Voici une revue rapide des progiciels gratuits accessibles [FREE AND SHAREWARE] dans la section SUITES du site KDNUGGETS (http://www.kdnuggets.com/software/suites.html).

L'étude sera approfondie pour les logiciels répondant aux spécifications suivantes : (1) il est possible d'enchaîner les traitements sans avoir à programmer ; (2) l'outil est capable de mettre en œuvre et de faire coopérer différentes méthodes ;(3) il est possible de mettre en place des expérimentations pour évaluer et comparer les performances de différentes approches sur un problème donné ; (4) le logiciel doit être assez « léger », ne nécessitant pas l'installation de bibliothèques complexes pour fonctionner (l'installation d'un SGBD spécifique par exemple).

1. Vue rapide des logiciels disponibles

1.1. ADAM

(http://datamining.itsc.uah.edu/adam/)

C'est une collection d'exécutables qui implémentent des méthodes de traitement de données. Le format de fichier de données standard est le format WEKA (ARFF). Si on veut enchaîner des traitements, il faudrait donc mettre à la suite la succession des traitements dans un fichier script BAT.

Particularité intéressante, certains modules peuvent traiter directement des images.

1.2. ALPHAMINER*

(http://www.eti.hku.hk/alphaminer/)

C'est un logiciel stand-alone qui implémente la chaîne de traitements sous forme de graphes. Il peut prendre en entrée des fichiers textes ou EXCEL.

Sa principale particularité est qu'il est capable de s'interfacer avec des plug-ins écrits dans d'autres contextes, il peut notamment récupérer des classes compilées dans WEKA par exemple.

1.3. DATABIONIC ESOM TOOLS

(http://databionic-esom.sourceforge.net/)

L'idée est de mettre les SOM à toutes les sauces : visualisation, classification et classement. Le logiciel est totalement dédié à ce paradigme.

1.4. GNOME DATA MINING TOOLS

(http://www.togaware.com/datamining/gdatamine/)

Un logiciel fonctionnant sous LINUX et implémentant quelques méthodes de traitement de données (arbres de décision, règles d'association, graphiques…). Les modules sont indépendants, ils possèdent leur propre interface, il ne semble pas possible de les enchaîner sans passer par la programmation.

1.5. IBM INTELLIGENT MINER

(http://www.developer.ibm.com/university/scholars/)

Apparemment, ce logiciel ne peut pas fonctionner dans le serveur de base de données DB2. Il s'appuie dessus pour la gestion des données.

La panoplie des méthodes usuelles (classement avec les arbres, classification, etc.) semble présente. En revanche, les possibilités d'enchaînement ne sont pas très claires. Le logiciel est gratuit pour l'enseignement.

1.6. MINING MART

(http://mmart.cs.uni-dortmund.de/)

Ce logiciel met l'accent sur la préparation des données, préalable primordial avant la mise en œuvre des techniques exploratoires. Sa particularité est la nécessité de s'interfacer avec un SGBD relationnel. Il peut traiter une série de tables reliées entre elles ; il ne peut pas en revanche appréhender un fichier texte externe.

Dans la version 0.22, il faut apparemment une version d'ORACLE pour le package fonctionne ?

L'interface respecte le standard des chaînes de traitements, plusieurs composants sont dédiés au pré-traitement des données : jointures de tables, requêtes, etc. Le rapprochement avec les outils d'ETL (Extraction Transfer Loading) est tout à fait approprié.

1.7. MLC++

(http://www.sgi.com/tech/mlc/)

C'est une librairie de classes C++ destiné à l'apprentissage supervisé. Le code source est disponible, il est impossible de le mettre en œuvre sans le compiler soi-même, de même si on veut enchaîner une série de manipulations, il faut passer par la programmation.

Le principal intérêt de cette bibliothèque est qu'elle a été écrite par KOHAVI, un acteur important de la communauté MACHINE LEARNING, et qu'elle constitue le cœur du logiciel MINESET de SGI.

1.8. MLJ (Machine Learning in JAVA)

(http://www.kddresearch.org/Groups/Machine-Learning/MLJ/)

C'est un portage de MLC++. Il ne semble pas présenter d'intérêt supplémentaire, mis à part que l'on dispose également du code source.

1.9. STAR PROBE

(http://www.roselladb.com/starprobe.htm)

C'est une application implémentant plusieurs techniques (arbres de décision, visualisation, etc.) qui peut fonctionner soit en stand-alone, soit comme une application client-serveur. Une version limitée est accessible directement sur le web, il s'agit d'une APPLET JAVA. Il existe

une interface graphique, en revanche, il ne semble pas possible de définir une chaîne de traitements.

1.10. ORANGE*

(http://magix.fri.uni-lj.si/orange/)

Développé par l'Université de Slovénie, ce logiciel correspond exactement à notre cahier de charges. Il est possible de programmer les traitements à l'aide de scripts en PYTHON, il est également possible de définir les traitements à l'aide d'un graphe représentant les « filières ». Le cœur des algorithmes de calcul sont compilés dans des DLL écrits en C++.

1.11. YALE*

(http://www-ai.cs.uni-dortmund.de/SOFTWARE/YALE/index.html)

Ce logiciel s'appuie sur le cœur de WEKA et propose une interface graphique permettant de représenter l'enchaînement des traitements sous forme d'arbre : arborescence avec, à la racine, la source de données. Ce logiciel intègre une particularité intéressante : des traitements types ont été définis et un wizard permet de définir assez aisément la suite de traitements que l'on veut mettre en place. Il n'est donc pas nécessaire (pas possible d'ailleurs) de composer soi même les icônes représentant les traitements.

Point très important, tant que WEKA n'intégrait pas d'interface graphique pour définir les chaînes de traitements, YALE pouvait représenter une alternative intéressante pour les allergiques de la programmation. Ca n'est plus d'actualité maintenant ? Il semble quand même que YALE propose une alternative intéressante, bien que compliquée, pour définir des opérations complexes (validation croisée sur des méthodes utilisant le boosting par exemple).

Des efforts en matière de représentations graphiques ont été réalisés récemment.

1.12. WEKA*

(http://www.cs.waikato.ac.nz/ml/weka/)

Faut-il vraiment en parler ? Une bibliothèque de méthodes faramineuse, un code assez bien écrit la plupart du temps, avec des architectures très accessibles : la reprise du code ne pose aucun problème. Un bond en avant a été réalisé avec l'introduction d'une interface graphique permettant de définir en enchaînement de traitements sans avoir recours à la programmation.

2. Comparaison des logiciels

2.1. Philosophie de la comparaison

Il y a plusieurs manières de comparer les logiciels, dans notre cas nous cherchons des logiciels qui permettent de mener des travaux dirigés dans le cadre de l'enseignement du data mining :

a. Nous nous attachons au fond et non pas à la forme, dans ce cas qu'importe l'interface du logiciel et son mode opératoire, l'essentiel est de pouvoir mettre en œuvre les méthodes de fouille de données.

b. Il faut néanmoins que l'interface utilisateur soit proche des logiciels commerciaux usuels pour que les étudiants puissent quand même se familiariser avec les standards, s'ils existent, reconnus dans l'industrie.

c. Il est nécessaire également que l'utilisation du logiciel ne requière pas des compétences particulières lors de sa mise en œuvre. Cela concerne surtout la préparation des données (possibilité d'importer simplement des fichiers en provenance de tableurs par exemple) et la définition des traitements (il ne doit pas être nécessaire d'apprendre un langage de script particulier pour définir des séries de traitements).

A partir de ces éléments, nous avons défini un cahier des charges des logiciels que nous voulons évaluer et comparer.

a. **Gratuité** : le logiciel et tous les modules associés doivent être accessibles gratuitement.

b. **Installation simplifiée du logiciel** : l'installation du logiciel sur la machine doit pouvoir se faire simplement, il ne doit surtout pas nécessiter l'installation de serveurs lourds. Le logiciel fonctionne en stand-alone.

c. **Gestion simplifiée des données** : qu'importe si les logiciels adoptent leur propre format, l'essentiel est que l'on puisse importer aisément des données au format texte issu de tableurs.

d. **Définition des opérations par diagramme de traitements** : c'est le standard actuel, que l'on parle de « filière » ou de « chaîne de traitements », l'idée est la même, définir l'enchaînement des opérations sous forme d'un graphe sans avoir à écrire une seule ligne de code.

e. **Evaluation des méthodes d'apprentissage :** concernant surtout l'apprentissage supervisé, le logiciel doit proposer les outils standards d'évaluation des performances (estimation de l'erreur en resubstitution, en test ou par ré-échantillonnage) ; la présence éventuelle des outils de scoring est également appréciée.

f. **Affichage des résultats, aide à l'interprétation et exportation des résultats** : les résultats doivent être lisibles simplement avec une mise en forme correcte. Il doit être possible de les exporter et de les reprendre dans les outils d'édition usuels (traitement de texte, diaporama, etc.)

2.2. Critères d'évaluation

Finalement, nous comparons les logiciels TANAGRA, WEKA, ORANGE (à voir pour ALPHA MINER et YALE – tout présenter serait irréaliste et amènerait plus de confusion qu'autre chose). On procèdera plus à une comparaison qualitative qu'à un classement ou à une notation des logiciels. Nous listons ci-dessous les critères à utiliser pour comparer les logiciels : ces critères découlent du cahier des charges défini ci-dessus, nous avons également repris certains des critères qui ont été mis en avant dans les études sur les logiciels publiées ces dernières années.

- Origine et promoteurs. Architecture du logiciel.
- Interface globale de définition des traitements (arbre ou graphe)
- Accès aux données (type de données traitées, types de variables gérées) et de manière plus générale la gestion des données : dans ce cas précis, tous les logiciels

évalués adoptent la même approche, lecture en une seule passe des données en mémoire, le fichier est alors libéré.

- Définition des traitements, leur enchaînement.
- Mode d'affichage et récupération des résultats
- La bibliothèque des traitements (description statistique, représentation et exploration graphique, sélection des individus et échantillonnage, sélection de variables, construction de variables synthétiques, classification, classement, régression, association, validation et évaluation des performances, ciblage)
- Evaluer les performances des méthodes et comparaisons
- Monter des expérimentations sur des jeux de données.
- Performances sur les gros ensembles de données (lignes et/ou colonnes).
- Exploration interactive graphique des données.

D'autres critères, importants dans le monde industriel, ont été sciemment mis de côté :

- Accès et interfaçage avec les SGBD.
- Déploiement des modèles construits (mise en production).
- Reporting dynamique (avec OLE par exemple, un graphique inséré dans un document word serait automatiquement mis à jour lorsque les traitements sont ré-exécutés).