

1 Objectif

Diagnostic et évaluation de la régression logistique.

Ce tutoriel décrit la mise en œuvre des outils d'évaluation et de diagnostic de la régression logistique binaire, disponibles depuis la version 1.4.33 de Tanagra. Les techniques et les formules afférentes sont présentées dans le fascicule de cours que nous avons mis en ligne récemment (septembre 2009 – « [Pratique de la Régression Logistique – Régression logistique binaire et polytomique¹](#) »). Il serait intéressant de le charger également afin de pouvoir s'y référer lorsque nous décrivons les résultats.

Nous traitons un problème de crédit scoring. Nous cherchons à déterminer à l'aide de la régression logistique les facteurs explicatifs de l'accord ou du refus d'une demande de crédit de clients auprès d'un établissement bancaire. Nous réalisons les opérations suivantes :

- Estimer les paramètres à l'aide de la régression logistique ;
- Récupérer la matrice de variance covariance pour implémenter différents tests de significativité de blocs de coefficients ;
- Evaluation de la régression à l'aide du test de Hosmer et Lemeshow ;
- Evaluation de la régression à l'aide du diagramme de fiabilité ;
- Evaluation de la régression à l'aide de la courbe ROC ;
- Analyse des résidus, détection des points atypiques et influents.

Nous utiliserons **Tanagra 1.4.33** dans un premier temps. Dans un deuxième temps, nous essaierons de reproduire les mêmes calculs à l'aide du **Logiciel R 2.9.1** (procédure **glm()**).

2 Données

Notre fichier « LOGISTIC_REGRESSION_DIAGNOSTICS.XLS² » comporte n = 100 observations. La variable à prédire est « ACCEPTATION.CREDIT » (« yes » ou « no »). Les variables prédictives sont

Nom	Description	Type
Age	Age du client	Quantitative
Income.Per.Dependent	Revenu par tête dans le ménage	Quantitative
Derogatory.Report	Au moins un problème avec l'établissement bancaire a été rapporté	Binaire

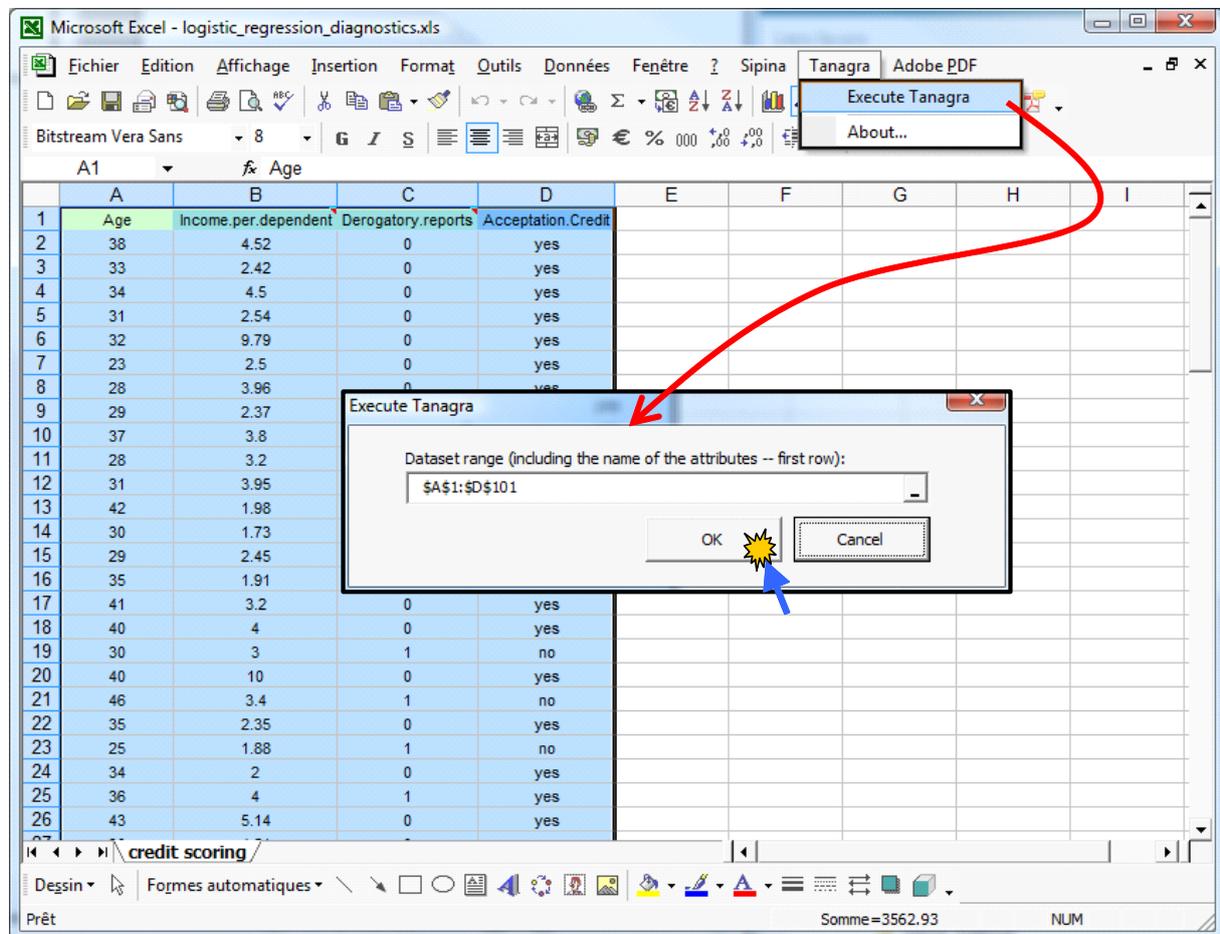
¹ http://eric.univ-lyon2.fr/~ricco/cours/cours/pratique_regression_logistique.pdf ; l'autre référence qu'il faut absolument avoir sous la main est bien entendu l'ouvrage de Hosmer et Lemeshow, « Applied Logistic Regression », John Wiley & Sons, Inc, Second Edition, 2000.

² http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/logistic_regression_diagnostics.zip

3 Analyse avec Tanagra

3.1 Importation des données

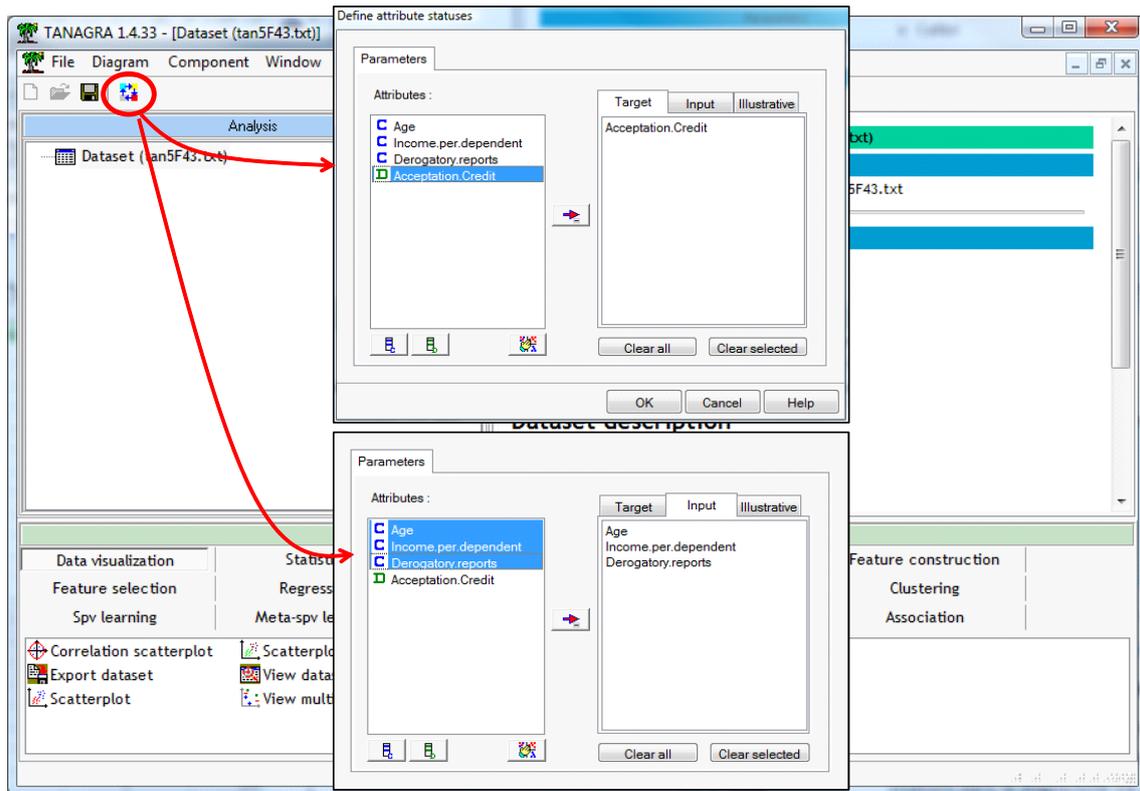
Première étape toujours dans tout logiciel de Data Mining, nous devons importer les données. Le plus simple avec Tanagra est d'ouvrir le fichier dans le tableur Excel, puis de lancer Tanagra via le menu installé à l'aide de la macro complémentaire TANAGRA.XLA³.



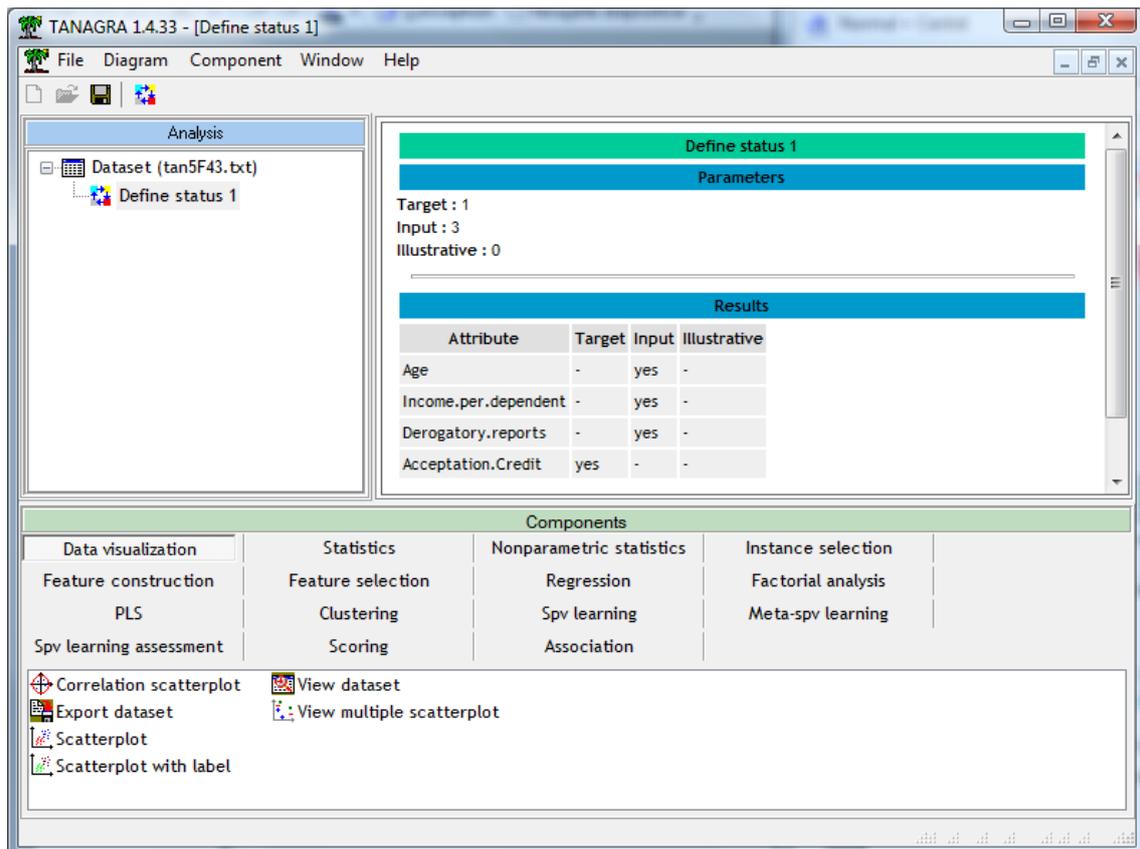
3.2 Définir le problème et lancer la régression logistique

Nous devons indiquer au logiciel la variable expliquée (ACCEPTATION.CREDIT → TARGET) et les variables explicatives (les autres → INPUT). Nous utilisons le composant DEFINE STATUS que nous insérons dans le diagramme via le raccourci dans la barre d'outils.

³ Voir <http://tutoriels-data-mining.blogspot.com/2008/03/importation-fichier-xls-excel-macro.html> concernant l'installation et l'utilisation de la macro complémentaire.



Nous validons, puis nous cliquons sur VIEW pour vérifier la sélection. Nous devrions obtenir l’affichage suivant.



Nous obtenons une p-value de 0.0011 avec la loi du KHI-2 à 3 degrés de liberté. Le modèle est donc globalement significatif au risque 5%.

R²-LIKE fournit les pseudo-R². Ils confrontent d'une manière ou d'une autre la vraisemblance du modèle étudié et du modèle trivial. Nous disposons de 3 indicateurs différents (Mc Fadden, Cox and Snell, Nagelkerke). Grosso modo, la régression ne vaut rien s'ils sont proches de 0 ; plus ils se rapprochent de 1, meilleur sera le modèle.

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 Chi ² test (LR)		
Chi-2	16.0094	
d.f.	3	
P(>Chi-2)	0.0011	
R ² -like		
McFadden's R ²	0.1372	
Cox and Snell's R ²	0.1479	
Nagelkerke's R ²	0.2149	

Nous disposons ensuite du tableau des coefficients (**Attributes in the Equation**). Pour chaque descripteur, y compris la constante, nous avons l'estimation de la valeur du coefficient, son écart-type, la statistique de Wald destinée à en évaluer sa significativité et la p-value s'y rapportant. Les probabilité critique en dessous de 0.05 sont surlignées en rouge vif ; celles en deçà de 0.10, en rouge moins vif (rose). Nous constatons par exemple que la variable AGE n'est pas significative à 5%, mais l'est à 10%. INCOME.PER.DEPENDANT en revanche n'es pas significative du tout, même à 10%. La variable la plus pertinente pour expliquer l'acceptation du dossier d'un client semble être DEROGATORY.REPORTS.

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

Toutes choses égales par ailleurs c.-à-d. à âge égal et à revenu par tête égal, les clients ayant rencontrés au moins une fois des problèmes avec leur banque ont $1/\exp(-1.929304) = 1/0.1452 =$

6.89 fois plus de chances de se voir refuser leur crédit (que ceux qui n'ont jamais eu de problèmes). Ces interprétations sous forme d'**ODDS RATIO** sont accessibles dans la dernière section du rapport. Nous avons également les **intervalles de confiance au niveau 95%**.

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

3.3 Tester la significativité d'un groupe de coefficients

Nous avons besoin de la matrice de variance covariance des coefficients pour évaluer la significativité simultanée d'un groupe de coefficients à l'aide du test de Wald. Tanagra 1.4.33 inclut un second onglet (COVARIANCE MATRIX) qui fournit ces valeurs.

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

Nous pouvons les copier dans EXCEL en actionnant le menu COMPONENT / COPY RESULTS⁴.

	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							
8							

Nous souhaitons tester « $H_0 : a(\text{AGE}) = a(\text{INCOME.PER.DEPENDENT}) = 0$ » par exemple. En piochant dans la matrice de variance covariance, nous pouvons former la statistique de Wald

$$\begin{aligned}
 W &= \begin{pmatrix} -0.062411 & 0.216797 \end{pmatrix} \begin{pmatrix} 1.13 \times 10^{-3} & -1.58 \times 10^{-3} \\ -1.58 \times 10^{-3} & 3.22 \times 10^{-2} \end{pmatrix}^{-1} \begin{pmatrix} -0.062411 \\ 0.216797 \end{pmatrix} \\
 &= \begin{pmatrix} -0.062411 & 0.216797 \end{pmatrix} \begin{pmatrix} 952.61 & 46.85 \\ 46.85 & 33.34 \end{pmatrix} \begin{pmatrix} -0.062411 \\ 0.216797 \end{pmatrix} \\
 &= 4.0097
 \end{aligned}$$

⁴ Nous utilisons sciemment le format scientifique pour conserver un maximum de précision.

Avec un KHI-2 à 2 degrés de liberté, nous obtenons une probabilité critique de 0.1347. Au risque 5%, nos données sont compatibles avec l'hypothèse de nullité simultanée des coefficients de AGE et INCOME.PER.DEPENDENT.

3.4 Test de Hosmer et Lemeshow

Le test de Hosmer et Lemeshow cherche à établir l'adéquation du modèle avec les données. Il se substitue au test basé sur le résidu déviance lorsque nous sommes dans une situation de données individuelles, par exemple lorsque nous avons des variables explicatives continues (quantitatives) comme c'est le cas pour nous⁵.

Nous branchons le composant HOSMER LEMESHOW TEST (onglet SPV LEARNING ASSESSMENT) à la suite de la régression. C'est le seul endroit où nous pouvons l'insérer dans le diagramme de toute manière.

Hosmer Lemeshow Goodness-of-Fit Test

Decile	Prob.	Positive		Negative		Total
		Observed	Expected	Observed	Expected	
1	0.417	3.000	2.894	7.000	7.106	10.000
2	0.600	4.000	5.287	6.000	4.713	10.000
3	0.726	7.000	6.835	3.000	3.165	10.000
4	0.764	7.000	7.512	3.000	2.488	10.000
5	0.796	8.000	7.817	2.000	2.183	10.000
6	0.823	9.000	8.096	1.000	1.904	10.000
7	0.843	8.000	8.365	2.000	1.635	10.000
8	0.859	9.000	8.516	1.000	1.484	10.000
9	0.870	10.000	8.643	0.000	1.357	10.000
10	0.946	8.000	9.035	2.000	0.965	10.000

Hosmer Lemeshow Statistic

	Chi-Square	d.f.	Significance
Goodness-of-fit test	4.4530	8	0.8141

Nous cliquons sur VIEW, nous obtenons le tableau des fréquences observées et théoriques. La statistique du test est égale à KHI-2 = 4.4530 avec une p-value = 0.8141.

Notre modèle est validé puisque la p-value est supérieure au risque de 5% prédéfini.

⁵ Dans le cas des données groupées, il vaut mieux utiliser la déviance calculée à partir du résidu déviance.

Remarque : Parce que nous travaillons sur un effectif relativement faible, de nombreuses cases du tableau des effectifs théoriques (espérés) sont inférieures à 5. La p-value calculée est sujette à caution. Il faudrait procéder à des regroupements.

Attention néanmoins, si nous parvenons à un tableau comportant moins de 6 lignes après regroupements, les auteurs du test ont remarqué que le modèle est toujours valide, quelle qu'en soit la pertinence (Hosmer et Lemeshow, 2000 ; page 151). Bref, tout ceci est à manipuler avec prudence.

3.5 Diagramme de fiabilité

Le diagramme de fiabilité cherche également à confronter les probabilités prédites par le modèle (les scores, en abscisse) et les probabilités observées (la proportion de positifs, en ordonnée) dans des groupes d'individus. Si le modèle est bien calibré, les points doivent former une droite.

Dans Tanagra, nous devons calculer explicitement les scores au préalable. En effet l'outil proposé est générique, il peut s'appliquer à tout classifieur capable de produire un score (une quantité proportionnelle à la probabilité d'être positif). Nous insérons donc le composant SCORE (onglet SCORING). Nous le paramétrons en spécifiant « YES » comme modalité « positive ».

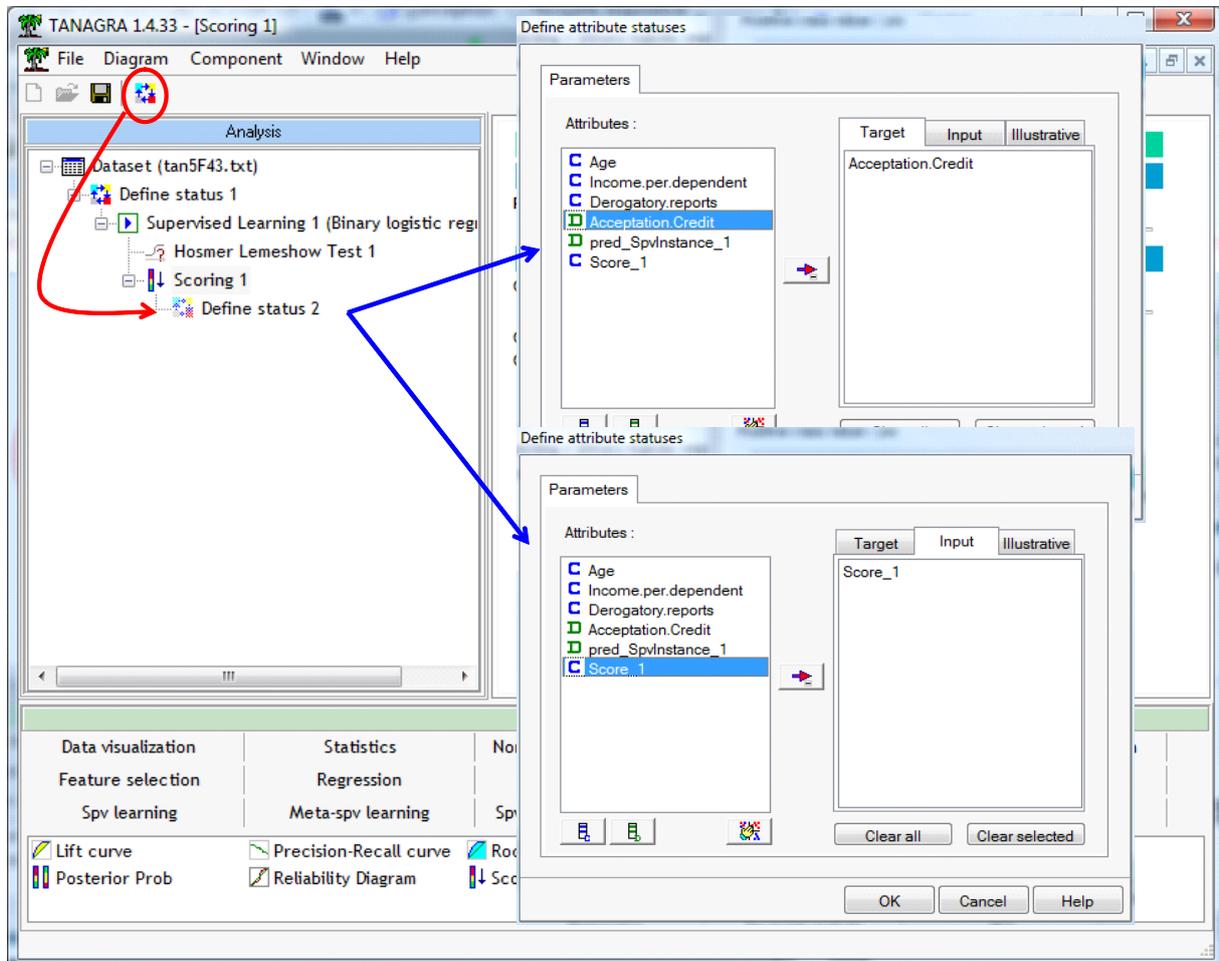
The screenshot shows the TANAGRA 1.4.33 interface. The main window displays the 'Analysis' tree on the left, where 'Scoring 1' is selected under 'Hosmer Lemeshow Test 1'. A context menu is open over 'Scoring 1', with 'Parameters...' selected. A red arrow points from this menu to the 'Parameters' tab of the 'Hosmer Lemeshow Goodness-of-Fit Test' dialog box. In this dialog, the 'Positive class value' is set to 'yes'. Below the dialog, the 'Hosmer Lemeshow Statistic' table is visible:

	Chi-Square	d.f.	Significance
Goodness-of-fit test	4.4530	8	0.8141

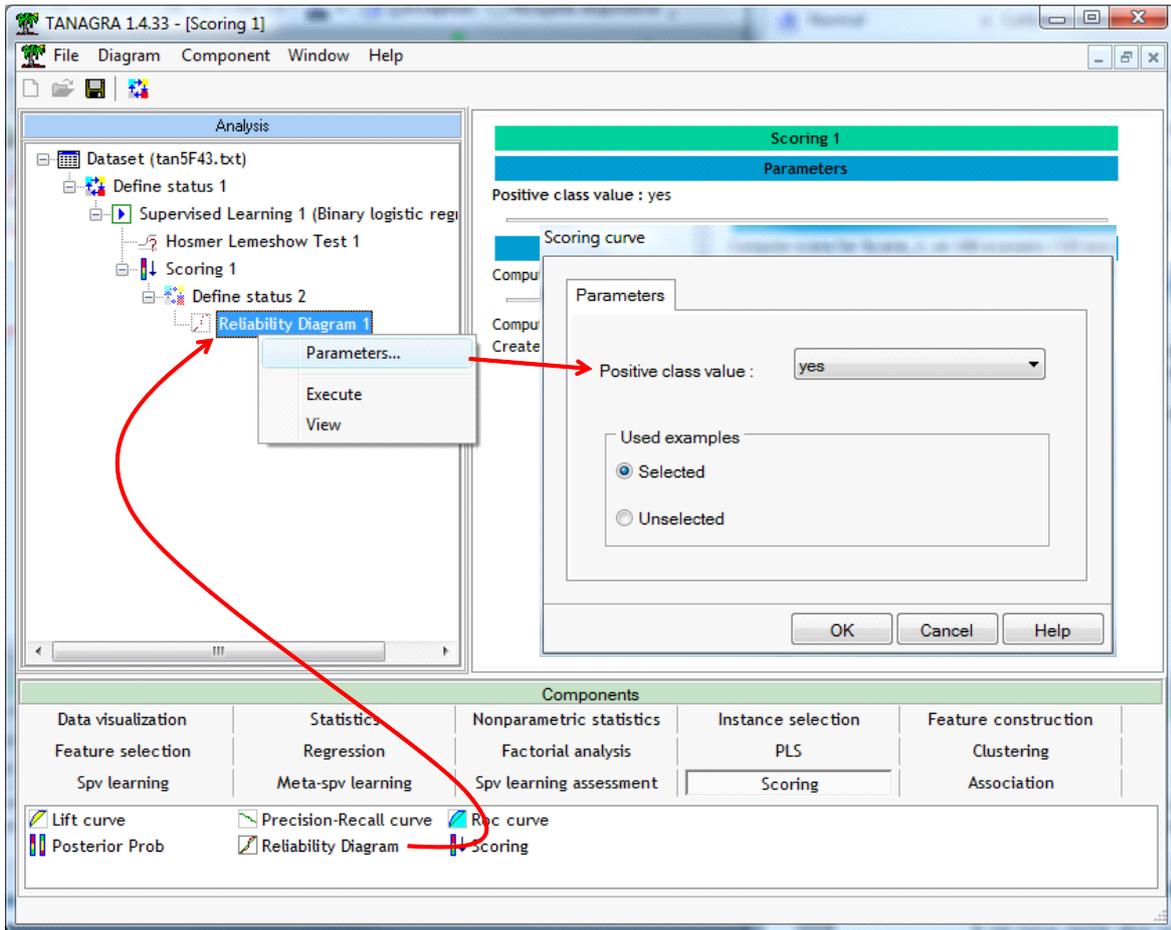
At the bottom of the interface, the 'Components' panel shows 'Scoring' selected in the 'Instance selection' category. A red arrow also points from the 'Scoring' component icon in this panel to the 'Parameters...' menu item.

Nous cliquons sur VIEW pour lancer les calculs. Une nouvelle colonne est ajoutée à l'ensemble de données.

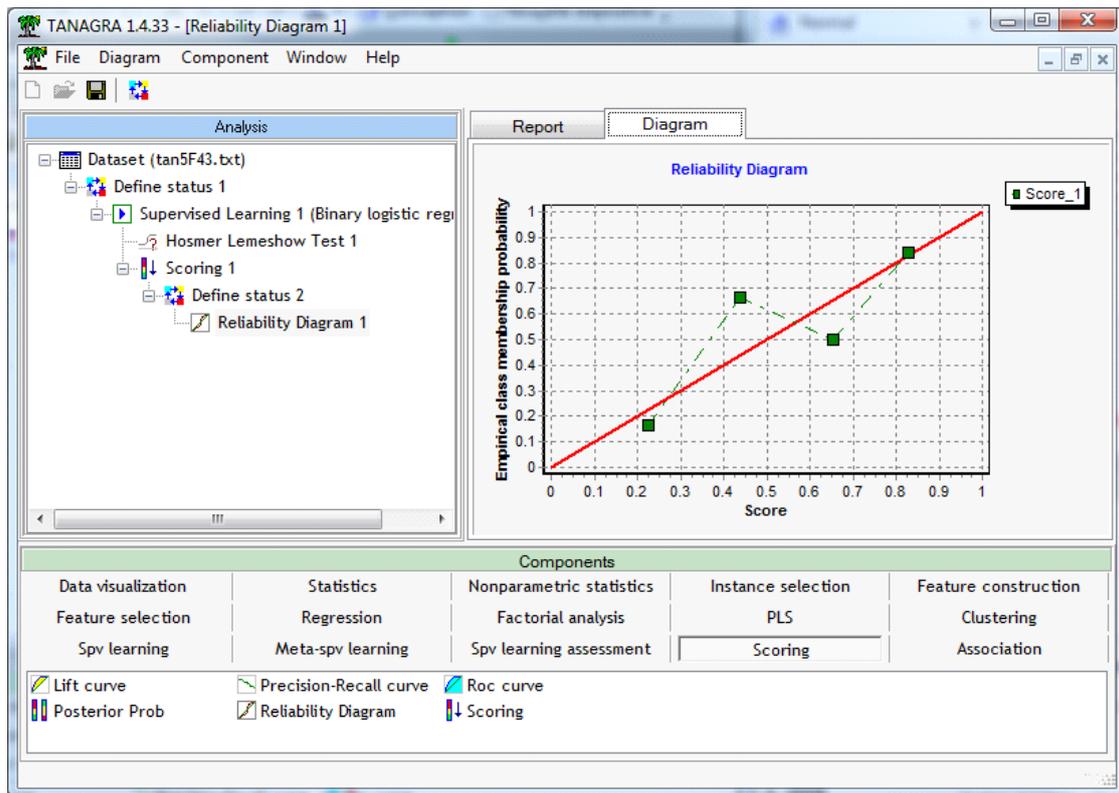
Nous insérons un nouveau composant DEFINE STATUS dans le diagramme. Nous mettons en TARGET la variable à prédire, en INPUT le score produit précédemment (SCORE_1). Notons que nous pouvons insérer plusieurs scores ici. Cela peut être utile pour les comparaisons de classifieurs. Autre remarque, n'importe quelle variable peut être utilisée comme score. Il suffit qu'il permette de trier les données selon leur « degré de positivité ».



Il ne nous reste plus qu'à introduire le composant RELIABILITY DIAGRAM (onglet SCORING). Nous le paramétrons en lui indiquant la modalité positive de la variable à prédire. Nous remarquons que si les données ont été préalablement subdivisées en échantillons d'apprentissage et de test, nous pouvons calculer le diagramme sur ce dernier.



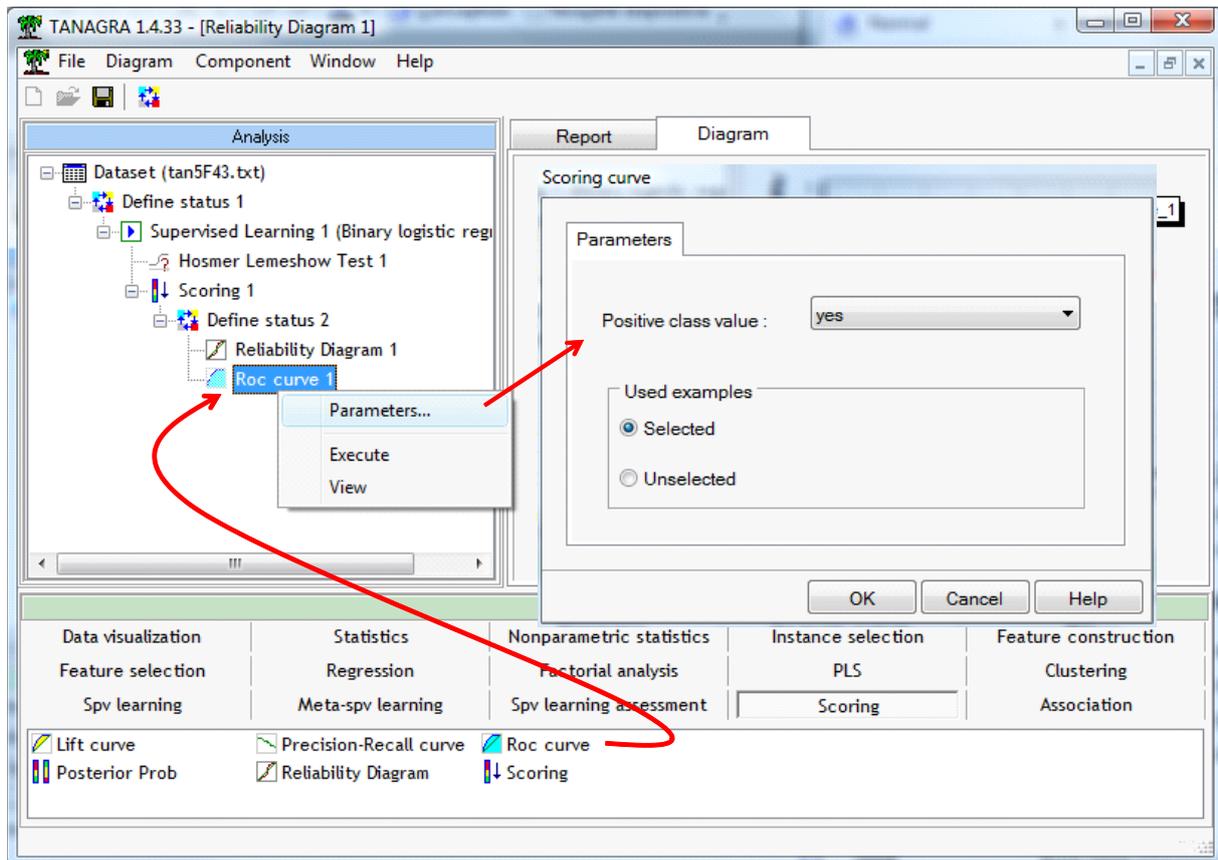
Nous cliquons sur VIEW.



Notre modèle n'est pas très bon. Les points associés aux groupes d'observations groupés (moyenne des scores en abscisse, proportion des positifs en ordonnée) ne sont pas très bien alignés. Il y a des zones où les scores sont surévalués (2nd point), et d'autres où ils sont sous-évalués (3^{ème} point).

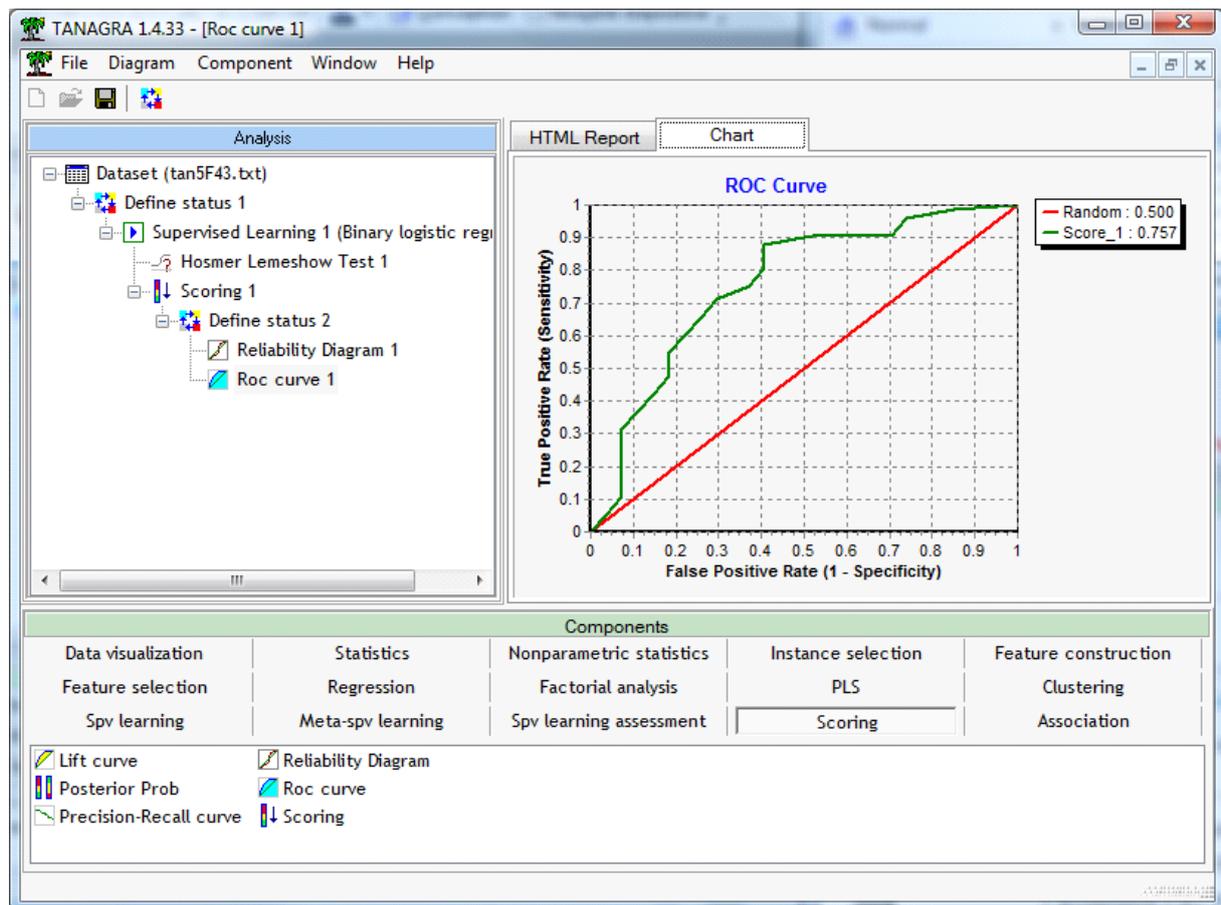
3.6 Courbe ROC

La courbe ROC évalue la capacité du modèle à placer les positifs devant les négatifs à partir des scores. Ces derniers ont déjà été calculés dans l'étape précédente. Il ne nous reste plus qu'à insérer le composant ROC CURVE (onglet SCORING) dans le diagramme. Nous le paramétrons en lui indiquant la modalité positive de la variable expliquée.



Puis nous cliquons sur VIEW.

Nous obtenons la courbe ROC. Elle est surtout intéressante lorsque l'on souhaite comparer des modèles. Plus intéressant dans notre contexte est le critère AUC (nous avons le détail des calculs dans l'onglet REPORT). Nous avons $AUC = 0.7575$. La discrimination fournie par le modèle est « acceptable » si l'on se réfère à l'interprétation usuelle (Hosmer et Lemeshow, 2000 ; page 162).



Bref, le modèle n'est pas très bon si l'on considère la majorité des indicateurs d'évaluation utilisés jusqu'à présent (matrice de confusion, diagramme de fiabilité, courbe ROC). Cet aspect est très important. Il ne faut pas se focaliser sur tel ou tel outil. L'intérêt d'en disposer de plusieurs est de pouvoir les croiser pour affermir les conclusions que l'on pourrait émettre sur la qualité du modèle.

3.7 Analyse des résidus

L'analyse des résidus est un outil important de la régression logistique, il permet de répondre à plusieurs questions. (1) Quels sont les points mal modélisés ? Si nous arrivons à les caractériser, nous saurons pourquoi certains profils sont mal restitués par le classifieur. (2) Quels sont les points atypiques, ceux dont les caractéristiques sont sensiblement différentes des autres ? (3) Quels sont les points influents, c.-à-d. si nous les retirions des données, les résultats obtenus seraient (significativement) différents.

Dans Tanagra, nous insérons le composant LOGISTIC REGRESSION RESIDUALS (onglet SPV LEARNING ASSESSMENT) à la suite de la régression logistique. Tout comme le test de Hosmer et Lemeshow, il ne peut pas être placé ailleurs. Nous cliquons sur VIEW.

The screenshot shows the TANAGRA 1.4.33 interface. The 'Residuals' report is displayed in a table format. The table has the following data:

	hat_logReg_1	pearson_logReg_1	std_pearson_logReg_1	difch
1	0.024189	0.508257	0.514518	
2	0.017029	0.545989	0.550698	
3	0.019377	0.449589	0.454010	
4	0.015076	0.506325	0.510186	
5	0.078039	0.238053	0.247923	
6	0.022197	0.396182	0.400653	
7	0.017131	0.395297	0.398727	
8	0.016699	0.484538	0.488636	
9	0.017845	0.532631	0.537448	
10	0.014700	0.429242	0.432432	
11	0.015140	0.434561	0.437889	
12	0.056369	-1.318656	-1.357471	
13	0.024359	-1.866338	-1.889493	
14	0.016119	0.480355	0.484273	
15	0.027664	0.614183	0.622859	

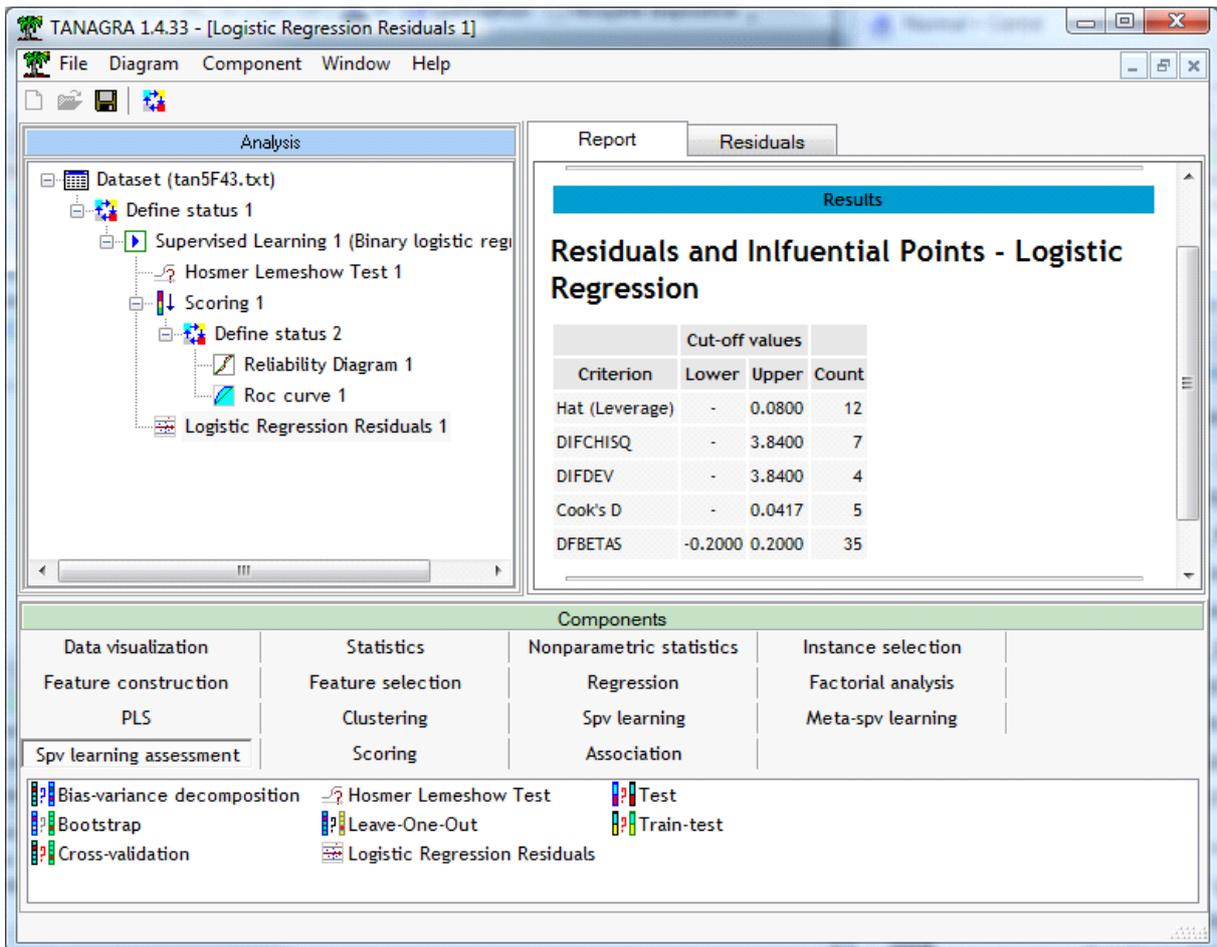
The 'Components' section at the bottom of the window lists various statistical and machine learning tools available in the software, including Bias-variance decomposition, Bootstrap, Cross-validation, Hosmer Lemeshow Test, Leave-One-Out, Logistic Regression Residuals, Test, Train-test, Factorial analysis, Spv learning assessment, Instance selection, PLS, Scoring, Feature construction, Clustering, and Association.

Les valeurs des indicateurs sont affichées dans une grille. Nous disposons pour chaque individu :

- HAT : le levier (Leverage est l'autre terme utilisé pour le désigner) ;
- PEARSON : résidu de Pearson ;
- STD_PEARSON : résidu de Pearson standardisé ;
- DIFCHISQ : contribution à la statistique de Pearson ;
- DEVIANCE : résidu déviance ;
- STD_DEVIANCE : résidu déviance standardisé ;
- DIFDEV : contribution à la déviance ;
- COOK : distance de Cook ;
- DEFBETA : DEFBETA pour chaque prédicteur, y compris la constante ;
- DEFBETAS : DEFBETA standardisé pour chaque prédicteur, y compris la variance.

Les valeurs « anormales », en deçà ou au-delà des seuils usuels décrits dans la littérature, sont signalées en gras. Mais plutôt que de se focaliser sur ces seuils, il est peut être plus intéressant de copier le tableau (COMPONENT / COPY RESULTS) et de les coller dans un tableur. En exploitant des fonctionnalités de TRI de ce dernier, nous pouvons mieux apprécier les groupes d'observations à problème.

Dans l'onglet REPORT de la fenêtre de résultats, nous avons un récapitulatif du nombre d'observations « suspectes » au regard de certains critères.



The screenshot shows the TANAGRA 1.4.33 interface. The 'Analysis' pane on the left shows a workflow: Dataset (tan5F43.txt) -> Define status 1 -> Supervised Learning 1 (Binary logistic regression) -> Hosmer Lemeshow Test 1 -> Scoring 1 -> Define status 2 -> Reliability Diagram 1 -> Roc curve 1 -> Logistic Regression Residuals 1. The 'Report' pane on the right shows the 'Residuals' report for 'Logistic Regression'.

Criterion	Cut-off values		Count
	Lower	Upper	
Hat (Leverage)	-	0.0800	12
DIFCHISQ	-	3.8400	7
DIFDEV	-	3.8400	4
Cook's D	-	0.0417	5
DFBETAS	-0.2000	0.2000	35

The 'Components' pane at the bottom lists various statistical methods: Data visualization, Feature construction, PLS, Spv learning assessment, Statistics, Feature selection, Clustering, Scoring, Nonparametric statistics, Regression, Spv learning, Association, Instance selection, Factorial analysis, and Meta-spv learning. A list of diagnostic tests is also shown: Bias-variance decomposition, Bootstrap, Cross-validation, Hosmer Lemeshow Test, Leave-One-Out, Logistic Regression Residuals, Test, and Train-test.

4 ANALYSE avec R

Nous ne détaillons pas toutes les opérations avec R, nous donnons les éléments qui permettent d'obtenir les résultats demandés. Le code source «LOGISTIC_REGRESSION_DIAGNOSTICS.R» est inclus dans l'archive distribué avec ce didacticiel.

4.1 Importation des données et régression logistique

Nous introduisons les commandes suivantes pour charger le fichier Excel et lancer la régression. Attention, il ne faut pas que le fichier soit en cours d'édition par ailleurs, cela ferait échouer l'importation.

Voici le code utilisé.

```

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\logistic regression\residuals\logistic_regressio...
#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)

```

Nous avons les résultats suivants. Ils concordent avec ceux de Tanagra.

```

R Console
> 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

```

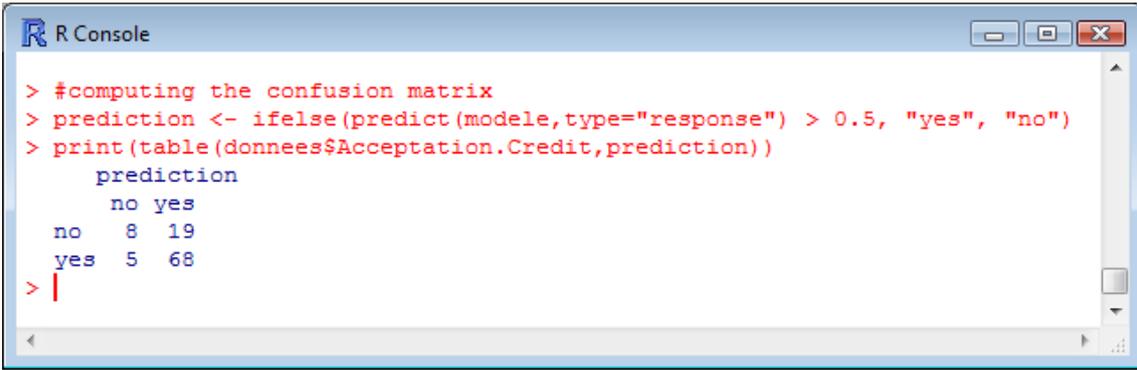
Pour obtenir la matrice de confusion, nous procédons en 2 temps : créer une colonne prédiction, puis confronter valeurs observées et valeurs prédites dans un tableau de contingence.

```

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\logistic regression\r...
#computing the confusion matrix
prediction <- ifelse(predict(modele, type="response") > 0.5, "yes", "no")
print(table(donnees$Acceptation.Credit, prediction))

```

Ainsi.



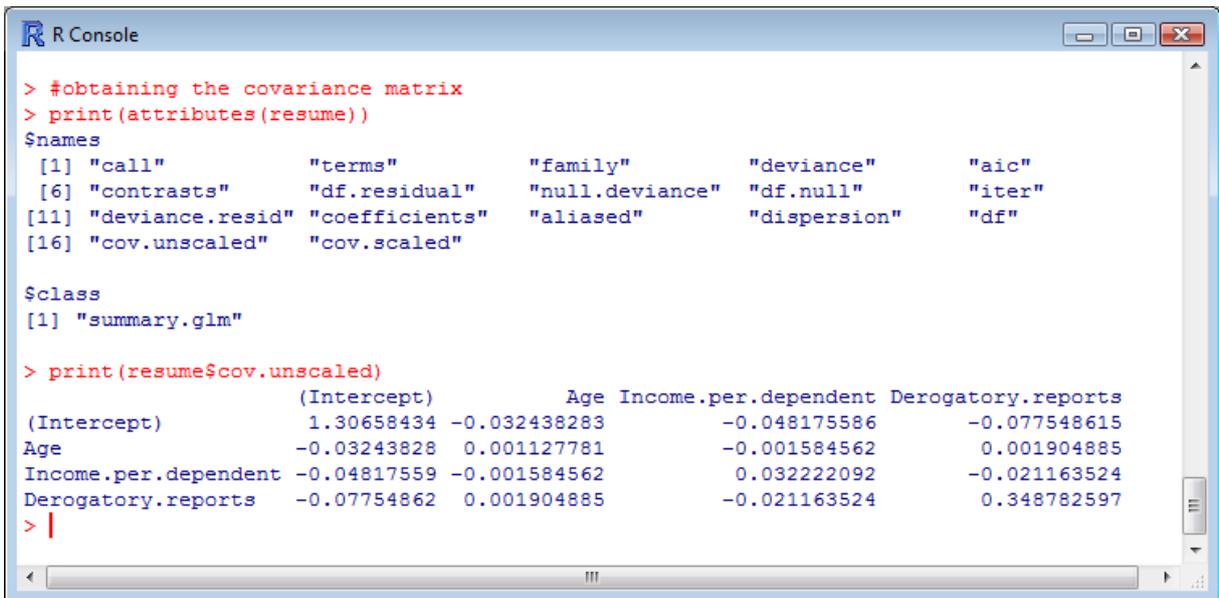
```

> #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
> |

```

4.2 Matrice de variance covariance des coefficients

Pour réaliser les différents tests de significativité, nous avons besoin de la matrice de variance covariance des coefficients. Elle est associée à l'objet « résumé » du modèle. Nous pouvons l'afficher de la manière suivante.



```

> #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"

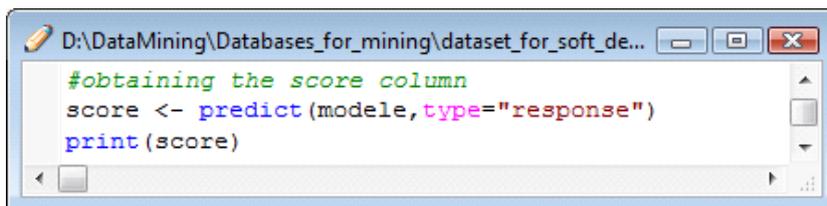
> print(resume$cov.unscaled)
      (Intercept)      Age Income.per.dependent Derogatory.reports
(Intercept)  1.30658434 -0.032438283      -0.048175586      -0.077548615
Age          -0.03243828  0.001127781      -0.001584562      0.001904885
Income.per.dependent -0.04817559 -0.001584562      0.032222092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

```

Pour les tests de significativité, R disposant d'opérateur matriciels puissants, nous pourrions les réaliser directement, sans avoir à passer par un tableur.

4.3 Obtenir les scores – Test de Hosmer et Lemeshow

Le test de Hosmer et Lemeshow, la courbe ROC et le diagramme de fiabilité reposent sur le score attribué aux individus. Cette étape est incontournable. Avec R, nous utilisons la fonction **predict()**.



```

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

```

Après, il ne reste qu'à exploiter adroitement cette colonne de valeurs pour produire l'indicateur demandé (graphique ou statistique de test). Nous nous en tiendrons à la statistique de Hosmer et

Lemeshow dans cette section. Le code utilisé est le suivant⁶, il correspond en tout point à la procédure implémentée dans Tanagra.

```

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
  #statistic
  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)))

```

Et nous obtenons les mêmes résultats bien entendu.

```

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

```

4.4 Résidus et points influents

R propose tout une panoplie d'outils pour produire les résidus et indicateurs d'influence.

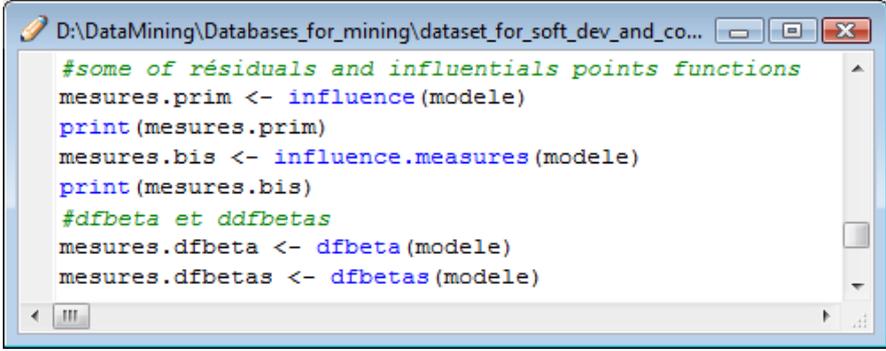
⁶ OK, OK, il y a sûrement des manières plus directes de programmer (plus dans la philosophie R). J'ai souhaité privilégier la clarté ici en décomposant chaque étape, d'où ce code très « scolaire ». Voici à titre de comparaison un programme trouvé sur internet. C'est plutôt joli car très concis. Mais si on veut faire fuir les « apprentis programmeurs » R, il n'y a pas mieux.

```

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\logistic regression\residuals\logistic_regression...
#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))

```

Les deux procédures convergent bien entendu, nous retrouvons le résultat fourni par Tanagra (ou l'inverse, bref tout va bien).

A screenshot of an R console window. The title bar shows the file path: "D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_co...". The console contains the following R code:

```
#some of residuals and influential 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() fournit les leviers (hat values), les DFBETA, les résidus déviance et de Pearson.

Influence.measures() fournit les DFBETAS, les DFFIT, les COVRATIO, les distances de Cook, et les leviers.

D'autres fonctions sont disponibles. On notera que les résultats sont identiques à ceux de Tanagra, à l'exception notable des DFBETA (et par conséquent des DFBETAS). Malgré mes recherches, je n'ai pas pu détecter l'origine de la différence. Les valeurs fournies par Tanagra sont identiques à celles de SAS et SPSS. Ca me rassure un peu. J'aurais aimé cependant comprendre la formule utilisée dans R. Des indications sur le sujet m'intéresseraient.

5 Conclusion

Nous avons essayé de donner un aperçu élargi des outils utilisés pour évaluer et diagnostiquer la régression logistique dans ce didacticiel. Certains sont spécifiques à la régression (test de Hosmer-Lemeshow, analyse des résidus) ; d'autres sont plus génériques, ils peuvent être utilisés pour tout classifieur sachant fournir une prédiction ou un score (matrice de confusion, courbe ROC).