Premiers résultats d'analyse de devoirs rendus lors d'une évaluation par les pairs dans un Mooc de statistiques

Mehdi Khaneboubi[1]
mehdi.khaneboubi@ens-lyon.fr

[1] Laboratoire STEF (ENS de Cachan - IFÉ/ENS de Lyon)

Résumé. Dans une perspective d'observation participante en ligne, nous avons suivi pendant 4 semaines un Mooc portant sur l'enseignement de techniques statistiques avec des instruments informatisés. Lors de l'évaluation par les pairs la plateforme attribuait automatiquement et aléatoirement aux évaluateurs autant de devoirs que disponible. C'est ainsi qu'après en avoir évalué 26 on en a ensuite examiné 75. Les productions finales des participants sont des documents particulièrement importants. De toutes les données produites dans un Mooc les devoirs des participants sont particulièrement pertinents pour identifier des apprentissages. Les devoirs déposés permettent-ils d'identifier des différences importantes ? Les étudiants exécutent-ils une procédure ou font-ils preuve de créativité ?

Mots-clés: MOOC, machine learning, évaluation par les pairs, logiciel R

Introduction

Les études sur les MOOC traitent peu des productions des apprenants et concernent en priorité leur investissement et leurs conditions de certifications (Gašević & al., 2014). Pourtant les productions des participants, sont des documents particulièrement importants. De toutes les données produites dans un Mooc (par exemple les journaux de connexions, les résultats aux quizz, les réponses aux questionnaires, les messages de forums…) les devoirs des participants sont celles les mieux à même de témoigner des apprentissages des participants. En somme, si l'on s'interroge sur la nature des apprentissages qui se produisent dans un Mooc les productions d'étudiants sont des documents primordiaux. Il s'agit de données hybrides, que l'on ne peut pas considérer comme purement déclaratives, ni comme de véritables traces d'activités. Enfin, pour les collecter il suffit le plus souvent de participer à un Mooc et saisir ainsi leur contexte de production pour prendre en compte ce qui a été demandé dans le cours et ce qui a été exposé par les participants. On a ainsi la possibilité de distinguer ce qui est le produit du dispositif pédagogique de ce qui est le résultat de processus individuels.

Dans les Mooc où l'enseignement est souvent très magistral comme sur les plates-formes Coursera ou edX, les activités des participants reposent principalement sur le suivit de vidéos, la réponse à des questions à choix multiples (QCM), des échanges sur des forums et le rendu d'une production parfois collective toujours évaluée par les pairs. Même si les QCM sont souvent conçus de façon à inciter les participants à être actifs plutôt qu'à vérifier des connaissances apprises par cœur, il s'agit tout de même d'un format général d'enseignement fortement centré sur l'enseignant et sur les contenus. La responsabilité des apprentissages effectifs repose presque exclusivement sur les étudiants. On peut donc se demander si ce format très magistral ne conduirait pas à une uniformisation des apprentissages et probablement des savoirs. En examinant lors de l'évaluation par les pairs, des rendus des participants à un Mooc on a aussi la possibilité de s'interroger sur les fonctions de l'évaluation par les pairs. Les devoirs déposés permettent-ils d'identifier des différences importantes entre les participants ? Les étudiants exécutent-ils une procédure ou font-ils preuve de créativité ? Les participants comprennent-ils ce qu'ils font ?

Pour chercher à répondre à ce questionnement exploratoire, nous avons suivi un Mooc portant sur l'apprentissage automatique (*machine learning*) disponible sur une grande plateforme. Lors de

l'évaluation par les pairs, on a évalué 101 rendus et collecté 75 devoirs que nous avons examinés au travers d'une grille d'évaluation selon des critères présentés plus loin.

L'évaluation par les pairs quelques repères

Le nombre important de participants dans les MOOC rend impossible une correction par des enseignants de chaque production rendue. C'est pourquoi lorsqu'une évaluation automatique est impossible, les apprenants se notent entre eux. L'évaluation par les pairs est une activité antérieure à l'arrivée des MOOC. Auparavant, cette méthode était utilisée principalement pour des raisons pédagogiques (Sadler & Good, 2006 ; Thomson, Smith, & Annesley, 2014 ; Topping, 1998).

Mais quels sont les écarts entre la notation d'un enseignant et celle des élèves ? On peut dire, d'un point de vue statistique, que les notes d'enseignants et d'apprenants évaluant les mêmes travaux sont très fortement corrélées (Kulkarni et al., 2013 ; Sadler & Good, 2006), mais pas suffisamment pour que ce soit satisfaisant d'un point de vue pédagogique.

Dans le cas d'une évaluation par les pairs réalisés par des élèves de sciences dans un lycée des USA au début des années 2000, Sadler & Good (2006) ont remarqué un phénomène plutôt normatif de la notation des élèves : les bons élèves ont de moins bonnes notes et les mauvais de meilleures que celle donnée par le professeur. La proximité entre les évaluations des élèves est celle de l'enseignant est très variable selon les tâches à réaliser, les critères d'évaluation, l'expérience des élèves, etc. En étudiant les évaluations par les pairs réalisés dans un cours d'algorithmique, Chinn (2005) affirme que les étudiants évaluent de mieux en mieux à mesure que le cours avance et qu'il existe un lien fort entre la qualité de leur évaluation et leur performance dans les évaluations ordinaires.

Dans une étude qui concerne l'enseignement supérieur, Falchikov & Goldfinch (2000) estiment que cette proximité est plus grande lorsque l'évaluation repose sur des critères généraux bien compris par les élèves plutôt que par une série d'items uniques. De plus, les évaluations des élèves et des enseignants se ressemblent davantage pour des procédures et des méthodes plutôt que des pratiques professionnelles. Enfin, les auteurs ne trouvent pas de meilleurs résultats en sciences et sciences de l'ingénieur que dans les autres disciplines ni entre les étudiants de premier et de deuxièmes cycles. Dans le cas d'un enseignement professionnel en Anglais langue étrangère au Japon, Saito & Fujita (2009) rapportent une similarité globale entre la notation des élèves et celle des enseignants avec des différences notables selon la difficulté des critères d'évaluation.

Statistiquement, la corrélation est importante entre notation des apprenants et de l'enseignant, d'un point de vue pédagogique il n'est pas possible de considérer les notes d'élèves comme valides sans une (ré)vision de ces notes par un enseignant (Salder & Good, 2006). C'est pourquoi différentes méthodes existent pour pondérer, corriger ou assister un enseignant dans la validation des notations d'apprenants de MOOC.

Par exemple la *Peer rank method* (Walsh, 2014) est directement inspirée de l'algorithme présumé du moteur de recherche Google. Il s'agit de donner un poids équivalent aux notes données et aux notes reçues : la notation d'un apprenant ayant de bonnes notes a plus d'importance que celle d'un apprenant avec de mauvaises. L'*Automated Essay Scoring* utilisé dans un module complémentaire d'edX et le *Calibrated Peer Review* utilisé dans Coursera, sont des systèmes d'évaluation par les pairs de textes simples et courts avec une intervention automatique permettant de réduire les tâches de correction des enseignants ou de les aider dans la validation. Cela se base sur des méthodes statistiques, des algorithmes de *machine learning* et du traitement automatique de texte (Balfour, 2013). D'après Piech et al. (2013) ces procédures peuvent être encore améliorées par des méthodes algorithmiques testées dans Coursera avec des améliorations significatives.

Matériel collecté

Présentation du cours et consignes d'évaluations

Dans une perspective d'observation participante en ligne, nous avons suivi pendant 4 semaines un Mooc portant sur l'enseignement de techniques statistiques avec des instruments informatisés. Il s'agit d'un Mooc portant sur l'apprentissage automatique (*machine learning*) dans une perspective pratique avec le langage R. Le terme *machine learning* revêt une dénomination trompeuse puisqu'il s'agit de ce que l'on appelle classiquement de méthodes d'analyse de données fortement centrées sur les techniques de prédiction. On retrouve donc les techniques statistiques traditionnelles comme les régressions ou l'analyse en composante principale. S'y ajoutent d'autres méthodes comme les arbres de classifications, la classification automatique non supervisée ou le *boosting*. Depuis 2001, les forêts d'arbres décisionnels (*Random Forest*) ont connu un succès important. Il s'agit d'un combiné des arbres de classifications et de *boosting* (pour le fonctionnement détaillé de l'algorithme cf. Breiman, 2001). Ses qualités en ont fait une technique incontournable.

Le cours traitait des techniques de mise en œuvre de ces algorithmes sur des données réelles avec le langage R (R Core Team, 2015). Il s'agissait essentiellement de modéliser statistiquement des données quantitatives et d'extrapoler la modélisation à d'autres données moins complètes. Le cours faisait explicitement référence au livre de Hastie et al. (2011) mais on y retrouve en substance l'essentiel de ce qui figure notamment dans le livre de Lantz (2013) :

- nettoyage et préparation des données,
- choix et application d'un algorithme qui produit un modèle statistique,
- évaluation du modèle conduisant éventuellement à son optimisation,
- réalisation d'une prédiction.

Le rendu final du Mooc consistait à appliquer cette méthode sur un jeu de données et à évaluer l'analyse faite par au moins 3 autres participants. Les variations possibles dans l'exécution de ce canevas sont possibles selon les données, le choix de l'algorithme ou les objectifs de l'analyse statistique. Différentes perspectives sont envisageables et offrent une combinaison importante dans la forme finale du rendu.

On trouvait dans le cours des exemples d'utilisations ainsi que des explications sur des traitements de données réels. Le cours insistait sur les techniques de validation croisée, des indicateurs de précisions (statistique du kappa, coefficient de corrélation…), d'échantillonnage, de manipulation et de visualisation de données. Le plus souvent en utilisant les algorithmes disponibles avec la bibliothèque de R intitulée *CARET* (pour Classification And REgression Training), le cours a traité d'un point de vue pratique :

- de l'analyse en composante principale (ACP) pour préparer les données,
- de la régression linaire simple en guise de premier exemple,
- des arbres de classification et de régression (CART),
- de bootstrap aggregating (*bagging*),
- des forêts d'arbres décisionnels (*Random Forest*),
- du *boosting*,
- de la classification naïve bayésienne,
- de l'analyse discriminante linéaire,
- du partitionnement en k-moyennes (*k-means*).

Tout ceci était présenté en 27 vidéos d'une durée totale d'environ 240 minutes. Chaque semaine était évaluée par un QCM composé de 5 à 4 questions qui demandait plusieurs heures de travail pour celui ou celle qui cherchait à y répondre en « jouant le jeu ». La production finale était basée sur un devoir

d'analyse et de prédictions de données. L'analyse était évaluée par les pairs et les prédictions étaient évaluées automatiquement. L'évaluation par les pairs comptait pour une part de 20 % dans la note finale permettant d'obtenir un certificat. La plateforme attribuait automatiquement aux évaluateurs autant de devoirs que disponible. C'est ainsi qu'après en avoir évalué 26 on en a ensuite collecté et examiné 75. Dans le Mooc l'évaluation par les pairs est présentée comme un moyen d'apprentissage, on pouvait y lire : « cela vous donne l'occasion d'apprendre de vos camarades étudiants, en se saisissant des astuces et des idées clés et en aidant aussi les autres. » Ce qui est bien sûr difficile à estimer en consultant les devoirs qui sont rendus avant de réaliser l'évaluation.

Le barème d'évaluation était le suivant :

- 10 points lorsqu'un fichier *Rmarkdown* et un fichier HTML ont été déposés sur Github, 5 points s'il y a seulement l'un des deux. 0 dans tout autre cas.
- 5 points si l'algorithme est utilisé et décrit, 3 s'il n'est pas décrit, 0 sinon.
- 5 points si le taux d'erreurs est évalué avec une validation croisée, 3 s'il est évalué sans validation croisée, 0 s'il n'est pas évalué.
- 1 point si le document à l'air non plagié, 0 sinon.

On remarque qu'il s'agit d'une évaluation assez succincte qui prend moins de 15 minutes par devoirs. Il faut souligner que le premier critère (10 points sur 21) est consacré à des éléments de communication des fichiers et non au travail statistique. C'est important, car cela dénote que les concepteurs du Mooc attachent une importance particulière à ce que les rendus soient facilement lisibles et évaluables. Par ailleurs, dès lors que les deux fichiers ont été déposés sur Github peu de variation dans la notation était possible j'ai par exemple délivré, en tant qu'évaluateur, 101 notes ayant pour moyenne 18/21.

Tâches à réaliser par les apprenants

Le devoir à rendre était fondé sur l'analyse d'un jeu de données produit par des sportifs. En réalisant de la musculation équipée de plusieurs dispositifs de traqueurs d'activités comme des accéléromètres, des données ont été produits rendant compte de réels exercices sportifs. L'objectif du travail à réaliser par les participants est de prédire la façon dont ils font les exercices en repérant des motifs réguliers avec un algorithme d'apprentissage automatique. Il fallait ensuite déposer un document rédigé en anglais de moins de 2000 mots. En se basant sur une copie qui nous a paru excellente, on estime que les étapes idéales de l'analyse auraient pu être les suivantes :

1 Préparation des données
- 1.1 supprimer (et donc identifier) les colonnes contenant trop de non-réponses
- 1.2 supprimer les colonnes contenant des identifiants uniques
- 1.3 produire et visualiser des statistiques descriptives
- 1.4 supprimer les colonnes avec très peu de variance ou valeur unique
- 1.5 normaliser les variables quantitatives
- 1.6 créer une base d'apprentissage et une base de test selon une proportion choisie

2 Création et utilisation du modèle
- 2.1 Choisir et appliquer un algorithme
- 2.2 Évaluer le modèle
- 2.3 Optimiser le modèle
- 2.4 Visualisations et interprétations des résultats
- 2.5 Calculer une prédiction

À noter que les étapes 1.4 et 1.6 ne sont pas nécessaires si on utilisait *Random Forest*.

En outre, une série d'actions en sus de l'analyse était nécessaire et importante dans l'évaluation. Il fallait d'abord rédiger l'analyse dans *Rmarkdown*. *Markdown*[1] un langage à balise permettant de produire des documents structurés. On peut le comparer à du *wikicode*, ou a une simplification du HTML intégrant des éléments de *Latex*. *Rmarkdown*[2] est une implémentation du *Markdown* permettant d'intégrer directement dans un unique document des scripts R. On produit par ce moyen des fichiers dans différents formats (natifs, HTML, PDF, etc.). Il fallait ensuite déposer un fichier HTML (produit à partir du *Rmarkdown*) et le fichier *Rmarkdown* « source » dans un répertoire du site *Github*[3] et déposer le lien vers ce répertoire sur la plateforme du Mooc.

Grille d'analyse

On a regardé 75 travaux déposés par des participants selon les 10 critères ci-dessous. Ces critères ont été établis après avoir examiné 26 copies dans une perspective inductive inspirée par la *Grounded Theory* (Glaser & Strauss, 2010).

Deux items portent sur la communication des résultats, il s'agit d'élément important puisque le barème d'évaluation donnée 10 points sur 21 selon les qualités d'exécutions de ces critères.

- Qualité de la mise en page (+/~/-). Cet item prend en compte le degré de complexité du fichier *Rmarkdown*. Y a-t-il eu une utilisation de fonctions autres que celle de titre ?
- Utilisation avancée de *Github* (oui/non) : y a-t-il un lien permettant de lire directement le fichier *HTML* ?
- Qualité de la rédaction (+/~/-) : est-ce que le texte dit quelque chose ? On a remarqué que la quantité et la qualité du texte sont fortement liées.

Le *machine learning*, comme tout travail statistique, requiert des tâches de nettoyages et de préparations des données faisant souvent appel à de la visualisation, les deux items ci-dessous cherchaient à caractériser ces éléments.

- Qualité de la préparation des données (+/~/-), les variables sont-elles normalisées ? Y a t – il eut suppression des variables avec peu de variances ?
- Quelle bibliothèque graphique est employée s'il y a des graphiques (base, ggplot, rattle, lattice…) et qualités des visualisations (+/~/-), les visualisations sont-elles basées sur des sorties par défaut ou véritablement travaillées ?

L'exécution de l'analyse fait appelle à des compétences propres à *R* comme les fonctions et les bibliothèques de fonctions.

- Quel(s) algorithme(s) de *machine learning* est/sont employé(s) : randomforest, gbm, glm, CART, lda…
- Quelle bibliothèque de R est convoqué pour appliquer la fonction de machine learning indiquée à la variable précédente ? Cet item est important, car, dans R, une bibliothèque de fonction dédiée à un algorithme offre plus d'options par rapport aux bibliothèques de fonctions qui regroupent des algorithmes comme CARET.

Caractériser les compétences en programmation et en statistique a été réaliser grâce aux deux items ci-dessous.

1 http://daringfireball.net/projects/markdown
2 http://rmarkdown.rstudio.com
3 https://github.com

- Lisibilité du script *R* (+/~/-) : le script est-il lisible ? Y a-t-il des boucles ou l'utilisation des fonctions *apply* plutôt que des répétitions ? Cet élément dénote d'une maîtrise dans les compétences en programmation.
- Originalité de l'analyse (+/~/-). Est-ce que l'analyse tire parti des règles pour s'en affranchir ? Y a-t-il une volonté d'optimisation du modèle ? Une comparaison entre algorithmes ? L'utilisation d'un algorithme peu fréquent (autre que CART ou Random Forest) ?

Table 1 Exemple des résultats de dépouillement. Les signes "+", "~" et "-" sont dans l'ordre des degrés de qualité, "." pour l'absence de réponse.

n°	mise en page	html github	rédaction	préparation	library graphique	graphiques	algo	library algo	lisibilité scripts	Originalité
50	~	oui	+	+	base	~	rf	randomForest	+	~
71	~	non	-	+	base	~	rf	CARET	~	~
79	~	non	+	+	.	.	knn	CARET	+	+

Résultats

Élément périphérique à l'analyse

Rmarkdown est presque toujours utilisé de façon très élémentaire. La mise en page était plutôt le produit d'une configuration par défaut à de rares exceptions près, qui ont par exemple ajouté une table des matières (2 copies sur 75). La consigne de dépôt de deux fichiers a été plutôt suivie : on trouve 4 copies sur 75 qui n'ont pas fait le travail, mais ont tout de même déposé une URL.

La rédaction accompagnant les scripts est minimaliste avec seulement 7 devoirs sur 75 que l'on peut considérer comme véritablement explicatifs. Les modes de programmations respectent les canons présentés dans le cours, on trouve ainsi seulement 7 devoirs pour lesquels on a considéré que le script était difficile à lire. Par ailleurs, les fonctions de classification automatique sont gourmandes en ressources matérielles et les scripts peuvent être longs à s'exécuter. C'est pourquoi, sans être majoritaire, le recours au calcul parallèle a été assez souvent convoqué.

Machine learning

L'algorithme le plus utilisé (65 devoirs sur 75) est celui des forêts d'arbre décisionnel (*Random Forest*). Probablement, car il produit les taux de mal classé les plus bas. Il permet par ailleurs de s'abstenir de réaliser un certain nombre d'actions de nettoyage. On remarque qu'un seul participant a obtenu de résultats comparables avec la méthode des K plus proches voisins (KNN). Le deuxième algorithme le plus utilisé est CART (intitulé *rpart* dans R) souvent employé en comparaison avec *Random Forest*. On a remarqué très peu d'utilisation d'algorithmes n'ayant pas été présentés dans le cours comme C5.0, KNN, SVM ou de réseaux de neurones. La bibliothèque de fonction la plus utilisée est celle présentée dans le cours : CARET (52 sur 75). Il y a peu d'utilisation des bibliothèques liées aux algorithmes per se. Les bibliothèques utilisées le plus souvent sont *Random Forest* et *rpart* (23 devoirs sur 75).

On trouve globalement peu d'originalité dans le travail de modélisation statistique réalisé. 9 rendus sur 75 ont été considérés comme original c'est-à-dire s'écartant ou tirant profit du canevas présenté ci-dessus. Le fonctionnement des algorithmes n'est quasiment jamais présenté (1 devoir sur 75 y fait allusion) tout comme l'interprétation des résultats qui n'est presque jamais réalisée (2 copies sur 75). En revanche, le travail sur la préparation des données est plutôt soigné et c'est un élément qui avait été bien traité dans le cours.

On trouve une erreur fréquente qui consiste à réaliser un test de validation croisé avec Random Forest alors que c'est une opération redondante. Dans le même registre, il n'est pas nécessaire avec *Random Forest* de supprimer les variables ayant peu de variances ou une valeur unique alors que les participants l'ont fait de façon quasi systématique. En revanche, discuter le nombre d'arbres à produire dans la forêt avait du sens d'un point de vue pratique, mais ça n'a été que très peu fait.

Rendus typiques

L'impression générale qui ressort de l'ensemble des rendus est la prédominance d'une uniformité dans les devoirs. Le schéma typique d'un rendu est le suivant :

1. Titre, nom, date
2. Copier-coller du paragraphe de présentation des données figurant sur le Mooc
3. Chargement de la bibliothèque CARET
4. Importation des données
5. Tris dans les variables : suppression des NA, des variables d'identifiants et des variables avec peu de variance
6. Création d'une base d'apprentissage dont les proportions peuvent aller de 10/90 à 90/10
7. Application de Random Forest avec les paramètres par défaut de CARET
8. Matrice de confusion renvoyée par la fonction de CARET
9. Prédiction sur les 20 individus fournis par le cours

Des variations notables par rapport à ce canevas sont bien entendues fréquentes et d'abord dans la structure de cette liste. Peut intervenir par exemple des sections comme la visualisation des données après l'importation, différent calcul des taux d'erreurs de classement, plus de soin accordé à la création de la base d'apprentissage ou dans l'examen des critères d'efficiences.

On trouve ensuite des variations dans chacune des sections. Le paragraphe de présentation est le plus souvent un copier/coller de la consigne, on trouve plus rarement une rédaction personnalisée. Les données sont souvent importées directement depuis le web avec mention qu'il s'agit d'un souci de reproductibilité. De même lorsqu'un algorithme requiert des tirages au sort, comme c'est le cas pour le *boosting* (dont *Random Forest*), une « graine est fixée » permettant de reproduire sur une autre machine le même tirage au sort et obtenir les mêmes résultats.

Lorsque le participant fait appel à des graphiques, ce sont le plus souvent ceux produits par défaut par une bibliothèque. On remarque très peu de variété dans les types de graphiques. Les représentations les plus fréquentes sont : arbre CART, corrélation plot, représentation des variables importantes dans *Random Forest* et graphique de corrélation. Seul un rendu a montré un véritable travail de visualisation de données.

Lors de la préparation des données, on peut trouver une normalisation des variables quantitatives selon une technique présentée dans le cours, un devoir (numéroté 57) applique même une technique plus sophistiquée que celle du cours. On trouve souvent une comparaison de l'efficience de différents algorithmes. Enfin, on rencontre moins souvent une comparaison de l'efficacité d'un algorithme (le plus souvent *Random Forest*) avec différents paramètres (principalement le nombre de variables à essayer dans la construction des arbres, paramètre intitulé *mtry* dans R), c'est bien entendu employé pour trouver une application optimale, c'est-à-dire en cherchant à minimiser les taux de mal classés.

Conclusion et perspectives

L'uniformité générale des copies est le premier trait marquant de l'ensemble des copies examiné. Il faut souligner l'utilisation par la majorité des participants des forêts d'arbre de classification (65 devoirs sur 75). La bibliothèque CARET, qui regroupe plusieurs algorithmes de machine learning, mais qui fournit moins d'options que les bibliothèques dédiées, est la plus utilisée avec 52 devoirs qui la convoque.

L'exercice à réaliser n'est pas difficile, mais peu de participants le font véritablement de façon complète. Une raison possible réside dans le barème, la moitié des points porte sur la capacité à utiliser *Rstudio* et *Github*. Probablement aussi parce que regarder toutes les vidéos et faire tous les quizz ne suffit pas pour en être capable. Il faudrait des éléments socialisants qui contraignent à des interactions et à la formulation des actions, c'est d'ailleurs le rôle du forum. Il faudrait donc regarder les interventions sur le forum des participants ayant le mieux réalisé l'exercice.

Pour aller plus loin, on pourrait aussi automatiser la méthode notamment en réalisant des travaux sur les fichiers *Rmarkdown* qui constituent des documents structurés dans lesquels figurent aussi bien les scripts R que les textes. Il serait surement profitable de reprendre le corpus et la méthode et réaliser un codage totalement inductif avec un logiciel de traitement qualitatif de données comme celui basé sur R et développé par Huang (2011). Enfin, un regard sur dans les données démographiques et dans les logs l'activité des participants ayant des copies remarquables pourrait permettre une mise en perspective éclairée.

Bibliographie

Balfour, S. P. (2013). Assessing Writing in MOOCs : Automated Scoring and Calibrated Peer Review. *Research and Practice in Assessment*, 8(Summer), 40–48. http://www.rpajournal.com/dev/wp-content/uploads/2013/05/SF4.pdf

Breiman, L. (2001), Random Forests, Machine Learning 45(1), 5-32.

Chinn, D. (2005). Peer Assessment in the Algorithms Course. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (p. 69–73). New York, NY, USA : ACM.

Falchikov, N., & Goldfinch, J. (2000). Student Peer Assessment in Higher Education : A Meta-Analysis Comparing Peer and Teacher Marks. *Review of Educational Research*, 70(3), 287-322.

Gašević, D., Kovanović, V., Joksimović, S., & Siemens, G. (2014). Where is Research on Massive Open Online Courses Headed ? A data analysis of the MOOC Research Initiative. The International Review of Research in Open and Distance Learning. 15, (5).

Glaser, B. G., & Strauss, A. L. (2010). La découverte de la théorie ancrée: Stratégies pour la recherche qualitative. Paris: Armand Colin.

Hastie, T., Tibshirani, R., & Friedman, J. (2011). The Elements of Statistical Learning : Data Mining, Inference, and Prediction, Second Edition (2nd ed. 2009. Corr. 7th printing 2013 edition). New York, NY : Springer.

Huang, R. (2011). RQDA: R-based Qualitative Data Analysis. R package version 0.2-2. URL http://rqda.r-forge.r-project.org/.

Kulkarni, C., Wei, K. P., Le, H., Chia, D., Papadopoulos, K., Cheng, J., Klemmer, S. R. (2013). Peer and Self Assessment in Massive Online Classes. *ACM Trans. Comput.-Hum. Interact.*, 20(6), 33 :1–33 :31.

Lantz, B. (2013). Machine Learning with R. Birmingham : Packt Publishing.

Piech, C., Huang, J., Chen, Z., Do, C., Ng, A., & Koller, D. (2013). Tuned models of peer assessment in MOOCs. *arXiv preprint* arXiv :1307.2579. http://arxiv.org/abs/1307.2579

R Core Team. (2015). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. Consulté à l'adresse https://www.R-project.org

Sadler, P. M., & Good, E. (2006). The impact of self-and peer-grading on student learning. *Educational assessment*, 11(1), 1–31. http://www.tandfonline.com/doi/abs/10.1207/s15326977ea1101_1

Saito, H., & Fujita, T. (2009). Peer-Assessing Peers' Contribution to EFL Group Presentations. *RELC Journal*, 40(2), 149-171.

Thomson, P., Smith, A., & Annesley, S. (2014). Exploration of the effects of peer teaching of research on students in an undergraduate nursing programme. *Journal of Research in Nursing*.

Topping, K. (1998). Peer Assessment Between Students in Colleges and Universities. *Review of Educational Research*, 68(3), 249-276.

Walsh, T. (2014). The PeerRank Method for Peer Assessment. http://arxiv.org/abs/1405.7192