Chapter 5 Decision Tree

LECTURER: ZHIHUA JIANG

Content

5.1 Basic Procedure

5.2 Attribute Selection

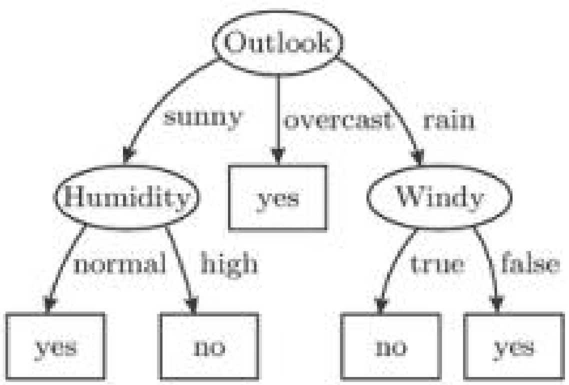
5.3 Continuous Value and Missing Value

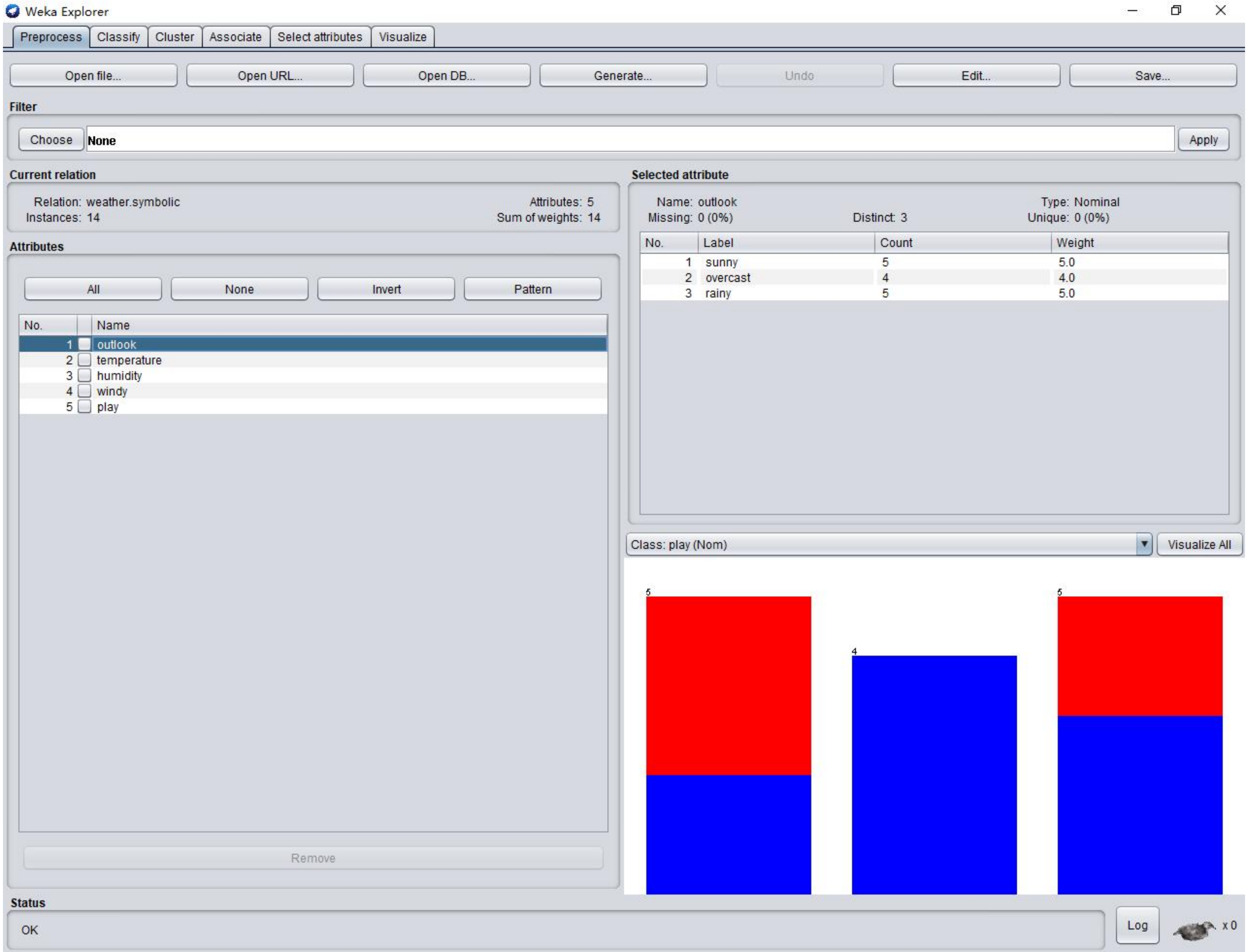
5.4 Random Forest

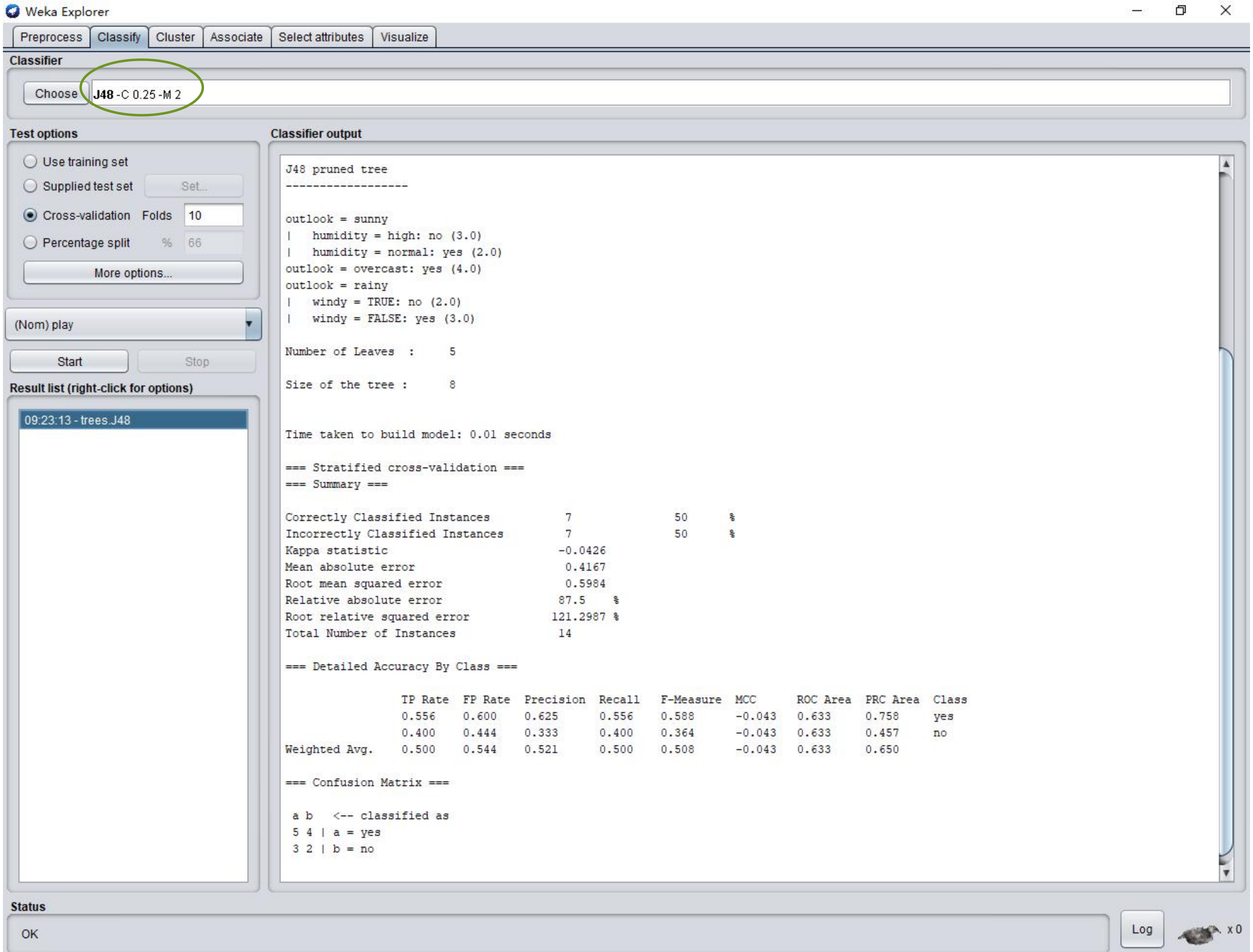
5.1 Basic Procedure

- Idea: make decisions based on a tree
  - Internal node: test on some attribute(s)
  - Branch: some value of an attribute
  - Leaf: decision result

Outlook	*Temp*	*Humidity*	*Windy*	*Golf?*
rainy	hot	high	false	no
rainy	hot	high	true	no
overcast	hot	high	false	yes
sunny	mild	high	false	yes
sunny	cool	normal	false	yes
sunny	cool	normal	true	no
overcast	cool	normal	true	yes
rainy	mild	high	false	no
rainy	cool	normal	false	yes
sunny	mild	normal	false	yes
rainy	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
sunny	mild	high	true	no



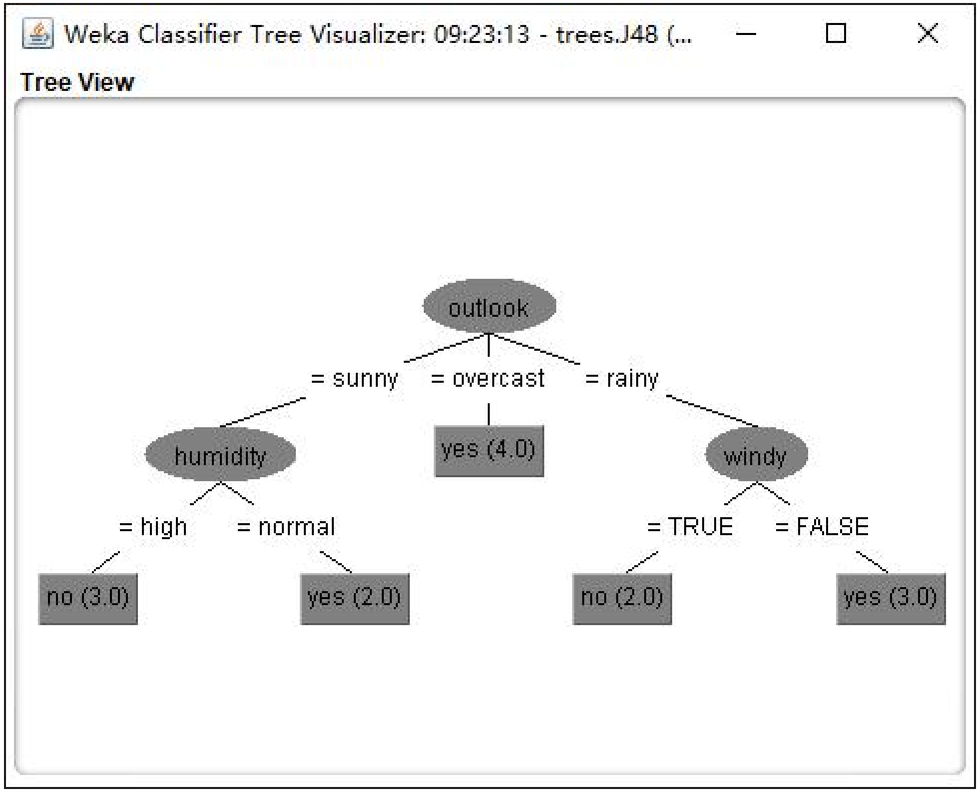

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: weather.symbolic
Instances: 14
Attributes: 5
Sum of weights: 14
Attributes
All
None
Invert
Pattern
No.
Name
1 outlook
2 temperature
3 humidity
4 windy
5 play
Remove
Selected attribute
Name: outlook
Type: Nominal
Missing: 0 (0%)
Distinct: 3
Unique: 0 (0%)
No.
Label
Count
Weight
1 sunny 5 5.0
2 overcast 4 4.0
3 rainy 5 5.0
Class: play (Nom)
Visualize All
5
4
5
Status
OK
Log
x 0


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) play
Start
Stop
Result list (right-click for options)
09:23:13 - trees.J48
Classifier output
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: 0.01 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
Status
OK
Log
x 0

```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```




```
=== 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
```

Brief History

- CLS (Concept Learning System)
  - The first decision tree learning algorithm, 1966
- ID3
  - Begin to be the mainstream technology in ML, 1979
- C4.5
  - The most commonly used algorithm, 1993
- CART (Classification And Regression Tree), 1984
- RF (Random Forest)
  - The most powerful decision tree algorithm, 2001

Basic algorithm

(the ID3 algorithm)

1. If all the instances are from exactly one class, then the decision tree is an answer node containing that class name.
2. Otherwise,
   (a) Define a_{best} to be an attribute with some mechanism
   (b) For each value $v_{best,i}$ of a_{best}, grow a branch from a_{best} to a decision tree constructed recursively from all those instances with value $v_{best,i}$ of attribute a_{best}.

5.2 Attribute Selection

- Information gain (信息增益)
- Gain ratio (增益率)
- Gini index (基尼指数)

Information gain

- Information entropy
  - Measure the purity of a data set
  - The smaller the entropy, the higher the purity
  - Definition: data set *D*, |*Y*| classes, the ratio of the k^{th} class p_k

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$

  - Information gain: to evaluate the change on the information entropy caused by a candidate partition

Information gain

- Information gain
  - all values on a **discrete** attribute a: $\{a^1, a^2, \ldots, a^v\}$
  - D^v: the subset where each sample takes the value a^v on a
  - The gain of the partition on a:

$$\mathrm{Gain}(D,a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v)$$

  - Best attribute:

$$a_* = \arg\max_{a \in A} \mathrm{Gain}(D,a)$$

The ID3 algorithm

The split using the feature *windy* results in two children nodes, one for a *windy* value of true and one for a *windy* value of false. In this data set, there are six data points with a true *windy* value, three of which have a *play* value of yes and three with a *play* value of no. The eight remaining data points with a *windy* value of false contain two no's and six yes's. The information of the *windy*=true node is calculated using the entropy equation above. Since there is an equal number of yes's and no's in this node, we have

$$I_E([3,3]) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

For the node where *windy*=false there were eight data points, six yes's and two no's. Thus we have

$$I_E([6,2]) = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8} = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} = 0.8112781$$

To find the information of the split, we take the weighted average of these two numbers based on how many observations fell into which node.

$$I_E([3,3],[6,2]) = I_E(\text{windy or not}) = \frac{6}{14}\cdot 1 + \frac{8}{14}\cdot 0.8112781 = 0.8921589$$

To find the information gain of the split using *windy*, we must first calculate the information in the data before the split. The original data contained nine yes's and five no's.

$$I_E([9,5]) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940286$$

Now we can calculate the information gain achieved by splitting on the *windy* feature.

$$IG(\text{windy}) = I_E([9,5]) - I_E([3,3],[6,2]) = 0.940286 - 0.8921589 = 0.0481271$$

Outlook	*Temp*	*Humidity*	*Windy*	*Golf?*
rainy	hot	high	false	no
rainy	hot	high	true	no
overcast	hot	high	false	yes
sunny	mild	high	false	yes
sunny	cool	normal	false	yes
sunny	cool	normal	true	no
overcast	cool	normal	true	yes
rainy	mild	high	false	no
rainy	cool	normal	false	yes
sunny	mild	normal	false	yes
rainy	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
sunny	mild	high	true	no

Gain ratio

- Information gain prefers to an attribute which has more distinct values, while gain ratio prefers to fewer values
- Definition:

$$\mathrm{Gain\_ratio}(D,a) = \frac{\mathrm{Gain}(D,a)}{\mathrm{IV}(a)} \qquad \mathrm{IV}(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

- The C4.5 algorithm: first Gain(D, a) > average; then the biggest gain ratio.

Gini index

- Measure the probability that two randomly chosen samples belong to different classes
- The smaller the Gini, the higher the purity
- Definition:

$$\mathrm{Gini}(D) = \sum_{k=1}^{|\mathcal{Y}|} \sum_{k' \neq k} p_k p_{k'}$$

$$= 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2 \,.$$

$$\mathrm{Gini\_index}(D, a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Gini}(D^v)$$

$$a_* = \underset{a \in A}{\arg\min}\ \mathrm{Gini\_index}(D, a).$$

The CART algorithm

	sunny	
class	Y	2
	N	3

$$Gini(sunny) = 1 - (\frac{2}{5})^2 - (\frac{3}{5})^2 = 0.48$$

	rain	
class	Y	3
	N	2

$$Gini(rain) = 1 - (\frac{3}{5})^2 - (\frac{2}{5})^2 = 0.48$$

	overcast	
class	Y	4
	N	0

$$Gini(overcast) = 1 - (\frac{4}{4})^2 - (\frac{0}{4})^2 = 0$$

$$Gini_{split}(Outlook) = \frac{5}{14} \ 0.48 + \frac{5}{14} \ 0.48 + \frac{4}{14} \ 0 = 0.343$$

$$\mathrm{Gini\_index}(D, a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Gini}(D^v)$$

Outlook	*Temp*	*Humidity*	*Windy*	*Golf?*
rainy	hot	high	false	no
rainy	hot	high	true	no
overcast	hot	high	false	yes
sunny	mild	high	false	yes
sunny	cool	normal	false	yes
sunny	cool	normal	true	no
overcast	cool	normal	true	yes
rainy	mild	high	false	no
rainy	cool	normal	false	yes
sunny	mild	normal	false	yes
rainy	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
sunny	mild	high	true	no

Attribute	Feature	Y	N	GINI (i)	GINI I_{split}
Outlook	sunny	2	3	0.48	0.343
	rain	3	2	0.48	
	overcast	4	0	0	
Temperature	hot	2	2	0.5	0.44
	cool	3	1	0.375	
	mild	4	2	0.444	
Humidity	high	3	4	0.490	0.368
	normal	6	1	0.245	
Windy	false	6	2	0.375	0.429
	true	3	3	0.5	

5.4 Continuous Value and Missing Value

- Bi-partition (二分法) for continuous value
  - Sort *n* distinct values on a continuous attribute a $\{a^1, a^2, \ldots, a^n\}$
  - Split *D* into D_t^+ and D_t^- w.r.t. a splitting point *t*
  - Candidate

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leqslant i \leqslant n-1 \right\}$$

  - Choose the best

$$\begin{aligned} \mathrm{Gain}(D, a) &= \max_{t \in T_a} \mathrm{Gain}(D, a, t) \\ &= \max_{t \in T_a} \mathrm{Ent}(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} \mathrm{Ent}(D_t^\lambda) \end{aligned}$$

1. (10 points) Consider the following data set. Each training example has the form of (*x*, *y*). The input *x* has three attributes : *refund* (categorical), *marital status* (categorical), *taxable income* (continuous). The output *y* = *cheat* belongs to the set {yes, no}.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

a) (4 points) According to the information gain, which attribute is the first attribute chosen to construct the best decision tree ? Why ?

b) (4 points) Draw the whole decision tree learned from the data set.

c) (2 points) Given the following testing example, what is the value of *cheat* according to the learned decision tree ?

$$\mathrm{Gain}(D,a)=\mathrm{Ent}(D)-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\mathrm{Ent}(D^v)$$

, where

$$\mathrm{Ent}(D)=-\sum_{k=1}^{|\mathcal{Y}|}p_k\log_2 p_k$$

Here, $\mathrm{Ent}(D)=-\left(\frac{3}{10}\log\frac{3}{10}+\frac{7}{10}\log\frac{7}{10}\right)=0.8813$

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Ent($D_{refund=yes}$)=0, Ent($D_{refund=no}$)=0.9852

Gain(D, *Refund*)=0.88-(0.3×0+0.7×0.99)=0.1917

Ent($D_{marital\ status=single}$)=1, Ent($D_{marital\ status=married}$)=0, Ent($D_{marital\ status=divorced}$)=1

Gain(D, *marital status*)=0.88-(0.4×1+0.4×0+0.2×1)=0.2813

For the *taxable income* attribute, since it is continuous, we should determine the best splitting value.

First, sort all values on *taxable income*:

{60, 70, 75, 85, 90, 95, 100, 120, 125, 220}

Second, according to the bi-partition method, candidates of splitting points are:

T = {65, 72.5, 80, 87.5, 92.5, 107.5, 110, 122.5, 127.5}

The formula is:

$$\mathrm{Gain}(D,a)=\max_{t\in T_a}\ \mathrm{Gain}(D,a,t)$$
$$=\max_{t\in T_a}\ \mathrm{Ent}(D)-\sum_{\lambda\in\{-,+\}}\frac{|D_t^\lambda|}{|D|}\mathrm{Ent}(D_t^\lambda)$$

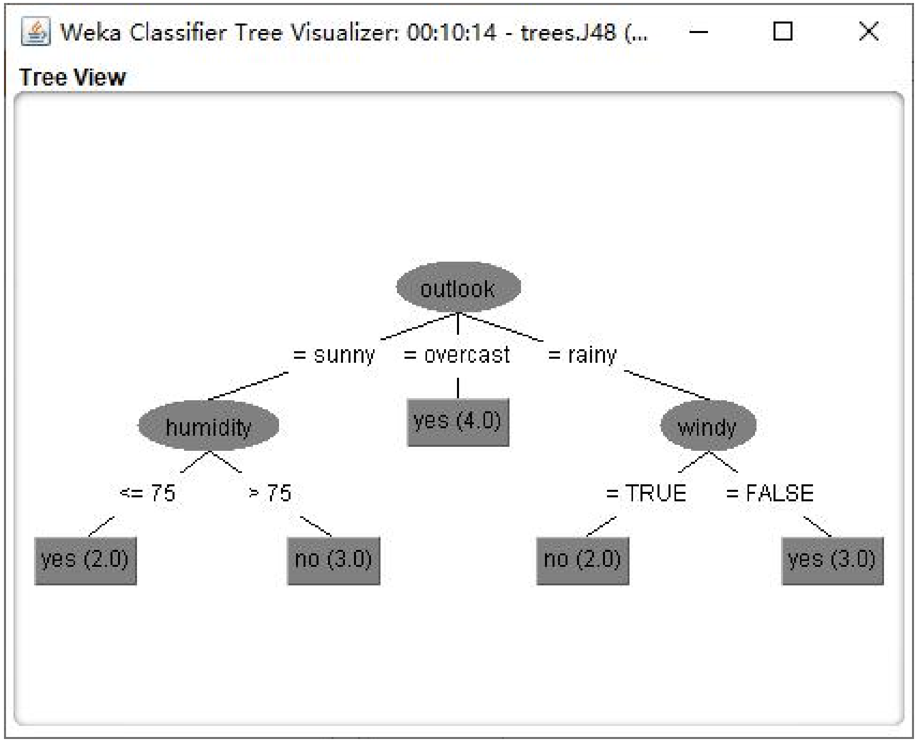
Gain(D, "Taxable Income", t=65)=0.0548
Gain(D, "Taxable Income", t=72.5)=0.1178
Gain(D, "Taxable Income", t=80)=0.1916
Gain(D, "Taxable Income", t=87.5)=0.0058
Gain(D, "Taxable Income", t=92.5)=0.0348
Gain(D, "Taxable Income", t=97.5)=0.2813
Gain(D, "Taxable Income", t=110)=0.1916
Gain(D, "Taxable Income", t=122.5)=0.1178
Gain(D, "Taxable Income", t=127.5)=0.0548
The best splitting point is t = 97.5,
Therefore, the first partition attribute is *marital status* or Taxable Income < 97.5.

0.88-[0.9*(-3/9log3/9-6/9log6/9)+0.1*0]=0.0548
0.88-[0.8*(-5/8log5/8-3/8log3/8)+0.2*0]=0.1178

```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature numeric
@attribute humidity numeric
@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
```




```
=== 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
```

5.3 Continuous Value and Missing Value

- Weight for missing value
  - $\tilde{D} \subseteq D$: set where each sample has a value on *a*
  - $\tilde{D}^V \subseteq \tilde{D}$: set where each sample takes value a^V on *a*
  - $\tilde{D}_k \subseteq \tilde{D}$: set where each sample belongs to the k^{th} class

$$\tilde{D} = \bigcup_{k=1}^{|\mathcal{Y}|} \tilde{D}_k, \qquad \tilde{D} = \bigcup_{v=1}^{V} \tilde{D}^v.$$

$$\rho = \frac{\sum_{\boldsymbol{x} \in \tilde{D}} w_{\boldsymbol{x}}}{\sum_{\boldsymbol{x} \in D} w_{\boldsymbol{x}}},$$

$$\tilde{p}_k = \frac{\sum_{\boldsymbol{x} \in \tilde{D}_k} w_{\boldsymbol{x}}}{\sum_{\boldsymbol{x} \in \tilde{D}} w_{\boldsymbol{x}}} \quad (1 \leqslant k \leqslant |\mathcal{Y}|),$$

$$\tilde{r}_v = \frac{\sum_{\boldsymbol{x} \in \tilde{D}^v} w_{\boldsymbol{x}}}{\sum_{\boldsymbol{x} \in \tilde{D}} w_{\boldsymbol{x}}} \quad (1 \leqslant v \leqslant V).$$

$$\sum_{k=1}^{|\mathcal{Y}|} \tilde{p}_k = 1, \sum_{v=1}^{V} \tilde{r}_v = 1.$$

5.3 Continuous Value and Missing Value

- Information gain:

$$\mathrm{Gain}(D,a) = \rho \times \mathrm{Gain}(\tilde{D},a)$$

$$\mathrm{Ent}(\tilde{D}) = -\sum_{k=1}^{|\mathcal{Y}|} \tilde{p}_k \log_2 \tilde{p}_k$$

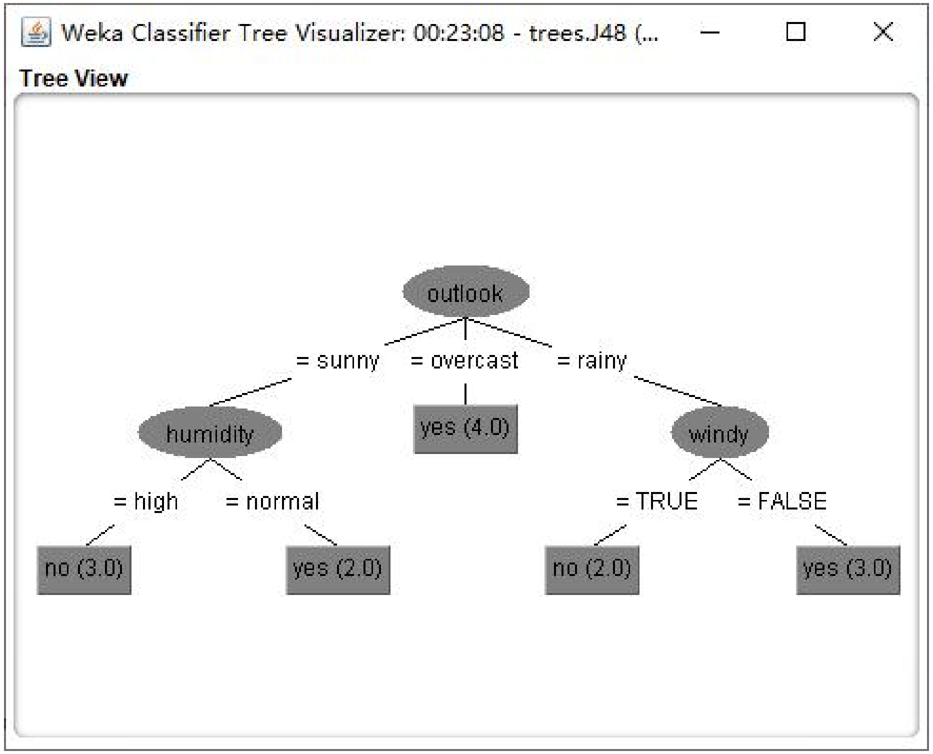
$$= \rho \times \left(\mathrm{Ent}\left(\tilde{D}\right) - \sum_{v=1}^{V} \tilde{r}_v \,\mathrm{Ent}\left(\tilde{D}^v\right)\right)$$

- Reset the *weight* w_x *of sample* x if need:
  - If x has some value on a, just keep w_x
  - Otherwise, first join x into each node corresponding to a^v and then set the weight of x to $\tilde{r}_v \cdot w_x$

```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,?,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,?,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,?,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```




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

Correctly Classified Instances           8               57.1429 %
Incorrectly Classified Instances         6               42.8571 %
Kappa statistic                          0.1429
Mean absolute error                      0.369
Root mean squared error                  0.5405
Relative absolute error                 77.5    %
Root relative squared error            109.5471 %
Total Number of Instances               14
```

```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
?,?,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,?,high,FALSE,yes
?,mild,high,FALSE,yes
?,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
?,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,?,normal,FALSE,yes
?,mild,normal,TRUE,yes
?,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```


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

Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                         -0.1395
Mean absolute error                      0.5403
Root mean squared error                  0.5727
Relative absolute error                113.4615 %
Root relative squared error            116.0707 %
Total Number of Instances               14
```

5.4 Random Forest

Algorithm 15.1 *Random Forest for Regression or Classification.*

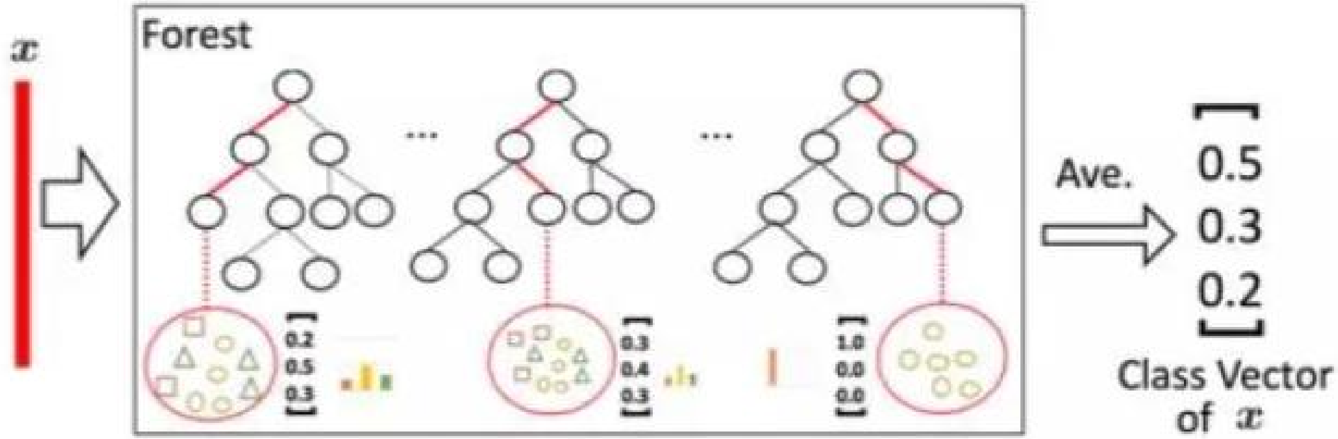
1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

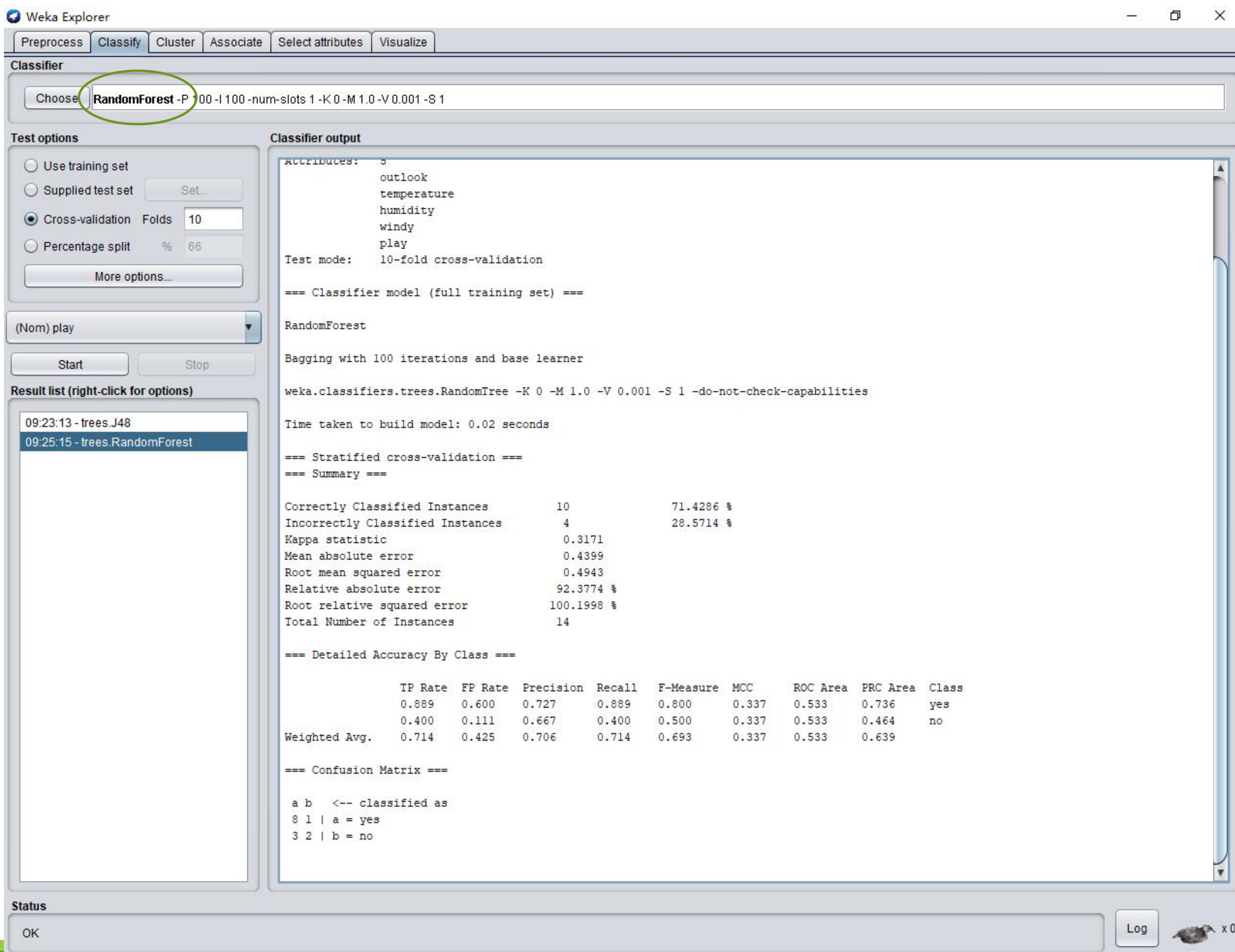
To make a prediction at a new point x:

Regression: $\hat{f}_{\mathrm{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\mathrm{rf}}^B(x) = \textit{majority vote } \{\hat{C}_b(x)\}_1^B$.

Example




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) play
Start
Stop
Result list (right-click for options)
09:23:13 - trees.J48
09:25:15 - trees.RandomForest
Classifier output
outlook
temperature
humidity
windy
play
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
RandomForest
Bagging with 100 iterations and base learner
weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities
Time taken to build model: 0.02 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 10 71.4286 %
Incorrectly Classified Instances 4 28.5714 %
Kappa statistic 0.3171
Mean absolute error 0.4399
Root mean squared error 0.4943
Relative absolute error 92.3774 %
Root relative squared error 100.1998 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.889 0.600 0.727 0.889 0.800 0.337 0.533 0.736 yes
0.400 0.111 0.667 0.400 0.500 0.337 0.533 0.464 no
Weighted Avg. 0.714 0.425 0.706 0.714 0.693 0.337 0.533 0.639
=== Confusion Matrix ===
a b <-- classified as
8 1 | a = yes
3 2 | b = no
Status
OK
Log
x 0

4.4 Random Forests

- RF vs. other ensembles