Abir Razzak

INFO 371 – Data Mining

Assignment 3

# Task 1

|  |  |
| --- | --- |
| Parameters | Output |
|  | === 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) |
|  | === Run information ===  Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.2 -S -1.0 -c -1  Relation: supermarket  Instances: 4627  Attributes: 217  [list of attributes omitted]  === Associator model (full training set) ===  Apriori  =======  Minimum support: 0.2 (925 instances)  Minimum metric <confidence>: 0.9  Number of cycles performed: 16  Generated sets of large itemsets:  Size of set of large itemsets L(1): 38  Size of set of large itemsets L(2): 225  Size of set of large itemsets L(3): 302  Size of set of large itemsets L(4): 80  Size of set of large itemsets L(5): 2  Best rules found: |
|  | === Run information ===  Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.25 -S -1.0 -c -1  Relation: supermarket  Instances: 4627  Attributes: 217  [list of attributes omitted]  === Associator model (full training set) ===  Apriori  =======  Minimum support: 0.25 (1157 instances)  Minimum metric <confidence>: 0.9  Number of cycles performed: 15  Generated sets of large itemsets:  Size of set of large itemsets L(1): 30  Size of set of large itemsets L(2): 132  Size of set of large itemsets L(3): 79  Size of set of large itemsets L(4): 6  Best rules found: |
|  | === Run information ===  Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.175 -S -1.0 -c -1  Relation: supermarket  Instances: 4627  Attributes: 217  [list of attributes omitted]  === Associator model (full training set) ===  Apriori  =======  Minimum support: 0.17 (810 instances)  Minimum metric <confidence>: 0.9  Number of cycles performed: 17  Generated sets of large itemsets:  Size of set of large itemsets L(1): 40  Size of set of large itemsets L(2): 297  Size of set of large itemsets L(3): 530  Size of set of large itemsets L(4): 218  Size of set of large itemsets L(5): 15  Best rules found:  1. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866 <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(3)  2. 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. biscuits=t milk-cream=t total=high 907 ==> bread and cake=t 820 <conf:(0.9)> lift:(1.26) lev:(0.04) [167] conv:(2.89)  4. biscuits=t vegetables=t total=high 950 ==> bread and cake=t 858 <conf:(0.9)> lift:(1.25) lev:(0.04) [174] conv:(2.86)  5. baking needs=t fruit=t total=high 963 ==> bread and cake=t 869 <conf:(0.9)> lift:(1.25) lev:(0.04) [175] conv:(2.84) |

The data presented were at 3 different levels for the lowerBoundMinSupport. At the default value of 0.1 we see that different various best rules were generated and the itemset bin numbers. At values of 0.25 and 0.5 we see that Weka no longer generated a best rules list out. At values less than 0.175 (in my testing, I used the values 0.1005, 0.15) the output is identical to 0.1. Testing at a value of 0.175 for the lowerBoundMinSupport yielded a list of 5 best rules found in the output as well as different itemset bins from 0.1.

# Task 2

When trying to open the .csv file from Kaggle directly in Weka, this error message is presented.

A screenshot of a cell phone

Description automatically generated

Line 102 is as follows:

2016-10-30,12:15:29,47,Ella's Kitchen Pouches

Here we see that apostrophe’s cause Weka to think that the beginning of a string is occurring. As per the instructions, I went and cleared out all apostrophe’s and replaced them with an empty string. The new attribute should now read: “Ellas Kitchen Pouches”. There were 30 instances where this occurred. “Ella’s Kitchen Pouches” and “Valentine’s Day Card” were the two biggest offenders of this.

After creating the pivot table and saving it as a .csv file, I then opened the .csv file of the pivot table and removed the rows and columns as per the instructions. I also noticed a column and row labeled “(blank)” that had no data in it, so I removed that column and row as well. The column **“Adjustment**” was not the second column of my output. In Excel, the adjustment column was actually column ‘AV’. I used Ctrl+F to find this column, however in the instructions of the lab it mentions that this column should have been the 2nd column. After saving the cleaned up .csv file, I tried opening the file in Weka and was presented with this error message:

A screenshot of a cell phone

Description automatically generated

I then opened the .csv file in a plain text editor (I used Visual Studio Code) and inserted a new line break at the end of line 9532 (end of file) to create a blank line at the end of the .csv file. This fixed the error message that was occurring when trying to open the .csv file generated from Excel. Upon opening the .csv file in Weka I was pleasantly greeted with the follow page:

A screenshot of a cell phone

Description automatically generated

Taking a look at the raw data in the editor tab, we see that Weka nicely greys out the empty cells, so that looking through the data for numbers is much more apparent and easier on the eyes:

A screenshot of a cell phone

Description automatically generated

# Task 3

First step is to apply the NumericToBinary unsupervised filter to the Transaction attribute. Here is a before and after applying the filter.

|  |  |
| --- | --- |
| Before | After |
|  |  |

We observe that Weka renames all the attributes to [old\_name]\_binarized after applying the NumericToBinary filter. We can now use this information to determine if an item has been purchased or not, and how many transactions that item has appeared in (where 0 represents that the item does not appear in the transaction, and 1 represents that the item does appear in the transaction). The counts are graphically displayed in the bottom right of the application, showing us how many times the item appears in the dataset of 9531 transactions. 0 does not have any counts in Weka, because in our dataset, we presented 0 items as blank, therefore after the binarization, the none translates into nothing, rather than 0. Instead, we can determine 0 by subtracting the count value of 1 from 9531 (total number of transactions apparent in the dataset); this will then yield our count for how many times the item is not present in a transaction. After looking at the data a bit, I continued on to the next set of instructions, which is to run the A-priori association rules on this dataset. Included below is a table of my parameters for the associations and their outputs.

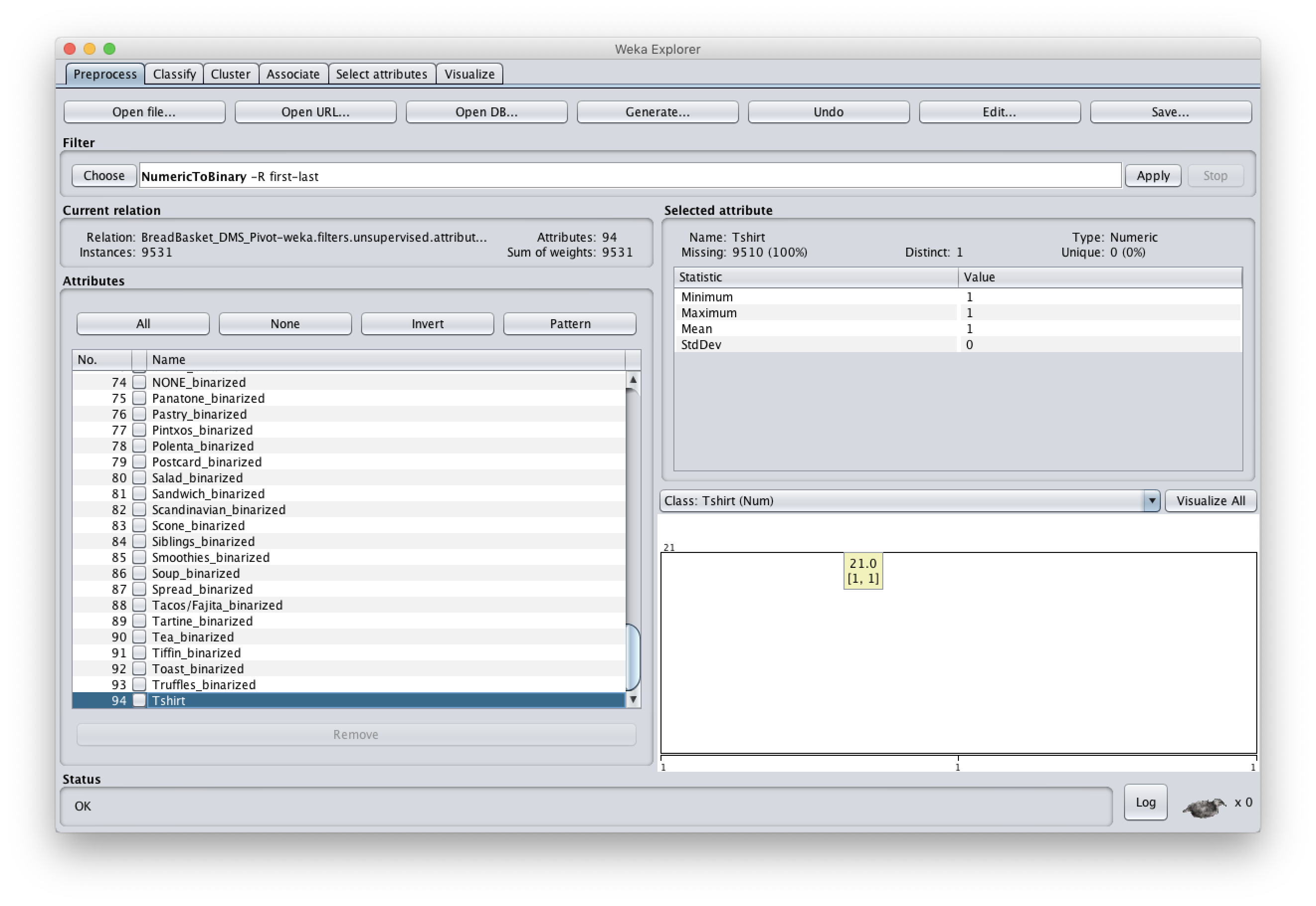
|  |  |
| --- | --- |
| A-priori Parameters | Output |
|  | === Run information ===  Scheme: weka.associations.FilteredAssociator -F "weka.filters.MultiFilter -F \"weka.filters.unsupervised.attribute.ReplaceMissingValues \"" -c -1 -W weka.associations.Apriori -- -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1  Relation: BreadBasket\_DMS\_Pivot-weka.filters.unsupervised.attribute.NumericToBinary-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R1  Instances: 9531  Attributes: 94  Afternoon with the baker\_binarized  Argentina Night\_binarized  Art Tray\_binarized  Bare Popcorn\_binarized  Bowl Nic Pitt\_binarized  Bread Pudding\_binarized  Brioche and salami\_binarized  Caramel bites\_binarized  Cherry me Dried fruit\_binarized  Chicken sand\_binarized  Chicken Stew\_binarized  Chimichurri Oil\_binarized  Christmas common\_binarized  Coffee granules \_binarized  Drinking chocolate spoons \_binarized  Duck egg\_binarized  Dulce de Leche\_binarized  Ellas Kitchen Pouches\_binarized  Extra Salami or Feta\_binarized  Fairy Doors\_binarized  Farm House\_binarized  Gift voucher\_binarized  Gingerbread syrup\_binarized  Hack the stack\_binarized  Half slice Monster \_binarized  Hearty & Seasonal\_binarized  Hot chocolate\_binarized  Jammie Dodgers\_binarized  Keeping It Local\_binarized  Kids biscuit\_binarized  Lemon and coconut\_binarized  Mighty Protein\_binarized  Mineral water\_binarized  My-5 Fruit Shoot\_binarized  Nomad bag\_binarized  Olum & polenta\_binarized  Pick and Mix Bowls\_binarized  Raspberry shortbread sandwich\_binarized  Raw bars\_binarized  Spanish Brunch\_binarized  The BART\_binarized  The Nomad\_binarized  Valentines card\_binarized  Vegan Feast\_binarized  Vegan mincepie\_binarized  Victorian Sponge\_binarized  Alfajores\_binarized  Bacon\_binarized  Baguette\_binarized  Bakewell\_binarized  Basket\_binarized  Bread\_binarized  Brownie\_binarized  Cake\_binarized  Chocolates\_binarized  Coffee\_binarized  Coke\_binarized  Cookies\_binarized  Crepes\_binarized  Crisps\_binarized  Eggs\_binarized  Empanadas\_binarized  Focaccia\_binarized  Frittata\_binarized  Fudge\_binarized  Granola\_binarized  Honey\_binarized  Jam\_binarized  Juice\_binarized  Medialuna\_binarized  Mortimer\_binarized  Muesli\_binarized  Muffin\_binarized  NONE\_binarized  Panatone\_binarized  Pastry\_binarized  Pintxos\_binarized  Polenta\_binarized  Postcard\_binarized  Salad\_binarized  Sandwich\_binarized  Scandinavian\_binarized  Scone\_binarized  Siblings\_binarized  Smoothies\_binarized  Soup\_binarized  Spread\_binarized  Tacos/Fajita\_binarized  Tartine\_binarized  Tea\_binarized  Tiffin\_binarized  Toast\_binarized  Truffles\_binarized  Tshirt  …  (No best rules found – nothing else was after the list of attributes) |

Huh? That’s an odd output – Let’s check the logs. An error had occurred. The contents of the log file:

14:44:59: Command: weka.associations.FilteredAssociator -F "weka.filters.MultiFilter -F \"weka.filters.unsupervised.attribute.ReplaceMissingValues \"" -c -1 -W weka.associations.Apriori -- -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

14:44:59: weka.associations.Apriori: Cannot handle numeric attributes!

The error message is saying that A-priori cannot handle numeric attributes… How odd. I looked through the dataset, and I found the offending attribute:



The Tshirt attribute was not binarized for some reason. After many attempts of reopening the file and running the filter with different selections, I decided to remove the Tshirt attribute altogether after running the NumericToBinary filter. That seemed to fix both the error being thrown, as well as letting me run the normal A-priori association. There did not seem to be a workaround for this besides removing the attribute altogether. Please let that be a factor when reviewing the results of my Weka outputs. Back to the data collection:

|  |  |
| --- | --- |
| A-priori Parameters | Output |
|  | === 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: BreadBasket\_DMS\_Pivot-weka.filters.unsupervised.attribute.NumericToBinary-Rfirst-last-weka.filters.unsupervised.attribute.NumericToBinary-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R1-weka.filters.unsupervised.attribute.Remove-R94  Instances: 9531  Attributes: 93  Afternoon with the baker\_binarized  Argentina Night\_binarized  Art Tray\_binarized  Bare Popcorn\_binarized  Bowl Nic Pitt\_binarized  Bread Pudding\_binarized  .  .  .  Toast\_binarized  Truffles\_binarized  === Associator model (full training set) ===  No large itemsets and rules found! |
|  | Apriori  =======  Minimum support: 0.05 (477 instances)  Minimum metric <confidence>: 0.9  Number of cycles performed: 19  Generated sets of large itemsets:  Size of set of large itemsets L(1): 10  Size of set of large itemsets L(2): 2  Best rules found: |
|  | Apriori  =======  Minimum support: 0.01 (95 instances)  Minimum metric <confidence>: 0.9  Number of cycles performed: 20  Generated sets of large itemsets:  Size of set of large itemsets L(1): 31  Size of set of large itemsets L(2): 31  Size of set of large itemsets L(3): 3  Best rules found: |
|  | Apriori  =======  Minimum support: 0 (10 instances)  Minimum metric <confidence>: 0.9  Number of cycles performed: 20  Generated sets of large itemsets:  Size of set of large itemsets L(1): 57  Size of set of large itemsets L(2): 271  Size of set of large itemsets L(3): 201  Size of set of large itemsets L(4): 7  Best rules found:  1. NONE\_binarized=1 Tartine\_binarized=1 13 ==> Coffee\_binarized=1 12 <conf:(0.92)> lift:(1.94) lev:(0) [5] conv:(3.41)  2. Hot chocolate\_binarized=1 Medialuna\_binarized=1 NONE\_binarized=1 11 ==> Coffee\_binarized=1 10 <conf:(0.91)> lift:(1.91) lev:(0) [4] conv:(2.89) |
|  | Apriori  =======  Minimum support: 0 (1 instances)  Minimum metric <confidence>: 0.9  Number of cycles performed: 20  Generated sets of large itemsets:  Size of set of large itemsets L(1): 93  Size of set of large itemsets L(2): 1098  Size of set of large itemsets L(3): 3620  Size of set of large itemsets L(4): 4442  Size of set of large itemsets L(5): 2896  Size of set of large itemsets L(6): 1460  Size of set of large itemsets L(7): 627  Size of set of large itemsets L(8): 204  Size of set of large itemsets L(9): 42  Size of set of large itemsets L(10): 4  Best rules found:  1. Hearty & Seasonal\_binarized=1 Cake\_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. Hot chocolate\_binarized=1 NONE\_binarized=1 Scone\_binarized=1 5 ==> Coffee\_binarized=1 5 <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)  6. Spanish Brunch\_binarized=1 Bread\_binarized=1 Sandwich\_binarized=1 5 ==> Coffee\_binarized=1 5 <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)  7. 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)  8. Cake\_binarized=1 Juice\_binarized=1 NONE\_binarized=1 5 ==> Coffee\_binarized=1 5 <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.62)  9. Bread Pudding\_binarized=1 4 ==> Coffee\_binarized=1 4 <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.1)  10. Extra Salami or Feta\_binarized=1 Juice\_binarized=1 4 ==> Coffee\_binarized=1 4 <conf:(1)> lift:(2.1) lev:(0) [2] conv:(2.1) |

*\*abbreviated to fit within table. All attributes are listed in the output.*

Using a lowerBoundMinSupport of 0.025, caused my laptop to run into a never-ending process. I had to force quit Weka because after about 20 minutes, Weka was still trying to process the data. However, sticking with 2 decimal points, and working my down from 0.05, to 0.01 seemed to work no problem. Afterwards, I went to test my luck again with a value of 0.0075, and the Weka compiled it properly. Not sure why it hung earlier when I tried 0.025… strange. I kept decreasing the number until I got to 0.001. Finally, Weka generated some best rules. Only two though however, so I tried to lower the number some more. Simply shifting the number down by 1 digit, to 0.0001 did the trick. Weka generated 10 best rules. The sweet spot will exist somewhere between 0.001 and 0.0001 for the lowerBoundMinSupport to generate the full 10 best rules.