



DIPLOME NATIONAL D'INGENIEUR



ECOLE SUPÉRIEURE PRIVÉE D'INGÉNIERIE ET DE TECHNOLOGIES

www.esprit.tn - E-mail : contact@esprit.tn

Siège Social : 18 rue de l'Usine - Charguia II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889

Annexe : Z.I. Chotrana II - B.P. 160 - 2083 - Pôle Technologique - El Ghazala - Tél. : +216 70 685 685 - Fax. : +216 70 685 454

PROJET DE FIN D'ÉTUDES

DIPLÔME NATIONAL D'INGÉNIEUR

SPÉCIALITÉ : Informatique

**Réalisation d'un système de détection
d'anomalies pour JO24**

Réalisé par : ABDELLI Farah

Encadrante académique : SBISSI Samia

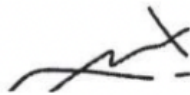
Encadrant professionnel : SNOUSSI Hichem



Signatures

Signature de l'encadrant de l'entreprise

Mr SNOUSSI Hichem



Signature de l'encadrant pédagogique

Mme SBISSI Samia



Dédicaces

Je dédie ce projet :

A ma chère mère,

A mon cher père,

Qui n'ont jamais cessé de me soutenir et de m'encourager pour que je puisse atteindre mes objectifs. Ceux qui ont sacrifié leurs plus belles années pour embellir les miennes, je vous dois ma réussite, aucun mot ne serait assez pour témoigner de l'étendue des sentiments que j'éprouve à leur égard,

Quoi que je fasse je ne vous remercierai jamais assez pour votre bienveillance et vos sacrifices.

A ma très chère sœur Dina

Puisse Dieu te donne santé, bonheur, courage et surtout réussite

A ma très chère cousine Mimi et ma tante Eya

Qui m'avez toujours soutenu et encouragé durant ce stage.

A ma chère grand-mère,

Qui je souhaite une bonne santé

A mes très chers amis

Pour leurs aides et supports dans les moments difficiles

A toute ma famille et tous mes amis,

Je vous offre ce travail, le fruit de mes efforts et le résultat de longues années d'études, en témoignage de reconnaissance et de gratitude pour l'encouragement et le soutien que vous n'avez cessé de m'apporter.

Abdelli Farah

Remerciements

Au terme de ce travail, je tiens à adresser mes profonds et sincères remerciements à toutes les personnes qui ont contribué au succès de mon stage de fin d'études.

Je remercie tout d'abord mes chers parents, à qui je dois mon parcours et ma réussite. Je remercie également toute ma famille et tous mes amis pour leur aide et leur soutien durant le déroulement de mon stage de fin d'études.

Je tiens à remercier Mr Snoussi Hichem, responsable de mon stage, pour sa disponibilité et ses conseils judicieux qu'il n'a cessé de me prodiguer tout au cours de ce projet.

Je tiens à remercier Mme Sbissi Samia, ma superviseure, qui m'a conseillé tout au long de mon processus de travail et a contribué à la réussite du projet.

Je remercie particulièrement Mme El Mawass Nour et Mr Babiga Berragah pour leurs intérêts, leurs patiences, leurs soutiens et leurs contributions à la réalisation de ce travail.

Que tous les membres de l'équipe ANR-DISCRET trouvent ici l'expression de mes gratitude pour leur sympathie et pour avoir facilité mon intégration au sein de l'équipe.

Finalement, je saisis cette occasion pour remercier les membres du jury de m'honorer par leur acceptation de me prêter leur attention et d'évaluer mon travail.

Glossaire

UTT: Université de Technologie de Troyes

CRISP-DM : Cross Industry Standard Process for Data Mining

SEMMA: Sample, Explore, Modify, Model, Assess

KDD : Knowledge Discovery in Databases

BOW: Bag Of Words

TF-IDF: Term Frequency and Inverse Document Frequency

W2V: Word To Vector

ML: Machine Learning

DL: Deep Learning

NLP: Neural Language Process

RL: Régression Logistique

SVM: Support Vector Machine

RF: Random Forest

BERT : Bidirectional Encoder Representations from Transformers

ARIMA : Auto Regressive Integrated Moving Average

LSTM : Long Short-Term Memory

ANN: Artificial Neural Network

RNN: Recurrent Neural Network

ROC : Receiver Operating Characteristic

AUC : Area Under Curve

Table des matières

<i>Dédicaces</i>	<i>3</i>
<i>Remerciements.....</i>	<i>5</i>
<i>Glossaire</i>	<i>6</i>
<i>Table des matières</i>	<i>7</i>
<i>Liste des Figures.....</i>	<i>9</i>
<i>Liste des tableaux</i>	<i>10</i>
<i>Résumé.....</i>	<i>11</i>
<i>Introduction générale.....</i>	<i>12</i>
<i>Chapitre 0 : Contexte du projet.....</i>	<i>13</i>
I. Cadre du projet	14
1. Présentation de l'UTT	14
2. Présentation du CapSec.....	15
II. Objectif du Projet.....	16
III. Méthodologie du projet.....	17
1. Méthodologies.....	17
2. Choix.....	18
3. Comparaison	19
<i>Chapitre 1 : Compréhension du métier</i>	<i>20</i>
I. Objectif métier.....	21
II. Objectifs Data science	21
III. Plan du projet	21
1. Étapes de planification du projet.....	22
2. Outils et technologies utilisés.....	23
<i>Chapitre 2 : Compréhension des données</i>	<i>25</i>
I. Collecte de données	26
1. French Tweet	26
2. Notre-Dame :	27
3. Collecte Paris avril 2021	27
4. Collecte Paris mai 2021	28
5. Orages Île-de-France juin 2021	29
<i>Chapitre 3 : Préparation des données</i>	<i>31</i>
I. Prétraitement des données.....	32
II. Représentation des documents	35
1. Représentation par sac de mots	36

2.	Représentation par Word2Vec	36
3.	Représentation par TF-IDF	37
III.	L'approche de validation	38
Chapitre 4 : Modélisation.....		40
I.	Classification des textes et analyse de sentiments.....	41
1.	Techniques de modélisation	41
2.	Mesure des performances.....	42
3.	Construction des modèles	43
II.	Détection d'anomalies	50
1.	Techniques de modélisation	50
2.	Mesure des performances.....	50
3.	Construction des modèles	51
Chapitre 5 : Évaluation.....		59
I.	Évaluation des résultats	60
1.	Classification et analyse de sentiments	60
2.	Détection d'anomalies.....	63
II.	Modèle approuvé	64
Chapitre 6 : Déploiement		65
Conclusion et perspectives		69
Webographie.....		70
Annexes.....		72

Liste des Figures

Figure 1: Université de Technologie de Troyes [5]	15
Figure 2: Le département X (CapSec)[7]	16
Figure 3: Processus de la méthode KDD [53]	17
Figure 4: Processus de la méthode SEMMA [11]	18
Figure 5: Processus de la méthode CRISP-DM [11]	19
Figure 6: Extrait de l'ensemble de données "french_tweets"	26
Figure 7: Extrait de l'ensemble de données "Notre Dame"	27
Figure 8: Extrait de l'ensemble de données "Paris avril 2021"	28
Figure 9 : Extrait de l'ensemble de données "Paris mai 2021"	29
Figure 10 : Extrait de l'ensemble de données (Orages Ile de France 2021)	30
Figure 11 : Code de source de la partie prétraitement des données (1)	33
Figure 12 : Code de source de la partie prétraitement des données (2)	34
Figure 13: Un extrait de données avant et après le pré traitement	35
Figure 14: Exemple représentation BOW	36
Figure 15: Représentation Word2Vec	36
Figure 16 : Exemple de la représentation Word2Vec	37
Figure 17: Exemple de la représentation TF-IDF	37
Figure 18: Exemple de la représentation TF-IDF(2)	38
Figure 19: L'approche de validation	39
Figure 20 : Matrice de confusion [51]	42
Figure 21: Paramètre coût (petite valeur /grande valeur) [57]	44
Figure 22: Entraînement du modèle SVM	45
Figure 23: Entraînement du modèle RF	46
Figure 24: Entraînement du modèle RL	46
Figure 25: Implémentation du modèle TextBlob	47
Figure 26: Entraînement du modèle CamemBERT	48
Figure 27: Calcul du score pour la méthode de CamemBERT	49
Figure 28: Exemple de courbe ROC [58]	51
Figure 29: Principe de détection	52
Figure 30 : Résultat de la prédiction du modèle ARIMA (signal original vs signal prédit)	53
Figure 31 : Résultat de la prédiction du modèle ARIMA (intervalle de confiance)	53
Figure 32 : Résultat de la prédiction en utilisant la fenêtre glissante	54
Figure 33 : Courbe d'évolution des écarts calculés entre le signal prédit et le signal original	54
Figure 34 : Courbe d'évolution des écarts calculés entre le signal prédit et le signal original après l'utilisation de la fenêtre glissante	55
Figure 35: Redimensionnement des vecteurs en 3 dimensions	55
Figure 36: Entraînement du modèle LSTM	56
Figure 37: Résultat de la prédiction du modèle LSTM (signal original vs signal prédit)	56

Figure 38: Implémentation de la méthode ROC pour le choix du seuil.....	57
Figure 39: courbe ROC pour le choix du seuil.....	58
Figure 40 : Les performances des différents classifieurs	61
Figure 41 : Mesures de performance	62
Figure 42 : Courbe de l'évolution des ratios de sentiment dans le temps	63
Figure 43: Courbe ROC ARIMA vs LSTM.....	63
Figure 44: Interface de connexion.....	66
Figure 45: Visualisation détaillée sur la collecte de Mai	67
Figure 46: Visualisation des anomalies détectées pendant le mois de Mai en utilisant le filtre date	67
Figure 47 : Visualisation des anomalies détectées pendant le mois de Mai en utilisant le filtre mots clés.....	68
Figure 48 : Visualisation des anomalies détectées pendant le mois de Juin avec le modèle LSTM	68
Figure 49 : Equation logistique de la RL [33].....	75
Figure 50 : La méthode SVM [33].....	75
Figure 51 : Architecture RNN [38]	75
Figure 52 : Architecture d'un auto encodeur	75
Figure 53 : Architecture du modèle LSTM [38]	75
Figure 54 : Architecture du modèle LSTM [48]	75

Liste des tableaux

Tableau 1: Outils et technologies utilisés	23
Tableau 2: Tableau comparatif entre les méthodes KDD , SEMMA et CRISP-DM.....	74

Résumé

Le travail présenté dans ce rapport, qui a été effectué au sein du laboratoire CapSec de l'Université de Technologie de Troyes, entre dans le cadre du projet de fin d'études pour l'obtention du diplôme national d'ingénieur en informatique.

Il concerne la réalisation d'un système de détection d'anomalies.

Ce système va assurer la surveillance des événements sportifs des Jeux Olympiques 2024 à l'aide de la détection et de la localisation, en temps réel, des situations inhabituelles ou critiques dans les zones urbaines tout en bénéficiant de la puissance de l'apprentissage profond.

Abstract

The work presented in this report, which was carried out in the CapSec laboratory of the University of Technology of Troyes, is part of the end of studies project for obtaining the national diploma in computer engineering.

It concerns the realization of an anomaly detection system. This system will provide surveillance of sporting events for the 2024 Olympic Games using real-time detection and localization of unusual or critical situations in urban areas while benefiting from the power of deep learning.

Introduction générale

Twitter compte plus de 340 millions d'utilisateurs actifs, et la participation de ces utilisateurs de Twitter a conduit à la génération rapide de données, en particulier dans le contexte de sujets populaires tels que les reportages, la politique et le sport. Ce réseau est devenu l'un des médias les plus populaires sur les réseaux sociaux aujourd'hui. Twitter devient de plus en plus populaire en tant qu'outil de recherche riche pour résoudre divers problèmes de sciences sociales et de science des données.

Il a été utilisé avec succès comme source de données pour l'analyse de texte, l'exploration de sentiments et d'opinions, la modélisation de sujets, la classification et la synthèse de texte, etc.

Les utilisateurs de Twitter partagent des informations opportunes sur divers événements en cours, reflétant généralement leurs opinions personnelles, leurs réactions émotionnelles et leurs points de vue controversés. Presque toute personne impliquée ou suivant l'événement peut partager des informations en temps réel. Par conséquent, au fur et à mesure que l'événement se déroule, ces informations peuvent atteindre n'importe où dans le monde. Par conséquent, les médias sociaux peuvent être considérés comme une source précieuse d'informations à jour générées par les groupes d'utilisateurs dans le contexte de presque tous les événements.

Bien que la détection d'anomalies dans les séries chronologiques soit un domaine de recherche mature, elle a récemment commencé à être appliquée pour détecter des anomalies basées sur les émotions dans une grande quantité de données en continu. Les anomalies basées sur les émotions sont définies comme une augmentation soudaine de la série chronologique de tweets liés à des émotions positives, neutres ou négatives.

L'objectif de cette recherche est de développer et d'évaluer une technologie qui détecte automatiquement les anomalies en se basant sur l'apprentissage profond. Tout d'abord, nous utilisons des algorithmes ML pour classifier les textes. Ensuite, une classification des sentiments est utilisée pour diviser le flux de données d'entrée en trois flux indépendants (positif, neutre et négatif). Puis, nous proposons deux modèles de détections d'anomalies : le modèle ARIMA, basé sur le modèle ARMA, qui s'adapte aux séries chronologiques non stationnaires et le modèle LSTM qui représente une classe de réseaux de neurones récurrents, afin d'analyser les pics anormaux du nombre de tweets. Ces méthodes ont été testées sur cinq ensembles de données différents qui seront bien décrits dans ce rapport.

Finalement, nous résumons et présentons des perspectives de chacune de nos réalisations.

Chapitre 0 : Contexte du projet

Introduction

A travers ce premier chapitre, nous allons exposer en premier lieu une étude préliminaire du projet, son cadre général, l'environnement de travail à savoir l'organisme d'accueil (UTT) et sa mission. J'exposerai ensuite une description de mon projet en précisant son contexte et ses objectifs majeurs. Dans la partie suivante, nous décrirons la méthodologie choisie dans ce rapport. Nous terminerons le premier chapitre avec les différents outils et technologies utilisés.

I. Cadre du projet

Dans le cadre de ma formation d'ingénieurs Informatique à l'École Supérieure Privée d'ingénierie et de Technologies (ESPRIT), j'ai eu l'opportunité de mettre en œuvre un projet au sein de l'UTT, afin de compléter mes études universitaires et mettre en pratique et bien utiliser mes aptitudes et mes compétences.

1. Présentation de l'UTT

Établissement public créé à Troyes en 1994, l'UTT est aujourd'hui parmi les 10 écoles d'ingénieurs les plus importantes en France.[1]

Elle forme plus de 3100 étudiants chaque année, de postbac à bac+5 et bac+8.

À la fois université et grande école, l'UTT s'adosse à ses 8 équipes de recherche pour proposer des formations couvrant tout le spectre universitaire : Licence, Master, Ingénieur et Doctorat, des formations courtes professionnalisantes (Diplômes d'Université), des programmes de Mastère spécialisé®, de la VAE et des certifications en langues.[2]

UTT-Troyes, un modèle à forte intensité de recherche qui articule la recherche fondamentale, la recherche disciplinaire et la recherche technologique dont le but est de répondre aux grands défis sociétaux. [3]

Les activités de recherche sont menées par des professeurs permanents, des associés, des chercheurs de projet, des PAST, des doctorants et des employés techniques et administratifs qui travaillent dans les domaines des sciences de l'ingénieur, des sciences et technologies de l'information et de la communication.[3]

Parmi les plateformes scientifiques et technologiques on peut citer ;

- Nano'mat : Plateforme de nano fabrication et nano caractérisation des matériaux pour l'optique, la mécanique, la biologie et les agro-ressources.
- CapSec : Capteurs dédiés à la sécurité.
- EcoCloud : Analyse et évaluation des impacts environnementaux.
- CyberSec : Cyber Sécurité.
- Living Lab ActivAgeing : Design et évaluation de solutions technologiques pour l'autonomie des personnes âgées.
- Num3D : Numérisation 3D et ingénierie virtuelle.

- Adhère : Élaboration et caractérisation de dépôts.
- PRESAGES : Plateforme de Recherche d'Expérimentation et de Simulation des Activités de Gestion des Évènements de Sécurité.

La recherche regroupe plus de 360 personnes, dont 123 enseignants-chercheurs, 186 doctorants, 20 personnels techniques et administratifs.[4]



Figure 1: Université de Technologie de Troyes [5]

2. Présentation du CapSec

Géré par l'équipe LM2S, CapSec est une plateforme de réseau de capteurs sans fil embarqués. Son objectif principal est de fournir une ressource clé aux partenaires industriels et aux laboratoires académiques pour tester et valider leurs solutions technologiques basées sur des réseaux de capteurs sans fil.[6]



Figure 2: Le département X (CapSec)[7]

II. Objectif du Projet

Afin d'assurer la surveillance des événements sportifs des Jeux Olympiques 2024, l'UTT participe avec le laboratoire LICIT (Laboratoire ingénierie, circulation, transports) qui est implanté sur les sites de l'IFSTTAR (L'Institut français des sciences et technologies des transports, de l'aménagement et des réseaux)-ENTPE (L'École nationale des travaux publics de l'État) et les équipes d'ORANGE-Lab au projet ANR DISCRET : L'Agence nationale de la recherche qui est une agence de moyens, qui finance la recherche publique et la recherche partenariale en France. Et DISCRET c'est Démonstrateur d'Identification de Situations Critiques via la Remontée de données multi sources pour l'alErte en Temps-réel. Son objectif est de démontrer la possibilité de détecter et de localiser, en temps réel, des situations inhabituelles ou critiques dans les zones urbaines (par exemple, attaques, incendies, conditions météorologiques soudaines ou événements liés à la foule, etc.). Basée de l'analyse des données du réseau de sondes de téléphones portables et celle des services utilisés par les abonnés (réseaux sociaux, divertissement, etc.) tout en bénéficiant de la puissance de l'apprentissage automatique. L'UTT utilise des modules de sa plateforme TweetCapt® pour compléter et contextualiser la détection de situations anormales par les données de téléphonie mobile. Il s'agit alors de partir des données collectées sur Twitter pour documenter la situation détectée par la partie mobile (tenue par Orange et IFSTTAR).

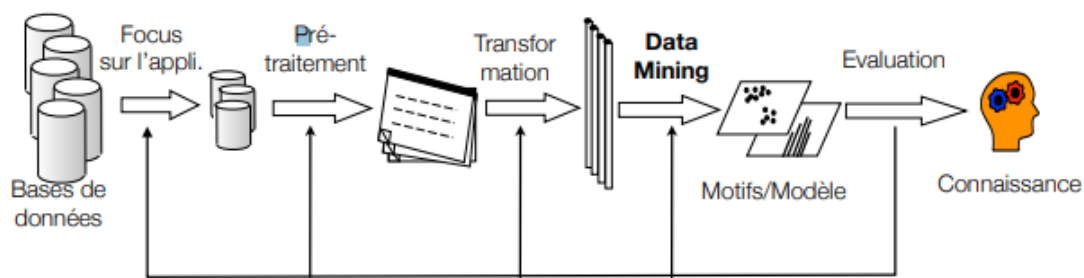
III. Méthodologie du projet

1. Méthodologies

a. KDD

KDD (Extraction de Connaissances à partir des Données) est une méthode permettant aux experts d'extraire des modèles et/ou des informations requises à partir de données. Il se compose de cinq étapes : sélection, prétraitement, transformation, datamining et interprétation/évaluation. [11] (Voir annexe 0 pour plus de détails)

Le processus KDD



Processus itératif et interactif

Figure 3: Processus de la méthode KDD [53]

b. SEMMA

SEMMA (sample, explore, modify, model, assess) qui se traduit en français par : échantillonne, explore, modifie, modélise, évalue). [54] Il a une structure similaire à KDD, mais comme il ne se concentre pas trop sur les étapes spécifiques aux données, il est plus facile à appliquer aux tâches générales de la science des données. De plus, contrairement à KDD, il a un caractère strictement cyclique. [54] (Voir annexe 0 pour plus de détails)

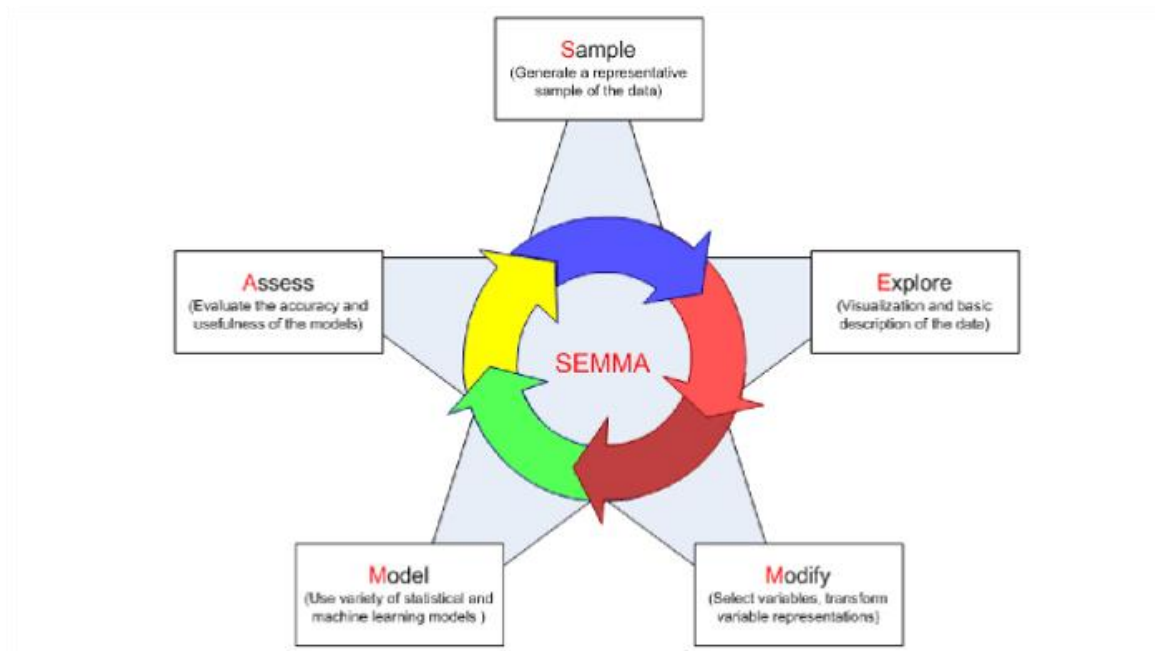


Figure 4: Processus de la méthode SEMMA [11]

2. Choix

a. CRISP-DM

Pour structurer cette étude, nous avons choisi la méthode CRISP-DM.

La méthode CRISP, appelée CRISP-DM à sa création en 1996 par IBM, a été conçue à la base pour des projets de datamining. Totalement indépendante des outils et technologies utilisés en entreprise, cette méthode doit son succès et sa généralisation à tous projets de big data grâce à son schéma d'application standard. Bâtie en six étapes, la méthode CRISP est idéale pour résoudre un problème défini tant elle met l'accent sur l'identification des besoins et sur les objectifs métiers. [12] (Voir annexe 0 pour plus de détails).

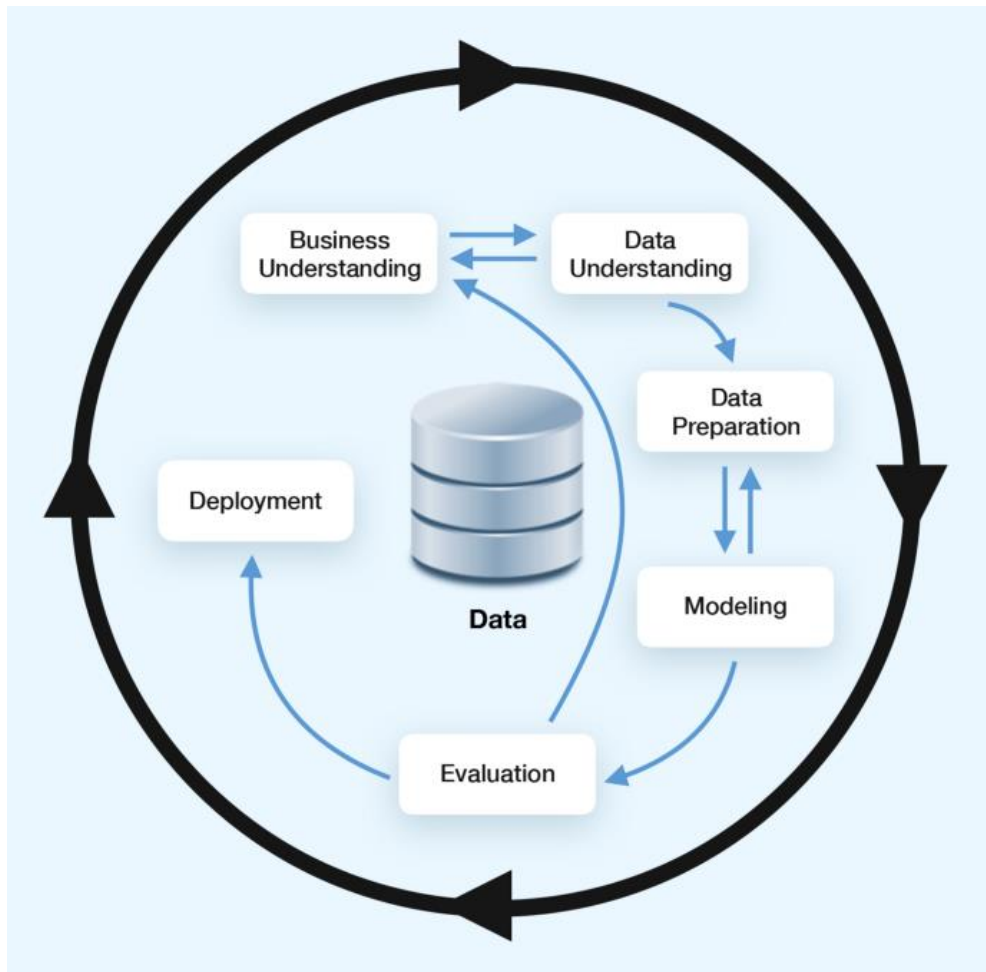


Figure 5: Processus de la méthode CRISP-DM [11]

3. Comparaison

KDD et SEMMA sont presque identiques. Chaque étape de KDD correspond directement à une étape de SEMMA; le processus CRISP-DM combine l'étape de prétraitement sélectif (KDD) ou d'exploration d'échantillons (SEMMA) à l'étape de compréhension des données. Il comprend également la phase de compréhension et de déploiement de l'entreprise. [11]

La principale différence de structure entre CRISP-DM et les deux autres méthodes est que les transitions entre les étapes peuvent être inversées. [11] Ainsi, si les experts constatent que les données ne sont pas suffisantes pour résoudre l'objectif du projet dans la phase de modélisation, ils peuvent revenir à la phase de préparation des données, sélectionner différentes variables cibles, générer des fonctionnalités, etc., au lieu de revenir au début de la boucle. [11] Pour cette raison que nous avons choisi cette méthode.

Chapitre 1 : Compréhension du métier

Introduction

La première étape de chaque projet basé sur l'approche CRISP-DM est la compréhension du métier, qui se concentre sur la compréhension des objectifs et des exigences du projet. En résumé, ce seraient les piliers des prochaines étapes de ce projet. Nous terminerons le premier chapitre avec les différents outils et technologies utilisés.

I. Objectif métier

L'objectif principal de ce stage est de réaliser un système de détection d'anomalies qui détecte et localise des situations inhabituelles ou critiques dans les zones urbaines.

II. Objectifs Data science

Les objectifs techniques sont parallèles aux objectifs métiers. Rien ne peut se faire sans l'autre. Nous avons donc utilisé différentes techniques :

Prétraitement et extraction des caractéristiques : Cette étape consiste à nettoyer les données fournies afin de leur donner du sens et de les rendre plus pertinentes.

Machine Learning et Deep Learning : Cela se traduit par l'application de plusieurs modèles d'apprentissage pour la classification des données et la détection d'intrusions.

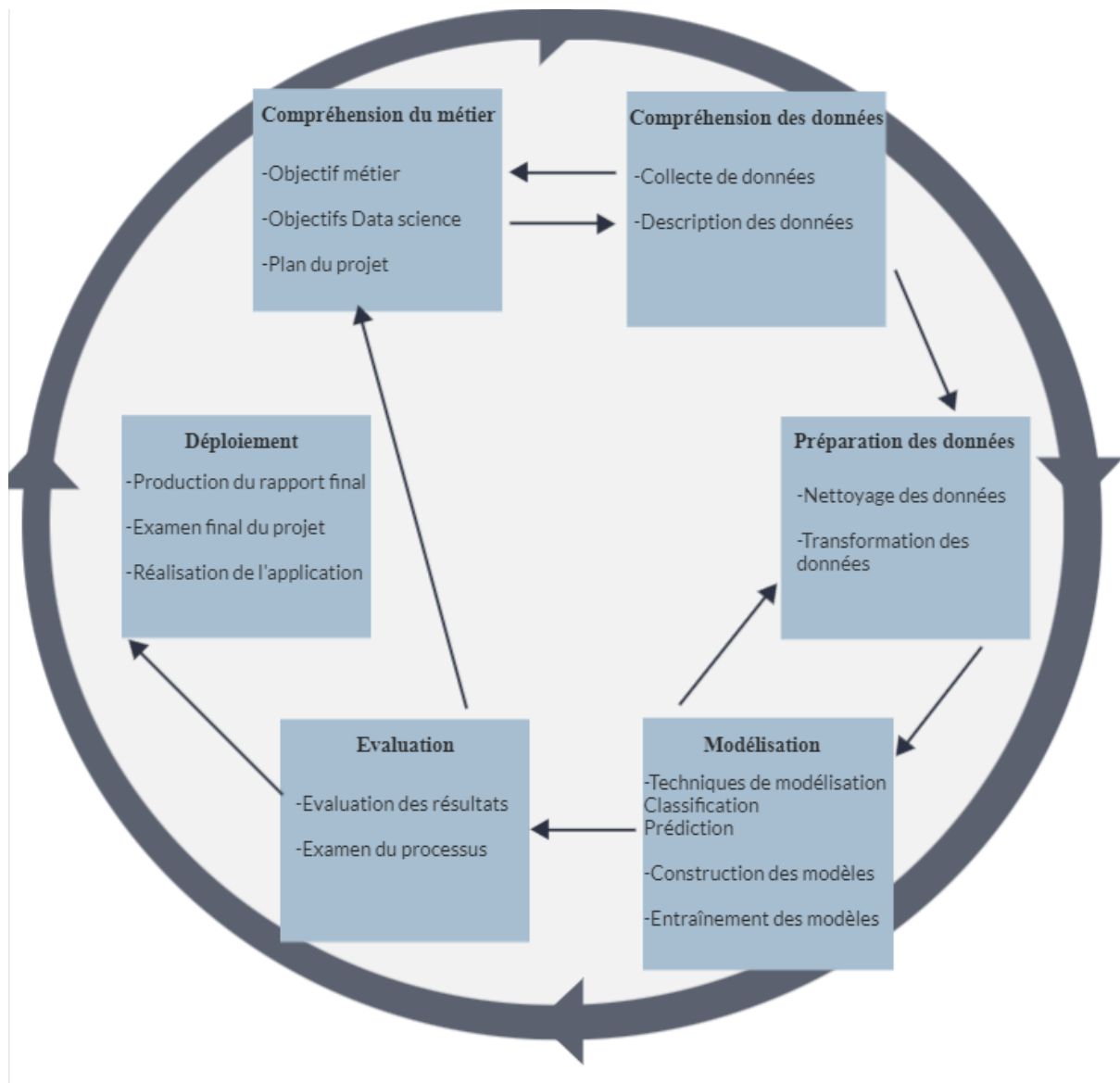
Visualisation : Des visualisations qui résument les données et vous aident à mieux comprendre.

III. Plan du projet

Cette partie vise à décrire les principales étapes qui auraient lieu au cours de ce projet. De plus, nous parlerons des différents outils et technologies utilisés pour atteindre les objectifs d'exploration de données et l'objectif métier.




1. Étapes de planification du projet





Comme précisé ci-dessus, le plan qui sera utilisé dans ce projet est CRISP-DM qui, grâce à sa spécification agile, peut nous faire atteindre nos objectifs facilement.



2. Outils et technologies utilisés

Tableau 1: Outils et technologies utilisés

Outils, technologies et langage	Définitions
<p>Google Colab ou Colaboratory</p>  <p>[13]</p>	<p>C'est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur votre ordinateur à l'exception d'un navigateur. [13]</p>
<p>Jupyter Notebook</p>  <p>[13]</p>	<p>C'est une application Web Open Source permettant de créer et de partager des documents contenant du code (exécutable directement dans le document), des équations, des images et du texte. Avec cette application il est possible de faire du traitement de données, de la modélisation statistique, de la visualisation de données, du Machine Learning, etc. Elle est disponible par défaut dans la distribution Anaconda. [13]</p>
<p>Python (Environnement : Jupyter kernel)</p>  <p>[14]</p>	<p>Un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl. [14]</p>

<p>Power BI</p>  <p>[16]</p>	<p>C'est une solution de Business Intelligence développée par Microsoft pour permettre aux entreprises d'agréger, d'analyser et de visualiser les données en provenance de sources multiples. [15]</p>
<p>Microsoft Teams</p>  <p>[18]</p>	<p>C'est un hub de collaboration qui reprend les fonctionnalités de Microsoft suivantes :</p> <p>Exchange : logiciel de travail de groupe associé à une messagerie électronique</p> <p>SharePoint : application permettant le partage d'informations entre plusieurs utilisateurs</p> <p>Skype Entreprise ou Lync : plateforme de communication « unifiée » intégrant les moyens de communication comme les appels vidéo ou audio et la messagerie instantanée. [17]</p>
<p>Slak</p>  <p>[19]</p>	<p>C'est une plate-forme de communication collaborative propriétaire ainsi qu'un logiciel de gestion de projets créé par Stewart Butterfield en août 2013 et officiellement lancée en février 2014. [19]</p>
<p>Github</p>  <p>[8]</p>	<p>C'est un service de gestion d'hôte virtuel et de développement logiciel qui permet à ses utilisateurs de gérer et stocker leurs projets. Il assure également le suivi et le contrôle des modifications qui y sont apportées</p>

Conclusion

L'étape de compréhension du métier et l'ensemble du domaine nous aide à nous immerger dans cette étude. Cela nous permet d'aborder les prochaines étapes solides et bien informées dans le domaine. Sans ce passage, tout ce qui va suivre nous sera plus difficile à attaquer.

Chapitre 2 : Compréhension des données

Introduction

Lors de l'élaboration de la méthodologie CRISP, une question s'est posée : quelles données nous aurons besoin ? C'est propre ? Et maintenant il faut répondre. Pour aller plus loin, l'étape de collecte de données sera nécessaire. Nous finirons par une description de tout ce qui nous est fourni dans nos ensembles de données afin de vérifier leurs qualités.

I. Collecte de données

Cette partie est réalisée par mon collègue « Dame Seck » via l'API Twitter. Ce dernier est un service public fourni par Twitter qui permet d'envoyer des requêtes automatiques au sein des tweets et qui peut être utilisé pour surveiller ou analyser le contenu des tweets.

Dans cette étude nous avons exploité 5 ensembles de données :

1. French Tweet

Le premier jeu de données est intitulé « french_tweets ». Un ensemble de données étiquetées qui est accessible pour le grand public sur la plateforme Kaggle sous forme d'un fichier « csv ». Ce fichier se compose de 1 526 724 lignes et 2 colonnes : la colonne « text » qui contient des tweets en langue française et la colonne « label » qui représente le sentiment de chaque tweet ('0' pour le sentiment négatif et '1' pour le sentiment positif) avec 49% des tweets qui sont positifs et les restes qui sont négatifs.

	label	text
0	0	Seulement si vous avez l'intention d'être assa...
1	1	Bonjour, twitterville.
2	1	Je suis allé à une fête de fiançailles aujourd...
3	0	Perdre mon temps à rester au repos à la maison
4	1	Là, j'ai posté l'original? ~ 6h2ns
5	0	Et il est confondu avec le non-aller-à-mcfly's...
6	1	Même ici lol
7	0	Le travail va être l'enfer. 8 heures puis dorm...
8	1	Vous devez arrêter la sissyine et m'ajouter su...
9	1	Certains d'entre eux sont vraiment vraiment ég...
10	1	Je t'aime morgan freeman, tu seras toujours di...
11	0	je suis perdu. Aidez-moi à trouver une bonne m...
12	1	- les nouveaux cheveux sont mignons
13	1	Après la fête du bal a été greaaat
14	1	Pourquoi merci! C'est un bday amère. Difficile...
15	1	Descendre le rivage le matin
16	1	Mmm ça a l'air bien! Ou un blizzard butterfing...
17	0	Je veux que les gens partent afin que je puiss...
18	0	Enfin dans la. Mais est-il en train d'arracher...

Figure 6: Extrait de l'ensemble de données "french_tweets"

Pour les ensembles qui suivent, nous les avons obtenus par la collecte des tweets.

2. Notre-Dame :

La collecte de l'incendie de Notre-Dame de Paris : L'incendie majeur qui est survenu à la cathédrale Notre-Dame de Paris, les 15 et 16 avril 2019, pendant près de 15 heures.

Cet ensemble de données se compose de tweets français contenant au moins l'un des mots suivants : "NotreDame", "Paris", "Cathédrale", "feu", "pompiers".

La période de temps associée aux tweets va du "15-04-2019 17:30" au "18-04-2019 18:00", ce qui correspond au temps écoulé entre le départ estimé de l'incendie et la fin de ce désastre.

Les données sont sous la forme d'un fichier json. Ce dernier se compose de 9 082 497 lignes et 4 colonnes :

1. Id_str : l'id du tweet
2. Created_at : la date et l'heure du tweet
3. Lang : la langue du tweet qui est toujours le français
4. Full_text : le contenu du tweet

id_str	created_at	lang	full_text
1118369241827368960	2019-04-17 04:23:00+00:00	fr	«Les pompiers de @Paris ont fait un travail re...
1118369176706519040	2019-04-17 04:22:44+00:00	fr	Image bouleversante qui m'a été partagée par u...
1118369117910835200	2019-04-17 04:22:30+00:00	fr	Un ami pompier vient de m'envoyer une vidéo du...
1118369087711797248	2019-04-17 04:22:23+00:00	fr	👏 Magnifique Chant 🙏🏠📺\n\n🎵🎶 Nous te saluons,...
1118369067331661824	2019-04-17 04:22:18+00:00	fr	Un ami pompier vient de m'envoyer une vidéo du...
...
1118211378496643072	2019-04-16 17:55:42+00:00	fr	👏 La Thèse accidentelle de l'incendie de #Notr...
1118211371416600576	2019-04-16 17:55:40+00:00	fr	Image bouleversante qui m'a été partagée par u...
1118211363866861568	2019-04-16 17:55:39+00:00	fr	Un GIGAPIXEL au drone sur #NotreDame pour con...
1118211361480302592	2019-04-16 17:55:38+00:00	fr	Un ami pompier vient de m'envoyer une vidéo du...

Figure 7: Extrait de l'ensemble de données "Notre Dame"

3. Collecte Paris avril 2021

La deuxième collecte est celle de Paris avril 2021 pendant laquelle il y a eu de l'attaque du 23 avril 2021 à Rambouillet : Un attentat terroriste islamiste perpétré au couteau au commissariat de Rambouillet. L'agresseur a fait de la victime une assistante administrative au poste de police.

La période de temps associée aux tweets va du "01-04-2021 09:00" au "30-04-2021 12:00".

Cette collecte est aussi sous la forme d'un fichier json. Ce dernier se compose de 5434802 lignes et 4 colonnes :

1. Id_str : l'id du tweet
2. Created_at : la date et l'heure du tweet
3. Lang : la langue du tweet qui est toujours le français
4. Full_text : le contenu du tweet

id_str	created_at	lang	full_text
1382113500311515136	2021-04-13 23:28:31+00:00	fr	⚡FRFLASH -Une famille poursuit l'Etat après le...
1382113501687246848	2021-04-13 23:28:31+00:00	fr	@MagicCaylloux @ClippersFR @TomCiaravino @LeSt...
1382113503339819008	2021-04-13 23:28:32+00:00	fr	« Un enfant a été enlevé, ceci une alerte enlè...
1382113503490797568	2021-04-13 23:28:32+00:00	fr	Cet homme viens de faire un accident et il est...
1382113503423643648	2021-04-13 23:28:32+00:00	fr	Victoire pour paris https://t.co/MUq4zexUj
...
1382134502860947456	2021-04-14 00:51:58+00:00	fr	MERCI DE NE PAS FAIRE D'APPELS DE PHARE QUAND ...
1382134502944821248	2021-04-14 00:51:58+00:00	fr	@neymarjr @LParedss LE ROI ICI C'EST PARIS ❤️💙...
1382134506950438912	2021-04-14 00:51:59+00:00	fr	125 élus pareils....\n👤\nPkoï on vote?
1382134508359663616	2021-04-14 00:52:00+00:00	fr	FR [ALERTE — À RELAYER] L'#alerteenlèvement co...
1382134509290844160	2021-04-14 00:52:00+00:00	fr	macron devant le décompte des morts https://t...

Figure 8: Extrait de l'ensemble de données "Paris avril 2021"

4. Collecte Paris mai 2021

Cet ensemble de données se compose de tweets français pendant laquelle il y a eu plusieurs évènements autour du 1er et du 15 mai 2021

La période de temps associée aux tweets va du « 01-05-2021 08:27" au « 25-05-2021 19:40 ".

Cette collecte est aussi sous la forme d'un fichier json. Ce dernier se compose de 2 775 158 lignes et 3 colonnes : Id_str ,Created_at , keyword.

	created_at	id_str	keyword
0	2021-05-01 08:27:34+00:00	1388409751852273666	police
1	2021-05-01 08:27:37+00:00	1388409762853830657	feu
2	2021-05-01 08:27:38+00:00	1388409766171529217	feu
3	2021-05-01 08:27:38+00:00	1388409767064965120	attaque
4	2021-05-01 08:27:38+00:00	1388409767064965120	paris
...
1019	2021-05-25 09:44:17+00:00	1397126363820220419	chute
1020	2021-05-25 09:44:17+00:00	1397126363983798272	paris
1021	2021-05-25 09:44:17+00:00	1397126364743049216	chute
1022	2021-05-25 09:44:17+00:00	1397126366441709568	paris
1023	2021-05-25 09:44:17+00:00	1397126367360266244	paris

2775158 rows × 3 columns

Figure 9 : Extrait de l'ensemble de données "Paris mai 2021"

5. Orages Île-de-France juin 2021

Cet ensemble de données se compose de tweets français qui contient des évènements météorologiques autour du 19 et 20 juin 2021

La période de temps associée aux tweets va du « 14-06-2021 12:03" au « 23-06-2021 12:09 ".

Ce fichier se compose de 1 397 891 lignes et 3 colonnes : Id_str ,Created_at , keyword.

	created_at	id_str	keyword
0	2021-06-14 12:03:11+00:00	1404378879486304260	feu
1	2021-06-14 12:03:12+00:00	1404378885190467584	lyon
3	2021-06-14 12:03:13+00:00	1404378886851416068	alerte
5	2021-06-14 12:03:13+00:00	1404378888831131649	accident
2	2021-06-14 12:03:13+00:00	1404378885337321474	paris
...
4508	2021-06-23 12:09:49+00:00	1407642038888173573	paris
4509	2021-06-23 12:09:49+00:00	1407642040708501504	orage
4510	2021-06-23 12:09:49+00:00	1407642040972812292	seisme
4511	2021-06-23 12:09:50+00:00	1407642044156284928	paris
4512	2021-06-23 12:09:51+00:00	1407642047855665159	paris

1397891 rows × 3 columns

Figure 10 : Extrait de l'ensemble de données (Orages Ile de France 2021)

Conclusion

Il est vrai que tout au long de cette étape, nous avons réussi à comprendre nos données et à préparer tout le terrain favorable pour le travail. Mais tout ce que nous économisons n'est pas suffisant pour faire une prédiction parfaitement précise, car toutes les données ne sont pas pertinentes. Il y a des textes bruts qui contiennent beaucoup d'aléatoire qui affectent à l'estimation des modèles : les accents, les minuscules, les signes de ponctuation, les caractères spéciaux... Tout cela nous amène à une prochaine étape, très importante dans projets de science des données qui est la préparation de données

Chapitre 3 : Préparation des données

Introduction

Comme mentionné dans le chapitre précédent, notre problème repose sur les données de sentiment fournies par les réseaux sociaux Twitter. Ainsi, nos entrées sont en langage naturel et nécessitent un traitement spécifique à travers différentes tâches : tokenisation, normalisation de mots, lemmatisation et stemming ... La partie suivante est la partie de représentation des documents où il y a toutes les transformations de textes réalisées avant d'entamer l'étape de modélisation. Nous terminons ce chapitre par l'approche de validation qui est une étape nécessaire avant de commencer l'entraînement.

I. Prétraitement des données

La phase de nettoyage se déroule sur plusieurs étapes comme suit :

1. Écrire tout en minuscule,
2. Suppression des retours à la ligne,
3. Suppression des pseudo Twitter (exp : utilisateur @Utilisateur),
4. Suppression d'adresse url (modèle 'http :'),
5. Suppression des mentions Twitter (exp : @Paris),
6. Suppression des hashtags (exp : #NotreDame),
7. Suppression des chiffres et des caractères spéciaux (exp : « , ; : ! ? ' & \$ % () [] + - * / »),
8. Appliquer la lemmatisation et le stemming
9. Suppression des stop-words
10. Tokenization et suppression des short-words


```

class TweetCleaning():

    @staticmethod
    def remove_line_return(text):
        """Removes all line returns '\n' from text (str)"""
        return text.replace('\n', ' ')

    @staticmethod
    def remove_retweet_handle(text):
        """Removes the '@user_handle' from a text."""
        return re.sub(r'rt[\s]+', '', text)

    @staticmethod
    def remove_urls(text):
        return re.sub(r'https?:\/\/.*[\r\n]*', '', text)

    @staticmethod
    def remove_mentions(text):
        return re.sub('(@[A-Za-z]+[A-Za-z0-9-_-])', '', text)

    @staticmethod
    def remove_hashtags(tweet, pattern1, pattern2):
        r = re.findall(pattern1, text)
        for i in r:
            text = re.sub(i, '', text)

        r = re.findall(pattern2, text)
        for i in r:
            text = re.sub(i, '', text)
        return text

    @staticmethod
    def clean(text):

        text = text.lower()
        text = TweetCleaning.remove_line_return(text)
        text = TweetCleaning.remove_retweet_handle(text)
        text = TweetCleaning.remove_urls(text)
        text = TweetCleaning.remove_mentions(text)
        text = TweetCleaning.remove_hashtags(text, "# [\w]*", "#[\w]*")
        return text

```

Figure 11 : Code de source de la partie prétraitement des données (1)

```

@staticmethod
def remove_nonalphanumeric(text):
    accepted_characters = 'a-zôââçèéêîôû'
    r = re.compile('[^{} ]'.format(accepted_characters))
    return r.sub('', text)

lang_to_corpus = {
    'fr': 'fr_core_news_md'
}
lang_to_stop_words = {
    'fr': set(nltk.corpus.stopwords.words('french') + fr_supp_stop_words)
}

def __init__(self, lang):
    self.nlp = spacy.load(TweetCleaning.lang_to_corpus[lang])
    self.stopwords = TweetCleaning.lang_to_stop_words[lang]

@staticmethod
def remove_stop_words(self, text):
    tokens = nltk.word_tokenize(text)
    tokens = [token for token in tokens if not token in self.stopwords]
    return ' '.join(tokens)

@staticmethod
def tokenize(text, method='nltk'):
    if method=='nltk':
        return nltk.word_tokenize(text)

@staticmethod
def lemmatizer(self, text):
    lem = []
    doc = self.nlp(text)
    for word in doc:
        lem.append(word.lemma_)
    return ' '.join(lem)

@staticmethod
def remove_short_words(text):
    tokens = nltk.word_tokenize(text)
    tokens = [token for token in tokens if len(token)>=4]
    return ' '.join(tokens)

def process_text_pipeline(self, t):
    cl_text = self.clean(t)
    lem_cl_text = self.lemmatizer(cl_text)
    lem_cl_text = self.remove_nonalphanumeric(lem_cl_text)
    lem_cl_filtered_text = self.remove_stop_words(lem_cl_text)
    lem_cl_filtered_long_text = self.remove_short_words(
        lem_cl_filtered_text)
    return (
        cl_text, lem_cl_text, lem_cl_filtered_text, lem_cl_filtered_long_text)

def process_text(self, t):
    return self.process_text_pipeline(t)[-1]

```

Figure 12 : Code de source de la partie prétraitement des données (2)

```

|
cl= TweetCleaning(lang='fr')
df['clean_text'] = df['full_text'].apply(lambda x: cl.process_text(x))

df

```

«Les pompiers de @Paris ont fait un travail re...	pompier faire travail remarquable juge colonel...
Image bouleversante qui m'a été partagée par u...	image bouleversant partager personne dont neve...
Un ami pompier vient de m'envoyer une vidéo du...	ami pompier venir envoyer vidéo haut cathédral...
👉 Magnifique Chant 🙏🏠📺🎵🎶 Nous te saluons,...	magnifique chant saluer notredame résonne plac...
Un ami pompier vient de m'envoyer une vidéo du...	ami pompier venir envoyer vidéo haut cathédral...
...	...
👉 La Thèse accidentelle de l'incendie de #Notr...	thèse accidentel incendie notredame ancien ing...
Image bouleversante qui m'a été partagée par u...	image bouleversant partager personne dont neve...
Un GIGAPIXEL au drone sur #NotreDame pour con...	gigapixel drone notredame constater dégât notr...
Un ami pompier vient de m'envoyer une vidéo du...	ami pompier venir envoyer vidéo haut cathédral...

Figure 13: Un extrait de données avant et après le pré traitement

II. Représentation des documents

Certaines technologies ont des exigences spécifiques pour la forme de données. Par conséquent, il est généralement nécessaire de passer par la représentation des documents après l'étape de nettoyage des données.

Afin de réaliser cette étape, nous avons utilisé 3 modèles de représentation de texte. (Voir annexe 1 pour plus de détails sur chaque modèle)

1. Représentation par sac de mots

Ci-dessous, un exemple de la représentation Bag of Words :

```
from sklearn.feature_extraction.text import CountVectorizer
#Bag of words

bow_vectorizer = CountVectorizer(max_features=10000)
vectors = bow_vectorizer.fit_transform(unpickled_df['cleantweet']).todense()

vocab = pd.DataFrame([bow_vectorizer.vocabulary_])
vocab
```

	in sentence
work	3131
vraiment	3081
fatigué	1143
encore	1018
une	2980
...	...
dodge	884
caravan	499
tardives	2819
devant	823
shode	2630

Figure 14: Exemple représentation BOW

2. Représentation par Word2Vec

Ci-dessous, un exemple de la représentation Word2Vec.

```
from gensim.models import Word2Vec
tokenized_tweet = df[['tokenized_tweet']]

model_w2v = gensim.models.Word2Vec(
    tokenized_tweet,
    size=200, # desired no. of features/independent variables
    window=15, # context window size
    min_count=2, # Ignores all words with total frequency lower than 2.
    sg = 1, # 1 for skip-gram model
    hs = 0,
    negative = 10, # for negative sampling
    workers= 10, # no.of cores
    seed = 5
)

model_w2v.train(tokenized_tweet, total_examples= len(df['tokenized_tweet']), epochs=20)
```

Figure 15: Représentation Word2Vec

La figure ci-dessous nous montre la représentation Word2Vec du mot “difficile” :

```
# access vector for one word
from gensim.models import Word2Vec

vec = model_w2v.wv['difficile']
print(vec)
print(len(vec.tolist()))
```

```
[ 0.591768  0.49852157 -0.8075315  0.07227271 -0.2010611  0.3255881
-0.58028954 0.258532  0.1703059  0.1255826  0.34262758 -0.89381653
-0.00089967 -0.39633593 0.01445668 0.24596174 0.4985549 -0.13240826
0.13813108 -0.14262894 -0.39639845 -0.37943754 -0.14999345 0.3689228
-0.23712474 0.30315632 -0.65050185 0.05422625 -0.38110393 0.09294961
0.422006 -0.5701797 0.34163794 -0.03215797 0.09770731 -0.03386511
0.7707957 0.3790158 -0.23132537 -0.16039145 -0.2779246 -0.27478948
-0.8585178 0.53754586 -0.0570287 -0.0557829 -0.59831744 0.10488334
0.06355681 -0.06330421 0.20697628 -0.3162729 0.06680043 0.4290997
0.22692765 0.290153 -0.02243448 0.26377898 -0.15391319 -0.09453639
0.53614736 0.10463269 0.10506908 -0.17829235 0.00912608 -0.36133865
0.6359876 -0.78881204 -0.24620505 -0.0385017 -0.3168906 -0.17681998
0.0774869 -0.44733042 -0.24256398 0.1940978 -0.5835807 -0.38924965
-0.06683763 0.34440383 0.03379973 -0.05238726 -0.06751922 0.18006466
0.7944244 0.40473366 -0.02084917 -0.25913057 -0.17782286 0.16395451
-0.5420964 0.26527232 -0.08149093 -0.2942028 0.04152378 -0.01239694
-0.37213144 -0.12702633 -0.44941932 0.18317863 -0.73655623 -0.33316633
-0.33488742 -0.29864362 -0.06131997 -0.26788452 0.42837706 -0.4480648
-0.3086153 -0.2437071 0.39857033 0.39745373 0.48935255 -0.27501735
0.26883137 -0.5116149 -0.06602766 0.20254937 -0.42422998 -0.06389779
-0.13485168 0.09302993 -0.10159428 -0.19176587 0.5496295 -0.5775727
0.19793496 -0.12800665 -0.01054717 -0.07939072 0.08994389 0.04476302
-0.7448629 -0.2967951 -0.05284043 0.03169372 0.07838693 0.46033016
-0.22413094 0.08555877 0.09652608 0.28328347 0.6988406 0.4192989
0.5612326 -0.20453747 -0.09573516 -0.6997198 0.26101038 0.12529011
-0.21850494 -0.31422573 0.727717 -0.5089916 0.48096392 0.04108565
-0.13693173 -0.10887483 -0.44223362 -0.24561457 0.39744878 -0.28437334
-0.39132136 -0.15170582 0.4455774 -0.3852107 0.16074274 -0.15087259
0.34184238 -0.14139062 0.00846803 0.12342114 0.14818549 0.18083312
-0.16584444 -0.11234675 0.11123894 0.12037742 -0.6873538 -0.39499068
0.22859485 -0.09588903 -0.18952078 -0.21630085 -0.3372396 0.09449234
0.39417863 -0.00527316 -0.09284621 -0.18471268 -0.3755152 0.32093105
0.15419687 -0.383241 0.06946462 0.370315 0.05961451 0.0624068
0.02076108 0.24102505]
```

200

Figure 16 : Exemple de la représentation Word2Vec

3. Représentation par TF-IDF

Ci-dessous, un exemple de la représentation tf-idf.

```
#TF-IDF = TF * IDF

tfidf_vectorizer= TfidfVectorizer(max_df=0.90, min_df=2,max_features =10000)
tfidf = tfidf_vectorizer.fit_transform(df['cleantweet'])
tf_trans = TfidfTransformer( smooth_idf=True , use_idf=True)
tf_trans.fit(tfidf)

df_idf=pd.DataFrame(tf_trans.idf_,index=tfidf_vectorizer.get_feature_names(),columns=['idf_weights'])
df_idf.sort_values(by=['idf_weights']).head(10)
```

Figure 17: Exemple de la représentation TF-IDF

	idf_weights
aller	3.232010
faire	3.324117
pouvoir	3.417639
devoir	3.899229
vouloir	3.964573
bien	3.987471
jour	4.089554
aujourd'hui	4.149708
voir	4.189963
venir	4.199791

Figure 18: Exemple de la représentation TF-IDF(2)

III. L'approche de validation

L'approche de validation est Train-Validation-Test Split. L'ensemble de données doit être divisé en 3 parties. La première (la plus grande) est utilisée pour l'entraînement. L'ensemble de validation est utilisé pour le modèle sélection. Le dernier ensemble est utilisé pour tester les performances du modèle sélectionné. Les différents modèles peuvent être les mêmes d'un point de vue général, mais leurs hyper paramètres diffèrent. Donc, ils ne sont techniquement pas les mêmes et se comportent différemment.

```

train_ratio = 0.70
validation_ratio = 0.15
test_ratio = 0.15

# train is now 70% of the entire data set
# the _junk suffix means that we drop that variable completely
x_train_bow, x_test_bow, y_train_bow, y_test_bow = train_test_split(vectors, label, test_size=1 - train_ratio)

# test is now 15% of the initial data set
# validation is now 15% of the initial data set
x_val_bow, x_test_bow, y_val_bow, y_test_bow = train_test_split(x_test_bow, y_test_bow, test_size=test_ratio/(test_ratio + validation_ratio))

print("X_train_shape : ", x_train_bow.shape)
print("X_test_shape : ", x_test_bow.shape)
print("y_train_shape : ", y_train_bow.shape)
print("y_test_shape : ", y_test_bow.shape)

X_train_shape : (3474, 2421)
X_test_shape : (745, 2421)
y_train_shape : (3474,)
y_test_shape : (745,)

```

Figure 19: L'approche de validation

Conclusion

Les outils de préparation des données nous ont permis de nettoyer les données avant de les analyser. Cela fournit une base solide et fiable pour nous aider dans les prochaines phases de notre projet, en particulier la modélisation en permettant au projet de s'exécuter de manière plus rapide et plus transparente.

Chapitre 4 : Modélisation

Introduction

A ce stade, diverses techniques de modélisation sont sélectionnées et appliquées, et leurs paramètres sont calibrés aux meilleures valeurs. Généralement, il existe plusieurs techniques pour le même type de problème d'exploration de données. Alors maintenant, nous nous tournons vers Le cœur du projet est de modéliser nos données, ce qui nous permettra d'atteindre nos principaux objectifs : la classification des tweets et la détection d'intrusions.

I. Classification des textes et analyse de sentiments

La classification est le type de problème le plus fréquemment étudié en ML et en raison de sa large application.

Cette partie représente la partie classification du projet où nous avons utilisé plusieurs algorithmes et techniques afin de détecter et analyse les sentiments des données collectées.

1. Techniques de modélisation

a) Techniques machine learning

Nous avons un échange à long terme de recherche sur les algorithmes de classification. Ce que nous recherchons vraiment, ce ne sont que des algorithmes puissants. Les résultats étaient variables et c'était à nous de décider, d'où venait le choix de la forêt aléatoire, la régression logistique et SVM (voir annexe 2 pour plus de détails).

b) Textblob

La fonction de TextBlob qui nous intéresse permet pour un texte donné de déterminer le ton du texte et le sentiment de la personne qui l'a écrit.

Après le nettoyage, on a une représentation en BOW donc une collection de mots et pour chaque mot on va lui attribuer un score (de -1 à 1) ensuite on calcule la moyenne et cette dernière représente le score de sentiments des phrases entrées (seuls les mots qui figurent dans le dictionnaire sont considérés). (Voir annexe 2 pour plus de détails)

c) CamemBERT

C'est un modèle deep learning qui représente un transformateur de type Bert que Facebook a construit pour la langue française. Ce dernier bat tous les records, il surpasse de nombreux modèles français [44]. (Voir annexe 2 pour plus de détails)

2. Mesure des performances

Nous avons utilisé une métrique commune, pour tous les modèles, utilisée dans le domaine de la classification. Chaque sortie est un vecteur de dimension n correspondant à n classes à prédire. Nous utilisons des indicateurs de précision pour mesurer la qualité du classificateur. Cela exprime le nombre de classes prédites correctes par rapport au nombre total de prédictions de classes effectuées. Nous choisissons d'utiliser des métriques strictes et nous considérons la sortie comme équitable uniquement lorsque toutes les catégories prédites sont correctes.

- **Matrice de confusion**

En apprentissage automatique supervisé, la matrice de confusion est une matrice qui mesure la qualité du système de classification. L'un de ses avantages est qu'elle peut rapidement montrer si le système de classification est capable classer correctement. [50]

		Classe réelle	
		-	+
Classe prédite	-	True Negatives (vrais négatifs)	False Negatives (faux négatifs)
	+	False Positives (faux positifs)	True Positives (vrais positifs)

Figure 20 : Matrice de confusion [51]

La performance :

La performance ou la justesse (accuracy en anglais) est l'un des critères d'évaluation des modèles de classification. De manière informelle, elle fait référence à la proportion de prédictions correctes faites par le modèle. [52]

Equation :

$$\text{Justesse} = \frac{VP + VN}{VP + VN + FP + FN}$$

Le rappel

Le rappel représente la capacité du classificateur à trouver tous les échantillons de la classe grâce à sa prédiction. [51]

Équation :

$$\text{Rappel} = \frac{TP}{TP + FN}$$

La précision

Par conséquent, nous nous intéresserons également à la précision qui mesure le ratio de prédictions précises pour chaque catégorie par rapport au nombre de prédictions faites pour chaque catégorie. [51]

Équation
$$Précision = \frac{TP}{TP + FP}$$

F-mesure :

Nous pouvons calculer la « mesure F », afin d'évaluer le compromis entre rappel et précision, qui est leur moyenne harmonique. [51]

Équation:
$$F - mesure = 2 \times \frac{Précision \times Rappel}{Précision + Rappel} = \frac{2TP}{2TP + FP + FN}$$

3. Construction des modèles

a) Réglages des paramètres

Afin d'affiner son algorithme et de l'adapter à l'ensemble de données, nous avons joué sur les paramètres des différents modèles.

Voici un aperçu sur ce que nous avons utilisé dans notre cas :

- **Paramètre Kernel**

Notre choix c'est le noyau linéaire étant le plus couramment utilisé pour la classification des textes. Ce dernier est utilisé lorsque le nombre de caractéristiques est important. De plus l'entraînement du noyau linéaire est plus rapide qu'avec un autre et demande moins de paramètres à optimiser. (Voir annexe 2 pour plus de détails sur les 3 types de noyaux).

- **Paramètre Cost**

. Le paramètre de coût appelé C nous indique dans quelle mesure nous voulons éviter de mal classer chaque observation dans l'ensemble d'apprentissage. Plus il est élevé, plus le bord de l'hyperplan est petit. Ceci est très important lorsque nos données sont non linéaires pour éviter les fausses prédictions. En revanche, plus il est petit, plus le bord de l'hyperplan est grand.

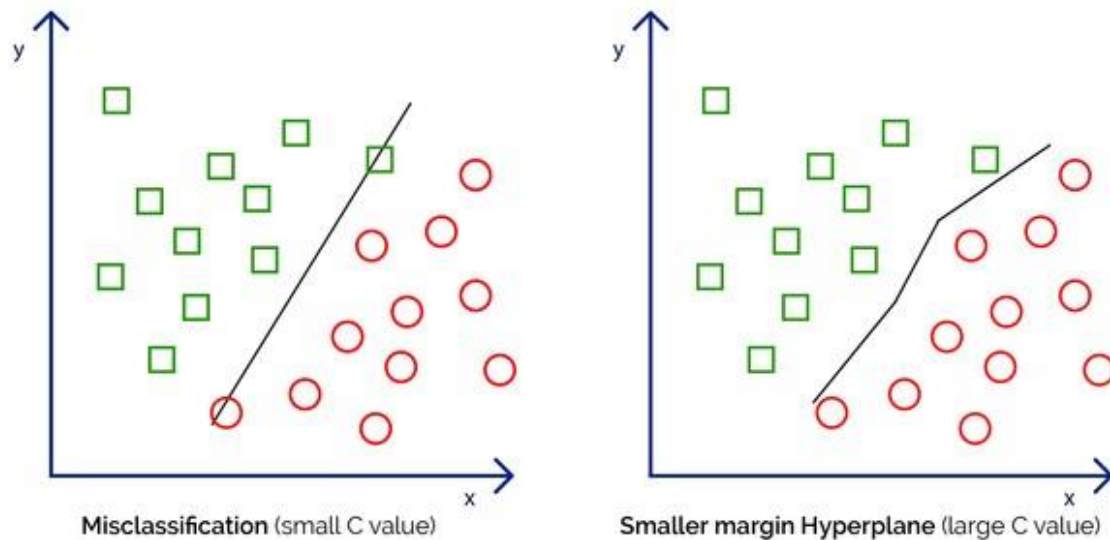


Figure 21: Paramètre coût (petite valeur /grande valeur) [57]

Dans notre projet, nous avons pris 3 valeurs C différentes pour voir à quel point cela a influencé notre score : 0.01 ; 1 ; 10

- **Paramètre `n_estimators`**

C'est le nombre d'arbres dans l'algorithme de forêt aléatoire qui indique à l'algorithme combien d'arbres doivent être cultivés.

Si le nombre d'observations est grand et que le nombre d'arbres est trop petit, alors certaines d'entre elles ne seront prédites qu'une seule fois, voire pas du tout. Dans notre exemple, nous commençons avec un grand nombre d'arbres, puis vérifions si la précision est stable, puis sélectionnons le nombre parfait d'arbres.

Ainsi, le nombre d'arbres retenus pour ce projet est de : 500.

- **Paramètre `multi_class`**

Nous avons choisi la régression logistique multinomiale avec des variables cibles de trois catégories nominales ou plus comme notre exemple de type de prédiction sentiment (positif / négatif / neutre). (Voir annexe 2 pour plus de détails sur les 3 types de RL).

b) Entraînement des modèles

Les données d'entrée pour les modèles de tous les classifieurs sont des documents. Pour prétraiter chaque classifieur, nous avons utilisé une vectorisation basée sur les modèles de représentation de texte mentionnés dans le chapitre précédent.

Les figures ci-dessous synthétisent les résultats des différents classifieurs sur l'ensemble de données "french_tweets", "Notre Dame "et "Paris avril 2021".

Pour la modélisation des techniques ML, nous avons utilisé le package Scikit-Learn qui est une bibliothèque Python gratuite pour l'apprentissage automatique.

- **SVM**

Le nombre total d'observations d'entraînement est de 350000.

Le nombre total d'observations testées est de 150000.

Le noyau utilisé est un noyau linéaire.

Le paramètre de coût est 0.01 ; 1 ; 10.

Après avoir entraîné le modèle pendant quelques minutes, nous devons l'appliquer à l'ensemble de test pour voir à quel point il est efficace

```
#svm
svc = svm.SVC(kernel='linear', C=1, probability=True).fit(x_train, y_train)
prediction = svc.predict_proba(x_test)

#evaluating
print("Accuracy :",metrics.accuracy_score(y_test, prediction))
print("Confusion Matrix :",confusion_matrix(y_test, prediction))
print(classification_report(prediction,y_test))
```

Figure 22: Entraînement du modèle SVM

- **RF**

Le nombre total d'observations d'entraînement est de 350000.

Le nombre total d'observations testées est de 150000.

Le nombre d'arbres utilisé est 500.

Après avoir entraîné le modèle pendant quelques minutes, nous devons l'appliquer à l'ensemble de test pour voir à quel point il est efficace

```
#Random forest
rf = RandomForestClassifier(n_estimators=500).fit(x_train,y_train)
predictrf = rf.predict(x_test)

#evaluating
print("Accuracy :",metrics.accuracy_score(y_test, prediction))
print("Confusion Matrix :",confusion_matrix(y_test, prediction))
print(classification_report(predict,y_test))
```

Figure 23:Entraînement du modèle RF

- **RL**

Le nombre total d'observations d'entraînement est de 350000.

Le nombre total d'observations testées est de 150000.

Le paramètre de coût est 1.

Le paramètre multi_class utilisé est multinomial.

Après avoir entraîné le modèle pendant quelques minutes, nous devons l'appliquer à l'ensemble de test pour voir à quel point il est efficace

```
#Logistic regression|
lreg = LogisticRegression(solver='lbfgs', multi_class='multinomial')
lreg.fit(x_train,y_train)
prediction = lreg.predict(x_test)

#evaluating
print("Accuracy:",metrics.accuracy_score(y_test, prediction))
print("Confusion Matrix :",confusion_matrix(y_test, prediction))
print(classification_report(prediction,y_test))
```

Figure 24: Entraînement du modèle RL

- **TextBlob**

La fonction TextBlob existe dans la bibliothèque TextBlob de Python qui est dédiée pour le traitement des données de texte.

Le nombre total d'observations d'entraînement est de 350000.

Le nombre total d'observations testées est de 150000.

Une fois appliquée, on transforme les résultats quantitatifs en 3 catégories (Negative, Positive et Neutral).

```
def sentiment_calc(text):
    try:
        return TextBlob(text).sentiment
    except:
        return None

df_tb['Polarity'] = df['cleantweet'].apply(sentiment_calc).apply(lambda x: x[0])
df_tb['Subjectivity'] = df[['cleantweet']].apply(sentiment_calc).apply(lambda x: x[1])

df_tb.loc[df_tb['Polarity'] <= -0.1, 'sentiment'] = 'Negative'
df_tb.loc[df_tb['Polarity'] >= 0.1, 'sentiment'] = 'Positive'
df_tb.loc[df_tb['Polarity'].between(-0.1,0.1), 'sentiment'] = 'Neutral'
```

Figure 25: Implémentation du modèle TextBlob

- **CamemBERT**

Pour la modélisation du modèle CamemBERT, nous avons utilisé le package TensorFlow qui est une bibliothèque logicielle gratuite et open source pour l'apprentissage automatique.

Le nombre total d'observations d'entraînement est de 350000.

Le nombre total d'observations testées est de 150000.

```

import tensorflow as tf
assert tf.__version__ >= "2.0"
from transformers import CamembertTokenizer
from transformers import TFCamembertForSequenceClassification

from transformers import TFCamembertForSequenceClassification

model = TFCamembertForSequenceClassification.from_pretrained("jplu/tf-camembert-base")

opt = tf.keras.optimizers.Adam(learning_rate=5e-6, epsilon=1e-08)
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)

from sklearn import metrics
from sklearn.base import BaseEstimator, TransformerMixin
class CamembertPreprocessor(BaseEstimator, TransformerMixin):
    def __init__(self, tokenizer, max_seq_length):
        self.tokenizer = tokenizer
        self.max_seq_length = max_seq_length
    def transform(self, X, y):
        # 1. Tokenize
        X_encoded = encode_reviews(self.tokenizer, X, self.max_seq_length)
        # 2. Labels
        y_array = np.array(y)
        return X_encoded, y_array

    def fit_transform(self, X, y):
        return self.transform(X, y)

def accuracy_vs_training_data(camembert_model, initial_weights,
                              preprocessor, sizes,
                              train_reviews, train_labels,
                              val_reviews, val_labels,
                              test_reviews, test_labels):

    test_accuracies = []
    for size in sizes:
        # Preprocess data
        X_train, y_train = preprocessor.fit_transform(
            train_reviews[:size], train_labels[:size]
        )
        X_val, y_val = preprocessor.transform(val_reviews, val_labels)
        X_test, y_test = preprocessor.transform(test_reviews, test_labels)

        # Reset weights to initial value
        camembert_model.set_weights(initial_weights)
        best_model = EarlyStoppingModel(
            camembert_model, max_epochs=4, batch_size=64,
            validation_data=(X_val, y_val)
        )

        # Train model
        best_model.fit(X_train, y_train)

        # Evaluate on test set
        y_pred = best_model.predict(X_test)
        test_acc = metrics.accuracy_score(y_test, y_pred)
        test_accuracies.append(test_acc)
        print("Test acc: " + str(test_acc))

    return test_accuracies

```

Figure 26: Entraînement du modèle CamemBERT

Après l'entraînement du modèle, on passe à l'étape de calcul du score de sentiment à l'aide de la formule :

$$\frac{\exp(b)}{\exp(b) + \exp(a)}$$

Ensuite, on transforme les résultats en 3 catégories (Negative, Positive et Neutral)(voir fig 27)

```
pred['logits']

array([[ 1.1559104 , -1.1422813 ],
       [-0.45602155,  0.40601686],
       [-1.7261404 ,  1.7338504 ],
       ...,
       [ 0.1071965 , -0.10716034],
       [ 0.21242768, -0.2224468 ],
       [ 0.77679527, -0.7360965 ]], dtype=float32)

dfcol = pd.DataFrame(pred['logits'], columns = ['a', 'b'])

#mise à l'échelle -1,1
def transfrom(x):
    return ((x - 0)/(1-0)) * (1+1) - 1

#calculate score
df['score'] = np.exp(df['b']) / ( np.exp(df['b']) + np.exp(df['a']))
df['score'] = transfrom(df['score'])

df.loc[df['score'] <= -0.1, 'sentiment'] = 'Negative'
df.loc[df['score'] >= 0.1, 'sentiment'] = 'Positive'
df.loc[df['score'].between(-0.1,0.1), 'sentiment'] = 'Neutral'
```

Figure 27: Calcul du score pour la méthode de CamemBERT

II. Détection d'anomalies

La détection d'anomalies est la deuxième partie de la modélisation de ce projet.

Dans laquelle nous avons proposé deux modèles de détections d'intrusions : le modèle ARIMA et le modèle LSTM. Ces méthodes ont été testées sur 3 ensembles de données “ Paris avril 2021 ”, “ Paris mai 2021 ” et “ Orages Ile de France 2021 ”.

1. Techniques de modélisation

a) ARIMA

C'est un modèle qui est basé sur le modèle ARMA qui s'adapte aux séries chronologiques non stationnaires. (Voir annexe 3 pour plus de détails sur le modèle ARIMA).

b) LSTM

Ce modèle représente une classe de réseaux de neurones récurrents utilisé dans le domaine du DL très largement utilisé en traitement du langage naturel. (Voir annexe 3 pour plus de détails sur le modèle LSTM).

2. Mesure des performances

- **ROC**

Une courbe ROC (receiver operating characteristic) est un graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux de vrais positifs en fonction du taux de faux positifs. [58]

Le taux de vrais positifs (TVP) est l'équivalent du rappel. Il est donc défini comme suit :

$$TVP = \frac{VP}{VP + FN}$$

Le taux de faux positifs (TFP) est défini comme suit :

$$TFP = \frac{FP}{FP + VN}$$

- **AUC**

AUC signifie "aire sous la courbe ROC". Cette valeur mesure l'intégralité de l'aire à deux dimensions situées sous l'ensemble de la courbe ROC (par calculs d'intégrales) de (0,0) à (1,1).[58]

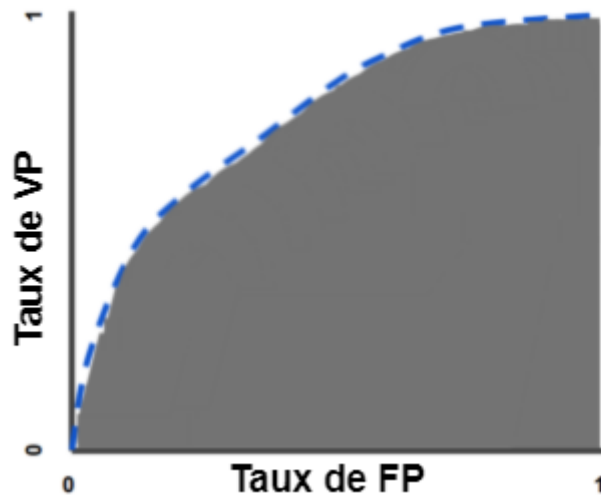


Figure 28: Exemple de courbe ROC [58]

3. Construction des modèles

a) Réglages des paramètres

- **p , d , q** : sont des entiers naturels et constituent les paramètres du modèle ARIMA.
 - p est le nombre de retards qui devraient être inclus dans le modèle autorégressif.
 - d est le nombre de fois qu'il faut différencier la série pour la rendre stationnaire.
 - q est l'ordre du modèle MA.

Dans notre projet, nous avons testé plusieurs valeurs (p,q) jusqu'à ce que nous avons obtenu les meilleurs résultats. Pour d est toujours égal à 0 pour un processus déjà stationnaire

- **time_steps** : signifie combien de valeurs existent dans une séquence.
- **layer_units**: pour déterminer le nombre de nœuds/neurones dans la couche.
- **batch_size**: La taille du lot est de 64, c'est-à-dire que pour chaque époque, un lot de 64 entrées sera utilisé pour entraîner le modèle. Cela dépend principalement de la taille de l'ensemble de données.
- **epochs**: Les époques sont le nombre de fois que le processus sera répété.
- **learning_rate**: contrôle la rapidité ou la lenteur avec laquelle un modèle de réseau neuronal apprend un problème.

b) Principe de détection

Nous avons à travers la collecte de Twitter la génération d'une courbe de fréquences.

Cette courbe peut être la fréquence globale ou une fréquence individuelle en filtrant la fréquence totale suivant un mot clé particulier.

Donc notre objectif est de faire la prédiction au fil de l'eau à chaque 10 min et d'enregistrer les résultats : anomalie ou pas anomalie.

Et dès que nous recevons une requête de l'orchestrateur à une date t^* d'intérêt, en se servant de l'antériorité sur la courbe de fréquence nous allons faire la prédiction en utilisant soit ARIMA ou LSTM.

Donc la valeur prédite est en rouge et la valeur en bleu est la valeur reçue instantanément à cette date t^* . A partir de là, nous calculons le delta entre ces deux valeurs et nous possédons un seuillage avec grand delta. Ensuite, nous prendrons les décisions.

Prenons la prédiction à t^* , le delta t^* s'il est supérieur à un certain seuil donc nous remontons une anomalie sinon pas d'anomalie. (Voir figure 29) Mais parfois nous ne pouvons pas visualiser des anomalies sur la courbe globale. Pour cette raison, ce que nous avons introduit c'est que maintenant nous rentrons dans les courbes de fréquences individuelles : nous allons disloquer le signal en plusieurs morceaux par exemple les tops 5 mots clés ensuite nous testons chacun d'entre eux pour voir sur lesquelles nous pourrions remonter des anomalies, et puis nous actualisons les résultats.

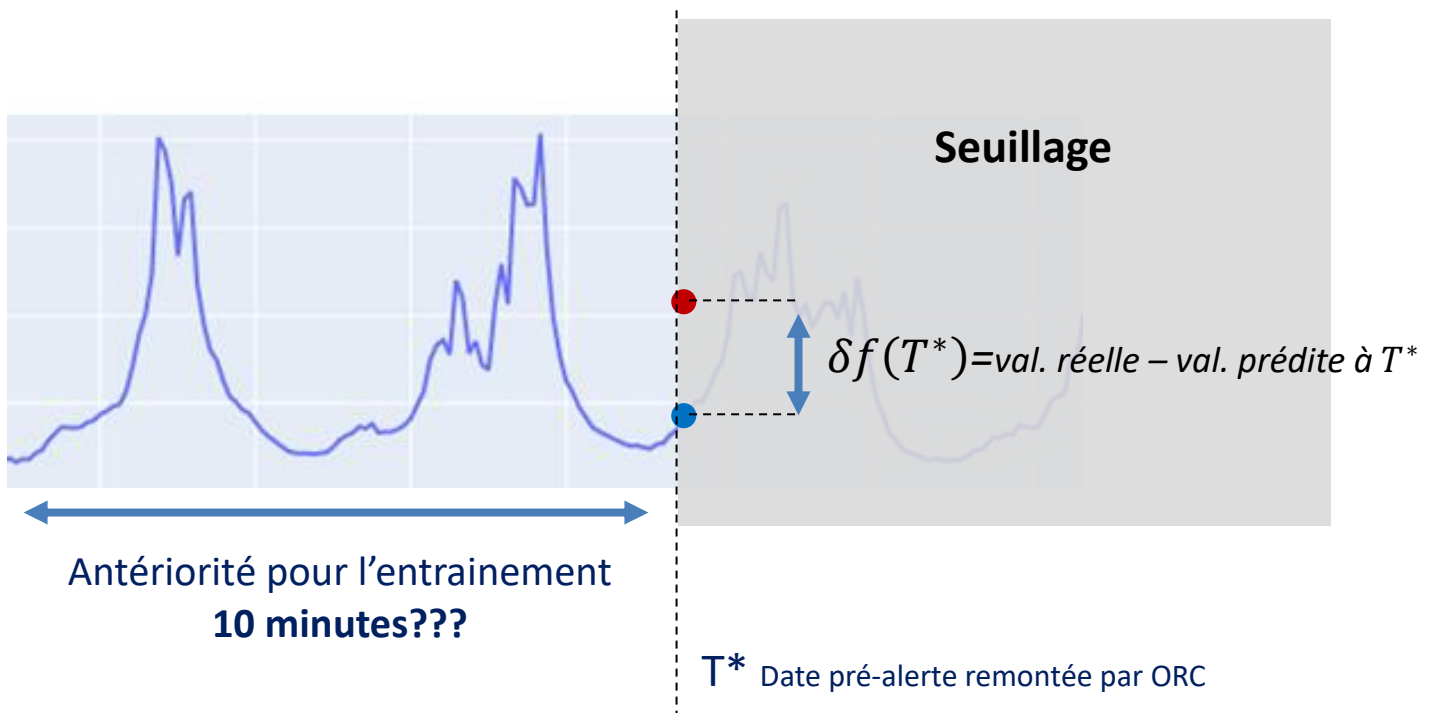


Figure 29: Principe de détection

c) Entraînement des modèles

Nous avons appliqué les deux modèles avec les données de la collecte de Paris avril-mai 2021. Celui-ci est entraîné sur une période de 30 jours du 01/04/2021 au 30/04/2021 pour faire la prédiction du mois suivant.

- **ARIMA**

Pour la modélisation du modèle ARIMA, nous avons utilisé le pacagé PyFlux qui est une bibliothèque pour l'analyse et la prévision de séries chronologiques.

Le nombre total d'observations d'entraînement est de 490000.

Le nombre total d'observations testées est de 210000.

Les paramètres ARIMA (7,0,6).

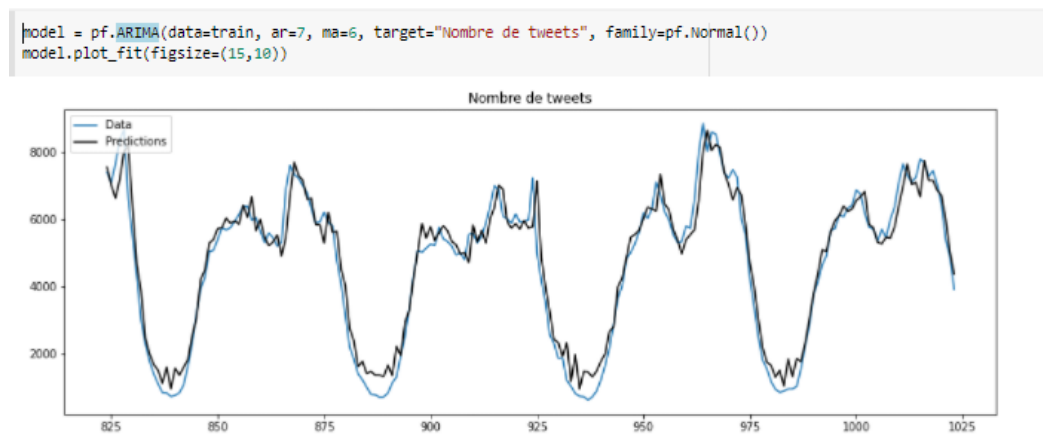


Figure 30 : Résultat de la prédiction du modèle ARIMA (signal original vs signal prédit)

Au départ, le résultat nous montre une bonne prédiction mais après au cours du temps la confiance diminue et c'est très normal. Donc nous avons utilisé une fenêtre glissante à $k+1$ (fig 31) pour améliorer les résultats et on appelle ça une prédiction de proche en proche à $t+1$.

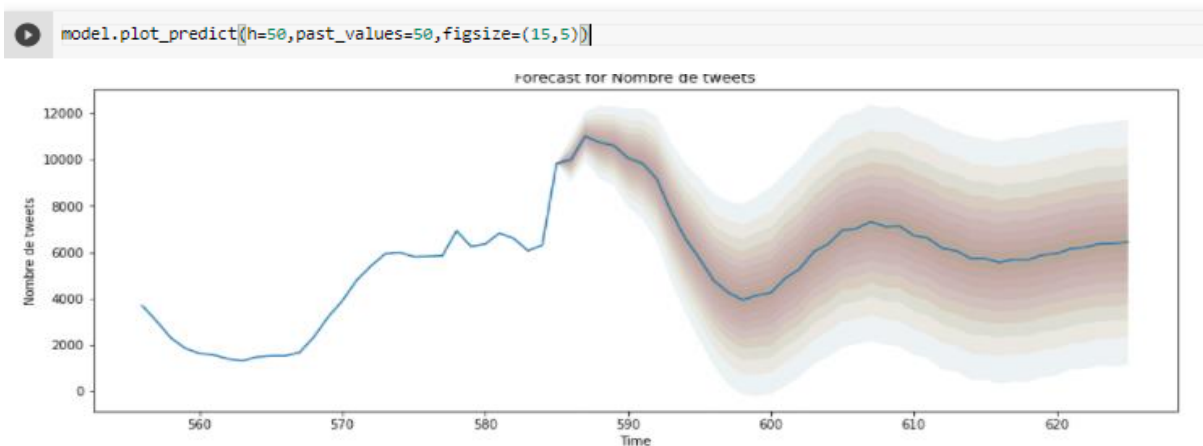
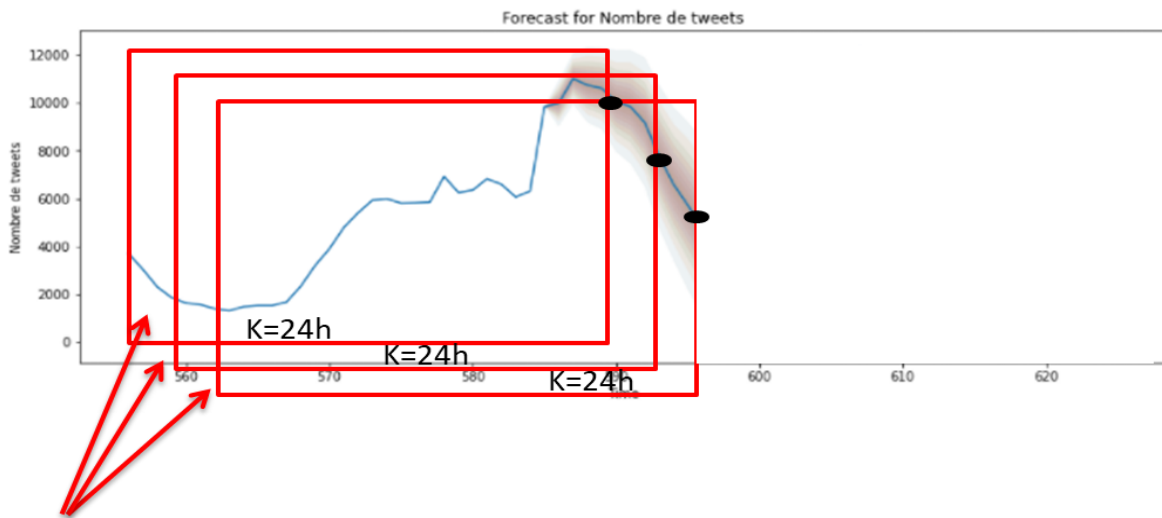


Figure 31 : Résultat de la prédiction du modèle ARIMA (intervalle de confiance)

Dans ce cas $k=24h$. Ci-dessous, le résultat de la prédiction en utilisant la méthode de la fenêtre glissante.



Fenêtre glissante avec $k=24h$

Figure 32 : Résultat de la prédiction en utilisant la fenêtre glissante

L'étape suivante c'est le calcul des écarts qui est la différence en valeur absolue entre le signal prédit et le signal original.

$$\text{Écart} = |\text{signal prédit} - \text{signal original}|$$

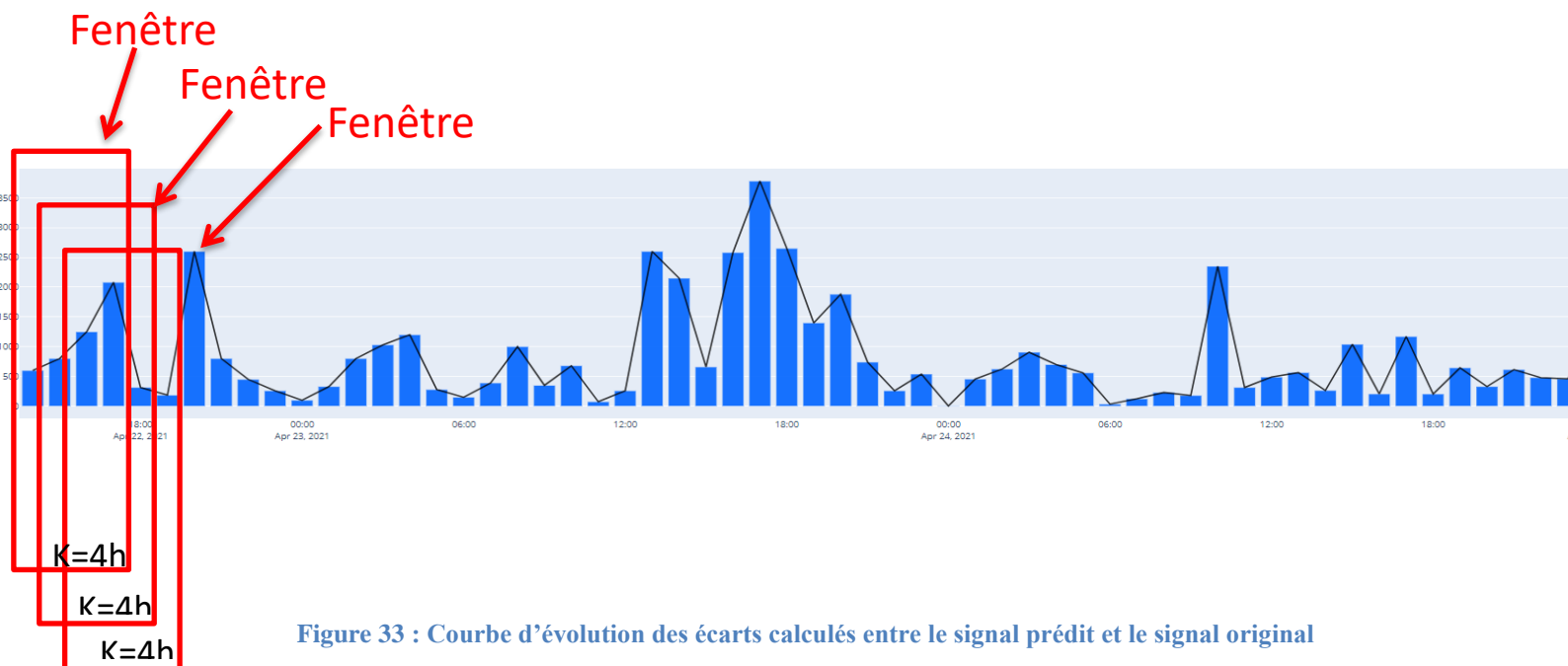


Figure 33 : Courbe d'évolution des écarts calculés entre le signal prédit et le signal original

La figure 32 présente des pics et plusieurs parties pointues tout au long de la période. Donc ici nous avons lissé la courbe en utilisant une fenêtre glissante qui crée à son tour une valeur moyenne mise à jour pour chaque k ligne(s). Sachant qu'un pas de temps est égal à 1h, le meilleur résultat était avec $K = 4$ pas = 4h. D'où nous avons obtenu le résultat suivant :

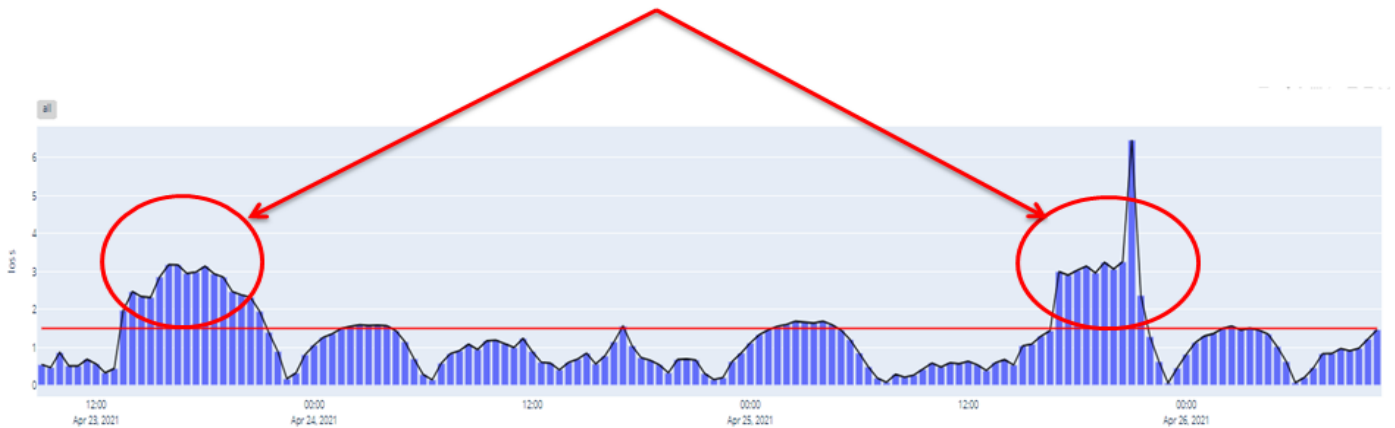


Figure 34 : Courbe d'évolution des écarts calculés entre le signal prédit et le signal original après l'utilisation de la fenêtre glissante

- LSTM

Tout d'abord, nous avons redimensionné les vecteurs train/test en des vecteurs de dimension 3 parce que le vecteur d'entrée du modèle ne peut être que dans cette forme.

```
def create_dataset(X, y, time_steps=1):
    Xs, ys = [], []
    for i in range(len(X) - time_steps):
        v = X.iloc[i:(i + time_steps)].values
        Xs.append(v)
        ys.append(y.iloc[i + time_steps])
    return np.array(Xs), np.array(ys)

#redimensionner les vecteurs train/test en des vecteurs de dimensions 3
TIME_STEPS = 10

# reshape to [samples, time_steps, n_features]

X_train1, y_train1 = create_dataset(data1, data1.freqtotale, TIME_STEPS)
X_test1, y_test1 = create_dataset(test1freqtotale, test1freqtotale.freqtotale, TIME_STEPS)
```

Figure 35: Redimensionnement des vecteurs en 3 dimensions

Ensuite, nous avons entamé les parties implémentation du modèle et entraînement.

Pour le modèle LSTM, nous avons utilisé Keras qui est le framework d'apprentissage en profondeur le plus couramment utilisé parmi les 5 meilleures équipes gagnantes de Kaggle [34].

Pour les paramètres, à chaque fois nous les faisons varier jusqu'à obtenir les meilleures performances.

```
#definir le modele
model = keras.Sequential()
model.add(keras.layers.LSTM(
    units=64,
    input_shape=(X_train1.shape[1], X_train1.shape[2])
))
model.add(keras.layers.Dropout(rate=0.2))
model.add(keras.layers.RepeatVector(n=X_train1.shape[1]))
model.add(keras.layers.LSTM(units=64, return_sequences=True))
model.add(keras.layers.Dropout(rate=0.2))
model.add(keras.layers.TimeDistributed(keras.layers.Dense(units=X_train1.shape[2])))

model.compile(loss='mae', optimizer='adam')

#entraîner le modele
history = model.fit(
    X_train1, y_train1,
    epochs=40,
    batch_size=32,
    validation_split=0.1,
    shuffle=True
)
```

Figure 36: Entraînement du modèle LSTM

Ci-dessous, le résultat de la prédiction.

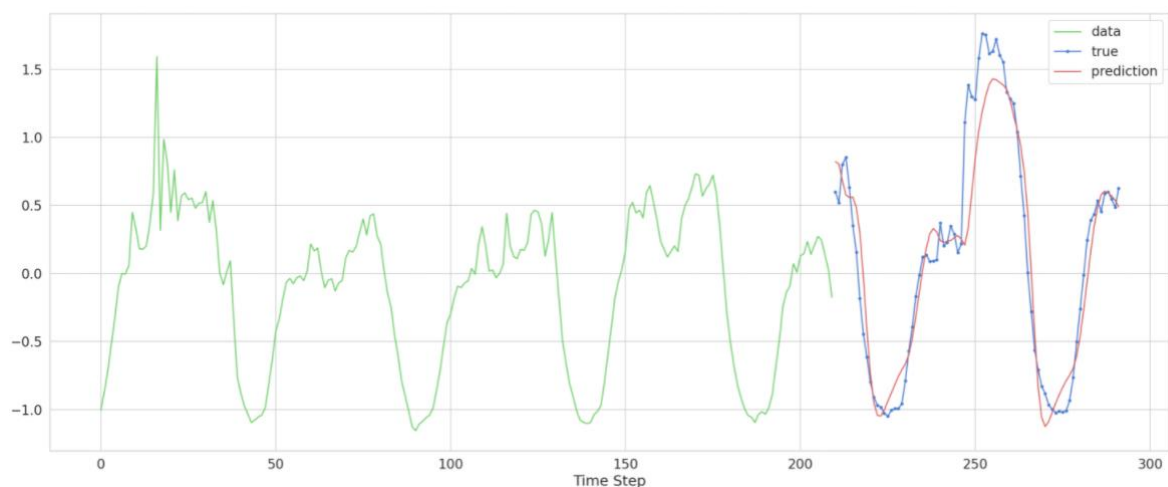


Figure 37: Résultat de la prédiction du modèle LSTM (signal original vs signal prédit)

Même chose pour LSTM, nous avons calculé les écarts entre les valeurs réelles et les valeurs prédites.

Pour le choix du seuil, il est important de décrire la méthode.

Pour cela, nous avons choisi de travailler avec la courbe ROC. Tout d'abord, nous avons calculé les faux positifs et les vrais positifs propres à chacune des valeurs de seuil possibles. Puis, à partir de ces valeurs nous avons dessiné la courbe ROC.

La meilleure valeur du seuil est celle qui maximise les vrais positifs et qui minimise les faux positifs en même temps.

Voir les figures 38-39 ci-dessous :

```
import more_itertools as mit

for x in mit.numeric_range(0, max(test_score_dffreq1['loss']), 1):

    test_score_dffreq1 = pd.DataFrame(index=test1freqtotale[TIME_STEPS:].index)
    test_score_dffreq1['loss'] = test_mae_lossfreq1
    test_score_dffreq1['threshold'] = x
    test_score_dffreq1['anomaly'] = test_score_dffreq1.loss > test_score_dffreq1.threshold

    from sklearn.metrics import confusion_matrix
    confusion_matrix = confusion_matrix(df['anomaly'][TIME_STEPS:], test_score_dffreq1['anomaly'])
    FP = confusion_matrix[0][1]
    FN = confusion_matrix[1][0]
    TP = confusion_matrix[1][1]
    TN = confusion_matrix[0][0]

    # Sensitivity, hit rate, recall, or true positive rate
    TPR = TP/(TP+FN)
    # Specificity or true negative rate
    TNR = TN/(TN+FP)
    # Fall out or false positive rate
    FPR = FP/(FP+TN)
    # False negative rate
    FNR = FN/(TP+FN)

#plot courbe roc
plt.plot(tab_auc['fpr'], tab_auc['tpr'], linewidth=5)
plt.plot([0,1],[0,1], linewidth=5)

plt.show()
```

Figure 38: Implémentation de la méthode ROC pour le choix du seuil

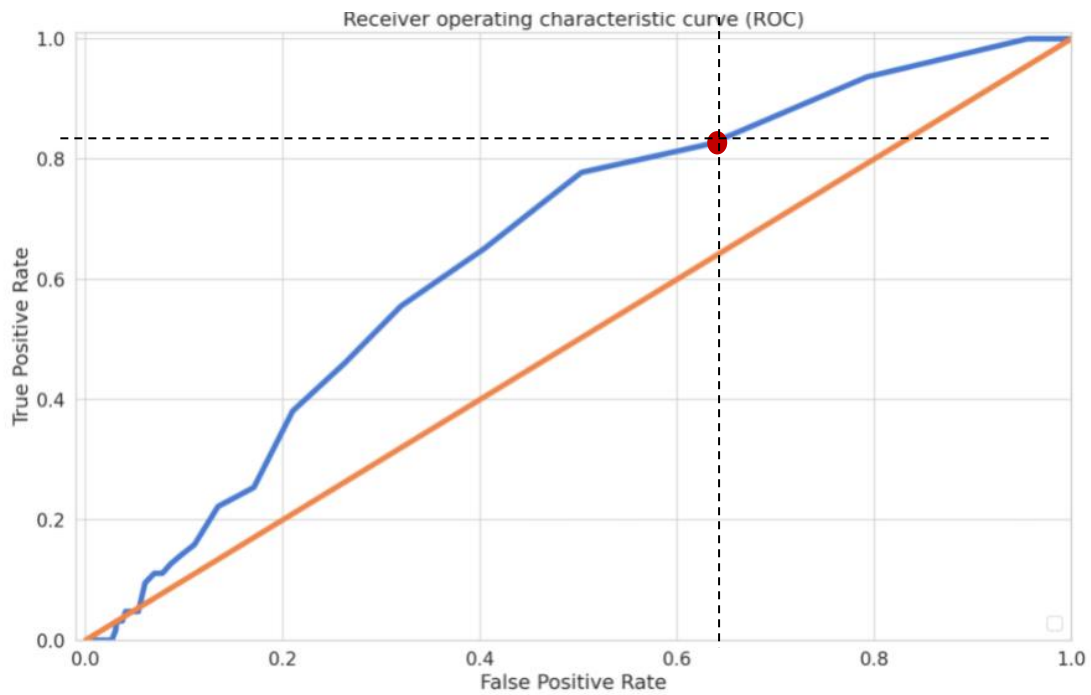


Figure 39: courbe ROC pour le choix du seuil

Conclusion

Cette phase était une transition obligatoire pour assurer un bon modèle qui nous permettrait enfin d'anticiper nos besoins du projet DISCRET. Nous passons maintenant à l'étape suivante pour en finir une solution adaptée et efficace

Chapitre 5 : Évaluation

Introduction

La modélisation se fait en différentes itérations, il est temps de voir à quoi ressemblaient les différents résultats et quel est le résultat de notre analyse. Pour cela, nous utiliserons les outils de test dont nous avons parlé plus tôt.

I. Évaluation des résultats

1. Classification et analyse de sentiments

a. SVM

○ Itération 1 : Cost = 0.01

	BOW			TF-IDF			Word2Vec		
	Prédite 0	Prédite 1	Prédite -1	Prédite 0	Prédite 1	Prédite -1	Prédite 0	Prédite 1	Prédite -1
Actuelle 0	31254	20458	16740	22664	3421	14173	36428	1433	2397
Actuelle 1	18632	25468	11705	13854	35649	15871	27451	20369	17554
Actuelle -1	2017	439	23287	5418	19263	19687	12445	16470	15453

BOW

- o Accuracy: 53%
- o Misclassification Rate: 46%

TF-IDF

- o Accuracy: 52%
- o Misclassification Rate: 47%

Word2Vec

- o Accuracy: 48%
- o Misclassification Rate: 51%

○ Itération 2 : Cost=1

	BOW			TF-IDF			Word2Vec		
	Prédite 0	Prédite 1	Prédite -1	Prédite 0	Prédite 1	Prédite -1	Prédite 0	Prédite 1	Prédite -1
Actuelle 0	19672	14086	6500	18801	5666	15791	19773	7441	13044
Actuelle 1	3699	49591	12084	3265	54295	7814	5214	48221	11939
Actuelle -1	5469	6219	32677	5899	10733	27644	2978	8536	32851

BOW

- o Accuracy: 67.9%
- o Misclassification Rate: 32.1%

TF-IDF

- o Accuracy: 67.16%
- o Misclassification Rate: 32.84%

Word2Vec

- o Accuracy: 67.23%
- o Misclassification Rate: 32.77%

b. RF

	BOW			TF-IDF			Word2Vec		
	Prédite 0	Prédite 1	Prédite -1	Prédite 0	Prédite 1	Prédite -1	Prédite 0	Prédite 1	Prédite -1
Actuelle 0	26471	5200	8587	33224	548	6486	22614	9772	7872
Actuelle 1	866	52431	12077	1228	53666	10480	1293	48955	15126
Actuelle -1	2546	14056	27763	12865	14200	17300	5249	10677	28439

BOW

- o Accuracy: 71.11%
- o Misclassification Rate: 28.89%

TF-IDF

- o Accuracy: 69.46%
- o Misclassification Rate: 30.54%

Word2Vec

- o Accuracy: 66.67%
- o Misclassification Rate: 33.33%

c. RL

	BOW			TF-IDF			Word2Vec		
	Prédite 0	Prédite 1	Prédite -1	Prédite 0	Prédite 1	Prédite -1	Prédite 0	Prédite 1	Prédite -1
Actuelle 0	20105	7548	12605	29648	2655	7955	27909	3201	9148
Actuelle 1	5222	49223	10929	6393	55117	3864	14556	42134	8684
Actuelle -1	699	3419	40247	6594	13726	24045	7199	6889	30277

BOW

- o Accuracy: 73.05%
- o Misclassification Rate: 26.95%

TF-IDF

- o Accuracy: 72.54%
- o Misclassification Rate: 27.46%

Word2Vec

- o Accuracy: 66.88%
- o Misclassification Rate: 33.12%

d. SVM vs RF vs RL

Représentation	SVM	RL	RF
BOW	67.96	73.05	71.11
TF-IDF	67.16	72.54	69.46
Word2Vec	67.23	66.88	66.67

Figure 40 : Les performances des différents classifieurs

Représentation	SVM	RL	RF
BOW	p=65.45, r=67.27, F0=66.35	p=72.82, r=73.94, F0=72.35	p=68.43, r=69.69, F0=68.00
TF-IDF	p=65.41, r=69.28, F0=67.29	p=71.49, r=73.90, F0=72.67	p=64.45, r=70.68, F0=67.38
Word2Vec	p=69.91, r=70.49, F0=69.87	p=65.53, r=69.16, F0=67.29	p=65.12, r=64.82, F0=66.03

Figure 41 : Mesures de performance

e. TextBlob

	BOW		
	Prédite 0	Prédite 1	Prédite -1
Actuelle 0	35565	2058	2635
Actuelle 1	16686	46322	2366
Actuelle -1	8972	283	35113

o Accuracy: 78.45%

o Taux de classification erronée: 21.55%

f. CamemBERT

	BOW		
	Prédite 0	Prédite 1	Prédite -1
Actuelle 0	32620	1127	6511
Actuelle 1	759	59634	4981
Actuelle -1	2100	1031	41237

o Accuracy: 89.61%

o Taux de classification erronée: 10.39%

Maintenant, nous considérons les ratios de sentiment au lieu du nombre de sentiment prise en compte du temps. Nous obtenons donc une courbe de l'évolution des ratios de sentiment dans le temps. (Voir figure 42)

Nous avons constaté que les deux courbes de sentiments positif et négatif semblent balancer à tout moment peu importe ce qui se passe.

Il y a aussi deux niveaux de pourcentage de sentiment : la courbe en noir qui est tout le temps entre 60% et 50% et même la courbe en rouge oscille entre 40% et 45%.

De plus, nous avons remarqué qu'il y a des gros tubes qui se succèdent et tout d'un coup autour du 25 avril des petits tubes qui se succèdent et se rapprochent.

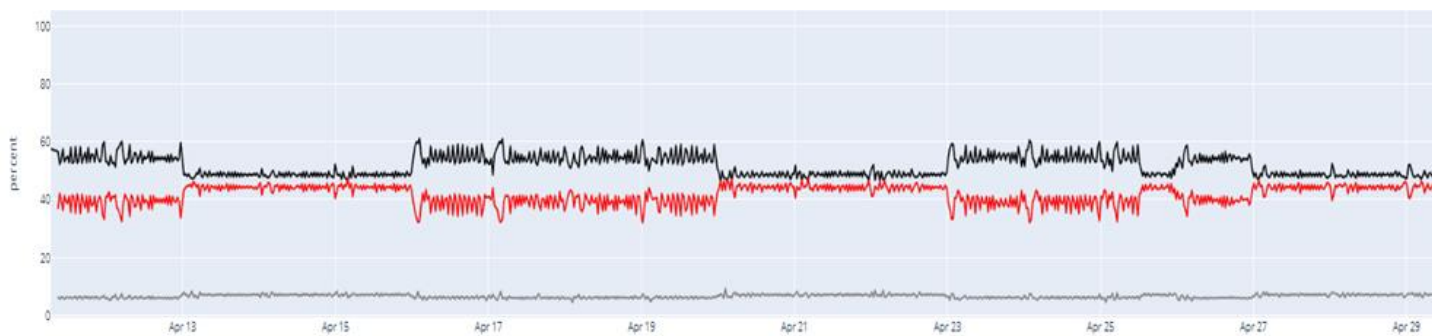


Figure 42 : Courbe de l'évolution des ratios de sentiment dans le temps

2. Détection d'anomalies

a. ARIMA vs LSTM

La courbe ROC ci-dessous montre que notre courbe LSTM est plus proche de la bordure gauche du graphique (plus précis). De l'autre côté, la zone sous la courbe pour ARIMA est plus grande et cela aussi décrit la précision. Ce que nous pouvons dire, c'est que les deux algorithmes ont échangé des faux positifs contre des vrais positifs et vrais positifs aux faux positifs.

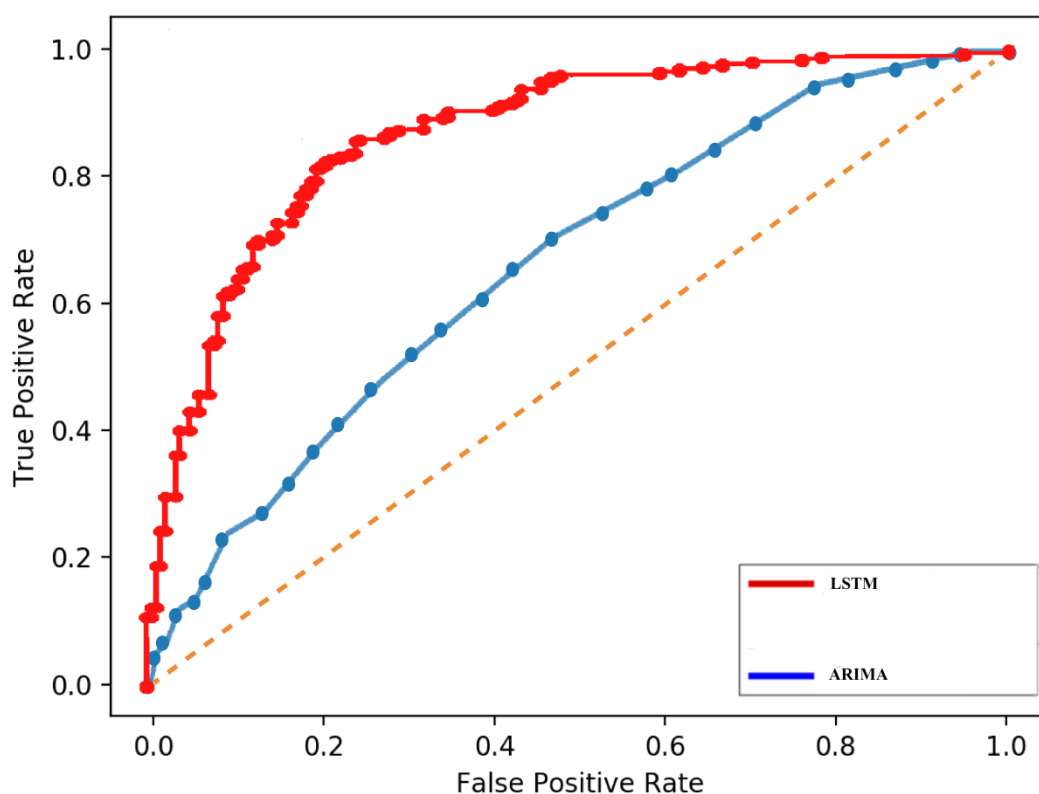


Figure 43: Courbe ROC ARIMA vs LSTM

o AUC ARIMA : 59.61%

o AUC LSTM : 81.48 %

II. Modèle approuvé

En raison des résultats que nous voyons ci-dessus, la partie restante est de choisir. Dans la partie analyse de sentiments, les modèles de classification SVM, RL et RF sont à nier complètement puisque Textblob et BERT ont donné les meilleurs pourcentages de précision. Nous avons opté pour le modèle BERT qui a bien catégorisé les sentiments (positif, négatif et neutre) avec une précision qui est égale à 89 % mais ça n'empêche que nous ne pouvons pas tirer une conclusion à partir des ratios de sentiments (figure 42). Nous avons remarqué qu'il y a plusieurs pistes qui émergent et pour le moment nous n'avons pas une qui nous sert de quelque chose car cette façon de faire ne nous aide pas à déterminer qu'il y a eu une anomalie.

Dans la partie détection d'anomalies, lorsque on compare ARIMA et LSTM choisir l'un d'eux signifie automatiquement choisir un bon modèle pour commencer. Donc nous avons opté pour le modèle LSTM et ce pour ces points :

- 81% de précision
- Moins de faux positifs que ARIMA : Il est très important pour nous de ne pas avoir de faux positifs.

Conclusion

Cette étape est très constructive, elle nous permet de revoir l'ensemble du projet pour vérifier s'il répond aux normes que nous nous sommes fixées au départ, surtout s'il y a des défauts à répondre aux exigences du client. Maintenant que nous sommes sûrs que chaque aspect de notre produit a été résolu, nous pouvons entrer en toute confiance dans la phase suivante, la phase de déploiement.

Chapitre 6 : Déploiement

Introduction

C'est la dernière étape du processus. Il s'agit de mettre le modèle résultant en production pour l'utilisateur final. L'objectif est de mettre les connaissances acquises grâce à la modélisation dans le processus de prise de décision sous une forme appropriée.

Par conséquent, selon l'objectif, le déploiement peut aller de la simple génération d'un rapport décrivant les connaissances acquises à la mise en œuvre d'une application, permettant l'utilisation du modèle acquis pour prédire la valeur inconnue d'intérêt de l'élément.

I. Visualisation

Nous avons utilisé PowerBI comme outil pour assurer la phase de déploiement.

Nous avons déployé notre solution développée et fourni des interfaces pour faire bouger les choses.

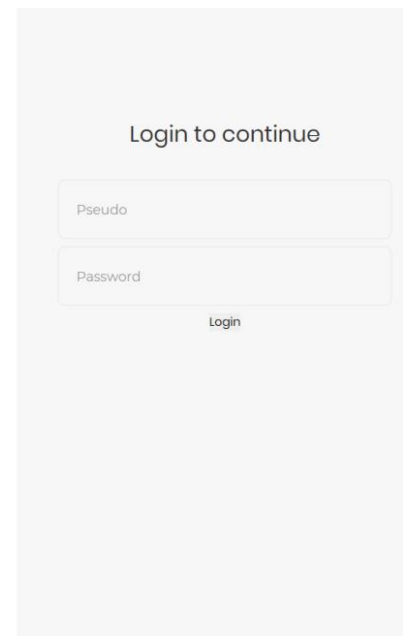


Figure 44: Interface de connexion

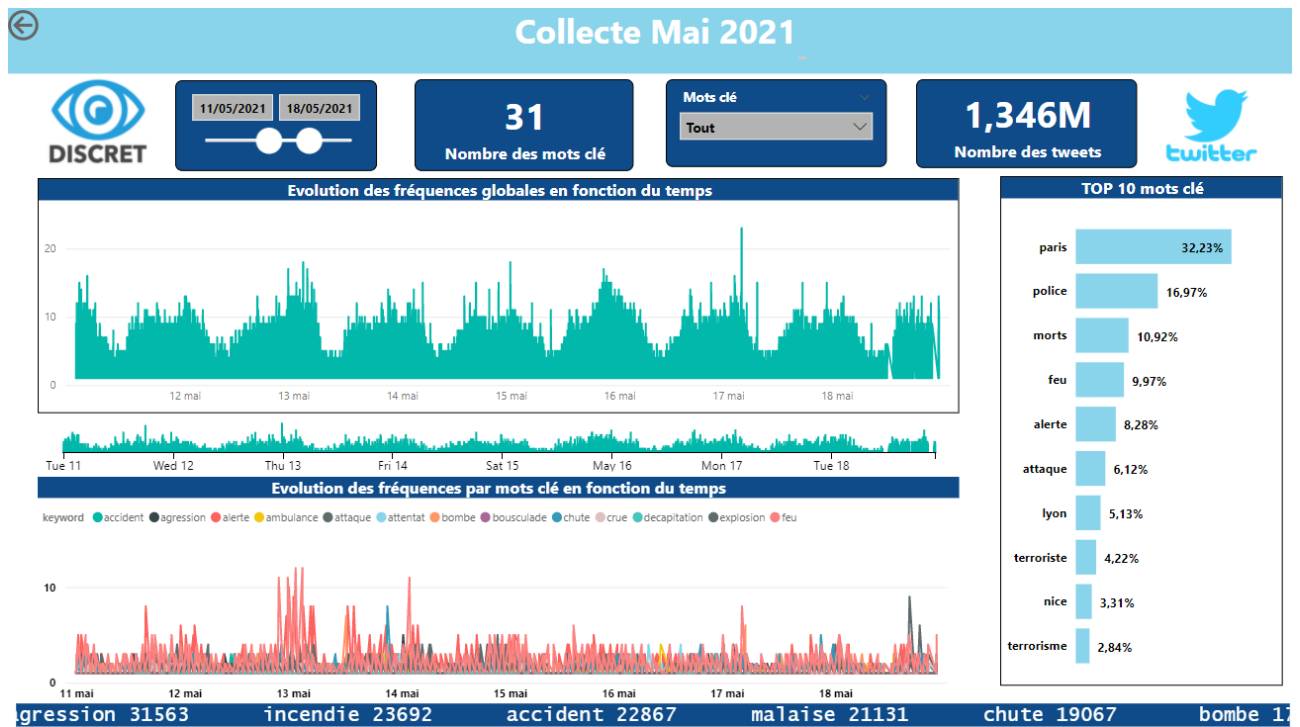


Figure 45: Visualisation détaillée sur la collecte de Mai

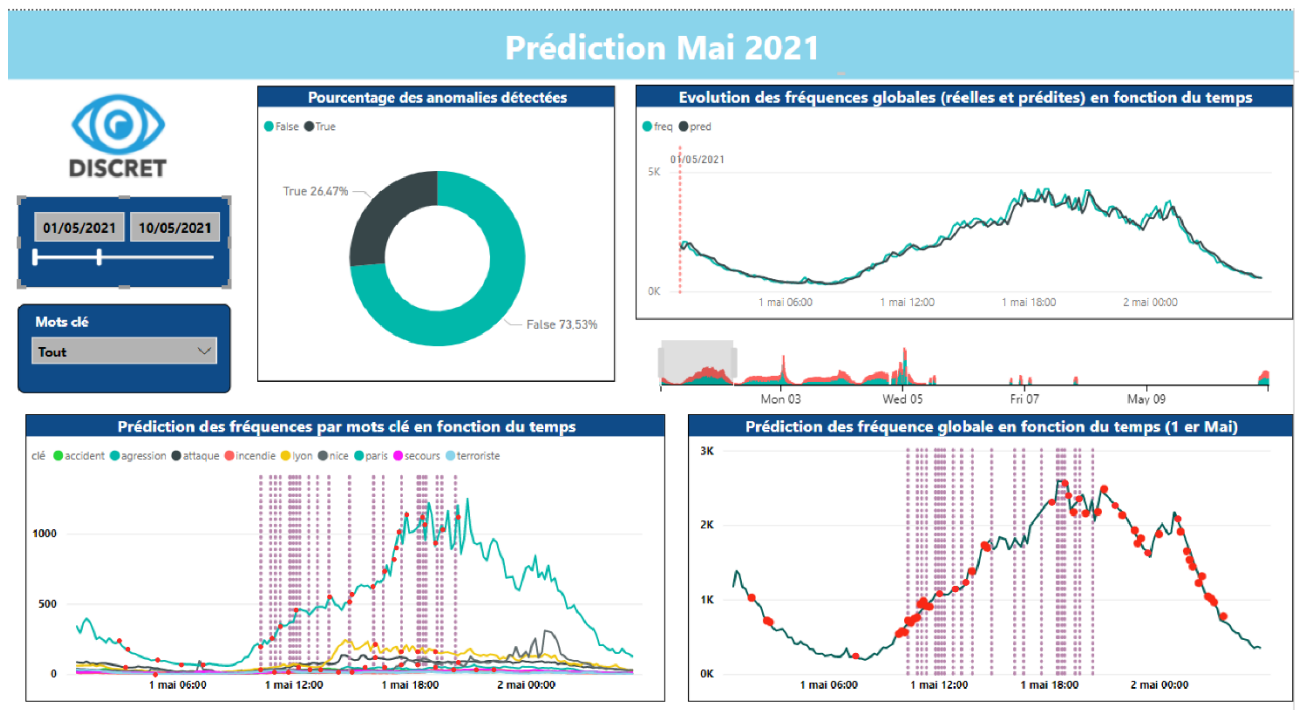


Figure 46: Visualisation des anomalies détectées pendant le mois de Mai en utilisant le filtre date

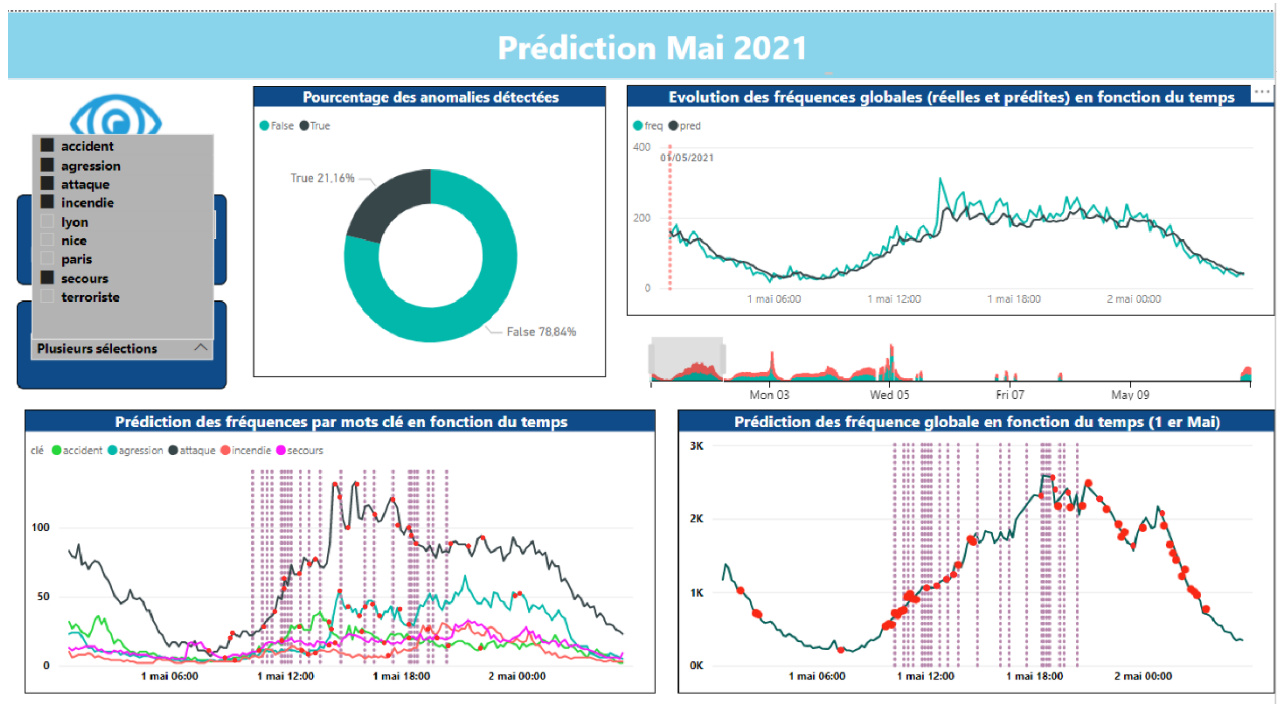


Figure 47 : Visualisation des anomalies détectées pendant le mois de Mai en utilisant le filtre mots clés

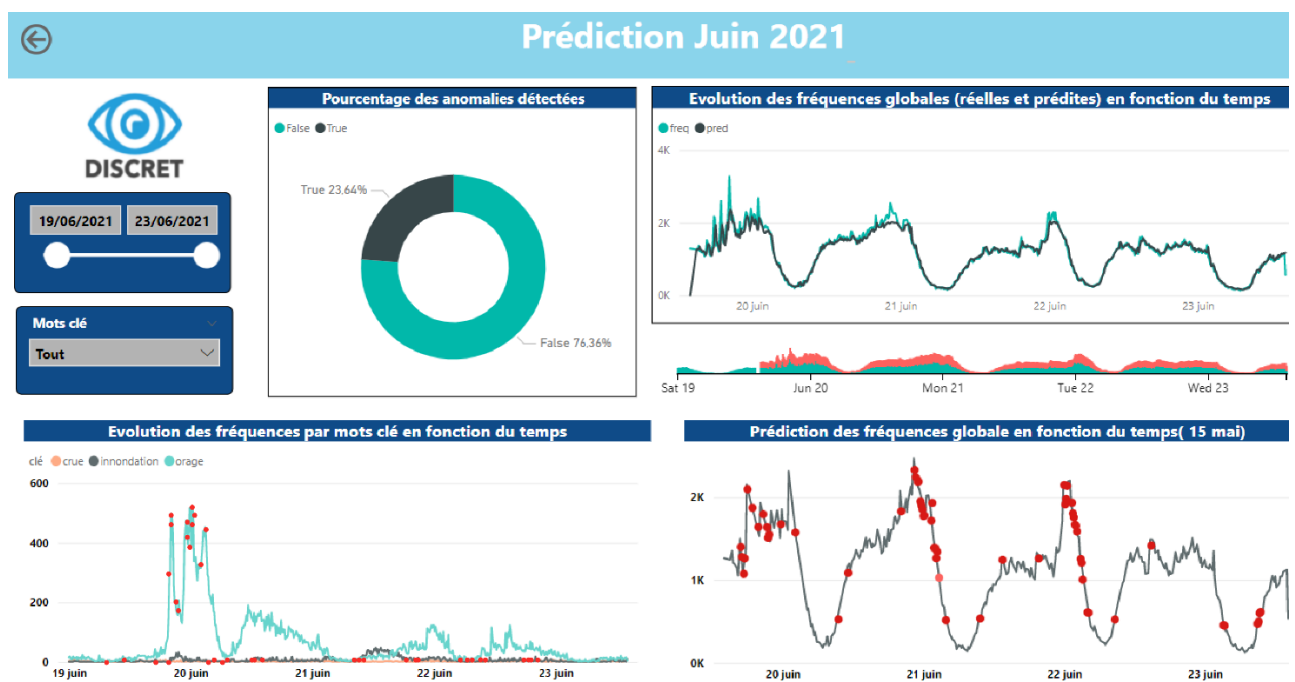


Figure 48 : Visualisation des anomalies détectées pendant le mois de Juin avec le modèle LSTM

Conclusion

Nous avons finalement terminé notre projet avec cette dernière étape de notre méthodologie.

Tout était inclus dans des interfaces. Nous avons également réussi à tout visualiser avec l'outil de reporting PowerBI.

Conclusion et perspectives

Les réseaux sociaux sont l'un des domaines les plus intéressants sur lesquels on puisse travailler aujourd'hui. Cela est dû au fait que l'analyste travaille pour comprendre le comportement de l'utilisateur.

Ce projet a été un vrai challenge pour nous. Dans ce travail, notre intérêt s'est porté sur l'analyse des sentiments avec des techniques d'apprentissage automatique et le traitement du langage naturel dont l'objectif principal est de réaliser un système de détection d'anomalies.

L'ensemble de données considéré s'appuie sur les opinions du réseau social Twitter. Un prétraitement de données a été effectué avant une visualisation et une analyse des données.

Nous voulons essayer plus de méthodes dans le Deep Learning et les ANN qui auraient pu être plus précis. Cependant, en raison de la contrainte de temps et de la complexité de la méthode elle-même, cette partie n'a pas été atteinte. Par contre, ce travail n'est pas encore achevé : une équipe complète le traitera dans un court laps de temps en évaluant et en optimisant les algorithmes mis en œuvre à partir de données plus volumineuses en temps réel.

Ce fut une expérience très intéressante et précieuse au cours de laquelle j'ai étudié et acquis des compétences et des connaissances techniques.

En raison du coronavirus et du confinement, le début du stage était un peu compliqué, mais sous la supervision de mon maître de stage, de l'équipe et de moi-même, nous avons réalisé le stage avec succès.

Enfin nous pouvons conclure que ce projet a été bénéfique à plusieurs niveaux notamment l'exploration des domaines de la régression et de la classification de textes, de l'apprentissage automatique, de l'apprentissage profond, de la visualisation de données et surtout le travail d'équipe. À l'avenir, tout cela sera dû au bon environnement dans lequel j'ai l'opportunité de travailler.

Webographie

- [1] <https://www.cfa-afia.com/etablissements/utt> (Dernier accès : Avril 2021)
- [2] <https://www.utt.fr/l-utt-en-bref/presentation-de-l-utt-et-ses-missions>
(Dernier accès : Avril 2021)
- [3] <https://www.utt.fr/english-version/la-recherche-de-lutt/la-recherche-de-lutt>
(Dernier accès : Avril 2021)
- [4] <https://www.utt.fr/recherche-utt> (Dernier accès : Avril 2021)
- [5] <https://etu.utt.fr/assets/img/home.jpg?cachebuster-prod>
- [6] <https://recherche.utt.fr/capsec> (Dernier accès : Avril 2021)
- [7] https://entreprises.utt.fr/medias/photo/batiment-recherche-4-1539869490339-jpg?ID_FICHE=1481 (Dernier accès : Avril 2021)
- [8] <https://www.lebigdata.fr/github-tout-savoir> (Dernier accès : Avril 2021)
- [10] https://fr.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining (Dernier accès : Avril 2021)
- [11] <https://medium.datadriveninvestor.com/data-science-project-management-methodologies-f6913c6b29eb>
- [12] <https://datascientest.com/les-methodes-propres-aux-projets-big-data> (Dernier accès : Avril 2021)
- [13] <https://ledatascientist.com/google-colab-le-guide-ultime/> (Dernier accès : Avril 2021)
- [14] <https://info.blaisepascal.fr/langages/python> (Dernier accès : Avril 2021)
- [15] <https://www.lebigdata.fr/power-bi-microsoft> (Dernier accès : Avril 2021)
- [16] <https://www.synaltic.fr/wp-content/uploads/2020/11/Logo-Power-BI-Benchmark.jpg>
- [17] <https://www.flexsi.fr/2020/04/24/microsoft-teams/> (Dernier accès : Avril 2021)
- [18] https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Microsoft_Office_Teams_%282018%E2%80%932019%29.svg/1200px-Microsoft_Office_Teams_%282018%E2%80%932019%29.svg.png (Dernier accès : Avril 2021)
- [19] [https://fr.wikipedia.org/wiki/Slack_\(plateforme\)](https://fr.wikipedia.org/wiki/Slack_(plateforme)) (Dernier accès : Avril 2021)
- [20] <https://fr.quora.com/Quest-ce-que-le-NLP-Natural-Language-Processing> (Dernier accès : Avril 2021)
- [21] <https://www.proxem.com/le-lexique-du-traitement-automatique-des-langues> (Dernier accès : Avril 2021)
- [22] <https://www.lebigdata.fr/traitement-naturel-du-langage-nlp-definition> (Dernier accès : Avril 2021)
- [23] <https://ichi.pro/fr/tokenisation-pour-le-traitement-du-langage-naturel-177543891237588> (Dernier accès : Avril 2021)
- [24] <https://www.actuia.com/contribution/victorbigand/tutoriel-tal-pour-les-debutants-classification-de-texte/> (Dernier accès : Avril 2021)
- [25] <https://openclassrooms.com/fr/courses/4470541-analysez-vos-donnees-textuelles/4854971-nettoyez-et-normalisez-les-donnees> (Dernier accès : Avril 2021)
- [26] <https://www.24pm.com/117-definitions/261-modele-du-sac-de-mots> (Dernier accès : Avril 2021)
- [27] <https://www.quentinfily.fr/tf-idf-pertinence-lexicale/> (Dernier accès : Avril 2021)
- [28] https://ethen8181.github.io/machine-learning/clustering_old/tf_idf/tf_idf.html (Dernier accès : Avril 2021)
- [29] <https://fr.ert.wiki/wiki/Word2vec> (Dernier accès : Avril 2021)
- [30] <https://datascientest.com/machine-learning-tout-savoir> (Dernier accès : Avril 2021)
- [31] https://fr.wikipedia.org/wiki/Apprentissage_supervise%C3%A9 (Dernier accès : Avril 2021)
- [32] <https://actualiteinformatique.fr/intelligence-artificielle/lapprentissage-supervise-ai> (Dernier accès : Avril 2021)
- [33] <https://moncoachdata.com/blog/modeles-de-machine-learning-expliques/> (Dernier accès : Mai 2021)
- [34] <https://keras.io/> (Dernier accès : Juillet 2021)
- [35] https://datafranca.org/wiki/Analyse_des_sentiments (Dernier accès : Mai 2021)
- [36] <https://opensofty.com/fr/2020/1/20/analyse-des-sentiments-avec-textblob-et-python/> (Dernier accès : Mai 2021)

- [37] <https://www.hebergementwebs.com/tutoriel-sur-lintelligence-artificielle/intelligence-artificielle-reseaux-de-neurones> (Dernier accès : Mai 2021)
- [38] https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_r%C3%A9currents (Dernier accès : Mai 2021)
- [39] <https://dataanalyticspost.com/Lexique/reseaux-de-neurones-recurrents/> (Dernier accès : Mai 2021)
- [40] <https://www.franceculture.fr/emissions/le-journal-des-sciences/le-journal-des-sciences-du-mardi-19-novembre-2019> (Dernier accès : Mai 2021)
- [41] <https://medium.com/@vitalshchutski/french-nlp-entamez-le-camembert-avec-les-librairies-fast-bert-et-transformers-14e65f84c148> (Dernier accès : Mai 2021)
- [42] <https://dataanalyticspost.com/Lexique/auto-encodeur/> (Dernier accès : Mai 2021)
- [43] <https://openclassrooms.com/fr/courses/5801891-initiez-vous-au-deep-learning/5814621-initiez-vous-aux-autoencodeurs> (Dernier accès : Mai 2021)
- [44] <https://www.actuia.com/contribution/cedric-vasseur/bert-et-sa-version-francaise-camembert/> (Dernier accès : Mai 2021)
- [45] https://fr.wikipedia.org/wiki/D%C3%A9tection_d%27anomalies (Dernier accès : Mai 2021)
- [46] https://fr.wikipedia.org/wiki/S%C3%A9rie_temporelle (Dernier accès : Mai 2021)
- [47] <https://ledatascientist.com/arima/> (Dernier accès : Mai 2021)
- [48] <https://openclassrooms.com/fr/courses/5801891-initiez-vous-au-deep-learning/5814656-decouvrez-les-cellules-a-memoire-interne-les-lstm> (Dernier accès : Mai 2021)
- [49] <https://www.aclweb.org/anthology/N19-1423/> (Dernier accès : Mai 2021)
- [50] https://fr.wikipedia.org/wiki/Matrice_de_confusion (Dernier accès : Mai 2021)
- [51] <https://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-modele-de-machine-learning/4308256-evaluez-un-algorithme-de-classification-qui-retourne-des-valeurs-binaires> (Dernier accès : Mai 2021)
- [52] <https://developers.google.com/machine-learning/crash-course/classification/accuracy?hl=fr> (Dernier accès : Mai 2021)
- [53] https://perso.liris.cnrs.fr/marc.plantevit/ENS/M2_FD.pdf (Dernier accès : Juin 2021)
- [54] <https://fr.wikipedia.org/wiki/SEMMA> (Dernier accès : Juin 2021)
- [55] <https://fr.blog.businessdecision.com/methode-crisp-la-cle-de-la-reussite-en-data-science/> (Dernier accès : Juin 2021)
- [56] <https://larmarange.github.io/analyse-R/regression-logistique.html> (Dernier accès : Juillet 2021)
- [57] https://ichi.pro/assets/images/max/724/1*ogwigF24YOPXZHsedVzCaA.jpeg
- [58] <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=fr> (Dernier accès : Juillet 2021)

Annexe 0

I. Méthodologies

1. KDD

Regardons de plus près chaque étape. Chaque étape consiste en un ensemble d'actions prédéterminées qui sont effectuées :

1-Sélection : Créez un ensemble de données cible ou un sous-ensemble de variables ou d'échantillons de données qui nécessitent une exploration plus approfondie.

2-Prétraitement : Prétraitement des données cibles pour obtenir des données cohérentes.

3-Transformation : Transformation de données à l'aide de méthodes de réduction ou de transformation de dimensions.

4-Data-mining : Trouver des modèles intéressants dans une représentation spécifique qui dépend de l'objectif de l'exploration de données (par exemple : la prédiction).

5-Interprétation / Évaluation : Interprétation et évaluation du modèle. [11]

Une fois le cycle terminé et toutes les étapes terminées, l'expert recevra des informations d'évaluation indiquant si les connaissances ont bien été acquises. Sinon, le cycle se répète à partir de n'importe quelle étape où la cible de mise à jour est utilisée, jusqu'à ce que la cible soit atteinte. [11]

2. SEMMA

Regardons de plus près chaque étape.

1-Echantillonnage : Prendre une partie d'un grand ensemble de données, assez grand pour extraire des informations importantes, assez petit pour fonctionner rapidement.

2-Exploration : L'exploration des données peut aider à obtenir des informations et des idées et à améliorer le processus de découverte en recherchant des tendances et des anomalies.

3-Modification : L'étape de modification des données se concentre sur la création, la sélection et la transformation de variables pour se concentrer sur le processus de sélection du modèle. Cette étape peut également rechercher des valeurs aberrantes et réduire le nombre de variables.

4-Modélisation : Il existe différentes techniques de modélisation, et chaque type de modèle a ses avantages et convient aux objectifs spécifiques de l'exploration de données.

5-Accès : L'objectif de cette dernière étape est d'évaluer la fiabilité et l'utilité des résultats, et d'estimer les performances. [11]

De la même manière que dans KDD, SEMMA se répétera jusqu'à atteindre l'objectif fixé. [11]

3. CRISP-DM

La méthode CRISP se décompose en 6 étapes :

1. La compréhension du problème métier : La première étape consiste à bien comprendre les éléments métiers et problématiques que la Data Science vise à résoudre ou à améliorer. [55]

2. La compréhension des données : Cette phase vise à déterminer précisément les données à analyser, à identifier la qualité des données disponibles et à faire le lien entre les données et leur signification d'un point de vue métier. [55]

3. La préparation des données : Cette phase de préparation des données regroupe les activités liées à la construction de l'ensemble précis des données à analyser, faite à partir des données brutes. Elle inclut ainsi le classement des données en fonction de critères choisis, le nettoyage des données, et surtout leur recodage pour les rendre compatibles avec les algorithmes qui seront utilisés. [55]

4. La modélisation : C'est la phase de Data Science proprement dite. La modélisation comprend le choix, le paramétrage et le test de différents algorithmes ainsi que leur enchaînement, qui constitue un modèle. Ce processus est d'abord descriptif pour générer de la connaissance, en expliquant pourquoi les choses se sont passées. Il devient ensuite prédictif en expliquant ce qu'il va se passer, puis prescriptif en permettant d'optimiser une situation future. [55]

5. L'évaluation : L'évaluation vise à vérifier le(s) modèle(s) ou les connaissances obtenues afin de s'assurer qu'ils répondent aux objectifs formulés au début du processus. Elle contribue aussi à la décision de déploiement du modèle ou, si besoin est, à son amélioration. [55]

6. Le déploiement : Il consiste en une mise en production pour les utilisateurs finaux des modèles obtenus. Il peut ainsi aller de la simple génération d'un rapport décrivant les connaissances obtenues jusqu'à la mise en place d'une application, permettant l'utilisation du modèle obtenu, pour la prédiction de valeurs inconnues d'un élément d'intérêt. [55]

Dans le tableau 1, nous résumons les correspondances présentées.

Tableau 2:Tableau comparatif entre les méthodes KDD , SEMMA et CRISP-DM [11]

KDD	SEMMA	CRISP-DM
---	---	Business Understanding
Selection	Sample	Data Understanding
Preprocessing	Explore	
Transformation	Modify	Data Preparation
Data Mining	Model	Modeling
Interpretation/Evaluation	Assess	Evaluation
---	---	Deployment

Annexe 1

I. Natural language processing

Natural Language Processing (NLP) est une branche très importante du Machine Learning et donc de l'intelligence artificielle. Le NLP est la capacité d'un programme à comprendre le langage humain. [20] Il s'agit donc d'une discipline informatique à part entière qui recouvre de nombreux sujets et méthodes, qui sont à l'origine notamment des moteurs de recherche.

La plupart des techniques de traitement naturel du langage reposent sur le Deep Learning ou apprentissage profond. Les algorithmes d'intelligence artificielle sont entraînés à partir de données afin d'apprendre à analyser le langage humain pour y trouver des patterns et des corrélations.[21]

Les algorithmes ont pour rôle d'identifier et d'extraire les règles du langage naturel, afin de convertir les données de langage non structuré sous une forme que les ordinateurs pourront comprendre. [22]

II. Les techniques de prétraitement

Le traitement de normalisation de texte repose principalement sur quatre tâches de base : tokenisation, normalisation de mots, suppression des stopwords et lemmatisation et stemming. Ces tâches seront détaillées dans les sections suivantes.

1. La tokenisation

La tokenisation décompose le texte brut en mots, des phrases appelées jetons. Ces jetons aident à comprendre le contexte ou à développer le modèle de la NLP. La tokenisation aide à expliquer la signification du texte en analysant la séquence des mots.[23]

Signes de ponctuation, chiffres, liens Web et caractères spéciaux... sont généralement retirés, mais peut être mis de côté pour le l'élimination. Cela dépend en grande partie du contexte et de la philosophie de l'algorithme.

2. La normalisation de mots

Normaliser le texte signifie le mettre à la même casse [24], généralement toutes les lettres en minuscule pour avoir les mêmes mots au début et dans toute autre partie des phrases soient représentées de la même manière.

La normalisation de mots implique également le regroupement des mots / jetons avec différentes formes dans un format standardisé, ce qui peut effectivement

éliminer les traces d'erreurs d'orthographe, mais peut être utile pour obtenir des quantités standardisées et un petit nombre de jetons.

3. La lemmatisation et le stemming

Ces deux méthodes sont très couramment utilisées dans le traitement du langage naturel car permettent de représenter plusieurs dérivées d'un mot sous un même mot donc nous allons uniquement garder la racine du mot. [24]

Le processus de « lemmatisation » consiste à représenter les mots sous leur forme canonique. Par exemple pour un verbe, ce sera son infinitif. Pour un nom, son masculin singulier. L'idée étant encore une fois de ne garder que le sens des mots utilisés dans le corpus. [25]

Dans le cas du Stemming, consiste à ne retenir que la racine des mots étudiés. Le but étant de supprimer les suffixes, préfixes et autres des mots afin de ne conserver que leur origine. [25] C'est un procédé plus simple que la lemmatisation et plus rapide à exécuter puisqu'on les mots sont essentiellement tronqués, contrairement à la lemmatisation qui nécessite un dictionnaire. [25]

4. Suppression de stopwords

Vient ensuite l'étape de suppression des stopwords qui est cruciale, car elle va supprimer dans le texte tous les mots qui n'ont que peu d'intérêt sémantique. Les mots vides sont en effet tous les mots les plus courants d'une langue (déterminants, pronoms, etc..) qui n'ont aucune valeur informative pour la compréhension du sens d'un document et corpus. Ils sont très fréquents et ralentissent notre travail : nous voulons donc les supprimer. [24]

5. Autres techniques :

Le prétraitement peut inclure d'autres étapes telles que la normalisation de la casse ou la suppression des accents. Le contexte des réseaux sociaux implique également un certain nombre de traitements spécifiques, d'où on peut citer :

- Suppression des liens hypertextes : Généralement, lors de l'analyse des sentiments sur du texte, les URL ne fournissent aucune information, au contraire, les URL déforment la prédiction de la subjectivité et de la polarité d'un texte donné. En prenant l'exemple de « www.magnifique.fr », ce texte serait de classe positive, alors qu'il est totalement neutre, cette mauvaise prédiction serait due à la présence du mot opinion dans l'url.
- Supprimer les lettres en double : les internautes ont tendance à utiliser une série de lettres en double dans le même mot pour exprimer la force du sens du mot. Cependant, ce genre de répétition de lettres produira des mots erronés qui

ne se trouvent pas dans le dictionnaire, c'est pourquoi les lettres obsolètes doivent être éliminées.

- Substitution de hashtags : les hashtags sont une sorte de marque sous la forme d'un ou plusieurs mots reliés entre eux et commençant par un dièse. Ce style d'écriture est souvent utilisé sur les réseaux sociaux, en taguant le contenu avec des mots-clés pour les mettre en valeur.

III. Les représentations de textes

I. Représentation par sac de mots

Le modèle du sac de mots est une représentation simplificatrice utilisée dans le traitement du langage naturel et la récupération d'informations. [26]

Dans ce modèle, un texte (tel qu'une phrase ou un document) est représenté comme le sac (multi set) de ses mots, sans tenir compte de la grammaire ni même de l'ordre des mots, en conservant toutefois la multiplicité. [26]

Il est couramment utilisé dans les méthodes de classification de documents dans lesquelles l'occurrence de chaque mot est utilisée comme caractéristique pour la formation d'un classifieur. [26]

II. Représentation par TF-IDF

TF-IDF sont les acronymes de « Terme Frequency » et « Inverse Document Frequency ». Ils suivent la logique du Cosinus de Salton. On cherche à accorder une pertinence lexicale à un terme au sein d'un document. En ce qui concerne TF-IDF, on applique une relation entre un document, et un ensemble de documents partageant des similarités en matière de mots clés. On recherche en quelque sorte une relation de quantité / qualité lexicale à travers un ensemble de documents. [27]

Pour une requête avec un terme X, un document a plus de chances d'être pertinent se à la requête, si ce document possède une certaine occurrence de ce terme en son sein, et que ce terme possède une rareté dans d'autres documents reliés au premier. [27]

Équation : TF-IDF

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

Où

t : désigne les termes ;

d : désigne chaque document ;

D : désigne la collection de documents.

[28]

Équation : IDF

$$idf(t, D) = \log \frac{|D|}{1 + |\{d \in D : t \in d\}|}$$

Le numérateur : D : infère à notre espace documentaire. Il peut également être vu comme $D = d_1, d_2, \dots, d_n$ où n est le nombre de documents de votre collection. [28]

Le dénominateur : $|\{d \in D : t \in d\}|$ implique le nombre total de fois où le terme t est apparu dans tout votre document d (le $d \in D$ restreint le document à être dans votre espace de document actuel). Notez que cela implique que peu importe si un terme apparaît 1 fois ou 100 fois dans un document, il sera toujours compté comme 1, car il apparaît simplement dans le document. Quant au plus 1, il est là pour éviter la division par zéro. [28]

Ci-dessous, un exemple de la représentation tf-idf.

III. Représentation par Word2Vec

Word2vec est une technologie de traitement du langage naturel. L'algorithme word2vec utilise un modèle de réseau neuronal pour apprendre les associations de mots à partir d'un grand corpus de texte. Une fois formé, ce modèle peut détecter des synonymes ou suggérer d'autres mots pour certaines phrases. Comme son nom l'indique, word2vec représente chaque mot différent, avec une liste spécifique de nombres appelés vecteur. Les vecteurs sont soigneusement sélectionnés afin qu'une simple fonction mathématique (similarité cosinus entre vecteurs) indique la similitude sémantique entre les mots représentés par ces vecteurs. [29]

Annexe 2

I. Classification et analyse de sentiments

1. Classification avec Machine learning

Le machine learning ou apprentissage automatique est un domaine scientifique, et plus particulièrement une sous-catégorie de l'intelligence artificielle. Elle consiste à laisser des algorithmes découvrir des "patterns", à savoir des motifs récurrents, dans les ensembles de données. Ces données peuvent être des chiffres, des mots, des images, des statistiques... [30]

L'apprentissage automatique utilise des méthodes statistiques et des algorithmes auto-améliorés, tout comme les humains apprennent de l'expérience passée, mais il est généralement plus efficace.

Tout ce qui peut être stocké numériquement peut être utilisé comme données pour le l'apprentissage automatique. En détectant les patterns dans ces données, les algorithmes apprennent et améliorent leurs performances lors de l'exécution de tâches spécifiques. [30]

Parfois, interpréter des modèles ou extraire des informations à partir des données ne peut pas être une tâche évidente. À mesure que la disponibilité des ensembles de données augmente, la demande d'algorithmes d'apprentissage automatique qui gèrent ces complexités augmente également.

On a 3 catégories d'apprentissage : apprentissage supervisé, apprentissage non supervisé et apprentissage semi-supervisé.

Mais dans cette partie on s'intéresse à l'apprentissage supervisé.

Contrairement à l'apprentissage non supervisé, l'apprentissage supervisé est une tâche d'apprentissage automatique qui apprend des fonctions de prédiction à partir d'exemples annotés.[31]

Cette catégorie a tendance à utiliser un ensemble de données pour générer un modèle qui utilise un vecteur de caractéristiques x pour entrer et sortir des informations pour obtenir l'étiquette correspondante du vecteur de caractéristiques.

Notre entrée n'est donc plus l'entrée de la méthode de résolution classique que nous avons définie dans le passé. Même la valeur de sortie réelle est considérée ici comme « l'entrée » de notre modèle.

Les applications d'apprentissage supervisé sont généralement divisées en deux catégories, la classification et la régression. Par exemple, lorsque la valeur de sortie est une catégorie telle que le genre, la couleur, la marque etc... qui est vraie ou fausse, la classification est pertinente. Lorsque la sortie est une valeur calculée réelle (comme le poids, le prix, le salaire), des problèmes de régression se produisent. [32]

Les algorithmes supervisés, avec des balises appliquées, incluent la régression linéaire, la régression logistique, réseaux de neurones, SVM, arbre de décision, Naïve Bayes, kNN, forêts aléatoires, etc.

Techniques ML

a. Random Forest

Les forêts d'arbres décisionnels ou forêts aléatoires (Random Forest) sont une technique d'apprentissage d'ensemble qui s'appuie sur des arbres de décision. Le modèle de forêt aléatoire implique la création de plusieurs arbres de décision en utilisant un ensemble de données séparés à partir des données d'origine. Et en sélectionnant au hasard un sous-ensemble de variables à chaque étape de l'arbre de décision. Ensuite, le modèle sélectionne tous les modes prédits pour chaque arbre de décision. [33]

b. Régression logistique

La régression logistique est similaire à la régression linéaire, mais elle est utilisée pour modéliser la probabilité d'un nombre limité de résultats (généralement deux). Il y a plusieurs raisons d'utiliser la régression logistique au lieu de la régression linéaire lors de la modélisation des probabilités de résultats. [33]

Une équation logistique est créée de telle sorte que les valeurs des résultats ne peuvent être qu'entre 0 et 1 (voir ci-dessous).

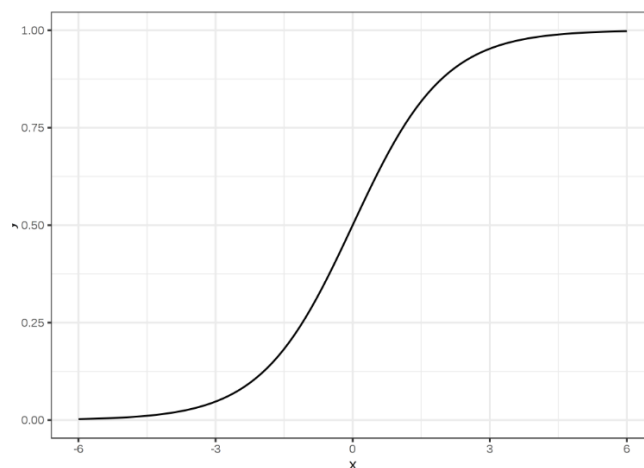


Figure 49 : Equation logistique de la RL [33]

- **Paramètres**

Multi_class :

La régression logistique ordinaire ou régression logistique binaire vise à expliquer une variable d'intérêt binaire (c'est-à-dire de type « oui / non » ou « vrai / faux »). Les variables explicatives qui seront introduites dans le modèle peuvent être quantitatives ou qualitatives.

La régression logistique multinomiale est une extension de la régression logistique aux variables qualitatives à trois modalités ou plus, la régression logistique ordinaire aux variables qualitatives à trois modalités ou plus qui sont ordonnées hiérarchiquement

c. Support Vector Machine (SVM)

La machine à vecteurs de support est une technique de classification supervisée, qui peut en fait devenir assez compliquée, mais elle est très intuitive au niveau le plus élémentaire. [33]

On suppose qu'il existe deux types de données. La machine à vecteurs de support trouvera un hyperplan ou une frontière entre les deux classes de données, ce qui maximisera la marge entre les deux classes (voir ci-dessous). Il existe plusieurs plans pour séparer les deux catégories, mais un plan peut maximiser la marge ou la distance entre les catégories. [33]

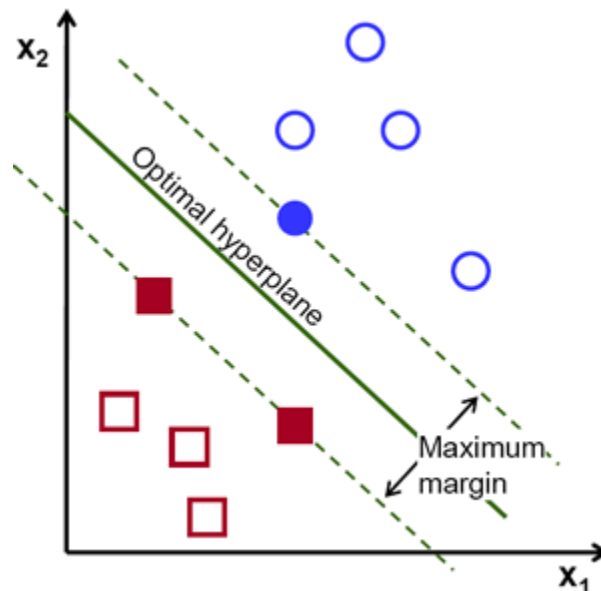


Figure 50 : La méthode SVM [33]

2. Classification avec le deep learning

Dans cette partie, nous allons définir en premier c'est quoi l'analyse de sentiments. Ensuite, nous allons nous attarder quelques instants sur quelques notions sur le deep learning. Cette partie se termine par la description des deux modèles que nous avons utilisés afin de classer les tweets en fonction des sentiments et par la suite l'affichage des résultats obtenus.

a. Analyse de sentiments

L'analyse des sentiments (également appelée analyse des opinions) fait référence à l'utilisation du traitement du langage naturel (NLP) et de l'apprentissage automatique pour identifier et caractériser les états affectifs et les opinions à partir de textes ou de données vocales. L'analyse des sentiments peut être appliquée aux avis des clients, aux billets de blogue, ou micro-blogues (Twitter) aux réponses aux enquêtes et aux médias sociaux. [35]

b. Notions sur DL

Nous ne pouvons pas définir le mot apprentissage en profondeur sans savoir ce que sont les réseaux de neurones. En regardant l'architecture du cerveau, on voit très bien son énorme complexité. Mais comment cela peut-il nous inciter à construire une machine « intelligente » ?

i. Deep Learning (DL)

C'est une sous-catégorie de l'apprentissage automatique. Tout comme l'apprentissage automatique, l'apprentissage en profondeur comprend également l'apprentissage supervisé, non supervisé et par renforcement. Comme mentionné précédemment, l'idée de l'IA a été inspirée par le cerveau humain. Essayons donc de relier les morceaux ici. L'apprentissage en profondeur s'inspire des réseaux de neurones artificiels, tandis que le réseau de neurones artificiels communément appelé ANN est inspirée des réseaux de neurones biologiques humains.[37]

ii. Méthode d'apprentissage en profondeur

Il existe différentes manières d'appliquer le deep learning. Chaque méthode proposée à un cas d'utilisation spécifique, tel que le type de données dont vous disposez, si vous souhaitez appliquer un apprentissage supervisé ou non supervisé, et quelles tâches souhaitez-vous utiliser pour les données à résoudre. Par conséquent, sur la base de ces facteurs, nous pouvons choisir l'une des meilleures solutions à nos problèmes.

Certaines des méthodes d'apprentissage en profondeur sont :

- **Réseau neuronal récurrent (RNN)**

Un réseau de neurones récurrents est un réseau de neurones artificiels présentant des connexions récurrentes. Un réseau de neurones récurrents est constitué d'unités (neurones) interconnectées interagissant non linéairement et pour lesquelles il existe au moins un cycle dans la structure. Les unités sont reliées par des arcs (synapses) qui possèdent un poids. [38]

Ce sont des réseaux de neurones et les informations peuvent se propager dans les deux sens, de la couche profonde à la première couche. Dans ce cas, ils sont plus proches de la vraie fonction du système nerveux, plutôt qu'unilatéral. Ces réseaux ont des connexions périodiques dans le sens de garder des informations en mémoire : ils peuvent considérer un certain nombre d'états passés à tout moment. [39]

Les réseaux de neurones répétitifs conviennent aux données d'entrée de différentes tailles. Ils sont particulièrement utiles pour l'analyse de séries chronologiques. Ils sont utilisés pour la reconnaissance automatique de la parole ou de l'écriture manuscrite. [38]

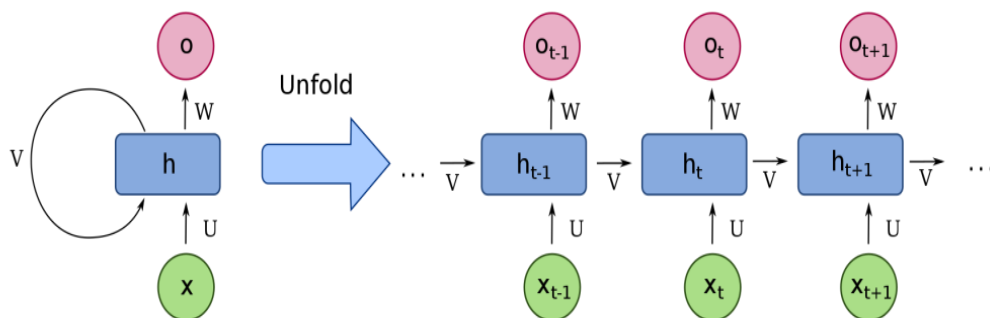


Figure 51 : Architecture RNN [38]

- **Auto-encodeurs**

Les auto-codeurs sont des algorithmes d'apprentissage non supervisés basés sur des réseaux de neurones artificiels qui vous permettent de créer une nouvelle représentation d'un ensemble de données. Globalement, il est plus compact et comporte moins de descripteurs, ce qui réduit la dimensionnalité de l'ensemble de données. L'architecture de l'auto encodeur se compose de deux parties : un encodeur et un décodeur. [42]

L'image de la figure ci-dessous montre l'architecture d'un auto-encodeur.

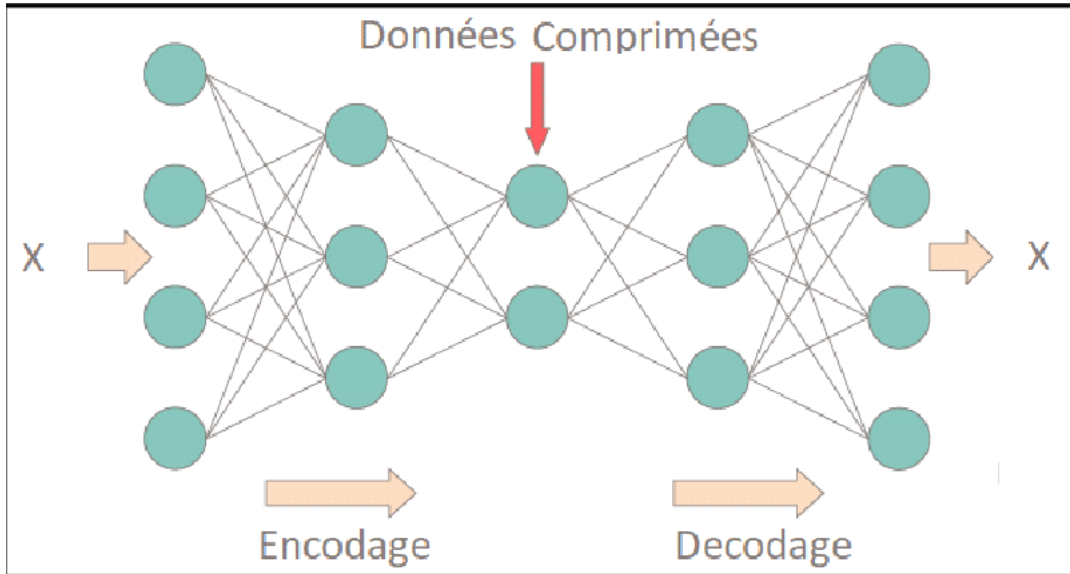


Figure 52 : Architecture d'un auto encodeur

L'encodeur prend des données de grande taille et les compresse dans une taille plus petite. [43]

Le décodeur prend des données de petite taille et les projette à une plus grande taille. [43]

La valeur centrale est appelée le code ; elle doit contenir les informations de l'entrée de manière compressée. [43]

L'apprentissage de l'auto-encodeur se fait par la rétro-propagation du gradient. C'est simplement un réseau qui cible l'entrée elle-même. [43]

c. Modèles de classification : CamemBERT

En raison de l'efficacité du modèle de langage pré-entraîné dans de nombreuses tâches en dessous de la NLP, il a reçu beaucoup d'attention.

Il a été démontré que le modèle de langage pré-entraîné améliore réellement de nombreuses tâches de traitement du langage, telles que la classification des sentiments

Bidirectional Encoder Representations from Transformers (BERT) est un modèle de représentation du langage pré-entraîné basé sur la technologie du DL proposé par l'équipe d'intelligence artificielle de Google 2018. [40]

Différent des autres modèles de représentation du langage, grâce aux contextes gauche et droit de toutes les couches, BERT peut générer une représentation bidirectionnelle à partir d'un texte d'entrée non étiqueté. [49] Il a été appliqué dans diverses tâches NLP, telles que la classification de texte et la réponse aux questions, et a réalisé une excellente performance. [49]

Camembert est un transformateur type BERT que Facebook a construit pour la langue française issu d'un travail entre Facebook AI Research, l'Inria et ALMANach.. [44]

3. Autre modèle : Text Blob

TextBlob est un package basé sur les librairies Python pour effectuer des opérations d'analyse de texte simples et complexes sur des données textuelles telles que le balisage vocal, l'extraction de phrases, l'analyse de sentiments, la classification, la traduction, etc. [36] Ce sont des librairies d'analyse de textes où il y a des dictionnaires qui pour chaque mot nous donnent la polarité de ce mot (le score de sentiment).

Annexe 3

I. Détection d'anomalies

• Série temporelle

Une série temporelle, ou série chronologique, est une suite de valeurs numériques qui représentent le changement d'une quantité spécifique au fil du temps. Une telle séquence de variables aléatoires peut être représentée mathématiquement afin d'analyser son comportement, et il est généralement possible de comprendre son évolution passée et de prédire son comportement futur. Ce virage mathématique utilise le plus souvent les concepts de probabilité et de statistique. [46]

• Détection d'anomalies

Dans l'exploration de données, la détection d'anomalies consiste à identifier des éléments rares, des événements ou des observations à l'origine de soupçons. Ces résultats sont très différents de la plupart des autres données. En règle générale, les anomalies représentent des problèmes tels que la fraude bancaire, des défauts structurels, des problèmes médicaux ou des erreurs de texte. Les intrusions sont également appelées valeurs aberrantes, bruit, écarts ou anomalies.

Dans le cadre de la détection d'intrusions sur les réseaux informatiques, les objets intéressants ne sont généralement pas des objets rares, mais des pics d'activité inattendus. Le modèle ne suit pas la définition des anomalies car il s'agit d'un objet rare, et de nombreuses méthodes de détection d'anomalies (en particulier les méthodes non supervisées) ne peuvent identifier ces anomalies que si l'anomalie a été correctement identifiée. Dans ce cas, l'algorithme d'analyse de partition de données peut être en mesure de détecter ces problèmes. [45]

• Modèle de détection

L'une des méthodes de prévision de séries temporelles les plus répandues est la méthode ARIMA. Un modèle connu par ses bonnes performances.

La deuxième méthode est les réseaux LSTM qui sont une extension pour les réseaux neuronaux récurrents, qui étendent leur mémoire. Ces réseaux sont connus par leurs bonnes performances sur la détection d'intrusions et qui permettent de prendre de meilleures décisions.

a. ARIMA

ARIMA signifie : AutoRegressive Integrated Moving Average C'est un modèle qui prédit la valeur future d'une série chronologique dans certains aspects de la structure statistique de la série observée. [47]

Composantes du modèle

Le modèle ARIMA est une généralisation, pour les séries non-stationnaires, du modèle ARMA qui est lui-même la composition des modèles AR (auto-régressif) et MA (Moyennes Glissante ou *Moving Average*). [47]

Modèle AR :

Ce modèle se base sur le caractère auto-régressif de la série. Il est donc applicable qu'aux séries auto-régressives. Une série (ou un processus) est auto-régressif d'ordre n lorsque sa valeur à un instant t dépend linéairement des n valeurs précédentes. [47]

$$x_t = c + \epsilon_t + \sum_{i=1}^n p_i x_{t-i}$$

Où ϵ est un bruit blanc et c une constante. Appliquer le modèle AR revient donc à trouver les coefficients p_i ainsi que la variance du bruit ϵ et la constante c . [47]

Modèle MA :

Ce modèle considère que la série (ou la variable) peut s'écrire comme combinaison linéaire de valeur actuelle d'un processus stochastique et de ses n valeurs précédentes. On parle d'un MA d'ordre n . La série peut donc s'écrire de la façon suivante : [47]

$$x_t = \mu + \sum_{i=1}^n \theta_i \epsilon_{t-i}$$

b. Mémoire longue à court terme (LSTM)

Un réseau Long short-term memory (LSTM), ou encore Mémoire longue à court terme, est l'architecture la plus couramment utilisée dans la pratique qui peut résoudre le problème de l'évanouissement du gradient. Le réseau LSTM a été proposé par Sepp Hochreiter et Jürgen Schmidhuber en 1997. L'idée derrière le LSTM est que chaque unité de calcul est associée non seulement à l'état latent h , mais aussi à l'état c de la cellule mémoire. Le passage de c_{t-1} à c_t se fait par transfert à gain constant égal à 1. De cette manière, l'erreur sera propagée à l'étape précédente (jusqu'à 1000 pas dans le passé) sans la disparition du gradient. L'état de l'unité peut être modifié en autorisant ou en bloquant la porte mise à jour (porte d'entrée). Il y a aussi, une porte qui vérifie si l'état de l'unité est véhiculé sur la sortie de la cellule LSTM (porte de sortie). La version la plus étendue de LSTM utilise également des portes (porte d'oubli) qui permettent de réinitialiser l'état de la cellule. [38]

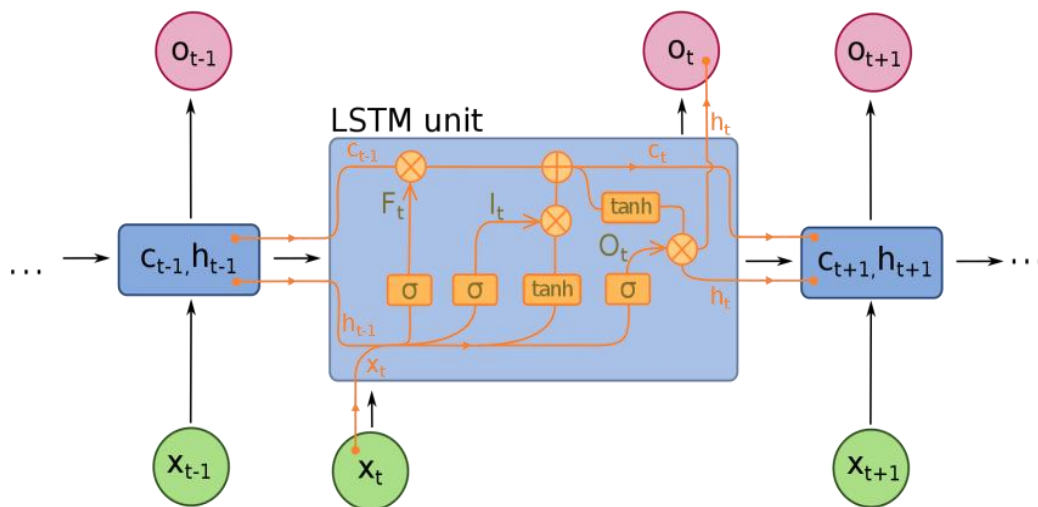


Figure 53 : Architecture du modèle LSTM [38]

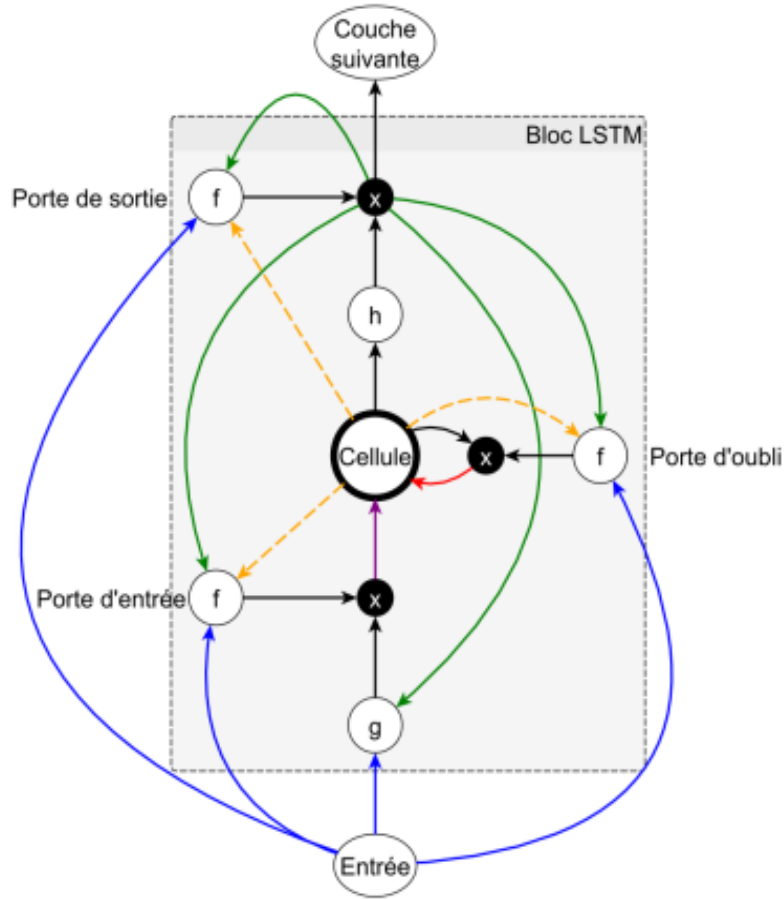


Figure 54 : Architecture du modèle LSTM [48]

Valeur initiale : les opérateurs $c = 0$ et $h = 0$ représentent le produit matriciel d'Hadamard (le produit du terme et du terme). Les symboles sigma et tanh désignent respectivement la fonction sigmoïde et la fonction tangente hyperbolique, bien que d'autres fonctions d'activation soient également possibles. [48]

$$\begin{aligned}
 F_t &= \sigma(W_F x_t + U_F h_{t-1} + b_F) \\
 I_t &= \sigma(W_I x_t + U_I h_{t-1} + b_I) \\
 O_t &= \sigma(W_O x_t + U_O h_{t-1} + b_O) \\
 c_t &= F_t \circ c_{t-1} + I_t \circ \tanh(W_c x_t + U_c h_{t-1} + b_c) \\
 h_t &= O_t \circ \tanh(c_t) \\
 o_t &= f(W_o h_t + b_o)
 \end{aligned}$$