Agents qui apprennent à partir d'exemples

`http://www.grappa.univ-lille3.fr/grappa/index.php3?info=apprentissage`

- Introduction
- La problématique
- Méthodes symboliques
  - Arbres de décision
- Méthodes non symboliques ou adaptatives
  - Réseaux de neurones

Apprentissage à partir d'exemples

L'approche probabiliste. Exemple: Établir un diagnostic dans le domaine médical.

- Il faut être capable d'associer le nom d'une maladie à un certain nombre de symptômes présentés par des malades
- Les malades forment la population.
- Les symptômes sont les descriptions des malades.
- Les maladies sont les classes.
- On suppose qu'il y a un classement correct, c.-à-d. une application qui associe à tout malade une maladie.
- Apprendre à établir un diagnostic: Associer une maladie à une liste de symptômes de sorte que cette association corresponde au classement correct défini ci-dessus.

Définitions

- Π: la population
- D: l'ensemble des descriptions
- $Cl = \{1, \ldots, c\}$ l'ensemble des classes
- $X : \Pi \rightarrow D$: fonction qui associe une description à chaque élément de la population
- $Y : \Pi \rightarrow \{1, \ldots, c\}$: fonction de classement qui associe une classe à tout élément de la population.
- Une fonction $C : D \rightarrow Cl$ est appelée fonction de classement ou procédure de classification

Le but de l'apprentissage est de rechercher une procédure de classification $C : D \rightarrow Cl$ telle que $C \circ X = Y$ ou plutôt telle que $C \circ X$ soit une bonne approximation de Y.

Définition

- $A_1, \ldots A_n$: Ensembles d'attributs logiques, symboliques ou numériques qui sont de domaines $D_1, \ldots D_n$.
- $D = D_1 \times D_2 \ldots \times D_n$.
- Exemples:
  - Patient est décrit par:
    * Ensemble de symptômes
    * Suite de mesures: Température, etc.
  - Un client est décrit par l'ensemble des données: âge, sexe, etc.
- Comment exprimer le fait que $C \circ X$ est une bonne approximation de Y ?

Bonne approximation

- $C \circ X$ est une bonne approximation de Y si $C \circ X$ est rarement différente de Y
- On suppose l'existence d'une distribution de probabilités sur Π et on dit que $C \circ X$ est rarement différent de Y, si il est peu probable qu'ils diffèrent.
- Soient Π probabilisé, P la probabilité définie sur Π et D discret. Définitions:
  - $P(d) = P(X^{-1}(d))$: probabilité qu'un élément de Π ait d pour description.
  - $P(k) = P(Y^{-1}(k))$: probabilité qu'un élément de Π soit de classe k.
  - $P(d/k) = P(X^{-1}(d)/Y^{-1}(k))$: probabilité qu'un élément de classe k ait d pour description.
  - $P(k/d) = P(Y^{-1}(k)/X^{-1}(d))$: probabilité qu'un élément ayant d pour description soit de classe k.
  - Formule de Bayes: $$P(k/d) = \frac{P(d/k)P(k)}{P(d)}$$

On connaît $P(d)$, $P(k)$ et $P(d/k)$ pour tout $d \in D$ et $k \in \{1, \ldots, c\}$.

Comment choisir C ?

Exemple

- Π: la population française
- Un attribut logique: répondeur
- L'espace de description: $\{repondeur, \overline{repondeur}\}$
- Deux classes: $\{riche, \overline{riche}\}$
- Informations:

classe k	$riche$	$\overline{riche}$
$P(k)$	0.4	0.6
$P(repondeur/k)$	0.8	0.45

- C_{maj}: Attribuer à chaque description la classe majoritaire (ici: $\overline{riche}$)

$C_{vraisemblance}$

- Attribuer à une description d la classe pour laquelle cette observation est la plus probable si on observe d
- Pour l'exemple: $riche$ pour propriétaire de répondeur, $\overline{riche}$ pour les autres.
- Problème:
  - Supposons $Cl = \{employe\ Telecoms, medecins, ouvriers\}$ et $P(repondeur/employe\ Telecoms) = 1$.
  - $C_{vraisemblance}$ donne pour propriétaire d'un répondeur toujours $employe\ Telecoms$.

C_{Bayes}

- Attribuer à une description d la classe k qui maximise la probabilité $P(k/d)$ qu'un élément ayant d pour description soit de classe k.
- $P(k/d)$ peut être estimée en utilisant la formule de Bayes
- On choisit la classe k qui maximise le produit $P(d/k)P(k)$.
- Exemple:
  - $P(repondeur/riche)P(riche) = 0.8 \times 0.4 = 0.32$
  - $P(\overline{repondeur}/riche)P(riche) = 0.2 \times 0.4 = 0.08$
  - $P(repondeur/\overline{riche})P(\overline{riche}) = 0.45 \times 0.6 = 0.27$
  - $P(\overline{repondeur}/\overline{riche})P(\overline{riche}) = 0.55 \times 0.6 = 0.33$
- Ici: $C_{vraisemblance} = C_{Bayes}$
- Cas spécial: classes équiprobables
  $\Rightarrow$ toujours $C_{vraisemblance} = C_{Bayes}$

Comparaison

- On définit l'erreur $E(d)$ comme la probabilité qu'un élément de Π de description d soit mal classé par C
- $E(d) = P(Y \neq C/X = d)$
- L'erreur de classification
$$E(C) = \sum_{d \in D} E(d)P(X = d)$$
- Pour l'exemple:
  - $E(C_{maj}) = 0.4$
  - $E(C_{vraisemblance}) = E(rep)P(rep) + E(\overline{rep})P(\overline{rep}) = P(\overline{riche}/rep)P(rep) + P(riche/\overline{rep})P(\overline{rep}) = P(rep/\overline{riche})P(\overline{riche}) + P(\overline{rep}/riche)P(riche) = 0.27 + 0.08 = 0.35$
- Théorème: La règle de décision de Bayes est celle dont l'erreur de classification est minimale

Remarques

- Si une fonction de classement est correcte, alors $E(C_{Bayes}) = 0$
- Une fonction de classement correcte existe, ssi la probabilité que des individus appartenant à des classes différents aient des descriptions identiques est nulle.
- Dans ce cas, le problème est dit déterministe.
- Très rare en pratique:
  - Généralement les paramètres descriptifs dont on dispose ne sont pas suffisants pour classifier correctement tout.
  - Les données sont généralement inexactes.
- Il faut connaître les probabilités (difficiles à estimer).

Classification supervisée

- Apprentissage supervisé/non supervisé
- Choix du langage de description - définir les attributs susceptibles pertinents
- Langage de description $D = D_1 \times D_2 \times \ldots D_n$ fixé. $\vec{x}, \vec{y}$ sont des éléments de D, les classes $\{1, \ldots, c\}$ fixées
- On suppose une loi de probabilité P sur D fixée mais inconnue.
- On suppose une loi de probabilité conditionnelle $P(./.)$ fixée mais inconnue.
- Échantillon S de m exemples $(\vec{x}, c(\vec{x}))$ tirés selon $P(.,.)$ définie par $P(\vec{x}, y) = P(\vec{x})P(y/\vec{x})$.
- Problème de classification supervisée: Inférer une fonction de classement dont l'erreur de classification est petite.
- Parfois il faut pondérer les erreurs.

Bien classer et bien prédire

- Exemple d'une procédure de classification:
  - On mémorise tous les exemples de l'échantillon d'apprentissage dans une table.
  - Lorsqu'une nouvelle description est présentée au système, on recherche dans la table.
  - Si on trouve la description, on sort le résultat correspondant
  - Sinon, on choisit une classe au hasard.
- Cette procédure ne fait pas d'erreur sur les exemples.
- Mais son pouvoir prédictif est faible.
- Objectif: Une procédure de classification devrait dépasser au moins le pouvoir prédictif de la procédure majoritaire.

Erreur réelle et apparente

- L'erreur réelle est l'erreur de classification $E(C)$
- L'erreur apparente est définie par:
  - S un échantillon de taille m
  - C une procédure de classification
  - taux d'erreur apparent sur S est $E_{app}(C) = \frac{err}{m}$ où err est le nombre d'exemples de S mal classés par C.
- Il faut minimiser $E(C)$ mais l'apprenant ne connaît que $E_{app}(C)$ sur S.
- $E_{app}(C)$ tend vers $E(C)$ pour des échantillons de plus en plus grands.

Les méthodes de classification supervisée

- Problème difficile:
  - On ne connaît pas les lois de probabilité
  - L'espace de recherche (fonction de classement) est énorme
  - L'échantillon est petit
- Le classifieur naïf de Bayes
- Méthodes paramétriques et non paramétriques
- Minimiser l'erreur apparente
- Choix de l'espace des hypothèses
- Estimer l'erreur réelle
  - Utilisation d'un ensemble Test
  - Re-échantillonage

Le classifieur naïf de Bayes

- $C_{Bayes}(d) = \underset{k \in \{1,\ldots,c\}}{argmax}\ P(k/d) = \underset{k \in \{1,\ldots,c\}}{argmax}\ P(d/k)P(k)$
- $P(d/k)$ et $P(k)$ sont inconnues: $C_{Bayes}(\vec{d}) = \underset{k \in \{1,\ldots,c\}}{argmax}\ P((d_1,\ldots,d_n)/k)P(k)$
- Pour remplacer $P((d_1,\ldots,d_n)/k)$ et $P(k)$ par des estimations faites sur l'échantillon S on estime $P(k)$ par $\widehat{P}(k)$, la proportion d'éléments de classe k dans S.
- L'estimation de $P((d_1,\ldots,d_n)/k)$ est difficile. On fait l'hypothèse simplificatrice: les valeurs des attributs sont indépendantes connaissant la classe: $P((d_1,\ldots,d_n)/k) = \Pi_{i \in \{1,\ldots,n\}} P(d_i/k)$
- On estime $P(d_i/k)$ par $\widehat{P}(d_i/k)$ (proportion d'éléments de k ayant valeur d_i pour l'attribut i) $C_{NaiveBayes}(d) = \underset{k \in \{1,\ldots,c\}}{argmax}\ \Pi_{i \in \{1,\ldots,n\}} \widehat{P}(d_i/k) \times \widehat{P}(k)$
- Facile à mettre en oeuvre, fourni un seuil de performance pour les autres méthodes

Méthodes paramétriques et non paramétriques

- Méthodes paramétriques
  - On suppose que la loi de probabilité fait partie d'une famille paramétrée de distributions
  - On essaie d'estimer les paramètres (par exemple: la moyenne et l'écart-type pour la distribution normale)
- Méthodes non paramétriques
  - On fait aucune hypothèse
- étudiées en Statistique

Minimiser l'erreur apparente

- L'erreur apparente est une version très optimiste de l'erreur réelle
- Il y a beaucoup de fonctions de D dans $\{1, \ldots, c\}$. C'est impossible de toutes les explorer.
- On peut limiter la recherche d'une fonction à un espace d'hypothèses $\mathcal{C}$
- Éviter des hypothèses trop spécialisées, exemples:
  - Si l'espace d'hypothèses n'est pas restreint, on pourrait toujours choisir la procédure de classification donné comme exemple avec erreur apparente nulle.
  - Recherche d'une fonction polynôme dont la courbe représentative passe par n points.
  - Programme qui traduit un texte de 2000000 pages de l'anglais vers le français et comporte 4000000 pages de code.

Minimiser l'erreur apparente

- On prend un espace d'hypothèses $\mathcal{C}$. Soit C_{opt} la procédure optimale de $\mathcal{C}$.
- Problème difficile: approcher C_{opt}
- On peut choisir C_{emp} qui minimise l'erreur apparente
- Mais on ne peut pas à la fois sélectionner un classifieur à l'aide d'un ensemble d'apprentissage et juger sa qualité avec le même ensemble.
- On utilise un ensemble test

Choix d'un espace d'hypothèse

- Souvent on a des suites d'ensembles de procédures de classification $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \cdots \mathcal{C}_k \subseteq \cdots$ où k est une mesure de complexité du système d'apprentissage liée à la capacité.
- On essaie de trouver k de sorte que $C_{k,emp}$ ait le plus faible erreur réelle possible.
- On peut minimiser l'erreur apparente en complexifiant de plus en plus l'espace de recherche.
- Exemple: Pour la recherche d'une fonction polynôme dont la courbe représentative passe par n points on choisit comme espace d'hypothèse les polynômes de degrée k de plus en plus grand.

Estimer l'erreur réelle

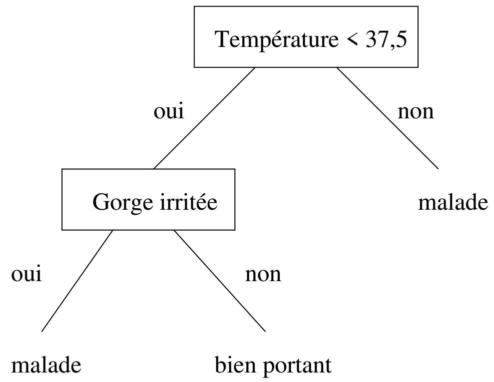
- Utilisation d'un ensemble Test
  - On partitionne l'échantillon en un ensemble d'apprentissage S et un ensemble test T
  - La répartition est faite aléatoirement.
  - On génère une procédure de classification C en utilisant S.
  - On estime avec $\widehat{E}(C)$ l'erreur réelle $E(C)$. $\widehat{E}(C)$ est donné par l'erreur apparente de C mesurée sur l'ensemble test T

$$\widehat{E}(C) = \frac{\sharp malclasses(T)}{\sharp T}$$

- Techniques de re-échantillonage
  - On divise l'échantillon on k parties. On fait k session d'apprentissage en utilisant $k-1$ parties pour apprendre et une partie pour tester.

Les arbres de décision

- Apprentissage symbolique
- On cherche une procédures de classification compréhensible
- Exemple:



Les arbres de décision

- Un arbre de décision est un arbre au sens informatique
- Les noeuds sont repérés par des positions $\in \{1, \ldots, p\}^*$, où p est l'arité maximale des noeuds.
- Les noeuds internes sont les noeuds de décision.
- Un noeud de décision est étiqueté par un test qui peut être appliqué à chaque description d'un individu d'une population.
- Chaque test examine la valeur d'un unique attribut.
- Dans les arbres de décision binaires on omet les labels des arcs.
- Les feuilles sont étiquetées par une classe.

Les arbres de décision

- À chaque arbre, on associe naturellement une procédure de classification.
- À chaque description est associée une seule feuille de l'arbre.
- La procédure de classification représenté par un arbre correspond à des règles de décision.
- Exemple:
  - SI Température $<$ 37,5 ET gorge irritée ALORS malade
  - SI Température $<$ 37,5 ET gorge non irritée ALORS malade
  - SI Température $\geq$ 37,5 ALORS malade

Construire des arbres de décision

- Étant donné un échantillon, on veut construire un arbre
- échantillon S, classes $\{1, \ldots, c\}$, arbre t
- À chaque position p de t correspond un sous-ensemble de S qui contient les éléments de S qui satisfont les tests de la racine jusqu'à p.
- On définit pour chaque p:
  - $N(p)$ = le cardinal de l'ensemble des exemples associé à p
  - $N(k/p)$ = le cardinal de l'ensemble des exemples associé à p de classe k
  - $P(k/p) = N(k/p)/N(p)$ = la proportion d'éléments de classe k à la position p

Exemple

- L'arbre de décision de l'exemple.
- On dispose d'un échantillon de 200 patients.
- 100 sont malades et 100 bien portants.
- Répartition (M malades, S bien portants):

	gorge irrité	gorge non irrité
température < 37,5	(6 S, 37 M)	(91 S, 1 M)
température ≥ 37,5	(2 S, 21 M)	(1 S, 41 M)

- On a:
  - $N(11) = 43$, (11 est la position de l'arbre qui corresponds à la feuille la plus à gauche)
  - $N(S/11) = 6$, $N(M/11) = 37$, $P(S/11) = \frac{6}{43}$, $P(M/11) = \frac{37}{43}$

Exemple: Banque

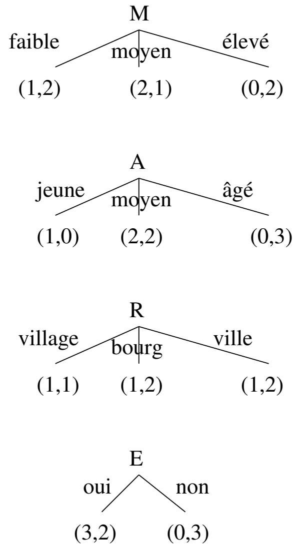
client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevé	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

- M: La moyenne des montants sur le compte
- A: Tranche d'âge du client
- R: Localité de résidence du client
- E: Études supérieures
- I: Consultation du compte par Internet
- On veut construire un arbre de décision pour savoir si un client consulte son compte par Internet.

Exemple (suite)

- On construit l'arbre d'une façon descendante
- On choisit un test, on divise l'ensemble d'apprentissage S et on réapplique récursivement l'algorithme.
- On initialise avec l'arbre vide.
- Des 8 éléments de S, 3 sont de classe oui et 5 de classe non.
- S est caractérisé par $(3, 5)$.
- Si le noeud n'est pas déjà terminal on choisit un test.
- Ici il y a 4 choix (M,A,R,E)

Les 4 choix



Quel test choisir ?

Mesurer le degré de mélange des exemples

- Un test est intéressant s'il permet une bonne discrimination.
- Une fonction qui mesure le degré de mélange des exemples doit
  - prendre son maximum lorsque les exemples sont équirépartis (ici par exemple (4,4)).
  - prendre son minimum lorsque les exemples sont dans une même classe ((0,8) ou (8,0)).
- Il existe plusieurs fonctions comme cela
- Entropie: $$Entropie(p) = -\sum_{k=1}^{c} P(k/p) \times log_2(P(k/p))$$
- Gini:
$$\begin{aligned} Gini(p) &= 1 - \sum_{k=1}^{c} P(k/p)^2 \\ &= 2 \sum_{k<k'} P(k/p)P(k'/p) \end{aligned}$$

Exemple pour attribut E

- $Entropie(\epsilon) = -\frac{3}{8}log_2\frac{3}{8} - \frac{5}{8}log_2\frac{5}{8} \simeq 0,954$
- $Entropie(1) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} \simeq 0,970$
- $Entropie(2) = -\frac{0}{3}log_2\frac{0}{3} - \frac{3}{3}log_2\frac{3}{3} = 0$
  (Attention: On considère $0 \times log_2(0) = 0$)
- $Gini(\epsilon) = 2 \times \frac{3}{8} \times \frac{5}{8} \simeq 0,469$
- $Gini(1) = 2 \times \frac{3}{5} \times \frac{2}{5} = 0,480$
- $Gini(2) = 2 \times \frac{0}{3} \times \frac{3}{3} = 0$

Choisir un test

- Soit f la fonction choisie (Gini ou Entropie par exemple)
- On définit le gain pour un Test choisi.
- n est l'arité du test et P_j la proportion d'éléments de S à la position p qui vont en position pj (qui satisfont la j-ème branche du Test)
- $$Gain(p,T) = f(p) - \sum_{j=1}^{n}(P_j \times f(pj))$$

Exemple

- $Gain(\epsilon, M) = Entropie(\epsilon)$
  $-(\frac{3}{8}Entropie(1) + \frac{3}{8}Entropie(2) + \frac{2}{8}Entropie(3)) =$
  $Entropie(\epsilon) - 0,620$
- $Gain(\epsilon, A) = Entropie(\epsilon)$
  $-(\frac{1}{8}Entropie(1) + \frac{4}{8}Entropie(2) + \frac{3}{8}Entropie(3)) =$
  $Entropie(\epsilon) - 0,500$
- $Gain(\epsilon, R) = Entropie(\epsilon)$
  $-(\frac{2}{8}Entropie(1) + \frac{3}{8}Entropie(2) + \frac{3}{8}Entropie(3)) =$
  $Entropie(\epsilon) - 0,870$
- $Gain(\epsilon, E) = Entropie(\epsilon)$
  $-(\frac{5}{8}Entropie(1) + \frac{3}{8}Entropie(2)) =$
  $Entropie(\epsilon) - 0,607$
- Gain maximal pour A.

Généralités

- Idée: Diviser récursivement et le plus efficacement possible les exemples de l'ensemble d'apprentissage par des tests définis à l'aide d'attributs, jusqu'à ce qu'on obtienne des sous-ensemble ne contenant (presque) que des exemples appartenant à une même classe.

- On a besoin des trois opérations suivantes:
  - Décider si un noeud est terminal.
  - Sélectionner un test à associer à un noeud.
  - Affecter une classe à une feuille

Algorithme générique

```
entrée: langage de description, échantillon S
Début
   Initialiser à l'arbre vide,
       la racine est le noeud courant
   répéter
      Décider si le noeud courant est terminal
      Si le noeud est terminal alors
         Affecter une classe
      sinon
         Selectionner test et créer sous-arbre
      Passer au noeud suivant non-exploré
   jusqu'à obtenir un arbre de décision
Fin
```


Généralités

- Arbre de décision parfait.

- Il n'existe pas toujours.

- Le meilleur arbre est l'arbre parfait le plus petit.

- L'algorithme précédent ne remet jamais en cause un choix effectué.

- L'erreur réelle peut être importante.

- En pratique, on construit l'arbre et ensuite on élague.

L'algorithme CART

- génère un arbre de décision binaire

- Langage de représentation: attributs binaires, qualitatifs, continus

- Attribut binaire: test binaire

- Attribut qualitatif: tout test qui partitionne en deux classes

- Attribut continus: infinité de tests possibles

- On suppose prédéfini un ensemble de tests binaires.

- Échantillon S, ensemble test T.

La phase d'expansion

- entrée: ensemble d'apprentissage A
- On utilise la fonction $Gini$
- Décider si un noeud est terminal:
  Un noeud p est terminal si $Gini(p) \leq i_0$ ou $N(p) \leq n_0$, où i_0 et n_0 sont des paramètres à fixer.
- Sélectionner un test à associer à un noeud:
  Soit p une position et T un test. P_g (resp. P_d est la proportion d'éléments qui vont sur le noeud $p1$ (resp. $p2$).

$$\Delta(p,T) = Gini(p) - (P_g \times Gini(p1) + P_d \times Gini(p2))$$

  On choisit le test qui maximise $\Delta(p,T)$
- affecter une classe à une feuille:
  On choisit la classe majoritaire.

La phase d'élagage

- On construit une suite d'arbre et choisit celui minimisant l'erreur apparent sur T.
- La suite est donné par $t_0 t_1 \dots t_k$ (avec t_0 l'arbre obtenu dans la phase d'expansion et t_k une feuille)
- On construit t_{i+1} à partir de t_i en utilisant A comme suit:
- On calcule $g(p) = \dfrac{\Delta_{app}(p)}{|u_p|-1}$ où u_p est le sous-arbre de t_i en position p et

$$\Delta_{app}(p) = \frac{MC(p) - MC(u_p)}{N(p)}$$

  où $N(p)$ est le nombre d'exemples de A associés à p, $MC(p)$ le nombre d'exemples mal-classés à p si on élague t_i en position p et $MC(u_p)$ le nombre d'exemples associés à p mal classés par u_p. On choisit la position qui minimise $g(p)$.
- Choix final: On choisit l'arbre dans la suite qui minimise l'erreur apparente de T.

L'algorithme C4.5

- Langage de représentation: comme CART mais des ensembles de tests n-aires.
- Décider si un noeud est terminal: p est terminal si tous les éléments associés à ce noeud sont dans une même classe où si on ne peut sélectionner aucun test.
- Sélectionner un test:
  - On envisage seulement les tests qui ont au moins deux branches contenant au moins deux éléments (ces paramètres peuvent être modifiés).
  - On choisit le test qui maximise le gain en utilisant la fonction entropie.
  - La fonction $Gain$ privilégie les attributs ayant une grand nombre de valeurs. On la modifie:

$$GainRatio(p,T) = \frac{Gain(p,T)}{Splitinfo(p,T)}$$

    avec $$Splitinfo(p,T) = -\sum_{j=1}^{n} P'(j/p) \times log_2(P'(j/p))$$

    où n est l'arité et $P'(j/p)$ la proportion des éléments présents à p prenant la j-ème valeur du test T.

C4.5

- Affecter une classe à une feuille:
  On attribue la classe majoritaire. S'il n'y a pas d'exemples on attribue la classe majoritaire du père.
- Phase d'élagage
- Améliorations:
  - Attributs discrets
  - Attributs continus
  - Valeurs manquantes