Data science :
fondamentaux et études de cas

Machine learning avec Python et R

Éric Biernat
Michel Lutz

Préface de Yann LeCun,
Directeur de Facebook
Artificial Intelligence Research

EYROLLES

OCTO Technology
There is a better way

Data science :
fondamentaux et études de cas

Nous vivons une époque très excitante, qui ramène l'analyse de données et les méthodes quantitatives au cœur de la société. L'aboutissement de nombreux projets de recherche, la puissance de calcul informatique disponible et des données à profusion permettent aujourd'hui d'incroyables réalisations, grâce au travail des data scientists.

Un livre de référence pour les data scientists

La data science est l'art de traduire des problèmes industriels, sociaux, scientifiques, ou de toute autre nature, en problèmes de modélisation quantitative, pouvant être résolus par des algorithmes de traitement de données. Cela passe par une réflexion structurée, devant faire en sorte que se rencontrent problèmes humains, outils techniques/informatiques et méthodes statistiques/algorithmiques. Chaque projet de data science est une petite aventure, qui nécessite de partir d'un problème opérationnel souvent flou, à une réponse formelle et précise, qui aura des conséquences réelles sur le quotidien d'un nombre plus ou moins important de personnes.

Éric Biernat et Michel Lutz proposent de vous guider dans cette aventure. Ils vous feront visiter les vastes espaces de la data science moderne, de plus en plus présente dans notre société et qui fait tant parler d'elle, parfois par l'intermédiaire d'un sujet qui lui est corollaire, les big data.

Des études de cas pour devenir kaggle master

Loin des grands discours abstraits, les auteurs vous feront découvrir, claviers à la main, les pratiques de leur métier de data scientist chez OCTO Technology, l'un des leaders français du domaine. Et vous mettrez également la main à la pâte : avec juste ce qu'il faut de théorie pour comprendre ce qu'impliquent les méthodes mathématiques utilisées, mais surtout avec votre ordinateur personnel, quelques logiciels gratuits et puissants, ainsi qu'un peu de réflexion, vous allez participer activement à cette passionnante exploration !

Au sommaire

É. Biernat

Éric Biernat dirige l'activité Big Data Analytics chez OCTO Technology, l'un des leaders français sur le marché de la data science et des big data. Il a embrassé le mouvement Big Data Analytics en 2011 et ne l'a plus lâché depuis, en accompagnant ses clients qui souhaitent tirer profit des opportunités offertes par cette science. Kaggle master, Éric s'illustre régulièrement lors de compétitions de data science et intervient dans de nombreux cycles de conférences sur la thématique des big data, dans la presse spécialisée ou auprès de comités exécutifs.

M. Lutz

Suite à un parcours initial en gestion et finance, Michel Lutz s'est lancé un nouveau challenge en soutenant une thèse de doctorat en génie industriel. Durant ses années de recherche, visant à utiliser des méthodes de mathématiques appliquées dans un contexte industriel, il a développé une certaine orthodoxie statistique qui a été bien bousculée lorsqu'il a découvert le monde de la data science. Désormais, il se plonge avec enthousiasme dans les techniques de machine learning grâce à son activité de consultant chez OCTO Technology.

À qui s'adresse cet ouvrage ?

– Aux développeurs, statisticiens, étudiants et chefs de projets ayant à résoudre des problèmes de data science.

– Aux data scientists, mais aussi à toute personne curieuse d'avoir une vue d'ensemble de l'état de l'art du machine learning.

Technology
There is a better way

www.editions-eyrolles.com

Data science :
fondamentaux et études de cas

DANS LA MÊME COLLECTION

B. PHILIBERT. – **Bootstrap 3 : le framework 100 % web design.**
N°14132, 2015, 318 pages.

C. CAMIN. – **Développer avec Symfony2.**
N°14131, 2015, 474 pages.

S. PITTION, B. SIEBMAN. – **Applications mobiles avec Cordova et PhoneGap.**
N°14052, 2015, 184 pages.

H. GIRAUDEL, R. GOETTER. – **CSS 3 : pratique du design web.**
N°14023, 2015, 372 pages.

C. DELANNOY. – **Le guide complet du langage C.**
N°14012, 2014, 844 pages.

K. AYARI. – **Scripting avancé avec Windows PowerShell**.
N°13788, 2013, 358 pages.

W. BORIES, O. MIRIAL, S. PAPP. – **Déploiement et migration Windows 8**.
N°13645, 2013, 480 pages.

W. BORIES, A. LAACHIR, D. THIBLEMONT, P. LAFEIL, F.-X. VITRANT. – **Virtualisation du poste de travail Windows 7 et 8 avec Windows Server 2012**.
N°13644, 2013, 218 pages.

J.-M. DEFRANCE. – **jQuery-Ajax avec PHP**.
N°13720, 4e édition, 2013, 488 pages.

L.-G. MORAND, L. VO VAN, A. ZANCHETTA. – **Développement Windows 8 - Créer des applications pour le Windows Store**.
N°13643, 2013, 284 pages.

Y. GABORY, N. FERRARI, T. PETILLON. – **Django avancé.**
N°13415, 2013, 402 pages.

P. ROQUES. – **Modélisation de systèmes complexes avec SysML.**
N°13641, 2013, 188 pages.

SUR LE MÊME THÈME

M.-R. AMINI. – **Apprentissage machine, de la théorie à la pratique**.
N°13800, 2015, 272 pages.

M.-R. AMINI, E. GAUSSIER. – **Recherche d'information**.
N°13532, 2013, 234 pages.

A. CORNUÉJOLS, L. MICLET. – **Apprentissage artificiel**.
N°12471, 2010, 804 pages.

R. BRUCHEZ. – **Les bases de données NoSQL et le Big Data**.
N°14155, 2015, 322 pages.

Data science :
fondamentaux et études de cas

Machine learning avec Python et R

Éric Biernat
Michel Lutz

EYROLLES

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

Table des matières

Première partie

Le B.A.-ba du data scientist

1

Savoir poser un problème de data science

Introduction

Nous avons tous des problèmes… tout le temps… notre voiture ne démarre pas, notre patron nous fatigue, notre enfant ne fait pas ses nuits, etc. Hélas ! Au risque de vous décevoir, sachez que le *machine learning* ne permet pas de résoudre tous les problèmes. En revanche, il permet d'apporter des éléments de réponse à certains d'entre eux. Par exemple, aucun algorithme ne changera pour vous le démarreur de votre automobile. Néanmoins, si vous disposez de suffisamment de données concernant le fonctionnement de votre véhicule, un algorithme pourrait détecter une panne et vous suggérer de changer le démarreur. Peut-être même pourrait-il vous le suggérer avant même que le démarreur ne rende l'âme ! Mais pour cela, il est nécessaire de traduire votre problème humain en éléments qui puissent être analysés par un algorithme de *machine learning*. Autrement dit, vous devez être capable de poser votre problème sous la forme d'un problème de *data science*. Globalement, la démarche est simple : (1) il vous faut des données ; (2) vous devez savoir ce que vous voulez en faire, puis (3) comment le faire. Ce chapitre va vous donner des éléments pratiques pour répondre à ces questions. Après l'avoir lu, vous saurez mieux qualifier et rechercher des données. Vous saurez aussi poser votre problème de *data science* selon un vocabulaire précis. Enfin, vous saurez comment structurer vos données pour qu'un algorithme de *machine learning* puisse leur être appliqué.

Préliminaire : qu'est-ce que le machine learning ?

Même s'il est actuellement dopé par les nouvelles technologies et de nouveaux usages, le *machine learning* n'est pas un domaine d'étude récent. On en trouve une première définition dès 1959, due à Arthur Samuel, l'un des pionniers de l'intelligence artificielle, qui définit le *machine learning* comme le champ d'étude visant à donner la capacité à une machine d'apprendre sans être explicitement programmée. En 1997, Tom Mitchell, de l'université de Carnegie Mellon, propose une définition plus précise :

« A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E ».

Tom Mitchell illustre ensuite cette définition par quelques exemples (tableau 1-1).

Tableau 1-1. Quelques exemples de machine learning proposés par Mitchell

Cas d'application	Checkers learning	Handwriting recognition	Robot driving learning
Tasks T	Playing checkers	Recognizing and classifying handwritten words within images	Driving on public four-lane highways using vision sensors
Performance measure P	Percent of games won against opponents	Percent of words correctly classified	Average distance traveled before an error (as judged by human overseer)
Training experience E	Playing practice games againts itself	A database of handwritten words with given classifications	A sequence of images and steering commands recorded while observing a human driver

Un robot qui apprend à conduire ? Cela peut paraître un peu loin de vos préoccupations… mais illustrons brièvement ce que peut faire le *machine learning* avec un cas simple, sans doute plus proche de votre quotidien : un filtre antispam. Dans un premier temps, on peut imaginer que la « machine » (votre service de messagerie) va « analyser » la façon dont vous allez classer vos mails entrants en spam ou pas. Grâce à cette période d'« apprentissage », la machine va déduire quelques grands critères de classification. Par exemple, la probabilité que la machine classe un mail en spam va augmenter si le mail contient des termes tels qu'« argent », « rencontre facile » ou « viagra » et si l'expéditeur du mail n'est pas dans votre carnet d'adresses. A contrario, la probabilité de classement en spam va baisser si l'expéditeur est connu et que les mots du mail sont plus « classiques ».

On laissera le lecteur choisir la définition du *machine learning* avec laquelle il est à l'aise. Le tout est de bien comprendre la métaphore de l'apprentissage de la machine (à opposer à une programmation impérative de type règle « IF... THEN... ELSE... »), qui permettra à la machine d'apprendre à partir de données réelles. Avec le *machine learning*, on passe d'une informatique impérative basée sur des hypothèses à une informatique probabiliste basée sur des informations réelles. Mais pour se lancer dans cette aventure, il nous faut avant tout des données !

Au commencement était la donnée…

Un prérequis indispensable

La *data science* est une démarche empirique qui se base sur des données pour apporter une réponse à des problèmes. Donc, avant toute chose, assurez-vous d'avoir des données… Cela peut paraître idiot de le préciser, mais nous voyons souvent dans nos projets que l'accès aux données utiles à l'analyse n'est pas toujours aisé. Tantôt elles ne sont pas disponibles ou certaines manipulations techniques (jointure, etc.) ne sont pas faisables, ou encore interdites pour des raisons de confidentialité, tantôt le service informatique n'est pas capable de les extraire, tantôt elles n'existent tout simplement pas.

De plus, même si l'objectif de ce livre n'est pas de parler de *big data*, ayez en tête qu'en *machine learning*, la plupart du temps, plus il y a de données, mieux c'est ! En effet, le *machine learning* s'appuie souvent sur des algorithmes de *data-mining* connus depuis longtemps, souvent développés avant les années 2000. Hélas ! Leurs performances ont bien souvent été limitées par le manque de données disponibles ou de capacités informatiques pour traiter de larges volumes de données. C'est que certains problèmes de *data science* ne donnent parfois des résultats probants qu'à partir de plusieurs centaines de millions d'observations (on parle parfois d'analyse de signaux faibles, car non observables sur des ensembles de données réduits tels que ceux traités par les études statistiques classiques). Aujourd'hui, la hausse des capacités de stockage et la profusion de données numériques qu'elles engendrent, couplées à des moyens de calcul informatique de plus en plus puissants, font revenir ces algorithmes sur le devant de la scène. Ayez donc conscience que les gros volumes de données ne sont pas un problème pour votre travail, mais au contraire une formidable opportunité d'y trouver des informations précieuses !

Très bien, vous savez maintenant qu'il vous faut des données, peut-être même beaucoup de données… mais il se peut que ce terme soit un peu abstrait pour vous, alors clarifions-le dès à présent.

Que sont les données ?

Un bon vieux dictionnaire de statistique indique qu'une donnée est « *le résultat d'une observation faite sur une population ou sur un échantillon*[1] » (Dodge, 2007). Une donnée est donc un nombre, une caractéristique, qui m'apporte une information sur un individu, un objet ou une observation. Par exemple, 33 est un nombre sans intérêt[2], mais si quelqu'un vous dit « J'ai 33 ans », 33 devient une donnée qui vous permettra d'en savoir un peu plus sur lui.

1. La remarque du statisticien : dans ce livre, on ne s'intéressera pas aux problématiques d'échantillonnage propres à la statistique et étudiant les possibilités de tirer des conclusions au sujet d'une population en en analysant un sous-ensemble. En *machine learning*, on recherche des informations utiles dans un ensemble de données, sans se poser de questions sur ce qu'elles reflètent d'une population plus vaste (point de vue *data-mining*).

2. Pour le commun des mortels, bien sûr. Cédric Villani dira sans doute que c'est un nombre très intéressant, car c'est le plus grand entier naturel qui ne peut pas être exprimé comme une somme de nombres triangulaires différents, ou que c'est le deuxième nombre uniforme de la classe U3 et que c'est aussi un nombre semi-premier, mais nous nous écartons là du propos de ce livre !

Généralement, on lie les données à des variables parce que le nombre/la caractéristique varie si on observe plusieurs objets/individus/observations. En effet, si on s'intéresse à l'âge de tous les lecteurs de ce livre, on sait qu'il est défini par un nombre compris entre, disons, 15 et 90, et qu'il variera d'un lecteur à l'autre. Ainsi, si l'on nomme $X_{âge}$ la variable « âge du lectorat », les données mesurant cet âge sont égales à $x_{1âge}$, $x_{2âge}$, …, $x_{mâge}$, où $x_{1âge}$ est l'âge du lecteur 1, $x_{2âge}$ l'âge du lecteur 2, et ainsi de suite jusqu'à $x_{mâge}$, où m représente le nombre total de lecteurs.

Tel est le matériau brut que va manipuler le *data scientist* : des variables exprimées concrètement par des données et qui lui permettent de décrire un ensemble d'objets/individus/observations. Ces données peuvent prendre diverses formes que nous allons désormais détailler.

Les principaux types de données

On distingue généralement les données quantitatives des données qualitatives.

Les données quantitatives sont des valeurs qui décrivent une quantité mesurable, sous la forme de nombres sur lesquels on peut faire des calculs (moyenne, etc.) et des comparaisons (égalité/différence, infériorité/supériorité, etc.). Elles répondent typiquement à des questions du type « combien ». On fait parfois la différence entre :

- les données quantitatives continues, qui peuvent prendre n'importe quelle valeur dans un ensemble de valeurs : la température, le PIB, le taux de chômage, en sont des exemples ;
- et les données quantitatives discrètes, qui ne peuvent prendre qu'un nombre limité de valeurs dans un ensemble de valeurs : le nombre d'enfants par famille, le nombre de pièces d'un logement, etc.

Les données qualitatives décrivent quant à elles des qualités ou des caractéristiques. Elles répondent à des questions de la forme « quel type » ou « quelle catégorie ». Ces valeurs ne sont plus des nombres, mais un ensemble de modalités. On ne peut pas faire de calcul sur ces valeurs, même dans l'éventualité où elles prendraient l'apparence d'une série numérique. Elles peuvent toutefois être comparées entre elles et éventuellement triées. On distingue :

- les données qualitatives nominales (ou catégorielles), dont les modalités ne peuvent être ordonnées. Par exemple : la couleur des yeux (bleu, vert, marron, etc.), le sexe (homme, femme), la région d'appartenance (68, 38, etc.) ;
- et les données qualitatives ordinales, dont les modalités sont ordonnées selon un ordre « logique ». Par exemple : les tailles de vêtements (S, M, L, XL), le degré d'accord à un test d'opinion (fortement d'accord, d'accord, pas d'accord, fortement pas d'accord).

Le tableau 1-2 résume ces différents types de données ainsi que les opérations qu'ils supportent.

Tableau 1-2. Les opérations supportées par chaque type de données

Type de données	Opérations supportées
Quantitatives continues	Calculs, égalité/différence, infériorité/supériorité
Quantitatives discrètes	Calculs, égalité/différence, infériorité/supériorité
Qualitatives nominales	Égalité/différence
Qualitatives ordinales	Égalité/différence, infériorité/supériorité

Maintenant que vous savez la différence entre les diverses données, vous vous demandez peut-être où les chercher… Bonne question en effet ! Voici quelques pistes qui pourront vous aider.

D'où viennent les données ?

La réponse à cette question n'est pas bien difficile : elles viennent de partout ! C'est d'ailleurs bien pour cela qu'on observe de nos jours un tel engouement pour la *data science*, le *machine learning* et l'analyse de données en général. Précisons tout de même un peu. En premier lieu, distinguons les données dites privées des données publiques.

Les données privées sont tout simplement les données qui en théorie n'appartiennent qu'à vous ou à votre organisation. Si vous travaillez en entreprise, ce pourrait être les bases de données internes de votre société, les divers documents électroniques ou numérisés disponibles (e-mails, fichiers informatiques, documents scannés, images et vidéos, etc.). Ce pourrait être aussi les fichiers de *logs*[3] générés par vos machines (serveurs informatiques, équipements de production, etc.) et par vos applications informatiques (progiciels, applications web, etc.), les informations remontées par les capteurs et objets connectés, et bien d'autres. Il y a déjà beaucoup de connaissances à en tirer.

Toutefois, les plus ambitieux d'entre vous pourraient aussi avoir envie d'aller plus loin en exploitant des données publiques, c'est-à-dire accessibles à tous. Dans ce cas, vous disposez d'une source de données quasi infinie : Internet. Pour cela, trois modes de collecte de données existent.

- Les *open data*, qui correspondent à la mise à disposition gratuite de données de la société civile, sur des sites tels que www.data.gov, www.data.gouv.fr, http://opendata.alsace.fr, etc.
- Les *open* API (*Application Programming Interface*), qui sont des technologies permettant d'accéder à des données sur Internet. Elles vous permettent de récupérer par exemple des données mises à disposition par Google, Twitter, etc. Pour en savoir plus sur les API disponibles, consultez par exemple l'annuaire http://www.programmableweb.com.
- Et bien sûr, le Web en tant que tel est lui aussi directement source de données. Pour cela, il faut un minimum d'expertise en programmation pour être capable de faire ce que l'on nomme du web *scraping*, qui consiste à récupérer des données directement à partir des pages des sites Internet.

Nous avons également parlé d'images, de vidéos. Vous vous demandez peut-être ce que cela a à voir avec les types de données qui vous ont été présentés précédemment. En fait, lorsqu'on traite informatiquement des objets de types images et vidéos, on leur applique des traitements visant à les réduire à une suite de nombres interprétables par un algorithme de *machine learning*. Vous en verrez des exemples dans la suite de ce livre. Ces objets, assez complexes à manipuler, sont dits non structurés. À l'opposé, les données les plus faciles à traiter sont celles qui proviennent directement des bases de données : indexées[4], prêtes à être traitées, elles sont dites structurées. Sachez qu'il existe un niveau de structuration intermédiaire, dit semi-structuré. Celui-ci correspond à divers formats de fichiers informatiques simples à traiter, mais qui ne bénéficient pas de l'indexation

3. Un fichier de *log* est un fichier enregistrant tous les événements recensés par une machine ou une application.

4. L'index est la structure qu'un système de gestion de bases de données informatique impose à ses données. Il permet de les retrouver rapidement et de simplifier toutes les opérations qu'on peut leur appliquer (recherche, tri, jointure, agrégation).

propre aux données extraites des bases de données informatiques. Le tableau 1-3 résume ces différents niveaux de structuration.

Tableau 1-3. Les différents niveaux de structuration des données

Niveau de structuration	Modèle de données	Exemples	Facilité de traitement
Structuré	Système de données relationnel objet/colonne	Base de données d'entreprise…	Facile (indexé)
Semi-structuré	XML, JSON, CSV, logs	API Google, API Twitter, web, logs…	Facile (non indexé)
Non structuré	Texte, image, vidéo	web, e-mails, documents…	Complexe

Voilà, vous savez identifier la matière première du *data scientist* et vous savez où la chercher. À présent, vous brûlez sans doute d'impatience de pouvoir l'exploiter grâce aux nombreux algorithmes conçus pour cela. Mais avant de les passer en revue, laissez-nous vous donner une idée générale de ce que savent faire tous ces algorithmes.

Les algorithmes : pour faire quoi ?

Sous les données, des liens… plus ou moins certains !

Quel que soit l'algorithme qu'il utilise, le *data scientist* n'a qu'une seule idée en tête : découvrir des liens dans ses données (on parle souvent de *pattern*). Dans le cadre de l'emploi de méthodes de *machine learning*, on suppose donc qu'il existe un lien au sein des données et que nos algorithmes vont nous aider à le trouver.

Attention, il s'agit d'une hypothèse forte : par défaut, il n'y a généralement pas de lien dans les données. Cela peut paraître surprenant, car nous avons plutôt tendance à en voir partout ! Ce phénomène bien connu en recherche s'appelle le biais de publication (Cucherat *et al*., 1997). En science, il désigne le fait que les chercheurs et les revues scientifiques ont plus tendance à publier des expériences ayant obtenu un résultat positif (statistiquement significatif) que des expériences ayant obtenu un résultat négatif. Ce biais de publication donne aux lecteurs une perception biaisée (vers le positif) de l'état de la recherche.

En plus du biais de publication, un ensemble de phénomènes bien connus en statistiques peuvent aussi amener à conclure artificiellement à des liens dans les données.

- La corrélation fallacieuse, pour désigner des séries de données corrélées entre elles alors qu'elles n'ont a priori aucune raison de l'être. Deux explications peuvent être apportées à ce type de corrélation :
  - soit les séries n'ont aucun lien entre elles, mais sont toutes deux corrélées à une troisième variable cachée (Pearl, 1998) ;
  - soit de grands jeux de données génèrent naturellement des corrélations, sans aucune relation de causalité : si l'on considère un grand nombre de variables totalement indépendantes, on observera toujours un petit nombre de corrélations significatives (Granville, 2013) !

- La corrélation fallacieuse au sens de K. Pearson, utilisée dans le contexte très spécifique des données de composition. Dans un article de 1897, appliquée à la mesure d'organes humains, il montra que deux variables indépendantes *X* et *Y* peuvent sembler corrélées si rapportées à une troisième variable *Z*. La taille du fémur est indépendante de la taille du tibia, mais le ratio taille du fémur/taille de l'humérus est corrélé au ratio taille du tibia/taille de l'humérus ! L'analyse des données de compositions est un champ assez spécifique des statistiques, c'est pourquoi c'est la dernière fois que nous en parlerons dans ce livre.
- La régression fallacieuse, dans le cadre de l'estimation d'une régression linéaire simple ou multiple. On peut observer des modèles qui semblent très explicatifs, mais faux du fait de l'influence du temps sur le modèle (Granger et Newbold, 1974).
- Le paradoxe de Bertrand, qui nous montre que les résultats issus de nos analyses peuvent être influencés par les méthodes et les mécanismes qui produisent les variables (Bertrand, 2010).
- Le mauvais usage de la *p-value,* indicateur souvent utilisé sans précaution par certains statisticiens pour évaluer une significativité statistique et qui aboutit très facilement à rejeter une hypothèse nulle de façon totalement artificielle lorsque le nombre d'observations est grand (Lin *et al.*, 2013).

Les précautions d'usage ayant été exposées, nous supposerons un lien dans les données dans la suite de cet ouvrage, complexe peut-être, mais bel et bien présent. D'ailleurs, ces écueils sont plutôt la conséquence de mauvaises applications des méthodes statistiques classiques dans des contextes inappropriés, notamment de larges volumes de données. Néanmoins, voyez-y une invitation à la prudence : on gagne toujours à réfléchir à deux fois aux résultats de ses analyses et à remettre en question ses conclusions ! De toute manière, une approche *machine learning* est toujours une prise de position probabiliste qui implique de renoncer à une vision déterministe et figée du monde. Elle va inférer des règles, avec des marges d'incertitude et potentiellement changeantes.

Ces considérations métaphysiques étant faites, parlons plus concrètement de la façon dont les liens entre les données peuvent être découverts.

Une taxinomie des algorithmes

Les algorithmes ne sont pas tous destinés aux mêmes usages. On les classe usuellement selon deux composantes[5] :

- le mode d'apprentissage : on distingue les algorithmes supervisés des algorithmes non supervisés ;
- le type de problème à traiter : on distingue les algorithmes de régression de ceux de classification.

Tous les algorithmes qui seront présentés dans ce livre sont définis selon ces deux axes, comme l'indique le tableau 1-4.

5. Comme toute classification, celle-ci est nécessairement imparfaite : on trouvera toujours des algorithmes qui appartiennent à plusieurs classes, ou d'autres qui n'y trouvent pas leur place. Néanmoins, elle a l'avantage de donner une vision unifiée et simplifiée de la multitude d'algorithmes existants et c'est pourquoi nous l'utiliserons dans ce livre.

Tableau 1-4. Taxinomie des algorithmes présentés dans ce livre

Algorithme	Mode d'apprentissage	Type de problème à traiter
Régression linéaire univariée	Supervisé	Régression
Régression linéaire multivariée	Supervisé	Régression
Régression polynomiale	Supervisé	Régression
Régression régularisée	Supervisé	Régression
Naive Bayes	Supervisé	Classification
Régression logistique	Supervisé	Classification
Clustering hiérarchique	Non supervisé	-
Clustering non hiérarchique	Non supervisé	-
Arbres de décision	Supervisé	Régression ou classification
Random forest	Supervisé	Régression ou classification
Gradient boosting	Supervisé	Régression ou classification
Support Vector Machine	Supervisé	Régression ou classification
Analyse en composantes principales	Non supervisé	-

Donnons un peu plus de détails sur la signification de cette taxinomie.

Algorithmes supervisés et non supervisés

La différence entre algorithmes supervisés et non supervisés est fondamentale. Les algorithmes supervisés extraient de la connaissance à partir d'un ensemble de données contenant des couples entrée-sortie. Ces couples sont déjà « connus », dans le sens où les sorties sont définies a priori. La valeur de sortie peut être une indication fournie par un expert : par exemple, des valeurs de vérité de type OUI/NON ou MALADE/SAIN. Ces algorithmes cherchent à définir une représentation compacte des associations entrée-sortie, par l'intermédiaire d'une fonction de prédiction.

A contrario, les algorithmes non supervisés n'intègrent pas la notion d'entrée-sortie. Toutes les données sont équivalentes (on pourrait dire qu'il n'y a que des entrées). Dans ce cas, les algorithmes cherchent à organiser les données en groupes[6]. Chaque groupe doit comprendre des données similaires et les données différentes doivent se retrouver dans des groupes distincts. Dans ce cas, l'apprentissage ne se fait plus à partir d'une indication qui peut être préalablement fournie par un expert, mais uniquement à partir des fluctuations observables dans les données.

Le petit exemple qui suit illustre les principes de ces deux familles d'algorithmes. Imaginons un ensemble d'individus décrits par deux variables d'entrée, X_1 et X_2. Dans le cas d'un apprentissage supervisé, il faudra leur adjoindre une variable de sortie Y, qui pourra par exemple prendre deux valeurs {O, X}. L'algorithme proposera alors une fonction de prédiction de la forme

6. On distingue plusieurs familles d'algorithmes non supervisés, dont les grands classiques sont : 1) la détection d'anomalies, qui correspond à un type d'analyse assez particulier que nous avons choisi de ne pas aborder dans cet ouvrage ; 2) la réduction de dimension, abordée dans ce livre au chapitre sur l'analyse en composantes principales ; 3) le *clustering*, sans doute la famille d'algorithmes non supervisés la plus répandue.

$Y = f(X_1, X_2)$. Dans le cas de l'apprentissage non supervisé, plus de Y : l'algorithme va trouver tout seul, comme un grand, deux groupes d'individus distincts, juste à partir des positions dans le plan défini par X_1 et X_2, et ce sans aucune autre indication. La figure 1-1 illustre ces deux formes d'apprentissage.

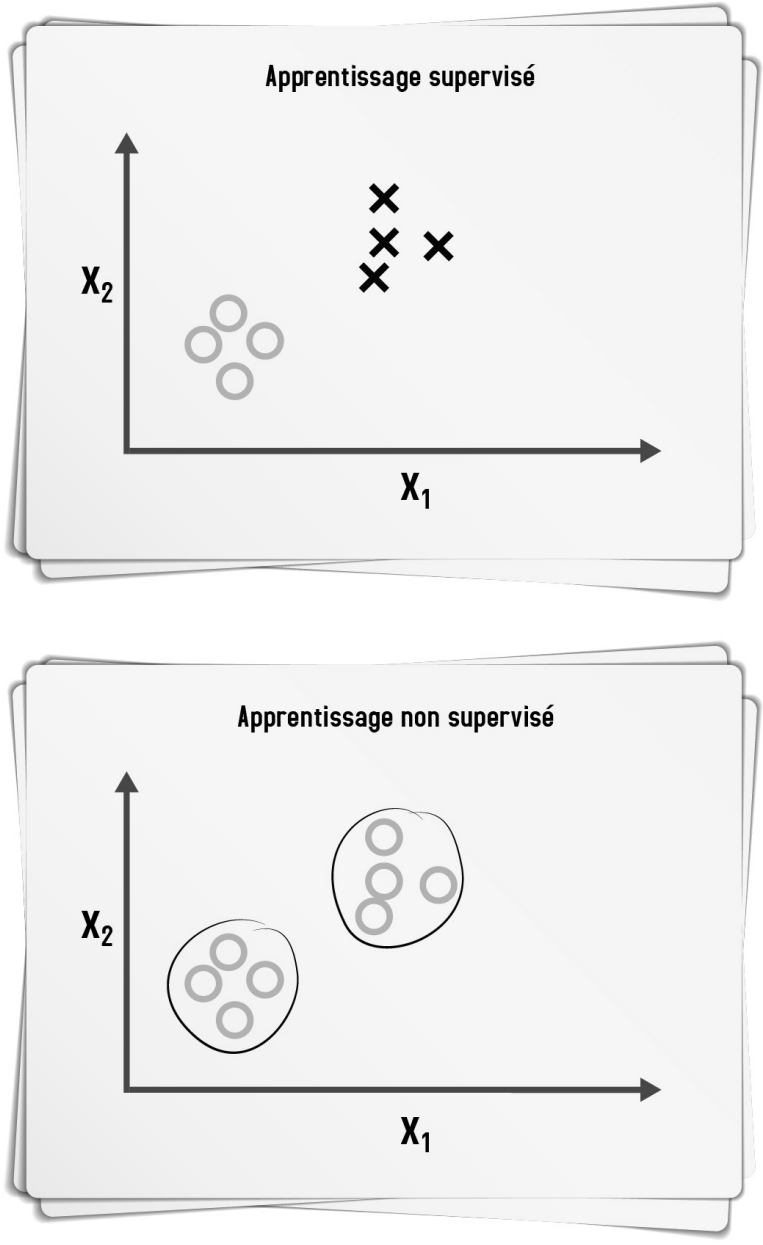

Figure 1-1 – *Apprentissages supervisé et non supervisé*

Généralement, les algorithmes supervisés sont plus performants, mais pour les utiliser, il faut être capable de spécifier une valeur de sortie, et cette information d'expert n'est pas toujours disponible.

Insistons sur ce point. À l'ère des *big data*, la donnée ne coûte pas cher. Vous avez sans doute, à force de le lire partout dans la presse ou en écoutant les consultants passer dans votre bureau, retenu les fameux « 3V » des *big data* (Volume – Variété – Vélocité), faible descripteur de ce que sont vraiment les *big data*. Alors oui, bien évidemment, le volume de données disponibles explose. Mais ces données ont-elles une valeur métier pour vous, êtes-vous capable de les valoriser ? C'est moins sûr. Nous y reviendrons plus tard, mais songez que pour obtenir un résultat probant pour un problème relativement simple de classification binaire équilibré (autant de 0 que de 1), il faudra déjà quelques dizaines de milliers de lignes. Chacune de ces lignes devra être parfaitement labélisée d'un point de vue métier. Disposez-vous des moyens humains pour constituer vos bases de données supervisées ? C'est une question non négligeable. Rassurez-vous toutefois avec cette information : une grande partie des efforts de la sphère académique se concentre aujourd'hui sur les algorithmes non supervisés, qui cherchent précisément à vous épargner de douloureuses et ennuyeuses journées de travail à labéliser à la main plusieurs millions de lignes.

Pour finir, notez qu'il existe une catégorie bien moins courante d'algorithmes hybrides basés sur une approche dite semi-supervisée. C'est une approche intermédiaire qui se base à la fois sur des observations de type entrée-sortie et sur des observations sans variable de sortie, mais elle ne sera pas développée dans ce livre.

Algorithmes de régression et de classification

La distinction régression/classification se fait au sujet des algorithmes supervisés. Elle distingue deux types de valeurs de sorties qu'on peut chercher à traiter. Dans le cadre d'un problème de régression, Y peut prendre une infinité de valeurs dans l'ensemble continu des réels (noté $Y \in \mathbb{R}$). Ce peut être des températures, des tailles, des PIB, des taux de chômage, ou tout autre type de mesure n'ayant pas de valeurs finies a priori.

Dans le cadre d'un problème de classification, Y prend un nombre fini k de valeurs ($Y = \{1, \ldots, k\}$). On parle alors d'étiquettes attribuées aux valeurs d'entrée. C'est le cas des valeurs de vérité de type OUI/NON ou MALADE/SAIN évoqués précédemment.

Voici à nouveau un petit exemple illustratif, à partir de la figure 1-2.

- L'image du haut répond au problème suivant : quel est le prix d'une maison en fonction de sa taille ? Ce prix peut prendre une infinité de valeurs dans $\mathbb{R}$, c'est un problème de régression.
- L'image du bas s'intéresse quant à elle à un autre problème : selon sa taille, une tumeur est-elle dangereuse ou bénigne ? Ici, on va chercher à classer les observations en fonction de valeurs de réponse possibles en nombre limité : OUI, NON (éventuellement PEUT-ÊTRE). C'est un problème de classification.

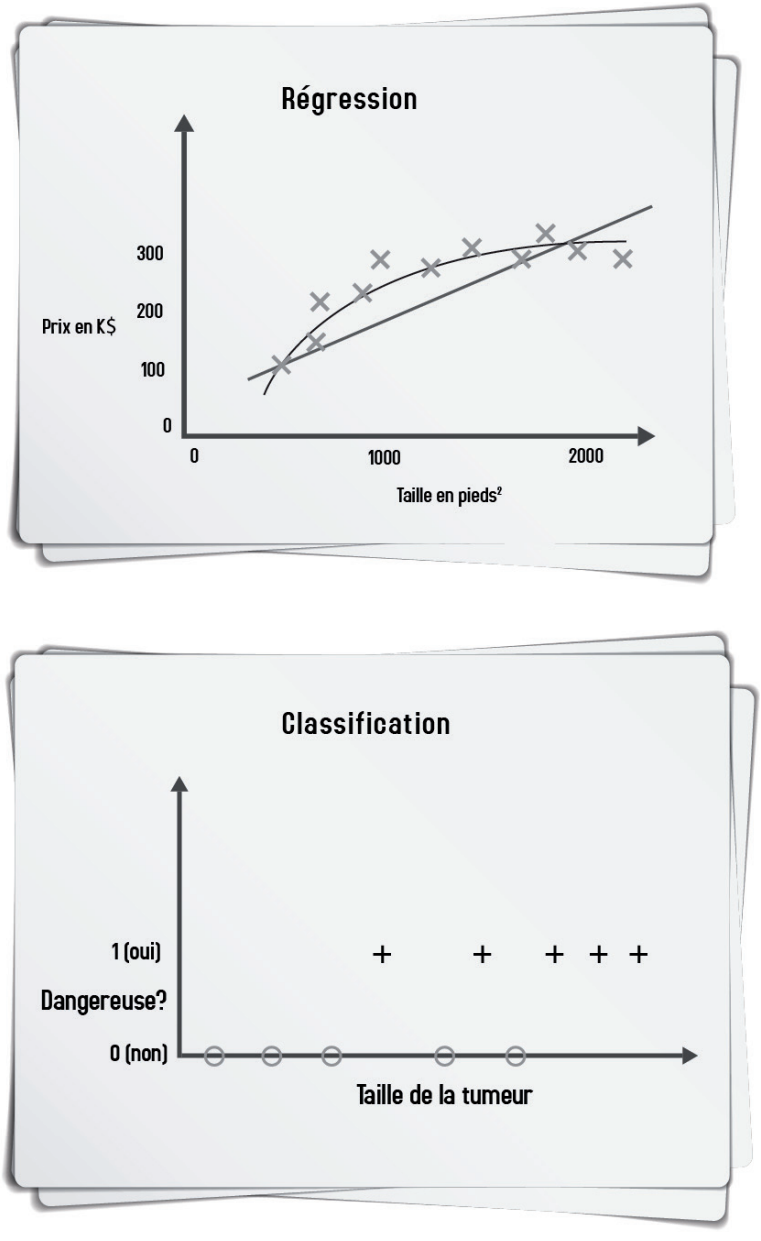

Figure 1-2 – *Algorithmes de régression et algorithmes de classification*

Pour les plus curieux

Les types d'algorithmes présentés dans cet ouvrage correspondent à deux paradigmes spécifiques du *machine learning* : l'induction (apprentissage supervisé) et le *clustering* (apprentissage non supervisé). D'autres paradigmes plus spécialisés existent : apprentissage par analogie, par renforcement, algorithmes génétiques, etc. Dans son cours d'introduction à l'intelligence artificielle, Chakraboty (2010) donne plus de détails sur ces divers paradigmes.

À titre indicatif et pour aiguiser votre curiosité, parlons brièvement de l'apprentissage par renforcement, qui semble promu à un bel avenir. Les algorithmes de cette famille consistent à apprendre un comportement face à diverses situations au cours du temps en optimisant une « récompense » quantitative. Ces algorithmes correspondent à un agent intelligent qui apprend au fil des expériences, en essayant de maximiser sa récompense. Pour avoir une idée du potentiel de l'approche, consultez le lien suivant (http://sarvagyavaish.github.io/FlappyBirdRL), qui explique comment un tel agent intelligent peut apprendre à jouer à *Flappy Bird* ! Sur cette thématique, vous pouvez aussi consulter le récent article de Mnih *et al*. (2015), rédigé par une équipe de DeepMind, société de Google spécialisée dans l'intelligence artificielle.

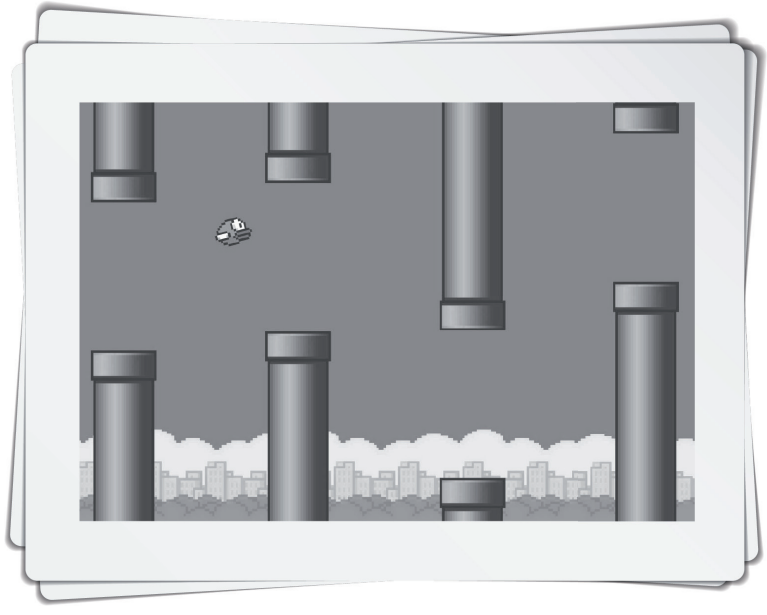

Figure 1-3 – *Le machine learning peut même vous aider à améliorer vos scores à Flappy Bird !*

Eh bien voilà ! Nous y sommes presque ! Vous savez qualifier les données utiles à votre étude et vous savez ce que vous pouvez et voulez en faire. Avant d'appliquer l'algorithme ultime qui répondra à tous vos problèmes, il vous reste à structurer vos données de façon à ce qu'elles puissent être digérées par un algorithme de *machine learning*.

Algorithmes et structures de données

Représentation matricielle des données

Nous avons vu que dans le monde du *machine learning*, tout objet, individu, observation, est décrit par un ensemble de variables X_1, X_2, …, X_j, j allant de 1 à n. Évidemment, tout l'intérêt du *machine learning* sera de trouver des régularités dans les données grâce à l'observation d'un grand nombre i d'individus[7], allant de 1 à m. La valeur de la variable X_1 de l'individu 1 se note x_{11}. Le cas général se note ainsi : x_{ij}, c'est-à-dire la valeur de la variable X_j de l'individu x_i.

Ces n variables décrivant m individus sont représentés dans ce qu'on appelle une matrice X de dimensions (m,n). On la représente comme suit :

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}$$

Chaque colonne correspond à une variable et chaque ligne correspond à un individu. Pour être encore plus clair, cette matrice est directement équivalente à la manière dont vous pourriez organiser vos données dans un tableur (figure 1-4).

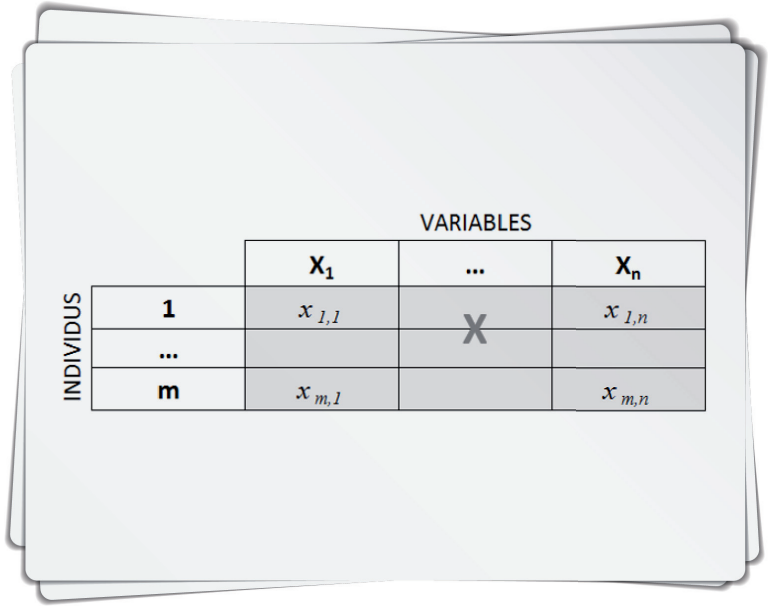

Figure 1-4 – *Une matrice, c'est un tableau, tout simplement !*

7. Ou objets, etc. : le lecteur aura compris qu'on se passera de cette énumération systématique pour faciliter la lecture !

Si le tableau n'a qu'une colonne (soit une seule variable) on ne parle plus de matrice, mais de vecteur. Le *machine learning* a pour seul objectif la manipulation de vecteurs et de matrices pour en dégager des représentations synthétiques des données observées.

Que font les algorithmes ?

Quel que soit l'algorithme utilisé, le but sera toujours le même. Les algorithmes non supervisés vont chercher à produire des représentations compactes des données, en regroupant les individus similaires compte tenu des valeurs de la matrice *X*. Pour cela, la matrice *X* est suffisante. Les algorithmes ont simplement besoin de moyens de mesurer les proximités entre les individus. Nous reparlerons de cela en détail lors de la présentation des algorithmes.

Pour les algorithmes supervisés, c'est un peu différent. *X* va représenter les valeurs d'entrée (on parle aussi d'attributs ou de *features*). Il faudra leur adjoindre, pour chaque individu, un vecteur *Y* représentant les valeurs de sortie ; il sera donc de dimension *(m,1)*. Le but du jeu est de décrire une relation liant *X* à *Y*. Pour cela, nous avons besoin d'un deuxième vecteur, nommé Θ (dans le cas d'un modèle linéaire, mais nous préciserons tout cela dans les chapitres à venir). Les valeurs de Θ, qu'on nomme paramètres ou poids du modèle, sont associées à celles de *X*, pour expliquer *Y*. Il faut ainsi une valeur de Θ pour chaque variable, Θ sera donc de dimension *(n,1)*. Grossièrement, cela revient à écrire :

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} \begin{pmatrix} \Theta_1 \\ \vdots \\ \Theta_n \end{pmatrix}$$

ou, sous une forme condensée : $Y=X\Theta$. Tout l'art du *machine learning* sera de trouver les bonnes valeurs de Θ, qui ne sont pas données mais inférées à partir des données. Lorsqu'on recherche ces valeurs Θ, on parle d'entraînement ou d'apprentissage du modèle. Pour un problème donné, différents algorithmes donneront des valeurs de Θ plus ou moins efficaces pour modéliser les données.

Mesurer des distances, apprendre les paramètres d'un modèle, voilà tout ce que le *data scientist* cherche à faire. Mais il existe de multiples façons de le faire. Nous allons vous les présenter dans la partie suivante, par une petite visite guidée des principaux algorithmes rencontrés dans le monde du *machine learning*.

Avant d'entrer dans le vif du sujet, nous aborderons un dernier point préliminaire dans le chapitre suivant : la question des outils informatiques. En effet, l'ordinateur est le meilleur allié du *data scientist* pour manipuler les données, avoir accès à des codes d'algorithmes prêts à l'emploi, etc.

À RETENIR Savoir poser un problème de data science

L'objectif du *machine learning* est d'apprendre à partir de données issues d'observations réelles.

Les données peuvent être de diverses natures et provenir de différentes sources. Selon le cas, elles sont plus ou moins complexes à manipuler.

Les algorithmes visent à en extraire des régularités qui permettront l'apprentissage. On distingue :

- les algorithmes supervisés (quand il existe une valeur à prédire) des algorithmes non supervisés (quand il n'y a pas de valeur à prédire) ;
- les algorithmes de régression (quand la valeur à prédire est continue) des algorithmes de classification (quand la valeur à prédire est catégorielle).

Pour utiliser ces algorithmes, les données doivent être mises en forme sous une représentation matricielle.

Références

Les références suivantes ont été utilisées pour proposer une définition du *machine learning* :

- Samuel AL. 1959. Some studies in machine learning using the game of checkers. *IBM Journal*, 3:3, p. 210-229.
- Mitchell TM. 1997. *Machine learning*. McGraw Hill.

Concernant les différents types de variables, rien de bien nouveau, ces distinctions sont présentées dans n'importe quel livre d'introduction à la statistique. Pour vous rafraîchir la mémoire :

- Pagès J. 2005. *Statistiques générales pour utilisateurs. Vol. 1 – Méthodologie*. Presses Universitaires de Rennes.
- Dodge Y. 2007. *Statistique – Dictionnaire encyclopédique* . Springer.

Pour toutes les personnes effrayées par les risques de liens artificiels dans les données, voici les références introductives mentionnées :

- Cucherat M., Boissel JP., Leizorovicz A. 1997. *Manuel pratique de méta-analyse des essais thérapeutiques.*
- Pearl J. 1998. Why there is no statistical test for confounding, why many think there is, and why they are almost right. *University of California – Cognitive Systems Laboratory Technical report.*
- Granville V. 2013. http://www.analyticbridge.com/profiles/blogs/the-curse-of-big-data.
- Pearson K. 1897. Mathematical Contributions to the Theory of Evolution – On a Form of Spurious Correlation Which May Arise When Indices Are Used in the Measurement of Organs. *Proceedings of the Royal Society of London*, 187.
- Granger CWJ. et Newbold P. 1974. Spurious Regression in Econometrics. *Journal of Econometrics*, 2, p. 111-120.
- Bertrand J. 2010. *Calcul des probabilités*. Nabu Press.
- Lin M., Lucas HC., Shmueli G. 2013. Too big to fail: large samples and the p-value problem. *Information Systems Research*, 24:4, p. 906-917.

Les algorithmes supervisés/non supervisés et de régression/de classification vont être détaillés en long et en large dans la partie qui suit, nous n'en dirons donc pas plus ici. Pour des informations plus générales concernant l'intelligence artificielle, voici un lien vers le cours introductif de Chakraboty mentionné dans ce chapitre :

- Chakraboty RC. 2010. http://www.myreaders.info/html/artificial_intelligence.html.

Et l'article de DeepMind au sujet de l'apprentissage par renforcement :

- Mnih V., Ka K., Silver D., Rusu DA., Veness J., Bellemare MG., Graves A., Riedmiller M., Fidjeland AK., Ostrovski G., Petersen S., Beattie C., Sadik A., Antonoglou A., King H., Kumaran D., Wierstra D., Legg S., Hassabis D. 2015. Human-level control through deep reinforcement learning, *Nature*, 518, p. 523-533.

Nous avons vu que la manipulation de matrices est au cœur du *machine learning*. Même si ce n'est pas indispensable à la compréhension de ce livre, ceux qui aimeraient se rafraîchir la mémoire peuvent lire le petit ouvrage suivant :

- Bouteloup J. *Calcul matriciel élémentaire*. PUF, Que sais-je ?