Rapport 1 - Projet de traitement de données massives

Jacob Comeau[†], Gabriel Jeanson[‡], Alexandra Prémont[◇]
[†] Département d'informatique et de génie logiciel, Université Laval, Québec, Canada
[‡] Département d'informatique et de génie logiciel, Université Laval, Québec, Canada
[◇] Département d'informatique et de génie logiciel, Université Laval, Québec, Canada

Abstract

Les enjeux des médias traditionnels sont, de plus en plus, grandissants. Avec le récent blocage des nouvelles par Meta, il est plus difficile pour les individus d'interagir et de partager sur les nouvelles. Dans ce rapport, on analyse des données collectées par l'API Facebook et on étudie différents algorithmes utiles pour prédire les commentaires des utilisateurs qui reçoivent des réponses sous les publications du journal Le Soleil. Les données nécessitent un prétraitement comme elles ont, entre autres, des valeurs manquantes, bruitées et aberrantes, un déséquilibre des classes et un nombre massif de dimensions. Certains attributs tels que le nombre de *likes* du commentaire semblent potentiellement pertinents pour la présente tâche de classification. Deux algorithmes de classification seront utilisés et comparés, soit la forêt aléatoire et le classificateur naïf de Bayes.

Mots-clés: Commentaires, Réponse, Facebook, prétraitement, Analyse, Classification

1. Introduction

Depuis le 1^{er} août 2023, Meta bloque l'accès au contenu d'information sur Facebook et Instagram. Cela vient limiter, du moins en ligne, l'échange entre les individus sur les différents sujets d'actualités. Dans ce rapport, on s'intéresse à prédire quels commentaires des utilisateurs reçoivent des réponses sous les publications publiées sur Facebook par le journal Le Soleil. Il faut bâtir des modèles pour réaliser cette tâche de classification binaire. On dispose de données collectées par l'API Graph de Facebook. Il y a, en fait, trois fichiers de données. Le premier fichier est *Posts.csv* et il contient les publications du journal Le Soleil. Le deuxième fichier est *Comments.csv* et contient les commentaires des utilisateurs sous les publications du journal. Le troisième fichier est *Test.csv*. Il contient les commentaires des utilisateurs également, mais les deux attributs qui peuvent être utilisés pour former l'étiquette sont masqués. Ce fichier sert à mesurer la performance du meilleur classifateur développé.

Le projet est divisé en deux parties et, ainsi, deux rapports. Ce premier rapport permet d'explorer et de se familiariser avec les bases des données et de prendre certaines décisions quant aux attributs initiaux, à la méthodologie et aux algorithmes à utiliser. En effet, il comprend l'analyse des données, la sélection des attributs initiaux, le traitement des données et la mise en place d'une procédure de tests. Les différents points discutés dans ce rapport seront importants pour la deuxième partie de ce projet où il faut implémenter les différents algorithmes afin de réaliser la tâche de classification et évaluer les prédictions obtenues.

2. Analyse des données et de leurs propriétés statistiques

2.1. Taille

Le fichier *Posts.csv* compte 37 504 tuples et 9 attributs. Le fichier *Comments.csv* inclut 935 698 tuples et 13 attributs. Le fichier *Tests.csv*, quant à lui, a 107 470 tuples et 13 attributs. Il s'agit de quantités massives de données. Le nombre de dimensions est massif dans les bases de données puisque, malgré que le nombre d'attributs ne soit pas très élevé, il y a un grand nombre de valeurs pour plusieurs des attributs tel qu'il est possible de le constater à l'aide des tables 1(a) et 1(b). Ainsi, certains algorithmes pourraient souffrir du fléau de

[†]jabcob.comeau.1@ulaval.ca [‡]gabriel.jeanson.1@ulaval.ca [◇]alexandra.premont.1@ulaval.ca

dimensionnalité en raison du grand nombre de dimensions des bases de données. En effet, le fléau de dimensionnalité cause une variété de problèmes, dont l'explosion combinatoire des sous-espaces et la perte de sens de la densité et de la distance entre les points.

Attributs	**Nombre de valeurs distinctes**
attachments.data	37 478
created_ time	37 503
id	37 504
mainTopic	2048
message	36 434
permanentlink_ url	37 504
secondTopic	8837
shares	578
title	36

(a) *Posts.csv*

Attributs	**Nombre de valeurs distinctes**
IDENTITY_ ATTACK	2445
INSULT	2557
PROFANITY	2597
SEVERE_ TOXIC	2425
THREAT	2132
TOXICITY	2408
comment_ count	144
created_ time	926 054
id	935 698
like_ count	278
message	906 514
parent	114 788
postID	25 977

(b) *Comments.csv*

Table 1. Nombre de valeurs distinctes par attribut des fichiers (a) *Posts.csv* et (b) *Comments.csv*

2.2. **Informations manquantes**

La table 2 regroupe tous les attributs ayant des valeurs manquantes des fichiers *Posts.csv* et *Comments.csv*. Pour *Posts.csv*, la proportion de valeurs manquantes est relativement faible pour tous les attributs. Il est intéressant de noter que tous les tuples ayant un *mainTopic* manquant ont aussi un *secondTopic* manquant. Tous les tuples ayant un *title* manquant ont aussi un *mainTopic* et un *secondTopic* manquant. De plus, trois titres ont la valeur : « Ce contenu n'est pas disponible actuellement » et leur *mainTopic* et leur *SecondTopic* sont manquants. Pour *Comments.csv*, le seul attribut avec des valeurs manquantes est *parent* et la proportion est élevée. Une valeur manquante est attendue pour cet attribut lorsqu'un commentaire est un commenaire racine, c'est-à-dire qu'il n'est pas en réponse à un autre et n'a donc pas de parent. La table ne comprend pas les valeurs manquantes du fichier *Tests.csv* puisque les seuls attributs avec des valeurs manquantes sont *comment_ count* et *parent*, qui ont été masqués intentionnellement.

Fichier	**Attributs**	**Nombre de valeurs manquantes**	**Proportion de valeurs manquantes (%)**
Posts.csv	*attachments.data*	2	0,01
	mainTopic	297	0,79
	message	460	1,23
	secondTopic	2297	6,12
	title	293	0,78
Comments.csv	*parent*	459 661	49,12

Table 2. Nombre et proportion de valeurs manquantes des attributs des fichiers *Posts.csv* et *Comments.csv*

2.3. **Attribut cible (étiquette)**

Le but du projet est de prédire quels commentaires des utilisateurs recevront des réponses. Aucun attribut des bases de bonnées ne correspond directement à cela. Un nouvel attribut nommé *has_answers* a été créé à partir de l'attribut *comment_count*. Si un commentaire n'a reçu aucune réponse ($comment\_count = 0$), on assigne la valeur 0 à *has_answers*. Si le nombre de réponses à un commentaire est supérieur à 0 ($comment\_count > 0$), on attribue la valeur 1 à *has_answers*. Il s'agit d'un attribut binaire asymétrique puisque l'état 1 est, à la fois, plus important et plus rare. À la figure 1, il y a la distribution de cet attribut. Il est possible d'y constater un déséquilibre entre les deux classes. En effet, il y a 87,60 % des commentaires qui n'ont aucune réponse et seulement 12,40 % des commentaires qui ont au moins une réponse. Le mode est 0 et la moyenne est 0,1240. Il risque d'y avoir un biais contre la classe rare, soit $has\_answers = 1$. Il faudra tenir compte de ce déséquilibre des classes autant lors de la sélection, de l'implémentation et de l'évaluation des algorithmes de classification.

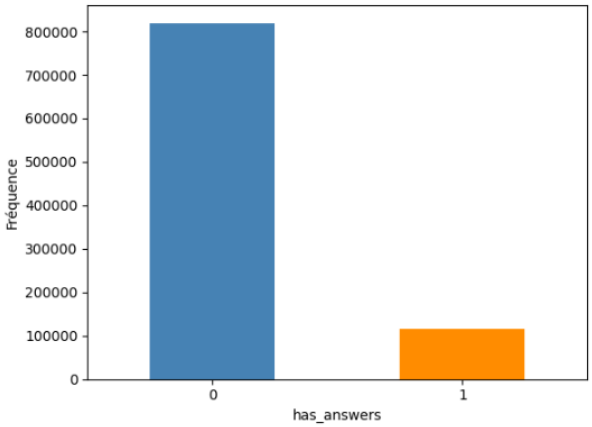


Figure 1. Distribution de *has_answers*

2.4. **Attributs de *Comments.csv***

Analysons, d'abord, les attributs numériques à l'aide de la table 3. Il est possible de remarquer que tous les scores se situent entre 0 et 0,9736, ce qui est plausible. Les catégories de toxicité avec la moyenne la plus élevée sont, en ordre, *INSULT*, *TOXICITY*, *PROFANITY*, *SEVERE_TOXICITY*, *IDENTITY_ATTACK* et *THREAT*. L'écart interquartile (IQR) est plus grand pour *TOXICITY* et d'*INSULT*. Cela signifie que les valeurs pour ses deux catégories varient plus que les autres. Par ailleurs, pour tous les scores de toxicité, il y a une asymétrie positive, c'est-à-dire que la majorité des données ont de faibles valeurs et qu'il y a une queue de valeurs très élevées. Autrement dit, la médiane est plus faible que la moyenne pour chacune des catégories de toxicité. En se basant sur la mesure selon laquelle il y a la présence de valeurs aberrantes au-delà de plus ou moins 1,5 fois l'intervalle interquartile (données aberrantes du diagramme en boîte), il y a des valeurs très distantes des autres et qui peuvent alors être considérées comme des valeurs aberrantes pour chacune des catégories de toxicité. Pour les attributs *comment_count* et *like_count*, il y a également une asymétrie positive et des valeurs qui peuvent être considérées comme aberrantes. Il y a un déséquilibre entre les classes, dû à un grand nombre de zéros pour les deux attributs. Il est intéressant de noter que le nombre moyen de mentions « J'aime » est plus élevé que le nombre moyen de commentaires en réponse à un commentaire.

Attribut	Minimum	1^{er} quartile	Médiane	Moyenne	3^{e} quartile	Maximum
IDENTITY_ATTACK	0,0000	0,0005	0,0016	0,0114	0,0055	0,9491
INSULT	0,0023	0,0095	0,0275	0,1090	0,1361	0,9736
PROFANITY	0,0000	0,0089	0,0113	0,0136	0,0047	0,9502
SEVERE_TOXICITY	0,0000	0,0004	0,0012	0,0136	0,0047	0,9502
THREAT	0,0000	0,0053	0,0055	0,0102	0,0062	0,9680
TOXICITY	0,0000	0,0083	0,0343	0,1025	0,1510	0,9818
comment_count	0,0000	0,0000	0,0000	0,5205	0,0000	252,0000
like_count	0,0000	0,0000	0,0000	1,3216	1,0000	836,0000

Table 3. Distribution des attributs numériques du fichier *Comments.csv*

Passons aux autres attributs. Pour l'attribut *created_time*, l'étendue des valeurs est du 27 février 2020 au 15 juin 2023. La base de données contient trois numéros d'identification, soit *id*, *parent* et *postID*. Il s'agit de numéros uniques pour identifier, respectivement, le commentaire, le commentaire parent et la publication. Les numéros ont deux parties, qui sont séparées par le symbole « _ ». Le deuxième numéro d'une publication est attribué au premier numéro d'un commentaire sous cette publication. Par exemple, si l'identifiant d'une publication est 192978590727638_722477749883613, l'identifiant d'un des commentaires sous cette publication pourrait avoir cette valeur 722477749883613_788168749338745. Ainsi, le numéro 722477749883613 est commun aux deux. L'attribut *message* correspond au contenu des commentaires. Il s'agit d'un attribut en langage naturel, qui peut inclure, à la fois, des caractères, des caractères spéciaux, des chiffres, des signes de ponctuation, des émojis, etc. Une grande majorité des commentaires (96,88 %) sont différents tel qu'il est possible de le constater à la table 1(b). La table 4 rassemble les cinq commentaires les plus fréquents sous les publications du journal Le Soleil. Le mode est « Bravo ». Ce petit échantillon montre que certains commentaires expriment la même idée ou des idées similaires, mais sont exprimés de différentes manières. Par exemple, la présence ou l'absence de ponctuation pour « Bravo » et « Bravo! ». Il y a d'autres types de variations comme l'utilisation de minuscules ou de majuscules, les fautes d'orthographe, les variations liées au genre et au nombre, les différentes conjugaisons des verbes, la présence ou non d'émoji, l'émoji utilisé, etc. Un prétraitement doit être appliqué à cet attribut, notamment une tokenisation, mais il faut bin sélectionner les opérations à effectuer afin de ne pas perdre de l'information.

message	**Fréquence**
Bravo	949
Bravo!	452
Félicitations	398
Non	193
Magnifique	177

Table 4. Tableau de fréquences des cinq valeurs de *message* les plus fréquentes

2.5. **Attributs de *Posts.csv***

La base de données *Posts.csv* ne contient qu'un seul attribut numérique, soit le nombre de partages de la publication (*shares*). Sa distribution est présentée dans la table 5. Les valeurs se situent entre 0 et 25 872. Cette étendue de valeurs est possible en réalité. Or, il y a une seule valeur pour 25 872 partages et la valeur la plus proche de celle-ci est 7 510. Cette valeur a un impact important sur la moyenne et l'écart-type entre autres. Elle se situe à plus de 1,5 fois l'écart interquartile de la moyenne. Elle peut être considérée comme une valeur aberrante. D'autres valeurs peuvent aussi être considérées comme aberrantes. Or, il

faut noter que 27,09 % des données n'ont aucun partage. Cela entraîne un déséquilibre des valeurs et contribue à l'asymétrie positive de la distribution.

Attribut	**Minimum**	**1^{er} quartile**	**Médiane**	**Moyenne**	**3^{e} quartile**	**Maximum**
shares	0,0000	0,0000	2,0000	21,3043	9,0000	25 872,0000

Table 5. Distribution des attributs numériques du fichier *Posts.csv*

À partir de l'attribut *created_ time*, il est possible de constater que le fichier *Posts.csv* comprend les publications sur Facebook du journal Le Soleil du 27 février 2020 au 2 juin 2023. Rappelons que les commentaires collectés se situent entre le 27 février 2020 et le 15 juin 2023. Les dates de début concordent et la date de fin est supérieure pour les commentaires que pour les publications, ce qui permet aux publications plus récentes de récolter des commentaires et que ce soit assez représentatif du nombre qu'ils auront au final. L'attribut *id* est un numéro d'identifiant unique par publication. L'attribut *permalink_ url* est le lien vers la publication et est une valeur unique et propre à chaque publication également. Pour chacun de ses deux attributs, la représentation semble assez uniforme et aucune valeur ne semble diverger des valeurs attendues. Les attributs *title* et *message* sont en langage naturel. Au même titre que l'attribut *message* de *Comments.csv*, ils nécessitent un prétraitement adéquat afin de ne pas perdre de l'information. Pour les attributs *mainTopic* et *secondTopic*, il y a des données manquantes (voir la table 2), bruitées et aberrantes. De plus, la représentation des données est hétérogène. À la table 6(a), il y a les 15 valeurs les plus fréquentes de l'attribut *mainTopic*. On y retrouve autant des années (2021, 2022 et 2023), des catégories d'articles générales (actualités, opinions, arts, affaires, sports et chroniques) que des catégories d'articles qui semblent plus spécifiques au journal Le Soleil (le-mag et ma-region). Il y a aussi une série de chiffres et une chaîne de caractères vide qui correspond à une valeur manquante. De plus, la catégorie actualité s'y retrouve deux fois : une fois au singulier et une fois au pluriel. Bref, il est difficile d'établir les valeurs attendues pour cet attribut, de définir une déviation attendue de cette norme et de distinguer une déviation anormale. Ce dernier constat s'applique aussi à *secondTopic* tel qu'il est possible d'observer dans la table 6(b). Il faut noter que les valeurs présentées aux tables 6(a) et 6(b) sont assez représentatives des valeurs de chacun des attributs au sens où il n'y a pas seulement du bruit pour les valeurs les plus fréquentes.

mainTopic	**Fréquence**
2022	9912
actualite	8898
actualites	3472
2023	2255
2021	2191
opinions	1539
arts	1502
affaires	1333
sports	1208
le-mag	628
553956616735728	515
composer-preview	417
chroniques	335
	297
ma-region	160

(a) *mainTopic*

secondTopic	**Fréquence**
covid-19	3083
	2297
01	1817
12	1813
03	1775
02	1726
11	1664
04	1261
justice-et-faits-divers	1245
la-capitale	894
05	778
06	751
09	726
08	706
10	679

(b) *secondTopic*

Table 6. Tableaux de fréquences des 15 valeurs les plus fréquentes de (a) *mainTopic* et de (b) *secondTopic*

3. Sélection des attributs initiaux

3.1. Corrélation des attributs avec *has_ answers*

Avant de sélectionner les attributs initiaux, on étudie la corrélation entre l'attribut cible *has_ answers* et les autres attributs numériques. La table 7 présente les valeurs des coefficients de Pearson et de Spearman entre l'attribut *has_ answers* et les autres attributs. On observe des coefficients de corrélation proches de 0 pour la majorité des attributs, à l'exception de *like_ count*. Il existe donc une corrélation intéressante entre *like_ count* et *has_ answers*.

Fichier	Attributs	Coefficient de Pearson	Coefficient de Spearman
Comments.csv	*TOXICITY*	-0,0328	-0,0073
	THREAT	0,0058	0,0399
	SEVERE_ TOXICITY	-0,0304	0,0141
	PROFANITY	-0,0403	0,0097
	INSULT	-0,0343	-0,0071
	IDENTITY_ ATTACK	0,0104	0,0328
	like_ count	0,2192	0,2407
Posts.csv	*shares*	-0,0065	0,0109

Table 7. Coefficients de Pearson et de Spearman entre les attributs numériques de *Posts.csv* et de *Comments.csv* et *has_ answers*

3.2. Analyse des attributs d'horodatage

L'analyse faite jusqu'ici ne permet pas d'établir la pertinence ou non des attributs d'horodatage quant à la tâche de classification à réaliser. On cherche alors à construire de nouveaux attributs à partir des attributs d'horodatage et à les étudier. Il est, d'abord, possible d'extraire l'heure de l'attribut *created_ time* du fichier *Comments.csv*. Analysons les corrélations de ce nouvel attribut avec l'attribut cible, et ce, à l'aide le la table 8.

Attribut	Coefficient de Pearson	Coefficient de Spearman
comment_ created_ time	0,0190	0,0152

Table 8. Coefficients de Pearson et de Spearman entre l'heure de création du commentaire et *has_ answers*

À première vue, les résultats ne semblent pas très intéressants comme les coefficients de corrélation sont relativement près de 0. La figure 2 permet d'illustrer la relation entre les deux attributs. Il est possible d'y constater que l'heure a un impact sur le fait d'obtenir une réponse ou non. Les commentaires publiés le matin semblent recevoir moins de réponses en moyenne. On peut conclure que l'attribut est potentiellement utile pour prédire quels commentaires reçoivent des réponses.

Il est, ensuite, possible d'extraire l'heure de création de la publication sous laquelle le commentaire a été publié et de faire une analyse similaire. Il ne semble pas y avoir de relation marquée avec l'attribut *has_ answers*.

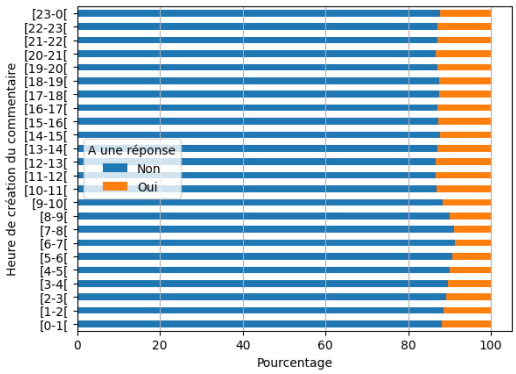


Figure 2. Pourcentage des commentaires ayant reçu ou non au moins une réponse en fonction de l'heure de création du commentaire

Il est, finalement, possible de combiner l'horodatage du commentaire et l'horodatage de la publication. On peut présumer que les commentaires écrits sous des publications qui datent de plus longtemps ont moins de chances d'obtenir une réponse. Cela se base sur le fait que la publication n'est plus d'actualité et se retrouve alors probablement moins dans le fil d'actualité des utilisateurs en général. On crée alors un nouvel attribut nommé *post-comment_ time_ difference*, qui correspond à la différence (en secondes) entre le temps de création du commentaire et le temps de création de la publication associée. On fait l'hypothèse qu'il y a une corrélation négative entre cet attribut et l'attribut cible *has_ answers*. La table 9 regroupe les coefficients de corrélation de Pearson et de Spearman. On observe une corrélation proche de 0 pour le coefficient de Pearson et on peut donc conclure qu'il n'y a pas vraiment de relation linéaire entre les deux attributs. Or, le coefficient de Spearman est de -0,1469. Il s'agit d'une corrélation négative non-négligeable. C'est bien ce qui était attendu, car, logiquement, plus la publication est récente, plus elle est d'actualité et donc plus les commentaires sont vus et ont de chances de recevoir au moins une réponse.

Attribut	**Coefficient de Pearson**	**Coefficient de Spearman**
post-comment_ time_ difference	-0,0173	-0,1469

Table 9. Coefficients de Pearson et de Spearman entre *post-comment_ time_ difference* et *has_ answers*

3.3. **Analyse des attributs linguistiques**

On fait, maintenant, une analyse pour les attributs linguistiques. À partir de l'attribut *message* de *Comments.csv*, on crée de nouveaux attributs : la présence d'au moins un point d'interrogation (*question_ mark_ comment*), d'au moins un point d'exclamation (*exclamation_ mark_ comment*), d'au moins symbole @ (*@_ comment*) et d'au moins un émoji (emoji_comment) dans un commentaire. La table 10 présente la corrélation entre ses nouveaux attributs et l'attribut cible *has_ answers*. Notons que les coefficients de Pearson et de Spearman sont identiques pour ses attributs. Les coefficients dans la table sont proches de zéro. Ainsi, les nouveaux attributs ne semblent pas avoir une forte relation avec l'attribut *has_ answers*. Les corrélations les plus fortes sont avec *emoji_ comment* et avec *question_ mark_ comment*, mais il faut noter qu'elles ne sont pas nécessairement représentatives puisque, par exemple, seulement 12,61 % des commentaires contiennent un point d'interrogation.

Attribut	Coefficient de Pearson	Coefficient de Spearman
question_ mark_ comment	0,0138	0,0138
exclamation_ mark_ comment	0,0043	0,0043
@_ comment	-0,0030	-0,0039
emoji_ comment	0,0188	0,0188

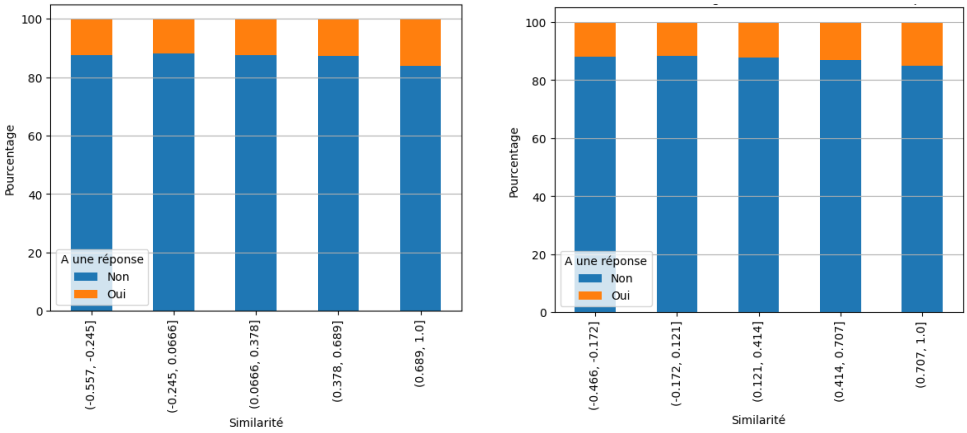
Table 10. Coefficients de Pearson et de Spearman entre *message* de *Comments.csv* et *has_ answers*

Il est possible de faire ressortir la valeur sémantique des attributs linguistiques. Deux nouveaux attributs sont alors construits. Le premier est la similarité entre le contenu d'un commentaire et le titre de la publication associée, nommé *similarity_ message-title*. Le deuxième est la similarité entre le contenu d'un commentaire et le contenu de la publication, nommé *similarity_ message-message*. La similarité a été obtenue en utilisant la librairie *Spacy* et en transformant les mots en plongements de mots (*embeddings*). La table 11 présente la corrélation entre ses nouveaux attributs et l'attribut cible *has_ answers*. Il est possible de remarquer que les coefficients de corrélation sont relativement proches de zéro. La corrélation semble légèrement plus forte pour la similarité entre le contenu d'un commentaire et le contenu de la publication associée (*similarity_ message-message*).

Attribut	Coefficient de Pearson	Coefficient de Spearman
similarity_ message-title	0,0112	0,0107
similarity_ message-message	0,0175	0,0168

Table 11. Coefficients de Pearson et de Spearman entre la similarité des attributs linguistiques et *has_ answers*

Les figures 3(a) et 3(b) permettent de visualiser la relation entre les nouveaux attributs de similarité et l'attribut cible *has_ answers*. On peut observer que plus le contenu du commentaire est semblable au titre de la publication et au contenu de la publication, plus le commentaire a des chances d'avoir au moins une réponse, et ce, malgré le fait que les coefficients de corrélation sont près de 0. En effet, la portion orange des graphiques augmente plus la similarité augmente.



(a) similarité entre le contenu d'un commentaire et le titre de la publication associée

(b) similarité entre le contenu du commentaire et le contenu de la publication associée

Figure 3. Pourcentage des commentaires ayant reçu ou non au moins une réponse en fonction de (a) *similarity_ message-title* (b) *similarity_ message-message*

3.4. **Attributs retenus**

Rappelons que l'attribut *comment_ count* est utilisé pour créer l'attribut cible, soit *has_ answers*. La sélection des attributs initiaux repose sur les éléments suivants : le fléau de dimensionnalité, la quantité de valeurs manquantes, la distribution des attributs, les valeurs aberrantes, la présence de bruit, la corrélation avec l'attribut cible *has_ answers* et le jugement critique et logique. Voici la liste des attributs retenus et les raisons qui motivent leur sélection :

- Contenu des commentaires (*message* de *Comments.csv*) : Il est possible d'extraire différentes informations de cet attribut et de créer de nouveaux attributs, ce qui vient réduire la dimension des données. Les analyses de la section 3.3 montre qu'il y a une certaine relation entre les nouveaux attributs extraits et l'attribut cible.
- Contenu des publications associées aux commentaires (*message* de *Posts.csv*) : Il semble y avoir une relation entre la similarité du contenu d'un commentaire et de la publication associée et l'attribut cible. La corrélation est relativement faible, mais mérite quand même d'être explorer davantage.
- Titre des publications associées aux commentaires (*title* de *Posts.csv*) : Il semble y avoir une relation entre la similarité du contenu d'un commentaire et du titre de la publication associée et l'attribut cible. La corrélation est relativement faible, mais mérite quand même d'être explorer davantage.
- Scores Perspective : Les corrélations observées entre chacun des attributs de score Perspective et l'attribut cible sont relativement proches de 0 (voir la table 7). Cela n'implique pas qu'il n'y a aucune relation. On peut, par exemple, présumer qu'un commentaire qui obtient des scores élevés a une plus forte probabilité d'avoir au moins une réponse. Les attributs reliés aux scores Perspective sont donc conservés à ce point-ci.
- Horodatage du commentaire (*created_ time* de *Comments.csv*) : À la section 3.2, un nouvel attribut est développé qui fait la différence de temps entre la création d'un commentaire et la publication associée. Cet attribut nécessite donc, à la fois, l'horodatage du commentaire et celui de la publication. La corrélation de Spearman est de -0,1469 entre ce nouvel attribut et l'attribut cible. Cette corrélation est notable et on décide alors de conserver l'attribut.
- Horodatage de la publication (*created_ time* de *Posts.csv*) : Cet attribut est retenu pour les mêmes raisons que l'horodatage du commentaire.
- Nombre de mentions « J'aime » (*like_ count* de *Comments.csv*) : La corrélation entre cet attribut et l'attribut cible est supérieur à 0,2. Il existe donc une relation entre les deux attributs. L'attribut like_count fait alors parti des attributs initiaux.
- Nombre de partages des publications (*shares* de *Posts.csv*) : La corrélation entre cet attribut et l'attribut cible est relativement proche de 0. Or, on peut penser que plus une publication a de partages, plus sa « visibilité » augmente et, ainsi, plus les commentaires sont vus et potentiellement répondus. Pour cette raison, on décide de garder l'attribut pour l'instant.

3.5. **Attributs utiles**

Certains attributs des fichiers *Post.csv* et *Comments.csv* ne sont pas directement utiles pour la tâche de classification, mais le sont pour d'autres raisons. L'attribut *id* de *Posts.csv* et l'attribut *postID* de *Comments.csv* permettent de fusionner les deux bases de données. Les attributs *permalink_ url* et *attachments.data* peuvent potentiellement être utilisés pour nettoyer ou enrichir les données.

3.6. **Attributs exclus**

D'autres attributs sont complètement écartés. Voici la liste des attributs exclus et leurs raisons respectives d'exclusion :

- Sujet primaire de la publication (*mainTopic* de *Posts.csv*) : Cet attribut a plusieurs valeurs manquantes. Les données sont bruitées et potentiellement aberrantes tel qu'a révèlé l'analyse de la section 2.5. De plus, le contenu sémantique du titre et du message de la publication permet, en quelque sorte, d'obtenir ce type d'information.
- Sujet secondaire de la publication (*secondTopic* de *Posts.csv*) : L'attribut est exclu, globalement, pour les mêmes raisons que *mainTopic*. En fait, cet attribut a même plus de valeurs manquantes et semble contenir plus de valeurs bruitées que *mainTopic*.
- Numéro d'identification des commentaires (*id* de *Comments.csv*) : Cet attribut est un numéro unique assigné à chaque commentaire et ne semble pas apporter d'informations utiles pour la présente tâche de classification.
- Numéro d'identification du commentaire parent (*parent* de *Comments.csv*) : Il est possible d'utiliser cet attribut pour recréer l'attribut cible *has_answers*. Cet attribut est masqué pour les données de test.

4. **Traitement des données**

Il est nécessaire de faire un prétraitement et un traitement sur les attributs initiaux.

4.1. **Nettoyage des données**

La première étape est d'effectuer un nettoyage des données. Parmi les attributs sélectionnés, il reste *message* et *title* de *Posts.csv* avec des valeurs manquantes. On veut, d'abord, vérifier s'il est possible d'extraire les informations manquantes à l'aide du lien vers la publication qui se retrouve dans l'attribut *permalink_url* ou du lien de l'article (*unshimmed_url*) qui se retrouve dans *attachments.data*. Sinon, on envisage d'ignorer les objets avec des valeurs manquantes. Il faut noter qu'il y a seulement 742 objets qui ont une valeur manquante pour *message* ou *title* ou les deux, ce qui représente 1,98 % des données de *Posts.csv* et 1,91 % des données de *Comments.csv* lorsqu'on y joint les attributs *Posts.csv* à l'aide du numéro d'identification de la publication. Certains attributs contiennent du bruit ou des aberrations. On compte utiliser la méthode du regroupement par classe (*binning*) avec lissage (le type de lissage utilisé va varier en fonction de l'attribut), ce qui va permettre de nettoyer le bruit en éliminant les variations dans chaque classe d'un attribut. Pour les trois variables en langage naturel (*message* de *Comments.csv*, *message* de *Posts.csv* et *title* de *Posts.csv*), il est nécessaire de faire, d'abord, une tokenisation, c'est-à-dire une segmentation des mots. Il est possible d'analyser la présence de ponctuations ou de caractères spéciaux comme vu à la section 3.3. De plus, il est possible d'uniformiser la casse, de corriger les erreurs, de supprimer les mots vides, de lemmatiser, de filtrer les non-mots, etc. Or, toutes ses opérations peuvent faire perdre de l'information. Il faudra alors examiner leur impact respectif.

4.2. **Transformation des données**

La deuxième étape est de transformer les données pour rendre la découverte de connaissances plus efficace et les résultats plus faciles à comprendre. On prévoit faire la construction et l'agrégation de nouveaux attributs. Par exemple, avec les attributs d'horodatage, il est possible de construire un attribut pour l'heure de la création d'un commentaire et un

attribut pour la différence entre le temps de création d'un commentaire et de la publication associée (voir section 3.2). En remplaçant des attributs par des attributs avec moins de valeurs possibles ou en combinant des attributs, cela permet, à la fois, de réduire les données et de rendre certaines relations plus évidentes. Pour les attributs numériques, on compte effectuer une normalisation centrer-réduire, ce qui va permettre d'avoir des échelles de valeurs plus comparables entre les attributs et de réduire l'impact des valeurs aberrantes (s'il en reste à la suite du regroupement par classe). Pour ce qui est des attributs linguistiques, on compte transformer les mots en plongements de mots afin de conserver seulement l'information sémantique. Il est plus simple pour un ordinateur de comparer des vecteurs de chiffres que des chaînes de caractères surtout si le contenu sémantique est important. Cependant, il faut prendre en compte, ici, la quantité massive de données. Le temps de calcul pour obtenir la similarité entre les commentaires et les publications de la section 3.3 a été non-négligeable. Les informations qui se retrouvent dans les attributs linguistiques sont intéressantes et potentiellement utiles pour la suite du projet. Une fois la tokenisation faite et le contenu sémantique extrait, l'utilisation des attributs linguistiques devient plus simple et exige moins en termes de temps de calcul en général (tout dépendant de l'opération effectuée).

4.3. **Réduction des données**

Rappelons qu'il y a, à la base, une quantité massive de données et un nombre massif de dimensions, ce qui peut potentiellement entraîner un fléau de dimensionnalité lors du prétraitement ou du traitement des données. Conjointement au nettoyage et à la transformation des données, il est important de considérer la réduction des données. Certaines opérations de prétraitement aident à réduire la dimensionnalité. La sélection préalable d'attributs initiaux vient exclure certains attributs. La construction de nouveaux attributs permet souvent de remplacer un attribut avec un grand nombre de valeurs possibles par un ou plusieurs attributs qui ont moins de valeurs possibles. L'agrégation d'attributs consiste à combiner plusieurs attributs en un, ce qui vient réduire le nombre d'attributs. Les opérations de traitement du langage naturel peuvent aussi permettre de réduire la dimensionnalité.

4.4. **Algorithmes de classification**

Deux algorithmes seront testés et comparés pour prédire quels commentaires des utilisateurs recevront des réponses.

Le premier algorithme est la forêt aléatoire. Un des principaux avantages de cet algorithme est qu'il permet de faire l'apprentissage d'arbres à partir de données massives, et ce, sans fléau de dimensionnalité. En effet, une forêt aléatoire fait autant un échantillonnage des dimensions et un échantillonnage des données. Un autre avantage important de cet algorithme est que les arbres de décision sont dé-corrélés, contrairement à l'approche des arbres de décision par ensemble. De plus, le fait de faire un échantillonnage sur les données permet de limiter le surapprentissage, contrairement aux arbres de décisions classiques.

Le deuxième algorithme est le classificateur naïf de Bayes. Cet algorithme est basé sur la $P(A_i|C_j)$ où A_i correspond à l'attribut i et C_j correspond à la classe j. Cette probabilité fonctionne bien pour les attributs ayant peu de valeurs différentes et qui sont nominaux, booléens et ordinaux. Ce n'est pas le cas de tous les attributs initiaux retenus. Afin de corriger ses deux limites, il faut réaliser un lissage des données, c'est-à-dire prévoir une faible probabilité pour les valeurs de l'attribut qui ne sont pas observées dans la classe et faire la discrétisation des attributs numériques. Cet algorithme permet également d'éviter le fléau de dimensionnalité, car il crée une lecture linéaire des données.

5. Description de la procédure de tests

Il n'est pas possible d'utiliser les données du fichier *Test.csv* pour évaluer la performance des classificateurs développés comme les attributs *comment_ count* et *parent* sont masqués. Il faut donc développer une procédure de tests.

Les données de *Comments.csv* seront divisés en deux : données d'entraînement (80 %) et données de test (20 %). Les données de test sont mises de côté (*holdout method*). Pour la forêt aléatoire et avec les données d'entraînement, on va effectuer une validation par erreur hors-agrégat. Cette technique consiste à ce que chaque classificateur soit testé en mesurant sa performance sur les échantillons qui ne sont pas utilisés pour l'entraînement. Pour le classificateur naïf de Bayes et avec les données d'entraînement, on va opter pour une validation croisée k-échantillons. Les données d'entraînement seront divisées en 10 échantillons égaux, l'entraînement se fera avec 9 échantillons et sera testé avec un échantillon et cela sera répété 10 fois. La moyenne des 10 tests sera faite. L'apprentissage sera refait sur le meilleur système.

La forêt aléatoire et le classificateur naïf de Bayes retenus seront appliqués sur les données de test. On aura donc la performance réelle des deux classificateurs. Le classificateur avec les mesures de performance les plus élevées sera utilisé pour faire les prédictions pour le fichier *Test.csv*.

En raison du déséquilibre des classes de l'attribut cible *has_ answers*, il faut être prudent avec les valeurs des métriques d'évaluation et leur interprétation. Par exemple, si un classificateur prédit toujours que $has\_answers = 0$, il aurait une exactitude (*accuracy*) élevée, car le nombre de vrais négatifs serait élevé. Il faut corriger le plus possible le déséquilibre des classes ou être particulièrement vigilant avec les métriques d'évaluation.

6. Conclusion

Ce rapport fait une exploration approfondie des données et des attributs qui se retrouvent dans les fichiers *Posts.csv* et *Comments.csv*. Quelques cas problématiques sont identifiés, notamment en ce qui a trait à la présence de bruit, au fléau de dimensionnalité, aux valeurs manquantes et aberrantes et au déséquilibre des classes. Cette exploration et quelques analyses supplémentaires ont été utilisées pour définir les attributs initiaux utilisés pour prédire quels commentaires recevront au moins une réponse. Certains attributs comme *like_ count* et *post-comment_ time_ difference* semblent avoir une relation notable avec l'attribut cible *has_ answers*. À partir de cela, il a été possible d'établir les algorithmes à implémenter autant pour le prétraitement que le traitement des données et la procédure de tests. Cela comprend, entre autres, la regroupement par classe, la similarité entre les attributs linguistiques, la création et l'agrégation d'attributs à partir des attributs d'horodatage, la forêt aléatoire, le classificateur naïf de Bayes et la validation croisée (hors-agrégat ou k-échantillons).

Ce premier rapport servira de base pour la suite du projet et facilitera le traitement des données. Or, les attributs et les algorithmes sélectionnés risquent d'évoluer. Dans le deuxième rapport, le processus itératif d'implémentation des algorithmes sera présenté et les résultats obtenus par les différents classificateurs seront étudiés et discutés.

Remerciements

Ce rapport reprend plusieurs notions et passages du cours « Analyse et traitement de données massives » (GLO-7027). Ils ont également guidé plusieurs analyses et choix.