# Tanagra

## Logiciels libres, spécificités et applications

Ricco RAKOTOMALALA

Université Lyon 2

Laboratoire ERIC

http://eric.univ-lyon2.fr/~ricco

# Ricco ?

Enseignant chercheur – CNU 27 – Informatique

Université Lumière Lyon 2

Culture Économétrie (Statistique)

Thèse Apprentissage automatique – Data Mining

- Arbres de décision, Sélection de variables, Échantillonnage, ...
- Applications *(classement de protéines, classement de planctons, reconnaissance de la langue, etc.)*

Développement et diffusion de logiciels libres (TANAGRA, *SIPINA*)

Rédaction et diffusion de didacticiels

Rédaction et diffusion de fascicules de cours

# Plan

1. Data Mining
2. Pourquoi le logiciel libre dans le data mining
3. Tanagra – Spécification, développement, promotion
4. Une application : classement de planctons

*Comment faire coopérer les techniques dans une seule plate-forme*

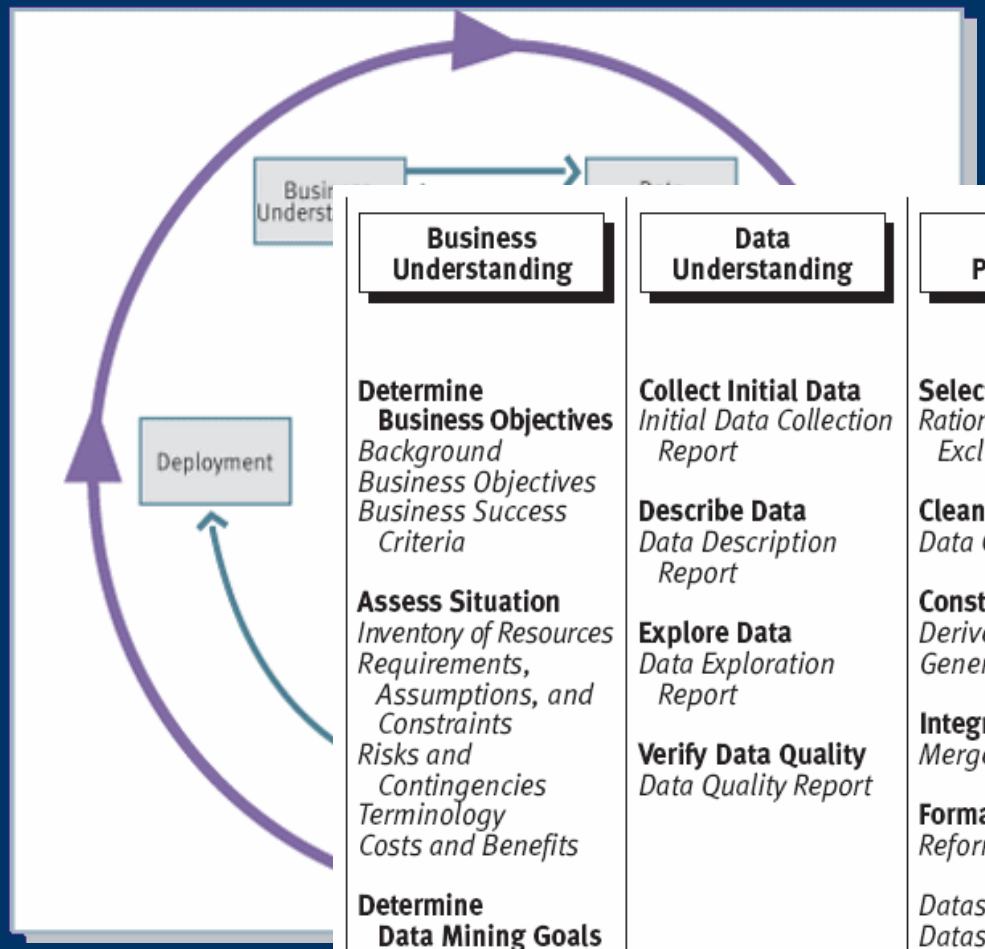
*5.* Comparaison avec les autres logiciels libres

*(Knime, Orange, R, RapidMiner, Weka,...)*

# 1. Data Mining

ECD : Extraction de connaissances à partir de données
(Knowledge Discovery in Databases)

# Data Mining




| Business Understanding | Data Understanding | Data Preparation | Modeling | Evaluation | Deployment |
|---|---|---|---|---|---|
| **Determine Business Objectives**<br>*Background*<br>*Business Objectives*<br>*Business Success Criteria* | **Collect Initial Data**<br>*Initial Data Collection Report* | **Select Data**<br>*Rationale for Inclusion/ Exclusion* | **Select Modeling Techniques**<br>*Modeling Technique*<br>*Modeling Assumptions* | **Evaluate Results**<br>*Assessment of Data Mining Results w.r.t. Business Success Criteria*<br>*Approved Models* | **Plan Deployment**<br>*Deployment Plan* |
| **Assess Situation**<br>*Inventory of Resources*<br>*Requirements, Assumptions, and Constraints*<br>*Risks and Contingencies*<br>*Terminology*<br>*Costs and Benefits* | **Describe Data**<br>*Data Description Report* | **Clean Data**<br>*Data Cleaning Report* | **Generate Test Design**<br>*Test Design* | **Review Process**<br>*Review of Process* | **Plan Monitoring and Maintenance**<br>*Monitoring and Maintenance Plan* |
| **Determine Data Mining Goals**<br>*Data Mining Goals*<br>*Data Mining Success Criteria* | **Explore Data**<br>*Data Exploration Report* | **Construct Data**<br>*Derived Attributes*<br>*Generated Records* | **Build Model**<br>*Parameter Settings*<br>*Models*<br>*Model Descriptions* | **Determine Next Steps**<br>*List of Possible Actions*<br>*Decision* | **Produce Final Report**<br>*Final Report*<br>*Final Presentation* |
| **Produce Project Plan**<br>*Project Plan*<br>*Initial Assessment of Tools and Techniques* | **Verify Data Quality**<br>*Data Quality Report* | **Integrate Data**<br>*Merged Data* | **Assess Model**<br>*Model Assessment*<br>*Revised Parameter Settings* | | **Review Project**<br>*Experience Documentation* |
| | | **Format Data**<br>*Reformatted Data* | | | |
| | | *Dataset*<br>*Dataset Description* | | | |

CRISP DM, Step-by-step Data Mining Guide, SPSS Publication

# Data Mining – Est-ce vraiment novateur ?

Définition (Fayyad, 1996) : Processus non trivial d'identification des structures inconnues, valides et potentiellement exploitables dans les bases de données.

Data Mining : Une nouvelle façon de faire de la statistique ?

http://cedric.cnam.fr/~saporta/DM.pdf

L'analyse des données est un outil pour dégager de la gangue des données le pur diamant de la véridique nature.» (J.P.Benzécri1973)

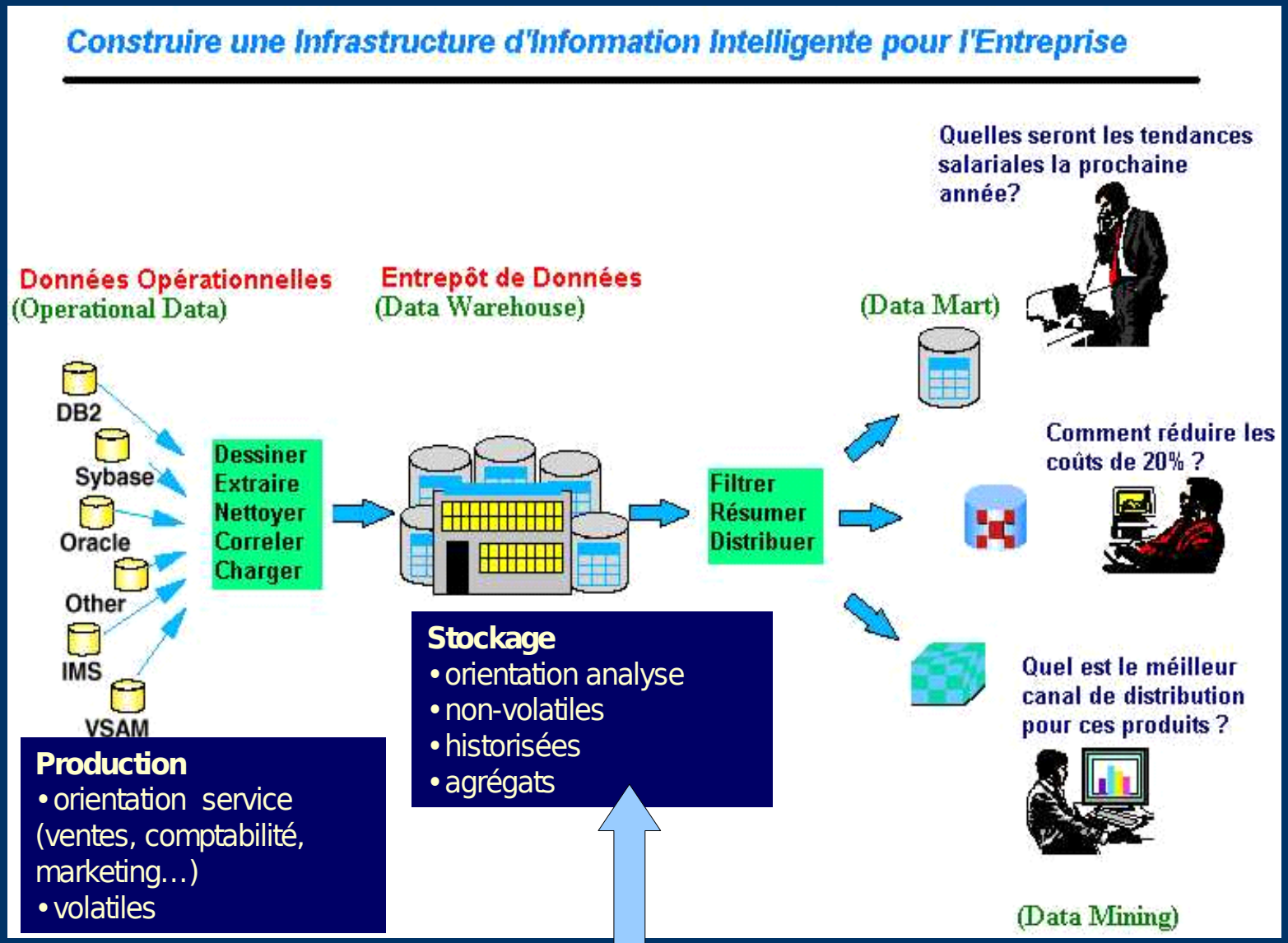
The basic steps for developing an effective process model ?

http://www.itl.nist.gov/div898/handbook/pmd/section4/pmd41.htm

1. Model selection
2. Model fitting
3. Model validation

Travailler sur des entrepôts de données

Faire partie intégrante du flux d'informations dans l'entreprise




Problème de volumétrie

Mixer des techniques d'horizons différents

Apprentissage automatique, Reconnaissance de formes, Statistique,

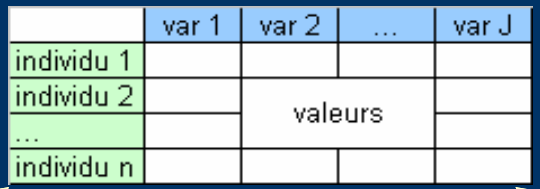
Analyse de données, ...

Statistiques
Théorie de l'estimation, tests
Économétrie

*Maximum de vraisemblance et moindres carrés*
*Régression logistique, ...*

Analyse de données
(Statistique exploratoire)
Description factorielle
Discrimination
Clustering

Méthodes géométriques, probabilités
ACP, ACM, Analyse discriminante, CAH, ...




Informatique
« Machine Learning »
Apprentissage symbolique
Reconnaissance de formes

Une étape de l'intelligence artificielle
Réseaux de neurones, algorithmes génétiques...

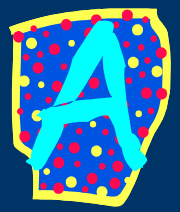
Informatique
(Base de données)
Exploration des bases de données

Volumétrie
Règles d'association, motifs fréquents, ...

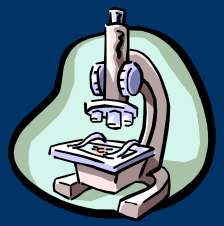
Très souvent, ces méthodes reviennent à optimiser les mêmes critères,
mais avec des approches / formulations différentes

# Traitement des données non structurées

## Textes, images, etc... autre que le simple « attribut-valeur »



Rôle fondamental de la préparation des données !



| | var 1 | var 2 | ... | var J |
|---|---|---|---|---|
| individu 1 | | | | |
| individu 2 | | valeurs | | |
| ... | | | | |
| individu n | | | | |

Prédiction
Structuration
Description
Association

<u>Les applications</u>
Filtrage automatique des e-mails (spams, ...)
Reconnaissance de la langue à une centrale téléphonique
Analyse des mammographies
Etc.

# 2. Data Mining et logiciel libre

*Attention, les informaticiens arrivent...*

## Développer des méthodes au cœur des entrepôts de données

Les B.D. sont surtout intéressés par le développement des plate-formes B.I.

Proximité très (trop) forte avec les applications industrielles (ORACLE, SQL-Server…)

Développement lourds, peu valorisants pour l' « apprentissage automatique » (publications)

*Récupération d'outils existants. Ex. intégration de WEKA dans PENTAHO...*

## Traitement des données non structurées

Trop spécifique – Impossible de développer un outil générique

Proximité des applications industrielles

## Développer des outils génériques de traitement de données

Intégrer des méthodes avec des finalités (origines) différentes

Pouvoir les faire coopérer entre elles

Tester et diffuser une nouvelle méthode publiée

Développement de plate-forme peu onéreuse, c'est le développement des algorithmes de traitements qui est difficile *(ex. RAPIDMINER et KNIME reposent en partie sur le moteur WEKA)*

# Quel public pour le logiciel libre de data mining ?
## Qui sont les utilisateurs, quels sont leurs besoins ?

### Un logiciel pour l'enseignement et le profil « utilisateur »

Les cours, explication des méthodes, outil pédagogique

Illustrer les techniques en cours, les mettre en oeuvre en TD

Sans connaissances spécifiques (langage de prog., etc.) - Former sur le fond et non la forme

Avec un niveau de qualité conforme aux « standards » du domaine

Les études « réelles » - les « dossiers » - les chercheurs des autres domaines (biologie, médecine, etc.)

### Une plate-forme pour la recherche

Plate-forme d'expérimentation pour tests à grande échelle

Implémenter ses méthodes (et les tester)

Les comparer (toutes choses égales par ailleurs i.e. dans le même environnement)

Les diffuser (pour d'autres, à des fins d'expérimentation, de comparaison)

Une publication n'est crédible que si reproductible (données, outils)

### Un outil pédagogique pour l'apprentissage de la programmation

Spécifications et conception de ce type de logiciel - Apprendre par l'exemple

Connaître les outils et les bibliothèques types

Sujets de stages pour les étudiants

# Logiciel libre de data mining

## Pourquoi (accès) libre ? Pourquoi *open source* ?

### Protéger les chercheurs, protéger les utilisateurs

A qui appartient un logiciel développé par un enseignant-chercheur ?

Est-ce le même statut que pour les ouvrages ?

Pourvoir développer sans contraintes

Pouvoir utiliser sans mauvaises surprises

### Diffusion du logiciel = valider les publications

Logiciels accessibles à tous → Comparaison et vérification des résultats

Reproduire « exactement » les expérimentations

### Comparer le code = comparer les implémentations

Comparer les interprétations d'un même problème (ex. Relieff WEKA)

Lecture du code par d'autres chercheurs (ex. Naive Bayes classifier)

Optimiser le code avec différentes versions

### Outil ouvert = Outil vivant

Introduire ses propres algorithmes

Discuter sur la base de prototypes et d'évolutions

Monter et partager des bibliothèques types (ex. générateurs aléatoires, fonctions de répartitions, les fameux packages...)

# Logiciel de data mining
## Quelles fonctionnalités implémenter

### Accès et préparation des données
Accéder à un fichier / une BD
Rassembler des sources différentes

### Méthodes de Fouille de données
Lancer les calculs avec différents algorithmes
Bibliothèque de méthodes

### Enchaîner les traitements
Faire coopérer les méthodes sans programmer

### Évaluer les connaissances
Validation croisée, etc.

### Exploiter les sorties
Rapports, visualisation interactive, etc.

### Appliquer/exploiter les modèles
Modèles en XML (PMML), code C, DLL compilées
Prédiction directe sur de nouveaux fichiers

Logiciels commerciaux
Prototypes de recherche

# Logiciel de data mining
## Quel mode opératoire ?

Logiciels pilotés par menu (STATISTICA, OPEN STAT, SIPINA, ...)

(+) Organisation de type « tableur »

(+) Rapidité de prise en main

(-) Enchaînement « à la main » des traitements

(-) Pas de trace des opérations effectuées

(-) Et donc reproductibilité difficile des traitements

Ligne de commande (SAS, S-PLUS, R, ...)

(+) Souplesse et puissance de la programmation

(+) Sauvegarde des traitements, reproductilibité

(-) Apprentissage d'un langage

Filière (diagramme de traitements) – Estampillé « Data Mining »

(+) Programmation « visuelle » - Pas d'apprentissage

(+) Enchaînement des traitements

(+) Sauvegarde des traitements, reproductilibité

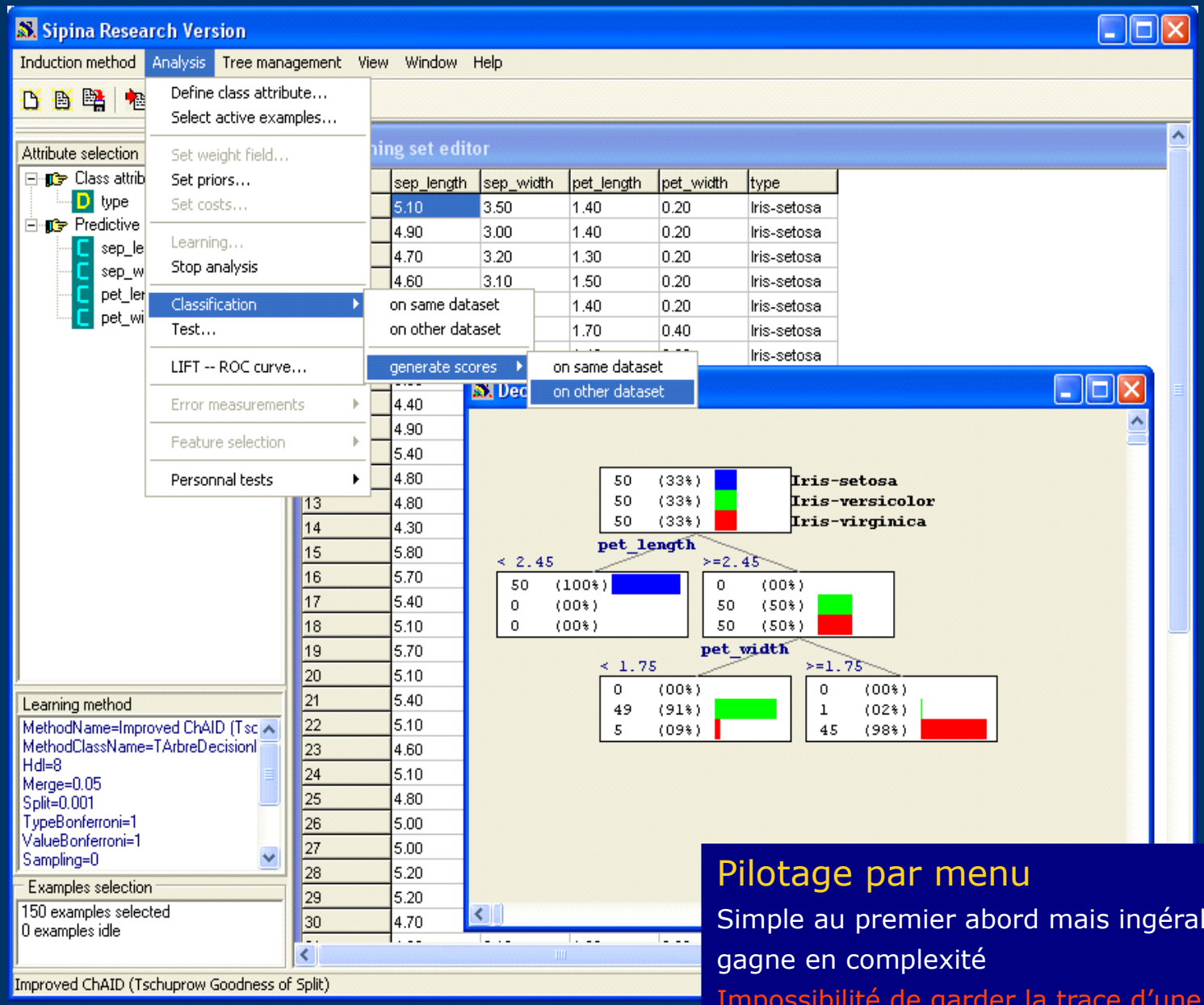
(-) Pas la puissance d'un « vrai » langage de programmation

SPAD, SAS Enterprise Miner, SPSS Clementine, S-PLUS Insightfull Miner, STATISTICA Data Miner, ...

KNIME, ORANGE, RAPIDMINER, TANAGRA, WEKA

# Exemple de pilotage par menu

## Sipina




## Pilotage par menu

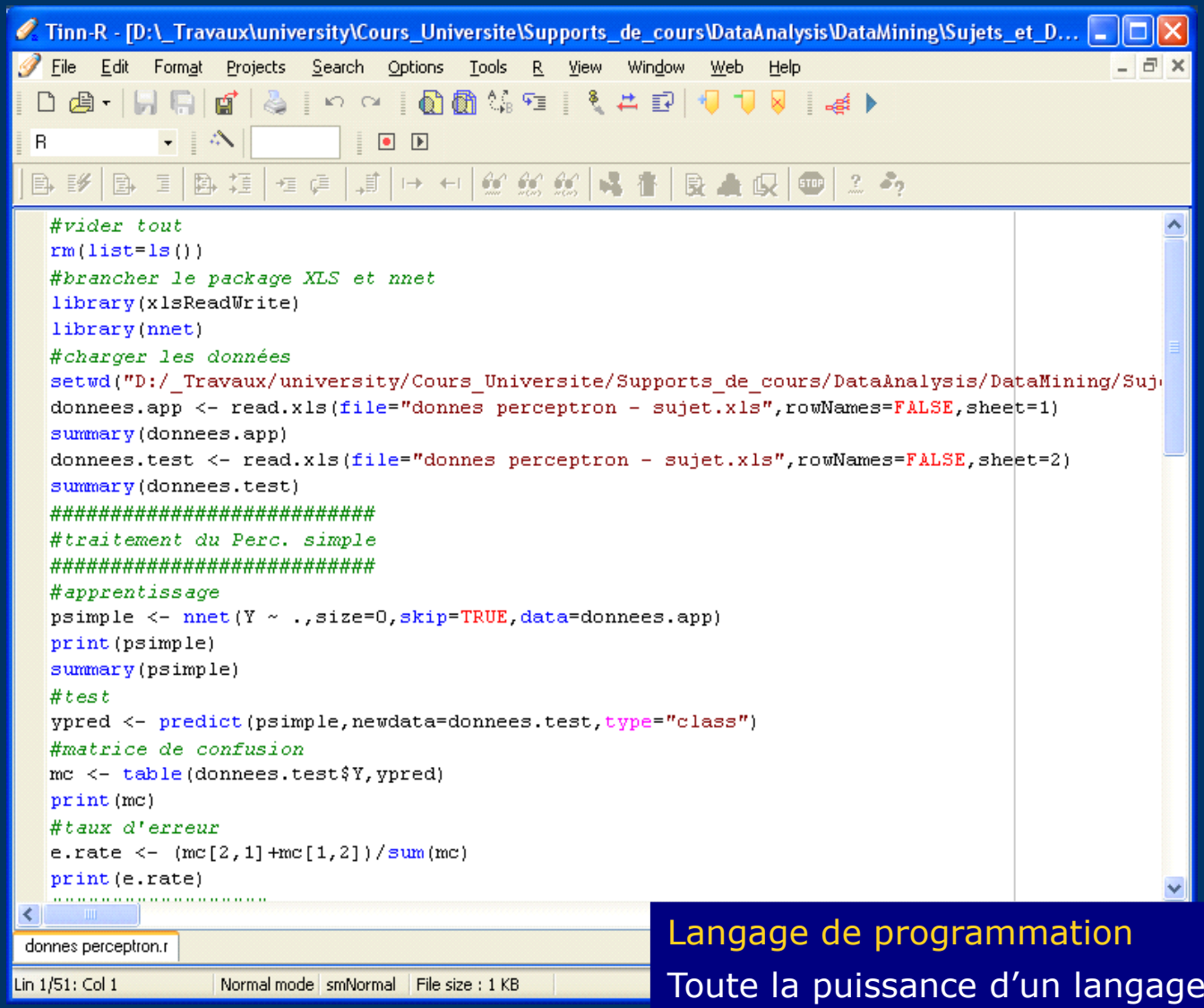
Simple au premier abord mais ingérable dès que le logiciel gagne en complexité

Impossibilité de garder la trace d'une analyse complète et donc de la reproduire

Exige une documentation complète et constamment à jour (Open Stat & Stat 4U sont dans la même situation)

# Exemple de ligne de commande + langage de programmation

## R




Langage de programmation

Toute la puissance d'un langage de programmation

L'accès au langage est une barrière à l'entrée qui rebute certains

# Exemple de diagramme de traitements

## Tanagra

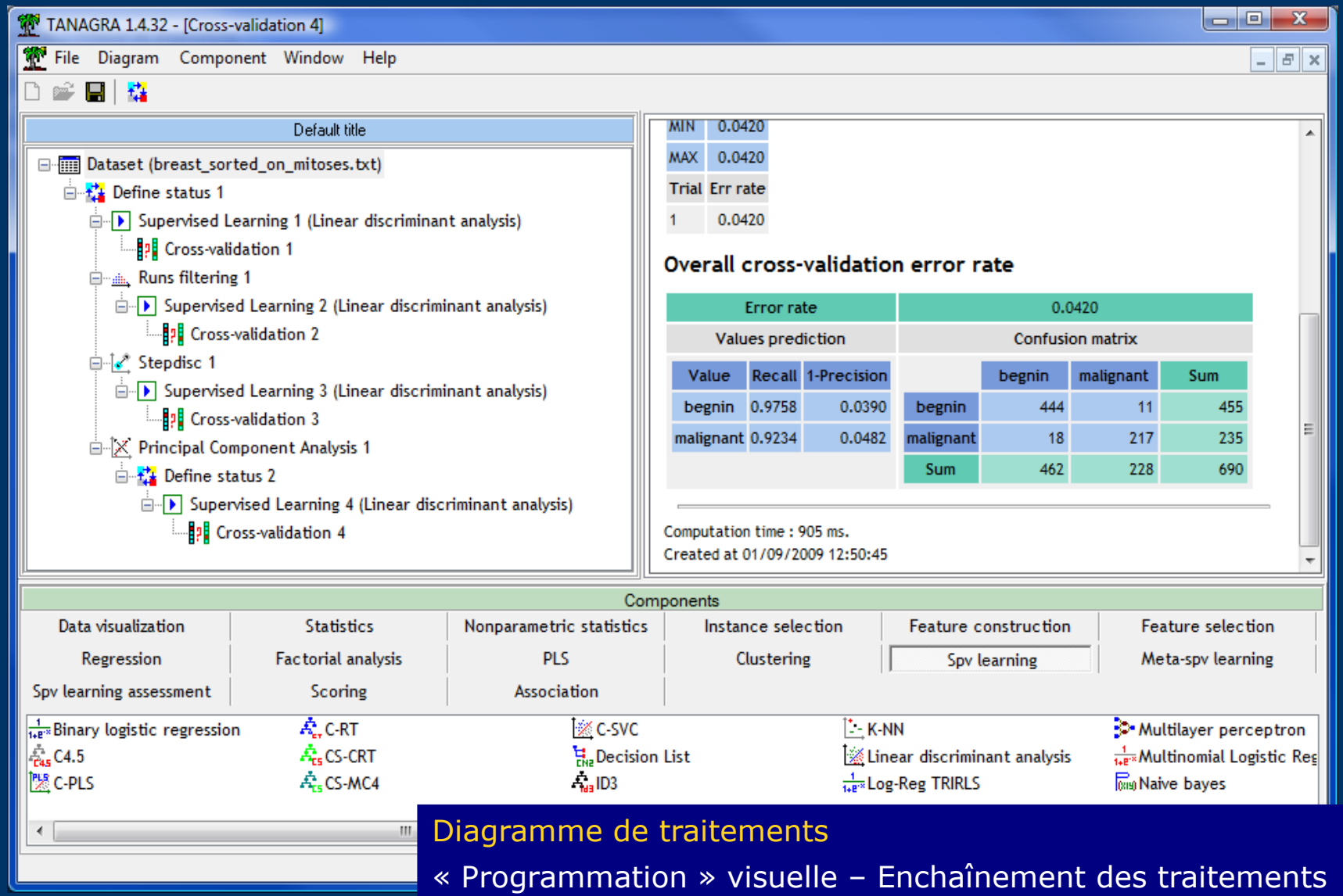



Diagramme de traitements

« Programmation » visuelle – Enchaînement des traitements

*Mais pas toutes les fonctionnalités d'un langage de programmation*

Mise à jour facilitée par adjonction de composants

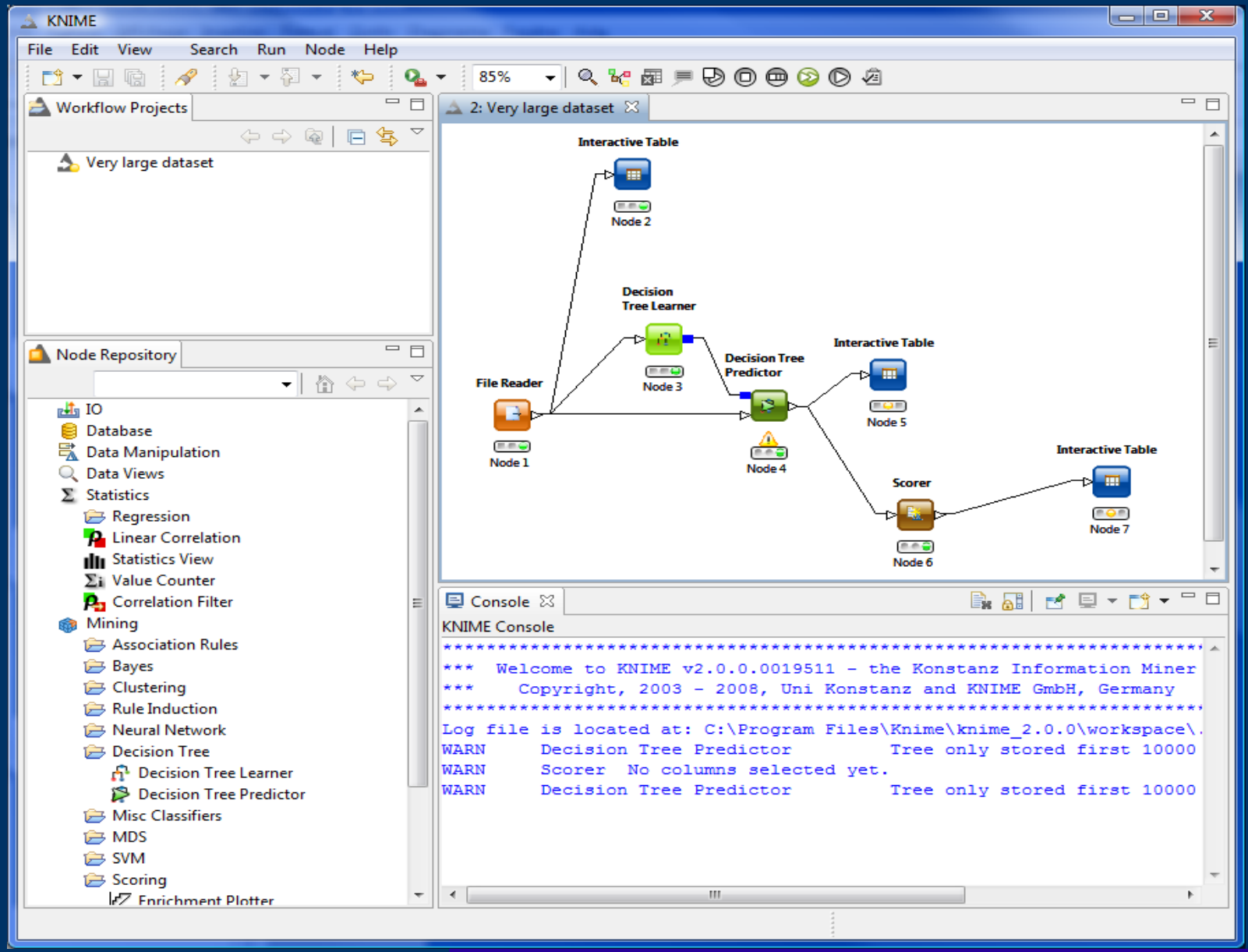
Garder une trace de l'analyse et pouvoir la sauvegarder

Possibilité de fragmenter la documentation par « composants »

C'est le standard actuel

# Exemple de diagramme de traitements

Knime




## Diagramme de traitements

Une autre manière de présenter les diagrammes de traitements

# 3. Tanagra

# Tanagra

Définir un cahier des charges aussi précis que possible

## Miser sur la simplicité d'utilisation

Installation simplifiée – Pas de serveurs lourds à installer

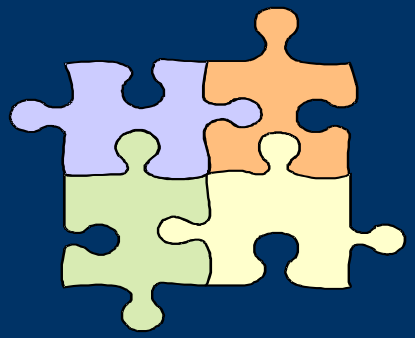
Gestion simplifiée des données - Format texte et accès au format tableur

Fonctionnement par diagramme de traitements

Couvrir les statistiques, l'analyse de données et le data mining. De manière unifiée.

Résultats lisibles, en adéquation avec les « standards »

Interfaçage avec les tableurs (Excel, Open Office Calc)



## Mettre définitivement de côté les aspects « professionnels »

Interfaçage fort avec les SGBD

Déploiement et mise en production des résultats

Reporting dynamique et performant

Exploration graphique évoluée et interactive des données

## Simplicité également pour le programmeur

Simplifier à l'extrême le code permettant d'ajouter une nouvelle méthode d'analyse

Minimiser le code dédié à la gestion des données et de l'interface

Pouvoir intégrer facilement n'importe quelle technique traitant des tableaux « individus x variables »

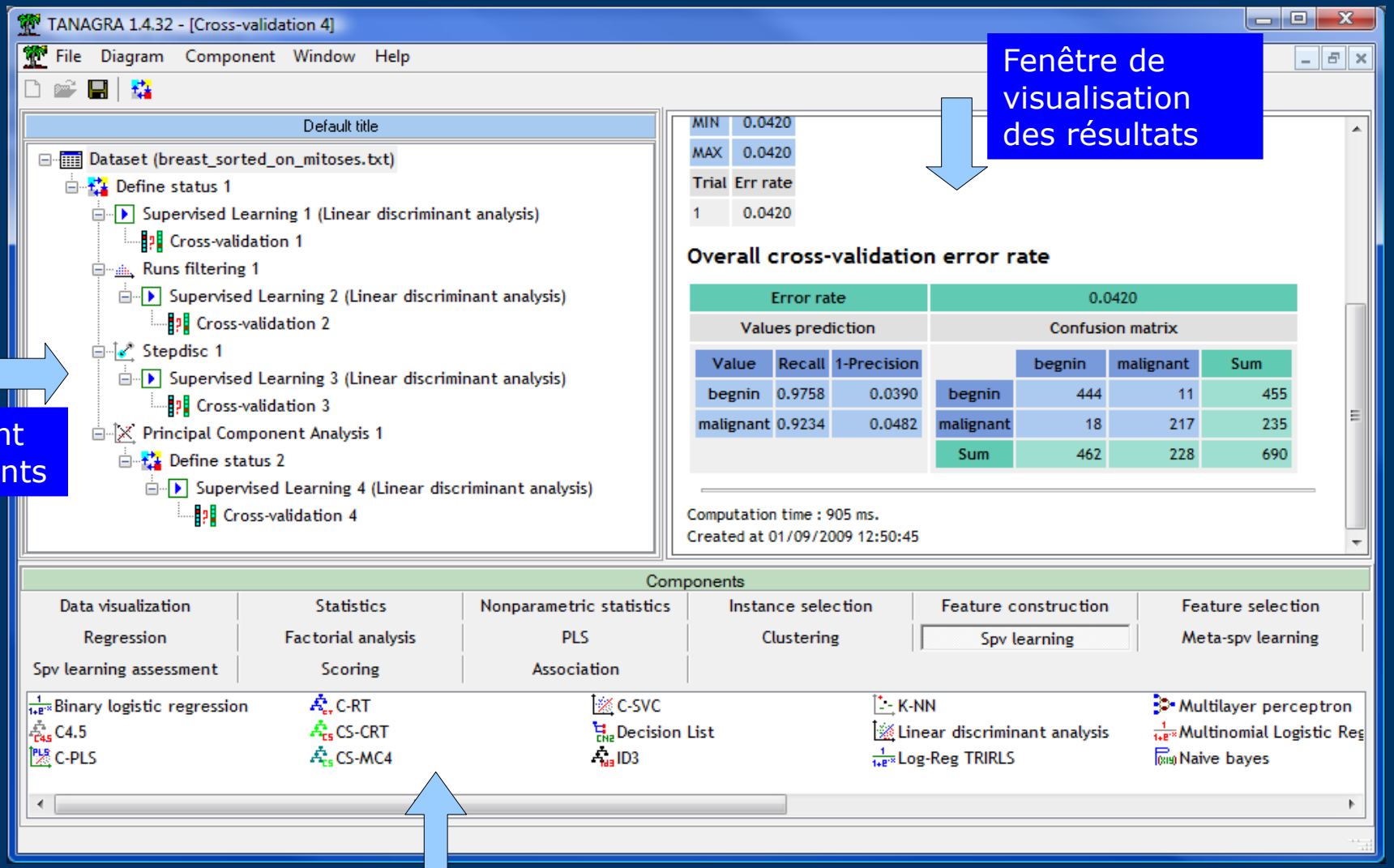
# Simplicité pour les utilisateurs
## Installation simplifiée et automatisée

### Tout doit être automatisé
L'utilisateur ne doit jamais avoir à intervenir à l'installation
Attention aux bibliothèques externes (SGBD, TCL/TK, PYTHON, etc.)
Choisir la configuration au pire cas

### Réduire les bibliothèques externes
Bibliothèque externe compilée = dépendance accrue
Bibliothèque payante = pieds et poings liés (y compris sur les architectures)
Miser sur des versions stables et sources libres
Attention à la gestion des mises à jour

### Mettre des exemples de traitements
L'utilisateur lance toujours « pour voir » sans lire la documentation

# Simplicité pour les utilisateurs
## Définir les traitements

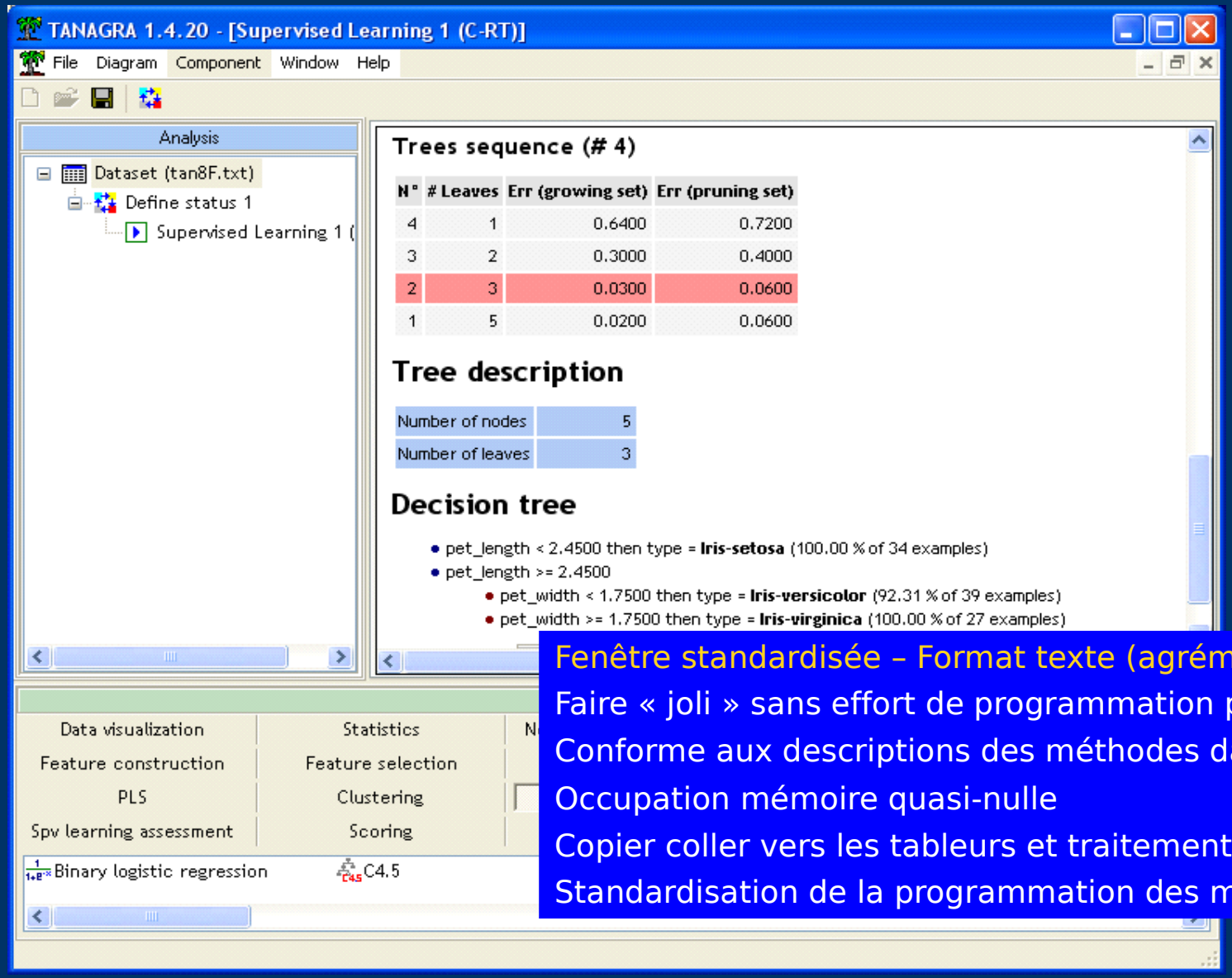



Fenêtre de visualisation des résultats

Enchaînement des traitements

Composants de calcul
Méthodes de data mining

# Simplicité pour les utilisateurs
## Standardisation des affichages




Fenêtre standardisée – Format texte (agrémenté de HTML)

Faire « joli » sans effort de programmation particulier

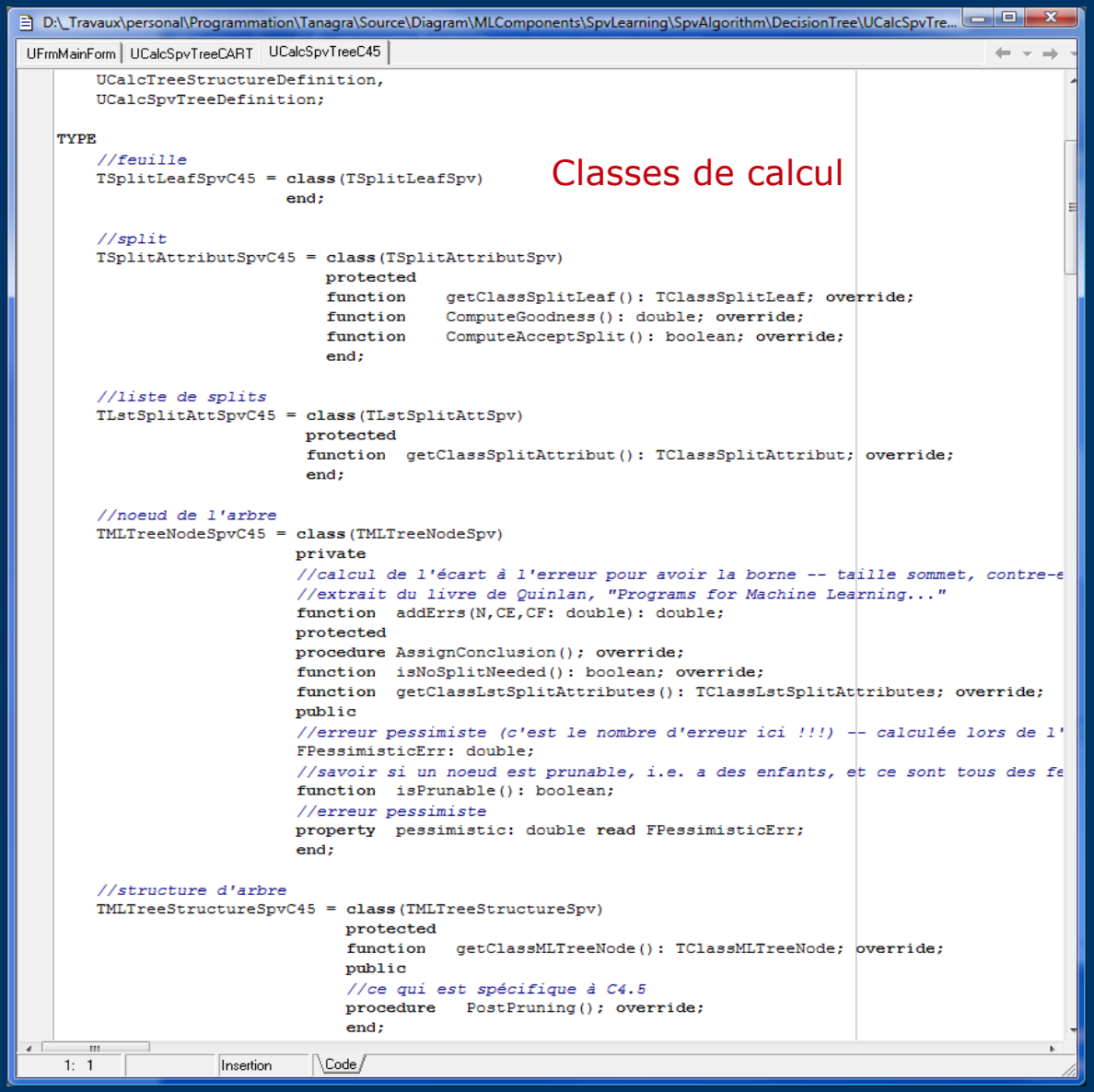
Conforme aux descriptions des méthodes dans les ouvrages

Occupation mémoire quasi-nulle

Copier coller vers les tableurs et traitement de texte

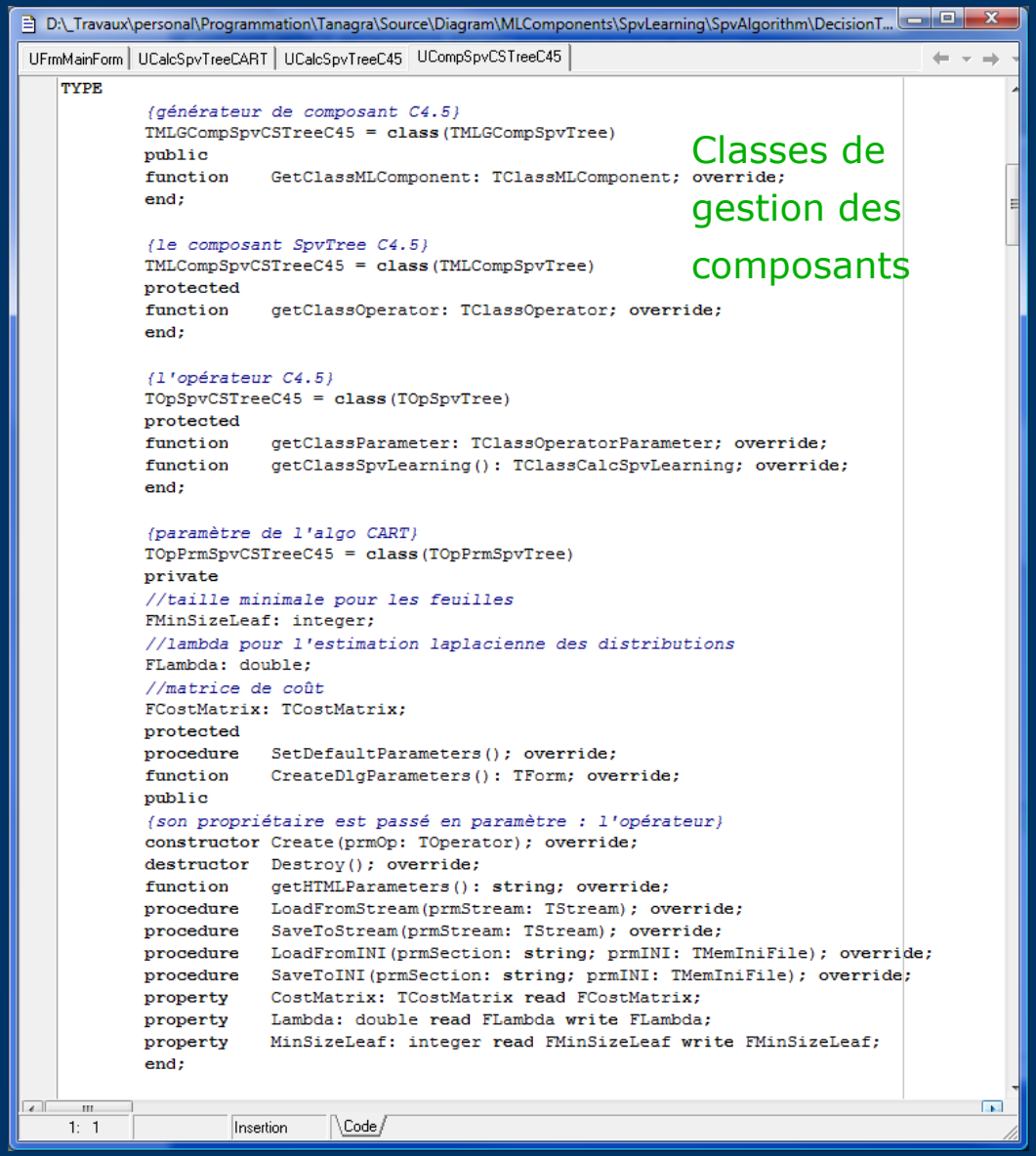
Standardisation de la programmation des méthodes

# Simplicité pour les programmeurs
## Vive la programmation objet (1/3)




```
    UCalcTreeStructureDefinition,
    UCalcSpvTreeDefinition;

TYPE
    //feuille
    TSplitLeafSpvC45 = class(TSplitLeafSpv)
                       end;

    //split
    TSplitAttributSpvC45 = class(TSplitAttributSpv)
                           protected
                           function    getClassSplitLeaf(): TClassSplitLeaf; override;
                           function    ComputeGoodness(): double; override;
                           function    ComputeAcceptSplit(): boolean; override;
                           end;

    //liste de splits
    TLstSplitAttSpvC45 = class(TLstSplitAttSpv)
                         protected
                         function  getClassSplitAttribut(): TClassSplitAttribut; override;
                         end;

    //noeud de l'arbre
    TMLTreeNodeSpvC45 = class(TMLTreeNodeSpv)
                        private
                        //calcul de l'écart à l'erreur pour avoir la borne -- taille sommet, contre-e
                        //extrait du livre de Quinlan, "Programs for Machine Learning..."
                        function  addErrs(N,CE,CF: double): double;
                        protected
                        procedure AssignConclusion(); override;
                        function  isNoSplitNeeded(): boolean; override;
                        function  getClassLstSplitAttributes(): TClassLstSplitAttributes; override;
                        public
                        //erreur pessimiste (c'est le nombre d'erreur ici !!!) -- calculée lors de l'
                        FPessimisticErr: double;
                        //savoir si un noeud est prunable, i.e. a des enfants, et ce sont tous des fe
                        function  isPrunable(): boolean;
                        //erreur pessimiste
                        property  pessimistic: double read FPessimisticErr;
                        end;

    //structure d'arbre
    TMLTreeStructureSpvC45 = class(TMLTreeStructureSpv)
                             protected
                             function   getClassMLTreeNode(): TClassMLTreeNode; override;
                             public
                             //ce qui est spécifique à C4.5
                             procedure   PostPruning(); override;
                             end;
```

# Simplicité pour les programmeurs
## Vive la programmation objet (2/3)




Classes de gestion des composants

# Simplicité pour les programmeurs
## Vive la programmation objet (3/3)

D:\Temp\Exe\tanagra_components.xml

```
<component class_name="TMLGCompSpvTreeC45">
        <name>
        C4.5
        </name>

        <bitmap>
        MLSpvTreeC45.bmp
        </bitmap>

        <description>
        Quinlan (1993), decision tree algorithm.
        </description>

        <precondition>
        At least one discrete attribute (TARGET) and one or more discrete/continuous attributes (INPUT) must be av
        </precondition>

        <target-attributes>
        Discrete class attribute.
        </target-attributes>

        <input-attributes>
        One or more discrete/continuous attributes.
        </input-attributes>

        <postcondition>
        A new column (discrete attribute) is added, it corresponds to the predicted values of the class attribute.
        </postcondition>

</component>
```

Fichier externe de gestion des composants pour les versions spécialisées [*et aussi au cas où on passait par une gestion par plug-ins*]

→ l'adjonction d'un composant est très peu contraignante

# Simplicité pour les programmeurs

## Encore plus loin dans la modularité : les plugins

### La solution idéale ?

L'application mère est une matrice qui gère et transmet les données

Les techniques sont des procédures programmées sous forme de bibliothèques externes

### Mais des contraintes fortes

Organisation ultra-rigoureuse des protocoles

→ Passage des informations et des données

→ Affichage des résultats

→ Documentation (fichier d'aide)

### Bref…

Souvent rédhibitoire, alors que l'objectif était d'offrir un outil modulaire

Intéressant si plugins = procédures de calculs qui renvoient des objets standardisés

Et qu'une vraie équipe organise la vie autour du logiciel

→ Le logiciel R est le seul à avoir su le faire

## Spécifications

Outil libre (*ça coûte moins cher*)

Largement diffusé (*pour avoir des programmeurs*)

Avec une large bibliothèque de classes (*calculs, conteneurs, etc.*)

Qui permet de faire des interfaces agréables, simplement, rapidement

## Pourquoi DELPHI pour Tanagra ?

A l'époque, DELPHI 6.0 PERSO était gratuite

Cours de DELPHI en L3 et M1 dans le département « Informatique – Statistique »

Accès aux anciennes bibliothèques de calculs, validées depuis longtemps déjà

Connaissance étendue des bibliothèques libres (Turbo Power, etc.)

Permet de faire des interfaces agréables, simplement, rapidement

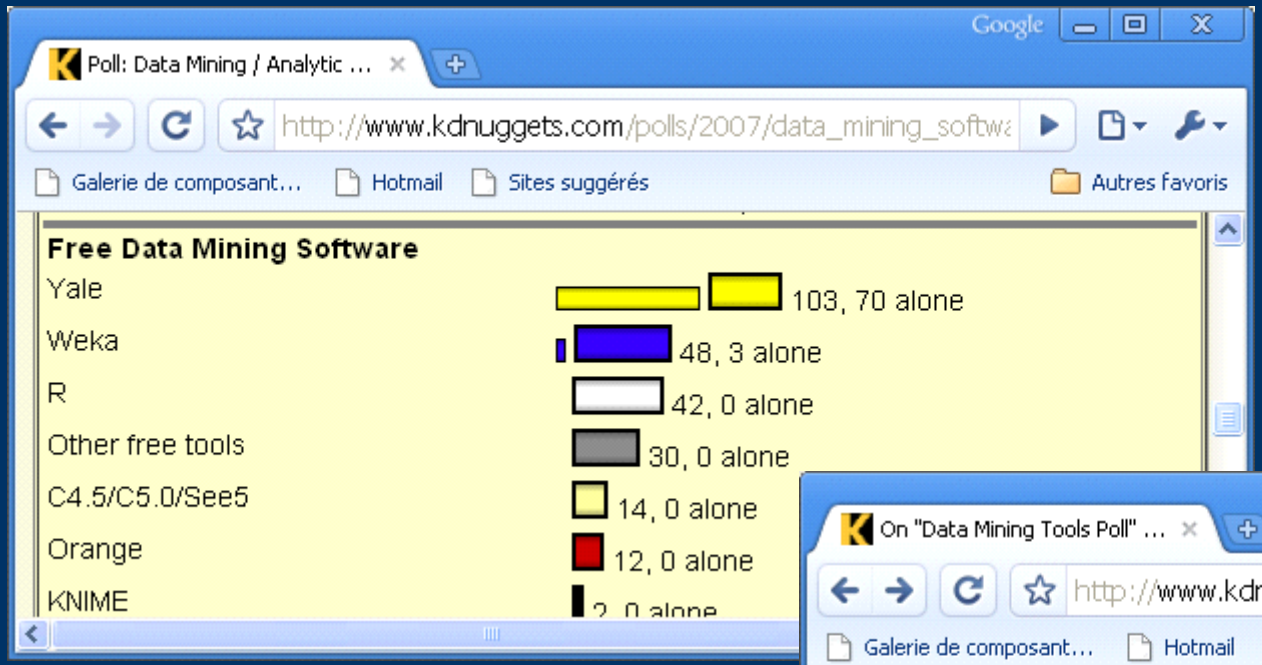
Affinités personnelles...

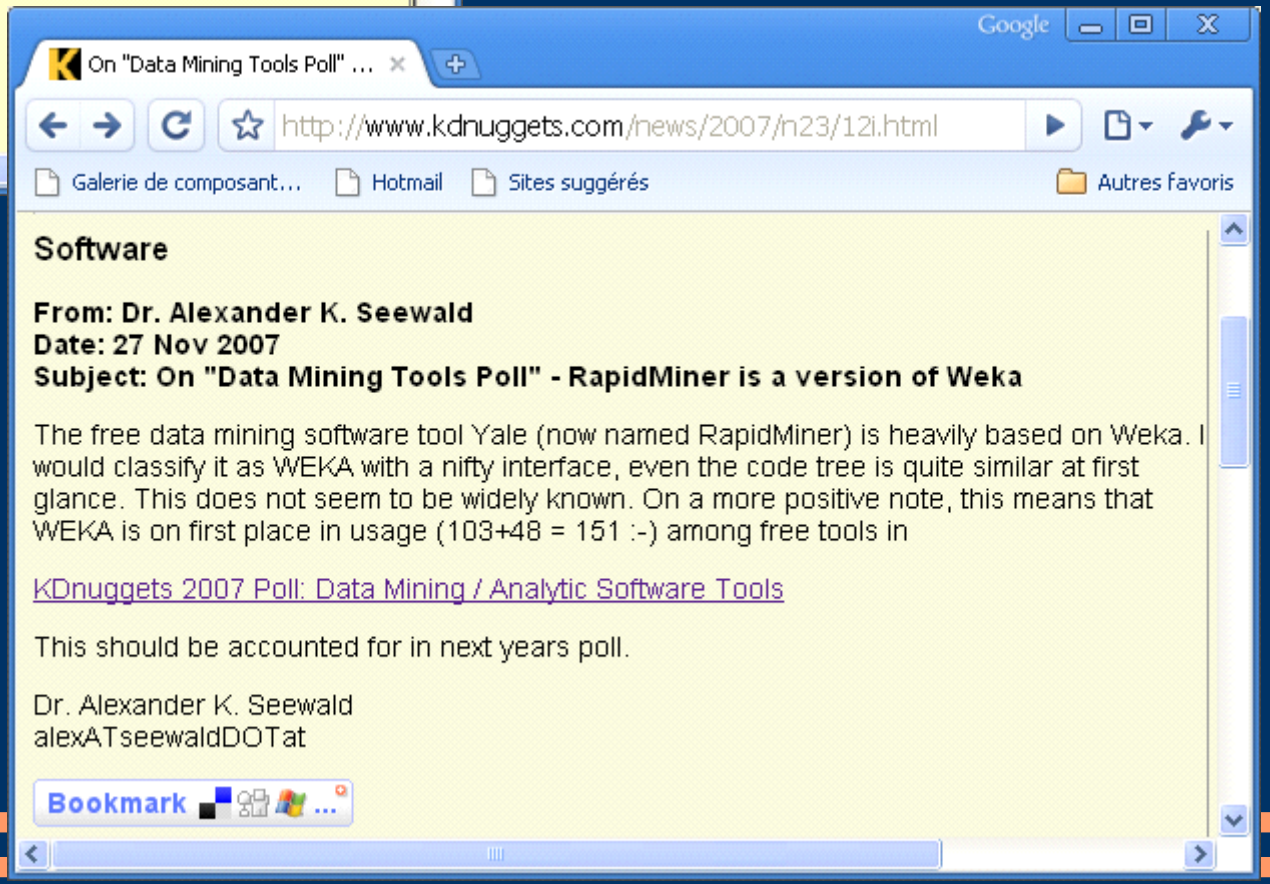
J'aurais du le faire en JAVA ? L'écueil WEKA →

# Implémentation

## Pourquoi ne pas avoir intégré des bibliothèques de calcul existantes ?

Sondage : quel logiciel utilisez vous en 2007 ?



# Promotion

Comment faire connaître le logiciel sans tomber dans le « commercial »

## Écrire un article de référence

Voilà toujours une publication de plus

Marquer le coup en annonçant le logiciel

C’est la référence que citeront les utilisateurs

## Documenter le logiciel

Documenter les méthodes : description théorique

Documenter leur mise en oeuvre : les tutoriels

Facilitée par le découpage en « composants » du logiciel

## Monter un site web attrayant (attractif)

La visibilité internet est primordiale

Le téléchargement du logiciel n’est pas le seul enjeu

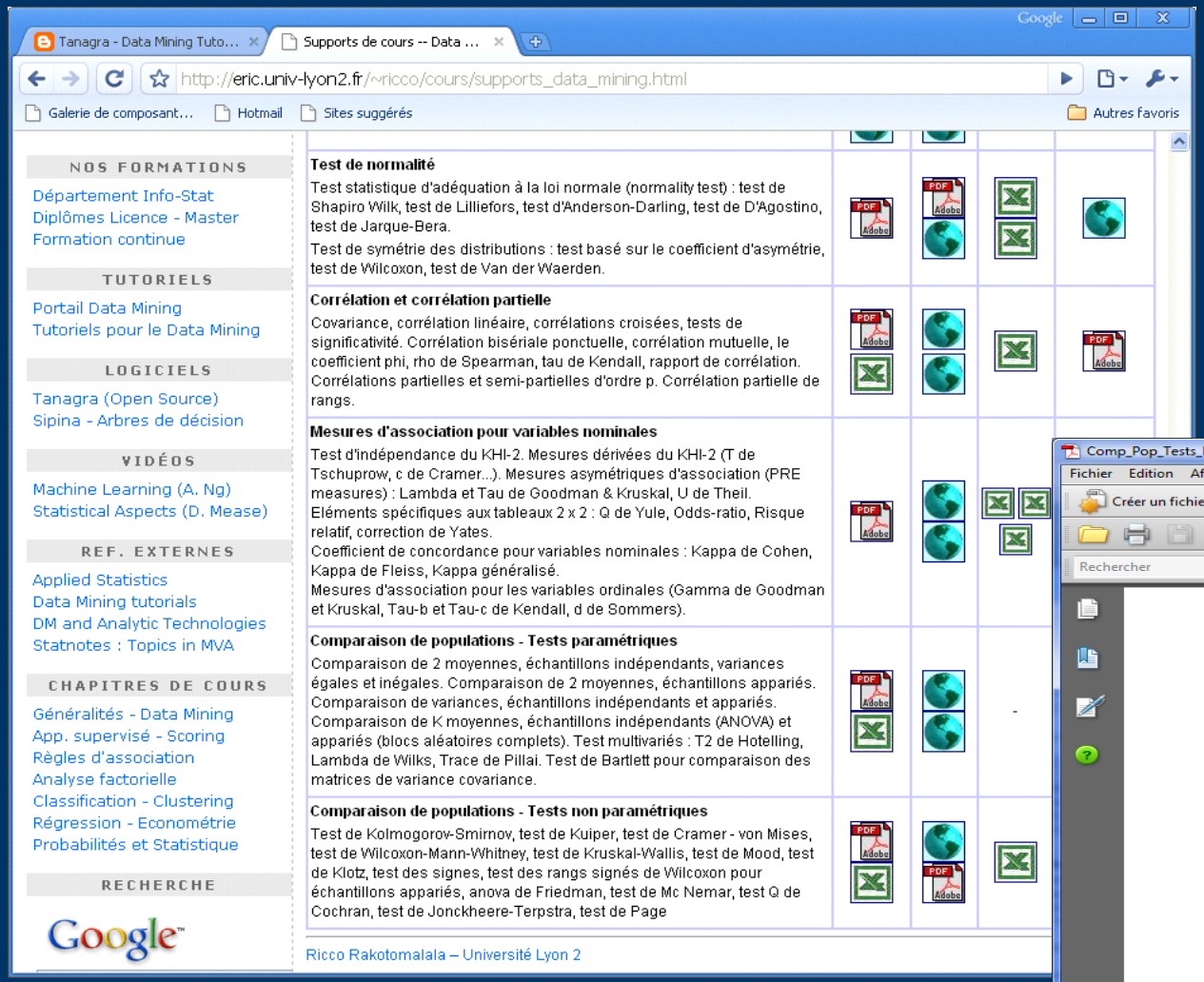
…et la promotion dans les conférences

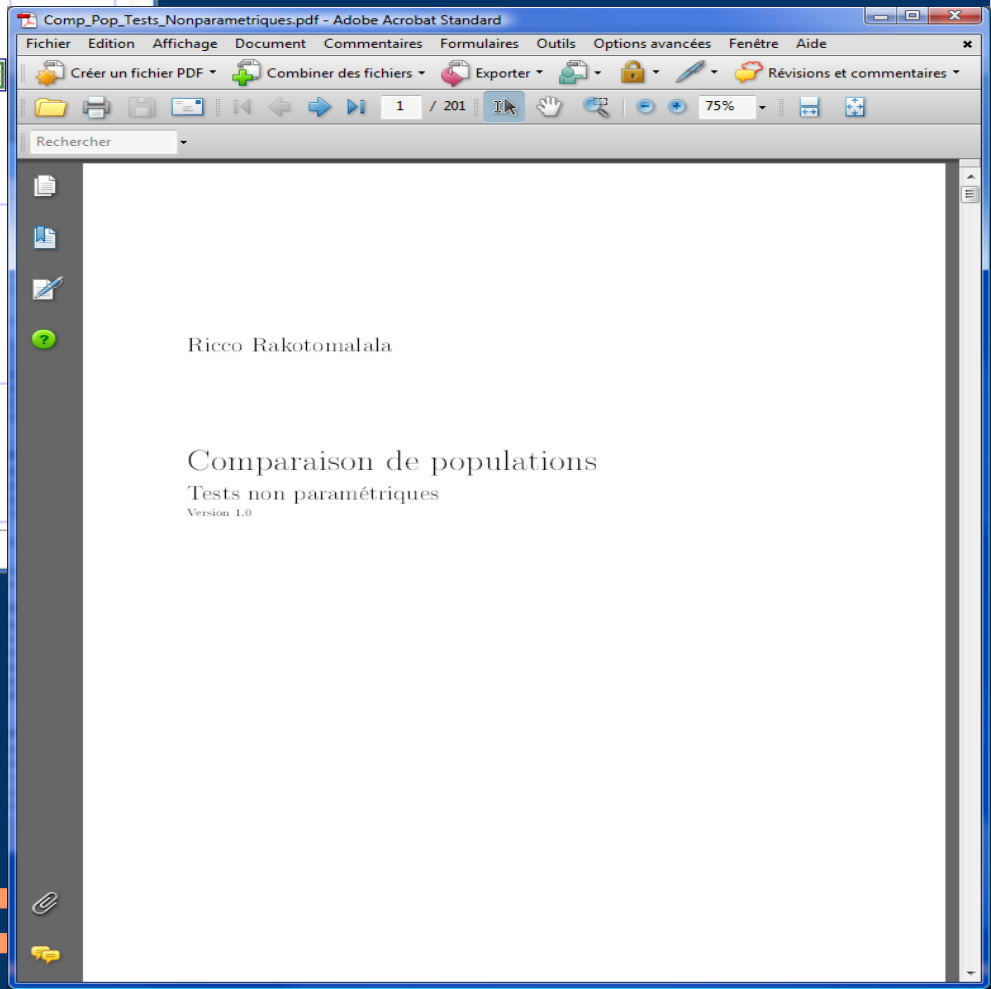
*Ateliers, démonstrations, contacts chercheurs, mailing- list, etc.*

# Promotion
## Le site web Tanagra

# Promotion
## Documentation des méthodes – Pointeurs vers les ressources

# Promotion
## Documentation des méthodes – Écrire et diffuser des supports libres

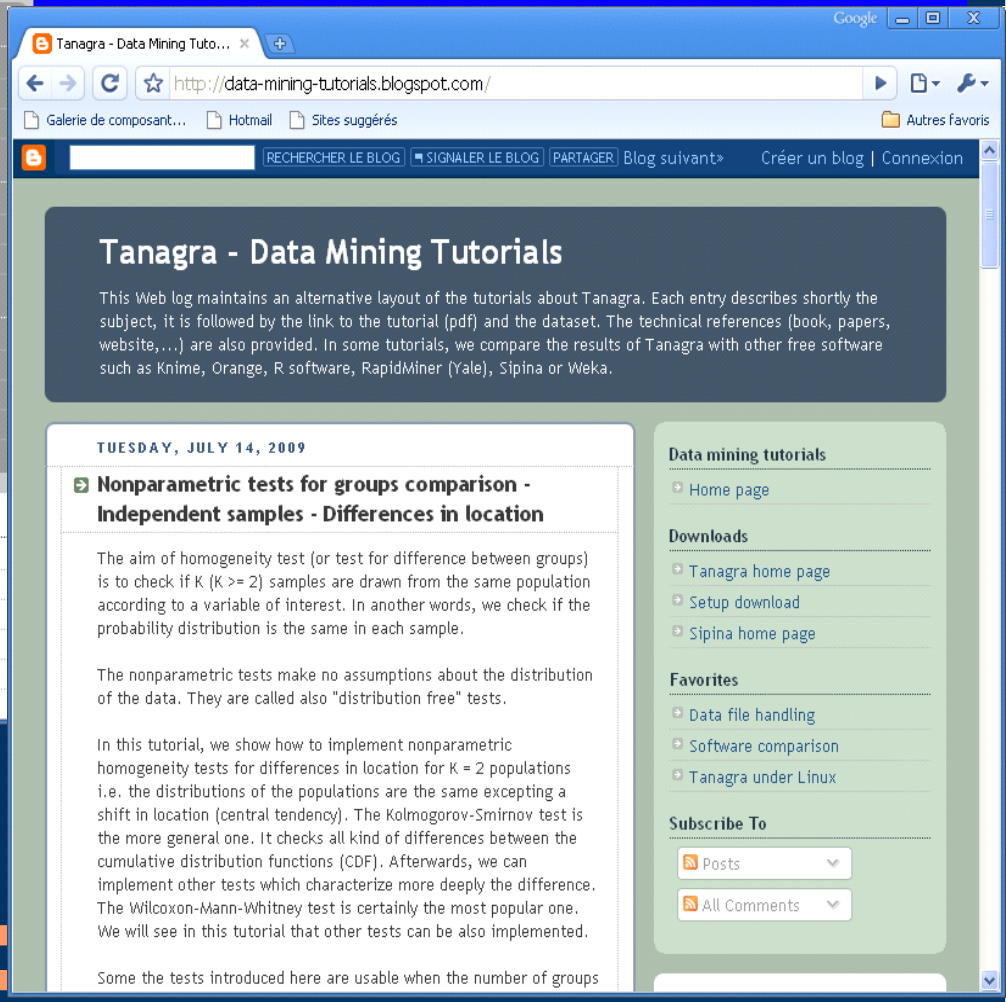


# Promotion
## Documenter la mise en œuvre des méthodes – Les tutoriels

~130 tutoriels en français à ce jour (09/2009)


~90 tutoriels en anglais à ce jour (09/2009)




Écriture du cahier des charges

Janvier 2003, plusieurs prototypes de janvier à juin 2003

Début du développement

Juillet 2003

Création du site web et mise en ligne

Janvier 2004 (~25 visiteurs par jour sur 2004)

Techniques implémentées (version 1.4.32 – Sept. 2009)

164 méthodes stat., analyse de données, data mining

Documentation libre en ligne (Sept. 2009)

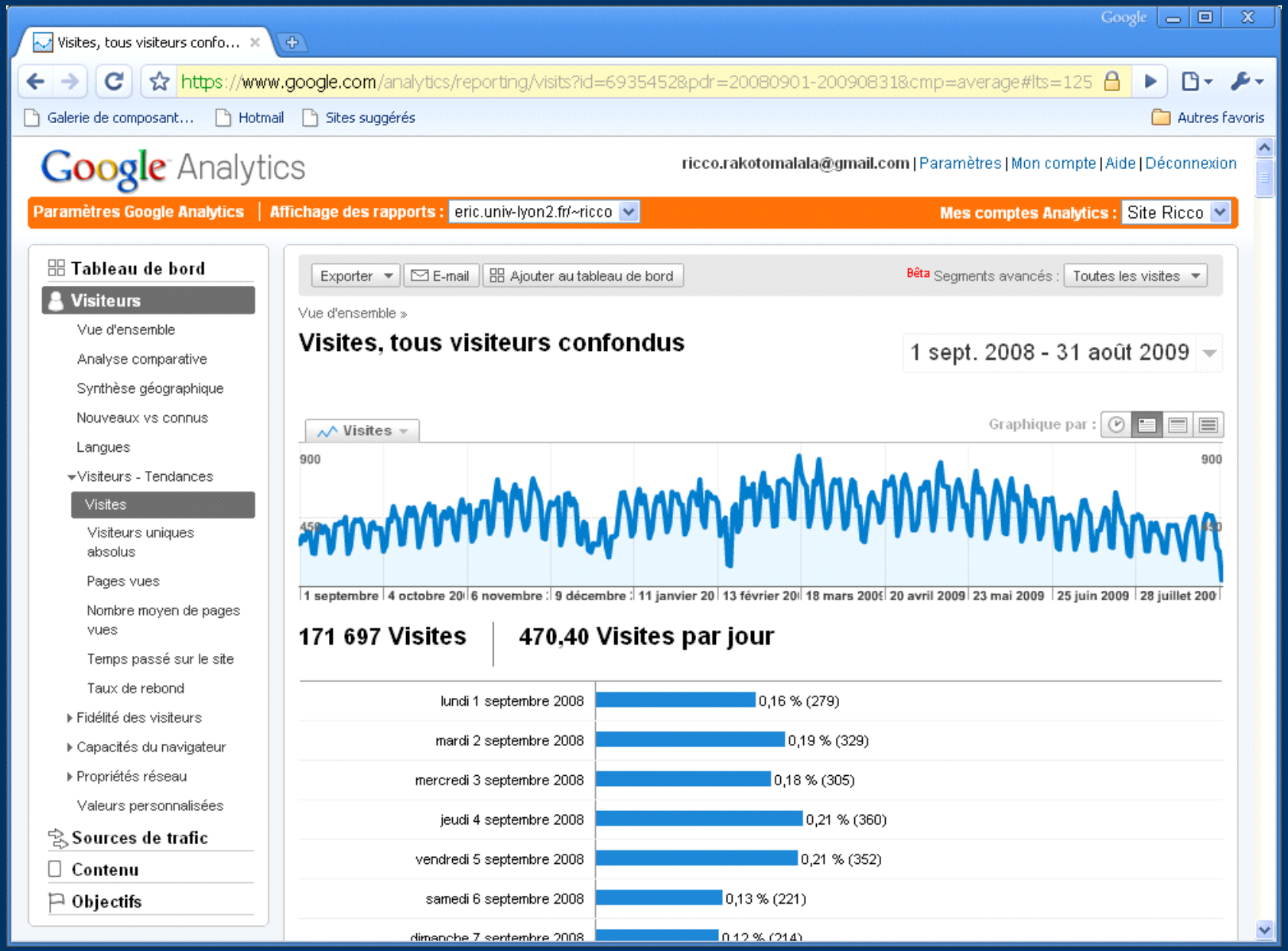
7 ouvrages libres en PDF

30 « slides » en PDF

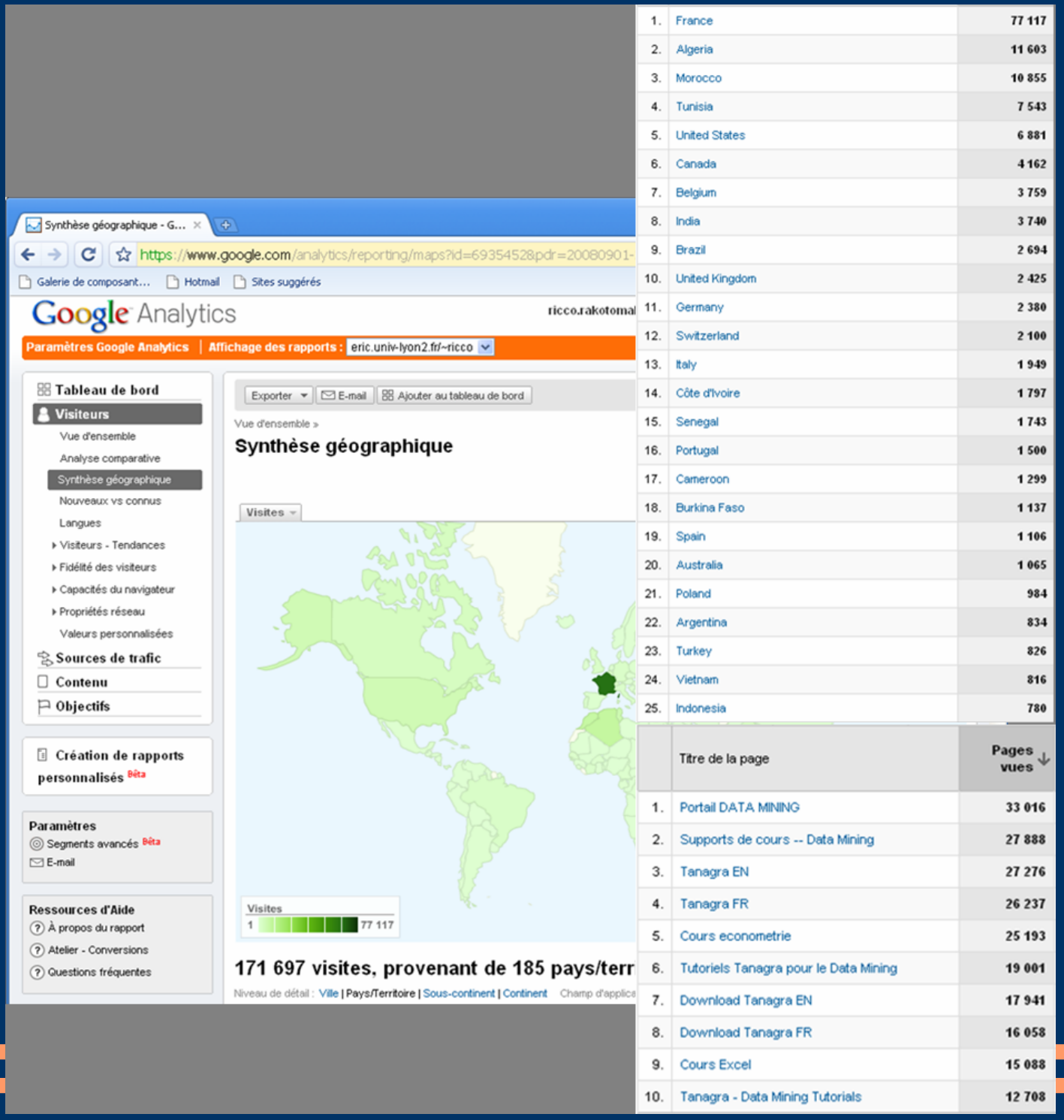
130 didacticiels en français

90 didacticiels en anglais

# Tanagra
## Bilan (2) – Diffusion 1/2



# Tanagra
## Bilan (2) – Diffusion 2/2

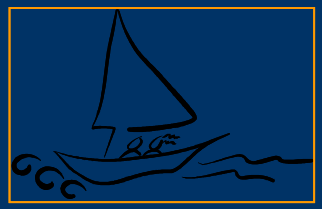



| | | |
|---|---|---|
| 1. | France | 77 117 |
| 2. | Algeria | 11 603 |
| 3. | Morocco | 10 855 |
| 4. | Tunisia | 7 543 |
| 5. | United States | 6 881 |
| 6. | Canada | 4 162 |
| 7. | Belgium | 3 759 |
| 8. | India | 3 740 |
| 9. | Brazil | 2 694 |
| 10. | United Kingdom | 2 425 |
| 11. | Germany | 2 380 |
| 12. | Switzerland | 2 100 |
| 13. | Italy | 1 949 |
| 14. | Côte d'Ivoire | 1 797 |
| 15. | Senegal | 1 743 |
| 16. | Portugal | 1 500 |
| 17. | Cameroon | 1 299 |
| 18. | Burkina Faso | 1 137 |
| 19. | Spain | 1 106 |
| 20. | Australia | 1 065 |
| 21. | Poland | 984 |
| 22. | Argentina | 834 |
| 23. | Turkey | 826 |
| 24. | Vietnam | 816 |
| 25. | Indonesia | 780 |

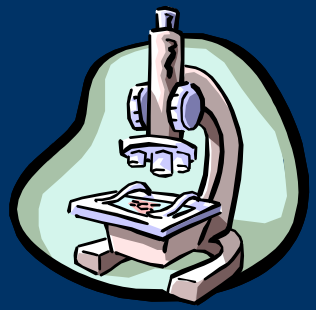
| | Titre de la page | Pages vues |
|---|---|---|
| 1. | Portail DATA MINING | 33 016 |
| 2. | Supports de cours -- Data Mining | 27 888 |
| 3. | Tanagra EN | 27 276 |
| 4. | Tanagra FR | 26 237 |
| 5. | Cours econometrie | 25 193 |
| 6. | Tutoriels Tanagra pour le Data Mining | 19 001 |
| 7. | Download Tanagra EN | 17 941 |
| 8. | Download Tanagra FR | 16 058 |
| 9. | Cours Excel | 15 088 |
| 10. | Tanagra - Data Mining Tutorials | 12 708 |

# 4. Classement automatique de planctons

# Le projet ZOOSCAN
## Récupération des données



Campagne de pêche



Les prélèvements sont scannés : ZOOSCAN

Individu = Une image de plancton

L'expert étiquette manuellement les objets

# Le projet ZOOSCAN
## Construction des descripteurs

Image originelle
fournie par le scanner


Image traitée en
niveau de gris, à partir
de laquelle sont
calculés les paramètres

• Paramètres de niveau de gris
*Mean, Mode, StdDev, etc.*

• Paramètres de taille
*Area, Perim, etc.*

• Paramètres de forme
*Circularity, Major, Minor, etc.*

• *Paramètres de position*
*X,Y, XM, YM, etc.*

**Références**
• Site WEB (Logiciel IMAGEJ)
http://rsb.info.nih.gov/ij/docs/menus/analyze.html

• Voir aussi le fichier IMAGEJ_Parameters.pdf

1. Classer le plus efficacement possible (avec ce qui est dispo)

2. Regrouper les classes de plancton

3. Produire de nouveaux descripteurs

# 5. Et les autres outils libres ?

# Knime

## Estampillé « Intelligent Data Analysis »


Université de Konstanz - Allemagne

Culture I.D.A
Code source libre C++

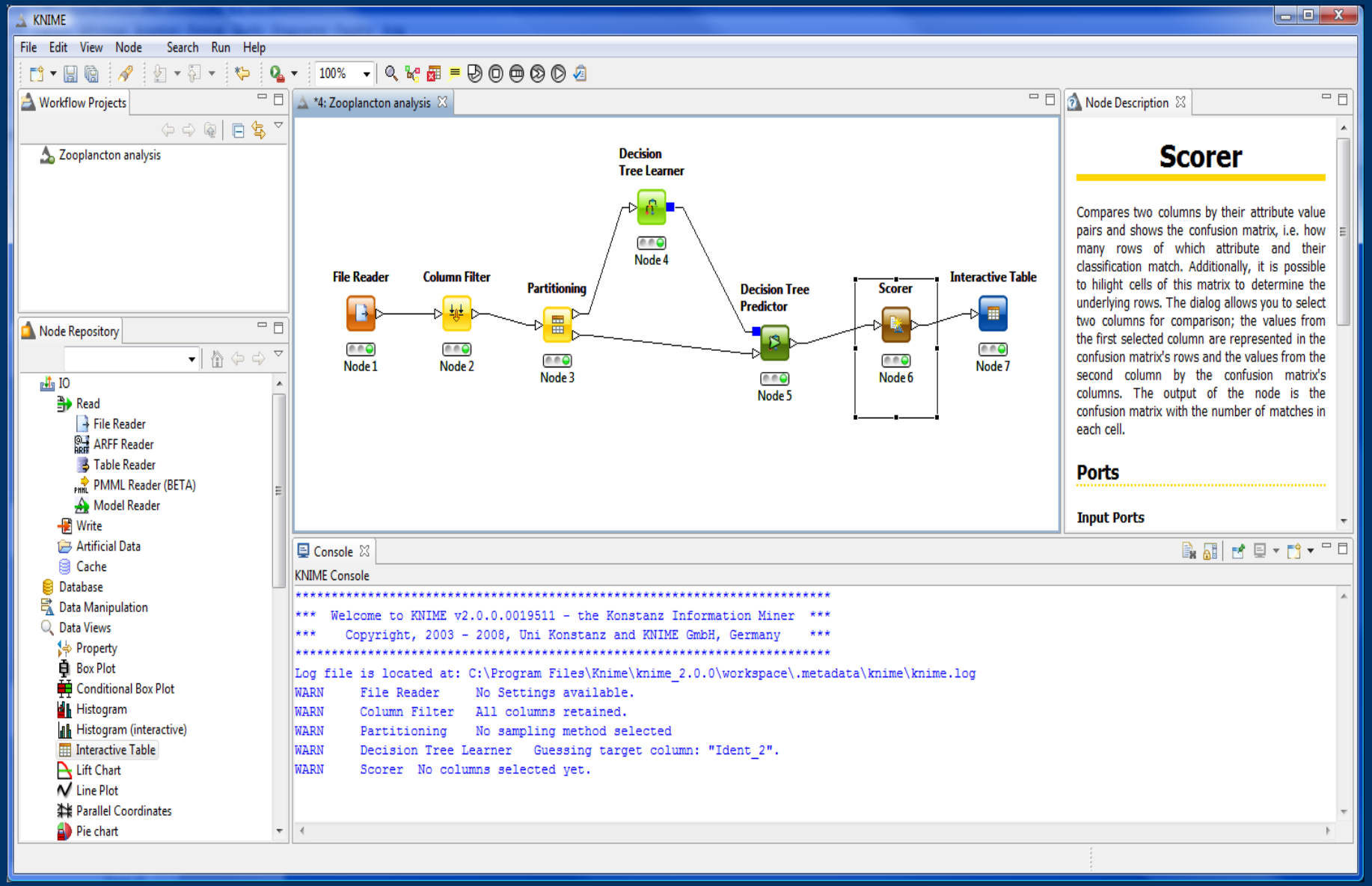
Doc sous forme de fichier d'aide intégré

Mode diagramme
Avec des fonctionnalités avancées (boucles,...)

Les méthodes sont des plugins
Possibilité d'importer des classes Weka
Possibilité d'intégrer des packages R

Multi-thread et possibilité de swap pour certaines méthodes, le mieux armé pour les gros volumes

# Knime
# Interface



# Orange
## Estampillé « Machine Learning »


A.I. Lab – Université de Lubiana – Slovénie

Culture I.A. - Machine Learning (ICML, …)
Code source libre C++

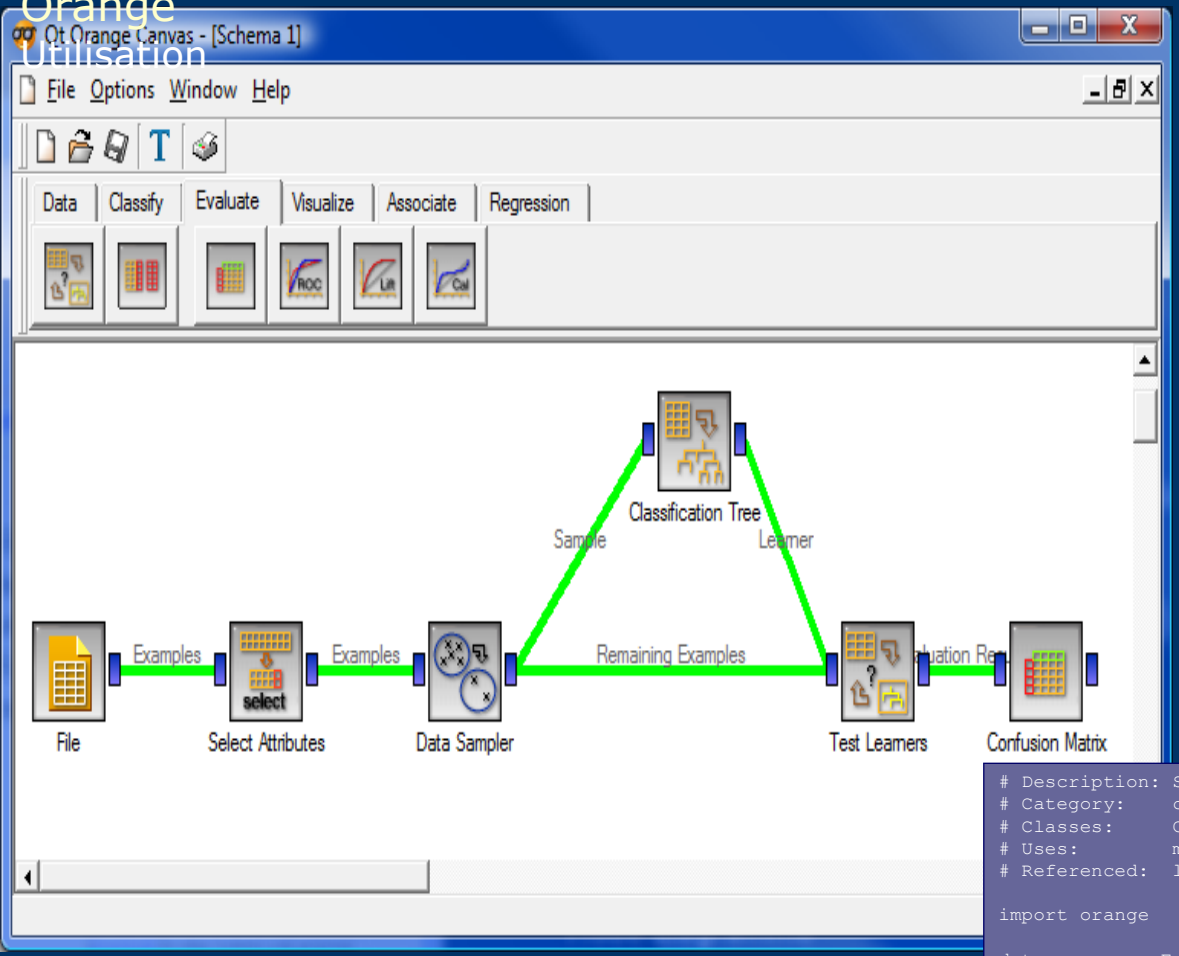
Site Web avec doc en ligne et guide

Mode diagramme
Programmation en Python

Les méthodes sont des plugins (DLL)

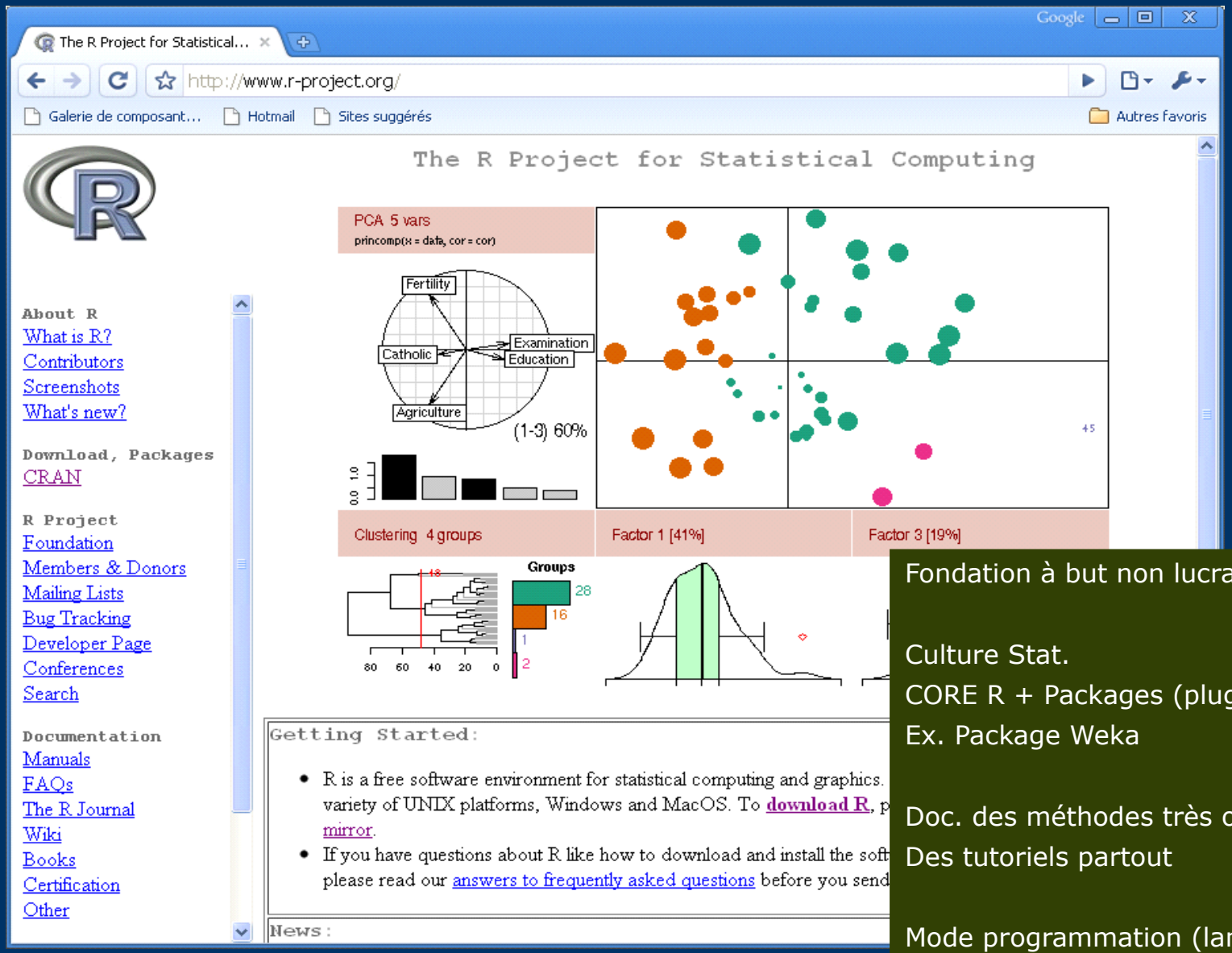
Très user-friendly

# Orange
## Utilisation




```
# Description: Shows how to construct an orange.ClassifierFromExampleTable
# Category:    classification, lookup classifiers, constructive induction, feature construction
# Classes:     ClassifierByExampleTable, LookupLearner
# Uses:        monk1
# Referenced:  lookup.htm

import orange

data = orange.ExampleTable("monk1")
a, b, e = data.domain["a"], data.domain["b"], data.domain["e"]

data_s = data.select([a, b, e, data.domain.classVar])
abe = orange.LookupLearner(data_s)

print len(data_s)
print len(abe.sortedExamples)

for i in abe.sortedExamples[:10]:
    print i
print

for i in abe.sortedExamples[:10]:
    print i, i.getclass().svalue
print

y2 = orange.EnumVariable("y2", values = ["0", "1"])
abe2 = orange.LookupLearner(y2, [a, b, e], data)
for i in abe2.sortedExamples[:10]:
    print i, i.getclass().svalue
print

y2 = orange.EnumVariable("y2", values = ["0", "1"])
abe2 = orange.LookupLearner(y2, [a, b], data)
for i in abe2.sortedExamples:
    print i, i.getclass().svalue
```

# R
## Estampillé « Statistique »




Fondation à but non lucratif

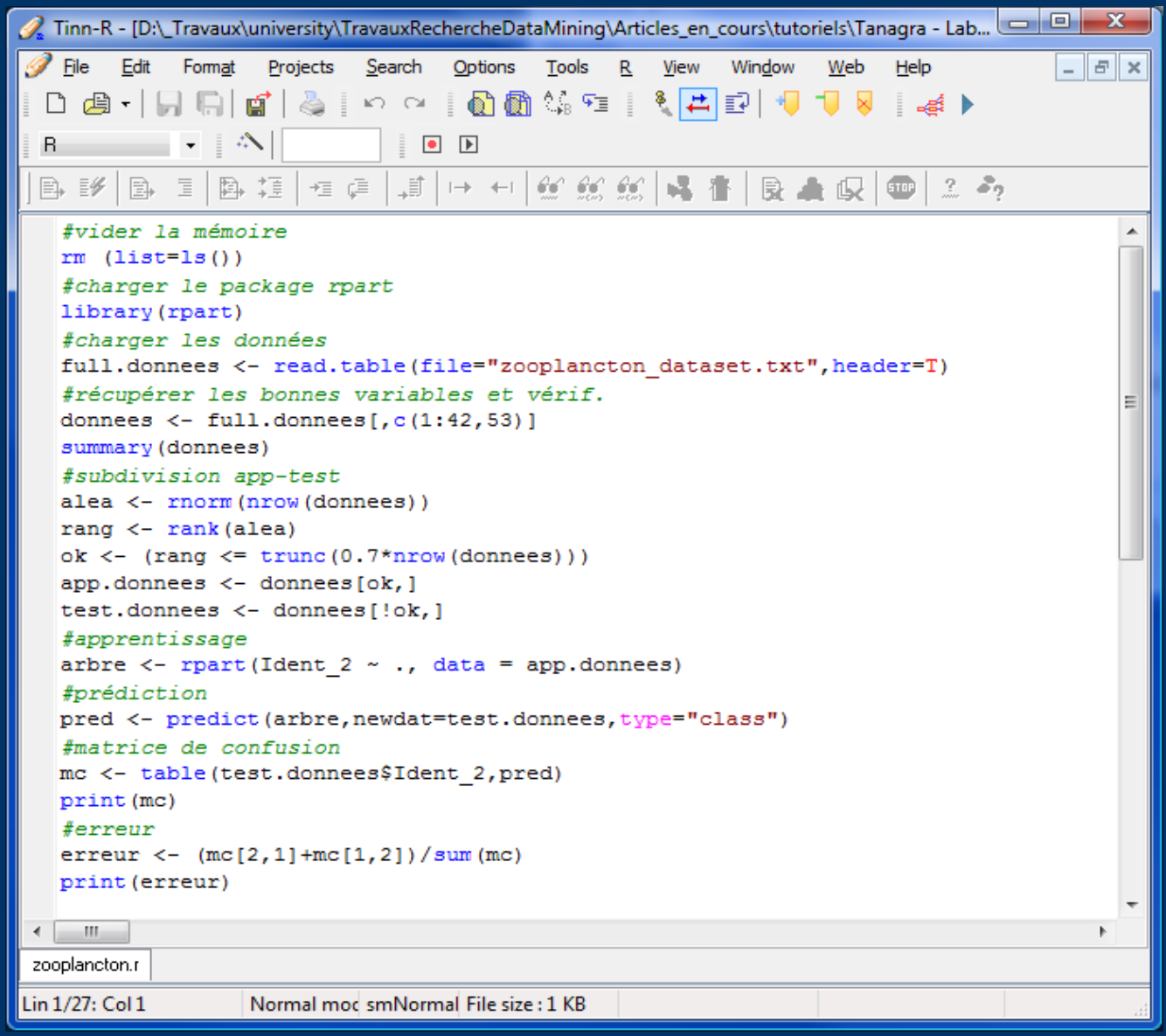
Culture Stat.
CORE R + Packages (plugins)
Ex. Package Weka

Doc. des méthodes très organisée
Des tutoriels partout

Mode programmation (langage S)

Quelques tentatives de création d'interfaces plus conviviales

# R
## Utilisation



# RapidMiner
## « Machine Learning »


Entreprise commerciale
Community Edition – Gratuite

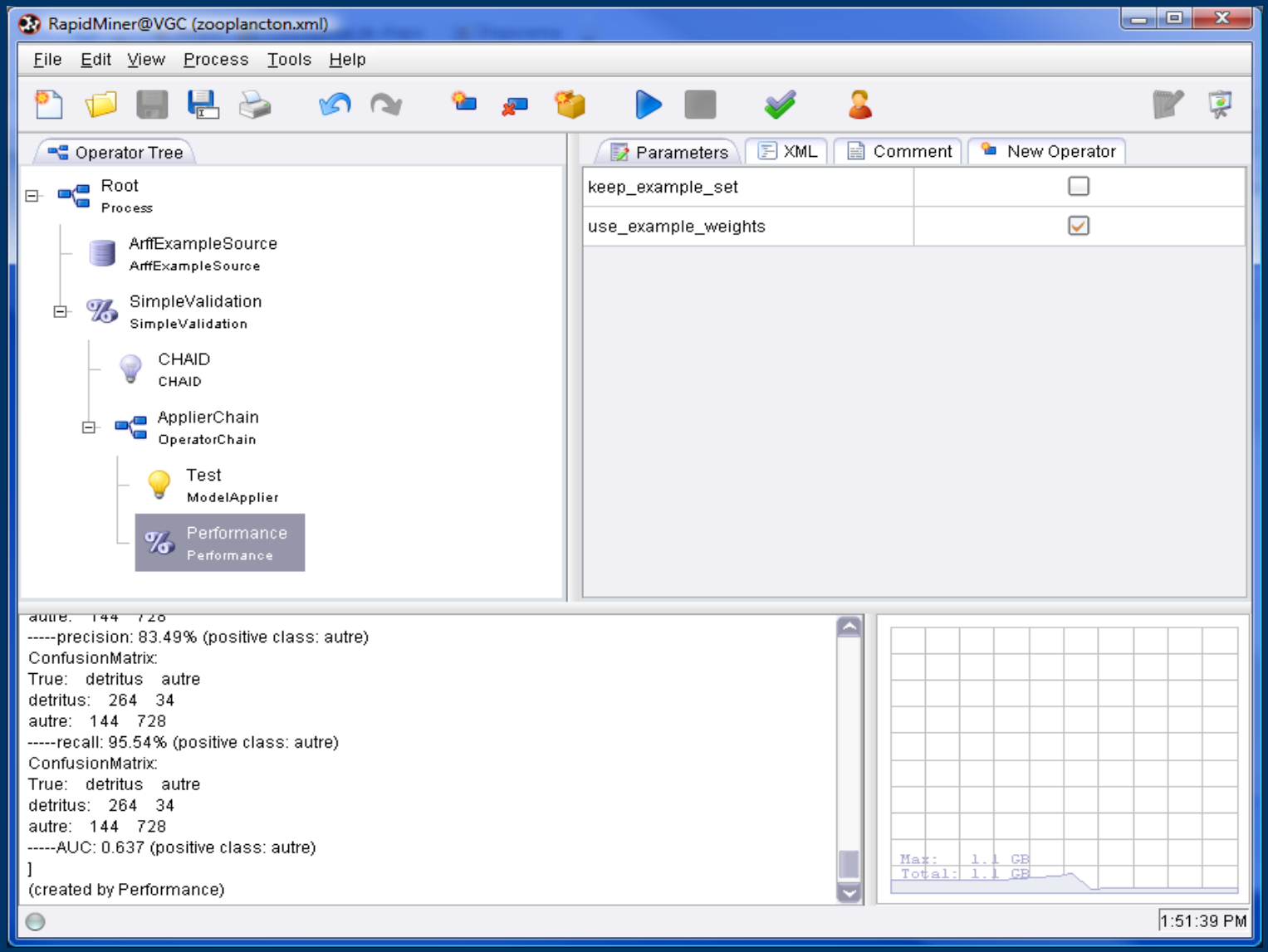
Dérivée de Yale (Licence GNU)
Il existe une version commerciale, sans code source
Code de calcul Weka, mais s'en démarque de plus en plus

Pas de documentation
Mais une multitude d'exemples « pré-programmées »

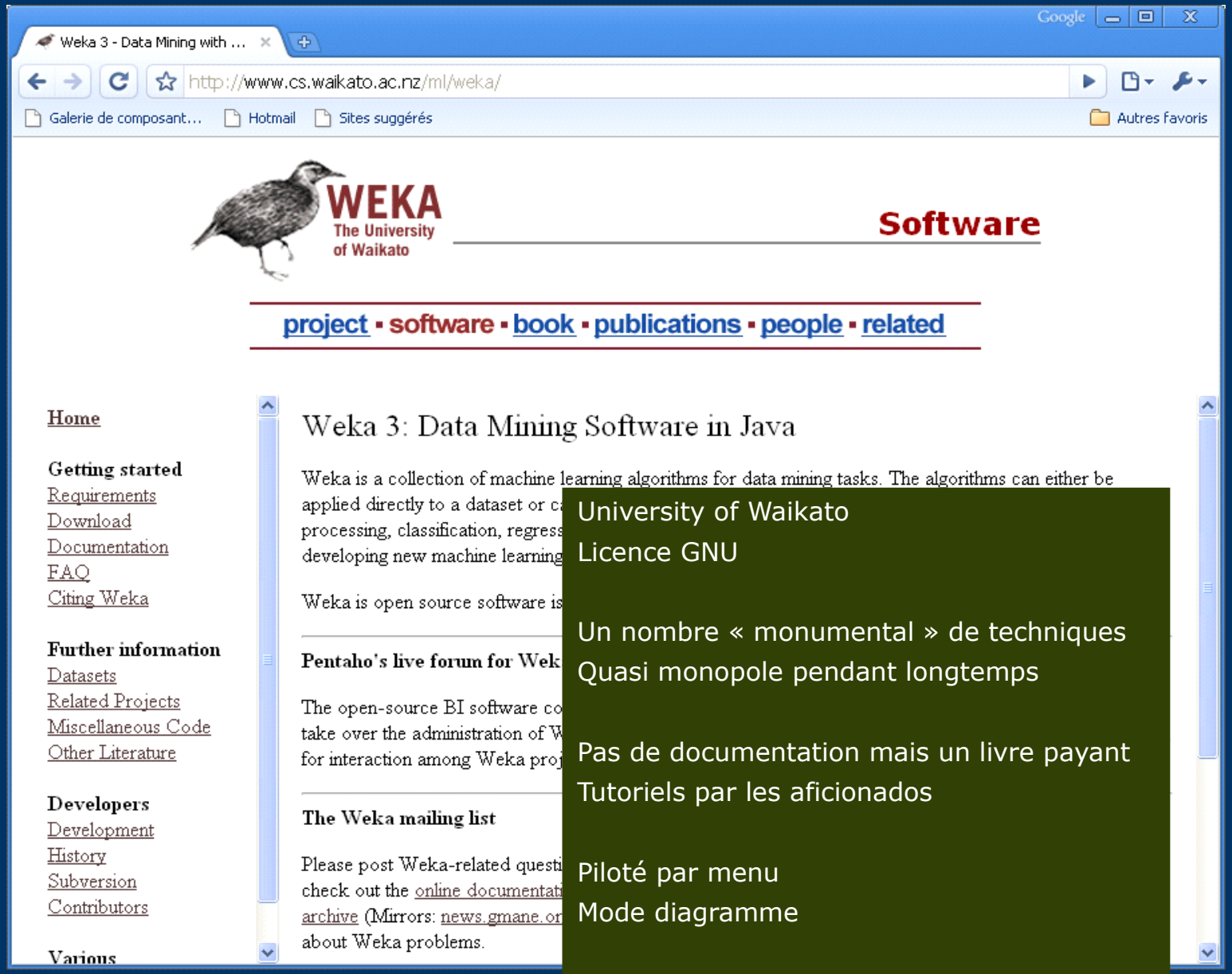
Mode diagramme arborescent

Une « profusion » de techniques de data mining

# RapidMiner
## Utilisation



# Weka
« Machine Learning »



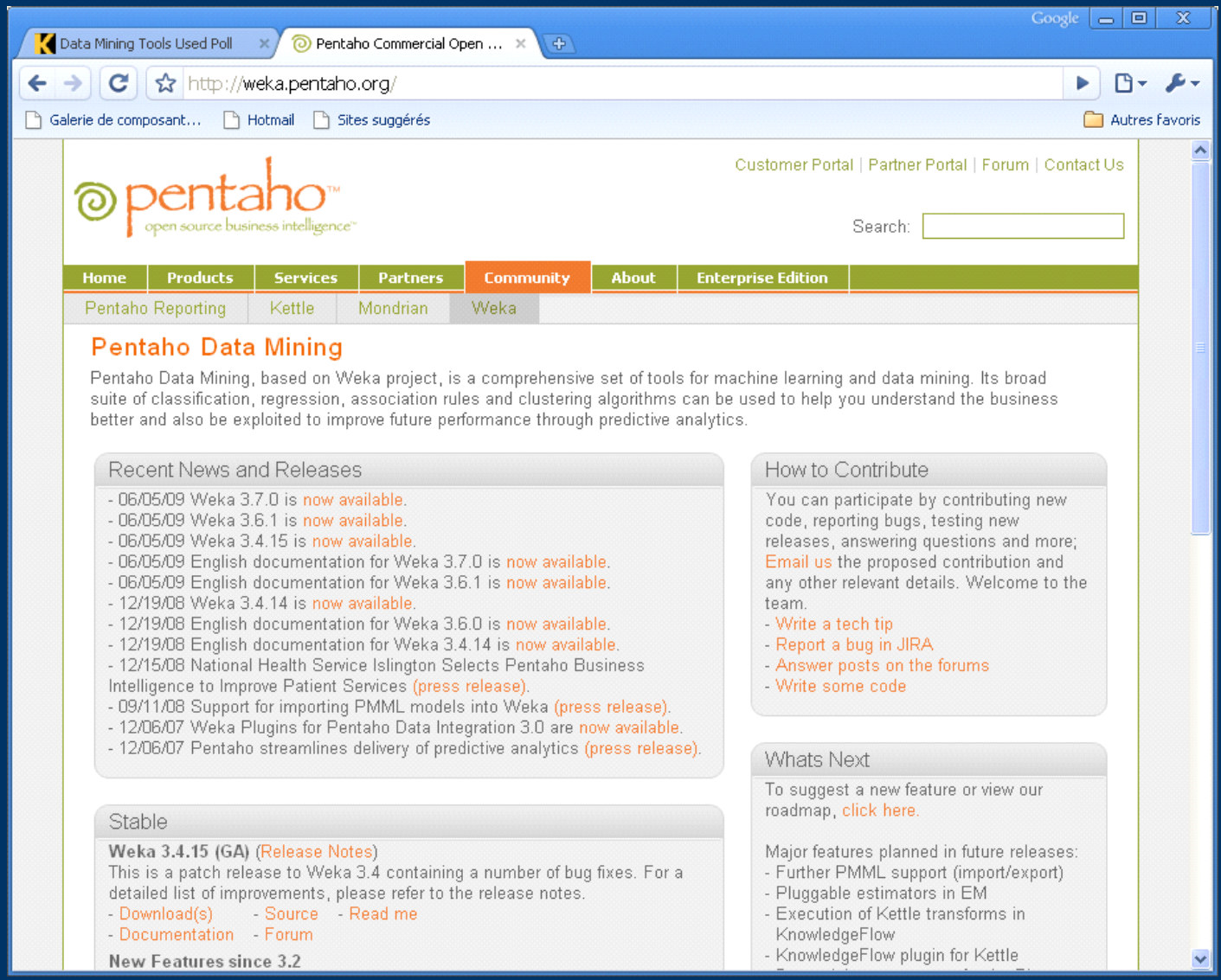

University of Waikato
Licence GNU

Un nombre « monumental » de techniques
Quasi monopole pendant longtemps

Pas de documentation mais un livre payant
Tutoriels par les aficionados

Piloté par menu
Mode diagramme

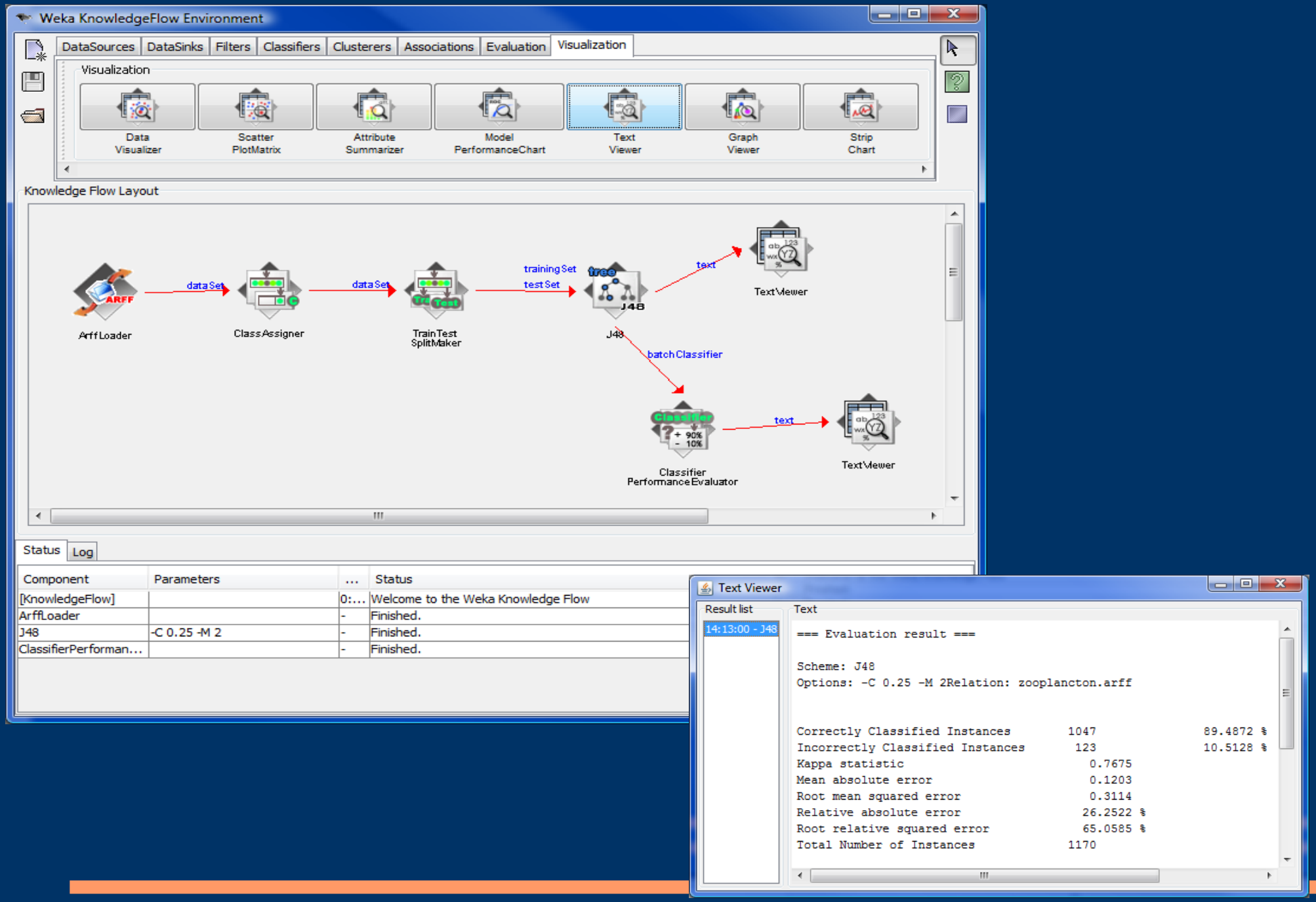
Mais quel avenir ? cf. version Pentaho


Data Mining Tools Used Poll
Pentaho Commercial Open ...
http://weka.pentaho.org/
Galerie de composant...
Hotmail
Sites suggérés
Autres favoris
pentaho
open source business intelligence
Customer Portal | Partner Portal | Forum | Contact Us
Search:
Home
Products
Services
Partners
Community
About
Enterprise Edition
Pentaho Reporting
Kettle
Mondrian
Weka
Pentaho Data Mining
Pentaho Data Mining, based on Weka project, is a comprehensive set of tools for machine learning and data mining. Its broad suite of classification, regression, association rules and clustering algorithms can be used to help you understand the business better and also be exploited to improve future performance through predictive analytics.
Recent News and Releases
- 06/05/09 Weka 3.7.0 is now available.
- 06/05/09 Weka 3.6.1 is now available.
- 06/05/09 Weka 3.4.15 is now available.
- 06/05/09 English documentation for Weka 3.7.0 is now available.
- 06/05/09 English documentation for Weka 3.6.1 is now available.
- 12/19/08 Weka 3.4.14 is now available.
- 12/19/08 English documentation for Weka 3.6.0 is now available.
- 12/19/08 English documentation for Weka 3.4.14 is now available.
- 12/15/08 National Health Service Islington Selects Pentaho Business Intelligence to Improve Patient Services (press release).
- 09/11/08 Support for importing PMML models into Weka (press release).
- 12/06/07 Weka Plugins for Pentaho Data Integration 3.0 are now available.
- 12/06/07 Pentaho streamlines delivery of predictive analytics (press release).
How to Contribute
You can participate by contributing new code, reporting bugs, testing new releases, answering questions and more; Email us the proposed contribution and any other relevant details. Welcome to the team.
- Write a tech tip
- Report a bug in JIRA
- Answer posts on the forums
- Write some code
Whats Next
To suggest a new feature or view our roadmap, click here.
Major features planned in future releases:
- Further PMML support (import/export)
- Pluggable estimators in EM
- Execution of Kettle transforms in KnowledgeFlow
- KnowledgeFlow plugin for Kettle
Stable
Weka 3.4.15 (GA) (Release Notes)
This is a patch release to Weka 3.4 containing a number of bug fixes. For a detailed list of improvements, please refer to the release notes.
- Download(s) - Source - Read me
- Documentation - Forum
New Features since 3.2


Data Mining Tools Used Poll
Discover why Pentaho Dat...
http://www.pentaho.com/products/data_mining/discover_data_mining.php
Galerie de composant...
Hotmail
Sites suggérés
Autres favoris
Products | Support & Services | Partners | Community
pentaho
open source business intelligence
Pentaho BI Suite
Enterprise Edition
Download
Pentaho Data Mining Enterprise Edition
Gain insight into hidden patterns and relationships in your data to discover indicators of future performance.
Discover
Try
Buy
Pentaho Data Mining
Once you've got analysis, reporting, and dashboards deployed, it's time to take your business intelligence (BI) to the next level by adding data mining and advanced analytics. This is a level of BI excellence that many organizations never manage to evolve to, however the importance of pushing ahead with advanced capabilities cannot be underestimated - they can provide a truly sustainable competitive advantage and enable your organization to maximize both its efficiency and effectiveness.
Data Mining is the process of running data through sophisticated algorithms to uncover meaningful patterns and correlations that may otherwise be hidden. These can be used to help you understand the business better and also exploited to improve future performance through predictive analytics. For example, data mining can warn you there's a high probability a specific customer won't pay on time based on an analysis of customers with similar characteristics.
Explore and Learn
Resources
Data Sheet
User Forum
Download
Deploying Data Mining Models
PMML Support
Request a Quote
Popular Links
BI Economics White Paper
More Links
Version 3
User Friendly
Cloud Ready
Community Powere
Pentaho BI Suite 3.5
Design. Deploy. Escape.

# Weka

## Utilisation en mode « Knowledge flow »



# Performances comparées
## Gros volumes (1/2)

| Logiciel | Temps de traitement (secondes) | | Occupation mémoire (Mo) | | | |
|---|---|---|---|---|---|---|
| | Importation | Induction arbre | Avant lancement | Après importation | Pic traitement | Après induction |
| KNIME | 47 | 270 | 92.6 | 160.4 | 245.8 | 245.8 |
| ORANGE | 90 | 130 | 24.9 | 259.5 | 795.7 | 795.7 |
| R (package rpart) | 24 | 157 | 18.8 | 184.1 | 718.9 | 718.9 |
| RAPIDMINER | 7 | 298 | 136.3 | 228.1 | 1274.4 | 1274.4 |
| SIPINA | 25 | 122 | 7.8 | 67.1 | 539.9 | 539.9 |
| TANAGRA | 11 | 33 | 7.0 | 53.1 | 121.6 | 73.5 |
| WEKA | 10 | 338 | 52.3 | 253.2 | 699.6 | 699.6 |

Arbres de décision, 500.000 obs., 21 descripteurs

# Performances comparées
## Gros volumes (2/2)

| Logiciel | Temps de traitement (sec.) | | Taux d'erreur en validation croisée (%) | Occupation mémoire (Mo) | | | |
|---|---|---|---|---|---|---|---|
| | Importation | Calcul | | Au lancement | Avec les données | Durant le traitement | Durant la validation croisée |
| ORANGE | 95 | 690 | 4% (6/135) | 25 | 118 | 317 | 406 |
| RAPIDMINER JMySVMLearner | 5 | 29 | 11% (15/135) | 124 | 210 | 338 | 608 |
| RAPIDMINER C-SVC (LIBSVM) | 5 | 9 | 2% (3/135) | 124 | 210 | 442 | 870 |
| TANAGRA - SVM | 12 | 130 | 4% (6/135) | 7 | 337 | 393 | 393 |
| TANAGRA C-SVC (LIBSVM) | 12 | 11 | 4% (6/135) | 7 | 337 | 406 | 406 |
| WEKA - SMO | 11 | 12 | 3% (4/135) | 54 | 243 | 489 | 595 |

SVM, 135 obs., 31809 descripteurs

Les outils se tiennent, tout dépend des méthodes et des caractéristiques des données !!!

# Comment choisir

## Quel logiciel pour quel contexte ?

Recherche (Data Mining)

Développer de nouvelles techniques

Les intégrer dans un environnement opérationnel

Pour des comparaisons à grande échelle

Les diffuser simplement et largement

→ Logiciel R

Avec les Packages

Utilisateur (Ou Recherche autre que Data Mining)

Contexte d'exploration des données

i.e. appliquer les techniques à des données,

Les faire coopérer (ces techniques)

Interpréter et publier les résultats

Enseignement

→ Les outils se valent

Critères de différenciation

Manipulation des données - texte/tableur/sgbd

Pouvoir les enchaîner (tous)

Traitement des très gros volumes (Knime ?)

Profusion des techniques (oui et non)

Outils graphiques (Knime, Orange)

Notoriété (Weka)

Et TANAGRA ?

Culture francophone du traitement des données

Machine Learning + Analyse de données et statistique

Un effort constant sur la documentation