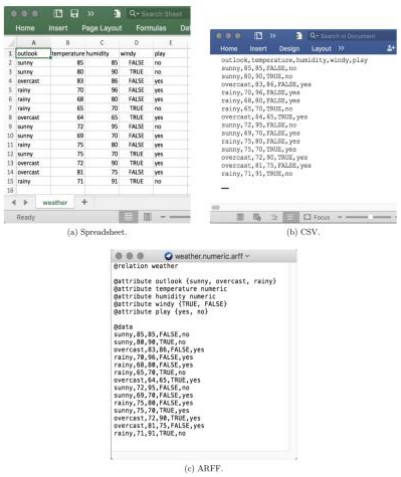
Week 3 Workshop

Decision trees with Weka

Suppose you have some data and you want to build a decision tree from it.

1. First, you need to prepare the data, then start the Explorer and load in the data.
2. Next you select a decision tree construction method, build a tree, and interpret the output.



(a) Spreadsheet. (b) CSV.

(c) ARFF.

Preparing the data

The data is often presented in a spreadsheet or database. However, WEKA's native data storage method is ARFF format. You can easily convert from a spreadsheet to ARFF.

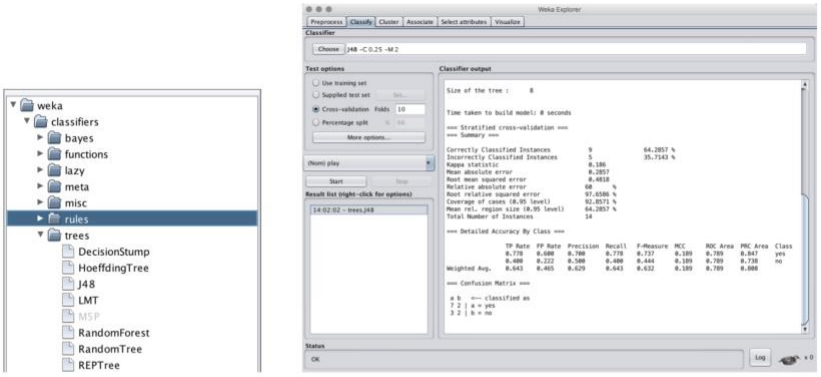
Loading the data into the Explorer

Let us load this data into the Explorer and start analyzing it.

1. Fire up WEKA to get the GUI Chooser panel.
2. Select Explorer from the five choices on the right-hand side. What you see next is the main Explorer screen. The six tabs along the top are the basic operations that the Explorer supports: right now we are on Preprocess.
3. Click the Open file button to bring up a standard dialog through which you can select a file.
4. Choose the weather.arff file.
5. If you have it in CSV format, change from ARFF data files to CSV data files. When you specify a .csv file it is automatically converted into ARFF format.
6. The screen now tells you about the dataset:
7. it has 14 instances and five attributes (center left); the attributes are called outlook, temperature, humidity, windy, and play (lower left). The first attribute, outlook, is selected by default (you can choose others by clicking them) and has no missing values, three distinct values, and no unique values; the actual values are sunny, overcast, and rainy and they occur five, four, and five times, respectively (center right).
8. A histogram at the lower right shows how often each of the two values of the class, play, occurs for each value of the outlook attribute.

Source: https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf#page17

9. The attribute outlook is used because it appears in the box above the histogram, but you can draw a histogram of any other attribute instead.
10. Here play is selected as the class attribute; it is used to color the histogram, and any filters that require a class value use it too.
11. The outlook attribute in is nominal. If you select a numeric attribute, you see its minimum and maximum values, mean, and standard deviation. In this case the histogram will show the distribution of the class as a function of this attribute.
12. You can delete an attribute by clicking its checkbox and using the Remove button.
    a. All selects all the attributes,
    b. None selects none,
    c. Invert inverts the current selection
    d. Pattern selects those attributes whose names match a user-supplied regular expression.
13. You can undo a change by clicking the Undo button.
14. The Edit button brings up an editor that allows you to inspect the data, search for particular values and edit them, and delete instances and attributes.



(a) Finding J4.8 in the classifiers list.

(b) The *Classify* tab.

Figure J4.8 in the Explorer.

15. Right-clicking on values and column headers brings up corresponding context menus.

Building a decision tree

To see what the C4.5 decision tree learner does with this dataset, use the J4.8 algorithm, which is WEKA's implementation of this algorithm.

- Click the Classify tab.
- First select the classifier by clicking the Choose button at the top left, opening up the trees section of the hierarchical menu and finding J48.
- The menu structure represents the organization of the WEKA code into modules and the items you need to select are always at the lowest level.
- Once selected, J48 appears in the line beside the Choose button along with its default parameter values.
- If you click that line, the J4.8 classifier's object editor opens up and you can see what the parameters mean and alter their values if you wish. The Explorer generally chooses sensible defaults.
- Having chosen the classifier, invoke it by clicking the Start button. WEKA works for a brief period—when it is working, the little bird at the lower right jumps up and dances—and then produces the output in the main panel.

Examining the output

I. At the beginning is a summary of the dataset, and the fact that 10-fold cross-validation was used to evaluate it. That is the default, and if you look closely you will see that the Cross-validation box at the left is checked.

II. Then comes a pruned decision tree in textual form.

III. The model that is shown here is always one generated from the full dataset available from the Preprocess panel.

IV. The first split is on the outlook attribute, and then, at the second level, the splits are on humidity and windy, respectively.

V. In the tree structure, a colon introduces the class label that has been assigned to a particular leaf, followed by the number of instances that reach that leaf, expressed as a decimal number because of the way the algorithm uses fractional instances to handle missing values.

VI. If there were incorrectly classified instances (there aren't in this example) their number would appear too: thus 2.0/1.0 means that two instances reached that leaf, of which one is classified incorrectly.

VII. Beneath the tree structure the number of leaves is printed; then the total number of nodes (Size of the tree). There is a way to view decision trees more graphically, which we will encounter later.

VIII. The next part of the output gives estimates of the tree's predictive performance.

IX. As well as the classification error, the evaluation module also outputs the Kappa statistic, the mean absolute error, and the root mean-squared error of the class probability estimates assigned by the tree.

X. Finally, for each class it also outputs various statistics. Also reported is the per-class average of each statistic, weighted by the number of instances from each class. All of these evaluation measures are discussed in Chapter 5 of the book.

Doing it again

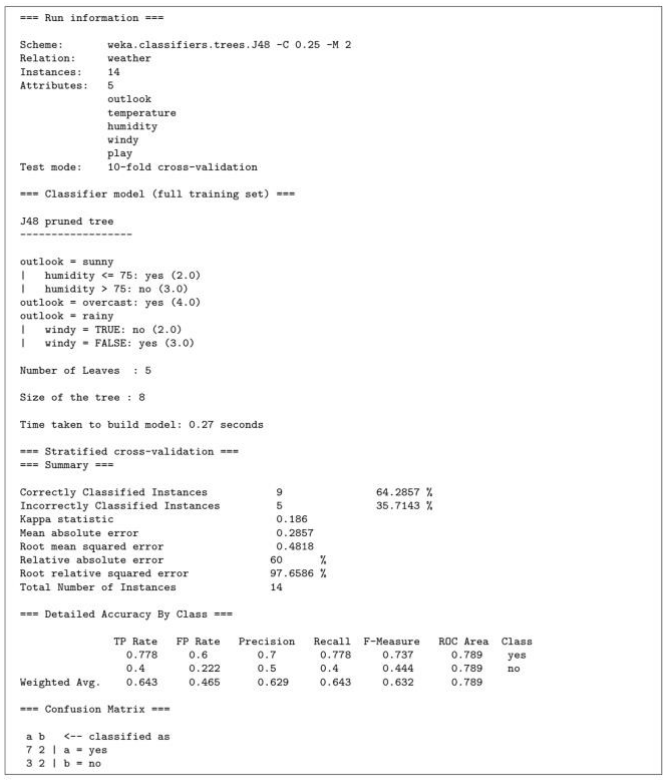
You can easily run J4.8 again with a different evaluation method. Select Use training set and click Start again.

Working with models

The small pane at the lower left of screen contains one highlighted line, is a history list of the results. The Explorer adds a new line whenever you run a classifier. To return to a previous result set, click the corresponding line and the output for that run will appear in the Classifier Output pane.

When things go wrong

Beneath the result history list, at the bottom, is a status line that says, simply, OK. Occasionally this changes to See error log, an indication that something has gone wrong.



```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather
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 <= 75: yes (2.0)
|   humidity > 75: no (3.0)
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: 0.27 seconds

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

Correctly Classified Instances           9               64.2857 %
Incorrectly Classified Instances         5               35.7143 %
Kappa statistic                          0.186
Mean absolute error                      0.2857
Root mean squared error                  0.4818
Relative absolute error                 60      %
Root relative squared error             97.6586 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.778    0.6        0.7       0.778     0.737      0.789    yes
                 0.4      0.222      0.5       0.4       0.444      0.789    no
Weighted Avg.    0.643    0.465      0.629     0.643     0.632      0.789

=== Confusion Matrix ===

 a b   <-- classified as
 7 2 | a = yes
 3 2 | b = no
```

Figure Output from the J4.8 decision tree learner.