1 - Topic

Tools for the diagnostic and the assessment of logistic regression.

This tutorial describes the implementation of tools for the diagnostic and the assessment of a logistic regression. These tools are available in Tanagra version 1.4.33 (and later).

We deal with a credit scoring problem. We try to determine by using logistic regression the factors underlying the agreement or refusal of a credit to customers. We perform the following steps:

- Estimating the parameters of the classifier;
- Retrieving the covariance matrix of coefficients;
- Assessment using the Hosmer and Lemeshow goodness of fit test;
- Assessment using the reliability diagram;
- Assessment using the ROC curve;
- Analysis of residuals, detection of outliers and influential points.

On the one hand, we use **Tanagra 1.4.33**. Then, on the other hand, we perform the same analysis using the **R 2.9.2 software [**glm(.) procedure**].**

2 - Dataset

Our data file « LOGISTIC_REGRESSION_DIAGNOSTICS.XLS[1] » contains n = 100 observations. The binary target attribute is « ACCEPTATION.CREDIT » (« yes » or « no »). The predictive variables are

Name	Description	Type
Age	Age of the customer	Continuous
Income.Per.Dependent	Income per dependent in the household	Continuous
Derogatory.Report	At least one problem with the bank was reported	Binary

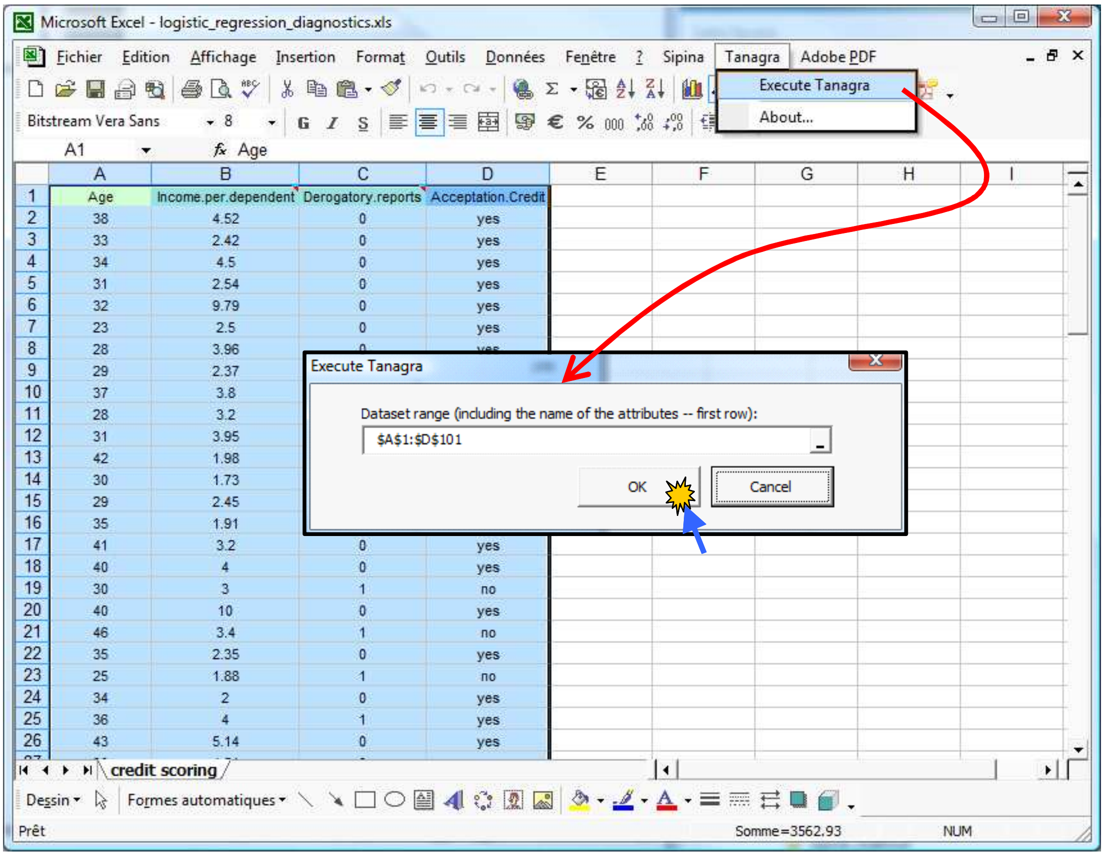
3 - Analysis in Tanagra

Importing the data file

To import the dataset, we open the file into Excel spreadsheet. Then, by the way of Tanagra.xla[2] add-in, we click on the TANAGRA / EXECUTE TANAGRA menu. Tanagra is automatically launched, the data file is imported.

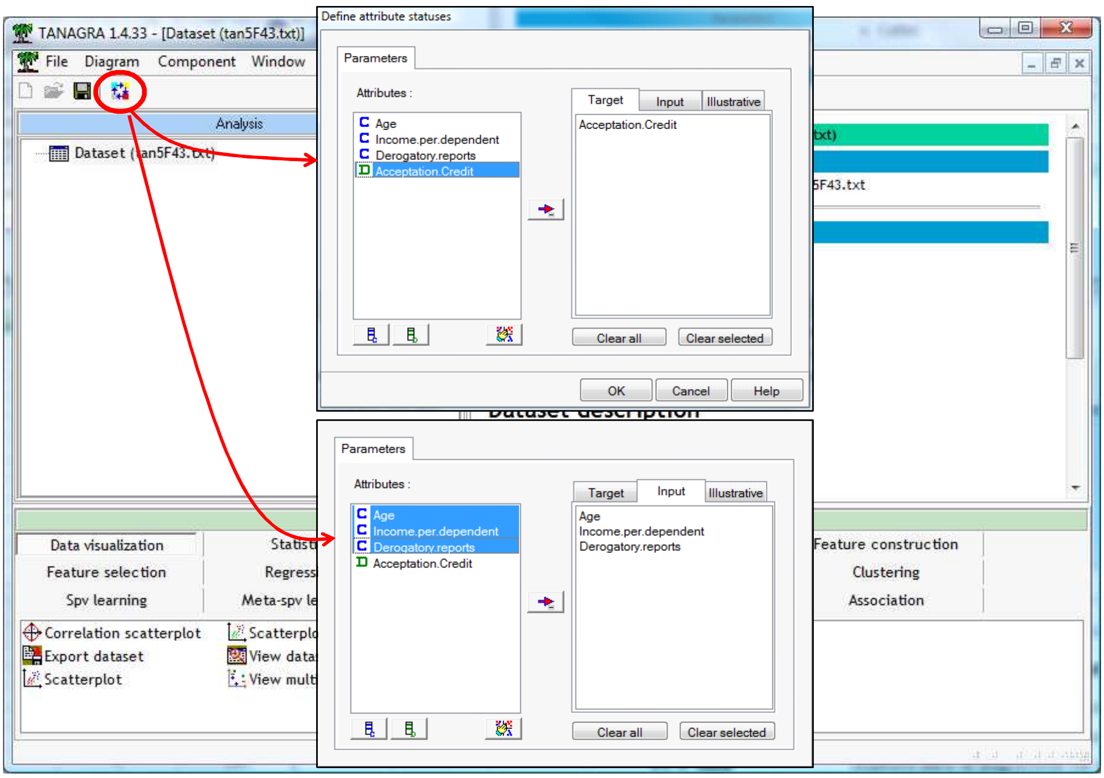
[1] http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/logistic_regression_diagnostics.zip

[2] http://data-mining-tutorials.blogspot.com/2008/10/excel-file-handling-using-add-in.html; a similar tool is available for Open office Calc: http://data-mining-tutorials.blogspot.com/2008/10/ooocalc-file-handling-using-add-in.html

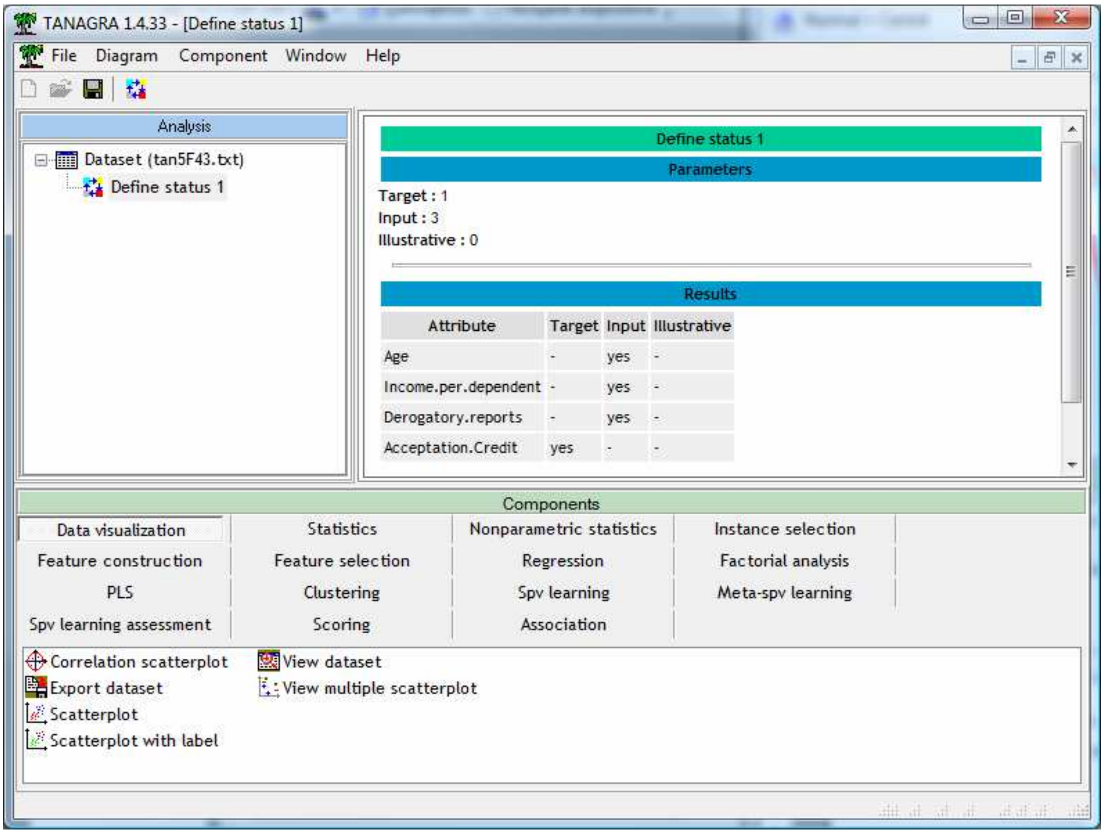


Launching the logistic regression

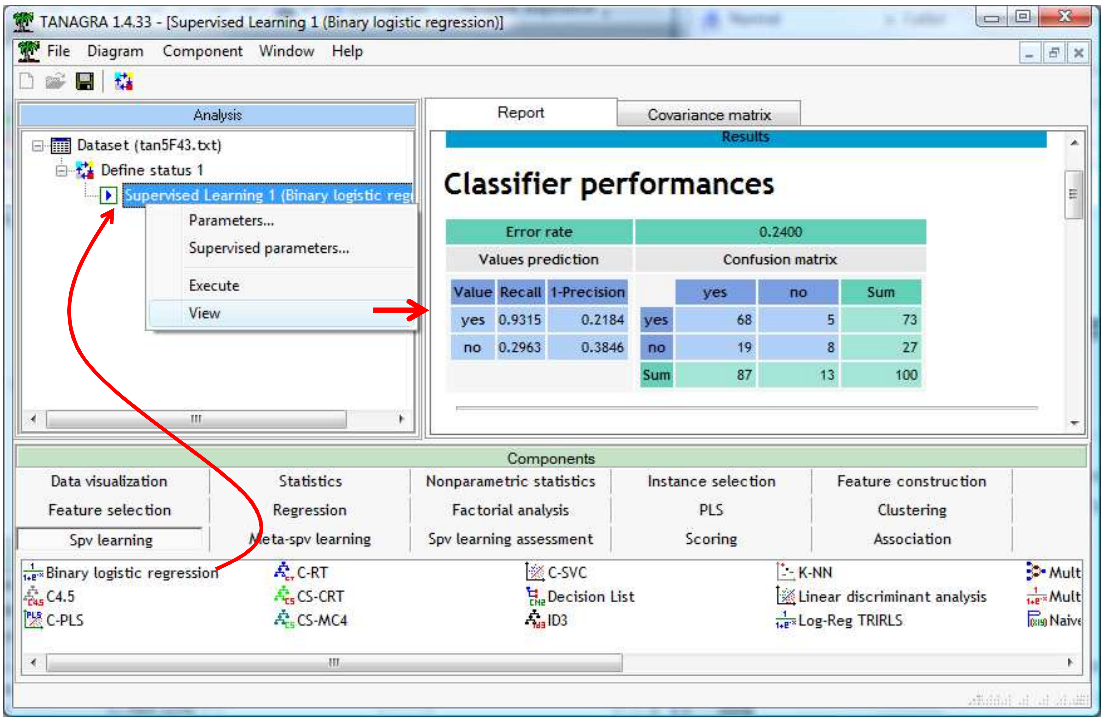
We add the DEFINE STATUS component into the diagram. We set ACCEPTATION.CREDIT as TARGET, the other attributes as INPUT.



We click on the OK button and we activate the VIEW menu. We obtain the following result.



We add the BINARY LOGISTIC REGRESSION tool (SPV LEARNING tab). We click on the VIEW menu.



The first information displayed is the confusion matrix, computed on the learning sample (**CLASSIFIER PERFORMANCES**). The resubstitution error rate is 0.24. We have the recall and (1.0-precision) for each value of the target attribute.

The **MODEL FIT STATISTICS** section assesses the global significance of the model. We compare the current model to the null model i.e. the model with only the intercept. Roughly speaking, the classifier is relevant if the AIC criterion (or BIC / SC criterion) of the model is lower than the AIC (BIC) of the null model. We observe here that SC(MODEL) = 119.063 < SC(INTERCEPT) = 121.257.

Adjustement quality

Predicted attribute		Acceptation.Credit
Positive value		yes
Number of examples		100
Model Fit Statistics		
Criterion	Intercept	Model
AIC	118.652	108.642
SC	121.257	119.063
-2LL	116.652	100.642
Model Chi² test (LR)		
Chi-2		16.0094
d.f.		3
P(>Chi-2)		0.0011
R²-like		
McFadden's R²		0.1372
Cox and Snell's R²		0.1479
Nagelkerke's R²		0.2149

The **MODEL CHI² TEST (LR)** section implements the likelihood ratio test. It allows also to assess the global significance of the model. The CHI-squared statistic CHI-2 = LR = -2LL[INTERCEPT] – (-2LL[MODEL]) = 116.652 – 100.642 = 16.0094. The degree of freedom is equal to the number of explanatory variables (3). Thus, the p-value of the test is 0.0011 according a chi-squared distribution. At the 5% significance level, we conclude that the model is globally significant.

R²-LIKE computes some pseudo r-squared statistics. When the value is close to 0, it means that the model is not relevant in comparison to the null model.

Attributes in the equation

Attribute	Coef.	Std-dev	Wald	Signif
constant	2.745220	1.1430	5.7685	0.0163
Age	-0.062411	0.0336	3.4541	0.0631
Income.per.dependent	0.216797	0.1795	1.4587	0.2271
Derogatory.reports	-1.929304	0.5906	10.6722	0.0011

Next we obtain the estimated parameters of the model i.e. the coefficients of the logistic regression. Each coefficient is evaluated with the Wald test. If the p-value is lower than the significance level, the parameter is significant. Here, at 5% significance level, only the parameters for the "Derogatory reports" and the intercept are significant. At the 10% significance level, “Age” becomes significant.

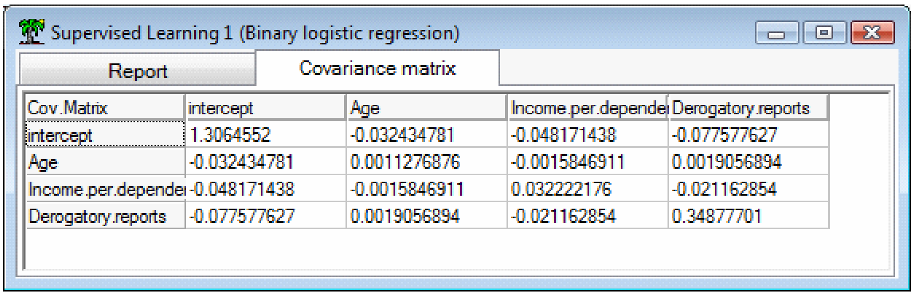
Odds ratios and 95% confidence intervals

Attribute	Coef.	Low	High
Age	0.9395	0.8797	1.0034
Income.per.dependent	1.2421	0.8737	1.7658
Derogatory.reports	0.1452	0.0456	0.4622

For people who have the same age and income per dependent, there are 6.89 times more chances to reject the credit when at least one derogatory report is made [1/exp(-1.929304) = 1/0.1452 = 6.89]. These are the **odds-ratios** described into the next table. We obtain also their confidence intervals at 95% confidence level.

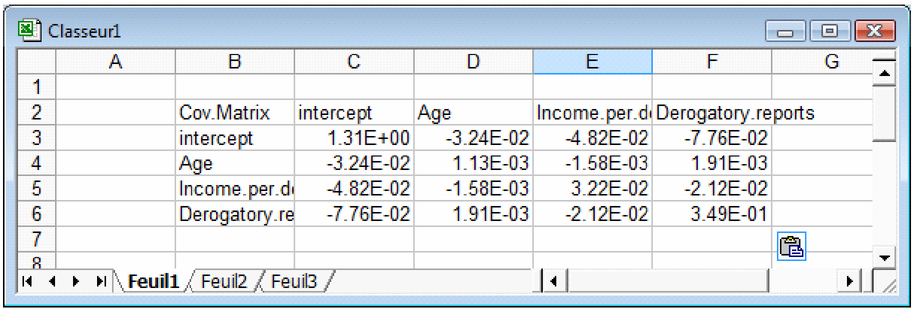
Testing the significance of a subset of coefficients

We need to the covariance matrix of the coefficients for the testing the simultaneous significance of a subset of them (Wald test). In Tanagra 1.4.33 version (and later), they are supplied in a second tab (COVARIANCE MATRIX) of the visualization window.



Cov.Matrix	intercept	Age	Income.per.depende	Derogatory.reports
intercept	1.3064552	-0.032434781	-0.048171438	-0.077577627
Age	-0.032434781	0.0011276876	-0.0015846911	0.0019056894
Income.per.depende	-0.048171438	-0.0015846911	0.032222176	-0.021162854
Derogatory.reports	-0.077577627	0.0019056894	-0.021162854	0.34877701

We can copy these values into the Excel spreadsheet (COMPONENT / COPY RESULTS menu).



	A	B	C	D	E	F	G
1							
2		Cov.Matrix	intercept	Age	Income.per.d	Derogatory.reports	
3		intercept	1.31E+00	-3.24E-02	-4.82E-02	-7.76E-02	
4		Age	-3.24E-02	1.13E-03	-1.58E-03	1.91E-03	
5		Income.per.d	-4.82E-02	-1.58E-03	3.22E-02	-2.12E-02	
6		Derogatory.re	-7.76E-02	1.91E-03	-2.12E-02	3.49E-01	
7							

For instance, if we want to test « H0 : a(AGE) = a(INCOME.PER.DEPENDENT) = 0 », we can compute the test statistic with the following formula

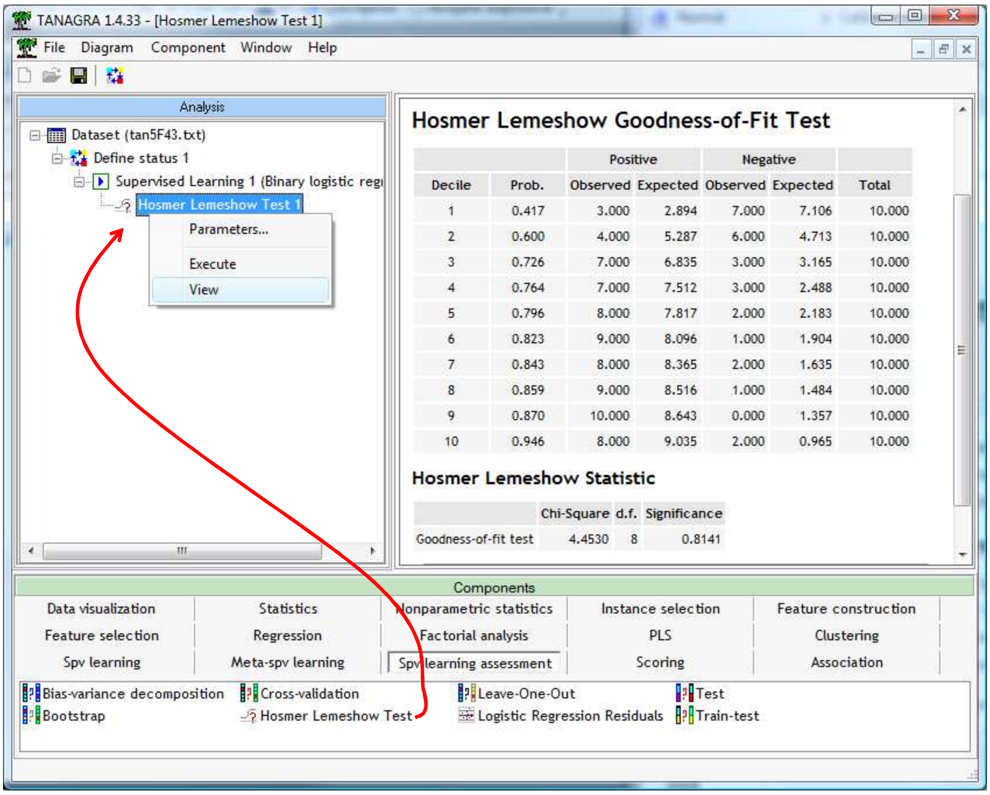
$$
\begin{aligned}
W &= \begin{pmatrix} -0.062411 & 0.216797 \end{pmatrix} \begin{pmatrix} 1.13\times10^{-3} & -1.58\times10^{-3} \\ -1.58\times10^{-3} & 3.22\times10^{-2} \end{pmatrix}^{-1} \begin{pmatrix} -0.062411 \\ 0.216797 \end{pmatrix} \\
&= \begin{pmatrix} -0.062411 & 0.216797 \end{pmatrix} \begin{pmatrix} 952.61 & 46.85 \\ 46.85 & 33.34 \end{pmatrix} \begin{pmatrix} -0.062411 \\ 0.216797 \end{pmatrix} \\
&= 4.0097
\end{aligned}
$$

With the CHI-2 distribution (2 d.f.), the computed p-value is 0.1347. At the 5% significance level, we cannot reject the null hypothesis.

Hosmer and Lemeshow goodness of fit test

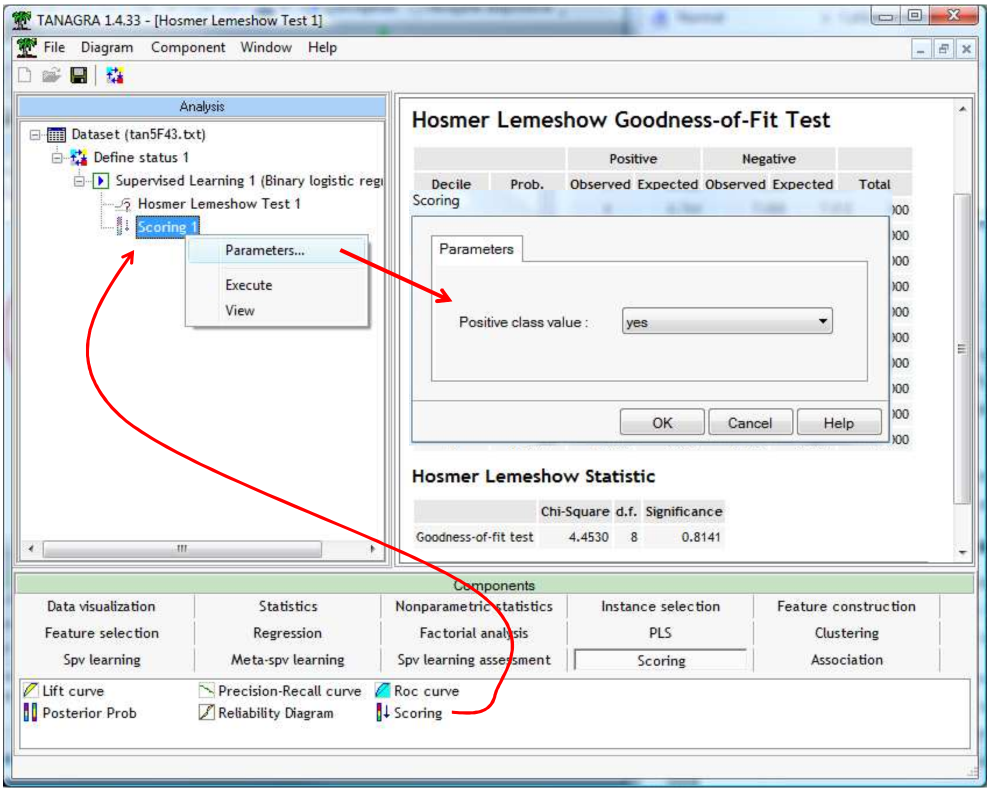
The Hosmer and Lemeshow is a test for the overall fit of the model. This test is especially appropriate when the explanatory variables are continuous. It is more robust than the standard chi-square test based on the deviance residuals. The model adequately fits the data if the test highlights non-significance. We add the HOSMER LEMESHOW TEST component (SPV LEARNING ASSESSMENT tab) behind the regression. This is the only location where we can insert it into the diagram anyway.

We click on the VIEW menu. We obtain the table for the calculations. The value of the test statistic is CHI-2 = 4.4530 with the p-value = 0.8141. Because the p-value is higher than the significance level (5%), we conclude than the model fits adequately the observed dataset.



Reliability diagram

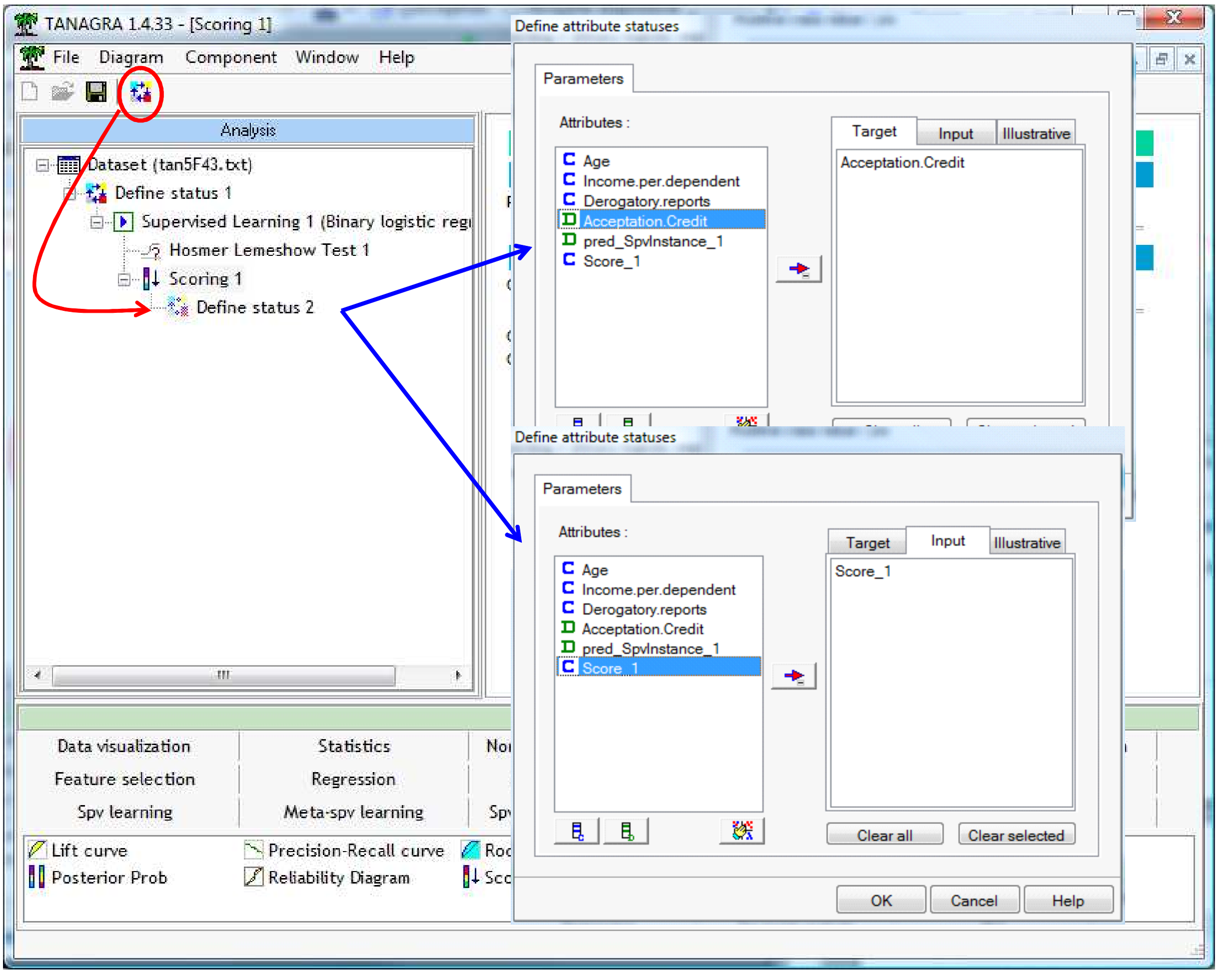
The reliability diagram compares the observed probability and the estimated posterior probability of the model. The instances are subdivided into groups. The model fits the dataset if the points are aligned on a straight line.



First, we must compute the posterior probability of "yes" supplied by the model. We use the SCORE (SCORING tab). We set "yes" as the positive class value.

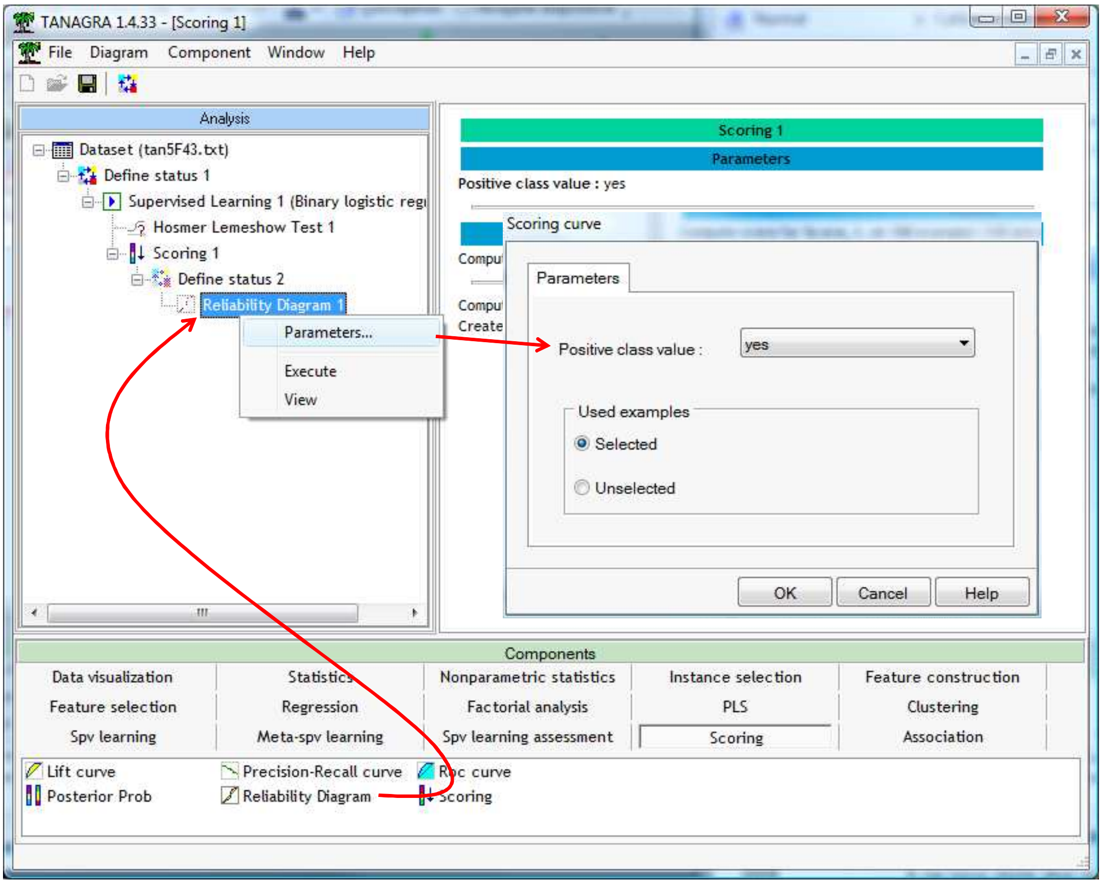
We click on the VIEW menu to launch the calculations. A new column is added to the current dataset. It corresponds to the computed posterior probability to be "yes" for each instance.

Then, we add the DEFINE STATUS tool. We set ACCEPTATION.CREDIT as TARGET, and SCORE_1 (the SCORE computed before) as INPUT. We note that we can set many scores as input. It is useful when we want to compare the performance of classifiers. We note also that the score column is not necessarily a probability. It must only make possible to rank the examples according to their tendency to be positive.

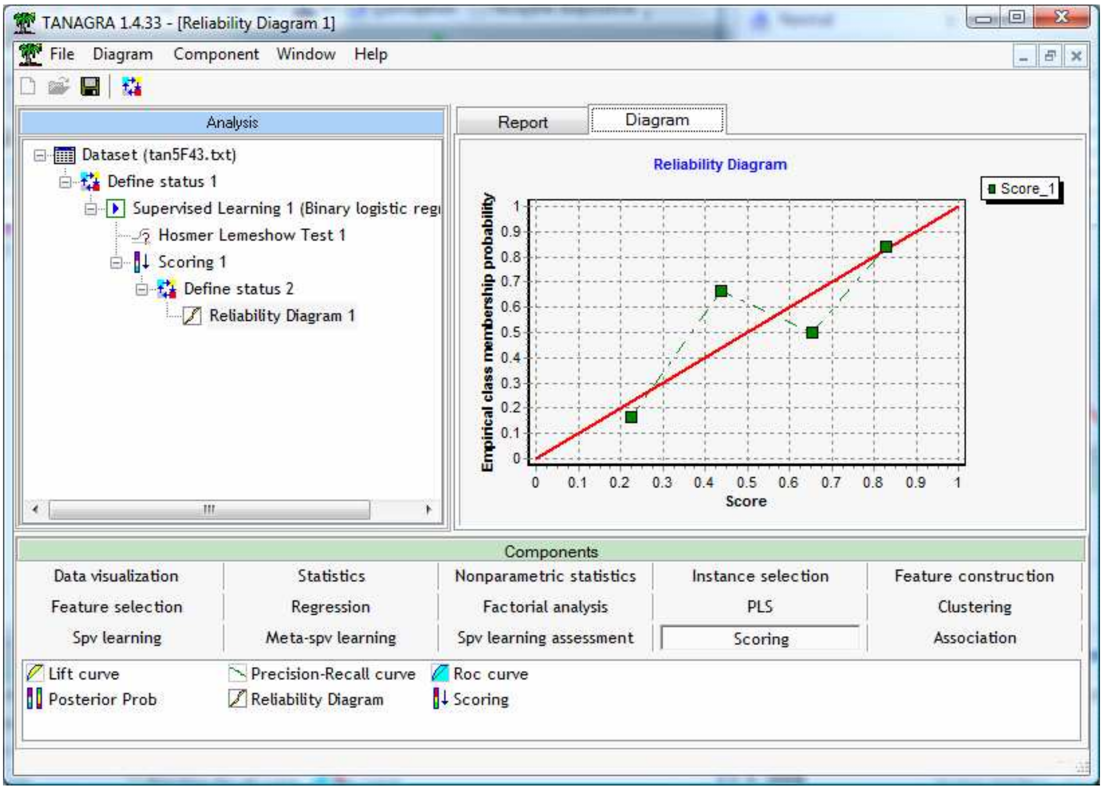


Then we add the RELIABILITY DIAGRAM tool (SCORING tab). We click on the PARAMETRES menu. We specify the positive class value (ACCEPTATION.CREDIT = "yes").

We observe that we can build the reliability diagram on the selected examples (learning sample) or on the unselected examples (test sample). If the dataset was subdivided previously using the SAMPLING component for instance, we can compute the reliability diagram on the test sample. The result is more robust.



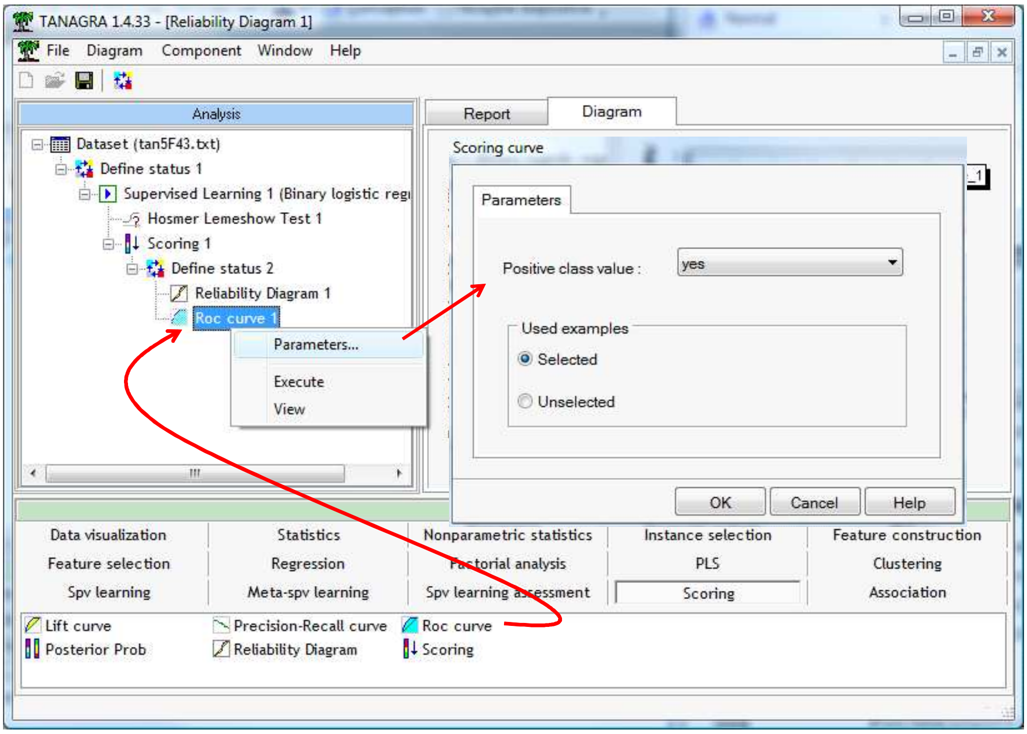
We click on the VIEW menu.



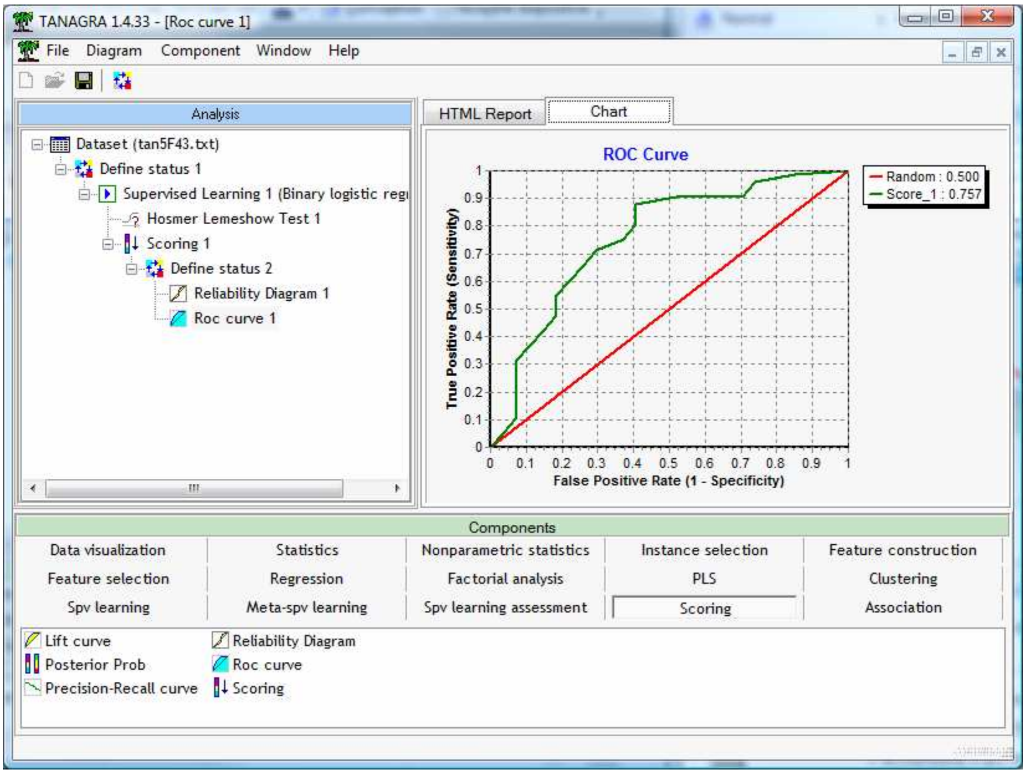
The model is not very good. We observe that the scores are overestimated into the second group; they are underestimated into the third one.

ROC curve

The ROC curve is a very useful tool for assessing the performance of a classifier. Among their multiple interpretations, we will say that it allows to evaluate the ability of the model to assign a higher score to the positive instances (than the negative instances). We add the ROC CURVE tool (SCORING). We set the following parameters.



We click on the VIEW menu. We obtain the ROC curve.

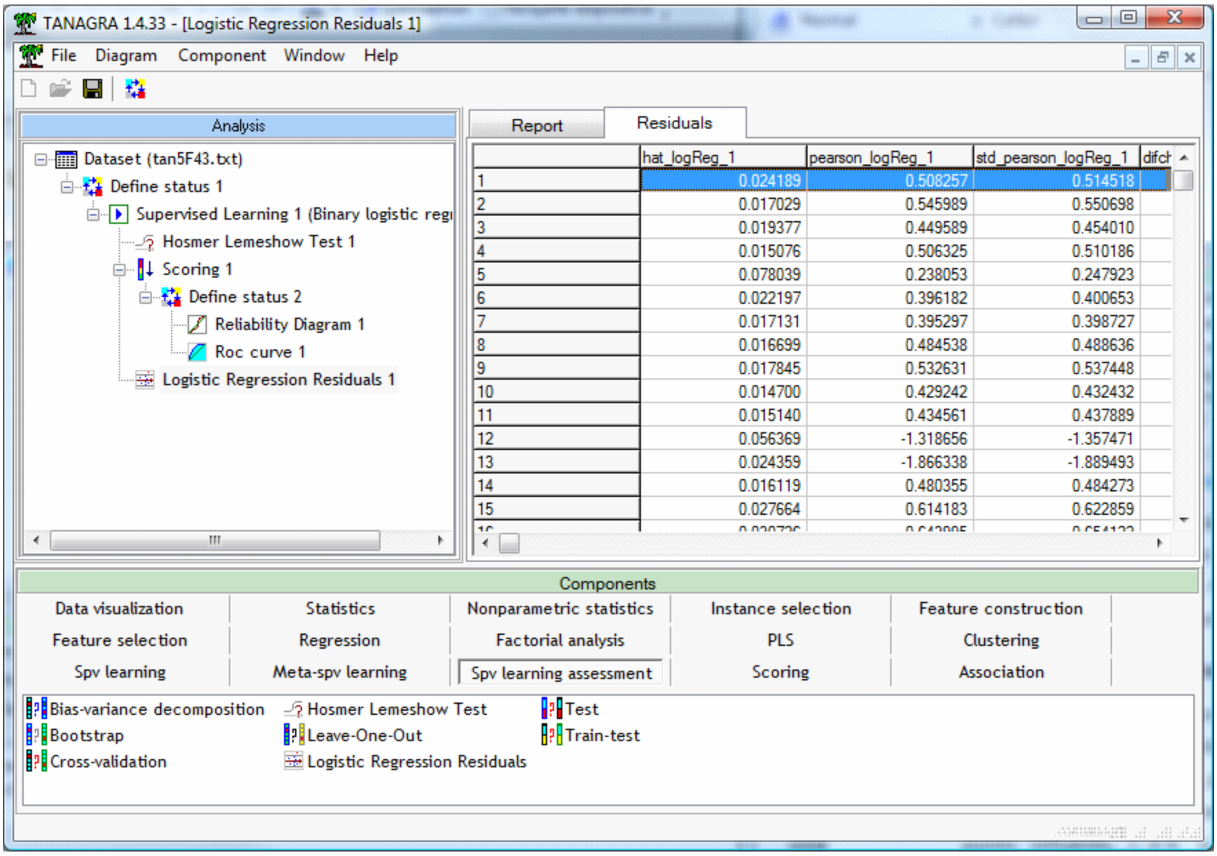


The comparison of the models is also possible with this tool. We obtain the area under curve AUC = 0.7575. It seems that our model is "fair" (http://gim.unmc.edu/dxtests/ROC3.htm).

By comparing the results provided by several indicators, we can get a more reliable opinion about the quality of the model. This aspect is very important. We must not focus on a single indicator.

Residual analysis

The residual analysis allows to check the validity of the model. It allows also to detect the outliers and influential points. We add the LOGISTIC REGRESSION RESIDUALS tool (SPV LEARNING ASSESSMENT tool) behind the logistic regression. We click on the VIEW menu.

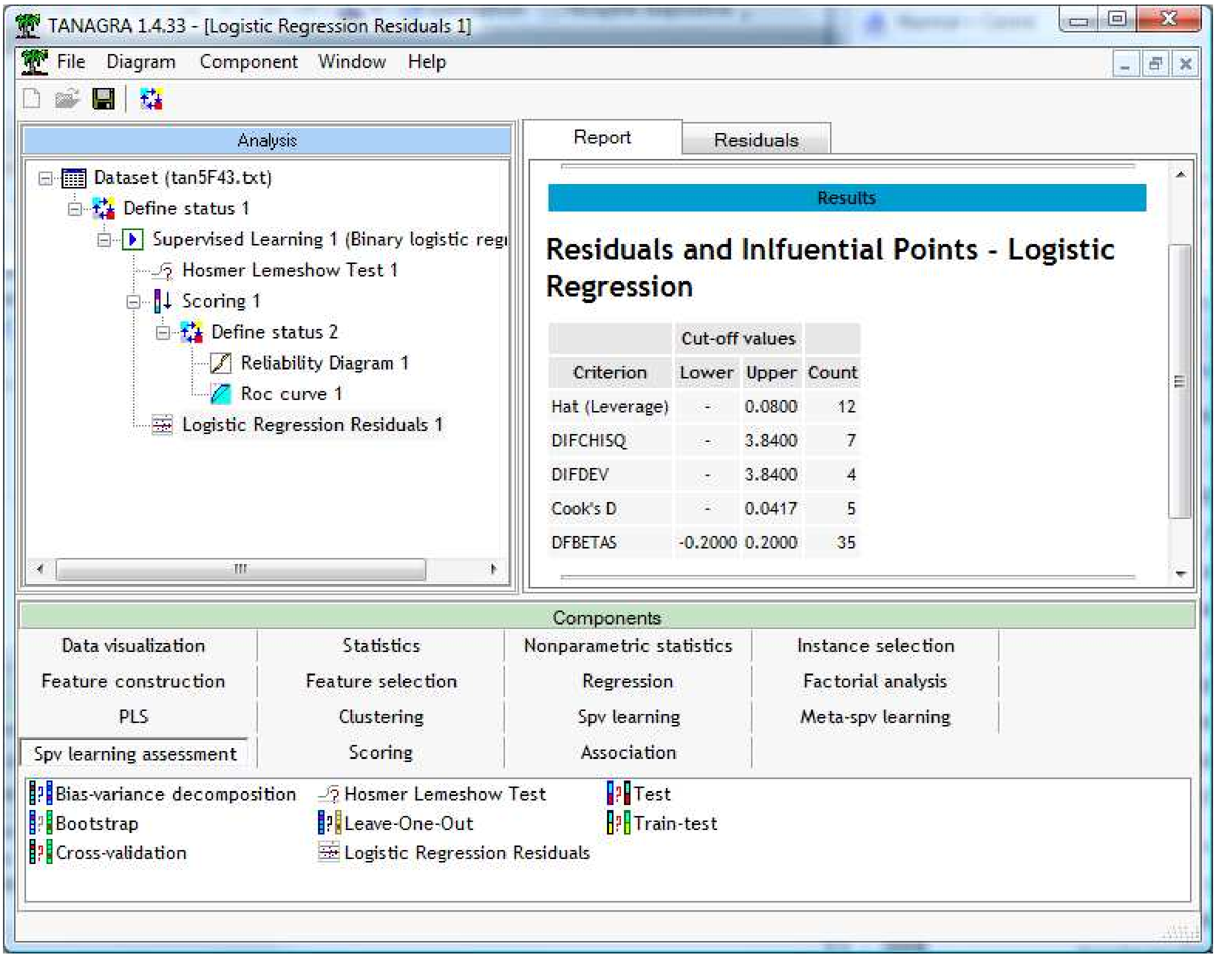


Some indicators are available. They are computed for each individual.

- HAT: leverage;
- PEARSON: Pearson residual;
- STD_PEARSON: standardized Pearson residual;
- DIFCHISQ: contribution to the Pearson statistic;
- DEVIANCE: deviance residual;
- STD_DEVIANCE: standardized deviance residual;
- DIFDEV: contribution to the deviance;
- COOK: Cook's distance ;
- DEFBETA: DFBETA for each explanatory variable, including the intercept;
- DFBETAS: standardized DFBETA.

The values which are higher than (and/or lower than) the cut values are in bold. But it is often more interesting to copy the values into a spreadsheet and sorting the table according some indicators. We can better detect the suspicious examples.

Into the REPORT tab, we have a summary of the number of outliers or influential points detected for each indicator.



4 - Analysis in R

We do not detail all operations with R. We give only the commands that allow to achieve the requested results. The source code “LOGISTIC_REGRESSION_DIAGNOSTICS.R” is included into the archive which is distributed with this tutorial.

Importing the data file and performing the logistic regression

We set the following commands to import the data file and to perform the logistic regression. We load the dataset in the XLS (Excel) file format using the "xlsReadWrite" package.

```
#clear the internal memory
rm(list=ls())
#in order to handle a XLS file format
library(xlsReadWrite)
#loading the dataset
setwd("D:/DataMining/Databases_for_mining/dataset_for_soft_dev_and_comparison/logistic regre
donnees <- read.xls(file = "logistic_regression_diagnostics.xls",rowNames = FALSE,sheet=1)
summary(donnees)
#performing the logistic regression
modele <- glm(Acceptation.Credit ~ ., data = donnees, family = "binomial")
resume <- summary(modele)
print(resume)
```

We obtain the same results as Tanagra.

```
> print(resume)

Call:
glm(formula = Acceptation.Credit ~ ., family = "binomial", data = do$

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2621  -0.7108   0.5680   0.7034   2.0814

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)           2.74522    1.14306   2.402  0.01632 *
Age                  -0.06241    0.03358  -1.858  0.06311 .
Income.per.dependent  0.21680    0.17951   1.208  0.22714
Derogatory.reports   -1.92930    0.59058  -3.267  0.00109 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 116.65  on 99  degrees of freedom
Residual deviance: 100.64  on 96  degrees of freedom
AIC: 108.64

Number of Fisher Scoring iterations: 4
```

To obtain the prediction of the model and the confusion matrix, we set.

```
#computing the confusion matrix
prediction <- ifelse(predict(modele,type="response") > 0.5, "yes", "no")
print(table(donnees$Acceptation.Credit,prediction))
```

We obtain.

```
> #computing the confusion matrix
> prediction <- ifelse(predict(modele,type="response") > 0.5, "yes", "no")
> print(table(donnees$Acceptation.Credit,prediction))
     prediction
      no yes
  no   8  19
  yes  5  68
>
```

Covariance matrix of coefficients

We need the covariance matrix in order to test the significance of a subset of coefficients. It is associated to the "summary object".

```
> #obtaining the covariance matrix
> print(attributes(resume))
$names
 [1] "call"           "terms"          "family"         "deviance"       "aic"
 [6] "contrasts"      "df.residual"    "null.deviance"  "df.null"        "iter"
[11] "deviance.resid" "coefficients"   "aliased"        "dispersion"     "df"
[16] "cov.unscaled"   "cov.scaled"

$class
[1] "summary.glm"

> print(resume$cov.unscaled)
                     (Intercept)          Age Income.per.dependent Derogatory.reports
(Intercept)           1.30658434 -0.032438283         -0.048175586       -0.077548615
Age                  -0.03243828  0.001127781         -0.001584562        0.001904885
Income.per.dependent -0.04817559 -0.001584562          0.032222092       -0.021163524
Derogatory.reports   -0.07754862  0.001904885         -0.021163524        0.348782597
> 
```

R has powerful tools for the matrix operations. We can use them directly.

Hosmer - Lemeshow test

For the Hosmer-Lemeshow test, the ROC curve and the reliability diagram, we need the "score" column. We use the **predict(.)** command.

```
#obtaining the score column
score <- predict(modele,type="response")
print(score)
```

Then, we write a function to compute the Hosmer and Lemeshow statistic. I think this source code is very simplistic, but it allows to the reader to understand the details of the treatments[3].

[3] For instance, here is a sorter source code loaded from the web. Of course, we obtain the same results.

```
#another Hosmer-Lemeshow procedure
#from https://stat.ethz.ch/pipermail/r-help/2008-September/174044.html
cut.score <- cut(score, breaks = quantile(score, probs = seq(0, 1, 0.1)), include.lowest = T)
obs <- xtabs(cbind(1 - y, y) ~ cut.score)
expect <- xtabs(cbind(1 - score, score) ~ cut.score)
chisq <- sum((obs - expect)^2/expect)
P <- 1 - pchisq(chisq, 10 - 2)
print(c("X^2" = chisq, Df = 10 - 2, "P(>Chi)" = P))
```

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\logistic regression\residuals\logistic_reg...

```
#Hosmer - Lemeshow test
y <- ifelse(unclass(donnees$Acceptation.Credit)==2, 1, 0)
total <- 0.0
previous <- 0.0
for (p in seq(0.1, 1, 0.1)){
  #covered examples into the quantile
  seuil <- quantile(score,p)
  examples <- (score > previous & score <= seuil)
  #number of covered examples
  m.obs <- length(which(examples))
  #positive
  m.pos.obs <- sum(y[examples])
  m.pos.expected <- sum(score[examples])
  #negative
  m.neg.obs <- m.obs - m.pos.obs
  m.neg.expected <- m.obs - m.pos.expected
  #statitic
  total <- total + (m.pos.obs - m.pos.expected)^2/m.pos.expected
  total <- total + (m.neg.obs - m.neg.expected)^2/m.neg.expected
  #next
  previous <- seuil
}
print(c("Hosmer Lemeshow Statistic" = total, "p-value" = pchisq(total,8,lower.tail=F)))
```

The result is consistent with those of Tanagra.

R Console

```
> print(c("Hosmer Lemeshow Statistic" = total, "p-value" = pchisq(total,8,lower.tail=F)))
Hosmer Lemeshow Statistic                   p-value
                4.4529796                 0.8141187
```

Residual analysis and influential points

Some commands enable to compute the indicators for the residual analysis.

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_co...

```
#some of résiduals and influentials points functions
mesures.prim <- influence(modele)
print(mesures.prim)
mesures.bis <- influence.measures(modele)
print(mesures.bis)
#dfbeta et ddfbetas
mesures.dfbeta <- dfbeta(modele)
mesures.dfbetas <- dfbetas(modele)
```

influence() provides the leverage (hat values), DFBETA, deviance and Pearson residuals.

Influence.measures() provides DFBETAS, DFFIT, COVRATIO, Cook's distance, leverage.

Other commands are available. We observe that all the results are consistent with those of Tanagra, excepting the DFBETA(s). I have not found the origin of the differences. I note only that the DFBETA(s) provided by Tanagra, SAS and SPSS (state-of-the-art commercial tools) are identical.

5 - Conclusion

In this tutorial, we tried to give an overview of the tools used to assess and diagnose the logistic regression. Some are specific to the regression (Hosmer-Lemeshow test, analysis of residuals), while others are more generic, they can be used for any classifier which can provide a prediction or a "score" (confusion matrix, ROC curve).