MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABBES LAGHROUR KHENCHELA

FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département de Mathématiques et informatique

Mémoire de fin d'étude pour l'obtenir de diplôme

De master informatique

Spécialité :Génie logicielle système distribuée

Thème

Mesure de la satisfaction de clients basée sur les Commentaires en ligne

Présenté Par :

Himeur Younes

HadjadjDhiaaElhak

Dirigé Par :

Dr Rahab Hichem

2021/2020

ملخص

هل تساءلت يومًا عن كيفية استخدام كتلة المعلومات المتوفرة لدينا اليوم. مثال على ذلك هو موجز الأخبار عبر الإنترنت أو مراجعات المتجر الإلكترونية أو وسائل التواصل الاجتماعي. تكمن المشكلة في أن هذه المعلومات غالبًا ما تكون وفيرة جدًا لدرجة أنها أصبحت تفيض وتكاد تكون غير صالحة للاستعمال. مع وجود أكثر من 6000 تغريدة على تويتر في الثانية، ربما نضيع الكثير من المعلومات ونضيع الميزة التي يملكها أولئك الذين يمكنهم استخدامها. من خلال الجمع بين التنقيب في النص ومعالجة اللغة الطبيعية والتعلم الآلي، يصبح من الممكن إنشاء تطبيق قادر على تحليل المشاعرنص بسرعة الضوء. التقاط مثل هذا الشعور يسمى تحليل المشاعر، وهذا ما نحاول تحقيقه في هذا المشروع. الهدف من هذا المشروع هو متابعة جميع الخطوات اللازمة لتطوير نموذج دكاء اصطناعي القادر على تحليل وتصنيف أي مشاعر للعملاء من خلال مراجعة النص مراجعتهم لمنتج، من الناحية النظرية والعملية باستخدام Python. سنرى كيفية تنقية وتحويل هذه البيانات النصية، بحيث يمكن استخدامها بكفاءة بواسطة النموذج. سنقوم أخيرًا ببناء نموذجنا واختباره من خلال تقييمات العملاء الحقيقية. فكر الآن فيما يمكننا تعلمه إذا استطعنا استخراج كل هذه المعلومات المفيدة من هذا النص في فترة زمنية معقولة، وهو ما سنحاول القيام به في هذا المشروع.

Abstract

Have you ever wondered how to use the mass of information we have today? An example is the online news feed, online store reviews or social media. The problem is that this information is often so abundant that it becomes overflowing and virtually unusable. With over 6,000 tweets per second, we're probably missing a ton of information and losing the edge over those who can use it. By combining text mining, natural language processing and machine learning, it becomes possible to create an application capable of analyzing the feeling of a text at the speed of light. Capturing such a feeling is called sentiment analysis, and that is what we are trying to achieve in this project. The objective of this project is to go through all the steps necessary to develop a machine learning model capable of analyzing and classifying any customer sentiment from a text review, in theory and in practice with Python. We will see how to clean and transform this textual data, so that it can be used efficiently by a model. We will finally build our model and test it with real customer reviews. Now think what we can learn if we can extract all of this useful information from this text in a reasonable amount of time, which is what we are going to try to do in this project.

Résumé

Vous êtes-vous déjà demandé comment utiliser la masse d'informations dont nous disposons aujourd'hui. Un exemple le fil d'actualité en ligne, les critiques de boutiques en ligne ou les médias sociaux. Le problème est que cette information est souvent si abondante qu'elle en devient débordante et pratiquement inutilisable. Avec plus de 6000 tweeds par seconde, nous manquons probablement une tonne d'informations et perdons l'avantage sur ceux qui peuvent l'utiliser. En combinant fouille de textes, le traitement du langage naturel et l'apprentissage automatique, il devient possible de créer une application capable d'analyser le sentiment d'un texte à la vitesse de la lumière. Capturer un tel sentiment est ce qu'on appelle l'analyse des sentiments, et c'est ce que nous cherchons à réaliser dans ce projet. L'objectif de ce projet est de parcourir toutes les étapes nécessaires au développement d'un modèle de machine Learning capable d'analyser et de classer tout sentiment client à partir d'une revue de texte, en théorie comme en pratique avec Python. Nous verrons comment nettoyer et transformer ces données textuelles, afin qu'elles puissent être utilisées efficacement par un modèle. Nous allons enfin construire notre modèle et le tester avec de vrais avis clients. Maintenant pensez à ce que nous pouvons apprendre si nous pouvons extraire toutes ces informations utiles de ce texte dans un délai raisonnable c'est ce que nous allons essayer de faire dans ce projet.

Remerciement

Nous remercions DIEU le tout puissant qui m'a donné la force, la volonté et le courage pour accomplir ce modeste travail. Je tiens à formuler ma gratitude et ma profonde reconnaissance.

Nous tenons à remercier très chaleureusement Dr Rahab Hichem, notre encadreur de mémoire pour ses conseils, ses encouragements et sa confiance. Nous le remercie aussi pour sa patience, sa bienveillance durant toute cette année de mémoire et son soutien.

Nous remercions les honorables membres de jury d'avoir accepté d'être membre de notre jury de mémoire, d'évaluer nos travaux et pour nous avoir honorés de leur présence.

Nous adressons également nos remerciements à nos amis et nos collègues et nos professeurs de l’université Abbes Laghrour qui nous ont soutenus pendant toutes ces années.

Enfin, c'est l'occasion pour nous d'adresser nos remerciements à nos parents pour le soutien inconditionnel qui nous a apporté au cours d'année de thèse ainsi que tout au long de nos études supérieures. Leur appui nous a été très précieux et nous leur en témoigne aujourd’hui notre plus grande reconnaissance. Nous remercions nos frères pour leur soutien moral.

Table des matières

Tables des figures

Table des tableaux

Introduction générale

Comme vous le savez, les données sont partout. Mais d'abord, permettez-nous de poser une question. Selon vous, combien de données ont été créées au cours des deux dernières années ?

Remontons le temps et mettons les choses en perspective.

L’écriture est apparue il y a un peu plus de 30 000 ans, l'impression elle-même est venue bien plus tard. Et il n'a qu'un peu moins de 600 ans. Enfin, le web est encore plus jeune et n'est apparu qu'il y a un peu plus de 30 ans.

A l'exclusion de la période précédant l'écriture. Les êtres humains créent et partagent encore des informations écrites depuis 32 000 ans. Alors permettez-moi de poser à nouveau ma question. Parmi toutes les données que nous avons créées au cours des 32000 dernières années, quel pourcentage pensez-vous avoir été créé au cours des deux dernières années ?

La réponse est 90 %, et c'est un chiffre assez conservateur. Comment est-ce possible ?

Eh bien, les données sont partout dans le monde d'aujourd'hui, elles sont créées à chaque instant par vous et par moi, et cela signifie que ce sont autant de données qu'il est possible d'exploiter.

Maintenant, pourriez-vous dire, qu'est-ce que les données ont à voir avec le texte ?

Eh bien, sous toutes ses formes, le texte représente une grande partie des données disponibles, environ 80 %, pour ainsi dire. Si vous pensez que c'est beaucoup, pensez à Google et ses 3 milliards de recherches par jour en pensant à Facebook et ses 350 millions de postes par jour, pensez à Twitter et ses 6000 tweets par seconde. Pensez à Wikipédia et à ses 29 milliards de mots, qui ne représentent finalement que les 4,6 milliards de personnes qui utilisent actuellement Internet et partagent des données textuelles.

Encore une fois, c'est vous et moi qui partageons sur les réseaux sociaux, ce sont aussi les entreprises qui partagent leurs rapports financiers, les médias qui partagent les dernières nouvelles, les éditeurs qui partagent des livres ou les universités qui publient leurs cours en ligne.

Sachez que nous savons que le texte représente une grande partie des données disponibles aujourd'hui.

Voyons comment nous pouvons en profiter. Tout d'abord, examinons un exemple que vous devriez connaître. Dites que vous êtes prêt à acheter un nouveau smartphone sur Amazon. Je ne sais pas pour vous, mais je suis du genre à comparer pas mal les différentes offres disponibles. Ce que je fais habituellement, c'est comparer les produits en fonction de leur avis et plus précisément à deux niveaux. Le premier est simplement le nombre d'étoiles entre un et cinq, ce qui est très facile à comprendre et à interpréter. Mais il y a autre chose qui est tout aussi précieux.

La critique de texte laissée à la lecture, cette critique donne beaucoup plus d'informations sur le produit que la note seule, elle explique pourquoi une telle note a été donnée. Il existe néanmoins une différence entre ces deux données. Alors que la notation est facilement compréhensible par ordinateur car il ne s'agit que d'un nombre, il lui est beaucoup plus difficile pour un ordinateur de comprendre et d'interpréter le texte.

1.1 Fouille de textes et NLP

Les informations contenues dans cette revue Amazon peuvent être classées en deux catégories de données structurées et non structurées.

Les données structurées sont toutes les données qui suivent un format spécifique et délimité, comme dans ce cas, la note est une information telle que l'endroit où la critique a été faite. Les données, qui ont créé la couleur du téléphone, sa capacité de stockage, ou ces informations peuvent être stockées et recherchées très facilement sur tous les smartphones disponibles sur le site Web qui définit.

Les données non structurées La revue de texte appartient à la deuxième catégorie appelée non structurées des informations pertinentes sont rassemblées ici et là sur les mots, ce qui rend la compréhension et l'utilisation beaucoup plus difficiles pour un ordinateur dans ce cas.

Des mots comme ‘’ sont glissants’’, ‘’mais’’,’’ euh’’, sont très informatifs sur la qualité. Malgré tout, de telles informations doivent être détectées dans le texte et elles ne sont pas immédiatement disponibles ou facilement consultables.

L'exploration de text mining et le Traitement du langage naturel sont là pour nous aider à récupérer et à extraire des informations significatives à partir de texte non structuré.

Il y a plusieurs raisons pour lesquelles vous pourriez vouloir faire cela, mais nous n'allons examiner que l'une des plus populaires, qui s'appelle un LP ou un traitement du langage naturel (**NLP**).

Il est utilisé dans un tas d'applications telles que des assistants virtuels tels que Siri, Alexa ou un assistant Google, quelle que soit la boîte de discussion que vous trouvez sur à peu près n'importe quel site Web, ou les fonctionnalités de Google Translate. Tous ces exemples sont de pures applications NLP car ils s'efforcent de comprendre toutes les langues.

> Pourquoi les machines ne peuvent-elles pas simplement comprendre le texte comme nous, les humains ?
> Pourquoi ont-ils besoin de tout un domaine de recherche ?

La langue que nous parlons est évidente pour nous, elle contient beaucoup de complexités, d'ambiguïtés, et elle est assez diversifiée. C'est quelque chose qui doit être transmis à l'ordinateur d'une manière ou d'une autre.

Le text mining et le NLP sont utilisées ensemble pour que l'ordinateur peut comprendre le langage humain.

1.2 Analyse de sentiments

Nous allons maintenant voir une autre application très spécifique de l'exploration de texte (text mining) et de L.P **l'analyse des sentiments**.

Sentiment analysais est utilisé pour générer des sentiments Frontex à l'aide de l'exploration de texte

Et puis ce seront des techniques. Prenons un exemple très précis. Disons que nous sommes des commerçants surveillant l'ancien marché. Nous devons trier les données pour prendre nos décisions. Nous avons des données plus strictes comme les stocks de pétrole ou les indicateurs de croissance économique qui font monter et descendre les prix du pétrole. Mais nous avons aussi beaucoup de données non structurées disponibles.

Supposons que nous ayons un fil d'actualités qui se met à jour toute la journée, ce fil d'actualités est une mine d'informations, à condition que nous puissions l'utiliser efficacement. Bien sûr, nous pourrions lire toutes les nouvelles une par une, mais nous passerions probablement toute la journée à le faire car elles sont constamment mises à jour.

Nous avons donc besoin d'une autre solution en tant que commerçant, notre objectif est d'extraire des sentiments de ces nouvelles, positives ou négatives, afin que nous puissions prendre une décision mieux informée le plus rapidement possible.

L'analyse des sentiments vise à détecter les sentiments dans le texte et s'applique à des domaines tels que les services à la clientèle, la reconnaissance de la marque ou même le commerce.

Chapitre 1 : Fouille d’opinion et analyse de sentiments

1. Introduction

L'expression sur les grandes annonces de nouvelles peut avoir un impact important sur le marché financier et le comportement des investisseurs entraînant des changements rapides Dans le monde actuel, la disponibilité et l'augmentation exponentielle de l'utilisation d'Internet ont conduit les individus à se référer pour communiquer et partager des données sur divers sujets allant des produits finaux à divers services compris les services de santé. L'idée de base derrière ces nouvelles technologies d'analyse émergentes est de comprendre, de prédire le comportement et l'attitude humains et de fournir des informations aux commerçants qui pourraient être utilisées pour prévoir et organiser le processus commercial avant de prendre des décisions d'investissement ou de gestion des risques.

Les destinations de commerce électronique sont la capitale du marché. En raison de la fiabilité des sites, personne n'a besoin de sortir de l'industrie, où ces sites sont plus fiables au point d'être mis en valeur.

Les clients se tournent vers les sites de commerce électronique alors qu'ils sont à l'affût pour que les produits aillent de l'avant. À ce stade, un grand nombre de sites Web d'organisations établies et rumeurs propulsent leurs produits, maintenir la confiance dans ces sites est l'objectif ultime, et l'analyse des sentiments est la principale préoccupation dans cette région. Le terme sentiment signifiant une vue ou une opinion exprimée et analyse signifiant la structure de quelque chose, donc mettre ces deux mots dans un sens aide à découvrir ces sentiments.

L'analyse des sentiments est une forme de traitement du langage naturel (NLP) qui permet de suivre l'humeur et l'attitude du public concernant tout élément ou sujet. L'analyse des sentiments ou l'exploration d'opinions, qui comprend la construction d'un schéma ou d'un modèle d'identification et d'étude de données visant à obtenir et à examiner les sentiments exprimés par les personnes p positivement ou négativement par l'analyse d'une grande quantité de données issues d'enquêtes,

Réactions et avis. C'est un domaine d'étude qui pourrait être utile de plusieurs façons. Par exemple, en marketing, il aide à fournir de meilleures analyses de produits ou même de surveiller les études de marché qui peuvent déterminer quelle version d'un produit ou d'un service est problématique ou populaire.

2. Analyse des sentiments en e-commerce

Aujourd'hui, le développement rapide d'Internet et de ses utilisateurs a modifié la façon dont les individus communiquent dans le monde entier, en particulier lorsqu'ils font des affaires, les applications Internet sur les opérations commerciales ont développé de nouvelles possibilités pour la façon dont les produits ou services sont vendus dans le monde aujourd'hui. L'accessibilité des plateformes de médias sociaux a permis aux internautes d'exprimer et de partager leurs opinions sur différents types de composants en fonction de leur expérience de vie, y compris les produits et services qu'ils apprécient.

L'analyse des sentiments est une technologie en plein essor qui exploite les demandes des clients sur la base du traitement du langage naturel. Cette motivation est généralement utilisée pour bien comprendre ce que veulent les clients, quand, pourquoi et comment ils le souhaitent, les détaillants doivent se tourner vers l'analyse des sentiments, évitant ainsi de faire les mêmes erreurs et de choisir les bonnes décisions en fonction des commentaires ou des critiques. Dans le cadre du commerce électronique, les achats en ligne sont un bon exemple de la façon dont les produits ou services sont vendus sur Internet. Les grands distributeurs comme Amazon et Alibaba ainsi que les petits distributeurs ont certainement eu des résultats décevants, l'un des principaux facteurs de la lenteur de leurs ventes était un assortiment de produits médiocre. Ces distributeurs étaient essentiellement incapables de mettre les bons produits en rayon, et les clients les punissaient en dépensant leur argent ailleurs.

La compréhension des consommateurs a toujours été une priorité la liste des distributeurs et l'utilisation de l'analyse des sentiments pour surveiller ces émotions ont été le principal motif pour les entreprises de comprendre à quel point l'exploration d'opinions sur les avis des clients peut être diversifiée et approfondie. Internet est un champ de mines de perspective, pouvoir accéder à ces opinions sur une variété de différentes plateformes est un avantage significatif pour toute entreprise cherchant à améliorer ses produits ou services.

3. Les techniques d'analyse des sentiments

Il existe de nombreuses applications et améliorations des algorithmes d'analyse des sentiments qui ont été proposées et utilisées depuis plusieurs années à ce jour. Ce projet vise à examiner de plus près les techniques les plus couramment utilisées dans le commerce de détail, en particulier dans le secteur du commerce électronique. Dans la classification des sentiments, il

existe deux domaines d'étude principaux comme Machine Learning et Lexicon, et chaque domaine a sa propre subdivision, comme le montre la Fig.1. Il y a également eu peu d'études combinant ces deux techniques et gagnant comparativement une meilleure efficacité dans l'opération d'analyse des sentiments.

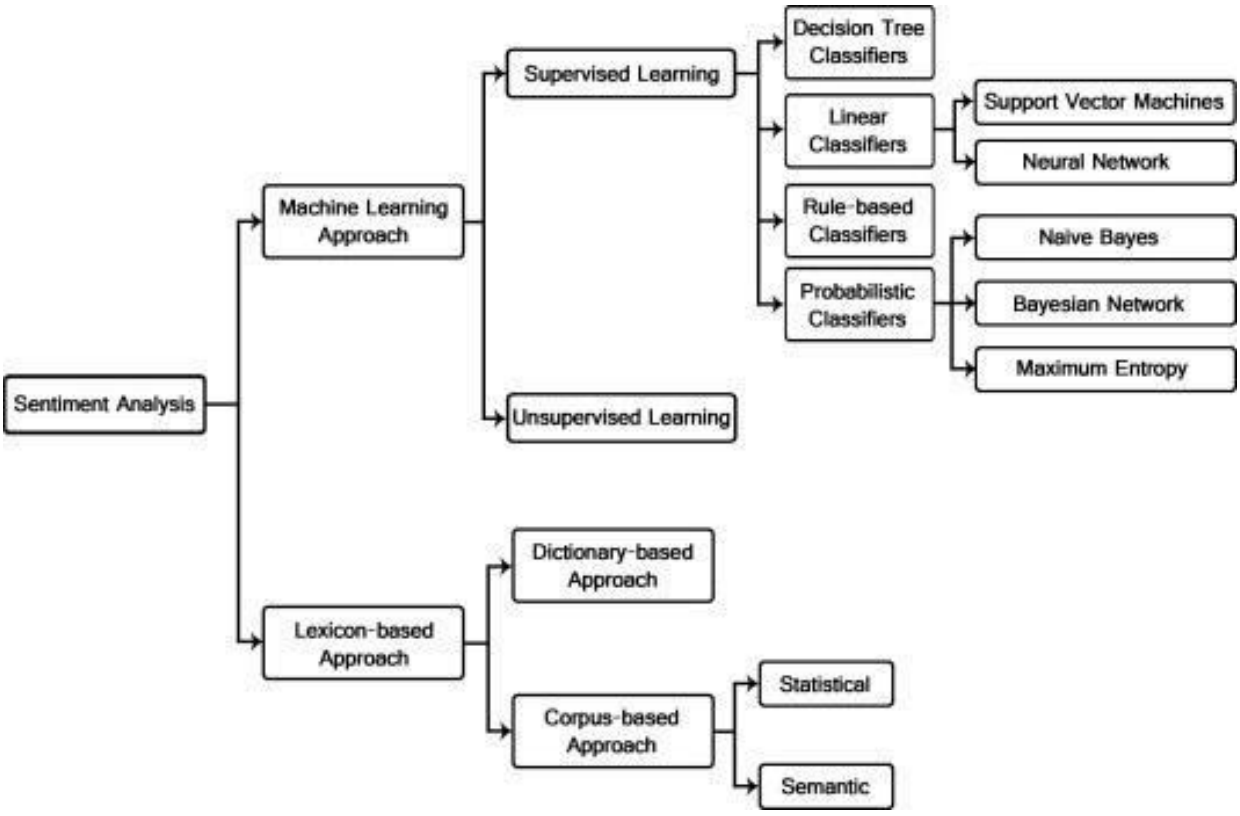


Figure. 1.Techniques de classification des sentiments

4. Approche d'apprentissage automatique

Diverses techniques peuvent être utilisées pour analyser les sentiments, parmi lesquelles l'une des techniques couramment utilisées dans le commerce de détail est la méthode Naïve Bayes. Il s'agit d'un algorithme d'apprentissage probabiliste dérivé du principe du choix bayésien Le classificateur Naive Bayes (NBC) fusionnerait une nouvelle compréhension avec une compréhension antérieure. Ces algorithmes de classification sont simples et ont des effets similaires à d'autres techniques. Dans le NBC, la probabilité d'un message x de classe c, est calculée à l'aide de l'équation suivante [7].

Formule :

$$P(c|x) = \frac{P(x|c)\,P(c)}{P(x)}$$

• P(c) - le prédicteur (attribut) de vraisemblance fournie par la classe (cible).
• P(c) - probabilité de la classe précédente.
• P(x) - la probabilité de la classe fournie par le prédicteur.
• P(x) - probabilité précédente du prédicteur.

La méthode dans laquelle un avis peut être classé comme positif (pouce levé) ou négatif (pouce vers le bas) qui est extrait des phrases et utilisé pour la classification des sentiments. L'algorithme a une efficacité dans la classification de texte avec une précision de 83%. Le modèle naïf de Bayes pour les très grands ensembles d'informations est simple à construire et particulièrement utile. Naïve Bayes est connu pour effectuer des techniques de classification même extrêmement avancées en plus de la simplicité

4.1 Approche basée sur le lexique

La méthode basée sur le lexique utilise des phrases de mots prédéfinies et les idiomes d'opinion où chaque phrase et chaque idiome est évalué comme un sentiment positif ou un sentiment négatif. La plupart des chercheurs ont utilisé des approches automatiques telles que des dictionnaires et des corpus pour attribuer les mots d'opinion, mais ils attribuent toujours manuellement les mots et les phrases dans les déclarations de point de vue pour garantir la bonne attribution des mots et des phrases.

Cette règle donne des estimations obliques comparatives aux mots sémantiquement proches, dans le cas des secteurs de la vente au détail comme les sites de commerce électronique. Il y a beaucoup de commentaires et de critiques révélateurs qui incluent de l'argot et des fautes d'orthographe en raison de différentes langues.

Cette situation se traduit par un travail difficile pour la conception et le développement de systèmes automatiques. De plus, pour évaluer le ressenti de la remarque, une compréhension préalable est nécessaire pour classer la polarité de l'opinion. Deux approches pourraient être utilisées sous le classificateur de lexique, telles que la méthode basée sur le dictionnaire et la méthode basée sur le corpus, pour collecter des

dictionnaires en ligne avec un certain nombre de déclarations d'opinion pour les synonymes et antonymes respectifs. De nouvelles phrases sont ajoutées à la liste des graines et la technique continue d'ajouter les phrases de manière itérative jusqu'à ce qu'aucune nouvelle phrase ne soit trouvée. Il a été souligné d'utiliser des vérifications manuelles pour nettoyer la liste enfin.

5. Analyse des sentiments : nouvelles opportunités

La montée des médias sociaux contre les détaillants fait face à une atmosphère dynamique et concurrentielle, les détaillants recherchant une plus grande mondialisation et une plus grande compétitivité sur les blogs et les plateformes de médias sociaux qui ont alimenté l'intérêt pour l'évaluation des sentiments.

C'est une technologie puissante avec un excellent potentiel pour aider les organisations à se concentrer sur des informations importantes dans leurs données qui nécessitent un mécanisme correct pour les transformer en compréhension. Plusieurs études montrent que le pourcentage de particuliers et d'entreprises utilisant des applications de plateforme de médias sociaux comme outil de gestion de la relation client (CRM) s'est considérablement amplifié. De nombreuses critiques et éloges sont normalement publiées et signalées quelques minutes seulement après la sortie d'un nouveau produit. L'analyse de ces informations permet aux entreprises de s'adapter à cette tendance croissante en réalisant certaines valeurs commerciales telles que l'augmentation de la clientèle, la fiabilité des clients (fidélité), la satisfaction des clients (plaisir/bonheur) et la réputation des clients tout en réalisant des revenus et des bénéfices plus élevés [10]. D'autre part, en analysant les forces et les faiblesses des caractéristiques divergentes des articles, ainsi qu'en découvrant les taux de satisfaction des autres consommateurs de différents produits, les entreprises peuvent également utiliser ces informations comme témoignages. Les clients aiment être entendus, en utilisant l'analyse des sentiments pour définir cette vue ou la classer pourrait avoir beaucoup d'effets. Si nous regardez les sites de commerce électronique où les meilleures critiques ou remarques sur tous les types de produits et services, la construction d'un système pour examiner ces critiques pourrait aider à faire de meilleurs choix et surveiller les marques, améliorer le support client, garder un œil sur leur concurrence, sur ou même mieux gérer une crise si elle est susceptible de se produire. Il n'y a rien de pire que de trouver une catastrophe ou un problème avant qu'il ne soit trop tard. Eh bien, l'analyse des

sentiments a aidé à découvrir tous ces problèmes, le stress et la perte d'argent ou de ressources.

6. Analyse des sentiments : les défis

Dans l'évaluation d'informations brutes telles que la désambiguïsation du sens des mots, les négociations, la comparaison, l'intensité et le sarcasme, les modèles d'analyse des sentiments sont confrontés à plusieurs difficultés en raison de la nature du langage humain. Premièrement, la désambiguïsation du sens des mots est un modèle qui reconnaît un mot comme positif, cependant, il peut refléter un sens négatif dans d'autres cas. Par exemple, "Un petit nombre d'étudiants est bénéfique pour les étudiants qui préfèrent étudier dans un petit environnement", mais si le lecteur souhaite une boîte plus grande, la taille de la petite boîte peut être négative. Le modèle doit donc être dans un domaine particulier afin d'éviter toute ambiguïté.

Les comparaisons peuvent également provoquer des confusions, en dehors de la désambiguïsation. Regardez Fossile, par exemple, c'est mieux que de regarder le diesel. Alors qu'une bonne opinion signifie le mot "meilleure", si le document traite de la montre Diesel, cela doit être considéré comme une polarité négative. L'intensité des mots peut également exagérer les vues qui peuvent conduire à l'attribution d'un document dans une classe incorrecte. La négation peut faire une erreur car elle peut modifier la polarité d'une phrase de positive à négative. Par exemple, "il y a une bonne possibilité que d'autres religions puissent dominer un pays". Cette phrase implique plus de variété qui est généralement considérée comme positive, mais elle peut être négative parce que l'auteur est préoccupé par une bonne opportunité. Le sarcasme est difficile car il nécessite une étude approfondie car le sarcasme utilise des mots positifs, mais l'auteur implique un contexte défavorable. Dans un autre scénario, un terme d'opinion jugé positif peut être considéré comme négatif car les individus n'expriment pas leurs opinions de la même manière. Certaines critiques peuvent être simples à comprendre pour les humains, mais difficiles à corriger en raison du contexte pour les ordinateurs.

L'analyse des sentiments n'est pas encore couramment utilisée car elle dépend d'énormes quantités d'informations et d'un expert pour opérer dessus. Pour développer un modèle d'analyse des sentiments, une entreprise a besoin d'informations à jour pour construire un modèle. Sinon, il est possible d'utiliser des sources Internet telles que WordNet ou des ensembles de données publics, des blogs et des plateformes de médias sociaux. Dans ce scénario, la plupart des ensembles de données publics ne sont pas spécifiques au contexte et les données doivent être sélectionnées pour répondre aux objectifs de l'entreprise. De plus, en raison de l'absence de financement, certaines langues ont un ensemble de données restreint, il est devenu difficile pour les entreprises de créer des modèles spécifiques aux langues. Si les entreprises veulent mettre en œuvre un tel système pour chaque utilisateur, il faudra dépenser beaucoup d'argent en raison de méthodes qui nécessitent beaucoup de puissance de calcul. Certains modèles peuvent être difficiles à mettre en œuvre dans certains cas car ils sont trop compliqués ou nécessitent beaucoup d'efforts pour les optimiser.

Les problèmes de confidentialité entravent également l'avancement de l'analyse des sentiments. Les entreprises et les chercheurs doivent évaluer la quantité d'informations qu'ils prennent en tant qu'individus, en particulier des pays occidentaux, qui valorisent la confidentialité sans envahir la vie quotidienne des consommateurs. Sinon, les consommateurs peuvent intenter une action en justice contre les entreprises pour violation de leur vie privée, et des réglementations sur la confidentialité des données doivent être promulguées par le gouvernement pour garantir que les entreprises sont transparentes quant à l'utilisation des données de leurs clients. En effet, l'une des principales raisons pour lesquelles la Chine peut rivaliser avec les États-Unis à l'avenir est que les individus chinois ne sont pas aussi préoccupés par la confidentialité des données que les individus occidentaux.

7. Conclusion

L'analyse des sentiments est un domaine important basé sur un calcul rapide, un grand volume de données et d'informations, des modèles mathématiques complexes basés sur l'apprentissage automatique et des statistiques pour comparer les avis des clients de sites Web de commerce électronique distincts. Il y a une amélioration significative des

résultats de nos jours lorsque vous utilisez des plateformes de médias sociaux telles que Twitter, Facebook et Instagram pour la collecte de données au lieu de données provenant du site Web le plus positif. Divers algorithmes d'apprentissage automatique robustes sont utilisés pour prédire le sentiment qui est généralement considéré comme le principal facteur d'influence pour les clients potentiels et potentiels afin de prendre des décisions d'achat efficaces. Cela peut offrir une meilleure expérience utilisateur et aider les entreprises à prendre des décisions ou à développer un modèle qui améliorera également les relations avec les clients. Les entreprises peuvent évaluer l'ampleur de l'acceptation du produit à l'aide d'une analyse des sentiments et peuvent développer des politiques pour améliorer leur produit. Les particuliers peuvent également utiliser des instruments d'exploration d'opinion pour créer des choix d'achat en comparant des produits concurrents.

Chapitre 2 : apprentissage automatique

1. Introduction

L'apprentissage automatique est un sous-domaine de l'intelligence artificielle (IA). L'objectif de l'apprentissage automatique est généralement de comprendre la structure des données et d'intégrer ces données dans des modèles qui peuvent être compris et utilisés par les gens.

Bien que l'apprentissage automatique soit un domaine de l'informatique, il diffère des approches informatiques traditionnelles. En informatique traditionnelle, les algorithmes sont des ensembles d'instructions explicitement programmées utilisées par les ordinateurs pour calculer ou résoudre des problèmes. Les algorithmes d'apprentissage automatique permettent plutôt aux ordinateurs de s'entraîner sur les entrées de données et d'utiliser une analyse statistique afin de générer des valeurs comprises dans une plage spécifique. Pour cette raison, l'apprentissage automatique permet aux ordinateurs de créer des modèles à partir d'exemples de données afin d'automatiser les processus de prise de décision basés sur les entrées de données.

En apprentissage automatique, les tâches sont généralement classées en grandes catégories. Ces catégories sont basées sur la manière dont l'apprentissage est reçu ou sur la manière dont la rétroaction sur l'apprentissage est transmise au système développé.

Deux des méthodes d'apprentissage automatique les plus largement adoptées sont l'apprentissage supervisé qui entraîne des algorithmes basés sur des exemples de données d'entrée et de sortie étiquetées par des humains, et l'apprentissage non supervisé qui fournit à l'algorithme aucune donnée étiquetée afin de lui permettre de trouver une structure dans son entrée. Les données. Explorons ces méthodes plus en détails.

1.1 Apprentissage supervisé

Dans l'apprentissage supervisé, l'ordinateur est fourni avec des exemples d'entrées qui sont étiquetés avec les sorties souhaitées. Le but de cette méthode est que l'algorithme puisse ‘’ apprendre ‘’ en comparant sa sortie réelle avec les sorties ‘’ enseignées ‘’ pour trouver des erreurs et modifier le modèle en conséquence. L'apprentissage supervisé utilise donc des modèles pour prédire les valeurs d'étiquette sur des données supplémentaires non étiquetées.

1.2 Apprentissage non supervisé

Dans l'apprentissage non supervisé, les données ne sont pas étiquetées, de sorte que l'algorithme d'apprentissage doit trouver des points communs parmi ses données d'entrée. Les données non étiquetées étant plus abondantes que les données étiquetées, les méthodes d'apprentissage automatique qui facilitent l'apprentissage non supervisé sont particulièrement précieuses.

L'objectif de l'apprentissage non supervisé peut être aussi simple que de découvrir des modèles cachés dans un ensemble de données, mais il peut également avoir un objectif d'apprentissage de caractéristiques, qui permet à la machine informatique de découvrir automatiquement les représentations nécessaires pour classer les données brutes.

2. Préparation des données pour l'apprentissage automatique

La préparation des données (également appelée « prétraitement des données ») est le processus de transformation des données brutes afin que les scientifiques et les analystes des données puissent les exécuter via des algorithmes d'apprentissage automatique pour découvrir des informations ou faire des prédictions.

Le processus de préparation des données peut être compliqué par des problèmes tels que :

1- Entrée manquante ou incomplète. Il est difficile d'obtenir chaque point de données pour chaque enregistrement d'un ensemble de données. Les données manquantes apparaissent parfois sous forme de cellules vides, de valeurs (par exemple, NULL ou N/A) ou d'un caractère particulier, tel qu'un point d'interrogation.

2-Valeurs aberrantes ou anomalies. Des valeurs inattendues apparaissent souvent dans une distribution de valeurs, en particulier lorsque vous travaillez avec des données provenant de sources inconnues qui manquent de mauvais contrôles de validation des données

3- Données mal formatées/structurées. Les données doivent parfois être extraites dans un format ou un emplacement différent. Un bon moyen de résoudre ce problème est de consulter des experts du domaine ou de joindre des données provenant d'autres sources

4- Valeurs incohérentes et variables catégorielles non standardisées. Souvent, lors de la combinaison de données provenant de plusieurs sources, nous pouvons nous retrouver avec

des variations de variables telles que les noms d'entreprise ou les noms de pays. Par exemple, un pays dans un système pourrait être « Algérie », tandis que dans un autre, il pourrait être « DZ ». Trouver toutes les variations et standardiser correctement améliorera considérablement la précision du modèle.

5-Caractéristiques/attributs limités ou épars. L'enrichissement des fonctionnalités, ou le développement des fonctionnalités de nos données, nous oblige souvent à combiner des ensembles de données provenant de diverses sources. La jointure de fichiers de différents systèmes est souvent entravée lorsqu'il n'y a pas de colonnes faciles ou exactes pour faire correspondre les ensembles de données. Cela nécessite ensuite la capacité d'effectuer une correspondance approximative, qui pourrait également être basée sur la combinaison de plusieurs colonnes pour obtenir la correspondance. Par exemple, combiner deux ensembles de données sur l'ID CLIENT (présent dans les deux ensembles de données) pourrait être facile. La combinaison d'un ensemble de données qui a des colonnes séparées pour CUSTOMER FIRST NAME et CUSTOMER LAST NAME avec un autre ensemble de données avec une colonne CUSTOMER FULL NAME, contenant « Last Name, First Name » devient plus délicate.

6- Le besoin de techniques telles que l'ingénierie des fonctionnalités. Même si toutes les données pertinentes sont disponibles, le processus de préparation des données peut nécessiter des techniques telles que l'ingénierie des fonctionnalités pour générer un contenu supplémentaire qui se traduira par des modèles plus précis et pertinents.

Importance de la préparation des données

La plupart des algorithmes d'apprentissage automatique nécessitent que les données soient formatées d'une manière très spécifique, de sorte que les ensembles de données nécessitent généralement une certaine préparation avant de pouvoir fournir des informations utiles. Certains ensembles de données ont des valeurs manquantes, invalides ou difficiles à traiter pour un algorithme. Si des données sont manquantes, l'algorithme ne peut pas les utiliser. Si les données ne sont pas valides, l'algorithme produit des résultats moins précis ou même trompeurs. Certains ensembles de données sont relativement propres mais doivent être mis en forme (par exemple, agrégés ou pivotés) et de nombreux ensembles de données manquent simplement de contexte commercial utile (par exemple, des valeurs d'ID mal définies), d'où la nécessité d'enrichir les fonctionnalités. Une bonne préparation des données produit des

données propres et bien organisées qui conduisent à des résultats de modèle plus pratiques et plus précis

3. Annotation de corpus

Il semble que chaque jour, il y ait de nouveaux problèmes passionnants que les gens ont appris à résoudre aux ordinateurs, de la façon de gagner aux échecs ou au Jeopardy à la détermination des itinéraires routiers les plus courts. Mais il existe encore de nombreuses tâches que les ordinateurs ne peuvent pas effectuer, en particulier dans le domaine de la compréhension du langage humain. Les méthodes statistiques se sont avérées être un moyen efficace d'aborder ces problèmes, mais les techniques d'apprentissage automatique fonctionnent souvent mieux lorsque les algorithmes sont fournis avec des pointeurs sur ce qui est pertinent dans un ensemble de données, plutôt que de simples quantités massives de données. Lorsqu'on parle de langage naturel, ces pointeurs se présentent souvent sous la forme d'annotations, des métadonnées qui fournissent des informations supplémentaires sur le texte. Cependant, pour enseigner efficacement un ordinateur, il est important de lui donner les bonnes données et de disposer de suffisamment de données pour apprendre. Dans cette section, nous parlerons des outils utilisés pour créer de bonnes données pour les tâches de ML.

3.1 L'importance de l'annotation linguistique

Tout le monde sait qu'Internet est une ressource incroyable pour toutes sortes d'informations qui peuvent vous apprendre à peu près tout : jongler, programmer, jouer d'un instrument, etc. Cependant, Internet contient une autre couche d'informations, et c'est ainsi que toutes ces leçons (et blogs, forums, tweets, etc.) sont communiquées. Le Web contient des informations sous toutes les formes de médias, y compris des textes, des images, des films et des sons, et la langue est le moyen de communication qui permet aux gens de comprendre le contenu et de lier le contenu à d'autres médias. Cependant, alors que les ordinateurs sont excellents pour fournir ces informations aux utilisateurs intéressés, ils sont beaucoup moins aptes à comprendre le langage lui-même.

La linguistique théorique et informatique se concentre sur la découverte de la nature profonde du langage et la capture des propriétés informatiques des structures linguistiques. Les technologies du langage humain (HLT) tentent d'adopter ces connaissances et algorithmes et de les transformer en programmes fonctionnels et performants qui peuvent avoir un impact

sur la façon dont nous interagissons avec les ordinateurs en utilisant le langage. Avec de plus en plus de personnes utilisant Internet chaque jour, la quantité de données linguistiques disponibles pour les chercheurs a considérablement augmenté, permettant aux problèmes de modélisation linguistique d'être considérés comme des tâches de ML, plutôt que de se limiter aux quantités relativement faibles de données que les humains sont capables de traiter. Par eux-mêmes.

Cependant, il ne suffit pas de simplement fournir à un ordinateur une grande quantité de données et de s'attendre à ce qu'il apprenne à parler - les données doivent être préparées de manière à ce que l'ordinateur puisse plus facilement trouver des modèles et des inférences. Cela se fait généralement en ajoutant des métadonnées pertinentes à un ensemble de données. Toute balise de métadonnées utilisée pour baliser les éléments de l'ensemble de données est appelée une annotation de corpus. Cependant, pour que les algorithmes apprennent de manière efficace et efficiente, l'annotation effectuée sur les données doit être précise et pertinente pour la tâche que la machine est invitée à effectuer. Pour cette raison, la discipline de l'annotation du langage est un maillon essentiel dans le développement de technologies intelligentes du langage humain.

Donner trop d'informations à un algorithme de ML peut le ralentir et conduire à des résultats inexacts, ou faire en sorte que l'algorithme soit tellement adapté aux données d'entraînement qu'il devient « surdimensionné » et fournit des résultats moins précis qu'il ne le pourrait autrement sur de nouvelles données. Il est important de bien réfléchir à ce que vous essayez d'accomplir et aux informations les plus pertinentes.

Les ensembles de données en langage naturel sont appelés corpus, et un seul ensemble de données annotées avec la même spécification est appelé corpus annoté. Les corpus annotés peuvent être utilisés pour entraîner des algorithmes de ML.

4. Séparation de données

La procédure de division les données en données d'apprentissage et données de tests est utilisée pour estimer les performances des algorithmes d'apprentissage automatique lorsqu'ils sont utilisés pour faire des prédictions sur des données non utilisées pour entraîner le modèle.

Il s'agit d'une procédure simple et rapide à effectuer, dont les résultats vous permettent de comparer les performances des algorithmes d'apprentissage automatique pour votre problème de modélisation prédictive. Bien que simple à utiliser et à interpréter, il y a des moments où la

procédure ne doit pas être utilisée, comme lorsque vous avez un petit ensemble de données et des situations où une configuration supplémentaire est requise, comme lorsqu'elle est utilisée pour la classification et que l'ensemble de données n'est pas équilibré.

L'introduction précédente a introduit l'idée de diviser l'ensemble de données en deux sous-ensembles :

Ensemble d'entraînement : un sous-ensemble pour entraîner un modèle (Training Set).

Ensemble de test : un sous-ensemble pour tester le modèle entraîné (Test Set).

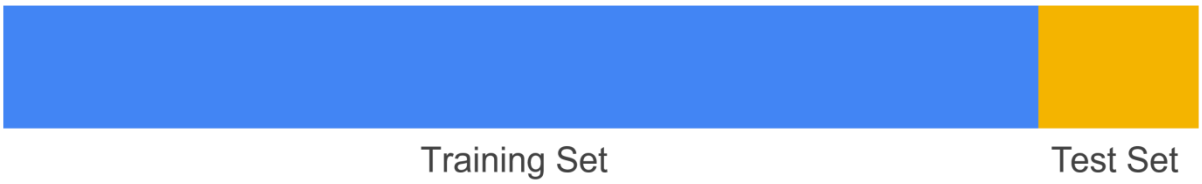


Figure 2.1 Découpage d'un seul ensemble de données en un ensemble d'apprentissage et un ensemble de test.

Assurez-vous que votre ensemble de test remplit les deux conditions suivantes :

1- Est suffisamment grand pour produire des résultats statistiquement significatifs.

2- Est représentatif de l'ensemble de données dans son ensemble. En d'autres termes, ne choisissez pas un ensemble de test avec des caractéristiques différentes de l'ensemble d'apprentissage.

En supposant que votre ensemble de test remplisse les deux conditions précédentes, votre objectif est de créer un modèle qui se généralise bien aux nouvelles données. Notre ensemble de test sert de proxy pour les nouvelles données. Par exemple, considérons la figure suivante. Notez que le modèle appris pour les données d'apprentissage est très simple. Ce modèle ne fait pas un travail parfait - quelques prédictions sont fausses. Cependant, ce modèle fait à peu près aussi bien sur les données de test que sur les données d'entraînement. En d'autres termes, ce modèle simple ne surajoute pas les données d'apprentissage.

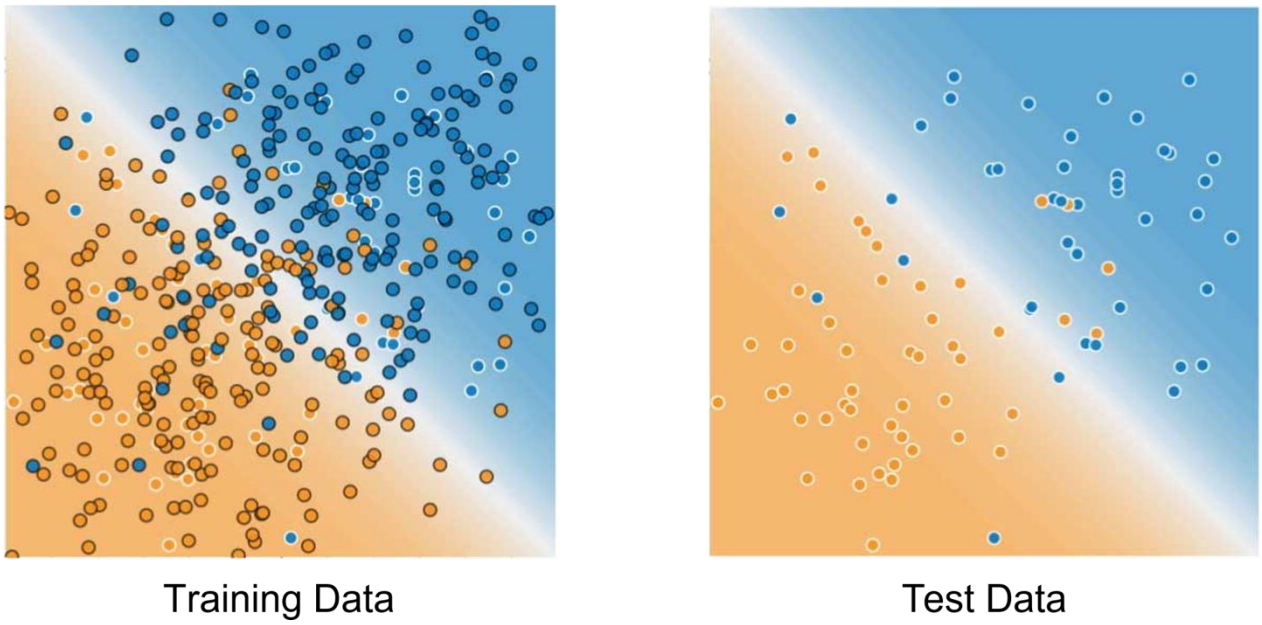


Figure 2.2 Validation du modèle entraîné par rapport aux données de test.

4.1 Ensemble de données d'entraînement

Un ensemble de données d'apprentissage est un ensemble d'exemples utilisé pendant le processus d'apprentissage et est utilisé pour ajuster les paramètres de, par exemple, un classificateur.

Pour les tâches de classification, un algorithme d'apprentissage supervisé examine l'ensemble de données d'apprentissage pour déterminer, ou apprendre, les combinaisons optimales de variables qui généreront un bon modèle prédictif. L'objectif est de produire un modèle entraîné (ajusté) qui se généralise bien à de nouvelles données inconnues. Le modèle ajusté est évalué à l'aide de ‘’nouveaux’’ exemples issus des ensembles de données conservés (ensembles de données de validation et de test) pour estimer la précision du modèle dans la classification de nouvelles données. Pour réduire le risque de problèmes tels que le sur-apprentissage, les exemples des ensembles de données de validation et de test ne doivent pas être utilisés pour entraîner le modèle.

4.2 Ensemble de données de test

Un ensemble de données de test est un ensemble indépendant de l’ensemble de données d'apprentissage, mais qui suit la même distribution de probabilité que le l’ensemble de

données d'apprentissage. Si un modèle ajusté à l'ensemble de données d'apprentissage s'adapte également bien à l'ensemble de données de test, un sur ajustement minimal a eu lieu.

Un ensemble de test est donc un ensemble d'exemples utilisés uniquement pour évaluer les performances (c'est-à-dire la généralisation) d'un classificateur entièrement spécifié. Pour ce faire, le modèle final est utilisé pour prédire les classifications des exemples dans l'ensemble de test. Ces prédictions sont comparées aux véritables classifications des exemples pour évaluer la précision du modèle.

L'ensemble de données de test est généralement utilisé pour évaluer le modèle final. Dans le cas où l'ensemble de données d'origine est partitionné en deux sous-ensembles (ensembles de données d'entraînement et de test), l'ensemble de données de test peut évaluer le modèle une seule fois (par exemple, dans la méthode d'exclusion). Certaines sources déconseillent une telle méthode. Cependant, lors de l'utilisation d'une méthode telle que la validation croisée, deux partitions peuvent être suffisantes et efficaces car les résultats sont moyennés après des cycles répétés d'entraînement et de test du modèle pour aider à réduire les biais et la variabilité.

4.3 Validation croisée

Un ensemble de données peut être divisé à plusieurs reprises en un ensemble de données d'apprentissage et un ensemble de données de validation : c'est ce qu'on appelle la validation croisée. Ces partitions répétées peuvent être effectuées de différentes manières, par exemple en les divisant en 2 ensembles de données égaux et en les utilisant comme entraînement/validation, puis validation/entraînement, ou en sélectionnant à plusieurs reprises un sous-ensemble aléatoire comme ensemble de données de validation. Pour valider les performances du modèle, un ensemble de données de test supplémentaire qui a été exclu de la validation croisée est parfois utilisé.

5. Algorithmes d'apprentissage automatiques

5.1 Types d'algorithmes d'apprentissage automatique

Il existe 3 types d'algorithmes d'apprentissage automatique (ML) :

5.1.1 Algorithmes d'apprentissage supervisé :

L'apprentissage supervisé utilise des données d'apprentissage étiquetées pour apprendre la fonction de mappage qui transforme les variables d'entrée (X) en variable de sortie (Y). En d'autres termes, il résout pour f dans l'équation suivante :

Y = f (X)

Cela nous permet de générer avec précision des sorties lorsque de nouvelles entrées sont données.

Il existe deux types d'apprentissage supervisé : la classification et la régression.

- **La classification** est utilisée pour prédire le résultat d'un échantillon donné lorsque la variable de sortie est sous forme de catégories. Un modèle de classification peut examiner les données d'entrée et essayer de prédire des étiquettes telles que ‘’malade’’ ou ‘’en bonne santé’’.

- **La régression** est utilisée pour prédire le résultat d'un échantillon donné lorsque la variable de sortie est sous la forme de valeurs réelles. Par exemple, un modèle de régression peut traiter des données d'entrée pour prédire la quantité de précipitations, la taille d'une personne, etc.

Les algorithmes que nous couvrons dans cette section

– Régression linéaire

– Régression logistique

– CART

– Naïve-Bayes

– K-Nearest Neighbors (KNN)

Sont des exemples d'apprentissage supervisé.

- **L'assemblage** est un autre type d'apprentissage supervisé. Cela signifie combiner les prédictions de plusieurs modèles d'apprentissage automatique qui sont individuellement faibles pour produire une prédiction plus précise sur un nouvel échantillon.

5.1.2 Algorithmes d'apprentissage non supervisé :

Les modèles d'apprentissage non supervisé sont utilisés lorsque nous n'avons que les variables d'entrée (X) et aucune variable de sortie correspondante. Ils utilisent des données d'apprentissage non étiquetées pour modéliser la structure sous-jacente des données.

- L'association est utilisée pour découvrir la probabilité de cooccurrence d'éléments dans une collection. Il est largement utilisé dans l'analyse du panier de marché. Par exemple, un modèle d'association peut être utilisé pour découvrir que si un client achète du pain, il/elle est susceptible à 80 % d'acheter également des œufs.

- Le clustering est utilisé pour regrouper des échantillons de telle sorte que les objets d'un même cluster soient plus similaires les uns aux autres qu'aux objets d'un autre cluster.

- La réduction de la dimensionnalité est utilisée pour réduire le nombre de variables d'un ensemble de données tout en garantissant que les informations importantes sont toujours transmises. La réduction de la dimensionnalité peut être effectuée à l'aide des méthodes d'extraction de caractéristiques et des méthodes de sélection de caractéristiques. Facture Sélection sélectionne un sous-ensemble des variables d'origine. L'extraction de caractéristiques effectue la transformation des données d'un espace de grande dimension vers un espace de faible dimension. Exemple : l'algorithme PCA est une approche d'extraction de caractéristiques.

5.1.3 L'apprentissage par renforcement

L'apprentissage par renforcement est un type d'algorithme d'apprentissage automatique qui permet à un agent de décider de la meilleure action suivante en fonction de son état actuel en apprenant des comportements qui maximiseront une récompense.

Les algorithmes de renforcement apprennent généralement les actions optimales par essais et erreurs. Imaginez, par exemple, un jeu vidéo dans lequel le joueur doit se déplacer à certains endroits à certains moments pour gagner des points. Un algorithme de renforcement jouant à ce jeu commencerait par se déplacer de manière aléatoire mais, au fil du temps, par essais et erreurs, il apprendrait où et quand il devait déplacer le personnage du jeu pour maximiser son total de points.

5.2 Exemple d’algorithmes d'apprentissage supervisé

5.2.1 Régression linéaire

En apprentissage automatique, nous avons un ensemble de variables d'entrée (x) qui sont utilisées pour déterminer une variable de sortie (y). Une relation existe entre les variables d'entrée et la variable de sortie. L'objectif du ML est de quantifier cette relation.

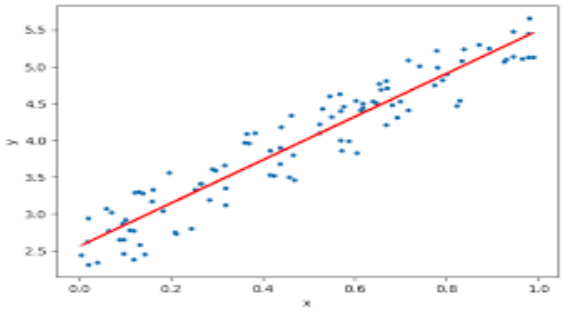


Figure 2.3 : La régression linéaire est représentée par une ligne sous la forme y = a + bx

Dans la régression linéaire, la relation entre les variables d'entrée (x) et la variable de sortie (y) est exprimée sous la forme d'une équation de la forme y = a + bx. Ainsi, le but de la régression linéaire est de connaître les valeurs des coefficients a et b. Ici, a est l'interception et b est la pente de la ligne.

La figure2.3 montre les valeurs x et y tracées pour un ensemble de données. L'objectif est d'ajuster une ligne qui est la plus proche de la plupart des points. Cela réduirait la distance (erreur) entre la valeur y d'un point de données et la ligne.

5.2.2 Régression logistique

Les prédictions de régression linéaire sont des valeurs continues (c'est-à-dire les précipitations en cm), les prédictions de régression logistique sont des valeurs discrètes (c'est-à-dire si un étudiant a réussi/échoué) après avoir appliqué une fonction de transformation.

La régression logistique est la mieux adaptée à la classification binaire : ensembles de données où y = 0 ou 1, où 1 désigne la classe par défaut. Par exemple, pour prédire si un événement se produira ou non, il n'y a que deux possibilités : qu'il se produise (que nous notons 1) ou qu'il ne se produise pas (0). Donc, si nous prédisions si un patient était malade, nous étiquèterions les patients malades en utilisant la valeur 1 dans notre ensemble de données.

La régression logistique est nommée d'après la fonction de transformation qu'elle utilise, appelée fonction logistique h(x)= 1/ (1 + ex). Cela forme une courbe en forme de S.

Dans la régression logistique, la sortie prend la forme de probabilités de la classe par défaut (contrairement à la régression linéaire, où la sortie est directement produite). Comme il s'agit d'une probabilité, la sortie se situe dans la plage 0-1. Ainsi, par exemple, si nous essayons de prédire si les patients sont malades, nous savons déjà que les patients malades sont notés 1, donc si notre algorithme attribue le score de 0,98 à un patient, il pense que le patient est très susceptible d'être malade.

Cette sortie (valeur y) est générée par une transformation logarithmique de la valeur x, en utilisant la fonction logistique h(x)= 1/ (1 + e^ -x). Un seuil est ensuite appliqué pour forcer cette probabilité dans une classification binaire.

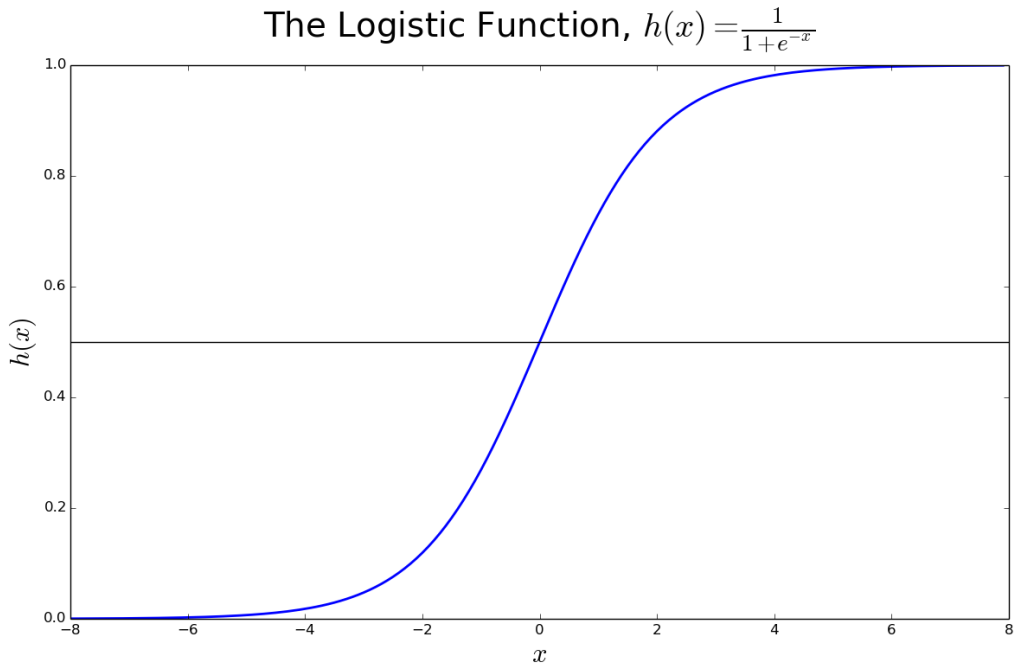


Figure 2.4 : Régression logistique pour déterminer si une tumeur est maligne ou bénigne. Classé comme malin si la probabilité h(x)>= 0,5

Dans la figure 2.4, pour déterminer si une tumeur est maligne ou non, la variable par défaut est y = 1 (tumeur = maligne). La variable x pourrait être une mesure de la tumeur, telle que la taille de la tumeur. Comme le montre la figure, la fonction logistique transforme la valeur x des différentes instances de l'ensemble de données, dans la plage de 0 à 1. Si la probabilité dépasse le seuil de 0,5 la tumeur est classée comme malin.

L'équation de régression logistique P(x) = e ^ (b0 +b1x) / (1 + e (b0 + b1x)) peut être transformée en ln(p(x) / 1-p(x)) = b0 + b1x.

L'objectif de la régression logistique est d'utiliser les données d'apprentissage pour trouver les valeurs des coefficients b0 et b1 de manière à minimiser l'erreur entre le résultat prévu et le résultat réel. Ces coefficients sont estimés à l'aide de la technique d'estimation du maximum de vraisemblance.

5.2.3 CART

Les arbres de classification et de régression (CART) sont une implémentation des arbres de décision.

Les nœuds non terminaux des arbres de classification et de régression sont le nœud racine et le nœud interne. Les nœuds terminaux sont les nœuds feuilles. Chaque nœud non terminal représente une seule variable d'entrée (x) et un point de division sur cette variable ; les nœuds feuilles représentent la variable de sortie (y). Le modèle est utilisé comme suit pour faire des prédictions : parcourir les divisions de l'arbre pour arriver à un nœud feuille et sortir la valeur présente au nœud feuille.

L'arbre de décision de la figure 2.5 ci-dessous classe si une personne achètera une voiture de sport ou une fourgonnette en fonction de son âge et de son état matrimonial. Si la personne a plus de 30 ans et n'est pas mariée, on parcourt l'arbre de la façon suivante : « plus de 30 ans ? » -> oui -> « marié ? » -> non. Par conséquent, le modèle produit une voiture de sport

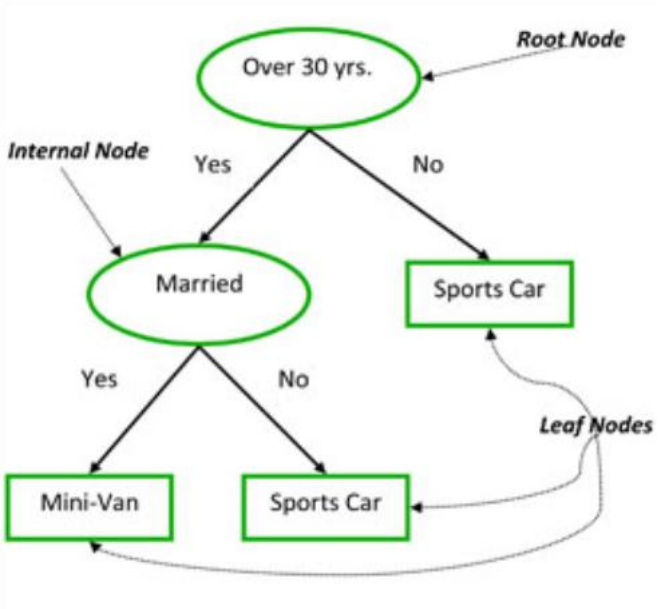


Figure 2.5: Parts of a decision tree

5.2.4 Bayes naïf

Pour calculer la probabilité qu'un événement se produise, étant donné qu'un autre événement s'est déjà produit, nous utilisons le théorème de Bayes. Pour calculer la probabilité qu'une hypothèse (h) soit vraie, compte tenu de nos connaissances préalables (d), nous utilisons le théorème de Bayes comme suit :

$P(h|d) = (P(d|h)\ P(h)) / P(d)$

Où :

P(h|d) = Probabilité postérieure. La probabilité que l'hypothèse h soit vraie, étant donné les données d, où $P(h|d) = P(d1|\ h)\ P(d2|\ h)\ldots P(dn|\ h)\ P(d)$

P(d|h) = Probabilité. La probabilité des données d étant donné que l'hypothèse h était vraie.

P(h) = probabilité a priori de classe. La probabilité que l'hypothèse h soit vraie (indépendamment des données)

P(d) = probabilité a priori du prédicteur. Probabilité des données (quelle que soit l'hypothèse)

Cet algorithme est appelé « naïf » car il suppose que toutes les variables sont indépendantes les unes des autres, ce qui est une hypothèse naïve à faire dans des exemples du monde réel.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Figure 2.6 : exemple de prédiction du statut de ‘’play’’ à l'aide de la variable ‘’ weather ‘’.

En utilisant la figure 2.6 comme exemple, quel est le résultat si météo = « sunny » ?

Pour déterminer le résultat play = « yes » ou « no » étant donné la valeur de la variable weather = « sunny », calculez P(yes|sunny) et P(no|sunny) et choisissez le résultat avec une probabilité plus élevée.

->P(yes|sunny)= (P(sunny|yes) * P(yes)) / P(sunny) = (3/9 * 9/14) / (5/14) = 0,60

-> P(no|sunny)= (P(sunny|no) * P(no)) / P(sunny) = (2/5 * 5/14) / (5/14) = 0,40

Ainsi, si le weather = « synny », le résultat est Play = « yes ».

5.2.5 KNN

L'algorithme K-Nearest Neighbors utilise l'ensemble de données complet comme ensemble d'apprentissage, plutôt que de diviser l'ensemble de données en un ensemble d'apprentissage et un ensemble de test.

Lorsqu'un résultat est requis pour une nouvelle instance de données, l'algorithme KNN parcourt l'ensemble de données pour trouver les k instances les plus proches de la nouvelle instance, ou le nombre k d'instances les plus similaires au nouvel enregistrement, puis génère la moyenne des résultats (pour un problème de régression) ou du mode (classe la plus fréquente) pour un problème de classification. La valeur de k est spécifiée par l'utilisateur.

6. Mesures de performance

Après avoir implémenté un modèle et obtenu des résultats sous la forme d'une probabilité ou d'une classe, l'étape suivante consiste à déterminer l'efficacité du modèle basé sur une métrique à l'aide d'ensembles de données de test. Différentes mesures de performance sont utilisées pour évaluer différents algorithmes d'apprentissage automatique. Nous nous concentrerons sur ceux utilisés pour les problèmes de classification. Nous pouvons utiliser des métriques de performance de classification telles que la précision, l'AUC (surface sous la courbe), la précision, le rappel, qui peuvent être utilisées pour trier les algorithmes principalement utilisés par les moteurs de recherche.

Les métriques que nous choisissons pour évaluer notre modèle d'apprentissage automatique sont très importantes. Le choix des métriques influence la façon dont les performances des algorithmes d'apprentissage automatique sont mesurées et comparées.

6.1.1 Matrice de confusion :

La matrice de confusion est l'une des métriques les plus intuitives et les plus simples utilisées pour trouver l'exactitude et la précision du modèle. Il est utilisé pour les problèmes de classification où la sortie peut être de deux ou plusieurs types de classes.

Pour plus d'explications, disons que nous résolvons un problème de classification où nous prédisons si une personne a un cancer ou non.

Donnons une étiquette à notre variable cible :

1 : Quand une personne a un cancer

0 : Lorsqu'une personne n'a PAS de cancer.

Maintenant que nous avons identifié le problème, la matrice de confusion est un tableau à deux dimensions (« Réel » et « Prévu »), et des ensembles de « classes » dans les deux dimensions. Nos classifications réelles sont des colonnes et celles prévues sont des lignes.

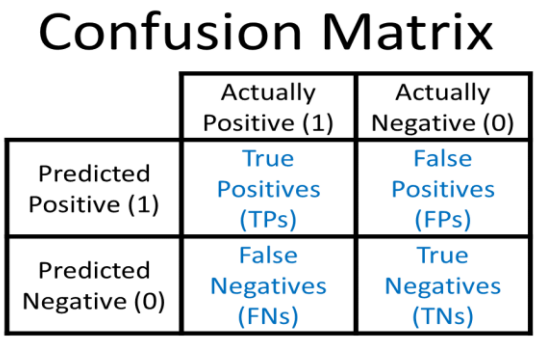


Figure 2.7 : Matrice de confusion

La matrice de confusion en soi n'est pas une mesure de performance en tant que telle, mais presque toutes les mesures de performance sont basées sur la matrice de confusion et les chiffres qu'elle contient.

6.1.2 Accuracy :

La précision dans les problèmes de classification est le nombre de prédictions correctes faites par le modèle sur toutes sortes de prédictions faites.

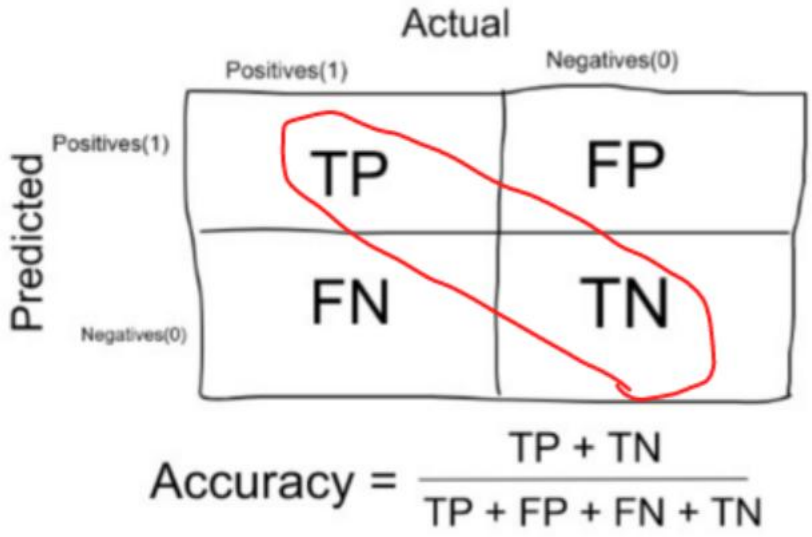


Figure 2.7 : Précision (Accuracy)

Dans le numérateur, se trouvent nos prédictions correctes (vrais positifs et vrais négatifs) (marquées en rouge dans la figure ci-dessus) et dans le dénominateur, sont le genre de toutes les prédictions faites par l'algorithme (bonnes et fausses).

Quand utiliser Accuracy:

La précision est une bonne mesure lorsque les classes de variables cibles dans les données sont presque équilibrées.

Exemple : 60 % des classes dans nos données d'images de fruits sont des pommes et 40 % sont des oranges.

Un modèle qui prédit si une nouvelle image est Apple ou Orange, 97% des fois correctement est une très bonne mesure dans cet exemple.

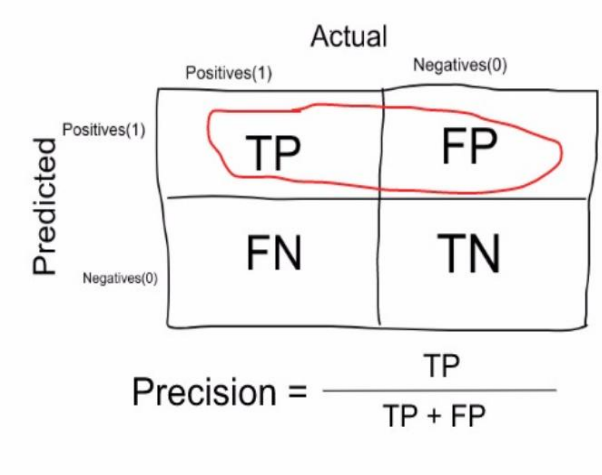
Quand NE PAS utiliser **Accuracy**:

La précision ne doit JAMAIS être utilisée comme mesure lorsque les classes de variables cibles dans les données sont majoritaires d'une classe.

Exemple : Dans notre exemple de détection de cancer avec 100 personnes, seules 5 personnes ont un cancer. Disons que notre modèle est très mauvais et prédit chaque cas comme No Cancer. Ce faisant, il a classé correctement ces 95 patients non cancéreux et 5 patients cancéreux comme non cancéreux. Maintenant, même si le modèle est terrible pour prédire le cancer, la précision d'un si mauvais modèle est également de 95%.

6.1.3 Précision :

Utilisons la même matrice de confusion que celle que nous avons utilisée précédemment pour notre exemple de détection du cancer.

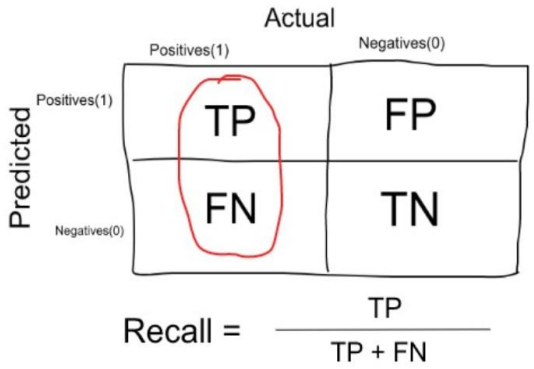


La précision est une mesure qui nous dit quelle proportion de patients que nous avons diagnostiqués comme ayant un cancer, avaient réellement un cancer. Les positifs prédits (les personnes prédites comme cancéreuses sont TP et FP) et les personnes ayant réellement un cancer sont TP.

Figure 2.8 : Précision (Precisinon)

Exemple : Dans notre exemple de cancer avec 100 personnes, seulement 5 personnes ont un cancer. Disons que notre modèle est très mauvais et prédit chaque cas comme un cancer. Puisque nous prédisons que tout le monde a un cancer, notre dénominateur (vrais positifs et faux positifs) est 100, et le numérateur, une personne ayant un cancer et le modèle prédisant son cas comme cancer est 5. Donc dans cet exemple, nous pouvons dire que le La précision d'un tel modèle est de 5%.

6.1.4 Rappel



Le rappel est une mesure qui nous indique quelle proportion de patients qui ont réellement eu un cancer a été diagnostiquée par l'algorithme comme ayant un cancer. Les vrais positifs (les personnes atteintes de cancer sont TP et FN) et les personnes diagnostiquées par le modèle ayant un cancer sont TP. (Remarque : FN est inclus parce que la personne avait en fait un cancer même si le modèle avait prédit le contraire).

Figure 2.9 : Rappel ou sensibilité

Exemple : Dans notre exemple de cancer avec 100 personnes, 5 personnes ont en fait un cancer. Disons que le modèle prédit chaque cas comme un cancer.

Ainsi, notre dénominateur (vrais positifs et faux négatifs) est 5 et le numérateur, personne atteinte de cancer, et le modèle prédisant son cas en tant que cancer est également 5 (puisque nous avons prédit correctement 5 cas de cancer). Donc dans cet exemple, on peut dire que le Rappel d'un tel modèle est de 100%. Et la précision d'un tel modèle (comme nous l'avons vu ci-dessus) est de 5%

Quand utiliser Précision et Quand utiliser Rappel :

Il est clair que le rappel nous donne des informations sur les performances d'un classificateur par rapport aux faux négatifs (combien en avons-nous manqué), tandis que la précision nous

donne des informations sur ses performances par rapport aux faux positifs (combien avons-nous été pris).

La *précision,* c'est être précis. Donc, même si nous avons réussi à capturer un seul cas de cancer, et nous l'avons capturé correctement, alors nous sommes précis à 100 %.

Le *rappel* ne vise pas tant à saisir correctement les cas, mais plus à saisir tous les cas qui ont « cancer » avec la réponse « cancer ». Donc, si nous disons simplement toujours que chaque cas est « cancer », nous avons un rappel de 100 %.

Donc si nous voulons nous concentrer davantage sur la minimisation des faux négatifs, nous voudrions que notre rappel soit aussi proche que possible de 100% sans que la précision soit trop mauvaise et si nous voulons nous concentrer sur la minimisation des faux positifs, alors notre objectif devrait être de faire Précision aussi proche que possible de 100%.

6.1.5 F Mesure(F Score) :

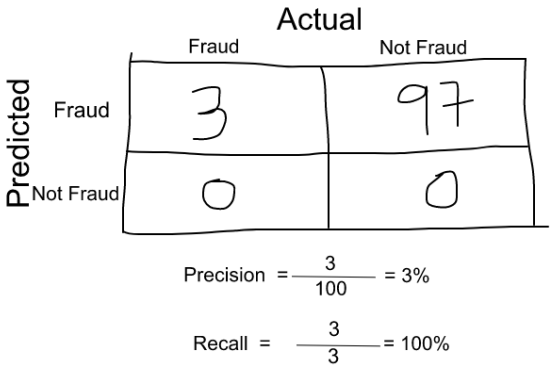
C’est mieux si nous pouvons obtenir un seul score qui représente à la fois la précision (P) et le rappel (R).

Une façon de le faire est simplement de prendre leur moyenne arithmétique. C’est-à-dire

Moyenne arithmétique = (P + R) / 2

Où **P** est la précision et **R** est le rappel. Mais c'est assez mauvais dans certaines situations.

Supposons que nous ayons 100 transactions par carte de crédit, dont 97 légitimes et 3 frauduleuses, et disons que nous avons trouvé un modèle qui prédit tout comme une fraude. (Horrible non !?)



La précision et le rappel pour l'exemple sont illustrés dans la figure ci-dessous.

Précision et rappel pour l'exemple de carte de crédit Maintenant, si nous prenons simplement la moyenne arithmétique des deux, alors cela revient à près de 51%. Nous ne devrions pas donner un score aussi modéré à un modèle terrible, car il ne fait que prédire chaque transaction comme une fraude.

Figure 2.10 : Précision et rappel pour l'exemple des cartes de crédit

Donc, nous avons besoin de quelque chose de plus équilibré que les moyennes arithmétiques et c'est la moyenne harmonique.

La moyenne harmonique est une sorte de moyenne lorsque x et y sont égaux. Mais lorsque x et y sont différents, alors il est plus proche du plus petit nombre que du plus grand nombre.

Pour notre exemple précédent, Score F = Moyenne harmonique (Précision, Rappel)

Score F = 2 * Précision * Rappel / (Précision + Rappel) = 2*3*100/103 = 5%

Donc, si un nombre est vraiment petit entre la précision et le rappel, le score F lève en quelque sorte un drapeau et est plus proche du plus petit nombre que du plus grand, donnant au modèle un score approprié plutôt qu'une simple moyenne arithmétique.

7. Conclusion

L'apprentissage automatique est un domaine d'étude qui examine l'utilisation d'algorithmes de calcul pour transformer des données empiriques en modèles utilisables. Le domaine de l'apprentissage automatique est né des communautés traditionnelles de statistiques et d'intelligence artificielle. Grâce aux efforts de méga-entreprises telles que Google, Microsoft, Facebook, Amazon, etc., l'apprentissage automatique est devenu l'un des sujets les plus brûlants de la science informatique au cours de la dernière décennie. Grâce à leurs processus commerciaux, d'immenses quantités de données ont été et seront collectées. Cela a permis de redynamiser les approches statistiques et informatiques pour générer automatiquement des modèles utiles à partir de données.

Chapitre 3 : Fouille d’opinion des données clients

1. Introduction

Ce travail a deux objectifs connexes ; d'abord, manipuler des données au format texte et en extraire des informations utiles. La deuxième consiste à utiliser ces informations pour prendre des décisions. C'est pourquoi nous avons choisi de combiner le NLP et text mining avec l'ensemble de données (ClientReviews) d'Amazon pour créer un modèle de sentiment qui peut prédire le sentiment du client à partir d'un texte de commentaire.

Les données Amazon offre une gamme très diversifiée et complexe de données textuelles, les utilisateurs écrivent ce qu'ils pensent d'un produit spécifique et comment ils le pensent. Au moment de la rédaction, il n'y a pas de règles d'orthographe, même si la grammaire est prise en compte, juste des mots, des prix, des URL, des dates, des notes et des avis clients. C'est précisément le fait que le texte n'est pas particulièrement bien écrit.

2. Approche de travail

Nous voyons que les données Amazon peuvent être très désordonnées, c'est pourquoil'étape1 sera consacrée à la façon de nettoyer l'ensemble de données en utilisant des outils de fouille de textes et le NLP, nous essaierons de comprendre quelles informations sont utiles et lesquelles n'est pas. Nous ne gardons donc que les éléments utiles pour plus tard.

Dans la deuxième étape, nous essaierons de représenter ces données texte restantes, en essayant de respecter une structure spécifique afin qu'elles puissent être utilisées par notre modèle d'apprentissage automatique.

Dans le l'étape3, nous construirons enfin notre modèle d'analyse des sentiments qui sera capable de classer les avis nouveaux et inconnus documents.

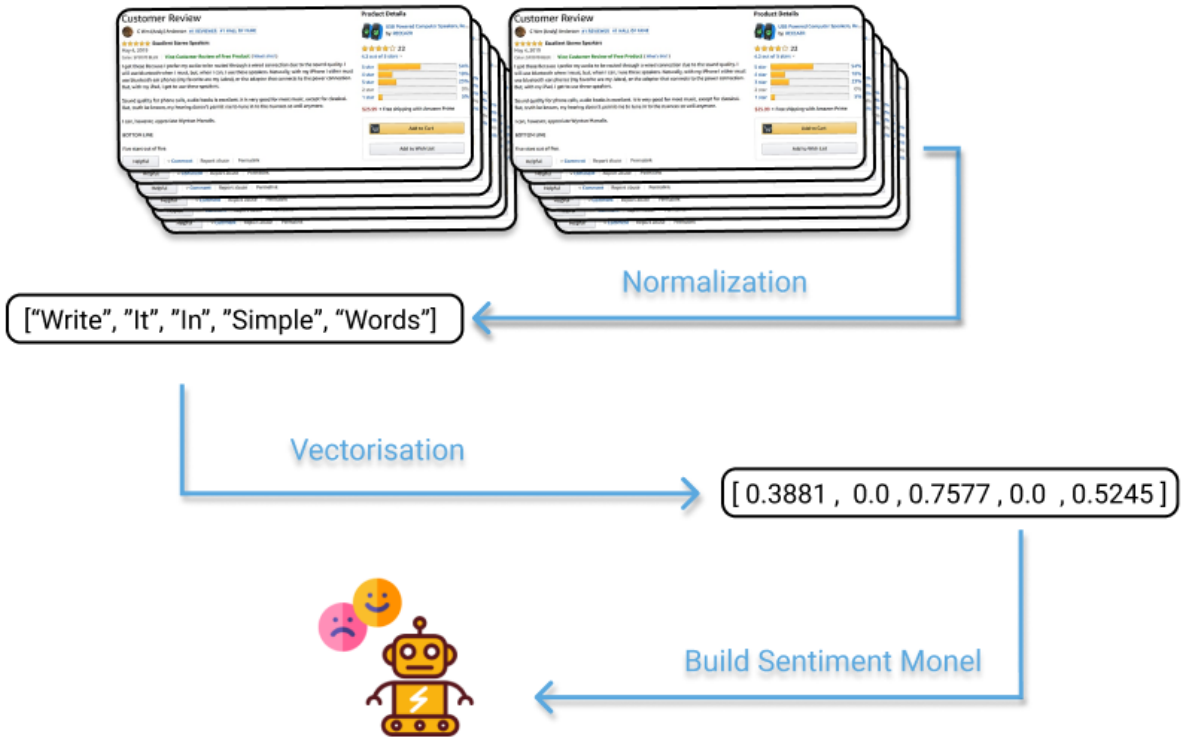


Figure 3.1 plan de travail

Pour créer un modèle de sentiment qui peut prédire le sentiment d'un avis client dans le format textuel, nous avons implémenté l'algorithme suivant.

Algorithme

```
var:amazoncustomer_reviews
Début
     -Extraire les données de l'ensemble de données
-Nettoyage des données
Supprimer les données inutiles
Vérifier les valeurs nulles
Vérifier si l'ensemble de données est équilibré
Équilibrer l'ensemble de données s'il est déséquilibré
-Normalisation des données
```

Écrire des fonctions en utilisant des expressions régulières pour

Supprimerles motindésirables dans le texte

-Tokenisation des données

Créer un tokenizer personnalisé pour supprimer les mots qui ne porte pas un sens tel que des mots vides ou les signes de ponctuations,

Puis transformer le texte brut en une liste de mots significatifs (liste de jetons)

-Vectorisation des données

Utiliser le vectoriseur TF-IDF pour transformer les jetons en une matrice que nous pouvons utiliser avec des algorithmes d'apprentissage automatique

-Appliquer la normalisation, la tokenisation et la vectorisation à notre ensemble de données

-Diviser les données en données d'entraînement et données de test

-Créer 3 modèles différents

-Évaluer et comparer les performances du modèle

Fin.

Nous choisissons de travailler avec un data set d'avis de consommateurs sur les produits Amazon, que nous parvenons à télécharger sous forme de fichier .csv

L’ensemble de données ressemble à ceci dans un format de fichier .csv

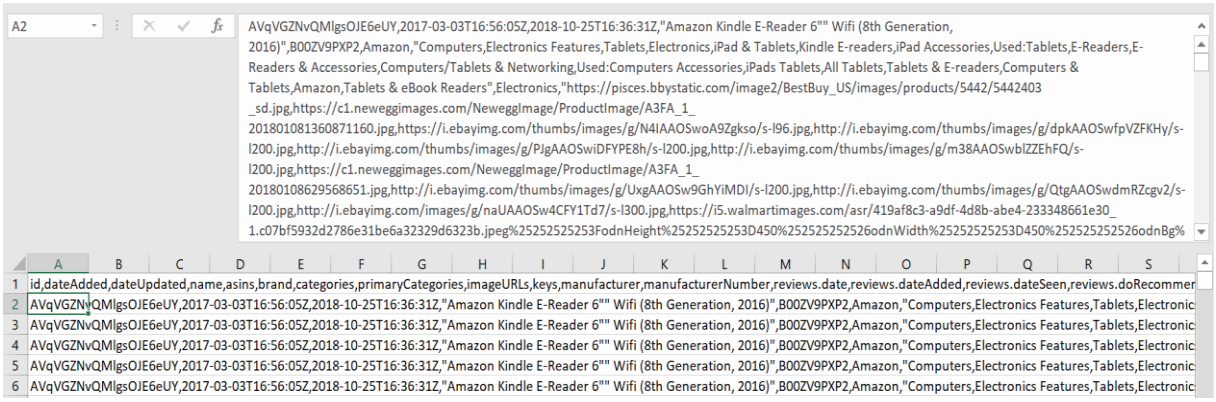


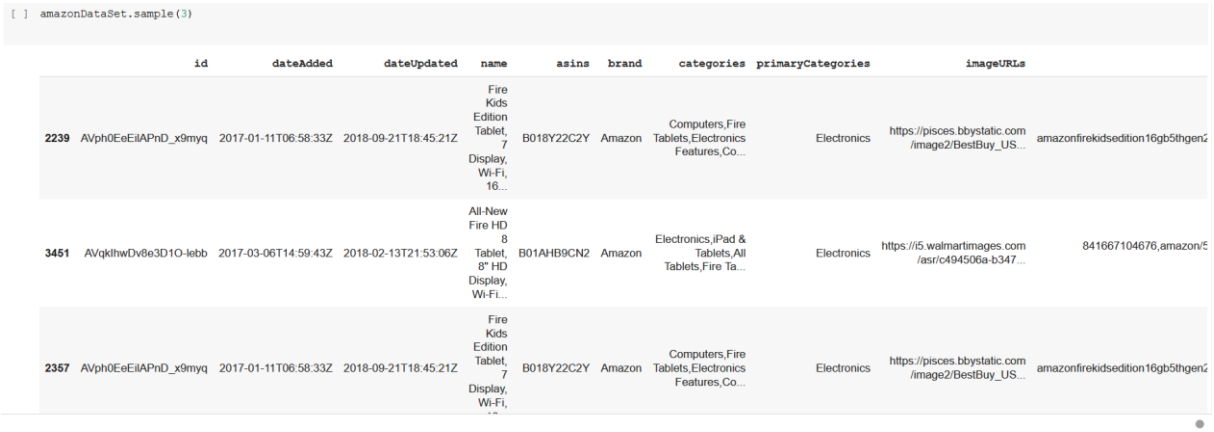
Figure 3.1 : un aperçu d'un échantillon de l'ensemble de données

3. Section 1 : Présentation de data set

Dans cette section, nous allons examiner l'ensemble de données que nous utiliserons.

Panda est une bibliothèque Python utilisée pour travailler avec des ensembles de données. Il a des fonctions d'analyse, de nettoyage, d'exploration et de manipulation des données. Panda nous permet d'analyser une grande quantité de données et de tirer des conclusions basées sur des théories statistiques. Panda peut nettoyer des ensembles de données désordonnés et les rendre lisibles et pertinents.

En utilisant le package python " pandas ", nous obtenons le Data frame stocké dans le fichier .csv,Ce qui nous donne la **figure 3.2**



	id	dateAdded	dateUpdated	name	asins	brand	categories	primaryCategories	imageURLs	
2239	AVph0EeEilAPnD_x9myq	2017-01-11T06:58:33Z	2018-09-21T18:45:21Z	Fire Kids Edition Tablet, 7 Display, Wi-Fi, 16...	B018Y22C2Y	Amazon	Computers,Fire Tablets,Electronics Features,Co...	Electronics	https://pisces.bbystatic.com/image2/BestBuy_US...	amazonfirekidsedition16gb5thgen2
3451	AVqklhwDv8e3D1O-lebb	2017-03-06T14:59:43Z	2018-02-13T21:53:06Z	All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	Electronics	https://i5.walmartimages.com/asr/c494506a-b347...	841667104676,amazon/5
2357	AVph0EeEilAPnD_x9myq	2017-01-11T06:58:33Z	2018-09-21T18:45:21Z	Fire Kids Edition Tablet, 7 Display, Wi-Fi,	B018Y22C2Y	Amazon	Computers,Fire Tablets,Electronics Features,Co...	Electronics	https://pisces.bbystatic.com/image2/BestBuy_US...	amazonfirekidsedition16gb5thgen2

Figure 3.2 : Exemple d'ensemble de données

à l'aide de la fonction "**`dtypes()`**", nous pouvons vérifier les types de données et les noms des attributs de l'ensemble de données.

```
amazonDataSet.dtypes
```

```
id                      object
dateAdded               object
dateUpdated             object
name                    object
asins                   object
brand                   object
categories              object
primaryCategories       object
imageURLs               object
keys                    object
manufacturer            object
manufacturerNumber      object
reviews.date            object
reviews.dateAdded       object
reviews.dateSeen        object
reviews.doRecommend       bool
reviews.id             float64
reviews.numHelpful       int64
reviews.rating           int64
reviews.sourceURLs      object
reviews.text            object
reviews.title           object
reviews.username        object
sourceURLs              object
dtype: object
```

Figure 3.3 : les attributs de data frame

3.1 Nettoyage des données

Après avoir extrait les données utilisables de l'ensemble de données et converti (vrai, faux) en (0,1) nous nous retrouvons avec un bloc de données qui ne contient que les attributs.

-"reviews.DoRecommend" **:** qui représente si le client qui a envoyé l'avis recommande ou non le produit que nous utiliserons au sein de notre variable indépendante

- "reviews.text"**:**le texte de l'avis réel

-"avis.Rating" **:** représente la note que le client attribue au produit

	reviews.doRecommend	reviews.rating	reviews.text
2164	1	5	Bought for a friend with basic needs. Money wa...
3004	1	4	This speaker's forte is it's drop and charge a...
3735	1	5	This product is easy to use. Screen size is gr...

Figure 3.4 : la data utilisable

Maintenant nous avons les données utilisables que nous avons vérifiées s'il y a des valeurs nulles car l'existence de valeurs nulles affecte la précision du modèle vérification des valeurs nulles.

```
[ ]  usebleData.isnull().sum()
     inputData = usebleData.drop(columns=['reviews.doRecommend','reviews.rating'])
     inputData.sample(3)
     usebleData.isnull().sum()
```

```
reviews.doRecommend    0
reviews.rating         0
reviews.text           0
dtype: int64
```

Figure 3.5 : vérification des valeurs nulles

3.2 Visualisation de l'ensemble de données

Matplotlib est une bibliothèque de traçage de graphiques de bas niveau en python qui sert d'utilitaire de visualisation. A l'aide de cet outil, nous pouvons obtenir des informations dans un format graphique qui nous aide à mieux comprendre les données.

Comprendre les données, dans ce cas, nous l'avons utilisé pour voir si notre ensemble de données est équilibré ou non.

```
[ ]  doRecommend = amazonDataSet["reviews.doRecommend"].value_counts()
     plt.pie(doRecommend, labels=doRecommend.index,
             autopct='%1.1f%%', shadow=True, startangle=90)
     plt.show()
```

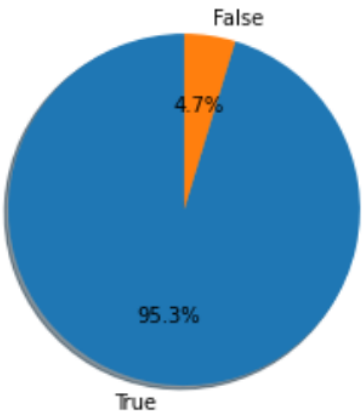


Figure 3.6 : vérification de l'équilibre des données

Nous notons que l'ensemble de données n'est pas équilibré, un ensemble de données déséquilibré affectera la précision renverra souvent un résultat négatif tout le temps que nous essayons de prédire quelque chose, ce qui entraîne un modèle inexact d'un modèle d'entraînement sur un ensemble de données qui a une majorité de valeurs négatives.

3.3 Traitement des données déséquilibré

Il existe plusieurs techniques pour gérer un ensemble de données déséquilibré pour simplicité, nous choisissons de gérer le déséquilibre en utilisant le sous-échantillonnage, ce qui signifie créer un ensemble de données qui contient toutes les entrées négatives et un nombre égal d'entrées positives, et nous nous retrouvons avec des données beaucoup plus petites 470 dans ce cas.

```
[ ]  doRecommendClass = usebleData[usebleData['reviews.doRecommend']==True]
     doNotRecommendClass = usebleData[usebleData['reviews.doRecommend']==False]

     doRecommendClass.shape,doNotRecommendClass.shape
```

```
     ((4765, 3), (235, 3))
```

```
[ ]  pos = doRecommendClass.sample(235)
     nig = doNotRecommendClass
     balancedDataset = pd.concat([pos,nig],axis=0)
     balancedDataset.shape
```

```
     (470, 3)
```

Figure 3.7 :sous-échantillonner l'ensemble de données

Si nous le vérifions maintenant avec Matplotlib, nous obtenons ceci

```
[ ]  doRecommend = balancedDataset["reviews.doRecommend"].value_counts()
     plt.pie(doRecommend, labels=doRecommend.index,
             autopct='%1.1f%%', shadow=True, startangle=90)
     plt.show()
```

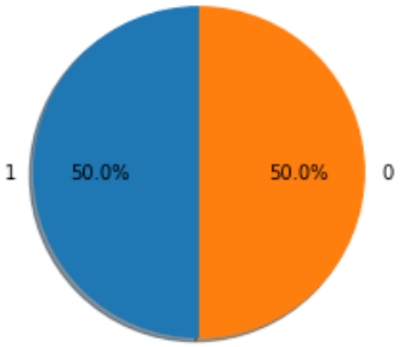


Figure 3.8 : l'ensemble de données équilibré

4. Section 2 : Normalisation du texte

Maintenant que nous avons une vision claire des données en main. Dans cette section, nous utiliserons différentes techniques d'exploration de données pour effectuer la normalisation du texte.

Le but de la normalisation du texte est de nettoyer le mieux possible le désordre que nous avons dans notre ensemble de données. Cela implique de passer par le nettoyage de certaines caractéristiques précises, mais aussi d'utiliser des outils puissants pour la tokenisation et le stemming.

La normalisation du texte visait en fait à réduire le caractère aléatoire d'un morceau de texte particulier.

4.1 RegEx

RegEx ou Regular Expression, est une séquence de caractères qui forme un modèle de recherche. RegEx peut être utilisé pour vérifier si une chaîne contient le modèle de recherche spécifié.

À l'aide de cet outil, nous avons écrit des fonctions pour supprimer les hashtags, les URL, les majuscules...

Ceci est une liste des fonctions que nous avons utilisées plus tard pour effectuer le nettoyage de texte

-delete_urls ()

```
[25] print('befor : '+ review)
     print ('after : '+ delete_urls(review))

     befor :  @review I love this!!!  https://Amazone.com #NLP #Fun
     after :  @review I love this!!!   #NLP #Fun
```

-delete_hashtag ()

```
[27] print('befor : '+ review)
     print ('after : '+ delete_hashtag(review))

     befor :  @review I love this!!!  https://Amazone.com #NLP #Fun
     after :  @review I love this!!!  https://Amazone.com NLP Fun
```

-to_loercase ()

```
[31] print('befor : '+ review)
     print ('after : '+ to_loewrcase(review))

     befor :  @review I LOVE THIS !!!  https://Amazone.com #NLP #Fun
     after :  @review i love this !!!  https://amazone.com #nlp #fun
```

-word_repetition ()

```
[36] print('befor : '+ review)
     print ('after : '+ word_repetition(review))

     befor :  @review hay loooook at this !!!  https://Amazone.com #NLP #Fun
     after :  @review hay look at this !!  https://Amazone.com #NLP #Fun
```

-punct_repetition ()

```
[8]  print('befor : '+ review)
     print ('after : '+ punct_repetition(review))

     befor :  @review hay loooook at this !!!????!!  https://Amazone.com #NLP #Fun
     after :  @review hay loooook at this !  https://Amazone.com #NLP #Fun
```

-**fix_contractions ()** : fonction utilisée pour remplacer les contractions par leurs formes étendues en utilisant le dictionnaire des contractions

Dictionnaire des contractions :

```
[ ] print(contractions.contractions_dict)
```

```
{"I'm": 'I am', "I'm'a": 'I am about to', "I'm'o": 'I am going to', "I've": 'I have', "I'll": 'I will', "I'll've": 'I will
```

-fix_contractions ()

```
[ ] print("unProcessed review: {}".format(review))
    print("Processed review: {}".format(fix_contractions(review)))
```

```
unProcessed review: LOOOOOOOOK at this ... I'd like it so much!
Processed review: LOOOOOOOOK at this ... I would like it so much!
```

4.2 Tokenisation

Les ordinateurs et les modèles d'apprentissage automatique ont besoin de phrases à jetons pour en savoir plus que simplement indiquer ce que sont les mots.

La tokenisation représente également une étape cruciale vers la représentation des mots via la factorisation.

NLTK

Le Natural Langage Toolkit, ou plus communément NLTK, est une suite de bibliothèques et de programmes pour le traitement symbolique et statistique du langage naturel (NLP) pour l'anglais écrit dans le langage de programmation Python.

Avec l'aide du package NLTK, nous construisons un tokenizer personnalisé qui supprime la ponctuation des mots pour ne garder que les morceaux qui ont un sens.

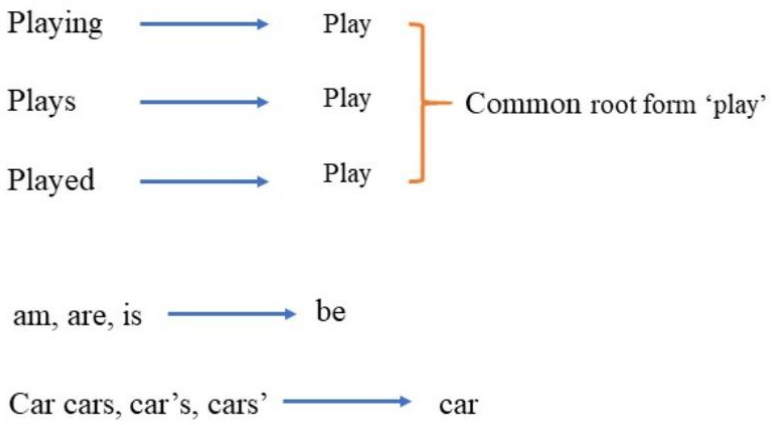
```
[ ] review = "these are 5 different words!?"
```

```
[ ] print("review tokens: {}".format(custom_tokenize(review,
                                                     keep_punct=True,
                                                     keep_alnum=True,
                                                     keep_stop=True)))
    print("review tokens: {}".format(custom_tokenize(review, keep_alnum=True)))
    print("review tokens: {}".format(custom_tokenize(review, keep_stop=True)))
    print("review tokens: {}".format(custom_tokenize(review, keep_punct=True)))
```

```
review tokens: ['these', 'are', '5', 'different', 'words', '!', '?']
review tokens: ['5', 'different', 'words']
review tokens: ['these', 'are', 'different', 'words']
review tokens: ['different', 'words']
```

4.3 Racinisation (Stemming)

Stem (racine) est la partie du mot à laquelle vous ajoutez des affixes de manière flexionnelle (changing/deriving) tels que (-ed,-ize, -s,-de,mis). Ainsi, la racine d'un mot ou d'une phrase peut entraîner des mots qui ne sont pas des mots réels. Les tiges sont créées en supprimant les suffixes ou les préfixes utilisés avec un mot.



Pour ce faire, nous avons écrit certaines fonctions en utilisant différentes techniques pour effectuer le stemming sur nos avis tokenizer.

```
[ ]  tokens = ["manager", "management", "managing"]
     print("Porter stems: {}".format(stem_tokens(tokens, porter_stemmer)))
     print("Lancaster stems: {}".format(stem_tokens(tokens, lancaster_stemmer)))
     print("Snowball stems: {}".format(stem_tokens(tokens, snoball_stemmer)))
```

```
Porter stems: ['manag', 'manag', 'manag']
Lancaster stems: ['man', 'man', 'man']
Snowball stems: ['manag', 'manag', 'manag']
```

4.4 Mettre tous ensemble

En utilisant toutes les fonctions ci-dessus, nous pourrions écrire une fonction **process_review ()** qui renverra une revue traitée sous la forme d'un tableau de toknes.

```
[ ] process_review(complex_review, verbose=True)

    Initial review: he looooook,
    THis is a big and complex review!!! ...
    We'd be glad if you couldn't normalize it!
    Check https://t.co/7777 and LET ME KNOW!!! #NLP

    review processing :
     he looooook,
    THis is a big and complex review!!! ...
    We'd be glad if you couldn't normalize it!
    Check  and LET ME KNOW!!! NLP

    Post Word processing review:
     he look,
    this is a big and complex review! .
    we would be glad if you could not normalize it!
    check  and let me know! nlp

    ['look',
     'big',
     'complex',
     'review',
     'would',
     'glad',
     'could',
     'not',
     'normal',
     'check',
     'let',
     'know',
     'nlp']
```

Maintenant que nous pouvons traiter nos avis, nous allons créer nos données d'entrée en appliquant la dernière fonction **process_review ()**au texte des avis.

```
[ ] balancedDataset["tokens"] = balancedDataset["reviews.text"].apply(process_review)
    balancedDataset.sample(10)
```

	reviews.doRecommend	reviews.rating	reviews.text	tokens
408	1	5	Great accessory for Alexis system..helpful and...	[great, accessori, alexi, fun]
2709	1	4	Great at home but still trying to figure out w...	[great, home, still, tri, figur, app, work, no...
2804	0	3	Its a decent reader for beginners. There is no...	[decent, reader, beginn, backlight, not, good,...
2505	0	3	I bought it cause I wanted to read more. That ...	[bought, caus, want, read, not, happen, tri, n...
1586	0	2	Can't really give a full review since the Kind...	[not, realli, give, full, review, sinc, kindl,...
330	0	2	I bought these as a gift for my adult kids to ...	[bought, gift, adult, kid, video, chat, colleg...

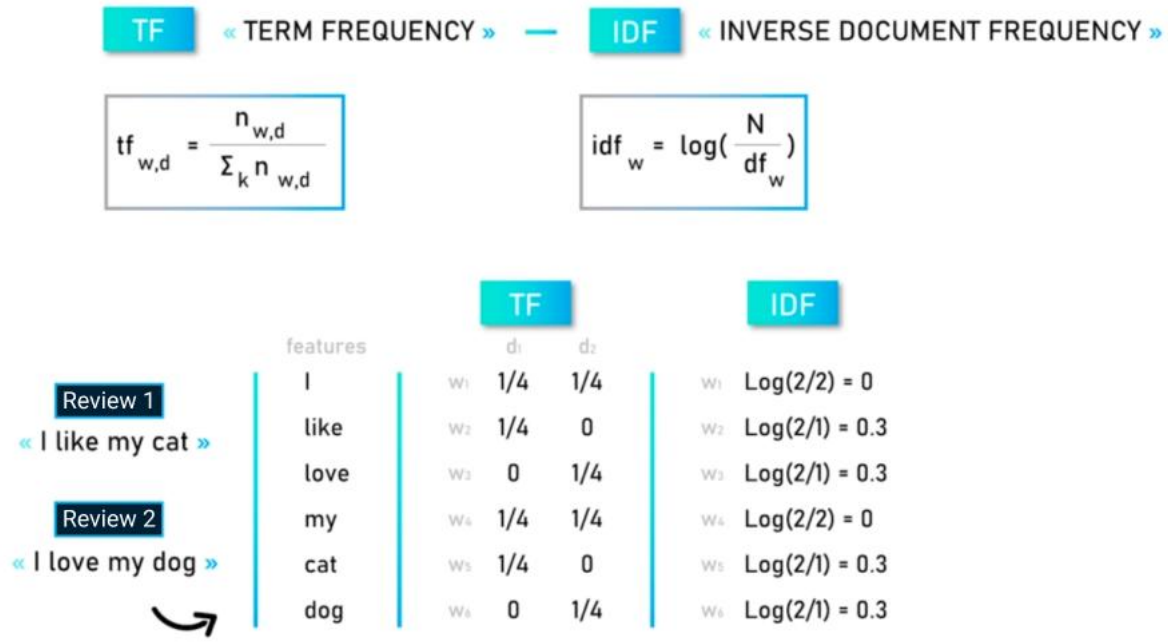
4.5 Représentation du texte

Pour la représentation textuelle, nous choisissons la technique TF-IDF.

Fréquence de terme - Fréquence de document inverse (TF-IDF)

TF-IDF, de l'anglais termfrequency-inverse document frequency, est une méthode de pondération souvent utilisée en recherche d'information et en particulier dans la fouille de textes. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Le poids augmente

proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus. Des variantes de la formule originale sont souvent utilisées dans des moteurs de recherche pour apprécier la pertinence d'un document en fonction des critères de recherche de l'utilisateur.



Fréquence du terme (TF)

La fréquence « brute » d'un terme est simplement le nombre d'occurrences de ce terme dans le document considéré (on parle de « fréquence » par abus de langage). On peut choisir cette fréquence brute pour exprimer la fréquence d'un terme.

Des variantes ont été proposées. Un choix plus simple, dit « binaire », est de mettre 1 si le terme apparaît dans le document et 0 sinon. À l'opposé, on peut normaliser logarithmiquement la fréquence brute pour amortir les écarts. Une normalisation courante pour prendre en compte la longueur du document est de normaliser par la fréquence brute maximale du document.

Fréquence inverse de document (IDF)

La fréquence inverse de document (inverse document frequency) est une mesure de l'importance du terme dans l'ensemble du corpus. Dans le schéma TF-IDF, elle vise à donner un poids plus important aux termes les moins fréquents,

Considérés comme plus discriminants. Elle consiste à calculer le logarithme (en base 10 ou en base 21) de l'inverse de la proportion de documents du corpus qui contiennent le terme :

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

où :

- $|D|$: nombre total de documents dans le corpus ;
- $|\{d_j : t_i \in d_j\}|$: nombre de documents où le terme t_i apparaît (c'est-à-dire $n_{i,j} \neq 0$).

Calcul de (TF-IDF)

Finalement, le poids s'obtient en multipliant les deux mesures :

$$\mathrm{tfidf_{i,j}} = \mathrm{tf_{i,j}} \cdot \mathrm{idf_i}$$

Exemple :

Prenons par exemple ce corpus

```
[ ]  corpus = [["i", "love", "nlp"],
               ["i", "miss", "you"],
               ["i", "love", "you"],
               ["you", "are", "happy", "to", "learn"],
               ["i", "lost", "my", "computer"],
               ["i", "am", "so", "sad"]]

     sentiment = [1, 0, 1, 1, 0, 0]
```

En utilisant la fonction **TfidfVectorizer ()** du package **Scikit-Learn**, nous pouvons vectoriser notre corpus comme il apparaît sous :

```
[ ]  tf_mtx.toarray()
```

```
array([[0.        , 0.        , 0.        , 0.        , 0.36831339,
        0.        , 0.        , 0.58951102, 0.        , 0.        ,
        0.71890333, 0.        , 0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.38819592,
        0.        , 0.        , 0.        , 0.75771163, 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.52457317],
       [0.        , 0.        , 0.        , 0.        , 0.43081598,
        0.        , 0.        , 0.68955073, 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.58216611],
       [0.        , 0.47249269, 0.        , 0.47249269, 0.        ,
        0.47249269, 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.47249269, 0.32711256],
       [0.        , 0.        , 0.55363834, 0.        , 0.28364372,
        0.        , 0.55363834, 0.        , 0.        , 0.55363834,
        0.        , 0.        , 0.        , 0.        , 0.        ],
       [0.55363834, 0.        , 0.        , 0.        , 0.28364372,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.55363834, 0.55363834, 0.        , 0.        ]])
```

Note : pour simplicité, nous avons sauté la documentation de certaines étapes

5. Section 3 : Modèle de sentiment

5.1 Séparation (Train /Test)

La fonction **train_test_split( )** sert à diviser les données en données de train et en données de test pour l'évaluation ultérieure des performances du modèle.

```
[ ]  X_train, X_test, y_train, y_test = train_test_split(X1, y1,
                                                         random_state=0,
                                                         train_size=0.80)
```

```
[ ]  print("Size of X_train: {}".format(len(X_train)))
     print("Size of y_train: {}".format(len(y_train)))
     print("\n")
     print("Size of X_test: {}".format(len(X_test)))
     print("Size of y_test: {}".format(len(y_test)))
     print("\n")
     print("Train proportion: {:.0%}".format(len(X_train)/
                                             (len(X_train)+len(X_test))))
```

```
Size of X_train: 376
Size of y_train: 376


Size of X_test: 94
Size of y_test: 94
```

5.2 Construire des modèles d'apprentissage automatique

Pour construire les modèles il suffit de transformer les données X_train et X_test en utilisant le vectoriseur.

Puis alimentez X_train avec Y-train aux fonctions fournies par le package Scikit-Learn dans ce travail que nous avons implémenté.

- Régression linéaire

```
[ ]  from sklearn.linear_model import LogisticRegression
```

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

- Régression logistique

```
[ ]  from sklearn import linear_model
```

```
[ ]  reg = linear_model.LinearRegression()
     reg.fit(X_train_tf, y_train)
```

-Naïf Bayes

```
[ ]  from sklearn.naive_bayes import MultinomialNB
     model = MultinomialNB()
     model.fit(X_train_count,y_train)
```

5.3 Indicateurs de performance

Nous pourrions imprimer la précision du modèle en comparant les prédictions et les sentiments réels à l'aide de la fonction **accuracy_score ().**

- Régression linéaire

```
[ ]  print("len reg Model Accuracy: {:.2%}".format(accuracy_score(y_test, y_pred_classes)))

     len reg Model Accuracy: 77.66%
```

- Régression logistique

```
[ ]  print("log reg Model Accuracy: {:.2%}".format(accuracy_score(y_test, y_pred_lr_tf)))

     log reg Model Accuracy: 72.34%
```

-Naïf Bayes

```
[ ]  X_test_count = v.transform(X_test)
     print("nav_bas Model Accuracy: {:.2%}".format(model.score(X_test_count, y_test)))

     nav_bas Model Accuracy: 78.72%
```

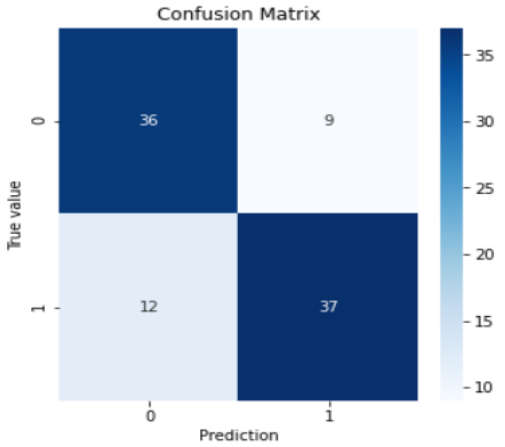
En utilisant cette fonction, nous pourrions tracer la matrice de confusion pour les différents modèles que nous allons créer

```
[ ]  import seaborn as sn

     def plot_confusion(cm):
       plt.figure(figsize = (5,5))
       sn.heatmap(cm, annot=True, cmap="Blues", fmt='.0f')
       plt.xlabel("Prediction")
       plt.ylabel("True value")
       plt.title("Confusion Matrix")
       return sn
```

Ce qui nous donne

- Régression linéaire

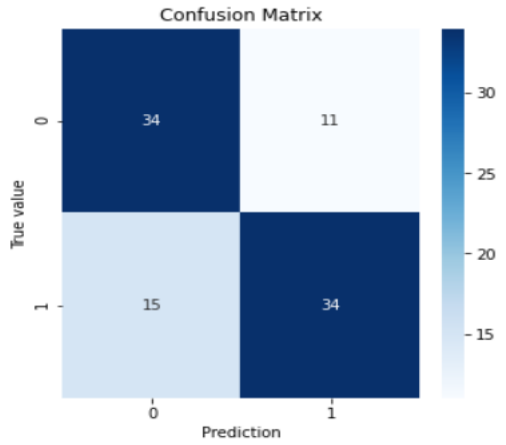
```
[ ]  plot_confusion(confusion_matrix(y_test, y_pred_classes))

     <module 'seaborn' from '/usr/local/lib/python3.7/dist-packages/seaborn/__init__.py'>
```



- Régression logistique

```
[ ]  plot_confusion(confusion_matrix(y_test, y_pred_lr_tf))
```

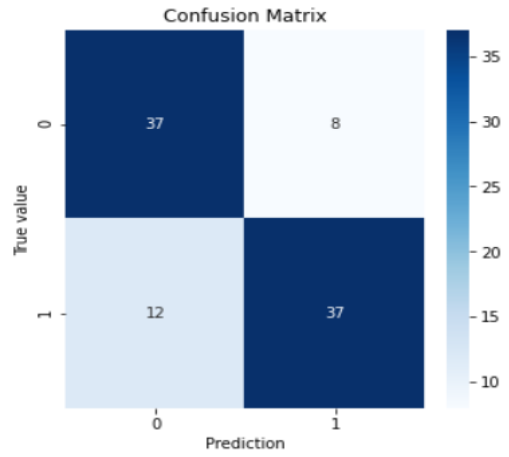
```
<module 'seaborn' from '/usr/local/lib/python3.7/dist-packages/seaborn/__init__.py'>
```



-Naïf Bayes

```
[ ]  plot_confusion(confusion_matrix(y_test, y_pred_nav_bas))
```

```
<module 'seaborn' from '/usr/local/lib/python3.7/dist-packages/seaborn/__init__.py'>
```



5.4 Comparaison

Après avoir obtenu les résultats des tests de nos modèles, nous calculons les différentes métriques de performance comme vous pouvez le voir dans le tableau ci-dessous.

	resultat du test				Indicateurs de performance			
model	**tp**	**fp**	**tn**	**fn**	**Accuracy**	**Precision**	**Rappel**	**F Score**
Régression linéaire	36	9	37	12	77,7%	80,0%	75,0%	77,5%
Régression logistique	34	11	34	15	72,3%	75,6%	69,4%	72,5%
Naïf Bayes	37	8	37	12	78,7%	82,2%	75,5%	78,9%

Comme le montre la comparaison, le naïf Bayes est le meilleur modèle pour la tâche en main. Pour cela nous recommandons ce modèle pour des travaux similaires.

6. Conclusion

Dans ce chapitre, nous avons présenté notre contribution au problème de la fouille d'opinion données de clients. Représentant les outils et les jeux de données utilise, ainsi que les étapes que nous avons suivies pour obtenir les résultats que nous montrons également pour trois différents modèles dont le but de faire la comparaison. Nous avons constaté la supériorité du classifieur Naïve Bayes dans ce type de problème en termes de performance.

Conclusion générale

Dans ce travail, nous avons exploré le domaine de l'analyse des sentiments ou fouille d'opinion qui, comme tous les autres domaines du traitement du langage naturel, a connu une évolution majeure depuis les années 2000 et a réalisé une évolution majeure et un grand intérêt depuis la naissance de l'apprentissage automatique.

Afin d'atteindre ces résultats, nous avons passé beaucoup de temps à lire et réviser des publications, des articles et des livres pour voir et comprendre les concepts et comment appliquer un modèle d'apprentissage automatique à notre problème.

Enfin, avant de passer aux perspectives, ce travail nous a permis de mettre en pratique notre connaissances d'apprentissage automatique, et le plus important est que nous fassions le premier pas vers l'apprentissage automatique, un des champs les plus importants de l'intelligence artificielle.

Comme perspectives à ce travaille, nous pouvons proposer:

• Le teste de notre modèle sur d'autres ensembles de données.

• Le développement et l'amélioration du modèle pour être plus précis dans la détection de la polarité des documents.

• Nous ne nous concentrons que sur les phrases en anglais, Il devrait être possible d'utiliser notre approche pour classer les sentiments dans d'autres langues.

Bibliographies

[1]. Pascal, Opzeeland, "6 *Proven Methods for Measuring Customer Satisfaction"*,userlike.com,30/10/ 2016,https://www.userlike.com/en/blog/6-proven-methods-for-measuring-your-customer-satisfaction.

[2]. -*"Consumer reviews of Amazon products"*,data.World,https://data.world/datafiniti/consumer-reviews-of-amazon-products

[3]. -Lisa, Tagliaferri,*" An Introduction to Machine Learning",*digitalocean.com,28/09/2017,https://www.digitalocean.com/community/tutorials/an-introduction-to-machine-learning

[4]. Jason, Brownlee, *"Train-Test Split for Evaluating Machine Learning Algorithms"*, machinelearningmastery.com, 24/07/2020 ,https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

[5]. -Juan, Villalobos, *"Test, training and validation sets"*,brainstobytes.com, 28/01/2020,https://www.brainstobytes.com/test-training-and-validation-sets/

[6]. -Reena, Shaw, *"The 10 Best Machine Learning Algorithms for Data Science Beginners"*,dataquest.io, 26/06/2019,https://www.dataquest.io/blog/top-10-machine-learning-algorithms-for-beginners/

[7]. Mohammed, Sunasra, *"Performance Metrics for Classification problems in Machine Learning"*,medium.com,11/10/2017,https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b

[8]. Great Learning," Sentiment *Analysis Using NLP"*, YouTube.com, 11/07/2020, https://www.youtube.com/watch?v=G6TbcyFxrms

[9]. TF-IDF,Wikipedia , 14/05/2021, https://fr.wikipedia.org/w/index.php?title=TF-IDF&oldid=182884607

[10]. . M. Aversa, "Spatial Big Data Analytics: The New Boundaries of RetailLocation Decision-Making," Wilfrid Laurier University, pp. 69-77, 2019.B. Liu, "Sentiment Analysis and Opinion Mining," Synthesis Lecturerson human language technologies, vol. 5, no. 1, pp. 1-167, 2012.T. Kim, "Trader sentiment on Alibaba is surging," CNBC News, London,

[11]. L. Liilian and Pang B, "“Thumbs up? Sentiment Classification usingMachine Learning Techniques” in Proceedings of EMNLP," pp. 79-86,

[12]. L. Bing, "Sentiment Analysis: A Multi-Faceted Problem," IEEE Intelligent Systems, vol. 25, no. 3, pp. 76-80, 2010.

[13]. E. Cambria and C. Havasi, "New Avenues in Opinion Mining and Sentiment Analysis," IEEE Intelligent Systems, vol. 28, no. 2, p. IEEE Intelligent

[14]. J. Jain, P. Panchal, N. Suryawanshi and A. P. M. A. S. Shinde, "Sentiment Analysis Using Supervised Machine Learning," Imperial Journal of Interdisciplinary Research (IJIR), vol. 2, no. 6, 2016.R. Rajput, "Review of Sentimental Analysis Methods using Lexicon

[15]. R. Rajput, "Review of Sentimental Analysis Methods using LexiconBased Approach," International Journal of Computer Science and MobileComputing, vol. 5, no. 2, pp. 159-166, 2016.

[16]. R. Wahome, "This Is How Twitter Sees the World: Sentiment AnalysisPart One," 2018. [Online]. Available: https://towardsdatascience.com/the-real-world-as-seen-on-twittersentiment-analysis-part-one-5ac2d06b63fb. [Accessed 25 february2019].

[17]. R. Rajput, "Review of Sentimental Analysis Methods using LexiconBased Approach," International Journal of Computer Science and MobileComputing, vol. 5, no. 2, pp. 159-166, 2016.

[18]. N. Federico , A. Carlo, F. Capeci and M. Cuadros , "Sentiment Analysison Social Media," IEEE/ACM International Conference on Advances inSocial Networks Analysis and Mining, pp. 919-926, 2012.