Apprentissage Artificiel
Présentation par les Arbres de Décision

Laurent Miclet

IRISA (Lannion)
France

EMINES 2024

Plan

Principes de l'AA

Apprentissage automatique (machine), *machine learning*.

Définition

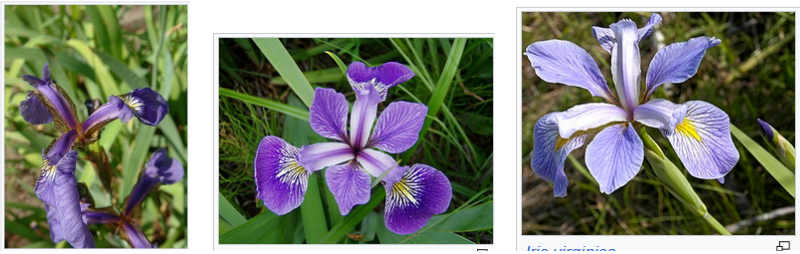
Un agent apprend si, après avoir fait des observations sur le monde, il les intègre et améliore ses performances.
Si l'agent est une machine : apprentissage artificiel.

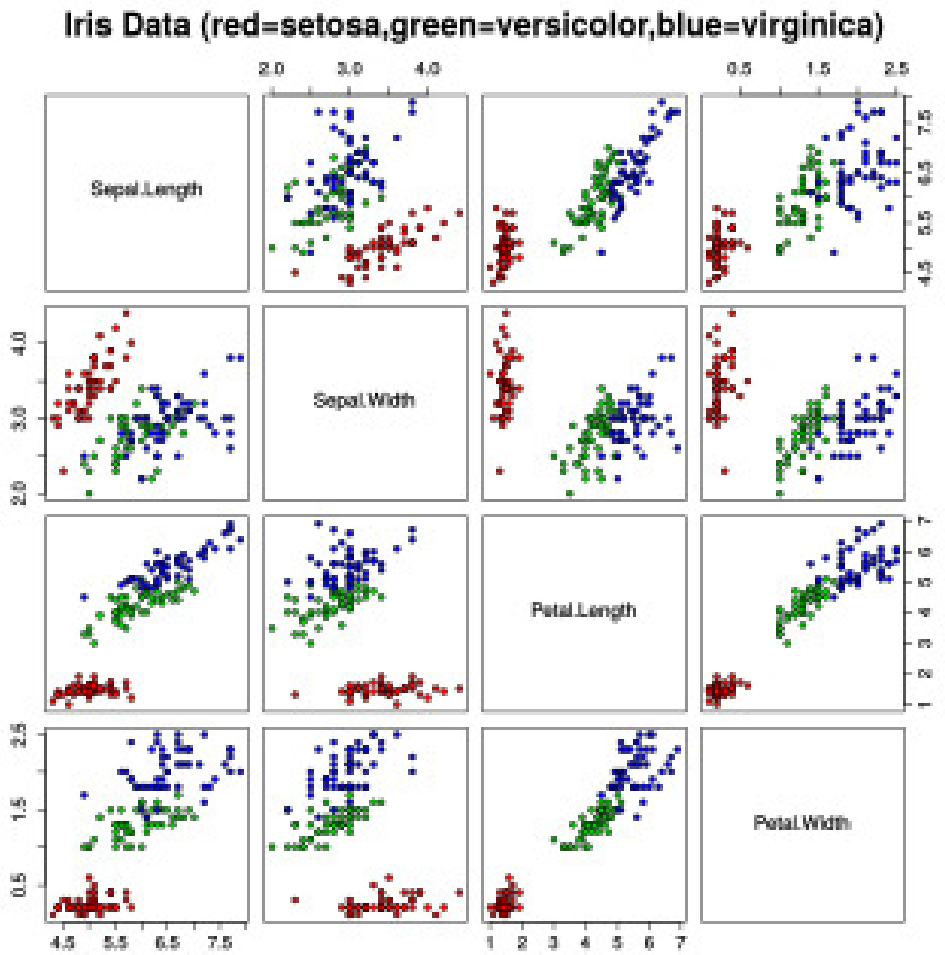
- Variations de l'environnement. Robot dans un labyrinthe.
- Ce que l'humain ne sait pas programmer. Reconnaissance de visages.

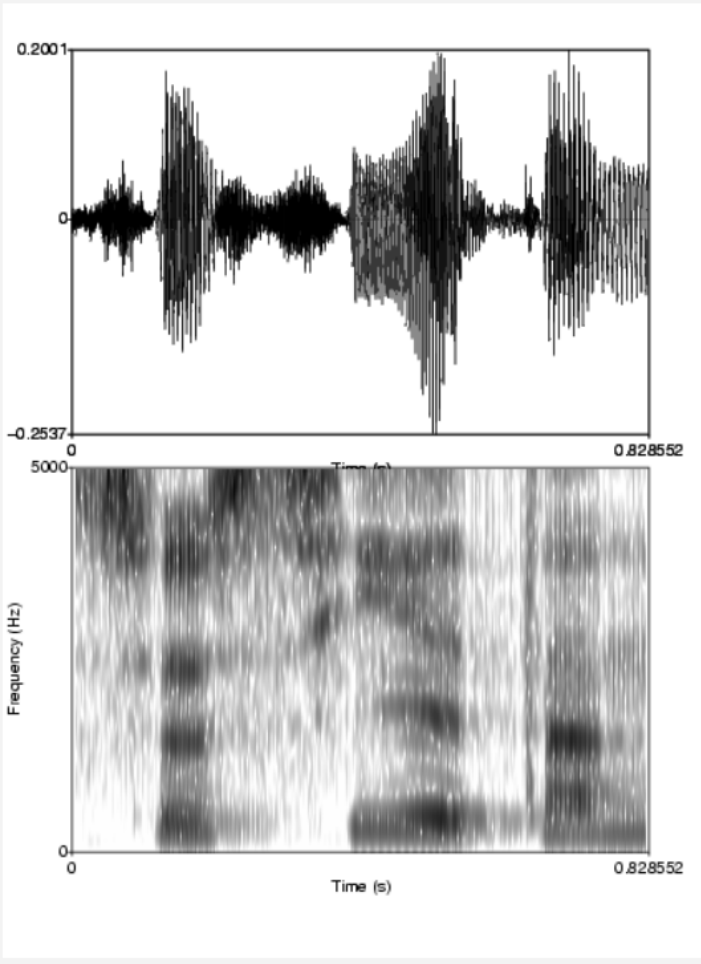
Types d'apprentissage

- Induction.
- Représentation des données. Cette semaine : Représentation factorisée.
- Classification *vs* Régression.
- Supervisé, non supervisé, renforcement (actif, en ligne).
- Peu de connaissances a priori *vs* transfert de connaissances.

Quelques types de données







Un exemple de réalisation

Reconnaissance de visages dans vos photos

Protocole suivi par Google Photos

1. La plate-forme regroupe les photos visage par visage.
2. L'utilisateur accède à l'ensemble des visages détectés.
3. Pour chaque visage, l'utilisateur peut enlever des photos et renseigner le nom de la personne.
4. Sur d'autres photos, Google Photos propose un nom et demande à l'utilisateur de confirmer ou d'infirmer.

Type d'apprentissage mis en jeu

1. Apprentissage non supervisé.
3. Etiquetage par l'expert et apprentissage hors ligne sur une base de données supervisée.
4. Apprentissage en ligne ou actif.

Plan

Les données pour décider dans quel restaurant on va

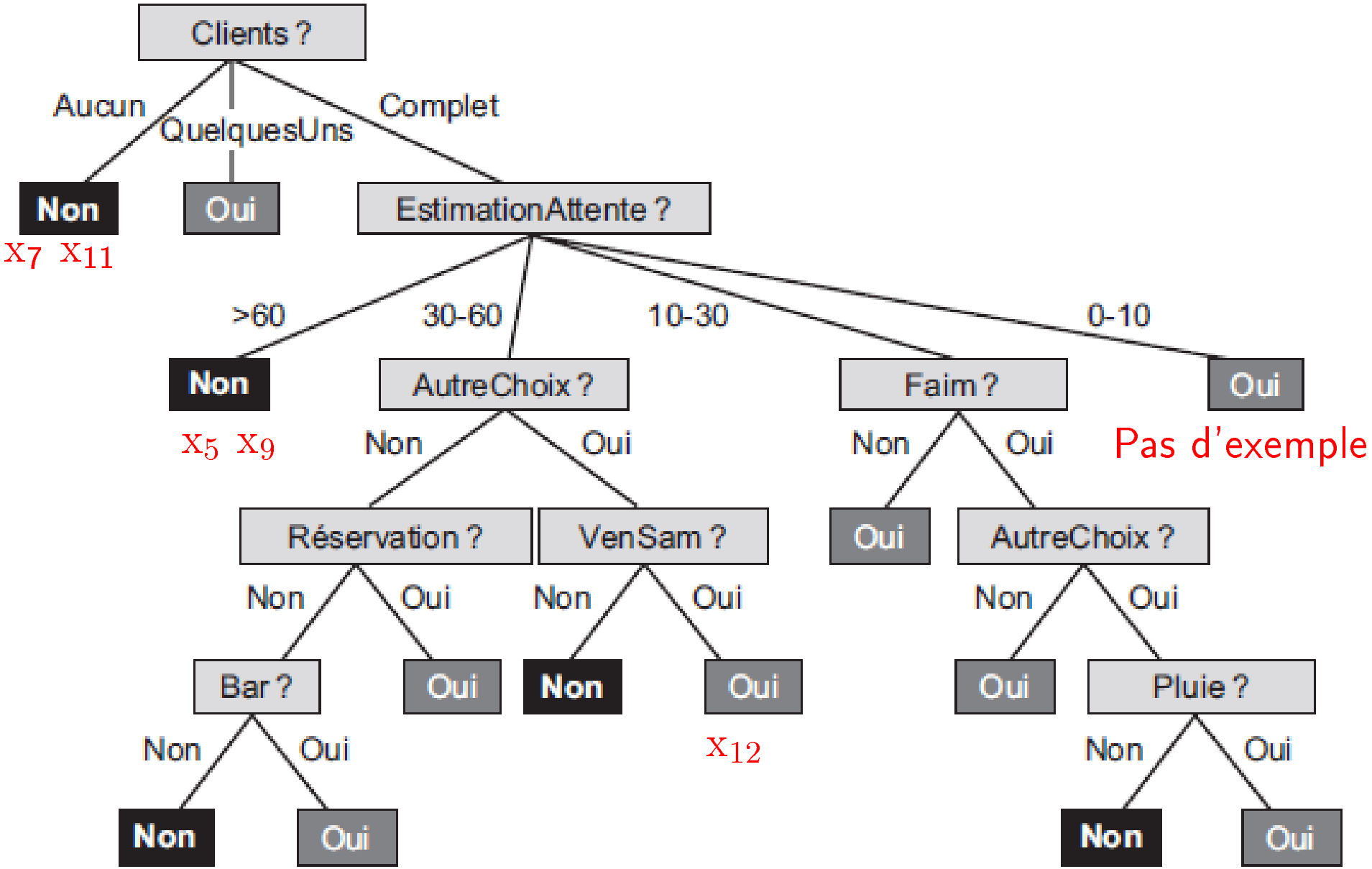
1. *Autre* : Existe-t-il dans les environs un autre restaurant qui conviendrait ?
2. *Bar* : Le restaurant dispose-t-il d'un espace bar confortable ?
3. *Ve* : Vrai : ouvert les vendredis et samedis ;
4. *Faim* : A-t-on faim ?
5. *Cli* : Combien de clients y a-t-il dans le restaurant ? Les valeurs possibles sont *Aucun*, *QuelquesUns* et *Complet*.
6. *Prix* : Quelle est la gamme de prix du restaurant : (€, €€ ou €€€) ?
7. *Pluie* : Pleut-il ?
8. *Resa* : A-t-on réservé ?
9. *Type* : Quel est le type du restaurant ? (Français, Italien, Thaïlandais ou Rapide).
10. *Est* : Quel est le temps d'attente estimé par le serveur ? (0 à 10 minutes, 10 à 30, 30 à 60 ou plus de 60.)

Données binaires, catégorielles (nominales), mais pas numériques.

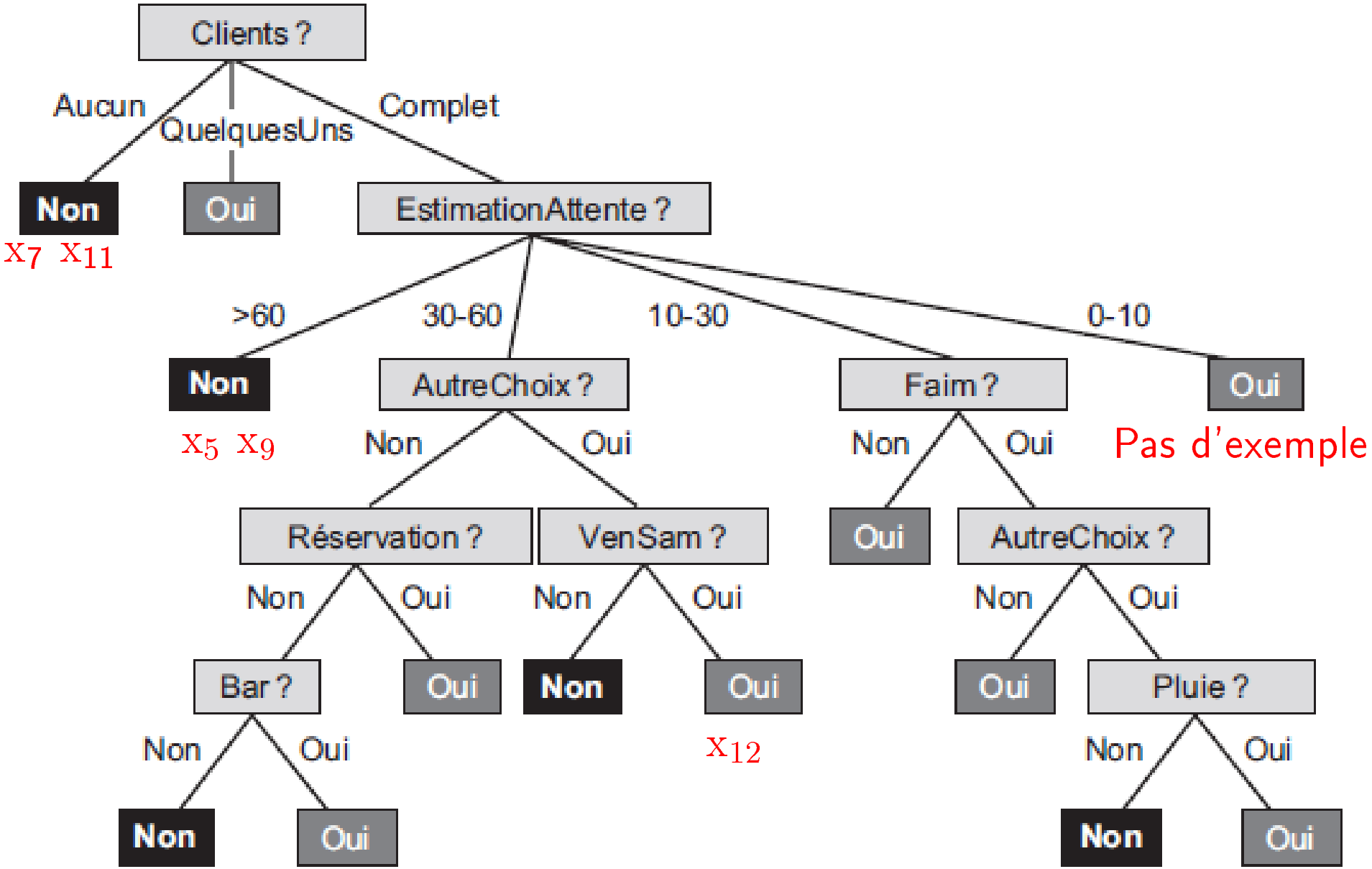
Les données d'apprentissage : 12 exemples sur 9216 possibles

Exemple	Attributs d'entrée										Sortie
	Autre	*Bar*	*Ve*	*Faim*	*Cli*	*Prix*	*Pluie*	*Résa*	*Type*	*Est*	*Attendre*
$\mathbf{x}_1$	*Oui*	*Non*	*Non*	*Oui*	*Parfois*	€€€	*Non*	*Oui*	*Français*	*0–10*	y_1 = *Oui*
$\mathbf{x}_2$	*Oui*	*Non*	*Non*	*Oui*	*Comp*	€	*Non*	*Non*	*Thai*	*30–60*	y_2 = *Non*
$\mathbf{x}_3$	*Non*	*Oui*	*Non*	*Non*	*Parfois*	€	*Non*	*Non*	*Rapide*	*0–10*	y_3 = *Oui*
$\mathbf{x}_4$	*Oui*	*Non*	*Oui*	*Oui*	*Comp*	€	*Oui*	*Non*	*Thai*	*10–30*	y_4 = *Oui*
$\mathbf{x}_5$	*Oui*	*Non*	*Oui*	*Non*	*Comp*	€€€	*Non*	*Oui*	*Français*	*>60*	y_5 = *Non*
$\mathbf{x}_6$	*Non*	*Oui*	*Non*	*Oui*	*Parfois*	€€	*Oui*	*Oui*	*Italien*	*0–10*	y_6 = *Oui*
$\mathbf{x}_7$	*Non*	*Oui*	*Non*	*Non*	*Aucun*	€	*Oui*	*Non*	*Rapide*	*0–10*	y_7 = *Non*
$\mathbf{x}_8$	*Non*	*Non*	*Non*	*Oui*	*Parfois*	€€	*Oui*	*Oui*	*Thai*	*0–10*	y_8 = *Oui*
$\mathbf{x}_9$	*Non*	*Oui*	*Oui*	*Non*	*Comp*	€	*Oui*	*Non*	*Rapide*	*>60*	y_9 = *Non*
$\mathbf{x}_{10}$	*Oui*	*Oui*	*Oui*	*Oui*	*Comp*	€€€	*Non*	*Oui*	*Italien*	*10–30*	y_{10} = *Non*
$\mathbf{x}_{11}$	*Non*	*Non*	*Non*	*Non*	*Aucun*	€	*Non*	*Non*	*Thai*	*0–10*	y_{11} = *Non*
$\mathbf{x}_{12}$	*Oui*	*Oui*	*Oui*	*Oui*	*Comp*	€	*Non*	*Non*	*Rapide*	*30–60*	y_{12} = *Oui*

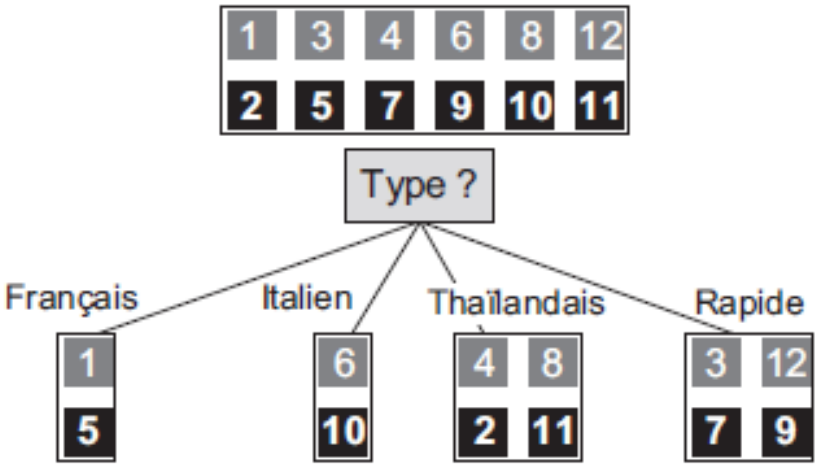
Un exemple d'arbre de décision



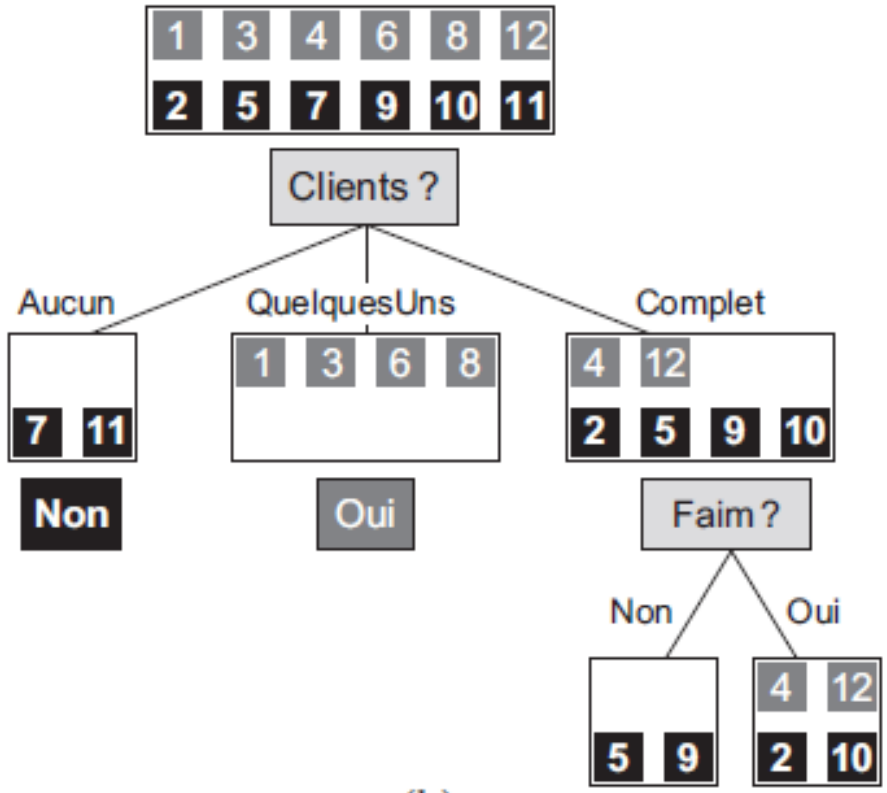
Un exemple d'arbre de décision



Qu'est-ce qu'un bon attribut à la racine de l'arbre ?



(a)



(b)

Entropie

Cas binaire

Entropie d'une v.a. booléenne qui vaut vrai avec la probabilité q :

$$\text{Entropie (en bits)} : \quad H(q) = -(q \log_2 q + (1-q) \log_2 (1-q)) .$$

Exemple : tirage d'une pièce

$$H(\text{Pièce Équilibrée}) = -(0{,}5 \log_2 0{,}5 + 0{,}5 \log_2 0{,}5) = 1 \text{ bits}$$

$$H(\text{Pièce Pipée}) = H(0{,}99) - (0{,}99 \log_2 0{,}99 + 0{,}01 \log_2 0{,}01) \approx 0{,}08 \text{ bits}$$

Cas général

Entropie d'une variable aléatoire H qui prend les valeurs H_k avec les probabilités respectives $P(v_k)$:

$$\text{Entropie} : \quad H(V) = -\sum_k P(v_k) \log_2 P(v_k) \text{ bits}$$

Entropie et construction d'un Arbre de Décision

"Clients" apporte plus d'information (moins d'entropie) que "Type" sur la division des données en "Oui" et "Non".

Quelle information reste après une division ?

Un attribut A comportant d valeurs distinctes divise l'ensemble d'apprentissage E en sous-ensembles $E_1, \ldots, E_d$.
Chaque sous-ensemble E_k contient p_k éléments positifs et n_k éléments négatifs, donc si l'on suit cette branche, on aura besoin de $H(p_k/(p_k+n_k))$ bits supplémentaires d'information pour répondre à la question.

Gain d'information apporté par une division

Un élément pris au hasard dans l'ensemble d'entraînement prendra la k-ième valeur sur l'attribut A avec la probabilité $(p_k+n_k)/(p+n)$, et l'entropie restante après le test sur cet attribut est donc :

$$Reste(A) = \sum_{k=1}^{d} \frac{p_k+k_k}{p+n} H\left(\frac{p_k}{p_k+n_k}\right) .$$

Entropie et construction d'un AD (suite)

Gain d'information apporté par un attribut

Le gain d'information apporté par le test sur l'attribut A est la réduction attendue en entropie :

$$Gain(A) = H(\frac{p}{p+n}) - Reste(A)\,.$$

Calcul sur "Clients" et "Type"

$$H(\frac{p}{p+n})$$

$$Gain(Clients) = 1 - \left[\frac{2}{12}H(\frac{0}{2}) + \frac{4}{12}H(\frac{4}{4}) + \frac{6}{12}H(\frac{2}{6})\right] \approx 0{,}541 \text{ bits},$$

$$Gain(Type) = 1 - \left[\frac{2}{12}H(\frac{1}{2}) + \frac{2}{12}H(\frac{1}{2}) + \frac{4}{12}H(\frac{2}{4}) + \frac{4}{12}H(\frac{2}{4})\right] = 0 \text{ bits},$$

Donc, *Clients* est un meilleur attribut que "Type" pour partager l'ensemble.

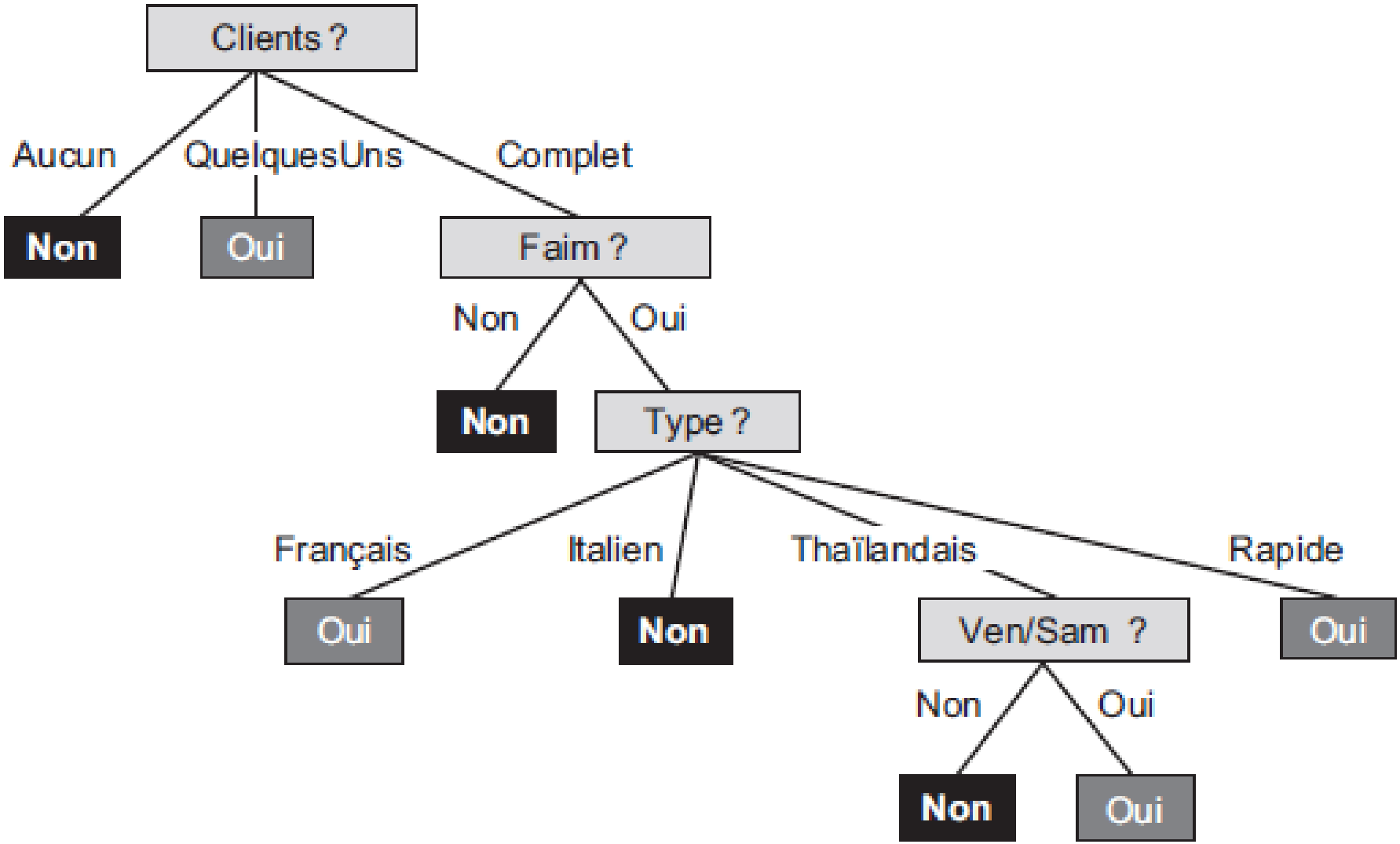
Algorithme

fonction APPRENTISSAGE-ARBRE-DÉCISION(*exemples*, *attributs*, *parent_exemples*) **retourne** un arbre

si *exemples* est vide **alors retourner** PLURALITÉ-VALEUR(*parent_exemples*)
sinon si tous les *exemples* ont la même classification **alors retourner** la classification
sinon si *attributs* est vide **alors retourner** PLURALITÉ-VALEUR(*exemples*)
sinon
  $A \leftarrow \text{argmax}_{a \in attributs}$ IMPORTANCE(a, *exemples*)
  arbre $\leftarrow$ un nouvel arbre de décision avec le test sur A à la racine
  pour chaque valeur v de A **faire**
    $exs \leftarrow \{e : e \in exemples \text{ et } e.A = v\}$
    sous_arbre $\leftarrow$ APPRENTISSAGE-ARBRE-DÉCISION(*exs*, *attributs* $- A$, *exemples*)
    ajouter une branche à *arbre* avec l'étiquette $(A = v)$ et le sous-arbre *sous_arbre*
  retourner *arbre*

IMPORTANCE : choix du meilleur attribut
PLURALITÉ-VALEUR : choix de la réponse la plus fréquente

Résultat


Clients ?
Aucun
QuelquesUns
Complet
Non
Oui
Faim ?
Non
Oui
Non
Type ?
Français
Italien
Thaïlandais
Rapide
Oui
Non
Ven/Sam ?
Oui
Non
Oui

Résultat : commentaires

- L'algorithme d'apprentissage ne teste pas *Pluie* et *Reservation*, car il peut classer tous les exemples sans ces critères.
- Il a détecté un schéma inattendu : on est plus patient en fin de semaine devant un restaurant thaïlandais.
- Il y a $2^6 \times 3^2 \times 4^2 = 9216$ possibilités et seulement 12 pour l'apprentissage. Tous pourront être classés par l'arbre construit.
- Si une feuille n'a pas d'exemple, elle est étiquetée par la majorité précédente (ou tirée au sort).
  C'est le cas de ((Client, complet), (Faim, Oui), (Type français)).
  Le nœud précédent, ((Client, complet), (Faim, Oui)) a 4 exemples, 2 "oui" et 2 "non".
- Ce n'est pas le cas ici, mais certaines feuilles peuvent être « impures ».

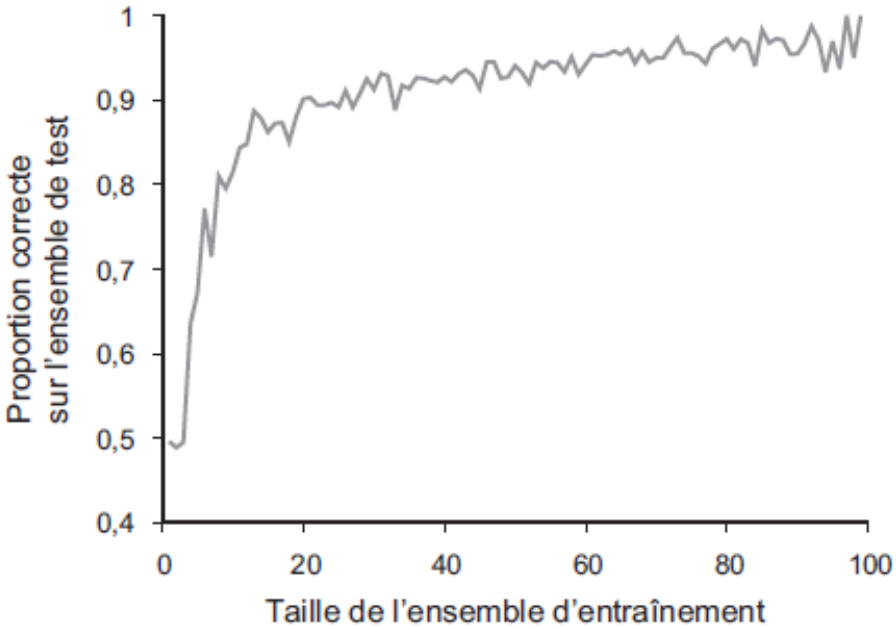
Plan

Algorithmes d'apprentissage supervisé : vocabulaire

- Espace des hypothèses $\mathcal{H}$. Tous les AD
- Hypothèse h produite par un algorithme $\mathcal{A}$. Un AD
- Généralisation (induction). Régression et Classification.
- Exemple étiqueté par un professeur (un expert, un oracle,...).
- Ensemble d'apprentissage : tous les exemples.
- Ensemble d'entraînement E ou S : données pour l'algorithme $\mathcal{A}$.
- Ensemble de validation V : réglage de l'algorithme $\mathcal{A}$.
- Ensemble de test T : estimation finale de la qualité de $\mathcal{A}$ après réglage.
- Erreur empirique (ou apparente) : mesurée sur l'ensemble d'entraînement.
- Erreur réelle : estimée sur l'ensemble de test T.

Convergence : courbe d'apprentissage d'un algorithme

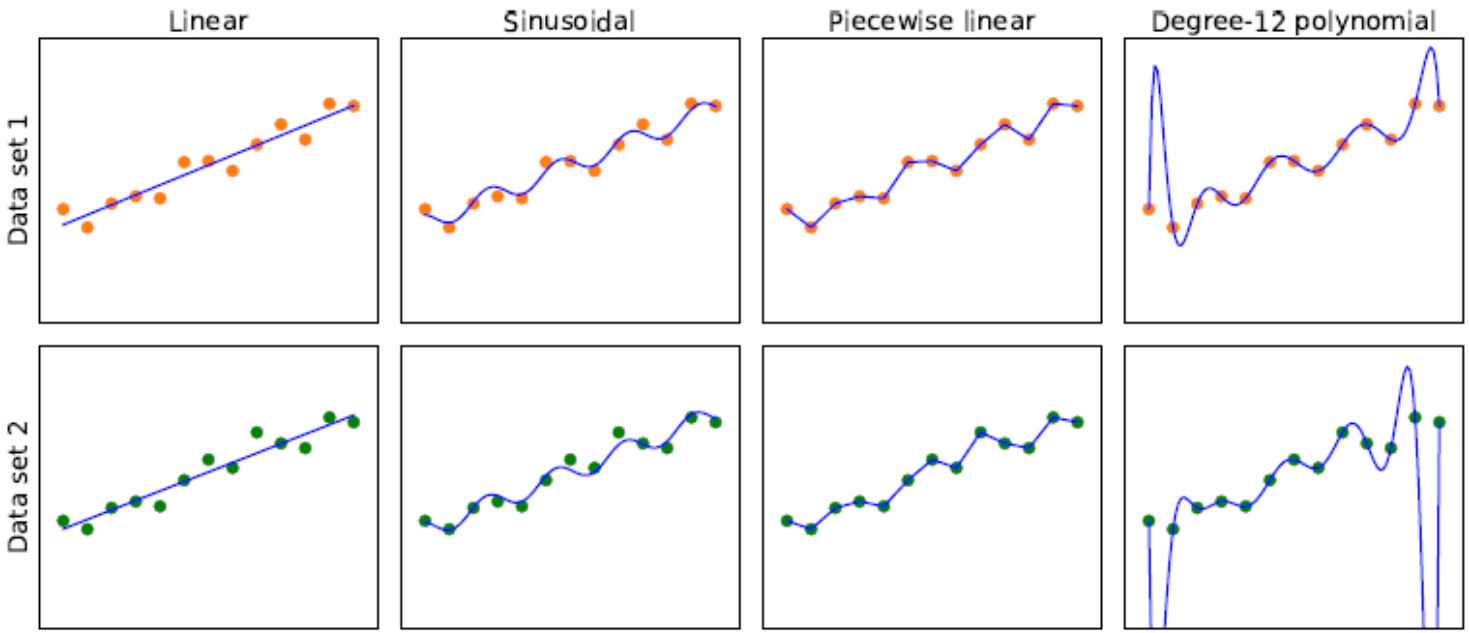
L'algorithme d'apprentissage fournit une hypothèse qu'il faut tester.

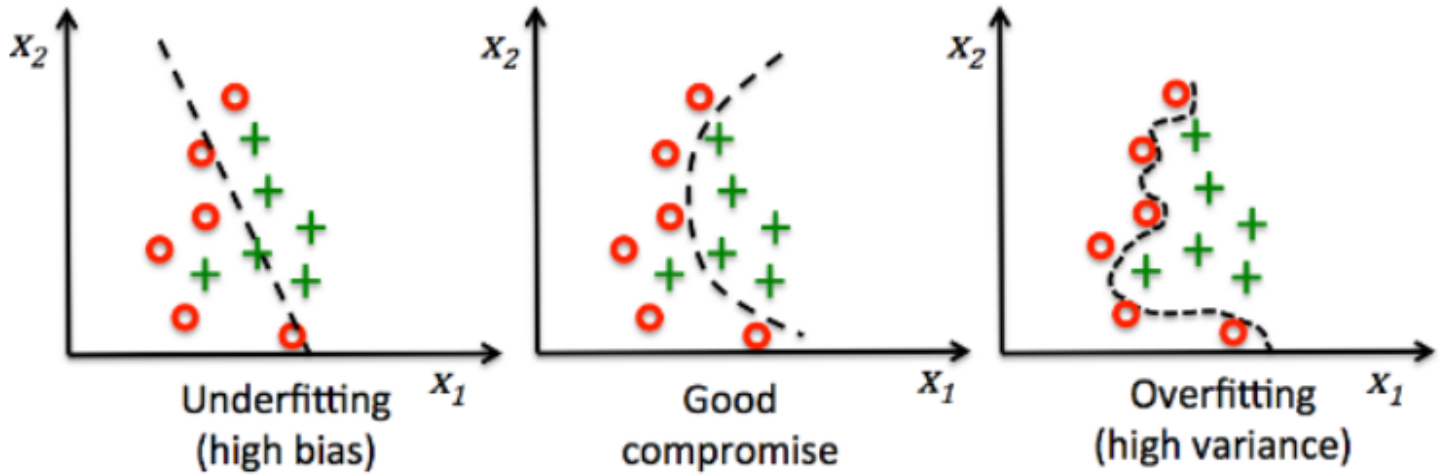


On a 100 exemples que l'on divise en ensemble d'entraînement E et ensemble de test T.

- E sert à apprendre une hypothèse.
- T mesure sa précision.
- On commence avec un E de taille 1, puis 2,. . .jusqu'à 99.
- Pour chaque taille, on répète 20 fois la division (au hasard) et on calcule la moyenne des résultats.
- La précision augmente avec la taille de T.

Biais et Variance. Régression et Classification.





Biais et Variance

Biais

La tendance d'une hypothèse de prédiction à s'éloigner de la valeur visée quand elle est moyennée sur différents ensembles d'entraînement.
Par exemple, les fonctions linéaires produisent un biais important.

Variance

Importance du changement que subit l'hypothèse quand les données d'entraînement varient.
Par exemple, les polynômes de degré 12 ont une grande variance.

Compromis Biais–Variance : trouver l'équilibre entre

Surapprentissage Des hypothèses complexes, de faible biais, qui s'adaptent trop bien aux données d'entraînement et qui généralisent mal.

Sousapprentissage Des hypothèses trop simples, de faible variance, qui s'adaptent plus ou moins bien aux données d'entraînement et qui généralisent mal.

Plan

Mesure de l'erreur

Hypothèses statistiques sur les exemples

- Les données sont stationnaires dans le temps.
- Les exemples sont statistiquement indépendants.
- Les exemples sont alors dits i.i.d. : indépendants et identiquement distribués.

Mesure de l'erreur sur un ensemble de test

Régression Moyenne des distances entre la courbe prédite et les exemples de test.

Classification : Matrice de confusion (M_{ij}) : Nombre d'exemples de test classés j de vraie classe j..
Taux d'erreur (estimation) : normaliser la somme de la diagonale
Accuracy : le contraire du taux d'erreur.

Mesure de l'erreur pour deux classes sur un ensemble de test

En ligne : la classe estimée. En colonne, la classe réelle.

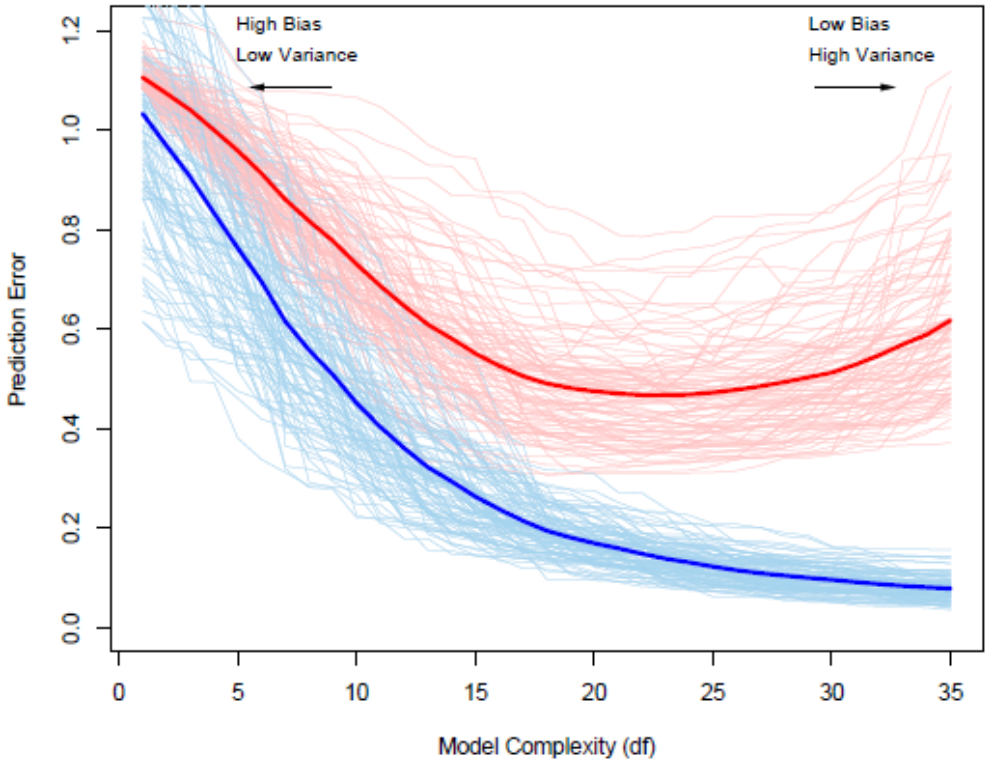
	+ (P)	− (N)
+	**Vrais positifs** (VP)	Faux positifs (FP)
−	Faux négatifs (FN)	**Vrais négatifs** (VN)

- *Taux de bonne prédiction (accuracy)* ou Précision ou (complément à 1 de l'erreur) Estimation du taux d'erreur réelle $\widehat{R}_{\text{Réel}}(h)$:

$$TEA = \frac{VP + VN}{P + N}$$

- On reverra la mesure des erreurs plus en détail.
- Ensemble d'entraînement : apparent ou empirique
- Ensemble de test : réel estimé

Réglage d'un hyper-paramètre par V



En abscisse : valeur d'un hyper-paramètre de complexité de l'algorithme

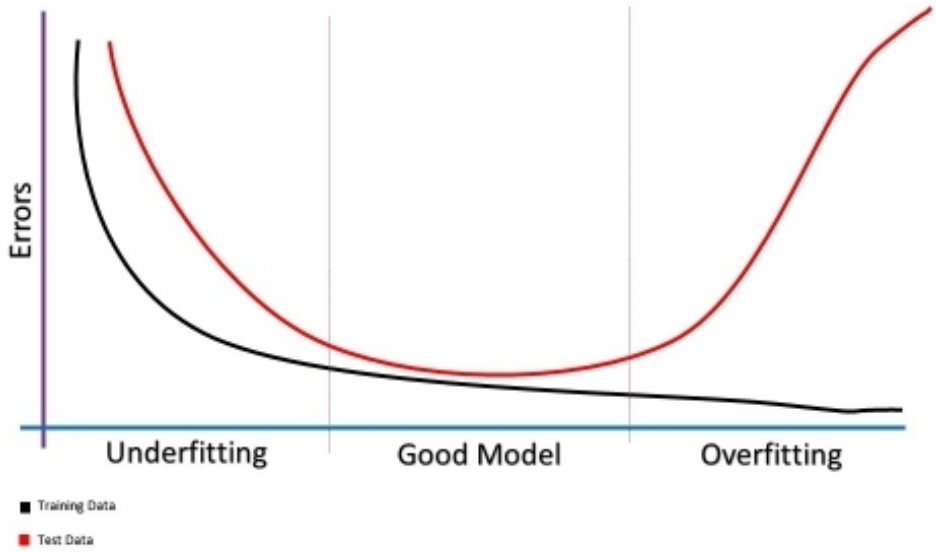
En ordonnée : Taux d'erreur

En bleu : erreur apparente sur l'ensemble d'entraînement E

En rouge : estimation de l'erreur réelle sur l'ensemble de validation V

Pour finir, utiliser une seule fois l'ensemble de test T

Réglage d'un algorithme itératif



En abscisse : nombre d'itérations
En ordonnée : estimation de l'erreur
En noir : erreur apparente sur l'ensemble d'entraînement E
En rouge : estimation de l'erreur réelle sur l'ensemble de test T
ou sur l'ensemble de validation V.

Régularisation

La technique de régularisation minimise le coût total d'une hypothèse, en prenant en compte à la fois sa perte (erreur) empirique et sa complexité

$$Cout(h) = PerteEmp(h) + \lambda\, Complexite(h)\ .$$

Par exemple, pour un Arbre de Décision, $Complexite(h)$ peut être le nombre de feuilles de l'arbre.

Dans un problème de régression polynômiale, $Complexite(h)$ est le degré du polynôme : s'il est trop élevé, il y aura du surapprentissage.

En *deep learning*, elle est un élément majeur de l'apprentissage : elle évite des minima locaux dans l'espace des paramètres (poids) du réseau.

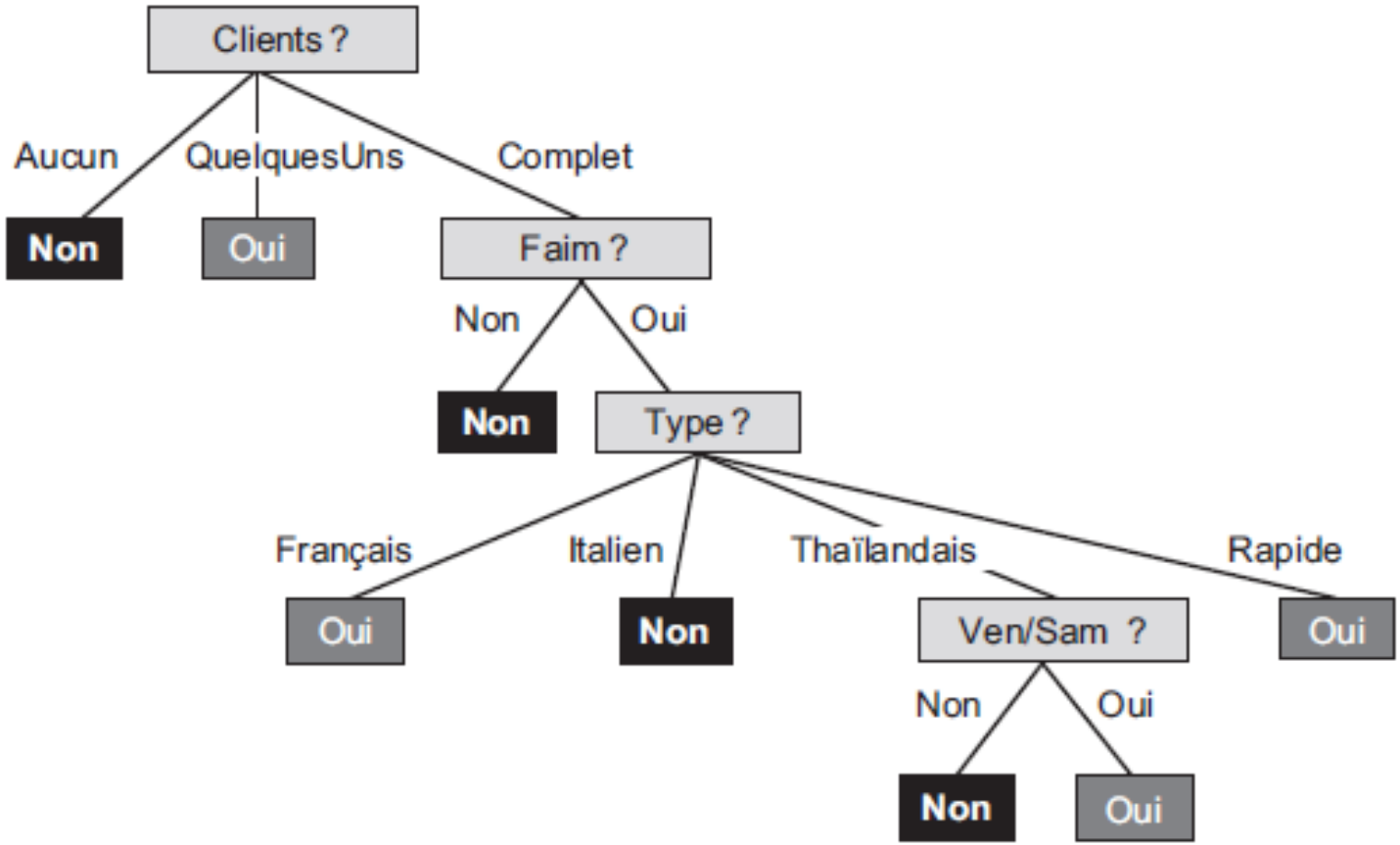
La régularisation traite le problème du compromis biais/variance.
On peut apprendre λ sur un ensemble de Validation.

Plan

Résultat pour le restaurant (rappel)

Élaguer le nœud Faim ? Il sépare les exemples 2 (Non), 4 (Oui), 5 (Non), 9 (Non), 10 (Non) et 12 (Oui).

Le remplacer par une feuille « Non ».



Élagage χ^2 contre le surapprentissage

- Éliminer ou garder un nœud construit.
- Compare son gain d'information à un gain théorique.

Supposons qu'on soit à un nœud ayant de p exemples positifs et n négatifs. On compare les valeurs p_k et n_k, d'exemples positifs et négatifs dans chacun des d sous-ensembles, avec les valeurs théoriques, $\hat{p}_k$ et $\hat{n}_k$:

$$\hat{p}_k = p \times \frac{p_k + n_k}{p + n} \qquad \hat{n}_k = n \times \frac{p_k + n_k}{p + n} \ .$$

$$\text{Puis on calcule} \qquad \Delta = \sum_{k=1}^{d} \frac{(p_k - \hat{p}_k)^2}{\hat{p}_k} + \frac{(n_k - \hat{n}_k)^2}{\hat{n}_k} \ .$$

Δ suit la distribution χ^2 (chi-2) avec $v - 1$ degrés de liberté. On peut donc utiliser une table de valeurs de χ^2 pour valider ou éliminer le nœud.

Pour $v = 4$, $\Delta = 7.82$ pour une confiance de 95% et
$\Delta = 11.35$ pour une confiance de 99%

Élagage par Régularisation ou *cost complexity pruning*

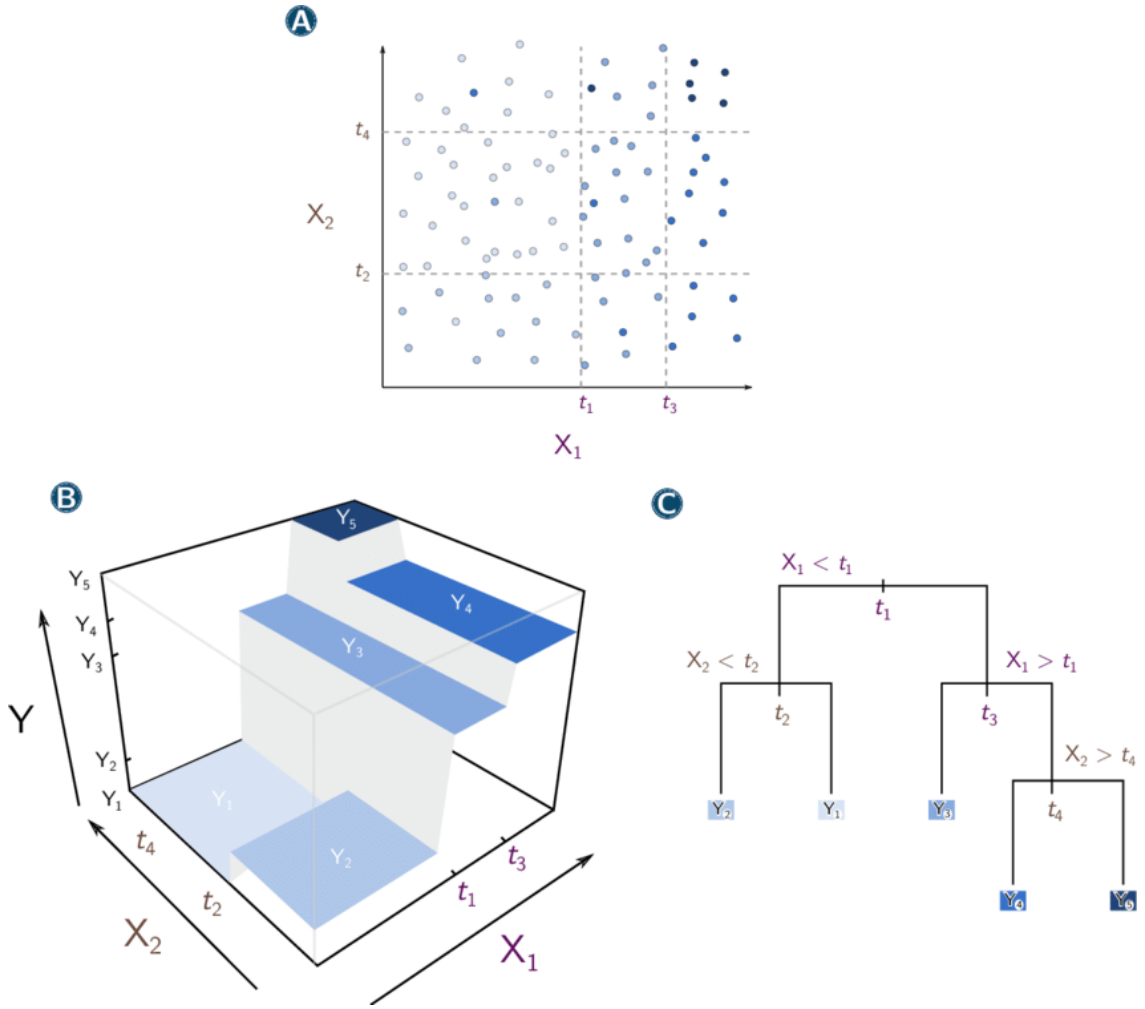
Pour un arbre T, on ajoute un terme de complexité à l'erreur apparente en choisissant une valeur pour α et en posant

$$R_\alpha(T) = R(T) + \alpha|\widetilde{T}|$$

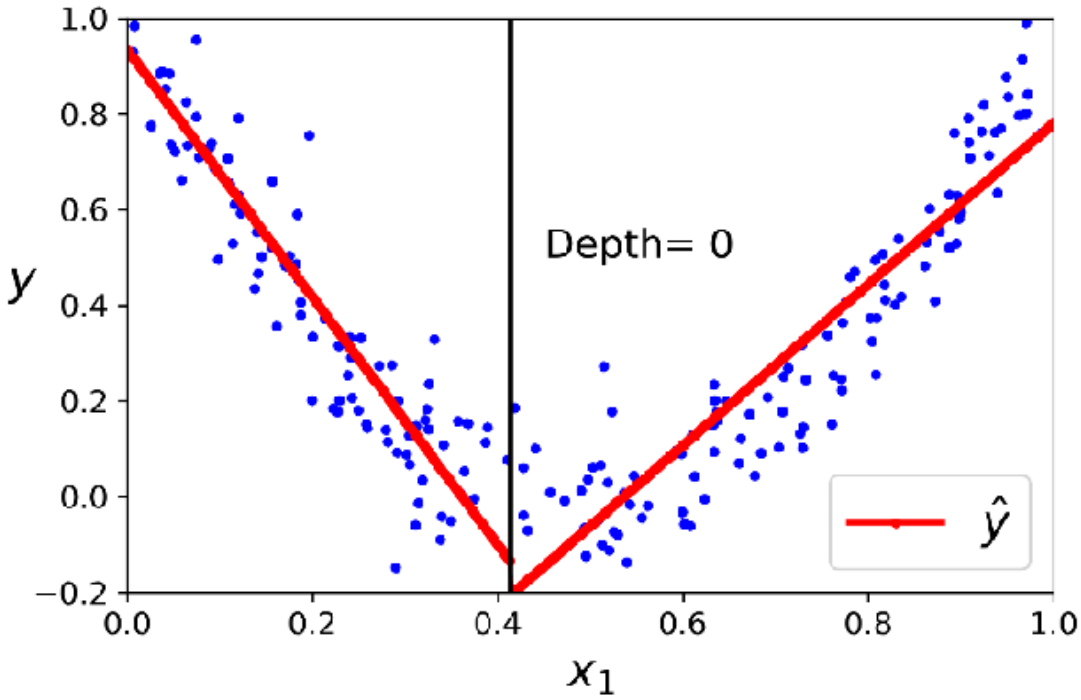
Avec $|\widetilde{T}|$ le nombre de feuilles de T et avec $R(T)$ l'erreur apparente.
L'élagage par régularisation (coût minimal) cherche le sous-arbre de T qui minimise $R_\alpha(T)$.

La complexité d'un nœud t pris isolément est $R_\alpha(t) = R(t) + \alpha$. Le sous-arbre de racine t se note T_t. On choisit la valeur efficace de α en imposant $R_\alpha(T_t) = R_\alpha(t)$, donc : $\alpha_{eff}(t) = \frac{R(t)-R(T_t)}{|T|-1}$.
Le nœud (non feuille) $t^\star$ qui a la plus petite valeur de $\alpha_{eff}(t)$ sera élagué : on détruit le sous-arbre $T_{t^\star}$.
On recommence sur l'arbre élagué jusqu'à ce que α_{eff} dépasse α.

Extension : attributs continus



Extension : arbre de régression



Plan

Construction d'un AD Attributs binaires

DF : mes Devoirs sont-ils Finis ?

MBH : Maman est-elle de Bonne Humeur ?

GP : Mon Goûter est-il Pris ?

FB : Est-ce qu'il Fait Beau ?

Décision : Aller jouer dehors

	DF	MBH	FB	GP	**DECISION**
1	VRAI	FAUX	VRAI	FAUX	**OUI**
2	FAUX	VRAI	FAUX	VRAI	**OUI**
3	VRAI	VRAI	VRAI	FAUX	**OUI**
4	VRAI	FAUX	VRAI	VRAI	**OUI**
5	FAUX	VRAI	VRAI	VRAI	**NON**
6	FAUX	VRAI	FAUX	FAUX	**NON**
7	VRAI	FAUX	FAUX	VRAI	**NON**
8	VRAI	VRAI	FAUX	FAUX	**NON**

DF	*VRAI*	*FAUX*
OUI	3	1
NON	2	2
8	5	3

GP	*VRAI*	*FAUX*
OUI	2	2
NON	2	2
8	4	4

MBH	*VRAI*	*FAUX*
OUI	2	2
NON	3	1
8	3	5

FB	*VRAI*	*FAUX*
OUI	3	1
NON	1	3
8	4	4

$$H(M|DF) = \frac{5}{8}J(DF = VRAI) + \frac{3}{8}J(DF = FAUX)$$

avec :

$$J(DF = VRAI) = -\frac{3}{5}log(\frac{3}{5}) - \frac{2}{5}log(\frac{2}{5})$$

$$J(DF = FAUX) = -\frac{1}{3}log(\frac{1}{3}) - \frac{2}{3}log(\frac{2}{3})$$

Soit :

$$H(M|DF) \approx 0.93$$

On a aussi : $H(M|FB) \approx 0.80$, $H(M|BH) \approx 0.93$, $H(M|GP) = 1$.
On choisit donc pour racine de l'arbre le test *Fait-il Beau ?* Au bout de la branche Gauche portant la valeur *VRAI* se trouve le tableau suivant des exemples corrects pour ce test :

	mes Devoirs sont-ils Finis ?	Maman est-elle de Bonne Humeur	Mon Goûter est-il Pris ?	**DECISION**
1	VRAI	FAUX	FAUX	**VRAI**
3	VRAI	VRAI	FAUX	**VRAI**
4	VRAI	FAUX	VRAI	**VRAI**
5	FAUX	VRAI	VRAI	**FAUX**

Sous la branche Droite, portant la valeur *FAUX* se trouve le tableau :

	mes Devoirs sont-ils Finis ?	Maman est-elle de Bonne Humeur	Mon Goûter est-il Pris ?	**DECISION**
2	FAUX	VRAI	VRAI	**VRAI**
6	FAUX	VRAI	FAUX	**FAUX**
7	VRAI	FAUX	VRAI	**FAUX**
8	VRAI	FAUX	FAUX	**FAUX**

Jouer au tennis

Aller jouer au tennis ?

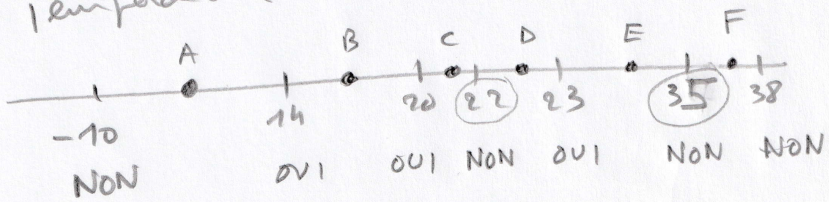
Un attribut numérique : la température T

Un attribut binaire *Libre* : le court est-il libre ?

	T	*Libre*	Décision
1	38	Oui	NON
2	24	Oui	OUI
3	−10	Oui	NON
4	14	Oui	OUI
5	20	Oui	OUI
6	35	Non	NON
7	22	Non	NON

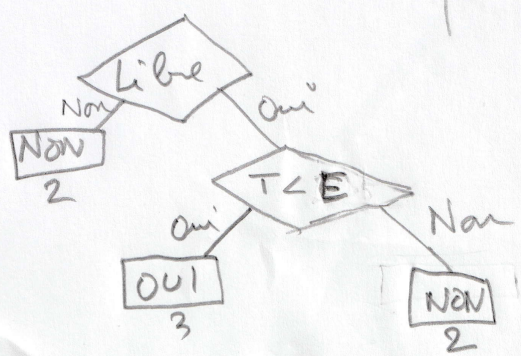
Libre	Décision OUI	NON
Oui	3	2
Non	0	2

Température



T	Décision OUI	NON
<A	0	1
>A	3	3

T	Décision OUI	NON
<E	3	2
>E	0	2



Russell-18.5 Énoncé

Suppose we generate a training set from a decision tree and then apply decision-tree learning to that training set. Is it the case that the learning algorithm will eventually return the correct tree as the training-set size goes to infinity ? Why or why not ?

Solution

The algorithm may not return the "correct" tree, but it will return a tree that is logically equivalent, assuming that the method for generating examples eventually generates all possible combinations of input attributes. This is true because any two decision tree defined on the same set of attributes that agree on all possible examples are, by definition, logically equivalent. The actually form of the tree may differ because there are many different ways to represent the same function.

Russell-18.3 Énoncé

Draw a decision tree for the problem of deciding whether to move forward at a road intersection, given that the light has just turned green.

Solution Montre la quantité d'exceptions a prévoir en modélisation par système expert... donc l'avantage de l'AA

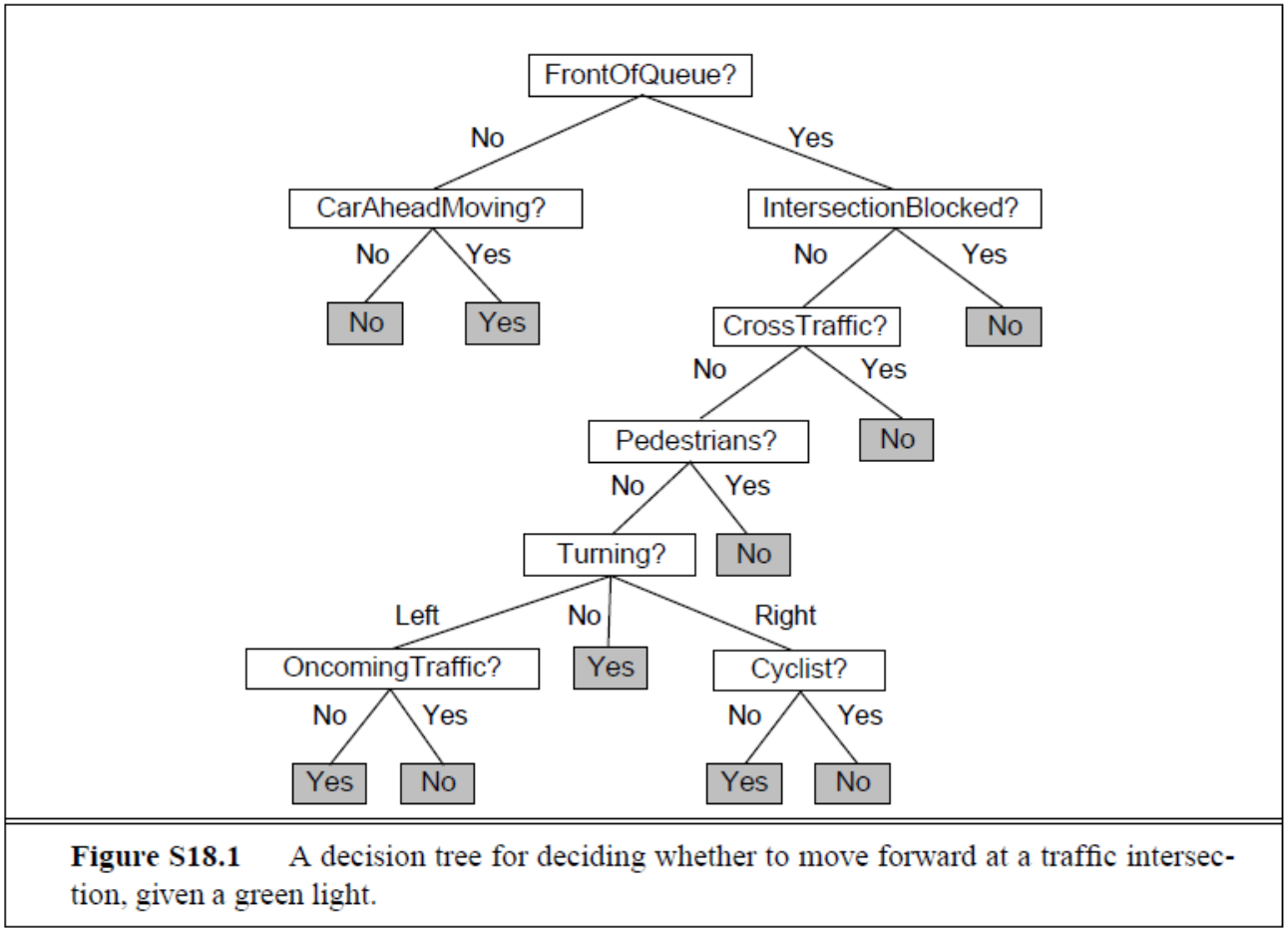


Figure S18.1 A decision tree for deciding whether to move forward at a traffic intersection, given a green light.

Plan

Prise en main de Python avec `scikit-learn`.

Essai d'une application avec les arbres de décision.

Utiliser les remarques `1.10.5. Tips on practical use`

`1.10.9. Minimal Cost-Complexity Pruning`

`https://scikit-learn.org/stable/modules/tree.html#classificatio`