Akademia WSB

Market Basket Analysis using Apriori algorithm

Subject: Analytical processing of business data
Master’s Degree in Management, Data Science Specialization
Teacher: Dr. Karol Jędrasiak
Student: Rahul Rahul

Introduction

Apriori is an algorithm for frequent item set mining and association rule learning over transactional database. 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 domain such as market basket analysis.

Basically, it is a data mining technique to find association rules in a dataset.

Tool used in this project: Weka 3.8.4

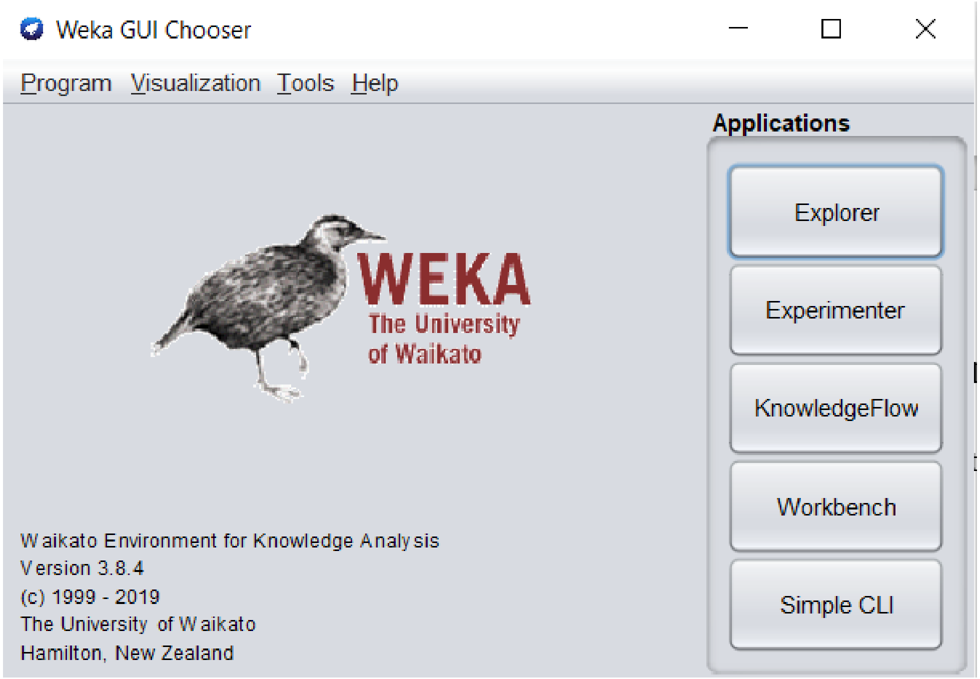
Dataset

I am using the default supermarket dataset given by Weka software titled “supermarket.arff”. The dataset contains a total of 4627 transactions (rows) and 217 attributes (columns). The objective of this project is to predict items the customers frequently buy together by generating a set of rules called Association Rules.

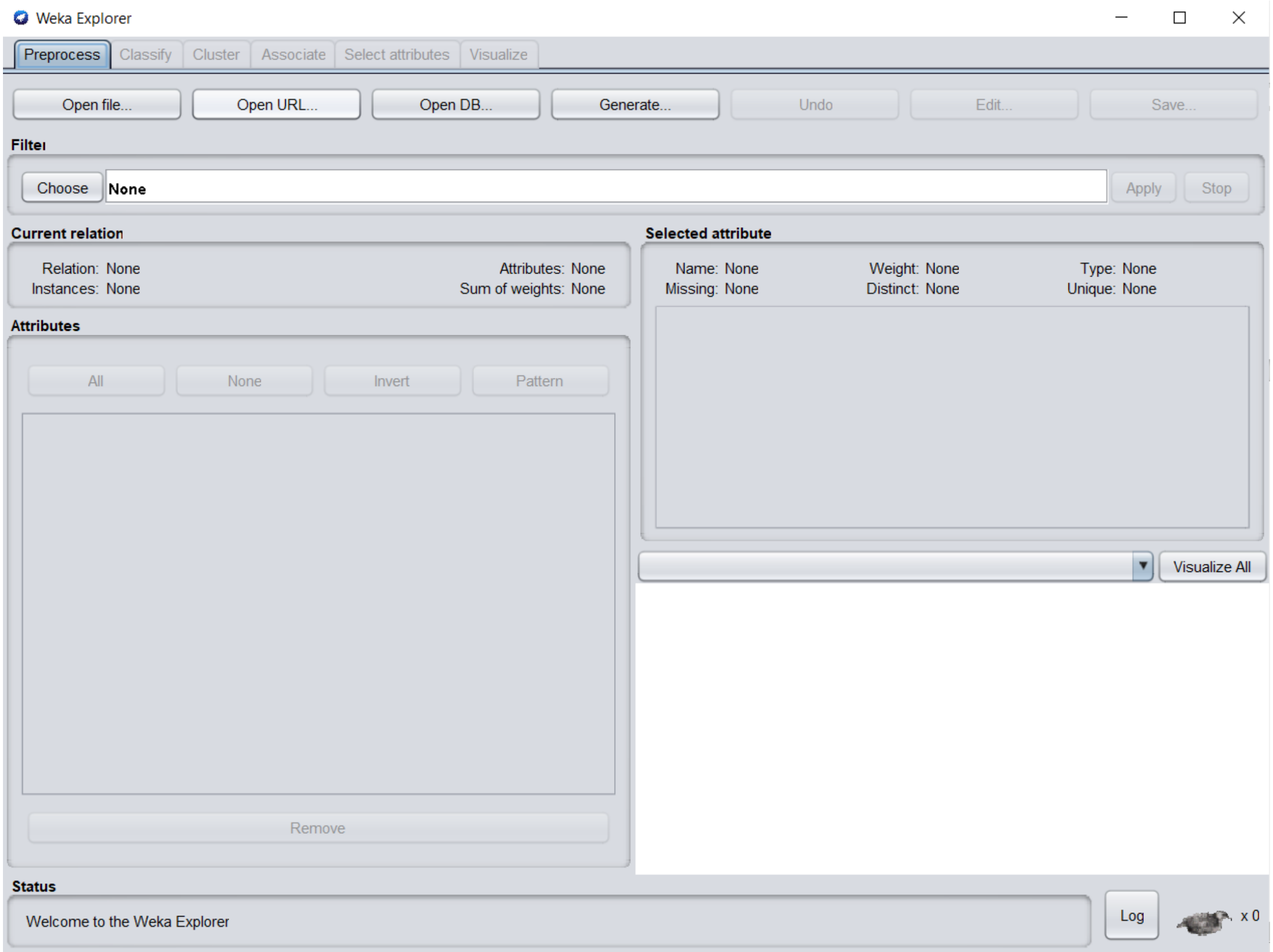
Before we go ahead and work on Apriori algorithm, we need to make sure that all the attributes in our dataset are nominal, binary or unary. This is because Apriori algorithm is only applicable to nominal, binary and unary attributes. If the data is not in Nominal, then we first convert it in to Nominal.

Implementation steps

Step 1: Select Explorer, then Open file



Waikato Environment for Knowledge Analysis
Version 3.8.4
(c) 1999 - 2019
The University of Waikato
Hamilton, New Zealand



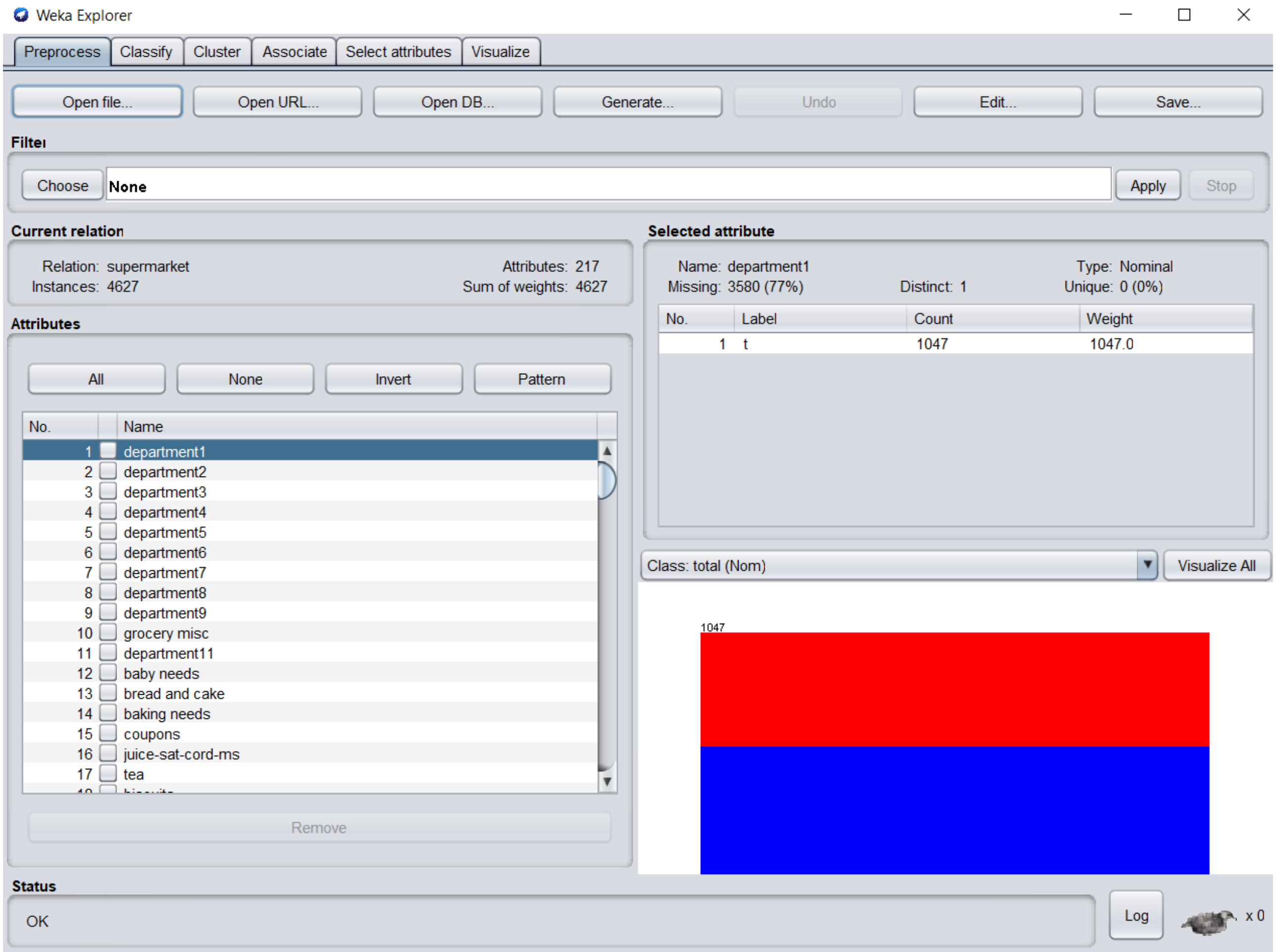
Filter: None

Relation: None
Instances: None
Attributes: None
Sum of weights: None

Name: None
Missing: None
Weight: None
Distinct: None
Type: None
Unique: None

Welcome to the Weka Explorer

Step 2: To make sure all the attributes are Nominal in a Dataset



Filter

None

Current relation

Relation: supermarket
Instances: 4627
Attributes: 217
Sum of weights: 4627

Selected attribute

Name: department1
Missing: 3580 (77%)
Distinct: 1
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	t	1047	1047.0

Class: total (Nom)

Attributes

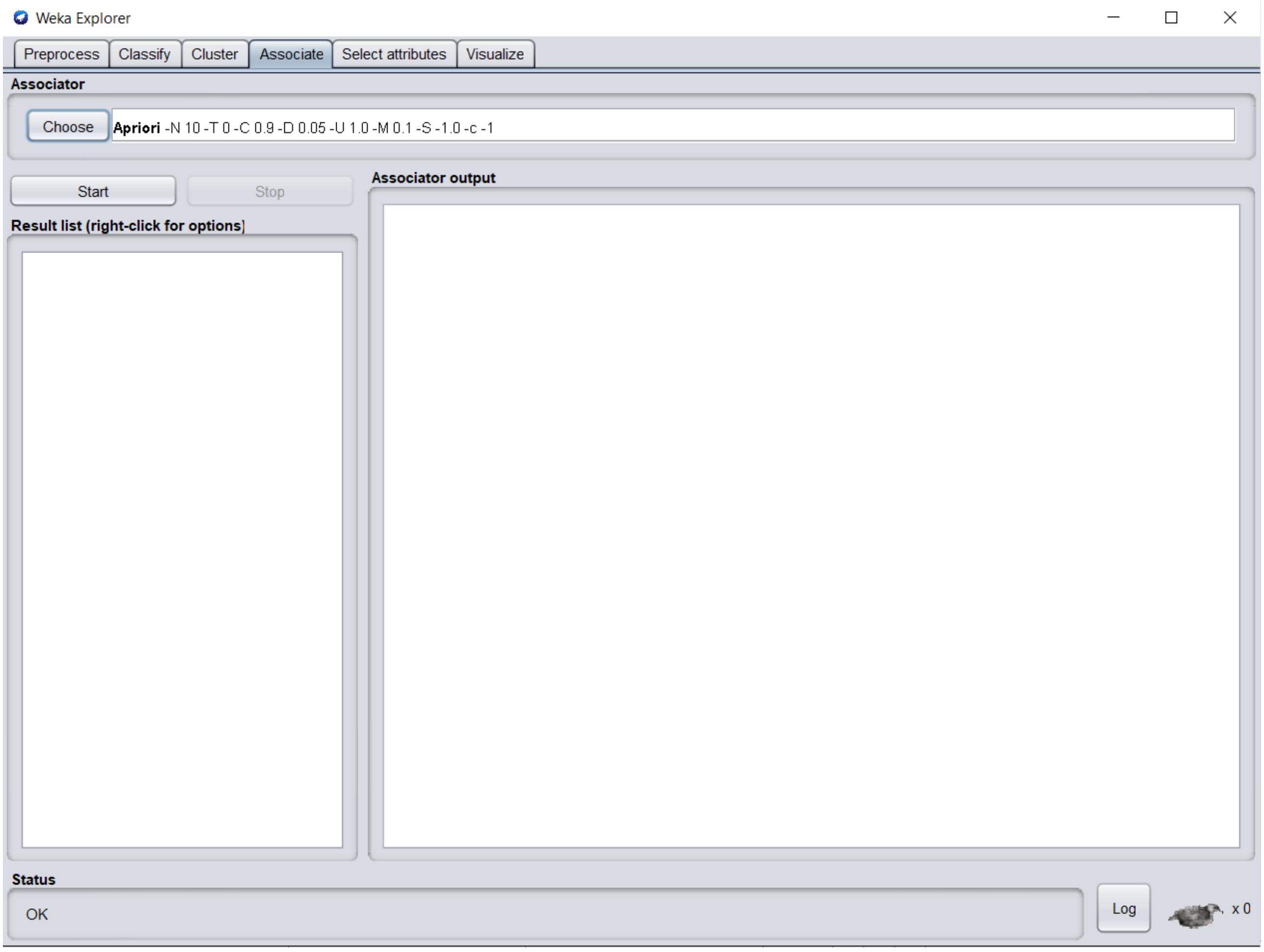
No.		Name
1	□	department1
2	□	department2
3	□	department3
4	□	department4
5	□	department5
6	□	department6
7	□	department7
8	□	department8
9	□	department9
10	□	grocery misc
11	□	department11
12	□	baby needs
13	□	bread and cake
14	□	baking needs
15	□	coupons
16	□	juice-sat-cord-ms
17	□	tea

Status

OK

As you can see the dataset contain total 217 attributes (columns) and 4627 instances (rows). We can also confirm that the “department1” attribute is a Nominal type. In the same way you can check the attribute type of the remaining attributes just by selecting that column. Since, I checked each attribute and found all the attributes of Nominal type. We can proceed and apply Apriori Algorithm.

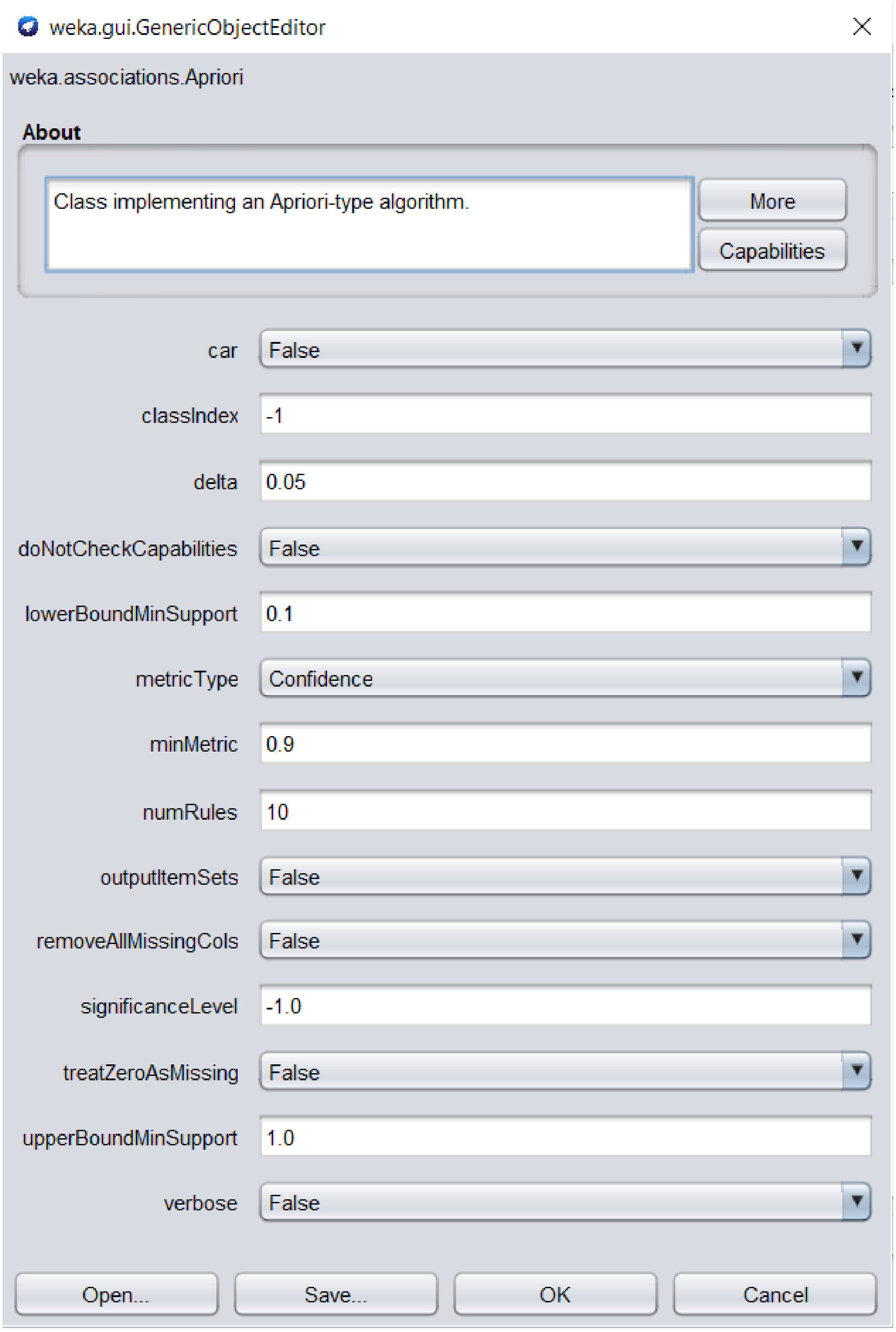
Step 3: Select “Associate” tab to find association rules in a Dataset.



Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

OK

Step 4: By default Weka has Apriori Algorithm in there. So, click on it to see different components of it.



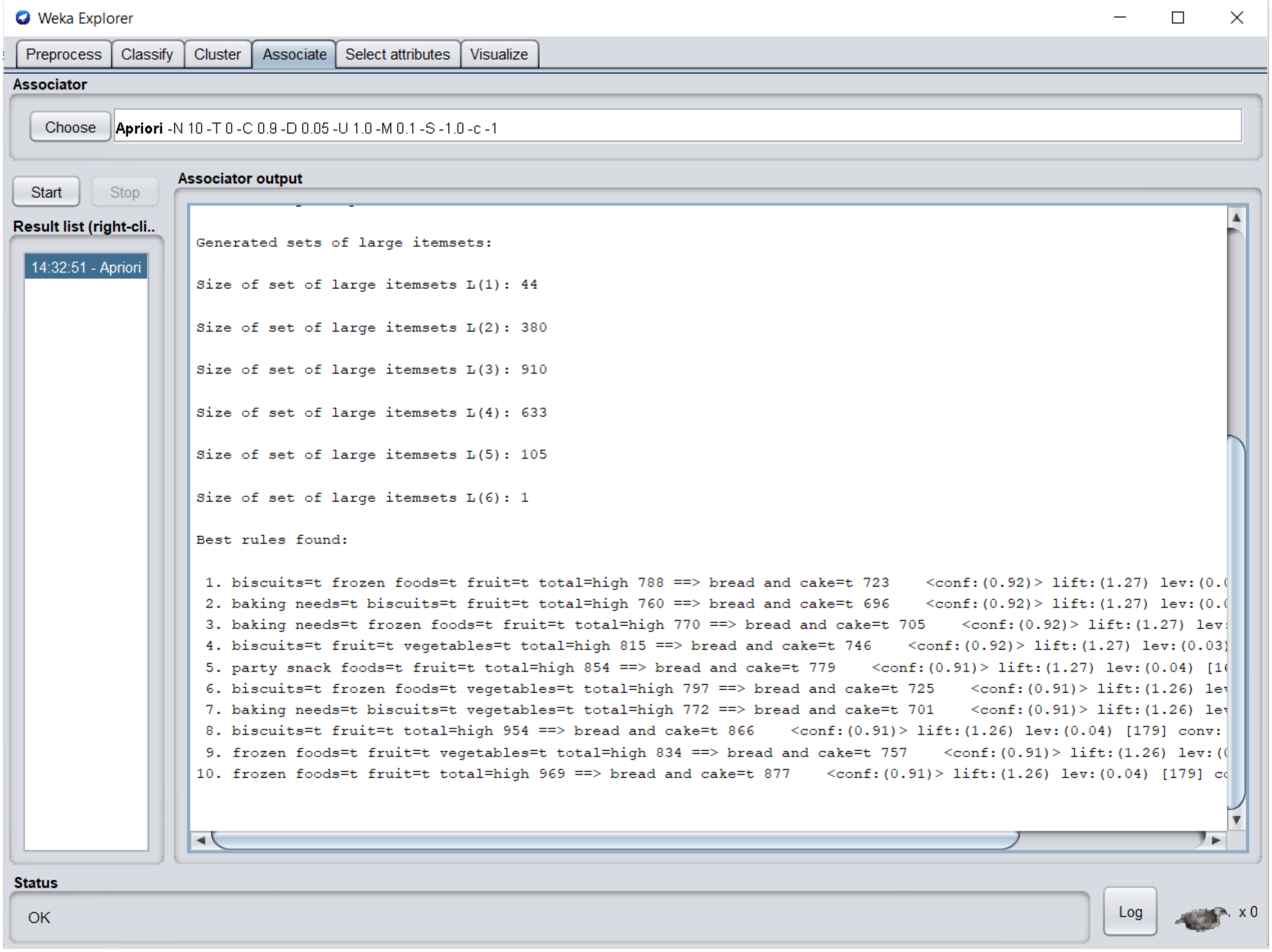
Class implementing an Apriori-type algorithm.

car: False
classIndex: -1
delta: 0.05
doNotCheckCapabilities: False
lowerBoundMinSupport: 0.1
metricType: Confidence
minMetric: 0.9
numRules: 10
outputItemSets: False
removeAllMissingCols: False
significanceLevel: -1.0
treatZeroAsMissing: False
upperBoundMinSupport: 1.0
verbose: False

The main components that are applicable are-

- **Lower bound min support:** lower interval
- **Upper bound minimum support:** upper interval
- **Delta:** Increment level
- **Metric type:** how we rank our association rules. (most common is confidence)
- **Min Metric:** min value of used metric, higher the confidence better is the rank.
- **Num Rules:** number of rule we want.

Step 5: Now apply Apriori algorithm on the Dataset. Press start and we can see, we get the required output.



Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

14:32:51 - Apriori

```
Generated sets of large itemsets:

Size of set of large itemsets L(1): 44

Size of set of large itemsets L(2): 380

Size of set of large itemsets L(3): 910

Size of set of large itemsets L(4): 633

Size of set of large itemsets L(5): 105

Size of set of large itemsets L(6): 1

Best rules found:

 1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    <conf:(0.92)> lift:(1.27) lev:(0.
 2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <conf:(0.92)> lift:(1.27) lev:(0.
 3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705    <conf:(0.92)> lift:(1.27) lev:
 4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    <conf:(0.92)> lift:(1.27) lev:(0.03)
 5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    <conf:(0.91)> lift:(1.27) lev:(0.04) [1
 6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725    <conf:(0.91)> lift:(1.26) le
 7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701    <conf:(0.91)> lift:(1.26) le
 8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:
 9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757    <conf:(0.91)> lift:(1.26) lev:(
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] c
```

OK

The output we got is:

=== Run information ===

Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation: supermarket
Instances: 4627
Attributes: 217
[list of attributes omitted]
=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.15 (694 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 44

Size of set of large itemsets L(2): 380

Size of set of large itemsets L(3): 910

Size of set of large itemsets L(4): 633

Size of set of large itemsets L(5): 105

Size of set of large itemsets L(6): 1

Best rules found:

1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723   <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)

2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696   <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)

3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705   <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)

4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746   <conf:(0.92)> lift:(1.27) lev:(0.03) [159] conv:(3.26)

5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779   <conf:(0.91)> lift:(1.27) lev:(0.04) [164] conv:(3.15)

6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725   <conf:(0.91)> lift:(1.26) lev:(0.03) [151] conv:(3.06)

7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701   <conf:(0.91)> lift:(1.26) lev:(0.03) [145] conv:(3.01)

8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866   <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(3)

9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757   <conf:(0.91)> lift:(1.26) lev:(0.03) [156] conv:(3)

10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877   <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(2.92)

Explanation and observation of output

- The first section of the output is basically the summary of what we did, i.e. which algorithm we used and what was the dataset we applied it to.
- The second section is the main output that contains the details of the algorithm performed on the dataset and the required association rules.
- It shows that are minimum support is 0.15, so we stopped right at 694 instances.
- The number of cycles performed is 17 that means we repeated the process 17 times to find top 10 rules
- We found 6 large item-sets (not ranked)
- Now we finally have our 10 rules. Let us understand how to interpret these rules.

Let’s interpret the first Rule

The first rule is saying if a consumer is buying “biscuits”, “frozen food” and “fruit”, then they are also buying “bread and cake” with a confidence of 0.92 or 92 % and support of 0.15. Since the confidence value (0.92) is greater than minMetric value (0.9) that we defined in Step 4. This association rule is a strong rule.

And also we can see that the rules are ranked according to the confidence. Therefore, first rule is more impactful compared to other derived rules

In the same way we can interpret the rest of the rules.

Conclusion

In my opinion Weka is a convenient tool to implement data mining techniques. Without writing single line of code, I managed to implement market basket analysis using Apriori algorithm. Apart from association rules problem this software can also solve other unsupervised learning such as clustering problems etc.

In this supermarket dataset we successfully managed to predict top association rules. These rules can be used in numerous marketing strategies such as:

- Changing the supermarket store layout according to trends.
- Customer behaviour analysis.
- Catalogue design
- Customized emails with add-on sales
- To identify the trending items the customers are buying