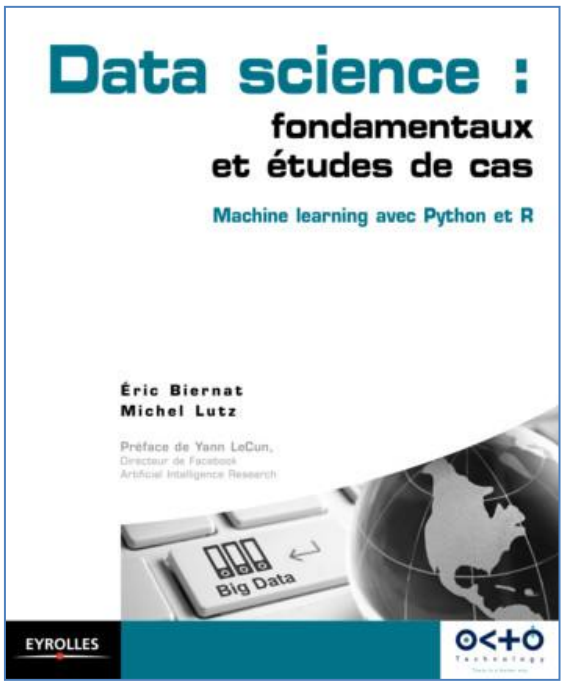
Data Science : fondamentaux et études de cas - Machine learning avec Python et R

Eric Biernat, Michel Lutz

Eyrolles, 2015.



« Data Science : fondamentaux et études de cas » surfe sur la vague du Data Science, très en vogue aujourd'hui, comme nous le montre Google Trends. L'ouvrage regroupe sur son seul titre toute une série de mots clés qui interpellent les personnes qui gravitent dans le domaine : machine learning[1] ; R et Python, deux logiciels[2] et langages de programmation[3] très populaires auprès des data scientists ; et même Big Data que l'on aperçoit sur la couverture.

Dans la quatrième de couverture, les auteurs définissent le Data Science comme « 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 ». Elle traduit bien l'idée de processus inhérent à l'exploration

[1] Après un petit creux dans les années 2010, le terme revient en force, toujours dixit Google Trends.
[2] KDnuggets, mai 2015, « Analytics, Data Mining, Data Science software/tools used in the past 12 months ».
[3] KDnuggets, Juillet 2015, « Primary programming language for Analytics, Data Mining, Data Science tasks : R, Python, or Other ».

statistique des données : définir des objectifs d'analyse, accéder aux données permettant d'y apporter des réponses, mettre en œuvre la technique statistique (ou data mining, ou machine learning, qu'importe le nom qu'on lui donne) pour obtenir un résultat compréhensible et opérationnel. L'activité nécessite des compétences fortes en informatique, notamment en programmation car les traitements sont rarement triviaux.

L'ouvrage se décompose en **3 grandes parties**. La **première** (chapitres 1 et 2) introduit la notion de data science et décrit l'environnement logiciel correspondant. La **seconde** (chapitres 3 à 13) énumère les principaux algorithmes de modélisation statistique (statistique exploratoire, machine learning, apprentissage automatique, data mining, pour moi ces termes sont équivalents dans le cadre qui nous intéresse) avec, une première sous-partie consacrée aux approches « classiques », et une seconde dédiée aux méthodes « modernes », performantes. La **troisième** partie aborde la pratique du data science, avec des études de cas, des descriptions de « challenges » (compétitions qui connaissent un très fort engouement actuellement) auxquels les auteurs ont participé, et aussi une sous-partie traitant des données temporelles.

Le **chapitre 1** cherche à délimiter le champ du data science[4]. Ce n'est pas chose facile car tout le monde cherche à se l'approprier. Les définitions pullulent, tout comme les documents qui nous tombent dessus à la moindre requête Google. Que l'on prenne le temps d'un chapitre pour essayer de cerner le domaine ne peut être qu'une bonne chose.

Les auteurs s'appuient sur le prisme du machine learning c.-à-d. les méthodes de traitement exploratoire des données. Ils les classifient en deux grands groupes : les approches supervisées, dans un schéma prédictif qui peut se référer à la régression ou le classement ; et les approches non supervisées où l'objectif est de constituer des groupes « similaires » d'objets ou de variables.

Les données sont le carburant des méthodes de machine learning. L'ouvrage décrit les sources potentielles, les différents types, et leur structuration. Ce dernier point est certainement celui qui nous interpelle le plus. Par rapport aux ouvrages de statistiques parus il y a une vingtaine d'années, le traitement des données semi-structurées ou non-structurées (texte, image, vidéo, etc.) est maintenant rentré dans la pratique courante. On voit très mal

[4] Le chapitre est accessible en ligne sur le site des éditions Eyrolles.

aujourd'hui une formation en statistique et /ou en data mining ne pas inclure des enseignements sur le traitement des données textuelles par exemple.

Le chapitre est conclu par un encadré « A retenir » résumant les notions importantes évoquées, et une bibliographie commentée. Les deux sont présents dans tous les chapitres. Voilà une démarche qui gagnerait à être généralisée dans les ouvrages à vocation scientifique.

Le **chapitre 2** nous parle des outils informatiques. Il est succinct (5 pages), sachant de toute manière que les auteurs ont pris le parti de faire la part belle à Python, beaucoup, et R, vraiment très peu (quelques lignes de code essentiellement concentrées dans les chapitres traitant des séries temporelles). A la sortie, nous retenons que le choix de l'outil dépend des contraintes de volumétrie et du type de traitement à mettre en place (batch ou temps réel).

Le **chapitre 3** entame la partie dévolue à la description des méthodes d'apprentissage automatique (terme français pour désigner ''machine learning''). Il aborde la régression linéaire univariée ou régression linéaire simple. Il est assez elliptique (6 pages). Le lecteur a une idée globale de la technique mais il n'est pas vraiment possible de comprendre dans le détail la teneur de l'algorithme. L'absence d'exemples de traitements sur données réalistes ne fournit pas non plus les codes de lectures des résultats que pourraient nous proposer les modules de Python ou de R. Pour ma part, je retiendrais avant tout l'idée d'un apprentissage de coefficients d'une régression simple via une descente du gradient. Cette présentation n'est pas vraiment usuelle. Cela montre que l'on peut aborder les problèmes de statistique avec différents prismes. La démarche m'est apparue très intéressante.

Les chapitres suivants sont de la même teneur : régression linéaire multiple (**chapitre 4**, 10 pages) ; régression polynomiale (**chapitre 5**, 9 pages) ; régression régularisée (**chapitre 6**, 6 pages) ; naive bayes (**chapitre 7**, 5 pages) ; régression logistique (**chapitre 8**, 12 pages) ; le clustering (**chapitre 9**, 10 pages) ; les arbres de décision (**chapitre 10**, 4 pages). Dans tous les cas, nous percevons à peu près les fondements des méthodes et leurs finalités. En revanche, les quelques formules mathématiques qui égayent le texte ne permettent pas vraiment d'en apprécier le mécanisme. Et l'absence d'exemples traités rend le tout assez abstrait.

Les auteurs s'attardent un peu plus sur les techniques « récentes » d'analyse prédictive qu'ils placent dans la sous-partie « artillerie lourde » : random forest (**chapitre 11**, 13 pages) ; gradient boosting (**chapitre 12**, 14 pages) ; support vector machine (**chapitre 13**, 20 pages). L'exposé est plus approfondi. La description des signatures des fonctions de scikit-learn (« la » bibliothèque dédiée au machine learning sous Python) est à noter. En effet, aux méthodes sont souvent associées des paramètres plus ou moins ésotériques. Elles pèsent sur le comportement des algorithmes. Autant savoir de quoi il retourne et comment les manipuler. J'avoue être resté sur ma faim quand même. Peut-être aurait-il mieux valu se focaliser sur ces 3 méthodes et les étudier plus précisément avec des exemples traités de manière poussée, quitte à laisser de côté la première sous-partie. Je ne sais pas. Cela aurait réduit le caractère encyclopédique de l'ouvrage ceci étant dit.

La troisième partie est certainement la plus intéressante de l'ouvrage. Elle est consacrée à la pratique du data science. Elle commence par la présentation des outils d'évaluation des modèles (**chapitre 14**). On notera en particulier la partition des données en 3 portions pour la démarche de modélisation : la première pour l'entraînement des modèles ; la seconde pour la validation, on s'en sert essentiellement pour identifier le meilleur paramétrage des algorithmes ; la troisième pour la mesure des performances (données de test).

La démarche est symptomatique de la pratique actuelle du machine learning. En effet, traditionnellement, on se contente de subdiviser les données en ensembles d'entraînement et de test. Mais, dans l'optique de maximisation des performances prédictives, très prégnant dans la culture « data science », utiliser l'échantillon test pour détecter le meilleur modèle ou le meilleur paramétrage lui ferait perdre son caractère d'arbre impartial. Passer par un troisième ensemble, dit de validation, pour l'optimisation des modèles permet à l'ensemble de test de conserver sa fraîcheur et son innocence (page 158).

Les deux chapitres suivants abordent les problèmes des espaces à grande dimension (**chapitre 15**) et du traitement des valeurs manquantes ou aberrantes (**chapitre 16**). Les auteurs cherchent avant tout à poser les problèmes. Ces domaines ont de telles ramifications qu'il faudrait de toute manière des ouvrages entiers pour les traiter sérieusement.

La **chapitre 17** correspond à une étude de cas sur les données TITANIC. Il s'agit d'une vraie étude de cas, reproductible pas à pas. La présentation des solutions pour enrichir les variables est particulièrement instructive. Au fil des pages, nous passons d'un taux de succès de 67.45% (page 204) à 82.38% (page 214). J'avoue avoir été enchanté par cet exemple. Il montre que le travail du data scientist s'apparente à un sacerdoce. Il passe par le choix des techniques et de leur paramétrage, tout le monde en parle. Mais il passe aussi par une préparation fine des variables. C'est nettement moins fun, j'en conviens, mais les résultats peuvent être tout aussi spectaculaires.

Le **chapitre 18** nous parle du challenge Tradeshift (2014) relatif à la catégorisation de textes. Les auteurs y ont concouru avec succès (2nde place). Il se lit comme un roman policier[5]. La démarche est exclusivement guidée par la performance. Il est difficile d'avoir les détails, mais quelques idées particulièrement intéressantes sont émises. Je ne connaissais pas le mécanisme de ''hashing trick'' par exemple (page 221). Il permet, entres autres, de tenir compte des modalités des variables catégorielles non présentes dans notre échantillon de données, au prix certes d'un décuplement de la dimensionnalité. Je savais qu'on pouvait s'appuyer sur l'appartenance des individus aux feuilles pour définir des proximités dans les Random Forest. En revanche, utiliser l'indice des feuilles comme variable explicative ne me serait jamais venu à l'esprit (page 223). Je connaissais le stacking depuis longtemps, mais c'est une des rares fois où je lis la description d'une implémentation opérationnelle et efficace de la solution (pages 227 et suivantes).

Il reste au final que l'empilement des modèles pour atteindre les meilleures performances est caricatural. Un ami m'expliquait récemment qu'il ne se sent pas d'enseigner à ses étudiants une démarche de modélisation consistant à « bourriner à fond » sur les méthodes (ce n'est pas vraiment poétique en plus). Oui, je comprends et je partage son avis. Mais nous sommes dans un contexte différent dans les challenges. Les auteurs restent très lucides sur la question d'ailleurs : « les modèles gagnants sont souvent d'une complexité terrifiante, impossible à mettre en production et encore moins à maintenir » (page 237). Une fois ceci posé, je reconnais que la trame du chapitre est particulière passionnante.

[5] Un retour d'expérience est disponible sur YouTube : https://www.youtube.com/watch?v=Vgosojbkx6E

La dernière sous-partie est consacrée aux séries temporelles. Le **chapitre 19** précise la notion de temporalité et fait la distinction entre les méthodes exponentielles (lissage exponentiel) et les méthodes probabilistes.

Le **chapitre 20** fait le lien entre machine learning et modélisation des séries temporelles. Une première idée avancée est qu'une préparation judicieuse des données permettent de tenir compte des spécificités des données temporelles (ex. saisonnalité), et d'utiliser les approches usuelles de machine learning. Une autre idée consiste à détecter des motifs séquentiels pour enrichir les bases d'entraînement présentées aux algorithmes prédictifs.

Le **chapitre 21** présente un cas pratique de modélisation. Les auteurs relatent une mission qu'ils ont menée pour le compte d'un de leurs clients. Il permet de se rendre compte des difficultés auxquelles peuvent être confrontés les data scientists. Là pour le coup, il n'est plus question de « bourrinage » à tort et à travers. Il faut une analyse fine reposant sur une bonne connaissance des caractéristiques du domaine et des données. Et au final, les utilisateurs sont souvent sensibles à cet argument, on s'appuie sur des méthodes simples mais robustes, en l'occurrence un perceptron simple dans cette étude. C'est un exemple à montrer à tous les apprentis data scientists. Il vient en contrepoint de celui du chapitre 18.

Le dernier chapitre (**chapitre 22**) est consacré au clustering des séries temporelles. Le principe est le suivant : plusieurs objets sont observés dans le temps, il en résulte plusieurs séries temporelles, l'objectif est de regrouper des objets présentant des profils de chroniques « similaires ». L'affaire n'est pas triviale, loin s'en faut. Il peut y avoir des différences d'échelle, du bruit, ou des décalages temporels (page 281). La matrice de distance entre objets présentée à l'algorithme de classification automatique (dans le cas d'une classification ascendante hiérarchique) doit traduire fidèlement l'idée métier de la proximité entre les séries (ex. deux séries journalières avec exactement les mêmes profils mais décalées d'une journée doivent-elles être considérées similaires ou pas ?). Les auteurs décrivent les différentes approches permettant de rendre les séries exploitables pour le clustering. Deux exemples illustrent leur propos. La première concerne le clustering des chroniques issues de Google Trends, les séries brutes sont directement utilisées. La seconde traite de données en provenance d'une usine de production de semi-conducteurs. Dans ce cas, des indicateurs (features) sont extraits des séries dans un premier temps (tendance,

autocorrélation, etc.). La classification est réalisée à partir de cette représentation intermédiaire.

En **conclusion**, je dirais que l'ouvrage d'Eric Biernat et Michel Lutz nous offre une sorte de balade (romantique ?) dans les jardins du Data Science. Clairement, le chercheur ou l'apprenti-chercheur en machine learning n'y trouveront pas leur compte. Mais est-ce vraiment un problème ? Il existe des ouvrages spécialisés pour cela. Certains sont accessibles librement sur le web (ex. le très fameux ''The Elements of Statistical Learning'' de Hastie, Tbishirani et Friedman). Il reste qu'une vue globale sur les techniques de data mining est toujours bonne à prendre, même si les auteurs se sont avant tout focalisés sur les méthodes d'analyse prédictive.

Le lecteur désireux d'opérationnalité immédiate se sentira mieux même si, tous comptes faits, seule l'étude de cas relatif aux données « TITANIC » (chapitre 17) est réellement reproductible dans les détails, avec des sorties commentées pas à pas. Les autres exemples de codes Python ou R servent essentiellement à appuyer le texte.

Finalement, le principal mérite de l'ouvrage est de nous donner une certaine perception de la pratique du data science aujourd'hui. Prenons les challenges par exemple. Beaucoup de gens en parlent dans mon entourage (*tu connais Kaggle ?*), mais c'est bien la première fois que je lis quelque chose de la part de personnes qui y ont réellement participé avec succès, qui disent des choses sensées sur le sujet (l'un entraîne l'autre sûrement), avec une mise en perspective permettant de cerner les bonnes pratiques assurant les meilleures performances dans ce domaine. Je le dis souvent à mes étudiants. Des supports de cours sur la décomposition biais-variance de l'erreur ou la maximisation de la marge, on en trouve partout. Mais des professionnels qui partagent leur expérience, c'est autrement plus rare et précieux.