1 Objectif

Traitement des classes déséquilibrées en régression logistique.

Bien souvent, les modalités de la variable à prédire ne sont pas également représentées en apprentissage supervisé. Si l'on s'en tient aux problèmes à deux classes, les positifs, ceux que l'on cherche à identifier justement, sont rares par rapport aux négatifs : les personnes malades sont (heureusement) peu nombreux par rapport aux personnes en bonne santé ; les fraudeurs constituent une infime minorité dans la population ; etc. Dans cette configuration, en travaillant avec un échantillon représentatif et en évaluant le modèle de prédiction avec la procédure usuelle (matrice de confusion + taux d'erreur), on se rend compte que le meilleur classifieur revient à prédire systématiquement la classe majoritaire (les négatifs), ce qui nous assure le plus faible taux d'erreur.

Par exemple, dans le fichier que nous traiterons dans ce didacticiel, les positifs représentent 1.77% de la population. En classant tous les individus « négatifs », nous sommes assurés d'obtenir un taux d'erreur de 1.77%. Cela paraît trop simpliste pour la majorité des praticiens : le meilleur modèle serait « pas de modèle », ça paraît difficile à défendre face à des personnes qui ne sont pas au fait des arcanes des principes bayesiens de l'apprentissage supervisé.

La stratégie la plus couramment admise pour surmonter cet écueil consiste à équilibrer artificiellement les données c.-à-d. mettre autant de positifs que de négatifs dans l'échantillon d'apprentissage. Sans que l'on ne sache pas très bien pourquoi, sans mettre en relation cette modification avec les caractéristiques de la technique d'apprentissage, sans en mesurer les conséquences sur le comportement du classifieur induit. Tout simplement parce que « c'est mieux ». La recette miracle en quelque sorte. En tous les cas, les ressources étant limitées, cela ne peut se faire que de deux manières : dupliquer les positifs (sur échantillonnage) ou bien n'utiliser qu'une fraction des négatifs (sous échantillonnage).

Dans ce didacticiel, nous souhaitons évaluer le comportement du sous échantillonnage lors du traitement d'une base très déséquilibrée à l'aide de la régression logistique. Pour ce faire, nous mettons en place le protocole d'évaluation suivant :

- Nous utilisons un échantillon d'apprentissage représentatif (900 observations, dont 15 positifs) pour construire le classifieur **M1**. Nous l'appliquons sur un échantillon test, représentatif également, de 3000 observations. Deux critères seront utilisés pour évaluer le modèle : (1) son aptitude à attribuer un score plus élevé aux positifs par rapport aux négatifs, la courbe ROC est tout à fait indiquée pour cela ; (2) son aptitude à classer correctement les individus, le tandem matrice de confusion – taux d'erreur convient bien pour cela.
- Dans un deuxième temps, nous équilibrons volontairement d'échantillon d'apprentissage de manière à avoir autant de positifs que de négatifs. Nous disposons donc de 30 observations pour construire le modèle de prédiction **M2** que nous appliquons sur le même échantillon test représentatif (3000 observations). Ainsi les performances sont directement comparables. Nous verrons ce qu'il en est en ce qui concerne nos deux critères d'évaluation.

L'intérêt de ce protocole est que nous disposons d'un échantillon test représentatif pour l'évaluation. Dans la pratique, ce n'est pas toujours possible, l'échantillon test n'est pas représentatif

non plus. Il faut dans ce cas corriger les ratios obtenus (taux d'erreur, précision) pour obtenir une évaluation non faussée du véritable comportement du modèle de prédiction[1]. Négliger cet aspect est souvent à l'origine de la fausse opinion qui accompagne la procédure d'équilibrage des données lors du traitement des classes déséquilibrées.

2 Données

Notre jeu de données comporte 3900 observations. La variable à prédire est « objective ». Les positifs sont très rares (69 observations). Nous disposons de 6 variables prédictives continues (V1 à V6). Nous avons scindé aléatoirement (sans stratification) les données en échantillons d'apprentissage (900 observations) et de test (3000), avec les caractéristiques suivantes :

Nombre de objective	status1		
objective	train	test	Total
positive	15	54	69
negative	885	2946	3831
Total	900	3000	3900

Nous avons introduit une première colonne « status1 » pour distinguer le rôle des individus dans l'expérimentation. Dans l'échantillon d'apprentissage de 900 observations, nous avons 15 positifs et 885 négatifs ; concernant l'échantillon test de 3000 observations, nous disposons de 54 positifs et 2946 négatifs. Échantillons d'apprentissage et de test sont tous deux représentatifs ici.

Une seconde colonne « status2 » sert à spécifier l'échantillon d'apprentissage équilibré de 30 observations. Il sera utilisé pour construire le second modèle de prédiction qui sera ensuite évalué sur l'échantillon test représentatif de 3000 observations.

Nombre de objective	status2		
objective	train	neglect	Total
positive	15	54	69
negative	15	3816	3831
Total	30	3870	3900

Précision très importante, les 30 individus qui constituent cet échantillon équilibré sont bien extraits de l'échantillon d'apprentissage représentatif de 900 observations comme nous pouvons le constater dans le tableau croisé suivant.

Nombre de status1	status2		
status1	train	neglect	Total
train	30	870	900
test		3000	3000
Total	30	3870	3900

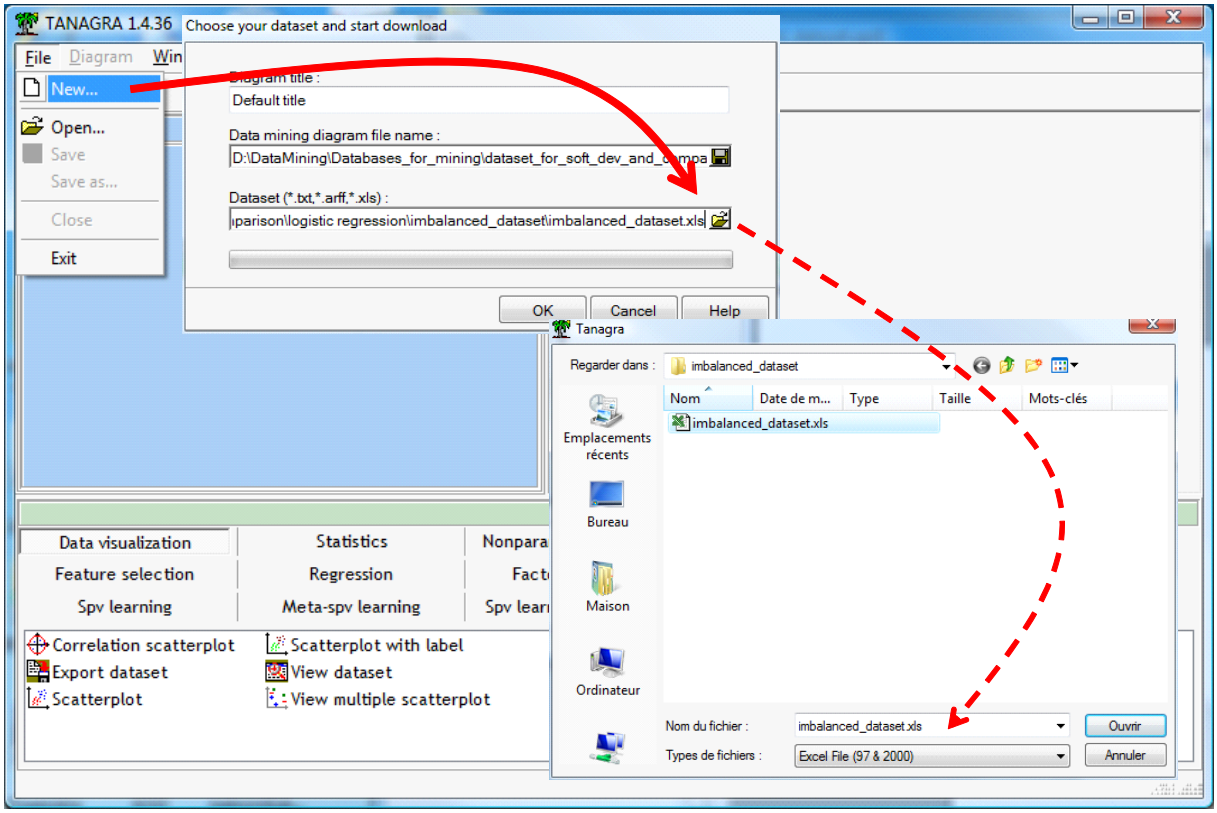
En effet, nous ne pouvons exploiter que des observations en provenance de l'échantillon d'apprentissage initial (900 observations) pour constituer le fichier d'apprentissage équilibré.

[1] Voir R. Rakotomalala, « Pratique de la régression logistique – Régression logistique binaire et polytomique – Version 2.0 », section 10.1 « Redressement pour les échantillons non représentatifs », pages 177 à 185, 2009.

3 Modéliser avec l'échantillon représentatif (M1)

3.1 Création d'un diagramme et importation des données

Après avoir démarré Tanagra, nous créons un nouveau diagramme (FILE / NEW) et nous sélectionnons le fichier « imbalanced_dataset.xls[2] » au format XLS. Attention, pour que l'importation fonctionne, le fichier ne doit pas être en cours d'édition dans le tableur Excel[3].



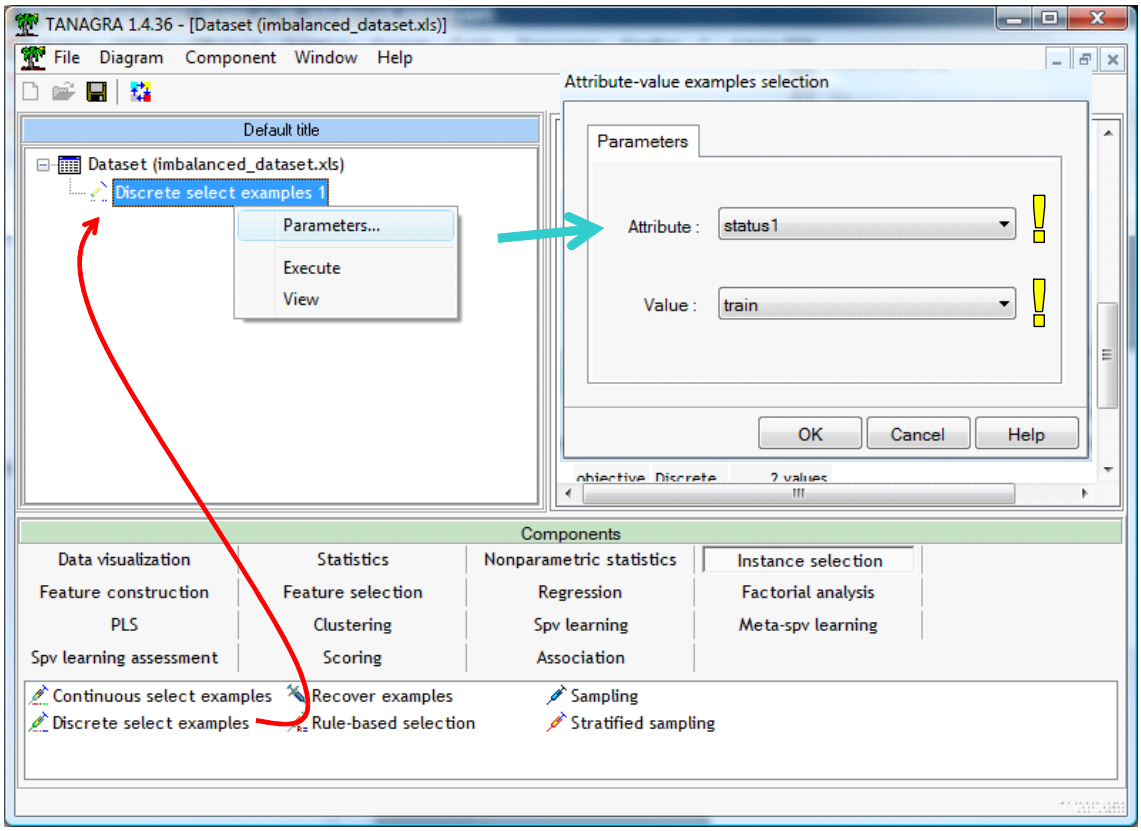
3900 observations et 9 colonnes sont chargées.

3.2 Constitution de l'échantillon d'apprentissage

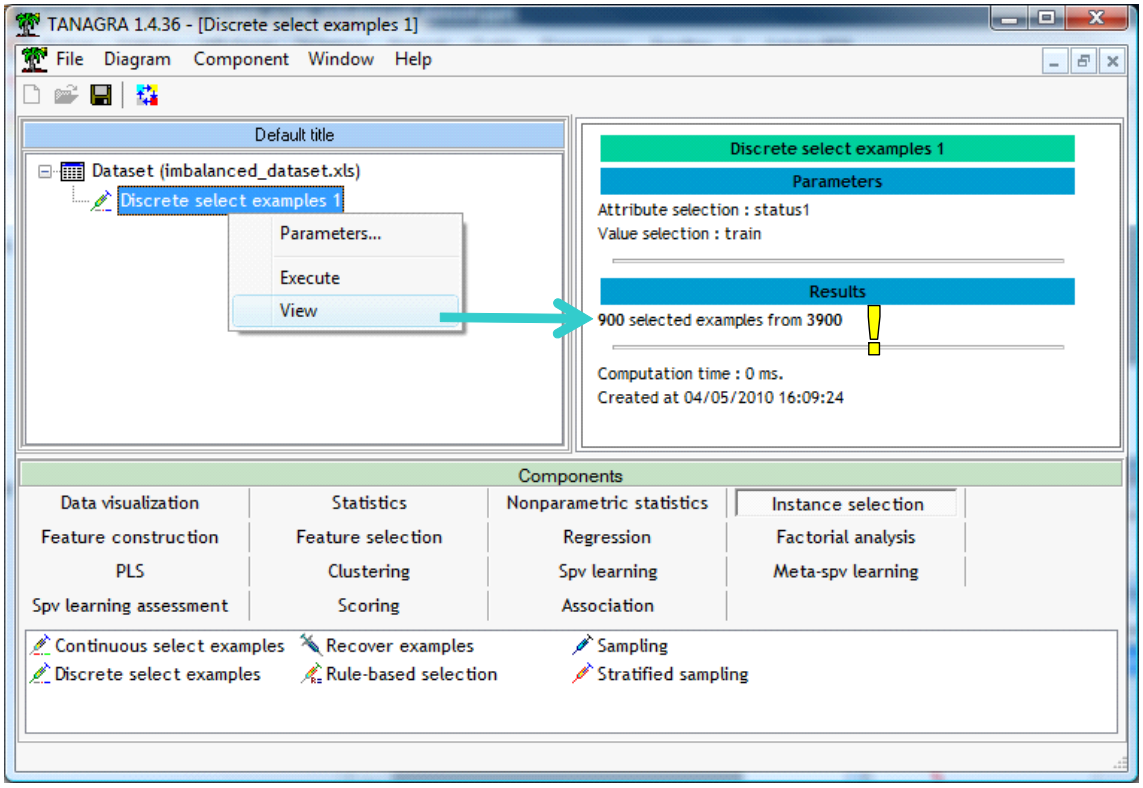
Première étape, nous devons sélectionner les observations dédiées à la construction du modèle. Nous insérons le composant DISCRETE SELECT EXAMPLES (onglet INSTANCE SELECTION). Nous le paramétrons (menu PARAMETERS) de manière à sélectionner les individus correspondant à « STATUS1 = TRAIN ».

[2] http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/imbalanced_dataset.xls

[3] Voir http://tutoriels-data-mining.blogspot.com/2008/03/importation-fichier-xls-excel-mode.html pour plus de détails. Il est également possible d'intégrer Tanagra dans Excel via une macro complémentaire, cf. http://tutoriels-data-mining.blogspot.com/2008/03/importation-fichier-xls-excel-macro.html

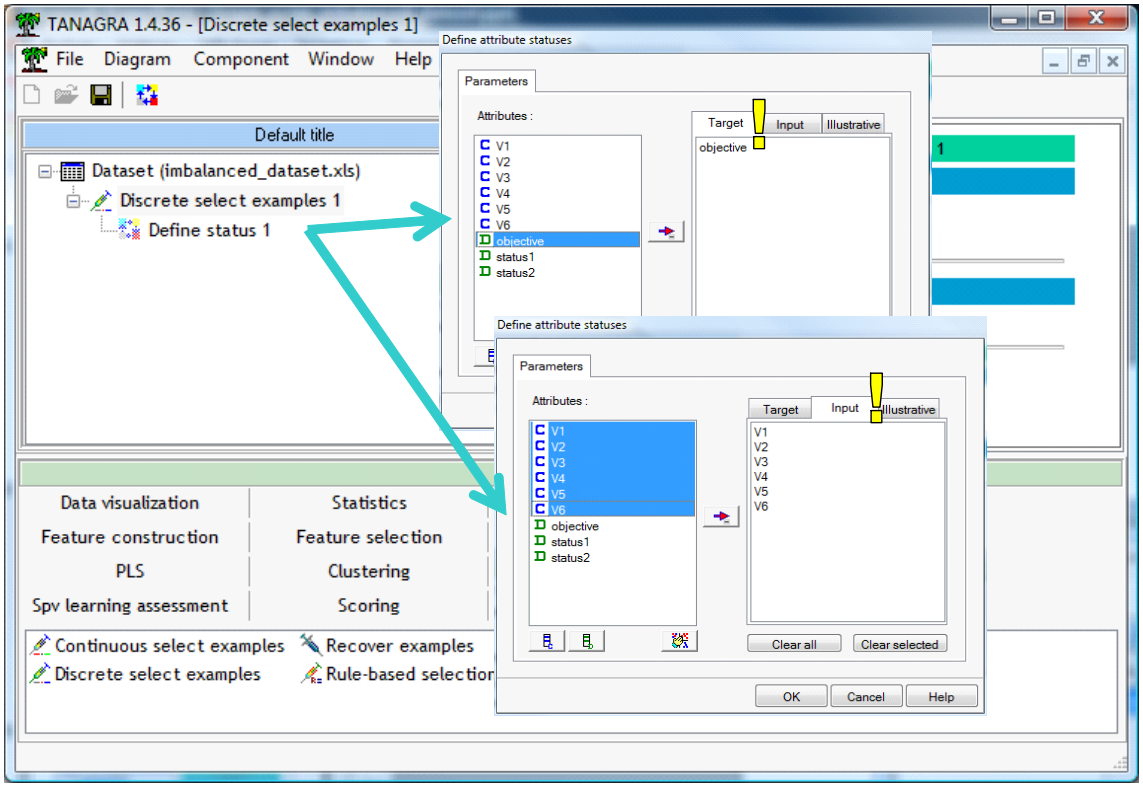


Nous cliquons sur VIEW, Tanagra nous indique que 900 observations parmi les 3900 sont sélectionnées pour la suite des opérations.

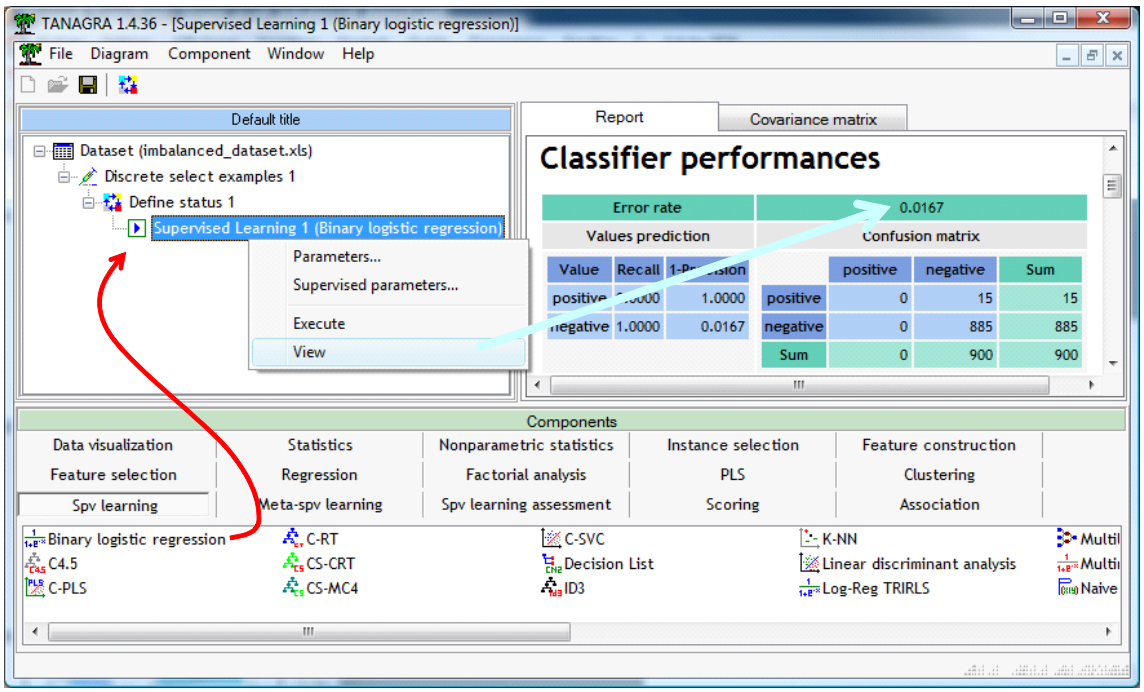


3.3 Construction du modèle prédictif

Nous utilisons le composant DEFINE STATUS (raccourci dans la barre d'outils) pour spécifier le rôle des variables : OBJECTIVE est la variable cible (TARGET), les variables V1 à V6 correspondent aux variables prédictives (INPUT).

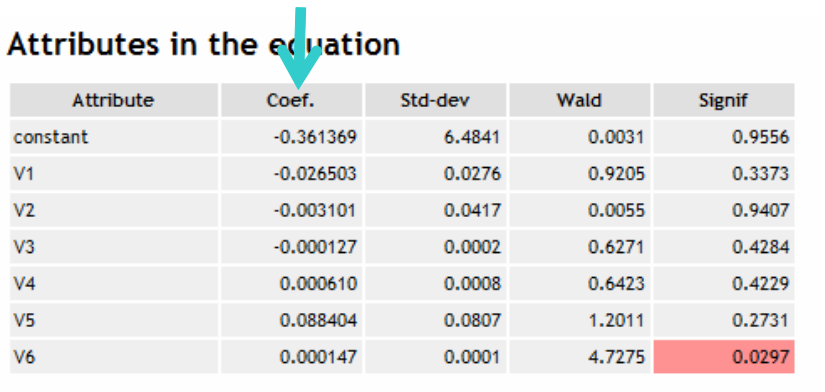


Il ne reste plus qu'à ajouter le composant BINARY LOGISTIC REGRESSION (onglet SPV LEARNING) pour la construction du modèle. Nous actionnons le menu contextuel VIEW.



La matrice de confusion nous indique que le modèle prédit systématiquement « négatif » sur l'échantillon d'apprentissage. Avec un taux d'erreur de 1.67% qui semble faible mais qui ne rime à rien en réalité. En effet, le modèle par défaut, celui qui n'exploite en aucune manière les informations prodiguées par les variables prédictives, aurait abouti à la même conclusion.

Et pourtant, il n'y a pas eu d'erreur, le processus de modélisation a fonctionné, nous disposons plus bas dans la fenêtre de visualisation des coefficients de la régression logistique.



Attributes in the equation

Attribute	Coef.	Std-dev	Wald	Signif
constant	-0.361369	6.4841	0.0031	0.9556
V1	-0.026503	0.0276	0.9205	0.3373
V2	-0.003101	0.0417	0.0055	0.9407
V3	-0.000127	0.0002	0.6271	0.4284
V4	0.000610	0.0008	0.6423	0.4229
V5	0.088404	0.0807	1.2011	0.2731
V6	0.000147	0.0001	4.7275	0.0297

A priori, ce résultat peut paraître décevant. Les corrections introduites par les praticiens visent justement à corriger ce comportement « anormal ». On aimerait pouvoir reconnaître au moins quelques positifs.

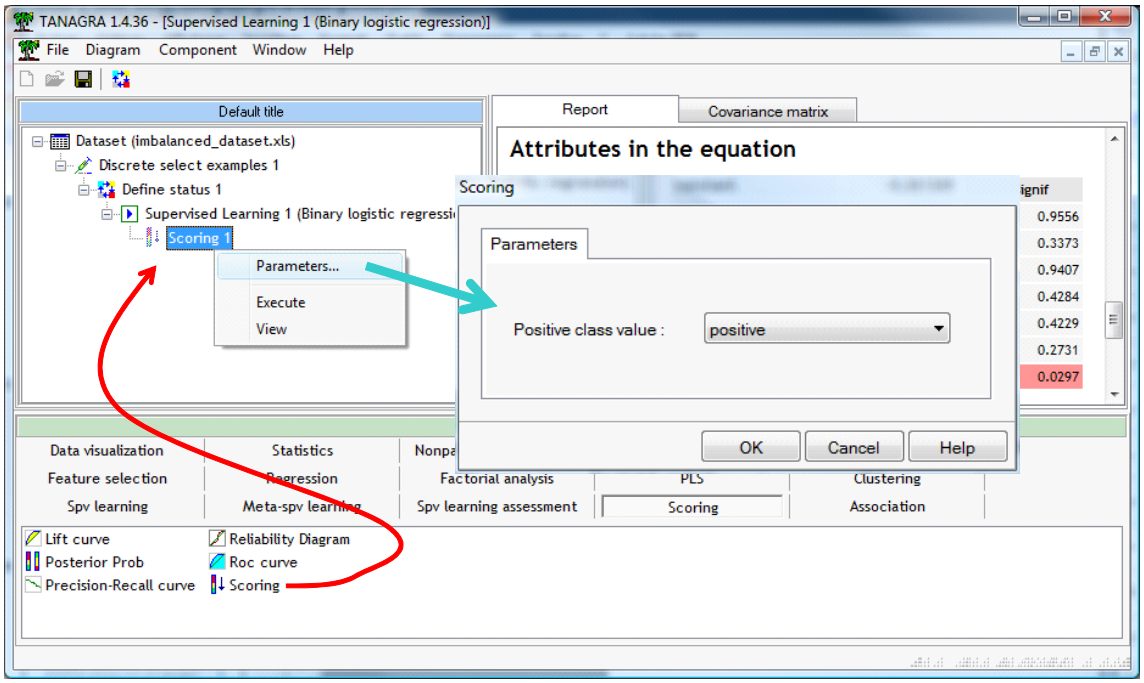
Mais est-ce que ce comportement est si anormal que cela ? Les choses ne sont pas aussi simples. Le modèle s'appuie sur un score (la probabilité d'être positif) pour attribuer la classe d'appartenance. On considère que le modèle est efficace s'il est capable d'attribuer un score en moyenne plus élevé aux positifs par rapport aux négatifs. Nous privilégierons donc la courbe ROC plutôt que la matrice de confusion (et indicateurs associés[4]) pour mesurer la performance du modèle[5]. Après, tout est affaire de seuil d'affectation que l'on peut corriger à souhait selon que l'on souhaite privilégier la sensibilité ou la précision[6].

Nous introduisons le composant SCORING (onglet SCORING) pour calculer les scores de « positivité » des observations. Nous le paramétrons comme suit.

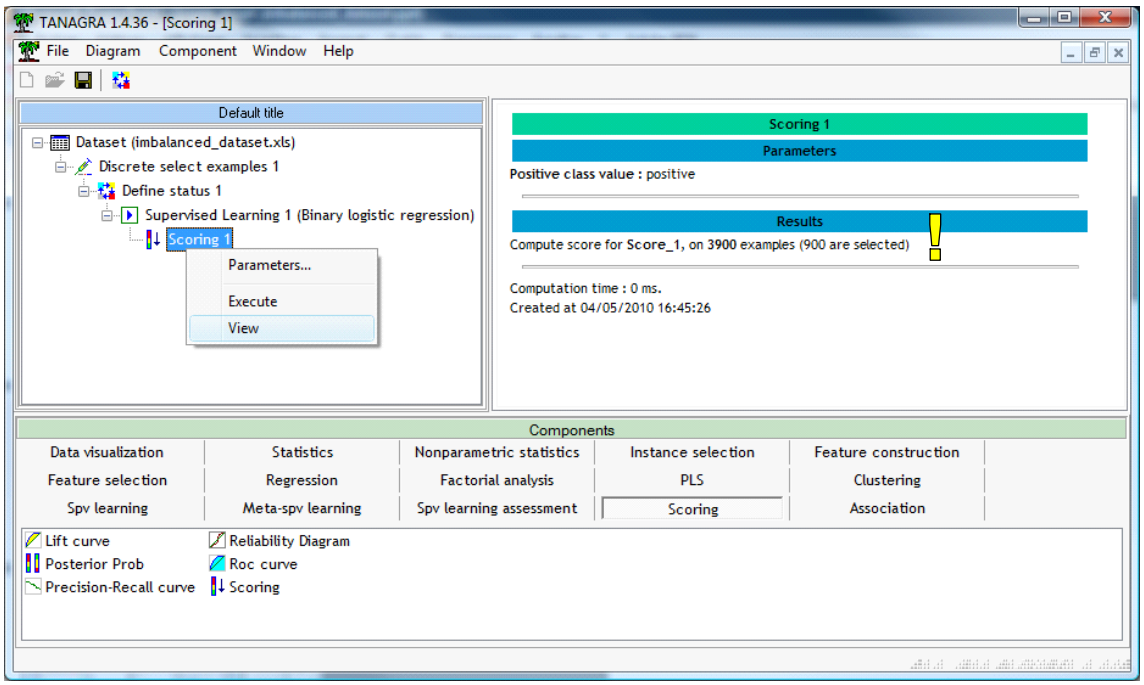
[4] Voir http://eric.univ-lyon2.fr/~ricco/cours/slides/Apprentissage_Supervise.pdf, page 10.

[5] Les avantages liés à l'utilisation de la courbe ROC pour l'évaluation des modèles sont multiples, voir http://eric.univ-lyon2.fr/~ricco/cours/slides/roc_curve.pdf

[6] Le modèle par défaut justement aurait attribué le même « score » à toutes les observations : la probabilité d'être négatif dans la population. De fait, son pouvoir discriminatoire est très mauvais car il attribue un score identique aux positifs et aux négatifs de la base.



Remarque très importante : si le modèle a bien été construit sur 900 individus, le composant calcule le score pour la totalité de la base. Les 3000 non sélectionnés sont donc considérés comme des « individus supplémentaires ».

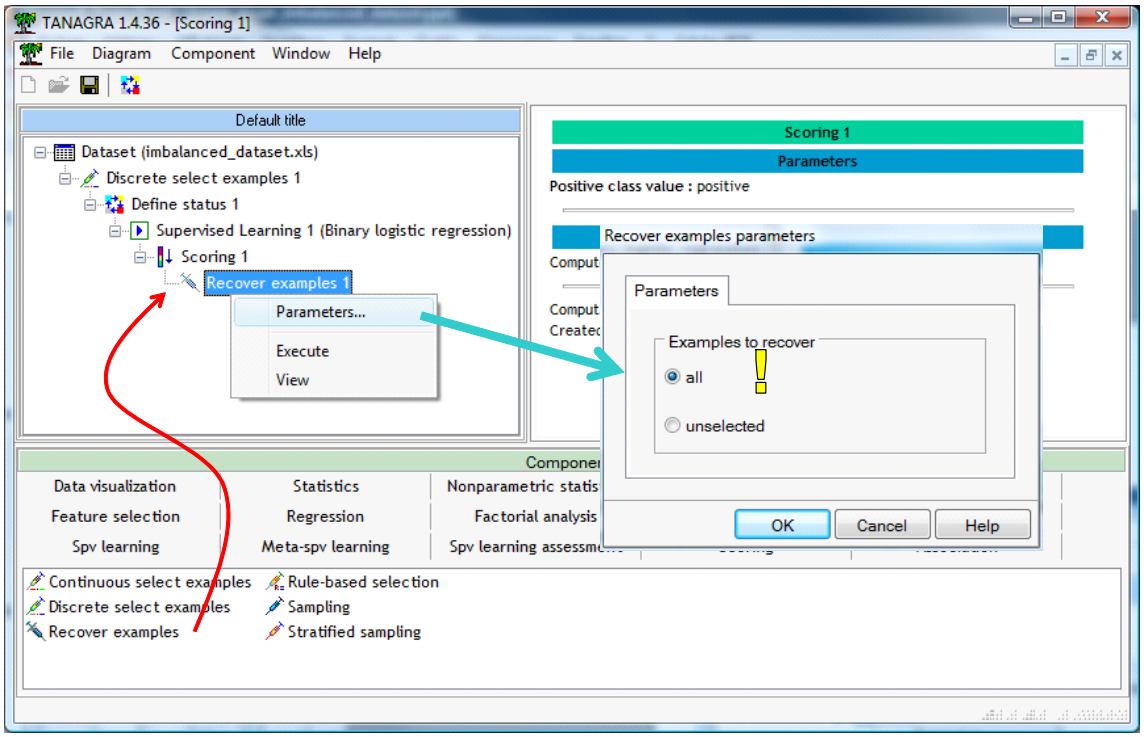


3.4 Évaluation sur l'échantillon test représentatif

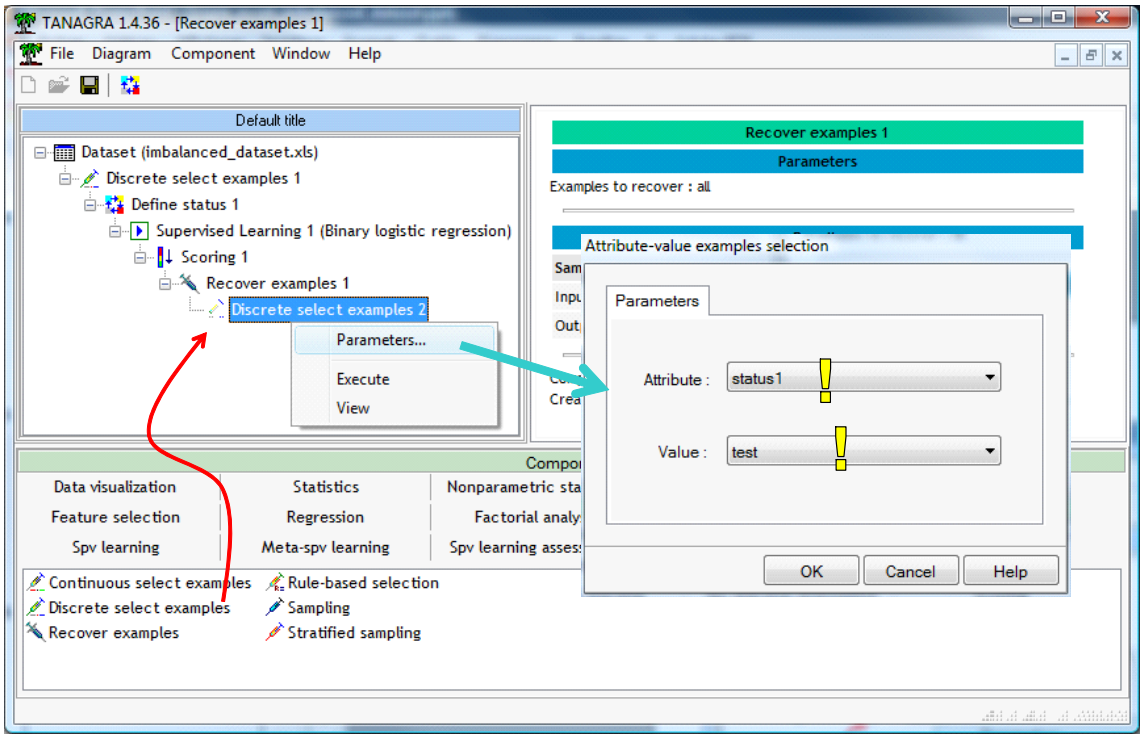
Nous allons utiliser l'échantillon test représentatif de 3000 observations pour évaluer le modèle. Il faut les activer pour que Tanagra puisse les exploiter. Deux étapes sont nécessaires[7].

[7] Il est possible de faire plus simple. Nous détaillons la procédure pour qu'elle soit facilement reproductible.

Dans un premier temps, nous réactivons toutes les observations. Nous introduisons le composant RECOVER EXAMPLES (onglet INSTANCE SELECTION) que nous paramétrons comme suit.



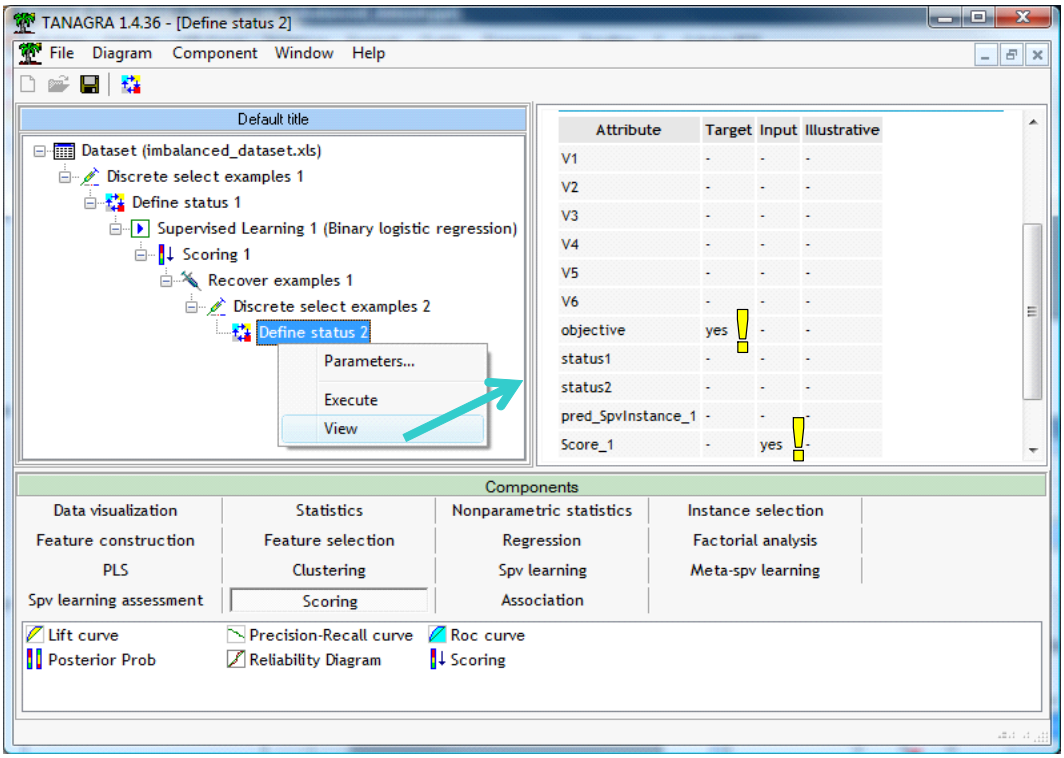
Il nous annonce que les 3900 observations sont maintenant actives. Nous devons dans un second temps procéder à une sélection où les observations actives correspondent maintenant à l'échantillon test « STATUS1 = TEST ». Pour ce faire, nous ajoutons le composant DISCRETE SELECT EXAMPLES avec le paramétrage suivant.



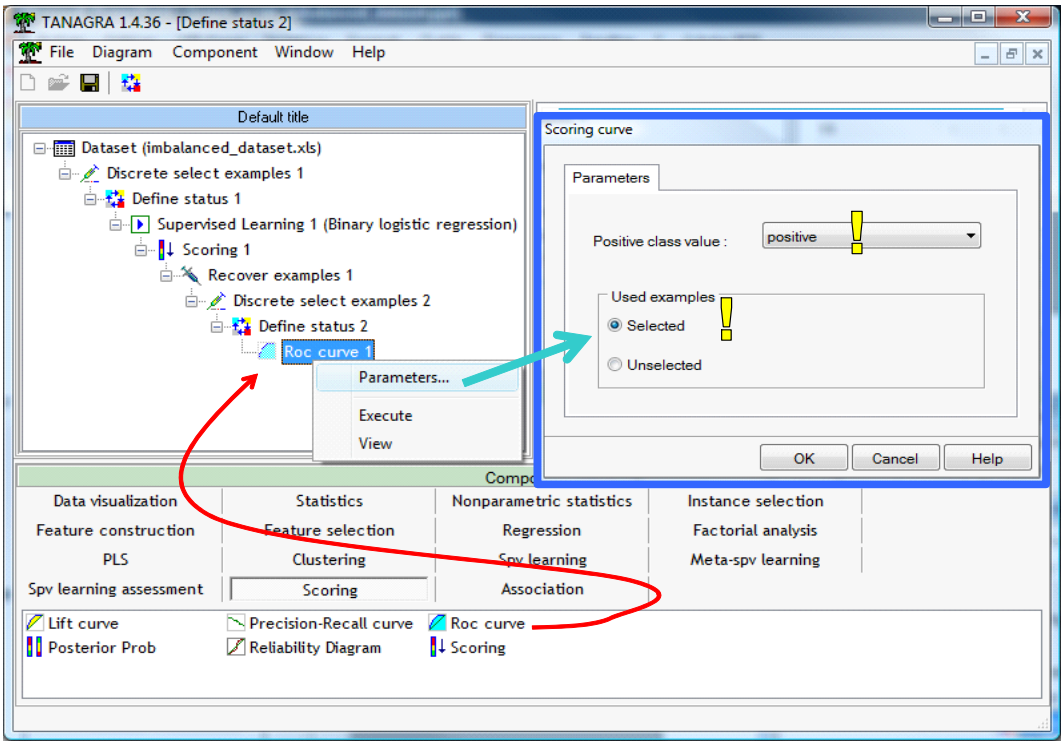
3000 individus sont maintenant activés pour la suite des opérations.

3.4.1 Courbe ROC

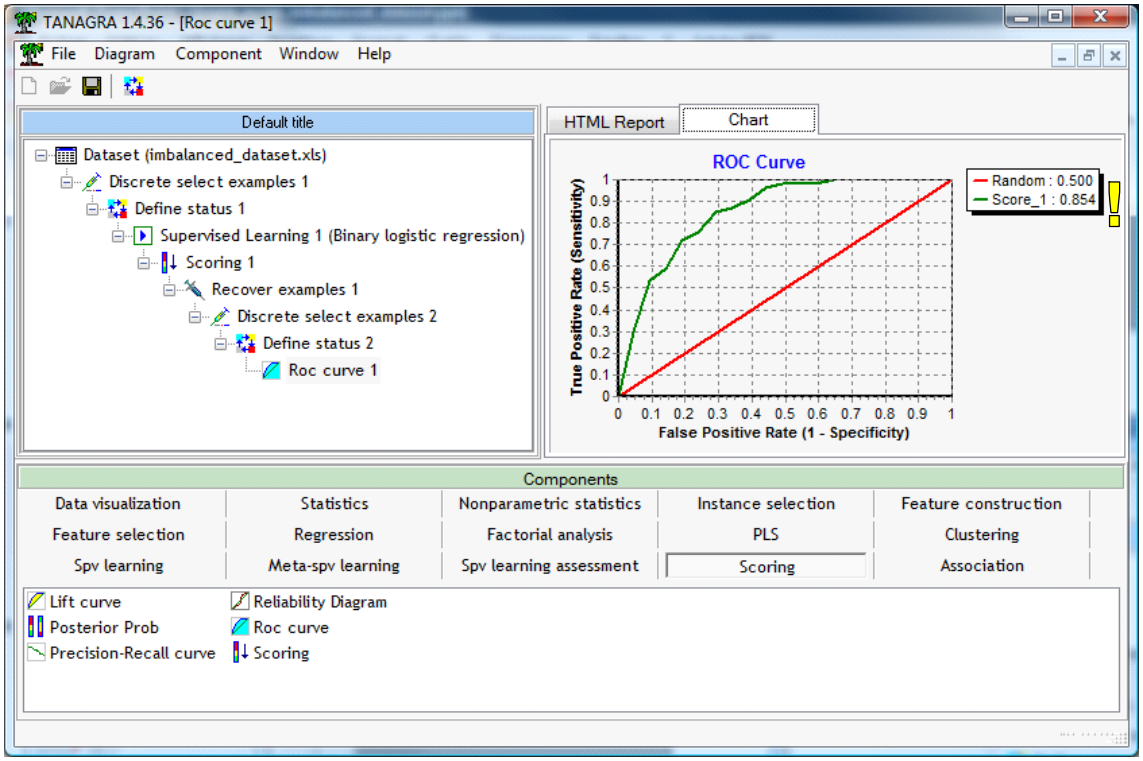
Pour élaborer la courbe ROC, nous avons besoin de la variable à prédire et du score calculé par la régression logistique. Nous introduisons le composant DEFINE STATUS, nous plaçons OBJECTVE en TARGET, SCORE_1 en INPUT.



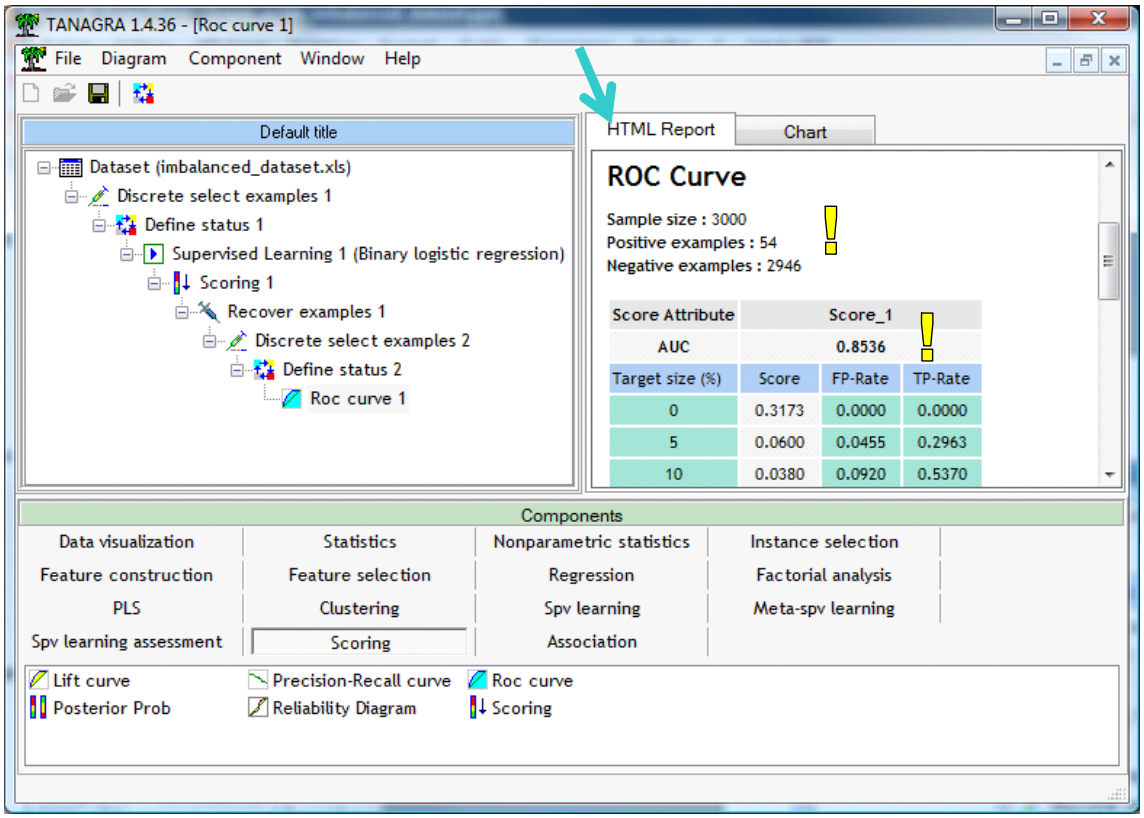
Nous rajoutons le composant ROC CURVE (onglet SCORING). Nous le paramétrons pour qu'il identifie les « positifs » des données sélectionnées c.-à-d. l'échantillon test de 3000 observations.



La courbe ROC montrer que le modèle n'est pas si mauvais que cela finalement. L'aire sous la courbe (AUC) est automatiquement fournie par Tanagra. La probabilité qu'un positif se voie attribuer un score plus élevé qu'à un négatif est de 85.4%.

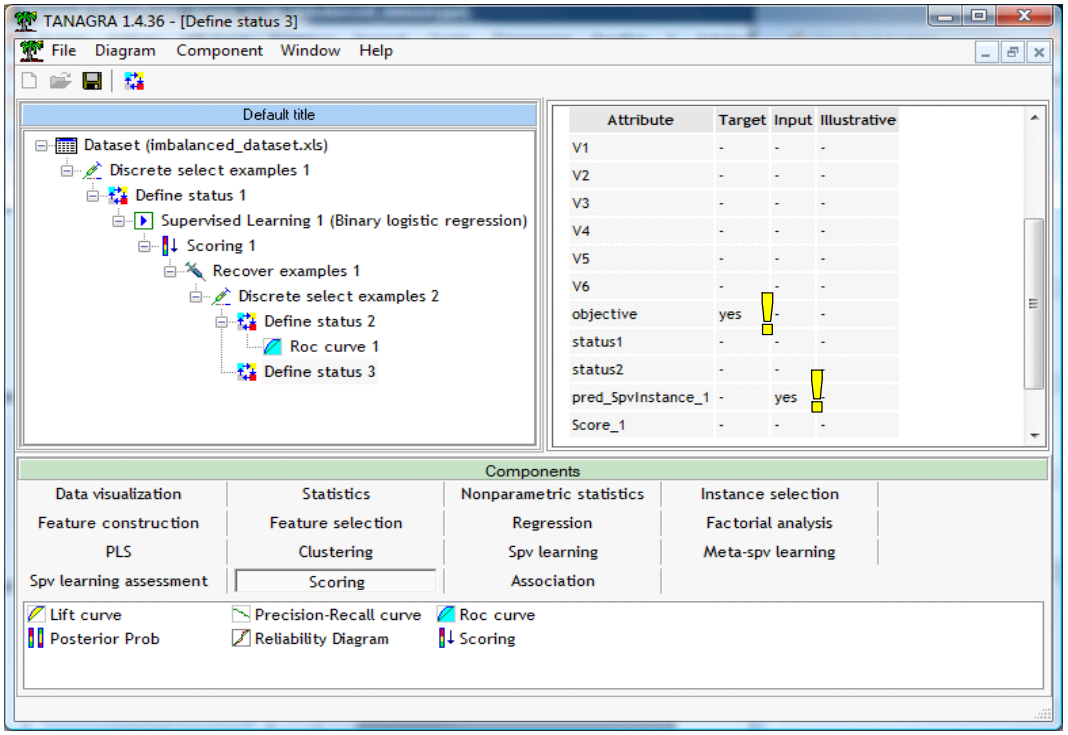


Dans l'onglet « HTML Report », nous avons le détail des calculs. Nous noterons entre autres qu'il y avait bien 54 positifs parmi les 3000 individus de l'échantillon test.

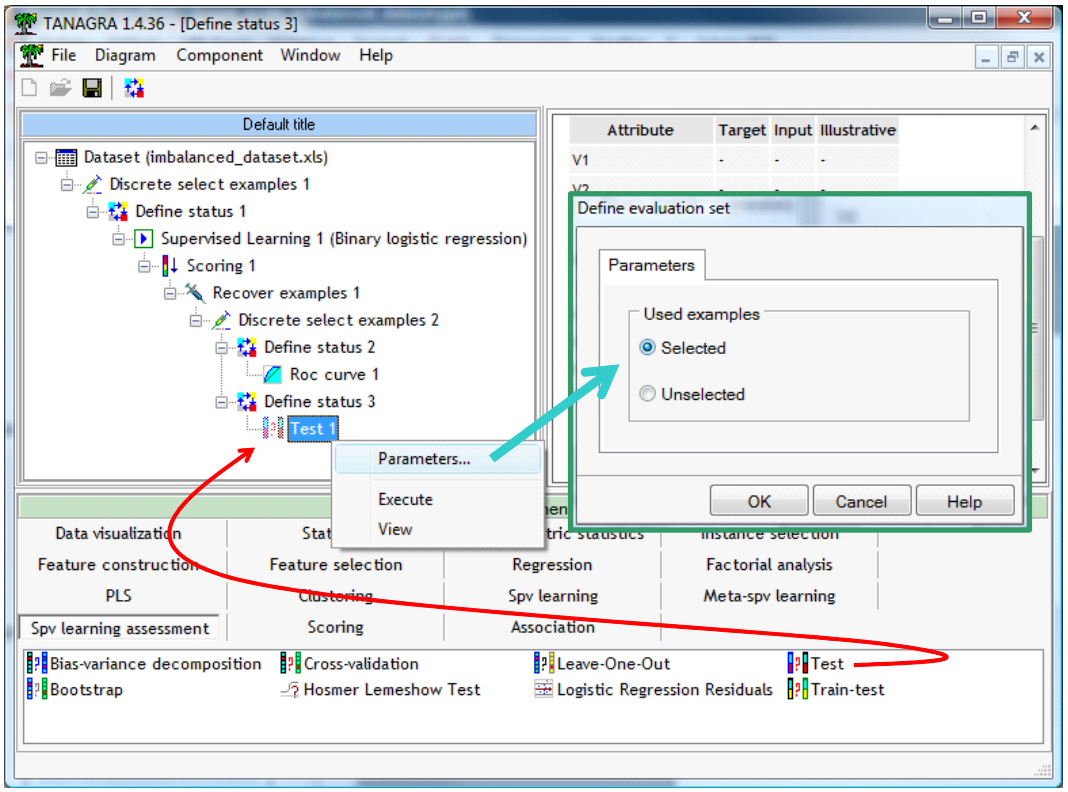


3.4.2 Matrice de confusion

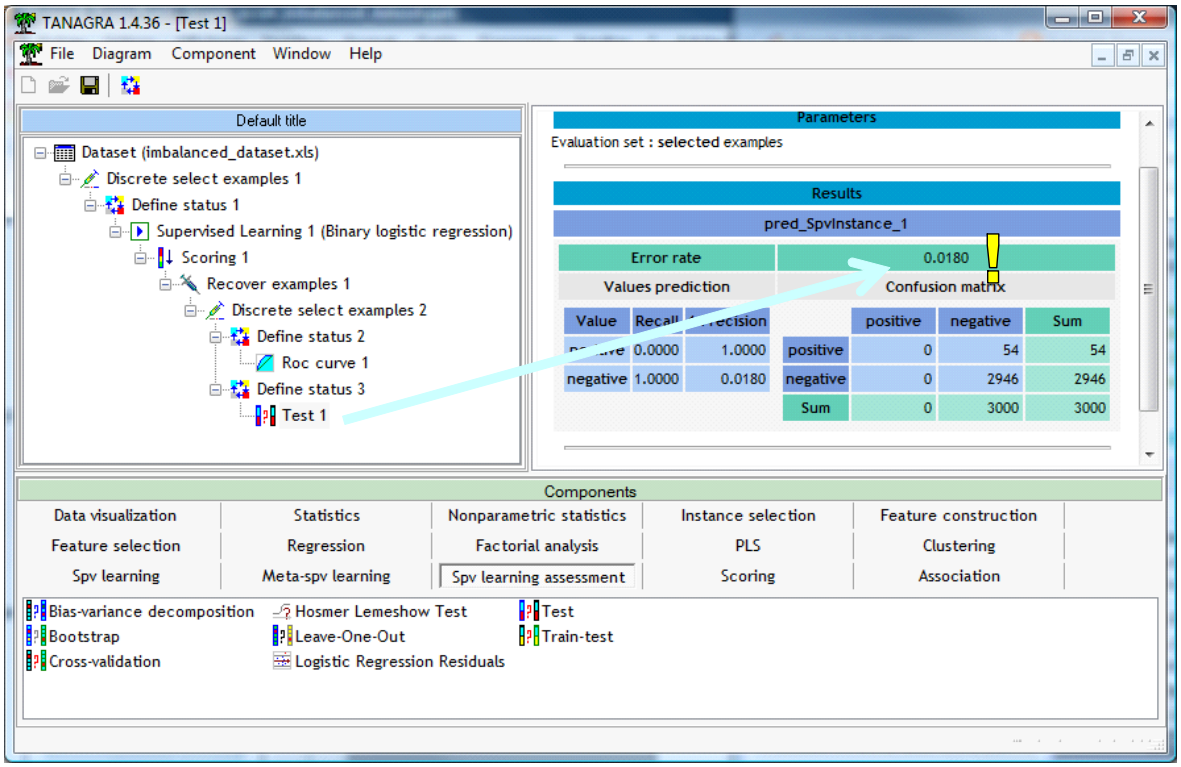
Voyons la matrice de confusion sur l'échantillon test. Avec un autre DEFINE STATUS, nous plaçons OBJECTIVE en TARGET, la prédiction PRED_SPVINSTANCE_1 en INPUT.



Nous plaçons alors le composant TEST (onglet SPV LEARNING ASSESSMENT) que nous paramétrons de manière à calculer la matrice de confusion sur les individus sélectionnés.



Nous cliquons sur VIEW. Le modèle classe tous les individus en « négatifs », avec un taux d'erreur de 1.80%. Sa sensibilité est nulle (0% = 0 /54) et sa précision n'est pas calculable.



Nous percevons surtout que la matrice de confusion n'est pas adaptée pour évaluer les performances des modèles prédictifs lorsque les classes sont très déséquilibrées. La courbe ROC montrer que nous avons un modèle efficace, loin du modèle par défaut ou du classement au hasard. La matrice de confusion et le taux d'erreur ne sont pas capables de retranscrire cela dans notre contexte très particulier.

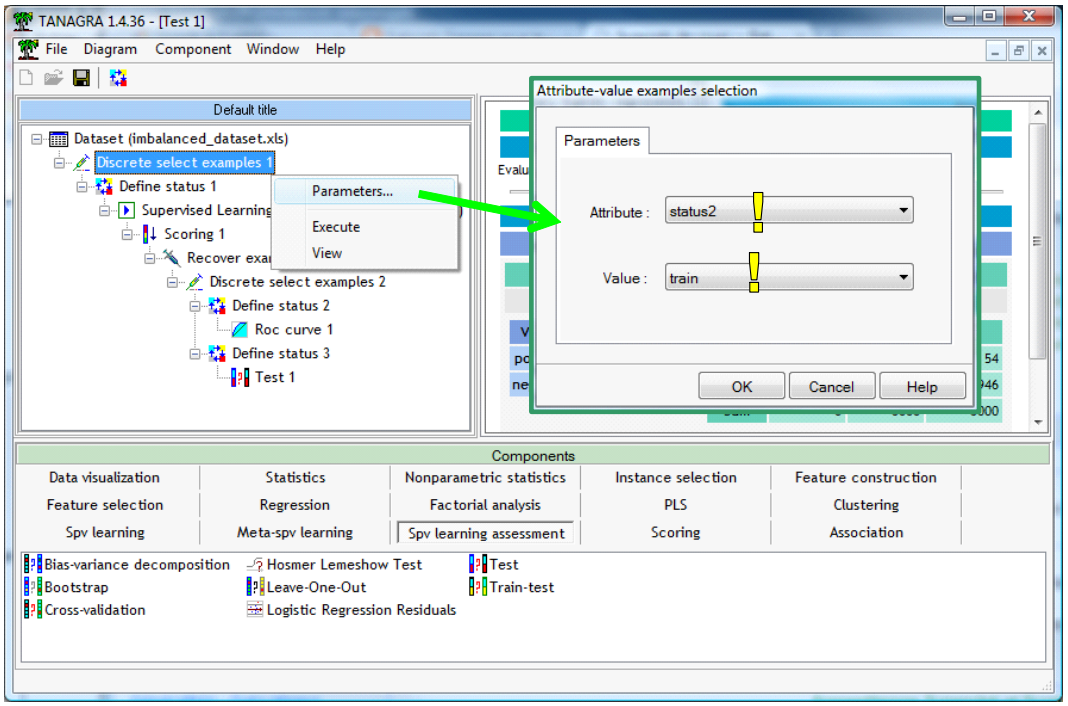
4 Modéliser avec un échantillon équilibré (M2)

Face à notre fichier, la majorité des praticiens du traitement de données vous diront qu'il faut absolument équilibrer les classes. Les ressources étant limitées, on ne peut pas inventer des données qui n'existent pas, une solution simple consiste à sous échantillonner parmi les négatifs, ceux qui sont abondants. C'est ce que nous avons fait dans cette deuxième partie. Parmi les 900 individus de l'échantillon d'apprentissage, nous avons récupéré tous les 15 positifs (ils sont déjà rares, on ne va pas s'amuser à n'en exploiter qu'une partie…) et extrait au hasard 15 négatifs.

Nous nous retrouvons donc à la racine de notre diagramme. Nous devons reproduire les opérations définies précédemment, tout en veillant à travailler sur les bons échantillons à chaque étape.

4.1 Constitution de l'échantillon d'apprentissage

Première étape, nous devons sélectionner les observations dédiées à la construction du modèle. Nous revenons sur le composant DISCRETE SELECT EXAMPLES 1. Nous le paramétrons (menu PARAMETERS) de manière à sélectionner les individus correspondant à « STATUS2 = TRAIN ».

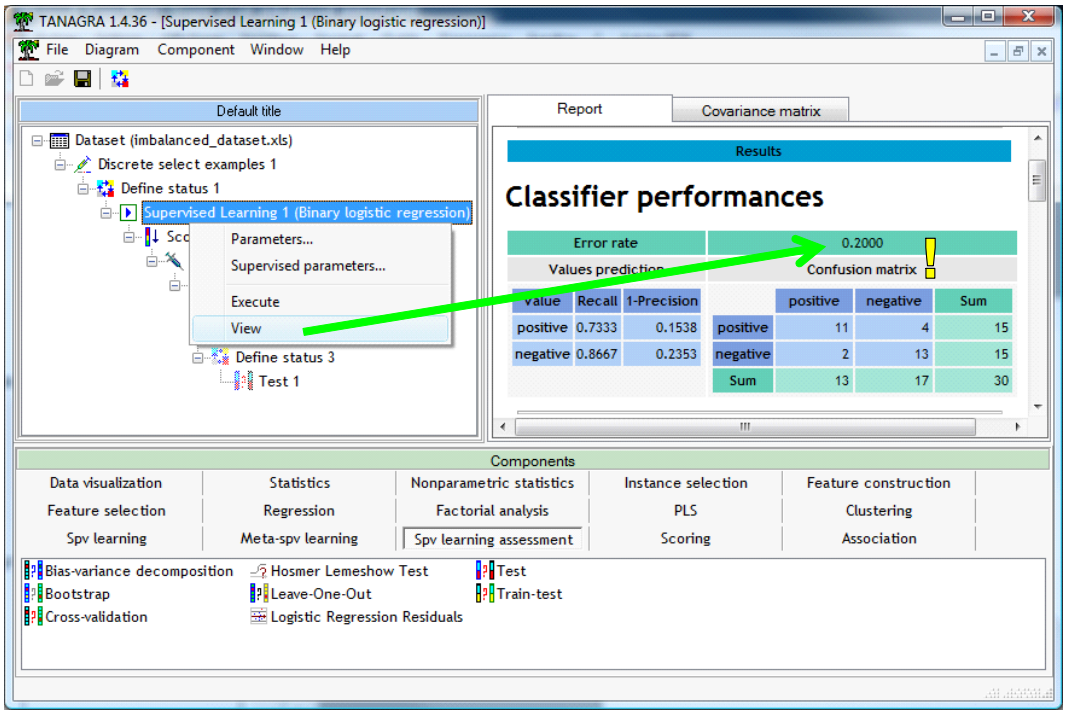


Tanagra nous indique que 30 observations parmi les 3900 sont sélectionnées Tout le reste du processus est identique.

4.2 Construction du modèle prédictif et évaluation

4.2.1 Construction du modèle prédictif

Nous revenons sur le composant SUPERVISED LEARNING 1 (BINARY LOGISTIC), nous cliquons sur VIEW pour lancer la construction du modèle **M2** via la régression logistique sur l'échantillon équilibré.

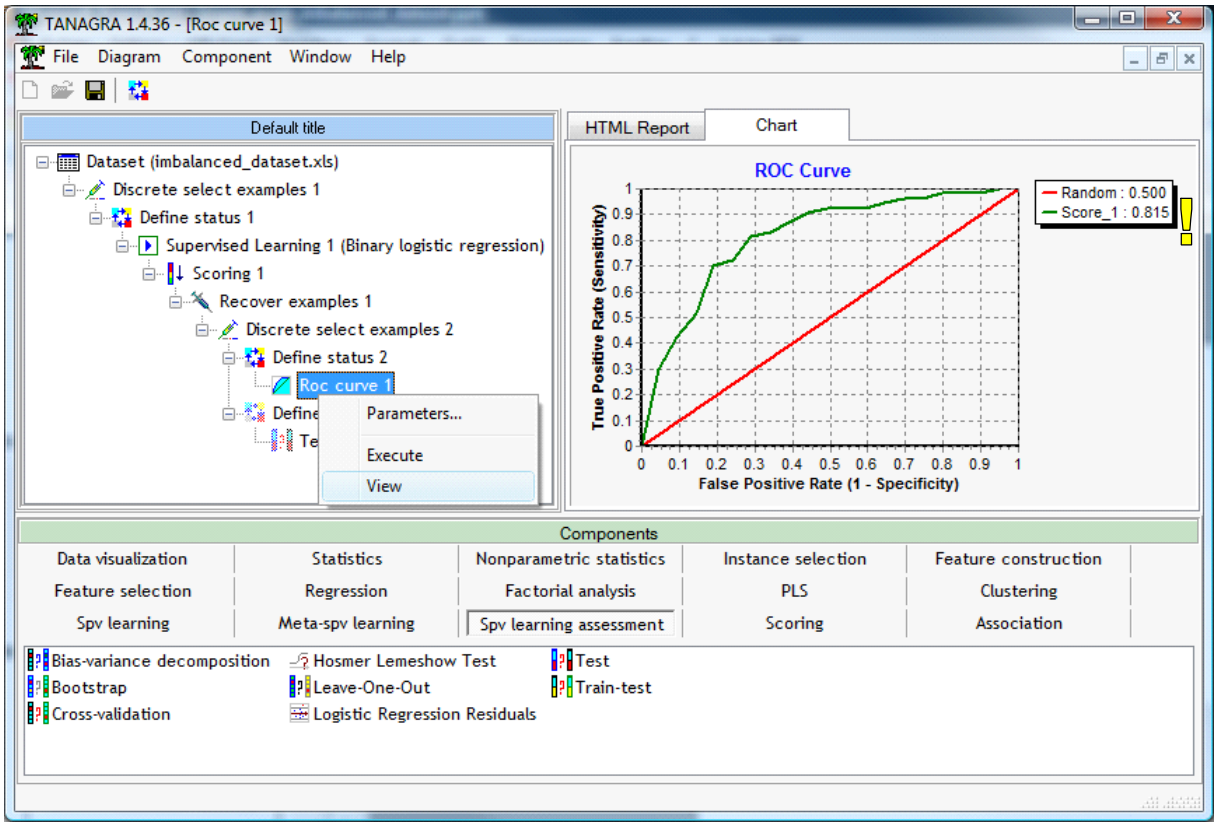


Nous avons un taux d'erreur de 20%, nettement moins bon que précédemment. Nous notons que la sensibilité du modèle est dans le même temps largement amélioré (73.3% = 11 / 15). Mais bon, tout cela est à relativiser, nous avons utilisé l'échantillon d'apprentissage, non représentatif de surcroît, pour évaluer le modèle.

Voyons ce qu'il en est sur l'échantillon test représentatif de 3000 observations.

4.2.2 Courbe ROC

Sur le composant ROC CURVE 1, nous cliquons sur le menu contextuel VIEW.

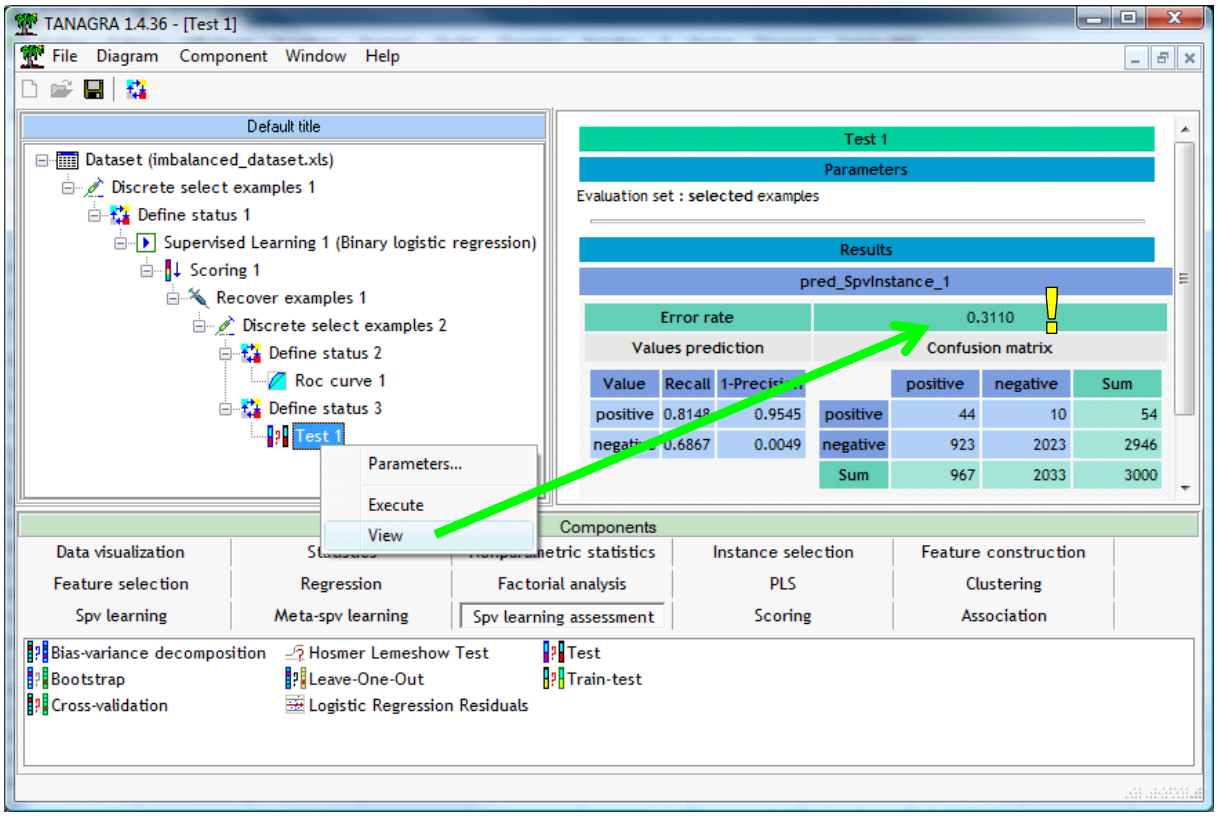


Même s'il semblait mieux reconnaître les positifs dixit la matrice de confusion en resubstitution, nous nous rendons compte que **le modèle M2 n'est pas meilleur que le précédent avec une aire sous la courbe AUC = 81.5%** (84.5% pour M1) ! On aurait une dégradation même. Elle est principalement due à la réduction (artificielle) de l'échantillon d'apprentissage. Disposant de moins d'informations (observations), l'algorithme d'apprentissage est moins performant.

4.2.3 Matrice de confusion

Nous cliquons sur le menu VIEW du composant TEST 1. La matrice de confusion sur l'échantillon test est à l'image de celle calculée en resubstitution : **la sensibilité du modèle est améliorée (81.5% = 44/54), mais avec une précision déplorable (4.55% = 44/967)[8], et finalement au détriment du taux d'erreur** (31.1%), alors qu'il est de 1.8% = 54/3000 si l'on classait toutes les observations en « négatif ».

[8] Lorsque l'on désigne un individu comme positif, il n'a que 4.55% de chances de l'être réellement.



C'est là généralement que fuse depuis le fond de la salle de cours la remarque acerbe : « ben alors, ça sert à rien d'équilibrer monsieur ? ». Je réponds invariablement par une phrase apprise à la télé : « c'est un peu plus compliqué que ça », ça n'engage à rien, et ça entretient le suspens.

5 Conclusion

Équilibrer artificiellement les données lorsque l'on traite un problème avec des classes fortement déséquilibrées n'est pas la panacée, mais ce n'est pas une hérésie non plus. Il faut simplement se rendre compte que, ce faisant, nous modifions le comportement du modèle en améliorant la sensibilité au détriment de la précision et, bien souvent, du taux d'erreur. Après, tout dépend des objectifs de l'analyse que nous menons. Si l'idée est de retrouver un maximum de positif, travailler avec un échantillon équilibré peut être effectivement bénéfique. Si on espère ainsi améliorer le taux d'erreur, on va au devant de grandes désillusions.

Enfin, dernier point important, **on peut améliorer la sensibilité sans avoir à triturer les données**. Il suffit pour cela de modifier le seuil d'affectation du modèle M1 construit sur l'échantillon d'apprentissage représentatif. Si $\pi(x)$ est la probabilité a posteriori d'être positif fournie par la régression logistique, la règle usuelle s'écrit

$$\text{Si } \pi(x) > 0.5 \text{ Alors Y = + Sinon Y = -}$$

Nous pouvons la généraliser facilement en écrivant

$$\text{Si } \pi(x) > \theta \text{ Alors Y = + Sinon Y = -}$$

Où θ est un paramètre que l'on fixe à notre convenance[9], en accord avec nos objectifs. Si l'on souhaite améliorer la sensibilité, on baisse la valeur de θ ; si on souhaite au contraire mettre l'accent sur la précision, on augmente le seuil θ.

Dans notre exemple, avec le modèle M1, si l'on fixe θ = 0.017005, nous obtiendrions la matrice de confusion suivante :

Nombre de objective	pred.theta.corrected		
objective	positive	negative	Total
positive	44	10	54
negative	767	2179	2946
Total	811	2189	3000

Pour la même sensibilité (81.5%) que le modèle M2 construit sur des données volontairement équilibrées, nous avons une précision améliorée (5.42% = 44/8111 vs. 4.55%) ou, cela résulte du même mécanisme, une spécificité meilleure (73.96% = 2179 / 2946 vs. 68.87% = 2023 / 2946). Les résultats annoncés par les courbes ROC sont confirmés ici : M1 est légèrement meilleur que M2 c.-à-d. à sensibilité égale, nous aurons une précision plus élevée.

[9] On pourrait aussi « l'optimiser » par validation croisée ou en utilisant un échantillon spécifique.