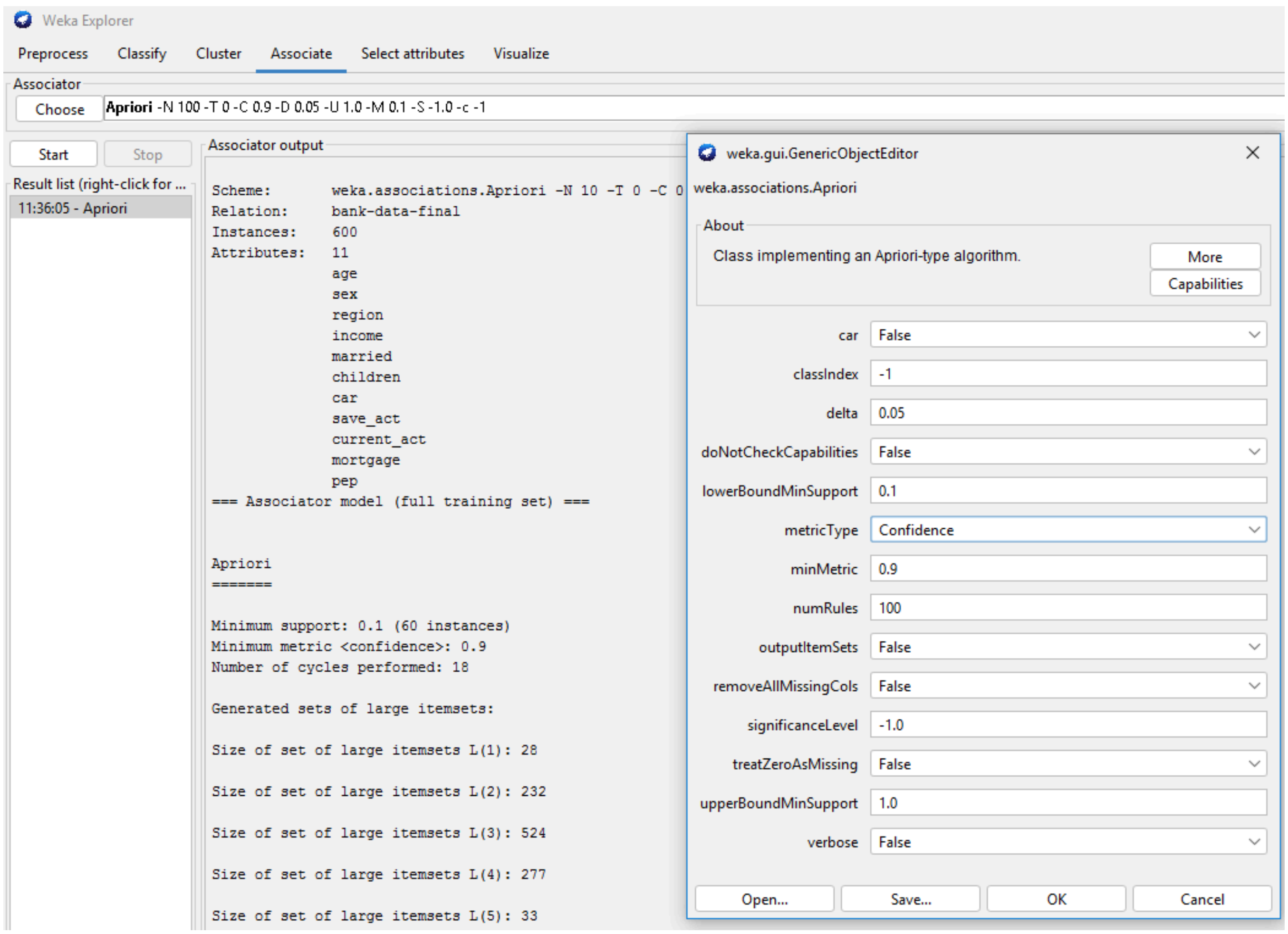
Applying Apriori Algorithm with number of rules=10



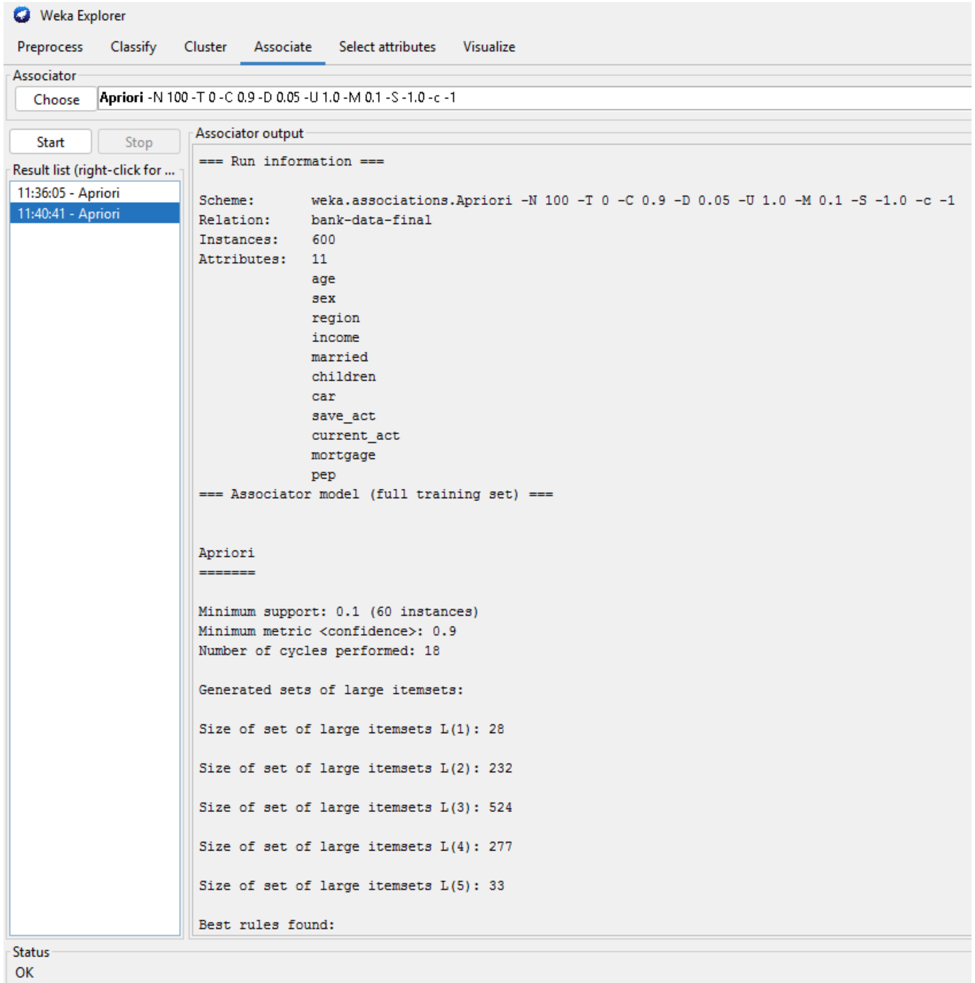
Result of Apriori Algorithm 10 rules (a)

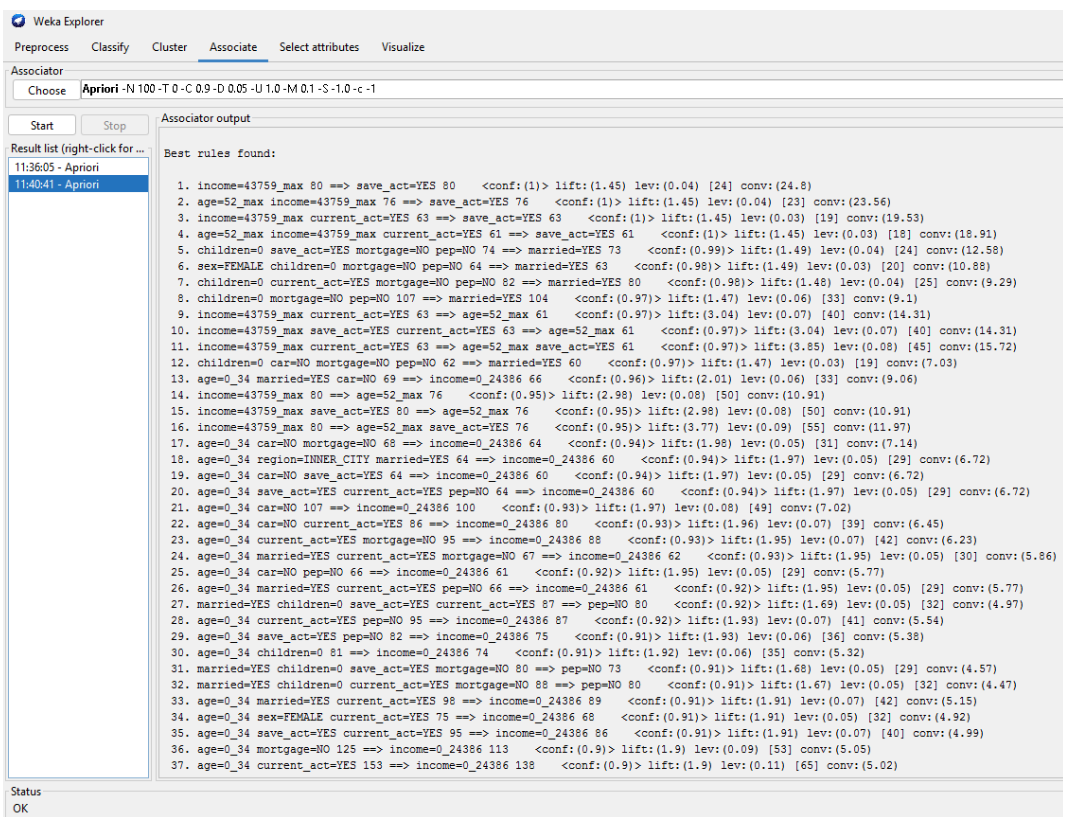


Result of Apriori Algorithm 10 rules (b)



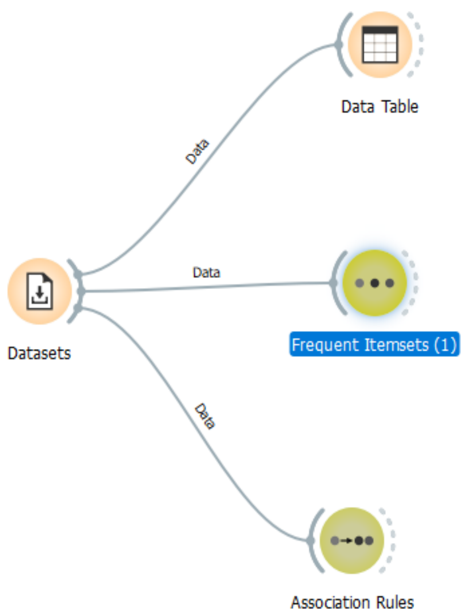
Setting the number of rules as 100

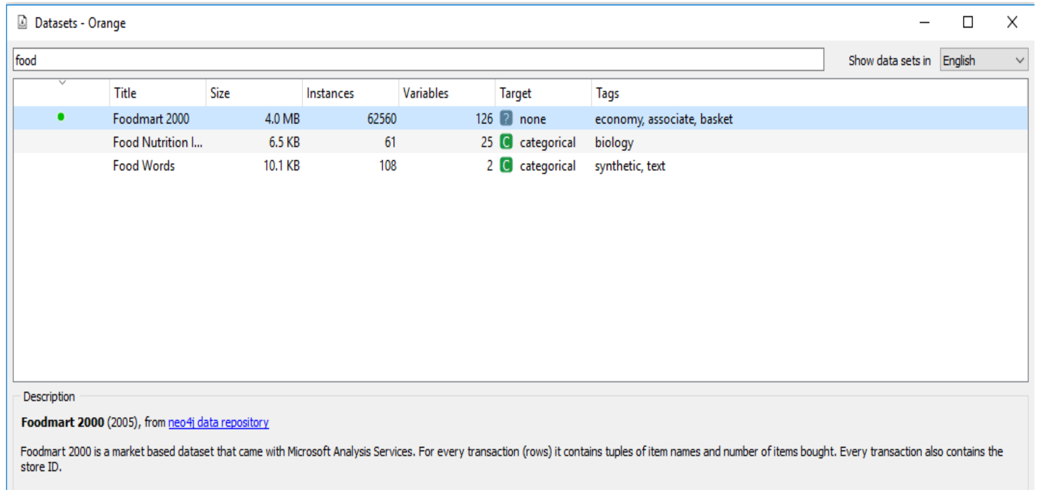




Result of apriori when number of rules are set to 100

Orange



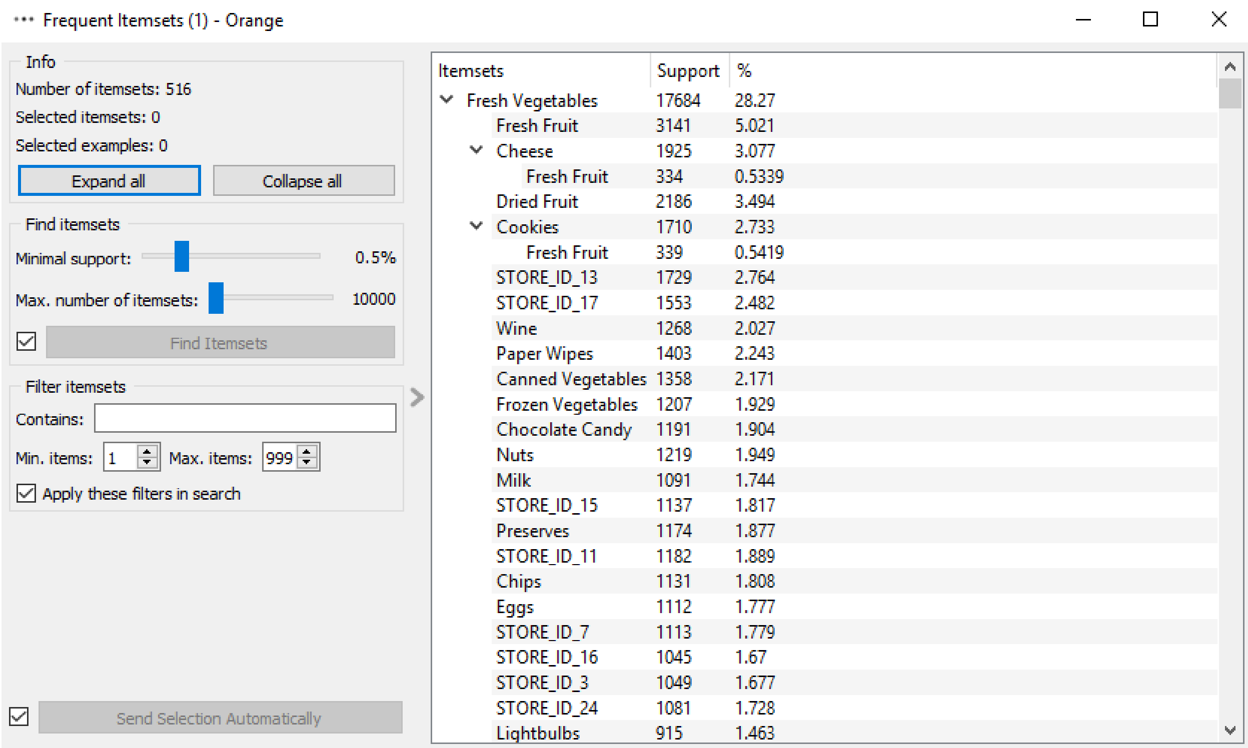


Loading foodmart dataset in orange

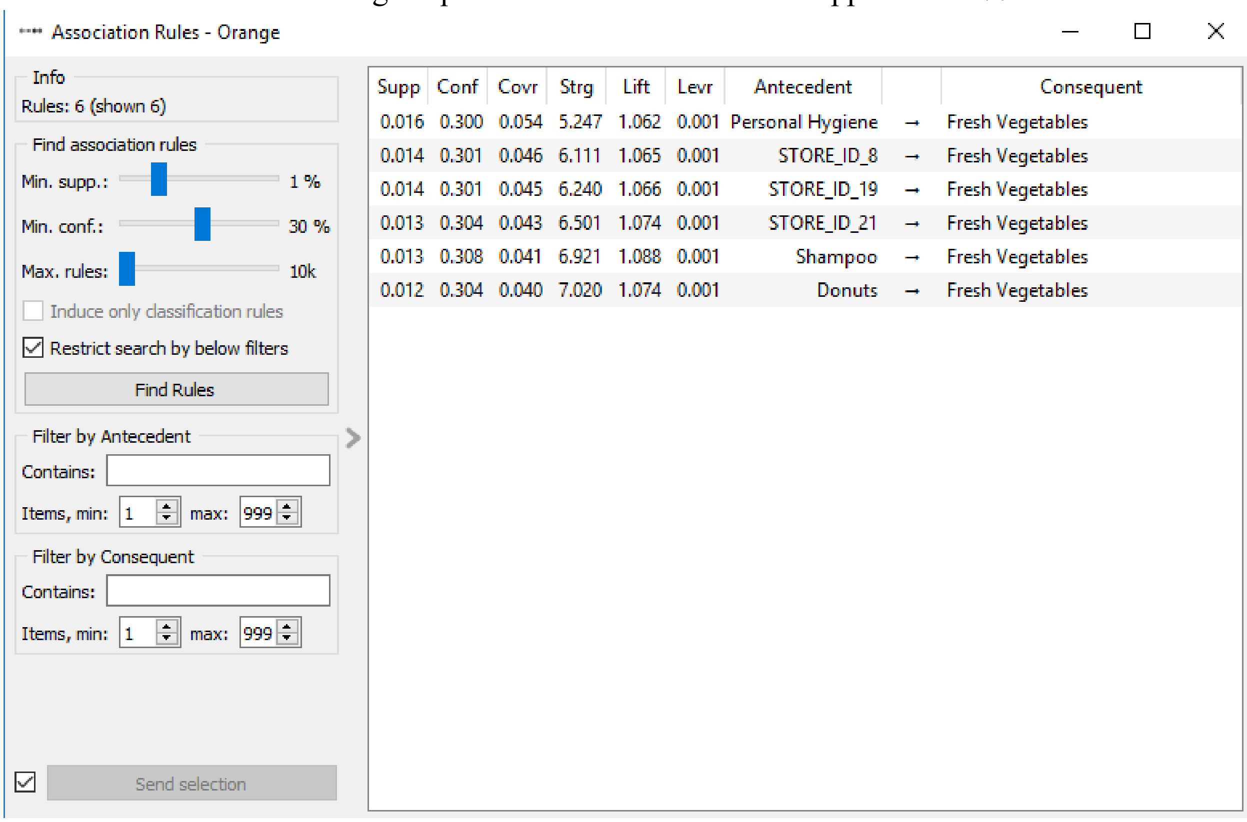
Applying association in Orange

	{...}
1	Pasta=3, Soup=2, STORE_ID_2=1
2	Soup=1, STORE_ID_2=1, Fresh Vegetables=3, Milk=3, Plastic Utensils=2
3	STORE_ID_2=1, Cheese=2, Deodorizers=1, Hard Candy=2, Jam=2
4	STORE_ID_2=1, Fresh Vegetables=2
5	STORE_ID_2=1, Cleaners=1, Cookies=2, Eggs=2, Preserves=1
6	Soup=1, STORE_ID_2=1, Cheese=2, Nasal Sprays=2
7	STORE_ID_2=1, Dips=1, Jelly=3, Tofu=1
8	STORE_ID_2=1, Cookies=2, Preserves=1, Dips=1
9	STORE_ID_2=1, Fresh Vegetables=1, Cleaners=2, Cereal=2, Deli Meats=2, Rice=1
10	Soup=1, STORE_ID_2=1, Jelly=1, Flavored Drinks=1, French Fries=2, Spices=1
11	STORE_ID_2=1, Beer=2, Hot Dogs=2, Personal Hygiene=2
12	STORE_ID_2=1, Fresh Vegetables=2, Cookies=2, Eggs=3, Bologna=2, Cooking Oil=2, Donuts=1
13	STORE_ID_2=1, Cookies=1, Fresh Fruit=2, Peanut Butter=1, Sliced Bread=2
14	STORE_ID_2=1, Fresh Vegetables=2, Dried Fruit=1, Paper Wipes=2, Sauces=1
15	Soup=2, STORE_ID_2=1, Milk=1, Fresh Fruit=1, Chocolate Candy=1, Cottage Cheese=2, Waffles=1
16	STORE_ID_2=1, Nasal Sprays=2, Dips=2, Personal Hygiene=2, Sliced Bread=1, Chips=2, Soda=2
17	STORE_ID_2=1, Fresh Vegetables=1, Peanut Butter=1, Sauces=1, Canned Vegetables=3, Juice=4, Popcorn=1
18	STORE_ID_2=1, Fresh Vegetables=1, French Fries=2, Fresh Fruit=2, Soda=1, Frozen Vegetables=2
19	STORE_ID_2=1, Fresh Vegetables=2, Canned Vegetables=2, Juice=2, Coffee=2, Gum=2
20	Soup=1, STORE_ID_2=1, Dried Fruit=2
21	STORE_ID_2=1, Cheese=2, Cookies=2, Fresh Fruit=2, Lightbulbs=4, Shampoo=2
22	STORE_ID_2=1, Rice=1, Bologna=2, Fresh Fruit=1
23	STORE_ID_2=1, Cheese=1, Coffee=1, Ice Cream=2
24	STORE_ID_2=1, Lightbulbs=1, Muffins=1
25	STORE_ID_2=1, Bologna=3, Soda=1, Canned Vegetables=2, Tuna=3
26	Soup=2, STORE_ID_2=1, Cooking Oil=2, Juice=2
27	STORE_ID_2=1, Spices=3, Fresh Fruit=3, Dried Fruit=3, Chips=1
28	STORE_ID_2=1, Waffles=1, Canned Vegetables=1, Muffins=2, Pots and Pans=1
29	STORE_ID_2=1, Bologna=2, Mouthwash=3
30	STORE_ID_2=1, Eggs=2, Soda=2
31	STORE_ID_2=1, Fresh Vegetables=1, Plastic Utensils=1, Jam=1, Paper Wipes=3, Chocolate Candy=2, Hamburger=2
32	STORE_ID_2=1, Fresh Vegetables=2, Maps=2
33	STORE_ID_2=1, Fresh Vegetables=2, Spices=2, Waffles=2, Candles=1
34	STORE_ID_2=1, Muffins=1, Maps=2, Tools=1
35	STORE_ID_2=1, Fresh Vegetables=3, Fresh Fruit=2
36	STORE_ID_2=1, Shampoo=2
37	STORE_ID_2=1, Milk=1, Jam=2, Cookies=2, Popsicles=2
38	STORE_ID_2=1, Fresh Fruit=1, Waffles=2
39	STORE_ID_2=1, Preserves=3, Deli Meats=2, Donuts=3, Fresh Fruit=1, Paper Wipes=3, Canned Vegetables=3
40	STORE_ID_2=1, Cleaners=2, Nasal Sprays=3, Cereal=3, Chocolate Candy=2, Frozen Chicken=3, Toilet Brushes=3
41	STORE_ID_2=1, Fresh Vegetables=4, Milk=2, Dried Fruit=1, Soda=1, Fresh Chicken=1

Visualizing the dataset in data table



Generating frequent itemsets with minimal support of 0.5%



Associations rules generated in Orange

POST EXPERIMENT EXERCISE

Simple CLI execution of Apriori algorithm in WEKA using the following command:

java weka.associations.Apriori -N 100 -T 1 -C 1.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -I -t directory-path\bank-data-final.arff

```
> java weka.associations.Apriori -N 100 -T 1 -C 1.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -I -t C:\Users\Student\Desktop\Tanmay\exp8\bank-data-final.arff


Apriori
=======

Minimum support: 0.1 (60 instances)
Minimum metric <lift>: 1.5
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 28

Large Itemsets L(1):
age=0_34 195
age=35_51 214
age=52_max 191
sex=FEMALE 300
sex=MALE 300
region=INNER_CITY 269
region=TOWN 173
region=RURAL 96
region=SUBURBAN 62
income=0_24386 285
income=24387_43758 235
income=43759_max 80
married=NO 204
married=YES 396
children=0 263
children=1 135
children=2 134
```

```
SimpleCLI
 59. age=0_34 sex=FEMALE 93 ==> income=0_24386 current_act=YES 68    conf:(0.73) < lift:(2.04)> lev:(0.06) [34] conv:(2.3)
 60. income=0_24386 current_act=YES 215 ==> age=0_34 sex=FEMALE 68    conf:(0.32) < lift:(2.04)> lev:(0.06) [34] conv:(1.23)
 61. age=0_34 married=YES 128 ==> income=0_24386 current_act=YES mortgage=NO 62    conf:(0.48) < lift:(2.03)> lev:(0.05) [31] conv:(1.46)
 62. income=0_24386 current_act=YES mortgage=NO 143 ==> age=0_34 married=YES 62    conf:(0.43) < lift:(2.03)> lev:(0.05) [31] conv:(1.37)
 63. age=0_34 195 ==> income=0_24386 current_act=YES pep=NO 87    conf:(0.45) < lift:(2.03)> lev:(0.07) [44] conv:(1.4)
 64. income=0_24386 current_act=YES pep=NO 132 ==> age=0_34 87    conf:(0.66) < lift:(2.03)> lev:(0.07) [44] conv:(1.94)
 65. age=0_34 mortgage=NO 125 ==> income=0_24386 save_act=YES 72    conf:(0.58) < lift:(2.02)> lev:(0.06) [36] conv:(1.66)
 66. income=0_24386 save_act=YES 171 ==> age=0_34 mortgage=NO 72    conf:(0.42) < lift:(2.02)> lev:(0.06) [36] conv:(1.35)
 67. age=0_34 married=YES current_act=YES 98 ==> income=0_24386 mortgage=NO 62    conf:(0.63) < lift:(2.02)> lev:(0.05) [31] conv:(1.82)
 68. income=0_24386 mortgage=NO 188 ==> age=0_34 married=YES current_act=YES 62    conf:(0.33) < lift:(2.02)> lev:(0.05) [31] conv:(1.24)
 69. age=0_34 195 ==> income=0_24386 car=NO current_act=YES 80    conf:(0.41) < lift:(2.02)> lev:(0.07) [40] conv:(1.34)
 70. income=0_24386 car=NO current_act=YES 122 ==> age=0_34 80    conf:(0.66) < lift:(2.02)> lev:(0.07) [40] conv:(1.92)
 71. age=0_34 save_act=YES 119 ==> income=0_24386 current_act=YES 86    conf:(0.72) < lift:(2.02)> lev:(0.07) [43] conv:(2.25)
 72. income=0_24386 current_act=YES 215 ==> age=0_34 save_act=YES 86    conf:(0.4) < lift:(2.02)> lev:(0.07) [43] conv:(1.33)
 73. age=0_34 married=YES car=NO 69 ==> income=0_24386 66    conf:(0.96) < lift:(2.01)> lev:(0.06) [33] conv:(9.06)
 74. income=0_24386 285 ==> age=0_34 married=YES car=NO 66    conf:(0.23) < lift:(2.01)> lev:(0.06) [33] conv:(1.15)
 75. income=0_24386 married=YES 195 ==> age=0_34 current_act=YES mortgage=NO 62    conf:(0.32) < lift:(2.01)> lev:(0.05) [31] conv:(1.22)
 76. age=0_34 current_act=YES mortgage=NO 95 ==> income=0_24386 married=YES 62    conf:(0.65) < lift:(2.01)> lev:(0.05) [31] conv:(1.89)
 77. age=0_34 195 ==> income=0_24386 save_act=YES current_act=YES 86    conf:(0.44) < lift:(2)> lev:(0.07) [43] conv:(1.38)
 78. income=0_24386 save_act=YES current_act=YES 132 ==> age=0_34 86    conf:(0.65) < lift:(2)> lev:(0.07) [43] conv:(1.9)
 79. children=1 135 ==> save_act=YES current_act=YES pep=YES 63    conf:(0.47) < lift:(2)> lev:(0.05) [31] conv:(1.42)
 80. save_act=YES current_act=YES pep=YES 140 ==> children=1 63    conf:(0.45) < lift:(2)> lev:(0.05) [31] conv:(1.39)
 81. age=0_34 195 ==> income=0_24386 married=YES current_act=YES pep=NO 61    conf:(0.31) < lift:(2)> lev:(0.05) [30] conv:(1.22)
 82. income=0_24386 married=YES current_act=YES pep=NO 94 ==> age=0_34 61    conf:(0.65) < lift:(2)> lev:(0.05) [30] conv:(1.87)
 83. age=0_34 married=YES 128 ==> income=0_24386 mortgage=NO 80    conf:(0.63) < lift:(1.99)> lev:(0.07) [39] conv:(1.79)
 84. income=0_24386 mortgage=NO 188 ==> age=0_34 married=YES 80    conf:(0.43) < lift:(1.99)> lev:(0.07) [39] conv:(1.36)
 85. children=1 135 ==> save_act=YES pep=YES 80    conf:(0.59) < lift:(1.99)> lev:(0.07) [39] conv:(1.69)
 86. save_act=YES pep=YES 179 ==> children=1 80    conf:(0.45) < lift:(1.99)> lev:(0.07) [39] conv:(1.39)
 87. income=0_24386 285 ==> age=0_34 car=NO mortgage=NO 64    conf:(0.22) < lift:(1.98)> lev:(0.05) [31] conv:(1.14)
 88. age=0_34 car=NO mortgage=NO 68 ==> income=0_24386 64    conf:(0.94) < lift:(1.98)> lev:(0.05) [31] conv:(7.14)
 89. age=0_34 current_act=YES 153 ==> income=0_24386 save_act=YES pep=NO 60    conf:(0.39) < lift:(1.98)> lev:(0.05) [29] conv:(1.3)
 90. income=0_24386 save_act=YES pep=NO 119 ==> age=0_34 current_act=YES 60    conf:(0.5) < lift:(1.98)> lev:(0.05) [29] conv:(1.48)
 91. income=0_24386 married=YES 195 ==> age=0_34 current_act=YES pep=NO 61    conf:(0.31) < lift:(1.98)> lev:(0.05) [30] conv:(1.22)
 92. age=0_34 current_act=YES pep=NO 95 ==> income=0_24386 married=YES 61    conf:(0.64) < lift:(1.98)> lev:(0.05) [30] conv:(1.83)
 93. age=0_34 195 ==> income=0_24386 current_act=YES 138    conf:(0.71) < lift:(1.97)> lev:(0.11) [68] conv:(2.16)
 94. income=0_24386 current_act=YES 215 ==> age=0_34 138    conf:(0.64) < lift:(1.97)> lev:(0.11) [68] conv:(1.86)
 95. age=0_34 region=INNER_CITY married=YES 64 ==> income=0_24386 60    conf:(0.94) < lift:(1.97)> lev:(0.05) [29] conv:(6.72)
 96. age=0_34 car=NO save_act=YES 64 ==> income=0_24386 60    conf:(0.94) < lift:(1.97)> lev:(0.05) [29] conv:(6.72)
 97. age=0_34 save_act=YES current_act=YES pep=NO 64 ==> income=0_24386 60    conf:(0.94) < lift:(1.97)> lev:(0.05) [29] conv:(6.72)
 98. income=0_24386 285 ==> age=0_34 region=INNER_CITY married=YES 60    conf:(0.21) < lift:(1.97)> lev:(0.05) [29] conv:(1.13)
 99. income=0_24386 285 ==> age=0_34 car=NO save_act=YES 60    conf:(0.21) < lift:(1.97)> lev:(0.05) [29] conv:(1.13)
100. income=0_24386 285 ==> age=0_34 save_act=YES current_act=YES pep=NO 60    conf:(0.21) < lift:(1.97)> lev:(0.05) [29] conv:(1.13)

=== Evaluation ===

Elapsed time: 0.031s
```

100 rules generated by apriori algorithm on simple cli weka