Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Détection automatique des signaux positifs de l'implication durable dans les conversations de consommateurs en parfumerie

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Yizhe WANG

Directeur de mémoire :

Damien Nouvel

Encadrant :

Marguerite Leenhardt

Année universitaire 2016/2017

TABLE DES MATIÈRES

Liste des figures **5**

Liste des tableaux **5**

Remerciement **7**

Résumé **9**

Introduction **11**

I Contexte général **13**

1 État de l'art **15**
1.1 Notion de l'implication durable 15
1.2 Travaux précédents 16

2 Méthodes **19**
2.1 Approches au niveau des données 20
2.2 Approches algorithmiques 24

II Expérimentations **27**

3 Corpus **29**
3.1 Pourquoi les avis des consommateurs ? 29
3.2 Introduction de la parfumerie en France 29
3.3 Collecte des données et prétraitements 30
3.4 Règles d'annotation 31

4 Classifieur Standard Appliqué **33**
4.1 Machine à vecteurs de support (SVM) 33
4.2 Méthodes d'extraction des caractéristiques implémentées 35

5 Expériences et Résultats **37**
5.1 Méthode d'évaluation 37
5.2 Sélection du noyau et des hyper-paramètres 38
5.3 Baseline sans adaptation liée à la classification asymétrique 41
5.4 Ajuster le poids des classes 43
5.5 Comparaison des méthodes de ré-échantillonnage 44
5.6 Comparaison entre les méthodes par ré-échantillonnage et les méthodes algorithmiques 46
5.7 Affinage du modèle 47

5.8 Expériences avec LSTM 49

6 Discussion **51**

Conclusion générale **55**

Bibliographie **57**

A extraits de corpus **63**

B extraits de codes **65**
B.1 SVM 65
B.2 LSTM 66

Liste des figures

2.1 Méthodes de ré-échantillonnage 20
2.2 Visualization de l'algorithme Smote 21
2.3 Visualization de l'algorithme Tomek Links 22
2.4 Visualization de l'algorithme Smote-TL 23
2.5 Visualization de la fonction du paramètre «classe weight» 25
3.1 Exemples d'expressions contenant les signaux demandés 31
4.1 Hyperplans de SVM 33
4.2 Classification Non-lineaire 34
5.1 Paramètres Candidats 39
5.2 Rapport de la classification en utilisant les différents hyper-paramètres . . 41
5.3 Schéma de Validation Croisée 42
5.4 Performance selon le poids de classe minoritaire 44
5.5 Changement du nombre d'échantillons 46
5.6 Algorithme LSTM 49

Liste des tableaux

2.1 Matrice de coûts 24
3.1 Statistiques sur le corpus 30
5.1 Matrice de Confusion 37
5.2 Résultats en utilisant uniquement SVM 42
5.3 Résultats en paramétrant le poid de la classe minoritaire 43
5.4 Comparaison entre les modèles SVM avec ou sans ajustement des coûts . . 44
5.5 Changement du nombre d'échantillons 45
5.6 Résultats en utilisant des méthodes de ré-échantillonnage 46
5.7 Comparaison de la performance de modèles SVM+Smote et SVM+Poids sur le coût 47
5.8 Paramètres à optimiser 48
5.9 Résultats avant et après l'optimisation des paramètres 48
5.10 Résultats en utilisant LSTM comme classifieur 50
6.1 Tableau de résultats 51
6.2 Analyse de l'erreur de la classification 52

Remerciement

Je tiens à remercier sincèrement en premier lieu toute équipe pédagogique de PluriTAL qui nous a offert une formation de haute qualité et nous a guidé et aidé au cours de nos études pendant deux ans.

Je remercie grandement le Professeur Damien Nouvel qui m'a fait l'honneur d'être mon directeur de mémoire pour ses conseils précieux, sa disponibilité et ses encouragements tout au long de la rédaction de ce mémoire.

Mes remerciements s'adressent également à ma tutrice de stage Marguerite Leenhardt qui m'a patiemment encadrée sur mes travaux réalisés pendant les 5 mois de stage. Ses connaissances riches dans le domaine et sa rigueur m'ont beaucoup aidé à la rédaction du mémoire.

Je dois aussi remercier tous mes collègues de travail et mes camarades, pour leur aide, la qualité de leur travail ainsi que leur bonne humeur à toute épreuve.

Résumé

La fouille d'opinions est un sujet exploité en TAL depuis longtemps. Néanmoins, au cours des dernières années, une simple détection d'opinion positive ou négative ne satisfait plus les chercheurs et les entreprises. Le monde des affaires est à la recherche d'un «aperçu des affaires». Beaucoup d'algorithmes d'apprentissage peuvent être utilisés pour traiter le problème. Cependant, leur performance en face de données déséquilibrées, souvent rencontrées dans l'industrie, est dégradée en raison des caractéristiques complexes inhérentes de ce type de corpus. Notre travail se concentre sur l'étude des techniques visant à traiter la classification asymétrique afin de réaliser notre projet en fouille d'opinions. Cinq méthodes ont été comparées : Smote, Adasyn, Tomek links, Smote-TL et modification du poids de la classe. Notre algorithme conventionnel choisi est SVM et l'évaluation est réalisée par le calcul des scores de précision, de rappel, de f-mesure et du temps d'exécution. Le classifieur LSTM a aussi été testé comme expérience complémentaire. Selon les résultats expérimentaux, la méthode en ajustant le poids sur le coût, qui nous permet d'obtenir notre meilleur F-mesure 67.82% avec le moins de temps d'exécution, obtient la meilleure performance.

Mots-clés : fouille d'opinions, classification asymétrique, SVM, ré-échantillonnage, apprentissage sensible aux coûts

INTRODUCTION

Ce travail a été réalisé au cours du stage chez XiKO, une entreprise spécialisée en TAL qui propose des solutions sémantiques pour le marketing et la publicité. Mining4Meaning™ et Koveri™ sont deux produits principaux créés par l'entreprise. Le premier vise à exploiter les données sur le marketing et les consommateurs afin de permettre au client de mieux connaître les tendances précoces du marché et de mieux comprendre et prévoir le comportement des consommateurs. Le second nous offre des services sur la classification fine du contenu de sites web et l'identification des intérêts et des intentions des visiteurs des sites.

Dans ce contexte, l'objectif de ce travail qui fait appel à la fouille d'opinions est de trouver une méthode pertinente à base d'apprentissage automatique, afin de détecter les signaux faiblement présentés dans les avis des acheteurs.

La fouille d'opinions et l'analyse du sentiment, font l'objet de recherches depuis longtemps. Néanmoins, avec le développement de la technologie et la conjoncture actuelle du marketing, les entreprises cherchent à découvrir plus de valeur cachée dans les textes afin de mieux comprendre les opinions et les besoins de leurs clients et de prendre les meilleures décisions d'affaires. Elles s'attendent plus qu'une simple détection d'opinion positive ou négative mais des appréciations plus fines, comme l'intention d'achat, la préférence pour les produits, la fidélité à la marque, y compris l'implication durable qui aide les experts en marketing à trouver l'explication du comportement de rachat au niveau individuel, et les aide ainsi à cibler les efforts de marketing et affiner les campagnes de marketing.

Bien que la détection des signaux en marketing ne soit pas beaucoup étudiée dans la fouille d'opinions, de nombreux algorithmes d'apprentissage automatique peuvent être utilisés pour faire l'analyse des textes : le réseau de neurones [Wiener et al., 1995] largement développé et utilisé en TAL ces derniers années ; l'arbre de décision [Lewis and Ringuette, 1994] ; les machines à vecteurs de support [Joachims, 1998]...

En effet, notre travail est une classification binaire. La difficulté majeure se concentre sur la faiblesse des signaux à détecter. C'est-à-dire qu'il existe deux classes très déséquilibrées dans notre corpus. Dans ce cas, les algorithmes de classification standards donnent des résultats peu satisfaisants. Le problème se produit et entrave la classification dans des applications. Par exemple, le diagnostic de conditions médicales rares [Murphy and Aha, 1992] ; la recherche et le filtrage d'informations [Lewis and Catlett, 1994] ; la détection de déversement de pétrole dans les images renvoyées par le satellite [Kubat et al., 1998]...

A travers ce travail, nous cherchons à résoudre le problème sur la classification asymétrique, rencontré au cours de notre projet de détection des signaux positifs de l'implication durable, faiblement présentés dans notre corpus dédié à la parfumerie, en comparant la performance des différentes techniques standard. Les machines à vecteur de support sont utilisées comme l'algorithme standard. Notre hypothèse est qu'un algorithme d'apprentissage sensible aux coûts devrait surpasser les méthodes

de ré-échantillonnage. Notez que notre évaluation ne se base pas que sur l'efficience, mais aussi sur l'efficacité.

Nous présenterons d'abord les travaux précédents et notre positionnement. Le deuxième chapitre est dédié à l'explication de la notion et l'introduction. Ensuite, nous étudierons les techniques et le classifieur standard que nous avons choisi pour résoudre nos problèmes sur la classification asymétrique. Le chapitre suivant est consacré à la présentation de notre corpus et nos règles d'annotation. Les expériences et les résultats seront montrés dans la partie suivante et nous terminerons par une discussion suivie par la conclusion.

Première partie

Contexte général

CHAPITRE 1

ÉTAT DE L'ART

Sommaire

1.1 Notion de l'implication durable **15**
1.2 Travaux précédents **16**

1.1 Notion de l'implication durable

La détermination de la polarité contextuelle globale dans les conversations et les commentaires, longs ou courts, a été depuis longtemps un des objets principaux dans le domaine de recherche en fouille d'opinions. Néanmoins, pour les entreprises, savoir comment leur image est aperçue par les consommateurs et si leurs produits plaisent ou pas n'est plus suffisant afin de prévoir les prochaines étapes de leurs développements. Elles ont besoin d'une analyse plus abstraite, comme la détection des signaux positifs de l'implication durable.

L'implication, est un des concepts clés dans les études en psychologie et en comportement du consommateur, qui est devenu central dans la recherche en marketing [Kim, 2004]. C'est une notion issue de la psychologie qui interprète l'implication d'un individu en étudiant la relation avec une autre personne, une cible ou un sujet [Michaelidou and Dibb, 2006]. Bien qu'il soit étudié depuis plus de 30 ans en marketing [Demangeot and Broderick, 2007], ce concept reste quand même vague en raison de son interdépendance avec des significations variées dans de nombreuses disciplines. Par conséquent, il a été utilisé comme un terme générique avec de nombreux vocabulaires semblables dans les domaines différents [Choubtarash et al., 2013]. Néanmoins, en marketing, il fait consensus que cette implication est une variable intrinsèque au niveau individuel qui est assimilée à l'attachement personnel aux objectifs ou aux événements [ABDOLVAND and Nikfar, 2012]. Il y a trois types de classification de l'implication en marketing et parmi lesquelles, le classement par nature qui a été proposée par Rothschild en 1975 est acceptée assez largement : l'implication durable(EI) et l'implication situationnelle(SI).

Contrairement à la SI qui est liée à la situation temporaire du consommateur à l'égard du produit, l'EI est considérée comme un état stable du consommateur auprès d'un produit [Houston, 1978]. Elle représente un niveau d'intérêt ou d'attachement d'un individu envers un produit à long terme [Richins and Bloch, 1986]. D'après Valette-Florence [Valette-Florence, 1989], l'EI se réfère à la fois à l'expérience ou la connaissance antérieure du produit et aux valeurs intérieures des individus. C'est-à-dire que l'EI sera positive pour une personne qui a testé ou utilisé un produit et qui a envie de continuer à l'utiliser pour longtemps ou lui porte une admiration intense.

1.2 Travaux précédents

Il y a deux types de méthodes souvent utilisées en fouille d'opinions : l'approche symbolique basée sur le lexique et l'approche statistique en utilisant l'apprentissage automatique.

L'approche basée sur le lexique utilise généralement une liste de mots ou d'expressions qui portent sur les opinions ou les sentiments des humains [Liu, 2012]. Pour décider de la polarité d'un verbatim, le calcul du nombre de ces termes est souvent réalisé [Ding et al., 2008]. Le sac de mots demandé pour cette méthode peut être créé manuellement [Tong, 2001] ou être généré semi-automatiquement en utilisant une petite liste de mots concernés afin d'obtenir une nouvelle liste avec un plus grand volume [Hu and Liu, 2004]. Et on peut aussi utiliser les listes des termes contenant des sentiments qui existent déjà, comme SentiWordNet [Baccianella et al., 2010], SenticNet [Cambria et al., 2016] et HowNet [Dong and Dong, 2006]. Ces dernières années, cette méthode a été beaucoup développé pour une analyse du sentiment au niveau des aspects. Jo et Oh [Jo and Oh, 2011] utilisent cette méthode afin de trouver les aspects évalués par les consommateurs dans les avis de produits et leur appréciation envers chaque aspect. Son système qui ne demande pas un travail d'annotation très coûteux a obtenu une meilleure performance par rapport aux modèles génératifs.

L'analyse du sentiment en utilisant la méthode statistique est en effet une tâche de classification dont l'étape essentielle est de choisir les caractéristiques, lexicales, syntaxique ou sémantiques, afin de représenter les termes et de choisir les algorithmes de la classification.

Parmi les méthodes possibles pour sélectionner les caractéristiques, TF-IDF [Salton and Buckley, 1988], simple à calculer et obtenant de bonnes performances, est largement utilisé. Plus récemment, la création de Word2Vec par Mikolov [Mikolov et al., 2013] est un évènement important pour les chercheurs en fouille de textes. Selon Mikolov, cette représentation vectorielle permet de rapprocher des mots ayant des sens similaires au sein d'un espace vectoriel. Su et al. [Su et al., 2014] a essayé d'implémenter Word2Vec en utilisant des modèles de réseaux de neurones pour regrouper les mots sémantiquement similaires et apprendre des représentations vectorielles des mots. Le package SVM est adopté ensuite pour classifier les commentaires et ils obtiennent une exactitude de plus de 90%. P Le et W Zuidema [Le and Zuidema, 2015] nous présente une méthode d'analyse des sentiments en utilisant le réseau neural récurrent avec une couche extensive LSTM (Long and Short Term Memory). Cette extension permet de stocker les informations contextuelle afin de réaliser des inférences sur des séquences. A Hassan [Hassan, 2017] a pris presque la même méthode que P Le et W Zuidema et montre que l'utilisation de vecteurs de mots obtenus à partir d'un modèle de réseaux de neurones non supervisé comme caractéristique avec le système RNN-LSTM peut augmenter la performance du système de la fouille d'opinions.

Dans la vie réelle, la fouille d'opinions ne concentre pas que dans la détection de la polarité avec l'intensité [Saleiro et al., 2017] et le nombre d'instances dans les classes est toujours déséquilibré. Comme le travail de M. Leenhardt et G. Patin [Leenhardt and Patin,] dans lequel ils ont proposé une méthode hybride en combinant la méthode textométrique et l'apprentissage automatique pour détecter les intentions d'achat en exploitant les verbatims dans des forums comme corpus. Y. Tang [Tang et al., 2009] a utilisé un SVM comme modèle de base pour traiter la classification largement déséquilibrée. Parmi les quatre variantes qu'il a modifié sur SVM, le

nouvel algorithme de SVM sous-échantillonnage répétitif (GSVM-RU) est le meilleur en termes d'efficacité. Parce qu'il permet de minimiser l'effet négatif de la perte d'information tout en maximisant l'effet positif du nettoyage des données dans le processus de sous-échantillonnage. De plus en extrayant beaucoup moins de vecteurs de support et, le temps demandé pour faire la prédiction SVM est considérablement accélérée. S. Li et Z. Wang [Li et al., 2011b] ont utilisé une méthode basée sur l'apprentissage semi-supervisé en modifiant la technique sous-échantillonnage pour la classification asymétrique du sentiment. Au lieu de faire le ré-échantillonnage, B. Krawczyk et M. Wozniak [Krawczyk et al., 2014] ont créé un ensemble efficace d'arbres de décision sensible aux coûts pour la classification asymétrique. Ils ont développé un algorithme évolutif pour la sélection simultanée des classifieurs et l'attribut des poids pour le processus. Ils ont trouvé que pour leur méthode, les résultats optimaux sont obtenus lorsque le coût est corrélé au ratio de déséquilibre en ajustant ce coût au double du ratio. Une nouvelle approche pour la classification des données déséquilibrée a été proposée par Z. Chunkai et W. Guoquan [Zhang et al., 2017] Cette approche a largement amélioré le résultat à travers la minimisation du coût.

Nos études se concentrent sur la sélection entre quatre méthodes ré-échantionnage et une technique au niveau d'algorithme afin de résoudre notre problème sur le déséquilibre. Diverses expériences ont été menées pour les comparer aux différents aspects. On a aussi testé l'adaptation de l'algorithme présenté par A Hassan, car il est présenté comme une méthode qui peut obtenir un meilleur résultat de classification en utilisant un nombre de données limité, ce qui semble pouvoir répondre à notre besoin.

CHAPITRE 2

MÉTHODES

Sommaire

2.1 Approches au niveau des données **20**
2.1.1 Sur-échantillonnage : Smote 20
2.1.2 Sur-échantillonnage : Adasyn 21
2.1.3 Sous-échantillonnage : Extraction of majority-minority Tomek links 22
2.1.4 Hybride : Smote-TL 23
2.2 Approches algorithmiques **24**

Il n'est pas rare, lorsque l'on travaille avec des données, de se trouver confronté au problème des classes déséquilibrées. C'est un scénario où le nombre d'observations appartenant à une classe est significativement inférieur à celui des autres classes. Ce problème prédomine dans les cas de détection de signaux faibles comme le vol d'électricité, les transactions frauduleuses dans les banques, l'identification de maladies rares, etc. Dans ce cas, le modèle prédictif développé avec des algorithmes conventionnels pourrait être biaisé et inexact et nous avons besoin des techniques complémentaires pour améliorer le modèle.

Pourquoi la performance des algorithmes standards baisse largement en face des données déséquilibrées ? La plupart des algorithmes de classification, comme Decision Tree et Logistic Regression, cherche à minimiser le taux d'erreur : le pourcentage de la prédiction incorrecte des étiquettes de classe. Ils ignorent la différence entre les différents types de classification erronée. En particulier, ils supposent implicitement que toutes les erreurs de classification représentent le même coût lors de l'apprentissage du modèle [Ganganwar, 2012].

Cependant, dans de nombreuses applications cette hypothèse n'est pas vraie. L'importance des classifications erronées entre différentes classes peuvent ne pas être identique. Par exemple, dans le diagnostic médical d'un certain cancer, si le cancer est considéré comme positif et non cancéreux comme négatif, alors la manque d'un cancer (le patient est réellement positif mais classé comme négatif) est beaucoup plus grave que la cas où on prend une personne ne portant pas le cancer comme cancéreux. Parce que le patient pourrait perdre la vie à cause du retard du diagnostic et du traitement, alors qu'à l'inverse des examens complémentaires permettront d'informer le faux positif.

Le traitement des jeux de données déséquilibrés implique souvent des stratégies telles que l'amélioration des algorithmes de classification ou l'équilibrage des différentes classes du corpus avant de fournir les données d'entraînement à l'algorithme d'apprentissage automatique.

2.1 Approches au niveau des données

L'objectif principal des approches de ré-échantillonnage est d'augmenter la fréquence de la classe minoritaire ou de diminuer la fréquence de la classe majoritaire. Ceci est fait afin d'obtenir approximativement le même nombre d'instances pour les classes.

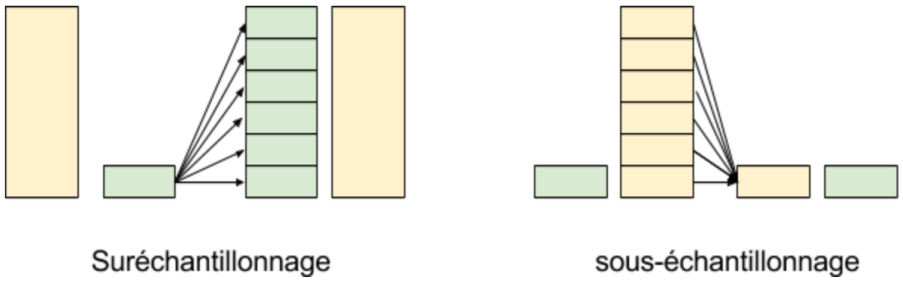


FIGURE 2.1 – Méthodes de ré-échantillonnage

2.1.1 Sur-échantillonnage : Smote

La procédure de sur-échantillonnage est d'ajouter des données sur la classe minoritaire pour équilibrer la distribution des classes. La méthode adoptée est simple au début comme le Random Over Sampling (ROS) qui équilibre la distribution des données par la duplication des exemples de la classe minoritaire au hasard. Cependant, l'approche présente l'inconvénient de conduire à un sur-apprentissage et de nombreuses recherches ont été menées pour résoudre le problème.

Smote [Chawla et al., 2002] est une méthode de sur-échantillonnage dans laquelle la classe minoritaire est sur-échantillonnée en créant des exemples «synthétiques». Il est un des algorithmes les plus utilisés pour améliorer la performance de classifieurs appliqués sur les données déséquilibrées. Beaucoup d'autres algorithmes sont développés à base de Smote, comme Bordline-Smote [Han et al., 2005] et Random-Smote [Li et al., 2011a].

Ce que Smote fait, c'est de fournir un ensemble de règles simples pour générer de nouvelles données «synthétisées». Voici comment il fonctionne :

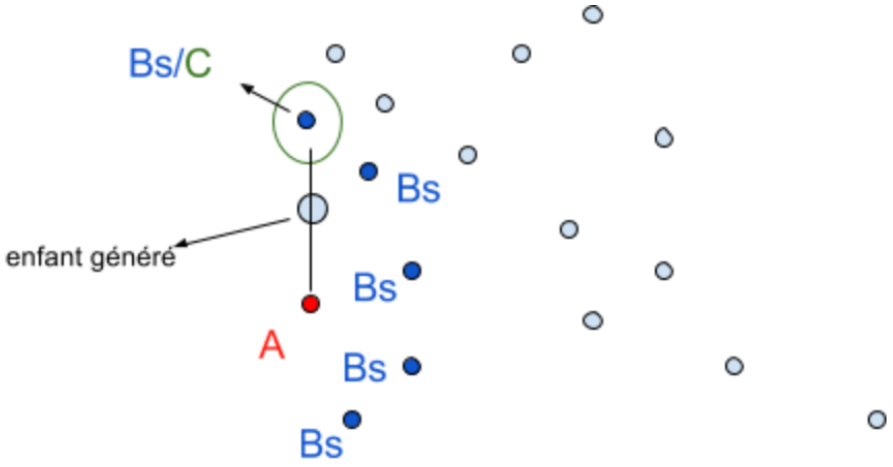


FIGURE 2.2 – Visualization de l'algorithme Smote

1. Choix d'une donnée A aléatoirement parmi les données ;
2. Recherche de k données Bs proches de A ;
3. Choix d'une donnée B à renommer C, une ligne imaginaire est tracée entre A et C ;
4. Génération d'une donnée sur cette ligne imaginaire à ajouter aux données d'origine ;
5. Itération du processus.

La formule pour générer les données synthétiques par Smote est la suivante :

$$D_{new} = D_i + (\hat{D}_l - D_i) \times \delta$$

Dans la formule, D_{new} est la donnée synthétisée, D_i est le point rouge («A» dans notre figure) et $\hat{D}_l$ est l'un des K voisins les plus proches de D_i. δ est un chiffre aléatoire entre 0 et 1.

Bien que chaque nouvelle donnée synthétique soit construite à partir de ses parents (la donnée choisie et l'un de ses voisins les plus proches), la donnée générée n'est jamais un double exact de l'un de ses parents.

2.1.2 Sur-échantillonnage : Adasyn

Adasyn (Adaptive synthetic sampling) a été proposé en 2008 par H. Haibo et Y. Bai [He et al., 2008]. L'idée essentielle d'Adasyn est d'utiliser une distribution pondérée pour différents groupes de la classe minoritaire en fonction de leur niveau de difficulté d'apprentissage. Plus les données sont difficiles à apprendre, plus de données synthétiques vont être générées. L'approche Adasyn améliore l'apprentissage par rapport aux distributions de données par deux façons : réduire le biais introduit par le déséquilibre de classe et déplacer de façon adaptative la limite de classification à l'égard des exemples difficiles à apprendre. Voici le processus de l'algorithme Adasyn :

1. Calcul du niveau de déséquilibre entre les classes ;

2. Calcul du nombre de données synthétisées à générer pour la classe minoritaire ;
3. Pour chaque groupe de classe minoritaire x_i, trouver les K voisins les plus proches en calculant la distance euclidienne et calcul de la distribution de densité ;
4. Calcul du nombre de données à générer pour x_i ;
5. Pour chaque exemple de données de classe minoritaire x_i, génération des exemples de données synthétiques.

La différence principale entre Adasyn et Smote est que le dernier génère le même nombre d'échantillons de données synthétiques pour chaque groupe de la classe minoritaire et le fait sans tenir compte des exemples voisins, ce qui augmente l'occurrence de chevauchements entre les classes [Wang and Japkowicz, 2004].

2.1.3 Sous-échantillonnage : Extraction of majority-minority Tomek links

Le sous-échantillonnage consiste à réduire ou éliminer certaines données sur la classe majoritaire pour équilibrer la distribution des classes. L'algorithme de base s'appelle RUS (Random Under Sampling) qui permet de réduire les données sur la classe majoritaire au hasard. Diverses méthodes ont été développées pour améliorer le résultat : Neighborhood Cleaning Rule [Laurikkala, 2001], Extraction of majority-minority Tomek links [Tomek, 1976], Instance Hardness Threshold [Smith et al., 2014].

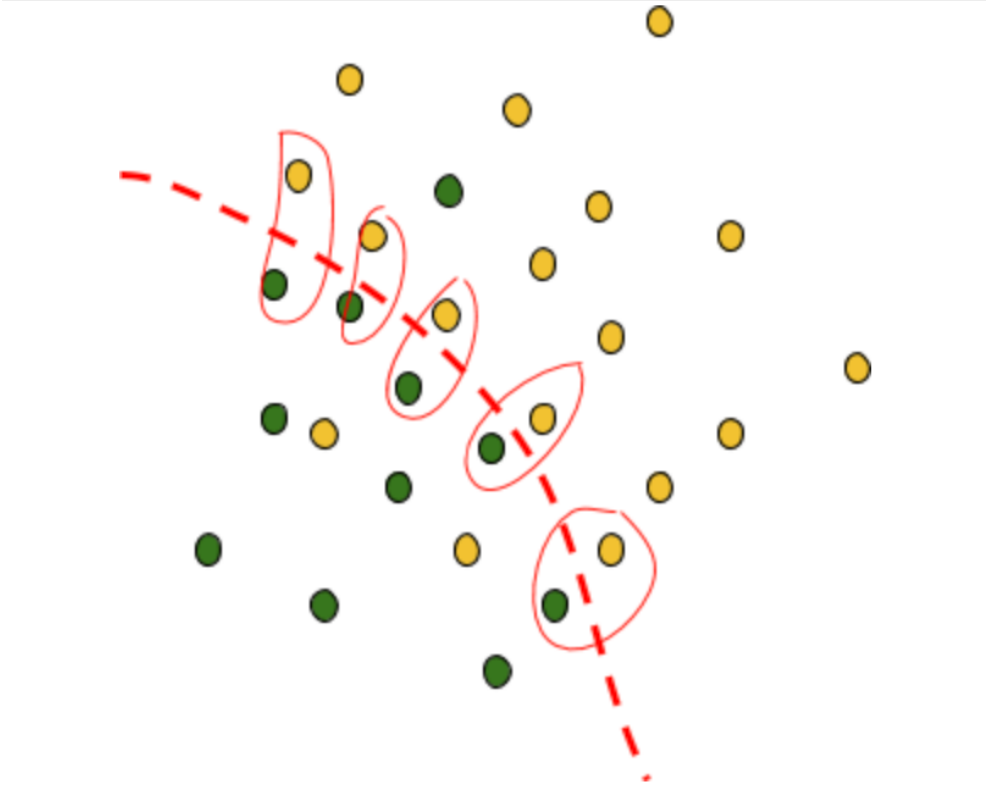


FIGURE 2.3 – Visualization de l'algorithme Tomek Links

Les «Tomek Links» sont en effet les cercles rouges montrés dans l'image au-dessus. Plus précisément, ce sont des paires de données qui sont les plus proches et qui sont autour de la ligne de séparation. Cela signifie que ce sont les données qui vont donner le plus de problèmes dans la plupart des algorithmes de classification. En supprimant ces paires de données, la séparation entre les deux classes sera élargie, de sorte que notre algorithme de classification fera moins d'erreurs.

2.1.4 Hybride : Smote-TL

La méthode hybride combine les approches de sur-échantillonnage et celles de sous-échantillonnage en éliminant des données dans la classe majoritaire et ajoutant des données dans la classe minoritaire afin de balancer la distribution des classes [Santoso et al., 2017].

Batista et R. Prati [Batista et al., 2004] ont comparé dix techniques pour traiter la classification asymétrique. Leurs expériences prouvent que le déséquilibre des classes n'entrave pas systématiquement la performance des systèmes d'apprentissage. Ils pensent que le problème est lié à l'apprentissage avec trop peu d'exemples de classes minoritaires et la présence d'autres facteurs plus complexes, tels que le chevauchement de classes. Et ils ont proposé deux méthodes hybrides Smote-TL et Smote-ENN qui intègrent un nettoyage supplémentaire des exemples bruyants après le processus Smote afin de produire des clusters de classe mieux définis.

L'approche Smote-TL est la combinaison des algorithmes Smote et Tomek Links. Elle a d'abord été utilisée pour améliorer la classification des exemples sur le problème de l'annotation des protéines en bioinformatique [Batista et al., 2003].

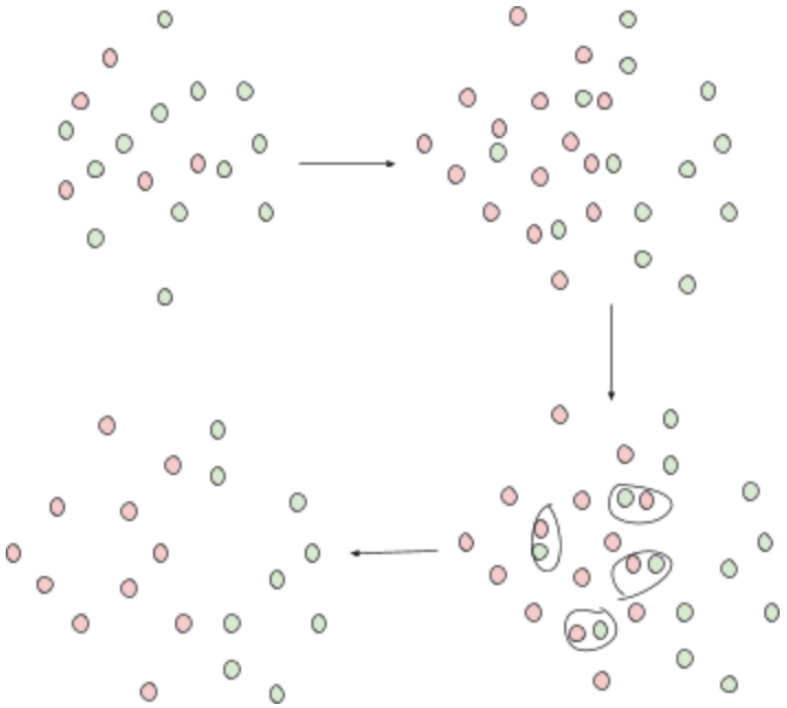


FIGURE 2.4 – Visualization de l'algorithme Smote-TL

Il existe d'autres méthodes de ré-échantillonnage plus complexes que celles que

nous avons utilisées. Cependant, les méthodes simples sont parfois les plus efficaces [Van Hulse and Khoshgoftaar, 2009].

2.2 Approches algorithmiques

Alors que les méthodes de ré-échantillonnage tentent d'équilibrer les distributions en considérant les proportions des exemples de classes, les méthodes fondées sur les algorithmes se concentrent sur l'adaptation des classifieurs. Il y a trois idées essentielles de solutions à ce niveau :

- Ajuster le poids de la classe (coût des erreurs de classification) ;
- Ajuster le seuil de décision ;
- Modifier un algorithme existant pour être plus sensible aux classes rares.

En considération du coût de la réalisation et de l'implémentation, dans nos travaux, nous avons adopté la première idée : utiliser un apprentissage sensible aux coûts qui tient compte des coûts associés aux exemples mal classés [Ting, 2002]. Plus concrètement, au lieu de créer des distributions de données équilibrées à l'aide de différentes stratégies de ré-échantillonnage, l'apprentissage sensible aux coûts cible le problème d'apprentissage déséquilibré en utilisant des matrices de coûts qui décrivent les coûts de classification erronée.

La matrice de coûts est similaire à une matrice de confusion. Pour une classification binaire, elle se concentre sur les faux positifs et les faux négatifs (voir la figure ci-dessous). Il n'y a aucun ajustement de poids associé aux vrais positifs et vrais négatifs car ils sont correctement identifiés.

Réel	Prédit		
		Positive	Négative
	Positive	0	C(FN)
	Négative	C(FP)	0

TABLE 2.1 – Matrice de coûts

Cette méthode vise à paramétrer un classifieur avec le coût total le plus bas :

$$Coûttotal = C(FN) \times FN + C(FP) \times FP$$

Afin de minimiser le coût total, beaucoup de boîtes à outils d'apprentissage automatique proposent des moyens d'ajuster «l'importance» des classes minoritaires. Scikit-learn, par exemple, propose de nombreux classifieurs qui prennent un paramètre optionnel «class weight» qui peut être spécifié. Voici un exemple, tiré directement de la documentation de Scikit-learn [1], montrant l'effet de l'augmentation du poids de la classe minoritaire par dix. La ligne noire continue représente la bordure séparatrice lors de l'utilisation des paramètres par défaut (les deux classes sont pondérées de manière égale) et la ligne rouge est le séparateur après l'utilisation du paramètre «class weight» qui a été modifié de 1 en 10.

1. http ://scikit-learn.org/stable/auto_examples/svm/plot_separating_hyperplane_unbalanced.html

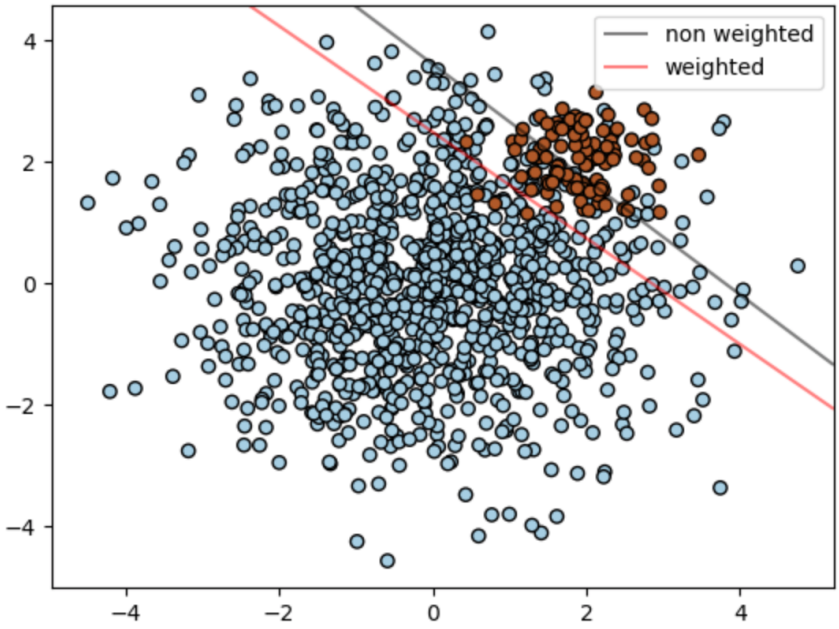


FIGURE 2.5 – Visualization de la fonction du paramètre «classe weight»

Comme ce qui est montré dans l'image, en ajoutant le paramètre «class weight», la classe minoritaire gagne de l'importance, vu que ses erreurs sont considérées plus coûteuses que celles de l'autre classe et le séparateur est ajusté pour réduire la perte.

Il faut aussi noter que l'ajustement de l'importance des classes n'a généralement d'influence directe que sur le coût des erreurs de classe (faux négatifs, si la classe minoritaire est positive). De ce fait, si le classifieur ne fait pas d'erreur sur la classification de la classe minoritaire, la modification des poids de classe peut n'avoir aucun effet.

Dans notre choix de la technique pour résoudre le problème généré par le déséquilibre des données, si on a donné une préférence aux approches de sur-échantillonnage, c'est parce qu'elles gardent plus d'informations originelles. Il n'y a pas de gagnant définitif entre l'apprentissage au niveau de l'algorithme, le sur-échantillonnage et le sous-échantillonnage. Tout dépend du corpus [Weiss et al., 2007].

Deuxième partie

Expérimentations

CHAPITRE 3

CORPUS

Sommaire

3.1 Pourquoi les avis des consommateurs ? 29
3.2 Introduction de la parfumerie en France 29
3.3 Collecte des données et prétraitements 30
3.4 Règles d'annotation 31

3.1 Pourquoi les avis des consommateurs ?

Le sentiment concerne essentiellement les attitudes, émotions et opinions envers une entité qui représente un thème ou un événement. Et ces sujets sont très souvent mentionnés dans les avis des consommateurs [Panchal et al., 2017].

Selon H. Ana María [Hornero et al., 2008], les avis des consommateurs servent à évaluer soit un produit spécifique qui a récemment été publié, soit plusieurs produits de la même catégorie. La recommandation des produits peut aussi apparaître dans les avis.

Bien que dans un forum sur la beauté et même sur Twitter on puisse trouver beaucoup de conversations à propos de parfums, ces types de données en ligne présentent plusieurs défauts qui entravent potentiellement le processus d'analyse du sentiment. Puisque les gens peuvent publier librement leur propre contenu, la qualité rédactionnelle de leurs commentaires ne peut pas être garantie. Par exemple, au lieu de partager des opinions liées aux sujets, les vendeurs diffusent des publicités sur les forums. Dans la plupart des sites d'avis, les commentaires publiés ont tous modérés par un ou plusieurs community managers. De plus, l'organisation et la structure du site d'avis facilitent la récupération et l'utilisation des données.

3.2 Introduction de la parfumerie en France

La parfumerie désigne l'art et l'industrie de la fabrication de parfums dont l'origine peut remonter à la civilisation en Mésopotamie. Même si la France n'est pas le premier pays au monde à inventer le parfum, elle est réellement le lieu de naissance du parfum moderne. Jusqu'à aujourd'hui, la France est quand même le centre du commerce et du design de parfum en Europe : un grand nombre des plus grandes marques de l'industrie de parfum, comme Chanel, Dior et Guerlain, sont d'origine française. Et en termes de ventes de parfums internationales, la France est le leader,

avec 30% du marché mondial [1].

Le marché du parfum en France est un des secteurs les plus importants dans l'industrie de la santé et la beauté du pays avec 4,666 Md€. chiffre d'affaires en 2014 [2] et il se développe dans le contexte d'un environnement extrêmement concurrentiel.

Afin de maintenir et augmenter leur position dans le marché, des investissements importants ont été faits par les marques dans la publicité et la promotion, ainsi que leurs produits et l'innovation du flacon. Le comportement du consommateur a aussi été analysé à dessin d'ajuster en temps voulu la stratégie de marché.

Lorsque nous inhalons des molécules odorantes du parfum, non seulement la sensation de l'odeur est créée, mais aussi des émotions et des expériences y associent. Et ces sentiments qui ont une influence significative sur le comportement du consommateur sont souvent exprimés dans les avis laissés sur les sites d'achat.

3.3 Collecte des données et prétraitements

Notre corpus est constitué de commentaires en français sur le site d'avis http ://www.beaute-test.com. Au lieu d'écrire le script pour récupérer les données, on a utilisé le service proposé par l'entreprise Import io [3], une entreprise américaine visant à convertir la masse de données sur les sites Web en données structurées et exploitables par des logiciels. Leur plate-forme efficace et évolutive permet aux utilisateurs de traiter des milliers d'URL très rapidement et d'accéder à des millions de lignes de données qu'ils utilisent pour différents objectifs.

http ://www.beaute-test.com est un site d'avis spécialisé dans les produits de beauté. Sur le site et dans la rubrique produit, une espace est attribué aux parfums qui est classé en sous-catégories : parfums femme, parfums hommes, parfums mixtes et parfums kids. Par rapport aux autres sites du même domaine, comme ciao.fr [4], amazon.fr [5] et fragrantica.fr [6], beaute-test.com [7] est notre premier choix. C'est non seulement parce que c'est un site professionnel et très réputé dans le domaine (beaucoup de consommateurs le consultent avant de faire leur achat et laissent leur avis sur le produit), mais aussi parce que ce sont plutôt des consommateurs qui laissent des avis courts mais pertinents. Contrairement, aux sites comme ciao.fr, on voit souvent des documentations à l'égard de l'évaluation de tous les aspects du parfum, ne contenant guère de sentiments personnels, et probablement écrits par les experts en parfumerie.

Nous avons extrait 20K verbatims sur le site et en avons sélectionné aléatoirement 9180 pour construire notre corpus. La longueur du commentaire varie largement : de 1 mot à 2514 mots.

Positive	Négative	Total
907	8273	9180

TABLE 3.1 – Statistiques sur le corpus

1. Source : https ://about-france.com/tourism/french-perfume.htm
2. Source : INSEE, ESANE
3. https ://www.import.io
4. http ://www.ciao.fr
5. https ://www.amazon.fr
6. https ://www.fragrantica.fr
7. http ://www.beaute-test.com

3.4 Règles d'annotation

L'implication durable est une relation consommateur-objet stable basée sur les besoins inhérents du consommateur. Lorsque le comportement du consommateur tend à un objectif à long terme [Ogbeide, 2014] ou reflète les sentiments durables à l'égard d'un produit ou d'une catégorie [Sirgy et al., 2014]. Selon ces explications, on a pris trois types d'expressions comme des signaux dénotant l'implication durable :

- l'expression de l'intention d''une utilisation prolongée :
  «Je porte ce parfum depuis plus de 6 ans et je n'en demords toujours pas...j'adore»
- l'expression du rachat :
  «J'ai toujours un flacon chez moi!!!! C'est mon parfum préféré que je rachèterai encore et encore .....!!!!!!»
- l'expression d'attachement très forte au niveau de l'adoption :
  «Une fragance fruitée subtile et enivrante je recommande ce parfum vous ne serez pas déçue l'essayer une fois c'est l'adopter!!!»

La figure 3.1 montre des exemples d'expressions pour repérer les signaux demandés.

Expressions de l'intention dune utilisation prolongée	Expressions de rachat	Expression de l'adoptation
ne change plus	acheterai encore	adopté
ai toujours un flacon	rachèterais	adoptation
reviens toujours	acheter à nouveau	adopter
c'est mon 4 eme flacon	reprendrai	
plusieurs flacons	y retourner	
encore fidèle		
3 fois que je l'achète		
ne peux m'en passer		
difficilement l'abandonner		
depuis bientôt 3 ans		
ne se quitte plus		
je ne le lâche plus		

FIGURE 3.1 – Exemples d'expressions contenant les signaux demandés

Ces trois types d'expressions se mélangent quelques fois dans un même commentaire et les expressions qui portent le même sens varient largement d'un auteur à l'autre. D'ailleurs, les signaux positifs de EIs ne sont pas toujours évidents dans les avis publiés. Comme dans les deux phrases «Un parfum floral oriental par excellence!! On ne peut plus s'en passer», l'EI positive n'est pas directement exprimée par des mots ou expressions comme «racheter» et «adopté», mais il y a une intention

très forte d'un usage prolongé comprise dans la phrase. D'ailleurs, toutes les phrases contenant les expressions qui ont les mêmes sens que les expressions prédéfinies par nos règles d'annotation ne portent pas vraiment les signaux à détecter. Par exemple, le verbatim « je l'ai utilisé depuis longtemps mais je l'aime plus» pourrait être détectée à tort comme positive.

Si l'expression d'affection très forte au niveau de l'adoption est choisie comme un règle d'annotation au lieu de l'ensemble de concepts contenant le sentiment d'admiration, c'est parce qu'il faut éviter des ambiguïtés et que c'est mieux d'avoir une frontière plus claire avec d'autres notions en marketing, comme la préférence. De ce fait, les avis de produits avec un sentiment positive assez fort comme «Je suis tomber amoureuse de ce parfum pour les filles qui aime les notes sucrés je conseil vivement» et «Un de mes parfums préférés!!!» sont annotés en «négative».

CHAPITRE 4

CLASSIFIEUR STANDARD APPLIQUÉ

Sommaire

4.1 Machine à vecteurs de support (SVM) 33
4.2 Méthodes d'extraction des caractéristiques implémentées 35

4.1 Machine à vecteurs de support (SVM)

La machine à vecteurs de support (SVM) est une technique d'apprentissage supervisé applicable à la fois à la classification et à la régression. Enraciné dans la théorie de l'apprentissage statistique développée par Vladimir Vapnik et ses collègues des laboratoires AT&T Bell en 1995 [Vapnik, 1995], SVM est basé sur le principe de la minimisation des risques structurels.

A l'origine, SVM a été élaboré pour la classification linéaire à deux classes. L'idée derrière est assez simple. SVM cherche à trouver un hyperplan de séparation optimal où la marge, la distance minimale entre ce séparateur et les points de données les plus proches (SVs), est maximale [Suykens, 2003].

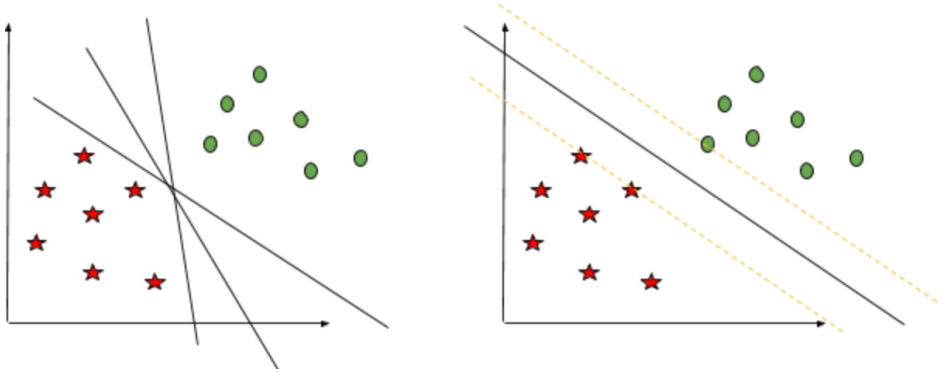


FIGURE 4.1 – Hyperplans de SVM

En effet, pour une classification linéaire simple, il existe généralement plusieurs séparateurs qui peuvent séparer les données parfaitement, comme ce qui est montré dans la figure 4.1. Néanmoins, intuitivement, un hyperplan se trouvant au milieu du vide entre les éléments de données des deux classes semble meilleur que les autres proches des éléments d'une ou des deux classes qui sont sensibles aux bruits.

Selon les expériences de Yang et Liu [Yang and Liu, 1999], SVM linéaire peut être étendu à une classification non-linéaire lorsque les points de données sont transformés en caractéristiques dans un même espace en utilisant un ensemble de fonctions non-linéaires. Et cet espace de caractéristiques peut être de très grande dimension.

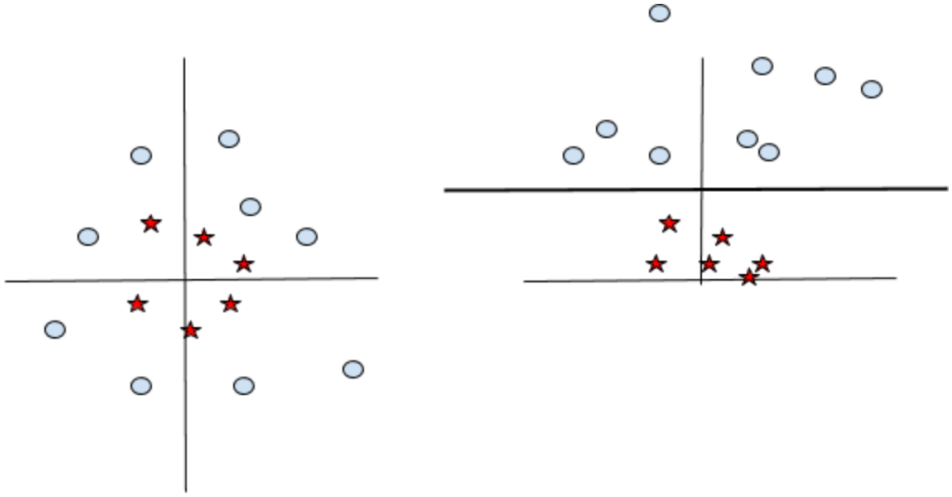


FIGURE 4.2 – Classification Non-lineaire

Par exemple, dans la figure au-dessus à gauche, il n'y a pas de frontière linéaire (une ligne droite qui sépare les deux catégories). Cependant, les vecteurs sont clairement séparés et il semble qu'il devrait être facile de les séparer. Quand nous ajoutons une troisième dimension, nous obtenons un espace en trois dimensions et une tranche dans cet espace, montré dans la figure au-dessus à droite est notre séparateur cherché. Dans ce cas, l'hyperplan n'est plus une ligne mais une surface.

Dans notre exemple, nous avons trouvé un moyen de classer des données non-linéaires en projetant intelligemment notre espace à une dimension supérieure. Cependant, il s'avère que le calcul de cette transformation peut être très coûteux : quelques fois, on doit ajouter beaucoup de dimensions et chacune d'entre elles implique un calcul complexe. Heureusement, il n'est pas forcément nécessaire d'implémenter cette transformation. Au lieu de déterminer l'hyperplan dans un espace en multiples dimensions, une représentation du noyau peut être utilisée et la solution est écrite comme une somme pondérée des valeurs de certaines fonctions du noyau évaluées sur les vecteurs de support.

La fonction de l'astuce du noyau (*kernel trick* en anglais) a été proposé par Aizerman, Braverman et Rozoner [Aizerman, 1964] pour résoudre la classification linéairement inséparable. En bref, un noyau est un raccourci qui nous aide à effectuer certains calculs plus rapidement en évitant des calculs dans un espace de haute dimension. Sa formule mathématique est suivante :

$$K(x, y) = \phi(x) \cdot \phi(y)$$

Dans la formule, K est la fonction noyau qui prend x et y comme entrées en n-dimensions. Normalement, pour calculer $\phi(x) \cdot \phi(y)$, il faut d'abord calculer séparément $\phi(x)$ et $\phi(y)$, et puis, faire le produit scalaire. Cependant, après ces manipulations compliquées dans l'espace de haute dimension, notre résultat est vraiment un

simple scalaire, Pour éviter d'impliquer les calculs compliqués ou même impossibles, on doit trouver un noyau pertinent à utiliser.

Comparé avec d'autres classifieurs standards, SVM est plus précis sur des données modérément déséquilibrées. Parce que seulment les vecteurs de support sont utilisés pour la classification et que de nombreux échantillons majoritaires éloignés du hyperplan de séparation peuvent être supprimés sans affecter la classification [Akbani et al., 2004]. Cependant, un classificateur SVM peut être sensible à un déséquilibre large entre les classes, ce qui entraîne une baisse des performances de classification sur la classe positive. Il est enclin à générer un classifieur qui a un fort biais d'estimation vers la classe majoritaire, résultant en un grand nombre de faux négatifs.

4.2 Méthodes d'extraction des caractéristiques implémentées

La sélection des caractéristiques est un processus de transformation des données brutes en fonctions qui représentent mieux le problème sous-jacent aux modèles prédictifs, ce qui se traduit par une précision améliorée du modèle sur des données non vues.

Dans notre implémentation, nous avons choisi le modèle TF-IDF.

Le TF-IDF (de l'anglais *term frequency-inverse document frequency*) est une méthode de pondération proposé par Salton et Buckley [Salton and Buckley, 1988], souvent utilisée en recherche d'information et en particulier dans la fouille de textes. La justification théorique a posteriori de ce schéma de pondération repose sur l'observation empirique de la fréquence des mots dans un texte qui est donnée par la loi de Zipf. Il est souvent utilisé comme une mesure statistique pour questionner l'importance d'un mot dans un document.

Afin de calculer le TF-IDF, il faut dans un premier temps savoir respectivement les valeurs TF et IDF. Et leur produit est le résultat final. Le TF mesure la fréquence d'apparition d'un terme dans un document. Étant donné que chaque document est de longueur différente, il est possible qu'un terme apparaisse beaucoup plus souvent dans les documents longs que dans les documents plus courts. Par conséquent, comme un moyen de normalisation, la fréquence de terme peut être divisée par la longueur du document :

$$F(t) = \frac{(Nombre\ de\ fois\ que\ le\ terme\ t\ apparaît\ dans\ un\ document)}{(Nombre\ total\ de\ termes\ dans\ le\ document)}$$

Le IDF (Inverse Document Frequency) mesure l'importance d'un terme. Lors du calcul de TF, tous les termes sont considérés comme importants au niveau parail, mais on sait que certains termes, tels que "le", "est" et "de", peuvent apparaître plusieurs fois mais avoir peu d'importance, il faut donc pénaliser les termes fréquents qui apparaissait dans de nombreux documents :

$$IDF(t) = \log_e \frac{(Nombre\ total\ de\ documents)}{Nombre\ de\ documents\ contenant\ le\ terme\ t}$$

CHAPITRE 5

EXPÉRIENCES ET RÉSULTATS

Sommaire

5.1 Méthode d'évaluation 37
5.2 Sélection du noyau et des hyper-paramètres 38
5.3 Baseline sans adaptation liée à la classification asymétrique 41
5.4 Ajuster le poids des classes 43
5.5 Comparaison des méthodes de ré-échantillonnage 44
5.6 Comparaison entre les méthodes par ré-échantillonnage et les méthodes algorithmiques 46
5.7 Affinage du modèle 47
5.8 Expériences avec LSTM 49

Toutes nos expériences ont été faites en utilisant Scikit-learn, une librairie d'apprentissage automatique pour le langage de programmation Python. Il comporte divers algorithmes de classification, de régression et de clustering, y compris les machines à vecteur de support, les forêts aléatoires, le boosting de gradient, k-means et DBSCAN. Il est conçu pour inter-opérer avec les librairies numériques et scientifiques Python telle que NumPy et SciPy.

5.1 Méthode d'évaluation

La sélection d'une méthode pertinente pour évaluer la performance de l'algorithme est une étape essentielle dans l'apprentissage avec des données déséquilibrées. De nombreuses matrices ont été utilisées et toutes sont basées sur la matrice de confusion comme ce qui est indiqué dans le tableau suivant.

Réel	Prédit		
		Positive	Négative
	Positive	True Positif (TP)	Faux Négatif(FN)
	Négative	Faux Positif (FP)	True Négatif (TN)

TABLE 5.1 – Matrice de Confusion

Avec une distribution de données fortement asymétrique, l'évaluation selon l'exactitude (*accuracy*) qui est calculée comme la méthode d'évaluation de la plupart des algorithmes de classification en fonction du pourcentage d'observations correctement classées n'est plus suffisant. Par exemple, pour un jeu de données très déséquilibré, un classificateur naïf qui prédit tous les échantillons comme négatifs obtiendra une

haute exactitude. Cependant, il sera totalement inutile pour détecter des échantillons positifs rares.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Pour faire face au problème généré par le déséquilibre entre les classes et obtenir un modèle de classification asymétrique optimal, la sensibilité et la spécificité sont habituellement adoptées pour surveiller respectivement la performance de classification sur deux classes. Cependant, parfois, on s'intéresse à la capacité de détecter efficacement une seule classe. Pour de tels problèmes, une autre paire de mesures, la précision et le rappel, est souvent adoptée [Tang et al., 2009]. Notez que le rappel est la même chose que la sensibilité. la f-mesure est aussi souvent utilisée pour intégrer la précision et le rappel dans une seule mesure pour la commodité de l'évaluation [Van Rijsbergen, 1979].

$$Précision = \frac{TP}{(TP + FP)}$$

$$Rappel = \frac{TP}{(TP + FN)}$$

$$F - Mesure = \frac{2 \times précision \times rappel}{précision + rappel}$$

Notez que tous nos scores (rappel, précision et f-mesure) sont seulement ceux de la classe ciblée.

Dans nos expériences, la sélection et l'optimisation du modèle sont faites en condition de la f-mesure. Cependant, il faut savoir que dans l'application métier, c'est souvent l'objective métier ou la demande du client qui décide du choix de modèle. Par exemple, dans le travail de la filtration du spam (la classe positive est spam), la sélection et l'optimisation du modèle sont souvent faites selon la précision. Parce que les faux négatifs (le spam va dans la boîte de réception) sont plus acceptables que les faux positifs. Or, dans la détection de transactions frauduleuses, le rappel est souvent pris comme la métrique, car les faux positifs (transactions normales signalées comme frauduleuses) sont plus acceptables que les faux négatifs (transactions frauduleuses non détectées).

5.2 Sélection du noyau et des hyper-paramètres

Les hyper-paramètres sont des paramètres dont les valeurs sont définies avant le début du processus d'apprentissage. Ils varient selon les modèles. Certains algorithmes simples, comme la régression des moindres carrés ordinaires n'en exigent aucun tandis que d'autres en demandent beaucoup, tel que le réseau de neurones.

Pour SVM, il existe un certain nombre de paramètres d'apprentissage qui peuvent être utilisés pour la régression parmi lesquels le paramètre de pénalité «C», qui détermine le compromis entre l'erreur d'apprentissage et la dimension de l'hyperlpan du modèle est le plus important. D'autres hyper-paramètres peuvent être proposés selon le type de noyau choisi. Dans notre travail, le noyau linéaire et le noyau gaussien qui sont largement utilisés pour SVM ont été testés.

La recherche manuelle et la recherche par grille sont les stratégies les plus largement utilisées pour l'optimisation des hyper-paramètres. Les expériences de James Bergstra et Yoshua Bengio [Bergstra and Bengio, 2012] montrent empiriquement et

théoriquement que les essais choisis au hasard sont plus efficaces pour l'optimisation des hyper-paramètres que les essais sur une grille. De ce fait, on a utilisé l'outil GridSearchCV proposé par Sk-learn qui permet d'optimiser les paramètres en comparant de manière exhaustive toutes les combinaisons de paramètres et les noyaux afin d'optimiser nos hyper-paramètres. Voici notre liste de candidates qui ont été choisis selon les expériences :

```
    parameter_candidates = [
  {'C': [0.2, 0.5, 1, 5, 10, 20, 50, 100], 'kernel': ['linear']},
  {'C': [0.2, 0.5, 1, 5, 10, 20, 50, 100], 'gamma': [1e-3, 1e-4], 'kernel': ['rbf']},
]
```

FIGURE 5.1 – Paramètres Candidats

Intuitivement, le paramètre «Gamma» définit à quel point l'influence d'un seul exemple d'entraînement peut influencer l'apprentissage, avec des valeurs faibles signifiant «léger» et des valeurs élevées signifiant «fort». Le paramètre «C» écarte la classification erronée des exemples d'apprentissage par rapport à la simplicité de la surface de décision. Un «C» faible rend la surface de décision lisse, alors qu'un «C» élevé vise à classer correctement tous les exemples d'entraînement en donnant au modèle la liberté de sélectionner plus d'échantillons comme vecteurs de support.

```
# Tuning hyper-parameters for precision
Best parameters set found on development set:

{'C': 50, 'gamma': 0.001, 'kernel': 'rbf'}

Grid scores on development set:

0.905 (+/-0.078) for {'C': 0.2, 'kernel': 'linear'}
0.888 (+/-0.068) for {'C': 0.5, 'kernel': 'linear'}
0.868 (+/-0.075) for {'C': 1, 'kernel': 'linear'}
0.812 (+/-0.056) for {'C': 5, 'kernel': 'linear'}
0.794 (+/-0.051) for {'C': 10, 'kernel': 'linear'}
0.778 (+/-0.054) for {'C': 20, 'kernel': 'linear'}
0.765 (+/-0.058) for {'C': 50, 'kernel': 'linear'}
0.761 (+/-0.058) for {'C': 100, 'kernel': 'linear'}
0.448 (+/-0.000) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 5, 'gamma': 0.001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 20, 'gamma': 0.001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 20, 'gamma': 0.0001, 'kernel': 'rbf'}
0.950 (+/-0.002) for {'C': 50, 'gamma': 0.001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 50, 'gamma': 0.0001, 'kernel': 'rbf'}
0.905 (+/-0.078) for {'C': 100, 'gamma': 0.001, 'kernel': 'rbf'}
0.448 (+/-0.000) for {'C': 100, 'gamma': 0.0001, 'kernel': 'rbf'}
```

```
# Tuning hyper-parameters for recall

Best parameters set found on development set:

{'C': 20, 'kernel': 'linear'}

Grid scores on development set:

0.569 (+/-0.038) for {'C': 0.2, 'kernel': 'linear'}
0.639 (+/-0.044) for {'C': 0.5, 'kernel': 'linear'}
0.683 (+/-0.041) for {'C': 1, 'kernel': 'linear'}
0.731 (+/-0.019) for {'C': 5, 'kernel': 'linear'}
0.727 (+/-0.041) for {'C': 10, 'kernel': 'linear'}
0.731 (+/-0.042) for {'C': 20, 'kernel': 'linear'}
0.728 (+/-0.044) for {'C': 50, 'kernel': 'linear'}
0.726 (+/-0.045) for {'C': 100, 'kernel': 'linear'}
0.500 (+/-0.000) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 5, 'gamma': 0.001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 20, 'gamma': 0.001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 20, 'gamma': 0.0001, 'kernel': 'rbf'}
0.519 (+/-0.020) for {'C': 50, 'gamma': 0.001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 50, 'gamma': 0.0001, 'kernel': 'rbf'}
0.568 (+/-0.036) for {'C': 100, 'gamma': 0.001, 'kernel': 'rbf'}
0.500 (+/-0.000) for {'C': 100, 'gamma': 0.0001, 'kernel': 'rbf'}
```

```
# Tuning hyper-parameters for f1
Best parameters set found on development set:

{'C': 5, 'kernel': 'linear'}

Grid scores on development set:

0.596 (+/-0.059) for {'C': 0.2, 'kernel': 'linear'}
0.691 (+/-0.056) for {'C': 0.5, 'kernel': 'linear'}
0.735 (+/-0.046) for {'C': 1, 'kernel': 'linear'}
0.763 (+/-0.028) for {'C': 5, 'kernel': 'linear'}
0.754 (+/-0.041) for {'C': 10, 'kernel': 'linear'}
0.751 (+/-0.041) for {'C': 20, 'kernel': 'linear'}
0.744 (+/-0.046) for {'C': 50, 'kernel': 'linear'}
0.741 (+/-0.046) for {'C': 100, 'kernel': 'linear'}
0.473 (+/-0.000) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 5, 'gamma': 0.001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 20, 'gamma': 0.001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 20, 'gamma': 0.0001, 'kernel': 'rbf'}
0.510 (+/-0.037) for {'C': 50, 'gamma': 0.001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 50, 'gamma': 0.0001, 'kernel': 'rbf'}
0.595 (+/-0.056) for {'C': 100, 'gamma': 0.001, 'kernel': 'rbf'}
0.473 (+/-0.000) for {'C': 100, 'gamma': 0.0001, 'kernel': 'rbf'}
```

FIGURE 5.2 – Rapport de la classification en utilisant les différents hyper-paramètres

Les résultats montrent que pour le noyau linéaire, la précision baisse lorsque la valeur "C" augmente, tandis que le rappel augmente jusqu'à C=100. Il y a un point pendant lequel ces deux mesures se croisent, qui doit être un point optimal en terme d'évaluation. Nous voyons aussi qu'en utilisant le noyau BRF, les valeurs de F-mesure ne varient pas beaucoup et elles sont toutes inférieures à celles qui utilisent le noyau linéaire. Comme nous avons besoin d'un modèle qui peut équilibrer les scores précision et rappel, on a choisi le noyau linéaire et le paramètre C = 5 pour notre baseline n'utilisant que l'algorithme SVM. Notez que l'ajustement des hyper-paramètres n'est fait que sur les données d'entraînement et que pour trouver le meilleur score de chaque méthode, les hyper-paramètres sont ajustés pour chaque configuration.

5.3 Baseline sans adaptation liée à la classification asymétrique

Pour savoir quelle est la meilleure technique pour classifier nos données largement déséquilibrées, on doit d'abord savoir le résultat obtenu sans utiliser ces techniques.

Il est courant que lors de l'exécution d'une expérience d'apprentissage automatique supervisée, si on apprend les paramètres d'une fonction de prédiction et que

l'on la teste sur les mêmes données, on peut se trouver confronté à une situation sur-apprentissage : un modèle qui répéterait simplement les étiquettes des échantillons que l'on vient de voir aurait un score parfait mais ne pourrait prédire quoi que ce soit sur les données non-vues.

Il faut donc séparer les données en deux parties pour constituer des ensembles d'apprentissage et de test, par exemple en utilisant la fonction «train test split» de Scikit-learn. Néanmoins, lors du développement du modèle, il existe toujours un risque de sur-ajustement sur l'ensemble de test et une troisième partie de données, les données de validation, est utile : l'apprentissage se déroule sur l'ensemble d'apprentissage, et puis l'évaluation est effectuée sur l'ensemble de validation. Et lorsque le modèle est stabilisé, l'évaluation finale peut être effectuée sur l'ensemble de test. Cependant, en partitionnant les données disponibles en trois ensembles, nous réduisons drastiquement le nombre d'échantillons pouvant être utilisés pour l'apprentissage du modèle, et les résultats peuvent dépendre d'un choix aléatoire particulier pour la paire d'ensemble d'entraînement et de validation. Pour éviter ce problème, on a utilisé le processus validation croisée dans l'expérimentation.

La validation croisée est une méthode d'évaluation qui consiste à diviser aléatoirement notre ensemble de données en k parties (dans notre test, k = 5). Chaque fois, (k-1) parties de données vont être utilisées comme échantillons d'apprentissage tandis que l'autre partie sera l'échantillon de validation. On répète cette opération jusqu'à ce que chacune des k parties aie servi de validation.

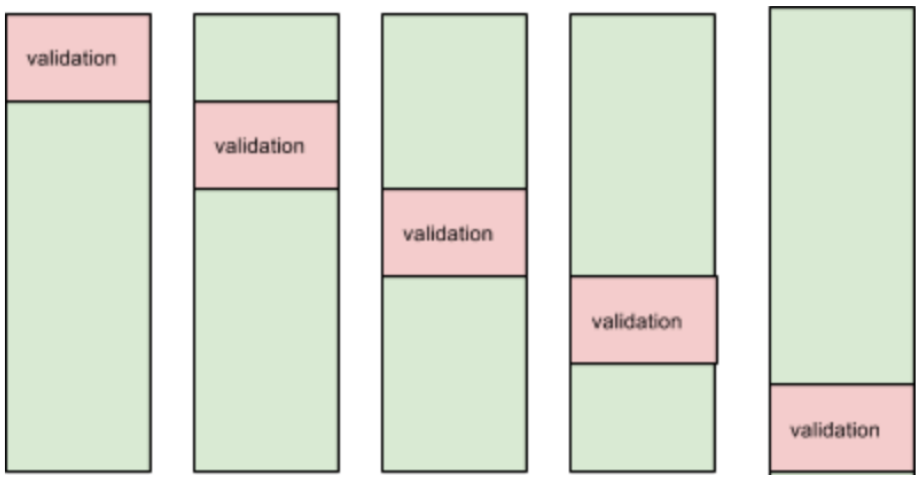


FIGURE 5.3 – Schéma de Validation Croisée

Avant d'entrer dans l'apprentissage, la tokenization et la suppression des mots vides ont été faites à l'aide de la librairie NLTK. Les résultats du modèle sans appliquer les techniques ciblant à la classification asymétrie sont montrés dans le tableau suivant :

Précision	Rappel	F-mesure	Temps
59.75%	62.50%	61.09%	1.66524s

TABLE 5.2 – Résultats en utilisant uniquement SVM

Les résultats du tableau 5.2 nous montre que, un modèle standard laisse des

marges de progression importantes sur notre corpus largement déséquilibré.

5.4 Ajuster le poids des classes

L'apprentissage sensible aux coûts est un type d'apprentissage en exploration de données qui prend en compte les coûts de mauvaise classification (et éventuellement d'autres types de coûts). Le but de ce type d'apprentissage reste à minimiser le coût total. La principale différence entre l'apprentissage sensible aux coûts et l'apprentissage insensible aux coûts est que l'apprentissage sensible aux coûts se focalise différemment les différentes erreurs de classification.

Scikit-learn propose une fonction afin de pénaliser les erreurs des échantillons appartenant à la classe minoritaire en ajoutant le paramètre «class weight» dans le modèle. Un poids de classe plus élevé signifie que l'on veut mettre plus d'importance sur une classe. Dans l'intention de trouver un meilleur poids pour notre corpus, on a essayé le poids de 1 fois à 15 fois à l'originel. Les résultats sont présentés dans le tableau suivant :

	Précision	Rappel	F-mesure
1 :1	67.48%	54.63%	60.36%
1 :2	63.20%	67.76%	65.40%
1 :3	57.22%	79.47%	63.12%
1 :4	54.96%	73.63%	62.71%
1 :5	53.491%	75.66%	62.68%
1 :6	52.73%	76.32%	62.37%
1 :7	51.54%	76.97%	61.74%
1 :8	54.38%	77.63%	61.30%
1 :9	53.29%	78.953%	61.70%
1 :10	52.69%	78.95%	61.54%
1 :11	49.79%	78.95%	61.87%
1 :12	49.59%	78.95%	61.38%
1 :13	49.60%	79.63%	61.11%
1 :14	49.60%	79.62%	61.11%
1 :15	49.19%	79.63%	65.50%

TABLE 5.3 – Résultats en paramétrant le poid de la classe minoritaire

La figure 5.4 montre l'évolution des métriques d'évaluation selon ce poids.

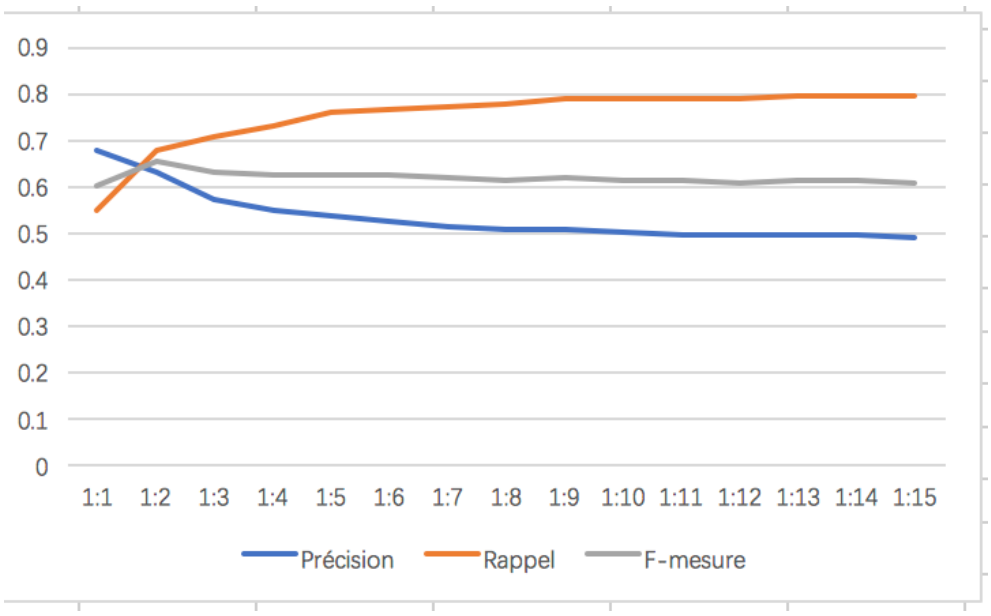


FIGURE 5.4 – Performance selon le poids de classe minoritaire

On peut remarquer dans la figure 5.4 que la précision augmente avec l'augmentation du poids sur la classe minoritaire. Et au contraire, le rappel baisse. Le score F-mesure arrive au pic lorsque le poids égale 2, et après, il a tendance à baisser mais très doucement. On peut aussi noter que la fluctuation de ces trois scores tend à être plate lorsque le poids est de plus en plus lourd. L'explication des tendances des scores est tout simple. Comme on a ajouté le poids sur la classification erronée de la classe minoritaire, pour minimiser le coût global, le modèle cherche à trouver un séparateur qui détecte plus de signaux demandés. Cela signifie que le faux négatif de la classification baisse et le rappel augmente. Au cours de la rechercher d'un hyperplan qui peut donner plus de «vrais positifs», le «faux positif» va aussi augmenter. Par conséquent, le rappel devient plus haut et la précision baisse.

On a pris finalement le poids 2, le point de l'intersection des trois lignes de scores, qui donne le meilleur résultat en terme de f-mesure

	Précision	Rappel	F1	Temps
SVM	59.75%	62.50%	61.09%	1.66524s
SVM+Ajustement du poids	63.20%	67.76%	65.40%	1.87307s

TABLE 5.4 – Comparaison entre les modèles SVM avec ou sans ajustement des coûts

On voit qu'en ajustant le poids sur les erreurs de classification de la classe minoritaire, bien que notre précision baisse un peu, le rappel et le F-mesure ont augmenté.

5.5 Comparaison des méthodes de ré-échantillonnage

L'implémentation des algorithmes ont été faite en utilisant la boît à outils «Imbalanced-learn» [Lemaitre et al., 2017]. Elle vise à fournir un large éventail de méthodes pour faire face au problème de jeu de données déséquilibré fréquemment rencontré dans l'apprentissage machine et la reconnaissance de formes. Elle dépend

uniquement de numpy, scipy et scikit-learn et elle est entièrement compatible avec scikit-learn, la librairie que nous utilisons pour implémenter notre classifieur.

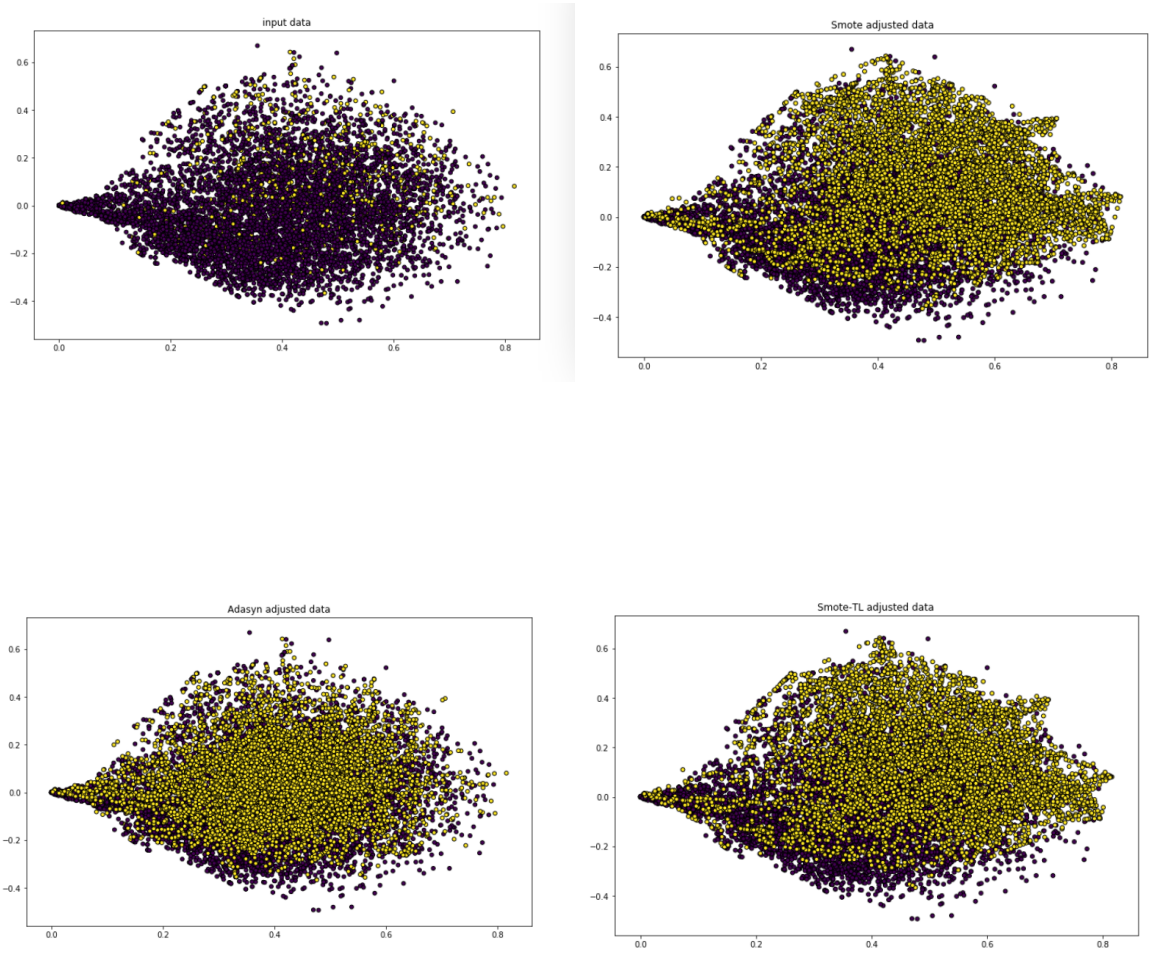
Dans nos expériences, on a testé quatres algorithmes au niveau de données : Smote, Adasyn, Tomek links et Smote-TL.

Voici le tableau qui montre le changement du nombre de données après avoir utilisé les quatres techniques complémentaires :

	Originel	Adasyn	TL	Smote	Smote+TL
nombre d'échantillons dans la classe minoritaire	907	6672	727	6559	6557
nombre total d'échantillons	9180	13231	7243	13118	13114

TABLE 5.5 – Changement du nombre d'échantillons

Pour avoir une vision plus directe :



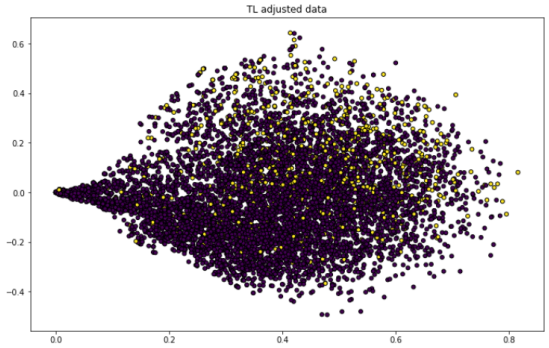


FIGURE 5.5 – Changement du nombre d'échantillons

Selon le tableau et les figures, le nombre de données de la classe minoritaire (en jaune) a beaucoup augmenté après avoir utilisé les algorithmes Smote, Adasyn et SmoteTL, tandis que de nombreuses données ont été éliminées en utilisant Tomek links. Ces changements dans les données sont directement liés aux méthodes utilisées. Or, il faut noter que Somte-TL et Smote nous avons donné presque le même nombre d'échantillons, alors que dans le cas où on n'utilise que Tomek links, le nombre d'échantillons a largement baissé. Cela signifie que l'étape de nettoyage de Smote-TL n'a guère fonctionné sur notre corpus et seulement 4 points de données ont été éliminés. Il faut aussi savoir que théoriquement, dans l'algorithme TL, les points de données éliminés appartenant tous à la classe minoritaire.

Le résultat au-dessous est la moyenne des résultats des méthodes sur 10 itérations.

	SVM	SVM+Smotes	SVM+S-TL	SVM+Adasyn	SVM+TL
Précision	59.75%	50.41%	53.42%	39.02%	66.67%
Rappel	62.50%	81.85%	76.79%	84.21%	60.53%
F-mesure	61.09%	62.31%	63.07%	53.33%	63.45%
Temps	1.66524s	5.04744s	17.08542s	4.50138	4.01805s

TABLE 5.6 – Résultats en utilisant des méthodes de ré-échantillonnage

On voit qu'en ajoutant la méthode de prétraitement Adasyn, on obtient notre meilleur rappel. C'est l'algorithme Tomek links qui permet à notre classifieur SVM d'arriver à sa meilleure précision et sa meilleure f-mesure, 63.45%, 2.36% de plus que le résultat obtenu en utilisant uniquement SVM.

Quant à l'efficacité de l'algorithme, comme ce qu'on a prévu, c'est la méthode hybride qui prend le plus de temps, puisque deux algorithmes de prétraitement sont exécutés successivement. L'exécution de la technique de sous-échantillonnage est le plus vite, car le nombre de données d'entraînement est le moins.

5.6 Comparaison entre les méthodes par ré-échantillonnage et les méthodes algorithmiques

Ici, on prend deux modèles qui nous donnent la meilleure f-mesure pour comparer leurs performances.

	Précision	Rappel	F-mesure	Temps
Smote+TL	66.67%	60.53%	63.45%	4.01805s
SVM + poids de classe 2	63.20%	67.76%	65.40%	1.61425s

TABLE 5.7 – Comparaison de la performance de modèles SVM+Smote et SVM+Poids sur le coût

Comparé avec notre méthode algorithmique, l'utilisation de TL donne une meilleure précision, 66.67%. Néanmoins, si on prend en compte de score F-mesure, c'est le modèle sensible aux coûts qui obtient les meilleurs résultats.

Le décalage du temps d'exécution entre les deux méthodes n'est pas large. Cependant, si le corpus est beaucoup plus large, la différence entre les deux va augmenter et la méthode d'ajustement du poids va économiser beaucoup plus de temps.

5.7 Affinage du modèle

Dans l'article précédent, on a vu que le modèle qui utilise l'apprentissage sensible aux coûts marche mieux sur notre corpus, puisqu'il donne la meilleure f-mesure et qu'il est plus simple à implémenter.

Afin d'améliorer la performance du modèle, la librairie GridSearchCV a encore une fois utilisée. Voici le tableau de paramètres que nous essayons d'optimiser :

Nom de paramètre	Fonction	Valeur sélectionnée par GridSearch CV
CountVectorizer : max_df	Utilisé pour supprimer les termes qui apparaissent trop fréquemment	0.5 : ignorer les termes qui apparaissent dans plus de 50% des documents
CountVectorizer : binary	Utilisé pour les modèles probabilistes discrets qui modélisent des événements binaires plutôt que des nombres entiers	True : la valeur de tf est soit 1 soit 0
TfidfTransformer : use_idf	Utilisé pour transformer une matrice de comptage en une représentation normalisée tf ou tf-idf	None : utiliser tf pour représenter les informations textuelles dans l'espace vectoriel
TfidfTransformer : norm	Utilisé pour normaliser les vecteurs	l2 : une norme l2 est la norme euclidienne, la norme la plus couramment utilisée pour mesurer la longueur d'un vecteur
SVC : C	Utilisé pour pénaliser l'erreur de classification	1
SVC() : kernel	Utilisé pour spécifier le type de noyau à utiliser dans l'algorithme	Linéaire
SVC() : classe_weight	Utilisé pour pénaliser l'erreur de classification de classe minoritaire (dans notre cas)	1 : 2

TABLE 5.8 – Paramètres à optimiser

	Précision	Rappel	F-mesure	Temps
Avant	63.20%	67.76%	65.40%	1.61425s
Après	60.20%	77.63%	67.82%	1.65359s

TABLE 5.9 – Résultats avant et après l'optimisation des paramètres

Selon le tableau au-dessus, il est clair que l'ajustement des paramètres permet d'améliorer le rappel de manière significative et aussi la f-mesure (+2.42). Notez que l'optimisation se fait en référence du score F-mesure.

5.8 Expériences avec LSTM

Le modèle LSTM, comme RNN, a aussi la forme d'une chaîne de modules du réseau neuronal récurrent. La différence se situe dans la structure du module - des connexions dans le réseau pour ajouter la capacité à mémoriser. La cellule d'état est une notion importante pour LSTM et les informations au-dedans peuvent être ajoutées ou supprimées. Cela est géré par ce qu'on appelle «une porte» dans la cellule.

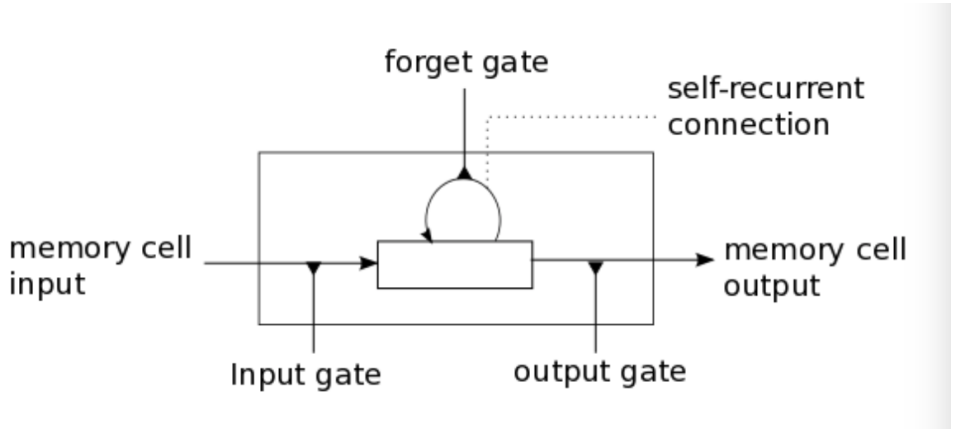


FIGURE 5.6 – Algorithme LSTM

Le processus dans le module n'est pas aussi compliqué que ce que l'image pourrait laisser penser. Le LSTM va tout d'abord décider d'éliminer une partie d'informations de la cellule d'état et la décision est faite par la couche «sigmoid» qui s'appelle «forget gate».

$$f_1 = (W_f \cdot [h_{t-1}, x_t] + b_f)$$

Ensuite, il faut choisir entre les nouvelles informations pour les sauvegarder dans la cellule. Cette étape va être réalisée par deux couches. L'une est «input gate layer» qui permet de choisir les valeurs qu'on va renouveler. L'autre est «tanh layer» visant à créer un nouveau vecteur correspondant à la valeur du nouveau candidat.

$$i_1 = (W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$i_1 = (tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$C_t = f_t * C_{t-1} + i_t * c_t$$

Enfin, il faut décider la sortie à base de l'état de la cellule récurrente. Il faut utiliser à la fois le résultat d'une couche sigmoid et faire passer l'état cellulaire à une couche tanh pour choisir la partie qu'on va prendre comme la sortie et forcer la valeur de l'état entre -1 et 1.

$$o_t = (W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = (o_t * tahn(C_t)$$

Selon la fonctionnalité de LSTM, il a une bonne performance sur l'analyse du sentiment qui implique des données séquentielles telles que des mots, des phrases. A. Hassan a fait des tests sur les modèles souvent utilisés dans la fouille d'opinions sur un corpus composé par les données IMDB [Hassan, 2017]. Le résultat montre que

pour réussir à entraîner un modèle statistique, en combinant l'algorithme word2vec non-supervisé avec LSTM, on a besoin de moins de données.

De ce fait, pour notre essai complémentaire, il devient naturellement notre premier choix

Dans nos expériences, on a utilisé l'algorithme Smote et TL pour équilibrer les données.

	Précision	Rappel	F-mesure
LSTM	33.6%	30.5%	31.98%
LSTM+Smote	15%	25.8%	18.97%
LSTM+TL	35.5%	30.5%	32.81%

TABLE 5.10 – Résultats en utilisant LSTM comme classifieur

Selon le tableau, notre modèle RNN ne fonctionne pas sur notre corpus pour détecter les signaux faibles. Le prétraitement avec TL améliore légèrement la performance, tandis que le résultat en utilisant Smote est pire.

CHAPITRE 6

DISCUSSION

Les données déséquilibrées ont une influence sévère sur la performance de l'algorithme standard de classification qui suppose que la distribution de la classe est équivalente. De nombreuses expériences ont été effectuées en utilisant différentes méthodes d'ajustement. Cependant, aucune méthode est le vainqueur absolu.

	Précision	Rappel	F-mesure	Temps
SVM	59.75%	62.50%	61.09%	1.66524s
SVM après optimisation	58.56%	67.76%	63%	1.66524s
SVM+ Smote	50.41%	81.85%	62.31%	5.04744s
SVM+S-TL	53.42%	76.79%	63.07%	17.08542s
SVM+Adasyn	39.02%	84.21%	53.33%	4.50138s
SVM+TL	66.67%	60.53%	63.45%	4.01805s
SVM+Ajustement du poids avant optimisation	63.20%	67.76%	65.40%	1.61425s
SVM+Ajustement du poids après optimisation	60.20%	77.63%	67.82%	1.65359s
LSTM	33.6%	30.5%	31.98%	—
LSTM+Smote	15%	25.8%	18.97%	—
LSTM+TL	35.5%	30.5%	32.81%	—

TABLE 6.1 – Tableau de résultats

Dans nos expériences, si la précision de notre modèle simple n'utilisant que SVM avant optimisation est plus haute par rapport à celle d'après optimisation, c'est parce que nous avons beaucoup plus d'échantillons négatifs que les positifs. Par conséquent, plus d'exemples sont classés comme des échantillons négatifs et on obtient une précision élevée avec moins de faux positifs. C'est aussi la raison pour laquelle la précision du modèle d'ajustement de poids de la classe après optimisation baisse un peu par rapport à celle d'avant optimisation. Le tableau 6.2 montre le détail de la classification :

	TN	FN	TP	FP
SVM avant optimisation	1605	57	95	64
SVM après optimisation	1597	49	103	72
SVM+Ajustement du poids avant optimisation	1609	49	103	60
SVM+Ajustement du poids après optimisation	1591	34	118	78

TABLE 6.2 – Analyse de l'erreur de la classification

On peut aussi remarquer dans nos résultats que Tomek links fonctionne pas mal sur notre corpus. Généralement, en utilisant une méthode de sous-échantillonnage, des informations importantes risquent d'être éliminées en même temps. Cependant, dans l'application réelle, il y a deux types de classification asymétrique : le déséquilibre relatif et la rareté absolue [Weiss, 2004]. Notre classe minoritaire contient 907 exemples et il est plutôt relativement rare par rapport à la classe majoritaire. C'est un cas d'un déséquilibre relatif. De ce fait, après avoir supprimé des données de la classe majoritaire et avoir des classes assez équilibrées, la performance du SVM serait aussi améliorée.

En comparant l'efficacité et l'efficience du modèle avec TL et celui en ajustant le poids de la classe minoritaire, on conclut que le dernier est plus performant sur notre corpus, puisqu'il prends moins de temps et nous permet d'obtenir notre meilleure f-mesure. En effet, il est connu qu'il y a des désavantages associés à l'utilisation des méthodes de ré-échantillonnage pour mettre en oeuvre un apprentissage sensible aux coûts. L'inconvénient du sous-échantillonnage est qu'il provoque l'élimination des données potentiellement utiles. Et l'inconvénient essentiel du sur-échantillonnage est qu'en faisant des copies exactes des exemples existants, le problème de sur-apprentissage risque d'apparaître [Weiss et al., 2007]. Un deuxième inconvénient du sur-échantillonnage est qu'il augmente le nombre d'exemples d'apprentissage, augmentant ainsi le temps d'apprentissage, ce qui est visible aussi dans notre expérience.

Néanmoins, cela ne signifie pas que la performance de la méthode en utilisant l'apprentissage sensible aux coûts est toujours meilleure que celle au niveau de données. L'expérience de G. Weiss et al. en 2007 nous montre aussi que l'algorithme d'apprentissage sensible aux coûts surpasse systématiquement les méthodes de ré-échantillonnage surtout quand nous nous concentrons exclusivement sur des ensembles de données comportant plus de 10 000 exemples et que la technique de sur-échantillonnage semble être la meilleure méthode pour les petits ensembles de données.

Les résultats avec LSTM ne sont pas arrivés à la hauteur de ce que l'on attendait. Et le prétraitement avec Smote n'a pas amélioré le résultat comme ce qu'on avait prévu. Cela peut être causé par le problème de sur-apprentissage. Nous n'avons pas obtenu de bons résultats en utilisant LSTM, cependant, on ne peux pas dire non plus qu'il n'a pas d'impact sur la classification asymétrique, parce que à cause de la manque du temps, nous n'avons pas beaucoup optimisé les paramètres qui influencent largement le résultat : taille de couche cachée, taille de minibatch, vitesses d'apprentissage, longueur de la phrase la plus longue...Bien que les RNN soient rarement utilisés pour la classification des données textuelles déséquilibrées, les expériences de Weninger et Schuller [Weninger and Schuller, 2011] ont prouvé la bonne performance de LSTM en classification asymétrique sur un corpus de cris d'animaux. Elhassan et al. [Elhassan et al., 2016] a aussi montré la force de réseau de neurones (ANN) en traitement des données textuelles déséquilibrées (Ecoli2) avec une préci-

sion 81.5% et un rappel 84.6%.

Conclusion générale

Dans ce mémoire, nous avons évalué 5 algorithmes souvent utilisés en classification asymétrique afin de trouver une bonne solution pour notre projet au sujet «Fouille d'opinions» : la détection de signaux positives de l'implication durable dans les avis des consommateurs en parfumerie. A la différence de beaucoup de travaux précédents, une analyse plus poussée sur les données et sur la demande du client est demandée dans notre travail pour que les règles d'annotation sont bien prédéfinies, puisque c'est une classification sur une notion assez abstraite. La difficulté essentielle rencontrée pendant la réalisation du projet, c'est le déséquilibre de notre corpus. L'évaluation est faite sur à la fois l'efficience et l'efficacité de l'algorithme en calculant les scores en précision, rappel et f-mesure et le temps d'exécution et l'approche en utilisant l'algorithme sensible aux coûts est la meilleure méthode avec un F-mesure de 67.82%. Notre essai avec LSTM n'est pas un succès en terme de f-mesure. Cependant, cet algorithme qui est connu pour sa bonne performance en traitement des dépendances de données séquentielles est une nouvelle piste qui mérite d'être bien évaluée sur sa performance en face de données déséquilibrées.

Les futurs travaux consistent à mettre en place d'autres méthodes qui donnent de bons résultats pour la classification des classes déséquilibrées, comme la méthode *Random over-sampling*, une méthode simple mais compétitive par rapport aux autres techniques de sur-échantillonnage plus complexes [Batista et al., 2004]. Il faut aussi concevoir une mesure d'évaluation adaptée au besoin du client qui s'attend à avoir un score de précision ou de rappel. Une évaluation sur les différents algorithmes d'apprentissage est aussi intéressante à faire. Au lien de calculer tf-itf, nous pouvons aussi essayer d'autres méthodes d'extraction de caractéristiques, par exemple, le sac de mot, afin d'avoir un meilleur résultat. Quant à l'apprentissage avec réseaux de neurones, en plus de l'optimisation des paramètres de notre modèle RNN, une évaluation de la performance de différents algorithmes (LSTM, CNN, GRU, etc.) pour la classification asymétrie mérite aussi des travaux complémentaires afin d'aider les autres à bien choisir le modèle de réseau de neurones pour classifier les données déséquilibrées textuelles.

BIBLIOGRAPHIE

[ABDOLVAND and Nikfar, 2012] ABDOLVAND, M. and Nikfar, F. (2012). Investigation of the relationship between product involvement and brand commitment. – Cité page 15.

[Aizerman, 1964] Aizerman, M. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837. – Cité page 34.

[Akbani et al., 2004] Akbani, R., Kwek, S., and Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. *Machine learning: ECML 2004*, pages 39–50. – Cité page 35.

[Baccianella et al., 2010] Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204. – Cité page 16.

[Batista et al., 2003] Batista, G. E., Bazzan, A. L., and Monard, M. C. (2003). Balancing training data for automated annotation of keywords: a case study. In *WOB*, pages 10–18. – Cité page 23.

[Batista et al., 2004] Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, 6(1):20–29. – Cité pages 23 et 55.

[Bergstra and Bengio, 2012] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305. – Cité page 38.

[Cambria et al., 2016] Cambria, E., Poria, S., Bajpai, R., and Schuller, B. W. (2016). Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *COLING*, pages 2666–2677. – Cité page 16.

[Chawla et al., 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357. – Cité page 20.

[Choubtarash et al., 2013] Choubtarash, N., Mahdieh, O., and Marnani, A. B. (2013). The study of the relationship between consumer involvement and purchase decision (case study: Cell phone). *Interdisciplinary Journal of contemporary Research in business*, 4(12):276–296. – Cité page 15.

[Demangeot and Broderick, 2007] Demangeot, C. and Broderick, A. J. (2007). Conceptualising consumer behaviour in online shopping environments. *International Journal of Retail & Distribution Management*, 35(11):878–894. – Cité page 15.

[Ding et al., 2008] Ding, X., Liu, B., and Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240. ACM. – Cité page 16.

[Dong and Dong, 2006] Dong, Z. and Dong, Q. (2006). *Hownet And The Computation Of Meaning: (With CD-ROM)*. World Scientific. – Cité page 16.

[Elhassan et al., 2016] Elhassan, T., Aljurf, M., Al-Mohanna, F., and Shoukri, M. (2016). Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method. *Journal of Informatics and Data Mining*. – Cité page 52.

[Ganganwar, 2012] Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47. – Cité page 19.

[Han et al., 2005] Han, H., Wang, W.-Y., and Mao, B.-H. (2005). Borderline-smote: a new over-sampling method in imbalanced data sets learning. *Advances in intelligent computing*, pages 878–887. – Cité page 20.

[Hassan, 2017] Hassan, A. (2017). Sentiment analysis with recurrent neural network and unsupervised neural language model. – Cité pages 16 et 49.

[He et al., 2008] He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1322–1328. IEEE. – Cité page 21.

[Hornero et al., 2008] Hornero, A. M., Corsico, A. M. H., Luzón, M. J., and Ornat, S. M. (2008). *Corpus linguistics: applications for the study of English*, volume 25. Peter Lang. – Cité page 29.

[Houston, 1978] Houston, M. J. (1978). Conceptual and methodological perspectives on involvement. *Research frontiers in marketing: Dialogues and directions*, pages 184–187. – Cité page 15.

[Hu and Liu, 2004] Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM. – Cité page 16.

[Jo and Oh, 2011] Jo, Y. and Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM. – Cité page 16.

[Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, pages 137–142. – Cité page 11.

[Kim, 2004] Kim, R. (2004). Factors influencing consumers' decision to purchase beef: a south korean case study. *Journal of International Food & Agribusiness Marketing*, 15(1-2):153–167. – Cité page 15.

[Krawczyk et al., 2014] Krawczyk, B., Woźniak, M., and Schaefer, G. (2014). Cost-sensitive decision tree ensembles for effective imbalanced classification. *Applied Soft Computing*, 14:554–562. – Cité page 17.

[Kubat et al., 1998] Kubat, M., Holte, R. C., and Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine learning*, 30(2-3):195–215. – Cité page 11.

[Laurikkala, 2001] Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution. *Artificial Intelligence in Medicine*, pages 63–66. – Cité page 22.

[Le and Zuidema, 2015] Le, P. and Zuidema, W. (2015). Compositional distributional semantics with long short term memory. *arXiv preprint arXiv:1503.02510*. – Cité page 16.

[Leenhardt and Patin,] Leenhardt, M. and Patin, G. Détecter les intentions d?achat dans les forums de discussion du domaine automobile: une approche robuste à l?épreuve des expressions linguistiques peu répandues. – Cité page 16.

[Lemaitre et al., 2017] Lemaitre, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5. – Cité page 44.

[Lewis and Catlett, 1994] Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the eleventh international conference on machine learning*, pages 148–156. – Cité page 11.

[Lewis and Ringuette, 1994] Lewis, D. D. and Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Third annual symposium on document analysis and information retrieval*, volume 33, pages 81–93. – Cité page 11.

[Li et al., 2011a] Li, J., Li, H., and Yu, J.-L. (2011a). Application of random-smote on imbalanced data mining. In *Business Intelligence and Financial Engineering (BIFE), 2011 Fourth International Conference on*, pages 130–133. IEEE. – Cité page 20.

[Li et al., 2011b] Li, S., Wang, Z., Zhou, G., and Lee, S. Y. M. (2011b). Semi-supervised learning for imbalanced sentiment classification. In *IJCAI proceedings-international joint conference on artificial intelligence*, volume 22, page 1826. – Cité page 17.

[Liu, 2012] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167. – Cité page 16.

[Michaelidou and Dibb, 2006] Michaelidou, N. and Dibb, S. (2006). Product involvement: an application in clothing. *Journal of Consumer Behaviour*, 5(5):442–453. – Cité page 15.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. – Cité page 16.

[Murphy and Aha, 1992] Murphy, P. and Aha, D. (1992). Uci repository of machine learning databases. department of information and computer science, university of california, irvine, ca. – Cité page 11.

[Ogbeide, 2014] Ogbeide, O. A. (2014). Knowing your customers to serve them better: Enduring involvement approach. *Global Research Journal of Business Management*, 2(2):5–14. – Cité page 31.

[Panchal et al., 2017] Panchal, V., Penwala, Z., Prabhu, S., Shetty, R., and Mahe, R. (2017). Sentiment analysis of product reviews and trustworthiness evaluation using trs. – Cité page 29.

[Richins and Bloch, 1986] Richins, M. L. and Bloch, P. H. (1986). After the new wears off: The temporal context of product involvement. *Journal of Consumer research*, 13(2):280–285. – Cité page 15.

[Saleiro et al., 2017] Saleiro, P., Rodrigues, E. M., Soares, C., and Oliveira, E. (2017). Feup at semeval-2017 task 5: Predicting sentiment polarity and intensity with financial word embeddings. *arXiv preprint arXiv:1704.05091*. – Cité page 16.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523. – Cité pages 16 et 35.

[Santoso et al., 2017] Santoso, B., Wijayanto, H., Notodiputro, K., and Sartono, B. (2017). Synthetic over sampling methods for handling class imbalanced problems: A review. In *IOP Conference Series: Earth and Environmental Science*, volume 58, page 012031. IOP Publishing. – Cité page 23.

[Sirgy et al., 2014] Sirgy, J., Rahtz, D., and Dias, L. (2014). Consumer behavior today. *Irvington, NY: Flatworld Knowledge Publishers*. – Cité page 31.

[Smith et al., 2014] Smith, M. R., Martinez, T., and Giraud-Carrier, C. (2014). An instance level analysis of data complexity. *Machine learning*, 95(2):225–256. – Cité page 22.

[Su et al., 2014] Su, Z., Xu, H., Zhang, D., and Xu, Y. (2014). Chinese sentiment classification using a neural network tool?word2vec. In *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*, pages 1–6. IEEE. – Cité page 16.

[Suykens, 2003] Suykens, J. A. (2003). *Advances in learning theory: methods, models, and applications*, volume 190. IOS Press. – Cité page 33.

[Tang et al., 2009] Tang, Y., Zhang, Y.-Q., Chawla, N. V., and Krasser, S. (2009). Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):281–288. – Cité pages 16 et 38.

[Ting, 2002] Ting, K. M. (2002). An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665. – Cité page 24.

[Tomek, 1976] Tomek, I. (1976). Two modifications of cnn. *IEEE Trans. Systems, Man and Cybernetics*, 6:769–772. – Cité page 22.

[Tong, 2001] Tong, R. M. (2001). An operational system for detecting and tracking opinions in on-line discussion. In *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification*, volume 1, page 6. – Cité page 16.

[Valette-Florence, 1989] Valette-Florence, P. (1989). Conceptualisation et mesure de l'implication. *Recherche et Applications en Marketing (French Edition)*, 4(1):57–78. – Cité page 15.

[Van Hulse and Khoshgoftaar, 2009] Van Hulse, J. and Khoshgoftaar, T. (2009). Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering*, 68(12):1513–1542. – Cité page 24.

[Van Rijsbergen, 1979] Van Rijsbergen, C. (1979). Information retrieval. dept. of computer science, university of glasgow. *URL: citeseer. ist. psu. edu/vanrijsbergen79information. html*, 14. – Cité page 38.

[Vapnik, 1995] Vapnik, V. N. (1995). The nature of statistical learning theory. – Cité page 33.

[Wang and Japkowicz, 2004] Wang, B. and Japkowicz, N. (2004). Imbalanced data set learning with synthetic samples. In *Proc. IRIS Machine Learning Workshop*, volume 19. – Cité page 22.

[Weiss, 2004] Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1):7–19. – Cité page 52.

[Weiss et al., 2007] Weiss, G. M., McCarthy, K., and Zabar, B. (2007). Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? *DMIN*, 7:35–41. – Cité pages 25 et 52.

[Weninger and Schuller, 2011] Weninger, F. and Schuller, B. (2011). Audio recognition in the wild: Static and dynamic classification on a real-world database of animal vocalizations. In *acoustics, speech and signal processing (ICASSP), 2011 IEEE international conference on*, pages 337–340. IEEE. – Cité page 52.

[Wiener et al., 1995] Wiener, E., Pedersen, J. O., Weigend, A. S., et al. (1995). A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th annual symposium on document analysis and information retrieval*, volume 317, page 332. Las Vegas, NV. – Cité page 11.

[Yang and Liu, 1999] Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM. – Cité page 34.

[Zhang et al., 2017] Zhang, C., Wang, G., Zhou, Y., and Jiang, J. (2017). A new approach for imbalanced data classification based on minimize loss learning. In *Data Science in Cyberspace (DSC), 2017 IEEE Second International Conference on*, pages 82–87. IEEE. – Cité page 17.

ANNEXE A

EXTRAITS DE CORPUS

Le corpus est un fichier TXT composé des avis de consommateurs extraits du site “beauté-test” :

je ne change plus de parfum ! !

J’ai toujours un flacon chez moi ! ! ! ! C’est mon parfum préféré que j’acheterai encore et encore ..... ! ! ! ! ! !

J’aime porter du parfum. Je ne sort pas sans être au préalable parfumée, j’aurais l’impression d’être nue et fade. Les parfums que j’aime me créent chacun un petit univers, accompagnent mon humeur du jour. L’"univers" d’Amour est un irrésistible cocon de douceur.

Joli parfum mais qui ne me correspond pas vraiment car j’aime les parfums qui se font sentir comme son grand-frère Coco ou Coco Mademoiselle. J’ai bientôt fini mon flacon de 50 ml mais je ne le rachèterai pas car la magie n’a pas opéré. Dommage !

Un de mes parfums prérérés, je les alterne selon on humeur . J’adore son odeur, qui tient bien, je trouve qu’il me va bien et j’ai souvent des compliments sur le sillage que mon parfum laisse, je le rachèterais sans problème

plus destiné au 20-35 ans

Je l’adore, il me remonte le moral. J’en mets dans mes cheveux pour le sentir quand je bouge. Je suis fan ! J’ai aussi pure poison et je préfère 100 fois la bouteille de celui ci, l’odeur est plus fraiche mais n’a rien à voir avec Hypnotic poison.

J’adore ce parfum, je ne le quitte plus

J’avais une promo de 50 % autrement je ne sais pas si je l’aurai acheté car il est cher pour une eau de toilette YR

l’essayer c est l’adopter

Trop sucrée et trop acide en même temp ! Un mélange vraiment éc$\frac{1}{2}$urant

Le parfum qui ne me quitte pas depuis mes 17 ans...une histoire d’amour qui dure depuis 10 ans déjà !

N’a rien à voir pour moi avec du parfum Mugler. Je n’adhère pas du tout à ce parfum

un de mes parfums préférés en tout cas celui du moment

Sa fragrance floral sucré plaira surement davantage aux plus jeunes

on se sent libre créative et unique ! ! !adapté pour les femmes qui ont de l’imagination !qui inventent leur style ! je rachèterai surtout l’élixir qui sera parfait

un parfum d hiver ! ! ideal je l adore il tiens vraiment bon et sent tres tres bon a recommander

ANNEXE B

EXTRAITS DE CODES

B.1 SVM

```
lemmatizer = WordNetLemmatizer()


with open("comment_sample.txt", 'r', encoding='utf-8') as f:
    reviews = f.readlines()
    reviews=[i.replace("\n", "") for i in reviews]
    reviews = [word for word in reviews if word not in stopwords.words('french')]
    reviews = [lemmatizer.lemmatize(wo) for wo in reviews]
with open("label_sample.txt", 'r', encoding='utf-8') as f:
    labels = f.readlines()
    labels=[m.replace("\n", "") for m in labels]
    labels = [1 if each == 'positive' else 0 for each in labels]
cn = 0
print(len(labels))
for c in labels:
    if c ==1:
        cn+=1
print(cn)
```

```
9108
907
```

```
def preprocess():
    data,target = reviews, labels
    count_vectorizer = CountVectorizer(binary=True)
    data = count_vectorizer.fit_transform(data)
    tfidf_data = TfidfTransformer(use_idf=True).fit_transform(data)

    return tfidf_data

def learn_model(data,target):

    pca = TruncatedSVD(n_components=3500)
    data = pca.fit_transform(data)
    kf = KFold(n_splits=5)
    for train_index, test_index in kf.split(data):
        data_train, data_test = data[train_index], data[test_index]
```

B.2 LSTM

```
n_words = len(vocab_to_int)
print(n_words)
graph = tf.Graph()
with graph.as_default():
    inputs_ = tf.placeholder(tf.int32, [None, None], name='inputs')
    labels_ = tf.placeholder(tf.int32, [None, None], name='labels')
    keep_prob = tf.placeholder(tf.float32, name='keep_prob')

# Size of the embedding vectors (number of units in the embedding layer)
embed_size = 300

with graph.as_default():
    embedding = tf.Variable(tf.random_uniform((n_words, embed_size), -1, 1))
    embed = tf.nn.embedding_lookup(embedding, inputs_)



with graph.as_default():
    # basic LSTM cell
    lstm = tf.contrib.rnn.BasicLSTMCell(lstm_size)
    # Add dropout to the cell
    drop = tf.contrib.rnn.DropoutWrapper(lstm, output_keep_prob=keep_prob)
    # Stack up multiple LSTM layers, for deep learning
    cell = tf.contrib.rnn.MultiRNNCell([drop] * lstm_layers)
    cell = tf.contrib.rnn.MultiRNNCell([drop] * lstm_layers)
    # Getting an initial state of all zeros
    initial_state = cell.zero_state(batch_size, tf.float32)

with graph.as_default():
    outputs, final_state = tf.nn.dynamic_rnn(cell, embed,
                                             initial_state=initial_state)
    print(outputs,final_state)

with graph.as_default():
    predictions = tf.contrib.layers.fully_connected(outputs[:, -1], 1, activatio
    cost = tf.losses.mean_squared_error(labels_, predictions)
    optimizer = tf.train.AdamOptimizer(learning_rate).minimize(cost)
```