Université Saad DAHLAB - Blida 1

Faculté des sciences

Département d'Informatique

Mémoire présenté par :

BENALI Ahmed Chouaib

Ghebriout Mohamed

Pour l'obtention du diplôme de Master

Filière : Informatique

Spécialité : Traitement Automatique de la Langue

Ingénierie des Logicielles

Sujet:

Vers une détection des sentiments et des phrases subjectives dans les réseaux sociaux

Soutenu le : 04-10-2021 devant le jury composé de:

Mme.Zahra	Université Blida 1	Présidente
Mme.Berramdane	Université Blida 1	Examinatrice
Mme.Mezzi	Université Blida 1	Promotrice

Résumé

L’évolution des technologies numériques a transformé le monde avec de nouveaux moyens de communications tels que les réseaux sociaux. Les plateformes des réseaux sociaux reposent sur des données textuelles produites et consommées par les utilisateurs et l’analyse de ces données est dantesque par des moyens traditionnels qui ne peuvent s’adapter au volume grandissant et à la variété. Les nouvelles innovations telles que l'apprentissage automatique sont conçues pour aider en transformant des données textuelles en information et profit. L’analyse des sentiments est l’une de ces applications les plus intéressantes notamment quand il s’agit de l’extraction et analyse des opinions des utilisateurs.

Notre ressource de donnée dans ce travail est Twitter, l’objectif principal est de définir et de mettre en œuvre des moyens pour l’analyse des sentiments de données textuelles, nous avons choisi pour cette tâche un dataset lié au Covid19. Notre travail est basé sur des techniques de traitement automatique du langage et des techniques d’apprentissage automatique comme : SVM,Random Forest,Logistic Regression, XGBoost. Les tests effectués ont montré que la combinaison Random Forest avec la représentation Bag of words a donné les meilleurs résultats avec un F1-score de 79%. A partir de là, nous avons réalisé une application web qui implémente cette combinaison pour atteindre nos objectifs initiaux.

Mots clés : Analyse des Sentiments, Analyse d’Opinion, Apprentissage automatique.

Abstract

The evolution of digital technologies has transformed the world with new communication channels such as social networks. Social networking platforms rely on textual data produced and consumed by users and the analysis of this data is daunting by traditional means that cannot keep up with the growing volume and variety. New innovations such as machine learning are designed to help by transforming textual data into information and profit. Sentiment analysis is one of the most interesting applications, especially when it comes to extracting and analysing users' opinions.

Our data resource in this work is Twitter, the main objective is to define and implement means for sentiment analysis of textual data, we have chosen for this task a dataset related to Covid19. Our work is based on Natural Language Processing techniques and Machine learning algorithms such as: SVM, Random Forest, Logistic Regression, XGBoost. The tests that we have performed showed that the combination of Random Forest together with Bag of word model representation has shown the best results with an F1-score of 79%. Based on those results, we have developed a web application so as to meet our initial requirements.

Keywords: Sentiment Analysis, Opinion Analysis, Machine Learning.

ملخص

لقد أدى تطور التقنيات الرقمية إلى تحويل العالم من خلال قنوات اتصال جديدة مثل الشبكات الاجتماعية.

تعتمد منصات الشبكات الاجتماعية على البيانات النصية التي ينتجها المستخدمون يستهلكونها ، ويعد تحليل هذه البيانات أمرًا شاقًا بالوسائل التقليدية التي لا يمكنها مواكبة الحجم والتنوع المتزايد. تم تصميم الابتكارات الجديدة مثل التعلم الآلي للمساعدة عن طريق تحويل البيانات النصية إلى معلومات وربح. يعد تحليل المشاعر أحد أكثر التطبيقات إثارة للاهتمام ، خاصة عندما يتعلق الأمر باستخراج وتحليل آراء المستخدمين.

مصدر البيانات الخاص بنا في هذا العمل هو Twitter ، والهدف الرئيسي هو تحديد وتنفيذ وسائل لتحليل المشاعر للبيانات النصية ، وقد اخترنا لهذه المهمة مجموعة بيانات متعلقة بـ Covid19. يعتمد عملنا على تقنيات معالجة اللغة الطبيعية وخوارزميات التعلم الآلي مثل: Random Forest ، SVM ، الانحدار اللوجستي ، XGBoost. أظهرت الاختبارات التي أجريناها أن الجمع بين Random Forest مع تمثيل Bag of Word Model قد أظهر أفضل النتائج مع درجة F1 تبلغ 79٪. بناءً على هذه النتائج ، قمنا بتطوير تطبيق ويب لتلبية متطلباتنا الأولية.

الكلمات الرئيسية: تحليل المشاعر ، تحليل الرأي ، التعلم الآلي.

Dédicaces

Dédicace de Chouaib:

Je dédie ce travail a toute ma famille et à tous ceux qui ont cru en moi jusqu'au au bout, ce travail n'aurait pu aboutir sans leur soutien et je dédie ce travail a ma fiancé Anfel qui m'a toujours encouragé et soutenu dans tout ce que j'entreprends, elle est le cœur de ma vie et ma partenaire de toujours.

Je termine cette dédicace a mes frères, ma maman et mon papa et spécialement à Mohamed qu'on appelle chaleureusement "Biyou" dans la famille.

Dédicace de Mohamed:

A mes parents, mon frère et mes sœurs , notre promotrice Melyara Mezzi et ceux qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours. A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.

Remerciement

Tout d'abord, Nous remercions « الله » qui nous a guidés sur le chemin droit tout au long du travail et de nous avoir donné la volonté et le courage d'achever ce modeste travail dans de bonnes conditions.

Au terme de ce travail, nous tenons à remercier chaleureusement et respectivement tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet de fin d'études, et tout particulièrement Dr.Melyara Mezzi qui a su nous accompagner et nous guider durant notre travail sur notre mémoire et pour la confiance qu'elle nous a attribué.

Nous n'oublions pas nos parents et toutes les familles Ghebriout et Benali.

Nos remerciements vont enfin à tous nos proches amis qui nous ont toujours encouragés au cours de la réalisation de notre projet.

Merci à toutes et à tous.

Table des matières

Liste des figures

Liste des tableaux

Liste d’acronymes

NLP : Natural Language Processing

EDA : L'analyse exploratoire des données .

ML: Machine learning

CSV : Comma Separated Values

OMS : Organisation mondiale de la santé

WFH: Work from Home

OL : Online Learning.

RT : Retweet

POS: Part of Speech Tagging (Balisage des parties du discours)

PMI : Pointwise Mutual Information (Information mutuelle de type Pointwise)

LSTM : Long short-term memory (Mémoire à court terme)

ANN : Artificial Neural networks (Réseaux neuronaux artificiels)

LR : Logistique régression

RF : Random Forest

SVM: Support Vector Machine

XGBoost : eXtreme Gradient Boosting

API: Application Programming Interface

BOW: Bag Of Words

CBOW: Common Bag Of Words

TF : Terme de Fréquence

IDF : Inverse Terme de Fréquence

NumPy : Numerical Python

Introduction Générale

1.Contexte global

Les utilisateurs du monde entier expriment leurs opinions sur les réseaux sociaux tels que Facebook et Twitter et ainsi le besoin de connaître le besoins des utilisateurs s'est fait ressentir pour les entreprises et plus particulièrement leur opinion sur un sujet précis.

L'utilisation des technologies récentes d'analyse textuelles facilite l'analyse pour extraire des connaissances tout en étant faite de manière automatique,ce processus s'appelle l'Analyse des sentiments.

L'analyse des sentiments est une branche de l'Informatique,elle associe les techniques et connaissances théoriques sur l'extraction,traitement et résultats des opinions exprimées à travers du texte entre autres, de nombreuses applications et résultats peuvent être réalisé sur différents domaines comme l'économie,la politique ou les études sociales.

Un exemple récent et très parlant ont été les campagnes électorales de Donald Trump et d' Emmanuel Macron,les deux équipes ayant fait appels à des cabinets de consulting (Cambridge Analytica pour l'équipe de Trump et Liegy Muller Pons pour l'équipe de Macron) qui utilisent l'analyse des sentiments pour prédire et analyser le comportement de l'électeur moyen susceptible de voter pour lesdits candidats,ce qui a d'ailleurs conduit au scandale de Cambridge Analytica en 2017 et qui a forcé Facebook a revoir toute sa politique en matière de gestion des données utilisateurs et leur exploitation à des fins commerciales.

2.Problématique

L'analyse des sentiments a attiré l'attention des chercheurs récemment pour extraire et analyser les émotions exprimées directement ou indirectement par les humains et plus particulièrement sur du texte comme sur les réseaux sociaux.

Twitter est devenu une place incontournable pour miner des données pour prétendre faire de l'analyse des sentiments.

Avec l'arrivée du virus hautement contagieux Covid_19 et la situation sanitaire qui a engendré une pandémie, de nombreux couvre-feux strictes ont été imposé dans le monde

entier et qui sont toujours en vigueur dans certaines parties a l'heure actuelle pour contenir la propagation du virus et ralentir sa circulation,ceci a contraint des millions de personnes à rester chez eux ce qui a conduit à une plus forte utilisation des réseaux sociaux,cette forte utilisation a permis aux populations pour exprimer leur ressenti vis-à-vis de cette période,que cela soit des sentiments positifs ou négatifs.

Accumuler et étudier ces tweets va être d'un grand bénéfice pour aider à détecter les vraies émotions ressentis durant cette période au sujet de la pandémie et du Covid 19 en général dans le monde entier,de même que des études physiologiques sont entrain d'être faites sur les populations atteintes par ce virus,une étude qui permettraient d'analyser les émotions permettrait de comprendre l'impact psychologique que cette pandémie a fait subir sur les populations touchées.

Un autre problème majeur auquel nous sommes confrontés lors du traitement des données Twitter c'est que d'une part, les dialectes sont associés à aucun forme d'écriture normalisée et contiennent du bruit, des fautes d'orthographe, des abréviations,des répétitions, et des mots qui ne suivent aucune règle grammaticale.

Un autre problème dans l'analyse des sentiments c'est le problème du classement de la polarité (Positive, Négative, Neutre) à partir de données textuelles à l'échelle Web qui est une tâche très difficile et coûteuse en raison de la grande quantité de données bruitées.

3.Objectifs de l'étude

L'objectif de cette étude est de présenter une approche pour l'exploration d'un phénomène social (la pandémie de covid_19) en utilisant l'analyse des sentiments et des techniques du traitement automatique de la langue.

Le but est de développer une approche grâce à l'apprentissage automatique (Machine Learning) de détection des sentiments sur 2 classes: positives ou négatives,de ses applications et de développer notre modèle en tant qu'une contribution à la problématique de l'analyse des sentiments.

4.Organisation du Mémoire

Après cette introduction générale, le reste de notre travail est structuré comme suit:

Le premier chapitre est consacré au domaine de l'analyse des sentiments, nous présenterons les différents niveaux d'analyse des sentiments,leurs avantages et inconvénients,les outils nécessaires pour en faire.

Par la suite, le 2eme chapitre se porte sur l'étude de 3 papiers de recherches qui nous ont inspirées pour la construction de notre approche ainsi que d'une étude comparative entre eux pour retenir notre approche finale.

Par ailleurs, le troisième chapitre portera sur la construction du modèle d'apprentissage automatique ainsi qu'une petite partie dédiée à l'analyse exploratoire des données (EDA) qui concerne le dataset utilisé ainsi que toute la démarche.

Ensuite,le quatrième et dernier chapitre abordera sur l'implémentation finale de notre solution ainsi que l'interface qui a été créée pour afficher nos résultats.

Finalement, nous clôturons ce mémoire par une conclusion générale ainsi que les limitations de l'étude et enfin par les perspectives prochaine pour améliorer l'étude.

Chapitre I :

Traitement du Langage Naturel et l'Analyse des Sentiments

1. Introduction

Dans ce chapitre nous allons expliquer et exposer le Traitement du Langage Naturel et aborder le domaine de l'Analyse des Sentiments en présentant ses différents types, niveaux.

2. Traitement Automatique du Langage Naturel

C'est une branche de l'informatique et plus précisément un sous domaine de l'intelligence artificielle qui consiste à donner aux ordinateurs la capacité de comprendre du texte ainsi que la parole de la même manière que les humains arrivent à le faire.

2.1 Définition

Le traitement automatique du langage naturel est un ensemble de concept théorique et de techniques pour analyser automatiquement du texte à un ou plusieurs niveaux d'analyse linguistique dans le but d'effectuer des traitements [1].

Les textes peuvent être de n'importe quelle langue,mode,genre comme ils peuvent être oraux ou écrits pour peu qu'ils soient des langages humains.

2.2 Objectif

L'objectif est d'accomplir un traitement du langage semblable à l'être humain pour "comprendre" le contenu avec les nuances du langage pour en tirer des connaissances,classer ou organiser lesdites informations.

2.3 Niveaux de Traitement du langage naturel

Il y'a différents niveaux de traitement du langage qui peuvent être décrits comme suits:

- Phonologie: C'est l'étude de l'organisation des sons au seins des différentes langues naturelles et on peut utiliser 3 règles dans l'analyse phonologique:

- ❖ Les règles phonétiques: pour les sons dans les mots.
- ❖ les règles phonémiques: pour les variations de prononciation lorsque les mots sont parlés ensemble.
- ❖ les règles prosodiques: pour la fluctuation de la tension et de l'intonation à travers une phrase.

- Morphologie: Ici on traite au niveau des mots qu'on appelle Morphème qui sont les plus petites unités,on peut à ce niveau découper un mot en préfixe,racine et enfin suffixe.
- Lexique: C'est l'interprétation du sens des mots individuels pour faire de l'étiquetage.
- Syntaxe: C'est l'analyse au niveau de la phrase pour découvrir la grammaire ou la structure.
- Sémantique: Ici on se concentre sur l'analyse des mots au niveau de la phrase pour découvrir le sens.

2.4 Applications du Traitement du langage naturel

Voici les applications fréquentes qui peuvent en découler:

- ★ Recherche d'Information: C'est le domaine qui étudie la manière de retrouver des informations dans un corpus qui est composé de documents qui sont stockées sur des bases de données, ces dernières peuvent être relationnelles ou non structurées.
- ★ Traduction automatique: c'est une traduction automatique d'une langue à une autre.
- ★ Système de dialogue: c'est un système d'interaction homme-machine,le type d'interaction peut être de manière textuelle,audio ou visuelle.
- ★ Intelligence Marketing: Les responsables marketing peuvent utiliser le traitement du langage naturel pour mieux comprendre leurs clients et utiliser ces informations pour créer des stratégies efficaces. Ceci permet d'analyser des sujets, des mots-clés et d'utiliser correctement des données non structurées. Il peut également être utilisé pour identifier les points sensibles des clients et garder un œil sur les concurrents de l'entreprise [2].

- ★ Publicité ciblée: La génération de prospects reste au cœur des préoccupations des entreprises. C'est la principale raison pour laquelle elles souhaitent toucher un maximum de personnes. Le traitement du langage naturel est une ressource extraordinaire pour placer la bonne publicité, au bon endroit et au bon moment. Cela se fait par l'analyse des mots clés, des habitudes de navigation des utilisateurs sur l'internet, des emails, ou des plateformes des réseaux sociaux. Les outils d'exploration de texte sont utilisés pour effectuer ces tâches.[2]
- ★ Assistance Vocale: les assistants vocaux utilisent la reconnaissance vocale, la compréhension et le traitement du langage naturel pour comprendre les commandes verbales de l'utilisateur et exécuter les actions en conséquence. Depuis leur introduction jusqu'à aujourd'hui, ils se sont transformés en un gadget très fiable, on peut citer des outils comme Alexa,Google Assistant ou Siri de Apple.

Le traitement qui nous intéresse le plus est bien évidemment l'analyse des sentiments.

3. Analyse des sentiments

3.1 Définition

C'est un ensemble de techniques et de principe théorique et qui se focalise sur l'extraction,traitement des sentiments liées au texte [3], ce qui permet par exemple à des entreprises de connaître en détail a partir des commentaires des clients leur avis sur un produit,par ailleurs les réseaux sociaux ont commencé à l'adopter avec le système des émoji.

On peut utiliser l'exemple suivant pour introduire le problème:

"(1) J'ai acheté un iPhone il y'a quelque jours de cela,(2) c'était un si beau téléphone, (3) l'écran était vraiment cool, (4) la qualité de son était clair aussi, (5) malgré que la longévité de la batterie n'était pas fameuse,c'était suffisant pour moi.(6) Par contre,ma mère était fâché de moi car je ne lui ai pas dit que je comptais l'acheter,(7) elle trouve que le téléphone était trop chère et voulait que je le rends au magasin. "

La question qui se pose est: qu'est qu'on veut récupérer comme info de ce texte ? ce qu'on peut noter c'est que:

- 2,3,4 sont des expressions positives.
- 5,6,7 sont des expressions négatives
- Toutes les opinions ont des sujets,objets le 2 aborde sur l'iPhone en entier tandis que le 3,4,5 est a propos de l'écran,qualité de son,longévité de la batterie.

Par ailleurs on peut noter les sources des opinions:

- 2,3,4,5 est l'auteur de ce texte.
- 6,7 c'est la mère de l'auteur.

Les opinions peuvent être exprimées sur tout,on utilise le terme "objet" pour dénoter l'entité sur lequel l'opinion a été réalisée, un objet peut avoir un ensemble d'attributs et composants,chaque composant peut être découpé en sous composant de tel sorte a avoir une structure arborescente.

Dans la pratique,un tel niveau de détail est impossible à réaliser dans la mesure où l'analyse devient plus détaillé mais aussi plus compliqué à gérer,du coup pour une nécessité de simplicité on appelle "feature" la compression des attributs et composants, l'objet en lui meme peut etre vu comme un "feature" spécial.

3.2 Types d'analyse

Il existe différents types modèles de classification:

Au niveau du document qu'on appelle la classification des sentiments: c'est le fait de classifier un document qui contient des opinions comme étant de classe positif ou négatif,on suppose que le document est déjà reconnu comme contenant des opinions mais chaque phrase ne peut être reconnu comme contenant des opinions, le but de classifier des phrases contenant des opinions ou non est appelée **une classification subjective et c'est au niveau des phrases.**

3.2.1: Classification des sentiments au niveau du document:

La classification s'opère sur les algorithmes supervisés et non supervisés.

3.2.1.1:Classification basé sur les algoritmes supervisées:

La classification des sentiments peut être formalisée comme un problème de classification sur 2 classes (positif/négatif) et dans ce cas les mots d'opinion comme : joli/mauvais/excellent/… sont importants à déterminer.

Naive Bayes & SVM sont les principaux algorithmes utilisés dans cet objectif et qui ont été démontrés comme étant les plus adaptées à cette classification.

Quelque exemples de caractéristiques (features) utilisées:

- Les termes et leur fréquence: les mots individuels et leur fréquence est un important caractéristique et dans certains cas la position dans le mot compte aussi, le TF-IDF peut être appliqué dans cette logique.
- Tags de partie de discours(Part of Speech Tags): les adjectifs sont d'importants indicateurs de subjectivité et d'opinion.
- Les mots/phrases d'opinion : les mots communément utilisés pour exprimer des sentiments positifs ou négatifs comme: "joli,mauvais" ou verbes comme "haïr ou aimer" ou des idiomes.
- Négation: ce sont des mots qui peuvent changer l'orientation mais ça ne veut pas dire que chaque occurrence d'une négation est une inversion de la polarité comme dans cet exemple : "Non seulement"; le non dans cet exemple n'inverse pas la polarité.

A part la classification des opinions en 2 classes (positifs/négatifs), d'autres chercheurs se sont concentré sur la prédiction d'une note d'évaluation comme le fait Uber par exemple (1-5 étoiles), le problème ici est formulée comme étant un problème de régression vu que les notes sont des nombres continues.

Par ailleurs une observation a été faite que la classification des sentiments est sensiblement sujette au domaine dans lequel la prédiction est faite et la raison pour cela est que les mots utilisés pour exprimer des opinions peuvent changer d'un domaine à un autre,par exemple le mot imprévisible est un mot plutôt négatif quand on aborde la question de la direction d'une voiture (direction imprévisible) mais devient positif dans une critique de film (intrigue imprévisible).

3.2.1.2:Classification basé sur les algoritmes non supervisées:

Un exemple qui explique comment on peut utiliser un algorithme non supervisé pour la classification des sentiment, Le chercheur P.Turney[4] et son équipe ont développé un algorithme qui peut classifier en utilisant des phrases syntaxiques fixes qui peuvent être utilisé pour exprimer des opinions et qui consiste en 3 étapes:

1. Extraction des phrases qui contiennent des adjectifs et des adverbes qui sont des indicateurs de subjectivité,l'algorithme se charge de les extraire avec le contexte autour du mot selon les règles décrites dans le tableau(1) suivant:

Premier mot	Deuxième mot	Troisième mot (non extrait)
1. JJ	NN or NNS	anything
2. RB,RBR, or RBS	JJ	not NN nor NNS
3. JJ	JJ	not NN nor NNS
4. NN or NNS	JJ	not NN nor NNS
5. RB, RBR , or RBS	VB,VBD,VBN, or VBG	anything

Tableau 1 : Modèles d'étiquetage de partie du discours(POS) pour l'extraction de phrases de deux mots .

Par exemple la ligne 2,le premier mot est soit un adverbe et le 2eme mot un adjectif et le 3eme mot ne peut être un nom pour être extrait.

2. Estimer l'orientation des phrases extraites en utilisant la mesure PMI (Pointwise Mutual Information) qui peut être calculé par ceci:

$PMI(term1, term2) = log_2(Pr(term1 \wedge term2) / Pr(term1)Pr(term2))$

3. L'algorithme calcule la moyenne des opinions de toutes les phrases dans le texte et classe comme étant recommandé si la moyenne est positive sinon non recommandée.

3.2.2: Classification des sentiments au niveau de la phrase:

Pour une phrase donnée,deux tâches sont réalisé:

1. Classification subjective: Déterminer si la phrase est une phrase subjective ou objective.
2. Classification des sentiments au niveau de la phrase: Si la phrase est subjective déterminer si elle exprime une opinion positive ou négative.

La plupart des approches étudient le problème dans les 2 phases en même temps,les deux problèmes étant des problèmes de classification,le plus approprié reste les algorithmes supervisé

Un des problèmes quand on applique des algorithmes supervisés est l'effort manuel requis pour labelliser un corpus,deux chercheurs dénommé E. Riloff et J. Wiebe) [5] se sont penché sur la question et ont proposé une approche automatique pour cela,voici l'approche qui a été proposé:

1. l'Algorithme utilise 2 outils de classification : HP-Subj et HP-Obj,ces outils utilisent une liste lexicale composée de n-grams et qui contiennent de bons indicateurs subjectifs ou objectifs.
2. HP-Subj classe une phrase comme étant subjective si elle contient 2 ou plus d'indicateurs subjectifs.
3. HP-Obj classe une phrase comme étant objective si elle contient 2 ou plus d'indicateurs objectifs
4. Les phrases extraites sont directement ajoutées comme des caractéristiques pour des algorithmes de machine learning pour apprendre plus de modèles pour se renforcer,ce qui va permettre de

mieux accroître la précision des outils a chaque itération comme une boucle fermée.

Une critique de cet algorithme est que le auteurs assument une hypothèse qui veut qu'un seul sujet peut exprimer une opinion en particulier,celle-ci est mise à mal dès que la phrase est composée de plusieurs auteurs et de plusieurs opinions.

3.2.3: Lexique d'opinion

Le lexique d'opinion est un lexique où sont générés tous les mots d'opinions ainsi que les phrases ou idioms,chaque mot peut être classé en 2 catégorie: les types de base ainsi que les mots de comparaison.

Les types de base ne sont que les mots d'opinion mentionnés auparavant,les mots de comparaison ne reflètent pas une opinion mais servent à comparer 2 avis différents comme dans cet exemple: "La voiture X est meilleure que la voiture Y".

Il existe 3 approches pour collecter le lexique: manuel,via un dictionnaire ou une approche basée sur un corpus.

1. L'approche manuel: c'est une approche qui demande beaucoup de temps et n'est donc utilisée qu'en combinant avec une des 2 approches restantes.
2. L'approche dictionnaire: une des approches les plus simple est d'utiliser un ensemble de mots racines ainsi qu'un dictionnaire en ligne comme WordNet,on collecte manuellement les mots racines qui ont une polarité connu pour ainsi les combiner avec WordNet en recherchant tous les antonymes et synonymes,cette nouvelle liste servira comme une nouvelle entrée pour WordNet,etc..

   Cette boucle s'arrête lorsque tous les antonymes et synonymes ont été trouvés,une inspection manuelle peut être décidée plus tard pour corriger d'éventuelles erreurs.

   Une critique de cette approche est qu' elle est dépendante du domaine,le dictionnaire n'aura pas de résultat probant si on change de domaine.

3. L’approche Corpus: cette approche comble le problème posé par l’approche dictionnaire.

Un exemple réalisé par les chercheurs (Vasileios Hatzivassiloglou et Kathleen R. McKeown) [6] consiste de la manière suivante:

On a une liste de mots racine qui sont des adjectives et un ensemble de règle linguistique pour identifier d’autres adjectives ainsi que leur polarité.

Par exemple le mot “et” peut être associé à une règle qui veut que tous les adjectifs en conjoint ont généralement la même polarité comme dans cet exemple:

“Cette voiture est magnifique ET spacieuse ”

Si le mot racine “magnifique” est connu comme étant un mot positif,on peut inférer grâce à cette approche que le mot “spacieuse” est positif aussi,cette approche peut être appliquée à tous les connecteurs connus et il est connu sous le nom de Cohérence du Sentiment.

L'apprentissage est réalisé grâce à un modèle log linéaire et il est appliqué à un large corpus pour déterminer si 2 adjectifs conjoints ont la même polarité ou non,ce qui forme un graphe,on applique un algorithme de classification pour produire 2 clusters : les mots positifs et négatifs.

3.3 Processus général de l'Analyse des Sentiments

La figure ci-dessous résume le processus général d'une analyse de sentiment.

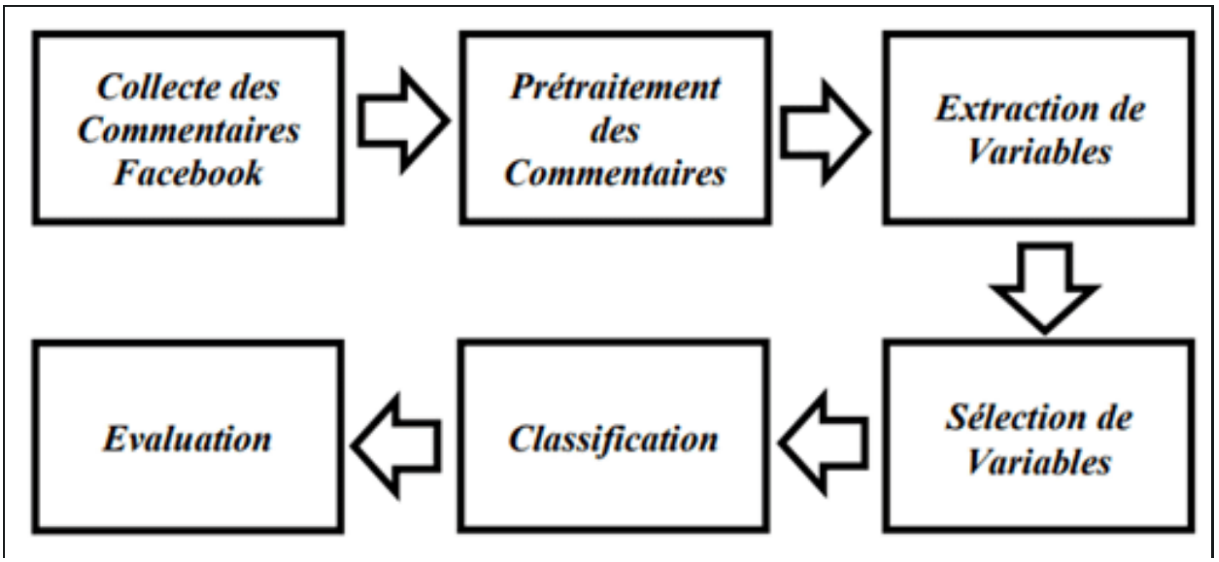

Figure 1 : Processus général de l'analyse des sentiments

3.4 L'Analyse des Sentiments sur les Réseaux Sociaux

L'analyse des sentiments sur les réseaux sociaux est un domaine récent d'activité car elle aide les entreprises pour automatiser leur département marketing et tout ce qui a trait aux relations publique,si par exemple un client est sur le point de résilier et fait part de son mécontentement sur les canaux de l'entreprise via Instagram,Facebook ou Twitter,les modeles de l'analyse des sentiments peuvent aider à prédire à l'avance, voir gérer la crise via des chatbots par exemple.

Pour les besoins de notre recherche,nous avons opté pour Twitter qui est un réseau social ou les utilisateurs s'échangent des messages appelées Tweet limité à 280 caractères, le nombre d'utilisateurs actifs a l'heure actuelle se situe vers les 206 millions.

Twitter présente des avantages non négligeable comme sa réactivité et la brièveté de ses messages,ce qui en fait pour les chercheurs et les entreprises une source importante pour miner les opinions des utilisateurs sur leur produits ou sur une thématique en particulier,les récents événement avec le Covid 19 ont démontré que Twitter est un réseau social actif et essentiel pour suivre l'actualité en temps réel.

Le graphe ci-dessous [7] montre que le taux de téléchargement journalier que compte Twitter au cœur de la pandémie qui a connu un pic vers l'année 2020 et qui coïncide avec les couvre-feu instauré dans le monde entier et l'arrêt momentané de plusieurs entreprises pour freiner la propagation de l'épidémie.

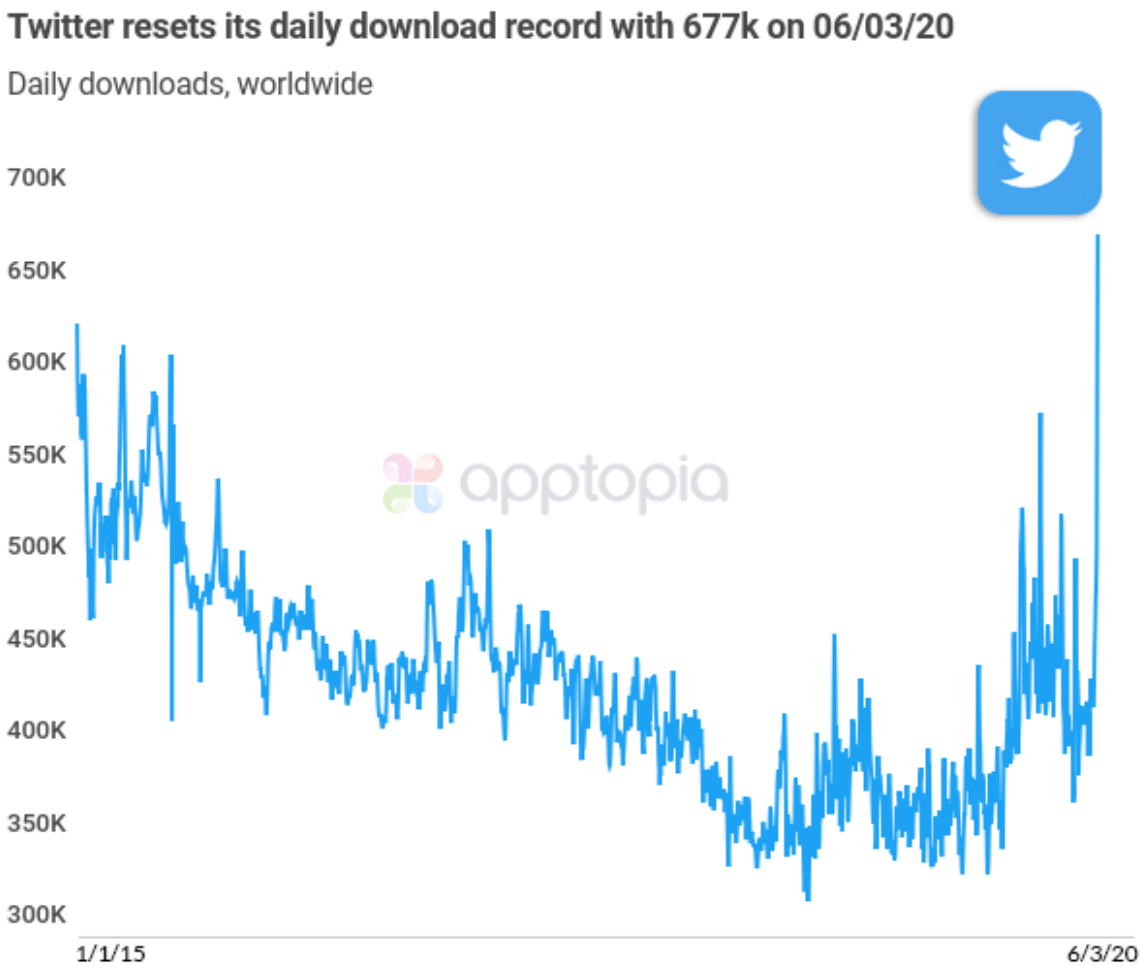

Figure 2 : Le taux de téléchargement journalier de l'application Twitter

3.5 Conclusion

Ce chapitre présente une définition clair du traitement automatique de la langue ainsi que l'analyse des sentiments et ses multiples applications dans le monde réel agrémenté d'exemple pour expliquer différentes techniques,c'est un sujet d'actualité et particulièrement en ces temps trouble de Pandémie pour mieux analyser les sentiments des utilisateurs vis à vis des enjeux actuels.

Dans le prochain nous allons aborder la conception théorique de notre application finale, ce chapitre sera composé de l'architecture globale ainsi que tous les aspects théoriques nécessaires à l'implémentation de notre solution finale.

Chapitre II :Travaux Connexes

Introduction

Pour implémenter notre solution finale, une étude comparative entre les différentes approches semble nécessaires,que ce soit en termes d'algorithmes choisis,en termes de performances résultantes,etc..

Dans ce chapitre,nous allons présenter 3 articles connexes à notre travail ainsi qu'une étude comparative sur plusieurs critères qui nous semblent essentiels pour la recherche et création d'une solution finale.

D'après Ohio State News [8] ,il y'a 87.000 papiers scientifiques en rapport avec le Covid19 depuis le début de la pandémie jusqu'à Février 2021 et d'après Dimensions [9] il y'aurait plus spécifiquement 6496 articles scientifiques qui lierait les deux thèmes Covid 19 et analyse des sentiments.

Le but final de cette évaluation est de se concentrer sur les aspects les plus importants qui seront pris en compte dans la réalisation de notre solution finale.

Ce chapitre sera divisé comme suit:(a modifier)

- Une petite description de l'article et sur les motivations de l'étude avec quelques informations clés.
- Une présentation de la méthodologie choisie
- Les résultats finaux avec les scores choisis par chaque étude.

I) Sentiment Identification in Covid-19 Specific Tweet:

1) **Description:** L'objectif de cet article est de déterminer l'impact du Coronavirus sur les émotions des personnes en analysant les sentiments décrits dans leur commentaires sur Twitter en utilisant des hashtags comme #COVID19 avec Twitter API, l'approche qui a été utilisé pour arriver aux sentiments des utilisateurs a été la suivante:
    - Récupération des tweets.
    - Pré-processing des tweets.
    - Construction d'un modèle Bag of Words.
    - Méthode de notation pour prédire la polarité des tweets.
    - Comparaison des différents algorithmes pour la classification des tweets.

Voici l'architecture système présentée dans l'article:

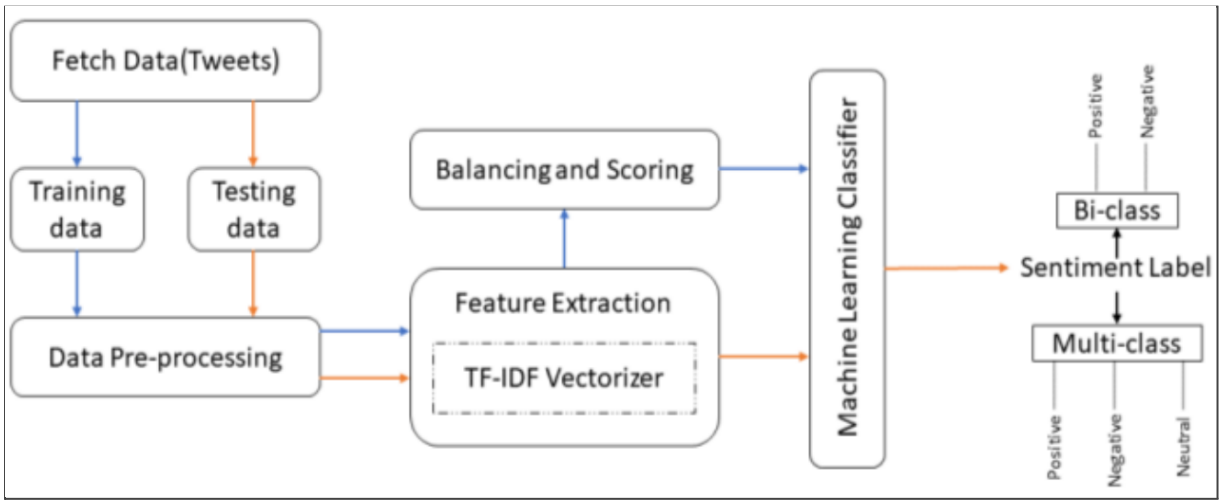

Figure 3 : Architecture système

3 datasets ont été utilisé à cet effet: Le premier contient les tweets qui ont été extrait avec le hashtag: #coronavirus, le 2eme dataset a été extrait avec le hashtag : #Covid19 et le 3eme dataset n'est que la combinaison des 2 premiers datasets.

Chaque dataset contient 10.000 tweets à l'exception du dernier qui contient 20.000 car il n'est que la combinaison des deux premiers dataset.

2) **Résultats**:

L'expérience a été réalisée suivant 3 scénarios:

1. Classification Binaire (Positive-Negative)
2. Classification sur 3 Classes (Positive-Neutre-Négative)
3. Cross-dataset Evaluation

Scénario 1 : Classification Binaire

La classification binaire a été réalisé en utilisant soit des unigrams,n-grammes pour les 3 datasets et voici les principales conclusions:

- Les algorithmes Decision Tree,SVM et Random Forest ont été meilleurs que les autres algorithmes avec une précision proche de 91% tandis que le score F1 avoisinait entre 85 et 86%.

Scénario 2: Classification Multiclasse (3 classes)

- Les algorithmes Decision Tree et SVM ont été meilleurs que les autres algorithmes.
- Decision Tree est meilleur avec le dataset 1
- SVM est meilleur avec le dataset 2 et 3
- Les deux algorithmes ont une précision proche de 88 à 89% tandis que le score F1 avoisinait entre 86 a 87%.

Dans cette étude il a été observé que pour tous les datasets,les modèles unigrams fonctionnent mieux que les modèles n-grammes dans les deux scénarios 1 et 2.

Scénario 3 : Cross Dataset Evaluation

Pour la classification cross dataset pour le scénario 1 (classification binaire) du dataset 1 vers le dataset 2,voici les principales conclusions:

- XGBoost et Logistic Regression ont été meilleurs que les autres algorithmes.
- La précision atteint en moyenne 80% quand les features unigrammes et bigrammes sont combinées.

Pour la classification cross dataset pour le scénario 1 (classification binaire) du dataset 2 vers le dataset 1,voici les principales conclusions:

- XGBoost a été le meilleur algorithme.
- La précision atteint en moyenne 82% quand les features unigrammes et bigrammes sont combinées

Pour la classification cross dataset pour le scénario 2 (classification multiclasse) du dataset 1 vers le dataset 2,voici les principales conclusions:

- Random Forest a été le meilleur algorithme.
- La précision atteint en moyenne 45%.

Pour la classification cross dataset pour le scénario 2 (classification multiclasse) du dataset 2 vers le dataset 1,voici les principales conclusions:

- Random Forest a été le meilleur algorithme.
- La précision atteint en moyenne 46% avec un feature unigramme.

NB: Pour le 3eme Scénario:

- XGBoost a été meilleur pour une classification binaire.
- Random Forest a été meilleur pour une classification multiclass

3) Résumée

- Le but a été la création d'un modèle pour prédire les sentiments exprimés par les utilisateurs sur Twitter à propos du Covid-19.
- Les datasets sont crée manuellement avec Twitter API a partir de 2 hashtags; "#COVID19" et" #coronavirus".
- Les résultats montrent que SVM et Decision Tree ont été les meilleurs algorithmes mais SVM a été plus robuste et consistant.
- Les résultats montrent aussi que la plupart des modèles performent mieux avec un model unigram et bigram quand c'est une classification binaire mais que dans le cas de la classification multiclasse, unigram a été meilleur.
- Les modèles ont atteint une précision de 93% dans le meilleur des cas.
- Pour l'évaluation cross-dataset,les algorithmes ont été meilleurs quand c'est une classification binaire et XGBoost été meilleur avec une précision de 82% alors que pour une classification multi classe, Random Forest été meilleur avec une précision de 46%.
- En comparaison, la classification binaire a été meilleure pour les 3 datasets avec une meilleure performance.

II) Tweets Sentiment Analysis during Covid-19 Pandemic

1. **Description**:

L'objectif de cet article est de découvrir l'impact du virus Covid19 sur les gens et d'utiliser des techniques de Machine Learning et plus particulièrement le clustering K-Means pour trouver des motifs qui aident à comprendre le point de vue des personnes sur le virus et la pandémie en général.

Les données sont générées en utilisant l'API Twitter à travers quatre ensembles de données:

1- Organisation mondiale de la santé (OMS) (3500 tweets) : Cet ensemble de données est collecté à partir de la timeline du compte de l'OMS sur Twitter et les tweets générés sont destinés à trouver les mots les plus fréquents qui sont utilisés dans la limite de 3500 tweets

2- Ministère de la Santé de Bahreïn (3500 tweets) : Même approche que pour le premier jeu de données, les mots les plus fréquents.

3- Jeu de données de tweets en anglais (34 attributs et 23.490 instances) : Les tweets sont générés à l'aide de l'API Twitter avec des mots/hashtags comme "COVID19" ou "coronavirus".

4- Ensemble de données de Tweets en arabe (34 attributs et 13.088 instances) : Ce jeu de données est généré en utilisant 2 mots clés en arabe : (كوفيد 19 ، كورونا), les attributs sont les mêmes que le jeu de données anglais, la différence est l'emplacement des tweets.

Les deux premiers ensembles de données sont utilisés pour obtenir plus d'informations sur les mots les plus fréquemment utilisés et les 2 derniers sont utilisés pour le clustering et les tests.

2. Résultats

Dataset 1

3 Catégories de Sentiments : Positif-Neutre-Négatif que l'on peut reprendre dans ce tableau

Tweet	Résultat	Pourcentage
Neutral	7.184	43.2%
Positive	4.870	29.3%
Négative	4.572	27.5%

Tableau 2: Le résultat avec les trois catégories

La figure suivante montre d'où proviennent les tweets avec leur sentiment :

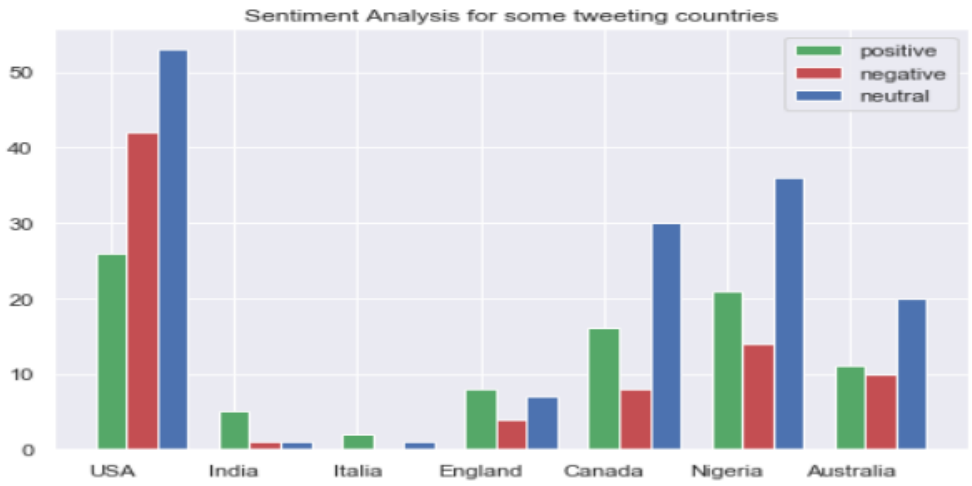

Figure 4 : Analyse des sentiments pour certains tweets des pays

Dataset2:

Les résultats sont présentés dans ce tableau avec les 3 mots les plus courants

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Mask	State	Pandémie	Confirmed	Want	Manipulate	Backed
Home	Great	Thank	Death	Looks	Refusing	Stake
Wear	Mask	Support	Deaths	Going	Scientist	Firm
Cluster 7	Cluster 8	Cluster 9	Cluster 10	Cluster 11	Cluster 13	Cluster 14
Died	Lockdown	Steps	Défends	Crisis	Class	Differenc e
Future	Vaccine	Returning	Président	Update	Middle	Spent
Economy	Crisis	Ships	Unproven	Thank	Lower	Imagine

Tableau 3 : Certains des mots communs à chaque groupe

Nous pouvons trouver certains motifs(patterns) comme les clusters 1, 3, 4, 7 et 10 mais avec des chevauchements comme pour les clusters 1 et 2 avec le mot Mask.

3ème Dataset: Le résultat est repris dans ce tableau mais nous pouvons voir que les performances du mini batch k-means sont faibles car il y a des mots répétés tels que “people”.

Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**	**Cluster 6**	**Cluster 7**	**Cluster 8**
Let	Positive	Imagine	Heading	Post	State	People	People
Right	China	Executive	Scientist	India	Virus	Security	New
Know	Deaths	Hydroxy chloroquine	Cashless	Poverty	People	Flaws	Daily

Tableau 4 : Certains des mots communs à chaque groupe

4ème Dataset : Jeu de données arabe K-means

Voici les résultats, il y a un motif dans le cluster 1 (nom des villes), 2(regarder les médias) ,3 (Religion), 4(Corona) , 8 (groupe de mots psychologiques).

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
المدينه	شاهد	رمضان	بفيروس	فيروس	العذاب	فيروس	مرشدك
الرياض	فديو	المسجد	حاله	كورونا	الأدنى	بعد	سريه
الدمام	تشوفه	الأقصى	اصابه	بعد	والبيان	مع	النفسي

Tableau 5 : Certains des mots communs à chaque cluster

4ème Dataset : Jeu de données arabe avec Mini batch k-means

Nous pouvons observer certains motifs comme le groupe 1 (noms), le groupe 4 (Corona) et le groupe 5 (noms de pays/villes).

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
قحط	كورونا	مرشدك	بفيروس	السعوديه	تشوفه
مرزوقه	فيروس	جايحه	اصابه	الإمارات	فديو
انا	بعد	مبادره	تسجيل	الرياض	شاهد

Tableau 6 : Certains des mots communs à chaque cluster

En comparant les 2 ensembles de données entre l'anglais et l'arabe, nous pouvons observer que l'ensemble de données anglaises était mieux groupé que l'arabe et plus raisonnable où la

majorité est regroupée dans un seul cluster (arabe), le K-Means a mieux performé que le mini batch k-means mais le temps pour le construire est significativement plus faible mais conservant la même performance pour l'ensemble de données arabes.

3. **Résumée**:
    - L'utilisation du clustering k-means pour trouver un motif qui identifie le sentiment des gens pendant la pandémie de COVID-19 .
    - La plupart des mots utilisés portent sur la pandémie COVID-19 dans les ensembles de données de l'OMS et du ministère de la santé du Bahreïn.
    - La plupart des tweets sont neutres aux États-Unis, en Australie, au Nigeria, au Canada et en Angleterre.
    - Cependant, l'Italie et l'Inde ont la majorité de tweets positifs. Cela montre que les habitants de l'Italie et de l'Inde sont plus optimistes que les autres .
    - En constate que le modèle k-means mini-batch a nécessité moins de temps pour être construit par rapport au modèle k-means avec une légère différence dans la performance
    - Finalement, l'absence de bibliothèques arabes dans Python et leur utilisation limitée dans R , il est extrêmement difficile de traiter le jeu de données en arabe.

III) Global Sentiment Analysis of COVID-19 Tweets over time

1. Description

Ce papier présente une analyse des sentiments sur Twitter liée au Coronavirus et dans les aspects de la vie de tous les jours. Ce qui nous intéresse, c'est les tweets relatifs aux deux événements: WFH: Work from Home et Online Learning.

Différents modèles ont été utilisés comme Long Short Term Memory (LSTM) et Artificial Neural Networks (ANN) pour la classification des sentiments.

Une Analyse Exploratoire des Données ou (EDA) a été faite pour un dataset pour avoir plus d'informations sur le nombre de cas confirmé par jour sur les pays les plus touchés par le covid et pour nous permettre d'avoir une comparaison dans le changement des sentiments avec le nombre de cas depuis le début de la pandémie jusqu'en Juin 2020.

A cause du couvre feu pour contenir l'extension du foyer épidémique, les personnes dans le monde entier ont été obligées de travailler depuis la maison pour certains métiers et les écoles/universités ont été sommées d'accélérer leur programme de mise en place de l'apprentissage en ligne.

Un dataset a été labellisé en utilisant un lexicon de sentiment nommé VADER. Ce dataset a été, ensuite, utilisé pour la classification des modèles LSTM et ANN.

Bien que l'analyse des sentiments des tweets peut refléter le sentiment général, cela n'explique pas l'impact actuel du virus. C'est pour cela que l'EDA est appliqué au dataset pour pouvoir comparer entre le sentiment publique et l'analyse EDA.

Le papier a réalisé une classification binaire et détection d'émotion: la peur et la confiance.

4 datasets ont été utilisé en tout:

- Dataset A: Coronavirus Tweets dataset:

165.116 Tweets récolté en utilisant le hashtag #coronavirus avec TwitterScraper depuis le 1er Janvier 2020 jusqu'au 29 Juin 2020.

Pour extraire les pays les plus touchées puisque la majorité des utilisateurs n'activent pas la localisation pour envoyer leur tweet ,la localisation a été extraite depuis leur profilé en utilisant Tweepy,dès que les coordonnées ont été extrait,ils ont utilisé une base de donnée "datahub.io" pour matcher la localisation qui a été ensuite assignée au tweet.

Pour obtenir le sentiment du tweet,VADER a été utilisé pour 3 classes: Positive,Négative et Neutre.

- Dataset B: Online Learning dataset:

40.756 tweets réc »olté en utilisant le hashtag #onlinelearning et #onlineclasses du 19 Février 2020 jusqu'au 26 Juin 2020,les sentiments et localisation ont été extrait de la même manière que le dataset A.

- Dataset C: Work from Home dataset:

41.349 tweets récolté en utilisant le hashtag #workfromhome et #WFH du 22 Février 2020 jusqu'au 3 Mai 2020,la même méthode que le dataset A a été utilisée pour la localisation et le sentiment.

- Dataset D: Covid19 Dataset

Obtenu de Kaggle,contient des infos sur les nombre confirmé de cas,morts,cas de rémissions chaque jour dans tout le globe depuis le 22 Janvier 2020 jusqu'au 22 Août 2020.

2. Résultats

A: Précision des modèles ML:

Dans le dataset Coronavirus en utilisant VADER:

- 84.5% en précision en utilisant LSTM
- 76% en précision en utilisant ANN

On peut observer dans la figure un gap entre le % des négatifs et sentiments positifs est plus grand entre Février et Mars.

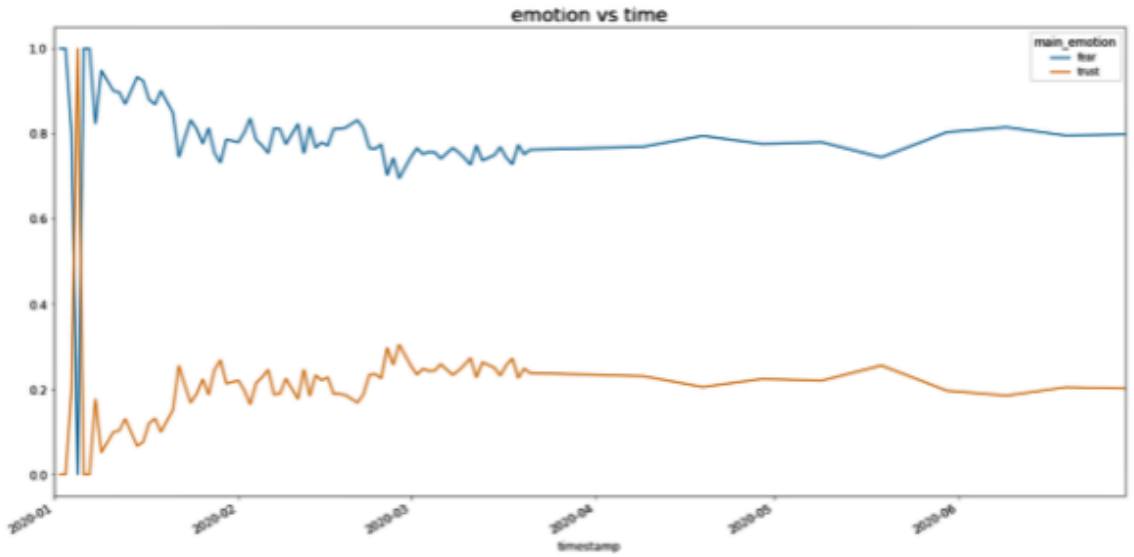

Figure 5 :Sentiment de peur et de confiance par rapport au temps

La figure illustre que la peur prédomine largement celle de la confiance durant la pandémie, les pays qui un meilleur % des sentiments positifs sont: Bangladesh,Pakistan,Mali,South Africa alors que Australian,Indie,Canada,USA,Turkey,UK et le Brésil sont les pays où la proportion est plus négative.

B: Les pays les plus touchés

1) USA

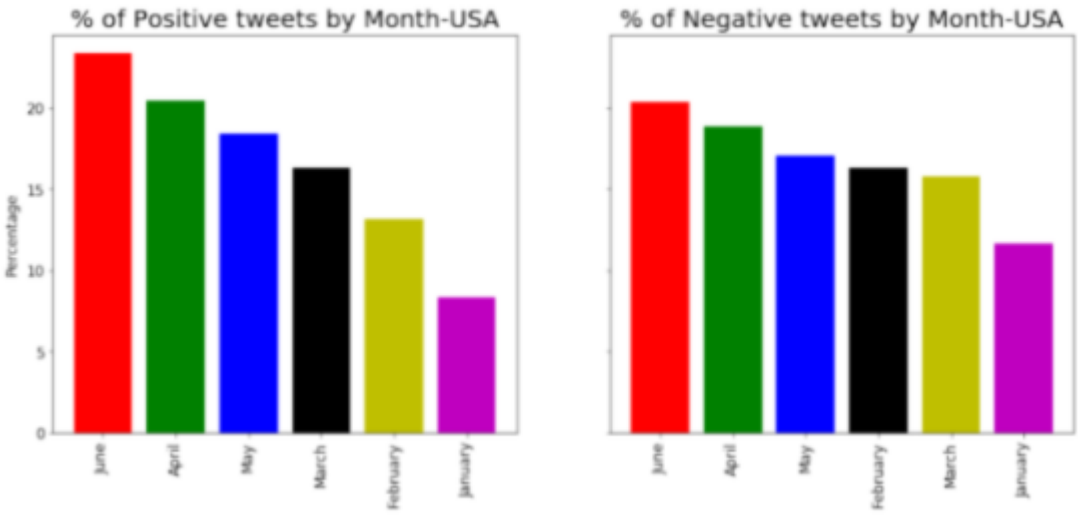

Figure 6 : Pourcentage de sentiments positifs et négatifs aux États-Unis par mois

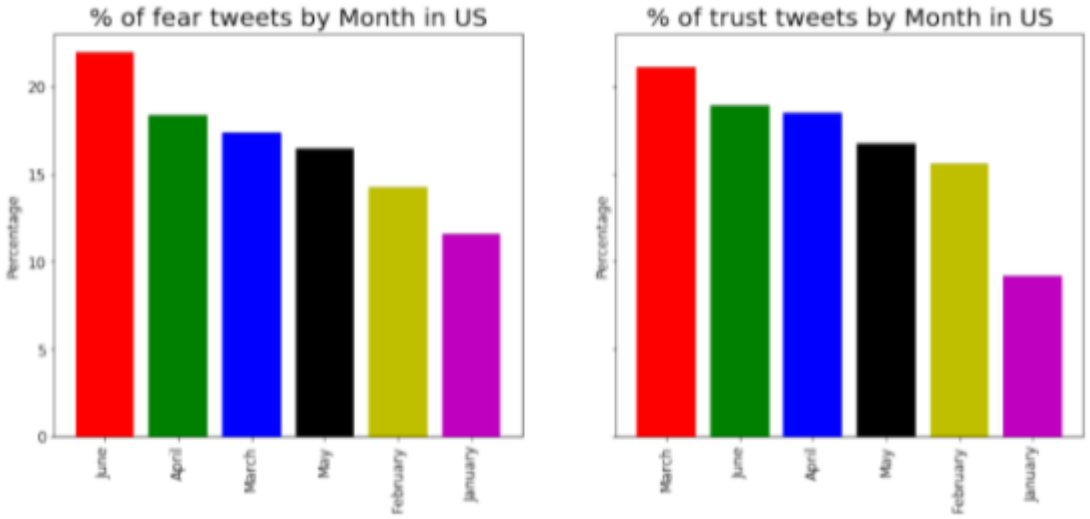

Figure 7 : Pourcentage d'émotions de peur et de confiance aux États-Unis par mois

2) Inde

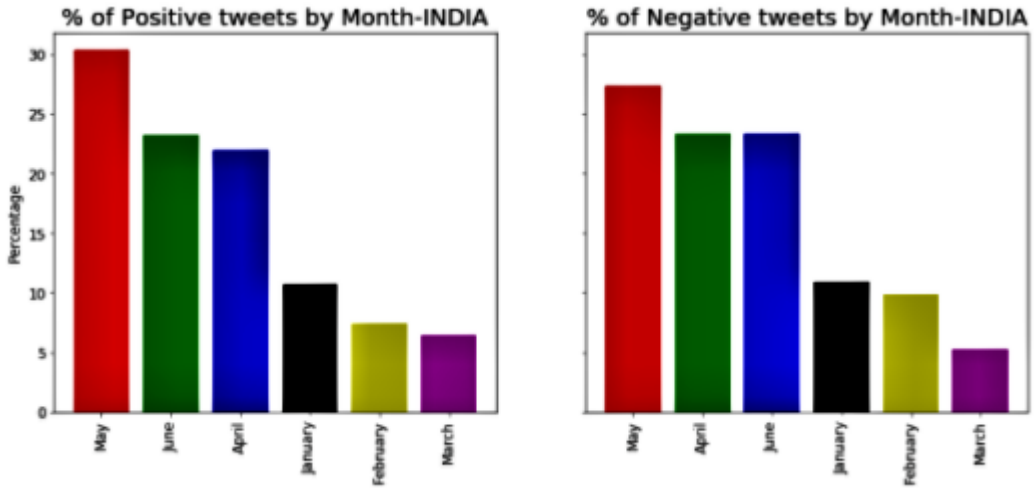

Figure 8 : Pourcentage de sentiments positifs et négatifs en Inde par mois

L'augmentation des tweets positifs et négatifs est probablement due à l'augmentation exponentielle au nombre des cas en Mai.

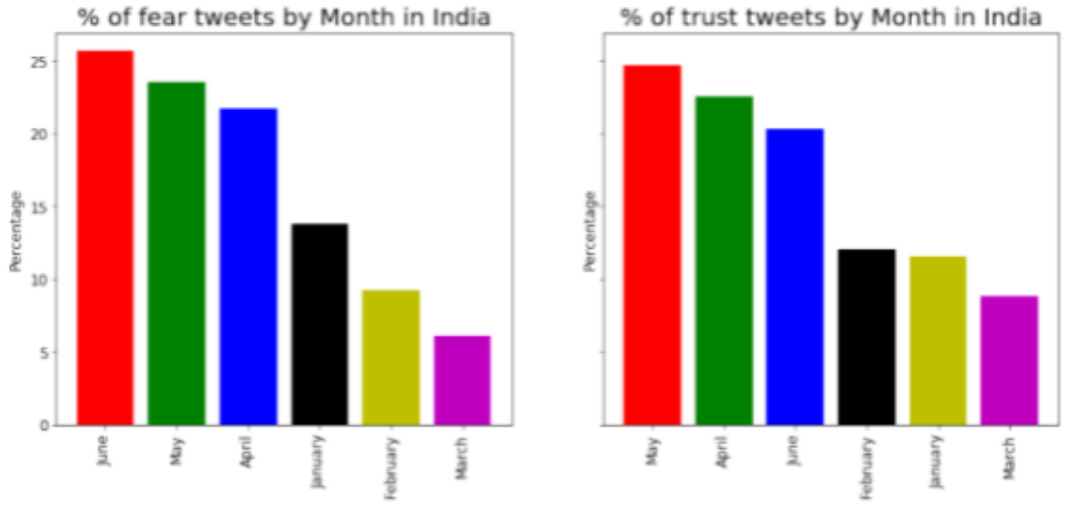

Figure 9 : Pourcentage de Confiance et de Peur en Inde par mois

3) Brésil

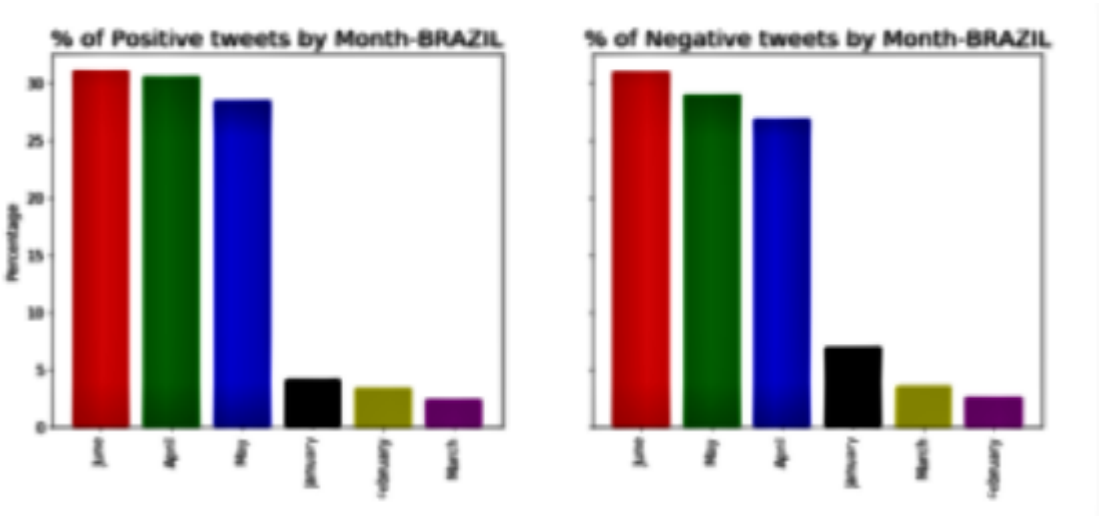

Figure 10 : Pourcentage de sentiments positifs et négatifs au Brésil par mois

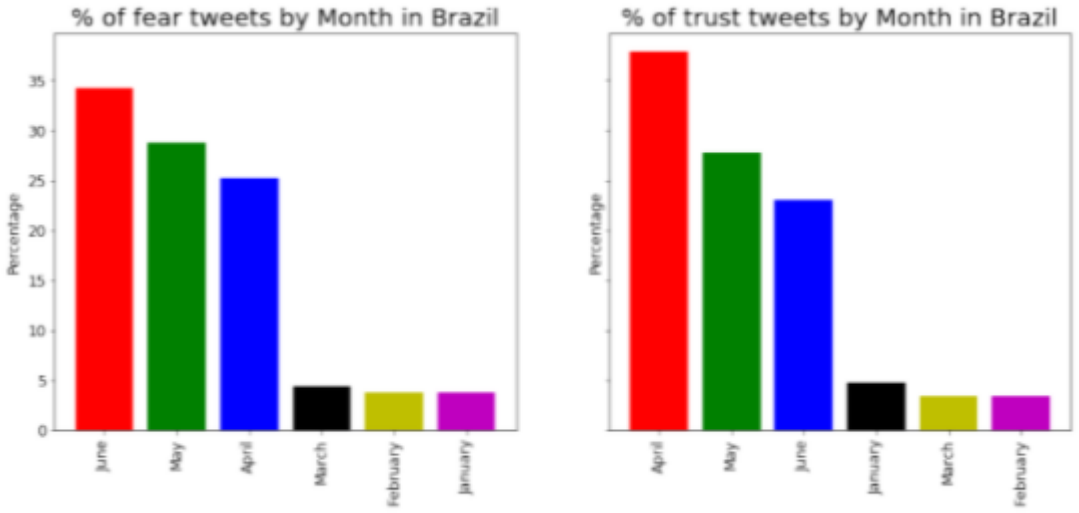

Figure 11 : Pourcentage de confiance et de peur au Brésil par mois

D)Work From Home (WFH) et Online Learning (OL)

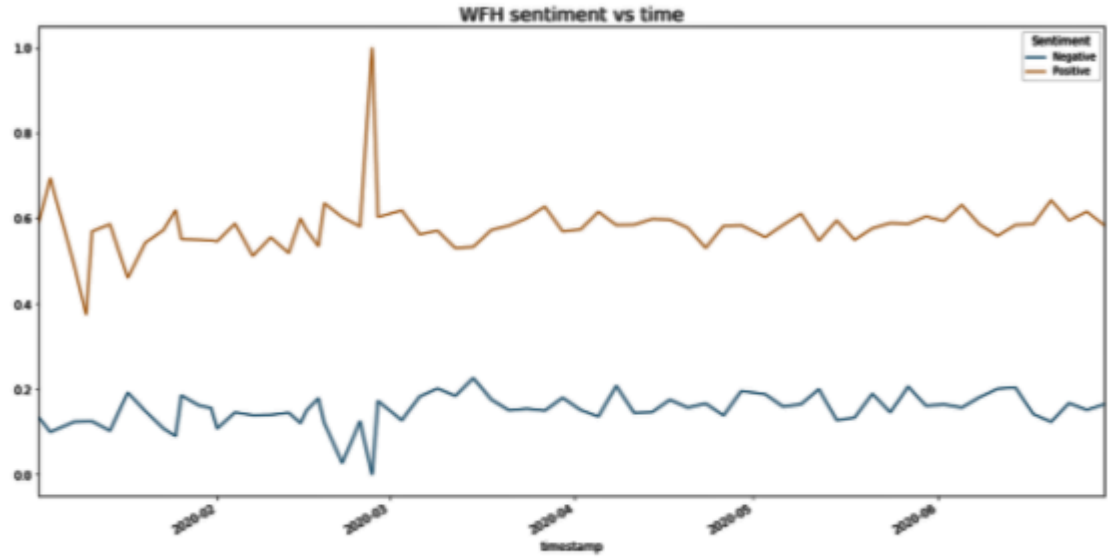

Figure 12 : Le sentiment du WFH par rapport au temps

Le sentiment général positif lié au travail dans la maison est toujours resté supérieur au sentiment négatif comme le montre la figure.

Figure 13 : OL sentiment vs temps

La proportion des tweets liés à l'apprentissage en ligne a été positive la plupart du temps hormis quelques cas comme le montre la figure d'avant.

3. Résumée

Ce chapitre présente 3 articles liés à l'analyse des sentiments pour le thème de la Pandémie du Covid_19 ainsi que des thèmes annexes tel que le télétravail ou le e-learning ainsi que leurs résultats avec leur modèle de Machine Learning utilisé.

Les résultats de ce chapitre vont nous permettre d'approfondir notre approche ainsi que d'identifier ce qui se fait actuellement dans le domaine de l'analyse des sentiments par rapport au Covid_19 pour pouvoir situer notre contribution dans ce domaine.

Pour conclure ce chapitre,une étude comparative a été réalisée pour comparer les 3 articles entre eux sur un nombre de critères précis,ces critères vont nous aider pour la suite du projet pour raffiner notre approche.

Comparaison finale des 3 articles:

Suite à notre étude des trois article, nous les avons comparé sur plusieurs critères et le tableau (12) représente la résultante finale de cette comparaison:

	Article 1	Article 2	Article 3
Type de polarité	*Binaire *Multiclass(3 classe)	Multiclass	Binaire
Nombre de tweets	1er dataset ⇒ 10.00 tweets 2eme dataset ⇒ 10.00 tweets 3eme dataset ⇒combinaison entre les deux (20.000 tweets)	1er dataset ⇒ 3500 tweets 2eme dataset ⇒ 3500 tweets 3eme dataset ⇒ 23.490 tweets 4eme dataset ⇒ 13.088 tweets	1er dataset ⇒ 165.116 tweets 2eme dataset ⇒40.756 tweets 3eme dataset⇒41.349 tweets 4eme dataset⇒Obtenu de Kaggle
Nombre de datasets	3	4	4
Objectif	Déterminer l'impact du covid-19 sur les émotions des personnes	Découvrir l'impact de la covid-19 sur les gens	Découvrir l'impact du covid-19 sur deux événement : -Travail chez sois (WFH) -Apprentissage en ligne(OL)
Algorithme utilisés	-Logistic Regression -Multinomial Naive Bayes	-K-means	-Réseau Neuronaux (LSTM)

	-Decision Tree -Random Forest -Support vector Machine -XGBoost	-Mini-Batch k-means	-Réseau Neuronaux Artificiels (ANN)
Mesure de performance	1)Classification binaire : Decision Tree,SVM,Random Forest ⇒Précision proche de 91% ⇒ Score F1 entre 85 et 86% 2)Classification Multiclass: Decision Tree,SVM ⇒Précision proche de 88 à 89% ⇒Score F1 entre 86 a 87% 3)Cross Dataset Evaluation: - XGBoost et Logistic Regression ⇒ précision 80 % -XGBoost ⇒ precision 82% -Random Forest⇒ precision 45% Random Forest⇒ precision 46%		-LSTM ⇒ Précision de 84,5% -ANN ⇒ Précision de 76%

Tableau 7 : Résultat finale de la comparaison

Conclusion:

Dans ce chapitre nous avons pu aborder les papiers de recherche qui nous ont inspiré à fixer notre cadre pour cette recherche ainsi que la comparaison entre les différentes techniques utilisées,leurs avantages et inconvénients pour nous situer par rapport à notre travail.

Le 3eme chapitre abordera la structure générale d'un système d'analyse des sentiments ainsi que les concepts théoriques nécessaires à l'élaboration de notre solution finale ainsi que toutes les étapes requises à l'implémentation finale.

Chapitre III :
Conception de la Solution

Introduction

Dans ce chapitre,nous allons introduire les principales théories concernant le traitement des données ainsi que le processus de nettoyage et une petite analyse exploratoire,ensuite nous aborderons les techniques utilisé ainsi que les algorithmes utilisé et nous présenterons a la fin les résultats expérimentaux ainsi que les mesures de performances choisi pour évaluer et comparer les modèles entre eux.

1) Architecture Global

Dans cette partie, nous allons aborder l'architecture globale d'un système d'analyse des sentiments.

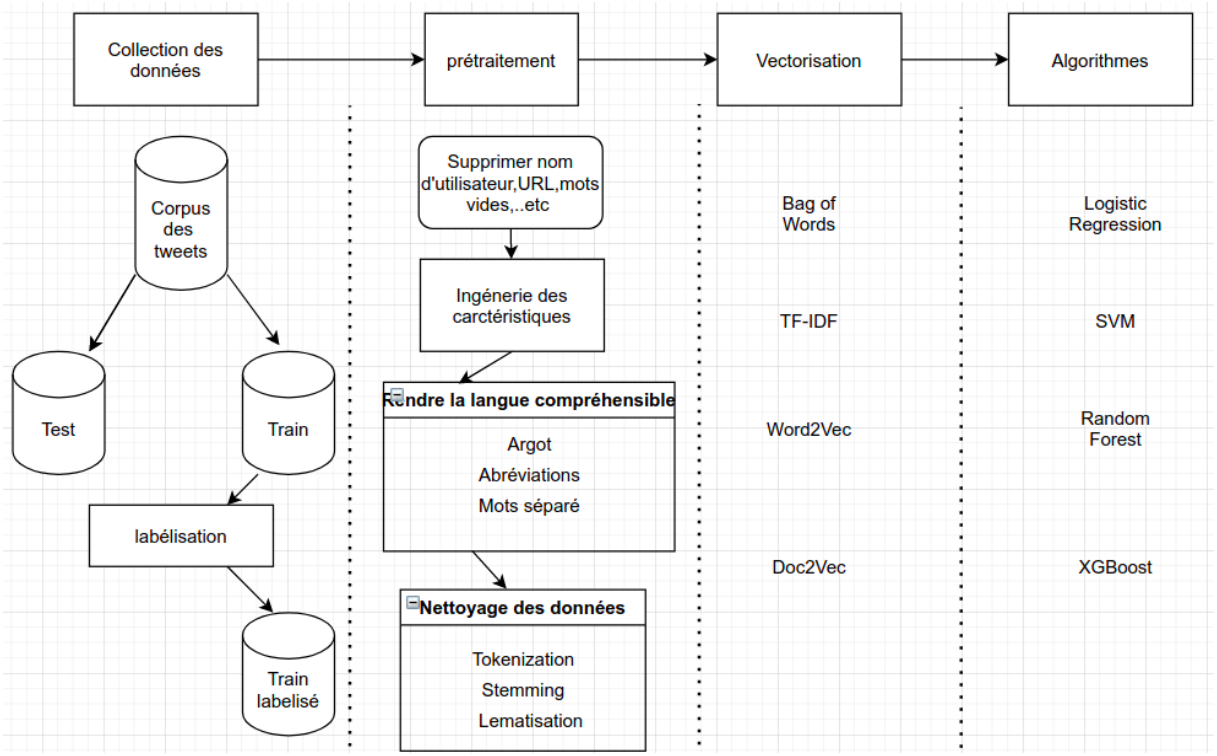

Figure 14 : L'architecture globale d'un système d'analyse des sentiments

La figure d'au dessus décrit visuellement les principales étapes de construction d'un système d'analyse des sentiments.

La première étape consiste dans la collecte des données, dans le cas d'un système d'analyse pour un réseau social comme Twitter, ce sera un ensemble de tweets regroupés dans un ensemble qu'on appelle un dataset.

Ce dataset est ensuite divisé en 2 parties qu'on dénomme partie d'entraînement et partie de test, la partie d'entraînement sert à entraîner nos algorithmes tandis que la partie test sert à

tester nos modèles pour savoir si nos prédictions sont juste ou erroné, le tout est calculé via des métriques.

Cette division peut s'opérer avant ou après les prétraitements, si c'est avant il est préférable de combiner les deux parties pour éviter la répétition des tâches sur les deux.

Labelliser un dataset consiste à ajouter des labels dans la partie test pour que les algorithmes d'apprentissages automatiques puissent apprendre les labels associés données d'entrée,dans un cas de système d'analyse des sentiments basé sur un corpus de tweets,les données sont les tweets eux même.

Après la collection vient l'étape du prétraitement qui consiste à nettoyer l'ensemble de données pour enlever toutes les impuretés du dataset pour préparer ce dernier aux algorithmes d'apprentissage automatiques,ceci consiste avec des opérations tels que le stemming ou la lemmatisation pour préparer le dataset aux algorithmes.

Vient ensuite l'étape de la vectorization qui consiste à transformer le dataset qui a déjà été nettoyé pour être transformé en un ensemble de vecteurs,la raison de l'existence de cette étape est la vitesse d'exécution dans un souci d'optimisation des algorithmes.

La dernière étape est l'application des algorithmes de Machine Learning sur les vecteurs déjà préparé,les algorithmes apprennent sur l'ensemble d'entraînement des données pour être ensuite testé sur la partie test,dans le cas d'un système d'analyse des sentiments,le but est que les algorithmes puissent apprendre du mieux possible les labels associés à chaque tweet pour pouvoir faire la prédiction d'une classe de labels quand un tweet est présenté en dehors de la partie d'entraînement, dans le cas d'une classification binaire,si le score du système dépasse 50% on considère que le système fait une meilleur prédiction qu'un être humain et donc considéré valable.

Dans la suite de ce chapitre,nous allons aborder en détail tous les aspects théoriques nécessaires pour la modélisation de la solution finale.

2) Traitement des données

Le traitement des données consiste à collecter les données, à en évaluer la qualité et à les nettoyer. Les données brutes recueillies pour un projet à partir de diverses sources sont généralement dans différents formats et ne conviennent pas à une analyse et une modélisation plus poussées.
Parfois, ces données recueillies ne sont pas vraiment propres et bien structurées. Il est alors difficile de travailler avec ces données, ce qui conduit à faire des erreurs, d'obtenir des informations trompeuses et de perdre un temps précieux.

Figure 15 : Processus de brassage des données

Dans notre cas, les données étant sous forme de texte, il est très important de nettoyer les tweets collectés afin de minimiser les différents problèmes que nous pourrions rencontrer après l'évaluation.

2.1 Récupération des données

Nous avons choisi pour notre recherche un ensemble de données sur kaggle qui donne un aperçu de la tendance générale du sentiment du discours public concernant la pandémie de COVID-19 sur Twitter [13]. Le jeu de données est mis à jour chaque semaine et continuera jusqu'à ce que le développement du jeu de données de tweets sur le coronavirus (COVID-19) soit terminé.
En raison de la politique des développeurs de Twitter [14], le groupe de recherche ne peut pas publier les données textuelles, mais seulement les ID des tweets et leurs annotations. Par conséquent, nous utilisons Hydrator une application pour récupérer ces tweets à partir de l'API de Twitter.

Pour accéder à l'API de Twitter, nous avons demandé un compte de développeur et obtenu l'autorisation. En utilisant Hydrator, nous avons pu collecter 9664 tweets sur 10000.
La raison pour laquelle nous n'avons pas pu collecter tous les tweets est que certains d'entre eux ont été supprimés par les utilisateurs ou rendus privés.
Hydrator nous aide à transformer ces ID de tweet en JSON ou CSV.

Le résultat de ce processus est un ensemble de données de 36 colonnes qui ont été réduite à quatre colonnes :
- Lang : la langue de chaque tweet .
- Text : contient les données textuelles qui sont le contenu du tweet en anglais.
- User_location : contient la localisation de chaque tweet .
- Polarity_final : caractéristique binaire contenant les annotations où 0 correspond à un tweet négatif et 1 à un tweet positif qui a été créé pour les besoins de notre recherche.

L'exemple sera illustré dans le chapitre 4 qui est le chapitre de l'implémentation de notre solution finale.

2.2 Nettoyage des données:

Après avoir réussi à récupérer nos données à partir de l'API Twitter et à construire notre dataset, il est maintenant temps d'évaluer nos données et d'appliquer les traitements requis pour en faire un modèle prêt pour les algorithmes.

Il existe deux types de problèmes que nous devons évaluer et nettoyer dans un ensemble de données : les problèmes de qualité et les problèmes d'ordre.

Un jeu de données de mauvaise qualité est un jeu de données qui ne nous intéresse pas,quelques problèmes de qualité généralement observés sont les valeurs manquantes, les données incohérentes, les types de données incorrects et les doublons.

D'autre part, les données désordonnées présentent des problèmes de structure, c'est-à-dire des problèmes d'ordre. Comme le dit Hadley Wickham dans son article Tidy Data [15], les données sont ordonnées lorsque :

- Chaque variable forme une colonne.
- Chaque observation forme une ligne.
- Chaque type d'unité d'observation forme un tableau.

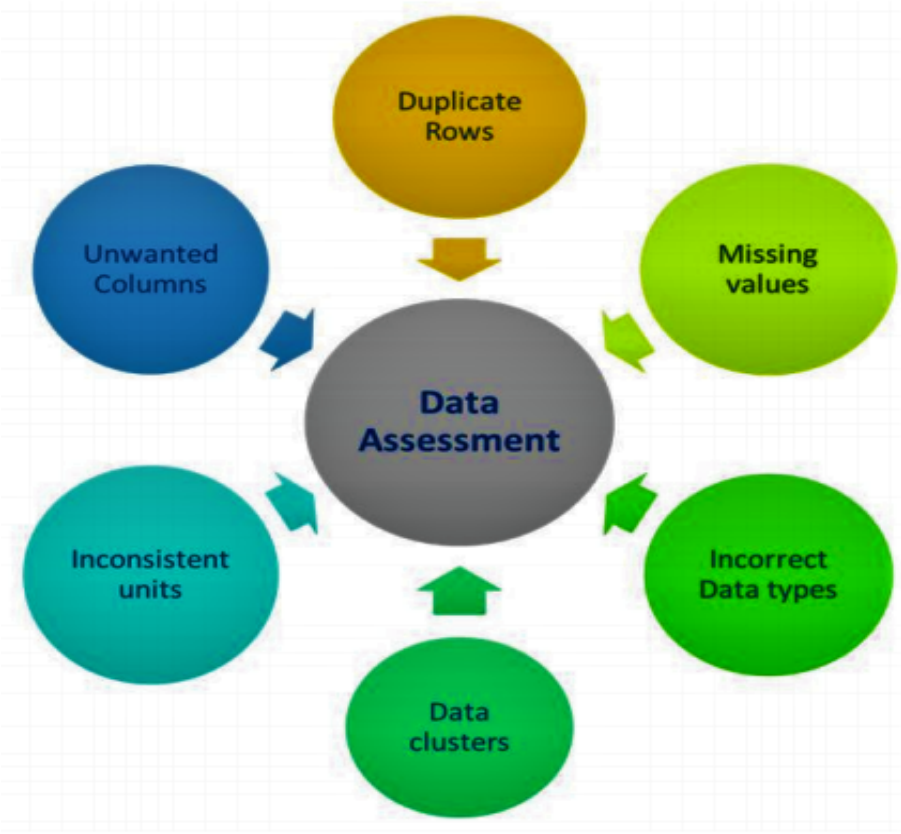

Figure 16 : Processus d'évaluation des données

- **Problèmes de qualité**

Après avoir évalué l'ensemble de données,nous avons découvert que notre jeu de données présente les problèmes de qualité suivants :
- URLs : Dans la colonne des tweets, les URL sont présentes dans les tweets.
- Mentions : Les mentions au format @user sont également présentes dans les tweets.
- Nouvelles lignes : comme nous avons récupéré les données à partir d'une API qui a renvoyé un fichier Json, les nouvelles lignes ont été exprimées dans le format “\n”.
- Espaces supplémentaires : certains tweets comportaient plus d'un caractère d'espacement entre deux .
- Caractères spéciaux : Les textes contenaient des caractères spéciaux comme les “_” ou “-”
- Des mots courts :Nous devons être un peu plus prudents en sélectionnant la longueur des mots que nous voulons supprimer. Ainsi, nous avons décidé de supprimer tous les mots d'une longueur égale ou inférieure à 3. Par exemple, des termes comme "hmm", "oh" sont très peu utiles. Il est préférable de s'en débarrasser.
- Emojis : Les emojis, les drapeaux, les symboles de transport et de carte et les pictogrammes ont été supprimés des tweets.
- Hashtags : les hashtags ont été séparés des tweets dans une colonne séparée.

Le nettoyage de l'ensemble de données de ces problèmes de qualité et de propreté a été effectué à l'aide d'expressions régulières() .
L’exemple sera illustré dans le chapitre 4 qui est le chapitre de l'implémentation de notre solution finale.

Expressions régulières:
Les expressions régulières sont une séquence de caractères qui spécifie un modèle de recherche [16]. Les modèles utilisés pour trouver les problèmes de qualité discutés précédemment sont décrits dans le tableau suivant

Problèmes	Expression régulière
URLs	('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|"[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+')
Mentions	'@[\w\-]+'
Nouvelles lignes	'\n+'
Hashtags	r'#.*?(?=\s|$)'
Espaces supplémentaires	'\s+'

Tableau 8 : Nettoyage des données avec des expressions régulières

Le résultat du processus de nettoyage est un jeu de données propre de 4 caractéristiques (hashtags , lang , text, polarity_final) et 9664 lignes.

L'exemple sera illustré dans le chapitre 4 qui est le chapitre de l'implémentation de notre solution finale.

3 Analyse exploratoire des données (AED)

L'analyse exploratoire des données désigne le processus critique consistant à effectuer des recherches sur des données afin de découvrir des motifs réguliers, de tester des hypothèses et de vérifier des suppositions à l'aide de statistiques sommaires et de représentations graphiques [17].

Pour mieux comprendre nos données, nous appliquons des techniques statistiques et logiques afin de décrire et d'illustrer, de condenser et de récapituler, et d'améliorer la qualité des données.

Selon Shamoo et Resnik (2003) [18] diverses procédures analytiques permettent de tirer des inférences inductives des données et de distinguer le signal (le phénomène d'intérêt) du bruit (les fluctuations statistiques) présent dans les données.

3.1 Analyse univariée

Ce type d'analyse des données ne comporte qu'une seule variable. L'analyse des données univariées est donc la forme d'analyse la plus simple puisque les informations ne portent que sur une seule quantité qui change. Elle ne traite pas des causes ou des relations et l'objectif principal de l'analyse est de décrire les données et de trouver des modèles,de décrire les données et de trouver les modèles qui existent en leur sein.

3.1.1 Polarity_final

Nous allons commencer par examiner notre variable cible.

La colonne Polarity_final de notre ensemble de données est une variable binaire de valeurs uniques 0 et 1 où :

- 0 : Tweet négatif
- 1 : Tweet Positif

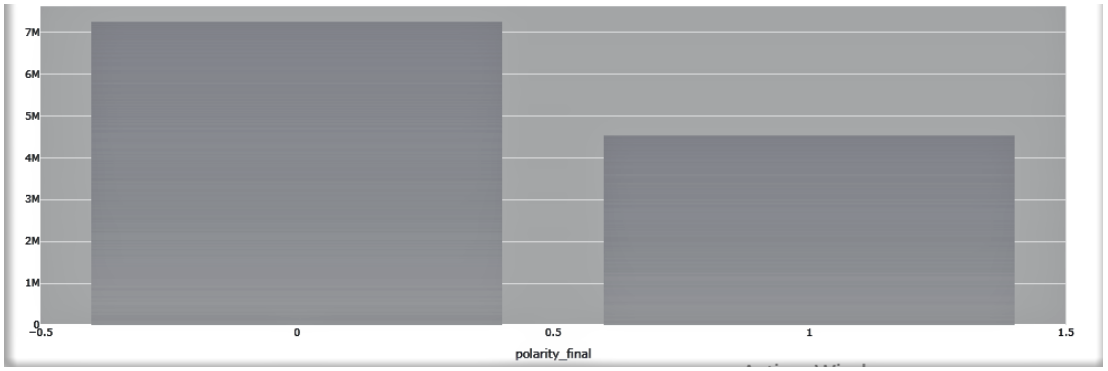

Figure 17 : nombre de tweets par polarity_final

Comme le montre le graphe, le nombre de tweets négatifs est plus élevé,ce qui est logique dans le contexte sanitaire actuel.

3.1.2 Text

La caractéristique texte est la caractéristique la plus importante de notre jeu de données. Elle contient les tweets que nous voulons analyser pour prédire le sentiment du tweet.

Pour mieux comprendre cette caractéristique, nous allons examiner plusieurs caractéristiques des tweets et en extraire des informations.

Tout d'abord, nous allons examiner les mots les plus courants dans les tweets en utilisant la technique du WordCloud [19].

Un WordCloud est une visualisation dans laquelle les mots les plus fréquents apparaissent en grande taille et les mots moins fréquents en plus petite taille.

Nous allons visualiser tous les mots de nos données à l'aide du WordCloud dans le plot ci-dessous.

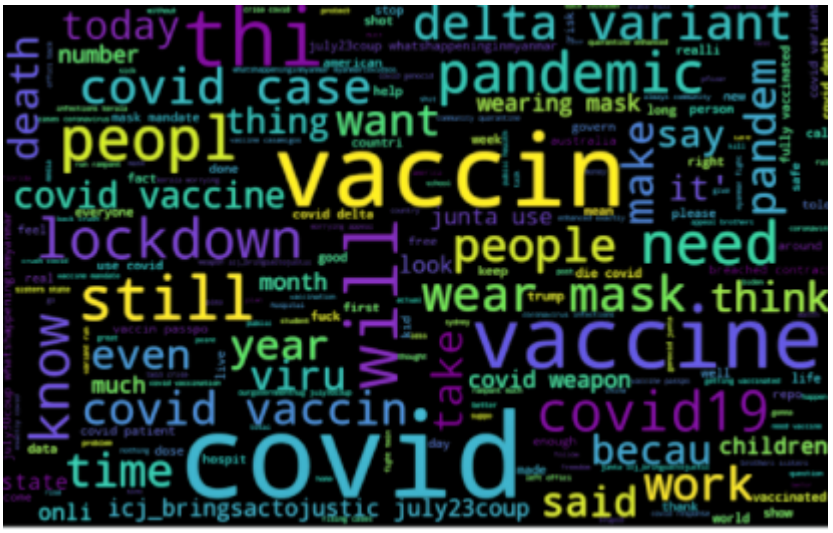

Figure 18 : Tous les mots WordCloud

Nous pouvons considérer la plupart des mots comme neutres jusqu'à ce que nous comprenions le contexte dans lequel ils ont été utilisés. Des mots comme vaccin, people,need ,et covid sont les plus fréquents.

Cependant, cela ne nous donne aucune idée sur la classification des mots . Par conséquent, nous allons tracer des nuages de mots séparés pour les deux classes (positifs et négatifs) dans nos données.

Le prochain Wordcloud est extrait de la partie négative des tweets.

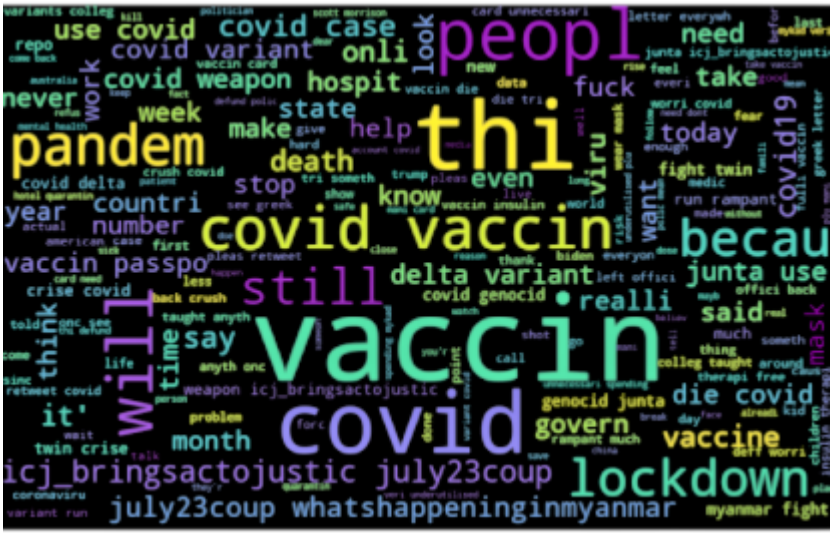

Figure 19 : Negatifs tweets WordCloud

La plupart des mots fréquents sont compatibles avec le sentiment, c'est-à-dire des tweets négatifs.
Les mots comme: “lockdown”, “weapon” ,”die” et “hospital” sont des mots négatifs dans ces tweets.
Le prochain Wordcloud est extrait de la partie positifs des tweets.

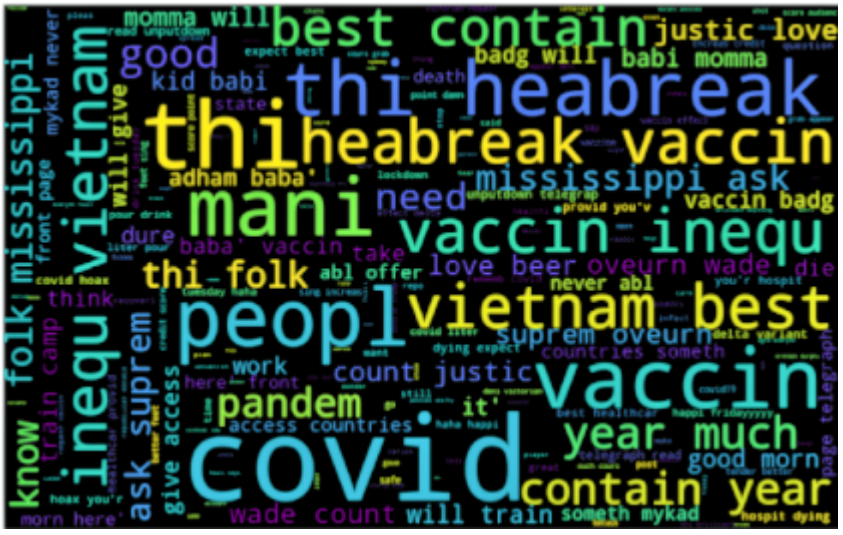

Figure 20 : Positifs tweets WordCloud

Comme on peut le voir, la plupart des mots ont une connotation positive. Il semble donc que nous ayons de bonnes données textuelles sur lesquelles travailler.

4 Encodage de texte

L'encodage de texte est un processus qui consiste à convertir un texte significatif en une représentation numérique/vectorielle afin de préserver le contexte et la relation entre les mots et les phrases, de sorte qu'une machine puisse comprendre le modèle associé à tout texte et puisse comprendre le contexte des phrases. Parce que les machines ne comprennent pas les caractères, mots ou phrases, nous ne pouvons pas transmettre du texte brut à des machines en tant qu'entré tant que nous ne les convertissons pas en chiffres. Nous devons donc procéder à l'encodage du texte.
Il existe de nombreuses méthodes pour convertir un texte en vecteurs numériques,notre recherche va se baser sur 4 techniques et qui sont:

- Bag of Words (BOW)
- Encodage TF-IDF
- Encodage Word2Vec
- Encodage Doc2Vec

Ces 4 modèles sont actuellement parmi ceux qui sont les plus utilisés dans la littérature [20] et qui ont tous hérité de la technique Word Embeddings.

Word Embedding est une technique de représentation vectorielle d'un mot,ceci permet aux algorithmes d'apprentissage automatiques de « comprendre » le mot, en effet si un mot comme « chien » est représenté par un vecteur [0 .32,0 .22,0.60,0.80] et que tous les mots dans le dictionnaire sont encodé dans le même format,il est donc possible de comparer les vecteurs de mots entre eux en utilisant par exemple la distance Cosine ou une distance euclidienne,

Une bonne représentation permet par exemple de trouver une distance proche entre le mot "chien" et le mot "maître " , en plus de pouvoir permettre d'atténuer le problème de la malédiction de la dimensionnalité,ceci permet aux algorithmes de mieux performer les différentes opérations mathématiques comme la factorization matricielle,le produit scalaire,etc

Nous allons par la suite comparer les 4 modèles entre eux pour trouver la meilleure combinaison avec les algorithmes que nous allons choisir par la suite.

4.1 Bag Of Words(BOW):

Un modèle Bag Of Words, ou BoW en abrégé, est un moyen d'extraire des caractéristiques du texte pour les utiliser dans la modélisation, par exemple avec des algorithmes d'apprentissage automatique.Cette approche est très simple et flexible, et peut être utilisée de multiples façons pour extraire des caractéristiques de documents.BoW est une représentation du texte qui décrit l'occurrence des mots dans un document. Il implique deux éléments :

- Un vocabulaire de mots connus.
- Une mesure de la présence de mots connus.

On l'appelle "Bag" Of Words, car toute information sur l'ordre ou la structure des mots dans le document est écartée. Le modèle se préoccupe uniquement de la présence de mots connus dans le document, et non de leur emplacement dans le document.L'intuition est que les documents sont similaires s'ils ont un contenu similaire. De plus, à partir du contenu seul, nous pouvons apprendre quelque chose sur le sens du document.
Bag Of Words peut être aussi simple ou complexe que vous le souhaitez. La complexité réside à la fois dans la façon de concevoir le vocabulaire des mots connus (ou tokens) et dans la façon de noter la présence des mots connus.

4.2 Codage TF-IDF (Term Frequency — Inverse Data Frequency)

TF-IDF a été inventée pour la recherche de documents et la récupération d'informations. Il fonctionne en augmentant proportionnellement au nombre de fois qu'un mot apparaît dans un document, mais est compensé par le nombre de documents qui contiennent le mot. Ainsi, les mots qui sont communs dans tous les documents, ont un rang faible même s'ils apparaissent plusieurs fois, car ils ne signifient pas grand-chose pour ce document en particulier.
TF-IDF pour un mot dans un document est calculé en multipliant deux métriques différentes :
- Term Frequency (La fréquence des termes) : C'est l'occurrence du mot actuel $t$dans la phrase actuelle $d$par rapport au nombre total de mots dans la phrase actuelle.

$$tf(t,d) = log(1 + freq(t,d))$$

- Inverse Data Frequency (Fréquence inverse des données) : logdu nombre total de mots N dans l'ensemble du corpus de données Dpar rapport au nombre total de phrases $d$contenant le mot courant t

$$idf(t,D) = log(N \div count(d \in D: t \in d))$$

En multipliant ces deux nombres, on obtient le score TF-IDF d'un mot dans un document.

$$tf\ idf(t,d,D) = tf(t,d) \times idf(t,D)$$

Plus le score est élevé, plus le mot est pertinent dans ce document particulier.
En utilisant le codage TF-IDF, nous pouvons transformer nos données textuelles en nombres ou en fréquences qui représentent le poids de chaque mot dans chaque tweet et dans l'ensemble du jeu de données.

Pour appliquer cet encodage, nous avons fait appel au package python SCIKIT-LEARN [21]et à l'application de la méthode TfidfVectorizer [22] avec les paramètres suivants :

- Min_df = 2
Ignorer les termes dont la fréquence de document est strictement inférieure à 2.

- Max_df = 0,9
Ignorer les termes dont la fréquence de document est strictement supérieure à 90%.

- Sublinear_tf = True
Appliquer une mise à l'échelle TF sublinéaire, c'est-à-dire remplacer TF par 1 + log(TF) pour normaliser les données.

Les résultats de la méthode d'encodage de texte TF-IDF sur notre jeu de données est une matrice éparse de 9664 lignes et 6155 caractéristiques.

4.3 Encodage Word2Vec

Le Word Embeddings est le moyen moderne de représentation vectorielle des mots,l'objectif est la réduction des vecteurs de grande dimension en des vecteurs de plus petite dimensions tout en conservant la similarité du contexte dans le corpus.

Word2Vec est l'une des techniques les plus populaires pour apprendre le word embeddings à l'aide d'un réseau neuronal. Elle a été développée par Tomas Mikolov en 2013 chez Google ([23].

Le Word Embedding est basé sur la création d'un vecteur pour chaque mot dans notre corpus de telle sorte à ce que,si on essaie de visualiser le résultat,chaque mot occupe 1 dimension.

L'objectif est de faire en sorte que les mots ayant un contexte similaire occupent des positions spatiales proches.
Mathématiquement, le cosinus de l'angle entre de tels vecteurs devrait être proche de 1, c'est-à-dire, angle proche de 0.

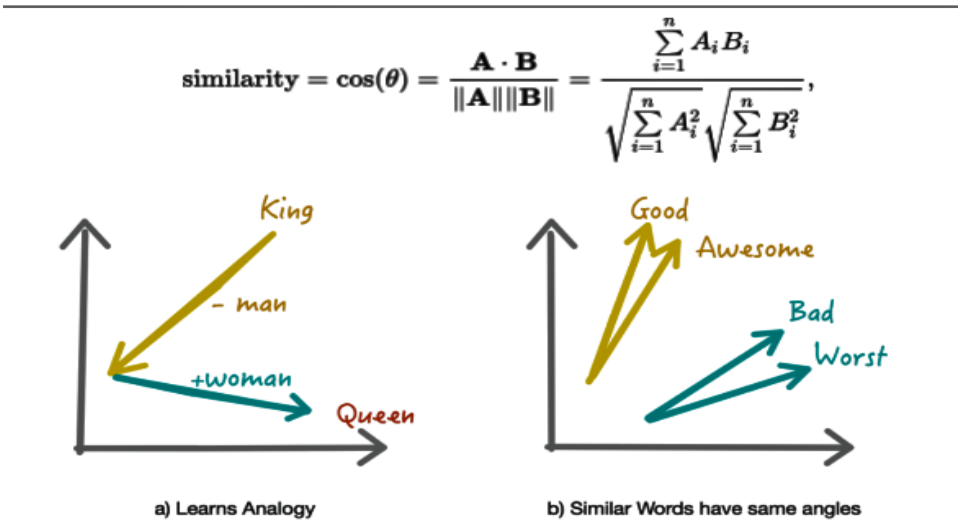

Figure 21 : Similarité Cosine

Word2Vec est une méthode permettant de construire un tel embedding.Elle peut être obtenue à l'aide de deux méthodes (impliquant toutes deux des réseaux neuronaux) : Skip Gram et Common Bag Of Words (CBOW).
La méthode CBOW tend à prédire la probabilité d'un mot en fonction d'un contexte. Un contexte peut être un seul mot adjacent ou un groupe de mots environnants. Le modèle Skip-gram fonctionne de manière inverse, il essaie de prédire le contexte d'un mot donné.
Les deux techniques ont leurs propres avantages et inconvénients . Cependant, selon Mikolov, le modèle Skip Gram fonctionne bien avec une petite quantité de données et représente bien les mots rares.
Pour appliquer cet encodage, nous avons fait appel à la bibliothèque Python Gensim et au module Word2Vec avec les paramètres suivants :
- Taille = 200
La dimension des vecteurs d'incorporation des mots est de 200

- Fenêtre = 5
Les mots qui sont 5 à gauche et à droite des mots cibles sont considérés comme des mots de contexte.

- Min_count = 2
Ignorer les termes qui apparaissent seulement 2 fois dans le corpus.

- Sg = 1
Modèle Skip-Gram

- Hs = 0
l'échantillonnage négatif sera utilisé

- Negative = 10
10 mots de bruit devraient être tirés dans l'échantillonnage négatif

- Workers = 2
Utiliser 2 fils de travail pour entraîner le modèle (entraînement plus rapide avec les machines multicœurs)

- Epochs = 20
Le nombre d'itérations (Epochs) sur le corpus est de 20.

Comme nos données contiennent des tweets et pas seulement des mots, nous avons utilisé les vecteurs de mots du modèle word2vec pour créer une représentation vectorielle pour un tweet entier. Nous avons pris la moyenne de tous les vecteurs mots présents dans le tweet et la longueur du vecteur résultant sera la même. Nous allons répéter le même processus pour tous les tweets de nos données et obtenir leurs vecteurs.

Le résultat de l'encodage du texte à l'aide de Word2vec est une matrice éparse qui représente la similarité contextuelle entre les mots dans un format numérique que nous pouvons introduire dans nos modèles pour effectuer la classification des sentiments.

4.4 Encodage Doc2Vec :

Doc2vec, est une extension de word2vec, est utilisé pour générer des vecteurs de représentation de morceaux de texte (c'est-à-dire des phrases, des paragraphes, des documents, etc.).

4.5 Mise à l'échelle et fractionnement des données

La mise à l'échelle des données a été rendu possible grâce à l'utilisation de Text Blob qui est un package Python pour calculer la polarité primaires des tweets,la mise à l'échelle a été rendu possible grâce à une condition qui fait que chaque polarité qui est en dessous 1.00 est une polarité négative et de ce fait a été labellisé en 0 et pour chaque polarité au dessus de 1.00,une polarité positif a été attribué et de ce fait labellisé en 1.

Après avoir mis à l'échelle les caractéristiques de chacun de nos ensembles de données, nous allons passer à l'étape de division. La division des ensembles de données en ensembles d'entraînement et de test est importante pour créer un modèle d'apprentissage automatique. Pour former un modèle d'apprentissage automatique, quel que soit le type d'ensemble de données que nous utilisons, nous devons diviser l'ensemble de données en données d'entraînement et en données de test.

L'ensemble de données d'apprentissage sera utilisé pour former le modèle, tandis que l'ensemble de données de test est conservé comme données inédites pour tester les prédictions de notre modèle.
Lors de la division d'un ensemble de données, il y a deux choses à prendre en compte :
- Si nous disposons de peu de données d'apprentissage, les estimations de notre modèle présentent un biais plus important.
- Si nous avons peu de données de test, la statistique de performance aura une plus grande variance.
Les ensembles de données doivent être divisés de manière à ce que ni le biais ni la variance ne soient trop élevés, ce qui dépend davantage de la quantité de données dont nous disposons.
Parce que nos deux ensembles de données ont 9664 lignes, nous avons décidé de diviser notre ensemble de données en une division 70:30.
Chacun de nos jeux de données sera divisé en 70 % de jeu d'entraînement et 30% de jeu de test. Nous utiliserons l'ensemble d'essai pour estimer les performances de notre modèle à l'aide de mesures d'évaluation qui comparent les prédictions du modèle aux résultats attendus dans l'ensemble d'essai.

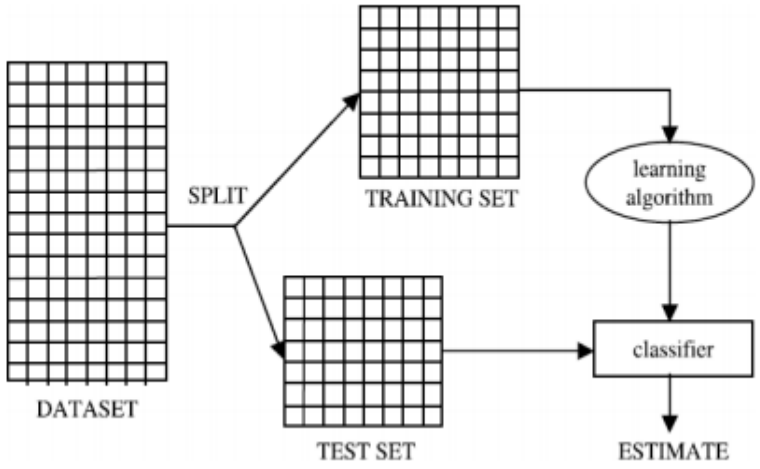

Figure 22 : Division des données

5 Modèles d'apprentissage automatique

L'analyse du sentiment des tweets COVID-19 dans notre corpus de tweets est un problème de classification.Nous essayons de classer chaque tweet comme positif ou négatif.
La classification est le processus de mise en correspondance et de regroupement des points de données dans des catégories prédéfinies.
Dans notre cas, ces catégories prédéfinies sont les classes positives et négatives. Avec l'aide de ces points de données d'entraînement pré-catégorisés, la classification dans les programmes d'apprentissage automatique utilise un algorithme pour classer les points de données non vus, comme l'ensemble de test, dans leurs classes pertinentes.

Un modèle de classification tente de tirer une conclusion à partir des valeurs observées. Étant donné les entrées, un classificateur tente de prédire la valeur d'un ou plusieurs résultats. Les résultats sont des étiquettes dans la variable cible de notre ensemble de données.
Les algorithmes de classification dans l'apprentissage automatique utilisent des données d'entraînement en entrée dans le but de prédire la probabilité que les données qui suivent appartiennent à l'une des classes.
En bref, la classification est une forme de reconnaissance des formes. Dans notre cas, les algorithmes de classification appliqués à nos données de formation trouvent les mêmes mots ou sentiments dans les ensembles de données futurs.
Nous allons explorer en détail les algorithmes de classification de la régression logistique , la machine à vecteurs de support,le XG Boost et la Random Forest , et nous expliquerons comment les utiliser pour l'analyse des sentiments afin de catégoriser les données textuelles par polarité des sentiments. Nous entraîneront ensuite nos classificateurs pour voir lequel d'entre eux s'adapte le mieux à nos données et prédit les sentiments (Positif ou négatif) dans nos tweets.

5.1 Modèle de régression logistique

La régression logistique est un modèle d'apprentissage automatique de classification. Elle est basée sur l'analyse de l'effet des variables d'entrée sur la variable cible.
Elle doit son nom à la fonction utilisée au cœur de la méthode, la fonction logistique.
La fonction logistique, également appelée fonction sigmoïde. Il s'agit d'une courbe en forme de S qui peut prendre n'importe quel nombre à valeur réelle et le transformer en une valeur comprise entre 0 et 1, mais jamais exactement à ces limites.

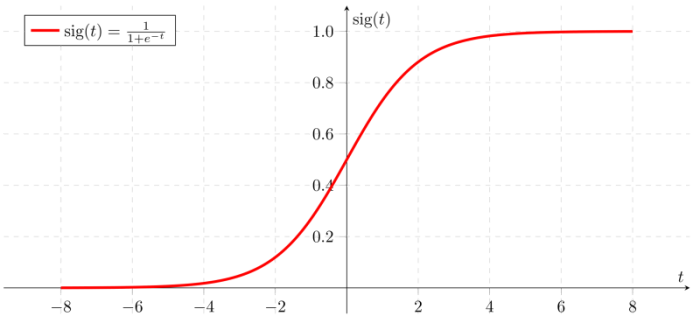

Figure 23 : Fonction sigmoïde

Contrairement au modèle de régression linéaire où nous prédisons la valeur de $y$en utilisant une équation linéaire et produisons une ligne qui s'adapte à nos points de données, le modèle de régression logistique alimente cette équation linéaire, nos données d'entrée Xcombinées linéairement en utilisant des valeurs de coefficient βa la fonction logistique pour avoir en sortie une valeur entre 0 et 1.

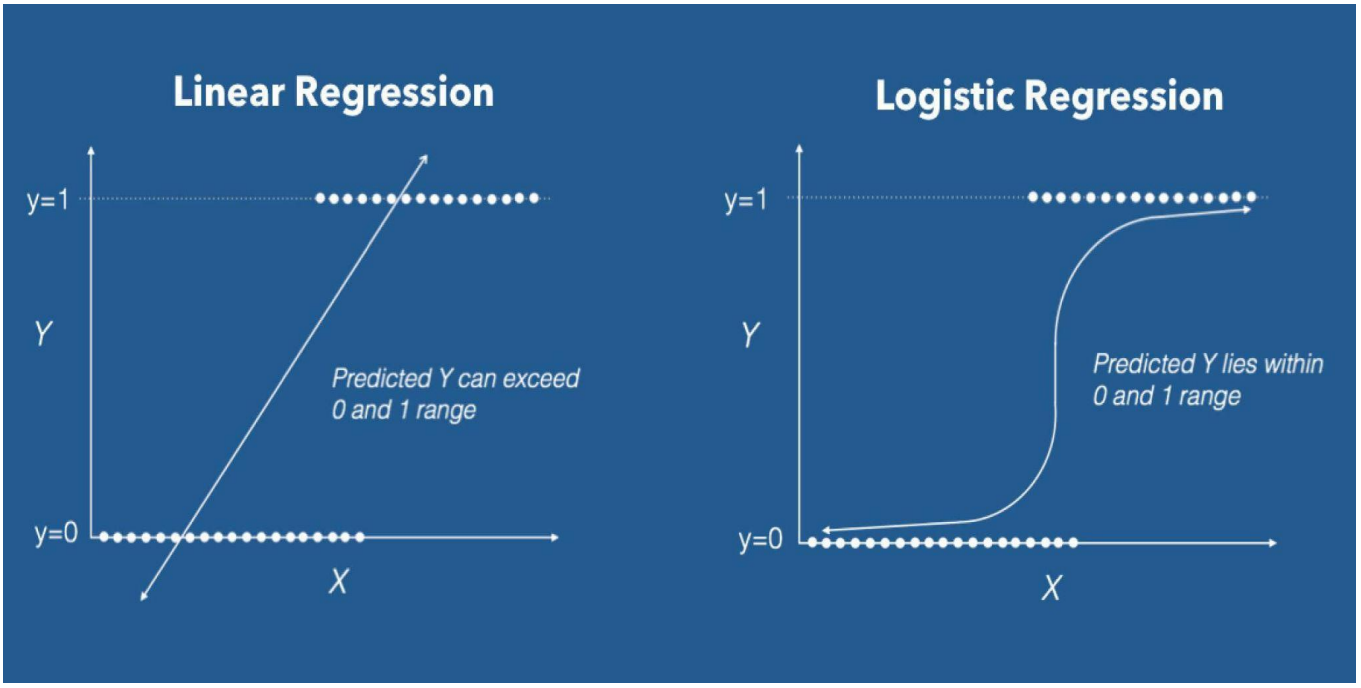

Figure 24 : Régression logistique vs régression linéaire

Les résultats de l'utilisation de la fonction logistique sont que les prédictions se situeront dans la plage 0 et 1.c'est-à-dire par rapport au nombre de classes de notre variable cible.

En d'autres termes, nous modélisons la probabilité qu'une entrée X appartienne à la classe par défaut (y = positif), nous pouvons l'écrire formellement comme suit :

$$0 < P(positif(i) = 1) = 1 / 1 + e(-(\beta 0 + \beta 1 x 1(i) 1 + \cdots + \beta p x p(i))) < 1$$

Parce que la régression logistique renvoie les probabilités de chaque tweet appartenant à la classe par défaut “Positive”, nous aurons besoin d'un seuil (généralement 0,5) pour comparer ces probabilités. La probabilité est supérieure à cette valeur seuil, le tweet est prédit comme appartenant à la classe positive, sinon il est prédit comme n'appartenant pas à la classe positive.
Pour interpréter les poids dans la régression logistique, nous devons reformuler la dernière équation de sorte que le côté droit de la formule soit linéaire. Pour ce faire, nous devons calculer le logarithme du rapport de cotes.

$$log(P(Positif = 1)/1 - P(Positif = 1)) = log(P(Positif = 1)/P(Positif = 0)) = \beta 0 + \beta 1 x 1 + \ldots..$$

Le rapport de cotes est simplement la probabilité de l'événement $(positif(i) = 1)$ divisée par la probabilité de l'absence d'événement $(1 - (Positif(i) = 1)$, on les enroule dans le

logarithme et on obtient les côtes logarithmiques. Comme ça , nous obtenons un côté droit linéaire de la formule de régression logistique, que nous pouvons interpréter.

5.2 Support Vector Machine (SVM)

SVM est un algorithme supervisé de machine learning qui peut être utilisé pour des problèmes de classification ou de régression mais plus généralement utilisé pour de la classification.

L'idée est de trouver un hyperplan pour séparer un dataset en 2 classes comme l'image le résumé ci dessous:

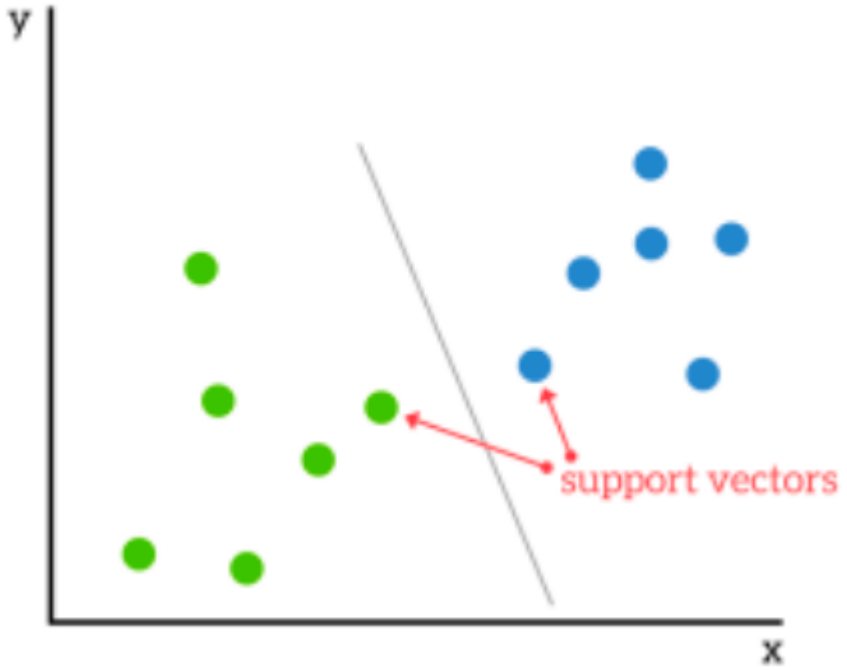

Figure 25 : Séparation de deux classes Model SVM

Les éléments critiques d'un dataset sont les "support vectors" car ce sont les points les plus proches de l'hyperplan car ils sont ceux qui guident la position de ce dernier.

L'hyperplan est la ligne qui sépare le dataset dans ce cas de figure, plus les points sont loin de l'hyperplan plus on est sur qu'ils ont été correctement classifié, ce qu'on cherche c'est que les points soient les plus loin possible de l'hyperplan tout en étant certain qu'ils sont du bon côté de la ligne.

La distance entre l'hyperplan et les points les plus proches est connue sous le nom de "Marge", le but est de choisir un hyperplan avec la plus grosse marge pour être sûr de bien séparer les deux classes.

Ceci change lorsqu' on a affaire à un dataset ou il n'est plus possible de créer un hyperplan clair pour séparer les deux classes,il faut dans ce cas "monter" en dimension et cette technique s'appelle "Kernelling".
Les avantages de SVM sont la précision,fonctionne mieux dans un petit dataset mais ne fonctionne pas efficacement dans un dataset ou les classes peuvent se chevaucher entre eux.

5.3 Random Forest (forêt aléatoires)

La forêt aléatoire est une méthode d'ensemble qui combine plusieurs arbres de décision pour classifier. Le résultat de la forêt aléatoire est donc généralement meilleur que celui des arbres de décision.
Les forêts aléatoires sont un algorithme d'apprentissage supervisé. Il peut être utilisé à la fois pour la classification et la régression. C'est également l'algorithme le plus flexible et le plus facile à utiliser. Une forêt est composée d'arbres. On dit que plus il y a d'arbres, plus une forêt est robuste. Les forêts aléatoires créent des arbres de décision sur des échantillons de données sélectionnés au hasard, obtiennent une prédiction de chaque arbre et sélectionnent la meilleure solution au moyen d'un vote. Elle fournit également un assez bon indicateur de l'importance de la caractéristique.

Les forêts aléatoires ont de nombreuses applications, comme les moteurs de recommandation, la classification d'images et la sélection de caractéristiques. Elle peut être utilisée pour classer les demandeurs de prêts loyaux, identifier les activités frauduleuses et prédire les maladies. Elle est à la base de l'algorithme Boruta, qui sélectionne les caractéristiques importantes dans un ensemble de données.

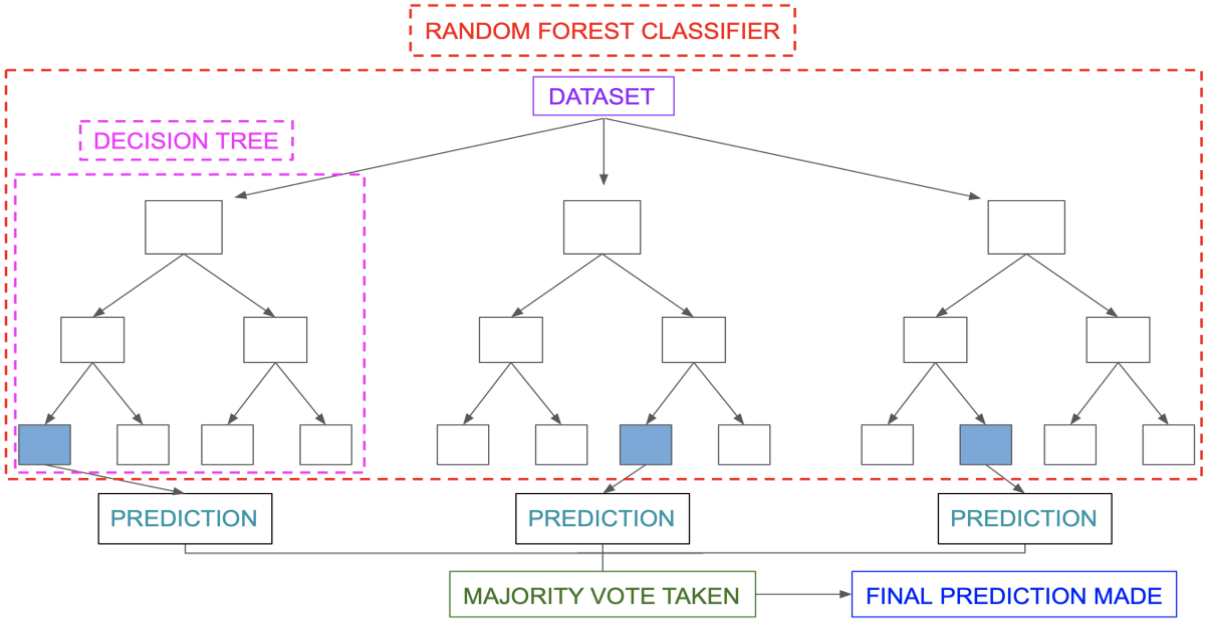

Figure 26 : Random Forest classifier

5.4 XGBoost

XGBoost qui signifie "eXtreme Gradient Boosting" est un algorithme supervisé de machine learning,elle est une implémentation du framework Gradient Boosting ,c'est aujourd'hui une bibliothèque open source complète.

XGBoost est une extension des arbres de décision boostés par le gradient (GBM), spécialement conçue pour améliorer la vitesse et les performances.

XGBoost utilise un type d'arbre de décision appelé CART: Classification and Decision Tree

Le boosting est une méthode qui utilise le principe de l'apprentissage d'ensemble, mais dans un ordre séquentiel,c'est un processus qui combine les décisions de plusieurs modèles sous-jacents et utilise une technique de vote pour déterminer la prédiction finale.

Les forêts aléatoires et l'ensachage sont deux méthodes d'apprentissage d'ensemble célèbres.

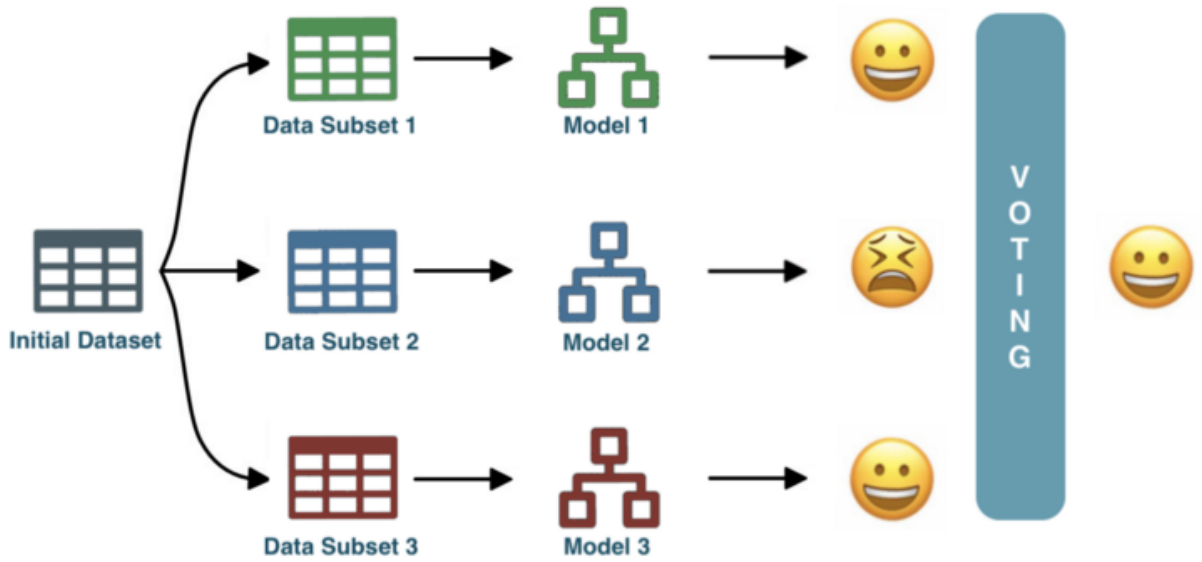

Figure 27 : Un exemple d'Ensemble Learning avec la méthode Bagging et une stratégie de vote majoritaire

Le renforcement est un type d'apprentissage d'ensemble qui utilise le résultat du modèle précédent comme entrée du modèle suivant. Au lieu d'entraîner les modèles séparément, chaque nouveau modèle étant formé pour corriger les erreurs des précédents.
A chaque itération, les résultats correctement prédits reçoivent un poids inférieur et ceux qui ont mal prédit un poids plus élevé. Il utilise ensuite une moyenne pondérée pour produire un résultat final.

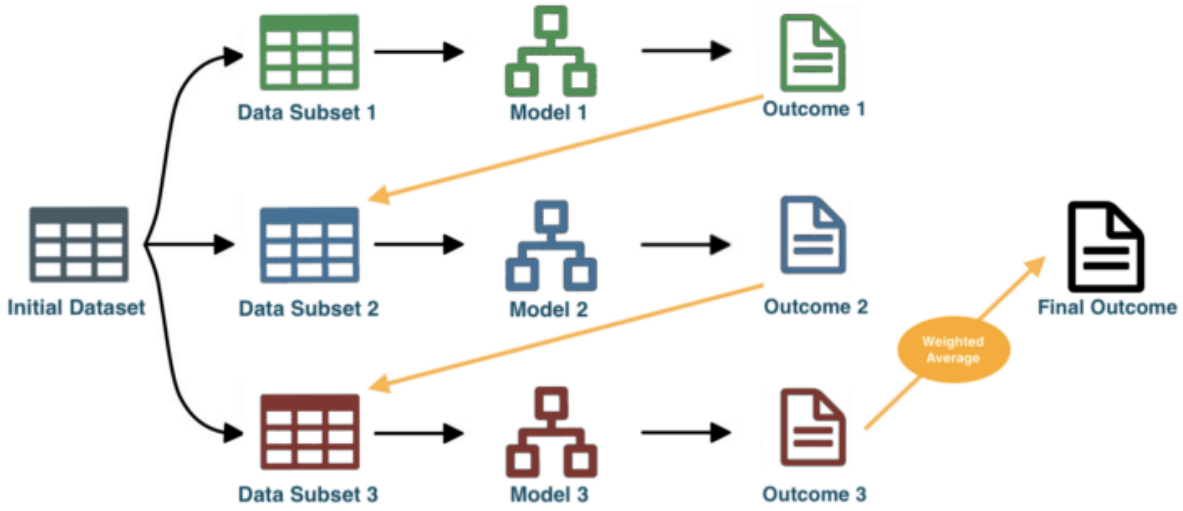

Figure 28 : Un exemple d'Ensemble Learning avec la méthode Boosting, en utilisant la stratégie Weighted-Average

Le Gradient Boosting est une méthode d'amplification où les erreurs sont minimisées à l'aide d'un algorithme de descente de gradient, la descente de gradient est un algorithme d'optimisation itératif utilisé pour minimiser une fonction de perte.

La fonction de perte quantifie la distance entre notre prédiction et le résultat réel pour un point de données donné. Plus les prévisions sont bonnes, moins la sortie de la fonction de perte sera faible.

Lorsque nous construisons notre modèle, l'objectif est de minimiser la fonction de perte sur tous les points de données. Par exemple, l'erreur quadratique moyenne (MSE) est la fonction de perte la plus couramment utilisée pour la régression.

XGBoost est une implémentation optimisée de la méthode de Gradient Boosting

6 Mesures de performance

Il existe de nombreux types de métriques d'évaluation disponibles pour tester les prédictions d'un modèle. Il s'agit notamment de l'exactitude, du score F1, de la matrice de confusion, de la précision, du rappel, etc.
Dans cette partie, nous allons énumérer et définir les mesures de performance que nous avons utilisées pour évaluer nos modèles et les comparer afin de pouvoir choisir le meilleur modèle.

6.1 Matrice de confusion

La matrice de confusion est un moyen de mesurer les performances de notre modèle de classification. Une matrice de confusion résume les informations sur les classes réelles et prédites. Considérons la disposition suivante de la matrice de confusion où 1 dénote un tweet positif et 0 dénote un tweet négatif.

	Classe Actuellement Positive	Classe Actuellement Négatif
Classe Prédite Positivement	TP	FP
Classe Prédite Négativement	FN	TN

Tableau 9 : Exemple matrice de confusion

Il existe quatre types de résultats d'une matrice de confusion :
- Vrais positifs (TP) : le modèle prédit qu'un point de données appartient à la classe Positive et il appartient réellement à cette classe.
- Vrais négatifs (TN) : le modèle prédit qu'un point de données appartient à la classe négative et qu'il appartient réellement à cette classe. appartient réellement à cette classe.
- Faux positifs (FP) : le modèle prédit qu'un point de données appartient à la classe négative mais en réalité il n'y appartient pas.
- Faux négatifs (FN) : le modèle prédit qu'un point de données appartient à la classe positive alors qu'en réalité, c'est le cas.

6.2 Accuracy

L'exactitude ou l'accuracy est définie comme la proportion de prédictions correctes pour les données de test. Elle peut être calculée en divisant le nombre de prédictions correctes par le nombre total de prédictions. En d'autres termes, il s'agit de la somme des Vrais Positifs et des Vrais Négatifs divisée par le total de toutes les prédictions : Vrais Positifs, Vrais Négatifs, Faux Positifs et Faux Négatifs.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

Elle est utilisée au mieux lorsque nous avons affaire à des données équilibrées.

63 Precision

La précision et le rappel sont deux autres mesures de performance que nous utiliserons. La précision est définie comme la fraction de vrais positifs parmi toutes les prédictions qui appartiennent à la classe positive

$$Precision = TP/(TP + FP)$$

La précision est un bon choix pour évaluer notre modèle lorsque nous voulons être certains de notre prédiction afin d'éviter les erreurs de type II. Dans notre cas, l'erreur de type II serait de classer un tweet comme étant négatif alors qu'en réalité, il l'est.

6.4 Recall

Le rappel est défini comme la fraction de vrais positifs par rapport à toutes les prédictions qui appartiennent vraiment à la classe positive.

$$Recall = TP/(TP + FN)$$

Le rappel est un excellent choix pour évaluer notre modèle lorsque nous voulons capturer le plus de positifs possible. Il s'agit également d'une mesure importante sur laquelle il faut se concentrer pour éviter l'erreur de type I. Ce type d'erreur est ce sur quoi nous devons nous concentrer, car il est plus grave de classer un tweet comme positif alors qu'en réalité il ne l'est pas.

6.5 F1 Score

Le score F1 est la moyenne harmonique de la précision et du rappel. Sa valeur maximale est 1, ce qui indique une précision et un rappel parfaits, tandis que sa valeur minimale est 0, ce qui signifie que l'une ou l'autre de ces deux valeurs est nulle. Ce score maintient un équilibre entre la précision et le rappel pour notre modèle de classification.

$$F - measure = (2 * Precision * Recall) / (Precision + Recall)$$

Le score F1 doit être utilisé lorsque nous voulons nous assurer que les tweets positifs sont correctement classés, mais aussi lorsque nous voulons attraper le plus grand nombre possible de tweets négatifs aussi.

Conclusion

Dans ce chapitre nous avons tout d'abord introduit les principaux concepts théorique qui nous aident à modéliser notre recherche,les traitements effectué sur notre dataset ainsi que les difficultés rencontré,ensuite nous avons aborder les algorithmes utilisés dans notre cadre de recherche ainsi que les principaux mesures de performances retenus avec a la fin une présentation de nos résultats expérimentaux.

Chapitre IV :
Implémentation

Introduction

Dans ce dernier chapitre, nous allons aborder la partie implémentation de notre solution finale agrémenté du code qu'on a utilisé pour aboutir à notre solution finale ainsi que d'une application pour démontrer la faisabilité de notre effort.En ce sens, nous allons tout d'abord introduire les datasets utilisés ainsi que les packages utilisé pour réaliser notre projet,nous aborderons les prétraitement réalisées sur les datasets pour les préparer ensuite pour nos algorithmes de Machine Learning ainsi que toutes les étapes pour arriver aux résultats finaux.Pour finir nous allons aborder le déploiement qui a été réalisé ainsi la présentation de notre application final qui consiste à la détection des sentiments suivant un mot clé ou une liste de tweets suivant notre modèle pré entraîné auparavant pour la prédiction.

1 Packages utilisé

Les packages qui ont été utilisé pour notre projet peuvent être résumé dans ce qui suit:

1.1 : Python

Python est le langage de programmation choisi pour mener à bien notre projet,en effet,contrairement a R qui est plutôt orienté statistique, Python offre un vaste éventail de package qui en font un allié de taille pour tous les ingénieurs en Machine Learning et en Data Science en général.

L'inconvénient de Python réside dans le fait que c'est un langage interprété et non compilé, ce qui en fait un langage "lent" dans les traitements de haut niveau ou pour le Big Data en général ou on lui préfère un langage comme Scala.

Ce qu'on perd dans la vitesse est récupéré dans sa facilité d'écriture et d'interprétation ainsi qu'une large bibliothèque de package, ce qui en fait aujourd'hui un des langages les plus versatiles sur le marché actuellement,que ce soit en terme de développement web avec Flask,en Machine Learning avec Tensorflow ou PyTorch ainsi que l'écriture des scripts pour automatiser les tâches comme avec Selenium.

1.2 : Pandas

Pandas est une librairie Python,son nom est la contraction de "Python Data Analysis", elle permet l'analyse de donnée et la manipulation des données,en particulier elle permet d'utiliser une structure de donnée appelé DataFrame qui est une forme de matrice facilement modifiable ainsi que la manipulation des séries temporelles.

1.3 : NumPy

Numpy est une librairie Python, son nom est la contraction de “Numerical Python”, elle est destiné à manipuler des matrices ou des tableaux multidimensionnels ainsi que des fonctions mathématiques qui opèrent sur des tableaux.

1.4 : Scikit-Learn

Scikit-learn est une librairie Python destiné à l'apprentissage automatique et propose plusieurs algorithmes de machine learning prêt à être utilisé ainsi qu’une documentation riche avec plusieurs exemples, elle propose toute sorte d’algorithme comme des algorithmes de classification,régression ou de clustering, Scikit-learn est désigné de tel sorte à ce qu’elle s'intègre avec les libraires tel que NumPy.

1.5 : Gensim

Gensim est une librairie Python destiné à l'apprentissage non supervisé avec le ”topic modeling “ en utilisant des outils statistiques,Gensim est connu pour leur implémentation des modeles tels que Word2Vec,Doc2Vec, Latent Semantic Analysis,TF-IDF entre autres.

1.6 : TextBlob

TextBlob est une libraire pour traiter des données textuelles,elle offre une API pour un accès direct via des fonctions pour faire de l’analyse des sentiments,classification, étiquetage morpho-syntaxique (Part of Speech Tagging),etc...

2. Dataset

Dans cette partie nous allons aborder sur le dataset utilisé plus en détail ainsi que tous les prétraitements qui ont été utilisés afin de le préparer pour les algorithmes choisis pour l'entraînement afin de pouvoir prédire le sentiment des futures tweets ainsi que des mots.

Le dataset qui a été retenu est issu de Kaggle (inclure lien), la particularité de ce dataset est qu’il est mis à jour chaque semaine et c’est un dataset qui est utilisé dans des compétitions Kaggle.Le dataset contient près de 1 Milliard de tweets et surtout il est mis à jour constamment depuis le début de la pandémie jusqu'à aujourd'hui.

Depuis le scandale de Cambridge Analytica ,les grands groupes technologiques tels que Facebook et Twitter ou Amazon ont changé de politique vis à vis de l’exploitation de leur données et Twitter n’en fait pas exception, en effet désormais les tweets ne sont désormais plus accessible en clair en dehors de l’application Twitter et pour les miner il faut désormais passer par l’API de Twitter pour les exploiter.

Pour pouvoir être capable d'exploiter les tweets,le dataset sur Kaggle est un fichier tabulaire CSV qui contient 2 colonnes: les ids des tweets ainsi que leur polarité via TextBlob.

Pour notre projet nous avons extrait les id des tweets pour les passer vers un package appelé Hydrator pour pouvoir extraire le dataset, le résultat de cette opération est un dataset de 10.000 tweets et qui contient 34 colonnes.

Ce dataset tel qu'il est a été divisé en 2 fichiers qu'on a dénommé train et test,c'est a dire que train contient plus de 5000 tweets ainsi que le fichier test.Comme nous n'avons pas besoin de tous ces détails pour la construction de notre modèle de classification binaire, nous avons supprimé plusieurs colonnes.

Ensuite nous avons procédé à la labellisation du dataset, en effet le dataset n'étant pas labellisé,il a fallu trouver un moyen de labelliser automatiquement nos tweets pour la partie entraînement et cela a été rendu possible grâce à l'utilisation de TextBlob.

La polarité résultante est une valeur comprise entre -1 jusqu'à +1, nous avons donc besoin de normaliser cette valeur pour qu'elle corresponde à une valeur comprise entre 0 et 1 strictement,nous avons donc établi cette règle arbitraire comme suit

Figure 29 : Normalisation des labels en 2 classe: 0 ou 1

La résultante est une colonne qui contient des valeurs binaires 0 ou 1 dans la partie entraînement de notre dataset.Cette partie étant finis,nous avons au final 3 fichiers dont le format étant un csv:
Train,Test et Combi (qui est le diminutif de "combinaison" des 2 fichiers).
La raison de l'existence de combi est que pour faire les prétraitements,il est plus pratique de combiner les deux fichier train et test en même temps pour les faire.

3.Classificateurs et Extracteurs

Dans cette partie nous allons aborder nos classificateurs(Algorithmes) et nos extracteurs de caractéristique (Feature Extraction).

L'extraction des caractéristiques a pour but de réduire le nombre de caractéristique présent déjà dans un dataset et pour faire cela, on a besoin d'en créer de nouveau en utilisant ceux qui existent déjà auparavant, les nouveaux doivent pouvoir résumer toutes les informations contenues dans les originaux.

Concernant les Classificateurs,nous allons utiliser 4 algorithmes: Logistic Regression, SVM , Random Forest et XGBoost.

Concernant les Extracteurs nous allons en utiliser 4 également: Bag of Words, TF-IDF,Word2Vec et Doc2Vec.

Nous allons au final choisir un combo entre les 4 algorithmes et les 4 extracteurs pour choisir la meilleure combinaison possible en utilisant la métrique F1.

Voici un exemple de code utilisé pour Bag of Words ou Sac de mots en français.

```
#Bag of Words
bow_vectorizer = CountVectorizer(max_df=0.90, min_df=2, max_features=1000, stop_words='english')
bow = bow_vectorizer.fit_transform(combi['text'])
```

Figure 30 : Fonction pour appliquer Bag of Words sur la colonne text.

Nous allons finir cette partie par l'entraînement des algorithmes en utilisant les extracteurs,nous allons faire une combinaison entre les deux a chaque fois.

Voici un exemple de notre meilleur combinaison,c'est a dire Random Forest et Bag of Words

```
#RANDOM FOREST
from sklearn.ensemble import RandomForestClassifier
```

```
#BOW FEATURES (the best combo)
rf = RandomForestClassifier(n_estimators=400, random_state=11).fit(xtrain_bow, ytrain)
prediction = rf.predict(xvalid_bow)
print(classification report(yvalid, prediction))
```

Figure 31 : Fonction Random Forest avec Bag of Words

4.Résultat Expérimentaux :

Pour juger de la bonne performance de nos modèles, nous allons utiliser les métriques d'évaluation, précédemment définies et interpréter les résultats.
En commençant par la matrice de confusion qui compte les valeurs TP, TN, FP et FN, nous pourrons ensuite calculer l'exactitude(Accuracy), la précision (Precision), le rappel(Recall) et le score F1 du modèle de classification.
L'ensemble d'apprentissage représente 70 % de l'ensemble de données, tandis que les 30 % restants sont utilisés comme ensemble de test.

Voici la matrice de confusion de notre meilleur modèle :

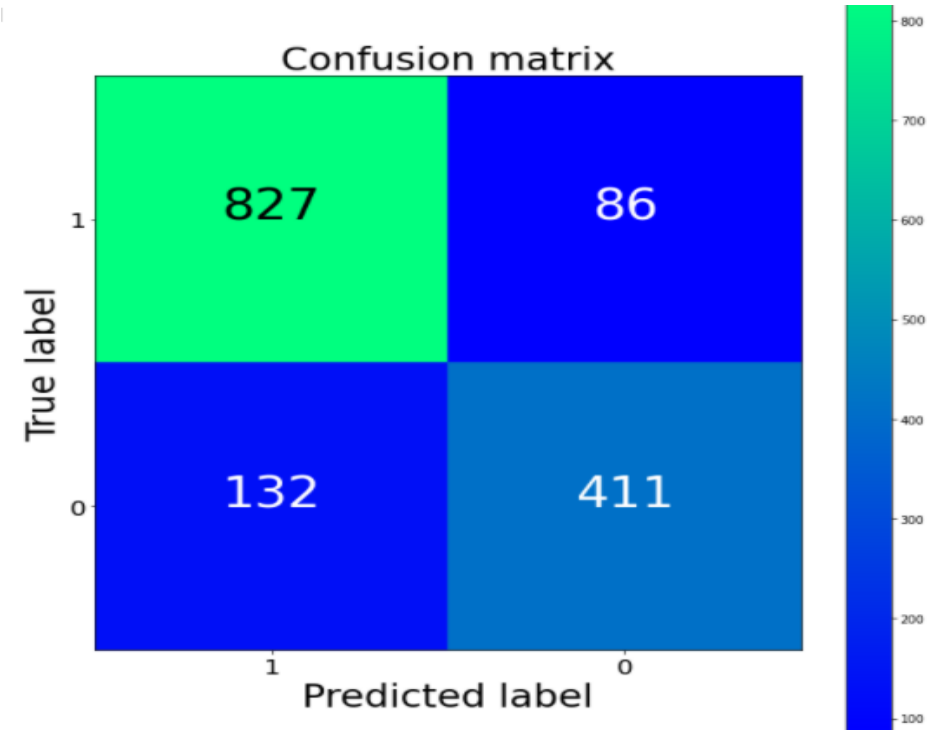

Figure 32 : Matrice de confusion de Random Forest avec Bag Of Words

Le nombre de tweets positifs qui ont été correctement prédits par notre modèle est de 827.
Le nombre de tweets négatifs correctement prédits par notre modèle est de 411.
Le nombre de tweets positifs que notre modèle a prédit négatifs est de 86.
Le nombre de tweets négatifs que notre modèle a prédit positifs est de 132.

Le rapport de classification suivant dans le tableau reprend l'exactitude, la précision, le rappel et le score F1 du modèle.

Metric	Accuracy	Précision	Recall	F1 score
Score	0.85	0.83	0.76	0.79

Tableau 10 : Rapport de classification de RF et Bag of Words

5.Sélection du modèle :

Maintenant que nous avons entraîné et évalué avec succès le modèle de régression logistique, le modèle de machine à vecteurs de support (SVM), le modèle de forêt aléatoire(Random Forest) et le modèle XG Boost en utilisant le Sac de Mots (Bag of Words) ,TF-IDF et Word2Vec et Doc2Vec, nous allons passer à la phase de sélection du modèle.

Dans cette partie, nous allons sélectionner la meilleure combinaison qui a obtenu les meilleures performances en suivant la métrique F1 .
Nous avons résumé les résultats des modèles de la section précédente dans le tableau ci-dessous

	Modèle	Précision	Recall	F1-Score
BOW	LR	0.64	0.83	0.72
	SVM	0.66	0.82	0.73
	RF	0.83	0.76	**0.79**
	XGBoost	0.83	0.73	0.78
TF-IDF	LR	0.56	0.90	0.69
	SVM	0.67	0.84	0.75
	RF	0.90	0.70	0.78
	XGBoost	0.79	0.73	0.76
Word2Vec	LR	0.56	0.82	0.66
	SVM	0.59	0.79	0.68
	RF	0.90	0.56	0.69
	XGBoost	0.80	0.70	0.75
Doc2Vec	LR	0.44	0.78	0.56
	SVM	0.41	0.82	0.55
	RF	0.78	0.44	0.56
	XGBoost	0.76	0.59	0.66

Tableau 11 : les principaux résultats

Les résultats montrent des scores proches si on utilise la mesure F1 comme référence,Random Forest dépasse de très peu le modèle de XGBoost quand on utilise Bag of Words comme extracteur et XGBoost arrive aussi en 3eme position quand on utilise le modèle TF-IDF, on note d'ailleurs que le meilleur modèle a un taux élevé de précision avec un taux de Rappel plus bas a 73%.

Le meilleur score de Rappel a été réalisé par l'algorithme de Logistic Regression avec un modèle de TF-IDF ,quant à la précision de 90%,deux combinaisons arrivent ex-aequo: le meilleur algorithme de Random Forest mais avec deux modèles: TF-IDF et Word2Vec.

On notera que Doc2Vec a réalisé les plus faibles scores de notre série d'évaluation.

Il apparaît clairement dans la figure d'auparavant que la meilleure combinaison est l'algorithme Random Forest avec Bag of Words comme extracteur, nous allons utiliser cette combinaison pour créer notre prédiction dans l'application finale.

6.Interface utilisateur :

Nous allons présenter dans cette partie l'interface utilisateur final qui permet de tester nos modèles,étant donnée que notre meilleur modèle était la combinaison entre Random Forest et Bag of Words avec un score F1 de 79%,nous allons l'utiliser pour prédire les sentiments positifs ou négatifs d'un utilisateur selon le choix qu'il fait: Soit en introduisant un mot donné ou bien une liste de tweets pour prédire le sentiment global des tweets, le tout sera visuellement illustré par 3 visualisation

- Diagramme à barres (Bar Chart)
- Courbe de Subjectivité
- Courbe de Polarité

Pour faciliter la compréhension de l'application, un graphique de structure arborescente a été créé pour illustrer comment l'utilisateur peut naviguer à travers l'application.

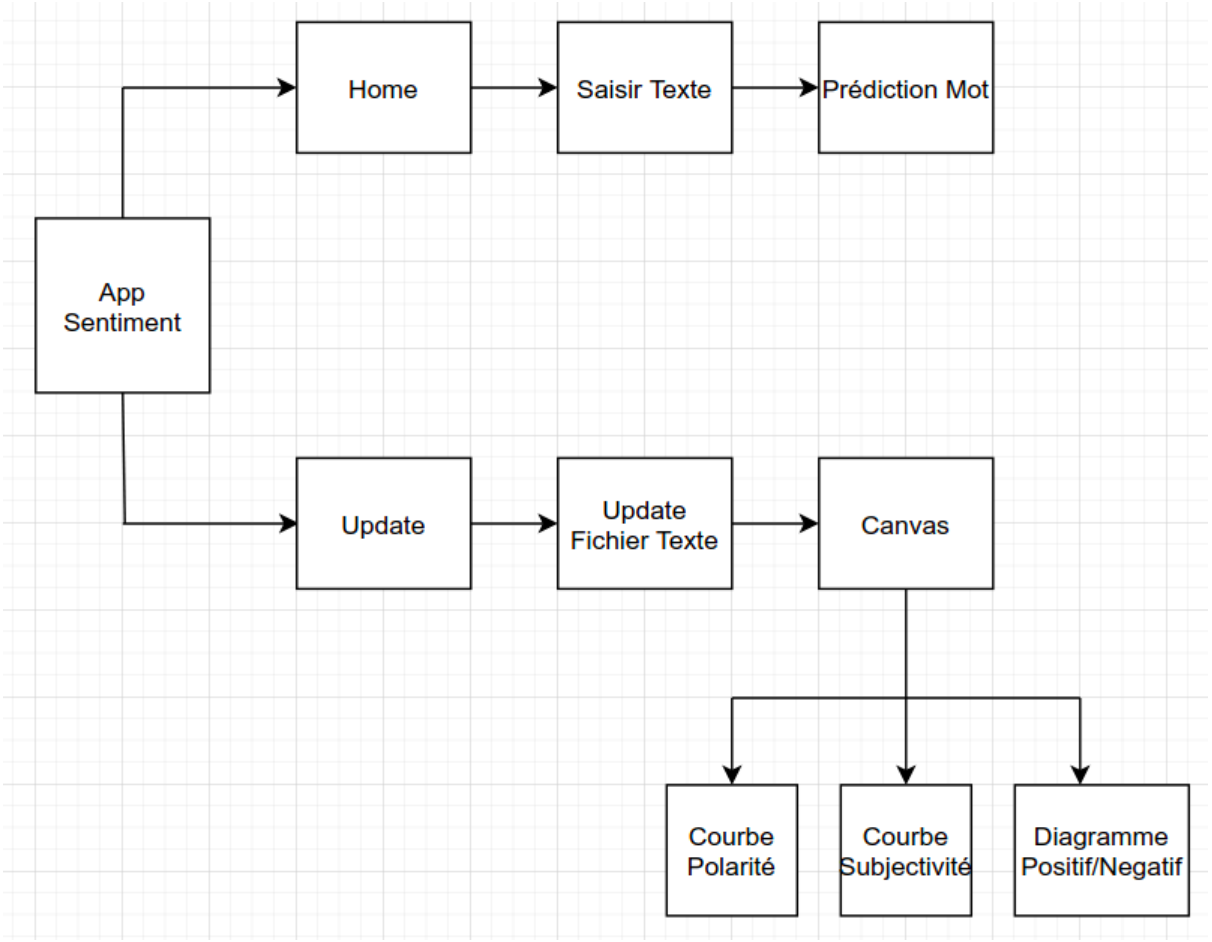

Figure 33 : la structure arborescente de navigation de l'application

L'utilisateur après avoir cliqué sur le lien de l'application est directement renvoyé sur la page d'accueil de l'interface

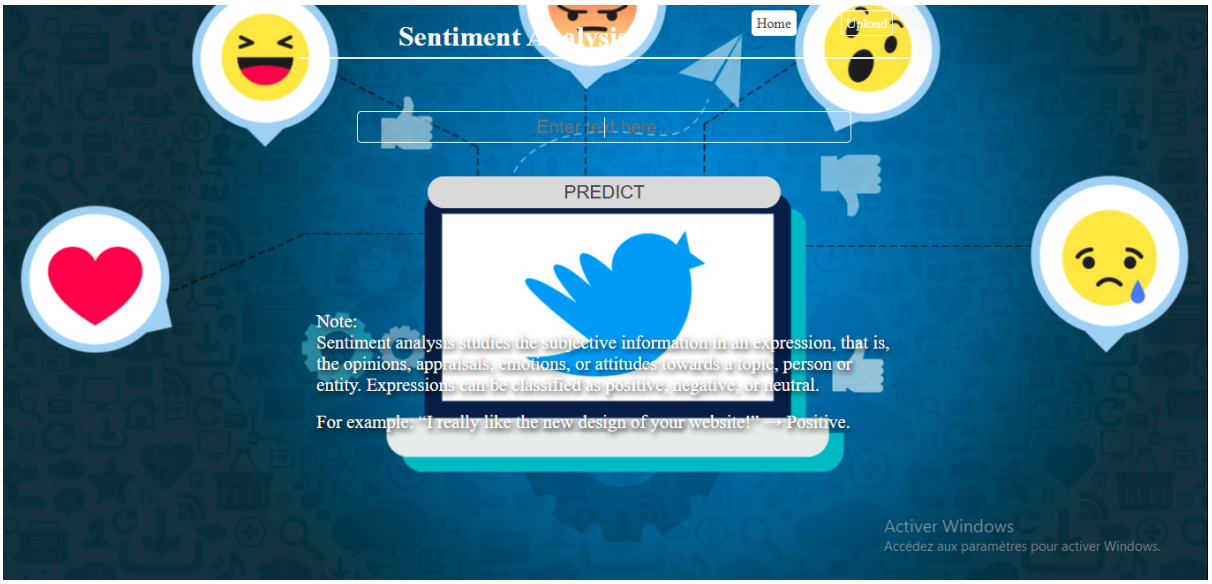

Figure 34 : L'interface principale ou la page Home.

L'utilisateur peut également télécharger une liste de tweets regroupé dans un fichier .txt dans cette interface

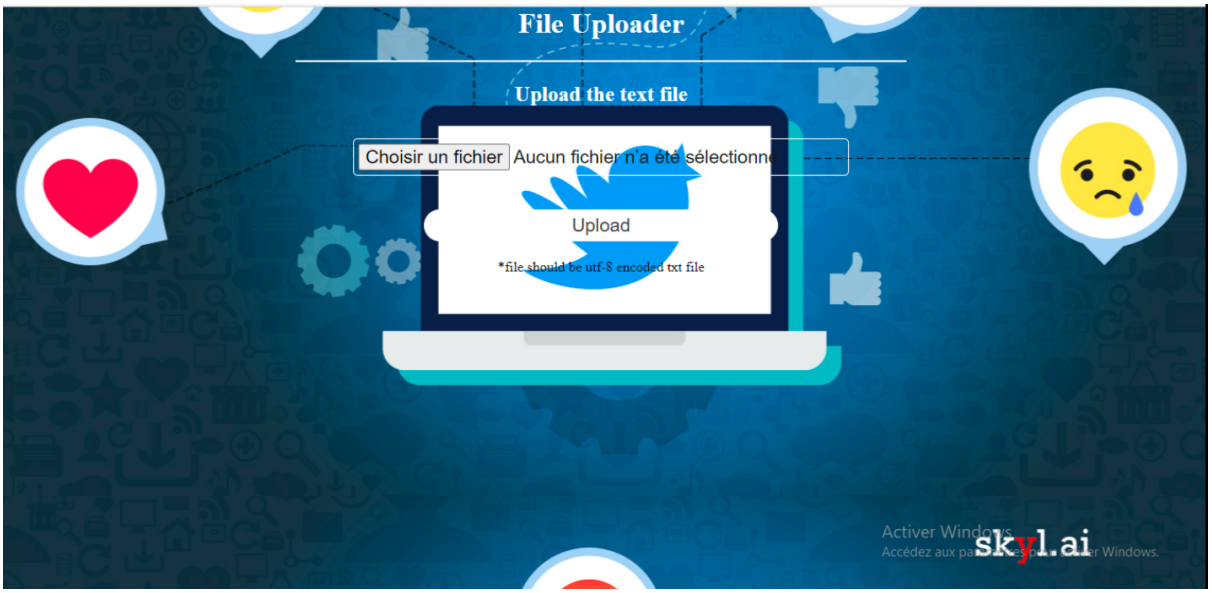

Figure 35 : l'interface pour télécharger une liste des tweets

Les résultats seront affiché dans cette dernière interface
Nous avons en premier lieu une courbe de subjectivité des tweets en donnée,l'affichage se fait via Canvas.Js

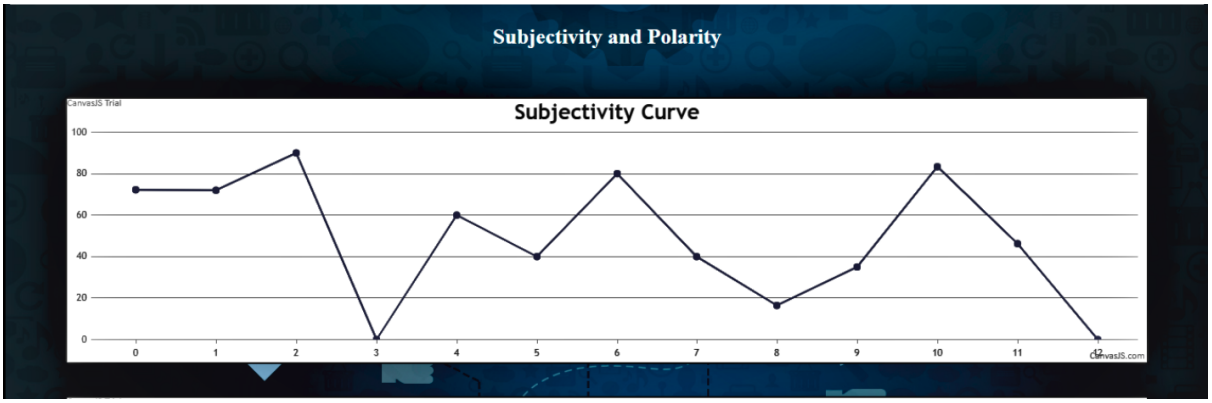

Figure 36 : la courbe de subjectivité des tweets

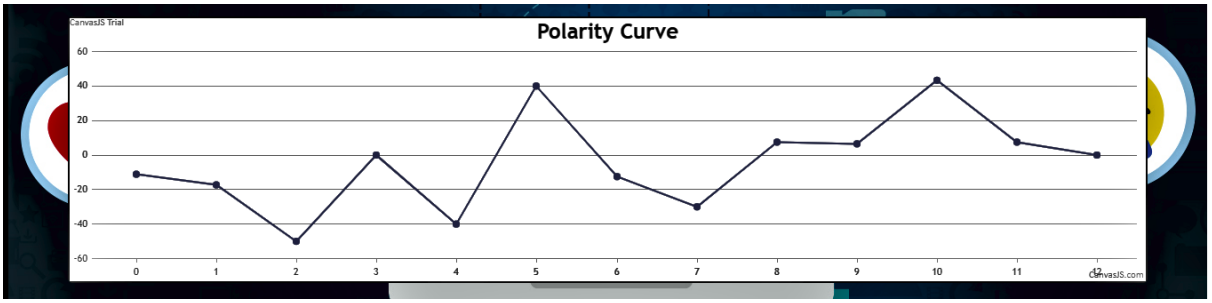

Figure 37 : la courbe de polarité des tweets

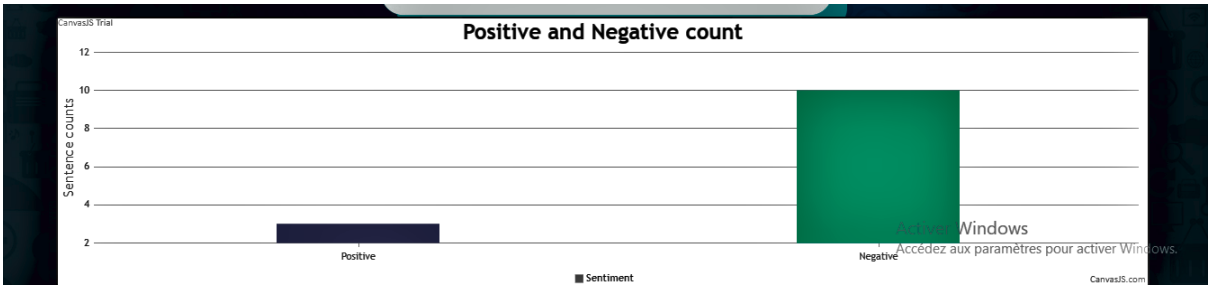

Figure 38 : le diagramme à bande pour calculer le nombre de tweets positifs et négatifs.

Conclusion

Dans ce chapitre, nous avons présenté l'essentiel de notre travail qui consiste à créer un système d'analyse d'opinion pour détecter les sentiments sur Twitter,pour l'implémentation nous avons utilisé 4 algorithmes ainsi que 4 extracteurs de caractéristiques pour choisir la meilleur combinaison pour notre système en utilisant la mesure F1 pour les discriminer entre eux.

Nous avons par ailleurs choisis plusieurs bibliothèques Python pour examiner les tweets,faire les prétraitement et enfin implémenter notre solution ainsi que la visualisation qui s'accompagne.

Conclusion Générale

Synthèse

Dans ce travail,nous avons exploré le domaine de l'Analyse des Sentiments qui est considéré actuellement comme un domaine d'actualité et qui connaît une évolution majeure.

Afin d'atteindre nos résultats nous avons passé un bon moment a lire les publications,articles pour pouvoir prétendre à conceptualiser notre problème et enfin appliquer du Machine Learning pour pouvoir arriver à notre solution finale.

L'idée initiale était de créer un modèle de classification multiclasse sur un dataset purement Algérien mais nous nous sommes confronté à deux problèmes majeurs: les données ainsi que la faiblesse d'introduction de Twitter en Algérie, en effet les réseaux sociales majeures en Algérie sont Facebook et Instagram,bien devant Twitter ou il existe une communauté solide mais fermé.

Le manque de données a été un élément majeur pour changer notre approche, en effet les données dites "historical" de Twitter sont payant,il aura fallu soit payer pour avoir accès à l'API historical de Twitter pour avoir des données qui remontent au début de la Pandémie pour pouvoir prétendre faire une analyse complète ou soit se restreindre à le faire a la moitié de l'année 2021 ou l'impact a changé et surtout déployer un script pour miner une quantité de tweets sur une période d'au moins 2 mois minimum.

C'est alors que notre approche a changé pour se focaliser sur un dataset déjà existant et tenu à jour régulièrement pour pouvoir miner des données sur une période de temps bien précise,d'où au recours au dataset sur Kaggle.

Ayant résolu le problème des données sur notre problématique qui est le Covid_19,nous nous sommes intéressé à l'application du Machine Learning mais faute de temps,nous nous sommes restreint à l'appliquer sur un problème binaire: les tweets positifs ou négatifs ainsi que leur polarité et niveaux de subjectivité.

Nous avons donc utilisé 4 algorithmes ainsi que 4 extracteurs de features pour les comparer entre eux pour pouvoir choisir la meilleure combinaison pour notre solution finale.

Dans ce mémoire nous proposons un système de classification subjective des opinions des utilisateurs sur Twitter basé sur 2 classes: Positive et Négatif.

Perspectives

Notre système étant en phase de développement et est absolument loin d'être complet, pour l'amélioration nous proposons:

- Etablire une classification multiclasse
- Etablire une analyse plus fine des sentiments comme les sentiments majeurs telle que la haine,l'amour,la joie,la détresse,..etc
- L'analyse des tendances en utilisant les hashtags par exemple
- Déployer les analyses en utilisant un vrai dashboard et récupérer les données en temps réel en utilisant le Streaming via l'API de Twitter.

Bibliographie

[1] S.Bird,E.Klein,E.Loper, *Natural Language Processing with Python*, O.Reilly,2009.

[2] J.Sterne,*Artificial Intelligence for Marketing: Practical Applications* ,Wiley , 2017.

[3] B.Liu, *Sentiment Analysis: Mining Opinions,Sentiments and Emotions* ,Cambridge University Press, 2015.

[4] Peter D.Turney, *Thumbs Up or Thumbs Down ? Semantic Orientation Applied to Unsupervised Classification of Reviews*,ACL,2002.

[5] E.Riloff,J.Wiebe,*Learning extraction patterns for subjective expression*.In Proceedings of the 2003 conference on empirical methods in NLP (EMNLP '03). Associations for Computational Linguistics,USA; 105-112. DOI: https://doi.org/10.3115/1119355.119369

[6] V.Hatzivassiloglou,K McKeown,1997,*Predicting the semantic orientation of adjectives*,In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (ACL '98/EACL '98).Association for Computational Linguistics,USA,174-181. DOI: https://doi.org/10.3115/976909.979640

[7] "*Manifestations,Covid-19...Twitter bat son record de téléchargement en une semaine*",[Enligne],Disponible : https://www.lefigaro.fr/[Accès: 08-07-2021]

[8] "*More than 87.000 scientific papers on coronavirus since the pandemic"*,[En ligne],Disponible: https://news.osu.edu/, [Accès: 03-08-2021].

[9] "*Dimensions*",[Enligne]Disponible:https://app.dimensions.ai,[Accés: 04-07-2021].

[10] M.Sethi S.Pandey,P.Trar et P.Soni,*"Sentiment Identification in COVID-19 Specific Tweets"*,2020 International Conference on Electronics and Sustainable CommunicationSystems(ICESC),2020,pp,509-516, doi:10.1109/ICESC48915.2020.9155674

[11] M.Alanezi,N.Hewahi,*Tweets Sentiment Analysis during COVID-19 Pandemic*,2020 International Conference on Data Analytics for Business and Industry: Way towards a Sustainable Economy (ICBABI),IEEE,DOI: 10.1109/ICDABI51230.2020.9325679

[12] Mansoor, M., Gurumurthy, K., & Prasad, V. R. (2020). *Global Sentiment Analysis Of COVID-19 Tweets Over Time*. *arXiv preprint arXiv:2010.14234*.

[13] "*Covid Twitter Sentiment Analysis Dataset*",[En ligne],Disponible: https://www.kaggle.com/mejbahahammad/covid-twitter-sentiment-analysis-datasets [Accès:02-06-2021].

[14] "*Developer Policy*", [En ligne] , Disponible: https://developer.twitter.com/en/developer-terms/policy[Accès: 12-03-2021]

[15] Hadley Wickham,*"Tidy Data"*, Journal of Statistical Software,2014.

[16] "*Regular Expressions Operations*", [En ligne] , Disponible: docs.python.org/3/library/re.html[Accès:12-03-2021]

[17] "*What is Exploratory Data Analysis ?*", [En ligne] , Disponible: https://towardsdatascience.com/exploratory-data-analysis [Accès:04-042021].

[18] A.Shamoo,D.Resnik,*"Responsible Conduct of Research"*,Oxford University Press,2009.

[19] "*Wordcloud for Python documentation*" , [Enligne] , Disponible: https://amueller.github.io/word_cloud/ [Accès:17-08-2021]

[20] "*NLP: Word Embeddings Techniques Demystified*",[En ligne],Disponible: https://towardsdatascience.com/, [Accès:02-09-2021]

[21] "*Scikit-learn*",[En ligne],Disponible:https://scikit-learn.org/stable/,[Accès:15-06-2021]

[22] "*Tf-idf Vectroizer*" , [En ligne] , Disponible: https://scikit-learn.org, [Accès:17-06-2021]

[23] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances *in neural information processing systems* (pp. 3111-3119).