

Classification par arbre de décision

Classification par arbre de décision

Arbre de décision

L'algorithme arbre de décision est l'un des algorithmes les plus polyvalents de l'apprentissage automatique, qui peut effectuer à la fois une analyse de classification et une analyse de régression.

Il est très puissant et fonctionne très bien avec des ensembles de données complexes.

Il fonctionne en divisant l'ensemble de données en une structure arborescente basée sur certaines règles et conditions, puis donne une prédiction basée sur ces conditions.

C'est un algorithme très simple à comprendre.

Typologie

Il existe deux principaux types d'arbre de décision en fouille de données :

- Les arbres de classification (Classification Tree) permettent de prédire à quelle classe la variable-cible appartient, dans ce cas la prédiction est une étiquette de classe,
- Les arbres de régression (Regression Tree) permettent de prédire une quantité réelle (par exemple, le prix d'une maison ou la durée de séjour d'un patient dans un hôpital), dans ce cas la prédiction est une valeur numérique.

Présentation général d'arbre de décision

“ On se donne un ensemble X de N exemples notés x_i dont les P attributs sont quantitatifs ou qualitatifs. Chaque exemple x est étiqueté, c'est-`a-dire qu'il lui est associée une « classe » ou un « attribut cible » que l'on note y .

Un arbre de décision est un classificateur représenté sous forme d'arbre tel que:

- Les nœuds de l'arbre testent les attributs.
- Il y a une branche pour chaque possibilité de l'attribut testé.
- Les feuilles spécifient les catégories(deux ou plus).

Présentation général d'arbre de décision

“ L’arbre de décision peut être ensuite exploité de différentes manières :

1. en y classant de nouvelles données ;
2. en faisant de l’estimation d’attribut ;
3. en extrayant un jeu de règles de classification concernant l’attribut cible
4. en interprétant la pertinence des attributs

Vocabulaire

Arbre, nœud, racine, feuille

- Un arbre est constitué de **nœuds** connectés entre eux par des **branches**.
- Une branche entre deux nœuds est orientée :
l'un des nœuds de la connexion est dit « **nœud parent** », et l'autre « **nœud enfant** ».
- Chaque nœud est connecté à un et **un seul nœud parent**, sauf le **nœud racine** qui n'a pas de parent.
- Chaque nœud peut être connecté à **0 ou n nœuds enfants**.
- Les deux caractéristiques précédentes font qu'**un arbre n'est pas un réseau (ou graphe)**.
- Un nœud qui n'a pas de parents est appelé « **nœud racine** » ou « **racine** ».
- Un nœud qui n'a pas de nœuds enfants est appelé « **nœud feuille** » ou « **feuille** ».

Vocabulaire

Variable cible et variables prédictives

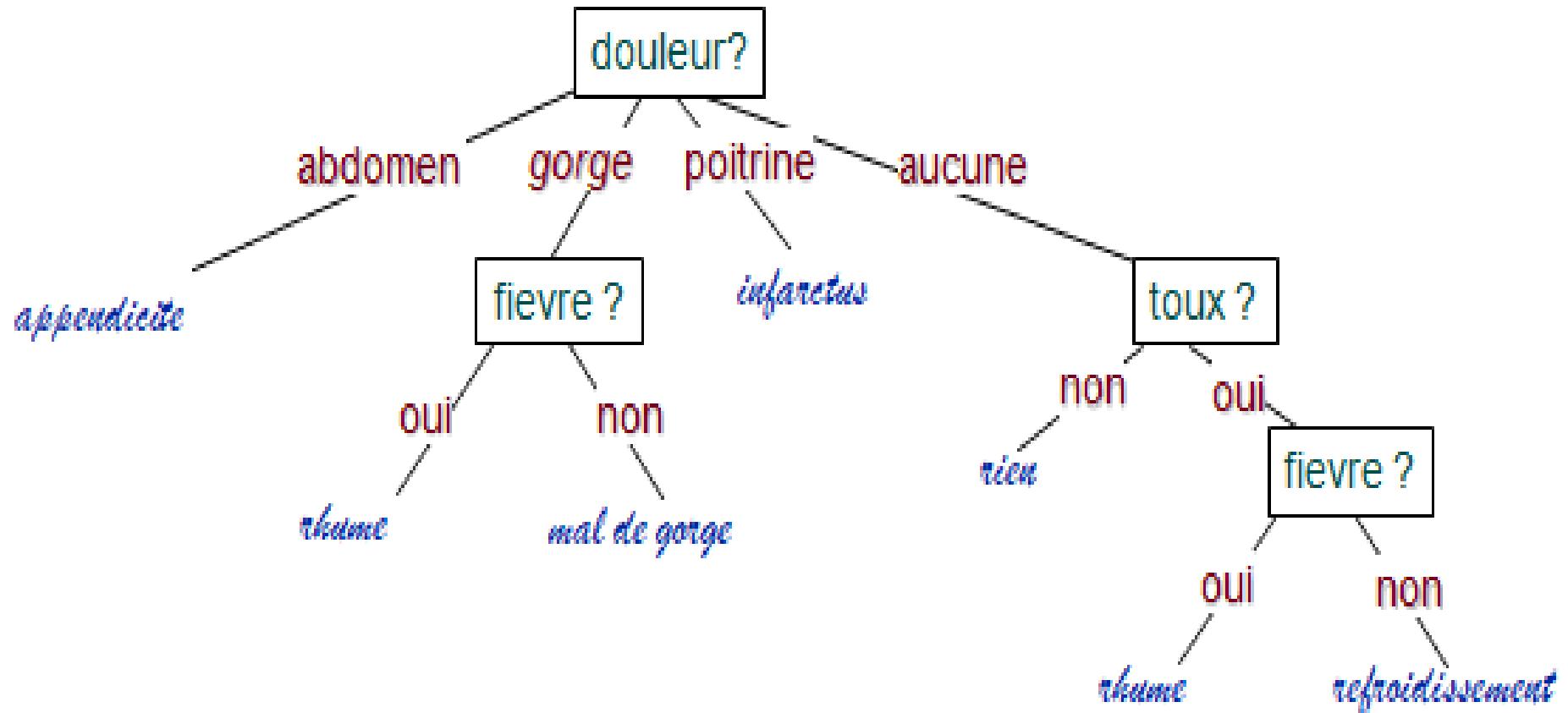
- “ Comme toutes les méthodes supervisées, un arbre de décision travaille sur une variable cible avec plusieurs variables prédictives.
 - Chaque **nœud non-feuille** correspond à une **variable prédictive**.
 - Chaque **nœud feuille** correspond à la **variable cible**.
 - Chaque **branche** correspond à une **valeur pour la variable prédictive** du nœud parent (ou un ensemble de valeurs).

Un exemple : Détection de la grippe

- “ Apparition soudaine de fièvre élevée
 - “ Le patient est fatigué
 - “ Rhinorrhée (nez qui coule)
 - “ Toux
 - “ Douleurs à la gorge
 - “
- Enrouement, douleurs dorsales, des membres et céphalées

□Grippe

Représentation sous forme d'arbre



	Toux	Fièvre	Poids	Douleur
Marie	non	oui	normal	gorge
Fred	non	oui	normal	abdomen
Julie	oui	oui	maigre	aucune
Elvis	oui	non	obese	poitrine

Autre exemple : la ballade du chien

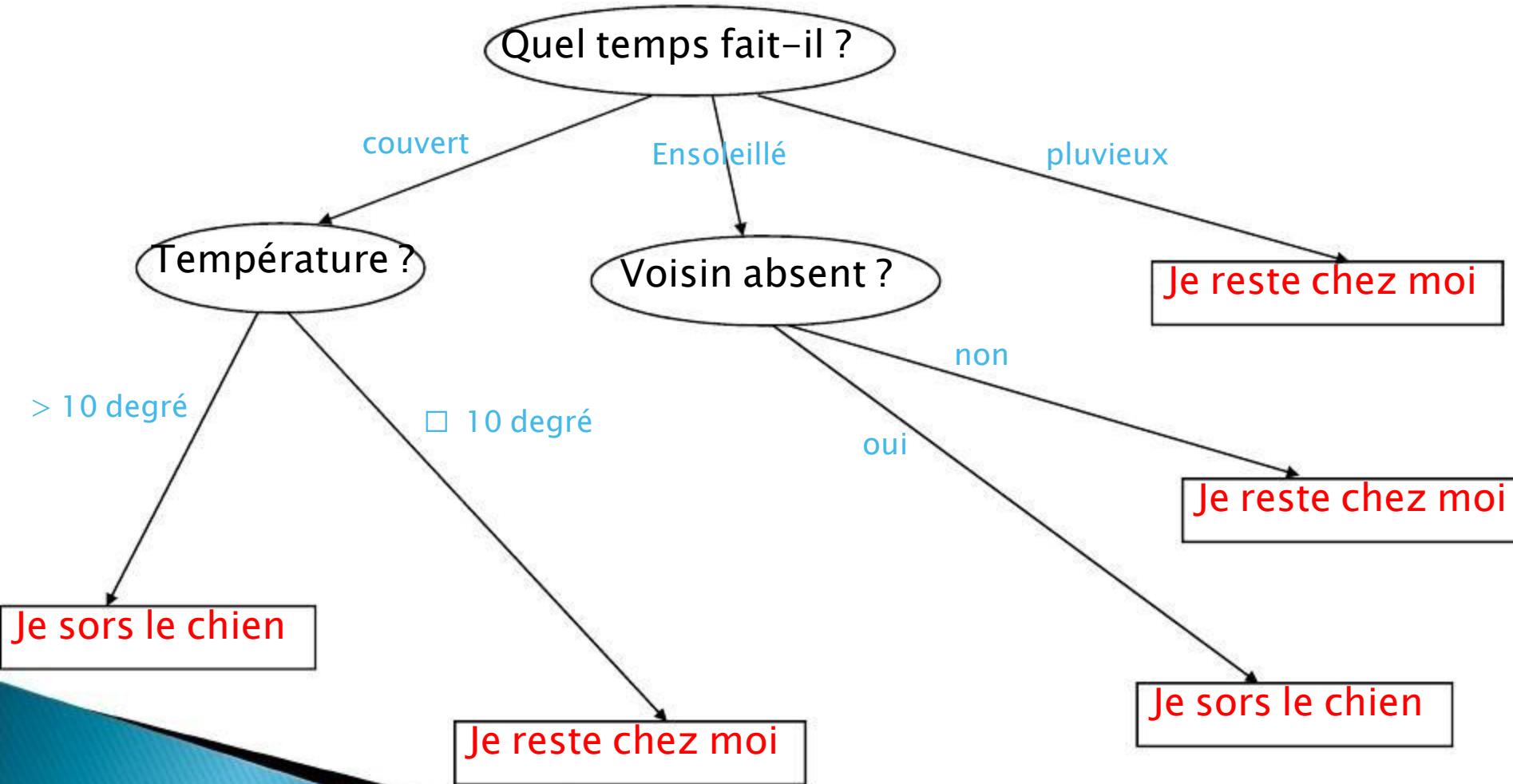
“ Attributs

- ❖ quel temps fait-il ? {pluvieux, ensoleillé, couvert}
- ❖ Température extérieure : attribut numérique
- ❖ Voisin parti avec son chat : attribut booléen

“ Décision à prendre

- ❖ Sortir ou non le chien

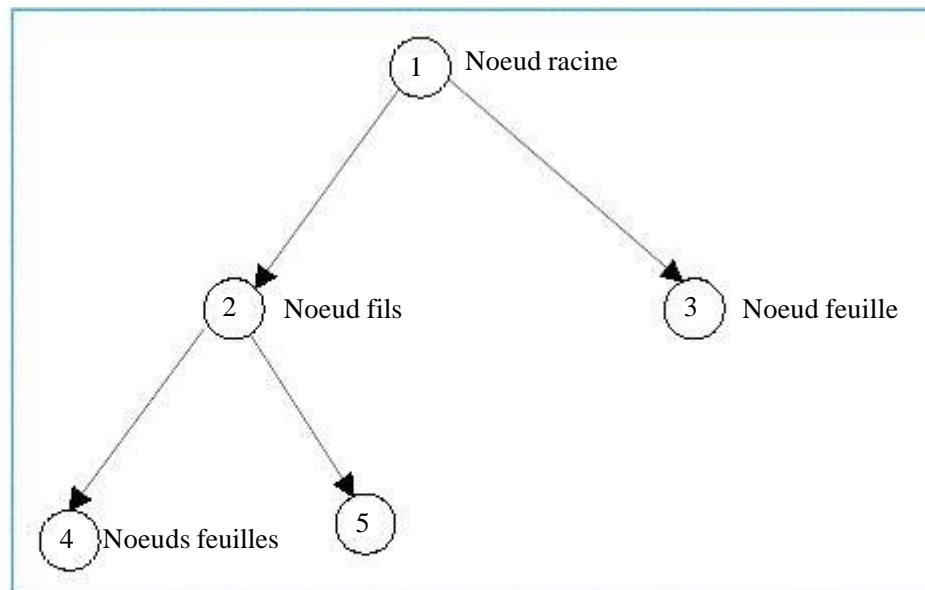
Arbre de décision



Arbre de décision : Structure

Un arbre de décision est composé :

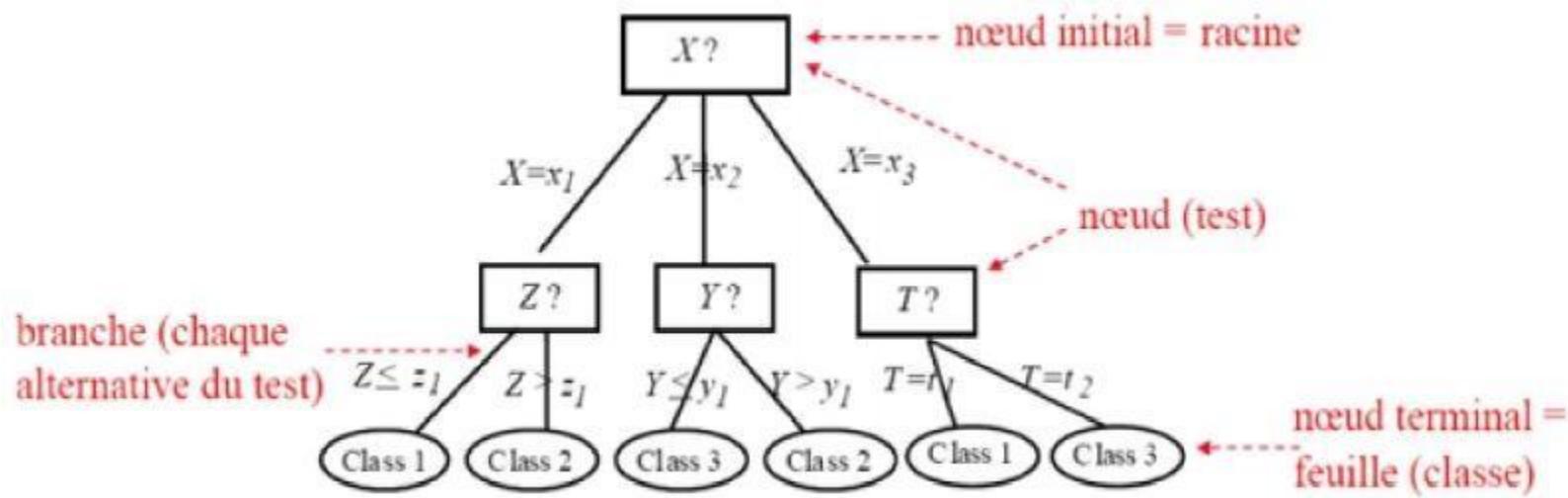
- ✓ D'un nœud racine par lequel entre les enregistrements,
- ✓ De questions,
- ✓ De réponses qui conditionnent la question suivante,
- ✓ De nœuds feuilles qui correspondent à un classement.



structure d'un arbre de décision

Présentation général d'arbre de décision

Structure



Règle de classification: aller de la racine à une feuille en effectuant les tests des nœuds.

Classe d'une feuille: classe majoritaire parmi les exemples d'apprentissage appartenant à cette feuille

Présentation général d'arbre de décision

Intérêts

- Les arbre de décision sont des classifieurs interprétable contrairement ou réseau de neurone par exemple .
- Ils fonctionnement facilement sur données qualitative
- Ils fonctionnement bien (tant que le nombre de caractéristique n'est pas trop grand)

Arbre de décision

“ Les avantages et les limites

Avantages	Limites
<ul style="list-style-type: none">▪ Simples et visuel pour la lecture et l'interprétation des résultats .▪ Modèle commutatif de classification sous forme de question.▪ Permet d'exploiter les décisions pessimistes et optimistes en fonction des probabilistes.▪ Convient aux modèles quantitatives et qualitatives.	<ul style="list-style-type: none">▪ Le difficulté réside dans le calcules de probabilités pertinents.▪ Devient très complexe lorsqu'on intégré des algorithme et que l'on augmentes le nombre de questions et d'hypothèse.

Procédure de construction (1)

Stratégie : Induction descendante

Recherche en meilleur d'abord sans retour arrière (gradient)
avec une fonction d'évaluation

Choix récursif d'un attribut de test jusqu'à critère d'arrêt

Fonctionnement :

On choisit le premier attribut à utiliser pour l'arbre : le plus informatif

Après ce choix, on se trouve face au problème initial sur des sous-ensembles d'exemples.

D'où un algorithme récursif.

Procédure de Construction (2)

- “ recherche à chaque niveau, l'attribut le plus discriminant
- “ Partition (nœud P)
 - si (tous les éléments de P sont dans la même classe) alors retour;
 - pour chaque attribut A faire
 - ☞ évaluer la qualité du partitionnement sur A;
 - utiliser le meilleur partitionnement pour diviser P en P₁, P₂, ...P_n
 - pour i = 1 à n faire Partition(P_i);

Procédure de Construction (3)

Questions pratiques lors de la construction d'arbres
Quand l'on construit un arbre de décision, les questions pratiques que l'on peut énumérer sont les suivantes :

- ✓ Comment choisir le bon attribut
- ✓ Quelle devrait être la profondeur de l'arbre ?
- ✓ Comment considérer les attributs continus ?
- ✓ Qu'est ce qu'un bon critère de division ?
- ✓ Que se passe t-il quand il manque des valeurs d'attributs ?
- ✓ Comment améliore t-on l'efficacité de l'arbre de décision

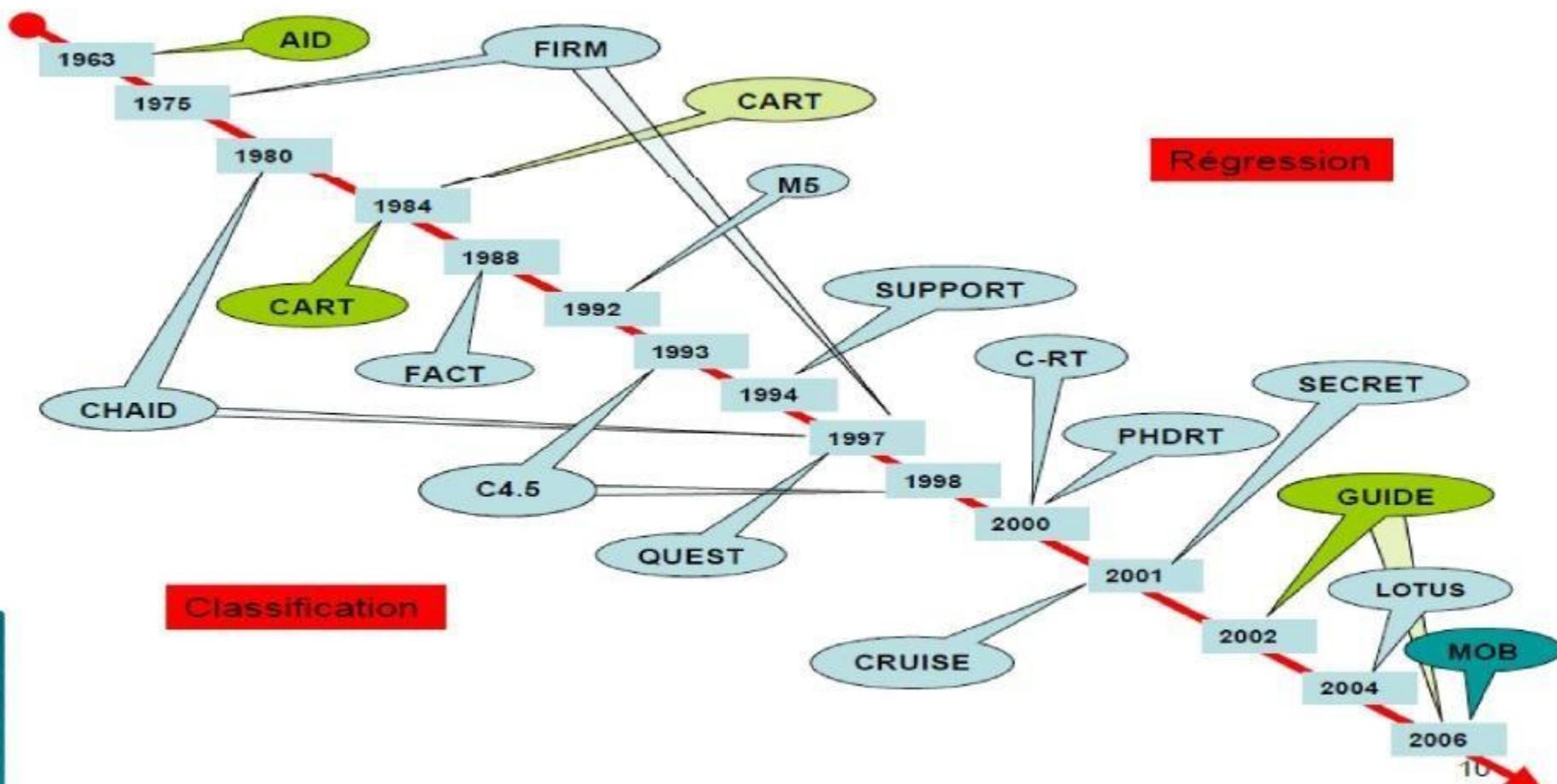
Construction de l'arbre de décision

? Comment choisir le meilleur attribut :

Il nous faut donc un indicateur (une mesure) qui permet d'évaluer objectivement la qualité d'une segmentation et ainsi de sélectionner le meilleur parmi les descripteurs candidats à la segmentation sur un sommet.

Construction de l'arbre de décision

- “ Plusieurs algorithmes proposés pour la construction d'un arbre de décision



Les algorithmes de construction d'un arbre de décision:

Plusieurs variantes: ID3, C4.5, CART, CHAID

Différence principale: mesure de sélection d'un attribut et critère de branchement (split)

- ✓ Gain d'Information (ID3, C4.5)
- ✓ Indice Gini (CART)
- ✓ Table de contingence statistique χ^2 (CHAID)

Les algorithmes de construction d'un arbre de décision:

- ID3
- C4.5
- CART

ID3

- Un algorithme mathématique pour construire l'arbre de décision.
- Inventé par J. Ross Quinlan en 1979.
- Utilise la théorie de l'information inventée par Shannon en 1948.
- Construit l'arbre de haut en bas, sans retour en arrière.
- Information Gain est utilisé pour sélectionner l'attribut le plus utile pour la classification.

Construction d'un arbre de décision

Algorithme ID3 : Principe

- Déterminer un attribut A à placer en racine de l'arbre
 - A = Meilleur Attribut (Exemples)
- Pour chaque valeur de A, créer une branche étiquetée avec cette valeur ➔ un nouveau nœud fils de la racine
- Classer les exemples dans les nœuds fils en considérant tous les attributs excepté celui qui vient d'être mis à la racine
- Si tous les exemples d'un nœud fils sont homogènes, affecter leur classe au nœud ➔ Une feuille de l'arbre
- sinon recommencer à partir de ce nœud

L'entropy

- Une formule pour calculer l'homogénéité d'un échantillon.
- Un échantillon complètement homogène a une entropie de 0.
- Un échantillon (avec des divisions égales) a une entropie de 1.
- Entropie (s) = $- p_+ \log_2 (p_+) - p_- \log_2 (p_-)$ pour un échantillon d'éléments négatifs et positifs.
- La formule pour l'entropie est:

$$Entropy(S) = \sum_{i=1}^c p_i \log_2 p_i$$

L'entropy

- ✓ L'entropie fournit une définition de descripteurs les plus significatifs, et c'est l'un des concepts majeurs de la méthode ID3 et beaucoup d'autres.
- ✓ Elle est utilisée dans la construction d'arbre de décision. Il a été formalisé par l'ingénieur en génie électrique Claude Shannon en 1948 .
- ✓ C'est une fonction mathématique qui, intuitivement, correspond à la quantité d'information contenue ou délivrée par une source d'information.

L'entropy

- “ Dans le cas d'une classe binaire avec deux proportions p_+ et p_- l'entropie est donnée par :

$$E(X) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

- “ Remarques :
 - $0 \leq E(X) \leq 1$
 - Si $p_+ = 0$ ou $p_- = 0$ alors $E(X) = 0$
 - Si $p_+ = p_- = 0.5$ alors $E(X) = 1$ (entropie max)

Gain d'information

- f Pour sélectionner l'attribut avec le plus grand gain d'information
- Soient P et N deux classes et S un ensemble d'instances avec p éléments de P et n éléments de N
- f L'information nécessaire pour déterminer si une instance prise au hasard fait partie de P ou N est(entropie)

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

Gain d'information

- Il permet d'avoir l'attribut qui crée les branches les plus homogènes. Le gain d'information est basé sur la diminution de l'entropie après la division d'un ensemble de données par rapport à un attribut.
 - L'entropie de l'ensemble de données est d'abord calculée.
 - L'ensemble de données est ensuite divisé sur les différents attributs.
 - L'entropie pour chaque branche est calculée. Ensuite, il est ajouté proportionnellement, pour obtenir l'entropie totale pour la division.
 - L'entropie résultante est soustraite de l'entropie avant la division.
 - Le résultat est le gain d'information ou la diminution de l'entropie.
 - L'attribut qui donne le plus grand IG est choisi pour le nœud de décision.

Gain d'information

Soient les ensembles $\{S_1, S_2, \dots, S_v\}$ formant une partition de l'ensemble S , en utilisant l'attribut A

Toute partition S_i contient p_i instances de P et n_i instances de N

Entropie, ou l'information nécessaire pour classifier les instances dans les sous-arbres S_i est:

$$E(A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

Le gain d'information par rapport au branchement sur A est

$$Gain(A) = I(p, n) - E(A)$$

$$Gain(S, A) = Entropie(S) - \sum_{v \in values(A)} |S_v| / |S| Entropie(S_v)$$

Choisir l'attribut qui maximise le gain

Exemple: Météo et match de foot

Day	Outlook	Temp.	Humidit	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Attribut but

2 classes: yes et no

Température est un nominal

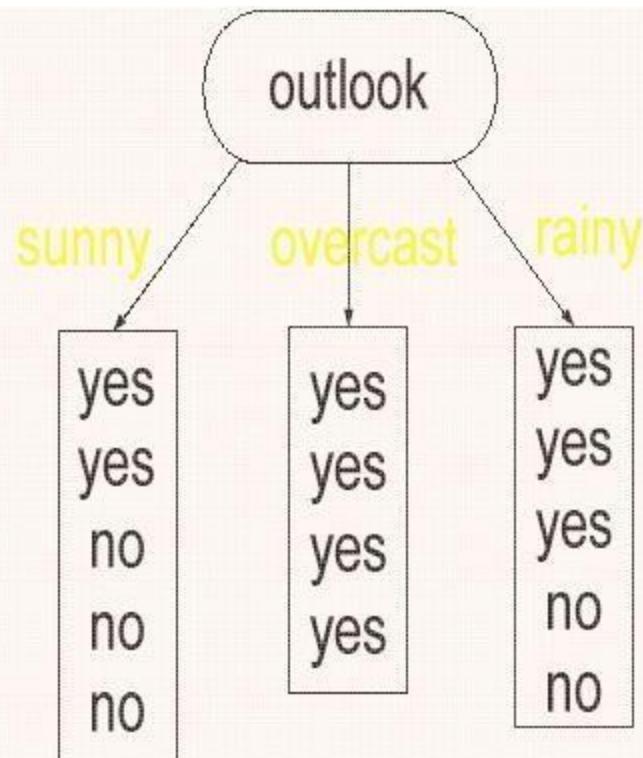
On veut pouvoir décider/prédire si un match de foot va avoir lieu ou pas

Construction: Exemple

Calculer:

- $P(\text{play} = \text{"yes"})$
- $P(\text{play} = \text{"no"})$
- $P(\text{play} = \text{"no"} \mid \text{overcast} = \text{"sunny"})$
- $P(\text{play} = \text{"yes"} \mid \text{overcast} = \text{"sunny"})$
-

Exemple



Exemple : Suite

Calculer l'entropie pour l'attribut *outlook* :

outlook	p_i	n_i	$I(p_i, n_i)$
sunny	2	3	0,971
overcast	4	0	0
rain	3	2	0,971

On a

$$E(outlook) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

Alors $Gain(outlook) = I(9,5) - E(outlook) = 0.246$

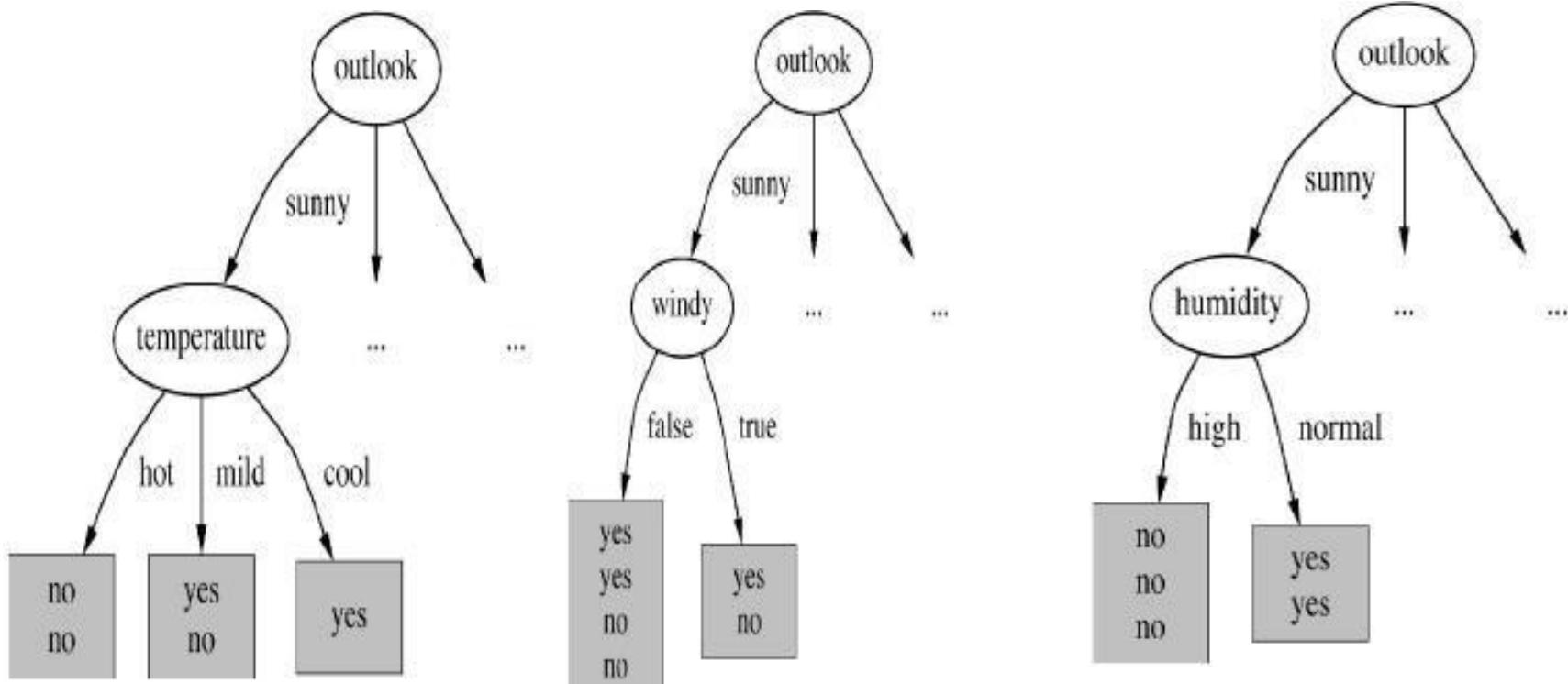
De manière similaire

$$Gain(temperature) = 0.029$$

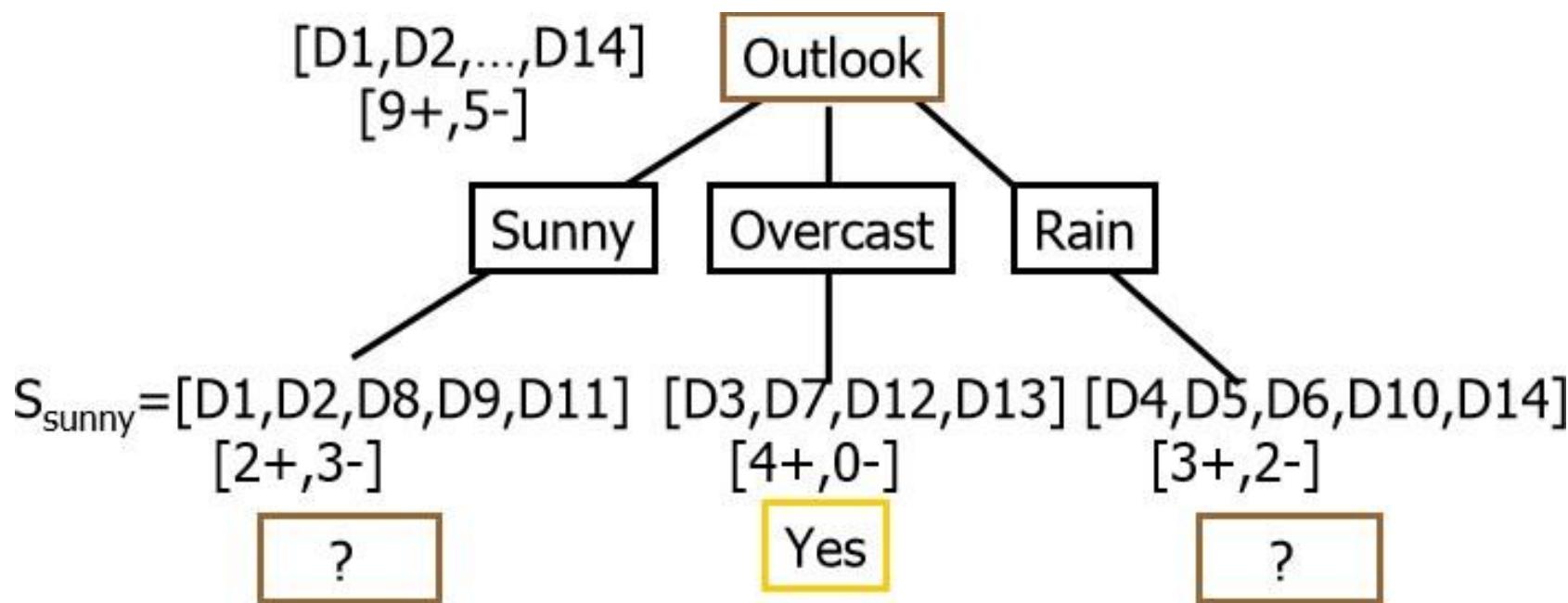
$$Gain(humidity) = 0.151$$

$$Gain(windy) = 0.048$$

Choix du deuxième attribut



Choix du deuxième attribut



$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = 0.970 - (3/5)0.0 - 2/5(0.0) = 0.970$$

$$\text{Gain}(S_{\text{sunny}}, \text{Temp.}) = 0.970 - (2/5)0.0 - 2/5(1.0) - (1/5)0.0 = 0.570$$

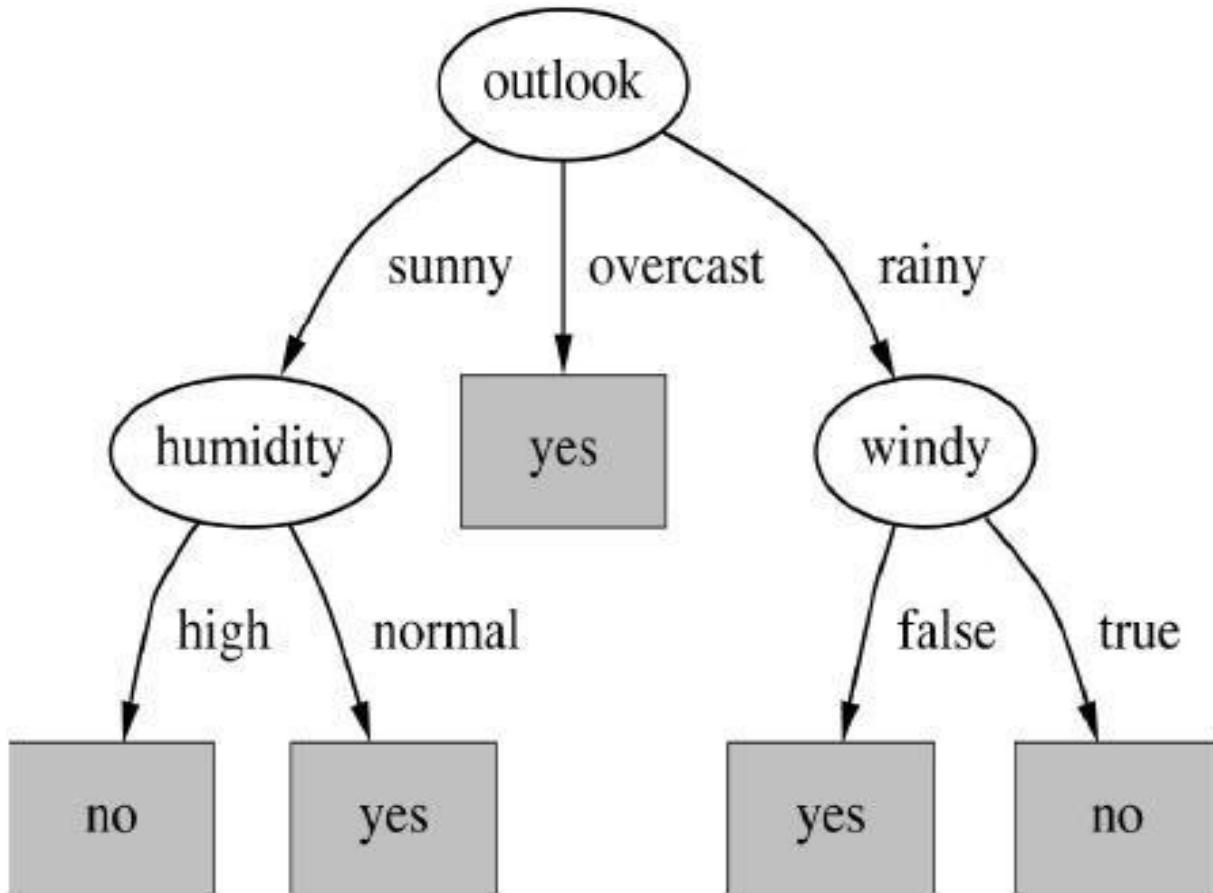
$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) = 0.970 - (2/5)1.0 - 3/5(0.918) = 0.019$$

Étape suivante

- “ Sélection d'un deuxième attribut
- “ On peut examiner:
 - ✓ Température, Humidity ou Windy pour Outlook = “sunny”
 - ✓ Gain(“Température”) = 0.571 bits
 - ✓ Gain(“Humidity”) = 0.971 bits
 - ✓ Gain(“Windy”) = 0.019 bits
- “ Et on continue...

← Humidity est
choisi

Arbre de décision final



Interprétation de l'arbre

pertinence des attributs :

➤ L'arbre de décision construite nous donne des informations sur la **pertinence des attributs** vis-à-vis de la classe :

- l'attribut « Température » n'étant pas utilisé dans l'arbre ; ceci indique que cet attribut n'est pas **pertinent** pour déterminer la classe (la décision).
- si l'attribut « outlook » vaut « sunny », l'attribut « windy » n'est pas **pertinent** ;
- si l'attribut «outlook » vaut « rain », c'est l'attribut « Humidity » qui ne l'est pas.

Exercices d'application

Soient les exemples suivants :

1. Calculer l'entropie de l'ensemble d'exemples par rapport à la valeur de la classe.

Instance	Classe	a1	a2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

2. Quel est le gain de l'attribut a2.

Corrigé

1. En appliquant :

$$I(n, p) = \left(\frac{p}{p+n}\right) \log_2\left(\frac{p+n}{p}\right) + \left(\frac{n}{p+n}\right) \log_2\left(\frac{p+n}{n}\right)$$

nous avons 3 exemples positifs et trois négatifs donc $I(n, p) = 1$

2. En appliquant :

$$E(T) = \sum \left(\frac{p_i + n_i}{p+n}\right) I(p_i, n_i)$$

où T est un test sur l'attribut a2 on aura :

$$E(T) = \frac{4}{6} * I(2, 2) + \frac{2}{6} * I(1, 1) = 1$$

Par conséquent, le gain est égal à 0.

Exercice 1

Calculer le gain d'information de chaque attribut des données ci-dessous.

Exemple	Trans?	Pass?	Anim?	Classe
courir	bas	bas	haut	MoM
marcher	haut	bas	haut	MoM
fondre	bas	bas	bas	CoS
cuire	haut	haut	haut	CoS

Exercice—solution

$$Gain(S, A) = Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Anim?) = E(S) - \frac{|S_{haut}|}{|S|} E(S_{haut}) - \frac{|S_{bas}|}{|S|} E(S_{bas})$$

$$\begin{aligned} &= E(S) - \frac{|S_{haut}|}{|S|} - \left(\left(\frac{2}{3} \log \frac{2}{3} \right) + \left(\frac{1}{3} \log \frac{1}{3} \right) \right) - \frac{|S_{bas}|}{|S|} - \left((1 \log 1) + (1 \log 1) \right) \\ &= E(S) - \frac{|S_{haut}|}{|S|} (.39 + .53) - \frac{|S_{bas}|}{|S|} (0) \end{aligned}$$

$$Gain(S, Anim?) = 1 - \frac{3}{4} (.39 + .53) - \frac{1}{4} (0) = 1 - .69 = .31$$

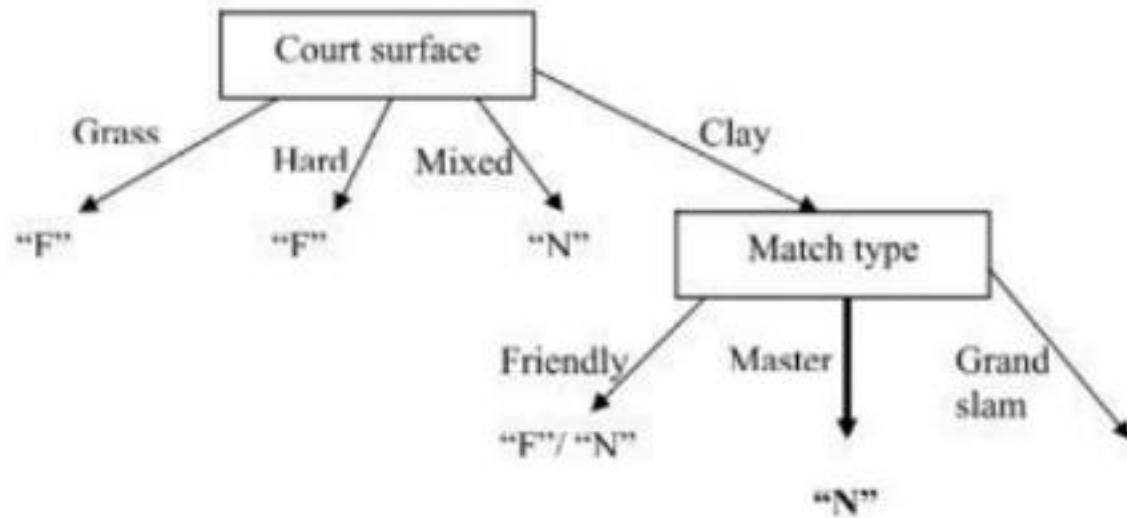
$$Gain(S, Pass?) = 1 - \frac{1}{4} (0) - \frac{3}{4} (.39 + .53) = 1 - .69 = .31$$

$$Gain(S, Trans?) = 1 - \frac{2}{4} (.5 + .5) - \frac{2}{4} (.5 + .5) = 1 - 1 = 0$$

Exercice 2

Elaborer l'arbre de décision associée à cette base de connaissance (ID3)

Time	Match type	Court surface	Best Effort	Outcome
Morning	Master	Grass	1	F
Afternoon	Grand slam	Clay	1	F
Night	Friendly	Hard	0	F
Afternoon	Friendly	Mixed	0	N
Afternoon	Master	Clay	1	N
Afternoon	Grand slam	Grass	1	F
Afternoon	Grand slam	Hard	1	F
Afternoon	Grand slam	Hard	1	F
Morning	Master	Grass	1	F
Afternoon	Grand slam	Clay	1	N
Night	Friendly	Hard	0	F
Night	Master	Mixed	1	N
Afternoon	Master	Clay	1	N
Afternoon	Master	Grass	1	F
Afternoon	Grand slam	Hard	1	F
Afternoon	Grand slam	Clay	1	F



Time	Match type	Court surface	Best Effort	Outcome
Afternoon	Grand slam	Clay	1	F
Afternoon	Grand slam	Clay	1	N
Afternoon	Grand slam	Clay	1	F

wonder

Avantages d'ID3

- Des règles de prédiction compréhensibles sont créées à partir des données d'apprentissage.
- Construit l'arbre le plus rapide.
- Construit un arbre court.
- A seulement besoin de tester suffisamment d'attributs jusqu'à ce que toutes les données sont classées.
- La recherche de nœuds feuilles permet d'élaguer les données de test, ce qui réduit le nombre de tests.
- L'ensemble de données entier est recherché pour créer le noeud de l'arbre.

Inconvénients d'ID3

- Les données peuvent être sur-ajustées ou sur-classifiées, si un petit échantillon est testé.
- Un seul attribut à la fois est testé pour prendre une décision.
- La classification des données continues peut être coûteuse en calcul, car de nombreux arbres doivent être générés pour voir où casser le continuum.

C 4.5 : amélioration de ID3

- L'algorithme C4.5 a également été inventé par Ross Quinlan, il en a fait un livre édité en 1993.
- C'est un algorithme qui est basé sur ID3 mais qui a quelques éléments en plus :
 - la possibilité de manipuler des valeurs continues (en les “discrétilisant” lors de la mise en arbre).
 - adaptation de la fonction de gain qui n'a plus tendance à aller vers l'attribut avec le plus de valeurs possibles;
 - la possibilité de gérer des attributs avec des valeurs manquantes;
 - la possibilité de post-élaguer son arbre pour éviter l’“overfitting”;

Construction d'un arbre de décision en présence d'attributs numériques

- “ ID3 ne prend pas en compte les attributs de type numérique (que les attributs qualitatifs)
- “ Le successeur C4.5 de ID3 prend en compte les attributs de type numérique, des attributs dont l'arité est élevée (voire infinie).
- “ La construction d'un arbre de décision par C4.5 est identique dans son principe à la construction par ID3
- “ Dans le cas de C4.5, un noeud de l'arbre de décision peut contenir un test du fait que la valeur d'un attribut numérique est inférieure à un certain seuil :
 - cela correspond donc à un nouveau pseudo-attribut binaire.

Comment traiter les attributs numériques?

Les attributs numériques sont transformés en ordinaux / nominaux. Ce processus est appelé discréétisation

“

Les valeurs des attributs sont divisées en intervalles :

- “ Les valeurs des attributs sont triées
- “ Des séparations sont placées pour créer des intervalles / classes pures (On détermine les valeurs des attributs qui impliquent un changement de classes

Ce processus est très sensible au bruit

“

Le nombre de classes doit être contrôlé :

- “ Solution: On spécifie un nombre minimum d'éléments par intervalle
- “ On combine les intervalles qui définissent la même classe

Test d'un attribut numérique

- Jeu de données « jouer au tennis ? » avec des attributs quantitatifs et nominaux.

Jour	Ciel	Température	Humidité	Vent	Décision
1	Ensoleillé	27.5	85	Faible	NON
2	Ensoleillé	25	90	Fort	NON
3	Couvert	26.5	86	Faible	OUI
4	Pluie	20	96	Faible	OUI
5	Pluie	19	80	Faible	OUI
6	Pluie	17.5	70	Fort	NON
7	Couvert	17	65	Fort	OUI
8	Ensoleillé	21	95	Faible	NON
9	Ensoleillé	19.5	70	Faible	OUI
10	Pluie	22.5	80	Faible	OUI
11	Ensoleillé	22.5	70	Fort	OUI
12	Couvert	21	90	Fort	OUI
13	Couvert	25.5	75	Faible	OUI
14	Pluie	20.5	91	Fort	NON

Test d'un attribut numérique

“Considérons les exemples dont l'attribut «Ciel» vaut «Ensoleillé», soit l'ensemble $X_{\text{Ciel}} = \text{Ensoleillé}$ d'exemples ayant un seul attribut numérique comme suit

Jour	Température «Jouera au tennis?»
1	27.5Non
2	25Non
8	21Non
9	19.5Oui
11	22.5Oui

Test d'un attribut numérique

Exemple : attribut température.

- 1) Ordonner toutes les valeurs dans l'ensemble d'apprentissage
- 2) Considérer les points de coupure où il y a un changement de classes
- 3) Choisir les points de coupure qui maximisent le gain en information

Température	Jour	« Jouer au tennis? »
19.5	9	Oui
21	8	Non
22.5	11	Oui
25	2	Non
27.5	1	Non

Test d'un attribut numérique

- “ Pour déterminer le seuil s pour partitionner cet ensemble d'exemples. C4.5 utilise les règles suivantes :
 - ne pas séparer deux exemples successifs ayant la même classe ; donc, on ne peut couper qu'entre les exemples x_9 et x_8 , x_8 et x_{11} , x_{11} et x_2 ;
 - si on coupe entre deux valeurs v et w ($v < w$) de l'attribut, le seuil s est fixé à v ou w (on aurait pu aussi utiliser $(v+w)/2$) ;
 - choisir s de telle manière que le gain d'information soit maximal.
- “ Remarque :
 - une fois le seuil s fixé et le noeud créé, chaque sous-arbre pourra à nouveau tester la valeur de cet attribut ;
 - contrairement au cas des attributs qualitatifs qui produisent des nœuds ayant autant de branches que l'attribut prend de valeurs différentes, l'ensemble des valeurs prises par un attribut numérique est coupé en deux : chaque partie peut donc encore être raffinée jusqu'à ne contenir que des exemples ayant même valeur cible.

Test d'un attribut numérique

“ Application : l'entropie de l'ensemble d'exemples est :

$$H(\mathcal{X}) = -\left(\frac{2}{5} \ln_2 \frac{2}{5} + \frac{3}{5} \ln_2 \frac{3}{5}\right) \approx 0.971$$

“ Pour $s = 21$, le gain d'information est :

$$\text{Gain}(\mathcal{X}, \text{Température}, s = 21) = H(\mathcal{X}) - \left(\frac{1}{5}H(\mathcal{X}_{\text{Température} < 21}) + \frac{4}{5}H(\mathcal{X}_{\text{Température} \geq 21})\right)$$

“ Avec

$$H(\mathcal{X}_{\text{Température} < 21}) = -(1 \ln_2 1 + 0 \ln_2 0) = 0$$

“ Et

$$H(\mathcal{X}_{\text{Température} \geq 21}) = -\left(\frac{1}{4} \ln_2 \frac{1}{4} + \frac{3}{4} \ln_2 \frac{3}{4}\right) \approx 0.608$$

“ Soit

$$\text{Gain}(\mathcal{X}, s = 21) \approx 0.971 - \left(\frac{1}{5} \times 0 + \frac{4}{5} \times 0.608\right) \approx 0.485$$

Test d'un attribut numérique

“ De la même manière, en fonction du seuil, le gain d'information est alors :

seuil	Gain(\mathcal{X} , Température, s)
$s = 21$	0.485
$s = 22.5$	0.02
$s = 25$	0.42

- “ C4.5 effectue ce traitement pour chaque attribut quantitatif et détermine donc pour chacun un seuil produisant un gain d'information maximal.
- “ Le gain d'information associé à chacun des attributs quantitatifs est celui pour lequel le seuil entraîne un maximum.
- “ Finalement, l'attribut choisi (parmi les quantitatifs et les nominaux pour lesquels le principe est identique ID3) est celui qui produit un gain d'information maximal.

Problème : Prédire si une personne achètera un produit en fonction de son âge.

Données d'entraînement :



Âge Achat (Oui/Non)

22 Oui

25 Oui

30 Non

35 Oui

40 Non

50 Non



-
1. Déterminer les candidats possibles (en utilisant $(V+W)/2$)
 2. Calculer le gain du candidat 32.5

Étapes suivies par C4.5 pour gérer les valeurs continues :

- 1. Trier les valeurs continues :**
 - L'algorithme trie les âges : 22, 25, 30, 35, 40, 50.
- 2. Définir des points de séparation possibles :**
 - C4.5 examine les points **entre chaque paire de valeurs** (moyennes des valeurs adjacentes) :
 - Seuils candidats= $\{(22+25)/2, (25+30)/2, (30+35)/2, (35+40)/2, (40+50)/2\}$
 - Seuils candidats} = $\{(22+25)/2, (25+30)/2, (30+35)/2, (35+40)/2, (40+50)/2\}$
 - |Seuils candidats= {23.5, 27.5, 32.5, 37.5, 45.0}

Calcul du gain pour le seuil $S = 32.5$:

1. **Diviser les données en deux groupes :**

- Groupe 1 ($\hat{A}ge \leq 32.5$)
- Groupe 2 ($\hat{A}ge > 32.5$)

2. **Calculer l'entropie initiale (avant division) :** Total : 6 exemples

- $P(\text{Oui}) = 3/6 = 0.5$
- $P(\text{Non}) = 3/6 = 0.5$

$$\text{Entropie initiale} = -[0.5 \log_2(0.5) + 0.5 \log_2(0.5)] = 1$$

3. Calculer l'entropie pour chaque groupe après division :

- Groupe 1 ($\hat{A}ge \leq 32.5$) :

3 exemples (2 Oui, 1 Non)

- $P(\text{Oui}) = 2/3, P(\text{Non}) = 1/3$

$$\text{Entropie}(\text{Groupe 1}) = -\left[\frac{2}{3} \log_2\left(\frac{2}{3}\right) + \frac{1}{3} \log_2\left(\frac{1}{3}\right)\right] = 0.918$$

- Groupe 2 ($\hat{A}ge > 32.5$) :

3 exemples (1 Oui, 2 Non)

- $P(\text{Oui}) = 1/3, P(\text{Non}) = 2/3$

$$\text{Entropie}(\text{Groupe 2}) = -\left(\frac{1}{3} \log_2\left(\frac{1}{3}\right) + \frac{2}{3} \log_2\left(\frac{2}{3}\right)\right) = 0.918$$

Calculer l'entropie pondérée après division :

- Taille du Groupe 1 : 3
- Taille du Groupe 2 : 3

$$\text{Entropie après division} = \frac{3}{6} \cdot 0.918 + \frac{3}{6} \cdot 0.918 = 0.918$$

Calculer le gain d'information :

$$\text{Gain d'information} = \text{Entropie initiale} - \text{Entropie après division}$$

$$\text{Gain} = 1 - 0.918 = 0.082$$

Les problèmes liés au calcul de gain

Le gain d'information présente un inconvenient. Il a tendance à préférer les attributs qui ont beaucoup de valeurs différentes, et qui partagent les données en nombreux petits sous-ensembles purs

Deux solutions :

Rendre tous les attributs binaires

Mais perte d'intelligibilité des résultats

Introduire un facteur de normalisation dans le calcul

$$Gain_norm(S, A) = \frac{Gain(S, A)}{\sum_{i=1}^{nb \text{ valeurs de } A} \frac{|S_i|}{|S|} \cdot \log \frac{|S_i|}{|S|}}$$

On utilise le `split_info` pour avoir le `gain_ratio`

Application sur le jeu de données "Jouer au tennis"

On va étudier le cas où la racine de l'arbre Ciel est égale à "Ensoleillé"

- on commence tout d'abord par l'attribut "Température" :

$$\text{Splintinfo}(2,2,1) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{2}{5} \log_2 \frac{2}{5} - \frac{1}{5} \log_2 \frac{1}{5} = 0.464$$

Donc le Rapport de gain est :

$$\text{Rapport de gain}(X, \text{Température}) = \frac{0.737}{0.464} = 1.58$$

- Pour l'attribut "Humidité" :

$$\text{Splintinfo}(3,2) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.380$$

Donc le Rapport de gain est :

$$\text{Rapport de gain}(X, \text{Humidité}) = \frac{0.737}{0.380} = 1.93$$

Day	Outlook	Temp.	Humidit	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Le maximum entre les deux rapports de gain est celui de l'attribut "Humidité" pour le cas où Ciel est égale à la valeur "Ensoleillé". Ce qu'est bien trouvé avec l'algorithme ID3.

Valeurs d'attributs manquantes

Une valeur manquante peut avoir plusieurs significations. Entre autres possibilités:

- ✓ le champ n'était pas applicable,
- ✓ l'événement n'a pas eu lieu
- ✓ les données n'étaient pas disponibles.
- ✓ Il se peut que la personne qui a saisi les données ne connaît pas la valeur exacte ou ne s'est pas inquiété de l'absence de remplissage d'un champ
- ✓

Valeurs d'attributs manquantes

Pour les valeurs manquantes des attributs , C4.5 a un mécanisme efficace qui distingue deux cas :

- Attributs non valués dans l'ensemble d'apprentissage .
- Attributs de la donnée à classer non valués .

Attributs non valués dans l'ensemble d'apprentissage

- “ Plusieurs solutions peuvent être envisagées, les plus générales étant :
- on laisse de côté les exemples ayant des valeurs manquantes ; **ennuyeux car le nombre d'exemples diminue** ;
 - le fait que la valeur de l'attribut soit manquante est une information en soit : **on ajoute alors une valeur possible à l'ensemble des valeurs de cet attribut qui indique que la valeur est inconnue** ;
 - la valeur la plus courante pour l'attribut en question parmi les exemples classées dans ce noeud est affectée à la place de la valeur manquante ;
 - les différentes valeurs observées de cet attribut parmi les exemples couverts par le même noeud sont affectées avec des poids différents en fonction de la proportion d'exemples de l'ensemble d'apprentissage couverts par ce noeud pour les différentes valeurs de cet attribut.

Attributs non valués dans l'ensemble d'apprentissage

“ Pour calculer le gain d'information, on ne tient compte que des exemples dont l'attribut est valué. Soit \mathcal{X} l'ensemble d'exemples couverts par le nœud courant (dont on est en train de déterminer l'attribut à tester) et les exemples dont l'attribut est valué. On redéfinit :

$$\mathcal{X}_{\text{sans } ?} \subset \mathcal{X}$$

$$H(\mathcal{X}) = H(\mathcal{X}_{\text{sans } ?})$$

$$\text{Gain } (\mathcal{X}, a) = (H(\mathcal{X}) - \sum_{v \in \text{valeurs}(a)} \frac{|\mathcal{X}_{\text{sans } ?, a=v}|}{|\mathcal{X}_{\text{sans } ?}|} H(\mathcal{X}_{\text{sans } ?, a=v})) \frac{|\mathcal{X}_{\text{sans } ?}|}{|\mathcal{X}|}$$

Attributs non valués dans l'ensemble d'apprentissage

- “ Supposons que l'exemple x_{12} ait ? à la place de Couvert comme valeur de son attribut « Ciel ». Déterminons le test à placer en racine de l'arbre de décision.
- “ L'entropie de l'ensemble d'apprentissage X est maintenant :

$$H(\mathcal{X}) = -\frac{8}{13} \ln_2 \frac{8}{13}$$

Gain (\mathcal{X} , Ciel) \approx

Day	Outlook	Temp.	Humidit	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$\begin{aligned}& \frac{3}{13} \left(-\frac{3}{5} \ln_2 \frac{3}{5} - \frac{2}{3} \ln_2 \frac{2}{3} \right) + \\& \frac{5}{13} \left(-\frac{3}{5} \ln_2 \frac{3}{5} - \frac{2}{5} \ln_2 \frac{2}{5} \right) \\\approx & 0.199\end{aligned}$$

Attributs non valués dans l'ensemble d'apprentissage

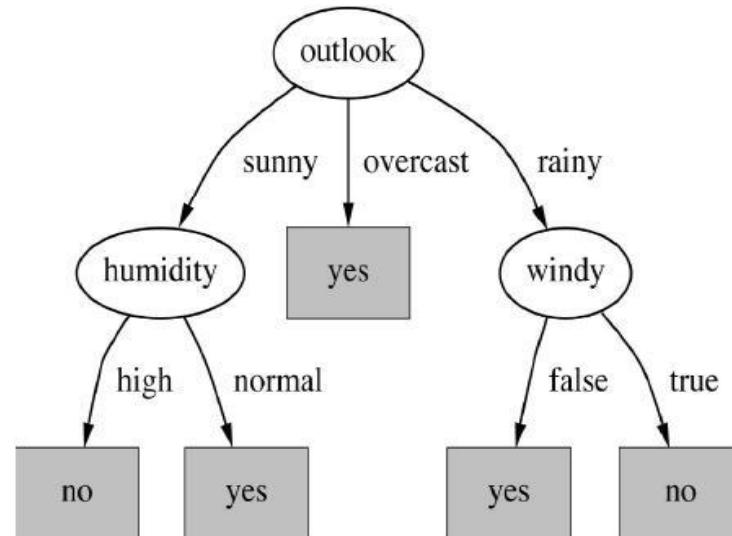
- “ Demeurant l'attribut fournissant un gain maximal, « Ciel » est placé à la racine de l'arbre.
- “ L'exemple 12 est affecté avec les poids $5/13$, $3/13$ et $5/13$ à chacune des branches, respectivement « Ensoleillé », « Couvert » et «Pluie» ; les autres exemples sont affectés à leur branche respective avec un poids 1 pour chacun.

Classification d'une donnée ayant des attributs non valués

- “ On se place ici non plus dans la phase de construction de l’arbre de décision, mais lors de son utilisation pour prédire la classe d’une donnée.
- “ Lors de sa descente dans l’arbre, si un nœud teste un attribut dont la valeur est inconnue, C4.5 estime la probabilité pour la donnée de suivre chacune des branches en fonction de la répartition des exemples du jeu d’apprentissage couverts par ce noeud. Cela détermine une fraction de donnée qui poursuit sa descente selon chacune des branches.
- “ Arrivé aux feuilles, C4.5 détermine la classe la plus probable à partir de ces probabilités estimées.
- “ Pour chaque classe, il fait la somme des poids ; la classe prédictive est celle dont le poids est maximal.

Classification d'une donnée ayant des attributs non valus

- “ Dans l’arbre de décision obtenu sur « jouer au tennis ? » avec des attributs nominaux, classons la donnée (Ciel =?, Température =Tiède, Humidité =?, Vent = Faible).
- “ Le noeud racine testant l’attribut « Ciel », sa valeur étant inconnue dans cette donnée à classer, on calcule la proportion d’exemples correspondant à chaque valeur :
 - 5 « Ensoleillé » ;
 - 4 « Couvert » ;
 - 5 « Pluie ».



(Ciel =?, Température =Tiède, Humidité =?, Vent = faux). ?classe

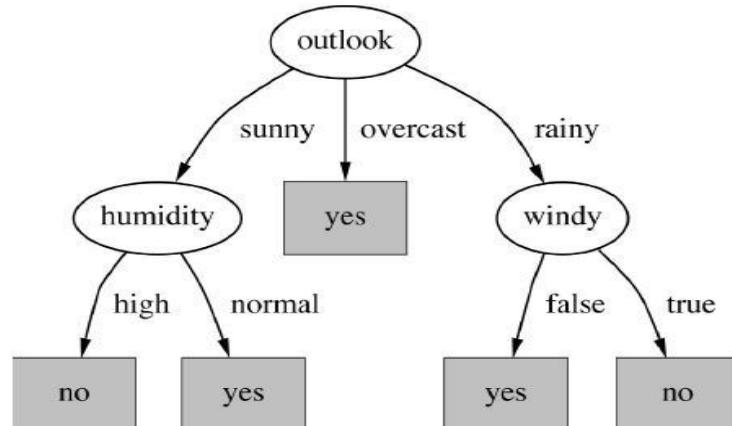
Sunny : 5/14, 3/5 (no) et 2/5 (Yes)

Overcast : 4/14, yes

Rainy : 5/14, yes

Yes : $5/14 * 2/5 + 4/14 + 5/14 = 11/14$

NO : $5/14 * 3/5 = 3/14$



Classification d'une donnée ayant des attributs non valués

- “ Donc, on poursuit la classification en transmettant les poids $5/14$ vers le nœud testant l'attribut « Humidité », **$4/14$ le long de la branche « Couvert » vers l'étiquette « oui » et $5/14$ vers le nœud testant l'attribut « Vent ».**
- “ La valeur de l'attribut « Humidité » est inconnue également. Parmi les exemples « Ensoleillé », il y en a 3 dont l'attribut « Humidité » vaut « Elevée », 2 dont cet attribut vaut « Normale », soit $3/5$ et $2/5$ respectivement. Puisque $5/14$ exemple a suivi cette branche depuis la racine, **on obtient $5/14 \times 3/5 = 3/14$ exemple atteignant l'étiquette « non » et $5/14 \times 2/5 = 1/7$ exemple atteignant l'étiquette « oui ».**
- “ L'attribut « Vent » a la valeur « Faible » ; **le $5/14$ d'exemple qui ont suivi cette branche depuis la racine est donc classé comme « oui ».**
- “ En résumé, il y a $3/14$ exemple qui atteint une étiquette « non » et $1/7 + 4/14 + 5/14 = 11/14$ exemple qui atteint une étiquette « oui ».
- “ On en conclut que la classe la plus probable de cette donnée est « oui ».

ID3 vs. C4.5

“ ID3 construit un arbre de décision :

- les attributs doivent tous être qualitatifs et ils sont considérés comme nominaux ;
- ne peut pas prendre en charge des exemples ou des données dans lesquels il manque la valeur d'attributs ;
- utilise les notions d'entropie et de gain pour déterminer l'attribut à tester dans chaque noeud.

“ En ce qui concerne C4.5 :

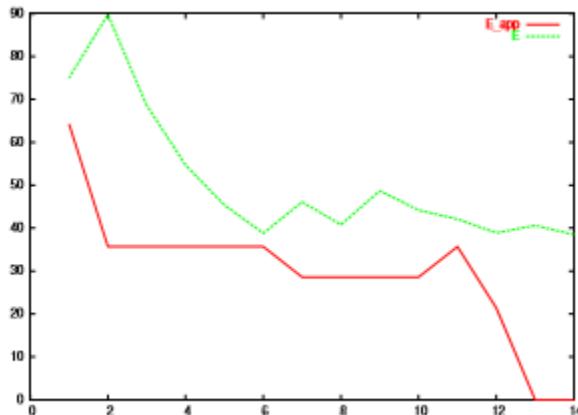
- les attributs peuvent être qualitatifs ou quantitatifs ;
- peut utiliser des exemples dans lesquels la valeur de certains attributs est inconnue lors de la construction de l'arbre de décision ;
- peut prédire la classe de donnée dont la valeur de certains attributs est inconnue ;
- utilise le rapport de gain pour déterminer l'attribut à tester dans chaque noeud.

Surapprentissage (Overfitting)

On dit qu'un algorithme de classification sur-apprend (overfit) aux données d'entraînement s'il génère un arbre de décision (ou toute autre représentation des données) qui dépend trop sur des caractéristiques (features) non pertinentes des instances d'entraînement,

=> S'entraîne bien sur les données d'apprentissage mais relativement mal sur les instances non apprises.

Surapprentissage (Overfitting)



Variation de l'erreur mesurée sur l'ensemble de tous les exemples et de l'erreur E (estimée) en fonction du nombre d'exemples utilisés pour construire l'arbre de décision.

- Ce schéma illustre que plus le nombre d'exemples utilisés pour construire le modèle augmente, plus l'erreur sur ce jeu diminue et tend vers 0.
- Mais pour l'erreur en généralisation, quand le nombre d'exemples utilisés augmente, l'erreur en généralisation commence par diminuer puis elle augmente (c'est précisément là où elle est minimale que l'on a construit le meilleur modèle, celui qui fait une erreur minimale).
- Quand l'apprentissage se poursuit, le modèle se complique, la probabilité d'erreur augmente, et le modèle produit du sur-apprentissage.

Surapprentissage (Overfitting)

Cela signifie que l'arbre donnera une très bonne précision sur l'ensemble de données d'apprentissage mais donnera une mauvaise précision dans les données de test.

Il existe de nombreuses façons pour résoudre ce problème :

Grâce au réglage des hyperparamètres:

Nous pouvons définir la profondeur maximale de notre arbre de décision en utilisant le paramètre **max_depth**. Plus la valeur de max_depth est grande, plus notre arbre sera complexe. L'erreur d'entraînement diminuera si nous augmentons la valeur max_depth mais lorsque nos données de test entreront en jeu, nous obtiendrons une très mauvaise précision.

Par conséquent, Nous avons besoin d'une valeur qui ne sur-ajustera pas ni sous-adaptera à nos données et pour cela, Nous pouvons utiliser GridSearchCV.

Surapprentissage (Overfitting)

min_samples_split une autre méthode qui consiste à définir le nombre minimum d'échantillons pour chaque division. Ici, nous spécifions le nombre minimum d'échantillons requis pour découper. Par exemple, nous pouvons utiliser un minimum de 10 échantillons pour prendre une décision. Cela signifie que si un nœud a moins de 10 échantillons, alors en utilisant ce paramètre, nous pouvons arrêter la division supplémentaire de ce nœud et en faire un nœud feuille.

min_samples_leaf - représente le nombre minimum d'échantillons requis pour être dans le nœud feuille. Plus vous augmentez le nombre, plus il y a de possibilité de surapprentissage.

max_features - cela nous aide à décider du nombre de fonctionnalités à considérer lors de la recherche de la meilleure répartition

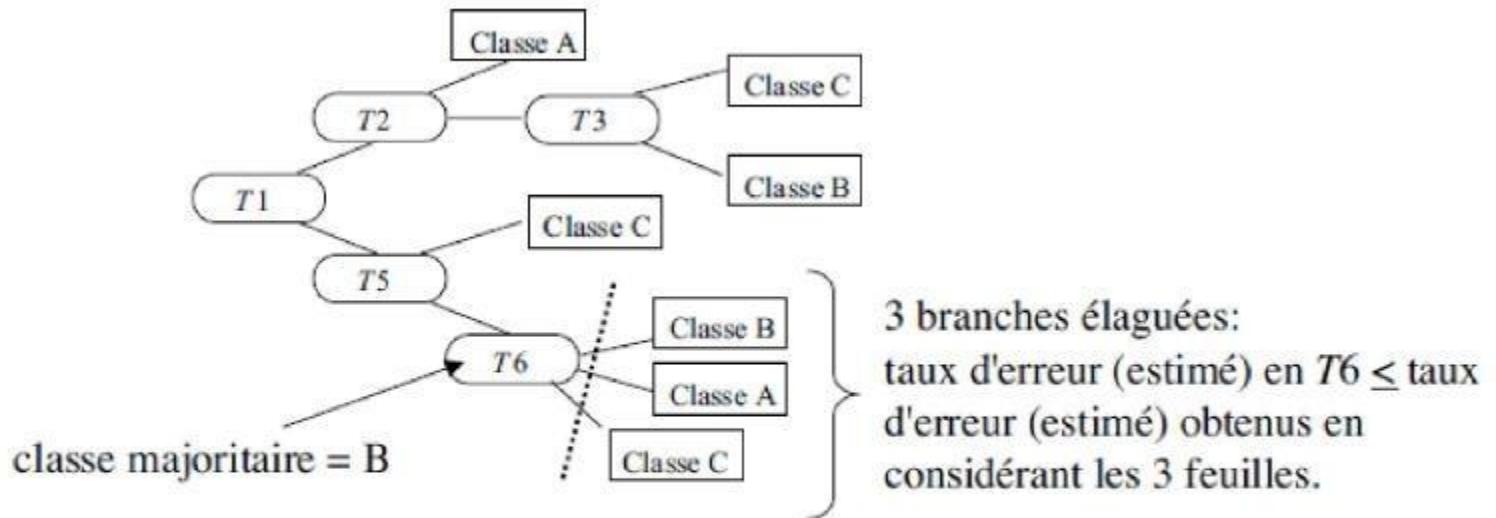
Surapprentissage (Overfitting)

Une deuxième approche standard pour réduire le sur-ajustement est de sacrifier la précision de classification sur l'ensemble d'entraînement en faveur de la précision dans la classification des données de test (invisibles).

Ceci peut être réalisé en élaguant l'arbre de décision.

Prunning

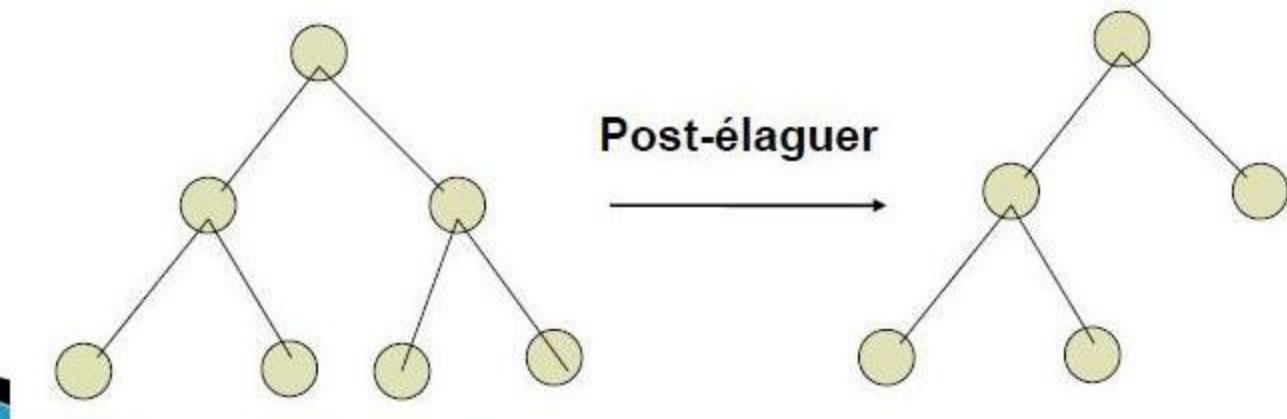
Objectif: supprimer les parties de l'arbre qui ne semblent pas performantes pour prédire la classe de nouveaux cas
=> remplacées par un noeud terminal (associe à la classe majoritaire).



Solutions

“ Deux grandes classes de solutions :

- Arrêter la croissance de l'arbre tôt. En pratique, cette solution est plutôt difficile à implémenter en raison de la difficulté à identifier le point d'arrêt (pré-élagage).
- Déployer l'arbre jusqu'à ce que l'algorithme s'arrête. Post-élaguer l'arbre. C'est une méthode plus populaire.



Pré-élagage : enjeux

- Selon quel critère renoncera-t-on à développer un nœud ?
- Gain pas assez important :
On risque d'ignorer un test qui " prépare le terrain" pour un ou plusieurs tests plus discriminants.
- Comment fixer le seuil des gains intéressants ?
- Intérêt : diminue le temps de calcul de l'arbre .

Pré -élagage

Pour mieux comprendre le concept, voici un exemple :

Chaque branche de l'arbre en évolution correspond à une règle incomplète telle que SI $x = 1$ ET $z = \text{oui}$ ET $q > 63,5 \dots$ PUIS . . . et également à un sous-ensemble d'instances actuellement «en traitement».

Si toutes les instances ont la même classification, disons c_1 , le nœud d'extrémité (branche) est traité par l'algorithme TDIDT comme un nœud feuille étiqueté par c_1 . Une telle branche complétée correspond à une règle (complétée), telle que SI $x = 1$ ET $z = \text{oui}$ ET $q > 63,5$ ALORS classe = c_1 .

Si toutes les instances n'ont pas la même classification, le nœud doit être étendu à un sous-arbre en divisant sur un attribut. En suivant une stratégie de pré-élagage, le nœud (c'est-à-dire le sous-ensemble) est d'abord testé pour déterminer si une condition de résiliation s'applique ou non. Si ce n'est pas le cas, Le nœud est développé comme d'habitude.

Pré -élagage

Si telle est le cas, en utilisant une «suppression de branche», un «vote majoritaire» ou une autre stratégie similaire (La stratégie la plus courante est probablement le «Vote majoritaire»)

Dans ce cas le nœud est traité comme un nœud feuille étiqueté avec la classification la plus fréquente pour les instances du sous-ensemble (la «classe majoritaire»).

L'ensemble des règles pré-élaguées classera à tort certaines des instances dans l'ensemble d'entraînement. Cependant, la précision de classification de l'ensemble de test peut être plus grande que pour l'ensemble de règles non apprises.

Pré -élagage

Plusieurs critères peuvent être appliqués à un nœud pour déterminer si ou non, un pré-élagage doit avoir lieu.

Deux d'entre eux sont:

- Taille limite (size cutoff) : Taillez si le sous-ensemble contient moins de 5 ou 10 instances, disons
- Coupure de profondeur maximale (Maximum depth cutoff) : Taillez si la longueur de la branche est de 3 ou 4.

Pré -élagage

La figure montre les résultats obtenus pour divers ensembles de données en utilisant TDIDT avec gain d'informations pour la sélection d'attributs.

Dans chaque cas, une validation croisée de 10 fois est utilisée, avec un seuil de taille de 5 instances, 10 instances ou pas de seuil (c'est-à-dire non élagué).

	No cutoff		5 Instances		10 Instances	
	Rules	% Acc.	Rules	% Acc.	Rules	% Acc.
breast-cancer	93.2	89.8	78.7	90.6	63.4	91.6
contact_lenses	16.0	92.5	10.6	92.5	8.0	90.7
diabetes	121.9	70.3	97.3	69.4	75.4	70.3
glass	38.3	69.6	30.7	71.0	23.8	71.0
hypo	14.2	99.5	11.6	99.4	11.5	99.4
monk1	37.8	83.9	26.0	75.8	16.8	72.6
monk3	26.5	86.9	19.5	89.3	16.2	90.1
sick-euthyroid	72.8	96.7	59.8	96.7	48.4	96.8
vote	29.2	91.7	19.4	91.0	14.9	92.3
wake_vortex	298.4	71.8	244.6	73.3	190.2	74.3
wake_vortex2	227.1	71.3	191.2	71.4	155.7	72.2

Pré -élagage

La figure montre les résultats avec un seuil de profondeur maximal de 3, 4 ou illimité à la place. La stratégie du «vote majoritaire» est utilisée.

	No cutoff		Length 3		Length 4	
	Rules	% Acc.	Rules	% Acc.	Rules	% Acc.
breast-cancer	93.2	89.8	92.6	89.7	93.2	89.8
contact_lenses	16.0	92.5	8.1	90.7	12.7	94.4
diabetes	121.9	70.3	12.2	74.6	30.3	74.3
glass	38.3	69.6	8.8	66.8	17.7	68.7
hypo	14.2	99.5	6.7	99.2	9.3	99.2
monk1	37.8	83.9	22.1	77.4	31.0	82.2
monk3	26.5	86.9	19.1	87.7	25.6	86.9
sick-euthyroid	72.8	96.7	8.3	97.8	21.7	97.7
vote	29.2	91.7	15.0	91.0	19.1	90.3
wake_vortex	298.4	71.8	74.8	76.8	206.1	74.5
wake_vortex2	227.1	71.3	37.6	76.3	76.2	73.8

Pré -élagage

Les résultats obtenus montrent clairement que le pré-élagage est important. Cependant, il est essentiellement ad hoc. Aucun choix de taille ou de profondeur de coupure ne permet de produit systématiquement de bons résultats sur tous les ensembles de données.

Ce résultat renforce celui de Quinlan, selon lequel le problème avec pré-élagage est que le «seuil d'arrêt» n'est «pas facile à faire

- Un seuil élevé peut mettre fin à la division avant, alors qu'une valeur trop faible n'entraîne guère de simplification ».

De ce fait, il est hautement souhaitable de complètement automatiser ce processus sans le besoin (pour l'utilisateur) de sélectionner une valeur de seuil de coupure.

Un certain nombre de moyens de le faire ont été proposés, mais en pratique l'utilisation du post-élagage, s'est avéré plus populaire.

Post -élagage

“ Principe:

- ✓ Construire l'arbre complet.
- ✓ Pour chaque nœud interne, regarder s'il ne serait pas meilleur de le remplacer :
 - Par une feuille.
 - Par un de ses fils (son fils le plus fréquent).

Deux types d'élagage sont effectués dans C4.5 :

- ✓ remplacement d'un sous-arbre : consiste à remplacer un sous-arbre par une feuille ;
- ✓ promotion d'un sous-arbre : consiste à rassembler deux nœuds dans un seul nœud.

Post -élagage

L'objectif ainsi est de convertir un arbre complet en un plus petit élagué qui prédit la classification des instances invisibles au moins aussi précisément que fait l'arbre complet.

Cette méthode a plusieurs variantes, telles que :

- **Reduced Error Pruning,**
- **Pessimistic Error Pruning,**
- **Minimum Error Pruning**
- **Error Based Pruning**

Les concepts associés à chaque technique variés, mais nous allons voir les concepts de base

Post -élagage

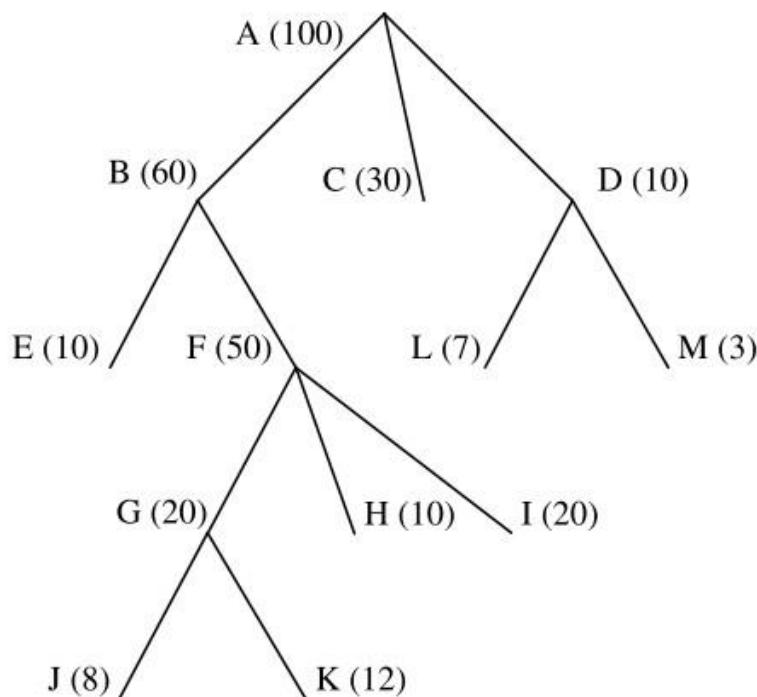
Lors du post-élagage d'un arbre de décision tel que le précédent, nous recherchons des nœuds dans l'arbre qui ont un sous-arbre descendant de profondeur un (c'est-à-dire tous les nœuds d'un niveau inférieur sont des nœuds feuilles). Tous ces sous-arbres sont candidats à la post-taille.

Si une condition d'élagage (qui sera décrite ci-dessous) est remplie le sous-arbre suspendu au nœud peut être remplacé par le nœud lui-même.

Nous travaillons du bas de l'arbre vers le haut et taillez un sous-arbre à la fois. le continue jusqu'à ce que plus aucun sous-arbre ne puisse être élagué.

Post -élagage

Supposons que nous ayons un arbre de décision complet généré par TDIDT.



Dans notre arbre, les seuls candidats à la taille sont les sous-arbres suspendus à partir des nœuds G et D.

Post -élagage

La branche du nœud racine A vers un nœud feuille tel que J correspond à un règle de décision. Nous nous intéressons à la proportion d'instances invisibles pour lesquelles cette règle s'applique qui sont incorrectement classées. Nous appelons cela le taux d'erreur au nœud J (une proportion de 0 à 1).

En partant du bas de l'arbre vers le haut, nous commençons par considérer le remplacement du sous-arbre «suspendu au» nœud G par G lui-même, en tant que nœud feuille dans un arbre élagué.

- Comment se termine le taux d'erreur de la branche (règle tronquée) en G comparer avec le taux d'erreur des deux branches (règles complètes) se terminant chez J et K?
- Est-il avantageux ou nuisible à la précision prédictive de l'arbre de diviser au nœud G?
- Nous pourrions envisager de tronquer la branche plus tôt, disons au nœud F. Cela serait-il bénéfique ou nuisible?

Post -élagage

Pour répondre à de telles questions, nous avons besoin d'un moyen d'estimer le taux d'erreur à n'importe quel nœud d'un arbre.

- Utiliser l'arborescence pour classer les instances dans un ensemble de données précédemment invisibles appelées ensemble d'élagage et compter les erreurs. Notez qu'il est impératif que le jeu d'élagage soit supplémentaire au «Ensemble de test» utilisé. L'utilisation d'un ensemble d'élagage est une approche raisonnable mais peut être irréaliste lorsque la quantité de données disponibles est faible.

Post -élagage

- Une alternative qui prend beaucoup moins de temps d'exécution est d'utiliser une formule pour estimer le taux d'erreur. Une telle formule est susceptible d'être fondée sur des probabilités et d'utiliser des facteurs tels que le nombre d'instances correspondant au nœud appartenant à chacune des classes et la probabilité a priori de chaque classe.

Post élagage de l'arbre

Taille d'erreur réduite

Cette méthode a été proposée par Quinlan. Il s'agit de la méthode la plus simple et la plus compréhensible d'élagage des arbres de décision. Elle considère que chacun des nœuds de décision de l'arbre est candidat à l'élagage. La méthode consiste à supprimer le sous-arbre enraciné à ce nœud, ce qui en fait un nœud feuille.

Les données disponibles sont divisées en trois parties:

- Les exemples de formation,
 - Les exemples de validation utilisés pour l'élagage de l'arbre,
 - Un ensemble d'exemples de tests utilisés pour fournir une estimation impartiale de la précision sur les futurs exemples non vus.
- Si le taux d'erreur du nouvel arbre est égal ou supérieur à celui de l'arbre d'origine et que ce sous-arbre ne contient aucun sous-arbre avec la même propriété, le sous-arbre est remplacé par le nœud feuille, cela signifie que l'élagage est effectué. Sinon, ne le taillez pas.

le coût d'erreur du nœud est calculé à l'aide de l'équation suivante:

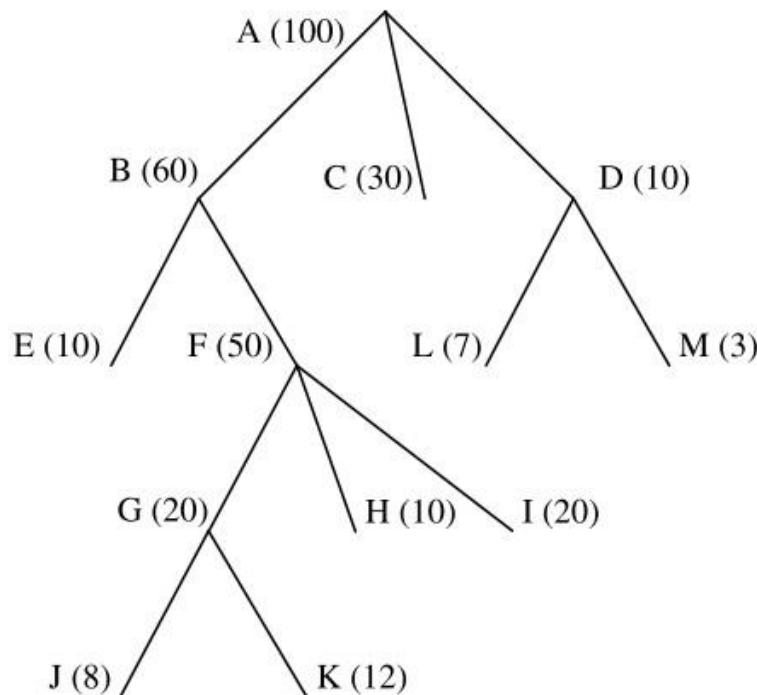
$$R(t) = r(t) \times p(t)$$

$$r(t) = \frac{\text{no of examples missclassified in node}}{\text{no of all examples in node}}$$

$$p(t) = \frac{\text{no of examples in node}}{\text{no of total examples}}$$

Pré -élagage

La branche du nœud racine A vers un nœud feuille tel que J correspond à une règle de décision. Nous nous intéressons à la proportion d'instances invisibles pour lesquelles cette règle s'applique qui sont incorrectement classées. Nous appelons cela le taux d'erreur au nœud J (une proportion de 0 à 1 inclus).

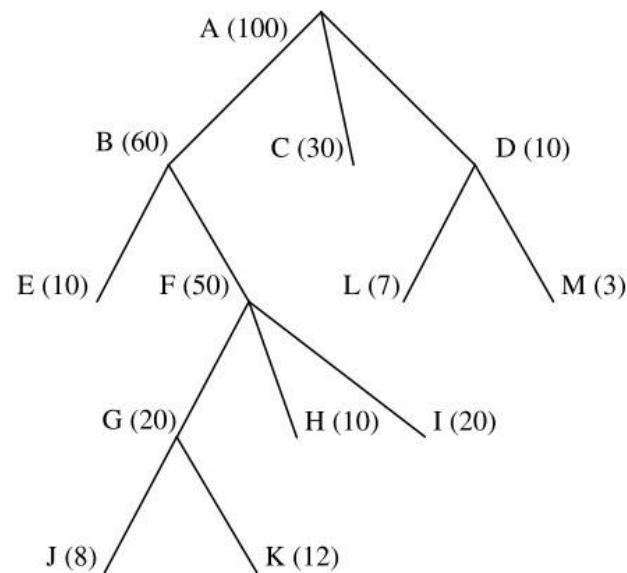


Node	Estimated error rate
A	0.3
B	0.15
C	0.25
D	0.19
E	0.1
F	0.129
G	0.12
H	0.05
I	0.2
J	0.2
K	0.1
L	0.2
M	0.1

Pré -élagage

La branche du nœud racine A vers un nœud feuille tel que J correspond à une règle de décision. Nous nous intéressons à la proportion d'instances invisibles pour lesquelles cette règle s'applique qui sont incorrectement classées. Nous appelons cela le taux d'erreur à nœud J (une proportion de 0 à 1 inclus).

Pour estimer le taux d'erreur du sous-arbre suspendu au nœud G, nous prenons la moyenne pondérée des taux d'erreur estimés à J et K.
la valeur est $(8/20) \times 0,2 + (12/20) \times 0,1 = 0,14$.

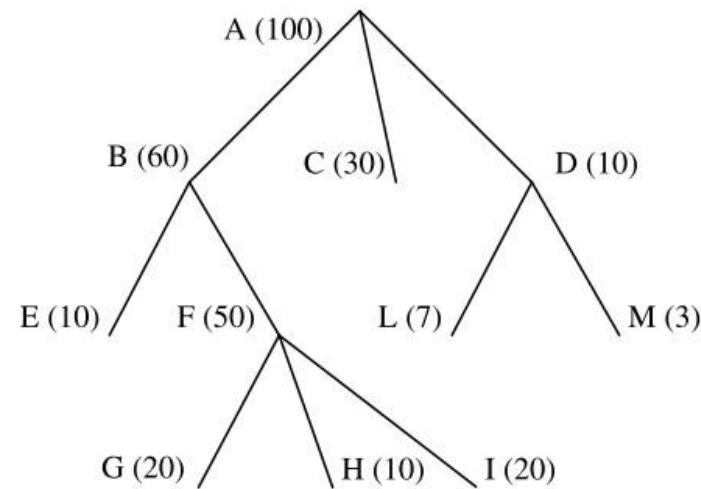


Pré -élagage

Nous appellerons cela l' estimation du taux d'erreur au nœud G car il est calculé à partir de l'estimation du taux d'erreur des nœuds en dessous.

Nous devons maintenant comparer cette valeur avec la valeur obtenue à partir du tableau 0,12, que nous appellerons l'estimation statique du taux d'erreur à ce node.

Dans le cas du nœud G, la valeur statique est inférieure à la valeur sauvegardée. Ce qui signifie que le fractionnement au nœud G augmente le taux d'erreur à ce nœud, qui est manifestement contre-productif. On élague donc le sous-arbre descendant du nœud G pour obtenir.



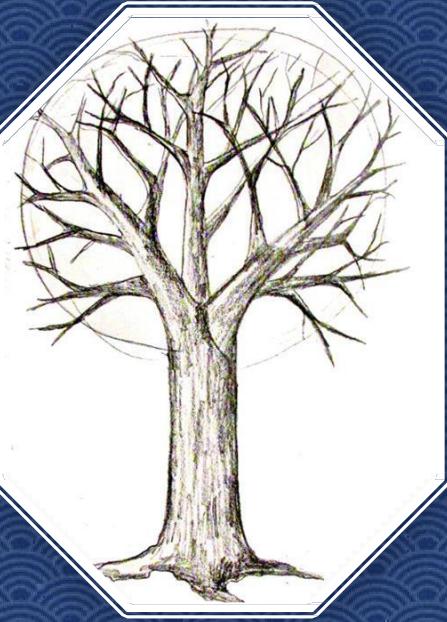
Pré -élagage

Dans un cas extrême, cette méthode pourrait conduire à un arbre de décision post-élagué jusqu'à son nœud racine, indiquant que l'utilisation de l'arbre est susceptible de conduire à un taux d'erreur plus élevé, c'est-à-dire des classifications plus incorrectes. Heureusement ceci est probablement très rares.

Les arbres de décision post-élagage sembleraient être une approche acceptée que de les pré-élaguer. Sans doute la disponibilité immédiate et la popularité du système de classification C4.5 a eu une grande influence sur ceci.

Cependant, une objection pratique importante au post-élagage est qu'il y a une surcharge de calcul importante impliquée dans la génération d'un arbre complet uniquement puis d'en jeter une partie ou peut-être la plupart.

Cela n'a peut-être pas d'importance avec de petits ensembles de données expérimentaux, mais les ensembles de données «réels» peuvent contenir plusieurs millions d'instances et les problèmes de faisabilité informatique et de mise à l'échelle des méthodes deviennent inévitablement importants.



CART

Classification And Regression Tree

Rappel

Prédiction : lorsque la prédiction y' doit être la plus proche possible de la vraie réponse y , associée à x .

Estimation: Il s'agit dans ce cas d'estimer la fonction (inconnue) qui à X associe Y .

la régression et la classification diffèrent par la nature de la sortie Y .

Rappel

Régression : Y est continue, typiquement lorsque $Y = R$. Le modèle statistique s'écrit alors sous la forme suivante :

$$Y = s(X) + \varepsilon$$

La fonction de régression $s : X \rightarrow R$ est inconnue et nous cherchons à l'estimer à partir des mesures (X_i, Y_i) dont nous disposons dans l'échantillon L_n .

Ce modèle statistique est appelé modèle de régression non-paramétrique puisqu'essentiellement aucune contrainte a priori n'est imposée à la fonction de régression s , contrairement aux modèles paramétriques comme par exemple le modèle de régression linéaire. Dans un tel modèle, on cherche en effet s sous la forme d'une combinaison linéaire des coordonnées des composantes de X et les coefficients de cette combinaison linéaire, appelés les paramètres du modèle,

- **Tableau Comparatif Rapide :**

Critère	ID3 (Entropie)	C4.5 (Gain Ratio)	CART (Gini/Variance)
Type	Classification	Classification	Classification + Régression
Variables	Catégorielles	Catégorielles/Numériques	Tous types
Arbre	Binaire/Multiway	Multiway	Binaire strict

Indice de Gini

Un autre algorithme d'arbre de décision CART (Classification and Regression Tree)

Il utilise l'indice de Gini pour créer des points de partage.

Objectif : Minimiser l'**impureté** (définir *Gini* vs *Entropie*).

$$Gini = \sum_{k=1}^K p_k(1 - p_k) = 1 - \sum_{k=1}^K p_k^2$$

Où, pi est la probabilité qu'un tuple de D appartienne à la classe Ci. L'index de Gini considère une division binaire pour chaque attribut.

Vous pouvez calculer une somme pondérée de l'impureté de chaque partition. Si une division binaire sur l'attribut A partitionne les données D en D₁ et D₂, l'index de Gini de D est:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

Dans le cas d'un attribut discret, le sous-ensemble qui donne l'indice gini minimum pour celui choisi est sélectionné en tant qu'attribut de fractionnement. Dans le cas d'attributs à valeurs continues, la stratégie consiste à sélectionner chaque paire de valeurs adjacentes en tant que point de partage possible et point avec un indice de Gini plus petit choisi comme point de fractionnement.

$$\Delta Gini(A) = Gini(D) - Gini_A(D).$$

L'attribut avec un index Gini minimum est choisi comme attribut de division.

Avantages

- Simple à comprendre, interpréter et visualiser.
- Les arbres de décision effectuent implicitement une sélection de caractéristiques.
- Peut gérer à la fois des données numériques et catégoriques .
- Les arbres de décision nécessitent relativement peu d'effort de la part des utilisateurs pour la préparation des données.
- Les relations non linéaires entre les paramètres n'affectent pas les performances de l'arborescence.

désavantages

- Les apprenants peuvent créer des arbres très complexes qui peuvent causer l'overfitting.
- Les arbres de décision peuvent être instables.
- Les algorithmes gloutons ne peuvent pas garantir l'optimal global. Cela peut être atténué par la formation de plusieurs arbres, où les fonctions et les échantillons échantillonnés de manière aléatoire avec remplacement.
- Les apprenants des arbres de décision créent des arbres biaisés si certaines classes dominent . Il est donc recommandé d'équilibrer l'ensemble de données avant de l'ajuster avec l'arbre de décision.



Variable dépendante catégorique

Variable dépendante catégorique

Indice de GINI

Un score de Gini donne une idée de la qualité d'une division par la mixité des classes dans les groupes créés par la division.

$$Gini\ index = 1 - \sum_{i=1}^n p_i^2$$

Indice de Gini

- ◎ Calculer l'index de Gini pour l'ensemble de données.
- ◎ Pour chaque attribut / caractéristique:
 - ◎ Calculer l'indice de Gini pour toutes les valeurs catégorielles
 - ◎ Calculer le gain de GINI
- ◎ Choisissez le meilleur attribut de gain de GINI.
- ◎ Répétez jusqu'à l'obtention de l'arbre souhaité.

Indice de Gini - exemple

P(Past Trend=Positive): 6/10
P(Past Trend=Negative): 4/10

P(Past Trend = Positive & Return = Up) = 4/6
P(Past Trend = Positive & Return = Down) = 2/6

$$\text{Gini index} = 1 - ((4/6)^2 + (2/6)^2) = 0.45$$

p(Past Trend = Negative & Return = Up) = 0
P(Past Trend = Negative & Return = Down) = 4/4

$$\text{Gini index} = 1 - ((0)^2 + (4/4)^2) = 0$$

Alors:

$$\begin{aligned}\text{Gini Gain for Past Trend} &= (6/10)0.45 + (4/10)0 \\ &= 0.27\end{aligned}$$

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Negative	High	Low	Down
Positive	Low	High	Up
Positive	High	High	Up
Negative	Low	High	Down
Positive	Low	Low	Down
Negative	High	High	Down
Negative	Low	High	Down
Positive	Low	Low	Down
Positive	High	High	Up

Indice de Gini - exemple

Indice Gini pour Open Interest
0.47

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Negative	High	Low	Down
Positive	Low	High	Up
Positive	High	High	Up
Negative	Low	High	Down
Positive	Low	Low	Down
Negative	High	High	Down
Negative	Low	High	Down
Positive	Low	Low	Down
Positive	High	High	Up

Indice de Gini - exemple

Indice Gini pour Trading Volume
0.34

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Negative	High	Low	Down
Positive	Low	High	Up
Positive	High	High	Up
Negative	Low	High	Down
Positive	Low	Low	Down
Negative	High	High	Down
Negative	Low	High	Down
Positive	Low	Low	Down
Positive	High	High	Up

Indice de Gini - exemple

GINI
Index

Past
Trend 0.27

Open
Interest 0.47

Trading
Volume 0.34

Indice de Gini - exemple

P(Open Interest=High): 2/6
P(Open Interest=Low): 4/6

P (Open Interest = High & Return = Up),= 2/2
P (Open Interest = High & Return = Down) = 0

$$\text{Gini index} = 1 - (\text{sq}(2/2) + \text{sq}(0)) = 0$$

P (Open Interest = Low & Return = Up) = 2/4
P (Open Interest = Low & Return = Down) = 2/4

$$\text{Gini index} = 1 - (\text{sq}(0) + \text{sq}(2/4)) = 0.50$$

Alors:

$$\begin{aligned}\text{Gini Index for Open Interest} &= (2/6)0 + (4/6)0.50 \\ &= 0.33\end{aligned}$$



Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Positive	Low	High	Up
Positive	High	High	Up
Positive	Low	Low	Down
Positive	Low	Low	Down
Positive	High	High	Up

Indice de Gini - exemple

GINI
Index

Open
Interest 0.33

Trading
Volume 0



Variable dépendante continue

Variable dépendante catégorique

L'écart type

L'écart type est utilisé pour calculer l'homogénéité d'un échantillon numérique. Si l'échantillon numérique est complètement homogène, l'écart-type est égal à zéro.

$$SD = \sqrt{\frac{\sum |x - \bar{x}|^2}{n}}$$

L'écart type

- ◎ L'écart type de la cible est calculé.
- ◎ L'ensemble de données est ensuite divisé en différents attributs. L'écart type pour chaque branche est calculée.
- ◎ L'écart type résultant est soustrait de l'écart type avant la division.
- ◎ L'attribut avec la plus grande réduction d'écart-type est choisi pour le nœud de décision.

L'écart type- exemple

Étape 1:
Calculer l'écart type de la
cible.

**Standard deviation (Hours
Played) = 9.32**

Outlook	Temperature	Humidity	Windy	Hours Played
Rainy	Hot	High	False	26
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	62
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30

L'écart type- exemple

$$S(T, X) = \sum_{c \in X} P(c)S(c)$$

Étape 2:
Calculer l'écart type pour
chaque attribut.

Outlook	Temperature	Humidity	Windy	Hours Played
Rainy	Hot	High	False	26
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	62
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30

L'écart type- exemple

Étape 2:
Calculer l'écart type pour
chaque attribut.

$$\begin{aligned} S(\text{Hours}, \text{Outlook}) &= P(\text{Sunny}) * \\ &S(\text{Sunny}) + P(\text{Overcast}) * \\ &S(\text{Overcast}) + P(\text{Rainy}) * S(\text{Rainy}) \\ &= (4/14) * 3.49 + (5/14) * 7.78 + \\ &(5/14) * 10.87 \\ &= \mathbf{7.66} \end{aligned}$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

L'écart type- exemple

$$SDR(T, X) = S(T) - S(T, X)$$

$$\begin{aligned} SDR(\text{Hours}, \text{Outlook}) \\ = S(\text{Hours}) - S(\text{Hours}, \text{Outlook}) \\ = 9,32 - 7,66 = 1.66 \end{aligned}$$

Étape 3:
Calculer la réduction de l'écart
type pour chaque attribut.

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14



		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
SDR=0.17		

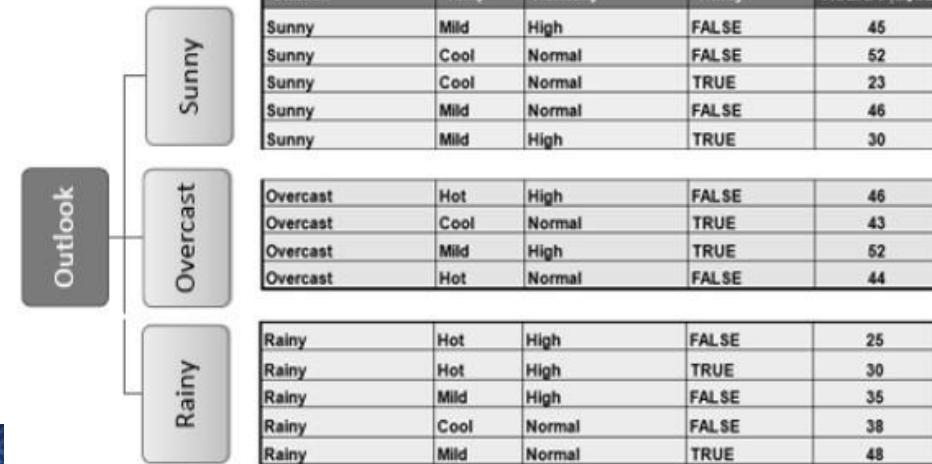
		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
SDR=0.28		

		Hours Played (StDev)
Windy	False	7.87
	True	10.59
SDR=0.29		



L'écart type- exemple

Étape 4:
Le jeu de données est divisé en
fonction des valeurs de l'attribut
sélectionné.



critères d'arrêt

- ◎ Échantillons minimum pour un fractionnement de nœud
- ◎ Nombre minimal d'échantillons pour un nœud terminal
- ◎ Profondeur maximale de l'arbre
- ◎ Nombre maximum de nœuds terminaux.
- ◎ ...

Remarque :

Remarques:

1.CART est un outil puissant mais naïf :

1. Il ne "voit pas" la forêt (globalité) , chaque feuille prédit la **moyenne** des y (vs **classe majoritaire** en classification).

2.La régression CART a des limites structurelles :

1. Impossible de capturer des tendances linéaires/non-linéaires fines sans méthodes ensemblistes.

3.Solutions = Compromis :

1. Interprétabilité (CART élagué)
2. Performance (Forêts Aléatoire/Boosting).

Les méthodes d'ensemble

Quelques Concepts

méthode ensembliste

Principe de méthode ensembliste

L'objectif des méthodes d'ensemble (voir Dietterich (2000a)) est de construire une collection de prédicteurs, pour ensuite agréger l'ensemble de leurs prédictions. En régression, agréger les prédictions de q prédicteurs revient par exemple à en faire la moyenne : chaque prédicteur fournit un y_l , et la prédition finale est $\frac{1}{q} \sum_{l=1}^q \hat{y}_l$.

En classification, l'agrégation consiste par exemple à faire un vote majoritaire parmi les labels des classes fournis par les prédicteurs.

Soulignons le fait que l'étape d'agrégation de ces méthodes est toujours très simple et n'est pas optimisée, contrairement aux méthodes dites d'agrégation de modèles, qui pour une famille de prédicteurs donnée, cherche la meilleure manière de les combiner pour obtenir un bon prédicteur agrégé

Méthode dite d'aggregation

Ainsi, au lieu d'essayer d'optimiser une méthode “en un coup”, les méthodes d’ensemble génèrent plusieurs règles de prédiction et mettent ensuite en commun leurs différentes réponses.

L’heuristique de ces méthodes est qu’en générant beaucoup de prédicteurs, on explore largement l’espace des solutions, et qu’en agrégeant toutes les prédictions, on dispose d’un prédicteur qui rend compte de cette exploration. Ainsi, on s’attend à ce que le prédicteur final soit meilleur que chacun des prédicteurs individuels.

Bagging

bagging

C'est une méthode introduite par Breiman (1996) pour les arbres, et directement issue de la remarque selon laquelle les arbres CART sont instables.

Considérons une méthode de prédiction (appelée règle de base), qui construit sur L_n un prédicteur $h^*(.,L_n)$.

Le principe du Bagging est de tirer un grand nombre d'échantillons, indépendamment les uns des autres, et de construire, en appliquant à chacun d'eux la règle de base, un grand nombre de prédicteurs. La collection de prédicteurs est alors agrégée en faisant simplement une moyenne ou un vote majoritaire.

Le tirage se fait de façon aléatoire et avec remise

Bagging utilisation pour CART

On estime assez naturellement l'erreur de prédiction par bootstrap out of bag, ce qui prévient le sur-ajustement

Pour CART, on peut choisir parmi différentes stratégies :

- ◎ Construire des arbres complets sans élagage
- ◎ Construire un arbre d'au plus q feuilles
- ◎ Construire des arbres complets et élaguer par validation croisée

En pratique, on garde souvent la première stratégie, chacun des arbres a un faible biais mais une grande variance

La moyenne réduit avantageusement cette variance

Avantage : très simple à mettre en œuvre

Inconvénients : Temps de calcul un petit peu important

Boosting

Boosting

Introduit par Freund et al. (1996), le Boosting est une des méthodes d'ensemble les plus performantes à ce jour.

Étant donné un échantillon d'apprentissage L_n et une méthode de prédiction (ou règle de base), qui construit sur L_n un prédicteur $h^*(.,L_n)$. Le principe du Boosting est de tirer un premier échantillon bootstrap $L_{\Theta 1,n}$, où chaque observation a une probabilité $1/n$ d'être tirée, puis d'appliquer la règle de base pour obtenir un premier prédicteur $h^*(.,L_{\Theta 1,n})$.

Ensuite, l'erreur de $h^*(.,L_{\Theta 1,n})$ sur l'échantillon d'apprentissage L_n est calculée.

Boosting

Un deuxième échantillon bootstrap $L_{\Theta 2}$ n est alors tiré mais la loi du tirage des observations n'est maintenant plus uniforme. La probabilité pour une observation d'être tirée dépend de la prédiction de $h^*(.,L_{\Theta 1} n)$ sur cette observation. Le principe est, par le biais d'une mise à jour exponentielle bien choisie, d'augmenter la probabilité de tirer une observation mal prédite et de diminuer celle de tirer une observation bien prédite. Une fois le nouvel échantillon $L_{\Theta 2}$ n obtenu, on applique à nouveau la règle de base $h^*(.,L_{\Theta 2} n)$.

On tire alors un troisième échantillon $L_{\Theta 3}$ n , qui dépend des prédictions de $h^*(.,L_{\Theta 2} n)$ sur L_n et ainsi de suite. La collection de prédicteurs obtenus est alors agrégée en faisant une moyenne pondérée, là encore via des poids exponentiels bien choisis.

Boosting vs bagging

Le Boosting est une méthode séquentielle, chaque échantillon étant tiré en fonction des performances de la règle de base appliquée sur l'échantillon précédent.

En cela, le Boosting diffère de façon importante du Bagging, où les échantillons sont tirés indépendamment les uns des autres, et peuvent être obtenus en parallèle.

L'idée du Boosting est de se concentrer de plus en plus sur les observations mal prédites par la règle de base, pour essayer d'apprendre au mieux cette partie difficile de l'échantillon en vue d'améliorer les performances globales.

Randomizing outputs

Breiman (2000a) introduit la méthode Randomizing Outputs pour les problèmes de régression, qui est une méthode d'ensemble de nature différente.

Le principe est de construire des échantillons indépendants dans lesquels on altère les sorties de l'échantillon d'apprentissage. La modification que subissent les sorties est obtenue en rajoutant une variable de bruit à chaque Y_i de L_n .

On obtient alors une collection d'échantillons “à sorties randomisées”, puis on applique une règle de base sur chacun et on agrège enfin l'ensemble des prédicteurs obtenus.

L'idée de Randomizing Outputs est, encore, qu'en appliquant une règle de base sur des échantillons à sorties randomisées, on obtient une collection de prédicteurs différents les uns des autres.

Random Subspace

Un autre type de méthode d'ensemble est introduit dans Ho (1998). Il n'est plus ici question de jouer sur l'échantillon, mais plutôt d'agir sur l'ensemble des variables considérées.

Le principe de la méthode Random Subspace est de tirer aléatoirement un sous-ensemble de variables et d'appliquer une règle de base sur Ln qui ne prend en compte que les variables sélectionnées.

On génère alors une collection de prédicteurs chacun construit en utilisant des variables différentes, puis on agrège ces prédicteurs.

Les sous-ensembles de variables sont tirés indépendamment pour chaque prédicteur. L'idée de cette méthode est de construire plusieurs prédicteurs, chacun étant spécialisé et réputé bon dans un sous-espace de X particulier, pour en prédicteur sur l'espace d'entrée tout entier.

Random Forest

Random Forest

Algorithmes très récents (années 2000) pour la classification et la regression

Amélioration du Bagging dans le cas spécifique de l'algorithme CART

Idée : Utiliser une combinaison ou une aggregation d'un grand nombre de modèles tout en évitant l'overfitting

RF peuvent être utilisées pour variable de réponse catégorielle, «classification», ou continue «régression».

Les variables prédictives peuvent être soit catégorique ou continue.

L'objectif est de rendre les modèles (arbres) construits plus indépendants entre eux

Cette indépendance va permettre de rendre le vote des experts (différents CART) plus efficace

Approche très fructueuse en grande dimension, par exemple dans le cadre des bio-puces, signaux, images ou courbes

Avantages :

Encore une fois très simple à mettre en œuvre

Cout numérique peu important en regard des performances obtenues

Random Forest Avantages

- gérer naturellement la régression et la classification (multiclasse);
- sont relativement rapides à former et à prévoir;
- ne dépendent que d'un ou deux paramètres de réglage;
- avoir une estimation intégrée de l'erreur de généralisation;
- peut être utilisé directement pour des problèmes de grande dimension;
- peut facilement être mis en œuvre en parallèle.
- mesures d'importance variable;
- pondération différentielle des classes;
- imputation des valeurs manquantes;

Algorithme de mise en œuvre

40

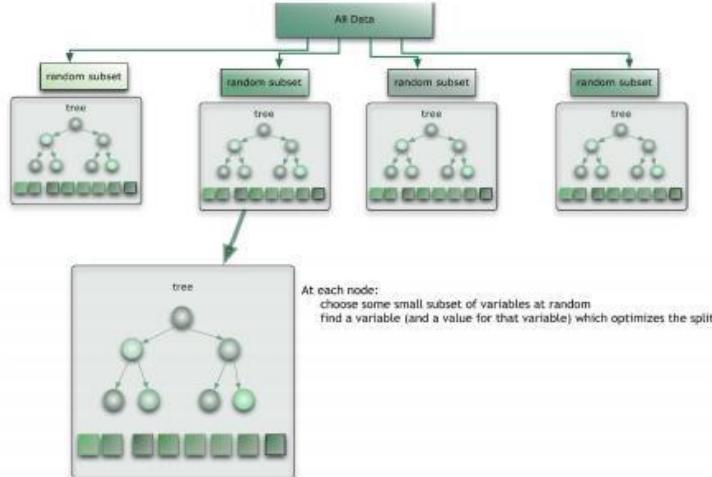
- $z = \{(x_1, y_1) \dots (x_n, y_n)\}$ échantillon d'apprentissage, x_i décrit par p variables explicatives
- Pour $b = 1 \dots B$ (B représente le nombre d'arbres formés dans la forêt)
 - Tirer un échantillon aléatoire z_b avec remise parmi z
 - Estimer un arbre sur z_b avec *randomisation* des variables :
 - Pour la construction de chaque noeud de chaque arbre, *on tire uniformément q variables parmi p* pour former la décision associée au noeud
- En fin d'algorithme, on possède B arbres que l'on moyenne ou qu'on fait voter pour la régression ou la classification
- En général, un choix optimal pour q est à peu près $q = \sqrt{p}$

Algorithme de mise en œuvre

But = obtenir des arbres les + décorrélatés possible

→ chaque arbre appris sur un sous-ensemble (~2/3) aléatoire différent de s exemples d'apprentissage

→ chaque nœud de chaque arbre choisi comme « split » optimal parmi k variables tirées aléatoirement dans les entrées (avec $k \ll d$ la dim des entrées)



Stratégie d'élagage d'un RF

- ◎ Chaque arbre appris avec algo CART sans élagage
- ◎ On limite fortement la profondeur p des arbres (~ 2 à 5)
- ◎ On se limite en général à des arbres de très faibles profondeur (voire $q = 2$)

(Dans le cas du Bagging, il faut un arbre profond pour qu'il ne soit pas trop correlé aux arbres formés par Bagging puisque seul les échantillons d'apprentissage changent)

Dans le cas de Random Forests, le tirage aléatoire des variables explicatives à chaque noeud aboutit à des arbres non corrélés. Chacun des petits arbres est moins performant mais l'union fait la force.

FOUILLE DE DONNÉE

Pr Nawal SAEL

OBJECTIF

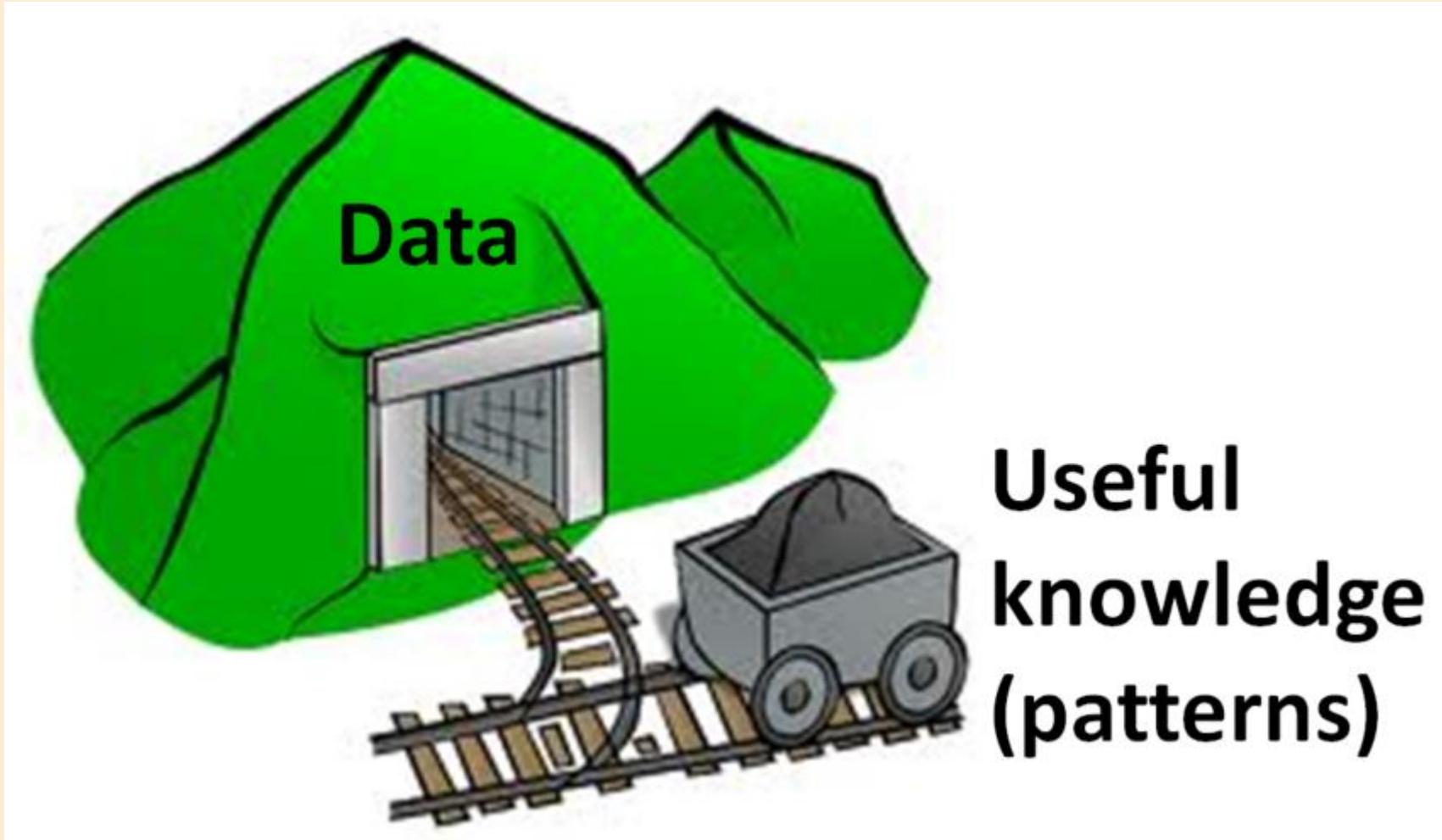
L'objectif de ce cours est de présenter le processus de mise en œuvre d'un projet de fouille de donnée et permettre sa mise en oeuvre

DÉROULEMENT

- ✖ Introduction générale
- ✖ Acquisition des données
- ✖ Visualisation des données et analyse exploratoire
- ✖ Prétraitement :
 - ✖ Valeurs manquantes....
 - ✖ Valeurs aberrantes
 - ✖ Vérification des déséquilibres
 - ✖ Transformation des données
- ✖ Feature engeneering
- ✖ Evaluation d'un modèle
- ✖ Algorithme supervisé : DT et RF
- ✖ Algorithme non supervisé :
 - ✖ K-mean, classification hiérarchique ascendante
 - ✖ Règle d'association (Apriori)

-
- ❖ Pourquoi la fouille de donnée?
 - ❖ Qu'est ce que la fouille de donnée
 - ❖ la fouille de donnée et d'autres concepts
 - ❖ ECD?
 - ❖ Processus de mise oeuvre?
 - ❖ Domaines d'application?
 - ❖ Des exemples ...

POURQUOI LA FOUILLE DE DONNÉE



**Useful
knowledge
(patterns)**

Qu'est ce que La fouille de donnée

- ▶ Terme récent (1995) représentant un mélange d'idées et d'outils provenant de la Statistique, Science de l'information et l'Informatique.
- ▶ Extraction d'informations intéressantes
 - non triviales,
 - implicites,
 - préalablement inconnues et
 - potentiellement utiles

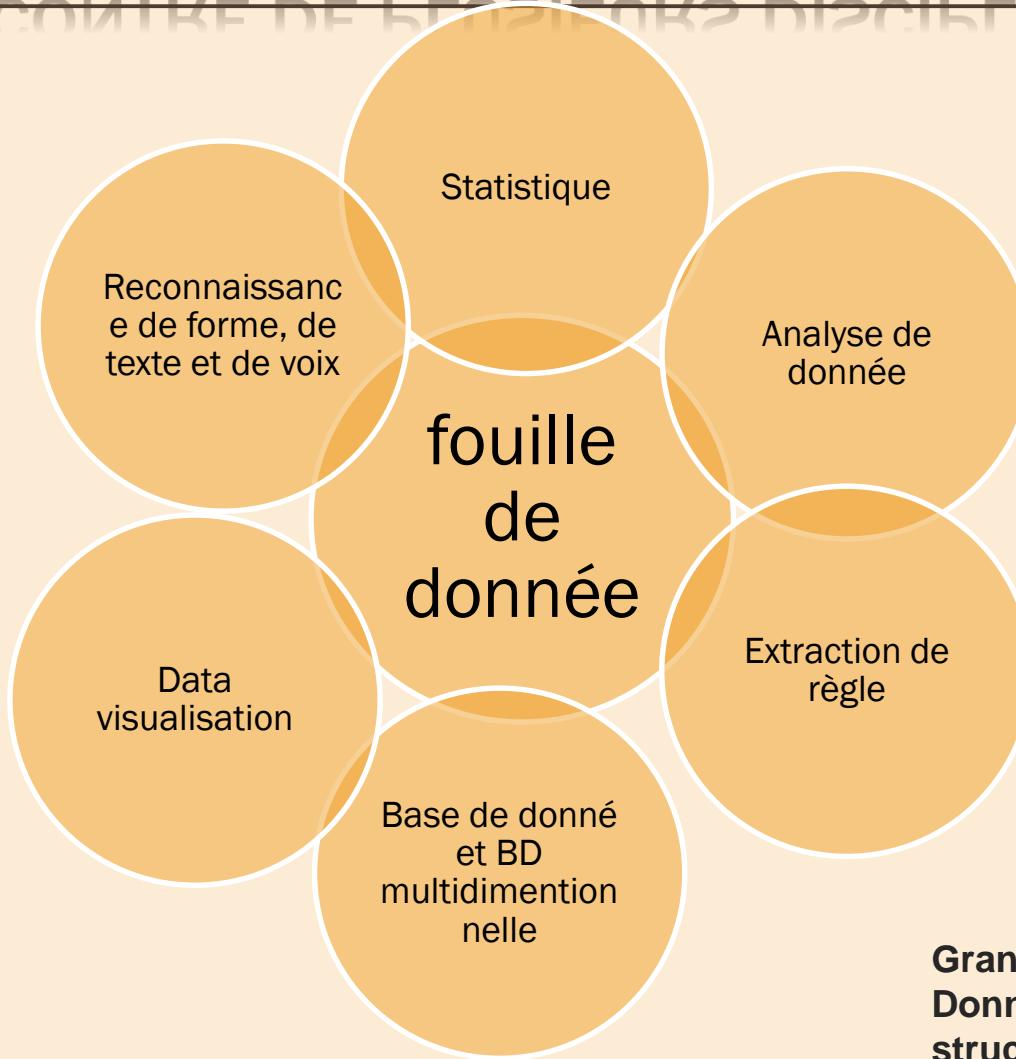
à partir de données.
- ▶ Autres appellations:
 - ECD (Extraction de Connaissances à partir de Données)
 - KDD (Knowledge Discovery from Databases)
 - Analyse de données/patterns, business intelligence, fouille de données, etc.

DÉFINITION 2

- Un processus de découverte de règle, relations, corrélations et/ou dépendances à travers une grande quantité de données, grâce à des méthodes statistiques, mathématiques, d'apprentissage automatique et reconnaissances de formes.

Les origines du fouille de donnée

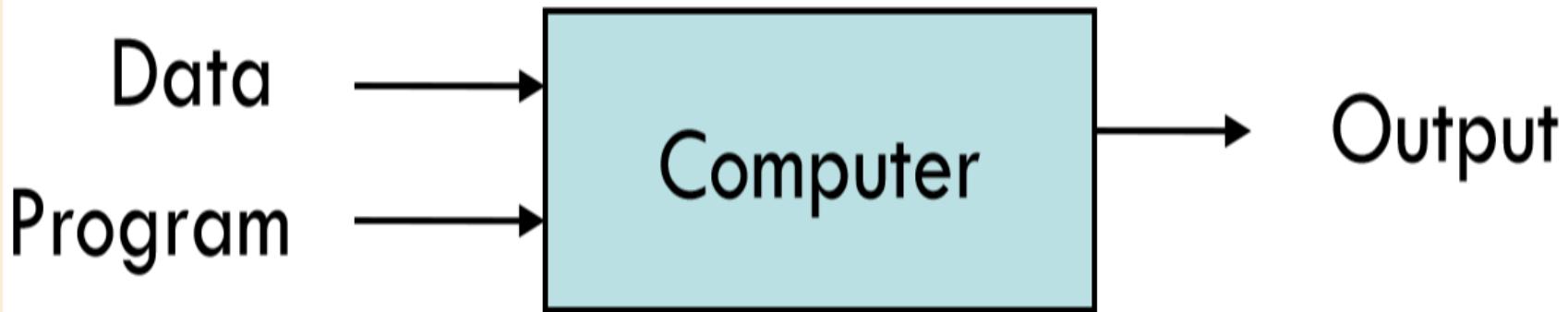
RENCONTRE DE PLUSIEURS DISCIPLINE



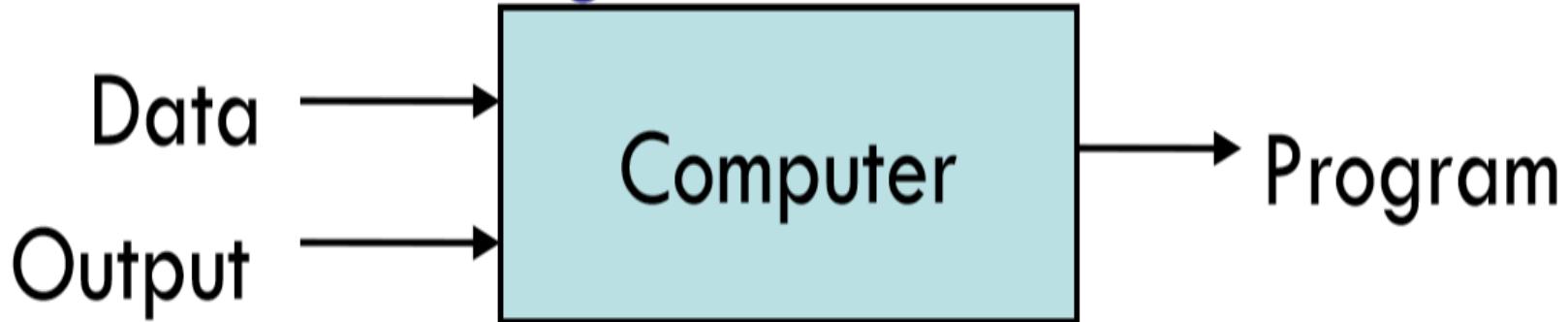
Grande dimension des données -
Données hétérogènes - Données non
structurées

POURQUOI LA FOUILLE DE DONNÉE

Traditional Programming



Machine Learning



INTRODUCTION À LA FOUILLE DE DONNÉE

Programmation classique:

- Développement de programmes pour la gestion de stock
- Administration d'une entreprise,
- Gestion d'un cabinet de formations,

On peut trouver un algorithme qui permet de trouver les résultats à partir des données saisies par l'utilisateur.

Exemple:

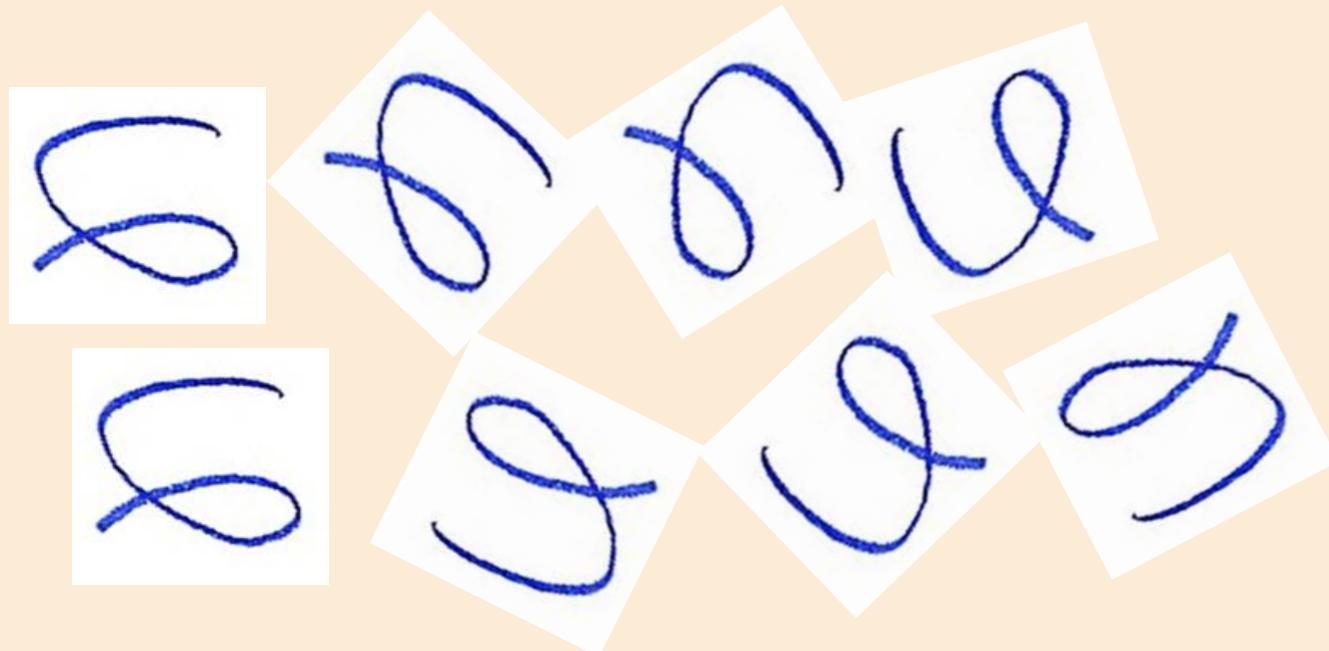
Programmation classique



INTRODUCTION À LA FOUILLE DE DONNÉE

fouille de donnée: On ne dispose pas d'une formule ou d'une méthode analytique qui permet de trouver les résultats à partir des données.

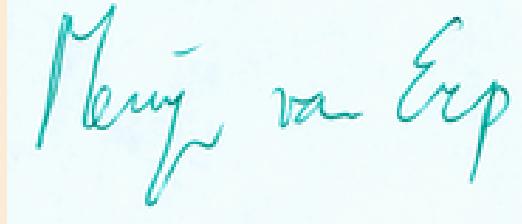
Connaissez vous une méthode analytique ou un algorithme précis qui permet de reconnaître l'une des images suivantes ? S'agit-il du chiffre 6 ou 9 ou la lettre v ?



INTRODUCTION À LA FOUILLE DE DONNÉE

Vérification de signature sur un chèque:

Connaissez vous une méthode analytique ou un algorithme précis qui permet de vérifier si l'image suivante (extraite de la partie à droite en bas d'un chèque) représente la signature correcte d'un client d'une banque ou non ?



fouille de donnée pour la lutte contre covid-19:

ML pour prédire la propagation du virus, pour aider à diagnostiquer le virus, prédire la mortalité,...

Préparation d'un modèle de deep learning pour identifier le COVID-19 au scanner....

QUELQUES APPLICATIONS DE FOUILLE DE DONNÉE

Exemples:

- **Prédiction des prix:** Estimer le prix d'une maison en fonction de sa superficie, sa localisation, possibilité de Parking ou non etc... Ces estimations sont faites en observant d'autres produits similaires pour en tirer des conclusions.
- **Diagnostique médical:** En se basant sur les données médicales d'un patient, l'algorithme peut diagnostiquer si le sujet est atteint d'une maladie donnée. Parfois, ces algorithmes peuvent alerter d'un incident grave de santé avant que cela n'arrive, notamment pour les crises cardiaques.

QUELQUES APPLICATIONS DE FOUILLE DE DONNÉE

https://www.01net.com/actualites/ia-ces-chercheurs-ont-decouvert-un-puissant-antibiotique-grace-a-l-apprentissage-automatique-1862232.html?fbclid=IwAR0sdwCFrxJdMg2d75jAtGZ1EF_JH21ouRfr98v2hx2cEHpzYcH0R9QSTvM

Le 21 avril 2020, Des chercheurs du MIT (Massachusetts Institute of Technology) ont créé un algorithme capable de trouver automatiquement de nouveaux antibiotiques, ce qui devrait ravir l'industrie pharmaceutique en s'appuyant sur des techniques d'apprentissage automatique.

Cette découverte est d'autant plus importante que les bactéries ont tendance à devenir de plus en plus résistantes. Cette nouvelle méthode fondée sur l'intelligence artificielle pourrait donc sauver les vies des millions de personnes.

QUELQUES APPLICATIONS DE FOUILLE DE DONNÉE

- **Recommandation de produits:** Ce type de système se base sur les historiques d'achats, les recherches faites en ligne (Tracking Web) par un internaute pour lui recommander des produits qui pourront l'intéresser. Pour Amazon, cette fonctionnalité est critique car elle est au cœur de l'augmentation des volumes de vente et par conséquent des gains de la société.
- **Regroupement d'items:** Ce type de technique sert notamment pour l'application Iphoto d'Apple pour regrouper les images en fonction des gens qui s'y retrouvent. Généralement, les données n'ont pas d'étiquettes, et l'algorithme tentera de retrouver des items similaires et les regroupera dans un même groupe.

QUELQUES APPLICATIONS DE FOUILLE DE DONNÉE

- **Conduite autonome:** En apprenant le comportement de conduite des humains, les algorithmes de Deep Learning avec l'apprentissage par renforcement (reinforcement learning) permettent d'apprendre des tâches complexes comme la conduite.

- **Banque et Assurance:**

- Prédire qu'un client va quitter sa banque.

- Définir ceux qui veulent changer leur assurance vie.

- Améliorer la satisfaction client en proposant les offres les plus

- pertinentes possibles.

- Prise de décision pour donner un crédit à un client ou non, prédire les risques,

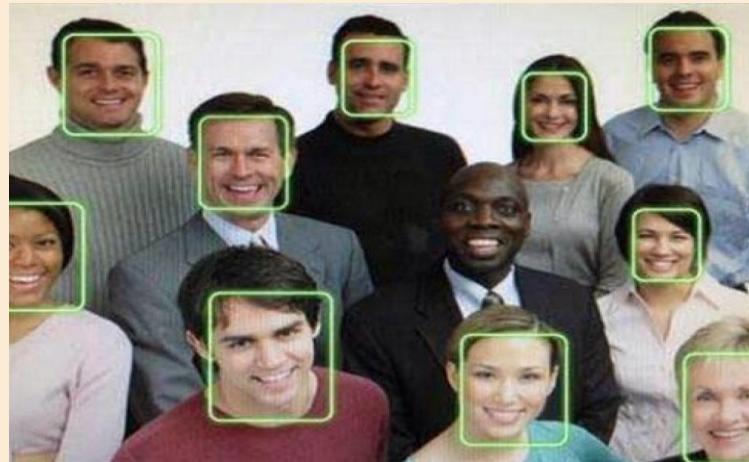
QUELQUES APPLICATIONS DE FOUILLE DE DONNÉE

Une banque reçoit quotidiennement plusieurs demandes d'approbation de crédit. Elle veut automatiser leurs processus d'évaluation. La banque n'a pas de formule magique pour savoir quand il faut accorder un crédit, mais il a beaucoup de données.

L'existence de données nous ramène à l'approche d'apprentissage des données. Donc, la banque utilise l'historique des documents des anciens clients pour trouver la bonne formule d'approbation de crédit.

QUELQUES APPLICATIONS DE FOUILLE DE DONNÉE

La reconnaissance faciale: une catégorie de logiciels biométriques. Le logiciel utilise des algorithmes de deep learning pour comparer une capture en direct ou une image numérique à l'empreinte stockée afin de vérifier l'identité d'un individu. Plusieurs techniques de reconnaissance faciale sont implémentées.



Peut être utilisée pour le contrôle d'absence des étudiants ou₁₈ d'employés d'une entreprise.

QUELQUES APPLICATIONS DE FOUILLE DE DONNÉE

La détection d'émotions dans le texte ou dans l'image est un usage classique du **fouille de donnée** qu'on désigne parfois par le terme de « **sentiment analysis** ».



Peut être utilisé pour contrôler le comportement des employés en contact direct avec les clients...

DOMAINE D'APPLICATION

- Recherche sur Internet
- Biologie computationnelle
- La médecine
- La finance
- Le Commerce électronique
- L'exploration de l'espace
- La robotique
- L'extraction d'informations
- Les réseaux sociaux
- Le débogage des logiciels
-

FOUILLE DE DONNÉE VS. INFORMATIQUE DÉCISIONNELLE (BUSINESS INTELLIGENCE)

- ▶ L'informatique décisionnelle (... BI pour Business Intelligence) désigne les moyens, les outils et les méthodes qui permettent de collecter, consolider, modéliser et restituer les données d'une entreprise en vue d'offrir une aide à la décision et de permettre aux responsables de la stratégie d'une entreprise d'avoir une vue d'ensemble de l'activité traitée.

- Sélectionner les données (par rapport à un sujet et/ou une période)
- Trier, regrouper ou répartir ces données selon certains critères
- élaborer des calculs récapitulatifs « simples » (totaux, moyennes conditionnelles, etc.)
- Présenter les résultats de manière synthétique (graphique et/ou tableaux de bord) REPORTING

- ▶ La fouille de donnée est proche de ce cadre, mais il introduit une dimension supplémentaire qui est la modélisation « exploratoire » (déttection des liens de cause à effet, validation de leur reproductibilité)

FOUILLE DE DONNÉE VS BIG DATA

Big data est un terme qui désigne un grand ensemble de données, qui dépassent le type simple d'architectures de gestion de données.

Le data mining se réfère à l'activité consistant à parcourir de grands ensembles de données pour rechercher des informations pertinentes.

FOUILLE DE DONNÉE VS APPRENTISSAGE AUTOMATIQUE (MACHINE LEARNING)

- ✖ Macjine Learning concerne l'étude, la conception et le développement d'algorithmes qui permettent aux ordinateurs d'apprendre sans être explicitement programmés (définition d'Arthur Samuel).
- ✖ La fouille de donnée peut être définie comme le processus qui, à partir de données apparemment non structurées, tente d'extraire des connaissances et / ou des modèles intéressants inconnus. Au cours de ce processus, des algorithmes d'apprentissage automatique sont utilisés

DES STATISTIQUES ... À LA FOUILLE DE DONNÉE

► Statistique

Les **statistiques** sont principalement une discipline mathématique qui se concentre sur la collecte, l'analyse et l'interprétation de données. Elles utilisent des méthodes de calculs pour faire des inférences à partir d'échantillons de données, estimer des paramètres, et tester des hypothèses.

- Quelques centaines d'individus
- Quelques variables recueillies
- Fortes hypothèses sur les lois statistiques suivies

DES STATISTIQUES ... À LA FOUILLE DE DONNÉE

Statistique :

Objectif : Décrire les caractéristiques d'un ensemble de données (description), estimer des paramètres (moyenne, écart-type, etc.), tester des hypothèses et faire des prédictions basées sur des lois probabilistes.

Exemples de techniques statistiques :

- ✖ Calcul de la moyenne, de la médiane, et de l'écart-type.
- ✖ Test d'hypothèses (ex. : test t de Student, test du chi carré).
- ✖ Estimation de la probabilité d'un événement

DES STATISTIQUES ... À LA FOUILLE DE DONNÉE

Analyse de données :

Englobe des méthodes pour explorer, nettoyer, transformer et organiser les données en vue de les comprendre et de prendre des décisions éclairées. Elle inclut aussi bien des méthodes statistiques que des techniques de visualisation et de modélisation des données.

- Quelques dizaines de milliers d'individus
- Quelques dizaines de variables
- Construction de tableaux: Individus * Variables
- Importance du calcul et de la représentation visuelle

DES STATISTIQUES ... À LA FOUILLE DE DONNÉE

Analyse de données :

Objectif : Comprendre les données, découvrir des relations ou des tendances, et souvent visualiser ces informations pour faciliter la prise de décision.

Exemples d'analyse de données :

- ✖ Exploration visuelle des données à l'aide de graphiques (histogrammes, diagrammes de dispersion, etc.).
- ✖ Regroupement de données (ex. : segmentation de clients).
- ✖ Création de modèles simples pour comprendre les relations entre les variables.

DES STATISTIQUES ... À LA FOUILLE DE DONNÉE

✗ Fouille de donnée

La **fouille de données** est un processus plus avancé qui utilise des techniques d'apprentissage automatique et des algorithmes pour découvrir des modèles cachés ou des connaissances dans de grandes quantités de données

- + Quelques millions d'individus
- + Quelques centaines de variables
- + Nombreuses variables non numériques
- + Population constamment évolutive (difficulté de l'échantillonage)
- + Nécessité de calcul rapide
- + On ne cherche pas nécessairement l'optimum mathématique mais plutôt un modèle qu'un non statisticien pourrait appréhender

DES STATISTIQUES ... À LA FOUILLE DE DONNÉE

✗ Fouille de donnée

Objectif : Découvrir des modèles cachés ou des structures dans de grandes quantités de données, souvent pour des prédictions, des recommandations ou des classifications.

Exemple concret : Analyse d'un dataset de clients

Imaginons un scénario où nous avons un **jeu de données de clients d'un magasin**. Les données comprennent des informations comme :

- L'âge des clients
- Le sexe des clients
- Le montant total des achats
- La fréquence des visites au magasin

L'objectif est de comprendre quel facteur influence le plus les dépenses d'un client dans ce magasin.

A DISTINGUER

- ✖ La fouille de donnée n'est pas du tout la génération de cubes multidimensionnels à partir d'une BD Relationnelle
- ✖ La fouille de donnée est très différent de :
 - + Chercher un numéro de tel dans une grande BD
 - + Trouver des mots clés à partir de google
 - + Générer un histogramme des salaires pour différents groupes d'âge
 - + envoyer une requête SQL dans une base de données et lire la réponse

LA FOUILLE DE DONNÉE EST

- Trouver des groupes de personnes ayant des préférences similaires
- Les chances de cancer sont-elles plus élevées si vous habitez près d'une ligne électrique?

LE FOUILLE DE DONNÉE AUJOURD'HUI

- ✖ Ses techniques ne sont pas toutes récentes
- ✖ Ce qui est nouveau
 - + Grandes capacités de stockage et de traitement, ce qui permet de faire sortir le DM des labos de recherche pour entrer dans les entreprises
- ✖ Du fouille de donnée au data science (data Analytics, DM, ML, DL, Transformes, IA générative...)

DES SCÉNARIOS D'APPLICATION DE LA FOUILLE DE DONNÉE

Issu du livre de Adriaans and Zantige (d'après B. Espinasse)

- ▶ Un éditeur vend 5 sortes de magazines : sport, voiture, maison, musique, cinéma.
- ▶ Il veut étudier ses clients pour découvrir de nouveaux marchés ou vendre plus à ses clients habituels.

Quelques questions

1. Combien de personnes ont pris un abonnement à un magazine de cinéma cette année ?
2. A-t-on vendu plus d'abonnement de magazines de sport cette année que l'année dernière ?
3. Est-ce que les acheteurs de magazines de musique sont aussi amateurs de cinéma ?
4. Quelles sont les caractéristiques principales des lecteurs de magazine de cinéma ?
5. Peut-on prévoir les pertes de clients et prévoir des mesures pour les diminuer ?

1 : Combien de personnes ont pris un abonnement à un magazine de cinéma cette année ?

Requête SQL à partir des données opérationnelles suffit si les tables concernées ont été suffisamment indexées.

2 : A-t-on vendu plus d'abonnement de magazines de sport cette année que l'année dernière ?

- ▶ Nécessite de garder toutes les dates de souscription, même pour les abonnements résiliés.
- ▶ Requêtes multidimensionnelles de type OLAP.

3 : Est-ce que les acheteurs de magazine de musique sont aussi amateurs de cinéma ?

- ▶ Exemple simplifié de problème où l'on demande si les données vérifient une règle.
- ▶ Réponse formulée par une valeur estimant la probabilité que la règle soit vraie.
- ▶ Utilisation d'outils statistiques.

4. Quelles sont les caractéristiques principales des lecteurs de magazine de cinéma ?

Question plus ouverte, il s'agit de trouver une règle et non plus de la vérifier ou de l'utiliser.

5 : Peut-on prévoir les pertes de client et prévoir des mesures pour les diminuer ?

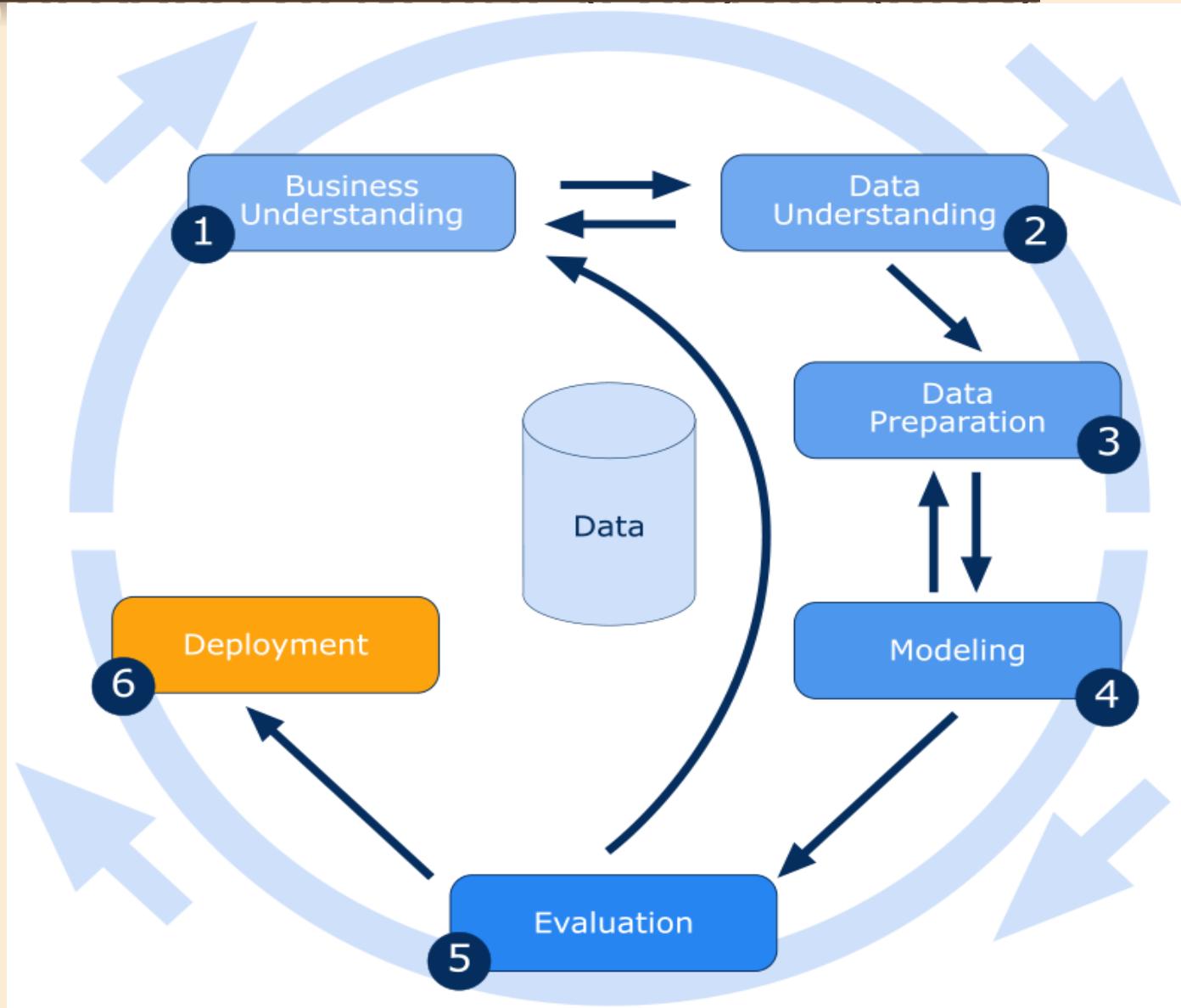
Question ouverte : il faut disposer d'indicateurs comme durée d'abonnement, délai de paiement, ...

C'est pour ce type de questions que sont mis en oeuvre les outils d'analyse et de fouille de données

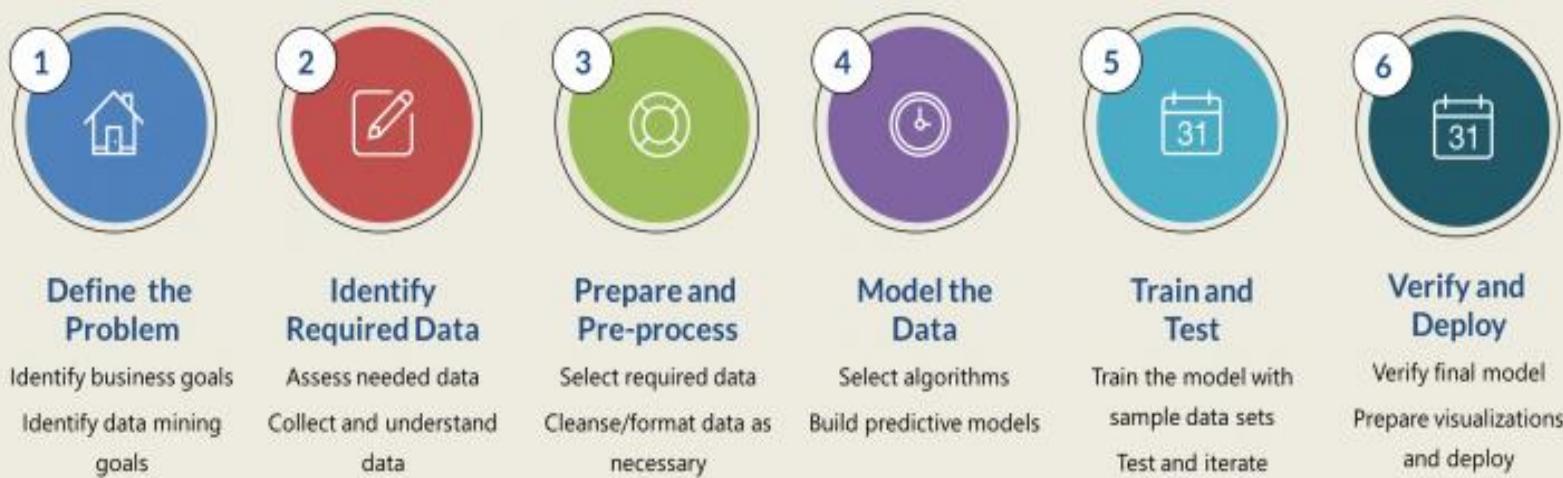
+++
+++

Méthodologie de découverte d'information ou de fouille de donnée

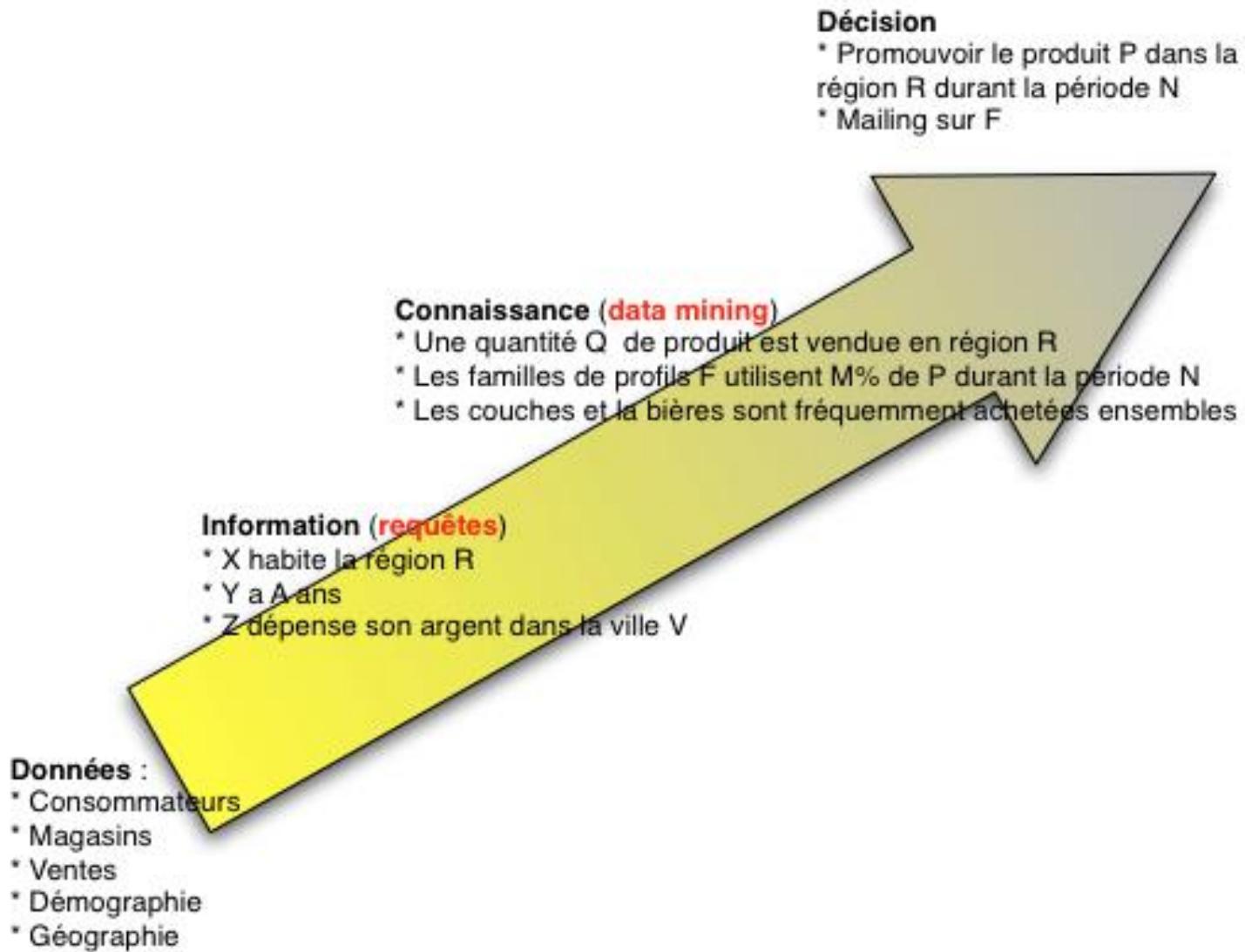
UNE DÉMARCHE PLUS QU'UNE THÉORIE : PROCESSUS GÉNÉRAL (ECD) OU (KDD)



PROCESSUS GÉNÉRAL ECD



DONNÉES, INFORMATION, CONNAISSANCE



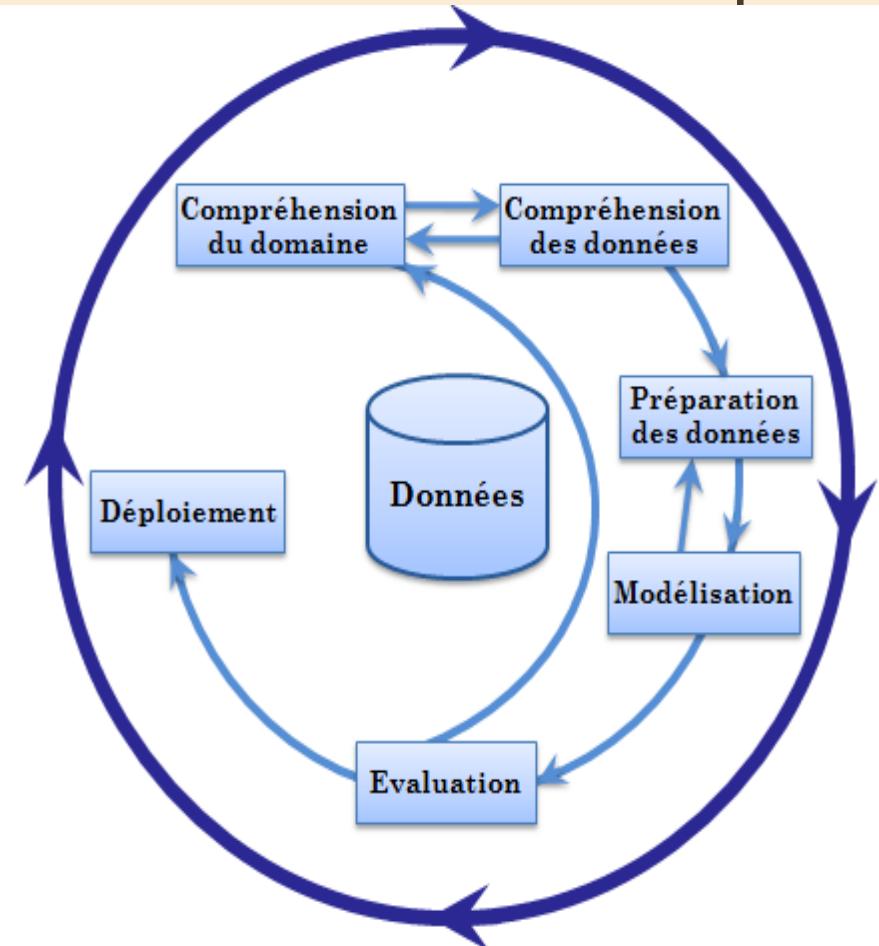
STANDARD DE RÉFÉRENCE

ETAPES DE L'ECD : CRISP-DM

- ✖ *Cross-Industry Standard Process For fouille de donnée*
- ✖ Conçu en 1996 par Daimler, SPSS et NCR.
- ✖ Conçu pour être indépendant des outils et du domaine d'application.
- ✖ Doit sa réussite au fait qu'il se base sur des expériences réelles de fouille de donnée.

ETAPES DE L'ECD : CRISP-DM

✖ CRISP-DM : un processus de références



PROCESSUS du DATA MINING		
Acteurs	Étapes	Phases
Maître d'œuvre	Objectifs	1 : Compréhension du métier
	Données	2 : Compréhension des données
	Traitements	3 : Préparation des données
Maître d'ouvrage	Modélisation	4 : Modélisation
	Evaluation	5 : Évaluation de la modélisation
Maître d'ouvrage	Déploiement	6 : Déploiement des résultats de l'étude

DIFFICULTÉS TECHNIQUES DU FOUILLE DE DONNÉE

1ère difficulté : comprendre les données : du bon sens !

La fouille de donnée travaille sur des tableaux de données. La première difficulté est de comprendre ces tableaux. Tant que les données ne sont pas comprises, on ne peut rien faire !

2ème difficulté : les statistiques, l'analyse de données

La fouille de donnée utilise les notions statistiques avec leurs difficultés propres. Toutefois, cette difficulté se résout partiellement si la première difficulté est correctement résolue.

3ème difficulté : algorithmique

Il faut comprendre un minimum des algorithmes spécifiques du fouille de donnée pour comprendre les principes, les usage et les limites du fouille de donnée.

4ème difficulté : utilisation d'un logiciel

En plus de connaître les principes généraux du fouille de donnée, il faut apprendre à se servir d'un logiciel particulier.

ÉTAPE DU PROCESSUS DE ECD

1. Connaître le domaine d'application (Connaissance du buts de l'application)
2. Sélection des données cibles (Analyse Exploratoire des Données)
3. Prétraitement des données
 1. Data cleaning
 2. Outlier detection
 3. Umbalanced data processing
 4. Réduction et transformation de données
 5. Normalisation si nécessaire
 6. Choix des algorithmes de fouille
4. fouille de donnée (Recherche des modèles intéressants)
5. Evaluation des patterns et présentation de la connaissance : Visualisation, transformation, etc.
6. Utilisation de la connaissance

TYPOLOGIE DES MÉTHODES DE FOUILLE DE DONNÉE

Quelle méthode utiliser, est définie par rapport :
aux objectifs de l'étude ?
aux données disponibles ?

LES TACHES DU FOUILLE DE DONNÉE

Contrairement aux idées reçues, La fouille de donnée n'est pas le remède miracle capable de résoudre toutes les difficultés ou besoins de l'entreprise.

Cependant, une multitude de problèmes d'ordre intellectuel, économique ou commercial peuvent être regroupés, dans leur formalisation, dans l'une des tâches suivantes :

LES TÂCHES DU FOUILLE DE DONNÉE

- ✖ La description
- ✖ La classification
- ✖ L'estimation
- ✖ Les règles d'association (Découverte de motifs séquentiels)
- ✖ La segmentation (clustering).
- ✖ La prévision

LA DESCRIPTION

Parfois le but de la fouille est simplement de décrire ce qui se passe sur une Base de Données compliquée en expliquant les relations existantes dans les données pour, dans un premier lieu comprendre le mieux possible les individus, les produits et les processus présents dans cette base.

Une bonne description d'un comportement implique souvent une bonne explication de celui-ci.

C'est souvent l'une des premières tâches demandées à un outil de fouille de donnée.

- ✖ Statistique descriptif
- ✖ ACP, Analyse factorielle

LA CLASSIFICATION

“ La classification consiste à examiner des caractéristiques d'un élément nouvellement présenté afin de l'affecter à une classe d'un ensemble prédéfini. ” [BERRY97]

Dans le cadre informatique, les éléments sont représentés par un enregistrement et le résultat de la classification viendra alimenter un champ supplémentaire par exemple : homme / femme, oui / non, rouge / vert / bleu, ...

Les techniques les plus appropriées à la classification sont :

- les arbres de décision, RF, NB...
-

L'ESTIMATION

L'estimation est similaire à la classification à part que la variable de sortie est numérique plutôt que catégorique.

Le résultat d'une estimation permet de procéder aux classifications grâce à un barème.

Cette technique sera souvent utilisée en marketing, combinée à d'autres, pour proposer des offres aux meilleurs clients potentiels.

La technique la plus appropriée à l'estimation est : **les réseaux de neurones**.

ESTIMATION : EXEMPLE

Par exemple on cherche à estimer la lecture de tension systolique d'un patient dans un hôpital, en se basant sur l'âge du patient, son genre, son indice de masse corporelle et le niveau de sodium dans son sang. La relation entre la tension systolique et les autres données vont fournir un modèle d'estimation. Et par la suite nous pouvons appliquer ce modèle dans d'autres cas.

L'ESTIMATION (RÉGRESSION)

- ✖ Prévoir la valeur d'une variable continue donnée en fonction des valeurs d'autres variables, en supposant un modèle de dépendance linéaire ou non linéaire
- ✖ Études approfondies dans les domaines de la statistique et des réseaux neuronaux
- ✖ Exemples
 - Prévision des numéros de vente d'un nouveau produit sur la base des dépenses publicitaires
 - Prévision des vitesses du vent en fonction de la température, de l'humidité, de la pression atmosphérique, etc.
 - Prévision des séries chronologiques des indices boursiers

RÈGLES D'ASSOCIATION

Corrélations (ou relations) entre attributs

- ✖ Applications : grande distribution, gestion des stocks, web(pages visitées), etc.
- ✖ Exemple
 - + BD commerciale : panier de la ménagère
 - + Articles figurant dans le même ticket de caisse
 - + Ex: achat de riz + vin blanc ==> achat de poisson

RÈGLE D'ASSOCIATION : MOTIF SÉQUENTIEL

Liaisons entre événements sur une période de temps

- ✖ Extension des règles d'association
 - + Prise en compte du temps (série temporelle)
 - + Achat Télévision ==> Achat Magnétoscope d'ici 5 ans
- ✖ Applications : marketing direct (anticipation des commandes), bioinformatique (séquences d'ADN), bourse (prédiction des valeurs des actions).

Ex: 60% des consommateurs qui commandent la bière commandent de l'aspro juste après

L'ANALYSE DES CLUSTERS

Partitionnement logique de la base de données en clusters (technique : **Clustering**)

- ✖ Clusters : groupes d'instances ayant les mêmes caractéristiques
- ✖ Applications : Economie (segmentation de marchés), médecine (localisation de tumeurs dans le cerveau), etc.

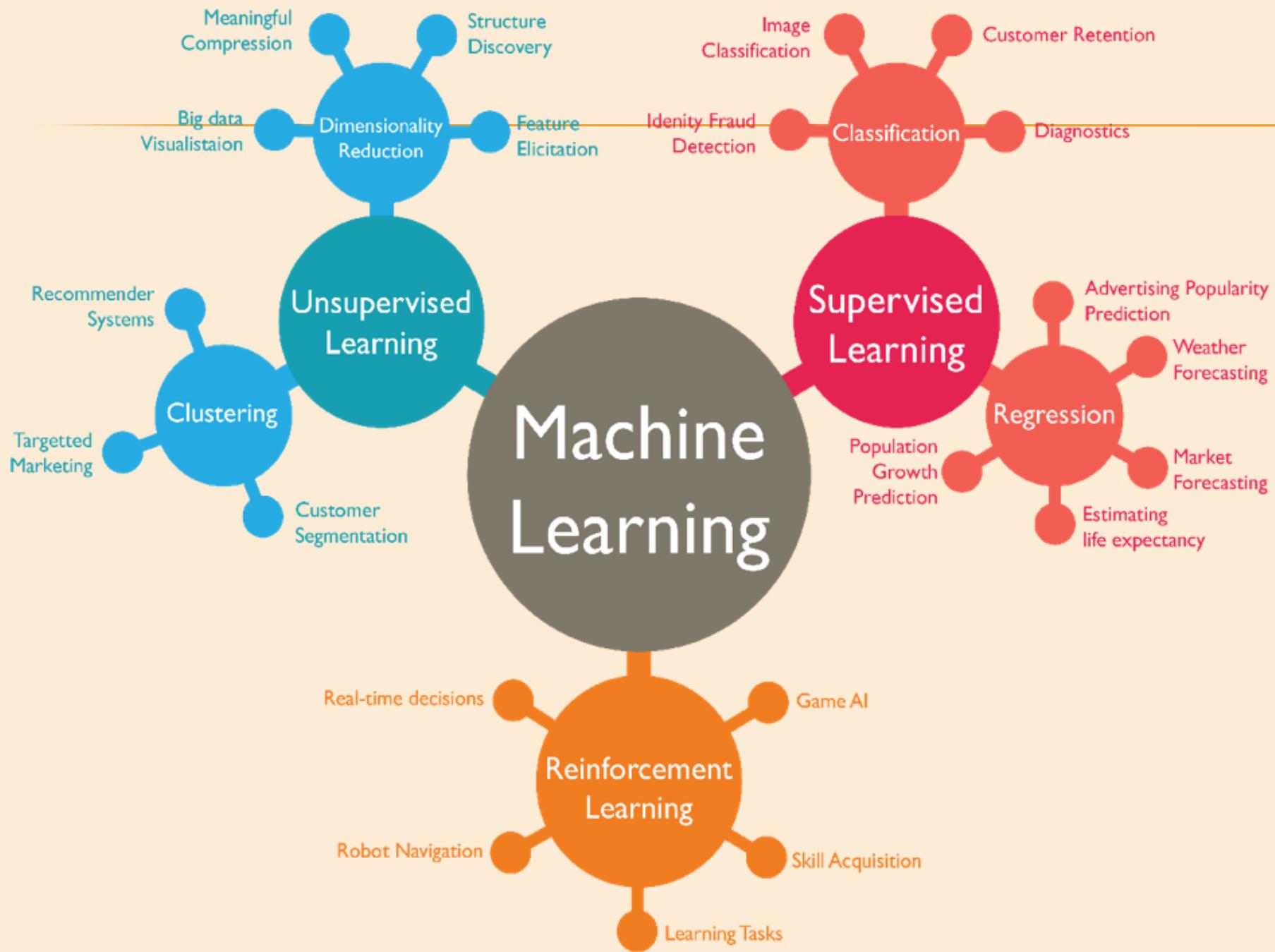
LA PREVISION

- ✖ La prévision est similaire à l'estimation mise à part que pour la prévision, les résultats portent sur le futur.
- ✖ Exemples
 - + Prévoir le temps qu'il va faire
 - + Prévoir le gagnant du championnat de football par rapport à une comparaison des résultats des équipes
 - +

D'autres concepts

LES TYPES D'APPRENTISSAGE AUTOMATIQUE

- + Apprentissage supervisé (supervised learning)
- + Non supervisé (unsupervised learning)
- + Semi supervisé (semi-supervised learning)
- + Par renforcement(reinforcement learning)

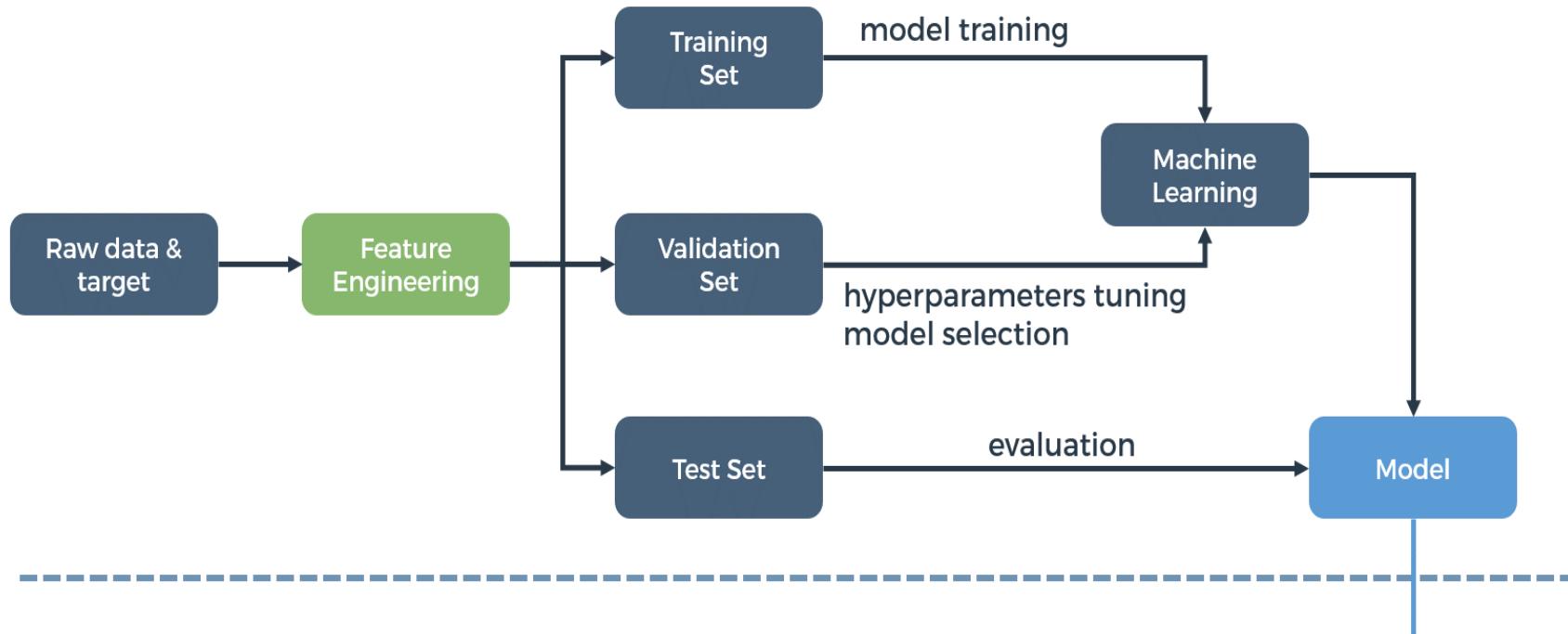


Apprentissage supervisé (supervised learning)

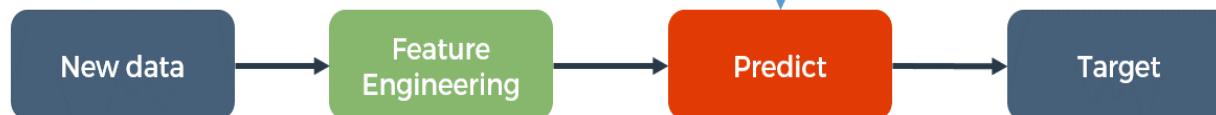
Des données (entrées) annotées de leurs sorties pour entraîner le modèle, c'est-à-dire que à chaque un **entrée** est associée à une classe **cible (sortie)**, une fois entraîné, le modèle (l'algorithme de ML) devient capable de prédire (éventuellement avec un pourcentage d'erreur) la cible sur de nouvelles données non annotées.

APPRENTISSAGE SUPÉRVISÉ

TRAINING



PREDICTING



TYPES

✖ Apprentissage supervisé :

+ **Régression** : est le type d'apprentissage dans lequel des données étiquetées sont utilisées, et ces données sont utilisées pour effectuer des prédictions sous **une forme continue**.

La régression est une technique de modélisation prédictive qui étudie la relation entre une variable dépendante [Outputs] et une variable indépendante [Inputs]. La sortie de l'entrée est toujours un graphe (linéaire en générale).

Cette technique est utilisée pour la prévision du temps, la modélisation de séries chronologiques, l'optimisation de processus. Ex: - L'un des exemples de la technique de régression est la prédition du prix de la maison, où le prix de la maison sera déterminé à partir des entrées telles que le nombre de pièces, la localisation, la facilité de transport, l'âge de la maison, la superficie de la maison.

TYPES

✖ Apprentissage supervisé :

+ **Classification** : La classification est le type d'apprentissage supervisé dans lequel les données étiquetées sont utilisées pour effectuer des prédictions sous **une forme non continue**.

La sortie de l'information n'est pas toujours continue et le graphique n'est pas linéaire. Dans la technique de classification, l'algorithme apprend à partir des données saisies, puis utilise cet apprentissage pour classer une nouvelle observation. Cet ensemble de données peut simplement être bi-classe, ou multi-classe aussi.

Ex: - Un des exemples de problèmes de classification est de vérifier si le courrier électronique est du spam ou non du spam en entraînant l'algorithme pour différents mots ou e-mails de spam

Apprentissage supervisé (supervised learning)

Exemple:

Les données d'entrée représentent des images et la cible (ou *target* en anglais) représente la catégorie de photos.

Voiture



Oiseau



Chat



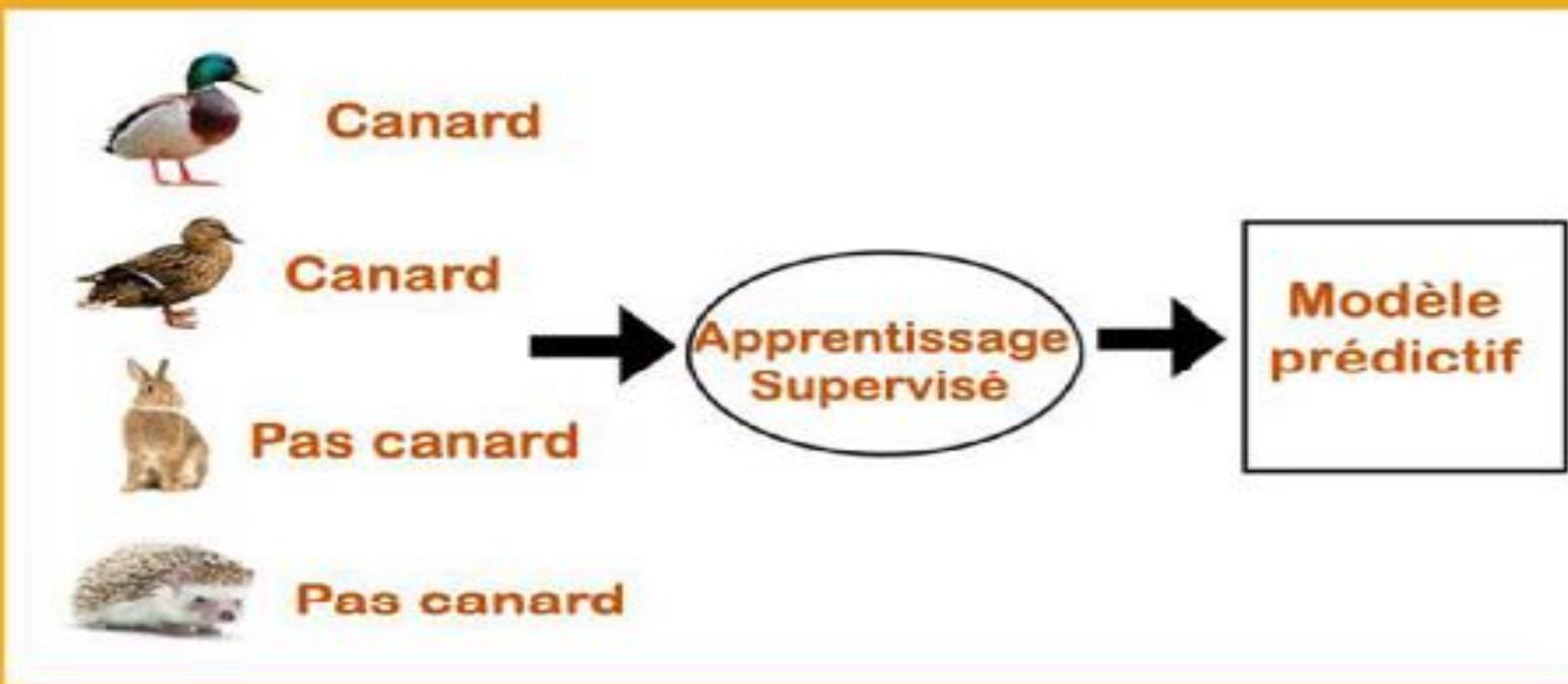
Chien



cheval



Apprentissage supervisé (supervised learning)



Apprentissage supervisé (supervised learning)

Techniques supervisées (prédictive)

Classification

**Régression
Estimation**

Arbre de décision

Réseaux de neurone

Naïve Bayes

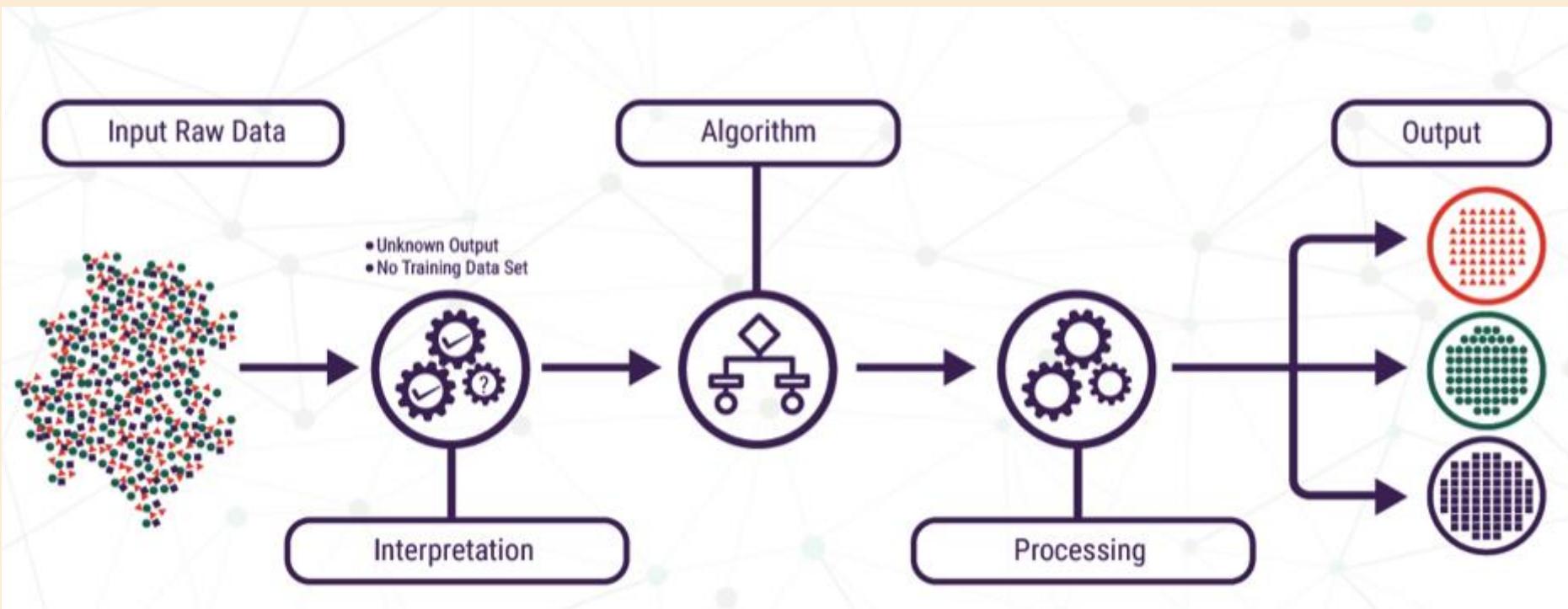
**Machines à
vecteurs supports
(SVM)**

K-Nearest Neighbour (K-NN)

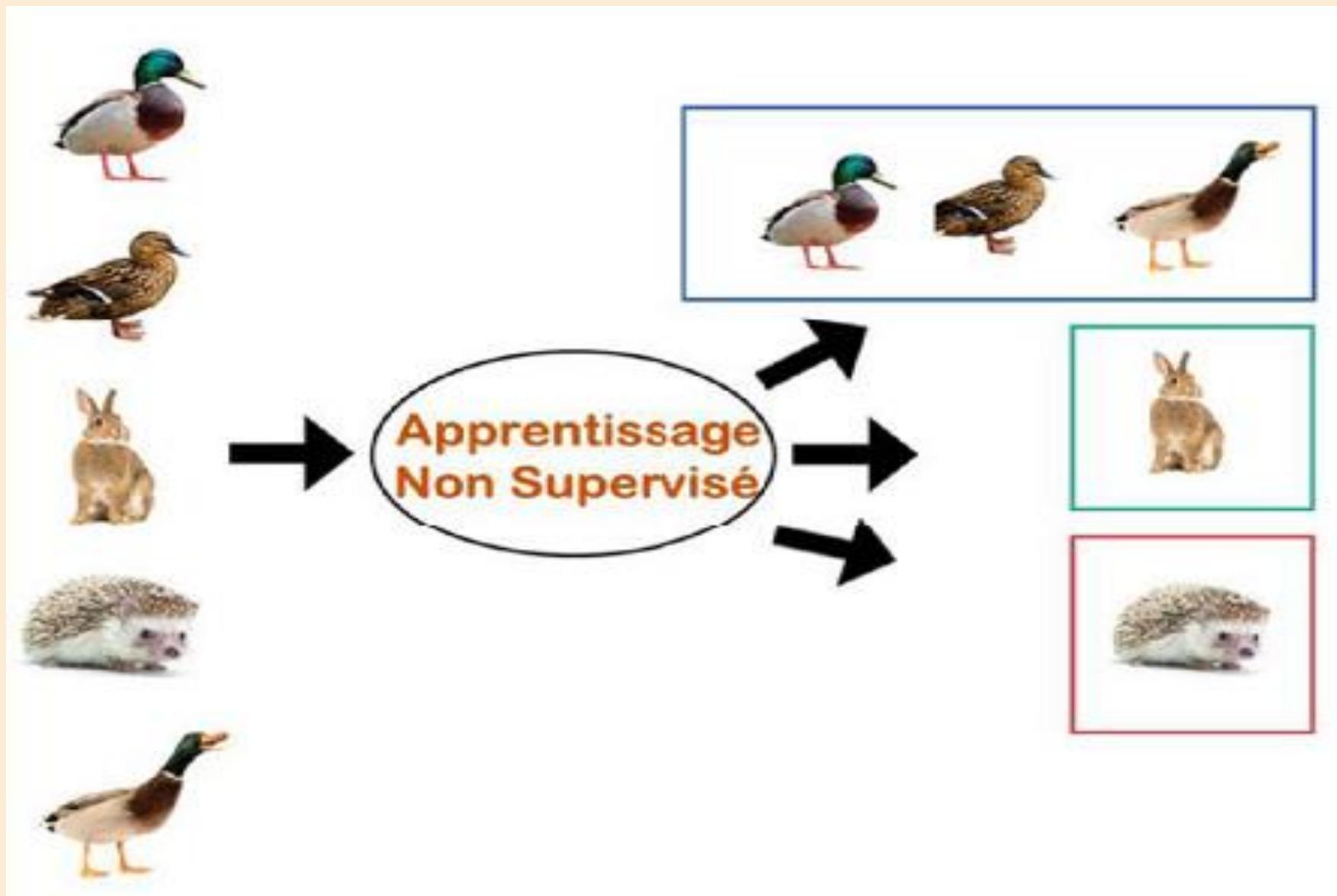
Apprentissage non supervisé (unsupervised learning)

En apprentissage non supervisé, les données d'entrées ne sont pas annotées. L'algorithme d'entraînement s'applique dans ce cas à trouver les similarités et distinctions au sein de ces données, et à regrouper ensemble celles qui partagent des caractéristiques communes. Dans notre exemple, les photos similaires seraient ainsi regroupées automatiquement au sein d'une même catégorie.

APPRENTISSAGE NON SUPÉRVISÉ



Apprentissage non supervisé (unsupervised learning)



TYPE

- ✖ Clustering : le processus de regroupement d'entités similaires. Les données groupées sont utilisées pour créer des clusters. Le but de cette technique est de rechercher des similitudes dans les données ainsi que de déterminer à quel groupe de nouvelles données doivent appartenir. Ex : k-means, hierarchical clustering, db-scan..
- ✖ Réduction de dimensionnalité : Cette technique est utilisée pour supprimer les caractéristiques indésirables dans les données. Elle concerne le processus de conversion d'un ensemble de données ayant de grandes dimensions en données comportant les mêmes informations et de petites tailles. Ces techniques sont utilisées lors de la résolution de problèmes d'apprentissage automatique pour obtenir de meilleures fonctionnalité. ACP, AFDM...
- ✖ Règle d'association : Les **règles d'association** sont des techniques de fouille de données utilisées pour découvrir des relations intéressantes entre des éléments dans de grandes bases de données. Elles sont souvent appliquées dans le **panier d'achat** pour identifier quels produits sont fréquemment achetés ensemble.

Apprentissage non supervisé (unsupervised learning)

Techniques non supervisées (descriptive)

Réduction de
dimensionalité

Clustering

Associations

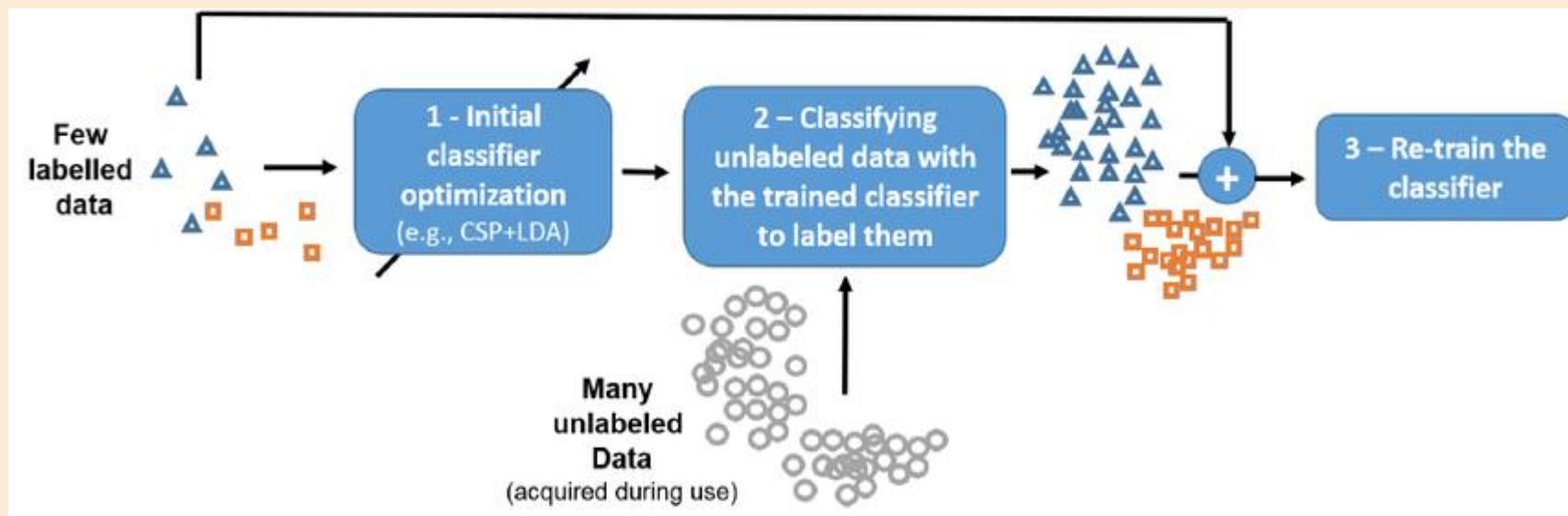
K-Means
K-moyennes

APRIORI
FP-GROUTH

Apprentissage semi-supervisé (semi-supervised learning)

- ✖ L'apprentissage semi-supervisé est une classe de techniques d'apprentissage automatique qui utilise un ensemble de données étiquetées et non étiquetées. Il se situe ainsi entre l'apprentissage supervisé qui n'utilise que des données étiquetées et l'apprentissage non supervisé qui n'utilise que des données non étiquetées. Il a été démontré que l'utilisation de données non étiquetées, en combinaison avec des données étiquetées, permet d'améliorer significativement la qualité de l'apprentissage.
- ✖ Un autre intérêt provient du fait que l'étiquetage de données nécessite souvent l'intervention d'un utilisateur humain. Lorsque les jeux de données deviennent très grands, cette opération peut s'avérer fastidieuse. Dans ce cas, l'apprentissage semi-supervisé, qui ne nécessite que quelques étiquettes, revêt un intérêt pratique évident.

APPRENTISSAGE SEMI-SUPÉRVISÉ



APPRENTISSAGE PAR RENFORCEMENT

L'apprentissage par renforcement est le quatrième type d'apprentissage automatique dans lequel aucune donnée brute n'est donnée en entrée, mais un algorithme d'apprentissage par renforcement doit comprendre la situation par lui-même.

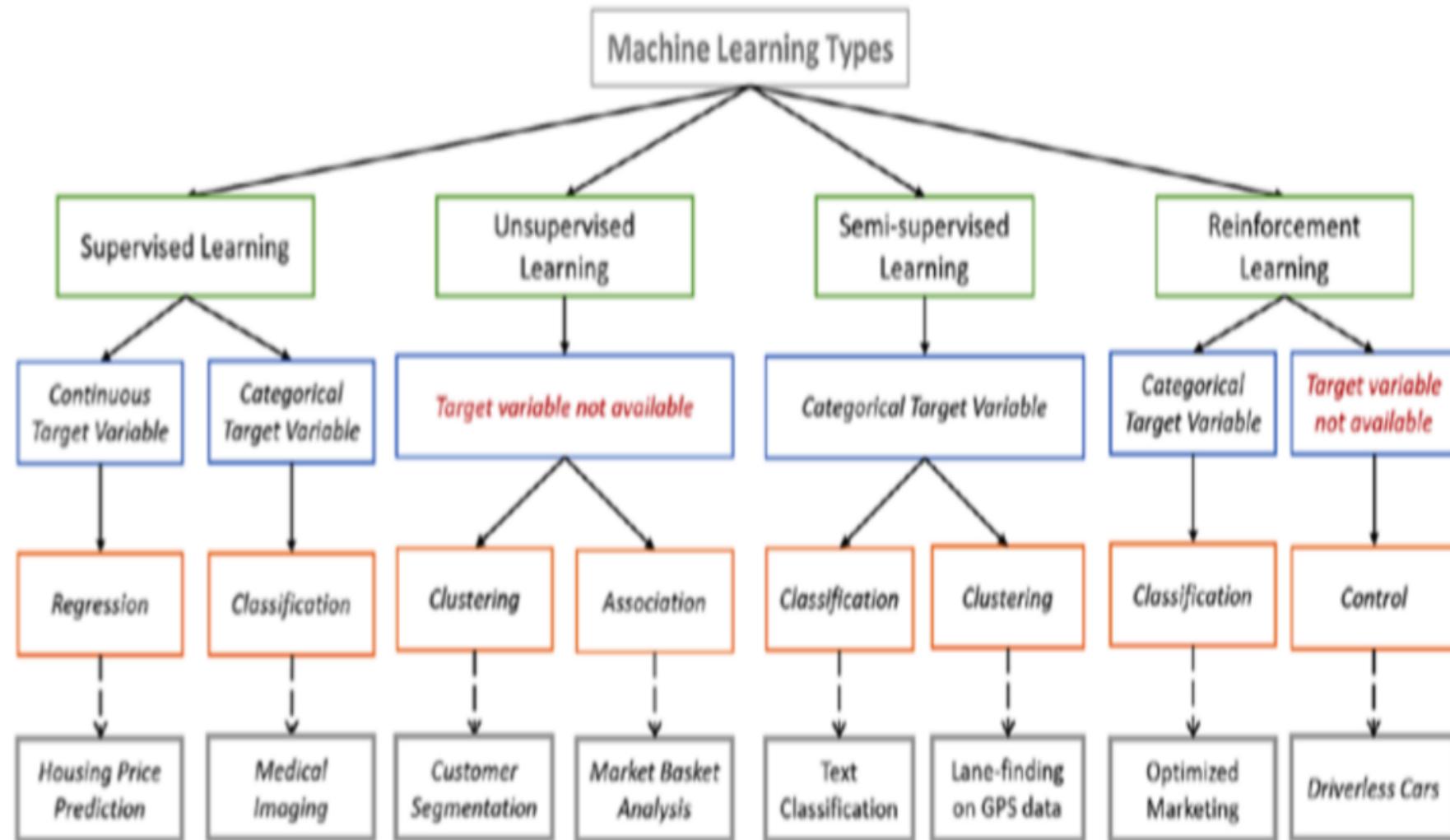
L'apprentissage par renforcement est fréquemment utilisé pour la robotique, les jeux et la navigation. Avec l'apprentissage par renforcement, l'algorithme découvre via les essais et les erreurs les actions qui génèrent les avantages les plus significatifs.

Ce type de formation comporte trois composantes principales: l'agent qui peut être décrit en tant qu'apprenant ou décideur, l'environnement qui décrit tout ce avec quoi l'agent interagit et les actions qui représentent ce que l'agent peut faire.

Ex : Jeu, Robotique...

APPRENTISSAGE PAR RENFORCEMENT





LES 7 ÉTAPES DE FOUILLE DE DONNEES

La fouille de données ne se résume pas à un ensemble d'algorithmes mais suit une succession d'étapes:

1) Compréhension du problème (Business Understanding) :

Définir l'objectif du projet.

Comprendre les besoins de l'entreprise ou de l'utilisateur.

Identifier les problèmes spécifiques que le projet de data mining doit résoudre.

2) L'acquisition de données :

l'algorithme se nourrissant des données en entrée, c'est une étape importante. Il en va de la réussite du projet, de récolter des données pertinentes et en quantité suffisante.

3) La préparation des données :

les données recueillies doivent être retouchées avant utilisation. En effet, certains attributs sont inutiles, d'autre doivent être modifiés afin d'être compris par l'algorithme, et certains éléments sont inutilisables car leurs données sont incomplètes.

LES 7 ÉTAPES DE FOUILLE DE DONNEES

4) Formation et apprentissage du modèle (Modeling)

5) L'évaluation : une fois l'algorithme d'apprentissage automatique entraîné sur un premier jeu de donnée, on l'évalue sur un deuxième ensemble de données afin de vérifier que le modèle ne fasse pas de surapprentissage.

6) Réglage des paramètres: Ajuster les paramètres pour de meilleures performances

7) Le déploiement : le modèle est déployé en production pour faire des prédictions, et potentiellement utiliser les nouvelles données en entrée pour se ré-entraîner et être amélioré.

LES DONNÉES

LES DONNÉES

- ✖ Une information brute, non traitée ou non interprétée et qui peut être enregistrée et stockée.
- ✖ Les données peuvent prendre de nombreuses formes, notamment des chiffres, des textes, des images, des sons, des vidéos, des mesures, des faits, etc.

Types de données :

1. Données Structurées :

- Les données structurées sont organisées dans un format tabulaire avec des lignes et des colonnes.
- Chaque colonne représente un attribut ou une caractéristique spécifique, tandis que chaque ligne correspond à une instance ou un enregistrement.
- Les bases de données relationnelles, les feuilles de calcul Excel et les fichiers CSV sont des exemples de données structurées.
- La fouille de donnée sur des données structurées est souvent effectué à l'aide de techniques de classification, de régression et de clustering.

Types de données :

2.Données Semi-Structurées :

- Les données semi-structurées ne suivent pas un modèle tabulaire strict, mais elles ont une certaine structure ou organisation.
- Elles sont souvent stockées au format XML, JSON, HTML, ou sous forme de documents textuels avec une structure interne.
- Les données semi-structurées sont couramment utilisées dans le web mining, l'extraction d'informations à partir de pages web, et l'analyse de texte.

Types de données :

3. Données Non Structurées :

- Ne suivent aucune structure spécifique
- Elles incluent des données textuelles, des images, des vidéos, des fichiers audio, des courriels, etc.
- les plus difficiles à analyser car elles ne sont pas directement compatibles avec les algorithmes traditionnels.
- La fouille de donnée sur des données non structurées implique souvent l'utilisation de techniques d'apprentissage automatique de détection et de traitement du langage naturel (NLP).

CARACTÉRISTIQUES DES DONNÉES :

1. Dimensionnalité :

- La dimensionnalité fait référence au nombre d'attributs ou de caractéristiques dans un jeu de données.
- Les données à haute dimensionnalité peuvent poser des défis en termes de visualisation, de calcul et de réduction de dimension.
- La fouille de donnée sur des données à haute dimensionnalité peut nécessiter des techniques de sélection de caractéristiques ou de réduction de dimension pour extraire des informations significatives.

CARACTÉRISTIQUES DES DONNÉES :

2. Densité :

- La densité des données mesure la proportion d'informations pertinentes par rapport à la taille totale du jeu de données.
- Les données denses contiennent généralement peu de valeurs manquantes ou de valeurs nulles, tandis que les données clairsemées ont de nombreuses valeurs manquantes ou nulles.

CARACTÉRISTIQUES DES DONNÉES :

3. Sparsité :

- La sparsité des données indique le degré de distribution inégale des valeurs des attributs.
- Dans des données très dispersées, la plupart des attributs auront des valeurs proches de zéro, sauf quelques-uns avec des valeurs significatives.
- La sparsité peut affecter la performance des algorithmes de classification et de clustering, et des techniques spécifiques sont souvent utilisées pour y remédier.

LES DONNÉES (TABLEAU)

Les données peuvent être vues comme une collection d'individu (d'objets, enregistrements) et leurs attributs.

- ▶ Un attribut est une propriété et ou une caractéristique d'un individu.
- ▶ Un ensemble d'attributs décrit un individu (ou un objet).

Objet
Ou
individu

Attributes				
ID#	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	90K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

ATTRIBUT

- ▶ Attribut : un champ de données, représentant une caractéristique ou fonctionnalité d'un objet de données.

Par exemple, _ID client, nom, adresse

- ▶ types:

nominal
binaire

ordinal

Numérique:

Discret

Continue

Intervalle-échellonné

Ratio-échellonné

TYPE DES ATTRIBUTS NOMINAUX

- ▶ Nominaux: catégories, états, ou des «noms de choses»
couleur de cheveux = {noir, blond, brun, gris, rouge, blanc}
la profession, les numéros d'identification, codes postaux
- ▶ Binaire
 - Attribut nominal avec seulement 2 états (0 et 1)
 - Binaire symétrique: les deux résultats tout aussi importants par exemple, le sexe
 - Binaire asymétrique: les deux résultats n'ont pas la même importance.
par exemple, test médical (positif vs négatif)
 - Convention: 1 à attribuer au résultat le plus important (par exemple, le VIH positif)
- ▶ Ordinal
 - Les valeurs ont un ordre significatif (classement) mais d'amplitude entre les valeurs successives est pas connue.
 - Taille = {petites, moyennes, grandes}, les grades, les mentions

TYPES DES ATTRIBUTS NUMÉRIQUES

- ▶ Nombre (entier ou valeur réelle), discret ou continu

- ▶ Intervalle

Les données d'intervalle sont des données continues où les différences entre les valeurs ont un sens, mais il n'y a pas de zéro absolu (zéro arbitraire). Cela signifie qu'un "zéro" dans ce type de données ne représente pas l'absence totale de la caractéristique mesurée.

Mesuré sur une échelle des unités de taille égale

Les valeurs ont un ordre

Exemple : La température en degrés Celsius ou Fahrenheit. La différence entre 30 °C et 40 °C est la même que la différence entre 70 °C et 80 °C, soit 10 °C. Mais un zéro en Celsius ou Fahrenheit ne représente pas l'absence de température (il y a toujours une température, même à zéro).

TYPES DES ATTRIBUTS NUMÉRIQUES

Ratio

Les données de type ratio sont également des données continues, mais dans ce cas, elles ont un zéro absolu, ce qui signifie que zéro représente l'absence totale de la caractéristique mesurée. Cela permet de faire des opérations de multiplication et de division, ce qui n'est pas possible avec les données d'intervalle.

Exemple : Le poids, la taille, le revenu, la distance, le temps.

Par exemple, un poids de 0 kg signifie qu'il n'y a pas de poids, et un poids de 10 kg est le double d'un poids de 5 kg. Les ratios ont donc un sens : on peut dire que quelqu'un qui pèse 10 kg pèse deux fois plus lourd qu'une personne de 5 kg.

ATTRIBUTS TEMPORELLES

Ce sont des données qui évoluent dans le temps, comme les séries chronologiques.

Exemple : les données de la bourse, les températures mesurées chaque jour.

Extraction des connaissance

**STATISTIQUE ET FOUILLE DE
DONNÉE**

On pourrait croire que les techniques de fouille de donnée viennent pour remplacer les statistiques. En fait, ils s'intègrent de plus en plus et il sont omniprésente dans plusieurs étapes. On les utilise :

- + pour faire une analyse préalable,
- + pour estimer ou alimenter les valeurs manquantes,
- + pendant le processus pour évaluer la qualité des estimations,
- + après le processus pour mesurer les actions entreprises et faire un bilan.

Par ailleurs, certaines techniques statistiques récentes (Analyse en composantes principales, analyse factorielle des correspondances, ...) peuvent être apparentées aux techniques de fouille de donnée.

Statistiques et fouille de donnée sont tout à fait complémentaires.

STATISTIQUES DESCRIPTIFS (RAPPEL)

1. Mesures de tendance centrale :

- ✖ Ces mesures permettent de décrire la "position centrale" des données, c'est-à-dire où se situe la plupart des valeurs.
- ✖ **Moyenne (ou moyenne arithmétique)** : La somme de toutes les valeurs divisée par le nombre total de valeurs.
 - ✖ $\text{Moyenne} = \frac{\sum x_i}{n}$
 - ✖ Exemple : Moyenne des salaires dans une entreprise.
- ✖ **Médiane** : La valeur centrale lorsque les données sont triées par ordre croissant ou décroissant. Si le nombre d'observations est impair, c'est la valeur du milieu, sinon c'est la moyenne des deux valeurs du milieu.
 - ✖ Exemple : Médiane de la taille des étudiants dans une classe.
- ✖ **Mode** : La valeur la plus fréquente dans un jeu de données.
 - ✖ Exemple : Le mode des couleurs préférées dans un sondage.

STATISTIQUES DESCRIPTIFS (RAPPEL)

2. Mesures de dispersion (ou de variabilité) :

Ces mesures donnent une idée de la dispersion ou de l'étendue des valeurs dans un jeu de données. **Étendue (Range)** : La différence entre la valeur maximale et la valeur minimale dans les données.

- ✖ **Etendue=Valeur maximale–Valeur minimale**
 - + Exemple : L'étendue des températures en été dans une région.
- ✖ **Variance** : La moyenne des carrés des écarts par rapport à la moyenne. Elle donne une mesure de la dispersion des données.
où x_i sont les valeurs et μ est la moyenne.
- ✖ **Écart-type** : La racine carrée de la variance, il permet de donner la dispersion des données dans les mêmes unités que les données elles-mêmes.
- ✖ **Coefficient de variation** : Le rapport de l'écart-type à la moyenne, souvent utilisé pour comparer la dispersion relative de deux distributions différentes.

$$\text{Variance} = \frac{\sum(x_i - \mu)^2}{n}$$

$$\text{Écart-type} = \sqrt{\text{Variance}}$$

où σ est l'écart-type et μ est la moyenne.

$$CV = \frac{\sigma}{\mu}$$

STATISTIQUES DESCRIPTIFS (RAPPEL)

3. Mesures de forme :

Ces mesures donnent des informations sur la symétrie et la forme de la distribution des données.

- ✖ **Skewness (asymétrie)** : Une mesure de l'asymétrie de la distribution des données. Si la distribution est symétrique, la skewness sera proche de zéro. Une skewness positive indique une asymétrie vers la droite, tandis qu'une skewness négative indique une asymétrie vers la gauche.
 - + Exemple : La skewness des revenus dans une population, souvent positive car la majorité des gens gagnent des revenus plus bas et quelques-uns gagnent beaucoup plus.

STATISTIQUES DESCRIPTIFS (RAPPEL)

4. Mesures de position :

Ces mesures permettent de décrire la position d'un certain pourcentage de données.

- ✖ **Quartiles** : Les quartiles divisent les données en quatre parties égales. Les trois quartiles principaux sont :
 - + **Q1 (premier quartile)** : 25 % des données sont inférieures à cette valeur.
 - + **Q2 (deuxième quartile)** : correspond à la médiane (50 % des données sont inférieures ou égales à cette valeur).
 - + **Q3 (troisième quartile)** : 75 % des données sont inférieures à cette valeur.
- ✖ **Percentiles** : Divisent les données en 100 parties égales. Par exemple, le 90e percentile est la valeur en dessous de laquelle 90 % des données se situent.

STATISTIQUES DESCRIPTIFS (RAPPEL)

- Dans le cas de deux variables ou plus on s'intéresse à la corrélation, au rapport de corrélation ou encore au test du khi 2 associé à une table de contingence.

Ces notions sont associées à différents graphiques comme le nuage de points (scatterplot), les diagrammes-boîtes parallèles,...
Ou plus encore la matrice de corrélation

EXTRACTION DES CONNAISSANCE

2 ère partie : Le prétraitement

DATA PREPROCESSING

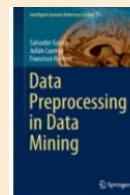
“Data preprocessing includes data preparation, compounded by integration, cleaning, normalization and transformation of data; and data reduction tasks; such as feature selection, instance selection, discretization, etc.

...

The result expected after a reliable chaining of data preprocessing tasks is a final dataset, which can be considered correct and useful for further data mining algorithms.”

We refer to data preparation as the set of techniques that initialize the data properly to serve as input for a certain DM algorithm.

Data reduction comprises the set of techniques that, in one way or another, obtain a reduced representation of the original data.



S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015

DATA PREPROCESSING

- ✖ Sampling
- ✖ Data preparation (cleaning, integration)
- ✖ Data transformation (transformation, normalization)
- ✖ Unbalanced data : Downsampling , Upweighting.....
- ✖ Feature engineering (...dimentionality reduction)
- ✖ Spliting

QUALITÉ D'UN DATA SET

la clé d'un bon modèle de Machine Learning est la qualité des données utilisées pour l'entraîner.

Toutefois, lors de la collecte de données, il est utile d'avoir une définition plus concrète de la qualité. De façon générale, un bon data set doit répondre à 3 contraintes qui sont:

Fiabilité (ou validité)

Représentation des caractéristiques

Réduction du biais

FIABILITÉ

- Un modèle formé à partir d'un ensemble de données fiables (degré auquel on peut faire confiance aux données) est plus susceptible de produire des prédictions utiles qu'un modèle formé à partir de données non fiables. La mesure de fiabilité permet d'avoir une visibilité sur les performances. Pour y arriver, il faut savoir :

1. Quelle est la fréquence des erreurs d'étiquetage?
2. Vos caractéristiques sont-elles bruyantes?

On ne peut jamais avoir un ensemble de données sans bruit. Un peu de bruit est acceptable.

3. Les données sont-elles correctement filtrées pour le problème?

Par exemple,

Requêtes de recherche de robots pour la détection de spam mais pas dans d'autre contexte

FIABILITÉ

Qu'est ce qui rend un data set non fiable

- + Validité
- + Précision
- + Complétude
- + Cohérence

FIABILITÉ

Validité

- ✖ ***Les types de données*** : les valeurs d'une colonne doivent être d'un type de données particulier, par exemple, numérique, date, etc.
- ✖ ***Contraintes de plage***: par exemple, les nombres doivent être compris dans une plage donnée.
- ✖ ***Contraintes obligatoires*** : par exemple certaines colonnes ne peuvent pas être vides.
- ✖ ***Unicité***: un champ ou plusieurs champs combinés doit être unique dans un data set.
- ✖ ***bonne étiquettes***

FIABILITÉ

Précision

Cette tâche n'est clairement pas simple. Car définir toutes les valeurs valides possibles permet de repérer facilement les valeurs non valides, cela ne signifie pas pour autant qu'elles sont exactes et encore moins qu'elles sont précises.

La différence entre exactitude et validité : Par exemple, dire que vous vivez en Afrique est, certes, vrai. Cependant, cette réponse n'est pas **précise**.

FIABILITÉ

Complétude

- ✖ *Présence des valeurs manquantes*

Cohérence

- ✖ *Données en double* : Par exemple, un serveur a téléchargé deux fois par erreur les mêmes journaux.
- ✖ *Donnée intentionnelles* : Par exemple, quelqu'un a tapé un chiffre supplémentaire ou un thermomètre a été laissé au soleil.
- ✖ *Données incompatibles* : âge et date de naissance



RÉDUCTION DU BIAIS

- le biais peut être exprimé comme une « distance » entre le meilleur modèle pouvant être appris par l'algorithme et le vrai modèle. En machine learning, on cherche, en général, un équilibre entre biais et variance, de telle sorte que ces deux erreurs soient à peu près égales.
- La nature de l'erreur dépend du type de problème considéré. Par exemple, dans un problème de classification d'images, l'erreur pourra être « le % de fois où le modèle se trompe en choisissant les classes » ; dans le cadre d'un problème de régression, le biais pourrait être une erreur des moindres carrés...

DATA SET SIZE

- ✖ Les modèles simples sur de grands ensembles de données sont généralement plus efficaces que les modèles sophistiqués sur de petits ensembles de données.
- ✖ Mais, Il est souvent difficile de collecter suffisamment de données pour un projet d'apprentissage automatique. Parfois, cependant, il y a trop de données, et vous devez sélectionner un sous-ensemble d'exemples pour l'apprentissage

SAMPLING DATA

Il est souvent difficile de collecter suffisamment de données pour un projet d'apprentissage automatique. Cependant, parfois il y a trop de données, et vous devez sélectionner un sous-ensemble d'exemples pour l'entraînement du modèle

En fin de compte, la réponse dépend du problème: que nous voulons prédire et quelles fonctionnalités voulons-nous?

SAMPLING DATA : TYPES

- ✖ **Échantillonnage aléatoire simple** : Il y a une probabilité égale de sélectionner un élément particulier
- ✖ **Échantillonnage sans remplacement** : Au fur et à mesure que chaque élément est sélectionné, il est retiré de la population
- ✖ **En échantillonnage avec remise**, le même objet peut être ramassé plusieurs fois
- ✖ **Échantillonnage stratifié** : Divisez les données en plusieurs partitions ; puis tirer des échantillons aléatoires de chaque cloison

DATA PREPROCESSING

L'acquisition des données

fait référence à la collecte des données nécessaires pour une analyse dans le but de résoudre un problème spécifique. Les données peuvent provenir de plusieurs sources, telles que :

- ✖ **Bases de données relationnelles** (SQL, NoSQL)
- ✖ **API** (pour accéder à des données en ligne)
- ✖ **Fichiers locaux** (CSV, Excel, JSON, etc.)
- ✖ **Web scraping** (pour collecter des données depuis des sites Web)
- ✖ **Capteurs et IoT** (Internet des objets)

L'acquisition doit garantir que les données collectées sont pertinentes, de bonne qualité, et adaptées aux objectifs de l'analyse. C'est une étape cruciale, car des données incorrectes ou de mauvaise qualité peuvent fausser toute l'analyse suivante.

L'acquisition des données (Chargement d'un dataset) en Python :

Les datasets sont souvent stockés sous divers formats tels que CSV, Excel, JSON, SQL, ou même des bases de données NoSQL. L'une des premières étapes d'un projet de Data Mining est de charger ces données en Python afin qu'elles puissent être explorées et manipulées. Le module `pandas` en Python est très populaire pour charger des fichiers .

- **CSV (Comma-Separated Values)**

```
import pandas as pd  
dataset = pd.read_csv('path_to_file.csv')
```

- **Excel** : (nécessite d'installer `openpyxl` ou `xlrd` pour lire les fichiers `.xlsx`).

```
dataset = pd.read_excel('path_to_file.xlsx')
```

- **JSON** : Pour les données en format JSON, vous pouvez utiliser la fonction `read_json` de `pandas` :

```
dataset = pd.read_json('path_to_file.json')
```

- **Bases de données SQL** : Vous pouvez charger des données directement depuis une base de données en utilisant `pandas` avec la fonction `read_sql`. Vous aurez besoin d'une connexion à la base de données, souvent via `sqlite3` ou un connecteur spécifique.

```
import sqlite3  
conn = sqlite3.connect('database.db')  
query = 'SELECT * FROM table_name'  
dataset = pd.read_sql(query, conn)
```

EXEMPLE SUR UNE ÉTUDE DE CAS

Tout au long de cette partie, nous travaillerons sur le jeu de données « Titanic » est un jeu de données populaire souvent utilisé à des fins éducatives et d'introduction à l'analyse de données. Il contient des informations sur les passagers du RMS Titanic, qui a coulé après avoir heurté un iceberg lors de son voyage inaugural en 1912.

EXEMPLE SUR UNE ÉTUDE DE CAS

L'ensemble de données comprend généralement diverses caractéristiques relatives aux passagers, telles que l'âge,

- + le sexe, la classe de billet, le tarif et le statut de survie.
- + PassengerId: Un identifiant unique pour chaque passager.
- + Survived: Indique si un passager a survécu (1) ou n'a pas survécu (0).
- + Pclass: La classe de billet du passager (1ère, 2ème ou 3ème classe).
- + Name: Le nom du passager.
- + Sex: Le sexe du passager.
- + Age: l'âge du passager.
- + Siblings/Spouses Aboard: Nombre de frères et sœurs ou de conjoints à bord du Titanic.
- + Parents/Children Aboard: Nombre de parents ou d'enfants à bord du Titanic.
- + Ticket: Le numéro du billet.
- + Fare: Le prix payé pour le billet.
- + Cabin: Le numéro de la cabine.
- + Embarked: Le port d'embarquement (C = Cherbourg, Q = Queenstown, S = Southampton)

CHARGEMENT ET EXPLORATION INITIALE

```
import pandas as pd
titanic_data = pd.read_csv('titanic.csv')
titanic_data
```

Out[1]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

VÉRIFICATION DE LA QUALITÉ DES DONNÉES APRÈS ACQUISITION :

Une fois que les données sont récupérées, il est important de vérifier leur qualité avant de passer à l'analyse. Cela inclut des étapes comme :

- **Vérification de la structure des données** : s'assurer que les colonnes et les types de données sont corrects.
- **Vérification des valeurs manquantes** : identifier et gérer les valeurs manquantes dans les données.
- **Exploration initiale** : examiner les premières lignes des données pour détecter les incohérences évidentes.

VÉRIFICATION DE LA STRUCTURE DES DONNÉES

```
python
```

```
# Vérifier les premières lignes des données  
print(dataset.head())
```

```
# Vérifier les types de données  
print(dataset.dtypes)
```

```
# Vérifier les valeurs manquantes  
print(dataset.isnull().sum())
```

ANALYSE DE LA STRUCTURE DES DONNÉES

- Nous examinons le nombre de lignes et de colonnes ainsi que les premières valeurs.

💡 Nombre de lignes et colonnes :

```
print("Nombre de lignes :", data.shape[0])
```

```
print("Nombre de colonnes :", data.shape[1])
```

1. Nombre de lignes et colonnes

```
1 data.shape
```

```
(891, 12)
```

```
1 print("Nombre de lignes :" ,df.shape[0])
2 print("Nombre de colonnes :" , df.shape[1])
```

```
Nombre de lignes : 891
```

```
Nombre de colonnes : 12
```

ANALYSE DE LA STRUCTURE DES DONNÉES

💡 Aperçu des premières lignes

```
# Afficher les 5 premières lignes  
print(df.head())
```

```
1 # Afficher Les 5 premières Lignes  
2 print(df.head())  
  
   PassengerId  Survived  Pclass \\\n0              1        0     3  
1              2        1     1  
2              3        1     3  
3              4        1     1  
4              5        0     3  
  
                                         Name  Sex  Age  SibSp \\\n0           Braund, Mr. Owen Harris  male  22.0      1  
1  Cumings, Mrs. John Bradley (Florence Briggs Th... female  38.0      1  
2                   Heikkinen, Miss. Laina  female  26.0      0  
3            Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35.0      1  
4           Allen, Mr. William Henry  male  35.0      0  
  
   Parch      Ticket      Fare Cabin Embarked  
0    0         A/5 21171    7.2500   NaN      S  
1    0          PC 17599   71.2833   C85      C  
2    0    STON/O2. 3101282   7.9250   NaN      S  
3    0         113803  53.1000  C123      S  
4    0         373450   8.0500   NaN      S
```

💡 Afficher les 5 dernières lignes

```
print(df.tail())
```

```
1 # Afficher Les 5 dernières Lignes  
2 print(df.tail())  
  
   PassengerId  Survived  Pclass \\\n886          887        0     2  
887          888        1     1  
888          889        0     3  
889          890        1     1  
890          891        0     3  
  
                                         Name  
886  Montvila, Rev. Juozas  
887  Graham, Miss. Margaret Edith  
888  Johnston, Miss. Catherine Helen "Carrie"  
889  Behr, Mr. Karl Howell  
890  Dooley, Mr. Patrick  
  
   Sex  Age  SibSp  Parch      Ticket      Fare Cabin Embarked  
886  male 27.0      0      0       211536  13.00   NaN      S  
887 female 19.0      0      0       112053  30.00   B42      S  
888  female  NaN      1      2       W./C. 6607  23.45   NaN      S  
889  male 26.0      0      0       111369  30.00  C148      C  
890  male 32.0      0      0       370376   7.75   NaN      Q
```

ANALYSE DE LA STRUCTURE DES DONNÉES

📌 Informations générales sur les colonnes:
print(data.info())

```
1 print(df.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   PassengerId 891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object  
 4   Sex          891 non-null    object  
 5   Age          714 non-null    float64 
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object  
 9   Fare          891 non-null    float64 
 10  Cabin        204 non-null    object  
 11  Embarked     889 non-null    object  
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

📌 Vérification des types de données :
print(data.dtypes)

```
1 print(data.dtypes)

PassengerId      int64
Survived        int64
Pclass          int64
Name            object
Sex             object
Age             float64
SibSp           int64
Parch           int64
Ticket          object
Fare            float64
Cabin          object
Embarked        object
dtype: object
```

ANALYSE DE LA STRUCTURE DES DONNÉES

❖ Statistiques descriptives: Les statistiques descriptives vous fournissent un aperçu des tendances centrales (moyenne, médiane), de la dispersion (écart-type, variance) et des valeurs extrêmes.

```
print(data.describe())
```

	# Résumé des colonnes numériques					
	PassengerId	Survived	Pclass	Age	SibSp	\
count	891.000000	891.000000	891.000000	714.000000	891.000000	
mean	446.000000	0.383838	2.308642	29.699118	0.523008	
std	257.353842	0.486592	0.836071	14.526497	1.102743	
min	1.000000	0.000000	1.000000	0.420000	0.000000	
25%	223.500000	0.000000	2.000000	20.125000	0.000000	
50%	446.000000	0.000000	3.000000	28.000000	0.000000	
75%	668.500000	1.000000	3.000000	38.000000	1.000000	
max	891.000000	1.000000	3.000000	80.000000	8.000000	
	Parch Fare					
count	891.000000	891.000000				
mean	0.381594	32.204208				
std	0.806057	49.693429				
min	0.000000	0.000000				
25%	0.000000	7.910400				
50%	0.000000	14.454200				
75%	0.000000	31.000000				
max	6.000000	512.329200				

QUALITÉ DES DONNÉES

Les problèmes les plus connus concernant les données sont :

- Valeurs manquantes
- Bruit et valeurs aberrantes
- Données en double
- Valeurs incohérentes
- Asymétrie des données
- Données déséquilibrées

NETTOYAGE DES DONNÉES : VALEURS MANQUANTES ET ABERRANTES

incomplètes

manque de valeurs d'attribut,
manque de certains attributs d'intérêt,
contenant des données agrégées seulement, par exemple,
Métier = "" (données manquantes)

- broyées: contenant du bruit, erreurs, ou aberrantes, par exemple, Salaire = "- 10" (une erreur)
- incompatibles: contenant des divergences dans les codes ou les noms, par exemple, Age = "42", Anniversaire = "03/07/2010" été-noté "1, 2, 3", maintenant Evaluation "A, B, C"
- Intentionnelle : entre les enregistrements en double (par exemple, les données manquantes déguisée)
1 janvier comme anniversaire de tout le monde?

NETTOYAGE DES DONNÉES : EXEMPLE

Id Client	CP	Sexe	Revenu	Âge	Marié	Montant
1001	75000	M	75000	C	M	5000
1002	4000	F	-40000	40	V	4000
1003	92100		1000000	45	C	7000
1004	6260	M	50000	0	C	1000
1005	29000	F	99999	30	D	3000

D'après Des données à la connaissance, de Daniel T. Larose, p. 26

Que peut-on constater dans ce tableau ?

- + CP : 4000 ? 6260 ?
- + Sexe : il y a un champ manquant.
- + Revenu : -40000 ? Négatif ! 1000000 ? C'est beaucoup. 99999 (c'est très précis), unité
- + Age : C ? 0 ?
- + Marié : que signifient les symboles ?
- + Montant : le problème de la monnaie.

COMMENT TRAITER LES DONNÉES MANQUANTES

Les solutions possibles sont :

- ✖ Ignorer les tuples : opération faite d'habitude quand l'étiquette de classe manque.
- ✖ Remplir les valeurs manquantes manuellement : opération ennuyeuse et infaisable
- ✖ Employer une constante globale pour remplir les valeurs manquantes : par exemple, "inconnu", ce qui engendre la création d'une nouvelle classe.
- ✖ Utiliser une valeur statistique pour remplir les valeurs manquantes.

COMMENT TRAITER LES DONNÉES MANQUANTES

Les solutions possibles sont :

- ✖ Les méthodes courantes pour les attributs numériques : Remplacer les valeurs manquantes par la moyenne, la médiane ou le mode de la colonne.
- ✖ Estimer ces valeurs manquantes par des méthodes d'induction comme la régression, les réseaux de neurones simples ou multicouches, ou les graphes d'induction.
- ✖ Pour des valeurs catégorielles : Remplissez avec la valeur la plus probable. Pour se faire, utilisez une technique d'inférence, par arbre décision
- ✖ Pour les données de séries temporelles : Remplissage avant et remplissage arrière : Utiliser la valeur précédente (forward fill) ou la valeur suivante (backward fill) pour compléter les valeurs manquantes.

DÉTECTION ET TRAITEMENT DES VALEURS MANQUANTES

- Les valeurs manquantes peuvent poser problème lors de l'entraînement des modèles. Certaines colonnes du dataset contiennent des valeurs manquantes.

💡 Détection des valeurs manquantes :

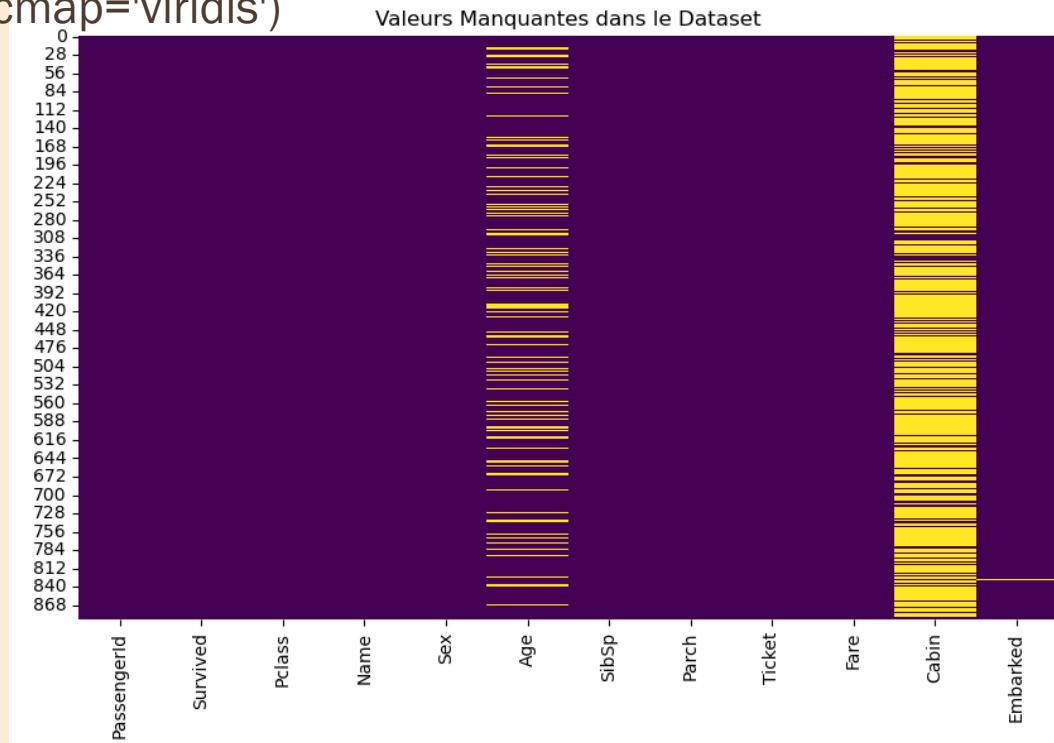
```
print(data.isnull().sum())
```

1	# Nombre de valeurs manquantes par colonne
2	print(data.isnull().sum())
	PassengerId 0
	Survived 0
	Pclass 0
	Name 0
	Sex 0
	Age 177
	SibSp 0
	Parch 0
	Ticket 0
	Fare 0
	Cabin 687
	Embarked 2
	dtype: int64

DÉTECTION ET TRAITEMENT DES VALEURS MANQUANTES

📌 Visualisation des valeurs manquantes :

```
import seaborn as sns  
import matplotlib.pyplot as plt  
plt.figure(figsize=(10,6))  
sns.heatmap(data.isnull(), cbar=False, cmap='viridis')  
plt.show()
```



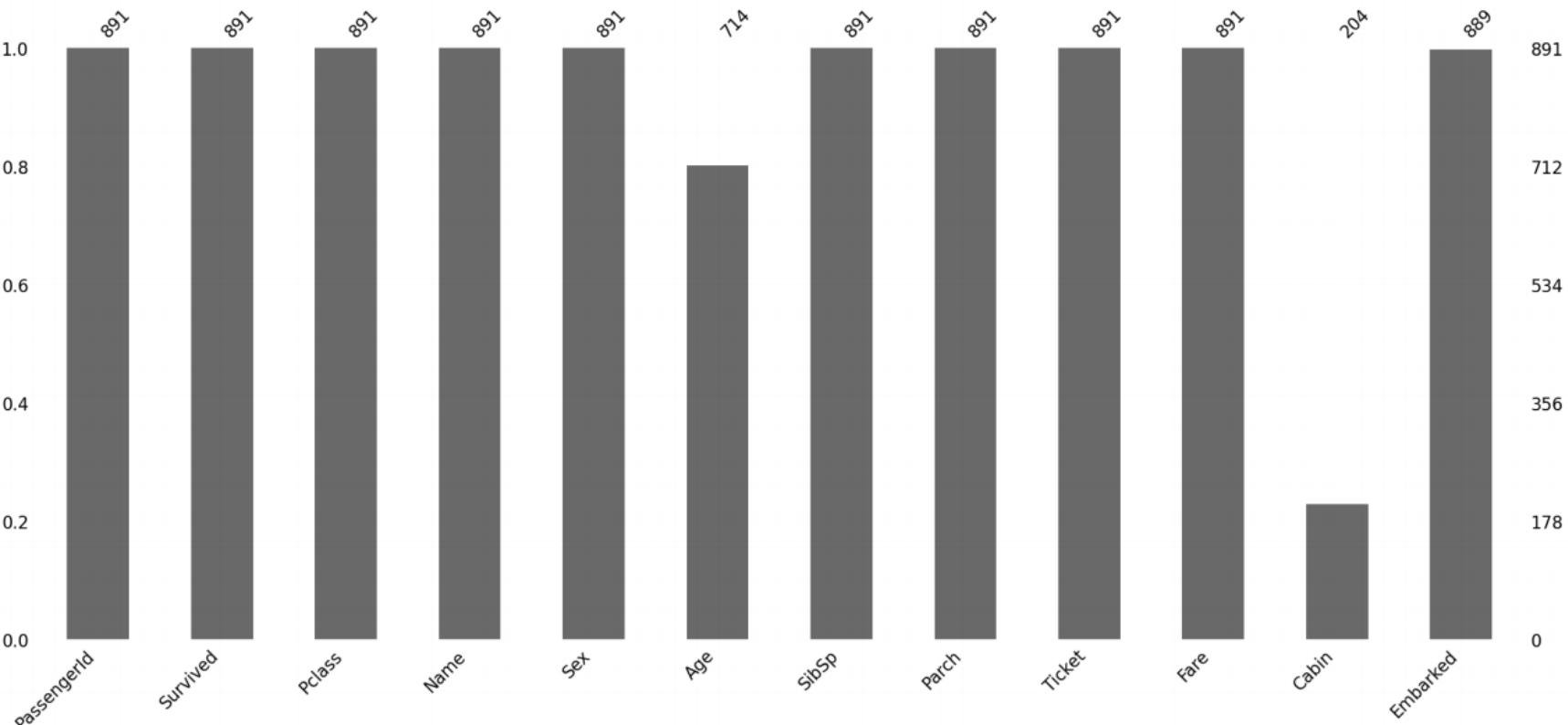
DÉTECTION ET TRAITEMENT DES VALEURS MANQUANTES

❖ Visualisation des valeurs manquantes :

```
import missingno as msno  
msno.bar(titanic_data)
```

Out[7]:

<AxesSubplot:



TRAITEMENT DES VALEURS MANQUANTES

Nous devons gérer les valeurs manquantes

☒ Suppression des colonnes inutiles :

```
data.drop(columns=["PassengerId", "Name", "Ticket", "Cabin"],  
inplace=True)
```

☒ Imputation des valeurs manquantes :

+ Définir une valeur constante de remplacement

```
("constant": SimpleImputer(strategy="constant",  
fill_value="Unknown"),)
```

+ Imputation statistique :

```
"mean": SimpleImputer(strategy="mean"),
```

```
"median": SimpleImputer(strategy="median"),
```

```
"most_frequent": SimpleImputer(strategy="most_frequent"),
```

DÉTECTION ET TRAITEMENT DES VALEURS MANQUANTES

Nous devons gérer les valeurs manquantes

❖ Remplacement des valeurs manquantes :

```
data["Age"].fillna(data["Age"].median(), inplace=True)
```

```
data["Embarked"].fillna(data["Embarked"].mode()[0], inplace=True)
```

```
1 # avant  
2 data["Age"]
```

```
0    22.0  
1    38.0  
2    26.0  
3    35.0  
4    35.0  
...  
886   27.0  
887   19.0  
888   NaN  
889   26.0  
890   32.0  
Name: Age, Length: 891, dtype: float64
```

```
1 # Remplacer les âges manquants par la médiane  
2 data["Age"].fillna(data["Age"].median(), inplace=True)
```

```
1 # après  
2 data["Age"]
```

```
0    22.0  
1    38.0  
2    26.0  
3    35.0  
4    35.0  
...  
886   27.0  
887   19.0  
888   28.0  
889   26.0  
890   32.0  
Name: Age, Length: 891, dtype: float64
```

DÉTECTION ET TRAITEMENT DES VALEURS MANQUANTES

💡 Remplacement des valeurs manquantes :

```
data["Embarked"].fillna(data["Embarked"].mode()[0],  
inplace=True)
```

```
1 print(data['Embarked'].unique())  
2 print(data['Embarked'].value_counts())
```

```
['S' 'C' 'Q' nan]  
S    644  
C    168  
Q     77  
Name: Embarked, dtype: int64
```

```
1 data["Embarked"].isnull().sum()
```

```
2
```

```
1 # Remplacer par la valeur la plus fréquente  
2 data["Embarked"].fillna(data["Embarked"].mode()[0], inplace=True)
```

```
1 data["Embarked"].isnull().sum()
```

```
0
```

TRAITEMENT DES VALEURS MANQUANTES

Remarques :

D'autre solutions possibles sont :

- ✖ Estimation à l'aide de techniques d'apprentissage automatique :
 - Imputation avec k-NN (KNN)
 - Imputation de régression
 - Imputation basée sur le clustering (K-Means)
 - Imputation basée sur RF
 -

Exemple python

```
"knn": KNNImputer(n_neighbors=5),  
"iterative_regression":IterativeImputer(estimator=RandomForestRegressor(n_estimators=10, random_state=0)),
```

```
import pandas as pd
import datawig

# Define a list of imputation techniques to test
imputation_techniques = {
    "SimpleImputer": datawig.SimpleImputer(input_columns=columns_to_impute,
output_column="Age"),
    "RegressionImputer": datawig.RegressionImputer(input_columns=columns_to_impute,
output_column="Age"),
    "ClassificationImputer": datawig.ClassificationImputer(input_columns=columns_to_impute,
output_column="Age"),
    "MLPImputer": datawig.MLPImputer(input_columns=columns_to_impute,
output_column="Age"),
}

# Test and evaluate each imputation technique
for technique, imputer in imputation_techniques.items():
    # Train the imputer
    imputer.fit(train_df=impute_data, num_epochs=50)
    # Predict and impute missing values
    imputed_data = imputer.predict(impute_data)
```

TRAITEMENT DES VALEURS MANQUANTES

- ✖ Outils commerciaux : qui permettent d'exploiter les connaissances de domaine simple (par exemple, le code postal, la vérification orthographique) pour détecter les erreurs et apporter des corrections.

ANALYSE EXPLORATOIRE DES DONNÉES (EDA)

L'Exploration des Données (EDA) consiste à explorer les données afin d'en comprendre les caractéristiques de manière visuelle et statistique avant de procéder à une analyse approfondie ou à la construction de modèles.

Cette étape permet de :

- ✖ Identifier des tendances, des patterns et des relations entre les variables.
- ✖ Vérifier les hypothèses.
- ✖ Déetecter des anomalies ou des valeurs aberrantes.
- ✖ Vérifier la distribution des variables.

Les principales étapes de l'EDA comprennent :

- ✖ **Visualisation des données**
- ✖ **Analyse statistique des distributions.**
- ✖ **Recherche d'anomalies ou de corrélations.**
- ✖ **Analyse des relations entre variables** (bivariées, multivariées).

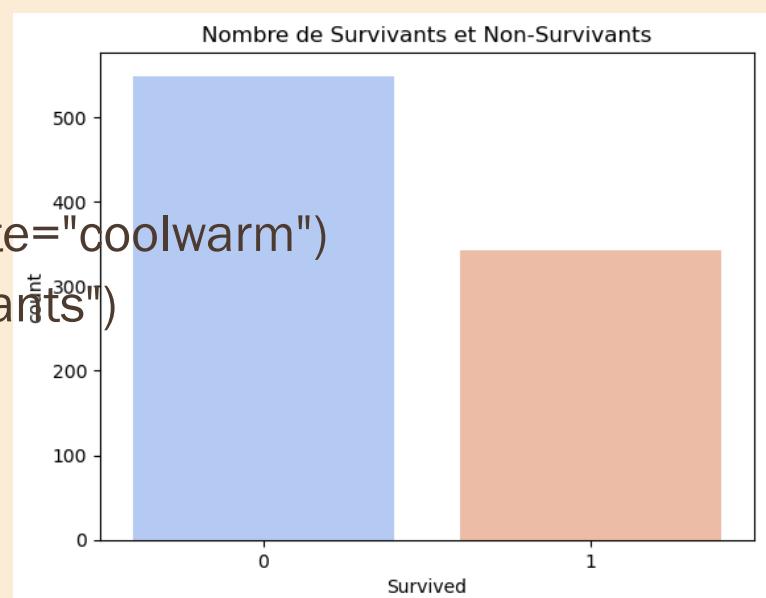
ANALYSE EXPLORATOIRE DES DONNÉES (EDA)

Visualisation des données :

Une première visualisation concerne le label et devrait permettre de compter le nombre de classe et visualiser l'équilibre du label

❖ Distribution des survivants :

```
import seaborn as sns  
import matplotlib.pyplot as plt  
sns.countplot(x="Survived", data=data, palette="coolwarm")  
plt.title("Nombre de Survivants et Non-Survivants")  
plt.show()
```



ANALYSE EXPLORATOIRE DES DONNÉES (EDA)

Analyse de distribution des données:

Des graphiques comme les histogrammes ou les boxplots pour examiner la distribution des variables numériques.

```
import matplotlib.pyplot as plt
import seaborn as sns

# Histogramme pour une colonne
sns.histplot(dataset['colonne_numerique'], kde=True)
plt.title('Histogramme de la colonne_numerique')
plt.show()

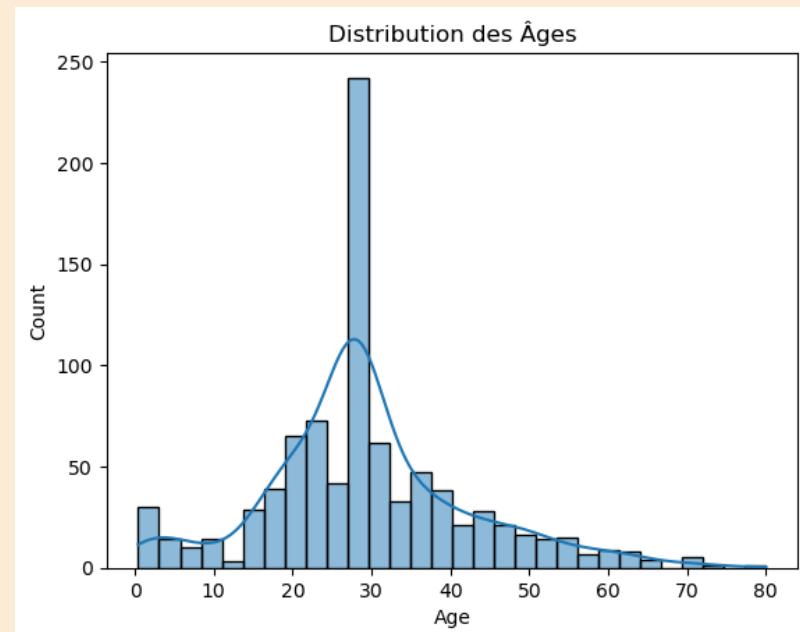
# Boxplot pour vérifier les outliers
sns.boxplot(x=dataset['colonne_numerique'])
plt.title('Boxplot de la colonne_numerique')
plt.show()
```

ANALYSE EXPLORATOIRE DES DONNÉES (EDA)

❖ Exemple sur la distribution des âges dans titanic dataset:

Ce graphique de **Seaborn** est utilisée pour tracer une distribution des données, permettant d'afficher à la fois un histogramme et une estimation de la densité de probabilité. Il est très utile pour visualiser la distribution d'une variable numérique et voir l'asymétries des données

```
import seaborn as sns  
import matplotlib.pyplot as plt  
sns.histplot(data["Age"], bins=30, kde=True)  
plt.title("Distribution des Âges")  
plt.show()
```

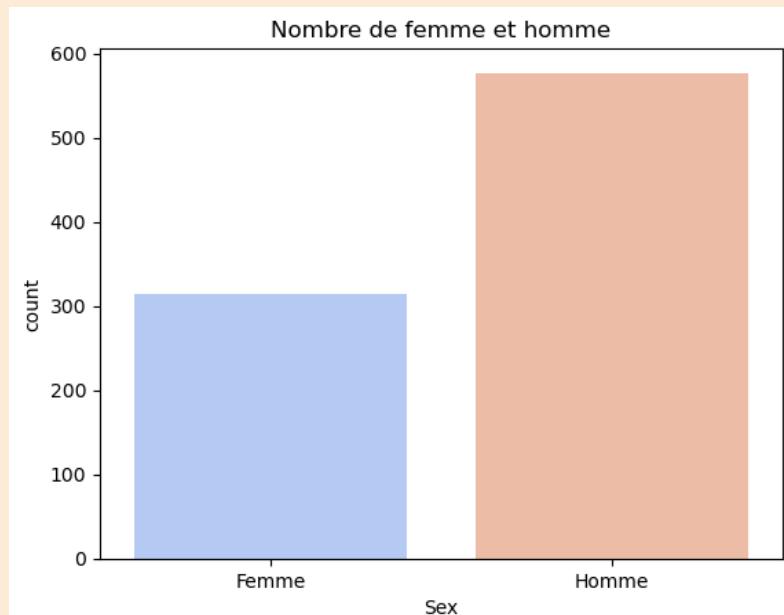


Remarque : la distribution n'est pas symétrique, d'où le remplacement des valeurs manquantes par la médiane

ANALYSE EXPLORATOIRE DES DONNÉES (EDA)

📌 Distribution des âges:

```
import seaborn as sns  
import matplotlib.pyplot as plt  
sns.countplot(x="Sex", data=data, palette="coolwarm")  
plt.title("Nombre de femme et homme")  
plt.xticks(ticks=[0,1], labels=["Femme" , "Homme"])  
plt.show()
```



ANALYSE EXPLORATOIRE DES DONNÉES (EDA)

Analyse bivariée :

L'analyse bivariée explore les relations entre deux variables. Vous pouvez observer comment deux variables interagissent à travers des graphiques tels que les diagrammes de dispersion (scatter plots) ou les corrélations.

1. Scatter plot pour les variables continues :

```
sns.scatterplot(x=dataset['colonne1'],y=dataset['colonne2'])  
plt.title('Diagramme de dispersion entre colonne1 et colonne2') plt.show()
```

2. Matrice de corrélation :

Pour visualiser les corrélations entre les variables numériques, vous pouvez utiliser une matrice de corrélation avec seaborn.

```
corr_matrix = dataset.corr() # Afficher la matrice de correlation  
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm') plt.title('Matrice de corrélation') plt.show()
```

ANALYSE EXPLORATOIRE DES DONNÉES (EDA)



Matrice de corrélation pour Titani dataset:

```
import seaborn as sns
```

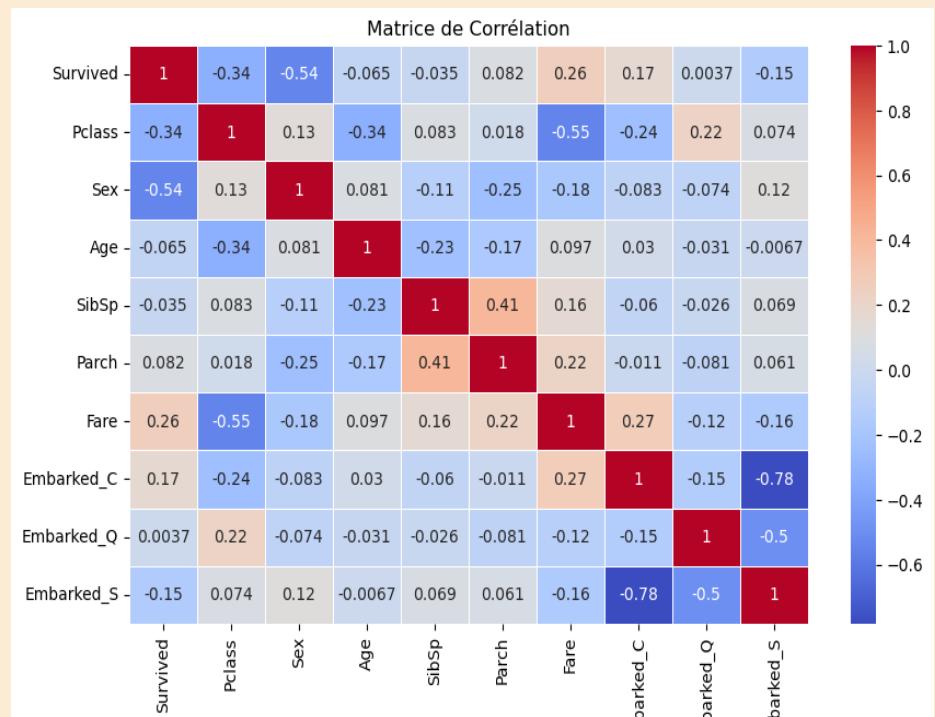
```
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(10, 6))
```

```
sns.heatmap(data.corr(), annot=True, cmap="coolwarm", linewidths=0.5)
```

```
plt.title("Matrice de Corrélation")
```

```
plt.show()
```



DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

- Repérer ces dernières au moyen d'une règle pré-établie.
- méthodes graphiques : histogramme, boxplot, scatterplots
- Clustering : détecter les exceptions
- Par partitionnement (binning)
 - Trier et partitionner les données
 - Lisser les partitions par la moyenne, la médiane, les bornes, ...
- Inspection humaine et informatique combinée : détection des valeurs suspectes et vérification humaine

DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

Binning

Partitionnement à largeur égale (distance) :

Il divise la plage en N intervalles de taille égale :

si A et B sont les valeurs les plus basses et les plus élevées de l'attribut, la largeur des intervalles sera : $W = (B-A)/N$.

Le plus simple Mais les valeurs aberrantes peuvent dominer la présentation

Partitionnement à profondeur égale (fréquence) :

Il divise la plage en N intervalles, chacun contenant approximativement le même nombre d'échantillons

Bonne mise à l'échelle des données La gestion des attributs catégoriels peut être délicate.

DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

Méthodes plus avancées

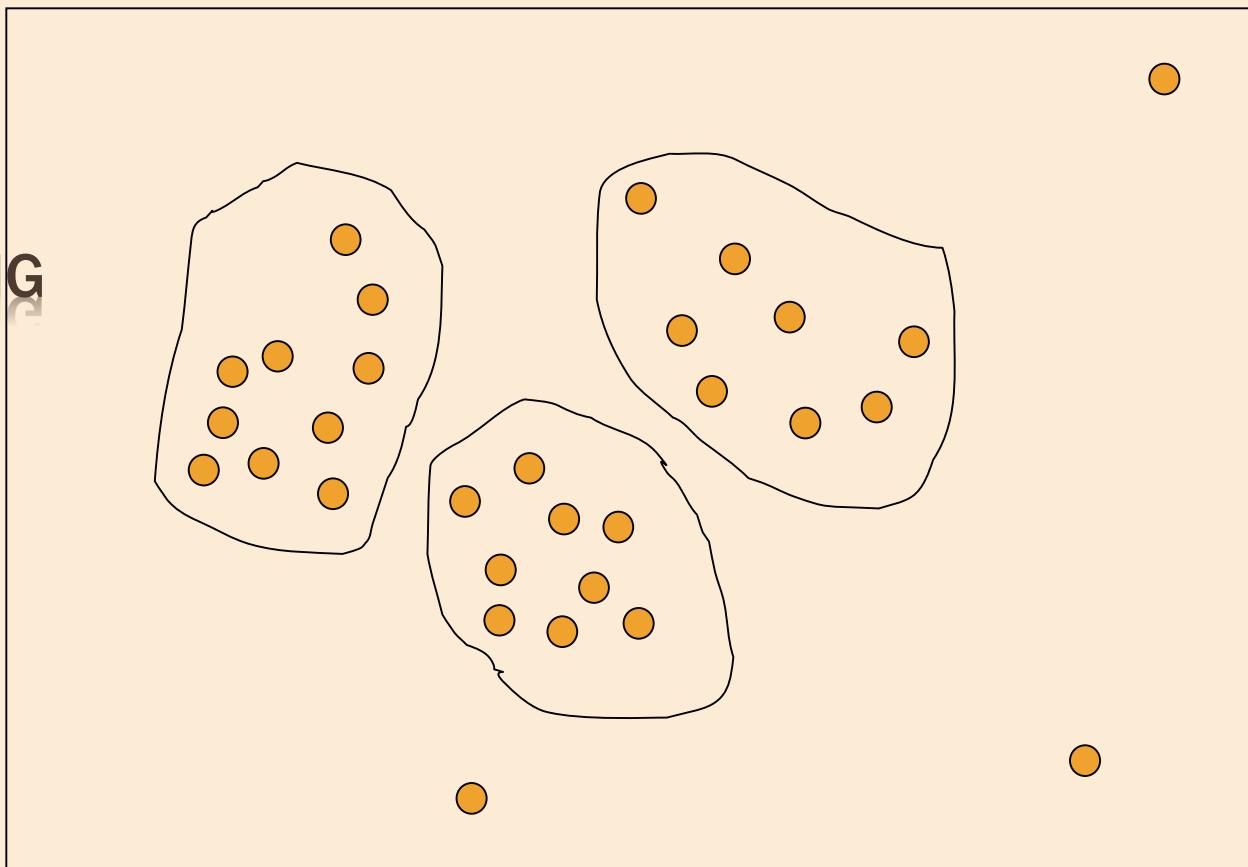
- ✖ Normalisation Z-score :
- ✖ Méthode du quartile - Calculez la différence entre le troisième quartile (Q3) et le premier quartile (Q1). - Identifiez les valeurs aberrantes comme des points de données inférieurs à $Q1 - 1,5 * IQR$ ou supérieurs à $Q3 + 1,5 * IQR$.
- ✖ Méthode ML:

Isolation Forest

Clustering (DBSCAN par exemple)

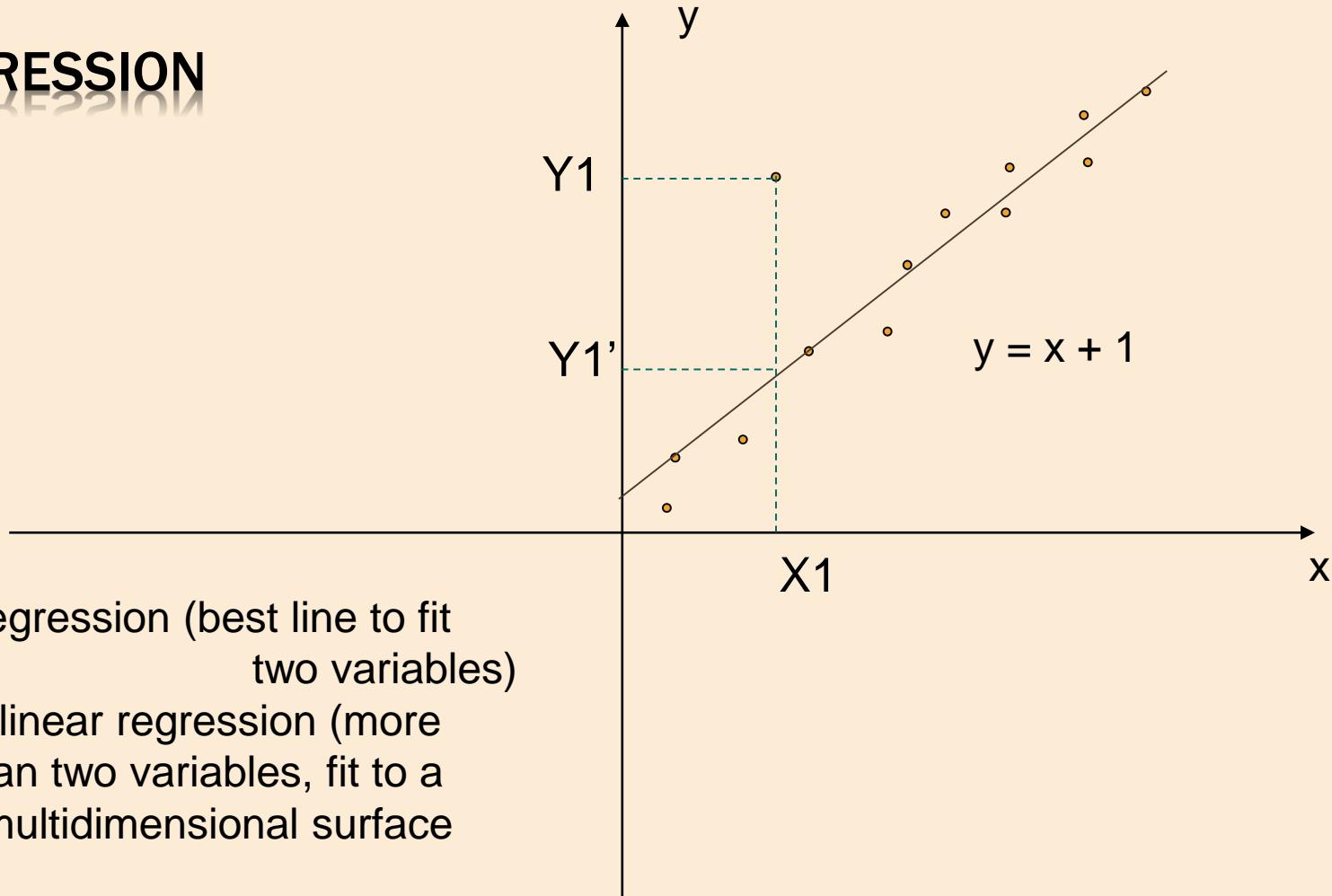
DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

CLUSTERING



DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

REGRESSION



DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

Que faire avec les données aberrantes ?

Trois solutions s'offrent à nous :

- 1 : On supprime l'individu. Le risque est de supprimer par la même occasion des valeurs utiles. Si la population est nombreuse, ce risque devient très faible.
- 2 : On remplace la valeur aberrante par une autre valeur : la valeur moyenne, une valeur prise au hasard dans la distribution des valeurs. Le risque est de produire une incohérence par rapport aux autres valeurs (aberration croisée).
- 3 . Imputation
- 4 : On réduit les valeurs possibles pour la valeur du fait de l'existence de corrélations entre la variable à valeur avec d'autres variables renseignées.

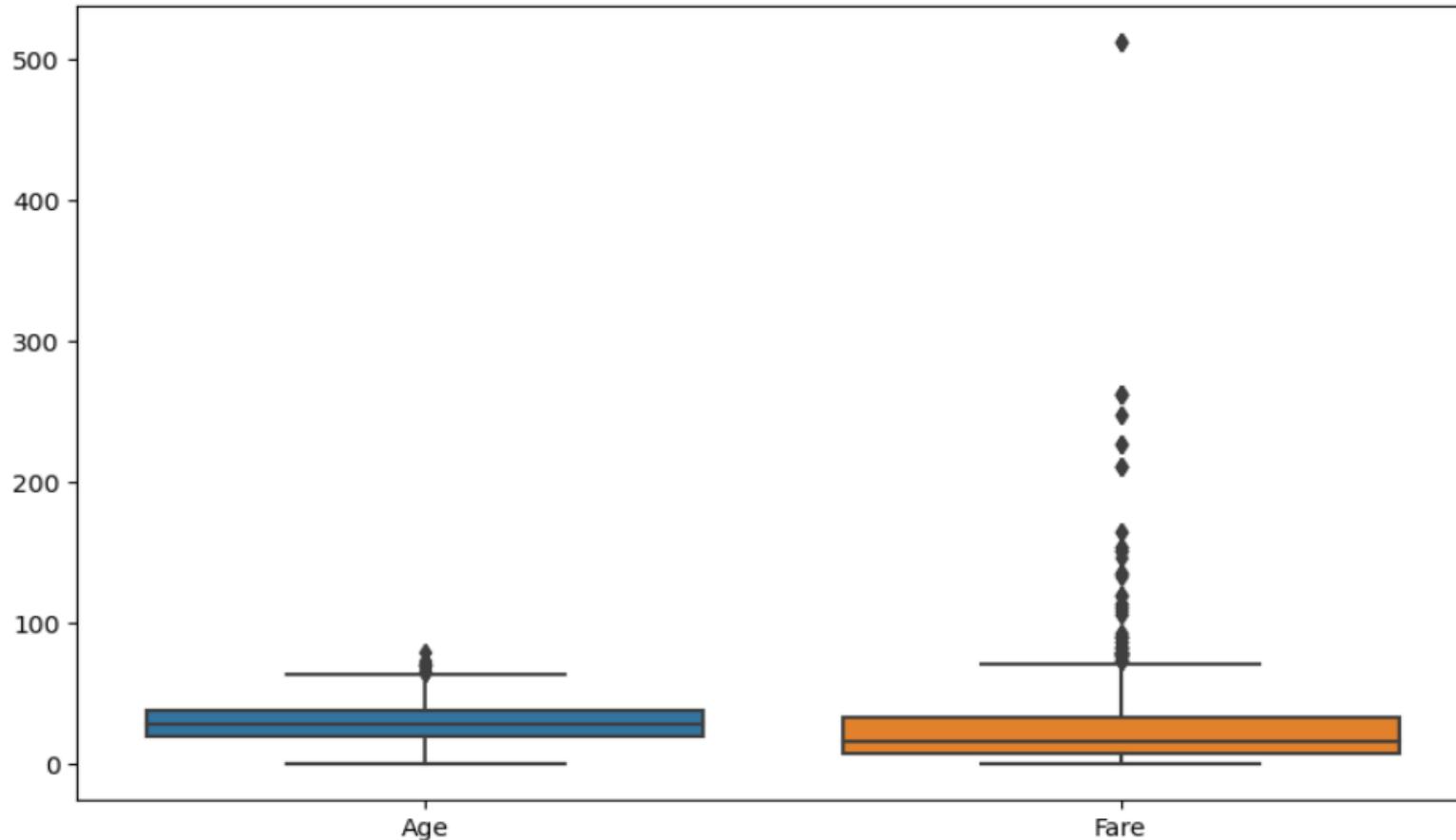
DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

- ✖ Destection des outliers sur titanic dataset: les boxplots
 - + Histogramme
 - + Scatterplots
 - + L'Interquartile Range (IQR) est une mesure statistique qui représente l'étendue des 50% centraux d'un jeu de données, c'est-à-dire la différence entre le troisième quartile (Q3) et le premier quartile (Q1). Il permet d'identifier la dispersion des données et est souvent utilisé pour détecter les valeurs aberrantes (outliers).
 - + Z-Score or Standard Deviation
 - + l'Isolation Forest : est un algorithme de détection d'anomalies qui isole les observations rares en divisant de manière aléatoire les données. Les anomalies sont isolées plus rapidement que les points normaux, ce qui permet de les identifier efficacement.

DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

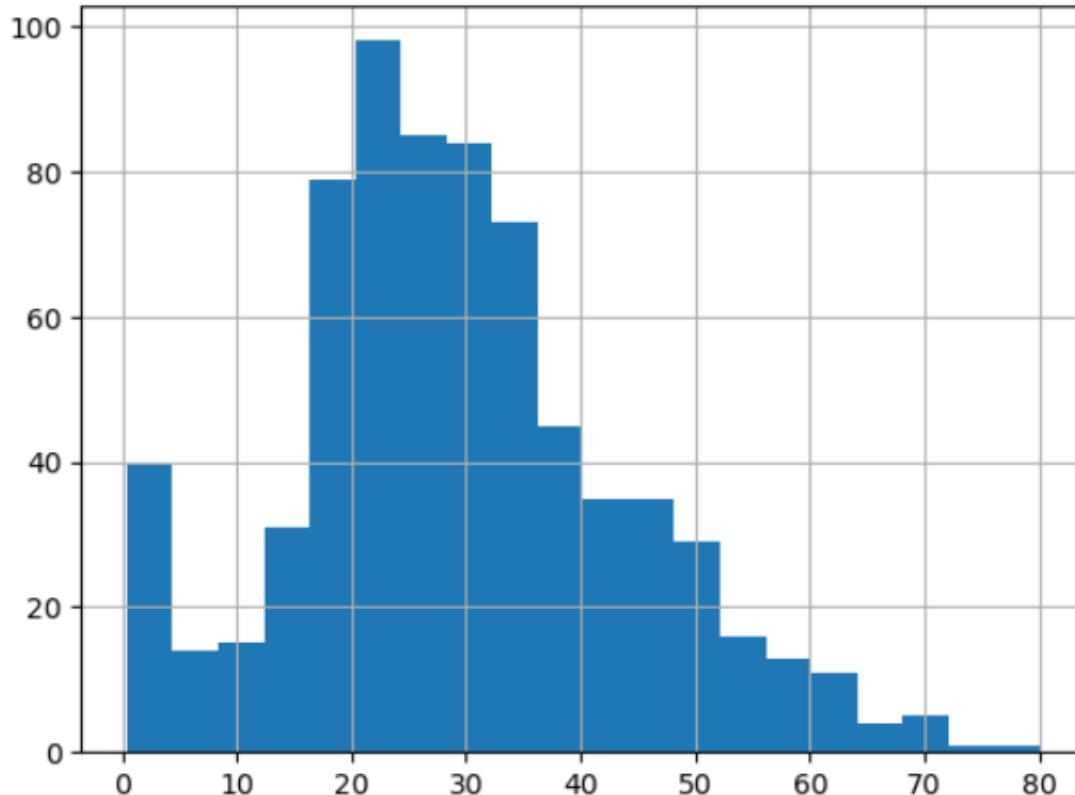
```
import seaborn as sns
import matplotlib.pyplot as plt
# Visualize outliers using box plot
plt.figure(figsize=(10, 6))
sns.boxplot(data=d[['Age', 'Fare']])
plt.title('Box Plot for Age and Fare in Titanic Dataset')
plt.show()
```

Box Plot for Age and Fare in Titanic Dataset



DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

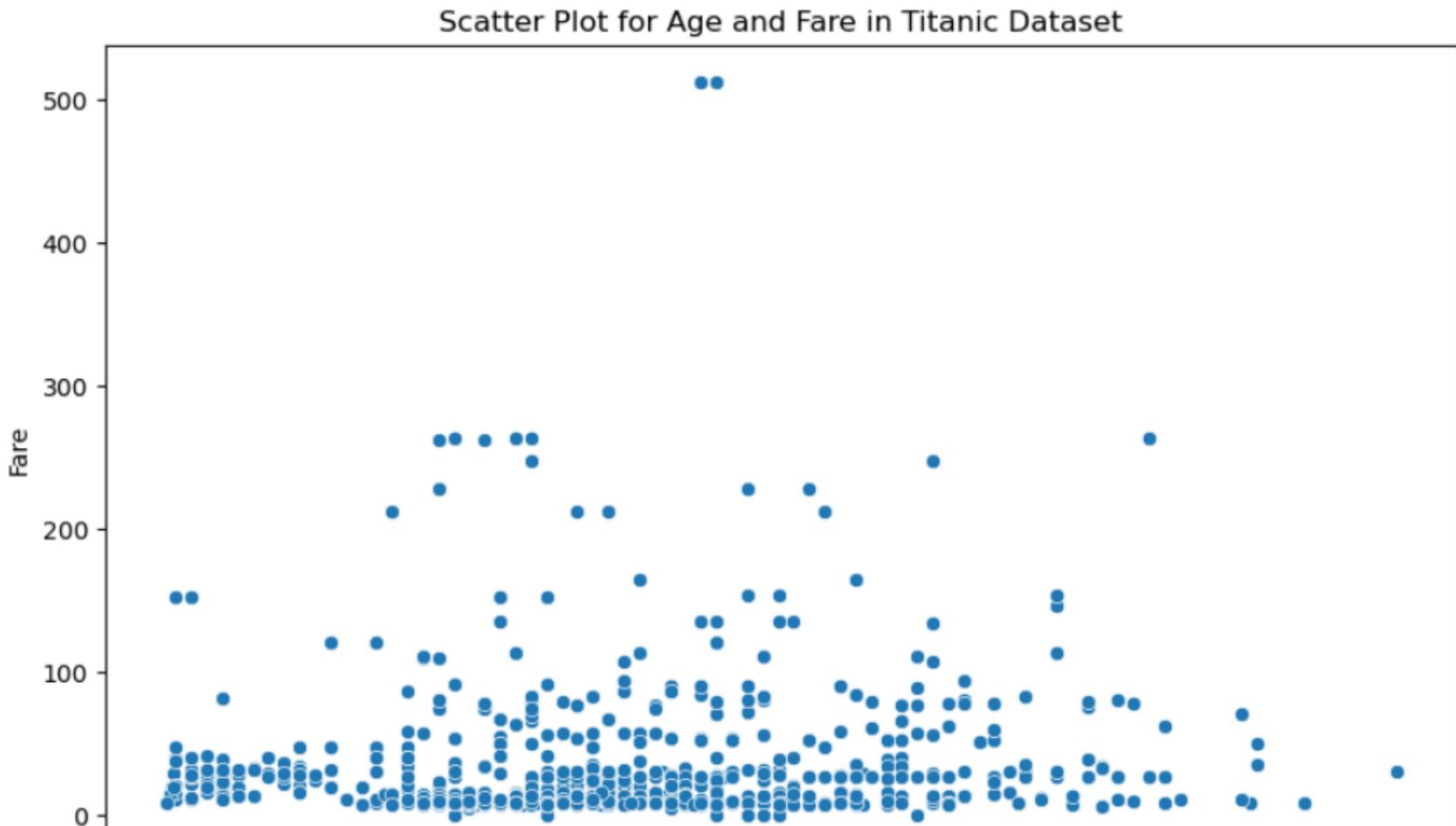
```
import matplotlib.pyplot as plt  
plt.figure()  
d['Age'].hist(bins='auto', alpha=1)  
plt.show()
```



Remarque: ce graphique permet aussi de visualiser l'asymétrie des données

DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

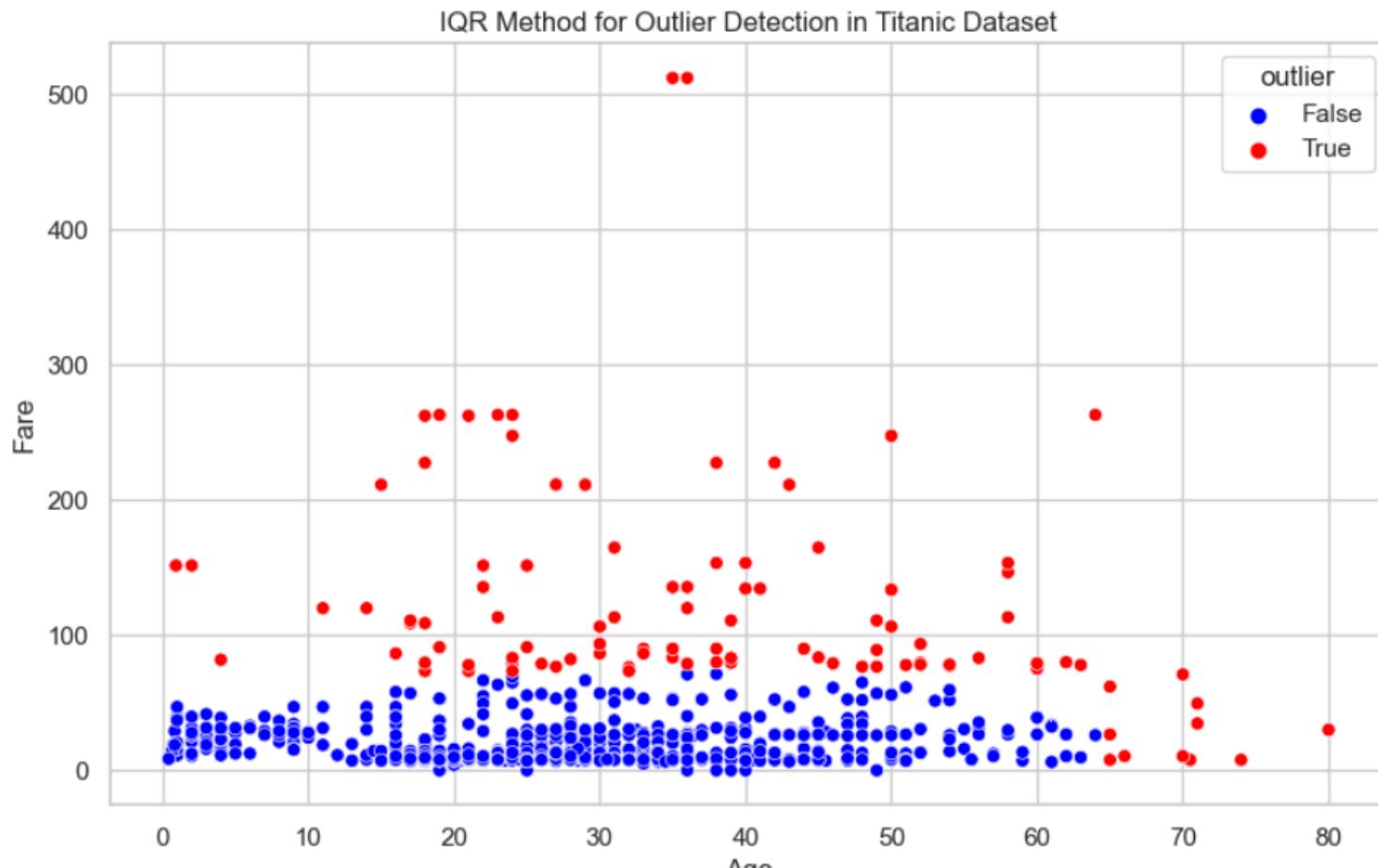
```
# Visualize outliers using scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=d, x='Age', y='Fare')
plt.title('Scatter Plot for Age and Fare in Titanic Dataset')
plt.show()
```



DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

In [105]:

```
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.scatterplot(data=d, x='Age', y='Fare', hue='outlier', palette={True: 'red', False: 'blue'}, legend='full')
plt.title('IQR Method for Outlier Detection in Titanic Dataset')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.show()
```



DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

cette méthode identifie les points de données qui sont éloignés de la moyenne en fonction de l'écart type. Les points de données au-delà d'un certain seuil sont considérés comme des valeurs aberrantes.

In [88]:

```
d.shape
```

Out[88]:

```
(714, 2)
```

In [89]:

```
# Z-Score method
from scipy import stats
import numpy as np
z_scores = stats.zscore(d)
z_scores
```

Out[89]:

	Age	Fare
0	-0.530377	-0.518978
1	0.571831	0.691897
2	-0.254825	-0.506214
3	0.365167	0.348049
4	0.365167	-0.503850
...
885	0.640719	-0.105320
886	-0.185937	-0.410245
887	-0.737041	-0.088774
889	-0.254825	-0.088774
890	0.158503	-0.509523

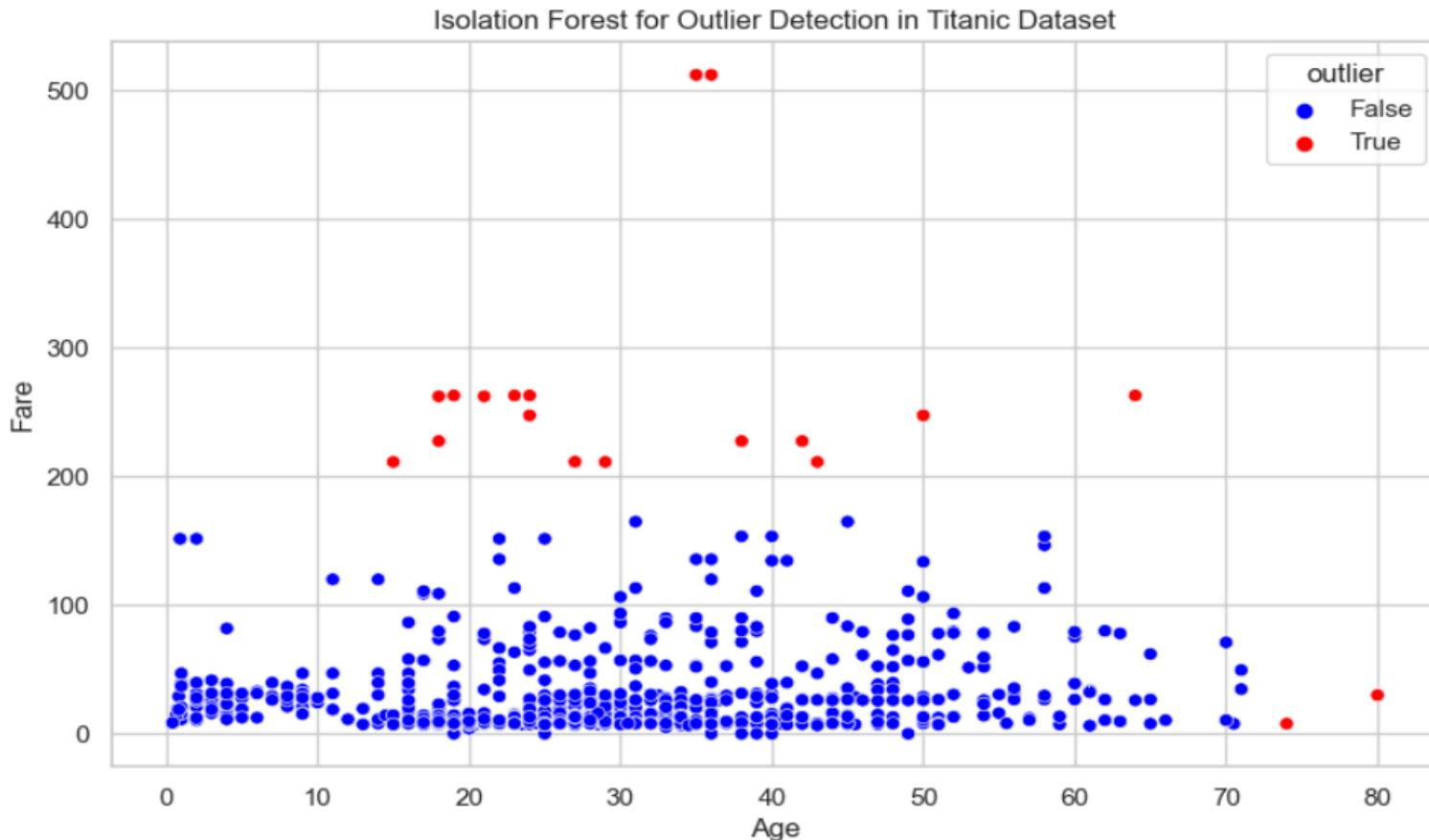
Z-Score or Standard Deviation Method

714 rows × 2 columns

DÉTECTION DES ANOMALIES ET VALEURS ABERRANTES :

In [92]:

```
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.scatterplot(data=d, x='Age', y='Fare', hue='outlier', palette={True: 'red', False: 'blue'}, legend='full')
plt.title('Isolation Forest for Outlier Detection in Titanic Dataset')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.show()
```



TRAITEMENT DES ANOMALIES ET VALEURS ABERRANTES :

La gestion des anomalies et des valeurs aberrantes dépend du contexte et des objectifs de votre analyse. Voici quelques approches pour les traiter :

- **Supprimer** : Lorsque les anomalies ne sont pas importantes pour l'analyse.
- **Transformer** : Lorsque vous souhaitez réduire l'impact des valeurs aberrantes sans les supprimer.
- **Imputer** : Lorsque vous voulez remplacer les anomalies par des valeurs plausibles.
- **Modèles robustes** : Lorsque vous ne souhaitez pas traiter explicitement les anomalies.

L'INTÉGRATION DES DONNÉES

- Intégration de données:

Combine les données provenant de sources multiples dans un entrepôt cohérent

L'intégration de schéma: source différentes => schémas de BD différents

- Intégrer des métadonnées à partir de différentes sources

Problème d'identification de l'entité: Identifier les entités du monde réel à partir de multiples sources de données, par exemple, Bill Clinton = William Clinton

- La détection et la résolution des conflits de valeurs de données

Pour la même entité du monde réel, les valeurs d'attributs provenant de différentes sources sont différentes (l'heure dans <> serveurs par exemple, les métriques et les unités de mesures)

GESTION DES ATTRIBUTS TEMPORELS

- ✗ timestamps : définir l'intervalle de temps le plus approprié et faire attention au décalage horaire lors de l'intégration des données de différentes sources

ENLEVER LA REDONDANCE

- ✖ Les données redondantes se produisent souvent lors de l'intégration de plusieurs bases de données
- ✖ L'identification des objets: Le même attribut ou objet peut avoir différents noms dans différentes bases de données
- ✖ Données dérivées: Un attribut peut être "dérivé« d'un autre dans une autre table, par exemple, le revenu annuel
- ✖ Les Attributs redondants peuvent être détectés par analyse de **corrélation** et d'analyse de **covariance**
- ✖ L'Intégration minutieuse des données provenant de sources multiples peut aider à réduire ou même à éviter les redondances et les incohérences et améliorer la vitesse et la qualité de la phase de data mining

ENLEVER LA REDONDANCE

Pour enlever les données redondantes (ou dupliquées) dans un jeu de données, vous pouvez utiliser plusieurs techniques en fonction du type de redondance. Voici comment procéder dans le cadre d'un **DataFrame Pandas** :

- drop_duplicates()** : Supprime les lignes identiques.
- unique()** : Retourne les valeurs uniques d'une colonne.
- reset_index()** : Réinitialise l'index après avoir supprimé des doublons.
- dropna()** : Supprime les lignes contenant des valeurs manquantes avant ou après la suppression des doublons.

ENLEVER LA REDONDANCE

1. Supprimer les lignes dupliquées :

Si vous avez des lignes entièrement identiques dans votre jeu de données, vous pouvez les supprimer à l'aide de la méthode **drop_duplicates()** de Pandas.

Exemple en Python :

```
python
# Supprimer les lignes redondantes
df_unique = df.drop_duplicates()
print(df_unique)
```

Explication :

La fonction **drop_duplicates()** supprime toutes les lignes qui sont identiques. Par défaut, elle supprime les doublons sur toutes les colonnes,

Pour supprimer les doublons en se basant sur des colonnes spécifiques :

```
python
df_unique = df.drop_duplicates(subset=['Name', 'Age'])
```

ENLEVER LA REDONDANCE

2. Supprimer les valeurs redondantes dans une colonne spécifique :

Si vous avez une colonne avec des valeurs répétées et que vous voulez ne garder que les valeurs uniques, vous pouvez utiliser **drop_duplicates()** sur cette seule colonne ou utiliser **unique()**.

Exemple en Python :

python

```
# Garder les valeurs uniques dans la colonne 'Name' unique_names =  
df['Name'].drop_duplicates() # Alternativement, utiliser unique() pour une colonne  
unique_names = df['Name'].unique() print(unique_names)
```

3. Supprimer les doublons basés sur plusieurs colonnes :

Si vous souhaitez supprimer les doublons en vous basant sur une combinaison de plusieurs colonnes, vous pouvez spécifier ces colonnes dans l'argument **subset** de **drop_duplicates()**.

Exemple :

python

```
# Supprimer les doublons basés sur les colonnes 'Name' et 'Age' df_unique =  
df.drop_duplicates(subset=['Name', 'Age']) print(df_unique)
```

TRANSFORMATION DES DONNÉES (CONVERSION)

- ✖ Certaines techniques d'extraction de connaissance peuvent accepter les valeurs nominaux d'autres non (réseaux neuronaux, régression, k plus proche voisin).
- ✖ Ces techniques nécessitent des attributs uniquement numériques.

De plus

- ✖ Convertir les champs nominaux dont les valeurs ont un ordre en valeur numérique permet d'utiliser les opérateurs de comparaison tel que ">" et "<".

TRANSFORMATION DES DONNÉES (CONVERSION)

NOMINALE AU NUMÉRIQUE

Integer encoding :

Dans un premier temps, chaque valeur de catégorie unique se voit attribuer une valeur entière. Par exemple, "rouge" vaut 1, "vert" vaut 2 et "bleu" vaut 3.

C'est ce qu'on appelle un codage d'étiquette ou un codage d'entier et est facilement réversible.

Pour certaines variables, cela peut suffire. Les valeurs entières ont une relation ordonnée naturelle entre elles et les algorithmes d'apprentissage automatique peuvent être en mesure de comprendre et d'exploiter cette relation. Par exemple, des variables ordinaires comme l'exemple « lieu » ci-dessus seraient un bon exemple où un codage d'étiquette serait suffisant.

TRANSFORMATION DES DONNÉES (CONVERSION)

NOMINALE AU NUMÉRIQUE

One hot encoding :

Pour les variables catégorielles pour lesquelles aucune relation ordinaire de ce type n'existe, l'encodage entier n'est pas suffisant. En fait, utiliser cet encodage et permettre au modèle de supposer un ordre naturel entre les catégories peut entraîner des performances médiocres ou des résultats inattendus (prédictions à mi-chemin entre les catégories).

Dans ce cas, un codage one-hot peut être appliqué à la représentation entière. C'est là que la variable entière codée est supprimée et qu'une nouvelle variable binaire est ajoutée pour chaque valeur entière unique. Dans l'exemple de la variable « couleur », il y a 3 catégories et donc 3 variables binaires sont nécessaires. Une valeur « 1 » est placée dans la variable binaire pour la couleur et des valeurs « 0 » pour les autres couleurs.

TRANSFORMATION DES DONNÉES (CONVERSION)

ONE HOT ENCODING

Type		Type	AA_Onehot	AB_Onehot	CD_Onehot
AA		AA	1	0	0
AB	Onehot encoding	AB	0	1	0
CD		CD	0	0	1
AA		AA	0	0	0

EN PYTHON : *sklearn.preprocessing.OneHotEncoder*

OU

pd.get_dummies

TRANSFORMATION DES DONNÉES (CONVERSION)

DESCRITISATION

Objectif :

Réduire le nombre de valeurs pour un attribut continu donné en partitionnant la plage de l'attribut en intervalles, Les étiquettes d'intervalle remplacent les valeurs d'attribut réelles

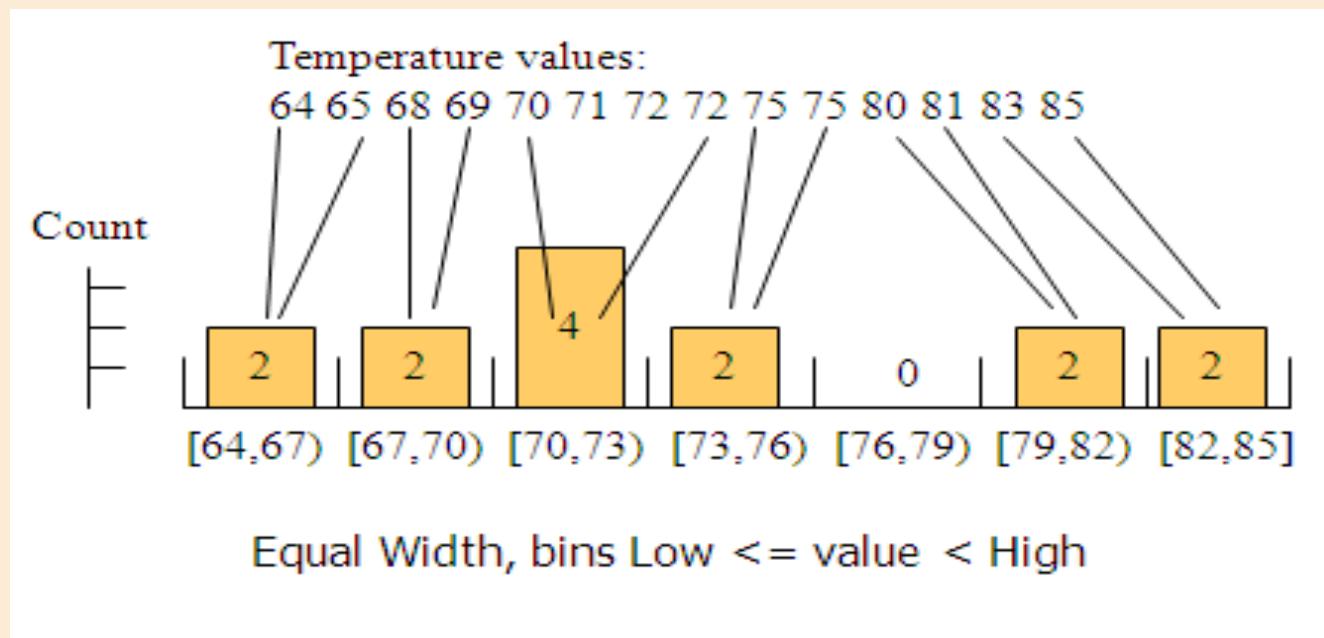
Méthodes :

- Binning (comme expliqué précédemment)
- Analyse de cluster (sera discutée plus tard)
- Discrétisation basée sur les technique ML (l'entropie (supervisée), khi 2,....)

TRANSFORMATION DES DONNÉES (CONVERSION)

DISCRÉTISATION DE DONNÉE

- Certaines méthodes exigent des valeurs discrètes, par exemple la plupart des versions de Naïve Bayes.
La discréétisation est très utile pour la génération d'un résumé de données, il est appelée aussi "binning".
- Exemple : valeurs de température

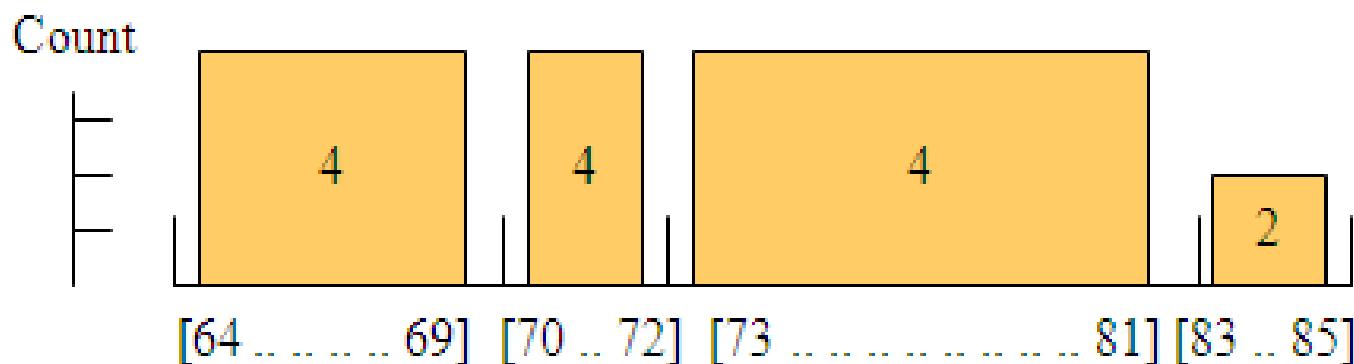


TRANSFORMATION DES DONNÉES (CONVERSION)

OU

Temperature values:

64 65 68 69 70 71 72 72 75 75 80 81 83 85



Equal Height = 4, except for the last bin

TRANSFORMATION DES DONNÉES (CONVERSION)

DESCRITISATION

Quel méthode de discréétisation sera la meilleure ? Comme toujours, cela dépendra de :

- ✖ Du problème, des besoins des utilisateurs, etc...
- ✖ De l'évaluation
 - + Nombre total d'intervalles
 - + Nombre d'incohérences causées
 - + Taux de performance prédictif

TRANSFORMATION DES DONNÉES (CONVERSION)

📌 Encodage des variables catégoriques sur le dataset Titanic pour l'attribut sex :

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
data['Sex'] = le.fit_transform(data['Sex']) # 1: male, 0: female
```

```
1 print(data['Sex'].unique())
2 print(data['Sex'].unique())
3 print(data['Sex'].value_counts())
```

```
['male' 'female']
['male' 'female']
male      577
female    314
Name: Sex, dtype: int64
```

```
1 #Label encoding the data
2 from sklearn.preprocessing import LabelEncoder
3 le = LabelEncoder()
4 #1: male 0: female
5 data['Sex']= le.fit_transform(data['Sex'])
6 print(data['Sex'].value_counts())
```

```
1     577
0     314
Name: Sex, dtype: int64
```

TRANSFORMATION DES DONNÉES (CONVERSION)

🔨 Encodage des variables catégoriques 🔨 Encodage des variables catégoriques sur le dataset Titanic pour l'attribut embarked :

```
data = pd.get_dummies(data, columns=["Embarked"])
```

```
1 print(data['Embarked'].unique())
2 print(data['Embarked'].unique())
3 print(data['Embarked'].value_counts())
['S', 'C', 'Q']
['S', 'C', 'Q']
S    646
C    168
Q    77
Name: Embarked, dtype: int64
```

```
1 # One-Hot Encoding
2 data = pd.get_dummies(data, columns=["Embarked"])

1 data
   Survived Pclass Sex Age SibSp Parch Fare Embarked_C Embarked_Q Embarked_S
0         0     3   1  22.0     1     0  7.2500         0         0       1
1         1     1   0  38.0     1     0  71.2833         1         0       0
2         1     3   0  26.0     0     0  7.9250         0         0       1
3         1     1   0  35.0     1     0  53.1000         0         0       1
4         0     3   1  35.0     0     0  8.0500         0         0       1
...
886        0     2   1  27.0     0     0  13.0000         0         0       1
887        1     1   0  19.0     0     0  30.0000         0         0       1
888        0     3   0  28.0     1     2  23.4500         0         0       1
889        1     1   1  26.0     0     0  30.0000         1         0       0
890        0     3   1  32.0     0     0  7.7500         0         1       0
```

891 rows × 10 columns

LA TRANSFORMATION DE DONNÉE (NORMALISATION ET STANDARDISATION)

Normalization et standardization : Ces deux techniques permettent d'améliorer la performance des algorithmes en réduisant l'impact des différences d'échelle entre les variables. Par exemple, les variables dont l'échelle est beaucoup plus grande que d'autres peuvent dominer l'apprentissage si elles ne sont pas transformées.

1. Normalisation (Min-Max Scaling) :

- **But** : Mettre à l'échelle les données pour qu'elles aient une plage de valeurs spécifiée, généralement entre 0 et 1.
- **Utilité** : Utile pour les algorithmes sensibles à l'échelle des données, comme les réseaux neuronaux et les k-NN.

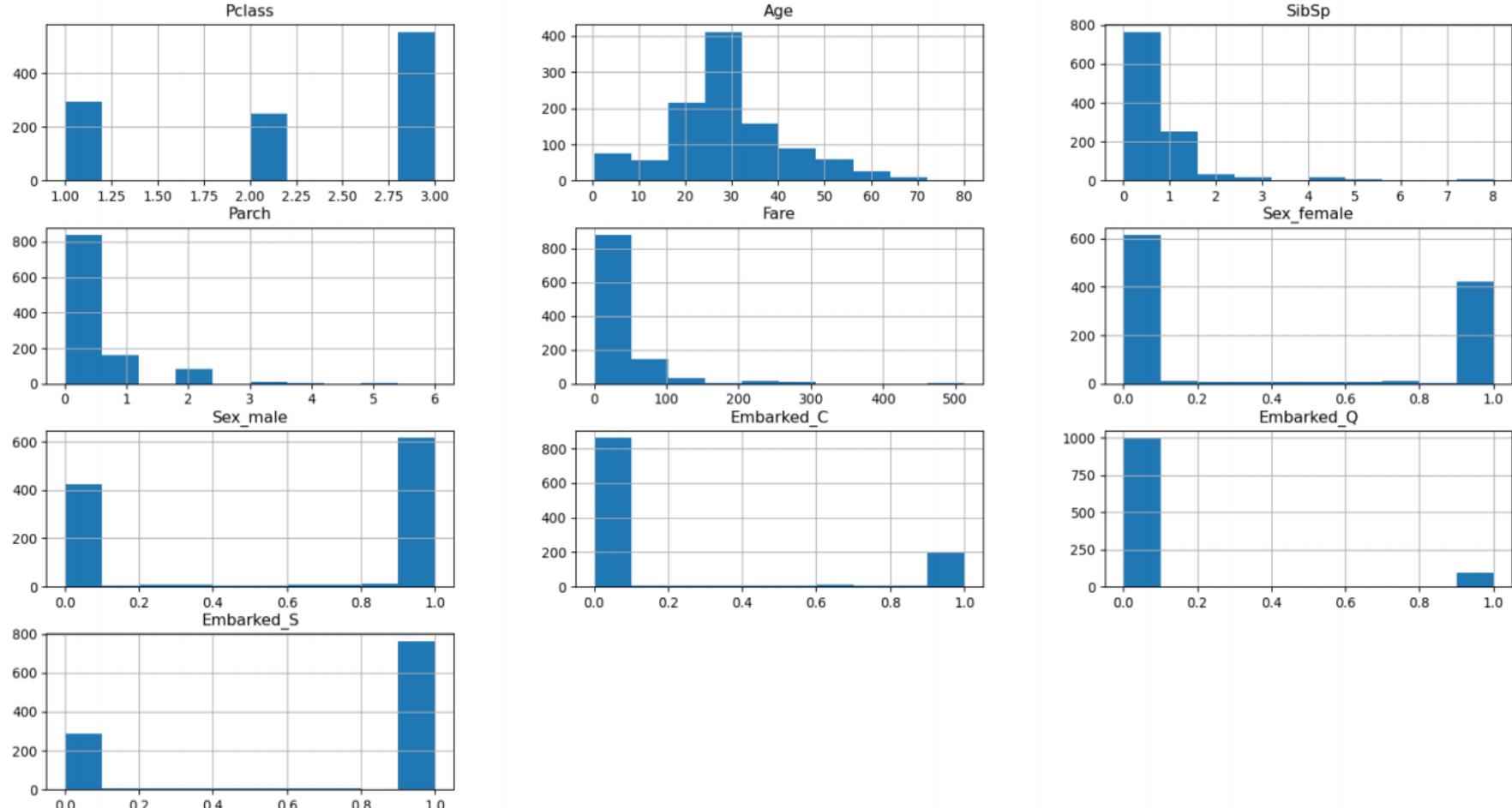
- **Formule** :

$$X_{\text{norm}} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

LA TRANSFORMATION DE DONNÉE (NORMALISATION ET STANDARDISATION)

Exploration sur Titanic dataset

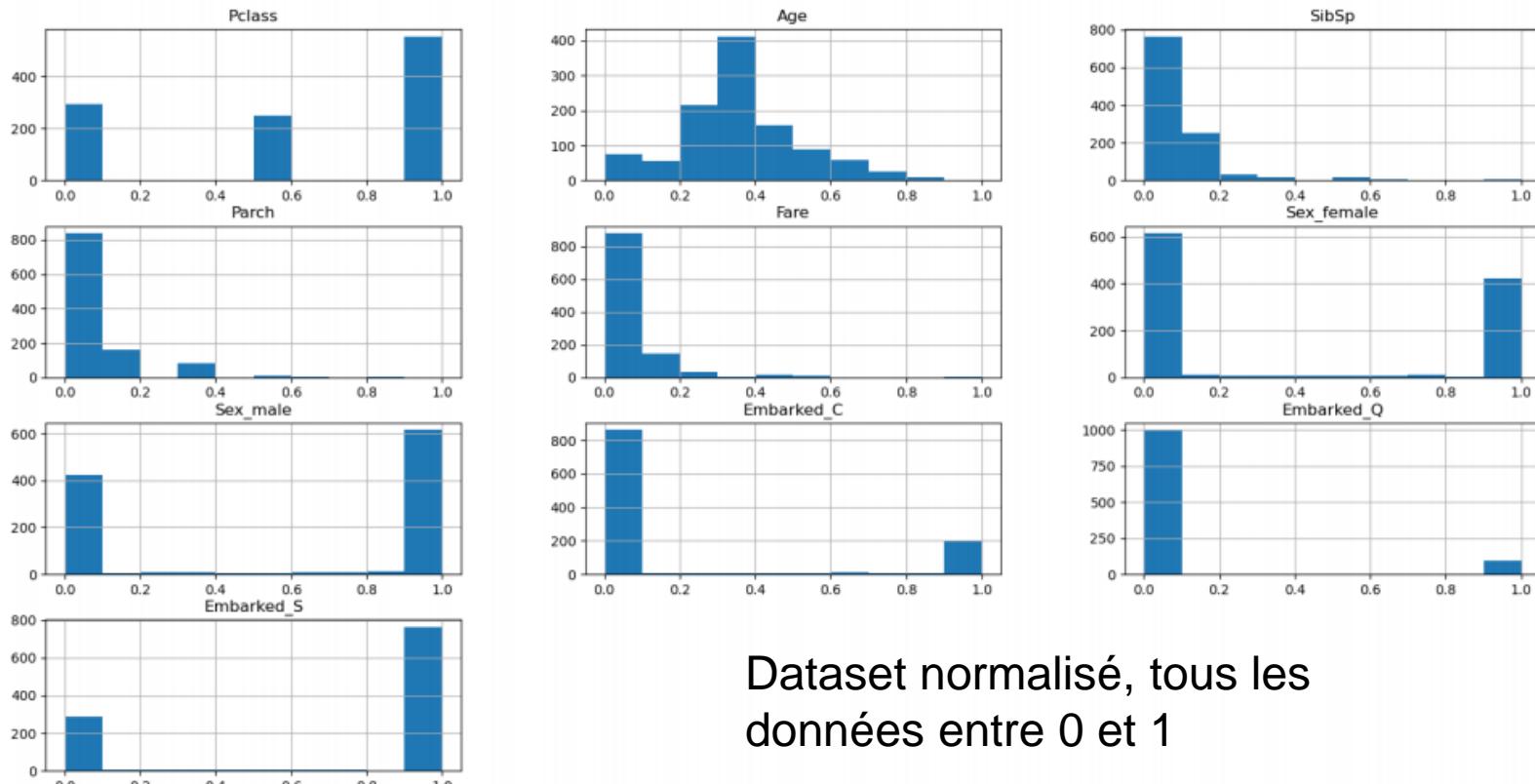
```
import matplotlib.pyplot as plt
X_s.hist(figsize=(20,10))
plt.show()
```



LA TRANSFORMATION DE DONNÉE (NORMALISATION ET STANDARDISATION)

Exploration sur Titanic dataset,

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_s = scaler.fit_transform(X_s)
from pandas import DataFrame
X_s = DataFrame(X_s, columns=[  
    'Pclass', 'Age', 'SibSp', 'Parch', 'Fare', 'Sex_female', 'Sex_male', 'Embarked_C', 'Embarked_Q', 'Embarked_S'  
])
X_s.hist(figsize=(20,10))
plt.show()
```



LA TRANSFORMATION DE DONNÉE (NORMALISATION ET STANDARDISATION)

2. Standardisation (Z-Score Scaling) :

- **But** : Centrer les données autour de la moyenne avec une variance de 1.
- **Utilité** : Utile pour les modèles comme la régression linéaire, la SVM et les arbres de décision. Elle est moins sensible aux outliers que la normalisation.

- **Formule** :

$$X_{\text{std}} = \frac{X - \mu}{\sigma}$$

Où μ est la moyenne et σ est l'écart-type.

- **Code python**

```
'''from sklearn.preprocessing import StandardScaler
transformer = StandardScaler()
X = transformer.fit_transform(X)
from pandas import DataFrame
X = DataFrame(X, columns=['Pclass', 'Age', 'SibSp', 'Parch', 'Fare', 'Sex_female', 'Sex_male','Embarked_C', 'Embarked_Q', 'Embarked_S'])'''
```

BIBLIOTHÈQUE PYTHON DE BASE POUR LE TRAITEMENTS DES PROBLÈMES DES DONNÉES

Récapitulatif des bibliothèques :

- **Pandas** : Pour la gestion, la manipulation et la transformation des données (chargement, nettoyage, fusion).
- **Matplotlib** : Pour la visualisation de base des données.
- **Seaborn** : Pour des visualisations plus avancées, notamment statistiques.
- **Datawig** : permet d'imputer automatiquement les valeurs manquantes dans un jeu de données en utilisant des modèles d'apprentissage automatique.
- **Scikit-learn** : Pour l'apprentissage automatique, la modélisation et le prétraitement des données.

Ces bibliothèques sont essentielles pour effectuer des analyses et des prétraitements efficaces dans un pipeline de **Fouille de donnée**. Vous pouvez commencer par charger les données avec Pandas, les nettoyer et les transformer, puis utiliser Matplotlib et Seaborn pour explorer et visualiser les données, datawig pour l'imputation et enfin appliquer des techniques de machine learning avec **Scikit-learn**.

UMBALANCED DATA

Un ensemble de données de classification avec des proportions de classe asymétriques est appelé déséquilibré. Les classes qui constituent une grande proportion de l'ensemble de données sont appelées classes majoritaires. Ceux qui constituent une proportion minimale sont des classes minoritaires.

Qu'est-ce qui compte comme déséquilibré? Les proportions sont très diverses

Degree of imbalance	Proportion of Minority Class
Mild	20-40% of the data set
Moderate	1-20% of the data set
Extreme	<1% of the data set

UMBALANCED DATA

Détection de fraude

L'analyse bioinformatique des maladies (cancer)

ADN mining

.....

Pb :

Avec très peu de points positifs par rapport aux points négatifs, l'apprentissage du modèle passe le plus clair de son temps à donner des exemples négatifs et à ne pas apprendre suffisamment des exemples positifs.

Dans de nombreux cas, la base d'apprentissage peu ne contenir aucun exemple positif. Par conséquent, le modèle sera moins informatifs.

UMBALANCED DATA

Comment réagir à ce problème?

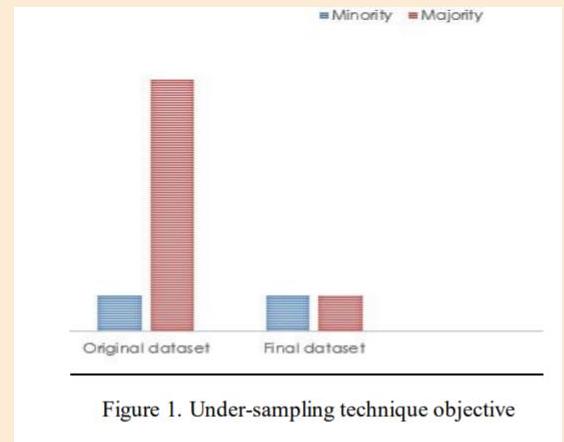
En présence de données déséquilibrées, essayez d'abord de vous entraîner sur la vraie distribution. Si le modèle fonctionne bien et généralise, c'est terminé! Sinon :

- ✖ Le sous échantillonnage (undersampling)
- ✖ La sur échantillonnage (oversampling)
- ✖ hybride

Bibliothèque : imbalanced-learn

UMBALANCED DATA - UNDERSAMPLING

Il existe différentes techniques pour effectuer le processus de sous-échantillonnage, nous allons lister ici les plus utilisés.



Le **sous-échantillonnage aléatoire** (Random under-sampling) divise en échantillons partiels la classe majoritaire en prélevant au hasard des échantillons avec ou sans remplacement.

Les **liens Tomek (Tomek Link)** sont des paires d'instances très proches, mais de classes opposées. Supprimer les instances de la classe majoritaire de chaque paire augmente l'espace entre les deux classes, facilitant le processus de classification.

UMBALANCED DATA - UNDERSAMPLING

AllKNN : « édite » l'ensemble de données en supprimant des échantillons, ceux qui ne sont pas « assez » similaires avec leur voisins.

Elle applique un algorithme du plus proche voisin KNN, pour chaque échantillon dans la classe à sous-échantillonner, les plus proches voisins sont calculé et si le critère de sélection n'est pas rempli, le l'échantillon est retiré. Deux critères de sélection sont actuellement disponibles : la majorité ou la totalité, les Plus Proches Voisins ont appartenir à la même classe que l'échantillon inspecté pour le garder dans l'ensemble de données

NearMiss vise à équilibrer la distribution des classes en éliminant de manière aléatoire les exemples de classe majoritaire. Lorsque des instances de deux classes différentes sont très proches l'une de l'autre, on supprime l'instances de la classe majoritaire pour augmenter l'espaces entre les deux classes. Pour éviter les problèmes de perte d'information dans la plupart des techniques de sous-échantillonnage, les méthodes du Nearneighbor sont largement utilisées.

UMBALANCED DATA - UNDERSAMPLING

✗ Undersampling, Titanic dataset

```
In [17]:
```

```
y.value_counts()
```

```
Out[17]:
```

```
0    549  
1    342  
Name: Survived, dtype: int64
```

```
In [18]:
```

```
from imblearn.under_sampling import RandomUnderSampler  
from imblearn.over_sampling import RandomOverSampler  
# Undersampling  
# Undersampling: Reduce the size of the majority class.  
undersampler = RandomUnderSampler(sampling_strategy='majority')  
X_under, y_under = undersampler.fit_resample(X, y)  
print(y_under.value_counts())
```

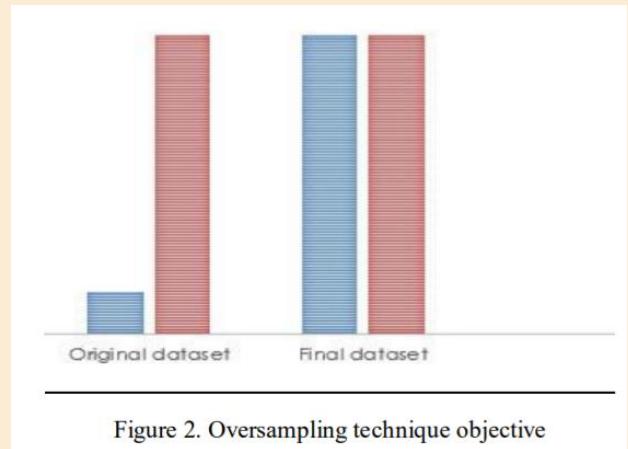
```
0    342
```

```
1    342
```

```
Name: Survived, dtype: int64
```

UNBALANCED DATA - OVERSAMPLING

Suréchantillonnage Le suréchantillonnage consiste à ajouter plus de copies du classe minoritaire.



Suréchantillonnage aléatoire : suréchantillonner la classe minoritaire en prélevant des échantillons au hasard avec remise. Générer de nouvelles instances dans la classes minoritaire par échantillonnage aléatoire avec remise à partir de l'échantillons actuellement disponibles (est la stratégie naïve).

UNBALANCED DATA - OVERSAMPLING

Suréchantillonnage Le suréchantillonnage consiste à ajouter plus de copies du classe minoritaire.

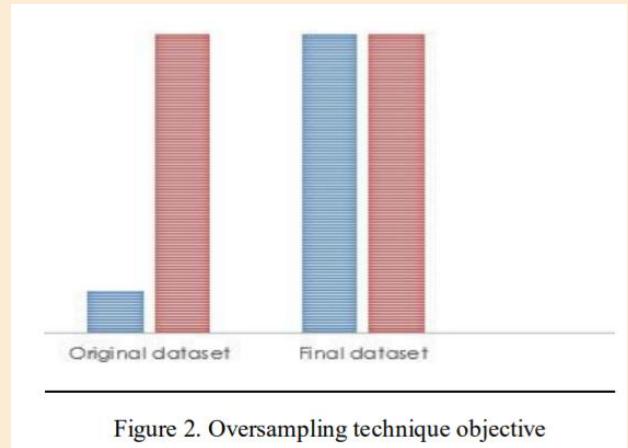


Figure 2. Oversampling technique objective

La génération de données synthétiques, une technique similaire de suréchantillonnage consiste à créer des échantillons synthétiques. Ici, nous allons utiliser SMOTE ou le suréchantillonnage minoritaire synthétique Technique. SMOTE utilise l'algorithme des voisins les plus proches pour générer de nouvelles données synthétiques que nous pouvons utiliser pour la formation de notre modèle.

UNBALANCED DATA - OVERSAMPLING

Approche d'échantillonnage synthétique adaptatif (ADASYN) diffère de SMOTE par le mode de génération du points d'échantillonnage synthétiques pour les points de données minoritaires. Dans ADASYN, nous considérons une distribution de densité r_x , qui décide ainsi du nombre d'échantillons synthétiques à généré pour un point particulier, alors que dans SMOTE ; là un poids uniforme est défini pour tous les points minoritaires.

SVM-SMOTE utilise un algorithme SVM pour détecter l'échantillon à utiliser pour générer de nouveaux échantillons synthétiques avant de suréchantillonner à l'aide de SMOTE.

UMBALANCED DATA - OVERSAMPLING

✖️ Oversampling, Titanic dataset

```
In [19]:
```

```
# Oversampling
# Oversampling: Increase the size of the minority class.
oversampler = RandomOverSampler(sampling_strategy='minority')
X_over, y_over = oversampler.fit_resample(X, y)
print(y_over.value_counts())

0    549
1    549
```

UNBALANCED DATA - HYBRID

La solution hybride est une technique composée avec deux approches anciennes, suréchantillonnage et sous-échantillonnage.

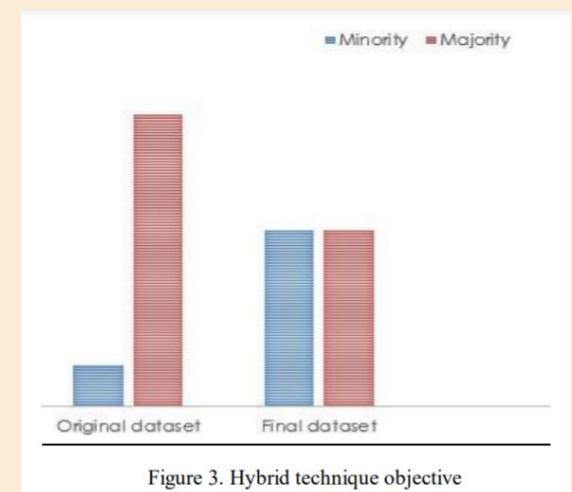


Figure 3. Hybrid technique objective

SMOTETomek est une méthode qui effectue un suréchantillonnage en utilisant SMOTE et le nettoyage en utilisant les liens Tomek. Il combine sur/sous-échantillonnage à l'aide de liens SMOTE et Tomek. Cette dernière technique peut être utilisée comme méthode de nettoyage des données ou comme méthode de sous-échantillonnage. Elle est utilisée dans le suréchantillonner des ensembles d'apprentissage comme méthode de nettoyage des données. Nous avons également la technique **SMOTEENN**, qui combine le sur- et le sous-échantillonnage en utilisant SMOTE et Edited Nearest Neighbors.

UNBALANCED DATA - HYBRID

In [20]:

```
#hybrid sampling
#undersampling on the majority class to reduce its size and oversampling on the minority class to increase its size
under=RandomUnderSampler()
x_res1,y_res1=under.fit_resample(X,y)
print("UnderSampling:",x_res1.shape,y_res1.shape)
print(y_res1.value_counts())
over=RandomOverSampler()
x_res2,y_res2=over.fit_resample(x_res1,y_res1)
print("OverSampling:",x_res2.shape,y_res2.shape)
print(y_res2.value_counts())
```

UnderSampling: (684, 10) (684,)

0 342

1 342

Name: Survived, dtype: int64

OverSampling: (684, 10) (684,)

0 342

1 342

Name: Survived, dtype: int64

Titanic dataset

In [21]:

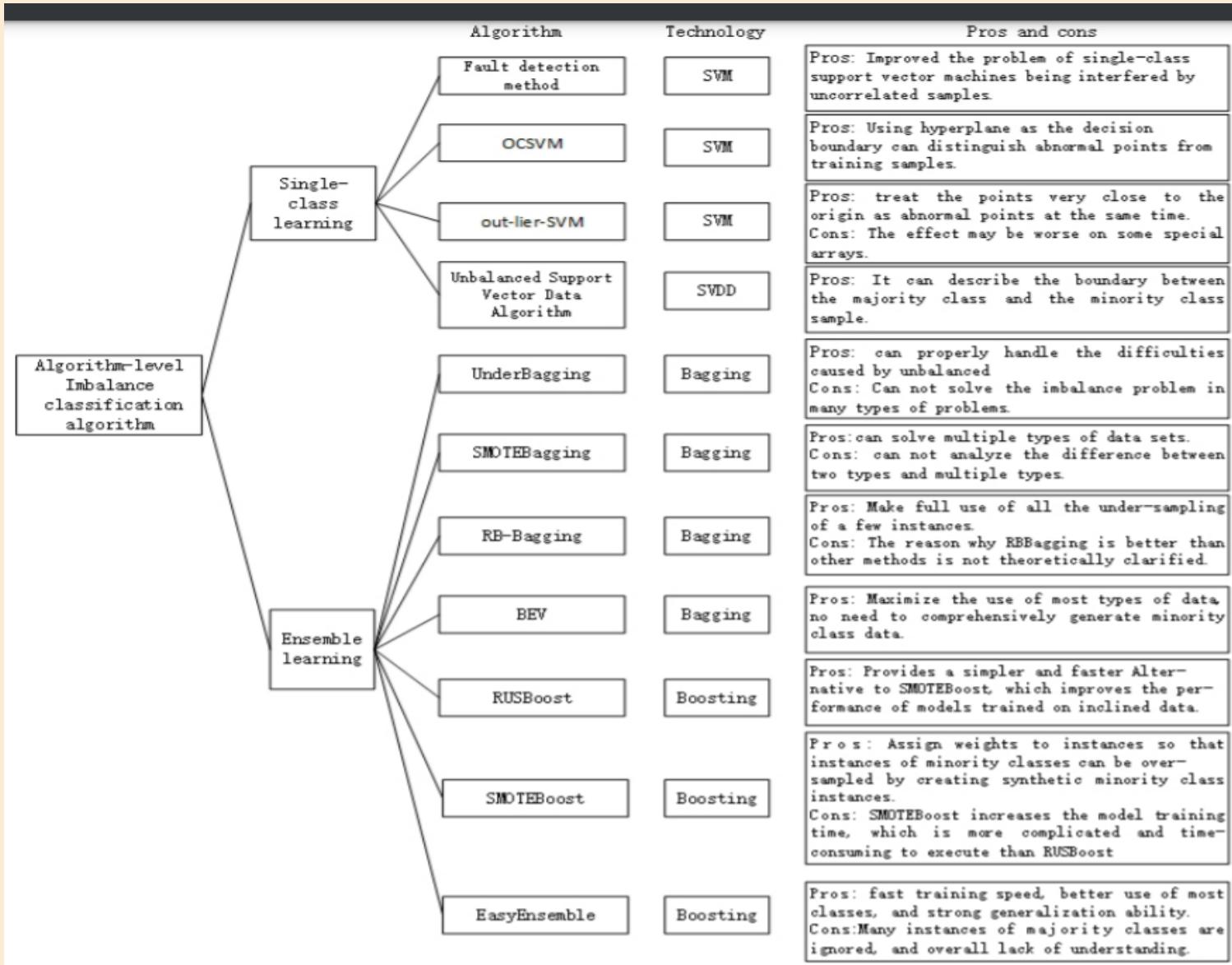
```
#Synthetic Data Generation
#Generate synthetic samples for the minority class using techniques like SMOTE (Synthetic Minority Over-sampling Techni
from imblearn.over_sampling import SMOTE
smote = SMOTE(sampling_strategy='minority')
X_s, y_s = smote.fit_resample(X, y)
print(y_s.value_counts())
```

0 549

1 549

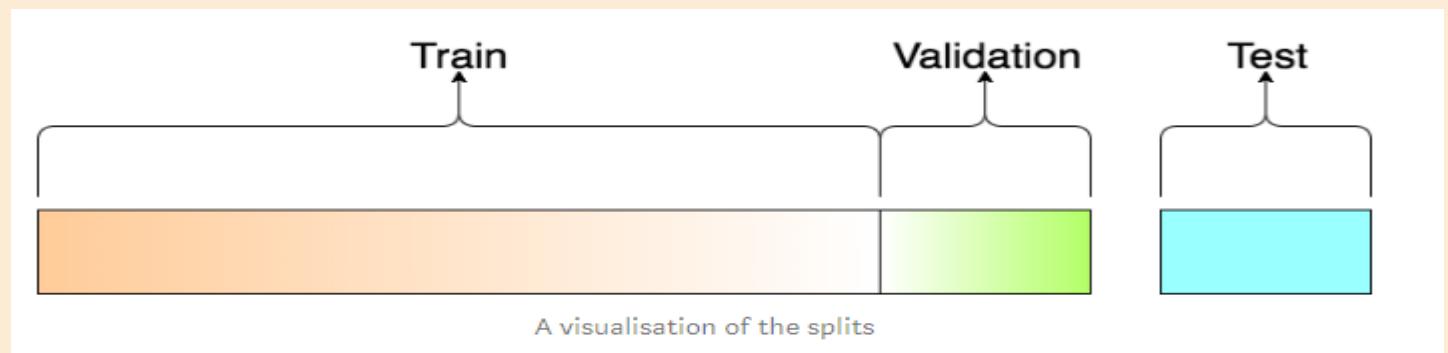
Name: Survived, dtype: int64

UMBALANCED DATA : AUTRES TECHNIQUES



SPLITTING

- Après avoir collecté les données et échantillonné si nécessaire, l'étape suivante consiste à fractionner les données en ensembles d'apprentissage, de validation et de test.
- Plusieurs fois, le jeu de validation est utilisé comme jeu de test, mais ce n'est pas une bonne pratique. L'ensemble de test est généralement bien organisé. Il contient des données soigneusement échantillonnées couvrant les différentes classes auxquelles le modèle serait confronté, lorsqu'il est utilisé dans le monde réel.



SPLITTING

HOLDOUT STRATEGY

1 Split your data into train / validation / test



2 For each parameter combination

Parameter (e.g., depth) A
6 7 8 9 10 11 12 13 14 15 16 17
Parameter B (e.g., n trees)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

TRAIN A MODEL

COMPUTE METRIC ON VALIDATION SET

VALIDATION METRIC

3 Choose the parameter combination with the best metric

A 6 14 B

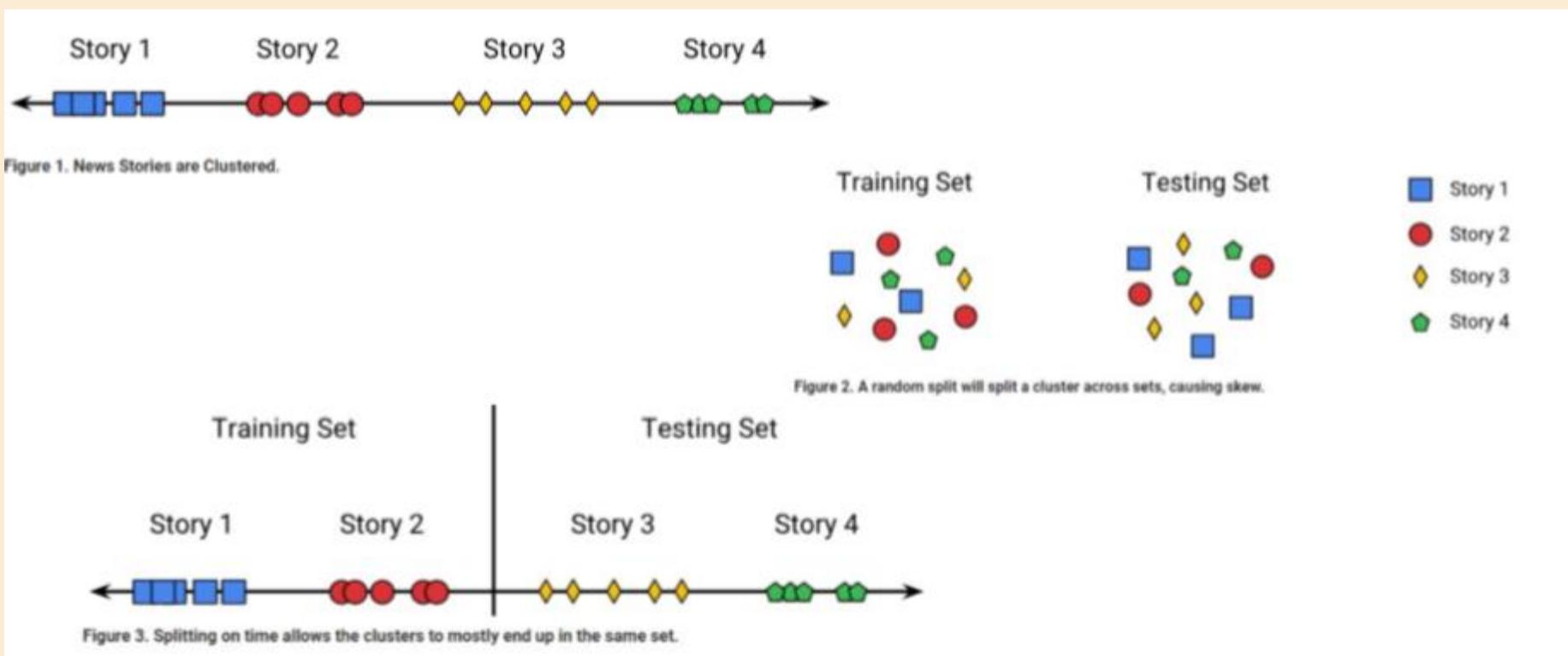
Retrain model on all training data

Compute metric on test set

TEST METRIC
(can compare with other models)

SPLITTING

- Bien que le fractionnement aléatoire soit la meilleure approche pour de nombreux problèmes de ML, ce n'est pas toujours la bonne solution. Par exemple, considérons des ensembles de données dans lesquels les exemples sont naturellement regroupés en exemples similaires.



SPLITTING

- ✖ La procédure de répartition du jeu de données en ces 3 blocs dépend principalement de 2 choses. Tout d'abord, le nombre total d'échantillons et du modèle à construire.
- ✖ Certains modèles nécessitent des données importantes sur lesquelles s'entraîner. Dans ce cas, il faut envisager une base de test assez volumineuse.
- ✖ Les modèles avec très peu d'hyper paramètres sont faciles à valider et à ajuster. Dans ce cas, on peut réduire la taille de notre jeu de validation. Toutefois, si le modèle comporte de nombreux hyper paramètres, Il faut disposer d'un jeu de validation volumineux (on peut également envisager une validation croisée).

SPLITTING

- ✖ Hyper-paramètres:
 - + Défini des concepts de niveau supérieur sur le modèle, tels que la complexité ou la capacité d'apprentissage.
 - + Ne peut pas être appris directement à partir des données dans le processus d'apprentissage des modèles standard et doit être prédéfini.
 - + Peut être décidé en définissant différentes valeurs, en formant différents modèles et en choisissant les valeurs qui testent mieux.

Quelques exemples d'hyper-paramètres:

- + Nombre de feuilles ou profondeur d'un arbre de décision
- + Nombre de facteurs latents dans une factorisation matricielle
- + Taux d'apprentissage (dans de nombreux modèles)
- + Nombre de couches cachées dans un réseau de neurones profonds
- + Nombre de clusters dans un cluster k-means ...

SPLITTING

Dans l'ensemble, comme dans bien d'autres aspects du data mining, le ratio de fractionnement train-test-validation est également très spécifique aux différents cas d'utilisation et il est plus en plus facile de juger sur le meilleur ratio avec plus d'entraînement et de construction de modèles.

VALIDATION CROISÉE

- ✖ Une méthode plus sophistiquée est la validation croisée . 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 le modèle $AB_{A \cup B}$ avec $A \cup B$ et on mesure son taux d'erreur sur C (c'est-à-dire, le nombre d'exemples de C dont la classe est mal prédite par $AB_{A \cup B}$) : E_C .
- ✖ Ensuite, on construit le modèle $BC_{B \cup C}$ avec $B \cup C$ et on mesure l'erreur sur A : E_A .
- ✖
- ✖ Le taux d'erreur E est alors estimé par la moyenne de ces trois erreurs

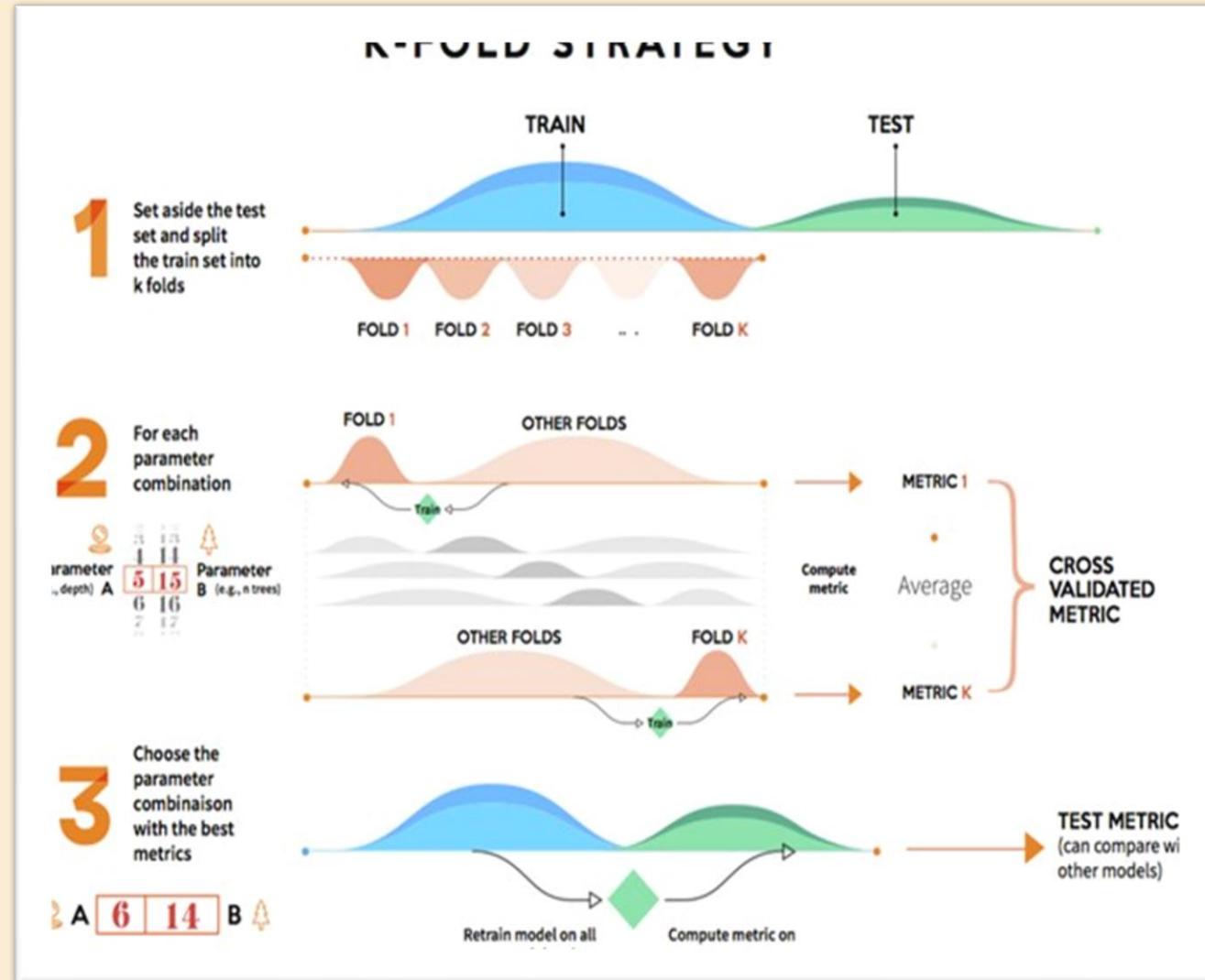
$$E = E_A + E_B + E_C$$

Remarque : Habituellement, on prend $n = 10$. Cette méthode est dénommée validation croisée en n-plis (n-fold cross-validation).

VALIDATION CROISÉE

K-Fold Cross Validation

Les données sont divisées en k sous-ensembles et la méthode d'exclusion est répétée k fois, de sorte qu'à chaque fois, l'un des k sous-ensembles est utilisé comme ensemble de validation et les autres $k-1$ sous-ensembles sont rassemblés pour former un entraînement ensemble



STRATIFIED K-FOLD CROSS VALIDATION

La validation croisée stratifiée de k fois qui est utilisée lorsqu'il y a un déséquilibre important dans les variables de réponse.

Par exemple, en cas de classification, il peut y avoir plusieurs fois plus d'échantillons négatifs que d'échantillons positifs. Pour de tels problèmes, une légère variation dans la technique de validation croisée K Fold est faite, de sorte que à chaque fois on a approximativement le même pourcentage d'échantillons de chaque classe cible que l'ensemble complet, ou en cas de problèmes de prédiction, la valeur de réponse moyenne est approximativement égal dans tous les plis. Cette variation est également connue sous le nom de pli en K stratifié.



TECHNIQUE DE BOOTSTRAP

- Le jeu d'apprentissage est constitué en effectuant N tirages avec remise parmi l'ensemble des exemples. Cela entraîne que certains exemples du jeu d'apprentissage seront vraisemblablement sélectionnés plusieurs fois, et donc que d'autres ne le seront jamais. En fait, la probabilité qu'un certain exemple ne soit jamais tiré est simplement :

$$(1 - \frac{1}{N})^N$$

- La limite quand $N \rightarrow +\infty$ de cette probabilité est $e^{-1} \approx 0,368$. Les exemples qui n'ont pas été sélectionnés constituent le jeu de test.
- Le jeu d'apprentissage contient 63,2 % des exemples du jeu d'exemples initial (en moyenne). L'erreur est calculée en combinant l'erreur d'apprentissage E_{app} et l'erreur de test est par la formule suivante :

$$E = 0,632 \times E_{test} + 0,368 \times E_{app}$$

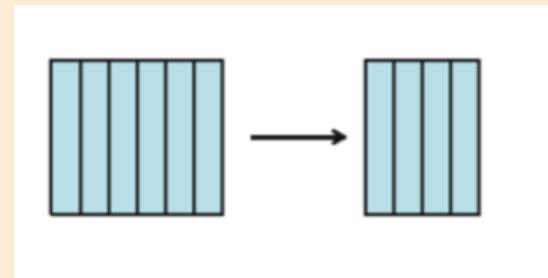
Ensuite, cette procédure est itérée plusieurs fois et une erreur moyenne est calculée.

Réduction de dimension

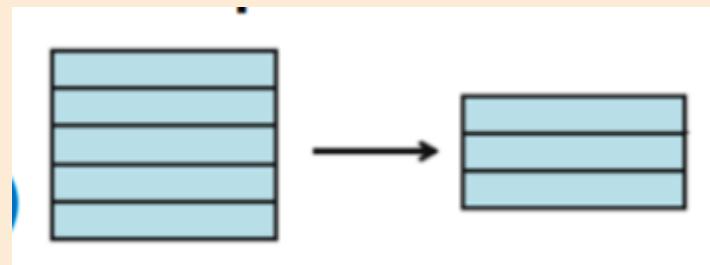
REDUCTION DE DIMENSION

Quels sont les problèmes fondamentaux qui doivent être résolus dans les réductions de dimensionnalité?

- ✖ Réduire la dimension des données
 - + Feature selection

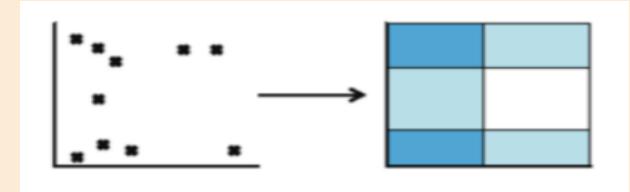


- ✖ Comment supprimer les exemples redondants et/ou à conflits ?
 - + élection d'instance (sélection de prototype vs sélection d'ensemble d'entraînement)

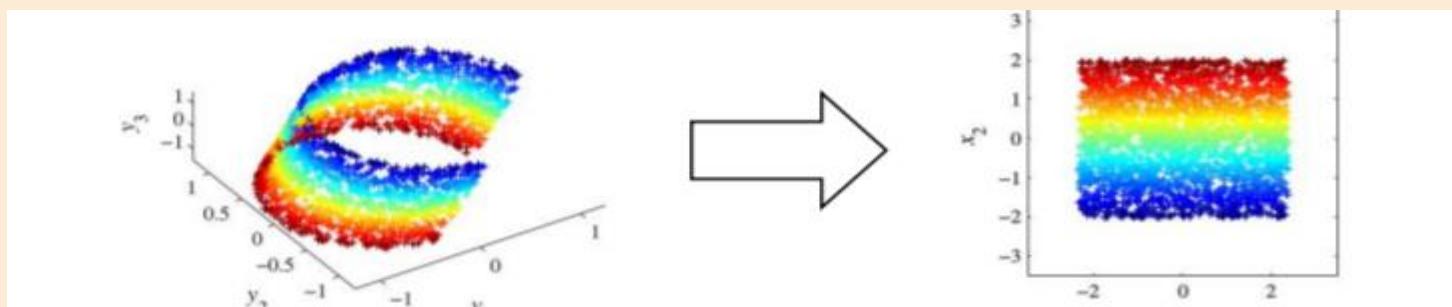


REDUCTION DE DIMENSION

- Comment simplifier le domaine d'un attribut ?
 - Discrétisation
 - Diviser l'intervalle numérique (continu ou non) en intervalles.
 - Mémoriser les étiquettes des intervalles.
 - Il est crucial pour les règles d'association et certains algorithmes de classification, qui n'acceptent que des données discrètes.



- Comment combler les lacunes dans les données ?
 - Extraction de caractéristiques



REDUCTION DE DIMENSION

La réduction de dimension est une technique de réduction de l'espace des caractéristiques pour obtenir un modèle d'apprentissage automatique stable et statistiquement solide, évitant les problème dûs à la dimension. Il existe principalement deux approches pour effectuer la réduction de dimensionnalité : la sélection de caractéristiques et la transformation de caractéristiques.

Feature Selection : Obtenir un sous-ensemble des fonctionnalités importantes et de supprimer les fonctionnalités colinéaires ou moins importantes.

Réduction de dimensionnalité: est également connue sous le nom d'extraction de caractéristiques, a pour objectif d'essayer de projeter les données de grande dimension dans des dimensions inférieures.

FEATURE SELECTION

- ✖ Les attributs sont essentiellement toutes les dimensions présentes dans les données. Mais tous, au format brut, représentent-ils les connaissances sous-jacentes que nous souhaitons apprendre de la meilleure façon possible?.
- ✖ La sélection est un autre élément clé du processus d'apprentissage automatique. De ce fait, il est important de considérer la sélection des fonctionnalités dans le processus de sélection du modèle. Si vous ne le faites pas, vous pouvez introduire par inattention un biais dans vos modèles, ce qui peut entraîner un sur-apprentissage

FEATURE SELECTION

- ✖ le problème de sélection des fonctionnalités (SC) ou variables (Feature Subset Selection, FSS) consiste à trouver un sous-ensemble des variables qui optimise la probabilité du classement correct
- ✖ Pourquoi la sélection de variables est-elle nécessaire?
 - + Plus d'attributs, moins de succès dans le classement
 - + Travailler avec moins de variables minimise la complexité du problème et diminuer le temps d'exécution
 - + Avec moins de variables la possibilité de généraliser augmente

FEATURE SELECTION

- ✖ Feature selection est différente de la réduction de dimensionnalité. Les deux méthodes cherchent à réduire le nombre d'attributs dans le jeu de données, mais une méthode de Réduction de dimensionnalité consiste à créer de nouvelles combinaisons d'attributs alors que les méthodes de sélection d'entités incluant et excluant des attributs présents dans les données sans les modifier.
- ✖ Elle consiste en l'utilisation de certains algorithmes pour sélectionner automatiquement un sous-ensemble de vos attributs d'origine.
- ✖ Ici, on ne crée pas / ne modifie pas les attributs actuelles, mais les élague plutôt pour réduire le bruit / la redondance.

FEATURE SELECTION

- ✖ La sélection des attributs est elle-même utile, mais elle agit principalement comme un filtre, en désactivant les attributs qui ne sont pas utiles. Les méthodes de sélection des caractéristiques peuvent être utilisées pour identifier et supprimer les attributs inutiles, non pertinents et redondants des données qui ne contribuent pas à la précision d'un modèle prédictif ou qui peuvent en fait diminuer la précision du modèle.
- ✖ Elle permet de :
 - + Améliorer les performances de prévision des prédicteurs,
 - + Fournir des prédicteurs plus rapides et plus rentables,
 - + Mieux comprendre le processus sous-jacent que génère les données.

FEATURE SELECTION

La **sélection de caractéristiques** (Feature Selection) est une étape clé dans un projet de **Data Mining**. Elle vise à identifier les variables les plus importantes (ou significatives) pour la construction du modèle, tout en réduisant la complexité et en améliorant les performances.

FEATURE SELECTION

- ✖ Il existe trois classes générales d'algorithmes de sélection de caractéristiques:
 - + Méthodes par filtrage,
 - + Méthodes dite wrapper
 - + Méthodes intégrées (embadded).

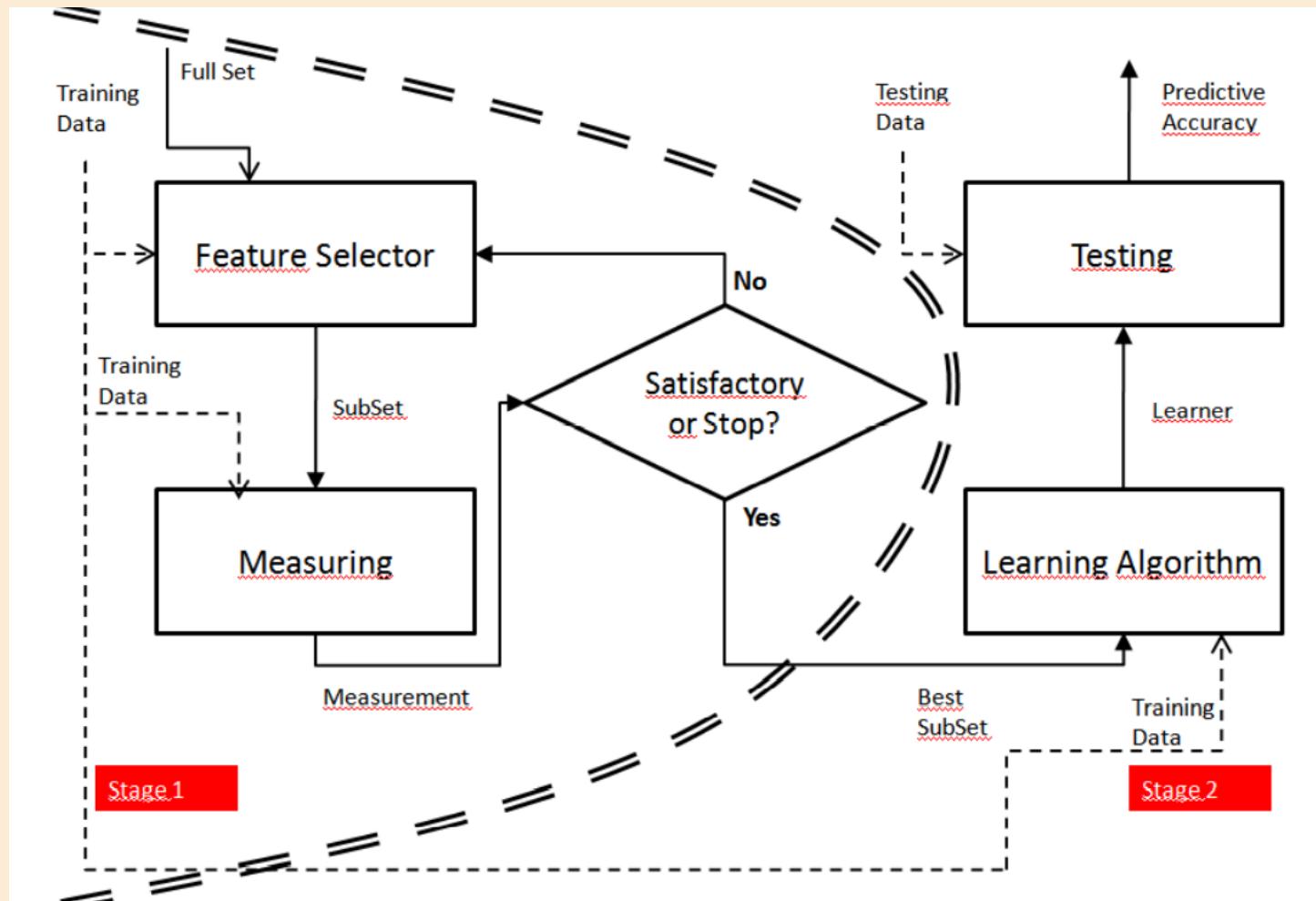
FEATURE SELECTION

- ✖ Les méthodes de sélection d'attribut à base de filtre appliquent une mesure statistique pour attribuer un score à chaque caractéristique (attribut).
- ✖ Les entités sont classées en fonction du score et sélectionnées pour être conservées ou supprimées de l'ensemble de données.
- ✖ Les méthodes sont souvent uni-variées et considèrent l'entité indépendamment ou par rapport à la variable dépendante. Exemples : Parmi les exemples de méthodes de filtrage, citons ANOVA, le test du chi2, le gain d'information et le coefficient de corrélation.

MÉTHODE PAR FILTRAGE

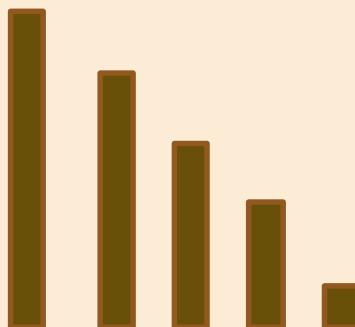
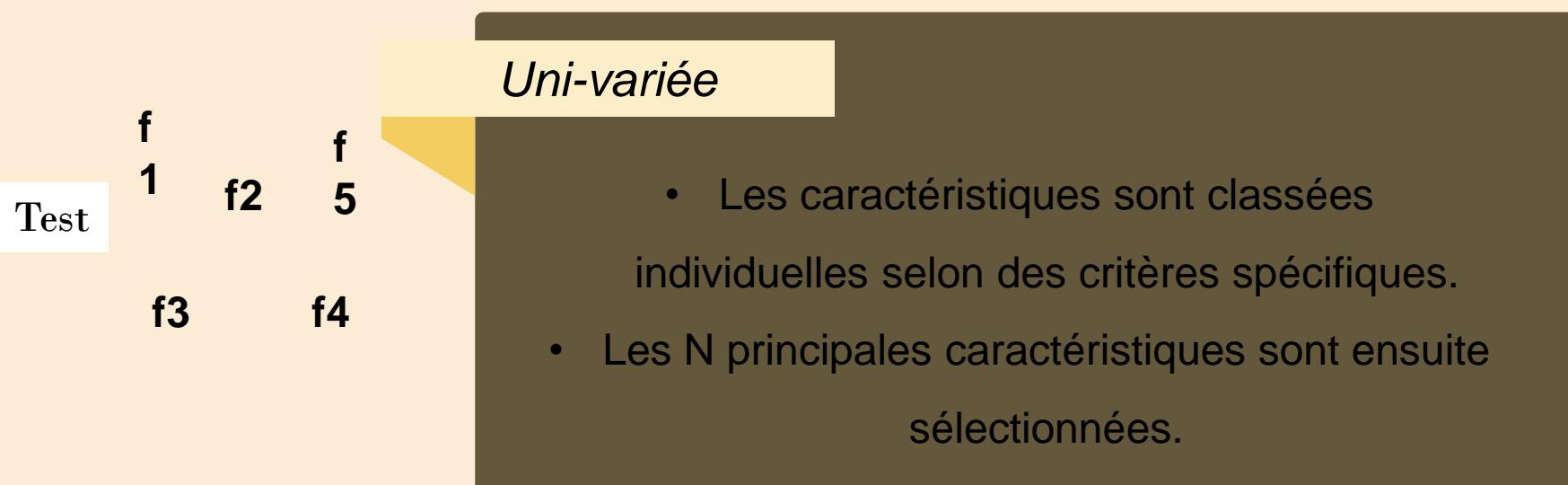
- ✖ Les méthodes de filtrage appartiennent à la catégorie des méthodes de sélection d'entités qui sélectionnent des entités indépendamment du modèle d'algorithme d'apprentissage automatique.
- ✖ C'est l'un des plus grands avantages des méthodes de filtrage.
- ✖ Les fonctionnalités sélectionnées à l'aide de méthodes de filtrage peuvent être utilisées comme entrée dans tous les modèles d'apprentissage automatique.
- ✖ Un autre avantage des méthodes de filtrage est qu'elles sont très rapides.

MÉTHODE PAR FILTRAGE



Processus

MÉTHODE PAR FILTRAGE



MÉTHODE PAR FILTRAGE - UNIVARIÉE

ANOVA

Definition

Anova effectue pour chaque entité une analyse de la variance où la variable de classe est expliquée par l'entité. La valeur statistique F est utilisée comme score.

$$F = \frac{\text{Variance between groups}}{\text{Variance within groups}}$$

MÉTHODE PAR FILTRAGE - UNIVARIÉE

Variance $\sigma^2 = \frac{\sum(x-\mu)^2}{N}$

Age	40	47	44	46	45	42	43	43,85	4,97	
Weight	105	73	80	43	68	55	99	74,71	425,34	
					Mean	Variance				

MÉTHODE PAR FILTRAGE - UNIVARIÉE

Remarque

L'un des principaux inconvénients des méthodes de filtrage uni-varié est qu'elles peuvent sélectionner des fonctionnalités redondantes car la relation entre les attributs n'est pas prise en compte lors de la prise de décisions..

```
import numpy as np
import pandas as pd
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.datasets import load_iris

# Charger un jeu de données exemple (Titanic dataset)
data = load_titanic()
X = data.data
y = data.target

# Utiliser SelectKBest avec f_classif pour effectuer un test ANOVA
selector = SelectKBest(f_classif, k=5) # Sélectionner les 5 meilleures caractéristiques
X_new = selector.fit_transform(X, y)

# Afficher les indices des caractéristiques sélectionnées
selected_features = selector.get_support(indices=True)
print("Indices des caractéristiques sélectionnées par ANOVA-like:")
print(selected_features)

# Afficher les scores ANOVA pour chaque caractéristique
print("Scores ANOVA pour chaque caractéristique :")
print(selector.scores_)
```

Cas Réel sur Titanic avec 5 caractéristiques

In [28]:

```
from sklearn.feature_selection import f_classif
# Filter method - SelectKBest with f_classif
fvalue_selector = SelectKBest(f_classif, k=5)
X_kbest_fvalue = fvalue_selector.fit_transform(X_s, y_s)
selected_features = X_s.columns[fvalue_selector.get_support(indices=True)]
print("Selected Features using f_classif:")
print(selected_features)
```

```
Selected Features using f_classif:
Index(['Pclass', 'Fare', 'Sex_female', 'Sex_male', 'Embarked_C'], dtype='object')
```

MÉTHODE PAR FILTRAGE - MULTI-VARIÉES

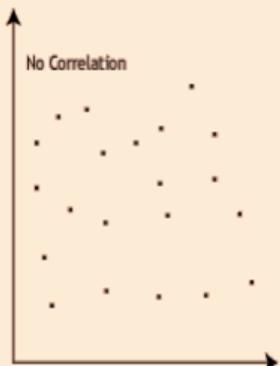
Definition

Les méthodes de filtrage multi-variées sont capables de supprimer les fonctionnalités redondantes des données puisqu'elles prennent en compte la relation mutuelle entre les fonctionnalités. Des méthodes de filtrage multi-variées peuvent être utilisées pour supprimer les entités dupliquées et corrélées des données.

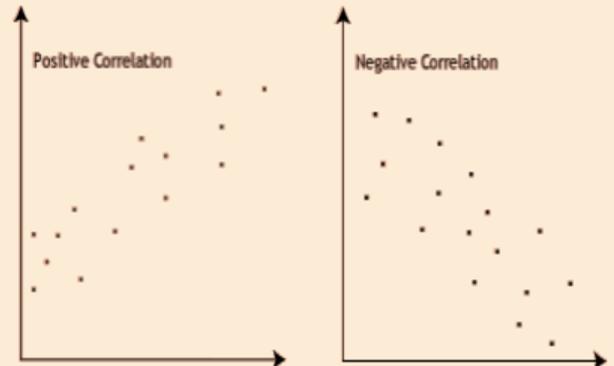
MÉTHODE PAR FILTRAGE - MULTI-VARIÉES

Pearson Correlation

Les entités à forte corrélation sont plus linéairement dépendantes et ont donc presque le même effet sur la variable dépendante. Ainsi, lorsque deux fonctionnalités ont une corrélation élevée, nous pouvons supprimer l'une des deux



$$r = \frac{\text{Covariance between the two features}}{\text{Variance of feature } a * \text{variance of feature } b}$$



```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Charger le dataset Titanic de seaborn (vous pouvez aussi utiliser votre propre
dataset)
titanic = sns.load_dataset('titanic')

# Sélectionner uniquement les colonnes numériques pour la corrélation
numeric_cols = titanic.select_dtypes(include=['float64', 'int64']).columns

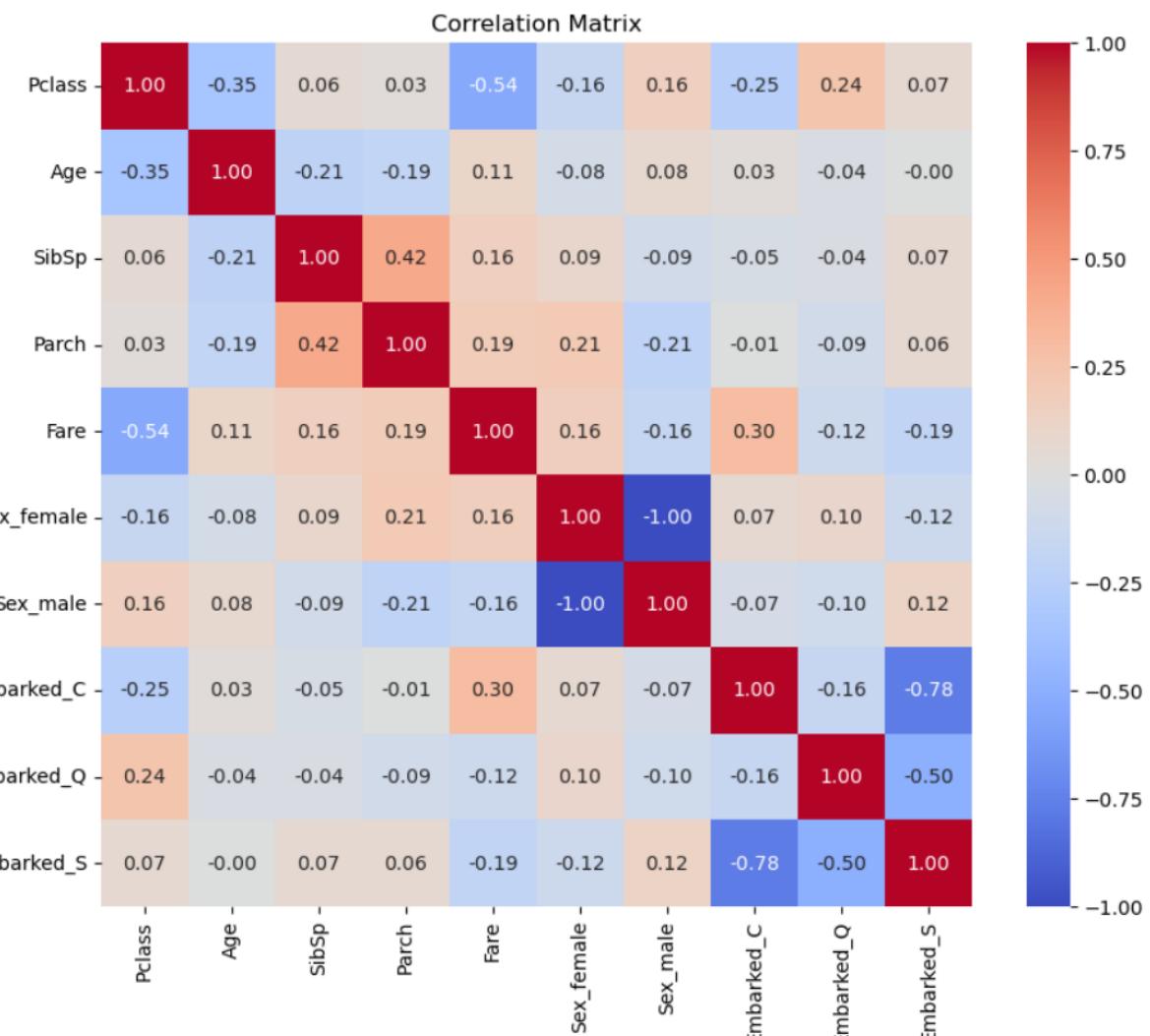
# Calculer la matrice de corrélation de Pearson
correlation_matrix = titanic[numeric_cols].corr(method='pearson')

# Afficher la matrice de corrélation
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f',
cbar=True)
plt.title('Matrice de Corrélation - Titanic Dataset')
plt.show()
```

```

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Calculate the correlation matrix
correlation_matrix = X_s.corr()
# Plot the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()

```



In [26]:

```
correlation_matrix = X_s.corr()
# Set the threshold for correlation
threshold = 0.4
# Find and display the correlated features
correlated_features = set()
for i in range(len(correlation_matrix.columns)):
    for j in range(i):
        if abs(correlation_matrix.iloc[i, j]) > threshold:
            colname = correlation_matrix.columns[i]
            correlated_features.add(colname)

print("Correlated Features:")
print(correlated_features)
```

Correlated Features:

```
{'Fare', 'Embarked_S', 'Parch', 'Sex_male'}
```

MÉTHODE PAR FILTRAGE - MULTI-VARIÉES

Test de χ^2

Le **test du Chi-deux (χ^2)** est une méthode statistique utilisée pour déterminer si deux variables catégorielles sont indépendantes ou non. Il est couramment utilisé dans les tableaux de contingence pour analyser les relations entre les variables. Plus précisément, le test χ^2 compare la fréquence observée d'événements à la fréquence attendue sous l'hypothèse d'indépendance.

Hypothèses du test du Chi-deux :

Hypothèse nulle (H_0) : Les variables sont indépendantes.

Hypothèse alternative (H_1) : Les variables ne sont pas indépendantes.

Formule

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Où :

- O_i est la fréquence observée dans chaque cellule.
- E_i est la fréquence attendue sous l'hypothèse d'indépendance.

In [27]:

```
#from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
# Apply SelectKBest for feature selection
k_best_selector = SelectKBest(chi2, k=5) # You can set k to the number of top features you want to select
X_selected = k_best_selector.fit_transform(X_s, y_s)
# Display selected features
selected_features = X_s.columns[k_best_selector.get_support(indices=True)]
print("Selected Features:")
print(selected_features)
```

Selected Features:

```
Index(['Pclass', 'Fare', 'Sex_female', 'Sex_male', 'Embarked_C'], dtype='object')
```

FEATURE SELECTION : WRAPPERS

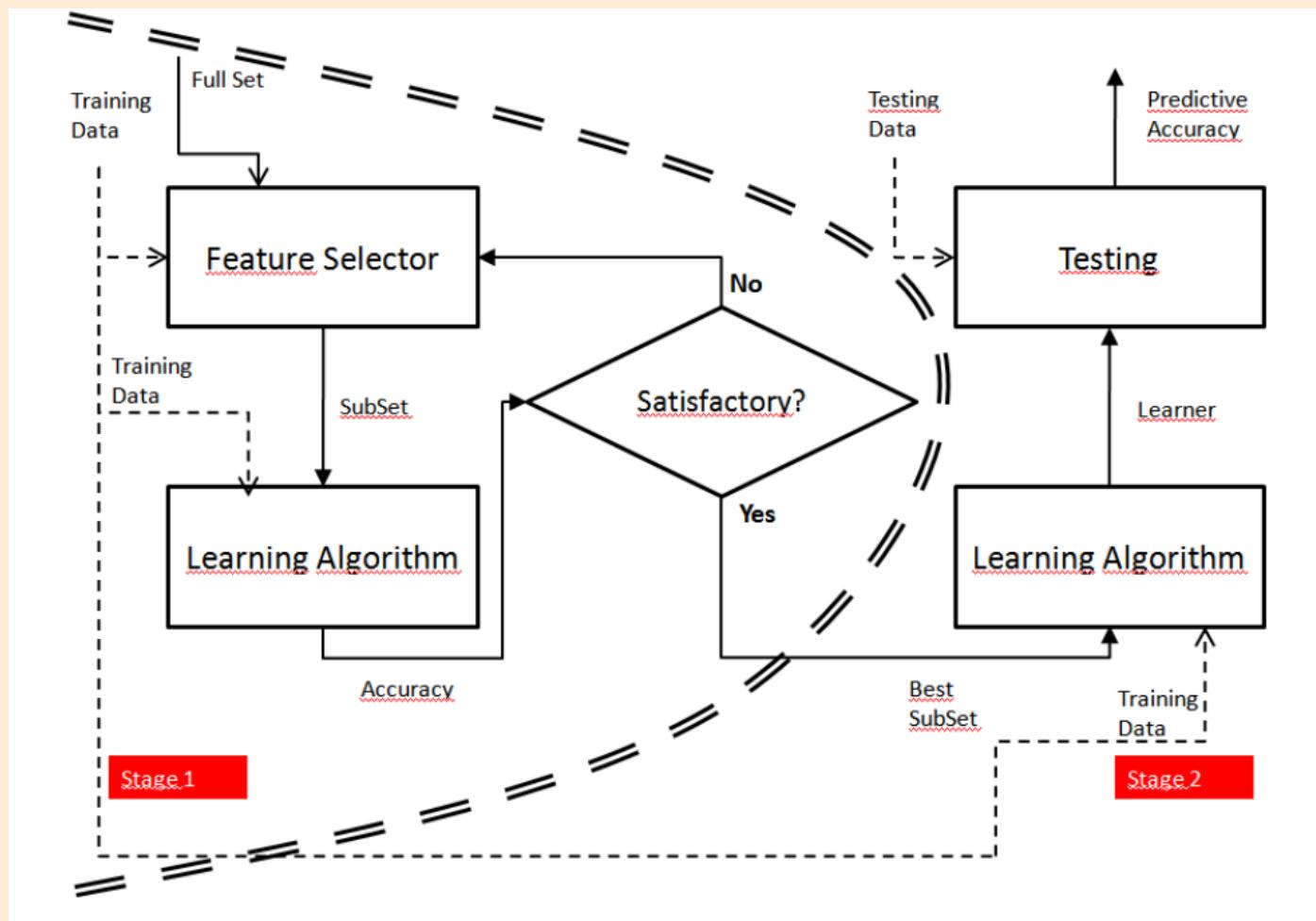
- Dans certains scénarios, vous pouvez utiliser un algorithme d'apprentissage automatique spécifique pour former votre modèle. Dans de tels cas, les caractéristiques sélectionnées à l'aide de méthodes de filtrage peuvent ne pas constituer l'ensemble de caractéristiques optimal pour cet algorithme spécifique.
- Il existe une autre catégorie de méthodes de sélection des fonctionnalités qui sélectionne les fonctionnalités les plus optimales pour l'algorithme spécifié. De telles méthodes sont appelées méthodes wrapper.
- Les méthodes wrappers considèrent la sélection d'un ensemble de caractéristique comme un problème de recherche, dans lequel différentes combinaisons sont préparées, évaluées et comparées à d'autres combinaisons.

FEATURE SELECTION : WRAPPERS

- ✖ Les méthodes d'encapsulation sont basées sur des algorithmes de recherche (greedy search), car elles évaluent toutes les combinaisons possibles d'attributs et sélectionnent la combinaison qui produit le meilleur résultat pour un algorithme d'apprentissage automatique spécifique.
- ✖ Un inconvénient de cette approche est que le test de toutes les combinaisons possibles des fonctionnalités peut s'avérer très coûteux en calcul, en particulier si le nombre d'attribut est très grand.
- ✖ Comme indiqué précédemment, les méthodes d'encapsulation peuvent trouver le meilleur ensemble de fonctionnalités pour un algorithme spécifique. Cependant, l'inconvénient est que ces fonctionnalités ne sont peut-être pas optimales pour tous les autres algorithmes d'apprentissage automatique.

FEATURE SELECTION : WRAPPERS

Processus



FEATURE SELECTION : WRAPPERS

- ✖ Les méthodes d'encapsulation pour la sélection des fonctionnalités peuvent être divisées en trois catégories: sélection de fonctionnalités en avant, sélection de fonctionnalités en arrière et sélection de fonctionnalités exhaustive.
- ✖ **Forward feature selection** : Dans la première phase de la sélection, les performances du classifieur sont évaluées par rapport à chaque caractéristique. L'attribut le plus performant est sélectionnée parmi tous les attributs.
- ✖ Dans la deuxième étape, le premier attribut est essayée en combinaison avec tous les autres. La combinaison de deux fonctionnalités offrant les meilleures performances en termes d'algorithme est sélectionnée.
- ✖ Le processus se poursuit jusqu'à ce que le nombre spécifié d'entités soit sélectionné.

FEATURE SELECTION : WRAPPERS

- ✖ Backwards feature selection : comme son nom l'indique, c'est exactement le contraire de la première que nous avons étudiée.
- ✖ Lors de la première étape de la sélection, une fonctionnalité est supprimée du jeu de donnée en alternance et les performances du classificateur sont évaluées.
- ✖ L'ensemble de fonctionnalités offrant les meilleures performances est conservé. Dans la deuxième étape, un attribut est à nouveau supprimé à tour de rôle et les performances de toutes les combinaisons de fonctionnalités à l'exception des 2 fonctionnalités sont évaluées.
- ✖ Ce processus se poursuit jusqu'à ce que le nombre spécifié d'entités reste dans l'ensemble de données.

FEATURE SELECTION : WRAPPERS

Sequential Feature Selector (SFS) est une technique de sélection de caractéristiques (features) qui consiste à ajouter ou supprimer progressivement des caractéristiques dans le but d'optimiser une certaine mesure de performance d'un modèle. Cette méthode est particulièrement utilisée pour améliorer l'efficacité des modèles en réduisant la dimensionnalité, ce qui peut conduire à une meilleure généralisation et à un meilleur temps de calcul. Il y'a deux types de **Sequential Feature Selector** :

- ✖ **Forward Selection (Sélection avant)** : Commence avec un modèle vide (aucune caractéristique) et ajoute les caractéristiques les plus pertinentes une par une. À chaque itération, la caractéristique qui améliore le mieux la performance du modèle est ajoutée.
- ✖ **Backward Selection (Sélection arrière)** : Commence avec l'ensemble complet des caractéristiques et les enlève progressivement. À chaque itération, la caractéristique qui dégrade le moins la performance du modèle est supprimée.

FEATURE SELECTION : WRAPPERS

- ✖ Forward Selection using sequentiel feature selection sur Titanic dataset

```
In [30]:
```

```
from sklearn.model_selection import train_test_split
# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X_s, y_s, test_size=0.2, random_state=42)
```

```
In [31]:
```

```
from sklearn.feature_selection import SequentialFeatureSelector
from sklearn.metrics import accuracy_score
# Forward Feature Selection using SequentialFeatureSelector
model = RandomForestClassifier(random_state=42)
sfs_forward = SequentialFeatureSelector(model, direction='forward', n_features_to_select=None, cv=5)
X_train_sfs_forward = sfs_forward.fit_transform(X_s, y_s)
# Display selected features (forward selection)
print("Selected Features (Forward Selection):", X_s.columns[sfs_forward.support_])
```

```
Selected Features (Forward Selection): Index(['Pclass', 'Sex_female', 'Sex_male', 'Embarked_C', 'Embarked_Q'], dtype='object')
```

FEATURE SELECTION : WRAPPERS

- ✖ Backword feature Selection using sequentiel feature selection sur Titanic dataset

```
In [33]:
```

```
# Backward Feature Selection using SequentialFeatureSelector
sfs_backward = SequentialFeatureSelector(model, direction='backward', n_features_to_select=None, cv=5)
X_train_sfs_backward = sfs_backward.fit_transform(X_s, y_s)
print("Selected Features (Backward Selection):", X_s.columns[sfs_backward.support_])

Selected Features (Backward Selection): Index(['Pclass', 'Age', 'Fare', 'Sex_female', 'Sex_male'], dtype='object')
```

FEATURE SELECTION : WRAPPERS

- ✖ Inconvénients de ces deux techniques :
 - + Impossible de supprimer un attribut qui peut devenir inutiles après l'ajout d'autres fonctionnalités pour FS.
 - + Impossible de réévaluer l'utilité d'un attribut après qu'elle a été supprimée pour BS.
- ✖ Sélection avant : Fonctionne mieux lorsque le sous-ensemble optimal a peu de variables
- ✖ Sélection arrière : Fonctionne mieux lorsque le sous-ensemble optimal a plusieurs variables

FEATURE SELECTION : WRAPPERS

- ✖ **Exhaustive feature selection** : Dans ce cas, les performances d'un algorithme d'apprentissage automatique sont évaluées par rapport à toutes les combinaisons possibles des caractéristiques du jeu de données.
- ✖ Le sous-ensemble de fonctionnalités offrant les meilleures performances est sélectionné.
- ✖ L'algorithme de recherche exhaustive est l'algorithme le plus glouton(greedy) de toutes les méthodes d'encapsulation, car il essaie toutes les combinaisons de fonctionnalités et sélectionne la meilleure.
- ✖ Un inconvénient de la sélection exhaustive des fonctionnalités est qu'elle peut être plus lente que les autres, car elle évalue toutes les combinaisons d'entités.

FEATURE SELECTION : WRAPPERS

Plus-L minus-R selection (LRS)

LRS

LRS combine l'idée de FS et BS. Elle démarre à partir d'un ensemble vide, en ajoutant d'abord des fonctionnalités L à chaque tour, puis en supprimant les fonctionnalités R afin que la valeur d'évaluation de la métrique soit optimale. \$

FEATURE SELECTION : WRAPPERS

Sequential Floating Selection

A été introduite pour traiter le problème d'imbrication.

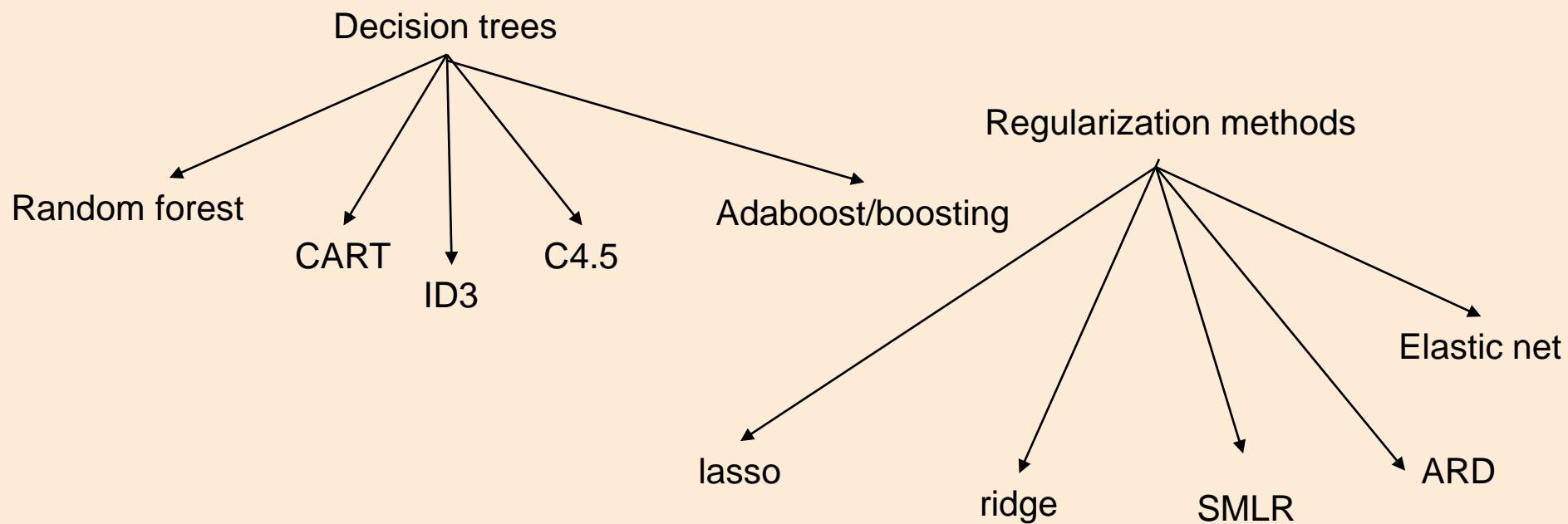
- Le meilleur sous-ensemble de fonctionnalités, T, est initialisé en tant qu'ensemble vide et à chaque étape, une nouvelle fonctionnalité est ajoutée.
- Après cela, l'algorithme recherche les caractéristiques qui peuvent être supprimées de T jusqu'à ce que l'erreur de classification correcte n'augmente pas.
- Cet algorithme est une combinaison des méthodes séquentielle avant et séquentielle arrière. Le « meilleur sous-ensemble » de caractéristiques est construit en fonction de la fréquence à laquelle chaque attribut est sélectionné dans le nombre de répétitions donné.
- En raison de la complexité temporelle de l'algorithme, son utilisation n'est pas recommandée pour les ensembles de données avec un grand nombre d'attributs (disons plus de 1000).

FEATURE SELECTION : INTÉGRÉES

- Dans les méthodes intégrées, l'algorithme de sélection de caractéristiques est intégré dans le cadre de l'algorithme d'apprentissage.
- Les méthodes intégrées combinent les qualités des méthodes filter et wrapper. Il est implémenté par des algorithmes qui ont leurs propres méthodes de sélection de caractéristiques.
- Un algorithme d'apprentissage tire parti de son propre processus de sélection de variables et effectue simultanément une sélection de caractéristiques et une classification/régression.
- La technique intégrée la plus courante est l'algorithme d'arbre comme RandomForest, ExtraTree et ainsi de suite.
- D'autres méthodes intégrées sont le LASSO avec la pénalité L1 et Ridge avec la pénalité L2 pour la construction d'un modèle linéaire. Ces deux méthodes réduisent de nombreuses fonctionnalités à zéro ou presque à zéro.

FEATURE SELECTION : INTÉGRÉES

- Les méthodes intégrées déterminent quelles fonctionnalités contribuent le plus à la précision du modèle lors de sa création.



FEATURE SELECTION : INTÉGRÉES

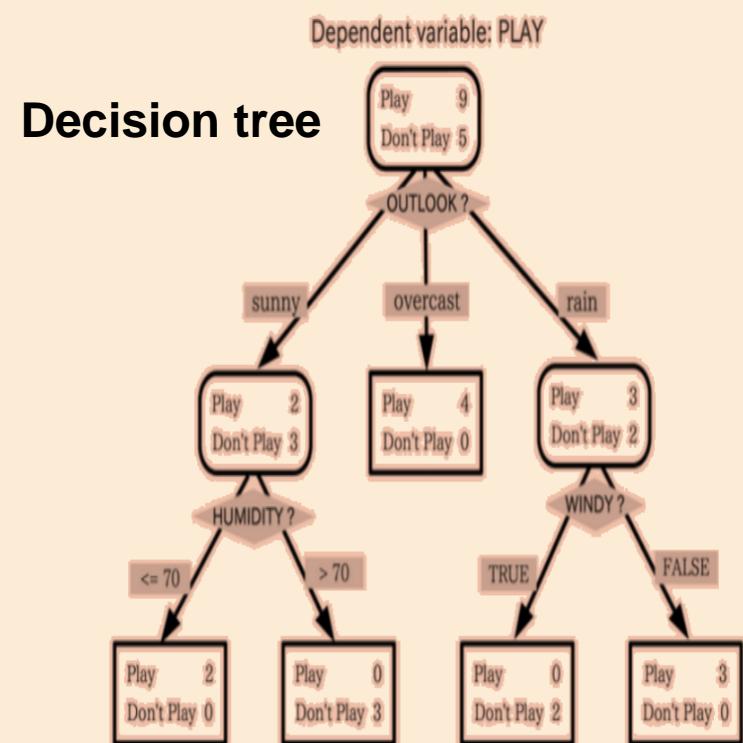
DECISION TREES FOR FEATURE SELECTION

Entropy !

GINI
Index
!

Information
gain !

- Partitionnement récursif de l'espace.
- À chaque nœud, une nouvelle fonctionnalité est sélectionnée.
- L'implicité des arbres sélectionne de bonnes caractéristiques avec des informations mutuelles.

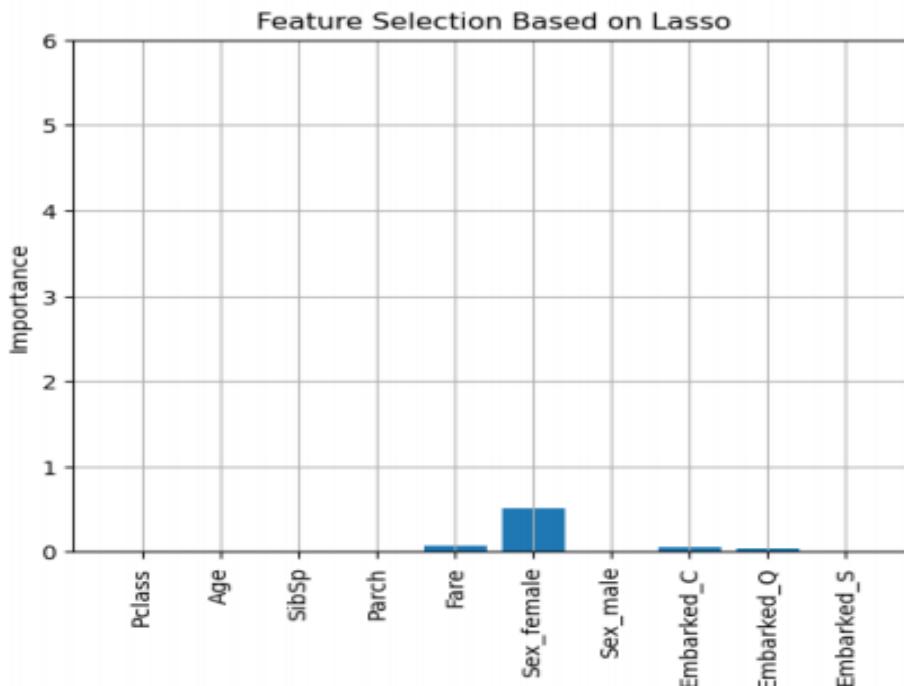


RF POUR FS

- ✖ Les forêts aléatoires sont l'un des algorithmes d'apprentissage automatique les plus populaires.
- ✖ Ils ont un tel succès car ils fournissent en général une bonne performance prédictive, un faible sur-ajustement et une interprétabilité facile.
- ✖ Cette interprétabilité est donnée par le fait qu'il est simple de déduire l'importance de chaque variable sur la décision de l'arbre.
- ✖ En d'autres termes, il est facile de calculer la contribution de chaque variable à la décision.

FEATURE SELECTION : EMBEDDED (INTÉGRÉES)

```
In [34]:  
from sklearn.linear_model import Lasso, Ridge, ElasticNet  
lasso_selector = Lasso(alpha=1e-05)  
lasso_selector.fit(X_s, y_s)  
# Print selected features  
selected_features_lasso = X_s.columns[lasso_selector.coef_ > 0]  
print("Selected Features using LASSO:")  
print(selected_features_lasso)  
  
Selected Features using LASSO:  
Index(['Fare', 'Sex_female', 'Embarked_C', 'Embarked_Q'], dtype='object')  
  
In [35]:  
plt.bar(X_s.columns, lasso_selector.coef_)  
plt.xticks(rotation=90)  
plt.grid()  
plt.title("Feature Selection Based on Lasso")  
plt.xlabel("Features")  
plt.ylabel("Importance")  
plt.ylim(0, 6)  
plt.show()
```



Embedded feature Selection using Lasso sur Titanic dataset

FEATURE SELECTION : EMBEDDED (INTÉGRÉES)

```
In [38]:
```

```
!pip install --upgrade scipy
```

```
Requirement already satisfied: scipy in c:\users\soukaina\anaconda3\lib\site-packages (1.13.1)
Requirement already satisfied: numpy<2.3,>=1.22.4 in c:\users\soukaina\anaconda3\lib\site-packages (from scipy) (1.24.4)
```

```
In [40]:
```

```
ridge = Ridge(alpha=1.0, solver="sag")
ridge.fit(X_s, y_s)
```

```
Out[40]:
```

```
Ridge(solver='sag')
```

```
In [41]:
```

```
# Using np.abs() to make coefficients positive.
selected_features_ridge = X_s.columns[ridge.coef_ > 0]
print("Selected Features using Ridge:")
print(selected_features_ridge)
```

```
Selected Features using Ridge:
```

```
Index(['Fare', 'Sex_female', 'Embarked_C', 'Embarked_Q'], dtype='object')
```

Embedded feature Selection
using Ridge sur Titanic
dataset

FEATURE SELECTION



✖ Quelle méthode choisir :

- ✖ Les méthodes de filtrage se révèlent très utiles lorsque vous avez un jeu de données avec autant d'entités. Vous pouvez l'utiliser comme première étape de votre processus de sélection d'entités afin de vous débarrasser des entités corrélées et constantes. Vous pouvez également l'utiliser si vous ne savez pas avec quel modèle travailler pour votre problème d'apprentissage automatique.
- ✖ Les méthodes wrapper et les méthodes intégrées peuvent être utilisées pour les ensembles de données de moindre dimensionnalité, car si vous utilisez autant de fonctionnalités, cela peut être assez coûteux. Et essayez toujours d'utiliser diverses méthodes dans chaque catégorie avant de décider des fonctionnalités à utiliser.
- ✖ Si la sélection de fonctionnalités ne répond pas à votre problème d'apprentissage automatique, essayez d'utiliser l'extraction de fonctionnalités pour obtenir une nouvelle meilleure représentation des fonctionnalités de données.

LA RÉDUCTION DE DIMENSIONNALITÉ

La réduction de la dimensionnalité, en revanche, crée de nouvelles caractéristiques qui sont des combinaisons ou des projections des caractéristiques d'origine. L'objectif ici est de représenter les données dans un espace de plus faible dimension tout en préservant autant d'information que possible. Cela peut être utile pour simplifier le modèle et éviter le surapprentissage.

Quelques techniques de réduction de dimensionnalité populaires :

- ✖ **Analyse en composantes principales (PCA)** : Une méthode de réduction linéaire qui projette les données dans un nouvel espace en maximisant la variance.
- ✖ **Analyse discriminante linéaire (LDA)** : Utilisée principalement pour la classification, elle cherche à maximiser la séparation entre les classes.
- ✖ **t-SNE et UMAP** : Des méthodes non linéaires utilisées pour la visualisation de données de haute dimension.

LA RÉDUCTION DE DIMENSIONNALITÉ

Réduction de dimensionnalité à 5 composantes sur Titanic dataset

In [48]:

```
from sklearn.decomposition import PCA
# Apply PCA to the scaled numerical features
pca = PCA(n_components=5)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)
```

In [49]:

```
X_train_pca.shape
```

Out[49]:

```
(878, 5)
```