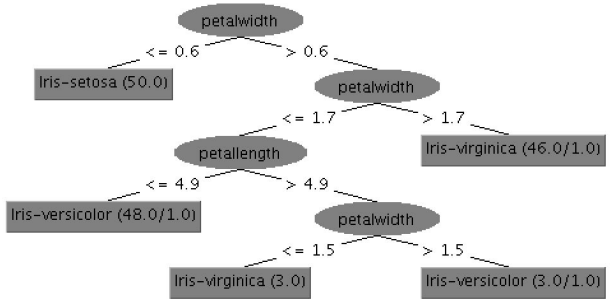
Fundamental of Data Mining
Due 10/30/2017
Assignment #3

John Warlop

1 Use the following learning schemes to analyze the iris data(in iris.arff)
One-R => Weka.classifiers.OneR
Naive Bayes => Weka.classifiers.bayes.NaiveBayes
C4.5 => Weka.classifiers.j48.J48

a. Give a brief description of the Decision Tree Model

The decision tree(see below) has 9 nodes and 5 leaves. Overall classification accuracty is 96%

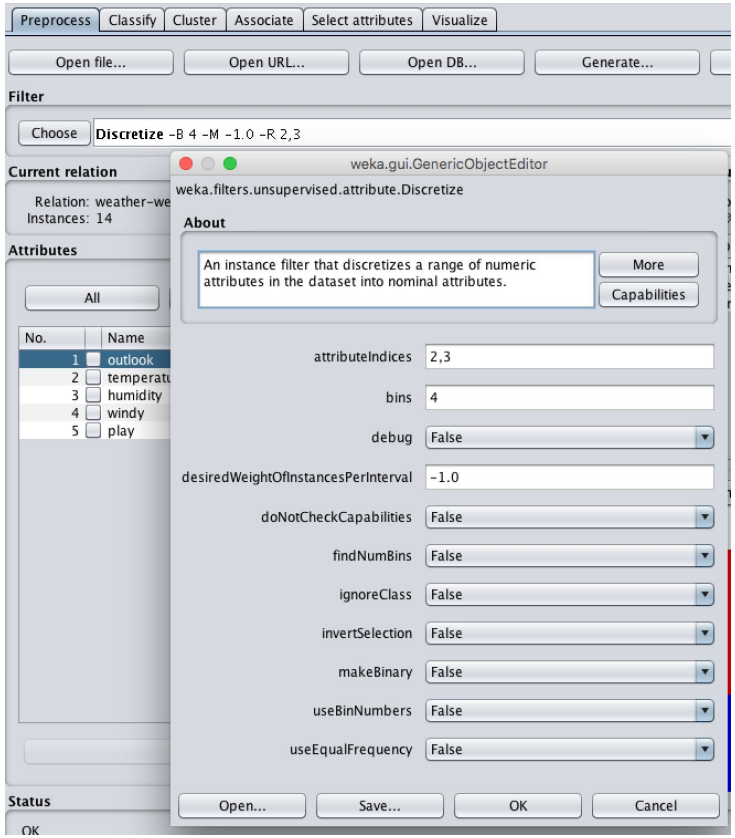



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.035
Root mean squared error                  0.1586
Relative absolute error                  7.8705 %
Root relative squared error             33.6353 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    0.000    1.000      0.980    0.990      0.985    0.990     0.987     Iris-setosa
                 0.940    0.030    0.940      0.940    0.940      0.910    0.952     0.880     Iris-versicolor
                 0.960    0.030    0.941      0.960    0.950      0.925    0.961     0.905     Iris-virginica
Weighted Avg.    0.960    0.020    0.960      0.960    0.960      0.940    0.968     0.924
```

	b. Do the decisions/models produced by the classifiers, in relation to the iris dataset, make sense to you? Why? Links: One-R | Naive Bayes | Decision Tree Yes, they do make sense to me. The decision tree is easy to follow, especially when you look at the graphic of tree. Naive Bayes is probably the most difficult to follow how the answer was arrived at. One-R just chooses the attribute that match class to highest degree. c. How did each one of the methods perform? We will cover the evaluation techniques later in the class – for now you can choose common sense or one of the techniques that Weka presents with the model (training data set, cross-validation or % split). The decision tree performed a tad better than Naive Bayes and the One-R performed the worst. d. Which method provided you with the most/least knowledge (insight into your data set/rules/patterns) and why? I believe the decision tree provided the most insight into the data and the One-R gave the least. The decision tree is nice because you can actually see how you could make an algorithm just by looking at the decision tree(i.e. use series of if statements). The One-R just chooses the attribute that best matches the class. It does not really tell you too much about the data
2	Data preparation is an essential step in data mining. How the training data set is presented to a method can drastically affect the produced model's performance. Use the J48 Decision tree-learning scheme to analyze Weather.numeric.arff and weather.nominal.arff (the data sets come with the Weka installation) data set. Make predictions for the 'temperature' attribute for both data sets. LINK==>

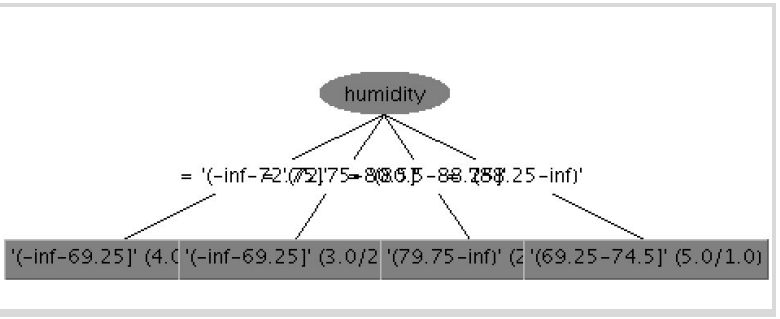
a. Try to use J48 on weather.numeric.arff with no modifications to the dataset. Did you get an error? The method only performs on nominal class data – Use the DiscretizeFilter (unsupervised Discretize) filter, in the preprocess tab, before applying the learning method. Be sure to note how you discretized the dataset and take a moment to consider why you made the choice? Did you discretize all the attributes? How many bins did you discretize each attribute into?

When I loaded weather.numeric.arff and then tried to use J48 on temperature, the start button was greyed out -- I could not use it. From image below, you can see that I only discretized #2(temperature) and #3(humidity). I played with bins and 4 appeared best.



b. Analyze the output of the model that learned the discretized attribute ‘temperature’? What was the performance, can you improve it? What did the model tell you about the data? ? (hint: you can modify the number of bins in the discretize filter to try to improve the model performance or mimic the nominal dataset)

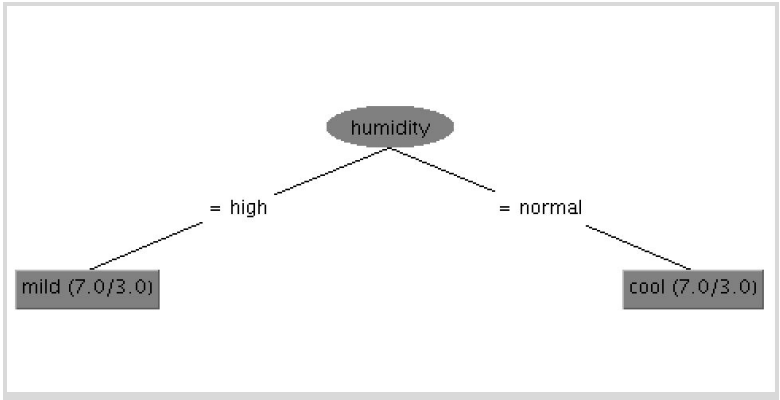
I was able to improve performance by changing the number of bins to 4. 64% was best performance I could get by playing around with bin #’s.




```
J48 pruned tree
------------------

humidity = '(-inf-72.75]': '(-inf-69.25]' (4.0/1.0)
humidity = '(72.75-80.5]': '(-inf-69.25]' (3.0/2.0)
humidity = '(80.5-88.25]': '(79.75-inf)' (2.0)
humidity = '(88.25-inf)': '(69.25-74.5]' (5.0/1.0)

Number of Leaves  :     4

Size of the tree :      5


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           9               64.2857 %
Incorrectly Classified Instances         5               35.7143 %
Kappa statistic                          0.507
Mean absolute error                      0.25
Root mean squared error                  0.4155
Relative absolute error                 64.6739 %
Root relative squared error             92.5141 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.750    0.300    0.500      0.750    0.600      0.411    0.763     0.634     '(-inf-69.25]'
                 1.000    0.100    0.800      1.000    0.889      0.849    0.900     0.800     '(69.25-74.5]'
                 0.000    0.083    0.000      0.000    0.000      -0.113   0.292     0.143     '(74.5-79.75]'
                 0.500    0.000    1.000      0.500    0.667      0.645    0.600     0.643     '(79.75-inf)'
Weighted Avg.    0.643    0.126    0.657      0.643    0.616      0.528    0.688     0.614
```

c. Analyze the output of the model that learned the nominal attribute ‘temperature’? What was the performance, can you improve it? What did the model tell you about the data? How do the results differ from the model produced on the discretized version of the same attribute?

When I ran the nominal data using temperature as the class, the classifier did not get any correct. This told me that some non-nominal data when discretized will give you a better classifier than with nominal data. I believe this the case because with non-nominal data, you can adjust discretation to give yourself more decision points and you can play around with bin sizes. Since classifier was 0% correct, you could negate humidity and then get 100% classifier.




```
J48 pruned tree
------------------

humidity = high: mild (7.0/3.0)
humidity = normal: cool (7.0/3.0)

Number of Leaves  :     2

Size of the tree :      3


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           0               0      %
Incorrectly Classified Instances        14             100      %
Kappa statistic                         -0.5806
Mean absolute error                      0.5079
Root mean squared error                  0.5842
Relative absolute error                110.9185 %
Root relative squared error            118.9573 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.400    0.000      0.000    0.000      -0.400   0.400     0.333     hot
                 0.000    1.000    0.000      0.000    0.000      -1.000   0.167     0.321     mild
                 0.000    0.200    0.000      0.000    0.000      -0.258   0.600     0.400     cool
Weighted Avg.    0.000    0.600    0.000      0.000    0.000      -0.617   0.357     0.347
```

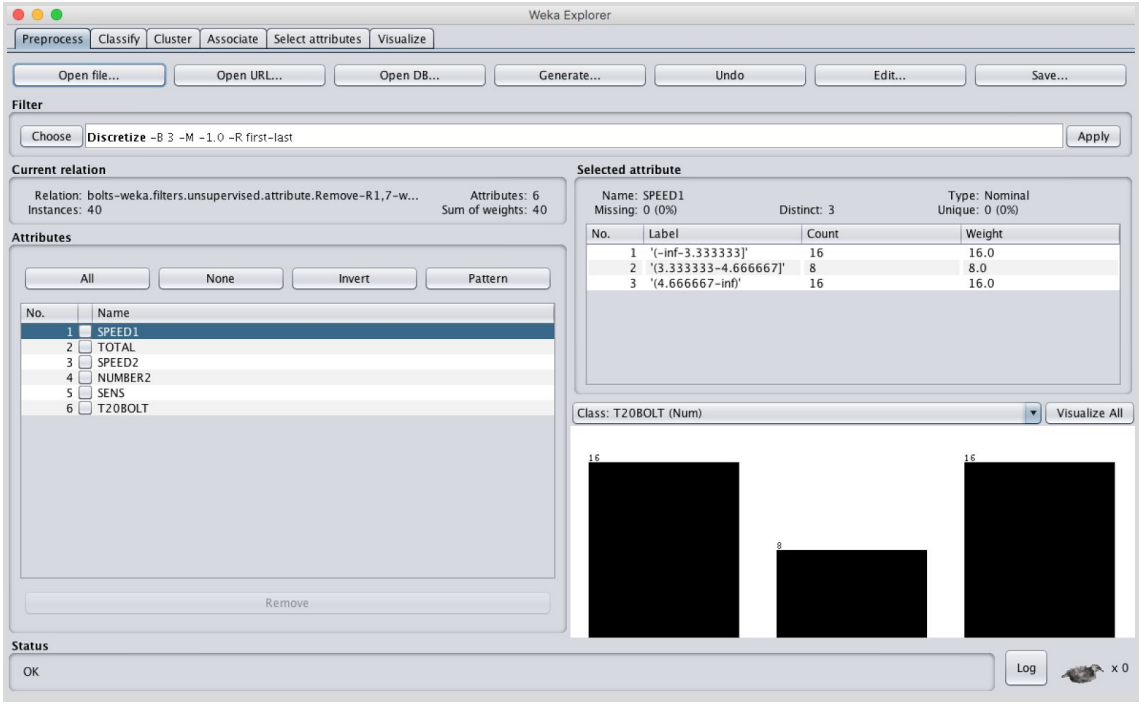
3 Use the J48 Decision tree learning scheme to analyze the bolts data (bolts.arff without the TIME attribute). The dataset describes the time needed by a machine to produce and count 20 bolts. (More details can be found in the file containing the dataset, you can open the file using a file editor to read the comments)

No.	1: RUN	2: SPEED1	3: TOTAL	4: SPEED2	5: NUMBER2	6: SENS	7: T20BOLT
	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	25.0	2.0	10.0	1.5	0.0	6.0	11.4
2	24.0	2.0	10.0	1.5	0.0	10.0	35.12
3	30.0	2.0	10.0	1.5	2.0	6.0	22.56
4	2.0	2.0	10.0	1.5	2.0	10.0	16.78
5	40.0	2.0	10.0	2.5	0.0	6.0	33.34
6	37.0	2.0	10.0	2.5	0.0	10.0	24.08
7	16.0	2.0	10.0	2.5	2.0	6.0	18.44
8	22.0	2.0	10.0	2.5	2.0	10.0	7.88
9	33.0	2.0	30.0	1.5	0.0	6.0	18.01
...	17.0	2.0	30.0	1.5	0.0	10.0	12.97
...	28.0	2.0	30.0	1.5	2.0	6.0	12.36
...	27.0	2.0	30.0	1.5	2.0	10.0	17.13
...	14.0	2.0	30.0	2.5	0.0	6.0	12.68
...	13.0	2.0	30.0	2.5	0.0	10.0	14.93
...	4.0	2.0	30.0	2.5	2.0	6.0	15.9
...	21.0	2.0	30.0	2.5	2.0	10.0	20.08
...	23.0	6.0	10.0	1.5	0.0	6.0	26.84
...	35.0	6.0	10.0	1.5	0.0	10.0	68.52
...	19.0	6.0	10.0	1.5	2.0	6.0	79.48
...	34.0	6.0	10.0	1.5	2.0	10.0	21.2
...	31.0	6.0	10.0	2.5	0.0	6.0	57.78
...	9.0	6.0	10.0	2.5	0.0	10.0	71.22
...	38.0	6.0	10.0	2.5	2.0	6.0	34.4
...	15.0	6.0	10.0	2.5	2.0	10.0	12.0
...	39.0	6.0	30.0	1.5	0.0	6.0	86.3
...	8.0	6.0	30.0	1.5	0.0	0.0	71.59
...	26.0	6.0	30.0	1.5	2.0	6.0	74.44
...	11.0	6.0	30.0	1.5	2.0	0.0	72.73
...	6.0	6.0	30.0	2.5	0.0	6.0	66.95
...	20.0	6.0	30.0	2.5	0.0	0.0	72.85
...	10.0	6.0	30.0	2.5	2.0	6.0	70.97
...	32.0	6.0	30.0	2.5	2.0	0.0	89.34
...	1.0	4.0	20.0	2.0	1.0	8.0	10.78
...	3.0	4.0	20.0	2.0	1.0	8.0	9.39
...	5.0	4.0	20.0	2.0	1.0	8.0	9.84
...	7.0	4.0	20.0	2.0	1.0	8.0	13.94
...	12.0	4.0	20.0	2.0	1.0	8.0	12.33
...	18.0	4.0	20.0	2.0	1.0	8.0	7.32
...	29.0	4.0	20.0	2.0	1.0	8.0	7.91
...	36.0	4.0	20.0	2.0	1.0	8.0	15.58

```
% SUMMARY:
%
% Data from an experiment on the affects of machine adjustments on
% the time to count bolts.  Data appear as the STATS (Issue 10) Challenge.
%
% DATA:
A manufacturer of automotive accessories provides hardware, e.g. nuts, bolts, washers and
screws, to fasten the accessory to the car or truck. Hardware is counted and packaged
automatically.  Specifically, bolts are dumped into a large metal dish.  A plate that forms
the bottom of the dish rotates counterclockwise.  This rotation forces bolts to the outside of
the dish and up along a narrow ledge.  Due to the vibration of the dish caused by the spinning
bottom plate, some bolts fall off the ledge and back into the dish.  The ledge spirals up to a
point where the bolts are allowed to drop into a pan on a conveyor belt.  As a bolt drops, it
passes by an electronic eye that counts it.  When the electronic counter reaches the preset
number of bolts, the rotation is stopped and the conveyor belt is moved forward.

There are several adjustments on the machine that affect its operation.  These include; a
speed setting that controls the speed of rotation (SPEED1) of the plate at the bottom of the
dish, a total number of bolts (TOTAL) to be counted, a second speed setting (SPEED2) that is
used to change the speed of rotation (usually slowing it down) for the last few bolts, the
number of bolts to be counted at this second speed (NUMBER2), and the sensitivity of the
electronic eye (SENS).  The sensitivity setting is to insure that the correct number of bolts
are counted.  Too few bolts packaged causes customer complaints.  Too many bolts packaged
increases costs.  For each run conducted in this experiment the correct number of bolts was
counted.  From an engineering standpoint if the correct number of bolts is counted, the
sensitivity should not affect the time to count bolts.  The measured response is the time
(TIME), in seconds, it takes to count the desired number of bolts.  In order to put times on a
equal footing the response to be analyzed is the time to count 20 bolts (T20BOLT).
Below are the data for 40 combinations of settings.  RUN is the order in which the data were
collected.
%
Analyze the data.  What adjustments have the greatest effect on the time to count 20 bolts?
How would you adjust the machine to get the shortest time to count 20 bolts?  Are there any
unusual features to the data?
```


a. Analyze the model produced. What adjustments (if you were to make any) would have the greatest effect on the time to count 20 bolts (attribute: T20Bolt) (i.e. what is the most important/selective attribute/value pair in the tree)?

As seen from images below, I removed Time and Run#. I disretized first-last. When I ran with different class variables, the one that had the best classification was speed one. I used 3 bins. One thing I could not figure out was why I could not discretize T20BOLT.



```
J48 pruned tree
------------------

TOTAL = '(-inf-16.666667]'
|   T20BOLT <= 35: '(-inf-3.333333]' (12.0/4.0)
|   T20BOLT > 35: '(4.666667-inf)' (4.0)
TOTAL = '(16.666667-23.333333]': '(3.333333-4.666667]' (8.0)
TOTAL = '(23.333333-inf)'
|   T20BOLT <= 35: '(-inf-3.333333]' (8.0)
|   T20BOLT > 35: '(4.666667-inf)' (8.0)

Number of Leaves  :     5

Size of the tree :      8


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          33               82.5    %
Incorrectly Classified Instances         7               17.5    %
Kappa statistic                          0.7266
Mean absolute error                      0.1396
Root mean squared error                  0.3026
Relative absolute error                 32.4186 %
Root relative squared error             65.0702 %
Total Number of Instances               40

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.813    0.167    0.765      0.813    0.788      0.640    0.862     0.749     '(-inf-3.333333]'
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     '(3.333333-4.666667]'
                 0.750    0.125    0.800      0.750    0.774      0.632    0.872     0.815     '(4.666667-inf)'
Weighted Avg.    0.825    0.117    0.826      0.825    0.825      0.709    0.894     0.826
```

b. According to the classifier, how would you adjust the machine(the other attributes) to get the shortest time to count 20 bolts?

I would adjust the SPEED1. As we look at tree, we want T20BOLT <=35 and we see that SPEED1 is between 0 and 3.333. From the tree, we'd take the TOTAL=23.3 to inf(we want 20+ bolts). From there we would choose the T20BOLT <= 35 because we want 20 bolts. This problem reminds me of the bean bag problem. You've got so many variables, each variable has multiple dependencies on the other. I would have messaged the data more. Since we are trying to get at least 20 bolts in bag(make customer happy) and not too many to keep costs down. I would have changed input file to have three ranges: (0-17),(18-22),(23-inf).

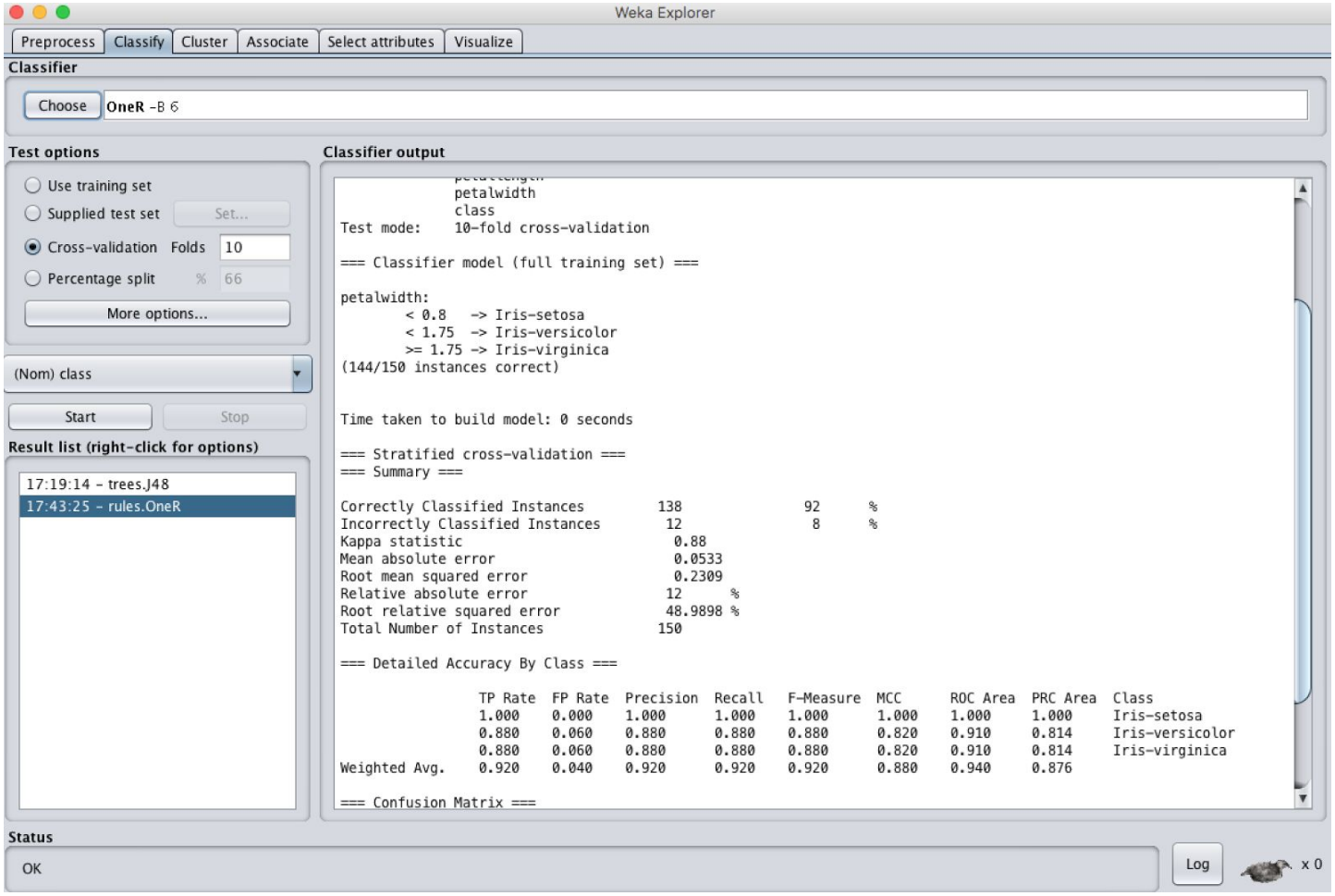
Appendix - Weka Output

	Fundamental of Data Mining ########## Problem #1 ########## ########## One-R ########## ########## Naive Bayes ########## ########## Decision Tree ########## ########## Problem #2 - Weather ########## === Run information Weather === === Run information Weather.nominal === ########## Problem #3 ##########	

Problem #1

========== #1 One-R ==========

HOME



```
=== Run information ===

Scheme:       weka.classifiers.rules.OneR -B 6
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    10-fold cross-validation

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

petalwidth:
        < 0.8          -> Iris-setosa
        < 1.75         -> Iris-versicolor
        >= 1.75 -> Iris-virginica
(144/150 instances correct)


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         138               92      %
Incorrectly Classified Instances        12                8      %
Kappa statistic                          0.88
Mean absolute error                      0.0533
Root mean squared error                  0.2309
```

```
Relative absolute error                 12      %
Root relative squared error             48.9898 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Iris-setosa
                 0.880    0.060    0.880      0.880    0.880      0.820    0.910     0.814     Iris-versicolor
                 0.880    0.060    0.880      0.880    0.880      0.820    0.910     0.814     Iris-virginica
Weighted Avg.    0.920    0.040    0.920      0.920    0.920      0.880    0.940     0.876

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 44  6 |  b = Iris-versicolor
  0  6 44 |  c = Iris-virginica
```

========== *#1 Naive Bayes* ==========

=== Run information Naive Bayes ===

HOME

```
Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    10-fold cross-validation

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

Naive Bayes Classifier

                        Class
Attribute          Iris-setosa Iris-versicolor  Iris-virginica
                        (0.33)          (0.33)          (0.33)
============================================================
sepallength
  mean                  4.9913          5.9379          6.5795
  std. dev.              0.355          0.5042          0.6353
  weight sum                50              50              50
  precision             0.1059          0.1059          0.1059
```

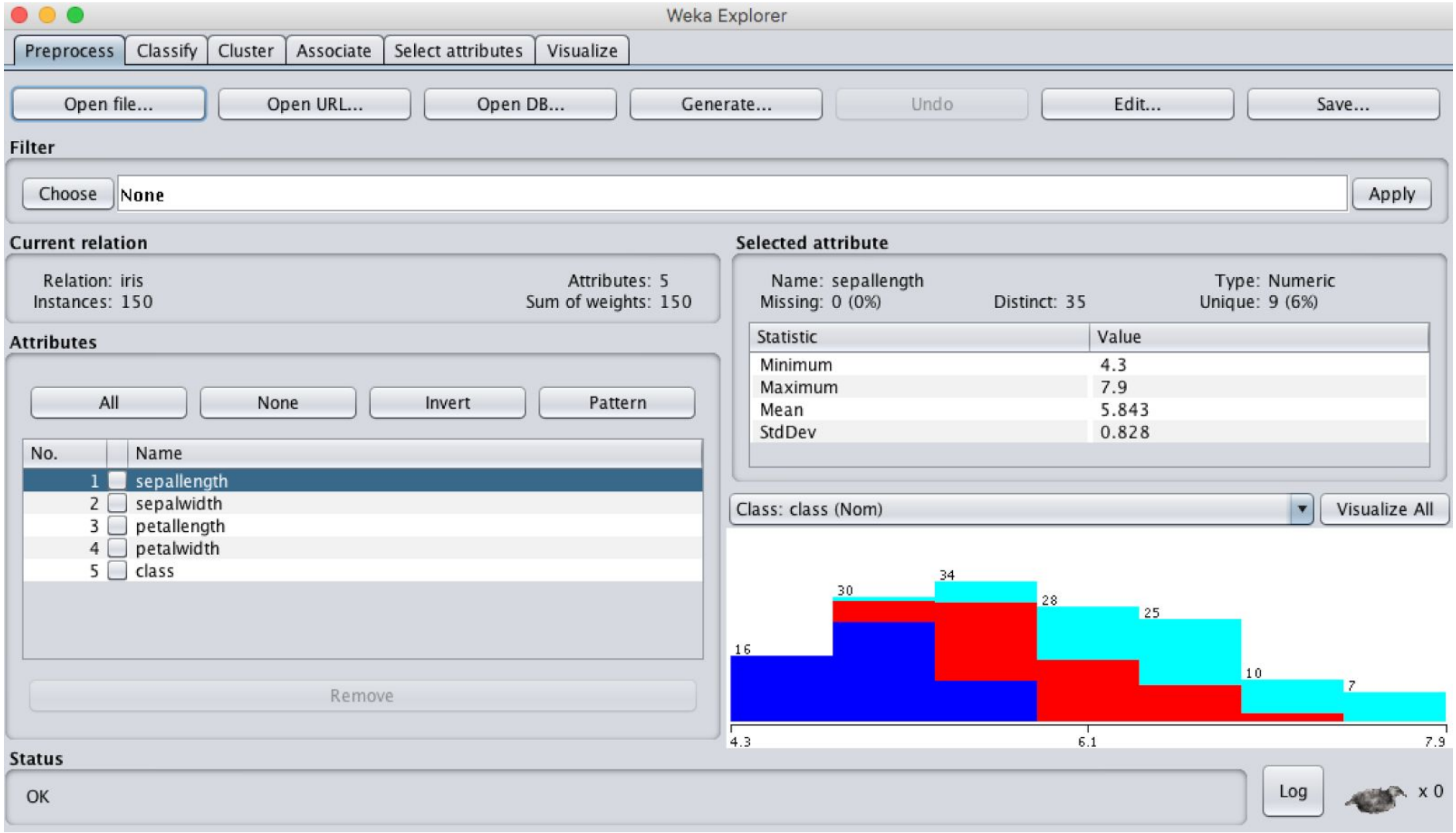
```
sepalwidth
  mean                    3.4015          2.7687          2.9629
  std. dev.               0.3925          0.3038          0.3088
  weight sum                  50              50              50
  precision               0.1091          0.1091          0.1091

petallength
  mean                    1.4694          4.2452          5.5516
  std. dev.               0.1782          0.4712          0.5529
  weight sum                  50              50              50
  precision               0.1405          0.1405          0.1405

petalwidth
  mean                    0.2743          1.3097          2.0343
  std. dev.               0.1096          0.1915          0.2646
  weight sum                  50              50              50
  precision               0.1143          0.1143          0.1143



Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
```

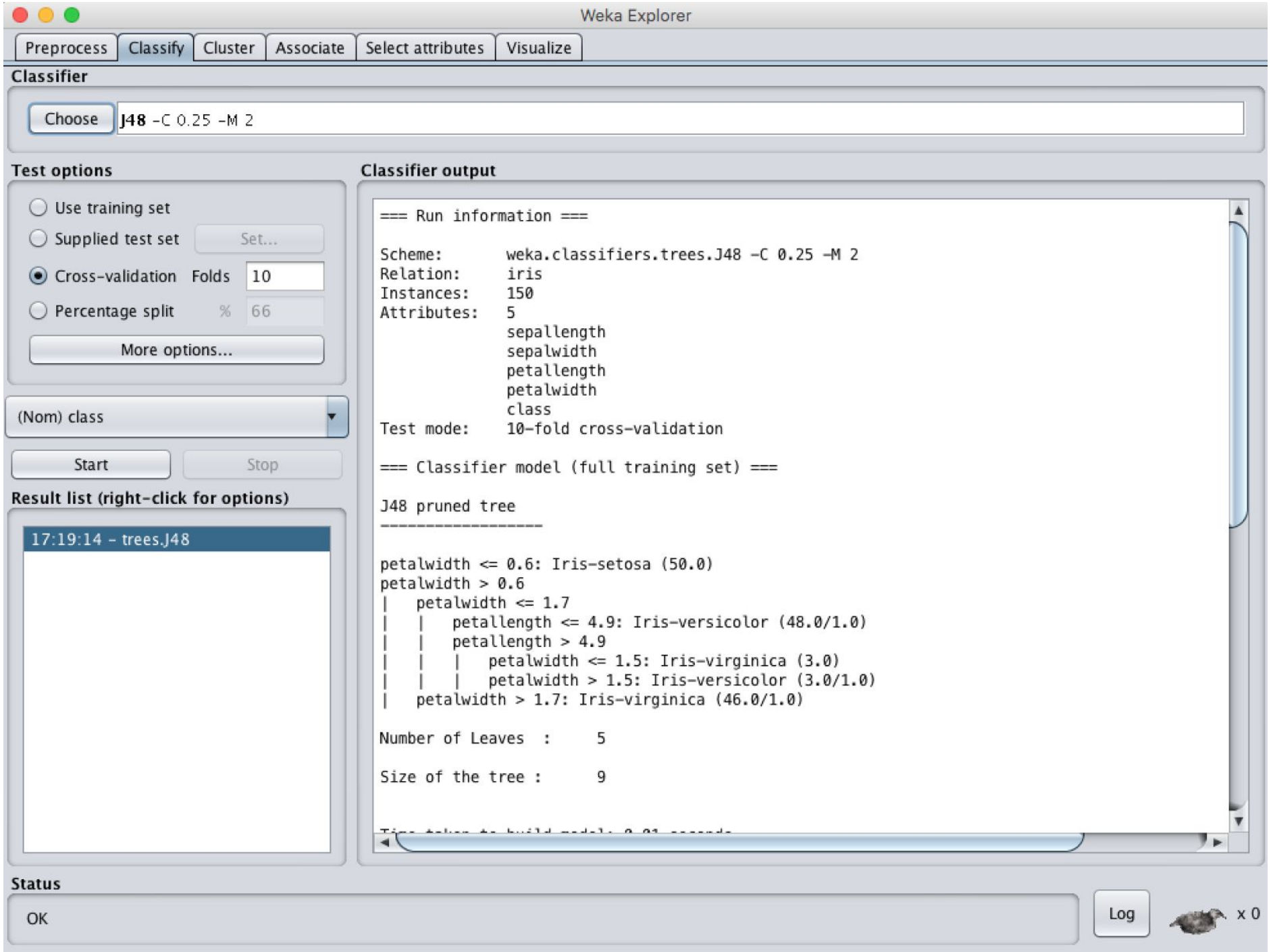
```
Kappa statistic                          0.94
Mean absolute error                      0.0342
Root mean squared error                  0.155
Relative absolute error                  7.6997 %
Root relative squared error             32.8794 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Iris-setosa
                 0.960    0.040    0.923      0.960    0.941      0.911    0.992     0.983     Iris-versicolor
                 0.920    0.020    0.958      0.920    0.939      0.910    0.992     0.986     Iris-virginica
Weighted Avg.    0.960    0.020    0.960      0.960    0.960      0.940    0.994     0.989

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 48  2 |  b = Iris-versicolor
  0  4 46 |  c = Iris-virginica
```

========== *#1 Decision Tree* ==========

=== Run information ===

HOME




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
17:19:14 - trees.J48
Classifier output
=== Run information ===
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
class
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
J48 pruned tree
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
| petalwidth <= 1.7
| | petallength <= 4.9: Iris-versicolor (48.0/1.0)
| | petallength > 4.9
| | | petalwidth <= 1.5: Iris-virginica (3.0)
| | | petalwidth > 1.5: Iris-versicolor (3.0/1.0)
| petalwidth > 1.7: Iris-virginica (46.0/1.0)
Number of Leaves : 5
Size of the tree : 9
Status
OK
Log
x 0

```
Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    10-fold cross-validation

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

J48 pruned tree
------------------

petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)

Number of Leaves  :       5

Size of the tree :  9
```

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.035
Root mean squared error                  0.1586
Relative absolute error                  7.8705 %
Root relative squared error             33.6353 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    0.000    1.000      0.980    0.990      0.985    0.990     0.987     Iris-setosa
                 0.940    0.030    0.940      0.940    0.940      0.910    0.952     0.880     Iris-versicolor
                 0.960    0.030    0.941      0.960    0.950      0.925    0.961     0.905     Iris-virginica
Weighted Avg.    0.960    0.020    0.960      0.960    0.960      0.940    0.968     0.924

=== Confusion Matrix ===

  a  b  c   <-- classified as
 49  1  0 |  a = Iris-setosa
```

```
0 47  3 |  b = Iris-versicolor
0  2 48 |  c = Iris-virginica
```

========== Problem #2 - Weather ==========

=== Run information Weather ===

HOME

	ARFF FILE: WEATHER @relation weather @attribute outlook {sunny, overcast, rainy} @attribute temperature real @attribute humidity real @attribute windy {TRUE, FALSE} @attribute play {yes, no} @data sunny,85,85,FALSE,no sunny,80,90,TRUE,no overcast,83,86,FALSE,yes rainy,70,96,FALSE,yes rainy,68,80,FALSE,yes rainy,65,70,TRUE,no overcast,64,65,TRUE,yes sunny,72,95,FALSE,no sunny,69,70,FALSE,yes rainy,75,80,FALSE,yes sunny,75,70,TRUE,yes overcast,72,90,TRUE,yes overcast,81,75,FALSE,yes rainy,71,91,TRUE,no	

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather-weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-last
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    10-fold cross-validation

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

J48 pruned tree
------------------

humidity = '(-inf-68.1]': '(-inf-66.1]' (1.0)
humidity = '(68.1-71.2]': '(-inf-66.1]' (3.0/2.0)
humidity = '(71.2-74.3]': '(70.3-72.4]' (0.0)
humidity = '(74.3-77.4]': '(80.8-82.9]' (1.0)
humidity = '(77.4-80.5]': '(66.1-68.2]' (2.0/1.0)
humidity = '(80.5-83.6]': '(70.3-72.4]' (0.0)
humidity = '(83.6-86.7]': '(82.9-inf)' (2.0)
humidity = '(86.7-89.8]': '(70.3-72.4]' (0.0)
humidity = '(89.8-92.9]': '(70.3-72.4]' (3.0/1.0)
humidity = '(92.9-inf)': '(68.2-70.3]' (2.0/1.0)

Number of Leaves  :   10

Size of the tree :    11


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           4               28.5714 %
Incorrectly Classified Instances        10               71.4286 %
Kappa statistic                          0.1463
Mean absolute error                      0.1488
Root mean squared error                  0.3612
Relative absolute error                 81.693  %
Root relative squared error            117.9882 %
```

```
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.083    0.000      0.000    0.000      -0.113   0.688     0.321     '(-inf-66.1]'
                 0.000    0.077    0.000      0.000    0.000      -0.077   0.385     0.071     '(66.1-68.2]'
                 0.000    0.167    0.000      0.000    0.000      -0.167   0.333     0.143     '(68.2-70.3]'
                 0.667    0.273    0.400      0.667    0.500      0.337    0.697     0.349     '(70.3-72.4]'
                 0.000    0.000    0.000      0.000    0.000      0.000    ?         ?         '(72.4-74.5]'
                 0.000    0.250    0.000      0.000    0.000      -0.213   0.292     0.143     '(74.5-76.6]'
                 0.000    0.000    0.000      0.000    0.000      0.000    ?         ?         '(76.6-78.7]'
                 0.000    0.000    0.000      0.000    0.000      0.000    0.423     0.071     '(78.7-80.8]'
                 0.000    0.000    0.000      0.000    0.000      0.000    0.462     0.071     '(80.8-82.9]'
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     '(82.9-inf)'
Weighted Avg.    0.286    0.135    0.229      0.286    0.250      0.139    0.570     0.320

=== Confusion Matrix ===

 a b c d e f g h i j   <-- classified as
 0 0 0 1 0 1 0 0 0 0 | a = '(-inf-66.1]'
 0 0 0 0 0 1 0 0 0 0 | b = '(66.1-68.2]'
 0 0 0 1 0 1 0 0 0 0 | c = '(68.2-70.3]'
 0 0 1 2 0 0 0 0 0 0 | d = '(70.3-72.4]'
 0 0 0 0 0 0 0 0 0 0 | e = '(72.4-74.5]'
 1 1 0 0 0 0 0 0 0 0 | f = '(74.5-76.6]'
 0 0 0 0 0 0 0 0 0 0 | g = '(76.6-78.7]'
 0 0 0 1 0 0 0 0 0 0 | h = '(78.7-80.8]'
 0 0 1 0 0 0 0 0 0 0 | i = '(80.8-82.9]'
 0 0 0 0 0 0 0 0 0 2 | j = '(82.9-inf)'

=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2 -J
Relation:     weather-weka.filters.unsupervised.attribute.Discretize-Y-B3-M-1.0-Rfirst-last
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    10-fold cross-validation
```

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

J48 pruned tree
------------------

outlook = sunny
|   play = yes: 'B1of3' (2.0/1.0)
|   play = no: 'B3of3' (3.0/1.0)
outlook = overcast
|   windy = TRUE: 'B1of3' (2.0/1.0)
|   windy = FALSE: 'B3of3' (2.0)
outlook = rainy: 'B1of3' (5.0/1.0)

Number of Leaves  :       5

Size of the tree :  8


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           3               21.4286 %
Incorrectly Classified Instances        11               78.5714 %
Kappa statistic                         -0.2031
Mean absolute error                      0.486
Root mean squared error                  0.5743
Relative absolute error                106.4883 %
Root relative squared error            117.8073 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.500    0.375    0.500      0.500    0.500      0.125    0.521     0.592     'B1of3'
                 0.000    0.300    0.000      0.000    0.000      -0.330   0.100     0.239     'B2of3'
                 0.000    0.500    0.000      0.000    0.000      -0.471   0.400     0.325     'B3of3'
Weighted Avg.    0.214    0.389    0.214      0.214    0.214      -0.175   0.366     0.415
```

```
=== Confusion Matrix ===

 a b c   <-- classified as
 3 1 2 | a = 'B1of3'
 1 0 3 | b = 'B2of3'
 2 2 0 | c = 'B3of3'

================================ 2a weather.arff #bins 4 ================================
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather-weka.filters.unsupervised.attribute.Discretize-B4-M-1.0-Rfirst-last
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    10-fold cross-validation

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

J48 pruned tree
------------------

humidity = '(-inf-72.75]': '(-inf-69.25]' (4.0/1.0)
humidity = '(72.75-80.5]': '(-inf-69.25]' (3.0/2.0)
humidity = '(80.5-88.25]': '(79.75-inf)' (2.0)
humidity = '(88.25-inf)': '(69.25-74.5]' (5.0/1.0)

Number of Leaves  :       4

Size of the tree :  5


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===
```

```
Correctly Classified Instances           9               64.2857 %
Incorrectly Classified Instances         5               35.7143 %
Kappa statistic                          0.507
Mean absolute error                      0.25
Root mean squared error                  0.4155
Relative absolute error                 64.6739 %
Root relative squared error             92.5141 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.750    0.300    0.500      0.750    0.600      0.411    0.763     0.634     '(-inf-69.25]'
                 1.000    0.100    0.800      1.000    0.889      0.849    0.900     0.800     '(69.25-74.5]'
                 0.000    0.083    0.000      0.000    0.000      -0.113   0.292     0.143     '(74.5-79.75]'
                 0.500    0.000    1.000      0.500    0.667      0.645    0.600     0.643     '(79.75-inf)'
Weighted Avg.    0.643    0.126    0.657      0.643    0.616      0.528    0.688     0.614

=== Confusion Matrix ===

 a b c d   <-- classified as
 3 0 1 0 | a = '(-inf-69.25]'
 0 4 0 0 | b = '(69.25-74.5]'
 2 0 0 0 | c = '(74.5-79.75]'
 1 1 0 2 | d = '(79.75-inf)'
```

=== Run information Weather.nominal ===

```
Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather.symbolic
Instances:    14
```

```
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    10-fold cross-validation

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

J48 pruned tree
------------------

outlook = sunny
|   humidity = high: no (3.0)
|   humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)

Number of Leaves  :       5

Size of the tree :  8


Time taken to build model: 0 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                         -0.0426
Mean absolute error                      0.4167
Root mean squared error                  0.5984
Relative absolute error                 87.5    %
Root relative squared error            121.2987 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.556    0.600    0.625      0.556    0.588      -0.043   0.633     0.758     yes
                 0.400    0.444    0.333      0.400    0.364      -0.043   0.633     0.457     no
Weighted Avg.    0.500    0.544    0.521      0.500    0.508      -0.043   0.633     0.650

=== Confusion Matrix ===

 a b   <-- classified as
 5 4 | a = yes
 3 2 | b = no
```

========== #3 ==========