Lab Report: Lab2 - Association Analysis -1

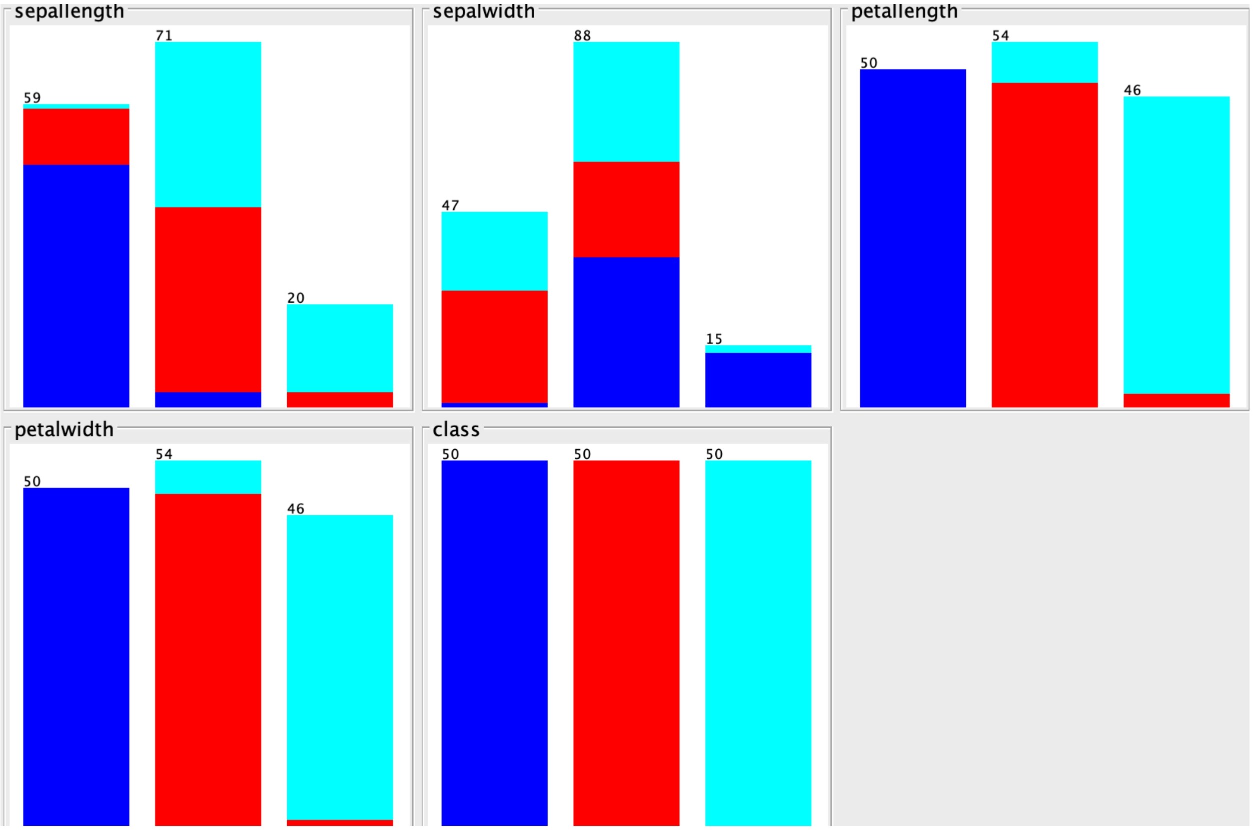
Dhanush Kumar Reddy Narayana Reddy (dhana004), Udaya Shanker Mohanan Nair (udamo524)

2025-03-26

Introduction

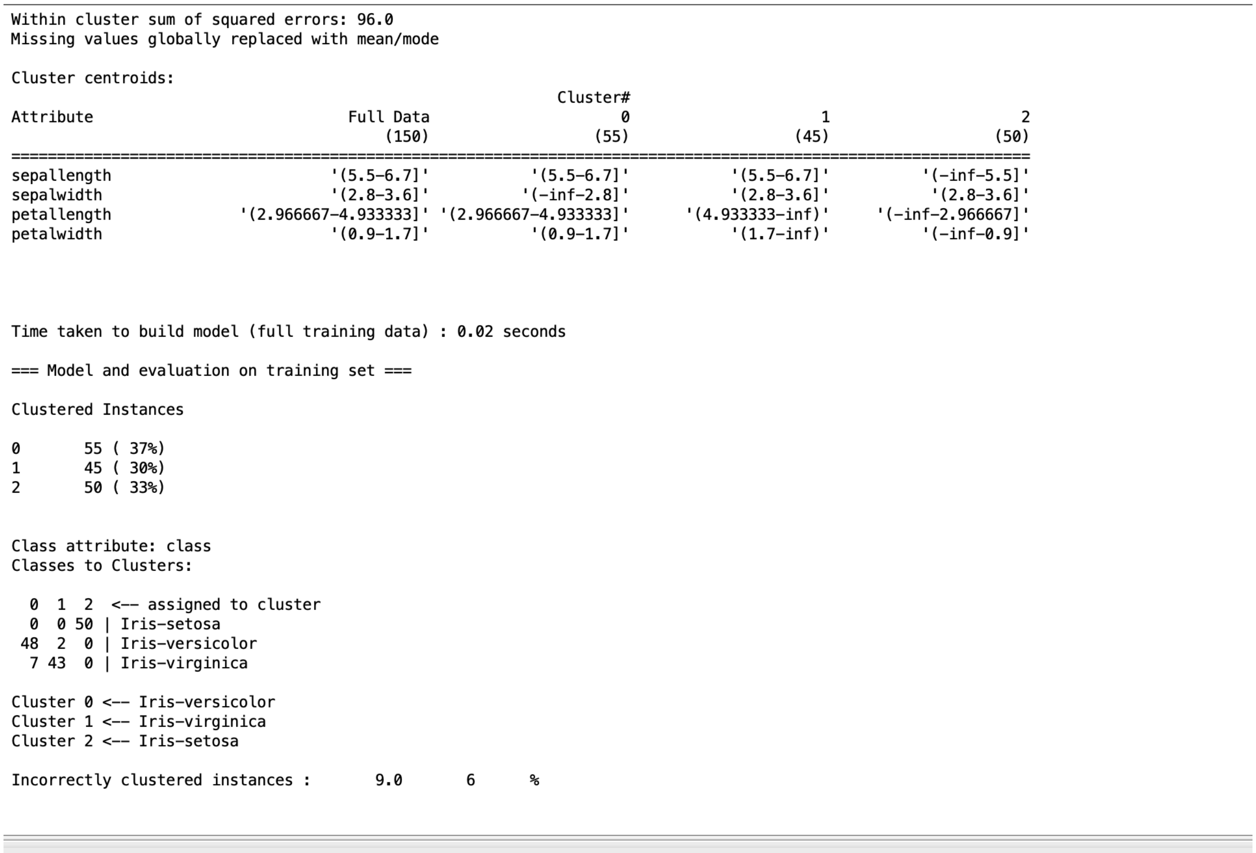
Load and Discretize the Dataset

Iris Dataset was loaded into weka. Inorder to do Association, we are using Apriori algorithm for which we requires attributes to be discrete(we use discretize filter in pre-process).



Clustering

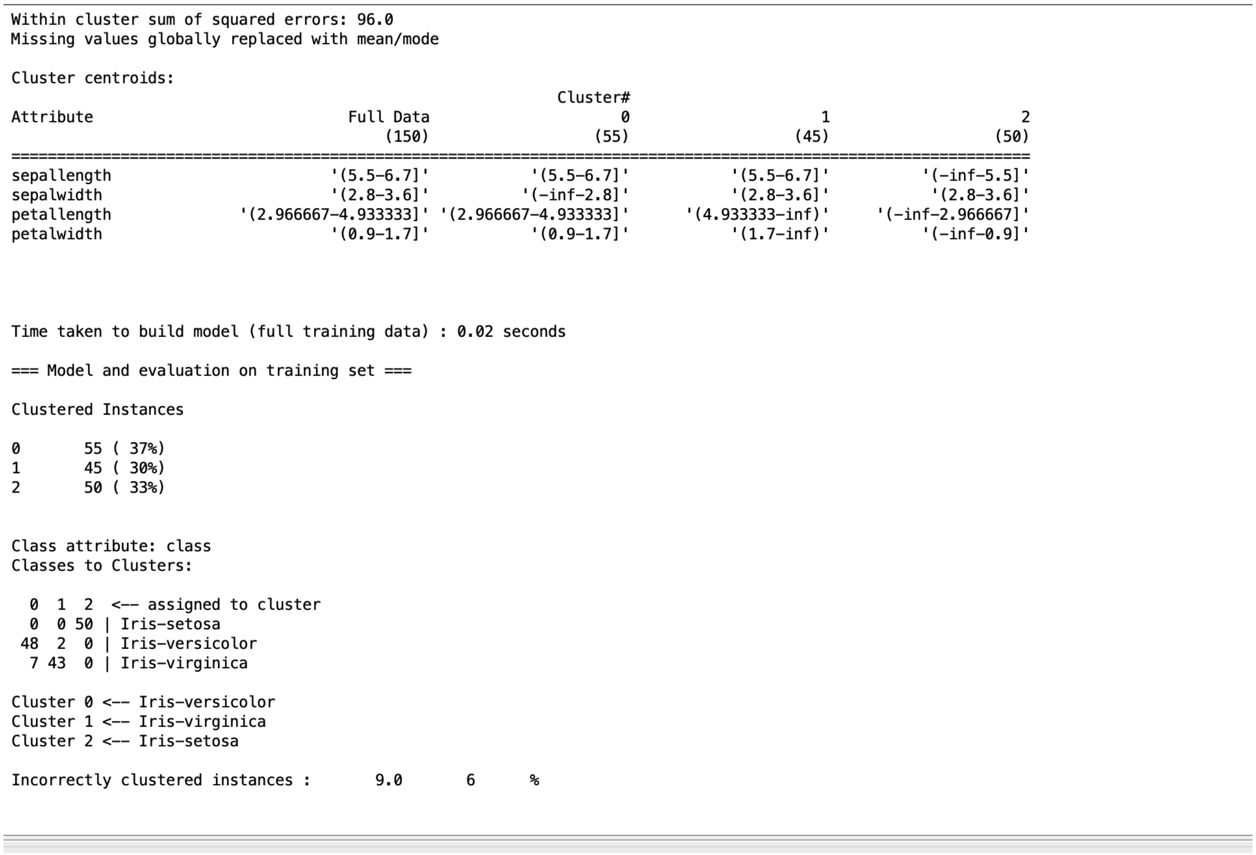
Here we are clustering the dataset using **SimpleKmeans** algorithm. Number of clusters is set to three and seed value is set to 10. For Cluster mode, we choose **Classes to clusters evaluation** to compare the results with the actual labels. Here we igonored the attribute class.



```
Within cluster sum of squared errors: 96.0
Missing values globally replaced with mean/mode

Cluster centroids:
                                                            Cluster#
Attribute                            Full Data                     0                     1                     2
                                         (150)                  (55)                  (45)                  (50)
================================================================================================================
sepallength                        '(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]'
petallength              '(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) : 0.02 seconds

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

Clustered Instances

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


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0  0 50 | Iris-setosa
 48  2  0 | Iris-versicolor
  7 43  0 | Iris-virginica

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

Incorrectly clustered instances :       9.0       6      %
```

Association Analysis

Now, Here we are doing associating(finding relationship) between values of the attribute and the class labels. For this purpose we employed, **Apriori** Algorithm for the dataset. The following results we got.

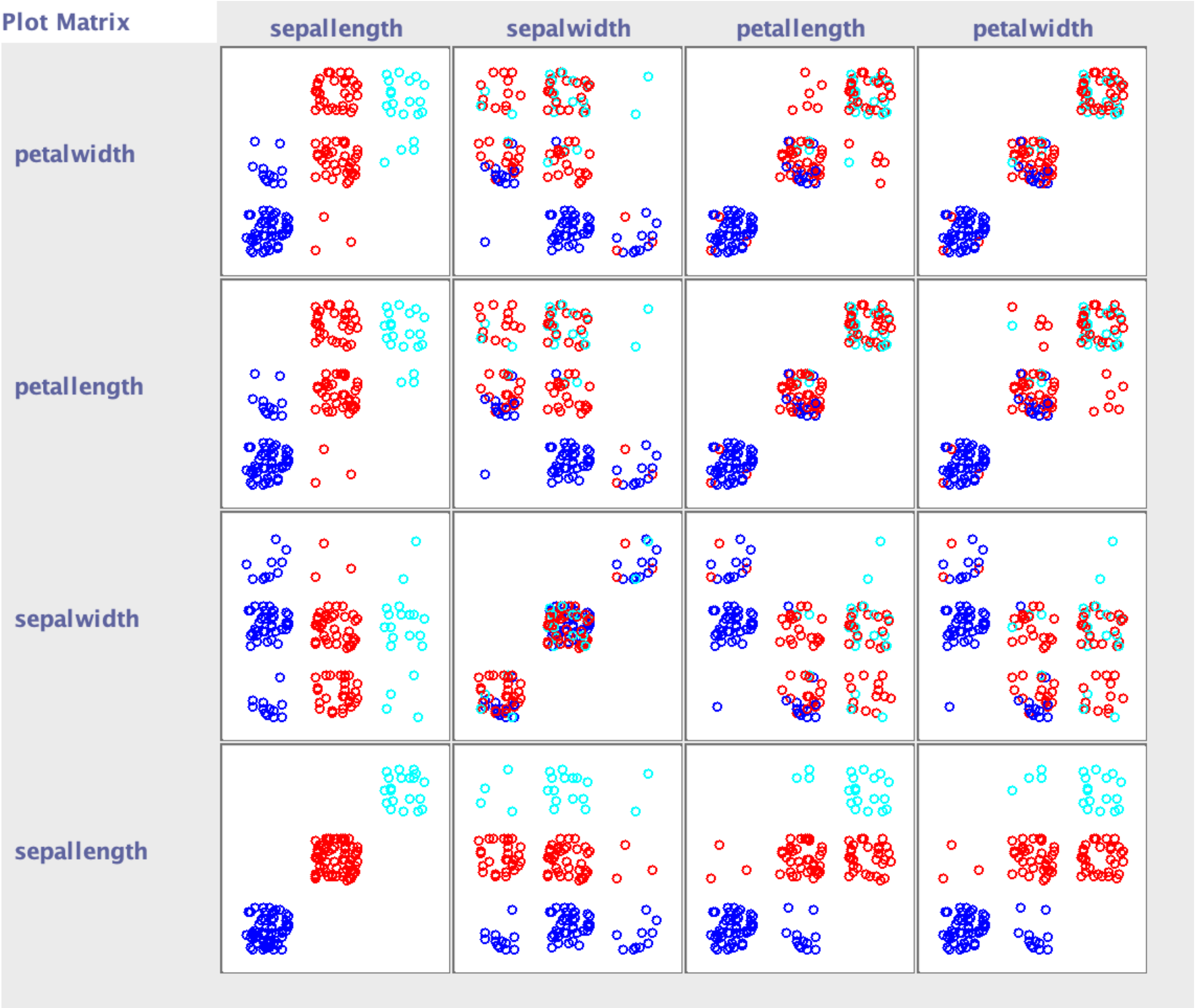


```
Within cluster sum of squared errors: 96.0
Missing values globally replaced with mean/mode

Cluster centroids:
                                                         Cluster#
Attribute                              Full Data                0                    1                    2
                                           (150)             (55)                 (45)                 (50)
=============================================================================================================
sepallength                          '(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]'
petallength                '(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) : 0.02 seconds

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

Clustered Instances

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


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0  0 50 | Iris-setosa
 48  2  0 | Iris-versicolor
  7 43  0 | Iris-virginica

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

Incorrectly clustered instances :       9.0       6      %
```

Visualization

Visualizing the dataset to find out the relationship between attributes and clusters.

Cluster 0 (Red) → Iris-versicolor Cluster 1 (Cyan) → Iris-virginica Cluster 2 (Blue) → Iris-setosa

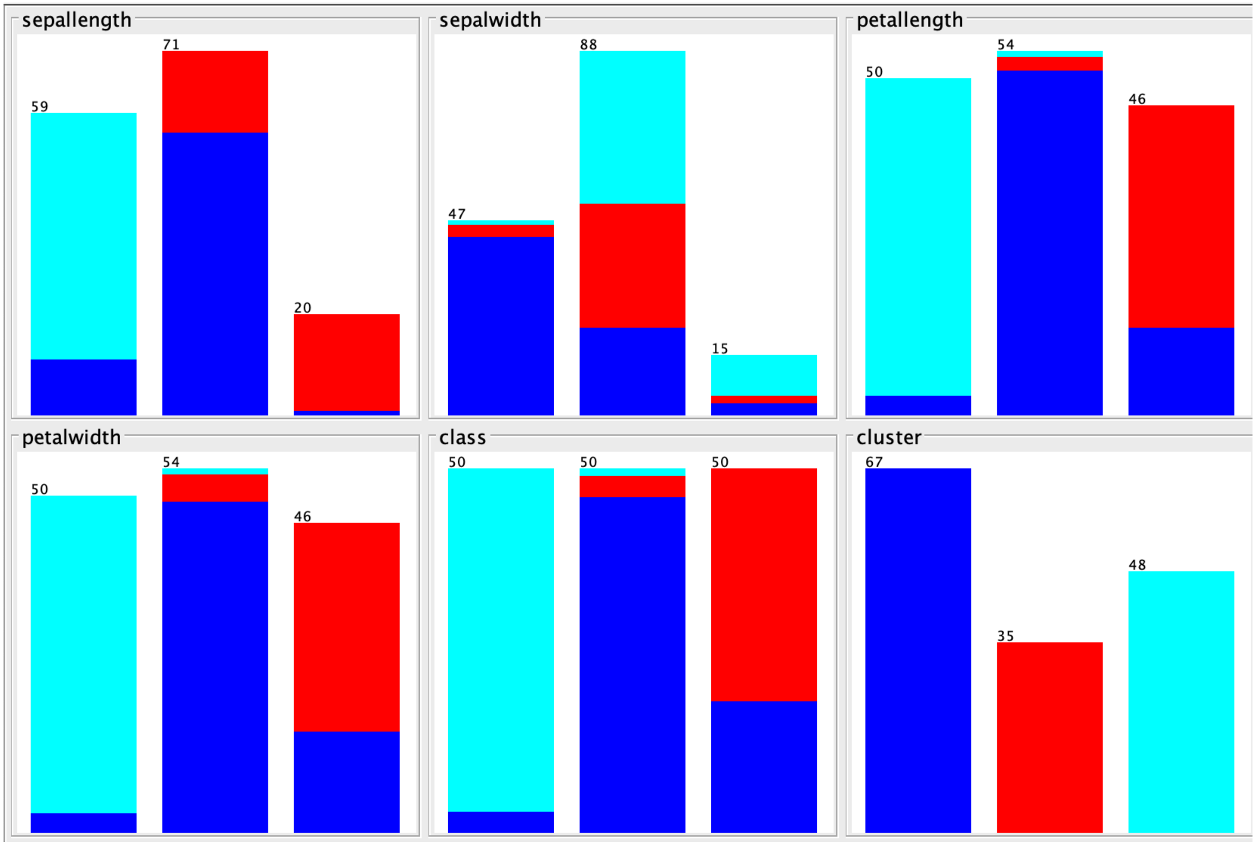


The following observations are observed:

1. Cluster 2 is clearly scene separated all other clusters in most of the combinations, whereas cluster 0 and cluster 1 seems to have some overlap in some of the combinations.
2. **Petal Length vs Petal Width** plot shows that there is a proper separation among all clusters compared to other combinations, even though Iris Versicolor and Iris Virginica are closer but distinguishable
3. **Sepal Length vs Sepal Width** plot shows that there is a high overlap between Iris Versicolor and Iris Virginica.

Describe clusters through Association

Here, in preprocess we gone use Filter Addcluster with number of clusters as 3 and seed as 10 with ignoring class attribute.



Then in associate tab, choose Apriori Algorithm, got the following results:

```
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): 15

Size of set of large itemsets L(2): 18

Size of set of large itemsets L(3): 14

Size of set of large itemsets L(4): 5

Size of set of large itemsets L(5): 1

Best rules found:

 1. petalwidth='(-inf-0.9]' 50 ==> petallength='(-inf-2.966667]' 50    conf:(1)
 2. petallength='(-inf-2.966667]' 50 ==> petalwidth='(-inf-0.9]' 50    conf:(1)
 3. class=Iris-setosa 50 ==> petallength='(-inf-2.966667]' 50    conf:(1)
 4. petallength='(-inf-2.966667]' 50 ==> class=Iris-setosa 50    conf:(1)
 5. class=Iris-setosa 50 ==> petalwidth='(-inf-0.9]' 50    conf:(1)
 6. petalwidth='(-inf-0.9]' 50 ==> class=Iris-setosa 50    conf:(1)
 7. petalwidth='(-inf-0.9]' class=Iris-setosa 50 ==> petallength='(-inf-2.966667]' 50    conf:(1)
 8. petallength='(-inf-2.966667]' class=Iris-setosa 50 ==> petalwidth='(-inf-0.9]' 50    conf:(1)
 9. petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50 ==> class=Iris-setosa 50    conf:(1)
10. class=Iris-setosa 50 ==> petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50    conf:(1)
```

Following Association rules are generated.

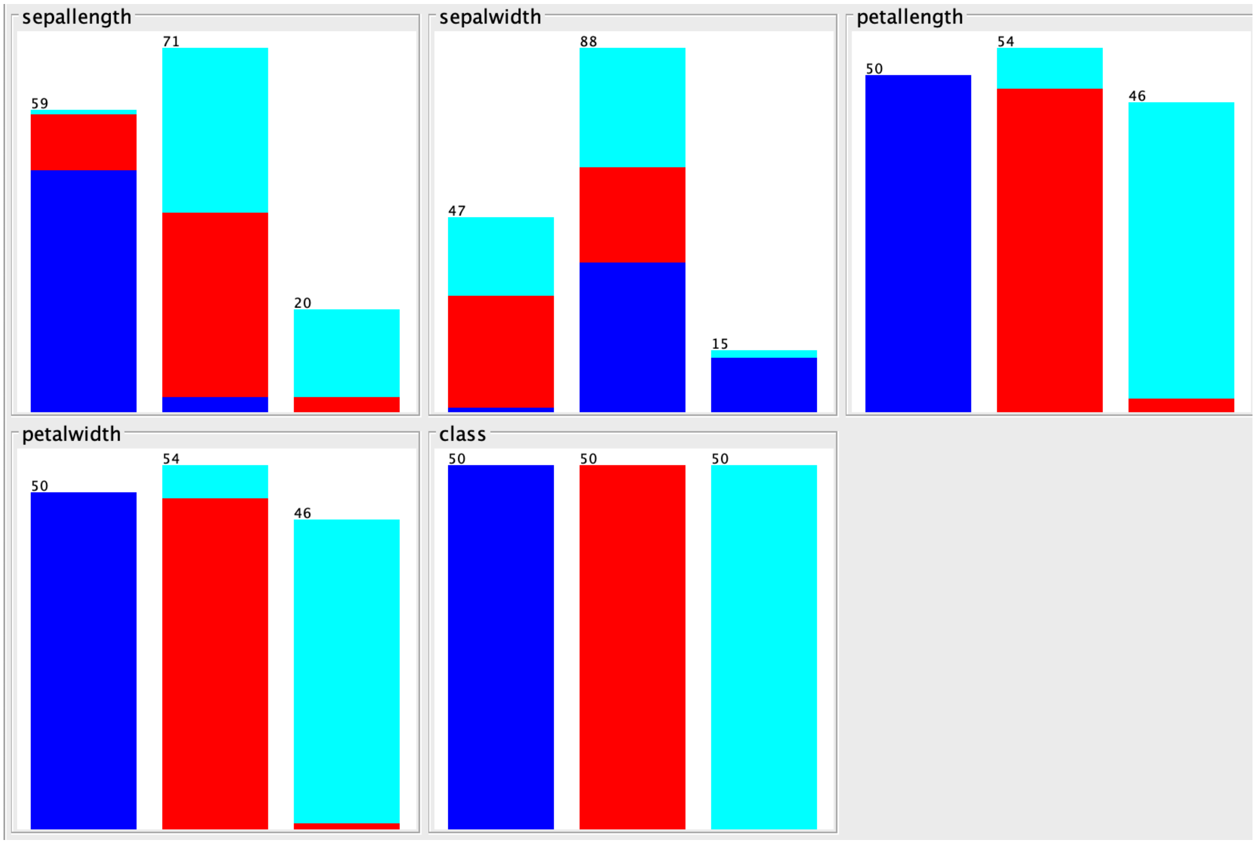
Rule for Cluster 2(Iris Setosa) Rule 1: petalwidth=‘(-inf-0.9]’ 50 ==> petallength=‘(-inf-2.966667]’ 50 (confidence: 1) Rule 2: class=Iris-setosa 50 ==> petallength=‘(-inf-2.966667]’ 50 (confidence: 1) Rule 3: class=Iris-setosa 50 ==> petalwidth=‘(-inf-0.9]’ 50 (confidence: 1)

For Cluster 0 has medium petal dimensions and for Cluster 1 has larger petal dimensions.

Different Number of Cluster

Here, we used a same clustering algorithm(SimpleKMeans). We got the following results:

For Pre Process(with filter Discrete of 3 bins), visualization of attributes



Now in Cluster tab, we choose SimpleKmeans as Clustering method with number of clusters as 5 and seed as 10, we got the following results

=== Run information ===

Scheme:weka.clusterers.SimpleKMeans -N 5 -A “weka.core.EuclideanDistance -R first-last” -I 500 -S 10 Relation: iris-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4-weka.filters.unsupervised.attribute.AddCluster-Wweka.clusterers.SimpleKMeans -N 2 -A “weka.core.EuclideanDistance -R first-last” -I 500 -S 10-weka.filters.unsupervised.attribute.Remove-R6-weka.filters.unsupervised.attribute.AddCluster-Wweka.clusterers.SimpleKMe -N 3 -A “weka.core.EuclideanDistance -R first-last” -I 500 -S 10-I4-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4-weka.filters.unsupervised.attribute.Remove-R6-weka.filters.unsupervised.attribute.Discretize-B5-M-1.0-R1-4-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4 Instances: 150 Attributes: 5 sepallength sepalwidth petallength petalwidth Ignored: class Test mode:Classes to clusters evaluation on training data === Model and evaluation on training set ===

kMeans

Number of iterations: 3 Within cluster sum of squared errors: 77.0 Missing values globally replaced with mean/mode

Cluster centroids: Cluster# Attribute Full Data 0 1 2 3 4 (150) (52) (44) (14) (4) (36) ======================= sepallength ‘(5.5-6.7]’ ‘(5.5-6.7]’ ‘(5.5-6.7]’ ‘(-inf-5.5]’ ‘(6.7-inf)’ ‘(-inf-5.5]’ sepalwidth ‘(2.8-3.6]’ ‘(-inf-2.8]’ ‘(2.8-3.6]’ ‘(3.6-inf)’ ‘(2.8-3.6]’ ‘(2.8-3.6]’ petallength ‘(2.966667-4.933333]’ ‘(2.966667-4.933333]’ ‘(4.933333-inf)’ ‘(-inf-2.966667]’ ‘(2.966667-4.933333]’ ‘(-inf-2.966667]’ petalwidth ‘(0.9-1.7]’ ‘(0.9-1.7]’ ‘(1.7-inf)’ ‘(-inf-0.9]’ ‘(0.9-1.7]’ ‘(-inf-0.9]’

Time taken to build model (full training data) : 0.02 seconds

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

Clustered Instances

0 52 (35%) 1 44 (29%) 2 14 (9%) 3 4 (3%) 4 36 (24%)

Class attribute: class Classes to Clusters:

0 1 2 3 4 <– assigned to cluster 0 0 14 0 36 | Iris-setosa 45 2 0 3 0 | Iris-versicolor 7 42 0 1 0 | Iris-virginica

Cluster 0 <– Iris-versicolor Cluster 1 <– Iris-virginica Cluster 2 <– No class Cluster 3 <– No class Cluster 4 <– Iris-setosa

Incorrectly clustered instances : 27.0 18 %

Then in Associate tab, we choose the same algorithm Apriori Algorithm, we got the rules as the following

=== 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: iris-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4-weka.filters.unsupervised.attribute.AddCluster-Wweka.clusterers.SimpleKMeans -N 2 -A “weka.core.EuclideanDistance -R first-last” -I 500 -S 10-weka.filters.unsupervised.attribute.Remove-R6-weka.filters.unsupervised.attribute.AddCluster-Wweka.clusterers.SimpleKMe -N 3 -A “weka.core.EuclideanDistance -R first-last” -I 500 -S 10-I4-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4-weka.filters.unsupervised.attribute.Remove-R6-weka.filters.unsupervised.attribute.Discretize-B5-M-1.0-R1-4-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4 Instances: 150 Attributes: 5 sepallength sepalwidth petallength petalwidth class === Associator model (full training set) ===

Apriori

Minimum support: 0.3 (45 instances) Minimum metric : 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

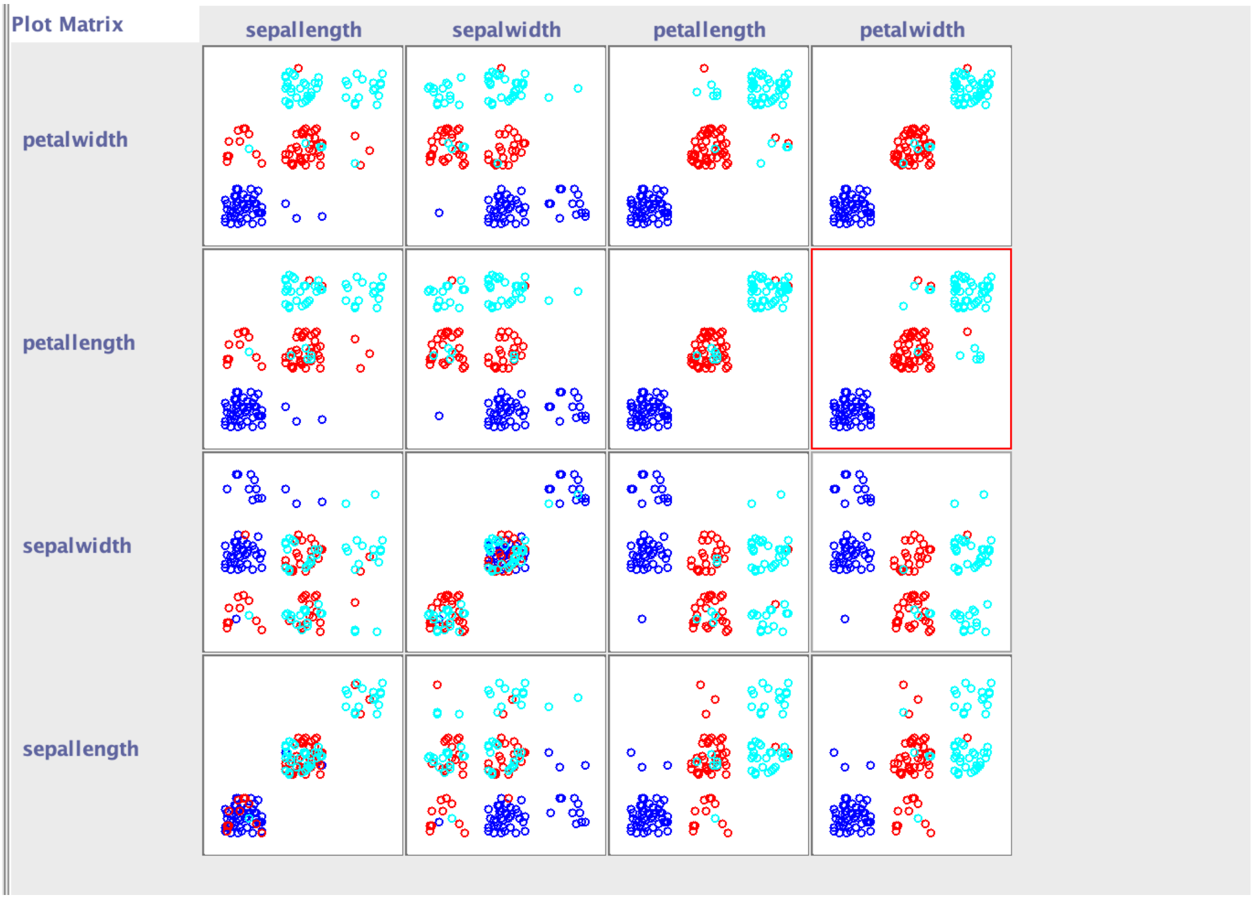
Size of set of large itemsets L(4): 1

Best rules found:

1. petalwidth=‘(-inf-0.9]’ 50 ==> petallength=‘(-inf-2.966667]’ 50 conf:(1)
2. petallength=‘(-inf-2.966667]’ 50 ==> petalwidth=‘(-inf-0.9]’ 50 conf:(1)
3. class=Iris-setosa 50 ==> petallength=‘(-inf-2.966667]’ 50 conf:(1)
4. petallength=‘(-inf-2.966667]’ 50 ==> class=Iris-setosa 50 conf:(1)
5. class=Iris-setosa 50 ==> petalwidth=‘(-inf-0.9]’ 50 conf:(1)
6. petalwidth=‘(-inf-0.9]’ 50 ==> class=Iris-setosa 50 conf:(1)
7. petalwidth=‘(-inf-0.9]’ class=Iris-setosa 50 ==> petallength=‘(-inf-2.966667]’ 50 conf:(1)
8. petallength=‘(-inf-2.966667]’ class=Iris-setosa 50 ==> petalwidth=‘(-inf-0.9]’ 50 conf:(1)

9. petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50 ==> class=Iris-setosa 50 conf:(1)
10. class=Iris-setosa 50 ==> petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50 conf:(1)

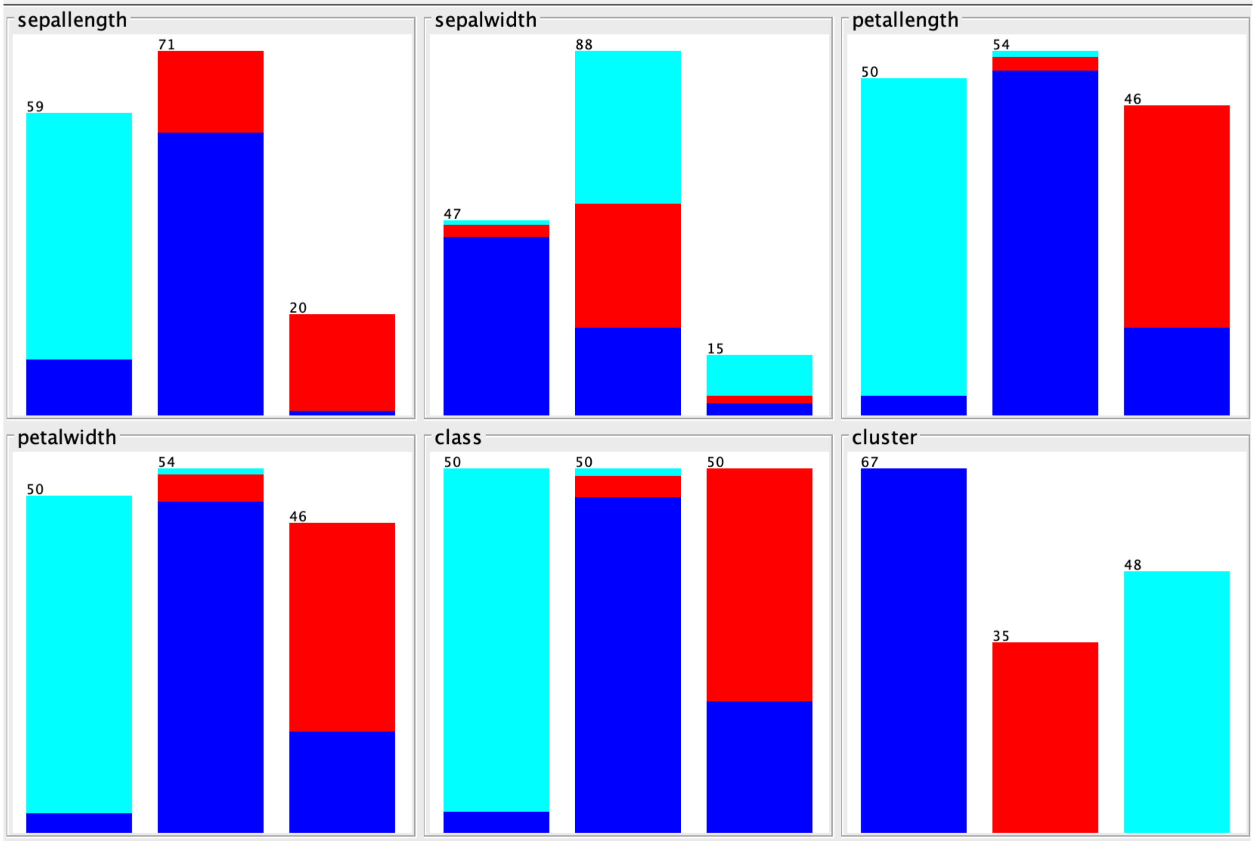
Then in visualize tab we got the relationship plots of clusters and labels



Different Clustering Algorithm

Here, we used a different clustering algorithm(EM Algorithm). We got the following results:

For Pre Process(with filter Discrete of 3 bins), visualization of attributes



Now in Cluster tab, we choose EM as Clustering method with number of clusters as 3 and seed as 10, we got the following results

=== Run information ===

Scheme:weka.clusterers.EM -I 100 -N 3 -M 1.0E-6 -S 10 Relation: iris-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4-weka.filters.unsupervised.attribute.AddCluster-Wweka.clusterers.SimpleKMeans -N 2 -A “weka.core.EuclideanDistance -R first-last” -I 500 -S 10-weka.filters.unsupervised.attribute.Remove-R6-weka.filters.unsupervised.attribute.AddCluster-Wweka.clusterers.SimpleKMeans -N 3 -A “weka.core.EuclideanDistance -R first-last” -I 500 -S 10-I4-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4 Instances: 150 Attributes: 6 sepallength sepalwidth petallength petalwidth class Ignored: cluster Test mode:Classes to clusters evaluation on training data === Model and evaluation on training set ===

EM

Number of clusters: 3

```
Cluster
```

Attribute 0 1 2 (0.34) (0.33) (0.33) == sepallength ‘(-inf-5.5]’ 12.9846 48 1.0154 ‘(5.5-6.7]’ 37.0893 4.0001 32.9106 ‘(6.7-inf)’ 4.0007 1 17.9993 [total] 54.0746 53.0001 51.9253 sepalwidth ‘(-inf-2.8]’ 29.1681 2 18.8318 ‘(2.8-3.6]’ 23.9064 37.0001 30.0934 ‘(3.6-inf)’ 1 13.9999 3 [total] 54.0746 53.0001 51.9253 petallength ‘(-inf-2.966667]’ 1.0001 50.9999 1.0001 ‘(2.966667-4.933333]’ 49.9925 1.0002 6.0073 ‘(4.933333-inf)’ 3.082 1.0001 44.9179 [total] 54.0746 53.0001 51.9253 petalwidth ‘(-inf-0.9]’ 1.0001 50.9999 1.0001 ‘(0.9-1.7]’ 51.0495 1.0002 4.9503 ‘(1.7-inf)’ 2.025 1.0001

45.9749 [total] 54.0746 53.0001 51.9253 class Iris-setosa 1.0001 50.9999 1.0001 Iris-versicolor 50.8505 1.0002 1.1494 Iris-virginica 2.2241 1.0001 49.7758 [total] 54.0746 53.0001 51.9253

Time taken to build model (full training data) : 0.04 seconds

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

Clustered Instances

0 51 (34%) 1 50 (33%) 2 49 (33%)

Log likelihood: -2.79413

Class attribute: cluster Classes to Clusters:

0 1 2 <– assigned to cluster 47 3 17 | cluster1 3 0 32 | cluster2 1 47 0 | cluster3

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

Incorrectly clustered instances : 24.0 16 %

Then in Associate tab, we choose the same algorithm Apriori Algorithm, we got the rules as the following

=== 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: iris-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4-weka.filters.unsupervised.attribute.AddCluster-Wweka.clusterers.SimpleKMeans -N 2 -A “weka.core.EuclideanDistance -R first-last” -I 500 -S 10-weka.filters.unsupervised.attribute.Remove-R6-weka.filters.unsupervised.attribute.AddCluster-Wweka.clusterers.SimpleKMe -N 3 -A “weka.core.EuclideanDistance -R first-last” -I 500 -S 10-I4-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-4 Instances: 150 Attributes: 6 sepallength sepalwidth petallength petalwidth class cluster === Associator model (full training set) ===

Apriori

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

Generated sets of large itemsets:

Size of set of large itemsets L(1): 15

Size of set of large itemsets L(2): 18

Size of set of large itemsets L(3): 14

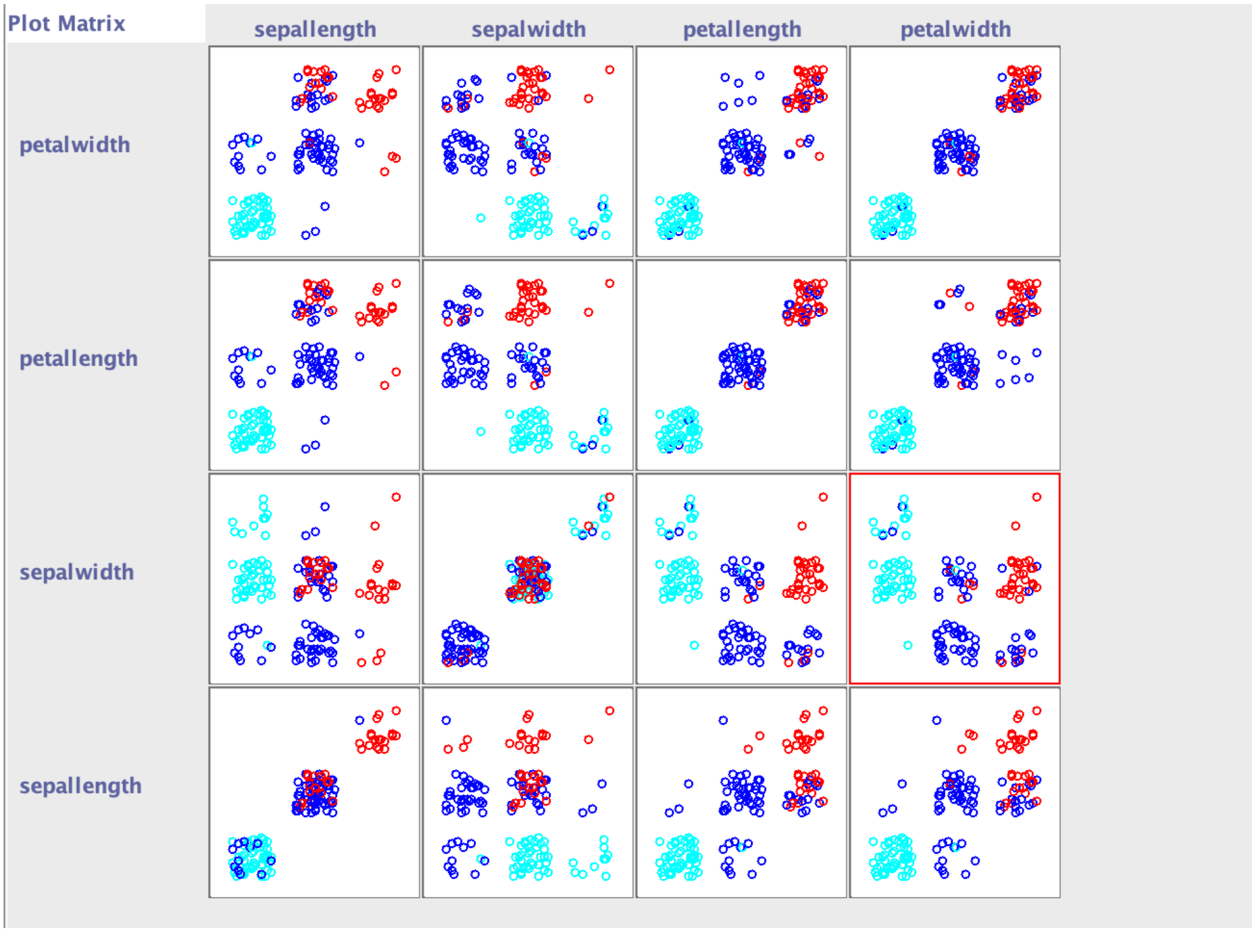
Size of set of large itemsets L(4): 5

Size of set of large itemsets L(5): 1

Best rules found:

1. petalwidth=‘(-inf-0.9]’ 50 ==> petallength=‘(-inf-2.966667]’ 50 conf:(1)
2. petallength=‘(-inf-2.966667]’ 50 ==> petalwidth=‘(-inf-0.9]’ 50 conf:(1)
3. class=Iris-setosa 50 ==> petallength=‘(-inf-2.966667]’ 50 conf:(1)
4. petallength=‘(-inf-2.966667]’ 50 ==> class=Iris-setosa 50 conf:(1)
5. class=Iris-setosa 50 ==> petalwidth=‘(-inf-0.9]’ 50 conf:(1)
6. petalwidth=‘(-inf-0.9]’ 50 ==> class=Iris-setosa 50 conf:(1)
7. petalwidth=‘(-inf-0.9]’ class=Iris-setosa 50 ==> petallength=‘(-inf-2.966667]’ 50 conf:(1)
8. petallength=‘(-inf-2.966667]’ class=Iris-setosa 50 ==> petalwidth=‘(-inf-0.9]’ 50 conf:(1)
9. petallength=‘(-inf-2.966667]’ petalwidth=‘(-inf-0.9]’ 50 ==> class=Iris-setosa 50 conf:(1)
10. class=Iris-setosa 50 ==> petallength=‘(-inf-2.966667]’ petalwidth=‘(-inf-0.9]’ 50 conf:(1)

Then in visualize tab we got the relationship plots of clusters and labels


Plot Matrix
sepallength
sepalwidth
petallength
petalwidth
petalwidth
petallength
sepalwidth
sepallength