République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Informatique

Département d'intelligence artificielle et science des données.

Mémoire de Licence

Filière: Informatique

Spécialité:Informatique Académique

Analyse d'Opinion sur la Guerre en Ukraine sur les Réseaux Sociaux

Sujet Proposé par :
Mme CHALLAL Zakia :

Soutenu le :../06/2022 **Présenté par :MEKKI Feriel**

Devant le jury composé de: **KHAOUNI Amel**

M FERGUENE Farid **Président**
M STITRA Insaf **Membre**

Binôme n° :057 / 2022

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux de nous avoir donné la force et la patience d'accomplir ce travail.
En second lieu nous tenons à exprimer nos plus sincères remerciements à notre encadreur Mme Z. CHALLAL,
pour sa disponibilité et ses précieux conseils.
Nous tenons aussi à remercier nos chers parents d'avoir été une telle source de courage et de persévérance, que Dieu vous garde pour nous.
Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Merci

Résumé

L'analyse d'opinions est un domaine en croissance rapide. Il existe un nombre de sites web

2.0 appelés réseaux sociaux qui offrent différentes plateformes aux utilisateurs afin de donner

des avis sur un sujet par des commentaires, des publications et des réactions.

Une méthode précise de prédiction des

opinions pourrait nous permettre d'extraire les avis des internautes et de prédire leurs préférences.

Dans ce projet, nous nous intéressons aux différents algorithmes disponibles pour l'exploration

d'opinions. Dans ce contexte, nous avons procédé à une étude bibliographique se rapportant

aux méthodes utilisées dans l'étude du comportement des internautes.

On a mis en place une approche

data mining basée sur une technique d'apprentissage supervisé et non supervisé qui identifie les textes

subjectifs (porteurs d'opinions), puis les caractérise en polarité positive, négative ou neutre.

Mots-clé : Analyse d'opinions, réseaux sociaux,prétraitement,machine learning, Big Data

Table des Matières

Introduction

A l'opposé des faits qui sont des énoncés objectifs et universels sur les entités et les événements dans ce monde, l'opinion est un avis, un jugement personnel que l'on s'est forgé sur une question ou un sujet en discussion et qui ne relève pas de la connaissance purement rationnelle.

L'internet social a récemment explosé avec la disponibilité de documents textuels exprimant des opinions ou des sentiments, comme dans les groupes de discussion, les blogs, les forums et autres sites Web consacrés aux critiques de produits. Les opinions disponibles sur Internet ont un impact considérable sur les internautes. L'enquête de Pang et Lee (2008) montre que 80 % des internautes ont recherché des avis sur un produit, et qu'ils sont prêts à payer le double pour un produit dont l'avis est plus favorable qu'un autre. D'où la prise de conscience de l'importance de l'opinion sur le web, et le grand intérêt de l'analyse d'opinion ces dernières années. L'analyse d'opinion est une technique d'analyse de texte qui utilise la linguistique informatique et le traitement du langage naturel pour identifier et extraire automatiquement un sentiment ou une opinion à l'intérieur d'un texte (positif, négatif, neutre, etc.)

Dans le cadre de notre projet de fin d'études de licence, nous nous intéressons à l'analyse des opinions sur la guerre en Ukraine sur les réseaux sociaux. Pour ce faire, nous allons procéder à l'extraction des données à l'aide du réseau social Twitter, ces données seront ensuite pré traitées, et utilisées pour construire un modèle d'apprentissage automatique

Notre mémoire est structuré principalement en trois chapitres présentés comme suit :

Chapitre 1 – État de l'art : Dans cette partie, nous expliquons en premier lieu les concepts de fouille d'opinions, ensuite, nous introduirons les notions d'apprentissage automatique.

Chapitre 2 – Conception de l'approche : Dans ce chapitre, nous décrivons notre approche de conception, suivie par la modélisation de l'application.

Chapitre 3 – Implémentation et expérimentation :

Nous concluons ce mémoire par une présentation de l'aspect technique et de l'implémentation de l'outil en décrivant les différents modules utilisés.

Chapitre 1

État de L'ART

1.1 introduction

Ce chapitre est organisé comme suit : nous définissons en premier lieu l'analyse d'opinion, ensuite nous citons ces approches, dont l'apprentissage automatique, nous introduisons enfin l'étape de pré traitement.

1.2 Définition de l'analyse d'opinion

Les opinions sont subjectives et reflètent les sentiments des gens ou des perceptions au sujet des entités et des événements. L'analyse d'opinion est un domaine de recherche en plein essor, il se définit comme l'analyse automatisée des informations en vue de déterminer si une opinion a une tonalité plutôt positive, négative ou neutre
La fouille d'opinion se compose de plusieurs tâches, qu'il est utile ou non de mettre en œuvre selon les applications visées. :
– détection de la présence ou non de l'opinion ;
– classification de l'axiologie de l'opinion (positif, négatif, neutre) ;
– classification de l'intensité de l'opinion ;
– identification de l'objet de l'opinion
– identification de la source de l'opinion

1.3 Approche d'analyse d'opinion

[1]Dans la revue de littérature, on retrouve en général deux approches d'analyse de sentiments :

-l'approche basée sur un lexique (Lexicon based approach) qui utilise un lexique prédéfinie.

-l'approche basée sur une machine d'apprentissage (Machine Learning based approach) qui utilise la classification subjective des textes à partir d'un large ensemble de données.

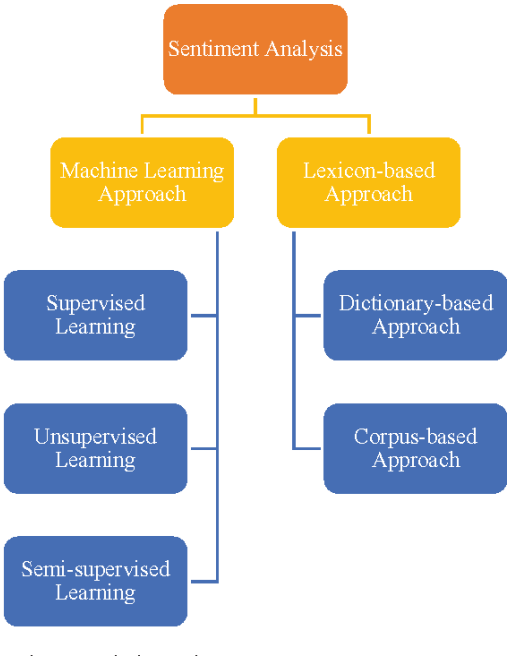

Figure1.1 -Diagramme d'approche d'analyse des sentiments

1.3.1 L'approche d'analyse des sentiments basée sur un lexique

L'approche d'analyse des sentiments basée sur un lexique est celle qu'on retrouve le plus dans la revue de littérature. Elle aide à identifier la polarité d'un texte en se servant de deux catégories de mots prédéfinis et pondérés, appelé lexique ou dictionnaire. Elle identifie tous les mots positifs ou négatifs au sein d'un texte. Le dictionnaire est constitué d'un petit ensemble de mots d'opinion subdivisés en deux catégories. L'une des catégories comporte des mots dont la terminologie est plus positive, tandis que l'autre catégorie regroupe les mots associés à un sentiment plus négatif. Pour établir un dictionnaire, ils commencent généralement par les mots les plus intuitifs et qui expriment un sentiment positif, le nombre de ces mots est amplifié par leurs synonymes constitue la catégorie positive. Une catégorie de mots négatifs peut être automatiquement constituée à partir des antonymes de mots de la catégorie positive, puis elle est amplifiée par d'autres mots jugés négatifs. À chaque mot est associée une pondération positive ou négative. La somme des mots est soit positive ou négative et représente une évaluation globale du sentiment dans le texte.

L'algorithme utilisé dans l'approche basée sur un lexique peut être exprimé comme suit :

1. Initialiser la note totale du sentiment s 0
2. Pour chaque mot du texte, vérifier la présence dans le lexique :
   (a) Si le mot est présent :
      i. Si le mot est positif, alors : s = s + w
      ii. Si le mot est négatif, alors : s = s – w
3. Regarder le sentiment total du texte
   (a) Si s > au seuil, alors le sentiment du texte est positif
   (b) Si s < au seuil, alors le sentiment du texte est négatif

L'approche basée sur un lexique présente un inconvénient majeur qui est l'incapacité de trouver des mots d'opinion avec des orientations spécifiques au domaine et au contexte.

1.3.2 L'approche basée sur l'apprentissage automatique

[1] L'approche d'analyse de sentiments basée sur une machine d'apprentissage (Machine Learning based approach) utilise des algorithmes statistiques et des fonctionnalités linguistiques de classifications. À partir d'un ensemble de données appelé « data set », les algorithmes sont entraînés et peuvent, à partir de ces modèles, faire des prédictions sur d'autres données qu'ils peuvent rencontrer à l'avenir. L'approche basée sur une machine d'apprentissage est subdivisée en méthodes d'apprentissages supervisées et non-supervisées et apprentissage par renforcement.

1-Apprentissage supervisé

[2] Les méthodes dites supervisées sont celles qui utilisent de grands ensembles de données labellisées ; elles nécessitent la formation de deux ensembles de données : un ensemble d'apprentissage(ou training set) et un autre de test.
Le training set est utilisé par un classificateur automatique pour apprendre les caractéristiques de différenciation des textes afin d'entrainer le système.
L'ensemble de données de test est quant à lui utilisé pour vérifier la façon dont ce classificateur (ou l'algorithme de classification) performe.

Il existe plusieurs algorithmes de classification dans les techniques dites supervisées :

Regression logistique :

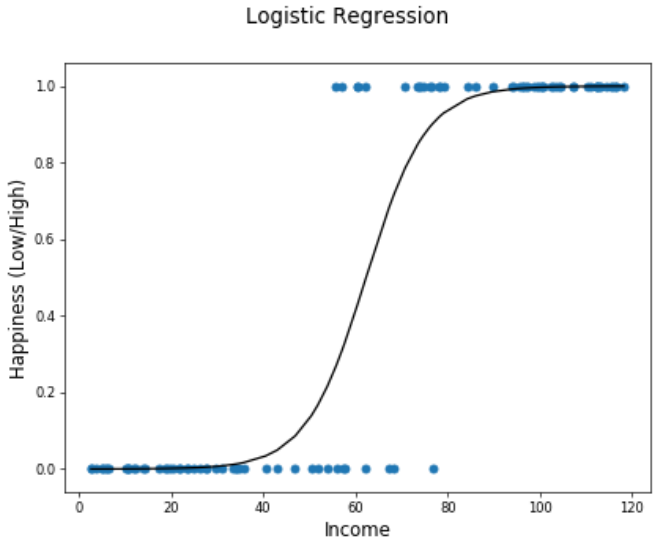

Figure1.2 -Regression Logistique[3]

La régression logistique utilise la fonction sigmoïde ci-dessus pour renvoyer la probabilité d'une étiquette. Il est largement utilisé lorsque le problème de classification est binaire — vrai ou faux, gagnant ou perdant, positif ou négatif... La fonction sigmoïde génère une sortie de probabilité. En comparant la probabilité avec un seuil prédéfini, l'objet est affecté à une étiquette en conséquence.

Arbre de decision :

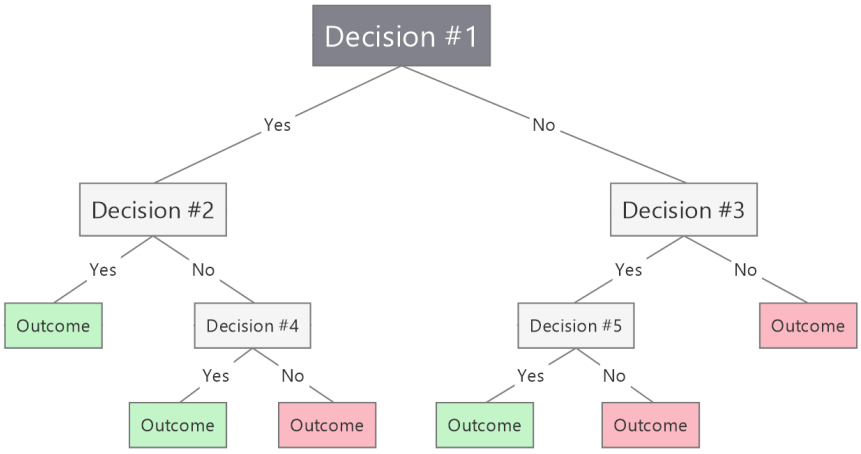

Figure1.3 -Arbre de Décision[4]

L'arbre de décision construit des branches d'arbre dans une approche hiérarchique et chaque branche peut être considérée comme une instruction if-else. Les branches se développent en partitionnant l'ensemble de données en sous-ensembles basés sur les caractéristiques les plus importantes. La classification finale se produit aux feuilles de l'arbre de décision.

Forêt aléatoire :

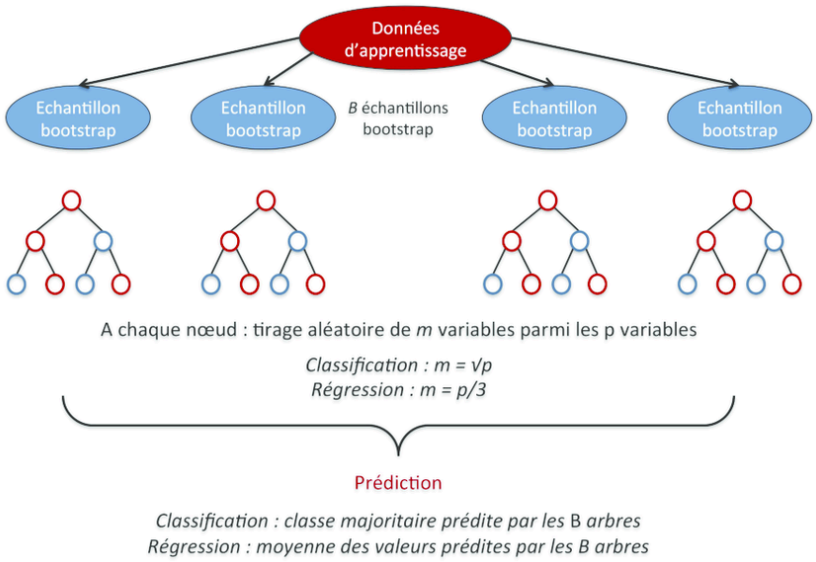

Figure1.4 -Foret Aléatoire [5]

Comme son nom l'indique, la forêt aléatoire est une collection d'arbres de décision. Il s'agit d'un type courant de méthodes d'ensemble qui agrègent les résultats de plusieurs prédicteurs. La forêt aléatoire utilise en outre une technique d'ensachage qui permet à chaque arbre d'être formé sur un échantillonnage aléatoire de l'ensemble de données d'origine et de prendre le vote majoritaire des arbres. Comparé à l'arbre de décision, il a une meilleure généralisation mais moins interprétable, en raison de plus de couches ajoutées au modèle.

K plus proche voisins :

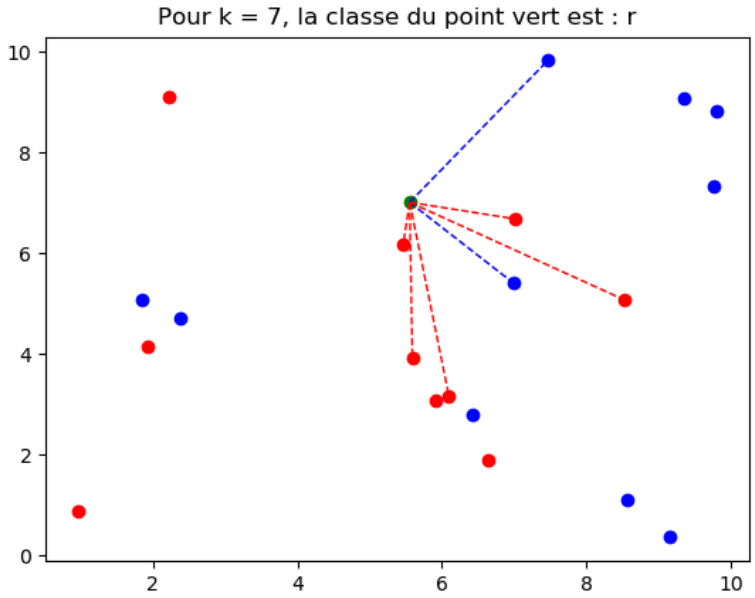

Figure1.5 -K plus proches voisins[6]

Vous pouvez penser à k algorithme du plus proche voisin comme représentant chaque point de données dans un espace à n dimensions - qui est défini par n caractéristiques. Et il calcule la distance entre un point à un autre, puis attribue l'étiquette des données non observées en fonction des étiquettes des points de données observés les plus proches. KNN peut également être utilisé pour créer un système de recommandation, consultez mon article sur "Filtrage collaboratif pour la recommandation de films" si ce sujet vous intéresse.

Naive bayes :

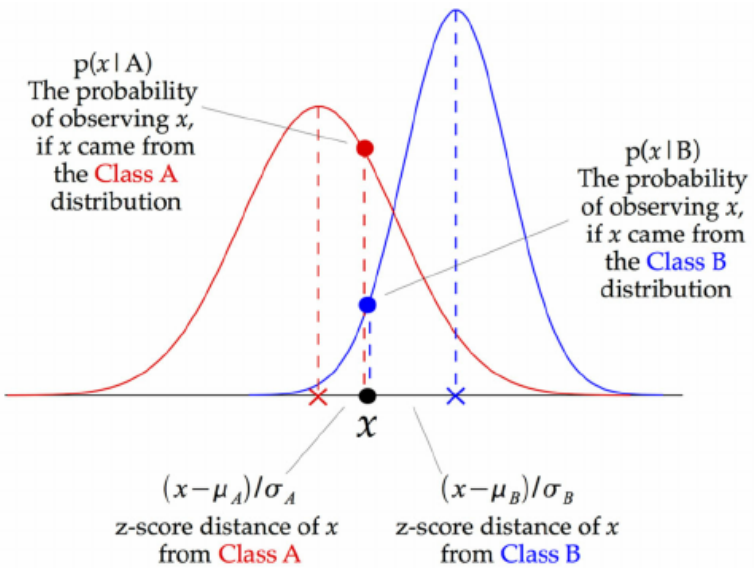

Figure1.6 -Naive Bayes[7]

Naive Bayes est basé sur le théorème de Bayes - une approche pour calculer la probabilité conditionnelle basée sur des connaissances antérieures et l'hypothèse naïve selon laquelle chaque caractéristique est indépendante l'une de l'autre. Le plus grand avantage de Naive Bayes est que, bien que la plupart des algorithmes d'apprentissage automatique reposent sur une grande quantité de données d'entraînement, il fonctionne relativement bien même lorsque la taille des données d'entraînement est petite.

SVM :

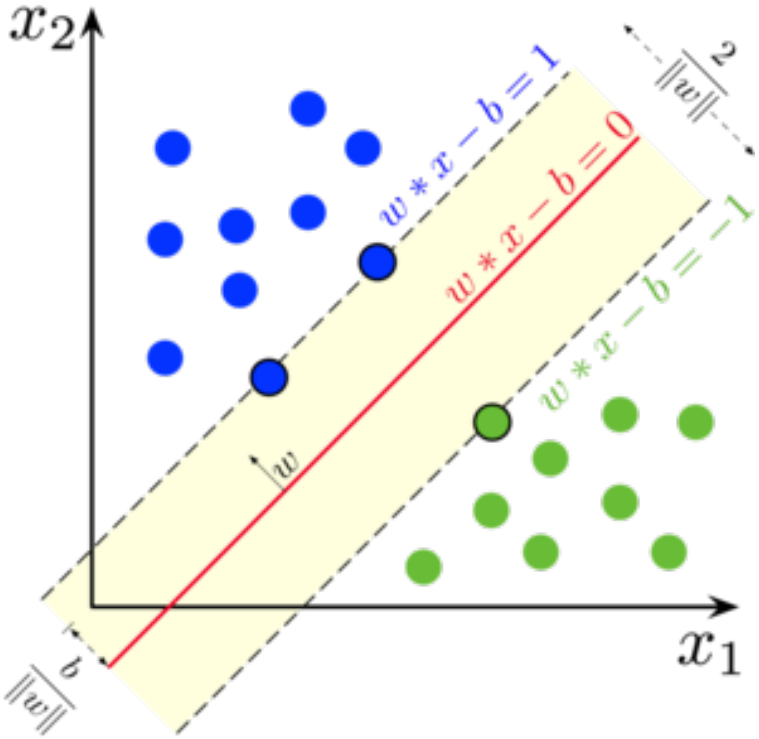

Figure1.7 -SVM[8]

La machine à vecteurs de support trouve le meilleur moyen de classer les données en fonction de la position par rapport à une frontière entre classe positive et classe négative. Cette frontière est connue sous le nom d'hyperplan qui maximise la distance entre les points de données de différentes classes. Semblable à l'arbre de décision et à la forêt aléatoire, la machine à vecteurs de support peut être utilisée à la fois dans la classification et la régression, SVC (classificateur de vecteurs de support) est pour le problème de classification.

2-Apprentissage non-supervisé

[9]À la différence de l'apprentissage supervisé, l'apprentissage non supervisé est celui où l'algorithme doit opérer à partir d'exemples non étiquetés (unlabelled data). En effet, dans ce cas de figure, l'apprentissage se fait de manière entièrement indépendante. Des données sont alors renseignées à la machine sans qu'on lui fournisse des exemples de résultats. On attend donc de la machine qu'elle crée elle-même les réponses grâce à différentes analyses et au classement des données.

Regroupement(clustering)

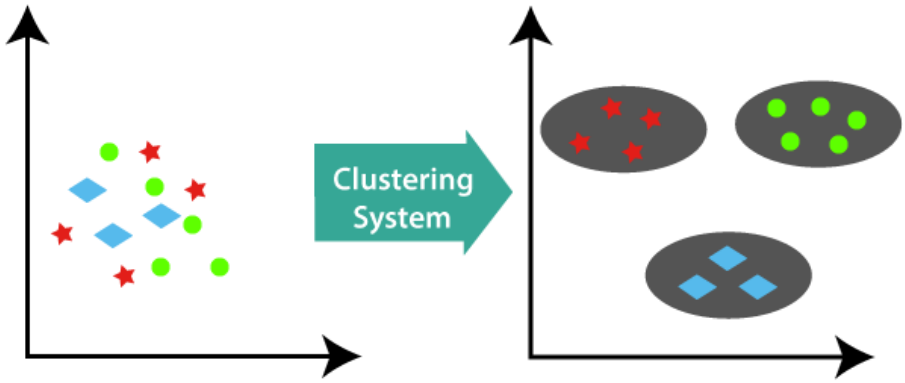

Figure1.8 -Regroupement[10]

Le clustering (regroupement) est un concept important en matière d'apprentissage non supervisé. Il s'agit principalement de trouver une structure ou un modèle dans une collection de données non catégorisées. Les algorithmes de clustering traitent les données et trouveront des clusters (groupes) naturels s'ils existent.
Il existe différents types de clustering :

Hierarchical Clustering

Le clustering hiérarchique est un algorithme qui construit une hiérarchie de clusters. Cela commence par toutes les données qui sont affectées à un cluster qui leur est propre. Ici, deux clusters proches vont être dans le même cluster. Cet algorithme se termine lorsqu'il ne reste plus qu'un cluster.

K-means Clustering

K signifie qu'il s'agit d'un algorithme de clustering itératif qui aide à trouver la valeur la plus élevée pour chaque itération. Initialement, le nombre souhaité de clusters est sélectionné. Dans cette méthode de clustering, on doit regrouper les points de données en k groupes. Un k plus grand signifie des groupes plus petits avec plus de granularité de la même manière. Un k inférieur signifie des groupes plus grands avec moins de granularité. La sortie de l'algorithme est un groupe d'"étiquettes". Il attribue un point de données à l'un des k groupes. Dans le clustering k-means, chaque groupe est défini en créant un centroïde pour chaque groupe. Les centroïdes sont comme le coeur du cluster, qui capture les points les plus proches d'eux et les ajoute au cluster.

3-L'apprentissage par renforcement

L'apprentissage par renforcement est un autre type d'apprentissage automatique, où les agents apprennent à prendre des mesures en fonction de leur interaction avec l'environnement, dans le but de maximiser les récompenses. Il ressemble le plus au processus d'apprentissage de l'homme, suivant une approche par essais et erreurs.

1.3.3 Prétraitement

Définition

[11]Les données brutes du monde réel sous forme de texte, d'images, de vidéos, etc., sont désordonnées, et contiennent des erreurs et des incohérences. Les machines aiment traiter des informations claires et ordonnées, elles lisent les données sous forme de 1 et de 0. Il est donc nécessaire de nettoyer et formater ces données avant l'analyse. Le prétraitement (preprocessing) est l'étape du processus d'exploration et d'analyse de données qui prend des données brutes et les transforme en un format pouvant être compris et analysé par l'ordinateur.

De bonnes données prétraitées sont encore plus importantes que les algorithmes les plus puissants, au point que les modèles d'apprentissage automatique entraînés avec de mauvaises données pourraient nuire à l'analyse.

Types de données d'apprentissage automatique

Il existe deux types de données :

1. données numériques : caractéristiques avec des valeurs continues sur une échelle, statistiques ou liées à des nombres entiers. Les valeurs numériques sont représentées par des nombres entiers, des fractions ou des pourcentages.
2. Données categorielles (categorical data) : groupes ou des catégories non numériques

Étapes de prétraitement des données

Évaluation de la qualité des données : Il s'agit d'une examination attentive des données pour avoir une idée de leur qualité globale, de leur pertinence par rapport au projet et de leur cohérence. Il existe un certain nombre d'anomalies de données et de problèmes inhérents à surveiller dans presque tous les ensembles de données, par exemple des données incompatibles, des valeurs aberrantes, ou des données manquantes

Data cleaning : Le nettoyage des données consiste à ajouter des données manquantes et à corriger, réparer ou supprimer des données incorrectes ou non pertinentes d'un ensemble de données.

Transformation des données : La transformation des données dans les formats appropriés dont nous avons besoin pour l'analyse et d'autres processus en aval. Cela se produit généralement dans un ou plusieurs des cas ci-dessous : agrégation, normalisation, discréditation, et génération de hiérarchie de concepts.

Feature sélection : la sélection des caractéristiques est le processus consistant à décider quelles features sont les plus importantes pour l'analyse. Ces fonctionnalités seront utilisées pour former des modèles ML.

Natural Language Processing

Le traitement du langage naturel (NLP) est un domaine de l'intelligence artificielle (IA) qui rend le langage humain intelligible aux machines. La NLP combine la puissance de la linguistique et de l'informatique pour étudier les règles et la structure du langage et créer des systèmes intelligents capables de comprendre, d'analyser et d'extraire le sens du texte et de la parole.

Les principales sous-tâches du NLP sont la tokenisation, la segmentation, la lemmatisation, le stemming, et la suppression des stop words.

Nous définirons ces taches dans le chapitre suivant.

1.4 Conclusion :

Dans ce chapitre nous avons fait un survole sur les principaux fondements, méthodes d'analyse d'opinion et les étapes nécessaire pour notre système en incluant le prétraitement des données, classification d'opinion et la partie apprentissage automatique (machine learning). Dans le prochain chapitre nous allons passer à la partie conception.

Chapitre 2

Conception de l'approche

2.1 Introduction

Précédemment, nous avons abordé les différents concepts d'analyse d'opinion. Dans ce chapitre nous allons proposer un système pour analyser les données provenant des réseaux sociaux sur la guerre en Ukraine. Nous rappelons que notre solution doit permettre aux utilisateurs d'analyser les opinions de la guerre sur twitter en fonction des paramètres choisis, tels que la langue, la localisation et la période .Cette solution sera axée principalement sur des modèles de Machine Learning.

2.2 Identification des besoins

Le rôle principal de notre application consiste en la classification des textes issus des réseaux sociaux en trois classes : négative, neutre ou positive. Pour ce faire, nous devons d'abord extraire ces données et leurs effectuer tous les traitements nécessaires afin qu'elles soient en mesure d'être analysées, enfin nous serons capables de visualiser les résultats de la classification.

2.3 Étapes du processus d'analyse d'opinion

Tout processus d'analyse d'opinion repose sur quatre étapes majeures, à savoir : le data mining, le preprocessing, l'analyse des données et enfin l'interprétation des résultats. Dans ce qui suit, une vue globale sur notre projet :

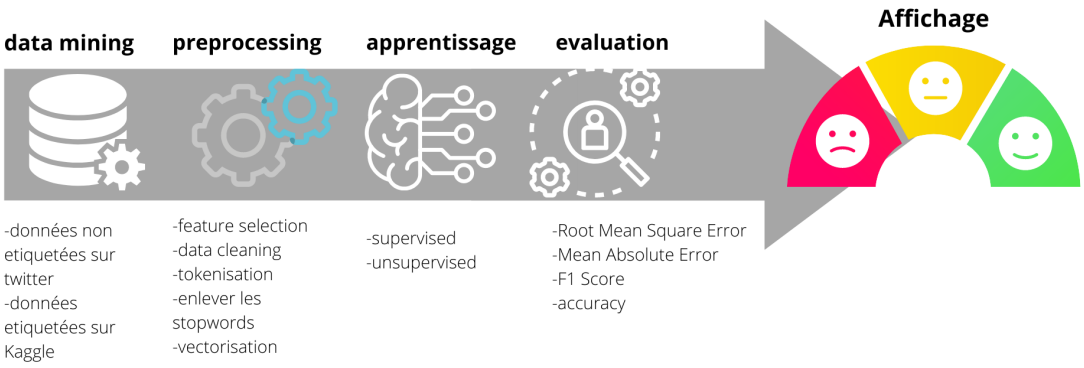

Figure2.1 - Data mining

2.3.1 Collecte des données

Le choix du réseau social Twitter s'explique par le fait que contrairement aux autres plateformes sociales, presque tous les tweets des utilisateurs sont entièrement publics et récupérables. Les données Twitter sont également assez spécifiques. L'API de Twitter nous permet de faire des requêtes complexes comme extraire chaque tweet sur un certain sujet au cours des vingt dernières minutes, ou extraire les tweets non retweetés d'un certain utilisateur. Nous pouvons également cibler les utilisateurs qui vivent spécifiquement dans un certain emplacement, connu sous le nom de données spatiales. De plus, les données de Twitter peuvent être une grande porte d'entrée sur les idées du grand public et sur la manière dont il reçoit un sujet. Cela, combiné à l'ouverture et à la généreuse limitation de débit de l'API de Twitter, peut produire des résultats puissants

Après extraction des tweets, nous nous retrouvons avec des données brutes et non structurées. Elles nécessitent un traitement pour les transformer en données compréhensibles par la machine, ceci nous amène à la deuxième étape.

2.3.2 Pré-traitement

Les données issues des réseaux sociaux (ou bien les tweets), sont à l'état brut. Généralement se sont des phrases courtes et spontanées, écrites dans un langage familier ou soutenu, et qui contiennent beaucoup de caractères spéciaux tels que les hashtags, les liens, les emojis ...etc
Feature selection : La sélection des features améliore la qualité du modèle, tout en optimisant le processus de modélisation. Si nous incluons des colonnes inutiles pour générer un modèle, davantage d'UC et de mémoire sont consommées pendant le processus d'apprentissage, et il faut plus d'espace de stockage pour le modèle terminé. Indépendamment du problème des ressources, l'utilisation de colonnes inutiles peut diminuer la qualité du modèle de plusieurs façons

Data cleaning () : nous procédons au data cleaning en enlevant les noms d'utilisateurs, caractères spéciaux et URLs, les espaces inutiles, en corrigeant les fautes d'orthographe à l'aide de dictionnaires et en convertissant les mots en minuscule.

Tokenization : Tokenizer une phrase revient à la séparer en tokens, c'est-à-dire en mots ou symboles distincts. D'un texte on extrait un vecteur de tokens, dans notre cas un vecteurs de mots.

Enlever les stopwords : un stop word est un mot très fréquent dans une langue et que l'on retrouve régulièrement dans des phrases, comme par exemple : les conjonctions de coordination, ces mots, ne portant aucune opinion, risquent de ralentir le processus d'analyse, alourdir cette tâche et même fausser les résultats.

La vectorisation : Ce processus utilise des modèles de langage pour mapper des mots à un espace vectoriel. Un espace vectoriel représente chaque mot par un vecteur de nombres réels. Il existe trois techniques très connues pour convertir un texte en vecteurs de caractéristiques numériques, à savoir le bags of words, la vectorisation tf-idf et le word embedding

Bags Of Words (BOW) : Tout d'abord, nous définissons un vecteur de longueur fixe où chaque entrée correspond à un mot dans notre dictionnaire de mots prédéfini. La taille du vecteur est égale à la taille du dictionnaire. Ensuite, pour représenter un texte à l'aide de ce vecteur, nous comptons combien de fois chaque mot de notre dictionnaire apparaît dans le texte et nous mettons ce nombre dans l'entrée de vecteur correspondante. Le résultat de cette représentation est une matrice

creuse

de taille (n, m) où : n : le nombre d'observations dans la collection de données. M : le nombre de tokens uniques dans l'ensemble de la collection de données

Par exemple, si notre dictionnaire contient les mots life, is, the, best, et que nous voulons vectoriser le texte « life is best», nous aurions le vecteur suivant : (1, 1, 0, 1).

Le problème avec cette méthode est qu'elle ne capture pas le sens du texte, ou le contexte dans lequel les mots apparaissent.

Tf-Idf : Cette approche repose sur le même schéma que celui du Bags Of Words, excepté qu'on donne un poids important aux tokens qui apparaissent souvent dans un texte en particulier mais pas dans tous les textes du corpus. Ces mots apportent beaucoup d'information sur le contenu du texte, on normalise donc les tokens de chaque texte selon l'information qu'ils apportent.

$$TF = \frac{Nombre\ de\ fois\ qu'un\ mot\ "X"\ apparaît\ dans\ un\ document}{nombre\ de\ mots\ présents\ dans\ le\ document}$$

$$IDF = log\left(\frac{Nombre\ de\ documents\ présents\ dans\ un\ corpus}{Nombre\ de\ documents\ contenant\ "X"}\right)$$

$$TF\ IDF = TF * IDF$$

Figure2.2 -TF IDF CALCUL

Word embedding : convertit un mot en un vecteur à n dimensions. Les mots liés tels que « Ukraine » et « slave » correspondent à des vecteurs similaires à n dimensions, tandis que des mots différents tels que « Algérie » et « croissant » ont des vecteurs différents. De cette façon, le "sens" d'un mot peut être reflété dans son intégration, un modèle est alors capable d'utiliser cette information pour apprendre la relation entre les mots.

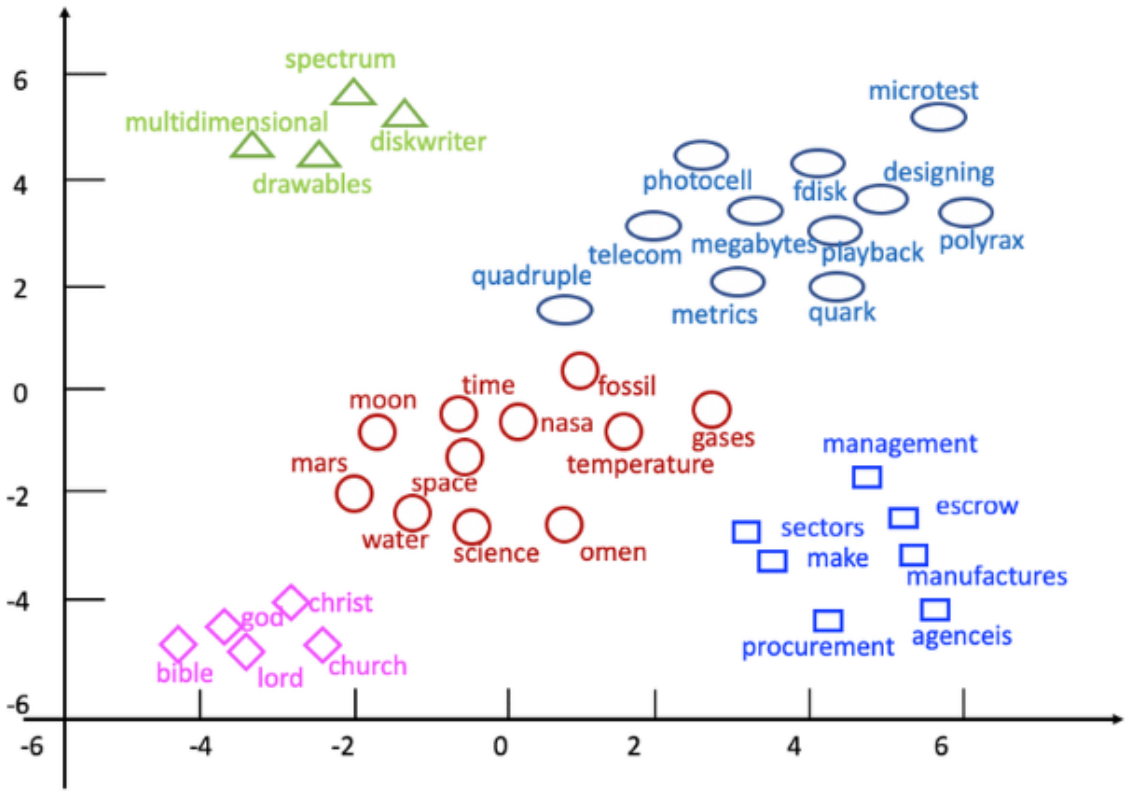

Figure2.3 -embedding[12]

Cependant l'utilisation de cette méthode pour des données très large peut poser problème, car celle-ci a tendance a sur ajuster (overfit) les données, En ayant un vocabulaire plus large, la méthode de "word embedding" est susceptible d'attribuer des règles à des mots qui ne sont que rarement vus dans le training. Nous utilisons donc la méthode Tf-Idf afin de vectoriser nos données. La phase de prétraitement des données est très importante car l'amélioration des performances d'un modèle repose sur le traitement et la représentation optimale des données pour un problème donné.

2.3.3 Analyze d'opinion

Après avoir effectué toutes les étapes précédentes, les données sont prêtes à être analysées. Il existe plusieurs approches, englobant plusieurs algorithmes chacune, pour réaliser cette tâche, nous allons utiliser différents modèles supervisés et non supervisés pour choisir à la fin le meilleur en matière de predictions.
Il y'a deux méthodes pour classifier une opinion en machine learning :

-Nous pouvons entraîner le modèle de classification supervisé sur des

données déjà étiquetées , ensuite appliquer le modèle construit sur notre base de données qui n'est pas étiquetée , la base de données utilisée pour construire le modèle est extraite de Kaggle et constituée de tweets et de commentaires faits sur Narendra Modi et d'autres dirigeants ainsi que sur l'opinion populaire envers le prochain Premier ministre de la nation (dans le contexte des élections générales tenues en Inde - 2019).Tous les tweets sont nettoyés à l'aide de Pythons re et également de NLP avec une étiquette sentimentale allant de -1 à 1.Les algorithmes utilisés pour entraîner notre modèle sont Naive Bayes , SVM et KNN car ils sont les plus utilisés pour la classification en classes multiples -Nous pouvons avoir recours à l'apprentissage non supervisé puisque nos données ne sont pas étiquetées , nous choisissons le clustering car la réduction de dimension et l'association ne sont pas adaptés à la classification de l'opinion

Cette phase consiste à générer un modèle de prédiction, qui permet de prédire la classe d'appartenance d'un texte. Dans notre cas, c'est définir si notre texte contient une opinion positive, négative ou neutre. Pour ensuite l'évaluer et le sauvegarder.

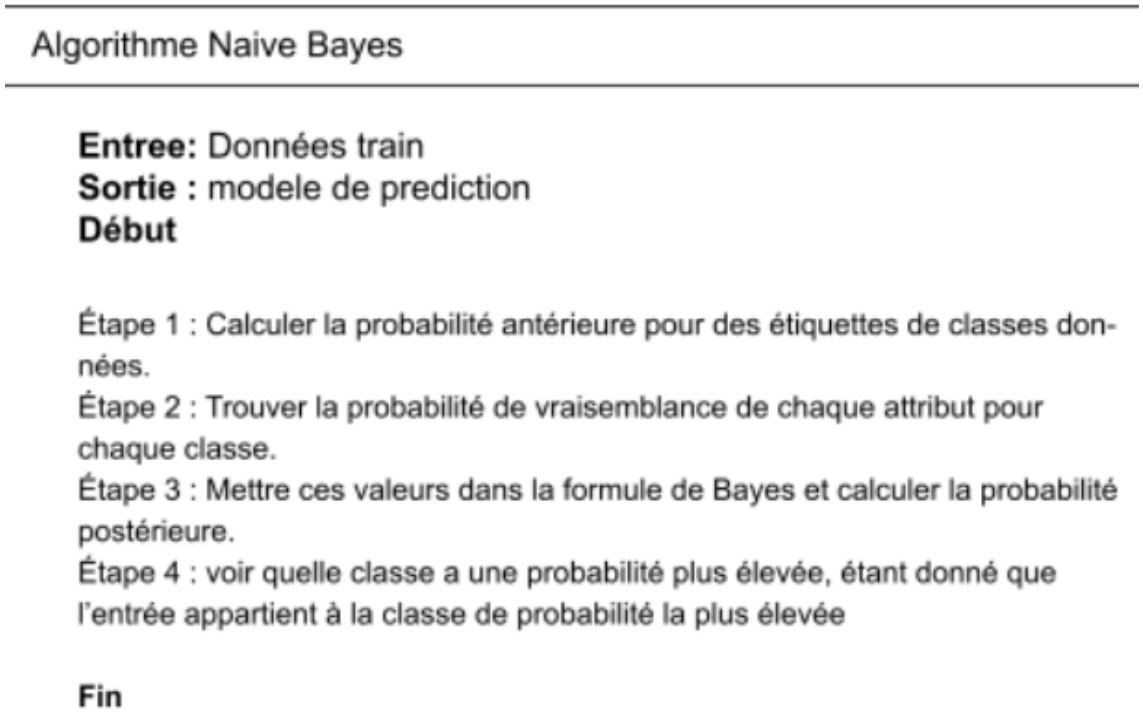

Figure2.4 -Naive Bayes

Algorithme K-nearest neighbors (KNN)

Entree: Données train
Sortie : modele de prediction
Début

Étape 1 : Sélectionner le nombre K de voisins
Étape 2 : Calculer la distance (Euclidienne ou Manhattan)
Étape 3 : Prendre les K voisins les plus proches selon la distance calculée.
Étape 4 : Parmi ces K voisins, compter le nombre de points appartenant à chaque catégorie.
Étape 5 : Attribue le nouveau point à la catégorie la plus présente parmis ces K voisins.

Fin

Figure2.5 -Knn

Algorithme Support vector Machine (SVM)

Entree: Données train
Sortie : modele de prediction
Début

Étape 1 : préparer la matrice des patrons
Étape 2 : Choisir la fonction noyau à utiliser
Étape 3 : Choisir les paramètres de la fonction noyau et la valeur de C
(valeurs suggérées par le logiciel SVM ou essai-erreur)
Étape 4 : Exécuter l'algorithme d'apprentissage pour trouver i
Étape 5 :Les données nouvelle peuvent être classées en fonctions des i et des valeurs supports trouvés

Fin

Figure2.6 -SVM

Algorithme Clustering (K_means)

Entree: Données
Sortie : trois clusters
Début

Étape 0 : Initialisation
Tirer aléatoirement 3 individus. Ces 3 individus correspondent aux centres initiaux des 3 classes.
Étape 1 :
Calculer la distance entre les individus et chaque centre.
Étape 2 :
Affecter chaque individu au centre le plus proche.
Étape 3 :
Calculer les centres de gravité des groupes qui deviennent les nouveaux centres.
Boucle iterative
Recommencer les étapes 1, 2 et 3 tant que les individus sont réaffectés à de nouveaux groupes après une itération.

Fin

Figure2.7 -Clustering

2.3.4 Conclusion

A la fin de ce chapitre, nous sommes arrivés à répondre à notre problématique en proposant une interface qui répond au besoin de l'utilisateur. La réalisation de cette solution fera l'objet de la prochaine partie du rapport.

Chapitre 3

Chapitre 3 : Implémentation

3.1 Introduction

Après avoir présenté une solution globale pour répondre à notre problématique concernant l'analyse d'opinion dans les réseaux sociaux, nous allons aborder les outils permettant de mettre en œuvre notre solution.

3.2 1.Environnements et technologies utilisées

3.2.1 technologies utilisées

Python

Python est un langage de script de haut niveau, structuré et open source. Conçu pour être orienté objet, il n'en dispose pas moins d'outils permettant de se livrer à la programmation fonctionnelle ou impérative, c'est d'ailleurs une des raisons qui lui vaut son appellation de langage agile.. Il est également apprécié pour la clarté de sa syntaxe. C'est avec ce langage de programmation que nous avons implémenté toutes les fonctions, méthodes et étapes de notre processus de fouille d'opinions.[13]

Bibliothéques

Collecte des données

Twint est un outil de scraping Twitter avancé écrit en Python qui permet de scraper les Tweets des profils Twitter sans utiliser l'API de Twitter.Twint utilise les opérateurs de recherche de Twitter

pour vous permettre de récupérer les Tweets d'utilisateurs spécifiques, de récupérer les Tweets relatifs à certains sujets, hashtags et tendances. De plus, l'API Twitter a une limite de récupération de seulement 3200 tweets alors que twint n'a pas de limite de téléchargement de tweets.[13]

Prétraitement et structuration des données

(Natural Language Toolkit)NLTK est une bibliothèque logicielle en Python permettant un traitement automatique des langues, développée par Steven Bird et Edward Loper du département d'informatique de l'université de Pennsylvanie. En plus de la bibliothèque, NLTK fournit des démonstrations graphiques, des données-échantillon, des tutoriels, ainsi que la documentation de l'interface de programmation (API). Cette bibliothèque nous a fourni toutes les ressources linguistiques nécessaires au prétraitement des données. [13]

Pandas est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles. Pandas est un logiciel libre sous licence BSD. Cette bibliothèque nous aider pour l'importation ainsi que pour la structuration des données[13]

est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. [13]

Apprentissage

Scikit-learn est une bibliothèque libre Python destinée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme Inria. Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des logistiques régressions, des algorithmes de classication, et les machines à vecteurs de support. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python, notamment NumPy et SciPy. Cette bibliothèque nous a été utile lors de la vectorisation des données par le biais d'outils tels que : CountVectorizer et TdfVectorizer. [13]

Environnements utilisées

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS2. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. [14]

3.3 Architecture globale du système

Notre système se divise en 3 parties principales : la première est celle de la collecte de données, elle commence avec l'extraction de données non étiquetées sur twitter avec un code python qui utilise la bibliothèque "Twint". Le résultat est sous la forme json, d'autres données étiquetés sont récupérées de kaggle[15] pour l'apprentissage supervisé (données qui expriment les sentiments et opinions des utilisateurs de Twitter sur plusieurs sujet), le modèle sera ensuite appliqué sur nos données extraites ce qui nous amène à la deuxième partie, le prétraitement de données (Data prerocessing).

Les fichiers contenant les données annotées et nettoyées passent ensuite par le prétraitement assuré par notre système en utilisant les différentes

bibliothèques Python. De là, on passe à l'entrainement de notre modèle en utilisant scikit learn, ce qui marque la fin de cette partie.

La dernière partie nommée "Visualisation", ou on affiche le meilleure algorithme et les statistiques resultantes qui repondent a notre problematique sur la guerre en Ukraine

3.3.1 Collecte des données

Twint nous permet d'extraire des données importantes sur le sujet de la guerre en Ukraine comme le texte, l'identiant, les liens, les hashtags, la date, l'heure, la localisation, la langue etc. Notre data set est constitué de 97538 lignes et de 36 colones. On sauvegarde ensuite le résultat obtenu dans un fichier Json pour pouvoir l'utiliser en dehors de la console python.

```
15   c = twint.Config()
16   c.Search = "ww3"
17   c.Store_json = True
18   c.Output = 'file.json'
19   #Run
20   twint.run.Search(c)
21   c.Since = "2022-05-09"
22
23   data=pd.read_json('file.json',lines=True)
```

Figure3.1 -data mining

3.3.2 Prétraitement

```
data=data[data["language"] == "en"]
new_data=data.drop(['name','likes_count','geo','source',
'thumbnail','hashtags','cashtags','retweet','quote_url',
'id','link','user_rt_id','user_rt','retweet_id','reply_to',
'conversation_id','created_at','time','timezone','user_id',
'username','place','urls','photos','video','language',
'mentions','translate','replies_count','retweets_count',
'retweet_date','trans_src','trans_dest'], axis=1)
```

Figure3.2 -Feature selection

Le nettoyage des données collectées consiste en la suppression ou conversion de toute entité n'ayant aucune relation avec l'opinion (lien URL,

emojis, nom d'utilisateur,ponctuations...)

```python
def remove_usernames_links(tweet):
    tweet = re.sub('@[^\s]+','',tweet) #enlever les identifiants
    tweet = re.sub('http[^\s]+','',tweet) #enlever les URL
    tweet = re.sub("[^-9A-Za-z ]", "" , tweet)
    tweet = "".join([i.lower() for i in tweet if i not in string.punctuation])
    tweet = re.sub("\s+"," ", tweet)#enlever les espaces inutiles
    return tweet

new_data['tweet'] = new_data['tweet'].apply(remove_usernames_links)
```

Figure3.3 -Data cleaning

Tokenisation

Afin de découper chaque message en termes, nous avons utilisé la fonction word_tokenize de la librairie NLTK Pour la suppression des mots vides et la correction de fautes d'orthographe, les bibliothèques NLTK et spellchecker comportent des mots prédéfinis pour des langues comme l'anglais

```python
#_____________________________________________________________________
def tokeniseur(text):
    tweet = TweetTokenizer()
    text=tweet.tokenize(text)
    return text
new_data['tweet'] = new_data['tweet'].apply(tokeniseur)
#_____________________________________________________________________
spell = SpellChecker(distance=1)
def Correct(tweet):
    for x in tweet:
        x=spell.correction(x)
    return tweet
new_data['tweet'] = new_data['tweet'].apply(Correct)
#_____________________________________________________________________
def stopword(tweet):
    stopwords = nltk.corpus.stopwords.words('english')
    tweet = [i for i in tweet if i not in stopwords]
    return tweet
new_data['tweet'] = new_data['tweet'].apply(stopword)
```

Figure3.4 -Tokenisation

3.3.3 Vectorisation

TfidfVectorizer effectue la transformation TF-IDF à partir d'une matrice de comptes fournie.

```
120    x=new_data['tweet'].values
121    vect = TfidfVectorizer().fit(x)
122    x = vect.transform(x)
123    print(x)
```

Figure3.5 -Vectorisation

3.3.4 Partitionnement de données

Nous devons diviser notre ensemble de données en ensembles d'apprentissage et de test pour évaluer les performances de notre modèle d'apprentissage automatique. Le train set est utilisé pour ajuster le modèle. Le deuxième ensemble est appelé l'ensemble de données de test, cet ensemble est uniquement utilisé pour les prédictions.

```
x=df['clean_text'].values
y=df['category'].values

x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=50)
```

Figure3.6 -Data splitting

3.3.5 Modélisation

Naïve-Bayes

Le classificateur multinomial Naive Bayes convient à la classification avec des caractéristiques discrètes (par exemple, le nombre de mots pour la classification de texte). La distribution multinomiale nécessite normalement un nombre entier de caractéristiques. Cependant, en pratique, les comptages fractionnaires tels que tf-idf peuvent également fonctionner.

```
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB()
clf.fit(x_train_vectorized,y_train)
prediction_naive_bayes=clf.predict(x_test_vectorized)
print("Naive Accuracy Score -> ",accuracy_score(prediction_naive_bayes,y_test)*100)
yfit = SVM .predict(f)
print(yfit)
```

Figure3.7 -Naïve-Bayes

K nearest neighbors

Cet algorithme est utilisé pour résoudre les problèmes du modèle de classification. L'algorithme K-plus proche voisin ou K-NN crée essentiellement une frontière imaginaire pour classer les données. Lorsque de nouveaux points de données arrivent, l'algorithme essaie de prédire cela au plus proche de la ligne de démarcation.

```
from sklearn.neighbors import KNeighborsClassifier

# 2. instantiate the model (with the default parameters)
knn = KNeighborsClassifier()
# 3. fit the model with data (occurs in-place)
knn.fit(x_train_vectorized, y_train)
prediction_knn=knn.predict(x_test_vectorized)
print("knn Accuracy Score -> ",accuracy_score(prediction_knn,y_test)*100)
yfit = SVM .predict(f)
print(yfit)
```

Figure3.8 -k-nearest neighbors

SVM

SVC est une classe capable d'effectuer une classification binaire et multi-classes sur un ensemble de données.

```
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(x_train_vectorized,y_train)
# predict the labels on validation dataset
predictions_SVM = SVM.predict(x_test_vectorized)
# Use accuracy_score function to get the accuracy
print("SVM Accuracy Score -> ",accuracy_score(predictions_SVM,y_test)*100)
yfit = SVM .predict(f)
print(yfit)
```

Figure3.9 -Support-Vector Machine

Clustering

k-means est l'un des algorithmes les plus populaires en Clustering. L'approche derrière cet algorithme simple est à peu près quelques itérations et des clusters de mise à jour selon les mesures de distance qui sont calculées à plusieurs reprises. k est le nombre de clusters qui doivent être formés

```
from sklearn.cluster import KMeans
km=KMeans(n_clusters=3,init='k-means++')
km.fit(f)
new_data["cluster"] = km.labels_
```

Figure3.10 -Clustering Kmeans

3.3.6 Évaluation

Après avoir construit et entrainer les trois modèles avec le même ensemble de données, nous passons à l'évaluation an de déterminer le meilleur modèle avec la meilleure représentation des données.Pour cela ,nous utilisons la matrice de confusion, la précision,le rappel et le f1-score

Matrice de confusion

Également connue sous le nom de tableau de contingence, la matrice de confusion permet d'évaluer les performances d'un modèle de classification. Elle montre donc à quel point un certain modèle peut être confus lorsqu'il fait des prédictions. Dans sa forme la plus simple.
Elle compare les données réelles pour une variable cible à celles prédites par un modèle. Les prédictions justes et fausses sont révélées et réparties par classe, ce qui permet de les comparer avec des valeurs définies.

Précision

La précision indique le rapport entre les prévisions positives correctes et le nombre total de prévisions positives. Ce paramètre répond donc à la question suivante : sur tous les enregistrements positifs prédits, combien sont réellement positifs ?

Rappel

le rappel est un paramètre qui permet de mesurer le nombre de prévisions positives correctes sur le nombre total de données positives. Il permet de répondre à la question suivante : sur tous les enregistrements positifs, combien ont été correctement prédits ?

F1-Score

Le score F1 est une moyenne harmonique de la précision et du rappel. Il équivaut au double du produit de ces deux paramètres sur leur somme. Sa valeur est maximale lorsque le rappel et la précision sont équivalents.

Clustering silhouette score

Le coefficient de silhouette est calculé à l'aide de la distance moyenne intra-cluster (a) et de la distance moyenne du cluster le plus proche (b) pour chaque échantillon. Le coefficient de silhouette pour un échantillon est (b - a) / max(a, b). Pour clarifier, b est la distance entre un échantillon et le cluster la plus proche dont l'échantillon ne fait pas partie.

Naïve-Bayes

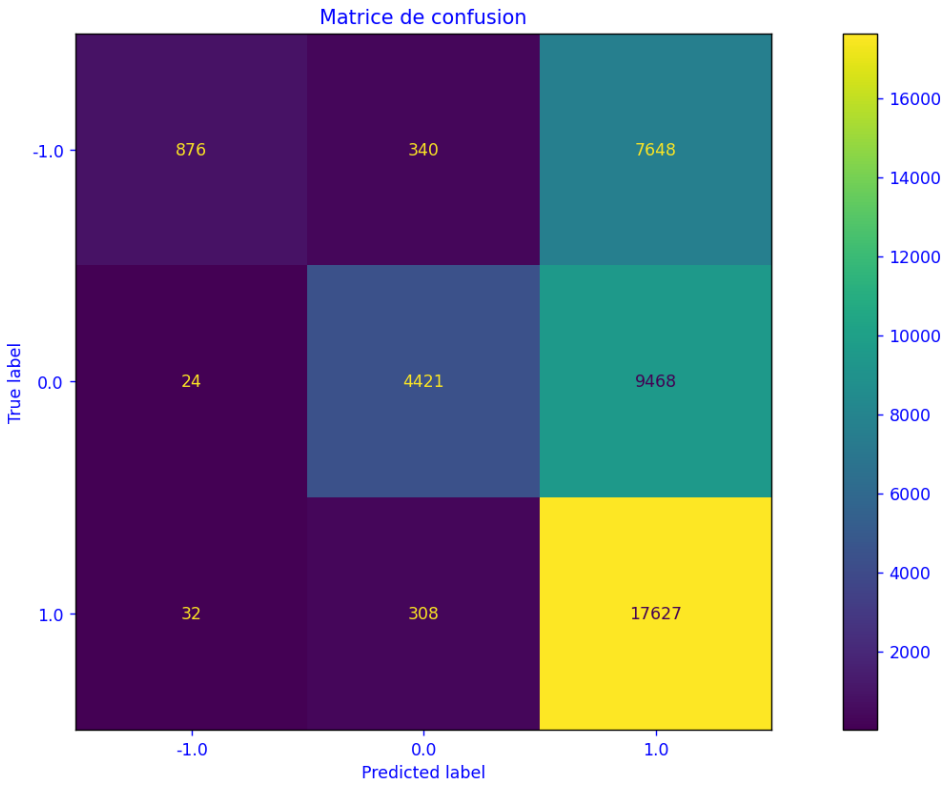

Figure3.11 -Matrice de confusion Naïve-Bayes

Naive Bayes score :

	precision	recall	f1-score	support
-1.0	0.94	0.11	0.19	8864
0.0	0.88	0.32	0.47	13913
1.0	0.51	0.98	0.67	17967
accuracy			0.57	40744
macro avg	0.78	0.47	0.45	40744
weighted avg	0.73	0.57	0.50	40744

KNN

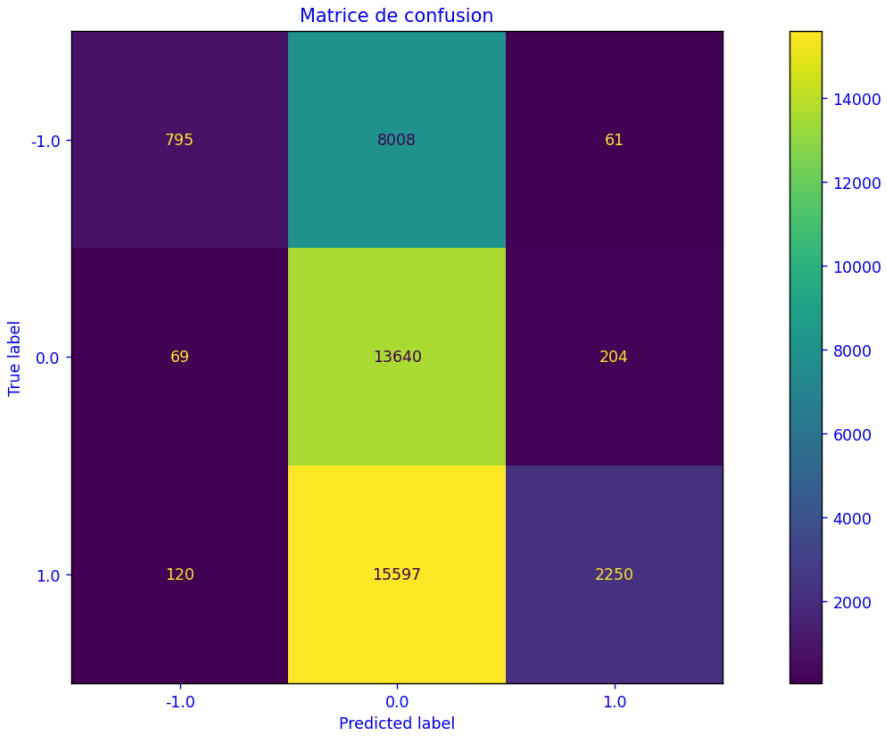

Figure3.12 -Matrice de confusion k-nearest neighbors

KNN score :

	precision	recall	f1-score	support
-1.0	0.80	0.10	0.17	8864
0.0	0.37	0.98	0.53	13913
1.0	0.89	0.13	0.23	17967
accuracy			0.41	40744
macro avg	0.69	0.40	0.31	40744
weighted avg	0.69	0.41	0.32	40744

Figure3.13 -Knn Score

SVM

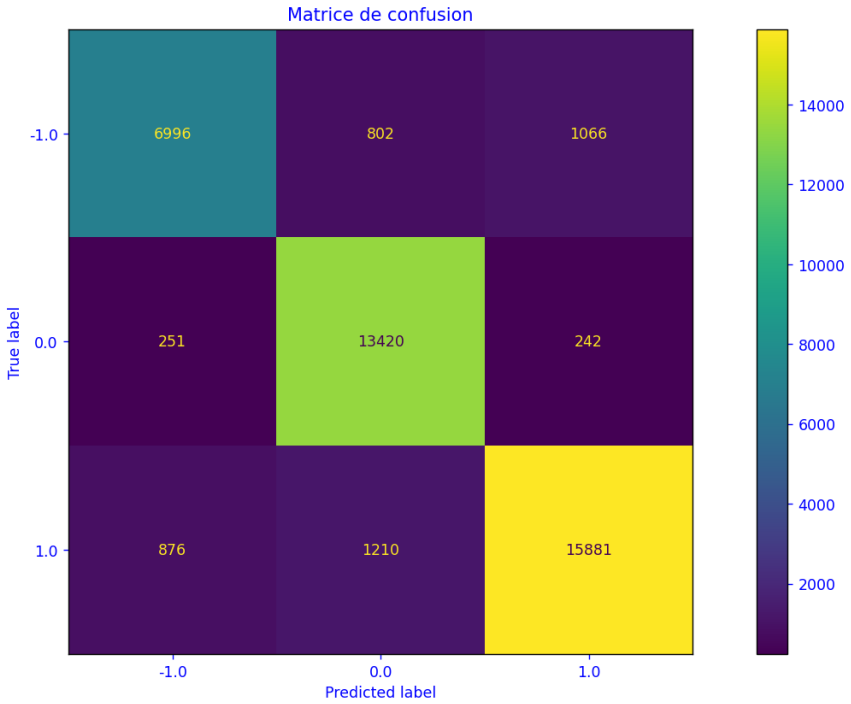

Figure3.14 -Matrice de confusion Support Vector Machine

SVM score :

	precision	recall	f1-score	support
-1.0	0.86	0.80	0.83	8864
0.0	0.88	0.96	0.92	13913
1.0	0.92	0.89	0.90	17967
accuracy			0.89	40744
macro avg	0.89	0.88	0.88	40744
weighted avg	0.89	0.89	0.89	40744

Clustering

```
evaluation de K_means : 0.0010160589342743777
```

Figure3.15 -silhouette_score K_means

D'après les résultats, on déduit que le meilleur modèle pour notre ensemble de données, est le modèle SVM avec la représentation TFIDF.Les resultats de notre étude sont donc presentés par ce modéle

3.3.7 Résultats

Les résulats de l'analyse est la suivante

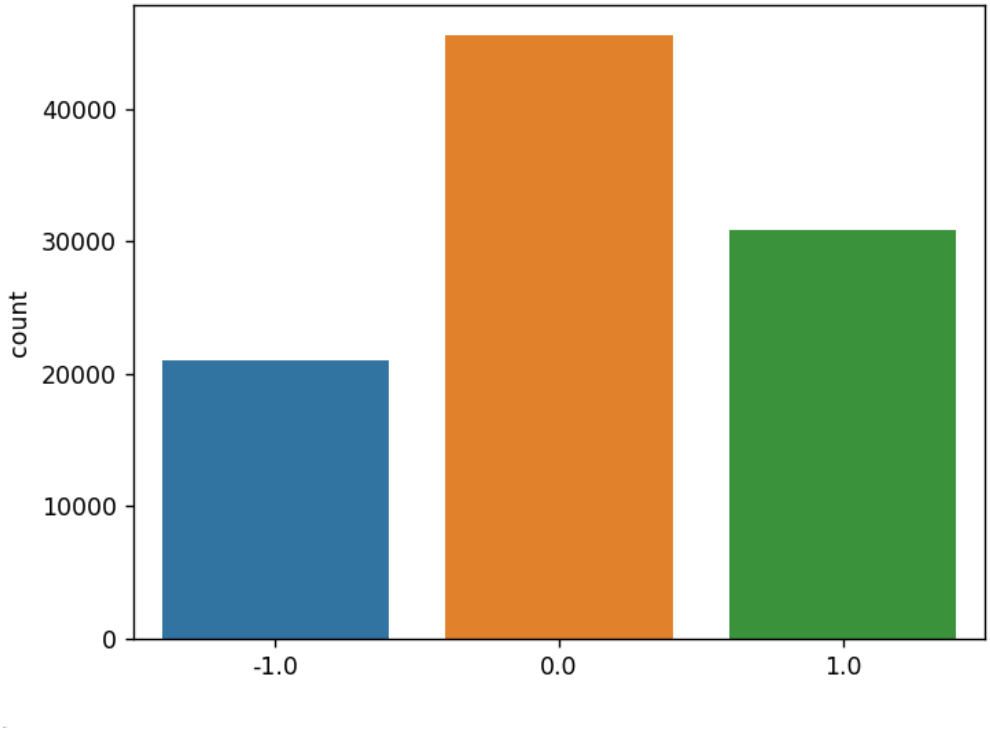

Figure3.16 -Diagramme en barre Polarité des opinions

0 indique que l'opinion est neutre, les internautes n'ont pas d'opinion sur le sujet
1 indique que l'opinion est positive , donc que les personnes veulent une guerre
-1 indique que l'opinion est négative, entre autre, les internautes ne veulent pas de declanchement d'une troisiéme guerre mondiale

```
pourcentage de tweets positifs :
31.683036355061617
pourcentage de tweets neutres :
46.728454551046774
pourcentage de tweets negatifs :
21.588509093891613
```

Figure3.17 -pourcentage de la polarité

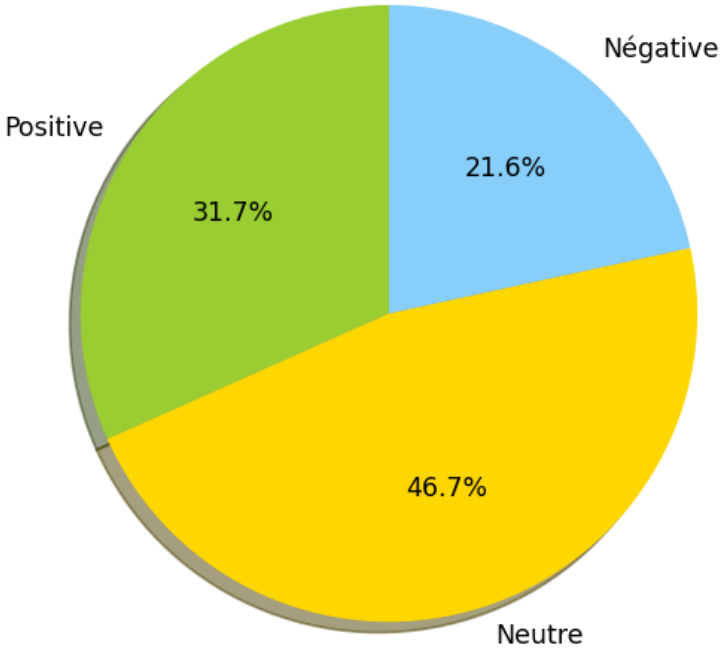

Figure3.18 -Diagramme circulaire pourcentage de la polarité

3.4 Conclusion

Pour conclure nous avons présenté dans ce chapitre les différents aspects de réalisation du projet, en passant par les outils permettant de mettre en œuvre notre solution, et la présentation de la réalisation de notre approche d'analyse pas à pas à travers les différents scripts réalisés depuis l'extraction et prétraitement des données jusqu'à la classification et l'affichage des résultats.

Conclusion

Outre la capacité de briser les frontières géographiques, les réseaux sociaux cassent les barrières en termes de communication, bousculent les pensées, créent le débat et jouent un rôle d'influence au sein de la société. Un exemple de sujet polémique et d'actualité sur les réseaux sociaux est la guerre de la Russie contre l'Ukraine. Dans ce projet, nous avons analysé l'opinion des utilisateurs sur l'essor de cette guerre : La guerre en Ukraine va-t-elle déclencher une troisième guerre mondiale ou pas ?

Notre approche est fondée l'extraction de tweets sur le réseau social TWITTER , ensuite la génération de plusieurs modèles de classification afin de déterminer le plus pertinent d'entre eux et de l'utiliser pour polariser les opinions sur la guerre en Ukraine en trois classes : positive, négative et neutre. Le modèle choisi est le SVM, avec précision de 89%. Les résultats sont présentés comme suit :

-pourcentages des tweets positifs :31.68%
-pourcentages des tweets neutre : 46.72%
-pourcentages des tweets négatifs : 21.58%

Les résultats sont une sources de plusieurs interpretations, nous pouvons cependant confirmer que 67% des internautes ne veulent pas déclancher une troisième guerre mondiale. Une autre approche possible serait celle du Deep Learning. En effet, Les progrès réalisés cette dernière décennie en Deep Learning ont profité à plusieurs domaines, ce qui lui a permis de s'imposer comme une approche incontournable en Machine Learning. La classification automatique des documents textuels est l'une des tâches dans laquelle le Deep Learning s'avère particulièrement utile et performant.

Bibliographie

[1] https ://monkeylearn.com/sentiment-analysis/ : :text=Sentiment%20analysis%20algorithms%20fall%20into,rule%2Dbas

[2] https ://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501(consulté le 13 Mai)

[3] https ://openclassrooms.com/en/courses/6389626-train-a-supervised-machine-learning-model/6405876-understand-the-logistic-regression-algorithm

[4] https ://blog.mindmanager.com/blog/2021/05/11/decision-tree-diagrams/

[5] https ://www.researchgate.net/figure/Construction-dune-foret-aleatoire_fig7_281184702

[6] https ://www.mathweb.fr/euclide/2019/06/25/les-k-plus-proches-voisins/

[7] https ://www.researchgate.net/figure/Illustration-of-how-a-Gaussian-Naive-Bayes-GNB-classifier-works-For-each-data-point_fig1_255695722

[8] https ://turfmining.fr/svm/

[9] https ://www.guru99.com/unsupervised-machine-learning.html(consulté le 13 Mai)

[10] https ://laptrinhx.com/machine-learning-clustering-algorithm-2431293740/

[11] https ://monkeylearn.com/natural-language-processing/

[12] https ://www.researchgate.net/figure/2D-PCA-projection-of-word-embeddings-Five-different-word-clusters-are-shown_fig2_332892222

[13] https ://pypi.org/

[14] https ://code.visualstudio.com/

[15] https ://www.kaggle.com/datasets/cosmos98/twitter-and-reddit-sentimental-analysis-dataset ?select=Twitter_Data.csv