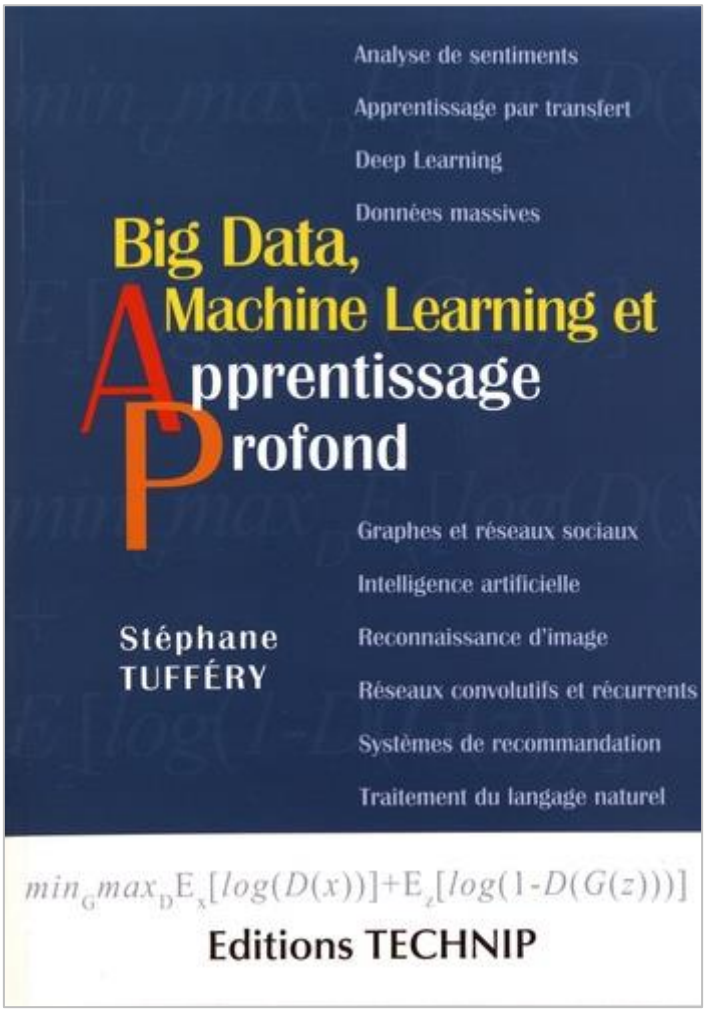
Big Data, Machine Learning et Apprentissage Profond

Stéphane Tufféry

Technip, 2019.



Le premier ouvrage de Stéphane Tufféry que j'ai eu en main a été « Data Mining et Scoring » (Dunod, 2002). Il m'avait beaucoup intrigué à l'époque parce que ça faisait un moment déjà que j'enseignais la matière à l'Université. Mais je l'associais plutôt à une compétence assez restreinte des chargés d'études statistiques. Je m'étais rendu compte à la lecture du livre qu'il y avait un espace pour l'industrialisation de la pratique du data mining en entreprise. Certains chapitres étaient particulièrement pertinents dans cet esprit, comme celui consacré aux « (Les) facteurs de succès d'un projet de data mining » (chapitre 6) par exemple. Signe des temps, il était de bon aloi de parler de « data mining » plutôt que de « machine learning » à l'époque. Les choses ont bien changé depuis comme le montre l'inversion de la popularité des termes de recherche observables sur Google Trends (Figure 1).

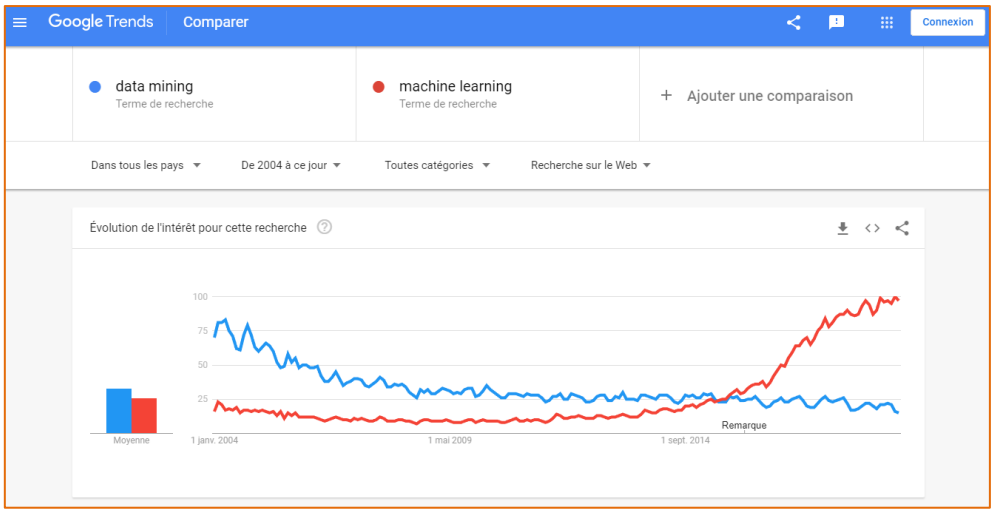


Figure 1 - Popularité des termes de recherche "data mining" et "machine learning" de 2004 à aujourd'hui

Son second ouvrage que j'ai lu en détail est « Data Mining et Statistique décisionnelle – L'intelligence des données » (Technip, 2012). Il reprend en partie le précédent. Mais il a plutôt une vocation encyclopédique en passant en revue la majorité des techniques d'apprentissage automatique. Nous l'avions tellement apprécié dans notre département que nous l'offrions aux majors de promotion du Master SISE (Statistique et Informatique – Data Science) lors de la cérémonie annuelle de remise de diplôme.

Lorsque Stéphane m'a sollicité pour relire la préversion de « **Big Data, Machine Learning et Apprentissage Profond** » (Technip, 2019), bien sûr que je me suis précipité dessus. Je donne des cours à l'Université, dans des domaines où les évolutions de tout ordre (méthodologiques, technologiques) sont très rapides. Je suis constamment à l'affût des outils, des thématiques, qui me permettent d'améliorer l'intérêt et la consistance de mes enseignements. C'est comme si on demandait à un bègue d'atteindre le débit de parole de Jacques Brel (La valse à mille temps). Il signe tout de suite. Emblématique de l'époque actuelle, on retrouve dans le titre les termes aujourd'hui « à la mode » : big data, machine learning, apprentissage profond (deep learning). Sur la couverture de l'ouvrage, nous observons également les principales applications de la data science contemporaine : analyse des sentiments, reconnaissance d'images, système de recommandation, etc.

Sans que cela corresponde à un découpage séquentiel, l'ouvrage est composé de plusieurs grands thèmes à mon sens : les chapitres didactiques qui permettent de situer les sujets phares de la science des données (« Big Data », « Intelligence artificielle », « Le traitement des grands volumes de données ») ; ceux consacrés aux outils et à l'optimisation des traitements (« Les outils informatique pour le Big Data », « Le traitement des grands volumes de données », « Big Data avec R », « Big Data avec d'autres logiciels ») ; les techniques de machine learning, dont le

fameux Deep Learning tellement apprécié aujourd'hui (« Data Science et méthodes pour le Big Data », « L'apprentissage profond (Deep Learning) », « L'apprentissage profond en pratique ») ; les applications du machine learning (« La reconnaissance de l'écriture manuscrite », « Le traitement du langage naturel », « L'analyse des réseaux sociaux »).

Particularité intéressante, l'auteur a fait l'effort de référencer les outils, les packages R en l'occurrence, pour les techniques ou pratiques décrites. Il illustre opportunément son propos de code R pour donner un tour plus concret à son discours tout au long de l'ouvrage. Je le fais moi-même pour mes supports de cours. Je me suis longuement demandé si, ce faisant, je n'enfermais pas les étudiants dans une culture spécifique à un logiciel. A l'usage, en effectuant mes enseignements, je me suis rendu compte que le langage R est assez explicite. Les étudiants conservent suffisamment de recul pour s'attacher au fond (la compréhension des méthodes) plutôt qu'à la forme (l'expression du code R).

Dans ce qui suit, je décrirai les différents sujets traités par l'ouvrage. Malgré ce que j'ai pu annoncer ci-dessus concernant les thématiques abordées, je respecterai l'ordre chronologique des chapitres décidé par l'auteur pour ne pas perdre le lecteur.

Chapitre 1 - **Le Big Data.** Il tente de cerner le phénomène Big Data, universellement populaire ces dernières années (Figure 2 ; page 12 de l'ouvrage).

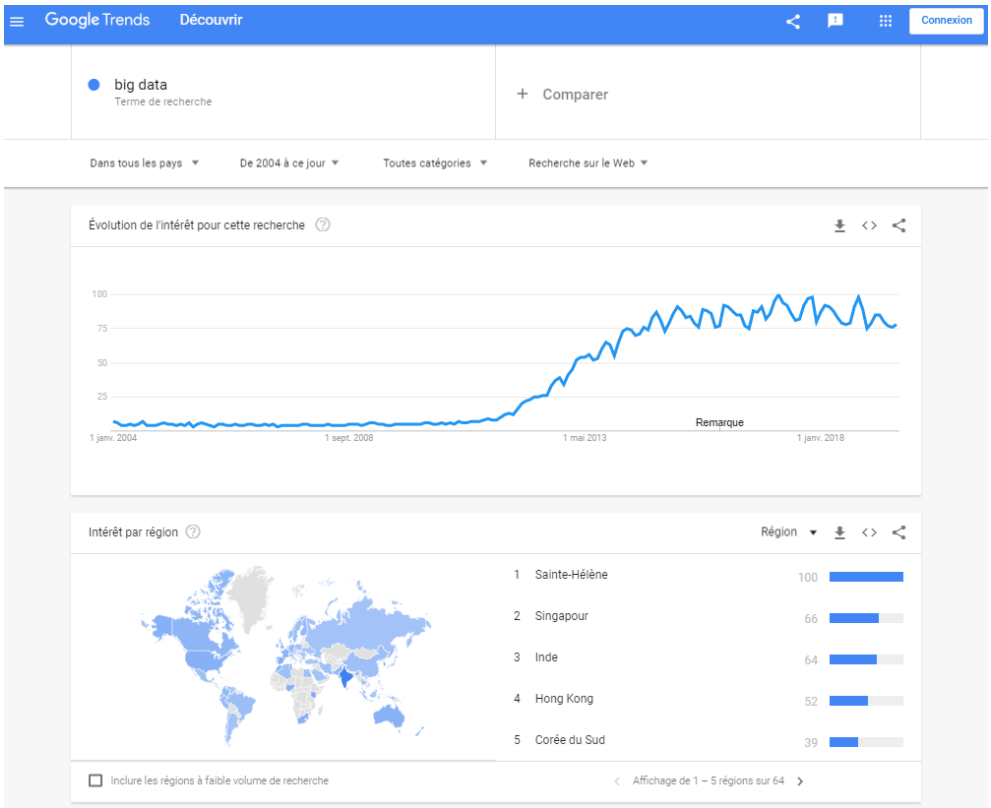


Figure 2 - Popularité du terme de recherche "big data" depuis 2004

L'auteur le présente sous les angles des différentes applications et acteurs. Le fait est que le phénomène Big Data est omniprésent dès que le recueil des données est possible c.-à-d. quasiment tous les domaines de la société aujourd'hui, y compris la criminalité (page 35). La vague Open Data (page 40) ne fait qu'amplifier l'engouement pour la donnée. Je fais moi-même travailler mes étudiants sur ces sources d'information parce qu'elles sont souvent très riches, et peuvent donner lieu à des analyses à forte valeur ajoutée, notamment lorsque nous sommes en mesure de croiser plusieurs bases.

Chapitre 2 - Les outils informatiques pour le Big Data. Il est consacré aux défis informatiques du Big Data. Le calcul distribué en est un avec le système Hadoop et la programmation MapReduce. Les bases de données NoSQL, qui permet d'aller au-delà des bases relationnelles, adaptées pour la production mais pas pour le stockage massif, en est un autre. Cette obsession pour le recueil et la thésaurisation pose la question de la protection des données personnelles (page 52). La préoccupation est légitime. Elle est systématiquement évoquée par le public lorsque je fais des présentations sur la data science. Je m'en sors toujours par une pirouette en disant que Ricco le citoyen s'en inquiète bien sûr mais n'a pas vraiment de réponse, et que Ricco le scientifique, fan des méthodes statistiques et des technologies informatiques, met un voile pudique sur le sujet. « Science sans conscience n'est que ruine de l'âme » disait Rabelais. L'inquiétude ne date pas d'hier. Elle est intemporelle et dépasse largement le cadre du Big Data.

Chapitre 3 - Le traitement des grands volumes de données. Les stratégies pour l'appréhension de la volumétrie sont abordées dans ce chapitre. Elles peuvent être statistiques (échantillonnage, modèles parcimonieux) ou informatiques (complexité des calculs, parallélisation des traitements, exploitation de la GPU – processeur graphique). Beaucoup de personnes parlent de la GPU, mais j'ai rarement vu des applications reproductibles de son exploitation en machine learning. L'auteur montrera une expérimentation concrète dans le contexte de l'apprentissage profond dans le chapitre 6 (section 6.4, page 220).

Chapitre 4 - Data science et méthodes pour le Big Data. On rentre dans la technique avec ce chapitre. Les méthodes « modernes » d'analyse prédictive, pour certains basés sur des algorithmes d'optimisation, sont présentées : la régression pénalisée (Ridge et Lasso), les stratégies d'agrégation de modèles, notamment d'arbres de décision (bagging, forêts aléatoires, boosting, extra-trees). Ces approches font partie de la panoplie indispensable du statisticien. Elles sont enseignées dans les formations en data science. La présentation est accompagnée d'indications des packages R qui les implémentent. La question du paramétrage, et de son influence sur les performances des modèles agrégés, est discuté.

Je m'autorise un petit aparté ici. Plusieurs packages proposent la même méthode parfois. Les étudiants me demandent sur la base de quels critères choisir. Je leur fais une réponse en deux temps. Regardez déjà si l'outil a donné lieu à une publication dans le journal de R (The R Journal). Cela veut dire que le package est passé par un processus d'évaluation par des scientifiques. Je suis plutôt enclin à trouver cela rassurant. Dans un deuxième temps, testez par vous-même. Avec notre connaissance des méthodes, nous sommes en mesure d'évaluer la qualité des implémentations, en termes de performances prédictives, de richesse et pertinence du paramétrage, et de performances calculatoires (rapidité et capacité à traiter des grandes bases). Le code source est disponible. Un data scientist est un informaticien qui a obligatoirement (!) des compétences avancées en programmation. Il lui appartient d'inspecter les programmes, voire de les modifier pour en dériver des variantes plus efficaces pourquoi pas.

Chapitre 5 - L'apprentissage profond (Deep Learning). Pour le mordu des techniques de machine learning que je suis, les affaires réellement sérieuses commencent avec ce chapitre. Le terme « deep learning » est galvaudé, c'est le moins qu'on puisse dire. Combien fois je me suis précipité sur une référence qui me paraissait sérieuse, pour me rendre compte qu'on nous servait des cours de réseaux de neurones vieux de 30 ans. J'en attendais beaucoup. J'ai dévoré ce chapitre lorsque j'ai reçu la première épreuve du livre.

Après une présentation succincte des principes et librairies de réseaux de neurones profonds, l'auteur commence par décrire le perceptron, sous sa forme théorique avec les formules mathématiques associées, et sous sa forme pratique via des fonctions R (page 99). Il met l'accent sur la nécessité de normaliser les données pour assurer une bonne exploitation de l'information disponible. La différence de performances selon que l'on normalise ou pas les variables est flagrante sur la base Spam utilisée pour illustrer le processus d'apprentissage.

Les réseaux de neurones convolutifs (CNN – convolutional neural nets), si populaires aujourd'hui pour le traitement d'images, constituent le cœur du chapitre. Plusieurs passages clés permettent de comprendre le principe sous-jacent : « (les CNN) exploitent le fait que les variables en entrée ne sont pas indépendantes mais sont liées par une structure spatiale » (page 102) ; « (dans un CNN), une unité n'est connectée qu'à un sous-ensemble des unités de la couche précédente, ce sous-ensemble correspondant à une région locale… » (page 103). On comprend que ce type de réseau soit si bien adapté aux images qui correspondent parfaitement à ces caractéristiques. Il détaille par la suite les principales couches et mécanismes d'apprentissage du CNN : la convolution, le pooling (sous-échantillonnage), la normalisation par lot, le dropout, etc. L'architecture type d'un CNN est présenté en page 117. Très honnêtement, il faut relire plusieurs fois cette partie pour en saisir toutes les subtilités. Les sections (5.15 et 5.16) dédiées à

l'utilisation des CNN pour le traitement d'image permet de mieux en appréhender les mécanismes. Peut-être même que je conseillerais de passer directement au chapitre suivant (Chapitre 6 – L'apprentissage profond en pratique) dévolu à la pratique des réseaux de neurones profonds sous R pour renforcer notre compréhension. Par expérience, je me suis rendu compte que la mise en œuvre sur des données exemples permet de mieux appréhender les concepts théoriques. Et puis, se mettre devant un ordinateur est toujours plus rassurant pour un informaticien.

Les réseaux récurrents et les auto-encodeurs (avec des exemples de traitements sous R) sont ensuite traités, de manière moins approfondie (que le CNN) cependant. Le chapitre se termine enfin par un tour d'horizon des autres applications de l'apprentissage profond : la détection d'objets, la voiture autonome, jeux de go et d'échecs, etc.

Chapitre 6 - L'apprentissage profond en pratique. « Théorie sans pratique = culotte sans élastique » était une phrase que j'aimais bien déclamer sur tous les tons quand j'étais enfant. Je ne sais pas d'où elle me vient, mais elle est assez pertinente dans nos métiers je trouve. Avec ce 6ème chapitre, l'auteur nous emmène dans l'action et c'est tant mieux. Le discours est articulé autour de la présentation de différentes librairies pour le deep learning.

L'auteur commence avec MXNET, sous R ici, mais elle est également disponible sous Python. Il décrit l'application d'un réseau de convolution sur la base MNIST (reconnaissance de chiffres manuscrits). Le code R commenté (page 165 et suivantes) permet de détailler les étapes. Quand-même, je conseille au lecteur de consulter au préalable la présentation approfondie de la base (chapitre 7, page 237) pour apprécier pleinement les étapes de préparation des données (page 166). La structure du réseau sous forme graphique (page 172) est importante pour comprendre les successions de commandes R qui permettent d'élaborer son architecture (pages 167 et 168).

Toujours avec MXNET, nous attaquons la base CIFAR-10 ensuite (pages 177 et suivantes). Le processus est le même que précédemment. Stéphane met en lumière la possibilité de sauvegarder les modèles entraînés (page 181), que l'on peut utiliser pour le déploiement par la suite. Via le même mécanisme, nous pouvons exploiter des modèles qui ont été pré-entrainés par d'autres chercheurs. Cette fonctionnalité est primordiale lorsque l'on sait que la construction d'un réseau efficace peut mobiliser des ressources de calcul considérables.

Nous passons au tandem Tensorflow-Keras ensuite (pages 183 et suivantes), également disponible sous R et Python. La démarche est la même (qu'avec MXNET) pour le traitement des bases MNIST et CIFAR-10. Un exemple supplémentaire est fourni avec la discrimination des images de chiens et de chats (page 205).

L'installation de ces librairies sur nos machines n'est jamais une sinécure. Elle devient périlleuse lorsque nous souhaitons exploiter les capacités du processeur de la carte graphique (GPU – Graphical Processor Unit) (pages 220 et suivantes). L'auteur décrit les étapes qui lui ont permis d'en arriver à bout. Je pense quand-même qu'il faudra s'accrocher pour pouvoir reproduire l'opération. Ne disposant pas du matériel adéquat, je n'ai pas pu tester en ce qui me concerne.

PyTorch est la troisième bibliothèque explorée dans ce 6ème chapitre. Le jeu de données Fashion MNIST est mis à contribution cette fois-ci. L'objectif est de pouvoir reconnaître des catégories de vêtements. Nous connaissons la trame à ce stade de notre lecture. L'expression des commandes est différente simplement, d'autant que nous travaillons sous Python dans cette partie.

Pour conclure ce chapitre, je dirais que j'y ai passé énormément de temps. Je l'ai lu et relu à plusieurs reprises avec d'attention. Il est à prévoir que j'y puise allègrement des idées pour mes propres travaux dirigés sur machine en Master, en faisant travailler les étudiants sur d'autres bases d'images. Elles sont pléthores sur le web.

Chapitre 7 – La reconnaissance de l'écriture manuscrite. Ce chapitre est conçu comme une étude de cas sur la base MNIST. Stéphane prend le temps de décrire la base et la problématique cette fois (sections 7.1 et 7.2). Il applique d'abord les méthodes prédictives dites « classiques » (analyse discriminante, régression logistique, perceptron, etc.). Nous découvrons à cette occasion la librairie H2O, que j'avais moi-même exploré sous R et surtout sous Python. Dans le meilleur des cas, un perceptron à 3 couches cachées, le taux d'erreur en test est de 1.12%. Il travaille ensuite sur les réseaux de neurones convolutifs avec MXNET et KERAS (le code source R décrit surtout le traitement avec MXNET) (page 271). Un comité de réseaux que l'on fait voter atteint le taux de 0.39%, montrant, sur cette base tout du moins, la supériorité des réseaux convolutifs dans le traitement d'images. Il faut quand même un niveau d'expertise élevé pour comprendre le code R de constitution du comité (page 274).

Chapitre 8 – Big Data avec R. R, on en parle justement dans ce 8ème chapitre. Après une présentation rapide du fonctionnement de R, Stéphane s'attarde sur la gestion de la mémoire (sections 8.3 et 8.4). A raison, parce qu'elle constitue un des goulots d'étranglement de l'utilisation du logiciel dans le traitement des données volumineuses. Les différents aspects de la programmation sous R sont ensuite présentés (implémentation des boucles, vectorisation, parallélisation, etc.). Des comparaisons de temps de calcul permettent de mettre en perspective les variantes d'instructions destinées à réaliser les mêmes traitements. La section 8.8 (Traitement des données massives sous R) a particulièrement retenu mon attention. Nous y découvrons les packages spécialisés pour le traitement des fortes volumétries, pour l'appréhension et manipulation des grandes tables de données (readr, data.table, dplyr), pour déporter une partie

des objets sur disque et non plus en mémoire vive pour éviter de saturer cette dernière (bigmemory, ff, packages pour les traitements sous Hadoop). Des exemples d'implémentation de la régression sur des très grandes bases de données permet de situer l'efficacité des solutions (avec bigglm notamment).

R a manifestement beaucoup progressé ces dernières années. J'avais mené une petite comparaison avec d'autres outils il y a quelques années (Régression logistique sur les grandes bases, **2012**). R n'y était pas vraiment à son avantage. Refaire la même expérimentation – à machine égale, ce qui n'est plus possible en ce qui me concerne – avec les mêmes outils donnerait des résultats très différents je pense. Il semble qu'avec la version 3.5.0, des progrès sensibles aient été introduits encore. L'idée ancienne qui catalogue R comme un outil peu adapté aux très grandes volumétries n'est plus vraiment justifiée. Elle repose souvent sur la méconnaissance de l'outil, de ses évolutions, et des packages qui l'accompagnent. Une expérimentation récente (février 2019), où je compare les performances de R et Python lors du chargement et la manipulation d'une très grande base (une dizaine de millions d'observations), montre de manière spectaculaire les performances du dispositif « logiciel R + packages judicieusement choisis ».

Chapitre 9 – Big Data avec d'autres logiciels. Des autres logiciels, il en question justement dans ce chapitre 9. On y trouve pêlemêle des librairies optimisées pour le calcul matriciel, H2O, les outils Microsoft, SAS, etc. Je retiendrai particulièrement 2 sections en ce qui me concerne. D'une part, les sections consacrés à l'exploitation de l'environnement Spark avec les packages sparklyr et sparkR (section 9.5). La compétence Spark est très demandée par les acteurs du traitement des données massives. D'autre part, la présentation de Python dans le contexte du machine learning (section 9.6). La discussion des performances comparée de R et Python revient souvent dans les échanges que je peux avoir avec mes collègues. Certains ont des positions très tranchées, voire hystériques. Je ne rentrerai jamais dans ce jeu. J'ai fait le pari de la double compétence pour mes étudiants et ça fonctionne plutôt bien. Après, selon le contexte de l'étude, les objectifs, l'environnement dans lequel ils évolueront (si vous tombez chez des fadas de Python, vous, stagiaire, n'allez pas commencer à faire du prosélytisme pour les convertir à R…), il leur appartient de choisir à bon escient l'outil qui leur paraît le plus adapté. Stéphane propose un tableau des équivalences entre les packages spécialisés pour R et Python (page 390).

Chapitre 10 – Le traitement du langage naturel. Le « natural language processing » (NLP) est une des applications phares du machine learning. Le domaine n'est pas récent. Stéphane en décrit rapidement plusieurs aspects (identification de la langue, tokenisation, identification des catégories grammaticales, lemmatisation, etc.). Les affaires sérieuses (si je puis dire)

commencent avec la représentation vectorielle des documents (section 10.4), préalable indispensable pour la réalisation de traitements statistiques subséquentes. On nous parle ensuite de la réduction de dimensionnalité avec la création de représentation intermédiaires correspondant à des thématiques identifiées (par le calcul), ou encore la prise en compte de la proximité des termes (contexte). Stéphane étudie ensuite les méthodes neuronales en analyse textuelle (section 10.5). Sujet que j'ai rarement vu abordé de manière convaincante dans les références francophones. Il y a matière à réaliser des choses intéressantes pourtant.

L'utilisation des réseaux de neurones récurrents (LSTM : long short-term memory) est détaillée pour la génération de textes et, ce qui m'intéresse surtout dans le cadre de mes enseignements, dans la catégorisation automatique des documents. Il faut quand-même méchamment s'accrocher pour tout saisir. Je trouve pour ma part que l'architecture du réseau illustratif en page 475 (sous Keras) est très parlant. La couche LSTM vient contextualiser les thématiques dégagées par le prolongement de mots (word embedding).

D'autres exemples suivent : l'application d'un perceptron H20 pour la catégorisation de documents où l'on s'appuie sur le représentation intermédiaire Word2vec ; l'utilisation d'un réseau de neurones convolutif, l'enjeu étant de transformer la représentation vectorielle des documents en représentation matricielle ; la détection de spams à l'aide d'un réseau de neurones récurrent. L'intérêt pour nous est de pouvoir identifier, pour chaque exemple, les étapes essentielles pour pouvoir les reproduire sur d'autres corpus.

Chapitre 11 – L'analyse des réseaux sociaux. Le chapitre 11 est consacré à l'analyse des réseaux sociaux. Nous y découvrons, entres autres, l'analyse des relations entre entités (personnes le plus souvent, mais ça peut autre d'autres types d'objets) représentés sous forme de graphes. L'enjeu est de les analyser pour identifier les influences, les positions relatives des uns par rapport aux autres (notion de centralité), pour identifier par exemple des communautés (section 11.6). Schématiquement, l'idée serait d'appliquer des algorithmes de classification automatique (clustering) sur des graphes. L'idée est connue, l'originalité tient à la source d'information (un graphe) à traiter. Un exemple de traitement sous R est proposé (section 11.6.5). L'analyse des tweets (Twitter) est un autre sujet brûlant du domaine (section 11.10). L'auteur déroule toutes les étapes, en partant de l'extraction des messages en ligne via l'API de Twitter, jusqu'à l'application des techniques de machine learning, pour obtenir un regroupement sémantique des termes par exemple.

Chapitre 12 – L'intelligence artificielle. Le dernier chapitre de l'ouvrage a plus une vocation culturelle en nous donnant un aperçu des développements récents de l'intelligence artificielle. Dès qu'il n'y a pas de formules, dès qu'il n'y a pas de code R, ça m'intéresse moins j'avoue. Mais

il reste que nous avons le devoir de pouvoir positionner nos connaissances, nos compétences, dans un cadre plus global en relation avec les évolutions de notre société. Ce chapitre y contribue.

En conclusion de cette fiche de lecture, je dirais que l'on peut percevoir un double niveau de lecture de l'ouvrage de Stéphane Tufféry. D'une part, nous disposons d'un tour d'horizon global, un panorama des méthodes et pratiques de la data science aujourd'hui, avec un accent particulier sur l'apprentissage profond. D'autre part, nous disposons d'une référence opérationnelle qui nous permet de mettre en œuvre concrètement sur des exemples réels les compétences nouvellement acquises à la lecture des chapitres techniques.

En ce qui me concerne, j'ai beaucoup aimé la lecture de « Big Data, Machine Learning et Apprentissage Profond ». La plupart des chapitres correspondent peu ou prou aux thématiques que j'assure moi-même en Master SISE à l'Université. J'ai passé énormément de temps sur les chapitres 5 (Deep Learning), 6 (Pratique du Deep Learning), 8 (Programmation R avancée) et 10 (Traitement du langage naturel, l'utilisation des réseaux de neurones profonds en particulier) parce qu'ils me permettront de faire évoluer mes propres enseignements. Les étudiants sont gagnants dans l'histoire, et c'est tant mieux.