

1 Topic

Understanding the naive bayes classifier for discrete predictors.

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature (http://en.wikipedia.org/wiki/Naive_Bayes_classifier). Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in **Tanagra (version 1.4.36 and later)** a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier.

In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others.

In the second part, we use various tools on the same dataset ([Weka 3.6.0](#), [R 2.9.2](#), [Knime 2.1.1](#), [Orange 2.ob](#) and [RapidMiner 4.6.0](#)). We try above all to understand the obtained results.

2 Naïve bayes classifier

Let $\aleph = (X_1, \dots, X_J)$ the set of predictors, Y is the target attribute (with K values). In this tutorial, we assume that all the predictors are discrete¹. For one instance ω that we want to classify, the Bayesian rule consists in to maximize the class posterior probability i.e.

$$\hat{y}(\omega) = y_{k^*} \Leftrightarrow y_{k^*} = \arg \max_k P[Y = y_k / \aleph(\omega)]$$

The classification is based on a good estimation of the conditional probability $P(Y/X)$. It can be rewritten as follows.

¹ If we have some continuous predictors, we can discretize them – see <http://data-mining-tutorials.blogspot.com/2010/05/discretization-of-continuous-features.html>

$$P[Y = y_k / \aleph(\omega)] = \frac{P(Y = y_k) \times P[\aleph(\omega) / Y = y_k]}{P[\aleph(\omega)]}$$

Because we want to maximize this quantity according to y_k , and that the denominator of the formula does not depend on this one, we can use the following rule.

$$\hat{y}(\omega) = y_{k^*} \Leftrightarrow y_{k^*} = \arg \max_k P(Y = y_k) \times P[\aleph(\omega) / Y = y_k]$$

2.1 Conditional independence of predictors

The probability $P(Y = y_k)$ is easy to estimate from a sample. We compute the relative frequencies of the values. We use the "m probability estimate" which is more reliable, especially when we deal with a small dataset. Let n_k the number instances belonging to the class y_k into the learning sample, we have

$$\hat{P}(Y = y_k) = p_k = \frac{n_k + m}{n + m \times K}$$

If we set $m = 1$, we have the Laplace estimate.

The main difficulty is to give a reliable estimation of the conditional probability $P[\aleph(\omega) / Y = y_k]$. We often introduce some hypothesis to make the estimation tractable (e.g. linear discriminant analysis, logistic regression). In the case of the naive bayes classifier, we claim that the predictors are independent conditionally to the values of the target attribute i.e.

$$P[\aleph(\omega) / Y = y_k] = \prod_{j=1}^J P[X_j(\omega) / Y = y_k]$$

The number of parameters to estimate is dramatically lowered. For the predictor X which takes L values, we use

$$\hat{P}[X = l / Y = y_k] = p_{l/k} = \frac{n_{kl} + m}{n_k + m \times L}$$

n_{kl} is the number of instances belonging to $(X=l)$ and $(Y=y_k)$. We set usually $m = 1$ for the Laplace correction. In any case, $m > 0$ is necessary to avoid the problem resulting from $n_{kl} = 0$.

To simplify the calculations, we use instead the natural logarithms of the conditional probability formula. The classification rule becomes

$$\hat{y}(\omega) = y_{k^*} \Leftrightarrow y_{k^*} = \arg \max_k \left\{ \ln P(Y = y_k) + \sum_{j=1}^J \ln P[X_j(\omega) / Y = y_k] \right\}$$

In the follows, we name $d(y_k, \aleph)$ the classifications functions

$$d(y_k, \aleph) = \ln P(Y = y_k) + \sum_{j=1}^J \ln P[X_j(\omega) / Y = y_k]$$

2.2 Numerical example

2.2.1 Computing the conditional probabilities

We want to predict a heart disease from the characteristics of patients. The target attribute is DISEASE (positive, negative). The predictors are EXANG (yes or no) and CHEST_PAIN (asympt, atyp_angina, non_anginal, typ_angina).

We count the number of instances for each class.

Nombre de disease	
disease	Total
positive	104
negative	182
Total	286

We get the estimation of the prior probabilities ($m = 1$)

$$p_+ = \frac{104 + 1}{286 + 2} = 0.3646$$

$$p_- = \frac{182 + 1}{286 + 2} = 0.6354$$

We compute also the conditional probabilities of each value of the descriptors according to the values of the target attribute.

Nombre de disease	exang			Total
	yes	no		
positive	68	36		104
negative	19	163		182
Total	87	199		286

P(exang/disease)	
0.6509	0.3491
0.1087	0.8913

Nombre de disease	chest_pain				Total
	asympt	atyp_angina	non_anginal	typ_angina	
positive	81	8		11	104
negative	39	95	41	7	182
Total	120	103	52	11	286

P(chest_pain/disease)			
0.7593	0.0833	0.1111	0.0463
0.2151	0.5161	0.2258	0.0430

2.2.2 Classifying a new instance

We want to assign a class to a new instance ω with the following characteristics: EXANG = yes, CHEST_PAIN = asympt, we calculate the classifications functions:

$$d[+, \mathcal{N}(\omega)] = \ln 0.3646 + (\ln 0.6509 + \ln 0.7593)$$

$$= -1.7137$$

$$d[-, \mathcal{N}(\omega)] = \ln 0.6354 + (\ln 0.1087 + \ln 0.2151)$$

$$= -4.2095$$

Since $d(+, \mathcal{N}) > d(-, \mathcal{N})$, we assign the instance to the group (DISEASE = positive).

For a computer scientist, programming these calculations is very easy. This is perhaps for this reason that this method is as popular with researchers.

2.3 Why the naive bayes classifier is efficient?

2.3.1 The naive bayes classifier is a linear classifier

In spite of the unsophisticated assumption underlying of the naive bayes classifier, it is rather efficient in comparison with other approaches. Yet, it is not very popular with final users because people think that we cannot obtain an explicit model to classify a new instance. We cannot compare the weight of each predictor in the classification process. This opinion, widespread, is incorrect. We can obtain an explicit model, and this is a linear combination of the (binarized) predictors.

2.3.2 Classifier with 1 explanatory variable

First, we consider the situation where we have only one predictor X with L values $\{1, 2, \dots, L\}$. We create L binary attributes I_l

$$I_l(\omega) = \begin{cases} 1 & \text{si } X(\omega) = l \\ 0 & \text{sinon} \end{cases}$$

The classification function $d(y_k, \mathcal{S})$ for the class y_k is

$$\begin{aligned} d(y_k, \mathcal{S}) &= a_0 + \sum_{l=1}^L a_l \times I_l \\ &= \ln p_k + \sum_{l=1}^L \ln p_{l/k} \times I_l \end{aligned}$$

But, we know that

$$I_1 + \dots + I_L = 1 \Rightarrow I_L = 1 - (I_1 + \dots + I_{L-1})$$

We can rewrite the classification function:

$$\begin{aligned} d(y_k, \mathcal{S}) &= \ln p_k + \sum_{l=1}^L \ln p_{l/k} \times I_l \\ &= (\ln p_k + \ln p_{L/k}) + \sum_{l=1}^{L-1} \ln \frac{p_{l/k}}{p_{L/k}} \times I_l \\ &= a_{0,k} + a_{1,k} \times I_1 + a_{2,k} \times I_2 + \dots \end{aligned}$$

This is a linear classifier. We have the same representation bias as linear discriminant analysis or logistic regression. But the estimation of the parameters of the model is different.

2.3.3 Classifier with J discrete explanatory variables

Because we have an additive model, the generalization to several predictors is easy. The j -th variable X_j takes L_j values, we write the classification function as the follows

$$d(y_k, \mathcal{S}) = \left(\ln p_k + \sum_{j=1}^J \ln p_{L_j/k}^j \right) + \sum_{j=1}^J \sum_{l=1}^{L_j-1} \ln \frac{p_{l/k}^j}{p_{L_j/k}^j} \times I_l^j$$

Where $p_{l/k}^j = \hat{P}(X_j = l / Y = y_k)$, I_l^j is the binary attribute associated to the value l of X_j .

2.3.4 Numerical example

Let us consider the example with two predictors above. We set A the binary attribute for (exsang = yes); B1, B2 and B3, the binary attributes for (chest_pain = asympt), (chest_pain = atyp_angina) and (chest_pain = non_angina).

$$d[+, \mathfrak{N}] = \ln 0.3646 + \ln 0.3491 + \ln 0.0463 + \ln \frac{0.6509}{0.3491} \times A + \ln \frac{0.7593}{0.0463} \times B1 + \ln \frac{0.0833}{0.0463} \times B2 + \ln \frac{0.1111}{0.0463} \times B3$$

$$= -5.1342 + 0.6232 \times A + 2.7973 \times B1 + 0.5878 \times B2 + 0.8755 \times B3$$

$$d[-, \mathfrak{N}] = \ln 0.6534 + \ln 0.8913 + \ln 0.0430 + \ln \frac{0.1087}{0.8913} \times A + \ln \frac{0.2151}{0.0430} \times B1 + \ln \frac{0.5161}{0.0430} \times B2 + \ln \frac{0.2258}{0.0430} \times B3$$

$$= -3.7148 - 2.1041 \times A + 1.6094 \times B1 + 2.4849 \times B2 + 1.6582 \times B3$$

For the unlabeled instance (EXANG: yes, CHEST_PAIN: asympt), the values for the binary attributes are (A : 1 ; B1 : 1 ; B2 : 0 ; B3 : 0). By applying the classification functions,

$$d[+, \mathfrak{N}] = -5.1342 + 0.6232 \times 1 + 2.7973 \times 1 + 0.5878 \times 0 + 0.8755 \times 0 = -1.7337$$

$$d[-, \mathfrak{N}] = -3.7148 - 2.1041 \times 1 + 1.6094 \times 1 + 2.4849 \times 0 + 1.6582 \times 0 = -4.2095$$

The conclusion is consistent with the preceding calculation mode (section 2.2.2). But now, we have an explicit model that we can deploy easily. Moreover, by analyzing the parameter associated to each binary attribute, we can interpret the influence of each predictive variable over the target attribute. And yet, Tanagra (**version 1.4.36** and later) is one of the only tools which provides these parameters of the classification functions.

2.3.5 The case of the binary problem

When we deal with a binary target attribute $Y \in \{+, -\}$, we can simplify the classification functions and produce a single decision function $d(\mathfrak{N})$. For numerical example above, we obtain

$$d(\mathfrak{N}) = d(+, \mathfrak{N}) - d(-, \mathfrak{N})$$

$$= -1.4194 + 2.7273 \times A + 1.1878 \times B1 - 1.8971 \times B2 - 0.7828 \times B3$$

The decision rule becomes

$$\text{If } d[\mathfrak{N}(\omega)] > 0 \text{ then } \hat{y}(\omega) = + \text{ else } \hat{y}(\omega) = -$$

For the unlabeled instance (EXANG: yes, CHEST_PAIN: asympt), we obtain

$$d(\mathfrak{N}) = -1.4194 + 2.7273 \times 1 + 1.1878 \times 1 - 1.8971 \times 0 - 0.7828 \times 0$$

$$= 2.4958$$

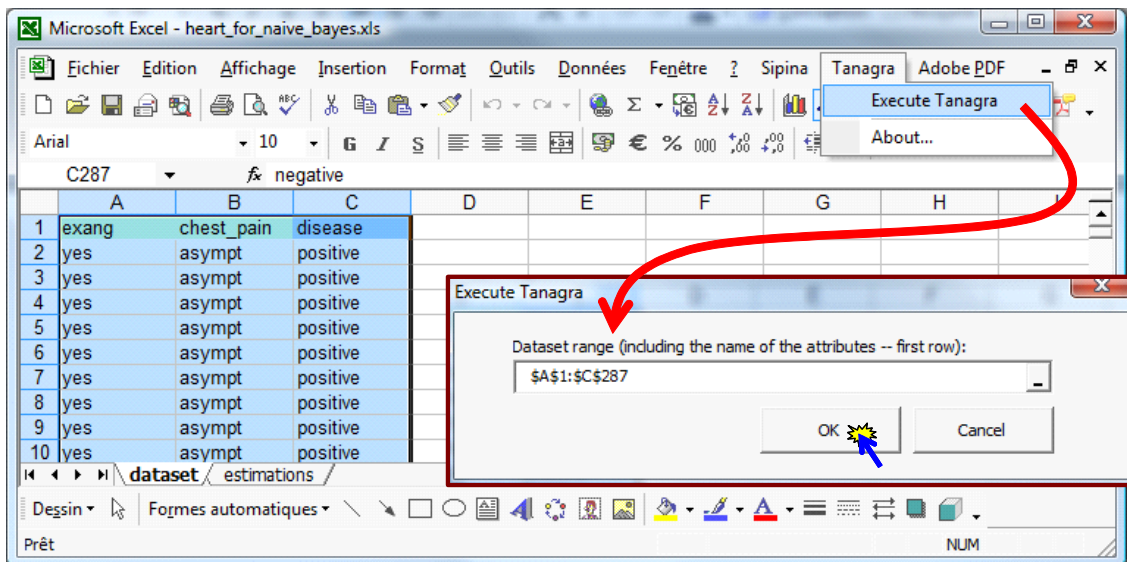
3 Dataset

In this tutorial, we use the "Heart Disease" dataset from the UCI server (Heart Disease Dataset - <http://archive.ics.uci.edu/ml/datasets/Heart+Disease>). After some data cleansing, we have 286 instances (http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/heart_for_naive_bayes.zip). We want to predict the values of DISEASE from EXANG (2 values) and CHEST_PAIN (4 values). The original dataset contains more predictors. But we prefer to use only these attributes in order to give more details about the calculations.

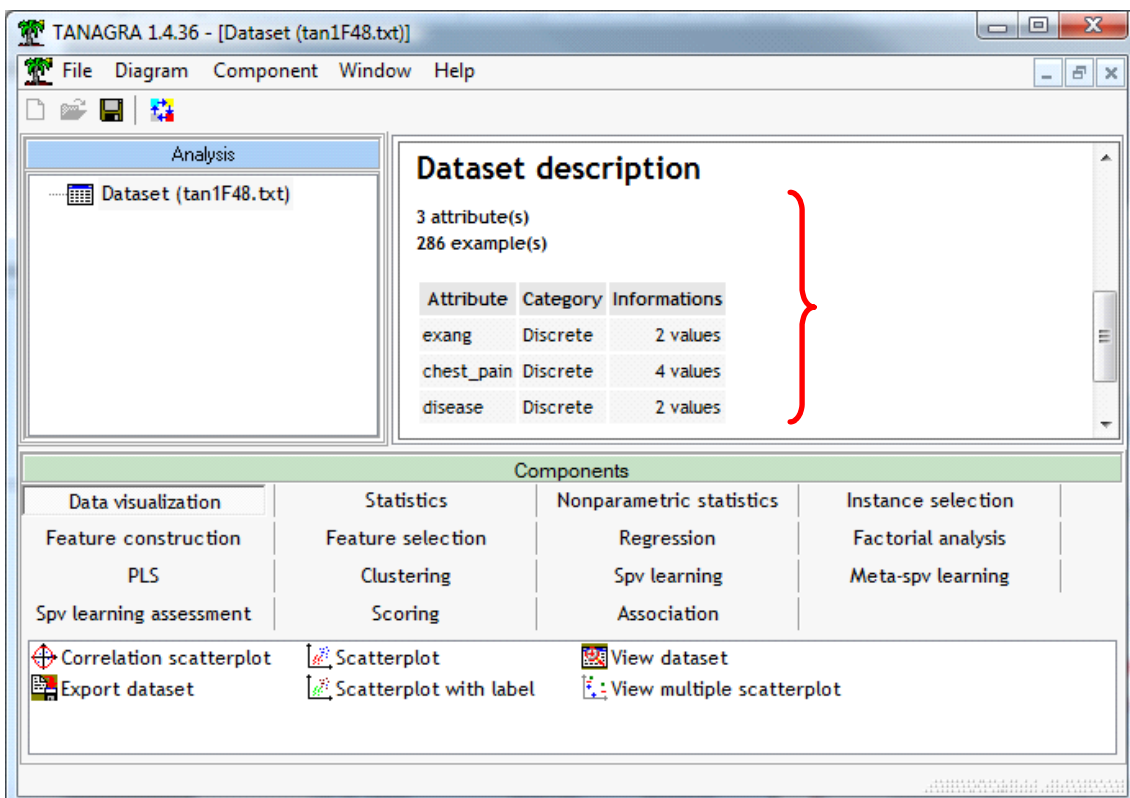
4 Analysis with Tanagra

4.1 The naive bayes classifier under Tanagra

Importing the dataset. To import the dataset into Tanagra, we open the data file into Excel spreadsheet. We select the data range. Then we click on the TANAGRA / EXECUTE menu².

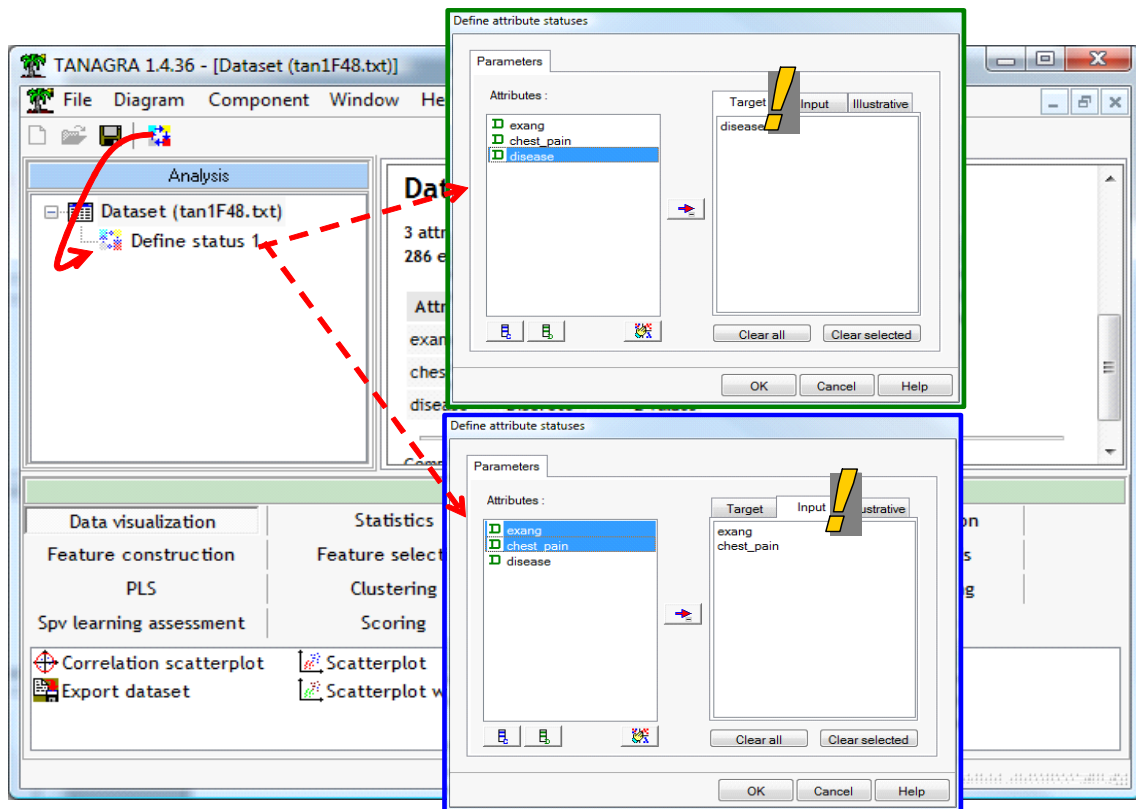


Tanagra is automatically launched. We have 286 instances and 3 discrete variables.

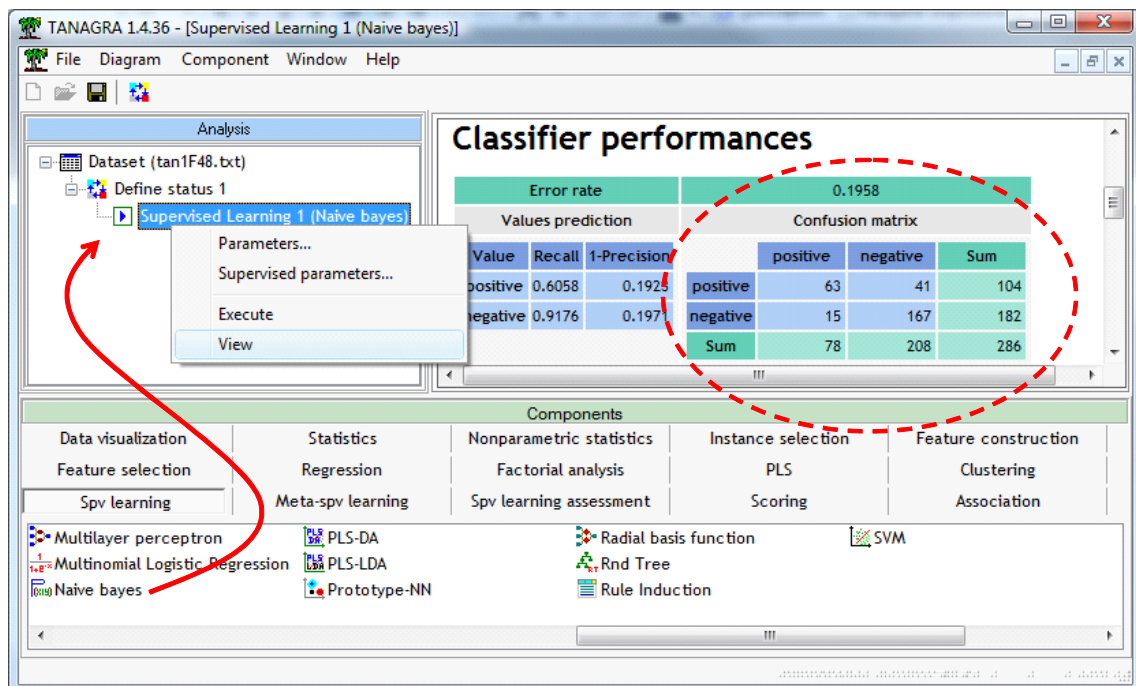


² About the TANAGRA menu into Excel, see <http://data-mining-tutorials.blogspot.com/2008/10/excel-file-handling-using-add-in.html>

Learning the model. Before the learning process, we must specify the target attribute (DISEASE) and the predictors (EXANG, CHEST_PAIN). We use the DEFINE STATUS component.

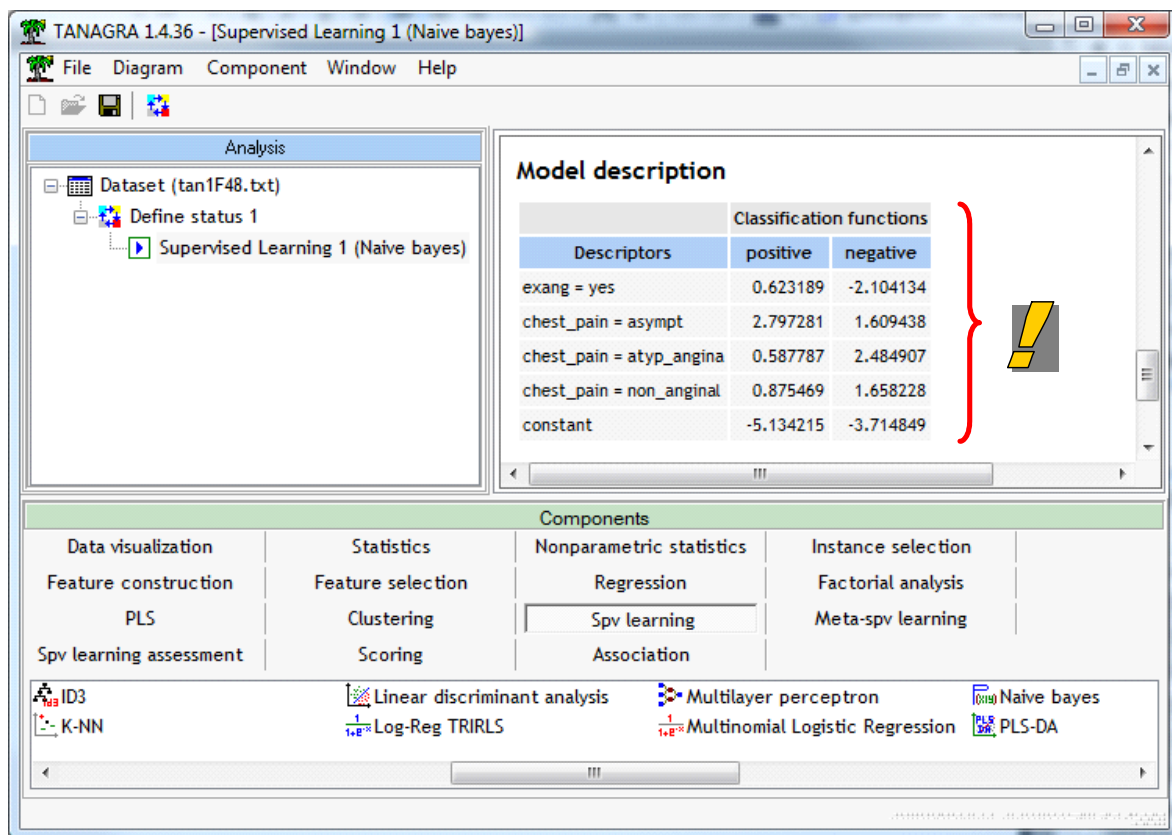


We can add the NAIVE BAYES CLASSIFIER component now (SPV LEARNING tab) into the diagram. We click on the VIEW menu to get the results.



Into the upper part of the visualization window, we get the confusion matrix and the error rate computed on the learning sample (resubstitution error rate).

The main new feature in the version 1.4.36 of Tanagra is in the lower part of the window. "**Model description**" provides the parameters of the classification functions. The deployment of the model outside the Tanagra context (e.g. into a spreadsheet) is greatly facilitated.



We obtain the parameters computed manually previously (section 2.3.4). Because we deal with a binary problem, we can extract the decision function.

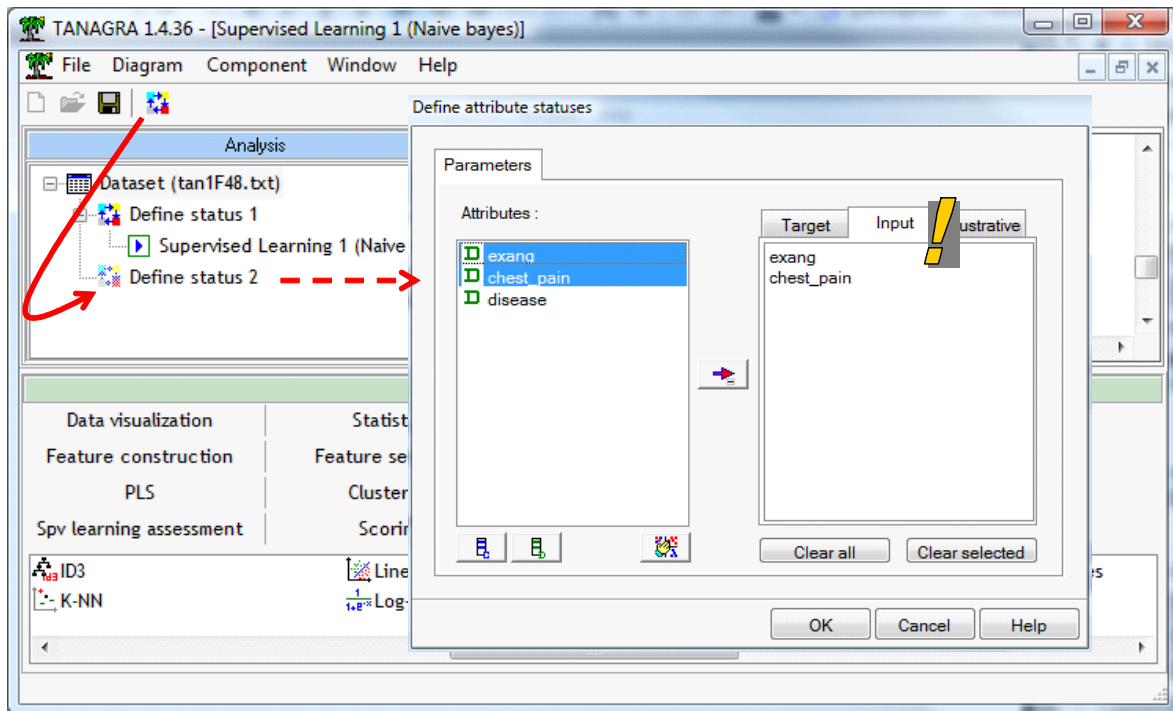
Descriptors	Classification functions		Decision function
	positive	negative	
exang = yes	0.623189	-2.104134	2.7273
chest_pain = asympt	2.797281	1.609438	1.1878
chest_pain = atyp_angina	0.587787	2.484907	-1.8971
chest_pain = non_anginal	0.875469	1.658228	-0.7828
constant	-5.134215	-3.714849	-1.4194

Note: As optional in Tanagra, we can get the details of cross tabulations used for the computation of the conditional probabilities.

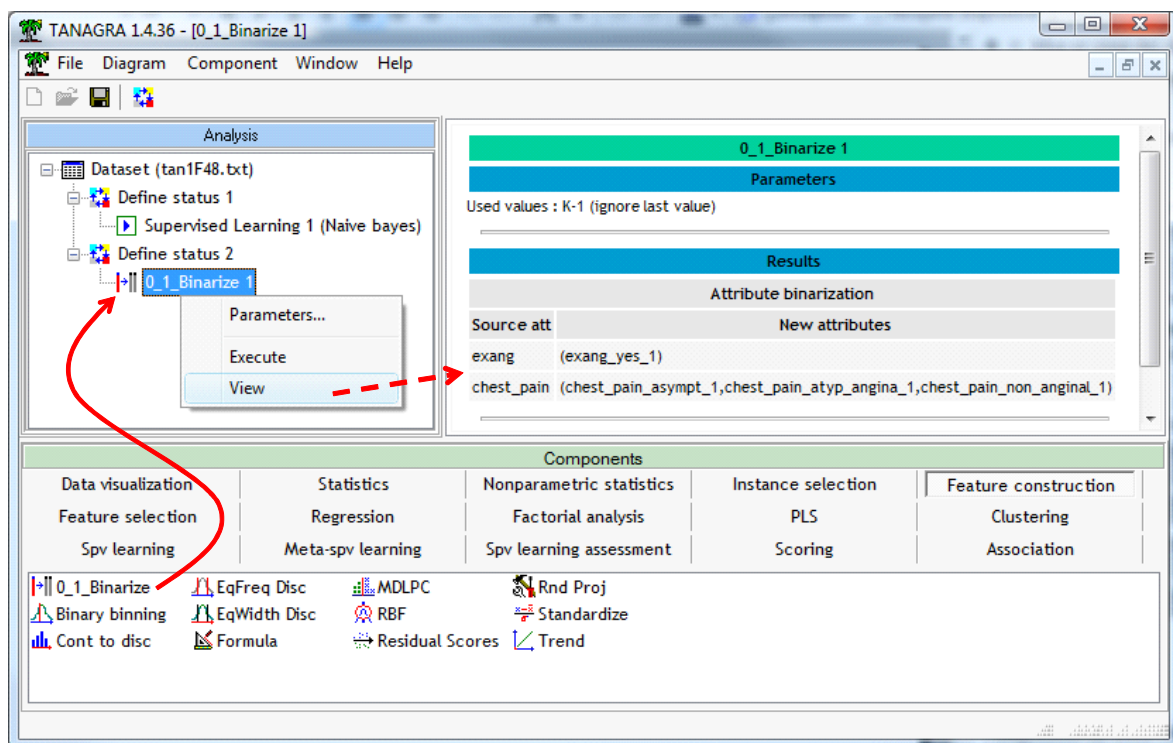
4.2 About the others linear classifiers

The methods analyzed in this section can provide a linear classification function. Our aim is to compare the parameters (coefficients) of the models.

We must explicitly recode the discrete predictors before. We add the DEFINE STATUS component at the root of the diagram. We set EXANG and CHEST_PAIN as INPUT.

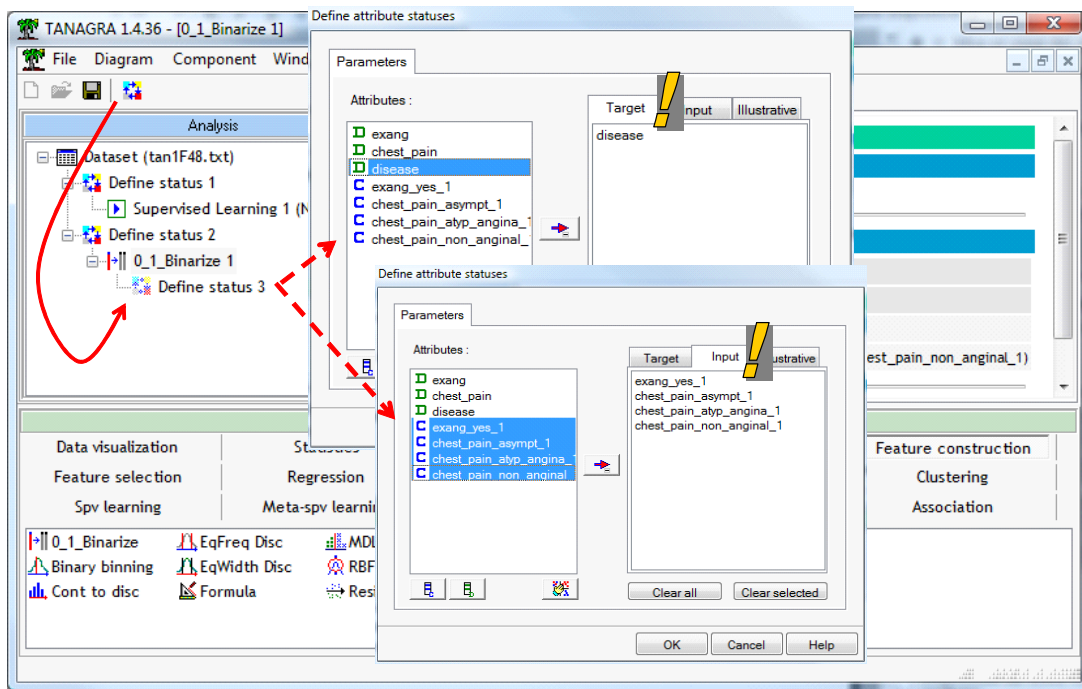


We insert the o_1_BINARIZE component (FEATURE CONSTRUCTION tab). We click on the VIEW menu.



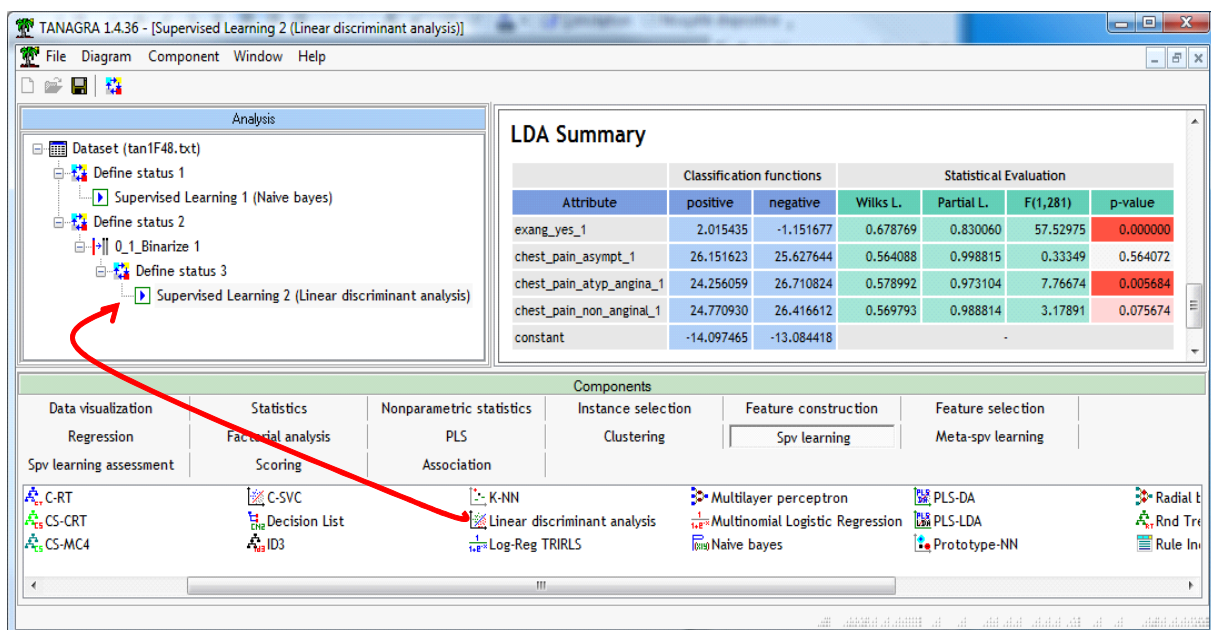
EXANG is transformed into one binary attribute 1 (yes) / 0 (no), CHEST_PAIN into three binary variables.

We insert again the DEFINE STATUS component for the learning phase. We set DISEASE as TARGET, the binary attributes as INPUT.



4.2.1 Linear discriminant analysis (LDA)

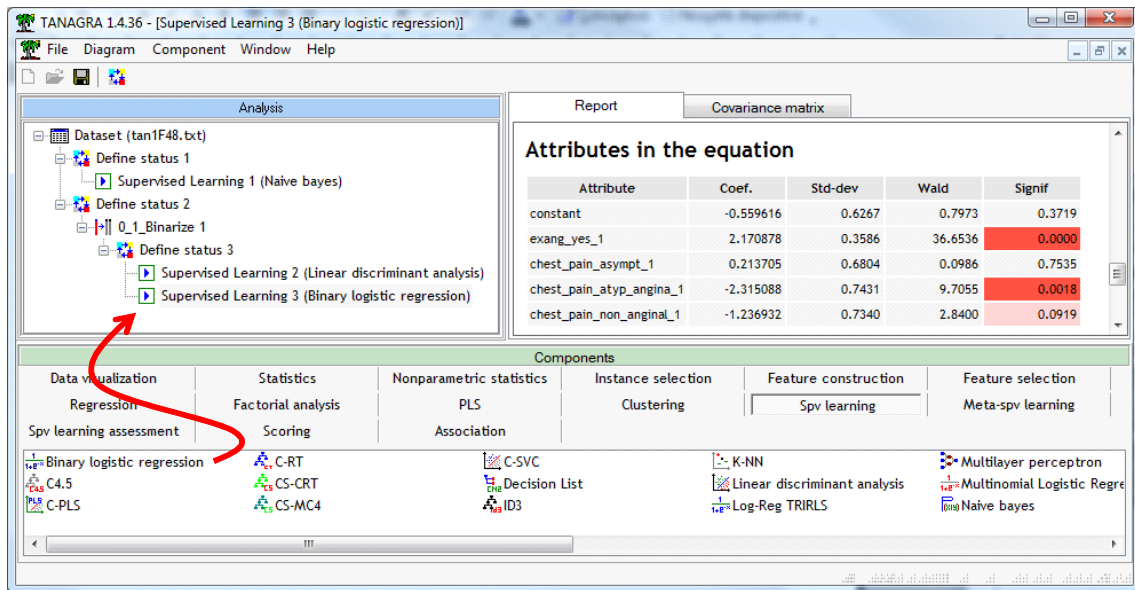
The linear discriminant analysis (LINEAR DISCRIMINANT ANALYSIS – SPV LEARNING tab) provides classification functions, one for each class. Various indicators enable to evaluate the global relevance of the model, and the relevance of each predictor.



We can derive the decision function from the classification functions for the binary problems.

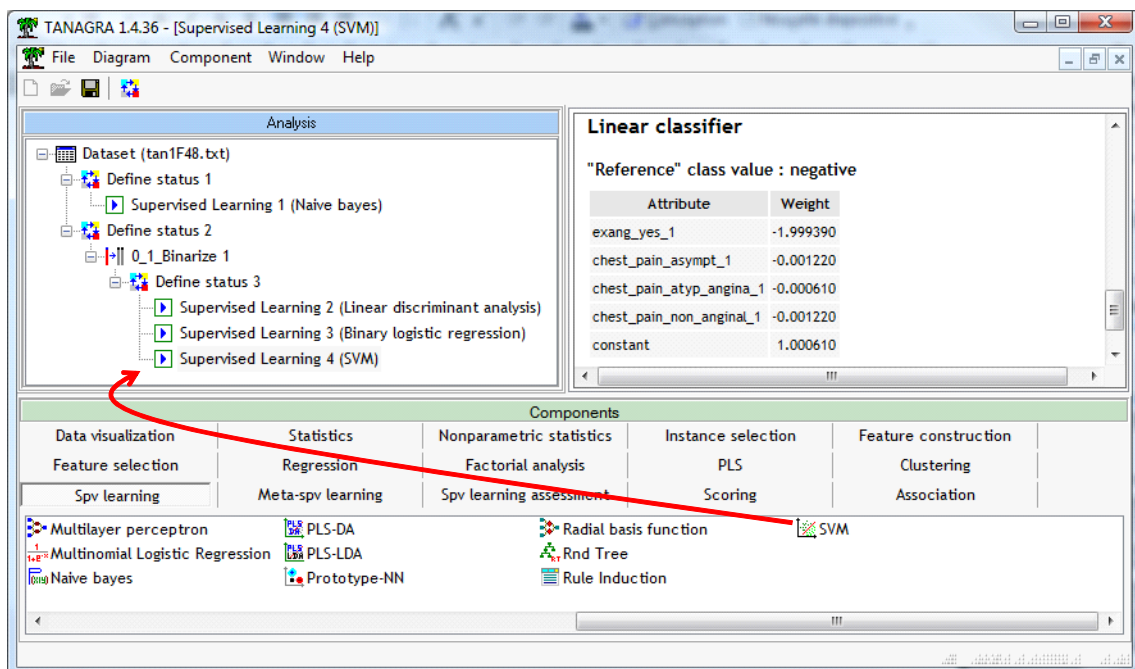
4.2.2 Logistic regression

The logistic regression (BINARY LOGISTIC REGRESSION – SPV LEARNING tab) is a method widely used by statisticians. Surprisingly, it is not really popular with the machine learning community. It provides directly the decision function. The sign of the parameters is computed according the reference group. This is the positive group here.



4.2.3 Linear SVM (Support Vector Machine)

The SVM component (SPV LEARNING tab) provides also the decision function.



The sign of the parameters is computed according to the reference group. This is the negative group here. Thus, we must multiply the coefficients by "-1" to obtain results comparable to the other approaches.

4.3 Comparing the decision functions

The representation bias is the same, but the learning bias is not the same. It is very interesting to compare the parameters obtained from the various approaches during the learning process.

Naive bayes, linear discriminant analysis and logistic regression are consistent if we consider the sign of the coefficients. But, their values are a little different. On the other hand, SVM approach provides very different coefficients. It seems also that (EXANG = YES) is the most relevant

predictor according to the absolute value of the coefficient. This is rather coherent with the results of the other approaches.

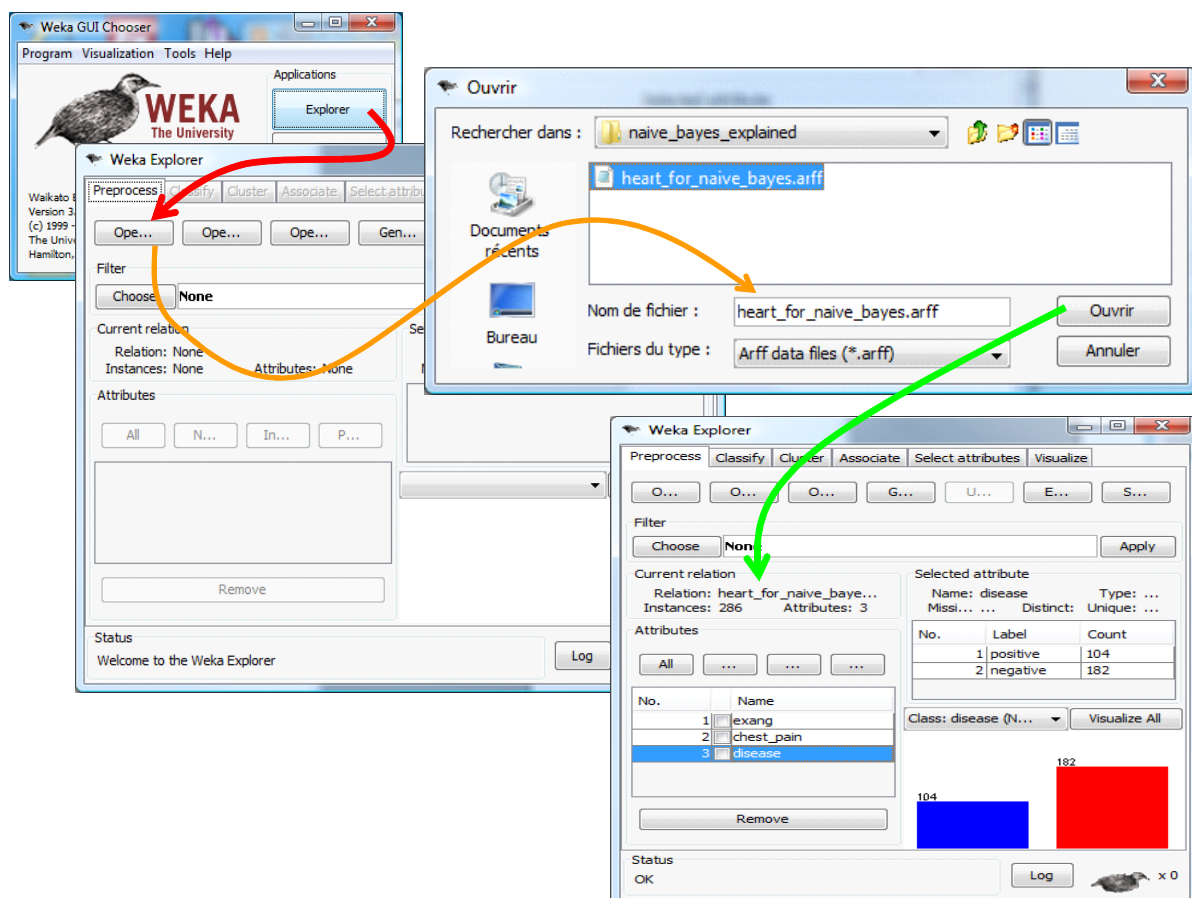
Descriptors	Naive Bayes	LDA	Logistic Reg.	Linear SVM
exang = yes	2.7273	3.1671	2.1709	1.9994
chest_pain = asympt	1.1878	0.5240	0.2137	0.0012
chest_pain = atyp_angina	-1.8971	-2.4548	-2.3151	0.0006
chest_pain = non_anginal	-0.7828	-1.6457	-1.2369	0.0012
constant	-1.4194	-1.0130	-0.5596	-1.0006

About the generalization error rate, on a very simple dataset such as we use in this tutorial, there are not a discrepancy. We use a bootstrap to estimate the error rate (<http://data-mining-tutorials.blogspot.com/2009/07/resampling-methods-for-error-estimation.html>). We obtain about 20% whatever the method used. These methods often present similar performance on real datasets, except for very specific configurations (e.g. large number of predictors in comparison of the number instances, etc.).

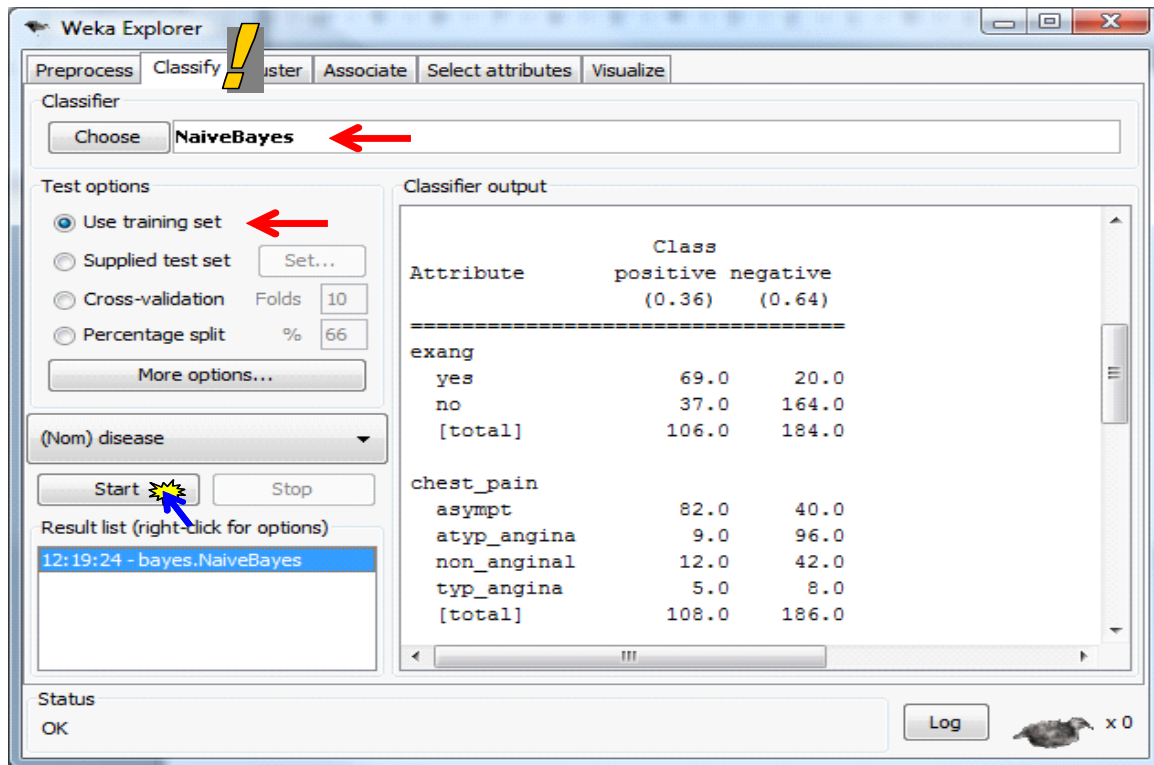
5 Analysis with the other free tools

5.1 Weka 3.6.0

We use Weka in the EXPLORER mode (<http://www.cs.waikato.ac.nz/ml/weka/>). We launch the tool, then we load the HEART_FOR_NAIVE_BAYES.ARFF data file (Weka file format).



We activate the CLASSIFY tab. We select the NAÏVE BAYES approach. The training sample is both used for the learning and the testing process. We click on the START button.



We obtain the same confusion matrix as Tanagra. Into the middle part of the visualization window, Weka provides the cross tabulations used for the computation of the various conditional probabilities. We have no information however about the classification functions.

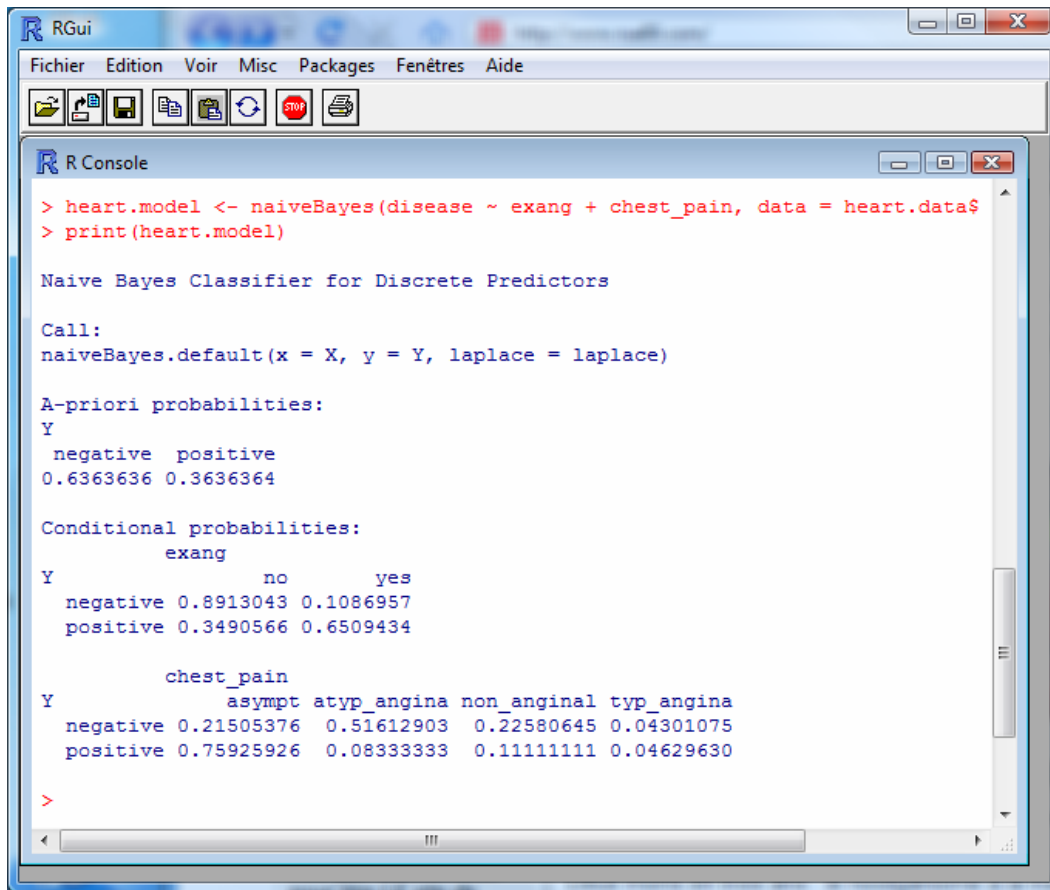
5.2 R 2.9.2

We use the e1071 package (<http://cran.r-project.org/web/packages/e1071/index.html>) under R (<http://www.r-project.org/>). The source code of our program is the following.

```
#clear the memory
rm (list=ls())
#load the dataset
heart.data <- read.table(file="heart_for_naive_bayes.txt",sep="\t",header=T)
#build the model
library(e1071)
heart.model <- naiveBayes(disease ~ exang + chest_pain, data = heart.data, laplace = 1.0)
print(heart.model)
```

The option "laplace = 1" corresponds to "m = 1" for the calculations of the conditional probabilities.

R provides these conditional probabilities in tables. We can compare them to those computed manually previously (section 2.2.1).



```

RGui
Fichier Edition Voir Misc Packages Fenêtres Aide
R Console
> heart.model <- naiveBayes(disease ~ exang + chest_pain, data = heart.data$
> print(heart.model)

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
  negative positive
0.6363636 0.3636364

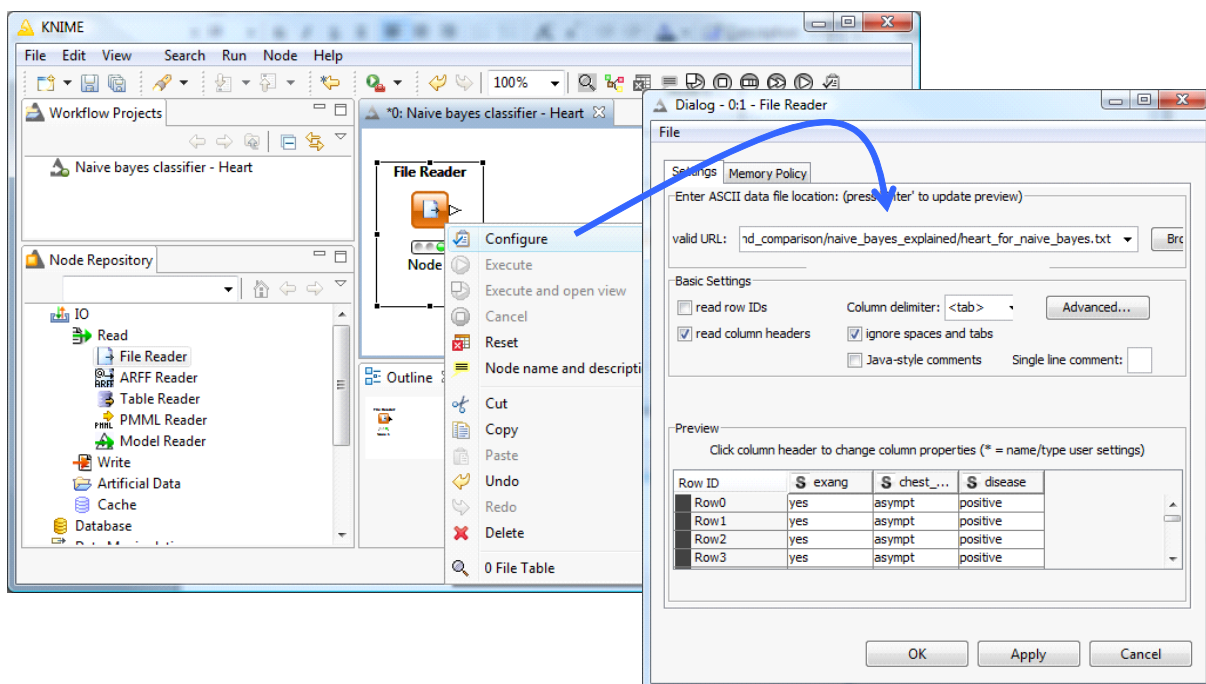
Conditional probabilities:
          exang
Y          no      yes
negative 0.8913043 0.1086957
positive 0.3490566 0.6509434

          chest_pain
Y          asympt atyp_angina non_anginal typ_angina
negative 0.21505376 0.51612903 0.22580645 0.04301075
positive 0.75925926 0.08333333 0.11111111 0.04629630
>

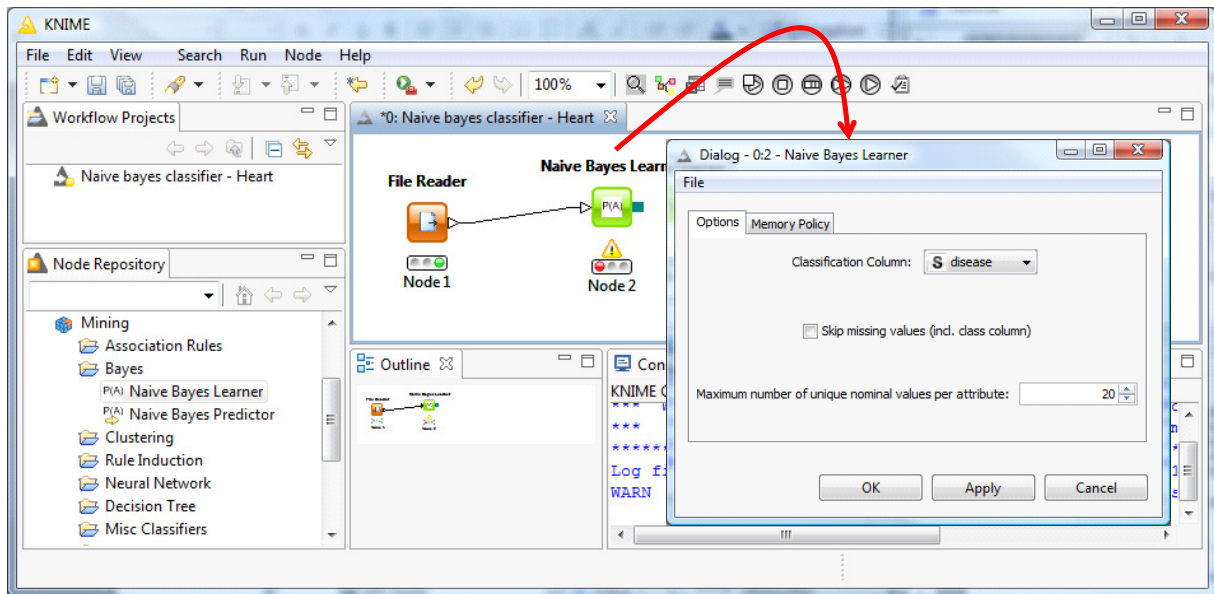
```

5.3 Knime 2.1.1

We start Knime (<http://www.knime.org/>). We create a workflow (FILE / NEW). We use the FILE READER component (IO / READ) to read the data file "HEART_FOR_NAIVE_BAYES.TXT".



Then, we add the NAÏVE BAYES LEARNER component (MINING / BAYES). We specify the target attribute (DISEASE).



We click on the EXECUTE AND OPEN VIEW contextual menu. We get the cross tabulations used for the calculations of the conditional probabilities.

The screenshot shows the 'Naive Bayes Learner View - 0:2 - Naive Bayes Learner' dialog box. It displays the following information:

Class counts for disease

Class:	negative	positive
Count:	182	104

P(chest_pain | class=?)

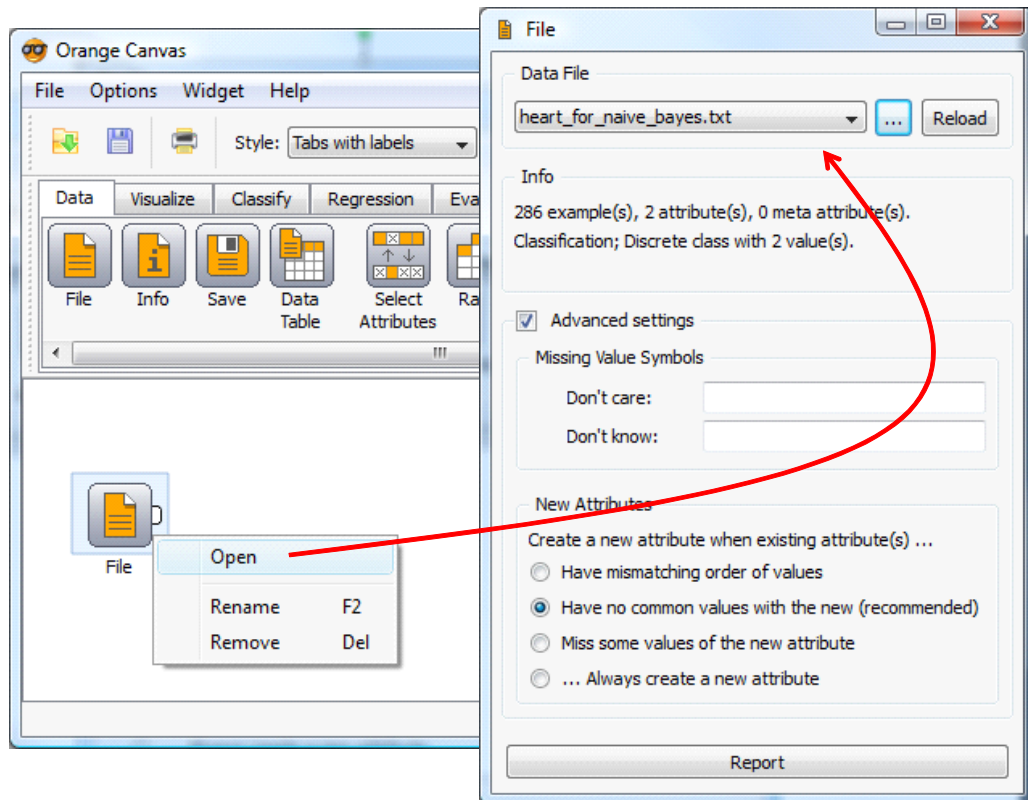
Class/chest_pain	asympt	atyp_angina	non_anginal	typ_angina
negative	39	95	41	7
positive	81	8	11	4
Rate:	120/286	103/286	52/286	11/286

P(exang | class=?)

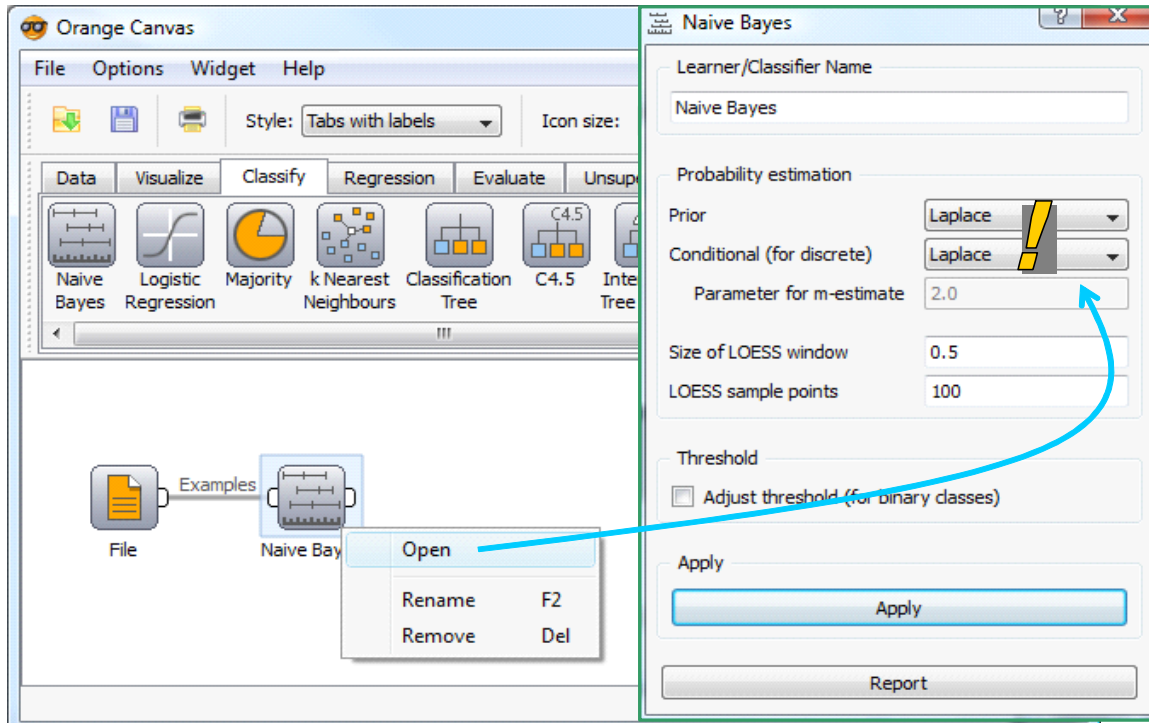
Class/exang	no	yes
negative	163	19
positive	36	68
Rate:	199/286	87/286

5.4 Orange 2.ob

We start Orange (<http://www.aialab.si/orange/>). We use the FILE component (DATA) to read the data file "HEART_FOR_NAIVE_BAYES.TXT".

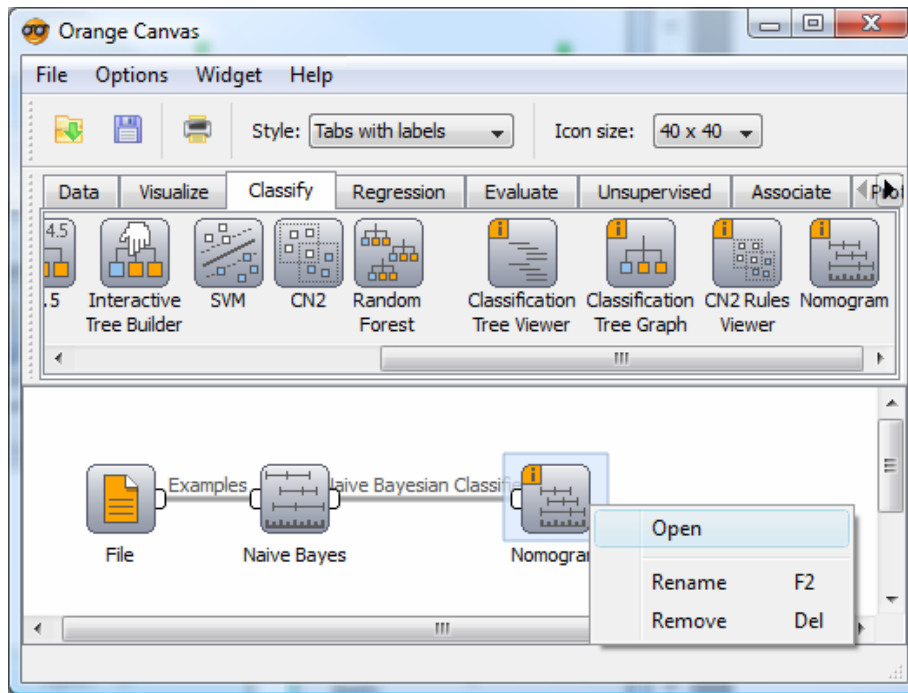


Then we add and we connect the NAIVE BAYES component (CLASSIFY). We click on the OPEN menu to set the parameters of the learning algorithm.

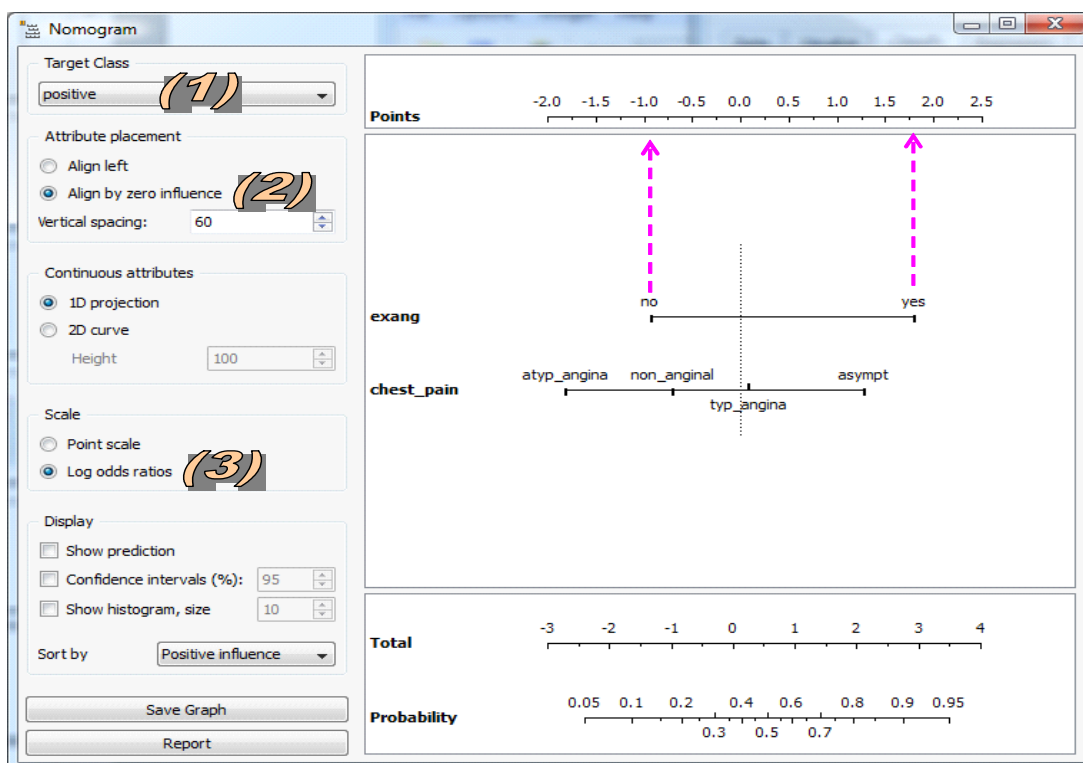


We ask the Laplace correction for the probability estimation i.e. $m = 1$.

To visualize the model, we add the NOMOGRAM component that we connect to NAÏVE BAYES. We click on the OPEN menu.



The nomogram is a visualization tool which is not really popular. Yet, it provides a very intuitive visualization of the influence of each predictor on the target attribute³. We use the following settings: (1) target class = positive; (2) align the attributes according to the null influence; (3) the scale is the odds-ratio. We obtain the following visualization window.

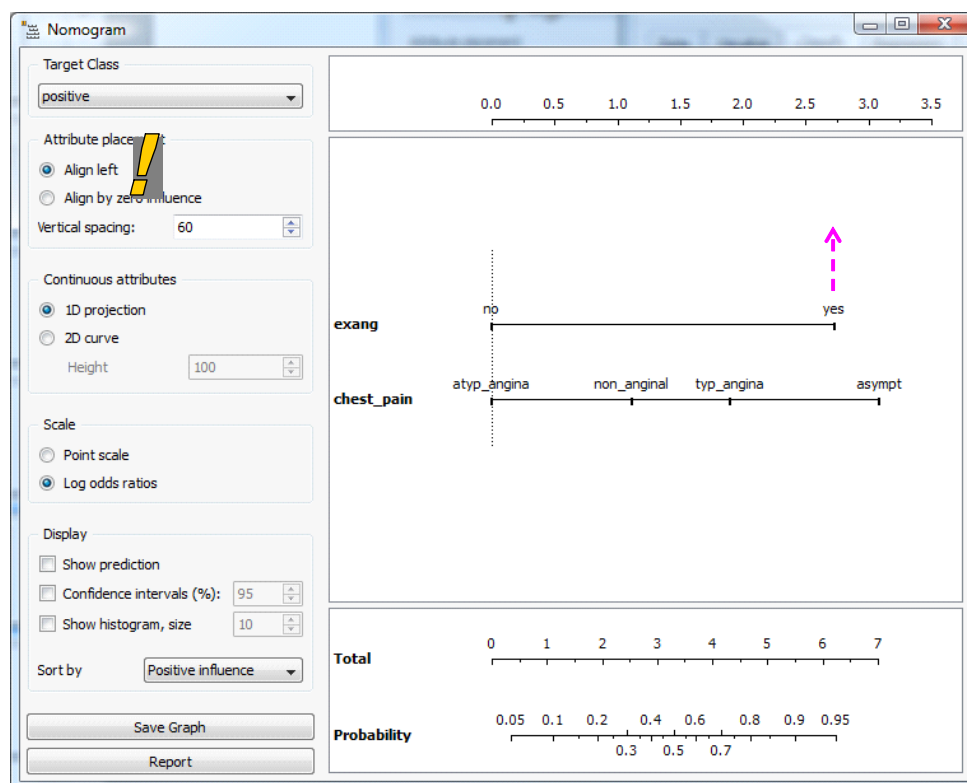


³ <http://www.ailab.si/blaz/papers/2004-PKDD.pdf>

Somehow, the values provided by Orange are very similar to those provided by Tanagra. Let us consider the EXANG predictor. We use the following cross-tabulation to compute the conditional probabilities.

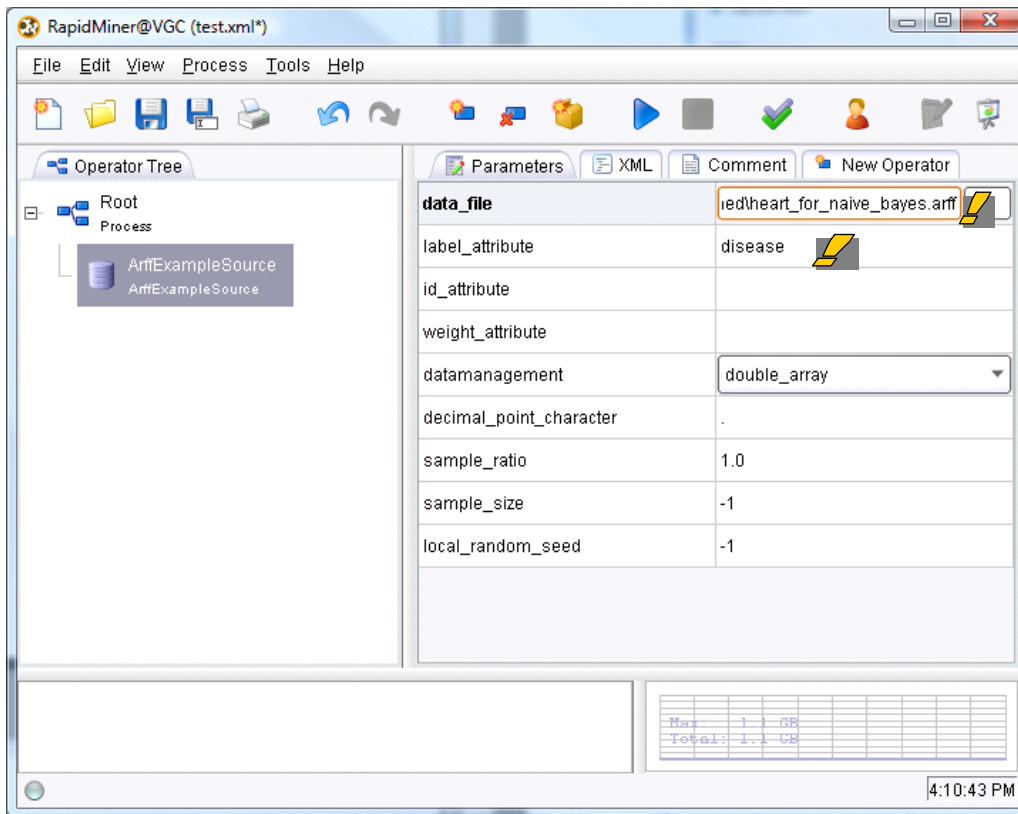
Nombre de disease	exang		Total
	yes	no	
positive	68	36	104
negative	19	163	182
Total	87	199	286

The "odds" is the ratio between the number of positive instances and the number of negative ones. For the whole sample, it is $\frac{104}{182} = 0.57$. Within the "EXANG = yes" group, the odds is $\frac{68}{19} = 3.58$. Orange computes the natural logarithm of the ratio between these odds i.e. $\ln \frac{3.58}{0.57} = 1.83$. This explains the positioning of this modality into the nomogram. In the same way, for the group "EXANG = no", we get $\ln \frac{0.22}{0.57} = -0.95$. Thus, we can conclude that EXANG = YES has a positive influence on the presence of the DISEASE (DISEASE = yes). If we want a "left alignment", Orange computes the logarithm of the ratio between the odds of the (EXANG = YES) and (EXANG = NO) groups i.e. $\ln \frac{3.58}{0.22} = 2.79$. This value is very similar (the nomogram does not use the Laplace estimate) to the coefficient of EXANG = YES into the decision function provided by Tanagra.

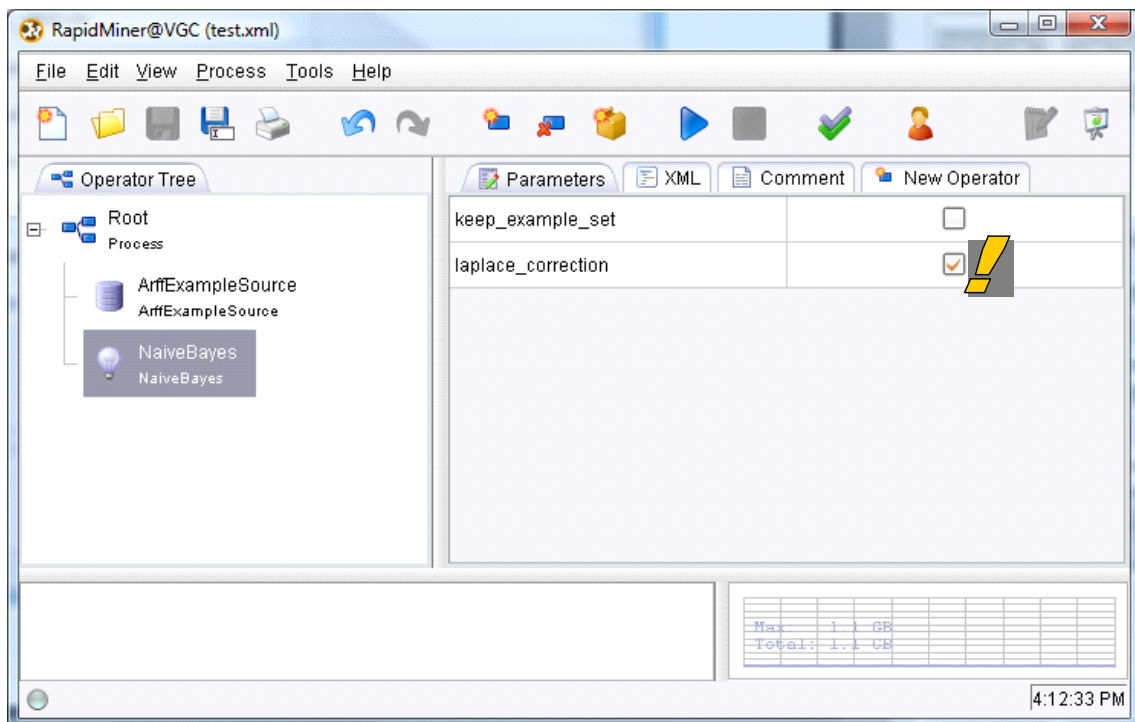


5.5 Rapidminer 4.6.0

We launch RapidMiner (<http://rapid-i.com/content/view/181/190/>). We create a new project by clicking on the FILE / NEW menu. We insert the ARFF EXAMPLE SOURCE component with the following settings.



Then we add the NAIVE BAYES component. It uses also the Laplace correction for the estimation of the conditional probabilities.



We launch the calculations (PROCESS RUN). Various visualization widows are provided. We select the "distribution table". We get the conditional probabilities used in the classification process.

SimpleDistribution
NaiveBayes

Text View Plot View Distribution Table

Attribute	Parameter	positive	negative
exang	value=yes	0.654	0.104
exang	value=no	0.346	0.896
exang	value=unknc	0.000	0.000
chest_pain	value=asymj	0.779	0.214
chest_pain	value=atyp_	0.077	0.522
chest_pain	value=non_	0.106	0.225
chest_pain	value=typ_ar	0.038	0.038

Save...

2 distributions
(created by NaiveBayes)
P Mar 23, 2010 4:13:14 PM: [NOTE] Process finished successfully after 0 s

4:13:27 PM

6 Conclusion

We highlight into this tutorial an original visualization of the naive bayes classifier when we deal with discrete predictors. The main advantage of this representation is that we can easily deploy the classifier. It is especially interesting in a business process context.

About the continuous predictors, we incorporate a naive bayes classifier intended to the continuous predictors in a future version of Tanagra. We will see also that we can obtain a classifier which is a linear combination of the predictive variables (or the squares of the variables according to the assumptions of homoscedasticity or heteroscedasticity). The deployment of the model is also easy for the case of continuous predictors.

Finally, the naive bayes approach becomes particularly interesting when we combine it with a feature selection system. We will describe it in a future tutorial. Thus, we consider only the relevant and not redundant variables for construction of the final model. The interpretation of the results is easier in this context. Moreover, the classifier is more reliable according to the Occam's Razor principle.