

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université de Carthage  
Ecole Supérieure de la Statistique et de l'Analyse de  
l'Information



*Rapport de projet de fin d'études pour l'obtention du*

DIPLOME D'INGÉNIEUR EN STATISTIQUE ET ANALYSE DE L'INFORMATION

*Réalisé par*

**Mohamed Abderrahim JRIDI**

---

Classification des commentaires publiés sur les réseaux sociaux : Segmentation des prospects pour la banque Zitouna

---

Soutenu le 15/06/2022 devant le Jury composé de :

Mr. Ghazi BEL MUFTI,

Président du Jury

Mr. Mokhtar KOUKI,

Rapporteur

Mr. Farouk MHAMDI,

Encadrant universitaire

Mr. Marwen BEN NASR,

Encadrant de l'entreprise

*Réalisé À :*



Année Universitaire 2021/2022

## **Remerciements**

Je tiens à remercier toute l'équipe de la banque Zitouna pour m'avoir accepté comme étant un stagiaire et pour leurs confiances qu'ils m'ont accordées.

Je voudrais remercier évidemment Mr. Marwen BEN NASR, représentant de la Direction Organisation et Méthodes de la Banque Zitouna, pour sa patience, son aide et pour ses conseils pertinents.

Je présente ma gratitude à mon encadrant universitaire et mon enseignant à l'École Supérieure de la Statistique et de l'Analyse de l'Information, Mr. Farouk MHAMDI de m'avoir accepté de m'encadrer et de m'avoir encouragé tout au long mon stage de fin d'études.

Je tiens à remercier encore ma famille et mes amis pour leurs encouragements tout au long de ma carrière universitaire et professionnelle.

Finalement, je tiens à remercier Mr. Ghazi BEL MUFTI, le président du jury, ainsi que le rapporteur, Mr. Mokhtar KOUKI, pour avoir accepté d'évaluer mon Projet de Fin d'Études.

## Résumé

Mon projet de fin d'études a été réalisé au sein de la Banque Zitouna afin d'obtenir mon diplôme d'Ingénieur en Statistique et Analyse de Données.

Le but de ce projet est, en premier lieu, de construire une base de données contenant les commentaires et les réactions des publications partagées sur la page Facebook de la banque Zitouna avec les méthodes du web scraping et de suivre, en deuxième lieu, les prospects qui sont qualifiés pour être de nouveaux clients pour la banque Zitouna. Ceci est dans le but d'augmenter le nombre de clients pour la banque et pour améliorer les services du marketing digitale.

**Mots clés** — Web scraping, base de données, les prospects, machine learning, deep learning, traitement automatique du langage naturel, marketing digitale

## Abstract

My graduation project was carried out at the Zitouna Bank in order to obtain my diploma in Statistics and Data Analysis.

The goal of this project is, firstly, to build a database containing comments and reactions of the publications shared on the Facebook page of Zitouna bank using web scraping techniques and to trace, secondly, the prospects that are qualified to be new customers for Zitouna bank. The purpose is to increase the number of customers for the bank and to improve the digital marketing services.

**Mots clés** — Web scraping, database, prospects, machine learning, deep learning, natural language processing, digital marketing

# Table des matières

<b>Remerciements</b>	<b>1</b>
<b>Résumé/Abstract</b>	<b>2</b>
<b>Introduction générale</b>	<b>11</b>
<b>1 Présentation du cadre du Projet</b>	<b>12</b>
1.1 Introduction . . . . .	12
1.2 Organisme d'accueil : Banque Zitouna . . . . .	12
1.3 Présentation du projet . . . . .	14
1.3.1 Problématique . . . . .	14
1.3.2 Objectif . . . . .	14
1.4 Méthodologie du travail : MLOps . . . . .	15
1.5 Conclusion . . . . .	16
<b>2 État de l'art</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.2 Traitement automatique du langage naturel . . . . .	17
2.2.1 Partie linguistique : . . . . .	17
2.2.2 Partie modélisation . . . . .	19
2.3 Les couches cachées utilisées pour la modélisation NLP . . . . .	19
2.3.1 RNN : Réseau de neurones récurrent . . . . .	19
2.3.2 LSTM : Mémoire à long et court terme . . . . .	20
2.4 Méthodes d'ensemble : Bagging vs Boosting . . . . .	21
2.4.1 Bagging . . . . .	21

2.4.2	Boosting . . . . .	22
2.5	Métriques d'évaluation de la performance des modèles . . . . .	23
2.5.1	Matrice de confusion . . . . .	23
2.5.2	Accuracy . . . . .	23
2.5.3	F score . . . . .	24
2.5.4	Recall . . . . .	24
2.5.5	Précision . . . . .	24
2.6	Overfitting vs Underfitting . . . . .	25
2.6.1	Sous-apprentissage, underfitting . . . . .	25
2.6.2	Sur-apprentissage, overfitting . . . . .	26
2.7	Conclusion . . . . .	27
<b>3</b>	<b>Collecte des données</b>	<b>28</b>
3.1	Introduction . . . . .	28
3.2	Types des bases de données . . . . .	28
3.2.1	Base de données structurée/ Relational database . . . . .	29
3.2.2	Base de données non structurée/ Non-relational database . . . . .	29
3.2.3	MongoDB . . . . .	30
3.3	Préparation de l'environnement de travail . . . . .	32
3.4	Les librairies nécessaires . . . . .	33
3.5	Méthodologie du web scraping . . . . .	35
3.6	Exportation des données dans un fichier excel . . . . .	38
3.6.1	Création d'une base de données pour l'étude NLP . . . . .	39
3.6.2	Création d'une base de données pour la segmentation des prospects . . . . .	41
3.7	Conclusion . . . . .	42
<b>4</b>	<b>Classification des commentaires et segmentation des prospects</b>	<b>43</b>
4.1	Introduction . . . . .	43
4.2	Traitement du langage propre : NLP . . . . .	43
4.2.1	Pré-traitement des données . . . . .	43
4.2.2	Visualisation des données . . . . .	47

4.3	Segmentation des prospects . . . . .	61
4.3.1	Ingénierie des variables . . . . .	61
4.3.2	Statistique descriptive . . . . .	65
4.3.3	Modélisation . . . . .	66
4.4	Comparaison des modèles : . . . . .	69
4.5	Conclusion . . . . .	69
<b>5</b>	<b>ChatBot : déploiement des modèles</b>	<b>71</b>
5.1	Introduction . . . . .	71
5.2	Construction du chatbot . . . . .	72
5.2.1	Mécanisme de réception des messages sur la page Facebook de la banque Zitouna . . . . .	72
5.2.2	Étapes de renvoie des réponses aux utilisateurs . . . . .	74
5.2.3	Etablir la connection https avec la machine locale . . . . .	75
5.3	Présentation des fonctionnalités du ChatBot . . . . .	77
5.4	Conclusion . . . . .	78
	<b>Conclusion générale</b>	<b>80</b>

# Table des figures

1.1	Banque Zitouna . . . . .	12
1.2	Structure de gouvernance . . . . .	13
1.3	Déploiement du modèle avec la méthode MLOps(source : databricks.com) . . . . .	16
2.1	Réseau de neurones récurrent (source : simplilearn.com) . . . . .	20
2.2	Couche LSTM (source : lstm-scheduler by jiangyifangh) . . . . .	21
2.3	Matrice de confusion(source : lebigdata.fr) . . . . .	23
2.4	Overfitting vs Underfitting (source : What is Overfitting ?   IBM) . . . . .	25
2.5	Exemple d'un problème de overfitting (source : v7labs.com) . . . . .	27
3.1	Types des base de données (source :medium.com, Database models) . . . . .	29
3.2	MongoDB . . . . .	30
3.3	Méthodologie du web scraping . . . . .	32
3.4	Création et activation de l'environnement . . . . .	33
3.5	Selenium pour web scraping (source : Web Scraping with Selenium   Python Programming   Automation) . . . . .	34
3.6	WebDriver (source : testup.io) . . . . .	35
3.7	Enregistrement d'une publication . . . . .	36
3.8	Serveur mongoDB . . . . .	36
3.9	Algorithme du bot Scrapy . . . . .	37
3.10	Vue d'ensemble sur mongodb . . . . .	38
3.11	Itération des documents par un curseur (source : guru99.com) . . . . .	39
3.12	Données filtrées . . . . .	40
4.1	Table de données pour le traitement du NLP . . . . .	44

4.2	Table de données après pré-traitement . . . . .	47
4.3	Tokenization des commentaires . . . . .	48
4.4	Bar plot sur la fréquence des tokens . . . . .	49
4.5	Nuage de mots pour la classe autre . . . . .	50
4.6	Nuage de mots pour la classe "particulier" . . . . .	50
4.7	Nuage de mots pour la classe professionnelle . . . . .	51
4.8	Nuage de mots pour la classe service . . . . .	51
4.9	Architecture d'une couche LSTM (source : pluralsight.com) . . . . .	53
4.10	Exemple de tokenisation . . . . .	55
4.11	Interface mlflow . . . . .	56
4.12	Modèle 1 : NLP avec les couches LSTM . . . . .	57
4.13	Modèle 1 : Les métriques obtenues avec les couches LSTM . . . . .	57
4.14	Modèle 1 : Accuracy vs val accuracy . . . . .	58
4.15	Modèle 1 : Loss vs val loss . . . . .	58
4.16	modèle 3 : NLP Avec les couches RNN simples . . . . .	60
4.17	Comparaison entre les deux modèles . . . . .	60
4.18	Tableau pour le problème de segmentation . . . . .	61
4.19	Distribution de la variable target . . . . .	65
4.20	Distribution de la variable target selon la variable genre . . . . .	65
4.21	Distribution de la variable target selon la variable lieux de résidence . . . . .	66
4.22	Régression logistique : courbe de ROC . . . . .	67
4.23	Régression logistique : indicateurs de performances obtenus sur l'échantillon de test . . . . .	67
4.24	Forêt aléatoire : courbe de roc . . . . .	68
4.25	Random forest : indicateurs de performances obtenus sur l'échantillon de test . . . . .	68
4.26	XGboost : importance des variables . . . . .	69
5.1	Exemple d'une réponse lente de la part de la banque Zitouna . . . . .	71
5.2	Chatbot Messenger (source : thecodespace.in) . . . . .	72
5.3	Architecture django (source : developer.mozilla.org) . . . . .	73
5.4	Application django . . . . .	73

5.5	connection à ngrok . . . . .	76
5.6	Adresse URL de rappel . . . . .	77
5.7	Exemple d'une conversation avec le bot messenger . . . . .	77
5.8	Message reçu par un utilisateur . . . . .	78

# Liste des tableaux

2.1	Exemple : Tokenisation . . . . .	18
4.1	Préparation des commentaires pour la modélisation NLP . . . . .	52
4.2	Modèle 1 : Performances obtenues par classe . . . . .	59
4.3	Comparaison des résultats . . . . .	69

## Introduction générale

De nos jours, les données sont devenues l'une des ressources commerciales les plus précieuses de la planète. Lorsqu'il s'agit de prendre des décisions commerciales, la règle la plus fondamentale est de faire d'abord des recherches. Ce qui aide les chefs de projets de n'importe quelles entreprises à prendre les décisions les plus efficaces surtout avec l'accessibilité des données sur internet.

Il est important pour la banque, en vue de ses activités, d'améliorer ses relations avec ses clients. Mais évidemment augmenter et attirer de nouveaux clients n'est pas assez facile. Par contre, si le marketeur réussit à convaincre une personne de se profiter d'un produit offert par la banque, il contribuera, évidemment, à améliorer la relation entre la banque et ses clients.

L'un des plus grands défis pour les gestionnaires de projet est de savoir comment profiter de ces énormes quantités de données afin de comprendre les comportements des gens. Le rôle d'un data scientist est d'être capable de manipuler des milliers de données et de les exploiter pour aider les marketeurs et les chefs de projets à prendre les meilleures décisions.

Avec la connaissance du métier banque offerte par les experts de la banque Zitouna et le métier data science, on peut améliorer le système du CRM<sup>1</sup> et de mettre en place un chatbot lié à la page Facebook de la banque pour améliorer les relations entre les marketeurs et les clients.

Ce rapport distinguera essentiellement 5 chapitres : Dans le premier chapitre nous allons traiter le cadre du projet où nous parlerons de l'organisme d'accueil, l'objectif et la problématique de ce projet. Le deuxième chapitre a été consacré pour donner une synthèse sur l'état de l'art. C'est dans ce chapitre que nous traiterons les définitions du traitement automatique du langage naturel et de l'apprentissage automatique. Dans le troisième chapitre, nous aborderons le volet pratique de la collection des données via le web scraping. Dans le quatrième chapitre nous étudions la modélisation de données pour établir une classification des commentaires collectés et segmenter les prospects. Terminant avec le cinquième chapitre dans

---

1. Customer Relationship Management

lequel nous allons construire un chatbot basé sur les résultats obtenus par les méthodes TAL<sup>2</sup> traitées.

---

2. Traitement automatique du langage propre

# Chapitre 1

## Présentation du cadre du Projet

### 1.1 Introduction

Dans ce chapitre, nous présenterons le cadre du projet de fin d'études. Nous commençons par définir l'organisme d'accueil, une présentation du projet puis, nous terminons avec la méthodologie de travail.

### 1.2 Organisme d'accueil : Banque Zitouna

La banque Zitouna est une banque commerciale universelle, connue aussi comme une banque citoyenne, islamique à forte responsabilité sociale ayant une volonté à contribuer à l'expansion économique. Elle est créée en octobre 2009 et est commencée ses activités en Mai 2010. La banque Zitouna offre plusieurs produits et services conformes aux lois et principes de la finance islamique aux particuliers ,aux professionnels et aux entreprises.



FIGURE 1.1 – Banque Zitouna

Les missions de la banque Zitouna est répartie principalement autour de 4 axes : Partici-

per à la modernisation du système bancaire et financier national et contribuer au développement économique et social du pays, de répondre à une demande de plus en plus pressante pour des produits et services financiers conformes aux principes de la Finance Islamique, d'accompagner la clientèle dans les différentes phases de financement et d'assurer à la clientèle une excellente qualité de services et un conseil dévoué.

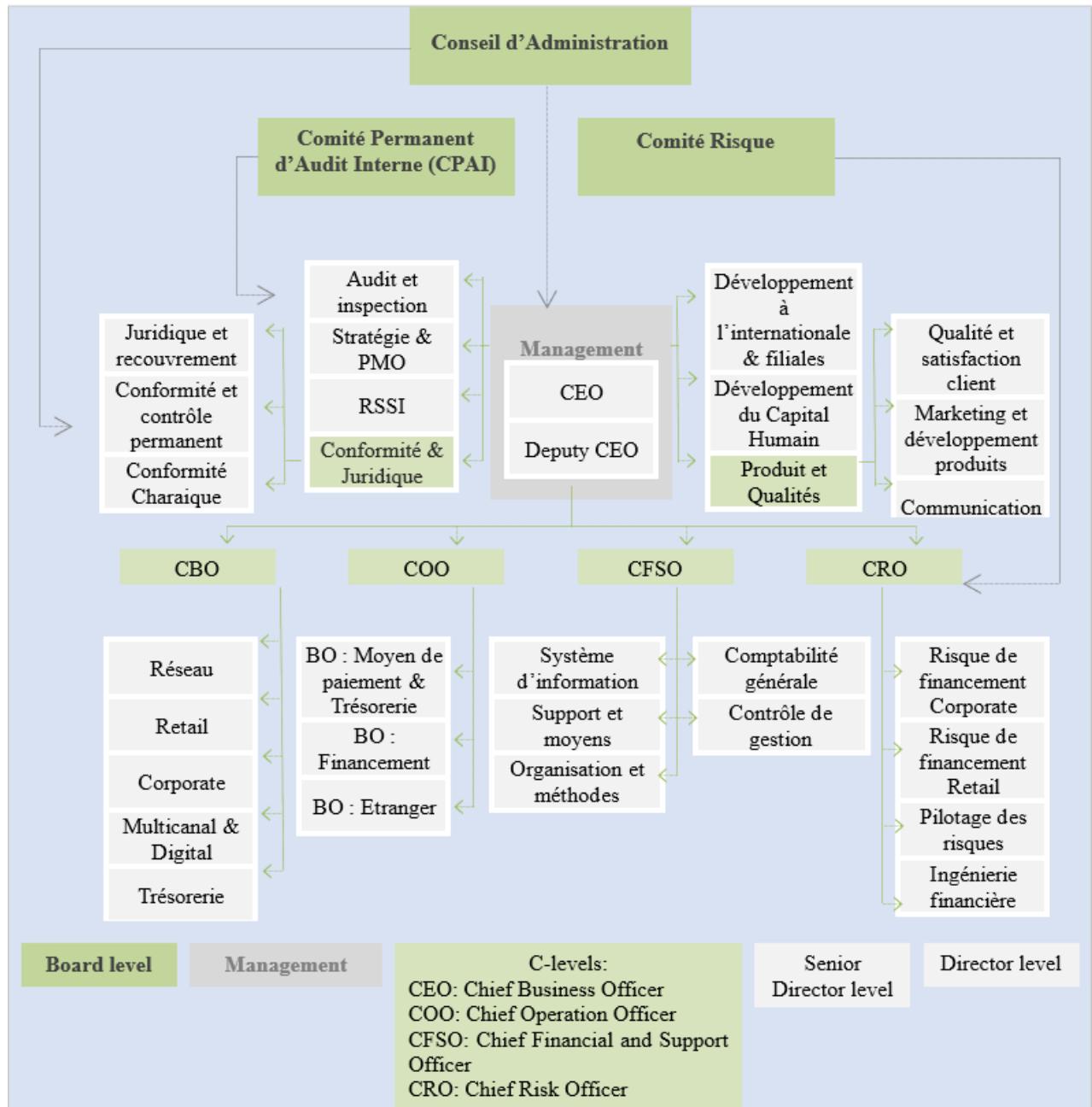


FIGURE 1.2 – Structure de gouvernance

Mon Stage du Projet de fin d'Étude s'est déroulé dans le département Produit et Qualités sous la supervision du chef du projet Mr. Marwen BEN NASR.

## 1.3 Présentation du projet

Les instituts bancaires, y compris la banque Zitouna, sont des instituts financières qui cherchent à maximiser leurs profits en offrant des services et des produits aux clients particuliers et professionnelles. Donc il est important d'augmenter le nombre de clients pour la banque Zitouna afin d'améliorer son profil.

### 1.3.1 Problématique

Avoir un site Web et des pages sur les réseaux sociaux est important pour communiquer les nouvelles de l'entreprise et avoir des retours d'expériences. Il peut aussi aider à trouver des faiblesses et des défauts d'une entreprise. Mais il ne suffit pas pour attirer des gens et rechercher des nouveaux clients.

La banque Zitouna offre ses services uniquement aux clients qui font leurs premiers pas vers une agence de la banque pour demander en retour un service bien déterminé.

Cette méthode est devenue de plus en plus défavorable pour la banque parce qu'un client, aujourd'hui, veut que quelqu'un lui propose un service mieux qu'il le fait à son tour. Nous vivons à l'heure de la transformation numérique.

De plus, Il existe toujours une gamme de personnes qui n'ont jamais essayé à interagir avec les banques et d'autres qui ne font pas confiance dans les banques et on peut trouver parmi eux des bons clients qui peuvent profiter des services offerts par la banque ainsi que la banque peut profiter d'eux mais cette classe de personnes ne présente actuellement que des manques à gagner pour la banque. Le grand défi est alors de collecter des informations sur cette classe des personnes en particulier et de les segmenter selon leurs centres d'intérêts afin de maximiser les clients du banque Zitouna.

### 1.3.2 Objectif

L'objectif de mon projet est de collecter, dans un premier lieu, les informations à partir des réseaux sociaux et de segmenter les prospects selon leurs centres d'intérêts dans une deuxième étape et de trouver ceux qui peuvent être intéressés par un service ou un produit particulier et pouvant être des clients pour la banque zitouna. Enfin, tout ce qui reste est de

mettre en place un chatbot pour ces nouveaux clients afin de rendre les choses plus agiles en leur expliquant clairement les procédures et les étapes.

## 1.4 Méthodologie du travail : MLOps

Pour bien expérimenter les modèles Machine Learning qu'on implémente et pour les déployer, Il est un peu nécessaire d'utiliser les méthodes MLOps.

MLOps est un ensemble de pratiques de collaboration et de communication entre les scientifiques des données (Data Scientists) et les professionnels des opérations. L'application de ces pratiques augmente la qualité, simplifie le processus de gestion et automatise le déploiement des modèles de Machine Learning et de Deep Learning dans des environnements de production à grande échelle.

Une des méthodes que nous utiliserons pour suivre, comparer et choisir le modèle Machine Learning le plus performant est la méthode Mlflow.

MLflow est une plateforme Open Source désignée pour gérer le cycle de vie du Machine Learning, y compris l'expérimentation, la reproductibilité, le déploiement des modèles. MLflow offre actuellement quatre composants :

- **MLflow Tracking** : pour enregistrer et interroger les expériences : code, données, configuration et résultats.
- **MLflow Projects** : pour programmer les algorithmes d'apprentissage automatique dans un format permettant de reproduire les exécutions sur n'importe quelle plate-forme.
- **MLflow Models** : pour déployer des modèles d'apprentissage automatique dans divers environnements de service.
- **Model Registry** : Pour stocker et gérer des modèles dans une base de données centrale.

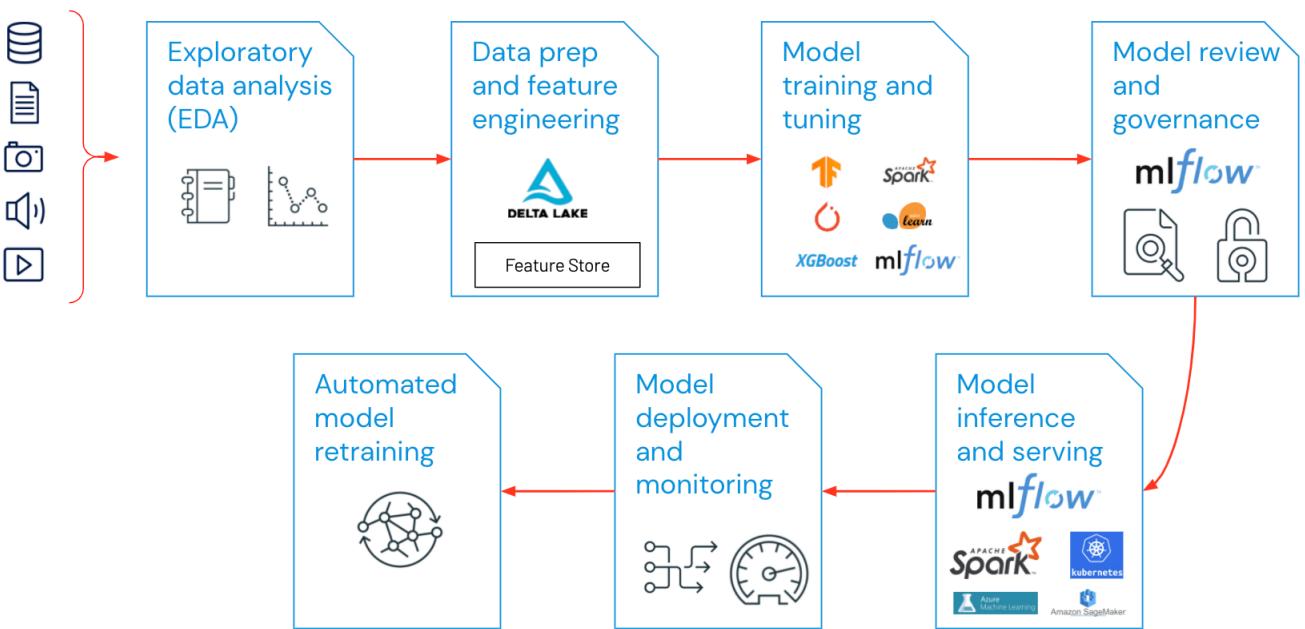


FIGURE 1.3 – Déploiement du modèle avec la méthode MLOps(source : databricks.com)

## 1.5 Conclusion

Dans ce chapitre, nous avons présenté l'organisme d'accueil l'objectif du projet et la méthodologie de travail. Dans le deuxième chapitre, nous allons détailler l'état de l'art du langage naturel ainsi que les techniques de Machine Learning que nous allons étudiées.

# Chapitre 2

## État de l'art

### 2.1 Introduction

Dans ce chapitre, nous clarifions les parties théoriques des méthodes utilisées pour le traitement automatique du langage naturel et les algorithmes d'apprentissages automatiques utilisés dans ce projet ainsi que les métriques de performances.

### 2.2 Traitement automatique du langage naturel

Le traitement automatique du langage naturel TAL, Connue aussi par le NLP, comporte la compréhension et la manipulation des textes et du langage naturel par les machines. C'est une combinaison entre la science des données, la science informatique et la linguistique.

Autrement, c'est la capacité d'une machine à comprendre le langage humain tel qu'il est parlé et écrit. On peut utiliser les méthodes et les techniques NLP dans la traduction des textes, l'analyse sentimentale, les chatbots, la reconnaissance vocale, ...

Dans ce projet, nous allons utiliser les techniques NLP pour la classification des textes.

Le traitement automatique du langage naturel se fait en deux étapes :**la partie linguistique et la partie modélisation.**

#### 2.2.1 Partie linguistique :

C'est la partie de pré-traitement des textes pour les transformer en données exploitables au niveau de la deuxième partie de la modélisation. Un texte est représenté par un ensemble

de mots. Le but de cette partie est d'examiner chaque mot séparément sans tenir compte du sens du texte.

**Vocabulaires :** Un texte est présenté comme un corpus et un mot sera présenté comme un token issu de la procédure Tokenisation.

Les étapes de pré-traitement d'un corpus sont les suivantes :

### Nettoyage :

Le nettoyage d'un corpus est important avant de passer aux autres procédures. Cette procédure consiste à retirer les émojies et les URL.

### Tokénisation

La tokénisation des textes consiste à découper chaque corpus en des tokens. Le but est de construire un dictionnaire de vocabulaire composés par toutes les tokens utilisés dans les corpus et d'attribuer à chaque token, la fréquence d'apparition dans un corpus.

Prenons l'exemple de ces deux corpus : "today is my day", "this is my day". Dans ce cas, grâce à la tokenisation, nous pouvons construire un dictionnaire contenant les tokens suivants :{today, is, my, day, this}. Maintenant, nous pouvons créer un tableau pour décrire combien de fois un token a été utilisé dans ces deux corpus.

Corpus	today	is	my	day	this
today is my day	1	1	1	1	0
this is my day	0	1	1	1	1
Somme	1	2	2	2	1

TABLE 2.1 – Exemple : Tokenisation

### Stemming

Un token peut être écrit dans plusieurs formes. Le stemming consiste à supprimer les préfixes et les suffixes de chaque token.

Par exemple en appliquant le stemming sur le mot "transformation", elle deviendra "transform"

## D'autres opérations

Il existe d'autres opérations pour nettoyer un corpus comme la suppression des chiffres, la suppression des ponctuations et des symboles, la suppression des mots d'arrêt (Stop Words) et de passer tous les mots en minuscules. Ces opérations sont nécessaires pour pouvoir découper un corpus en tokens par la procédure de tokenisation.

### 2.2.2 Partie modélisation

Dès que les données sont pré-traitées, il ne reste que de développer un algorithme d'apprentissage pour classifier les corpus qui sont des commentaires dans notre cas. Nous appuierons dans cette partie sur les méthodes de Deep Learning qui utilisent des couches de réseaux de neurones pour l'extraction automatique des caractéristiques de données.

## 2.3 Les couches cachées utilisées pour la modélisation NLP

### 2.3.1 RNN : Réseau de neurones récurrent

Le RNN<sup>1</sup> fonctionne sur le principe de l'enregistrement de la sortie d'une couche particulière et de sa réinjection dans l'entrée afin de prédire la sortie de la couche. Les noeuds des différentes couches du réseau de neurones sont compressés pour former une seule couche de réseau de neurones récurrent. Par exemple dans la figure ci dessous, "x" est la couche d'entrée, "h" est la couche cachée et "y" est la couche de sortie. A, B et C sont les paramètres du réseau utilisés pour améliorer la sortie du modèle. À tout moment t, l'entrée actuelle est une combinaison des entrées  $x(t)$  et  $x(t-1)$ . La sortie à un moment donné est renvoyée au réseau pour améliorer la sortie.

---

1. Recurrent Neural Network

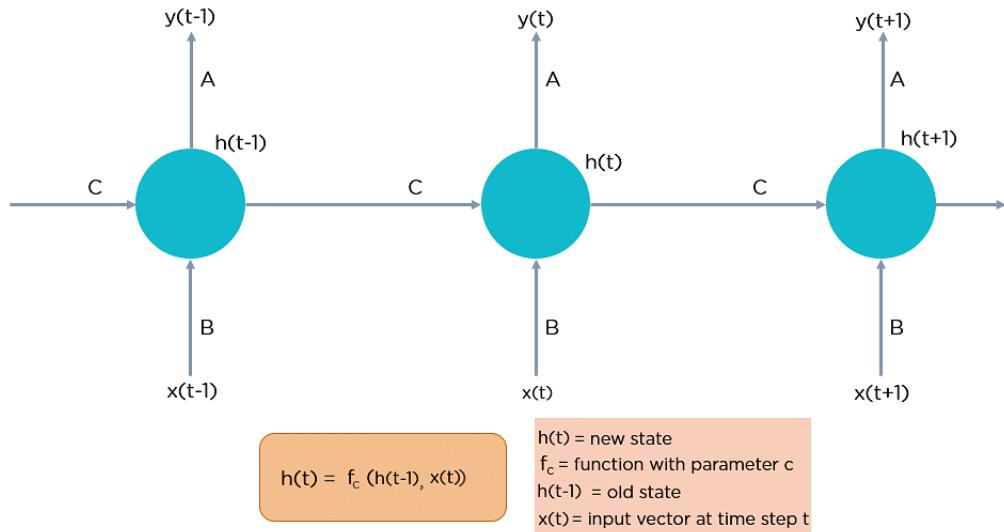


FIGURE 2.1 – Réseau de neurones récurrent (source : simplilearn.com)

En théorie, les RNN sont absolument capables de gérer de telles "dépendances à long terme". Un humain pourrait choisir avec soin leurs paramètres pour résoudre des problèmes de ce type. Le problème a été exploré par Hochreiter (1991) et Bengio, et al. (1994), qui ont trouvé des raisons fondamentales pour lesquelles cela pourrait être difficile.

### 2.3.2 LSTM : Mémoire à long et court terme

Les couches LSTM<sup>2</sup> sont des types particuliers d'un réseau de neurones récurrent, qui sont capables d'apprendre des dépendances à long terme. Ils ont été introduits par Hochreiter & Schmidhuber en 1997.

Tous les réseaux de neurones récurrents ont la forme d'une chaîne de modules répétitifs de réseau de neurones. Les couches LSTM ont la même structure en chaîne, mais le module itératif a une structure différente. Au lieu d'avoir une seule couche pour le réseau de neurones, il y en a quatre, qui interagissent d'une manière différente.

---

2. Long Short Term Memory

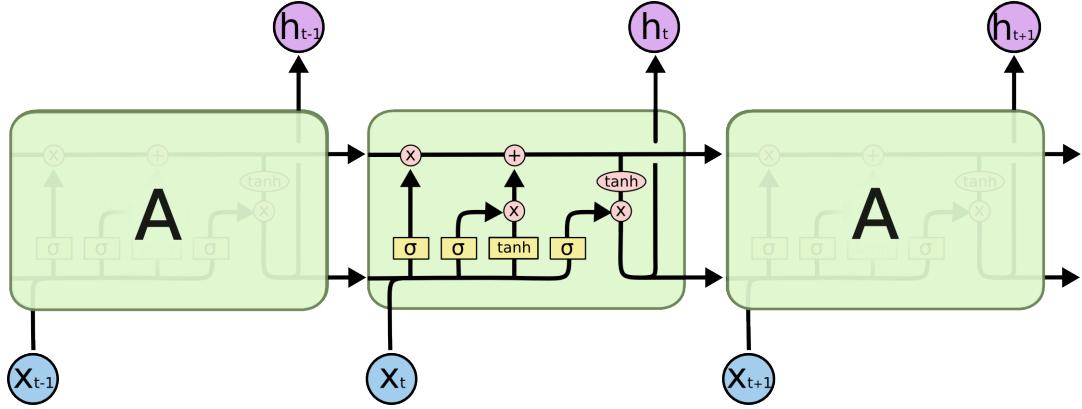


FIGURE 2.2 – Couche LSTM (source : lstm-scheduler by jiangyifangh)

## 2.4 Méthodes d'ensemble : Bagging vs Boosting

Bagging et boosting sont des techniques ensemblistes, où un ensemble d'apprenants faibles sont combinés pour créer un apprenant fort qui obtient de meilleures performances qu'un seul.

Les méthodes ensemblistes sont des concepts d'apprentissage automatique dans lequel plusieurs modèles sont formés à l'aide du même algorithme d'apprentissage. Le bagging est un moyen de réduire la variance de la prédiction en générant des données supplémentaires pour la formation à partir d'un ensemble de données en utilisant des combinaisons avec des répétitions pour produire des ensembles multiples de données originales. Le boosting est une technique itérative qui ajuste le poids d'une observation en fonction de la dernière classification. Si une observation a été classée de manière incorrecte, elle tente d'augmenter le poids de cette observation. Le boosting permet en général de construire des modèles prédictifs forts.

### 2.4.1 Bagging

Le bagging, également connu sous le nom d'agrégation bootstrap, est une méthode d'apprentissage d'ensemble couramment utilisée pour réduire la variance dans un ensemble de données bruitées. Dans cette méthode, un échantillon aléatoire de données d'un ensemble d'apprentissage est sélectionné avec remplacement, ce qui signifie que les points de données individuels peuvent être choisis plus d'une fois. Après avoir généré plusieurs échantillons de données, ces modèles faibles sont ensuite entraînés indépendamment et, selon le type de tâche - régression ou classification, par exemple - la moyenne ou la majorité de ces prédictions donnent

une estimation plus précise.

Exemple : Les forêts aléatoires

1. Supposons qu'il y ait N observations et M caractéristiques dans l'ensemble de données d'apprentissage. Un échantillon de l'ensemble de données d'apprentissage est pris au hasard avec remplacement.
2. Un sous-ensemble de M caractéristiques est sélectionné aléatoirement et la caractéristique qui donne la meilleure division est utilisée pour diviser le nœud de manière itérative.
3. L'arbre est cultivé jusqu'au plus grand.
4. Les étapes ci-dessus sont répétées n fois et la prédiction est donnée sur la base de l'agrégation des prédictions d'un nombre n d'arbres.

## 2.4.2 Boosting

Les méthodes de boosting sont axées sur la combinaison itérative d'apprenants faibles pour construire un apprenant fort capable de prédire des résultats plus précis. Pour rappel, un apprenant faible classe les données légèrement mieux qu'une supposition aléatoire. Cette approche peut fournir des résultats robustes pour les problèmes de prédiction, et peut même surpasser les réseaux neuronaux et les machines à vecteurs de support, *SVM*

Les algorithmes de boosting peuvent différer dans la manière dont ils créent et regroupent les apprenants faibles au cours du processus séquentiel. Trois types populaires de méthodes de boosting incluent :

Boosting adaptatif ou AdaBoost, L'optimisation par gradient, XGBoost.

L'algorithme boosting :

1. Tirez un sous-ensemble aléatoire d'échantillons de formation d1 sans remplacement de l'ensemble de formation D pour former un apprenant faible C1.
2. Tirez un deuxième sous-ensemble de formation aléatoire d2 sans remplacement de l'ensemble de formation et ajoutez 50% des échantillons qui ont été précédemment faussement classés/misclassés pour former un apprenant faible C2.
3. Trouver les échantillons d3 dans l'ensemble de formation D sur lesquels C1 et C2 ne sont pas d'accord pour former un troisième apprenant faible C3.
4. Combinez tous les apprenants faibles par vote majoritaire.

## 2.5 Métriques d'évaluation de la performance des modèles

Lorsqu'on parle d'intelligence artificielle et de traitement du langage naturel, on parle souvent de précision, de recall, de F score et de l'accuracy. Ce sont des moyens pour mesurer la qualité d'une modélisation.

### 2.5.1 Matrice de confusion

Les matrices de confusion représentent les comptes des valeurs prédites et réelles. La sortie "TN" signifie True Negative qui indique le nombre d'exemples négatifs classés avec précision. De même, "TP" signifie True Positive, qui indique le nombre d'exemples positifs classés avec précision. Le terme "FP" indique une valeur de faux positif, c'est-à-dire le nombre d'exemples négatifs réels classés comme positifs ; et "FN" signifie une valeur de faux négatif qui est le nombre d'exemples positifs réels classés comme négatifs.

The Confusion Matrix

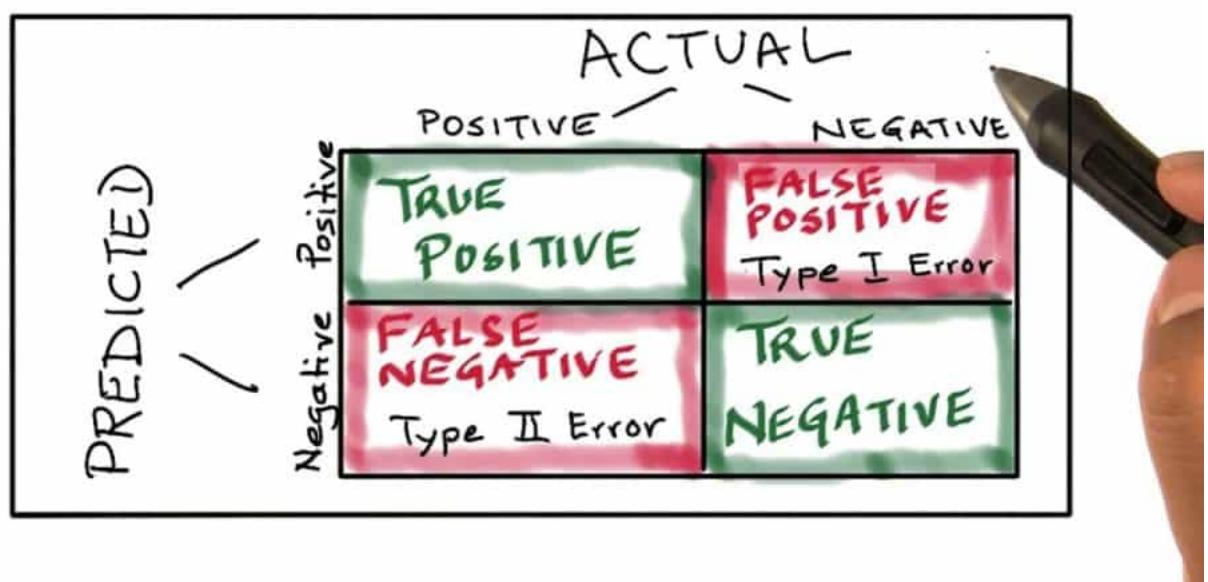


FIGURE 2.3 – Matrice de confusion(source : lebigdata.fr)

### 2.5.2 Accuracy

C'est une métrique qui décrit généralement les performances du modèle pour toutes les classes. Elle est utile lorsque toutes les classes sont d'importance égale. Elle est calculée comme

le rapport entre le nombre de prédictions correctes et le nombre total de prédictions.

$$Accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}}$$

### 2.5.3 F score

Le score F1 est la moyenne harmonique de la précision et du rappel. Ce score est généralisé par le score  $F_\beta$  qui applique des pondérations supplémentaires, en valorisant l'un de la précision ou du rappel plus que l'autre.

$$F\_score = \frac{precision * recall}{precision + recall}$$

La valeur la plus élevée possible d'un score F est 1,0, ce qui indique une précision et un rappel parfaits, et la valeur la plus faible possible est 0, si la précision ou le rappel est nul.

### 2.5.4 Recall

Le rappel est calculé comme le rapport entre le nombre d'échantillons positifs correctement classés comme positifs et le nombre total d'échantillons positifs. Le rappel mesure la capacité du modèle à détecter les échantillons positifs. Plus le rappel est élevé, plus le nombre d'échantillons positifs détectés est important.

$$Recall = \frac{True_{positive}}{True_{positive} + False_{negative}}$$

### 2.5.5 Précision

La précision est calculée comme le rapport entre le nombre d'échantillons positifs correctement classés et le nombre total d'échantillons classés comme positifs (soit correctement, soit incorrectement). La précision mesure l'exactitude du modèle à classer un échantillon comme positif.

$$Precision = \frac{True_{positive}}{True_{positive} + False_{positive}}$$

## 2.6 Overfitting vs Underfitting

Généralement, dans le cas où on obtient un modèle de mauvaise performance. Un de ces deux problèmes persistent : Problème de underfitting (sous-apprentissage) et problème de overfitting (sur-apprentissage).

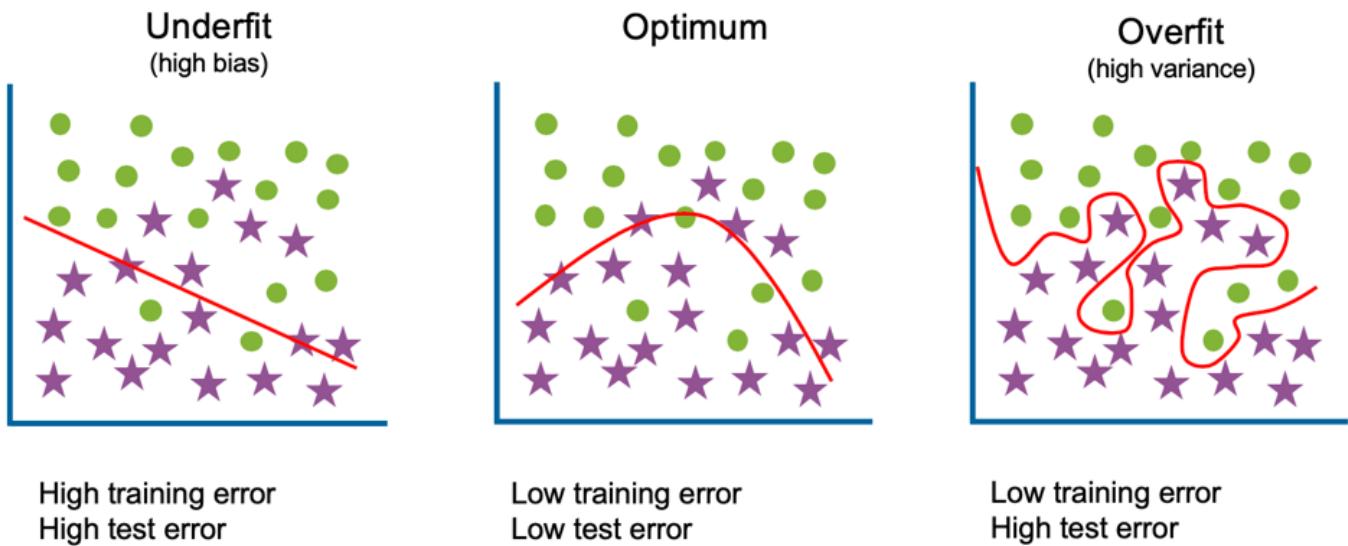


FIGURE 2.4 – Overfitting vs Underfitting (source : What is Overfitting ? | IBM)

### 2.6.1 Sous-apprentissage, underfitting

Après le développement des algorithmes de Machine learning ou de deep learning, on peut être face à un problème de sous-apprentissage. Les problèmes de underfitting sont des problèmes où le modèle n'est pas bien entraîné et il retourne de mauvaises prédictions même sur les données d'entraînement.

Le sous-apprentissage est un scénario en science des données dans lequel un modèle de données est incapable d'ajuster la relation entre les variables d'entrée et de sortie avec précision, générant un taux d'erreur élevé à la fois sur l'ensemble de formation et sur les données de test. Il se produit lorsqu'un modèle est trop simple, ce qui peut être le résultat d'un modèle nécessitant plus de temps d'apprentissage, plus de caractéristiques d'entrée ou moins de régularisation.

Pour corriger ce problème, on peut utiliser l'une de ces techniques :

- **Augmenter la complexité du modèle :** L'arrêt trop rapide de l'entraînement peut également entraîner un modèle sous-adapté. Par conséquent, en prolongeant la

durée de la formation, on peut l'éviter. Cependant, il est important d'être conscient du surentraînement et, par la suite, du sur-ajustement. Il est essentiel de trouver un équilibre entre ces deux scénarios.

- **Augmenter le nombre de caractéristiques et/ou des variables**
- **Augmenter le nombre d'époques**

## 2.6.2 Sur-apprentissage, overfitting

Il s'agit d'un écueil courant dans les algorithmes d'apprentissage profond dans lesquels un modèle tente de s'adapter entièrement aux données d'apprentissage et finit par mémoriser les modèles de données ainsi que le bruit et les fluctuations aléatoires.

Ces modèles ne parviennent pas à généraliser et à obtenir de bonnes performances dans le cas de scénarios de données inédits, ce qui va à l'encontre de l'objectif du modèle.

La variance élevée de la performance du modèle est un indicateur d'un problème de sur-ajustement.

Le temps d'apprentissage du modèle ou sa complexité architecturale peuvent être à l'origine d'un sur-ajustement du modèle. Si le modèle s'entraîne trop longtemps sur les données d'entraînement ou s'il est trop complexe, il apprend le bruit ou les informations non pertinentes dans l'ensemble de données.

Pour corriger les problèmes de sur-apprentissage, on peut utiliser une de ces techniques :

- **Cross validation** : on divise l'ensemble de données en deux parties : les données "test" et les données "train". On doit construire le modèle en utilisant l'ensemble "train". L'ensemble "test" est utilisé pour la validation en temps réel.
- **Regularisation** : Il s'agit d'une forme de régression qui régularise ou réduit les estimations des coefficients vers zéro. Cette technique décourage l'apprentissage d'un modèle plus complexe. Un modèle de régression qui utilise la technique de régularisation L1 est appelé Lasso regression et le modèle qui utilise L2 est appelé Ridge Regression. La ridge regression ajoute la "valeur au carré" du coefficient

comme terme de pénalité à la fonction de perte. Ici :

$$\sum_{i=1}^n (y_i - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

La Lasso regression ajoute la valeur absolue du coefficient comme terme de pénalité à la fonction de perte.

$$\sum_{i=1}^n (y_i - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- **EarlyStopping** : Lorsqu'on construit un apprenant avec une méthode itérative, on arrête le processus d'ajustement avant la dernière itération. Cela empêche le modèle de mémoriser l'ensemble de données.
- **Dropout** : Ignorer des neurones choisies au hasard durant l'entraînement du modèle .

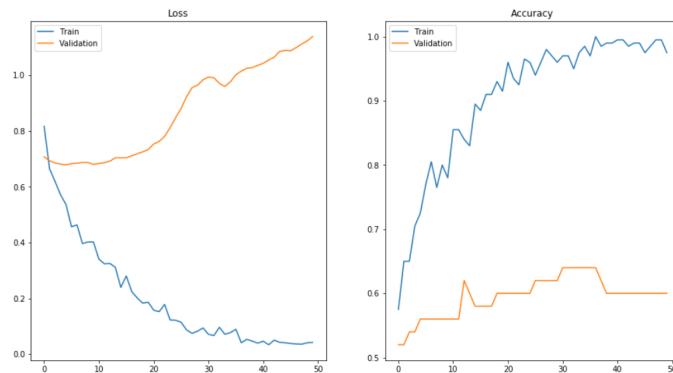


FIGURE 2.5 – Exemple d'un problème de overfitting (source : v7labs.com)

## 2.7 Conclusion

Nous avons présenté dans ce chapitre les méthodes de Machine Learning et les métriques d'évaluation de performance des modèles considérés. Dans le troisième chapitre nous nous intéressons par la collecte des données.

# Chapitre 3

## Collecte des données

### 3.1 Introduction

Ce chapitre couvre les étapes de construction de la base de données avec les techniques du web scraping. Nous commençons par définir le type de bases de données que nous utiliserons pour sauvegarder les données extraites du Facebook. Puis nous parlerons de la méthodologie pour scraper les commentaires, les réactions ainsi que les informations des prospects ayant réagi sur la page Facebook de la banque Zitouna.

### 3.2 Types des bases de données

Avant de commencer la partie du web scraping, nous devons définir dans quel type de bases de données devons-nous sauvegarder et enregistrer les données. En fait, deux types de bases de données existent : Les bases de données structurées et les bases de données non structurées.

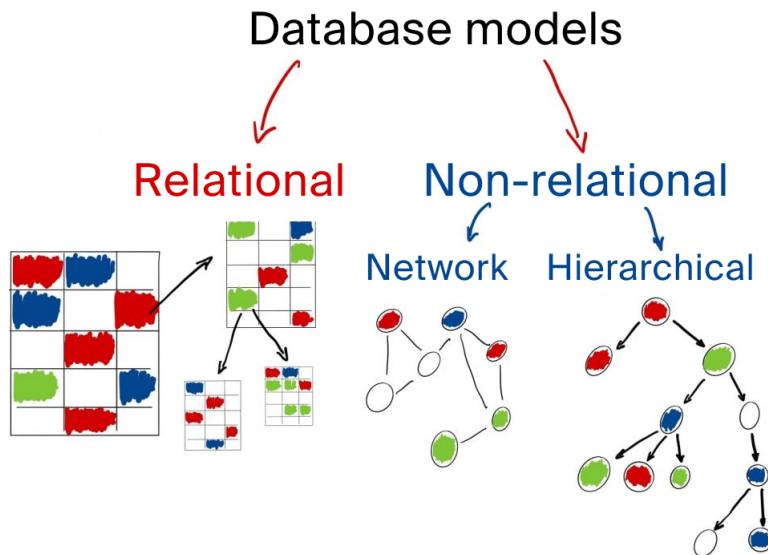


FIGURE 3.1 – Types des base de données (source :medium.com, Database models)

### 3.2.1 Base de données structurée/ Relational database

On parle d'une base de données structurée s'il s'agit des mêmes variables pour chaque observation. Ce sont des données bien traitées et formatées. Mais ce type de base de données est déconseillée pour les textes, les extraits sonores, les images, ...

Le point fort des bases de données structurées est qu'elles sont faciles à manipuler et se sont plus fluides pour les problèmes de classification.

Une base de données structurée est constituée de plusieurs tables. Chaque table ayant une dimension de  $n$  lignes et  $m$  variables.  $table.shape = (n, m)$

Une ligne représente une observation et une colonne représente une variable.

### 3.2.2 Base de données non structurée/ Non-relational database

Une base de données non structurée est désignée pour stocker les données dans leur format d'origine. Et après, lors d'une analyse de données, on peut travailler avec les données dont on a besoin. Un avantage important pour la base de données non structurés est que les données non structurées peuvent être collectées rapidement. Ce qui rend un taux d'accumulation plus rapide des données à leurs états natifs pour les exploiter selon le besoin.

Les données, dans une base de données non structurée, prennent souvent le format d'un document. Ces documents sont stockés dans un fichier de type JSON (JavaScript Object Notation).

Un format JSON est définie comme un pair '*nom* : *valeur*'. Les noms et les valeurs sont séparées par deux points ':' et deux pairs '*nom* : *valeur*' sont séparés par une virgule ','. Une donnée composée par un ensemble de pairs est encapsulé par des accolades '{}'. Par exemple le document ci dessous contient deux paires d'informations : le nom et l'âge.

```
{"name" : "Alfred","Age" : 24}
```

Une valeur d'un pair peut être présentée comme une autre donnée. Par exemple : dans le document ci dessous, on trouve que l'économie du pays est une forme d'un autre document contenant 2 paires d'informations : le PIB et le taux de chômage

```
{"Pays" : "Tunisie","Économie" : {"Pib" :"39.96 milliard","taux de chomage" :"14 %"}}
```

Un ensemble de documents représente une collection.

Une base de données non structurée est présentée d'une ou plusieurs collections.

Dans notre cas, on s'intéresse à l'extraction de commentaires du Facebook qui prennent souvent le type d'un texte. Donc, nous choisirons à sauvegarder les données dans une base de données non structurée et nous avons choisi de travailler avec une base de type MongoDB.

### 3.2.3 MongoDB

Pour ce présent projet, nous travaillerons avec mongodb qui est une plateforme pour la conception des données non structurées.



FIGURE 3.2 – MongoDB

MongoDB est un projet open source lancé en 2009 et développé pour la conception des bases de données non-SQL. Mongodb peut être utilisé sous une de ces trois éditions suivantes :

- a. MongoDB Community Server : cette édition est destinée pour les utilisateurs qui veulent travailler avec leurs propres projets et enregistrer leurs données localement sur leurs machines. Nous utiliserons cette édition de MongoDB.

b. MongoDB Enterprise Server : Cette édition est destinée pour travailler et sauvegarder les données dans les serveurs de l'entreprise.

c. MongoDB Atlas : Avec cette édition, nous pouvons enregistrer nos données sans besoins de les télécharger localement dans nos machines ou dans les serveurs de l'entreprise. De plus, nous pouvons les utiliser de n'importe quel machine. Il suffit de créer un compte sur mongoDB est de rappeler le nom de l'utilisateur et le mot de passe du compte mongo. Dans ce cas, on a la possibilité d'enregistrer les données dans les clouds comme Google Cloud, AWS et Microsoft Azure.

MongoDB dispose de pilotes pour les principaux langages de programmation et environnements de développement comme Python, Java et C#.

MongoDB nous offre une solution pour visualiser nos données enregistrées et les manipuler. C'est avec cette version qu'on va suivre les données au fur et à mesure de la collecte des données. Nous parlons ici du MongoDB Compass.

Nous avons compris les différents types de base de données et dans laquelle nous allons sauvegarder les données. Maintenant, nous passerons à la méthodologie suivie pour commencer le web scraping et l'extraction des informations de Facebook.

La banque Zitouna partage quotidiennement des publications dans sa page Facebook sur ces produits et ces services. Notre objectif est de collecter ces commentaires et ces réactions de toutes les publications. Premièrement, nous allons sauvegarder dans un fichier texte (d'extension .txt) les liens vers ces publications que nous souhaiterons collecter ces commentaires. Puis en parcourant ces publications une par une, nous construirons une base de données contenant tous les commentaires ainsi que la publication correspondante à chaque gamme de commentaires et une base de données contenant toutes les réactions avec les publications correspondantes.

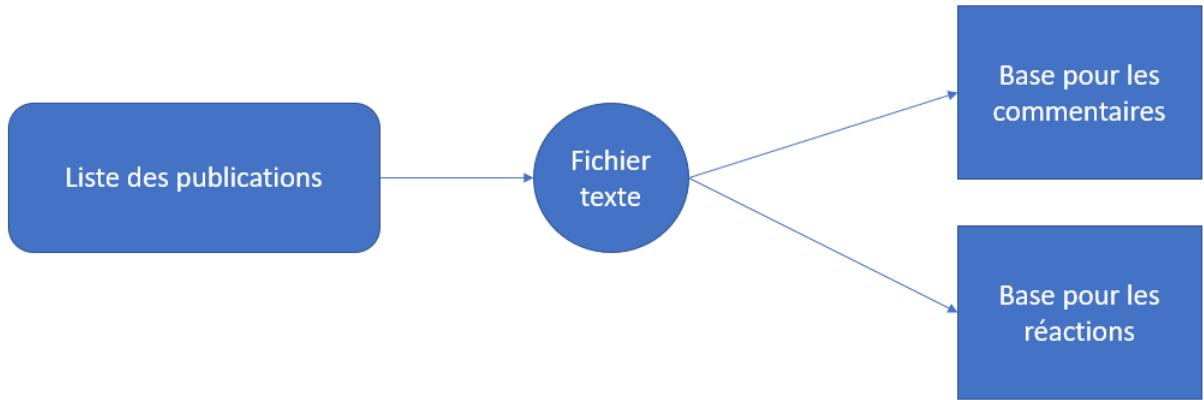


FIGURE 3.3 – Méthodologie du web scraping

Nous allons créer un processus pour le scraping des données et automatiser les procédures pour que les données seront collectées d'une manière continue et agile. C'est-à-dire de créer un algorithme pour extraire les données d'une manière automatique d'une part, où on parle de l'automatisation de la partie du web scraping et d'avoir la possibilité de suivre les données en temps réel. Pour l'automatisation de notre algorithme, nous devons initialiser et préparer notre environnement de travail. Pour ce fait, nous avons créé un bot dédié au scraping des données.

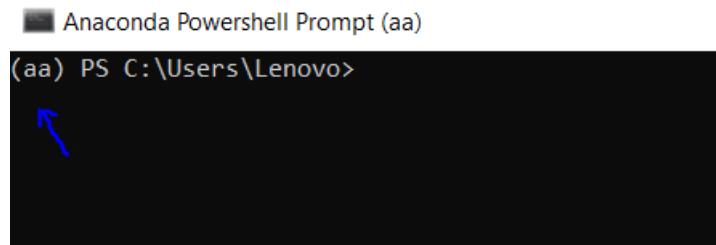
### 3.3 Préparation de l'environnement de travail

Il est important d'organiser l'environnement de travail pour éviter le plus possible les erreurs liées aux problèmes comme la corruption des fichiers, l'installation des librairies incompatibles ou qui peuvent affecter d'autres projets, ...

L'installation des librairies est faite avec pip et la préparation de l'environnement sera faite avec conda. Pip est dédié pour l'installation des librairies et des packages alors que conda est un système open source de gestion des environnements et des librairies (packages) qui fonctionne sous Windows, macOS, Linux et z/OS.

Pour créer un environnement virtuel où nous allons installer toutes les librairies nécessaires pour notre projet, on ouvre le terminal ou Anaconda Prompt et on crée un nouvel environnement par la commande : **> conda create -name myenv** avec 'myenv' est le nom de l'environnement. Puis, pour activer cet environnement, il suffit d'utiliser cette commande : **> conda activate myenv**. Si tout se passera bien, le nom de l'environnement doit être visible

dans le terminal comme la figure ci-dessous. Dans notre cas, le nom de l'environnement est : 'aa'



The screenshot shows a dark-themed terminal window titled 'Anaconda Powershell Prompt (aa)'. The command '(aa) PS C:\Users\Lenovo>' is displayed at the top. A blue arrow points to the '(aa)' prefix, indicating the active environment.

FIGURE 3.4 – Création et activation de l'environnement

Toutes les librairies seront installées dans cet environnement avec PIP. Pour vérifier que pip est bien installé ainsi que sa version : > **pip -V**. Nous avons créé notre environnement, l'étape suivante est d'installer les librairies qu'on en a besoin pour notre algorithme pour le Web Scraping.

### 3.4 Les librairies nécessaires

Nous allons utiliser toutes ces librairies tout au long le projet :

- i. Pandas : pandas est une bibliothèque destiné pour la programmation avec Python pour la manipulation et l'analyse de données. Elle offre notamment des structures de données et des opérations pour la manipulation des tableaux et des séries.
- ii. time : cette librairie fournit des fonctions liées au temps.
- iii. Selenium : C'est une librairie implémentée sous python pour exécuter des actions JavaScript via un code. Selenium supporte l'automatisation informatique 'browser automation'. C'est à dire d'automatiser des commandes pour des besoins de test par exemple afin de trouver des bugs sur les sites web. Dans notre cas, nous allons utiliser Selenium pour exécuter des commandes et des actions javascript pour extraire les informations.

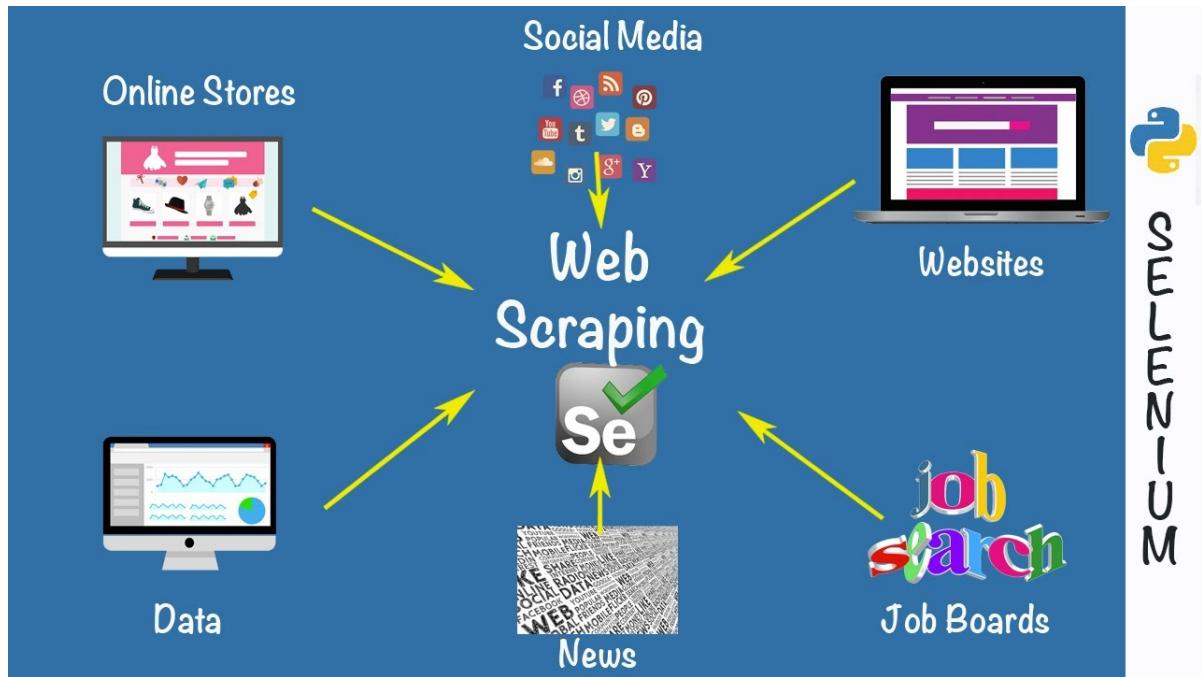


FIGURE 3.5 – Selenium pour web scraping (source : Web Scraping with Selenium | Python Programming | Automation)

Les différentes exceptions qu'on doit gérer dans notre processus au cours de la collection des données sont les suivantes :

- **NoSuchElementException** : Erreur déclenchée lorsqu'un élément est introuvable.
- **ElementClickInterceptedException** : La fonction ou l'élément Javascript Click (pour cliquer sur un bouton) ne peut pas être exécuté.
- **StaleElementReferenceException** : Lorsque l'élément à exécuter devient vicié.
- **ElementNotInteractableException** : Cette erreur est déclenchée lorsqu'un élément est présent dans la hiérarchie DOM d'un corps HTML mais les interactions avec cet élément toucheront un autre élément à cause d'un ordre de peinture dans CSS .
- **NoSuchWindowException** : Il est déclenché lorsque la fenêtre cible à changer est introuvable.

Toutes ces erreurs sont englobées dans la classe mère `selenium.common.exceptions`.

vi. Pymongo : C'est une librairie contenant des outils pour travailler avec MongoDB.

Nous pouvons installer chacune de ces librairies en utilisant la commande suivante dans le terminal ou Anaconda Prompt : > **pip install <Package>**. Par exemple pour installer pandas : > **pip install Pandas**

L'environnement de travail est créé et préparé, les librairies sont installées. Nous sommes prêts à commencer notre méthodologie pour la collection de données.

### 3.5 Méthodologie du web scraping

Pour commencer à collecter les données du Facebook, nous devons créer notre agent Scrapy et lui attribuer quelques fonctionnalités. Le Scraping sera fait en temps réel, c'est-à-dire que notre agent Scrapy se connectera au Facebook avec son propre compte comme étant un utilisateur simple et il va parcourir les publications une par une. Nous créerons un compte Facebook pour Scrapy puis nous installerons le pilote webdriver. C'est à partir de ce pilote que Scrapy peut ouvrir une fenêtre Google et se connecter au Facebook. Le pilote webdriver doit être importé à partir Selenium : » **from selenium import webdriver**

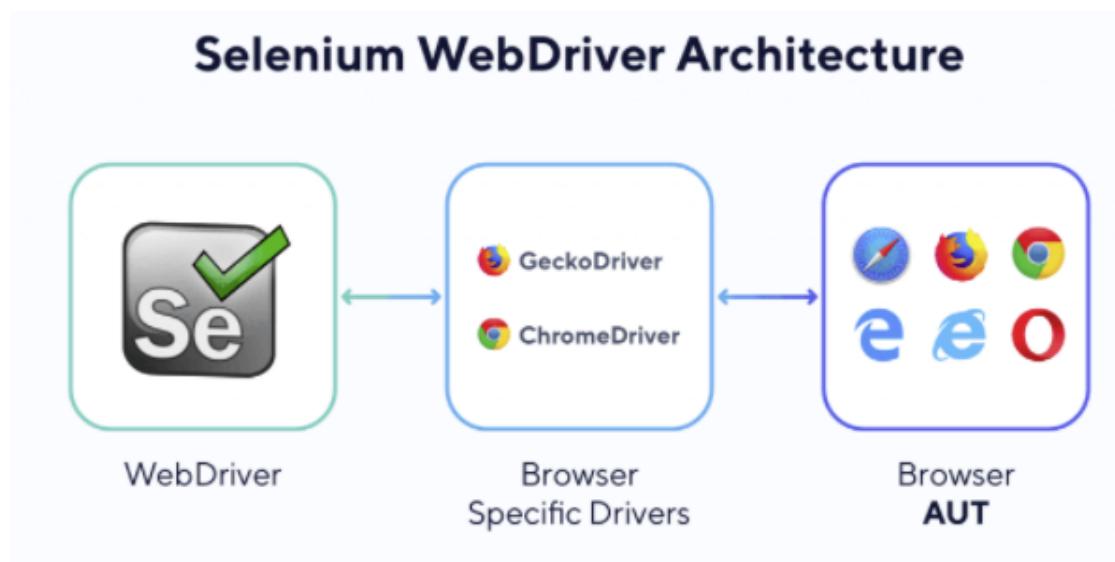


FIGURE 3.6 – WebDriver (source : testup.io)

L'étape suivante est d'implémenter quelques fonctions pour notre agent Scrapy :

- a. `connect_fb` : pour se connecter au facebook avec son propre compte en passant l'adresse mail et le mot de passe en paramètre
- b. `scroll_down` : pour défiler une page web vers le bas

c. get\_more\_links : pour enregistrer les liens vers les publications de la banque Zitouna.

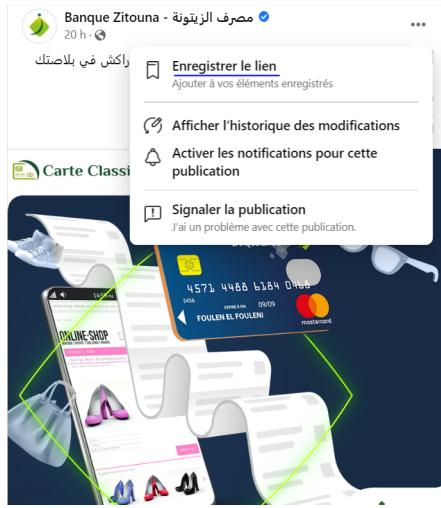


FIGURE 3.7 – Enregistrement d'une publication

d. extract\_profile\_link : pour extraire et enregistrer les variables d'une personne publiées dans son profil.

e. get\_comments : pour sauvegarder les commentaires d'une publication

f. get\_reacts : pour sauvegarder les réactions d'une publication

Du même, nous devons créer notre base de données où Scrapyy sauvegardera les données et nous devons créer un objet pour pointer sur cette base. Pour ce fait, nous devrons activer le serveur mongoDB. En activant le serveur, un terminal associé à mongoDB s'affichera comme montre la figure ci dessous :

```
C:\Program Files\MongoDB\Server\5.0\bin\mongod.exe
{"t":("date","2022-05-29T02:24:24.147+01:00"), "s":"I", "c":"CONTROL", "id":20536, "ctx":"initandlisten", "msg":"Flow Control is enabled on this deployment"}, {"t":("date","2022-05-29T02:24:24.987+01:00"), "s":"W", "c":"FTDC", "id":23718, "ctx":"initandlisten", "msg":"Failed to initialize Performance Counters for FTDC", "attr":{"error":{"code":179, "codeName":"WindowsPdhError", "errmsg":"PdhAddEnglishCounterW failed with l'objet spécifié n'a pas été trouvé sur l'ordinateur.'}}}, {"t":("date","2022-05-29T02:24:24.987+01:00"), "s":"I", "c":"FTDC", "id":20625, "ctx":"initandlisten", "msg":"Initializing full-time diagnostic data capture", "attr":{"dataDirectory":"C:/data/db/diagnostic.data"}}, {"t":("date","2022-05-29T02:24:25.025+01:00"), "s":"I", "c":"REPL", "id":6015317, "ctx":"initandlisten", "msg":"Setting new configuration state", "attr":{"newState":"ConfigReplicationDisabled", "oldState":"ConfigPreStart"}}, {"t":("date","2022-05-29T02:24:25.087+01:00"), "s":"I", "c":"NETWORK", "id":23015, "ctx":"listener", "msg":"Listening on", "attr":{"address":"127.0.0.1"}}, {"t":("date","2022-05-29T02:24:25.089+01:00"), "s":"I", "c":"NETWORK", "id":23016, "ctx":"listener", "msg":"Waiting for connections", "attr":{"port":27017, "ssl":off}}, {"t":("date","2022-05-29T02:25:24.000+01:00"), "s":"I", "c":"STORAGE", "id":22430, "ctx":"Checkpointer", "msg":"Wired Tiger message", "attr":{"message":"[1653787523:999752][1424:140715010118992], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 6, snapshot max: 6 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 50784"}}, {"t":("date","2022-05-29T02:26:24.033+01:00"), "s":"I", "c":"STORAGE", "id":22430, "ctx":"Checkpointer", "msg":"Wired Tiger message", "attr":{"message":"[1653787584:32508][1424:140715010118992], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 6, snapshot max: 6 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 50784"}}, {"t":("date","2022-05-29T02:27:24.085+01:00"), "s":"I", "c":"STORAGE", "id":22430, "ctx":"Checkpointer", "msg":"Wired Tiger message", "attr":{"message":"[1653787644:85623][1424:140715010118992], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 8, snapshot max: 8 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 50784"}}
```

FIGURE 3.8 – Serveur mongoDB

Nous avons connecté au serveur mongoDB, on crée un tunnel entre le client API<sup>1</sup> et le serveur mongod via le port ouvert par le serveur qui est 27017 dans notre cas. Ce tunnel s'établit par la fonction MongoClient : » **client = pymongo.MongoClient("mongodb://localhost :port")** Par la suite il nous reste que de créer notre première base de données à remplir ultérieurement par notre agent Scrapy la création se fait en passant le nom de la base comme attribut au client : » **DataBase = client.nom\_de\_la\_base**

Le bot Scrapy est destiné à suivre cet algorithme dans l'ordre :

- Sauvegarder le plus possible des publications dans un fichier texte.
- Parcourir ce fichier par ligne : Chaque ligne contient l'URL d'une publication.
- Enregistrer les commentaires et les réactions dans la base MongoDB.

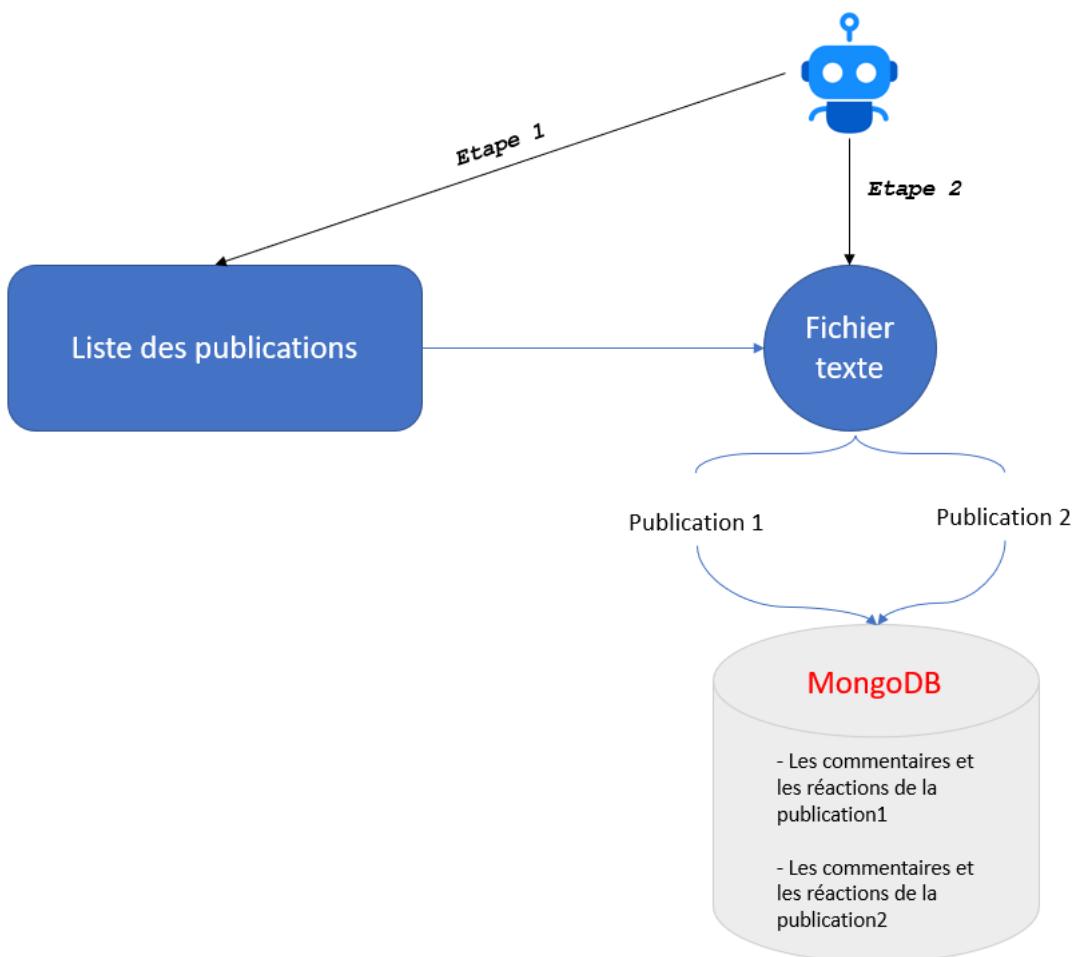


FIGURE 3.9 – Algorithme du bot Scrapy

Chaque publication prend la forme d'une collection dans la base de données créée, c'est-à-dire que chaque collection sera destinée par une seule publication. La création d'une

---

1. Application Programming Interface

collection se fait en passant le nom de la collection comme une variable ou un attribut à notre base de données. » **collection = DataBase['publication\_1']** . Rappelons qu'une collection est composées par des documents, les documents de chaque collection seront remplis dans l'ordre suivant : le premier document prend le texte de la publication correspondante, à partir du deuxième document, on passera dans chaque document le nom du prospect, son commentaire et ses données. Puis nous enregistrons les réactions dans les derniers documents. Par exemple dans le dernier document, on trouve le nom du prospect, sa réaction et ses informations partagées sur son profil.

FIGURE 3.10 – Vue d'ensemble sur mongodb

### 3.6 Exportation des données dans un fichier excel

A partir de notre base de données dans mongoDB, nous exporterons deux tableaux sous des fichier excel, une contenant les commentaires pour l'étude NLP, et l'autre contenant la classe du chaque commentaire déduit par la modélisation NLP, l'historique des réactions du chaque personne qui a laissé un commentaire, et d'autres variables comme lieu de résidence, l'âge, et le genre.

### 3.6.1 Creation d'une base de donnees pour l'tude NLP

Le premier tableau que nous devons exporter du mongoDB doit contenir une variable de tous les commentaires scrapes du page Facebook de la banque Zitouna.

Chaque gamme de commentaires relative a la publication i se trouve dans la collection *publication\_i*. Rappelons que le nom de notre base de donnees est *facebook*. Pour manipuler les donnees de cette collection en particulier, nous devons se pointer sur cette collection par la commande : » **facebook[publication\_i]**.

Nous sommes diriges vers la collection *publication\_i*. Un curseur est initialise dans le premier document. Ce curseur parcourt toute la collection document par document. Donc nous devons specifier les donnees extraire de chaque document.

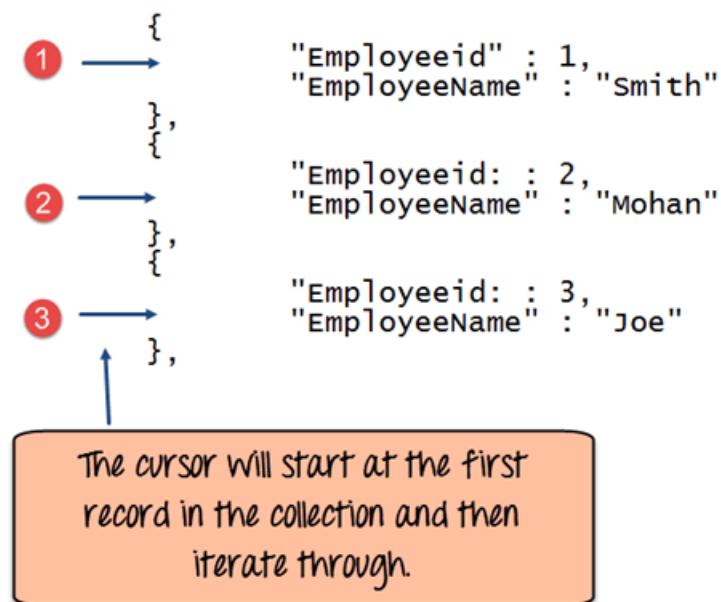


FIGURE 3.11 – Iteration des documents par un curseur (source : guru99.com)

Tous les documents d'une collection ayant leurs propres id sont generes alatoirement par le serveur mongoDB. la deuxieme tape est donc de filtrer ces documents pour ne prendre que les commentaires et les noms des prospects. On peut filtrer les documents du mongoDB par la fonction »**find(projection = project)** , **project = { '\_id : 0'}**». Dans l'objet project on a specifi que la valeur de l'id doit tre invisible pour le curseur. Dans ce cas, nous allons retenir toutes les variables sauf l'id. Or nous ne voudrons retenir que les commentaires ainsi que les noms, Il faut alors mettre toutes les variables invisibles pour le curseur sauf la variable commentaire et la variable nom. »**find(projection = project)** , **project = { '\_id : 0', 'commentaire' : 1, 'nom' : 1 }**».

Après avoir filtré notre base, certains documents deviendront vides et semblant comme la figure ci dessous.

The screenshot shows a MongoDB query interface with the following configuration:

- Documents** tab selected.
- FILTER**: `{ field: 'value' }`
- PROJECT**: `{'_id':0, 'commentaire' : 1, 'name' : 1}`
- SORT**: `{ field: -1 } or [[ 'field', -1 ]]`
- COLLATION**: `{ Locale: 'simple' }`
- OPTIONS**: `MAX TIME MS: 60000`, `SKIP: 0`, `LIMIT: 0`
- FIND** and **RESET** buttons.
- Displaying documents 1 - 20 of 81.

The results list contains four documents, each with a collapsed icon. The first document's expanded view shows:

```

{
  "name": "████████",
  "commentaire": "تحب نشري سيارة"
}

```

The other three documents are collapsed.

FIGURE 3.12 – Données filtrées

En itérant ces documents, nous ne retiendrons que ceux ayant la variable commentaire différente à la valeur nulle, i.e que la variable commentaire existe. Supposant que la base filtrée est enregistrée dans result : `result = facebook1[publication].find(projection = project)`, `project = {'_id' : 0, 'commentaire' : 1, 'nom' : 1 }`, on pointe le curseur dans le  $j^{eme}$  document : `result[j]` puis nous vérifions si le clé 'commentaire' existe dans ce  $j^{eme}$  document ou non : » `'commentaire' in list(result[index].keys())`. Si c'est le cas, on enregistre ce commentaire dans un tableau de type dataframe. Ajoutons une variable publication contenant le texte de la publication où on a partagé chaque commentaire. L'exportation du dataframe vers un fichier excel se fait par la fonction :

» `DataFrame.to_excel('non_du_fichier.xlsx')`.

**Étiquetage des données** : Après avoir construit la base de données, il ne reste qu'étiquer les commentaires selon leurs classes d'appartenances :

a. S'il s'agit d'un commentaire où le prospect veut exprimer ses sentiments sans laisser un message concret, par exemple : Merci beaucoup, la tunisie va gagner son match aujourd'hui,

... Ce type de message est labélisé par 'autre'.

b. Si le prospect veut acheter une maison, une voiture, ou qu'il veut voyager, ou de faire d'autre achats comme les achats des biens de consommation, de financer des études universitaires ou d'avoir recours à des soins médicaux. Dans ce cas, notre prospect est particulier et ce type de commentaires est labélisé par 'particulier'.

c. Si le prospect veut s'investir dans un projet, financer des équipements professionnels pour un projet, de faire des financements d'acquisition de matériels de transport, de biens immobiliers ou terrain à usage commercial et industriel ou bien de faire des financements d'aquisition des équipements médicaux, Ce prospect est alors désigné d'être un professionnel et son commentaire est labélisé par 'professionnelle'

d. Si le prospect veut savoir des informations sur un type des cartes, savoir combien peut on accéder à la banque à distance ou bien de bénéficier d'un transfert d'argent, ce pprospect est alors un demendeur d'un service et son commentaire est à labélisé par 'service'.

Nous pouvons nous retrouver avec des commentaires dont on ne peut pas les classer sans avoir une idée sur la publication. Par exemple : une publication d'un offre de voiture et un prospect qui a répondu dans son commentaire par : Comment peut faire ? dans ce cas nou allons classer ce commentaire par particulier parceque la publication est dirigée vers les particuliers.

### **3.6.2 Crédation d'une base de données pour la segmentation des prospects**

Dans cette partie, nous exportons une base de données du mongodb contenant les variables suivantes : le nom du prospect, son dernier commentaire, ses réactions qu'il a réagit dans facebook ainsi que ces données qu'il a partagées dans son profil Facebook. La variable commentaire sera remplacée par sa classe d'appartenance selon l'étude NLP que nous ferons dans le chapitre suivant.

Cette base de données dépend du résultat obtenu suite à la classification des commentaires. Nous allons définir les différentes variables de cette base de données ainsi que l'étiquetage des données dans le quatrième chapitre

### **3.7 Conclusion**

Dans ce chapitre, nous avons exploré la méthodologie suivie pour extraire les données que nous avons besoin de Facebook. Nous discuterons dans le chapitre suivant les méthodes abordées pour la classification des commentaires et la segmentation des prospects.

# Chapitre 4

## Classification des commentaires et segmentation des prospects

### 4.1 Introduction

Nous avons défini les deux bases de données que nous allons travailler avec dans ce chapitre. La première partie de ce chapitre sera consacrée pour la classification des commentaires avec les méthodes NLP. Nous utiliserons les résultats obtenus de cette partie pour segmenter les prospects du Facebook selon leurs besoins et de prédire ceux qui peuvent être de nouveaux clients pour la banque Zitouna.

### 4.2 Traitement du langage propre : NLP

Ce paragraphe est destiné pour étudier les commentaires avec les techniques du NLP. Deux étapes existent pour le traitement automatique du langage propre ou NLP : le pré-traitement des textes et la partie modélisation/Deep Learning.

#### 4.2.1 Pré-traitement des données

Les données que nous avons collectées pour le NLP sont des commentaires, donc ce sont des textes. Le pré-traitement et le nettoyage de données est une transformation des textes pour mettre ces données interprétables pour la partie modélisation. Nous commençons par présenter les librairies que nous utiliserons.

### Les librairies utilisées :

i. re, Regular Expression : Une expression régulière définit un ensemble de chaînes qui lui correspondent. Les fonctions de ce module nous permettent de vérifier si une chaîne particulière correspond à une expression régulière donnée

ii. NLTK, Natural Language Toolkit : Cette librairie est développée pour la création de programmes Python destinés à travailler avec des données sur le langage naturel humain. Elle fournit une suite de bibliothèques de traitement de texte pour la classification, la tokénisation, et l'analyse syntaxique.

iii. sklearn : C'est une bibliothèque d'apprentissage automatique pour le langage Python. Elle présente divers algorithmes de classification, de régression et de regroupement.

iv. wordcloud : Une librairie pour générer les nuages des mots

v. `plotly` : Plotly fournit des outils graphiques, analytiques et statistiques en ligne pour particuliers et professionnels, ainsi que des bibliothèques de graphiques scientifiques pour Python.

## Nettoyage des textes

La table des données qu'on va travailler avec est représentée dans la figure ci dessous :

Entrée [9]:	1 df.head()																								
Out[9]:	<table><thead><tr><th>y</th><th>publication</th><th>commentaire</th><th>name</th></tr></thead><tbody><tr><td>0</td><td>particulier</td><td>1 يتصدى إداريًا لأخيب</td><td>[REDACTED]</td></tr><tr><td>1</td><td>service</td><td>1 salam alaykom enheb nhot flousi fi bank ezitou...</td><td>[REDACTED]</td></tr><tr><td>2</td><td>service</td><td>1 salam alaykom belahi el moda5arat toksed bohom...</td><td>[REDACTED]</td></tr><tr><td>3</td><td>service</td><td>1 Banque Zitouna - مصرف الزيتونة - bounjour toksed...</td><td>[REDACTED]</td></tr><tr><td>4</td><td>service</td><td>1 Mohamed Hani va sa7bi yzidouk 1 pour cent ken ...</td><td>[REDACTED]</td></tr></tbody></table>	y	publication	commentaire	name	0	particulier	1 يتصدى إداريًا لأخيب	[REDACTED]	1	service	1 salam alaykom enheb nhot flousi fi bank ezitou...	[REDACTED]	2	service	1 salam alaykom belahi el moda5arat toksed bohom...	[REDACTED]	3	service	1 Banque Zitouna - مصرف الزيتونة - bounjour toksed...	[REDACTED]	4	service	1 Mohamed Hani va sa7bi yzidouk 1 pour cent ken ...	[REDACTED]
y	publication	commentaire	name																						
0	particulier	1 يتصدى إداريًا لأخيب	[REDACTED]																						
1	service	1 salam alaykom enheb nhot flousi fi bank ezitou...	[REDACTED]																						
2	service	1 salam alaykom belahi el moda5arat toksed bohom...	[REDACTED]																						
3	service	1 Banque Zitouna - مصرف الزيتونة - bounjour toksed...	[REDACTED]																						
4	service	1 Mohamed Hani va sa7bi yzidouk 1 pour cent ken ...	[REDACTED]																						

FIGURE 4.1 – Table de données pour le traitement du NLP

Nous nous intéressons par la variable commentaire afin de les nettoyer et de les rendre plus utiles pour les modèles Deep Learning.

- Nous commençons par transformer tous les commentaires en minuscules par la fonction lower

```
» for i in range(df.shape[0]):
```

```
...     df.commentaire[i] = df.commentaire[i].lower()
```

- Puis, nous traiterons les caractères spéciaux comme :

- \n : pour les retours à la ligne.

- \t : pour les tabulations.

- \r : pour les retours au début de ligne.

De plus, nous voudrons supprimer tous les chiffres, les parenthèses, les guillemets, les accolades, les points d'interrogations et d'exclamations, ... Pour chercher ces caractères et les supprimer, nous utiliserons la fonction sub() de la librairie re<sup>1</sup>, l'idée est qu'à chaque fois on trouve un de ces critères, nous les remplaçons par une chaîne vide.

```
1 for k in df.commentaire:  
2     k = re.sub(r'[0-9]', '', k) #les numeros de 0 à 9  
3     k = re.sub(r'"<>\.==*()_"', ' ', k) #suppression de ces caractères:  
4         "<>\.==*()_"  
5     k = re.sub(r'[\n\t\r]', ' ', k) #suppression des caractères suivantes: \n{  
    retour à la ligne}, \t{tabulation.}, \r{pour retourner au début de la  
ligne}  
6     k = re.sub(r'[?;:/!]', ' ', k) #suppression des points d'interrogation, d'  
exclamations, ..., ? ; :/ !
```

Listing 4.1 – Supprimer les caractères spéciaux

-Nous voudrons de même de supprimer les emojis utilisés dans les commentaires. Les emojis sont présentés par des Unicodes. Les unicodes sont des représentations d'extension ASCII qui permettent de représenter beaucoup d'autres caractères. Par exemple : \U0001F600-\U0001F64F pour les emojis SMILES. Pour supprimer ces emojis qui prennent le format des unicodes, nous utiliserons la librairie re. re doit compiler d'abord ces unicodes pour qu'il puisse les reconnaître dans un texte par la fonction compile. Puis, nous les remplaçons par une chaîne vide.

```
1 def remove_emoji(string):
```

1. Regular expression

```

2 emoji_pattern = re.compile("["
3     u"\U00010000-\U0001ffff"
4     u"\U0001F600-\U0001F64F" # émoticones sourires
5     u"\U0001F300-\U0001F5FF" # symboles et pictogrammes
6     u"\U0001F680-\U0001F6FF" # symboles de transport et de carte
7     u"\U0001F1E0-\U0001F1FF" # les drapeaux
8     "]+", flags=re.UNICODE)
9
10 return emoji_pattern.sub(r'', string)

```

Listing 4.2 – supprimer les emojis

- L'étape suivante est de supprimer les mots d'arrêts (Stop words). Les mots d'arrêts sont des mots qui sont très utilisés dans nos phrases sans avoir une importance dans le sens des phrases. Par exemple : de, duquelle, donc, serait, et, à, au, la, les, ... Nous utiliserons la librairie NLTK pour télécharger les mots d'arrêts en Arabe, en Francais et en Anglais.

» `nltk.download()`.

Nous créerons une liste contenant tous les mots d'arrêts en arabe, en français et en anglais. Pour importer ces mots d'arrêts du librairie NLTK , on importe stopwords, une sous librairie du NLTK, et on utilise la fonction words().

```

1 from nltk.corpus import stopwords
2 stop_words_fr_ar_eng = stopwords.words('french')
3 for i in stopwords.words('arabic'):
4     stop_words_fr_ar_eng.append(i)
5 for i in stopwords.words('english'):
6     stop_words_fr_ar_eng.append(i)

```

Listing 4.3 – Importer les mots d'arrêts

Les commentaires du Facebook sont écrits en TUNIZI : la langue arabe tunisienne (appelée aussi Derja). Il est indispensable de créer un dictionnaire de vocabulaire contenant les mots d'arrêts utilisés les commentaires pour les supprimer. Finalement, la liste que nous devons créer doit contenir les mots d'arrêts en Arabe, en Français , En anglais et en TUNIZI

- On trouve que parfois une lettre peut être répétée plusieurs fois d'une manière consécutives dans un mot. Par exemple : au lieu d'écrire projet, on écrit projeeeeeeeet. dans ce cas, nous effectuerons une transformation de ces types des mots. Si une lettre est écrite n fois avec  $n > 2$  consécutivement, nous laissons que 2 lettres et nous supprimons les autres  $n-2$  lettres.

-Stemming : Pour transformer les mots à leurs racines, nous utiliserons la fonction stem(). par exemple : les mots transformation et transformations seront être transformés vers un seul mot qui est transform. nous réduisons alors le nombre de mots(de 2 mots à un seul mot)

```

1 from nltk.stem.snowball import FrenchStemmer, ArabicStemmer, EnglishStemmer
2 for i in df.commentaire:
3     text = i.split()
4     stemming = []
5     for j in text:
6         x = EnglishStemmer().stem(j)
7         x = ArabicStemmer().stem(x)
8         x = FrenchStemmer().stem(x)
9     stemming.append(x)

```

Listing 4.4 – Transformer les mots à leurs origines

La table finale après le nettoyage des commentaires :

	y	publication	commentaire	name
0	particulier	1	تقدير dar حبيب شر	[REDACTED]
1	service	1	salam alaykom enhib nhot flous bank e...chn...	[REDACTED]
2	service	1	salam alaykom belah el modaarat tok bohem kont...	[REDACTED]
3	service	1	مرحبا زين مصطفى زينون bounjour tok kont blok w...	[REDACTED]
4	service	1	sab yzidouk cent ken tar hak hii kai	[REDACTED]

FIGURE 4.2 – Table de données après pré-traitement

Les données sont bien traitées et nettoyées, nous aborderons la visualisation des données (Data Visualisation) pour avoir une idée sur les mots les plus utilisés dans nos commentaires et dans chacune des 4 classes.

#### 4.2.2 Visualisation des données

Pour la visualisation des commentaires, nous devons couper chaque commentaire mot par mot et visualiser la fréquence de chaque mot. On parle ici de la Tokenisation. Nous découpons les commentaires par la fonction CountVectorizer(). Cette fonction Convertit une collection de documents textuels en une matrice de mots/tokens..

$$\begin{pmatrix} 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 1 & 3 & 0 & 1 & 1 \end{pmatrix}$$

Chaque valeur de cette matrice par exemple présente le nombre d'apparition du token/mot de la colonne j dans le corpus/texte i.

La fonction CountVectorizer est développée sous la librairie Sklearn/Scikit-learn.

```
1 from sklearn.feature_extraction.text import CountVectorizer  
2 vect = CountVectorizer()  
3 cv = vect.fit_transform(df.commentaire)  
4 #convertir la matrice générée par countVectorizer en dataframe  
5 word_counter = pd.DataFrame(  
6     cv.toarray(),  
7     columns = vect.get_feature_names())
```

Listing 4.5 – Transformer les mots à leurs origines

word\_counter est un DataFrame contenant tous les mots en colonnes et les commentaires en ligne. Nous ajouterons une variable y qui est la classe des commentaires. la variable y ayant 4 modalités : Particulier, Professionnelle, Service, Autre.

La table word counter ayant 7215 lignes et 9608 colonnes.

FIGURE 4.3 – Tokenization des commentaires

Nous utiliserons plotly pour générer un histogramme sur les mots les plus utilisés.

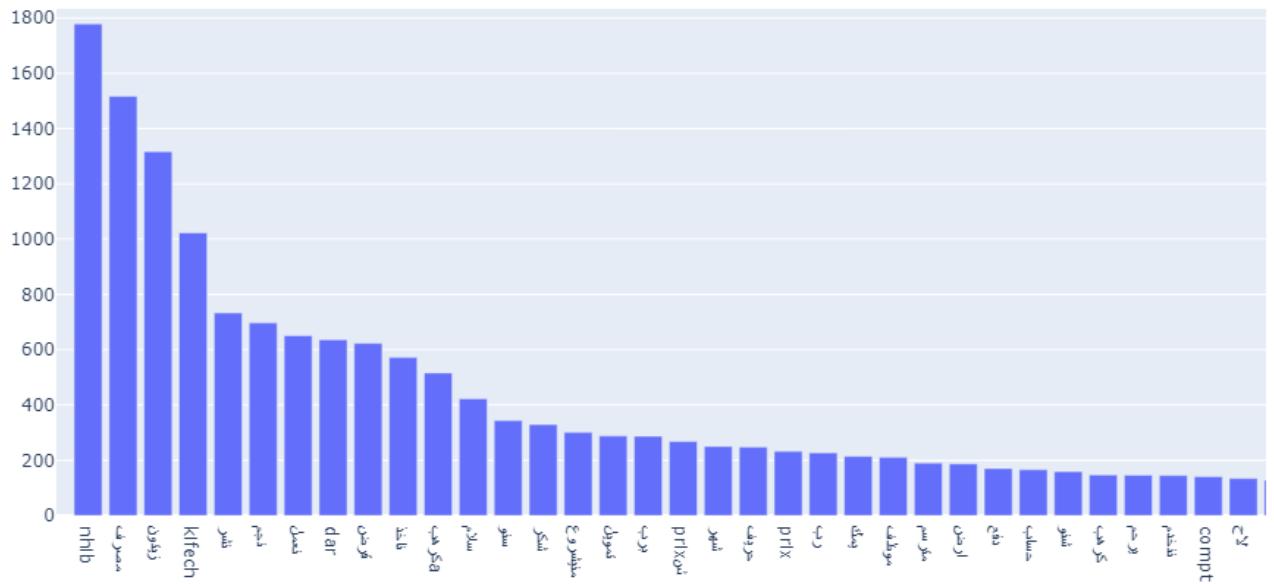


FIGURE 4.4 – Bar plot sur la fréquence des tokens

On observe que le mot le plus utilisé est : nhib (je veux) de 1780 fois puis les mots banque et zitouna (écrits en arabe) de 1500 fois et 1300 fois respectivement. Les mots les plus utilisés ont un sens et ne peuvent pas être considérés comme des mots d'arrêts. Ce bar plot est une sorte de vérification sur les traitements que nous avons déjà fait pour les commentaires.

Pour avoir une idée sur le type des mots les plus utilisés dans chacune des classes des commentaires, nous générerons un nuage de mots (wordcloud) en utilisant la librairie wordcloudFa parce que cette librairie supporte les mots en arabe.

- Nuage de mots pour les commentaires de classe autre :



FIGURE 4.5 – Nuage de mots pour la classe autre

On observe la dominance des mots banque, zitoun, quoi (écrits en arabe dans le word-cloud) et les mots bravo, nhib, ... La majorité de ces mots expriment un avis, un sentiment, un souhait. D'où leurs appartences aux commentaires de la classe "autre"

- Nuage de mots pour les commentaires de classe particulier :



FIGURE 4.6 – Nuage de mots pour la classe "particulier"

On observe la dominance des mots voiture, zitoun, crédit (écrits en arabe dans le wordcloud) et les mots kifech, nhib, dar, ... La majorité de ces mots expriment un besoin des produits particuliers comme l'achat d'une voiture ou d'un maison. D'où leurs appartances aux commentaires de classe particulier.

- Nuage de mots pour les commentaires de classe professionnelle :



FIGURE 4.7 – Nuage de mots pour la classe professionnelle

On observe la dominance des mots je fais, terre, crédit, projet (écrits en arabe dans le wordcloud) et les mots nhib, kifech, ... La majorité de ces mots expriment un besoin de financement, d'investissement. D'où leurs appartences aux commentaires de classe professionnelle.

- Nuage de mots pour les commentaires de classe service :

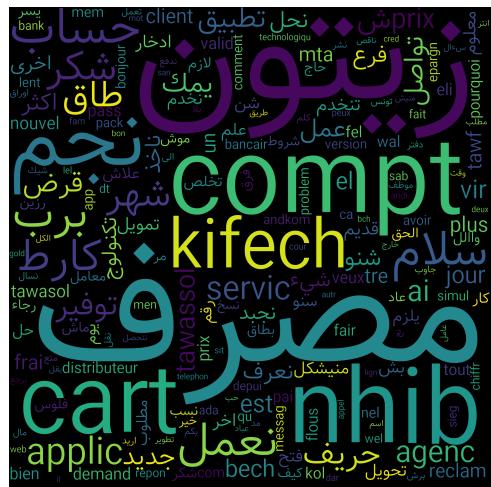


FIGURE 4.8 – Nuage de mots pour la classe service

On observe la dominance des mots zitoun (écrit en arabe dans le wordcloud) et les mots compt, kifech, cart, applic, agence ... La majorité de ces mots expriment une demande d'information sur un service particulier. D'où leurs appartéances aux commentaires de classe service.

## NLP : Partie modélisation

Cette partie est consacrée pour l'implémentation d'un modèle d'apprentissage pour notre problématique NLP afin de faire la classification des commentaires en 4 classes.

On prépare notre base de données avec laquelle on va travailler dans la deuxième partie du NLP : la modélisation. Certains commentaires ne sont composés que par un seul mot "kifech". Ce mot se réfère à quoi ? - à la publication. Pour ces genres de commentaires, il est indispensable de trouver une solution pour inclure les textes des publications dans les commentaires pour la classification.

publication	commentaire
particulier	je veux acheter une voiture
autre	comment

publication	commentaire
particulier	particulier je veux acheter une voiture
autre	autre comment

TABLE 4.1 – Préparation des commentaires pour la modélisation NLP

Nous travaillerons dans cette partie avec les algorithmes d'apprentissage approfondies et nous utiliserons les couches LSTM<sup>2</sup> et les couches RNN<sup>3</sup>

a. L'architecture derrière les couches LSTM :

Une couche LSTM est représentée par des portes ou gates. Chaque porte ayant son rôle de sauvegarder quelques informations, d'ignorer les informations inutiles... il y en a dans une couche LSTM 4 portes :

1. forget gate, porte d'oubli : décider quelles informations doit-on jeter de l'état de la cellule. Cette décision est prise par une couche sigmoïde appelée "forget gate layer". Elle examine  $h_{t-1}$  et  $x_t$ , et produit un nombre entre 0 : "se débarrasser complètement de ceci" et 1 : "garder complètement ceci" pour chaque nombre dans l'état cellulaire  $C_{t-1}$ . Soit la fonction sigmoid :  $\sigma(x) = \frac{1}{1+\exp{-x}}$ . La fonction du forget gate est :  $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$  avec W est le vecteur poids alors que b est le vecteur biais.

2. input gate layer : décider quelles nouvelles informations devons-nous stocker dans l'état de la cellule. Ce processus comporte deux parties. Tout d'abord, une couche sigmoïde

---

2. Long Short Term Memory  
3. recurrent neural network

appelée "input gate layer" décide des valeurs à mettre à jour.  $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ . Ensuite, une couche tanh crée un vecteur de nouvelles valeurs candidates,  $\tilde{C}_t$ , qui pourraient être ajoutées à l'état.  $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$ . Dans l'étape suivante, nous allons combiner ces deux couches pour créer une mise à jour de l'état.

3. il est temps de mettre à jour l'ancien état de la cellule,  $C_{t-1}$ , dans le nouvel état de la cellule  $C_t$ .

Nous multiplions l'ancien état par  $f_t$ , en oubliant les choses que nous avons décidé d'oublier plus tôt. Puis nous l'ajoutons  $i_t \cdot \tilde{C}_t$ . Ce sont les nouvelles valeurs candidates, mises à l'échelle par combien nous avons décidé de mettre à jour chaque valeur d'état.

4. output gate : Enfin, nous devons décider de ce que nous allons produire. Cette sortie sera basée sur l'état de nos cellules, mais sera une version filtrée. D'abord, nous exécutons une couche sigmoïde qui décide des parties de l'état de la cellule que nous allons sortir.  $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ . Ensuite, nous faisons passer l'état de la cellule par tanh (pour que les valeurs soient comprises entre -1 et 1) et nous le multiplions par la sortie de la porte sigmoïde, sigmoid gate, afin de ne sortir que les parties que nous avons décidées.

$$h_t = o_t * \tanh(C_t)$$

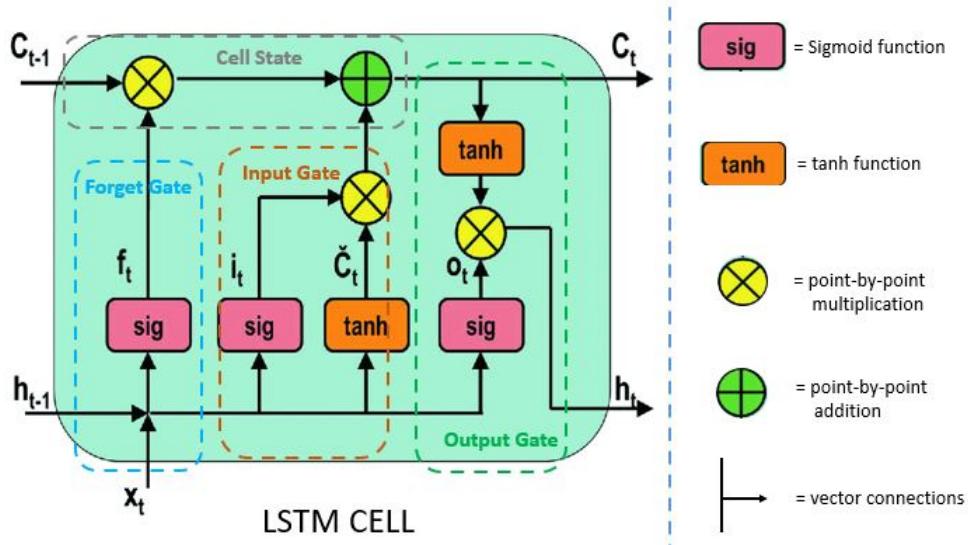


FIGURE 4.9 – Architecture d'une couche LSTM (source : pluralsight.com)

## b. Deep Learning

Maintenant, nous sommes prêts pour commencer à implémenter notre modèle Deep Learning pour la classification de commentaires.

Le vecteur d'entrée, the input, doit être formater comme :

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1m} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nm} \end{bmatrix}$$

où  $x_i$  représente le  $i^{eme}$  commentaire et  $x_{ij}$  représente le  $j^{eme}$  mot utilisé dans ce  $i^{eme}$  commentaire.

Nous pouvons générer cette transformation avec la fonction tokenizer de la librairie Keras.

La fonction Tokenizer implémentée sous Keras permet de diviser des textes en mots ou plus pratiquement en tokens, chaque mot est transformé par son indice dans son texte, les tokens seront réorganiser et réarranger selon leurs fréquences d'apparition dans le corpus. Par exemple on suppose que le mot le plus utilisé est "token" qui est utilisé 5 fois dans les documents et la première apparition de ce mot est dans l'indice 5. Donc au lieu d'affecter l'entier 5 à ce mot, on lui affecte l'indice 1 puisque c'est le mot le plus utilisé.

Exemple : texte1 = "today is my day", texte2 = "she is winning her game this day"

On applique la fonction *pad\_sequences* pour ajouter des 0 au début de chaque texte seulement pour que les vecteurs soient de même taille.

```

1 #Exemple
2
3 from keras.preprocessing.text import Tokenizer
4 from keras.preprocessing.sequence import pad_sequences
5
6 texte = ["today is my day", "she is winning her game this day"]
7
8 tokenizer = Tokenizer(split=' ')
9 tokenizer.fit_on_texts(texte)
10 X = tokenizer.texts_to_sequences(texte)
11 X = pad_sequences(X)
```

Listing 4.6 – Exemple de tokenisation des corpus

```
{'is': 1, 'day': 2, 'today': 3, 'my': 4, 'she': 5, 'winning': 6, 'her': 7, 'game': 8, 'this': 9}
array([[0, 0, 0, 3, 1, 4, 2],
       [5, 1, 6, 7, 8, 9, 2]])
```

FIGURE 4.10 – Exemple de tokenisation

Dans notre cas, nous nous limitons uniquement aux 1000 tokens les plus utilisés,  $max\_features = 1000$ .

```
1 max_features = 1000
2 tokenizer = Tokenizer(max_features, split=' ')
3 tokenizer.fit_on_texts(data['commentaire'].values)
4 X = tokenizer.texts_to_sequences(data['commentaire'].values) #today[5] is
   [856] my[300] day[670] -> [0 0 ... 0 0 0 5 856 300]
5 X = pad_sequences(X)
```

Listing 4.7 – Tokenisation des commentaires

Nous avons créé notre matrice d’entrée qui est de dimension (6016,160).

Pour suivre les essais que nous allons faire pour optimiser la performance du modèle tout en visualisant à chaque fois les paramètres ainsi que les métriques, nous utiliserons la méthode Mlflow. La commande : » **mlflow.tensorflow.autolog()** permet d’enregistrer, pour chaque modèle implémenté, les paramètres comme le nombre d’époques, le batch\_size, les callbacks comme EarlyStopping, la validation\_split, ... et les enregistrer. Et pour visualiser toutes ces données, on utilise cette commande dans le terminal > **mlflow ui**. un lien sera affiché suite à cette commande qui nous conduit vers l’interface mlflow. Dans cette partie, nous allons représenter seulement les modèles les plus performants que nous avons réussi à implémenter. Dans la figure ci dessous un exemple de l’interface mlflow.

The screenshot shows the mlflow interface with a search query "metrics.rmse < 1 and params.model = 'tree'" applied. There are 23 matching runs listed. The 'Metrics' tab is selected, showing detailed performance metrics and parameters for each run.

FIGURE 4.11 – Interface mlflow

Nous allons présenter le modèle le plus performant que nous avons obtenu.

Les couches LSTM exigent que l'entrée doit être une matrice de 3 dimensions(n lignes, m colonnes, l profondeur), c'est pour cette raison que nous utiliserons d'abord les couches d'intégrations, Embedding Layer. Grace à cette couche, chaque mot sera représenté par un vecteur d'entier. D'où la matrice X sera de dimension 3 puisque chaque valeur  $x_{ij}$  sera représentée par un vecteur d'entiers. La matrice résultante par la couche d'Embedding sera l'entrée dans les couches LSTM. Nous ajouterons une couche LSTM de 50 unités. pour éviter le problème de overfitting nous avons choisi de :

- Ignorer 30% des neurones
- Utiliser les fonctions l1 et l2 de régularisation : recurrent\_regularizer='l1\_l2'
- Utiliser la technique EarlyStopping.

La fonction d'activation pour cette couche est la fonction tanh :  $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Puis on ajoute une couche de sortie de 4 neurones pour prédire la classe de chaque commentaire. La fonction d'activation utilisée dans cette couche est la fonction sigmoid :  $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$ . Finalement nous utiliserons la fonction loss  $MSE = \text{mean_squared_error} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$  et la fonction d'optimisation : Adam.

Model: "sequential_18"		
Layer (type)	Output Shape	Param #
embedding_18 (Embedding)	(None, 160, 30)	150030
lstm_18 (LSTM)	(None, 50)	16200
dense_18 (Dense)	(None, 4)	204

Total params:	166,434
Trainable params:	166,434
Non-trainable params:	0

FIGURE 4.12 – Modèle 1 : NLP avec les couches LSTM

L'algorithme s'est arrêté au 18<sup>eme</sup> epoch et a montré un accuracy sur la partie train de 84% et sur la partie valid. de 75% et un loss sur la partie train de 7.6% et sur la partie valid. de 10%

Name	Value
accuracy ↗	0.843
loss ↗	0.076
stopped_epoch ↗	18
val_accuracy ↗	0.752
val_loss ↗	0.103

FIGURE 4.13 – Modèle 1 : Les métriques obtenues avec les couches LSTM

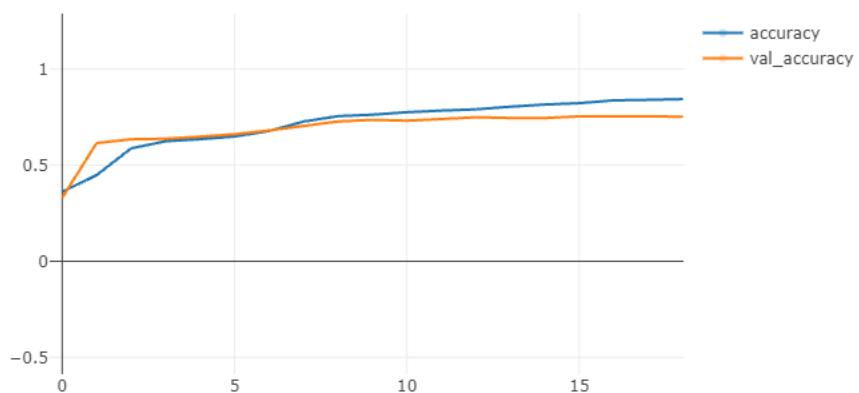


FIGURE 4.14 – Modèle 1 : Accuracy vs val accuracy

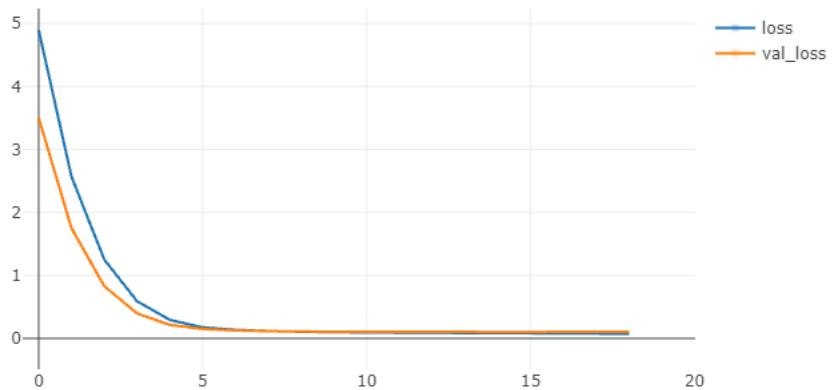


FIGURE 4.15 – Modèle 1 : Loss vs val loss

En testant cet algorithme sur la partie test de notre base par la fonction evaluate(), nous trouverons une valeur pour la précision de 74% et une valeur de loss de 10%. C'est à dire que notre modèle peut fonctionner sur d'autres données.

```
» model.evaluate(X_test,Y_test)
```

```
38/38 [=====] - 1s 28ms/step - loss : 0.1079 -
accuracy : 0.7417
```

Finalement, nous évaluerons notre modèle par un tableau de rapport résumant les métriques pour chaque classe.

	precision	recall	f1-score	support
0	0.76	0.83	0.80	366
1	0.75	0.81	0.78	488
2	0.70	0.61	0.65	263
3	0.64	0.34	0.45	87
accuracy			0.74	1204
macro avg	0.71	0.65	0.67	1204
weighted avg	0.74	0.74	0.73	1204

TABLE 4.2 – Modèle 1 : Performances obtenues par classe

En étudiant ce tableau de rapport pour la classe 0 : la précision est de 76%, un recall de 83% et un f-score de 80%. Ce qui montre un modèle performant et prédit bien d'autres données.

Maintenant, nous implémentons un autre modèle avec les couches RNN simples.

RNN ou Recurrent Neural Networks : C'est une classe de réseaux de neurones où les connexions entre les noeuds forment un graphe dirigé ou non dirigé le long d'une séquence temporelle. Cela leur permet de présenter un comportement dynamique temporel. Les RNN peuvent utiliser leur état interne (mémoire) pour traiter des séquences d'entrées de longueur variable, ce qui les rend applicables à des tâches telles que la reconnaissance non segmentée et connectée de l'écriture manuscrite ou la reconnaissance de la parole. les RNN montrent leurs performances dans les problèmes de NLP. Un exemple plus détaillé sur les couches RNN : soit  $X$  : le vecteur d'entrée. les Noeuds dans les couches RNN ne sont que comme tout autre noeud dans une couche cachée d'un réseau de neurones et l'entrée de cette couche  $x$  sera transformé par la fonction  $f(x)$  tq :  $f(x) = \sigma(\sum_{i=1}^n w_i x_i + b)$  avec  $\sigma$  est la fonction d'activation,  $w$  : vecteur de poids et  $b$  est le biais.

comme les LSTM, il faut d'abord une couche d'embedding, puis nous ajouterons 50 unités d'une chouche RNN et finalement une couche de sortie de 4 neurones. Pour éviter l'overfitting, nous avons utilisé dans ce modèle seulement la technique de 'EarlyStopping'.

Model: "First-RNN-Model"		
Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 160, 1000)	5001000
Hidden-Recurrent-Layer (SimpleRNN)	(None, 50)	52550
Output-Layer (Dense)	(None, 4)	204
<hr/>		
Total params:	5,053,754	
Trainable params:	5,053,754	
Non-trainable params:	0	

FIGURE 4.16 – modèle 3 : NLP Avec les couches RNN simples

L'algorithme s'est arrêté au 5<sup>eme</sup> epoch et a donné une précision sur la partie train de 88% et sur la partie de valid. de 75% et une perte sur la partie train de 5.1% et sur la partie valid. de 9.3%

Au lieu de visualiser les métriques de ce modèle comme nous l'avons déjà fait avec le modèle précédent, nous allons comparer ces métriques de ce modèle avec les métriques du modèle précédent.

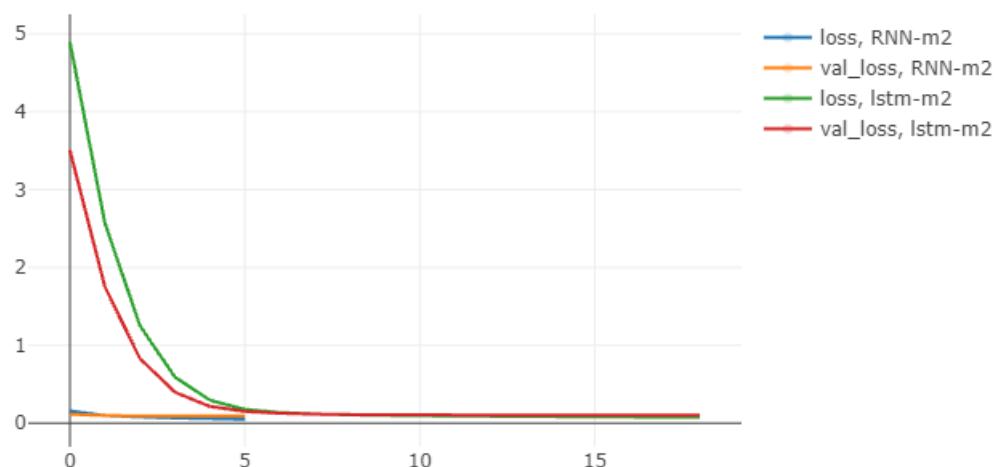


FIGURE 4.17 – Comparaison entre les deux modèles

Dans les 5 premiers epochs, l'architecture avec les couches RNN montre sa performance par rapport à celui avec les couches LSTM mais les performances du modèle avec les couches RNN se stabilisent au 5<sup>eme</sup> epoch alors que celle avec les couches LSTM s'améliorent de plus en plus. Donc, nous allons retenir le modèle avec les couches LSTM pour implémenter notre chatbot pour la classification des messages. Mais avant la construction du chatbot, nous passerons vers la segmentation des prospects pour identifier ceux qui sont qualifiés d'être de nouveaux clients pour la banque.

## 4.3 Segmentation des prospects

### 4.3.1 Ingénierie des variables

Dans la partie précédente, nous avons créé un algorithme pour classifier les commentaires. Dans cette partie nous étudierons chaque prospect à partir de son commentaire, son historique des réactions et d'autres variables comme l'âge, le genre, le lieu de résidence ... La table que nous allons travailler avec est mal organisée. Notre premier objectif est d'organiser la table et de bien définir toutes les variables.

0	1	2	3	4	5	6 ... 19
ÉVÉNEMENTS MARQUANTS	PHOTOS	NaN	NaN	NaN	NaN	NaN ... NaN
COORDONNÉES [REDACTED] 1614460\nFacebook	GÉNÉRALES\ninHomme\ninGenre	INFOS	ÉVÉNEMENTS MARQUANTS	[REDACTED]	PHOTOS	SPORT\nأحرى الجم الرئيسي الساحلي
COORDONNÉES [REDACTED] 1614460\nFacebook	GÉNÉRALES\ninHomme\ninGenre	INFOS	ÉVÉNEMENTS MARQUANTS	[REDACTED]	PHOTOS	SPORT\nأحرى الجم الرئيسي الساحلي
COORDONNÉES [REDACTED] 1614460\nFacebook	GÉNÉRALES\ninHomme\ninGenre	INFOS	ÉVÉNEMENTS MARQUANTS	[REDACTED]	PHOTOS	SPORT\nأحرى الجم الرئيسي الساحلي
INFO GÉNÉRALES\ninHomme\ninGenre	NOMS\ninLucara\nnPseudo	AUTRES	ÉVÉNEMENTS MARQUANTS	PHOTOS	MENTIONS J'AIME\ninFitness Diet\ninFord Tunisie\ninG...	MUSIQUE\ninMFM Radio Tunisie\ninStation de radio\nin...
...	...	...	...	...	...	...
A étudié à Lycée Secondaire H.H.A.W ( Page Off... Marié(e)		SCOLARITÉ\ninLycée Secondaire H.H.A.W ( Page Off...	SITUATION AMOUREUSE\ninMarié(e)	ÉVÉNEMENTS MARQUANTS	PHOTOS	VIDÉOS ... NaN
A étudié à fac de lettres	De Grombalia, Nabil, Tunisia	Suivi par 42 personnes	SCOLARITÉ\ninfac de lettres\ninUniversité	LIEUX DE RÉSIDENCE\ninGrombalia, Nabil, Tunisia...	COORDONNÉES\nin[REDACTED]\nFacebook	GÉNÉRALES\ninHomme\ninGenre ... NaN
COORDONNÉES\nin[REDACTED]\nFacebook	GÉNÉRALES\ninHomme\ninGenre	INFOS	ÉVÉNEMENTS MARQUANTS	[REDACTED]	PHOTOS	VIDÉOS MUSIQUE\ninByehsido\ninGeorge Wassef ... NaN
A travaillé à Being Hyper!	A étudié à Histoire	A étudié à Faire rire le voisin de classe quan...	Habite à Bizerte	De Tunis	SCOLARITÉ\ninHistoire\ninUniversité\ninFaire rire le...	EMPLOI\ninBeing Hyper! ... NaN
Menzel Bourguiba	A étudié à bizerte	A étudié à Lycée Secondaire Mateur - 2 Mars 1934	De Tunis	SCOLARITÉ\ninbizerte\ninUniversité\ninLycée Secondar...	EMPLOI\ninMenzel Bourguiba	LIEUX DE RÉSIDENCE\ninTunis\ninVille d'origine ... NaN

FIGURE 4.18 – Tableau pour le problème de segmentation

La première étape est de reconstruire ces variables afin d'obtenir les informations suivantes : Lieux de résidence, état civil, scolarité, emploi, âge et genre.

Nous allons parcourir le tableau ligne par ligne et à chaque fois on trouve une information importante on l'ajoute dans la variable correspondante, Au cas où on n'a pas trouvé une

information concernant le genre par exemple, on le note comme une valeur nulle.

Pour les données de personnes, il y a deux types de variables :

- le premier type contient des valeurs étiquetées comme

\*LIEUX DE RÉSIDENCE\nTunis\nVille actuelle\*\* . Donc ici on voit bien que cette valeur appartient à la variable \*\*LIEUX DE RÉSIDENCE\*\*.

- Le deuxième type contient des valeurs non étiquetées comme \*\*Habite à Sousse\*\* ou \*\*Célibataire\*\*. Ces valeurs ne sont pas étiquetées comme le premier type

1. s'il s'agit des valeurs étiquetées, la valeur est alors facile à distinguer dans quelle variable elle appartient.

2. s'il s'agit de valeurs non étiquetées, on peut définir dans quelle variable ces valeurs appartiennent selon cette règle :

- \* Si la valeur commence par \*\*habite à\*\*, alors elle appartient à la variable

\*LIEUX DE RÉSIDENCE\*

- \* Si la valeur commence par \*\*A étudié\*\*, alors elle appartient à la variable

\*SCOLARITÉ\*

- \* Si la valeur commence par \*\*Travaille chez\*\*, alors elle appartient à la variable

\*EMPLOI\*

- \* Si la valeur prend une valeur de cette liste \*\*['Célibataire', 'Marié', 'En union libre', 'Fiancé', 'En couple']\*\*, alors elle appartient à la variable \*SITUATION AMOUREUSE\*

	commentaire	name	reaction	LIEUX DE RÉSIDENCE	INFOS GÉNÉRALES	SCOLARITÉ	EMPLOI	SITUATION AMOUREUSE
0	NaN	NaN	NaN	None	None	None	None	None
1	الله يرحمه ونفعه ويسكنه جلت نعمته... فربي پیشتر...	[REDACTED]	NaN	None	INFOS GÉNÉRALES\ninHomme\nGenre	None	None	None
2	ان الله وان الله راجحون ، الله يرحمو	[REDACTED]	NaN	LIEUX DE RÉSIDENCE\ninBizeribe\inVille actuelle\inB...	INFOS GÉNÉRALES\nin7 février\nDate de naissance	SCOLARITÉ\ninESC\inUniversité	EMPLOI\ninbanquier\nin16 avril 2022 à aujourd'hui...	SITUATION AMOUREUSE\ninCélibataire
3	الله اکبر ان الله وان الله راجحون	[REDACTED]	NaN	LIEUX DE RÉSIDENCE\ninKheredine, Tunis, Tunisia\...	INFOS GÉNÉRALES\ninHomme\nGenre	SCOLARITÉ\ninfac de droit et des sciences politi...	EMPLOI\ninCommerçant	SITUATION AMOUREUSE\ninMarié(e)
4	الله يرحمه ونفعه والي ربى	[REDACTED]	NaN	None	INFOS GÉNÉRALES\ninHomme\nGenre	None	None	None

### — LIEUX DE RÉSIDENCE :

La variable LIEUX DE RÉSIDENCE peut avoir la date de déménagement On va ajouter cette information dans une nouvelle variable \*déménagement en\*

### — variable infos générales

- \* Cette variable contient plusieurs informations comme genre et date de naissance

...

- \* Nous avons besoin seulement de ces deux variables : \*Genre\* et \*Lieu de Résidence\*

\* le format d'une valeur de cette variable est le suivant : Homme\nGenre\nItalien et Anglais\Langues

\* Si \*\*genre\*\* existe, on va prendre la valeur juste avant du mot \*genre\*

\* Si \*\*Date de naissance\*\*existe, on va prendre la valeur juste avant du mot \*Date de naissance\*

\* On peut trouver la valaur suivante : \*\*Hommes\nIntéressé(e) par\*\*

\* Dans ce cas, si genre n'existe pas, on va le mentionner comme l'opposé de la variable 'intéressé par' c'est à dire si cette personne est interessée par hommes, elle est une femme et vice versa

#### — Scolarité

\* La variable \*scolarité\* peut avoir une date

\* On va ajouter cette information dans une nouvelle variable \*\*time\*\*

cette variable peut avoir la classe de l'établissement, on va stocker cette information dans une autre variable \*\*classe\*\*

par exemple :IHEC\nUniversité\nPromotion 2011

scolarite : IHEC, classe : Université, time : Promotion 2011

#### — EMPLOI

\* La variable \*emlpoi\* peut avoir une date

\* On va ajouter cette information dans une nouvelle variable \*\*timeE\*\*

\* par exemple banquier\n16 avril 2022 à aujourd'hui\nBanque Zitouna \n7 juillet 2018 à aujourd'hui

\* emploi banquier, time : 16 avril 2022 à aujourd'hui

La variable âge n'est pas définie explicitement mais il y a d'autres variables qu'on peut utiliser pour déduire l'âge. Par exemple : la date de naissance, l'année d'entrée aux études universitaires,..

Pour compter le nombre de réactions, nous créons une sous table avec les variables de réactions et nom. Puis, nous supprimons les lignes où la réaction est une valeur nulle. Ensuite, on regroupe la table par la variable nom avec la fonction » **grouby('name').sum()**.

Finalement on fusionne cette table de réactions avec celle contenant les commentaires.

```
» pd.merge(table_mère,table_réaction, on='left', by='name').
```

On termine par réaffecter un score aux réactions par la règle suivante :

$$S = -3 * \text{reaction\_angry} + 2 * \text{reaction\_care} + (-0.5) * \text{reaction\_laugh} + 1 * \text{reaction\_like} + 3 * \text{reaction\_love} + (-2) * \text{reaction\_sad} + 0.5 * \text{reaction\_wow}.$$

La variable lieux de résidence prend 9 modalités :

- DIRECTION REGIONALE TUNIS
- DIRECTION REGIONALE SFAX
- DIRECTION REGIONALE SOUSSE
- DIRECTION REGIONALE CAP BON
- DIRECTION REGIONALE NORD
- DIRECTION REGIONALE SUD EST
- DIRECTION REGIONALE SUD OUEST
- DIRECTION REGIONALE MONASTIR
- Etranger.

La variable cible de cette table est target. Cette variable target possède deux modalités {0 : prospect non qualifié, 1 : prospect qualifié }.

Après la déduction de la variable âge, Nous trouvons que 1.5% de ses valeurs sont non nulles. Nous supprimons donc la variable âge de notre table de données.

On remplace les valeurs nulles par 0 et on suppose que s'il y a moins de deux valeurs non nulles par observation, alors il s'agit d'un faux profil et on affecte la variable cible par 0. De même, si la variable reaction\_score est négative ou la classe du commentaire y est égale à 0, on affecte la variable cible par 0. sinon target va prendre 1 comme valeur.

Finalement, on récupère la base suivante :

y	name	LIEUX DE RÉSIDENCE	SITUATION AMOUREUSE	Genre	classe	emploi	reaction_score	target
0 1	[REDACTED]		0	0 0	0 0	0 0	0.0	0
1 3	[REDACTED]		0	0 Homme	0 0	0 0	0.0	0
2 3	[REDACTED]		0	0 Homme	0 0	0 0	0.0	0
3 1	[REDACTED]	North_East		0 Homme	Université Coiffeur	0 0	0.0	1
4 0	[REDACTED]	Etranger		Marié Homme	Université	0 0	0.0	0
5 1	[REDACTED]	North_East		0 Femme	Université	0 0	0.0	1
6 1	[REDACTED]	0		0 0	0 0	0 0	0.0	0

### 4.3.2 Statistique descriptive

On observe qu'environ de 1100(59%) individus ont été labélisé comme non qualifiés d'être des clients pour la banque Zitouna au contraire des autres 800(40%) des prospects.

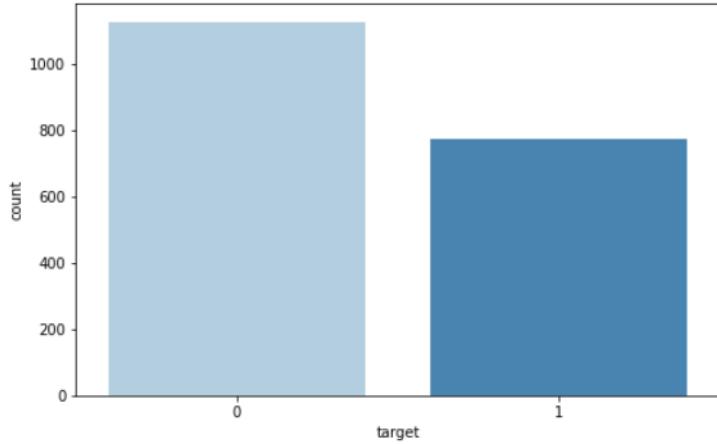


FIGURE 4.19 – Distribution de la variable target

D'après le bar plot ci dessous, on observe que parmi les personnes qui ne peuvent pas être des clients pour la banque, plus que 40% d'eux n'ont pas spécifiés leurs genres. Alors que parmi ceux qui peuvent être des clients pour la banque, moins de 15 % n'ont pas spécifiés leurs genres.

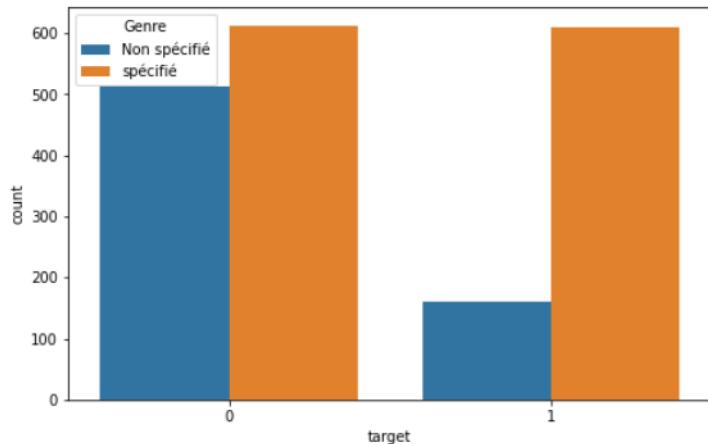


FIGURE 4.20 – Distribution de la variable target selon la variable genre

D'après le bar plot ci dessous, on observe plus que 60% des personnes qui ne peuvent pas être de nouveaux clients pour la banque n'ont pas spécifiés leurs lieux de résidences. Et seulement 15% des personnes qui peuvent être de nouveaux clients n'ont pas spécifié leurs lieux de résidence.

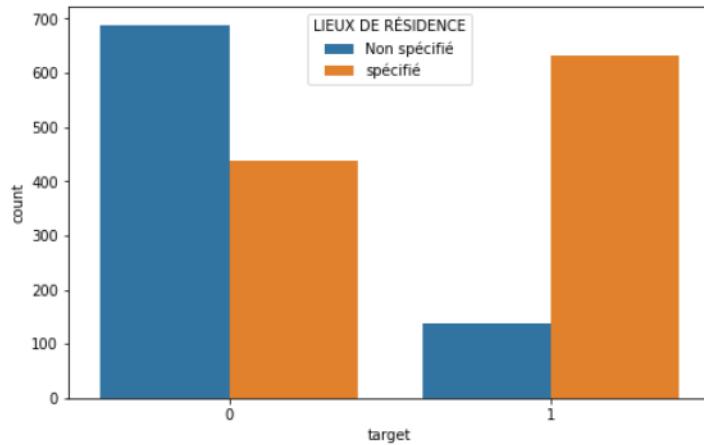


FIGURE 4.21 – Distribution de la variable target selon la variable lieux de résidence

### 4.3.3 Modélisation

Dans cette partie, nous utiliserons la régression logistique, une des méthodes de bagging, les forêts aléatoires, et une des méthodes de boosting, XGBoost. Mais avant de commencer, on fait un encodage pour les variables catégorielles.

y	name	LIEUX DE RÉSIDENCE	SITUATION AMOUREUSE	Genre	classe	emploi	reaction_score	target
0	1	470	0	0	0	0	2	0
1	3	1040	0	0	2	0	0	2
2	3	1409	0	0	2	0	0	2
3	1	437	4	0	2	2	75	2
4	0	1202	3	7	2	2	0	2
...	...	...	...	...	...	...	...	...
1891	1	1245	0	7	0	1	0	2
1892	1	1068	3	0	2	2	0	2
1893	2	1292	0	0	2	0	0	2
1894	1	1268	4	0	1	2	32	2
1895	1	940	4	0	0	2	221	2

1896 rows × 9 columns

Nous divisons les données en 80% pour le train et 20% pour le test.

### Régression logistique

La régression logistique est utilisée pour calculer ou prédire la probabilité d'un événement binaire (oui/non), d'où le nom classification binaire.

$\sigma(x) = \frac{1}{1+\exp^{-(\beta_0 + \beta_1 x)}}$  avec  $\beta_0$  est le bias,  $x$  est la variable explicative et  $\beta_1$  est le vecteur poids associé à  $x$

En pratique, nous allons utiliser la fonction logistique implémentée dans la librairie Sklearn.

```
» from sklearn.linear_model import LogisticRegression
```

Nous utiliserons la fonction perte  $l_2$ . La régression logistique dans notre cas minimise la fonction de coût suivante :

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^n \log(\exp(-y_i(X_i^T w + c)) + 1)$$

La courbe de ROC nous montre une bonne performance de notre modèle vue le nombre des valeurs bien prédites : vrais positifs et vrais négatifs

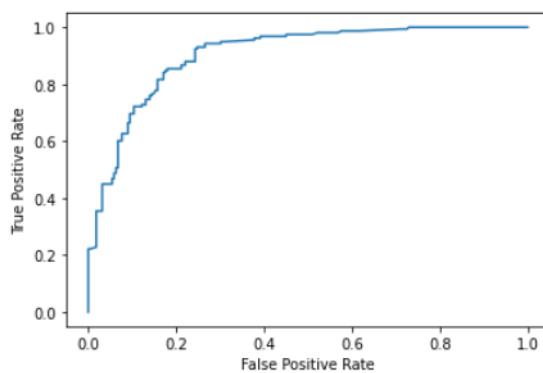


FIGURE 4.22 – Régression logistique : courbe de ROC

Nous remarquons que pour la classe 0, nous avons obtenu une précision de 83%, un f score de 85% et un recall de 86%. Et pour la classe 1, nous avons obtenu une précision de 79%, un f score de 78% et un recall de 76%.

	precision	recall	f1-score	support
0	0.83	0.86	0.85	222
1	0.79	0.76	0.78	158
accuracy			0.82	380
macro avg	0.81	0.81	0.81	380
weighted avg	0.82	0.82	0.82	380

FIGURE 4.23 – Régression logistique : indicateurs de performances obtenus sur l'échantillon de test

## Bagging : random forest

L'algorithme nous a donné un score de 97%

La courbe de ROC pour le random forest montre la performance de notre modèle selon les valeurs des vrais positifs et vrais négatifs.

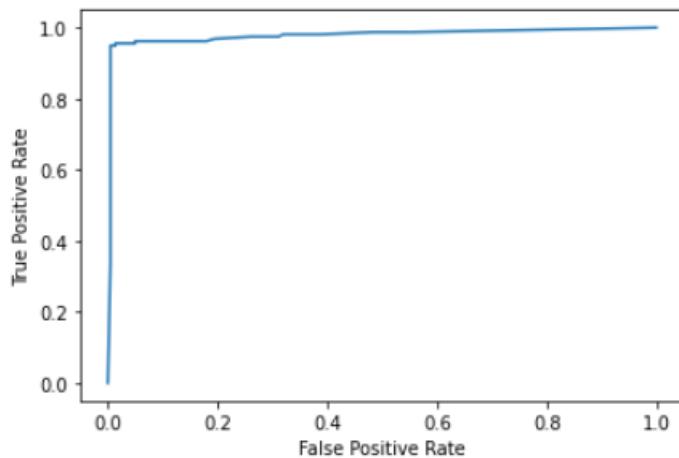


FIGURE 4.24 – Forêt aléatoire : courbe de roc

Selon la figure ci dessous, nous retenons que la précision et le f-score des deux classes sont plus que 97% et un recall de 100% pour la classe 0 et 95% pour la classe 1

	<code>precision</code>	<code>recall</code>	<code>f1-score</code>	<code>support</code>
0	0.97	1.00	0.98	222
1	0.99	0.95	0.97	158
<code>accuracy</code>			0.98	380
<code>macro avg</code>	0.98	0.97	0.98	380
<code>weighted avg</code>	0.98	0.98	0.98	380

FIGURE 4.25 – Random forest : indicateurs de performances obtenus sur l'échantillon de test

## Boosting : XGBOOST

Le boosting est une technique d'ensemble dans laquelle de nouveaux modèles sont ajoutés pour corriger les erreurs commises par les modèles existants. Les modèles sont ajoutés séquentiellement jusqu'à ce qu'aucune amélioration ne puisse être apportée.

Avec l'algorithme Xgboost, nous obtiendrons un score sur le train de : 99.22% et un score sur le test de : 97.37%

Les modèles Xgboost nous interprète encore l'importance de chaque variable selon le f-score.

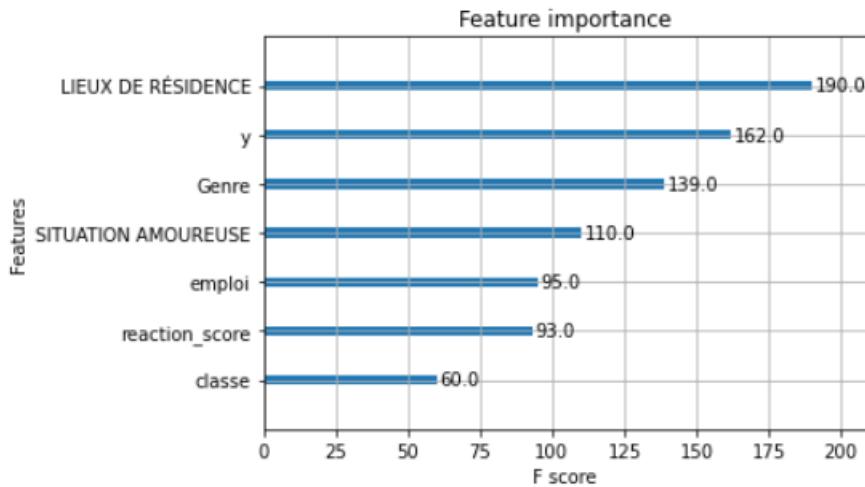


FIGURE 4.26 – XGboost : importance des variables

On observe que la variable la plus corrélée avec la variable cible est la variable :"LIEUX DE RÉSIDENCE" puis on trouve la variable "y" qui représente la classe des commentaires alors que la variable la moins corrélée avec la variable cible est la classe qui représente le niveau d'éducation.

## 4.4 Comparaison des modèles :

	Accuracy sur le train	Accuracy sur le test
Régression logistique	0.84	0.81
Forêt aléatoire	0.99	0.97
Xgboost	0.99	0.97

TABLE 4.3 – Comparaison des résultats

Nous avons trouvé les meilleurs résultats avec les algorithmes d'ensembles : une précision de 99% sur la partie d'entraînement et 97% sur la partie de test. Alors qu'avec la régression logistique, nous avons trouvé une précision de 84% sur le train et 81% sur le test.

## 4.5 Conclusion

Dans ce chapitre, nous avons développé un modèle de classification de commentaires que nous avons collectés de Facebook et nous avons réussi à segmenter les prospects selon leurs

commentaires et leurs réactions avec les méthodes de bagging, boosting et régression logistique. Nous avons trouvé des résultats encourageants avec les algorithmes Random Forest et Xgboost.

# Chapitre 5

## ChatBot : déploiement des modèles

### 5.1 Introduction

Le but de ce projet de fin d'études est de créer un chatbot pour la page facebook de la banque Zitouna qui peut distinguer les utilisateurs qui peuvent être de nouveaux clients pour la banque à partir d'un simple message. De nos jours, le chatbot est devenu un besoin nécessaire pour toutes les entreprises, pour améliorer le temps de réponse aux demandeurs d'informations et pour stabiliser une belle image à l'entreprise. sans un chatbot, une personne peut attendre un jour pour avoir une réponse comme le cas de la banque Zitouna.

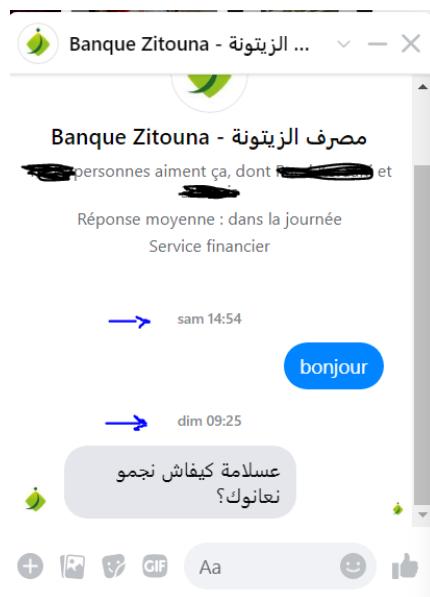


FIGURE 5.1 – Exemple d'une réponse lente de la part de la banque Zitouna

Dans ce chapitre nous déployerons notre modèles NLP pour construire un chatbot simple qui sert à classer les besoins de chaque prospect à partir de son message. Au premier

lieu, nous allons créer un serveur pour enregistrer les messages du Facebook. Nous utiliserons le framework Django.

Puis, nous définirons une fonction permettant de renvoyer des messages aux utilisateurs qui est la fonction requests.

Finalement, nous transmettrons une connexion https à notre machine locale en utilisant ngrok.

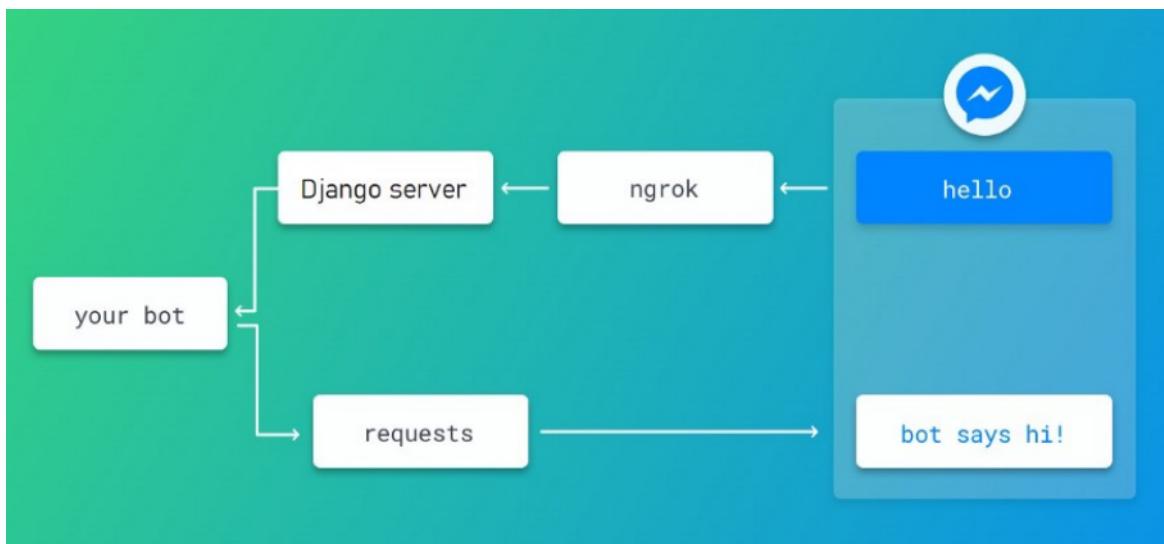


FIGURE 5.2 – Chatbot Messenger (source : thecodespace.in)

## 5.2 Construction du chatbot

### 5.2.1 Mécanisme de réception des messages sur la page Facebook de la banque Zitouna

La première étape pour la création de chatbot sur messenger est de créer un projet Django pour écouter les messenges reçus du Facebook Messenger.

Django est un framework web Python de haut niveau qui permet de développer rapidement des sites web sécurisés et faciles à maintenir. L'architecture implémenté par Django regroupe les fichiers Urls, views, model, et template. il reçoit une demande ou un request par le fichier URLs, traite cette demande dans le fichier views puis il envoie une réponse HTTP Response.

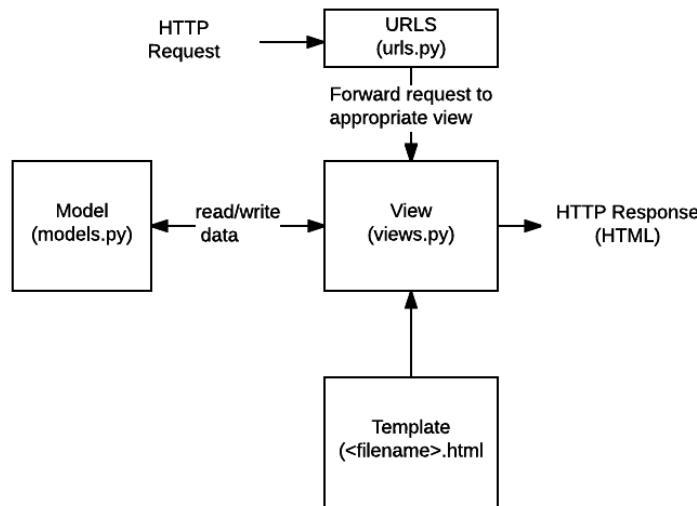


FIGURE 5.3 – Architecture django (source : developer.mozilla.org)

Pour commencer, on installe Django via la commande `> pip install django` et on crée notre projet django `> django-admin.py startproject nom_du_projet`. Suite à la création du projet, le fichier urls se crée automatiquement. Pour lancer le serveur Django , on utilise la commande suivante : `> python manage.py runserver`. Les fichiers View et model se trouvent dans notre application ‘AIchatbot’. On peut activer l’application via `> django-admin startapp AIchatbot`

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows files in the project structure:
  - OPEN EDITORS:** settings.py, urls.py, tests.py, \_\_init\_\_.py, urls.py, views.py, manage.py.
  - CHATBOTPT:** tokenizer.pickle, urls.py, views.py, db.sqlite3, manage.py, .ngrok.exe.old, ngrok-v3-stable-windows-amd64.zip, ngrok.exe.
- Code Editor:** The code for `views.py` is displayed:
 

```

class AibotView(generic.View):
    def get(self, request, *args, **kwargs):
        if self.request.GET['hub.verify_token'] == '123456':
            return HttpResponse(self.request.GET['hub.challenge'])
        else:
            return HttpResponse('Error, invalid token')

    @method_decorator(csrf_exempt)
    def dispatch(self, request, *args, **kwargs):
        return generic.View.dispatch(self, request, *args, **kwargs) # Post function to handle Facebook messages
    def post(self, request, *args, **kwargs):
        # Converts the text payload into a python dictionary
        incoming_message = json.loads(self.request.body.decode('utf-8'))
        # Facebook recommends going through every entry since they might send
        # multiple messages in a single call during high load
        for entry in incoming_message['entry']:
            for message in entry['messaging']:
                # check to make sure the received call is a message call

```
- Terminal:** Shows command-line output:
 

```

PS C:\Users\Lenovo\OneDrive - Ministere de l'Enseignement Superieur et de la Recherche Scientifique\Bureau\chatbotpt> cd aibot
PS C:\Users\Lenovo\OneDrive - Ministere de l'Enseignement Superieur et de la Recherche Scientifique\Bureau\chatbotpt\airobot> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
May 31, 2022 - 20:46:31
Django version 4.0.4, using settings 'airobot.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```

FIGURE 5.4 – Application django

## 5.2.2 Étapes de renvoie des réponses aux utilisateurs

Pour renvoyer une réponse à l'utilisateur nous devons implémenter la fonction post qui reçoit le message de l'utilisateur et retourne une réponse. Dans cette fonction, nous devons faire un nettoyage de message comme nous l'avons fait dans le nettoyage des commentaires dans le troisième chapitre dans la partie linguistique du NLP. Puis nous importons notre modèle 'neutral network' par la fonction » `models.load_model(_nom_du_modèle)` et nous prédisons la classe du message. Enfin, suivant la classe obtenue, on retourne le message correspondant. la réponse sera envoyé via `post_message_url` qui est connecté par facebook developper (Application développé par Meta). Nous ajoutons quelques questions, réponses pour vérifier si le prospect est déjà un client de la banque zitouna. Si oui, nous avons déjà toutes les informations de ce prospect. Sinon on demande son numéro de téléphone pour le contacter dès que possible par le service marketing.

```
1 def post_facebook_message(fbid, recevied_message):
2     recevied_message=recevied_message.lower()
3     if recevied_message == 'oui':
4         responsee = 'pouvez vous nous envoyez votre client id qui commence
5         par: id_?'
6     elif recevied_message == 'non':
7         responsee = 'pouvez vous nous envoyer votre numéro telephone?'
8     elif recevied_message.startswith('id_') or all(char.isdigit() for char
9         in recevied_message):
10        responsee = "Merci pour visiter notre page facebook, pour avoir plus
11        d'informations sur nos produits, nous vous inviterons de visiter notre
12        site web: https://www.banquezitouna.com/fr/presentation-banque-zitouna\n
13        au revoir =D"
14    else:
15        df = pd.DataFrame([recevied_message], columns=["commentaire"])
16
17        X = tokenizer.texts_to_sequences(df.commentaire.values)
18        X = pad_sequences(X)
19        y = np.zeros((1,160))
20        for i in range(len(X[0])):
21            y[0][-i-1] = X[0][i]
22
23        model = models.load_model("AibotView/neural_network")
```

```
18 pred = model.predict(y)
19
20 print(pred)
21
22 t = int(np.where(pred == pred.max())[1])
23
24 if t == 0:
25     recevied_message = 'autre'
26
27 elif t ==1:
28     recevied_message = 'particulier'
29
30 elif t ==2:
31     recevied_message='professionnelle'
32
33 else:
34     recevied_message = 'service'
35
36
37
38 # Remove all punctuations, lower case the text and split it based on
39 space
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
```

Listing 5.1 – Fonction post pour renvoyer une réponse

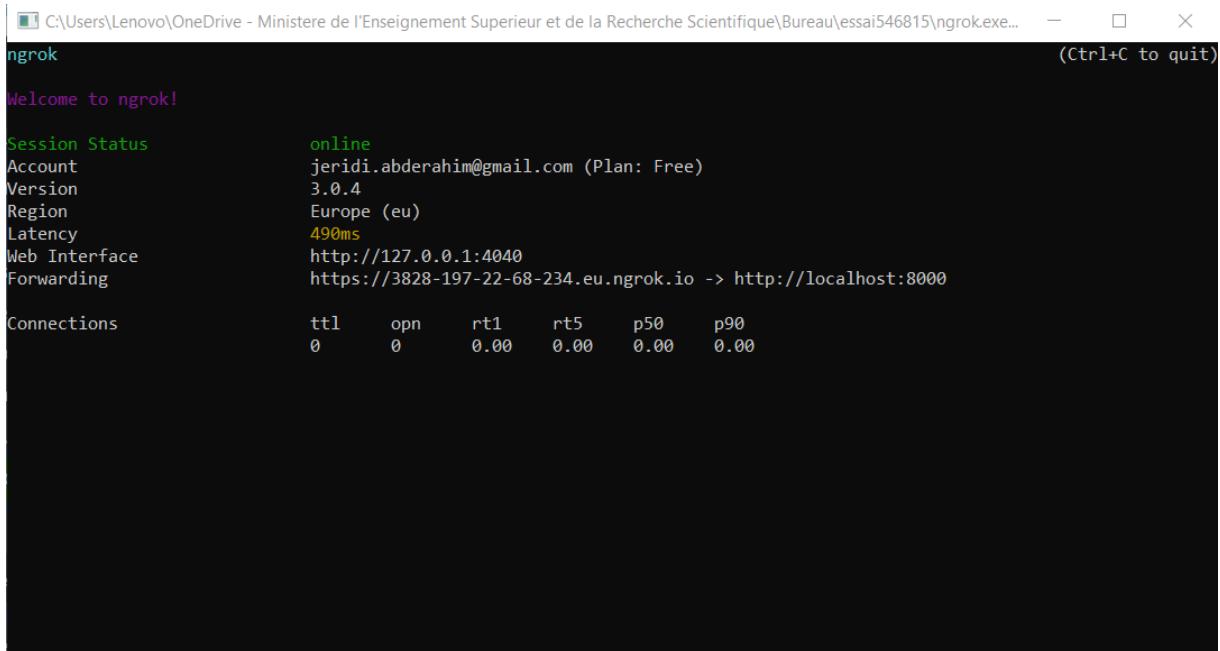
### 5.2.3 Etablir la connection https avec la machine locale

La connection https sera être configuré par ngrok.

ngrok est le bord de réseau programmable qui ajoute la connectivité, la sécurité, et l'observabilité à vos applications sans changement de code.

Pour activer le serveur ngrok, nous utilisons la commande > **ngrok http port** dans le terminal. dans notre cas, le port est 8000. Et une fenêtre contenant le lien de connexion dans

la variable : *Forwarding*



```
C:\Users\Lenovo\OneDrive - Ministere de l'Enseignement Supérieur et de la Recherche Scientifique\Bureau\essai546815\ngrok.exe... — □ ×
ngrok
Welcome to ngrok!

Session Status      online
Account            jeridi.abderahim@gmail.com (Plan: Free)
Version            3.0.4
Region             Europe (eu)
Latency            490ms
Web Interface     http://127.0.0.1:4040
Forwarding         https://3828-197-22-68-234.eu.ngrok.io -> http://localhost:8000

Connections        ttl     opn     rt1     rt5     p50     p90
                   0       0     0.00    0.00    0.00    0.00
```

FIGURE 5.5 – connection à ngrok

Notre serveur django est activé, la connexion s'est faite par ngrok. Il ne reste maintenant que de configurer facebook developper pour le lier avec le lien obtenu par ngrok. Pour ce fait, nous devons ajouter le lien obtenu par ngrok comme un url de rappel en spécifiant le nom de l'application développée sous django : *url\_ngrok\_nom\_application* puis on vérifie l'url en indiquant un jeton qui doit être ajouté dans la fonction get de l'application

```
1 def get(self, request, *args, **kwargs):
2
3     if self.request.GET['hub.verify_token'] == '123456':#jeton de vérification
4
5         return HttpResponse(self.request.GET['hub.challenge'])
```

Listing 5.2 – Jeton de vérification

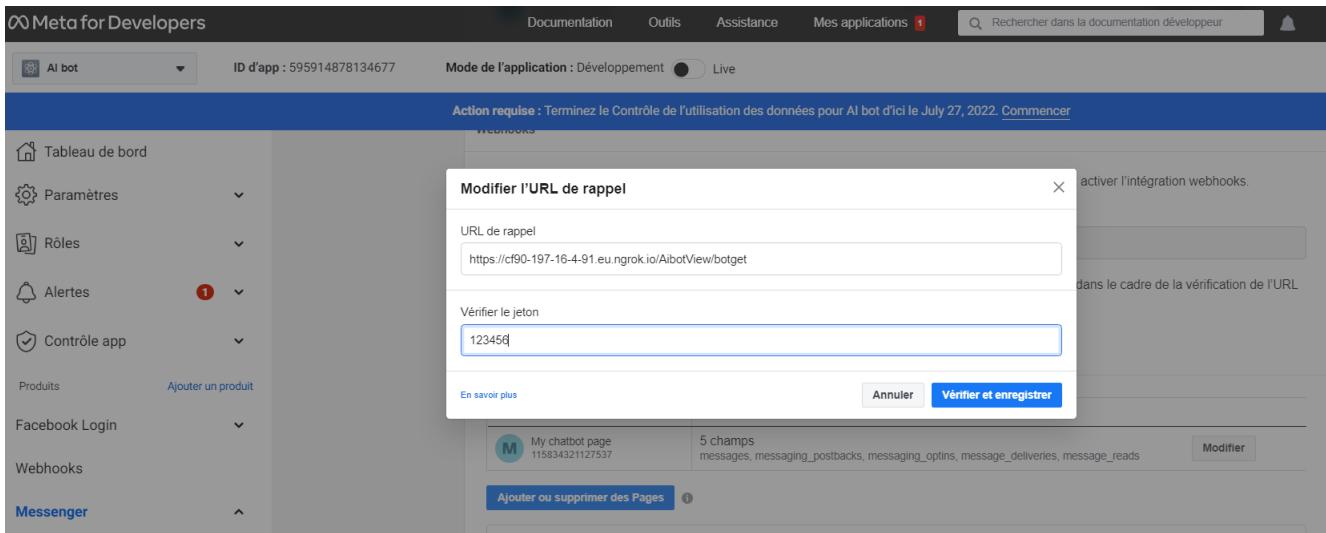


FIGURE 5.6 – Adresse URL de rappel

### 5.3 Présentation des fonctionnalités du ChatBot

Toutes les connections sont bien établies. il ne reste maintenant que de tester notre bot messanger.

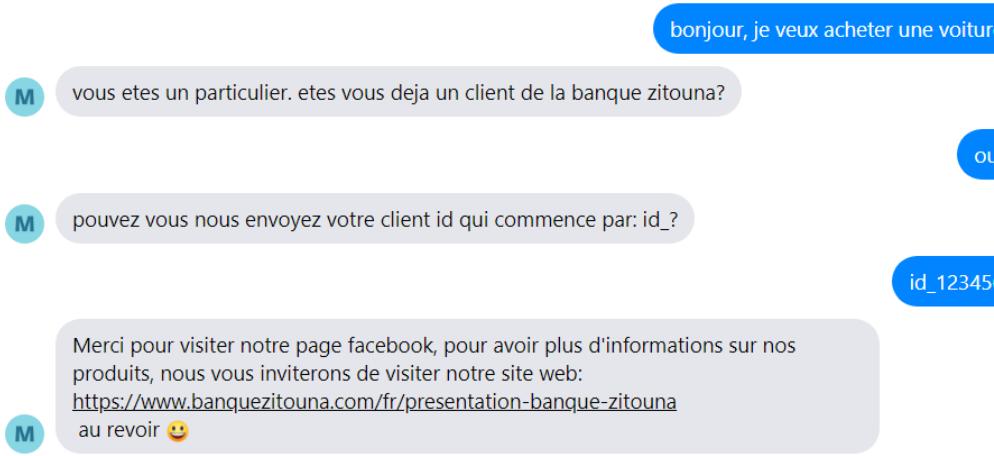
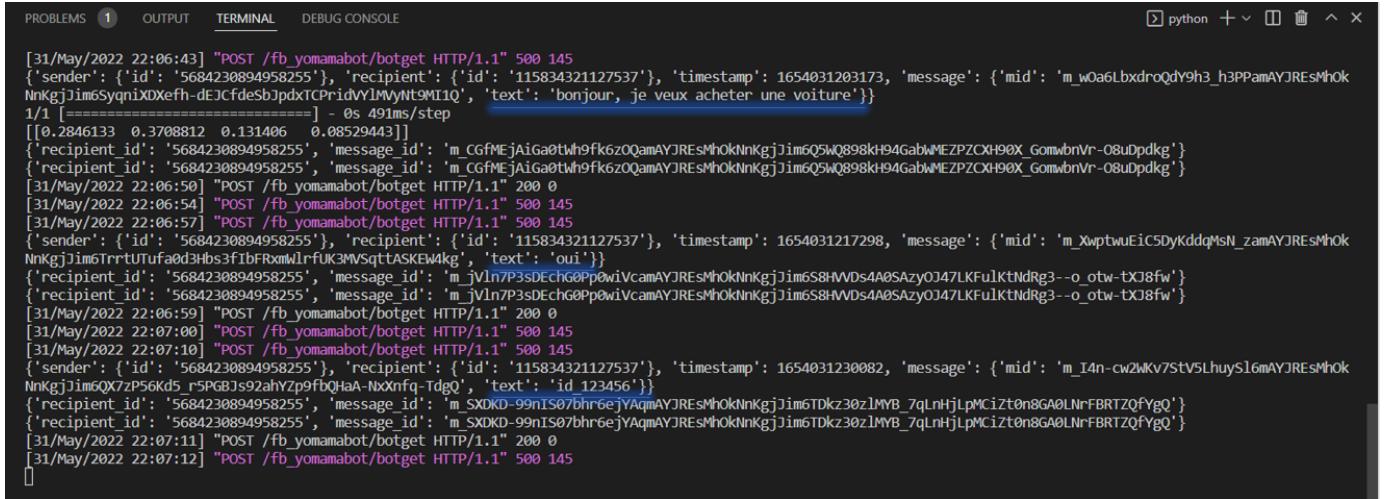


FIGURE 5.7 – Exemple d'une conversation avec le bot messenger

les messages reçus par l'utilisateurs sont affichés dans le terminal.



The screenshot shows a terminal window with several lines of text. The text is a log of messages received by a user, likely from a bot named 'yomamabot'. The log includes timestamps, message IDs, recipient IDs, and the text of the messages. The messages are in French, such as 'bonjour, je veux acheter une voiture'.

```
[31/May/2022 22:06:43] "POST /fb_yomamabot/botget HTTP/1.1" 500 145
{"sender": {"id": "5684230894958255"}, "recipient": {"id": "115834321127537"}, "timestamp": 1654031203173, "message": {"mid": "m_woa6LbxdroQdyh3_h3PPamAYJREsMhokNnKgjJim6SyqniDXefh-dEJCfdesBjpdxTCPrividV1MVNT9MI10", "text": "bonjour, je veux acheter une voiture"}}
1/1 [=----] 0s 491ms/step
[0.2846133 0.3708812 0.131406 0.08529443]
{"recipient_id": "5684230894958255", "message_id": "m_CgfMEjAiga0twh9fk6z0QamAYJREsMhokNnKgjJim6Q5WQ898kH94GabwMEZPZXH90X_GomwbvNr-08uDpdkg"}
{"recipient_id": "5684230894958255", "message_id": "m_CgfMEjAiga0twh9fk6z0QamAYJREsMhokNnKgjJim6Q5WQ898kH94GabwMEZPZXH90X_GomwbvNr-08uDpdkg"}
[31/May/2022 22:06:50] "POST /fb_yomamabot/botget HTTP/1.1" 200 0
[31/May/2022 22:06:54] "POST /fb_yomamabot/botget HTTP/1.1" 500 145
[31/May/2022 22:06:57] "POST /fb_yomamabot/botget HTTP/1.1" 500 145
{"sender": {"id": "5684230894958255"}, "recipient": {"id": "115834321127537"}, "timestamp": 1654031217298, "message": {"mid": "m_XwptwuEiC5DyKddqMsN_zamAYJREsMhokNnKgjJim6TrttUtufaod3fbfRxfmWrfuK3MVsqtASKEW4kg", "text": "oui"}}
{"recipient_id": "5684230894958255", "message_id": "m_jVln7P3sDechG0Pp0wiVcamAYJREsMhokNnKgjJim6S8HVVDs4A0SAzy0J47LKFu1KtNdRg3--o_otw-tXJ8fw"}
{"recipient_id": "5684230894958255", "message_id": "m_jVln7P3sDechG0Pp0wiVcamAYJREsMhokNnKgjJim6S8HVVDs4A0SAzy0J47LKFu1KtNdRg3--o_otw-tXJ8fw"}
[31/May/2022 22:06:59] "POST /fb_yomamabot/botget HTTP/1.1" 200 0
[31/May/2022 22:07:00] "POST /fb_yomamabot/botget HTTP/1.1" 500 145
[31/May/2022 22:07:10] "POST /fb_yomamabot/botget HTTP/1.1" 500 145
{"sender": {"id": "5684230894958255"}, "recipient": {"id": "115834321127537"}, "timestamp": 1654031230082, "message": {"mid": "m_I4n-cw2Wkv7Stv5LhuySl6mAYJREsMhokNnKgjJim6QX7zP56Kd5_r5PGBjs92ahyZp9fbQfaA-NxXnfq-TdgQ", "text": "id 123456"}}
{"recipient_id": "5684230894958255", "message_id": "m_SxDKD-99nIS07bhr6ejYaqmAyJREsMhokNnKgjJim6TDkz30z1MYB_7qLnHjLpMcizt0n8GA0LNrFBRTZQfygQ"}
{"recipient_id": "5684230894958255", "message_id": "m_SxDKD-99nIS07bhr6ejYaqmAyJREsMhokNnKgjJim6TDkz30z1MYB_7qLnHjLpMcizt0n8GA0LNrFBRTZQfygQ"}
[31/May/2022 22:07:11] "POST /fb_yomamabot/botget HTTP/1.1" 200 0
[31/May/2022 22:07:12] "POST /fb_yomamabot/botget HTTP/1.1" 500 145
```

FIGURE 5.8 – Message reçu par un utilisateur

## 5.4 Conclusion

Dans ce chapitre, nous avons créé notre chatbot sur messenger en utilisant les modèles que nous avons déjà développés dans le chapitre 4. Le chatbot est développé avec les framework Django et ngrok.

## Conclusion générale

Au cours de ce projet de fin d'études, nous avons réussi de construire une base de données sur les commentaires et les réactions du page Facebook de la banque Zitouna avec les méthodes du Web Scraping. La construction de cette base de données s'est faite dans mongoDB.

Nous avons analysé, d'une part, les commentaires collectés et les classer avec le traitement du language propre/Natural language processing(NLP). Dans cette étape , nous avons nettoyé les commentaires dans la partie linguistique du NLP. Puis nous avons construit notre architecture de modélisation pour classer ces commentaires dans la partie modélisation du NLP. Nous avons utilisé les couches LSTM et les couches RNN simples et nous avons trouvé presque les mêmes résultats. Cependant, le modèle avec les couches LSTM s'est entraîné plus que celui avec les couches RNN. Donc, on a choisi l'architecture avec les couches LSTM.

D'autre part, nous avons utilisé le résultat obtenu dans la partie NLP pour segmenter les prospects du Facebook à partir de leurs commentaires, leurs historiques des réactions et leurs données partagées sur leurs profils Facebook et de les classer en des prospects qualifiés ou non qualifiés d'être de nouveaux clients pour la banque Zitouna. Dans cette étape, nous avons d'abord sélectionné les variables et pré-traité les données. Puis nous avons utilisé trois modèles : Régression logistique, Random Forest et XGboost. Le meilleur résultat était obtenu avec XGboost et les forêts aléatoires avec un score de 99% sur les données d'entraînement et 97% sur les données test.

Finalement, nous avons implémenté un chatbot sur messenger pour gagner les profils qui peuvent être intéressants pour la banque.

Dans le futur proche, le chatbot sera utilisé par le service marketing. On sera ainsi capable de se profiter de plusieurs personnes qui veulent acheter un produit ou un service sans besoin d'aller vers la banque au moins dans une première étape.

Pour finir, ce stage que j'ai effectué au sein de la banque Zitouna a été une excellente occasion d'apprentissage et de développement professionnel. Ce stage m'a apporté beaucoup de confiance en soi. Il y a eu des hésitations au début sur l'organisation du travail et sur la

méthodologie de collecte de données. Mais avec la connaissance métier, j'ai réussi à trouver une démarche permettant d'obtenir des résultats fiables et pertinents. De plus, grâce à ce stage, j'ai enrichi mes connaissances avec les méthodes d'apprentissage automatique. L'apport de ce travail était une grande valeur ajoutée pour ma carrière.

# Bibliographie

- [1] <https://www.selenium.dev/selenium/docs/api/py/common/selenium.common.exceptions.html>.
- [2] Selenium (software). [https://en.wikipedia.org/wiki/Selenium\\_\(software\)](https://en.wikipedia.org/wiki/Selenium_(software)).
- [3] Web scraping. [https://en.wikipedia.org/wiki/Web\\_scraping](https://en.wikipedia.org/wiki/Web_scraping).
- [4] Rowel Atienza. Lstm by example using tensorflow. 2017.
- [5] Avijeet Biswal. Recurrent neural network (rnn) tutorial : Types, examples, lstm and more.  
last updated on 2022.
- [6] CHRISTOPHE. Linkedin contre facebook : quel est le meilleur choix pour votre entreprise ?
- [7] Pieter Coussement. Serving decision forests with tensorflow. 2021.
- [8] Chinmay das. What is machine learning and types of machine learning — part-1. 2017.
- [9] Sentiment Analysis Tools for Social Media Marketers. Dara fontein. 2017.
- [10] Ahmed Fawzy Gad. Evaluating deep learning models : The confusion matrix, accuracy, precision, and recall. 2020.
- [11] l'admin du ‘TheCodeSpace’. How to build a facebook messenger bot using django, ngrok,facebook api. 2022.
- [12] Stanislav Lukashevich. Use python to scrape & visualize likes on your linkedin posts. 2020.
- [13] Stanislav Lukashevich. Machine learning applied to international business. 2021.
- [14] Md. ZahidulIslam Sanjid Nusrat JahanPruttasha Md. ShihabUddin Md ArmanHossain Md. AbdulKader Jilani Md.Kowsher, AnikTahabilde. Lstm-ann bilstm-ann : Hybrid deep learning models for enhanced classification accuracy. 2021.
- [15] Amal Menzli. Tokenization in nlp : Types, challenges, examples, tools. 2021.
- [16] Alan Nichol. Deploy your facebook messenger bot with python. 2018.

- [17] Nishant. Gdpr compliance in web scraping. 2021.
- [18] Christopher Olah. Understanding lstm networks. 2015.
- [19] Sruthi E R. Understanding random forest. 2021.
- [20] C. Benavent Sophie Balech. Les techniques du nlp pour la recherche en sciences de gestion. 2019.
- [21] Iman Saladin B. AZHAR Reza Firsandaya Malik Winda Kurnia Sari, dian Palupi Rini. Sequential models for text classification using recurrent neural network. 2020.
- [22] Gopikrishna Yadam. Model monitoring using tensor flow data validation in tfx. 2021.
- [23] Banque Zitouna. présentation de l'entreprise. "<https://www.banquezitouna.com/fr>".