

Creating a CHAID classification tree with XLSTAT

[demoTree.xls](#)

Dataset for creating a CHAID classification tree

An Excel sheet containing both the data and the results for use in this tutorial can be downloaded by clicking [here](#).

The data are from [Fisher M. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7, 179 -188] and correspond to 150 Iris flowers, described by four variables (sepal length, sepal width, petal length, petal width) and their species. Three different species have been included in this study: setosa, versicolor and virginica.



Iris setosa, versicolor and virginica.

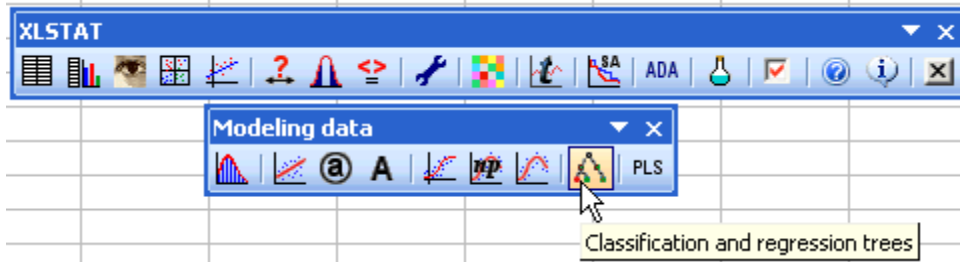
Goal of this CHAID classification tree

Our goal is to test if the four descriptive variables allow to efficiently predict to which species a flower corresponds, and in this case, to identify rules that would help classifying the flowers on the basis of the four variables.

Note: the same case is treated in the tutorial on discriminant analysis.

Setting up the dialog box to generate a CHAID classification tree

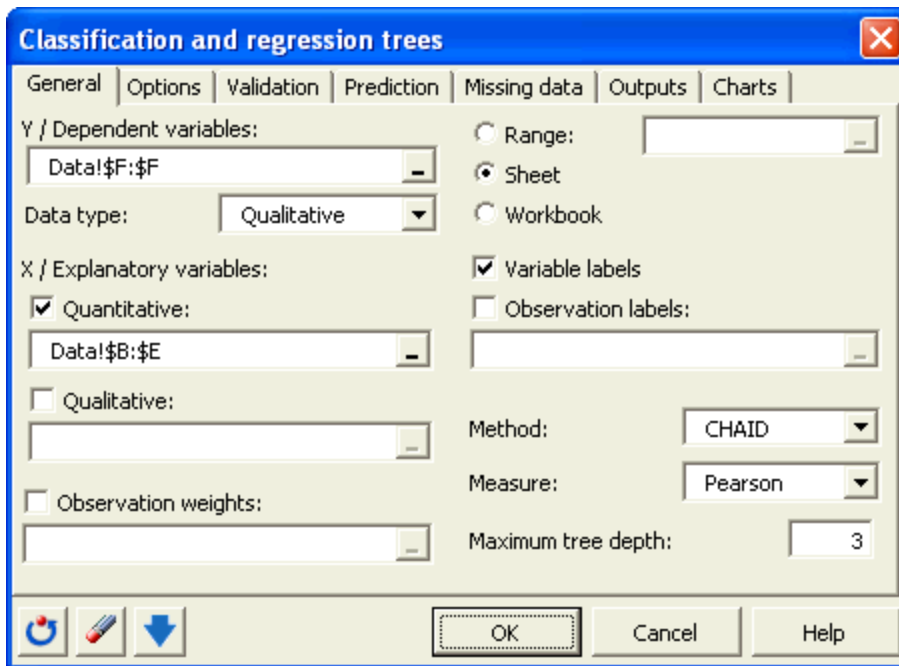
After opening XLSTAT, select the **XLSTAT / Modeling data / Classification and regression trees** command, or click the corresponding button of the **Modeling data** toolbar (see below).



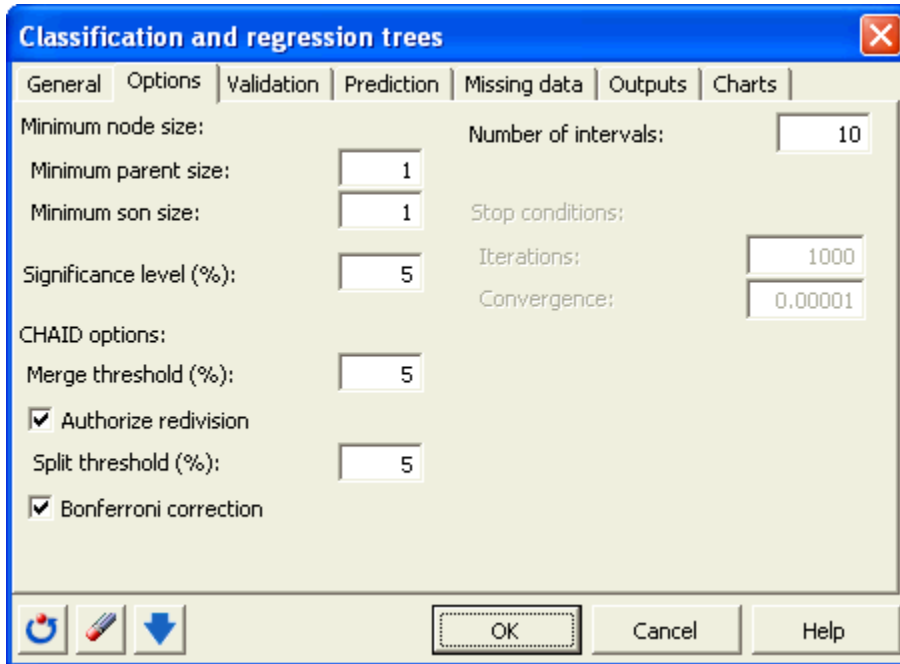
Once you've clicked the button, the dialog box appears. The **qualitative dependent variable** corresponds here to the "Species" variable.

The quantitative **Explanatory variables** are the four descriptive variables.

We choose to use the **CHAID algorithm** and we set the maximum tree depth to 3 to avoid obtaining a too complex tree.

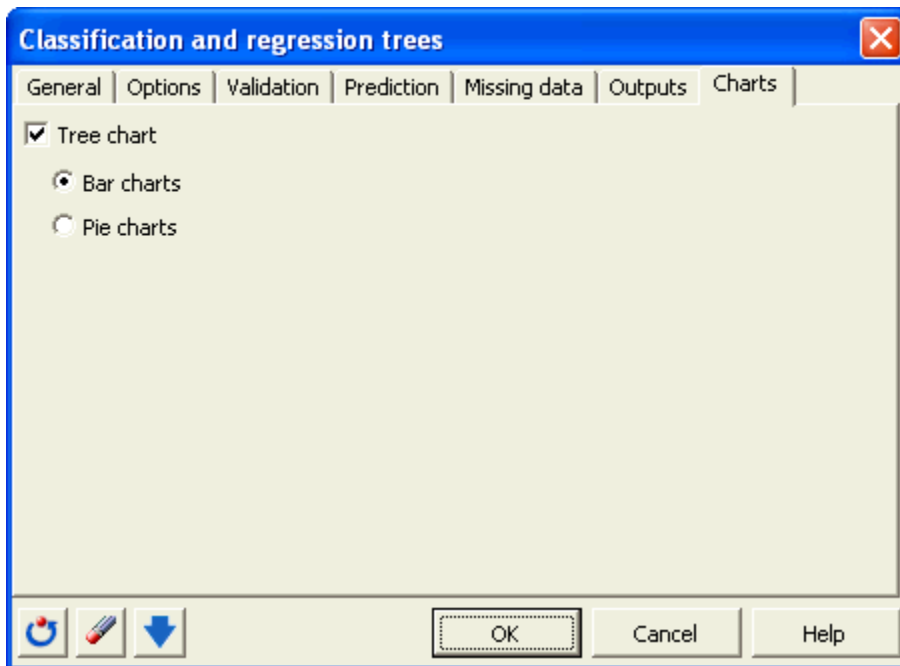


In the **Options** tab, several technical options allow to better control the way the tree is built.



In **Charts** tab we first select the **Bar charts** option to display the distribution of the species at each node.

As we will see later, the **Pie charts** option is also being used in this tutorial.



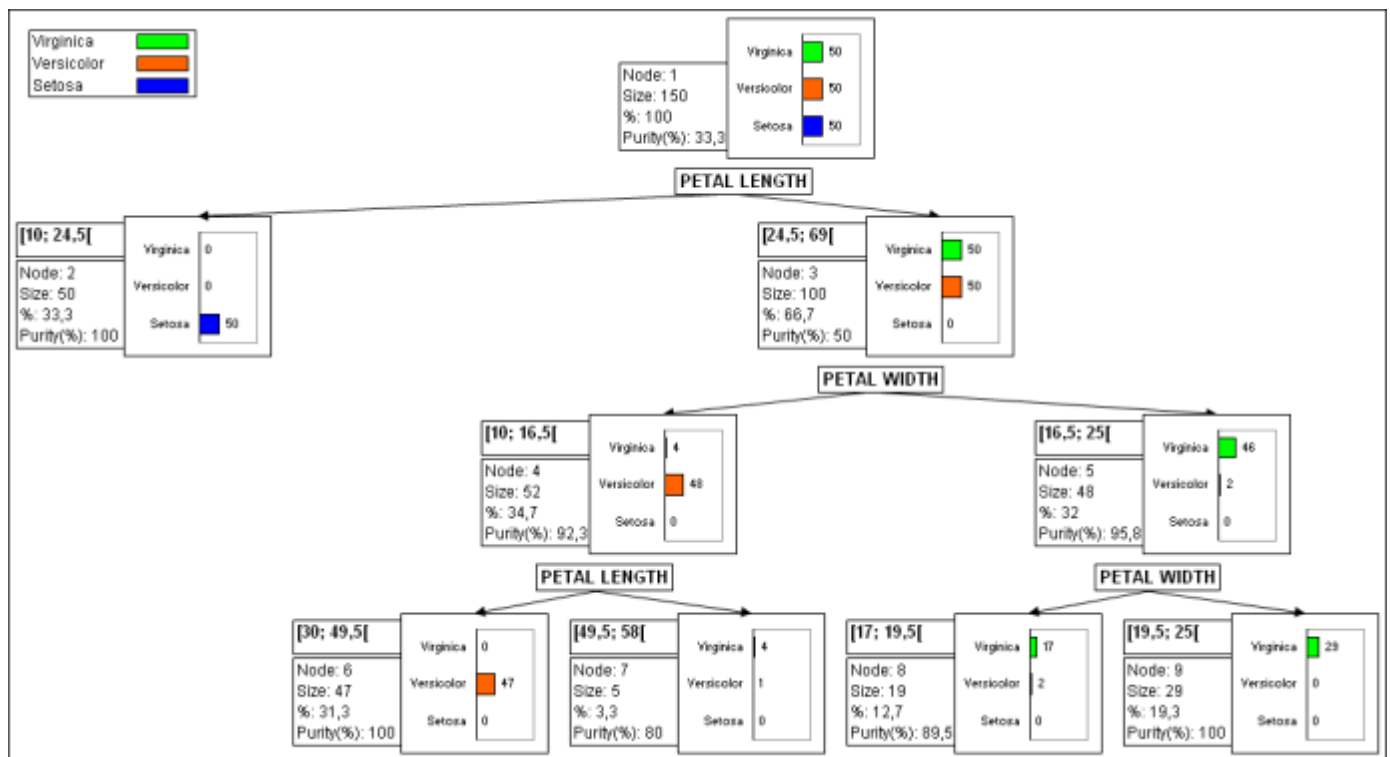
The computations begin once you have clicked on **OK**. The results will then be displayed.

Interpreting the results of a CHAID classification tree

Below the simple statistics for all the selected variables, XLSTAT displays information on the tree structure. This includes for each node, the p-value for the splitting, the number of objects at each node, the corresponding % the parent and son nodes, the split variable, the value(s) or intervals of the latter, and the purity that indicates what is the % of objects that belong to the dominating category of the dependent variable at this node.

Tree structure:								
Node	p-value	Objects	%	Parent node	Sons	Split variable	Values	Purity
1	0,841	150	100,00%		2; 3			33,33%
2	0,000	50	33,33%	1		PETAL LENG	[10; 24,5[100,00%
3	0,881	100	66,67%	1	4; 5	PETAL LENG	[24,5; 69[50,00%
4	0,885	52	34,67%	3	6; 7	PETAL WIDT	[10; 16,5[92,31%
5	0,258	48	32,00%	3	8; 9	PETAL WIDT	[16,5; 25[95,83%
6	0,000	47	31,33%	4		PETAL LENG	[30; 49,5[100,00%
7	0,000	5	3,33%	4		PETAL LENG	[49,5; 58[80,00%
8	0,000	19	12,67%	5		PETAL WIDT	[17; 19,5[89,47%
9	0,000	29	19,33%	5		PETAL WIDT	[19,5; 25[100,00%

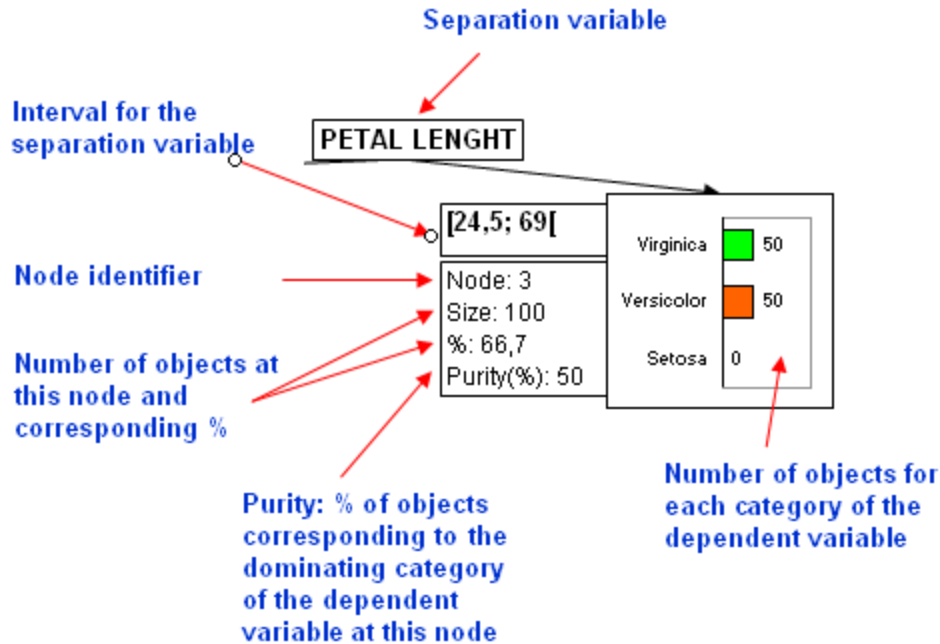
The next result displayed is the classification tree.



This diagram allows to visualize the successive steps during which the CHAID algorithm identifies the variables that allow to best split the categories of the dependent variable. Thus, we see that using only the petal length, the algorithm has found a rule that allows to perfectly separate

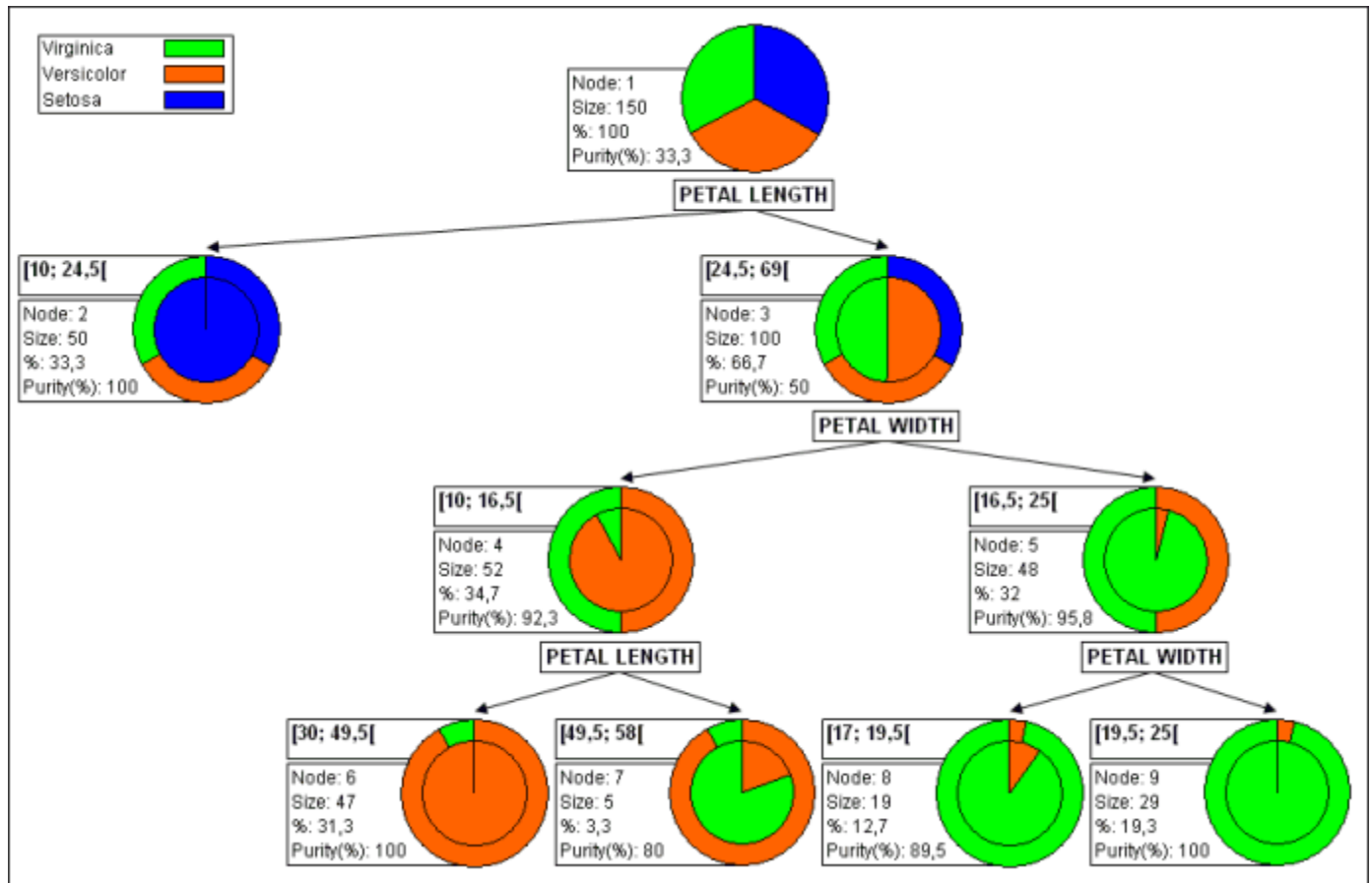
the Iris flowers of the setosa species. If the petal length is between 10 and 24.5 then the species is setosa.

The information available at each node is explained below.



The algorithm stops when no additional rule can be found, or when one of the limits set by the user are reached (number of objects at a parent or son node, maximum tree depth, threshold p-value for splitting).

XLSTAT offers a second possibility to visualize the classification trees. Instead of using bar charts, it uses pie charts. The latter are easier to read when they are many nodes and many categories for the dependent variable. The inner circle of the pie corresponds to the relative frequencies of the categories to which the objects contained in the node correspond. The outer ring shows distribution of the categories at the parent node.



The following table contains the rules built by the algorithm in a less visual but more readable way: the rules are written in natural language. The purity gives the % that corresponds to the majority category at the node level. The number of objects corresponding to the category is also displayed.

Node	red(SPECIE)	frequency	Purity	Rules
Node1	Setosa	50	33,33%	
Node2	Setosa	50	100,00%	If PETAL LENGHT in [10; 24,5[then SPECIE = Setosa in 100% of cases
Node3	Versicolor	50	50,00%	If PETAL LENGHT in [24,5; 69[then SPECIE = Versicolor in 50% of cases
Node4	Versicolor	48	92,31%	If PETAL WMDTH in [10; 16,5[and PETAL LENGHT in [24,5; 69[then SPECIE = Versicolor in 92,3% of cases
Node5	Virginica	46	95,83%	If PETAL WMDTH in [16,5; 25[and PETAL LENGHT in [24,5; 69[then SPECIE = Virginica in 95,8% of cases
Node6	Versicolor	47	100,00%	If PETAL LENGHT in [30; 49,5[and PETAL WMDTH in [10; 16,5[then SPECIE = Versicolor in 100% of cases
Node7	Virginica	4	80,00%	If PETAL LENGHT in [49,5; 58[and PETAL WMDTH in [10; 16,5[then SPECIE = Virginica in 80% of cases
Node8	Virginica	17	89,47%	If PETAL WMDTH in [17; 19,5[and PETAL LENGHT in [24,5; 69[then SPECIE = Virginica in 89,5% of cases
Node9	Virginica	29	100,00%	If PETAL WMDTH in [19,5; 25[and PETAL LENGHT in [24,5; 69[then SPECIE = Virginica in 100% of cases

In this way, we see that

"If PETAL LENGHT is in the interval [30; 49.5[and PETAL WIDTH is in the interval [10; 16.5[then SPECIES is Versicolor in 100% of cases"

this rule is verified by 47 flowers.

The rules that correspond to the leaves of the tree (the terminal nodes) allow to compute predictions for each observation, with a probability that depends on the distribution of the categories at the leaf level. These results are displayed in the "Results by object" table.

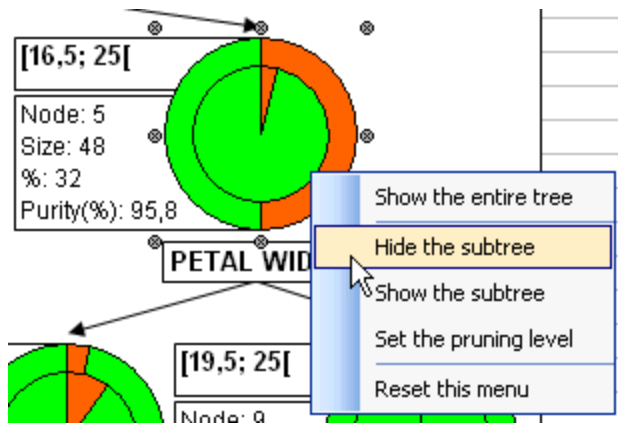
Results by object:					
Observation	Prior	Posterior	Pr(Setosa)	Pr(Versicolor)	Pr(Virginica)
Obs1	Setosa	Setosa	1,000	0,000	0,000
Obs2	Virginica	Virginica	0,000	0,000	1,000
Obs3	Versicolor	Versicolor	0,000	1,000	0,000
Obs4	Virginica	Virginica	0,000	0,000	1,000
Obs5	Virginica	Virginica	0,000	0,200	0,800
Obs6	Setosa	Setosa	1,000	0,000	0,000
Obs7	Virginica	Virginica	0,000	0,000	1,000
Obs8	Versicolor	Versicolor	0,000	1,000	0,000
Obs9	Versicolor	Virginica	0,000	0,105	0,895
Obs10	Setosa	Setosa	1,000	0,000	0,000
Obs11	Versicolor	Versicolor	0,000	1,000	0,000
Obs12	Versicolor	Virginica	0,000	0,200	0,800
Obs13	Virginica	Virginica	0,000	0,000	1,000
Obs14	Setosa	Setosa	1,000	0,000	0,000
Obs145	Virginica	Virginica	0,000	0,000	1,000
Obs146	Setosa	Setosa	1,000	0,000	0,000
Obs147	Virginica	Virginica	0,000	0,000	1,000
Obs148	Versicolor	Virginica	0,000	0,105	0,895
Obs149	Virginica	Virginica	0,000	0,000	1,000
Obs150	Setosa	Setosa	1,000	0,000	0,000

We see that 3 observations have been miss-classified by the algorithm. This result is almost identical to what is obtained with a discriminant analysis where the miss-classified observations are 5, 9, 12.

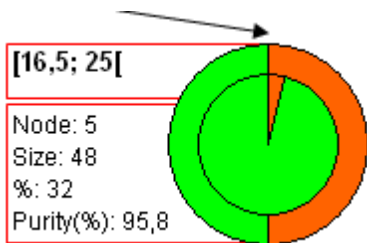
The confusion matrix summarizes the reclassification of the observations, and allows to quickly see the % of well classified observations, which is the ratio of the number of observations that have been well classified over the total number of observations. It is here equal to 98%.

Confusion matrix for the estimation sample:					
from \ to	Setosa	Versicolor	Virginica	Total	% correct
Setosa	50	0	0	50	100,00%
Versicolor	0	47	3	50	94,00%
Virginica	0	0	50	50	100,00%
Total	50	47	53	150	98,00%

The trees created by XLSTAT are partially dynamic. You can prune the tree at a given level for all branches, or you can prune only one given branch. To prune the tree you first need to click on a node. When the six grey dots appear around the node, right click the mouse to display the contextual menu:



If we decide to hide a subtree, the tree is then re-created without the branches starting from the selected node. The contours of the node are displayed in red color.



It is of course possible afterwards to display again the hidden subtree using the same contextual menu.