FOUILLE DE DONNEES

Anne LAURENT

laurent@lirmm.fr

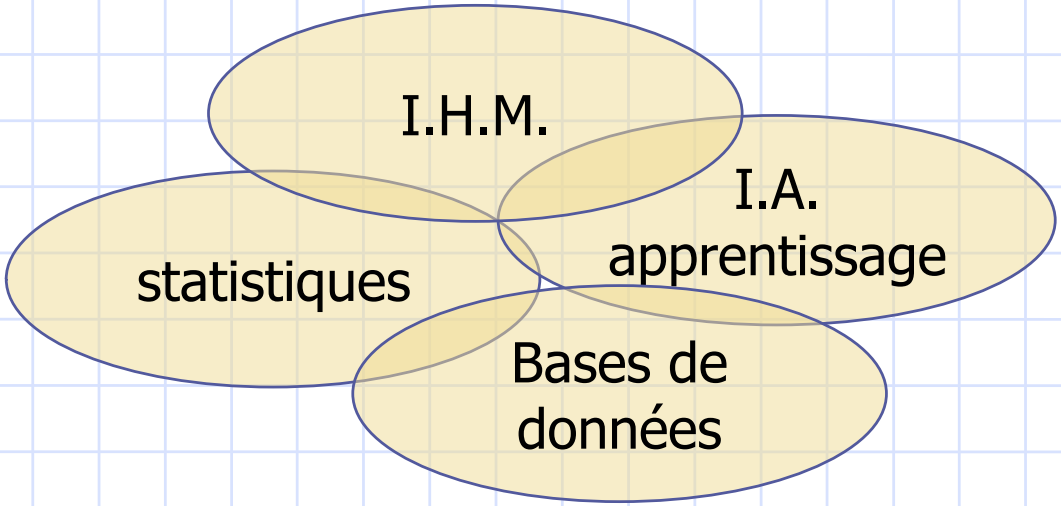
ECD

Pourquoi la fouille de données ?

- Données disponibles
- Limites de l'approche humaine
- Nombreux besoins :
  - Industriels,
  - Médicaux,
  - Marketing,
  - …

Qu'est-ce que la fouille de données ?

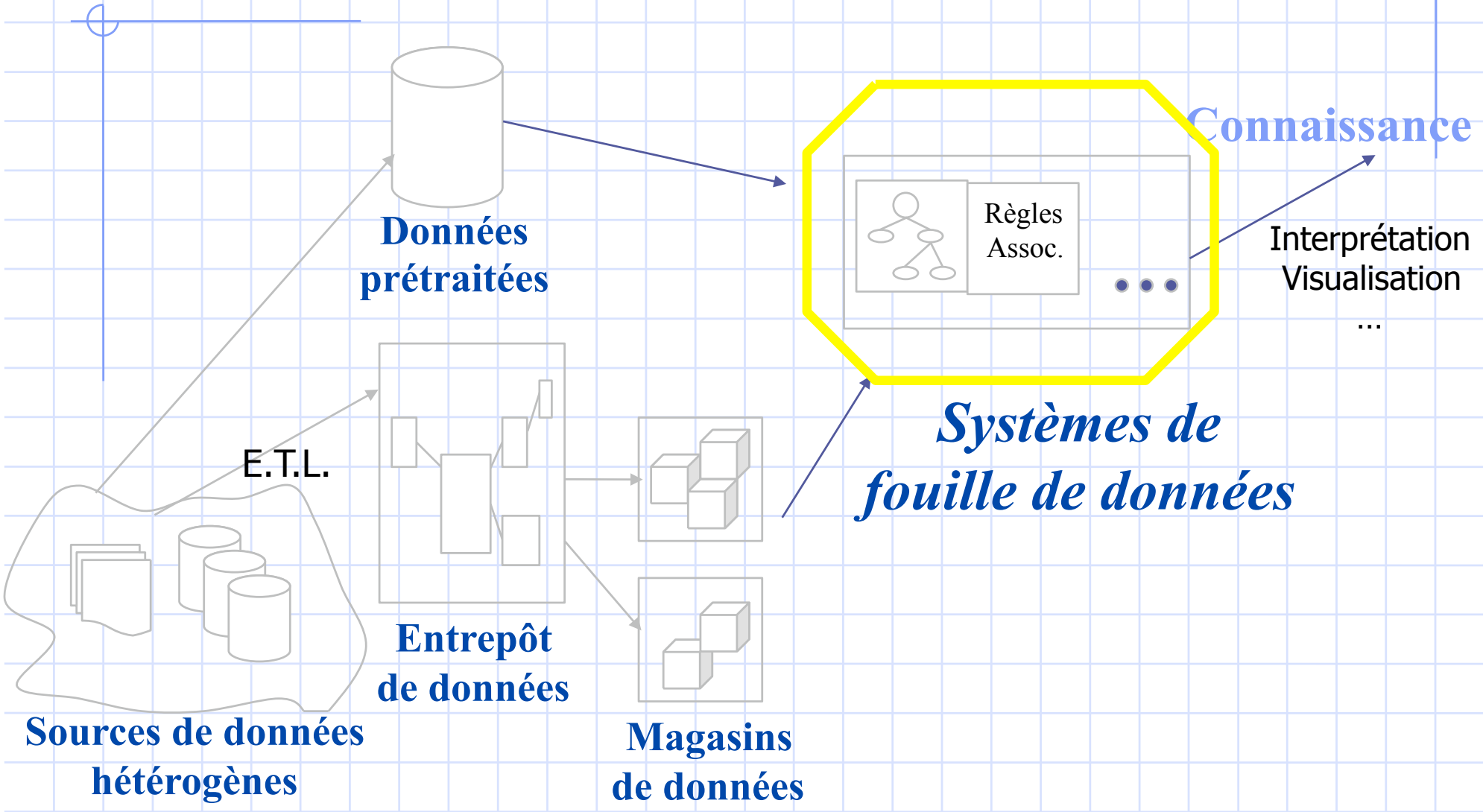
- Fayyad (1996) Knowledge Discovery in Databases : *"the non-trivial process of identifying valid, potentially useful and ultimately understandable patterns in data"*
- KDD vs. Data Mining



Buts : exemples d'application

- diagnostic médical
- profils de clients, mailing, accord de prêts bancaires, ...
- reconnaissance de caractères manuscrits
- finance, prévision d'évolutions de marchés
- Customer Relationship Management (**CRM**) : trouver, gagner et garder ses clients !
  - churning,
  - détection de fraudes,
  - détection de mauvais payeurs, …

Des données aux connaissances ...



Tâches de fouille de données

- Classification
- Estimation
- Prédiction
- Règles d'association
- Segmentation

Algorithmes supervisés et non supervisés

- Apprentissage supervisé :
  - On dispose d'un fichier décrivant des données alliant une description et une classe
  - On cherche une fonction de classification permettant d'induire la classe en fonction d'une description
- Apprentissage non supervisé :
  - On dispose d'un fichier de description des données sans classes connues a priori
  - On cherche à diviser ces données en catégories

Algorithmes prédictifs et descriptifs

Veut-on

- Trouver une fonction permettant de prédire la classe d'une données jamais vue

Ou

- Trouver des descriptions résumées et pertinentes expliquant les données
- La limite entre les 2 est floue ! (méthodes descriptives pour la prédiction)

Problématiques associées

- données pas forcément très *propres* :
  - données bruitées
  - données manquantes
  - Données aberrantes
  - doublons
- données numériques, symboliques

- pré-traitements
- post-traitements

Algorithmes de ce cours

- Classification supervisée :
  - Méthode de Bayes naïf
  - k plus proches voisins
  - Arbres de décision
  - Réseaux de neurones
- Classification non supervisée : k-means
- Règles d'association et motifs séquentiels
- Évaluation des méthodes

Méthode de Bayes naïf

- Comment classer un nouvel exemple en fonction d'un ensemble d'exemples pour lesquels on connaît la classe ?
- Soit un exemple d = (d1, ..., dn) et c classes k = 1, ..., c

$$\text{Classe}(d) = \underset{k}{\operatorname{argmax}} \prod_{i} \hat{P}(di|k) \cdot \hat{P}(k)$$

proportion d'exemples di parmi ceux de la classe k

proportion d'exemples de la classe k

Exemple : kite-surf ?

	TEMPS	HUMIDITE	VENT	KITE SURF
Ex1	Soleil	Haute	Oui	Oui
Ex2	Soleil	Basse	Non	Non
Ex3	nuageux	Basse	Oui	Oui
Ex4	pluvieux	Haute	Oui	Non
Ex5	pluvieux	Basse	Oui	Non
Ex6	Soleil	Basse	Oui	Oui
Ex7	pluvieux	Basse	Non	Non
	Soleil	***haute***	***Non***	***?***

- Va-t-on jouer s'il y a du soleil, beaucoup d'humidité et pas de vent ?

k plus proches voisins

- Raisonnement à partir de cas
- Utilisation des cas similaires pour prendre une décision
- Pas d'étape d'apprentissage (avantages et inconvénients)

Algorithme

- Décider du nombre de voisins à utiliser k (souvent k = nbre d'attributs + 1)
- Pour un enregistrement sur lequel il faut décider :
  - trouver les k plus proches voisins
  - combiner les classes des k plus proches voisins en une classe c

Choix de la distance

- Rappel : distance d $\Leftrightarrow \begin{cases} d(A,A) = 0 \\ d(A,B) = d(B,A) \\ d(A,C) < d(A,B) + d(B,C) \end{cases}$

- Distance sur chacun des attributs

$d(x,y) = |x-y|$ $\quad$ $d(x,y) = |x-y| / distance\_max$

- puis combinaison. distance euclidienne :

$$d(x,y) = \sqrt{[d_1(x_1,y_1)^2 + \ldots + d_n(x_n,y_n)^2]}$$

Choix de la classe

- on dispose des k classes des k plus proches voisins
- choix de la classe du nouvel exemple :
  - classe majoritaire
  - classe pondérée
- Le résultat change en fonction de tous ces choix (distance, combinaison, calcul de la classe)

Exemple : va-t-on jouer au tennis avec cette méthode ?

- on choisit k = 4
- distance euclidienne

  d(A,A)=0

  d(A,B)=1
- calcul des voisins
- combinaison des classes des voisins

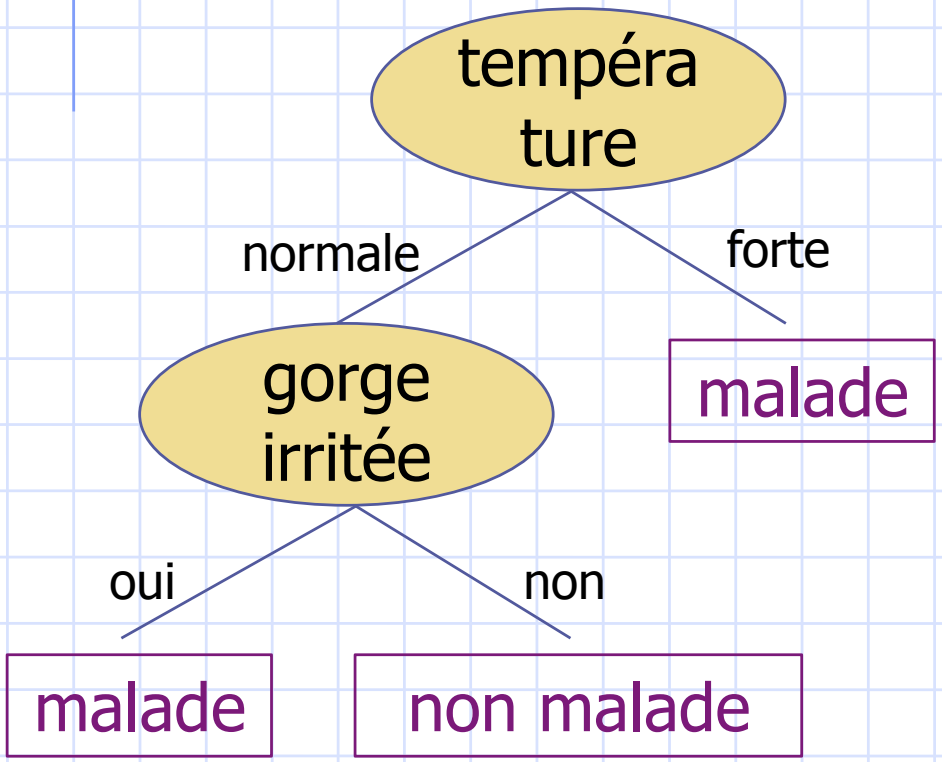
Exemple : kite-surf ?

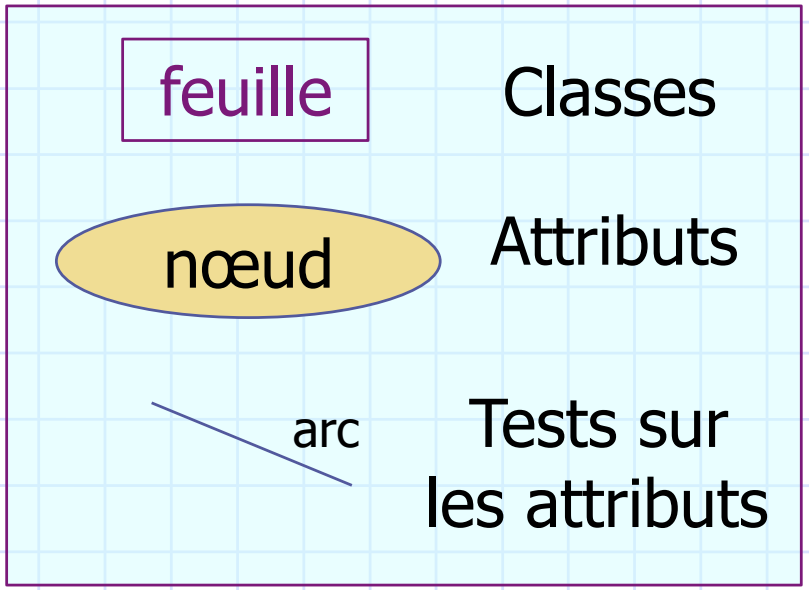
	TEMPS	HUMIDITE	VENT	KITE SURF
Ex1	Soleil	Haute	Oui	Oui
Ex2	Soleil	Basse	Non	Non
Ex3	nuageux	Basse	Oui	Oui
Ex4	pluvieux	Haute	Oui	Non
Ex5	pluvieux	Basse	Oui	Non
Ex6	Soleil	Basse	Oui	Oui
Ex7	pluvieux	Basse	Non	Non
	Soleil	***haute***	***Non***	***?***

- Va-t-on jouer s'il y a du soleil, beaucoup d'humidité et pas de vent ?

Arbres de décision

- Représentation graphique d'une procédure de décision
- Représentation compréhensive ⇒ règles





Problématiques associées

- **Choix des attributs tests** (divisions successives de la base d'apprentissage)
- **Critère d'arrêt**
- **But :** construire un arbre le plus petit possible
- **Heuristique**. Algorithme *glouton*.
- Plusieurs algorithmes (ID3, C4.5)

Algorithme de construction

- Nœud Courant ← racine
- **Répéter**
  - Si le nœud courant est terminal
  - Alors l'étiqueter Nœud Courant ← Classe
  - Sinon
    - Sélectionner un attribut test
    - Créer le sous-arbre
  - Passer au nœud suivant non exploré
- **Jusqu'à** obtention d'un arbre

Critère d'arrêt

- Plusieurs tests possibles pour décider si le nœud courant est terminal :
  - il n'y a plus assez d'exemples
  - les exemples ne sont *pas trop mélangés* (une classe se dégage). seuil d'impureté.
- On étiquette avec la classe majoritaire

Sélection de l'attribut test

- Quel est l'attribut dont la connaissance nous aide le plus sur la classe ?
- Plusieurs critères possibles : test de Gini, gain d'information, entropie, ...
- ID3 : entropie de Shannon

Entropie de Shannon

- Entropie de l'attribut A
- A a i valeurs possibles X_1, ..., X_i
- Il y a k classes C_1, ..., C_k

$$H_s(C|A) = - \sum_i P(X_i) \sum_k P(C_k|X_i). \log((P(C_k|X_i)))$$

Exemple : va-t-on jouer au tennis avec cette méthode ?

- Construction de l'arbre
- Racine : choix du 1er attribut test
  - Calcul de H(C|temps)
  - Calcul de H(C|humidité)
  - Calcul de H(C|vent)
- Division de la base d'exemple
- Nœuds terminaux ?

Exemple : kite-surf ?

	TEMPS	HUMIDITE	VENT	KITE SURF
Ex1	Soleil	Haute	Oui	Oui
Ex2	Soleil	Basse	Non	Non
Ex3	nuageux	Basse	Oui	Oui
Ex4	pluvieux	Haute	Oui	Non
Ex5	pluvieux	Basse	Oui	Non
Ex6	Soleil	Basse	Oui	Oui
Ex7	pluvieux	Basse	Non	Non
	Soleil	***haute***	***Non***	***?***

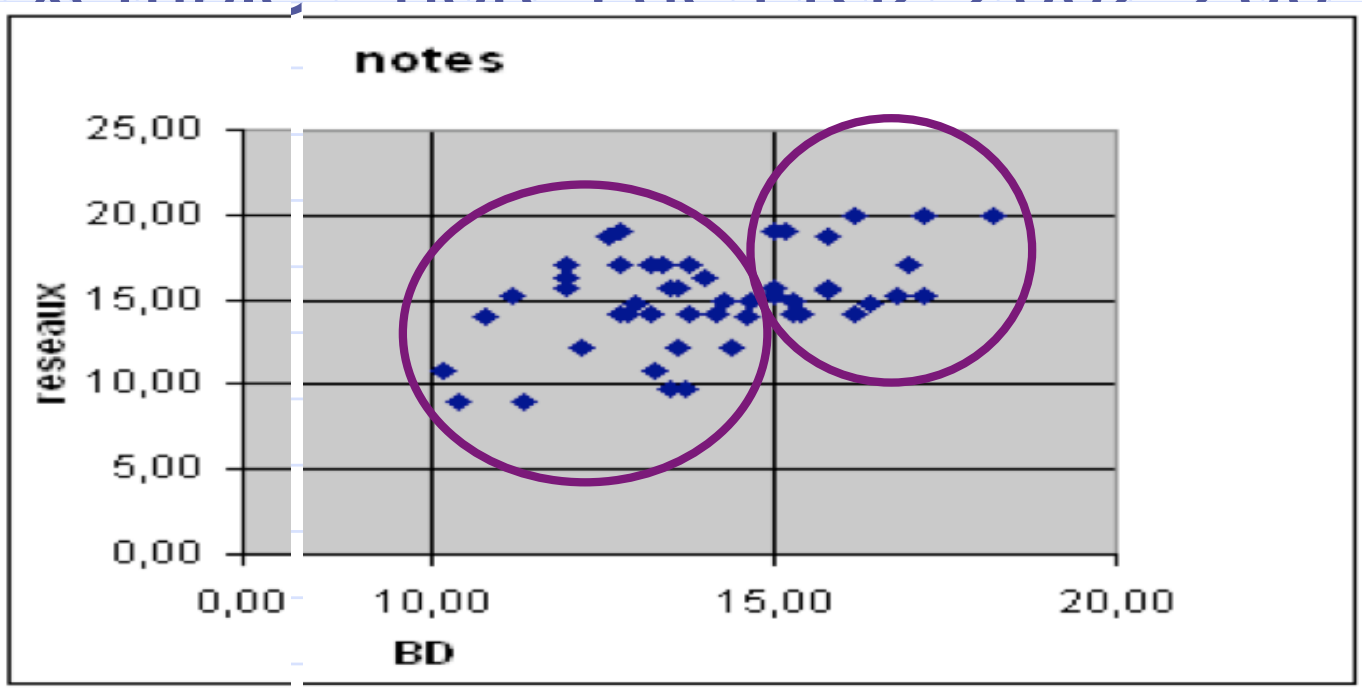
- Va-t-on jouer s'il y a du soleil, beaucoup d'humidité et pas de vent ?

Attributs continus

- ID3 ne les prend pas en charge
- discrétisation par un expert
- algorithme C4.5 :
  - test et calcul de l'entropie avec toutes les coupures possibles entre les valeurs successives
  - exemple sur les valeurs 3,4,8 on testera
    - < 3,5 vs. > 3,5
    - < 6 vs. > 6

Segmentation (Clustering)

- But : diviser la population en groupes
- Minimiser la similarité intra-groupe
- Maximiser la similarité inter-groupes
- Exemple : notes des IG2 2002-2003



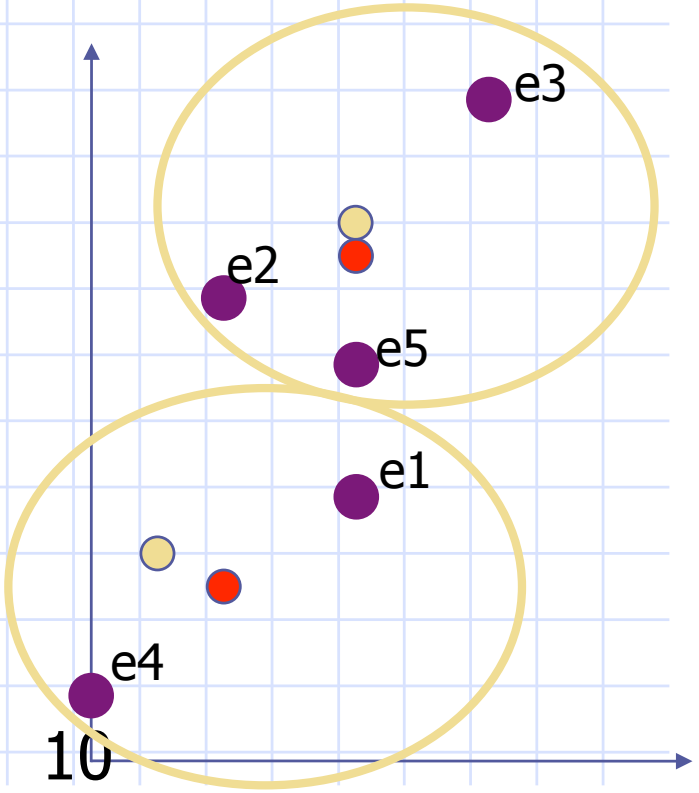
Algorithme des k-moyennes

1. Choisir le nombre de groupes à créer k
2. Choisir k centres initiaux c_1, .., c_k
3. Pour chaque exemple, l'affecter au groupe i dont le centre est le plus proche
4. Si aucun exemple ne change de groupe
5. Alors STOP
6. Sinon
   a) Calculer les nouveaux centres :
      Pour i = 1 à k
         c_i est la moyenne des éléments du groupe
   b) Aller en 3)

Exemple : faire 2 groupes d'étudiants

- Centres initiaux : c1=(11,13) c2=(14,18)
  - d(e1,c1)= $[(14-11)^2 + (14-13)^2]^{1/2}$= **3.16**
  - d(e1,c2) = $[(14-14)^2 + (14-18)^2]^{1/2} \approx 4$
  - d(e2,c1)= 4.12 d(e2,c2) ≈ **2.24**
  - d(e3,c1) > d(e3,c2)
  - d(e4,c1) < d(e4,c2)
  - d(e5,c1) > d(e5,c2)
- Nouveaux centres :
  - c'1 = ((14+10)/2,(14+11)/2) = (12,12.5)
  - c'2 = ((12+16+14)/3),(17+20+16)/3)=(14,17.6)
- calcul de d(e1,c'1) d(e1,c'2) …
- résultat inchangé ⇒ FIN

e1	14	14
e2	12	17
e3	16	20
e4	10	11
e5	14	16



Problèmes

- Nécessité de l'existence d'une distance
- Choix de k
- Influence du choix des centres initiaux sur le résultat
- Normalisation des attributs

Evaluation des méthodes

- Apprentissage supervisé : évaluation sur une base d'exemples test
- Méthodes de séparation entre les bases d'apprentissage et de test.
  - on dispose de deux bases séparées
  - on coupe la base en deux
  - validation croisée. Leave One Out.

Validation croisée

- Découpage de la base d'exemples en n sous-base b_1, ... , b_n
- n apprentissages :
  - On apprend sur n-1 sous-bases
  - On teste sur la sous-base restante
  - Moyenne des n résultats
- n = 10 fonctionne bien
- Leave one out

Critères d'évaluation

- Taux de bon apprentissage

Parmi tous les exemples, quelle proportion est bien classée ?

- Précision de la classe k

Parmi les exemples classés dans la classe k, quelle proportion est effectivement de la classe k ?

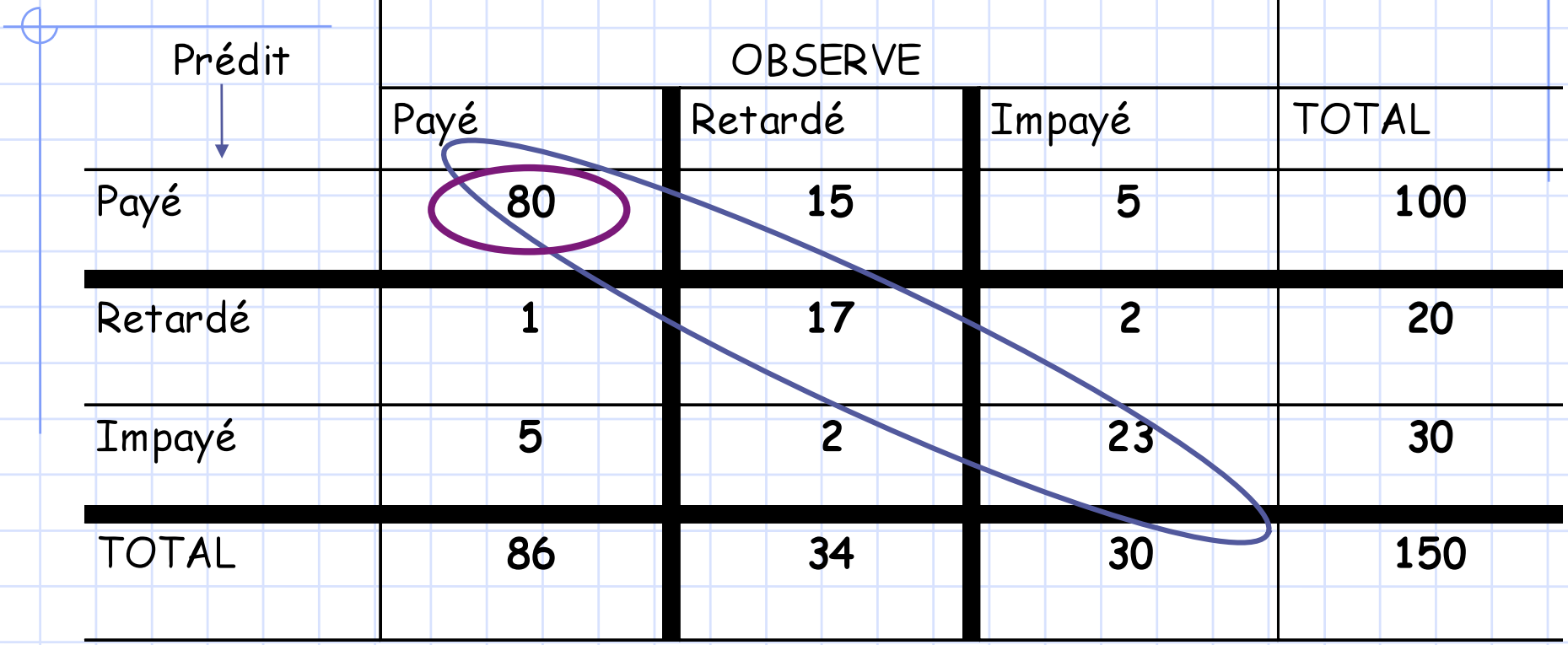
- Rappel de la classe k

Parmi les exemples de la classe k, quelle proportion se retrouvent classés dans la classe k ?

- Précision contre Rappel
- Matrice de confusion : table de contingence

Matrice de confusion

Prédit ↓	OBSERVE			
	Payé	Retardé	Impayé	TOTAL
Payé	80	15	5	100
Retardé	1	17	2	20
Impayé	5	2	23	30
TOTAL	86	34	30	150



- **Validité du modèle (taux d'apprentissage) :** nombre de cas exacts (=somme de la diagonale) divisé par le nombre total : 120/150 = 0.8

- **Rappel** de la classe Payé : nombre de cas prédits et observés « payé » divisé par le nombre total de cas observés « payés » : 80/86 = 0.93
- **Précision** de la classe Payé : nombre de cas observés et prédits « payé » divisé par le nombre total de cas prédits « payés » : 80/100 = 0.8

Traitement des données manquantes

- Attention à la sémantique :
  - La donnée peut-elle exister ?
- Plusieurs méthodes :
  - les oublier
  - les remplacer :
    - valeurs majoritaire
    - valeur moyenne
    - ...

Logiciels de fouille de données

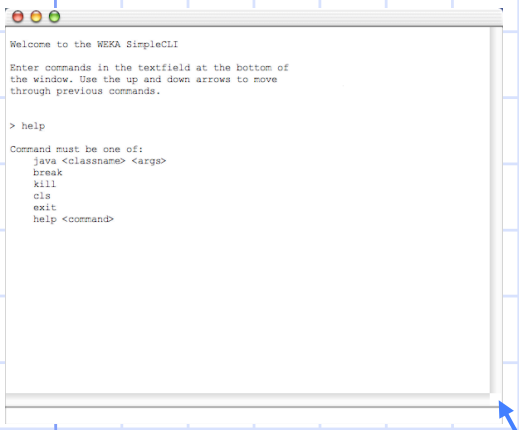
- Intelligent Miner (ww.ibm.com)
- Entreprise Miner (SAS Institute)
- MineSet (Silicon Graphics Inc.)
- Clementine (Integral Solutions Ltd, racheté par SPSS)
- DBMiner (www.dbminer.com)
- Oracle Darwin data mining suite
- Oracle Discoverer
- Weka
- Rainbow

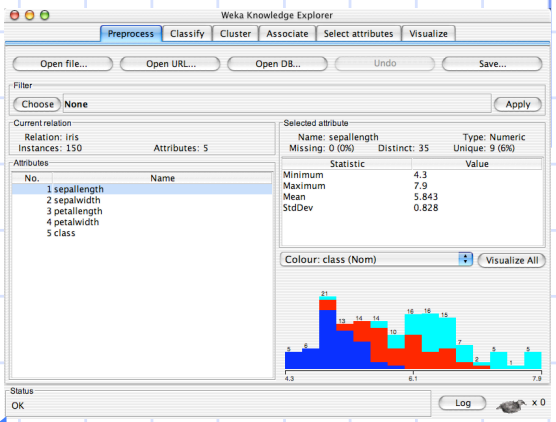
WEKA

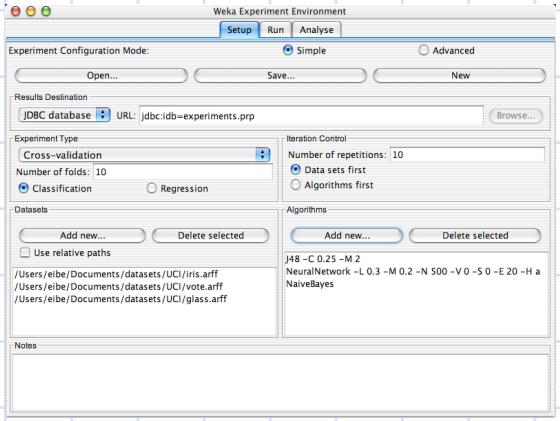
- Logiciel de data mining open source comprenant un grand nombre d'algorihtmes de fouille de données

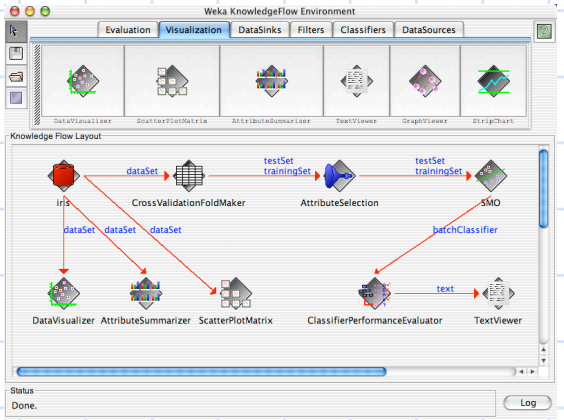
- Interface graphique ou utilisation des classes java (ligne de commande ou code java)

Interface graphique WEKA

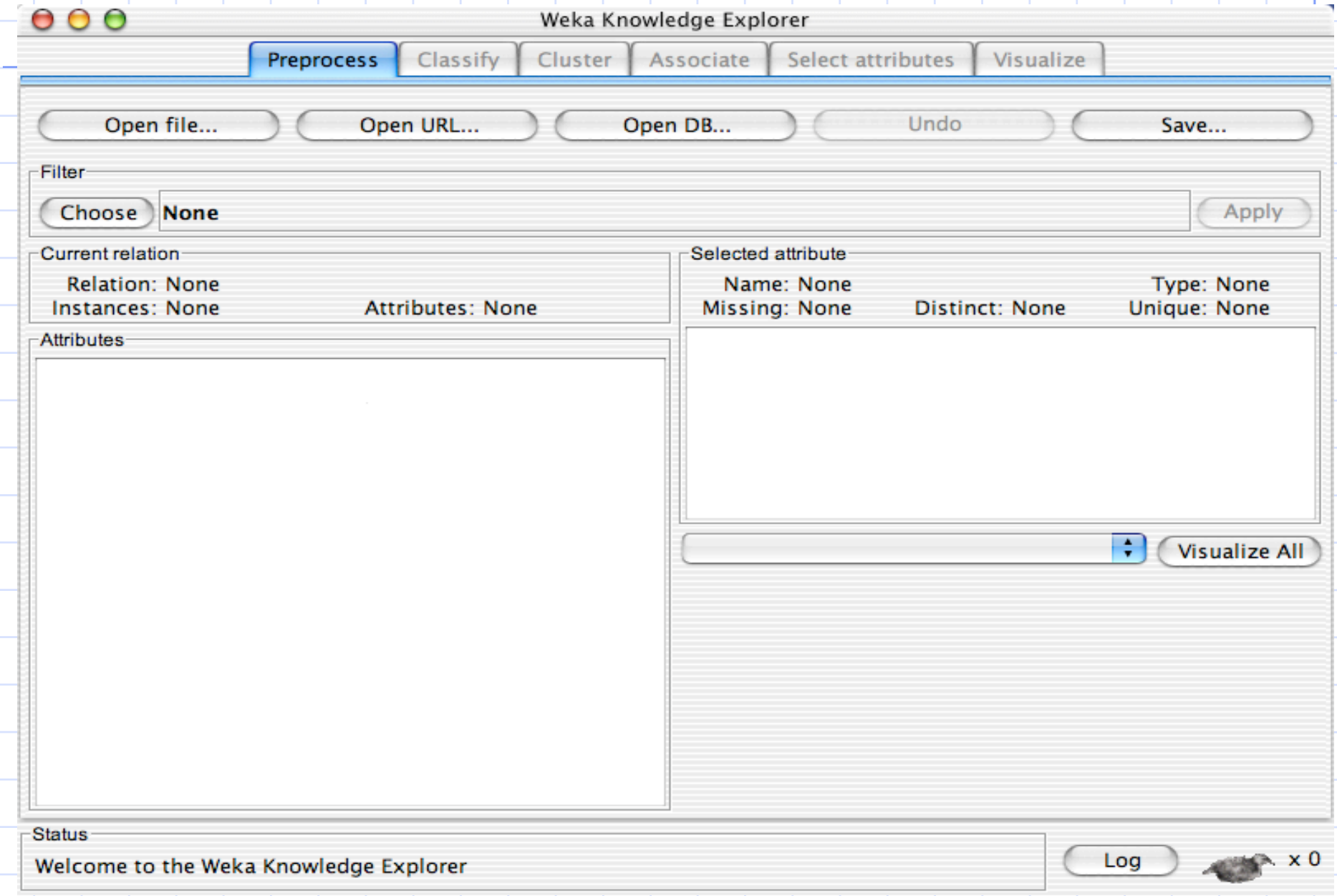




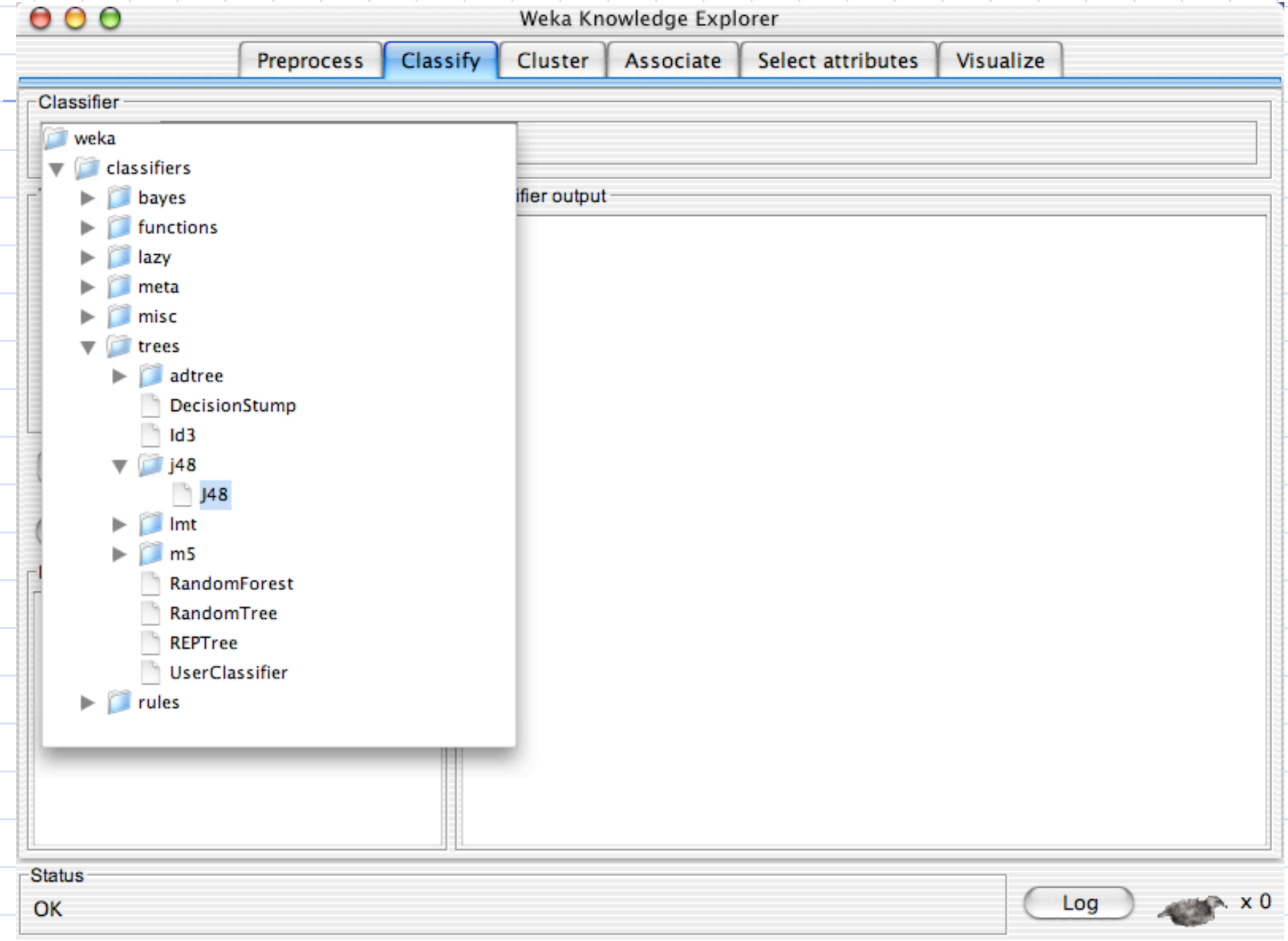




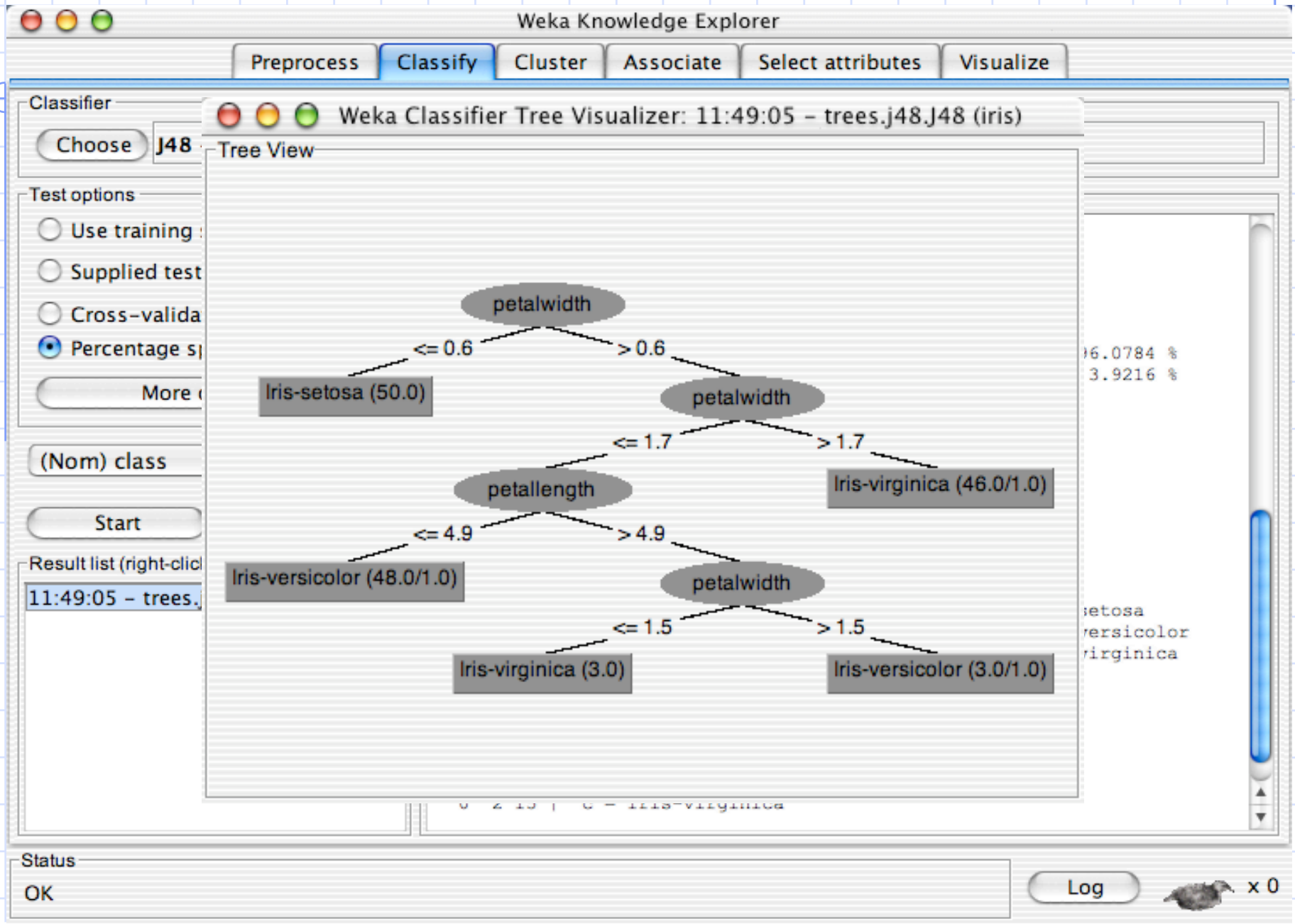
Explorer WEKA



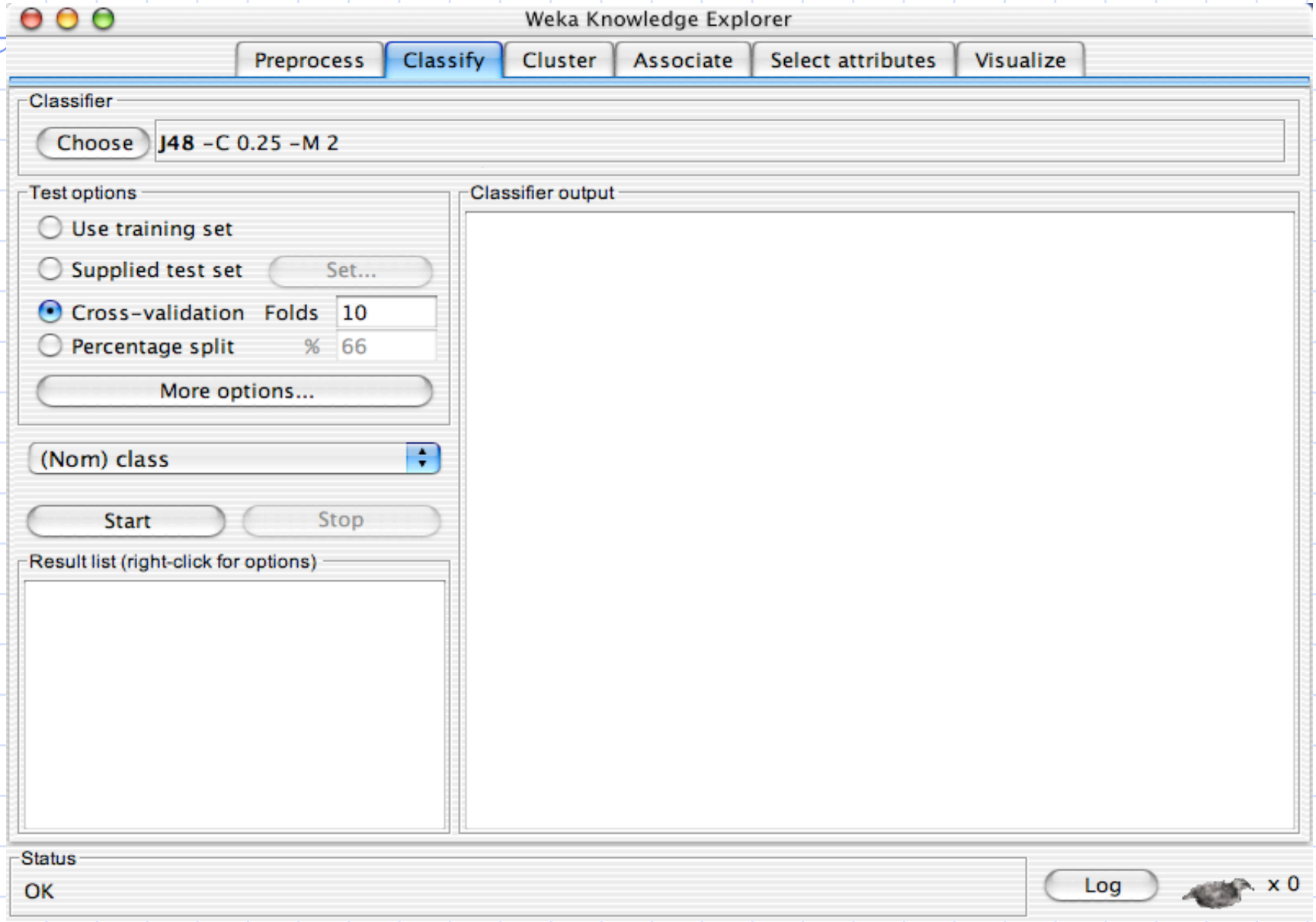
Classification



Exemple : arbres de décision C4.5



Evaluation


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
(Nom) class
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0

Matrice de confusion WEKA

```
Class attribute: play
Classes to Clusters:

 0 1  <-- assigned to cluster
 5 4 | yes
 3 2 | no

Cluster 0 <-- yes
Cluster 1 <-- no

Incorrectly clustered instances : 7.0   50%
```

Fouille de données et reporting

- Nombreux outils de reporting, tableaux de bord, ...
- Outils décisionnels OLAP (ETL – data warehouse/data marts)
- Mais pas d'automatisation

Conclusion

- il existe de nombreuses (autres) méthodes
- il n'y a pas de meilleure méthode
- méthode à choisir selon
  - les données (continues ? manquantes ? volumineuses ? denses ? ...)
  - la tâche
  - le temps de calcul dont on dispose
- règle du rasoir d'Ockham :

« pluralitas non est ponenda sine neccessitate »

« Les choses essentielles ne doivent pas être multipliées sans nécessité »

- autres types de données

Références. Bibliographie

- Livres :
  - *Introduction au Data Mining*. M.Jambu. Eyrolles. 1998.
  - *Data Mining: Concepts and Techniques*. J. Han and M. Kamber, The Morgan Kaufmann Series in Data Management Systems, 2000.
- Sites internet :
  - KD Nuggets
  - UCI machine learning repository