1. Objectif

Estimation du taux d'erreur des méthodes supervisées par des techniques de ré échantillonnage.

L'évaluation des classifieurs est une question récurrente en apprentissage supervisé. Parmi les différents indicateurs existants, la performance en prédiction calculée à l'aide du taux d'erreur (ou son complémentaire à 1, le taux de bon classement) est un critère privilégié. Du moins dans les publications scientifiques car, dans les études réelles, d'autres considérations sont au moins aussi importantes : l'évaluation des performances en intégrant les coûts de mauvaise affectation, l'interprétation des résultats, les possibilités de mise en production, etc.

Le taux d'erreur théorique est défini comme la probabilité de mal classer un individu dans la population. Bien entendu, il est impossible de le calculer directement, essentiellement parce qu'il n'est pas possible d'accéder à toute la population. Nous devons produire une estimation. Qui dit estimation dit utilisation d'un échantillon, un estimateur de bonne qualité doit être le moins biaisé possible (en moyenne, on tombe sur la bonne valeur du taux d'erreur théorique), et le plus précis possible (la variabilité autour de la vraie valeur est petite).

L'estimateur trivial est le taux d'erreur empirique que l'on appelle également taux d'erreur en resubstitution. Il s'agit de ré appliquer le modèle sur l'échantillon de données qui a servi à le construire. Tous les logiciels produisent cette estimation accompagnée d'un tableau de contingence, dite matrice de confusion, qui croise, pour l'ensemble des individus de l'échantillon, la vraie modalité prise par la variable à prédire et la modalité affectée par le modèle de classement. Le principal reproche que l'on peut adresser à l'erreur en resubstitution est qu'elle est fortement biaisée, on parle de « biais d'optimisme ». En effet, elle sous estime souvent le taux d'erreur théorique. La raison est simple, le fichier de données est à la fois « partie », il a servi à construire le modèle, et « juge », il est utilisé pour savoir si le modèle classe correctement. On montre que plus une observation pèse sur sa prédiction, plus l'optimisme sera important. L'exemple extrême est la méthode du plus proche voisin (1-ppv), le plus proche voisin d'un point est lui-même, le taux d'erreur en resubstitution est mécaniquement égal à zéro (dans un espace de description continu, c.-à-d. lorsque la probabilité que deux observations aient la même description est nulle). De manière générale, il y a un fort optimisme lorsque les techniques « collent » exagérément aux données (ex. un Perceptron avec trop de neurones dans la couche cachée, un arbre de décision trop grand) ou lorsque la dimensionnalité est trop importante au regard du nombre d'observations.

Pour se dégager de cet écueil, on conseille souvent de subdiviser l'échantillon en 2 parties : une première partie, dit fichier d'apprentissage, utilisée pour construire le modèle ; une seconde partie, dit fichier test, utilisée pour évaluer les performances du modèle. L'erreur ainsi mesurée est appelée « erreur en test ». Elle estime de manière non biaisée l'erreur théorique. Tout serait donc parfait si nous ne sommes pas confronté à un nouveau problème : quelle proportion des données devons nous consacrer à l'apprentissage ? La pratique veut que l'on réserve entre 60% et 70% pour l'apprentissage. Mais au delà de cette règle empirique, nous devons arbitrer entre deux exigences contradictoires, d'autant plus crucial que l'échantillon est de petite taille : plus nous réservons des

données pour l'apprentissage, moins l'estimation de l'erreur en test sera précise ; si nous favorisons la partie test, nous pénalisons l'apprentissage, nous retirons de l'information qui peut s'avérer déterminante pour la construction d'un modèle efficace.

Peut-on quantifier « échantillon de petite taille » si nous devons travailler sur un problème réel ? En deçà du millier d'observations, on pourrait considérer que la base est de petite taille. En réalité, il faut surtout appréhender le problème sous l'angle du rapport entre la complexité du modèle (ou le nombre de variables s'il s'agit par exemple d'un modèle linéaire) et le nombre d'observations.

Dans un contexte où les observations sont (relativement) rares, comment estimer au mieux le taux d'erreur théorique si nous souhaitons consacrer l'ensemble du fichier à la construction du modèle ?

Les techniques de ré échantillonnage permettent de répondre à cette question. Nous étudierons plus particulièrement la **validation croisée**, le **leave one out**, et le **bootstrap**. Il s'agit de répéter plusieurs fois, sous des configurations pré définies, le schéma apprentissage test. Attention, il s'agit bien d'une estimation de l'erreur du modèle construit sur l'ensemble des données. Les modèles intermédiaires, élaborés lors des apprentissages répétés, servent uniquement à l'évaluation de l'erreur. Ils ne sont pas accessibles à l'utilisateur, ils n'ont pas d'utilité intrinsèque.

Ce didacticiel vise à compléter le cours décrivant les techniques de ré échantillonnage accessible à l'adresse suivante : http://eric.univ-lyon2.fr/~ricco/cours/slides/resampling_evaluation.pdf. Nous utiliserons l'analyse discriminante linéaire (LDA) pour illustrer notre propos.

2. Données

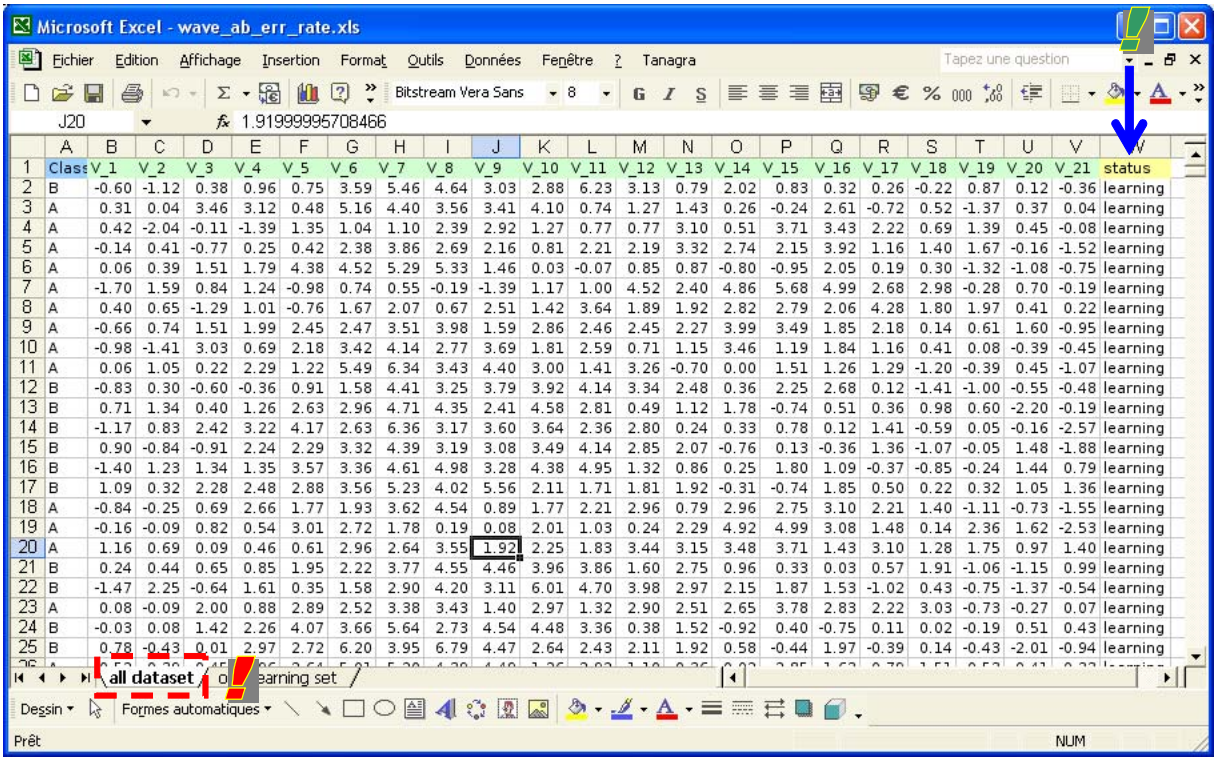
Nous utilisons les ondes de Breiman (Breiman et al., 1984) dans ce didacticiel. L'objectif est de prédire la catégorie d'un objet à partir de 21 mesures. Le fichier originel comporte 3 catégories d'objets, nous nous contenterons de la reconnaissance des 2 premières classes dans ce didacticiel.

Son principal intérêt est qu'il s'agit de données générées. Pour évaluer un modèle construit sur un échantillon de données, nous pouvons donc produire des observations en profusion, suffisamment en tous les cas pour obtenir une estimation précise de l'erreur théorique. Nous proposons donc le schéma d'expérimentation suivant :

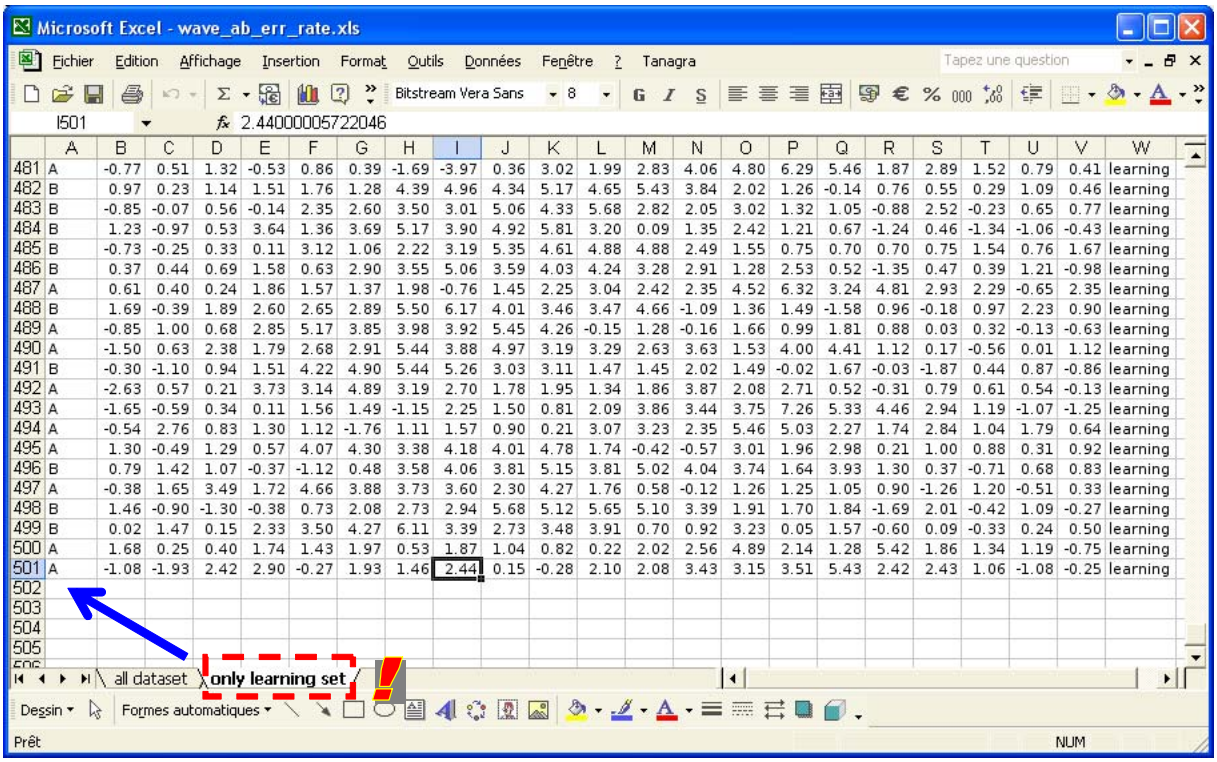
- 500 observations constituent le fichier d'apprentissage, nous l'utiliserons pour construire le modèle de prédiction LDA(500). Nous pouvons mesurer déjà l'erreur en resubstitution (*e-resub*).
- 42500 observations constituent le fichier test, nous l'utiliserons pour obtenir une estimation non biaisée de l'erreur théorique de LDA(500) (*e-test*). On peut penser que le nombre d'observations est suffisamment élevé pour obtenir une estimation précise.
- Lors d'une étude réelle, ces données dites de test ne sont pas disponibles (surtout en telle quantité). Nous ne pouvons utiliser que les 500 observations pour construire le modèle et en évaluer les performances : nous utiliserons pour cela, tour à tour, les techniques énumérées ci-dessus. Nous comparerons les résultats (*e-lvo*, *e-cv*, *e-boot*) avec la référence que constitue *e-test*.

Le classeur EXCEL (wave_ab_err_rate.xls) qui accompagne ce didacticiel est subdivisé en 2 feuilles :

- « all dataset » contient 43000 observations, avec une variable supplémentaire indiquant le statut des observations (status = apprentissage ou test) ;

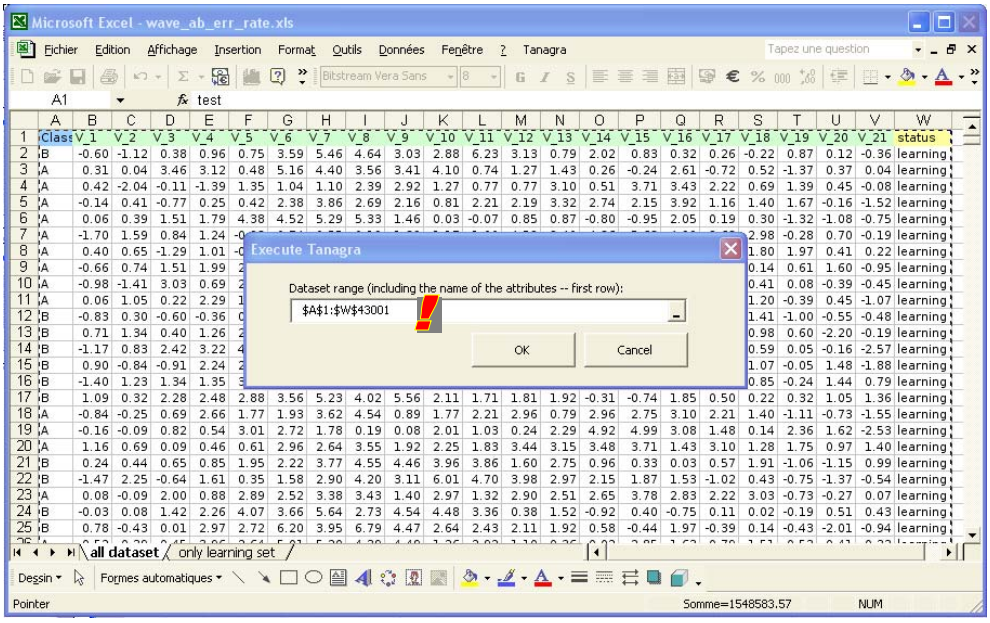


- « only learning set » ne contient que les 500 premières observations de la première feuille, correspondant à la fraction (status = apprentissage).

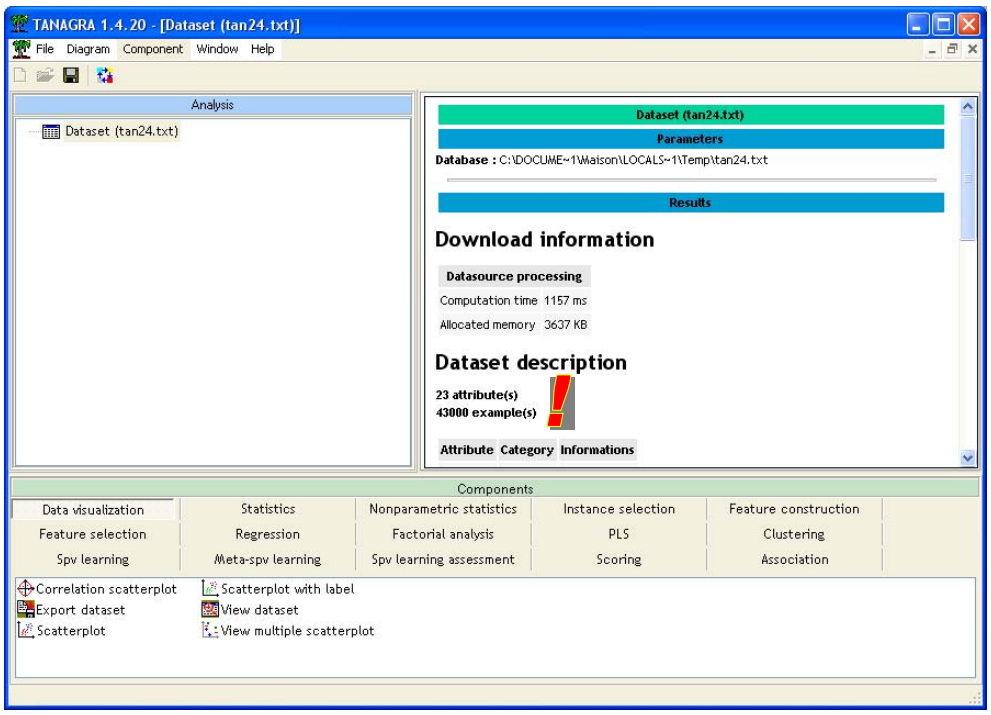


3. Apprentissage, erreur en resubstitution et erreur en test

Dans un premier temps, intéressons nous à la première feuille « all dataset ». Nous sélectionnons la plage de données et nous activons le menu TANAGRA/EXECUTE TANAGRA rendue disponible par la macro complémentaire[1] « TANAGRA.XLA ».



Après vérification des coordonnées de la plage de données, nous validons en cliquant sur le bouton OK. TANAGRA est automatiquement démarré. Les données sont chargées.

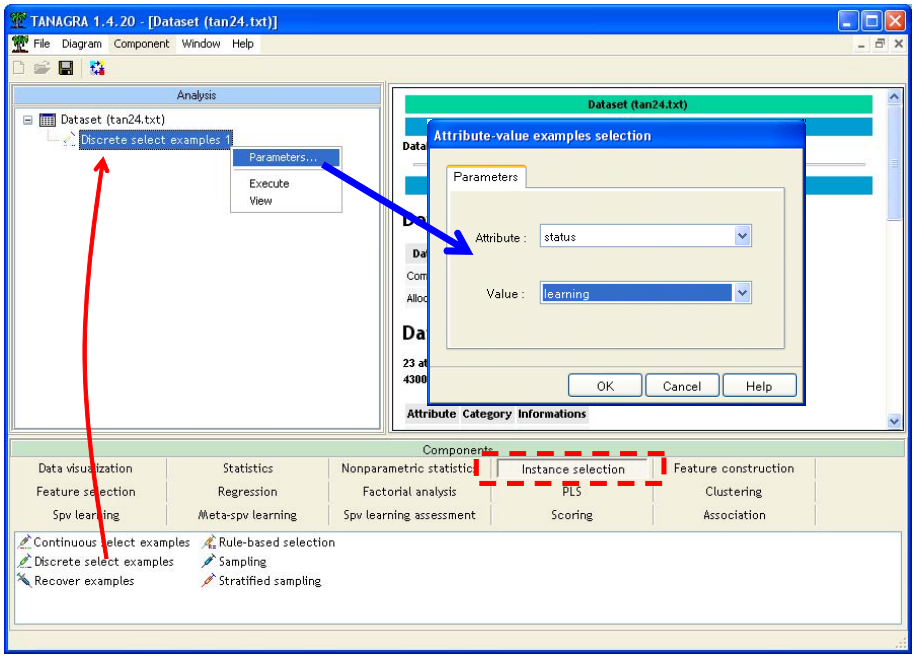


[1] Disponible depuis la version 1.4.11 de TANAGRA, voir le didacticiel http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_Excel_AddIn.pdf pour l'installation et l'utilisation de cette macro.

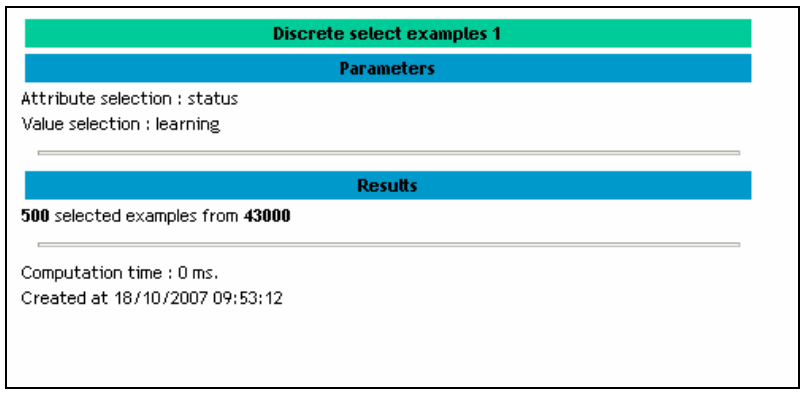
Nous devons avoir 23 variables (variable à prédire CLASS, les 21 descripteurs V1 à V21 et la variable STATUS) et 43000 observations.

Découpage apprentissage – test des données

Dans un premier temps, nous devons spécifier quelles seront les données utilisées pour la construction du modèle (données d'apprentissage) et celles qui serviront à l'évaluer (données test). Pour cela, nous utilisons le composant DISCRETE SELECT EXAMPLES (onglet INSTANCE SELECTION), il s'appuiera sur la variable supplémentaire STATUS pour subdiviser le fichier. Nous l'insérons dans le diagramme par glisser déplacer, puis nous activons le menu contextuel PARAMETERS pour indiquer que les données à sélectionner correspondent à la modalité LEARNING de STATUS.



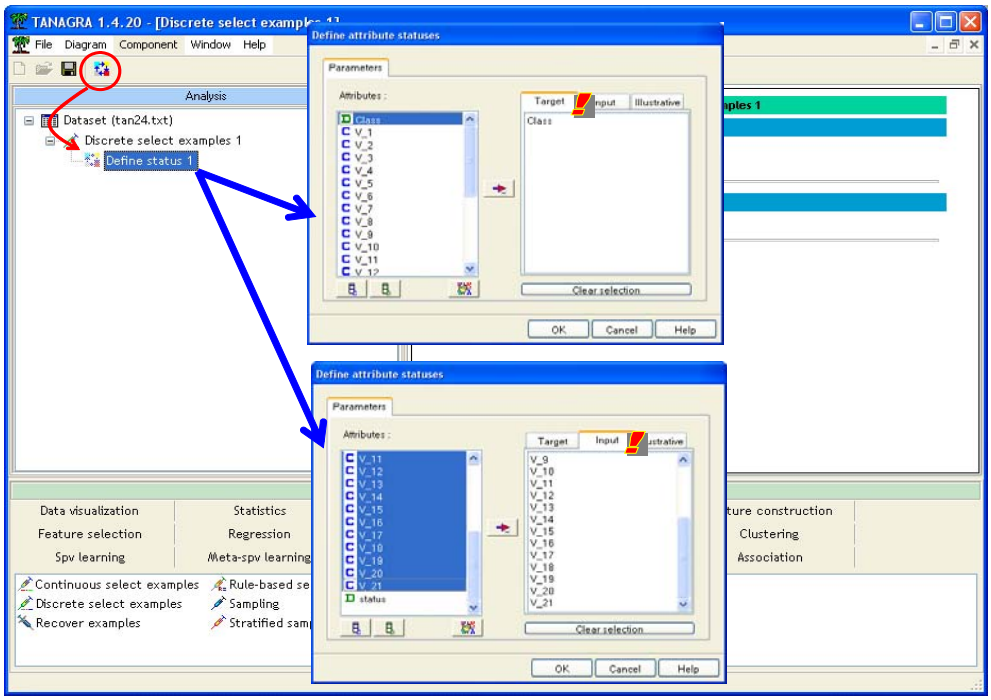
Après validation (bouton OK), nous cliquons sur le menu contextuel VIEW pour accéder aux résultats. TANAGRA nous indique bien que 500 observations sont sélectionnées pour les analyses maintenant.



Variable à prédire et variables prédictives

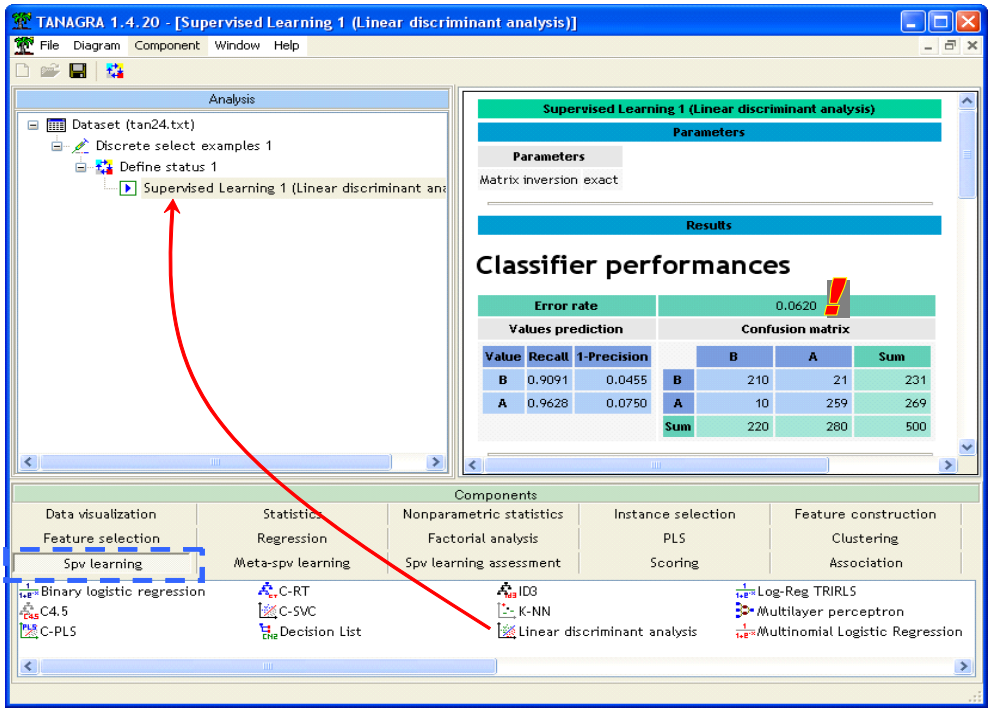
Il s'agit de préciser quelles sont les variables prédictives (INPUT) et la variable à prédire (TARGET) à l'aide du composant DEFINE STATUS. Le plus simple est de passer par le raccourci disponible dans la barre d'outils, la fenêtre de paramétrage apparaît automatiquement, nous plaçons en TARGET la

variable CLASS, en INPUT les variables V1 à V21. Bien entendu, la variable STATUS n'est pas utilisée ici.



Analyse discriminante et erreur en resubstitution

Nous pouvons construire le modèle de prédiction. Pour cela, nous plaçons dans le diagramme le composant LINEAR DISCRIMINANT ANALYSIS (onglet SPV LEARNING), toujours par glisser déplacer. Nous activons le menu VIEW pour accéder aux résultats. Qu'importe le modèle dans ce didacticiel, c'est la partie haute de la fenêtre de résultats qui nous intéresse au premier chef, il décrit la matrice de confusion et fournit l'erreur en resubstitution.

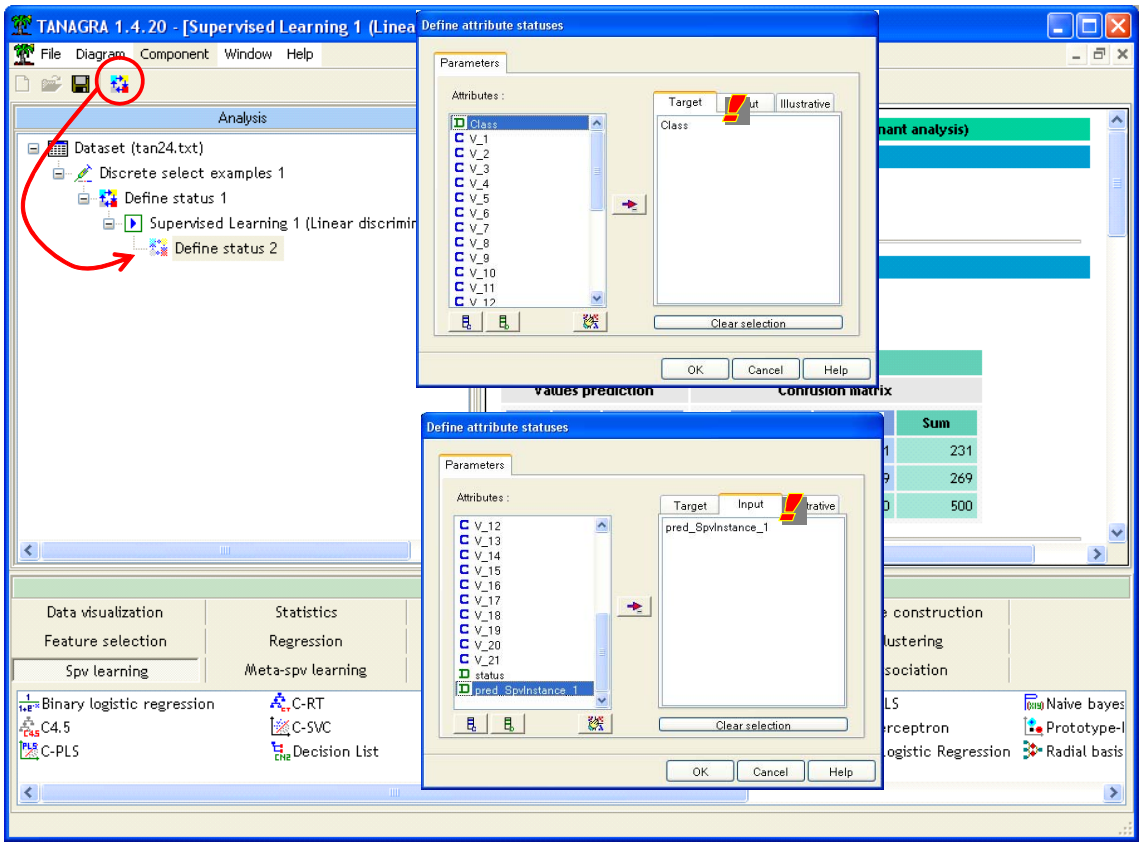


L'erreur en resubstitution est e-resub = 6.2%. Si on se fie à ce chiffre, lorsque nous classons une nouvelle observation à partir du modèle fourni par la LDA, nous avons 6.2% de chances de réaliser une mauvaise affectation. En dehors de toute considération experte, nous nous bornerons à dire que sans ce modèle, la meilleure prédiction serait d'affecter systématiquement CLASS = A (la modalité la plus fréquente de la variable à prédire), et dans ce cas le taux d'erreur serait 231/500 = 46.2% ; nous lisons cela dans la marge ligne de la matrice de confusion. A priori, nous avons un modèle d'excellente qualité.

Erreur en test

Pour le vérifier, nous allons mesurer les performances du même modèle sur les 42500 observations que nous avons mis de côté. Les composants « apprentissage supervisé » génèrent automatiquement une variable supplémentaire qui correspond à la prédiction du modèle, calculé sur la totalité du fichier, sur les données d'apprentissage donc, mais aussi sur les données mises de côté. Nous allons exploiter cette propriété.

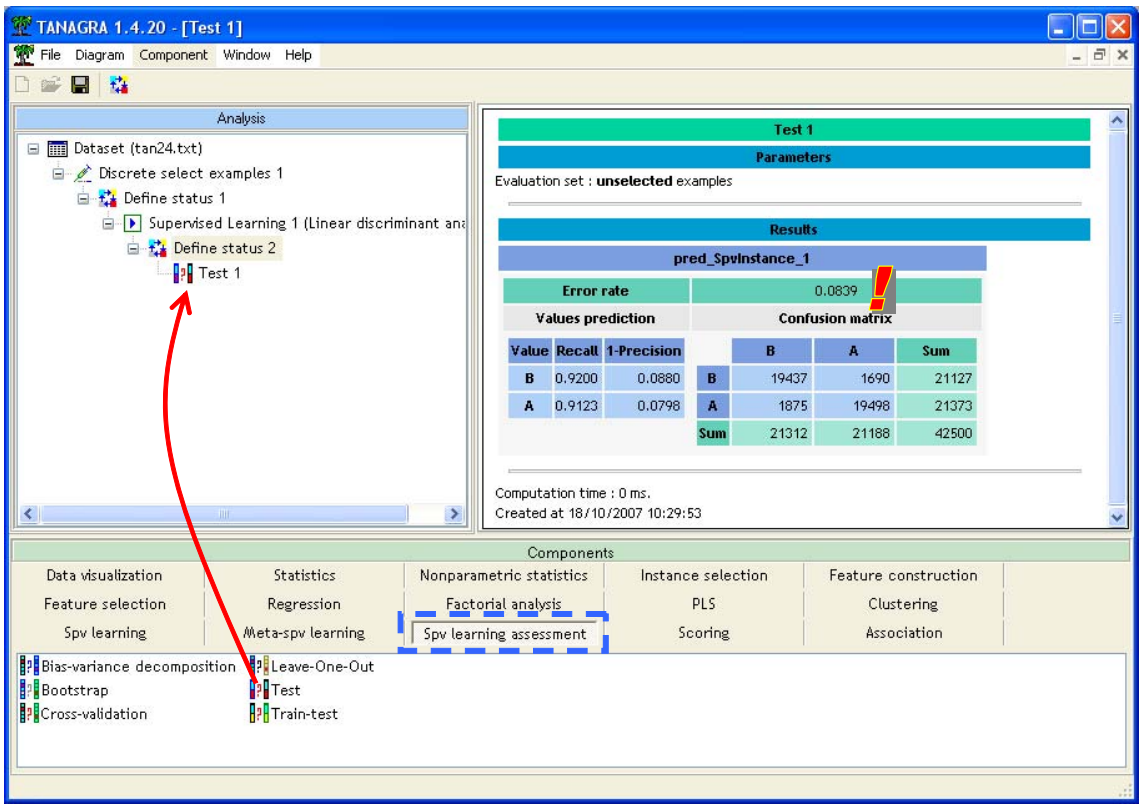
Pour cela, nous insérons de nouveau le composant DEFINE STATUS, nous plaçons en TARGET la variable CLASS, et en INPUT, la variable nouvellement générée PRED_SPV_INSTANCE_1.



Nous insérons ensuite le composant TEST (onglet SPV LEARNING ASSESSMENT) qui mesure le taux d'erreur en croisant les variables en TARGET et INPUT. Nous activons le menu VIEW pour obtenir les résultats.

Remarque 1 : Nous avons la possibilité de placer plusieurs variables en INPUT, cela permet de comparer les performances de plusieurs modèles.

Remarque 2 : Le composant est automatiquement paramétré pour que le calcul du taux d'erreur soit réalisé sur les données supplémentaires, mises de côté de la sélection des individus pour l'apprentissage. C'est ce que nous souhaitons faire ici. Nous avons néanmoins la possibilité de modifier ce paramétrage pour effectuer les calculs sur les données d'apprentissage. Dans ce cas, nous devons obtenir de nouveau l'erreur en resubstitution.



Le « véritable »[2] taux d'erreur de notre modèle de prédiction est en réalité de 8.39%. Il y a quand même un décalage par rapport à l'erreur en resubstitution évaluée précédemment (6.2%).

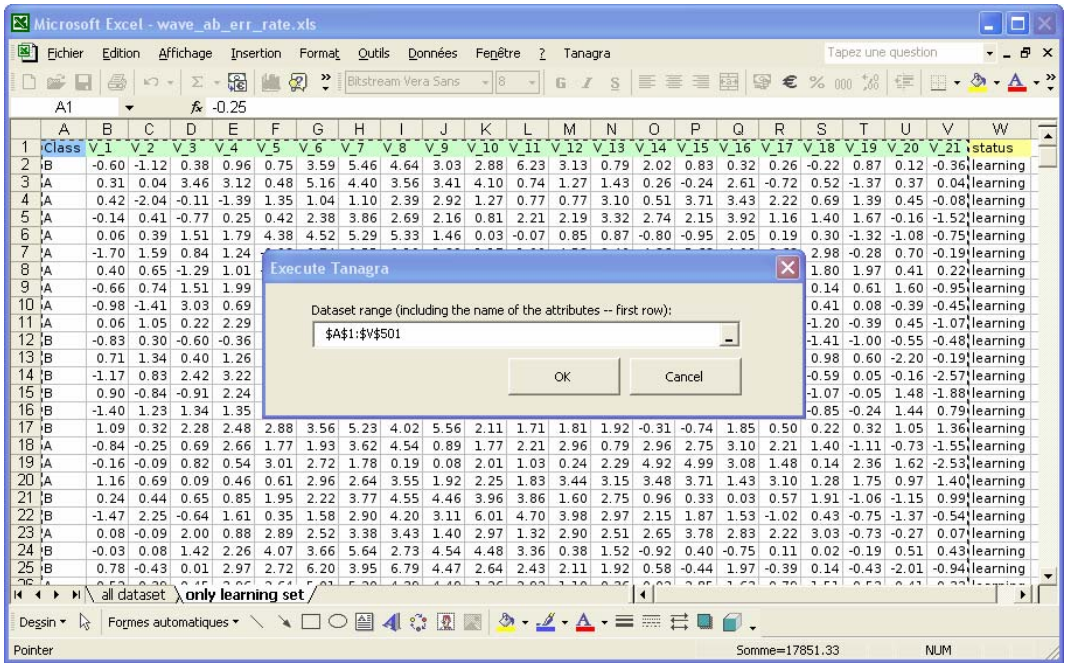
Remarque 3 : Et encore, l'analyse discriminante est une technique assez stable. Nous aurions utilisé un arbre de décision, type C4.5, qui a parfois une aptitude fâcheuse à épouser les données au plus près, les performances ne seraient pas du même ordre (e-test = 13.54%), et surtout le décalage serait plus important (e-resub = 2.2%). Nous y reviendrons à la fin de ce tutoriel.

4. Méthodes de ré échantillonnage

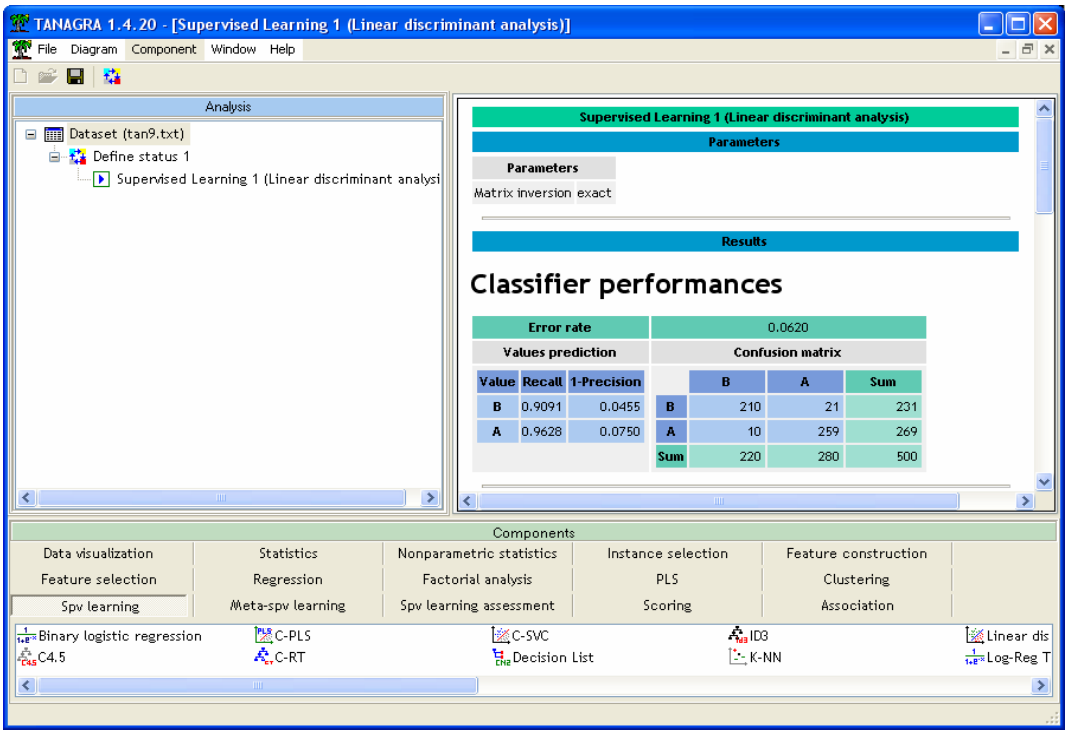
Nous pouvons clôturer cette session de travail en fermant TANAGRA. Revenons dans le classeur EXCEL, sélectionnons maintenant la seconde feuille « ONLY LEARNING SET ». En situation d'étude réelle, nous ne disposons que de ces 500 observations. Comme nous le disions plus haut, nous devons les utiliser à la fois pour construire le modèle et en mesurer les performances. Pour cela, nous allons mettre en œuvre les techniques de ré échantillonnage disponibles dans TANAGRA.

[2] Même si notre ensemble test comporte 42500 observations, il s'agit quand même d'un échantillon. L'erreur mesurée n'est qu'une estimation. On peut néanmoins penser qu'elle est assez précise au regard de la taille de l'échantillon test. C'est à ce titre que nous l'utiliserons comme référence pour les techniques de ré échantillonnage présentées par la suite.

Dans un premier temps, comme précédemment, nous sélectionnons la plage de données dans EXCEL, puis nous activons le menu TANAGRA/EXECUTE TANAGRA. Bien sûr, il est totalement inutile de sélectionner la dernière colonne STATUS, elle est constituée de la même valeur « LEARNING » pour toutes les lignes du tableau. Nous vérifions que la plage de cellules sélectionnée correspond bien à **$A$1:$V$501**



TANAGRA est automatiquement démarré, nous définissons les variables en TARGET (CLASS) et INPUT (V1 à V21). Puis nous insérons le composant LINEAR DISCRIMINANT ANALYSIS. Le modèle et l'erreur en resubstitution (6.2%) sont exactement les mêmes que précédemment. C'est normal. Le calcul est complètement déterministe et nous utilisons exactement les mêmes données pour l'apprentissage.



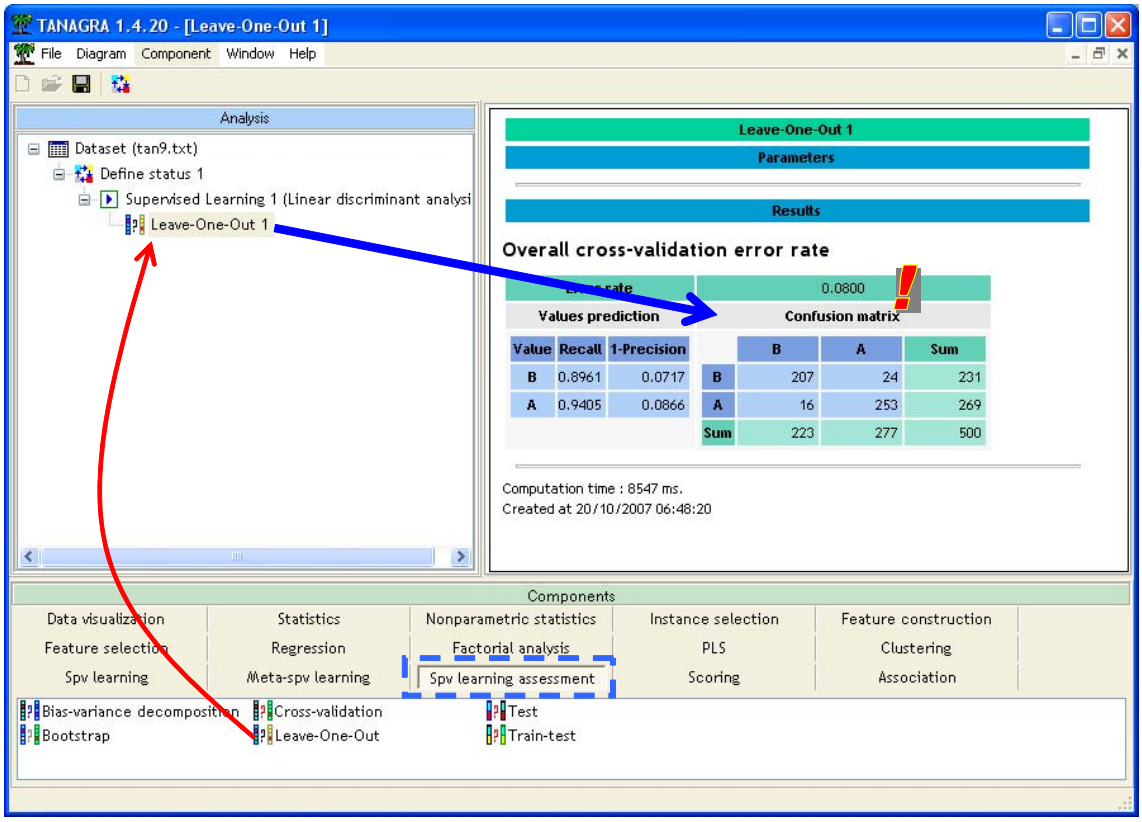
Dans ce qui suit, nous montrons comment utiliser les composants d'évaluation de l'erreur en ré échantillonnage dans TANAGRA. Nous vérifierons également si l'erreur calculée dans ce cas se rapproche de la « vraie » erreur (avec les réserves émises plus haut) 8.39% estimée sur le fichier de 42500 observations.

Erreur « leave one out »

L'idée de l'erreur *leave one out* est de réitérer l'opération suivante pour chaque observation i : construire le modèle sur les (n-1) observations en excluant l'observation numéro i, l'appliquer sur cet individu, et vérifier qu'il est bien classé ($d_i = 1$ s'il est mal classé, $d_i = 0$ sinon). L'erreur e-lvo s'écrit

$$e-lvo = \frac{1}{n}\sum_i d_i$$

Dans TANAGRA, il s'agit de glisser le composant LEAVE-ONE-OUT (onglet SPV LEARNING ASSESSMENT) à la suite de l'analyse discriminante et d'activer le menu VIEW pour obtenir les résultats. TANAGRA effectue donc 500 fois l'opération « apprentissage sur 499 observations et test sur 1 observation ». Ca semble considérable mais compte tenu des caractéristiques de notre étude (taille du fichier + méthode), le temps d'attente est raisonnable (#8 secondes).



Le taux d'erreur *e-lvo* est égal à 8%. L'estimation semble satisfaisante. De manière générale, la technique leave one out permet d'estimer correctement l'erreur, avec un biais faible, mais une variance plus élevée que les autres techniques. Elle est totalement inappropriée en revanche dans les cas de fort sur apprentissage (par exemple, ratio nombre de descripteurs – nombre d'observations très élevé associé à la méthode du plus proche voisin).

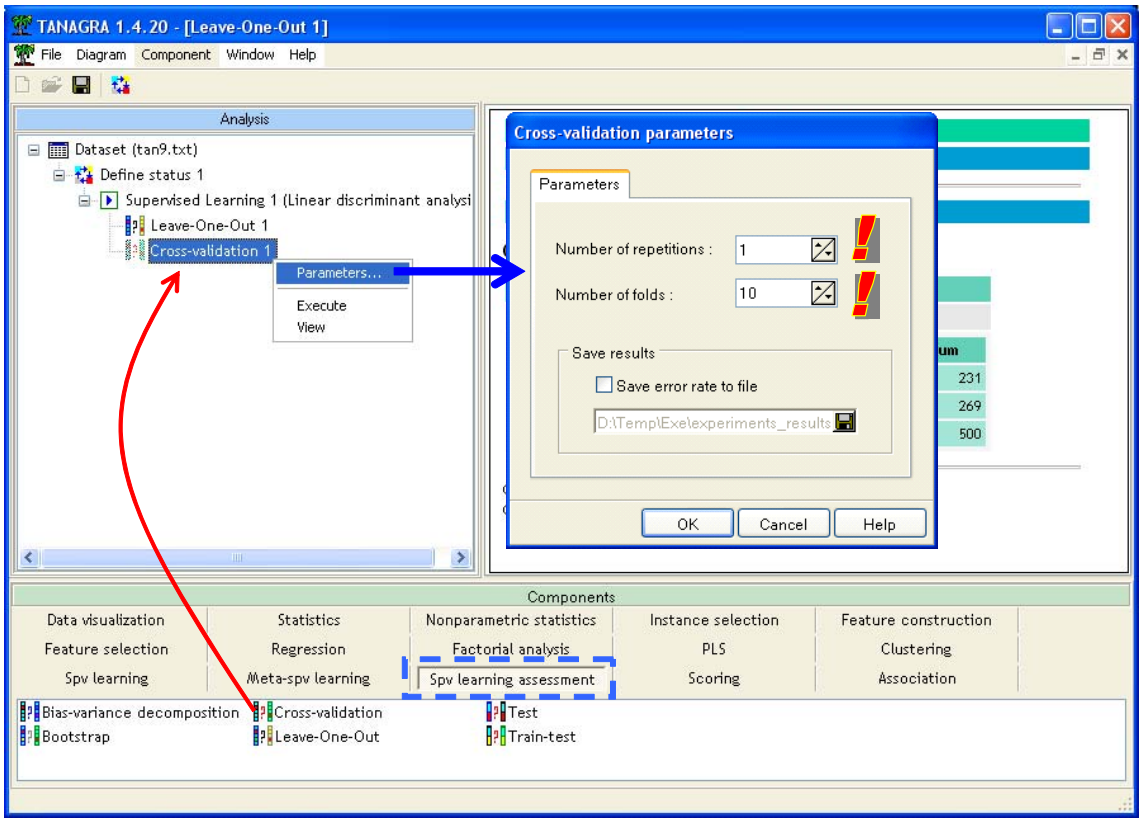
Erreur « validation croisée »

La validation croisée est une sorte de simplification du leave one out où l'on subdivise les données en K blocs. On répète alors K fois le processus suivant : apprentissage sur (K-1) blocs, test sur le K-ième bloc, nous mesurons l'erreur en test e_k. L'erreur e-cv s'écrit

$$e-cv = \frac{1}{K}\sum_k e_k$$

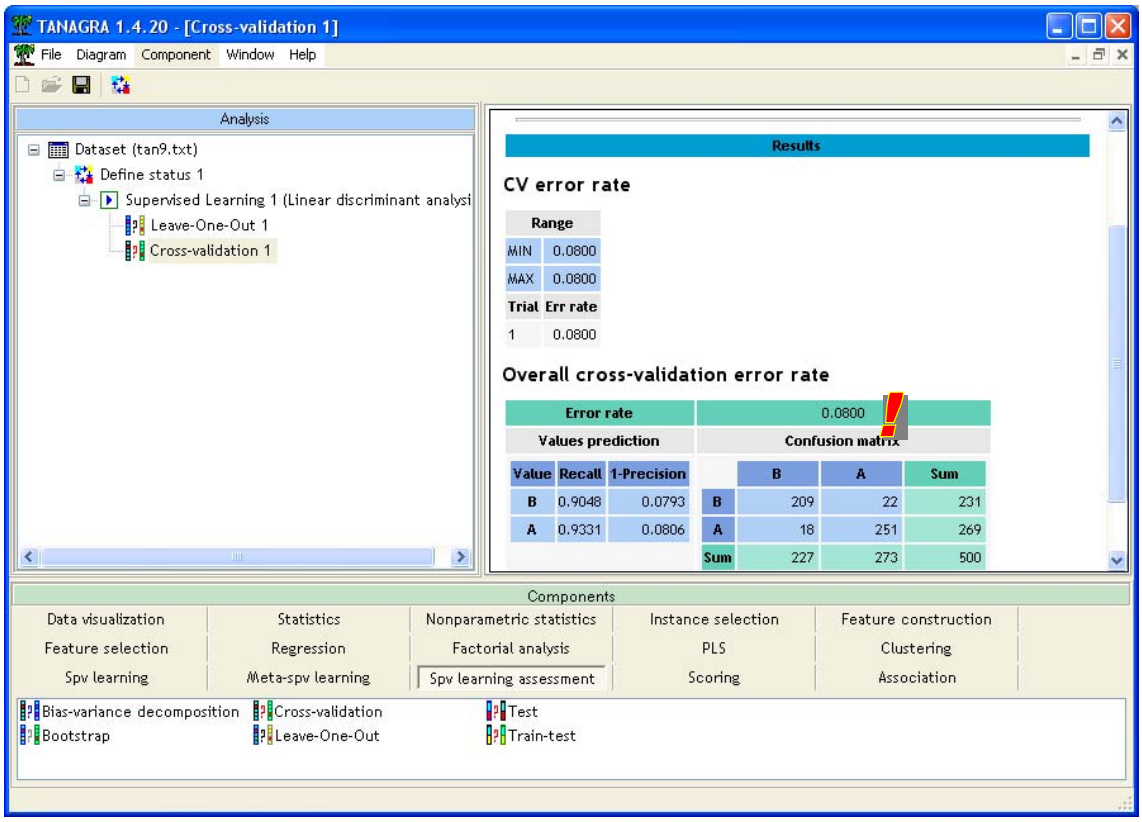
Son principal atout est de réduire la quantité de calcul par rapport au leave one out, sans détériorer la précision de l'estimation. Elle peut même s'avérer meilleure dans certaines situations. Les études empiriques montrent que K=10 semble un bon compromis.

Nous insérons le composant CROSS-VALIDATION à la suite de l'analyse discriminante. Nous activons le menu PARAMETERS pour définir les paramètres de la méthode.



Nous spécifions bien K=10 (NUMBER OF FOLDS), nous notons qu'il est possible de ré itérer plusieurs fois le processus (NUMBER OF REPETITIONS), cela permet d'apprécier la variabilité des résultats, au prix certes d'une quantité de calculs plus élevée. Dans notre cas, nous laissons NUMBER OF REPETITIONS = 1.

Après avoir validé, et activer menu VIEW, nous obtenons les résultats suivants :



Le taux d'erreur e-cv est 8%, identique à e-lvo. C'est un hasard, nous observons d'ailleurs que la matrice de confusion est un peu différente. Il reste néanmoins que les taux d'erreur mesurés en validation croisée et par leave one out sont souvent assez proches.

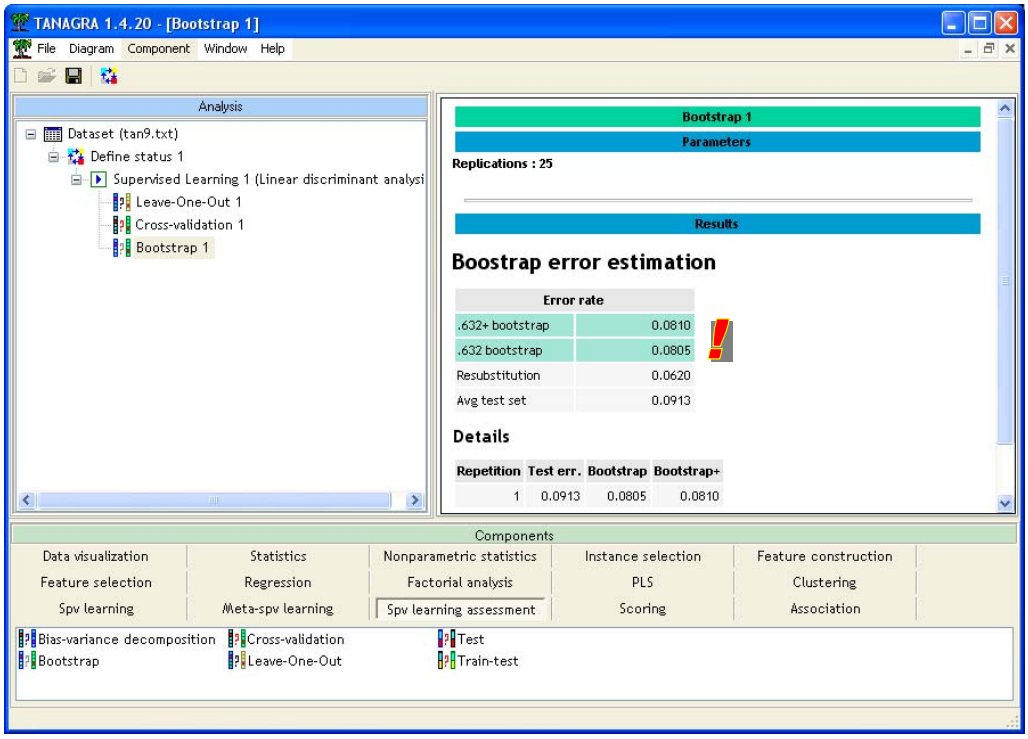
Erreur « bootstrap »

Contrairement aux deux techniques précédentes, le bootstrap ne cherche pas à estimer directement l'erreur, mais plutôt le biais d'optimisme. La présentation est un peu complexe, pour avoir plus de détails, mieux vaut consulter le document en ligne http://eric.univ-lyon2.fr/~ricco/cours/slides/resampling_evaluation.pdf

Deux estimateurs bootstrap sont disponibles dans TANAGRA, le standard 0.632 bootstrap et l'indicateur modifié 0.632 bootstrap+, censé tenir compte des spécificités de la technique d'apprentissage. Dans certaines situations, la correction peut être importante.

Le seul paramétrage possible est le nombre de réplications, dans la majorité des cas, 25 répétitions suffisent largement pour obtenir une estimation satisfaisante. En augmentant le nombre de réplications, nous réduisons la variance de l'estimateur. Le biais d'estimation lui reste stable. On convient généralement que le bootstrap est légèrement biaisé par rapport à la validation croisée, il présente en revanche une variance nettement plus faible à effort de calcul égal.

Nous insérons le composant BOOTSTRAP dans le diagramme. Nous obtenons les résultats suivants :



Le taux d'erreur *e-boot+* = 8.1%.

Nous résumons les résultats dans le tableau suivant

Erreur	**LDA**	**Ecart**
Resubstitution	6.2 %	- 2.19
« Vraie » (test)	**8.39 %**	**x**
LVO	8 %	- 0.39
CV (10)	8 %	- 0.39
Bootstrap+ (25)	8.1 %	- 0.29

Rappelons que ce que nous qualifions de « vraie » erreur est aussi une estimation dans ce didacticiel puisqu'il est calculé sur un échantillon. Nous pouvons néanmoins lui faire crédit car elle est calculée sur un ensemble de données à part extraite de la population, qui n'a en aucune manière participé à la construction des modèles, et qui possède un effectif suffisamment élevé.

Concernant les techniques d'estimation de l'erreur, celui qui se rapproche le plus du « vrai » taux d'erreur est l'estimateur « Bootstrap » avec un écart de -0.29 %. Mais les différences entre les techniques restent minimes, il ne faut surtout pas en tirer des conclusions hâtives. **La précision des estimateurs dépend des caractéristiques des données analysées et des méthodes d'apprentissage utilisées.**

Enfin, le fait que les techniques sous-estiment toutes la véritable erreur dans cet exemple est fortuit. Il arrive parfois que la technique de ré échantillonnage la surestime comme nous le verrons plus bas.

5. Conclusion

Dans ce didacticiel, nous avons mis en œuvre plusieurs techniques de ré échantillonnage pour évaluer l'erreur de prédiction sans disposer d'un échantillon test. Sur des données simulées, nous constatons que l'erreur ainsi évaluée est assez proche du véritable taux d'erreur que nous avons calculé par ailleurs sur un échantillon test de taille virtuellement infinie (si tant est que 42500 est suffisamment grand pour obtenir une précision satisfaisante).

Il reste que nous nous sommes placés dans un cadre plutôt favorable en combinant une méthode relativement stable (l'analyse discriminante) et un problème somme toute facile (500 observations pour 21 variables avec une distribution monomodale des classes, le test basé sur Lambda de Wilks confirme clairement cette appréciation). Nous avons voulu vérifier le comportement de ces techniques pour une méthode avec des caractéristiques assez différentes (arbre de décision C4.5), connue pour sa propension au sur apprentissage, nous obtenons les résultats suivants :

Erreur	**C4.5**	**Ecart**
Resubstitution	2.2%	- 11.34
« Vraie » (test)	**13.54%**	**x**
LVO	16%	+ 2.46
CV (10)	16.2%	+ 2.66
Bootstrap+ (25)	12.24%	- 1.3

Manifestement, C4.5 est nettement moins performant que la LDA sur ces données.

Concernant les estimations, l'erreur en resubstitution est clairement mauvaise, trop optimiste. Cela est du en grande partie aux caractéristiques de C4.5 qui a parfois tendance à trop coller aux données avec un arbre sur dimensionné.

De manière générale, la précision des estimations par ré échantillonnage est moindre. Nous constatons que dans certains cas (LVO, CV), l'erreur est surestimée.