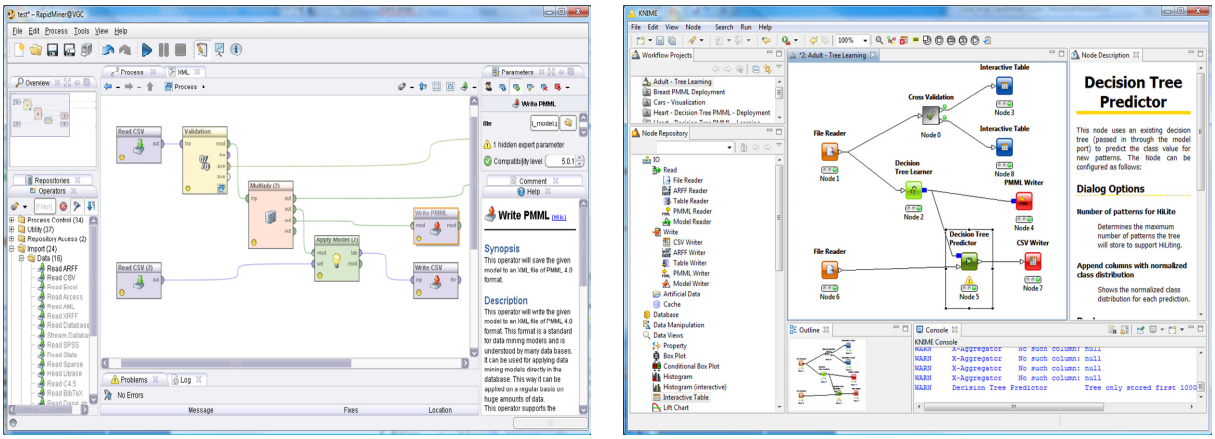
1 Theme

Description of the RapidMiner 5.0 GUI [This tutorial is a translation of the French version posted at 2010/10/04. The current version of RapidMiner at this time (2011/09/20) is 5.1.11.].

RapidMiner is a very popular data mining tool. It is (one of) the most used by the data miners according to the annual Kdnuggets polls (2011, 2010, 2009, 2008, 2007). There are two versions. We describe here the Community Edition which freely downloadable from the editor's website[1].



RapidMiner Knime

Figure 1 – Workspaces of RapidMiner 5.0 and Knime

The new RapidMiner 5.0 has a new graphical user interface which is very similar to that of Knime (Figure 1). The organization of the workspace is the same. The sequence of data processing (using operators) is described with a diagram called "process" into the RapidMiner documentation. In fact, this version 5.0 joined the presentation adopted by the vast majority of data mining software. Some features are shared with many tools, among others: the connection to the R software; the meta-nodes which implements a loop or a standard succession of operations; the description of the methods underlying operators which is continuously in the right part of the main window.

RapidMiner 5.0 having evolved substantially (compared with previous versions e.g. the version 4.6 described in one of our tutorials -- http://data-mining-tutorials.blogspot.com/2010/04/wrapper-for-feature-selection.html). I thought it was appropriate to study this in detail, evaluating its behavior in the context of a standard data mining analysis. We want to implement the following process: (1) creating a decision tree from a labeled dataset; (2) exporting the model (the classification tree) into a external file (PMML format) in order to a deployment thereafter; (3) assessing the model performance using a cross-validation resampling scheme; (4) applying the model on a set of unlabeled instances, the results, i.e. the values of the descriptors and the assigned class, must be exported into a CSV file. These are standard data mining tasks. We have described them in many tutorials. We want to check if it is easy to implement them with this new version of RapidMiner. Indeed, with the previous version, defining some sequences of operations was complicated. Implementing a cross-validation for instance was not really intuitive (http://data-mining-tutorials.blogspot.com/2008/11/decision-tree-and-cross-validation.html).

[1] I have not tried the Enterprise Edition. The comparison between the two versions is available online - http://rapid-i.com/content/view/181/190/lang,en/#enterprise

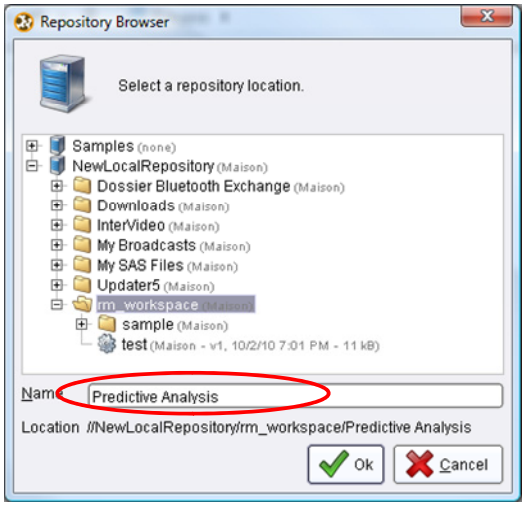
2 Dataset

We use two subsets of the "adult" database (http://archive.ics.uci.edu/ml/datasets/Adult - UCI server). The first, "adult_labeled.csv", contains the labeled instances. The "classe" column corresponds to the labels. The second one, "adult_unlabeled.csv", contains the unlabeled instances that we want to classify with the predictive model.

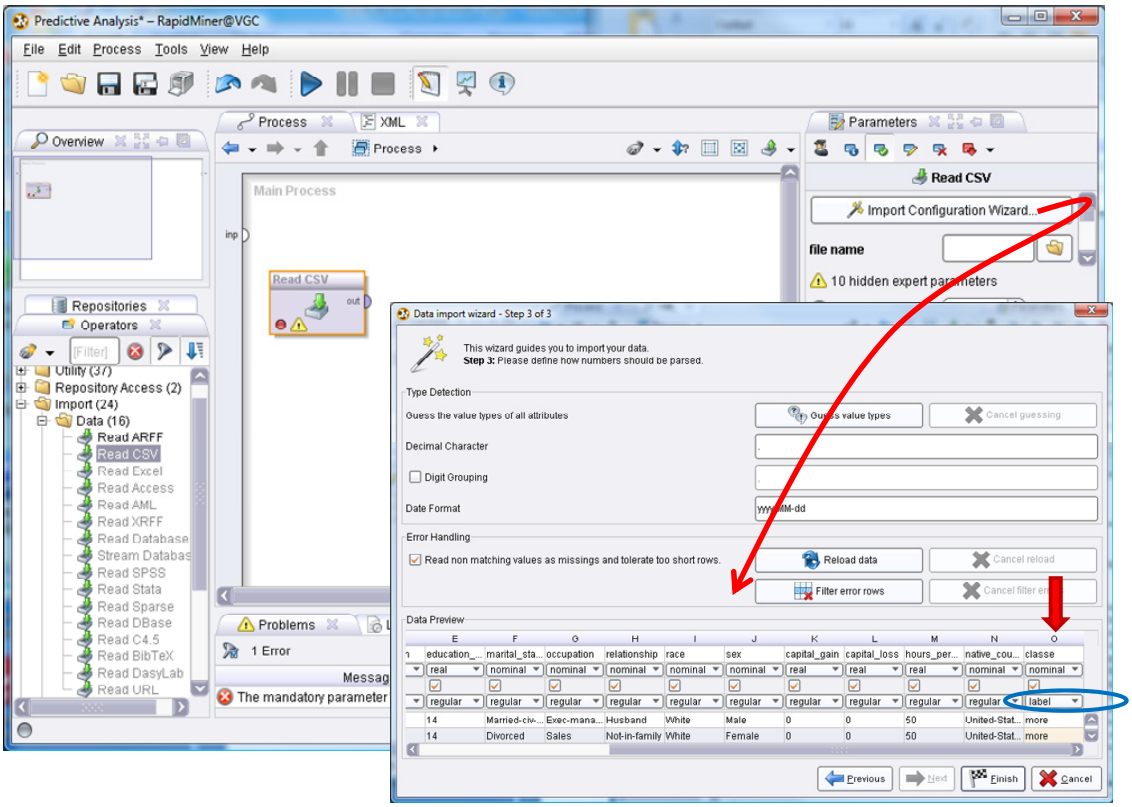
3 Performing a process with RapidMiner 5.0

3.1 Creating a process

We create a new process by clicking on the FILE / NEM menu. Into the settings dialog box, we set the project name "Predictive Analysis". We can now define a new process.



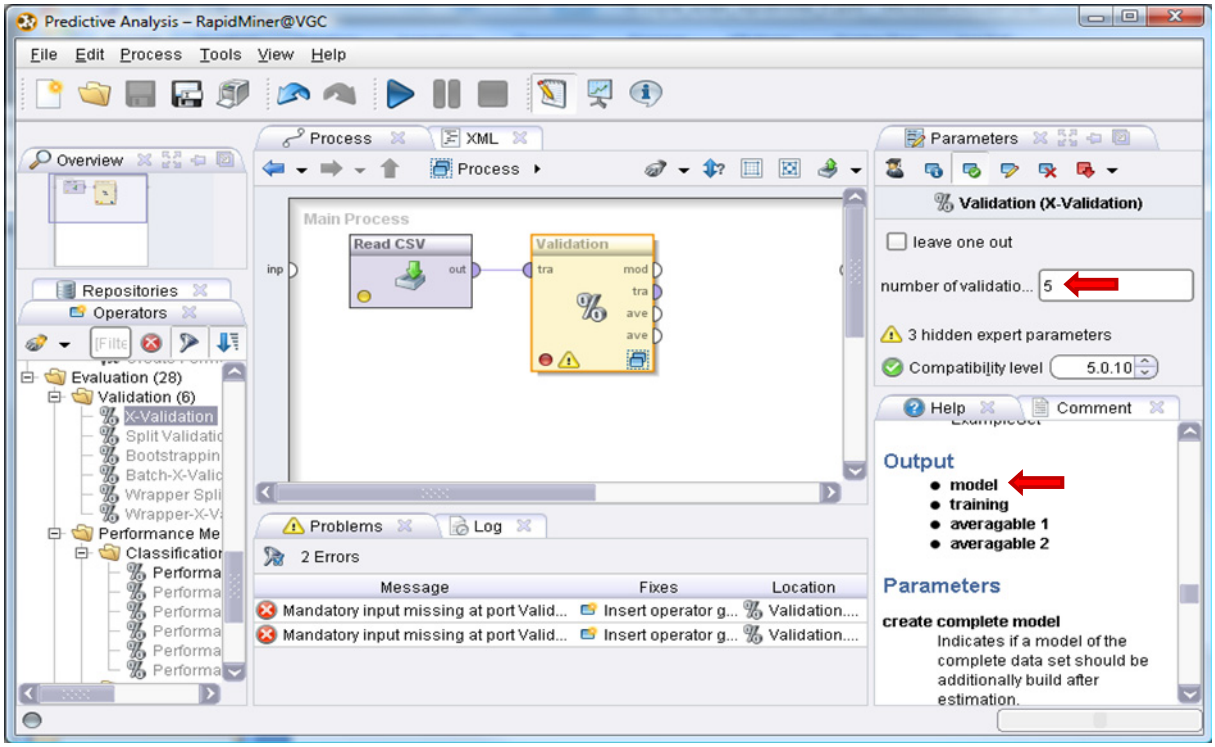
3.2 Importing the labeled dataset



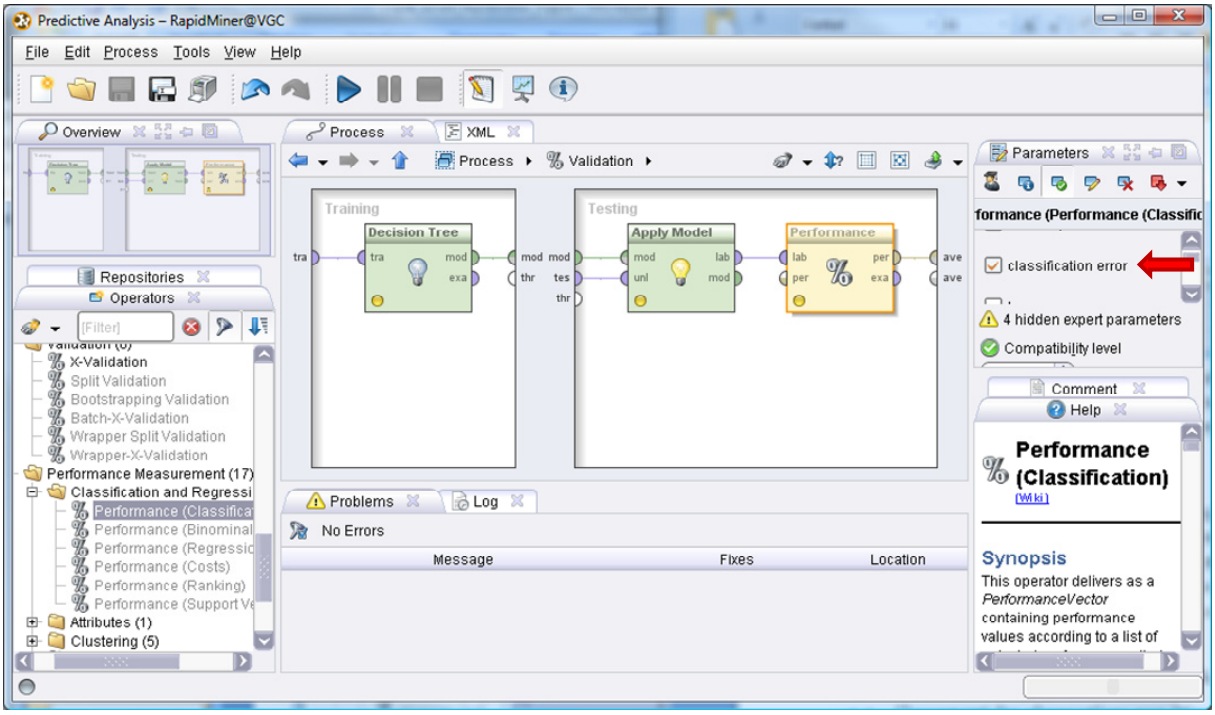
We use the READ CSV component to import the data file. We can set the data file name in the parameters section at the right pane of the main window. But, it seems better to use the importation wizard. Indeed, it allows us to specify the type of the variable, and to select the class attribute. For simplicity, we set as continuous all the numerical columns. The "classe" column contains the labels. We validate our settings by clicking on the FINISH button.

3.3 Classification tree induction - Cross-validation

The construction of the tree and its evaluation using cross-validation are combined into RapidMiner. Instead of insert the decision tree component into the workspace, we insert the X-VALIDATION component. We connect the data source (Read CSV) to this component. We set a **5**-cross validation. We observe that, according to the help panel, we obtain the classification tree learned on the whole dataset as output of the component.

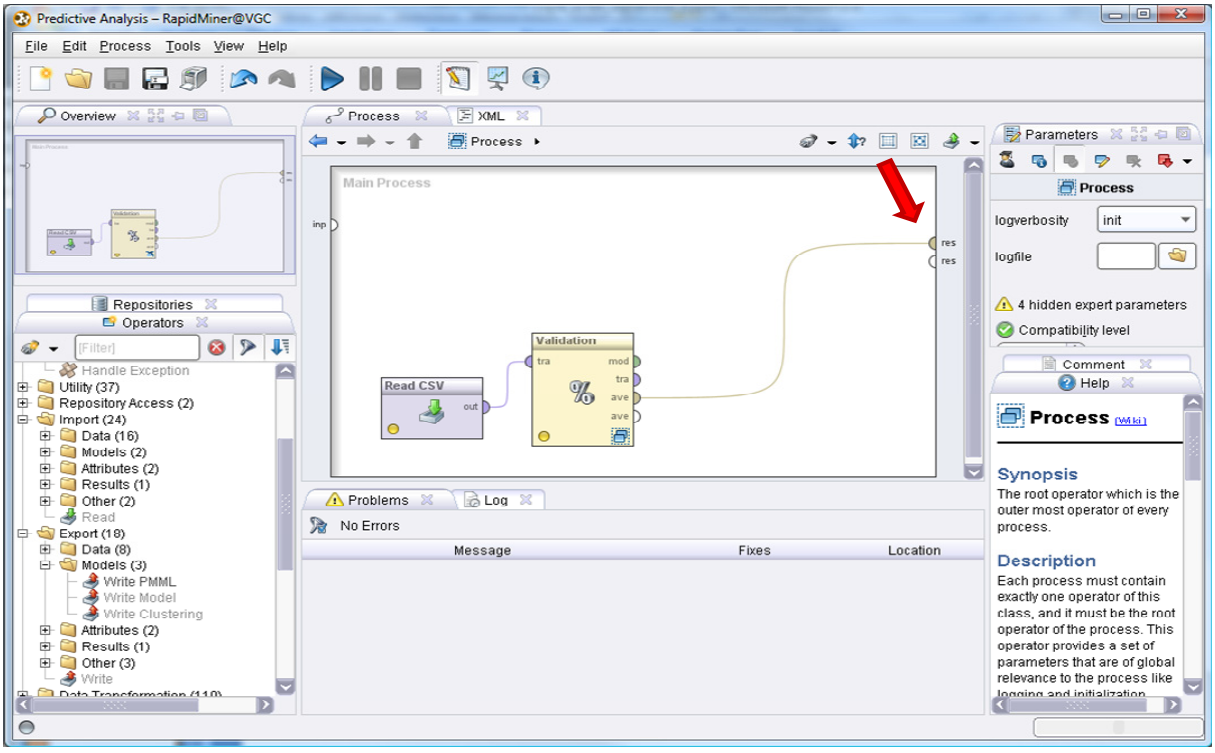


We have to select the learning algorithm and the performance criterion now. To do that, we double-click on the operator. We note that this is actually a meta-node (or a meta-operator) which allows to define a succession of operations.

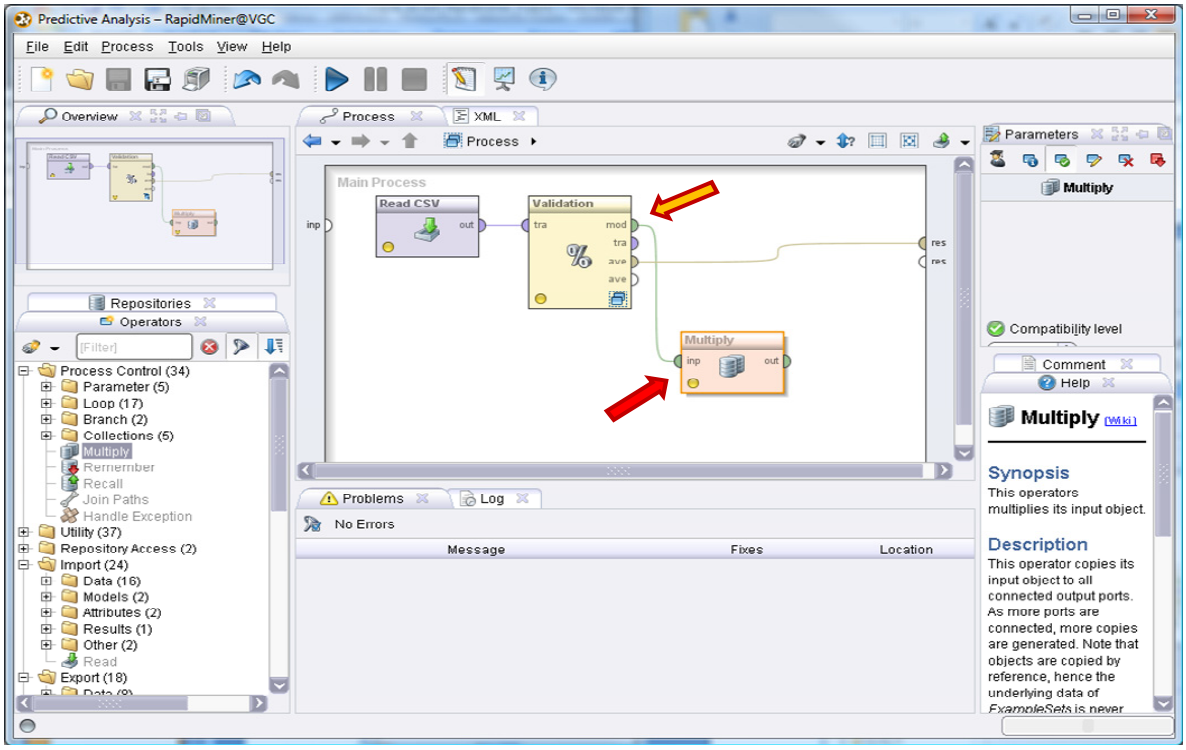


Into the TRAIINING part, we add the DECISION TREE operator. Into the TEST part, we set at the same time the model applier (APPLY MODEL) and the PERFORMANCE measurement tool. We set "classification error" (error rate) as criterion. Note the connections between the various operators. Note also the connections at the input and at the output of the meta-node.

In order to visualize the output of the treatments, we have to connect the cross-validation operator (the third slot) to the result slot (RES) of the diagram.



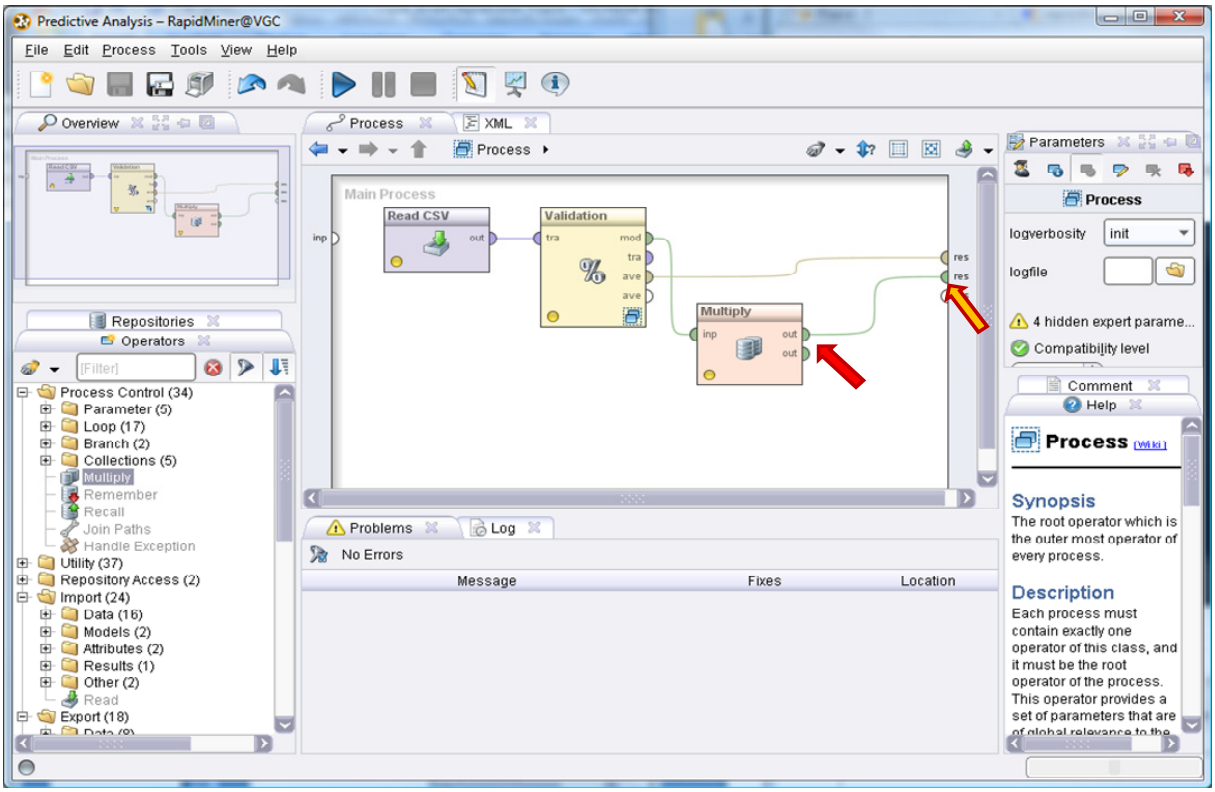
3.4 Duplication of the model



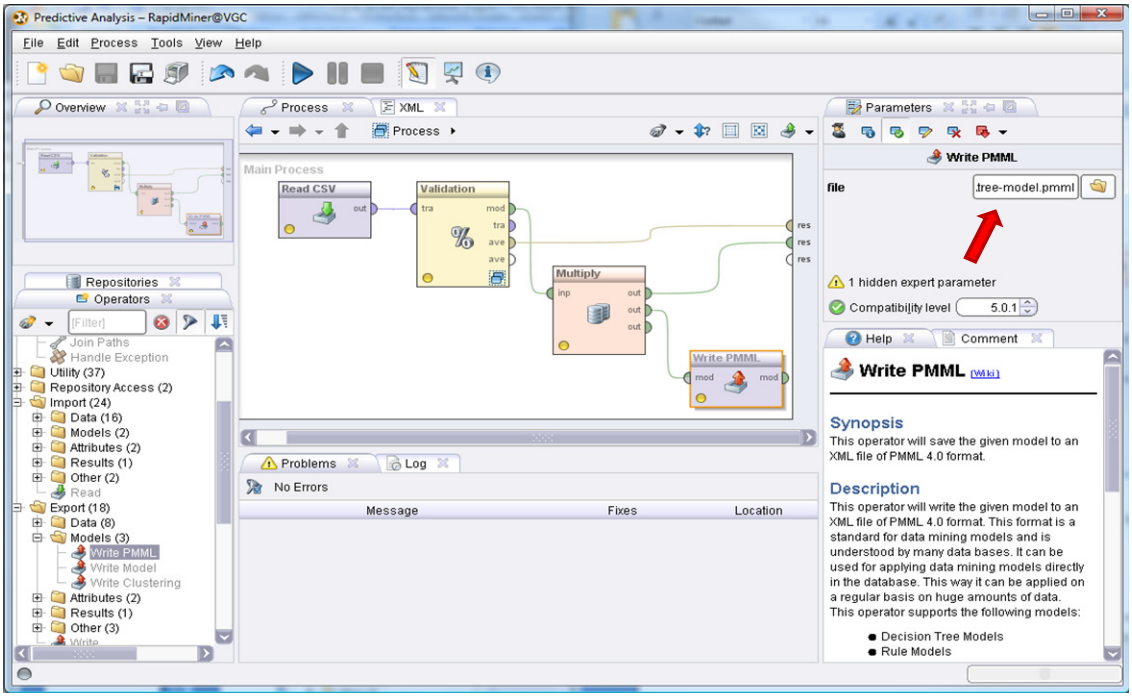
The same tree must be visualized, exported in a PMML format, and applied on unlabeled instances. When an object is used in different ways at the output of an operator, we must duplicate it using the MULTIPLY tool. We note that we use the MOD slot (MOD for model) at the output of VALIDATION.

3.5 Visualizing the classification tree

To visualize the tree, we add a connection from MULTIPLY (model) to the output of the process.



3.6 Storing the tree into a file (PMML format)



We use the WRITE PMML operator to save the tree. We specify the filename.

3.7 Classification of unlabeled instances

To apply the model on unlabeled instances and store the resulting dataset into a data file (CSV file format), we set the succession of the following operators.

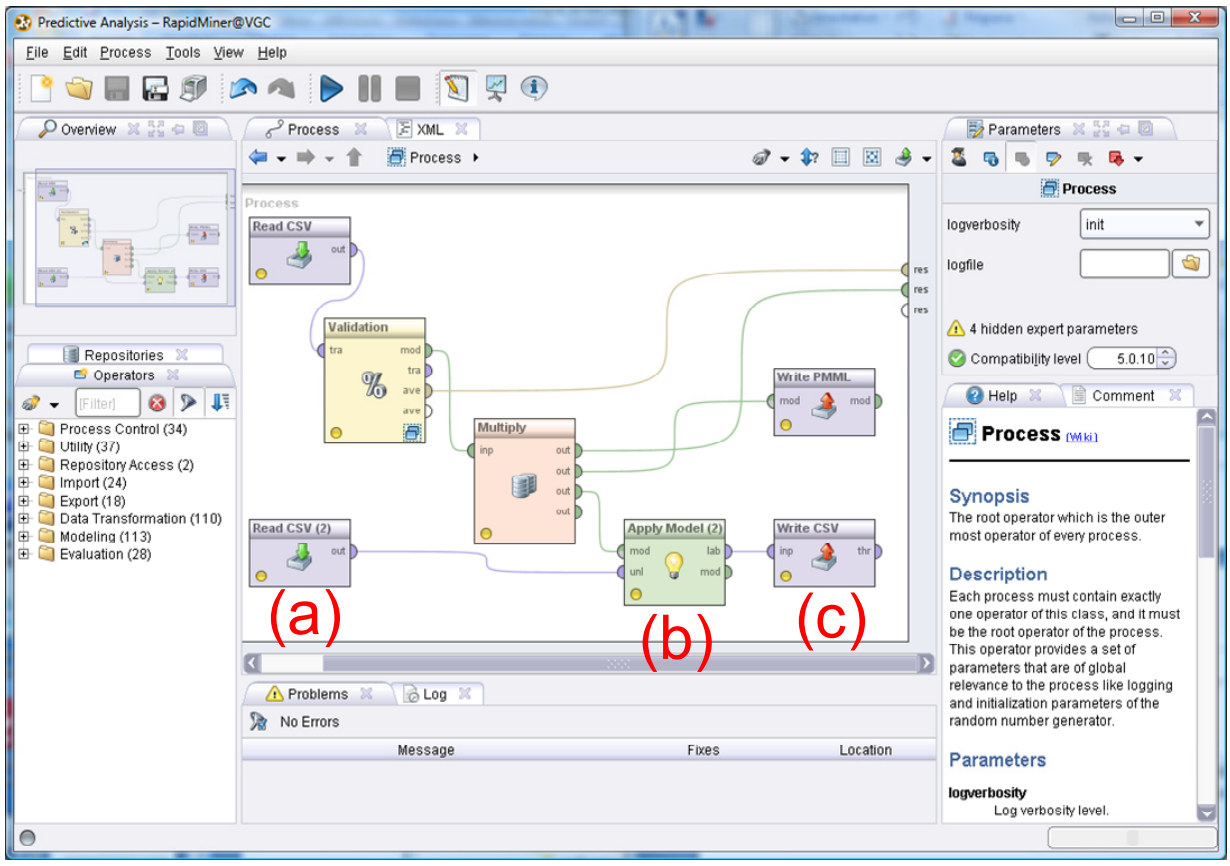


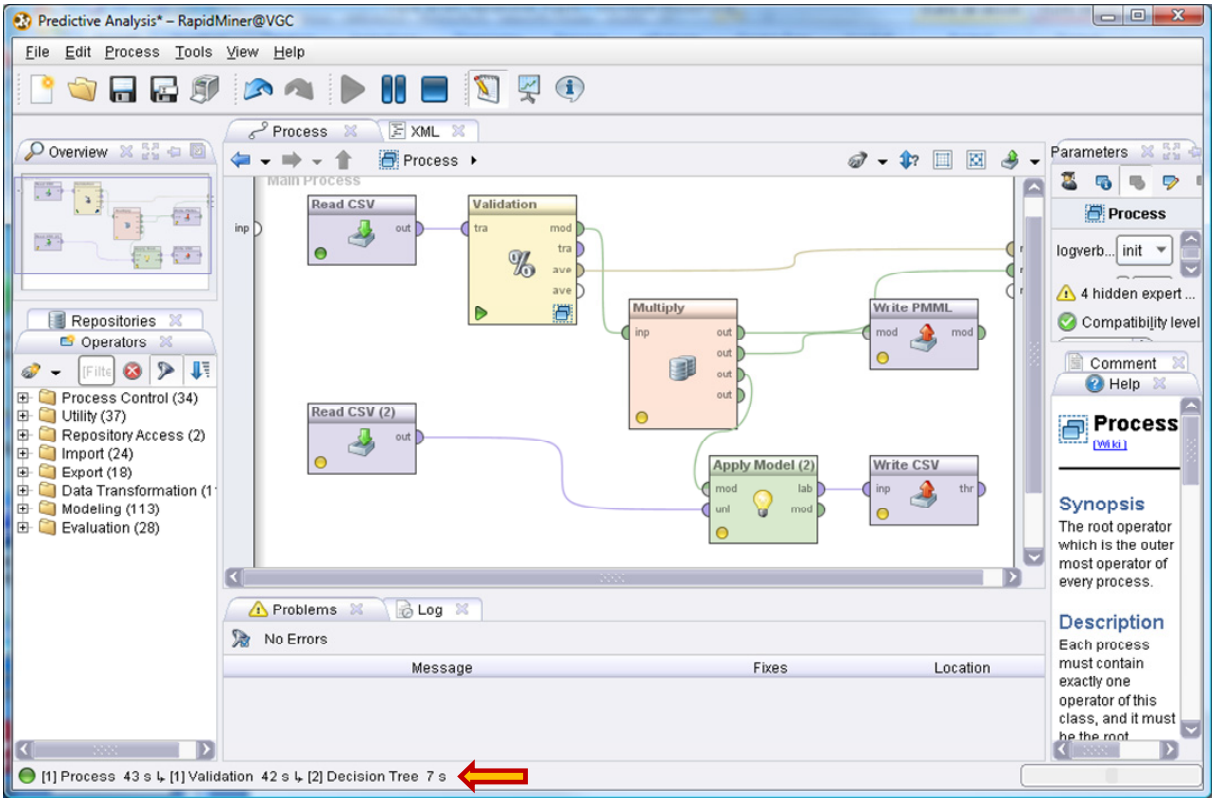
Figure 2 – The whole process with RapidMiner

(a) We import the unlabeled instances "adult_unlabeled.csv" using READ CSV. We must set the appropriate data type (REAL or NOMINAL).
(b) APPLY MODEL allows to apply the model on the unlabeled instances.
(c) WRITE CSV allows to save the new dataset, including the prediction column, into a new data file "adult_with_predictions.csv".

3.8 Launching the calculations

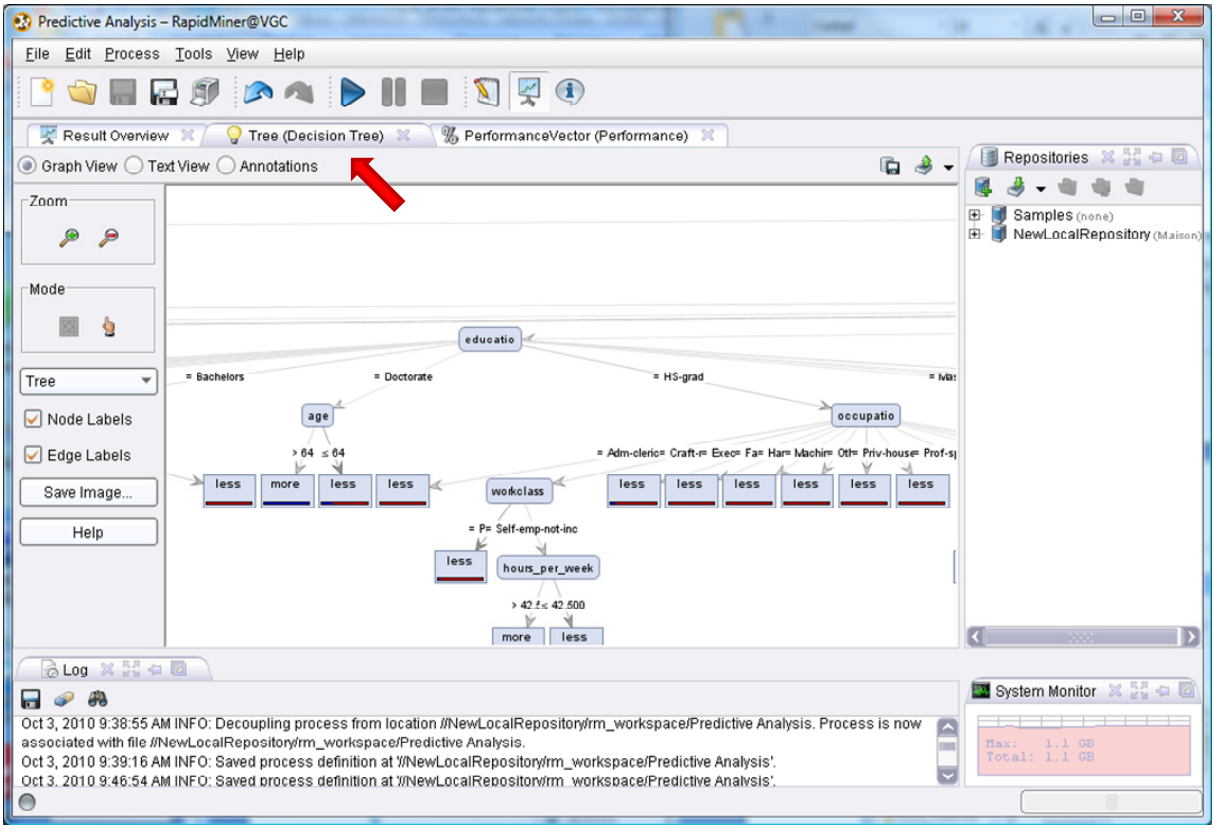
After we save the whole process, we can launch the calculations by clicking on the PROCESS / RUN menu. We can use also the shortcut into the toolbar.

The dataset being large, and the cross-validation greedy in computational resources, the calculation time may be long. We can follow the computations into the status bar of the application.

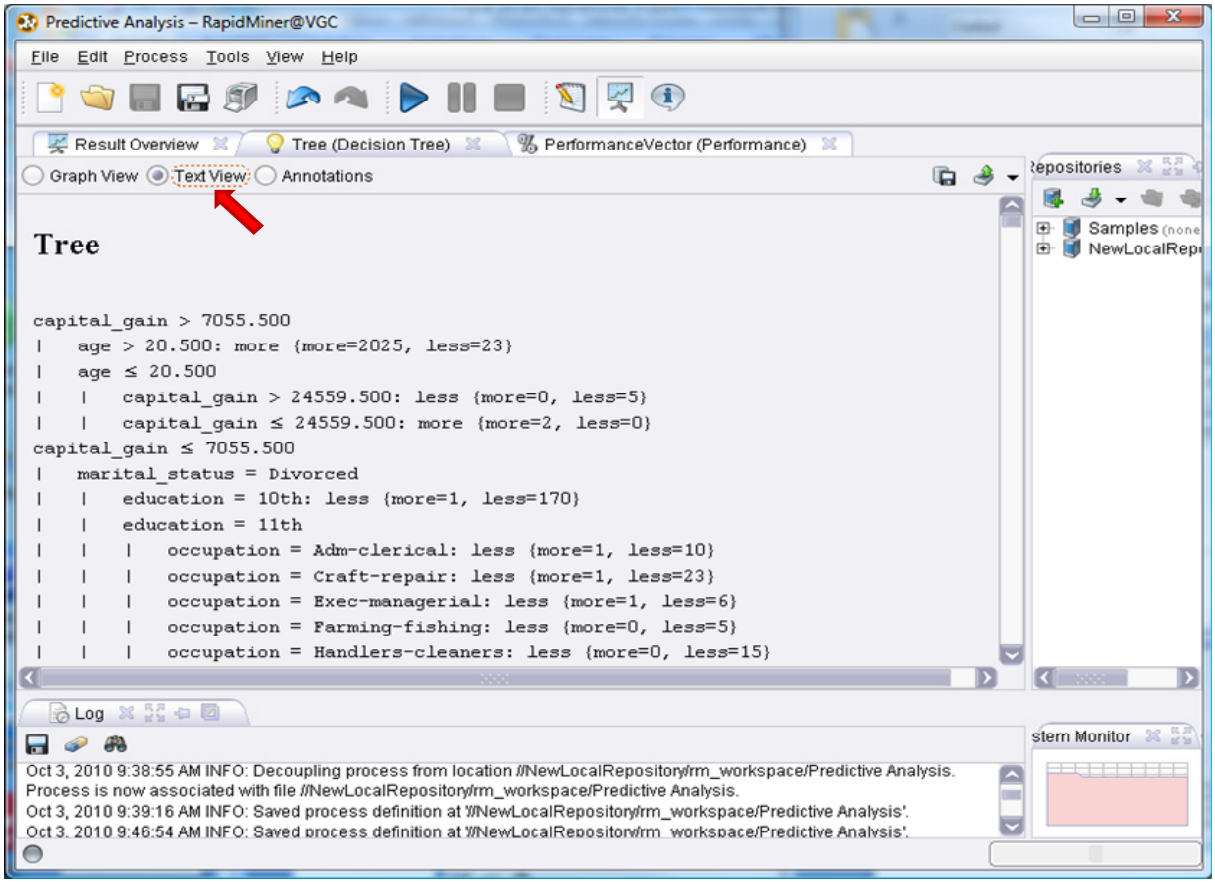


3.9 Visualizing the results

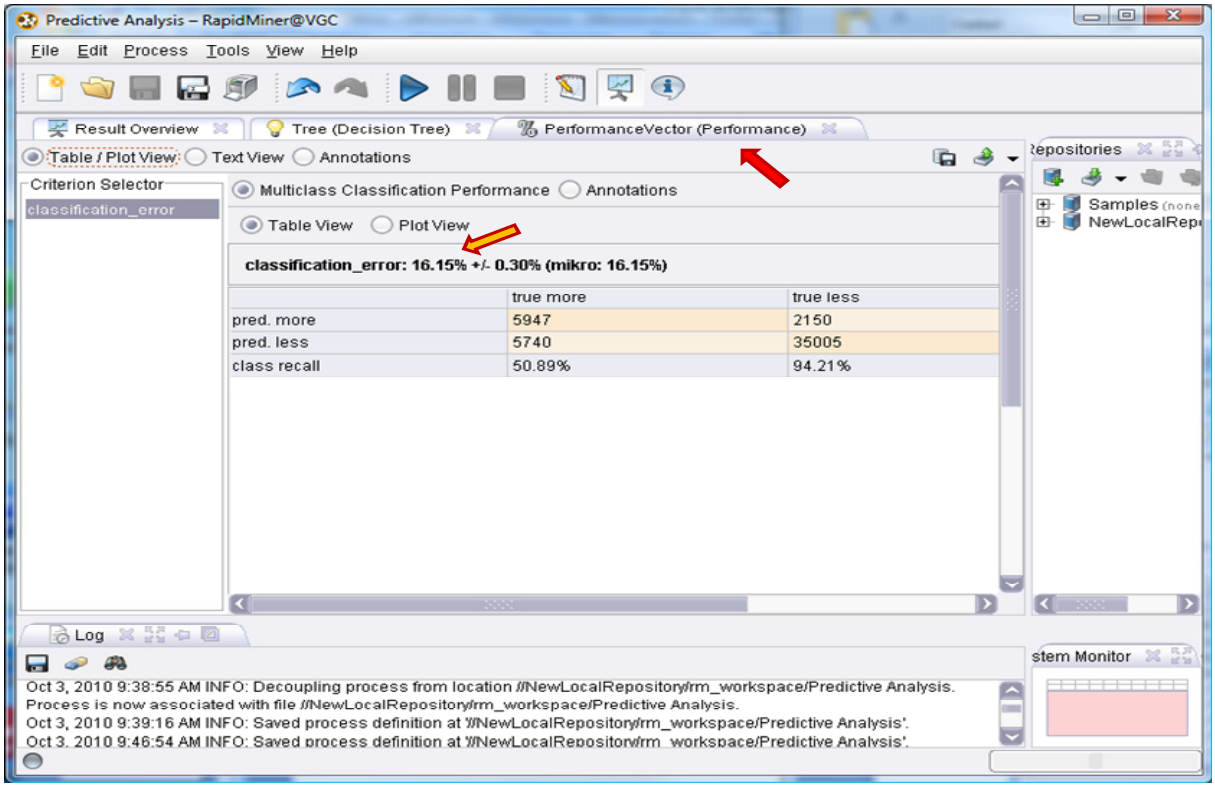
Multiple windows are generated when the calculations are completed. We can view the classification tree into the TREE tab. But the graphical representation is not really readable when we have a large tree.



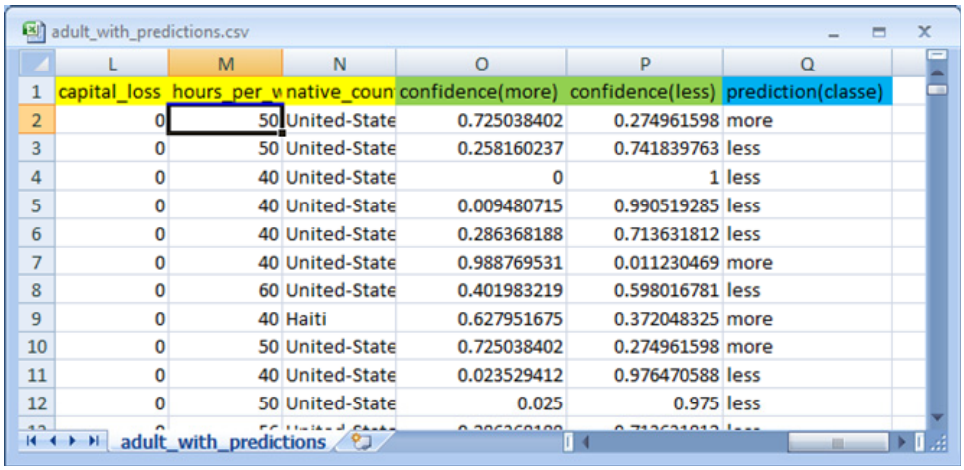
In some circumstances, the text representation is perhaps more convenient.



Into the "Performance Vector" tab, we can visualize the confusion matrix and the cross-validation error rate estimation. Thus, if we apply the tree on one instance from the population, the probability of misclassification is approximately 16.15%.



Last, about the deployment on unlabeled instances, we can visualize the generated data file (adult_with_predictions.csv) using a spreadsheet application. We observe at the same time: the prediction column; and the confidence (the estimated posterior probabilities) for each value of the class attribute. These values (posterior probabilities) are valuable to evaluate the reliability of the prediction.



4 Conclusion

We have briefly introduced the new RapidMiner 5.0 in this tutorial. The graphical user interface (GUI) presents significant changes compared to the previous one. It is in accordance with the standard commercial or freely downloadable tools (Orange, AlphaMiner, Knime, etc.). We note that

the behavior of RapidMiner, especially the management of the workspace and the specification of the processes are very similar to that of Knime. For the same analysis, we define the following diagram with Knime (Figure 3 – The same data mining process with Knime; see in comparison Figure 2).

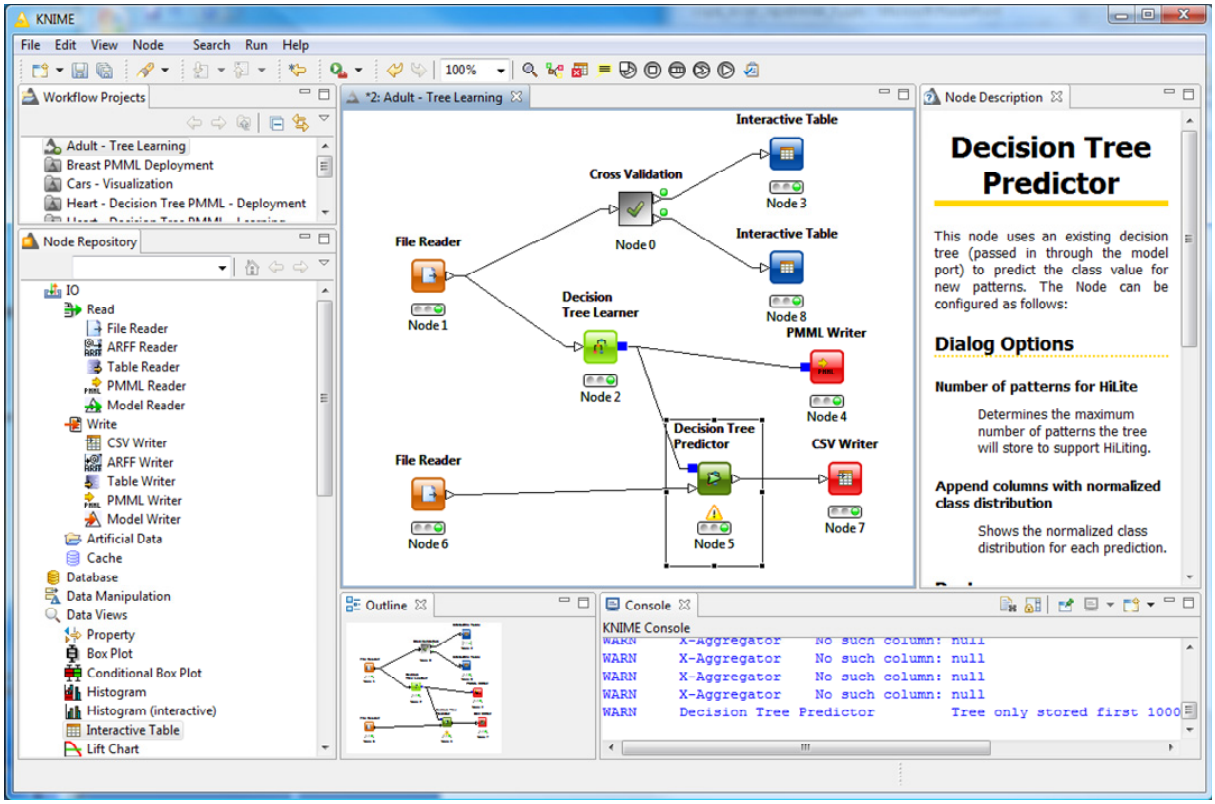


Figure 3 – The same data mining process with Knime

What is the better tool? I never answer in general to this kind of question. Because, these tools have not the same technical specifications, they respond to different constraints. Some of them are known, others are unknown. In my opinion, the good approach is to identify accurately the objective and the context of a study before to compare the behavior of the tools. That is what I try always to do on this website. The good news is that these tools are freely downloadable. We can try them, and make our own opinion.