# 1. Objectif

Quelques courbes pour évaluer les performances des classifieurs.

L'évaluation des classifieurs est une étape incontournable de l'apprentissage supervisé. Nous avons construit un modèle de prédiction, nous devons en mesurer les performances. D'un côté, nous avons la matrice de confusion et les indicateurs afférents, très populaires dans la recherche en apprentissage automatique (*ah… les fameux grands tableaux avec des moyennes de taux d'erreur sur des bases de données qui n'ont rien à voir entre elles…*) ; de l'autre, dans les applications, on privilégie les courbes qui semblent mystérieuses si l'on n'est pas du domaine (courbe ROC en épidémiologie, entre autres ; courbe de gain en marketing ; courbe rappel – précision en recherche d'information).

Dans ce didacticiel, nous montrons dans un premier temps comment construire ces courbes en détaillant les calculs dans un tableur. Puis, dans un deuxième temps, nous utilisons les logiciels Tanagra 1.4.33 et R 2.9.2 pour les obtenir. Nous comparerons les performances de la régression logistique et des SVM (support vector machine, noyau RBF) sur notre fichier de données.

## 1.1. Matrice de confusion et indicateurs associés

La première manière d'évaluer un classifieur consiste à confronter les valeurs observées de la variable dépendante $Y$ avec les valeurs prédites $\hat{Y}$ fournies par le modèle. L'outil privilégié est la matrice de confusion. Plusieurs ratios résumant les performances des classifieurs en sont déduits.

Prenons un exemple simple pour illustrer notre propos. Nous construisons à l'aide de la régression logistique un modèle destiné à prédire l'occurrence d'une maladie cardiaque (DISEASE : positif ou négatif) à partir des caractéristiques des patients (CHOLESTERAL, THALAC et OLDPEAK ; voir http://archive.ics.uci.edu/ml/datasets/Heart+Disease pour la description des variables). Nous avons appliqué le modèle sur un échantillon test comportant n = 20 observations. Voici le tableau contenant les données et la prédiction de la régression logistique (en gras les bonnes prédictions, en italique les mauvaises).

| cholesteral | thalac | oldpeak | disease | Prediction |
|---|---|---|---|---|
| 322 | 109 | 24 | positive | **positive** |
| 564 | 160 | 16 | negative | *positive* |
| 234 | 161 | 5 | negative | **negative** |
| 311 | 120 | 18 | positive | **positive** |
| 233 | 179 | 4 | negative | **negative** |
| 197 | 116 | 11 | negative | *positive* |
| 309 | 147 | 0 | positive | *negative* |
| 237 | 71 | 10 | positive | **positive** |
| 233 | 163 | 6 | positive | *negative* |
| 224 | 173 | 32 | positive | **positive** |
| 244 | 154 | 14 | positive | *negative* |
| 325 | 154 | 0 | negative | **negative** |
| 282 | 174 | 14 | positive | *negative* |
| 234 | 131 | 1 | negative | **negative** |
| 239 | 160 | 12 | negative | **negative** |
| 213 | 165 | 2 | negative | **negative** |
| 204 | 202 | 0 | negative | **negative** |
| 258 | 147 | 4 | negative | **negative** |
| 263 | 97 | 12 | positive | **positive** |
| 219 | 140 | 12 | positive | *negative* |

Une matrice de confusion, pour un problème à deux classes (Y = + ou -), prend la forme suivante

| Obs. vs. Préd. | Positif | Négatif | Total |
|---|---|---|---|
| **Positif** | a | b | a + b |
| **Négatif** | c | d | c + d |
| **Total** | a + c | b + d | n = a + b + c + d |

Concernant notre exemple, nous obtenons

| Nombre de disease | Prediction | | |
|---|---|---|---|
| disease | positive | negative | Total |
| positive | 5 | 5 | 10 |
| negative | 2 | 8 | 10 |
| Total | 7 | 13 | 20 |

Nous en déduisons une série d'indicateurs :

- Le taux d'erreur est la proportion de mal classés, il estime la probabilité de mal classer un individu pris au hasard dans la population lorsque l'on applique le modèle de prédiction.

$$\varepsilon = \frac{c+b}{n} = 1 - \frac{a+d}{n} = \frac{2+5}{20} = 0.35$$

- On appellera « cible » les individus qui ont été classés « positifs » par le modèle c.-à-d.

$$\text{cible} = \left\{\omega, \hat{Y}(\omega) = +\right\}$$

  La taille de la cible correspond à

$$\#\text{cible} = a + c = 7$$

- Le rappel (ou sensibilité ou taux de vrais positifs - TVP) représente la fraction des positifs intégrés dans la cible, il correspond à la probabilité $P(\omega \in \text{cible} / Y(\omega) = +)$. A partir de la matrice de confusion, nous formons

$$r = Se = \frac{a}{a+b} = \frac{5}{5+5} = 0.5$$

- La précision représente la proportion des positifs à l'intérieur de la cible, elle correspond à la probabilité $P(Y(\omega) = + / \omega \in \text{cible})$. A partir de la matrice de confusion, nous l'obtenons avec

$$p = \frac{a}{a+c} = \frac{5}{7} = 0.71$$

- Le taux de faux positifs correspond à la fraction des négatifs qui ont été intégrés dans la cible. Nous avons

$$TFP = \frac{c}{c+d} = \frac{2}{2+8} = 0.2$$

- La spécificité est la fraction des négatifs qui sont exclus de la cible, soit

$$Sp = \frac{d}{c+d} = 1 - TFP = \frac{8}{2+8} = 0.8$$

Un « bon » classifieur doit présenter d'une part un rappel élevé et, d'autre part, une précision et une spécificité élevée (et un taux de faux positifs faible). On se rend compte dans la pratique que lorsque l'on essaie d'améliorer le rappel, on dégrade souvent le second groupe d'indicateurs. Voyons pourquoi.

## 1.2. Probabilité d'affectation et score

Un classifieur se base très souvent sur la probabilité a posteriori d'être positif pour prédire la valeur de la variable dépendante. Pour un individu $\omega$, elle fournit

$$\hat{\pi}(\omega) = \hat{P}[Y(\omega) = + / X(\omega)]$$

La règle de classement s'écrit

$$\text{Si } \hat{\pi}(\omega) \geq \text{seuil Alors } \hat{Y}(\omega) = + \text{ Sinon } \hat{Y}(\omega) = -$$

Habituellement, **seuil = 0.5**. Ce qui revient à utiliser une règle d'affectation basée sur le maximum de la probabilité d'appartenance à une classe[1].

| cholesteral | thalac | oldpeak | disease | score | prediction |
|---|---|---|---|---|---|
| 322 | 109 | 24 | positive | **0.9335** | **positive** |
| 564 | 160 | 16 | negative | **0.8897** | **positive** |
| 234 | 161 | 5 | negative | 0.2417 | negative |
| 311 | 120 | 18 | positive | **0.8537** | **positive** |
| 233 | 179 | 4 | negative | 0.1349 | negative |
| 197 | 116 | 11 | negative | **0.6427** | **positive** |
| 309 | 147 | 0 | positive | 0.3956 | negative |
| 237 | 71 | 10 | positive | **0.9183** | **positive** |
| 233 | 163 | 6 | positive | 0.2397 | negative |
| 224 | 173 | 32 | positive | **0.5433** | **positive** |
| 244 | 154 | 14 | positive | 0.4468 | negative |
| 325 | 154 | 0 | negative | 0.3650 | negative |
| 282 | 174 | 14 | positive | 0.3446 | negative |
| 234 | 131 | 1 | negative | 0.4146 | negative |
| 239 | 160 | 12 | negative | 0.3546 | negative |
| 213 | 165 | 2 | negative | 0.1620 | negative |
| 204 | 202 | 0 | negative | 0.0406 | negative |
| 258 | 147 | 4 | negative | 0.3696 | negative |
| 263 | 97 | 12 | positive | **0.8608** | **positive** |
| 219 | 140 | 12 | positive | 0.4910 | negative |

**Figure 1 - Score et prédiction basée sur le seuil de 0.5**

[1] Tous les classifieurs en apprentissage supervisé entrent dans ce canevas. Certains sont de bons estimateurs de $\hat{\pi}(\omega)$ (ex. la régression logistique) ; d'autres estiment directement le mode (ex. SVM – support vector machine) ; d'autres encore en estiment mal la probabilité, mais estiment bien son mode (ex. bayesien naïf). Voir T. Hastie, R. Tibshirani, J. Friedman, « The elements of Statistical Learning – Data Mining, Inference and Prediction », Springer, 2001 ; page 381.

Nous observons cette correspondance entre la probabilité fournie par la régression logistique et la prédiction dans notre fichier DISEASE (Figure 1).

### 1.3. Faire varier le seuil d'affectation – Quelques courbes pour évaluer les classifieurs

**Le seuil de 0.5 est en réalité optimal pour une situation bien définie** : l'échantillon utilisé est représentatif (la proportion des positifs reflète la probabilité d'être positif dans la population), les coûts de mauvais classement sont symétriques et unitaires (mal prédire coûte 1, bien classer coûte 0). Plutôt que de se limiter à ce seuil, nous pouvons évaluer plus largement le comportement du classifieur en le faisant varier et en calculant pour chaque configuration la matrice de confusion. Cette idée est sous-jacente aux différents graphiques que nous présentons dans cette section.

La quantité $\hat{\pi}(\omega)$ est une probabilité qui indique le degré d'appartenance aux positifs d'un individu. On parle généralement de « **score** »[2] [$score(\omega)$]. On s'attend à ce que les individus positifs présentent des scores plus élevés que les négatifs. Reprenons notre exemple ci-dessus. Nous insérons dans le tableau le score fourni par la régression logistique. Puis nous ordonnons les individus selon le score décroissant (Figure 2). On notera que les positifs sont effectivement en majorité regroupés dans la partie haute du tableau, les négatifs dans la partie basse.

| cholesteral | thalac | oldpeak | disease | score |
|---|---|---|---|---|
| 322 | 109 | 24 | positive | 0.9335 |
| 237 | 71 | 10 | positive | 0.9183 |
| 564 | 160 | 16 | negative | 0.8897 |
| 263 | 97 | 12 | positive | 0.8608 |
| 311 | 120 | 18 | positive | 0.8537 |
| 197 | 116 | 11 | negative | 0.6427 |
| 224 | 173 | 32 | positive | 0.5433 |
| 219 | 140 | 12 | positive | 0.4910 |
| 244 | 154 | 14 | positive | 0.4468 |
| 234 | 131 | 1 | negative | 0.4146 |
| 309 | 147 | 0 | positive | 0.3956 |
| 258 | 147 | 4 | negative | 0.3696 |
| 325 | 154 | 0 | negative | 0.3650 |
| 239 | 160 | 12 | negative | 0.3546 |
| 282 | 174 | 14 | positive | 0.3446 |
| 234 | 161 | 5 | negative | 0.2417 |
| 233 | 163 | 6 | positive | 0.2397 |
| 213 | 165 | 2 | negative | 0.1620 |
| 233 | 179 | 4 | negative | 0.1349 |
| 204 | 202 | 0 | negative | 0.0406 |

**Figure 2 - Tableau des observations ordonnées selon un score décroissant**

On notera surtout qu'en faisant varier le seuil d'affectation, nous pouvons obtenir une série de matrice de confusion. Voyons ce qu'il en est pour chaque seuil candidat dans le tableau ci-dessus.

[2] Un score n'est pas forcément une probabilité. En réalité, toute quantité permettant de caractériser le degré de « positivité » des individus convient. Si l'on se réfère à la régression logistique, le score peut être la probabilité d'appartenance, mais cela peut être également le LOGIT. L'une et l'autre permettent d'ordonner identiquement les individus selon leur propension à être positif. Rappelons que le LOGIT varie entre -∞ et +∞.

<table>
<tr><th>Seuil d'affectation</th><th>Matrice de confusion et indicateurs</th></tr>
<tr><td>0.9335</td><td>
<table>
<tr><td>Obs. x Préd.</td><td>positif</td><td>négatif</td><td>total</td></tr>
<tr><td>positif</td><td>1</td><td>9</td><td>10</td></tr>
<tr><td>négatif</td><td>0</td><td>10</td><td>10</td></tr>
<tr><td>total</td><td>1</td><td>19</td><td>20</td></tr>
</table>
<table>
<tr><td>#cible</td><td>1</td></tr>
<tr><td>rappel</td><td>0.10</td></tr>
<tr><td>TFP</td><td>0.00</td></tr>
<tr><td>spécificité</td><td>1.00</td></tr>
</table>
</td></tr>
<tr><td>0.9183</td><td>
<table>
<tr><td>Obs. x Préd.</td><td>positif</td><td>négatif</td><td>total</td></tr>
<tr><td>positif</td><td>2</td><td>8</td><td>10</td></tr>
<tr><td>négatif</td><td>0</td><td>10</td><td>10</td></tr>
<tr><td>total</td><td>2</td><td>18</td><td>20</td></tr>
</table>
<table>
<tr><td>#cible</td><td>2</td></tr>
<tr><td>rappel</td><td>0.20</td></tr>
<tr><td>TFP</td><td>0.00</td></tr>
<tr><td>spécificité</td><td>1.00</td></tr>
</table>
</td></tr>
<tr><td>0.8897</td><td>
<table>
<tr><td>Obs. x Préd.</td><td>positif</td><td>négatif</td><td>total</td></tr>
<tr><td>positif</td><td>2</td><td>8</td><td>10</td></tr>
<tr><td>négatif</td><td>1</td><td>9</td><td>10</td></tr>
<tr><td>total</td><td>3</td><td>17</td><td>20</td></tr>
</table>
<table>
<tr><td>#cible</td><td>3</td></tr>
<tr><td>rappel</td><td>0.20</td></tr>
<tr><td>TFP</td><td>0.10</td></tr>
<tr><td>spécificité</td><td>0.90</td></tr>
</table>
</td></tr>
<tr><td>0.8608</td><td>
<table>
<tr><td>Obs. x Préd.</td><td>positif</td><td>négatif</td><td>total</td></tr>
<tr><td>positif</td><td>3</td><td>7</td><td>10</td></tr>
<tr><td>négatif</td><td>1</td><td>9</td><td>10</td></tr>
<tr><td>total</td><td>4</td><td>16</td><td>20</td></tr>
</table>
<table>
<tr><td>#cible</td><td>4</td></tr>
<tr><td>rappel</td><td>0.30</td></tr>
<tr><td>TFP</td><td>0.10</td></tr>
<tr><td>spécificité</td><td>0.90</td></tr>
</table>
</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>0.0406</td><td>
<table>
<tr><td>Obs. x Préd.</td><td>positif</td><td>négatif</td><td>total</td></tr>
<tr><td>positif</td><td>10</td><td>0</td><td>10</td></tr>
<tr><td>négatif</td><td>10</td><td>0</td><td>10</td></tr>
<tr><td>total</td><td>20</td><td>0</td><td>20</td></tr>
</table>
<table>
<tr><td>#cible</td><td>20</td></tr>
<tr><td>rappel</td><td>1.00</td></tr>
<tr><td>TFP</td><td>1.00</td></tr>
<tr><td>spécificité</td><td>0.00</td></tr>
</table>
</td></tr>
</table>

Nous noterons plusieurs propriétés intéressantes :

- A mesure que le seuil baisse, la taille de la cible (#cible) augmente. C'est tout à fait mécanique, nous intégrons plus d'observations dans les prédictions positives. Lorsque nous testons toutes les configurations possibles, et s'il n'y a pas d'ex-aequo, l'augmentation est d'une observation d'un seuil à l'autre.
- A mesure que nous faisons baisser le seuil d'affectation, et que la taille de la cible s'accroît, le rappel a tendance à augmenter. En effet, nous captons en priorité les observations positives qui présentent un score plus élevé que les négatifs.
- Mais, dans le même temps, nous prenons le risque de dégrader (diminuer) la précision et la spécificité : les chances d'avoir des observations négatives dans la cible augmente. Le taux de faux positifs, lui, a tendance à augmenter.

Les courbes destinées à caractériser les performances des classifieurs s'appuient sur ces valeurs successives (et ces propriétés) pour traduire leur comportement :

- **La courbe ROC** (Receiver Operating Characteristic) a pour origine la théorie du signal. Mais elle est très largement utilisée dans de multiples contextes. Elle met en balance le taux de faux positifs (TFP) en abscisse, et la sensibilité (le rappel, le taux de vrais positifs) en ordonnée. Elle est monotone croissante. Les coordonnées extrêmes sont (0, 0) et (1, 1). Sa popularité repose également sur la production d'un indicateur synthétique, l'aire sous la courbe (AUC), qui indique la probabilité d'assigner un score plus élevé à un positif qu'à un négatif. Dans le pire des cas, les scores attribués sont identiques chez les positifs et chez les négatifs, la courbe est confondue avec la diagonale principale.

- **La courbe de gain** (ou courbe lift cumulé) est beaucoup utilisée en marketing. Elle met en relation la taille relative de la cible (TRC = #cible/n) en abscisse et la sensibilité en ordonnée. Elle permet, entre autres, d'évaluer l'efficacité des ciblages réalisés lors des prospections par mailing. La courbe est monotonement croissante. Ici aussi, si le score ne permet pas de mettre en avant les positifs, la courbe est confondue avec la diagonale principale.

- **La courbe rappel – précision** est surtout mise en avant en recherche d'information. Elle oppose le rappel en abscisse et la précision en ordonnée. L'idée est de décrire la pertinence de l'ensemble de documents obtenus à l'issue d'une requête sur un système quelconque. La courbe est globalement décroissante. Mais elle n'est pas monotone. Si le classifieur attribue en priorité les scores élevés aux positifs, la précision est initialement élevée lorsque la cible est restreinte. Le rappel est en revanche mécaniquement mauvais, nous retrouvons peu de documents positifs. Lorsque nous augmentons la taille de la cible c.-à-d. nous introduisons des individus avec des scores de plus en plus faibles dans la cible, la précision se dégrade, alors que dans le même temps le rappel s'améliore. Au final, quand nous incluons tous les individus dans la cible, la précision est égale à la proportion de positifs dans le fichier.

Bien entendu, ce n'est plus un mystère au regard des indicateurs utilisés pour les composer, ces courbes ont de fortes connexions entre elles.

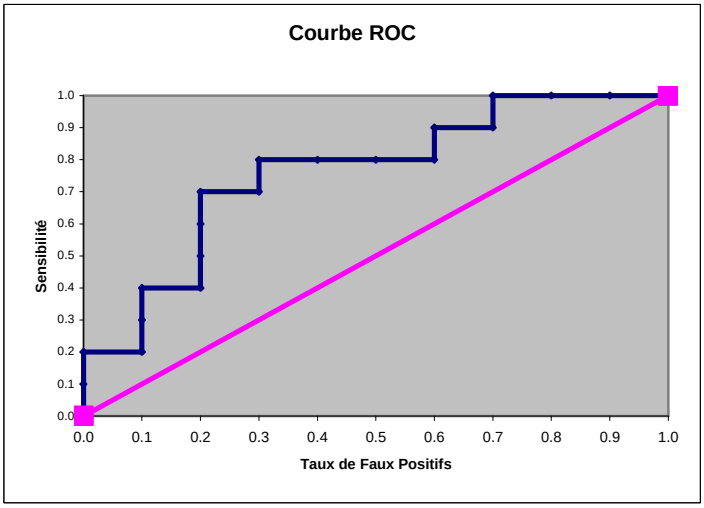
Ces courbes permettent de caractériser les classifieurs. Elles permettent aussi d'en comparer les performances. Il suffit de les rassembler dans un même repère. Il est ainsi possible d'établir des relations de domination[3] entre les méthodes sur le fichier étudié. Par exemple, si la courbe ROC du classifieur A est toujours situé au dessus de celle de B, nous savons qu'elle sera toujours meilleure quelle que soit la combinaison de coûts de mauvais classement utilisée. On a montré également que, dans de cas, la courbe rappel et précision de A sera toujours située au dessus de celle de B.

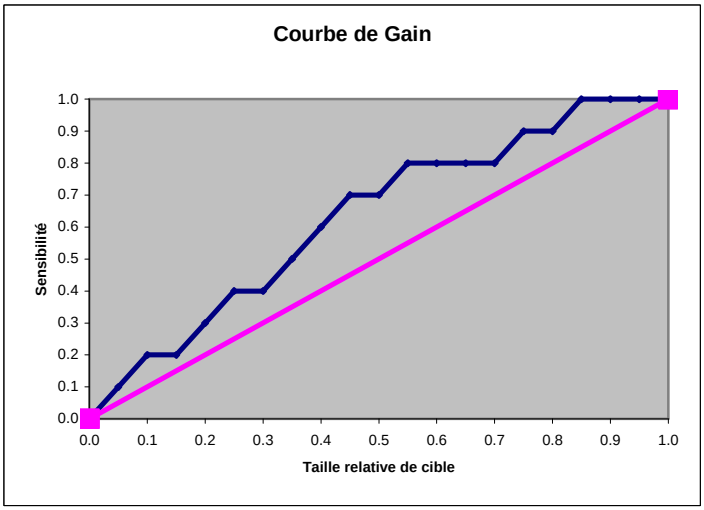
Dans notre feuille de calcul, puisque le tableau est déjà décroissant selon le score, nous pouvons calculer les quantités (a, b, c et d) des matrices de confusion relatifs aux seuils successifs. Nous en déduisons les indicateurs qui serviront à confectionner les courbes (TRC, rappel, précision, TFP).

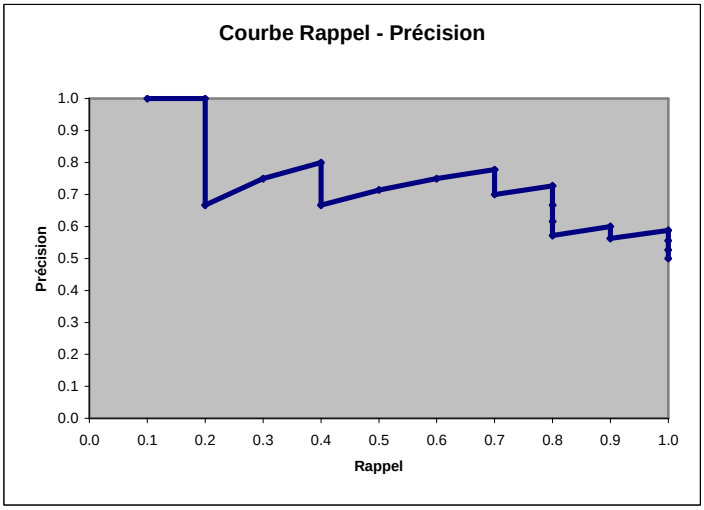
[3] Non, non, pas de fantasmes inutiles, il s'agit bien de performances de méthodes supervisées ici.

| cholesteral | thalac | oldpeak | disease | score | a | b | c | d | TRC | rappel | précision | TFP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 322 | 109 | 24 | positive | 0.9335 | 1 | 9 | 0 | 10 | 0.05 | 0.10 | 1.00 | 0.00 |
| 237 | 71 | 10 | positive | 0.9183 | 2 | 8 | 0 | 10 | 0.10 | 0.20 | 1.00 | 0.00 |
| 564 | 160 | 16 | negative | 0.8897 | 2 | 8 | 1 | 9 | 0.15 | 0.20 | 0.67 | 0.10 |
| 263 | 97 | 12 | positive | 0.8608 | 3 | 7 | 1 | 9 | 0.20 | 0.30 | 0.75 | 0.10 |
| 311 | 120 | 18 | positive | 0.8537 | 4 | 6 | 1 | 9 | 0.25 | 0.40 | 0.80 | 0.10 |
| 197 | 116 | 11 | negative | 0.6427 | 4 | 6 | 2 | 8 | 0.30 | 0.40 | 0.67 | 0.20 |
| 224 | 173 | 32 | positive | 0.5433 | 5 | 5 | 2 | 8 | 0.35 | 0.50 | 0.71 | 0.20 |
| 219 | 140 | 12 | positive | 0.4910 | 6 | 4 | 2 | 8 | 0.40 | 0.60 | 0.75 | 0.20 |
| 244 | 154 | 14 | positive | 0.4468 | 7 | 3 | 2 | 8 | 0.45 | 0.70 | 0.78 | 0.20 |
| 234 | 131 | 1 | negative | 0.4146 | 7 | 3 | 3 | 7 | 0.50 | 0.70 | 0.70 | 0.30 |
| 309 | 147 | 0 | positive | 0.3956 | 8 | 2 | 3 | 7 | 0.55 | 0.80 | 0.73 | 0.30 |
| 258 | 147 | 4 | negative | 0.3696 | 8 | 2 | 4 | 6 | 0.60 | 0.80 | 0.67 | 0.40 |
| 325 | 154 | 0 | negative | 0.3650 | 8 | 2 | 5 | 5 | 0.65 | 0.80 | 0.62 | 0.50 |
| 239 | 160 | 12 | negative | 0.3546 | 8 | 2 | 6 | 4 | 0.70 | 0.80 | 0.57 | 0.60 |
| 282 | 174 | 14 | positive | 0.3446 | 9 | 1 | 6 | 4 | 0.75 | 0.90 | 0.60 | 0.60 |
| 234 | 161 | 5 | negative | 0.2417 | 9 | 1 | 7 | 3 | 0.80 | 0.90 | 0.56 | 0.70 |
| 233 | 163 | 6 | positive | 0.2397 | 10 | 0 | 7 | 3 | 0.85 | 1.00 | 0.59 | 0.70 |
| 213 | 165 | 2 | negative | 0.1620 | 10 | 0 | 8 | 2 | 0.90 | 1.00 | 0.56 | 0.80 |
| 233 | 179 | 4 | negative | 0.1349 | 10 | 0 | 9 | 1 | 0.95 | 1.00 | 0.53 | 0.90 |
| 204 | 202 | 0 | negative | 0.0406 | 10 | 0 | 10 | 0 | 1.00 | 1.00 | 0.50 | 1.00 |

Nous pouvons maintenant former les différentes courbes.






Dans ce qui suit, nous verrons comment les produire avec **TANAGRA 1.4.33** et **R 2.9.2**. Au-delà de la simple construction des courbes, nous en profiterons pour comparer les performances respectives de la régression logistique et des SVM (« support vector machine », package **e1071**) sur nos données.

# 2. Données

Le fichier HEART_DISEASE_FOR_CURVES.XLS[4] comporte 270 observations. Par rapport à la configuration ci-dessus, où nous calculions manuellement les coordonnées sous Excel sur un échantillon test de 20 observations, nous avons modifié la répartition des individus en apprentissage et test : 150 sont dédiés à la construction du modèle, 120 à son évaluation[5]. Nous obtiendrons ainsi des courbes mieux lissées.

Nous disposons de 5 colonnes de valeurs. La première, dénommée SAMPLE (train ou test), permet d'identifier le statut des observations. Les 3 suivantes sont les descripteurs, tous continus (CHOLESTERAL, THALAC et OLDPEAK). Enfin, la dernière colonne correspond à la variable à prédire DISEASE (positif ou négatif) (Figure 3).

| sample | cholesteral | thalac | oldpeak | disease |
|---|---|---|---|---|
| train | 261 | 141 | 3 | positive |
| train | 263 | 105 | 2 | negative |
| train | 269 | 121 | 2 | negative |
| train | 177 | 140 | 4 | negative |
| train | 256 | 142 | 6 | positive |
| train | 239 | 142 | 12 | positive |
| train | 293 | 170 | 12 | positive |
| train | 407 | 154 | 40 | positive |
| train | 226 | 111 | 0 | negative |
| train | 235 | 180 | 0 | negative |

**Figure 3 - Les 10 premières lignes du fichier DISEASE**

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

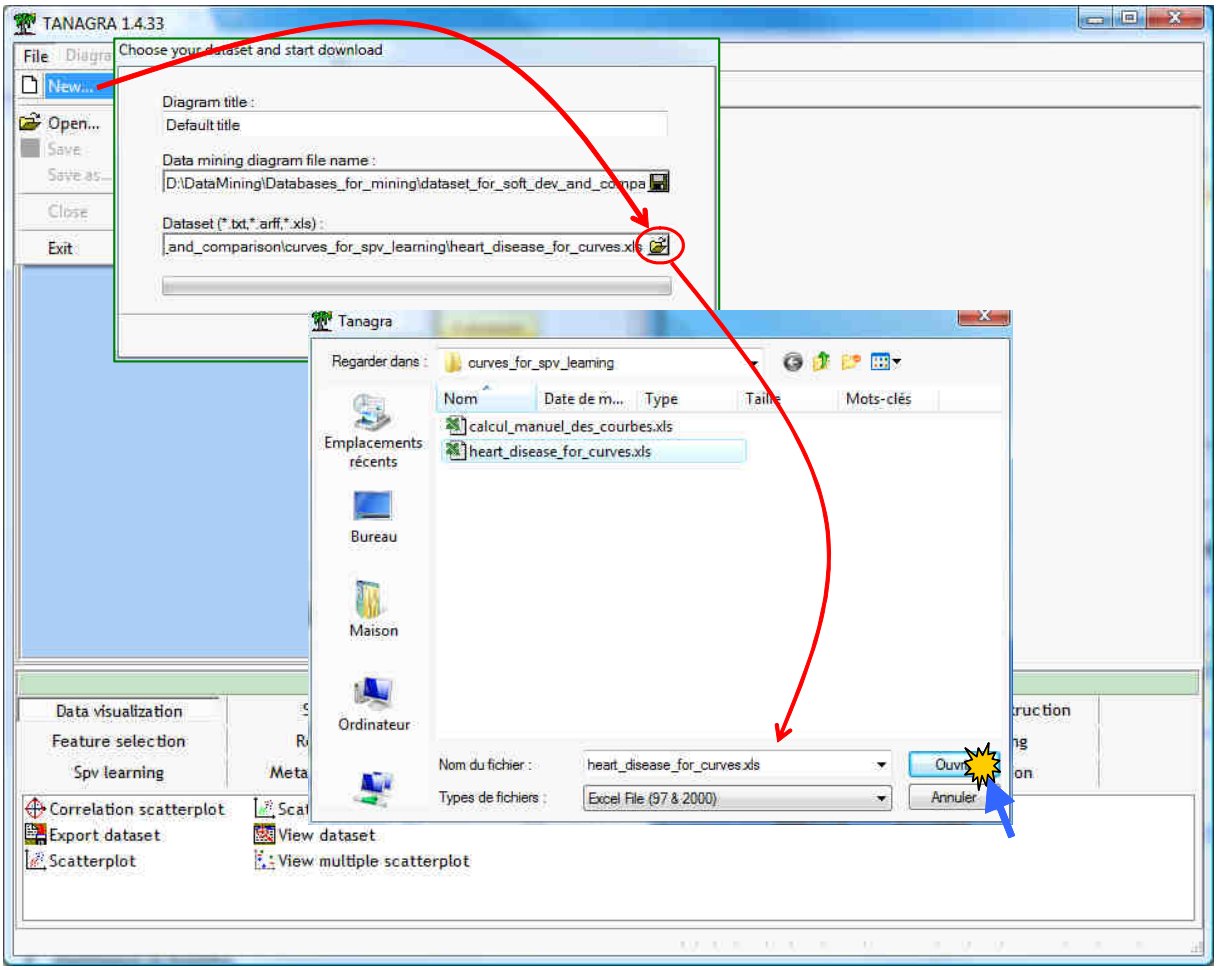
[5] Les deux ensembles de données, répartis différemment, sont disponibles dans le classeur Excel.

# 3. Construction des courbes avec Tanagra

## 3.1. Importation et subdivision des données en apprentissage et test

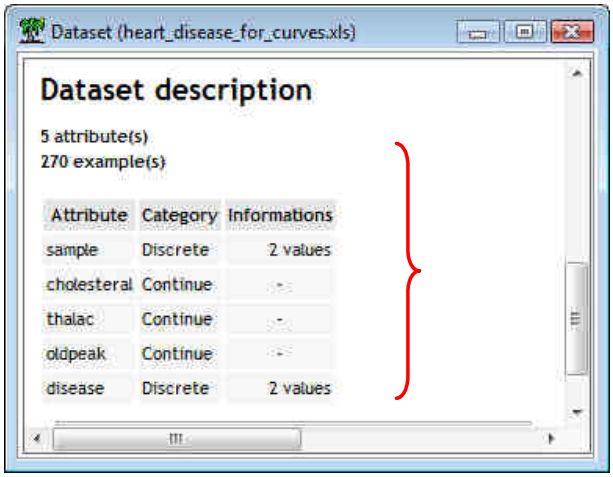
Tanagra sait importer directement un fichier XLS. Il faut que les données soient situées dans la première feuille de calcul, la première ligne est composée des noms de variables. Il s'appuie sur la seconde ligne de valeurs pour détecter le type des variables. Enfin, il ne faut pas que le fichier soit par ailleurs en cours d'édition dans Excel[6].

Après lancé Tanagra, nous actionnons le menu FILE / NEW pour créer un nouveau projet. Nous sélectionnons le fichier HEART_DISEASE_FOR_CURVES.XLS. Nous validons.



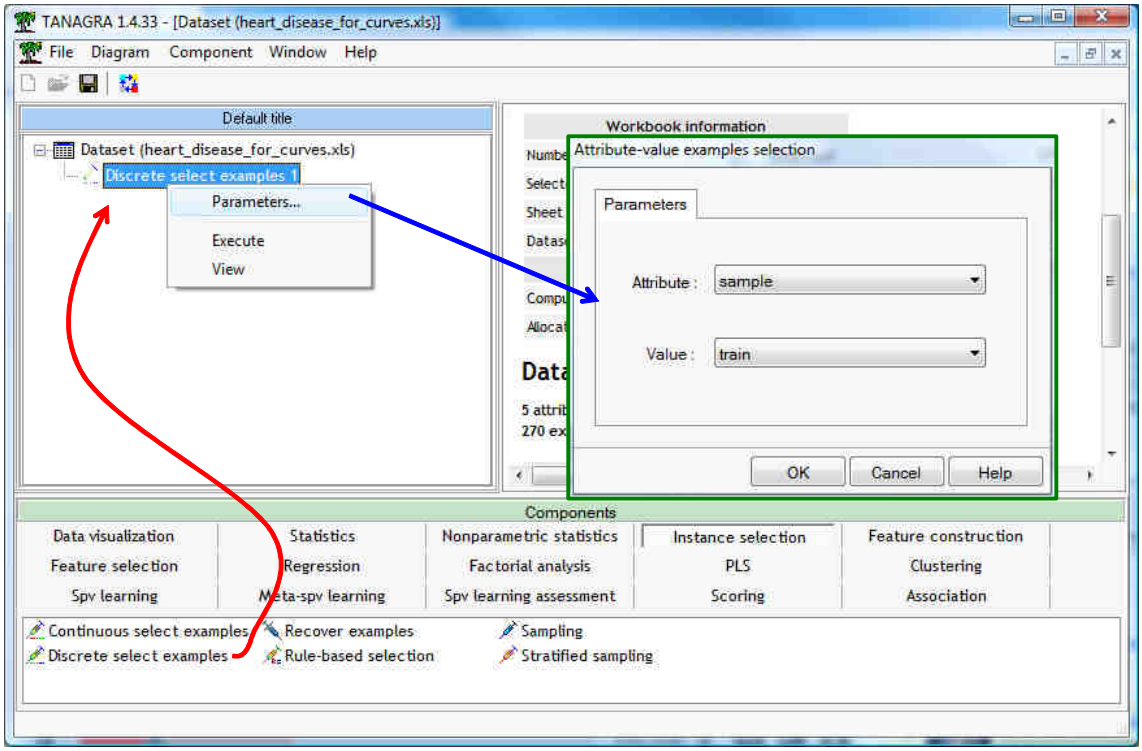

Le fichier est automatiquement chargé. Tanagra nous indique qu'il y a 270 observations et 5 colonnes dans la feuille qui a été importée.

[6] Voir http://tutoriels-data-mining.blogspot.com/2008/03/importation-fichier-xls-excel-mode.html pour plus de détails. Il est également possible d'envoyer les données d'Excel vers Tanagra en utilisant une macro complémentaire pré installée, voir http://tutoriels-data-mining.blogspot.com/2008/03/importation-fichier-xls-excel-macro.html




Nous devons partitionner les observations en deux sous-ensembles disjoints à l'aide de la colonne SAMPLE : le premier servira à l'apprentissage des modèles de prédiction, le second sera utilisé pour la construction des courbes.

Nous introduisons le composant DISCRETE SELECT EXAMPLES (onglet INSTANCE SELECTION) dans le diagramme. Nous cliquons sur le menu contextuel PARAMETERS pour spécifier les paramètres. Les individus actifs, utilisés pour l'apprentissage, correspondent à SAMPLE = TRAIN.

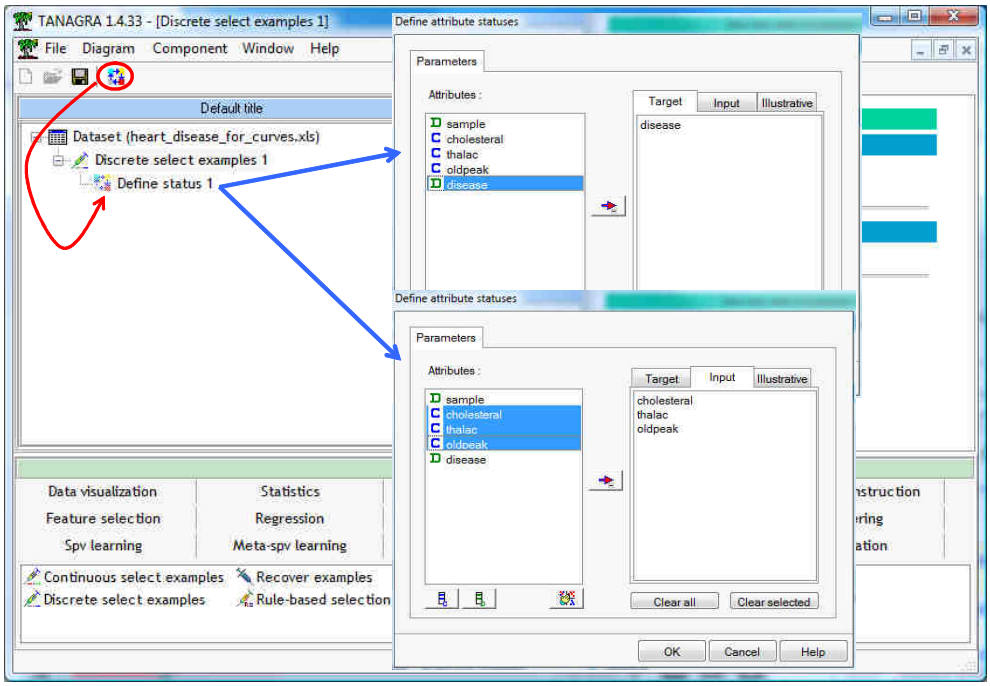



Nous validons et nous cliquons sur VIEW. Tanagra nous indique que 150 observations parmi les 270 initiaux sont sélectionnées[7].

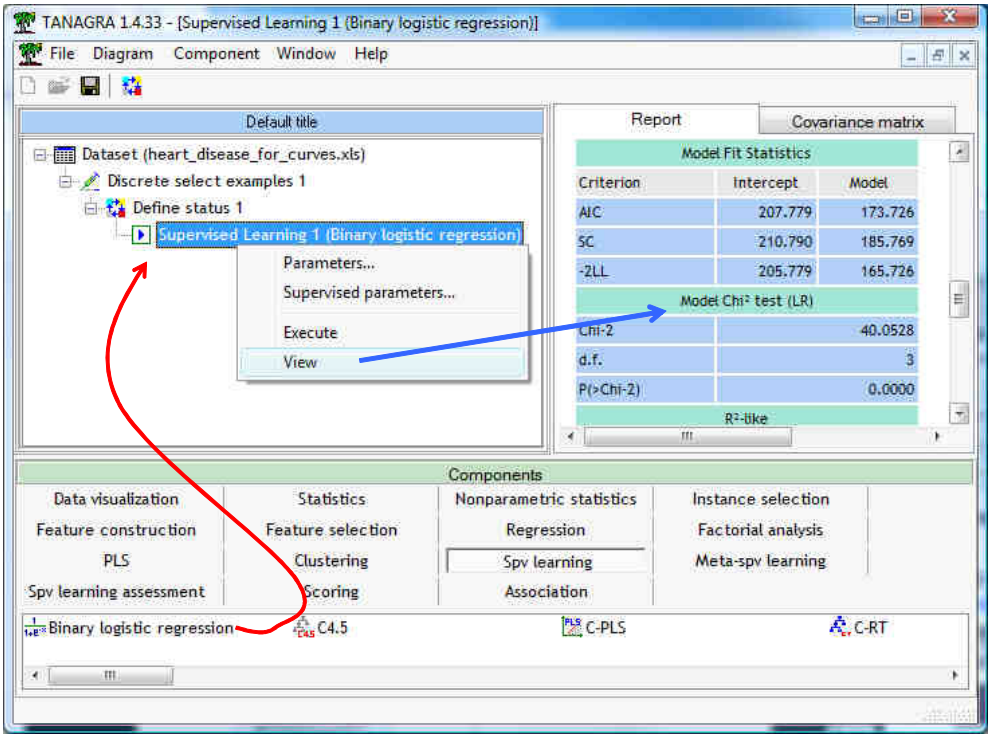
[7] Si l'on souhaite retrouver la configuration utilisée lors du calcul manuel des courbes sous Excel (250 en apprentissage, 20 en test ; section 1.3), il suffit d'intervertir les positions des feuilles à l'intérieur du classeur.

### 3.2. Apprentissage et attribution des scores

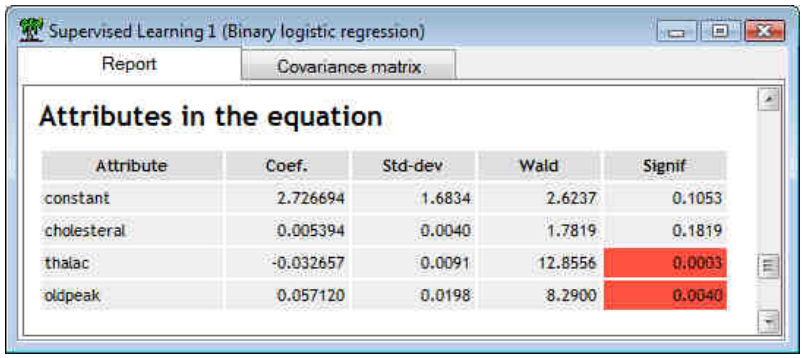
L'étape suivante est consacrée à la construction des modèles et à l'attribution des scores aux individus de l'échantillon test. Tout d'abord nous devons préciser le rôle des variables. Nous utilisons le composant DEFINE STATUS pour cela. Nous l'insérons dans le diagramme via le raccourci dans la barre d'outils. Nous plaçons DISEASE en TARGET ; CHOLESTERAL, THALAC en OLDPEAK en INPUT. La colonne SAMPLE n'est plus utilisée à ce stade.



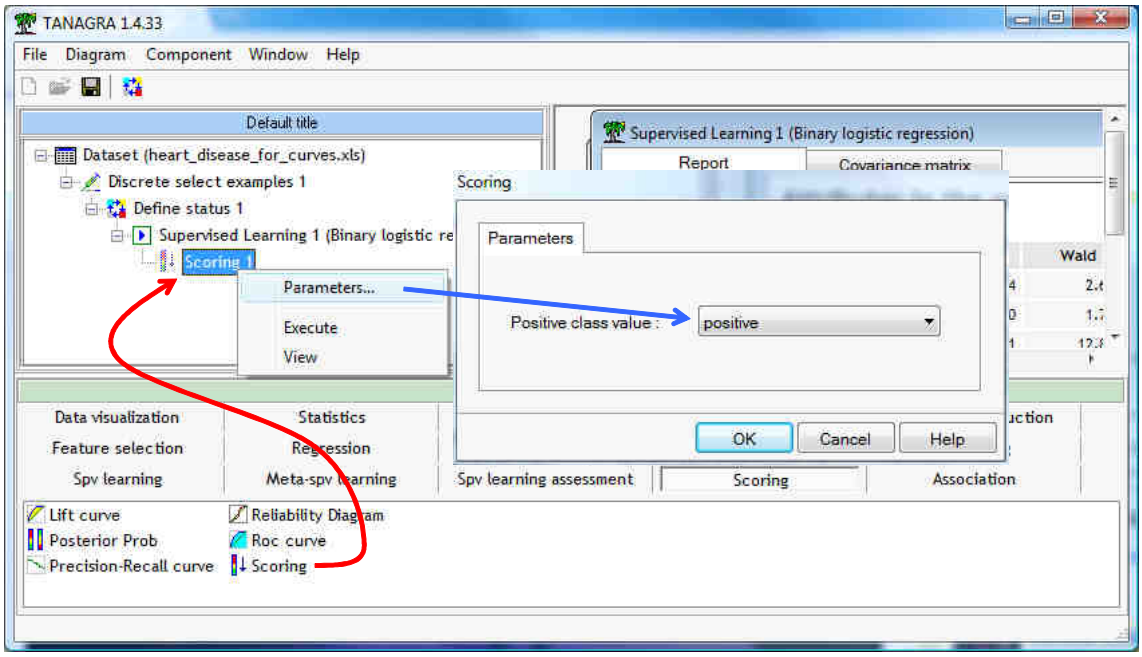

**Régression logistique**. Nous insérons le composant BINARY LOGISTIC REGRESSION dans le diagramme. Nous actionnons le menu VIEW pour obtenir les résultats.



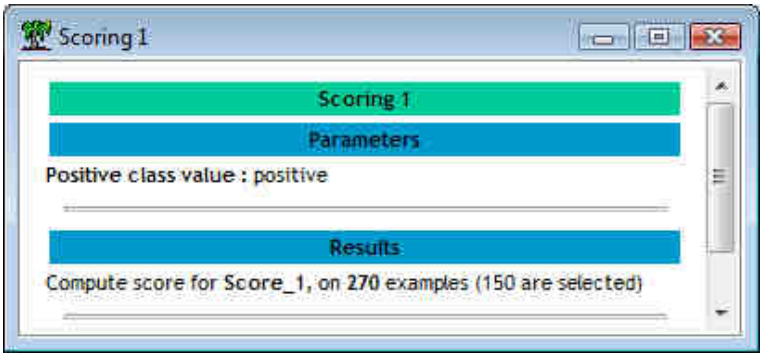
La régression est globalement significative à 5% avec un KHI-2 du test du rapport de vraisemblance de 40.05 et une probabilité critique < 0.0001. Les variables pertinentes, dont le coefficient est significativement non nul, sont THALAC et OLDPEAK.



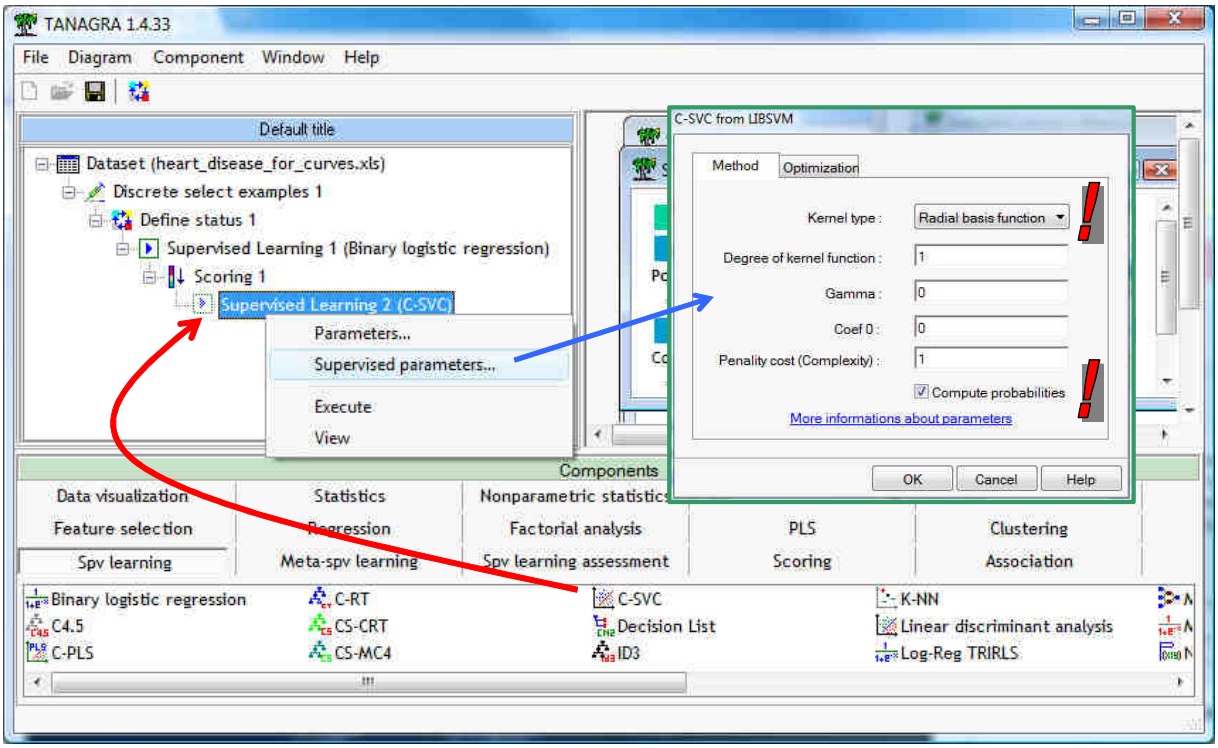

Nous devons calculer les scores de « positivité » des individus (la probabilité d'appartenance au groupe des positifs). Nous utilisons le composant SCORING (onglet SCORING). Nous le paramétrons de manière à ce que le composant calcule la probabilité pour la modalité POSITIVE de DISEASE.




Nous validons et nous cliquons sur VIEW. **Tanagra a calculé les scores pour toutes les observations de la base, qu'elles fassent partie de l'échantillon d'apprentissage ou de l'échantillon test**.

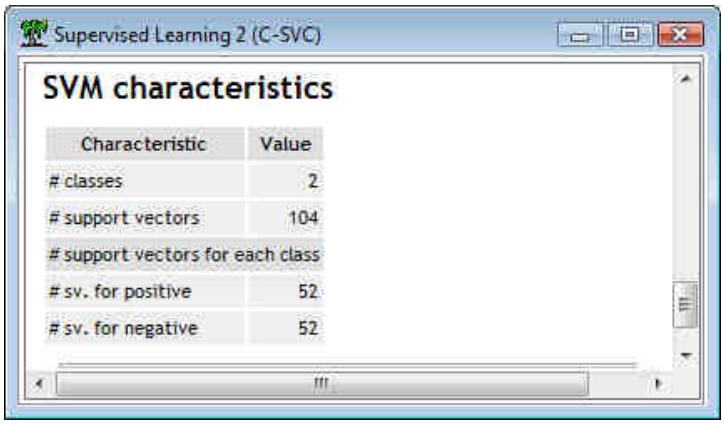


**Support vector machine**. Nous réitérons les mêmes opérations mais en utilisant les SVM avec un noyau RBF. Nous introduisons le composant C-SVC (onglet SPV LEARNING) dans le diagramme. C-SVC est une procédure en provenance de la fameuse librairie LIBSVM[8]. Nous le paramétrons en actionnant le menu contextuel SUPERVISED PARAMETERS.



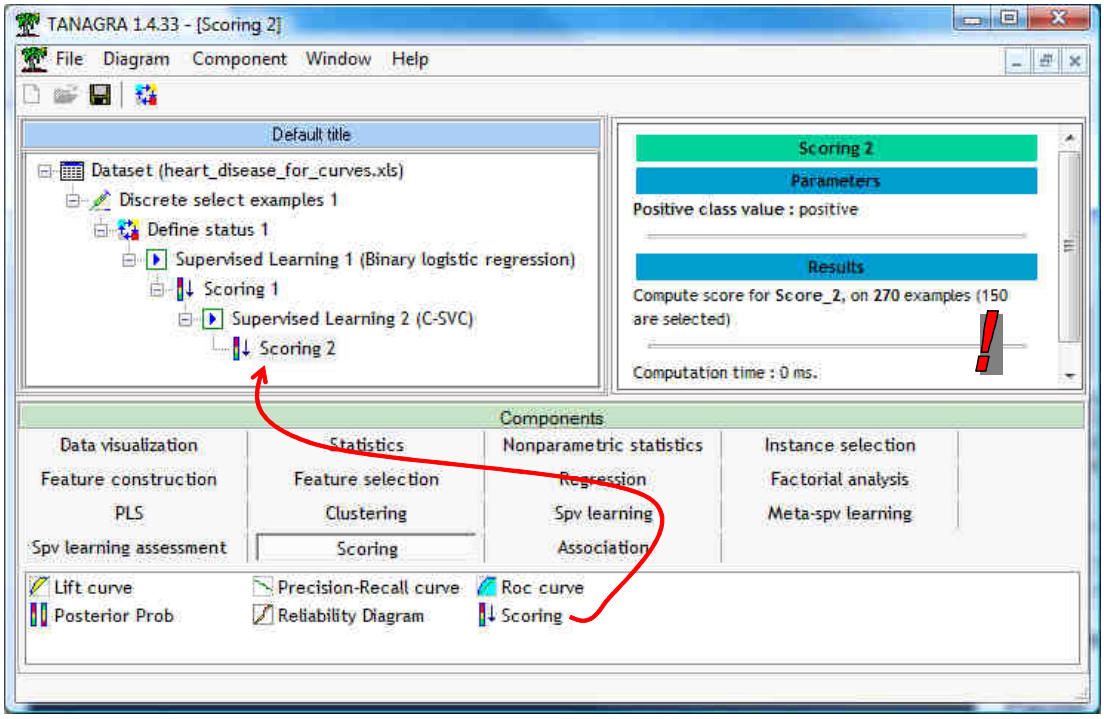

Nous demandons un noyau RBF (Radial Basis Function). Nous souhaitons également que les outils de calcul des probabilités d'affectation soient préparés.

Nous validons et nous cliquons sur VIEW. Les indications sont un peu laconiques, voire lapidaires.




Passons tout de suite au calcul des scores en insérant le composant SCORING. Nous le paramétrons toujours pour un calcul des probabilités a posteriori de la modalité POSITIVE de DISEASE.

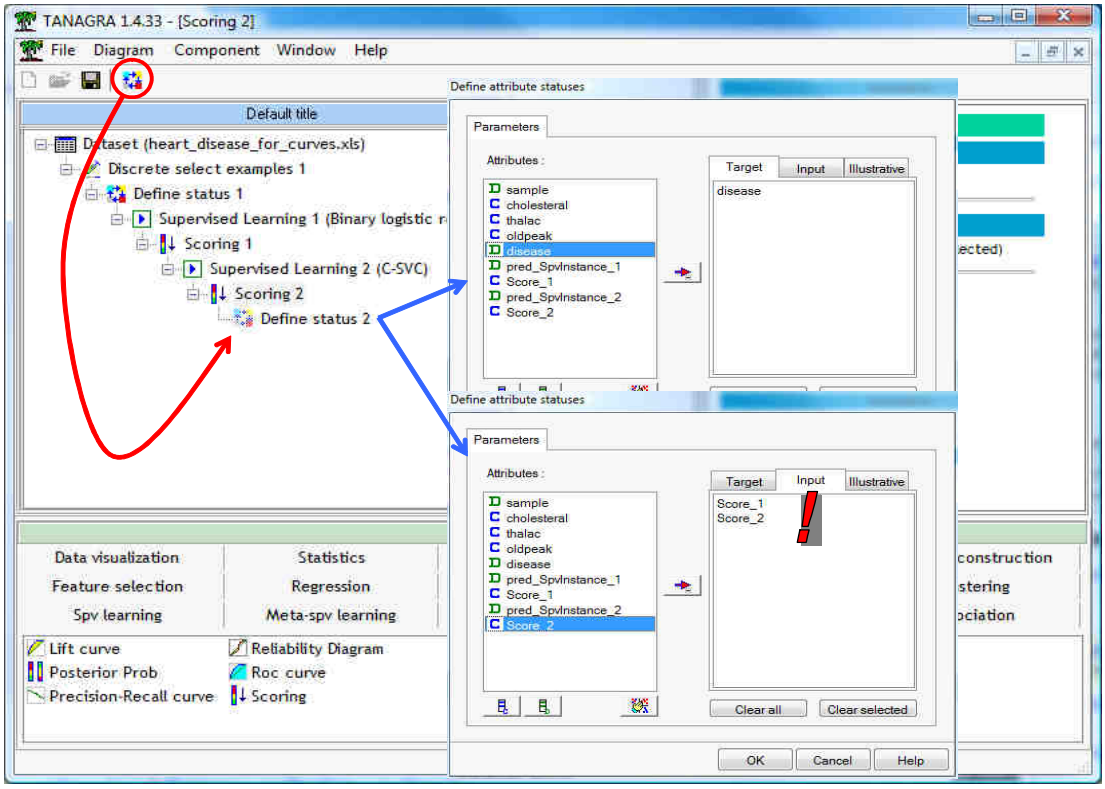
[8] http://www.csie.ntu.edu.tw/~cjlin/libsvm/




Nous validons le paramétrage et nous cliquons sur VIEW pour obtenir les scores.

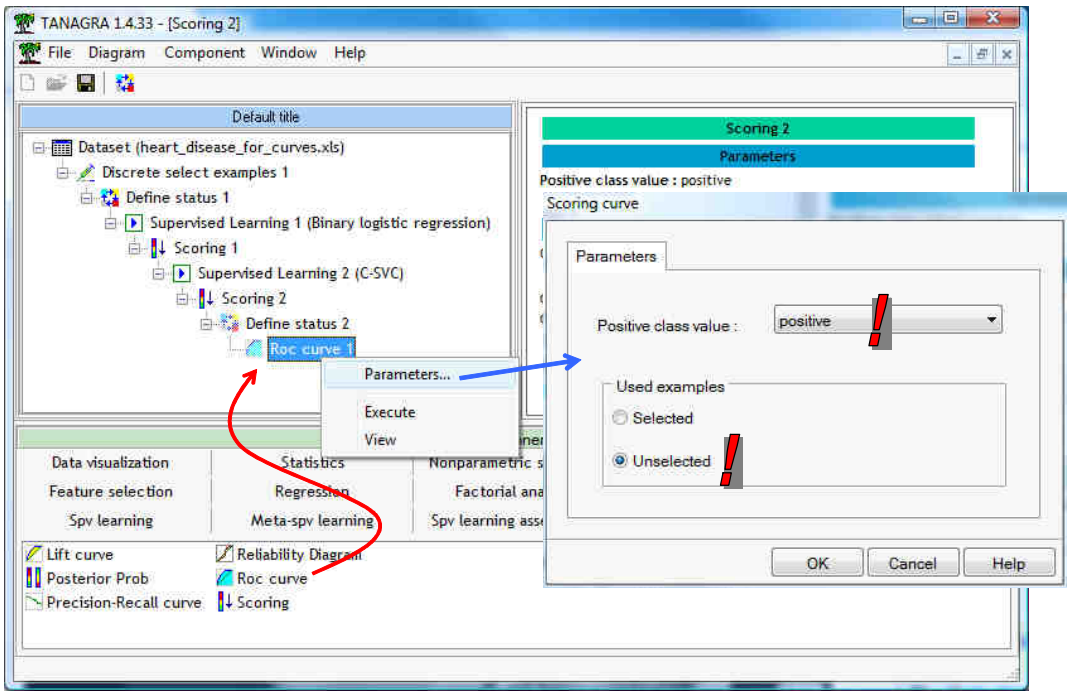
### 3.3. Elaboration de la courbe ROC

A ce stade, nous pouvons construire les courbes sur l'échantillon test et comparer les mérites respectifs de la régression logistique et des SVM. Tout d'abord, nous devons spécifier à Tanagra le nouveau rôle des variables. Nous introduisons le composant DEFINE STATUS. Nous plaçons DISEASE en TARGET ; les scores SCORE_1 (régression logistique) et SCORE_2 (SVM) en INPUT.

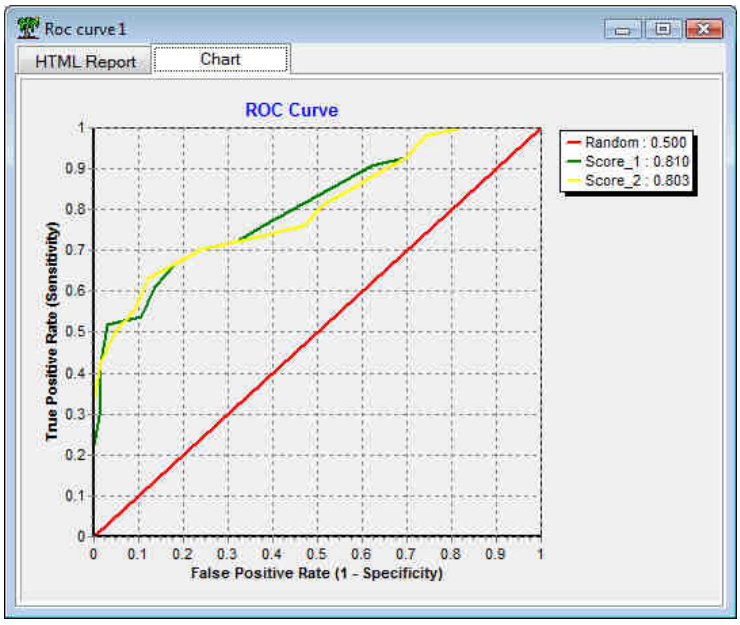


**Deux remarques importantes** : (1) Nous pouvons introduire autant de variables que l'on veut en INPUT. (2) Nous utilisons les scores fournis par l'apprentissage supervisé pour construire les courbes dans ce didacticiel. Mais en réalité n'importe quelle variable quantitative permettant d'ordonner les observations selon leur degré d'appartenance à la modalité positive de la variable à prédire conviendrait.

Nous insérons ensuite le composant ROC CURVE (onglet SCORING) dans le diagramme. Nous le paramétrons : d'une part pour lui indiquer la modalité positive de la variable TARGET, d'autre part pour lui spécifier qu'il doit réaliser les calculs sur les individus non sélectionnés précédemment c.-à-d. l'échantillon test.



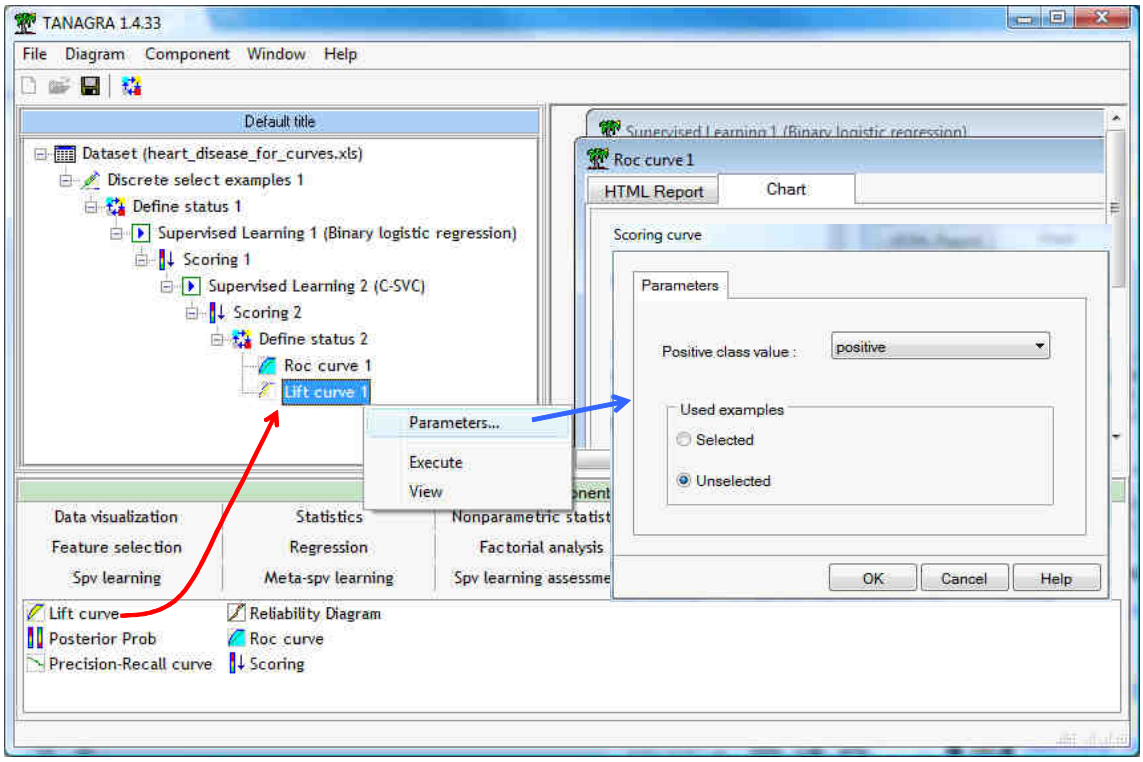

Nous validons et nous cliquons sur VIEW. Nous obtenons la courbe ROC.



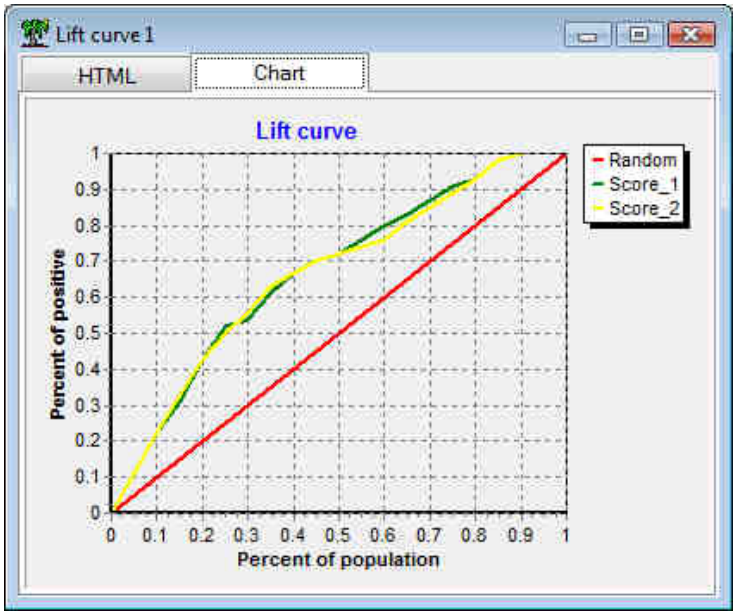
Les deux méthodes se valent sur notre fichier. Les deux courbes sont très proches. Tanagra fournit directement le critère AUC, nous avons AUC (régression logistique) = 0.810 et AUC (SVM) = 0.803.

### 3.4. Elaboration de la courbe de gain

Tout ayant été préalablement préparé, nous pouvons insérer directement le composant LIFT CURVE (onglet SCORING) dans le diagramme, en dessous de DEFINE STATUS 2, au même niveau que ROC CURVE 1. De nouveau, nous fixons, via la boîte de dialogue associée au menu PARAMETERS, les valeurs adéquates des paramètres (les mêmes que la courbe ROC).



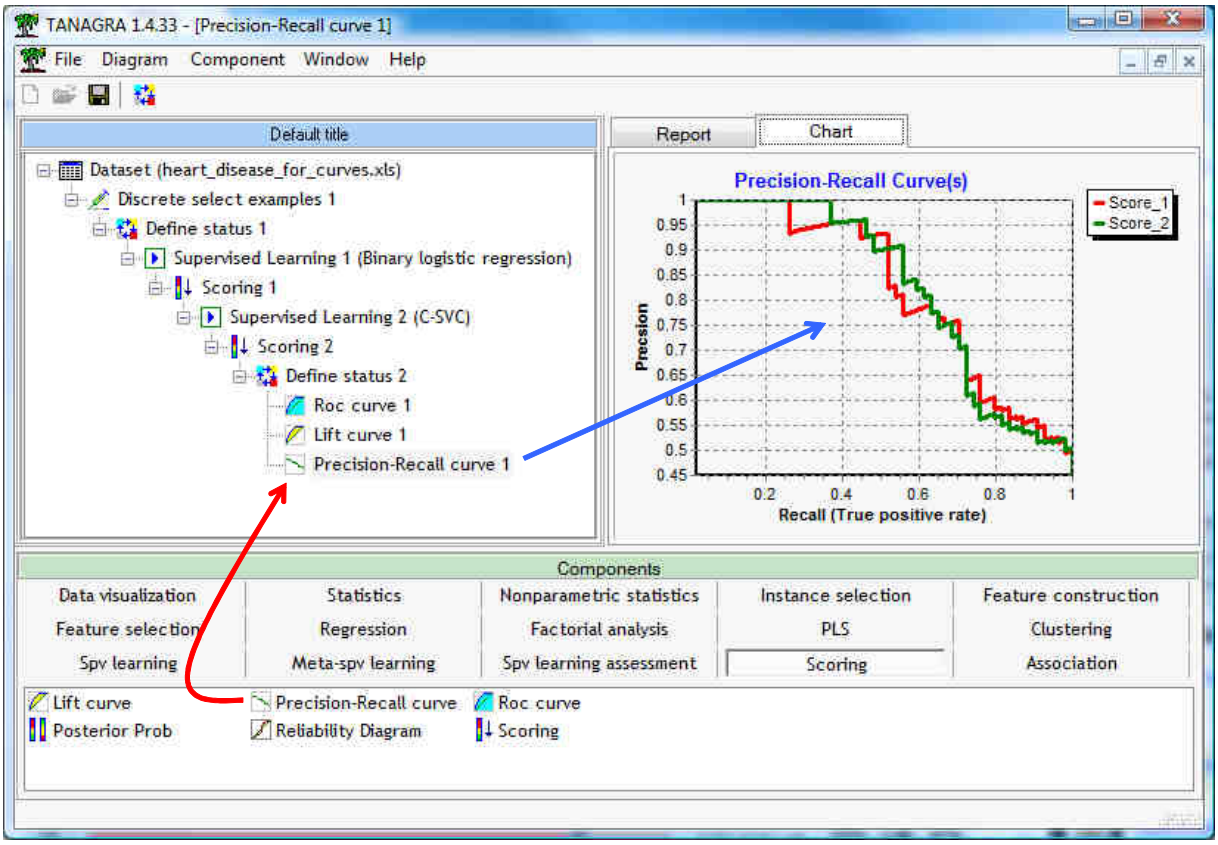

Il ne reste plus qu'à valider et cliquer sur VIEW.



Les conclusions sont les mêmes qu'avec la courbe ROC. Les deux méthodes se valent dans leur capacité à attribuer des scores élevés aux positifs, sur notre fichier tout du moins.

### 3.5. Elaboration de la courbe rappel – précision

Dernière courbe à construire, nous insérons le composant PRECISION RECALL CURVE (onglet SCORING) dans le diagramme. Nous spécifions les bons paramètres (les mêmes que précédemment) et nous validons.




L'allure globale de la courbe est très différente de deux précédentes, nous nous y attendions. Les conclusions restent les mêmes néanmoins : les performances sont du même ordre sur notre fichier de données.

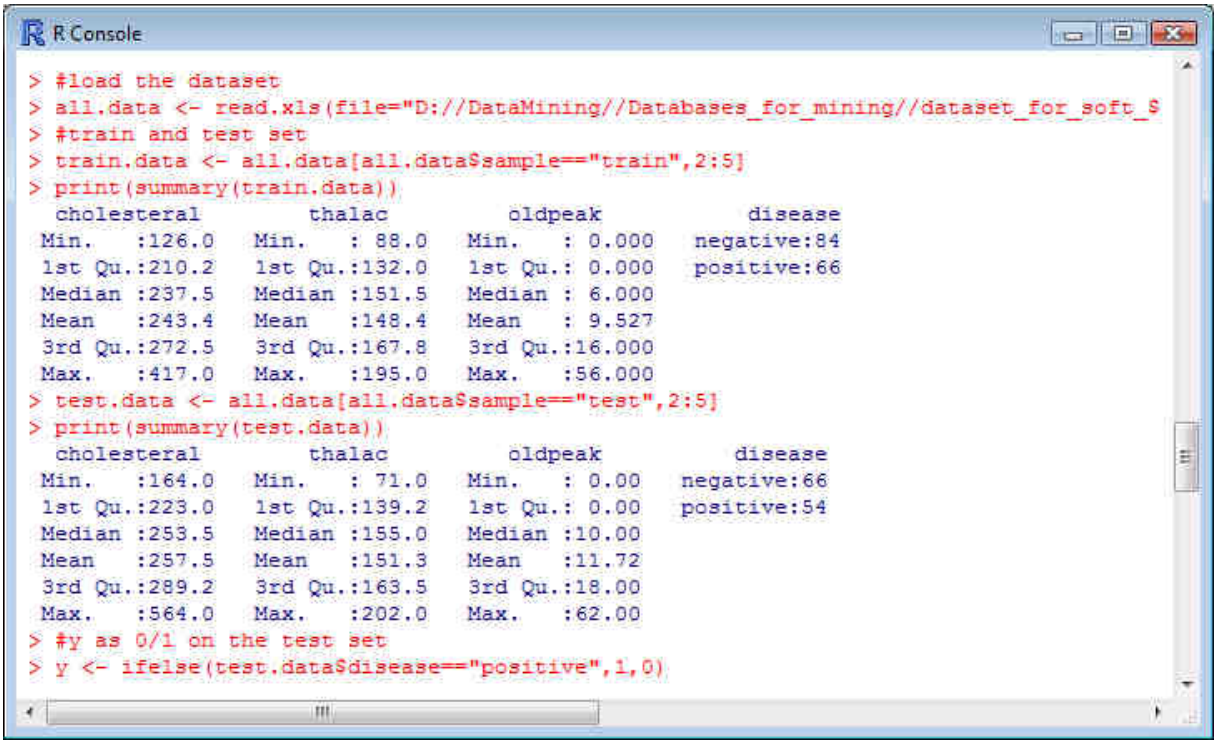
## 4. Construction des courbes avec R

La trame des opérations est exactement la même sous R. Nous ne commentons donc pas les résultats. Nous nous contentons d'afficher les commandes utilisées et les sorties du logiciel[9].

### 4.1. Importation et subdivision des données en apprentissage et test

Nous pouvons importer le fichier XLS à l'aide du package **xlsReadWrite**. Pour charger les données et préparer les échantillons, nous introduisons les commandes suivantes.

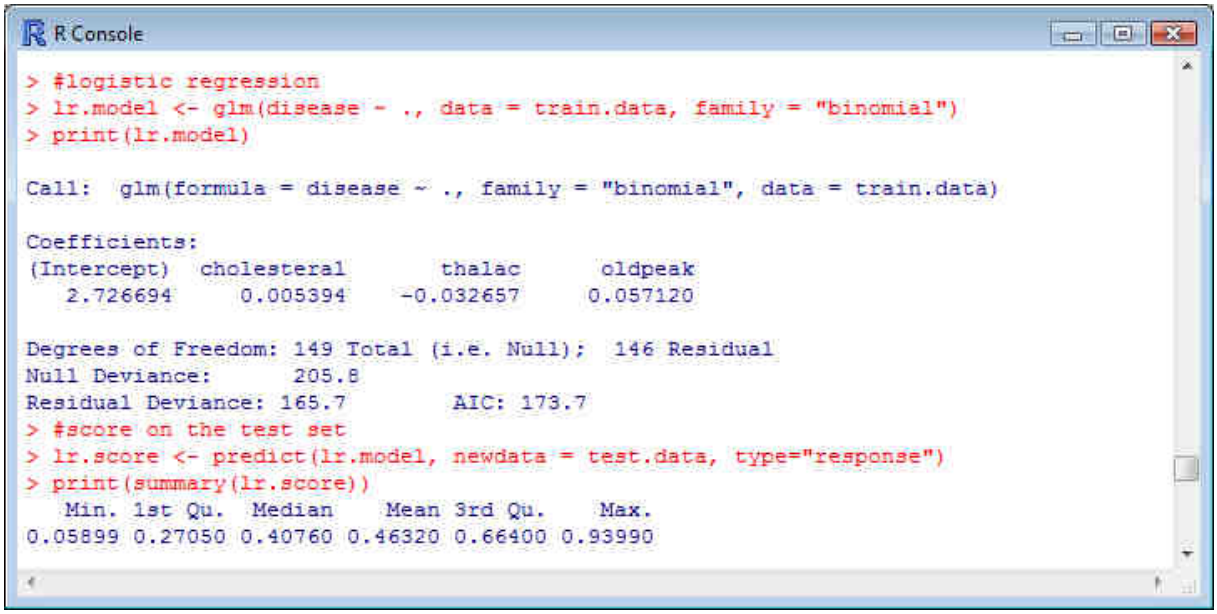
[9] Le code source R est inclus dans l'archive qui accompagne ce didacticiel.



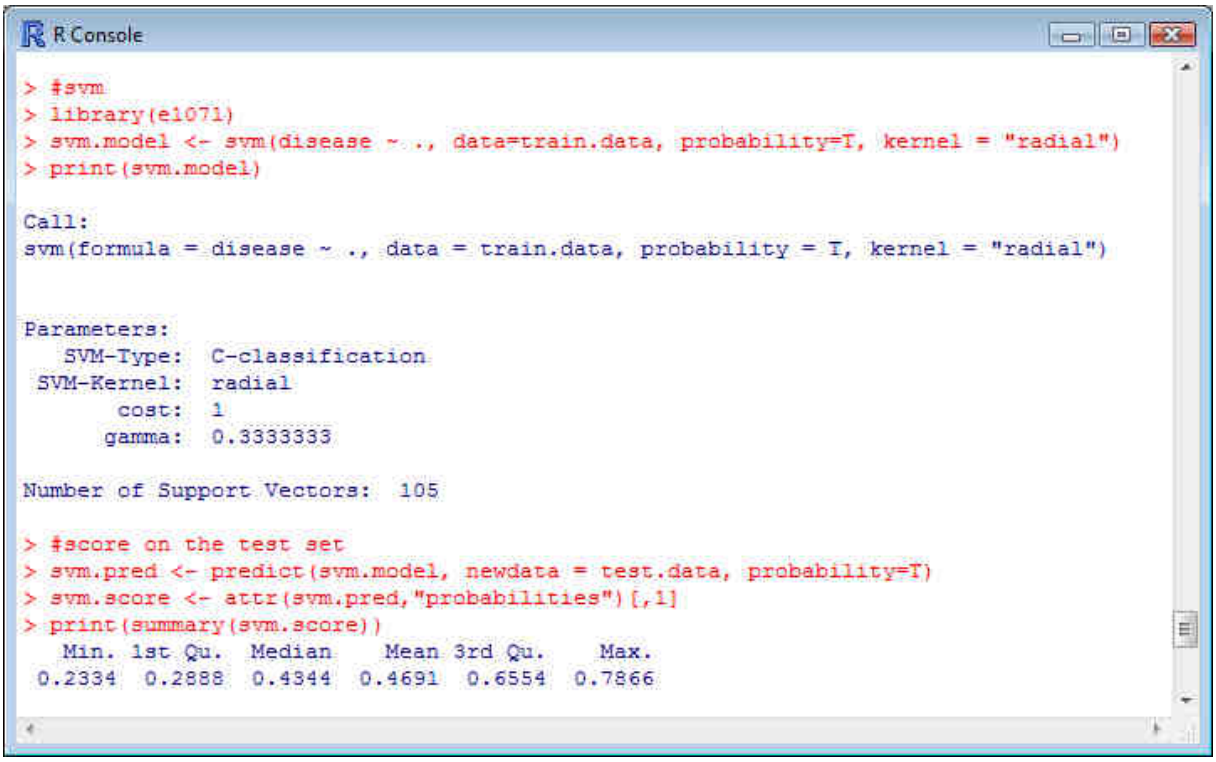

La dernière ligne de commande est destinée à coder la variable DISEASE (positif / négatif) en une variable Y binaire (1 / 0). Nous en aurons l'usage par la suite.

## 4.2. Apprentissage et attribution des scores

**Régression logistique**. Nous utilisons la fonction **glm()** pour réaliser la régression logistique sur l'échantillon d'apprentissage ; **predict()** permet ensuite de produire les scores pour l'échantillon test.



**Support vector machine**[10]. Concernant les SVM, nous utilisons la fonction **svm()** du package **e1071** basée sur la librairie LIBSVM. Autant que faire se peut (le nombre de paramètres que l'on peut manipuler est impressionnant), nous calquons le paramétrage sur celui de Tanagra.




### 4.3. Elaboration de la courbe ROC

Il ne nous reste plus qu'à élaborer les courbes ROC. Nous procédons en deux temps : (1) nous écrivons une fonction générique qui permet d'obtenir les coordonnées (abscisse et ordonnée) à partir de la variable à prédire Y codée 1/0 et des scores pour un échantillon quelconque ; (2) nous traçons les courbes dans le même repère pour comparer les algorithmes d'apprentissage.

La fonction générique est la suivante. Il faut bien évidemment que les deux vecteurs soient de même longueur. Nous privilégions une écriture très « scolaire » pour que tout un chacun puisse suivre le détail des opérations. Cette fonction est bien entendu réutilisable dans un contexte plus large, c'est pour cela que nous n'y avons pas intégré l'appel à la méthode d'apprentissage.

[10] Curieusement, nous n'avons pas exactement les mêmes résultats qu'avec Tanagra, pourtant censé utiliser la même bibliothèque LIBSVM. Nous avons 105 points supports dans R quand Tanagra en annonce 104. Les différences peuvent se situer à plusieurs niveaux : la version de la bibliothèque LIBSVM intégrée, Tanagra utilise la dernière en date (version 2.89 ; avril 2009) ; les valeurs par défaut des autres paramètres ; les valeurs d'initialisation de l'heuristique d'optimisation. Nous n'avons pas prise sur ce dernier élément.

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\curves_for_s...

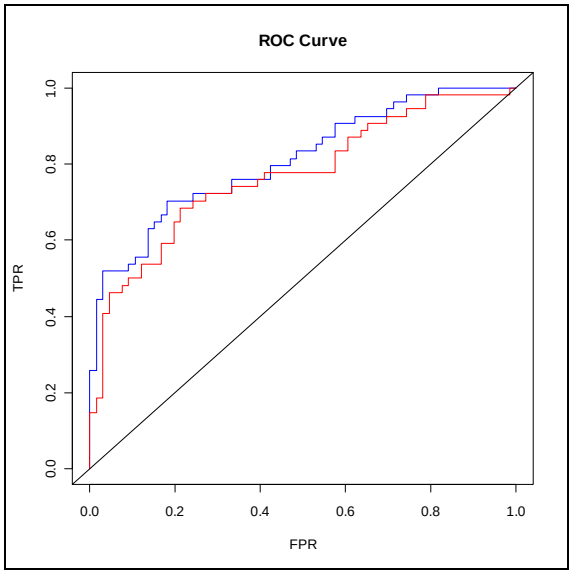
```
#function for roc curve
roc_curve <- function(y,score){
  #number of examples
  n <- length(y)
  #number of positive examples
  pos <- sum(y)
  #number of negative examples
  neg <- n - pos
  #size of the target
  target <- seq(1,n,1)
  #sorting values
  index <- sort(score,decreasing=T,index.return=T)
  sy <- y[index$ix]
  sscore <- score[index$ix]
  #cumulative number of positives
  c.pos <- cumsum(sy)
  #TPR - true positive rate
  tpr <- c.pos/pos
  tpr <- c(0,tpr)
  #cumulative number of negatives
  c.neg <- target - c.pos
  #FPR - false positive rate
  fpr <- c.neg/neg
  fpr <- c(0,fpr)
  #return values
  return(list(x=fpr,y=tpr))
}
```

L'appel de la fonction et le dessin des courbes s'obtiennent via les commandes suivantes.

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\curves_for_spv_learning\heart_di...

```
#ROC curve
lr.roc <- roc_curve(y,lr.score)
svm.roc <- roc_curve(y,svm.score)
plot(lr.roc$x,lr.roc$y,type="l",col="blue",main="ROC Curve",xlab="FPR",ylab="TPR")
lines(svm.roc$x,svm.roc$y,type="l",col="red")
abline(0,1)
```

Nous obtenons la courbe pour la régression logistique en bleu, en rouge celle pour les SVM.



## 4.4. Elaboration de la courbe de gain

Nous suivons la même trame pour la courbe de gain. La fonction s'écrit :

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\curves_for_spv_learn...

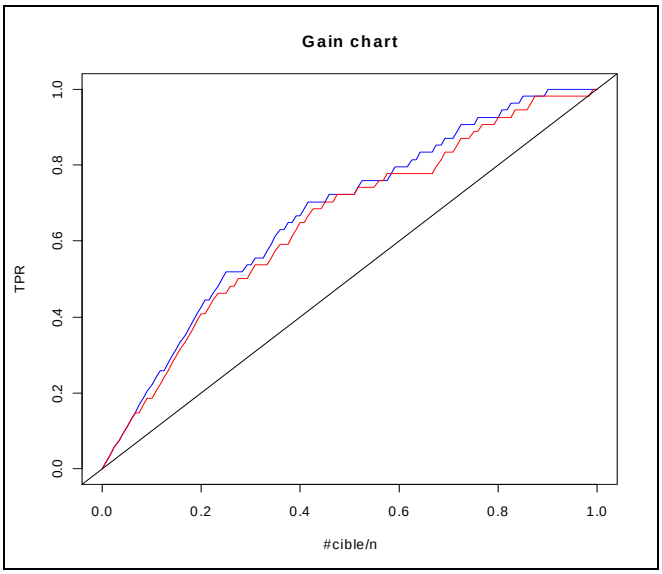
```
#function for cumulative lift curve
cum_lift_curve <- function(y,score){
  #number of examples
  n <- length(y)
  #number of positive examples
  pos <- sum(y)
  #size of the target
  target <- seq(1,n,1)
  #sorting values
  index <- sort(score,decreasing=T,index.return=T)
  sy <- y[index$ix]
  sscore <- score[index$ix]
  #cumulative number of positives
  c.pos <- cumsum(sy)
  #TPR - true positive rate
  tpr <- c.pos/pos
  tpr <- c(0,tpr)
  #relative size of the target
  rel.target <- target/n
  rel.target <- c(0,rel.target)
  #return values
  return(list(x=rel.target,y=tpr))
}
```

L'appel de la fonction et la confection des courbes.

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\curves_for_spv_learning\heart_disease_for_cu...

```
#Cumulative lift curve
lr.lift <- cum_lift_curve(y,lr.score)
svm.lift <- cum_lift_curve(y,svm.score)
plot(lr.lift$x,lr.lift$y,type="l",col="blue",main="Gain chart",xlab="#cible/n",ylab="TPR")
lines(svm.lift$x,svm.lift$y,type="l",col="red")
abline(0,1)
```

Pour obtenir finalement.



## 4.5. Elaboration de la courbe rappel – précision

Enfin, concernant la courbe rappel – précision, nous définissons la fonction suivante.

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\curves_for_sp...

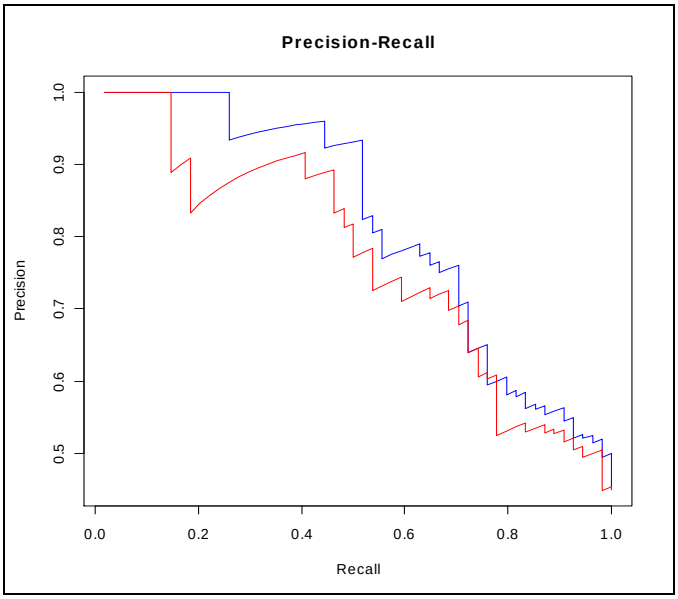
```
#function for recall-precision curve
recall_prec <- function(y,score){
  #number of examples
  n <- length(y)
  #number of positive examples
  pos <- sum(y)
  #size of the target
  target <- seq(1,n,1)
  #sorting values
  index <- sort(score,decreasing=T,index.return=T)
  sy <- y[index$ix]
  sscore <- score[index$ix]
  #cumulative number of positives
  c.pos <- cumsum(sy)
  #TPR - true positive rate
  tpr <- c.pos/pos
  #precision
  prec <- c.pos/target
  #return values
  return(list(x=tpr,y=prec))
}
```

L'appel de la fonction et l'élaboration du graphique.

D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\curves_for_spv_learning\heart_disease_for_curves.r*

```
#Recall precision curve
lr.rp <- recall_prec(y,lr.score)
svm.rp <- recall_prec(y,svm.score)
plot(lr.rp$x,lr.rp$y,type="l",col="blue",main="Precision-Recall",xlab="Recall",ylab="Precision")
lines(svm.rp$x,svm.rp$y,type="l",col="red")
```

Et nous obtenons.



# 5. Conclusion

Nous avons montré comment calculer quelques courbes destinées à évaluer les performances des classifieurs en apprentissage supervisé : « à la main » tout d'abord en détaillant les opérations dans Excel, puis à l'aide des logiciels Tanagra 1.4.33 et R 2.9.2.

Ces courbes sont plus riches que le simple taux d'erreur associé à une version unique de la matrice de confusion. En faisant varier le seuil d'affectation, nous pouvons définir toute une série de matrices de confusion et ainsi donner une vision plus large, globale, du comportement des classifieurs. En outre, selon les domaines d'étude, nous pouvons leur associer une interprétation opérationnelle. En scoring par exemple, la courbe de gain permet de quantifier précisément les performances à venir d'une campagne marketing.