**UNIVERSITÉ NATIONALE DU VIETNAM À HANOÏ**

**INSTITUT DE LA FRANCOPHONIE INTERNATIONALE**



MODULE FOUILLE DE DONNEES

Option : Systèmes Intelligents et Multimédia (SIM)

*Promotion:* XXIII

RAPPORT TRAVAUX PRATIQUE I

# Jeux de données d'évaluation de performance des étudiants

**Rédigé par:**
ADOUM Okim Boka

**Encadrant:**
Dr. Nguyen Thi Minh Huyen

Année académique : 2018 - 2019

# Contents

# List of Figures

# 1 Introduction générale

Dans le cadre du module de fouille de données, l'analyse des données est une famille de méthodes statistiques dont les principales caractéristiques sont d'être multidimensionnelles, descriptives et prédictives. Dans bon nombre de cas certaines méthodes aident à faire ressortir les relations pouvant exister entre les différentes données et à en tirer une information statistique qui permet de décrire de façon plus succincte les principales informations contenues dans ces données. Il y'a des techniques qui regroupe les données de façon à faire apparaître clairement ce qui les rend homogènes. L'objectif est de nous imprégner des conceptes fondamentaux. Pour cela, un travail pratique nous a été soumis. Ce travail consiste à choisir un jeux de données que nous devrions explorer, faire une l'analyse prédictive, considérée comme un type d'exploration de données. Le cœur de l'analyse prédictive se fonde sur la capture des relations entre les variables explicatives et les variables expliquées ou prédites, issues des occurrences passées, et l'exploitation de ces relations pour prédire les résultats futurs. C'est aussi possible de réaliser cette dernière sur un certain nombre d'échantillon pour produire automatiquement des règles à partir d'une base de données contenant des exemples validés, d'où l'apprentissage supervisé
Pour mener à bien cette tâche, nous avons comme jeux de données **Student-Por.xls** choisit sur le site "https://archive.ics.uci.edu/ml/datasets/student+performance", et utilisé un logiciel gratuit pour explorer les données "TANAGRA".

## 2 Énoncé du problème

L'objectif de ce jeux de données est d'arriver à prédire la performance des élèves dans l'enseignement secondaire ( cas des lycées portugaises). Pour cela, elle est produite en fonction de propriétés actifs qui entrent dans le processus direct de la formation d'un élève et qui ont des impact plus ou moins considérable dans son résultat Ce jeu de donnée concerne la performance des élèves de [Cortez et Silva, 2008], dans deux matières distinctes :

- les mathématiques (mat);
- la langue portugaise (por).

Ces données ont été modélisé sous des tâches de classification et de régression.

## 3 Présentation de données

Cette base de données **STUDENT** concerne les résultats des élèves de deux écoles portugaises dans l'enseignement secondaire. Les attributs de données comprennent les notes des élèves, les caractéristiques démographiques, sociales et scolaires et ont été collectés à l'aide de rapports et de questionnaires. Deux jeux de données sont fournis concernant la performance dans deux matières distinctes: les mathématiques (mat) et la langue portugaise (por). Pour notre cas précis contentons-nous du portugais. Retenons que l'attribut cible G3 a une forte corrélation avec les attributs G2 et G1. Cela se produit parce que3 est la note de dernière année (attribuée à la 3ème période), alors que G1 et G2 correspondent aux notes de 1re et 2ème périodes. Il est plus difficile de prédire G3 sans G2 et G1. Le fichier **Student-Por.xls [1]** contient 649 élèves, 33 attributs. Ce jeu de donnée étudie la classification et la régression comme tâches associées. Ci-dessous le tableau récapitulatif:

**Résumé** : Prédisez les performances des élèves dans l'enseignement secondaire (lycée).

| | | | | | |
|---|---|---|---|---|---|
| **Caractéristiques du jeu de données:** | Multivarié | **Nombre d'instances:** | 649 | **Surface:** | Social |
| **Caractéristiques d'attribut:** | Entier | **Nombre d'attributs:** | 33 | **Date du don** | 2014-11-27 |
| **Tâches associées:** | Classification, régression | **Valeurs manquantes?** | N / A | **Nombre de visites sur le Web:** | 425362 |

*Figure 1: Présentation*

## 4 Description des attributs [1]

- **school** - école d'élèves (binaire : 'GP' - Gabriel Pereira ou 'MS' - Mousinho da Silveira)
- **sex** - sexe de l'élève (binaire : 'F' - femme ou 'M' - homme);
- **age** - âge de l'élève (numérique : de 15 à 22);
- **address** - type de domicile de l'élève (binaire : 'U' - urbain ou 'R '- rural);

- **famsize** - taille de la famille (binaire :' LE3 '- inférieur ou égal à 3 ou' GT3 '-supérieur à 3);
- **pstatus** - statut de cohabitation des parents (binaire :' T '- vivre ensemble ou' A'- à part);
- **medu** - éducation de la mère (numérique : 0 - aucun, 1 - enseignement primaire (4e année), 2 - 5e à 9e année, 3 - enseignement secondaire ou 4 - enseignement supérieur);
- **fedu** - éducation du père (numérique : 0 - aucun, 1 - enseignement primaire (4e année), 2 - 5e à 9e année, 3 - enseignement secondaire ou 4 - enseignement supérieur);
- **mjob** - travail de mère ( nominaux : « enseignant », « soins de santé », « services » civils (administration ou police), «at_home» ou «autre»);
- **fjob** - emploi du père (nom : «enseignant», «soins de santé», soins civils « services » (par exemple , administratif ou de la police), « à domicile » ou « autre »);
- **reason** - raison de choisir cette école (nominale : proche de la « maison », l' école « réputation », « bien sûr » préférence ou « autre »);
- **guardian** - tuteur de l'étudiant (nom : «mère», «père» ou «autre» );
- **traveltime** - temps de déplacement domicile-école (numérique : 1 - <15 min., 2- 15 à 30 min., 3 - 30 min. À 1 heure ou 4 -> 1 heure);
- **studytime** - temps d'étude hebdomadaire (numérique : 1 - <2 heures, 2 - 2 à 5 heures, 3 - 5 à 10 heures, ou 4 -> 10 heures);
- **failures** - nombre d'échecs antérieurs (numérique : n si 1 <= n <3, sinon 4);
- **schoolsup** - soutien éducatif supplémentaire (binaire : oui ou non);
- **famsup** - soutien éducatif familial (binaire : oui ou non);
- **paid** - cours payants supplémentaires dans le cours (mathématiques ou portugais) (binaire : oui ou non);
- **activités** - activités parascolaires (binaire : oui ou non);
- **nursery** - école maternelle fréquentée (binaire : oui ou non);
- **higher** - veut suivre des études supérieures (binaire : oui ou non);
- **internet** - Accès à Internet à la maison (binaire : oui ou non);
- **romantic** - avec une relation amoureuse (binaire : oui ou non);
- **famrel** - qualité des relations familiales (numérique : de 1 - très mauvais à 5 - excellent);
- **freetime** - libre après l 'école (numérique : de 1 - très faible à 5 - très élevé);

- **goout** - sortir avec des amis (numérique : de 1 - très faible à 5 - très élevé);
- **Dalc** - consommation journalière d'alcool (numérique : de 1 - très faible à 5 - très élevé);
- **Walc** - consommation d'alcool le week-end (numérique : de 1 - très faible à 5 - très élevé);
- **health** - état de santé actuel (numérique : de 1 - très mauvais à 5 - très bon);
- **absences** - nombre d'absences scolaires (numériques : de 0 à 93) ces notes sont liées au sujet du cours, Mathématiques ou Portugais:
- **G1** - première note (numérique : de 0 à 20);
- **G2** - note de la deuxième période (numérique : de 0 à 20);
- **G3** - note finale (numérique : de 0 à 20, cible de sortie).

# 5 Analyse Exploratoire

Cette section est divisé en deux parties:

1. Réduction du nombre de variables (dimension) : méthodes factorielles
2. Réduction du nombre d'individus en regroupant par classes : méthodes de classification

## 5.1 Méthode factorielle [4]

Les données sont classifiés en deux types qui sont les valeurs discrètes dont nous allons appliqués l'analyse de correspondance multiples et les valeurs continues pour l'analyse en composante principal.

### 5.1.1 Analyse composante principale [4]

Elle permet de résumer un tableau d'individus multiplié par les variables à l'aide d'un petit nombre de facteurs,de visualiser le positionnement des individus les uns par rapport aux autres,de visualiser les corrélations entre les variables et enfin d'interpréter les facteurs.
Pour initier une analyse, nous devons désigner les variables actives. Nous utilisons le composant DEFINE STATUS. Nous plaçons toutes les variables des données à valeur continue en INPUT

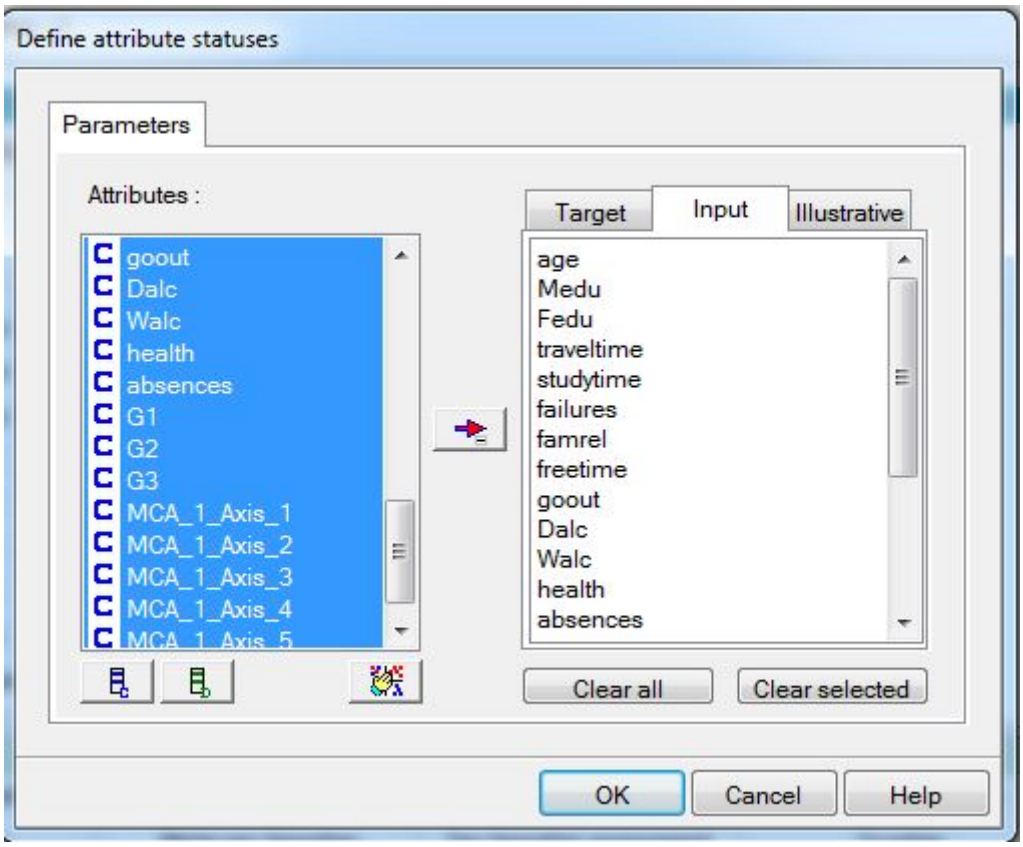



*Figure 2: Analyse composante principale*

Sur la figure ci_dessous nous remarquons que le niveau de corrélations ne dépasse pas la valeur 1 ce qui nous conduit à un tableau des valeurs propres montrant la proportion de variance reproduite sur les facteurs, individuellement (Proportion) et cumulativement (Cumulative).Et ceci est visible dans la partie Eigen value dans la fenêtre histogramme.

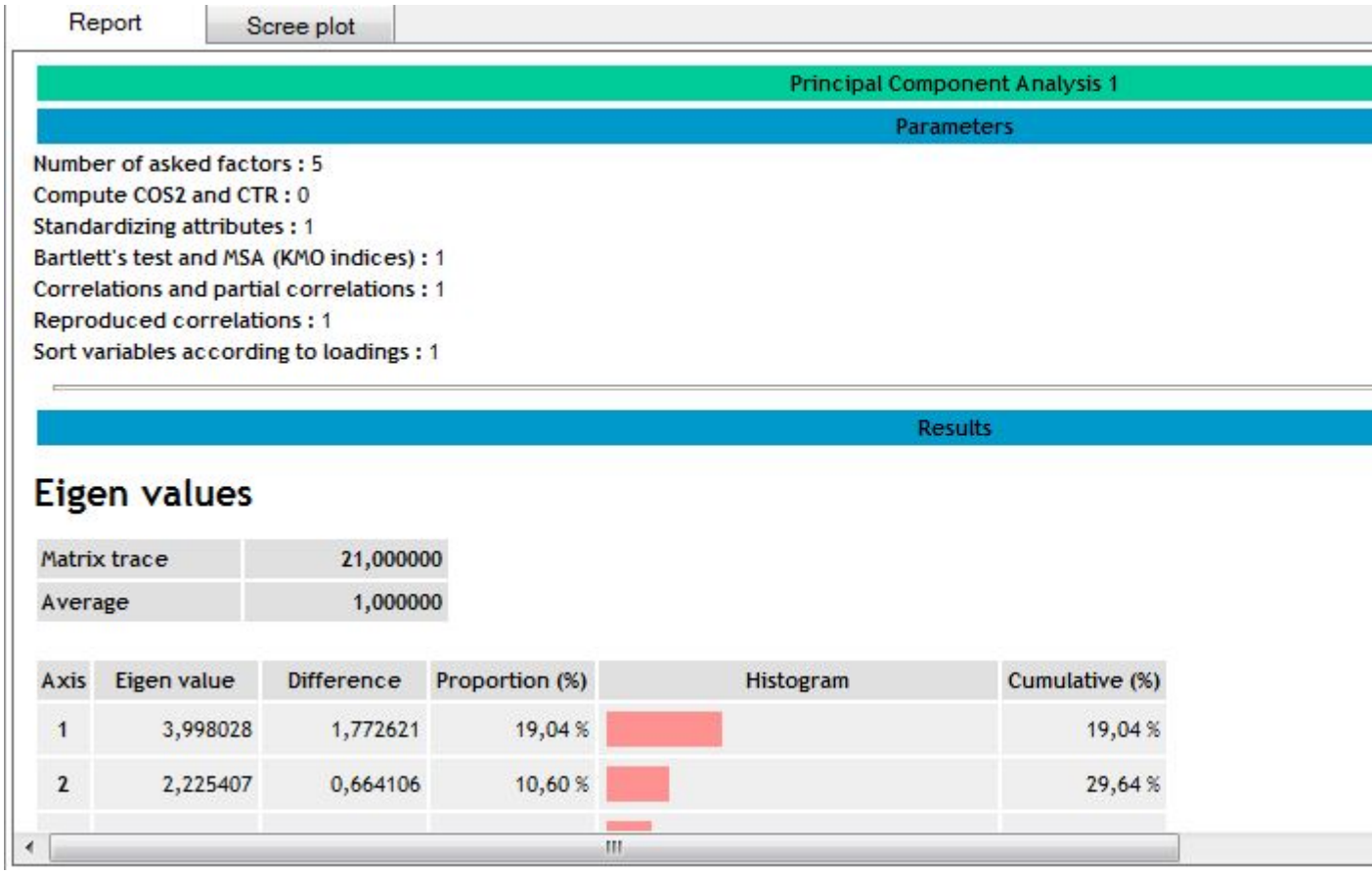



*Figure 3: Histogramme de valeur de Eigen*

Voyons plus en détail dans la fenêtre scree plot

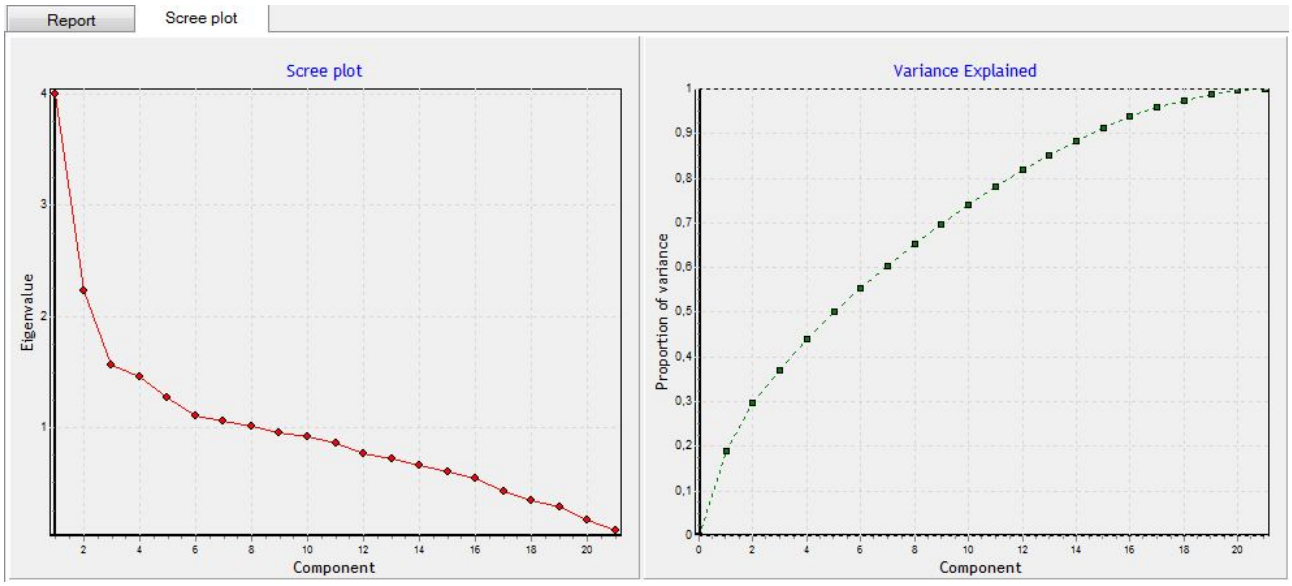



*Figure 4: la fenêtre scree plot*

En observant le graphe du coté Eigen Value, nous remarquons qu'entre 1 et 2 , une distance

importante ce qui prouve que la corrélation des données est limitée à partir de 1.
Les autres outils prouvent cette limite

Report | Scree plot

**Significance of Principal Components**

| Global critical values | |
|---|---|
| Kaiser-Guttman | 1 |
| Karlis-Saporta-Spinaki | 1,35136 |

**Eigenvalue table - Test for significance**

| Eigenvalues - Significance | | |
|---|---|---|
| Axis | Eigenvalue | Broken-stick critical values |
| 1 | 3,998028 | 3,645359 |
| 2 | 2,225407 | 2,645359 |
| 3 | 1,561301 | 2,145359 |
| 4 | 1,457738 | 1,812025 |
| 5 | 1,268721 | 1,562025 |
| 6 | 1,099841 | 1,362025 |
| 7 | 1,059965 | 1,195359 |
| 8 | 1,015211 | 1,052502 |
| 9 | 0,951033 | 0,927502 |

*Figure 5: importance des composants principal*

**Matrices**

Correlations

| | G2 | G3 | G1 | MCA_1_Axis_1 | failures | Medu | Walc | Dalc | Fedu | MCA_1_Axis_2 | absences | freetime | famre |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G2 | 1,00000 | 0,91855 | 0,86498 | -0,36516 | -0,38578 | 0,26404 | -0,16485 | -0,18948 | 0,22514 | 0,01868 | -0,12474 | -0,10668 | 0,0895 |
| G3 | 0,91855 | 1,00000 | 0,82639 | -0,35966 | -0,39332 | 0,24015 | -0,17662 | -0,20472 | 0,21180 | -0,01600 | -0,09138 | -0,12270 | 0,0633 |
| G1 | 0,86498 | 0,82639 | 1,00000 | -0,38399 | -0,38421 | 0,26047 | -0,15565 | -0,19517 | 0,21750 | 0,02390 | -0,14715 | -0,09450 | 0,0487 |
| MCA_1_Axis_1 | -0,36516 | -0,35966 | -0,38399 | 1,00000 | 0,24153 | -0,54417 | -0,02377 | 0,03775 | -0,42540 | 0,00000 | -0,03827 | 0,00612 | -0,069( |
| failures | -0,38578 | -0,39332 | -0,38421 | 0,24153 | 1,00000 | -0,17221 | 0,08227 | 0,10595 | -0,16592 | -0,08743 | 0,12278 | 0,10899 | -0,062( |
| Medu | 0,26404 | 0,24015 | 0,26047 | -0,54417 | -0,17221 | 1,00000 | -0,01977 | -0,00702 | 0,64748 | 0,11748 | -0,00858 | -0,01969 | 0,0244 |
| Walc | -0,16485 | -0,17662 | -0,15565 | -0,02377 | 0,08227 | -0,01977 | 1,00000 | 0,61656 | 0,03844 | 0,07849 | 0,15637 | 0,12024 | -0,093 |
| Dalc | -0,18948 | -0,20472 | -0,19517 | 0,03775 | 0,10595 | -0,00702 | 0,61656 | 1,00000 | 0,00006 | 0,03215 | 0,17295 | 0,10990 | -0,075 |
| Fedu | 0,22514 | 0,21180 | 0,21750 | -0,42540 | -0,16592 | 0,64748 | 0,03844 | 0,00006 | 1,00000 | 0,16896 | 0,02986 | 0,00684 | 0,0202 |
| MCA_1_Axis_2 | 0,01868 | -0,01600 | 0,02390 | 0,00000 | -0,08743 | 0,11748 | 0,07849 | 0,03215 | 0,16896 | 1,00000 | -0,22863 | 0,01739 | 0,0747 |
| absences | -0,12474 | -0,09138 | -0,14715 | -0,03827 | 0,12278 | -0,00858 | 0,15637 | 0,17295 | 0,02986 | -0,22863 | 1,00000 | -0,01872 | -0,089 |
| freetime | -0,10668 | -0,12270 | -0,09450 | 0,00612 | 0,10899 | -0,01969 | 0,12024 | 0,10990 | 0,00684 | 0,01739 | -0,01872 | 1,00000 | 0,1292 |
| famrel | 0,08959 | 0,06336 | 0,04879 | -0,06902 | -0,06265 | 0,02442 | -0,09351 | -0,07577 | 0,02026 | 0,07476 | -0,08953 | 0,12922 | 1,0000 |

*Figure 6: matrice de corrélation*

## 5.2 Méthode de classification

La classification (clustering) est une méthode mathématique d'analyse de données qui permet de faciliter l'étude d'une population d'effectif important (personnes, animaux, plantes, malades, gènes,...), dans l'optique de les regroupe en plusieurs classes de telle sorte que les individus d'une même classe soient le plus semblables possible et que les classes soient le plus distinctes possibles. Pour cela il y a plusieurs façons de procéder (qui peuvent conduire à des résultats différents...). Dans notre cas, nous allons utiliser la classification hiérarchique ascendante (CAH) et la méthode

K-MEANS. Ces deux deux méthodes nous permettront d'évaluer la performance des élèves en langue Portugaise.

### 5.2.1 La méthode K-MEANS

K-means est un algorithme non supervisé de clustering non hiérarchique. Il permet de regrouper en K clusters distincts les observations du data set. Ainsi les données similaires se retrouveront dans un même cluster. Par ailleurs, une observation ne peut se retrouver que dans un cluster à la fois (exclusivité d'appartenance). Une même observation, ne pourra donc, appartenir à deux clusters différents. Ci-dessous les étapes de l'application de la méthode K-MEANS sur notre dataset.

- Importer les données ;
- Réaliser quelques statistiques descriptives sur les variables actives ;
- Centrer et réduire les variables ;
- Réaliser la classification automatique via les K-Means sur les variables transformées, en décidant nous même du nombre de classes ;
- Visualiser les données avec la nouvelle colonne représentant la classe d'appartenance des individus ;
- Illustrer les classes à l'aide des variables actives, via des statistiques descriptives comparatives et/ou des graphiques judicieusement choisis ;
- Croiser la partition obtenue avec une variable catégorielle illustrative ;
- Exporter les données, avec la colonne additionnelle, dans un fichier.

L'étape 1 et 2 ont été déjà réalisés dans le document plus haut par conséquent nous allons entamer l'étape 3.

**Sélection des variables quantitatives sur la figure en dessous**

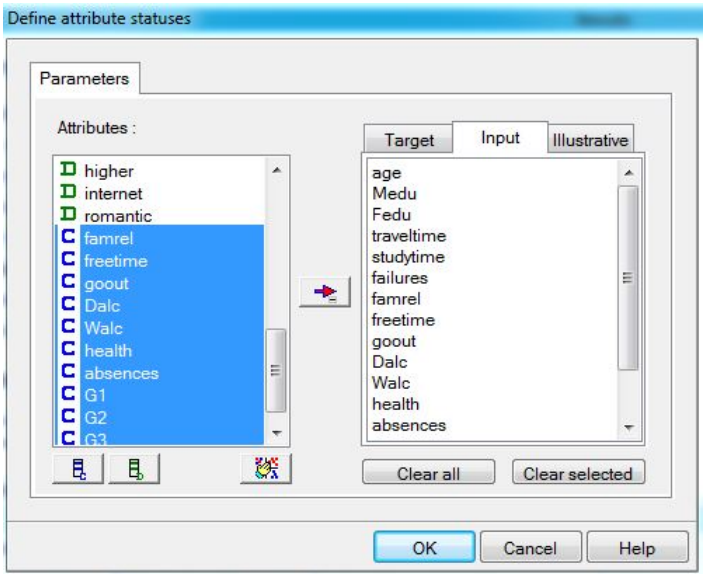



*Figure 7: sélection des variables quantitatives*

**Ajout de l'onglet MORE UNIVARIATE CONT STAT pour observer les statisques du jet de donnée**

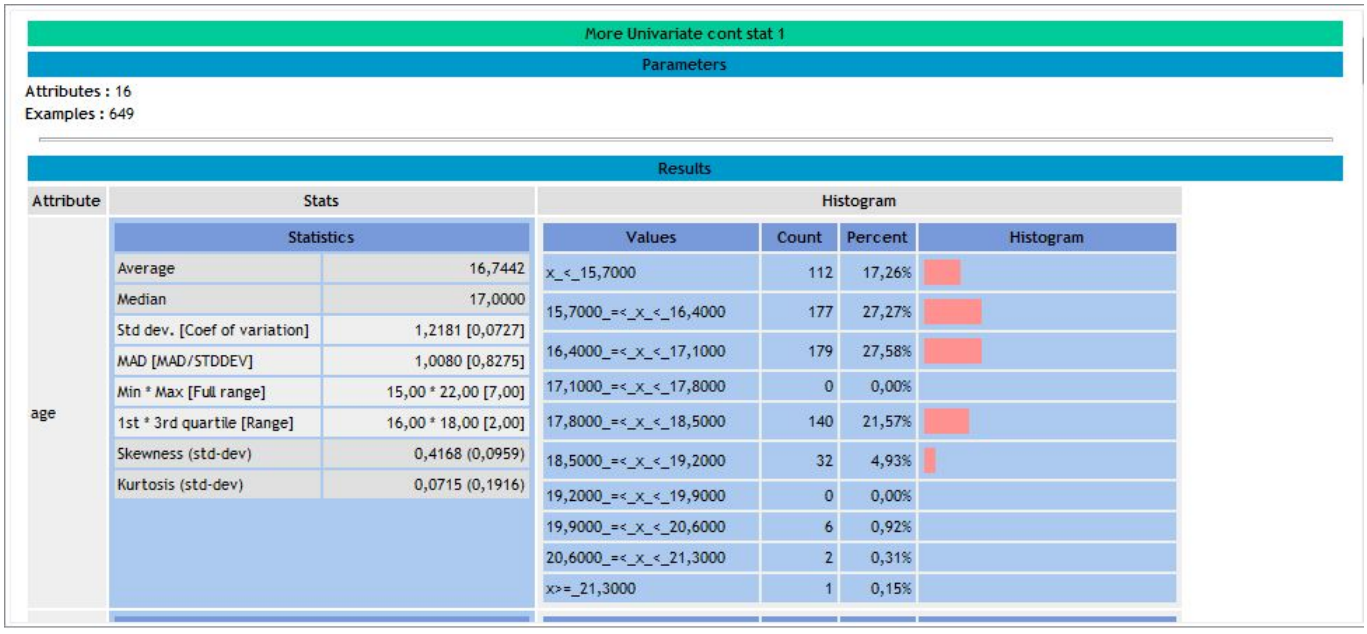



*Figure 8: L'onglet MORE UNIVARIATE CONT STAT*

**Nous allons centrer et réduire les variables avec le composant STANDARDIZE**

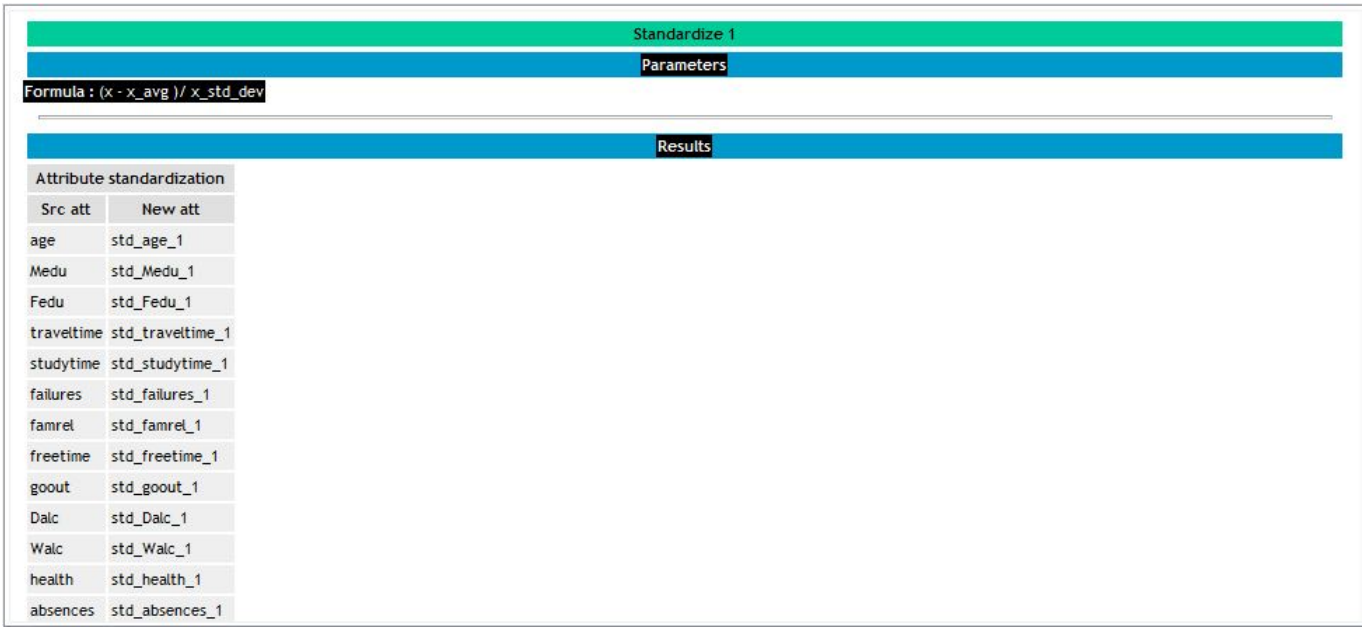



*Figure 9: composant STANDARDIZE*

**Nous allons transformer les variables pour pouvoir effectuer des calculs pour cela nous allons introduire un nouveau DEFINE STATUS, ensuite nous avons définis 3 groupes pour notre dataset**

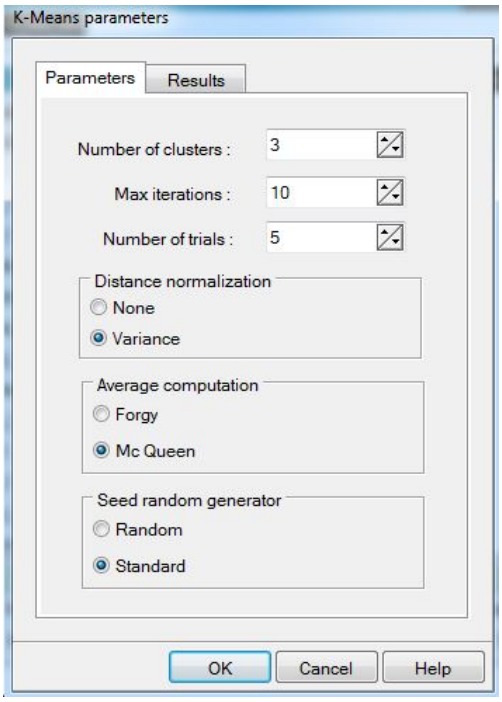
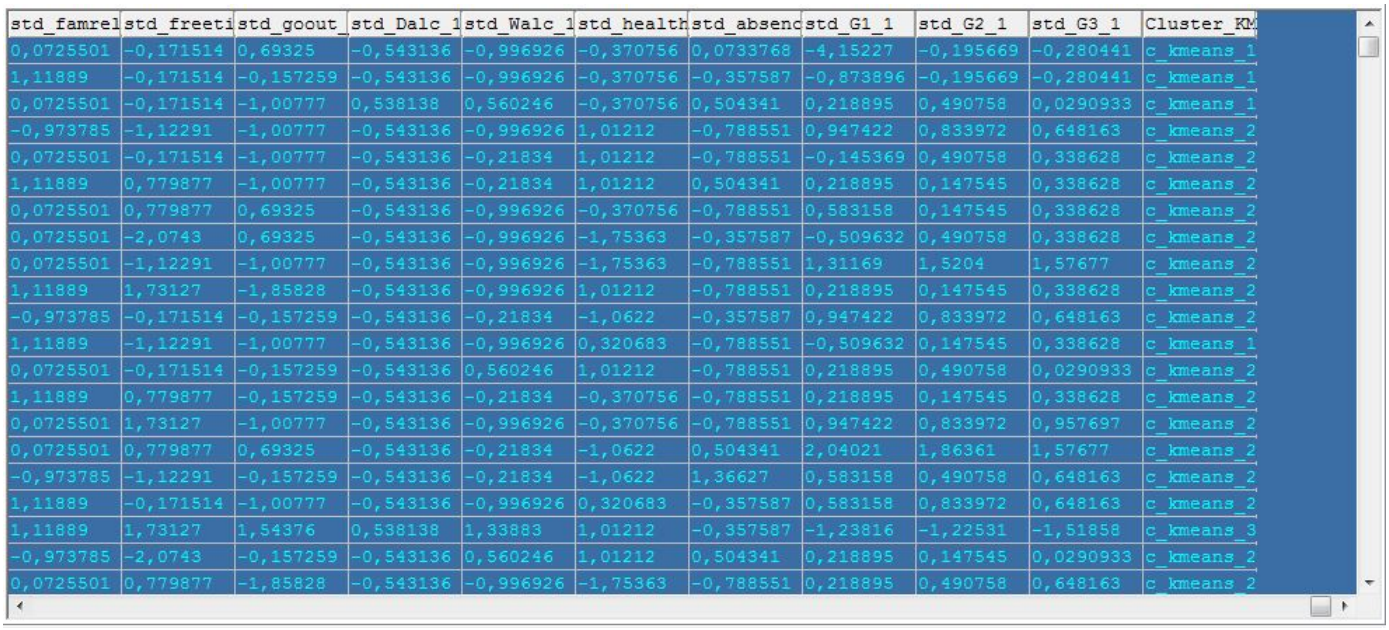



*Figure 10: fenêtre de parametrage de K-means*

- **1ére méthode de la répresentation**

  Nous allons affecter les éleves aux classes. Aprés executions de la méthode k-means,

le tableau généré produit automatiquement une variable supplémentaire dans la base courante. Le tableau décrit la classe affectée à chaque individu. Nous pouvons la visualiser avec le composant VIEW DATASET (onglet DATA VISUALIZATION). Elle est positionnée en dernière colonne.



| std_famrel | std_freeti | std_goout_ | std_Dalc_1 | std_Walc_1 | std_health | std_absenc | std_G1_1 | std_G2_1 | std_G3_1 | Cluster_KM |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,0725501 | -0,171514 | 0,69325 | -0,543136 | -0,996926 | -0,370756 | 0,0733768 | -4,15227 | -0,195669 | -0,280441 | c_kmeans_1 |
| 1,11889 | -0,171514 | -0,157259 | -0,543136 | -0,996926 | -0,370756 | -0,357587 | -0,873896 | -0,195669 | -0,280441 | c_kmeans_1 |
| 0,0725501 | -0,171514 | -1,00777 | 0,538138 | 0,560246 | -0,370756 | 0,504341 | 0,218895 | 0,490758 | 0,0290933 | c_kmeans_1 |
| -0,973785 | -1,12291 | -1,00777 | -0,543136 | -0,996926 | 1,01212 | -0,788551 | 0,947422 | 0,833972 | 0,648163 | c_kmeans_2 |
| 0,0725501 | -0,171514 | -1,00777 | -0,543136 | -0,21834 | 1,01212 | -0,788551 | -0,145369 | 0,490758 | 0,338628 | c_kmeans_2 |
| 1,11889 | 0,779877 | -1,00777 | -0,543136 | -0,21834 | 1,01212 | 0,504341 | 0,218895 | 0,147545 | 0,338628 | c_kmeans_2 |
| 0,0725501 | 0,779877 | 0,69325 | -0,543136 | -0,996926 | -0,370756 | -0,788551 | 0,583158 | 0,147545 | 0,338628 | c_kmeans_2 |
| 0,0725501 | -2,0743 | 0,69325 | -0,543136 | -0,996926 | -1,75363 | -0,357587 | -0,509632 | 0,490758 | 0,338628 | c_kmeans_2 |
| 0,0725501 | -1,12291 | -1,00777 | -0,543136 | -0,996926 | -1,75363 | -0,788551 | 1,31169 | 1,5204 | 1,57677 | c_kmeans_2 |
| 1,11889 | 1,73127 | -1,85828 | -0,543136 | -0,996926 | 1,01212 | -0,788551 | 0,218895 | 0,147545 | 0,338628 | c_kmeans_2 |
| -0,973785 | -0,171514 | -0,157259 | -0,543136 | -0,21834 | -1,0622 | -0,357587 | 0,947422 | 0,833972 | 0,648163 | c_kmeans_2 |
| 1,11889 | -1,12291 | -1,00777 | -0,543136 | -0,996926 | 0,320683 | -0,788551 | -0,509632 | 0,147545 | 0,338628 | c_kmeans_1 |
| 0,0725501 | -0,171514 | -0,157259 | -0,543136 | 0,560246 | 1,01212 | -0,788551 | 0,218895 | 0,490758 | 0,0290933 | c_kmeans_2 |
| 1,11889 | 0,779877 | -0,157259 | -0,543136 | -0,21834 | -0,370756 | -0,788551 | 0,218895 | 0,147545 | 0,338628 | c_kmeans_2 |
| 0,0725501 | 1,73127 | -1,00777 | -0,543136 | -0,996926 | -0,370756 | -0,788551 | 0,947422 | 0,833972 | 0,957697 | c_kmeans_2 |
| 0,0725501 | 0,779877 | 0,69325 | -0,543136 | -0,21834 | -1,0622 | 0,504341 | 2,04021 | 1,86361 | 1,57677 | c_kmeans_2 |
| -0,973785 | -1,12291 | -0,157259 | -0,543136 | -0,21834 | -1,0622 | 1,36627 | 0,583158 | 0,490758 | 0,648163 | c_kmeans_2 |
| 1,11889 | -0,171514 | -1,00777 | -0,543136 | -0,996926 | 0,320683 | -0,357587 | 0,583158 | 0,833972 | 0,648163 | c_kmeans_2 |
| 1,11889 | 1,73127 | 1,54376 | 0,538138 | 1,33883 | 1,01212 | -0,357587 | -1,23816 | -1,22531 | -1,51858 | c_kmeans_3 |
| -0,973785 | -2,0743 | -0,157259 | -0,543136 | 0,560246 | 1,01212 | 0,504341 | 0,218895 | 0,147545 | 0,0290933 | c_kmeans_2 |
| 0,0725501 | 0,779877 | -1,85828 | -0,543136 | -0,996926 | -1,75363 | -0,788551 | 0,218895 | 0,490758 | 0,648163 | c_kmeans_2 |

*Figure 11: 1ère méthode DATA VISUALISATION*

- **2ème méthode : la méthode graphique (nuage de points)**
  Une autre manière d'interpréter les résultats est de positionner les groupes dans l'espace des couples de variables. On peut ainsi analyser l'action conjointe de deux variables.Nous avons placé en abscisse G1(première note) et G2(note de la deuxième période) en ordonnée, sur la feuille une représentation colorée qui montre la dépendance entre les variables actives.

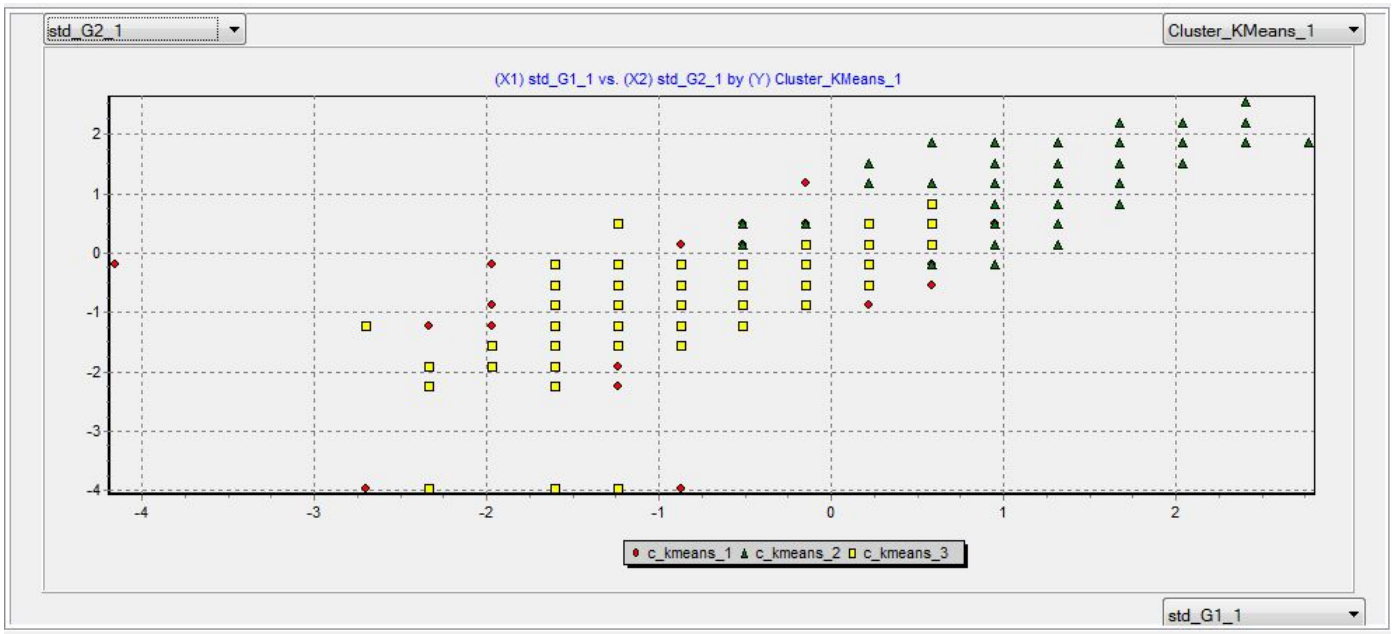



*Figure 12: Nuage de points*

### 5.2.2 La classification ascendante hierarchique (CAH)

Le principe de la CAH est de rassembler des individus selon un critère de ressemblance défini au préalable qui s'exprimera sous la forme d'une matrice de distances, exprimant la distance existant entre chaque individu pris deux à deux. Deux observations identiques auront une distance nulle. Plus les deux observations seront dissemblables, plus la distance sera importante. La CAH va ensuite rassembler les individus de manière itérative afin de produire un dendrogramme ou arbre de classification. La classification est ascendante car elle part des observations individuelles ; elle est hiérarchique car elle produit des classes ou groupes de plus en plus vastes, incluant des sous-groupes en leur sein. En découpant cet arbre à une certaine hauteur choisie, on produira la partition désirée.
En appliquant la composante view, nous voyons sur le dendogramme que le dataset est divisé en trois grandes classes subdivisés en sous classes. La division en classe est également visible au niveau du résultat du clustering

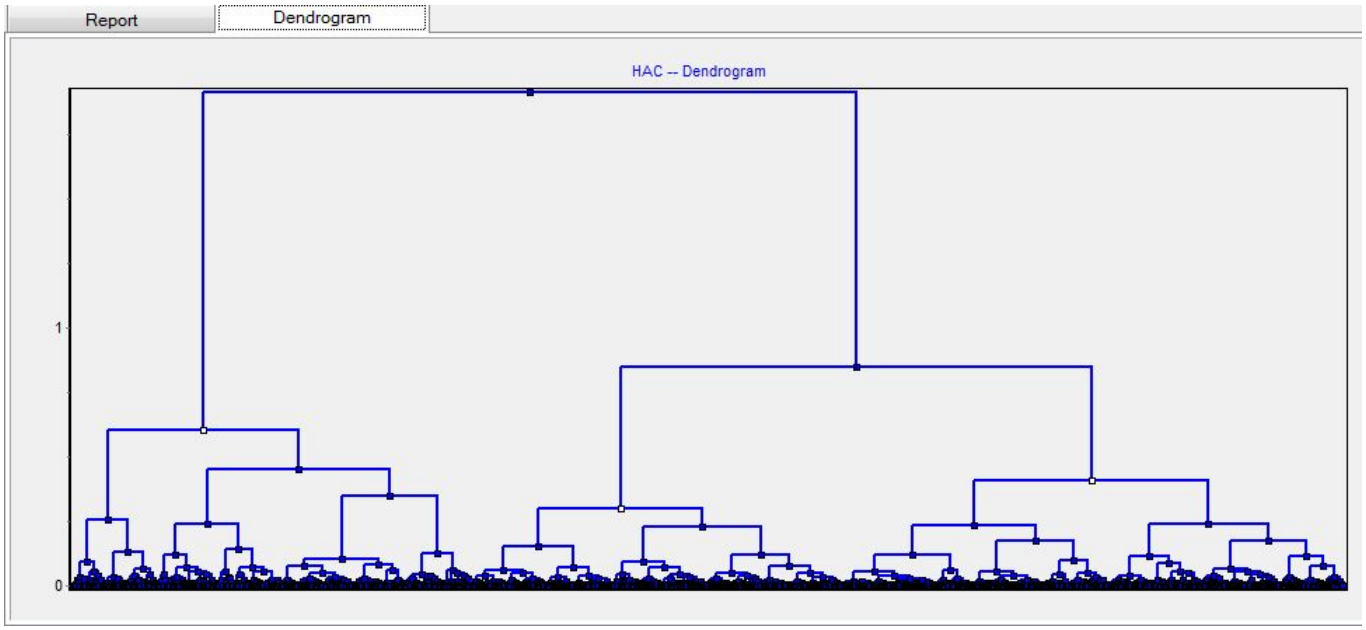



*Figure 13: Dendogramme*

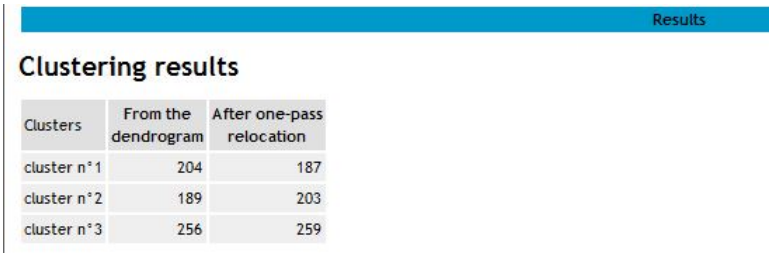



*Figure 14: Clustering*

# 6 Apprentissage Supervisé [7]

L'apprentissage est une discipline visant à la construction de règles d'inférence et de décision pour le traitement automatique des données. Elle s'applique à deux variantes le machine learning et la fouille de donnée(data-mining). La classification automatique supervisée:

- Elle consiste à examiner les caractéristiques d'un objet nouvellement présenté afin de l'affecter à une classe d'un ensemble prédéfini.
- Le modèle généré permet de prédire ou estimer la valeur manquante ou erronée en utilisant le modèle de classification comme référence.

## 6.1 Choix d'une méthode d'apprentissage supervisé

Après avoir utilisé différentes méthodes (méthodes factorielles et méthodes de classification) au niveau de l'analyse exploratoire, nous avons remarqué que l'attribut cible G3 a une forte corrélation avec les attributs G2 et G1. Cela se produit parce que G3 est la note de dernière

année (attribuée à la 3ème période), alors que G1 et G2 correspondent aux notes de 1ère et 2ème période. Il est plus difficile de prédire G3 sans G2 et G1, mais une telle prédiction est beaucoup plus utile.Pour ce faire le choix d'un meilleur algorithme d'apprentissage supervisé s'y impose.Parmi les différentes algorithmes d'apprentissage supervisé qui sont :

- Méthode de Bayes naïf
- K plus proches voisins
- Arbre de décision
- Réseaux de neurones

Nous avons opté pour le choix de l'Arbre de décision car :

- Il est lisible et facile à exécuter
- Performant sur de grands jeux de données : la méthode est relativement économique en termes de ressources de calcul.
- Peu de préparation des données
- Simple à comprendre et à interpréter

### 6.1.1 Construction de l'Arbre de décision [5]

Un arbre de décision est une représentation graphique d'une procédure de classification. Les noeuds internes de l'arbre sont des tests sur les champs ou attributs, les feuilles sont les classes. Lorsque les tests sont binaires, le fils gauche correspond à une réponse positive au test et le fils droit à une réponse négative.

L'objectif est de produire des groupes d'individus les plus homogènes possibles du point de vue de la variable à prédire.

Déterminer la bonne taille de l'arbre est une opération cruciale dans la construction d'un arbre de décision à partir de données. Elle détermine en grande partie ses performances lors de son déploiement dans la population.

Déterminer la bonne taille de l'arbre consiste donc à sélectionner, parmi les innombrables solutions que peuvent proposer l'induction, l'arbre le plus performant de la plus petite taille possible.

### 6.1.2 La méthode CART

Avec la méthode CART(Classification And Regression Trees), dans sa version la plus simple, l'échantillon de données est simplement fractionné en deux portions pour assurer ces deux étapes: Un échantillon d'expansion (growing set) sert à construire l'arbre le plus grand possible. Cet

arbre présente souvent une qualité de prédiction quasi-parfaite sur ces données.

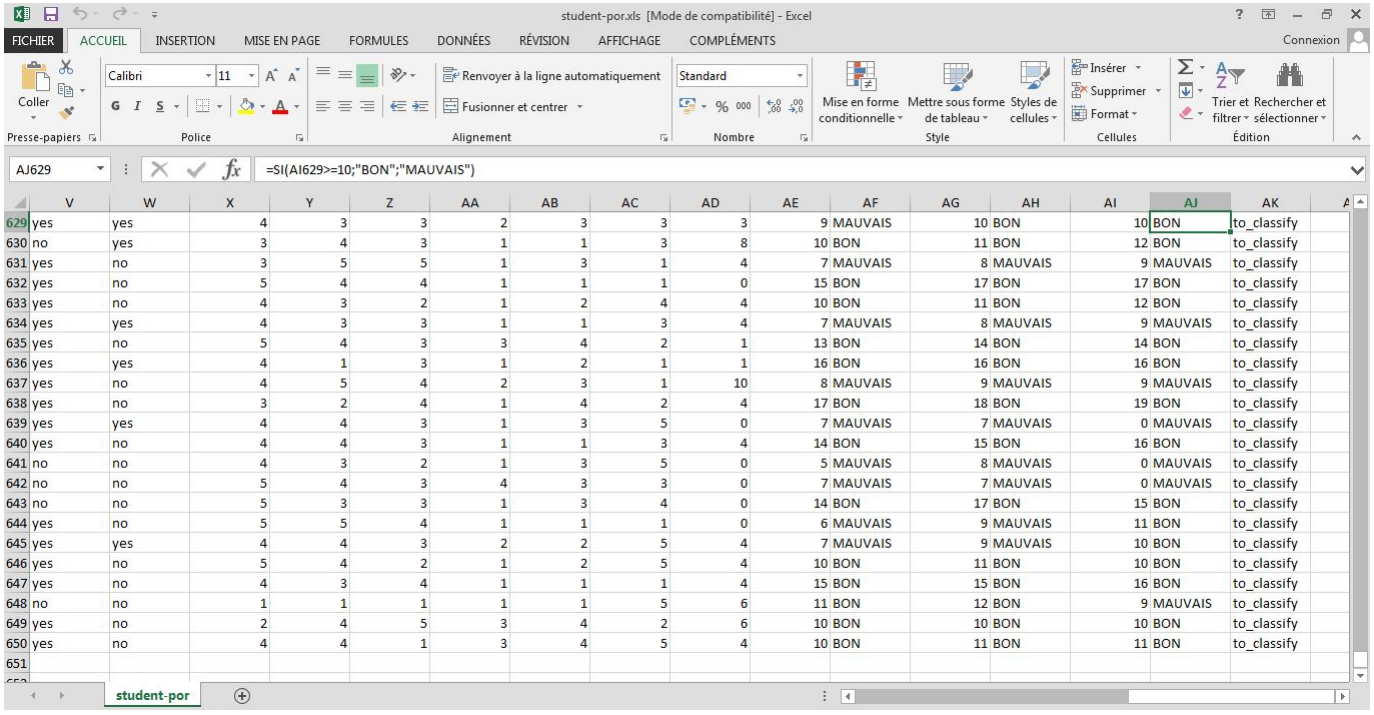
Ce second échantillon n'ayant pas servi lors de la construction de l'arbre, il devrait mieux rendre compte des performances des élèves

## 6.2 Préparation des données

L'objectif est de prédire la performance des élèves en s'appuyant principalement sur l'attribut G3 qui représente la note de fin d'année. Pour ce faire, nous devrons transformer les attributs G1,G2 et G3 en des valeurs discrètes ceci dans l'optique que les données soient facilement segmentale afin d'être utilisable par la méthode CART.

### 6.2.1 Pré-traitement des données

Une variable indicatrice statut permet de spécifier l'appartenance d'un individu à l'un ou l'autre des sous échantillons




| | V | W | X | Y | Z | AA | AB | AC | AD | AE | AF | AG | AH | AI | AJ | AK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 629 | yes | yes | 4 | 3 | 3 | 2 | 3 | 3 | 3 | 9 | MAUVAIS | 10 | BON | 10 | BON | to_classify |
| 630 | no | yes | 3 | 4 | 3 | 1 | 1 | 3 | 8 | 10 | BON | 11 | BON | 12 | BON | to_classify |
| 631 | yes | no | 3 | 5 | 5 | 1 | 3 | 1 | 4 | 7 | MAUVAIS | 8 | MAUVAIS | 9 | MAUVAIS | to_classify |
| 632 | yes | no | 5 | 4 | 4 | 1 | 1 | 1 | 0 | 15 | BON | 17 | BON | 17 | BON | to_classify |
| 633 | yes | no | 4 | 3 | 2 | 1 | 2 | 4 | 4 | 10 | BON | 11 | BON | 12 | BON | to_classify |
| 634 | yes | yes | 4 | 3 | 3 | 1 | 1 | 3 | 4 | 7 | MAUVAIS | 8 | MAUVAIS | 9 | MAUVAIS | to_classify |
| 635 | yes | no | 5 | 4 | 3 | 3 | 4 | 2 | 1 | 13 | BON | 14 | BON | 14 | BON | to_classify |
| 636 | yes | yes | 4 | 1 | 3 | 1 | 2 | 1 | 1 | 16 | BON | 16 | BON | 16 | BON | to_classify |
| 637 | yes | no | 4 | 5 | 4 | 2 | 3 | 1 | 10 | 8 | MAUVAIS | 9 | MAUVAIS | 9 | MAUVAIS | to_classify |
| 638 | yes | no | 3 | 2 | 4 | 1 | 4 | 2 | 4 | 17 | BON | 18 | BON | 19 | BON | to_classify |
| 639 | yes | yes | 4 | 4 | 3 | 1 | 3 | 5 | 0 | 7 | MAUVAIS | 7 | MAUVAIS | 0 | MAUVAIS | to_classify |
| 640 | yes | no | 4 | 4 | 3 | 1 | 1 | 3 | 4 | 14 | BON | 15 | BON | 16 | BON | to_classify |
| 641 | no | no | 4 | 3 | 2 | 1 | 3 | 5 | 0 | 5 | MAUVAIS | 8 | MAUVAIS | 0 | MAUVAIS | to_classify |
| 642 | no | no | 5 | 4 | 3 | 4 | 3 | 3 | 0 | 7 | MAUVAIS | 7 | MAUVAIS | 0 | MAUVAIS | to_classify |
| 643 | no | no | 5 | 3 | 3 | 1 | 3 | 4 | 0 | 14 | BON | 17 | BON | 15 | BON | to_classify |
| 644 | yes | no | 5 | 5 | 4 | 1 | 1 | 1 | 0 | 6 | MAUVAIS | 9 | MAUVAIS | 11 | BON | to_classify |
| 645 | yes | yes | 4 | 4 | 3 | 2 | 2 | 5 | 4 | 7 | MAUVAIS | 9 | MAUVAIS | 10 | BON | to_classify |
| 646 | yes | no | 5 | 4 | 2 | 1 | 2 | 5 | 4 | 10 | BON | 11 | BON | 10 | BON | to_classify |
| 647 | yes | no | 4 | 3 | 4 | 1 | 1 | 1 | 4 | 15 | BON | 15 | BON | 16 | BON | to_classify |
| 648 | no | no | 1 | 1 | 1 | 1 | 1 | 5 | 6 | 11 | BON | 12 | BON | 9 | MAUVAIS | to_classify |
| 649 | yes | no | 2 | 4 | 5 | 3 | 4 | 2 | 6 | 10 | BON | 10 | BON | 10 | BON | to_classify |
| 650 | yes | no | 4 | 4 | 1 | 3 | 4 | 5 | 4 | 10 | BON | 11 | BON | 11 | BON | to_classify |
| 651 | | | | | | | | | | | | | | | | |

*Figure 15: Pré-traitement des données*

Notre objectif est de produire, en nous appuyant sur la méthodologie CART, un arbre de décision à la fois performant et peu complexe c.-à-d. comportant le moins de feuilles (règles) possible. Bien entendu, pour que l'évaluation soit crédible, l'échantillon test ne doit être utilisé qu'en dernier ressort, pour comparer les performances des modèles alternatifs proposés. En aucune manière, il ne doit être mis à contribution pour guider l'exploration des solutions.

### 6.2.2 Choix d'un ensemble d'entraînement et d'un ensemble de validation dans les données

Pour une meilleure estimation des performances des méthodes d'apprentissage, nous allons scinder notre ensemble de données en deux parties : 520 observations pour la construction des modèles de prédiction ; 129 observations pour leur évaluation.

## 6.3 Arbre de décision – La méthode CART

[6] Cette méthode permet d'inférer des arbres de décision binaires, tous les tests étiquetant les noeuds de décision sont binaires. Le langage de représentation est constitué d'un certain nombre d'attributs. Ces attributs peuvent être binaires, qualitatifs ou continus.
Le nombre de tests à explorer va dépendre de la nature des attributs. A un attribut binaire correspond un test binaire. A un attribut qualitatif ayant n modalités, on peut associer autant de tests qu'il y a de partitions en deux classes, soit $2^{n-1}$ tests binaires possibles.
Enfin, dans le cas d'attributs continus, il y a une infinité de tests envisageables. Dans ce cas, on découpe l'ensemble des valeurs possibles en segments, ce découpage peut être fait par un expert ou fait de façon automatique.

### 6.3.1 Créer un diagramme et importer les données dans TANAGRA

Ci-dessous les différentes étapes qui se déroulent dans TANAGRA :

- Après avoir lancé TANAGRA, cliquez sur File/New et au niveau de type de fichier, on sélectionne Excel file et on clique sur notre fichier Student-por.xls puis ouvrir
- Donner un nom au diagramme  Performance des élèves

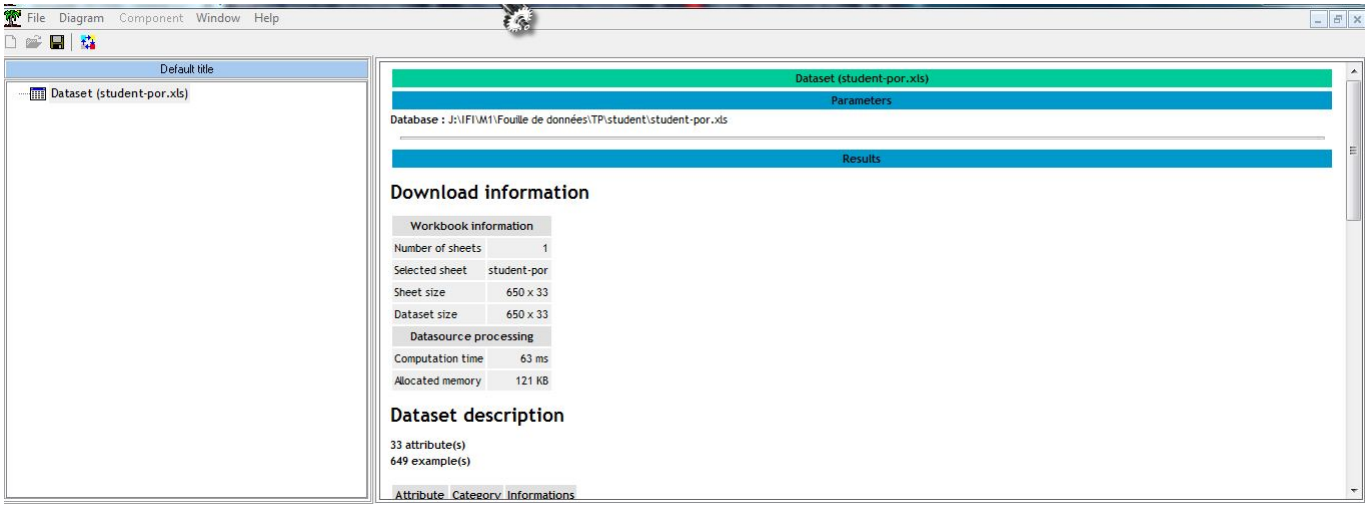



*Figure 16: Importation des données*

Sur la figure, Nous remarquons que nos données ont été chargés et que nous avons bien 649 instances et 33 attributs

### 6.3.2 Subdiviser les données

Nous introduisons le composant DISCRETE SELECT EXAMPLES (onglet INSTANCE SELECTION) dans le diagramme pour désigner les observations dédiées à l'apprentissage. Nous le paramétrons (menu contextuel PARAMETERS) : INDEX joue le rôle de variable de contrôle, les individus actifs correspondent à la valeur LEARNING.

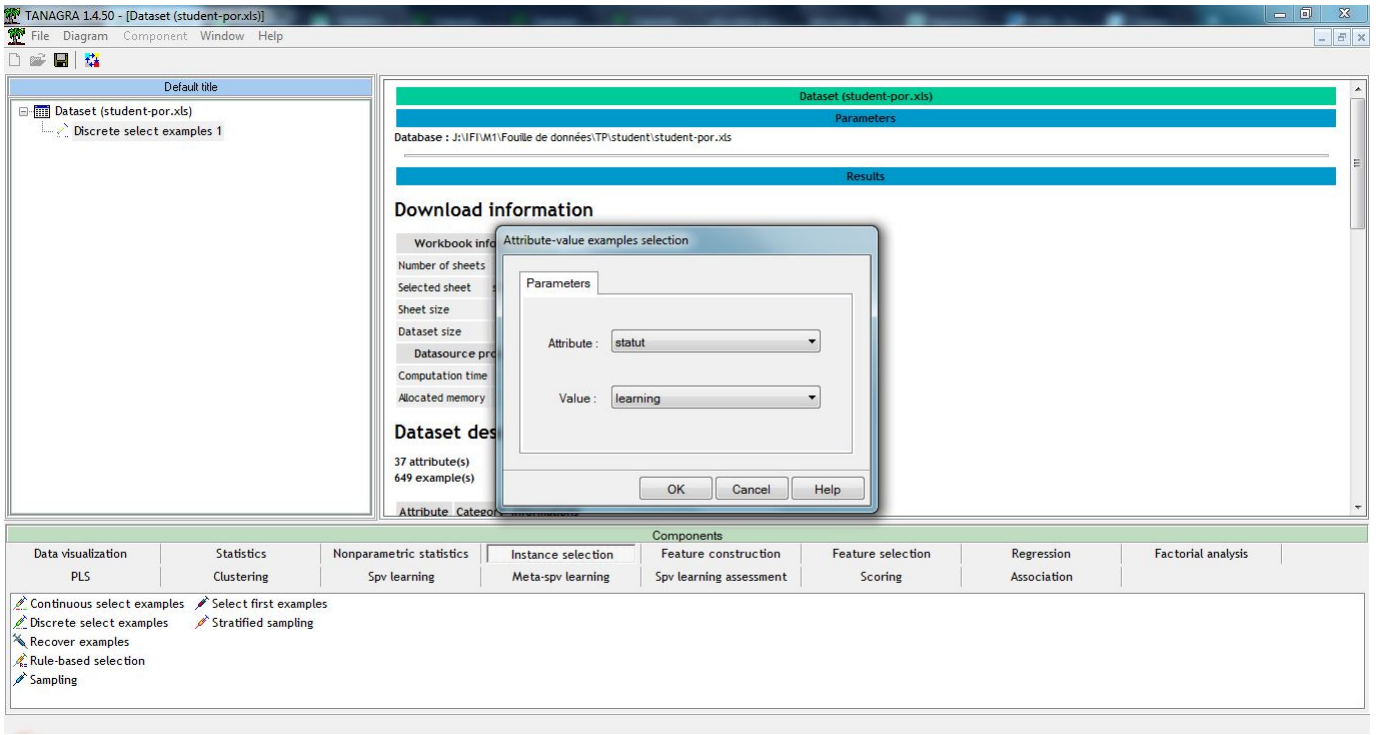



*Figure 17: Subdivision des données*

Après validation (bouton OK), le menu contextuel VIEW permet d'afficher le résultat de la sélection, 520 élèves sont sélectionnés pour l'induction de l'arbre.

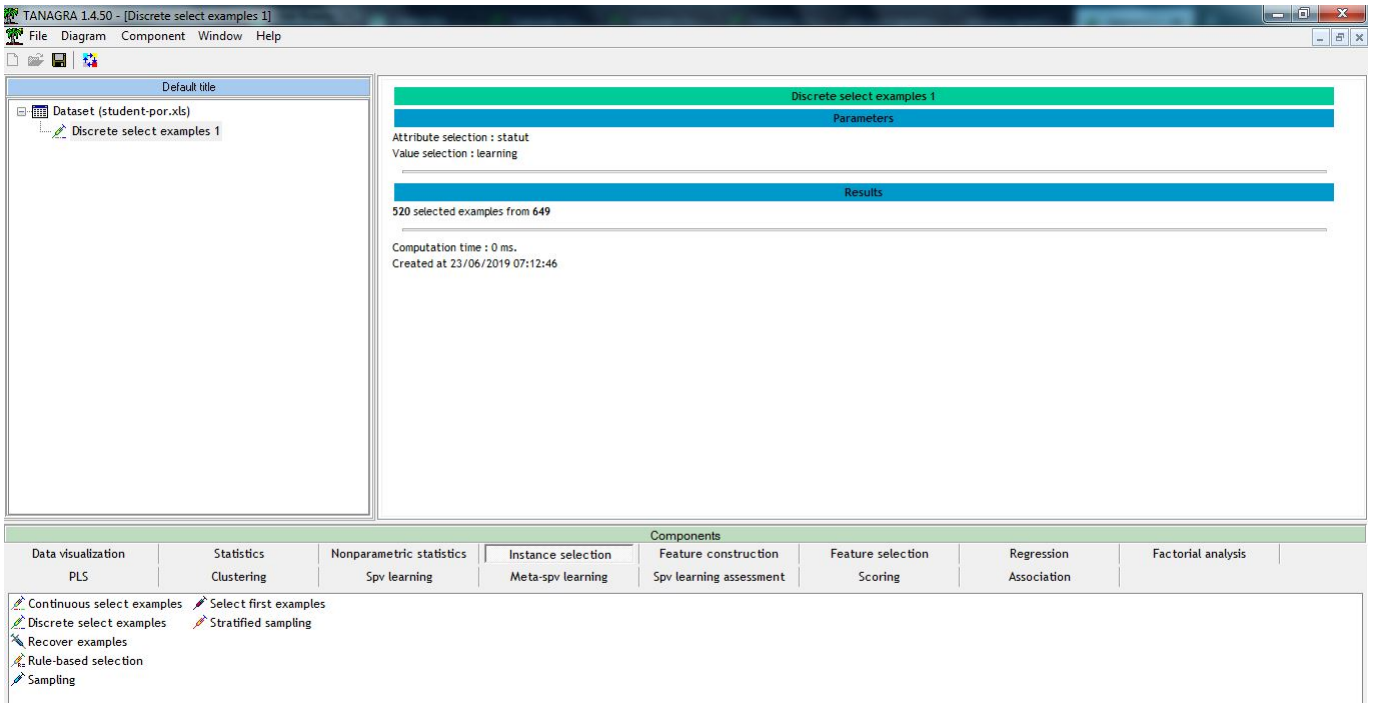



*Figure 18: Affichage des données subdivisés*

### 6.3.3 Variable dépendante et variables prédictives

Pour décrire le problème de prédiction à résoudre, nous introduisons le composant DEFINE STATUS dans le diagramme. Le plus simple est d'actionner le raccourci dans la barre d'outils. Nous plaçons la variable G3 en TARGET, et le reste des variables en INPUT,excepté celle de statut.

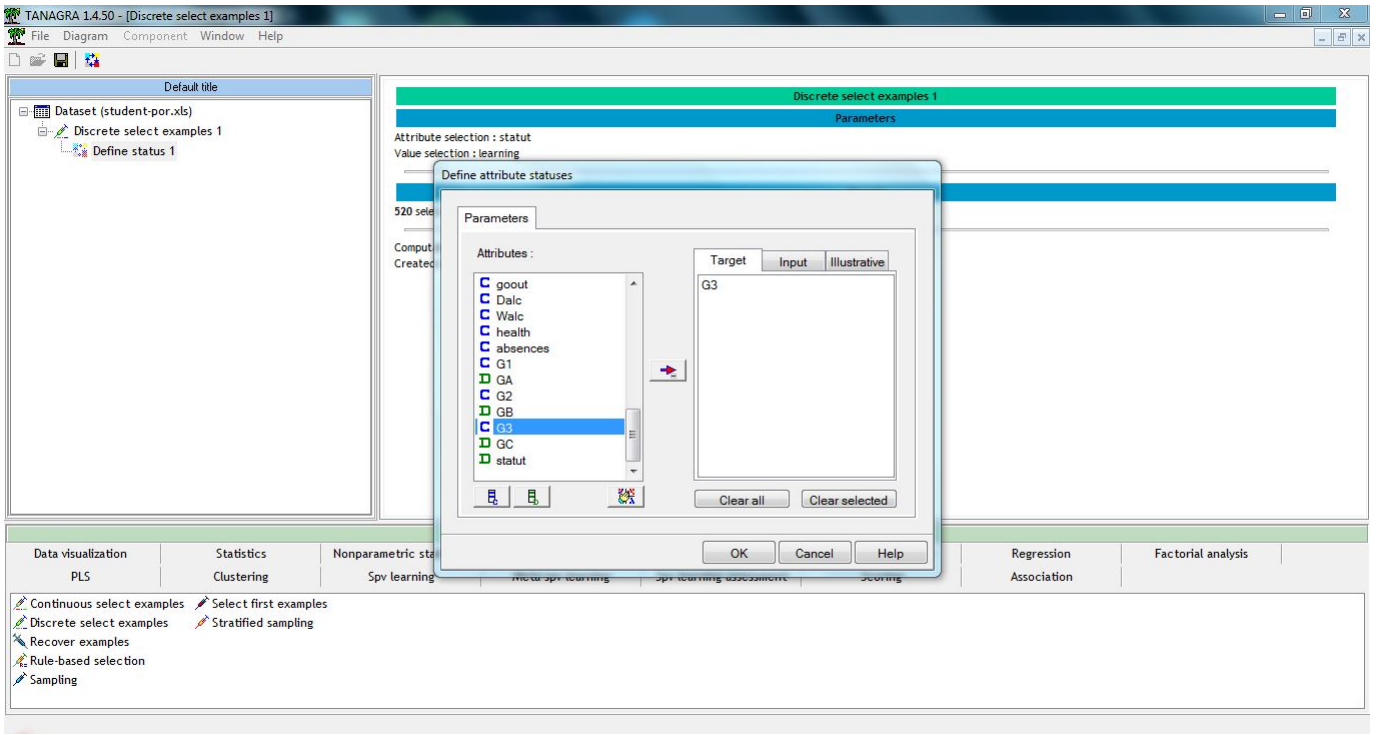



*Figure 19: Sélection des attributs pour l'apprentissage*

## 6.4 Apprentissage avec la méthode C-RT

Nous introduisons le composant C-RT (onglet SPV LEARNING) dans le diagramme. Nous activons le menu VIEW pour obtenir les résultats. Selon la puissance de la machine, le résultat sera plus ou moins long à venir. Détaillons les différentes sections du rapport fourni par TANAGRA.

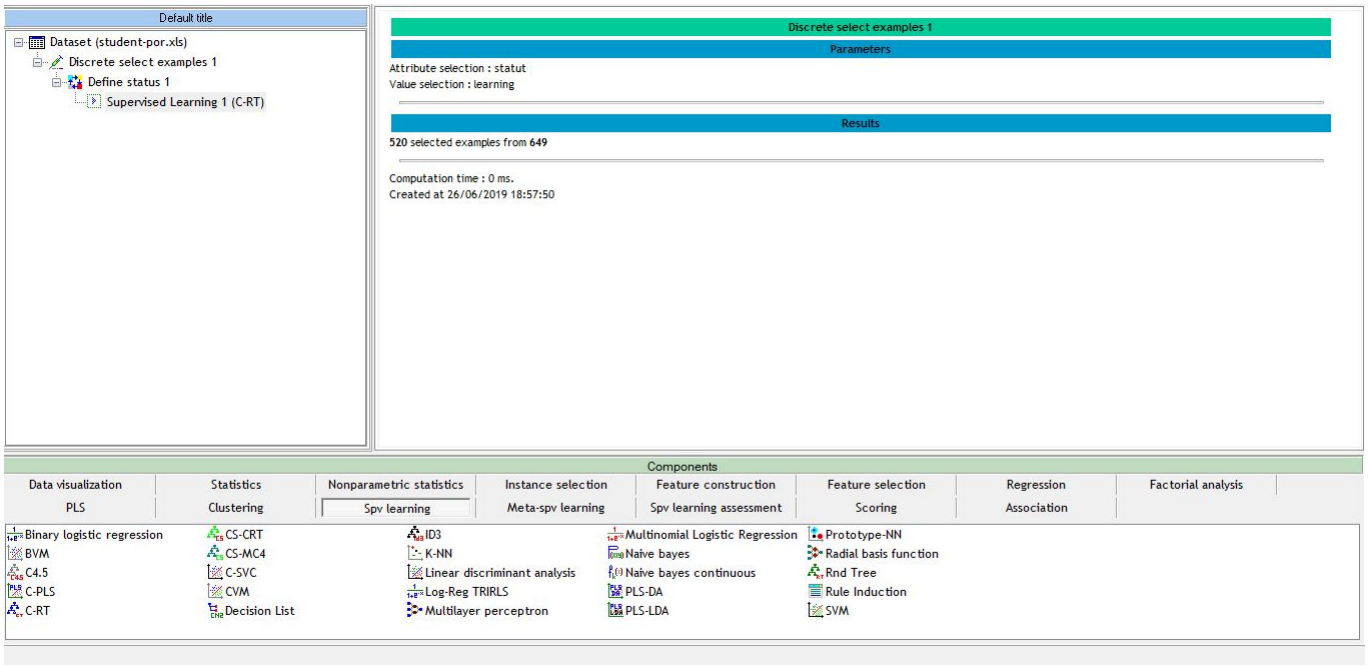



*Figure 20: Apprentissage C-RT*

### 6.4.1 Matrice de confusion

La matrice de confusion confronte les vraies valeurs et les valeurs prédites de GC sur les 520 observations ayant participé à l'apprentissage (growing + pruning). Elle est accompagnée du taux d'erreur qui est de 0.0577 dans notre exemple. Etant calculé sur les données ayant servi à construire l'arbre, cet indicateur est souvent optimiste. Néanmoins, l'importance de l'écart dépend en partie de l'aptitude de la technique à coller exagérément aux données.

**Classifier performances**

| Error rate | 0,0577 |
|---|---|
| Values prediction | Confusion matrix |

| Value | Recall | 1-Precision |
|---|---|---|
| BON | 0,9544 | 0,0200 |
| MAUVAIS | 0,8475 | 0,2958 |

| | BON | MAUVAIS | Sum |
|---|---|---|---|
| BON | 440 | 21 | 461 |
| MAUVAIS | 9 | 50 | 59 |
| Sum | 449 | 71 | 520 |

*Figure 21: Classifers performance-Matrice de confusion*

### 6.4.2 Partition des données

TANAGRA nous indique que parmi les 520 observations dédiées à l'apprentissage, il a réservé 348 observations pour l'expansion de l'arbre (growing set) et 172 pour le post élagage (pruning set). La partition a été effectuée de manière aléatoire.

**Data partition**

| | |
|---|---|
| Growing set | 348 |
| Pruning set | 172 |

*Figure 22: Data partition*

### 6.4.3 Séquence d'arbre

Nous détaillerons ce tableau plus loin. A ce stade, on se contentera de constater que l'arbre le plus grand lors de la phase d'expansion comporte 11 feuilles, le taux d'erreur sur le growing set est de 0.0172, sur le pruning set 0.1047. L'arbre optimal sur le pruning set comporte 5 feuilles, avec un taux d'erreur 0.0202 (pruning set). En appliquant la règle de l'écart type (1-SE RULE), l'arbre retenu par CART comporte 4 feuilles avec un taux d'erreur de 0.0202 sur le pruning set (à comparer avec les 11 feuilles obtenues initialement). Nous détaillerons plus loin la procédure de calcul utilisée par CART. Enfin l'arbre trivial composé de la seule racine présente un taux d'erreur de 0.0244

**Trees sequence (# 7)**

| N° | # Leaves | Err (growing set) | Err (pruning set) | SE (pruning set) | x |
|---|---|---|---|---|---|
| 7 | 1 | 0,1121 | 0,1163 | 0,0244 | 2,019259 |
| 6 | 2 | 0,0776 | 0,1105 | 0,0239 | 1,730793 |
| 5 | 4 | 0,0489 | 0,0756 | 0,0202 | 0,000000 |
| 4 | 5 | 0,0374 | 0,0756 | 0,0202 | - |
| 3 | 7 | 0,0259 | 0,0814 | 0,0208 | - |
| 2 | 9 | 0,0201 | 0,0872 | 0,0215 | - |
| 1 | 11 | 0,0172 | 0,1047 | 0,0233 | - |

*Figure 23: Description de l'arbre*

### 6.4.4 Description de l'arbre

**Tree description**

| | |
|---|---|
| Number of nodes | 7 |
| Number of leaves | 4 |

**Decision tree**

- GB in [BON] then GC = **BON** (99,30 % of 286 examples)
- GB in [MAUVAIS]
  - famrel < 4,5000
    - GA in [MAUVAIS] then GC = **MAUVAIS** (61,54 % of 26 examples)
    - GA in [BON] then GC = **BON** (77,78 % of 18 examples)
  - famrel >= 4,5000 then GC = **MAUVAIS** (94,44 % of 18 examples)

Computation time : 63 ms.
Created at 26/06/2019 20:11:26

*Figure 24: Description de l'arbre*

La dernière section décrit l'arbre de décision produit lors de l'induction. On se bornera à noter que les attributs GB(note deuxiéme trimestre), famrel(Qualité de relation familiale et GA(note premier trimestre) semblent les plus déterminants. Donc nous remarquons que l'attribut GB est de (99,30% sur 286 exemples) et famrel(94,44% sur 18 exemples), on en déduit que si l'élève a une forte note au second trimestre et une bonne qualité en relation familiale donc il aura une bonne note au troisième trimestre(G3).

## 6.5 Évaluation sur l'échantillon test

Les ensembles growing et pruning participent, chacun à leur manière, à l'élaboration du modèle de prédiction. A ce titre, ils fournissent une estimation optimiste des performances puisque l'arbre est optimisé pour ces données. Pour obtenir une évaluation réellement non biaisée, il faut utiliser un ensemble test qui n'a jamais participé, de près ou de loin, à l'apprentissage. C'est à ce stade que nous mettons à contribution les individus « Index = test ». Dans un premier temps, nous introduisons de nouveau le composant DEFINE STATUS dans le diagramme. Nous devons indiquer à TANAGRA la variable à prédire observée GC (TARGET) et la variable PRED_SPVINSTANCE_1 produite automatiquement par le composant C-RT (INPUT). Cette nouvelle colonne dans nos données comporte les valeurs prédites par l'arbre de décision, tant sur les données sélectionnées que sur les données non sélectionnées.

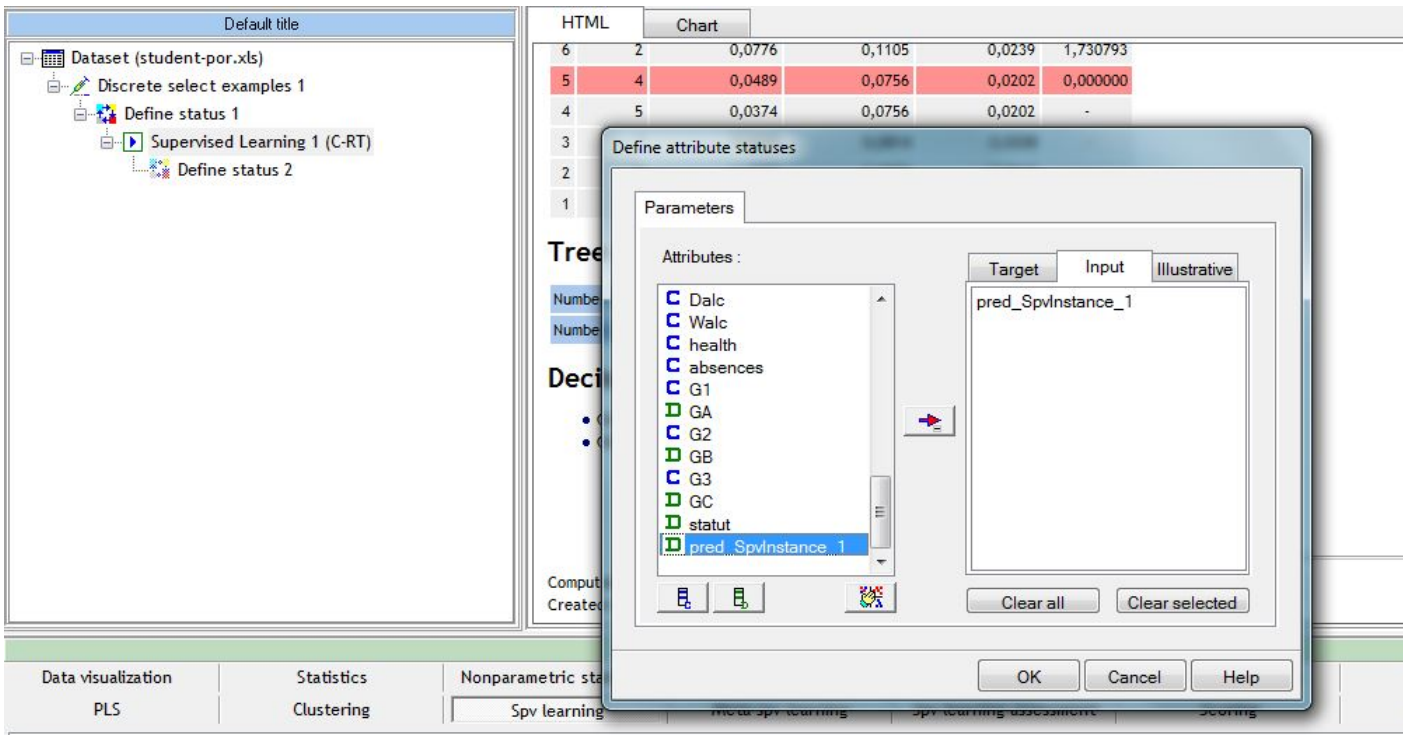



*Figure 25: Données Test Évaluation*

Dans un deuxième temps, nous introduisons le composant TEST (onglet SPV LEARNING ASSESSMENT) dans le diagramme. Elle est paramétrée par défaut pour construire la matrice

de confusion, et calculer le taux d'erreur, sur les données non sélectionnées.
Nous activons le menu VIEW. Nous obtenons un taux d'erreur de 0.1395 calculé sur les 129 individus que nous avons mis de côté initialement. Ce n'est qu'une estimation bien sûr. Mais étant calculé sur un effectif aussi élevé, nous pouvons penser qu'il est relativement fiable : l'intervalle de confiance à 95% est [0.1473 ; 0.1545

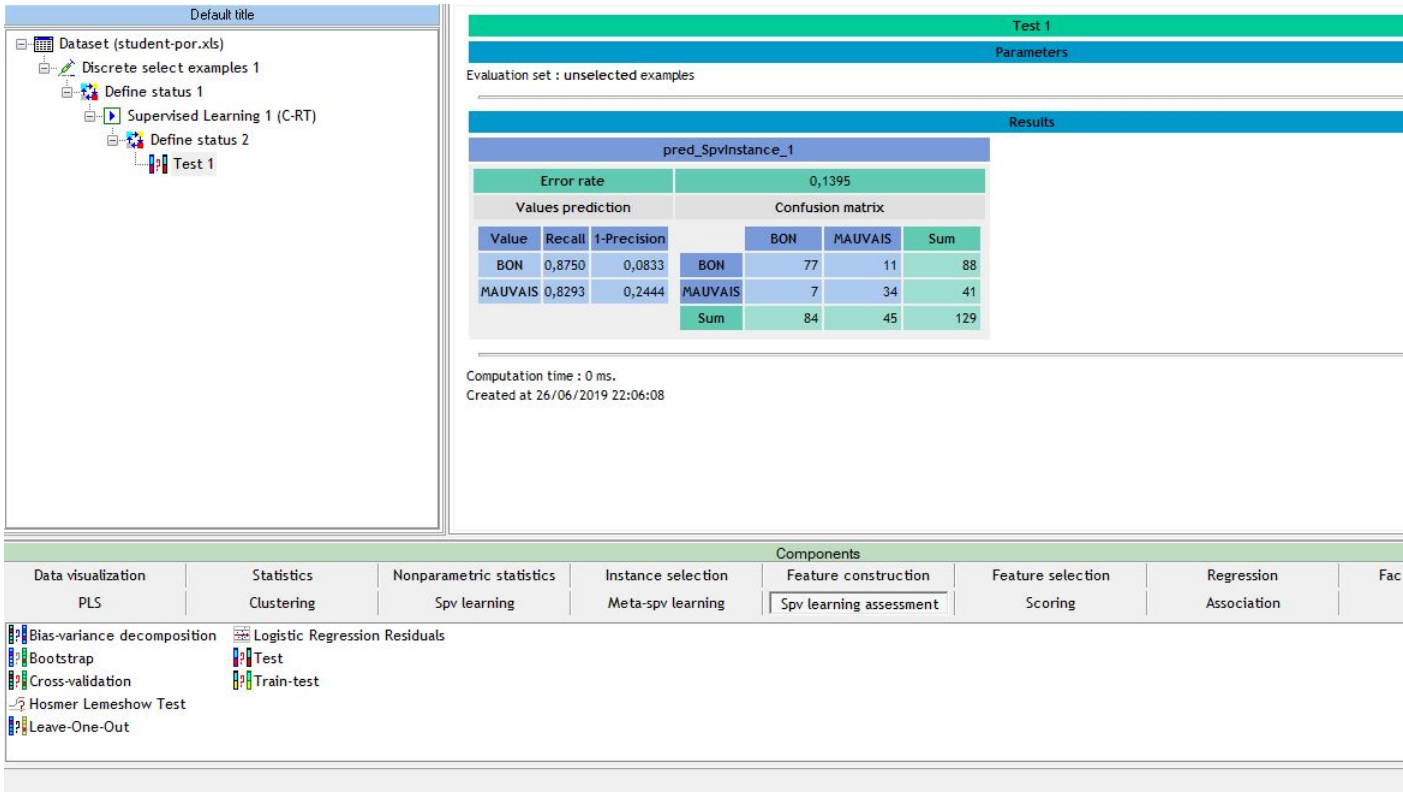



*Figure 26: Affichage Test Évaluation*

## 6.6 Quelques variantes autour du post-élagage

### 6.6.1 La 0-SE RULE

Une question revient très souvent chez les utilisateurs : pourquoi ne pas choisir directement l'arbre qui minimise l'erreur sur le pruning set ? Cette partie des données n'a pas servi à l'expansion de l'arbre, nous serons ainsi assurés de choisir un arbre « optimal ». Il y a plusieurs réponses possibles. La première repose sur le bon sens : pourquoi reporter sur l'échantillon pruning ce que nous voulions justement éviter sur l'échantillon growing ? A savoir éviter de trop optimiser sur un échantillon au risque d'ingérer indûment les spécificités des données ? La seconde réponse repose sur l'étude de la courbe opposant le nombre de feuilles de l'arbre (sa complexité) et le taux d'erreur sur le pruning set. Voyons comment obtenir cette courbe avec TANAGRA.

### 6.6.2 Courbe d'erreur en fonction de la complexité de l'arbre

Pour obtenir le détail de la courbe d'erreur, nous activons le menu contextuel SUPERVISED PARAMETERS du composant SUPERVISED LEARNING 1 (C-RT) dans le diagramme. Nous sélectionnons l'option SHOW ALL TREE SEQUENCE.

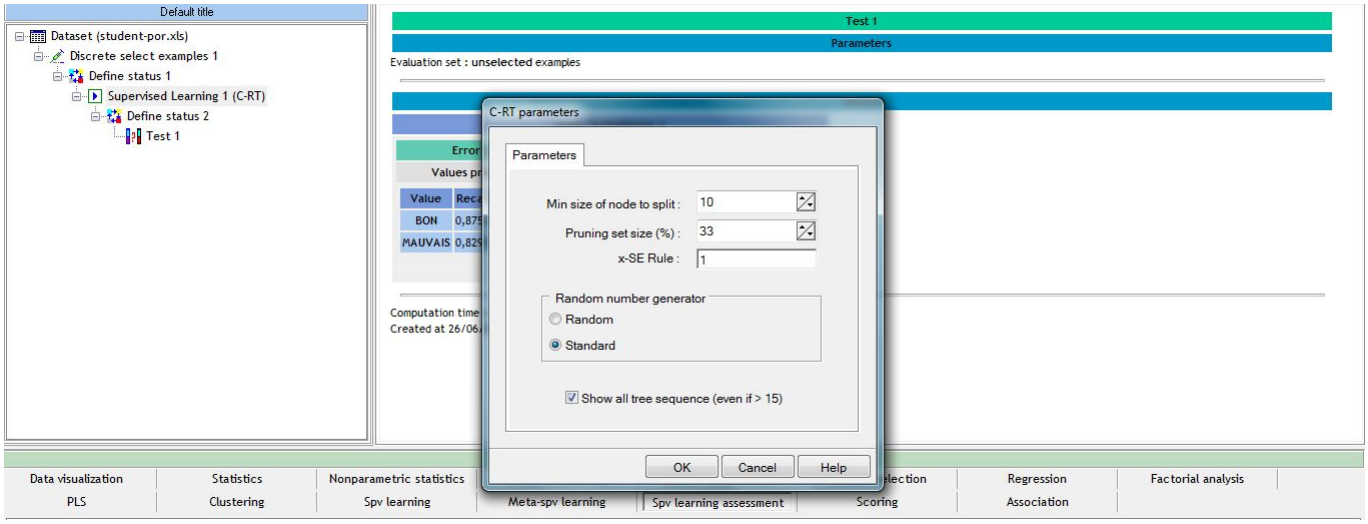



*Figure 27: Courbe d'erreur*

Nous actionnons le menu VIEW, le tableau retraçant les erreurs est détaillé maintenant. Nous remarquons plusieurs choses : le nombre de feuilles des arbres qui ont été testés n'est pas régulier, le mécanisme de coût complexité permet de réduire considérablement les solutions à évaluer ; l'erreur sur l'échantillon growing diminue constamment à mesure que le nombre de feuilles augmente ; l'erreur sur le pruning set diminue rapidement d'abord, semble stagner sur un palier, puis se dégrade lorsque le nombre de feuilles devient exagéré. Nous reproduisons le tableau (Figure 28) et le graphique correspondant (Figure 28). Nous constatons que les solutions allant d'un arbre avec 1 feuille à un arbre avec 2 feuilles sont similaires en termes de taux d'erreur sur l'échantillon pruning. L'arbre « optimal » comporte 4 feuilles, il propose un taux d'erreur de 0.0756. Mais nous comprenons aisément que nous avons tout intérêt à choisir un arbre proche du « coude » dans la courbe d'erreur (Figure 29). Nous conservons un bon niveau de performances avec un arbre réduit.

**Trees sequence (# 7)**

| N° | # Leaves | Err (growing set) | Err (pruning set) | SE (pruning set) | x |
|---|---|---|---|---|---|
| 7 | 1 | 0,1121 | 0,1163 | 0,0244 | 2,019259 |
| 6 | 2 | 0,0776 | 0,1105 | 0,0239 | 1,730793 |
| 5 | 4 | 0,0489 | 0,0756 | 0,0202 | 0,000000 |
| 4 | 5 | 0,0374 | 0,0756 | 0,0202 | - |
| 3 | 7 | 0,0259 | 0,0814 | 0,0208 | - |
| 2 | 9 | 0,0201 | 0,0872 | 0,0215 | - |
| 1 | 11 | 0,0172 | 0,1047 | 0,0233 | - |

*Figure 28: Complexité de l'arbre et taux d'erreur growing / pruning*

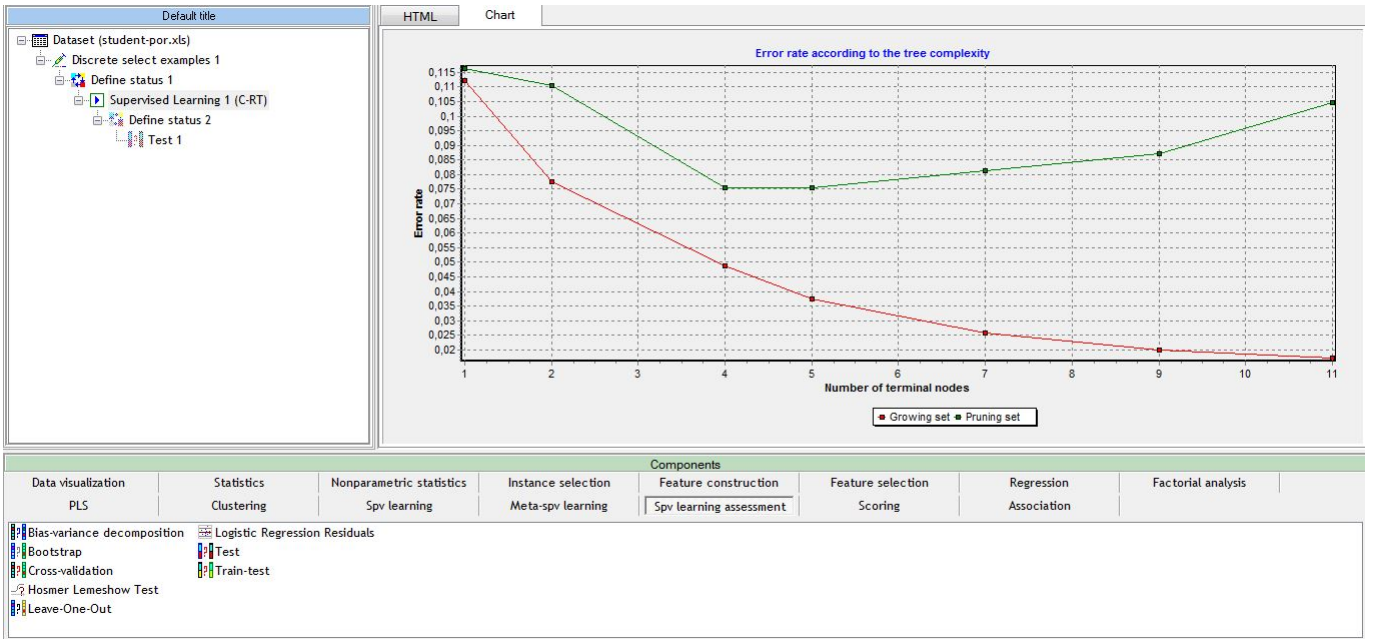



*Figure 29: Evolution du taux d'erreur en fonction de la complexité de l'arbre*

### 6.6.3 Fonctionnement de la règle de l'écart type : 1-SE RULE

La règle de l'écart-type : $\theta = 1$ nous permet de choisir dans notre tableau le plus petit arbre dont l'erreur sur le pruning set est en dessous de ce seuil. Il s'agit de l'arbre comportant 4 feuilles, avec un taux d'erreur de 0.0756. C'est le mécanisme que CART met en place pour sélectionner l'arbre final, il illustre à merveille le principe de parcimonie.

### 6.6.4 Performances de l'arbre 0-SE RULE $\theta = 0$

Néanmoins, nous voulons évaluer les performances de l'arbre optimal comportant 4 feuilles. Nous paramétrons de nouveau le composant SUPERVISED LEARNING 1 (C-RT) en imposant la valeur 0-SE RULE.

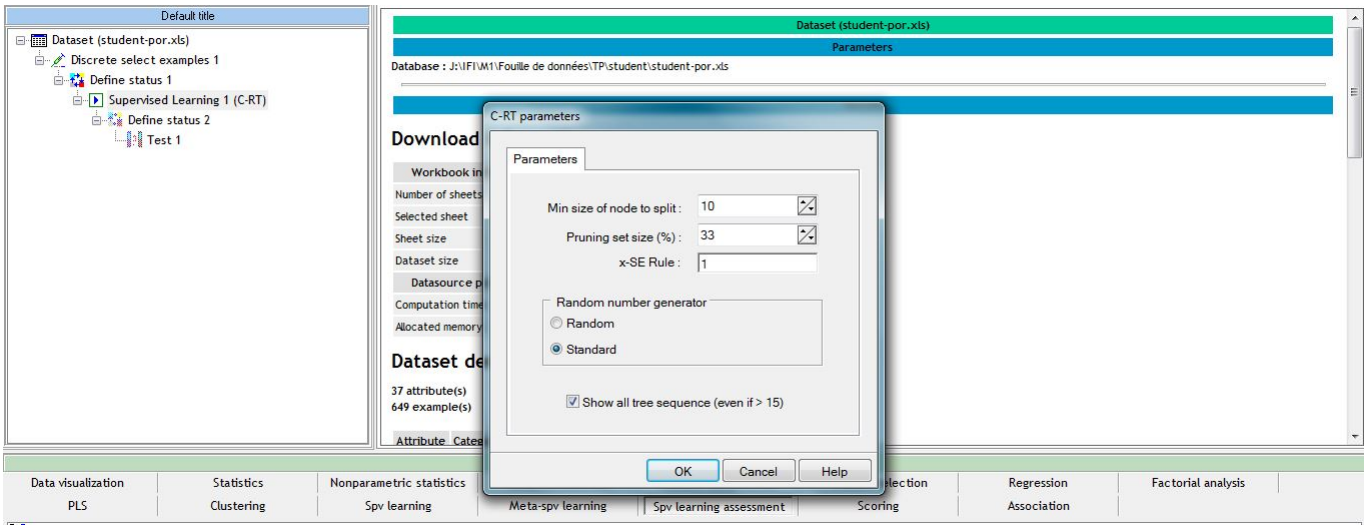



*Figure 30: Performance de l'arbre*

Voyons maintenant ce qu'il en est sur l'échantillon test, nous activons pour cela le menu VIEW du composant TEST 1 au bout du diagramme de traitement.

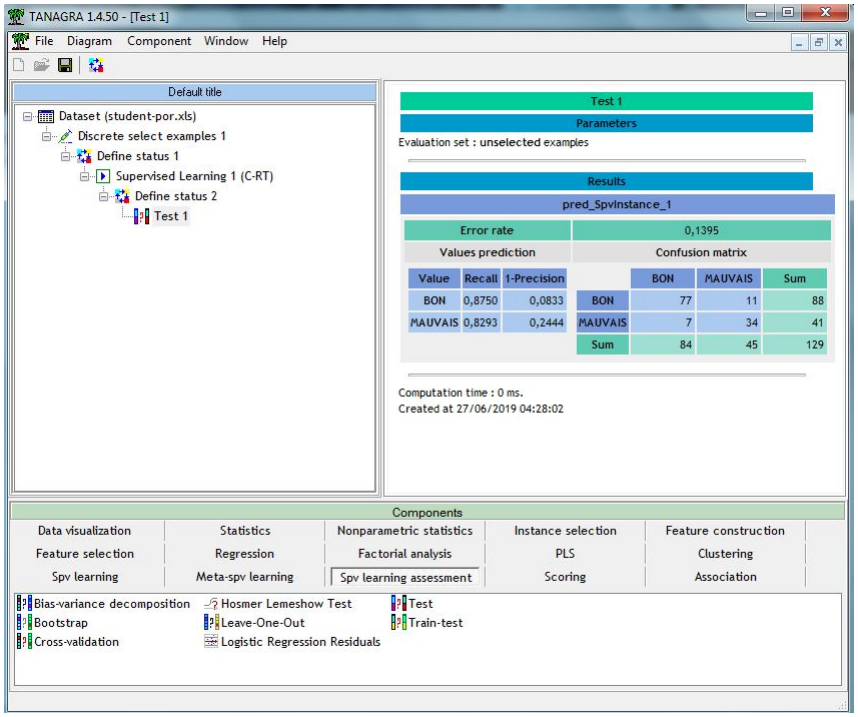



*Figure 31: Affichage Évaluation Test avec $\theta = 1$*

Le taux d'erreur en test est 0.1395, avec un intervalle de confiance à 88% égal à [0.1312 ; 0.1398].

### 6.6.5 Exploiter la courbe d'erreur du post élagage

La règle de l'écart type (1-SE RULE) vise à produire un arbre plus simple tout en conservant un bon niveau de performances. Intuitivement, nous comprenons qu'elle cherche à se rapprocher du « coude » dans le graphique de l'erreur (Figure 2), l'endroit où nous avons épuisé l'information utile et commençons à ingérer les spécificités du fichier de données dans l'arbre. Nous constatons également qu'elle est assez finalement approximative, dépendante du paramètre de pénalisation $\theta$. Dans cette section, nous allons essayer d'utiliser les outils à notre disposition (le Tableau 28 et la Figure 29), pour définir l'arbre correspondant à la solution souhaitée.

### 6.6.6 Déterminer le paramètre de $\theta$

A la lumière de la Figure 29, nous souhaitons produire l'arbre à 1 feuille. Le taux d'erreur sur le pruning set est de 0.1163. Pour déterminer la valeur du paramètre $\theta$ permettant de produire cet arbre, il faut définir l'erreur seuil de manière à ce qu'elle soit située entre 0.1497 (arbre à 4 feuilles) et 0.1539 (arbre à feuilles). En tâtonnant un peu, nous obtenons, entre autres $\theta$= 0.7 , avec un erreur seuil = 0.0756 + 0.5*0.005 = 0.0781

Nous modifions le paramètre du composant SUPERVISED LEARNING 1 (C-RT), nous introduisons la valeur $\theta$= 0.7

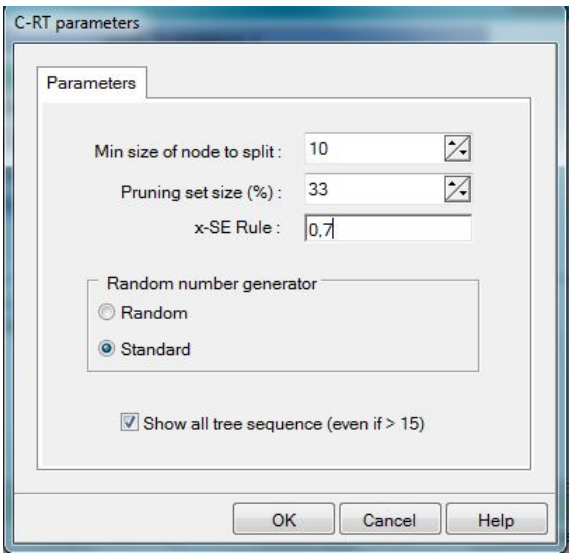



*Figure 32: paramétrage de Supervised Learing à $\theta = 0.7$*

L'arbre retenu comporte bien 4 feuilles.

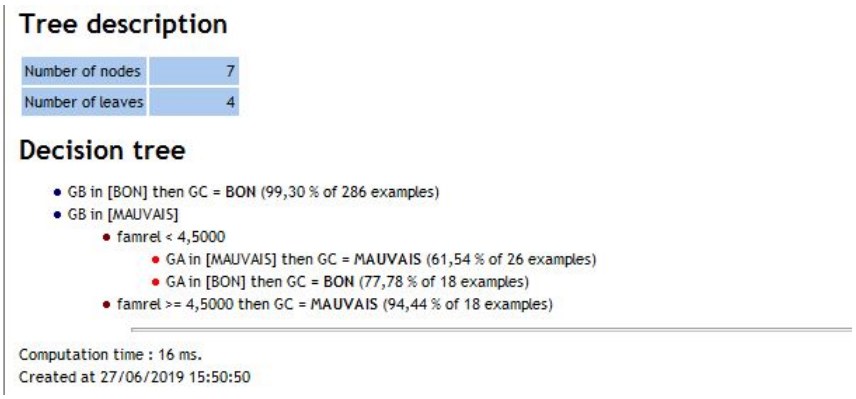



*Figure 33: Description de l'arbre avec $\theta = 0.7$*

### 6.6.7 Performances de la solution $\theta$= 0.7 (Arbre à 4 feuilles)

Il nous reste maintenant à évaluer les performances de l'arbre à 4 feuilles. Nous activons le menu VIEW du composant TEST 1, nous obtenons un taux d'erreur de 0.1163, avec un intervalle de confiance à 88% égal à [0.1312 ; 0.1398]. En comparant les différents taux d'erreur du TEST par rapport à (0, 0.7 et 1) , nous constatons que l'arbre à 4 feuilles suffit largement pour assurer un niveau de performances satisfaisant

## 6.7 Comparaison avec une autre méthode : l'algorithme K-NN [6]

Nous allons faire une étude avec la méthode K-NN(algorithme d'apprentissage supervisé) en gardant les mêmes paramètres (520 nombres d'individus) enfin de pouvoir comparer les résultats. Pour se faire, il est naturel d'utiliser une méthode de discrétisassions qui tienne compte de la variable à prédire.

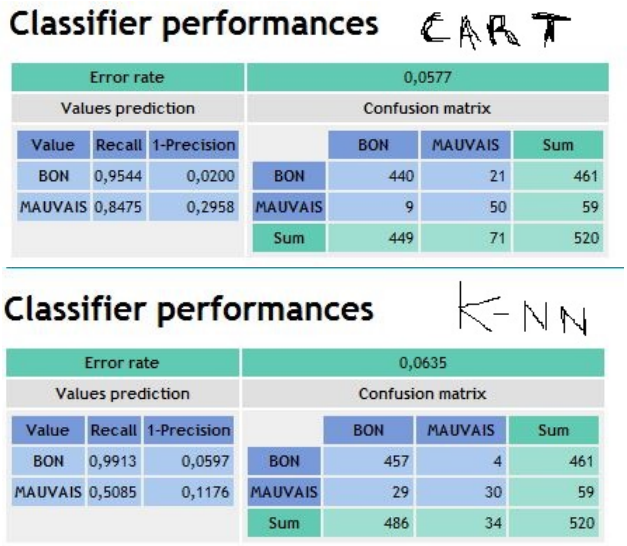



*Figure 34: Comparaison de deux méthodes CART et KNN*

Nous remarquons que le taux d'erreur au niveau du CART est de 0.0577 et celui de K-NN est 0.0635. La différence entre les deux est 0.0058 ce qui est acceptable donc en somme quelque soit l'algorithme supervisé utilisé nous devrons aboutir approximativement au même résultat.

# 7 Conclusion

Ce travail nous a permis de comprendre la structure d'un jeux de données. Parmi les techniques d'apprentissage des arbres de décision, CART est probablement celle qui détecte le mieux la bonne profondeur de l'arbre. Elle produit de ce fait, bien souvent, des modèles performants. En analysant la procédure de post-élagage, nous constatons qu'il est possible de simplifier encore l'arbre « optimal » détecté sur l'échantillon d'élagage (pruning set). L'objectif est de produire un arbre efficace avec une complexité réduite, mettant en jeu peu de variables, plus facile à manipuler et à interpréter. Il faut par ailleurs que l'on dispose de suffisamment d'observations, la partition growing/pruning participant à la fragmentation des données. En revanche, lorsque nous travaillons sur de très grands ensembles de données, CART, comme toute les méthodes s'appuyant sur le post-élagage, s'avère très gourmande en temps de calcul. En effet, l'arbre maximal élaboré lors de la phase d'expansion peut comporter un nombre invraisemblable de feuilles. Inutilement d'ailleurs puisque la très grande majorité des branches seront élaguées par la suite. Dans ce cas, surtout lors de la phase exploratoire d'appréhension des données où nous essayons avant tout de déceler assez rapidement les relations entre les variables, nous préférerons la méthode CHAID basée sur un mécanisme de pré-élagage : la technique essaie de définir une règle d'arrêt judicieuse durant l'expansion de l'arbre. Nous avons également comparer deux algorithmes dont celui de CART et K-NN, ce qui nous pousse à dire que tout type d'algorithme d'apprentissage donnera le même résultat. D'autre part, ce projet a été très important pour la mise en pratique de la théorie. Comme tout travail scientifique, ce projet est ouvert aux suggestions.

# 8 Références

[1 ] https://archive.ics.uci.edu/ml/index.php
[2 ] http://www.sharelatex.com
[3 ] https://tutoriels-data-mining.blogspot.com/2008/03/comparaison-de-classifieurs-validation.html
[4 ] http://eric.univ-lyon2.fr/ricco/tanagra/fr/tanagra.html
[5 ] http://eric.univ-lyon2.fr/ ricco/tanagra/fichiers/fr_Tanagra_Roc_Curve.pdf
[6 ] http://eric.univ-lyon2.fr/ ricco/tanagra/fr/contenu_tutoriaux_supervised_learning.html
[7 ]https://fr.wikipedia.org/wiki/Apprentissage_supervisé