# 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, <u>http://www2.chass.ncsu.edu/garson/PA765/pls.htm</u>]. 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<sup>1</sup>. 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<sup>2</sup>. 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<sup>3</sup>, SVM (Support Vector Machine from the LIBSVM library<sup>4</sup>), the Breiman's Random Forest approach<sup>5</sup>, or the Fisher's Linear Discriminant Analysis<sup>6</sup>.

# 2 Dataset

We use the ARRHYTMIA.BDM<sup>7</sup> 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. <u>http://data-mining-tutorials.blogspot.com/2008/11/classifier-comparison-using-predefined.html</u> (using a predefined test set as here) ; <u>http://data-mining-tutorials.blogspot.com/2008/11/classifier-comparison-cross-validation.html</u> (using a cross-validation [resampling] scheme) ; <u>http://data-mining-tutorials.blogspot.com/2008/11/classifier-comparison-cross-validation.html</u> (using a cross-validation [resampling] scheme) ; <u>http://data-mining-tutorials.blogspot.com/2008/11/classifier-comparison.html</u> (using another criterion than the error rate).

<sup>&</sup>lt;sup>1</sup> See S. Chevallier, D. Bertrand, A. Kohler, P. Courcoux, « Application of PLS-DA in multivariate image analysis », in J. Chemometrics, 20 : 221-229, 2006.

<sup>&</sup>lt;sup>2</sup> <u>http://data-mining-tutorials.blogspot.com/2008/11/pls-regression-for-classification-task.html</u>

<sup>&</sup>lt;sup>3</sup> <u>http://en.wikipedia.org/wiki/K-nearest\_neighbor\_algorithm</u>

<sup>&</sup>lt;sup>4</sup> <u>http://data-mining-tutorials.blogspot.com/2008/11/svm-using-libsvm-library.html</u>

<sup>&</sup>lt;sup>5</sup> http://data-mining-tutorials.blogspot.com/2008/11/random-forest.html

<sup>&</sup>lt;sup>6</sup> <u>http://en.wikipedia.org/wiki/Linear\_discriminant\_analysis</u>

<sup>&</sup>lt;sup>7</sup> <u>http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/arrhytmia.bdm</u> (TANAGRA binary format file).

# 3 Comparison of supservised learning methods

## 3.1 Importing the dataset and preparing the processing

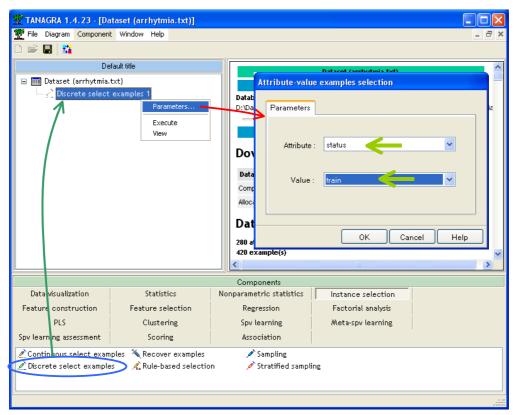
**Importing the data file.** We launch TANAGRA. We activate the FILE / OPEN menu. We select the ARRHYTMIA.BDM data file (Tanagra binary file format).

💯 TANAGRA 1.4.23	
File Diagram Window Help	
D New	
🛱 Open 🗾 Data mining diagram (emetu)	
Save N	
Save as	
Close Regarder dans : 😭 arthythmia V 🔇 🌶 🕫 🕮 •	
Exit Barrhytmis.bdm	
Mes documents récents	
a	
Bureau	
Mes documents	
Poste de travail	
Data visualization	ture construction
Data visualization Feature selection R Annuler	
	Clustering
Spv learning Meta-spv learning Spv learning assessment Scoring	Association
Correlation scatterplot Mc Scatterplot with label	
Export dataset III View dataset	
Catterplot	

We have 420 instances and 280 attributes.

💇 TANAGRA 1.4.23 - [Da	taset (arrhytmia.txt)]			
Tile Diagram Componen	t Window Help			- 8 ×
D 📽 🖪  🏭				
De	fault title		Dataset (arrhytmia.txt)	<u> </u>
Dataset (arrhytmia.	txt)			
		Database : D:\DataMining\Database	Parameters ss_for_mining\benchmark_datas	ets\arrhythmia\arrhytmia.txt
			Results	
		Download inf	ormation	
		Datasource processir	ng	
		Computation time 0 m	15.	
		Allocated memory 771	KB	
-		Dataset descr 280 attribute(s) 420 example(s)	iption	×
		Components		
Data visualization	Statistics	Nonparametric statistics	Instance selection	Feature construction
Feature selection	Regression	Factorial analysis	PLS	Clustering
Spv learning	Meta-spv learning	Spv learning assessment	Scoring	Association
Correlation scatterplot	🖉 Scatterplot with label			
Export dataset	🔣 View dataset			
🚀 Scatterplot	🔛 View multiple scatterp	olot		

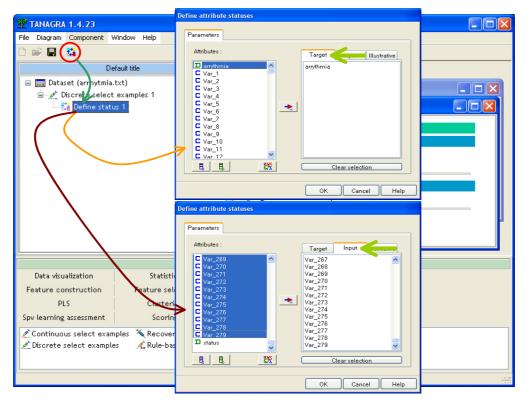
**Partitioning the dataset into train and test sets**. We use the STATUS column to define the train and test subsamples of our data. We insert the DISCRETE SELECT EXAMPLES (INSTANCE SELECTION tab) component into the diagram. We activate the PARAMETERS menu. In the setting box, we select STATUS as ATTRIBUTE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



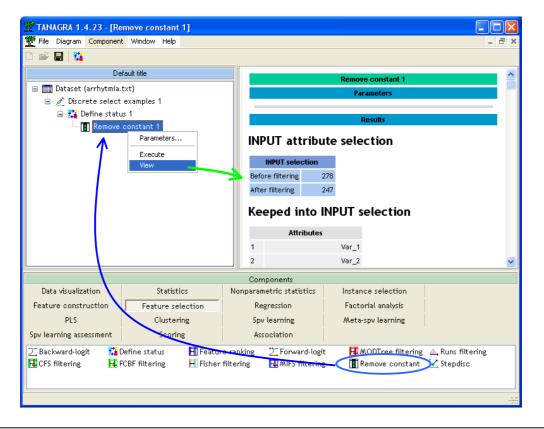
We confirm this choice by clicking on the OK button. We activate the contextual VIEW menu. We note that 232 instances now are assigned to the learning processes.

🕎 TANAGRA 1.4.23 - [Di	screte select examples 1]			
🏆 File Diagram Componer	t Window Help			_ 8 ×
🗅 📽 🔛 🔛				
De	efault title		Discrete select examples 1	
□ IIII Dataset (arrhytmia □ Discrete select		Attribute selection : sta Value selection : train 232 selected examples f Computation time : 0 ms Created at 12/05/2008	Parameters itus Results rom 420	
		Components		
Data visualization	Statistics	Nonparametric statistics	Instance selection	
Feature construction	Feature selection	Regression	Factorial analysis	
PLS	Clustering	Spv learning	Meta-spv learning	
Spv learning assessment	Scoring	Association		
∕ Continuous select exam ∕ Discrete select example		∕r Sampling n ∕r Stratified sampl	ing	
				a pa proven proven

**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.

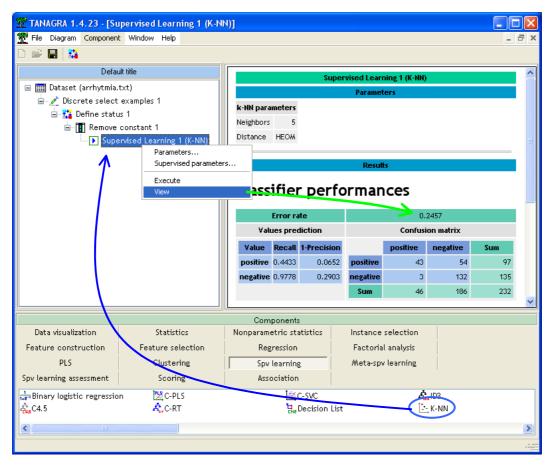


TANAGRA shows that 247 attributes are selected among the 278 descriptors. At the output of the component, the TARGET variable is not modified, but there are now 247 INPUT variables available for the modeling process.

#### 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.

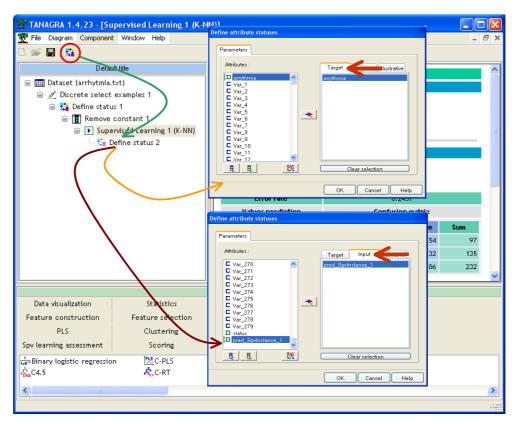
**Learning phase.** We insert the K-NN (SPV LEARNING tab) into the diagram. We click on the VIEW menu. Tanagra shows mainly the normalization parameters.



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.

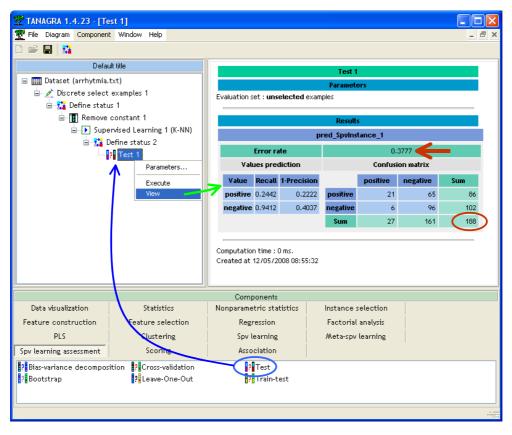
**Testing phase**. To obtain an honest estimation of the generalization error rate, we use the test sample. The only pitfall of this approach in our context is that the test sample size is not really enough to obtain a reliable estimation. But we know anyway that the error rate measured here will be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



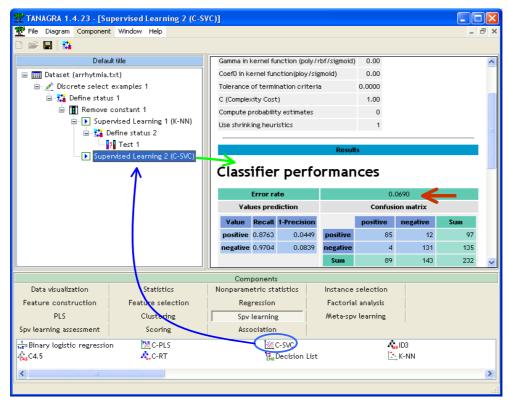
Second, we add the TEST component (SPV LEARNING ASSESSMENT tab) into the diagram. By default, it computes the error rate on the test set. The "true" error rate is 37.77%.

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%.



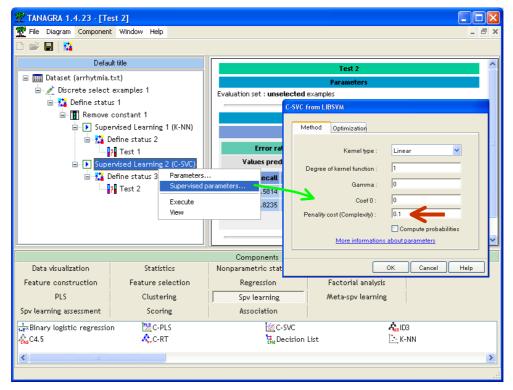
Once again, this value is not really interesting. We must evaluate the classifier on the test set (you can copy / paste a part of the diagram to repeat the sequence of operations – see <a href="http://data-mining-tutorials.blogspot.com/2009/06/copy-paste-feature-into-diagram.html">http://data-mining-tutorials.blogspot.com/2009/06/copy-paste-feature-into-diagram.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

💇 TANAGRA 1.4.23 - [Te	est 2]								
Tile Diagram Component	t Window Help							-	Ξ×
🗅 📽 🖪   🎎									
Defau	ult title				Test 2				~
🖃 🎹 Dataset (arrhytmia.	txt)				Paramete				
🛓 🖉 Discrete select	examples 1	Evaluation set : unselected examples							
🖃 🚰 Define statu	is 1		oc . anse		xampies				
🖮 👖 Remove	constant 1				Results	5			
	ervised Learning 1 (K-NN)				pred_Spvins	tance_2			
	Define status 2 2 Test 1		Error rat	te		0.3	2872 🧲	_	
-	ervised Learning 2 (C-SVC)	Valı	ues pred	iction		Confusi	on matrix		
	Define status 3 📘 📊	Value	Recall	1-Precisio	on	positive	negative	Sum	
l	P Test 2	positive	0.5814	0.264	47 positive	50	36	86	
	· · · · ·	negative	0.8235	0.300	00 negative	18	84	102	
					Sum	68	120	188	
									~
		Comp	ponents						
Data visualization	Statistics	Nonparame	tric stat	istics	Instance s	election			
Feature construction	Feature selection	Regr	ression		Factorial	analysis			
PLS	Clustering	Spv I	learning		Meta-spv	learning			
Spv learning assessment	Scoring	Asso	ciation						
<sup>1</sup> / <sub>1+2</sub> *Binary logistic regressio	n 🔀 C-PLS		<u>₩</u> c-	SVC		Ą.	IDG		
- <mark>- €4.5</mark> C4.5	AC-RT		Ha De	ecision Li	ist	<u> </u>	K-NN		
<									>
	<i>y</i>								

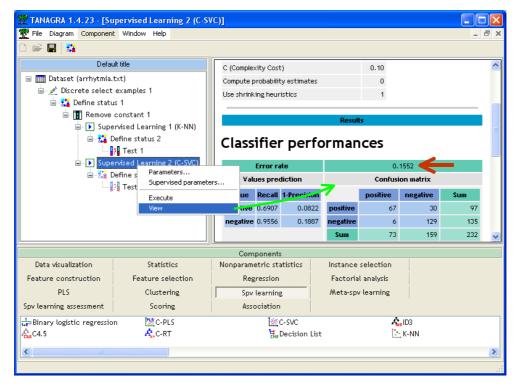
The test error rate is 28.72%. The improvement compared with the K-NN classifier is obvious.

**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.

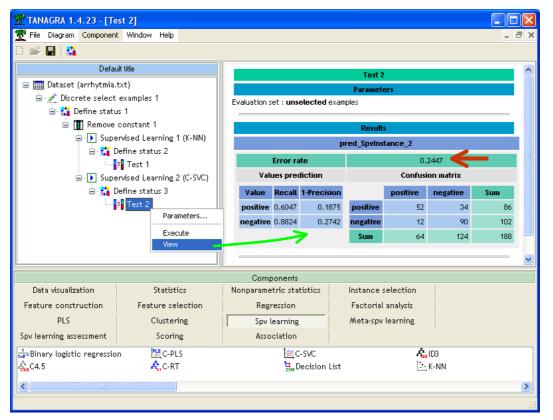
On our dataset, the gap between the learning and the test error rate seems to point out an overfitting. We try to set the new value of C to C = 0.1 (SUPERVISED PARAMETERS menu).



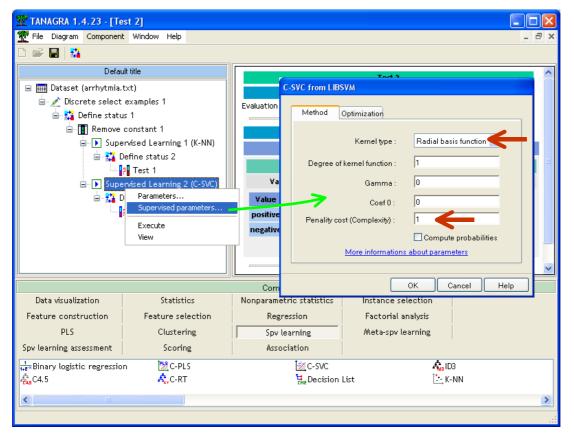
We observe that the resubstitution error rate is worse (15.52%).



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.



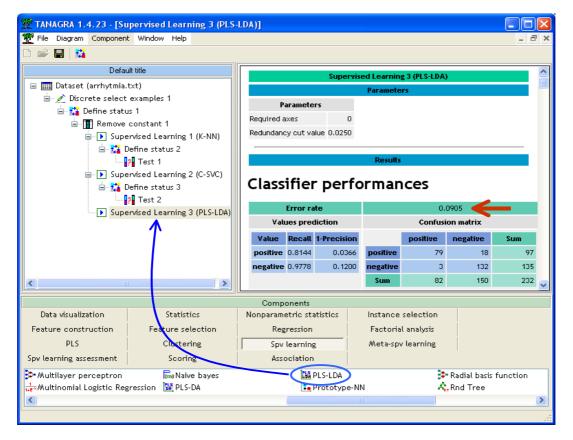
lassifier performances						
·						
E	Error ra	ite		0.:	2672	
Valu	es prec	diction		Confusi	on matrix	
Value	Recall	1-Precision		positive	negative	Sum
positive (	0.3711	0.0270	positive	36	61	97
negative (	0.9926	0.3128	negative	1	134	135
			Sum	37	195	232
		Apr	orent	issac	ne	

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

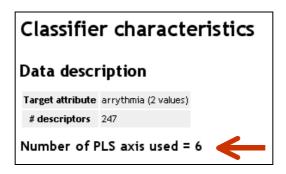
**PLS-LDA (6 factors)**. The PLS-LDA approach was described in a previous tutorial (<u>http://data-mining-tutorials.blogspot.com/2008/11/pls-regression-for-classification-task.html</u>).

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.



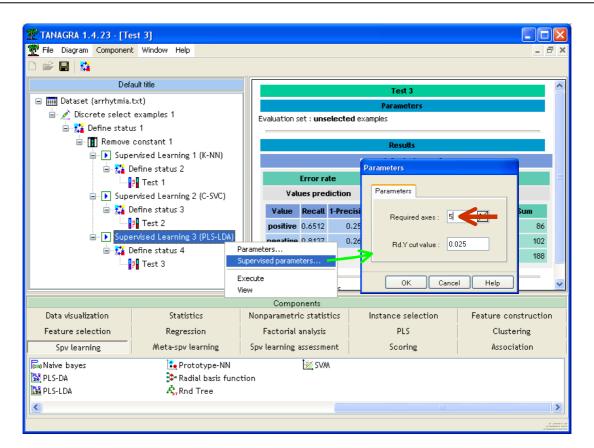
We fill out the diagram in order to obtain the test error rate. We have 26.06%.

TANAGRA 1.4.23 - [Te										
Tile Diagram Component	t Window Help								-	Ξ×
Defau	ult title					Test	3			
😑 🏢 Dataset (arrhytmia.	txt)	Parameters								
🛓 🖉 🖄 Discrete select	examples 1	Evaluation	Parameters Evaluation set : unselected examples							
🖨 🚰 Define statu	ıs 1									
🖮 📊 Remove	constant 1					Resut	ts			
	ervised Learning 1 (K-NN)				рг	ed_Spvins	tance 3			
	Define status 2		Error ra	to	- 1		_	2606 🧹		
	Part 1	W.	alues pred					on matrix		_
	ervised Learning 2 (C-SVC)				_				_	
	Define status 3	Value		1-Precis			positive	negative	Sum	
•	P Test 2 ervised Learning 3 (PLS-LDA)		e 0.6512	0.2		positive	56	30	86	
· · ·	ervised Learning 5 (PL5-LDA) Define status 4	negativ	<b>e</b> 0.8137	0.20	655	negative	19	83	102	
	Part 3	$\rightarrow$				Sum	75	113	188	
	· · · · · · · · · · · · · · · · · · ·	Computat	ion time · (	ושג						~
		Co	mponent	s						
Data visualization	Statistics	Nonparan	netric sta	tistics		Instance s	selection	Featur	e construct	ion
Feature selection	Regression	Facto	rial analy:	sis		PL	S	0	Clustering	
Spv learning	Meta-spv learning	Spv learn	ing assess	ment		Scor	ing	A	Association	
🔤 Naive bayes	🍡 Prototype-NN		🔣 S	VM						
🔛 PLS-DA	📴 Radial basis func	tion								
👪 PLS-LDA	🦧 Rnd Tree									
<										>

**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 redundacy - 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<sup>8</sup>.

		Supervi	sed Learnii	ng 3 (PLS-LD/	4)									
			Paramet	ers										
Р	aramete	:rs												
Required a	axes	5									Test	3		
Redundanc	cy cut va	alue 0.0250									Paramet	ers		
							Evaluat	on set :	: unse	elected exam	ples			
			Result	s										
Class	:										Result	s		
Class	ine	r perfo	orman	ces						р	red_Spvins	tance_3		
	Еггог га	ite		0.	1207 🧲			Err	гог га	te		0.	2394 🧲	_
Val	ues pre	diction			on matrix			∕alues	pred	liction		Confusi	on matrix	
Value	Recall	1-Precision		positive	negative	Sum	Vali	e Re	ecall	1-Precision		positive	negative	Sum
positive	0.7835	0.0843	positive	76	21	97	posit	<b>ve</b> 0.6	6512	0.2113	positive	56	30	86
negative	0.9481	0.1409	negative	7	128	135	nega	ive 0.8	8529	0.2564	negative	15	87	102
			Sum	83	149	232					Sum	71	117	188
											Т	est		
		4	Appre	entiss	age						16	551		

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.

<sup>&</sup>lt;sup>8</sup> Of course, the test file has few observations. We should not give too much importance to these small differences either.

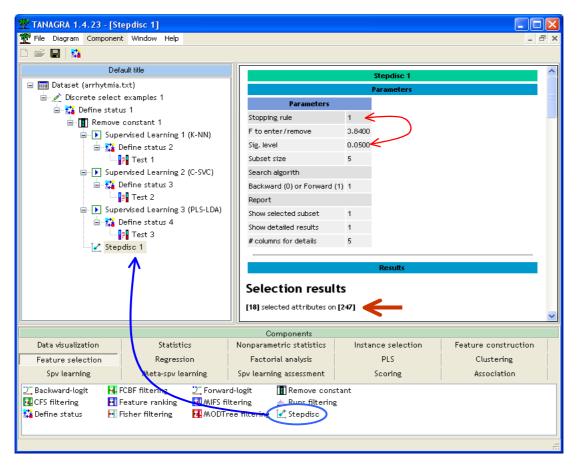
## 3.5 Linear discriminant analysis (LDA)

**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 - <u>http://en.wikipedia.org/wiki/Multivariate\_analysis\_of\_variance</u>).

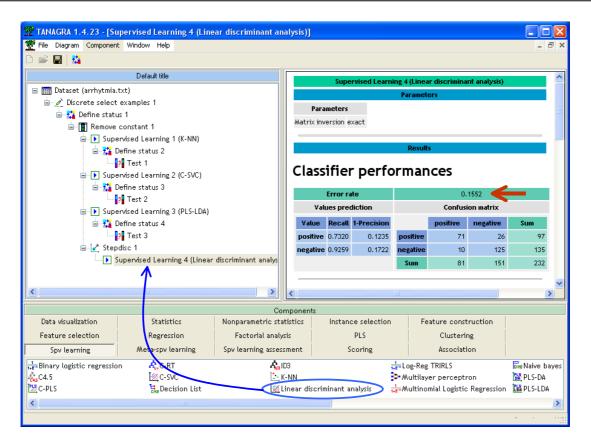
We insert the STEPDISC component (FEATURE SELECTION) into the diagram. We use the default settings. The component performs a FORWARD approach. The stopping rule is based on the comparison of the significance level<sup>9</sup> (0.05) specified by the user and the p-value<sup>10</sup> of the best predictor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



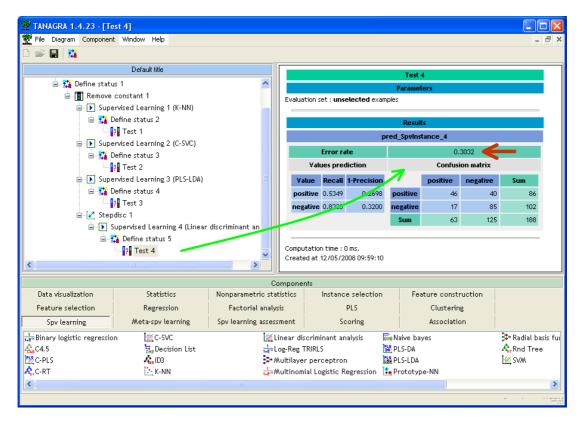
18 predictors are selected. We can add now the LINEAR DISCRIMNANT ANALYSIS (SPV LEARNING tab) to perform the construction of the classifier. The resubstitution error rate is 15.52%.

<sup>&</sup>lt;sup>9</sup> <u>http://en.wikipedia.org/wiki/Statistical\_significance</u>

<sup>&</sup>lt;sup>10</sup> <u>http://en.wikipedia.org/wiki/P-value</u>

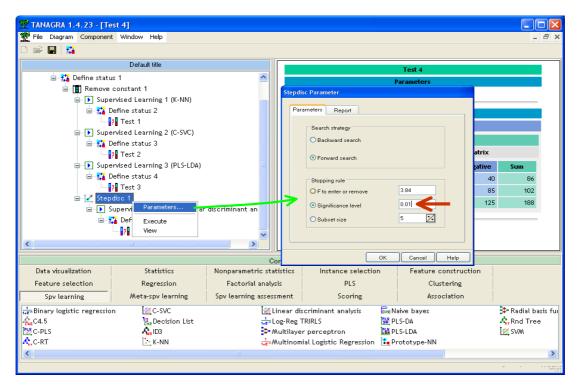


On the test set, the error rate becomes 30.32%. It is worse than all other methods discussed above. Clearly, there is an overfitting problem here. The number of selected variables is too high.

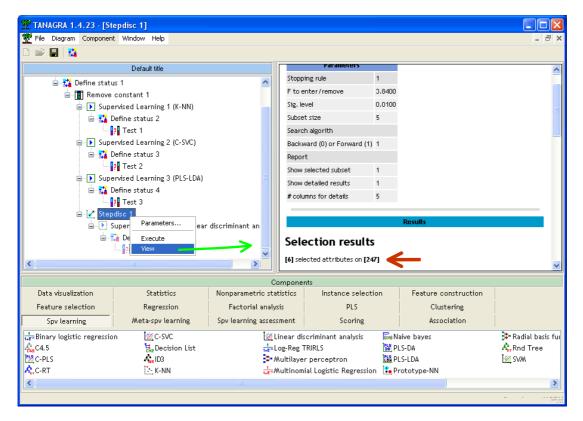


**Linear discriminant analysis (6 predictors)**. We try to decrease the number of selected predictors. To do that, we modify the settings of the algorithm (PARAMETERS menu). We set the

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.

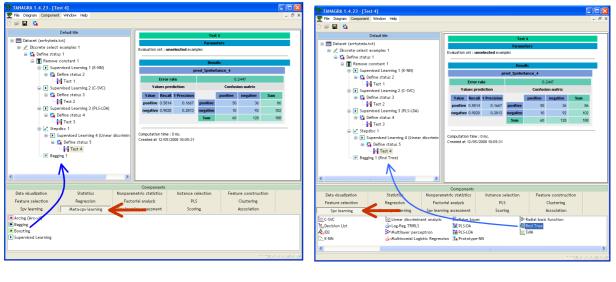
	Sup	ervised Leari	ning 4 (Line	ear discrimin	ant analysis)								
	Parameters												
Parameters										Test	4		
Matrix inv	atrix inversion exact						Parameters						
							Evaluation s	et : <b>uns</b>	elected exam	ples			
			Resu	tts									
Cl	: 6: ~ .									Resut	ts		
lass	iner	r perfo	orman	ices					р	red_Spvins	tance_4		
	Error ra	ite		0.	2241			Error ra	ite		0.	2447 🧲	
Valı	ues prec	liction		Confusi	ion matrix		Vali	ues pre	diction		Confus	ion matrix	
Value	Recall	1-Precision		positive	negative	Sum	Value	Recall	1-Precision		positive	negative	Sum
positive	0.5773	0.1642	positive	56	41	97	positive	0.5814	0.1667	positive	50	36	86
negative	0.9185	0.2485	negative	11	124	135	negative	0.9020	0.2813	negative	10	92	102
			Sum	67	165	232				Sum	60	128	188
Learning							Test						

However, the approach is dependent 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<sup>11</sup>. It designs a non-linear model.

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



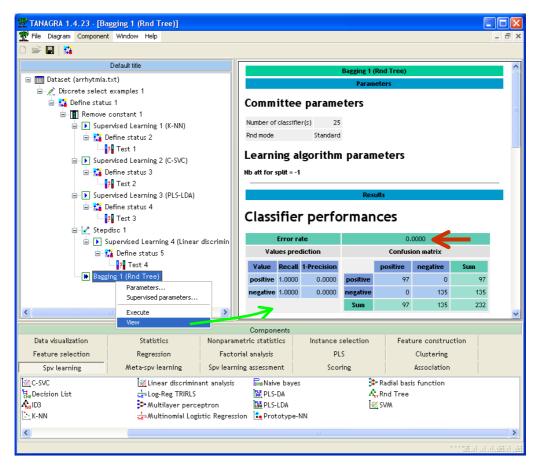
(1)

(2)

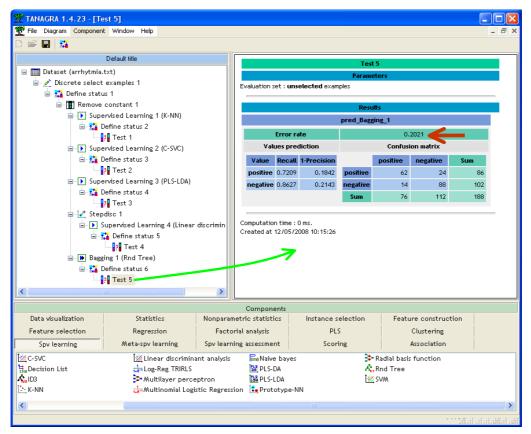
We click on the VIEW menu. The resubstitution error rate is 0%. It is usual for this approach.

<sup>&</sup>lt;sup>11</sup> <u>http://www.stat.berkeley.edu/~breiman/RandomForests/</u>; <u>http://en.wikipedia.org/wiki/Random\_forest</u>

<sup>&</sup>lt;sup>12</sup> <u>http://data-mining-tutorials.blogspot.com/2008/11/random-forest.html</u>



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.

TANAGRA 1.4.23 - [Test 5] File Diagram Component Window Help E. 学 🖬 | 👪 Default title 🖃 🏢 Dataset (arrhytmia.txt) 🛓 💉 Discrete select exam Evaluation set : **unselected** examples 😑 🔂 Define status 1 🖮 🔳 Remove constant 1 Supervised Learning 1 (K-NN) pred Bagging 1 🚊 🚰 Define status 2 Error ra - P Test 1 Values prediction Supervised Learning 2 (C-SVC) Confusion matrix 😑 🏂 Define status 3 Value Recall 1-Pr - P Test 2 positive 0.7209 0.1842 62 24 Supervised Learning 3 (PLS-LDA) negative 0.8627 🖮 🏠 Define status 4 of selected attributes for Pest 3 🚊 🛃 Stepdisc 1 omputation time : 0 ms Supervised Learning 4 (Linear discrimin Created at 12/05/2008 1 🖕 🏠 Define status 5 ed attrib - Past 4 🖮 🕨 Ba 🚰 Define status 6 Parameters OK Cancel Help P Test 5 Execute Components Data visualization Statistics Nonparametric statistics Instance selection Feature construction Feature selection Regression Factorial analysis PLS Clustering Meta-spv learning Spv learning assessment Scoring Association Spv learning 🗽 c-svo 🕍 Linear discriminant analysis 🔝 Naive bayes 🗫 Radial basis function 📜 Decision List Log-Reg TRIRLS 🔛 PLS-DA A. Rnd Tree A ID3 Multilayer perceptron 👪 PLS-LDA 🛛 🔣 SVAA 🔄 K-NN 📩 Multinomial Logistic Regression 🚺 Prototype-NN

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.

		Bagging 1 (	(Rnd Tree)				
		Param	neters				
ommitt	tee parame	eters					
lumber of clas	ssifier(s) 25						
nd mode	Standard						
earning	g algorithm	n paran	neters			Test 5	
b att for split	t = 50					Parameters	
						Evaluation set : unselected examples	
		Res	utts				
-						Results	
-	ier perfo						
Classifi	-		ices	0000		Results	
Classifi	or rate		1 <b>CES</b> 0.	0000		Results pred_Bagging_1	
Classifi Errc Values J	or rate prediction		ICES 0. Confusi	ion matrix	Sim	Results       pred_Bagging_1       Error rate     0,1596       Values prediction     Confusion matrix	Sum
Classifi Erro Values J Value Rec	or rate prediction call 1-Precision	orman	CCCS 0. Confusi positive	ion matrix negative	Sum 97	Results       pred_Bagging_1       Error rate     0.1596       Values prediction     Confusion matrix	Sum
Classifi Erro Values J Value Rec positive 1.00	or rate prediction call 1-Precision 0000 0.0000	positive	Ces 0. Confusi positive 97	ion matrix negative 0	97	Results         pred_Bagging_1         0.1596         Confusion matrix         Values prediction       confusion matrix         Value       Recall       1-Precision	
Classifi Erro Values J Value Rec	or rate prediction call 1-Precision 0000 0.0000	orman	CCCS 0. Confusi positive	ion matrix negative		Results         Free_Bagging_1         DISPOSE SUBJE         DISPOSE SUBJE         Value       Results         Value       Results         OISTON SUBJE         positive       negative         positive       negative         positive       negative         positive       negative         positive       negative	Sum

#### 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.