1 Objectif

Théorie et pratique des arbres de classification.

La classification automatique ou analyse typologique (« clustering » en anglais) vise les regrouper les individus en paquets homogènes. Les individus qui ont des caractéristiques similaires (proches) sont réunis dans un même groupe (cluster, classe) ; les individus présentant des caractéristiques dissemblables (éloignées) sont associés à des groupes différents.

Certaines difficultés sont récurrentes en classification automatique : la détermination du nombre de groupes, leur interprétation, etc. Deux d'entre elles sont particulièrement pénalisantes dans un contexte industriel. Le déploiement du modèle – on parle également d'industrialisation – n'est pas facile à mettre en œuvre. Il s'agit d'implanter dans le système d'information un programme permettant d'associer automatiquement un individu supplémentaire – n'ayant pas participé à la constitution de la partition – à un des groupes. Si ce processus repose sur des calculs complexes, lents et difficiles à mettre à jour. Les performances et la maintenance poseront problèmes sur le long terme.

La capacité à traiter de grandes bases est une seconde difficulté qui sera de plus en plus prégnante dans un contexte d'une augmentation constante de la volumétrie des données. Les méthodes de classification automatique les plus répandues ont peine à appréhender ces configurations, soit parce qu'elles requièrent un nombre de passage important sur la base de données, soit parce qu'elles nécessitent le calcul d'une matrice de distance entre individus, impossible à faire tenir en mémoire des ordinateurs sur de gros volumes.

Nous présentons dans ce tutoriel les arbres de classification. Ils apportent des solutions aux deux écueils précités. La démarche s'intègre dans un cadre cohérent par rapport aux arbres de décision et régression, bien connus en data mining. La différence réside dans la mise en place d'un critère multivarié pour quantifier la pertinence des segmentations durant la construction de l'arbre. Nous avions déjà présenté succinctement la méthode dans un précédent didacticiel (avril 2008)[1]. Mais nous nous étions focalisés sur les aspects opérationnels (manipulations dans Tanagra et lecture des résultats). Dans ce nouveau document, nous nous attardons sur les

[1] « Arbres de classification », avril 2008 ; http://tutoriels-data-mining.blogspot.fr/2008/04/arbres-de-classification.html

fondements théoriques de l'approche. Nous montrons que nous pouvons appréhender de manière indifférenciée les bases comportant des variables actives quantitatives ou qualitatives, ou un mix des deux. Par la suite, nous détaillons la mise en œuvre de la méthode à l'aide de plusieurs logiciels dont SPAD qui, à ma connaissance, est le seul à proposer une interface graphique interactive pour la construction des arbres de classification.

Ce texte reprend certains passages d'un article que j'avais naguère écrit sur les arbres de classification et leur intérêt dans des domaines où l'interprétation des résultats est au moins aussi importante que la performance brute (Rakotomalala et Le Nouvel, 2007).

2 Les arbres de classification – Fondements théoriques

2.1 Les arbres de classification

L'analogie avec les arbres de décision (Rakotomalala, 2005) joue pleinement pour comprendre la construction des arbres de classification. L'approche s'appuie sur un algorithme récursif de segmentation. Chaque subdivision vise à produire un partitionnement maximisant un critère de qualité en rapport avec la typologie multivariée. Dans cette optique, les arbres de classification constituent une extension de ces approches (arbres de décision ou arbres de régression) où, au critère de pureté et de variance, est substitué un critère d'homogénéité calculé sur l'ensemble des variables actives. Les feuilles de l'arbre représentent les classes produites par la typologie. L'objectif de l'apprentissage est de produire des groupes où l'on chercherait à minimiser par exemple l'inertie intra-classes. Chaque chemin partant de la racine à une feuille correspond à une règle logique désignant un groupe.

Replacé parmi les techniques de classification, les arbres de classification correspondent à une méthode descendante, divisive et monothétique. « Divisive » parce que le point de départ est la partition grossière rassemblant toutes les observations. La démarche consiste à fractionner itérativement les individus de manière à constituer des groupes homogènes. « Monothétique » parce que la subdivision est réalisée à partir des valeurs d'une variable, même si par ailleurs le degré d'homogénéité des groupes est calculé sur l'ensemble des variables actives. Les travaux de Chavent (1998) et Blockeel (1998) constituent des repères importants dans le domaine. J'y ai également un peu contribué.

2.2 Un exemple introductif

Prenons un exemple simple pour préciser les idées. Nous disposons de la description de n = 28 véhicules à l'aide de p = 5 variables (prix, cylindrée, puissance, poids, consommation).

Modele de vehicule	Prix	Cylindree	Puissance	Poids	Conso
Citroen ZX Volcane	28750	1998	89	1140	8.8
Daihatsu Cuore	11600	846	32	650	5.7
Fiat Panda Mambo L	10450	899	29	730	6.1
Fiat Tempra 1.6 Liberty	22600	1580	65	1080	9.3
Ford Fiesta 1.2 Zetec	19740	1242	55	940	6.6
Fort Escort 1.4i PT	20300	1390	54	1110	8.6
Honda Civic Joker 1.4	19900	1396	66	1140	7.7
Hyundai Sonata 3000	38990	2972	107	1400	11.7
Lancia K 3.0 LS	50800	2958	150	1550	11.9
Mazda Hachtback V	36200	2497	122	1330	10.8
Mitsubishi Galant	31990	1998	66	1300	7.6
Nissan Primera 2.0	26950	1997	92	1240	9.2
Opel Astra 1.6i 16V	25000	1597	74	1080	7.4
Opel Corsa 1.2i Eco	14825	1195	33	895	6.8
Opel Omega 2.5i V6	47700	2496	125	1670	11.3
Peugeot 306 XS 108	22350	1761	74	1100	9
Peugeot 806 2.0	36950	1998	89	1560	10.8
Renault Safrane 2.2. V	36600	2165	101	1500	11.7
Seat Alhambra 2.0	36400	1984	85	1635	11.6
Seat Ibiza 2.0 GTI	22500	1983	85	1075	9.5
Subaru Vivio 4WD	13730	658	32	740	6.8
Suzuki Swift 1.0 GLS	12490	993	39	790	5.8
Toyota Corolla	19490	1331	55	1010	7.1
Toyota Previa salon	50900	2438	97	1800	12.8
Volvo 850 2.5	39800	2435	106	1370	10.8
Volvo 960 Kombi aut	49300	2473	125	1570	12.7
VW Golt 2.0 GTI	31580	1984	85	1155	9.5
VW Polo 1.4 60	17140	1390	44	955	6.5

Figure 1 - Tableau de données "Autos"

Nous obtenons l'arbre suivant sous SPAD.

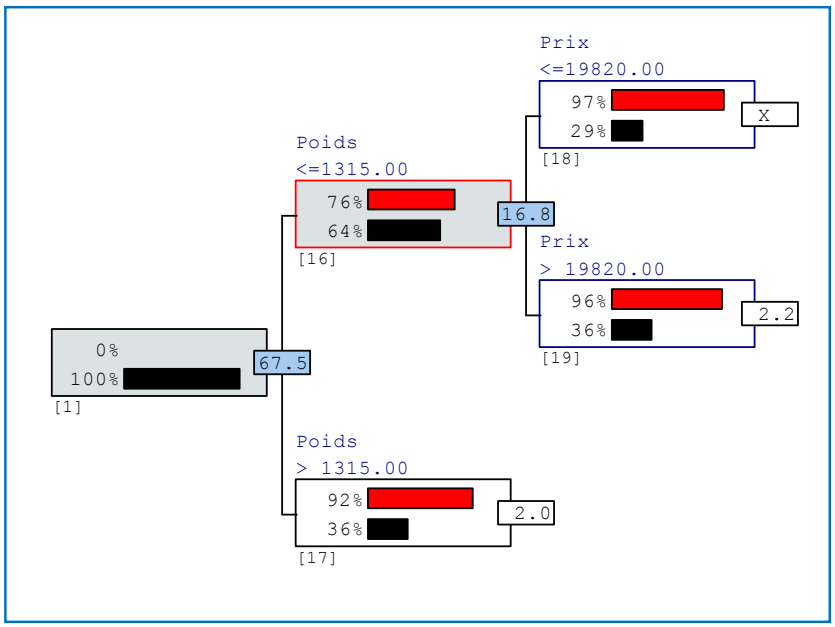


Figure 2 - Arbre de classification pour les données "Autos"

Description de l'arbre. Décrivons l'arbre sommairement :

- Le premier sommet est la racine de l'arbre (n° 1). Il recense la totalité des observations (100% en noir).
- « **Poids »** s'est imposé pour effectuer la première segmentation. C'est une variable quantitative, le seuil **1315** permet de définir 2 sous-groupes. Dans le sommet n° 17 (suivant la numérotation interne de SPAD), il contient 10 observations (soit 36% des n = 28 individus de la base). Le pourcentage en rouge (92%) indique le degré d'homogénéité du groupe. Nous y reviendrons plus loin lorsqu'il s'agira de décrire la pratique des arbres de classification à l'aide du logiciel SPAD. Dans l'autre sommet (n° 16), nous avons 18 observations (64 % de n = 28).
- La qualité de la segmentation est matérialisée par le chiffre « 67.5 » en fond bleu. Il s'agira pour nous de préciser son mode de calcul dans les sections qui viennent.
- Le sommet n° 16 est par la suite subdivisée en 2 sous-classes n° 18 et 19 avec respectivement 8 (29%) et 10 (36%) observations. Le « gain » d'information produit par la segmentation est de « 16.8 ».

Lecture des règles d'affectation. Nous avons donc une partition en 3 classes à la sortie. Les règles de désignation des groupes sont :

1. Si poids $\leq$ 1315 et prix $\leq$ 19820 Alors **groupe n°1**
2. Si poids $\leq$ 1315 et prix > 19820 Alors **groupe n°2**
3. Si poids > 1315 Alors **groupe n°3**

Les règles de désignation des groupes sont clairement lisibles, interprétables, et facile à implémenter dans les systèmes d'informations. Connaissant le poids et le prix d'un nouveau véhicule, nous saurons facilement à quel groupe l'associer.

Commentaires. D'autres remarques nous viennent à la lecture de ces résultats :

- Seules deux variables interviennent dans l'arbre. Alors que l'homogénéité des groupes est calculée sur l'ensemble des variables. Il y a donc un processus de sélection de variables dans la construction de la partition. C'est une problématique qui revient souvent dans les publications scientifiques.

- Néanmoins, pour caractériser les groupes, nous ne pouvons pas nous en tenir aux seules variables de segmentation apparentes. Tout comme dans un arbre de décision, certaines variables peuvent être pertinentes mais masquées par celles qui ont été sélectionnées pour définir les segmentations. La démarche usuelle d'interprétation des groupes via les méthodes factorielles ou les statistiques descriptives conditionnelles restent d'actualité ici.
- Nous disposons d'une hiérarchie de partitions imbriquées. A l'instar de ce qui peut se faire dans le cadre de la classification ascendante hiérarchique (CAH), nous pouvons définir et évaluer des scénarios de solutions cohérentes entres elles. Cette possibilité est particulièrement avantageuse en classification où, en pratique, la détermination du nombre idoine de classes reste un problème ouvert, et pour longtemps encore je pense.
- Il s'agit bien d'un partitionnement : un individu appartient (est affecté) à une et une seule classe.

Processus de construction de l'arbre. Nous retrouvons ici les mêmes problèmes à résoudre que lors de la construction d'un arbre de décision :

- Comment choisir la variable de segmentation sur un nœud ?
- Etant entendu que l'on choisit la variable la plus pertinente au sens d'un critère d'homogénéité : comment quantifier la dispersion d'un groupe ? comment quantifier le gain d'homogénéité lors d'une segmentation ?
- Lorsque nous souhaitons passer de la partition en K à (K + 1) classes, plusieurs feuilles sont candidates à la segmentation. Comment choisir celle que nous devons segmenter en priorité ? De fait, nous devons obtenir une partition hiérarchique indicée – un dendrogramme en d'autres termes – permettant de déterminer sans ambiguïté la séquence des subdivisions.
- On souhaite élaborer un arbre binaire. Les variables de segmentation peuvent être qualitatives ou quantitatives. Lorsqu'elle est qualitative binaire, la solution est évidente : chaque modalité induit une feuille. Mais lorsqu'elle est qualitative à L modalités (L > 2), comment procéder au regroupement de manière à obtenir une subdivision binaire ? Il faut que la stratégie soit cohérente avec l'objectif d'induire des groupes les plus purs possibles. De même, lorsque la variable prédictive est quantitative, nous devons produire un seuil de découpage optimal au sens de l'homogénéité des sous-groupes.

- Question récurrente en classification automatique, comment déterminer le nombre de groupe adéquat ? En d'autres termes, comment décider de l'arrêt de la construction de l'arbre ?

Dans les sections qui suivent, nous essaierons d'apporter des réponses à ces questions.

2.3 Mesure de qualité de segmentation

2.3.1 Indice d'homogénéité – L'inertie

L'inertie est une mesure naturelle de dispersion. Il s'agit d'une extension multivariée de la variance. Sur un ensemble C de n observations, nous le calculerons comme suit :

$$I(C) = \frac{1}{n}\sum_{i=1}^{n} d^2(i,G)$$

Où d() est la distance euclidienne, et G est le barycentre du nuage de points c.-à-d. le vecteur des moyennes calculées sur l'ensemble des p variables :

$$G = \left(\bar{x}_1;\ldots;\bar{x}_j;\ldots,\bar{x}_p\right)$$

Si les variables sont exprimées dans des unités différentes, certaines peuvent prendre le pas sur les autres dans la définition de la distance lorsqu'elles ont une variance plus grande. Le mieux est alors de les centrer et réduire systématiquement pour éviter cet inconvénient. Ces nouvelles variables s'écrivent :

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{\sigma_j}$$

Où σ_j est l'écart-type de la variable X_j.

Mécaniquement dans ce cas, le barycentre correspond à l'origine G = (0 ; … ; 0) et l'inertie totale des données est égal au nombre de variables

$$I = p$$

En pratique, nous préférons utiliser la somme des carrés des écarts à la moyenne S, avec

$$T = S(C) = n \times I = n \times p$$

Nous comprendrons pourquoi lorsque l'on explicitera le gain induit par une segmentation.

Pour notre fichier de données, n = 28 et p = 5, ainsi T = 28 x 5 = 140.

Après centrage et réduction avec les vecteurs « moyennes » et « écarts-type » suivants :

Moyenne	28393.75	1809.07	77.71	1196.96	9.08
Ecart-type	12163.37	612.42	31.68	303.42	2.19

Figure 3 - Moyennes et écarts-type des variables (Autos)

Nous obtenons le tableau des données centrées et réduites :

Modele de vehicule	Prix	Cylindree	Puissance	Poids	Conso
Citroen ZX Volcane	0.029	0.308	0.356	-0.188	-0.125
Daihatsu Cuore	-1.381	-1.573	-1.443	-1.803	-1.539
Fiat Panda Mambo L	-1.475	-1.486	-1.538	-1.539	-1.357
Fiat Tempra 1.6 Liberty	-0.476	-0.374	-0.401	-0.385	0.103
Ford Fiesta 1.2 Zetec	-0.711	-0.926	-0.717	-0.847	-1.129
Fort Escort 1.4i PT	-0.665	-0.684	-0.749	-0.287	-0.217
Honda Civic Joker 1.4	-0.698	-0.674	-0.370	-0.188	-0.627
Hyundai Sonata 3000	0.871	1.899	0.925	0.669	1.197
Lancia K 3.0 LS	1.842	1.876	2.282	1.164	1.288
Mazda Hachtback V	0.642	1.123	1.398	0.438	0.787
Mitsubishi Galant	0.296	0.308	-0.370	0.340	-0.673
Nissan Primera 2.0	-0.119	0.307	0.451	0.142	0.057
Opel Astra 1.6i 16V	-0.279	-0.346	-0.117	-0.385	-0.764
Opel Corsa 1.2i Eco	-1.116	-1.003	-1.412	-0.995	-1.038
Opel Omega 2.5i V6	1.587	1.122	1.493	1.559	1.015
Peugeot 306 XS 108	-0.497	-0.078	-0.117	-0.320	-0.034
Peugeot 806 2.0	0.703	0.308	0.356	1.196	0.787
Renault Safrane 2.2. V	0.675	0.581	0.735	0.999	1.197
Seat Alhambra 2.0	0.658	0.286	0.230	1.444	1.152
Seat Ibiza 2.0 GTI	-0.485	0.284	0.230	-0.402	0.194
Subaru Vivio 4WD	-1.206	-1.880	-1.443	-1.506	-1.038
Suzuki Swift 1.0 GLS	-1.308	-1.333	-1.222	-1.341	-1.494
Toyota Corolla	-0.732	-0.781	-0.717	-0.616	-0.901
Toyota Previa salon	1.850	1.027	0.609	1.987	1.699
Volvo 850 2.5	0.938	1.022	0.893	0.570	0.787
Volvo 960 Kombi aut	1.719	1.084	1.493	1.229	1.653
VW Golt 2.0 GTI	0.262	0.286	0.230	-0.138	0.194
VW Polo 1.4 60	-0.925	-0.684	-1.064	-0.797	-1.174

Figure 4 - Tableau des données centrées et réduites (Autos)

2.3.2 Régularisation – Classification sur facteurs

Classification sur facteurs. Le schéma ci-dessus convient lorsque les variables actives sont quantitatives. Il faudrait s'appuyer sur un autre type de distance si elles sont qualitatives, et un autre encore si nous avons un mix de variables quantitatives et qualitatives. Pour ne pas multiplier la gestion de cas particuliers, la classification sur facteurs constitue une solution élégante (Lebart et al., 2000 ; page 185 et suivantes). Ainsi, lorsque les variables sont toutes quantitatives, nous réalisons une ACP (analyse en composantes principales), puis nous procédons à la classification en calculant les inerties sur les facteurs ; lorsqu'elles sont toutes qualitatives, nous passons par une ACM (analyse des correspondances multiples) ; et lorsque nous avons un mix de variables, nous utilisons l'AFDM (analyse factorielle des données mixtes).

Axes factoriels deux à deux orthogonaux. Nous pouvons ainsi traiter dans un cadre unifié les différentes configurations. De plus, les facteurs étant deux à deux orthogonaux, l'utilisation de la distance euclidienne est parfaitement justifiée dans ce contexte.

Nettoyage (lissage) des données. Enfin, passer par les axes factoriels est totalement équivalent à procéder à partir des variables originelles si nous utilisons tous les facteurs (ex. avec une ACP sur p variables, nous obtenons p facteurs à la sortie). L'intérêt de cette étape préliminaire est que nous pouvons choisir un sous-ensemble de facteurs seulement pour réaliser la classification. Nous réalisons ainsi une sorte de nettoyage des données où le bruit associé aux données d'apprentissage, portées par les derniers facteurs à très faible variance, est éliminé. La construction de la partition se concentre sur les premiers axes factoriels, porteurs d'informations pertinentes, transposables dans la population.

Modele de vehicule	F1	F2	F3	F4	F5
Citroen ZX Volcane	0.170	0.489	0.074	-0.006	-0.017
Daihatsu Cuore	-3.458	0.172	-0.012	-0.171	0.250
Fiat Panda Mambo L	-3.306	-0.036	-0.113	0.009	0.112
Fiat Tempra 1.6 Liberty	-0.687	-0.128	-0.444	0.030	0.036
Ford Fiesta 1.2 Zetec	-1.936	0.093	0.291	-0.170	0.006
Fort Escort 1.4i PT	-1.165	-0.411	-0.248	0.074	-0.089
Honda Civic Joker 1.4	-1.147	-0.135	0.105	-0.102	-0.390
Hyundai Sonata 3000	2.487	0.550	-0.267	0.706	0.207
Lancia K 3.0 LS	3.781	0.828	0.230	-0.303	0.071
Mazda Hachtback V	1.961	0.732	-0.165	-0.104	-0.144
Mitsubishi Galant	-0.043	-0.117	0.828	0.425	0.073
Nissan Primera 2.0	0.372	0.312	-0.039	0.025	-0.312
Opel Astra 1.6i 16V	-0.846	0.258	0.363	-0.147	-0.088
Opel Corsa 1.2i Eco	-2.487	-0.206	-0.010	0.265	0.098
Opel Omega 2.5i V6	3.030	-0.095	0.429	-0.277	-0.113
Peugeot 306 XS 108	-0.470	0.172	-0.301	0.107	-0.132
Peugeot 806 2.0	1.498	-0.701	0.064	-0.024	-0.167
Renault Safrane 2.2. V	1.871	-0.353	-0.340	-0.069	-0.138
Seat Alhambra 2.0	1.683	-1.028	-0.179	0.071	-0.225
Seat Ibiza 2.0 GTI	-0.083	0.522	-0.491	0.158	-0.151
Subaru Vivio 4WD	-3.160	-0.320	-0.276	-0.423	0.268
Suzuki Swift 1.0 GLS	-2.995	0.107	0.146	-0.074	-0.022
Toyota Corolla	-1.676	-0.038	0.177	-0.027	-0.088
Toyota Previa salon	3.211	-1.120	0.122	0.128	0.298
Volvo 850 2.5	1.884	0.280	0.010	0.044	0.185
Volvo 960 Kombi aut	3.214	-0.113	-0.152	-0.408	0.270
VW Golt 2.0 GTI	0.375	0.254	-0.071	-0.019	0.222
VW Polo 1.4 60	-2.078	0.030	0.271	0.282	-0.020

Figure 5 - Coordonnées factorielles des individus (Autos)

Un exemple. Nous avons réalisé une ACP normée sur les données « Autos ». Les deux premiers facteurs sont porteurs de 97% de l'information disponible. Nous retranscrivons les coordonnées factorielles des individus dans le tableau ci-dessus (Figure 5).

Pour bien comprendre l'équivalence entre les deux espaces de représentation, calculons le carré de la distance entre les véhicules « Citroën ZX Volcane » et « Daihatsu Core » à partir des données centrées et réduites (Figure 4). Nous avons :

$$d_O{}^2(1,2) = (0.029-(-1.381))^2 + (0.308-(-1.573))^2 + \cdots + (-0.125-(-1.539))^2 = 13.3714$$

Si l'on utilise les coordonnées factorielles maintenant,

$$d_F{}^2(1,2) = (0170-(-3.458))^2 + (0.489-0.172)^2 + \cdots + (-0.017-0.250)^2 = 13.3714$$

Nous avons une correspondance exacte entre les distances. L'énorme avantage de passer par les axes factoriels est que nous pouvons nous contenter d'effectuer les calculs sur les axes pertinents. Par exemple, l'ACP nous indique que les 2 premiers facteurs représentent 97% de l'information disponible. Si l'on s'en tient à ces axes, nous aurons comme approximation de la distance entre ces individus :

$$d_{(F1,F2)}{}^2(1,2) = (0170-(-3.458))^2 + (0.489-0.172)^2 = 13.266$$

Elle est relativement précise.

2.3.3 Gain induit par une segmentation

Qualité d'une partition et choix de la variable de segmentation. Lors d'une segmentation, un groupe C est subdivisé en 2 sous-groupes C_1 et C_2. Pour ces derniers, nous pouvons calculer leurs inerties $S(C_1)$ et $S(C_2)$. Nous pouvons alors les additionner pour former l'inertie intra-classes $W(C_1, C_2) = S(C_1) + S(C_1)$. L'objectif de la typologie est de créer des sous-groupes aussi homogènes que possible. Par conséquent, lors de la segmentation, nous choisirons la configuration (la variable de segmentation) qui minimise l'inertie intra-classes W.

En vertu de la formule de Huygens[2], nous pouvons décomposer l'inertie totale en inertie intra-classes (W) et inter-classes (B), à savoir :

$$S(C) = W(C_1, C_2) + B(C_1, C_2)$$

De fait, minimiser W revient à maximiser B. La démarche peut être donc reformulée : **nous choisissons comme variable de segmentation celle qui maximise le gain d'inertie inter-classes B**. B représente l'inertie expliquée par l'appartenance aux groupes.

Ainsi, les arbres de classification constituent une vraie généralisation des arbres de régression. Cette idée peut être étendue aux arbres de décision si l'on considère que l'indice de Gini, utilisé dans la méthode CART (Breiman et al., 1984) par exemple, est une sorte de variance calculée sur variables qualitatives (Light et Margolin, 1971).

[2] C'est une extension multivariée de la décomposition de la variance.

Un exemple. Considérons la première segmentation introduite par SPAD lors du traitement du fichier « autos » (Figure 2). Deux sous ensembles d'observations sont définis :

C1	Prix	Cylindree	Puissance	Poids	Conso
Hyundai Sonata 3000	0.871	1.899	0.925	0.669	1.197
Lancia K 3.0 LS	1.842	1.876	2.282	1.164	1.288
Mazda Hachtback V	0.642	1.123	1.398	0.438	0.787
Opel Omega 2.5i V6	1.587	1.122	1.493	1.559	1.015
Peugeot 806 2.0	0.703	0.308	0.356	1.196	0.787
Renault Safrane 2.2. V	0.675	0.581	0.735	0.999	1.197
Seat Alhambra 2.0	0.658	0.286	0.230	1.444	1.152
Toyota Previa salon	1.850	1.027	0.609	1.987	1.699
Volvo 850 2.5	0.938	1.022	0.893	0.570	0.787
Volvo 960 Kombi aut	1.719	1.084	1.493	1.229	1.653
Moyenne	1.149	1.033	1.041	1.126	1.156
n1	10				

C2	Prix	Cylindree	Puissance	Poids	Conso
Citroen ZX Volcane	0.029	0.308	0.356	-0.188	-0.125
Daihatsu Cuore	-1.381	-1.573	-1.443	-1.803	-1.539
Fiat Panda Mambo L	-1.475	-1.486	-1.538	-1.539	-1.357
Fiat Tempra 1.6 Liberty	-0.476	-0.374	-0.401	-0.385	0.103
Ford Fiesta 1.2 Zetec	-0.711	-0.926	-0.717	-0.847	-1.129
Fort Escort 1.4i PT	-0.665	-0.684	-0.749	-0.287	-0.217
Honda Civic Joker 1.4	-0.698	-0.674	-0.370	-0.188	-0.627
Mitsubishi Galant	0.296	0.308	-0.370	0.340	-0.673
Nissan Primera 2.0	-0.119	0.307	0.451	0.142	0.057
Opel Astra 1.6i 16V	-0.279	-0.346	-0.117	-0.385	-0.764
Opel Corsa 1.2i Eco	-1.116	-1.003	-1.412	-0.995	-1.038
Peugeot 306 XS 108	-0.497	-0.078	-0.117	-0.320	-0.034
Seat Ibiza 2.0 GTI	-0.485	0.284	0.230	-0.402	0.194
Subaru Vivio 4WD	-1.206	-1.880	-1.443	-1.506	-1.038
Suzuki Swift 1.0 GLS	-1.308	-1.333	-1.222	-1.341	-1.494
Toyota Corolla	-0.732	-0.781	-0.717	-0.616	-0.901
VW Golt 2.0 GTI	0.262	0.286	0.230	-0.138	0.194
VW Polo 1.4 60	-0.925	-0.684	-1.064	-0.797	-1.174
Moyenne	-0.638	-0.574	-0.579	-0.625	-0.642
n2	18				

Figure 6 - Les classes C1 et C2 - et leurs barycentres respectifs - issus de la première segmentation (Autos)

Nous calculons les inerties dans chaque sous-groupe : $S(C_1)$ = 11.864 et $S(C_2)$ = 33.652. Nous pouvons en déduire l'inertie intra-classes W = S(C1,C2) = 11.864 + 33.652 = 45.516.

Sachant que l'inertie totale de l'ensemble des observations est T = S(C)= n x p = 28 x 5 = 140. L'inertie inter-classes B est égale à B = 140 – 54.516 = 94.484.

Rapportée à l'inertie totale, nous pouvons dire que la segmentation explique…

$$\frac{B}{T} = \frac{94.484}{140} = 0.675 = 67.5\%$$

…de l'information disponible. C'est le chiffre sur fond bleu à la sortie du premier sommet de l'arbre de SPAD (Figure 2).

Critère de Ward. Construire une subdivision qui maximise l'inertie inter-classes correspond exactement à la méthode de Ward. Le critère de Ward s'écrit :

$$\Delta = \frac{n_1 \times n_2}{n_1 + n_2} d^2(G_1, G_2)$$

Où (n_1, n_2) et (G_1, G_2) sont respectivement les effectifs et les barycentres des classes C_1 et C_2.

Voyons ce qu'il en est sur nos données à partir des informations des tableaux décrivant les classes (Figure 6).

$$\Delta = \frac{10 \times 18}{10+18}\left[(1.149-(-0.638))^2 + (1.033-(-0.574))^2 + \cdots + (1.156-(-0.642))^2\right]$$
$$= 94.484$$

Nous retrouvons l'inertie inter-classes (B) calculée ci-dessus.

La stratégie de subdivision d'un nœud peut être reformulée comme suit : **on choisit comme variable de segmentation celle qui maximise le critère de Ward**.

2.3.4 Hiérarchisation des partitions

Les inerties des groupes $S(C_k)$ s'additionnement pour former l'inertie intra-classes. Il est par conséquent possible de comparer les mérites des segmentations candidates situées sur deux feuilles différentes de l'arbre lors du passage d'une partition en K classes à une partition en (K+1) classes. En effet, si l'on se réfère à l'expression de l'inertie intra-classes pour K groupes :

$$W(C_1, C_2, \ldots, C_K) = S(C_1) + S(C_2) + \cdots + S(C_K)$$

Mettons que la dernière classe C_K est subdivisée en C_{K1} et C_{K2}. Nous avons :

$$W(C_1, C_2, \ldots, C_{K1}, C_{K2}) = S(C_1) + S(C_2) + \cdots + S(C_{K1}) + S(C_{K2})$$

Le **gain d'inertie global** lors du passage de K à (K + 1) classes s'écrit :

$$B = W(C_1, C_2, \ldots, C_K) - W(C_1, C_2, \ldots, C_{K1}, C_{K2})$$
$$= S(C_K) - \left[S(C_{K1}) + S(C_{K2})\right]$$
$$= \Delta$$

Les calculs ne concernent que le sommet à segmenter, mais le gain d'inertie est bien global. L'inertie expliquée (il en est de même pour la part d'inertie expliquée) par chaque segmentation

s'additionne également. Il est de fait possible de comparer les segmentations candidates sur les différentes feuilles.

Un exemple. Revoyons l'arbre généré par SPAD sur les données « autos » (Figure 2). L'inertie expliquée par la première subdivision est de 67.5%. Si nous segmentons le sommet n°16 (auquel est associée la valeur 16.8), l'inertie expliquée par la partition en 3 classes sera égale à 67.5 + 16.8 = 84.3% de l'information disponible. En revanche, si nous traitons le sommet n°17, elle serait égale à 67.5 + 2.0 = 69.5%. La première configuration – découpage du sommet n°16 avec un gain relatif d'inertie de 16.8% – est nettement plus avantageuse.

2.4 Binarisation de la segmentation

Dans les arbres de décision (et de régression), l'opportunité de rendre obligatoirement binaire chaque segmentation peut se discuter (Rakotomalala, 1997 ; pages 183 à 189)[3]. Elle est une nécessité dans les arbres de classification. En effet, nous souhaitons construire une hiérarchie de partitions emboîtées. Une seule classe additionnelle doit être générée à chaque étape pour permettre au praticien de choisir le nombre adéquat de classes K^* sans avoir à jongler avec les contraintes induites par la structure de l'arbre.

Le critère de Ward est utilisé pour choisir la variable de segmentation sur un nœud disions-nous plus haut. Il est calculé après regroupement binaire des modalités pour les variables qualitatives, et après discrétisation pour les variables quantitatives.

2.4.1 Discrétisation des variables de segmentation quantitatives

Principe. Pour une variable candidate X, l'algorithme opère en deux temps : (1) les valeurs de X sont triées de manière croissante ; (2) toutes les coupures candidates – la borne de découpage est située à mi-chemin entre deux valeurs successives de X – sont évaluées de manière à optimiser le critère de Ward (ou l'inertie inter-classes, c'est la même chose).

Voici un code R qui illustre la procédure. Nous recherchons la borne de découpage de la variable « Poids » pour la segmentation de la racine de l'arbre « Autos » (Figure 2).

```
#chargement des données
library(xlsx)
don <- read.xlsx("autos_small_ict.xlsx",header=T,sheetIndex=1)
rownames(don) <- as.character(don[,1])
```

[3] R. Rakotomalala, « Graphes d'induction », Université Lyon 1, 1997.

```
don <- don[-1]
print(head(don))

#fonction de centrage réduction
CR <- function(x){
  n <- length(x)
  m <- mean(x)
  et <- sqrt((n-1)/n*var(x))
  return((x-m)/et)
}

#centrage réduction des variables
don.cr <- as.data.frame(lapply(don,CR))
print(head(don.cr))

#matrice de distance sur données centrées-réduites
n <- nrow(don.cr)
d <- dist(don.cr,"euclidean")

#T = n * intertie totale
T <- n * ((1/n^2)*sum(d^2))
print(T)

#tri selon Poids
index <- order(don$Poids)

#recherche des bornes
#exclure les feuilles de - de 5 individus (paramètre arbre)
bornes <- numeric(0)
gain <- numeric(0)
k <- 0
for (i in 6:(n-4)){
  #première portion des données (branche gauche)
  dg <- don.cr[index[1:(i-1)],] #données
  ng <- nrow(dg) #nb. d'observations
  Sg <- ng * ((1/ng^2)*sum(dist(dg,"euclidean")^2)) #n * inertie
  #seconde portion des données (branche droite)
  dd <- don.cr[index[i:n],] #données
  nd <- nrow(dd) #nb d'obs
  Sd <- nd * ((1/nd^2)*sum(dist(dd,"euclidean")^2))
  #inertie intra
  W <- Sg + Sd
  #borne de découpage - à mi-chemin entre 2 points successifs
  b <- 0.5*(don$Poids[index[i-1]] + don$Poids[index[i]])
  #gain inertie (en pourcentage)
  B <- (T-W)/T*100
  #critère de Ward aurait fourni exactement le même résultat
  #mg <- colMeans(dg)
  #md <- colMeans(dd)
  #delta <- (ng*nd)/(ng+nd)*sum((mg-md)^2)
  #B <- delta/T*100
```

```
  #collecter
  k <- k+1
  bornes[k] <- b
  gain[k] <- B
}

#affichage : borne de découpage et gain associé
print(cbind(bornes,gain))
```

Nous obtenons le tableau suivant :

Bornes (Poids)	Gain (%)	Commentaire
917.5	41.4	
947.5	45.7	
982.5	51.4	
1042.5	55.7	
1077.5	52.5	
1080.0	53.2	
1090.0	55.2	
1105.0	56.0	
1125.0	60.9	
1140.0	59.7	
1147.5	65.8	
1197.5	64.9	
1270.0	64.8	
1315.0	67.5	**Solution optimale**
1350.0	60.4	
1385.0	54.5	
1450.0	46.5	
1525.0	41.7	
1555.0	29.5	

La borne de découpage optimale est Poids = 1315 avec un gain de 67.5%. Ce sont bien les valeurs que l'on retrouve dans l'arbre (Figure 2).

Temps de calcul. La somme des calculs à effectuer semble considérable, surtout si l'on pense qu'ils doivent être réitérés sur chaque nœud à segmenter. En réalité, le dispositif est très rapide et peut appréhender de grandes bases de données. Il existe des algorithmes de tris efficaces. Il est également possible de pré-trier les valeurs et de conserver les index de manière à ne pas avoir à répéter l'opération sur chaque nœud (mais au prix d'une occupation supplémentaire de l'espace mémoire). L'évaluation de chaque coupure peut être réalisée en temps linéaire puisque le critère de Ward ne repose que sur le calcul des barycentres conditionnels.

Réutilisation d'une variable de segmentation quantitative sur plusieurs sommets. Le découpage étant binaire à chaque nœud, la même variable continue peut être réintroduite à différents niveaux de l'arbre, avec des bornes de découpages différents cependant (Figure 7).

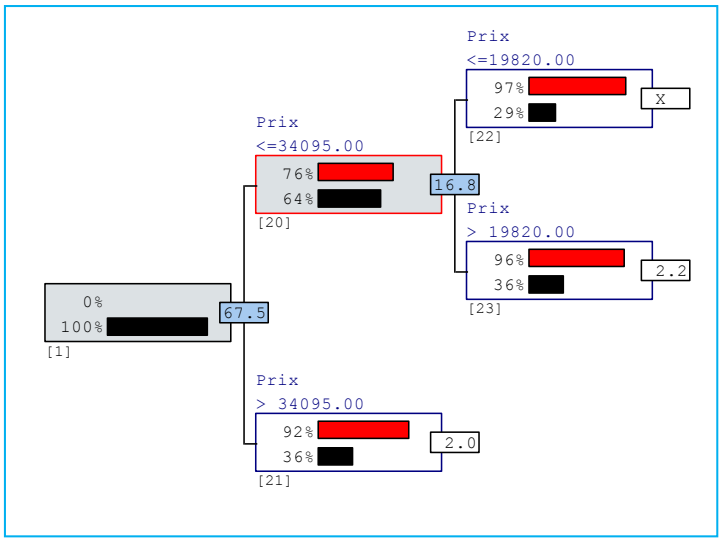


Figure 7 - Arbre de classification avec deux segmentations successives basées sur la variable "prix" (Autos)

2.4.2 Regroupement des modalités des variables de segmentation qualitatives

Lorsque la variable candidate est ordinale, il suffit d'ordonner les modalités et de tester les combinaisons binaires. Il y a (L-1) cas à tester si la variable présente L modalités. Nous nous retrouvons dans une procédure analogue au traitement des variables quantitatives.

La situation se complique compliquée lorsque la variable est nominale. Tester toutes les combinaisons possibles revient à tester $(2^{L-1}-1)$ cas, ce qui est impraticable dès que le nombre de modalités augmente. Pour donner un ordre d'idée, si L=20, il y a 524.287 configurations binaires à évaluer. Il faut impérativement trouver une stratégie proposant de « bons » résultats avec un temps de calcul raisonnable.

Une piste simple consiste à effectuer une classification ascendante hiérarchique (CAH) sur les modalités de la variable de segmentation candidate. L'inertie est toujours calculée sur l'ensemble des variables actives. Il s'agit d'une approche pas à pas, le nombre de tests est connu à l'avance, la complexité de calcul est quadratique par rapport au nombre de modalités. Et surtout la démarche est cohérente avec le processus de construction de l'arbre puisqu'il s'agit toujours de trouver le regroupement binaire des modalités qui maximise l'inertie inter-classes. Certes, les inconvénients de ce type d'optimisation sont connus. Des solutions sous-optimales peuvent se faire jour. Mais on peut se demander finalement si, en lissant ainsi l'exploration de l'espace des

solutions, nous ne nous prémunissons pas du surapprentissage. Les combinaisons « optimales » collent trop aux données traitées la plupart du temps. Elles ingèrent les particularités de l'échantillon d'apprentissage, non transposables à la population.

Dans la copie d'écran ci-dessous (Figure 8), nous montrons le processus de regroupement de modalité de la variable « ancienneté » pour le fichier « crédit » (dont nous reparlerons plus loin, section 4). La variable a été discrétisée et traitée comme une variable nominale dans cet exemple. Nous constatons que la dichotomie la plus pertinente correspond à {ancienneté + de 12 ans} d'un côté (les « vieux » clients), et {anc. 1 an ou moins, anc. de 1 à 4 ans, anc. de 4 à 6 ans, anc. de 6 à 12 ans} de l'autre (les clients plus récents).

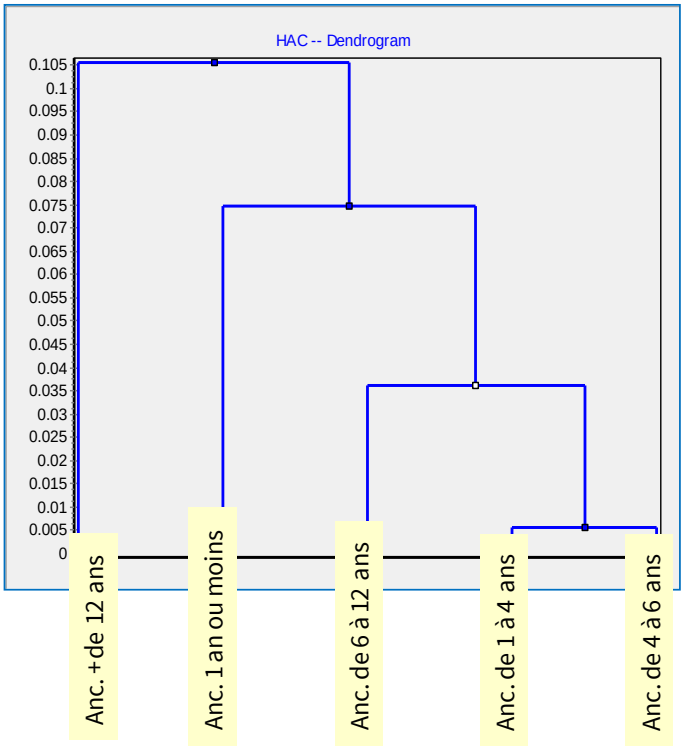


Figure 8 - Regroupement des modalités de "ancienneté" (Fichier "Crédit")

Une variable qualitative peut apparaître plusieurs fois dans l'arbre, avec des regroupements différents. Lors d'une segmentation, seules les modalités de la variable présentes sur le chemin analysé sont concernées bien entendu.

2.5 Détermination du nombre de groupes

La détermination du nombre de classes est le serpent de mer de la classification automatique. Des solutions et indicateurs existent. Mais elles sont souvent très spécifiques, et finalement peu convaincantes.

L'interprétation experte des résultats est la première solution pratique que l'on retrouve dans la littérature. En effet, mettre en avant des groupes qui ne correspondent à rien en termes métiers paraît peu raisonnable. Encore faut-il disposer du recul suffisant pour pouvoir lire correctement les informations qu'ils recèlent.

Les arbres de classification, qui est technique descendante, fournit une hiérarchie de solutions emboîtées. Nous sommes dans une situation analogue à la classification ascendante hiérarchique. Il est dès lors possible de suivre l'évolution des critères d'évaluation des partitions en fonction du nombre de classes. Une « inflexion » – la fameuse loi du coude – dans cette évolution indique souvent une modification de la structure des données. Nous pouvons ainsi nous référer à la décroissance des gains d'inertie inter-classes consécutifs à chaque subdivision (Figure 9), ou encore considérer la croissance de l'inertie expliquée en fonction du nombre de classes (Figure 10).

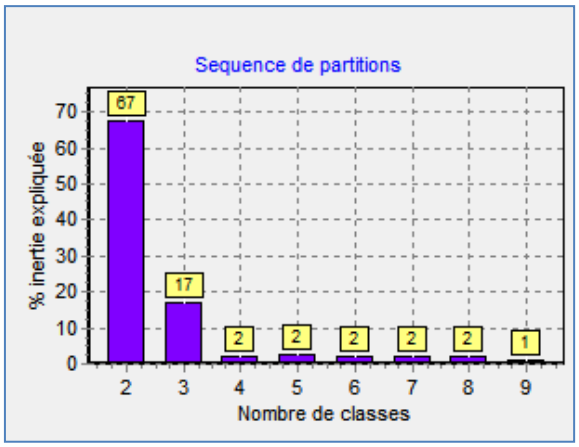


Figure 9 - Gain d'inertie pour chaque segmentation (Autos)

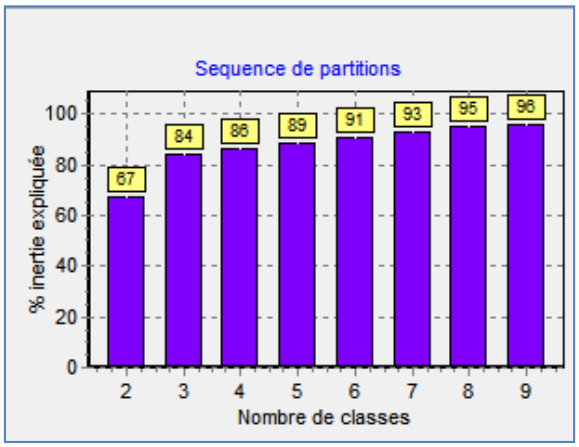


Figure 10 - Inertie expliquée en fonction du nombre de classes (Autos)

Les deux points de vue semblent converger sur une partition en 3 classes.

Bien sûr, d'autres pistes existent. Des paramètres inspirés des arbres de décision peuvent aider à guider la construction des arbres de classification : l'effectif minimum pour segmenter (nombre d'observation minimum sur un sommet pour initier une segmentation) ; l'effectif d'admissibilité (nombre d'observations minimum sur l'ensemble des feuilles pour valider une segmentation) ; pourquoi pas des tests MANOVA (multivariate analysis of variance) qui seraient une variante de la méthode AID/CHAID ; etc. La construction en deux temps, expansion (growing) et post-élagage (pruning), inspiré de CART (Breiman et al., 1984) peut aussi s'avérer fructueuse dans ce nouveau contexte. Tanagra propose cette solution[4]. Il cherche le « coude » sur la courbe de décroissance de l'inertie intra-classe calculée sur l'échantillon de validation.

2.6 La question du déploiement

Quelques solutions pour le déploiement. La question est peu traitée dans les articles scientifiques et les ouvrages. Elle est pourtant cruciale dans la pratique. Le data mining a pour vocation de mettre en évidence les phénomènes de causalité dans les informations collectées pour en tirer parti par la suite, d'une manière ou d'une autre. Mais il se doit également d'être opérationnel. Un ancien sondage (mai 2009)[5] des KDnuggets permet d'y voir un peu plus clair. Concernant l'industrialisation, une grande partie des enquêtés (35.8%) s'appuient directement sur l'outil ayant servi à la construction du modèle. J'avais moi-même montré comment cela pouvait être possible en m'appuyant sur le logiciel R[6] et le package « filehash »[7]. Dans un contexte de déploiement à grande échelle, cette solution n'est pas tenable. Un logiciel de data mining a pour vocation de créer des modèles. Le faire jouer un autre rôle induit des contraintes qui pèsent sur l'efficacité du dispositif. Cela impliquerait également la nécessité de déployer le logiciel de data mining sur tous les serveurs dédiés, uniquement à des fins de production et non d'analyse.

Une seconde partie des sondés (25.3%) s'appuient sur des systèmes de gestion de base de données, en utilisant le langage SQL. Seuls les modèles à base de règles sont exploitables dans ce cas. Une autre partie (16.8%) propose d'implanter le modèle en l'implémentant dans un

[4] « Arbres de classification », avril 2008 ; http://tutoriels-data-mining.blogspot.fr/2008/04/arbres-de-classification.html

[5] « Data Mining deployment poll », http://www.kdnuggets.com/polls/2009/deployment-data-mining-models.htm.

[6] http://www.r-project.org/

[7] http://tutoriels-data-mining.blogspot.fr/2010/06/deploiement-de-modeles-predictifs-avec.html

langage de programmation quelconque (Java par exemple). Nul doute que le dispositif est très efficace à l'usage. Mais cette démarche requiert des compétences en codage hors de portée du tout venant. Il nécessite par ailleurs un processus de validation (unitaire, intégration) qui peut s'avérer lourd. D'autant plus contraignant s'il apparaît nécessaire de mettre à jour régulièrement les modèles.

La solution PMML. Une partie des sondés disent utiliser le langage de déploiement PMML (Predictive Model Market Langage) (13.7% en 2009, sachant qu'il était à 4.2% en 2006, le pourcentage est vraisemblablement plus élevé encore aujourd'hui). C'est une solution viable dans un contexte de déploiement à grande échelle nécessitant des mises à jour régulières. J'avais exploré cette idée dans un tutoriel[8]. Il prend sa pleine mesure lorsque le format est accepté par un outil de management de données. J'avais utilisé à cet effet PDI-CE (Pentaho Data Integration - Community Edition)[9]. Mais la solution proposée par le standard PMML pour la classification automatique (clustering models[10]) m'a un peu déçu j'avoue. Le format ne prend en compte que les modèles basés sur des barycentres conditionnels (comme le produirait un k-Means) ou basés sur des distributions conditionnelles (comme le produirait l'algorithme EM). Nous devons faire face à plusieurs écueils. Il faut que le format de description intègre les éventuelles informations de transformation de données pour qu'il puisse l'appliquer aux individus supplémentaires. Le traitement de chaque individu nécessite des calculs de distances. Certes, le nombre de classes est faible généralement. Mais si on peut s'en passer c'est mieux.

A titre d'exemple, j'ai réalisé un K-Means en 3 classes sur données standardisées sur Knime.

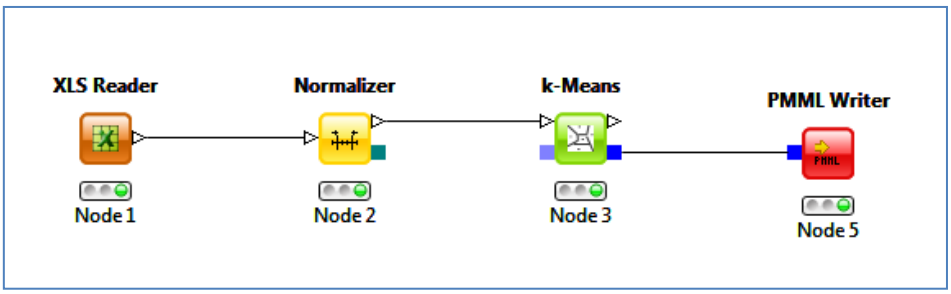


Figure 11 - "Filière" sous Knime pour exporter le modèle K-Means dans un fichier PMML

Nous obtenons la description suivante (fichier au format PMML) :

[8] « Le format PMML pour le déploiement de modèles », septembre 2010.

[9] http://community.pentaho.com/

[10] http://www.dmg.org/v4-1/ClusteringModel.html

```
<PMML xmlns="http://www.dmg.org/PMML-4_1" version="4.1">
 <Header copyright="Maison">
  <Application name="KNIME" version="2.9.1"/>
 </Header>
 <DataDictionary numberOfFields="5">
  <DataField name="Prix" optype="continuous" dataType="double">
   <Interval closure="closedClosed" leftMargin="-1.4486461509126525" rightMargin="1.8169887807162883"/>
  </DataField>
  <DataField name="Cylindree" optype="continuous" dataType="double">
   <Interval closure="closedClosed" leftMargin="-1.8456677693109576" rightMargin="1.864679922566038"/>
  </DataField>
  <DataField name="Puissance" optype="continuous" dataType="double">
   <Interval closure="closedClosed" leftMargin="-1.5101971293291756" rightMargin="2.240937675778776"/>
  </DataField>
  <DataField name="Poids" optype="continuous" dataType="double">
   <Interval closure="closedClosed" leftMargin="-1.7701521682192185" rightMargin="1.9516173268286972"/>
  </DataField>
  <DataField name="Conso" optype="continuous" dataType="double">
   <Interval closure="closedClosed" leftMargin="-1.5115167501757454" rightMargin="1.6682666353791546"/>
  </DataField>
 </DataDictionary>
 <ClusteringModel modelName="k-means" functionName="clustering" modelClass="centerBased" numberOfClusters="3">
  <MiningSchema>
   <MiningField name="Prix" invalidValueTreatment="asIs"/>
   <MiningField name="Cylindree" invalidValueTreatment="asIs"/>
   <MiningField name="Puissance" invalidValueTreatment="asIs"/>
   <MiningField name="Poids" invalidValueTreatment="asIs"/>
   <MiningField name="Conso" invalidValueTreatment="asIs"/>
  </MiningSchema>
  <ComparisonMeasure kind="distance">
   <squaredEuclidean/>
  </ComparisonMeasure>
  <ClusteringField field="Prix" compareFunction="absDiff"/>
  <ClusteringField field="Cylindree" compareFunction="absDiff"/>
  <ClusteringField field="Puissance" compareFunction="absDiff"/>
  <ClusteringField field="Poids" compareFunction="absDiff"/>
  <ClusteringField field="Conso" compareFunction="absDiff"/>
  <Cluster name="cluster_0" size="10">
   <Array n="5" type="real">
    1.1278550408798325 1.0142182044279404 1.0225938919709865 1.1053193068678013 1.1353170256875582
   </Array>
  </Cluster>
  <Cluster name="cluster_1" size="8">
   <Array n="5" type="real">
    -1.0867116737097104 -1.1862552851776753 -1.173060013580837 -1.159296522836354 -1.1868205593972525
   </Array>
  </Cluster>
  <Cluster name="cluster_2" size="10">
   <Array n="5" type="real">
    -0.2584857019120629 -0.06521397628579924 -0.08414588110631807 -0.17788208859871896 -0.18586057816975918
   </Array>
  </Cluster>
 </ClusteringModel>
</PMML>
```

Figure 12 - Description PMML du modèle K-Means (Données Autos)

Une lecture pourtant attentive de la documentation en ligne de PMML (Clustering Models) ne m'a pas permis de savoir s'il était possible d'introduire des informations de transformation (la transformation à l'aide de l'outil « Normalizer » dans le diagramme, Figure 11). Je ne sais pas très bien non plus comment procéder si l'on procède à une classification sur facteurs. Il faudrait dans ce cas enchaîner deux fichiers PMML : la première intègre les coefficients permettant de calculer les scores à partir des variables originelles, la seconde décrivant les centres de classes relativement aux axes factoriels.

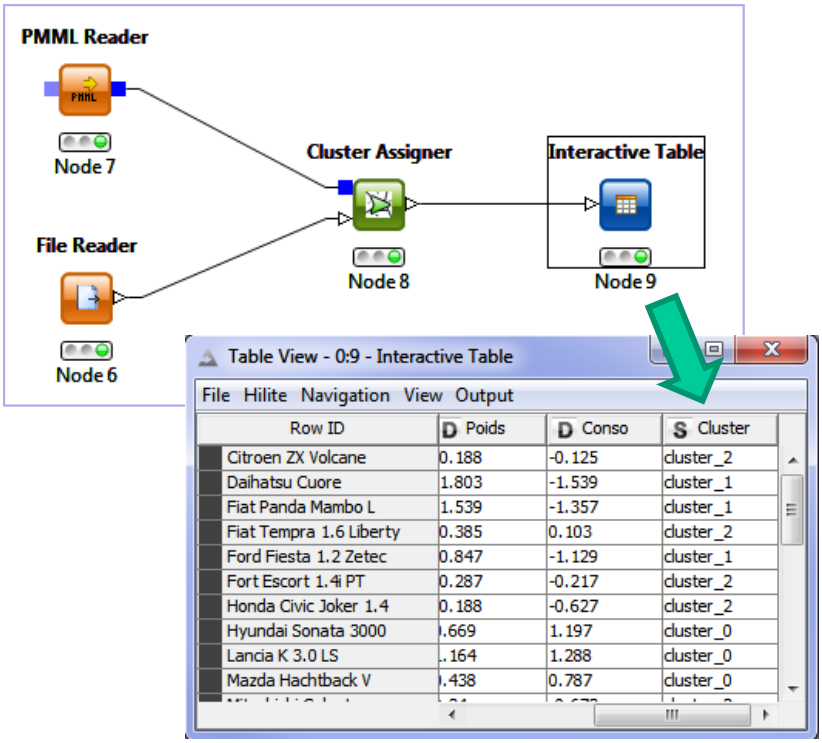


Figure 13 - Déploiement sous Knime - Fichier « Autos » - Variables centrées et réduites

J'ai appliqué le modèle PMML sur les données centrées réduites (Figure 13). J'ai constaté que les groupes assignés correspondent exactement à ceux définis lors de l'apprentissage. Le dispositif est parfaitement opérationnel. Mais j'ai du pour cela centrer et réduire explicitement les variables du fichier d'entrée. Pourquoi pas après tout. Nous devons simplement veiller à utiliser les moyennes et écarts-type calculés sur l'échantillon ayant servi à la construction des groupes si nous désirons déployer sur sur d'autres données.

Déploiement des arbres de classification. Les arbres de décision se démarquent clairement en matière de déploiement. La description d'un arbre au format XML ne pose aucun problème, nous pourrons prendre comme point de départ la structure PMML pour les arbres de décision (voir PMML – Tree Models).

Un arbre peut être converti en base de règles sans perte d'information, le format consacré aux ensembles de règles pourrait convenir également (voir PMML – Rule Set). D'autres expressions sont possibles. SPAD par exemple propose une exportation des règles sous forme de requête SQL. Pour le sommet n°18 de l'arbre (Figure 2), nous aurons :

```
SELECT *
FROM <TABLE>
WHERE(((Prix<=19820.00) AND(Poids<=1315.00)));
```

Manifestement, le passage aux règles logiques ouvre beaucoup de possibilités en matière de déploiement dans les systèmes d'information.

2.7 Description et interprétation des classes

L'interprétation des groupes à l'aide des variables actives et illustratives reste décisive dans notre contexte, même si l'arbre fournit par ailleurs un modèle simple en opérant une sélection de variables. Cela ne veut pas dire que les autres variables qui n'apparaissent pas dans les règles d'affectation ne sont pas importantes pour autant dans la compréhension de la structure sous-jacente aux classes.

2.7.1 Interprétation multivariée

Jouer sur la complémentarité entre l'analyse factorielle et la classification automatique est souvent bénéfique[11]. Nous disposons d'une vision multivariée du rôle des variables dans la constitution des groupes. Pour le fichier « Autos », nous avons réalisé une ACP, le premier facteur résume à lui seul 92.56% de l'information disponible (Figure 14).

Axis	Eigen value	Difference	Proportion (%)	Histogram	Cumulative (%)
1	4.627989	4.423144	92.56 %		92.56 %
2	0.204845	0.124311	4.10 %		96.66 %
3	0.080534	0.026105	1.61 %		98.27 %
4	0.054429	0.022227	1.09 %		99.36 %
5	0.032203	-	0.64 %		100.00 %
Tot.	5.000000	-	-	-	-

Figure 14 - Tableau des valeurs propres (ACP - Autos)

Il est très fortement corrélé avec toutes les variables (Figure 15).

Factor Loadings [Communality Estimates]

Attribute	Axis_1		Axis_2	
-	Corr.	% (Tot. %)	Corr.	% (Tot. %)
Prix	0.97941	*96 % (96 %)*	-0.06595	*0 % (96 %)*
Conso	0.96290	*93 % (93 %)*	-0.12750	*2 % (94 %)*
Cylindree	0.96037	*92 % (92 %)*	0.21551	*5 % (97 %)*
Puissance	0.95579	*91 % (91 %)*	0.25142	*6 % (98 %)*
Poids	0.95170	*91 % (91 %)*	-0.27310	*7 % (98 %)*
Var. Expl.	4.62799	*93 % (93 %)*	0.20484	*4 % (97 %)*

Figure 15 - Corrélation et Cos² des variables sur les 2 premiers facteurs (Autos)

[11] « La complementarité CAH et ACP », mars 2008.

Lorsque nous représentons les individus dans le premier plan factoriel en les illustrant selon leur cluster d'appartenance (attribué par l'arbre de classification, voir page 4), nous distinguons nettement la partition des véhicules en « petite » (C1), « moyenne » (C2) et « grande » voitures (C3), différenciation opérée essentiellement sur le premier plan factoriel (Figure 16).

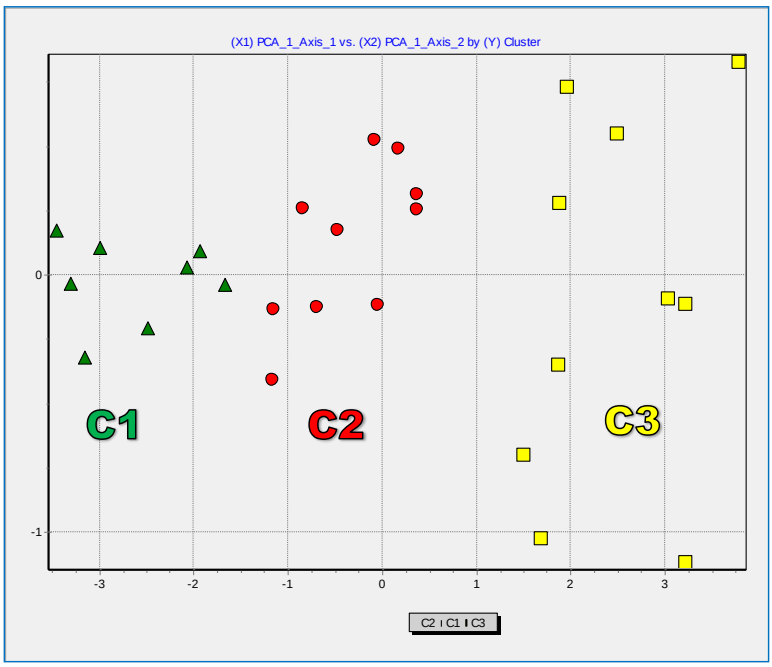


Figure 16 - Positionnement des groupes dans le premier plan factoriel

Les voitures sont désignées « petites » par la règle d'affectation « **Si** poids ≤ 1315 **et** prix ≤ 19820 **Alors** groupe n°1 (C1) ». On se rend compte avec l'ACP qu'elles consomment peu également, qu'elles ont un petit moteur et sont peu puissantes. Ces informations additionnelles fournies par l'ACP sont très importantes pour mieux appréhender la nature de ce groupe.

Les voitures « moyennes » (C2) …, etc. Sans rentrer dans les détails, il y a manifestement un effet taille très fort sur ce premier facteur[12]. Peut-être faudrait-il aller plus loin pour faire apparaître des résultats plus intéressants. Mais ce n'est pas notre propos dans ce tutoriel.

2.7.2 Interprétation univariée

L'analyse univariée consiste à analyser le rôle que peut jouer chaque variable dans la définition (si elle est active) ou dans l'interprétation (si elle est illustrative) des groupes, indépendamment des autres. La vision est tronquée puisqu'on ne tient pas compte des relations qui peuvent exister entre les variables. Mais elle est plus simple à décrypter.

[12] Voir http://www.jybaudot.fr/Analdonnees/taille.html

Différenciation entres les groupes. L'objectif est de confronter les groupes selon chaque variable. Lorsqu'elle est quantitative, nous procédons à une comparaison des moyennes conditionnelles (une ANOVA – analyse de variance – à 1 facteur le fait très bien) ; lorsqu'elle est qualitative, nous formons un tableau de contingence et nous mesurons le degré de liaison entre la variable et les clusters (un test du KHI-2 fait très bien l'affaire). Attention cependant, il ne s'agit pas d'un véritable test pour les variables actives puisqu'elles ont concouru à la constitution des groupes. On utilisera les résultats à titre purement indicatif dans ce cas. Il (le test) est en revanche parfaitement licite pour les variables illustratives.

Toutes les variables étant quantitatives dans le fichier « autos », seul le test ANOVA (Fisher) est calculé (Figure 17). Toutes les variables peuvent illustrer la différenciation entre les groupes.

Impact individuel des attributs

Attribut	Statistique	ddl 1	ddl 2	Proba crit.
Prix	68.5569 (F)	2	25	0.0000
Cylindree	49.8445 (F)	2	25	0.0000
Puissance	49.2719 (F)	2	25	0.0000
Poids	78.3595 (F)	2	25	0.0000
Conso	122.6748 (F)	2	25	0.0000

Figure 17 - Test ANOVA - Variables vs. clusters (Autos)

Si l'on s'attarde sur « prix », nous constatons que les moyennes conditionnelles sont respectivement de 14933 (petite), 25192 (moyenne) et 42364 (grande voiture) (Figure 18).

Prix / Cluster

Value	Examples	Average	Std-dev
C1	8	14933.1250	3533.0611
C2	10	25192.0000	4432.0219
C3	10	42364.0000	6452.1111
All	28	28393.7500	12386.5652

Variance decomposition

Source	Sum of square	d.f.
BSS	3503698284.3750	2
WSS	638830646.8750	25
TSS	4142528931.2500	27

Significance level

Statistics	Value	Proba
Fisher's F	68.556868	0.000000

Figure 18 - Moyennes de prix conditionnellement aux groupes (Autos)

Différenciation par rapport à l'ensemble de la population. L'autre repère est la totalité des individus, sans distinction de groupe. Pour les variables quantitatives, il s'agit de comparer la moyenne conditionnelle par rapport à la moyenne globale ; pour les qualitatives, nous comparons les proportions pour identifier les sur ou sous représentation des modalités. C'est l'idée de la valeur-test (Morineau, 1984) que l'on retrouve dans la majorité des outils / logiciels issus de l'école française de l'analyse de données. Elle repose sur une comparaison

d'indicateurs calculés sur échantillons imbriqués. Les réserves pour les variables actives restent de mise ici. La valeur-test permet surtout de hiérarchiser l'impact des variables dans la constitution ou l'interprétation des groupes.

Pour la subdivision en 3 classes du fichier « Autos », nous obtenons (Figure 19) :

Description of "Cluster"

Cluster=C1			
Examples			[28.6 %] 8
Att - Desc	Test value	Group	Overral
Continuous attributes : Mean (StdDev)			
Prix	-3.57	14933.13 (3533.06)	28393.75 (12386.57)
Poids	-3.81	838.75 (128.64)	1196.96 (308.99)
Puissance	-3.86	39.88 (10.45)	77.71 (32.26)
Cylindree	-3.90	1069.25 (259.34)	1809.07 (623.66)
Conso	-3.90	6.43 (0.51)	9.08 (2.23)
Discrete attributes : [Recall] Accuracy			

Cluster=C2			
Examples			[35.7 %] 10
Att - Desc	Test value	Group	Overral
Continuous attributes : Mean (StdDev)			
Cylindree	-0.25	1768.40 (257.56)	1809.07 (623.66)
Puissance	-0.33	75.00 (12.43)	77.71 (32.26)
Poids	-0.69	1142.00 (74.32)	1196.96 (308.99)
Conso	-0.72	8.66 (0.81)	9.08 (2.23)
Prix	-1.00	25192.00 (4432.02)	28393.75 (12386.57)
Discrete attributes : [Recall] Accuracy			

Cluster=C3			
Examples			[35.7 %] 10
Att - Desc	Test value	Group	Overral
Continuous attributes : Mean (StdDev)			
Conso	4.40	11.61 (0.73)	9.08 (2.23)
Prix	4.37	42364.00 (6452.11)	28393.75 (12386.57)
Poids	4.28	1538.50 (144.95)	1196.96 (308.99)
Puissance	3.96	110.70 (19.80)	77.71 (32.26)
Cylindree	3.93	2441.60 (339.57)	1809.07 (623.66)
Discrete attributes : [Recall] Accuracy			

Figure 19 - Valeurs-test des variables actives dans la formation des classes (Autos)

On observe que la moyenne de prix est de 14933.13 dans C1 alors qu'il est de 28393.75 pour l'ensemble des individus. La valeur-test associée est de -3.57, le prix moyen est significativement plus faible dans ce groupe [si l'on se réfère à un test à 5%, les valeurs critiques sont (très) approximativement de -2 et +2]. En étudiant le tableau, nous constatons que la nature des 3 classes (« petites voitures » - C1, « moyennes » - C2, « grandes » - C3) est indiscutable avec ce nouveau prisme.

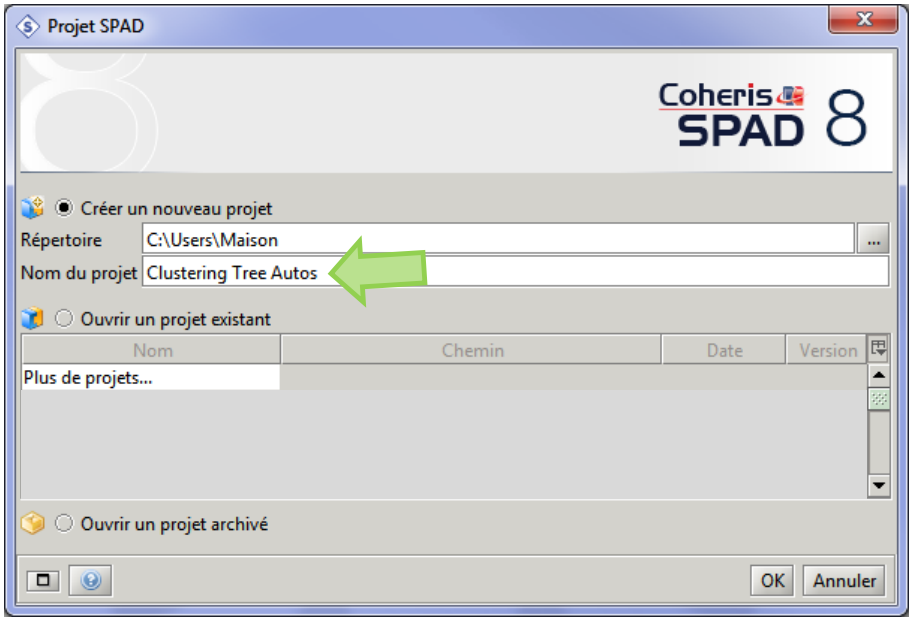
3 Arbres de classification avec SPAD (ICT)

SPAD est le seul outil à ma connaissance qui propose un dispositif interactif pour l'élaboration des arbres de classification. A l'instar de ce qui se fait pour le classement ou la régression[13], l'utilisateur peut inspecter chaque étape de construction du modèle, il peut influer en élaguant manuellement des branches, en choisissant explicitement les variables de segmentation, voire en modifiant manuellement les seuils pour les quantitatives et les regroupements de modalités pour les qualitatives. Ces fonctionnalités sont d'autant plus précieuses qu'à la différence des arbres de décision ou régression, nous devons surveiller plusieurs variables d'intérêt durant le processus de modélisation.

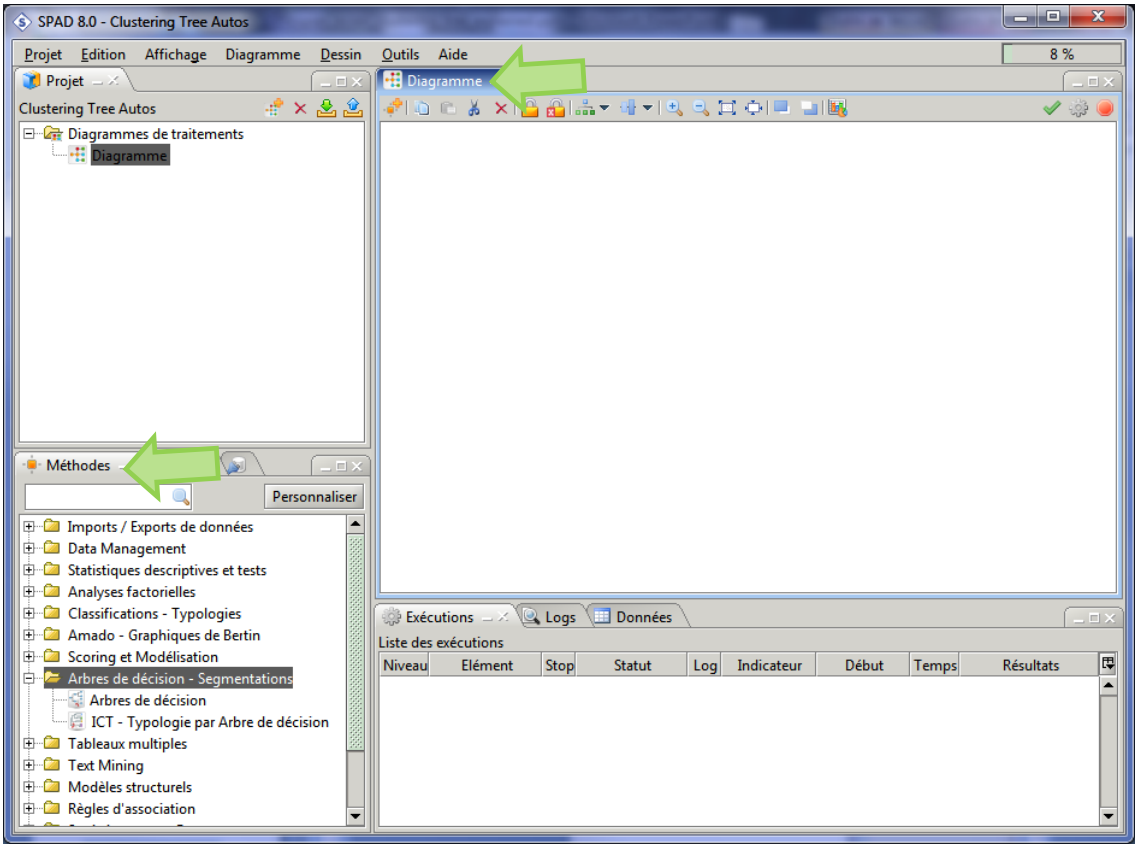
[13] « Nouveaux arbres interactifs dans SPAD 8 », août 2014.

3.1 Démarrage du logiciel et création d'un projet

Nous traitons le fichier des « autos » dans cette section. Au démarrage de SPAD, nous créons un nouveau projet que nous nommons « Clustering Tree Autos ».

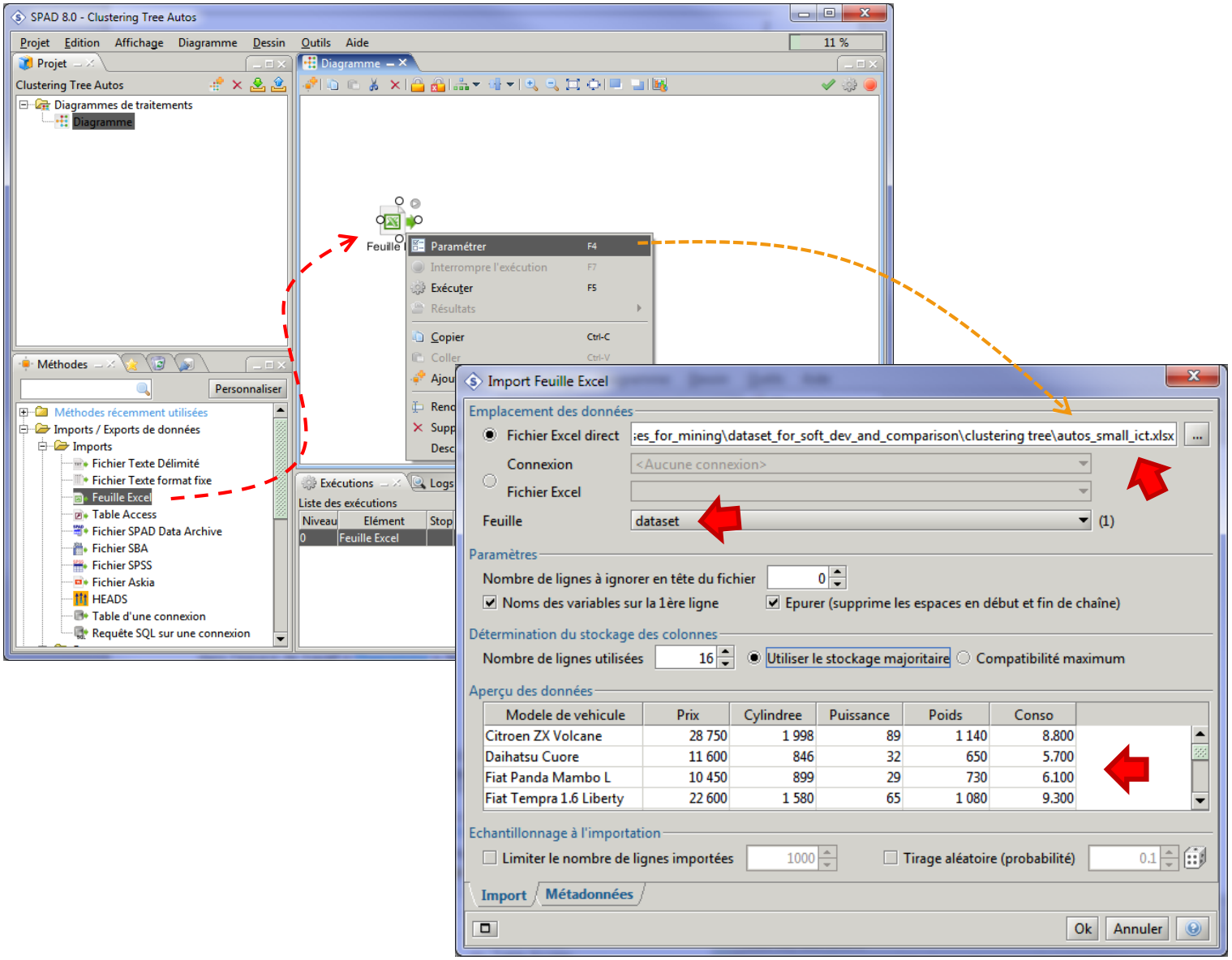


Nous pouvons définir l'enchaînement des opérations à mener sur les données dans l'espace de travail nommé « **Diagramme** », en y plaçant les outils accessibles dans la partie « **Méthodes** » (à gauche) et en les reliant entre elles.



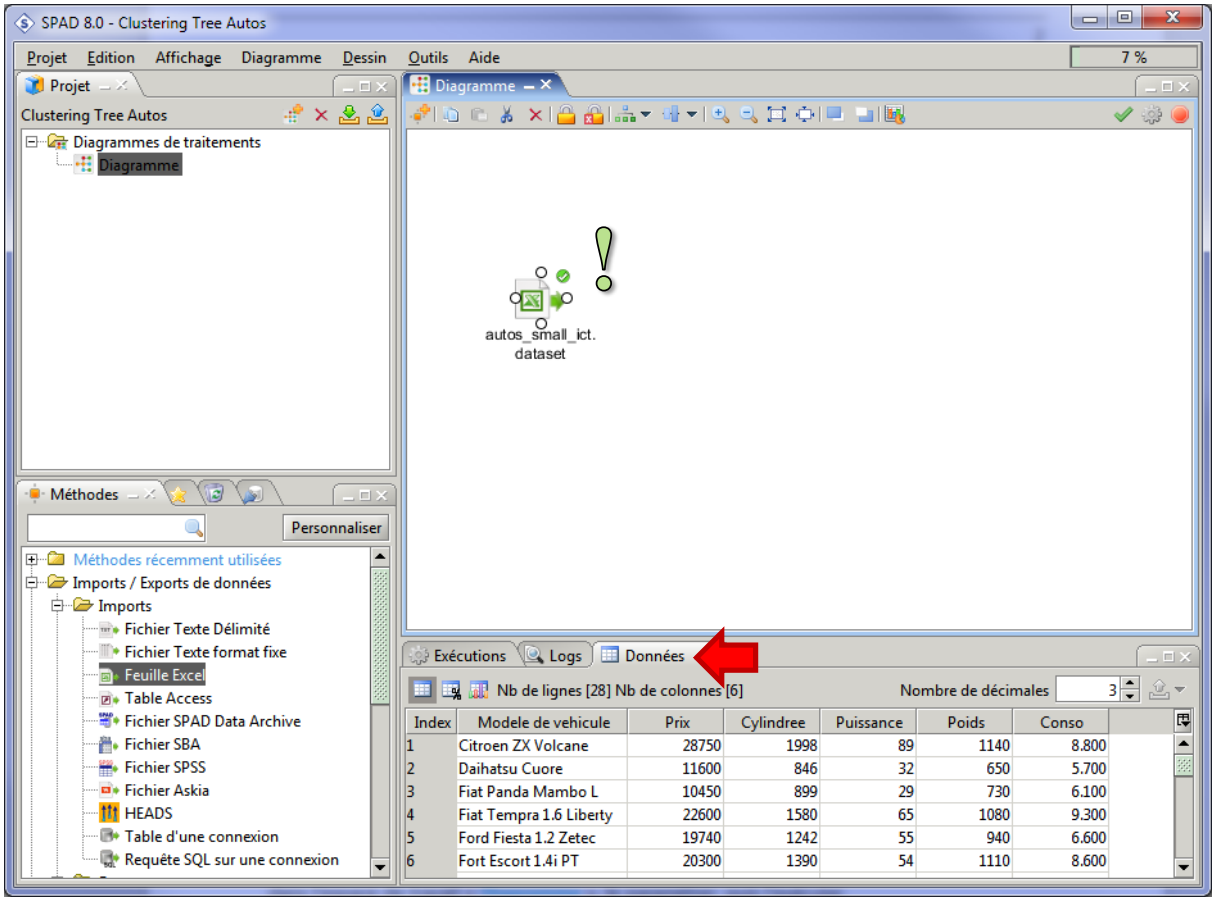
3.2 Importation des données

L'outil « **Feuille Excel** » permet d'importer le fichier « **autos_small_ict.xlsx** ». Nous le plaçons dans le diagramme et nous le paramétrons en actionnant le menu contextuel « **Paramétrer** ».



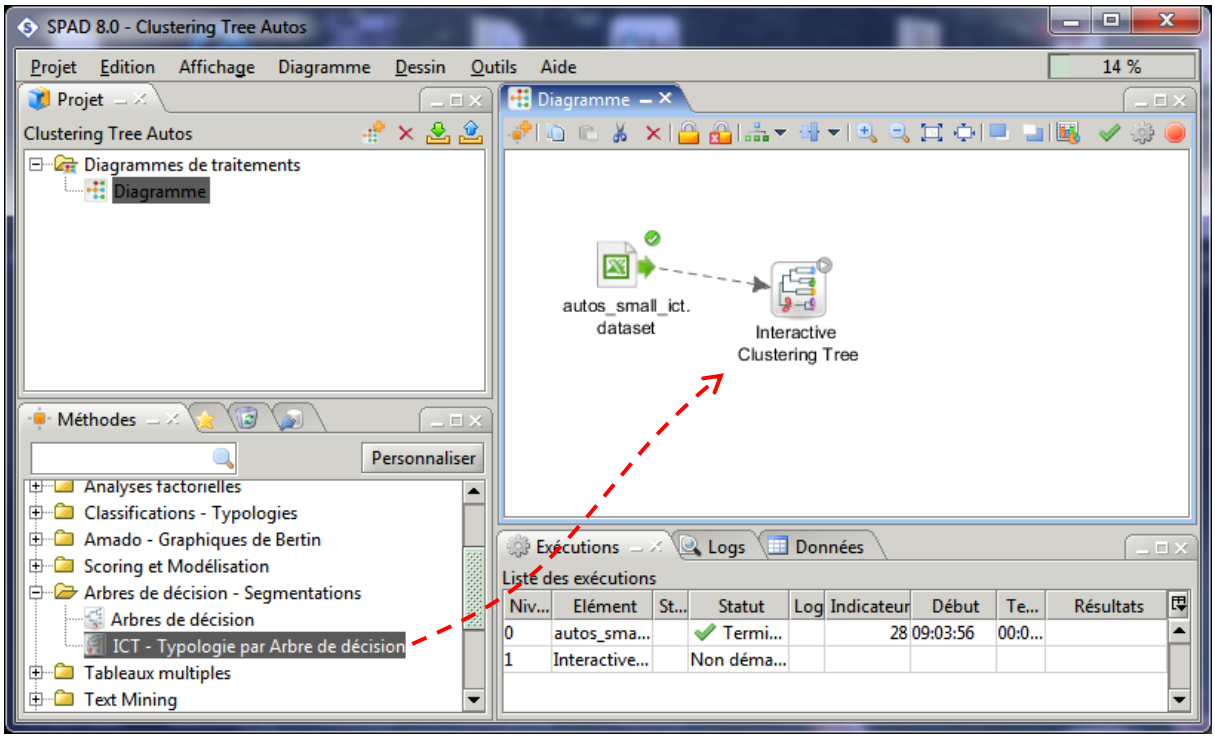
Nous désignons le fichier à traiter (**autos_small_ict.xlsx**), puis nous spécifions la feuille contenant les données (**dataset**). Une prévisualisation apparaît. Nous pouvons ainsi vérifier notre paramétrage et, apporter les éventuelles corrections si cela est nécessaire. Dans les « **Métadonnées** », nous indiquons que « **Modèle de véhicule** » correspond à un identifiant.

Nous validons. SPAD procède à l'importation. Nous pouvons inspecter les données dans l'onglet dédié de la fenêtre de suivi située dans la partie basse de l'interface principale. Nous disposons de 28 observations décrites par 6 variables (la première correspond aux identifiants).



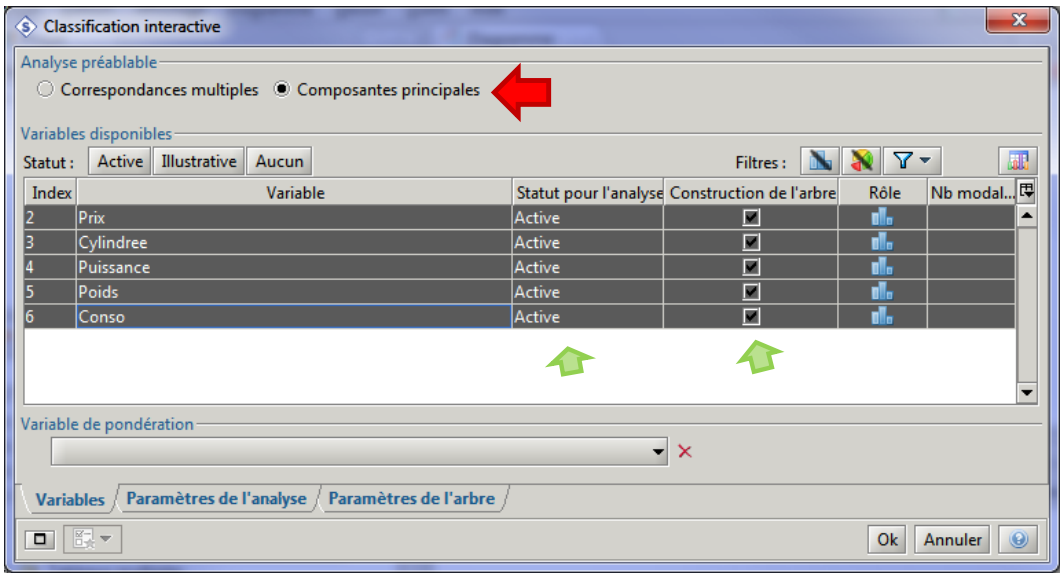
3.3 Paramétrage de l'arbre de classification ICT

Nous insérons ensuite **ICT** (Interactive Clustering Tree – Typologie par Arbres de décision).



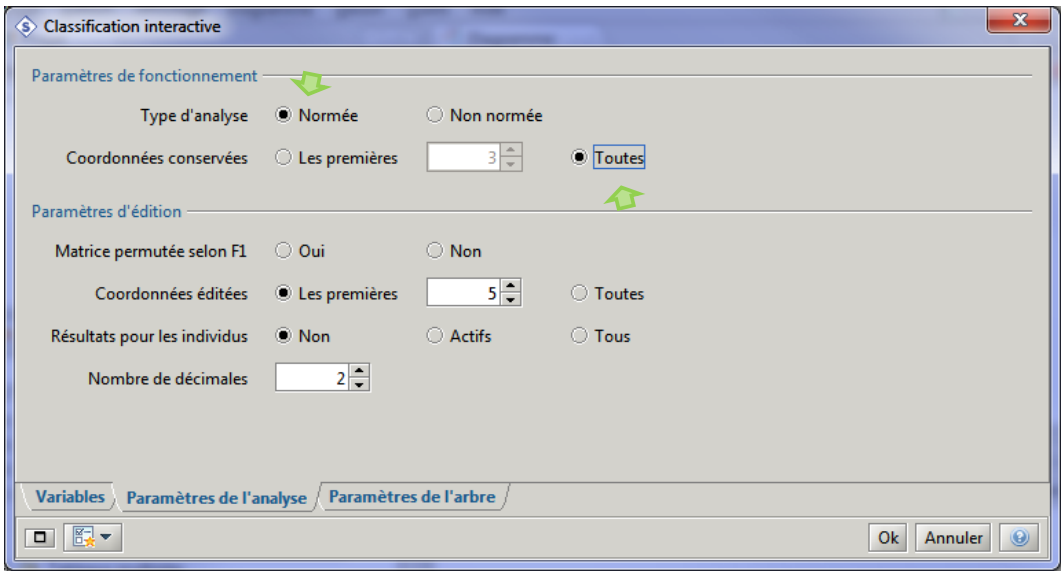
Nous pouvons spécifier les paramètres en cliquant sur le menu contextuel « Paramétrer ».

Variables. Dans l'onglet « Variables », nous choisissons l'Analyse en Composantes Principales comme méthode de préparation de variables. Nous constatons que SPAD passe nécessairement par une analyse factorielle avant de procéder à la classification. Nous désignons ensuite les variables actives (toutes).



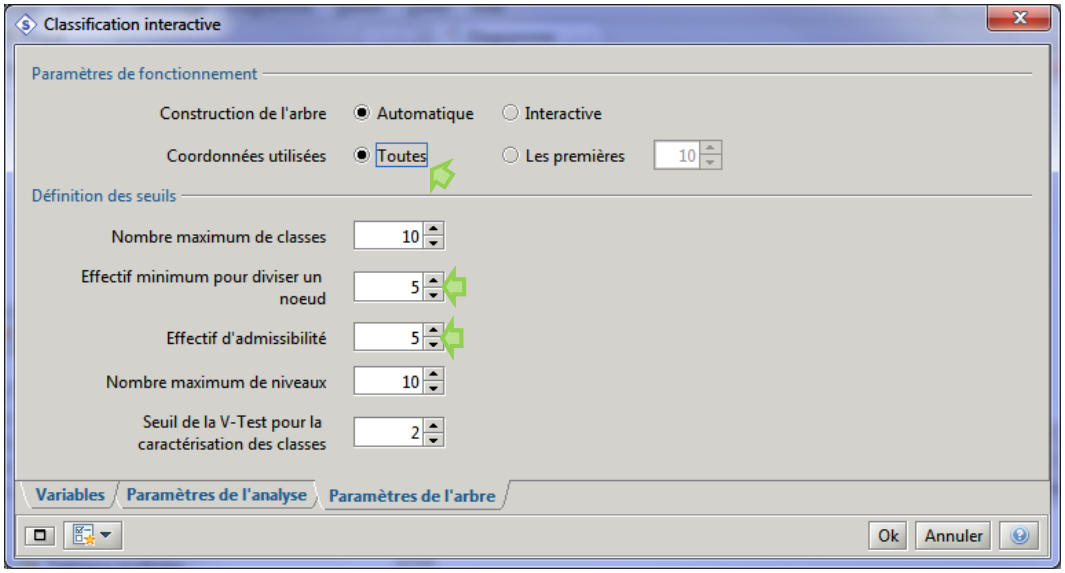
Nous comprenons également qu'elles prendront part à la **construction de l'arbre** c.-à-d. que ces variables peuvent être introduites comme variable de segmentation. La subtilité est importante. Une variable peut servir à calculer l'homogénéité des groupes, mais ne pas être utilisée pour élaborer les règles de désignation des classes, et inversement. Cela ouvre la porte à la prédiction multi-cible que nous traiterons dans un prochain tutoriel. Notons enfin qu'une variable peut être purement « illustrative » comme il est d'usage en analyse de données.

Paramètres de l'analyse.



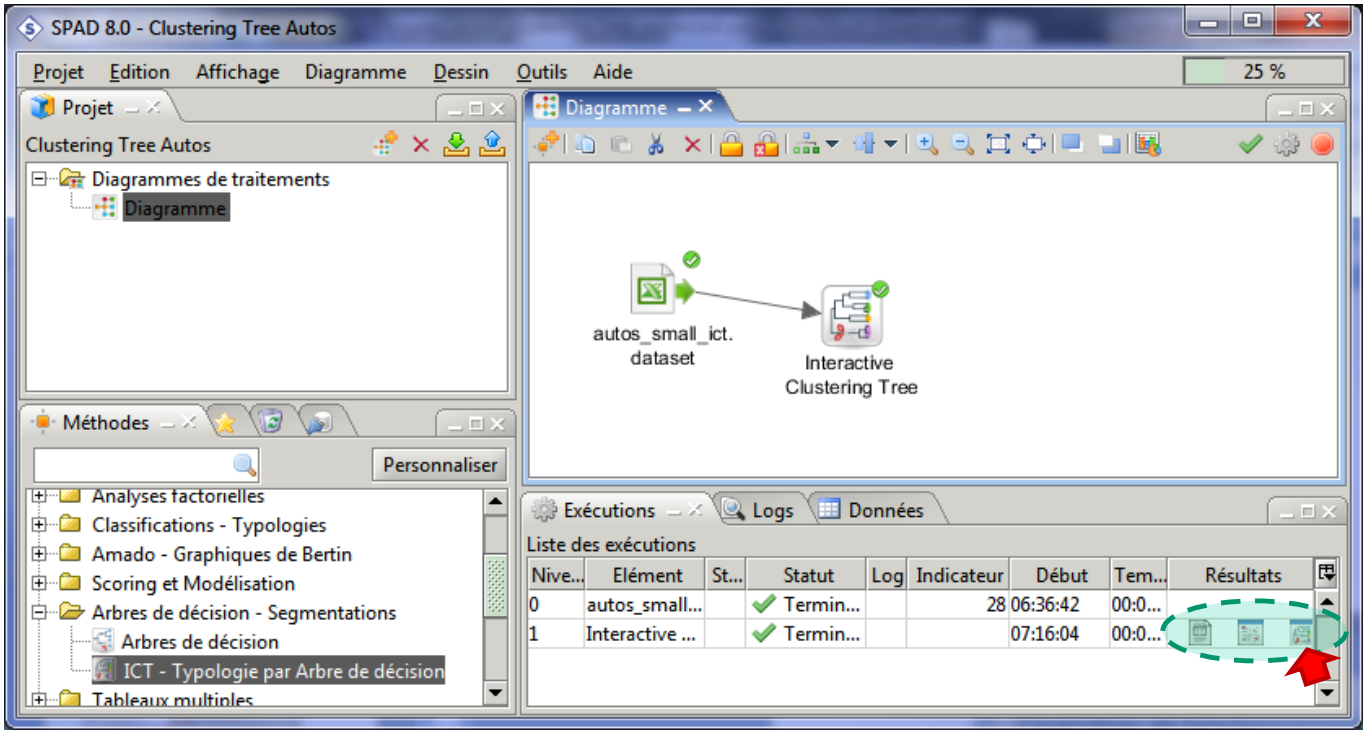
Dans les paramètres de l'analyse, nous indiquons passer par une ACP normée et nous conservons tous les facteurs générées par l'analyse.

Paramètres de l'arbre. Ils permettent de spécifier entres autres les facteurs utilisés pour la typologie. Nous avons la possibilité de filtrer l'information véhiculée par les données en choisissant un nombre réduit de facteurs pour le calcul des inerties.



Nous pouvons aussi définir des paramètres que l'on retrouve habituellement en induction par arbres (effectif minimum pour segmenter, effectif d'admissibilité, etc.).

3.4 Analyse interactive



L'arbre est automatiquement construit après validation des paramètres. Plusieurs types de résultats sont disponibles. Nous nous intéressons à l'arbre interactif en ce qui nous concerne c.-à-d. la 3ème icône dans la fenêtre d'exécution.

3.4.1 Visualisation de l'arbre

L'application de manipulation interactive de l'arbre est démarrée lorsque l'on clique sur la 3ème icône. ICT nous propose une partition en 5 classes compte-tenu de notre paramétrage.

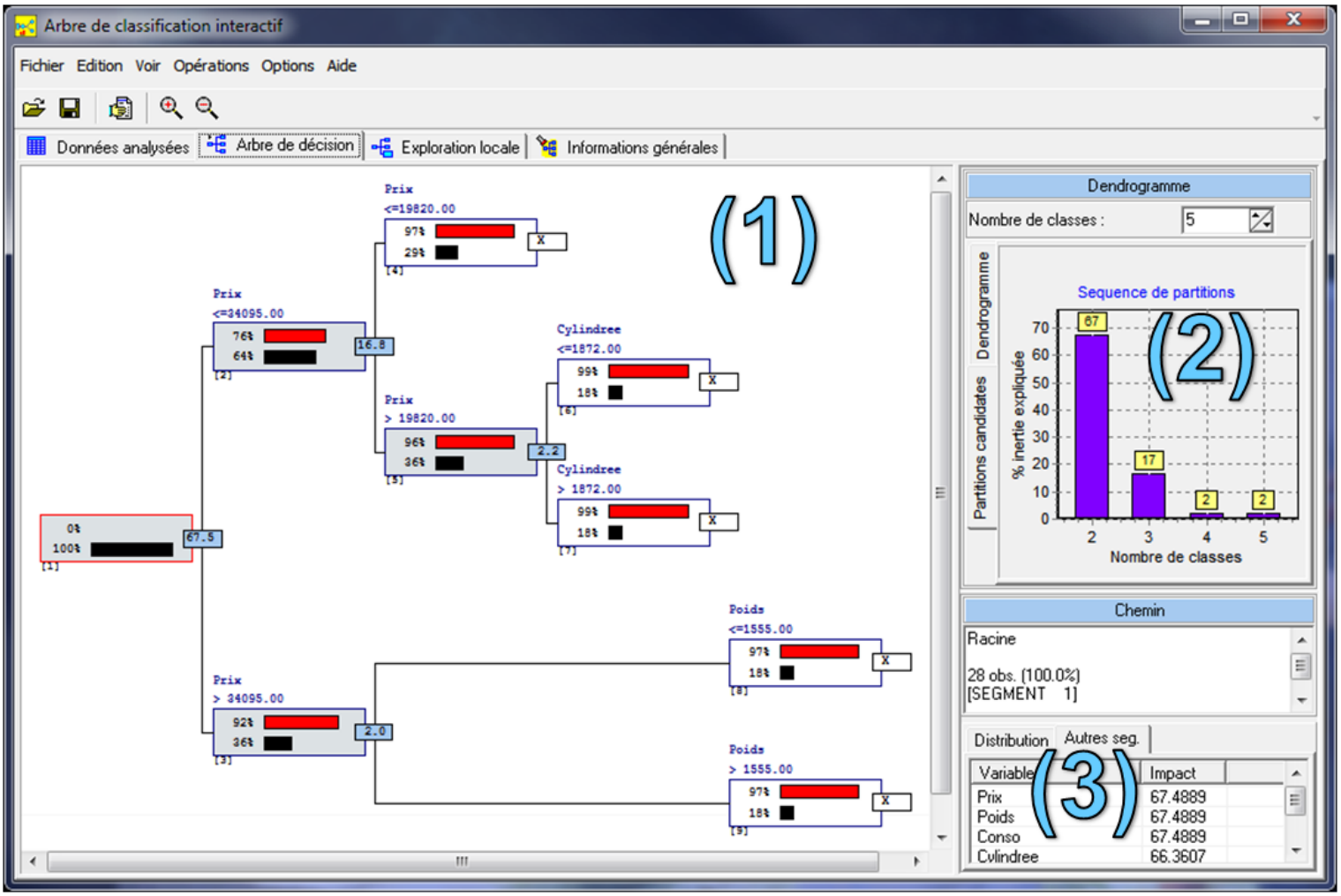


Figure 20 - Fenêtre principale de ICT - SPAD (Autos)

(1) L'arbre de classification est la première information qui attire notre attention. Les règles d'affectation aux groupes (feuilles) sont directement lisibles. La séquence de construction est matérialisée par la différence de niveaux entre les sommets et les valeurs qui leurs sont associées. Dans notre exemple, après la segmentation du sommet n°1 (gain = 67.5% sur fond bleu), c'est le n°2 qui a été partitionné (gain = 16.8%). Puis, le sommet n°5 a été traité avec un gain de 2.2%. Enfin, le sommet n°3 a été segmenté (gain = 2.0%). Si nous réalisons un post-

élagage, ce sont les feuilles n°8 et 9 qui seront éliminées en premier. Ce type de lecture est particulièrement intéressant. On se rend compte qu'il peut s'appliquer aux arbres de décision pour peu qu'on s'appuie sur une mesure de segmentation permettant de traduire un gain global consécutif à chaque partition. Pour la méthode CART avec l'indice de Gini, le gain de pureté traduit cette idée (Breiman et al., 1984 ; pages 32 et 33).

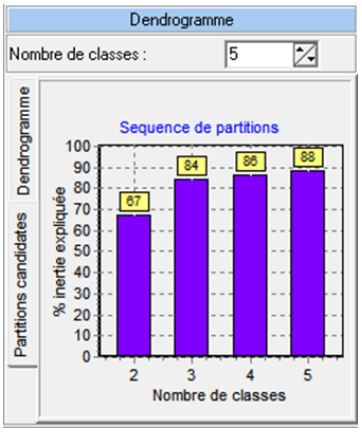
Nous observons deux barres à l'intérieur de chaque sommet. La barre noire indique la taille relative de la sous-population c.-à-d. son support rapporté à l'effectif global. S'il est trop faible, on a plutôt affaire un quelques points atypiques plutôt qu'à un réel phénomène de groupe avec des caractéristiques distinctives. La barre rouge correspond à un indice de compacité. Il indique le degré d'homogénéité du groupe c.-à-d. dans quel mesure les individus situés sur le sommet se ressemblent. Il est rapporté à l'inertie initiale (0% d'homogénéité sur la racine).

(2) Ce graphique indique les gains d'inertie inter-classes relatif à chaque opération de segmentation. Nous notons ici qu'à partir de la 4ème segmentation, le gain est faible. Nous pouvons nous servir de ces indications pour déterminer le nombre adéquat de groupes. Nous en reparlerons plus loin (section 3.4.2).

(3) Les segmentations concurrentes sur le sommet sélectionné, la racine en l'occurrence dans la copie d'écran. Nous constatons que pour le traitement du premier sommet (n°1) : « prix », « poids » et « conso » induisent la même amélioration de l'inertie inter-classes. Elles sont totalement interchangeables ici. Mais si nous modifions la variable de segmentation, nous n'avons pas la garantie que les parties basses de l'arbre seront les mêmes par la suite. Nous explorerons ce thème de manière approfondie plus loin (section 3.4.3).

3.4.2 Détermination du nombre de groupes

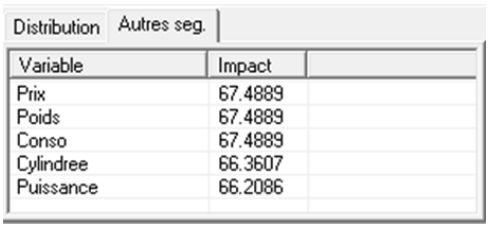
Nous pouvons nous appuyer sur le gain d'inertie expliquée pour déterminer le nombre de classes : lorsque le gain devient trop faible, nous pouvons stopper la construction de l'arbre. La courbe des gains permet de surveiller l'évolution (Figure 20 ; graphique **2**). Nous pouvons disposer d'un point de vue complémentaire en étudiant l'évolution de l'inertie expliquée (gains cumulés) en fonction du nombre de classes. Pour notre exemple, cela donnerait :



Il semble qu'une partition en 3 classes soit la plus appropriée pour les données « autos ». Nous fixons la valeur de « **Nombre de classes** » à 3.

3.4.3 Segmentations concurrentes – Analyse interactive

Revenons à la racine de l'arbre. Dans l'onglet « Autres seg. » (Autres segmentations) en bas à droite de la fenêtre principale, nous observons le gain obtenu si nous découpons le sommet avec l'une des variables listées.

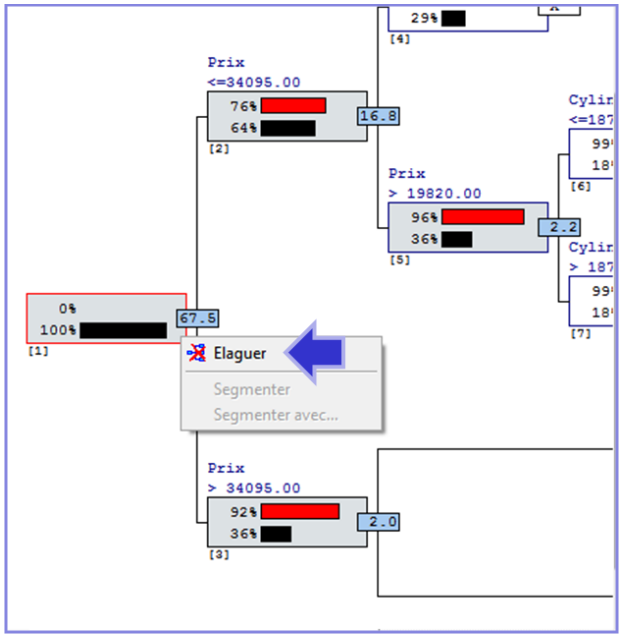


Variable	Impact
Prix	67.4889
Poids	67.4889
Conso	67.4889
Cylindree	66.3607
Puissance	66.2086

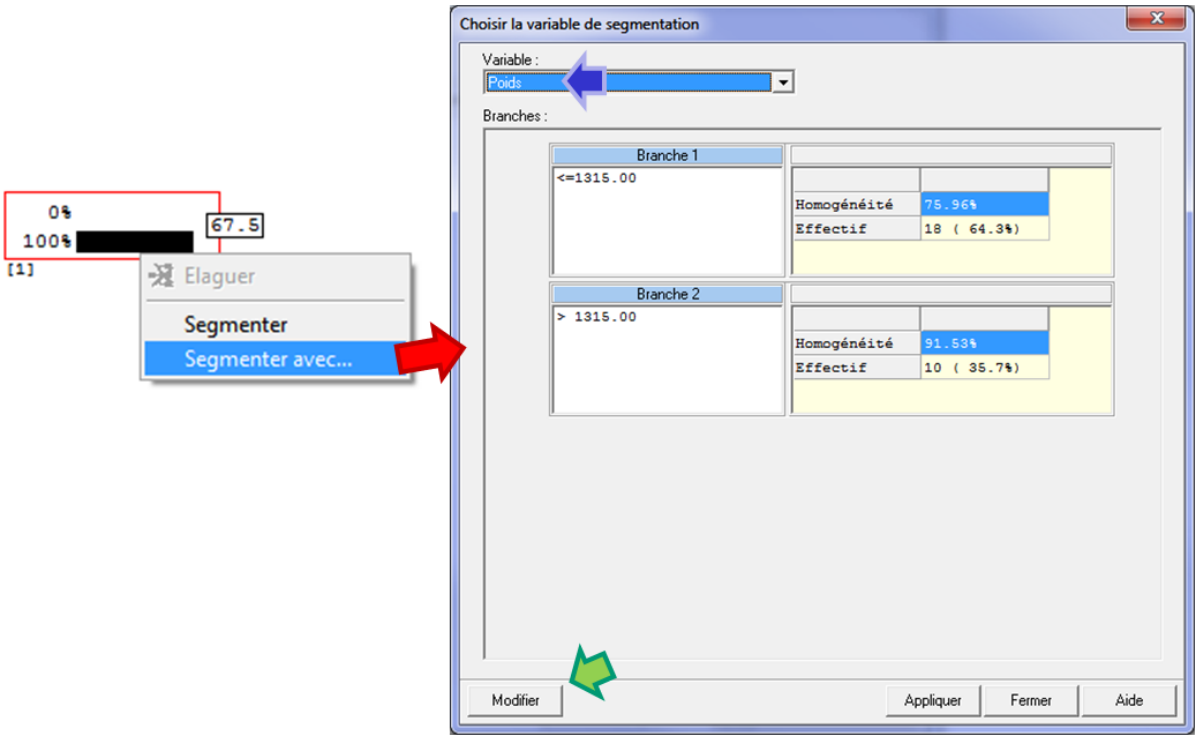
« Prix » est la première, c'est pour cela qu'elle a été utilisée (Figure 20). Mais nous constatons que « Poids » et « Conso » sont de qualité équivalente (gain = 67.4889). Le choix de « prix » est donc purement arbitraire, il a été choisi parce qu'il est placé devant les autres dans la liste des variables. Nous constatons là le principal intérêt des outils interactifs. Nous disposons d'une vision complète de la nature des résultats en inspectant chaque sommet. Certes, des outils non-interactifs (WinIdams[14] ou rpart de R dans sa partir *summary*) peuvent fournir la liste des segmentations concurrentes. Mais analyser des sorties qui peuvent être très touffues s'avère vite fastidieux. Et surtout, construire l'arbre de manière à lui faire choisir tel ou tel variable pour le traitement de tel ou tel sommet n'est pas possible (si un peu quand même dans WinIdams mais au prix d'une gymnastique peu évidente).

[14] http://tutoriels-data-mining.blogspot.fr/2014/10/induction-par-arbre-avec-winidams.html

Justement, nous souhaitons segmenter la racine avec le « Poids ». Pour cela, nous devons d'abord élaguer les parties basses de l'arbre. Nous effectuons un clic-droit sur le sommet. Dans le menu contextuel qui apparaît, nous cliquons sur « **Elaguer** ».



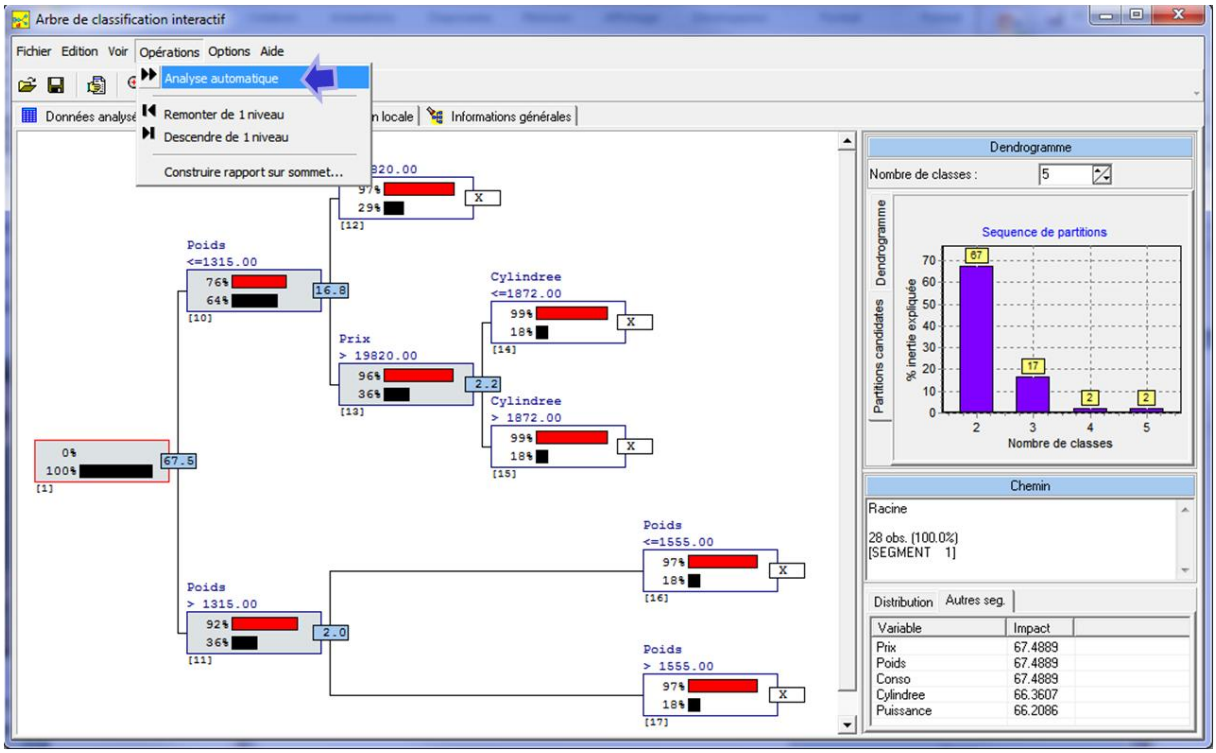
Puis nous actionnons le menu « Segmenter avec… »



Nous sélectionnons la variable « Poids » dans la liste. Notons qu'il est possible de modifier la borne de discrétisation (la valeur 1315) en cliquant sur le bouton « **Modifier** ». De même, il est

possible de modifier les regroupements pour les variables de segmentations qualitatives. Il faut toutefois que l'arbre demeure binaire.

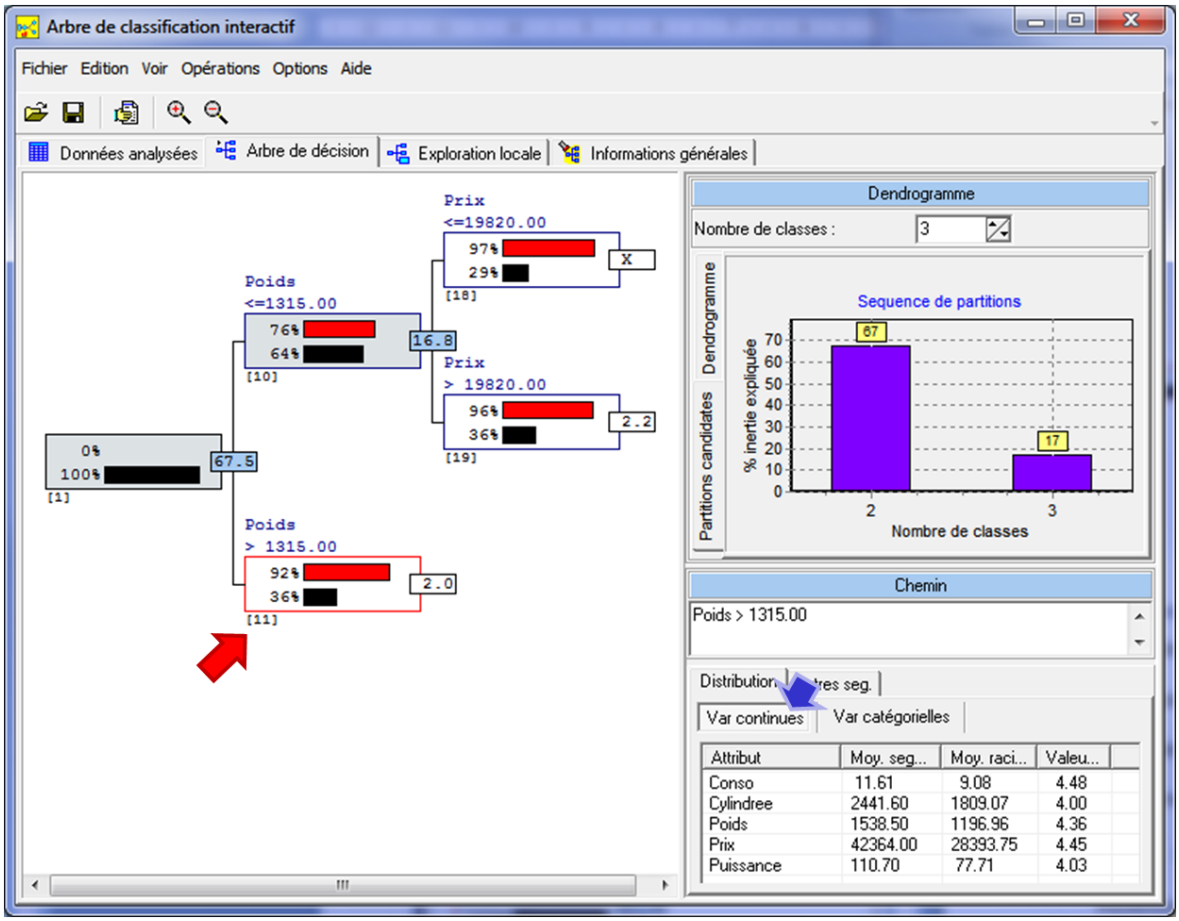
Nous pouvons ainsi construire l'arbre de proche en proche en définissant explicitement la variable à utiliser sur chaque nœud. Mais nous pouvons également laisser l'algorithme définir les parties restantes de l'arbre après cette segmentation de la racine à l'aide de « Poids ». Nous cliquons pour ce faire sur le menu « **Opérations / Analyse Automatique** ». Nous obtenons la structure suivante.



Le reste de l'arbre est identique à la première version (Figure 20). Ce n'est pas le cas généralement. La modification de la première segmentation modifie les choix (variables ou seuils) sur les autres nœuds tout simplement parce que nous n'avons pas exactement les mêmes observations dans les sommets.

3.4.4 Inspection d'un groupe

Revenons à une partition en 3 groupes. Nous souhaitons explorer le sommet correspondant à la règle « Si Poids > 1315 Alors Groupe N°3 ». Nous sélectionnons le sommet dans l'interface graphique et nous activons l'onglet « **Distribution** » dans la partie en bas à droite de la fenêtre principale. Seule la liste « Variable continue » est disponible puisque l'ensemble de nos variables sont quantitatives.



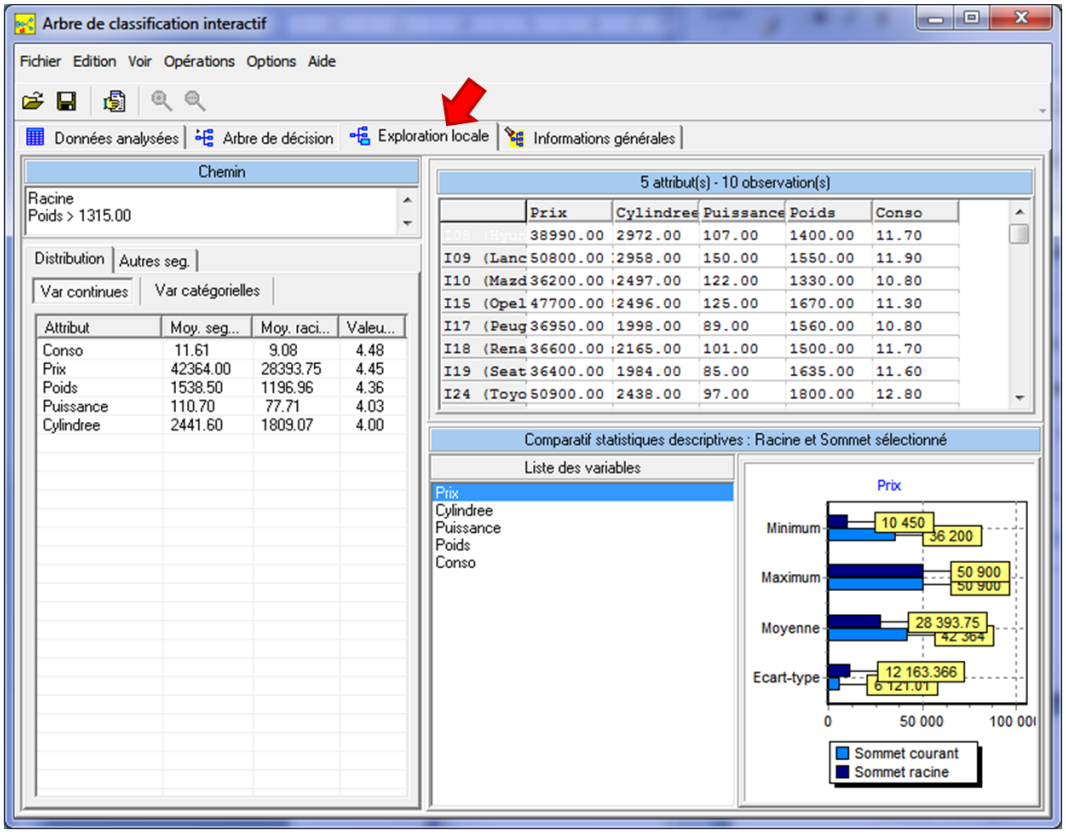
Nous y observons que la moyenne de « conso » dans la totalité de la population (échantillon) est de **9.08**. Dans cette sous-population (groupe), elle passe à **11.61**. L'écart est significatif avec une valeur test égale à **4.48**.

Le gap positif est du même ordre pour l'ensemble des variables (dixit les valeurs-test). Ce qui n'est guère étonnant si l'on prend en compte les résultats de l'ACP où l'on avait constaté que toutes les variables étaient fortement positivement liées entre elles.

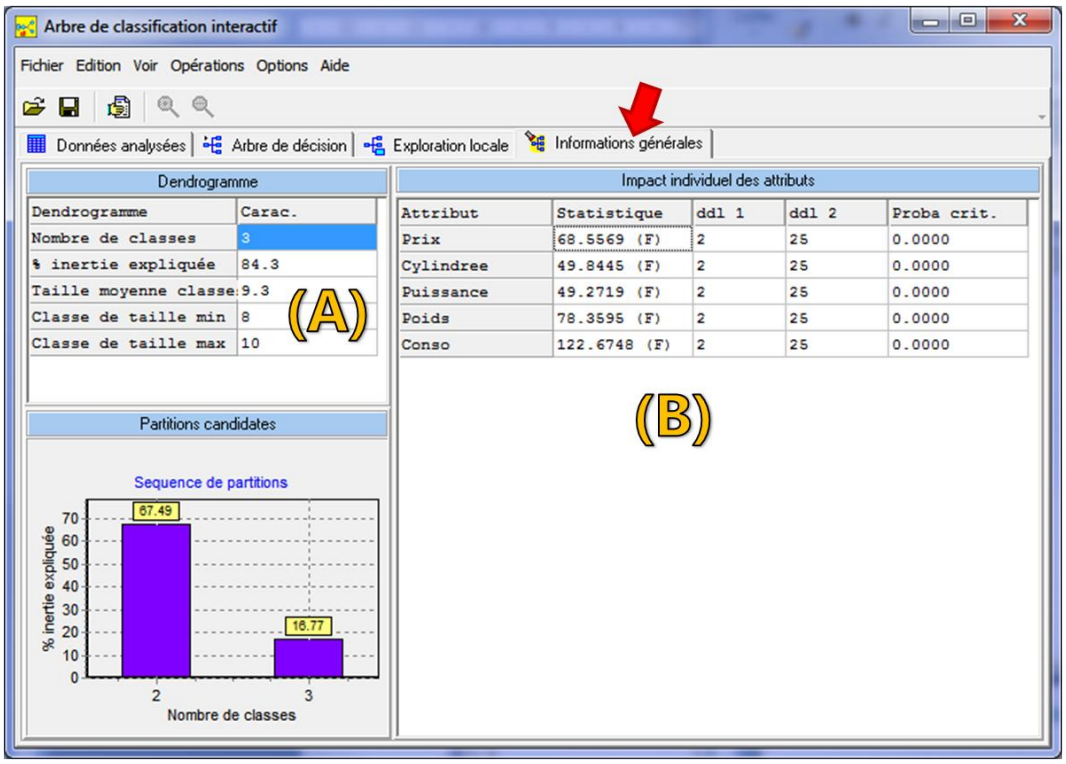
Les valeurs-test sont légèrement décalées par rapport à celles fournies par Tanagra (Figure 19). Cela provient du mode de calcul adopté. SPAD s'appuie sur la définition initiale de la valeur test c.-à-d. le quantile de la loi normale centrée réduite issue de la p-value du test de comparaison. La valeur est plus précise. Tanagra en revanche se contente de la statistique centrée et réduite[15]. Mais dans la mesure où l'approximation par la loi normale est de toute manière imprécise, c'est l'ordre de grandeur et surtout l'ordonnancement des valeurs qui importent. Ils sont bien identiques ici.

[15] http://tutoriels-data-mining.blogspot.fr/2008/04/interprter-la-valeur-test.html

Dans l'onglet « **Exploration locale** », nous disposons d'informations détaillées pour chaque sommet : la liste des observations couvertes, des informations plus précises concernant les statistiques conditionnelles et globales (min, max, écart-type en sus de la moyenne).



3.4.5 Evaluation globale de la partition



Dans l'onglet « **Informations générales** », nous observons les caractéristiques de la partition (A) : le nombre de classes, la part d'inertie expliquée, l'effectif moyen des classes, leurs tailles min et max. Dans « Impact individuel des attributs » (B), nous disposons des tests de caractérisation univariée. Toutes nos variables étant quantitatives, il s'agit d'un test ANOVA (F) – comparaison des moyennes conditionnelles (voir page 24). Comme nous le spécifions plus haut, il faut les consulter à titre indicatif lorsque les tests sont réalisés sur les variables actives.

3.5 Qualité de la segmentation – Comparaison avec la CAH

Une approche divisive monothétique implique des contraintes fortes dans la constitution des groupes. Nous pouvions craindre une moindre qualité de la partition au sens de l'inertie expliquée. Les expérimentations montrent qu'il n'en est rien (Rakotomalala et Le Nouvel, 2007). Par rapport à la CAH (classification ascendante hiérarchique) qui lui est comparable – la CAH propose des scénarios de solutions imbriquées – ses performances sont tout à fait honorables.

Nous avons voulu vérifier cela sur notre fichier « Autos ». Nous avons construit en partition en 3 classes avec l'outil CAH précédé d'une analyse en composantes principales.

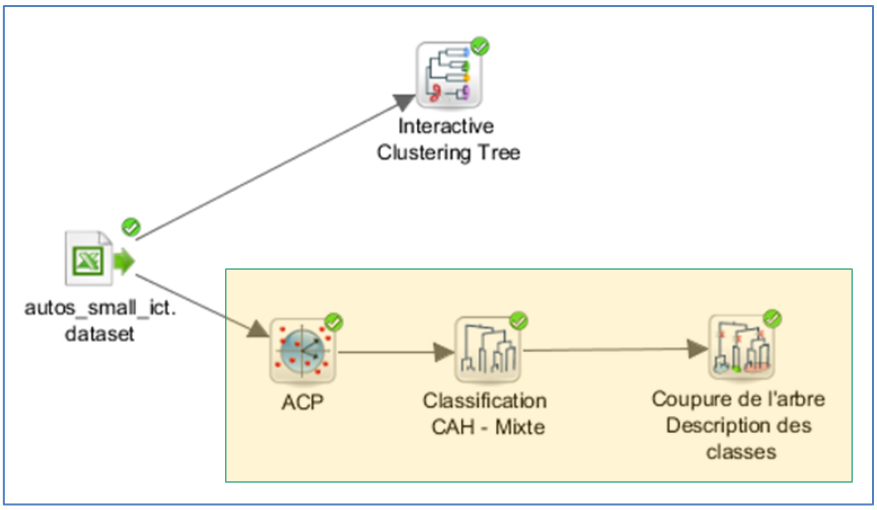


Figure 21 - CAH sous SPAD (Autos)

La part d'inertie expliquée par l'appartenance aux groupes est de 84.26%. Nous obtenons exactement la même partition. Mettons un petit bémol à cela quand même. Notre fichier est un peu « facile » avec des solutions évidentes. Il n'est pas très étonnant d'obtenir des résultats fortement convergents.

4 Arbres de classification avec Tanagra

Les arbres de classification sont disponibles dans Tanagra. Mais, à la différence de SPAD qui encapsule plusieurs opérations dans le composant ICT, nous devons décomposer les étapes : procéder explicitement à l'analyse factorielle préalable, sélectionner les facteurs à exploiter, placer l'arbre de classification en spécifiant les variables servant au calcul de l'inertie (les facteurs usuellement) et celles servant à la construction de la partition (les variables originelles a priori). Cette approche introduit en contrepartie une plus grande souplesse. Nous pouvons effectuer la classification sans passer par les facteurs notamment. En outre, Tanagra propose un dispositif inspiré de la méthodologie CART (construction en 2 phases : expansion + post-élagage) permettant d'obtenir des indications plus précises sur le bon nombre de classes.

4.1 Importation des données

Nous utilisons un autre jeu de données dans cette section. Il s'agit du fichier « **credit clustering.xlsx** », librement inspiré du fichier « credit.xls » disponible sur le site de cours de Marie Chavent (Université de Bordeaux)[16].

Dataset description

13 attribute(s)
468 example(s)

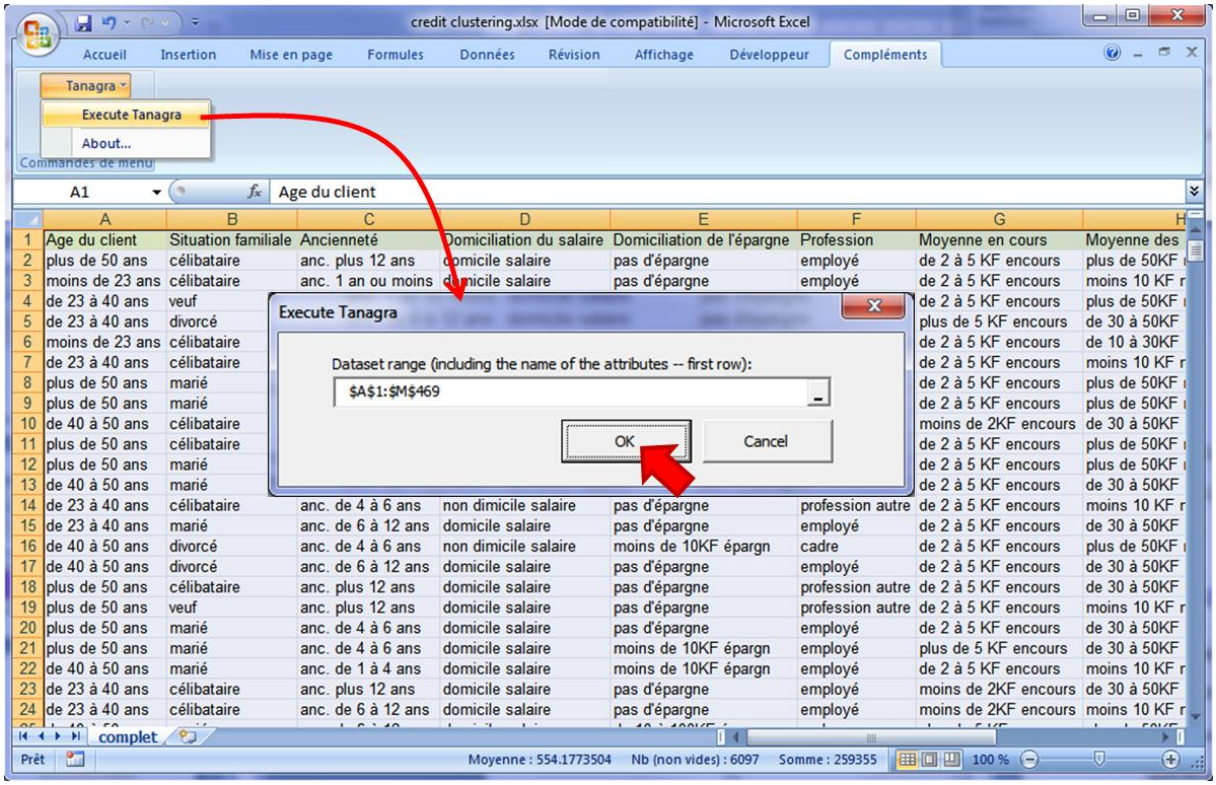
Attribute	Category	Informations
Age du client	Discrete	4 values
Situation familiale	Discrete	4 values
Ancienneté	Discrete	5 values
Domiciliation du salaire	Discrete	2 values
Domiciliation de l'épargne	Discrete	4 values
Profession	Discrete	3 values
Moyenne en cours	Discrete	3 values
Moyenne des mouvements	Discrete	4 values
Cumul des débits	Discrete	3 values
Autorisation de découvert	Discrete	2 values
Interdiction de chéquier	Discrete	2 values
Type de client	Discrete	2 values
score	Continue	-

Le fichier comporte 468 individus décrits par 13 variables : 9 correspondent à la signalétique ou le comportement du client, ce seront les variables actives ; 4 indiquent le jugement ou l'appréciation de la banque vis-à-vis du client (autorisation de découvert, interdiction de

[16] http://www.math.u-bordeaux1.fr/~machaven/teaching/

chéquier, type de client, score), ils constitueront les variables illustratives utilisées pour compléter la caractérisation des groupes. Toutes les variables sont qualitatives à l'exception de « score ». Nous utiliserons donc l'ACM (l'analyse factorielle des correspondances multiples) pour construire les facteurs à partir desquels seront calculées les inerties des classes.

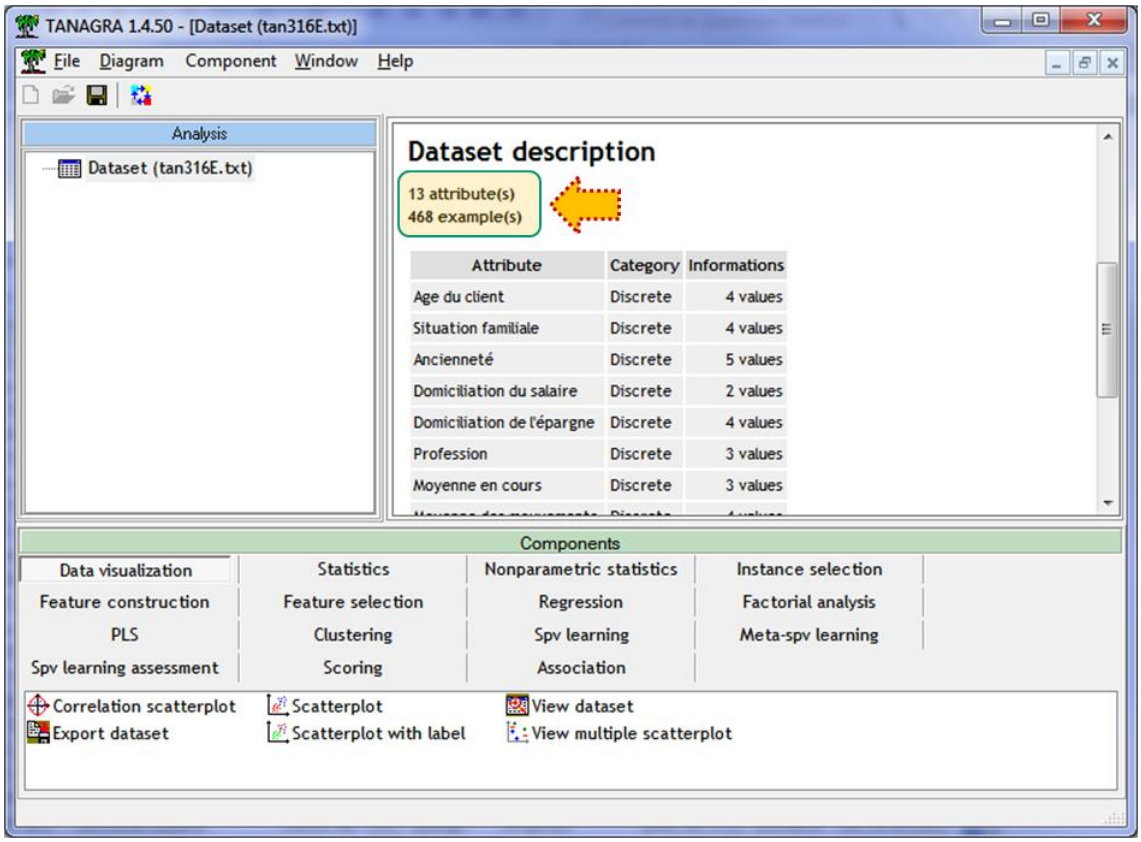
Nous chargeons le fichier dans le tableur Excel. Nous sélectionnons la plage de données et nous actionnons le menu TANAGRA / EXECUTE TANAGRA[17].



Nous vérifions les coordonnées de la plage puis nous cliquons sur le bouton OK.

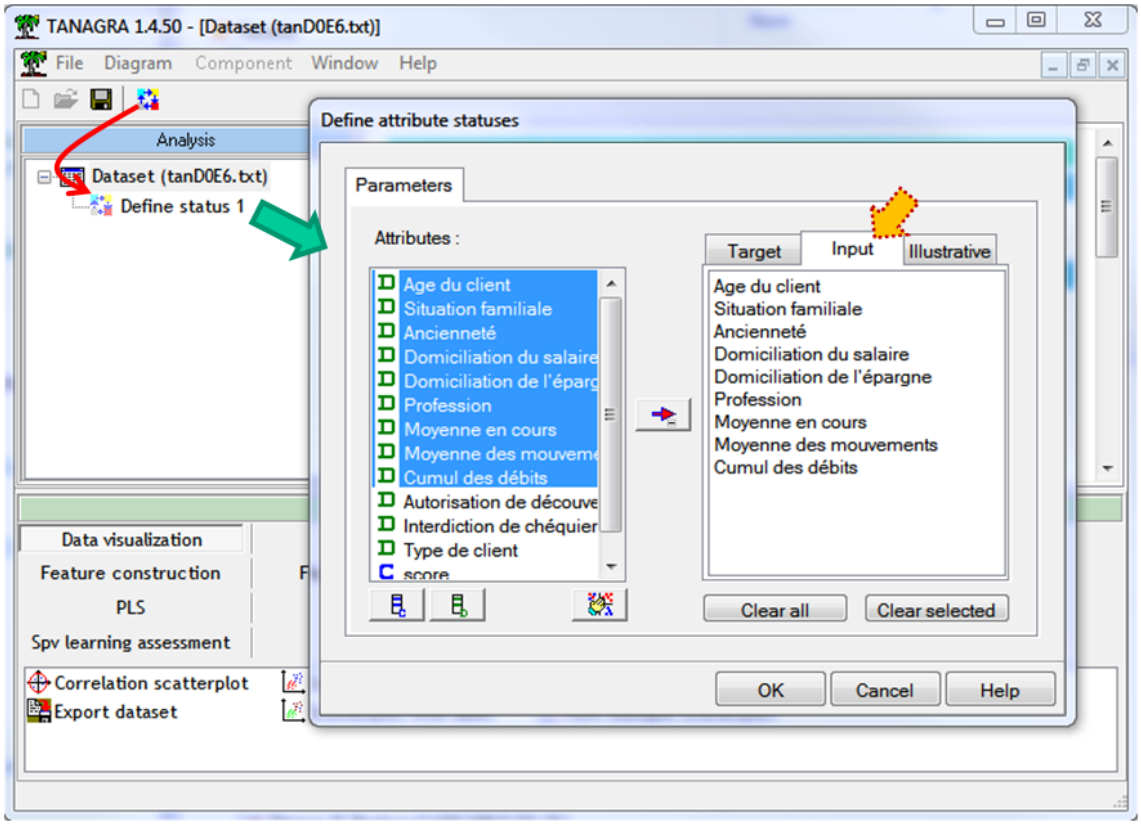
Tanagra est automatiquement démarré. Les données sont chargées. Une petite inspection est toujours utile à ce stade. Le logiciel a bien importé 13 variables et 468 observations (la première ligne correspond aux noms de variables).

[17] Installé à l'aide de la macro complémentaire « tanagra.xla ». Voir http://tutoriels-data-mining.blogspot.fr/2010/08/ladd-in-tanagra-pour-excel-2007-et-2010.html. Un add-on pour Libre Office et Open Office est également disponible, http://tutoriels-data-mining.blogspot.fr/2011/07/tanagra-addon-pour-openoffice-33.html



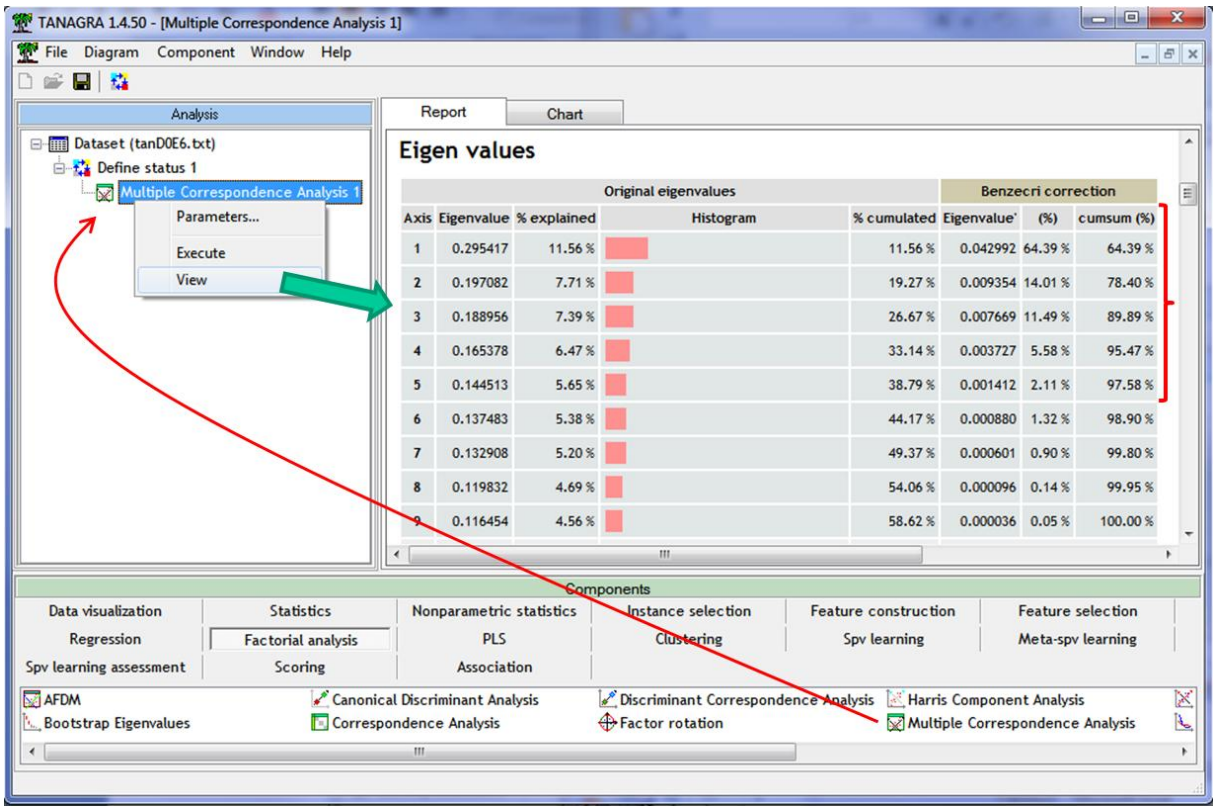
4.2 Analyse factorielle – ACM

Nous devons sélectionner les variables actives pour réaliser l'ACM. Nous utilisons le composant DEFINE STATUS accessible dans la barre d'outils.



Les variables « Age du client » … « Cumul des crédits » sont placées en INPUT.

Nous insérons ensuite le composant MULTIPLE CORRESPONDANCE ANALYSIS (onglet Factorial Analysis). Nous accédons aux résultats en cliquant sur le menu contextuel VIEW.

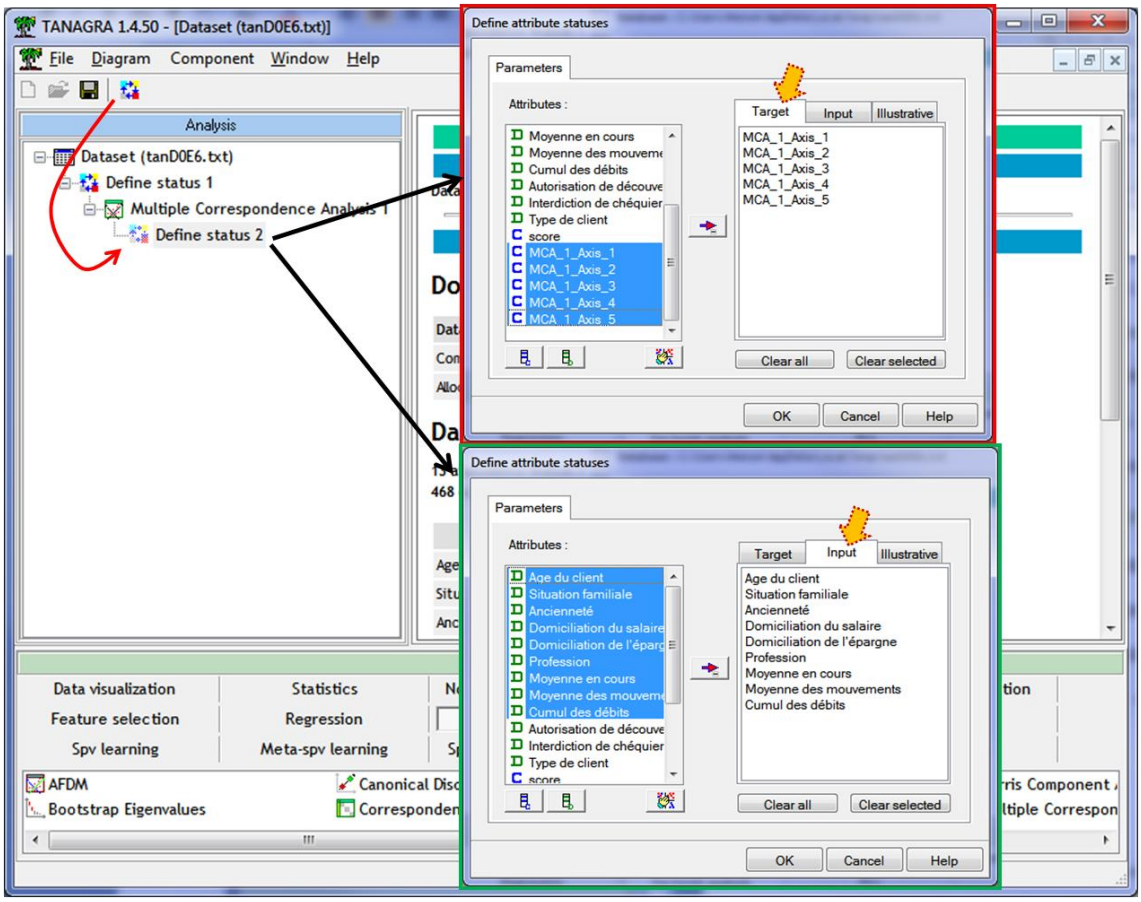


Au regard du critère du Kaiser c.-à-d. on retient les facteurs qui présentent une valeur propre supérieure à la moyenne (des valeurs propres), nous conserverions 9 axes. Mais après avoir appliqué la correction de Benzecri[18], on se rend compte que les 5 premiers axes sont amplement suffisants. Ces 5 facteurs sont automatiquement générés en sortie du composant et sont disponibles pour les calculs en aval.

4.3 Arbre de classification

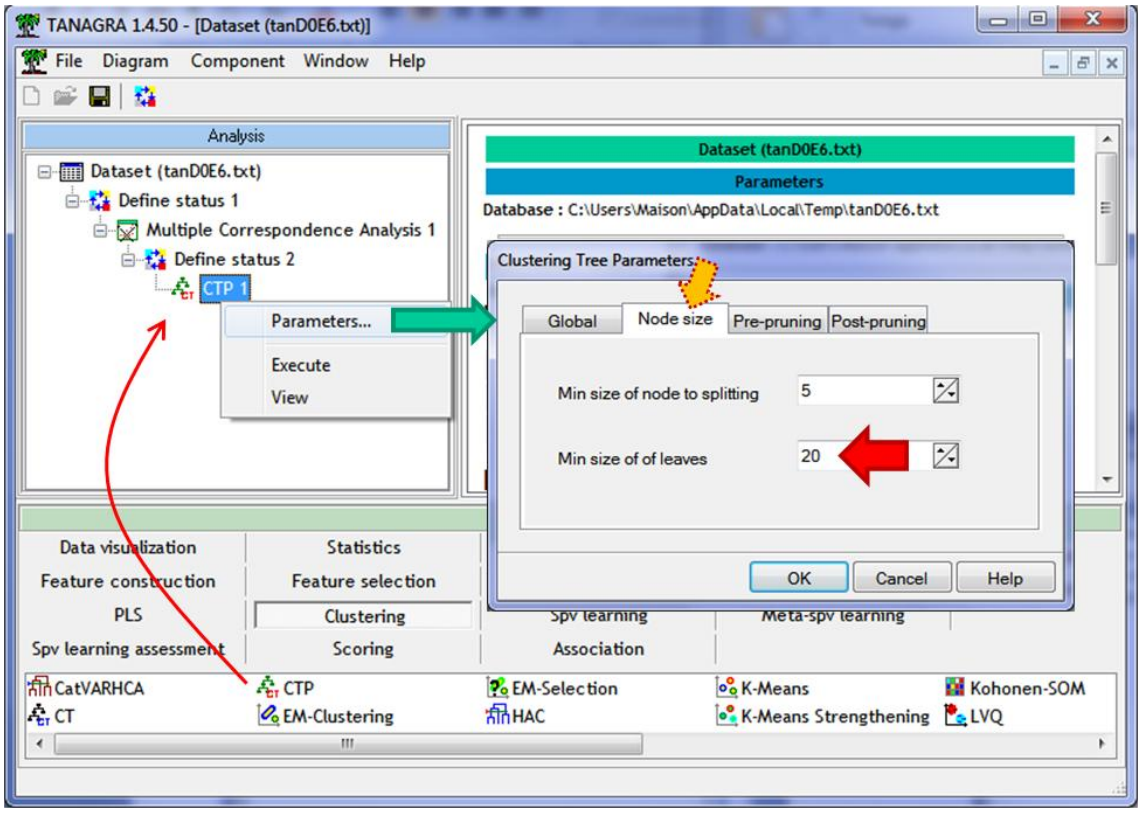
Nous devons spécifier les variables utilisées pour le calcul des inerties (les facteurs) et celles utilisées pour créer les groupes (les variables originales). Nous plaçons de nouveau le composant DEFINE STATUS dans le diagramme : en TARGET, nous plaçons les axes de l'ACM ; en INPUT, les variables « Age du client » … « Cumul des crédits ».

[18] Voir R. Rakotomalala, « Analyse des correspondances multiples – Diapos », août 2013 ; pages 26 à 29. http://tutoriels-data-mining.blogspot.fr/2013/08/analyse-des-correspondances-multiples.html



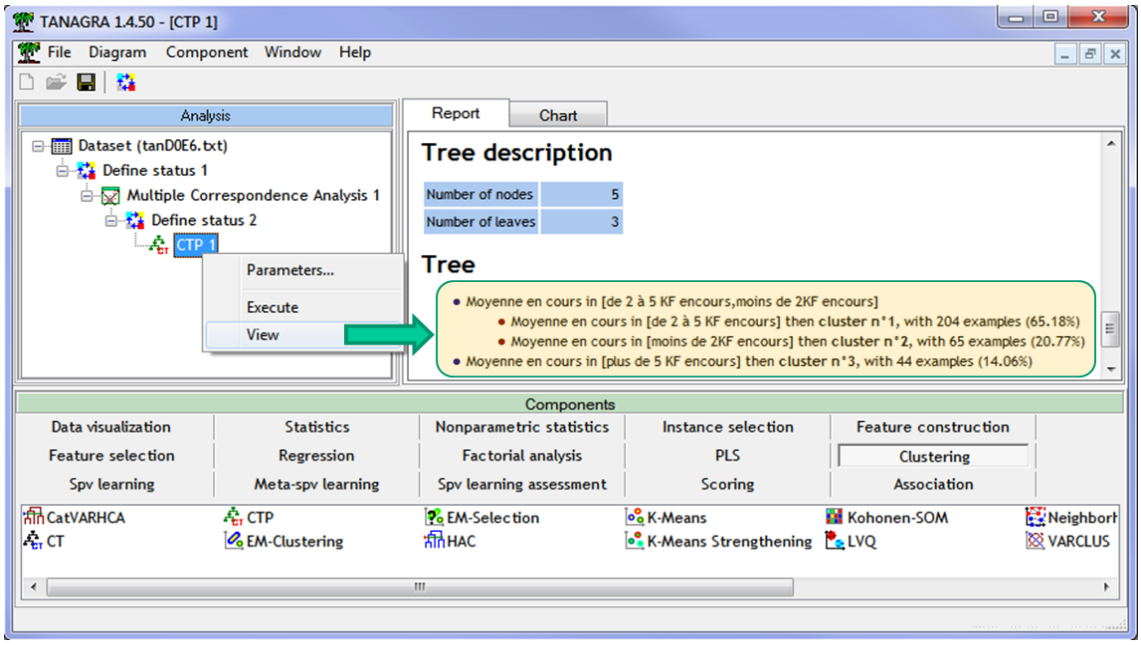
Puis nous plaçons le composant CTP «Clustering Tree with Post-pruning » (onglet Clustering).

Nous le paramétrons en actionnant le menu PARAMETERS.

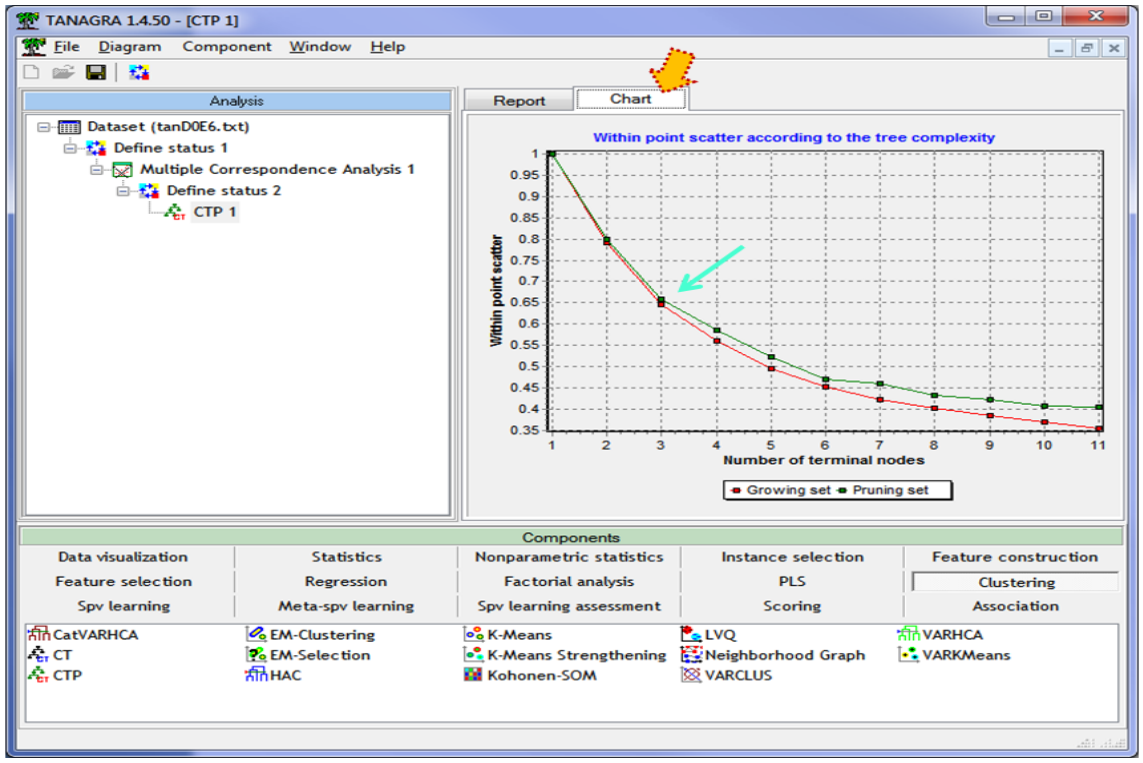


Nous ne souhaitons pas produire des feuilles (des groupes) avec moins de 20 individus.

Nous validons et nous cliquons sur VIEW. Voici l'arbre obtenu :



CTP propose une partition en 3 classes, reposant sur la variable « Moyenne en cours ».



Pour détecter le nombre de classes, CTP s'appuye sur la courbe de décroissance de l'inertie intra-classes de l'échantillon d'élagage (« pruning set », en vert) accessible dans l'onglet CHART de la fenêtre de visualisation. Il s'est attaché à chercher la « cassure » dans la courbe. Notons que la courbe peut remonter lorsque les groupes supplémentaires introduits ne sont pas

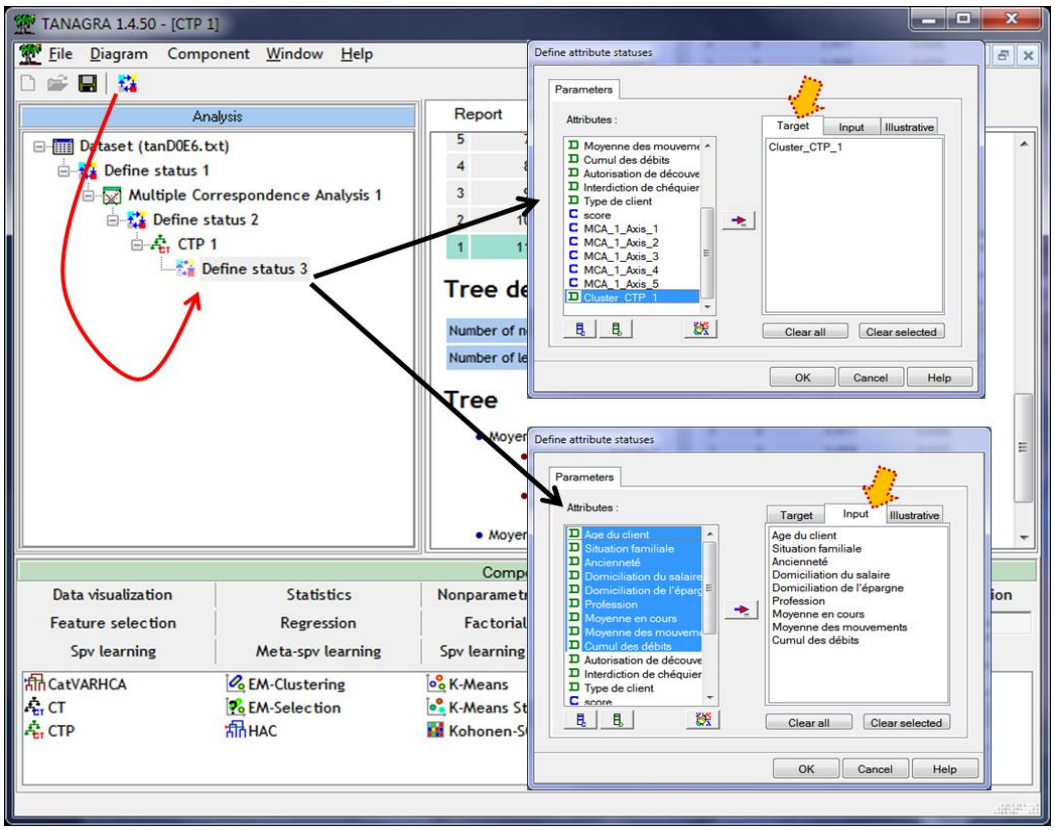
pertinents. La courbe calculée sur l'échantillon servant au calcul des segmentations (échantillon d'expansion, growing set en anglais – en rouge), elle, est toujours décroissante. Un décalage fort entre les deux courbes est annonciateur de surapprentissage.

4.4 Interprétation des groupes (variables actives)

Nous disposons des règles d'affectation :

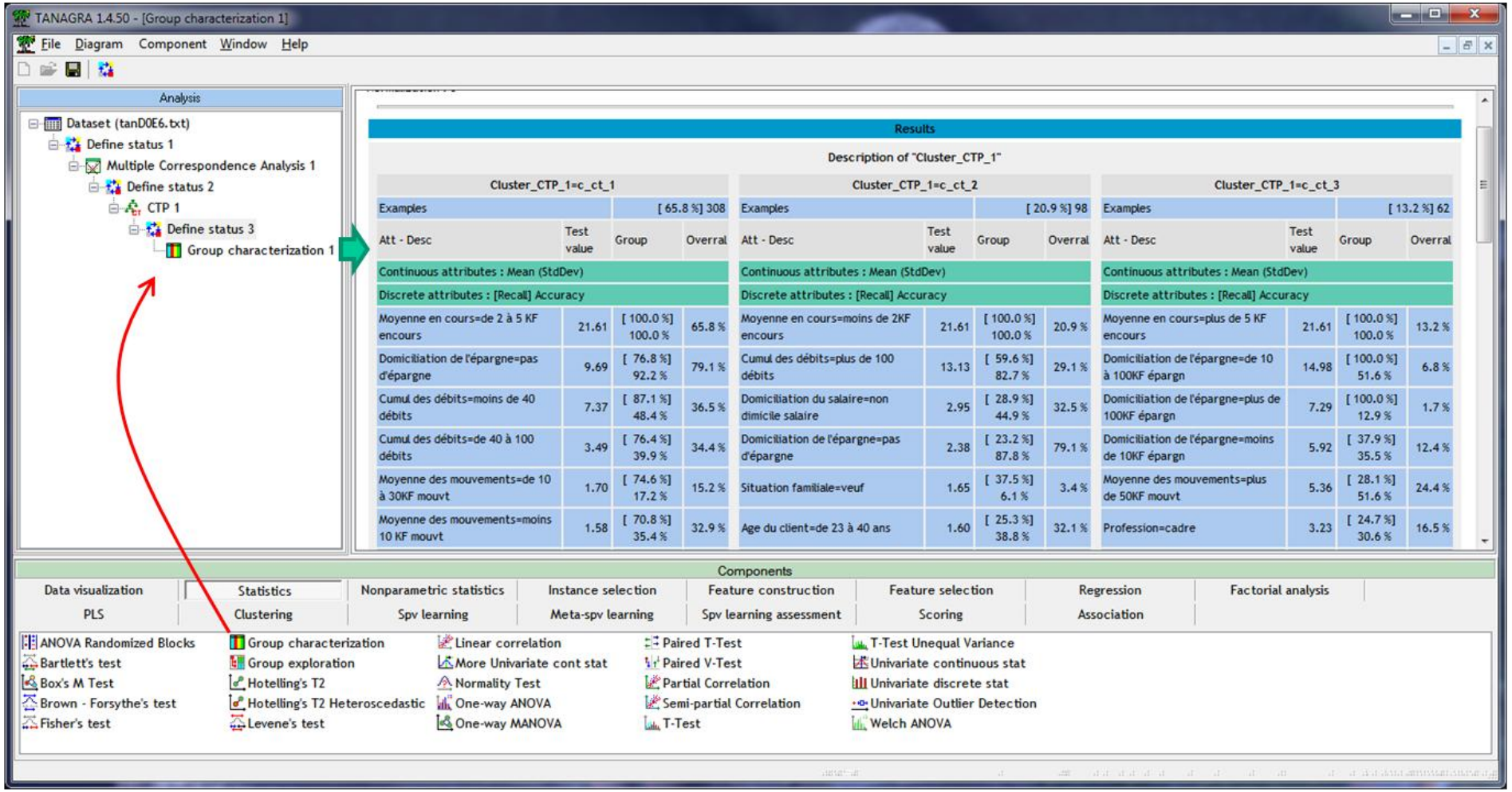
1. Si « moyenne en cours » in « 2 à 5 KF » alors **cluster n°1** (les lambdas)
2. Si « moyenne en cours » in « moins de 2 KF » alors **cluster n°2** (les dépourvus)
3. Si « moyenne en cours » in « plus de 5 KF » alors **cluster n°3** (les aisés)

Elles permettent d'associer un individu à un groupe. Elles nous donnent une indication sur la nature des groupes. Elles ne disent rien en revanche sur les caractéristiques des individus se rapportant aux autres variables. Nous allons utiliser le composant GROUP CHARACTERIZATION pour obtenir les descriptions conditionnelles.



Nous devons dans un premier temps indiquer les variables à utiliser pour la caractérisation. Nous introduisons le composant DEFINE STATUS. Nous plaçons la variable indiquant l'appartenance aux clusters généré par CTP en TARGET, les variables actives en INPUT.

Puis nous insérons l'outil GROUP CHARACTERIZATION (onglet STATISTICS).


TANAGRA 1.4.50 - [Group characterization 1]
File Diagram Component Window Help
Analysis
Dataset (tanD0E6.txt)
Define status 1
Multiple Correspondence Analysis 1
Define status 2
CTP 1
Define status 3
Group characterization 1
Results
Description of "Cluster_CTP_1"
Cluster_CTP_1=c_ct_1
Examples [65.8 %] 308
Att - Desc | Test value | Group | Overral
Continuous attributes : Mean (StdDev)
Discrete attributes : [Recall] Accuracy
Moyenne en cours=de 2 à 5 KF encours | 21.61 | [100.0 %] 100.0 % | 65.8 %
Domiciliation de l'épargne=pas d'épargne | 9.69 | [76.8 %] 92.2 % | 79.1 %
Cumul des débits=moins de 40 débits | 7.37 | [87.1 %] 48.4 % | 36.5 %
Cumul des débits=de 40 à 100 débits | 3.49 | [76.4 %] 39.9 % | 34.4 %
Moyenne des mouvements=de 10 à 30KF mouvt | 1.70 | [74.6 %] 17.2 % | 15.2 %
Moyenne des mouvements=moins 10 KF mouvt | 1.58 | [70.8 %] 35.4 % | 32.9 %
Cluster_CTP_1=c_ct_2
Examples [20.9 %] 98
Att - Desc | Test value | Group | Overral
Continuous attributes : Mean (StdDev)
Discrete attributes : [Recall] Accuracy
Moyenne en cours=moins de 2KF encours | 21.61 | [100.0 %] 100.0 % | 20.9 %
Cumul des débits=plus de 100 débits | 13.13 | [59.6 %] 82.7 % | 29.1 %
Domiciliation du salaire=non dimicile salaire | 2.95 | [28.9 %] 44.9 % | 32.5 %
Domiciliation de l'épargne=pas d'épargne | 2.38 | [23.2 %] 87.8 % | 79.1 %
Situation familiale=veuf | 1.65 | [37.5 %] 6.1 % | 3.4 %
Age du client=de 23 à 40 ans | 1.60 | [25.3 %] 38.8 % | 32.1 %
Cluster_CTP_1=c_ct_3
Examples [13.2 %] 62
Att - Desc | Test value | Group | Overral
Continuous attributes : Mean (StdDev)
Discrete attributes : [Recall] Accuracy
Moyenne en cours=plus de 5 KF encours | 21.61 | [100.0 %] 100.0 % | 13.2 %
Domiciliation de l'épargne=de 10 à 100KF épargn | 14.98 | [100.0 %] 51.6 % | 6.8 %
Domiciliation de l'épargne=plus de 100KF épargn | 7.29 | [100.0 %] 12.9 % | 1.7 %
Domiciliation de l'épargne=moins de 10KF épargn | 5.92 | [37.9 %] 35.5 % | 12.4 %
Moyenne des mouvements=plus de 50KF mouvt | 5.36 | [28.1 %] 51.6 % | 24.4 %
Profession=cadre | 3.23 | [24.7 %] 30.6 % | 16.5 %
Components
Data visualization | Statistics | Nonparametric statistics | Instance selection | Feature construction | Feature selection | Regression | Factorial analysis
PLS | Clustering | Spv learning | Meta-spv learning | Spv learning assessment | Scoring | Association
ANOVA Randomized Blocks
Bartlett's test
Box's M Test
Brown - Forsythe's test
Fisher's test
Group characterization
Group exploration
Hotelling's T2
Hotelling's T2 Heteroscedastic
Levene's test
Linear correlation
More Univariate cont stat
Normality Test
One-way ANOVA
One-way MANOVA
Paired T-Test
Paired V-Test
Partial Correlation
Semi-partial Correlation
T-Test
T-Test Unequal Variance
Univariate continuous stat
Univariate discrete stat
Univariate Outlier Detection
Welch ANOVA

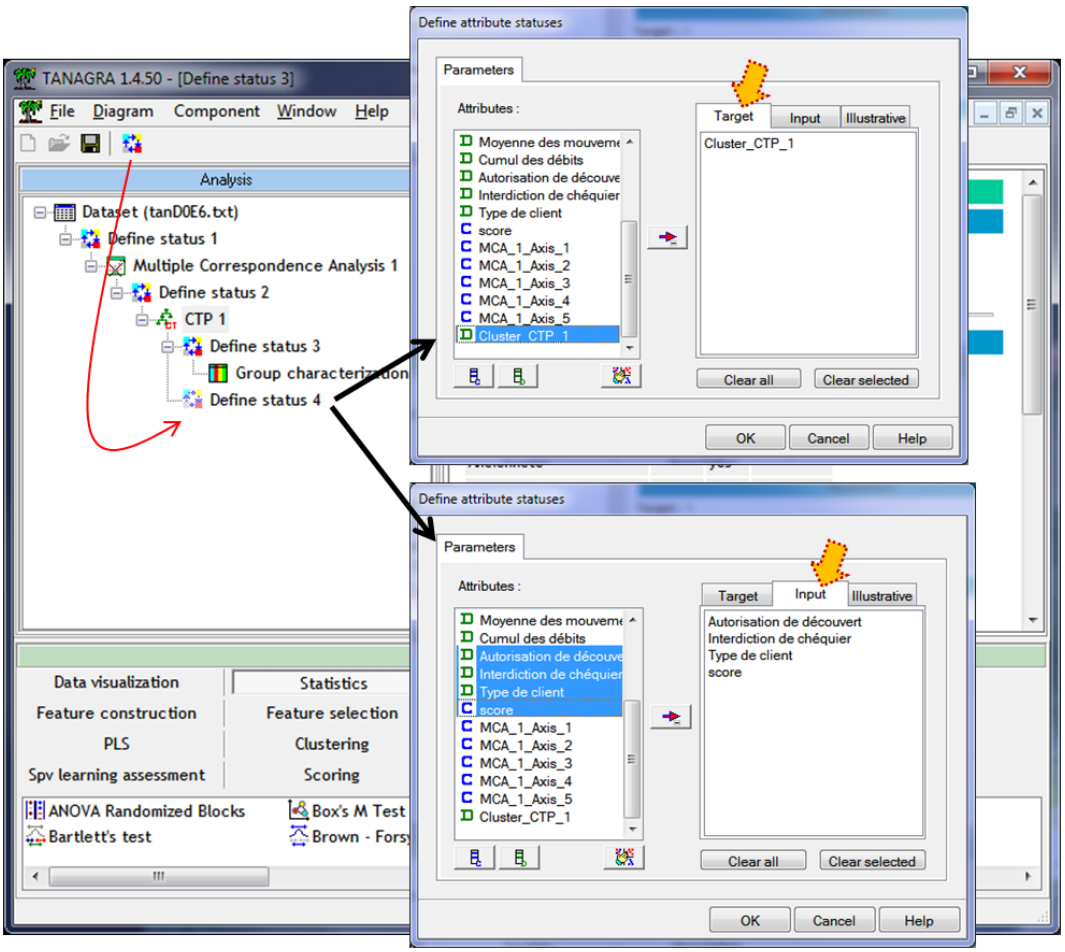
La domiciliation de l'épargne, le cumul des débits et la domiciliation des salaires définissent les principaux traits marquants des groupes.

1. Les « lambdas » ne domicilient pas leur épargne et ne cumulent pas (trop) les débits.
2. Les « dépourvus » croulent sous les débits, ne domicilient ni leur épargne ni leur salaire.
3. Les « aisés », en plus d'avoir des encours, ont également de l'épargne. On s'aperçoit qu'il y a une surreprésentation des cadres (30.6% vs. 16.5% dans la population globale). Non visible dans la copie d'écran, d'autres caractéristiques comme la domiciliation des salaires et l'ancienneté (6 à 12 ans) leurs sont également associées.

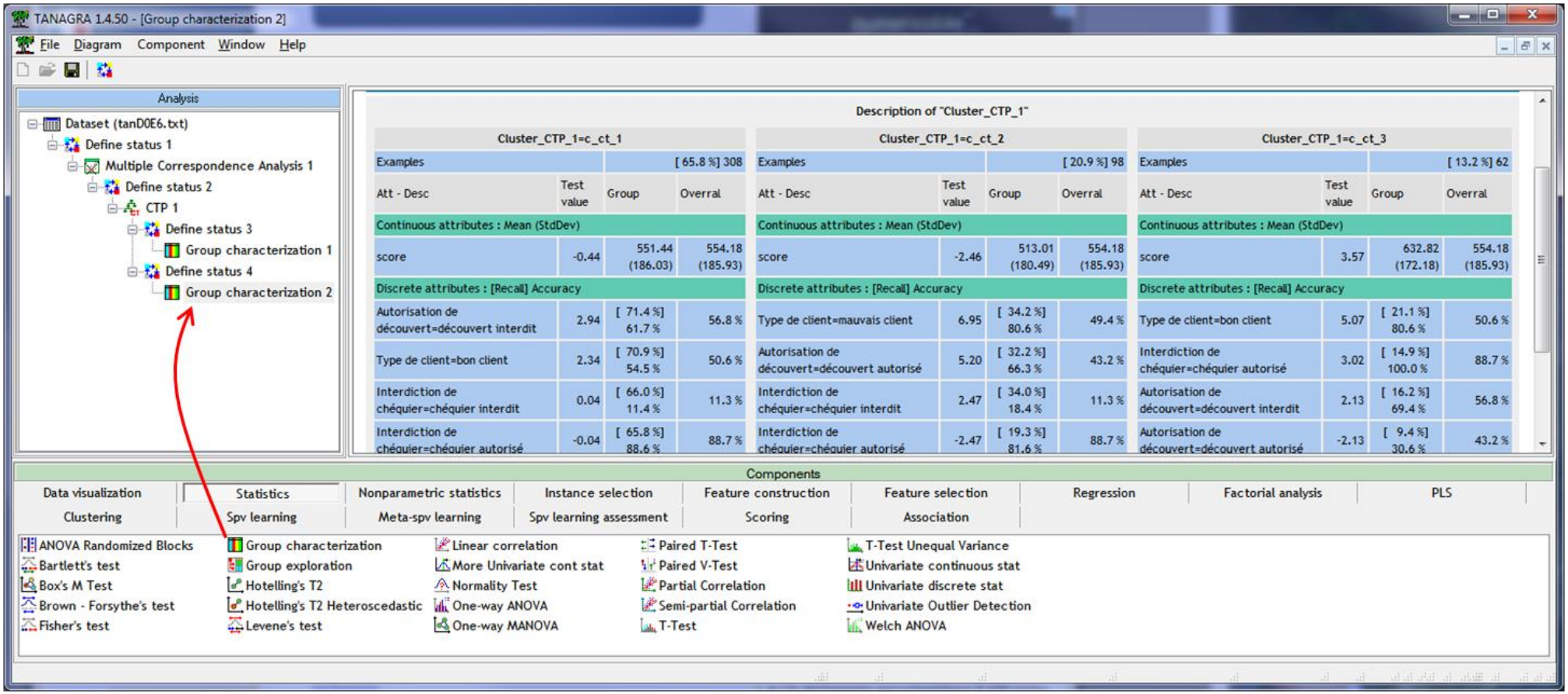
4.5 Interprétation des groupes (variables illustratives)

Voyons ce qu'il en est de la description des groupes à l'aide des variables illustratives qui indiquent, *grosso modo*, l'appréciation de l'établissement par rapport au client.

Nous introduisons de nouveau l'outil DEFINE STATUS, nous plaçons les clusters en TARGET, et les variables « Autorisation de découvert » à « Score » en INPUT.



Voyons ce qu'il en est de la caractérisation.

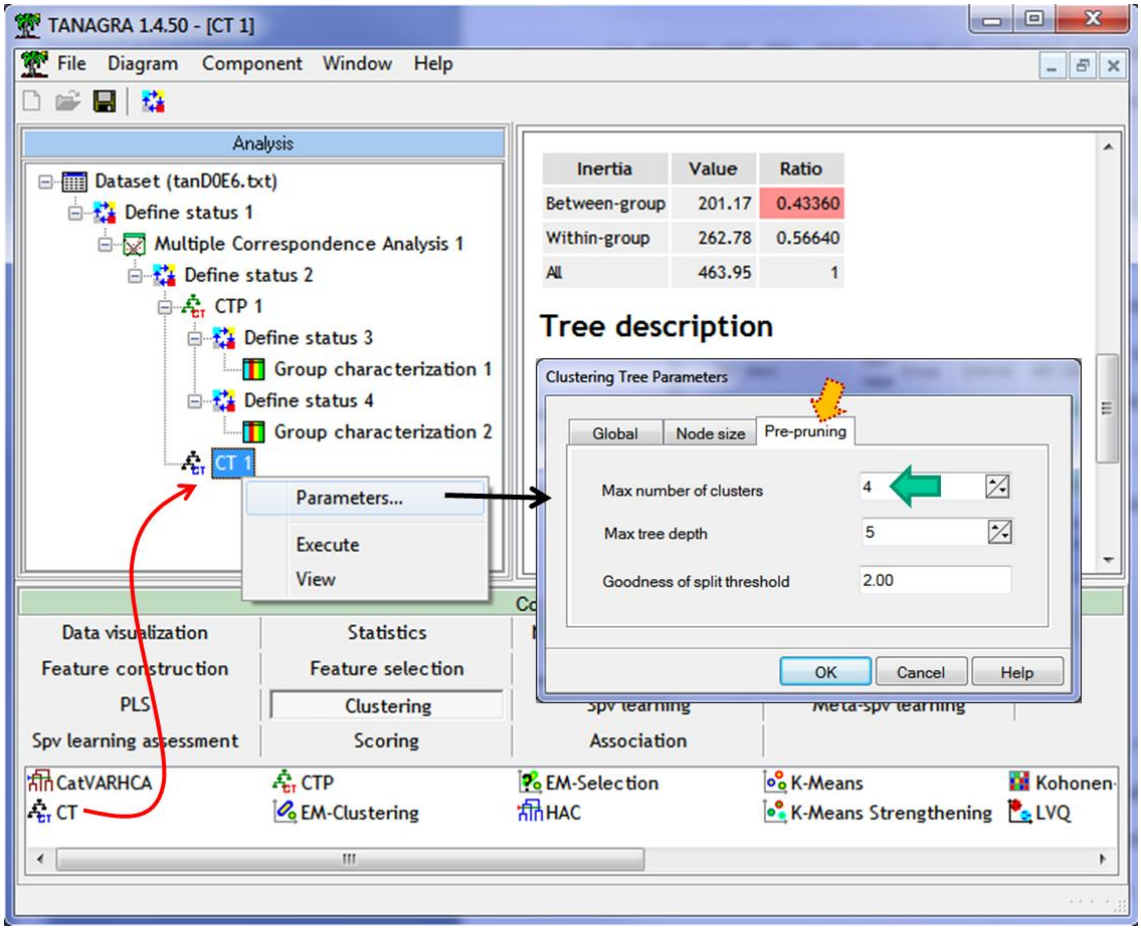

TANAGRA 1.4.50 - [Group characterization 2]
File Diagram Component Window Help
Analysis
Dataset (tanD0E6.txt)
Define status 1
Multiple Correspondence Analysis 1
Define status 2
CTP 1
Define status 3
Group characterization 1
Define status 4
Group characterization 2
Description of "Cluster_CTP_1"
Cluster_CTP_1=c_ct_1
Examples [65.8 %] 308
Att - Desc | Test value | Group | Overral
Continuous attributes : Mean (StdDev)
score | -0.44 | 551.44 (186.03) | 554.18 (185.93)
Discrete attributes : [Recall] Accuracy
Autorisation de découvert=découvert interdit | 2.94 | [71.4 %] 61.7 % | 56.8 %
Type de client=bon client | 2.34 | [70.9 %] 54.5 % | 50.6 %
Interdiction de chéquier=chéquier interdit | 0.04 | [66.0 %] 11.4 % | 11.3 %
Interdiction de chéquier=chéquier autorisé | -0.04 | [65.8 %] 88.6 % | 88.7 %
Cluster_CTP_1=c_ct_2
Examples [20.9 %] 98
Att - Desc | Test value | Group | Overral
Continuous attributes : Mean (StdDev)
score | -2.46 | 513.01 (180.49) | 554.18 (185.93)
Discrete attributes : [Recall] Accuracy
Type de client=mauvais client | 6.95 | [34.2 %] 80.6 % | 49.4 %
Autorisation de découvert=découvert autorisé | 5.20 | [32.2 %] 66.3 % | 43.2 %
Interdiction de chéquier=chéquier interdit | 2.47 | [34.0 %] 18.4 % | 11.3 %
Interdiction de chéquier=chéquier autorisé | -2.47 | [19.3 %] 81.6 % | 88.7 %
Cluster_CTP_1=c_ct_3
Examples [13.2 %] 62
Att - Desc | Test value | Group | Overral
Continuous attributes : Mean (StdDev)
score | 3.57 | 632.82 (172.18) | 554.18 (185.93)
Discrete attributes : [Recall] Accuracy
Type de client=bon client | 5.07 | [21.1 %] 80.6 % | 50.6 %
Interdiction de chéquier=chéquier autorisé | 3.02 | [14.9 %] 100.0 % | 88.7 %
Autorisation de découvert=découvert interdit | 2.13 | [16.2 %] 69.4 % | 56.8 %
Autorisation de découvert=découvert autorisé | -2.13 | [9.4 %] 30.6 % | 43.2 %
Components
Data visualization | Statistics | Nonparametric statistics | Instance selection | Feature construction | Feature selection | Regression | Factorial analysis | PLS
Clustering | Spv learning | Meta-spv learning | Spv learning assessment | Scoring | Association
ANOVA Randomized Blocks
Bartlett's test
Box's M Test
Brown - Forsythe's test
Fisher's test
Group characterization
Group exploration
Hotelling's T2
Hotelling's T2 Heteroscedastic
Levene's test
Linear correlation
More Univariate cont stat
Normality Test
One-way ANOVA
One-way MANOVA
Paired T-Test
Paired V-Test
Partial Correlation
Semi-partial Correlation
T-Test
T-Test Unequal Variance
Univariate continuous stat
Univariate discrete stat
Univariate Outlier Detection
Welch ANOVA

La messe est dite, mais pouvait-on vraiment en douter. La banque adore les « aisés » (score élevé, qualifiés de « bons clients »), un peu moins les « dépourvus » (score faible, qualifiés de mauvais clients, interdits de chéquier).

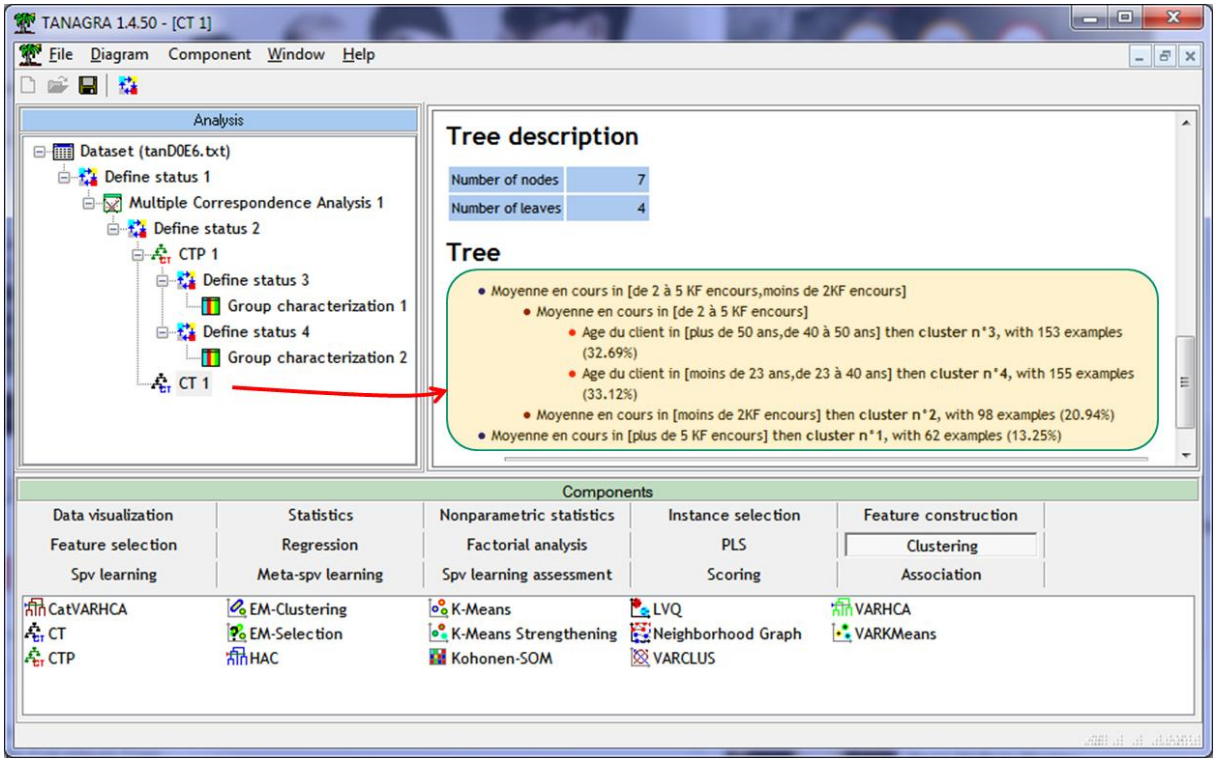
4.6 Spécification explicite du nombre de groupes

CTP essaie de détecter le bon nombre de groupes en s'appuyant sur une heuristique. Dans certains cas, nous souhaitons préciser explicitement le nombre de groupes à produire, notamment parce que nous voulons explorer d'autres scénarios. Il est plus judicieux dans ce cas d'utiliser le composant CT (clustering tree). Il exploite la totalité des données lors du processus de segmentation et nous pouvons spécifier le nombre de groupes à obtenir.

Mettons que nous désirons une partition en 4 classes maintenant. Nous serons alors dans la configuration suivante :



Nous obtenons une partition en 4 classes : les « lambdas » ont été subdivisés en 2 sous-groupes selon l'âge du client.



5 Arbres de classification sous R – Package « party »

Qu'il n'existe pas de package permettant de construire des arbres de classification sous R me paraissait extraordinaire. En cherchant un peu, j'ai (re)découvert le package « party »[19] qui permet de créer des arbres prédictifs avec une variable réponse multivariée c.-à-d. composée de plusieurs variables. Nous allons exploiter cette fonctionnalité pour construire un arbre de classification. Nous nous consacrons aux aspects opérationnels dans cette section. Pour les lecteurs désireux d'approfondir les aspects théoriques, je conseille l'article des auteurs du package (Horthorn et al., 2006). Voici le code R utilisé.

```
#chargement des données
library(xlsx)
don <- read.xlsx("autos_small_ict.xlsx",header=T,sheetIndex=1)
rownames(don) <- as.character(don[,1])
don <- don[-1]
colnames(don) <- c("Prix","Cylindree","Puissance","Poids","Conso")
print(head(don))


#fonction de centrage réduction
CR <- function(x){
  n <- length(x)
  m <- mean(x)
```

[19] http://cran.r-project.org/web/packages/party/party.pdf

```
  et <- sqrt((n-1)/n*var(x))
  return((x-m)/et)
}

#centrage réduction des variables
don.cr <- as.data.frame(lapply(don,CR))
print(head(don.cr))

#chargement de party
library(party)

#construction de l'arbre
arbre <- ctree(Prix+Cylindree+Puissance+Poids+Conso ~
don$Prix+don$Cylindree+don$Puissance+don$Poids+don$Conso, data =
don.cr,controls =
ctree_control(minsplit=10,minbucket=5,mincriterion=0.8))

#affichage
print(arbre)
```

Dans **ctree()**, les variables centrées réduites font office de variables cibles (placées devant le ~ dans la formule). Elles sont utilisées pour caractériser la pureté des sommets. Les variables originales sont utilisées comme variable de segmentation (placées après le ~). Comme pour tout algorithme d'induction d'arbre, plusieurs paramètres permettent de contrôler la conformation du modèle obtenu : « minsplit » est l'effectif nécessaire pour segmenter un sommet, « minbucket » est la taille des feuilles issues d'une segmentation, « mincriterion » est à comparer à « 1 – p-value » du test de significativité utilisé pour valider chaque segmentation.

Nous obtenons l'arbre suivant – avec **une partition en 4 classes** – à la sortie :

```
  Conditional inference tree with 4 terminal nodes

Responses: Prix, Cylindree, Puissance, Poids, Conso
Inputs:  don$Prix, don$Cylindree, don$Puissance, don$Poids, don$Conso
Number of observations:  28

1) don$Puissance <= 74; criterion = 1, statistic = 27
  2) don$Conso <= 6.8; criterion = 0.888, statistic = 13
    3)*  weights = 7
  2) don$Conso > 6.8
    4)*  weights = 7
1) don$Puissance > 74
  5) don$Puissance <= 106; criterion = 0.888, statistic = 13
    6)*  weights = 9
  5) don$Puissance > 106
    7)*  weights = 5
```

Figure 22 - Arbre "party" - Fichier "Autos"

Arbre qui, reproduit sous SPAD, prendrait la forme suivante :

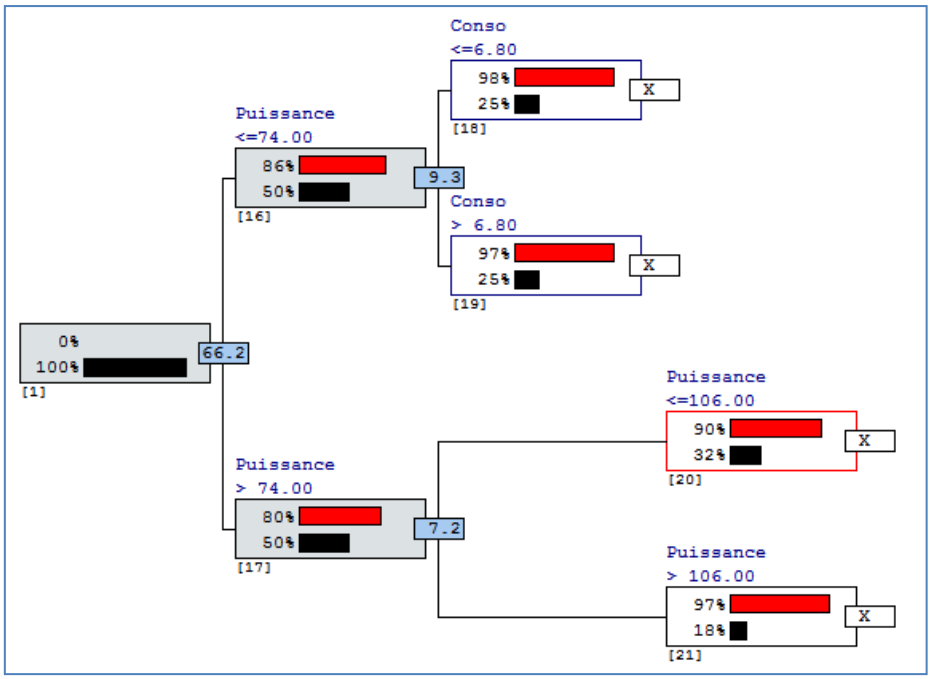


Figure 23 - Arbre "party" reproduit sous SPAD (autos)

Avec une part d'inertie expliquée de 82.6%.

6 Conclusion

Dans de nombreux domaines, l'interprétation est au moins aussi importante que la performance brute. Pouvoir expliquer un résultat permet de le valider, de le faire comprendre et d'emporter l'adhésion des décideurs. Les arbres de classification répondent parfaitement à cette spécification. La lecture des résultats est immédiate, elle est accessible à tout un chacun, y compris aux personnes totalement étrangères aux techniques de fouilles de données. Le déploiement, l'exploitation industrielle, se résume à produire les règles logiques d'appartenance aux groupes, leur exportation dans les systèmes d'information est extrêmement simple. Autre aspect important, sa complexité de calcul est bien maîtrisée, elle est identique aux arbres de décision classiques, nous pouvons mettre en œuvre la technique sur des grandes bases comportant des centaines de milliers d'observations. Enfin, les arbres de classification intègrent également une caractéristique qui a déjà largement fait la popularité des arbres de décision dans le domaine de la prédiction depuis une dizaine d'années, elle offre à l'expert du domaine la possibilité d'intervenir dans le processus d'exploration des connaissances.

Si la technique a été essentiellement définie pour la classification, nous pouvons l'étendre naturellement à la prédiction multi-supervisée. L'idée est de produire des groupes homogènes

selon une série de descripteurs à partir d'autres variables. On peut souhaiter par exemple vouloir expliquer le comportement d'achat de clients sur plusieurs produits à partir de leurs informations signalétiques. Le rôle des « variables illustratives » – couramment utilisées dans l'analyse typologique – est élargi et renforcé. Nous en reparlerons dans un prochain tutoriel.

7 Bibliographie

1. Blockeel H., De Raedt L., Ramon J. , « Top-Down Induction of Clustering Trees », ICML, 55—63, 1998.
2. Breiman L., Friedman J., Olshen R., Stone C., « CART : classification and regression trees », Wadsworth International, 1984.
3. Chavent M., « A monothetic clustering method », Pattern Recognition Letters, 19, 989—996, 1998.
4. Hothorn T., Hornik K., Zeileis A., « Unbiased recursive partitionning : A conditional inference framework », Journal of Computational and Graphical Statistics, 15(3), pp. 651-674, 2006.
5. Lebart L., Morineau A., Piron M., « Statistique Exploratoire Multidimensionnelle », Dunod, 3ème édition, 2000.
6. Light R., Margolin B., « An analysis of variance for categorical data », JASA, vol. 66, n°335, pp. 534-544, 1971.
7. Morineau A., "Note sur la Caractérisation Statistique d'une Classe et les Valeurs-tests", Bulletin Technique du Centre de Statistique et d'Informatique Appliquées, Vol 2, no 1-2, p 20-27, 1984.
8. Rakotomalala, R., « Arbres de décision », Revue Modulad, n°33, 2005.
9. Rakotomalala R., Le Nouvel T., "Interactive Clustering Tree : Une méthode de classification descendante adaptée aux grands ensembles de données", in "Data Mining et apprentissage statistique : application en assurance, banque et marketing", Revue RNTI-A-1, Editions CEPADUES, pp.75-94, 2007.