1 Theme

Comparing the behavior of the PLS-LDA (Partial Least Squares – Linear Discriminant Analysis) with various well-known supervised learning algorithms.

PLS regression is a regression technique usually designed to predict the values taken by a group of Y variables (target variables, dependent variables) from a set of variables X (descriptors, independent variables) [Garson, http://www2.chass.ncsu.edu/garson/PA765/pls.htm]. Initially defined for the prediction of continuous target variable, the PLS regression can be adapted to the prediction of one discrete variable - i.e. adapted to the supervised learning framework - in different ways[1]. The approah is called "PLS Discriminant Analysis" in this context. It incorporates the valuable qualities that we know usually into this new framework: the ability to process a representation space with very high dimensionality, a large number of noisy and / or redundant descriptors.

This tutorial is the continuation of a precedent paper dedicated to the presentation of some variants of the PLS-DA[2]. We describe the behavior of one of them (PLS-LDA – PLS Linear Discriminant Analysis) on a learning set where the number of descriptors is moderately high (278 descriptors) in relation to the number of instances (232 instances). Even if the number of descriptors is not really very high, we note in our experiment a valuable characteristic of the PLS approach: we can control the variance of the classifier by adjusting the number of latent variables. To assess this idea, we compare the behavior of the PLS-LDA with state-of-the-art supervised learning methods such as K-nearest neighbors[3], SVM (Support Vector Machine from the LIBSVM library[4]), the Breiman's Random Forest approach[5], or the Fisher's Linear Discriminant Analysis[6].

2 Dataset

We use the ARRHYTMIA.BDM[7] data file. There are 420 instances. We use 232 for the learning phase, 188 for the testing phase. The STATUS column enables us to specify the subsamples. The target attribute ARRHYTHMIA is binary. It points out the presence or the absence of the disease. All the descriptors are continuous (or regarded as such). Some variables are composed of a single value. They are useless for the prediction. They must not disturb the learning process.

The comparison of the performances of classifiers is a repeated theme in the supervised learning framework. We can find some tutorials dedicated to this subject on our website e.g. http://data-mining-tutorials.blogspot.com/2008/11/classifier-comparison-using-predefined.html (using a predefined test set as here) ; http://data-mining-tutorials.blogspot.com/2008/11/classifier-comparison-cross-validation.html (using a cross-validation [resampling] scheme) ; http://data-mining-tutorials.blogspot.com/2008/11/roc-curve-for-classifier-comparison.html (using another criterion than the error rate).

[1] See **S. Chevallier, D. Bertrand, A. Kohler, P. Courcoux, « Application of PLS-DA in multivariate image analysis », in J. Chemometrics, 20 : 221-229, 2006.**

[2] http://data-mining-tutorials.blogspot.com/2008/11/pls-regression-for-classification-task.html

[3] http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

[4] http://data-mining-tutorials.blogspot.com/2008/11/svm-using-libsvm-library.html

[5] http://data-mining-tutorials.blogspot.com/2008/11/random-forest.html

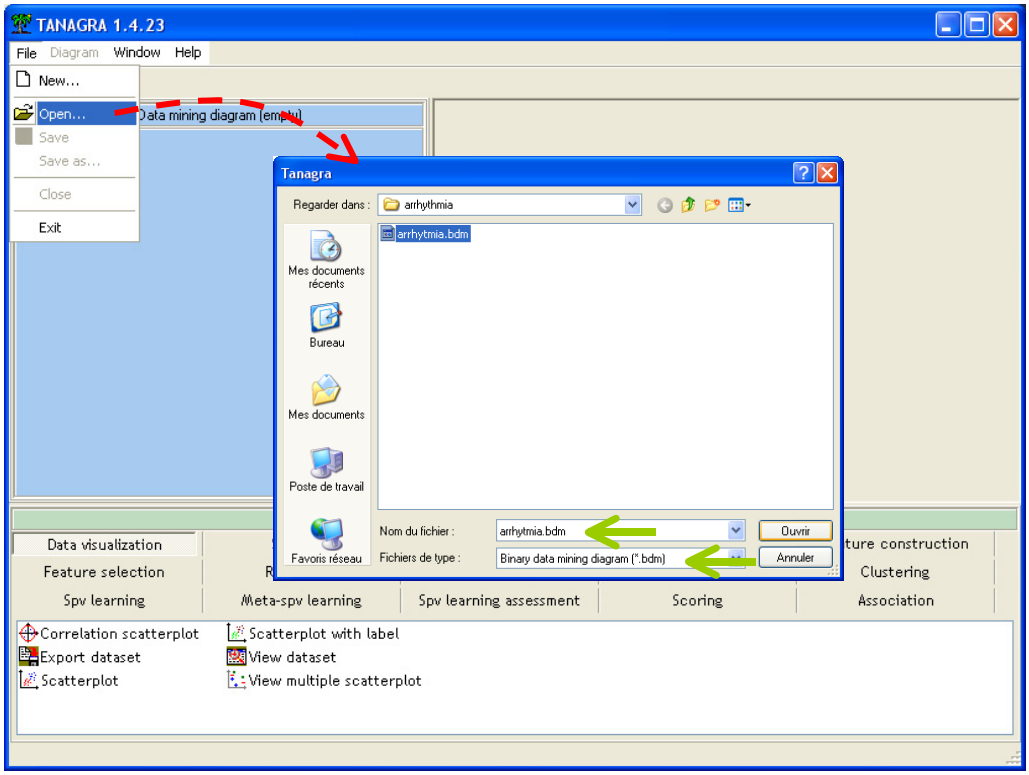
[6] http://en.wikipedia.org/wiki/Linear_discriminant_analysis

[7] http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/arrhytmia.bdm (TANAGRA binary format file).

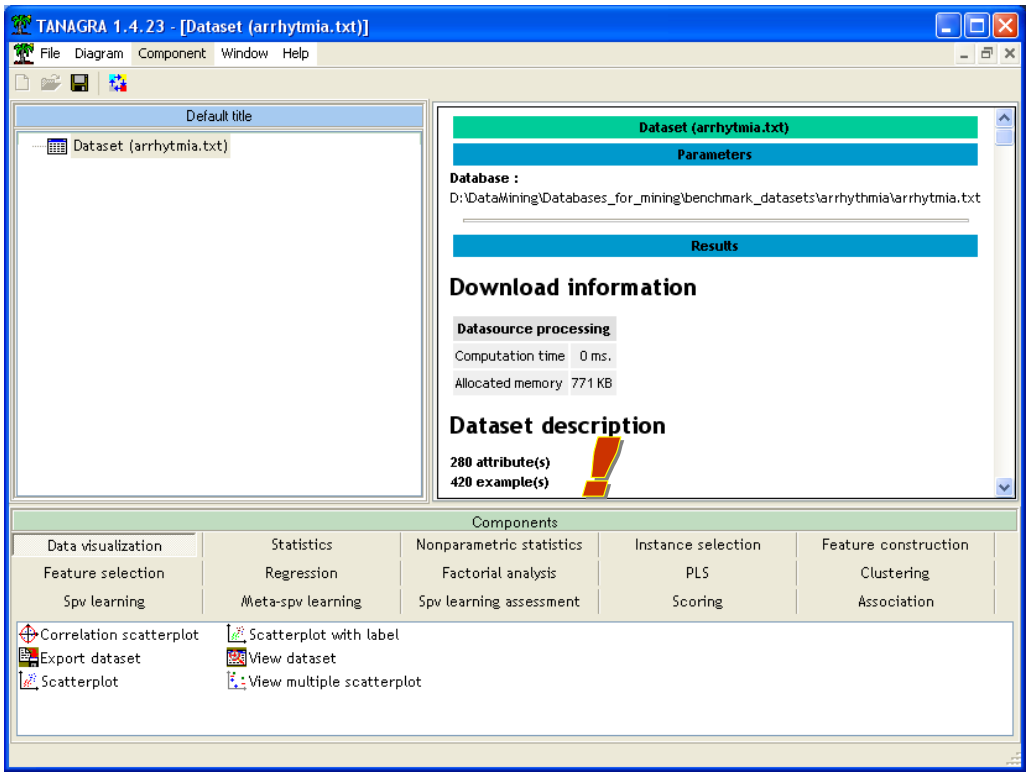
3 Comparison of supservised learning methods

3.1 Importing the dataset and preparing the processing

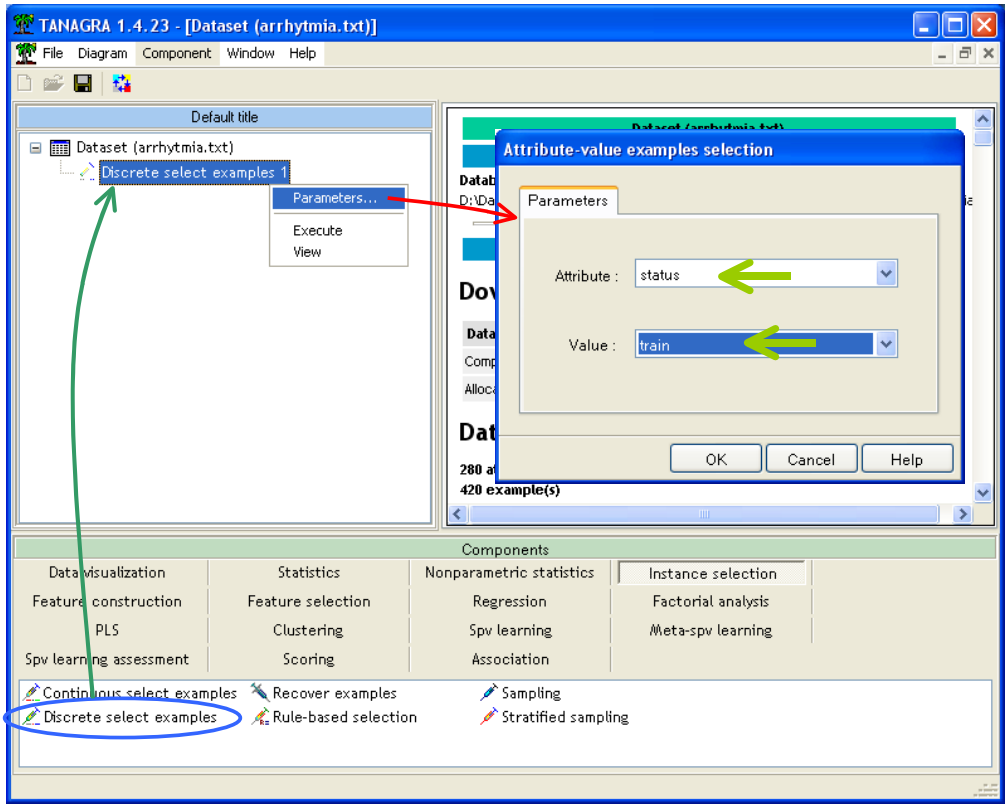
Importing the data file. We launch TANAGRA. We activate the FILE / OPEN menu. We select the ARRHYTMIA.BDM data file (Tanagra binary file format).



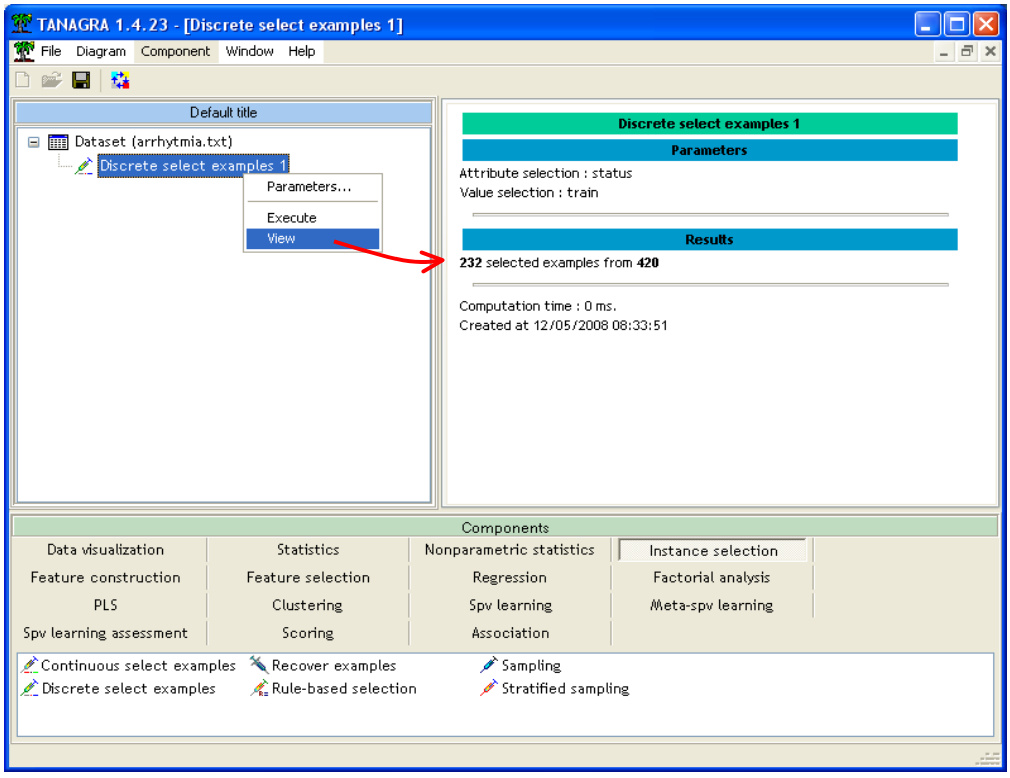
We have 420 instances and 280 attributes.



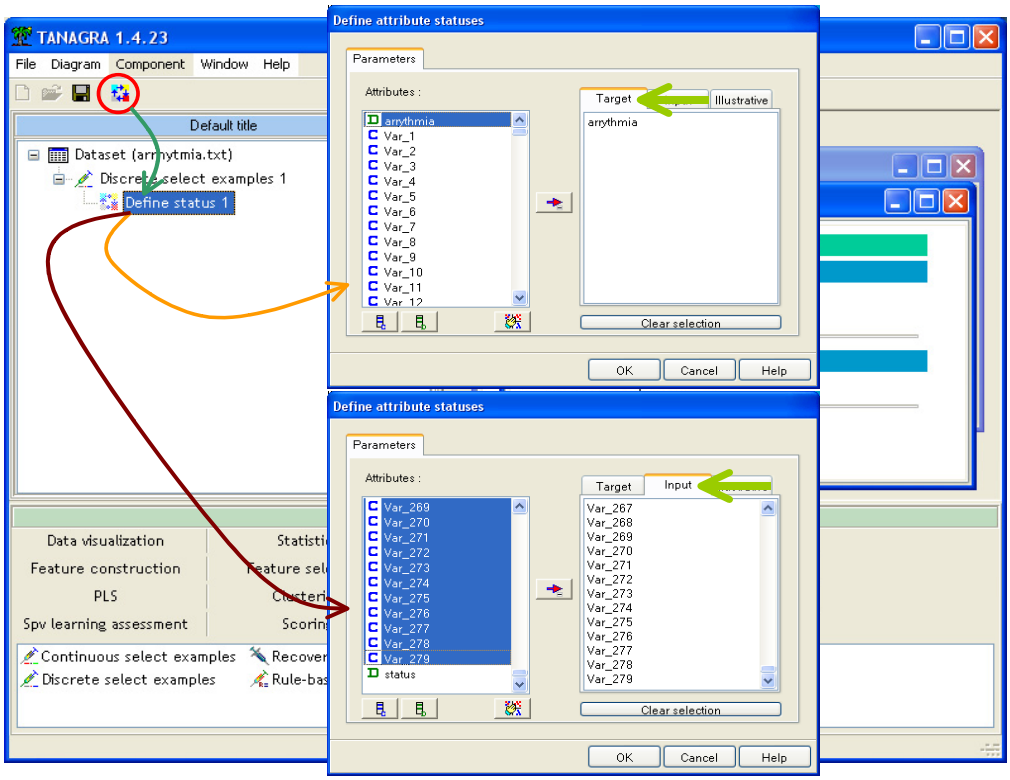
Partitioning the dataset into train and test sets. We use the STATUS column to define the train and test subsamples of our data. We insert the DISCRETE SELECT EXAMPLES (INSTANCE SELECTION tab) component into the diagram. We activate the PARAMETERS menu. In the setting box, we select STATUS as ATTRIBUTE and TRAIN as value. So, the instances labeled TRAIN are used as learning set in the subsequent part of the diagram.



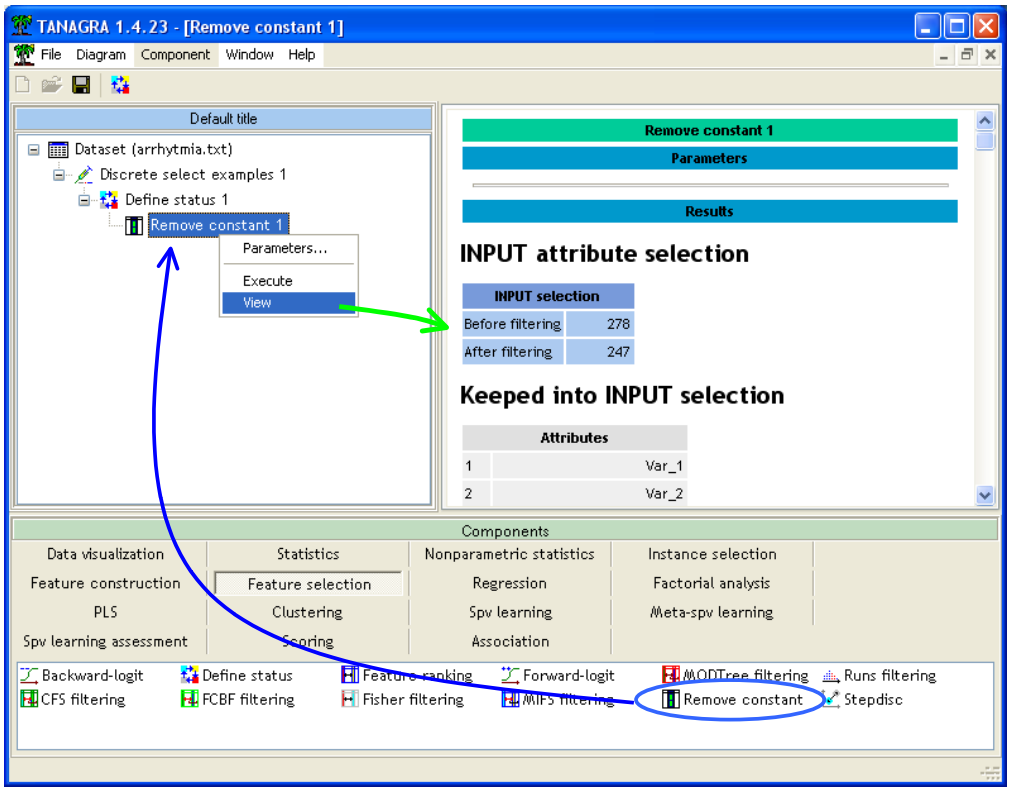
We confirm this choice by clicking on the OK button. We activate the contextual VIEW menu. We note that 232 instances now are assigned to the learning processes.



Specifying the target and the input variables. We add the DEFINE STATUS component (from the toolbar) to define the TARGET attribute (ARRYTHMIA) and the INPUT ones (VAR_1 to VAR_279).



Filtering the input attributes. Often, when we have a large number of descriptors, some of them are irrelevant for the prediction. This is the case more particularly of the constant variables i.e. the variables which have a single value. They must be eliminated from the selected predictors before starting the learning process. We insert the REMOVE CONSTANT (FEATURE SELECTION tab) into the diagram. We launch the filtering by clicking on the VIEW menu.

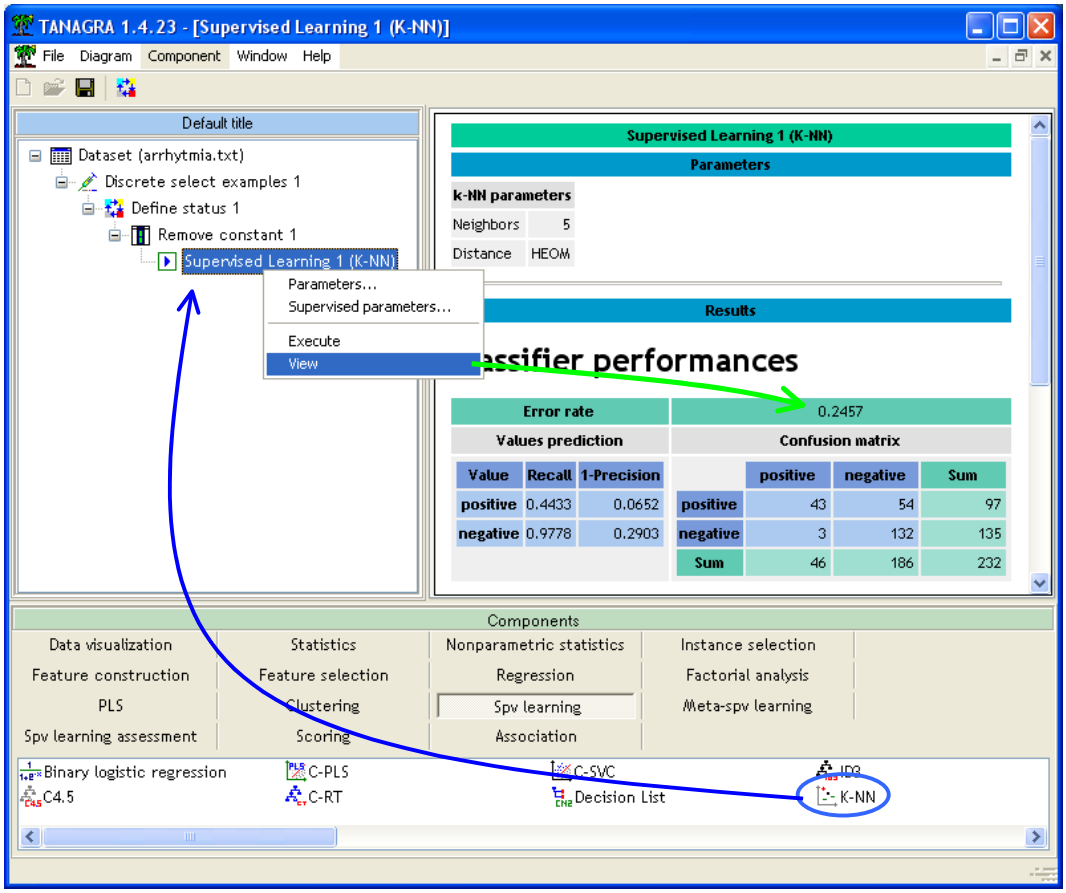


TANAGRA shows that 247 attributes are selected among the 278 descriptors. At the output of the component, the TARGET variable is not modified, but there are now 247 INPUT variables available for the modeling process.

3.2 Nearest neighbors classifier (K-NN)

The nearest neighbors classifier is certainly the worst approach than we can used in our context. Because the number of predictors is high compared with the number of instances, the local estimations of the conditional probabilities are not reliable. This method will be used as a reference. It will allow us to position the other techniques. The other approaches should be better.

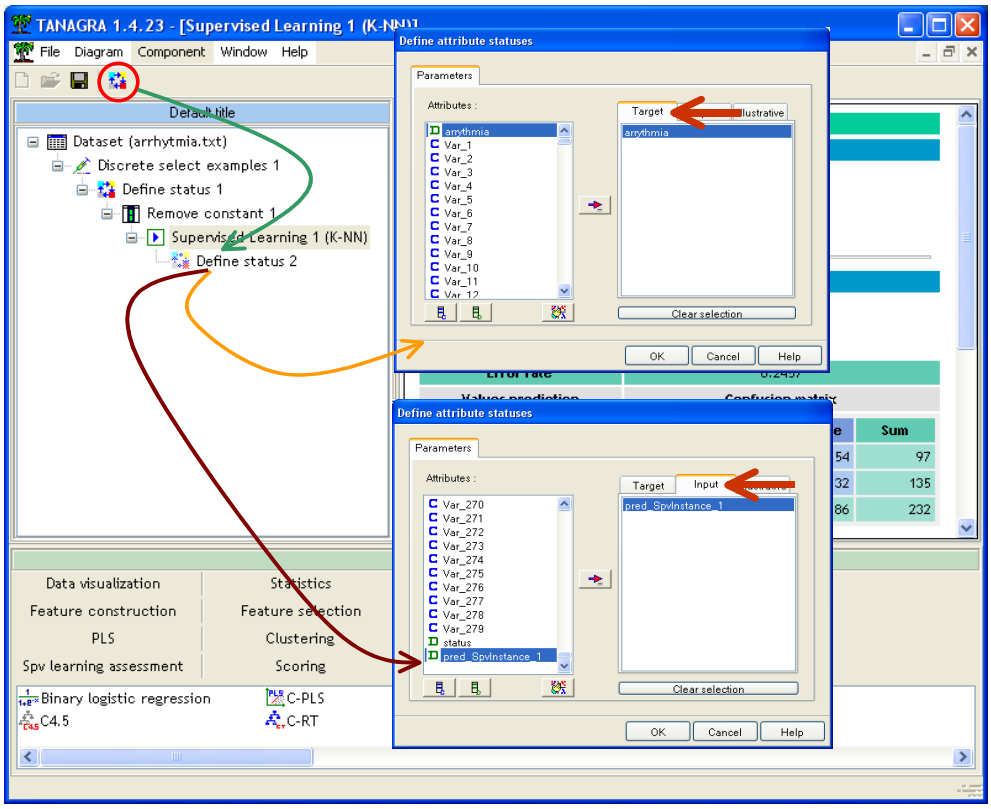
Learning phase. We insert the K-NN (SPV LEARNING tab) into the diagram. We click on the VIEW menu. Tanagra shows mainly the normalization parameters.



With a 5-NN (defaut parameter), the resubstitution error rate (computed on the learning set) is 24.57%. We know that this value is often optimistic, especially for the nearest neighbor classifier.

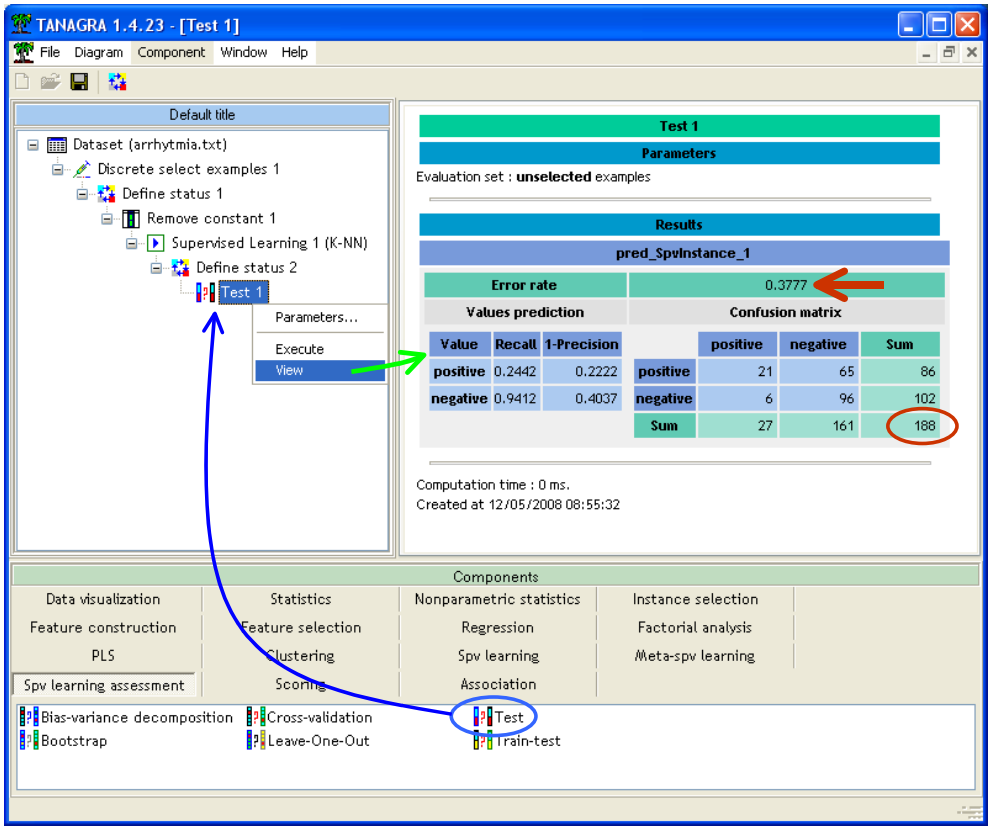
Testing phase. To obtain an honest estimation of the generalization error rate, we use the test sample. The only pitfall of this approach in our context is that the test sample size is not really enough to obtain a reliable estimation. But we know anyway that the error rate measured here will be always more interesting that the resubstitution error rate.

First, we insert the DEFINE STATUS component. We set the class attribute ARRYTHMIA as TARGET, the prediction PRED_SPVINSTANCE_1 of the classifier (generated automatically by Tanagra) as INPUT. This new column is generated for the instances used for the learning process, but it is generated also for the unused instances i.e. the instances belonging to the test sample.



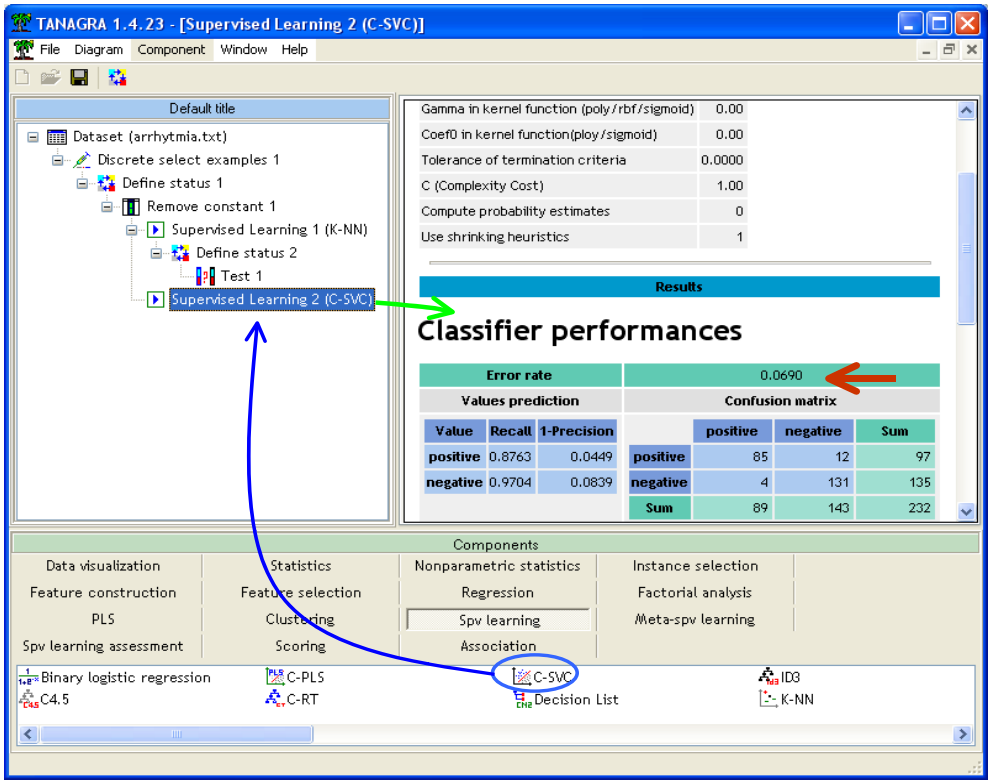
Second, we add the TEST component (SPV LEARNING ASSESSMENT tab) into the diagram. By default, it computes the error rate on the test set. The “true” error rate is 37.77%.

This error rate is not really exciting. But anyway, 5-NN is better than the default classifier which assignes the majority class systematically for individuals. The error rate of the default classifier is 86 / 188 = 45.74%.

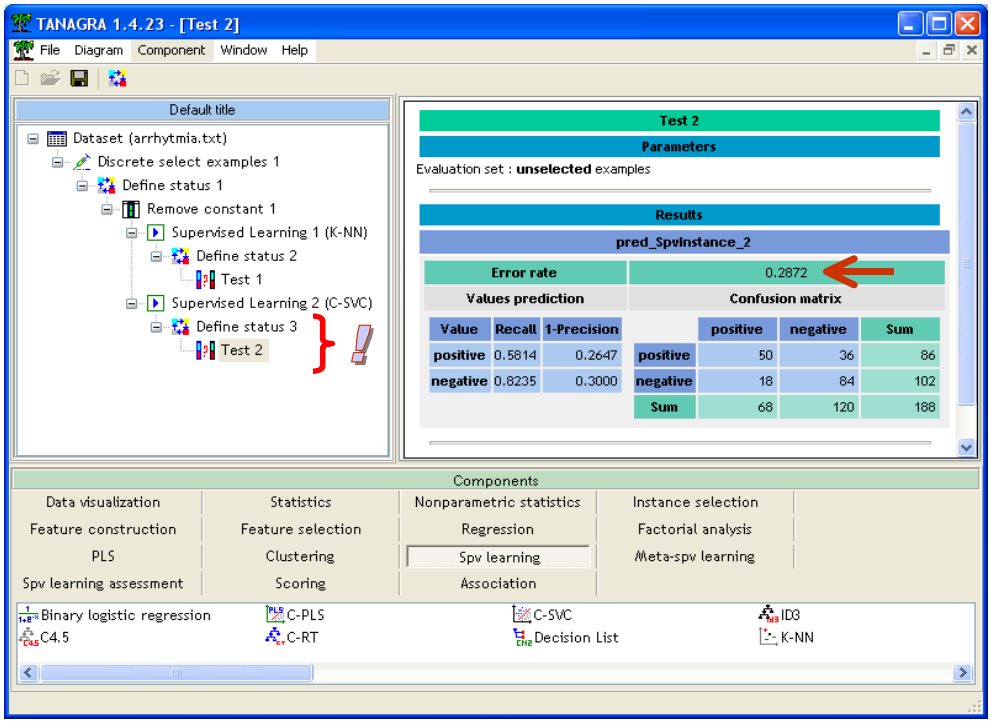


3.3 Support Vector Machine (SVM)

SVM – Linear kernel (C = 1.0). Let us try a linear SVM. It is a well regularized approach. It should not be disturbed by the high dimensionality of our dataset (compared with the number of instances). We repeat the train-test framework. We add the C-SVC (LIBSVM library) (SPV LEARNING tab) in our diagram. By default, it implements a linear kernel. The resubstitution error rate is 6.90%.



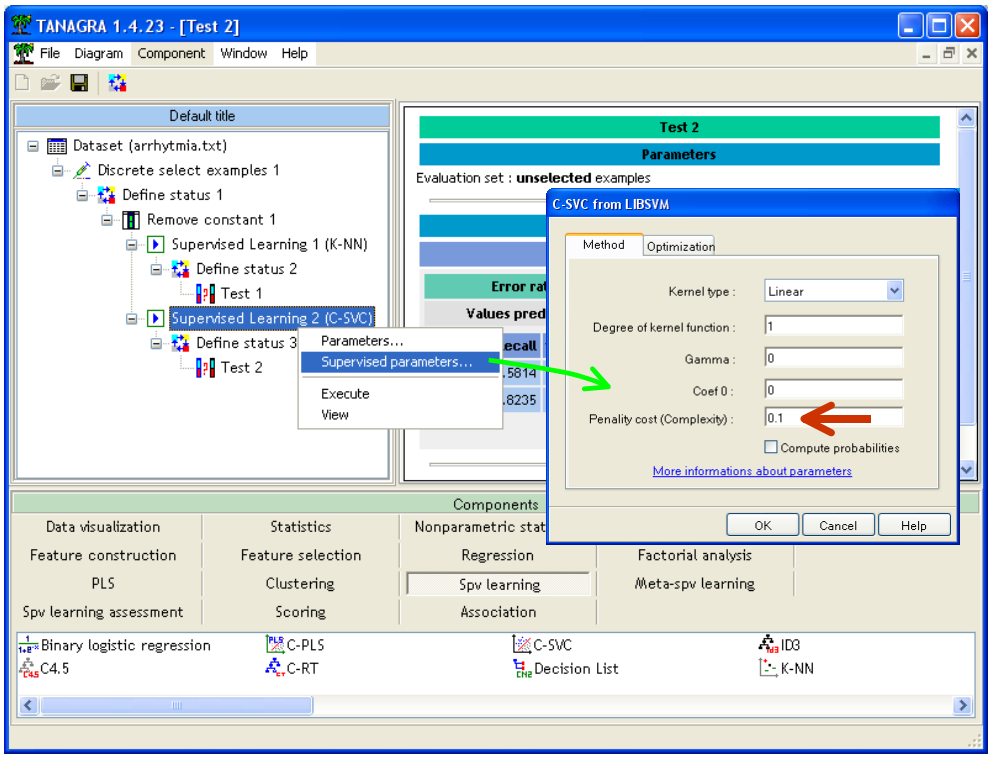
Once again, this value is not really interesting. We must evaluate the classifier on the test set (you can copy / paste a part of the diagram to repeat the sequence of operations – see http://data-mining-tutorials.blogspot.com/2009/06/copy-paste-feature-into-diagram.html).



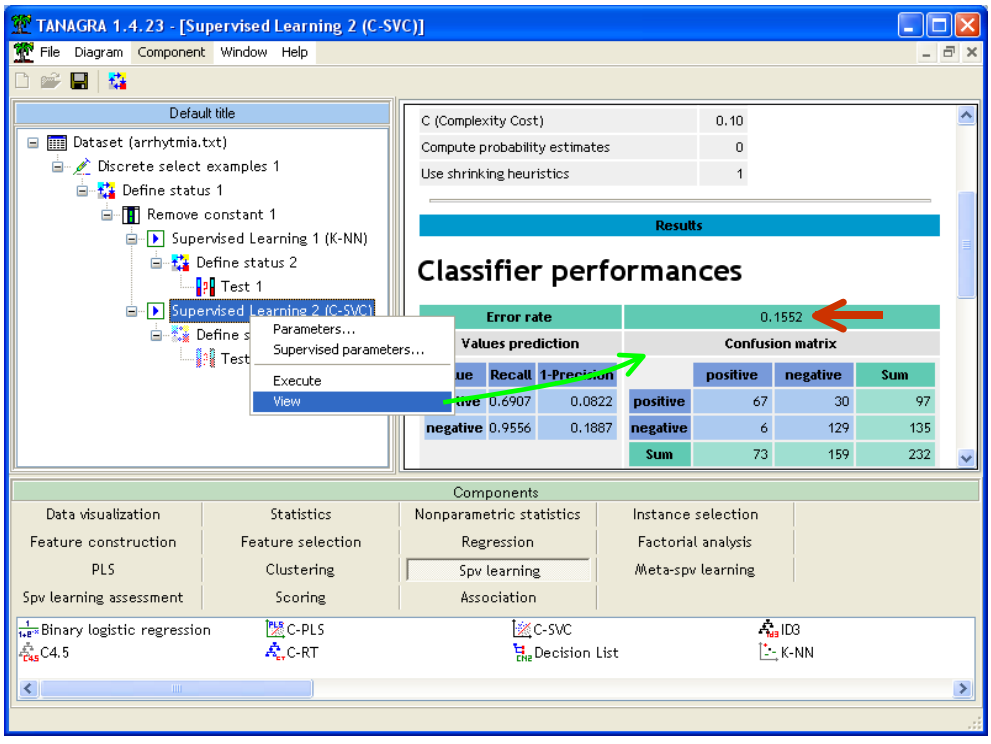
The test error rate is 28.72%. The improvement compared with the K-NN classifier is obvious.

Linear SVM with a strong regularization (C = 0.1). Most algorithms have parameters to guide the learning process. Often, they modify the regularization properties of the method i.e. its dependence on the training data. In the case of C-SVC, this is the purpose of the C parameter. We intensify the regularization property of the method when we decrease the value of C.

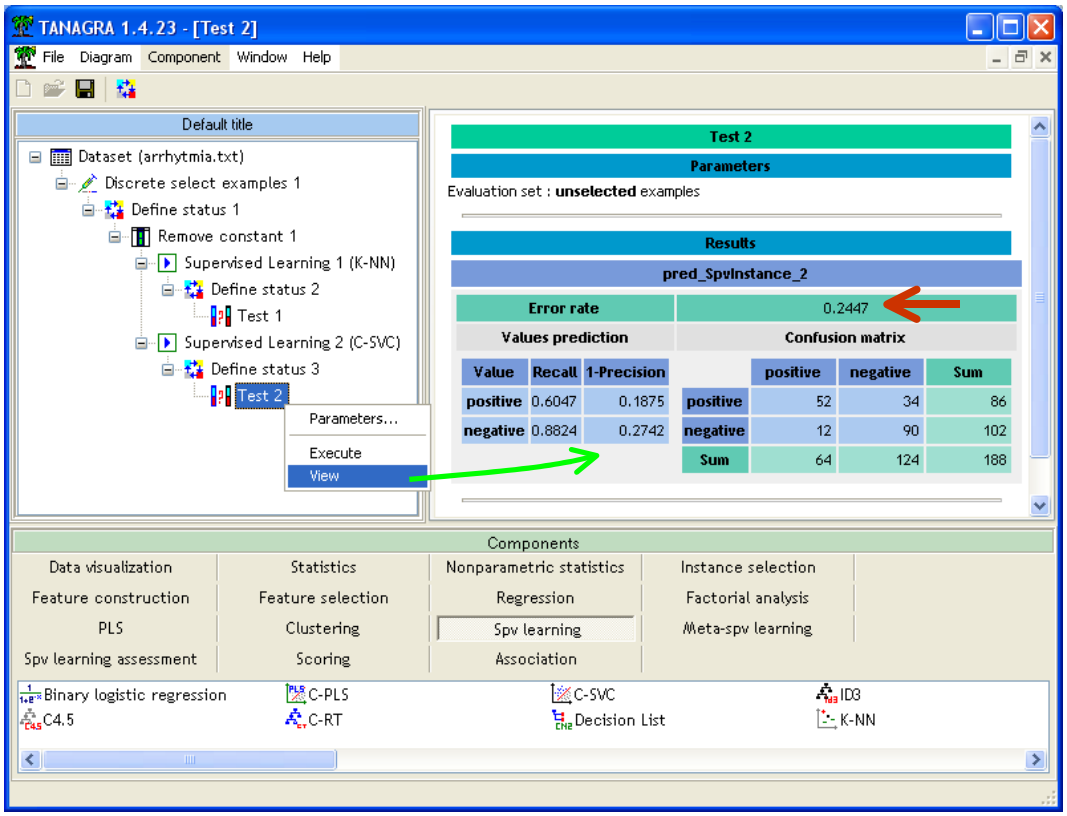
On our dataset, the gap between the learning and the test error rate seems to point out an overfitting. We try to set the new value of C to C = 0.1 (SUPERVISED PARAMETERS menu).



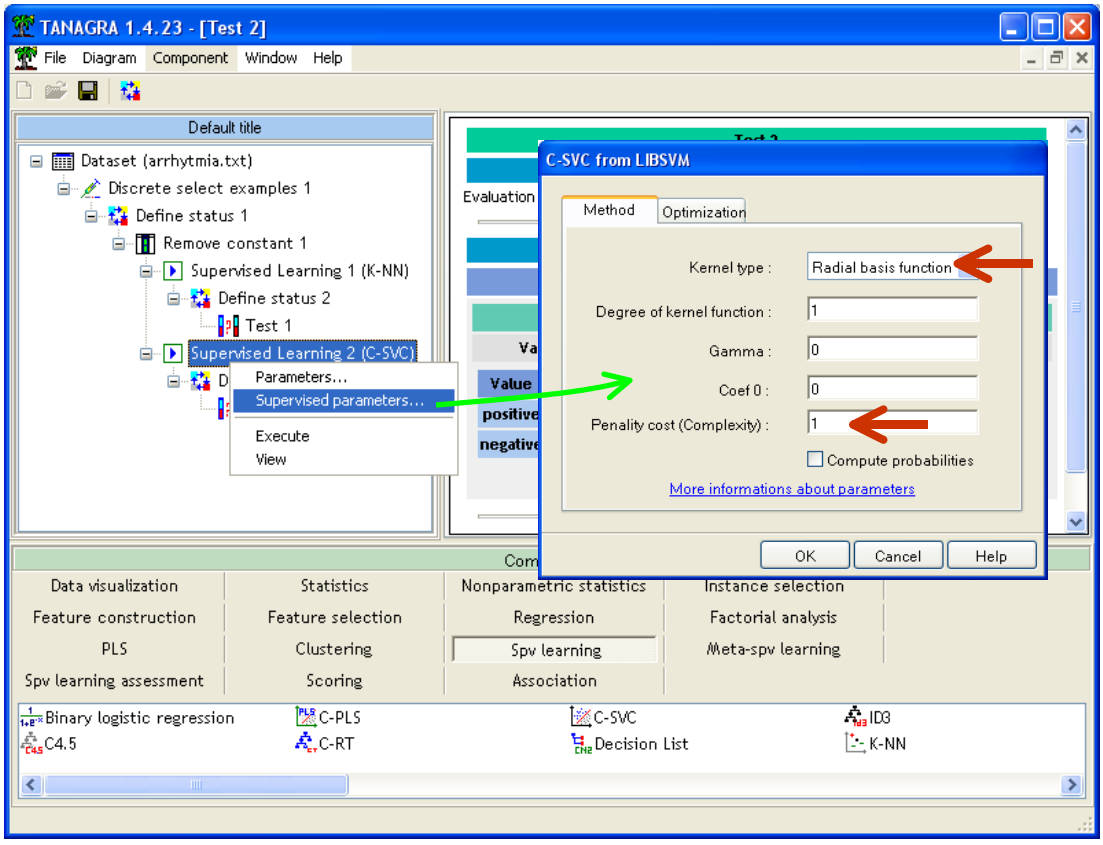
We observe that the resubstitution error rate is worse (15.52%).



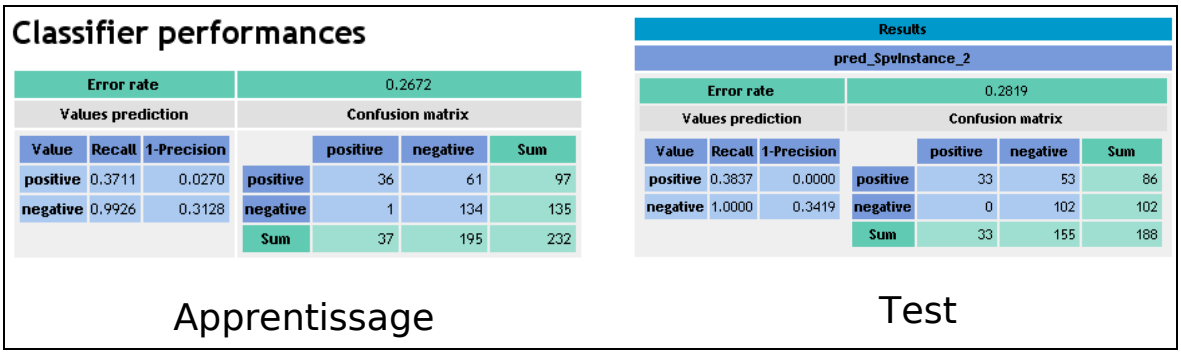
But when we click on the VIEW menu of the TEST component, we note that we improve the classifier capabilities: the test error rate is 24.47% now (against 28.72% when C was 1.0).



SVM-RBF (C = 1.0). What happens if we modify the kernel? We know that this parameter can heavily modify the behavior of the SVM. Here we try the RBF kernel. We set the C parameter to 1.



The resubstitution error rate is 26.72%, the test error rate becomes 28.19%.



Classifier performances

Error rate	0.2672
Values prediction	Confusion matrix

Value	Recall	1-Precision
positive	0.3711	0.0270
negative	0.9926	0.3128

	positive	negative	Sum
positive	36	61	97
negative	1	134	135
Sum	37	195	232

Apprentissage

Results

pred_SpvInstance_2

Error rate	0.2819
Values prediction	Confusion matrix

Value	Recall	1-Precision
positive	0.3837	0.0000
negative	1.0000	0.3419

	positive	negative	Sum
positive	33	53	86
negative	0	102	102
Sum	33	155	188

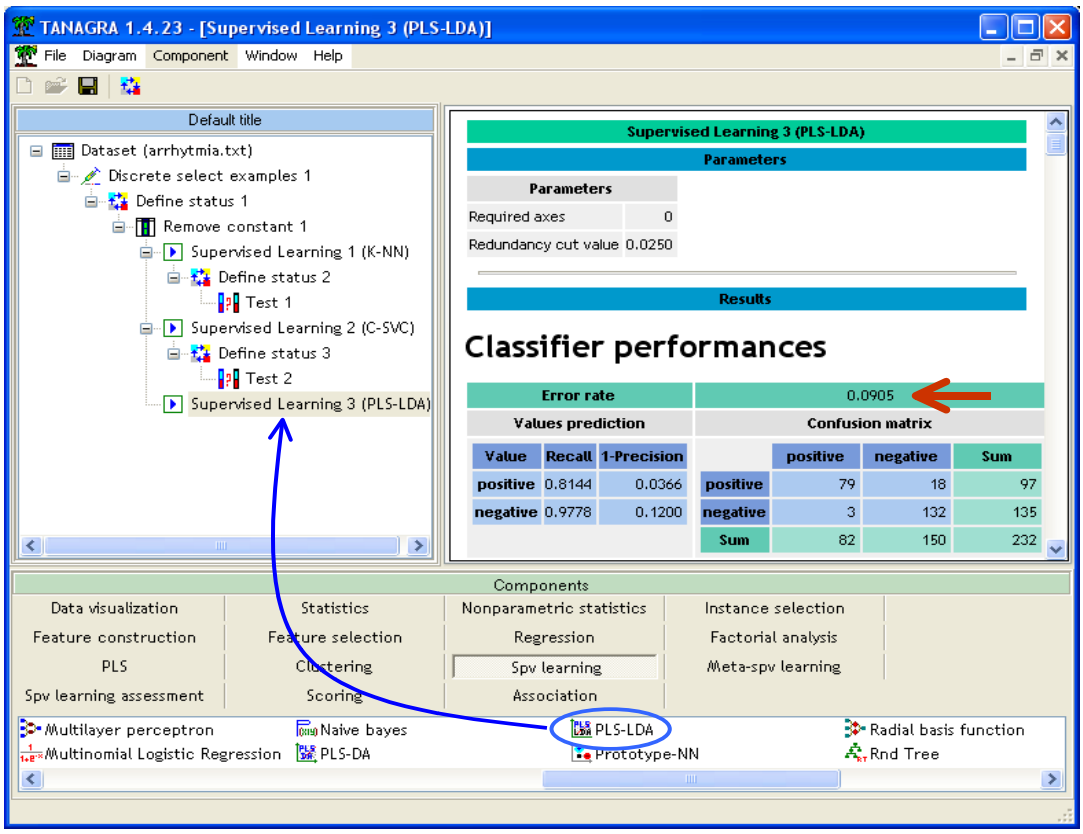
Test

3.4 PLS Regression – Linear Discriminant Analysis (PLS-LDA)

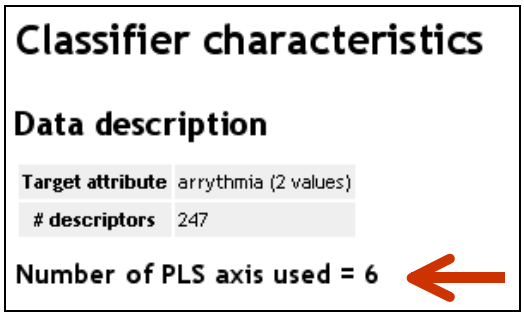
PLS-LDA (6 factors). The PLS-LDA approach was described in a previous tutorial (http://data-mining-tutorials.blogspot.com/2008/11/pls-regression-for-classification-task.html).

The modeling process is done in two steps : first, a PLS regression is performed on the dummy variables designed from the class attribute; then, a linear discriminant analysis is performed on the factors of the PLS regression. The regularization mechanism relies on the number of selected factors. Tanagra incorporates an automatic detection of the number of relevant factors based on the redundancy (proportion of explained variance) for the dummy variables associated to the classes.

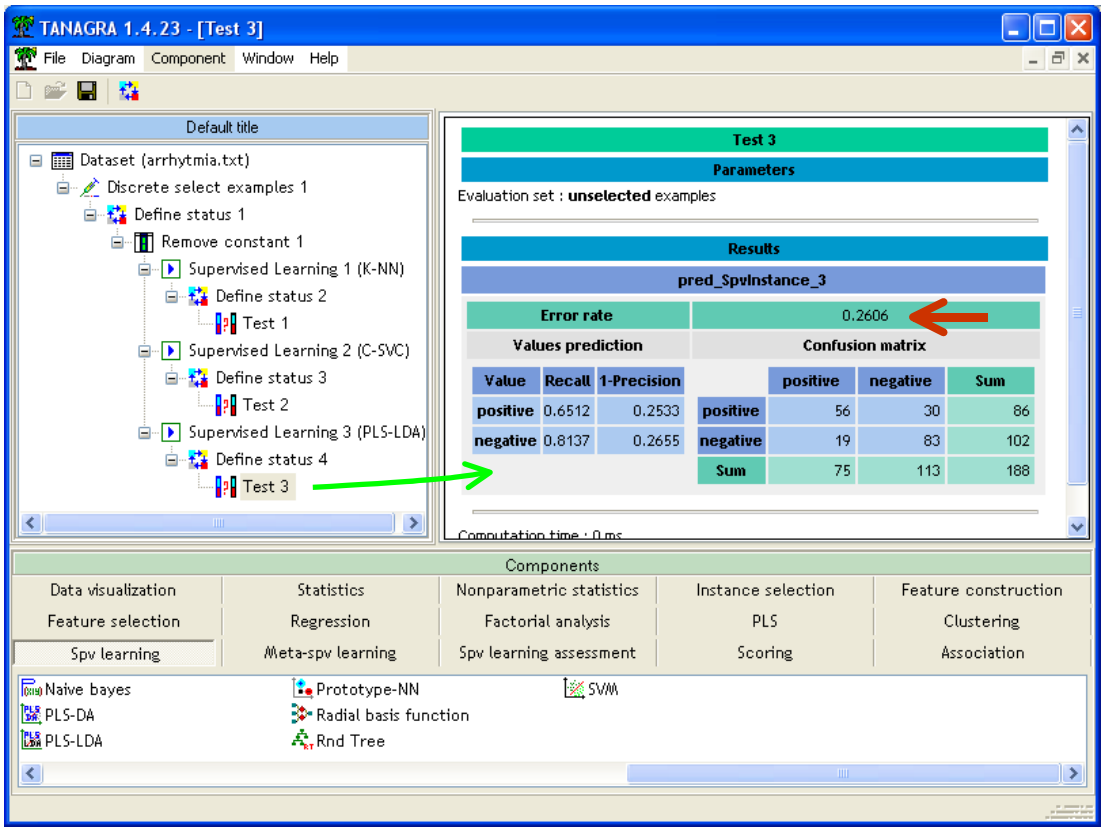
We add the PLS-LDA component into the diagram. We click on the VIEW menu. The resubstitution error rate is 9.05%.



The component detects automatically 6 relevant factors for the prediction.



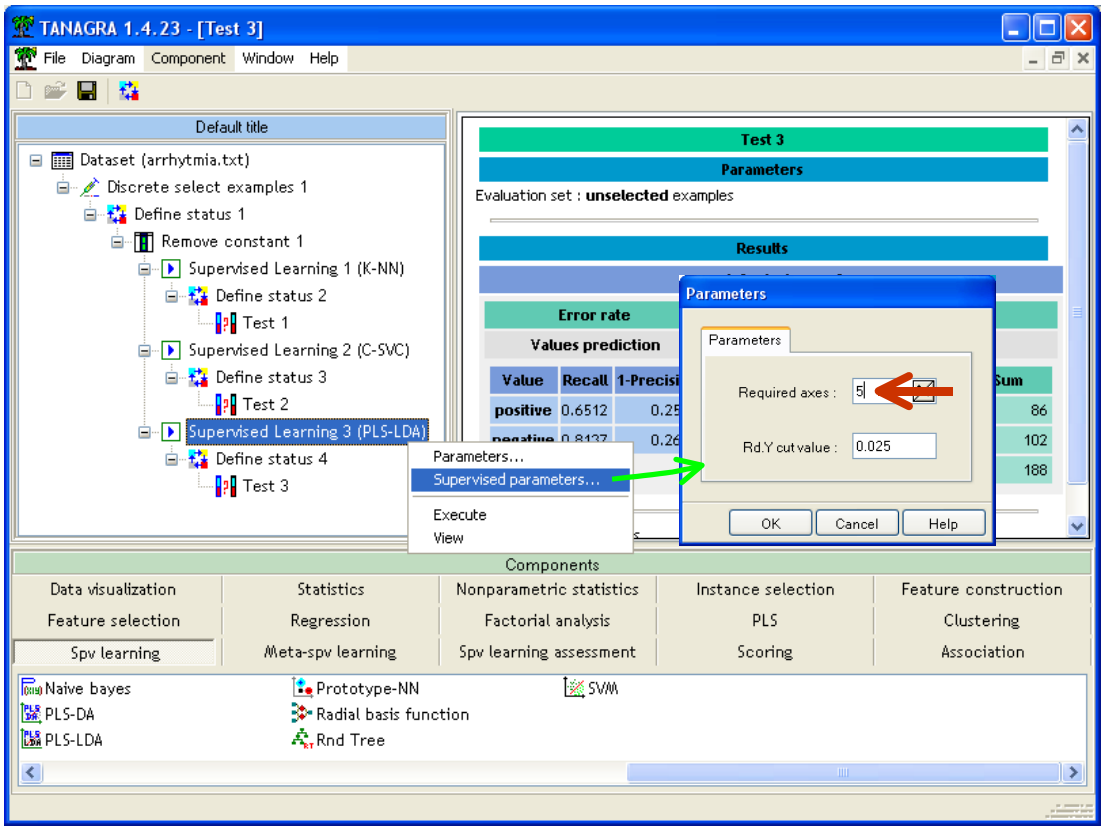
We fill out the diagram in order to obtain the test error rate. We have 26.06%.



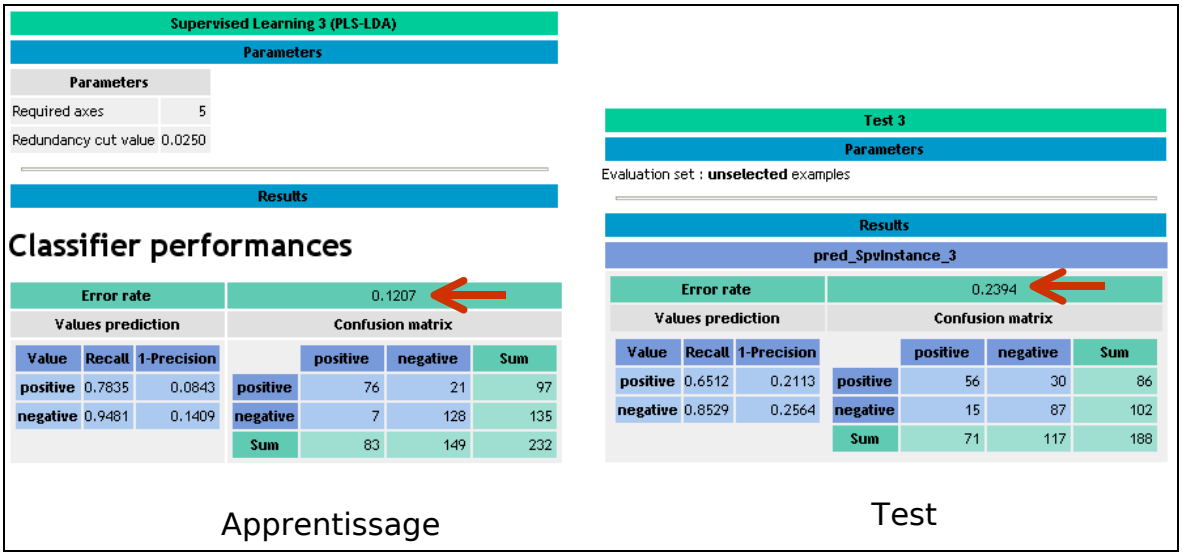
PLS-LDA with a strong regularization (5 factors). Here also, we can enhance the regularization property of the learning algorithm by modifying some parameters of the method.

We decrease the number of selected factors for the prediction. We set REQUIRED AXES = 5 (when REQUIRED AXES = 0, Tanagra tries to detect automatically the proper number of factors on the basis of the explained variance - the redundancy - of the class attribute).

The resubstitution error rate becomes 12.7%; and the test error rate 23.94%.



It's better than the various versions of SVM defined above[8].



By trial and error approach, we can tune the optimal number of factors. Note however that it is more appropriate to use a third dataset - says tuning set - for the optimization process. If we use the test set for this, the measured (test) error rate is distorded, it does not give either a reliable estimation of the "true" generalization error rate.

[8] Of course, the test file has few observations. We should not give too much importance to these small differences either.

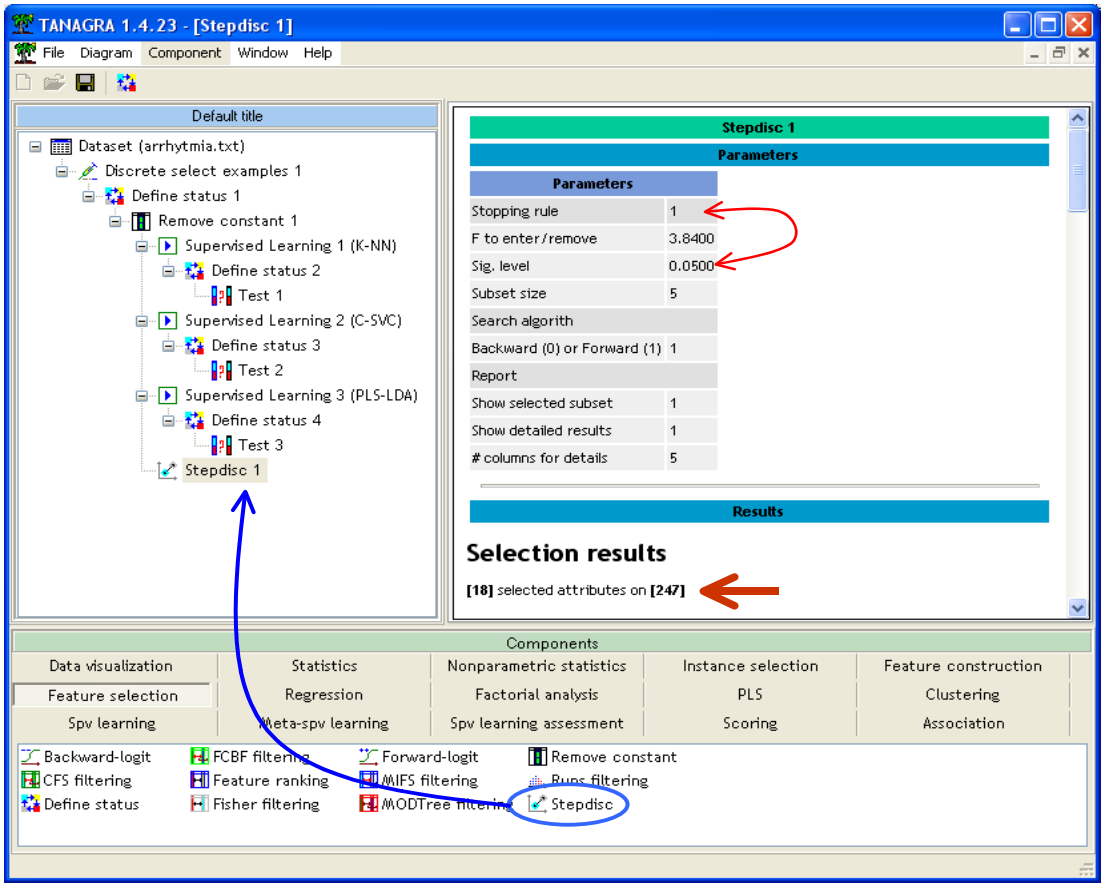
3.5 Linear discriminant analysis (LDA)

Linear discriminant analysis (18 predictors). Directly launch a discriminant analysis on this sort of problem is a sacrilege. Proceed with inversion of a matrix 247 x 247 is not a simple operation. In addition, it is likely that learning will be ineffective. The sample size (number of instances) is too small compared with dimensionality. The estimation of the variance co-variance matrix will be very unstable.

There are various approaches to regularize a linear discriminant analysis. In this tutorial, we use a variable selection process. This is a very simplistic way to decrease the over-dependance to the learning set. Yet, this is effective in many situations by reducing the variance of the classifier.

Compared to other techniques, the selection strategy is natural for the LDA. Indeed, the selection algorithm is consistent with the criterion used by the LDA to assess the group separability, namely the Wilks' Lambda (see MANOVA - http://en.wikipedia.org/wiki/Multivariate_analysis_of_variance).

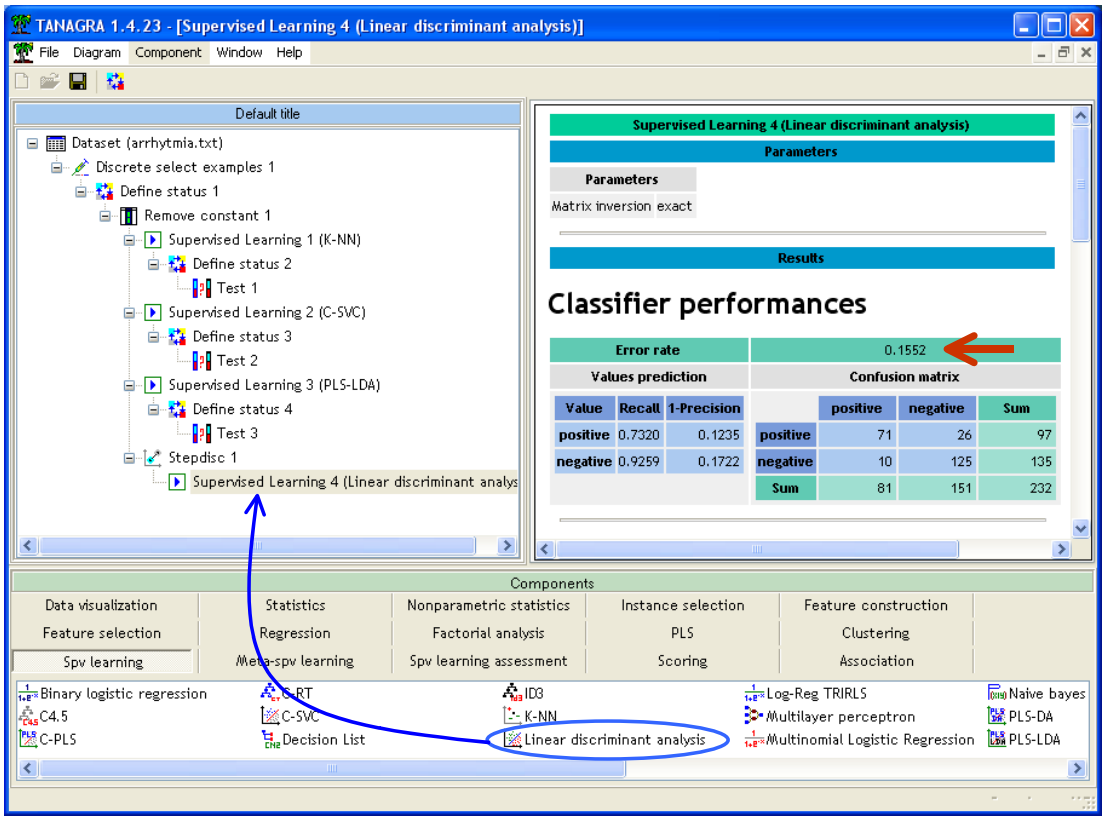
We insert the STEPDISC component (FEATURE SELECTION) into the diagram. We use the default settings. The component performs a FORWARD approach. The stopping rule is based on the comparison of the significance level[9] (0.05) specified by the user and the p-value[10] of the best predictor at each step. We click on the VIEW menu to obtain the results.



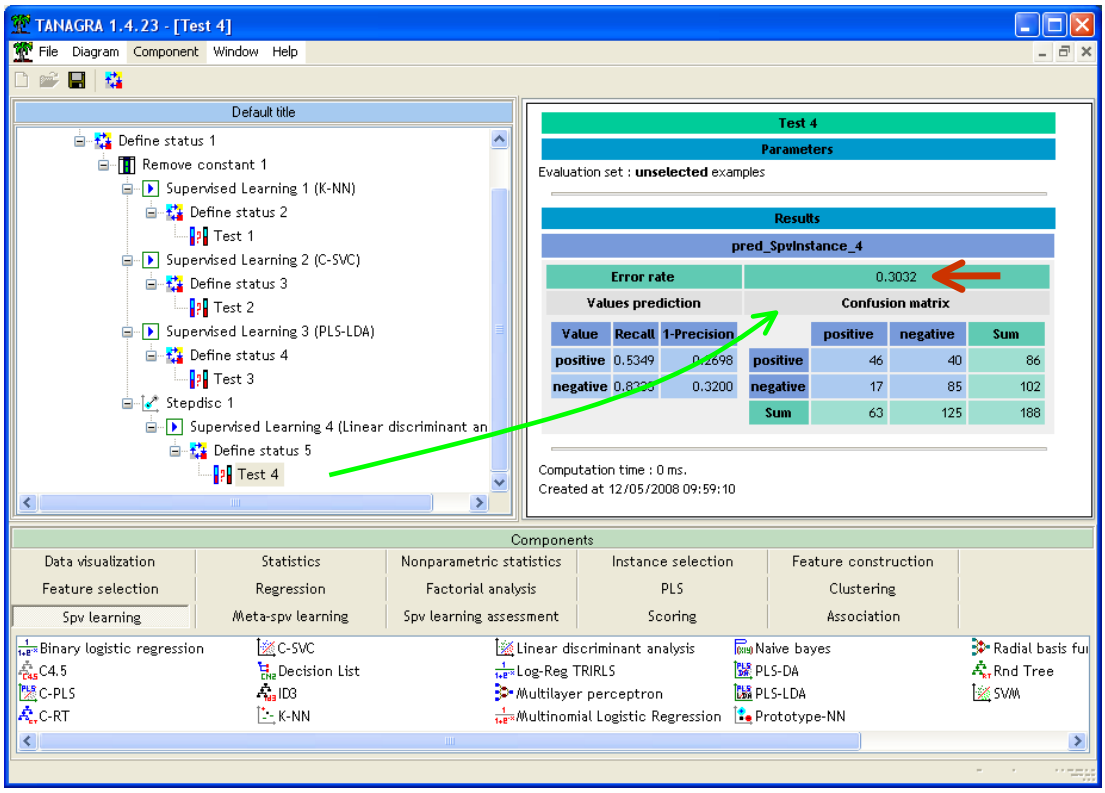
18 predictors are selected. We can add now the LINEAR DISCRIMNANT ANALYSIS (SPV LEARNING tab) to perform the construction of the classifier. The resubstitution error rate is 15.52%.

[9] http://en.wikipedia.org/wiki/Statistical_significance

[10] http://en.wikipedia.org/wiki/P-value

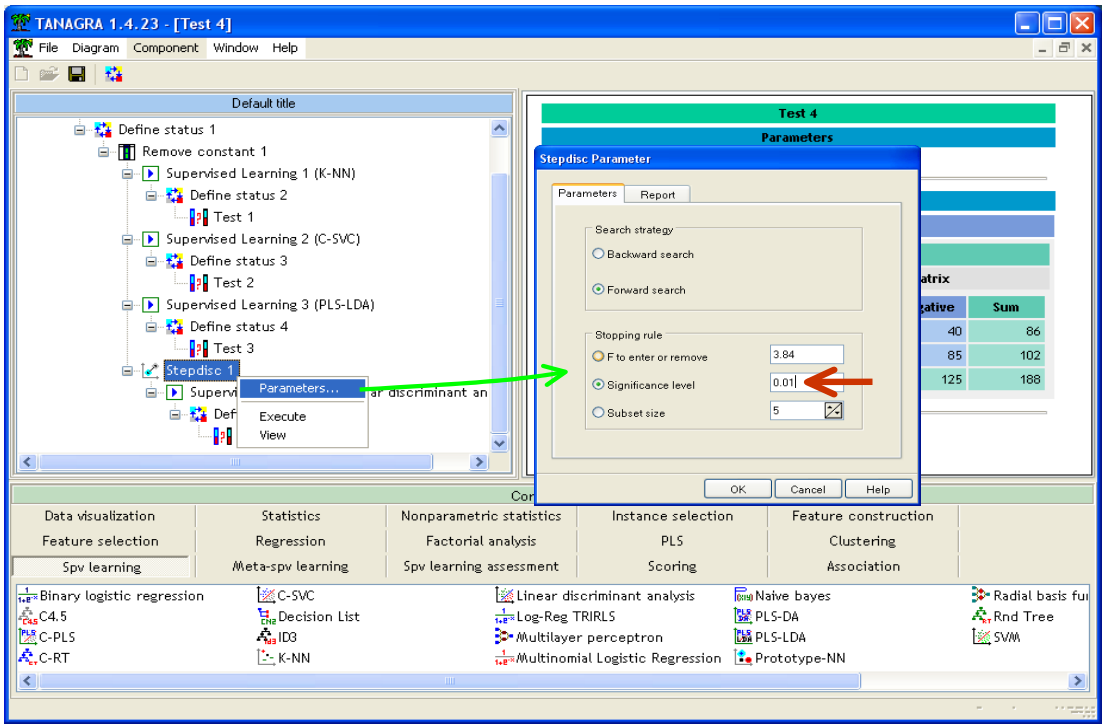


On the test set, the error rate becomes 30.32%. It is worse than all other methods discussed above. Clearly, there is an overfitting problem here. The number of selected variables is too high.

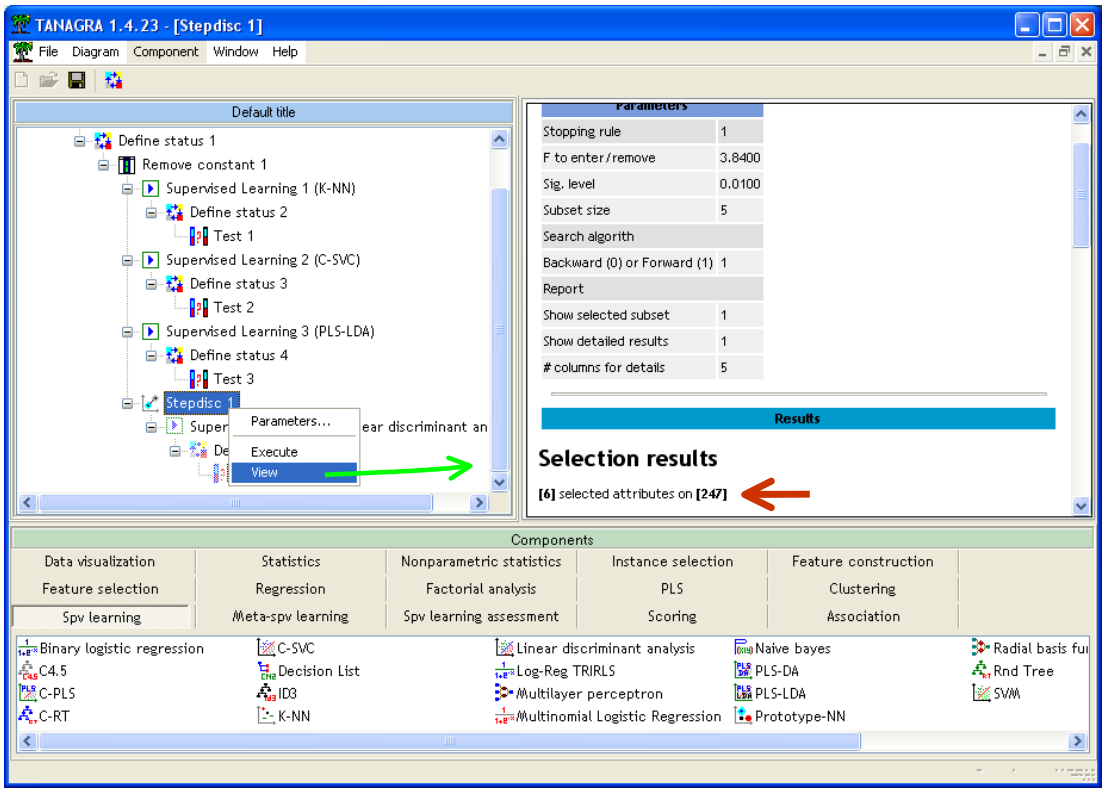


Linear discriminant analysis (6 predictors). We try to decrease the number of selected predictors. To do that, we modify the settings of the algorithm (PARAMETERS menu). We set the

significance level for selection to **0.01** (we are more demanding on the discriminatory power of the predictors to include into the model).

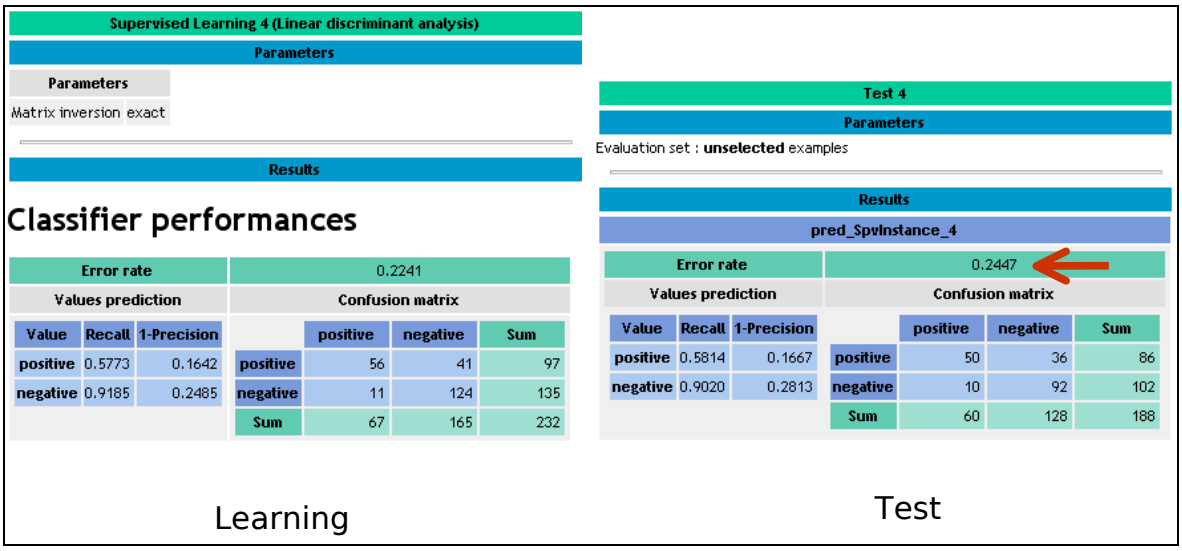


We click on the VIEW menu, 6 predictors are selected now.



About the classifier's performance, the resubstitution error rate is 22.41% and the test error rate 24.47%. This value is quite comparable to the results of SVM and PLS-LDA above. But the reduction in variance was obtained differently. Sometimes we wonder why we complicate our life with

sophisticated methods when a very simplistic approach, such as a variable selection in a linear discriminant analysis, can produce good results.

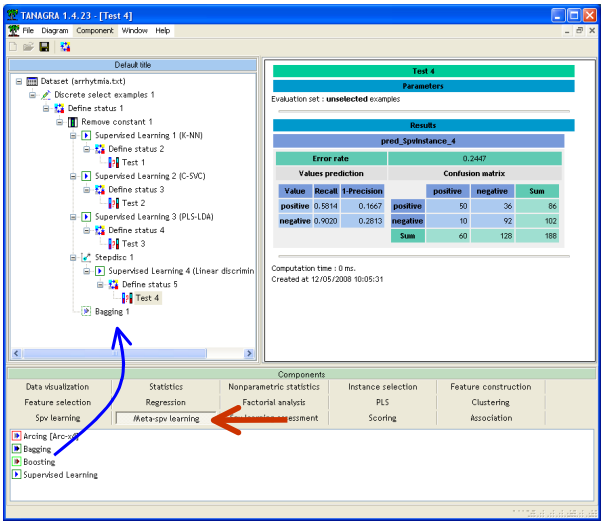


However, the approach is dependant to the number of selected descriptors. Specifying the right algorithm settings according the problem (dataset) characteristics is not always obvious.

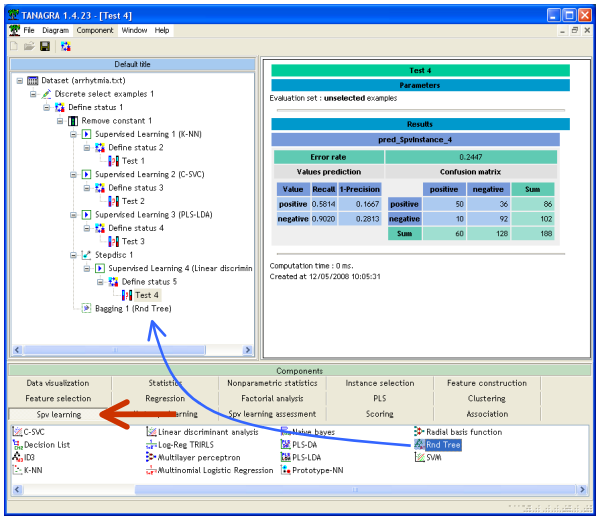
3.6 Random Forest

Random Forest. Random Forest (Breiman, 2001) is a very powerful approach for the prediction[11]. It designs a non-linear model.

To insert the random forest method into the diagram, we must proceed in two steps[12]: (1) adding the BAGGING component (META-SPV LEARNING tab) into the diagram; (2) embedding the RND TREE component (SPV LEARNING tab) into BAGGING.



(1)

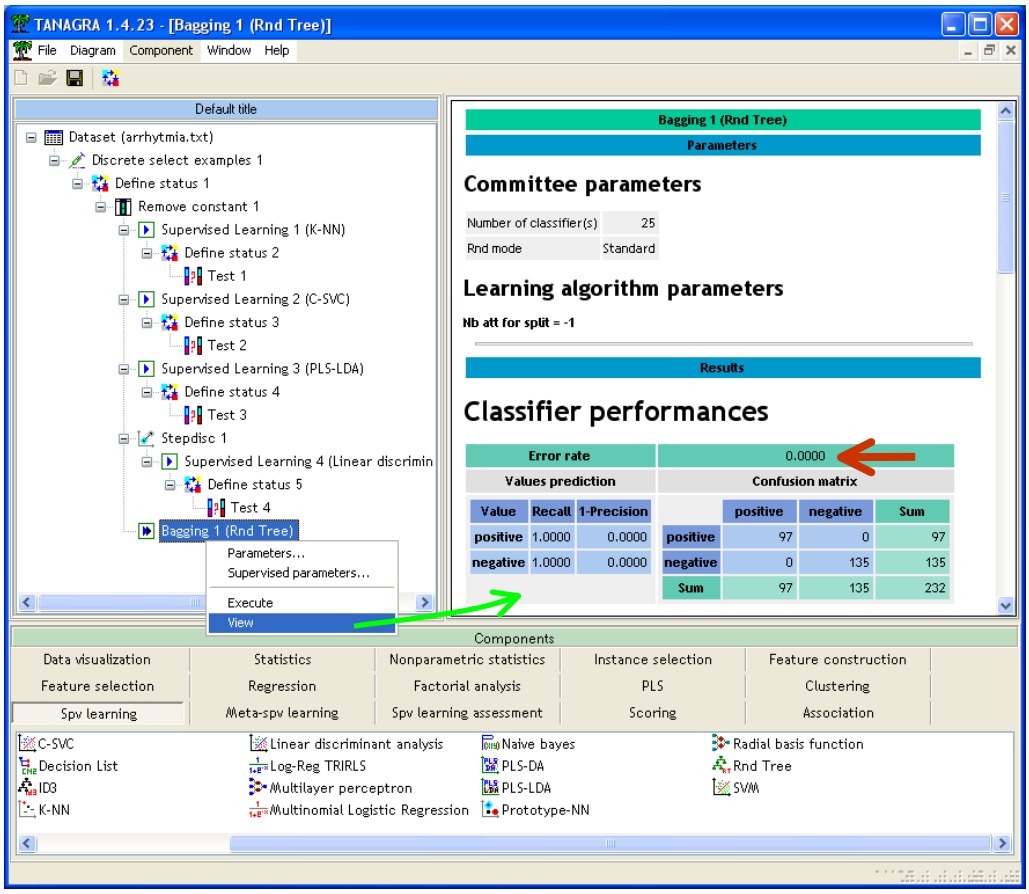


(2)

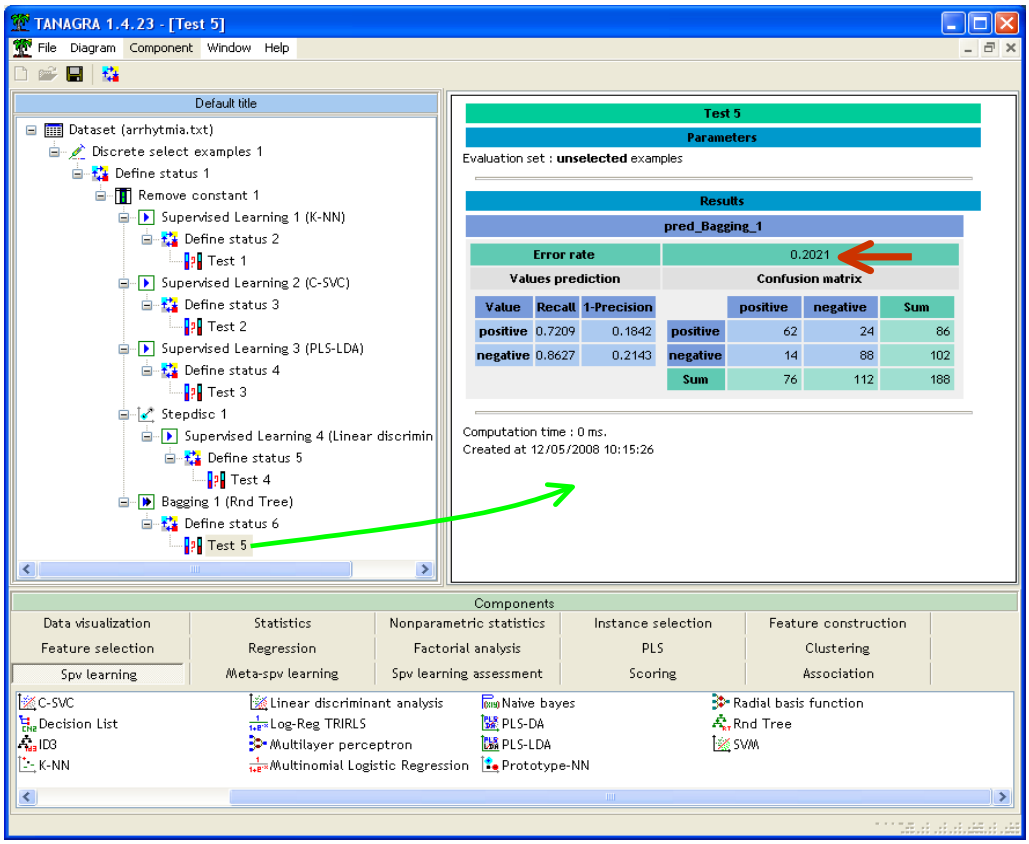
We click on the VIEW menu. The resubstitution error rate is 0%. It is usual for this approach.

[11] http://www.stat.berkeley.edu/~breiman/RandomForests/ ; http://en.wikipedia.org/wiki/Random_forest

[12] http://data-mining-tutorials.blogspot.com/2008/11/random-forest.html



Let us see the test error rate.



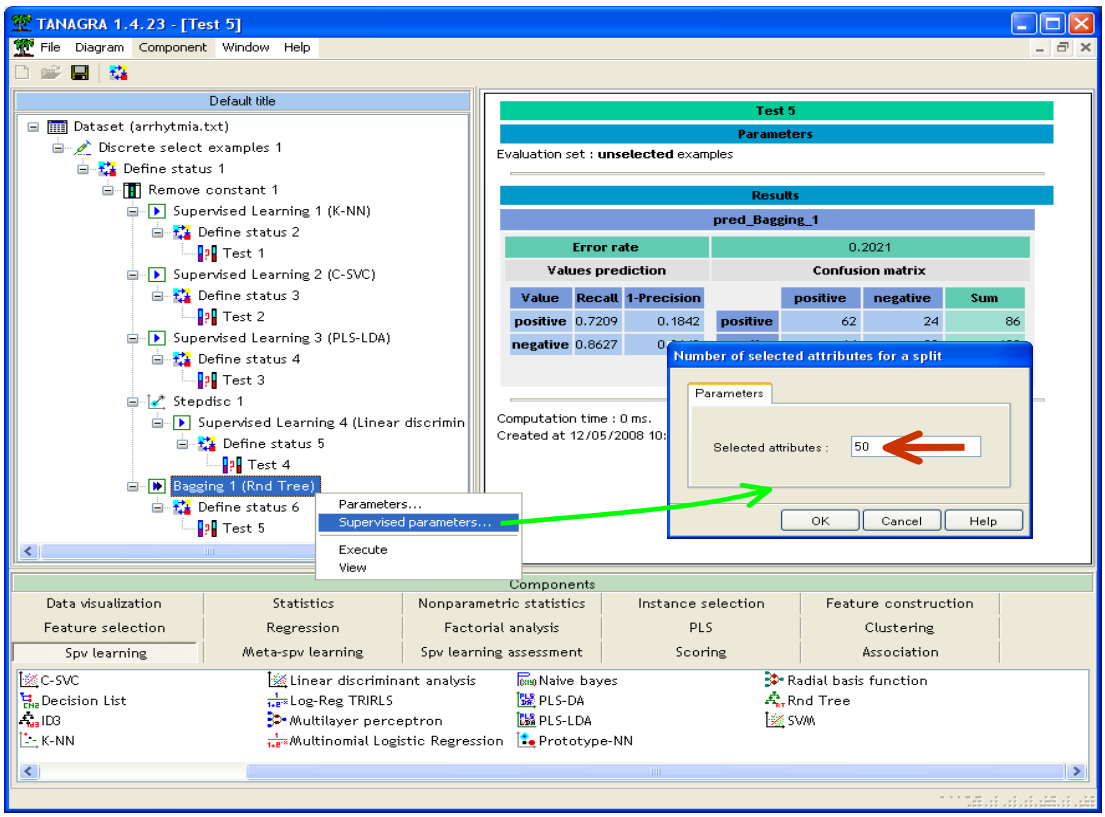
The test error rate is 20.21%. It is much better than any approaches discussed so far. Random Forest is more effective on our data. This is due to its non-linearity, but not only, the SVM with a RBF kernel and the K-NN had not given good results. Other qualities of SVM are beneficial here.

Paramétrer Random Forest (Split Variables = 50). We try to modify the regularization properties of the random forest. How do proceed here?

The only parameters we can manipulate are the number of variables selected for the splitting of each node and the number of trees. About this last one, the default value is 25 trees. We have tried to increase this value (50, 100...). No improvement was observed. This setting does not seem decisive for our dataset.

About the number of selected variables for the splitting of a node, if the use does not specify a value (SELECTED ATTRIBUTES = -1), Tanagra uses the default formula "P = ROUND(LOG2(J)) + 1", where J est the number of predictors into the dataset. On our dataset, J = 247, thus P = 9. How to set the parameter according to the data and the problem addressed is very difficult. The only practicable approach often is the trial and error. We change the value of P and we observe the consequences on the test sample. Here, we want to set P = 50.

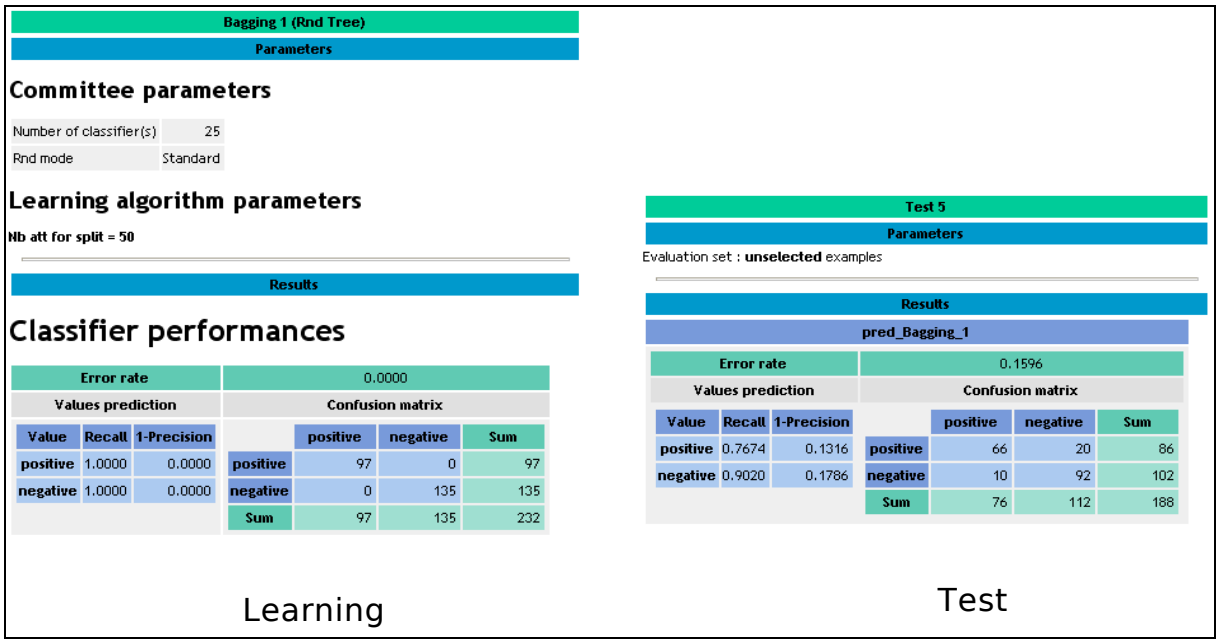
We click on the SUPERVISED PARAMETERS menu, we set SELECT ATTRIBUTES = 50.



We click on the VIEW menu. The resubstitution error rate is 0%.

But, on the test set, we note that we decrease again the error rate, it is 15.96%. This is the best results obtained on this dataset in this tutorial.

Perhaps, it is possible to obtain better results on this dataset. We invite the reader to try other learning scenarios. The global scheme is the same: we select a learning approach; we set the appropriate settings using the expert knowledge or by trial and error process; we measure the test error rate.



3.7 Summary of the main points

We summarize in a table the main results obtained in this tutorial. We set a “*” to group the methods according to the test error rate.

Method	Resubstitution error rate	Test error rate
K-NN (K = 5)	24.57%	**37.77%**
SVM Linear (C = 1.0)	6.90%	**28.72%**
SVM Linear (C = 0.1)	15.52%	**24.47%****
SVM – RBF (C = 1.0)	26.72%	**28.19%**
PLS-LDA (6 factors)	9.05%	**26.06%**
PLS-LDA (5 factors)	12.70%	**23.94%****
LDA (18 variables)	15.52%	**30.32%**
LDA (6 variables)	22.41%	**24.47%****
Random Forest (25 trees, Split = -1)	0.00%	**20.21%****
Random Forest (25 trees, Split = 50)	0.00%	**15.96%******

Clearly, "Random Forest" is the best approach on this dataset. The gap increases with an appropriate setting. It is followed by PLS-LDA and Linear SVM.

Whatever the method, their performances are heavily influenced by the parameters.