

Formation Machine Learning – les fondamentaux



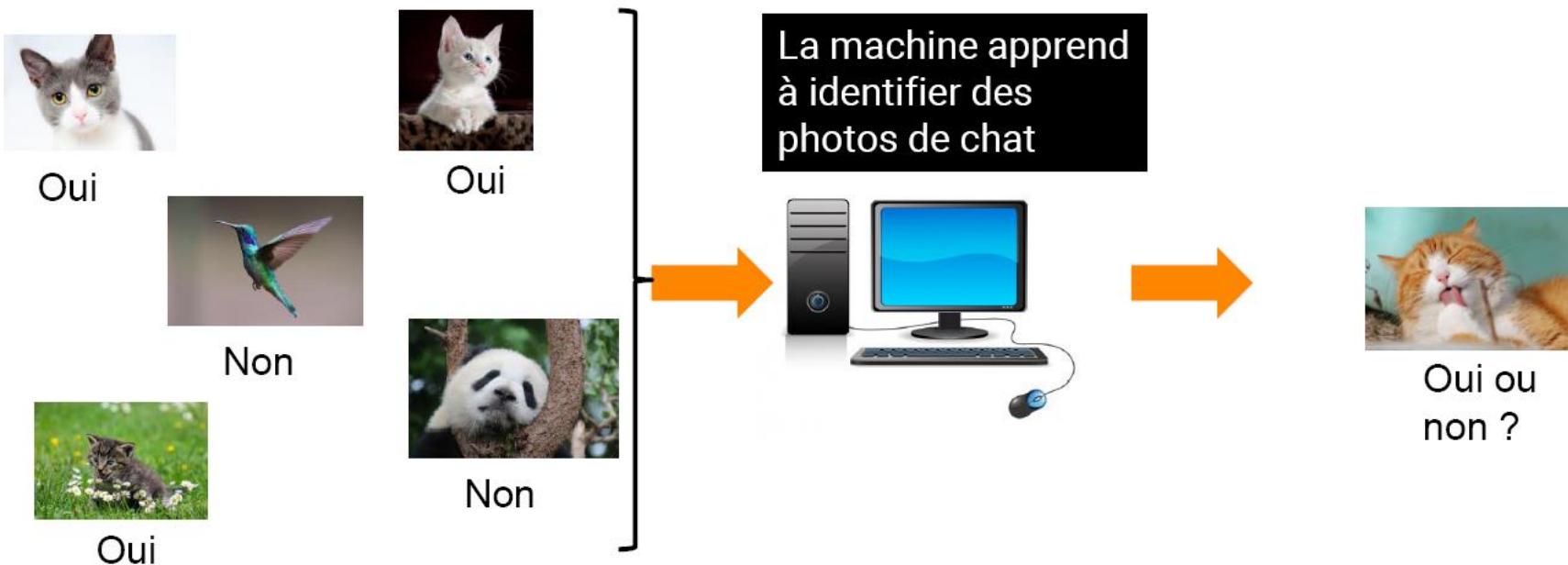
Programme de la formation

- **Bases du Machine Learning**
 - Contexte et outils
 - Les problématiques de machine learning
 - Les projets de machine learning
 - Les algorithmes
- **Les modèles et les étapes**
 - Feature engineering
 - Régression : détection de tendances et prévisions
 - Classification par catégories
 - Clustering : regroupement automatique par familles
- **Pratique**
 - Un modèle de machine learning supervisé en python
 - Un modèle de machine learning non supervisé en python
 - Un modèle de traitement de données textuelles en python
- **Le deep learning**
 - Les réseaux de neurones
 - Les principes du deep learning
 - Les différentes couches en deep learning
- **Pratique**
 - Un modèle de deep learning en python avec TensorFlow

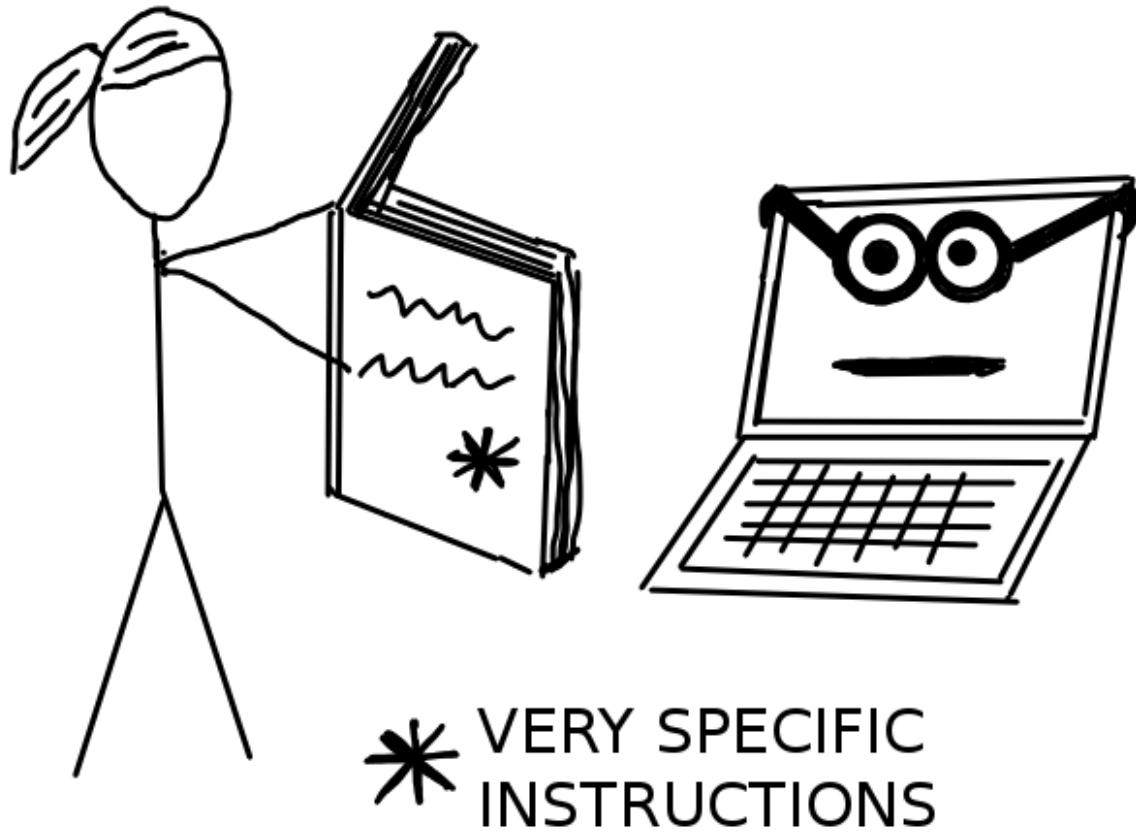
Tour de tables des acquis et des besoins

Le machine learning

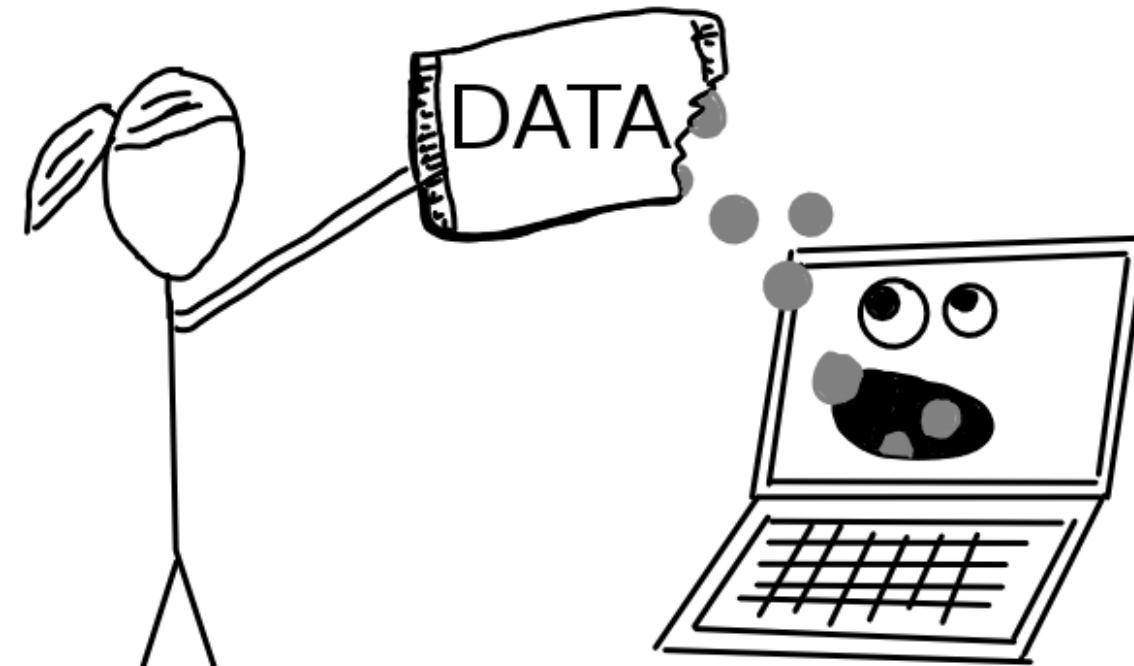
- **Machine Learning** : Techniques de calcul permettant aux ordinateurs d'apprendre à exécuter des tâches en *s'inspirant* de patterns perçus sur des données, avec peu d'intervention humaine. Les tâches se résument souvent à de la prédiction.



Without Machine Learning

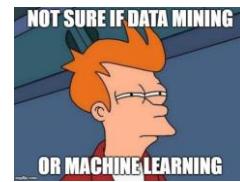


With Machine Learning



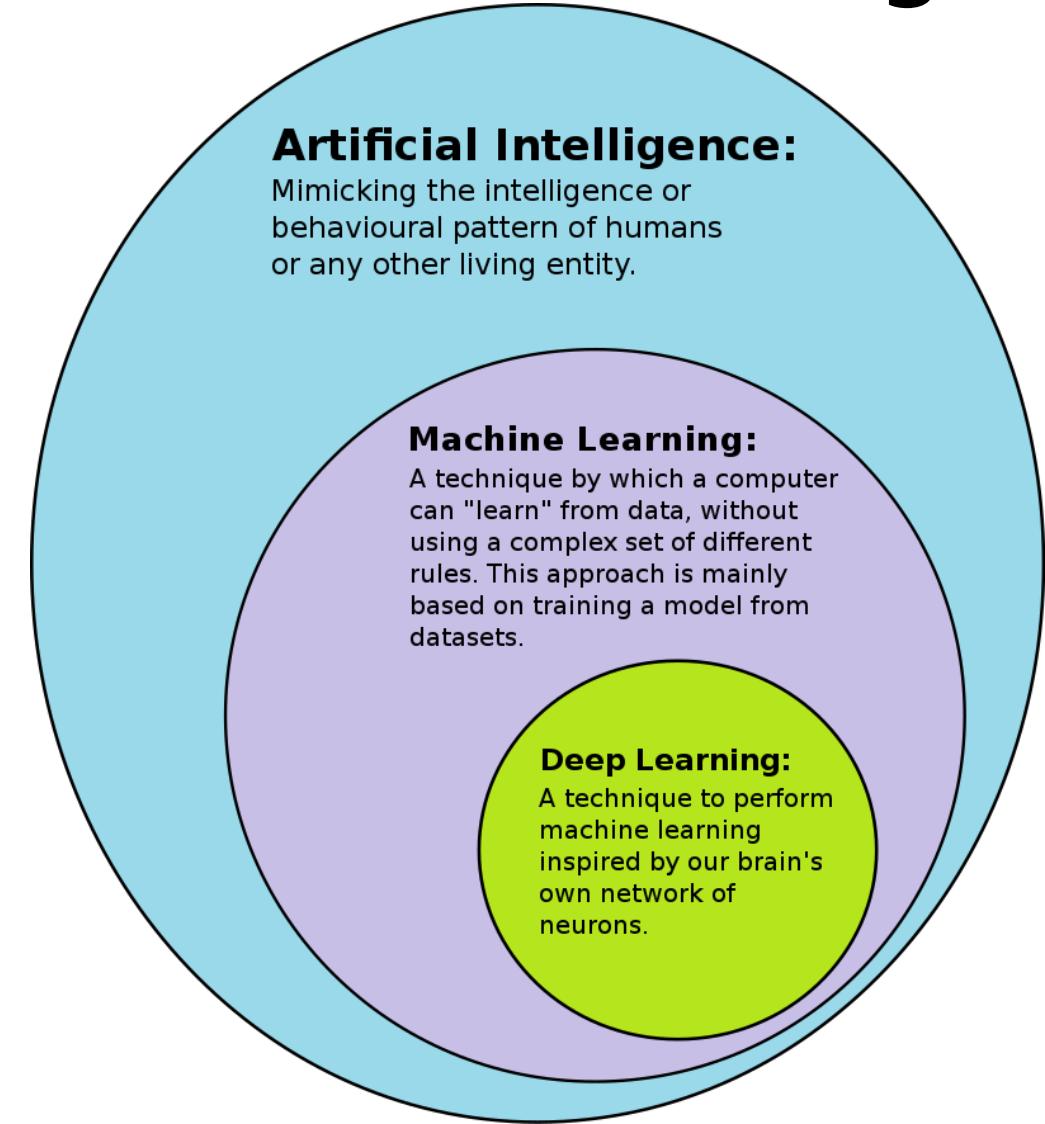
Quelques exemples

- Médecine : la machine *apprend* à identifier l'état sain/malade de patients à partir de données d'imagerie classées en sain/malade par des experts.
- E-commerce: la machine *apprend* à recommander des articles aux consommateurs en fonction de données de localisation géographique et d'achats passés.
- Assurance : la machine *apprend* à détecter le risque de fraude à partir de données de comportement et de fraudes passées.
- Emailing: la machine *apprend* à classer des emails en spam/non-spam à partir d'une base de mails déjà classés en spam/non-spam.

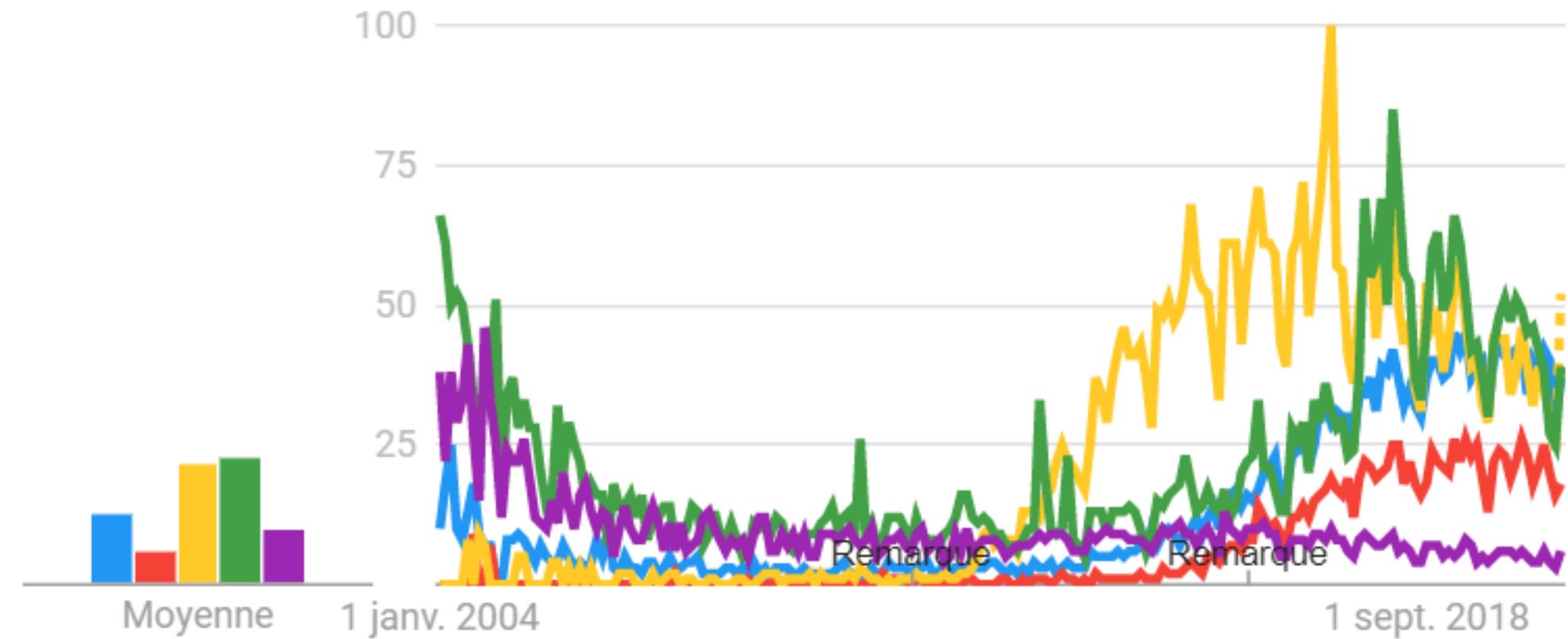


Le vocabulaire autour du machine learning

- Il existe de nombreux termes liés au machine learning :
 - Apprentissage
 - Data mining
 - Big data
 - Intelligence artificielle
 - Deep learning
 - ...

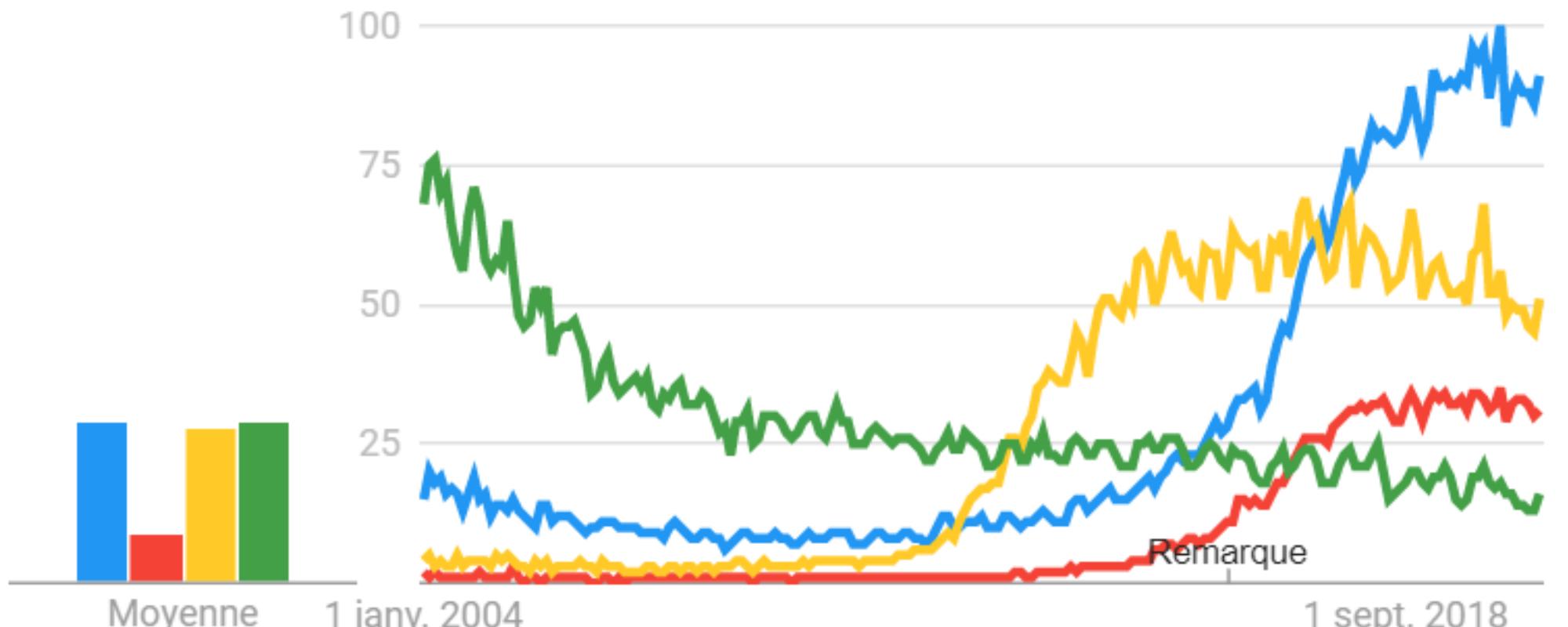


- machine learning
- deep learning
- big data
- intelligence artificielle
- data mining



France. 01/01/2004 – 16/09/2020. Recherche sur le Web.

● machine learning ● deep learning ● big data ● data mining



Dans tous les pays. 01/01/2004 – 16/09/2020. Recherche sur le Web.

Le machine learning par la pratique

<https://teachablemachine.withgoogle.com/>

Essayez de créer un modèle de ML et stockez ce modèle

On va construire un modèle de ML / DL et récupérer ce modèle.

Teachable Machine

Train a computer to recognize your own images, sounds, & poses.

A fast, easy way to create machine learning models for your sites, apps, and more – no expertise or coding required.

Get Started

Les outils

Sondage mené auprès de 2k data scientists : quels logiciels utilisez-vous ?

2 leaders open source : Python et R

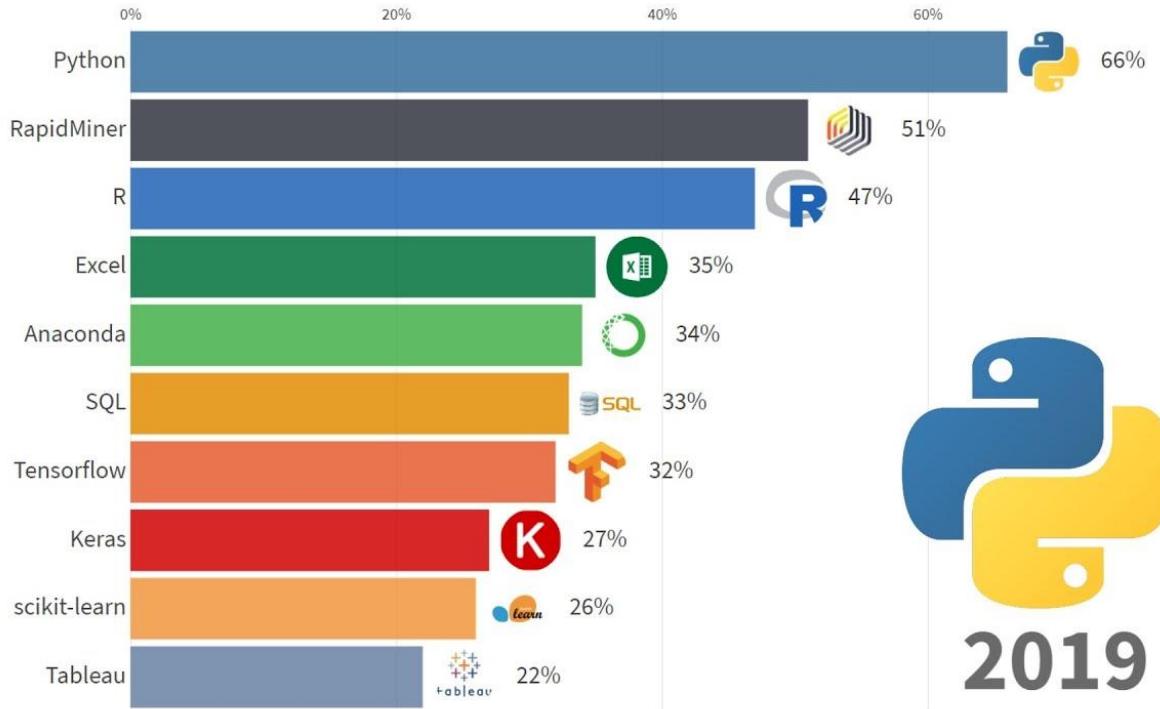
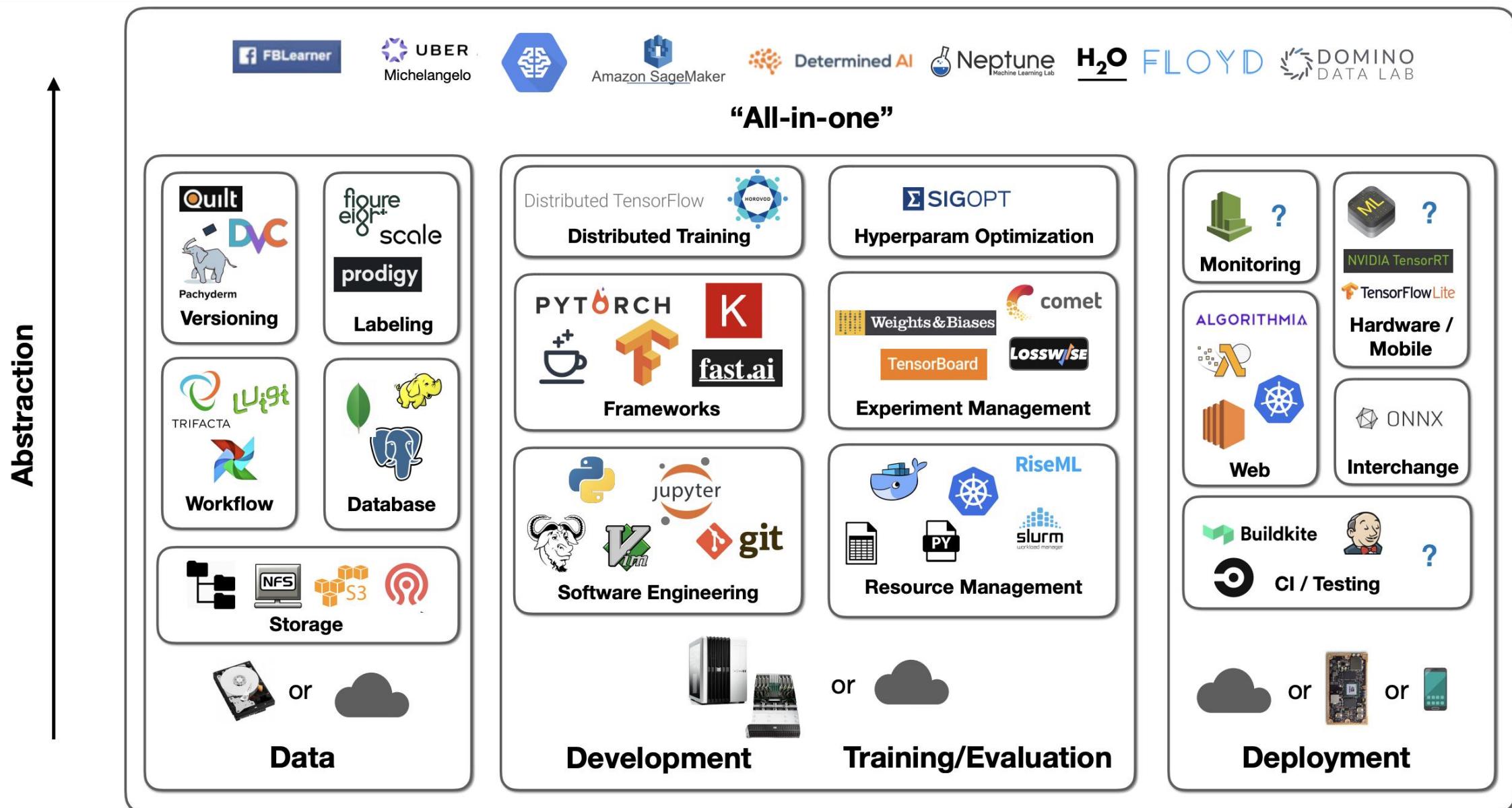


Figure 1: Magic Quadrant for Data Science and Machine Learning Platforms



Gartner 2021 Magic Quadrant for Data Science and Machine Learning Platforms



Choix entre Python et R – Pourquoi choisir ?

- Avoir des résultats détaillés immédiats : R.
- S'intéresser plutôt à l'aspect statistique / data mining : R.
- Utiliser des méthodes statistiques pointues : R.
- Dataviz : égalité.
- Gagner en Vitesse de calcul : Python.
- Faire du Deep Learning : Python (avec interface Keras et TensorFlow ou PyTorch).
- Développer une application web de dataviz rapidement : R-Shiny.
- Passer en production : Python

Les exemples de cette formation
seront basés sur Python

Outils



Le machine learning par des cas d'usage

Les cas d'usage

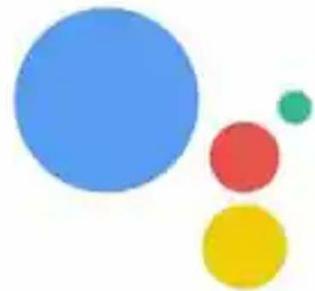
- Le nombre de cas d'usage en data science est illimité
- Chaque domaine d'application a ses propres cas d'usage :
 - Vente
 - RH
 - Comptabilité
 - Web
 - Enquêtes
 - Industrie
 - ...

L'intelligence artificielle

- Le cas Google Duplex
- Principe : prise de rendez-vous téléphonique
- Fonctionnement :
 - Vous définissez en ligne votre rendez-vous
 - Google Duplex, appelle et fixe le rendez-vous par un échange téléphonique

Google Duplex

Advancing AI for Everyone



L'intelligence artificielle

- Challenge :
 - Convertir une demande écrite en réservation
 - Être capable d'entretenir une « conversation » pour la réservation
- Techniques utilisées :
 - Deep learning
 - Utilisation de bases de données big data
 - Traitement du langage naturel

La maintenance prédictive

- Principe : être capable de prédire l'apparition d'une défaillance
- Méthodologie : apprendre à repérer les défaillances en utilisant les données historiques
- Risques : attention aux données d'apprentissage



Le cas SNCF

- Maintenance prédictive en utilisant des données sonores
- Données : enregistrement de trains avec ou sans défaillances
- Modèle : apprentissage sur les données sonores pour aboutir à une prévision de maintenance



Les moteurs de recommandation

- Essor de l'e-commerce
- Principe :
 - être capable de recommander des produits en fonction de votre comportement



Les moteurs de recommandation

- Fonctionnement :
 - La recommandation personnalisée (en fonction de vos actions)
 - La recommandation objet (en fonction des caractéristiques de l'objet)
 - La recommandation collaborative (en fonction des actions passées des clients vous ressemblant)
 - La recommandation hybride (combine les trois recommandations précédentes)

Les outils de ciblage client

- Principe : être capable d'identifier des clients « à risque » (crédit par exemple)
- Fonctionnement : utiliser les données historiques pour construire un modèle permettant de scorer un demandeur de crédit
- Outils : modèles de machine learning plus « classique »

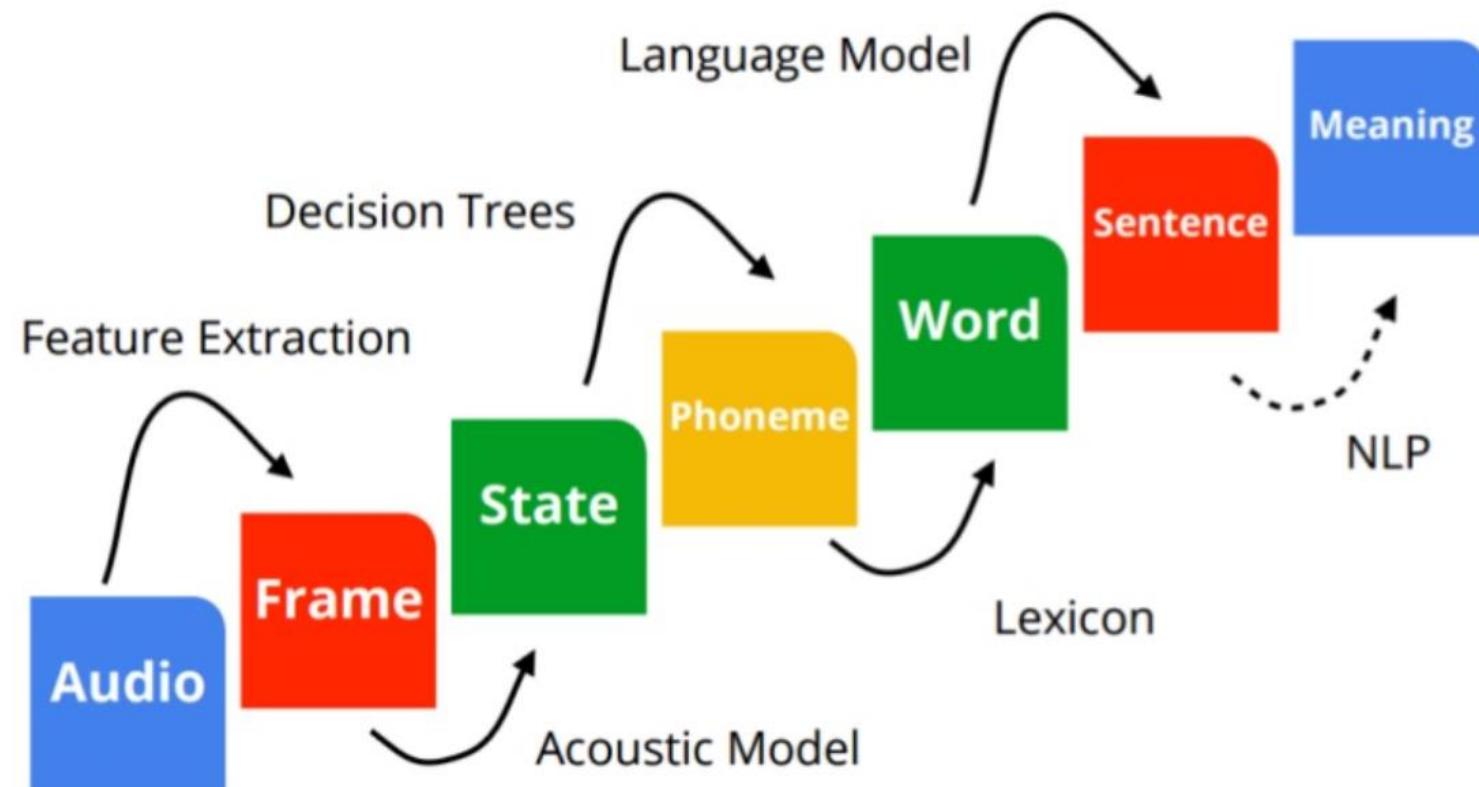
Automatisation des traitement de données RH

- Principe : associer des profils avec des missions
- Fonctionnement : utiliser les données historiques pour construire un modèle permettant de trouver le bon profil pour une mission
- Outils : modèles d'apprentissage sur données non structurés. Souvent des CV avec beaucoup de données textuelles et des descriptions « floues »

Assistant

- Principe : être capable de répondre à n'importe quelle demande
- Fonctionnement : utiliser les données des utilisateurs pour adapter la réponse à la demande
- Outils : modèles de deep learning et de traitement du langage
- https://assistant.google.com/intl/fr_fr/

Assistants – reconnaissance sonore



Retour au machine learning

Définition

- Le machine learning est associé à l'apprentissage d'une machine
- L'idée du machine learning est de créer un programme qui apprend grâce à l'expérience
- Quelques applications du machine learning :
 - Apprentissage de la fraude pour les cartes de crédit
 - Filtrage de l'information et recommandation pour les préférences des consommateurs
 - Véhicules autonomes afin de conduire sur des autoroutes

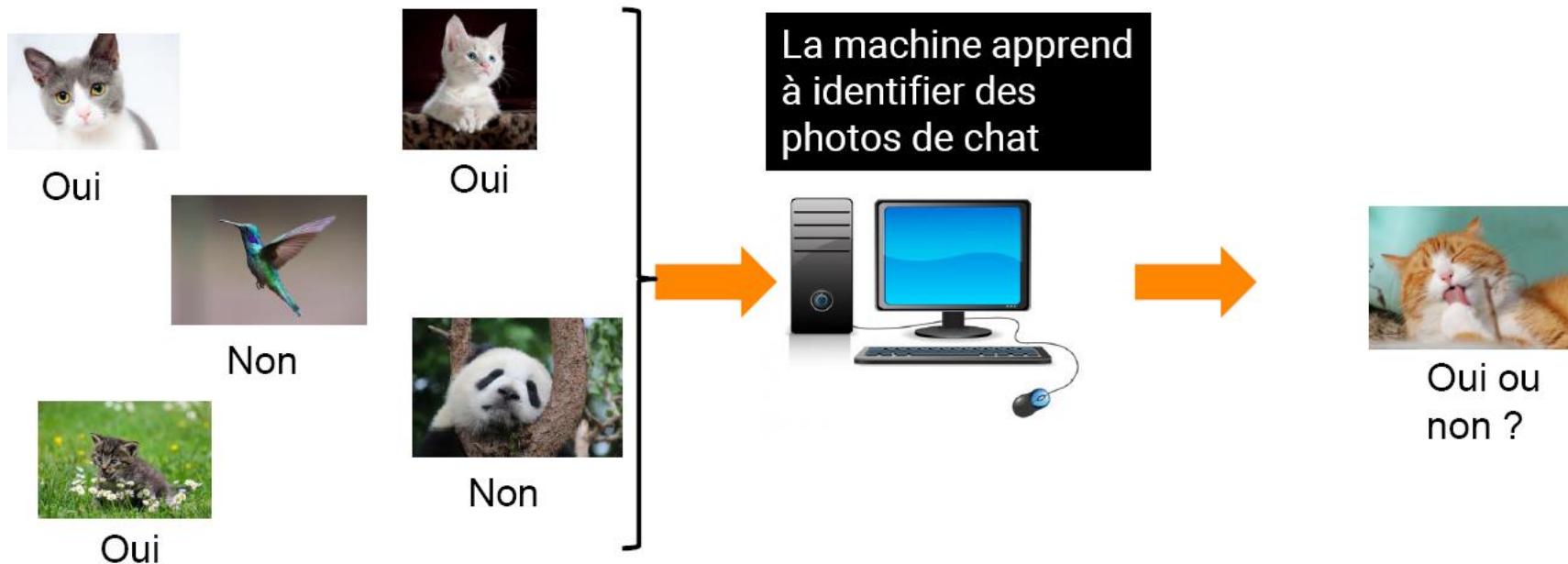
Les problèmes liés au machine learning

- Quel algorithme choisir suivant le problème traité ?
- Quels paramètres expérimentaux choisir afin d'obtenir des résultats satisfaisants ?
- Combien faut-il d'observations pour l'apprentissage ?
- Comment et quand ajouter des connaissances a priori dans le modèle ?
- Comment réduire l'apprentissage à une seule tâche et à un seul problème d'optimisation pour des cas complexes ?
- ...

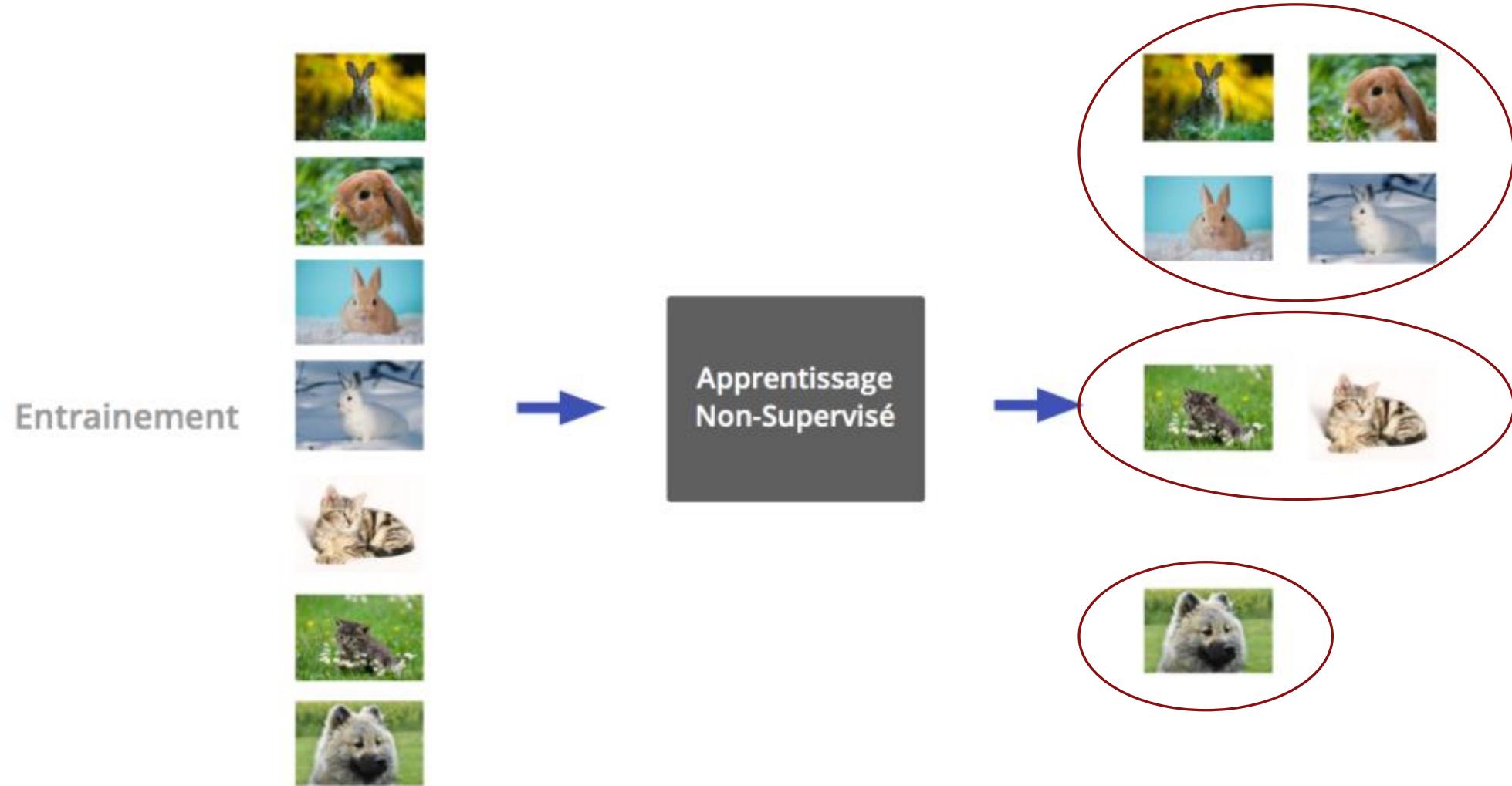
Les groupes d'algorithmes de machine learning

- Les algorithmes de machine learning peuvent être rassemblés dans 3 groupes principaux :
 - L'apprentissage supervisé
 - Cible (variable dépendante qualitative ou quantitative)
 - Prédicteurs (variables indépendantes)
 - Régression, arbre de décision, foret aléatoire, kNN, régression logistique, SVM...
 - L'apprentissage non supervisé
 - Pas de variable dépendante, pas de modélisation
 - Classer des individus dans différents groups ou trouver des associations
 - Apriori algorithm, K-means.
 - L'apprentissage par renforcement
 - Apprentissage pour prendre des décisions spécifique afin d'aboutir à un but précis
 - Markov Decision Process

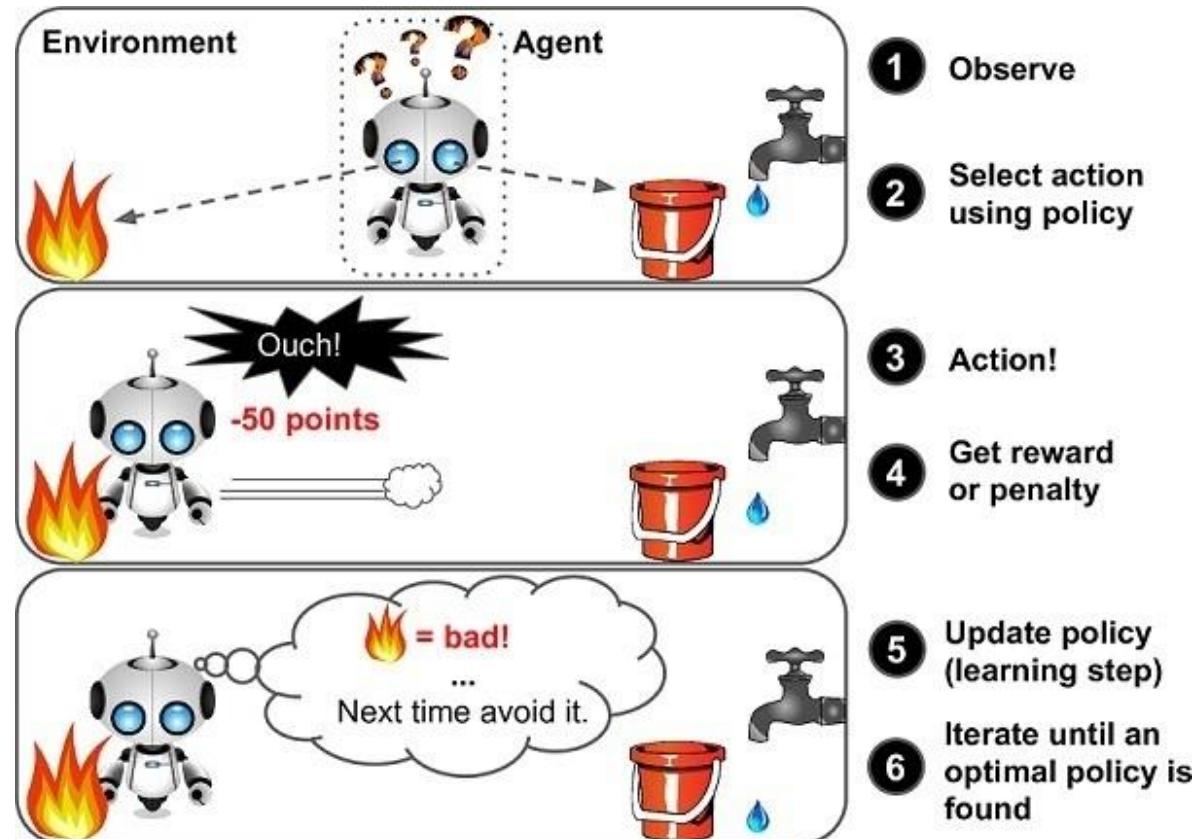
Apprentissage supervisé

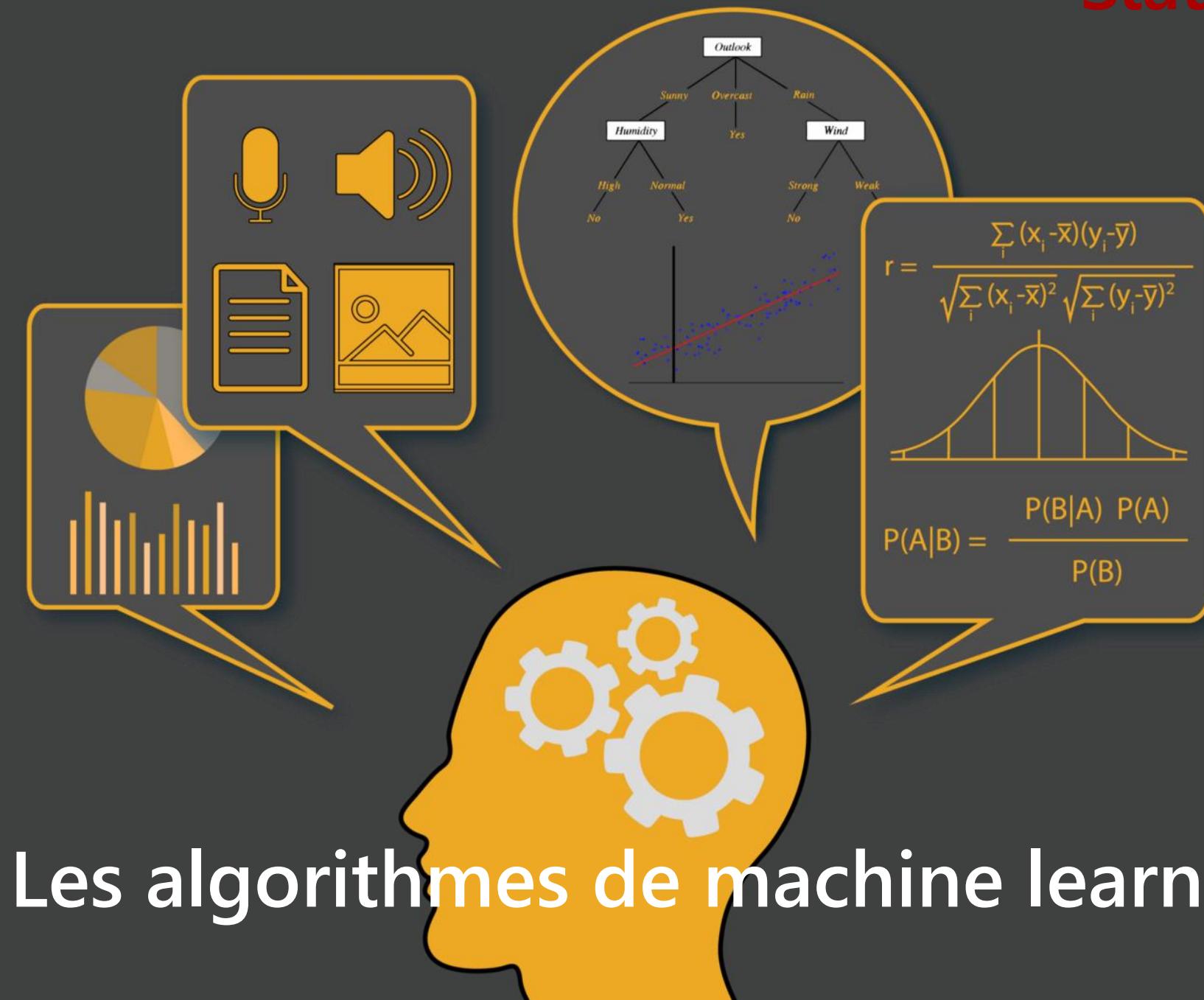


Apprentissage non supervisé

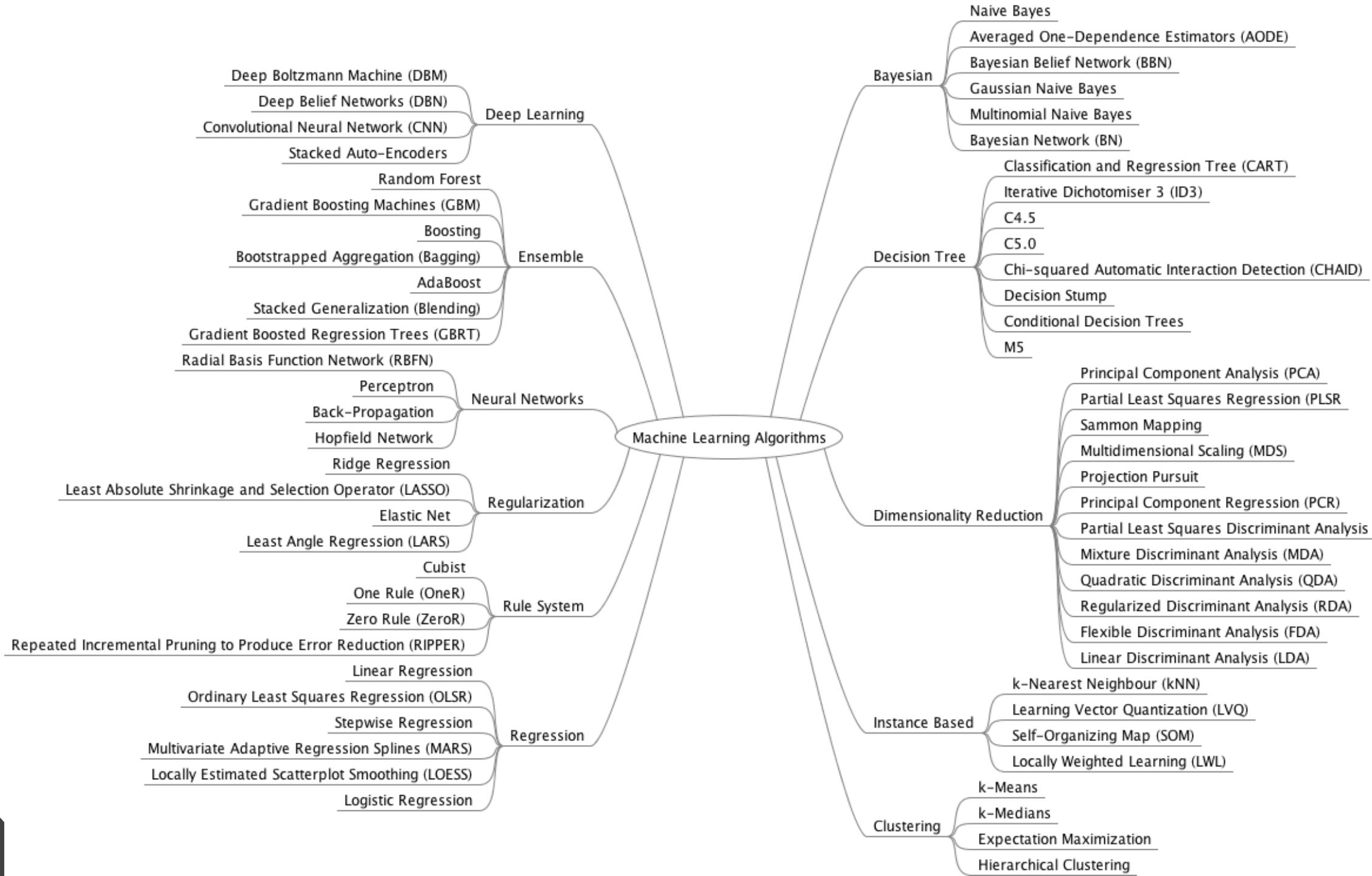


Apprentissage par renforcement



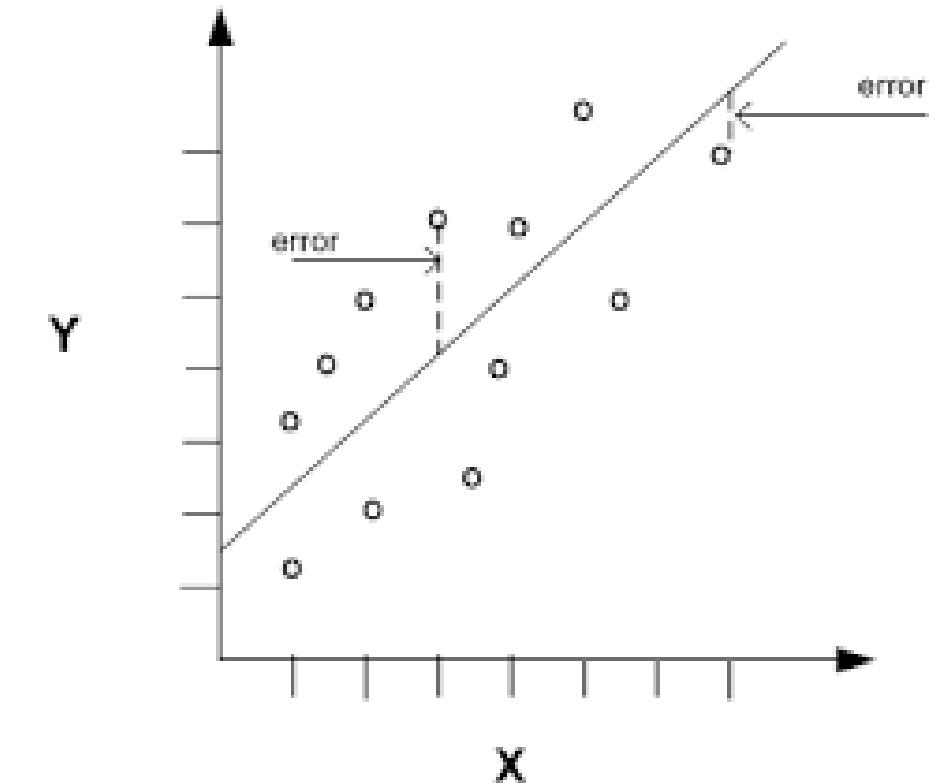


Les algorithmes de machine learning



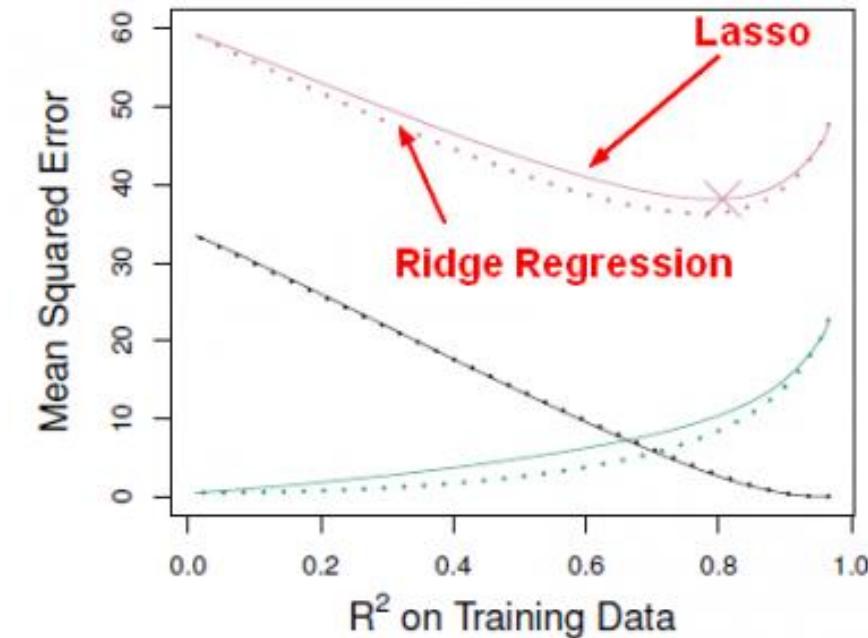
Regression

- Machine learning supervisé
- Prédiction d'une variable quantitative à partir de prédicteurs
- Quelques méthodes :
 - Ordinary Least Squares Regression (OLSR)
 - Stepwise Regression
 - Multivariate Adaptive Regression Splines (MARS)
 - Locally Estimated Scatterplot Smoothing (LOESS)



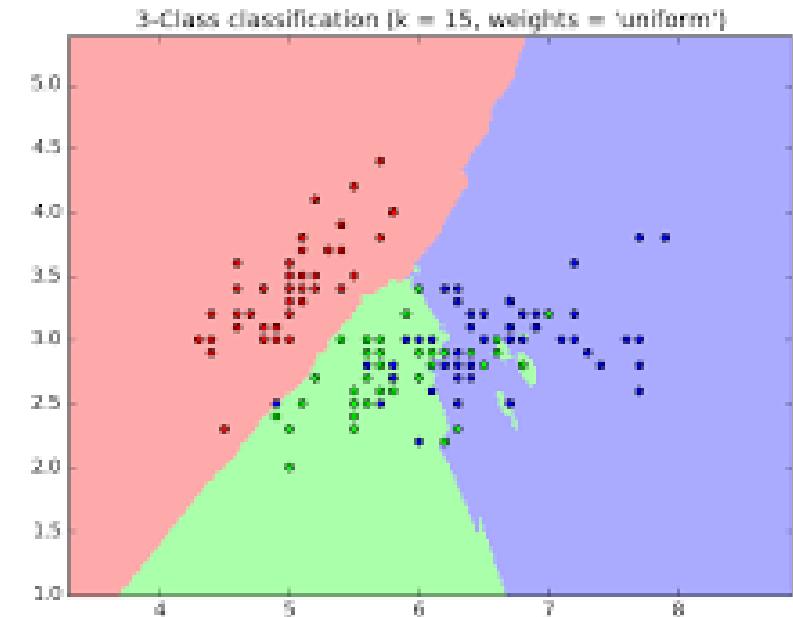
Regularization

- Machine learning supervisé
- Méthodes permettant de réduire le sur-apprentissage
- Quelques méthodes :
 - Ridge Regression
 - Least Absolute Shrinkage and Selection Operator (LASSO)
 - Elastic Net
 - Least-Angle Regression (LARS)



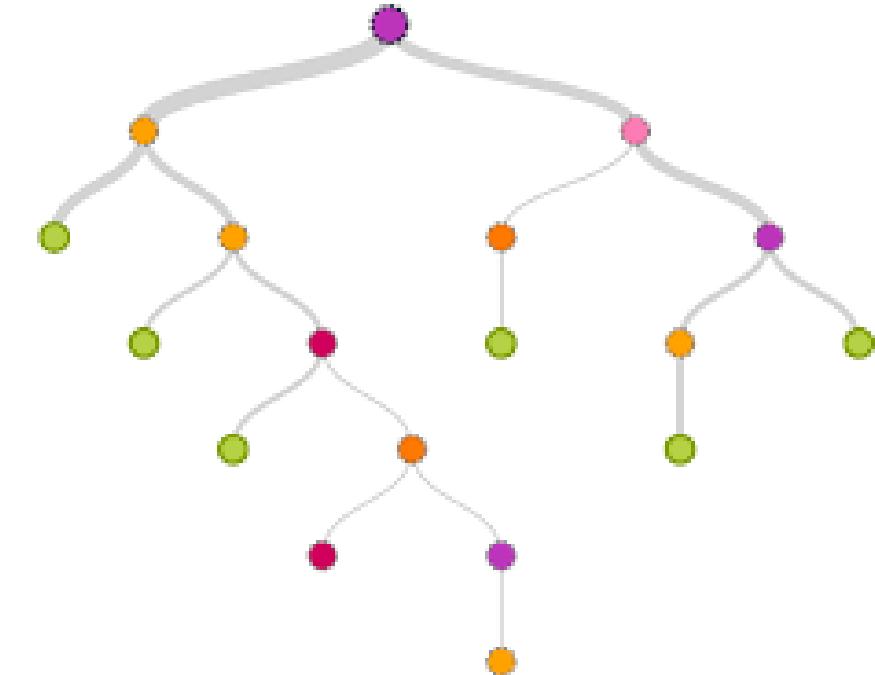
Instance-based

- Machine learning supervisé
- Prédiction d'une variables en étudiant les observations les plus proches
- Quelques méthodes :
 - k-Nearest Neighbour (kNN)
 - Learning Vector Quantization (LVQ)
 - Self-Organizing Map (SOM)
 - Locally Weighted Learning (LWL)



Arbre de décision

- Machine learning supervisé
- Prédiction d'une variable en construisant arbre de décision. Chaque branche permet de mieux discriminé la variable cible
- Quelques méthodes :
 - Classification and Regression Tree (CART)
 - Iterative Dichotomiser 3 (ID3)
 - C4.5 and C5.0
 - Chi-squared Automatic Interaction Detection (CHAID)



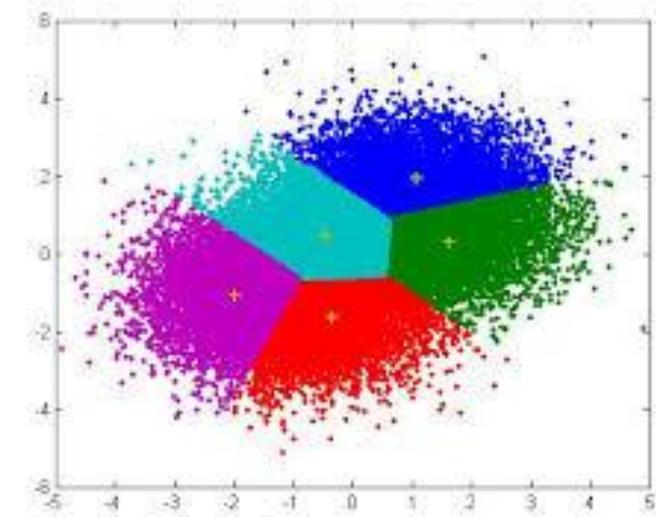
Bayesian

- Machine learning supervisé ou non supervisé
- Utilisation de la théorie de Bayes pour construire des modèles (utilisation de simulations)
- Quelques méthodes :
 - Naive Bayes
 - Gaussian Naive Bayes
 - Multinomial Naive Bayes
 - Bayesian Belief Network (BBN)
 - Bayesian Network (BN)



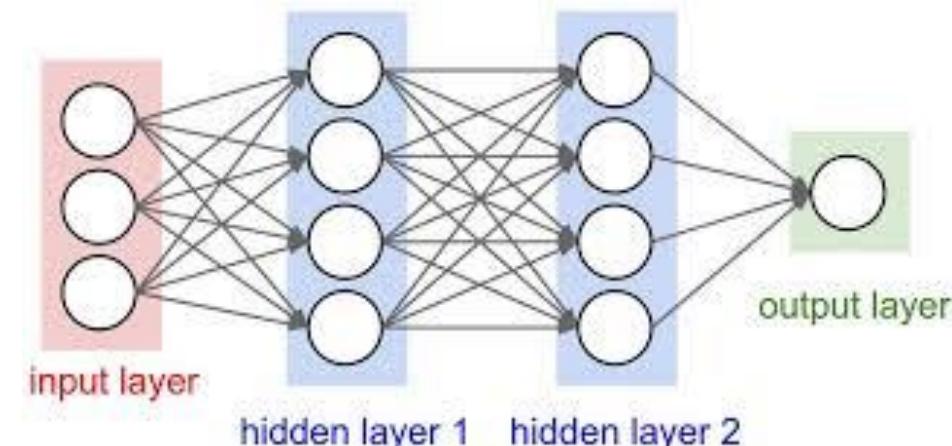
Clustering

- Machine learning non supervisé
- Obtention de groupes d'observations
- Quelques méthodes :
 - k-Means
 - k-Medians
 - Expectation Maximisation (EM)
 - Hierarchical Clustering



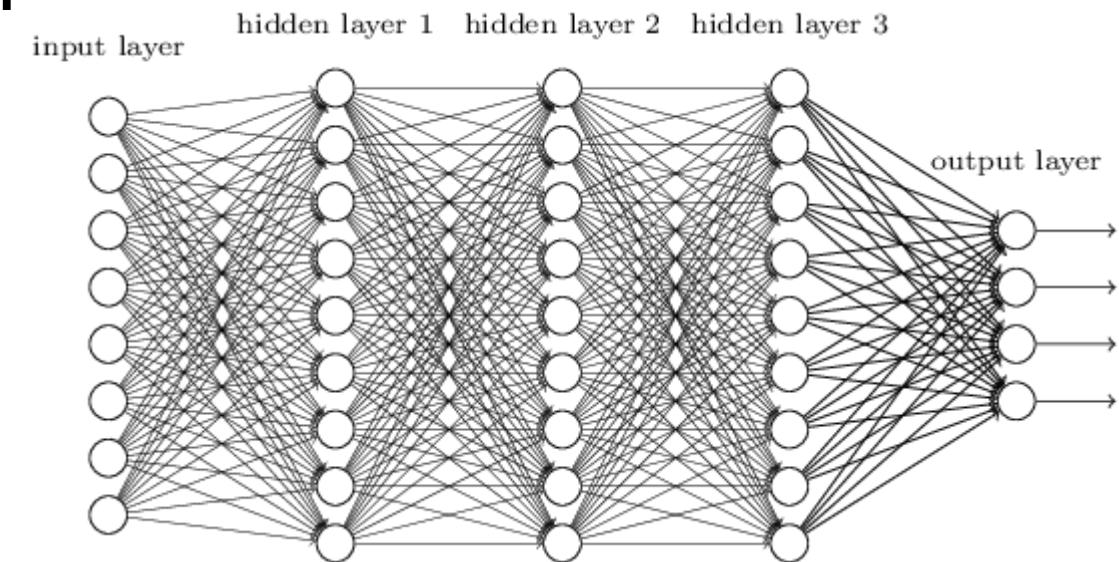
Artificial Neural Network

- Machine learning supervisé
- Prédiction d'une variables en utilisant des modèles de réseaux de neurones
- Quelques méthodes :
 - Perceptron
 - Back-Propagation
 - Radial Basis Function Network (RBFN)



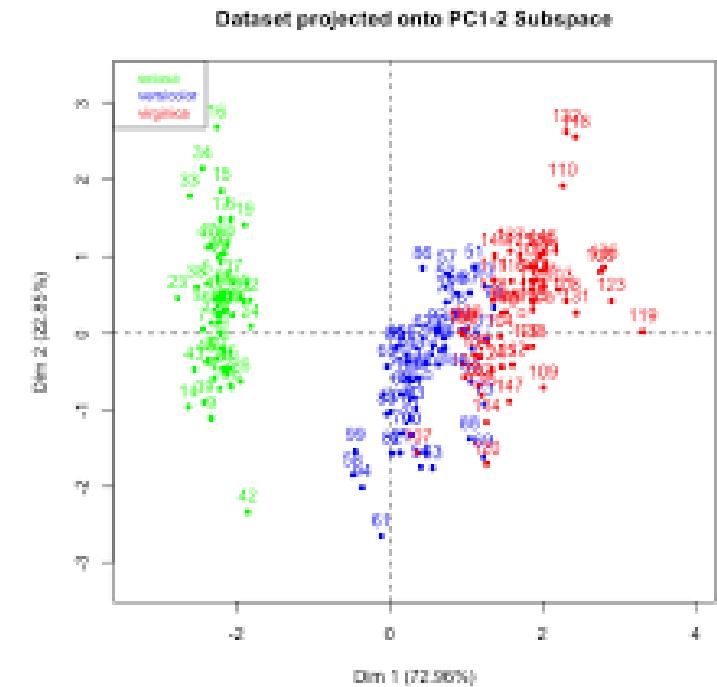
Deep Learning

- Deep learning supervisé ou non supervisé
- Utilisation de réseaux de neurones profonds
- Quelques méthodes :
 - Deep Belief Networks (DBN)
 - Convolutional Neural Network (CNN)
 - Stacked Auto-Encoders
 - Recurrent neural networks (RNN)



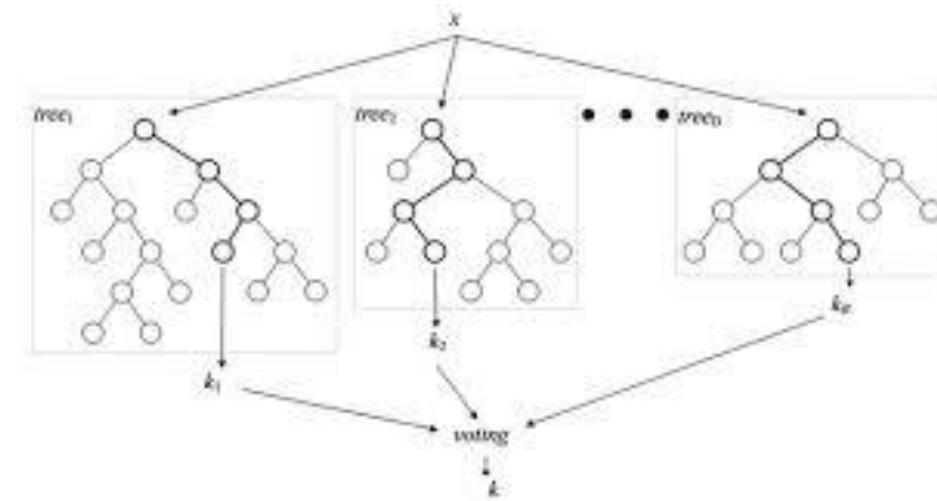
Réduction de dimension

- Principal Component Analysis (PCA)
- Principal Component Regression (PCR)
- Partial Least Squares Regression (PLSR)
- Multidimensional Scaling (MDS)
- Linear Discriminant Analysis (LDA)



Ensemble Algorithms

- Machine learning supervisé
- Combinaison de nombreux algorithmes d'apprentissage supervises simples pour obtenir un algorithme très efficace
- Quelques méthodes :
 - Boosting
 - Bootstrapped Aggregation (Bagging)
 - AdaBoost
 - Gradient Boosting Machines (GBM)
 - Random Forest

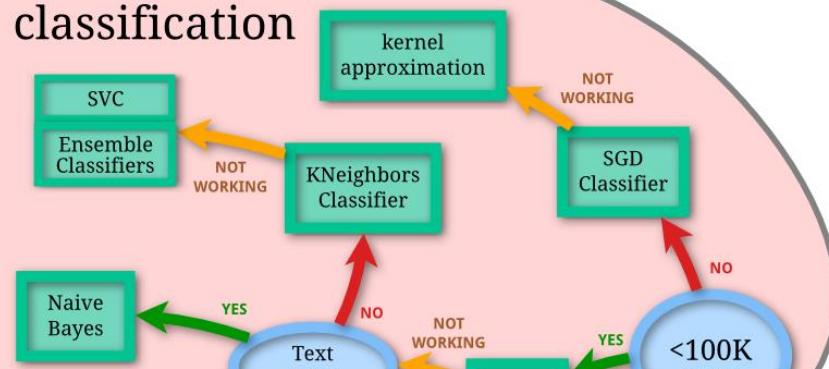


Quelle méthode utiliser ?

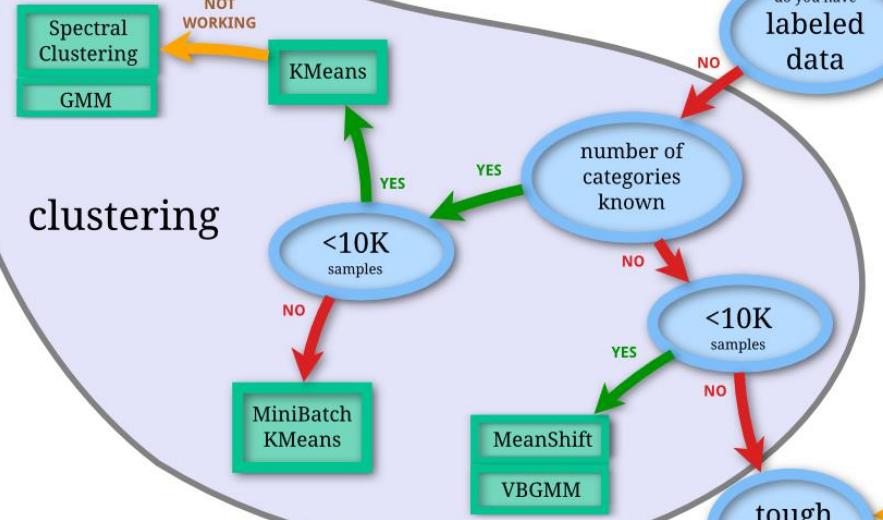
- Il n'existe pas de règle permettant d'associer à un cas, une méthode
- L'utilisation du machine learning est basé sur un processus itératif
- Points clés : il faut connaître les forces et les faiblesses de chaque approche afin d'accélérer vos développements
- L'obtention du meilleur modèle est un processus itératif long

scikit-learn algorithm cheat-sheet

classification



clustering



Back

scikit
learn

get
more
data

more
data

>50
samples

predicting a
category

predicting a
quantity

just
looking

predicting
structure

START

regression

SGD
Regressor

Lasso
ElasticNet

SVR(kernel='rbf')
EnsembleRegressors

RidgeRegression
SVR(kernel='linear')

few features
should be
important

Randomized
PCA

Isomap
Spectral
Embedding

LLE

kernel
approximation

dimensionality
reduction

tough
luck

structure

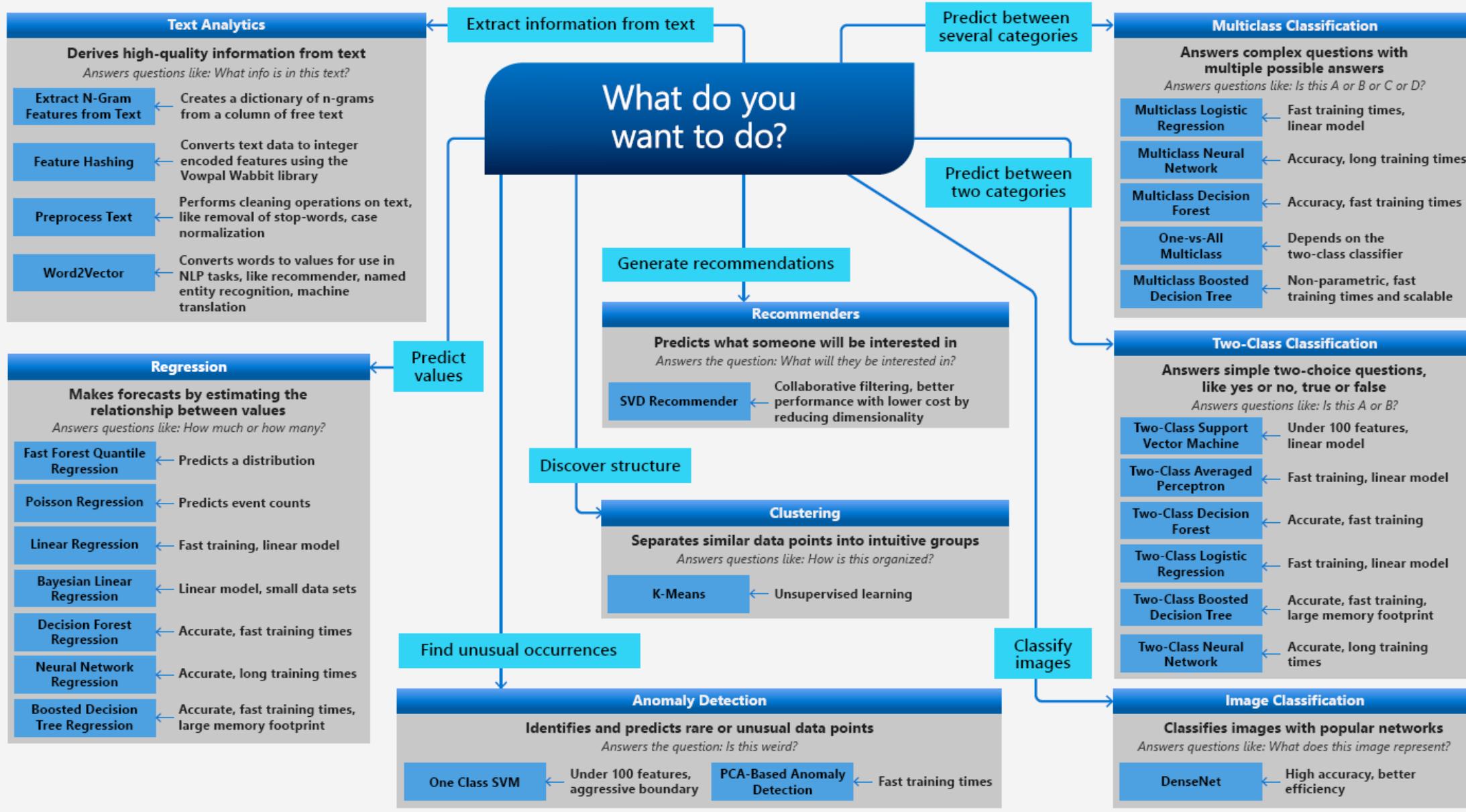
reduction

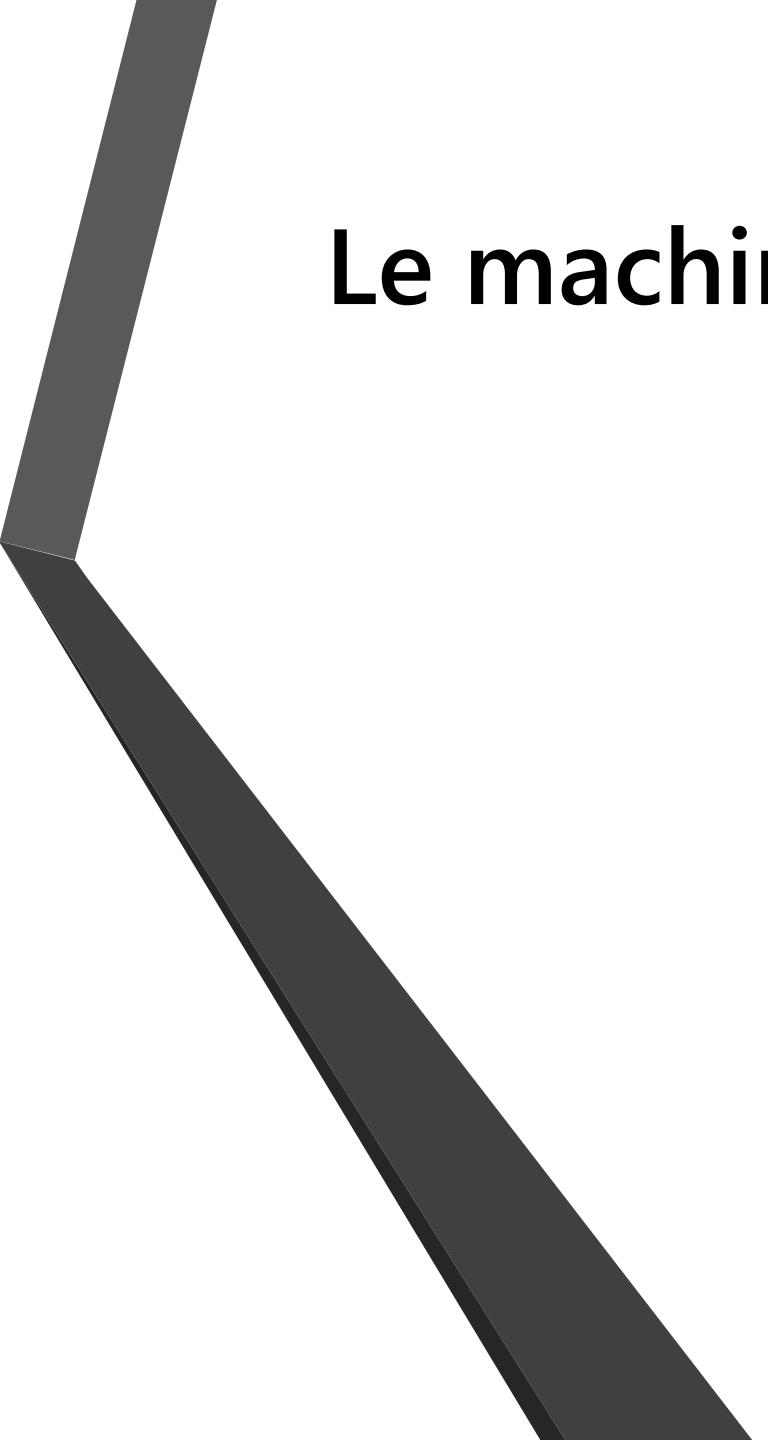


Microsoft Azure Machine Learning Algorithm Cheat Sheet

This cheat sheet helps you choose the best machine learning algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the goal you want to achieve with your data.

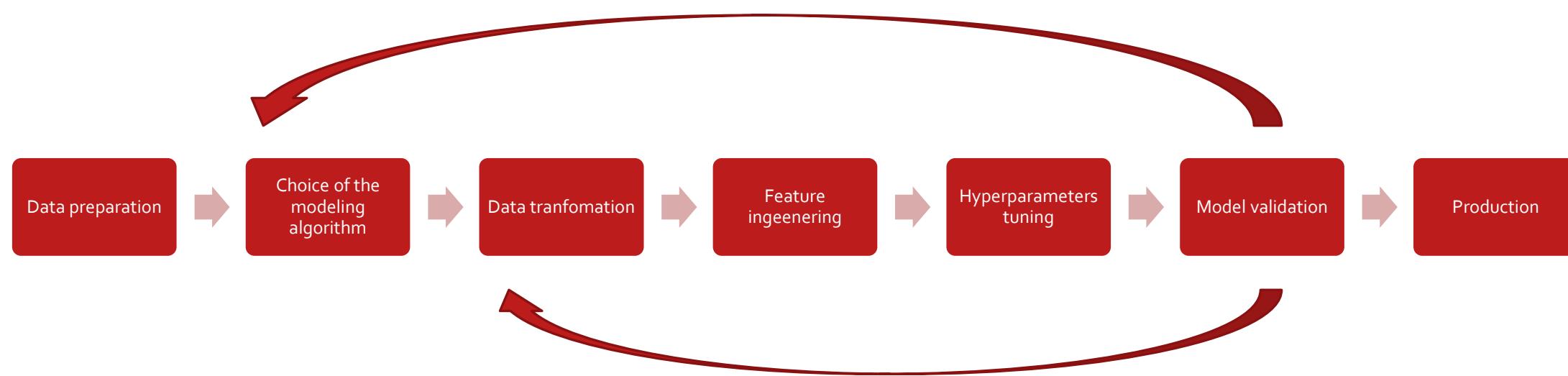
ecision
science company



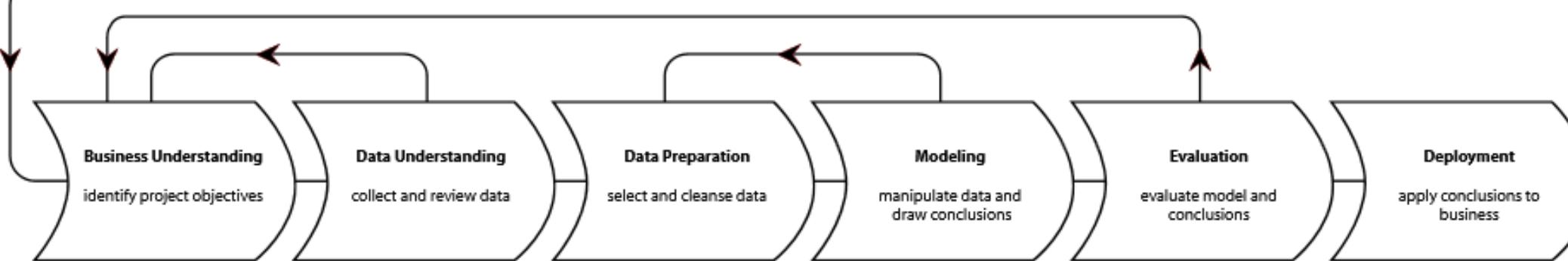


Le machine learning – Le processus de machine learning

Process de machine learning



Data Mining Life Cycle



Determine Business Objectives <i>Background</i> <i>Business Objectives</i> <i>Business Success Criteria</i> (Log and Report Process)	Collect Initial Data <i>Initial Data Collection Report</i> (Log and Report Process)	Data Set <i>Data Set Description</i> (Log and Report Process)	Select Modeling Technique <i>Modeling Technique</i> <i>Modeling Assumptions</i> (Log and Report Process)	Evaluate Results <i>Align Assessment of Data Mining Results with Business Success Criteria</i> (Log and Report Process)	Plan Deployment <i>Deployment Plan</i> (Log and Report Process)
Assess Situation <i>Inventory of Resources, Requirements, Assumptions, and Constraints</i> <i>Risks and Contingencies</i> <i>Terminology</i> <i>Costs and Benefits</i> (Log and Report Process)	Describe Data <i>Data Description Report</i> (Log and Report Process)	Select Data <i>Rationale for Inclusion/Exclusion</i> (Log and Report Process)	Generate Test Design <i>Test Design</i> (Log and Report Process)	Approved Models <i>Review Process</i> <i>Review of Process</i> (Log and Report Process)	Plan Monitoring and Maintenance <i>Monitoring and Maintenance Plan</i> (Log and Report Process)
Determine Data Mining Goals <i>Data Mining Goals</i> <i>Data Mining Success Criteria</i> (Log and Report Process)	Explore Data <i>Data Exploration Report</i> (Log and Report Process)	Clean Data <i>Data Cleaning Report</i> (Log and Report Process)	Build Model Parameter Settings <i>Models</i> <i>Model Description</i> (Log and Report Process)	Determine Next Steps <i>List of Possible Actions</i> <i>Decision</i> (Log and Report Process)	Produce Final Report <i>Final Report</i> <i>Final Presentation</i> (Log and Report Process)
Produce Project Plan <i>Project Plan</i> <i>Initial Assessment of Tools and Techniques</i> (Log and Report Process)	Verify Data Quality <i>Data Quality Report</i> (Log and Report Process)	Construct Data <i>Derived Attributes</i> <i>Generated Records</i> (Log and Report Process)	Assess Model <i>Model Assessment</i> <i>Revised Parameter</i> (Log and Report Process)		Review Project <i>Experience</i> <i>Documentation</i> (Log and Report Process)
		Integrate Data <i>Merged Data</i> (Log and Report Process)	Format Data <i>Reformatted Data</i> (Log and Report Process)		

a visual guide to CRISP-DM methodology

Generic Tasks
Specialized Tasks
(Process Instances)

SOURCE CRISP-DM 1.0
<http://www.crisp-dm.org/download.htm>

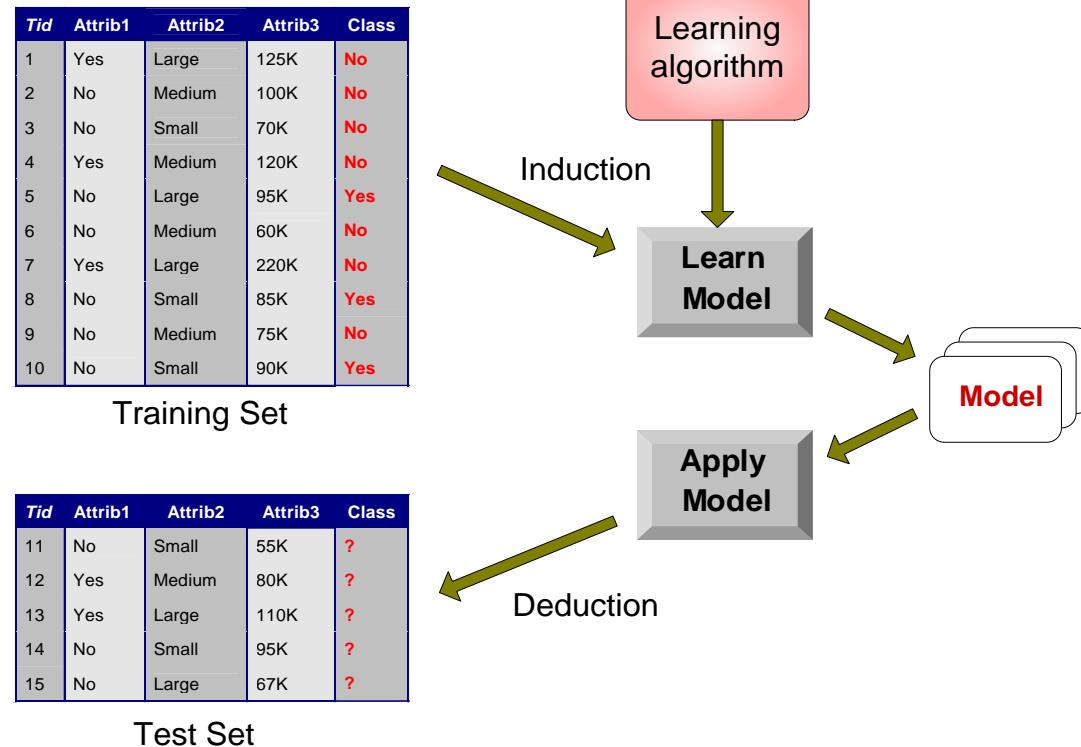
DESIGN Nicole Leaper
<http://www.nicoleleaper.com>



Classification – 2 étapes

- **Construction de modèle:** décrivant un ensemble de classes prédéterminées
 - Chaque tuple / échantillon est supposé appartenir à une classe prédéfinie, comme déterminé par l'étiquette de la classe
 - L'ensemble de tuples utilisé pour la construction du modèle est un ensemble d'apprentissage
 - Le modèle est représenté sous la forme de règles de classification, d'arbres de décision ou de formules mathématiques.
- **Utilisation du modèle:** pour classifier des objets futurs ou inconnus
 - Estimer la précision du modèle
 - L'étiquette connue de l'échantillon à tester est comparée au résultat classifié du modèle
 - L'ensemble de test est indépendant de l'ensemble de formation, sinon un sur-ajustement se produira
 - Si la précision est acceptable, utilisez le modèle pour classer les nuplets de données dont les étiquettes de classe ne sont pas connues.

Illustration



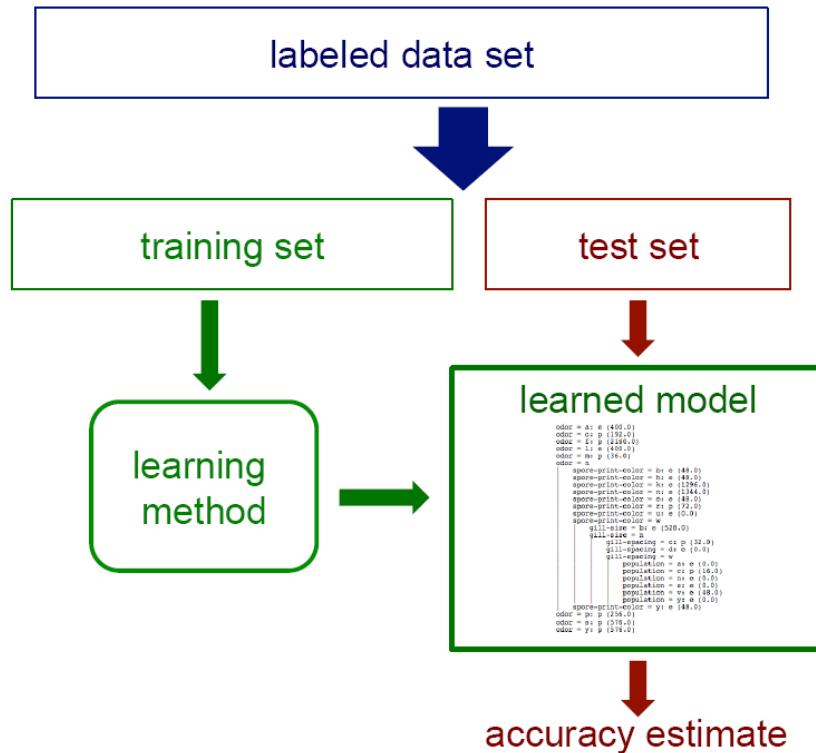
Comment valide-t-on un modèle ?

Validation du modèle

- Comment pouvons-nous obtenir une estimation de la précision d'un modèle appris?
 - lors de l'apprentissage d'un modèle, vous devez faire semblant de ne pas disposer des données de test.
 - si les étiquettes de l'ensemble de tests influencent le modèle appris de quelque manière que ce soit, les estimations de l'exactitude seront biaisées

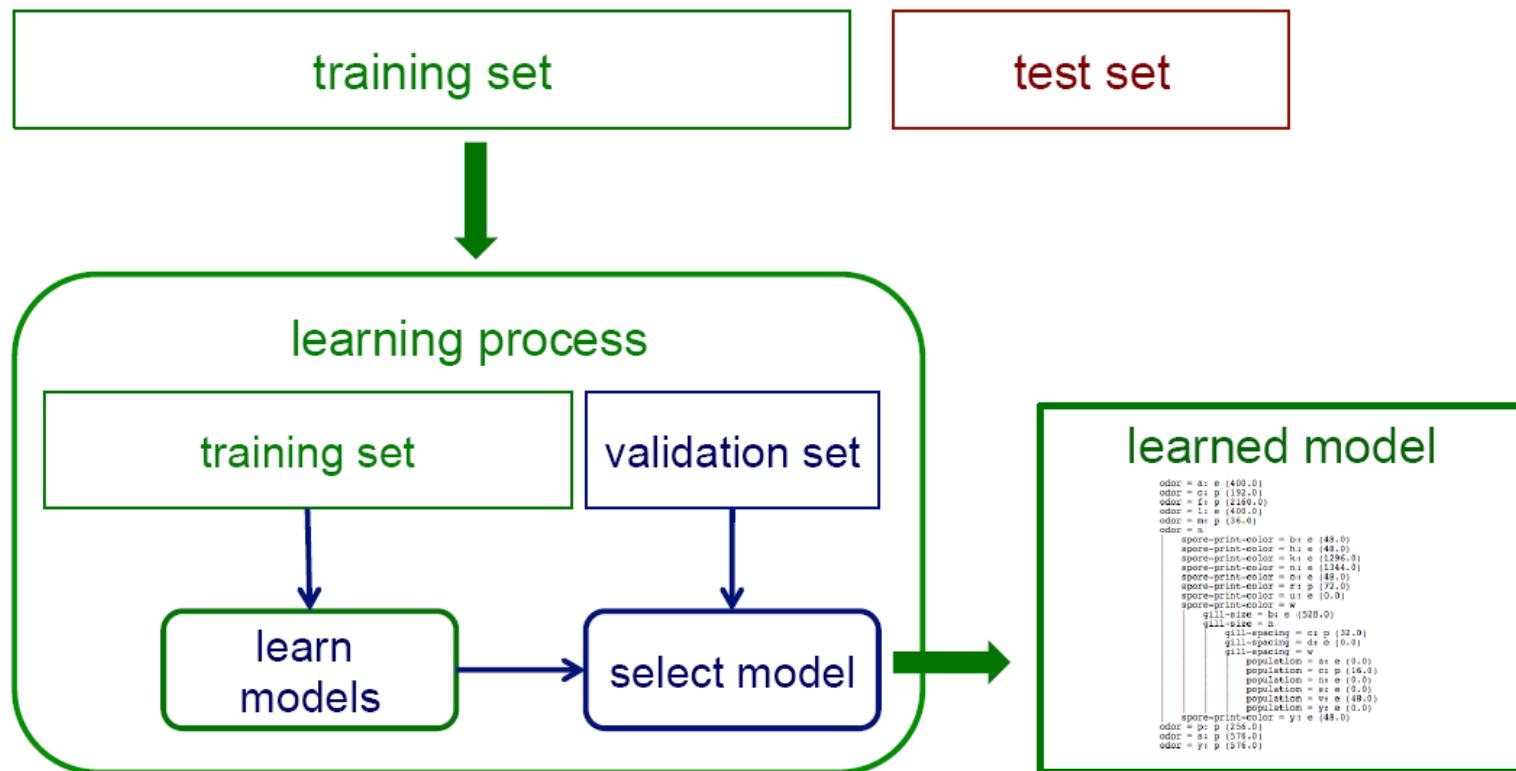
La démarche Train/Test

- On coupe le jeu de données en deux (échantillon d'apprentissage et échantillon test).
- On demande à la machine d'apprendre la tâche sur l'échantillon d'apprentissage.
- On demande à la machine d'exécuter la tâche apprise sur l'échantillon test.
- On compare les classes réelles et les classes prédites par la machine sur l'échantillon test. Si elles *matchent bien*, l'algo est performant. → il y a des indices permettant de mesurer cette performance.



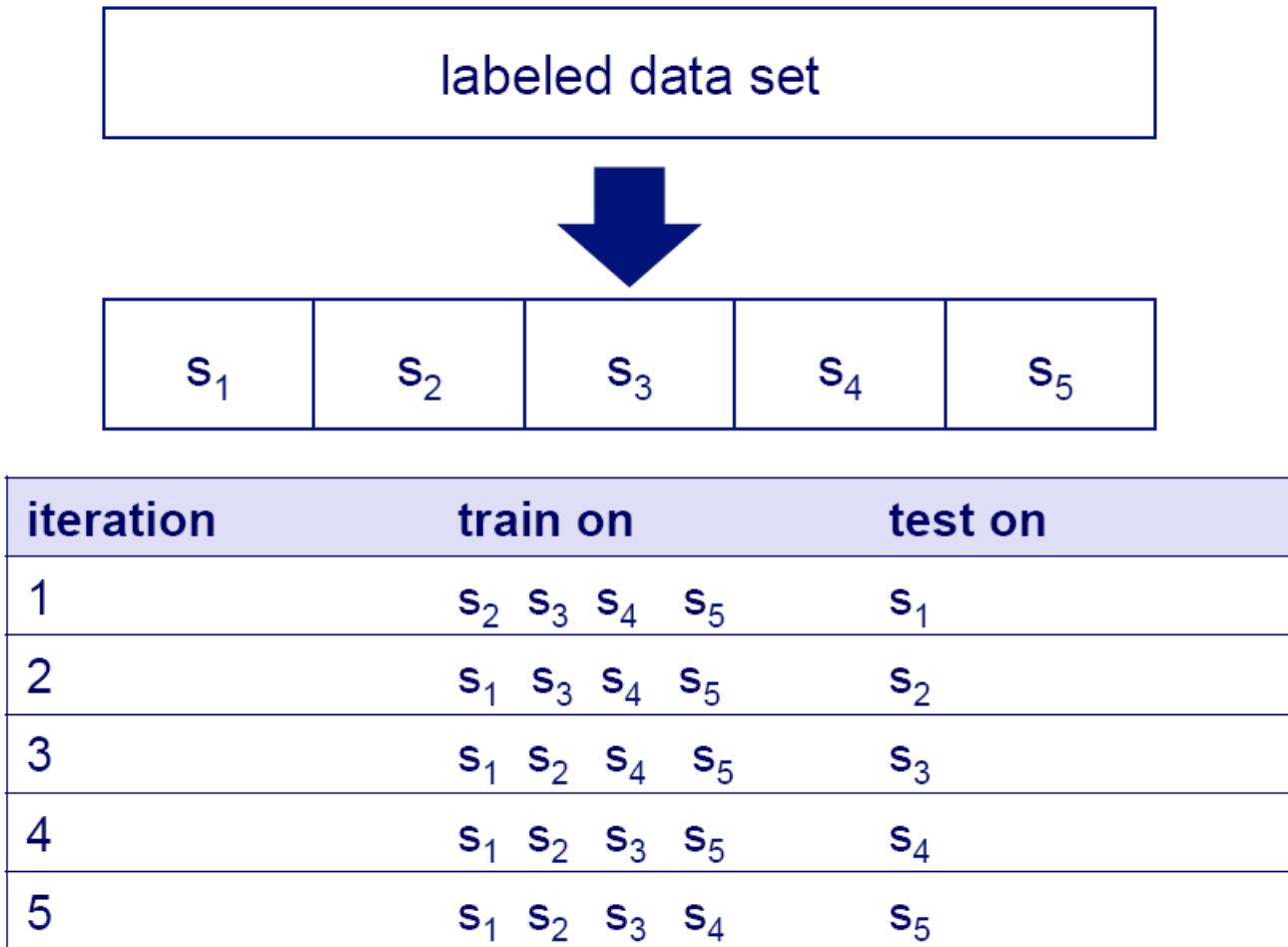
Apprentissage / validation / test

- Nous pourrions avoir besoin d'un jeu de validation supplémentaire pour estimer les valeurs d'hyperparamètres
- Dans ce cas, les données sont divisées en 3 sous-ensembles de données.

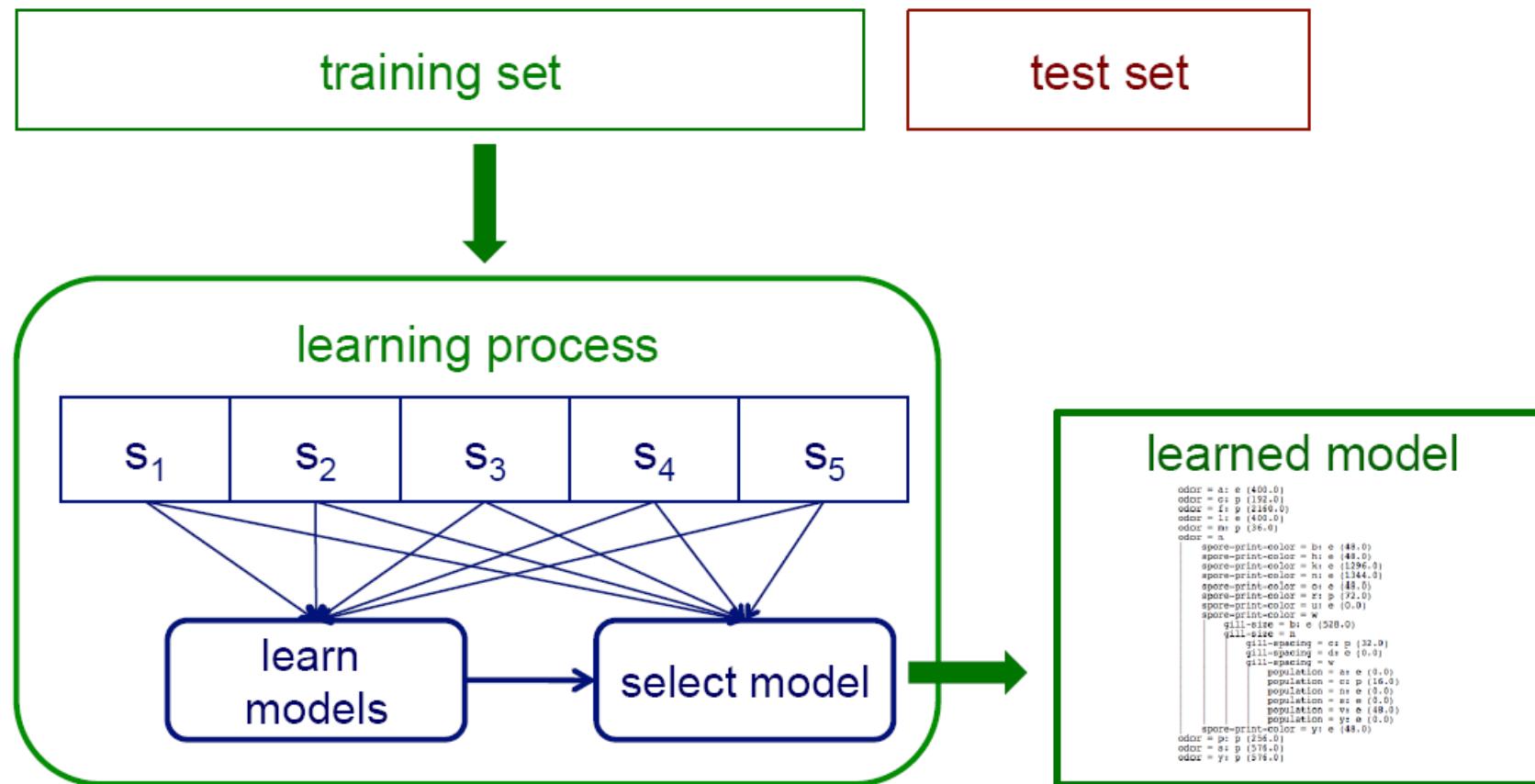


Validation croisée

- Itérativement, laissez un échantillon hors du jeu d'apprentissage et utilisez-le à des fins de test
- La validation croisée est utilisée pour valider votre modèle et est plus robuste qu'un train / test classique
- k devrait être choisi en fonction de différentes variables:
 - k sera plus grand quand N est grand
 - k sera plus petit si le temps d'entraînement est long
 - Les valeurs sont généralement comprises entre 5 et 10



Validation croisée



Ajustement des hyper-paramètres

- De nombreux algorithmes d'apprentissage automatique ont des hyperparamètres et il est difficile de fixer leur valeur.
 - Nombre de voisins dans k-NN
 - Profondeur d'un arbre
 - Noyau en SVM
 - ...
- La méthode principale est la recherche sur grille qui consiste à tester toutes les combinaisons possibles sur une grille.

Ajustement des hyper-paramètres

- Une recherche comprend:
 - un estimateur (régresseur ou classificateur);
 - un espace de paramètres;
 - une méthode de recherche ou d'échantillonnage de candidats;
 - un schéma de validation croisée; et
 - une fonction de score.

Validation du modèle

- Paramètres d'évaluation des performances
 - Comment évaluer les performances d'un modèle?
- Méthodes d'évaluation des performances
 - Comment obtenir des estimations fiables?
- Méthodes de comparaison de modèles
 - Comment comparer les performances relatives des modèles concurrents?

Quelques métriques

- Error Metrics for Regression Problems
 - Mean Absolute Error, Weighted Mean Absolute Error, Root Mean Squared Error, Root Mean Squared Logarithmic Error
- Error Metrics for Classification Problems
 - Logarithmic Loss, Mean F Score, Multi Class Log Loss, Hamming Loss, Mean Utility
- Métriques
 - Area Under Curve (AUC), Gini, Average Among Top P, Average Precision, Mean Average Precision

Métriques

- Voici les résultats des mesures les données UCI sur le cancer du sein

[http://mlr.cs.umass.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](http://mlr.cs.umass.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	71.7	.4534	.44	.16	.53	.44	.48	.7	48.11
C4.5	75.5	.4324	.27	.04	.74	.27	.4	.59	34.28
3NN	72.4	.5101	.32	.1	.56	.32	.41	.63	43.37
Ripp	71	.4494	.37	.14	.52	.37	.43	.6	22.34
SVM	69.6	.5515	.33	.15	.48	.33	.39	.59	54.89
Bagg	67.8	.4518	.17	.1	.4	.17	.23	.63	11.30
Boost	70.3	.4329	.42	.18	.5	.42	.46	.7	34.48
RanF	69.23	.47	.33	.15	.48	.33	.39	.63	20.78

Limite du % de bien classés

- Considérons un problème à 2 classes
 - Nombre d'exemples de classe 0 = 9990
 - Nombre d'exemples de classe 1 = 10
- Si le modèle prédit que tout est en classe 0, le % de ben classés est de $9990/10000 = 99,9\%$
- L'accuracy est trompeuse car le modèle ne détecte aucun exemple de classe 1

Méthodes d'évaluation des performances

- Comment obtenir une estimation fiable de la performance?
- Les performances d'un modèle peuvent dépendre d'autres facteurs que l'algorithme d'apprentissage:
 - Distribution de classe
 - Coût d'une mauvaise classification
 - Taille des ensembles d'apprentissage et de test

ROC (Receiver Operating Characteristic)

- Caractériser le compromis entre les résultats positifs et les fausses alarmes
- La courbe ROC représente TP (vrai positifs sur l'axe des y) en fonction de FP (faux positifs sur l'axe des x)
- Performances de chaque classifieur représentées par un point sur la courbe ROC
 - changer le seuil de l'algorithme, la distribution de l'échantillon ou la matrice des coûts modifie l'emplacement du point

ROC (Receiver Operating Characteristic)

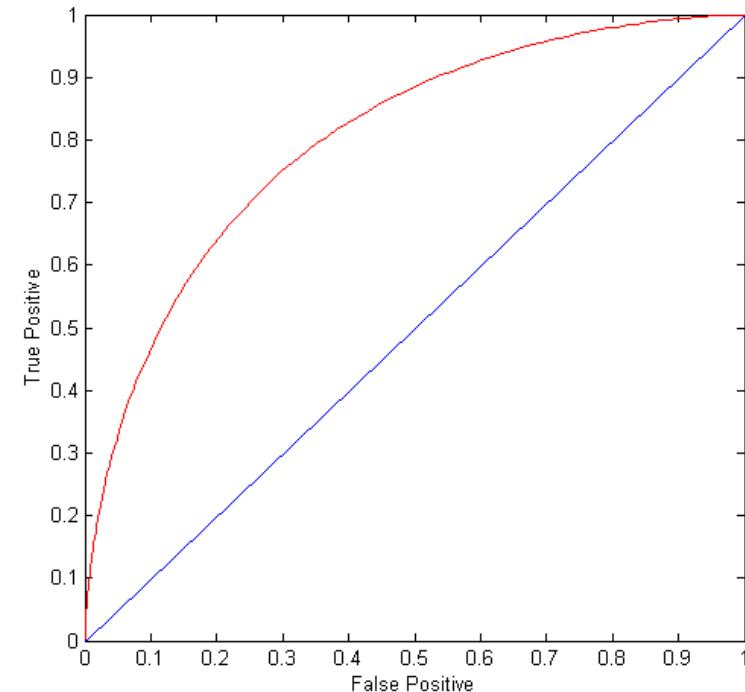
- Une courbe ROC (receiver operating characteristic) est un graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux de vrais positifs en fonction du taux de faux positifs :
 - Taux de vrais positifs
 - Taux de faux positifs
- Une courbe ROC trace les valeurs TVP et TFP pour différents seuils de classification. Diminuer la valeur du seuil de classification permet de classer plus d'éléments comme positifs, ce qui augmente le nombre de faux positifs et de vrais positifs. La figure ci-dessous représente une courbe ROC classique.

Courbe ROC

(TP,FP):

- (0,0): déclarer que tout est de classe négative
- (1,1): déclarer que tout est positif
- (1,0): idéal

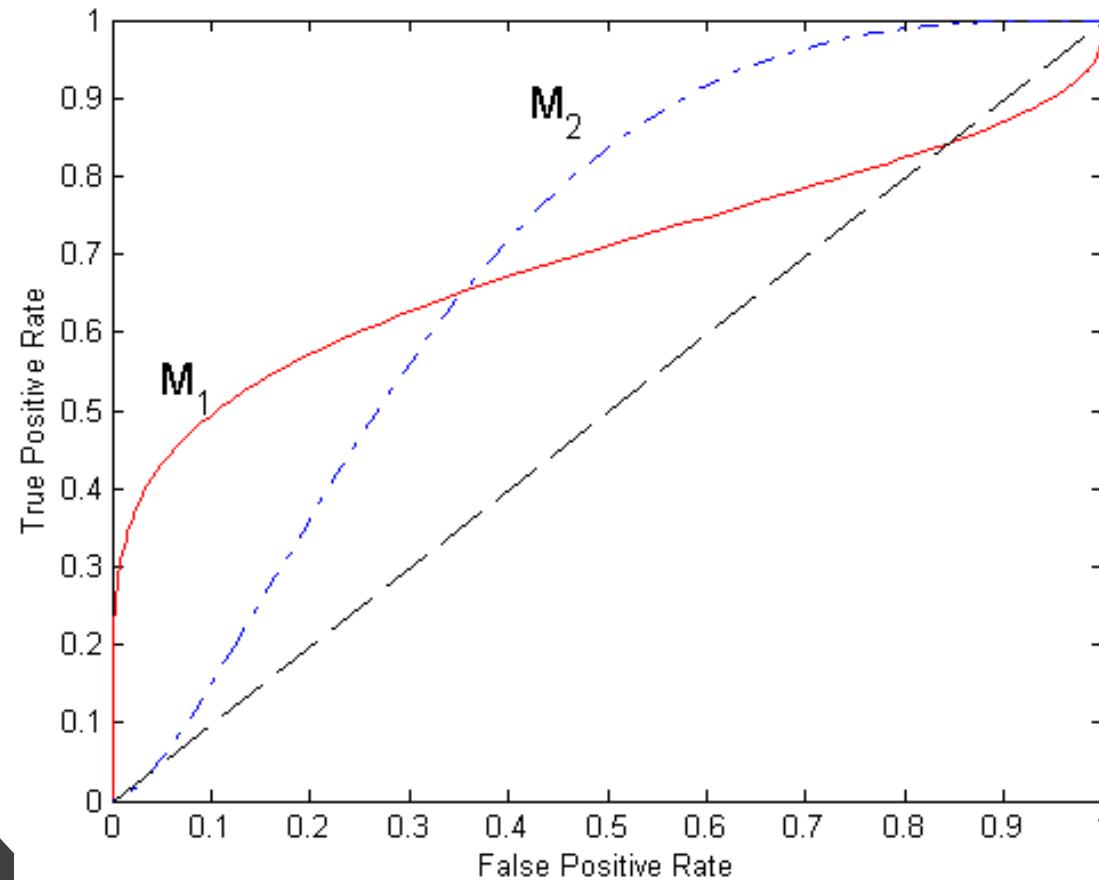
- Ligne diagonale:
 - Deviner au hasard
- En dessous de la diagonale:
 - la prédition est opposée à la vraie classe



Aire sous la courbe ROC

- AUC : aire sous la courbe ROC
- AUC signifie "aire sous la courbe ROC". Cette valeur mesure l'intégralité de l'aire à deux dimensions située sous l'ensemble de la courbe ROC (par calculs d'intégrales) de (0,0) à (1,1).
- L'AUC présente les avantages suivants :
 - L'AUC est invariante d'échelle. Elle mesure la qualité du classement des prédictions, plutôt que leurs valeurs absolues.
 - L'AUC est indépendante des seuils de classification. Elle mesure la qualité des précisions du modèle quel que soit le seuil de classification sélectionné.

Utilisation de la courbe ROC pour comparer des modèles



- Aucun modèle ne surpassé systématiquement l'autre
 - M₁ est mieux pour les petits FPR
 - M₂ est mieux pour les grands FPR
- Surface sous la courbe ROC
 - Idéal : Surface = 1
 - Devine au hasard: Surface = 0.5

Exemple : la classification

La classification en apprentissage supervisé

- Objectif : Être capable de prédire l'attrition d'un client
- Cible : Oui / Non
- Outils :
 - Régression logistique
 - SVM
 - Arbres de décision
 - Forêts aléatoires
 - Gradient Boosting Machine

Données

- Données sur un opérateur de télécommunication
- Cible : Churn?
- Variables explicatives : données clients

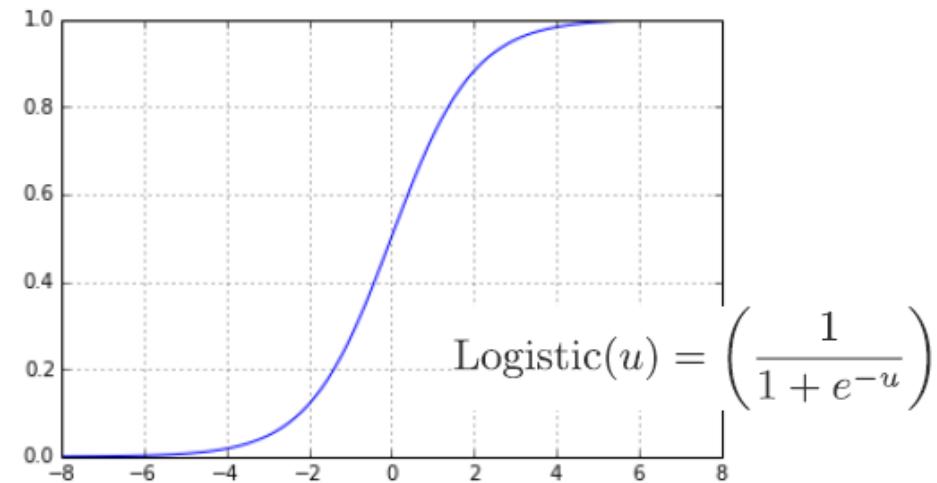
Utilisation de Python

- Préparation des données
 - Séparation apprentissage / test
-
- Passage au Notebook
 - Connectez-vous au JupyterHub avec le nom d'user donné par le formateur ou avec votre Anaconda local



1^{ère} approche : La régression logistique

- Classe de modèles
 - Forme de la fonction de décision – logistique



Modéliser la **probabilité** que x réponde au traitement

$$P(Y = 1|x) \approx a_1x_1 + a_2x_2 + \dots + a_px_p + b$$

+ transformer un nombre réel en un nombre entre 0 et 1

$$P(Y = 1|x) \approx \text{logistic} (a_1x_1 + a_2x_2 + \dots + a_px_p + b)$$

La régression logistique

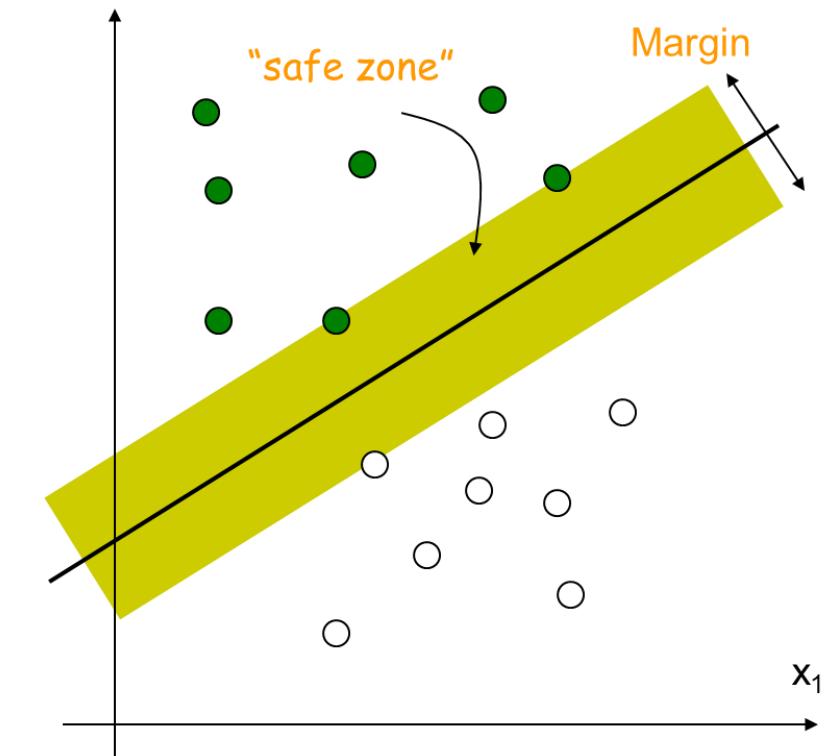
- On utilise une fonction linéaire pour modéliser la probabilité d'appartenance
- On utilise la méthode du maximum de vraisemblance pour estimer les paramètres de la régression
- Avantages:
 - Ne fait aucune hypothèse sur les distributions de classes dans l'espace des fonctionnalités
 - Étendu facilement à plusieurs classes (régression multinomiale)
 - Vision probabiliste naturelle des prédictions de classe
 - Apprentissage rapide
 - Bonne précision pour de nombreux ensembles de données simples
 - Résistant à l'overfitting
- Désavantages:
 - Décision linéaire

2nde approche : les SVM

- Ce sont des algorithmes d'apprentissage initialement construits pour la classification binaire.
- L'idée est de rechercher une règle de décision basée sur une séparation par hyperplan de marge optimale.
- Méthode relativement récente qui découle de premiers travaux théoriques de Vapnik et Chervonenkis en 1995, démocratisées à partir de 2000.

Les SVM

- On maximise la marge
- On utilise un noyau pour projeter les observations dans un espace séparable



2^{nde} approche : les SVM

- **Avantages :**
 - Cela fonctionne vraiment bien lorsque la marge de séparation est claire
 - Il est efficace dans les espaces de grandes dimensions.
 - Il est efficace dans les cas où le nombre de dimensions est supérieur au nombre d'échantillons.
 - Il utilise un sous-ensemble de points d'entraînement dans la fonction de décision (appelés vecteurs de support), de sorte qu'il utilise également efficacement la mémoire.
- **Les inconvénients :**
 - Il ne fonctionne pas bien lorsque nous avons de grandes quantités de données, car le temps d'apprentissage requis est plus long.
 - Il ne fonctionne également pas très bien lorsque le jeu de données a plus de bruit, c'est-à-dire que les classes cibles se chevauchent.
 - SVM ne fournit pas directement d'estimations de probabilité, celles-ci sont calculées à l'aide d'une coûteuse validation croisée.

Arbres de décisions

Exemple de problème:

Décider si vous souhaitez attendre une table dans un restaurant, en fonction des attributs suivants:

- Alternative: y a-t-il un restaurant alternatif à proximité?
- Bar: y a-t-il un bar confortable à attendre?
- Vendredi / samedi: est aujourd'hui vendredi ou samedi?
- Hungry: avons-nous faim?
- Clients: nombre de personnes au restaurant (aucune, une partie, complète)
- Prix: fourchette de prix (\$, \$\$, \$\$\$)
- Il pleut: il pleut dehors?
- Réservation: avons-nous fait une réservation?
- Type: type de restaurant (français, italien, thaï, burger)
- Attente: temps d'attente estimé (0-10, 10-30, 30-60,> 60)

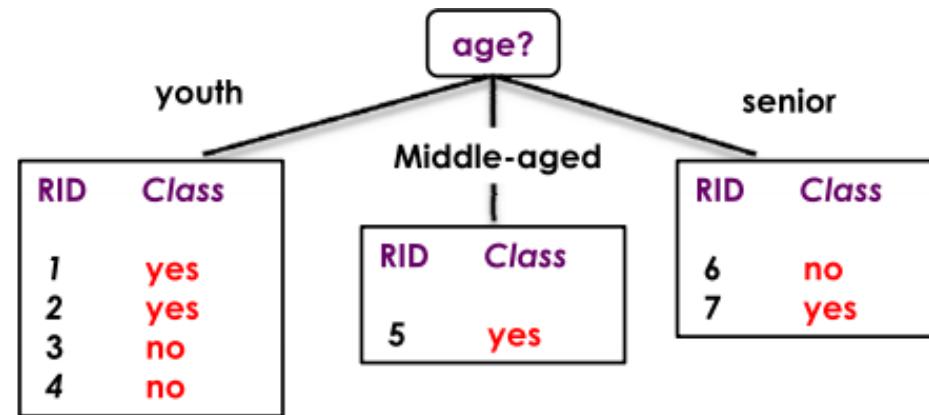
Représentations basées sur les attributs

Example	Attributes										Target <i>Wait</i>
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X_6	F	T	F	T	Some	\$\$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	\$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	\$\$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	\$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	\$\$\$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	\$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	\$	F	F	Burger	30–60	T

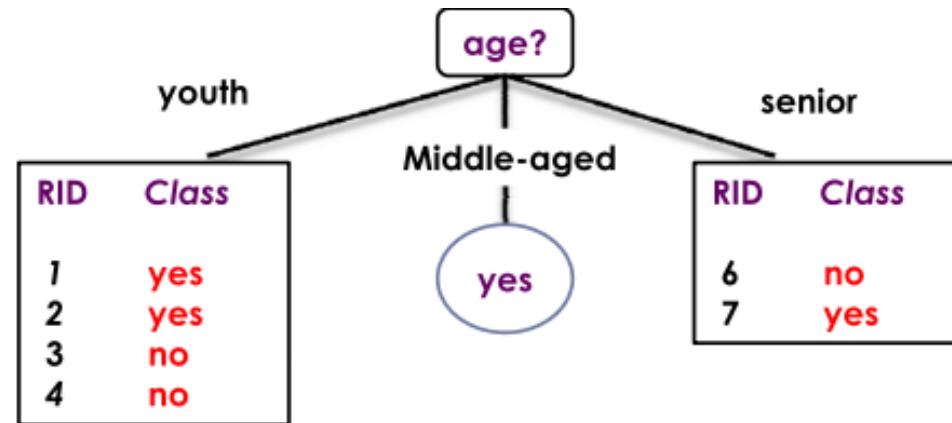
Construction d'un arbre

- **Principe**
 - Algorithme de base (adopté par ID3, C4.5 et CART): un algorithme glouton
 - L'arbre est construit de manière descendante
- **Itérations**
 - Au début, tous les observations d'apprentissage sont à la racine
 - Les observations sont partitionnés de manière récursive en fonction des attributs sélectionnés
 - Les attributs de test sont sélectionnés sur la base d'une mesure heuristique ou statistique (par exemple, gain d'information)
- **Conditions d'arrêt**
 - Tous les échantillons pour un noeud donné appartiennent à la même classe
 - Il n'y a pas d'attribut restant pour le partitionnement plus poussé
 - Il n'y a plus d'échantillons

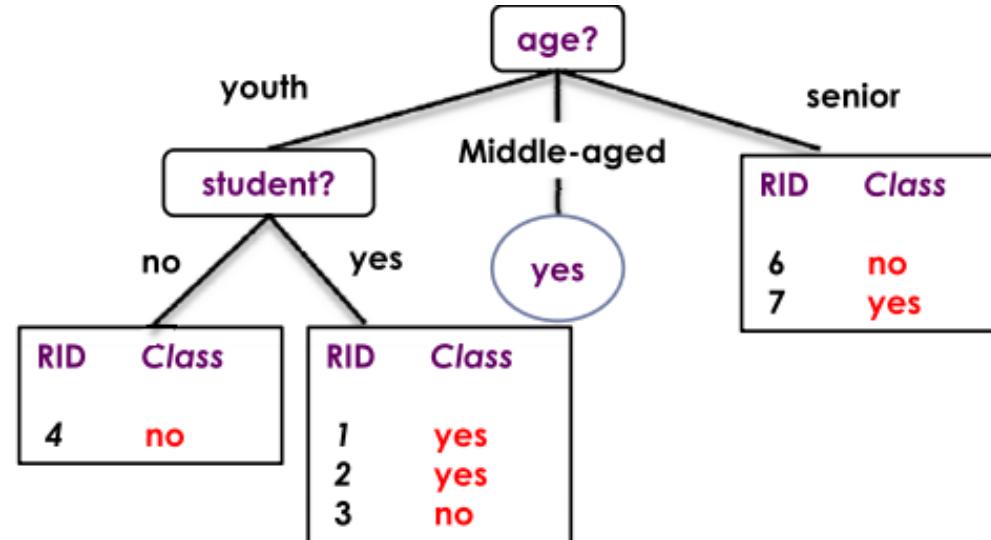
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no



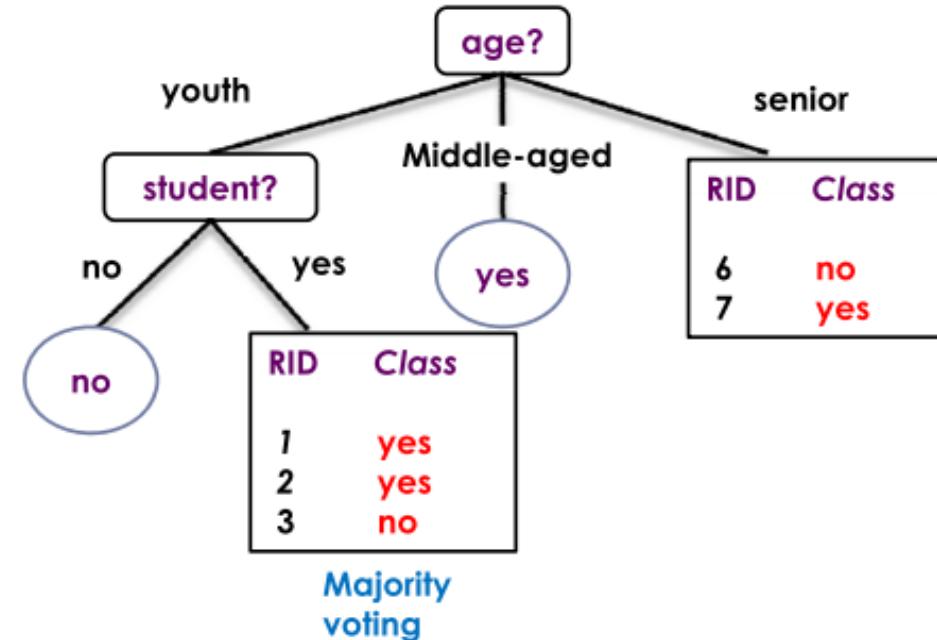
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



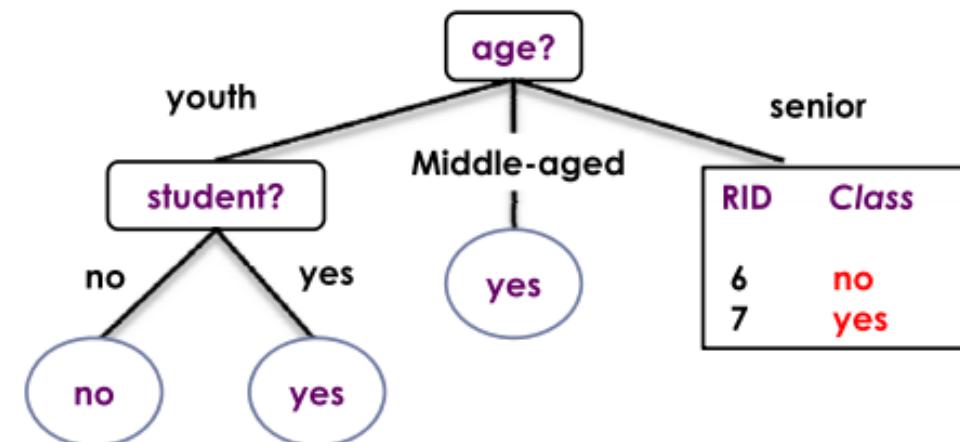
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



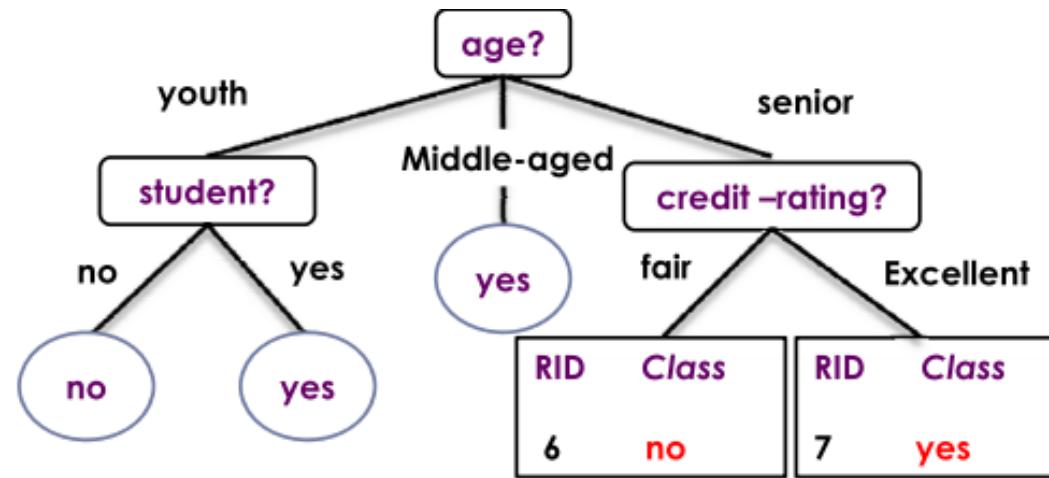
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



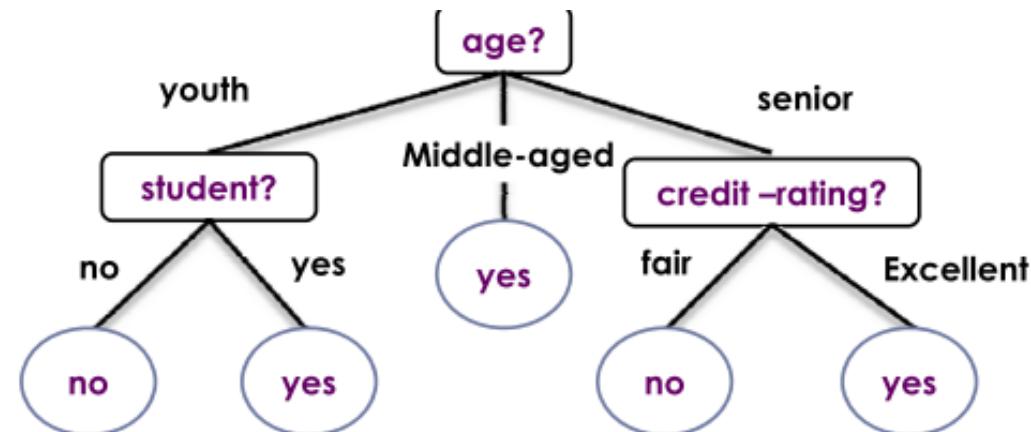
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



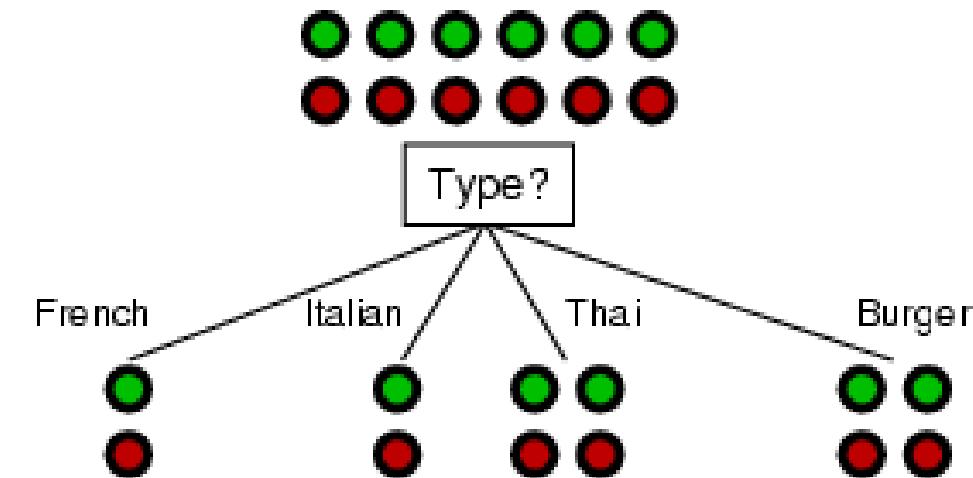
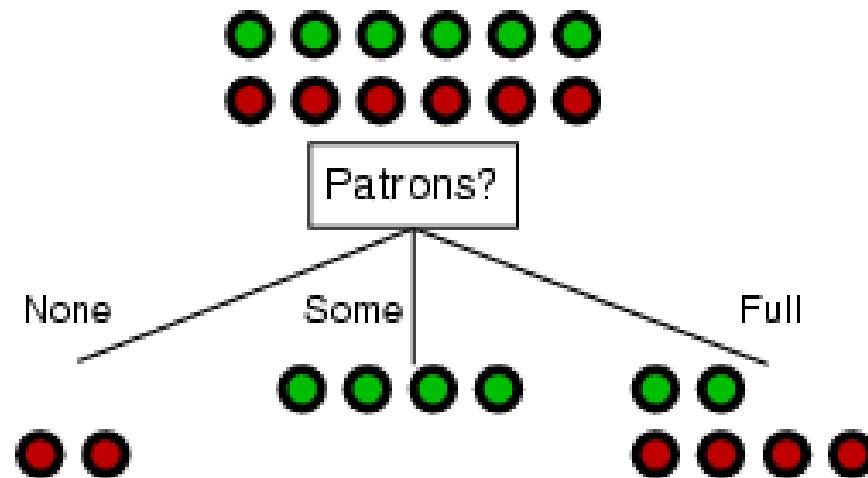
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	yes
7	senior	yes	excellent	no

Induction d'arbres

- Stratégie gloutone.
 - Divisez les enregistrements en fonction d'un indice qui optimise certains critères.
- Problèmes
 - Déterminer comment diviser les enregistrements
 - Comment spécifier la condition de test d'attribut?
 - Comment déterminer le meilleur partage?
 - Déterminer quand arrêter le fractionnement

Choix de la variable

- Idée: un bon attribut divise les exemples en sous-ensembles qui sont (idéalement) "tous positifs" ou "tous négatifs"



Comment determiner la meilleure séparation

- Approche gloutonne :
 - Les nœuds avec une distribution de classe homogène sont préférés
- Besoin d'une mesure de l'impureté du noeud

C0: 5
C1: 5

Non homogène,
Haut degré d'impureté

C0: 9
C1: 1

Homogène,
Faible degré d'impureté

Comparaison des méthodes de sélection d'attributs

- Les trois mesures donnent de bons résultats mais
 - Gain d'information:
 - biaisé vers des attributs à valeurs multiples
 - Ratio de gain:
 - a tendance à préférer les divisions non équilibrées dans lesquelles une partition est beaucoup plus petite que les autres
 - Indice de Gini:
 - biaisé en attributs à valeurs multiples
 - a des difficultés lorsque le nombre de classes est grand
 - a tendance à favoriser les tests aboutissant à des partitions de taille égale et à la pureté des deux partitions

Classification basée sur un arbre de décision

- Avantages:
 - Facile à construire / implémenter
 - Extrêmement rapide pour la classification des enregistrements inconnus
 - Les modèles sont faciles à interpréter pour les arbres de petite taille
 - La précision est comparable aux autres techniques de classification pour de nombreux ensembles de données simples
 - Les modèles arborescents ne supposent aucune hypothèse sur la distribution des données sous-jacentes: non paramétrique
 - Avoir une méthode de sélection de fonctions intégrée qui les rend insensibles à la présence de variables inutiles

Classification basée sur un arbre de décision

- Désavantages
 - Calcul coûteux en apprentissage
 - Certains arbres de décision peuvent être trop complexes et ne pas généraliser bien les données.
 - Moins d'expressivité: il peut y avoir des concepts difficiles à apprendre avec des arbres de décision limités

Overfitting et élagage

- Overfitting: un arbre peut sur-apprendre les données d'apprentissage
 - Trop de branches, certaines peuvent refléter des anomalies dues au bruit ou à des valeurs aberrantes
 - Mauvaise précision pour les nouveaux échantillons
- Deux approches pour éviter l'overfitting
 - Pré-élagage: arrêtez la construction de l'arbre à un stade précoce - ne divisez pas un nœud si cela risque de faire tomber la mesure de qualité sous un seuil
 - Difficile de choisir un seuil approprié
 - Post-élagage : Supprimez les branches d'un arbre «complètement développé» - obtenez une séquence d'arbres élagués progressivement
 - Utilisez un ensemble de données différent de celui de l'apprentissage pour déterminer quel est «le meilleur arbre élagué»

Forêt aléatoire

Définition

- Forêt aléatoire (ou forêts aléatoires) est un classifieur d'agrégation composé de nombreux arbres de décision.
- Le terme vient de forêts à décision aléatoire proposées par Tin Kam Ho de Bell Labs en 1995.
- La méthode combine l'idée de « bagging » de Breiman et la sélection aléatoire de caractéristiques.

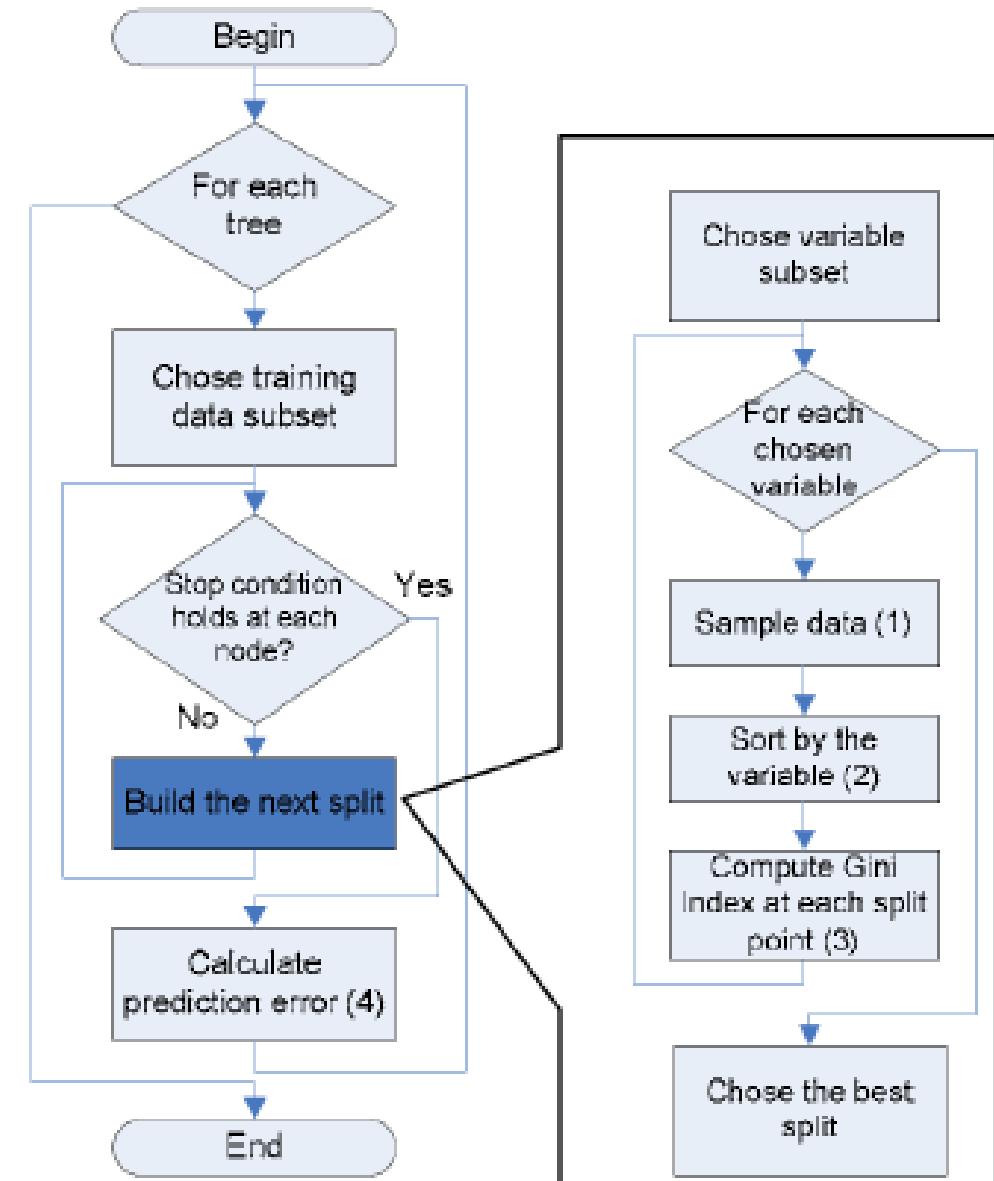
Algorithme

Chaque arbre est construit en utilisant l'algorithme suivant:

1. Soit N le nombre de cas d'apprentissage et le nombre de variables dans le classificateur est M .
2. Le nombre m de variables d'entrée à utiliser pour déterminer la décision à un nœud de l'arbre; m devrait être beaucoup moins que M .
3. Choisissez un ensemble d'entraînement pour cet arbre en choisissant n fois avec remplacement parmi tous les N cas d'entraînement disponibles (c.-à-d. Prélever un échantillon bootstrap). Utilisez le reste des cas pour estimer l'erreur de l'arbre en prédisant ses classes.
4. Pour chaque nœud de l'arbre, choisissez au hasard m variables sur lesquelles baser la décision. Calculez la meilleure répartition en fonction de ces m variables dans l'ensemble d'apprentissage.
5. Chaque arbre est entièrement développé et non élagué (comme dans la construction d'un classifieur d'arbres normal).

Pour la prédiction, un nouvel échantillon est placé dans l'arbre. On lui attribue l'étiquette de l'échantillon d'apprentissage dans le nœud terminal où elle aboutit. Cette procédure est itérée sur tous les arbres de l'ensemble et le vote moyen de tous les arbres est consigné sous forme de prédiction aléatoire de forêt.

Algorithme



Random Forest

- Les séparations sont choisies en fonction d'une mesure de pureté:
 - Par exemple. erreur au carré (régression), indice de Gini ou devinace (classification)
- Comment choisir N?
 - Construire des arbres jusqu'à ce que l'erreur ne diminue plus
- Comment choisir M?
 - Essayez de recommander les valeurs par défaut, la moitié et deux fois, et choisissez le meilleur.

Avantages

- Les avantages de la forêt aléatoire sont:
- C'est l'un des algorithmes d'apprentissage les plus précis disponibles. Pour de nombreux ensembles de données, il produit un classifieur extrêmement précis.
- Il fonctionne efficacement sur les grandes bases de données.
- Il peut gérer des milliers de variables d'entrée sans suppression de variable.
- Il donne des estimations des variables importantes dans la classification.
- Il génère une estimation non biaisée interne de l'erreur de généralisation au fur et à mesure que la construction de la forêt progresse.
- Il dispose d'une méthode efficace pour estimer les données manquantes et maintient la précision lorsqu'une grande partie des données sont manquantes.

Conclusions

- RF est rapide à construire. Encore plus rapide pour prédire
- Sur le plan pratique, le fait de ne pas nécessiter de validation croisée pour la sélection du modèle accélère considérablement l'entraînement de 10x à 100x ou plus.
- Entièrement parallélisable... pour aller encore plus vite!
- Sélection automatique de prédicteurs parmi un grand nombre de candidats
- Résistance à l'entraînement excessif
- Capacité à gérer des données sans prétraitement
- Les données n'ont pas besoin d'être rééchelonnées, transformées ou modifiées
- Résistant aux valeurs aberrantes
- Traitement automatique des valeurs manquantes

Application des différents algorithmes avec Python et Scikit-Learn dans un Notebook

**Exercice sur un modèle de régression – qui
trouvera le meilleur modèle ?**

Application des différents algorithmes avec Python et Scikit-Learn dans un Notebook

Exercice sur un modèle non supervisé

