

Intelligence artificielle

Objectifs du cours

- Avoir un discours « juste » sur l'intelligence artificielle
- Connaître les grands domaines de l'intelligence artificielle
 - Savoir illustrer ces domaines
- Comprendre les difficultés et les principes des points suivants

Introduction



L'intelligence artificielle,
1972

- Avant d'aborder l'intelligence artificielle, il nous faut débroussailler la notion d'« intelligence »
- Comment définir l'intelligence ?
- Le terme « cognition » serait plus approprié !

Qu'est-ce que l'intelligence

- Perception ?
- Raisonnement ?
- Connaissances ?
- Compréhension ? Interprétation ?
- Mémoire ?
- Capacité d'apprendre ?
- Expertise ?
- Créativité ?
- Action ? Réaction ?
- Autonomie ? Collaboration ? Interactions ?

Qu'est ce que l'intelligence artificielle

- **Rendre l'ordinateur capable de ...**
- **Simuler** les capacités de ...
- **Donner l'illusion** des capacités de ...
 - Perception ?
 - Raisonnement ?
 - Connaissances ?
 - Compréhension ? Interprétation ?
 - Mémoire ?
 - Capacité d'apprendre ?
 - Expertise ?
 -

Objectifs de l'IA

- Construire des machines pour une large variété d'applications augmentant les capacités de résolution de problèmes « mal posés »
- Formaliser ce qu'est la cognition
- Mécaniser le cognition
- Utiliser des modèles computationnels pour comprendre des comportements complexes
- Rendre l'interaction avec les systèmes computationnels aussi simple qu'avec les humains

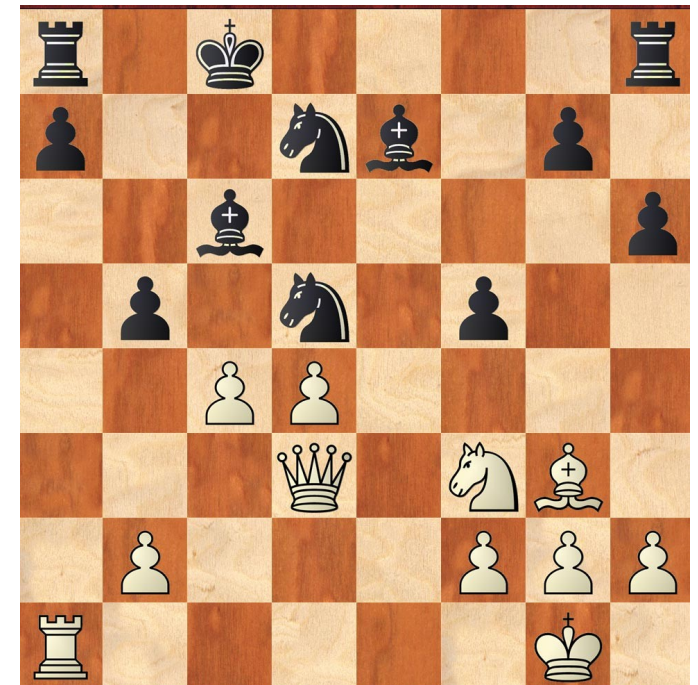
Exemples : raisonnement / diagnostique

- Aide à la surveillance et au diagnostic médical
- Difficulté
 - Structuration des données
 - Manipulation des connaissances (raisonnement inductif)
 - Construction d'une base de connaissance
 - Diagnostic en ligne



Exemples : raisonnement/décision/planification

- Jeux d'échec
- Problème énoncé vers 1950, résolu en 1997 (deep blue)
- Choix de l'action adéquate pour atteindre un but
- Difficulté :
 - le but a atteindre est loin de l'action immédiate
 - Le comportement de l'adversaire est incertain



Exemples : robotique

- Difficultés

- Intégration des capacités de perception, de raisonnement et d'action
- Autonomie
- Bouclage avec un environnement physique
 - très aléatoire
 - Imprécis
 - inconnus



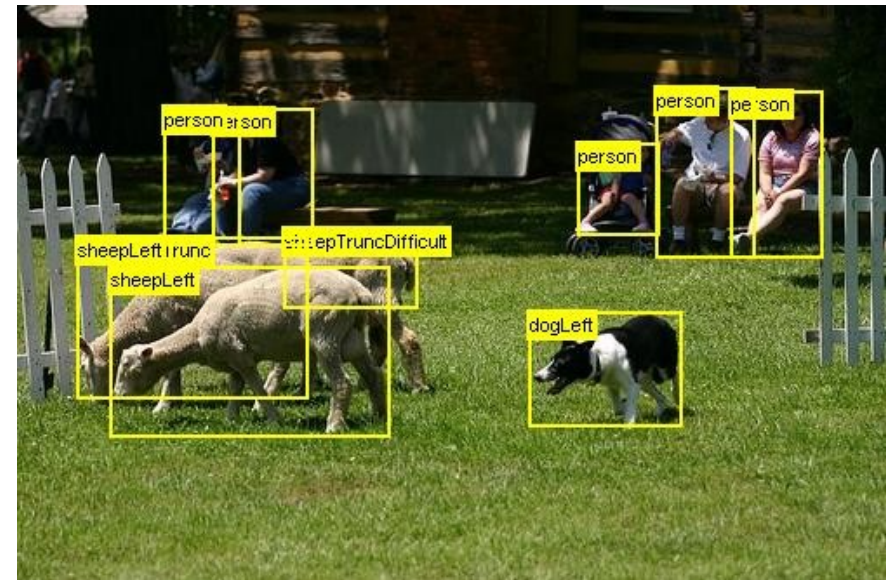
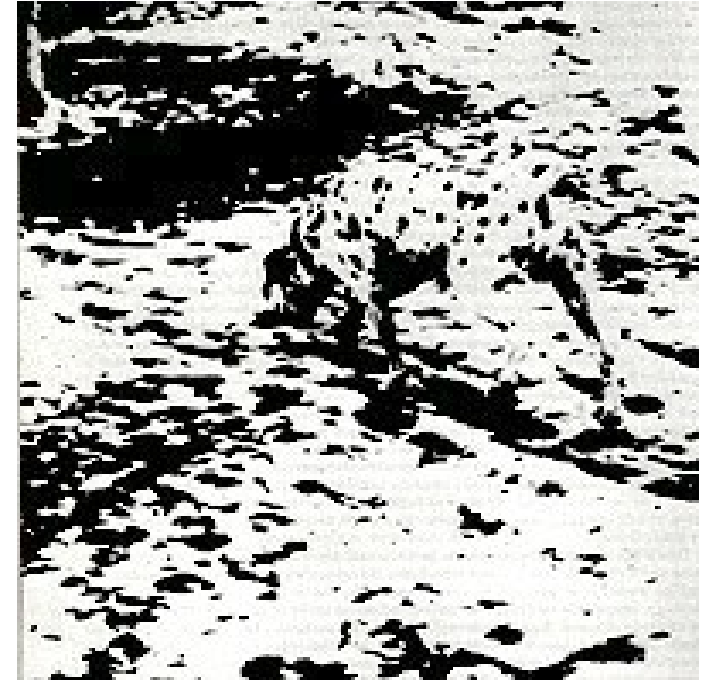
Exemples : robotiques / coopération / dialogue

- Résolution de problèmes par une cohorte de robots
- Difficultés
 - Planification des actions
 - Coopération entre les robots



Exemples : la perception

- Analyse et interprétation d'images
- Difficultés
 - Qu'est ce qui est important dans l'image ?
 - Localiser les éléments saillants dans une image
 - Identifier les « objets » dans l'images
- Pose la question de l'accès en *sens*

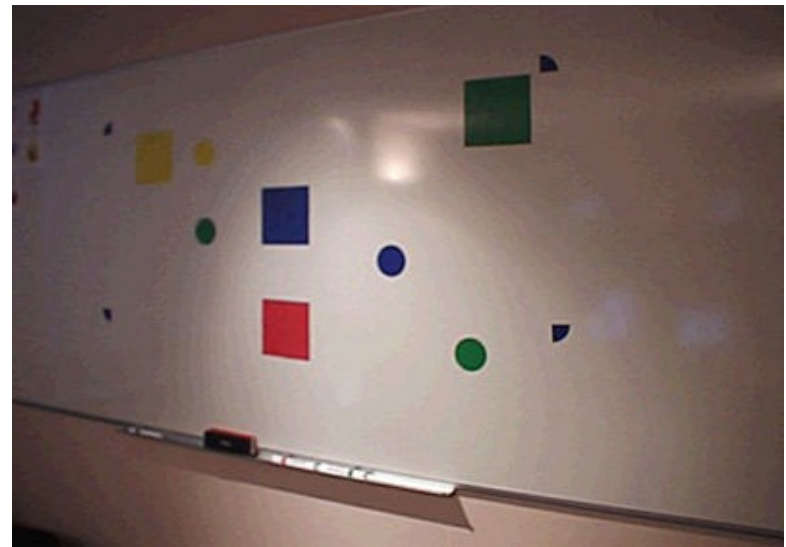
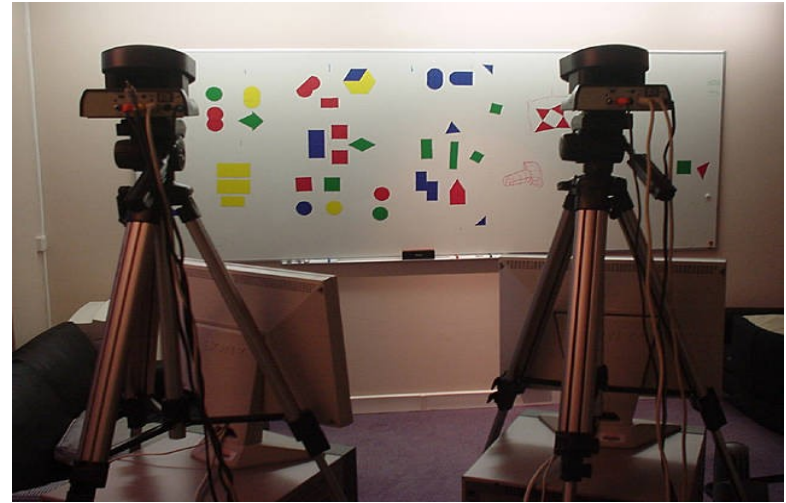


Exemples : apprentissage automatique

- Profilage des utilisateurs (pour leur rendre des services)
 - Moteur de recommandations de programmes TV
 - EUREKA !
 - Propose de « faire découvrir des programmes que les abonnés n'auraient pas déniché tout seul »
 - Google
 - Profilage des utilisateurs par rapport à leurs requêtes et leurs mail
 - Sélection de publicités
 - Commerce en ligne
 - Recommandation de livre/CD par rapport à vos achats précédents

Exemple : autonomie

- Talking heads, Luc Steels SONY



Exemples : systèmes distribués

- Simulation de foule
 - Application aux films
 - Évaluation des plans d'évacuation
- Difficulté
 - Donner l'impression d'une foule d'individus alors que les « agents » sont les mêmes
 - Objectif de « réalisme » du comportement



<http://www.massivesoftware.com/film.html>

Pourquoi l'intelligence artificielle ?

- Point de vue des **sciences cognitives**
 - Comprendre la cognition naturelle par des modèles informatiques
- Point de vue de **l'ingénieur**
 - Résoudre des problèmes difficiles par n'importe quelle méthode
- Point de vue du **théoricien**
 - Découvrir les lois universelles de la cognition

Bref historique

- Depuis l'Antiquité ... mécanisation du calcul
 - Bacon / Pascal / Descartes / Leibniz
 - Babbage & Ada de Lovelace / Boole
 - Réflexion sur les fondements des mathématiques
- Les pionniers de l'IA (~1936 - ~1956)
 - Turing : les fonctions calculables sont définies par une machine
 - Von Neumann : architecture des ordinateurs, réflexions sur les automates
 - McCulloch & Pitts (1943) : Premier modèle formel du neurone
 - Hebb (1949) : assemblées de neurones et règle d'apprentissage

Bref historique

- L'IA comme méthodes générales (~1956 - ~1968)
 - La pensée comme manipulation de représentations discrètes des connaissances: IA symbolique
 - Démonstrateurs de théorèmes (principe de résolution)
 - Résolveurs universels de problèmes (GPS)
 - Le système CHECKER
 - Premier connexionnisme
 - Le Perceptron [Rosenblatt, 1957-1962]
 - MAIS des échecs :
 - La traduction automatique
 - Les experts ne sont pas des experts universels !?
 - Le Perceptron est limité

Bref historique

- « Knowledge is power » (~1968 - ~1980)
 - Représentation des connaissances
 - Représentations structurées : réseaux sémantiques, scripts, schémas et frames (précurseurs des LO), ...
 - Des extensions de la logique : logiques non monotones, logiques temporelles, logique floue, ...
 - De nombreux systèmes « intelligents » : ARCH, AM, MAGGIE, BORIS, PLANNER, ...
 - Utilisation des connaissances/raisonnement : les systèmes experts
 - Problème
 - Comment acquérir toutes les connaissances nécessaires ?
 - Comment généraliser ces expériences (souvent appliquées à des « problèmes jouet ») ?

Bref historique

- Méthodes générales d'apprentissage (~1980 - ~1990)
 - Méthodes symboliques
 - De nombreuses réalisations et nouvelles techniques
 - Arbres de décision
 - Nouveaux principes
 - Espace des versions
 - Renouveau du connexionnisme et du mouvement subsymbolique
 - Hopfield (1982)
 - Le Perceptron Multi-Couches (1985)
 - IA distribuée (Algorithmes Génétiques, Vie Artificielle, ...)

Bref historique

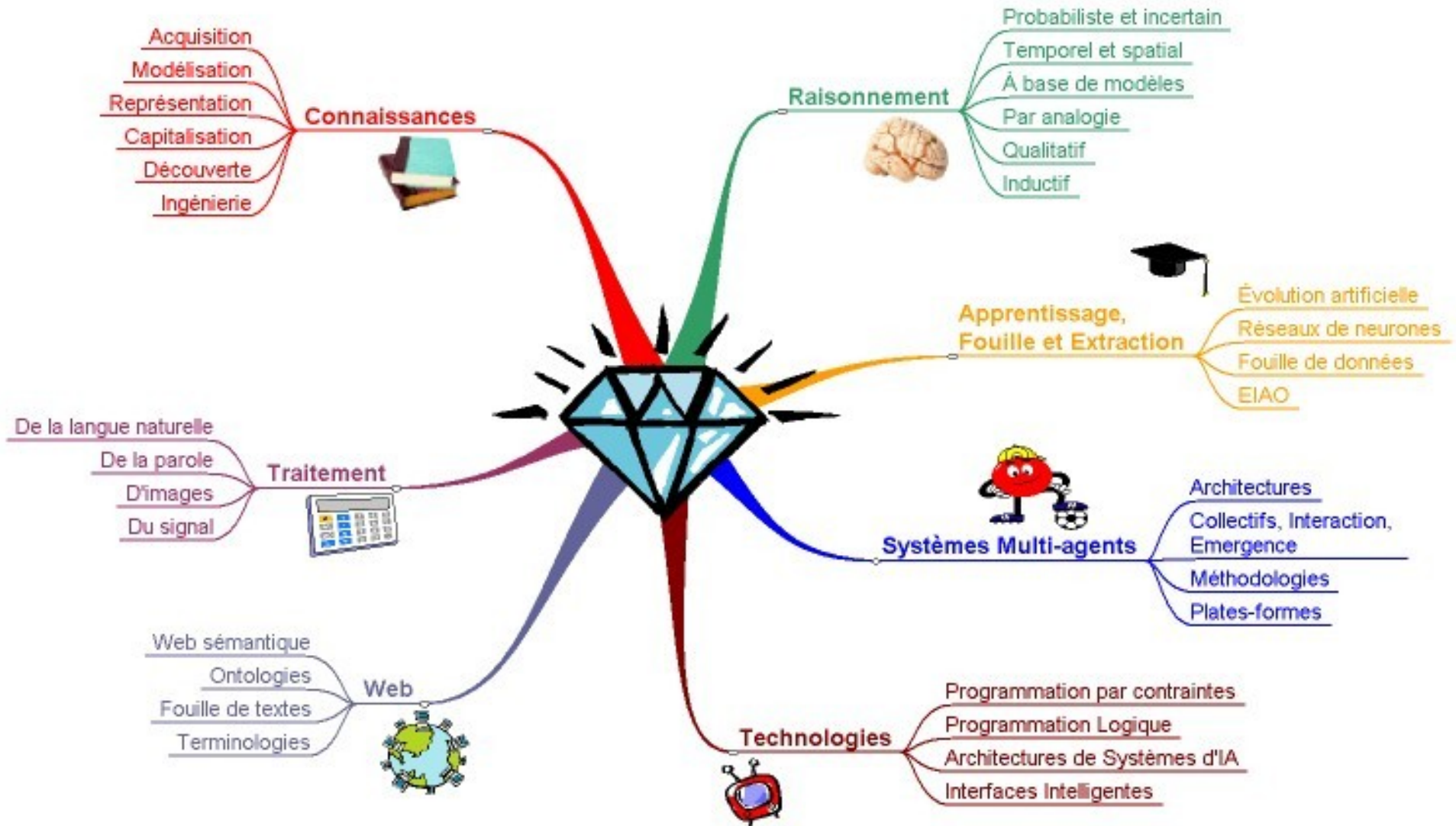
- On ne parle plus beaucoup de connaissances (~1990 - ...)
 - Nouvelles méthodes subsymboliques (réseaux de neurones, algorithmes génétiques, réseaux bayésiens, chaînes de Markov, ...)
 - Irruption de théoriciens
 - Besoins industriels

L'IA aujourd'hui, le constat

- **Ambitions initiales abandonnées**
 - on ne pense plus faire une IA à court terme
 - peu cherche à faire des machines intelligentes !
- **Les retombées de l'IA sont partout**
 - objets, agents, méthodologies, représentation des connaissances
 - approches causales, qualitatives
 - fouille de données, fouille de texte
 - statistiques non linéaires (réseaux neuronaux)
 - programmation par contraintes
 - nouvelles méthodes d'optimisation (évolution artificielle)
- **Vous les utilisez au quotidien sans le savoir**

Les domaines de l'IA

Domaines de l'IA



Organisation des domaines de l'IA par l'association Française d'Intelligence Artificielle

Connaissance

- Modélisation des connaissances
 - Systèmes experts
 - Systèmes à base de connaissances
- Représentation des connaissances
 - Formalisation de la représentation des connaissances
 - Représentation d'une information floue, imprécise
 - Représentation du temps et de l'espace
- Acquisition/maintient des connaissances
- Web sémantique
 - Formalisation des connaissances sur internet
 - Favoriser les recherches d'information

Raisonnement

- Diagnostic
- Aide à la décision
 - Théorie des jeux
- Ordonnancement et planification d'action
- Induction/déduction/construction de preuve formelle
- Modèles de raisonnement
 - Qualitatif
 - Par analogies (raisonnement à partir de cas)
 - Flou
 - ...
- Résolution automatique de problèmes
 - Heuristiques
 - Programmation par contraintes
 - Programmation logique (Prolog, ASP)

Apprentissage

- Apprentissage formel
 - Raisonnement par abduction
- Apprentissage statistique
 - Classification
 - Catégorisation
 - Modélisation statistique
- Réseaux de neurones
- Fouille de données / extraction de connaissances

Systemes distribués

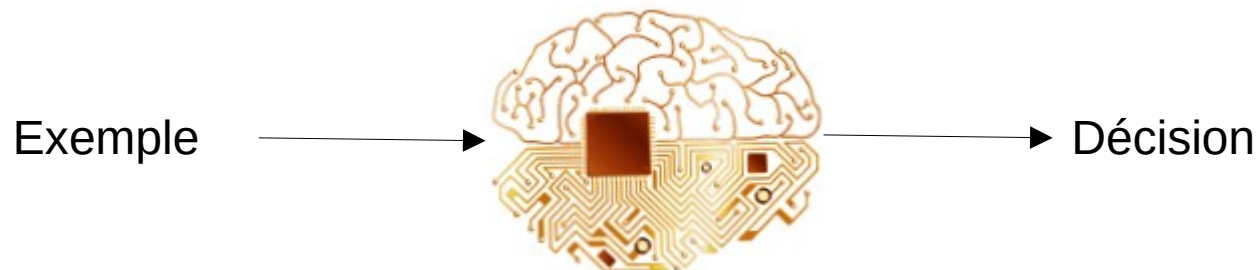
- Étude des systèmes complexes
 - Modélisation systèmes multi-agents
 - Émergence
- Résolution distribué des problèmes
 - Swarm intelligence / intelligence en essaim
- Dialogue/langage
 - Modélisation du langage
- Coopération entre machines
- Interaction H/M
- Robotique

Machine Learning

Apprentissage automatique / machine learning

- Contexte général

- L'ordinateur est un agent qui a qui on va apprendre à réaliser une tâche
- La tâche peut se réduire à prendre une décision à partir un exemple
- L'apprentissage se fait à partir d'un ensemble d'exemples disponibles qui sont présentés à l'agent
- L'agent qui a « appris » peut reproduire la tâche lorsque de nouveaux exemples lui sont présentés



Apprentissage automatique / machine learning

- L'agent est une « boîte noire » qui calcule une décision à partir d'un exemple
 - Cette boîte noire est un programme informatique
 - Ce programme n'est pas écrit directement par un programmeur mais généré par la procédure d'apprentissage
 - L'apprentissage automatique peut être vu comme un substitut à la programmation d'un agent
- On dénote cet agent comme un ***classifieur***
- Alors quand l'utiliser ?
 - La tâche est complexe et difficile à formaliser en terme de programme
 - On dispose d'exemples

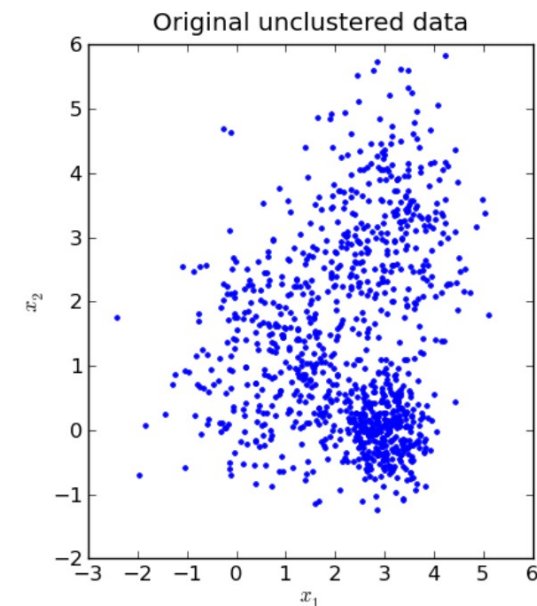
Quelques exemples d'apprentissage

- Exemples de tâche
 - son filtre anti-spam
 - la reconnaissance de caractères
 - la gestion des crédits bancaires
 - la détection de fraudes téléphoniques & bancaires
 - la reconnaissance des chiffres sur les chèques, les lettres
 - les recommandations sur Amazon, Spotify, etc
 - conduite autonome
 - la classification automatique d'images satellite

Contexte mathématique/informatique simplificateur

- La notion d'observation est souvent réduite à la notion d'un **exemple statistique**
 - Un exemple est décrit par un ensemble (fini) d'**attributs**
 - Schématiquement : une ligne de tableau pandas
 - Mathématiquement, un exemple comme un point dans un espace à grande dimension
 - *!Simplification!* → attributs numériques uniquement
- 1 exemple = 1 vecteur
- L'ensemble des exemples constitue **les données**

Title	rank	genre	description	director	actors	year	runtime	rating	votes	revenue_millions
Guardians of the Galaxy	1	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13
Prometheus	2	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.46
Split	3	Horror,Thriller	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	7.3	157606	138.12
Sing	4	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	270.32
Suicide Squad	5	Action,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.02
...
Resident Evil: Afterlife	994	Action,Adventure,Horror	While still out to destroy the evil Umbrella C...	Paul W.S. Anderson	Milla Jovovich, Ali Larter, Wentworth Miller,K...	2010	97	5.9	140900	60.13
Project X	995	Comedy	3 high school seniors throw a birthday party L...	Nima Nourizadeh	Thomas Mann, Oliver Cooper, Jonathan Daniel Br...	2012	88	6.7	164088	54.72
Hostel: Part II	997	Horror	Three American college students studying abroa...	Eli Roth	Lauren German, Heather Matarazzo, Bijou Philli...	2007	94	5.5	73152	17.54

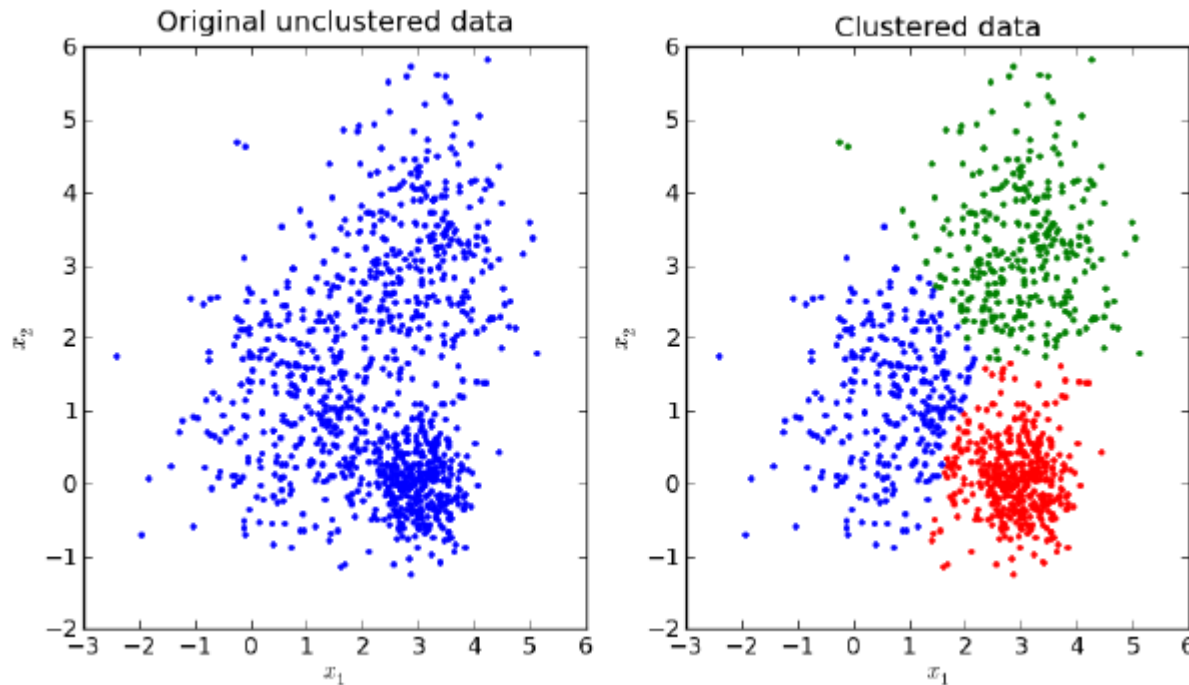


Trois grands types d'apprentissage

- Descriptif
- Prédicatif
- Prescriptif

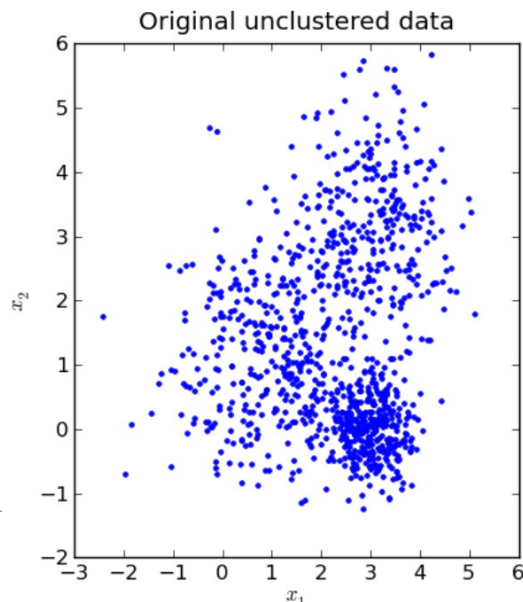
Apprentissage descriptif (non-supervisé)

- L'objectif est d'identifier des régularités dans les données qui puissent être utilisées pour l'analyser



Apprentissage prédictif (apprentissage supervisé)

- L'objectif est d'apprendre à reproduire des décisions associées à un exemple
 - Le jeu d'apprentissage est composé d'exemples étiquetés
 - Étiquette numérique : tâche de régression
 - Étiquette symbolique : tâche de classification
 - Vocabulaire
 - L'étiquette est l'attribut à expliquer
 - Les autres attributs sont des attributs explicatifs



Quelle sera la valeur de x_2 si on donne une nouvelle valeur de x_1 ?

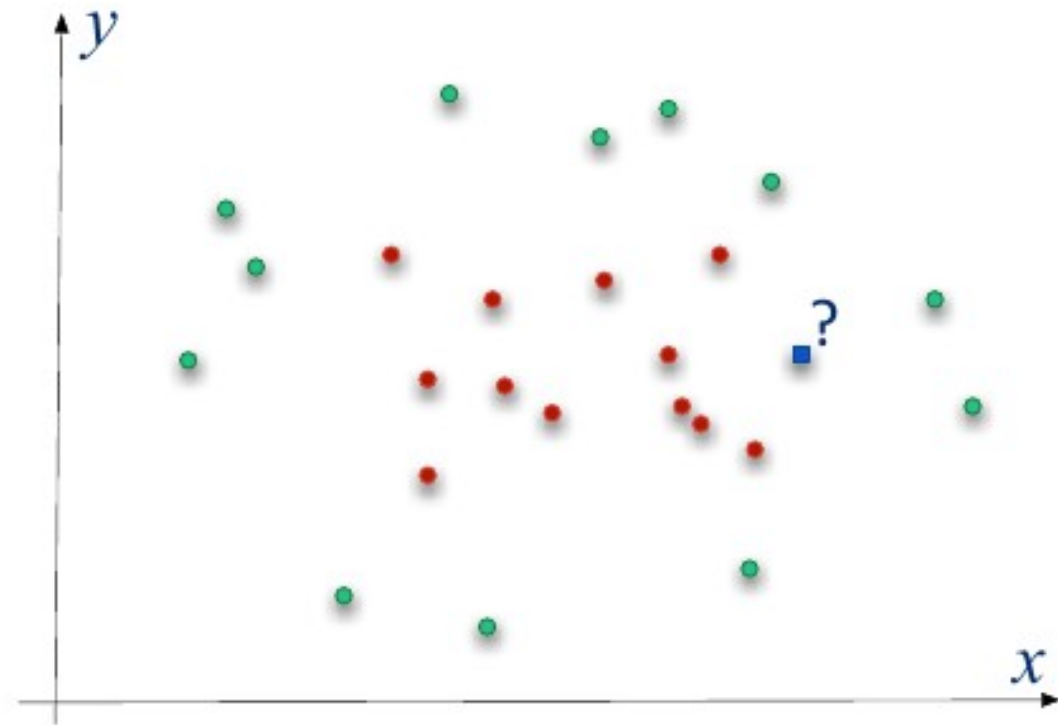
Apprentissage prescriptif

- nouvelles formes de tâche d'apprentissage qui vise à donner des informatiques utilisables par des humains
- Questions nouvelles
 - Tâche d'apprentissage d'un ordre (recommandations)
 - Recherche d'explication à la décision
 - Chercher des causalités dans les données
- Exemple des limites des méthodes
 - J'observe que les gens en maillot mange souvent des glaces
 - Je voudrais vendre d'avantage de glaces
 - => je demande au gens de se mettre en maillot

Apprentissage supervisé

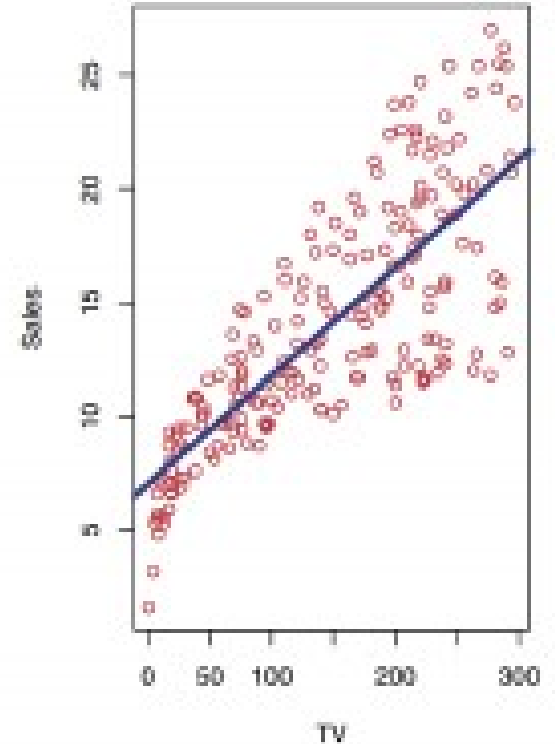
Apprentissage supervisé (classification)

- On dispose d'exemples d'apprentissage
 - 1 exemple =
 - 1 point dans l'espace
 - Une couleur (la décision ou la classe)
- On veut
 - Déterminer la couleur d'un nouvel exemple



Apprentissage supervisé (régression)

- On dispose d'exemples d'apprentissage
 - 1 exemple =
 - 1 point dans l'espace
 - Une valeur quantitative
- On veut
 - Déterminer la valeur associée à un nouvel exemple



Données en apprentissage

Attributs (features)

Label

Instance

Identifier	Gender	Age	Education level	Married	Nb of children	Salary	Profession	To prospect?
I_21	M	43	Master	Y	3	55,000	Architect	YES
I_34	M	25	Sophomore	N	0	21,000	Nurse	NO
I_38	F	34	PhD	Y	2	35,000	Univ. Prof.	YES
I_39	F	67	Bachelor	Y	5	20,000	Retired	NO
I_58	F	56	Technical studies	Y	4	27,000	Employee	NO
I_73	M	40	Graduate	N	2	31,000	Salesman	YES
I_81	F	51	Master	Y	3	75,000	CEO	YES

Exemple pour la classification d'images satellite :

- instance = 1 pixels
- features = valeurs des bandes spectrales (ou indices)
- label = occupation du sol / hauteur d'herbe

Formalisation du problème d'apprentissage automatique

- On dispose d'un ensemble d'apprentissage

$$S = \{(X_i, y_i)\}_{i=1..n}$$

- X_i est un exemple étiqueté avec la classe y_i
 - X_i est un vecteur
 - y_i est un scalaire (un nombre/un label)
- n est la taille du jeu de données
- On suppose qu'il existe une régularité
 - $y = f(X)$ (ou $P(X,y)$ en probabilité)
- L'apprentissage consiste donc à découvrir f sachant qu'on a des observations

$$y_i = f(X_i) + \epsilon_i$$

Formalisation du problème d'apprentissage automatique

- On cherche donc à trouver **f** ... mais l'espace de toutes les fonctions possible est trop grand
 - impossible de trouver la solution informatiquement
 - on ne connaît qu'un nombre fini de valeurs : ne permet pas de connaître **f** totalement
- On cherche plutôt une fonction **h**
 - **h** est une fonction d'une forme particulière (espace d'hypothèses)
 - On cherche une approximation de **f**
 - **h** ~ **f** sur les X_i ... on cherche à minimiser une erreur
$$\frac{1}{n} \sum_{i=1}^n (y_i - h(X_i))^2$$
 - mais aussi sur les futurs points : besoin de **généralisation** ...

Exemple de la régression comme problème d'apprentissage automatique

- Exemple de régression

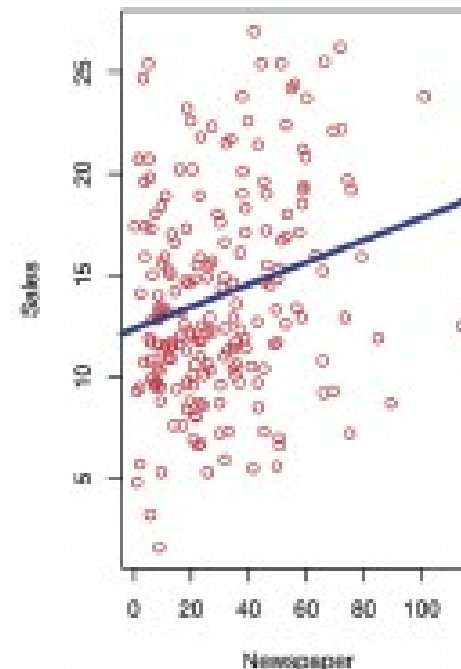
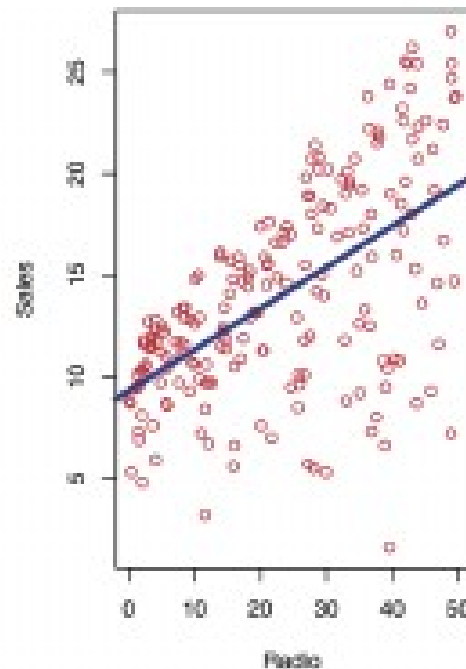
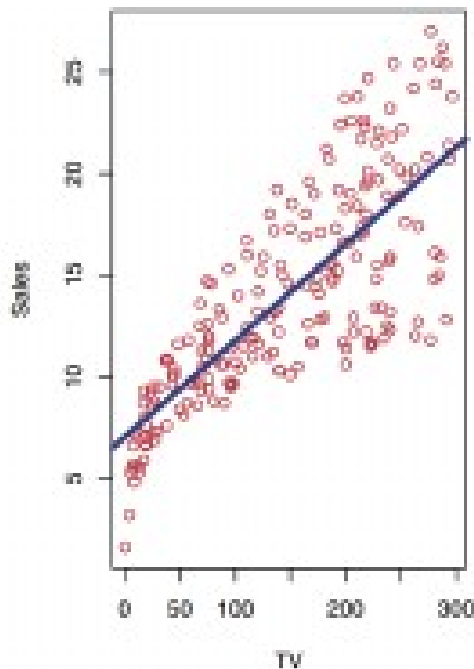
- Features

- Budget pub TV (x_1)
 - Budget pub radio (x_2)
 - Budget pub journal(x_3)

$$X = (x_1, x_2, x_3)$$

- Sortie : volume de ventes

- On cherche à prévoir le volume des ventes en fonction des budgets dépensés



Exemple de la régression comme problème d'apprentissage automatique

- Exemple d'espace d'hypothèse

- On cherche une fonction de la forme

$$Y = a.x_1 + b.x_2 + c.x_3 + d$$

- (a,b,c,d) sont les paramètres du modèle

- Trouver une fonction = trouver les valeurs pour les paramètres

- De manière un peu plus formelle, on note généralement

- Les paramètres du modèle $\theta = \{a, b, c, d\}$

- Une fonction estimée (en fait, l'estimation de ses paramètres) h_θ

Exemple de la régression comme problème d'apprentissage automatique

- Le problème de trouver une fonction est transformé en un problème de minimisation de fonction
 - Regression = minimisation de l'erreur
 - Cas d'une régression linéaire

$$\min_{\beta, \alpha} \sum_{i=1}^n (y_i - (\beta x_i + \alpha))^2$$

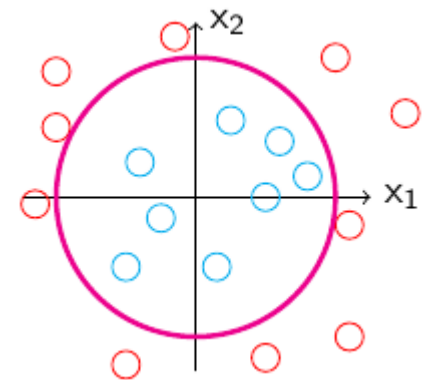
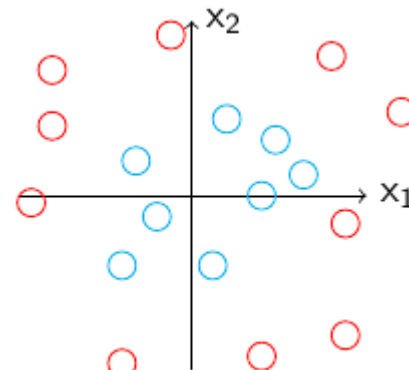
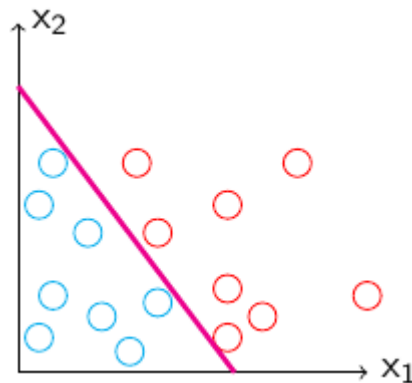
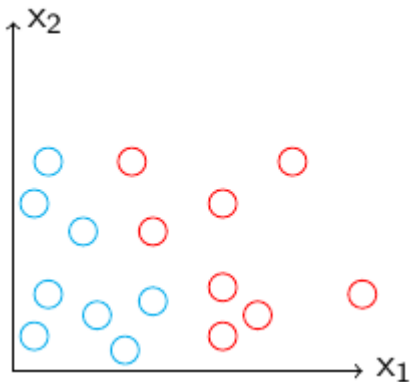
- Notation matricielle

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

La classification est un problème similaire

- Le problème de classification est un problème de discrimination
 - Il s'agit d'identifier une frontière entre les exemples qui discrimine le mieux les classes
 - C'est de nouveau un problème de construction d'une fonction
 - Il est transformé en un problème d'optimisation d'une mesure de qualité



Erreur de classification : la matrice de confusion

- La matrice de confusion est une généralisation pour plusieurs classes
 - Exemple d'une belle matrice

		Truth					Total
		Asphalt	Concrete	Grass	Tree	Building	
Predicted	Asphalt	2385	4	0	1	4	2394
	Concrete	0	332	0	0	1	333
	Grass	0	1	908	8	0	917
	Tree	0	0	0	1084	9	1093
	Building	12	0	0	6	2053	2071
	Total	2397	337	908	1099	2067	6808

Mesure de l'erreur

- Les approches usuelles cherchent à minimiser l'erreur de prédiction d'un classifieur
- Formulation générale du problème similaire à celle de la régression
 - Trouver les paramètres d'un modèle par la minimisation d'une erreur (ou d'une *loss*)

$$\min_{\theta} \sum_{i=1}^n \mathcal{L}(\mathbf{Y}_i, h_{\theta}(\mathbf{X}_i))$$

- NB : cette minimisation n'est pas magique ... et peut même être très complexe
 - Il existe des techniques très avancées pour le résoudre
 - Ces techniques sont des solutions approchées (voir non exactes)

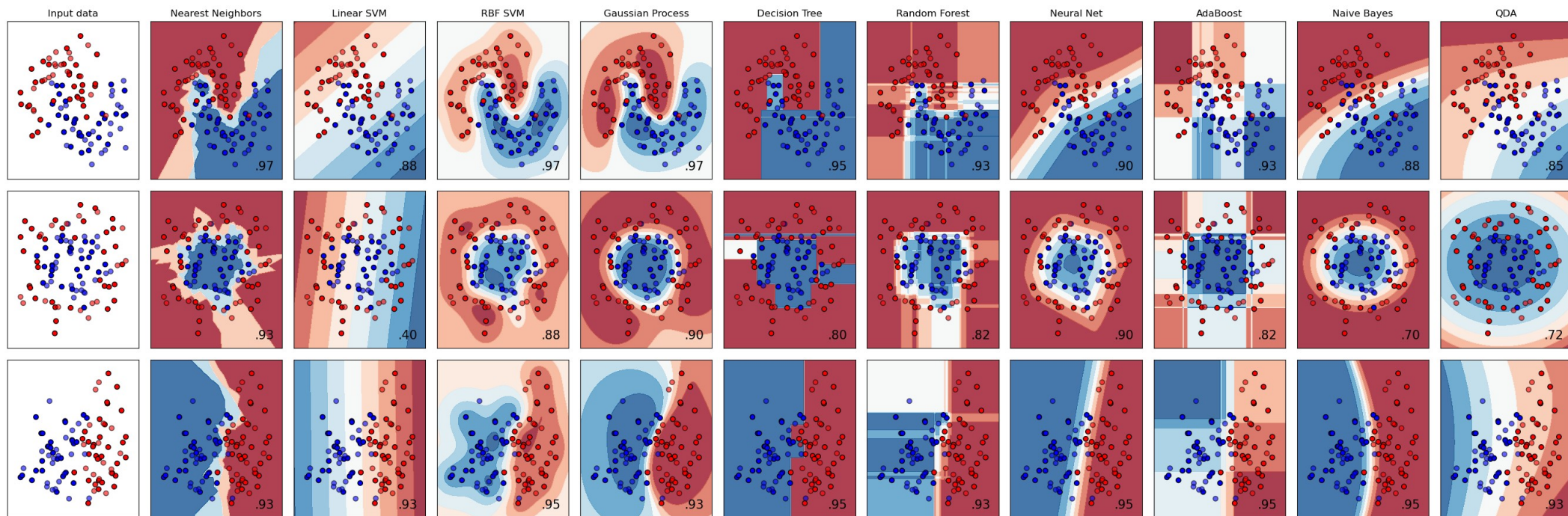
Le problème du *data-scientist*

- Faire un apprentissage automatique c'est choisir parmi différents espaces d'hypothèse lequel sera généralisera le mieux
- En pratique
 - 1)Un espace d'hypothèse = une méthode d'apprentissage particulière (et il y en a beaucoup)
 - 2)Généralisera le mieux = calcul d'une erreur
 - 3)Une méthodologie : séparation d'un jeu d'apprentissage et d'un jeu de test

Espaces d'hypothèse ? → type de classifieur

- Plein de modèles existent (le type de classifieur):
 - Arbres de décision
 - SVM
 - Modèle bayésiens
 - Réseaux de neurone
- Ce qui les différencie
 - La forme des frontières qu'ils peuvent engendrer et donc leur performances
 - Les temps de calcul
 - Leur interprétabilité
- Dans ce cours : on utilise la librairie *sklearn* qui inclus un très grand nombre de modèles

Différentes « formes » des limites des types de classifieurs



Source : sklearn

À propos des modèles : paramètre et hyper-paramètres

- Paramètres
 - Les paramètres du modèle sont les éléments dont la valeur va être déterminée par le processus d'apprentissage
 - Ex : en régression linéaire, les poids sont les paramètres du modèle
- Hyper-paramètres
 - Certains modèles peuvent être configurés en amont de l'apprentissage
 - Ces configurations sont associées à des hyper-paramètres
 - Ex : en régression polynomiale, le degré du polynôme est un hyper-paramètre
- Attention : en fonction de la méthode d'apprentissage mise en œuvre, un même paramètre peut être paramètre ou hyper paramètre

-
- Ok ...
 - mais est-ce le bon espace d'hypothèse ??
 - Comment fait-on pour choisir le « meilleur » modèle
 - Est-ce que le meilleur modèle est celui qui fait le moins d'erreur ?
 - Et de quelle erreur parle-t-on ?
 - Erreur sur le jeux d'apprentissage
 - Erreur sur les nouveaux exemples

Exemple de la régression comme problème d'apprentissage automatique

- Quelques constats :
 - Trop de paramètres (flexibilité) conduit à un problème de sur-apprentissage
 - Ex. île avec 1 mouton noir → tous les moutons sont noirs ?
 - On parle de **variance**
 - on cherche un modèle qui généralise !
 - Trop peu de flexibilité (modèle trop simple) conduit à ne pas capturer bien le comportement de la régularité
 - On parle de **biais**
 - on cherche un modèle qui fait peu d'erreur
- Le problème d'un bon apprentissage est de faire un bon compromis entre biais et variance

Méthodologie : séparation jeu d'apprentissage et jeu de test

- Objectif
 - Évaluer un modèle appris en tenant compte du biais **et de la variance**
 - Pour cela, on regarde l'erreur sur un jeu de test (et pas sur le jeu d'apprentissage)
 - On veut montrer une performance indépendante du jeu d'apprentissage
 - Pour cela, on répète plusieurs fois avec différents jeux d'apprentissage pour avoir une erreur moyenne (et une variance de l'erreur)
- Trois approches classiques
 - 1) K-fold
 - 2) Leave-one-out

Méthodologie : séparation jeu d'apprentissage et jeu de test

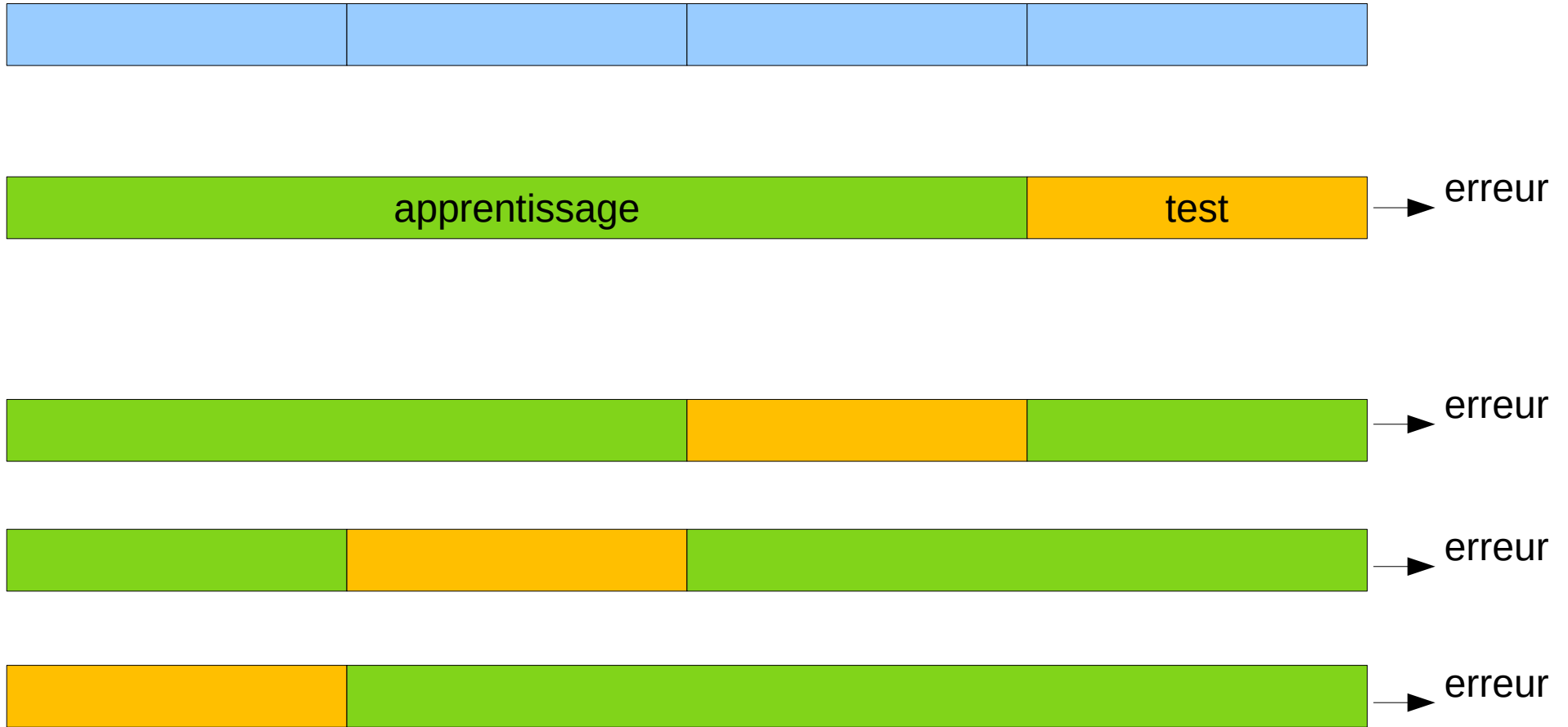
- Limiter la variance d'un modèle en évaluant sur erreur sur un jeu de test
- Exemple : on dispose d'un jeu avec 1000 exemples
 - 750 exemples pour construire le modèle (jeu d'apprentissage)
 - 250 exemples pour évaluer l'erreur (jeu de test)
- Les exemples du jeu de test ne doivent jamais servir dans la phase d'apprentissage !
- On « simule » le fait d'avoir de nouveaux exemples inconnus pour l'agent qui apprend
 - Exemple d'un quizz sur les étudiants

k-fold

- Principe
 - Un jeu d'exemple est répartie en k sous – ensemble du jeu d'apprentissage $\{D_i\}_{i=1..k}$
 - Pour chaque sous ensemble $i = 1..k$
 - On fait l'apprentissage du modèle sur $\{D_i\}_{i \neq k}$
 - On évalue l'erreur sur D_k
- Permet de répéter l'évaluation du modèle
 - La **valeur moyenne de l'erreur** donne une évaluation indépendante du jeu d'exemple
 - La **variance de l'erreur** indique une sensibilité éventuelle au choix des exemples d'apprentissage
- Version stratifiée : on cherche à respecter les proportions de chaque classe dans les sous ensembles

k-fold

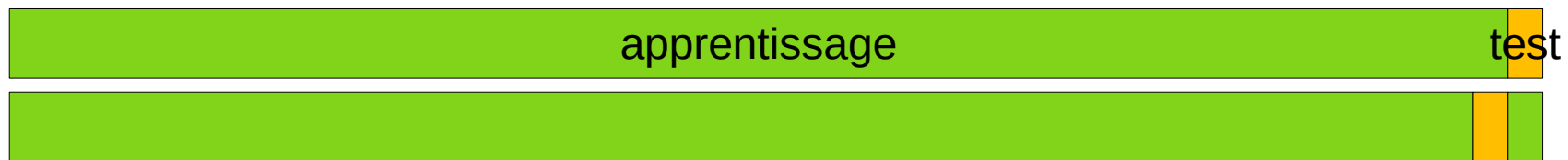
Exemple avec k=4



- erreur moyenne
- variance de l'erreur

Leave-one-out

- Principe du k-fold avec $k=n-1$
 - Le jeu de test ne comprend qu'un exemple
- Méthode intéressante pour les petits jeux de données
 - Conserve un maximum de données d'apprentissage
- Utiliser plus particulièrement en régression



(...)

Apprentissage supervisé : résumé

- Apprentissage supervisé = construire un classifieur à partir d'exemples étiquetés
 - Les **exemples** sont décrits par un ensemble de *features* et associés à un *label*
 - Les features/labels peuvent être continus ou catégoriels
 - On dispose d'un ensemble d'exemples (jeu d'apprentissage) => un tableau pandas !
 - Un **classifieur** correspond à
 - un type de frontière (*capacité de discrimination*)
 - des paramètres (plus ou moins) et des hyper-paramètres
 - une méthode pour estimer ses paramètres
 - L'apprentissage consiste à estimer les paramètres à partir d'exemples en minimisant une erreur
 - Séparation des exemples en jeu d'apprentissage et jeu de test
 - Évaluation des performances de classification avec des mesures (précision, rappel, matrice de confusion)