Objectif

Montrer les nouvelles fonctionnalités du diagramme de traitements (**version 1.4.7**). Elles concernent essentiellement les possibilités de copier-coller de composants ou de sous-branches entières du diagramme, les paramètres sont également dupliqués.

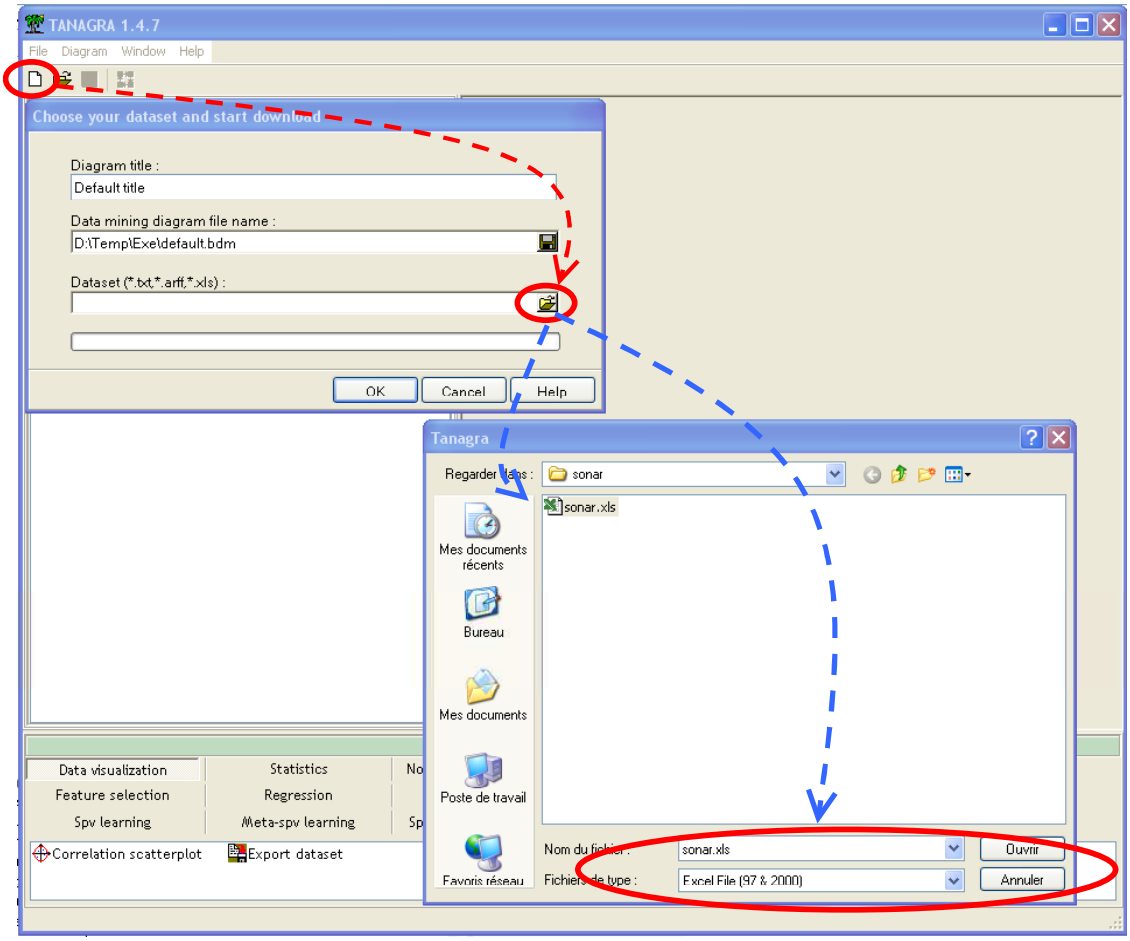
Fichier

Nous utilisons le fichier SONAR.XLS. Nous procédons à une comparaison des performances de plusieurs méthodes supervisées : la régression logistique, l'analyse discriminante, les K-plus proches voisins, les SVM et la régression PLS.

Manipuler le diagramme de traitements

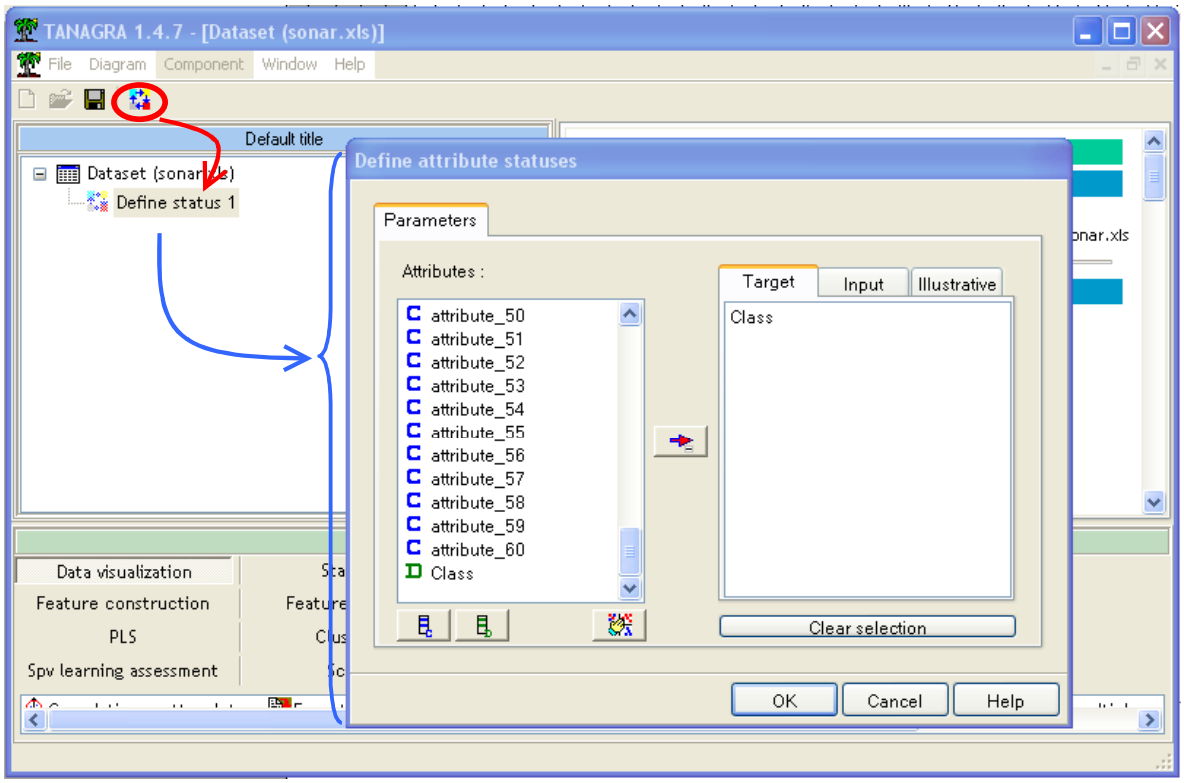
Importer les données

Première étape, créer un diagramme et importer les données : nous cliquons sur le menu FILE/NEW et sélectionnons le fichier SONAR.XLS.



Définir les variables de l'analyse

L'étape suivante consiste à définir l'attribut à prédire (TARGET : CLASS) et les descripteurs (INPUT : Tous les autres). Nous ajoutons, pour ce faire, le composant DEFINE STATUS en utilisant le raccourci dans la barre d'outils.



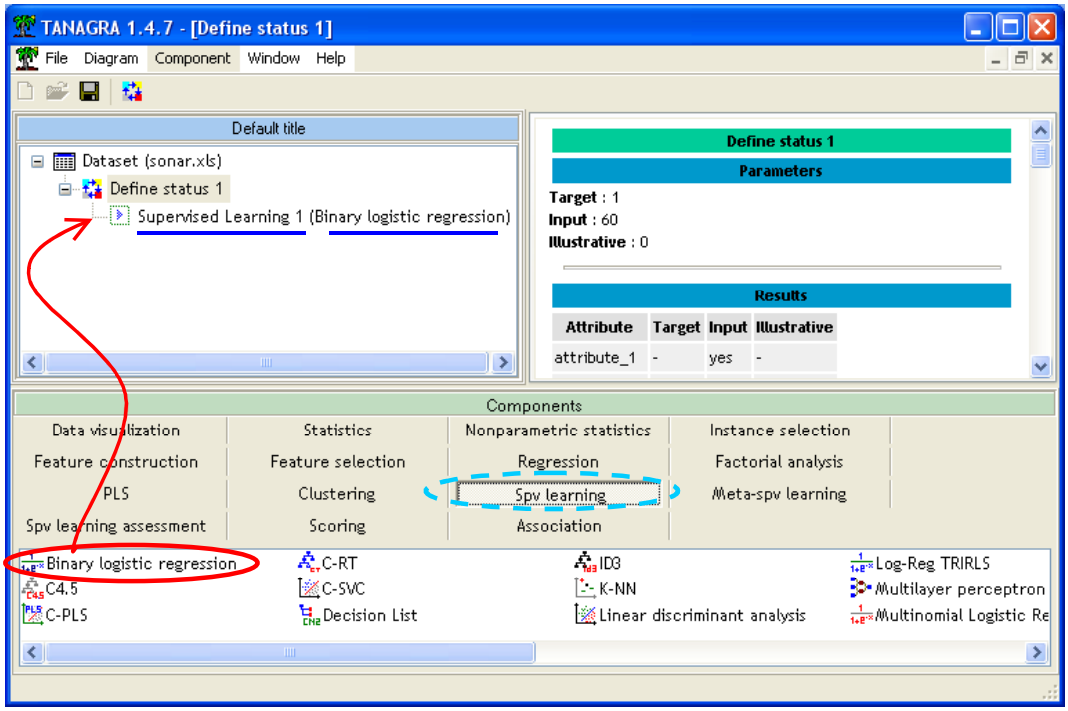
Placer la méthode d'apprentissage et son évaluation

Jusqu'à maintenant (version 1.0.0 → 1.4.6), l'adjonction d'une méthode supervisée dans le diagramme se faisait en deux étapes : d'abord placer le composant qui instancie la méthode (META SPV LEARNING), puis y insérer l'algorithme d'apprentissage (SPV LEARNING).

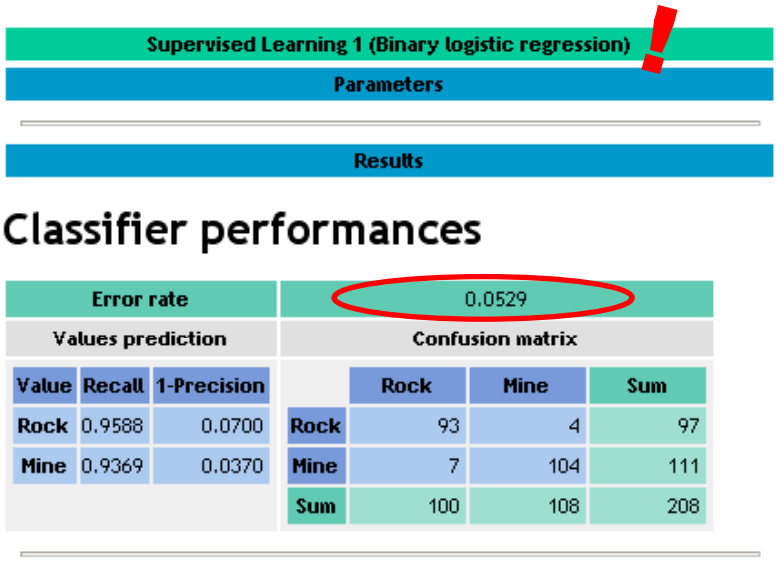
Dorénavant (version 1.4.7), dans le cadre de l'apprentissage simple, avec une seule instance de l'algorithme, il sera possible de placer directement le composant de la palette apprentissage supervisé dans le diagramme, TANAGRA effectue automatiquement le nécessaire pour que le composant META SPV soit correctement ajouté.

NOTE : Attention, si l'on veut mettre en oeuvre une procédure d'agrégation particulière, le « bagging » par exemple, l'insertion en deux temps est toujours requise : placer d'abord le meta-apprentissage avant d'y intégrer le composant représentant l'algorithme d'apprentissage.

Nous glissons donc le composant BINARY LOGISTIC REGRESSION (onglet SPV LEARNING) dans le diagramme, après avoir relâché la souris, nous observons qu'il a été complété correctement.

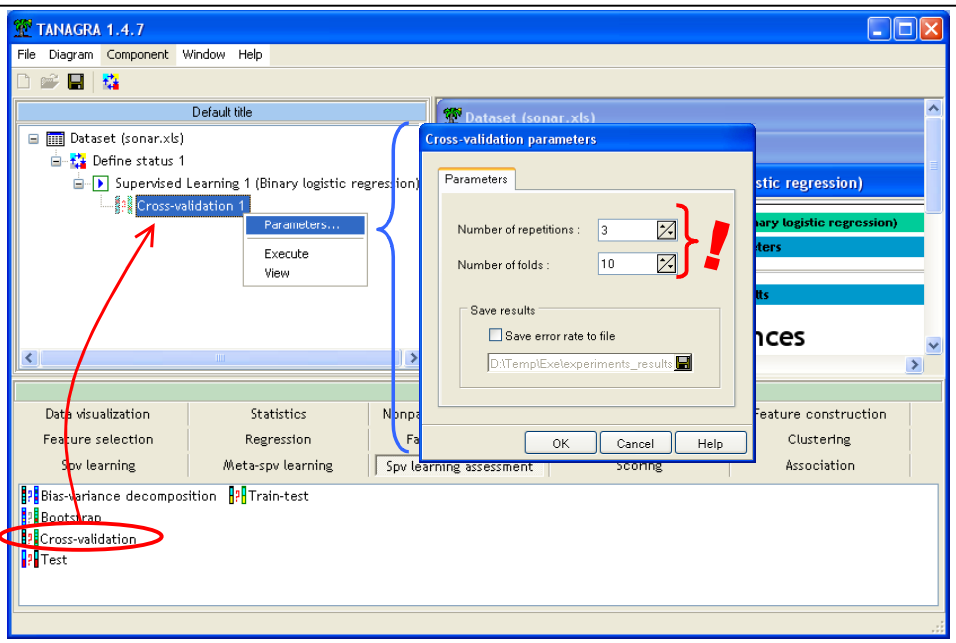


L'apprentissage donne un taux de mauvais classement de 5.29%.



Nous savons que ce taux d'erreur, estimé sur le fichier d'apprentissage, est biaisé. Pour obtenir une estimation plus crédible, nous voulons utiliser la validation croisée que nous retrouvons dans la palette SPV LEARNING ASSESSMENT. Nous modifions le paramétrage par défaut[1], nous le configurons de manière à ré-itérer 3 fois la « 10-fold cross validation ». Cette précision est importante, nous devrons utiliser le même protocole pour toutes les évaluations à venir dans notre didacticiel.

[1] Le paramétrage par défaut est de 5 fois une 2-fold validation croisée.



L'exécution indique un taux d'erreur de 30.5%, éloigné du résultat en resubstitution, notons que la procédure a bien respecté le paramétrage que nous avons défini (TRIALS = 3 ; FOLDS = 10).

Cross-validation 1

Parameters

Cross-validation parameters	
Folds	10
Trials	3

Results

CV error rate

Range	
MIN	0.3000
MAX	0.3100
Trial	**Err rate**
1	0.3000
2	0.3050
3	0.3100

Overall cross-validation error rate

Error rate	0.3050

Values prediction

Value	Recall	1-Precision
Rock	0.6403	0.3180
Mine	0.7422	0.2950

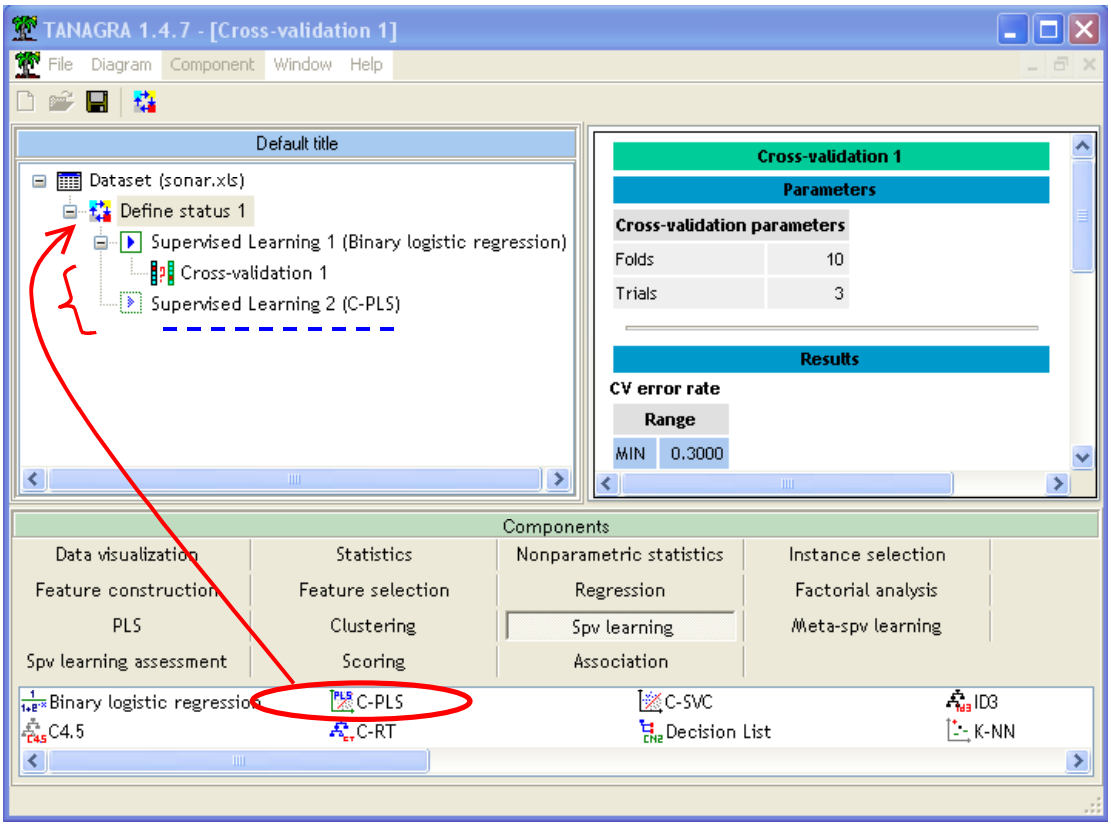
Confusion matrix

	Rock	Mine	Sum
Rock	178	100	278
Mine	83	239	322
Sum	261	339	600

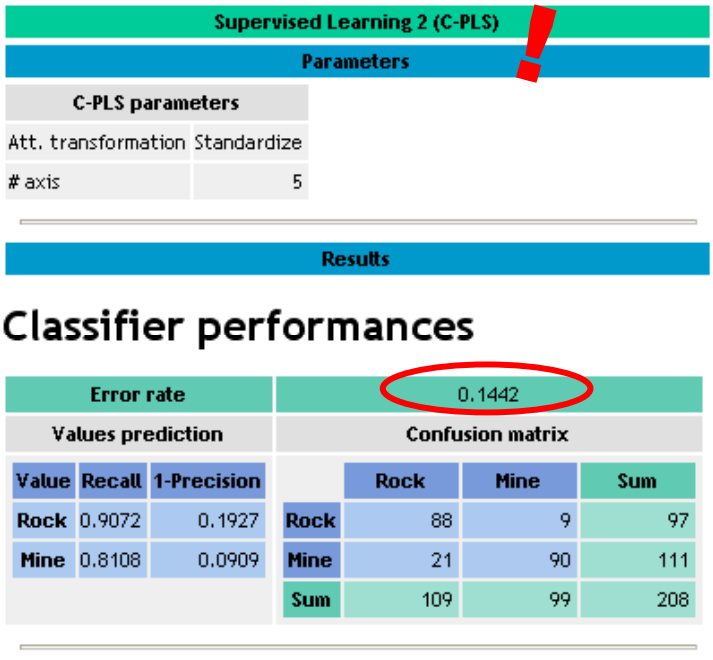
Ajouter une autre méthode d'apprentissage et son évaluation

Nous voulons ré-appliquer le même schéma pour la variante de la régression PLS dédiée à la discrimination (composant C-PLS de l'onglet SPV LEARNING).

Nous le glissons dans le diagramme, sur le composant DEFINE STATUS, il apparaît au même niveau que la régression logistique.



L'exécution annonce les performances suivantes en resubstitution.



Supervised Learning 2 (C-PLS)

Parameters

C-PLS parameters	
Att. transformation	Standardize
# axis	5

Results

Classifier performances

Error rate	0.1442

Values prediction

Value	Recall	1-Precision
Rock	0.9072	0.1927
Mine	0.8108	0.0909

Confusion matrix

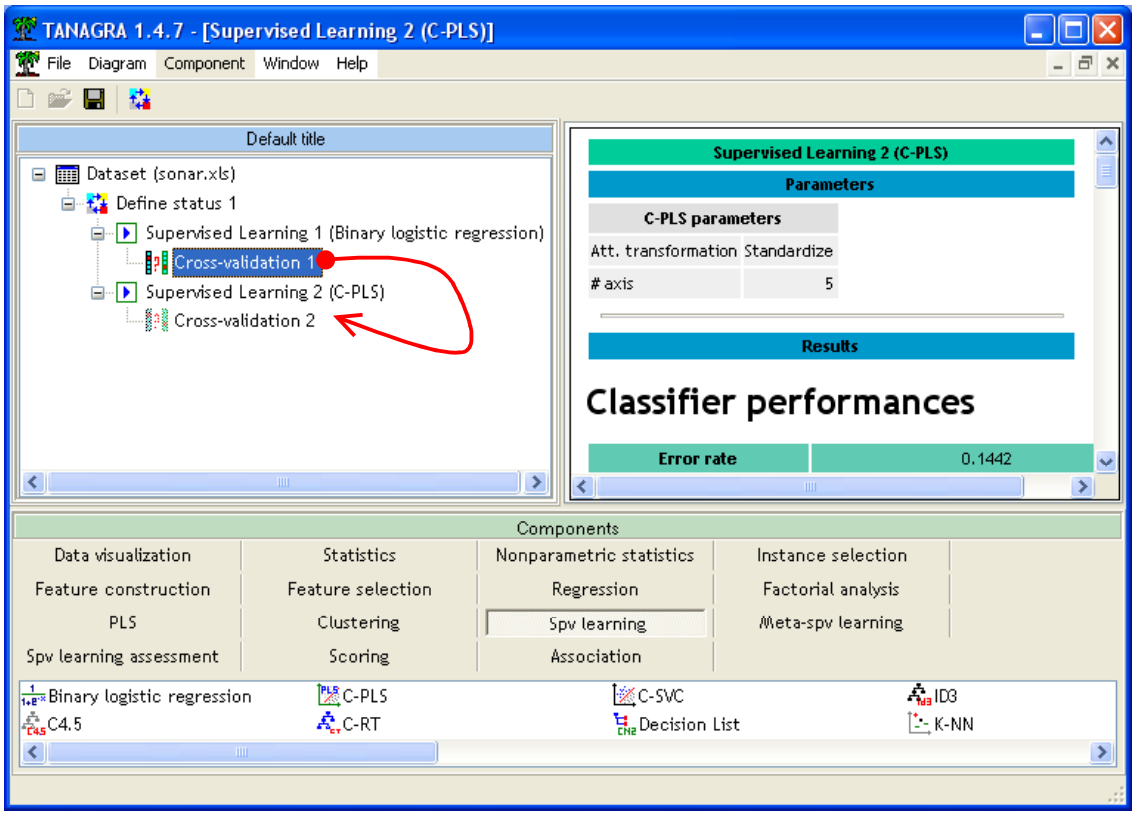
	Rock	Mine	Sum
Rock	88	9	97
Mine	21	90	111
Sum	109	99	208

De nouveau, nous devons placer la validation croisée, avec le même paramétrage que précédemment (3 TRIALS et 10 FOLDS).

Auparavant (*version 1.0.0 → 1.4.6*), nous devions reprendre le générateur de composant de la palette SPV ASSESSMENT, le placer dans le diagramme, le configurer en activant le menu PARAMETERS. Cela peut être très fastidieux, surtout si nous avons à répéter cette opération plusieurs fois dans notre étude.

Avec la version 1.4.7, nous pouvons copier le composant déjà dans le diagramme, en dupliquant également ses paramètres. Cela peut être très pratique lorsque nous avons un composant à placer à plusieurs endroits du diagramme. C'est le cas de la validation croisée dans notre exemple.

Pour dupliquer le composant *CROSS-VALIDATION 1*, il faut le sélectionner en cliquant dessus, puis par *glisser-déposer*, à l'aide de la souris, le positionner sur le composant *SPV LEARNING 1 (C-PLS)*. **L'opération doit être réalisée avec la souris, aucun raccourci clavier ou menu spécifique n'est prévu pour cette opération.**



Notons que le composant a été re-numéroté automatiquement (CROSS-VALIDATION 2).

L'exécution de la validation croisée indique un taux d'erreur de 25.67% ; point très important, le paramétrage (3 TRIALS ; 10 FOLDS) a bien été transmis au composant dupliqué.

Cross-validation 2

Parameters

Cross-validation parameters	
Folds	10
Trials	3

Results

CV error rate

Range	
MIN	0.2300
MAX	0.2900
Trial	**Err rate**
1	0.2900
2	0.2500
3	0.2300

Overall cross-validation error rate

Error rate	0.2567

Values prediction

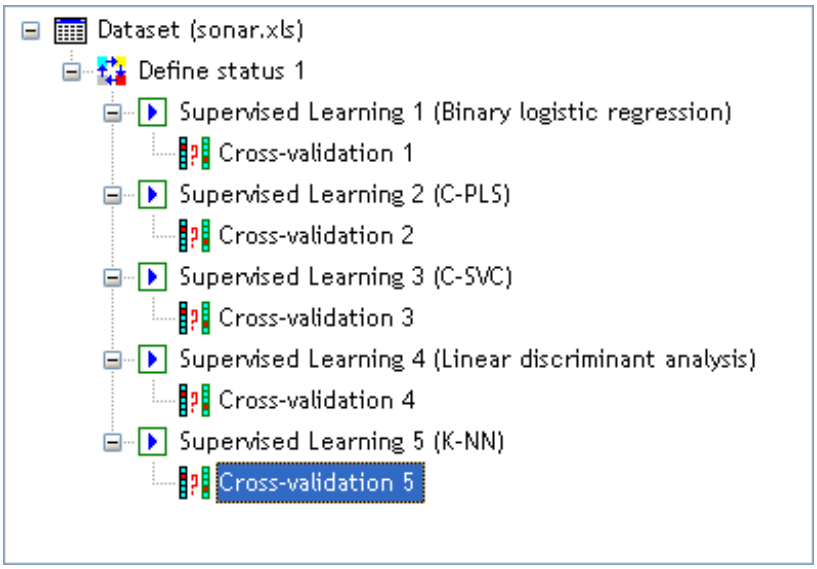
Value	Recall	1-Precision
Rock	0.7806	0.3000
Mine	0.7112	0.2103

Confusion matrix

	Rock	Mine	Sum
Rock	217	61	278
Mine	93	229	322
Sum	310	290	600

Evaluation de plusieurs méthodes

En réitérant ces manipulations, nous complétons notre étude de manière à obtenir le diagramme suivant.



Nous pouvons dès lors dégager un tableau de résultats recensant les taux d'erreurs pour chaque méthode.

Tableau 1 -- Taux d'erreur par méthode

Méthode	Erreur resubstitution (%)	Erreur validation croisée (%)
Régression logistique	5.29	30.5
Régression PLS (C-PLS)	14.42	25.67
SVM Linéaire (C-SVC)	12.02	25.33
Analyse Discriminante	10.10	23.50
K-Plus Proches Voisins (K-NN)	9.62	**14.17**

Manifestement, le bon modèle de prédiction dans ce problème est non-linéaire. Malgré l'éparpillement des points, la dimensionalité (60 descripteurs) est élevée relativement au nombre d'observations (208 exemples), l'algorithme des plus proches voisins est de très loin la plus performante.

Remarque : *Etonnamment, nous obtenons des résultats très différents des évaluations de type « apprentissage – test » où nous réservions 108 observations pour l'apprentissage, 100 observations pour le test (http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_Compare_Algorithms_On_Predefined_Test_Set.pdf). Dans ce cas, rappelons-le, l'analyse discriminante présente un taux d'erreur de 36%, les K-NN de 22%. Qu'est-ce qui peut expliquer une telle différence dans les résultats ?*

En fractionnant les données pour ne réserver que 108 observations pour l'apprentissage, nous pénalisions involontairement les méthodes sensibles à la dimensionalité, tout du moins, du ratio dimension de l'espace de représentation – nombre d'observations. C'est le cas de l'analyse discriminante et des plus proches voisins, les estimations globales ou locales des probabilités sont très problématiques, se traduisant ainsi par une qualité d'apprentissage médiocre. Il en est autrement en ce qui concerne les SVM qui sont très stables, nous noterons que le taux d'erreur mesuré dans les deux cas sont du même ordre.

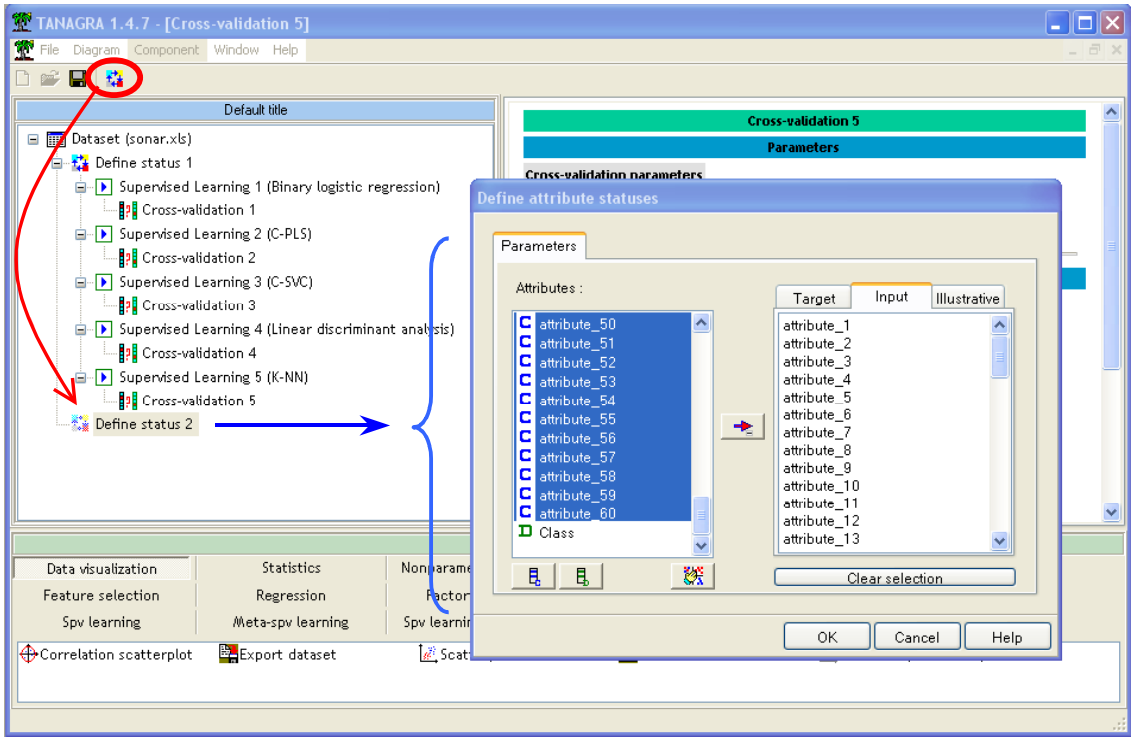
Moralité, les méthodes d'évaluation par (ré)-échantillonnage donnent parfois des résultats faussés. Le biais dépend à la fois des caractéristiques des données et de la méthode à évaluer. Ici, vu la faiblesse des effectifs, il semble plus indiqué d'utiliser la validation croisée pour évaluer les méthodes.

Réduction de la dimensionalité, duplication de portions du diagramme

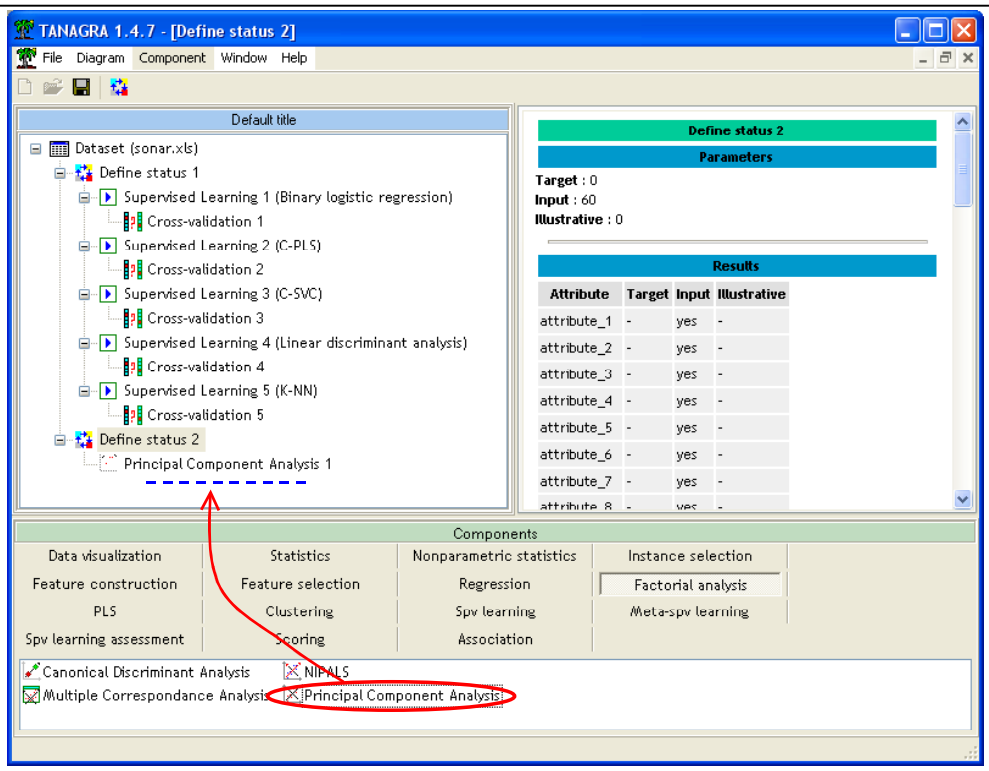
Face à ces résultats, nous pouvons nous demander s'il est judicieux de procéder à une réduction de la dimensionalité. C'est une très bonne idée si nous arrivons à préserver l'information « utile » pour le classement.

Nous complétons notre étude en utilisant la stratégie suivante : (1) nous procédons à une analyse en composantes principales sur les descripteurs, (2) puis nous proposons les axes factoriels comme INPUT des méthodes ci-dessus.

Repositionnons-nous sur la racine, la source de données du diagramme, à l'aide du raccourci de la barre d'outil, nous insérons un nouveau composant DEFINE STATUS. Nous plaçons en INPUT toutes les variables continues.



Nous plaçons le composant PRINCIPAL COMPONENT ANALYSIS sur DEFINE STATUS 2 du diagramme. Nous gardons le paramétrage par défaut, la méthode produira les 10 premiers axes factoriels.



L'exécution (menu VIEW) indique que les dix premiers axes traduisent 72% de l'information disponible[2].

Principal Component Analysis 1

Parameters

Generated axis : 10

Results

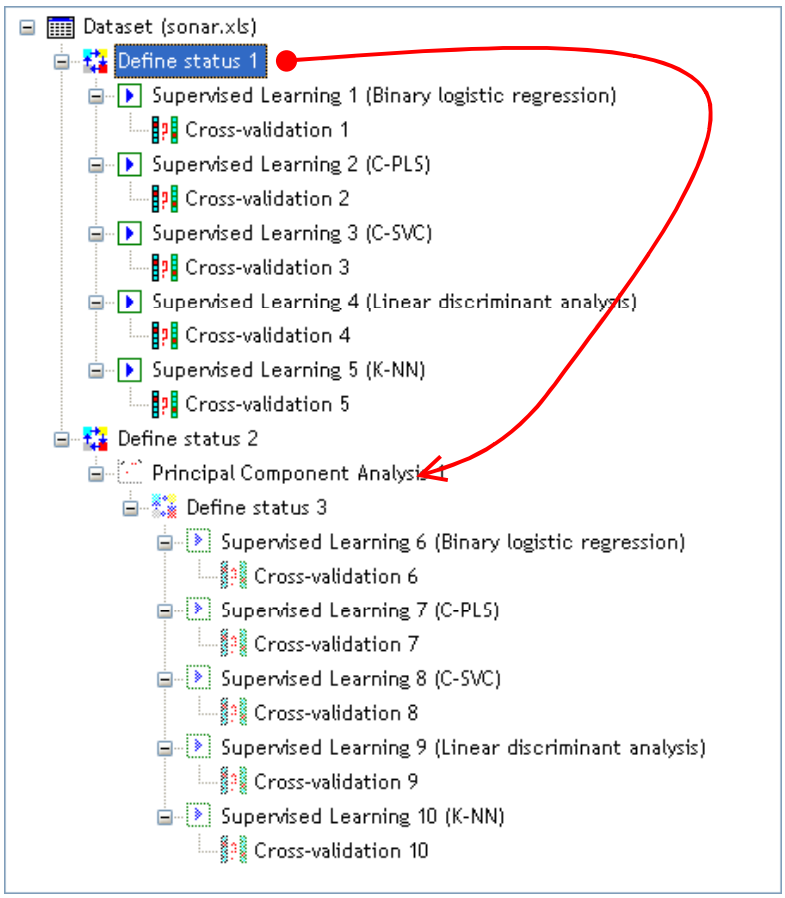
Eigen values

Axis	Eigen value	% explained	Histogram	% cumulated
1	11.907436	19.85%		19.85%
2	11.184611	18.64%		38.49%
3	5.068340	8.45%		46.93%
4	3.295124	5.49%		52.43%
5	2.965207	4.94%		57.37%
6	2.409907	4.02%		61.38%
7	2.003368	3.34%		64.72%
8	1.726094	2.88%		67.60%
9	1.497870	2.50%		70.10%
10	1.476919	2.46%		72.56%
11	1.251381	2.09%		74.64%
12	1.199677	2.00%		76.64%

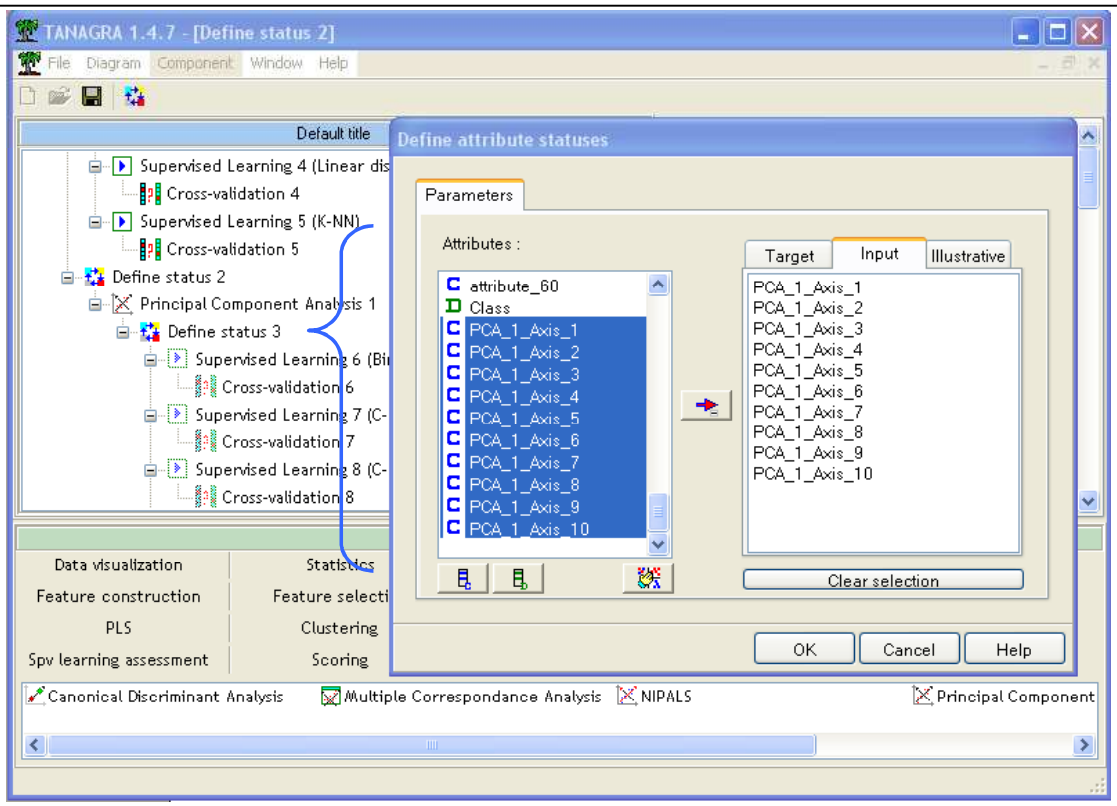
[2] Notre objectif étant de montrer le fonctionnement du logiciel, nous ne cherchons pas à optimiser le paramètre « nombre d'axes à retenir » dans ce didacticiel.

Il reste maintenant à replacer toute la séquence de composants ci-dessus : un DEFINE STATUS pour définir la cible (CLASS) et les INPUTS (les 10 axes factoriels) ; puis tous les couples « apprentissage – évaluation en validation croisée », en veillant à respecter le paramétrage qui a été défini pour l'expérimentation.

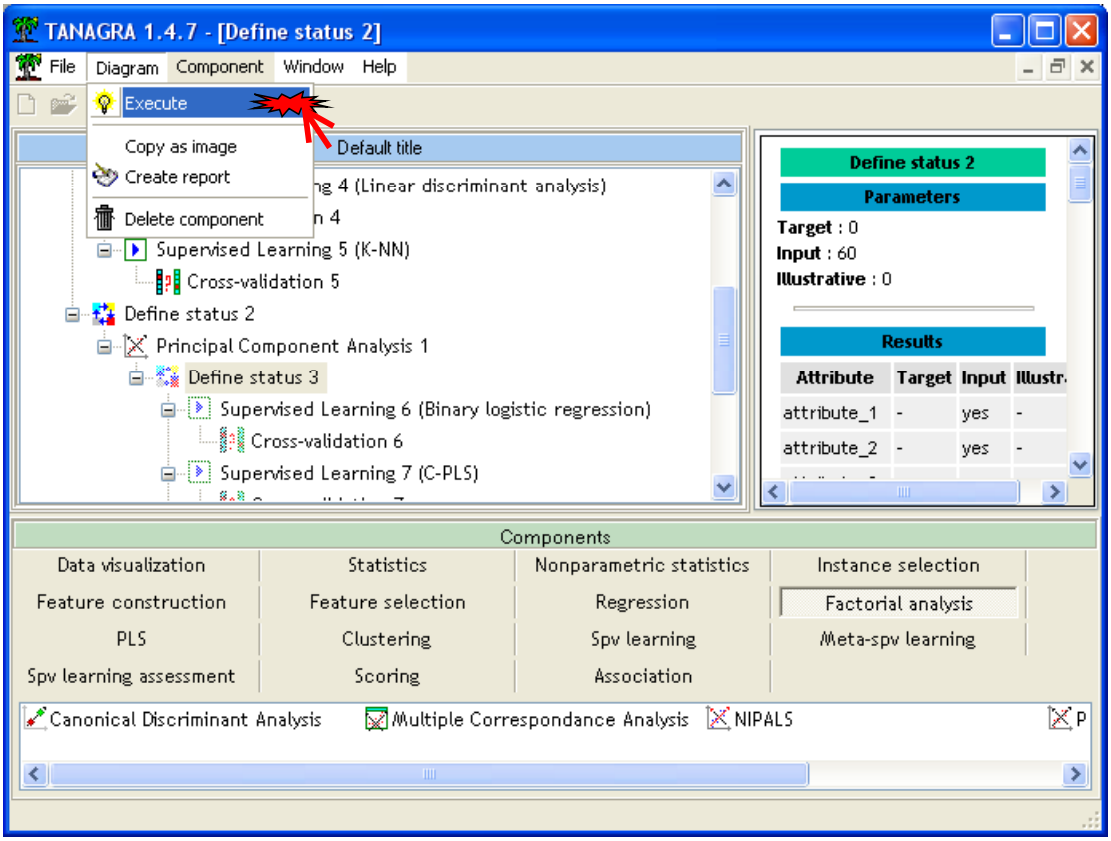
Ici également, reproduire la série de traitements en plaçant les composants un à un est éminemment rébarbatif. Nous pouvons maintenant décalquer la séquence, les branches du diagramme, en sélectionnant le composant DEFINE STATUS 1, puis, par glisser-déposer, le copier sur le composant PRINCIPAL COMPONENT ANALYSIS 1. Nous retrouvons alors les méthodes d'apprentissages et leurs évaluations respectives à l'aide la validation croisée. Tous les composants ont été re-numérotés correctement pour éviter les doublons.



Avant de lancer l'exécution, il faut bien entendu configurer (menu PARAMETERS) le composant DEFINE STATUS 3 de manière à spécifier comme INPUT les 10 axes factoriels, et comme TARGET l'attribut CLASS.



Reste à lancer l'exécution de l'ensemble des traitements supervisés, nous pouvons actionner l'option DIAGRAM / EXECUTE dans le menu principal de l'application.



Le Tableau 2 résume les performances.

Tableau 2 -- Taux d'erreur par méthode, utilisation des axes factoriels

Méthode	Erreur resubstitution (%)	Erreur validation croisée (%)
Régression logistique	16.35	20.17
Régression PLS (C-PLS)	16.83	21.67
SVM Linéaire (C-SVC)	18.27	20.83
Analyse Discriminante	15.87	21.50
K-Plus Proches Voisins (K-NN)	7.69	**16.33**

Dans cet exemple, toutes les méthodes linéaires bénéficient de la régularisation à l'aide des axes factoriels. La dégradation de la méthode des plus proches voisins, qui reste dans l'absolu la meilleure approche, laisse à penser qu'en jouant sur le nombre d'axes, nous pourrons influer sur les résultats.

Nous constatons surtout que la possibilité d'effectuer des copier-coller dans le diagramme de traitements nous facilite grandement la vie.