Assignment 5: Market Basket Analysis

CIS 435

Section 56

Summer Quarter

School of Continuing Studies

Northwestern University

Daniel Prusinski

Business Intelligence Data Analyst

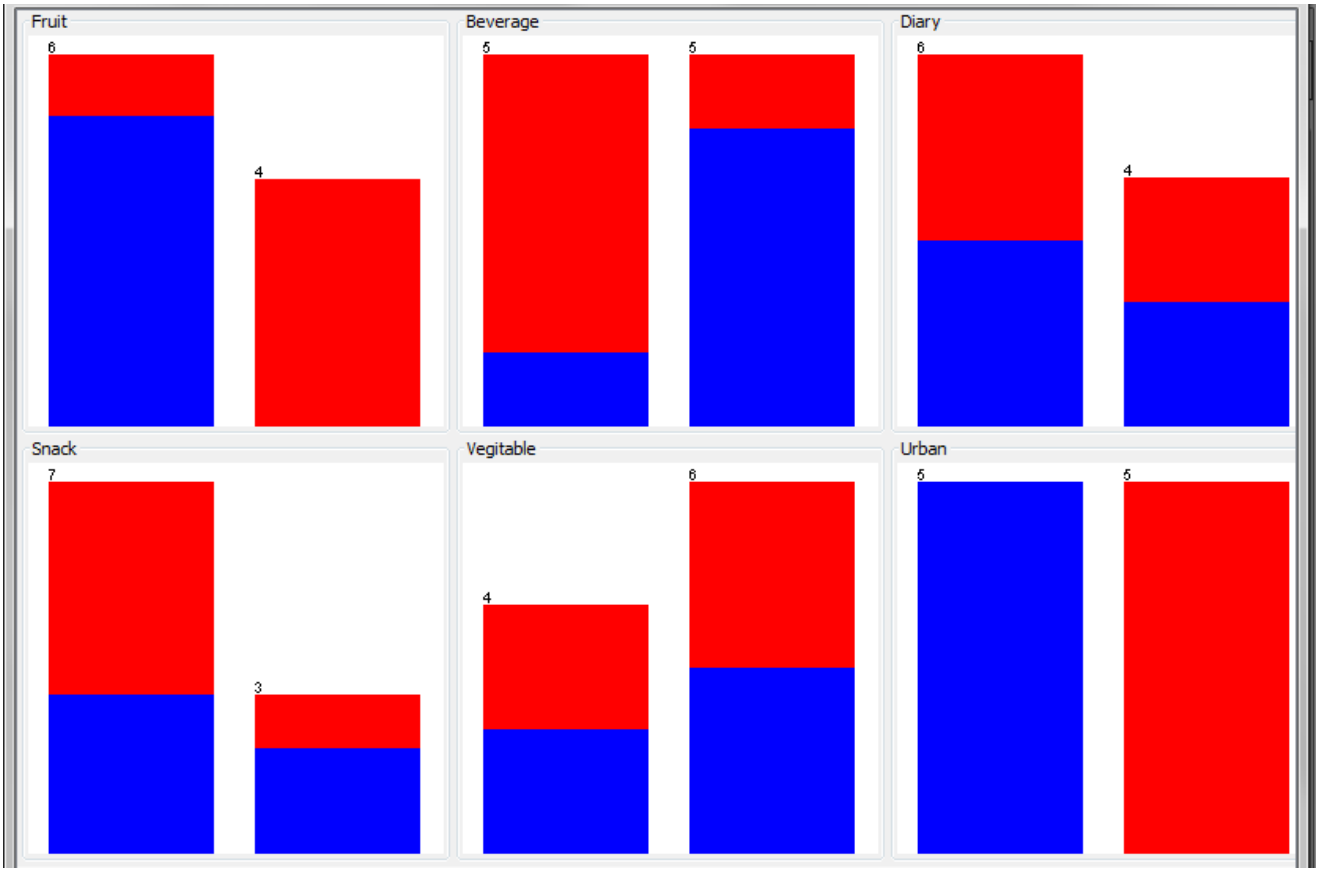
Target Corporation

Minneapolis, MN

In Compliance with Master of Science Predictive Analytics

Preliminary Data Analysis:

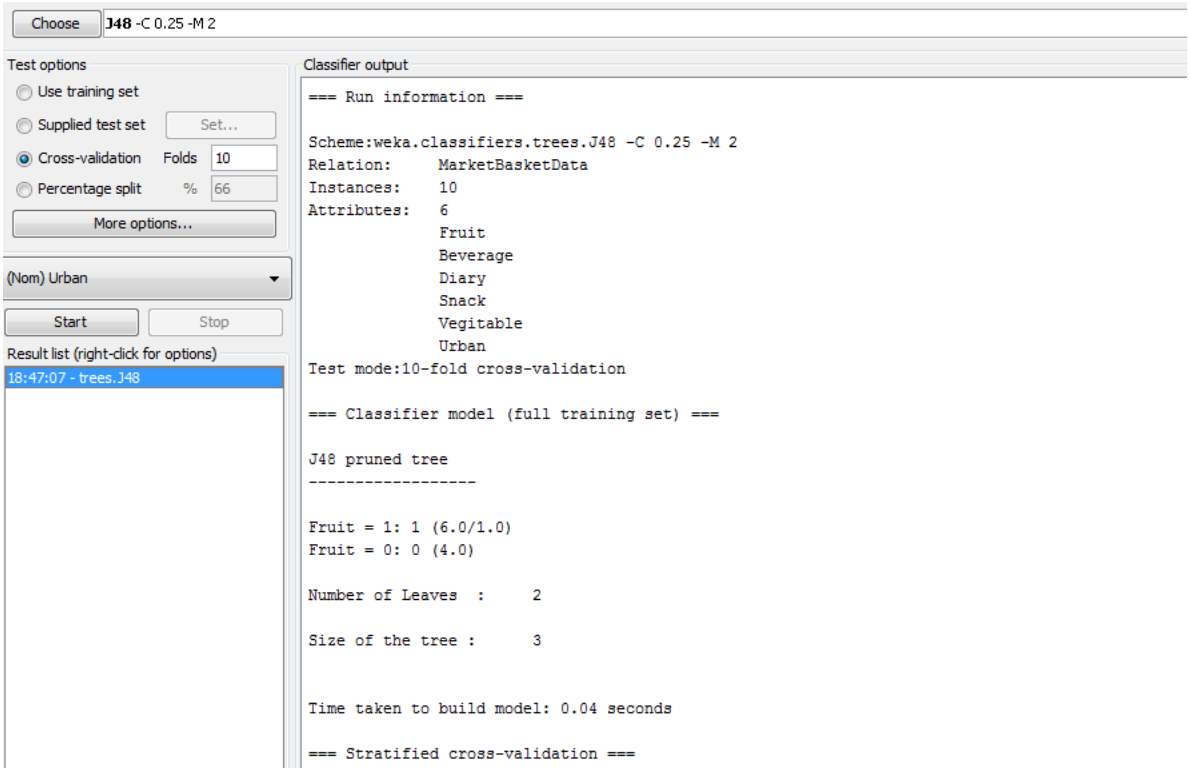
The Market Basket data is comprised of 10 instances, each instance is comprised of 6 attributes. My initial take-away is this data set has too small of a population from which to draw strong statistical analysis of to infer on a larger data set. Visually the data is expressed below:



Each variable appears to have a binary distribution such that the values for the variables are collated within the binary response with the exception for Urban. In this instance, there are two values for Urban and all the instances belonging to blue are in label one and all instances belonging to red are in label 0.

A. Initial analysis using the J48 Algorithm to construct a Decision Tree:

J48 is a top-down approach that separates the example data into subsets (decision tree) and new observations are scored through this tree. The output from J48 is shown below. The selection of the J48 can also be seen in the screenshot below:



=== Run information ===
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: MarketBasketData
Instances: 10
Attributes: 6
Fruit
Beverage
Diary
Snack
Vegitable
Urban
Test mode:10-fold cross-validation
=== Classifier model (full training set) ===
J48 pruned tree

Fruit = 1: 1 (6.0/1.0)
Fruit = 0: 0 (4.0)
Number of Leaves : 2
Size of the tree : 3
Time taken to build model: 0.03 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 9 90 %
Incorrectly Classified Instances 1 10 %
Kappa statistic 0.8
Mean absolute error 0.2

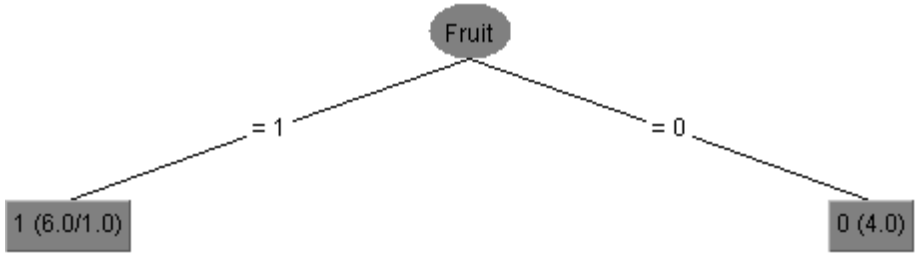
Root mean squared error 0.3464
Relative absolute error 36.6667 %
Root relative squared error 63.5085 %
Total Number of Instances 10
=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0.2	0.833	1	0.909	0.8	1
	0.8	0	1	0.8	0.889	0.8	0
Weighted Avg.	0.9	0.1	0.917	0.9	0.899	0.8	

=== Confusion Matrix ===

a	b	<-- classified as
5	0	a = 1
1	4	b = 0

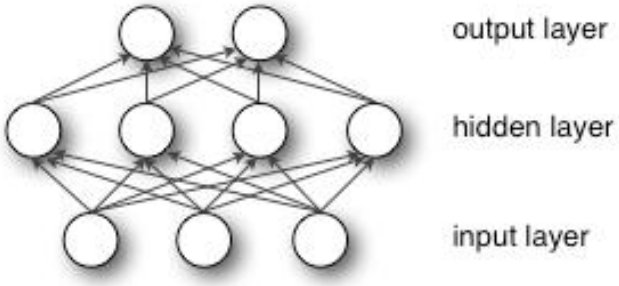
1. The rate of correctly classified instances using J48 is 9, or 90%.
2. The final tree is shown below.



I am curious to know why J48 chose Fruit over Urban, when Urban has a perfect classification? This can be seen visually in the prelimary analysis above.

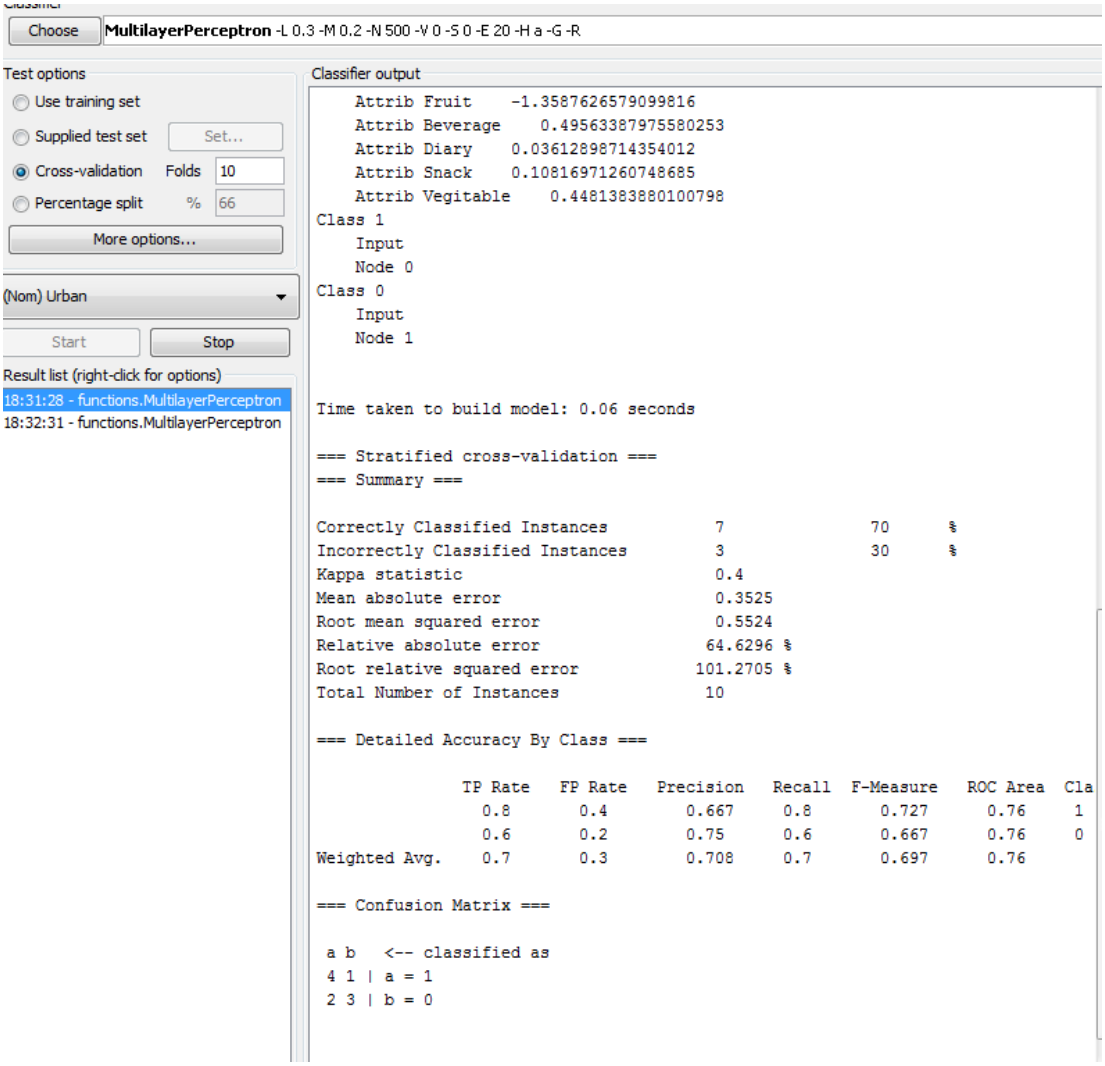
B. Analysis using Multilayer Perceptron (MLP):

Logistic Regression is used as a non-linear transformer in the MLP process. The goal of using logistic regression is to linearly separate data that initially is not linear. Shown below is an example of a Neural Network that has one hidden layer:



Using the approach highlighted above, I will fit the MarketBasket data with MLP. The output is shown below.

The pre-process of selecting the MLP and running it can be seen in the screen shot below:



=== Run information ===

Scheme:weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: MarketBasketData
Instances: 10
Attributes: 6
Fruit
Beverage
Diary
Snack
Vegitable
Urban
Test mode:evaluate on training data

=== Classifier model (full training set) ===

Sigmoid Node 0

Inputs Weights
Threshold 2.6367946317178976
Node 2 -3.243098628596855
Node 3 -4.6537223483909145
Node 4 1.5387459029926687
Sigmoid Node 1
Inputs Weights
Threshold -2.619510267515357
Node 2 3.175139575970039
Node 3 4.706404682587317
Node 4 -1.5571858085923795
Sigmoid Node 2
Inputs Weights
Threshold 1.197287162895843
Attrib Fruit 3.1915934158188843
Attrib Beverage -1.461353897190328
Attrib Diary -0.22431776336278011
Attrib Snack -0.4258112018179538
Attrib Vegitable -1.5442199565454817
Sigmoid Node 3
Inputs Weights
Threshold 1.7963480799010019
Attrib Fruit 4.166334854987007
Attrib Beverage -1.8809897753613294
Attrib Diary -0.3113764461723796
Attrib Snack -0.551828669087384
Attrib Vegitable -2.051130313452862
Sigmoid Node 4
Inputs Weights
Threshold 0.01738019645068224
Attrib Fruit -1.3587626579099816
Attrib Beverage 0.49563387975580253
Attrib Diary 0.03612898714354012
Attrib Snack 0.10816971260748685
Attrib Vegitable 0.4481383880100798
Class 1
Input
Node 0
Class 0
Input
Node 1
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	7	70	%
Incorrectly Classified Instances	3	30	%
Kappa statistic	0.4		

Mean absolute error 0.3525
Root mean squared error 0.5524
Relative absolute error 64.6296 %
Root relative squared error 101.2705 %
Total Number of Instances 10
=== Detailed Accuracy By Class ===

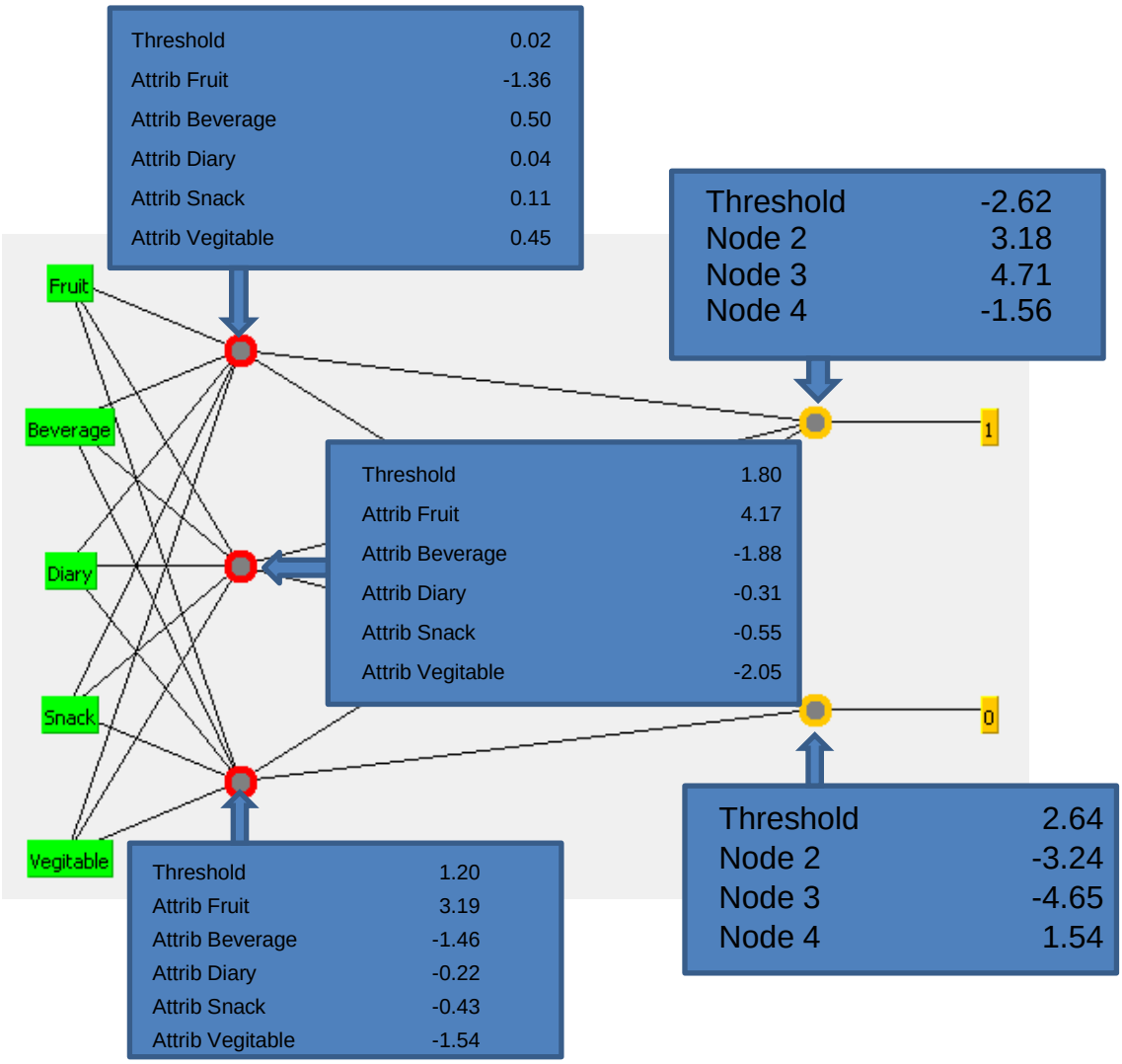
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.8	0.4	0.667	0.8	0.727	0.76	1
	0.6	0.2	0.75	0.6	0.667	0.76	0
Weighted Avg.	0.7	0.3	0.708	0.7	0.697	0.76	

=== Confusion Matrix ===

```
 a b   <-- classified as
 4 1 | a = 1
 2 3 | b = 01.
```

The rate of correctly classified instances is 70% or 7 instances were classified correctly. The highlighted section above shows this in the output above.

2. The visual representation of the network can be seem below along with weights:



The weights as seen above are best explained through the words of Professor David Leverington, "The perceptron is trained (i.e., the weights and threshold values are calculated) based on an iterative training phase involving training data. Training data are composed of a list of input values and their associated desired output values. In the training phase, the inputs and related outputs of the training data are repeatedly submitted to the perceptron. The perceptron calculates an output value for each set of input values. If the output of a particular training case is labelled 1 when it should be labelled 0, the threshold value (theta) is increased by 1, and all weight values associated with inputs of 1 are decreased by 1. The opposite is performed if the output of a training case is labelled 0 when it should be labelled 1. No changes are made to the threshold value or weights if a particular training case is correctly classified. This set of training rules is summarized as:

(eqn 2a)
If OUTPUT is correct, then no changes are made to the threshold or weights

(eqn 2b)
If OUTPUT = 1, but should be 0
then {theta = theta + 1}
and {$weight_x$ = $weight_x$ -1, if $input_x$ = 1}

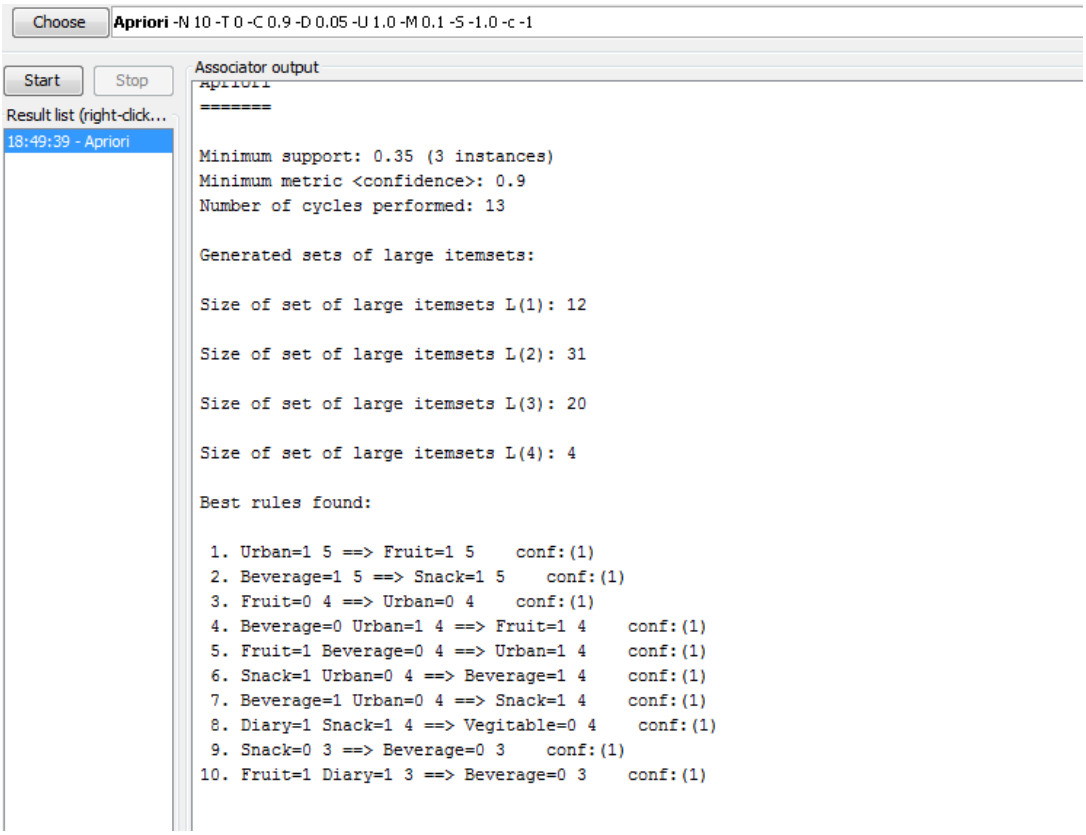
(eqn 2c)
If OUTPUT = 0, but should be 1
then {theta = theta - 1}
and {$weight_x$=$weight_x$ +1, if inputx = 1}

where the subscript x refers to a particular input-node and weight pair. The effect of the above training rules is to make it less likely that a particular error will be made in subsequent training iterations. For example, in equation (2b), increasing the threshold value serves to make it less likely that the same sum of products will exceed the threshold in later training iterations, and thus makes it less likely that an output value of 1 will be produced when the same inputs are presented. Also, by modifying only those weights that are associated with input values of 1, only those weights that could have contributed to the error are changed (weights associated with input values of 0 are not considered to have contributed to error). Once the network is trained, it can be used to classify new data sets whose input/output associations are similar to those that characterize the training data set. Thus, through an iterative training stage in which the weights and threshold gradually migrate to useful values (i.e., values that minimize or eliminate error), the perceptron can be said to "learn" how to solve simple problems."

One can see the negative and positive thresholds increase as the nodes increase in their forward moving notion. This is visually seen as the nodes move to the right of the diagram above. Each node had multiple iterations, which is seen through the increase in weight. Given that this data set is small, this impacts the size of the weights.

C. Analysis using Apriori method:

The Apriori algorithm uses multiple iterations to find minimum support for the minimum number of rules for a specific minimum confidence. The output generated from this algorithm can be seen below:



=== 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: MarketBasketData
Instances: 10
Attributes: 6
Fruit
Beverage
Diary
Snack
Vegitable
Urban
=== Associator model (full training set) ===
Apriori
=======
Minimum support: 0.35 (3 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 13
Generated sets of large itemsets:

Size of set of large itemsets L(1): 12
Size of set of large itemsets L(2): 31
Size of set of large itemsets L(3): 20
Size of set of large itemsets L(4): 4
Best rules found:
1. Urban=1 5 ==> Fruit=1 5 conf:(1)
2. Beverage=1 5 ==> Snack=1 5 conf:(1)
3. Fruit=0 4 ==> Urban=0 4 conf:(1)
4. Beverage=0 Urban=1 4 ==> Fruit=1 4 conf:(1)
5. Fruit=1 Beverage=0 4 ==> Urban=1 4 conf:(1)
6. Snack=1 Urban=0 4 ==> Beverage=1 4 conf:(1)
7. Beverage=1 Urban=0 4 ==> Snack=1 4 conf:(1)
8. Diary=1 Snack=1 4 ==> Vegitable=0 4 conf:(1)
9. Snack=0 3 ==> Beverage=0 3 conf:(1)
10. Fruit=1 Diary=1 3 ==> Beverage=0 3 conf:(1)

1. The minimum support value is .35 or 3 instances.
2. The minimum confidence value is .9.
3. Two examples of association rules:
   - If Dairy and Snack then don't buy vegetable
   - If Fruit and Dairy then don't buy Beverages.

D. Summary of Analysis

1. When comparing decision trees with neural nets, the following observations were discovered:
   - This is the first class I have taken where I have encountered neural networks, and of the three methods explored in this Exploratory Data Analysis (EDA) I feel the leas trained on this approach. While computationally heavy, the neural network utilizes techniques similar to linear and logistic regression for ascertaining fit. The divergence is seen in the multiple iterations that happen between the nodes as well as the output.
   - The decision tee correctly classified 90% of the instances compared to the 70% of the neural network. In my opinion, based on the root mean square error of .35 for the decision tree and .55 for the neural network, the decision tree is a better decision tool for this data.
   - Given that the data only had a few instances, in a situation with more data I believe the neural network would prove to have consistently better output. Simply put, Decision Tree is easier to understand and in my experience has better output when working with smaller data sets.
2. When comparing decision trees with association rules, the following insights were observed:
   - From a data mining perspective, I prefer association rules based on how the data is structured and the fact that it is a rather intuitive process. I would like to further explore how data structured in 3NF affects the speed of association process.

- The decision tee correctly classified 90% of the instances compared to the .35 support with .9 confidence of the association rule. In my opinion, based on the root mean square error of .346 for the decision tree and .35 support for the neural network, the decision tree is a better decision tool for this data.
- From a classification standpoint, I would use the association rule in a presentation with marketing personnel. The probability of association, despite its output, is of great value to grouping and how items are marketed. Advertising personnel love to see data through the lens of association because it gives them an advantage when discounting and growing the basket size.

3. When comparing neural nets and association, the following discoveries were observed.

- Rather than point out the classification percentages between neural nets and association, which can directly be seen above, I would like to point out the strengths of both methods. As noted earlier, neural nets are probably the most mathematically intense of the classifiers that we have learned in the MSPA program. In my opinion, more time needs to be spent on this technique based on the fact that the output often fits the data the best in both non-linear and linear data. From an inductive perspective, the neural net provides the most in depth analysis of grouping situations with items. As a data scientist, analyzing the different nodes proves to be very helpful in identifying different associations. Yet, from a presentation standpoint it would be very hard to explain this method to anyone not in the data-science field. In this instance, association works much better for grocery or shopping data based on the end goal for where we want the data to lead.
- The association does well with shopping data because it is easy to understand items groupings based on how customers consume goods. With a different data set, association may not be a better situation, bear in mind that the support was only 35% with 90% confidence.

Conclusion:

Three different algorithms were used to fit the market basket. Association and Decision Tree have been used in the past, but this was the first time using Multilayer Perceptron. From a numerical standpoint the decision tree was the best method. The neural net provided the most depth into the different relationships amongst the individual's items. From an industry perspective, the association rule is most often used based on the value it adds to increasing basket size.