

Drexel University
College of Computing and Informatics
INFO 371 – Data Mining Applications
Assignment 3

Name: Yuming Chen

Student Number: 320180939611

Due Date: 11:59pm, Sunday, May 9, 2021

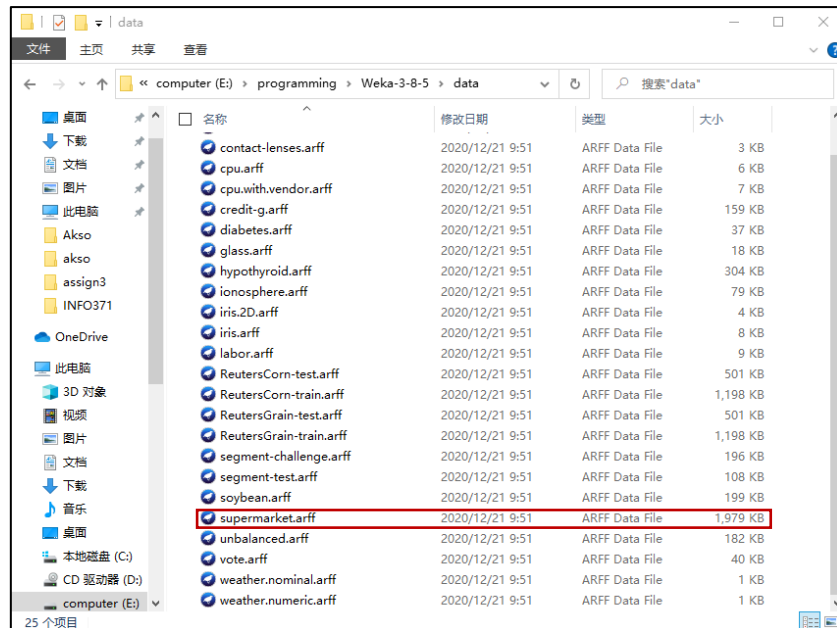
Catalog

Task 1. Mining Association Rules on a Given Data Set	1
1. Data Loading and Overview	1
2. Running in Weka.....	2
3. Analyzing.....	8
Task 2. Pre-processing Dirty Data for Association Rule Mining	15
1. Data preparation.....	15
2. Clean up the values and Convert the data.....	16
Task 3. Mining Association Rules for the Bread Basket Data	18
1. Data Loading	18
2. Set for Data Preprocessing.....	18
3. Apply Data Preprocessing.	19
4. Remove Transaction Attribute.....	19
5. Ready for Associate operation	20
6. Set for Apriori algorithm	20
7. Runing by using the default settings.....	21
8. Analyze the result	22
9. Re-run the A-priori with changed settings and Analyze the result.....	23

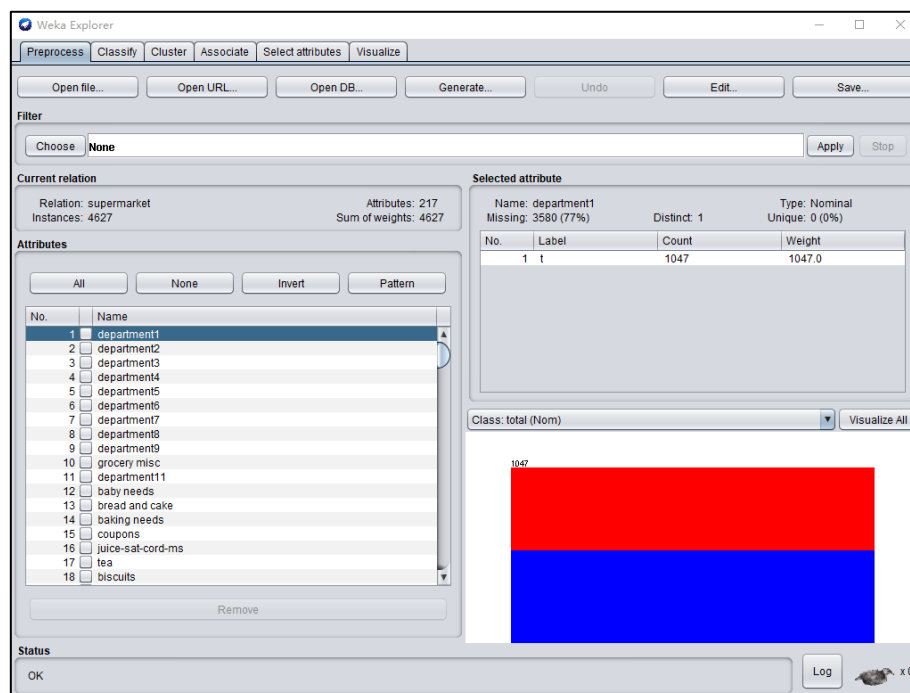
Task 1. Mining Association Rules on a Given Data Set

1. Data Loading and Overview

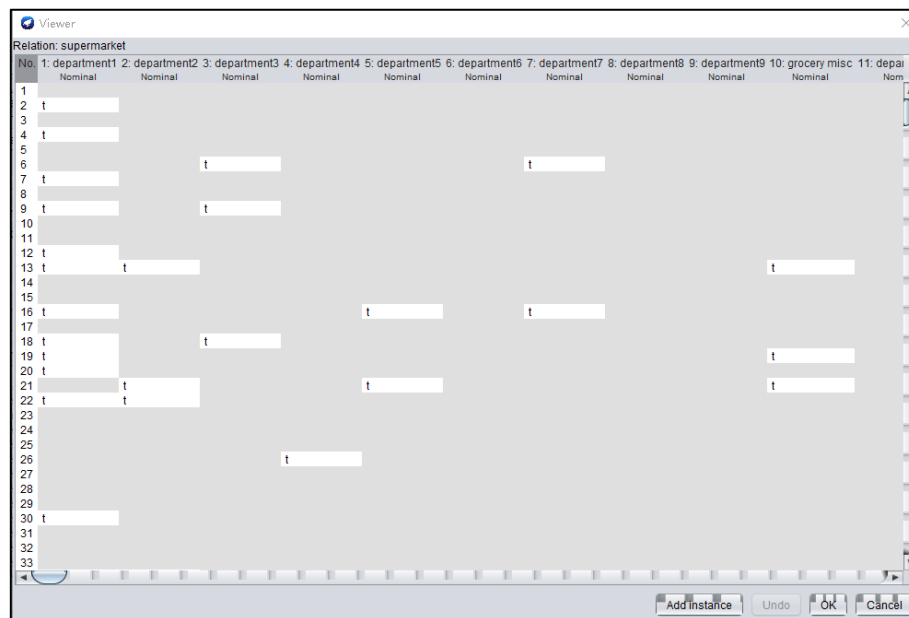
1.1 Go to weka installation folder. Under the installation folder, there is **data** folder. Open the **data** folder and find the data set **supermarket.arff**



1.2 Open **supermarket.arff** in Weka Explorer



1.3 View the content of the file using the 'Edit' button

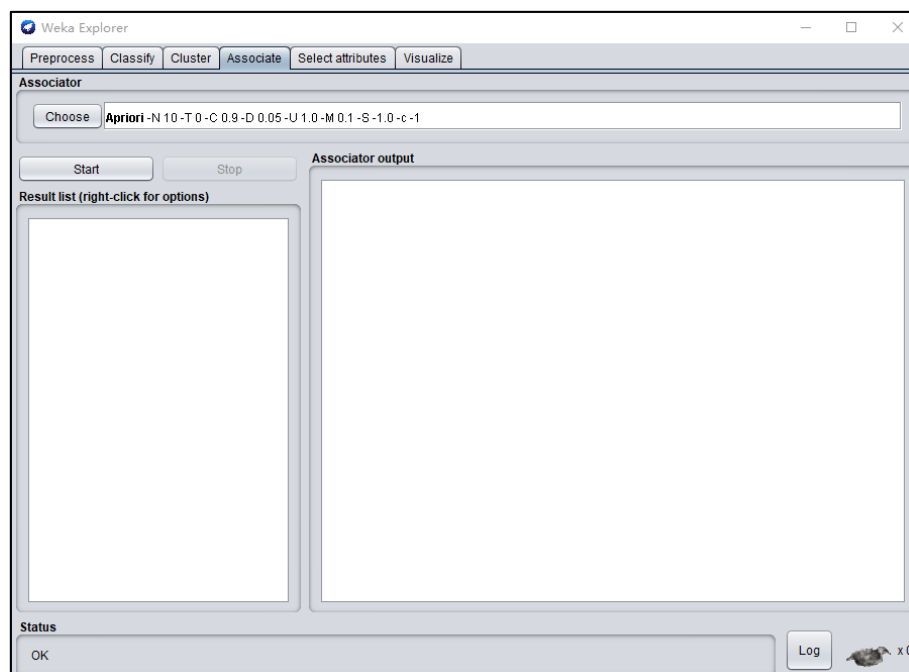


No.	1: department1	2: department2	3: department3	4: department4	5: department5	6: department6	7: department7	8: department8	9: department9	10: grocery misc	11: department11
1	t	t	t	t	t	t	t	t	t	t	t
2	t	t	t	t	t	t	t	t	t	t	t
3	t	t	t	t	t	t	t	t	t	t	t
4	t	t	t	t	t	t	t	t	t	t	t
5	t	t	t	t	t	t	t	t	t	t	t
6	t	t	t	t	t	t	t	t	t	t	t
7	t	t	t	t	t	t	t	t	t	t	t
8	t	t	t	t	t	t	t	t	t	t	t
9	t	t	t	t	t	t	t	t	t	t	t
10	t	t	t	t	t	t	t	t	t	t	t
11	t	t	t	t	t	t	t	t	t	t	t
12	t	t	t	t	t	t	t	t	t	t	t
13	t	t	t	t	t	t	t	t	t	t	t
14	t	t	t	t	t	t	t	t	t	t	t
15	t	t	t	t	t	t	t	t	t	t	t
16	t	t	t	t	t	t	t	t	t	t	t
17	t	t	t	t	t	t	t	t	t	t	t
18	t	t	t	t	t	t	t	t	t	t	t
19	t	t	t	t	t	t	t	t	t	t	t
20	t	t	t	t	t	t	t	t	t	t	t
21	t	t	t	t	t	t	t	t	t	t	t
22	t	t	t	t	t	t	t	t	t	t	t
23	t	t	t	t	t	t	t	t	t	t	t
24	t	t	t	t	t	t	t	t	t	t	t
25	t	t	t	t	t	t	t	t	t	t	t
26	t	t	t	t	t	t	t	t	t	t	t
27	t	t	t	t	t	t	t	t	t	t	t
28	t	t	t	t	t	t	t	t	t	t	t
29	t	t	t	t	t	t	t	t	t	t	t
30	t	t	t	t	t	t	t	t	t	t	t
31	t	t	t	t	t	t	t	t	t	t	t
32	t	t	t	t	t	t	t	t	t	t	t
33	t	t	t	t	t	t	t	t	t	t	t

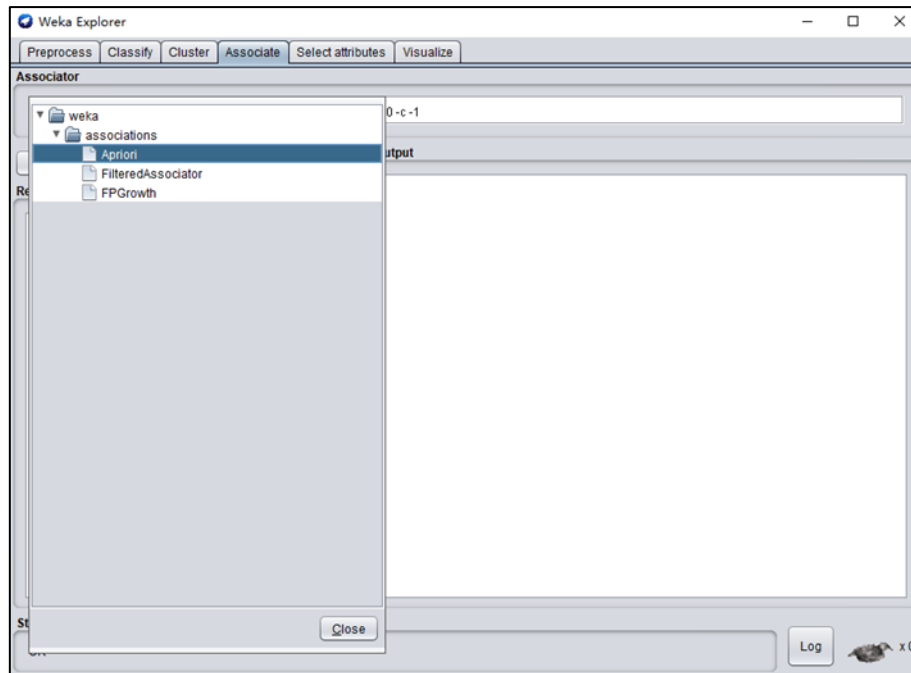
1.4 In the Each row is a transaction. Each column is an item or department number. A value 't' indicates the attribute appears in the transaction. We are interested in a market basket analysis by finding association rules among the columns/attributes.

2. Running in Weka

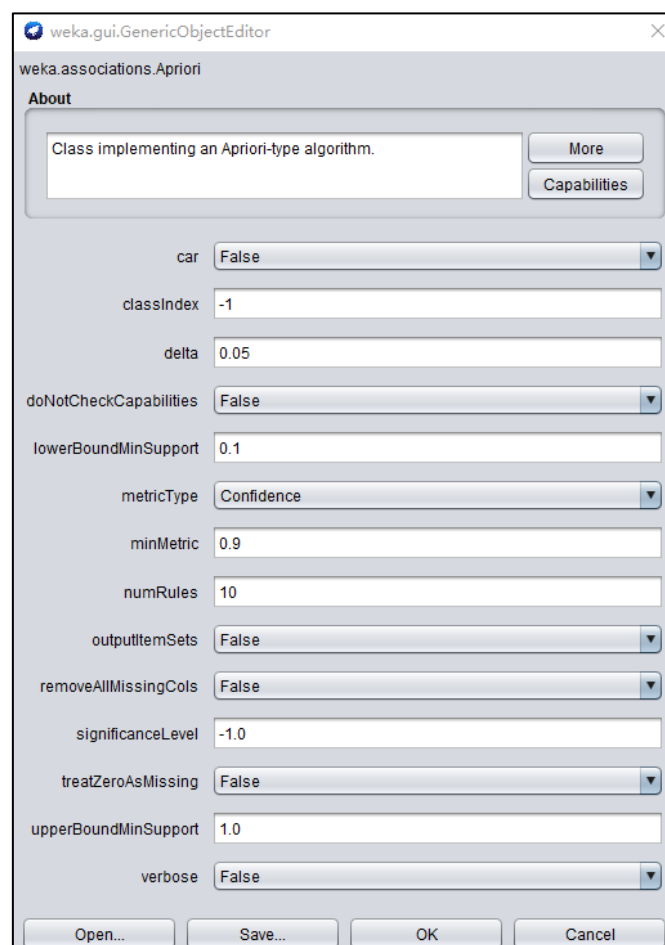
2.1 Click Associate Tab on top of the window.



2.2 Choose **weka->association->Apriori**



2.3 Click the field of **Apriori -N 10 -T xxxxxx** to open its property window



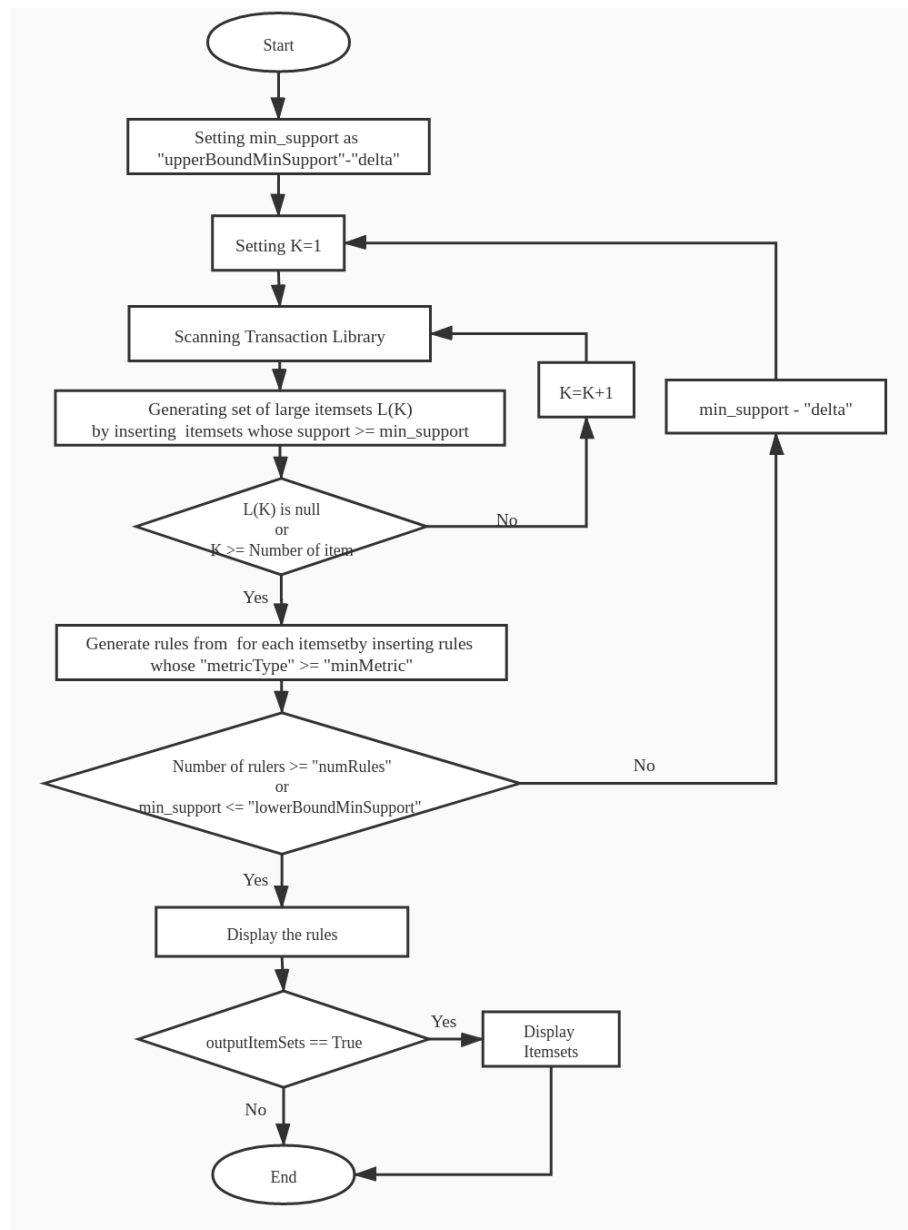
2.4 Explain the parameter, according to the Weka's manual or Helps

Parameter	Default	Description
delta	0.05	Weka iteratively decrease support by this parameter until min support is reached or required number of rules has been generated.
lowerBound MinSupport	0.1	Lower bound for minimum support which is the signal for Weka to stop iteration.
metricType	Confidence	<p>Set the type of metric by which to rank rules. for rule <i>premise</i> \rightarrow <i>consequence</i></p> <p>1. Confidence Confidence is the proportion of the examples covered by the premise that are also covered by the consequence (Class association rules can only be mined using confidence). The formula is:</p> $\frac{P(\text{premise}, \text{consequence})}{P(\text{premise})}$ <p>2. Lift Lift is a measure of the importance of the association that is independent of support. The formula is:</p> $\frac{P(\text{premise}, \text{consequence})}{P(\text{premise})P(\text{consequence})}$ <p>3. Leverage Leverage value greater than 0 is desirable. The formula is:</p> $\frac{P(\text{premise}, \text{consequence})}{-P(\text{premise})P(\text{consequence})}$ <p>4. Conviction Conviction is a measure of departure from independence. The formula is:</p> $\frac{P(\text{premise})P(! \text{consequence})}{P(\text{premise}, ! \text{consequence})}$
minMetric	0.9	Minimum metric score. Consider only rules with scores higher than this value.
numRules	10	Determines number of rules to find.
outputItemSets	True	Determine whether itemsets are output.
upperBound MinSupport	1.0	Upper bound for minimum support. Start iteratively decreasing minimum support from this value.

2.5 Explain how Weka finds association rules using an Apriori-type algorithm

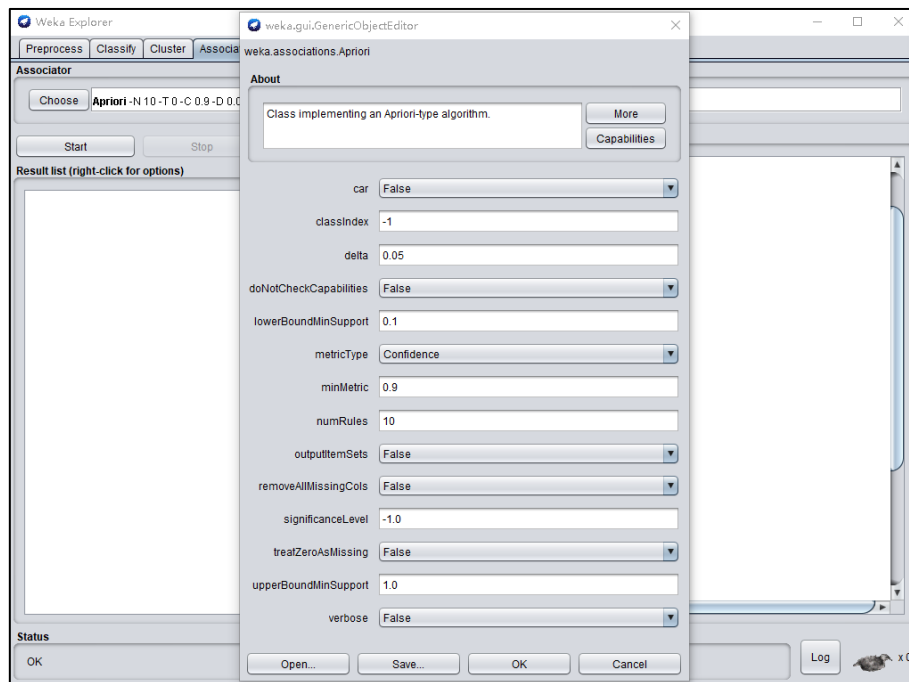
According to the algorithm, Weka Iteratively decreases the minimum support by **delta** (default 0.05 = 5%) starting with **upperBoundMinSupport** (default 1.0 = 100%). During the period, several sets which are made of itemsets whose support is higher than current minimum support, and rules are generated from those sets. Generated rules are ranked by **metricType** (default Confidence) and whose score lower than **minMetric** (default 0.9 for Confidence) are removed. When the **lowerBoundMinSupport** (default 0.1 = 10%) is reached or required number of rules – **numRules** (default value 10) have been generated, the Weka stops decreasing the minimum support. Then, the result is displayed and all the frequent itemsets found will be shown in the result if **outputItemSets** is set as True.

The flowchart of Apriori algorithm is:

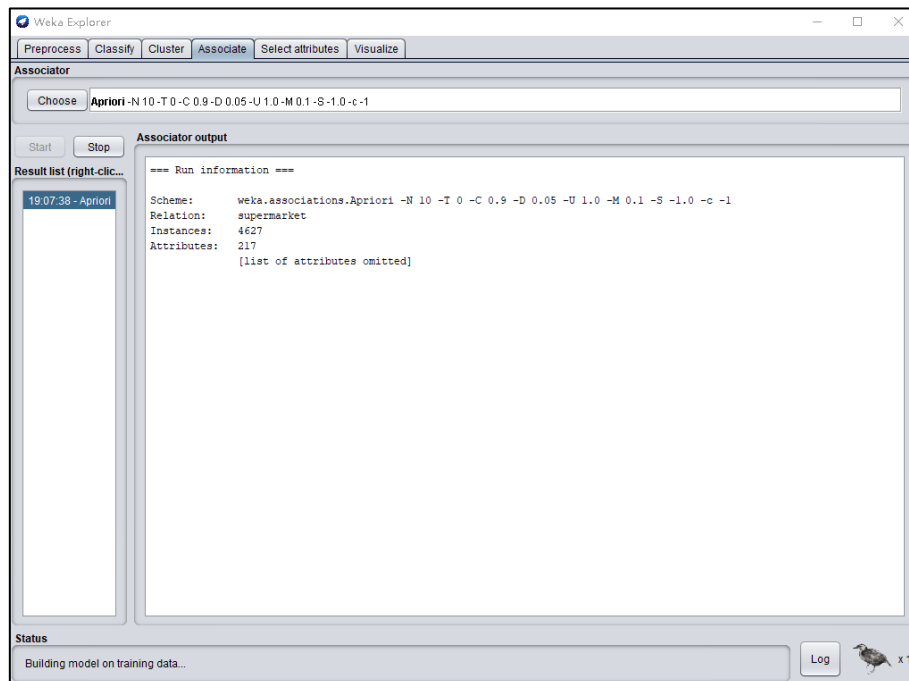


2.6 Set up the values of the parameters

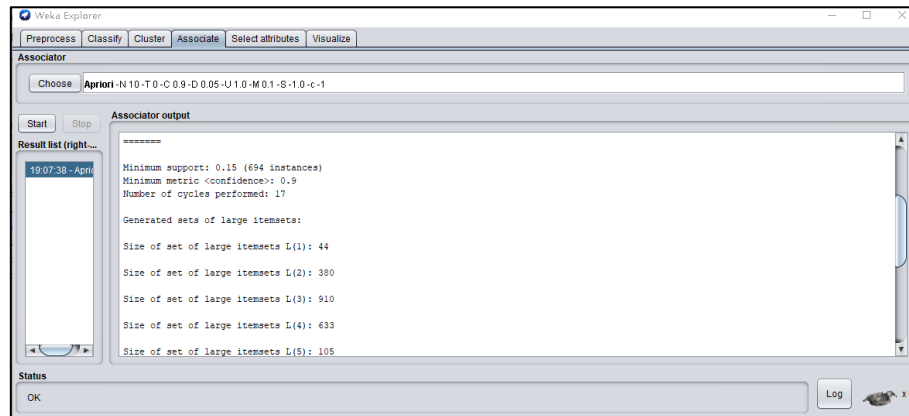
I use the default values as parameters.



2.7 Run the algorithm on Weka



2.8 Result



=== 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)

3. Analyzing

Comment the rules and discuss how their support and confidence values change and are related. Discuss the main findings of your experiments.

According to the above parameters setting, there are two main metrics to generate the rules:

Assume T dataset has n transactions, itemsets A and B .	
① Support	$Support(A \rightarrow B) = \frac{(A \cup B).count}{n}$
② Confidence	$Confidenc(A \rightarrow B) = \frac{(A \cup B).count}{A.count}$
③ The relation between confidence and support for rule $A \rightarrow B$ is that	$confidence(A \rightarrow B) = support(A \rightarrow B) / support(A)$

3.1 Comment the rules and discuss how their support and confidence values change and are related.

I comment the rules and focus on their support and confidence. The result shows 10 most relevant rules determined by the default argument **numRules** (10).

3.1.1 Rule1

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)

- Number of transactions containing $\{biscuits, frozen foods, fruit, total=high\}$ is **788**.
- Number of transactions containing $\{biscuits, frozen foods, fruit, total=high, bread and cake\}$ is **723**.
- Number of transactions is **4627**
- Calculate

① Support($\{biscuits, frozen foods, fruit, total=high\} \rightarrow \{bread and cake\}$):
$\frac{723}{4627} \approx 15.6\% > 10\%$
② Confidence($\{biscuits, frozen foods, fruit, total=high\} \rightarrow \{bread and cake\}$):
$\frac{723}{788} \approx 91.53\% > 90\%$

- Value change and related

- | |
|--|
| <p>① As the Number of transactions containing $\{biscuits, frozen foods, fruit, total=high, bread and cake\}$ increases, the value of confidence and support increases.</p> <p>② As the Number of transactions containing $\{biscuits, frozen foods, fruit, total=high\}$ increases, the value of confidence decreases and the value of support is not change.</p> |
|--|

- According to this rule, it shows that

There is a 92% chance for a customer to buy **bread and cake**, if he buys **biscuits, frozen foods** and **fruit** with a **high total price**.

3.1.2 Rule2

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)

- Number of transactions containing {baking needs, biscuits, fruit, total=high} is **760**.
- Number of transactions containing {baking needs, biscuits, fruit, total=high, breadandcake} is **696**.
- Number of transactions is **4627**
- Calculate

① Support({baking needs, biscuits, total=high} → { bread and cake }):

$$\frac{696}{4627} \approx 15.04\% > 10\%$$

② Confidence({baking needs, biscuits, fruit, total=high, breadandcake} → { bread and cake }):

$$\frac{696}{760} \approx 91.58\% > 90\%$$

- Value change and related

① As the **Number of transactions containing {baking needs, biscuits, fruit, total=high, bread and cake}** increases, the value of confidence and support increases.

② As the **Number of transactions containing {baking needs, biscuits, fruit, total=high}** increases, the value of confidence decreases and the value of support is not change.

- According to this rule, it shows that

There is a 92% chance for a customer to buy **bread and cake**, if he buys **baking needs, fruit** and **biscuits** with a **high total price**.

3.1.3 Rule3

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)

- Number of transactions containing {baking needs, frozen foods, fruit, total=high} is **770**.
- Number of transactions containing {baking needs, frozen foods, fruit, total=high, bread and cake} is **705**.
- Number of transactions is **4627**
- Calculate

① Support({baking needs, frozen foods, fruit, total=high} → { bread and cake }):

$$\frac{705}{4627} \approx 15.24\% > 10\%$$

② Confidence({baking needs, frozen foods, fruit, total=high} → { bread and cake }):

$$\frac{705}{770} \approx 91.56\% > 90\%$$

- Value change and related

- ① As the **Number of transactions containing {baking needs, frozen foods, fruit, total=high, bread and cake}** increases, the value of confidence and support increases.
- ② As the **Number of transactions containing {baking needs, frozen foods, fruit, total=high}** increases, the value of confidence decreases and the value of support is not change.

- According to this rule, it shows that

There is a 92% chance for a customer to buy **bread and cake**, if he buys **baking needs, frozen foods, fruit, total=high**

3.1.4 Rule4

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)

- Number of transactions containing {biscuits, fruit, vegetables, total=high} is **815**.
- Number of transactions containing {biscuits, fruit, vegetables, total=high, bread and cake} is **746**.
- Number of transactions is **4627**
- Calculate

- ① Support({biscuits, fruit, vegetables, total=high} → {bread and cake}):

$$\frac{746}{4627} \approx 16.12\% > 10\%$$

- ② Confidence({biscuits, fruit, vegetables, total=high} → {bread and cake}):

$$\frac{746}{815} \approx 91.53\% > 90\%$$

The confidence is also shown in the output **<conf:(0.92)>**

- Value change and related

- ① As the **Number of transactions containing {biscuits, fruit, vegetables, total=high, break and cake}** increases, the value of confidence and support increases.
- ② As the **Number of transactions containing {biscuits, fruit, vegetables, total=high}** increases, the value of confidence decreases and the value of support is not change.

- According to this rule, it shows that

There is a 92% chance for a customer to buy **bread and cake**, if he buys **biscuits, fruit and vegetable** with a **high total price**.

3.1.5 Rule5

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)

- Number of transactions containing {party snack foods, fruit, total=high} is **854**.
- Number of transactions containing {party snack foods, fruit, total=high., bread and cake} is **779**.
- Number of transactions is **4627**

- Calculate

① Support(*{party snack foods, fruit, total=high}* → *{bread and cake}*):

$$\frac{779}{4627} \approx 16.8\% > 10\%$$

② Confidence(*{party snack foods, fruit, total=high}* → *{bread and cake}*):

$$\frac{779}{854} \approx 91.21\% > 90\%$$

- Value change and related

① As the **Number of transactions containing** *{party snack foods, fruit, total=high, bread and cake}* increases, the value of confidence and support increases.

② As the **Number of transactions containing** *{party snack foods, fruit, total=high}* increases, the value of confidence decreases and the value of support is not change.

- According to this rule, it shows that

There is a 91% chance for a customer to buy **bread and cake**, if he buys **party snack foods** and **fruit** with a **high total price**.

3.1.6 Rule6

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)

- Number of transactions containing *{biscuits, frozen foods, vegetables, total=high}* is **797**.
- Number of transactions containing *{biscuits, frozen foods, vegetables, total=high, bread and cake}* is **725**.
- Number of transactions is **4627**
- Calculate

① Support(*{biscuits, frozen foods, vegetables, total=high}* → *{bread and cake}*):

$$\frac{725}{4627} \approx 15.67\% > 10\%$$

② Confidence(*{biscuits, frozen foods, vegetables, total=high}* → *{bread and cake}*):

$$\frac{725}{797} \approx 90.97\% > 90\%$$

- Value change and related

① As the **Number of transactions containing** *{biscuits, frozen foods, vegetables, total=high, bread and cake}* increases, the value of confidence and support increases.

② As the **Number of transactions containing** *{biscuits, frozen foods, vegetables, total=high}* increases, the value of confidence decreases and the value of support is not change.

- According to this rule, it shows that

There is a 91% chance for a customer to buy **bread and cake**, if he buys **biscuits, frozen foods** and **vegetables** with a **high total price**.

3.1.7 Rule7

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)

- Number of transactions containing { *baking needs, biscuits, vegetables, total=high* } is **797**.
- Number of transactions containing { *baking needs, biscuits, vegetables, total=high, bread and cake* } is **725**.
- Number of transactions is **4627**
- Calculate

① Support({*baking needs, biscuits, vegetables, total=high*} → {*bread and cake*}):

$$\frac{701}{4627} \approx 15.15\% > 10\%$$

② Confidence({*baking needs, biscuits, vegetables, total=high*} → {*bread and cake*}):

$$\frac{701}{772} \approx 90.80\% > 90\%$$

- Values relationship

① As the **Number of transactions containing {*baking needs, biscuits, vegetables, total=high, bread and cake*}** increases, the value of confidence and support increases.

② As the **Number of transactions containing {*baking needs, biscuits, vegetables, total=high*}** increases, the value of confidence decreases and the value of support is not change.

- According to this rule, it shows that

There is a 91% chance for a customer to buy **bread and cake**, if he buys **baking needs, biscuits** and **vegetables** with a **high total price**.

3.1.8 Rule8

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

- Number of transactions containing { *biscuits, fruit, total=high* } is **797**.
- Number of transactions containing { *biscuits, fruit, total=high., bread and cake* } is **725**.
- Number of transactions is **4627**
- Calculate

① Support({*biscuits, fruit, total=high*} → {*bread and cake*}):

$$\frac{866}{4627} \approx 18.71\% > 10\%$$

② Confidence({*biscuits, fruit, total=high*} → {*bread and cake*}):

$$\frac{866}{954} \approx 90.78\% > 90\%$$

- Value change and related

- ① As the **Number of transactions containing {biscuits, fruit, total=high, bread and cake}** increases, the value of confidence and support increases.
- ② As the **Number of transactions containing {biscuits, fruit, total=high}** increases, the value of confidence decreases and the value of support is not change.

- According to this rule, it shows that

There is a 91% chance for a customer to buy **bread and cake**, if he buys **biscuits** and **fruit** with a **high total price**.

3.1.9 Rule9

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)

- Number of transactions containing {fruit, frozen foods, vegetables, total=high} is **797**.
- Number of transactions containing {fruit, frozen foods, vegetables, total=high, bread and cake} is **725**.
- Number of transactions is **4627**
- Calculate

- ① Support({fruit, frozen foods, vegetables, total=high} → {bread and cake}):

$$\frac{757}{4627} \approx 16.36\% > 10\%$$

- ② Confidence({fruit, frozen foods, vegetables, total=high} → {bread and cake}):

$$\frac{757}{834} \approx 90.77\% > 90\%$$

- Value change and related

- ① As the **Number of transactions containing {fruit, frozen foods, vegetables, total=high, bread and cake}** increases, the value of confidence and support increases.
- ② As the **Number of transactions containing {fruit, frozen foods, vegetables, total=high}** increases, the value of confidence decreases and the value of support is not change.

- According to this rule, it shows that

There is a 91% chance for a customer to buy **bread and cake**, if he buys **fruit, frozen foods** and **vegetables** with a **high total price**.

3.1.10 Rule10

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)

- Number of transactions containing {frozen foods, fruit, total=high} is **969**.
- Number of transactions containing {frozen foods, fruit, total=high, bread and cake} is **877**.
- Number of transactions is **4627**

- Calculate

① Support(*{frozen foods, fruit, total=high}*→*{bread and cake}*):

$$\frac{877}{4627} \approx 18.95\% > 10\%$$

② Confidence(*{frozen foods, fruit, total=high}*→*{bread and cake}*):

$$\frac{877}{969} \approx 90.51\% > 90\%$$

- Value change and related

① As the **Number of transactions containing {frozen foods, fruit, total=high, bread and cake}** increases, the value of confidence and support increases.

② As the **Number of transactions containing {frozen foods, fruit, total=high}** increases, the value of confidence decreases and the value of support is not change.

- According to this rule, it shows that

There is a 91% chance for a customer to buy **bread and cake**, if he buys **frozen foods** and **fruit** with a **high total price**.

3.2 Main findings

Rule	Support	Confidence
<i>{biscuits, frozen foods, fruit, total=high}</i> → <i>{bread and cake}</i>	15.6%	91.75%
<i>{baking needs, biscuits, total=high}</i> → <i>{ bread and cake }</i>	15.04%	91.58%
<i>{baking needs, frozen foods, fruit, total=high}</i> → <i>{bread and cake}</i>	15.24%	91.56%
<i>{biscuits, fruit, vegetables, total=high}</i> → <i>{bread and cake}</i>	16.12%	91.53%
<i>{party snack foods, fruit, total=high}</i> → <i>{bread and cake}</i>	16.84%	91.21%
<i>{biscuits, frozen foods, vegetables, total=high}</i> → <i>{bread and cake}</i>	15.67	90.97%
<i>{baking needs, biscuits, vegetables, total=high }</i> → <i>{bread and cake}</i>	15.15%	90.80%
<i>{biscuits, fruit, total=high }</i> → <i>{bread and cake}</i>	18.71%	90.78%
<i>{fruit, frozen foods, vegetables, total=high}</i> → <i>{bread and cake}</i>	16.36%	90.77%
<i>{frozen foods, fruit, total=high}</i> → <i>{bread and cake}</i>	18.95%	90.51%

According to the result, the confidence values decrease from rule 1 to rule 10. The support values increase from rule 1 to rule 10 on the whole. The support and confidence values of these rules are similar. All the confidence values of the shown rules are more than 90% and the support values of the shown rules are more than 15%.

In the result, all shown rules have a consequence---“bread and cake”, which means there is a strong correlation with buying bread and cake with buying other items. Furthermore, “Biscuits” an “frozen foods” appear in many of the presented rules, which means they play a more important role to indicate the customers who are likely to buy bread and cake. The result is a good reference to help the manager of supermarket to develop selling strategies by the customer’s purchasing behavior.

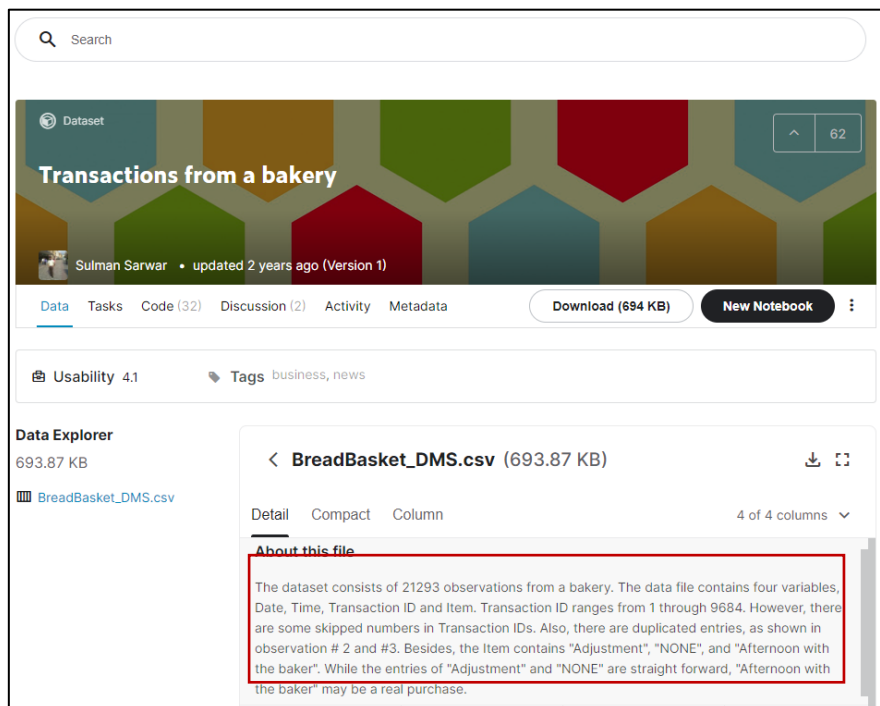
Task 2. Pre-processing Dirty Data for Association Rule Mining

In this task you will work on a Kaggle data set to prepare it for a market analysis by mining association rules. The data set is available at Kaggle.com: <https://www.kaggle.com/sulmansarwar/transactions-from-a-bakery>. In this task, you need to transform the sales data to transactional data for association rule mining.

1. Data preparation

1.1 Download data

Download the CSV data from either the above Kaggle link or the course shell as **BreadBasket_DMS.csv**.



Transactions from a bakery
Sulman Sarwar • updated 2 years ago (Version 1)

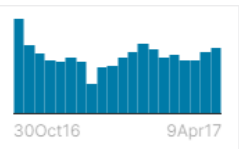
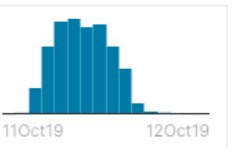
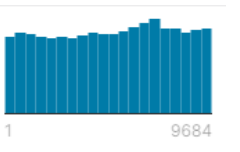
Data Explorer
693.87 KB
BreadBasket_DMS.csv

BreadBasket_DMS.csv (693.87 KB)

About this file

The dataset consists of 21293 observations from a bakery. The data file contains four variables, Date, Time, Transaction ID and Item. Transaction ID ranges from 1 through 9684. However, there are some skipped numbers in Transaction IDs. Also, there are duplicated entries, as shown in observation # 2 and #3. Besides, the Item contains "Adjustment", "NONE", and "Afternoon with the baker". While the entries of "Adjustment" and "NONE" are straight forward, "Afternoon with the baker" may be a real purchase.

1.2 Data Overview

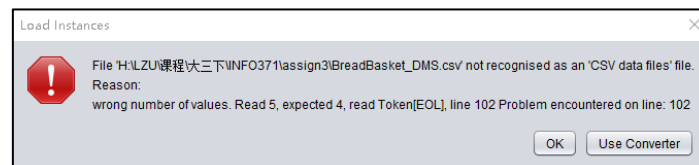
Date	Time	Transaction	Item
			<div>Coffee 26%</div> <div>Bread 16%</div> <div>Other (12497) 59%</div>
2016/10/30	9:58:11	1	Bread
2016/10/30	10:05:34	2	Scandinavian
2016/10/30	10:05:34	2	Scandinavian
2016/10/30	10:07:57	3	Hot chocolate
2016/10/30	10:07:57	3	Jam
.....
2017/4/9	15:04:24	9684	Smoothies

2. Clean up the values and Convert the data

Try to open the original **BreadBasket_DMS.csv** in Weka. Weka should complain and show error message. In this task, you are asked to clean up the data and convert it to the format that is ready for association rule mining.

Task 2.1 Clean up the values

According to the error message generated by Weka, I found the line **102** which caused the 1st problem in the original **BreadBasket_DMS.csv**.



In the line **102**, the problem was caused by the “” in “*Ella's Kitchen Pouches*”. Therefore, it is necessary to replace “” with question mark “?”.

	Date	Time	Transaction	Item
2	2016/10/30	9:58:11	1	Bread
.....				
101	2016/10/30	12:09:04	46	Coffee
102	2016/10/30	12:15:29	47	Ella's Kitchen Pouches
103	2016/10/30	12:15:29	47	Juice
.....				
21294	2017/4/9	15:04:24	9684	Smoothies

According to the above analyze, I used python script with “pandas” package to remove all the “” with question mark “?” in “**item**” attribute.

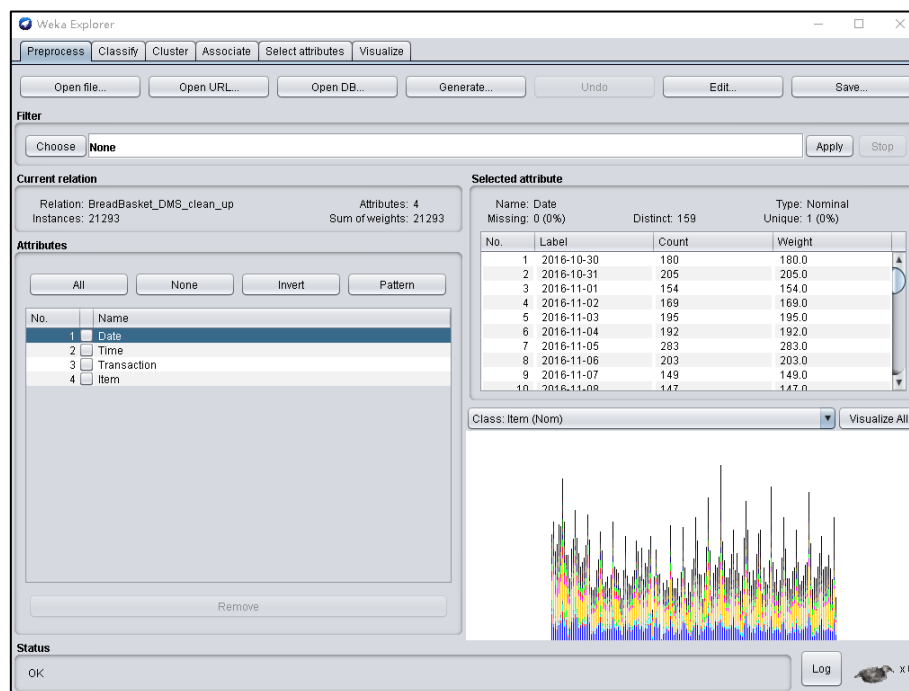
2.1.1 Python Code

```
import pandas as pd
df = pd.read_csv("BreadBasket_DMS.csv")
df["Item"] = df.Item.apply(lambda x: x.replace("'", "?"))
df.to_csv("BreadBasket_DMS_clean_up.csv")
```

2.1.2 Result

	Date	Time	Transaction	Item
2	2016/10/30	9:58:11	1	Bread
.....				
101	2016/10/30	12:09:04	46	Coffee
102	2016/10/30	12:15:29	47	Ella?s Kitchen Pouches
103	2016/10/30	12:15:29	47	Juice
.....				
21294	2017/4/9	15:04:24	9684	Smoothies

After cleaning up the values of Item column, Weka can open the csv file.



Task 2.2 Clean up the values

In this task, I wrote a python script to convert the two columns ‘Transaction’ and ‘Item’ in the current data set to a format such that each row is a transaction identified by its id and the columns are the items. Then, I named the new data file as **BreadBasket_DMS_pivot.csv**. The values of the new data set represent the counts of an item (column) in a transaction (row).

2.2.1 Python Code

```
import pandas as pd
df = pd.read_csv("BreadBasket_DMS_clean_up.csv")
df = df.drop(columns=["Date", "Time"])
df = pd.crosstab(index=df['Transaction'], columns=df['Item'])
df.to_csv("BreadBasket_DMS_pivot.csv", index=True)
```

2.2.2 Result

Transaction	Adjustment	Honey	Hot chocolate	Jam	Victorian Sponge
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	1	1	0	0
.....							
9684	0	0	0	0	0	0	0

Task 2.3 Remove “NONE” item

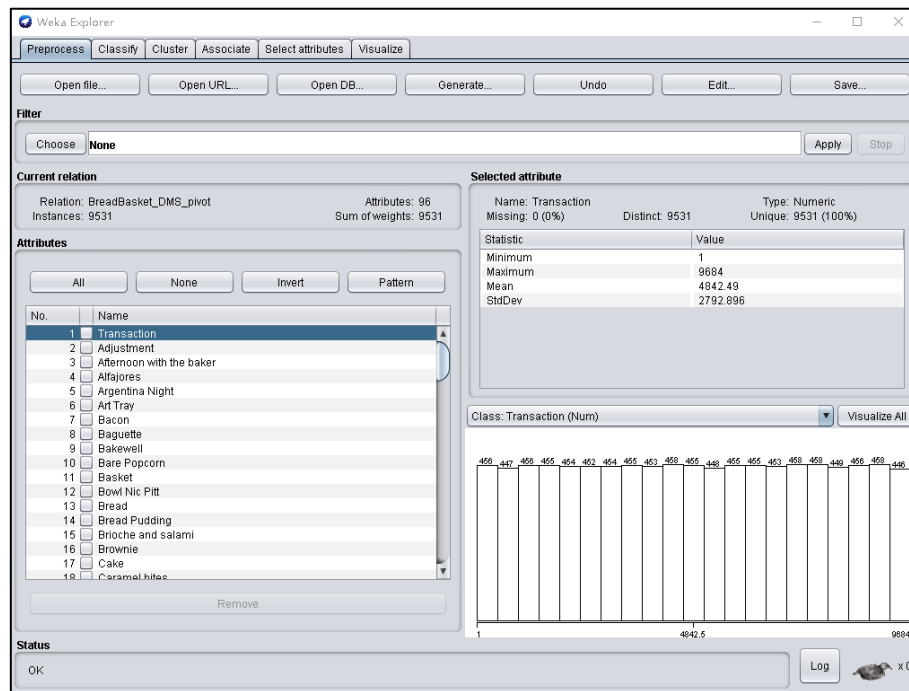
According to the description of this data set, “NONE” represents the item with missing name which may interfere the results. Therefore, I removed it with python script.

```
df = df.drop(columns=["NONE"])
```

Task 3. Mining Association Rules for the Bread Basket Data

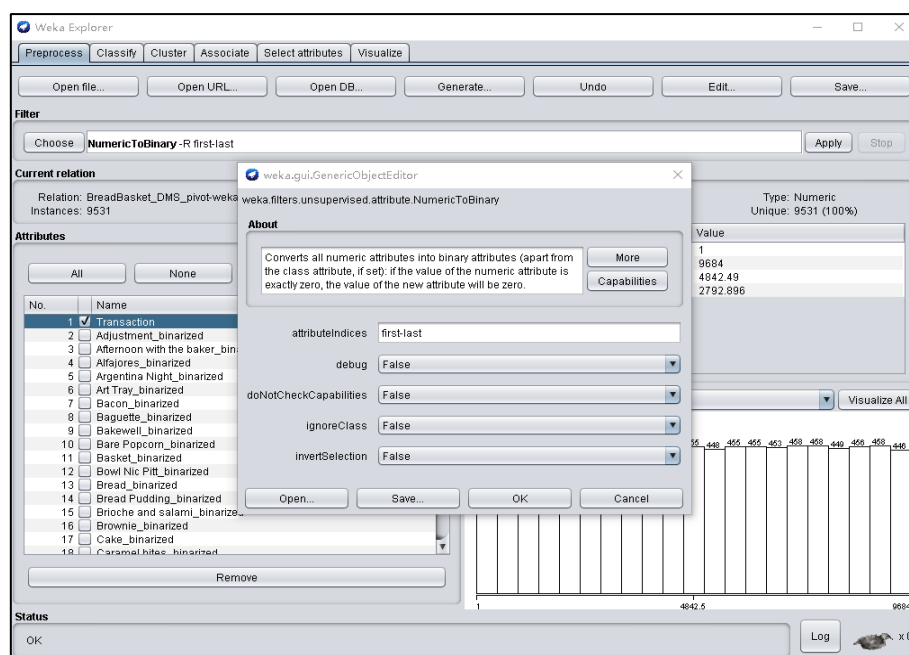
1. Data Loading

Open **BreadBasket_DMS_pivot.csv** in weka and select Transaction (if there is) as the class



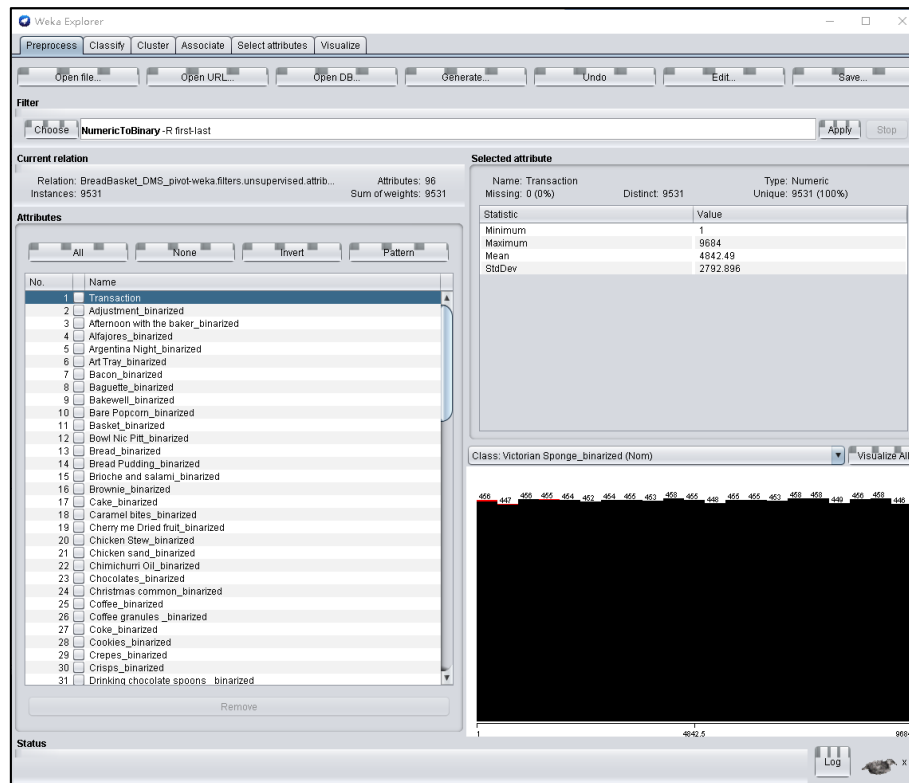
2. Set for Data Preprocessing

On Filter, choose **weka->filters->unsupervised->attribute->NumericToBinary -R first-last**



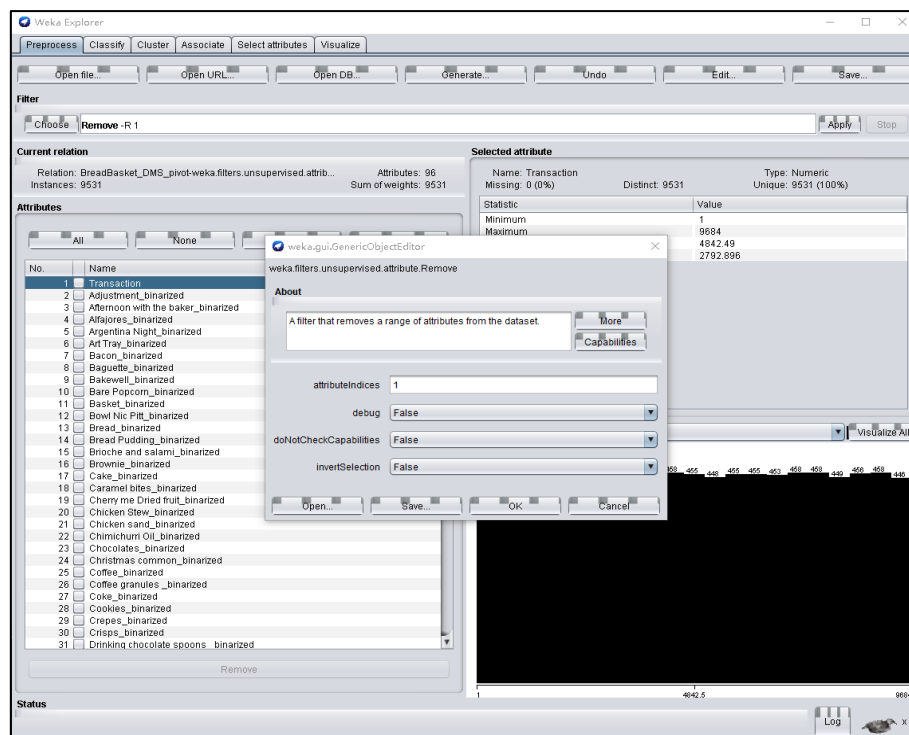
3. Apply Data Preprocessing

Click Apply



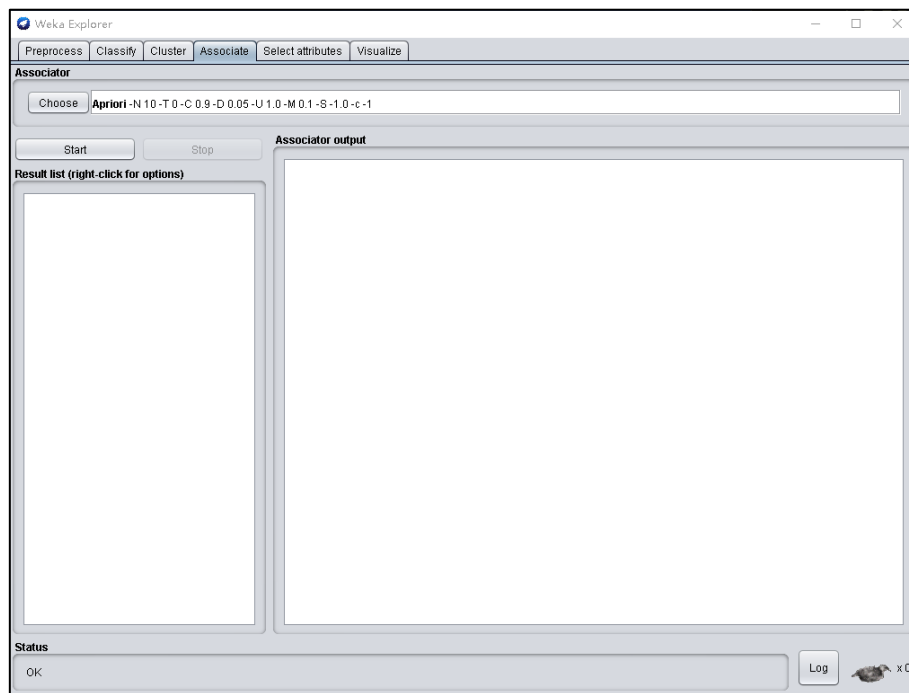
4. Remove Transaction Attribute

Check the Transaction attribute (if there is) and remove it



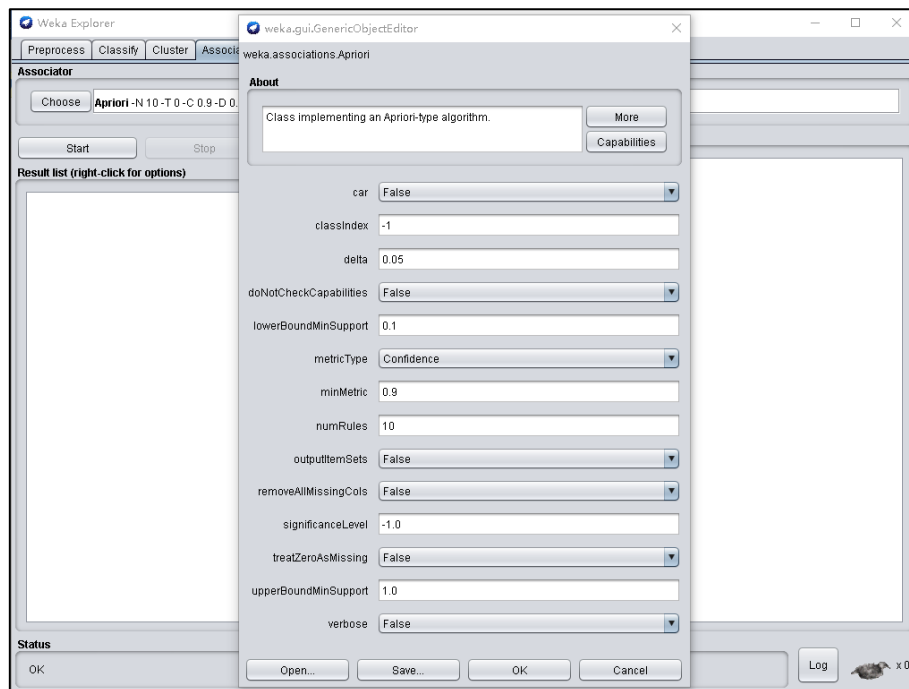
5. Ready for Associate operation

Click **Associate** Tab on top of the window.



6. Set for Apriori algorithm

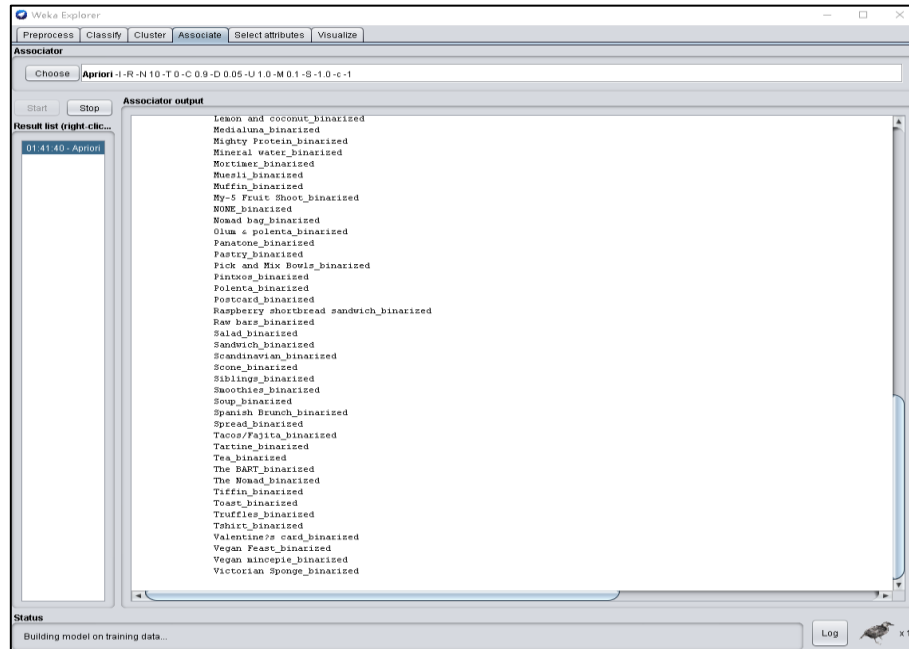
Choose **weka->association->Apriori**.



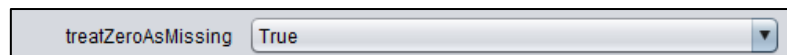
7. Runing by using the default settings

Click start by using the default settings.

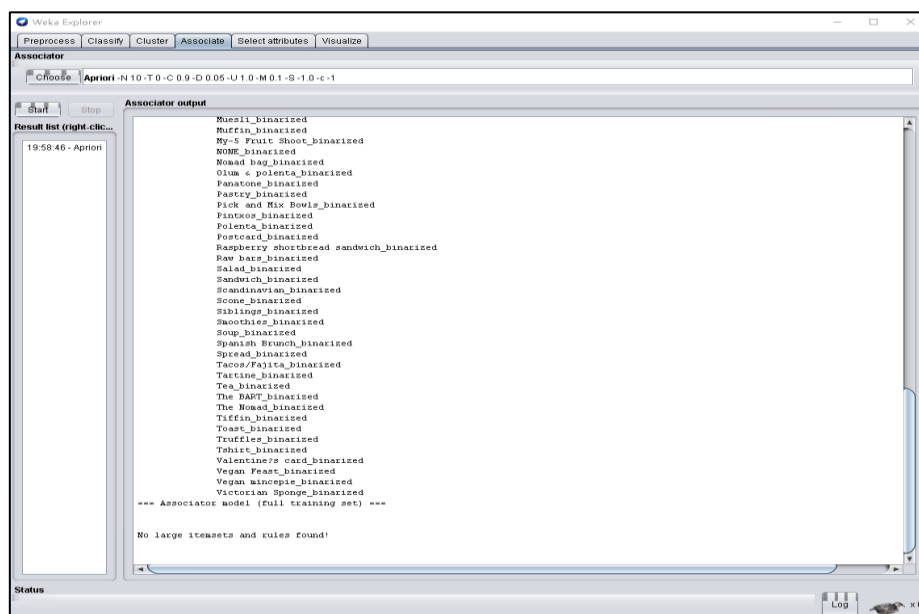
In the 1st time, the Weka was stuck, when building model on training data. This is because the large amount of 0 values in new data set extremely increased the running time of building model.



Therefore, I set **treatZeroAsMissing** as True which converts 0 values into missing value which decrease the running time.



After changing the parameter setting, the Weka ran successfully.



8. Copy the result to your report. Did you find any rules? Why?

```
=== Run information ===

Scheme:      weka.associations.Apriori -R -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -Z -c -1
Relation:      BreadBasket_DMS_pivot-weka.filters.unsupervised.attribute.Remove-R1-
weka.filters.unsupervised.attribute.NumericToBinary-Rfirst-last-
weka.filters.unsupervised.attribute.NumericToBinary-Rfirst-last
Instances:    9531
Attributes:    95
               Adjustment_binarized
               Afternoon with the baker_binarized
               Alfajores_binarized
               Argentina Night_binarized
               Art Tray_binarized
               Bacon_binarized
               Baguette_binarized
               Bakewell_binarized
               Bare Popcorn_binarized
               Basket_binarized
               Bowl Nic Pitt_binarized
               Bread_binarized
               Bread Pudding_binarized
               Brioche and salami_binarized
               Brownie_binarized
               Cake_binarized
               .....
               Tiffin_binarized
               Toast_binarized
               Truffles_binarized
               Tshirt_binarized
               Valentine?s card_binarized
               Vegan Feast_binarized
               Vegan mincepie_binarized
               Victorian Sponge_binarized
=== Associator model (full training set) ===

No large itemsets and rules found!
```

There are no rules found in the results. The reason is that until the **lowerBoundMinSupport** is reached or required number of rules – **numRules** has been generated, there are even no frequent itemsets, which are the itemsets for which support value (fraction of transactions containing the itemset) is above the minimum support. The number of items sold in a transaction is too small compared to the total number of transactions and items. Therefore, it is impossible to generate strong rule from itemsets under default setting.

9. Re-run the A-priori with changed settings. Test different values on the confidence constraints until you find some rules. Copy the results to your report. Discuss with what values you can find rules and why. Continue to discuss any interesting findings of your investigation.

9.1 Re-run and Testing

I re-ran the A-priori with changed settings in two major stages:

- The 1st stage is set the **lowerBoundMinSupport** as some values in a rough range.
- The 2nd stage is set the **lowerBoundMinSupport** as some values in a more accurate range generated from the 1st stage.

During the testing, other parameters were setting as default. To be specific, the **minMetric** was set as 0.8, which can give a better observation for the rules finding. This is because the A-priori will filter the rules whose confidence is lower than the **minMetric**.

weka.gui.GenericObjectEditor

weka.associations.Apriori

About

Class implementing an Apriori-type algorithm. More Capabilities

car False

classIndex -1

delta 0.05

doNotCheckCapabilities False

lowerBoundMinSupport 0.1

metricType Confidence

minMetric 0.6

numRules 10

outputItemSets False

removeAllMissingCols False

significanceLevel -1.0

treatZeroAsMissing True

upperBoundMinSupport 1.0

verbose False

Open... Save... OK Cancel

9.1.1 1st stage

In this stage, I set the **lowerBoundMinSupport** as 0.1, 0.01, 0.001 and 0.0001

lowerBound MinSupport	Result
0.1	No large itemsets and rules found!
0.01	<p>Minimum support: 0.01 (95 instances) Minimum metric <confidence>: 0.8 Number of cycles performed: 20</p> <p>Generated sets of large itemsets:</p> <p>Size of set of large itemsets L(1): 30 Size of set of large itemsets L(2): 28 Size of set of large itemsets L(3): 3</p> <p>Best rules found:</p>
0.001	<p>Minimum support: 0 (10 instances) Minimum metric <confidence>: 0.8 Number of cycles performed: 20</p> <p>Generated sets of large itemsets:</p> <p>Size of set of large itemsets L(1): 57 Size of set of large itemsets L(2): 243 Size of set of large itemsets L(3): 167 Size of set of large itemsets L(4): 4</p> <p>Best rules found:</p> <ol style="list-style-type: none"> 1. <conf:(0.88)> lift:(1.84) lev:(0) [6] conv:(2.8) 2. <conf:(0.87)> lift:(1.82) lev:(0) [5] conv:(2.62) 3. <conf:(0.86)> lift:(1.8) lev:(0) [5] conv:(2.45) 4. <conf:(0.83)> lift:(1.75) lev:(0) [6] conv:(2.36) 5. <conf:(0.83)> lift:(1.75) lev:(0) [4] conv:(2.1) 6. <conf:(0.82)> lift:(1.72) lev:(0) [12] conv:(2.49) 7. <conf:(0.81)> lift:(1.7) lev:(0) [21] conv:(2.54)
0.0001	<p>Minimum support: 0 (1 instances) Minimum metric <confidence>: 0.8 Number of cycles performed: 20</p> <p>Generated sets of large itemsets:</p> <p>Size of set of large itemsets L(1): 94 Size of set of large itemsets L(2): 1047 Size of set of large itemsets L(3): 3152 Size of set of large itemsets L(4): 3493 Size of set of large itemsets L(5): 2031 Size of set of large itemsets L(6): 924</p>

Size of set of large itemsets L(7): 349
Size of set of large itemsets L(8): 94
Size of set of large itemsets L(9): 15
Size of set of large itemsets L(10): 1

Best rules found:

1. <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.67)
2. <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.15)
3. <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.15)
4. <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)
5. <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)
6. <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)
7. <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.1)
8. <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.1)
9. <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.1)
10. <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.1)

9.1.2 2nd stage

According to the result of 1st stage, I decreased the **lowerBoundMinSupport** from 0.001 to 0.0005, 0.0001 pre time. In order to get more trustable result, I set the **minMetric** as 0.9.

lowerBound MinSupport	Result
0.0009	<p>Minimum support: 0 (9 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 20</p> <p>Generated sets of large itemsets:</p> <p>Size of set of large itemsets L(1): 59 Size of set of large itemsets L(2): 260 Size of set of large itemsets L(3): 188 Size of set of large itemsets L(4): 8</p> <p>Best rules found:</p>

0.0008	<p>Minimum support: 0 (8 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 20</p> <p>Generated sets of large itemsets:</p> <p>Size of set of large itemsets L(1): 62 Size of set of large itemsets L(2): 279 Size of set of large itemsets L(3): 218 Size of set of large itemsets L(4): 15</p> <p>Best rules found:</p>
--------	---

0.0007	<p>Minimum support: 0 (7 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 20</p> <p>Generated sets of large itemsets:</p> <p>Size of set of large itemsets L(1): 66 Size of set of large itemsets L(2): 310 Size of set of large itemsets L(3): 261 Size of set of large itemsets L(4): 22</p> <p>Best rules found:</p> <p>1. <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.67)</p>
0.0006	<p>Minimum support: 0 (6 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 20</p> <p>Generated sets of large itemsets:</p> <p>Size of set of large itemsets L(1): 72 Size of set of large itemsets L(2): 348 Size of set of large itemsets L(3): 331 Size of set of large itemsets L(4): 35</p> <p>Best rules found:</p> <p>1. <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.67) 2. <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.15) 3. <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.15)</p>
0.0005	<p>Minimum support: 0 (5 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 20</p> <p>Generated sets of large itemsets:</p> <p>Size of set of large itemsets L(1): 75 Size of set of large itemsets L(2): 387 Size of set of large itemsets L(3): 400 Size of set of large itemsets L(4): 65</p> <p>Best rules found:</p> <p>1.<conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.67) 2. <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.15) 3. <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.15) 4. <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62) 5. <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62) 6. <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)</p>

According to the result of re-running, I chose the **0.0005** as the **lowerBoundMinSupport** and **0.9** as the **minMetric** which can guarantee quality and quantity of generating rule.

9.3 Discuss rules

```
=== Run information ===

Scheme:      weka.associations.Apriori -R -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 5.0E-4 -S -1.0 -Z -c -1
Relation:      BreadBasket_DMS_pivot-weka.filters.unsupervised.attribute.Remove-R1-
weka.filters.unsupervised.attribute.NumericToBinary-Rfirst-last-
weka.filters.unsupervised.attribute.NumericToBinary-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R63
Instances:      9531
Attributes:      94
                Adjustment_binarized
                Afternoon with the baker_binarized
                Alfajores_binarized
                Argentina Night_binarized
                .....
                Toast_binarized
                Truffles_binarized
                Tshirt_binarized
                Valentine?s card_binarized
                Vegan Feast_binarized
                Vegan mincepie_binarized
                Victorian Sponge_binarized
=== Associator model (full training set) ===

Apriori
=====

Minimum support: 0 (5 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 20

Generated sets of large itemsets:

Size of set of large itemsets L(1): 75
Size of set of large itemsets L(2): 387
Size of set of large itemsets L(3): 400
Size of set of large itemsets L(4): 65

Best rules found:

1. Cake_binarized=1 Hearty & Seasonal_binarized=1 7 ==> Coffee_binarized=1 7    <conf:(1)> lift:(2.1) lev:(0)
[3] conv:(3.67)
2. Extra Salami or Feta_binarized=1 Toast_binarized=1 6 ==> Coffee_binarized=1 6    <conf:(1)> lift:(2.1)
lev:(0) [3] conv:(3.15)
3. Farm House_binarized=1 Toast_binarized=1 6 ==> Coffee_binarized=1 6    <conf:(1)> lift:(2.1) lev:(0) [3]
conv:(3.15)
4. Farm House_binarized=1 Juice_binarized=1 5 ==> Coffee_binarized=1 5    <conf:(1)> lift:(2.1) lev:(0) [2]
conv:(2.62)
5. Bread_binarized=1 Medialuna_binarized=1 Muffin_binarized=1 5 ==> Coffee_binarized=1 5    <conf:(1)>
lift:(2.1) lev:(0) [2] conv:(2.62)
6. Bread_binarized=1 Sandwich_binarized=1 Spanish Brunch_binarized=1 5 ==> Coffee_binarized=1 5
<conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)
```

9.3.1 Rule1

```
Cake_binarized=1 Hearty & Seasonal_binarized=1 7 ==> Coffee_binarized=1 7    <conf:(1)> lift:(2.1) lev:(0)
[3] conv:(3.67)
```

- Number of transactions containing {*Cake, Hearty & Seasonal*} is **7**.
- Number of transactions containing {*Cake, Hearty & Seasonal, Coffee*} is **7**.
- Number of transactions is **9531**

- Calculate

① $\text{Support}(\{ \text{Cake, Hearty \& Seasonal} \} \rightarrow \{ \text{Coffee} \}):$

$$\frac{7}{9531} \approx 0.00073 > 0.0005$$

② $\text{Confidence}(\{ \text{Cake, Hearty \& Seasonal} \} \rightarrow \{ \text{Coffee} \}):$

$$\frac{7}{7} = 1 > 0.9$$

- According to this rule, it shows that

A customer is likely to buy **coffee**, if he buys **cake** and items which is **Hearty & Seasonal**.

9.3.2 Rule2

Extra Salami or Feta_binarized=1 Toast_binarized=1 6 ==> Coffee_binarized=1 6 <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.15)

- Number of transactions containing $\{ \text{Extra Salami or Feta, Toast} \}$ is **6**.
- Number of transactions containing $\{ \text{Extra Salami or Feta, Toast, Coffee} \}$ is **6**.
- Number of transactions is **9531**
- Calculate

① $\text{Support}(\{ \text{Extra Salami or Feta, Toast} \} \rightarrow \{ \text{Coffee} \}):$

$$\frac{6}{9531} \approx 0.00062 > 0.0005$$

② $\text{Confidence}(\{ \text{Extra Salami or Feta, Toast} \} \rightarrow \{ \text{Coffee} \}):$

$$\frac{6}{6} = 1 > 0.9$$

- According to this rule, it shows that

A customer is likely to buy **coffee**, if he buys **Extra Salami or Feta** and **Toast**.

9.3.3 Rule3

Farm House_binarized=1 Toast_binarized=1 6 ==> Coffee_binarized=1 6 <conf:(1)> lift:(2.1) lev:(0) [3] conv:(3.15)

- Number of transactions containing $\{ \text{Farm House, Toast} \}$ is **6**.
- Number of transactions containing $\{ \text{Farm House, Toast, Coffee} \}$ is **6**.
- Number of transactions is **9531**
- Calculate

① $\text{Support}(\{ \text{Farm House, Toast} \} \rightarrow \{ \text{Coffee} \}):$

$$\frac{6}{9531} \approx 0.00062 > 0.0005$$

② $\text{Confidence}(\{ \text{Farm House, Toast} \} \rightarrow \{ \text{Coffee} \}):$

$$\frac{6}{6} = 1 > 0.9$$

- According to this rule, it shows that

A customer is likely to buy **coffee**, if he buys **Toast** and items from **Farm House**.

9.3.4 Rule4

Farm House_binarized=1 Juice_binarized=1 5 ==> Coffee_binarized=1 5 <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)

- Number of transactions containing {Farm House, Juice} is **6**.
- Number of transactions containing {Farm House, Juice, Coffee} is **6**.
- Number of transactions is **9531**
- Calculate

① Support({Farm House, Juice} → {Coffee}):

$$\frac{5}{9531} \approx 0.00052 > 0.0005$$

② Confidence({Farm House, Juice} → {Coffee}):

$$\frac{5}{5} = 1 > 0.9$$

- According to this rule, it shows that

A customer is likely to buy **coffee**, if he buys **Juice** and items from **Farm House**.

9.3.5 Rule5

Bread_binarized=1 Medialuna_binarized=1 Muffin_binarized=1 5 ==> Coffee_binarized=1 5 <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)

- Number of transactions containing {Bread, Medialuna, Muffin} is **5**.
- Number of transactions containing {Bread, Medialuna, Muffin, Coffee} is **5**.
- Number of transactions is **9531**
- Calculate

① Support({Bread, Medialuna, Muffin} → {Coffee}):

$$\frac{5}{9531} \approx 0.00052 > 0.0005$$

② Confidence({Bread, Medialuna, Muffin} → {Coffee}):

$$\frac{5}{5} = 1 > 0.9$$

- According to this rule, it shows that

A customer is likely to buy **coffee**, if he buys **Bread, Medialuna** and **Muffin**.

9.3.6 Rule6

Bread_binarized=1 Sandwich_binarized=1 Spanish Brunch_binarized=1 5 ==> Coffee_binarized=1 5 <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)

- Number of transactions containing {Bread, Sandwich, Spanish Brunch} is **5**.
- Number of transactions containing {Bread, Sandwich, Spanish Brunch, Coffee} is **5**.

- Number of transactions is **9531**
- Calculate

① Support(*{Bread, Sandwich, Spanish Brunch}*→*{Coffee}*):

$$\frac{5}{9531} \approx 0.00052 > 0.0005$$

② Confidence(*{Bread, Sandwich, Spanish Brunch}*→*{Coffee}*):

$$\frac{5}{5} = 1 > 0.9$$

- According to this rule, it shows that

A customer is likely to buy **coffee**, if he buys **Bread, Sandwich** and **Spanish Brunch**.

9.4 Other finding

Rule	Support	Confidence
<i>{Cake, Hearty & Seasonal}</i> → <i>{Coffee}</i>	0.00073	1
<i>{Extra Salami or Feta, Toast}</i> → <i>{Coffee}</i>	0.00062	1
<i>{Farm House, Toast}</i> → <i>{Coffee}</i>	0.00062	1
<i>{Farm House, Juice}</i> → <i>{Coffee}</i>	0.00052	1
<i>{Bread, Medialuna, Muffin}</i> → <i>{Coffee}</i>	0.00052	1
<i>{Bread, Sandwich, Spanish Brunch}</i> → <i>{Coffee}</i>	0.00052	1

According to the result, the confidence values decrease from rule 1 to rule 6. The support values increase from rule 1 to rule 6 on the whole. The support and confidence values of these rules are similar. All the confidence values of the shown rules are more than 90% and the support values of the shown rules are more than 0.05%. Though the confidence of all the rules are 100%, the support is too low which means these rules do not apply to many cases.

In the result, all shown rules have a consequence---“coffee”, which means there is a strong correlation with buying coffee with buying other items. The result may help the manager of the supermarket to investigate the customer behavior and develop strategies. For example, manager can organize some advertising activities for coffee to customers who buy cake and Hearty & Seasonal item.