De la Statistique à la Science des (grosses) Données

Résumé

Statistique, fouille ou Science des Données, les appellations changent le volume et la diversité des données explosent, les technologies se succèdent, les modèles et algorithmes se complexifient. L'estimation devient un apprentissage, la prévision remplace l'explication.

Le parcours pour devenir data scientist *est structuré en quatre parties :*

Saison 1 *(L3) Statistique élémentaire, descriptive* vs. *inférentielle.*

Saison 2 *(M1) Statistique Exploratoire multidimensionnelle et apprentissage non supervisé.*

Saison 3 *Apprentissage Statistique / Machine supervisé.*

Saison 4 *(M2) Technologies pour la Science des (grosses) Données.*

plus des réflexions sur : Statistique et Déontologie scientifique.

1 Origines de la *D*ata Science

Le terme de *data scientist* à été "inventé" par D. Patil (LinkedIn) [1] et J. Hammerbacher (Facebook) en cherchant comment caractériser les métiers des données pour les offres d'emploi : *Analyste, ça fait trop Wall Street ; statisticien, ça agace les économistes ; chercheur scientifique, ça fait trop académique. Pourquoi pas "data scientist" ?*

Une "définition" attribuée à J. Wills (Cloudera) est souvent reprise :*Data scientist (n) : Person who is better at statistics than any software engineer and better at software than any statistician*

La Science des Données n'est pas une nouvelle science créée *ex nihilo* mais l'association de compétences (informatique, mathématiques, métiers) résultat d'une longue évolution parallèle à celle des moyens de calcul et des volumes de données concernés. Cette évolution est passée par l'*analyse des données* en France, l'*Exploratory Data Analysis* ou EDA au USA, le *data mining* ou fouille des données puis la *Bioinformatique*.

En voici un bref résumé nécessairement schématique avec une chronologie linéaire :

1930-70 – hOctets Il était une fois la *Statistique* (inférentielle) : une question, (*e.g.* biologique), associée à une *hypothèse expérimentalement réfutable* H_0, une expérience *planifiée* avec un échantillon *représentatif* de $n \approx 30$ individus observés sur p (moins de 10) variables, un modèle *linéaire gaussien* supposé *vrai*, un test, une décision, donc une réponse qui peut être inférée à la population en contrôlant le risque (généralement 5%) de rejeter à tort H_0.

1970s – kO Les premiers outils informatiques se généralisant et, pour échapper à l'impérialisme du modèle linéaire, l'*analyse des données* (Caillez et Pages, 1976)[2] se développe en France ; l'*Exploratory Data Analysis* ou EDA aux États-Unis (Tukey 1977)[10]. L'objectif est alors de décrire ou explorer, prétendument sans modèle, des données déjà plus volumineuses.

1980s – MO En Intelligence Artificielle (IA), les *systèmes experts* expirent, supplantés par l'apprentissage des *réseaux de neurones*. La Statistique développe des modèles non-paramétriques ou fonctionnels.

1990s – GO *Data Mining* et *Premier changement de paradigme*. Les données ne sont plus *planifiées*, elles sont préalablement acquises et basées dans des entrepôts pour les objectifs usuels (*e.g.* comptables) de l'entreprise. L'aide à la décision les valorise : *From Data Mining to Knowledge Discovery* (Fayyad ; 1997)[4]. Les logiciels de fouille regroupent dans un même environnement des outils de gestions de bases de données, des techniques exploratoires et de modélisation statistique. C'est l'avènement du marketing quantitatif et de la gestion de la relation client (GRC ou CRM). L'IA se développe avec l'émergence du (*Machine Learning*) dont un sous-ensemble de méthodes est mis en exergue par le livre de Vapnik (1998) : *The Nature of Statistical Learning Theory*.

1. Entretien publié dans un article de l'Obs.

2000s –TO *Deuxième changement de paradigme*. Le nombre p de variables explose (de l'ordre de 10^4 à 10^6), notamment avec les biotechnologies omiques où $p >> n$ et la Bioinformatique. Le FDR (*False Discovery Rate*) de Benjamini et Hochberg (1995)[1] se substitue à la p-valeur et l'Apprentissage Statistique (Hastie et al. 2009)[5] sélectionne des modèles en optimisant leur complexité par un meilleur compromis *biais vs. variance* ; minimiser conjointement erreur d'*approximati* (biais) et erreur d'*estimation* (variance).

2010s – PO *Troisième changement de paradigme*. Dans les applications industrielles, le e-commerce, avec la géo-localisation, la *datafication* du quotidien où toutes les traces numériques sont enregistrées, c'est le nombre n d'individus qui explose ; les statistiques usuelles de test, toutes significatives, perdent leur utilité au profit des méthodes d'apprentissage non supervisées ou supervisées ; les bases de données se déstructurent et se stockent dans les nuages (*cloud computing*), les moyens de calculs se groupent (*cluster*), mais la puissance brute ne suffit plus à la voracité (*greed*) des algorithmes. Un troisième terme d'erreur est à prendre en compte : celle d'*optimisation*, induite par la limitation du temps de calcul ou celle du volume des données considéré ; leur flux nécessite la construction de décisions adaptatives ou séquentielles.

Une présentation plus détaillée de la "science des données" et ses implications notamment économiques est proposée par Besse et Laurent (2015).

2 Environnement logiciel

2.1 Logiciels de fouille de données

Dès les années 90, et provoquant l'avènement de la *fouille de données* (*data mining*), les éditeurs de logiciels commerciaux et les communautés de logiciels libres ont inclus dans leurs suites, en plus des modèles linéaires classiques, les différents algorithmes d'apprentissage au fur et à mesure de leur apparition. Ceux-ci ont été intégrés à un ensemble plus complet de traitement des données en connexion avec les gestionnaires de bases de données relationnelles, le tout pilotable par une interface graphique plus ou moins conviviale : *Clementine* de SPSS, *Enterprise Miner* de SAS, *Insightfull Miner* de Splus, KXEN, SPAD,

FIGURE 1 – *À copier 100 fois.*

Statistica Data Miner, Statsoft, WEKA... Leur apparente simplicité d'utilisation a largement contribué à la diffusion de méthodes sophistiquées dans des milieux difficilement perméables à une conceptualisation mathématique abstraite et peu armés pour des développements logiciels importants.

2.2 R *vs.* Python

Dans ce paysage en constante évolution ou révolution, les langages R (2015)[8] et Python (Rossum et Guido ; 1995)[9] jouent un rôle particulier. L'analyse des offres de stage et d'emploi montre de profonds changements dans les demandes. SAS, plébiscité jusqu'à la fin du siècle dernier est largement supplanté par R et maintenant Python pour des raisons d'évidente économie mais aussi de flexibilité.

R

Toute méthode d'apprentissage est implémentée en R sous la forme d'une librairie (*package*) librement accessible. C'est même le mode de diffusion privilégié de nouvelles méthodes. Pour faciliter la tâche de leurs utilisateurs et

surtout uniformiser l'intégration de méthodes développés par des auteurs différents, Kuhn (2008)[6] propose une méta-librairie (*caret*) pouvant exécuter plus de 200 méthodes ou variantes de méthodes à partir de la même syntaxe. Néanmoins et comme Matlab, R est un langage interprété ; même en utilisant des librairies spécifiques pour paralléliser certains calculs compilés en C, les temps d'exécution de R deviennent vite rédhibitoires avec des données un peu volumineuses. De plus, son utilisation est rendue impossible (ou très difficile) dès que les limites de la mémoire interne de l'ordinateur sont atteintes.

Python

Plus récent Ross (1995)[9], le langage Python s'est considérablement développé notamment pour le traitement et l'analyse de signaux, images et séries financières. Python permet de paralléliser facilement la préparation (*data munging*) de grosses données sans les charger en mémoire avant de passer à la phase d'exploration puis de modélisation qui est elle toujours traitée en chargeant les données en mémoire.

Une des librairies : *Scikit-learn* (Pedregosa et al. 2011)[7] met à disposition les principales méthodes d'apprentissage supervisées ou non. Cette librairie n'est pas ouverte au sens où le choix d'implémentation d'une méthode est décidé au sein du groupe des développeurs principaux. L'avantage est un développement intégré et homogène, l'inconvénient, qui peut être aussi un avantage, est un choix plus restreint de méthodes accessibles. Également interprété, Python s'avère beaucoup plus rapide que R en gérant par défaut les possibilités de parallélisation d'une machine, même sous Windows.

R vs. Scikit-Learn

Le choix entre ces deux environnements repose sur les quelques points suivants :

- R et ses librairies offrent beaucoup plus de possibilités pour une exploration, des sélections et comparaisons de modèles, des interprétations statistiques détaillées avec des graphes produits par défaut.
- Mise en œuvre souvent implicite des possibilités de parallélisation, même sous Windows, par les librairies de Python.
- *Scikit-Learn* ne reconnaît pas (ou pas encore ?) la classe *DataFrame* développée dans la librairie *pandas*. Cette classe est largement utilisée en R pour gérer différents types de variables. C'est un problème dans *Scikit-Learn* pour la prise en compte de variables qualitatives complexes. Une variable binaire est simplement remplacée par une indicatrice $(0, 1)$ mais, en présence de plusieurs modalités, une variable qualitative est remplacée par l'ensemble des indicatrices (*dummy* variables $(0, 1)$) de ses modalités. Ceci complique les stratégies de sélection de modèles et rend obscure leur interprétation.

En résumé, préférer R pour modéliser et interpréter des modèles statistiques mais préférer Python pour des modélisations efficaces à seule fin prédictive au détriment de l'interprétation. Les deux approches pouvant d'ailleurs être traitées de façon complémentaire.

Enfin, si les données sont trop volumineuses pour la mémoire interne voire pour le disque d'un ordinateur, ou encore si les données sont déjà archivées sur une architecture distribuée, d'autres approches sont à considérer et abordées en saison 4 avec Spark.

2.3 Reproductibilité des analyses

Donoho (2015)[3] insiste à juste titre sur la question importante de la reproductibilité des analyses. Les médias se font régulièrement l'écho de manquements déontologiques et plus généralement du problème récurrent du manque de reproductibilité des résultats publiés dans des journaux ou revues que ce soit par exemple en Biologie ou en Psychologie. Pour un statisticien, contribuer à la prise en compte de ces problèmes consiste à produire des chaînes de traitements ou d'analyses (*pipeline*) facilement transmissibles pour être reproductibles sur des matériels standards. Deux environnements s'y prêtent particulièrement. Le premier concerne l'automatisation de la production d'un rapport en intégrant des commandes R (librairie *sweave* ou *knitr*) ou Python (*pweave*) au sein d'un source LaTeX. Ces commandes, automatiquement exécutées, provoquent l'insertion de tableaux ou graphiques. Le deuxième, plus en amont, consiste à enregistrer systématiquement l'enchaînement des commandes et de leurs résultats numériques ou graphique dans un calepin (*notebook IPython* ou *Jupyter*). La sauvegarde est faite sous un format ré-exécutable dans un environnement similaire ou sous forme de fichier au format `html, pdf`. Ce type de résultat est obtenu en exécutant le bon noyau (Python, R, Julia...) dans le même environnement *Jupyter* à partir d'un simple navigateur. C'est pour cette raison que tous les tutoriels sont exécutables sous la forme d'un calepin, notamment pour les

- Tutoriels d'initiation à R.
- Tutoriels d'initiation à Python.

À exécuter et approfondir parallèlement à la maîtrise des principales méthodes.

3 Méthodes

L'historique précédent illustre schématiquement une progression pédagogique car il est difficile d'analyser de grands ensembles de données sans maîtriser les outils de base développés pour des données plus modestes à condition de bien identifier et faire coïncider les objectifs d'une étude : exploratoire, explicatif ou prédictif, avec ceux des méthodes mis en œuvre. C'est aussi une progression méthodologique, des outils les plus simples aux plus sophistiqués, pour aborder un nouvel ensemble de données.

Cette présentation propose donc de découper schématiquement la progression de la formation d'un *data scientist*, du L3 au M2, en quatre étapes ou *saisons* regroupant chacune un ensemble de scénarios ou épisodes couplant présentation théoriques et tutoriels pratiques des différentes méthodes et donc compétences à acquérir.

Saison 1 (L3) Statistique élémentaire, descriptive *vs.* inférentielle.

Saison 2 (M1) Statistique Exploratoire multidimensionnelle et apprentissage non supervisé.

Saison 3 Apprentissage Statistique / Machine supervisé.

Saison 4 (M2) Technologies pour la Science des (grosses) Données.

N.B. Cette formation s'appuie sur des compétences parallèlement acquises en Statistique mathématique, calcul des Probabilités, Optimisation, Analyse Fonctionnelle pour une compréhension approfondie des méthodes et algorithmes utilisées, de leurs limites, et en Informatique pour leur mise en exploitation.

4 "Oublis"

Certains points n'ont pas été intégrés à ce déroulement notamment en lien avec le **V** de *variété* ou celui de *vélocité*. Il faut se rendre à l'évidence qu'il n'est pas possible de former à bac+5 un mouton à 7 pattes supposé maîtriser toutes la "science des données". Il a fallu faire des choix laissant de côté certains points :

- Méthodes d'apprentissage machine mais pas d'apprentissage statistique comme celles issues du domaine de la logique formelle. La recherche de règles d'associations (problème du panier de la ménagère) en est une. Elle consiste à identifier les co-occurences les plus fréquentes ou significatives par un ensemble de règles logiques associant variables et valeurs de celles-ci. Elle n'est pas adaptée à des volumétrie importante.
- Traitement de données structurées (variété) : graphes, trajectoires, images, signaux. Ces dernières nécessitent la projection des données sur de bases fonctionnelles adaptées (Fourier, ondelettes, splines) ou l'utilisation de distances (trajectoires GPS, graphes) ou noyaux spécifiques.
- Traitement de flux de données (vélocité). L'apprentissage se fait en ligne, voire en temps réel, et sans stockage par des algorithmes d'optimisation stochastique pour produire des décisions séquentielles, des recommandations de produits par des algorithmes de bandit.

Références

[1] Y. Benjamini et Y. Hochberg, *Controlling the false discovery rate : a practical and powerful approach to multiple testing*, Journal of the Royal Statistical Society (1995), n° 85, 289–300.

[2] F. Caillez et J.M. Pages, *Introduction à l'Analyse des Données*, SMASH, 1976.

[3] David Donoho, *50 years of Data Science*, Princeton NJ, Tukey Centennial Workshop, 2015.

[4] U. M. Fayyad, *Editorial*, Data mining and Knowledge discovery **1** (1997), 5–10.

[5] T. Hastie, R. Tibshirani et J Friedman, *The elements of statistical learning : data mining, inference, and prediction*, Springer, 2009, Second edition.

[6] Max Kuhn, *Building Predictive Models in R Using the caret Package*, Journal of Statistical Software **28** (2008), n° 5.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot et E. Duchesnay, *Scikit-*

learn : Machine Learning in Python, Journal of Machine Learning Research **12** (2011), 2825–2830.

[8] R Core Team, *R : A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016, http://www.R-project.org.

[9] Guido Rossum, *Python Reference Manual*, Rap. tech. CS-R9525, CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, The Netherlands, 1995.

[10] John W. Tukey, *Exploratory data analysis*, 1977.