

# 1 Topic

## “Filter” approaches for the selection of discrete predictors in supervised learning.

The nature of the predictors' selection process has changed considerably. Previously, works in machine learning concentrated on the research of the best subset of features for a learning classifier, in the context where the number of candidate features was rather reduced and the computing time was not a major constraint. Today, it is common to deal with datasets comprising thousands of descriptors. Consequently, the problem of feature selection always consists in finding the most relevant subset of predictors but by introducing a new strong constraint: the computing time must remain reasonable.

There are 3 main feature selection approaches in supervised learning context. The first one is the **embedded** methods. The selection process is incorporated into the learning algorithms. Particularly, the criterion used for the selection of the variable is the same that the one used for the construction of the model. For instance, in the decision tree induction, the selection of a variable for splitting on a node is based on the entropy gain. And the construction of the whole tree can be viewed as a global maximization of entropy gain (the leaves of the tree vs. the root node). The second one is the **wrapper** approach. Wrapper uses the learning algorithm as a black box to select the best subset of features. It uses cross-validation to compare the error rate of the candidate subsets. Even if it can be very powerful in many cases, it presents two major disadvantages: it requires more computational resources because the learning process is repeatedly called, therefore it is not really suitable when the number of predictors is very large; the repeated use of cross-validation on a single dataset can lead to growth of the probability of overfitting i.e. finding solutions that performs well only on our dataset. The last one is the **filter** approach. It consists in selecting in an independent way, with ad hoc criterion, the best subset of features. Very efficient approaches were developed. They make it possible to treat quickly a dataset containing a large number of variables. In spite of its recognized qualities, this approach is based nevertheless on a strong conjecture which is not always well controlled: the subset of features selected by the filtering method would be most powerful whatever the characteristics of the subsequent learning algorithm implemented. Yet, empirical studies show that this approach is efficient, even if the subsequent supervised learning algorithm incorporates embedded feature selection techniques (e.g. decision tree induction).

In this tutorial, we are interested in correlation based filter approaches for discrete predictors. The goal is to highlight the most relevant subset of predictors which are highly correlated with the target attribute and, in the same time, which are weakly correlated between them i.e. which are not redundant. To evaluate the behavior of the various methods, we use an artificial dataset where we add irrelevant and redundant candidate variables. Then, we perform a feature selection based on the approaches analyzed. We compare the generalization error rate of the naive bayes classifier learned from the various subsets of selected variables. We lead the experimentation with Tanagra in a first time. Then, in a second time, we show how to perform the same analysis with other tools ([Weka 3.6.0](#), [Orange 2.ob](#), [RapidMiner 4.6.0](#) and [R 2.9.2](#) – package [FSelector](#)).

## 2 Correlation based filter method for feature selection

The process is based on the measurement of the association between variables. We use the generic term "correlation" even if we handle discrete variables.

### 2.1 Correlation between discrete variables

#### 2.1.1 Measuring the association

The measure of association between two discrete variables is computed from the contingency table formed from these variables. Let  $Y$  the target attribute  $\{y_1, \dots, y_k, \dots, y_K\}$ ,  $X \in \{x_1, \dots, x_l, \dots, x_L\}$  is the predictor. We use the following notation.

$Y \setminus X$	$x_1$	...	$x_l$	...	$x_L$	$\Sigma$
$y_1$						
$\vdots$			$\vdots$			
$y_k$		...	$n_{kl}$	...		$n_{k.}$
$\vdots$			$\vdots$			
$y_K$						
$\Sigma$			$n_{.l}$			$n$

We compute various relative frequencies:  $p_{kl} = \frac{n_{kl}}{n}$ ;  $p_{k.} = \frac{n_{k.}}{n}$ ;  $p_{.l} = \frac{n_{.l}}{n}$ .

The mutual information measures the mutual dependence between two variables:

$$I(Y, X) = \sum_k \sum_l p_{kl} \times \log_2 \frac{p_{kl}}{p_{k.} \times p_{.l}}$$

Let  $H(Y)$  [resp.  $H(X)$ ] the marginal entropy of  $Y$  (resp.  $X$ ), it measures the uncertainty about the variable. It can be considered also as a measure of dispersion. We have:

$$H(Y) = -\sum_k p_{k.} \log_2 p_{k.}$$

We define the symmetrical uncertainty as follows:

$$\rho_{y,x} = 2 \times \left[ \frac{I(Y, X)}{H(Y) + H(X)} \right]$$

It is defined into  $[0; 1]$ . In the numerator, we have a something like a covariance of the two variables. In the denominator, we have something like the variances of the two variables. In a certain point of view, the symmetrical uncertainty measure is similar to the Pearson's correlation coefficient.

The measure is symmetrical. Perhaps in the prediction context, it is more suitable to use an asymmetric measure. But, we do not forget that the same measure is used to evaluate the redundancy between the predictors. Thus, this measure is well adapted for our task.

**Numerical example.** We illustrate the calculation of the measure on the dataset used in this tutorial. We set "group" as target variable Y, "adoption-of-the-budget" as X. We obtain the following contingency table.

Nombre de group	adoption-of-the-budget			
group	n	other	y	Total
democrat	29	7	231	267
republican	142	4	22	168
Total	171	11	253	435

We divide all the values with the number of instances ( $n = 435$ )

Nombre de group	adoption-of-the-budget			
group	n	other	y	Total
democrat	0.0667	0.0161	0.5310	0.6138
republican	0.3264	0.0092	0.0506	0.3862
Total	0.3931	0.0253	0.5816	1.0000

For the mutual information, we have

$$I(Y, X) = 0.0667 \times \log_2 \frac{0.0667}{0.3931 \times 0.6138} + \dots + 0.0506 \times \log_2 \frac{0.0506}{0.5816 \times 0.3862}$$

$$= 0.4323$$

For the marginal entropy of Y

$$H(Y) = -[0.6138 \times \log_2 0.6138 + 0.3862 \times \log_2 0.3862] = 0.9623$$

And H(X)

$$H(X) = -[0.3931 \times \log_2 0.3931 + 0.0253 \times \log_2 0.0253 + 0.5816 \times \log_2 0.5816] = 1.1184$$

Thus, we obtain

$$\rho = 2 \times \left[ \frac{I(Y, X)}{H(Y) + H(X)} \right] = 2 \times \left[ \frac{0.4323}{0.9623 + 1.1184} \right]$$

$$= 0.4155$$

### 2.1.2 Testing the significance of the association

When the variables are independent, we have  $\rho = 0$ . The larger is the value of  $\rho$ , the higher is the dependency. But, we do not know starting from which value we can consider that the association is significant. We must use a statistical hypothesis testing scheme for that.

We define the test statistic

$$G = 2 \times n \times \ln(2) \times I(Y, X)$$

It follows a (chi-squared)  $\chi^2$  distribution with  $(K-1) \times (L-1)$  degrees of freedom under the hypothesis of independence between Y and X (null hypothesis). For the significance level  $\alpha$ , the critical region of the test is defined as  $G > \chi_{1-\alpha}^2$ , where  $\chi_{1-\alpha}^2$  is the percentile of the chi-squared distribution. Another way to define the critical region is to compare the p-value ( $p_c$ ) to the significance level.

**Numerical example.** For the analysis above, we obtain

$$\begin{aligned} G &= 2 \times n \times \ln(2) \times I(Y, X) \\ &= 2 \times 435 \times \ln(2) \times 0.4323 \\ &= 260.7046 \end{aligned}$$

Using the  $\chi^2$  distribution with  $(3-1) \times (2-1) = 2$  degrees of freedom, we obtain the p-value  $p_c = 3.16 \times 10^{-56} \approx 0$ . The association is very significant.

*Note: The measured association is almost always significant when we deal with a large dataset. This test of significance is not really useful in the data mining context. What matters most is that we can use the p indicator to detect the interesting predictors and to rank them.*

### 2.1.3 Testing the significance (again)

Another way to assess the significance of the association is to use the normal approximation of the distribution of  $\rho$ . This approach is more conservative (it favors the null hypothesis). But, it enables to compute the confidence interval of the measure.

Let

$$H(Y, X) = -\sum_k \sum_l p_{kl} \times \log_2 p_{kl}$$

The standard error can be obtained from<sup>1</sup>

$$\sigma_\rho^2 = 4 \times \sum_k \sum_l \frac{n_{kl} \left[ H(Y, X) \times \log_2 \left( \frac{n_k \times n_l}{n^2} \right) - [H(Y) + H(X)] \times \log_2 \left( \frac{n_{kl}}{n} \right) \right]^2}{n^2 \times [H(Y) + H(X)]^4}$$

Under the null hypothesis, the standard error becomes

$$\sigma_\rho^2(0) = 4 \times \frac{\sum_k \sum_l n_{kl} \times \left[ \log_2 \left( \frac{n_k \times n_l}{n \times n_{kl}} \right) \right]^2 - \frac{[H(Y) + H(X) - H(Y, X)]^2}{n}}{n^2 \times [H(Y) + H(X)]^2}$$

The critical region is defined as follows

$$\frac{\rho}{\sigma_\rho(0)} > u_{1-\alpha}$$

<sup>1</sup> <http://v8doc.sas.com/sashtml/stat/chap28/sect20.htm>

$u_{1-\alpha}$  is the quantile of the Gaussian distribution.

**Numerical example.** With our example above, we have  $H(Y, X) = 1.6484$ ; we compute the squared of the standard error

$$\sigma_{\rho}^2 = \frac{4 \times 1626.6563}{435^2 \times [0.9623 + 1.1184]^4} = \frac{6506.6252}{3546865.1306} = 0.0018$$

The confidence interval at  $(1 - \alpha) = 95\%$  confidence level is

$$\begin{aligned} & [\rho \pm u_{1-\alpha/2} \times \sigma_{\rho}] \\ & [0.415544 - 1.96 \times 0.0428 ; 0.415544 + 1.96 \times 0.0428] \\ & [0.3316 ; 0.4995] \end{aligned}$$

To assess the significance, we compute the standard error under the null hypothesis

$$\sigma_{\rho}^2(0) = 4 \times \frac{450.7425 - \frac{(0.9623 + 1.1184 - 1.6484)^2}{435}}{435^2 \times [0.9623 + 1.1184]^2} = 0.0022$$

We reject the null hypothesis at the significance level  $\alpha$  because

$$\frac{\rho}{\sigma_{\rho}(0)} = \frac{0.415544}{0.0469} = 8.8579 > u_{1-\alpha} = 1.6449$$

According to the hypothesis testing based on the chi-squared distribution, we conclude that the association between "group" and "adoption-of-budget" is statistically significant.

## 2.2 "Ranking" approaches based on the correlation measure

The ranking approach is based only on the association with the target attribute. It does not take into account the redundancies between the predictors. Roughly, we compute the correlation of each predictor with the target attribute. Then, we rank them in a decreasing order according  $\rho$ . To determine the right number of predictors, we can test their significance. But, we note that this strategy tends to select too many predictors. Empirical strategy such as detecting the sudden decreasing of the measurement, or directly specify the number of desired predictors can be used.

In our dataset, we observe the correlation of the 20 "best" predictors with the target attribute. All are significant. We note however that the decreasing is notable after the first predictor, after the second predictor, and after the 14-th predictor. These are elements that can help us choose the relevant subset of predictors.

But the main drawback of this approach is the total lack of consideration of redundancy. In fact, if the best (the most correlated with the target attribute) predictor is duplicated 10 times, all will be selected. It is really a problem, especially when we have to deal with data sets containing a large number of redundant variables.

## Calculations details

N°	Attribute	Values	Statistic	Statistic (Histogram)	p-value
1	physician-fee-freeze	3	0.708862		0.000000
2	corr_physician-fee-freeze	3	0.540679		0.000000
3	adoption-of-the-budget	3	0.415544		0.000000
4	el-salvador-aid	3	0.394048		0.000000
5	corr_adoption-of-the-budget-re	3	0.371640		0.000000
6	corr_el-salvador-aid	3	0.366040		0.000000
7	education-spending	3	0.333286		0.000000
8	aid-to-nicaraguan-contras	3	0.319763		0.000000
9	crime	3	0.313788		0.000000
10	corr_aid-to-nicaraguan-contras	3	0.288226		0.000000
11	corr_crime	3	0.287527		0.000000
12	mx-missile	3	0.282252		0.000000
13	corr_education-spending	3	0.273481		0.000000
14	corr_mx-missile	3	0.269558		0.000000
15	superfund-right-to-sue	3	0.205050		0.000000
16	duty-free-exports	3	0.197825		0.000000
17	corr_duty-free-exports	3	0.194450		0.000000
18	anti-satellite-test-ban	3	0.186272		0.000000
19	corr_superfund-right-to-sue	3	0.179718		0.000000
20	corr_anti-satellite-test-ban	3	0.160502		0.000000

### 2.3 Handling feature redundancy

In this section, we present the methods which take into account the redundancy between the predictors. The selected subset must be as parsimonious as possible. So we need to combine appropriately the correlation of predictors with the target variable on the one hand, and cross-correlations between selected variables on the other hand.

#### 2.3.1 The CFS approach

The CFS<sup>2</sup> method is based on the “merit” criterion. It evaluates globally the efficiency of a set of predictors, by realizing the trade-off between the relevance (according to the target attribute) and the redundancy (between the predictors).

$$merit = \frac{m \times \bar{\rho}_{y,x}}{\sqrt{m + m \times (m-1) \times \bar{\rho}_{x,x}}}$$

<sup>2</sup> M. Hall, S. Lloyd, « Feature subset selection: a correlation based filter approach », in 1997 Int. Conf. On Neural Information Processing and Intelligent Information Systems, pp/ 855-858, Springer, 1997.

Where  $\bar{\rho}_{y,x}$  is the mean of the correlation of the selected predictors and the target attribute;  $\bar{\rho}_{x,x'}$  is the mean of the correlation between the predictors.

Thus, the selection problem becomes an optimization problem. We must detect the subset which maximizes the MERIT criterion. We can use simple greedy strategies (stepwise approaches, forward or backward) or sophisticated ones (e.g. genetic algorithms, simulated annealing, etc.). In practice, a greedy approach, by smoothing the exploration of space of solutions, is enough. It avoids the overfitting.

**Complexity of the algorithm.** In the forward approach, the calculations of all the correlations can be made on one pass over the database. But, the selection of the best subset is quadratic in relation to the number of candidate predictors. Thus, this approach is especially convenient on a dataset with a large number of instances, but a moderate number of predictors. Otherwise, when the number of descriptors is very large, calculation and memory storage of all cross-correlations become a problem. It is more advantageous to calculate the correlations on the fly. The experiments show that the number of predictors finally selected is often small.

### 2.3.2 The MIFS approach

The MIFS (Mutual Information Feature Selection) algorithm uses a forward selection (Battiti, 1994)<sup>3</sup>. At each step, the predictor X which maximizes the following criterion is added (Z is the set of predictors already selected)

$$I(Y, X / M) = I(Y, X) - \beta \times \sum_{Z \in M} \frac{I(X, Z)}{m}$$

The search is stopped when we cannot improve the measure i.e. the last best predictor X\* is such as  $I(Y, X^* / M) \leq 0$ . The algorithm is quadratic in relation to the number of candidate predictors.

MIFS is not really better than CFS. In addition, it depends heavily on the  $\beta$  parameter. This is an advantage in some circumstances because we can adapt it to the problem characteristics. But in practice, it is not easy to set the good value according to the dataset.

### 2.3.3 The FCBF approach

[FCBF](#) (Yu and Liu, ICML 2003) uses also the symmetrical uncertainty measure. But the search algorithm is very different. It is based on the "predominance" idea. The correlation between an attribute X\* and the target Y is predominant if and only if

$$\rho_{y,x^*} \geq \delta \text{ et } \forall X (X \neq X^*), \rho_{x,x^*} < \rho_{y,x^*}$$

Concretely, a predictor is interesting if its correlation with the target attribute is significant (delta is the parameter which allows to assess this one); there is no other predictor which is more strongly correlated to it.

<sup>3</sup> R. Battiti, «Using mutual information for selecting features in supervised neural net learning», IEEE Transactions on Neural Networks, 5(4) : 537-550, 1994.

Based on this idea, the authors propose a search algorithm which runs in **quasilinear** time.

1.  $S$  is the set of candidate predictors,  $M = \emptyset$  is the set of selected predictors
2. Searching  $X^*$  (among  $S$ ) which maximizes its correlation with  $Y \rightarrow \rho_{y,x^*}$
3. If  $\rho_{y,x^*} \geq \delta$ , add  $X^*$  into  $M$  and remove  $X^*$  from  $S$
4. Remove also from  $S$  all the variables  $X$  such as  $\rho_{x,x^*} \geq \rho_{y,x^*}$  (Very important !)
5. If  $S \neq \emptyset$  then GOTO (2), else END of the algorithm

This approach is very useful when we deal with a dataset containing a very large number of candidate predictors. About the ability to detect the "best" subset of predictors, as we will see in this tutorial, we note that it is similar to CFS.

### 2.3.4 The MODTREE approach

The MODTREE (Rakotomalala and Lallich, 2002)<sup>4</sup> is based also on the ideas of relevance and redundancy. But it uses a different correlation measure. Using the principle of pairwise comparison, the correlation between two attributes is computed as follows:

$$r_{y,x} = \frac{g_{11}g_{22} - g_{12}g_{21}}{\sqrt{g_{1.} \times g_{2.} \times g_{.1} \times g_{.2}}}, \text{ where}$$

- $g_{11} = \frac{1}{2} \sum_k \sum_l n_{kl}^2$
- $g_{12} = \frac{1}{2} \sum_k \sum_l n_{kl} (n_{k.} - n_{kl})$
- $g_{1.} = g_{11} + g_{12}$
- $g_{21} = \frac{1}{2} \sum_k \sum_l n_{kl} (n_{.l} - n_{kl})$
- $g_{22} = \frac{1}{2} \sum_k \sum_l n_{kl} (n - n_{k.} - n_{.l} + n_{kl})$
- $g_{2.} = g_{21} + g_{22}$
- $g_{.1} = g_{11} + g_{21}$
- $g_{.2} = g_{12} + g_{22}$

Even if it is based on the pairwise comparison, the measure is computed in a linear time according to the number of instances. Thus, it is operational when we deal with very large dataset.

To implement the forward search, the partial correlation between  $Y$  and  $X$ , by controlling a third variable  $Z$ , is defined as follows ([http://en.wikipedia.org/wiki/Partial\\_correlation](http://en.wikipedia.org/wiki/Partial_correlation))

<sup>4</sup> Rakotomalala R., Lallich S., "Construction d'arbres de décision par optimisation", Revue Extraction des Connaissances et Apprentissage, vol. 16, n°6/2002, pp.685-703, 2002.

$$r_{y,x/z} = \frac{r_{y,x} - r_{y,z} \times r_{x,z}}{\sqrt{(1-r_{y,z}^2)(1-r_{x,z}^2)}}$$

When  $m$  predictors  $M = \{Z_1, \dots, Z_m\}$  are already selected, the partial correlation becomes

$$r_{y,x/z_1 \dots z_m} = \frac{r_{y,x/z_1 \dots z_{m-1}} - r_{y,z_m/z_1 \dots z_{m-1}} \times r_{x,z_m/z_1 \dots z_{m-1}}}{\sqrt{(1-r_{y,z_m/z_1 \dots z_{m-1}}^2)(1-r_{x,z_m/z_1 \dots z_{m-1}}^2)}}$$

Like CFS and MIFS, the search algorithm is quadratic according to the number of candidate predictors. We observe that we must compute all the cross-correlations between the predictors. And we must update them each time we select a predictor i.e. adding a predictor into  $M$ .

We use the following stopping rule: we do not select a new predictor at the step  $(m+1)$  if the best attribute meets to the condition

$$r_{y,x^*/z_1 \dots z_m} < \frac{1}{\sqrt{n-m}}$$

MODTREE is very similar to CFS and FCBF concerning to the ability to detect the best subset of predictors.

**Numerical example.** We detail the partial correlation mechanism with some variables of our dataset ( $Y = \text{« group »}$ ,  $X_1 = \text{« physician fee freeze »}$ ,  $X_2 = \text{« adoption-of-budget »}$ ,  $X_3 = \text{« education spending »}$ ).

First, we calculate the correlation

Y	X	r	r <sup>2</sup>
adoption-of-the-budget	physician-fee-freeze	0.5328	0.2838
adoption-of-the-budget	education-spending	0.4005	0.1604
adoption-of-the-budget	group	0.5464	0.2986
physician-fee-freeze	education-spending	0.4529	0.2051
physician-fee-freeze	group	0.8097	0.6556
education-spending	group	0.4545	0.2066

$(Y, X_1)$   $r_{y,x_1} = 0.8097$ ,  $X_1$  is the most correlated with  $Y$ .

$(Y, X_2 / X_1)$  Now, we characterize the additional information given by  $X_2$  about  $Y$ , by controlling the first variable  $X_1$ . Because  $r_{x_2,x_1} = 0.5328$  and  $r_{y,x_2} = 0.5464$ , we obtain

$$r_{y,x_2/x_1} = \frac{0.5464 - 0.8097 \times 0.5328}{\sqrt{(1-0.8097^2)(1-0.5328^2)}} = 0.2316$$

The association is less strong because  $X_2$  is both correlated with  $X_1$  and  $Y$ .

(Y, X<sub>3</sub> / X<sub>1</sub>, X<sub>2</sub>) If we want to characterize the association between Y and X<sub>3</sub>, by controlling simultaneously X<sub>1</sub> and X<sub>2</sub>, we must compute before

$$r_{y,x_3/x_1} = \frac{r_{y,x_3} - r_{y,x_1} \times r_{x_3,x_1}}{\sqrt{(1-r_{y,x_1}^2)(1-r_{x_3,x_1}^2)}} = \frac{0.4545 - 0.8097 \times 0.4529}{\sqrt{(1-0.8097^2)(1-0.4529^2)}} = 0.1678$$

$$r_{x_2,x_3/x_1} = \frac{r_{x_2,x_3} - r_{x_2,x_1} \times r_{x_3,x_1}}{\sqrt{(1-r_{x_2,x_1}^2)(1-r_{x_3,x_1}^2)}} = \frac{0.4005 - 0.5328 \times 0.4529}{\sqrt{(1-0.5328^2)(1-0.4529^2)}} = 0.2110$$

Then, we can calculate the second order partial correlation.

$$r_{y,x_3/x_1,x_2} = \frac{r_{y,x_3/x_1} - r_{y,x_2/x_1} \times r_{x_3,x_2/x_1}}{\sqrt{(1-r_{y,x_2/x_1}^2)(1-r_{x_3,x_2/x_1}^2)}} = \frac{0.1678 - 0.2316 \times 0.2110}{\sqrt{(1-0.2316^2)(1-0.2110^2)}} = 0.1251$$

### 3 Dataset

To analyze the behavior of the methods presented here, we use the famous "congress vote" dataset (<http://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records>). But we have modified the dataset characteristics by adding new predictors. We have duplicated twice the original predictors by using the following principle:

1. For each variable X, we generate "noise\_X" which have (approximately) the same values distribution, but which is not correlated to X.

For instance, about the variable "adoption-of-budget", we observe that the relative frequencies are very similar.

Nombre de adoption-of-the-budget	
adoption-of-the-budget	Total
n	39.31%
other	2.53%
y	58.16%
Total	100.00%

Nombre de noise adoption-of-the-budget-re	
noise_adoption-of-the-budget-re	Total
n	39.77%
other	4.60%
y	55.63%
Total	100.00%

But the variables are unrelated as we can see with the chi-square test for independence.

Results								
Row (Y)	Column (X)	Statistical indicator		Cross-tab				
		Stat	Value		y	other	n	Sum
		d.f.	4	n	90	9	72	171
		Tschuprow's t	0.054835	y	147	11	95	253
		Cramer's v	0.054835	other	5	0	6	11
		Phi <sup>2</sup>	0.006014	Sum	242	20	173	435
adoption-of-the-budget	noise_adoption-of-the-budget-re	Chi <sup>2</sup> (p-value)	2.62 (0.6240)					
		Lambda	0.000000					
		Tau (p-value)	0.0030 (0.6249)					
		U(R/C) (p-value)	0.0045 (0.5465)					

2. For each variable  $X$ , we have generated "corr\_X" which is highly correlated to  $X$ , independently to the target attribute  $Y$ . In our experimentation, the probability that  $X$  and corr\_X have the same value is (approximately) 97%.

About "adoption-of-budget", the result of the chi-square test for independence shows the strong association.

Results								
Row (Y)	Column (X)	Statistical indicator		Cross-tab				
		Stat	Value		n	y	other	Sum
		d.f.	4	n	165	6	0	171
		Tschuprow's t	0.978187	y	3	250	0	253
		Cramer's v	0.978187	other	0	0	11	11
		Phi <sup>2</sup>	1.913699	Sum	168	256	11	435
adoption-of-the-budget	corr_adoption-of-the-budget-re	Chi <sup>2</sup> (p-value)	832.46 (0.0000)					
		Lambda	0.950549					
		Tau (p-value)	0.9201 (0.0000)					
		U(R/C) (p-value)	0.8710 (0.0000)					

The purpose of this tutorial is to show the ability of algorithms for filtering predictors: (1) to identify those that are most effective among the original descriptors; (2) to remove the noisy attributes (noise\_ " ); (3) not to be misled by the lures that are redundant variables (corr\_) i.e. preferring  $X$  to corr\_X during selection. To avoid too restrictive selection, we evaluate the subset of selected predictors by implementing them in a naive bayes classifier. Indeed, we must select a small number of variables, but they must enable to build an efficient classifier.

## 4 Filtering approaches with Tanagra

### 4.1 Learning and assessing the model with all the predictors

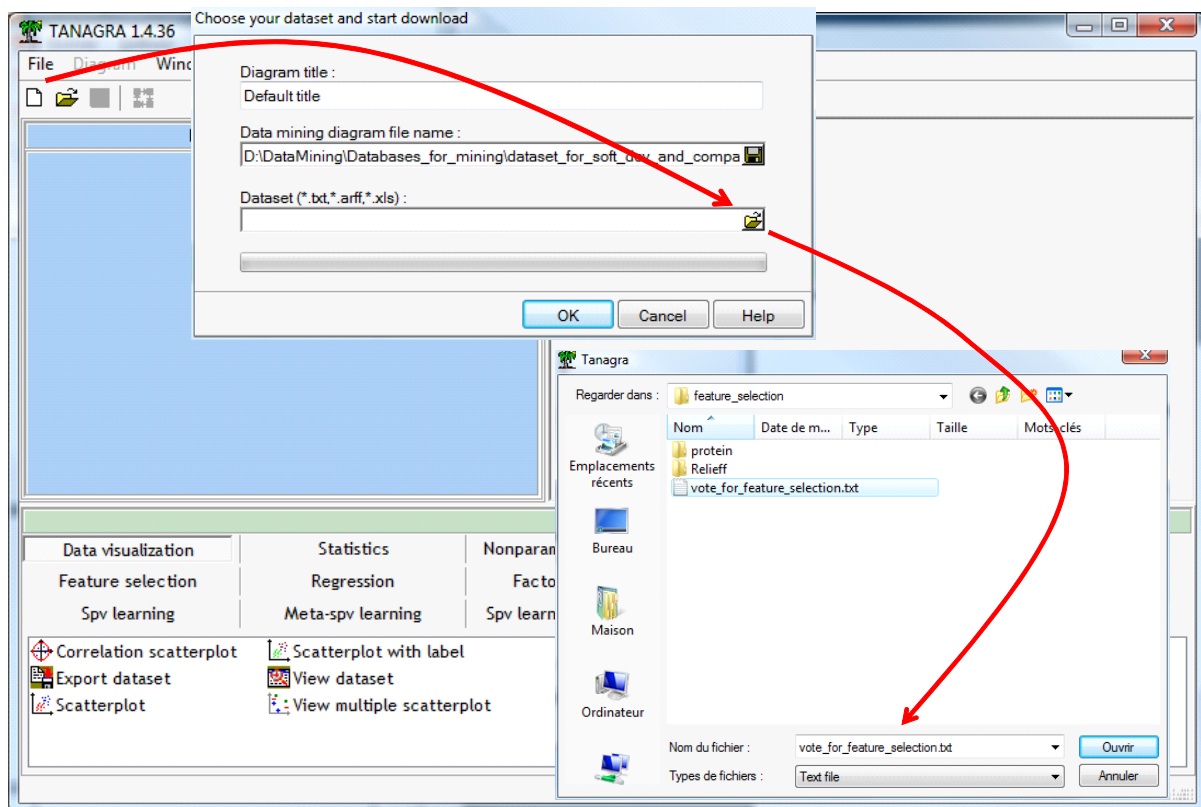
In a first step, we learn the model (naive bayes classifier<sup>5</sup>) using all the predictors ( $16 \times 3 = 48$ ). We estimate the generalization error rate using the bootstrap<sup>6</sup>.

#### 4.1.1 Importing the dataset

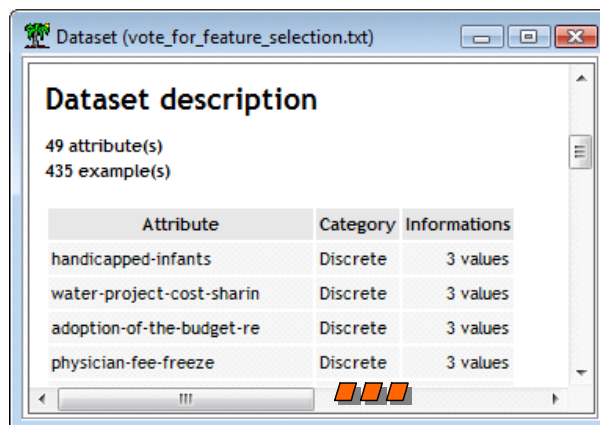
We create a new diagram by clicking on the FILE / NEW menu. We select the data file "vote\_for\_feature\_selection.txt".

<sup>5</sup> <http://data-mining-tutorials.blogspot.com/2010/07/naive-bayes-classifier-for-discrete.html>

<sup>6</sup> <http://data-mining-tutorials.blogspot.com/2009/07/resampling-methods-for-error-estimation.html>



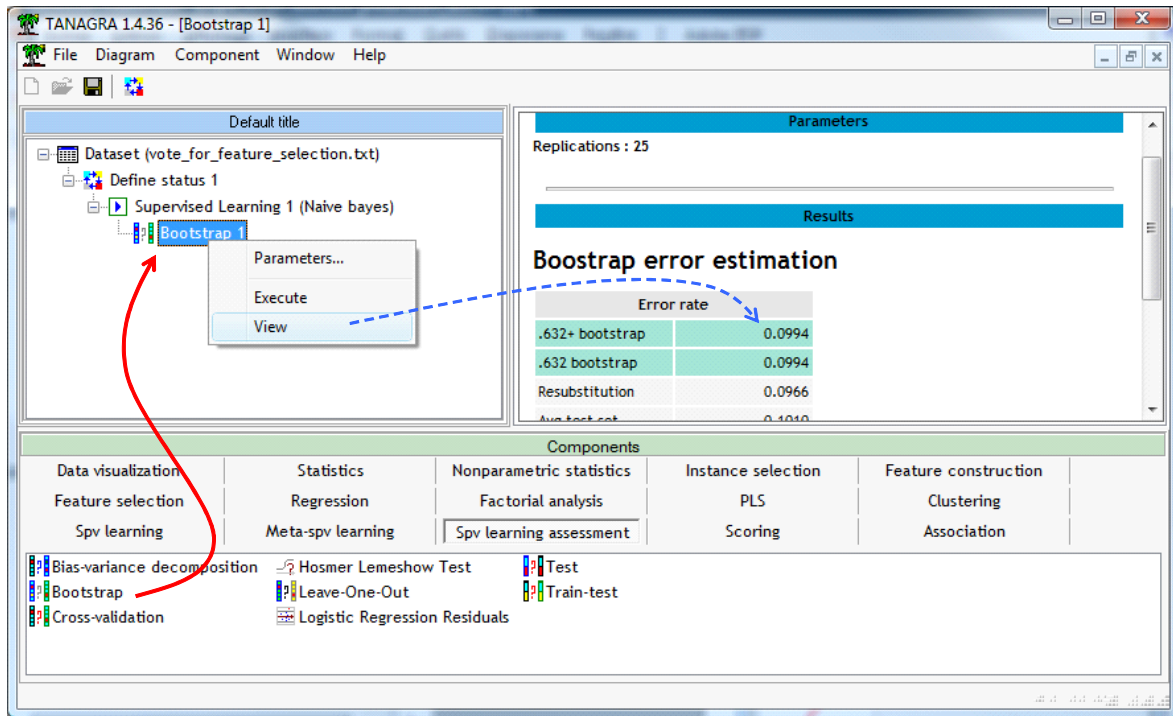
435 instances and 49 variables (1 target + 48 predictors) are available.



#### 4.1.2 Learning phase

We want to learn the naïve bayes classifier using all the predictors. The DEFINE STATUS component enables us to specify the target attribute and the predictors (input attributes).



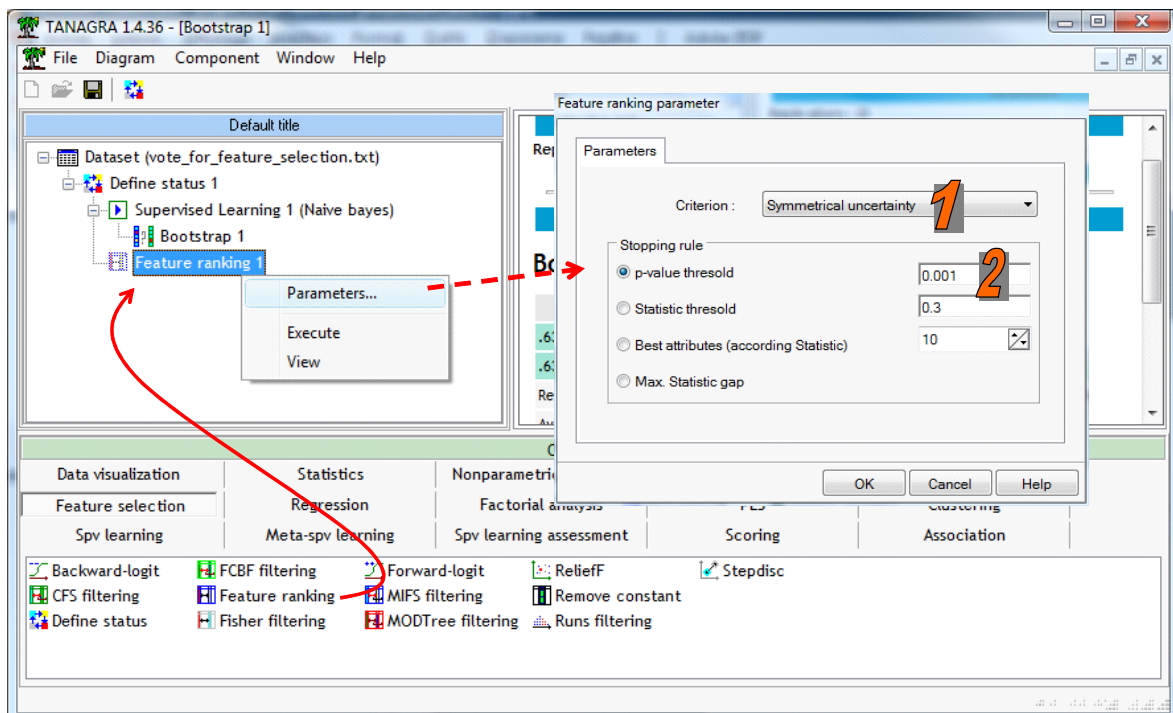


The "true" generalization error rate is 9.94%. This is the reference used in our study. We try to remove the irrelevant and redundant variables without decreasing the performance of the resulting classifier i.e. the classifier learned from the selected variables.

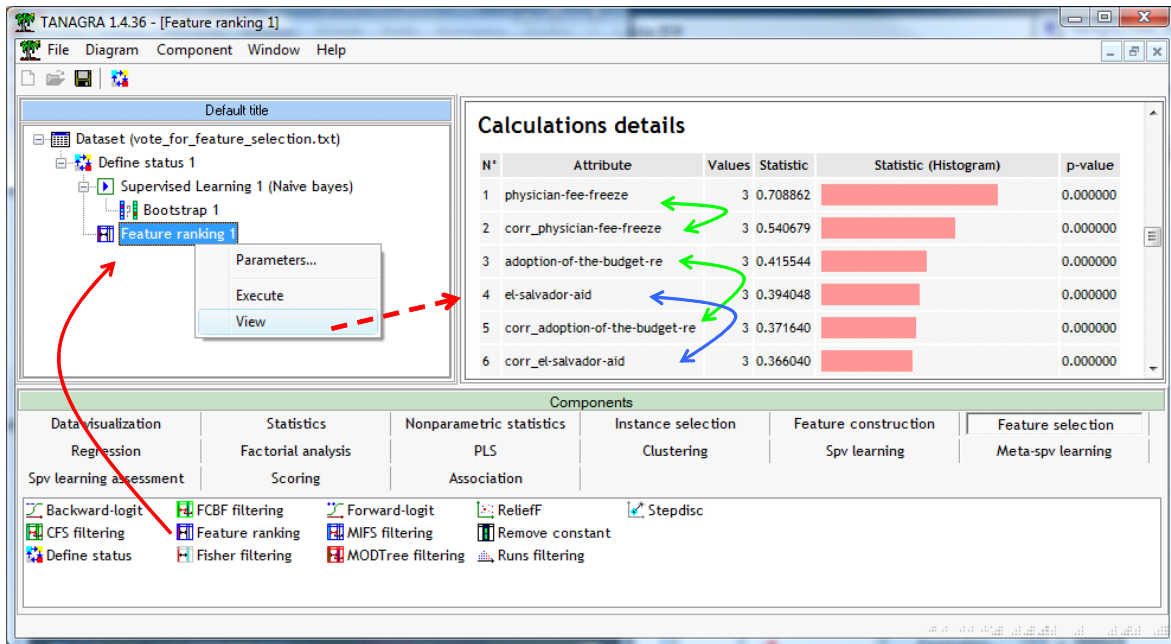
## 4.2 The behavior of the filtering approaches

### 4.2.1 "Ranking" approach

We add the FEATURE RANKING (FEATURE SELECTION tab) into the diagram. We use the symmetrical uncertainty in order to rank the classifier. Only the predictors which are significantly correlated to the target attribute are selected ( $\alpha = 0.001$ ).

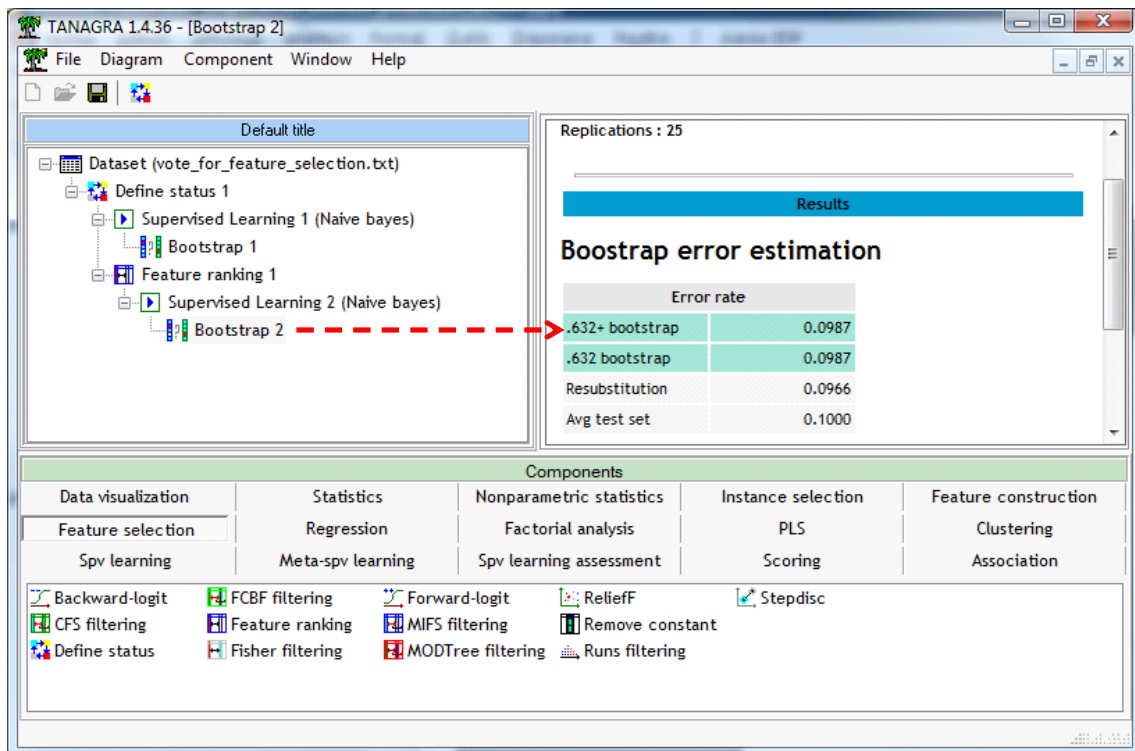


We click on the VEW menu. The ranking method highlights 28 predictors. The original and the correlated attributes come together. The approach does not handle the redundancy problem. But, none noisy attribute are inserted into the selected subset. The relevance is rightly treated.



Another problem is the setting of the algorithm. The usual values for alpha (5%, 10%, ...) in the hypothesis testing framework are not adapted.

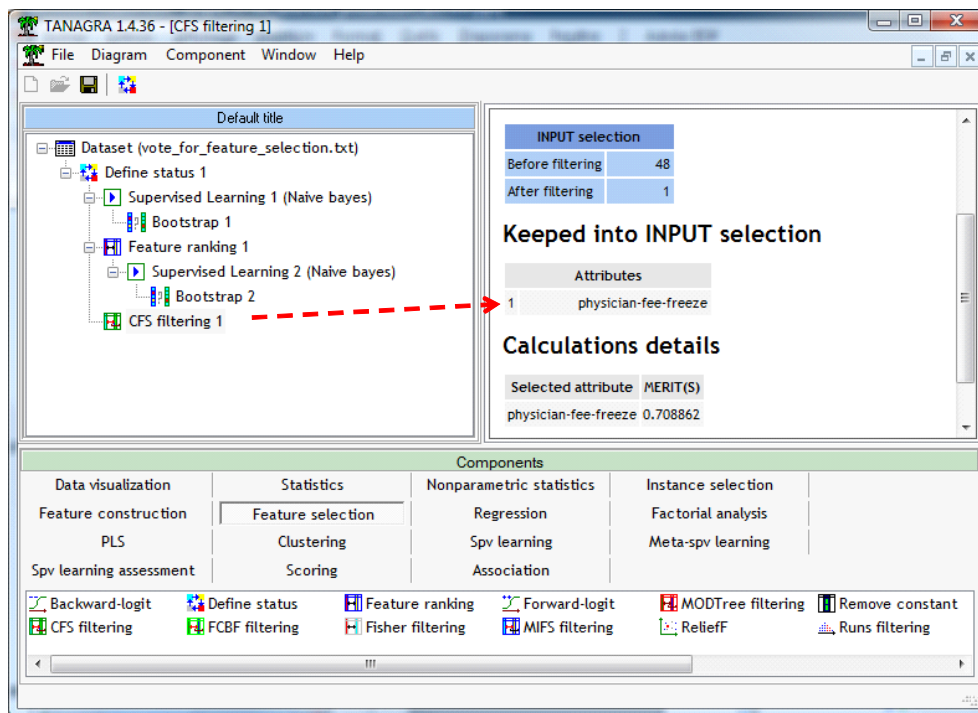
We add again the NAÏVE BAYES and the BOOTSTRAP. The bootstrap error rate of the model computed from the selected attribute is 9.87%. **Note:** at each step of the bootstrap, the whole path is executed i.e. we perform a feature ranking, and we learn the model from the selected attribute.



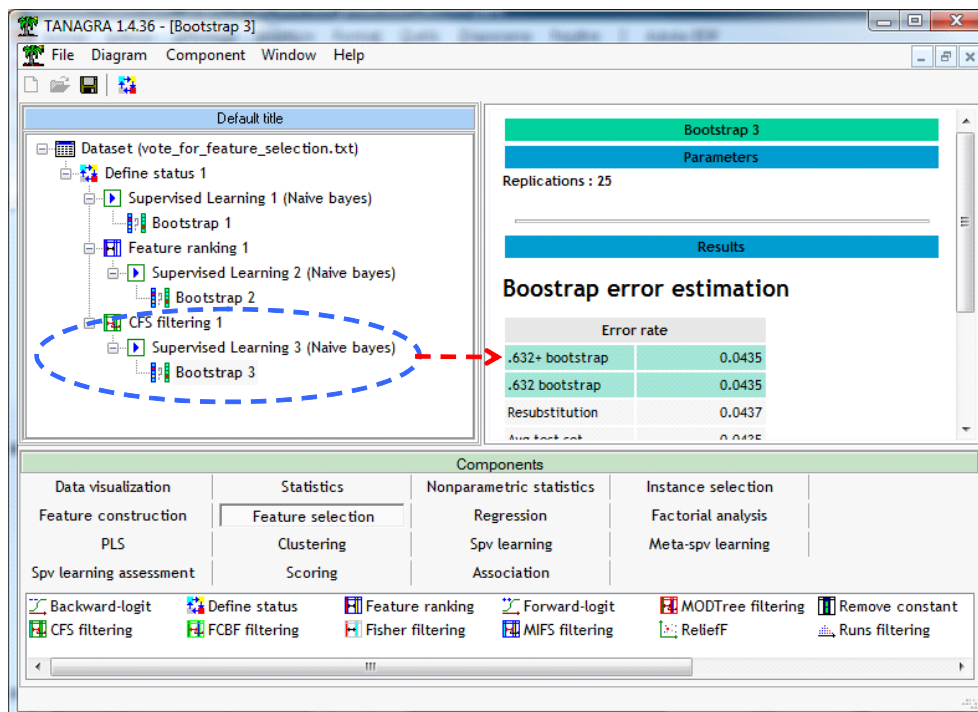
We will replicate the same experimental approach for the other selection techniques.

#### 4.2.2 CFS method

We insert the CFS FILTERING component (FEATURE SELECTION tab) into the diagram. We click on the VIEW menu.



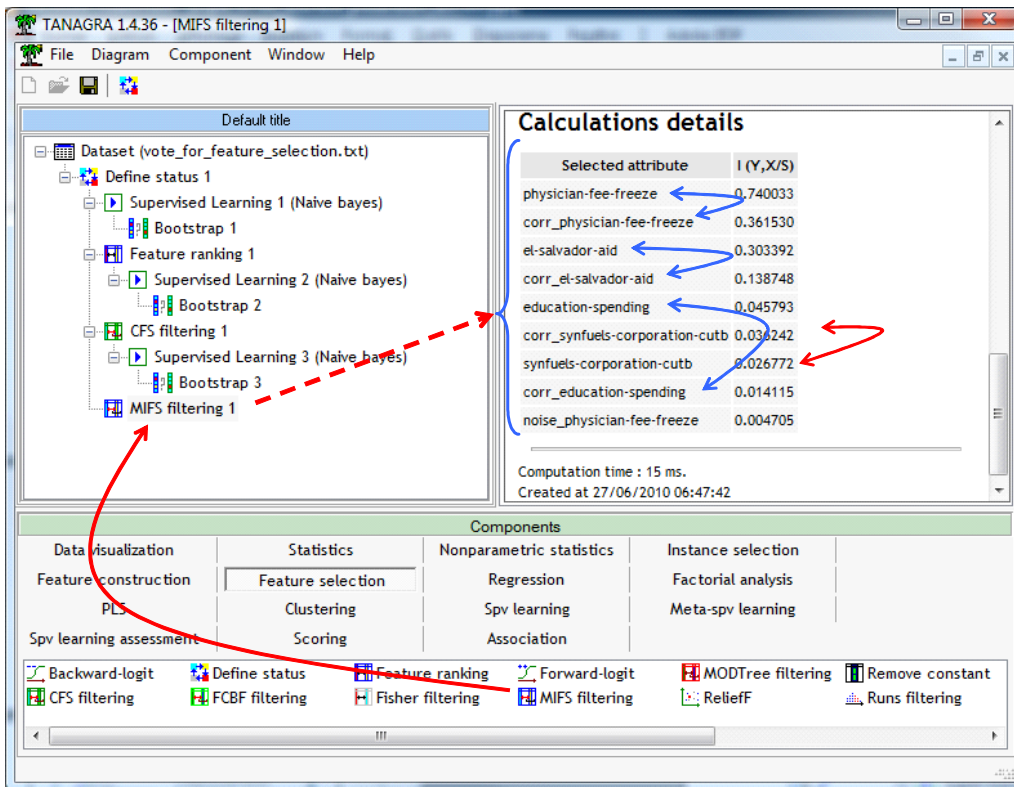
Only the « physician-fee-freeze » is selected (MERIT criterion = 0.709). Both “corr” and “noise” attributes are rejected.



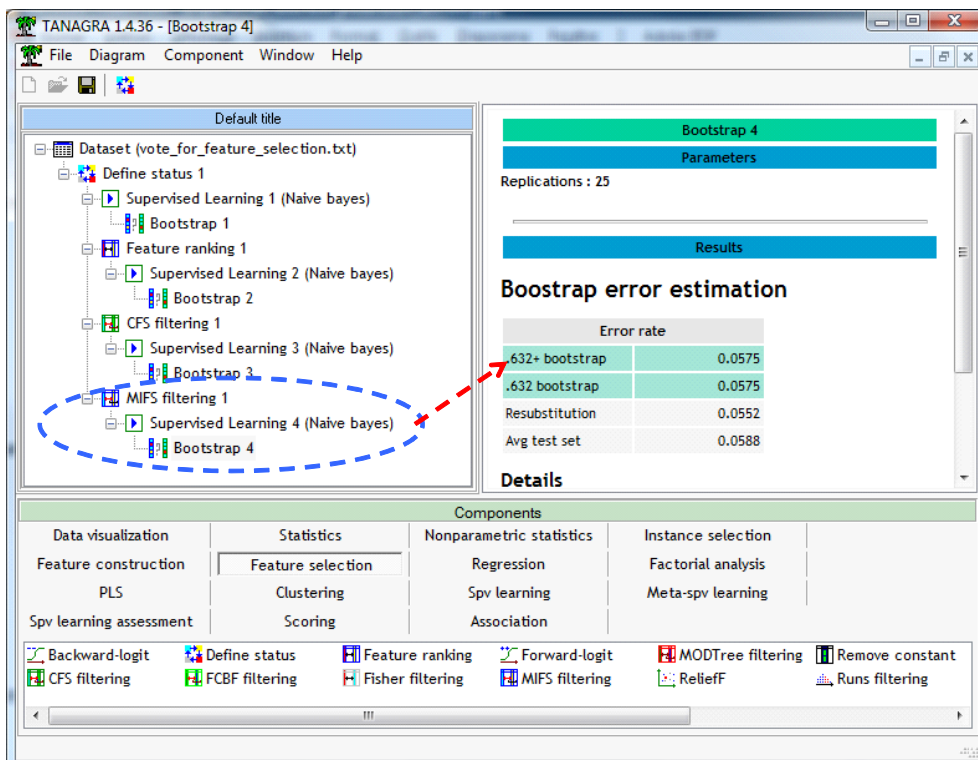
The bootstrap error rate is 4.35%. The method has drastically reduced the number of predictors and, in the same time, the performance of the classifier is considerably improved. This is the ideal scheme in a variable selection process.

### 4.2.3 MIFS method

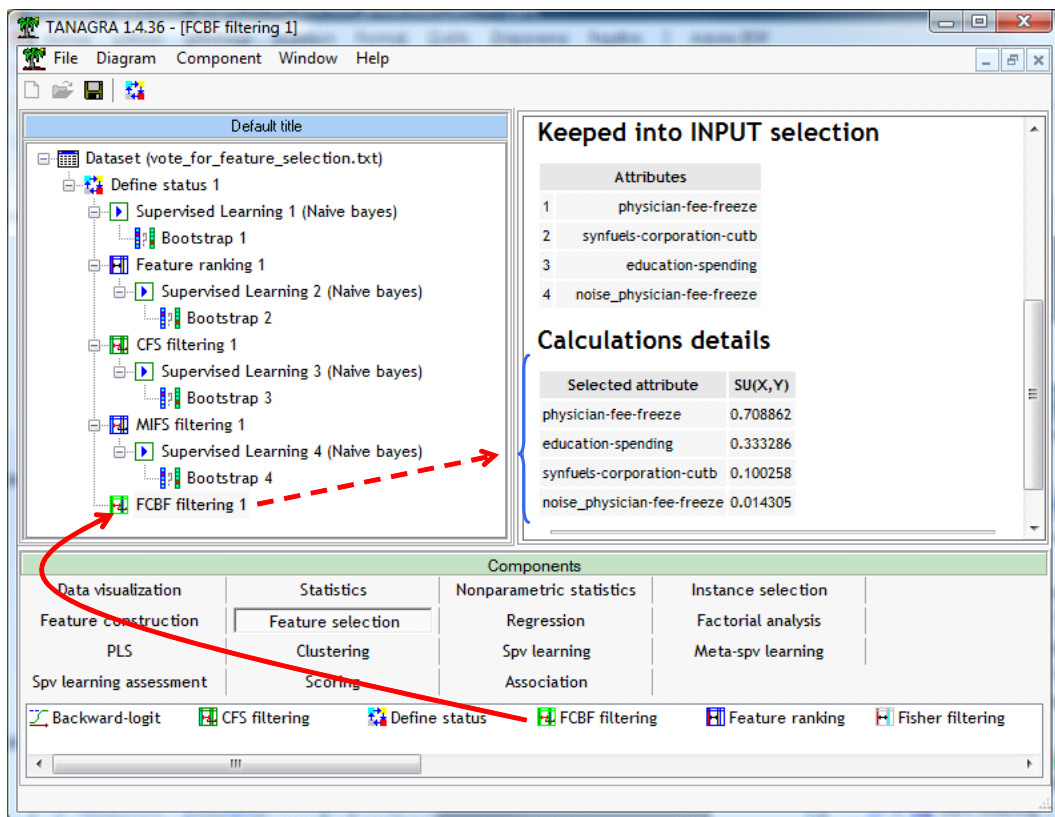
We apply the same framework for the MIFS approach, with the default settings ( $\beta = 1.5$ ).



Nine (9) predictors are selected. Some correlated attributes are inserted between the original predictors. This is really disappointing. Clearly, the value of  $\beta$  is inadequate. We must increase it to remove the correlated attributes. But we have not reference to set the adequate value. We must proceed by trial and error. The bootstrap error rate of the resulting classifier is 5.75%.

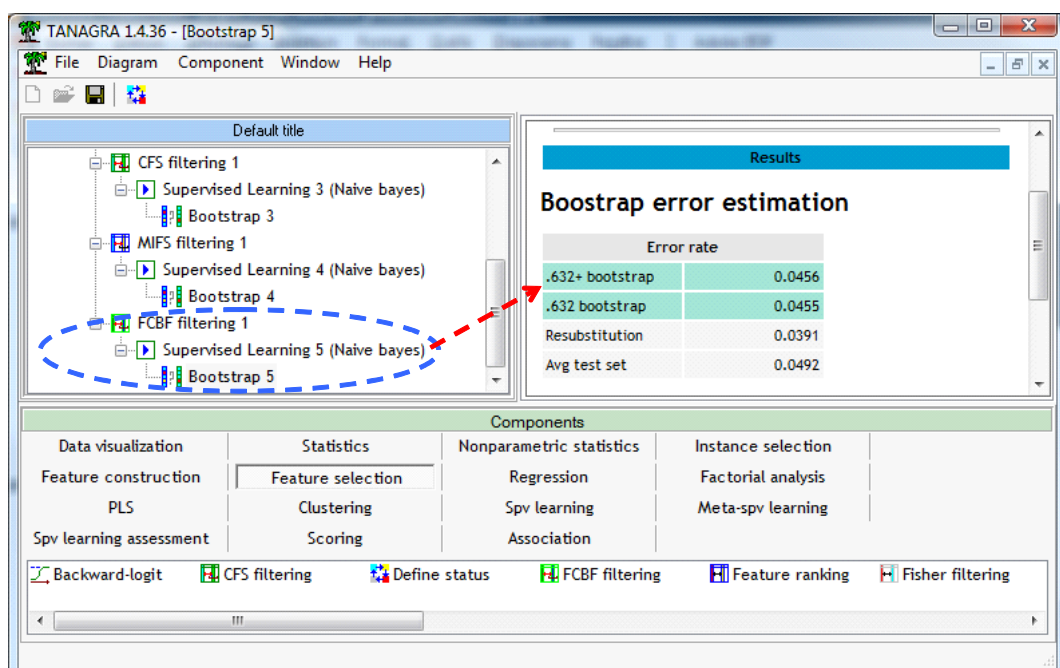


4.2.4 FCBF method



We set  $\delta = 0$  (default value) for FCBF. Four (4) predictors are selected. We note that a noisy variable is incorporated into the subset at the last position. Unlike to MIFS, we know how to set the parameter to avoid undesirable situation. If we set  $\delta = 0.1$ , the noisy variable "noise physician fee freeze" is not selected ( $\rho = 0.014305 < \delta = 0.1$ ).

The bootstrap error rate of the resulting naïve bayes classifier is 4.56%.



#### 4.2.5 MODTREE method

The MODTREE method selects 3 predictors, from the original predictors (no noisy or correlated variables).

INPUT selection	
Before filtering	48
After filtering	3

**Kept into INPUT selection**

Attributes	
1	adoption-of-the-budget-re
2	physician-fee-freeze
3	education-spending

**Calculations details**

Selected attribute	r (Y,X/S)	R2	Adj R2
physician-fee-freeze	0.809710	0.6556	0.6556
adoption-of-the-budget-re	0.231616	0.6741	0.6726
education-spending	0.125050	0.6792	0.6770

**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	

Fisher filtering   Forward-logit   MIFS filtering   **MODTree filtering**   ReliefF   Remove constant

We have successively the  $n^{\text{th}}$  order partial correlation ( $n = 0, 1, 2$ ):

$$r_{\text{group}, \text{physician-fee-freeze}} = 0.809710$$

$$r_{\text{group}, \text{adoption-of-budget} / \text{physician-fee-freeze}} = 0.231616$$

$$r_{\text{group}, \text{education-spending} / \text{physician-fee-freeze}, \text{adoption-of-budget}} = 0.125050$$

We have the values computed into the section 2.3.4.

Into the result table of Tanagra, we can see also the global coefficient of determination  $R_m^2$  computed from the partial correlation  $r_{y, x^* / z_1 \dots z_m} = r_m$  using the following formula

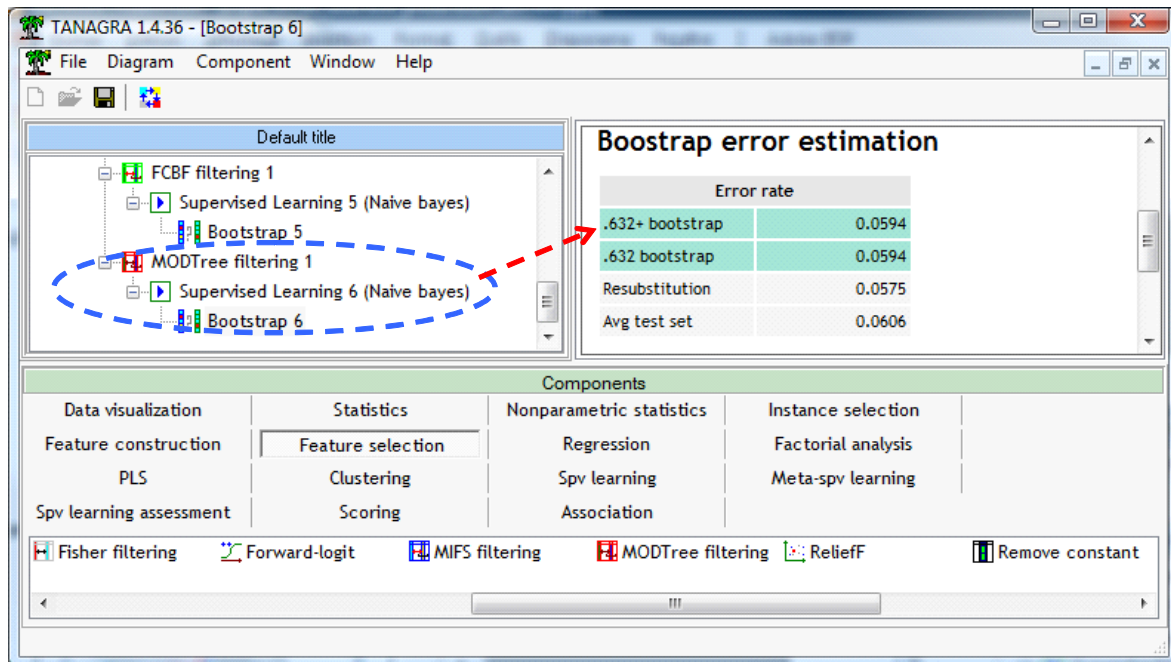
$$R_m^2 = 1 - \prod_{j=1}^m (1 - r_m^2)$$

The adjusted coefficient of determination  $\bar{R}_m^2$  is obtained from

$$\bar{R}_m^2 = 1 - \frac{n-1}{n-m-1} (1 - R_m^2)$$

We can understand easily that the MODTREE approach tries to maximize the adjusted coefficient of determination. During the forward search process, the stopping rule based on the partial correlation (section 2.3.4) has the same behavior than a stopping rule based on the decreasing of the adjusted coefficient of determination ( $\bar{R}_m^2$ ).

The bootstrap error rate of the resulting model is 5.94%.



### 4.3 Summary

We summarize the results in the following table.

Method	#Var. selected	#Var. « noise »	#Var. « corr »	Bootstrap error rate
All the variables	48	16	16	9.94%
Ranking ( $\alpha = 0.001$ )	28	0	14	9.87%
<b>CFS</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>4.35%</b>
MIFS ( $\beta = 1.5$ )	9	1	4	5.75%
FCBF ( $\delta = 0$ )	4	1	0	4.56%
MODTREE	3	0	0	5.94%

We note some trends, although it is obviously not possible to conclude from a single experiment.

- The feature reduction can increase the classifier performance. This is especially true if the classifier is easily disturbed by irrelevant attributes, such as the naive bayes classifier for instance. But surprisingly, some publications show that if the method embeds an internal

variable selection procedure (e.g. decision tree), it can get benefit also from the filtering algorithm.

- As expected, the ranking method can treat the relevance but not the redundancy. The predictors which are correlated with the original predictors, but no directly linked with the target attribute, are inserted into the selected subset. They disturb the interpretation of results.
- CFS, FCBF and MODTREE have similar behavior. They were able to identify good predictors for effective classification. At the same time, they removed rightly the noisy and the redundant variables.
- MIFS should have the same qualities. Its main drawback is that it is hard to parameterize (specifying the adequate value of  $\beta$ ).

We will discuss it in the conclusion. The same experimental design was conducted on other datasets, with very similar results.

## 5 Filtering approaches with other tools

In this section, we described the implementation of the filtering approaches with other tools.

### 5.1 Weka

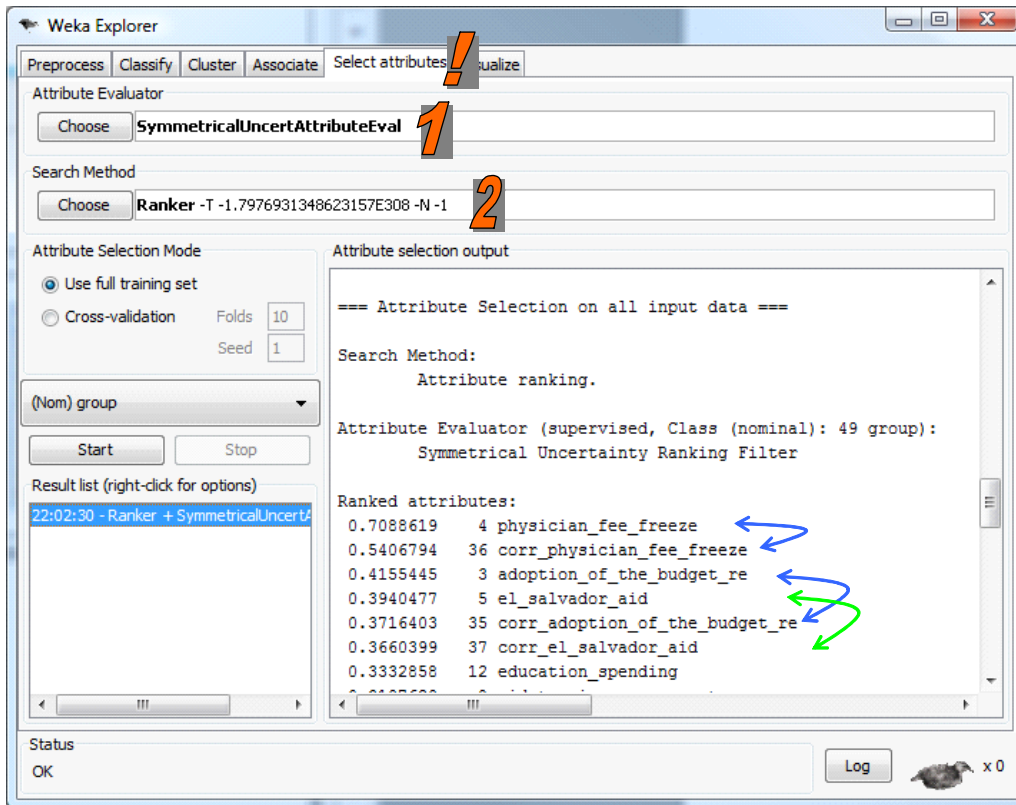
We use the EXPLORER mode. We load "vote\_for\_feature\_selection.arff" (OPEN FILE button).

The screenshot shows the Weka Explorer window with the 'SELECT ATTRIBUTES' tab selected. The 'Attributes' section lists 49 attributes, with 'group' selected. The 'Selected attribute' section shows 'handicapped\_infants' with a nominal type and a count of 236 for 'n', 12 for 'other', and 187 for 'y'. A bar chart visualizes these counts.

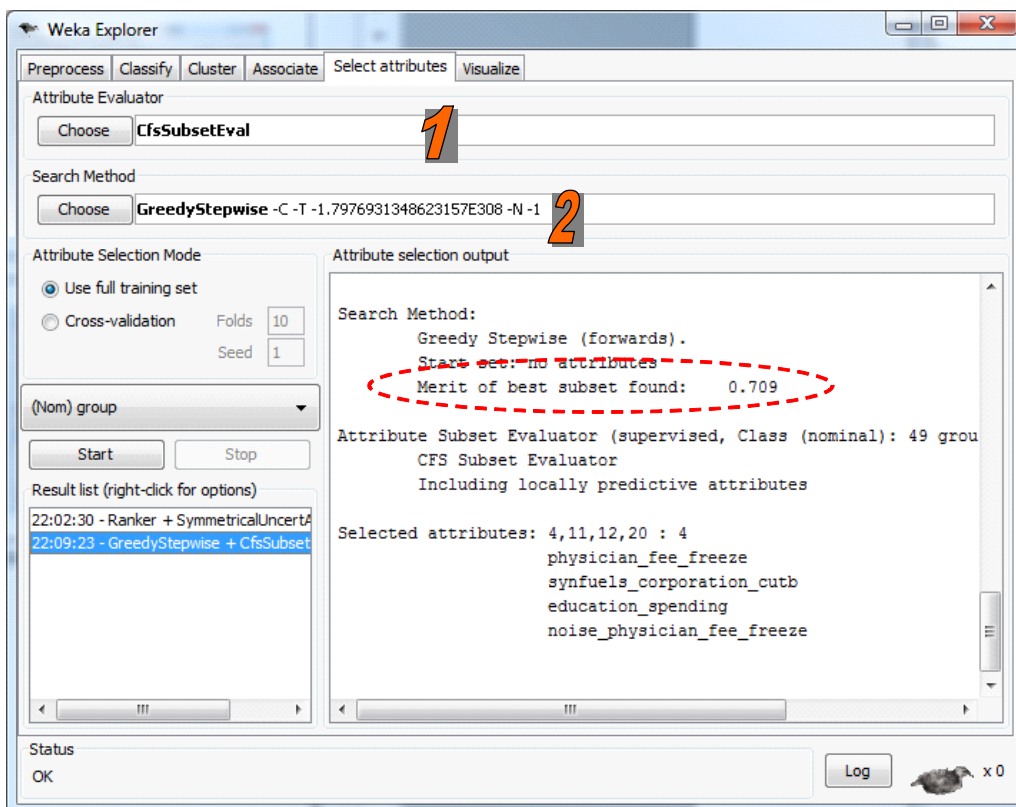
No.	Label	Count
1	n	236
2	other	12
3	y	187

We select the SELECT ATTRIBUTES tab. Many possibilities are available: ATTRIBUTE EVALUATOR enables to specify the measurement used; SEARCH METHOD specifies the search strategy. We set

the following parameters for the ranking approach. There is no stopping rule. The variables are simply ranked according the chosen criterion

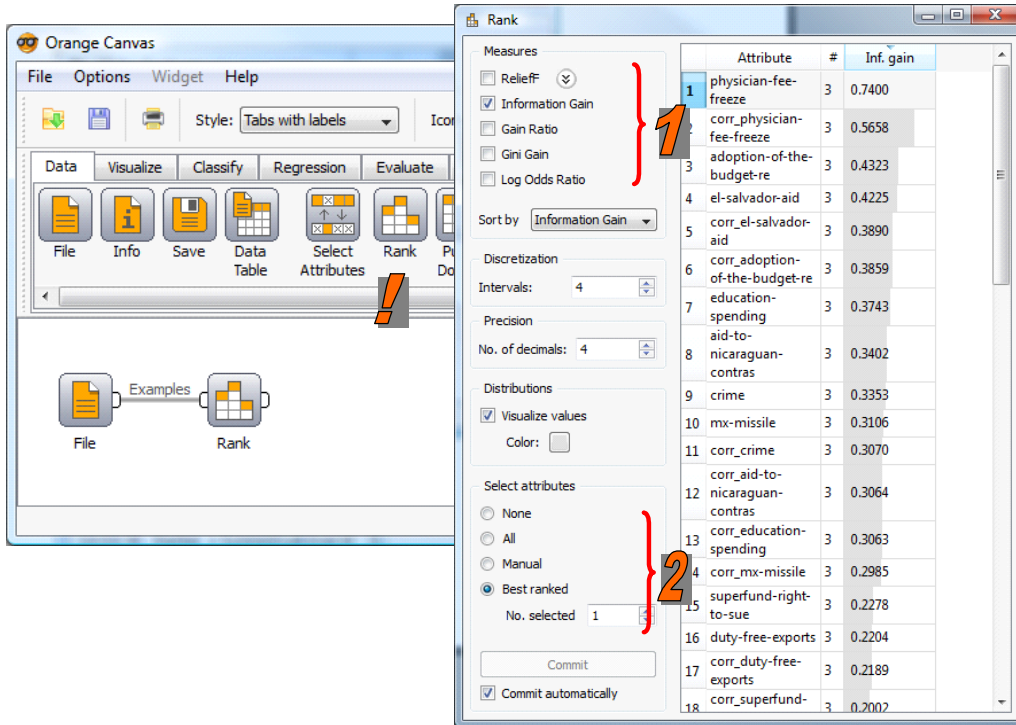


For the CFS approach, we specify the following settings and we obtain a subset of 4 variables.



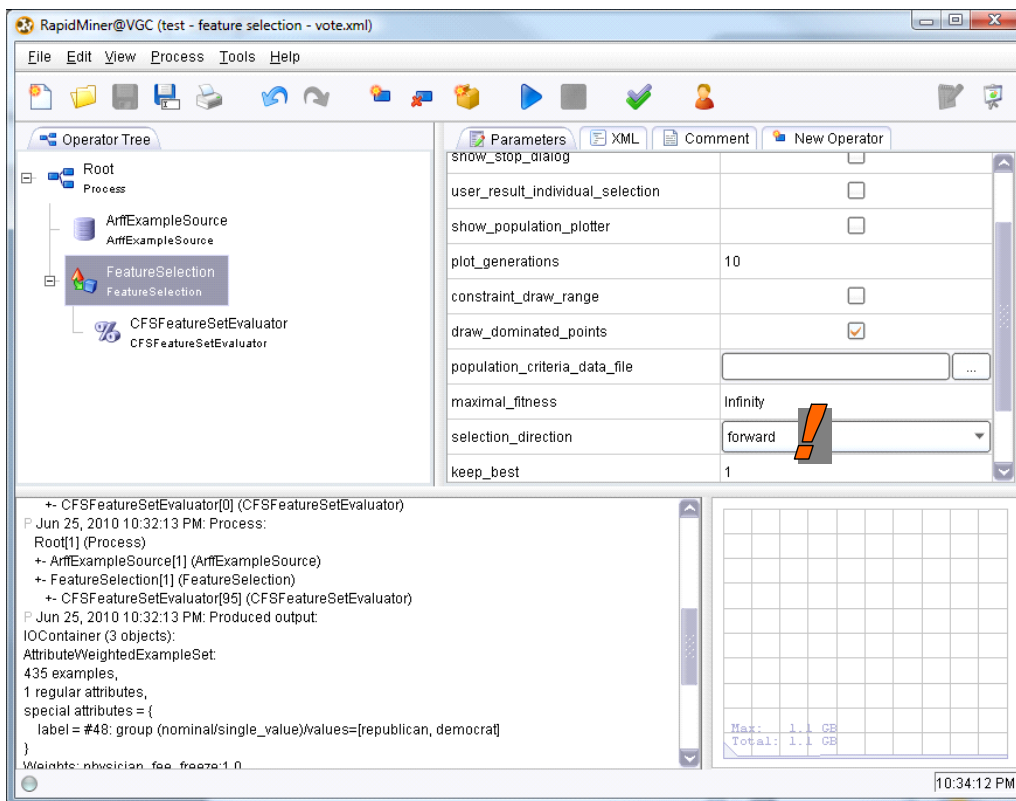
### 5.2 Orange

We use the RANK component for Orange. It simply ranks the attributes. It does not take into account the redundancy. Two parameters enable to guide the algorithm: (1) the measure of association used as ranking criterion; (2) the number of selected variables.

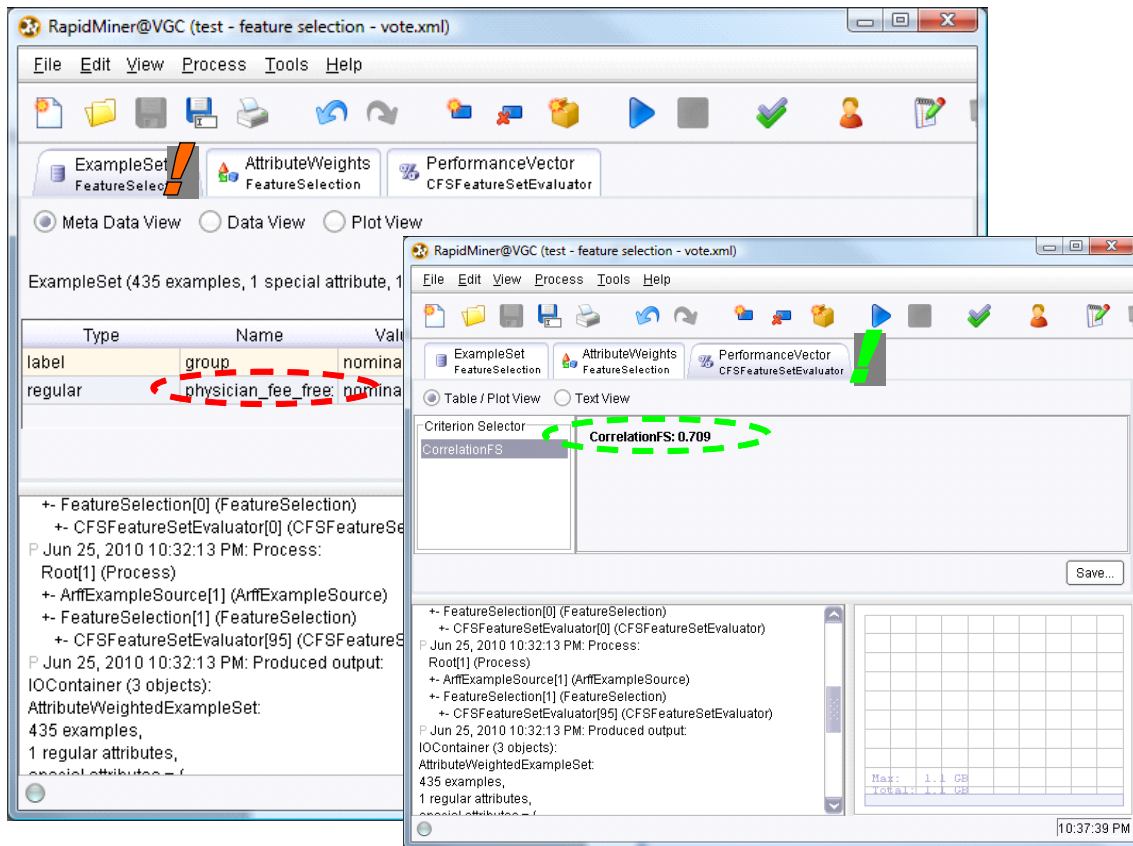


### 5.3 RapidMiner

We define the following diagram to implement the CFS approach.



RapidMiner provides a result identical to that of Tanagra. Only the variable "physician-fee-freeze" is selected, with a merit = 0.709.



#### 5.4 R – Package « [FSelector](#) »

We use the "[FSelector](#)" package with R. Here is the source code.

```
#clear the memory
rm (list=ls())

#load the dataset
vote.data <- read.table(file="vote_for_feature_selection.txt",header=T, sep="\t")

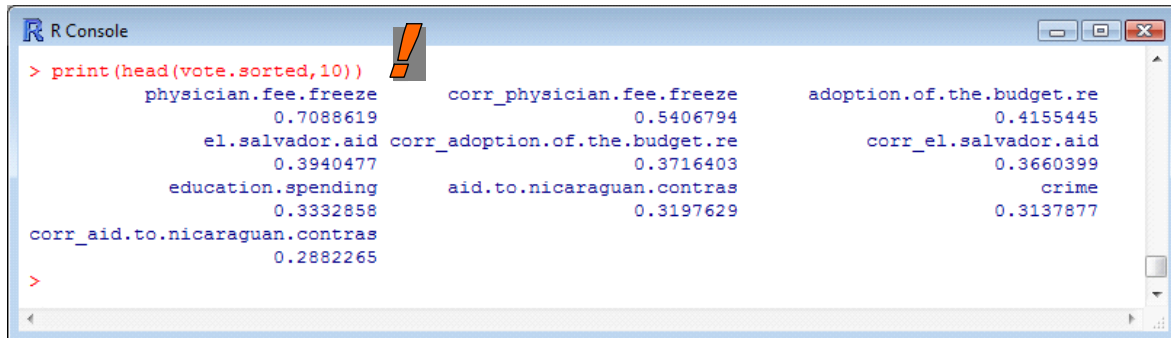
#loading the package
library(FSelector)

#*****
#ranking - Symmetrical uncertainty
#get the weight for each predictors
vote.ranking <- symmetrical.uncertainty(group ~ ., data = vote.data)
#sorting the result according the weight
index <- order(vote.ranking[[1]],decreasing=T)
vote.sorted <- vote.ranking[index,]
names(vote.sorted) <- rownames(vote.ranking)[index]
print(head(vote.sorted,10))

#*****
```

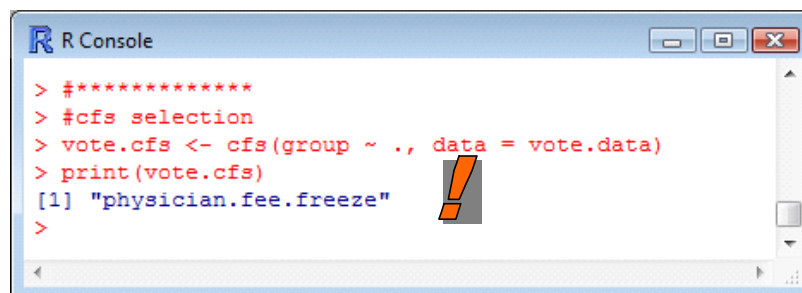
```
#cfs selection
vote.cfs <- cfs(group ~ ., data = vote.data)
print(vote.cfs)
```

The **symmetrical.uncertainty()** function calculates the association of each predictor with the target attribute. Then, we rank the variables according to this criterion.



```
> print(head(vote.sorted,10))
      physician.fee.freeze      corr_physician.fee.freeze      adoption.of.the.budget.re
      0.7088619              0.5406794              0.4155445
      el.salvador.aid corr_adoption.of.the.budget.re      corr_el.salvador.aid
      0.3940477              0.3716403              0.3660399
      education.spending      aid.to.nicaraguan.contras      crime
      0.3332858              0.3197629              0.3137877
      corr_aid.to.nicaraguan.contras
      0.2882265
```

For the **cfs()** function, we obtain only "physician-fee-freeze" (like Tanagra and RapidMiner).



```
> #*****
> #cfs selection
> vote.cfs <- cfs(group ~ ., data = vote.data)
> print(vote.cfs)
[1] "physician.fee.freeze"
```

## 6 Conclusion

We have described some filtering algorithms for discrete predictors in this tutorial. We have modified the characteristics of the datasets coming from the UCI server (<http://archive.ics.uci.edu/ml/>) in order to highlight the behavior of the various approaches.

We have led the same experimentations on other datasets such as IRIS, OPTIDIGITS, WEAVEFORM, KR-VS-KP or SPLICE<sup>7</sup>. We obtain very similar results.

<sup>7</sup> The continuous predictors are discretized with the MDLPC method -- <http://data-mining-tutorials.blogspot.com/2010/05/discretization-of-continuous-features.html>.