Ministère de l'Enseignement supérieur et de la recherche scientifique
Université Abderrahmane Mira de Bejaïa
Faculté des Sciences Exactes
Département de Mathématiques

Module: Data Mining **Année: 2022-2023**
Nature de document: Cours
Niveau : L3STID **N.BERMAD**
Chapitre: 2
Titre: Apprentissage Supervisé

1. Qu'est-ce qu'une donnée ?

1.1. Notations

On notera X un ensemble de données. Chaque donnée est décrite par un ensemble A d'attributs. Chaque attribut $a \in A$ prend sa valeur dans un certain ensemble de valeurs V_a. Ainsi, on peut considérer l'ensemble des données x dont les coordonnées balayent toutes les valeurs possibles des attributs: c'est l'espace des données que nous noterons D. Si l'on note $a_1,\dots,a_p$ les P attributs, $D=V_{a1}\times V_{a2}\times\dots\times V_{ap}$. Toute donnée appartient à cet ensemble et on a $x \subset D$.

Il est souvent utile d'avoir une représentation géométrique de l'espace des données ; chaque attribut correspondant à un axe de coordonnées. S'il y a P attributs, l'espace des données est un espace euclidien à P dimensions.

1

Il est souvent utile d'avoir une représentation géométrique de l'espace des données ; chaque attribut correspondant à un axe de coordonnées. S'il y a P attributs, l'espace des données est un espace euclidien à P dimensions.

1.2. Les différentes natures d'attributs

- Une donnée est un enregistrement au sens des bases de données, que l'on nomme aussi « individu » (terminologie issue des statistiques)
- Ou « instance » (terminologie orientée objet en informatique)
- Ou même « tuple » (terminologie base de données)
- Et « point » ou « vecteur » parce que finalement, d'un point de vue abstrait,

2

une donnée est un point dans un espace euclidien ou un vecteur dans un espace vectoriel.

- **Données continues (ou d'échelle)**

  - Dont les valeurs forment un sous-ensemble infini de R (exemple : salaire)

- **Données discrètes**

  - Dont les valeurs forment un sous-ensemble fini ou infini de N (exemple : nombre d'enfants)
  - Les données continues et discrètes sont des quantités :

    - ✓ On peut effectuer sur elles des opérations arithmétiques
    - ✓ Elles sont ordonnées (on peut les comparer par la relation d'ordre <)

3

- **Données catégorielles (ou qualitatives)**

  - Dont l'ensemble des valeurs est fini — ces valeurs sont numériques ou alphanumériques, mais quand elles sont numériques, ce ne sont que des codes et non des quantités (ex : PCS, n° de département)
  - Les données catégorielles ne sont pas des quantités
  - Mais sont parfois ordonnées : on parle de données catégorielles ordinales (exemple : « faible, moyen, fort »)
  - Données ordinales souvent traitées comme données discrètes
  - les données catégorielles nominales ne sont pas ordonnées

- **Données textuelles**

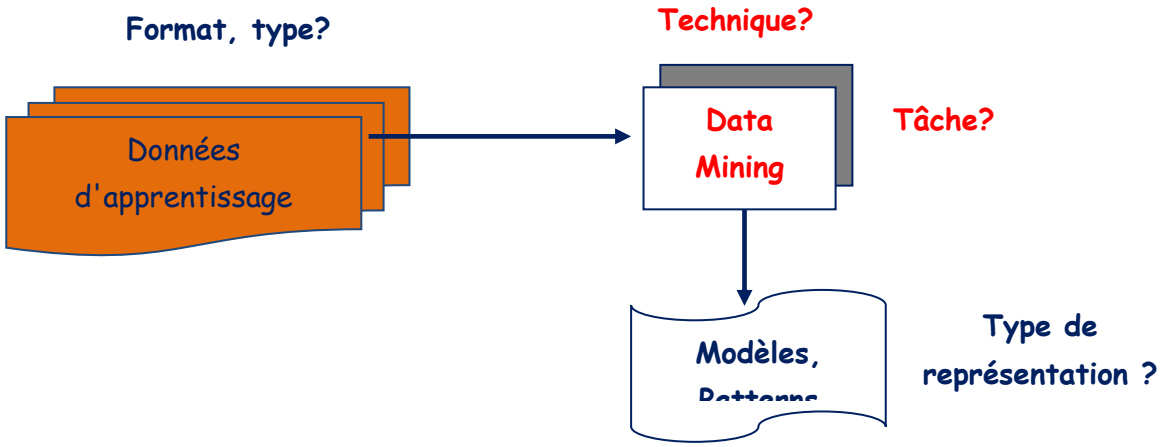
  - Lettres de réclamation, rapports, dépêches AFP…

4

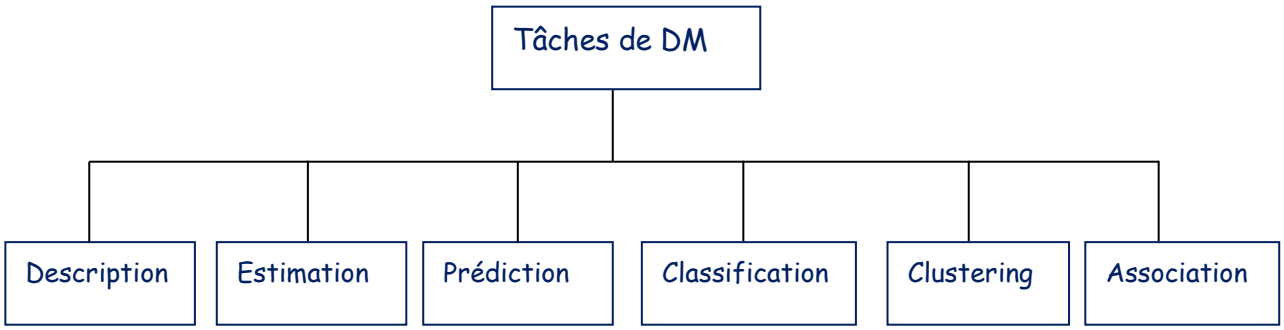
- Les données textuelles contiennent :
  - ✓ Des abréviations
  - ✓ Des fautes d'orthographe ou de syntaxe
  - ✓ Des ambiguïtés (termes dont le sens dépend d'un contexte
  - ✓ Non facilement détectable automatiquement).

2 . Les tâches de data Mining

2.1. Principe



Ils existent plusieurs méthodes de Data Mining :



2.2. La classification:

Définition :

> On dispose d'un ensemble X de N données étiquetées. Chaque donnée x_i est caractérisée par P attributs et par sa classe $y_i \in Y$. Dans un problème de classification, la classe prend sa valeur parmi un ensemble fini. Le problème consiste alors, en s'appuyant sur l'ensemble d'exemples $x = \{(x_i, y_i)_{i \in \{1,\dots,N\}}\}$, à prédire la classe de toute nouvelle donnée $x \in D$.

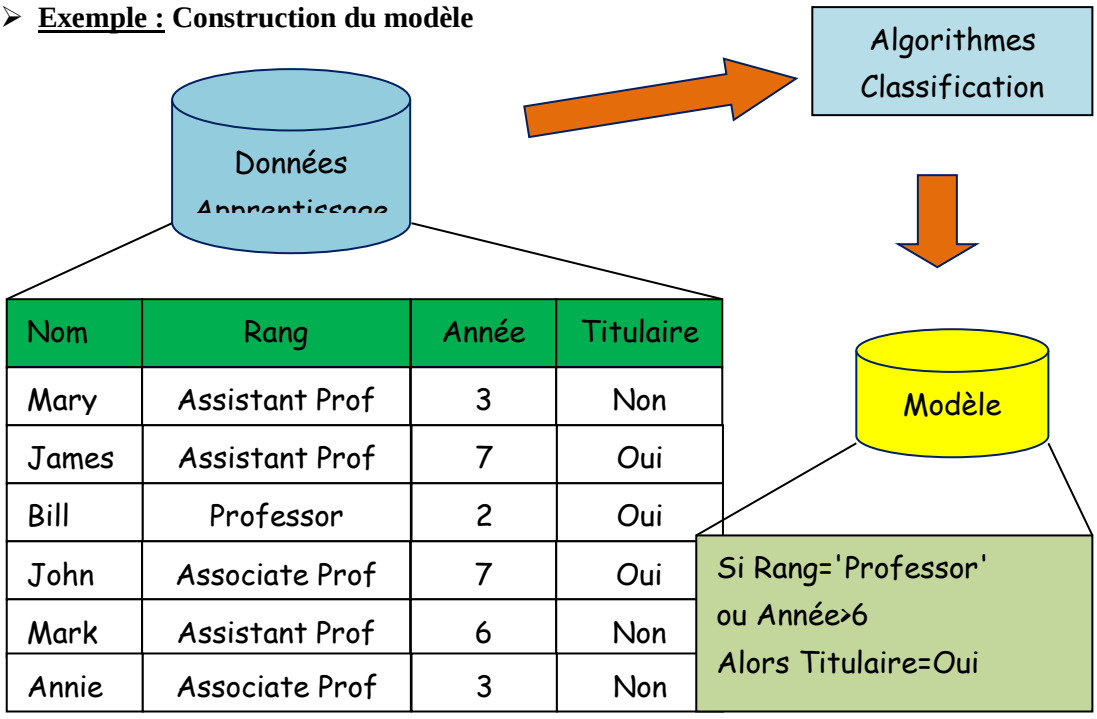
- La classification vise à prédire la classe d'appartenance d'un élément à partir de ses caractéristiques.
- C'est un apprentissage supervisé car les classes sont déterminées a priori.
- **Classes**
  - ✓ Groupes d'instances avec des profits particuliers
  - ✓ Possibilité de décider de l'appartenance d'une entité à une classe
- La classification traite des données catégoriques
- **Applications :**
  - ✓ Marketing direct (profils des consomateurs)
  - ✓ Grande distribution (classement des clients)
  - ✓ Médcine (malades/non malades),etc
- **Les thécniques les plus utilisées pour la classification :**
  - ✓ Les arbres de décision
  - ✓ Les réseaux de neuronnes
  - ✓ Les K-plus proches voisins
  - ✓ La classification bayésienne

- **2.3. Processus de Classification:** C'est un processus à deux étapes:
  - **La première étape :**
    - ✓ Construction du modèle à partir de l'ensemble d'apprentissage (training set)
    - ✓ Chaque instance est supposée appartenir à une classe
    - ✓ La classe d'une instance est déterminée par l'attribut "classe "
    - ✓ L'ensemble des instances d'apprentissage est utilisé dans la construction du modèle
    - ✓ Le modèle est représenté par des règles de classification, arbres de décision, formules mathématiques,…

- **La deuxième étape**

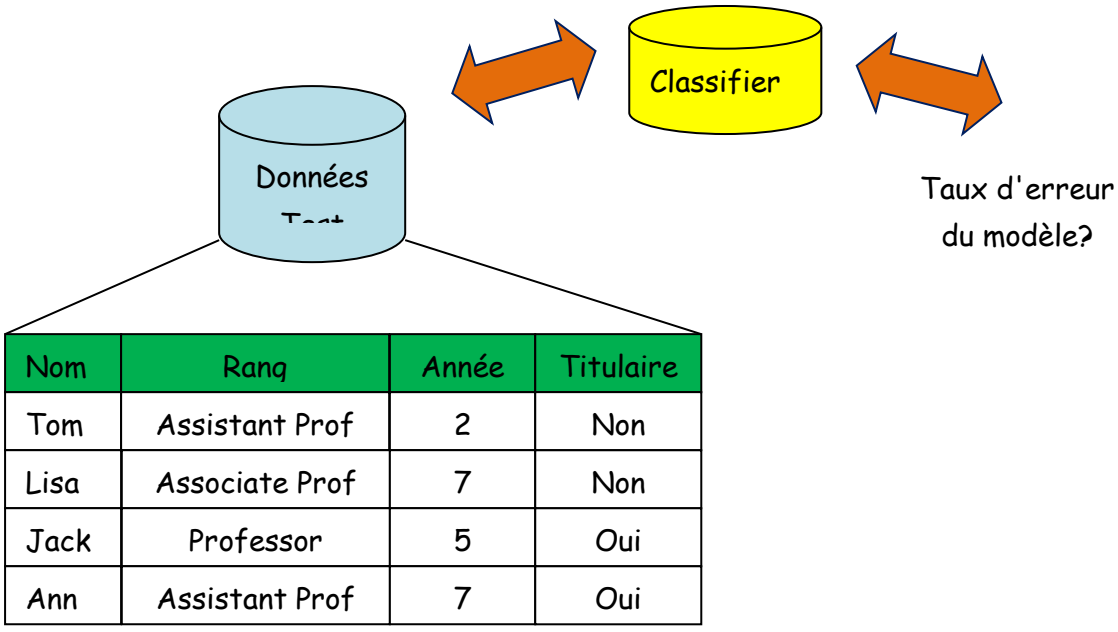
  - ✓ Utilisation du modèle : tester la précision du modèle et l'utiliser dans la classification de nouvelles données.
  - ✓ Classification de nouvelles instances ou instances inconnues
  - ✓ Estimer le taux d'erreur du modèle :
    - ➢ La classe connue d'une instance test est comparée avec le résultat du modèle
    - ➢ Taux d'erreur = pourcentage de tests incorrectement classés par le modèle

➢ **Exemple : Construction du modèle**

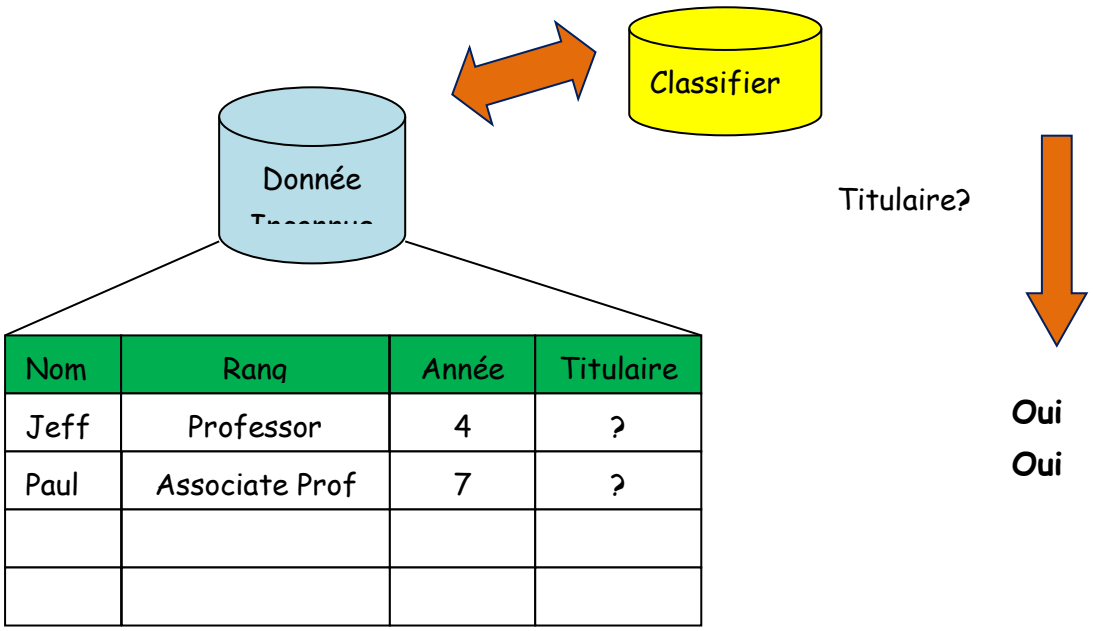


Nom	Rang	Année	Titulaire
Mary	Assistant Prof	3	Non
James	Assistant Prof	7	Oui
Bill	Professor	2	Oui
John	Associate Prof	7	Oui
Mark	Assistant Prof	6	Non
Annie	Associate Prof	3	Non

➢ **Exemple : utilisation du modèle**



Nom	Rang	Année	Titulaire
Tom	Assistant Prof	2	Non
Lisa	Associate Prof	7	Non
Jack	Professor	5	Oui
Ann	Assistant Prof	7	Oui



Nom	Rang	Année	Titulaire
Jeff	Professor	4	?
Paul	Associate Prof	7	?

- **2.4. Validation de la Classification**
  - **Estimation des taux d'erreurs :**
    - ✓ Partitionnement : apprentissage et test (ensemble de données important)
    - ✓ Utiliser 2 ensembles indépendants, e.g. ensemble d'apprentissage (2/3), ensemble test(1/3)

Apprentissage(D_t)	Validation (D/D_t)

  - **Validation croisées**
    - ✓ Diviser les données en K sous –ensembles
    - ✓ Utiliser $K-1$ sous-ensembles comme données d'apprentissage et un sous ensemble comme données test.
    - ✓ Une méthode plus sophistiquée. Pour cela, on découpe l'ensemble des exemples en n sous-ensembles mutuellement disjoints. Il faut prendre garde à ce que chaque classe apparaisse avec la même fréquence dans les n sous-ensembles (stratification des échantillons). Si $n = 3$, cela produit donc 3 ensembles A, B et C. On construit l'arbre de décision $AD_{A\cup B}$ avec $A\cup B$ et on mesure son taux d'erreur sur C, c'est-`a-dire, le nombre d'exemples de C dont la classe est mal prédite par $AD_{A\cup B}$: EC. Ensuite, on construit l'arbre de décision $AD_{B\cup C}$ avec $B \cup C$ et on mesure l'erreur sur A : EA. Enfin, on construit l'arbre de décision $AD_{A\cup C}$ avec $A \cup C$ en mesurant l'erreur sur B : EB. Le taux d'erreur E est alors estimé par la moyenne de ces trois erreurs mesurées $E = EA+EB+EC$ Habituellement, on prend $n = 10$. Cette méthode est dénommée « validation croisée en n-plis » (n-fold cross-validation).

- **2.5. Méthodes de classification**
  - **2.5.1.Méthodes des plus proches voisins**
    - Méthode dédiée à la classification (K-NN : nearset neighbor)
    - Méthode de raisonnement à partir de cas : prendre des décisions en recherchant un ou des cas similaires déjà résolus
    - Pas d'étape d'apprentissage : construction d'un modèle à partir d'un échantillon d'apprentissage (réseaux de neurones, arbres de décision,….)
    - Modèle =échantillon d'apprentissage + fonction de distance+fonction de choix de la classe en fonction des classes des voisins les plus proches.

➢ **2.5.1.1. Algorithme kNN : sélection de la classe**

✓ **Solution simple :** rechercher le cas le plus proche et prendre la même décision (Méthode 1-NN)

✓ Combinaison des K classes :

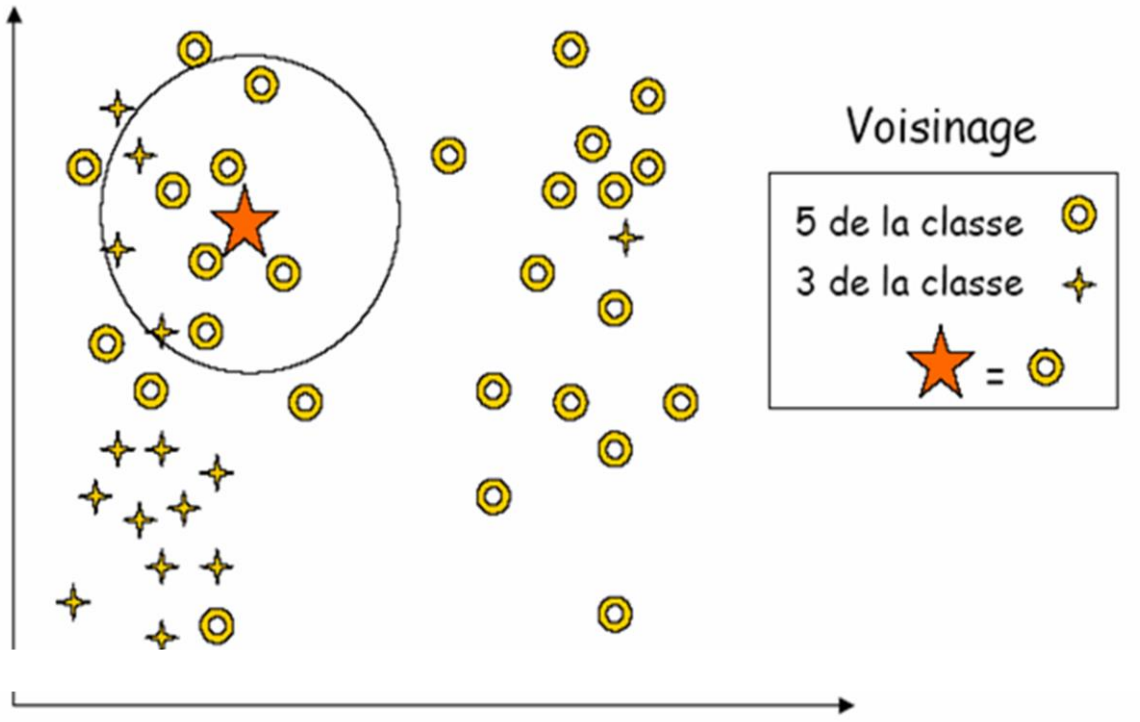
- Heuristique : K=nombre d'attribut+1
- Vote Majoritaire : prendre la classe majoritaire.

- ✓ Vote Majoritaire pondéré : chaque classe est pondérée, le poids de $c(X_i)$ est inversement proportionnel à la distance $d(Y, X_i)$

✓ Confiance : définir une confiance dans la classe attribuée = rapport entre les votes gagnants et le total des votes.

✓ La valeur de K est choisie de façon à obtenir la meilleure classification possible

➢ **2.5.1.2. Exemple:**



➢ **2.5.1.3. Avantages et limites:**

- ➢ Pas d'apprentissage : introduction de nouvelles données ne nécessite pas la reconstruction du modèle.
- ➢ Clarté des résultats
- ➢ Tout type de données
- ➢ Temps de classification est trop long
- ➢ Beaucoup d'espace mémoire pour stocker le modèle
- ➢ Distance et nombre de voisins : dépend de la distance, du nombre de voisin et du mode de combinaisons.

- **2.5.2.Classification Bayésienne**

Définition:

On suppose que n classes $C_1, C_2,\ldots,C_n$ sont définis a priori pour un ensemble d'élément .La règle de décision pour classifier un nouvel élément e est :

Classer e dans C_i si $P(C_i|e)>P(C_j|e) \forall\, i \neq j$

$P(C_i|e)$ est la probabilité que l'élément appartienne à la classe Ci sachant que e a été observé.

La règle de Bayes permet de déterminer $P(C_i|e)$ indirectement par :

$$P(C_i|e)=P(e|C_i)P(C_i)/P(e)$$

$P(e|\ Ci)$ est la probabilité d'observer e sachant que sa classe d'appartenances est Ci. $P(Ci)$ est la probabilité que l'élément appartienne à la classe .$P(e)$ est la probabilité d'observer e. Comme $P(e)$ est fixe, on peut le laisser tomber dans les calculs de la règle de décision qui devient :

Classer e dans C_i si $P(e|\ C_i)\ p(C_i) > P(e|\ C_j)\ P(C_j)\ \forall\, i \neq j$

Exemple

Supposons que l'on veuille déterminer le profil des utilisateurs d'Internet à partir des données de la figure ci-dessus. Ce profil pourrait servir à identifier des clients potentiels pour un fournisseur de service Internet. Les trois attributs, sexe, âge et revenu, identifient les caractéristiques des individus. La dernière colonne correspond à la classe de l'individu (Internet oui ou non)

Sexe	Age	Revenu	Internet
m	jeune	faible	non
m	jeune	élevé	oui
f	jeune	élevé	oui
m	jeune	faible	oui
f	vieux	élevé	non
f	vieux	faible	non
f	jeune	faible	non
m	vieux	élevé	oui
m	vieux	faible	non
f	vieux	faible	non

Soit un nouvel individu e, dont les caractéristiques sont : (*sexe=m, âge=vieux, revenu=élevé*).

La règle de décision est de classer l'individu dans la classe *internet* = oui si

P(internet=oui| sexe=m, âge =vieux, revenu=élevé)>
P(internet=non| sexe=m, âge =vieux, revenu=élevé)

En appliquant la règle de Bayes , la règle devient :classer l'individu dans la classe internet=oui si

P(sexe=m , âge =vieux, revenu=élevé| internet=oui)P(internet=oui)>
P(sexe=m, âge =vieux, revenu=élevé internet=non)P(internet=non).|

P(Ci) peut être estimé par la fréquence relative de la classe dans les données.

P(internet=oui)=4/10
P(internet=non)=6/10

Lorsqu'un élément e est représenté par un vecteur d'attributs $e_1, e_2, \ldots\ldots\ldots\ldots e_m$, $P(e|C_i)$ peut être estimée en supposant que les attributs sont indépendants (classification de Bayes naïve) :

$$P(e\ |Ci)=\prod_{j=1\ldots m} P(ej\,|Ci)$$

Pour des données nominales, $P(e_j|C_i)$ peut être estimée par la fréquence relative de la valeur e_j dans la classe C_i. Pour des données continues, l'estimation peut être basée sur une connaissance a priori de la distribution de probabilité (par exemple la distribution normale)

Dans notre exemple de données nominales,

P*(sexe=m, âge= vieux, revenu=élevé| internet=oui)= P(sexe=m |internet=oui) P(âge =vieux|internet=oui)* P *(revenu=élevé |internet=oui)*
*=3/4*1/4*3/4=9/64*

P*(sexe=m, âge=vieux, revenu=élevé|internet=non)=P (sexe=m|internet=non) P(âge=vieux|internet=non) P (revenu=élevé|internet=non)*

=2/6*4/6*1/6=1/27

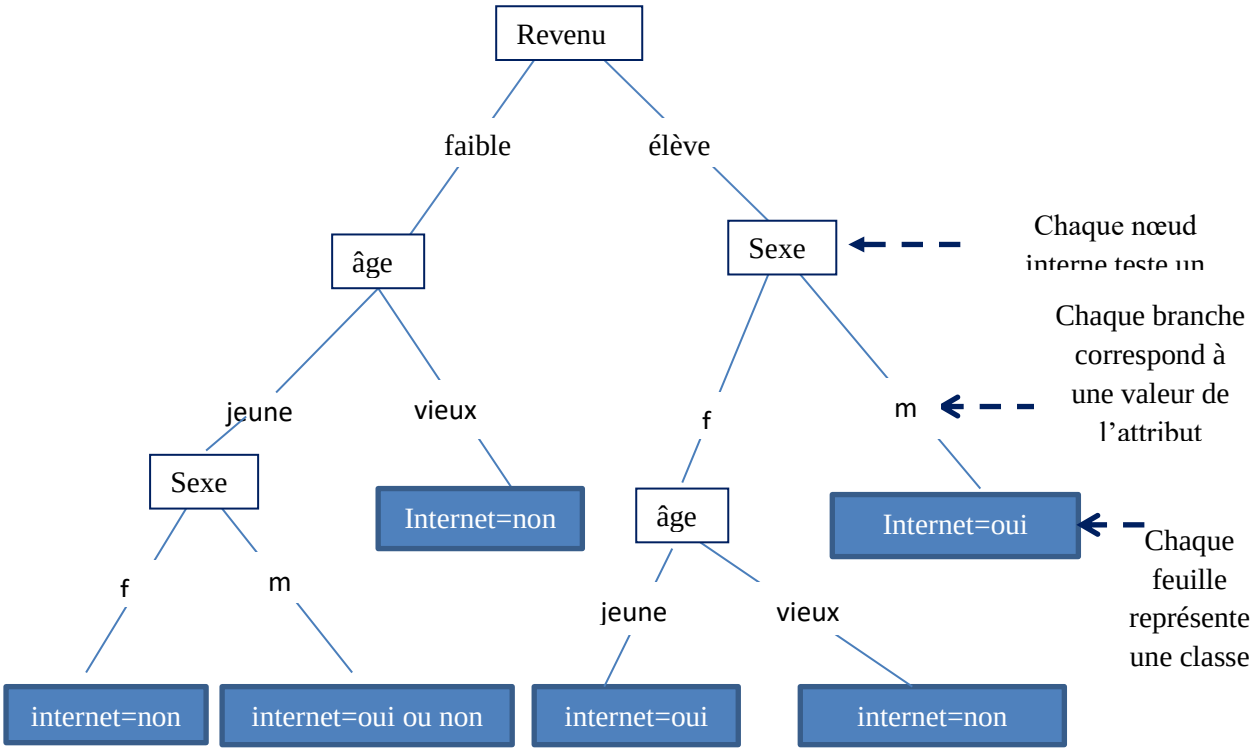
On obtient : 9/64*4/10=0.05625>1/27*6/10=0.022

L'individu est donc placé dans la classe internet=oui

- **2.5.3. Arbre de décision**
  - Un arbre de décision découpe un ensemble d'éléments en parties en fonction des valeurs d'attributs
  - Arbre= représentation graphique d'une procédure de classification
  - Un arbre de décision est un arbre où :
    - ✓ Nœud interne =attribut
    - ✓ Branche d'un nœud=un test sur un attribut
    - ✓ Feuilles=classes données
  - L'arbre de décision peut être ensuite exploité de différentes manières :
    - ✓ en y classant de nouvelles données
    - ✓ en faisant de l'estimation d'attribut
    - ✓ en extrayant un jeu de règles de classification concernant l'attribut cible
    - ✓ en interprétant la pertinence des attributs

Exemple

La figure ci-dessous montre un arbre de décision pour l'exemple du profil Internet. Chacun des nœuds représente une partition des éléments selon un attribut. Chacune des valeurs produit une branche de la partition. L'étiquette de la branche est la valeur de l'attribut



- **2.5.3.1. De l'arbre de décision aux règles de classification**
  - Une règle est générée pour chaque chemin de l'arbre (de la racine à une feuille)
  - Les paires attribut-valeur d'un chemin forment une conjonction
  - Le nœud terminal représente la classe prédite

  Exemples
  Quelques règles de l'arbre précédent sont **:**

  R1 :**SI** (*revenu=faible*) **ET** (*âge =jeune*) **ET** (*sexe=f*) **Alors** *internet=non*
  R2 :**SI** (*revenu=faible*) **ET** (*âge =vieux*) **Alors** *internet=non*
  R3 :**SI** (*revenu=élevé*) **ET** (*sexe =f*) **ET** (âge=jeune) **Alors** *internet=oui*
  R4 :**SI** (*revenu=élevé*) **ET** (*sexe =m*) **ET Alors** *internet=oui*
- **2.5.3.2. Construction de l'arbre de décision**
  - Deux phases dans la génération de l'arbre :
    - ✓ Arbre peut atteindre une taille élevée
  - Elaguer l'arbre (pruning)
    - ✓ Identifier et supprimer les branches qui représentent du bruit pour améliorer le taux d'erreur
  - Construction de l'arbre
    - ✓ Au départ, toutes les instances d'apprentissage sont à la racine de l'arbre
    - ✓ Sélectionner un attribut et choisir un test de séparation (split) sur l'attribut, qui sépare ''mieux '' les instances
    - ✓ Partitionner les instances entre les nœuds fils suivant la satisfaction des tests logiques
  - Traiter chaque nœud fils de façon récursive
  - Répéter jusqu'à ce que tous les nœuds soient des terminaux. Un nœud courant est terminal si :
    - ✓ Il n'y a plus d'attributs disponibles
    - ✓ Le nœud est ''pur'', c à d toutes les instances appartiennent à une seule classe
    - ✓ Le nœud est ''presque ''pur'', c à d la majorité des instances appartiennent à une seule classe
    - ✓ Nombre minimum d'instances par branche
  - Etiqueter le nœud terminal par la classe majoritaire.
  - Elaguer l'arbre obtenu (pruning)
    - ✓ Supprimer les sous arbres qui n'améliorent pas l'erreur de la classification
    - ✓ Eviter le problème de sur –spécialisation (overfitting), c à d , on a appris ''par cœur'' l'ensemble d'apprentissage , mais on n'est pas capable de généraliser.

2.5.3.2.1 Algorithmes de construction de l'arbre de décision

a. L'algorithme ID3

- Construction récursive d'un arbre de manière ''diviser pour régner'' descendante
- Attributs considérés énumératifs
- **Mesures de sélection d'attributs**
  - ✓ Sélectionner l'attribut avec le plus grand gain d'information
  - ✓ L'entropie E d'une partition $C_1, C_2, \ldots\ldots\ldots\ldots C_n$ mesure le niveau d'incertitude au sujet de la classe d'appartenance

$$E(C_1, C_2, \ldots\ldots\ldots\ldots) = -\sum_{i=1}^{n} p_i \log_2(p_i) \quad \text{où} \quad p_i = |c_i| / (\sum_{j=1}^{n} |C_j|)$$

  - ✓ Plus les éléments sont répartis également entre les classes, plus l'entropie est élevée
  - ✓ L'entropie d'un attribut est l'entropie de chacune des parties obtenues

$$Entropie(A) = \sum_{v \in valeur\,(aj)} \frac{\left|X_{a_j=v}\right|}{|X|} E\left(X_{a_j=v}\right)$$

  - ✓ Le gain en information de la partition par l'attribut A est l'entropie de la partition de l'attribut de la classification moins l'entropie de l'attribut :

$$Gain(A) = E(C_1, C_2, \ldots, C_n) - Entropie(A)$$

Algorithme 1 ID3

Nécessite: 2 paramètres: l'ensemble d'exemples X, l'ensemble d'attribut $A=\{a_{j\in\{1,\dots,p\}}\}$ où P est le nombre d'attributs restants à considérer

Créer un nœud racine

Si tous les éléments de X sont positifs **alors**

racine.étiquette $\leftarrow \oplus$

return racine

fin si

Si tous les éléments de X sont négatifs **alors**

racine.étiquette $\leftarrow \ominus$

return racine

fin si

Si $A=\emptyset$ **alors**

racine.étiquette $\leftarrow$ valeur la plus présente de la classe parmi les X

return racine

fin si

$a^* \leftarrow \arg\max_{a\in A} gain(X, a)$

racine.étiquette $\leftarrow a^*$

pour toutes les valeurs v_i de a^* **faire**

ajouter une branche à racine correspondant à la valeur v_i

former $X_{a^*= v_i} \subset X$ dont l'attribut a^* vaut v_i

si $X_{a^*= v_i} = \emptyset$ **alors**

à l'extrémité de cette branche, mettre une feuille étiquetée avec la valeur la plus présente de la classe parmi les X

sinon

à l'extrémité de cette branche, mettre ID3 $(X_{a^*= v_i}, A-\{a^*\})$

fin si

fin pour

return racine

Exemple

En considérant les données,

- ✓ l'entropie de la partition produite par l'attribut de classification internet est :

$E(C_{internet=oui}, C_{internet=non}) = -(0.4\ log_2(0.4) + 0.6\ log_2(0.6)) = 0.97$

- ✓ Pour partitionner l'ensemble des éléments de départ, on doit choisir parmi les trois attributs, sexe, âge, et revenu, celui dont le gain est maximal :

Gain en information de sexe :

- **Sexe=m:**
  internet=oui : 3/5=0.6
  internet=non : 2/5=0.4
- **Sexe=f :**
  internet=oui : 1/5=0.2
  internet =non :4/5=0.8

$Gain(sexe) = 0.97 - (0.5 * -(0.6\ log_2(0.6) + 0.4\ log_2(0.4)) + 0.5 * -(0.2\ log_2(0.2) + 0.8\ log_2(0.8))) = 0.12$

Gain en information de âge :

- **âge = jeune :**
  internet=oui : 3/5=0.6
  internet=non :2/5=0.4
- **âge = vieux :**
  internet=oui : 1/5=0.2
  internet=non :4/5=0.8

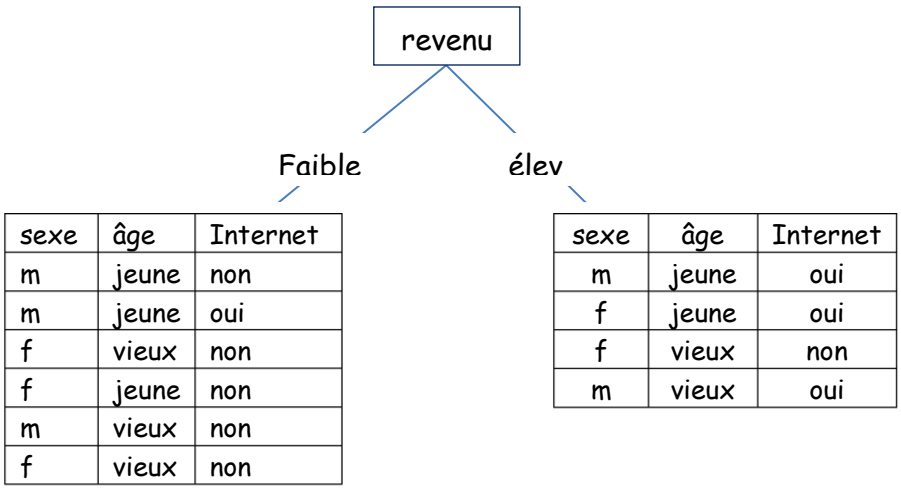
$Gain(âge) = 0.97 - (0.5 * -(0.6\ log2(0.6) + 0.4\ log2(0.4)) + 0.5 * -(0.2\ log2(0.2) + 0.8\ log2(0.8))) = 0.12$

Gain en information de revenu :

- **revenu=faible :**
  internet=oui : 1/6=0.17
  Internet=non : 5/6=0.83
- **revenu=élevé**
  internet=oui : 3/4=0.75
  Internet=non : 1/4=0.25

$Gain(revenu) = 0.97 - (6/10 * -(1/6\ log2(1/6) + 5/6\ log2(5/6)) + 4/10 * -(0.75\ log2(0.75) + 0.25\ log2(0.25))) = 0.26$

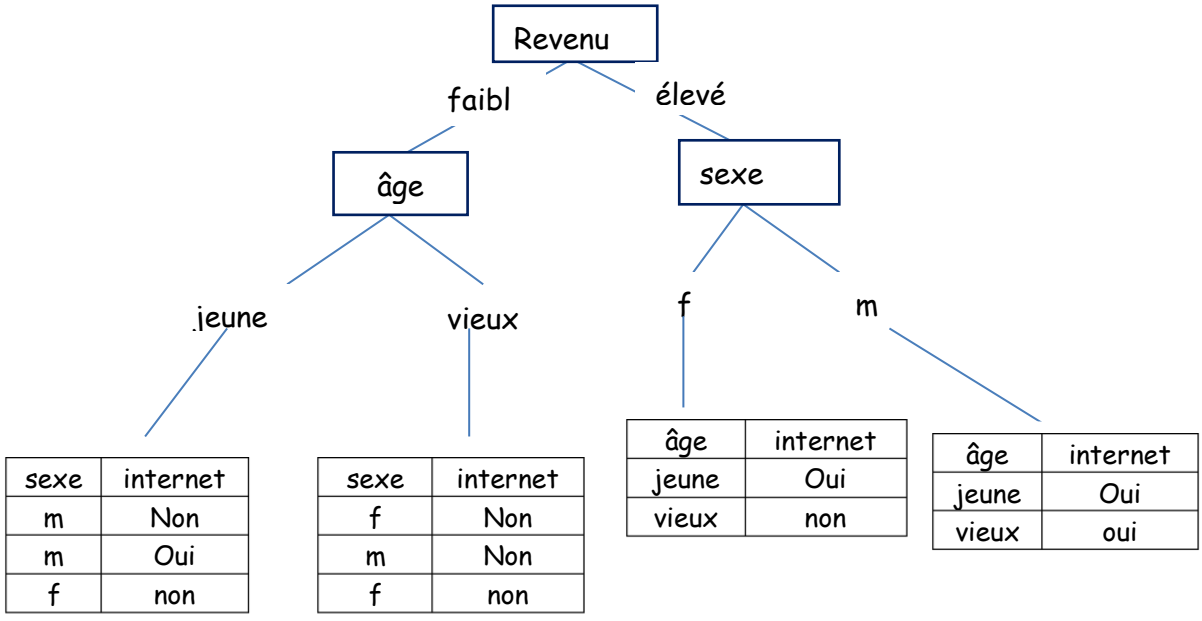
L'attribut revenu maximise le gain en information .En effet, les partitions obtenues pour les valeurs sont moins équilibrées que pour les autres attributs. Cet attribut réduit donc plus l'incertitude au sujet de la classe d'appartenance d'un élément. L'attribut revenu est donc choisi pour partitionner les éléments au premier niveau de l'arbre.



sexe	âge	Internet
m	jeune	non
m	jeune	oui
f	vieux	non
f	jeune	non
m	vieux	non
f	vieux	non

sexe	âge	Internet
m	jeune	oui
f	jeune	oui
f	vieux	non
m	vieux	oui

On répète le processus pour chacune des branches de l'arbre jusqu'à ce que la partie obtenue ne contienne que des éléments d'une des deux classes. Dans notre exemple, pour chacune des deux branches, le gain en information est le même pour les deux attributs qui restent. On peut alors choisir l'un ou l'autre. Pour fin d'illustration, prenons *âge* pour la branche *revenu = faible* et *sexe* pour *revenu=élevé*. Il est permis d'employer des attributs différents pour un niveau donné.



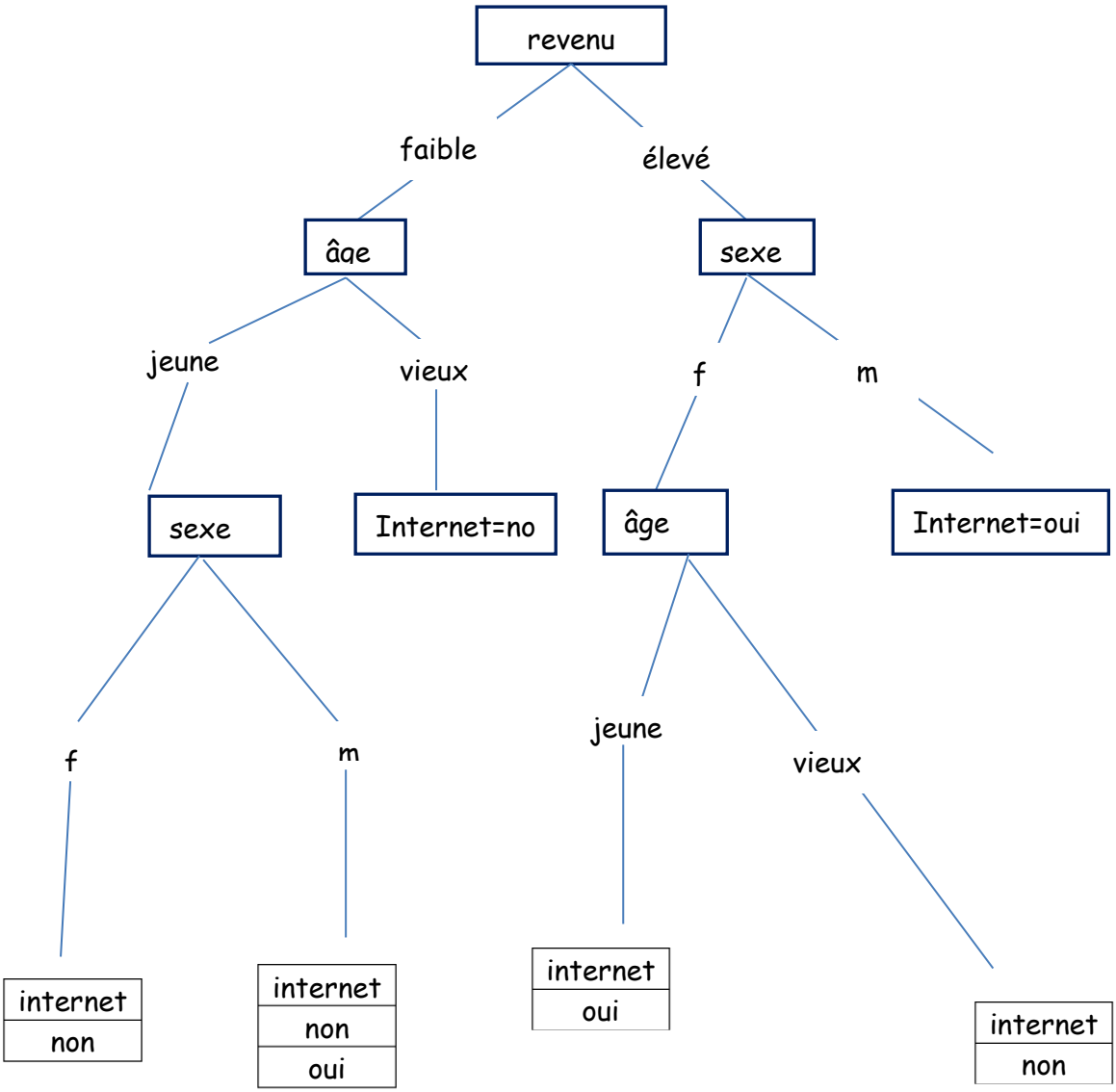
sexe	internet
m	Non
m	Oui
f	non

sexe	internet
f	Non
m	Non
f	non

âge	internet
jeune	Oui
vieux	non

âge	internet
jeune	Oui
vieux	oui

La division est terminée pour la deuxième feuille (*revenu= faible et âge=vieux*), car tous les éléments ont la même valeur *internet=non*. Il en est de même pour la feuille (*revenu=élevé et sexe=m*). Pour les deux autres, le découpage se poursuit produisant l'arbre suivant



Pour la feuille correspondant au chemin *revenu=faible* et *âge=jeune et sexe=m*, les deux éléments restants ne possèdent pas la même valeur pour la classe. Dans ce cas, on choisit normalement, la classe dont la fréquence est supérieure. Dans notre exemple, c'est l'égalité et on peut choisir arbitrairement une des deux classes.