

Introduction à l'Apprentissage Automatique

Cécile Capponi, Farah Cherfaoui, Rémi Eyraud
(avec l'aide inestimable de l'équipe QARMA du LIS)

L3 Informatique
2019-2020



Plan de cette intervention

1. L'apprentissage, qu'est-ce que c'est ?
2. Protocole et mesure de qualité
3. Un exemple d'algorithme d'apprentissage
4. Déroulement de l'UE et évaluation

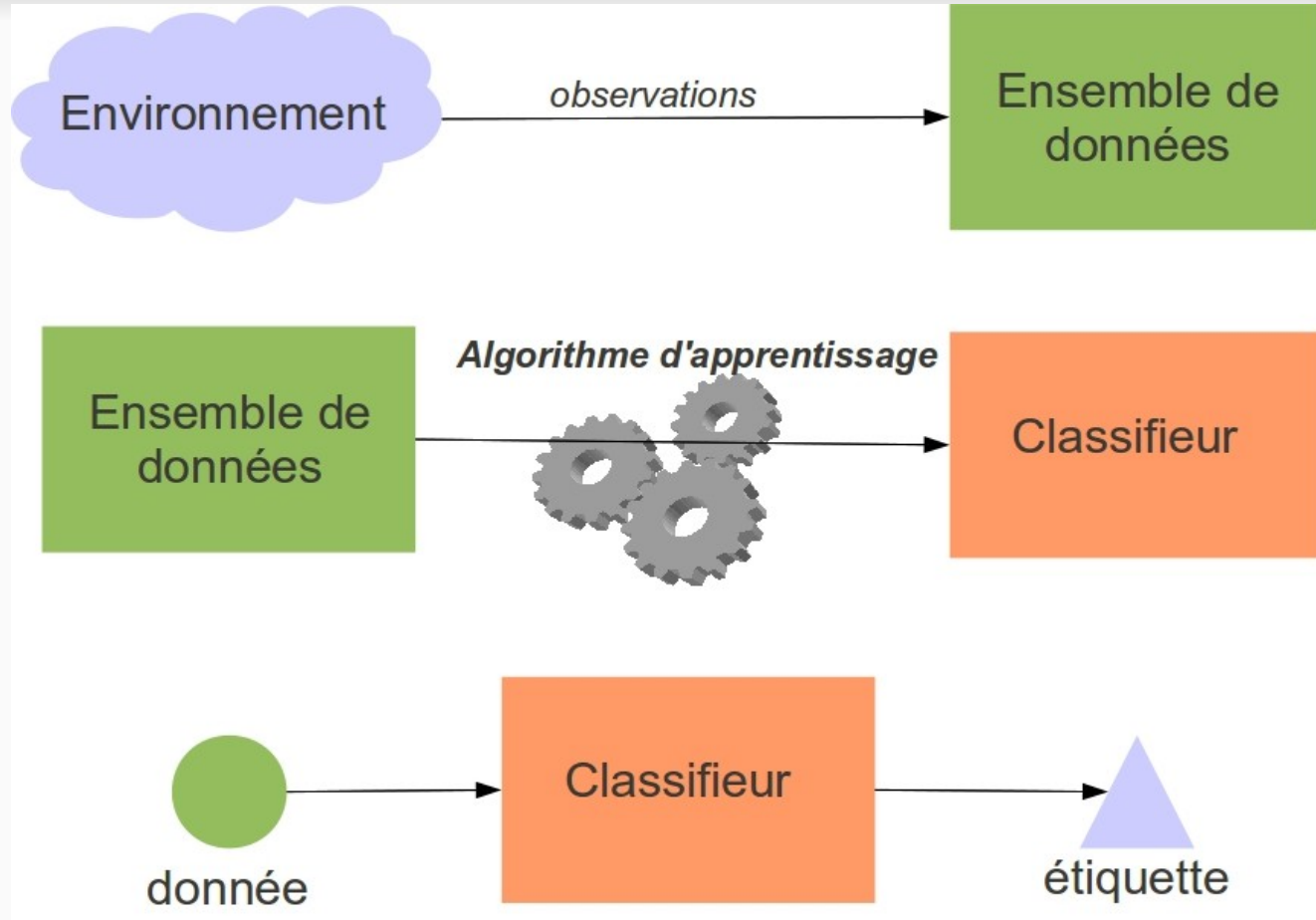
Qu'est-ce que l'apprentissage automatique



Apprentissage Automatique & Intelligence Artificielle & Science des Données

- 2019 : ces mots sont pratiquement synonymes
- Apprentissage machine :
 - Le **moteur** de la Science des Données
 - La partie aux succès récents et retentissants de l'IA

Schéma global de la classification supervisée



Apprentissage Automatique

- But : **extraire automatiquement** des données la connaissance permettant de prendre des bonnes décisions à l'avenir (sur d'autres / de nouvelles données)
- Moyen : inférer un modèle (mathématique...) qui capture les régularités (statistiques...) observables dans les **données d'apprentissage** : principe de **Généralisation**

Un exemple introductif

Alors que vous venez juste d'atterrir au Groeland pour la première fois, vous apercevez un mouton noir. Quelles conclusions en tirer ?

Un exemple introductif

Alors que vous venez juste d'atterrir au Groeland pour la première fois, vous apercevez un mouton noir. Quelles conclusions en tirer ?

- Il y a un et un seul mouton noir au Groeland (apprentissage par coeur, sous-généralisation)
- Certains moutons sont noirs au Groeland
- Tous les moutons du Groeland sont noirs (sur-généralisation)

Un apprentissage particulier : La classification supervisée

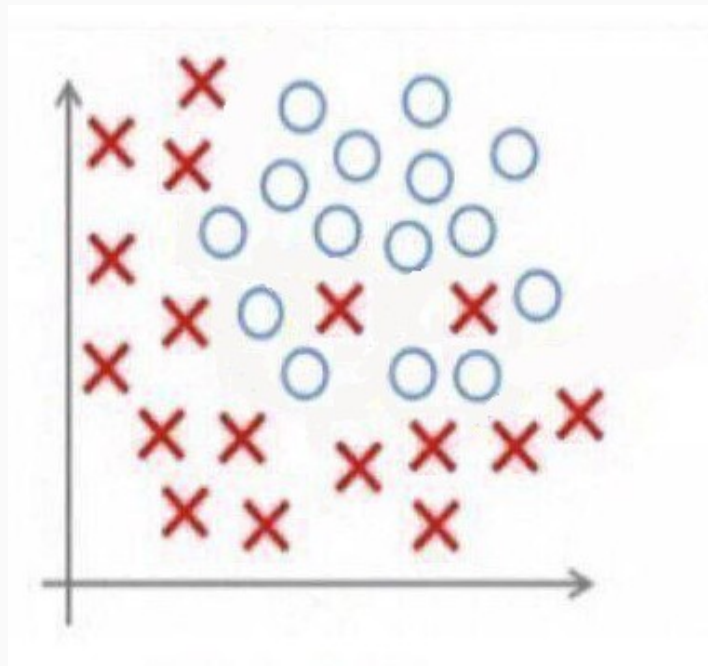
Classification : Le but est d'apprendre une fonction qui associe une classe à une description de donnée.

Supervisée : pour chaque donnée d'apprentissage on connaît sa classe.

On connaît donc le nombre (fini) de classes et leur sémantique à l'avance.

La classification supervisée : Exemple simple

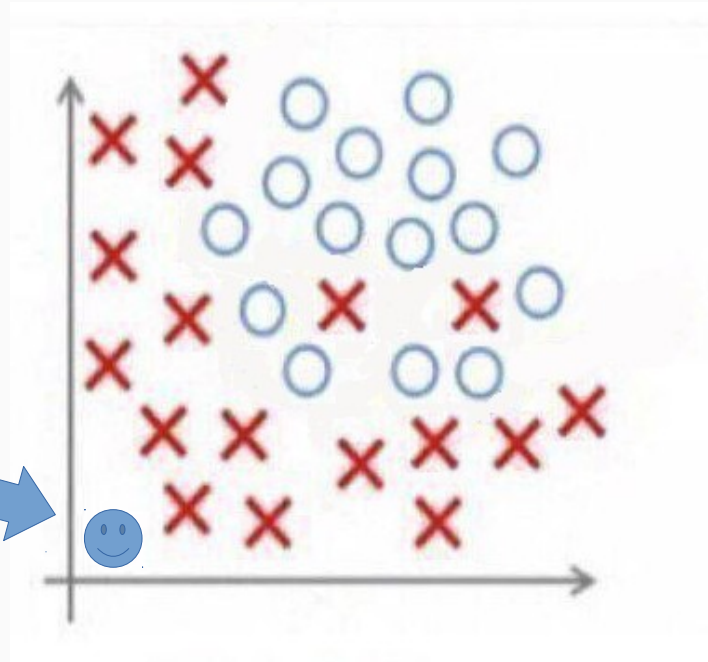
Données d'apprentissage :



La classification supervisée : Exemple simple

Nouvelle donnée
& Classification :

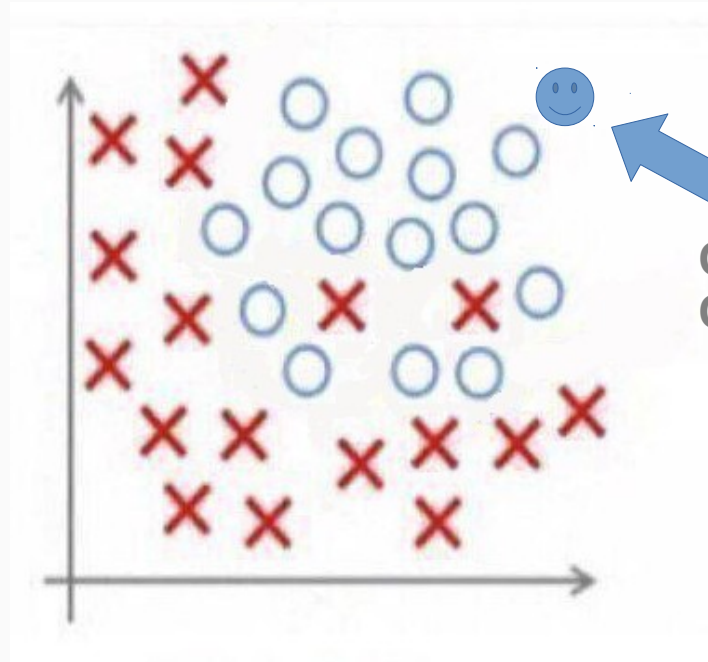
Quelle classe ?
Cercle ou croix ?



La classification supervisée :

Exemple simple

Nouvelle donnée
& Classification :

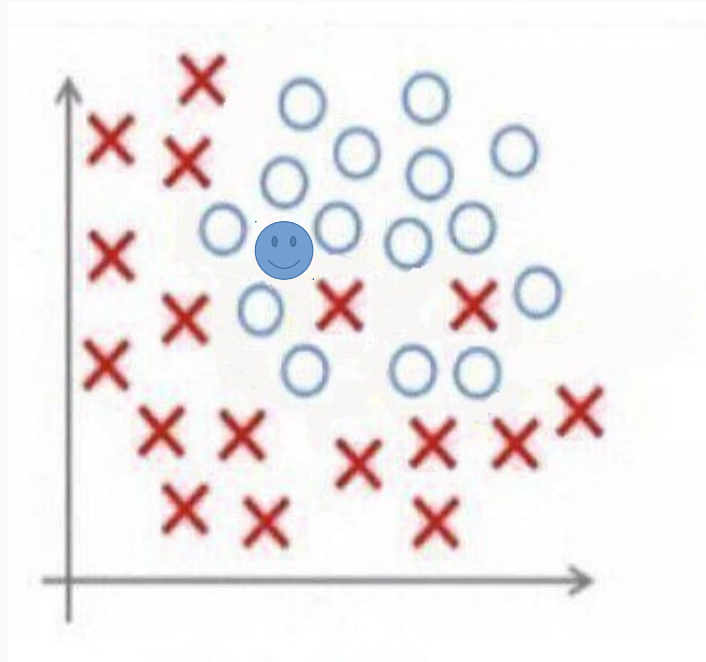


Quelle classe ?
Cercle ou croix ?

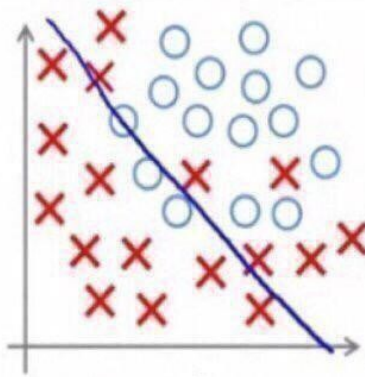
La classification supervisée : Exemple simple

Nouvelle donnée
& Classification :

Et pour cette donnée ?
Quelle classe ?
Cercle ou croix ?

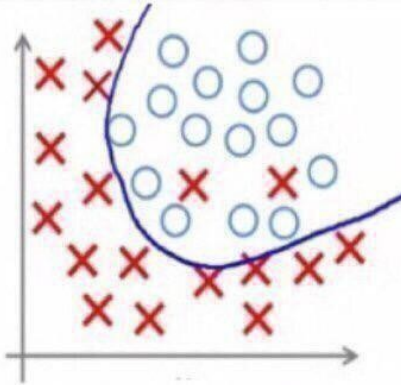


Sous-, Sur-, Correcte Généralisation

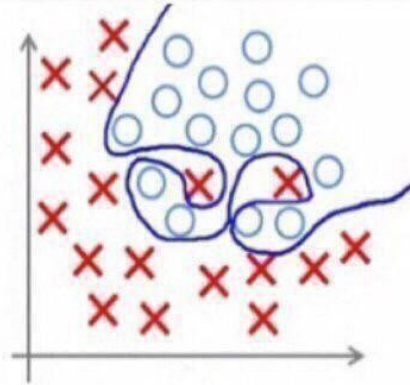


Under-fitting

(too simple to
explain the
variance)



Appropriate-fitting



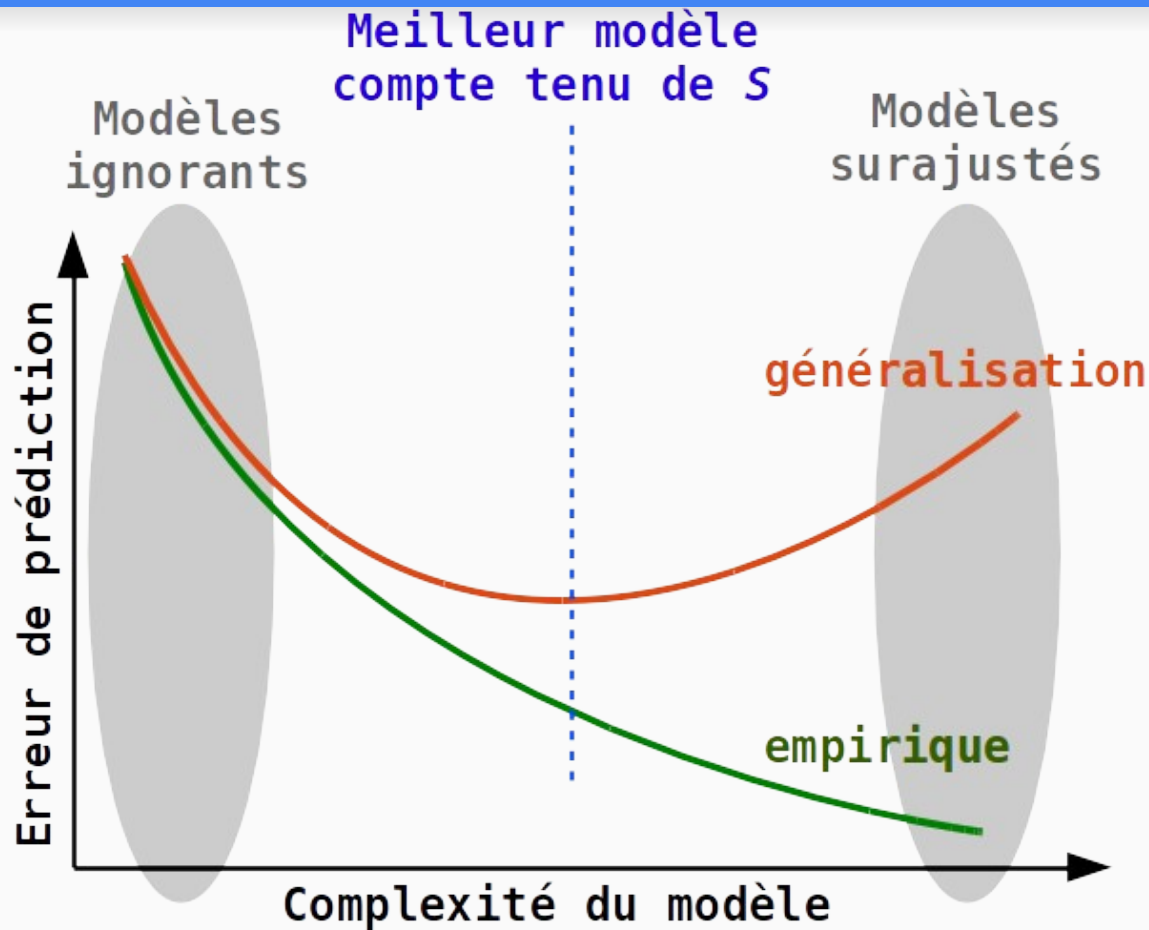
Over-fitting

(forcefitting -- too
good to be true)

Erreur empirique et Généralisation

Erreur empirique : erreur sur l'échantillon d'apprentissage S

Erreur en généralisation : Erreur sur toutes les données (inconnue)



Classification supervisée :

De vrais exemples

But : écarter automatiquement les SPAMs et autres messages non sollicités.

Données : des messages dont on sait s'ils sont des SPAMs ou non.

Objectif : construire un classifieur, capable d'attribuer une de ces deux classes à un nouveau message.

But : reconnaissance de chiffres manuscrits.

Données : des chiffres écrits sur une rétine de 16x16 pixels, associés à une classe parmi $\{0, 1, \dots, 9\}$

Objectif : attribuer la bonne classe (pattern recognition).

Modélisation de la classification supervisée

- **Attributs**(=variables=colonnes=features) :
un ensemble $\mathbf{X} = X_1 \times X_2 \dots \times X_d$ où chaque X_i est le domaine d'un attribut A_i symbolique ou numérique.
 - Ex.: $A_1 = \text{age}$, $X_1 = [0; 122]$, $A_2 = \text{fumeur}$, $X_2 = \{\text{oui}, \text{non}\}$
- **Classes**(=cible=target=label=etiquettes) :
Un ensemble fini de classes Y .
 - Ex.: $Y = \{\text{patient_à_risque}, \text{patient_sans_risque}\}$
- Une **variable aléatoire** $Z=(\mathbf{X}, Y)$ à valeurs dans $\mathbf{X} \times Y$.
 - Ex: le risque cardiaque est lié à l'âge et au fait de fumer

Modélisation de la classification supervisée

- Les **exemples**/données sont des couples (\mathbf{x}, y) de $\mathbf{X} \times Y$ tirés selon la distribution jointe :

$$P(Z=(\mathbf{x}, y)) = P(X=\mathbf{x})P(Y=y|X=\mathbf{x}).$$

- Un **échantillon** $\mathbf{S} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ est un ensemble fini d'exemples i.i.d. selon P .

Modélisation de la classification supervisée

Exemple
d'échantillon
d'apprentissage

**Ensemble des
attributs**

$\mathbf{X} = X_1 \times X_2$ avec

$X_1 = \text{age}$ et

$X_2 = \text{Fumeur}$

Classes Y

Valeurs
possibles :
{risque, pas
risque}

Ensemble des attributs		Classes Y
		Valeurs possibles : {risque, pas risque}
Age	Fumeur	Risque cardio
35	oui	risque
40	non	pas risque
60	oui	risque
20	oui	pas risque

Classifieur

- **Classifieur** : $f : \mathbf{X} \rightarrow Y$
 - Ex.: $f_1(\mathbf{x}) = \text{Si fumeur='oui' et age} > 59$
alors 'risque'
sinon 'pas risque'
- **Fonction de perte** (loss function) :
 $L(y_i, f(\mathbf{x}_i))$ égale à 0 si $y_i = f(\mathbf{x}_i)$ et à 1 sinon.
 - Ex.: $L(\text{'risque'}, f_1((35, \text{'oui'}))) = 1$
et $L(\text{'risque'}, f_1((65, \text{'oui'}))) = 0$

Modélisation de la classification supervisée (fin)

- Classifieur : $f : \mathbf{X} \rightarrow Y$
- Fonction de perte (loss function) :
 $L(y_i, f(\mathbf{x}_i))$ égale à 0 si $y_i = f(\mathbf{x}_i)$ et à 1 sinon
- La **fonction risque** (ou d'erreur) : espérance mathématique de la fonction de perte :

$$\mathbf{R(f)} = \int L(y, f(\mathbf{x})) dP(\mathbf{x}, y) = \int_{y \neq f(\mathbf{x})} dP(\mathbf{x}, y) = \mathbf{P(y \neq f(x))}$$

- Le **problème général de la classification supervisée** s'écrit :
Etant donné un échantillon $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ trouver un classifieur f
dont le risque $R(f)$ est le plus petit possible

Ce qu'il ne faut jamais oublier

- Le problème peut être non déterministe
(= plusieurs classes y peuvent correspondre au même vecteur \mathbf{x})
- Les données peuvent être bruitées
(= pas le bon y attribué à un \mathbf{x} et/ou les différentes valeurs composant \mathbf{x} ne sont pas bonnes)
- L'espace de description n'est pas suffisant pour décrire la complexité du problème auquel on s'attaque
(= chaque \mathbf{x} ne comporte pas suffisamment d'information pour permettre de trouver le y)

Un exemple d'algorithme
d'apprentissage :

Le classifieur naïf de Bayes

Règle de Bayes et optimalité

- Si on cherche à classer \mathbf{x} , quel est le meilleur y à lui attribuer ?

Réponse : celui qui maximise $P(y \mid \mathbf{x})$! C'est-à-dire le y le plus probable quand on connaît \mathbf{x} .

Mathématiquement : $f_{\text{Bayes}}(\mathbf{x}) = \operatorname{argmax}_y P(y \mid \mathbf{x})$

- C'est ce que l'on appelle la **règle de Bayes** et on peut prouver que c'est le meilleur classifieur possible (= celui dont le taux d'erreur en généralisation est le plus faible possible)

Tristesse immense & espoir infini

- La règle de Bayes n'est pas calculable à partir d'un ensemble de données.
- Le but de tout algorithme d'apprentissage est donc d'inférer un classifieur (presque) aussi bon que celui de la règle de Bayes.
- Bonne nouvelle : il existe des dizaines de (très) bons algorithmes pour faire ça ! Et il en reste très certainement de nouveaux à trouver.

Le classifieur naïf de Bayes

- La règle de Bayes peut se réécrire (à l'aide de la formule de Bayes) :

$$\operatorname{argmax}_y P(y|\mathbf{x}) = \operatorname{argmax}_y \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} = \operatorname{argmax}_y P(\mathbf{x}|y)P(y)$$

- $P(y)$ peut être estimé en calculant les fréquences de chaque classe dans l'échantillon d'apprentissage
- Plus difficile pour $P(\mathbf{x}|y)$ (vraisemblance) : si $\mathbf{x} = (x_1, x_2, \dots, x_d)$ alors il nous faut connaître $P(\{x_1, x_2, \dots, x_d\}|y)$ ce qui n'est pas possible, en particulier si ces valeurs ne sont pas indépendantes.

Le classifieur naïf de Bayes (2)

- Le problème (pour $d = 2$) :
 $P(\{x_1, x_2\}|y) = P(\{x_1\}|y) P(\{x_2\}|y, x_1)$ et $P(\{x_1, x_2\}|y) = P(\{x_2\}|y) P(\{x_1\}|y, x_2)$
- En général, ni $P(\{x_2\}|y, x_1)$ ni $P(\{x_1\}|y, x_2)$ ne sont pas (raisonnablement) estimables à partir d'un ensemble de données
- **Classifieur naïf de Bayes** : hypothèse (forte) d'indépendance des attributs :
 $P(\{x_2\}|y, x_1) = P(\{x_2\}|y)$ et $P(\{x_1\}|y, x_2) = P(\{x_1\}|y)$
En d'autres termes : les attributs sont indépendants deux à deux

Le classifieur naïf de Bayes (3)

- Si on met les étapes toutes ensembles, le classifieur naïf de Bayes est :

$$\operatorname{argmax}_y P(y|\mathbf{x}) = \operatorname{argmax}_y P(\mathbf{x}|y) P(y) \approx \operatorname{argmax}_y P(y) \prod_{i=1}^d P(x_i|y)$$

- $P(y)$ est estimable à partir des données
- $P(x_i | y)$ l'est aussi facilement :
 - Si x_i prend des valeurs discrètes, il suffit de calculer les fréquences des valeurs pour chaque classes
 - Si x_i prend des valeurs continues, on suppose que ces valeurs correspondent à une gaussienne et on évalue la moyenne et la variance à l'aide des données d'apprentissage.

Classifieur naïf de Bayes : Exemple

$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	y
●	A	3	►
●	B	3	♂
●	A	3	►
●	A	2	♂
●	B	2	♂
●	A	3	►
●	B	3	♂
●	B	2	►
●	B	2	♂
●	A	2	♂
●	B	3	►
●	B	2	♂

$$P(\blacktriangleright) = 5/12 = 0.42$$

$$P(\text{♂}) = 7/12 = 0.58$$

$$P(\text{●} | \blacktriangleright) = 2/5 = 0.4$$

$$P(\text{●} | \blacktriangleright) = 2/5 = 0.4$$

$$P(\text{●} | \blacktriangleright) = 1/5 = 0.2$$

$$P(\text{●} | \text{♂}) = 2/7 = 0.29$$

$$P(\text{●} | \text{♂}) = 4/7 = 0.57$$

$$P(\text{●} | \text{♂}) = 1/7 = 0.14$$

$$P(A | \blacktriangleright) = 3/5 = 0.6$$

$$P(B | \blacktriangleright) = 2/5 = 0.4$$

$$P(A | \text{♂}) = 2/7 = 0.29$$

$$P(B | \text{♂}) = 5/7 = 0.71$$

$$P(3 | \blacktriangleright) = 4/5 = 0.8$$

$$P(2 | \blacktriangleright) = 1/5 = 0.2$$

$$P(3 | \text{♂}) = 2/7 = 0.29$$

$$P(2 | \text{♂}) = 5/7 = 0.71$$

$$P(\blacktriangleright | \text{●} A 3) \Rightarrow P(\text{●} A 3 | \blacktriangleright) P(\blacktriangleright) = P(\text{●} | \blacktriangleright) P(A | \blacktriangleright) P(3 | \blacktriangleright) P(\blacktriangleright) = 0.04$$

$$P(\text{♂} | \text{●} A 3) \Rightarrow P(\text{●} A 3 | \text{♂}) P(\text{♂}) = P(\text{●} | \text{♂}) P(A | \text{♂}) P(3 | \text{♂}) P(\text{♂}) = 0.006$$

Classifieur naïf de Bayes en python

Ceci est un commentaire – importation de la fonction load_iris de la librairie scikit-learn :

from sklearn.datasets **import** load_iris

récupération des données Iris

donnees = load_iris()

Stockage de la matrice de description des données :

X = donnees.data

Stockage des classes de chaque donnée :

y = donnees.target

importation de la fonction découpant les données en test et train :

from sklearn.model_selection **import** train_test_split

Génération aléatoire des échantillons (X_train, y_train) et (X_test, y_test) :

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

Classifieur naïf de Bayes en python

importation de la librairie pour le classifieur naive Bayes :

from sklearn.naive_bayes **import** GaussianNB

Création d'une instance de ce classifieur :

classifieur = GaussianNB()

Apprentissage sur les données d'entraînement :

classifieur.fit(X_train, y_train)

Utilisation du classifieur appris sur les données de test :

y_predits = classifieur.predict(X_test)

Calcul du taux de réussite :

from sklearn.metrics **import** accuracy_score

print("Taux de réussite : ", accuracy_score(y_test,y_predits))

Validation d'un apprentissage

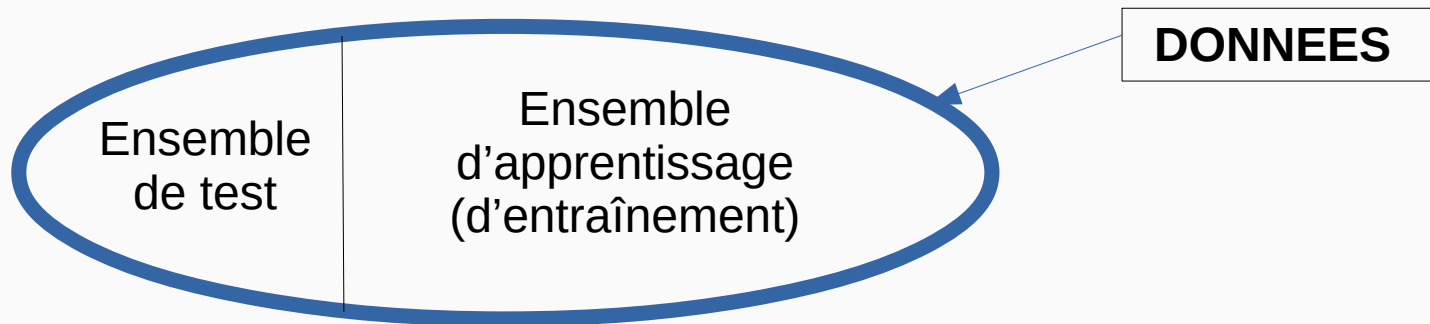
Mesures de qualité

L'apprentissage en pratique

- On dispose d'un **échantillon d'apprentissage** S qu'on suppose i.i.d.
- On recherche une fonction h de **classification** dont le **risque** est le plus faible possible.
- Il existe toujours une fonction f_{\min} de risque minimal... inaccessible !

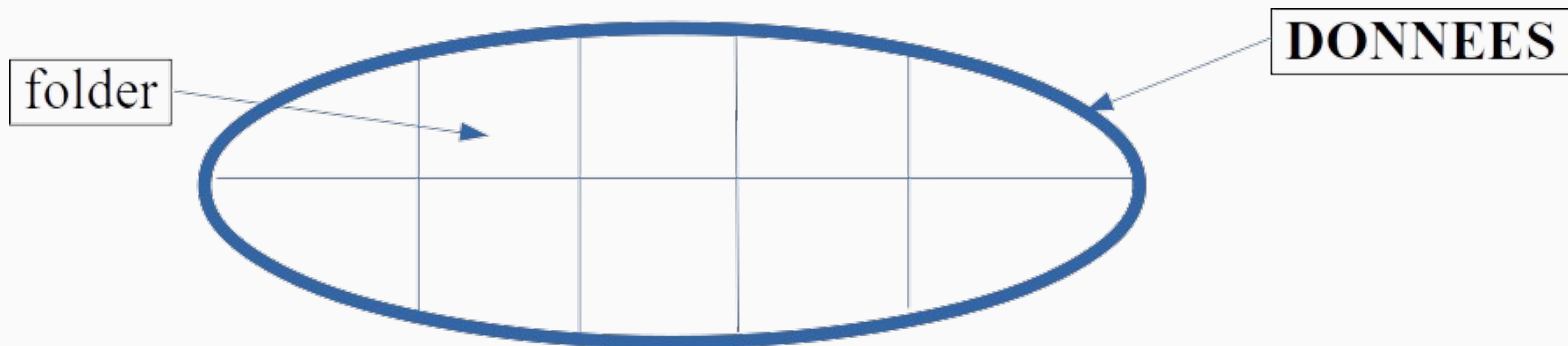
Validation empirique d'un apprentissage

- Plusieurs méthodes permettent de valider (ou d'infirmar) la qualité d'un processus d'apprentissage.
- Une des approches consiste à n'utiliser qu'une partie des données pour apprendre et à se servir des autres données pour tester le résultat.
- Différentes mesures permettent alors de comparer des processus (taux d'erreur, F-score, etc.)



Validation croisée (cross-validation)

- La **cross-validation** est une généralisation de la méthode précédente.
- Elle consiste à diviser les données en **c folders**, à en enlever un pour l'apprentissage puis à l'utiliser pour la phase de test.
- Le processus est ensuite réitéré sur chaque folder
- L'**erreur moyenne** tend alors vers l'erreur en généralisation (estimateur non-biaisé).



Mesure de qualité : classification binaire supervisée

- **Matrice de confusion :**

	Classé +	Classé -
Exemple +	V_p	F_n
Exemple -	F_p	V_n

- **Taux d'erreur** = 1 - taux de réussite (*accuracy*) =
$$\frac{F_p + F_n}{F_p + F_n + V_p + V_n}$$
- L'erreur (ou taux d'erreur) ne fait pas de distinction entre les erreurs : pas toujours une bonne mesure de qualité d'un apprentissage

Déroulement de l'UE et évaluation

Pédagogie 'innovante' : UE sur projet

- Aujourd'hui : TP de prise en main de python et de LA bibliothèque d'apprentissage automatique (scikit-learn), présentation du projet
- Ensuite : travail par équipe sauf le matin du 03 février où il y aura un debriefing de mi-UE
- Soutenances le 02 mars

Projet : classification Automatique d'images

- **Comment** : En groupe de 3 à 5 personnes
- **Objectif** : trouver et coder **en python** une solution à un problème de classification d'images (des détails – et les données – cet après-midi)
- **Moyens** : débrouillez-vous !
 - 1 bon livre : Apprentissage A😊ficiel, Cornuéjols & Miclet
 - 1 MOOC : <https://www.coursera.org/learn/machine-learning>
 - ... et vos profs sont dans le coin !

Evaluation

- **Rapport** : Maximum 4 pages A4 lisibles (+ le code en annexe)
- **Soutenance** :
 - Le 02 mars
 - Par groupe
 - Obligation d'expliquer les algorithmes d'apprentissage automatique utilisés