Sorbonne Université - Faculté des Sciences

PROJET PLDAC
Détection de revues spams

Auteurs :
GIANG Cécile
LENOIR Romain

Encadrants :
BASKIOTIS Nicolas
GUIGUE Vincent

11 octobre 2021

Table des matières

1 Introduction

Alors que le e-commerce connaît un essor fulgurant depuis une dizaine d'années, les consommateurs doivent eux faire face à un problème de taille : faire le bon choix lorsque l'on souhaite acheter un produit en ligne peut parfois s'avérer très difficile lorsque l'on se retrouve face aux millions de produits proposés par les catalogues de grandes plateformes de e-commerce telles que Amazon ou eBay. Afin de guider leur choix, une très grande majorité de futurs acheteurs choisi de se tourner vers les avis laissés par d'autres consommateurs, jouant alors une part importante dans l'acte d'achat. Ainsi d'après le baromètre *Ifop* pour *Opinion System*, spécialiste des avis clients contrôlés, 87% des internautes confirment consulter d'anciennes revues consommateurs pour guider leur choix. Pourtant, un tiers d'entre elles seraient fausses, selon Jean-David Lépineux, dirigeant de l'organisme existant depuis dix ans.

En effet, face à l'intérêt porté à l'opinion utilisateur, il devient très tentant pour les entreprises d'essayer de biaiser l'évaluation des consommateurs en faisant appel à des spammeurs payés pour introduire sur ces plateformes des fausses revues (*spam reviews* ou *fake reviews* en anglais). Leur but est de promouvoir leurs propres produits, ou au contraire de nuire à un produit concurrent.

Les plateformes de e-commerce touchées par cette pratique de plus en plus fréquente perdent peu à peu la confiance de leurs utilisateurs ; il devient alors primordial pour elles de rechercher constamment de nouvelles techniques de détection de revues spams afin de pouvoir garantir la fiabilité des avis consommateurs qu'elles proposent.

Ce problème de détection de fausses revues est bien plus difficile qu'il n'y paraît au premier abord. La plus grande difficulté réside en particulier dans le fait que, contrairement aux courriels spams, la plupart de ces revues spams est écrite par de vraies personnes, exerçant parfois de manière professionnelle : il devient alors très difficile voire presque impossible de distinguer une revue spam d'un avis honnête sur la base seule du langage. Pour vous en convaincre, voici une revue tirée du site de Bing Liu :

« This movie starring big names - Tom Hanks, Sandra Bullock, Viola Davis, and John Goodman - is one of the most emotionally endearing films of 2012. While some might argue that this film was "too Hollywood" and others might see the film solely because of the cast, it is Thomas Horn's performance as young Oskar that is deserving of awards. »

Si au premier abord rien dans ce commentaire ne nous alarme, il s'agit en réalité d'un faux avis : nous nous rendons ainsi compte que l'étude seule du contenu d'une revue n'est pas suffisante pour résoudre notre problème.

L'objectif de ce projet est ainsi de mettre en place un système de détection de revues frauduleuses. Les années précédentes, nos prédécesseurs s'étaient proposés d'implémenter une technique de détection de revues spams par graphe de défiance utilisateur-produit-revue, basée sur l'article *Review Graph based Online Store Review Spammer Detection* [3]. Cette méthode présentait le désavantage de ne jamais tenir compte du contenu textuel des revues et donc de passer à côté d'informations particulièrement précieuses. Nous nous proposons ainsi de reprendre cette technique de détection par graphe et de la complèter avec du traitement automatique du langage afin d'affiner la détection de fausses revues.

2 Données d'analyses et premiers traitements

La mise en place d'un classifieur de revues spams par apprentissage supervisé soulèverait de nombreuses difficultés, dont la plus problématique serait très certainement de trouver une base d'apprentissage complète et adaptée : comme nous l'avons souligné plus haut, le fait que ces fausses revues soient écrites par des personnes réelles rend presque impossible de jauger avec certitude l'honnêteté d'un avis sur la base seule de son contenu textuel. C'est la raison pour laquelle il existe si peu de bases de revues étiquetées, et ce malgré l'importance de la tâche pour les grandes plateformes de e-commerce. Ce n'est pas le seul inconvénient posé par l'apprentissage supervisé : celui-ci s'effectuant sur des avis écrits en langue vivante, un tel classifieur deviendrait très vite obsolète car sensible aux changements de styles d'écriture et aux néologismes. Le problème de la mise à l'échelle se pose également : pour traiter le problème de détection de revues spams en langue étrangère, il nous faudrait ré-apprendre sur une toute nouvelle base d'avis étiquetée par des experts. De même, changer le contexte du problème de détection *(passer par exemple d'avis sur des produits électroniques à des revues sur des restaurants)* supposerait aussi un nouvel apprentissage.

Le manque de flexibilité des méthodes supervisées nous a poussé à choisir de traiter le problème par une approche non-supervisée : nous souhaitons ainsi que notre système reste cohérent quel que soit le problème traité.

2.1 Données utilisées

Nous avons choisi pour ce projet de nous appuyer sur le fichier *Cellphones and accessories* fourni par J. McAuley et J. Leskovec [2]. Il s'agit d'une base non-étiquetée de revues Amazon sur des produits électroniques liés aux téléphones. Ce fichier contenant plus de 10 millions d'avis, nous décidons d'en extraire 60 000 revues concernant des produits contenant au moins 10 avis, afin de faire tenir nos données en mémoire.

2.2 Format des données et stockage

Le fichier *Cellphones and accessories* est un fichier au format JSON. Chaque ligne contient une revue sous forme de dictionnaire : les clés sont les descripteurs d'une revue *(note, achat vérifié, date de publication, identifiant utilisateur, identifiant produit, pseudo utilisateur, contenu textuel de la revue, titre de la revue, temps au format UNIX, nombre de votes, image, style)* et les valeurs du dictionnaire sont les valeurs de ces descripteurs.

Simplement charger les données telles quelles serait bien trop coûteux en mémoire : l'information sur les descripteurs (clés) est redondante. Nous choisissons donc de stocker les valeurs de description d'une revue dans un array appelé `data` où chaque revue est une liste de valeurs, et de conserver une unique liste de descripteurs `fields`. Retrouver le descripteur correspondant à la 3e information sur une revue de `data` revient donc à chercher la valeur de `fields` à l'indice 2.

```
{ 'overall': 5.0,
  'verified': True,
  'reviewTime': '02 8, 2007',
  'reviewerID': 'AKQOGUH6BOU9X',
  'asin': 'B000BT8UAW',
  'reviewerName': 'Hayman33',
  'reviewText': 'Works just fine! Delivery was earlier than predicted. I am well satisfied. I looked
                 everywhere in two towns to buy such a simple thing but they didn\'t have them. I was
                 told "they can order one". I thought to myself.......so what, so can I. This was much
                 easier. I ordered it one evening while wearing my pajamas and it came right to my door
                 ----no muss, no fuss. This is the way to buy such items! Thanks for a great job.',
  'summary': 'Motorola V170 battery',
  'unixReviewTime': 1170892800,
  'vote': '3',
  'image': None,
  'style': None }
```


FIGURE 1 – Exemple d'une revue de la base de départ

2.3 Pré-traitement : suppression des doublons non-spams

- **Champs de données vides :** Certaines revues présentent des données manquantes (pseudo utilisateur, nombre de votes, etc...). Dans le cadre de notre projet, seuls les descripteurs correspondant à l'identifiant produit, l'identifiant utilisateur et la date de publication nous sont nécessaires. Toute revue ne présentant pas l'une de ces informations est supprimée de la base.

- **Doublons non-spams :** Une rapide analyse de nos données nous indique que certaines revues ont été dupliquées dans la base : ce sont des avis écrits par un même auteur et concernant un même produit. Ces revues ne sont pas l'oeuvre d'utilisateurs malintentionnés mais le résultat de casses lors de fusion de différentes versions d'un même article par Amazon, voire simplement d'une mauvaise gestion des avis utilisateurs de leur part. Nous choisissons donc de les supprimer. A noter que certaines revues sont identiques, mais proviennent d'un même auteur pour des produits différents ou concernent un même produit mais sont écrites par des auteurs différents. Ces avis, sous la condition qu'ils excèdent une certaine longueur, sont très certainement les oeuvres de spammeurs ; nous les conservons donc.

3 Techniques simples de détection de spams

Dans cette section, nous nous intéressons de près à des techniques de détection de *fake reviews* assez intuitives et simples à mettre en place, et qui nous permettent d'obtenir rapidement une première mesure sur le caractère frauduleux d'une revue. Bien que ces techniques soient assez préliminaires et ne permettent pas de prendre en compte les relations qui lient deux revues (revues écrites par un même auteur, ou concernant un même produit), elles présentent l'avantage de tirer profit d'observations temporelles sur un groupe de revues donné, et d'observations comportementales sur les utilisateurs. Les techniques implémentées sont inspirées des travaux de Geli FEI et al. dans leur article *Exploiting Burstiness in Reviews for Review Spammer Detection* [1]

3.1 Détection de bursts de revues

Il est intéressant de noter qu'un spammeur agit très rarement seul : une entreprise cherchant à vendre son produit ou à nuire à un concurrent fera en règle générale appel aux services de plusieurs spammeurs, et non d'un spammeur isolé. Ainsi une bonne façon de détecter une potentielle période d'attaque sur un produit serait de détecter une arrivée massive et inhabituelle de revues, phénomène que nous appellerons par la suite **burst de revues**. Nous pourrons ainsi conjecturer qu'une fenêtre temporelle pendant laquelle le nombre de revues sur un certain produit est anormalement élevé correspond très certainement une période d'attaque d'un groupe de spammeurs.

Voici par exemple l'histogramme du nombre de revues postées chaque mois pour un amplicateur GSM.

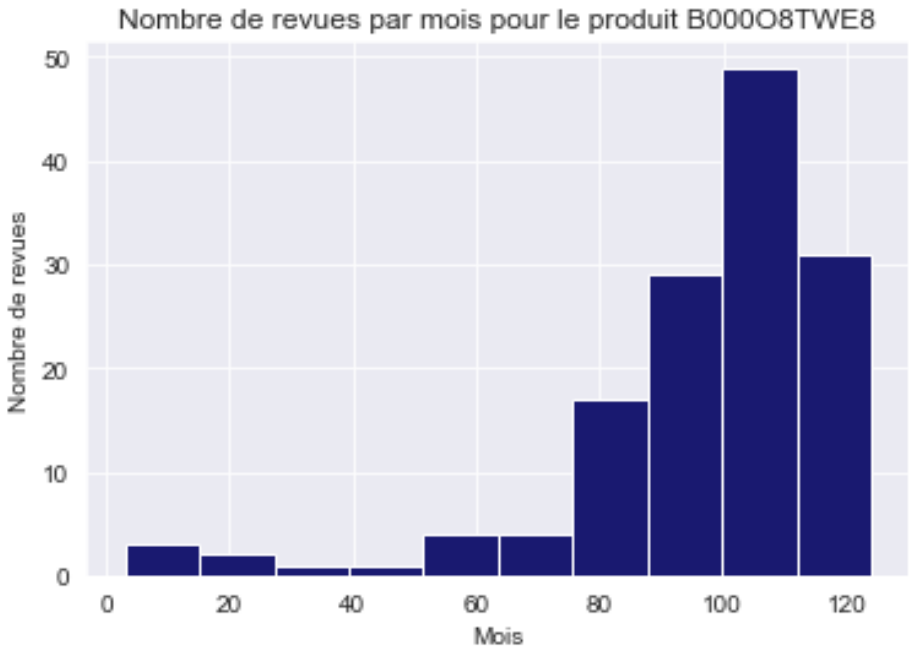


FIGURE 2 – Histogramme du nombre de revues par mois pour un amplicateur GSM

Nous pouvons très facilement remarquer une arrivée massive de revues entre les mois 90 et 120. Ce burst soudain de revues pourrait être lié à l'activité d'un groupe de spammeurs postant un nombre important de fausses revues pendant cette période. Mais si ce n'était pas le cas ? Et si cette augmentation soudaine de revues était en fait le fruit d'une campagne marketing réussie ? Et si l'entreprise vendant ce produit a tout simplement commencé à fournir un client ? Nous nous rendons très vite compte que prendre seulement en considération l'aspect temporel de ces revues ne nous permet pas réellement de tirer des conclusions quant à la défiance d'une revue ou d'un groupe de revues.

3.2 Déviations de notes

Comme nous l'avons déjà souligné, une entreprise s'offrant les services de spammeurs a pour objectif général de promouvoir son produit, ou bien au contraire de nuire à un produit concurrent. Cette information nous permet de déduire deux comportements de notation qui les caractérise :

- si un spammeur cherche à nuire à un produit, il aura tendance à lui donner une note bien inférieure à sa note moyenne actuelle
- si par contre il cherche à promouvoir un produit, le spammeur lui donnera plutôt une note très supérieure à sa note moyenne actuelle

Nous introduisons ainsi la notion de **déviation de notation** (*rating deviation* en anglais), qui quantifie l'écart d'une note donnée par un utilisateur à la note moyenne sur ce produit. Dans les faits, il est bien plus cohérent de comparer cette note à la note moyenne du produit sur une fenêtre temporelle : nous prenons en compte le fait qu'un produit peut s'améliorer dans le temps.

En notant $deviation(r)$ la déviation à la moyenne d'une revue r écrite au temps t, et en considérant une fenêtre de temps δ, on a donc :

$$deviation(r) = \|rating(r) - average\_rating(t, \delta)\|$$

Nous considérons ainsi que la déviation d'une revue est significative (i.e. qu'elle est dénote d'un comportement de notation suspect) si elle dépasse un certain seuil s. Une telle revue sera dite **déviante**. Ce seuil dépendant du produit considéré, il n'est pas possible de le fixer à l'avance. Nous choisissons donc d'afficher un histogramme sur les déviations de l'ensemble des revues pour un produit p afin de nous aider à choisir un seuil adapté.

En reprenant l'exemple de l'amplificateur GSM, l'histogramme sur les déviations nous montre qu'une différence de plus de 2.25 étoiles à la moyenne devient suspecte. Nous choisissons donc cette valeur comme seuil s et affichons l'évolution dans le temps de la moyenne mobile sur le nombre de revues dont la déviation est suspecte.

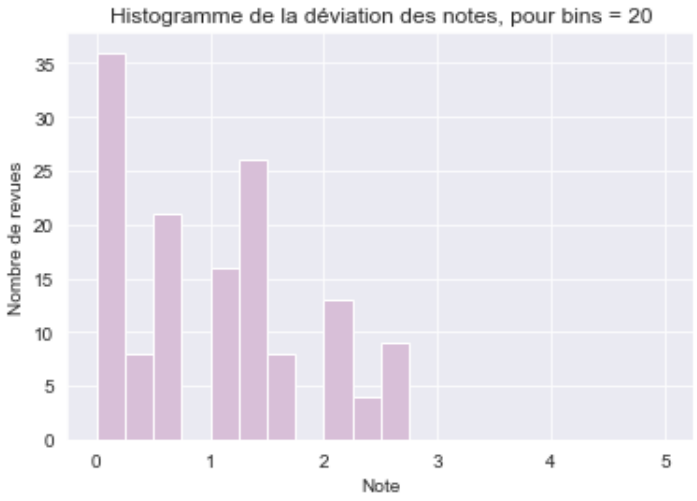


FIGURE 3 – Histogramme sur les déviations pour l'amplificateur GSM

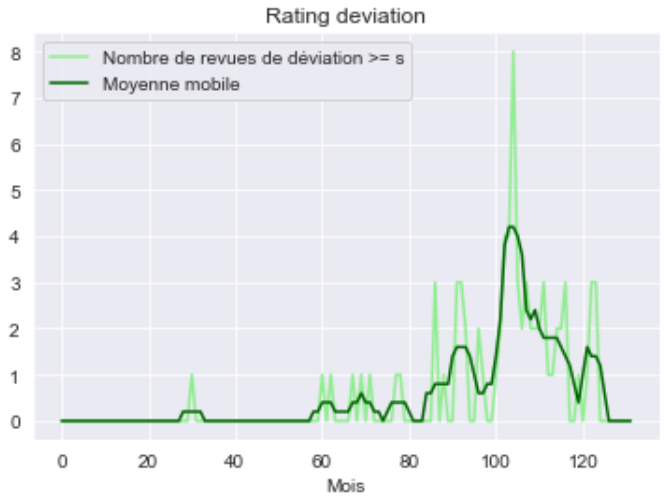


FIGURE 4 – Moyenne mobile du nombre de revues déviantes pour s = 2.25

Dans la figure 4, nous notons qu'un nombre anormalement élevé de revues dévie fortement de la moyenne. Cela vient confirmer la suspicion que nous avions soulevée à partir de l'analyse des bursts de revues : il semblerait bien qu'entre les mois 90 et 120, l'on ait affaire à une attaque de spammeurs.

Cette technique de détection de *fake reviews* par déviation de notation a un grand défaut : elle considère toute notation s'écartant un peu trop de la moyenne comme suspecte, or il est tout à fait possible que l'auteur d'une revue déviante soit tout à fait honnête et que son opinion diverge simplement de la moyenne. A titre d'exemple, sur les 141 revues composant la base d'avis pour l'amplificateur GSM, près de 23 dévient beaucoup de la notation moyenne, ce qui représente tout de même 16% des revues !

Combiner les techniques de détection de burst temporel et de déviation de notation est un bon moyen de palier aux faiblesses de ces deux techniques. Ainsi, nous pourrions considérer qu'une revue se trouvant dans un burst temporel et qui dévie largement de la moyenne mobile est très certainement une revue spam. Sur l'exemple de l'amplificateur GSM, nous réussissons ainsi à détecter 14 revues satisfiant les deux heuristiques. Ces revues sont représentées par des points rouges dans la figure 6.

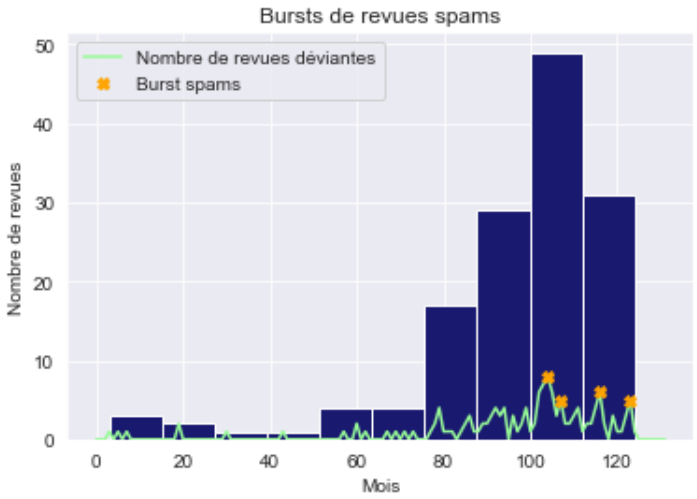


FIGURE 5 – Burst de reviews et déviation

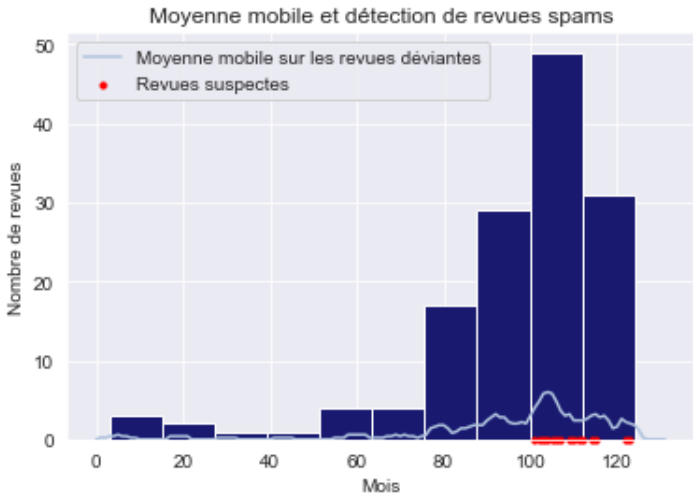


FIGURE 6 – Moyenne mobile et revues suspectes

3.3 Revues en doublons

Pour terminer, un dernier indicateur que nous pouvons exploiter est la présence de **doublons dits spams** dans notre base d'avis. En effet, si les revues en doublon écrites par un même auteur pour un même produit ont été enlevées de la base lors de la phase de pré-traitement, les doublons provenant d'auteurs différents ou écrits pour des produits différents sont, eux, une mine d'information. En effet, il n'est pas rare pour un spammeur de tout simplement copier-coller son commentaire d'une revue à une autre. Nous considérerons ainsi de telles revues comme fortement suspectes, et nous les appellerons **doublons spams** par la suite.

4 Graphe de défiance : algorithme PageRank

Les deux heuristiques sur les bursts temporels de revues et les déviations de notation sont très utiles pour une analyse rapide d'un jeu de données, mais présentent tout de même un inconvénient majeur : elles n'exploitent à aucun moment les relations qui lient les revues, les auteurs et les produits entre eux, passant à côté d'une source d'informations non négligeable.

Dans cette section, nous nous intéressons donc à la mise en place d'un algorithme de détection de revues spams par diffusion dans un graphe de défiance revue-auteur-produit, proposée par Guan WANG et al. dans leur article *Review Graph based Online Store Review Spammer Detection* [4].

4.1 Intuition de l'algorithme

L'algorithme PageRank de diffusion de la défiance proposé par Guan WANG et al. a pour vocation d'exploiter les relations liant les revues, les auteurs et les produits entre eux, et ainsi de passer outre les limitations des techniques purement basées sur les observations temporelles, comportementales ou sur les techniques linguistiques. Il repose en réalité sur un principe très simple : il faut accorder moins de crédibilité à un produit évalué par un spammeur, à une revue concernant un produit attaqué, à l'auteur d'une revue suspecte.

Guan WANG et al. proposent ainsi une toute nouvelle vision de notre problème de détection de revues spams. Les revues, les produits et les utilisateurs sont désormais vus comme des noeuds dans un graphe où chaque arc représente une relation auteur-revue, auteur-produit ou revue-produit. L'article introduit trois nouvelles mesures permettant d'estimer le niveau de défiance de chaque entité (chaque noeud) : la *confiance utilisateur*, *l'honnêteté revue* et *la fiabilité produit*, que nous détaillerons dans la sous-section suivante.

FIGURE 7 – Graphe de diffusion de la défiance selon Guan et al.

4.2 Scores de défiance

Les trois nouvelles mesures introduites permettent d'estimer le niveau de défiance de chaque noeud dans le graphe de défiance :

- **la confiance utilisateur :** permet de juger à quel point un utilisateur est digne de confiance. Intuitivement, nous nous rendons compte que si nous disposons des scores d'honnêteté sur l'ensemble des revues écrites par un utilisateur, il nous sera possible de calculer son score de confiance : plus un auteur écrit des revues honnêtes, plus il sera digne de confiance. Nous pouvons également relever d'autres critères qui nous permettront de définir un critère de défiance sur des auteurs :
  - la confiance utilisateur ne doit pas dépendre du nombre de revues qu'il a écrites mais de la somme de leur score d'honnêté
  - le score de confiance doit être proche de -1 pour un auteur suspect, et proche de 1 pour un auteur honnête
  - la confiance utilisateur ne doit pas croître et décroître linéairement, sans aucune considération sur le score de ses revues : une revue honnête ou suspecte doit faire augmenter/diminuer le score de confiance de manière plus forte si ces revues sont isolées, et moins forte si elles sont nombreuses

  Ces critères permettent de définir une mesure de confiance sur nos auteurs que nous noterons T (*trustiness*). Ainsi pour un utilisateur u dont l'ensemble des revues est noté R_u :

$$T(u) = \frac{2}{1 + e^{-H_u}} - 1 \quad \text{où} \quad H_u = \sum_{r \in R_u} H(r)$$

  La fonction de confiance ainsi définie pour un utilisateur est en fait une fonction logistique variant selon l'honnêté de ses revues :

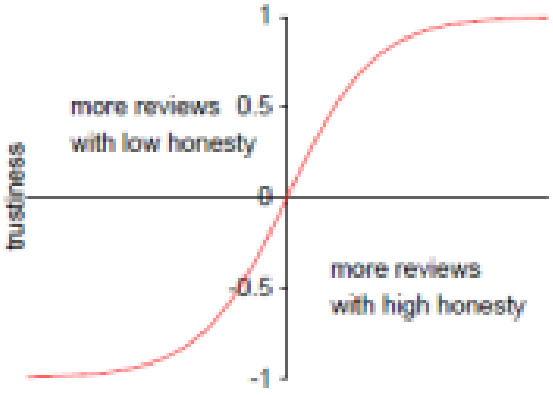


FIGURE 8 – Fonction confiance utilisateur selon l'honnêteté de ses revues

- **l'honnêteté revue :** mesure le degré d'honnêteté d'un avis laissé sur un produit. Nous voyons assez vite qu'elle dépend très fortement de la fiabilité du produit auquel elle fait référence *(nous aurons tendance à croire les revues concernant des produits fiables)*, ainsi que de son entente avec les revues concernant le même produit *(nous aurons tendance à croire les revues qui ne dévient pas trop de l'opinion générale)*. Il est à noter qu'une revue spam pourrait très bien être en accord avec d'autres revues si elles sont également frauduleuses.

  Nous définissons ainsi le score d'entente A comme une fonction sur les confiances utilisateurs. Ainsi pour une revue r et avec U_a l'ensemble des utilisateurs dont l'opinion concorde avec r sur le produit, et U_d l'ensemble des utilisateurs dont l'opinion diverge de r :

  $$A(r) = \sum_{u_a \in U_a} T(u_a) - \sum_{u_d \in U_d} T(u_d)$$

  Nous pouvons noter qu'avec une telle formulation, si r est en concordance avec des utilisateurs de confiance alors son score d'entente sera positif. Il sera par contre négatif dans le cas où r est en accord avec des utilisateurs peu fiables (score de confiance négatif). A partir de là, ils nous est possible de définir le score d'honnêté de la revue r concernant le produit p :

  $$H(r) = \|R(p)\| \left(\frac{2}{1 + e^{-A(r)}} - 1 \right)$$

  Nous nous retrouvons là encore avec un score de défiance compris entre -1 et 1.

- **la fiabilité produit :** estime si un produit est fiable. De façon intuitive, nous voyons bien que cette mesure dépend des notes données par des utilisateurs de confiance : s'ils ont tendance à lui attribuer une bonne note, alors le produit est certainement fiable. Si par contre ils lui attribuent généralement des notes basses, il y a fort à parier que ces produits ne sont pas dignes de confiance. Sachant μ la note moyenne sur le produit p, nous définissons la fiabitité R de p de la manière suivante :

  $$R(p) = \frac{2}{1 + e^{-\theta}} - 1 \quad \text{où} \quad \theta = \sum_{u : T(u) \geq 0} T(u)(rating_p - \mu)$$

4.3 Formalisation de l'algorithme

Algorithm 1 Iterative Computation Framework

Input: The set of store $\mathscr{S}$, review $\mathscr{RS}$, and reviewer $\mathscr{R}$
The agreement time window size Δt, review similarity threshold δ, and the $round$ of iterations
Output: The set of $reliability$ R, $honesty$ H, and $trustiness$ T
// Initialization step
Initialize store's reliability and reviewer's trustiness to 1, compute review's agreement using these initial values and Δt, δ according to (11) and (12)
$roundCounter = 0$
while $roundCounter < round$ **do**
  for $re \in \mathscr{RS}$ **do**
    compute $H(re)$ using (13)
  end for
  for $r \in \mathscr{R}$ **do**
    compute $T(r)$ using (6)
  end for
  for $s \in \mathscr{S}$ do
    compute $R(s)$ using (14) and (15)
  end for
  for $re \in \mathscr{RS}$ do
    update re's agreement using new $\mathscr{T}$ based on (11) and (12)
  end for
  $roundCounter$++
end while

FIGURE 9 – Algorithme PageRank pour la diffusion dans un graphe de défiance

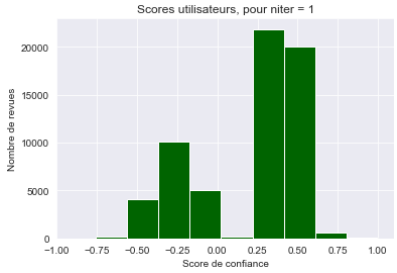
Le pseudo-code ci-dessus décrit l'algorithme PageRank tel que proposé par Guan WANG et al. dans leur article de référence. Il est composé de deux étapes clés :

- **Initialisation :** L'article original propose d'initialiser les scores utilisateurs et produits à 1, et les scores revues selon le score d'entente. Ainsi à l'initialisation, une revue aura un score d'honnêteté faible si sa note va à l'encontre de l'opinion générale.

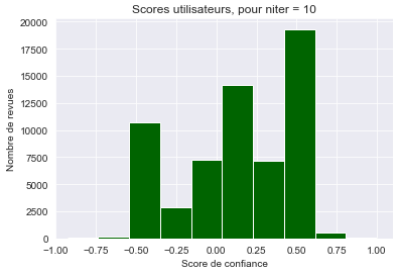
  Nous souhaitons intégrer à cette étape les heuristiques de détection de bursts de revues et de déviation de notation que nous avons déjà mises en place afin de tirer profit de nos connaissances *a priori*. Nous tentons donc de nuancer l'initialisation des scores de revues en mettant à -1 celles qui se trouvent à la fois dans une période de burst et dont la note dévie largement de la moyenne, à -0.75 les avis qui sont soit en burst, soit déviants, et au score d'entente le reste des revues.

- **Itérations :** Les scores revues, utilisateur et fiabilité sont calculés les uns après les autres selon l'état courant du graphe. La convergence arrive très vite : une dizaine d"itérations suffisent.

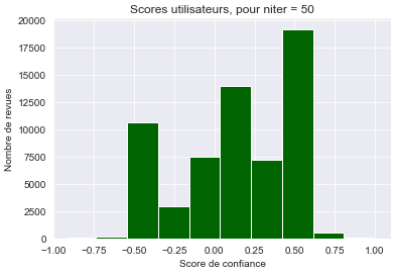
Voici les scores utilisateurs, produits et revues obtenus avec cette initialisation.



(a) $niter = 1$

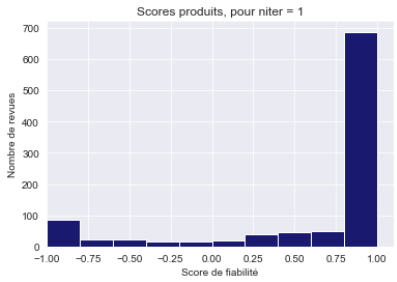


(b) $niter = 10$

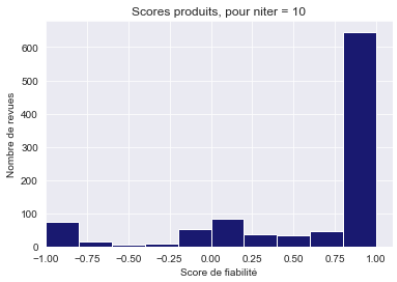


(c) $niter = 50$

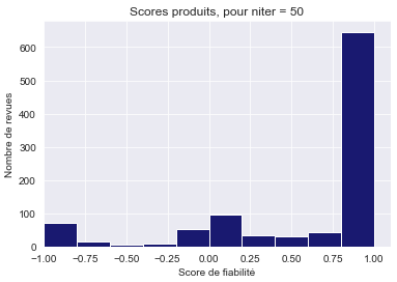
FIGURE 10 – Scores de confiance utilisateur



(a) $niter = 1$

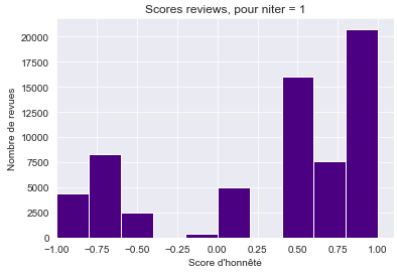


(b) $niter = 10$

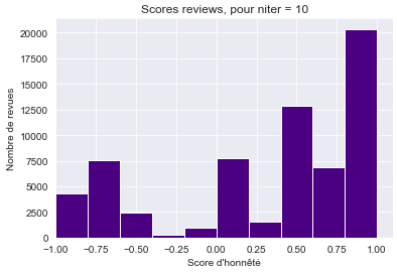


(c) $niter = 50$

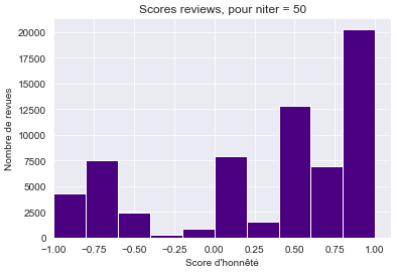
FIGURE 11 – Scores de fiabilité produit



(a) $niter = 1$



(b) $niter = 10$



(c) $niter = 50$

FIGURE 12 – Scores d'honnêteté revue

Nous constatons que l'algorithme de diffusion de la défiance ne nécessite que très peu d'itérations pour converger. Les scores sur les revues n'ont pratiquement pas bougé entre la 1ère et la 50e itération : nous nous rendons compte que la phase d'initialisation est cruciale !

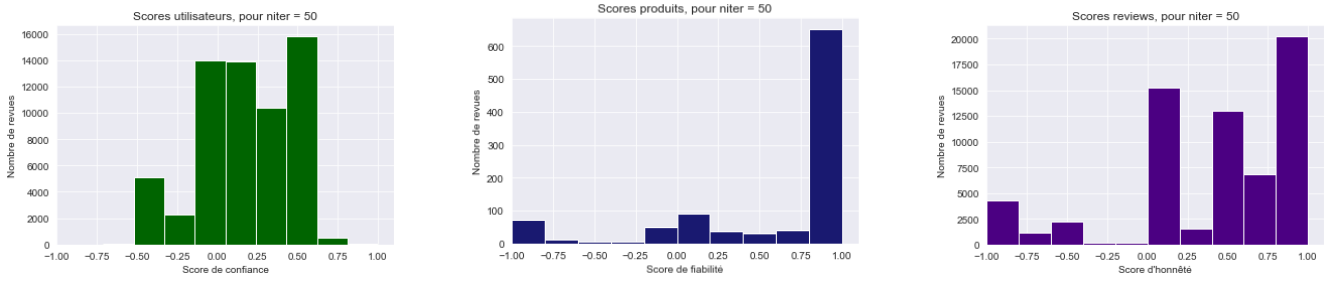
Or, à la 1ère itération, les scores des revues en doublon et des revues suspectes *(revues en période de burst ou déviantes)* sont fixés manuellement. Nous craignons qu'une mauvaise initialisation de l'algorithme impacte beaucoup sur les performances de notre système.

Le problème posé par l'initialisation courante est qu'elle n'exploite jamais les scores d'entente des revues en doublons et des revue suspectes : nous passons à côté d'une information précieuse qu'il nous faudrait utiliser. Elle nous serait en particulier utile dans le cas d'une revue se trouvant en période de burst : l'heuristique de burst étant plutôt incertaine, il vaudrait mieux éviter de compter sur une initialisation purement manuelle qui risque de ne pas beaucoup changer au fil des itérations.

4.4 Amélioration de l'initialisation

Plutôt que de changer l'initialisation des scores de revues, nous nous attachons à changer les scores utilisateurs et produits, ce qui nous garantirait une plus grande flexibilité.

- **Deuxième proposition pour l'initialisation :** comme pour la première initialisation, une revue qui est à la fois en doublon et dans une période de burst verra son score d'honnêteté initialisé à -1. Dans le cas où une revue est soit en doublon, soit suspecte *(revues dans un burst ou revues déviantes)*, nous initialiserons cette fois-ci les scores des utilisateurs et produits associés à 0 plutôt qu'à 1.

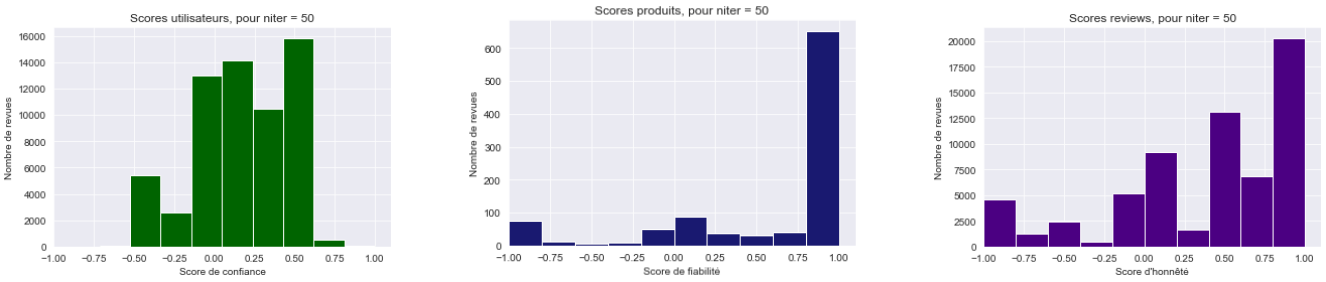


(a) Confiance utilisateur (b) Fiabilité produit (c) Honnêteté revue

FIGURE 13 – Scores de défiance sur les utilisateurs, produits et revues pour $niter = 50$

La propagation ne se passe pas comme nous le souhaiterions : beaucoup de revues voient maintenant leur honnêteté bloquée à 0, ce qui ne nous apporte aucune information sur leur degré de défiance. Ce phénomène résulte directement de l'initialisation des scores de fiabilité des produits à 0 pour les revues suspectes, scores qui interviennent directement dans le calcul des mesures d'honnêteté sur les revues.

- **Troisième proposition pour l'initialisation :** Pour palier à ce problème, nous ajoutons simplement un biais à l'initialisation des scores des utilisateurs et des produits sur les revues suspectes *(en période de burst ou déviantes)*. Ces scores sont ainsi initialisés à -0.02 au lieu de 0, ce qui nous permet d'éviter un blocage de la propagation des scores au niveau revue.



(a) Confiance utilisateur (b) Fiabilité produit (c) Honnêteté revue

FIGURE 14 – Scores de défiance sur les utilisateurs, produits et revues pour $niter = 50$

La propagation de scores n'est désormais plus aussi stagnante qu'auparavant, et les scores obtenus semblent maintenant tout à fait cohérents : sur nos 60 000 revues, près de 20 000 sont complètement honnêtes et moins de 5000 sont très suspectes (honnêteté à -1). Parmi les produits, la très grande majorité est fiable et seuls 80 d'entre eux semblent avoir été attaqués (fiabilité à -1). Enfin, les scores de confiance sur les utilisateurs sont bien plus nuancés : ils tournent entre -0.5 et 0.75.

Une analyse rapide sur les 50 produits les moins fiables (produits attaqués, score à -1) et les 50 produits les plus fiables (score à 1) nous permet d'identifier les phénomènes suivants :

- pour les produits de pires scores, nous remarquons qu'un burst de revue s'accompagne souvent (mais pas toujours) d'une chute ou d'une remontée assez brute de la note moyenne : un tel phénomène confirmerait nos suspicions d'attaque sur le produit. Si le critère n'est pas respecté, nous pouvons émettre l'hypothèse qu'un produit n'est en réalité pas attaqué.

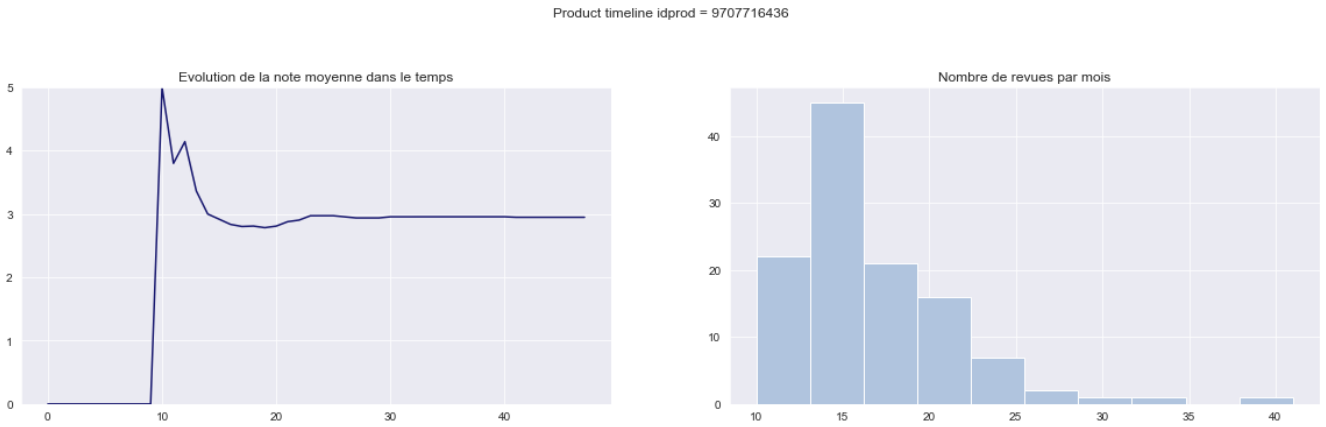


FIGURE 15 – Evolution de la note moyenne et du nombre de revues pour un produit de score -1

- pour les produits de meilleurs scores par contre, un burst de revue (même très fort) ne s'accompagne d'aucun changement particulier dans la note moyenne.

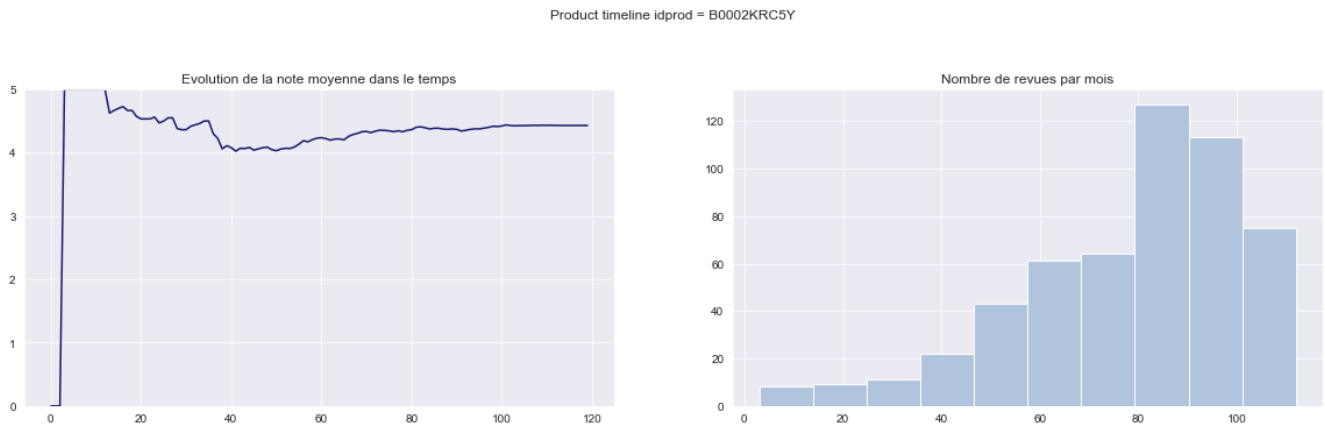


FIGURE 16 – Evolution de la note moyenne et du nombre de revues pour un produit de score 1

Une simple analyse de graphe ne suffit pas à confirmer nos suspicions. Nous nous proposons d'enrichir les résultats apportés par la méthode de graphe avec des techniques de traitement de langage.

5 Traitement Automatique du Langage : SVM pour l'analyse des bursts de revues

5.1 Intuition générale

À ce stade du projet, plusieurs techniques de détection de revues spams ont été proposées et implémentées : nous avons identifié des périodes de burst potentiellement spams grâce par une analyse temporelle de l'arrivée des revues, introduit une mesure de déviation à la moyenne qui nous permet de quantifier les comportements de notation suspects, et enfin attribué un score de défiance à chaque auteur, produit et revue dans une optique de diffusion d'information dans un graphe avec un algorithme PageRank.

Nous souhaitons maintenant vérifier si les soupçons levés par nos différentes métriques sur certaines revues sont fondés. Pour cela, nous nous proposons d'extraire parmi les revues notées par notre algorithme de graphe les 50 produits de pires scores *(score d'honnêteté proche de -1)*, ainsi que les 50 produits de meilleurs scores *(score d'honnêteté proche de 1)*. L'enjeu est maintenant d'analyser la **timeline** de ces produits (i.e. le nombre de revues postées par mois) en comparaison avec l'évolution de la note moyenne du produit dans le temps. Nous souhaiterions ainsi extraire de ces observations des caractéristiques propres aux revues spams, qui nous permettraient de les différencier des revues honnêtes.

Voici la **pipeline** proposée pour mettre en place cette analyse :

- Choisir un produit et analyser sa timeline ainsi que l'évolution de sa note dans le temps. Dans le cas d'un produit de score faible (produit attaqué), nous nous attendrions à ce qu'une telle période s'accompagne d'une remontée ou d'une baisse significative de la note moyenne. Au contraire pour un produit fiable et donc non attaqué, nous nous attendrions à ce que les bursts temporels n'aient que peu à aucun effet sur la note globale.

- Déterminer une fenêtre de temps correspondant à une période de burst. C'est dans cette fenêtre que nous nous attacherons à vérifier la présence de revues spams.

- Représenter les revues de cette période sous forme de vecteurs de bigrammes *(tf-idf)*, puis calculer une matrice de similarité sur chaque paire de revues. Deux revues seront considérées similaires si elles dépassent un certain seuil s que nous déterminons manuellement à l'aide d'un histogramme sur les similarités. Les similarités supérieures à s sont mises à 1, les autres à 0.

- Nous appliquons maintenant un deuxième seuillage sur les sommes des similarités entre une revue et toutes les autres. Si la revue est similaire à plus de t revues, nous considérons qu'elle a été rédigée par un groupe de spammeurs et l'étiquetons à 1 (revue spam), sinon à -1. Nous nous retrouvons donc avec un corpus de revues annotées, où la classe positive correspond à l'ensemble des revues spams sur la période de burst tandis que la classe négative correspond aux revues honnêtes

- Nous souhaitons maintenant extraire de ce corpus l'ensemble des **features discriminants** (bigrammes discriminants) pour chacune des deux classes. Nous espérons ainsi être capables de retrouver des caractéristiques sur les revues spams à partir de cette analyse. Nous nous proposons donc d'entraîner un SVM sur notre base de revues étiquetée, et extrayons de la matrice des poids obtenue les bigrammes les plus représentatifs de chaque classe. Une analyse de ces features sera nécessaire pour déterminer si oui ou non les revues de la classe 1 ont bien l'air d'être des spams.

- Enfin la même analyse doit être faite en période hors burst afin de vérifier les hypothèses émises et de valider nos résultats.

5.2 Exemples illustrés

5.2.1 Analyse sur un produit jugé "attaqué"

Nous nous proposons d'abord de mettre en oeuvre cette analyse sur l'un des produits auxquels l'algorithme PageRank a attribué un score parmi les plus faibles : d'après la technique de graphe, ce produit serait donc attaqué. Nous prenons comme produit à analyser l'amplificateur GSM que nous avions déjà étudié, et qui a un score de fiabilité à -1 *(id produit : B000O8TWE8)*.

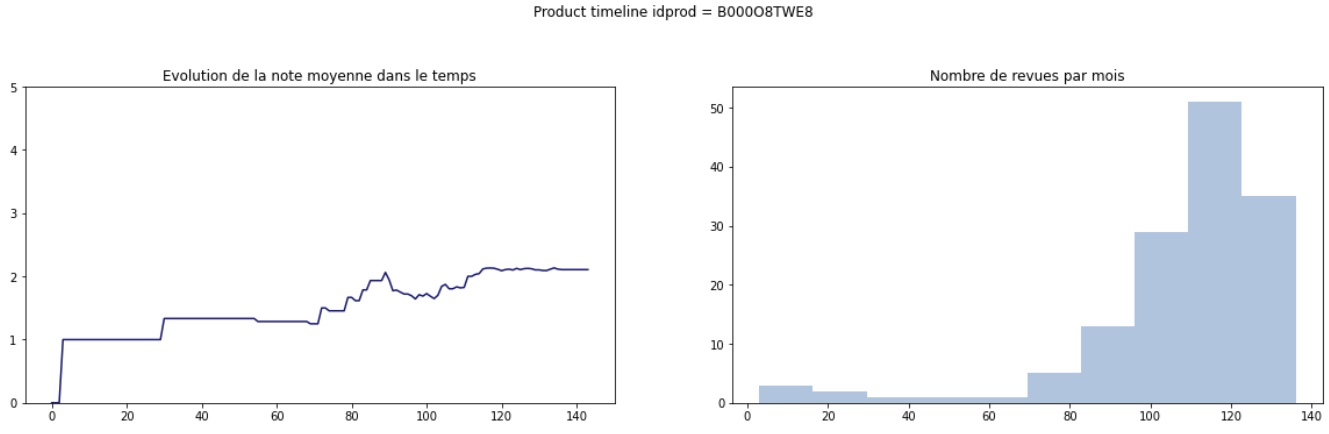


FIGURE 17 – Evolution de la note moyenne et du nombre de revues par mois

La figure 17 nous permet de détecter une période de burst entre les mois 100 et 140. Nous suspectons que cette période de burst est bien une période d'attaque de spammeurs car la moyenne, qui jusqu'au mois 99 tournait autour de 1 étoile, se met tout d'un coup à augmenter. Nous émettons donc l'hypothèse que pendant cette fenêtre de temps, l'entreprise aurait payé des spammeurs afin qu'il postent des avis positifs sur le produit. Nous souhaitons vérifier cette hypothèse par la technique proposée dans cette section.

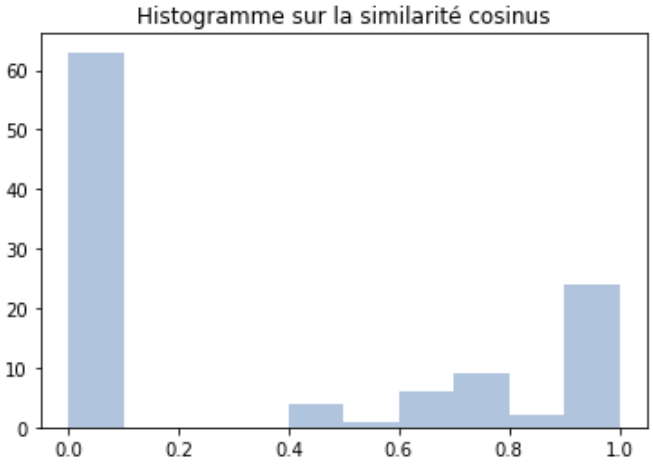


FIGURE 18 – Histogramme des similarités pour chaque paire de revues sur le produit

La figure 18 nous permet de constater qu'il y a un grand nombre de revues similaires pendant cette période de burst. Nous choisissons de considérer un seuil $s = 0.7$ au-delà duquel deux revues sont considérées similaires. Nous mettons ces paires à 1 dans la matrice de similarités et les autres à 0.

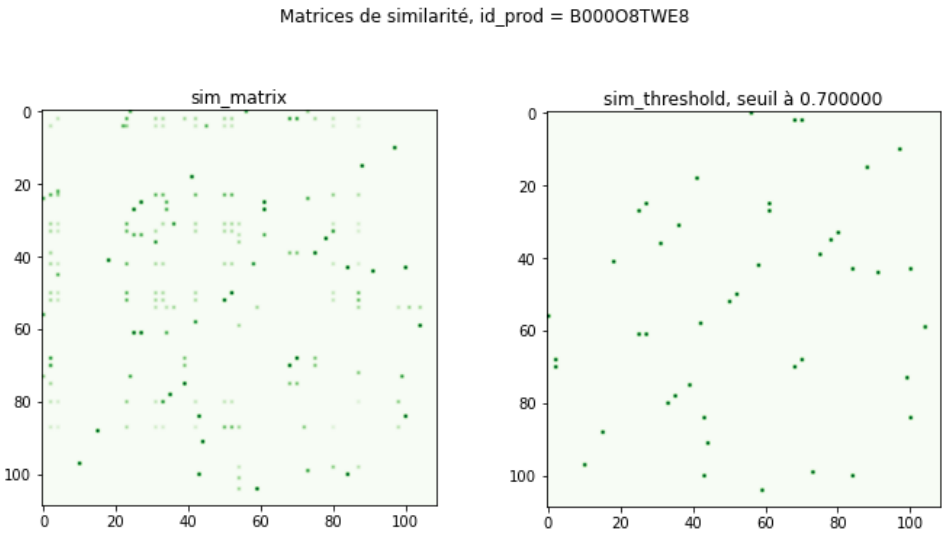


FIGURE 19 – Matrices de similarités initiale et seuillée sur les revues du produit

Nous appliquons maintenant un deuxième seuillage, qui s'effectue cette fois sur chaque paire de revues (i, j) où i est une revue fixée. Nous souhaitons ainsi mettre en évidence les revues qui sont fortement similaires à un grand nombre d'autres revues dans la période de burst, et qui auraient donc possiblement été écrites par un groupe de spammeurs. De telles revues seront considérées dans notre analyse comme potentiellement spams et nous les étiquetons à 1. Le reste des revues du burst sont labellisées à -1.

Nous entraînons maintenant un modèle de SVM pour lequel nous nous intéressons à la matrice de poids. Nous vonlons ainsi retrouver les bi-grammes discriminants pour chacune des deux classes (spam et non-spam), c'est-à-dire les bi-grammes ayant plutôt tendance à apparaître dans les revues d'une classe en particulier.

FIGURE 20 – Bigrammes discriminants pour les classes -1 (gauche) et 1 (droite) en période de burst

Nous voyons aisément dans ce diagramme que parmi les 5 features les plus discriminants sur les revues potentiellement spams, 3 sont particulièrement positives (*"sure good", "better recept", "improv recept"*) tandis que les deux autres sont neutres. Par contre pour la classe -1, nous retrouvons 1 bigramme négatif ("snake oil" qui signifie *arnaque*) et 4 neutres. Il y a donc un désaccord entre les revues similaires (classe 1 spam) et les revues moins similaires dans cette période de burst. Cela semble aller dans le sens de notre hypothèse : les revues spams, qui sont similaires car provenant d'un même groupe de spammeurs, auraient tendance à faire l'éloge du produit tandis que les revues provenant d'utilisateurs différents semblent plus neutres voire négatives. Nous vérifions qu'en période hors-burst, nous observions les mêmes tendances que pour la classe -1 en burst.

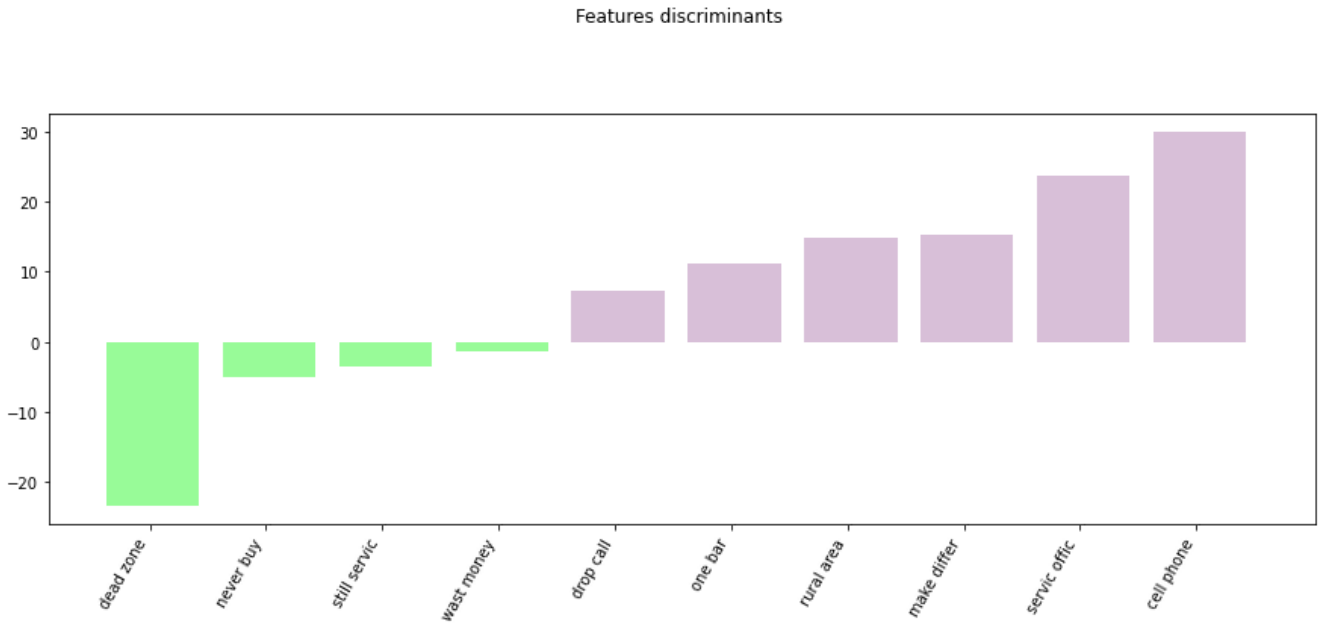


FIGURE 21 – Bigrammes discriminants pour les classes -1 (gauche) et 1 (droite) en période hors-burst

Les résultats confirment bien nos soupçons : en période hors-burst, les revues les plus similaires comme les revues moins similaires ont pour features discriminants des bigrammes négatifs ou neutres. Notre hypothèse d'attaque de spammeurs en période de burst se confirme donc.

5.2.2 Analyse sur un produit jugé "fiable"

Nous essayons cette fois-ci de faire la même étude, mais sur un produit jugé fiable par PageRank *(score à 1)*. L'analyse de la timeline du produit (Figure 22) nous montre que la période de burst entre les mois 120 et 140 n'amène aucun impact sur la note moyenne du produit. Nous émettons donc l'hypothèse que cette arrivée massive de revues n'est pas liée à une attaque de spammeurs mais est probablement le résultat d'une campagne marketing réussie.

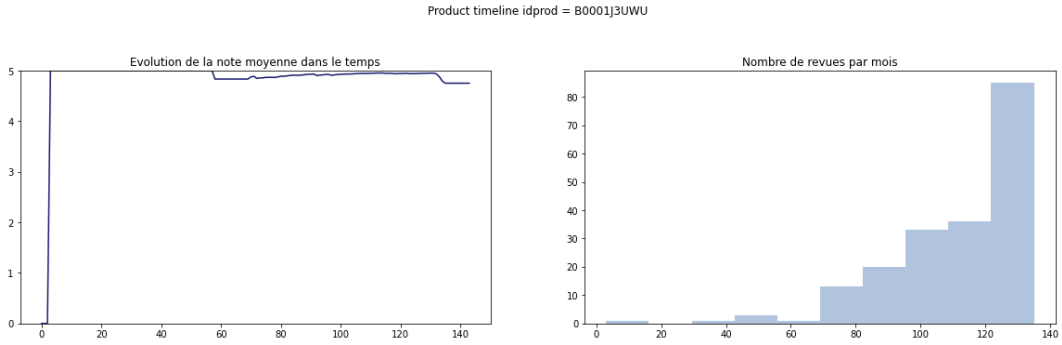


FIGURE 22 – Evolution de la note moyenne et du nombre de revues par mois

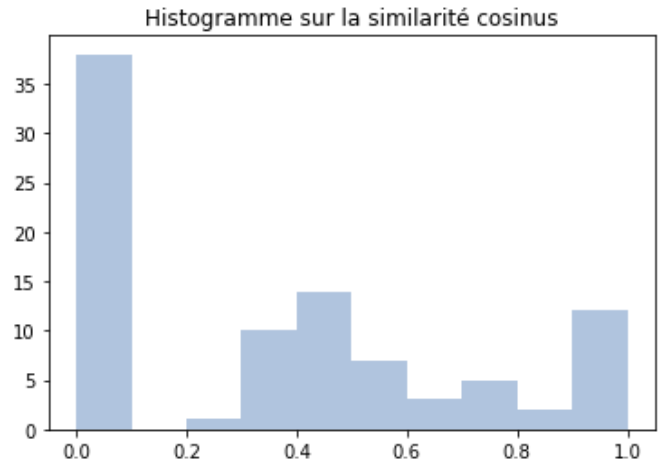


FIGURE 23 – Histogramme des similarités des paires de revues en période burst

Avec cet histogramme, nous pouvons constater qu'il existe un certain nombre de revues similaires dans la fenêtre de burst. Nous pensons cependant que ces revues similaires sont très probablement des revues très courtes telles que *"Good product"* ou *"OK"*.

Comme tout à l'heure, nous seuillons deux fois sur la matrice de similarités afin d'extraire les revues similaires à plusieurs autres *(classe 1, potentiel groupe de revues spams)*. Les autres revues sont classées -1. Etudions maintenant les bigrammes discriminants pour chacune de ces deux classes.

FIGURE 24 – Bigrammes discriminants pour les classes -1 (gauche) et 1 (droite) en période de burst

Cette fois-ci, les bigrammes discriminant les classes 1 et -1 sont tous positifs ou neutres. Il n'y a cette fois-ci aucune divergence d'opinion entre les revues similaires et les revues non-similaires en période de burst ! L'analyse en période hors-burst nous donne les mêmes résultats, comme nous pouvons le voir ci-dessous :

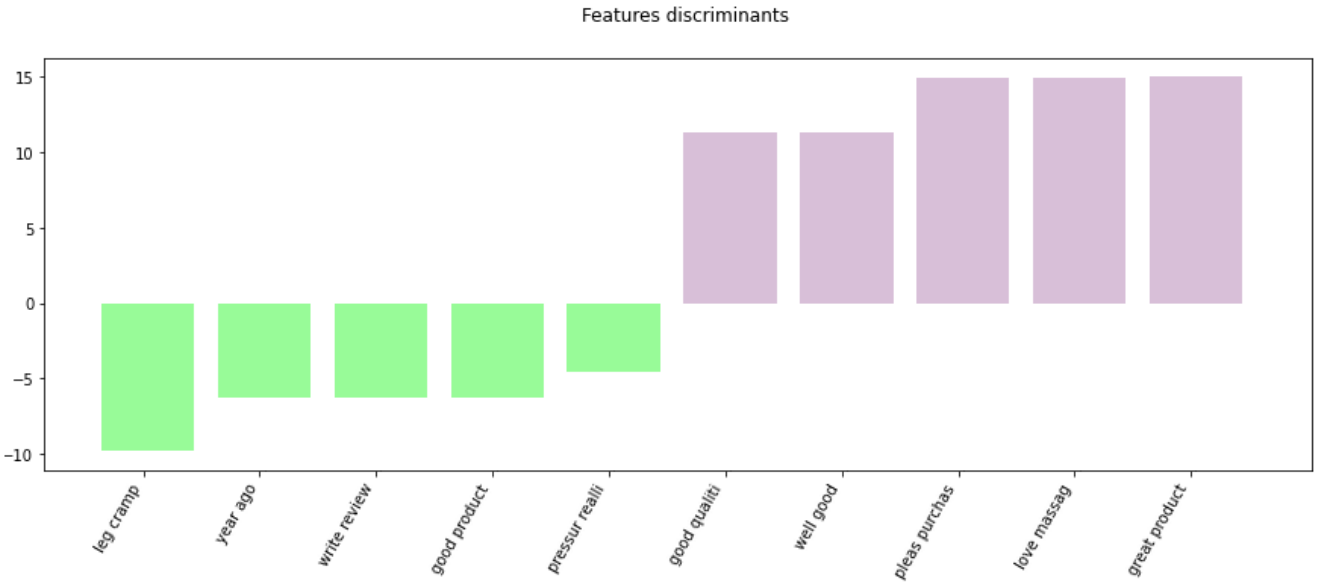


FIGURE 25 – Bigrammes discriminants pour les classes -1 (gauche) et 1 (droite) en période hors-burst

5.3 Recul sur les résultats

Sur ces deux exemples, notre nouvelle technique d'analyse textuelle de revues en période de burst s'est révélée assez utile pour valider les résultats obtenus avec l'analyse par graphe. Nous devons cependant nuancer cette observation, car notre analyse textuelle ne nous a pas permis de tirer de conclusions à partir des bigrammes discriminants sur un certain nombre de produit jugés "attaqués" par PageRank. Cela peut venir d'une attribution des scores qui n'est pas complètement juste par l'algorithme PageRank (un produit jugé non-fiable alors qu'il l'est) ; notre technique d'analyse par traitement du langage s'avère donc un bon moyen de ré-évaluer la méthode précédente.

La principale limite de notre technique de détection par analyse du langage est qu'elle ne prend justement en compte que le contenu des revues, et jamais des aspects liés à la notation ni au temps. Elle ne prend pas non plus en compte les liens existant entre auteurs, produits et revues. Une autre limite toute aussi importante est qu'il est très difficile de l'utiliser indépendemment des autres méthodes de détection : comment détermine-t-on les seuils à fixer ? Comment juger automatiquement et non manuellement que des features discriminants sont en accord ou en conflit ?

Enfin, il existe des cas de figure dans lesquelles cette technique fonctionnera très mal : dans le cas de revues de petite taille, ou dans le cas où un burst de spams positifs arrive pour un produit déjà bien noté (par exemple dans le cas où un vendeur essaie de promouvoir son produit, déjà bien noté mais ayant reçu trop peu d'avis).

6 Conclusion générale

Nous avons implémenté plusieurs techniques afin de distinguer les revues spams des revues honnêtes tout au long de ce projet. La première étape a été de mettre en place des heuristiques de détection par analyse de bursts et par déviation de notation. Ces heuristiques, bien qu'elles présentaient l'avantage de tirer profit d'informations temporelles sur les revues et comportementales sur les utilisateurs, n'utilisaient à aucun moment les relations liant les auteurs, les revues et les produits entre-eux. Elles ont donc été combinées à une technique de diffusion de défiance dans un graphe, nous permettant ainsi d'obtenir pour chaque entité *(auteur, produit, revue)* un score de défiance. L'absence d'analyse même du contenu des revues est comblée par une analyse du langage en période de burst sur les produits jugés comme "attaqués" par notre PageRanK. Au final, nous obtenons un système qui devrait être capable de détecter des avis spams dans une base de revues. Il nous est cependant très difficile de l'évaluer, puisque nous ne disposons pas de labels sur nos données pour vérifier nos prédictions.

Si le temps nous l'avait permis, nous aurions aimé approfondir le sujet en essayant cette fois-ci d'étudier les différents groupes de spammeurs que nous aurions réussi à identifier, et analyser les styles d'écritures et caractéristiques sur le vocabulaire utilisé, sur les tournures de phrases courantes, etc... afin de développer un système capable de prédire si une revue a été écrite par un groupe de spammeurs déjà connu.

Bibliographie

[1] Geli Fei **andothers**. "Exploiting Burstiness in Reviews for Review Spammer Detection". **in**: (2013). (**urlseen** 24/05/2021).

[2] J. McAuley et J. Leskovec. *Données Amazon*. 2013. URL: `http://snap.stanford.edu/data/web-Amazon.html` (**urlseen** 24/05/2021).

[3] Bing Liu **andothers**. "Review Graph based Online Store Review Spammer Detection". **in**: (2011). (**urlseen** 23/05/2021).

[4] Guan Wang **andothers**. "Review Graph based Online Store Review Spammer Detection". **in**: (2011). (**urlseen** 24/05/2021).