

CLASSIFICATION ET ARBRES DE DÉCISIONS

Programmation pour l'IA

Florent Capelli

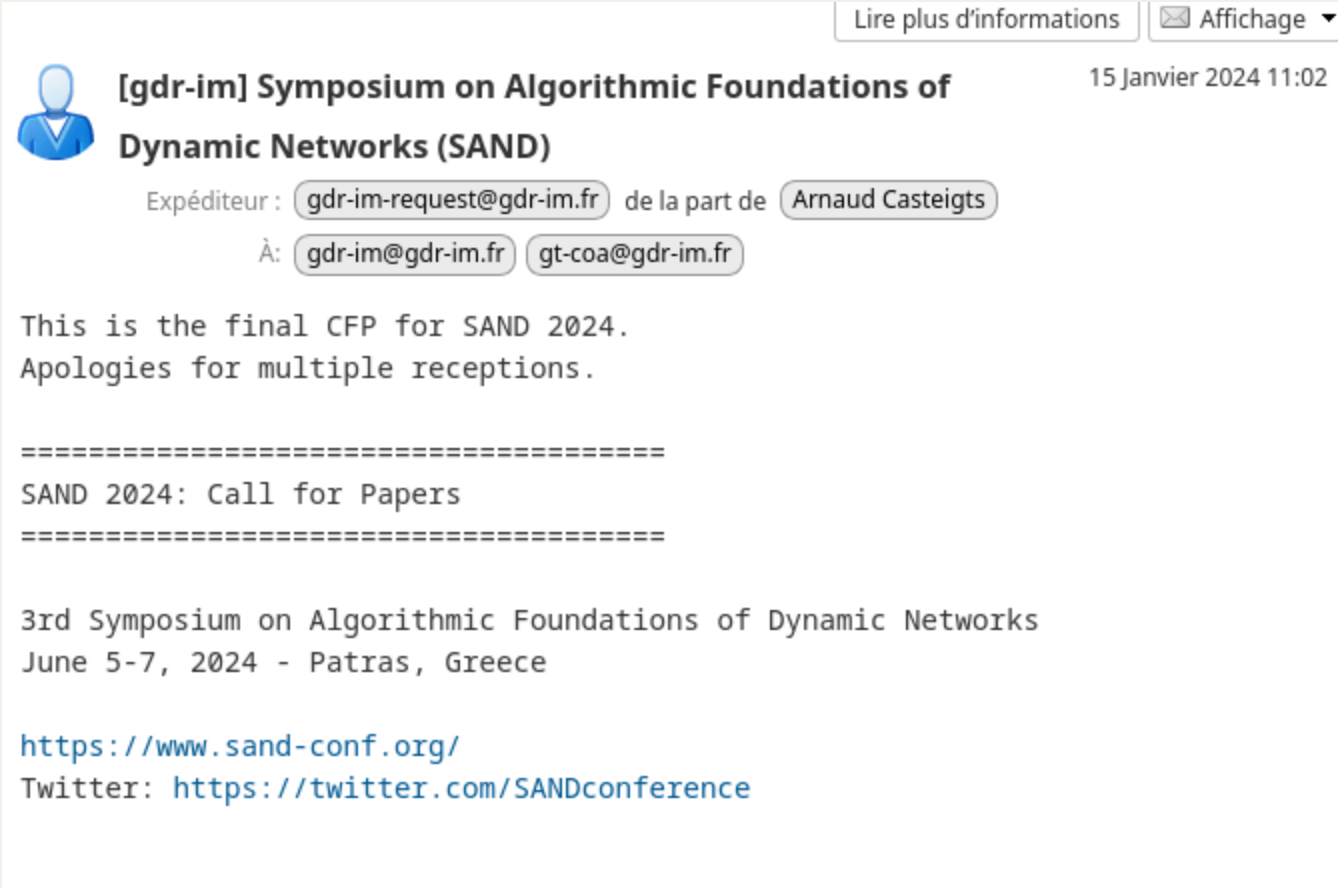
Université d'Artois, CRIL

PROBLÈME DE CLASSIFICATION

Exemples



spam



pas spam

Données : texte.

Exemples



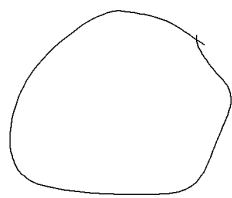
Chat

Données : images.

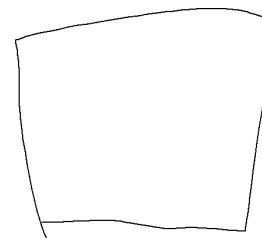


Tigre

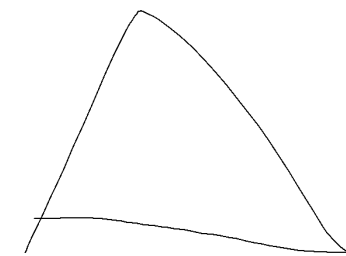
Exemples



Rond



Carré



Triangle

Données : liste de points.

Exemples

Âge	Métier	Revenus	Crédit initial	Taux	Durée	Accorder un crédit ?
32	Agent immobilier	45k€	100k€	2.5%	15	non
45	Enseignant	30k€	150k€	1.5%	15	oui
52	Électricien	40k€	100k€	2.5%	25	oui

Données : table.

Classification

On veut classer des objets $o \in \mathcal{O}$ dans des classes $\mathcal{C} = \{c_1, \dots, c_k\}$.

Par défaut : on cherche à associer chaque objet o à une unique classe $f(o) = c_i$.

- Comment définir/implémenter $f(o)$? *Implémenter f qui décide si une image contient un chat ou non ?*
- Est-ce que la valeur $f(o)$ est vraiment bien définie ? *Parfois (espèce d'une plante), parfois non (genre d'un morceau de musique).*

Modélisation

On suppose donc l'existence (théorique) des distributions de probabilité suivantes :

- Une distribution sur $\mathcal{O}$: $P(o)$ est la probabilité de rencontrer l'objet o .
 - Distribution théorique, inestimable mais nécessaire théoriquement
 - Ordre de grandeurs importants : si 98% de la population n'a pas la maladie A, un test pour la maladie A doit être meilleur que le test qui dit "non".
- Une distribution sur $\mathcal{O} \times \mathcal{C}$: $P(o, c)$ est la probabilité que o soit classé comme c .
 - Probabilité théorique
 - Intuition: si on classe o dans c avec probabilité $P(o, c)$, l'utilisateur sera "content".

Classification idéale ?

On note $P(c|o) = P(o, c)/P(o)$ la probabilité de classer l'objet o dans la classe c . Comment définir $f(o)$ qui classe "idéalement" ?

- f_{maj} (règle majoritaire): on retourne toujours la même classe c qui maximise $P(c)$, indépendamment de o .
 - **C'est stupide !** Oui et non, il faut que votre classifieur puisse battre cela !
- f_{vrais} (maximum de vraisemblance): on retourne c qui maximise $P(o|c) = P(o, c)/P(c)$.
- f_{Bayes} (règle de Bayes): on retourne c qui maximise $P(c|o) = P(o, c)/P(o)$.

La règle de Bayes semble être la plus raisonnable (on verra que c'est le cas).

Représentation

Nous n'accédons qu'à une **représentation** $r(o)$ des objets $o \in \mathcal{O}$ à classer.

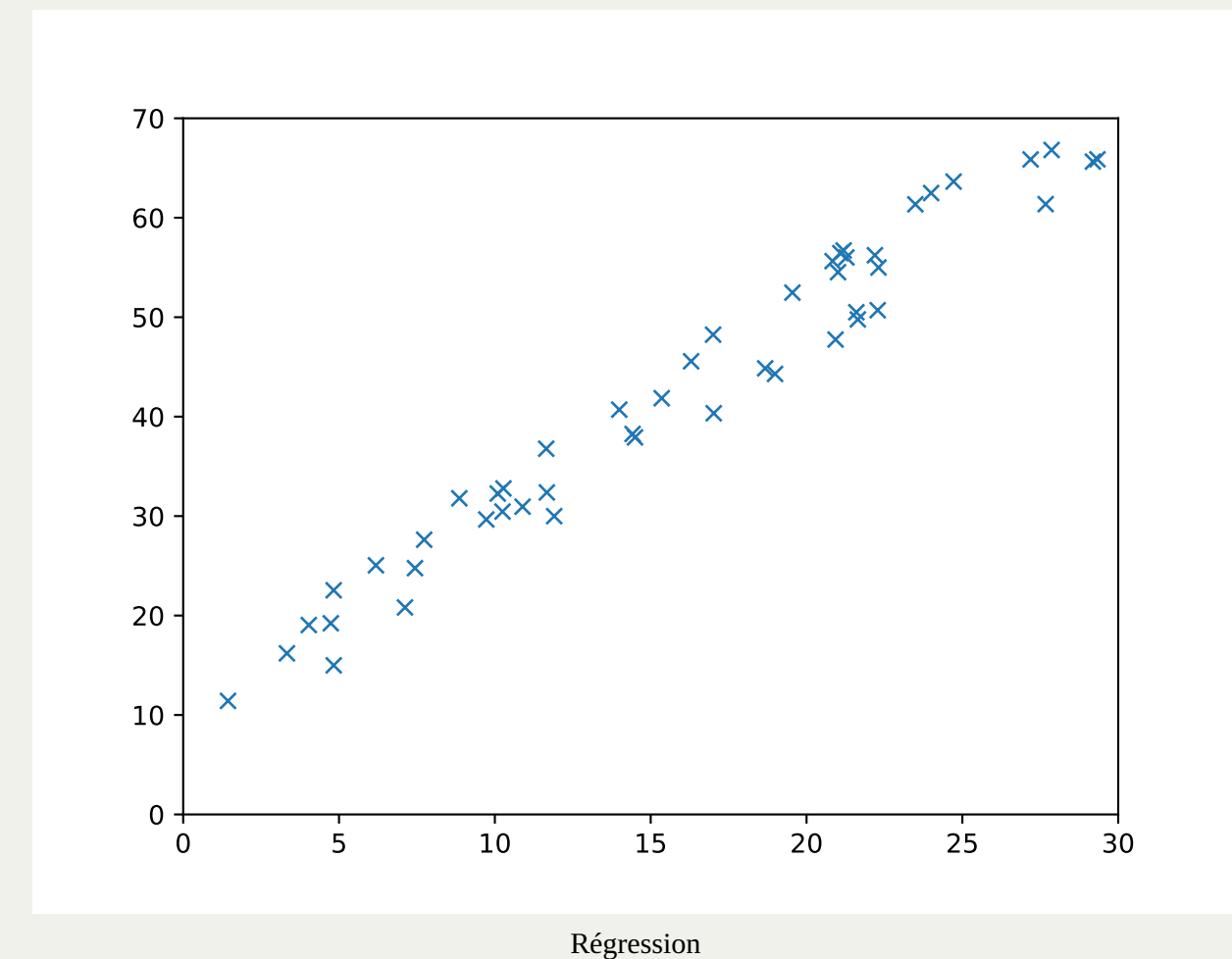
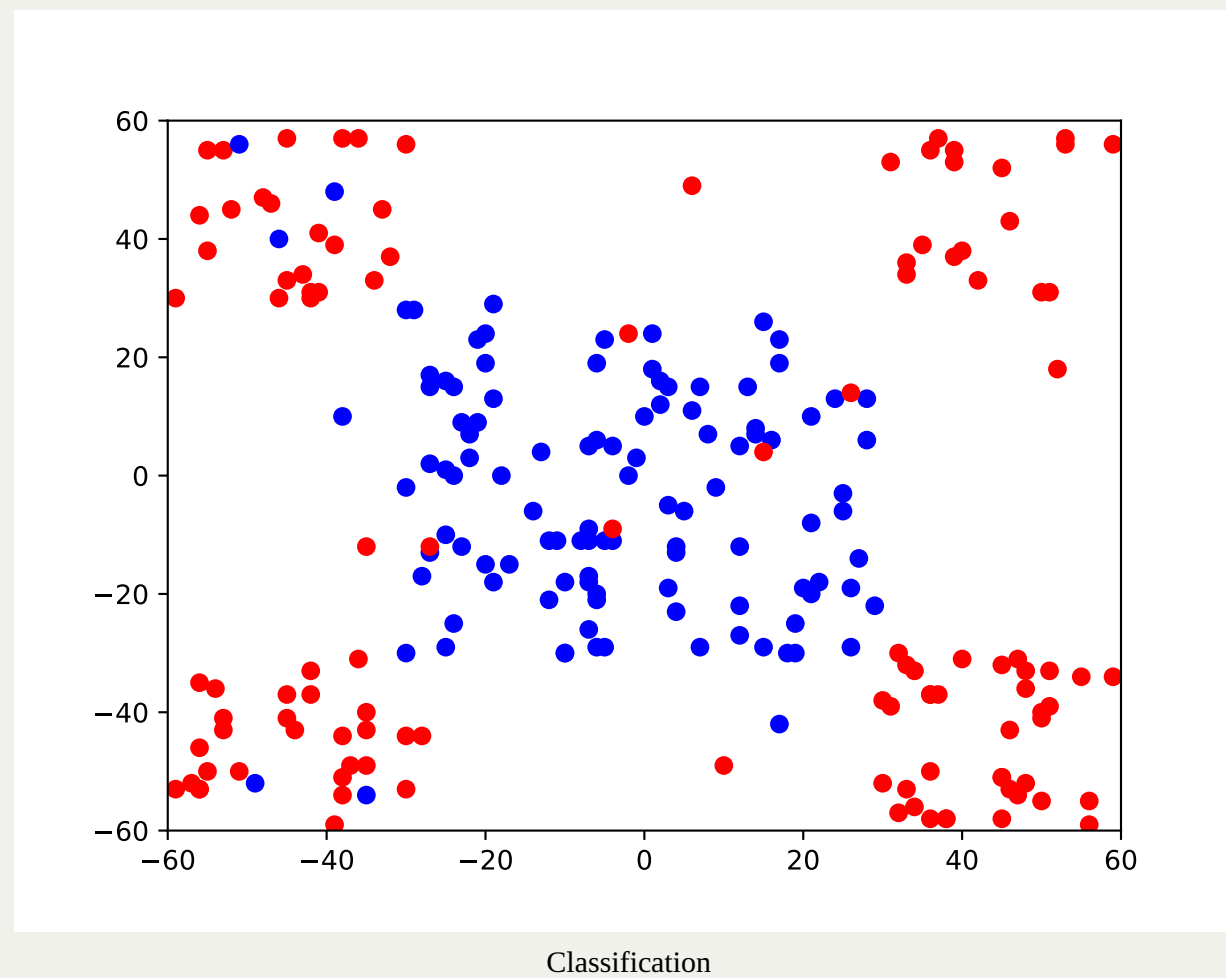
- Une photo d'un animal n'est pas l'animal lui-même.
- Doit-on ajouter l'équipe de foot supportée pour décider si on accorde un crédit ?

On a une **erreur incompressible** due au choix de la représentation. Deux objets peuvent avoir la même représentation mais pas la même classe :

Âge	Métier	Revenus	Crédit initial	Taux	Durée	A remboursé son crédit ?
32	Agent immobilier	45k€	100k€	2.5%	15	oui
32	Agent immobilier	45k€	100k€	2.5%	15	non

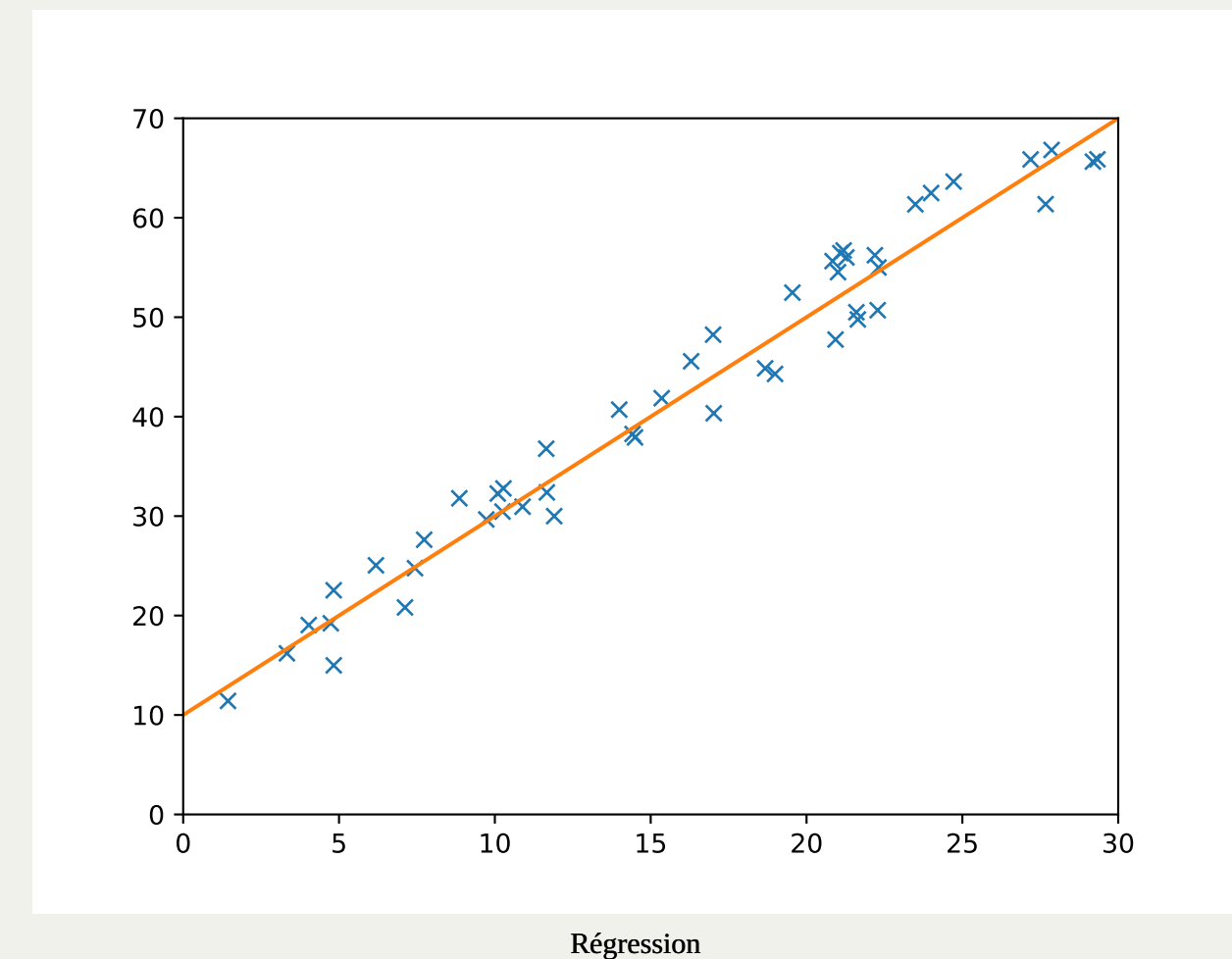
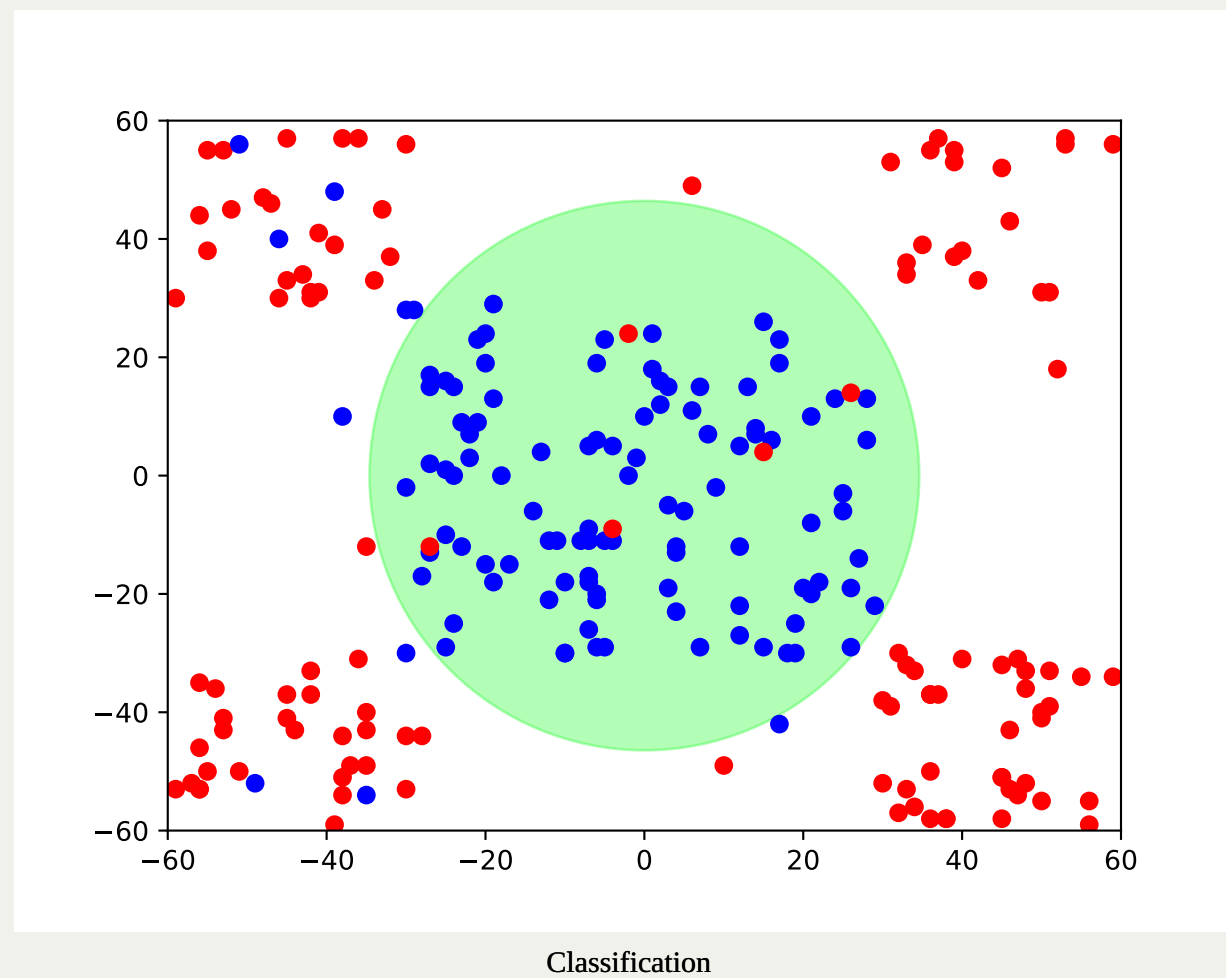
Classification $\neq$ Régression

- **Régression** : on veut estimer une quantité (ou plusieurs) $f(o) \in \mathbb{R}$ pour $o \in \mathcal{O}$.
 - Pourcentage du crédit remboursé.
 - Note donnée à un film par l'utilisateur.
- Classification avec classes $c_1, \dots, c_k$ peut être vu comme une tâche de regression:
 - $f(o) \in [k]$.
 - ou bien $f(o) \in [0, 1]^k$ avec $f(o)$ une probabilité que o soit classé comme c_k .



Classification $\neq$ Régression

- **Régression** : on veut estimer une quantité (ou plusieurs) $f(o) \in \mathbb{R}$ pour $o \in \mathcal{O}$.
 - Pourcentage du crédit remboursé.
 - Note donnée à un film par l'utilisateur.
- Classification avec classes $c_1, \dots, c_k$ peut être vu comme une tâche de regression:
 - $f(o) \in [k]$.
 - ou bien $f(o) \in [0, 1]^k$ avec $f(o)$ une probabilité que o soit classé comme c_k .



Comment classifier ?

On a une représentation $x = r(o)$ de nos objects et on cherche une fonction f telle que $f(x)$ est la classe la plus probable de o . Comment implémenter f ?

Regarder les feuilles



page 37

Sureau noir



page 26

Frêne élevé

6 B Feuille entière → feuille glabre → avec des @lobes, arrondis ou pointus 5

Sur le bord des feuilles, on voit...

des dents arrondies
marge crénelée



page 32

Peuplier tremble

des dents pointues
marge dentée



✗

En regardant avec une loupe, on voit sur les feuilles ...

aucun poil
feuille glabre

page 18

Charme commun

des poils minuscules
feuille poilue

page 38

Tilleul à petites feuilles

Extrait d'un livret du projet [Botascopia](#)

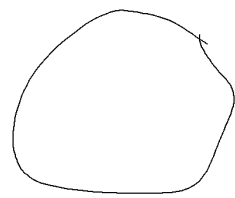
Classification des plantes :

- clé d'identification
- établie par des botanistes.

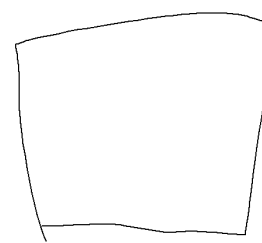
On utilise une connaissance experte.

Comment classifier ?

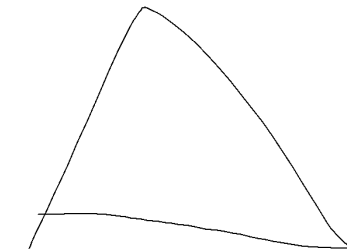
On a une représentation $x = r(o)$ de nos objects et on cherche une fonction f telle que $f(x)$ est la classe la plus probable de o . Comment implémenter f ?



Rond



Carré

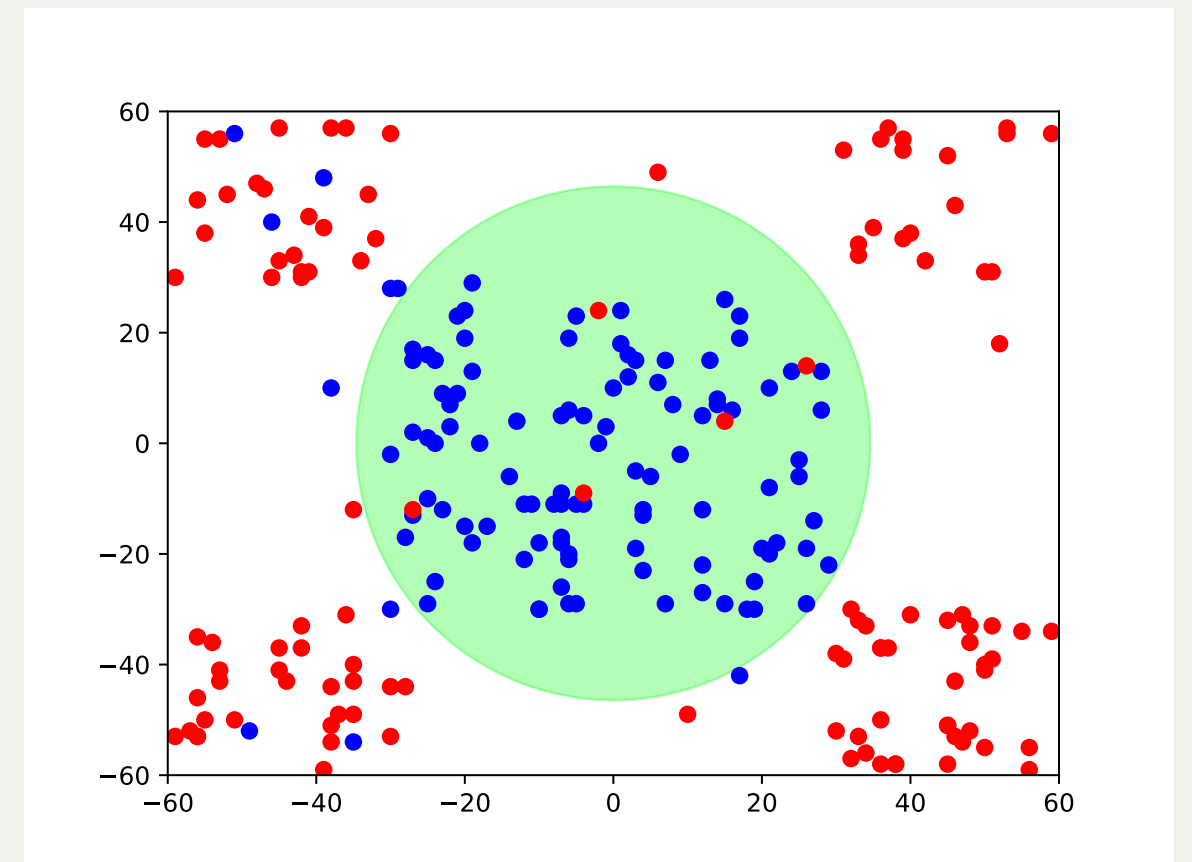
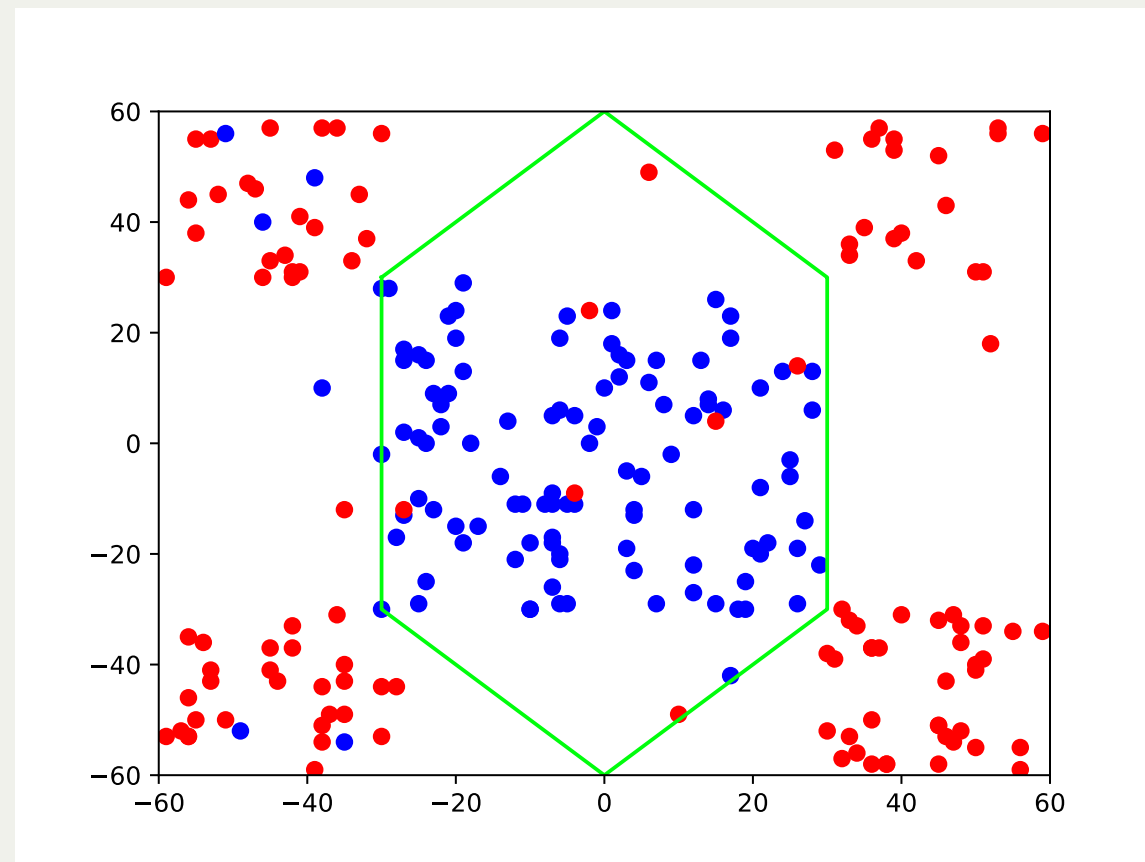
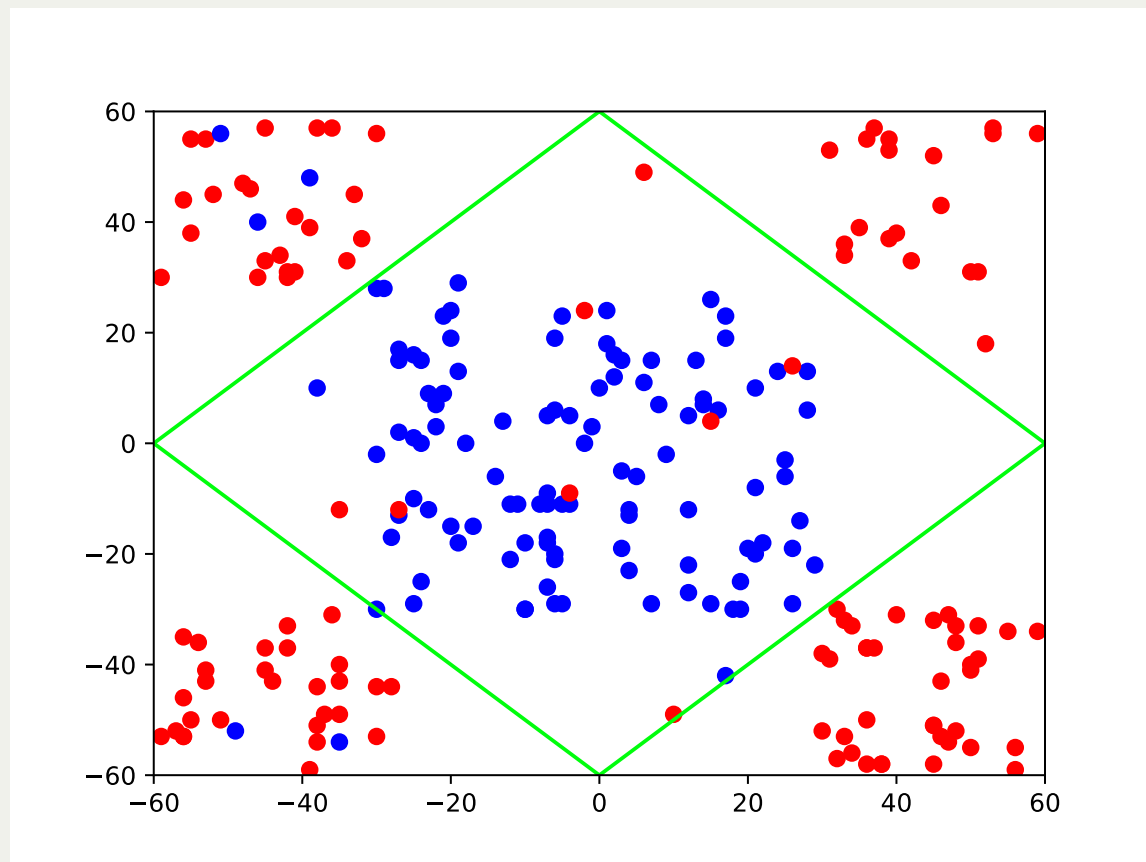


Triangle

\$1-classifier (voir TP1): **hard-code une heuristique suffisamment performante.**

Comment classifier ?

On a une représentation $x = r(o)$ de nos objects et on cherche une fonction f telle que $f(x)$ est la classe la plus probable de o . Comment implémenter f ?

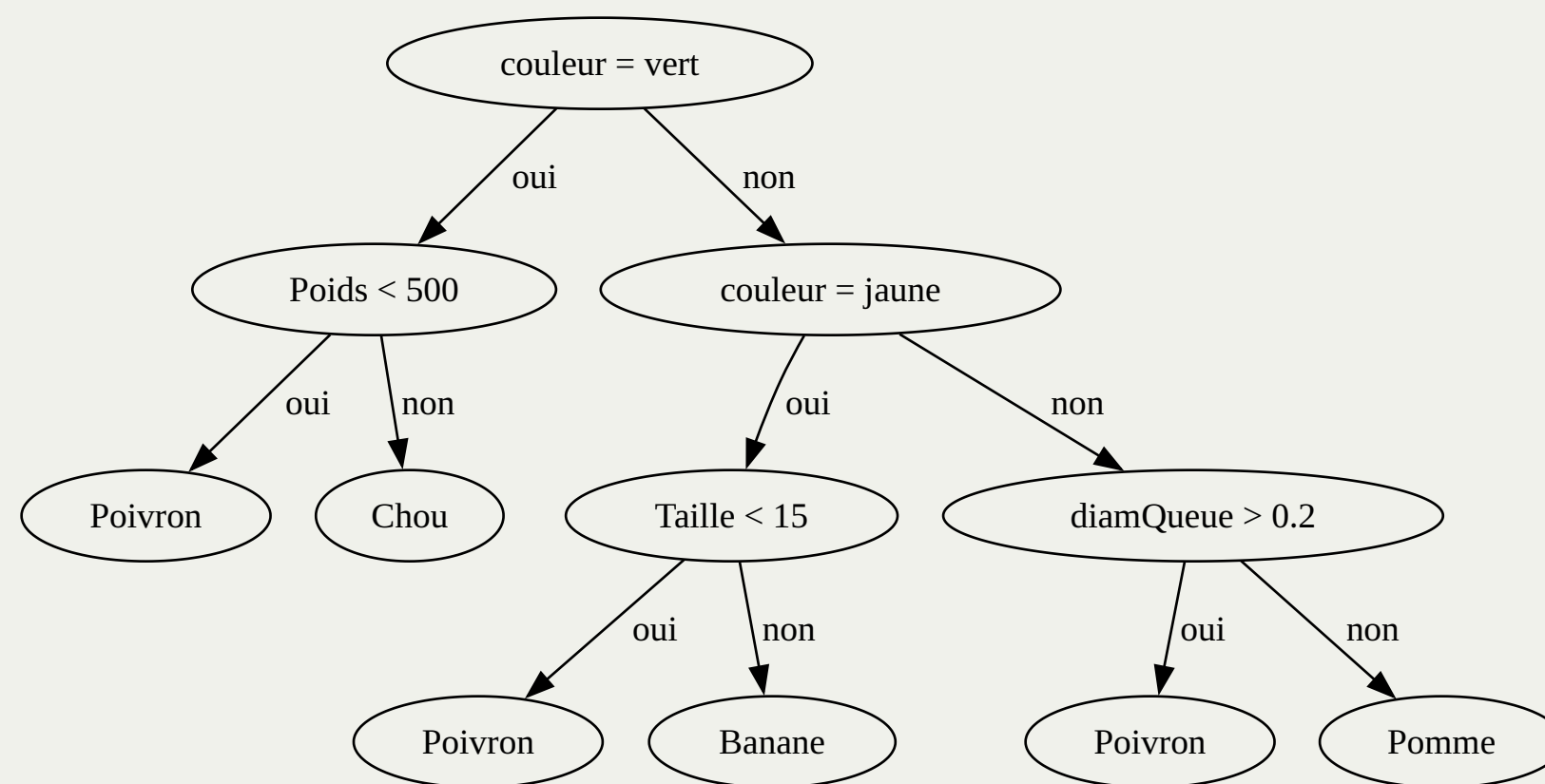


On apprend un modèle permettant à partir d'exemples :

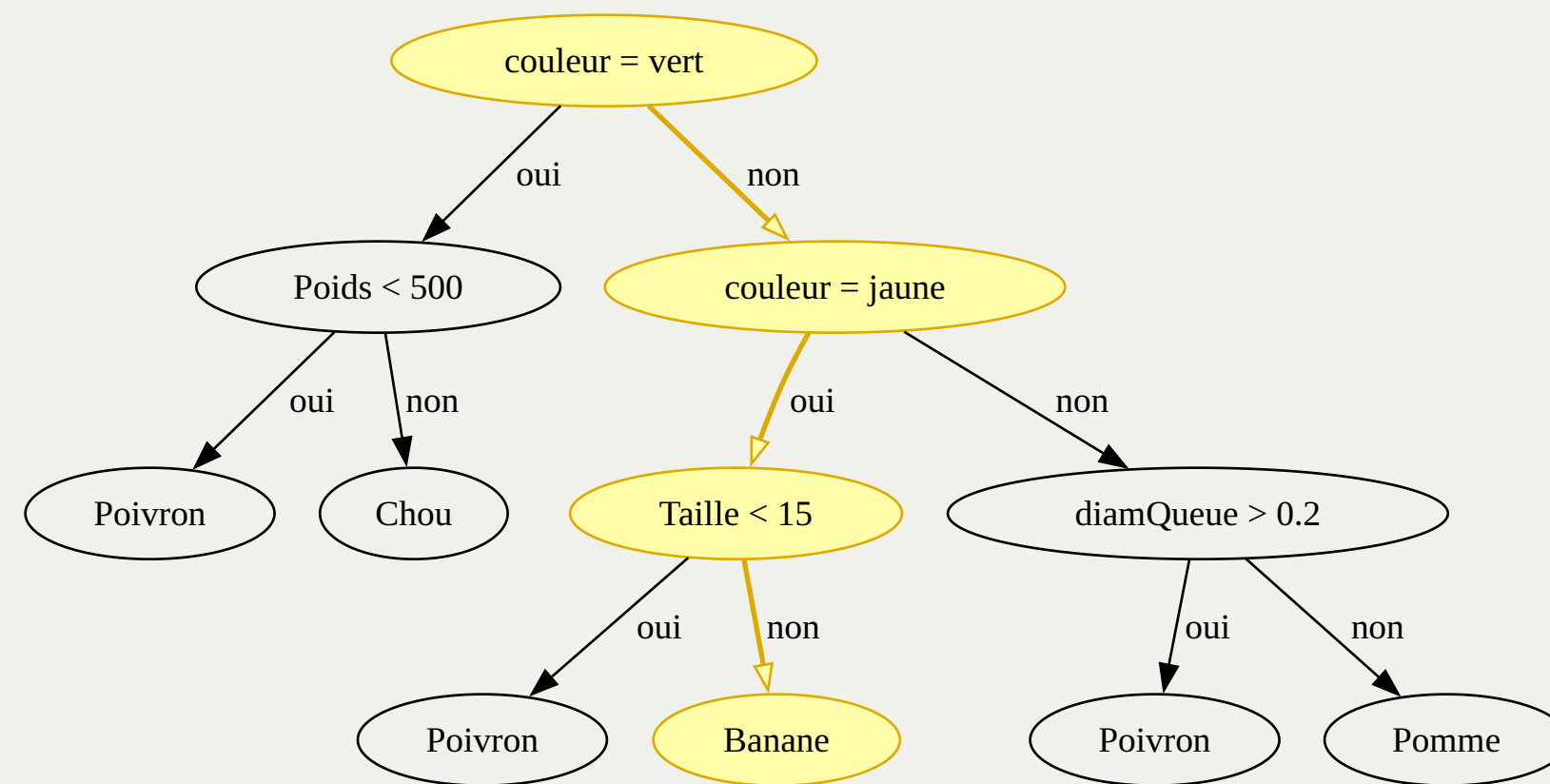
- On cherche une fonction de la forme $f(o; w_1, \dots, w_k)$.
- On cherche les meilleurs paramètres $w_1, \dots, w_k$ qui séparent les exemples.
- On vérifie sur des données tests que $f(o; w_1, \dots, w_k)$ généralise bien.

ARBRES DE DÉCISION

Exemple

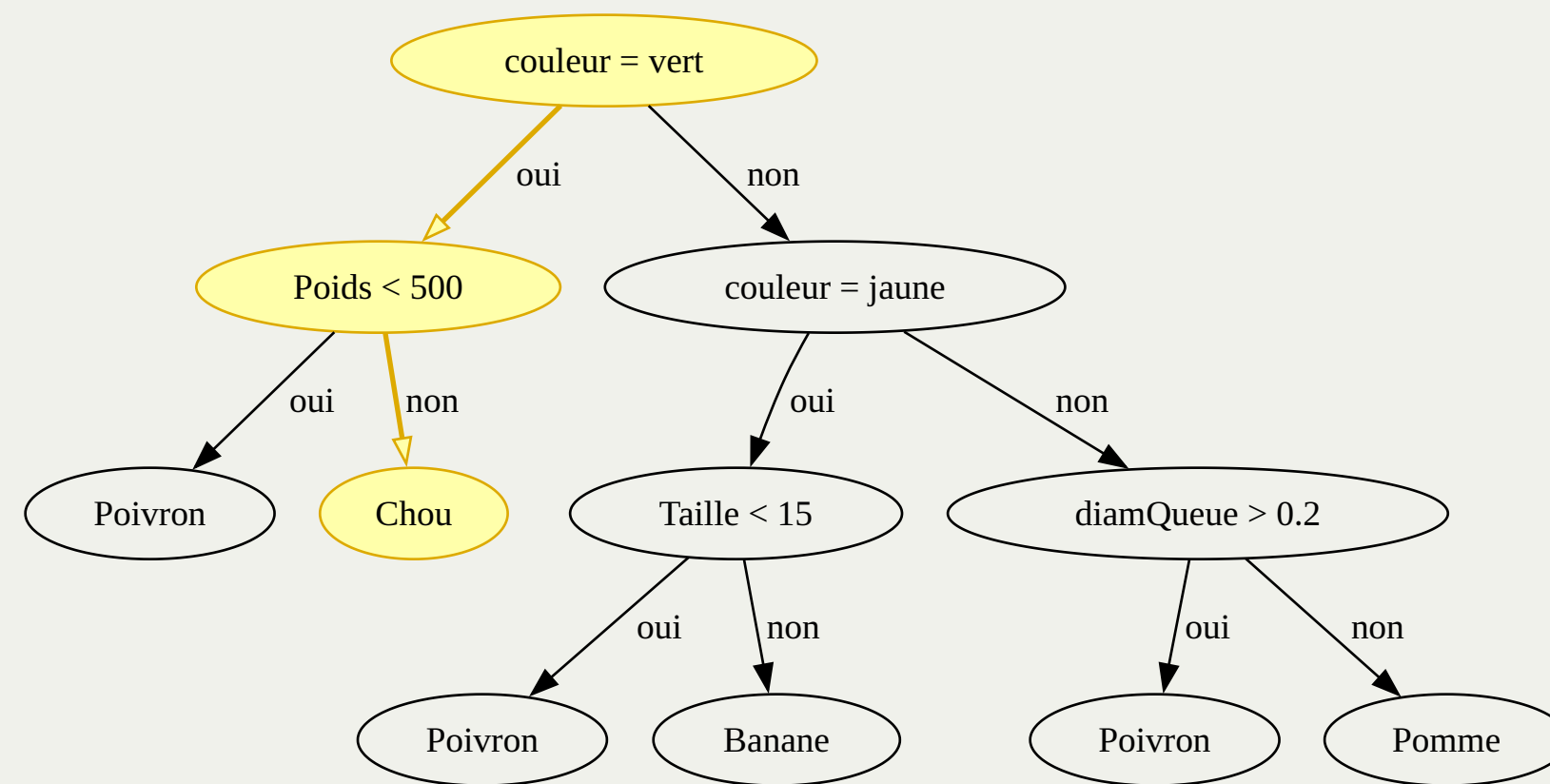


Exemple



`{"couleur": "jaune", "poids": "200g", "taille": "17cm"}`

Exemple



`{"couleur": "vert", "poids": "750g", "taille": "20cm"}`

Construction

- Supposons qu'on a un ensemble d'exemples $D = \{(x_i, c_i)\}_i$ où x_i est un vecteur de valeurs et c_i sa classe.
- On peut poser des questions $x[j] < v$ ou $x[j] = v$. Chaque question découpe le jeu de données en deux jeux D_0, D_1 .

Quelle est la meilleure question ?

- On cherche une question “clivante” qui sépare les classes.
- On veut une fonction qui est grande si le nombre d'objet dans chaque classe est similaire, et grande sinon (ie certaines classes sont plus représentée).

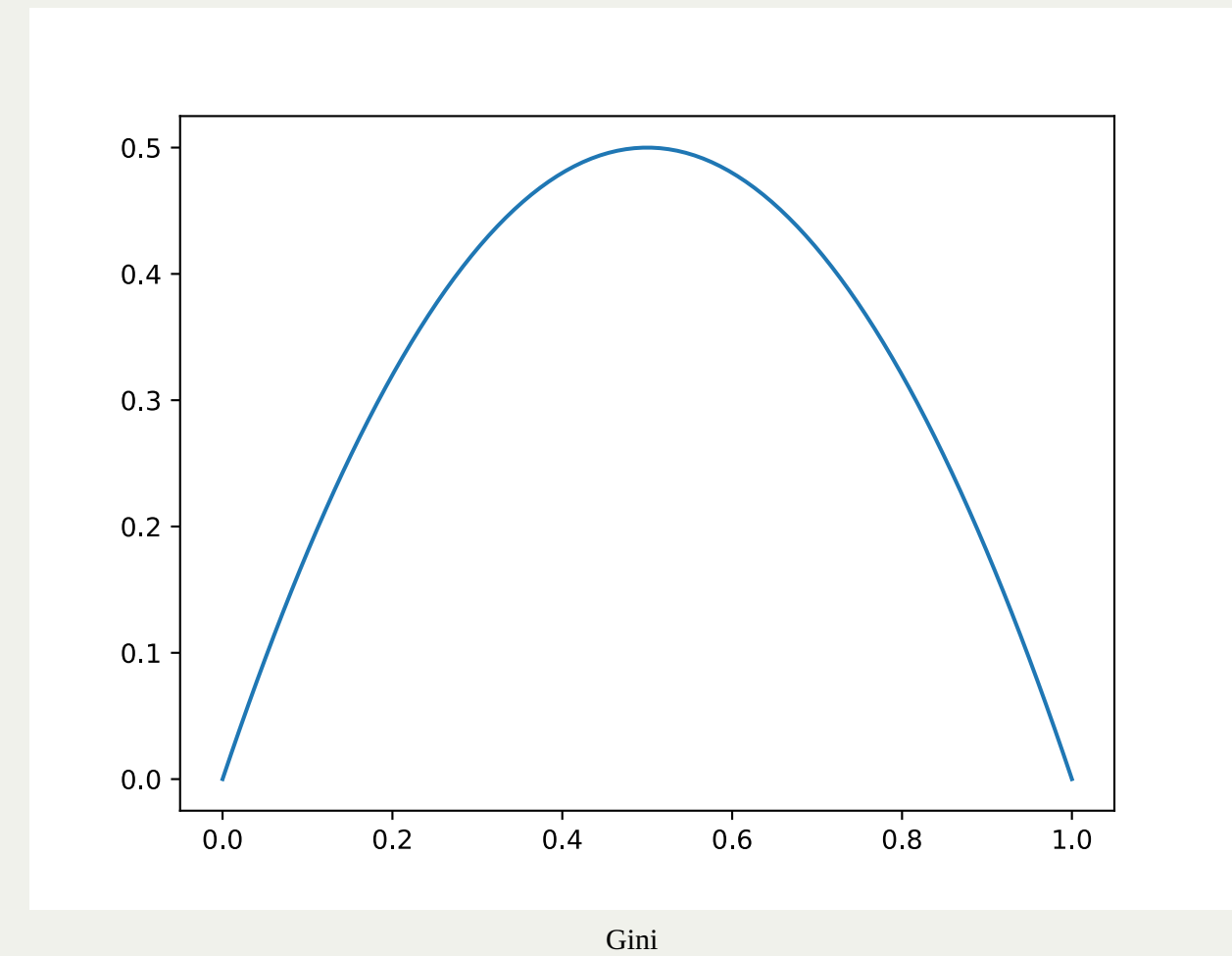
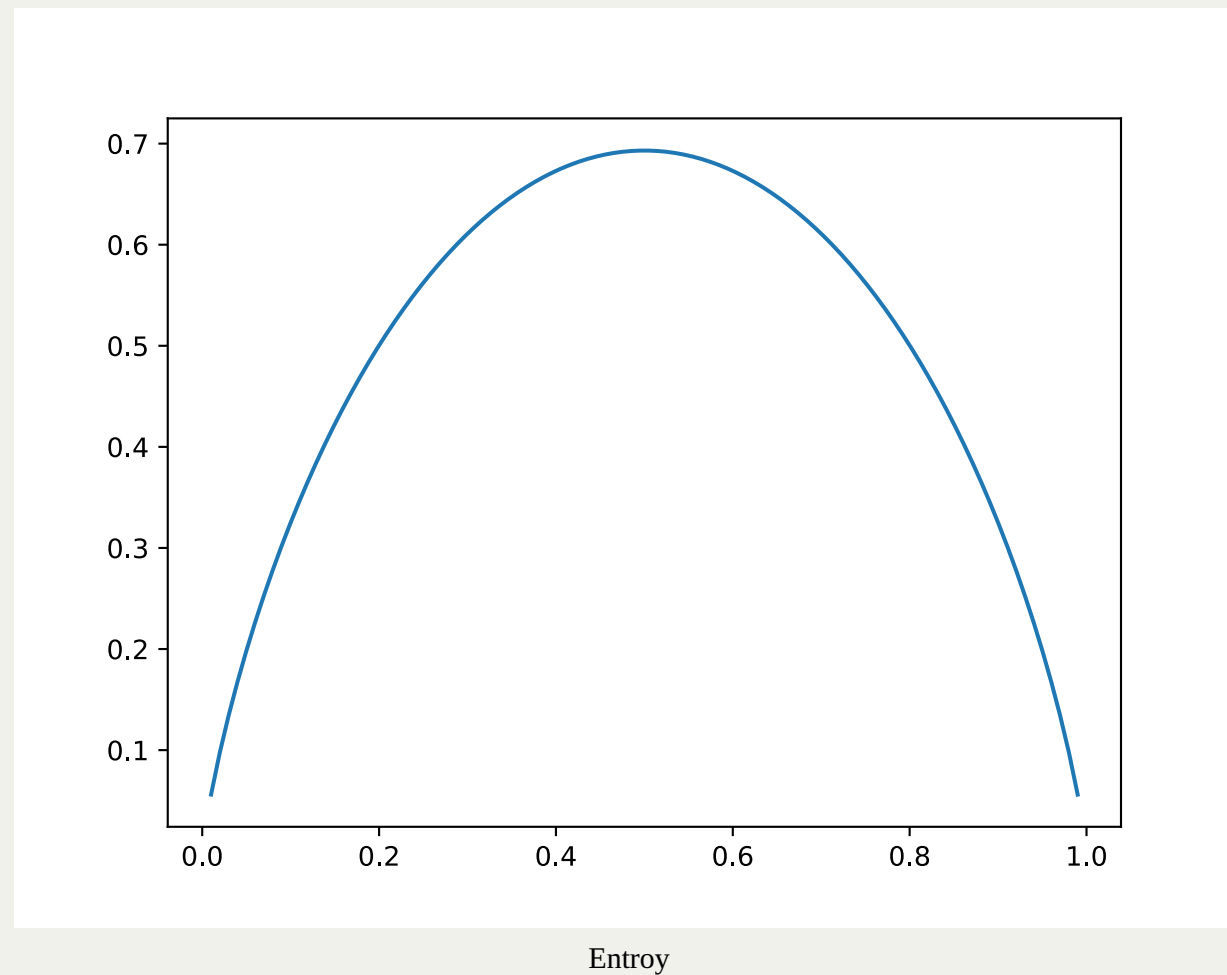
Fonctions de mélange

On note $D_c \subseteq D$ est l'ensemble des exemples de classe $c \in C$:

- Fonction de Gini: $g(D) = 1 - \sum_c (|D_c|/|D|)^2$.
- Entropie: $h(D) = -\sum_c (|D_c|/|D|) \log(|D_c|/|D|)$.

Dans le cas où on a juste deux classes:

- $g(D) = 1 - x^2 - (1 - x)^2$ et
- $h(D) = -x \log(x) - (1 - x) \log(1 - x)$ où $x = |D_0|/|D|$.



Gain

Si F est une fonction de mélange et Q une question séparant l'ensemble de données en D_0^Q (où Q est fausse) et D_1^Q (où Q est vraie) alors le gain est :

$$gain(D, Q) = F(D) - (|D_0^Q|/|D|)F(D_0^Q) - (|D_1^Q|/|D|)F(D_1^Q)$$

Exemple

$gini(D) = 1/2$

x	y	classe
1	6	oui
1	4	non
2	5	oui
4	8	non
6	5	oui
7	2	non

Exemple

$gini(D) = 1/2$

x	y	classe
1	6	oui
1	4	non
2	5	oui
4	8	non
6	5	oui
7	2	non

Question: $x < 4$.

- $gini(D_0) = 1 - (2/3)^2 - (1/3)^2 = 4/9$
- $gini(D_1) = 1 - (2/3)^2 - (1/3)^2 = 4/9$
- $gain(Q_1) = (1/2) - (3/6) \times (4/9) - (3/6) \times (4/9) = -5/18$

Example

$gini(D) = 1/2$

x	y	classe
1	6	oui
1	4	non
2	5	oui
4	8	non
6	5	oui
7	2	non

Question: $x < 4$.

- $gini(D_0) = 1 - (2/3)^2 - (1/3)^2 = 4/9$
- $gini(D_1) = 1 - (2/3)^2 - (1/3)^2 = 4/9$
- $gain(Q_1) = (1/2) - (3/6) \times (4/9) - (3/6) \times (4/9) = -5/18$

Question: $y < 5$.

- $gini(D_0) = 1 - (3/4)^2 - (1/4)^2 = 3/8$
- $gini(D_1) = 1 - 1^2 - 0^2 = 0$
- $gain(Q_2) = (1/2) - (4/6) \times (3/8) - (2/6) \times 0 = 1/4$

Exemple

$gini(D) = 1/2$

x	y	classe
1	6	oui
1	4	non
2	5	oui
4	8	non
6	5	oui
7	2	non

Question: $x < 4$.

- $gini(D_0) = 1 - (2/3)^2 - (1/3)^2 = 4/9$
- $gini(D_1) = 1 - (2/3)^2 - (1/3)^2 = 4/9$
- $gain(Q_1) = (1/2) - (3/6) \times (4/9) - (3/6) \times (4/9) = -5/18$

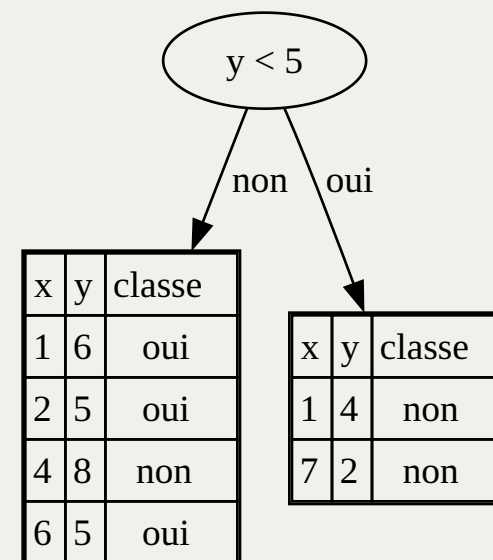
Question: $y < 5$.

- $gini(D_0) = 1 - (3/4)^2 - (1/4)^2 = 3/8$
- $gini(D_1) = 1 - 1^2 - 0^2 = 0$
- $gain(Q_2) = (1/2) - (4/6) \times (3/8) - (2/6) \times 0 = 1/4$

Q_2 a un meilleur gain que Q_1 , on la choisit.

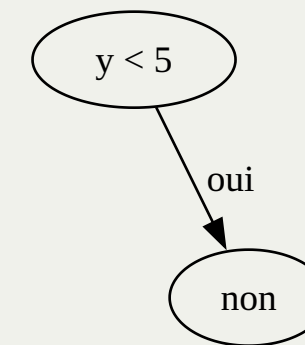
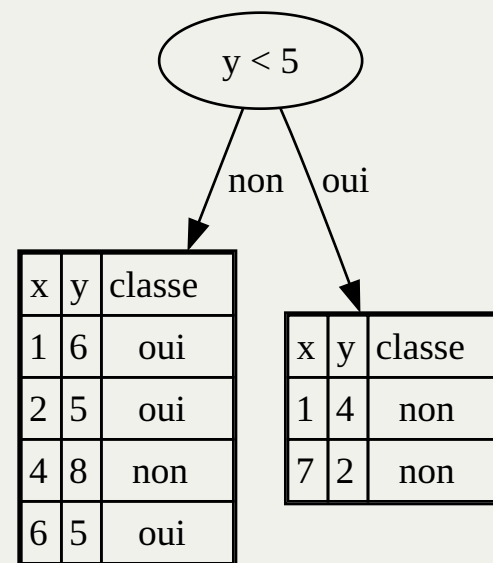
Exemple (cont.)

On fait une récursion sur chaque enfant :



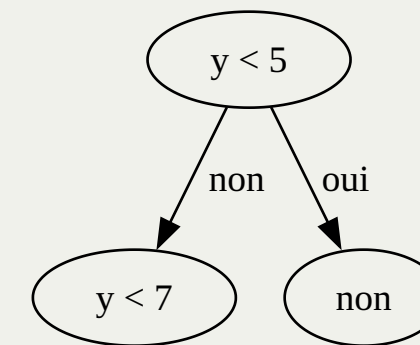
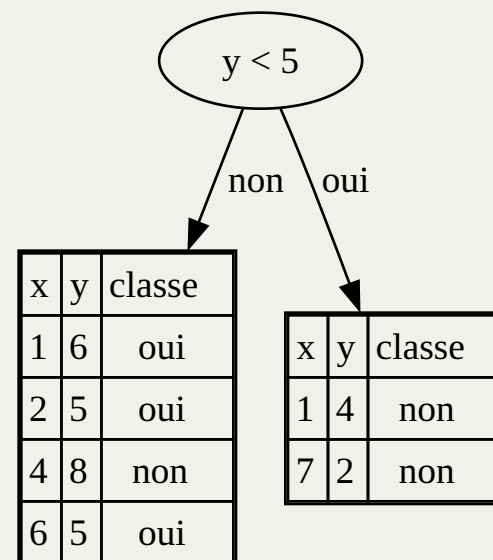
Exemple (cont.)

On fait une récursion sur chaque enfant :



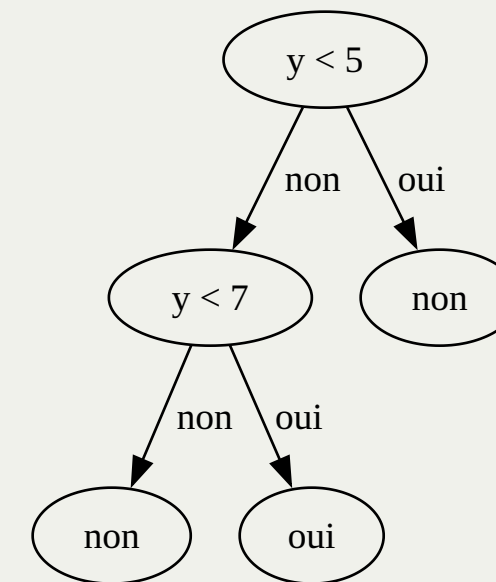
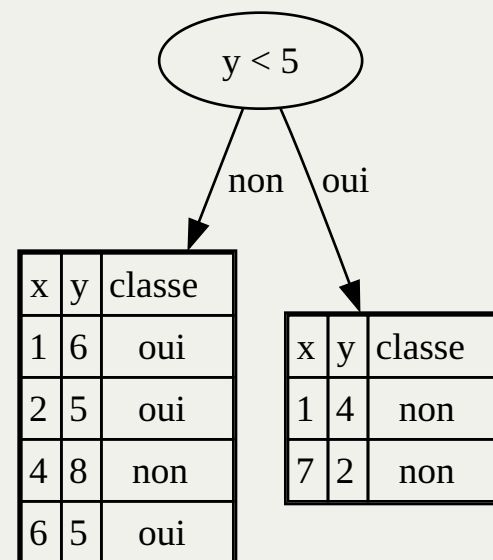
Exemple (cont.)

On fait une récursion sur chaque enfant :



Exemple (cont.)

On fait une récursion sur chaque enfant :



Défis

- Quand s'arrêter ?
 - quand il n'y a plus qu'une classe
 - quand il y a une classe représentée à plus de 90%
 - quand il y a une classe représentée à plus de **X**%
- Éviter le sur-apprentissage.
 - Élaguer les arbres en utilisant un ensemble de *validation*.
 - Borner la profondeur de l'arbre.
- Approche “glouton” non optimale.
 - Peut-être qu'une question moins bonne localement simplifie *après*.
 - NP-dur de construire l'arbre optimal pour un jeu de données.

+/- des arbres de décisions

Avantages

- Rapides
- Peu coûteux à entraîner.
- **Explicables** : voir [pyxai](#)
 - Pourquoi l'objet a été classé ainsi ?
 - Trouver un objet “proche” classé autrement.
- Peu découvrir des relations logiques entre les features.

Désavantages

- Grande variance aux données d'entraînement, risque de surapprentissage.
- Mauvais avec les événements rares.
- Ne teste pas de séparations linéaires; ne voit pas des relations polynomiales entre les features.
- Ne marche pas avec les features ayant beaucoup de valeurs possible (e.g., code postaux).
- Grande dimensions.

Aller plus loin

- **Random Forest :**
 - On entraîne (indépendamment) plusieurs arbres de décisions $T_1, \dots, T_k$.
 - On décide à la majorité : on dit que la classe c de o est la classe trouvée par le plus grand nombre d'arbre.
 - Diversité dans l'entraînement : on réduit les tests possibles pour éviter que $T_1, \dots, T_k$ ne soient trop proches.
- **Boosted trees:**
 - On entraîne des petits arbres $T_1, \dots, T_k$ avec des poids $w_1, \dots, w_k$.
 - On trouve la classe avec une somme pondérée : $\sum_i w_i T_i$.
 - Intuitivement : w_i mesure l'expertise du classifieur T_i (voir [AdaBoost](#)).