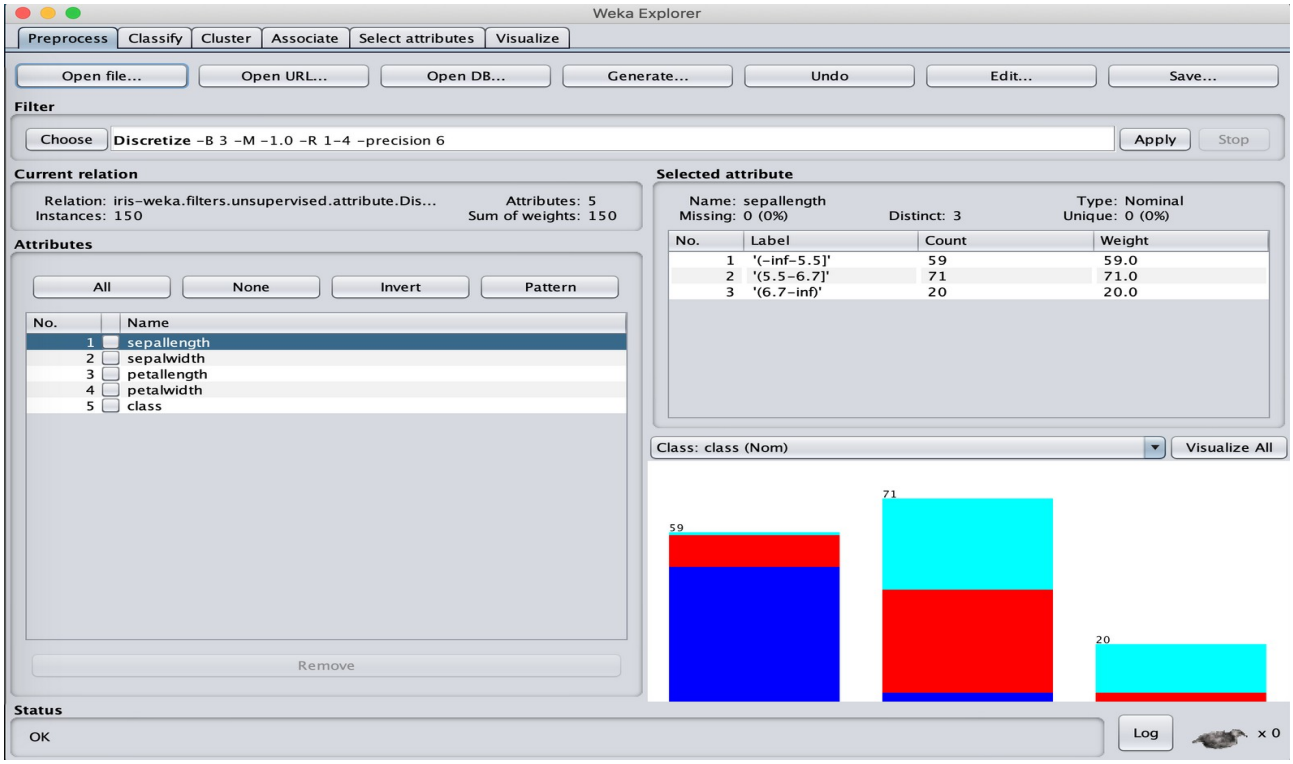
Data Mining
Lab2 - Association Analysis - 1

Zhixuan_Duan(zhidu838)
2020-02-25

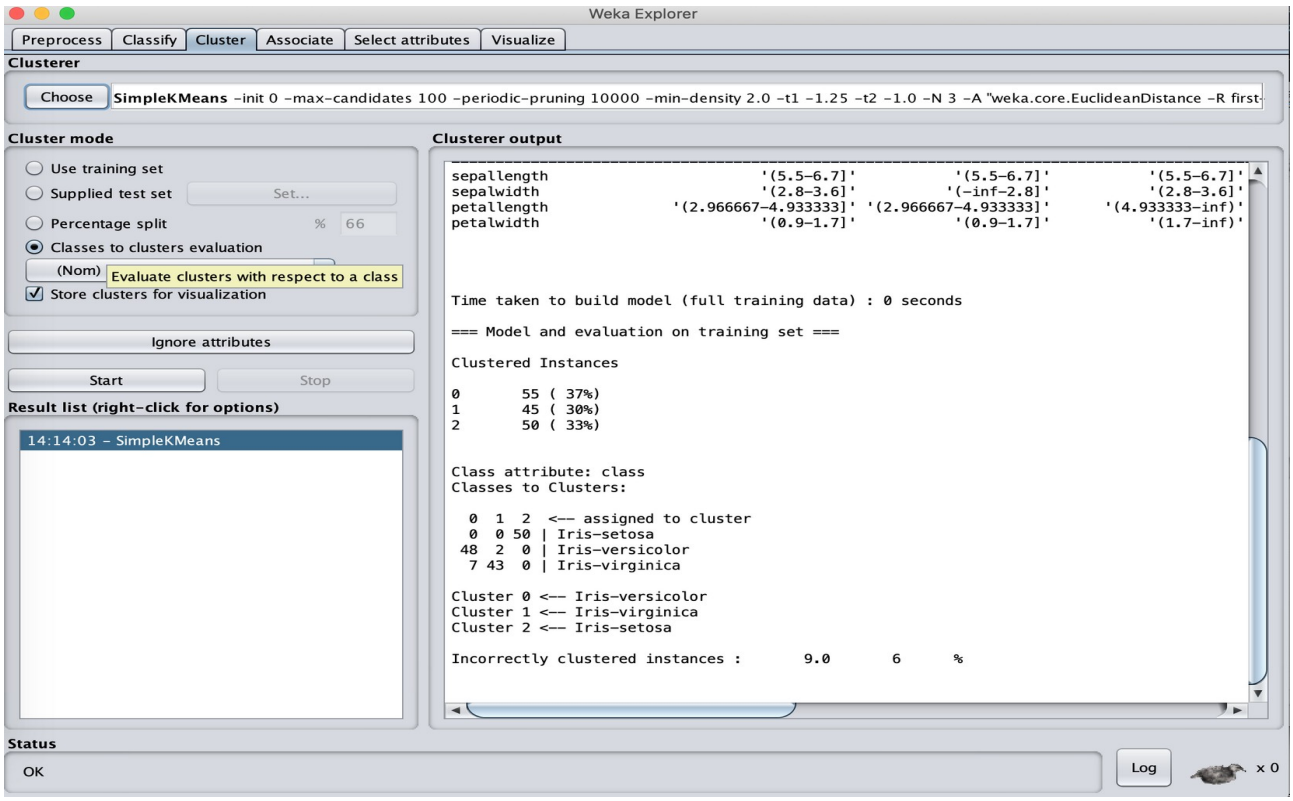
1. Discretize the iris dataset

Open the iris.arff file. Since the association analysis in Weka (Apriori algorithm) cannot cope with continuous attributes, we should discretize the iris dataset before starting the mining process. Weka provides several filters to apply to the data. You can see them by pressing the **Choose** button. We are interested in the Discretize filter that you can find by selecting the directory Unsupervised first and then the directory Attribute. Now, click on the line that has appeared to the right of the Choose button to edit the properties of the filter. You can find a detailed description of the filter by pressing the **More** button. Select the attributes indices 1-4 (meaning that you do not want to discretize the 5th attribute as it is the class and thus already discrete) and select 3 bins (number of states of the discretized attributes). Press **OK**. Press **Apply**. Now you can edit the data again by pressing the **Edit** button and see that the data has actually been discretized.



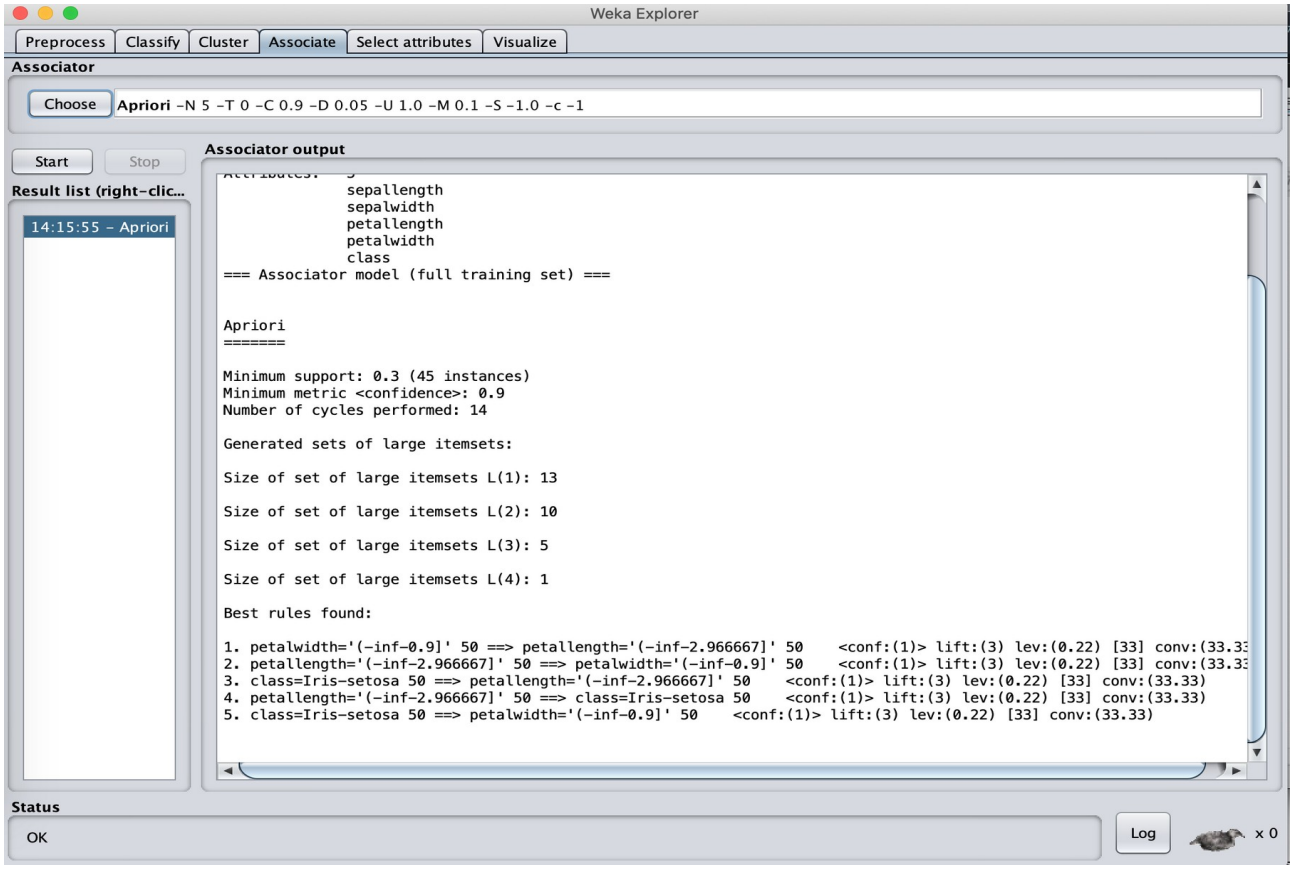
2. Clustering

Appy SimpleKmeans clusterer to the data with 3 clusters (since we know there are 3 types of Iris flowers) and seed value 10. In the Cluster mode, select Classes to clusters evaluation to crosstabulate the clustering and class labeling. Ignore the class attribute.



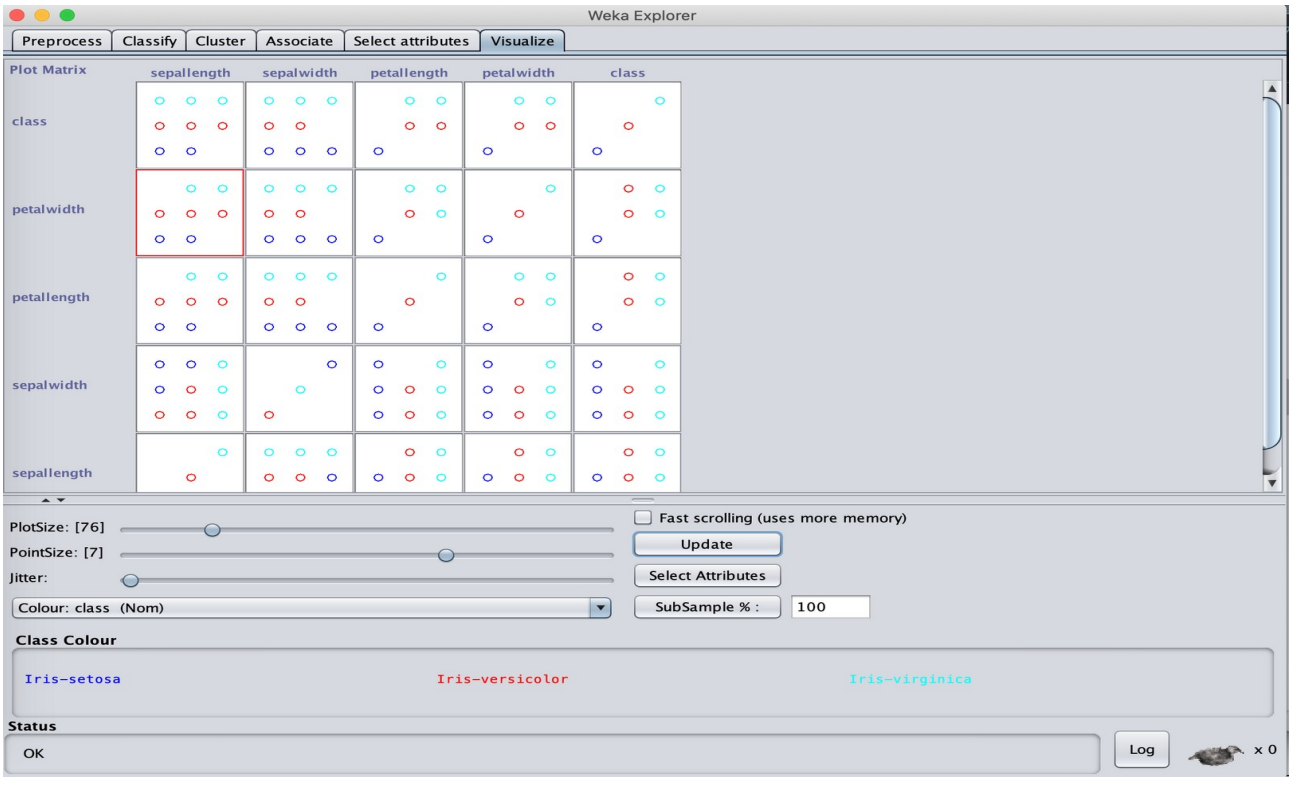
Association analysis

To perform association analysis, click on the **Associate** tab. Press the **Choose** button to select the association algorithm (we recommend to use the Apriori algorithm). Click on the line that has appeared to the right of the Choose button to edit the properties of the algorithm. You can find a detailed description of the algorithm by pressing the **More** button. Set the desired properties and press **OK**. Click **Start**. Check the output on the right hand side of the screen. Note that after the conjunctions of attribute-value pairs on the right and left hand sides of each rule, there is a number. That number indicates the support of the determinant and of the determinant plus the consequent.



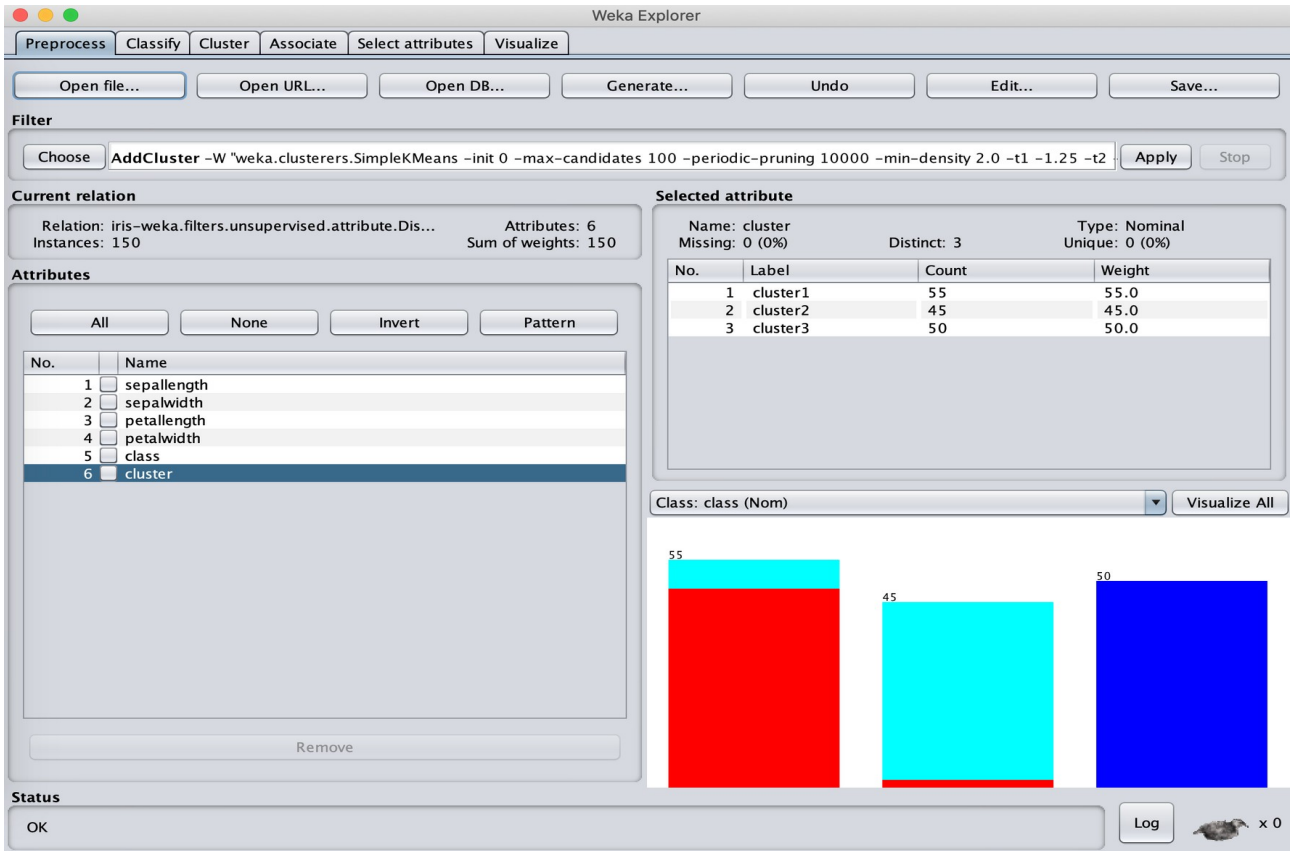
Visualization

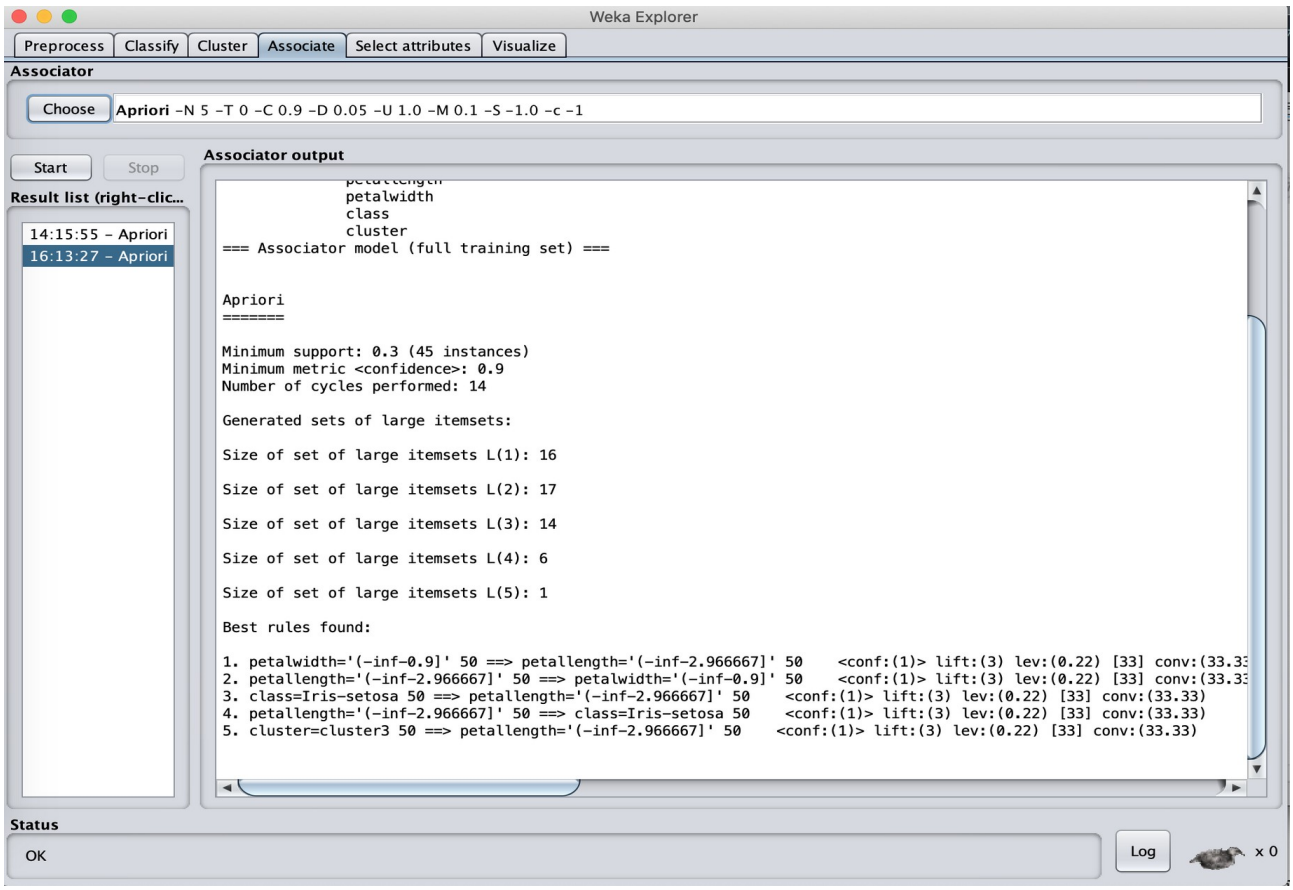
By clicking on the **Visualize** tab, you can see the data crosstabulated for each pair of attributes. Set your visualization preferences with the bars at the bottom of the screen.



Describing clustering through association analysis

Now we use association analysis to assist you in describing the clusters found in the Iris dataset. The first thing to do is to create a new attribute that represents the cluster label assigned to each instance. For this purpose, click on the Preprocess tab and select the AddCluster filter (in the Attribute directory within the Unsupervised directory). Click on the line that has appeared to the right of the Choose button to edit the properties of the filter, e.g. which clustering algorithm to use, number of clusters (we recommend to use 3 clusters), ignored attributes (ignore the class), etc. Check the section on clustering above if in doubt. Press Apply. Check that a 6th attribute has been created with the clustering label. Now, run the association analysis as indicated in the corresponding section above. Find rules that are accurate and such that the antecedent does not contain the class attribute and the consequent only contains the cluster attribute. Find such rules for the 3 clusters. This should help you to describe the instances grouped in each cluster. Repeat the exercise above with a different combination of clustering algorithm, number of clusters and/or number of bins in the discretization filter, in order to see whether you get better or worse results.





By printing more rules, we find such rules that only contains cluster as output:

1. petallength='(-inf-2.966667]' 50 ==> cluster=cluster3 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)

2. petalwidth='(-inf-0.9]' 50 ==> cluster=cluster3 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)

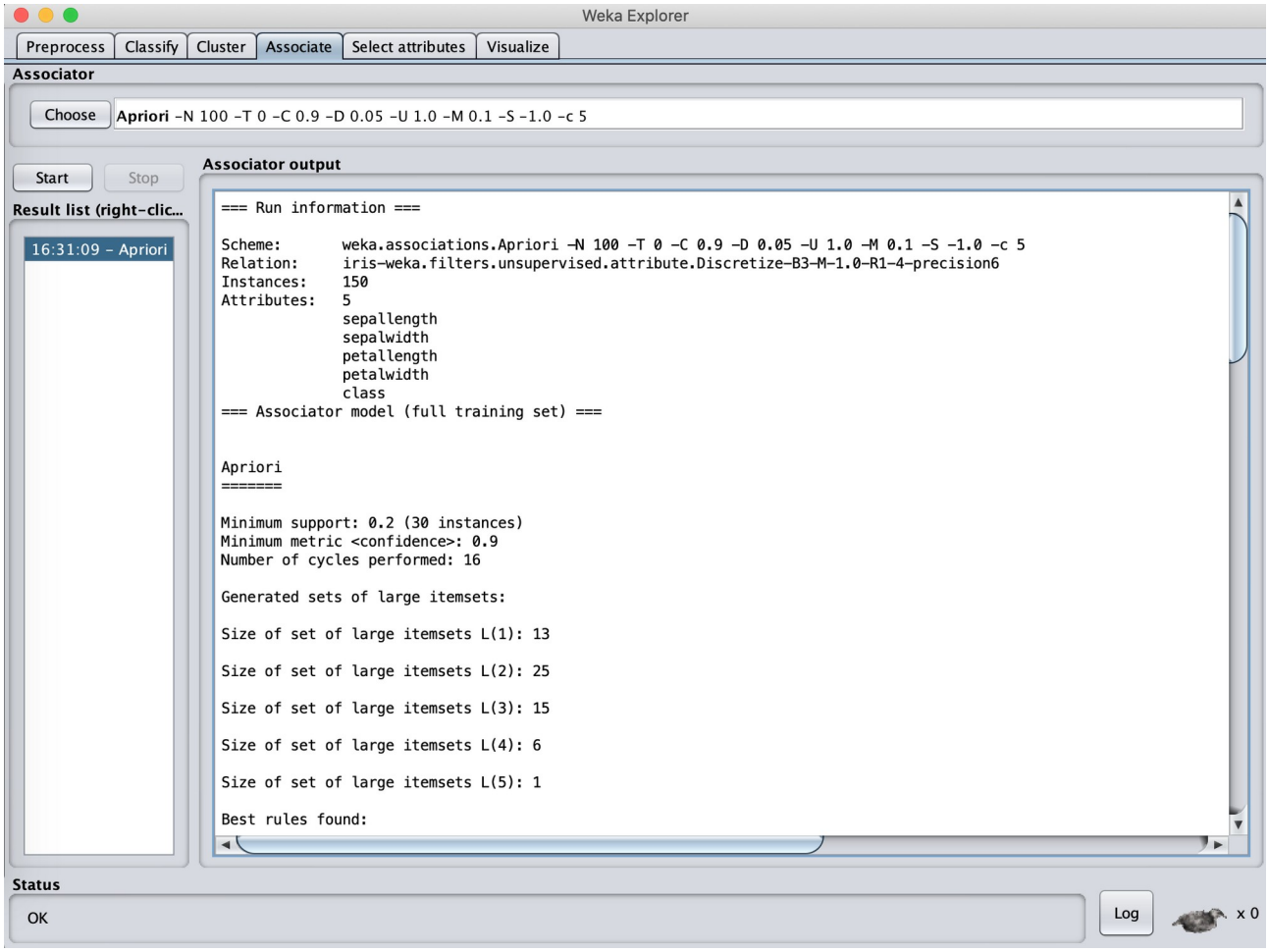
3. petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50 ==> cluster=cluster3 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)

Analysis:

After adding a new attribute cluster, and run association analysis based on the k-means and the Apriori-type algorithm, we find best 5 rules as indicated in the plot above. And by checking that, we find the minimum support is 0.3, which implies that there maybe some correlations we interested, and the minimum confidence is 0.9, which suggest the rules we find are pretty strong.

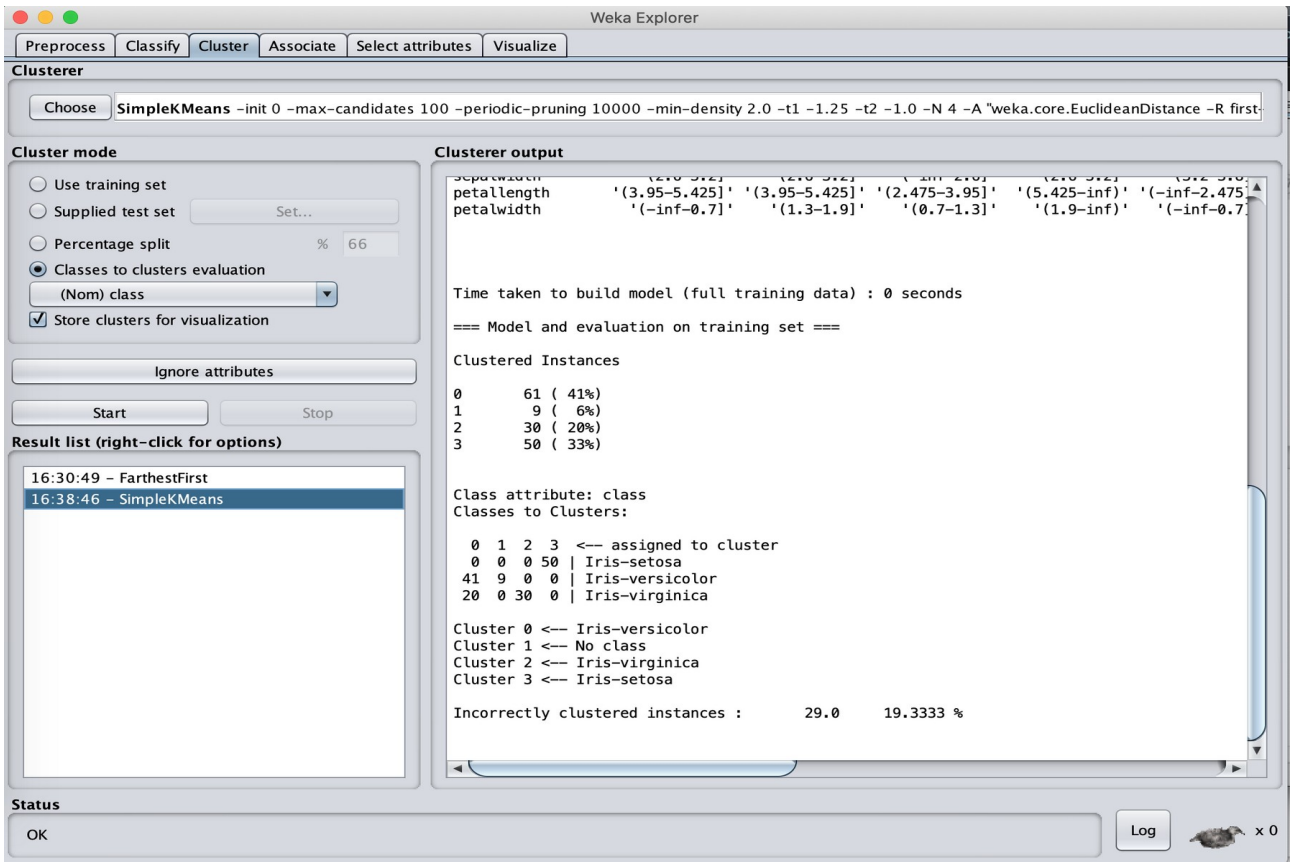
Different clustering algorithm (avoid using MakeDensityBasedCluster wrapper in this case)

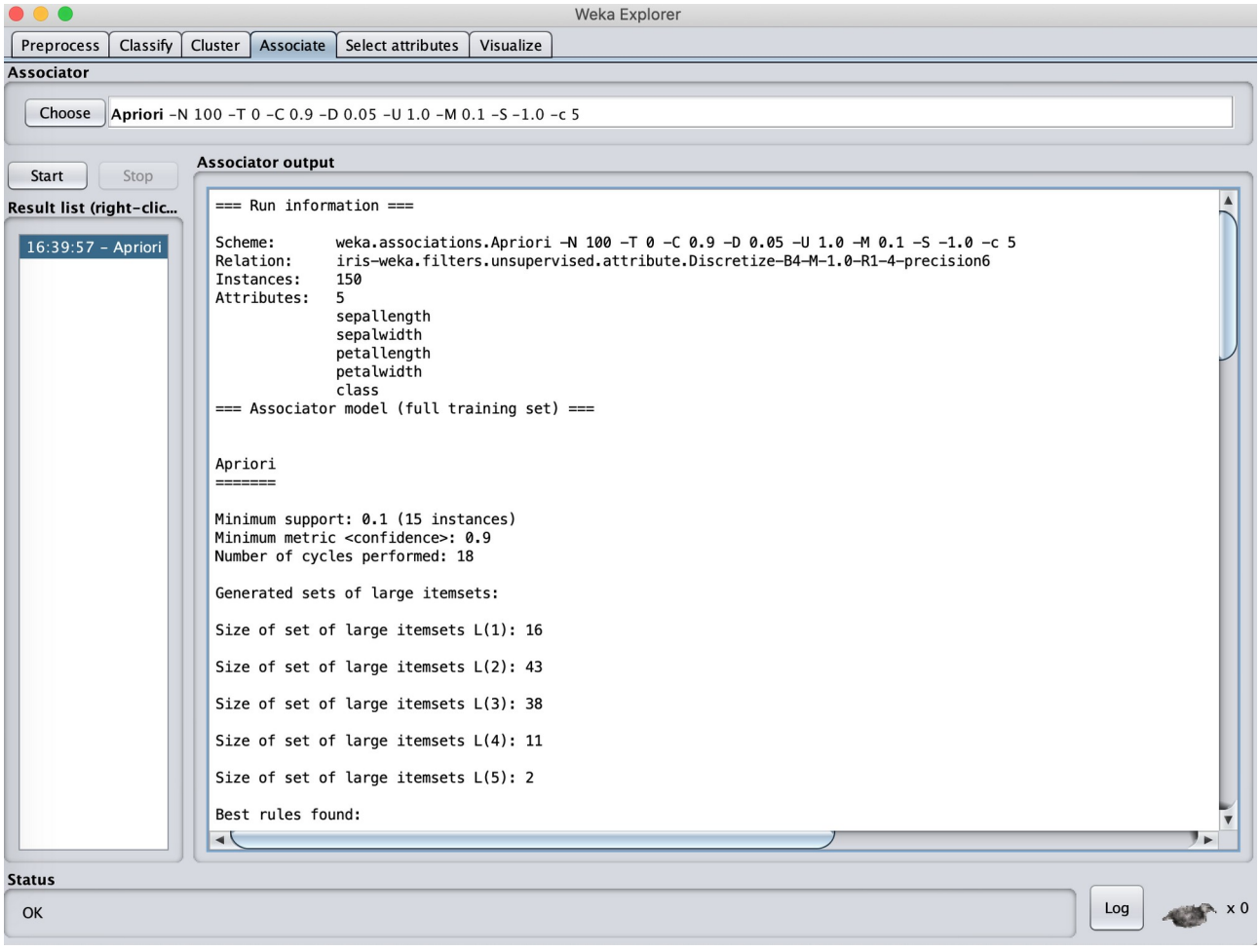
In the analysis above, we use k-means method to create cluster, and we use the FarthestFirst algorithm this time, and set the number of cluster to 3 as usual. Here are the results.



Analysis: From the results, we can see this method give lower support compare to the k-means clustering method, and by analysis its confidence level, we can also see the results provided fewer rules regarding to different clusters. And this maybe because the dataset have extreme values which affects the clustering process and then leads to worse association results.

Different number of clusters


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 4 -A "weka.core.EuclideanDistance -R first
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
16:30:49 - FarthestFirst
16:38:46 - SimpleKMeans
Clusterer output
petallength '(3.95-5.425]' '(3.95-5.425]' '(2.475-3.95]' '(5.425-inf)' '(-inf-2.475]
petalwidth '(-inf-0.7]' '(1.3-1.9]' '(0.7-1.3]' '(1.9-inf)' '(-inf-0.7]
Time taken to build model (full training data) : 0 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 61 (41%)
1 9 (6%)
2 30 (20%)
3 50 (33%)
Class attribute: class
Classes to Clusters:
0 1 2 3 <-- assigned to cluster
0 0 0 50 | Iris-setosa
41 9 0 0 | Iris-versicolor
20 0 30 0 | Iris-virginica
Cluster 0 <-- Iris-versicolor
Cluster 1 <-- No class
Cluster 2 <-- Iris-virginica
Cluster 3 <-- Iris-setosa
Incorrectly clustered instances : 29.0 19.3333 %
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -N 100 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c 5
Start
Stop
Result list (right-clic...
16:39:57 - Apriori
Associator output
=== Run information ===
Scheme: weka.associations.Apriori -N 100 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c 5
Relation: iris-weka.filters.unsupervised.attribute.Discretize-B4-M-1.0-R1-4-precision6
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
class
=== Associator model (full training set) ===
Apriori
=======
Minimum support: 0.1 (15 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18
Generated sets of large itemsets:
Size of set of large itemsets L(1): 16
Size of set of large itemsets L(2): 43
Size of set of large itemsets L(3): 38
Size of set of large itemsets L(4): 11
Size of set of large itemsets L(5): 2
Best rules found:
Status
OK
Log
x 0

Analysis: We choose different number of clusters like 4, 5 and 6. And the 4 clusters is the best among these three clusterings. Result has shown in the plot above.

By analysis them, it’s clearly illustrate that when cluster equals 3, the association analysis give the best output based on the association support and confidence level, also, it takes the least cycles to find the rules naturally.

This may because the original data has three different class which are quite dissimilar from each other, so when we analysis association relations between them, it will shows the true situation behind this dataset, if we want to increase or decrease the number of cluster, it will have fewer association rules returned.