

732A75 Data Mining Lab-2

Lakshidaa Saigiridharan (laksa656) and Sridhar Adhikarla (sriad858)

3/4/2019

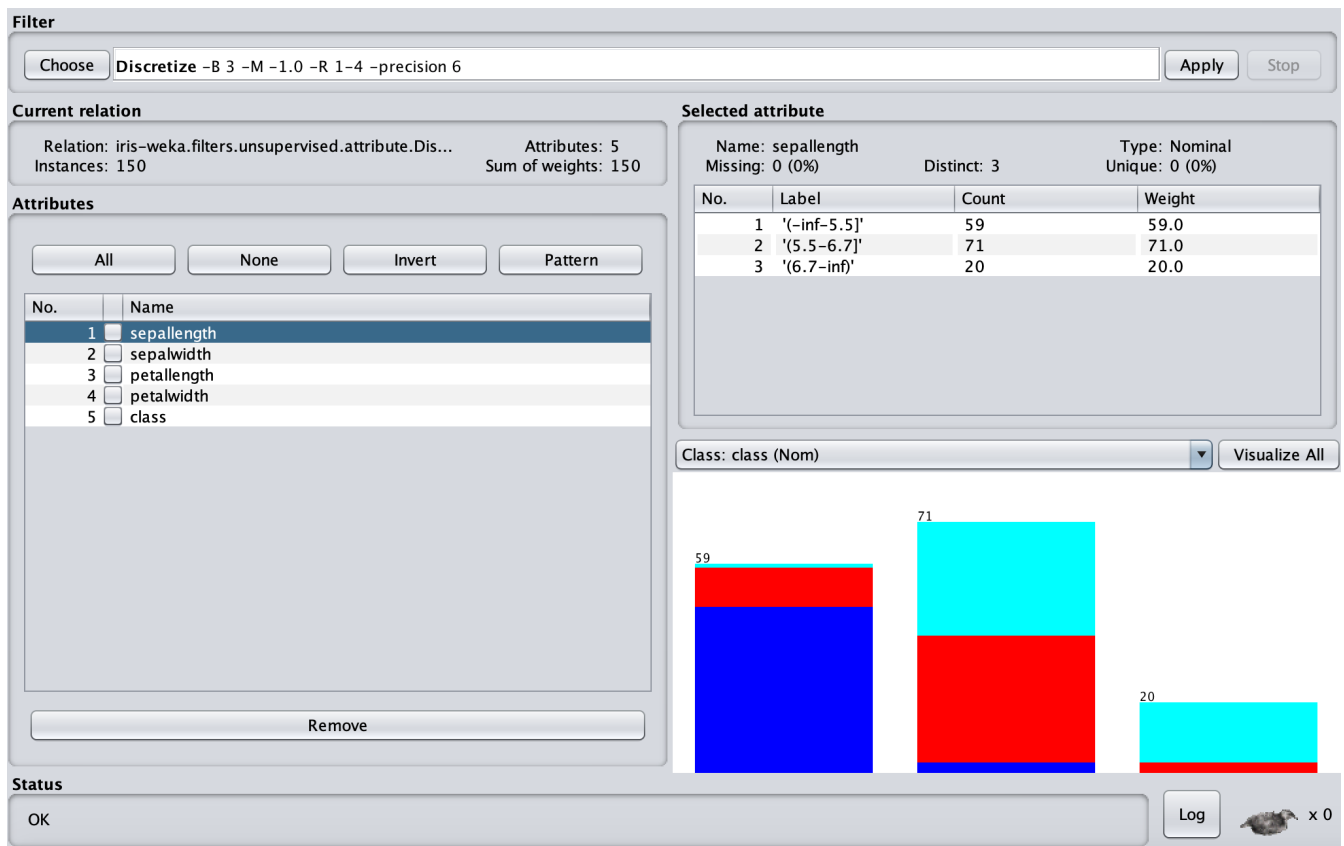
Association Analysis

Dataset :

The dataset used is the Iris dataset. The dataset consists of 50 samples from each of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample, they are the length and the width of sepal and petal.

Clustering :

We discretize the data before starting the mining process with number of bins set to 3.



We apply the SimpleKMeans clusterer to the data with 3 clusters (since we know there are 3 types of Iris flowers) and seed value 10. We get the following output :

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 3 -A "weka.core.EuclideanDistance" -R first-last" -I 500 -num-slots 1 -S 10

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)

- 16:31:12 - SimpleKMeans
- 16:34:40 - SimpleKMeans
- 16:34:47 - SimpleKMeans
- 16:34:55 - SimpleKMeans
- 16:34:58 - SimpleKMeans
- 16:36:07 - SimpleKMeans
- 17:00:05 - SimpleKMeans

Clusterer output

kMeans

Number of iterations: 3
Within cluster sum of squared errors: 96.0

Initial starting points (random):

Cluster 0: '\(5.5-6.7)\', '\(2.8-3.6)\', '\(2.966667-4.933333)\', '\(0.9-1.7)\'
Cluster 1: '\(6.7-inf)\', '\(2.8-3.6)\', '\(4.933333-inf)\', '\(1.7-inf)\'
Cluster 2: '\(-inf-5.5)\', '\(3.6-inf)\', '\(-inf-2.966667)\', '\(-inf-0.9)\'

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Full Data (150.0)	Cluster# 0 (55.0)	1 (45.0)	2 (50.0)
sepalwidth	'(5.5-6.7)'	'(5.5-6.7)'	'(5.5-6.7)'	'(-inf-5.5)'
sepalwidth	'(2.8-3.6)'	'(-inf-2.8)'	'(2.8-3.6)'	'(2.8-3.6)'
petalwidth	'(2.966667-4.933333)'	'(2.966667-4.933333)'	'(4.933333-inf)'	'(-inf-2.966667)'
petalwidth	'(0.9-1.7)'	'(0.9-1.7)'	'(1.7-inf)'	'(-inf-0.9)'

Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

Cluster	Count	Percentage
0	55	(37%)
1	45	(30%)
2	50	(33%)

Class attribute: class
Classes to Clusters:

Cluster	Count	Assigned to cluster
0	50	Iris-setosa
48	2	Iris-versicolor
7	43	Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-virginica
Cluster 2 <-- Iris-setosa

Incorrectly clustered instances : 9.0 6 %

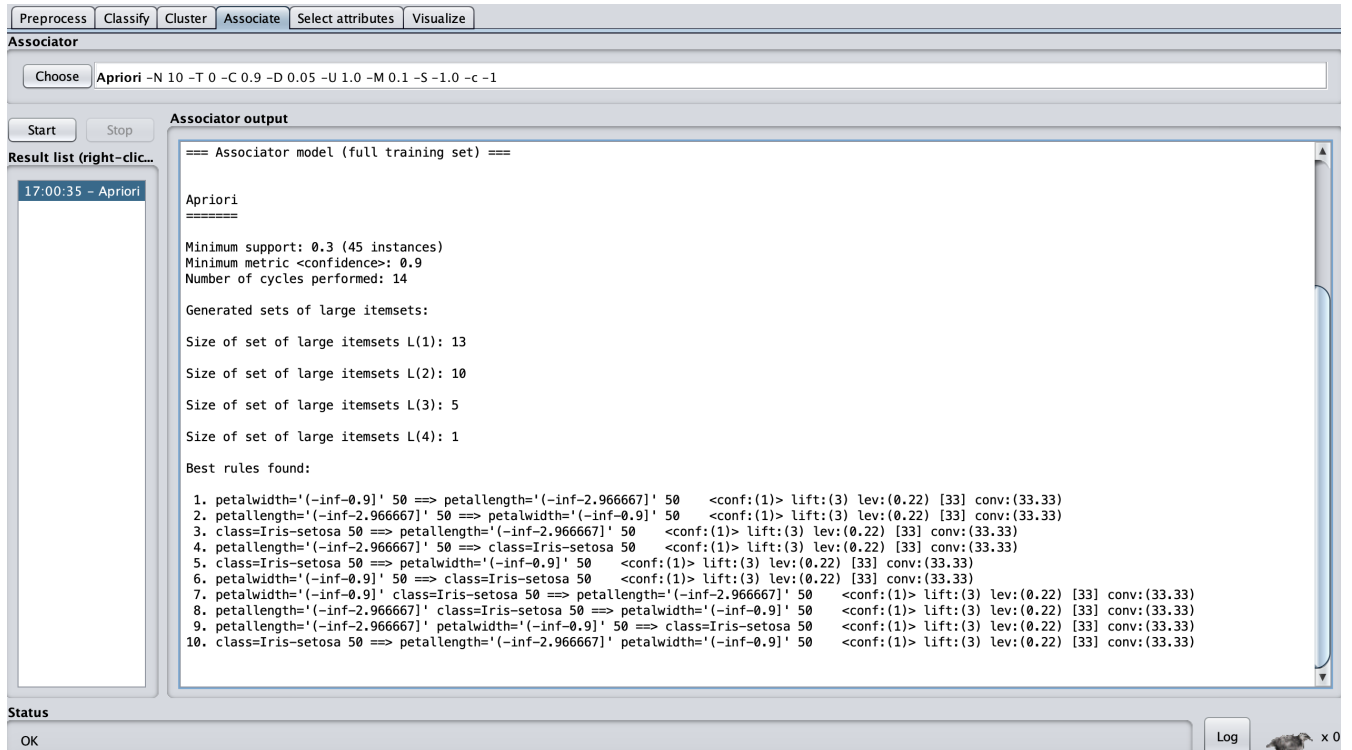
Status

OK Log x 0

Therefore, from these results it can be seen that k-means does a good job in classifying the species. Iris-setosa is seen to be correctly clustered Whereas, there are a few misclassifications for Iris-versicolor and Iris-virginica. The error rate obtained is 6% with an accuracy of 94% which is pretty good for a clusterer

Association analysis :

We perform association analysis by using the Apriori algorithm. The association rules obtained contain numbers on the right and left hand sides of the conjunctions of attribute-value pairs of each rule. That number indicates the support of the determinant and of the determinant plus the consequent.



The screenshot shows the Weka GUI with the 'Associate' tab selected. The 'Apriori' algorithm is chosen with parameters: -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1. The 'Associator output' pane displays the following results:

```
=== Associator model (full training set) ===

Apriori
=====

Minimum support: 0.3 (45 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 13
Size of set of large itemsets L(2): 10
Size of set of large itemsets L(3): 5
Size of set of large itemsets L(4): 1

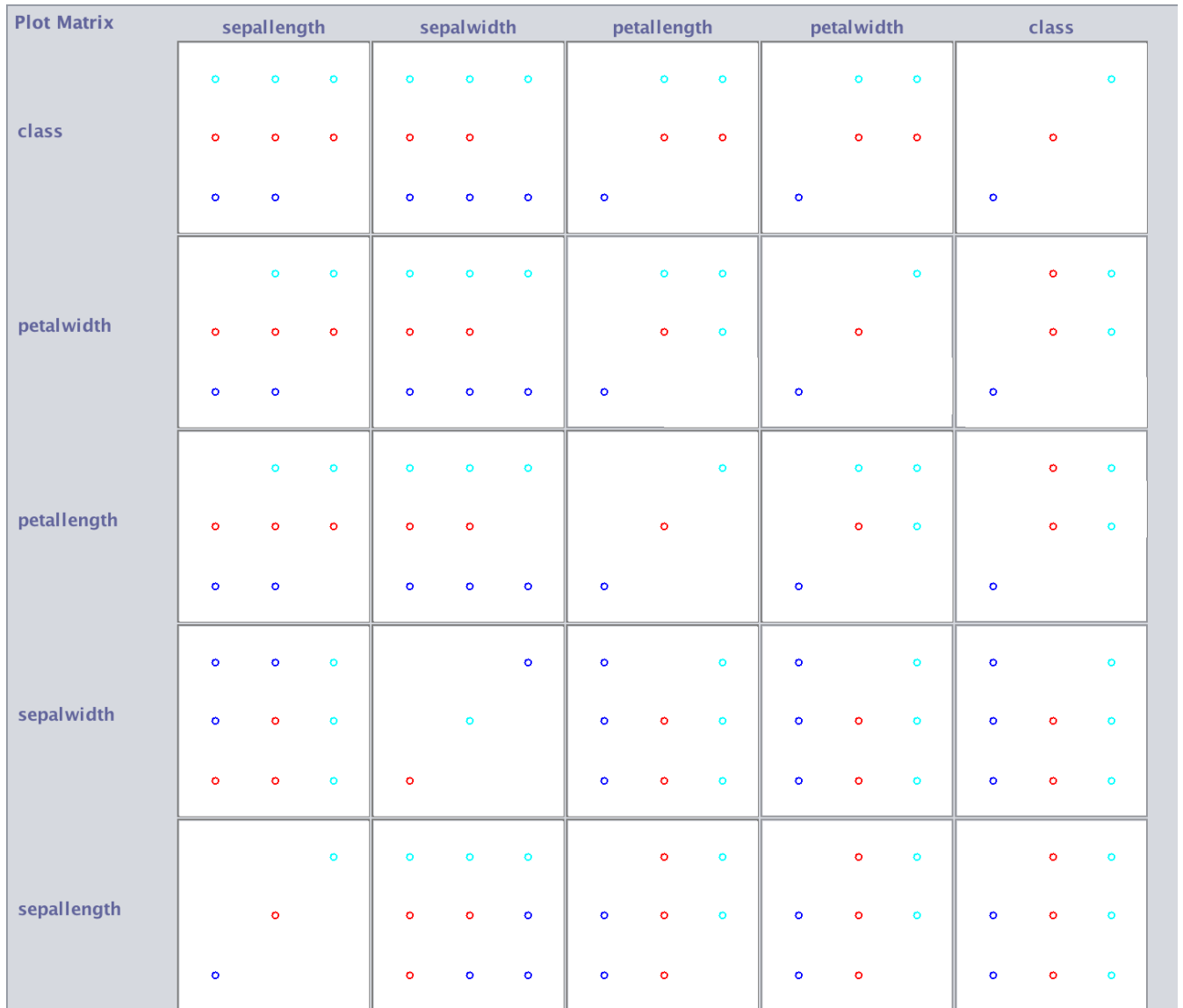
Best rules found:

1. petalwidth='(-inf-0.9]' 50 ==> petallength='(-inf-2.966667]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
2. petallength='(-inf-2.966667]' 50 ==> petalwidth='(-inf-0.9]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
3. class=Iris-setosa 50 ==> petallength='(-inf-2.966667]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
4. petallength='(-inf-2.966667]' 50 ==> class=Iris-setosa 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
5. class=Iris-setosa 50 ==> petalwidth='(-inf-0.9]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
6. petalwidth='(-inf-0.9]' 50 ==> class=Iris-setosa 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
7. petalwidth='(-inf-0.9]' class=Iris-setosa 50 ==> petallength='(-inf-2.966667]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
8. petallength='(-inf-2.966667]' class=Iris-setosa 50 ==> petalwidth='(-inf-0.9]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
9. petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50 ==> class=Iris-setosa 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
10. class=Iris-setosa 50 ==> petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
```

Therefore, it can be noted that the minimum support is 0.3 with a minimum metric of 0.9. The metric is of type confidence.

Visualization :

We then visualize the data. This is done by obtaining a plot matrix such that the data is crosstabulated for each pair of attributes.



As we have set the number of clusters to 3 in the initial steps, we see the 3 clusters in the above plot matrix represented by 3 different colours (cluster1 = purple, cluster2 = red, cluster3 = blue).

Describing clustering through association analysis :

We create a new attribute that represents the cluster label assigned to each instance. We set the number of clusters to 3 and the number of bins to 3. We apply the SimpleKMeans clustering algorithm again.

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 3 -A 'weka.core.EuclideanDistance -R first-last' -I 500 -num-slots 1 -S 10

Cluster mode

☐ Use training set
☐ Supplied test set Set...
☐ Percentage split % 66
☒ Classes to clusters evaluation
 (Nom) cluster
☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 16:31:12 - SimpleKMeans
- 16:34:40 - SimpleKMeans
- 16:34:47 - SimpleKMeans
- 16:34:55 - SimpleKMeans
- 16:34:58 - SimpleKMeans
- 16:36:07 - SimpleKMeans
- 17:00:05 - SimpleKMeans
- 17:14:51 - SimpleKMeans**

Clusterer output

kMeans

Number of iterations: 3
Within cluster sum of squared errors: 96.0

Initial starting points (random):

Cluster 0: '[5.5-6.7]','','(2.8-3.6)','','(2.966667-4.933333)','','(0.9-1.7)''
Cluster 1: '[6.7-inf]','','(2.8-3.6)','','(4.933333-inf)','','(1.7-inf)''
Cluster 2: '[(-inf-5.5)','','(3.6-inf)','','(-inf-2.966667)','','(-inf-0.9)''

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Full Data (150.0)	Cluster# 0 (55.0)	1 (45.0)	2 (50.0)
sepalwidth	'(5.5-6.7)'	'(5.5-6.7)'	'(5.5-6.7)'	'(-inf-5.5)'
sepalwidth	'(2.8-3.6)'	'(-inf-2.8)'	'(2.8-3.6)'	'(2.8-3.6)'
petalwidth	'(2.966667-4.933333)'	'(2.966667-4.933333)'	'(4.933333-inf)'	'(-inf-2.966667)'
petalwidth	'(0.9-1.7)'	'(0.9-1.7)'	'(1.7-inf)'	'(-inf-0.9)'

Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

Status

OK Log x 0

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 3 -A 'weka.core.EuclideanDistance -R first-last' -I 500 -num-slots 1 -S 10

Cluster mode

☐ Use training set
☐ Supplied test set Set...
☐ Percentage split % 66
☒ Classes to clusters evaluation
 (Nom) cluster
☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 16:31:12 - SimpleKMeans
- 16:34:40 - SimpleKMeans
- 16:34:47 - SimpleKMeans
- 16:34:55 - SimpleKMeans
- 16:34:58 - SimpleKMeans
- 16:36:07 - SimpleKMeans
- 17:00:05 - SimpleKMeans
- 17:14:51 - SimpleKMeans**

Clusterer output

Attribute	Full Data (150.0)	Cluster# 0 (55.0)	1 (45.0)	2 (50.0)
sepalwidth	'(5.5-6.7)'	'(5.5-6.7)'	'(5.5-6.7)'	'(-inf-5.5)'
sepalwidth	'(2.8-3.6)'	'(-inf-2.8)'	'(2.8-3.6)'	'(2.8-3.6)'
petalwidth	'(2.966667-4.933333)'	'(2.966667-4.933333)'	'(4.933333-inf)'	'(-inf-2.966667)'
petalwidth	'(0.9-1.7)'	'(0.9-1.7)'	'(1.7-inf)'	'(-inf-0.9)'

Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

Cluster	Count	Percentage
0	55	(37%)
1	45	(30%)
2	50	(33%)

Class attribute: cluster
Classes to Clusters:

Cluster	Count	Assigned to
0	1	cluster1
55	0	cluster1
0	45	cluster2
0	0	cluster3

Cluster 0 <-- cluster1
Cluster 1 <-- cluster2
Cluster 2 <-- cluster3

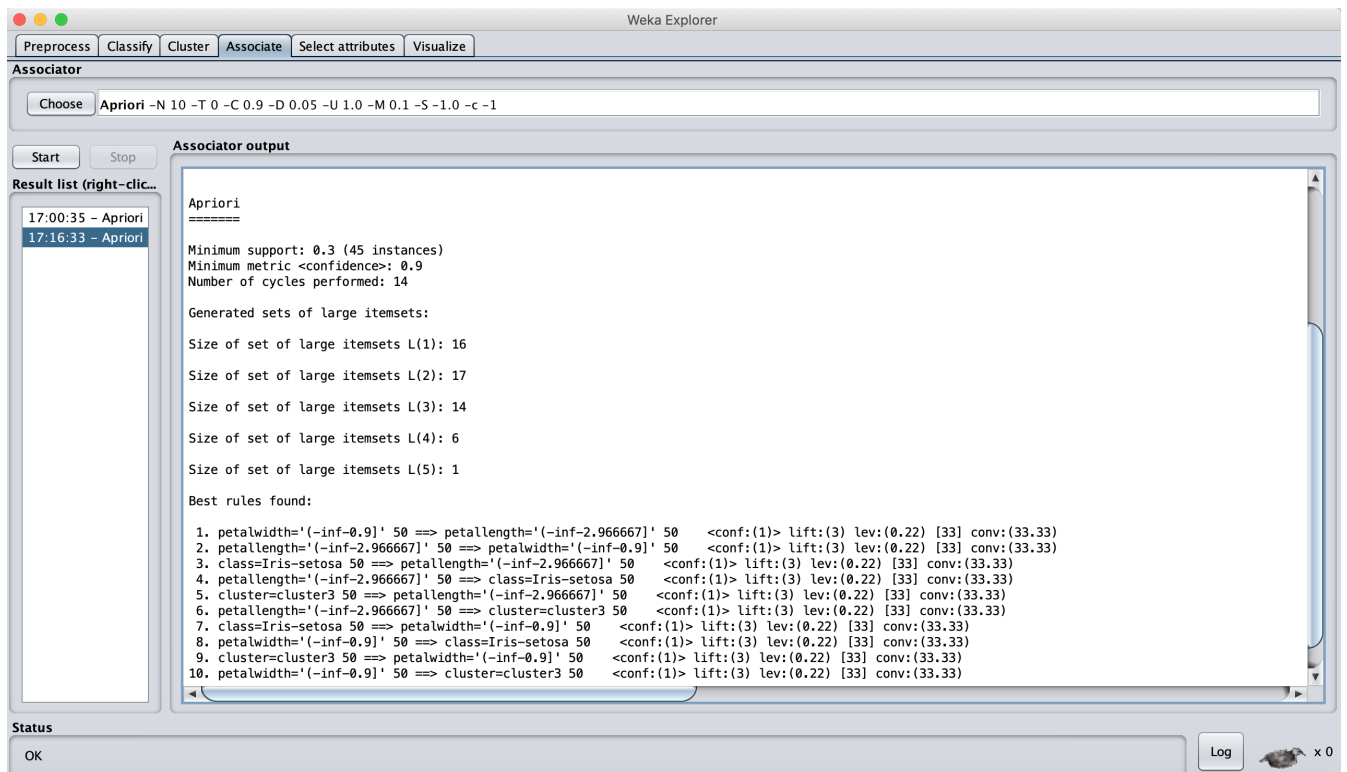
Incorrectly clustered instances : 0.0 0 %

Status

OK Log x 0

From these results, it can be seen that the clusterer performs extremely well with an accuracy of 100%. Therefore, our classifier performs much better on adding the new cluster attribute.

The association rules obtained for this are :



From this, we see that the rules that are accurate and such that the antecedent does not contain the class attribute and the consequent only contains the cluster attribute are :

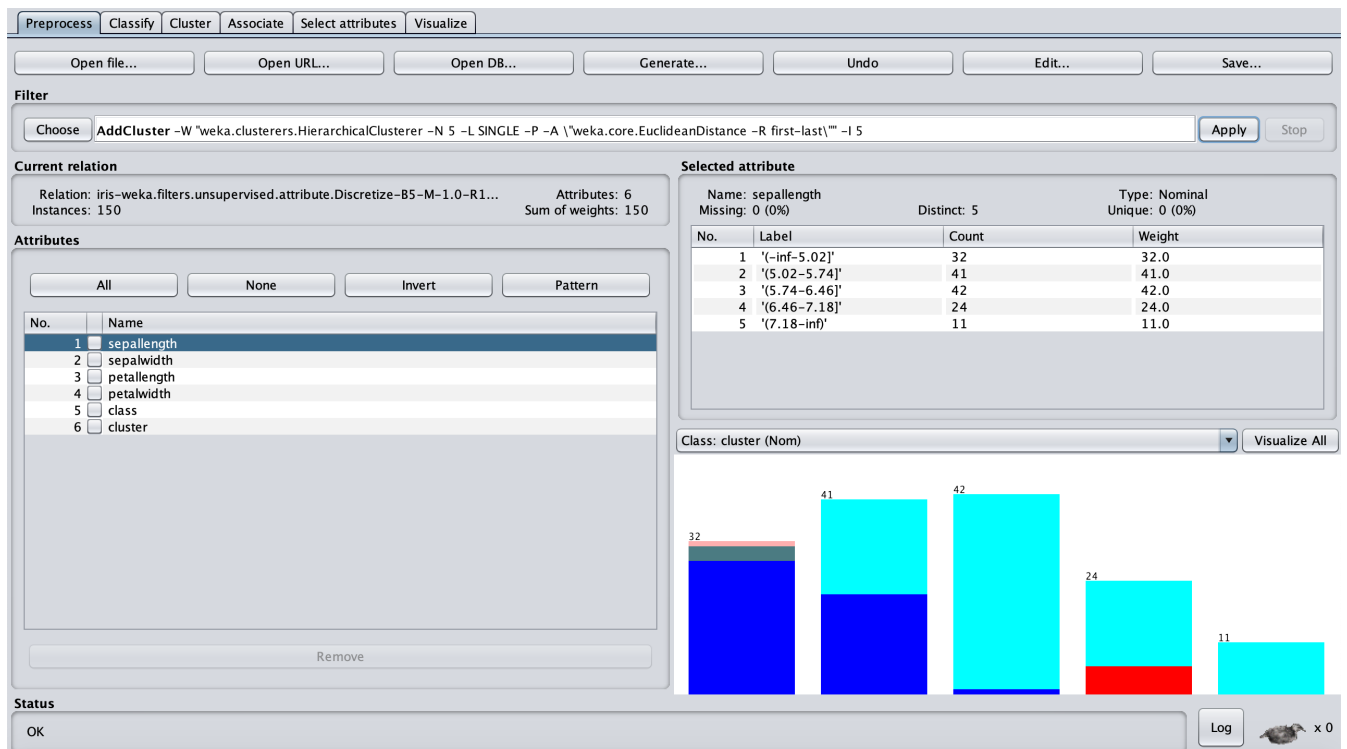
- 1) petalwidth='(-inf-0.9]' 50 ==> cluster=cluster3 50
- 2) petalwidth='(-inf-0.9]' 50 ==> cluster=cluster3 50

The above exercise is then repeated for different combinations of clustering algorithms, number of clusters and number of bins.

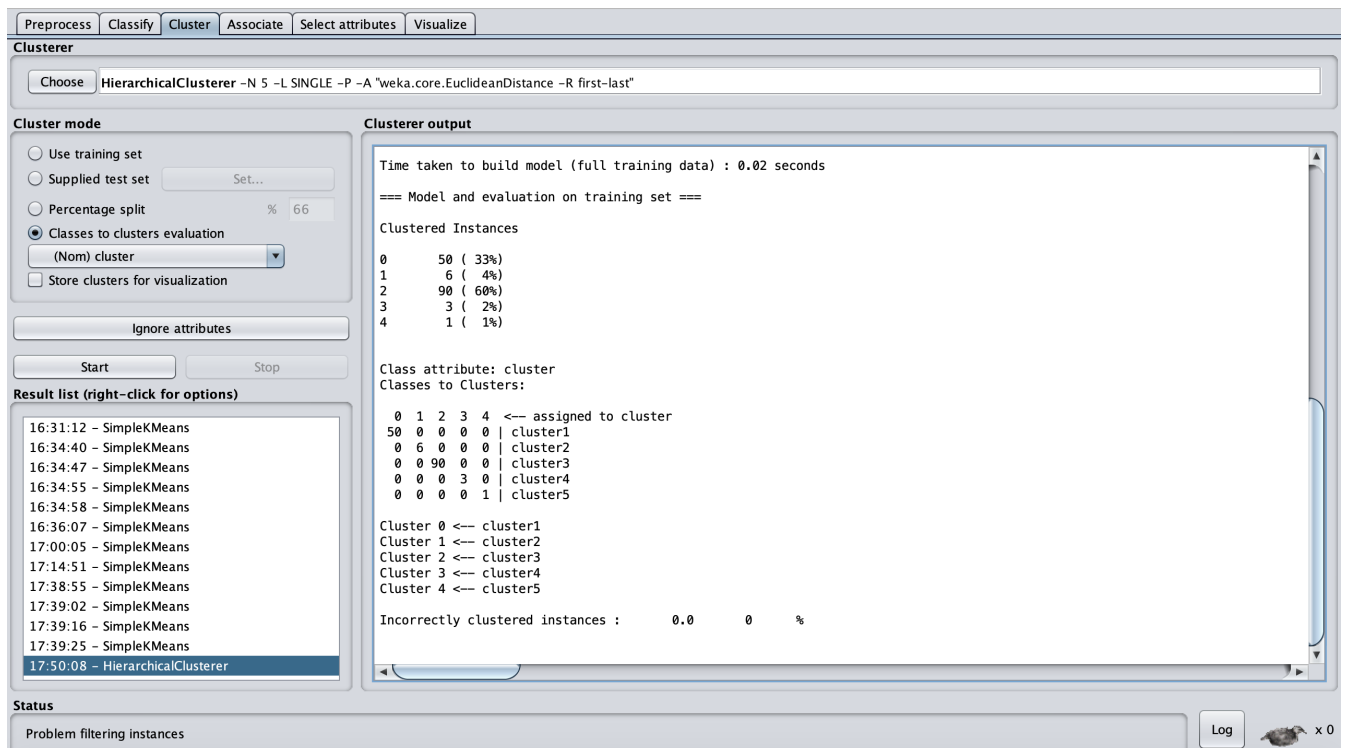
Combination 1 :

Clusterer : Hierarchical clustering, **Number of clusters** : 5, **Number of bins** : 5

The discretized data is represented as :

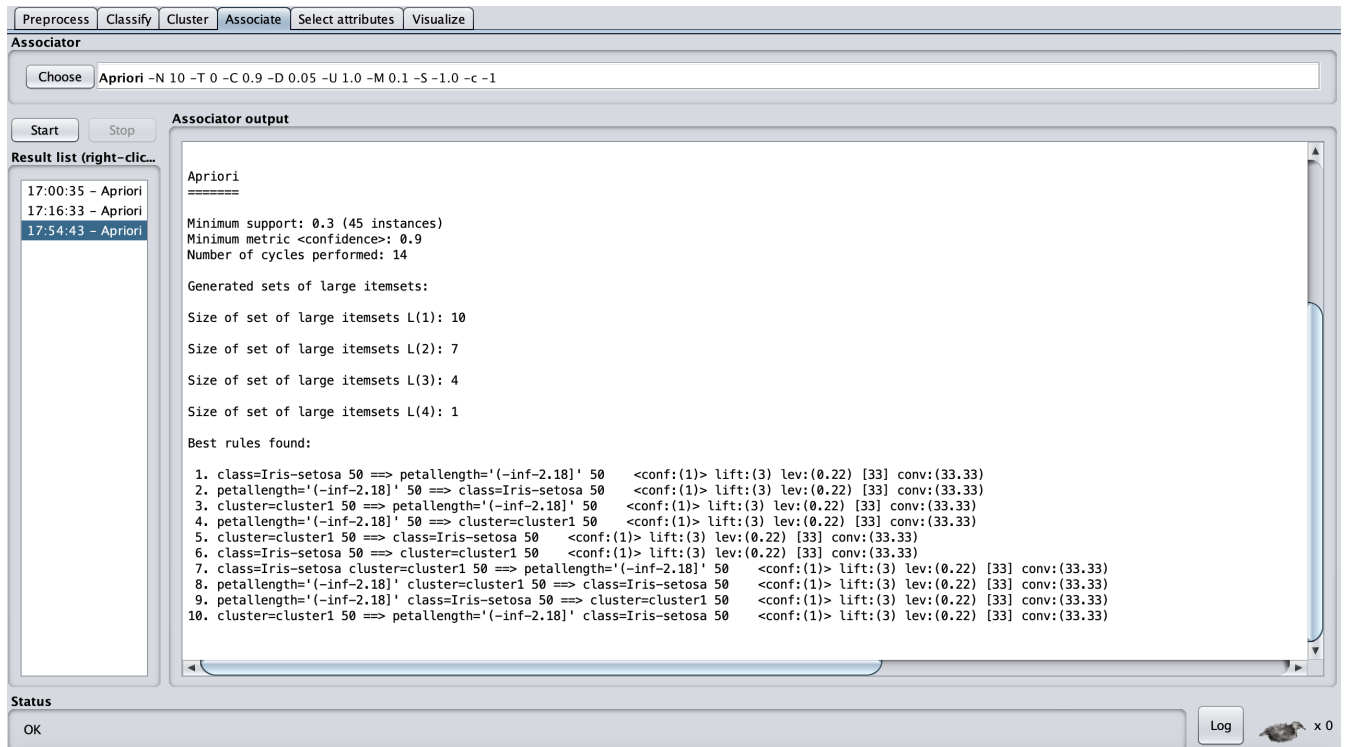


On running the hierachical cluster, we obtain the following clusters :



It can be seen that most of the attributes lie in the 3rd cluster. The last (5th) cluster consists of only one attribute.

We then run the apriori algorithm to obtain the association rules :



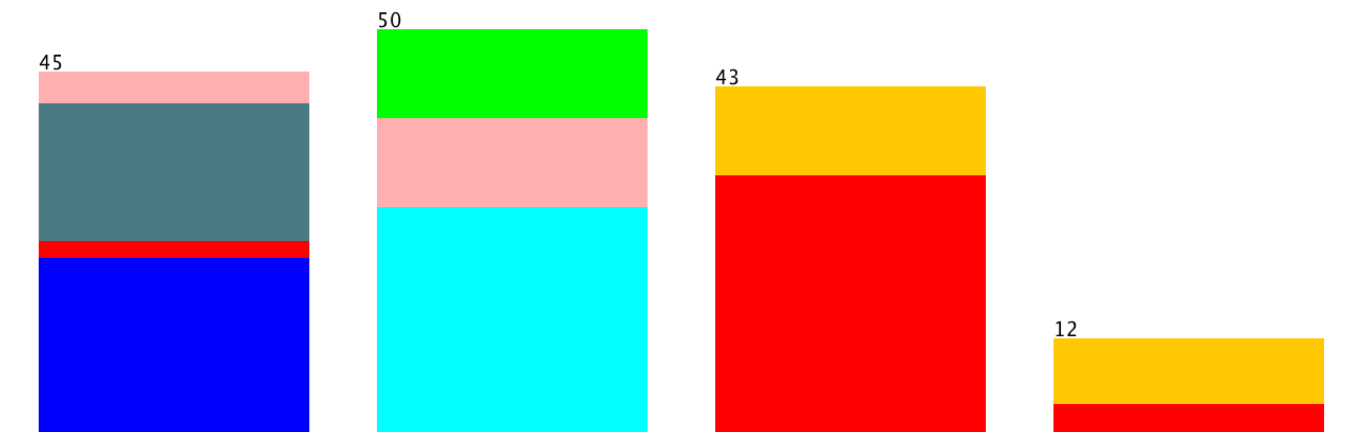
From this, we see that the rules that are accurate and such that the antecedent does not contain the class attribute and the consequent only contains the cluster attribute are :

- 1) petalength='(-inf-2.18]' 50 ==> cluster=cluster1 50

Combination 2 :

Clusterer : EM, Number of clusters : 7, Number of bins : 4

The discretized data is represented as :



On running the EM clusterer, we obtain the following clusters :

Preprocess Classify Cluster Associate Select attributes Visualize

Clusterer

Choose EM -I 100 -N 7 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

Cluster mode

☐ Use training set

☐ Supplied test set Set...

☐ Percentage split % 66

☒ Classes to clusters evaluation

(Nom) cluster

☐ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 16:34:47 - SimpleKMeans
- 16:34:55 - SimpleKMeans
- 16:34:58 - SimpleKMeans
- 16:36:07 - SimpleKMeans
- 17:00:05 - SimpleKMeans
- 17:14:51 - SimpleKMeans
- 17:38:55 - SimpleKMeans
- 17:39:02 - SimpleKMeans
- 17:39:16 - SimpleKMeans
- 17:39:25 - SimpleKMeans
- 17:50:08 - HierarchicalClusterer
- 19:10:53 - SimpleKMeans
- 19:39:45 - EM

Clusterer output

```

0 22 ( 15%)
1 38 ( 25%)
2 28 ( 19%)
3 17 ( 11%)
4 15 ( 10%)
5 11 ( 7%)
6 19 ( 13%)

Log likelihood: -3.5482

Class attribute: cluster
Classes to Clusters:

0 1 2 3 4 5 6 <-- assigned to cluster
22 0 0 0 0 0 0 | cluster1
0 38 0 0 0 0 0 | cluster2
0 0 28 0 0 0 0 | cluster3
0 0 0 17 0 0 0 | cluster4
0 0 0 0 15 0 0 | cluster5
0 0 0 0 0 11 0 | cluster6
0 0 0 0 0 0 19 | cluster7

Cluster 0 <-- cluster1
Cluster 1 <-- cluster2
Cluster 2 <-- cluster3
Cluster 3 <-- cluster4
Cluster 4 <-- cluster5
Cluster 5 <-- cluster6
Cluster 6 <-- cluster7

Incorrectly clustered instances : 0.0 0 %

```

Status

OK Log x 0

It can be seen that the attributes are distributed well among the 7 clusters.

We then run the apriori algorithm to obtain the association rules :

Preprocess Classify Cluster Associate Select attributes Visualize

Associator

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

Start Stop

Result list (right-click for options)

- 17:00:35 - Apriori
- 17:16:33 - Apriori
- 17:54:43 - Apriori
- 19:16:11 - Apriori
- 19:16:29 - Apriori
- 19:41:26 - Apriori
- 19:41:43 - Apriori

Associator output

```

==== cluster
==== Associator model (full training set) ====

Apriori
=====

Minimum support: 0.3 (45 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 9
Size of set of large itemsets L(2): 4
Size of set of large itemsets L(3): 1

Best rules found:

1. petalwidth='(-inf-0.7]' 50 ==> petallength='(-inf-2.475]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
2. petallength='(-inf-2.475]' 50 ==> petalwidth='(-inf-0.7]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
3. class=Iris-setosa 50 ==> petallength='(-inf-2.475]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
4. petallength='(-inf-2.475]' 50 ==> class=Iris-setosa 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
5. class=Iris-setosa 50 ==> petalwidth='(-inf-0.7]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
6. petalwidth='(-inf-0.7]' 50 ==> class=Iris-setosa 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
7. petalwidth='(-inf-0.7]' class=Iris-setosa 50 ==> petallength='(-inf-2.475]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
8. petallength='(-inf-2.475]' class=Iris-setosa 50 ==> petalwidth='(-inf-0.7]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
9. petallength='(-inf-2.475]' petalwidth='(-inf-0.7]' 50 ==> class=Iris-setosa 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)
10. class=Iris-setosa 50 ==> petallength='(-inf-2.475]' petalwidth='(-inf-0.7]' 50 <conf:(1)> lift:(3) lev:(0.22) [33] conv:(33.33)

```

Status

OK Log x 0

Here, it can be observed that it is not possible to find any rules that are accurate such that the antecedent does not contain the class attribute and the consequent only contains the cluster attribute.

Final conclusions :

On trying different combinations of clustering algorithms, number of clusters and number of bins, it is observed that when we the number of clusters was set to 3, the classes were well distributed in the clusters such that none of the clusters were sparse. On increasing the number of clusters, it could be seen that some clusters had very few attributes within them. It was also observed that more accurate association rules were obtained when using SimpleKMeans as a clusterer.