République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : Technologies de l'Information et de la Communication (TIC)

THEME

La Catégorisation des Documents Ecrits en Langue Arabe

Présenté par :

MEHIRIS Hichem Abdelmalik

LEKBIR Selma

Soutenu publiquement le : 22/06/2023

Devant le jury composé de:

Président : ……..

Examinateur : ……

Encadreur : Nouioua Mourad

2022/2023

Dédicace

Je dédie ce travail

A mes parents, qui ont été ma source inépuisable de soutien, d'encouragement et d'amour tout au long de mon parcours académique.

Leur soutien indéfectible et leurs sacrifices ont été les fondements de ma réussite.

Je leur suis profondément reconnaissant pour leur confiance en moi et pour leur constante inspiration.

Cette réalisation leur est dédiée avec une gratitude éternelle.

A mes Frères et mes Sœurs,

***Abdou**, **Moussa**, **Oussama**, **Soumia** et **Fatima**.*

A tous mes amis et mes camarades qui m'ont toujours encouragé,

et à qui je souhaite plus de succès.

*Sans oublier mon binôme **Lekbir Selma**, pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.*

Hichem

Dédicace

Je dédie ce travail

À mes chers parents,

Je vous adresse ma gratitude la plus profonde pour vos sacrifices inestimables, vous êtes ma source d'inspiration.

À ma précieuse sœur ***Sabrina****, mes chers frères* ***Haithem*** *et* ***Zinou,***

Et toute ma famille,

Je vous remercie du fond du cœur pour vos encouragements et votre soutien constant.

À mes chères amies, ***Mouna****,* ***Mebarka*** *et* ***Wafa****,*

Votre amitié précieuse ont rendu ce parcours plus joyeux et enrichissant.

Je tiens également à remercier mon binôme, ***Hichem****,*

Merci pour ton dévouement et ton professionnalisme. Ta contribution a été inestimable.

À tous ceux qui m'ont soutenu de près ou de loin,

Je vous adresse mes remerciements les plus chaleureux.

Selma

Remerciement

Tout d'abord, nous tenons à remercier Allah, le Tout-Puissant, pour nous avoir accordé la force, la patience et la sagesse nécessaires pour mener à bien ce travail.

Nous aimerions également adresser nos remerciements à notre encadrant ***Dr. NOUIOUA Mourad****. Nous le remercions pour son aide, sa disponibilité, pour le temps qu'il nous a consacré, pour sa supervision éclairée tout au long de la rédaction du mémoire.*

Nous remercions les membres de jury d'avoir accepté de juger notre travail.

Nous sommes reconnaissants à tous nos professeurs pour tous leurs efforts et leur collaboration tout au long de notre cycle d'étude.

Nous tenons à exprimer notre gratitude envers nos familles et surtout nos parents pour leur amour, leurs conseils ainsi que leur soutien inconditionnel.

Enfin, nous remercions les amis et collègues qui nous ont apporté leur soutien moral tout au long de notre démarche.

Hichem MEHIRIS

Selma LEKBIR

Résumé

La quantité de données textuelles arabes disponibles sur le World Wide Web a considérablement augmenté au cours des deux dernières décennies, ce qui en fait la quatrième langue la plus couramment utilisée sur le Web. Par conséquent, il existe un besoin croissant d'une classification efficace des textes arabe, en particulier pour le filtrage de contenu Web, la récupération d'informations et la détection des spams par e-mail. Plusieurs algorithmes d'apprentissage automatique ont été implémentés pour classer les documents arabes. Cependant, les résultats obtenus ne sont pas comparables à ceux obtenus dans d'autres langues telles que l'anglais.

Ce travail étudie l'impact de techniques de prétraitement et d'extraction de caractéristiques judicieusement choisies sur l'efficacité de différents algorithmes de classification des textes. Toutes les combinaisons possibles de ces techniques sont essayées. Les résultats rapportés démontrent le grand impact des techniques de prétraitement sur l'efficacité de classification des textes Arabes, en particulier la suppression des mots vides avec les techniques d'extraction de caractéristiques.

Mots clés: Classification des Textes Arabes, Apprentissage Automatique, Techniques de Prétraitement des données, Extraction de Caractéristiques.

.

Abstract

The amount of Arabic text data available on the World Wide Web has increased dramatically over the past two decades, making it the fourth most commonly used language on the Web. Therefore, there is a growing need for effective Arabic text classification, especially for web content filtering, information retrieval, and email spam detection. Several machine learning algorithms have been implemented to classify Arabic documents. However, the results obtained are not comparable to those obtained in other languages such as English.

This work studies the impact of judiciously chosen preprocessing and feature extraction techniques on the efficiency of different text classification algorithms. All possible combinations of these techniques are tried. The reported results demonstrate the great impact of pre-processing techniques on the efficiency of Arabic text classification, especially stopword removal with feature extraction techniques.

Keywords: Classification of Arabic Texts, Machine Learning, Data preprocessing techniques, Feature Extraction.

ملخص

في العقدين الماضيين ، تزايدت كمية البيانات النصية العربية المتاحة على شبكة الويب العالمية بشكل كبير، مما يجعلها رابع أكثر اللغات استخدامًا على الويب. لذلك، هناك حاجة متزايدة لتصنيف فعال للنصوص العربية ، وخاصة لتصفية محتوى الويب ، واسترجاع المعلومات، واكتشاف البريد الإلكتروني العشوائي. تم تنفيذ العديد من خوارزميات التعلم الآلي لتصنيف الوثائق العربية. ومع ذلك، فإن النتائج التي تم الحصول عليها لا يمكن مقارنتها مع تلك التي تم الحصول عليها بلغات أخرى مثل الإنجليزية.

يدرس هذا العمل تأثير تقنيات المعالجة المسبقة وتقنيات استخراج الميزات المختارة بحكمة على كفاءة خوارزميات تصنيف النص المختلفة. تمت تجربة جميع التوليفات الممكنة من هذه التقنيات. تظهر النتائج التي تم الإبلاغ عنها التأثير الكبير لتقنيات المعالجة المسبقة على كفاءة تصنيف النص العربي، وخاصة إزالة الكلمات الموقوفة بتقنيات استخراج الميزات.

الكلمات المفتاحية : تصنيف النصوص العربية ، التعلم الآلي ، تقنيات المعالجة المسبقة للبيانات ، استخراج الميزات.

Table des matières

Liste des abréviations

SVM	Les machines à vecteurs de support
NB	Naïve bayes
DT	Les arbres de décisions
KNN	K-plus proches voisins
TF	Term Frequency
IDF	Inverse document frequency
TF-IDF	Term Frequency x Inverse Document Frequency
CRM	Customer Relationship Management
Nr	Normalisation
St	Stemming
Lm	Lemmatisation
Tk	Tokenisation **&** Suppression des mots vides
ACC	Accuracy

Liste des figures

Liste des tableaux

Chapitre 1: Introduction Générale

1.1 La langue arabe

Récemment, il est devenu clair qu'il y a eu une augmentation significative du nombre d'internautes arabes, car la langue arabe est l'une des langues les plus répandues dans le monde, parlée par plus de 467 millions de personnes; et est considérée comme une langue sacrée dans l'islam en raison de son utilisation dans le Coran. Elle occupe la troisième place en termes de nombre de pays qui la reconnaissent comme langue officielle et la quatrième langue en termes de nombre d'utilisateurs sur Internet.

La langue arabe est connue pour sa complexité et sa richesse, avec une grammaire complexe et un vocabulaire étendu. La langue arabe se compose de 28 lettres qui peuvent être écrites dans plusieurs formes en fonction de leur position dans un mot. Cette variabilité de l'écriture arabe peut rendre difficile le traitement automatique de la langue

La croissance des données numériques arabes sur Internet a entraîné un énorme surplus difficile à gérer facilement. Des travaux ont donc commencé pour trouver des solutions afin d'organiser et de stocker ces informations et de trouver des moyens efficaces pour faciliter leur processus de récupération, ce qui a conduit à la nécessité de travailler sur la classification de ces informations.

La classification des textes arabes est l'un des sujets de recherche de grande importance dans le domaine de la fouille de textes afin que des informations de haute qualité soient extraites des textes et que les sujets auxquels appartiennent ces textes soient déterminés, en particulier lorsque ces textes sont volumineux en taille et ne peuvent pas être classés manuellement.

1.2 Motivation

La classification des textes est une problématique qui a fait l'objet de nombreuses études en langues étrangères, mais elle n'a pas retenu l'attention en langue arabe pour de nombreux facteurs représentés dans les difficultés d'analyse et de traitement de cette langue.

L'une des principales difficultés est la variabilité de l'écriture arabe, qui peut varier considérablement selon les régions et les contextes sociaux. Cette variabilité rend la reconnaissance des caractères arabes plus difficile que dans d'autres langues, puisqu'il faut considérer toutes les variantes possibles.

Un autre défi est la richesse de la langue arabe, qui a beaucoup de nuances et de subtilités dans le sens des mots et des phrases. Cela peut rendre la tâche d'analyse de texte beaucoup plus difficile, car il faut être capable de comprendre ces nuances pour interpréter correctement les données.

De plus, la complexité de la grammaire arabe rend difficile l'analyse automatique de texte à l'aide de techniques telles que l'analyse morphologique et syntaxique. Les règles de conjugaison et de déclinaison peuvent être complexes et les conjugaisons peuvent varier selon le temps, le sexe et le nombre, ce qui complique davantage l'analyse.

Enfin, lors de l'analyse de textes arabes, il est important de tenir compte des différences culturelles et du sens des mots et des phrases. Les mots et les expressions peuvent avoir différentes significations dans différents contextes culturels, ce qui nécessite une compréhension approfondie de la culture et des coutumes.

1.3 Objectif

L'objectif principal de cette étude est de déterminer la catégorie d'un texte arabe spécifique selon son contenu. Pour y parvenir, nous chercherons à extraire la meilleure combinaison appropriée pour la classification automatique des textes arabes en particulier, qui combine des techniques de pré-traitement, des méthodes d'extraction de caractéristiques avec des algorithmes

d'apprentissage supervisé les plus couramment utilisés et des méthodes ensemblistes. Et tout cela dans le but d'obtenir une classification plus précise.

1.4 Plan de mémoire

Ce mémoire va être organisé de la façon suivante :

- **Chapitre 02 : Catégorisation des textes arabe,** qui définit les différentes étapes du processus de catégorisation automatique des textes arabes.

- **Chapitre 03 : Méthodes supervisées et méthodes ensembliste,** Nous définissons les deux types d'apprentissage (supervisé et non supervisé), puis détaillons les méthodes de classification supervisée choisies ainsi que les méthodes d'apprentissage d'ensemble.

- **Chapitre 04 : Implémentation et Résultats,** montre le côté implémentation de notre application, l'approche que nous avons suivie dans la classification des textes arabes et l'analyse des résultats obtenus à partir des expérimentations que nous avons effectuées.

- **Enfin, nous concluons** ce mémoire en résumant les contributions que nous avons pu apporter et les perspectives de recherche dans le domaine.

Chapitre 2: Catégorisation des textes

2.1 Introduction

La catégorisation des textes est une étape fondamentale dans le domaine du traitement automatique du langage naturel. Elle permet d'organiser et de classer efficacement un grand volume de documents textuels en fonction de leurs caractéristiques et de leur contenu. Dans ce chapitre, nous explorerons en détail le processus de catégorisation des textes et ses différentes étapes.

2.2 Définition de la catégorisation des textes

- **Définition (1)**

  La catégorisation de textes (C.T) consiste à trouver des liens fonctionnels entre un ensemble de textes et un ensemble de catégories (étiquettes). [1]

- **Définition (2)**

  La catégorisation des textes est le processus d'association d'une valeur booléenne à chaque paire $(d_j, c_i) \in D \times C$, où D est l'ensemble des textes et C'est l'ensemble des catégories. Une valeur V(True) est associée au couple (d_j, c_i) si le texte dj appartient à la catégorie ci, sinon une valeur F(False) est associée. [1]

2.3 Le Processus de Catégorisation des Textes

Le processus reçoit en entrée un document textuel afin de lui trouver sa catégorie. La figure ci-dessous représente ce processus :

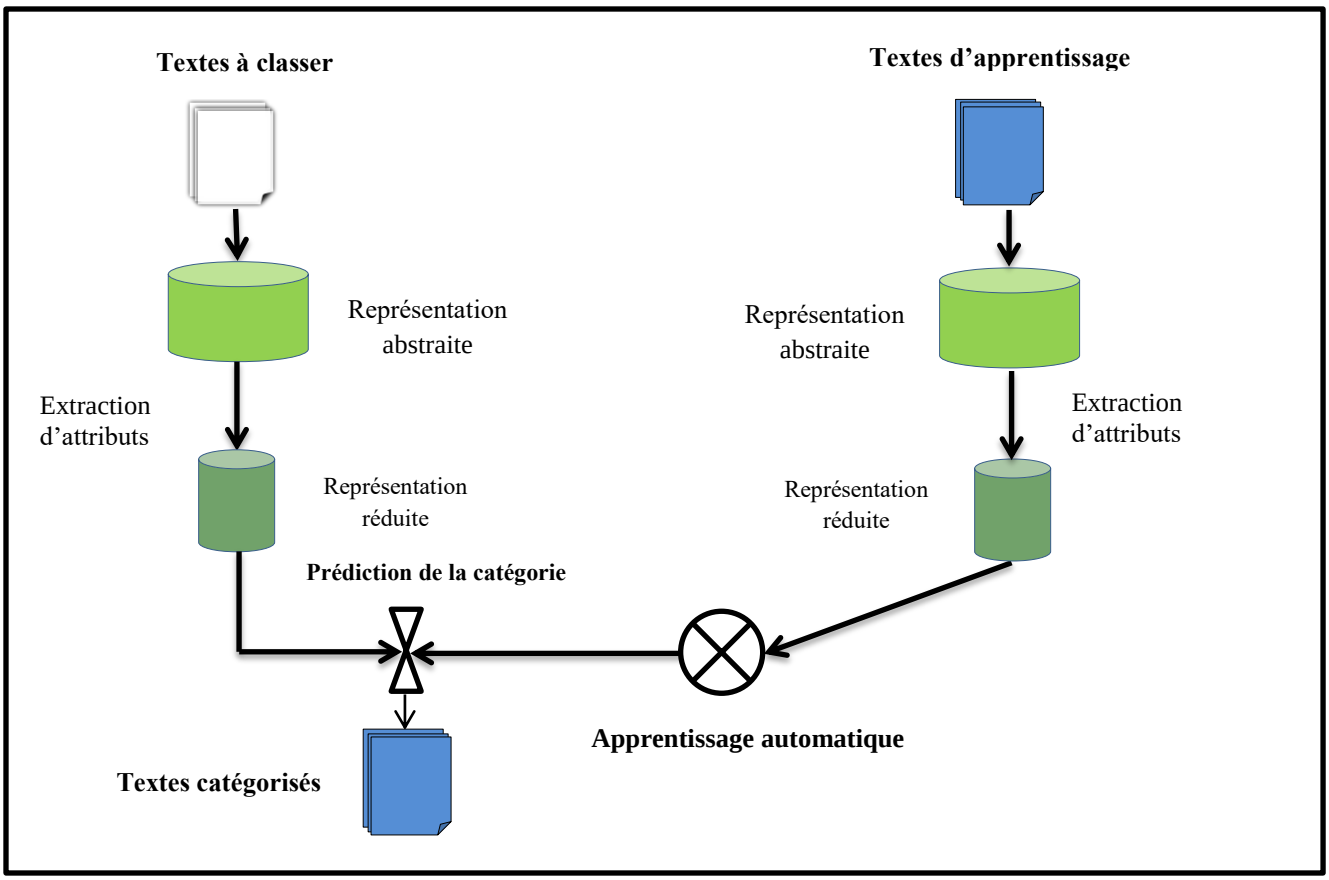

Figure 2.1 Processus de catégorisation des textes. [2]

Le but de la classification des textes est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu. Pour identifier la catégorie d'un texte, un ensemble d'étapes sont suivies :

- Pré-traitement.
- La représentation des textes.
- Pondération des termes.
- Réduction de dimensionnalité.
- Choix de classificateur.
- Evaluation des performances des modèles.

2.4 Pré-Traitement

Le prétraitement des textes est une phase capitale du processus de classification. En fait, dans les documents textuels, des nombreux mots apportent peu d'informations sur le document concerné. Pour cela, nous avons procédé par les étapes suivantes :

2.4.1 Suppression des caractères inutiles

Cette tâche est composée de deux opérations : la suppression des signes de ponctuation et la suppression des nombres et caractères latins.

a) Les signes de ponctuation

On supprime toute séquence de caractères de ponctuation séparés par des lettres ou des espaces, tels que des virgules et des points-virgules.

b) Les nombres et les caractères latins

Nous éliminons toutes les séquences de caractères comprises entre deux espaces et contenant des chiffres, et nous supprimons également les caractères latins.

2.4.2 Suppression des mots vide (Stop-words removal)

Cette opération consiste à enlever les mots qui appartenant à une liste prédéfinie qui contient les mots vide. Cette liste est appelée : La liste des mots vide. Les mots vides sont généralement des mots dépourvus de sens tels que des déterminants, des prépositions, des conjonctions, des adverbes, des adjectifs indéfinis, des pronoms, etc.

Pour l'arabe, la liste des mots vides comprend des pronoms comme (انا، نحن، هو، هي، هما ...), des adverbes (... فوق، تحت، منذ، الآن), des conjonctions (... من، إلى، مع، في، و) et des articles (... ثم، إذا، عدا). Ces mots sont inutiles pour distinguer les différentes catégories des textes. Ils peuvent donc être supprimés sans perdre d'informations utiles.

Leur élimination lors du prétraitement du document permet ensuite de gagner beaucoup de temps lors de la modélisation et de l'analyse du document. [3]

2.4.3 Tokenisation

Le principe de tokenisation est de décomposer un long texte en phrases ou en mots appelés « tokens » ou « jetons ». Elle consiste à découper les séquences de caractères en fonction de la présence ou l'absence de séparateurs (« espace » ou « retour à la ligne »). [4]

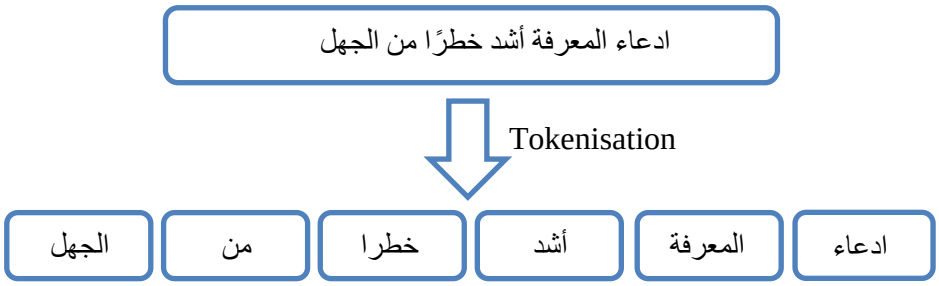

Figure 2.2 Exemple de « Tokenisation ».

2.4.4 Normalisation

La langue arabe connait des grandes variations de représentation textuelle, pour tenir compte les spécifications de cette langue et de remédier au problème de variation de représentation des caractères arabes dans les textes nous avons appliqué quelques méthodes de normalisation sur le corpus :

- Remplacer El Hamzah (آ، إ، أ) par (ا), car la plupart des textes arabes négligent l'ajout d'El hamza sur El Alif.
- Remplacer (ى) par (ي) à la fin des mots.
- Remplacer (ة) par (ه) encore à la fin des mots.
- Remplacer la séquence (ئ) par (ي).
- Eliminer les diacritiques (voyelles) et « Chedda ».

2.4.5 Racinisation (Stemming)

L'un des processus les plus importants pour classer les documents est le *stemming*. Le but est de générer des racines de mots (radical, stem en Anglais).[4]

Mot	Stem
المواصلات	وصل
بلادي	بلد
اخبارنا	خبر

Tableau 2.1 Exemple de « *Stemming* ».

2.4.6 Lemmatisation

La lemmatisation consiste à regrouper des mots avec des racines différentes mais ont le même sens. Il s'agit d'une analyse lexicale qui permet, par exemple, de placer les verbes à l'infinitif, aux formes singulières, de déterminer le genre (masculin/féminin) des noms, etc. [4]

2.5 La Représentation des Textes

La représentation des textes est une phase très importante dans le processus de catégorisation de textes car les algorithmes d'apprentissage automatique ne peuvent pas être utilisés directement pour classifier les textes. Pour cela, il est nécessaire d'utiliser une technique de représentation efficace permettant de représenter les textes sous une forme exploitable par ces algorithmes.

La représentation la plus couramment utilisée est celle du modèle vectoriel dans laquelle chaque texte est représenté par un vecteur de n termes pondérés. [5]

Parmi les différentes méthodes qui existent pour représenter les textes, on trouve :

2.5.1 Représentation par sac de mots

Cette méthode consiste à représenter les documents sous forme de vecteurs de mots. Le processus de conversion du texte d'un document en un ensemble de termes est appelé l'analyse lexicale. L'analyse lexicale permet d'identifier les espaces de séparation des mots, les signes de ponctuation, les chiffres, etc. L'avantage de cette représentation est qu'elle exclut toute analyse grammaticale et toute notion de distance entre les mots, mais l'inconvénient est qu'il est difficile de délimiter les mots dans certaines langues (comme l'arabe) [2]. Le tableau ci-dessous représente un exemple de représentation par sac de mots.

تجتمع مجتمعات انترنت الأشياء والروبوتات معًا لإنشاء انترنت الأشياء الروبوتية. إن انترنت الأشياء الروبوتية هو مفهوم يمكن من خلاله لأجهزة الاستشعار الذكية مراقبة الأحداث التي تحدث من حولهم ودمج بيانات المستشعر الخاصة بهم.

Mots	Occurrence	Mots	Occurrence
أشياء	3	روبوتية	2
مجتمعات	1	إن	1
روبوتات	1	استشعار	1
مفهوم	1	مراقبة	1
انترنت	3	ذكية	1
إنشاء	1	أحداث	1
و	2	بيانات	1
تحدث	1	أجهزة	1
هو	1	مستشعر	1
خاصة	1	خلال	1
يمكن	1	تجتمع	1
دمج	1	من	2
التي	1	معا	1

Tableau 2.2 La représentation de texte en « Sac de mots ».

2.5.2 Représentation par phrases

Certains chercheurs proposent d'utiliser les phrases comme unité de représentation au lieu des mots, car les phrases sont plus informatives que les mots seuls, elles conservent l'information

relative à la position du mot dans la phrase, et ont un degré d'ambiguïté inférieur à celui des mots constitutifs. [5]

2.5.3 Représentation par N-grammes

La représentation par N-grammes consiste à découper le texte en séquences de n caractères en déplaçant une fenêtre d'un caractère. Ce processus de découpage est peut-être aussi appliqué au niveau des mots et pas au niveau des caractères (Voire la **Figure 2.3**). Cette méthode permet de capturer automatiquement les racines des mots les plus courants sans passer par une phase de recherche de racine lexicale, indépendamment de la langue, les espaces sont pris en compte, car en fait, ne pas les prendre en compte va introduire du bruit. [2]

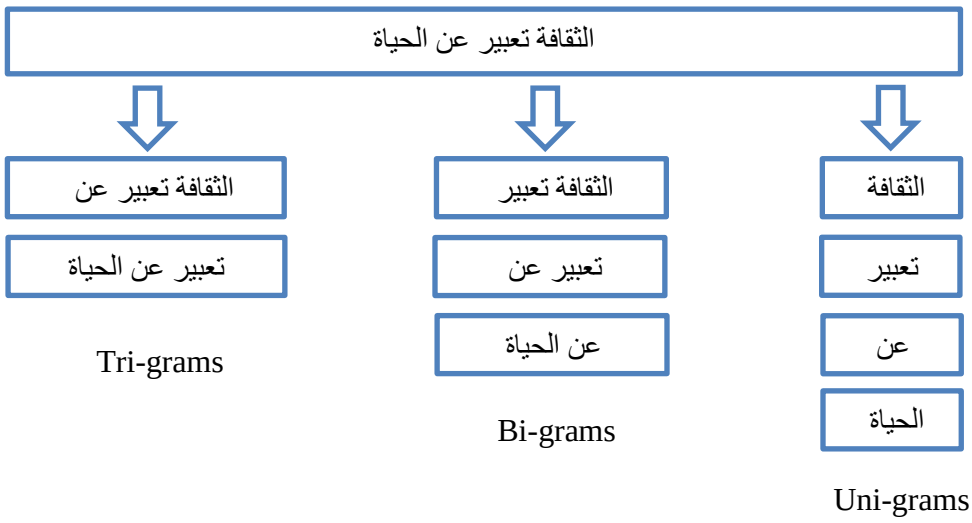

Figure 2.3 Exemple de « N-grammes de mots ».

2.6 Pondération des Termes

La phase de pondération des termes une phase complémentaire de la phase précédente. Elle permet d'évaluer l'importance des termes contenus dans un document, relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence des mots dans le corpus. [6]

Le but de cette pondération est de mieux utiliser les informations contenues dans le document pour améliorer les performances du système de classification de textes.

Il existe plusieurs systèmes de pondération comme la pondération booléenne, pondération par fréquence de mots et pondération par TF-IDF.

2.6.1 Pondération booléenne

La pondération booléenne consiste à simplement utiliser une attribution binaire aux termes. Plus précisément, si le terme est présent une ou plusieurs fois dans le document, on l'attribue un **1**, sinon **0** est attribué à ce terme dans le cas contraire. Cette représentation est peu informative car elle ne donne pas les informations nécessaires sur les occurrences d'un terme dans le document qui peut être une information importante pour l'opération de classification.

2.6.2 Pondération par fréquence de mot

Contrairement avec la méthode précédente, cette méthode pris en considération la fréquence d'apparition des mots. Elle consiste à présenter le texte sous forme des vecteurs dont les éléments renseignent non seulement sur la présence ou l'absence d'un terme mais aussi informe sur le nombre de présences du terme dans le texte.

2.6.3 Pondération TF-IDF

Le principe est de coder chaque élément par un scalaire appelé TFIDF pour donner un aspect mathématique aux documents textes. Cette représentation donne plus de poids aux termes qui apparaissent avec une haute fréquence dans peu de documents. [7]

Pour calculer le poids, on utilise les mesures suivantes :

- **Term frequency (TF) :** Un terme qui apparait plusieurs fois dans un document est plus important qu'un terme qui apparaît une seule fois.
- **Inverse document frequency (IDF) :** Un terme qui apparaît dans peu de documents est un meilleur discriminant qu'un terme qui apparaît dans tous les documents.

$$\textbf{IDF} = \log [N / Df]$$

- **Df :** nombre de documents contenant le terme.
- **N :** nombre total de documents du corpus.

- **TF-IDF (Term Frequency x Inverse Document Frequency):** TF-IDF permet de mesurer l'importance d'un terme dans un document relativement à l'ensemble des documents.
  - **TFIDF = TF** [t,d] * log [N / Df[t]]
  - **TF [t,d]** : fréquence du terme t dans le document d.

Comme mentionné précédemment, le modèle vectoriel est le plus utilisé pour la représentation de texte, qui permet une analyse très efficace de grandes collections de documents. Il a une structure de données simple, sans utiliser aucune information sémantique explicite.

2.7 Réduction de Dimensionnalité

Les méthodes de réduction de dimensionnalité permettent de projeter des données issues d'un espace de grande dimension en un espace de plus petite dimension. La réduction de dimensionnalité permet de réduire la complexité des problèmes d'apprentissage automatique, ce qui simplifie la résolution des problèmes d'optimisation associés en réduisant l'espace des solutions. En d'autres termes, la réduction de la dimensionnalité limite le nombre de possibilités à tester, permettant un traitement plus rapide des données. [8]

Pour réduire la dimensionnalité, on utilise deux catégories de méthodes : les méthodes de sélection d'attributs (*feature selection methods*) et les méthodes d'extraction d'attributs *(feature extraction methods).*

Les méthodes de sélection d'attributs servent à réduire la taille de l'espace d'apprentissage en sélectionnant uniquement un sous-ensemble des attributs existants. Le sous ensemble ne contient que les attributs jugés pertinents par l'algorithme de sélection d'attributs. [5]

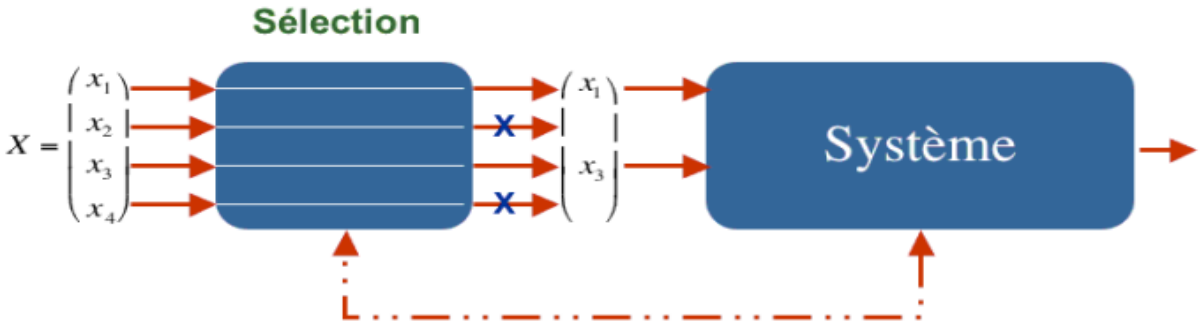

Figure 2.4. Sélection d'attributs.

Contrairement aux techniques de sélection d'attributs, les méthodes d'extraction d'attributs ont pour objectif de proposer, via une synthétisation, un sous-ensemble de nouveaux attributs à partir des attributs existants. [5] Les **Figure 2.4-Figure 2.5** illustrent les deux types de méthodes utilisées pour la réduction de dimensionnalité.

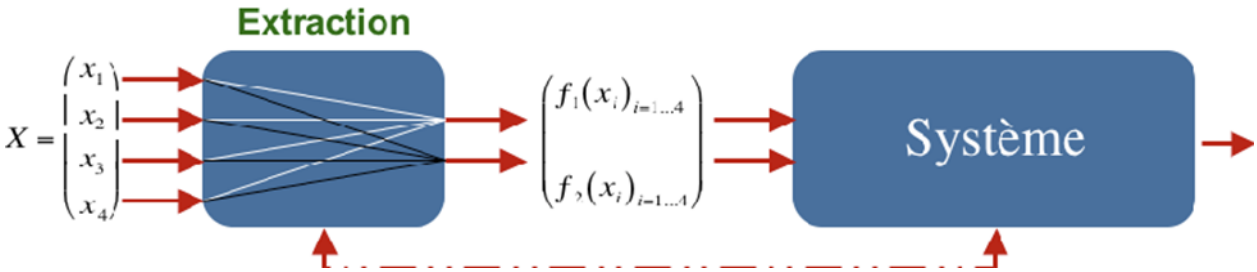

Figure 2.5. L'extraction d'attributs.

2.8 Classification des Textes

La classification des textes consiste à choisir la technique d'apprentissage (classificateur). Parmi les méthodes d'apprentissage les plus souvent utilisées : "Les arbres de décision", "Algorithme Naïve Bayes", "Machines à Vecteur de Support", "Algorithme des K-plus proches voisins". Le choix du classificateur se fait en fonction de l'objectif final à atteindre. Pour voire les détails de la phase de classification de texte, le lecteur pourra se référer au chapitre suivant où nous nous allons donner une présentation détaillée pour chaque une de ces méthodes.

2.9 Evaluation des Performances des Modèles

Dans le domaine de la recherche d'information et de l'apprentissage automatique, la validation des méthodes est généralement empirique. En utilisant des mesures telles que la précision, le rappel, la F-mesure, l'exactitude, la mesure de similarité, la courbe ROC et la matrice de confusion. Ces mesures nous permettent de quantifier la précision et l'exhaustivité de la classification. Une évaluation rigoureuse est essentielle pour garantir que le modèle est fiable et adapté aux besoins spécifiques de la classification des textes. Les mesures de performance utilisée dans notre étude seront également mentionnées dans le Chapitre 4:.

2.10 Conclusion

Ce chapitre a fourni une exploration approfondie des différentes étapes du processus de classification des textes, en mettant en évidence les techniques clés et les considérations à chaque étape. La compréhension de ces éléments est essentielle pour développer des modèles de classification efficace.

Chapitre 3: Méthodes Supervisées et Méthodes Ensemblistes

3.1 Introduction

Dans ce chapitre, nous allons exposer les concepts de l'apprentissage automatique. Plus précisément, nous présentons les deux types d'apprentissage automatique : l'apprentissage supervisé et l'apprentissage non supervisé. Nous nous concentrerons sur les méthodes de classification supervisée ainsi que sur les méthodes ensemblistes utilisée dans cette étude.

3.2 L'apprentissage Automatique (Machine Learning)

3.2.1 Définition

L'apprentissage automatique, ou *Machine Learning,* est une science moderne permettant de découvrir des répétitions (des patterns) dans un ou plusieurs flux de données et d'en tirer des prédictions en se basant sur des statistiques. En clair, Machine Learning se base sur le forage de données, permettant la reconnaissance de patterns pour fournir des analyses prédictives. Les premiers algorithmes de Machine Learning ne datent pas d'hier, puisque certains ont été conçus dès 1950, le plus connu d'entre eux étant le Perceptron.

Le Machine Learning révèle tout son potentiel dans les situations où des insights (tendances) doivent être repérés à partir de vastes ensembles de données diverses et variées, appelés le Big Data. [9]

Pour analyser de tels volumes de données, Machine Learning se révèle bien plus efficace en termes de vitesse et de précisions que les autres méthodologies traditionnelles. À titre d'exemple, Machine Learning est capable de déceler une fraude en une milliseconde, rien qu'en se basant sur des données issues d'une transaction (montant, localisation…), ainsi que sur d'autres informations historiques et sociales qui lui sont rattachées. En ce qui concerne l'analyse de

données transactionnelles, de données issues de plateformes CRM ou bien des réseaux sociaux, là encore le Machine Learning se révèle désormais indispensable. [9]

Le Machine Learning est réellement la science idéale pour tirer profit du Big Data et de ses opportunités. Cette technologie est en effet capable d'extraire les données de valeur parmi d'immenses sources d'informations complexes, et ce sans avoir à faire appel aux humains. Entièrement dirigé par les données, le Machine Learning convient donc parfaitement à la complexité du Big Data, dont il est réellement indissociable. Là où les outils analytiques traditionnels se heurtent bien souvent à un volume maximal de données pouvant être analysées, le Machine Learning révèle au contraire tout son potentiel lorsque les sources de données sont croissantes, lui permettant d'apprendre et d'affiner des insights avec une précision toujours améliorée. En clair, plus les données sont nombreuses, plus les ordinateurs dotés de Machine Learning sont puissants et peuvent découvrir des patterns enfouis dans ces données avec nettement plus d'efficacité que ne le ferait l'intelligence humaine. [9]

La **Figure 3.1** montre qu'il existe fondamentalement quatre approches d'apprentissage automatique : Apprentissage supervisé, semi-supervisé, non-supervisé et renforcé. Les algorithmes d'apprentissage supervisé peuvent être divisés en deux catégories selon le problème à résoudre : les algorithmes de classification et les algorithmes de régression.

Dans notre étude, nous utilisons l'apprentissage supervisé pour construire des modèles pour la catégorisation des textes écrits en langue Arabe en utilisant les algorithmes de classification.

Dans la suite de ce chapitre, nous allons présenter l'apprentissage supervisé en illustrant la différence entre les méthodes de régression et de classification. Nous allons aussi présenter en détails les algorithmes de classification supervisée que nous avons adoptées dans notre étude.

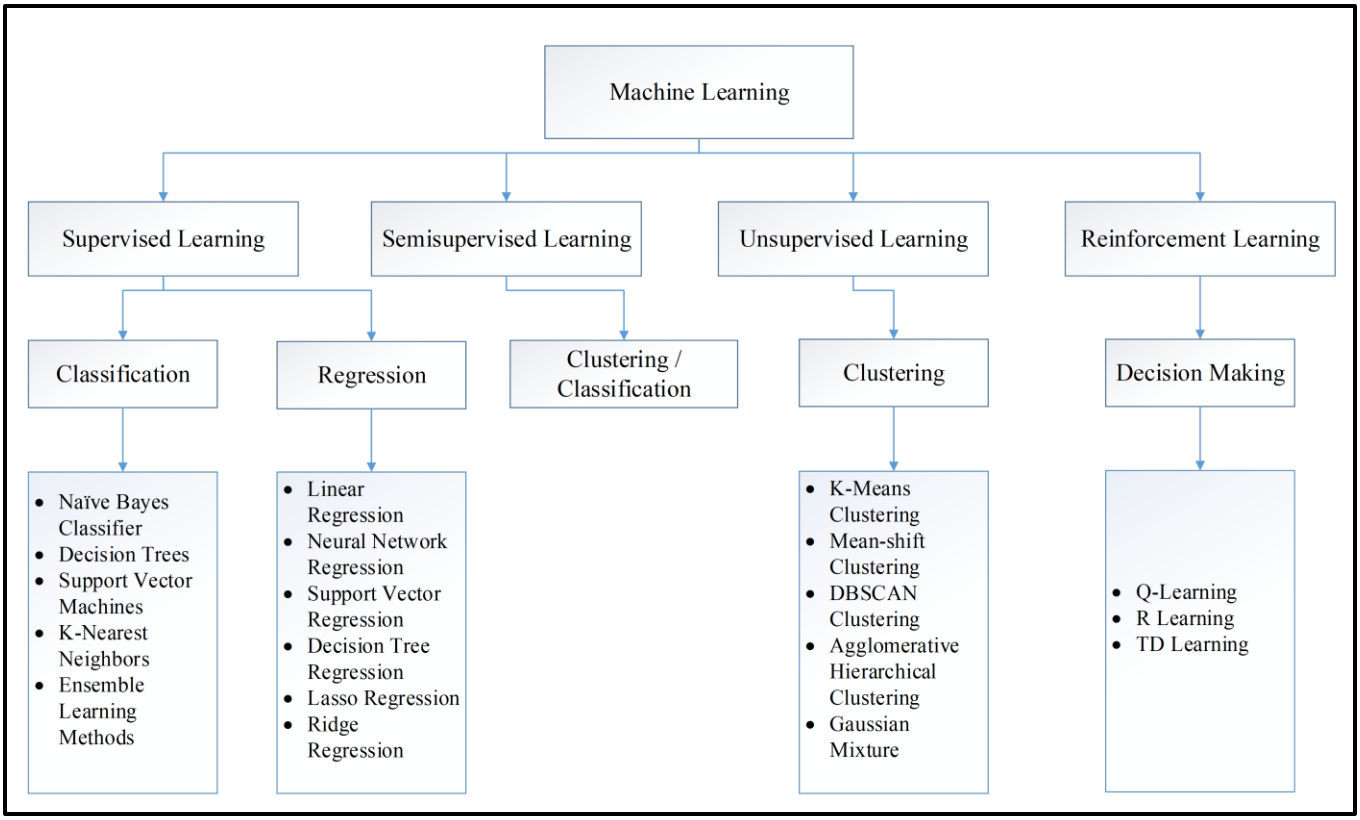

Figure 3.1 Types d'apprentissage automatique.

3.3 L'apprentissage Supervisé

L'apprentissage supervisé est une méthode d'apprentissage automatique, caractérisée par la création d'un algorithme qui apprend une fonction prédictive. Ceci est possible grâce à un entrainement à partir d'exemples annotés (étiquetés), qui inclus un groupe de variables d'entrée, accompagnées de leurs variables de sortie respectives. Ce processus d'entrainement est répété jusqu'à l'obtention d'une performance satisfaisante. Lors de chaque itération, la machine crée un certain nombre de règles, reliant les variables d'entrée aux variables de sortie. Ce processus permet au modèle d'apprendre à partir des données et d'appliquer les règles afin prédire, de façon précise, la valeur de sortie lorsqu'une valeur d'entrée est donnée. [10]

L'apprentissage supervisé peut être divisé en deux sous-catégories : la classification et la régression.

3.3.1 Classification

Les problèmes de classification utilisent un algorithme pour affecter avec précision des données de test à des catégories spécifiques. Autrement dit, la classification est un processus de recherche d'une fonction qui aide à diviser l'ensemble de données en classes basées sur différents paramètres (variables d'entrée). La tâche de l'algorithme de classification est de trouver la fonction de mappage pour mapper l'entrée (x) à la sortie discrète (y).

Les algorithmes de classification peuvent être utilisés pour classer les spams dans un dossier distinct de sa boîte de réception par exemple. Les classifieurs linéaires, les machines à vecteurs de support, les arbres de décision et les forêts d'arbres décisionnels sont tous des types courants d'algorithmes de classification. [11]

3.3.2 Régression

Contrairement aux algorithmes de classification, les algorithmes de régression servent à prédire une valeur numérique continue à partir de variables d'entrée. La tâche de l'algorithme de régression est de trouver la fonction de mappage pour mapper la variable d'entrée (x) à la variable de sortie continue (y). En monde réel, on a plusieurs problèmes de régression telles que la prédiction des tendances du marché, la prédiction des prix des maisons, etc. [12]

3.4 Les Méthodes de Classification Supervisée

3.4.1 Les arbres de décision

Un arbre de décision est un outil de modélisation prédictive qui peut être utilisé aussi bien pour la régression que pour la classification. Dans le cas d'un arbre décisionnel de classification, la variable de décision est une catégorie, alors que dans le cas d'un arbre décisionnel de régression, la variable de décision est continue. Les arbres de décision sont probablement l'un des algorithmes de machine learning les plus compréhensibles et peuvent être interprétés de manière relativement simple. Ces arbres peuvent être construits à travers une approche algorithmique qui sépare les données, dépendant de différentes conditions. Dans un arbre, les feuilles, en anglais *nodes*,

représentent les valeurs de la variable-cible et les embranchements correspondent à une combinaison de variables d'entrée menant à ces valeurs. [13]

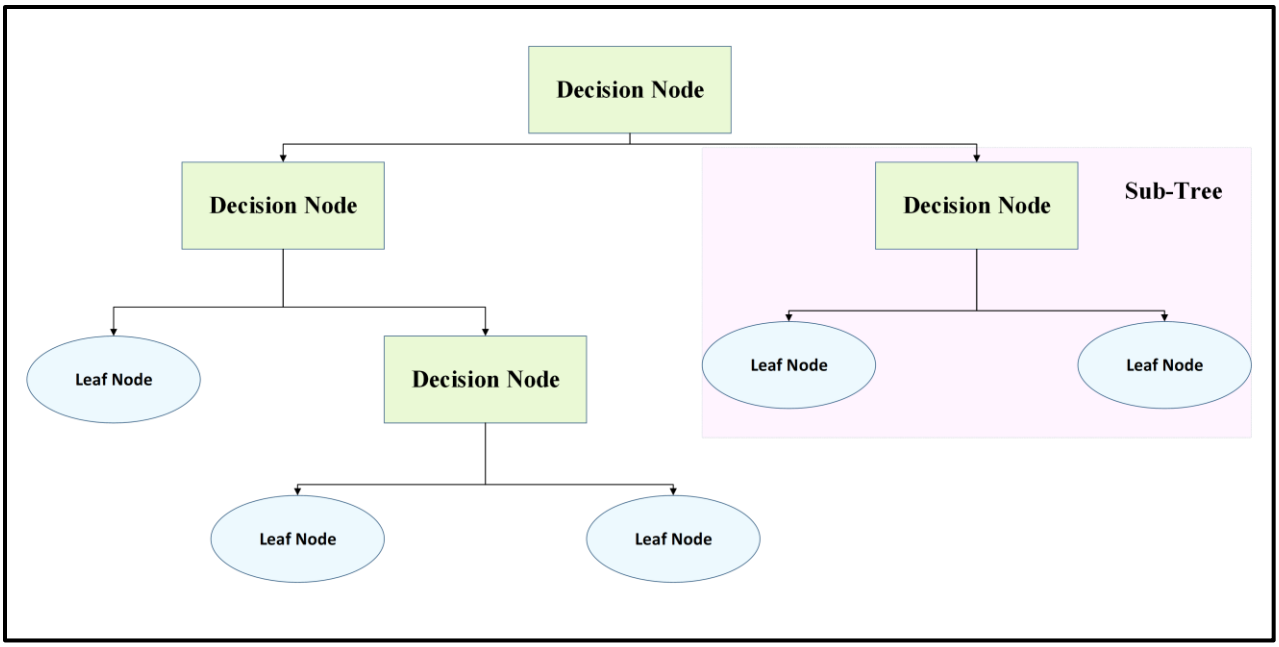

Figure 3.2 Arbre de décision.

a) Etapes de construction d'un arbre de décision

La création d'un arbre décisionnel nécessite les étapes suivantes :

1. Choisir une variable de prédiction qui classifie les données de la meilleure manière et qui attribue chaque point à la première feuille.
2. Descendre le long de l'arbre, en partant de la première feuille, tout en prenant des décisions par rapport à la classification de chaque point
3. Retourner à l'étape 1 et répéter ce même processus jusqu'à ce que chaque point soit classifié.

Afin d'évaluer les séparations au sein d'un arbre, nous pouvons utiliser le coefficient de Gini. Cette fonction peut prendre en compte des variables discrètes et il est calculé de la manière suivante:

$$Gini = 1 - \sum_{i=1}^{n} p_i^2$$

Où p_i correspond à la probabilité d'un élément d'être classifier dans une classe particulière.

Lors de la mise en place d'un arbre, notre but est de choisir les variables avec un coefficient de Gini le plus proche de celui de la première feuille.

Une autre méthode pour séparer les données est l'information gain. En effet, cette méthode est utilisée pour déterminer quelle variable nous donne le maximum d'information sur une classe. Elle utilise le concept d'entropie, qui mesure l'incertitude dans les données et est utilisé pour déterminer comment diviser les données dans un arbre décisionnel. Les formules pour ces deux mesures sont les suivantes :

$$Entropy = \sum_{i=1}^{n} -p_i \, log_2 p_i$$

$$Information\ gain = entropy(parent) - (weighted\ average) * entropy(children)$$ [13]

3.4.2 Algorithme de Naïve Bayes

La classification naïve bayésienne est une méthode d'apprentissage supervisé basée sur le théorème de Bayes et l'hypothèse que toutes les variables explicatives sont indépendantes les unes des autres. En d'autres termes, l'hypothèse signifie que l'existence d'une variable dans une classe est indépendante de l'existence d'une autre variable dans la même classe. Par conséquent, toute modification de l'une des variables ne doit pas affecter directement la valeur de l'autre variable dans le modèle. [14]

La classification naïve bayésienne nécessite l'entrainement du modèle, comme pour tout algorithme d'apprentissage supervisé, afin d'estimer les paramètres nécessaires à la classification. L'intérêt d'utiliser cette méthode est de trouver la probabilité d'une classe, ou d'une étiquette, en fonction de certains paramètres. On appelle ceci la probabilité postérieure, qui peut être calculée avec la formule suivante :

$$P(X|Y) = \frac{P(X) * P(Y|X)}{P(Y)}$$

Où :

- Y est l'ensemble des paramètres et X est l'ensemble des observations.
- P(X|Y) est la probabilité postérieure d'une classe et est donné par : $P(X|Y) = P(X \cap Y)/P(Y)$.
- P(X) est la probabilité antérieure d'une classe.
- P(Y|X) est la probabilité d'un indicateur, connaissant la classe. Elle est donnée par :

  $P(Y|X) = P(X \cap Y)/P(X)$
- P(Y) est la probabilité antérieure d'une variable dépendante.

Le modèle naïf bayésien est généralement utilisé pour la classification de documents, le filtrage des spam et les prédictions. Il existe différentes versions du modèle naïve bayésien, comme la Gaussian Naïve Bayes, Bernoulli Naïve Bayes, et Multinomial Naïve Bayes. [14]

3.4.3 Machines à vecteur de Support (SVM)

Les machines à vecteurs de support (SVM) sont des modèles d'apprentissage automatique supervisés qui se concentrent sur la résolution de problèmes mathématiques de discrimination et de régression. Ils ont été conceptualisés dans les années 1990 à partir d'une théorie de l'apprentissage statistique développée par les informaticiens russes *Vladimir Vapnik* et *Alexey Chervonenkis* : la théorie de *Vapnik-Chervonenkis.* En raison de la capacité du modèle à traiter des données de grande dimension, de ses garanties théoriques et des bons résultats obtenus en pratique, il a été rapidement adopté. SVM nécessite un petit nombre de paramètres et est apprécié pour sa facilité d'utilisation. [14]

a) Principe de SVM

Le principe de SVM est de réduire le problème de classification ou de discrimination à un hyperplan (espace des caractéristiques) dans lequel les données sont divisées en plusieurs classes, dont les frontières sont aussi éloignées que possible des points de données (ou "marge maximale"). D'où un autre nom pour les machines à vecteurs de support : les séparateurs à large marge.

La notion de bornes implique que les données sont linéairement séparables. Pour ce faire, les SVM utilisent des noyaux, qui sont des fonctions mathématiques pour projeter et séparer des données dans un espace vectoriel, les « vecteurs de support » étant les données les plus proches de la frontière. C'est la limite la plus éloignée de tous les points d'apprentissage qui est optimale et qui a donc la meilleure capacité de généralisation. [14].

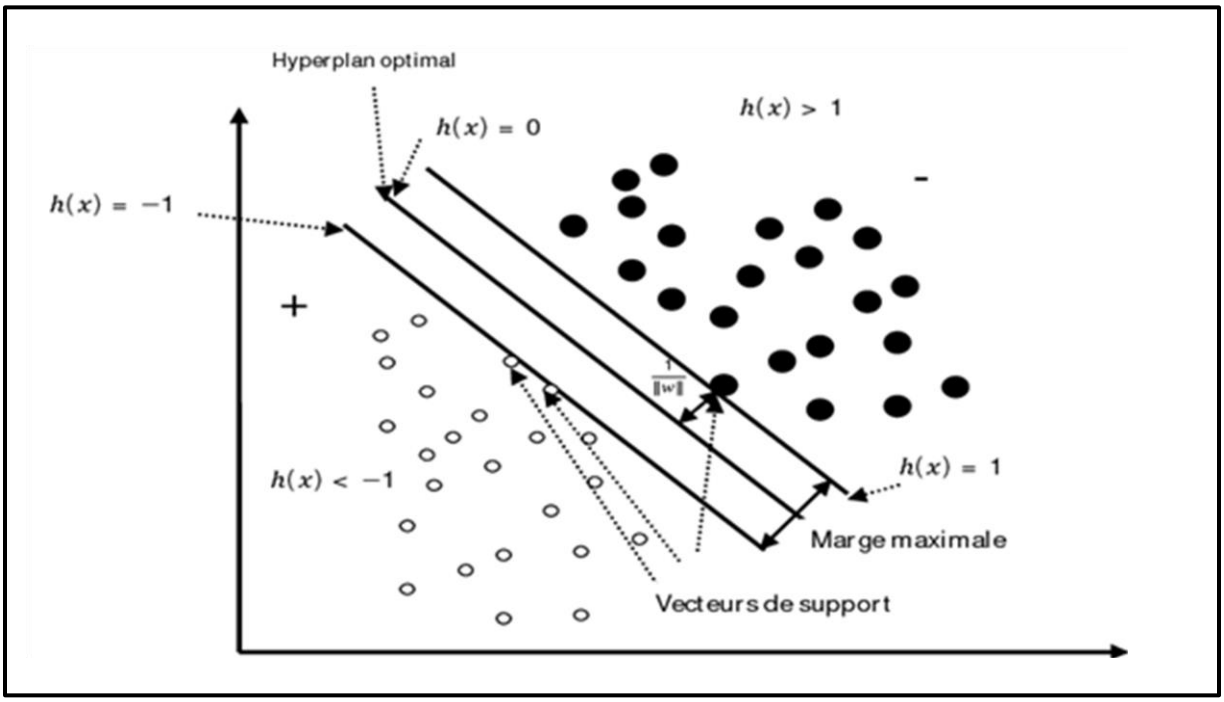

Figure 3.3 représentation de L'hyperplan séparateur optimal qui maximise la marge dans l'espace de re-description.

3.4.4 Algorithme des K-plus proches voisins (KNN)

K-Nearest Neighbors est l'un des algorithmes d'apprentissage supervisé qui peut être utilisé à la fois pour les problèmes de régression et de classification. Cependant, il est encore principalement utilisé pour des problèmes de classification. L'algorithme est formé sur des données étiquetées et tente de prédire la classe associée à la variable de test en calculant la distance entre la variable de test explicative et les points appris. En d'autres termes, cette méthode enregistre les points appris, puis classe les nouveaux points en fonction de leur similitude avec

les données enregistrées. Ci-dessous un exemple de méthode à trois voix. Le but est de déterminer à quelle classe appartient un point X. [14]

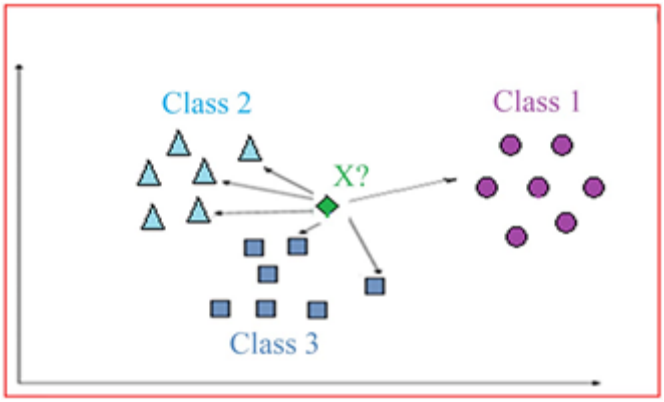

Figure 3.4 Exemple de la méthode K-plus proches voisins.

a) Principe de fonctionment de l'algorithm K-plus proche voisins

L'algorithme pour les modèles des K-plus proches voisins est le suivant :

1. Choisir le nombre K de voisins.
2. Calculer la distance Euclidienne des K voisins.
3. Prendre les K-plus proches voisins, dépendant du résultat du calcul de la distance.
4. Parmi ce K voisin, compter le nombre de points dans chaque classe.
5. Attribuer le nouveau point à la classe dont le nombre de voisins est maximal.
6. Le modèle est terminé.

La distance euclidienne peut être calculée de la manière suivante :

$$\|x_i - x_j\| = \sqrt{\sum_{k=1}^{p} |x_{ik} - x_{jk}|^2}$$

Où x_i et x_j sont des vecteurs avec p éléments. La distance euclidienne est un cas particulier de la distance L_q avec $q = 2$. [14]

3.5 L’Apprentissage Par Ensemble (Ensemble Learning Methods)

L’apprentissage supervisé peut être effectué en développant un modèle pour chaque méthode de classification. Une autre approche plus prometteuse est l'utilisation de méthodes d'apprentissage par ensemble. Dans le cas des modèles de base, un seul modèle est entraîné sur des données étiquetées à l'aide d'un algorithme ou d'une technique spécifique, comme les techniques que nous avons présentes dans la section suivante. La complexité des modèles de base peut varier, allant de modèles simples avec peu de paramètres à des modèles complexes avec un grand nombre de paramètres. Ces modèles effectuent des prédictions en se basant sur les motifs et les relations qu'ils ont appris lors de l'entraînement. [15]

En revanche, l'apprentissage par ensemble combine plusieurs modèles de base pour effectuer des prédictions ou prendre des décisions. Au lieu de se fier à un seul modèle, les méthodes d'apprentissage par ensemble utilisent un groupe de modèles de base. Chaque modèle de base peut être entraîné sur un sous-ensemble différent des données d'entraînement ou en utilisant des algorithmes ou des paramètres différents. Cela introduit une diversité dans les prédictions des modèles de base. L'objectif est d'exploiter la diversité et la sagesse collective de plusieurs modèles pour améliorer les performances et la robustesse globales du modèle d'ensemble.

Le cadre d'apprentissage d'ensemble qui en résulte est plus robuste que les modèles individuels qui composent l'ensemble, puisque l'assemblage réduit la variance des erreurs de prédiction. L'apprentissage d'ensemble tente d'obtenir des informations complémentaires à partir de ses différents modèles contributifs, c'est-à-dire qu'un cadre d'ensemble réussit lorsque les modèles contributifs sont statistiquement distincts. En d'autres termes, les modèles qui présentent des variations de performances lorsqu'ils sont évalués sur le même ensemble de données sont mieux adaptés pour former un ensemble. [16]

Nous verrons deux stratégies de l'apprentissage d'ensemble : l’apprentissage par changement de modèles et l’apprentissage par changement de données.

3.5.1 Changer les modèles (Voting ensemble learning methods)

Le principe de fonctionnement de ces méthodes est basé sur l'hypothèse qu'il peut y avoir des cas où différents modèles fonctionnent mieux sur certaines distributions dans l'ensemble de données, par exemple un modèle peut être bon pour différencier les chats des chiens mais pas si bon pour différencier les chiens des loups. D'autre part, un deuxième modèle peut distinguer avec précision les chiens et les loups, tout en produisant de mauvaises prédictions pour la catégorie "chat". Un ensemble de ces deux modèles peut tracer des limites de décision plus discriminatoires entre les trois classes de données [16].

Dans cette étude, nous utilisons la méthode de vote à la majorité (majority voting ensemble learning method)

***a)* La méthode de vote à la majorité (Majority voting ensemble learning method)**

La méthode de vote à la majorité est l'un des schémas d'ensemble les plus simple de la littérature. Dans cette méthode, un nombre de classificateurs contributeurs est choisi, et pour chaque échantillon, les prédictions des classificateurs sont calculées. Ensuite, comme son nom l'indique, la classe qui obtient la majeure partie de la classe du pool de classificateurs est considérée comme la classe prédite de l'ensemble.

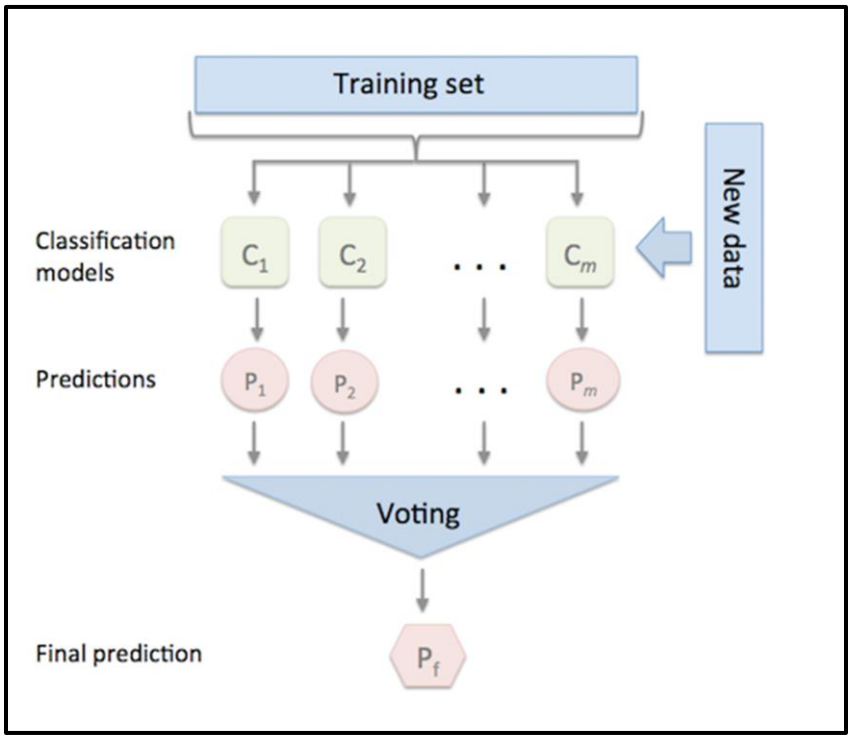

Figure 3.5 A Majority Voting Classifier

Une telle méthode fonctionne bien pour les problèmes de classification binaire, où il n'y a que deux candidats pour lesquels les classificateurs peuvent voter. Cependant, il échoue pour un problème avec de nombreuses classes car de nombreux cas se présentent, où aucune classe n'obtient une majorité claire des voix. Dans ce cas, nous choisissons généralement une classe au hasard parmi les meilleurs candidats, ce qui peut entraîner une marge d'erreur plus importante. [16]

3.5.2 Changer les données : Bagging ensemble learning method

Dans les cas où une quantité substantielle de données est disponible, nous pouvons diviser les tâches de classification entre différents classificateurs et les regrouper pendant le temps de prédiction, plutôt que d'essayer de former un classificateur avec de gros volumes de données.

La façon dont cela fonctionne est assez simple, nous formons différents classificateurs en utilisant divers "échantillons bootstrap" de données, c'est-à-dire que nous créons plusieurs sous-ensembles d'un seul ensemble de données en utilisant le remplacement. Cela signifie que les mêmes données d'échantillon peuvent être présentes dans plusieurs sous-ensembles, qui seront ensuite utilisés pour former différents modèles [16]. Ci-dessous, nous décrivons la méthode de bagging la plus courante dans ce scénario :

a) **La méthode de Bagging**

La méthode de Bagging, acronyme de "bootstrap aggregating", est l'une des méthodes les plus utilisées dans l’apprentissage par ensemble. Cette méthode qui vise à améliorer les performances et la stabilité d'un modèle en combinant les prédictions de plusieurs modèles de base. [16]

Voici les étapes principales de la méthode de bagging :

1. **Création d'échantillons Bootstrap :** Au stade initial, à partir de l'ensemble de données d'entraînement, des sous échantillons, appelés "échantillonnage bootstrap", sont générés en effectuant un échantillonnage aléatoire avec remplacement. Cela signifie que le même point de données peut être présent dans plusieurs sous-ensembles.

2. **Entraînement des modèles de base :** Sur chaque échantillon bootstrap, un modèle de base est entraîné indépendamment. Donc, plusieurs modèles de Machine Learning seront ajustés.
3. **Prédiction des modèles de base :** Une fois que les modèles de base sont entraînés, ils sont utilisés pour faire des prédictions sur l'ensemble de données de test ou de validation. Chaque modèle de base génère une prédiction pour chaque exemple de l'ensemble de données.
4. **Combinaison des prédictions :** Les prédictions des modèles de base sont ensuite combinées pour obtenir la prédiction finale. Dans le cas d'une classification, une méthode de vote majoritaire est généralement utilisée, où la classe prédite est déterminée en fonction des votes de chaque modèle de base. Pour la régression, la prédiction finale peut être la moyenne des prédictions des modèles de base. L'image ci-dessous illustre le mécanisme de la méthode de Bagging.

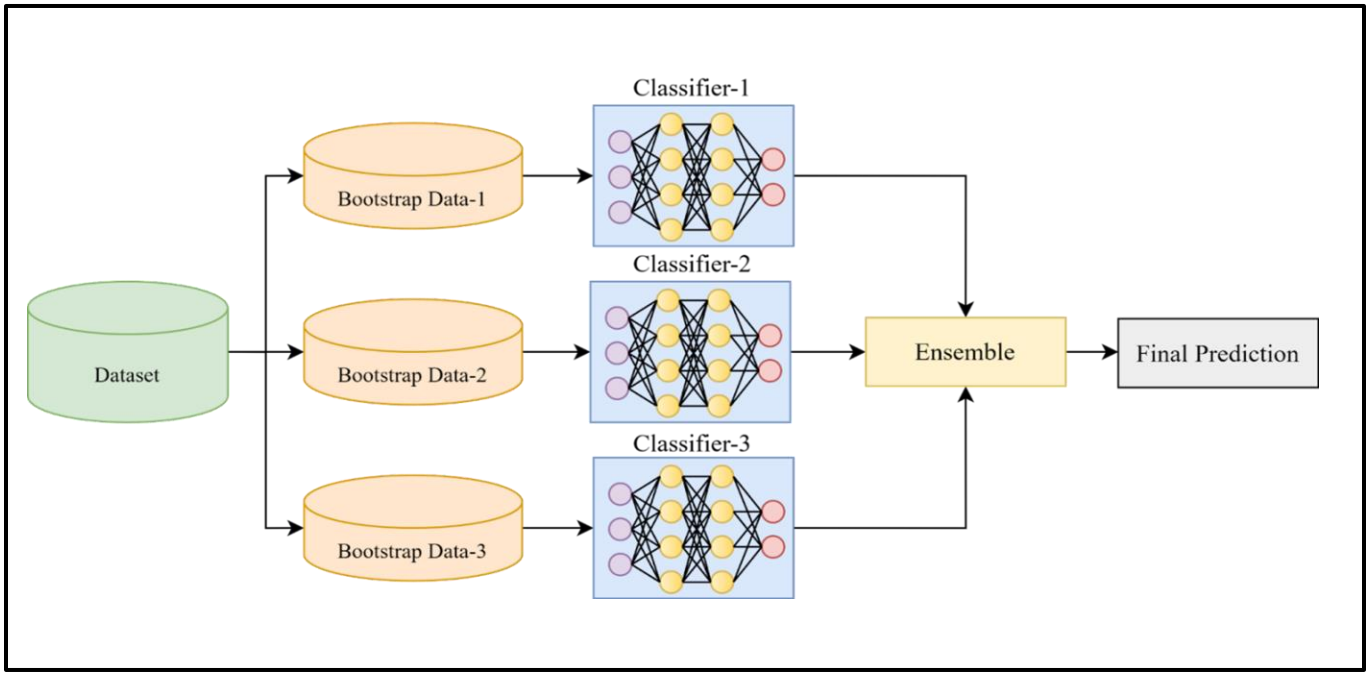

Figure 3.6 Bagging ensemble method.

On peut voire de l’image qu’un flux parallèle de traitement se produit avec la méthode de bagging. L'objectif principal de la méthode de Bagging est de réduire la variance dans les prédictions d'ensemble.[16]. Les méthodes d'ensemble populaires basées sur cette approche comprennent : Bagged decision trees, random forest classifiers, extra trees, etc.

3.6 Travaux connexes

Il existe de nombreuses études dans le domaine de la classification des textes, mais pour la langue arabe, le nombre de recherches est limité. Parmi ces travaux :

- **The Effects of Pre-Processing Techniques on Arabic Text Classification [17]**

Cette étude porte sur l'effet des techniques des techniques de prétraitement sur la classification de textes en langue arabe. Les résultats indiquent que le prétraitement des documents arabes, tel que la suppression des mots vide, la lemmatisation et la normalisation, peut jouer un rôle crucial dans l'amélioration de la performance de la classification.

- **A Survey of Arabic Text Classification Models [18]**

Ce travail propose une enquête approfondie sur les modèles de classification de textes en langue arabe. Il examine différentes approches, y compris les méthodes basées sur les règles, les méthodes statistiques et les techniques d'apprentissage automatique, et fournit des insights précieux sur les avantages et les limitations de chaque approche.

- **Arabic Text Classification: A Literature Review [19]**

Cette étude présente une revue de littérature sur la classification de textes en langue arabe. Elle examine de manière approfondie les études antérieures sur la classification de textes en arabe, en mettant en évidence les méthodes, les approches et les défis spécifiques rencontrés dans ce domaine. Cette revue de littérature offre une vue d'ensemble complète de l'état actuel de la classification de textes en arabe, ce qui peut être extrêmement pertinent pour notre travail en fournissant un contexte solide et une base théorique pour notre propre étude.

- **Arabic Text Classification Methods: Systematic Literature Review of Primary Studies [20]**

Cette recherche présente une revue systématique de la littérature sur les méthodes de classification de textes en langue arabe. Elle explore en détail les études primaires menées dans ce domaine, en analysant les différentes approches et techniques utilisées pour la classification de textes en arabe. Cette revue systématique fournit des informations essentielles sur les méthodes de classification les plus couramment utilisées pour les documents en arabe, ce qui peut être pertinent pour notre travail en aidant à identifier les approches les plus prometteuses et les plus adaptées à notre propre recherche.

- **The Effect of Ensemble Learning Models on Turkish Text Classification [21]**

L'étude évalue l'impact de différentes techniques d'ensemble, notamment le Vote majoritaire, le Bagging et le Boosting, sur la précision de la classification des textes en turc. Ces résultats peuvent être pertinents pour notre étude sur la catégorisation des documents écrits en langue arabe, car ils fournissent des insights sur les avantages potentiels de l'utilisation de techniques d'ensemble pour améliorer la précision de la classification.

3.7 Conclusion

Ce chapitre présente une synthèse sur l'apprentissage automatique avec plus de détails sur les méthodes de classification supervisée (Les machines à vecteurs de support, K-plus proches voisins, Les arbres de décision et Naïve Bayes) et les méthodes ensemblistes (Bagging, Voting).

Chapitre 4: Implémentation et Résultats

4.1 Introduction

Ce chapitre a pour objectif de présenter l'aspect implémentation de notre application, il s'agit donc de présenter le choix du langage de programmation et l'environnement de développement, ainsi que l'ensemble des résultats des expérimentations, nous discuterons ensuite de ces résultats. Enfin, nous présenterons une interface graphique en décrivant les différentes fonctions de notre application et montrerons un exemple qui nous permettra d'illustrer les résultats obtenus lors de l'exploitation de nos modèles de catégorisation.

4.2 Présentation de l'approche proposée

Notre système est basé sur l'utilisation de l'apprentissage automatique pour classer les documents texte en arabe. Le système prend en entrée une base de données contenant des documents en langue arabe et la transforme en une base de données de caractéristiques utilisables dans la phase d'apprentissage. Dans la première étape, il effectue des opérations de nettoyage et de préparation sur ces documents, qui comprennent des techniques telles que la **Normalisation** des mots, la **suppression des mots vides** avec **Tokenisation**, le **Stemming** et la **Lemmatisation**.

Dans la deuxième étape, après le prétraitement des données, les caractéristiques pertinentes sont extraites et sélectionnées pour la classification, en utilisant la pondération **TF-IDF** et la représentation par **Sac de mots** (*Bag-of-Word*), l'objectif étant de représenter chaque document comme un vecteur de caractéristiques significatives.

La base est ensuite divisée en deux parties, une pour l'entraînement et l'autre pour le test. Le module d'entraînement utilise la base d'entraînement et l'un des algorithmes d'apprentissage automatique couramment utilisés pour la classification des textes afin de fournir un modèle de

décision appliqué sur la base de test, le but est d'obtenir des résultats précis. Afin d'améliorer les performances de classification, des méthodes ensemblistes telles que le **Voting** et le **Bagging** sont utilisées pour combiner les prédictions des modèles individuels et produire une prédiction finale.

Dans la dernière étape, l'efficacité du système de classification est évaluée à l'aide de métriques d'évaluation telles que la **Précision**, le **Rappel**, le **Score F1** et l'**Exactitude**. Ces mesures nous donnent une indication de la performance de notre système et nous aident à comparer différents algorithmes ou techniques de classification.

La **Figure 4.1** résume ces étapes :

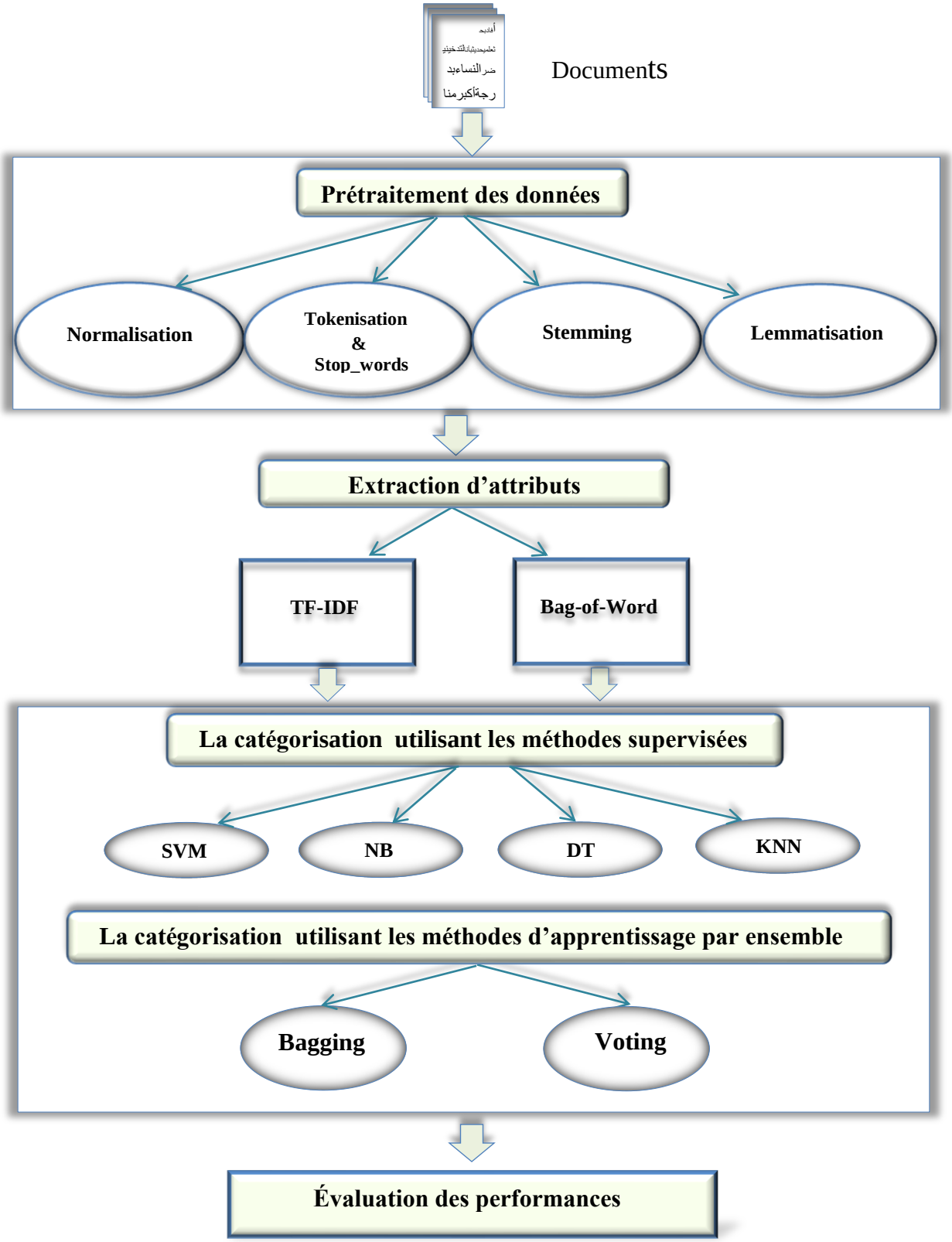

Figure 4.1 Architecture globale du système.

4.3 Outils utilisés

4.3.1 Langage utilisé

a) Python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, *Python* permet de travailler rapidement et d'intégrer les systèmes plus efficacement. Il peut être utilisé pour gérer des données volumineuses et effectuer des calculs complexes. Il existe ce qu'on appelle des bibliothèques qui aident le développeur à travailler sur des projets particuliers.

4.3.2 Bibliothèques utilisées

Nous définissons les bibliothèques les plus importantes utilisées dans notre travail :

a) Pandas

La bibliothèque logicielle open-source *Pandas* est spécifiquement conçue pour la manipulation et l'analyse de données en langage *Python*. Elle est à la fois performante, flexible et simple d'utilisation. [22]

b) Scikit-learn (Sklearn)

Sklearn est une bibliothèque libre *Python* destinée à l'apprentissage automatique. Elle comprend des fonctions pour utiliser des différents algorithmes de classification tel que la régression logistique et les machines à vecteurs de support.

Scikit-learn est fondamentalement écrit en *Python*, avec quelques algorithmes essentiels écrits en *Cython* pour optimiser les performances. [23]

c) NLTK

NLTK, ou *Natural Language Toolkit*, est une suite de bibliothèques logicielles et de programmes écrits en *Python*. Elle est conçue pour le traitement automatique des langues en langage *Python*. C'est l'une des bibliothèques de traitement naturel du langage les plus puissantes.

d) Matplotlib

Matplotlib est une bibliothèque *Python* open source. Cette bibliothèque est particulièrement utile pour créer des tracés, des histogrammes, des diagrammes à barre et tous types de graphiques à l'aide de quelques lignes de code. Il s'agit d'un outil très complet, permettant de générer des visualisations de données très détaillées. [24]

e) Tkinter

Tkinter est une bibliothèque standard de *Python* qui permet de créer des interfaces graphiques pour des applications. Elle est souvent utilisée car elle est simple à apprendre et à utiliser, et elle est disponible par défaut dans l'installation standard de *Python*.

4.3.3 L'environnement de développement

a) Jupyter Notebook

Jupyter Notebook est l'un des éditeurs les plus utilisés dans l'industrie de la science des données. Il exploite au mieux le fait que *Python* est un langage interprété, ce qui signifie que les lignes de code *Python* peuvent être exécutées une ligne à la fois et que le tout n'a pas besoin d'être compilé ensemble comme *C / C ++*.

Cela rend *Jupyter Notebook* l'éditeur idéal pour l'écriture et le prototypage de modèles d'apprentissage automatique. Puisqu'il y a une quantité importante de prétraitements effectués au départ, et après cela, il y a un processus répété de réglage des hyperparamètres et de prototypage de modèle, la possibilité d'exécuter une cellule (un groupe de lignes) ensemble à la fois donne aux *data scientists* la possibilité de régler leurs modèles facilement. [25]

Figure 4.2 Jupyter & Python.

4.4 Base de Données Testées

4.4.1 La première base de données

Nous avons utilisé dans notre travail un corpus de textes arabe dans les domaines : *business*, *entertainment*, *middle_east*, *scitech*, *sport* et *world*. Cet ensemble de données s'appelle *cnn-arabic-utf8,* et se compose de 5070 documents structurés en fichiers texte, le nombre de documents et de mots pour chaque classe varie d'une classe à l'autre. Il est disponible gratuitement sur le web [26].

Le tableau ci-dessous représente la répartition des documents par catégories :

Tableau 4.1 Le nombre de documents dans chaque catégorie.

	cnn-arabic-utf8					
Classes	***business***	***entertainment***	***middle_east***	***scitech***	***sport***	***world***
Nombre de documents	820	500	1500	550	700	1000
Total	5070					

4.4.2 La deuxième base de données

Nous avons utilisé une autre base de données appelée *Alj-News Arabic* [27]. Cet ensemble de données se compose d'articles d'actualité en langue arabe provenant du site web d'*Al Jazeera News*. Les articles sont collectés à partir de différentes sources d'actualités et couvrent une gamme de sujets. Chaque article est étiqueté avec des catégories spécifiques, telles qu'*art*, *economic*, *politics*, *science* et *sport*.

Le **Tableau 4.2** montre le nombre total de documents dans cette base de données et leur répartition entre les catégories.

Tableau 4.2 Répartition des documents dans les cinq catégories.

	Alj-News Arabic				
Classes	***art***	***economic***	***politics***	***science***	***sport***
Nombre de documents	240	240	240	240	240
Total	1200				

4.5 Critères d'évaluation des classificateurs

Comme déjà mentionné, afin d'évaluer la qualité des performances du classificateur, les résultats de classification sont évalués en utilisant les mesures de *précision*, de *rappel*, *F-score* et *d'exactitude (accuracy)* qui sont communément utilisées. Pour calculer ces mesures, on a besoin de définir les valeurs suivantes par rapport à une classe C (dans notre cas : une catégorie de texte C).

- ***TP : le nombre de vrais positifs (True positives).*** C'est le nombre d'instances qui sont correctement classés dans la catégorie *C*.

- ***TN : le nombre des vrais négatifs***. C'est le nombre d'instances qui sont correctement classés hors la catégorie *C*.

- ***FP : le nombre de faux positifs (False positives).*** C'est le nombre d'instances incorrectement classés dans la catégorie *C*.

- ***FN : le nombre de faux négatifs (False negatives).*** C'est le nombre d'instances incorrectement classés en dehors de la catégorie *C*.

Les mesures de *rappel*, *précision* et *F-score* sont définies comme suit :

- ***Le rappel (R)*** est la proportion des solutions pertinentes qui sont trouvées. Il mesure la capacité du système à donner toutes les solutions pertinentes. Dans notre cas, *Rappel (R)* présente le rapport du nombre de documents correctement attribués à la catégorie *C* au nombre total de documents appartenant réellement à la catégorie *C*.

$$Rappel\ (R) = \frac{TP}{TP + FN}$$

- ***La précision (P)*** est la proportion de solutions trouvées qui sont pertinentes. Elle mesure la capacité du système à refuser les solutions non-pertinentes. Dans le cas de la classification des textes, *la précision (p)* est le rapport du nombre de documents correctement attribués à la catégorie *C* au nombre total de documents classés comme appartenant à la catégorie *C*.

$$Précision\ (P) = \frac{TP}{TP + FP}$$

- ***F1-score*** est un indicateur qui combine le rappel et la précision elle est donnée par la formule suivante :

$$F - score\ (F) = 2 * \frac{R * P}{R + P}$$

- ***L'exactitude (Accuracy)*** représente le pourcentage de prédictions correctes effectuées par le modèle parmi l'ensemble des échantillons de données. Dans notre cas, l'exactitude est le rapport de document correctement classés au nombre total de documents classés.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

4.6 Expérimentations

Nous avons mené un processus de classification utilisant un ensemble de classificateurs, répartissant 80% de l'ensemble de données du corpus pour l'apprentissage et 20% pour le test.

4.6.1 Résultats avec les méthodes supervisées

a) Combinaisons des différentes méthodes de pré-traitement avec TF-IDF sur la base cnn-arabic-utf8

L'objectif de ces expérimentations est de tester les quatre algorithmes : les machines à vecteurs de support (SVM), Naïve bayes (NB), les arbres de décisions (DT) et k-plus proches voisins (KNN), en implémentant TF-IDF et la représentation par sac de mots pour ensuite comparer les résultats en termes de Précision, Rappel et F1-score, en plus de la comparaison en termes d'exactitude. Pour des raisons de simplicité, les différentes méthodes de pré-traitement sont abrégées comme suit : *Normalisation (Nr), Stemming (St), Lemmatisation (Lm)* et *Tokenisation avec suppression des mots vides (Tk).*

Nous avons commencé par appliquer les algorithmes SVM, NB, DT et KNN avec un ensemble de techniques de prétraitement, qui sont toutes utilisées avec la pondération TF-IDF. Les tableaux **4.3, 4.4, 4.5** et **4.6** représentent les résultats obtenus avec les méthodes de classification utilisées.

Tableau 4.3 Résultats obtenus avec le modèle SVM appliquant TF-IDF.

Techniques de pré-traitement	**SVM**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.94	0.94	0.94	94.87
Nr	0.94	0.95	0.94	94.77
St	0.94	0.93	0.94	94.37
Lm	0.93	0.93	0.93	93.29
Tk	0.94	0.94	0.94	**94.97**
Lm & St	0.94	0.93	0.94	94.37
Lm & Tk	0.93	0.93	0.93	93.68
Lm & Nr	0.92	0.92	0.92	92.80
St & Nr	0.93	0.93	0.93	93.98
St & Tk	0.94	0.94	0.94	94.67
Nr & Tk	0.93	0.93	0.93	93.29
St & Tk & Lm	0.94	0.94	0.94	94.67
St & Tk & Nr	0.94	0.94	0.94	94.47
Lm & Tk & Nr	0.93	0.93	0.93	93.29
St & Nr & Lm	0.94	0.94	0.93	94.57
Toutes les techniques	0.94	0.94	0.94	94.47

On remarque, d'après le tableau ci-dessus, que tous les scores sont très proches, mais la technique tokenisation avec la suppression des mots vide a donné le score le plus élevé avec une légère différence par rapport au reste des techniques.

Tableau 4.4 Résultats obtenus avec le modèle NB appliquant TF-IDF.

Techniques de pré-traitement	**NB**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.91	0.91	0.91	91.22
Nr	0.91	0.91	0.91	91.22
St	0.90	0.86	0.88	89.54
Lm	0.91	0.91	0.90	91.32
Tk	0.92	0.92	0.91	**92.11**
Lm & St	0.92	0.86	0.88	89.54
Lm & Tk	0.92	0.92	0.92	92.20
Lm & Nr	0.91	0.91	0.91	91.22
St & Nr	0.90	0.89	0.89	89.84
St & Tk	0.90	0.89	0.89	89.84
Nr & Tk	0.92	0.91	0.91	91.91
St & Tk & Lm	0.90	0.89	0.89	89.84
St & Tk & Nr	0.90	0.89	0.89	89.94
Lm & Tk & Nr	0.92	0.92	0.91	92.01
St & Nr & Lm	0.92	0.87	0.89	90.03
Toutes les techniques	0.90	0.89	0.89	89.84

D'après les résultats de **Tableau 4.3** et **Tableau 4.4**, et lors de l'application des modèles SVM et NB, on observe que les meilleures performances étaient dans le cas de l'utilisation de la technique de tokenisation avec la suppression des mots vides (Tk) par rapport aux autres techniques.

Tableau 4.5 Résultats obtenus avec le modèle DT appliquant TF-IDF.

Techniques de pré-traitement	**DT**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.76	0.75	0.75	75.44
Nr	0.77	0.76	0.76	76.62
St	0.77	0.77	0.77	76.72
Lm	0.76	0.75	0.75	75.24
Tk	0.77	0.76	0.76	76.23
Lm & St	0.77	0.76	0.76	75.64
Lm & Tk	0.78	0.78	0.78	**78.20**
Lm & Nr	0.77	0.77	0.77	77.02
St & Nr	0.75	0.75	0.75	75.64
St & Tk	0.75	0.75	0.75	75.83
Nr & Tk	0.77	0.76	0.76	76.33
St & Tk & Lm	0.75	0.75	0.75	75.83
St & Tk & Nr	0.74	0.74	0.74	74.45
Lm & Tk & Nr	0.77	0.76	0.77	76.82
St & Nr & Lm	0.76	0.75	0.76	75.24
Toutes les techniques	0.77	0.77	0.77	77.21

Le **Tableau 4.5** montre que l'algorithme DT avait une performance inférieure par rapport à SVM et NB, où nous notons que DT a obtenu **78.20%** comme valeur d'exactitude. Ce résultat est obtenu lors de l'utilisation de la technique de lemmatisation et de tokenisation avec suppression des mots vides.

Tableau 4.6 Résultats obtenus avec le modèle KNN appliquant TF-IDF.

Techniques de pré-traitement	**KNN**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.94	0.94	0.94	**94.87**
Nr	0.93	0.92	0.92	92.80
St	0.91	0.92	0.91	92.01
Lm	0.93	0.92	0.92	92.99
Tk	0.92	0.92	0.92	92.50
Lm & St	0.91	0.91	0.91	91.81
Lm & Tk	0.92	0.92	0.92	92.80
Lm & Nr	0.92	0.92	0.92	92.40
St & Nr	0.91	0.91	0.91	91.81
St & Tk	0.92	0.92	0.92	92.40
Nr & Tk	0.92	0.92	0.92	92.30
St & Tk & Lm	0.92	0.92	0.92	92.70
St & Tk & Nr	0.92	0.92	0.92	92.30
Lm & Tk & Nr	0.92	0.92	0.92	92.11
St & Nr & Lm	0.92	0.92	0.91	92.01
Toutes les techniques	0.92	0.92	0.92	92.50

D’après les résultats exposés dans le **Tableau 4.6**, on constate que dans le cas où aucune technique de prétraitement n’est utilisée, l’algorithme KNN donne des meilleures performances en termes d'exactitude, précision, rappel et F1-Mesure par rapport aux autres cas.

b) Combinaisons des différentes méthodes de pré-traitement avec sac de mots sur la base cnn-arabic-utf8

À fin de tester la performance des méthodes utilisées pour la représentation des textes, cette deuxième expérimentation a pour objectif de montrer les résultats des mêmes classificateurs en utilisant la représentation par sac de mots afin de les comparer avec les résultats de TF-IDF. Les résultats obtenus sont représentés dans les tableaux **4.7, 4.8, 4.9** et **4.10 :**

Tableau 4.7 Résultats obtenus avec le modèle SVM en utilisant la représentation par Sac de mots.

Techniques de pré-traitement	**SVM**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.94	0.94	0.94	**94.87**
Nr	0.92	0.92	0.92	92.70
St	0.94	0.94	0.94	94.37
Lm	0.94	0.94	0.94	93.29
Tk	0.93	0.93	0.93	93.78
Lm & St	0.94	0.94	0.94	94.37
Lm & Tk	0.93	0.93	0.93	93.68
Lm & Nr	0.92	0.92	0.92	92.80
St & Nr	0.93	0.93	0.93	93.98
St & Tk	0.94	0.94	0.94	94.67
Nr & Tk	0.94	0.94	0.94	94.67
St & Tk & Lm	0.94	0.94	0.94	94.67
St & Tk & Nr	0.94	0.94	0.94	94.47
Lm & Tk & Nr	0.93	0.93	0.93	93.29
St & Nr & Lm	0.93	0.93	0.93	93.98
Toutes les techniques	0.94	0.94	0.94	94.47

Le **Tableau 4.7** montre que les méthodes de pré-traitement n'ont donnée aucune amélioration avec la méthode SVM lorsqu'on utilise la représentation par sac de mots comme une méthode de représentation des textes. De plus, on remarque que, les résultats de SVM avec TF-IDF sont généralement mieux que les résultats de SVM avec la représentation par sac de mots.

Tableau 4.8 Résultats obtenus avec le modèle NB en utilisant la représentation par Sac de mots.

Techniques de pré-traitement	**NB**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.91	0.90	0.91	90.92
Nr	0.91	0.91	0.91	91.02
St	0.92	0.91	0.91	91.22
Lm	0.91	0.91	0.91	91.02
Tk	0.92	0.91	0.91	**91.61**
Lm & St	0.91	0.91	0.91	91.22
Lm & Tk	0.92	0.91	0.91	**91.61**
Lm & Nr	0.91	0.91	0.91	91.02
St & Nr	0.91	0.90	0.90	90.72
St & Tk	0.91	0.91	0.91	91.12
Nr & Tk	0.92	0.91	0.91	**91.61**
St & Tk & Lm	0.91	0.91	0.91	91.02
St & Tk & Nr	0.91	0.90	0.91	90.82
Lm & Tk & Nr	0.92	0.91	0.91	91.51
St & Nr & Lm	0.91	0.90	0.90	90.72
Toutes les techniques	0.91	0.90	0.91	90.82

Tableau 4.9 Résultats obtenus avec le modèle DT en utilisant la représentation par Sac de mots.

Techniques de pré-traitement	**DT**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.78	0.78	0.78	78.40
Nr	0.77	0.76	0.76	76.62
St	0.76	0.77	0.76	77.02
Lm	0.78	0.78	0.78	78.20
Tk	0.80	0.80	0.80	80.37
Lm & St	0.77	0.77	0.77	77.02
Lm & Tk	0.80	0.80	0.80	**80.57**
Lm & Nr	0.79	0.79	0.79	79.88
St & Nr	0.78	0.78	0.78	78.20
St & Tk	0.76	0.76	0.76	76.42
Nr & Tk	0.80	0.80	0.80	80.27
St & Tk & Lm	0.77	0.77	0.77	77.41
St & Tk & Nr	0.77	0.77	0.77	77.02
Lm & Tk & Nr	0.79	0.79	0.79	79.38
St & Nr & Lm	0.77	0.78	0.77	78.10
Toutes les techniques	0.76	0.76	0.76	76.42

D'après les résultats de NB présentés dans le **Tableau 4.8**, on peut voir que le meilleur résultat a été obtenu avec trois combinaison des méthodes de pré-traitement : tokenisation toue seule, tokenisation avec lemmatisation et tokenisation avec normalisation. Encore une fois, la méthode de tokenisation avec lemmatisation donne le meilleur résultat d'exactitude avec le classifieur DT (voir **Tableau 4.9**).

Notez que, pour le classificateur DT, la tokenisation avec lemmatisation a donné les meilleurs résultats de classification dans les deux cas, lors de l'utilisation de TF-IDF et aussi dans

le cas de sac des mots. Ce résultat signifie que la méthode DT nécessite l'utilisation de la tokenisation avec lemmatisation pour donner les meilleurs résultats.

Tableau 4.10 Résultats obtenus avec le modèle KNN en utilisant la représentation par Sac de mots.

Techniques de pré-traitement	**KNN**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.86	0.82	0.83	82.54
Nr	0.85	0.81	0.82	81.65
St	0.91	0.90	0.90	90.23
Lm	0.86	0.82	0.83	82.64
Tk	0.88	0.80	0.82	80.47
Lm & St	0.91	0.90	0.90	90.33
Lm & Tk	0.88	0.81	0.83	81.06
Lm & Nr	0.85	0.81	0.82	81.85
St & Nr	0.89	0.88	0.88	88.46
St & Tk	0.90	0.89	0.89	89.34
Nr & Tk	0.87	0.79	0.82	79.88
St & Tk & Lm	0.92	0.92	0.92	**92.50**
St & Tk & Nr	0.90	0.87	0.88	87.86
Lm & Tk & Nr	0.87	0.79	0.82	79.58
St & Nr & Lm	0.89	0.87	0.88	87.67
Toutes les techniques	0.89	0.87	0.88	87.96

Les résultats de KNN sont présentés dans le **Tableau 4.10** . Cette fois, le meilleur résultat d'exactitude a été enregistré avec la combinaison des trois méthodes de pré-traitement steming, tokenisation et lemmatisation. Nous remarquons aussi que, contrairement au résultat de KNN avec la méthode de TF-IDF, le résultat de KNN avec la représentation par sac de mots a été amélioré

par les différentes méthodes de pré-traitement, le meilleur résultat d'exactitude est (92.50%), ce résultat est 10% meilleur que le résultat de classification sans prétraitement (82.54%).

Globalement parlant, d'après les tableaux **4.7**, **4.8**, **4.9** et **4.10**, et après les avoir comparés avec les résultats de la première expérience, nous avons remarqué une légère diminution des résultats de classification des classificateurs SVM, NB et KNN en général, de sorte que NB était plus précis dans la première expérience avec une moyenne +3.98 % en terme d'exactitude, et KNN par +7,27 %, tandis que la diminution d'exactitude de SVM était avec une petite moyenne (0.15%) dans la deuxième expérience. Contrairement aux résultats du classificateur DT, qui étaient plus précis dans le cas de l'utilisation de la représentation par sac de mots.

La **Figure 4.3** représente la différence entre les résultats des quatre modèles dans les deux cas.

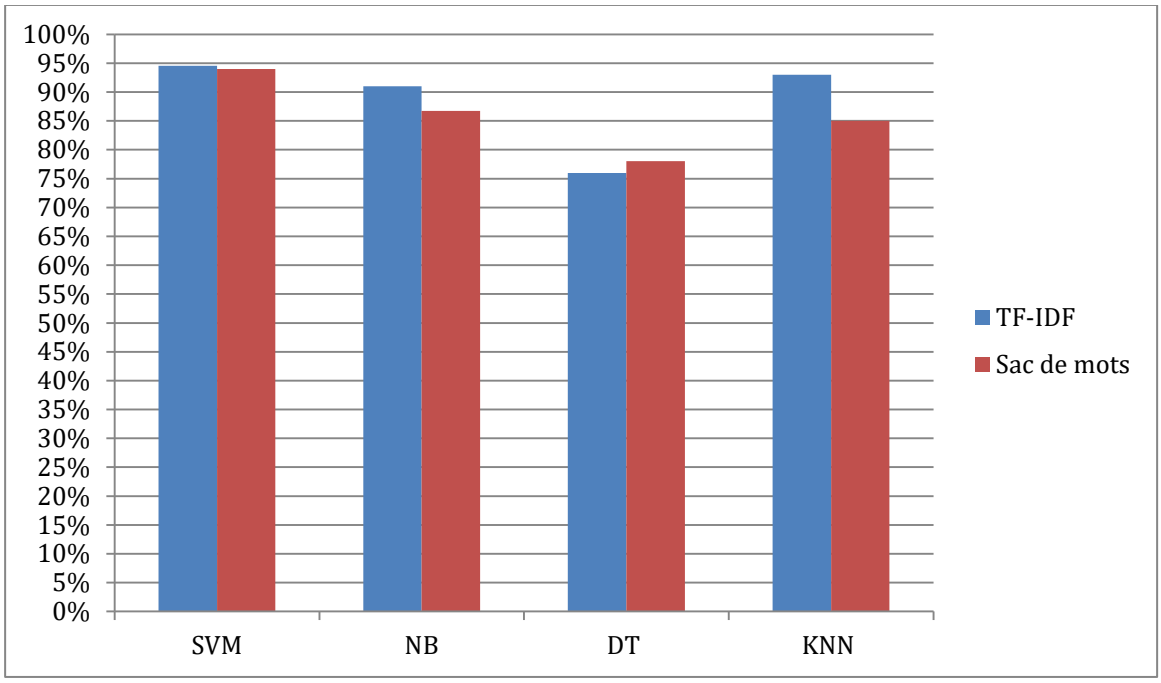

Figure 4.3 La différence entre les résultats en utilisant TF-IDF et Sac de mots.

c) Combinaisons des différentes méthodes de pré-traitement avec TF-IDF sur la base Alj-News Arabic

Afin d'assurer l'efficacité des quatre modèles dans la classification des textes arabes, nous avons utilisé La deuxième base de données (Alj-News Arabic).

Pour ce faire, nous avons suivi la même approche précédente, et les résultats sont présentés dans les tableaux suivants :

Tableau 4.11 Résultats du modèle SVM appliquant TF-IDF en utilisant la deuxième base de données.

Techniques de pré-traitement	**SVM**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.97	0.97	0.97	97.08
Nr	0.97	0.97	0.97	97.08
St	0.97	0.97	0.97	97.08
Lm	0.97	0.97	0.97	97.08
Tk	0.97	0.97	0.97	**97.50**
Lm & St	0.97	0.97	0.97	97.08
Lm & Tk	0.97	0.97	0.97	**97.50**
Lm & Nr	0.97	0.97	0.97	97.08
St & Nr	0.97	0.97	0.97	97.08
St & Tk	0.97	0.97	0.97	97.08
Nr & Tk	0.97	0.97	0.97	**97.50**
St & Tk & Lm	0.97	0.97	0.97	97.08
St & Tk & Nr	0.97	0.97	0.97	97.08
Lm & Tk & Nr	0.97	0.97	0.97	**97.50**
St & Nr & Lm	0.97	0.97	0.97	97.08
Toutes les techniques	0.97	0.97	0.97	97.08

Tableau 4.12 Résultats du modèle NB appliquant TF-IDF en utilisant la deuxième base de données.

Techniques de pré-traitement	**NB**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.96	0.96	0.96	96.66
Nr	0.97	0.97	0.97	97.08
St	0.96	0.96	0.96	96.25
Lm	0.97	0.97	0.97	97.08
Tk	0.97	0.97	0.97	**97.50**
Lm & St	0.96	0.96	0.96	96.25
Lm & Tk	0.97	0.97	0.97	**97.50**
Lm & Nr	0.97	0.97	0.97	97.08
St & Nr	0.96	0.95	0.95	95.83
St & Tk	0.96	0.96	0.96	96.25
Nr & Tk	0.97	0.97	0.97	**97.50**
St & Tk & Lm	0.96	0.96	0.96	96.25
St & Tk & Nr	0.96	0.96	0.96	96.66
Lm & Tk & Nr	0.97	0.97	0.97	**97.50**
St & Nr & Lm	0.96	0.95	0.95	95.83
Toutes les techniques	0.96	0.96	0.96	96.66

Les résultats du modèle SVM et NB montrent que bien que l'ensemble de données soit différent, les valeurs les plus élevées ont été données lors de l'application de tout ensemble de techniques comprenant la tokenisation avec la suppression des mots vides (Tk)

Tableau 4.13 Résultats du modèle DT appliquant TF-IDF en utilisant la deuxième base de données.

Techniques de pré-traitement	**DT**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.79	0.78	0.78	78.75
Nr	0.80	0.79	0.79	79.58
St	0.78	0.77	0.77	77.5
Lm	0.78	0.77	0.77	77.5
Tk	0.82	0.81	0.81	81.66
Lm & St	0.77	0.76	0.76	76.66
Lm & Tk	0.83	0.82	0.82	82.08
Lm & Nr	0.79	0.78	0.78	78.33
St & Nr	0.83	0.82	0.82	82.08
St & Tk	0.80	0.79	0.79	79.16
Nr & Tk	0.82	0.81	0.81	81.25
St & Tk & Lm	0.80	0.80	0.80	80.00
St & Tk & Nr	0.82	0.81	0.81	81.66
Lm & Tk & Nr	0.81	0.81	0.81	81.25
St & Nr & Lm	0.83	0.82	0.82	**82.91**
Toutes les techniques	0.80	0.79	0.79	79.16

Le tableau ci-dessus montre que le modèle DT a donné la meilleure valeur en combinant les trois techniques stemming, lemmatisation avec normalisation, par rapport au reste des techniques.

Tableau 4.14 Résultats du modèle KNN appliquant TF-IDF en utilisant la deuxième base de données.

Techniques de pré-traitement	**KNN**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.90	0.90	0.90	90.41
Nr	0.90	0.90	0.90	90.41
St	0.96	0.96	0.96	**96.25**
Lm	0.90	0.90	0.89	90.00
Tk	0.92	0.92	0.92	92.08
Lm & St	0.96	0.96	0.96	**96.25**
Lm & Tk	0.92	0.92	0.92	92.08
Lm & Nr	0.90	0.90	0.90	90.41
St & Nr	0.96	0.96	0.96	**96.25**
St & Tk	0.95	0.95	0.95	95.00
Nr & Tk	0.91	0.90	0.90	90.83
St & Tk & Lm	0.95	0.95	0.95	95.00
St & Tk & Nr	0.94	0.94	0.94	94.58
Lm & Tk & Nr	0.91	0.90	0.90	90.83
St & Nr & Lm	0.96	0.96	0.96	**96.25**
Toutes les techniques	0.94	0.94	0.94	94.58

Le **Tableau 4.14** montre que les scores les plus élevés ont été obtenus lors de l'application des combinaisons de techniques qui incluent le stemming.

d) Combinaisons des différentes méthodes de pré-traitement avec sac de mots sur la base Alj-News Arabic

Tableau 4.15 Les résultats du modèle SVM appliquant la représentation par sac de mots, en utilisant la deuxième base de données.

Techniques de prétraitement	**SVM**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.95	0.95	0.95	95.00
Nr	0.95	0.95	0.95	95.41
St	0.95	0.95	0.95	95.00
Lm	0.95	0.95	0.95	95.00
Tk	0.95	0.95	0.95	95.41
Lm & St	0.95	0.95	0.95	95.00
Lm & Tk	0.95	0.95	0.95	95.41
Lm & Nr	0.95	0.95	0.95	95.41
St & Nr	0.95	0.95	0.95	95.83
St & Tk	0.95	0.95	0.95	95.83
Nr & Tk	0.95	0.95	0.95	95.00
St & Tk & Lm	0.95	0.95	0.95	95.83
St & Tk & Nr	0.96	0.96	0.96	**96.25**
Lm & Tk & Nr	0.95	0.95	0.95	95.00
St & Nr & Lm	0.95	0.95	0.95	95.83
Toutes les techniques	0.96	0.96	0.96	**96.25**

Les résultats consignés dans le **Tableau 4.15** montrent que la combinaison de toutes les techniques de prétraitement dans ce cas a eu un effet positif sur l'efficacité du modèle SVM.

Tableau 4.16 Les résultats du modèle NB appliquant la représentation par sac de mots, en utilisant la deuxième base de données.

Techniques de prétraitement	**NB**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.96	0.96	0.96	96.25
Nr	0.96	0.96	0.96	96.25
St	0.95	0.95	0.95	95.41
Lm	0.96	0.96	0.96	96.25
Tk	0.97	0.97	0.97	**97.08**
Lm & St	0.95	0.95	0.95	95.41
Lm & Tk	0.97	0.97	0.97	**97.08**
Lm & Nr	0.96	0.96	0.96	96.25
St & Nr	0.95	0.95	0.95	95.41
St & Tk	0.95	0.95	0.95	95.83
Nr & Tk	0.96	0.96	0.96	96.66
St & Tk & Lm	0.95	0.95	0.95	95.83
St & Tk & Nr	0.95	0.95	0.95	95.41
Lm & Tk & Nr	0.96	0.96	0.96	96.66
St & Nr & Lm	0.95	0.95	0.95	95.41
Toutes les techniques	0.95	0.95	0.95	95.41

Ces résultats ont également montré que la technique de tokenisation avec la suppression des mots vide améliore l'efficacité des modèles et conduit souvent à de meilleurs résultats.

Tableau 4.17 Les résultats du modèle DT appliquant la représentation par sac de mots, en utilisant la deuxième base de données.

Techniques de prétraitement	**DT**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.84	0.83	0.83	83.33
Nr	0.84	0.83	0.83	83.33
St	0.79	0.79	0.79	79.16
Lm	0.81	0.80	0.81	80.83
Tk	0.83	0.82	0.83	82.91
Lm & St	0.80	0.80	0.79	80.00
Lm & Tk	0.83	0.82	0.83	82.91
Lm & Nr	0.84	0.83	0.83	83.33
St & Nr	0.82	0.82	0.82	82.50
St & Tk	0.82	0.82	0.82	82.50
Nr & Tk	0.83	0.82	0.83	82.91
St & Tk & Lm	0.81	0.81	0.81	81.66
St & Tk & Nr	0.81	0.81	0.81	81.66
Lm & Tk & Nr	0.82	0.82	0.82	82.50
St & Nr & Lm	0.81	0.81	0.81	81.66
Toutes les techniques	0.84	0.84	0.84	**84.16**

Nous notons à partir du tableau ci-dessus l'effet de la technique de normalisation dans l'amélioration des résultats de classification par rapport au reste des techniques

Tableau 4.18 Les résultats du modèle KNN appliquant la représentation par sac de mots, en utilisant la deuxième base de données.

Techniques de prétraitement	**KNN**			
	Précision	**Rappel**	**F1-Mesure**	**Accuracy** (%)
Sans prétraitement	0.71	0.64	0.64	64.16
Nr	0.71	0.64	0.64	64.16
St	0.84	0.77	0.78	**77.50**
Lm	0.71	0.63	0.64	63.74
Tk	0.80	0.50	0.52	50.83
Lm & St	0.84	0.77	0.78	**77.50**
Lm & Tk	0.80	0.50	0.52	50.83
Lm & Nr	0.71	0.64	0.64	64.16
St & Nr	0.85	0.77	0.78	77.08
St & Tk	0.84	0.75	0.77	75.83
Nr & Tk	0.78	0.56	0.59	56.66
St & Tk & Lm	0.84	0.75	0.77	75.83
St & Tk & Nr	0.82	0.75	0.76	75.41
Lm & Tk & Nr	0.78	0.56	0.59	56.66
St & Nr & Lm	0.85	0.77	0.78	77.08
Toutes les techniques	0.82	0.75	0.76	75.41

D'après les tableaux ci-dessus, et après avoir appliqué les quatre modèles au deuxième ensemble de données, nous avons remarqué une nette augmentation des résultats de classification par rapport au précédent, de sorte que le modèle KNN atteint 96,25 % en termes d'exactitude, tandis que le modèle DT avait une exactitude de 82,91 %, et 97,50 % obtenue par les modèles SVM et NB comme la valeur la plus élevée.

Et cela s'explique par la différence dans la taille des deux ensembles de données, ainsi que la différence dans les catégories.

4.6.2 Résultats avec Ensemble Learning Methods

Après avoir terminé l'étude expérimentale des quatre algorithmes de classification de base selon les techniques de prétraitement, la meilleure combinaison de chaque algorithme a été sélectionné et étudié avec ses ensembles learning de *Bagging* et de *Voting* correspondants, résultant en 10 modèles de classification différents. Chaque modèle a été testé à l'aide d'une « 10-fold cross validation », qui est une stratégie d'estimation des performances bien connue.

Dans cette stratégie, chaque jeu de données est divisé en 10 blocs. Un bloc est conservé comme données de validation pour le test du modèle, et les k - 1 blocs restants sont utilisés comme données d'apprentissage. Ensuite, le processus de validation croisée est répété 10 fois.

Le **Tableau 4.19** montre les résultats *d'exactitude (Accuracy)* de la classification des quatre classificateurs de base et de leurs ensembles learning modèles correspondants.

La colonne nommée « *Base classifier* » affiche les résultats d'exactitude obtenus en utilisant les algorithmes de classification directement sans l'utilisation des modèles d'ensemble learning. La colonne *« Majority voting on base classifiers »* donne les résultats de l'application de la méthode de vote à la majorité sur les classifieurs de base SVM, NB, DT et KNN.

La colonne *« Bagged classifiers»* représente les résultats de l'application de la méthode de bagging sur chaque classifieurs de base. Finalement, la colonne nommée « *Majoriting Voting on Bagged classifiers»* représente les résultats de l'application de la méthode de vote majoritaire mais en utilisant les modèle de bagging des classifieurs et non pas les classifieurs de base.

Tableau 4.19 Résultats d'Accuracy (ACC) des classificateurs de base et de leurs ensembles correspondants.

	Accuracy (ACC)			
Classifiers	***Base Classifiers***	***Majority Voting on base classifiers***	***Bagged Classifiers***	***Majoriting Voting on Bagged classifiers***
SVM	96.50%	96.58%	95.99%	96.33%
NB	96.02%		96.58%	
DT	81.16%		89.50%	
KNN	95.50%		95.83%	

Avant l'utilisation des méthodes d'apprentissage par ensemble, on peut voire du tableau 5.17 que les algorithmes SVM, NB, DT et KNN ont donné les valeurs de précision suivantes : 96.50%, 96.02%, 81.16% et 95.50% respectivement. Comme le montre la **Figure 4.4**, l'algorithme DT a obtenu la pire valeur de performance, alors que la meilleure valeur d'ACC a été obtenue par la méthode SVM (96.50%). La source du succès de SVM est l'espace de caractéristiques de haute dimension de cet algorithme de classification de texte. En fait, SVM est indépendant de la dimensionnalité de l'espace des fonctionnalités et est capable d'apprendre avec une petite quantité de données.

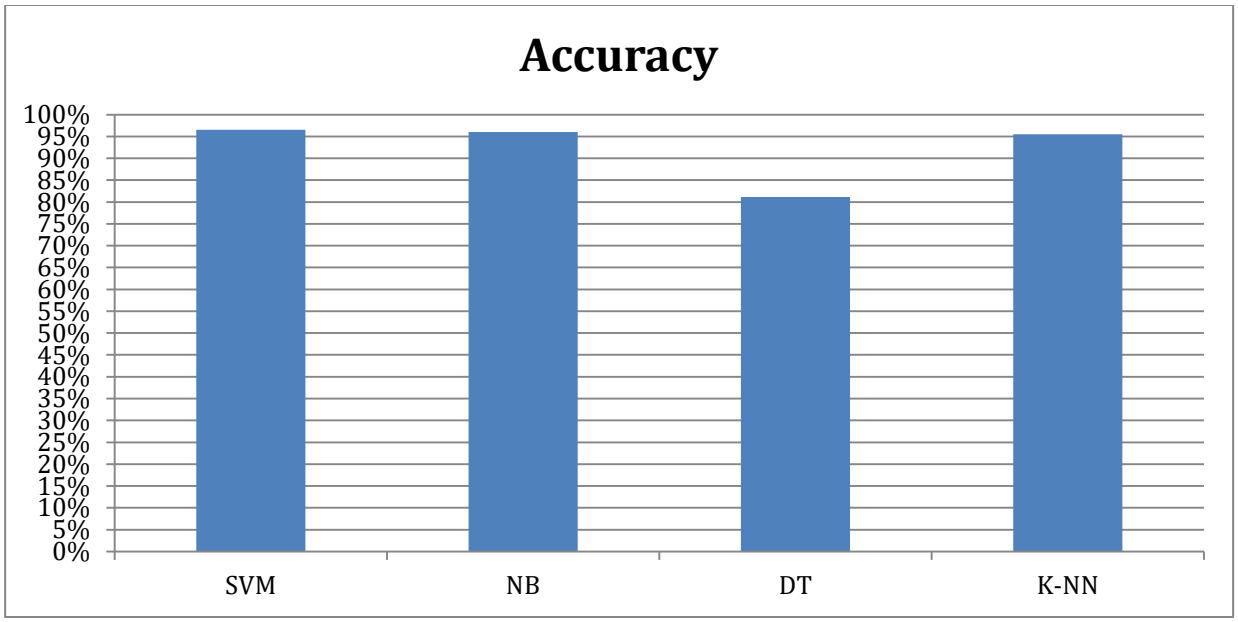

Figure 4.4 Histogramme des résultats d'Exactitude des classificateurs de base.

Regardant les méthodes d'apprentissage par ensemble, la **Figure 4.5** montre que les nouveaux résultats obtenus en appliquant les modèles d'apprentissage par ensemble Bagging et Voting sur les classificateurs de base sont plus précis ou proches des résultats de base. Des résultats plus précis ont été obtenus à l'aide de tous les modèles d'ensemble learning exécutés sur DT, NB et KNN par rapport aux résultats de base.

Par exemple, l'algorithme de classification de base DT donne une valeur ACC de 81.16%; cependant, lorsque le modèle d'ensemble Bagging est utilisé, le *Bagged DT* augmente la valeur d'exactitude 8.34% et obtient une exactitude de 89.50%. De plus, il y a une amélioration de la valeur ACC de 95,4% à 95.8% et de 96.02% à 96.4% lorsque ce modèle d'ensemble est exécuté dans l'algorithme KNN et NB respectivement.

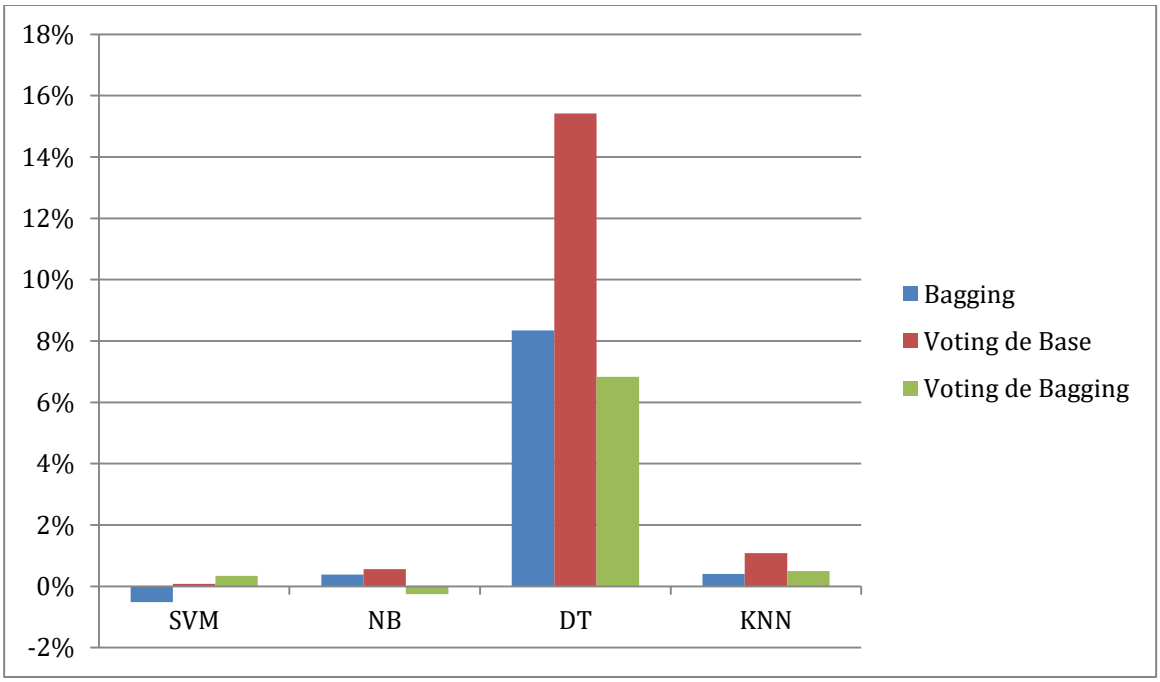

Figure 4.5 La différence d'Exactitude des modèles d'ensemble.

Il ressort également de la **Figure 4.5** que le modèle d'ensemble Bagging n'a pas eu d'effet positif sur le classificateur SVM. Étant donné que SVM est un classificateur puissant, le modèle d'ensemble ne peut pas améliorer la précision de SVM en général.

Après avoir utilisé le modèle de Voting pour les modèles de base et les modèles d'ensembles Bagging, les valeurs ACC ont augmenté de 0,08%, 0,56%, 15,42% et 1.08% pour les modèles SVM, NB, DT et KNN, respectivement. Et il a également amélioré les résultats des modèles d'ensembles Bagging. Par conséquent, nous concluons que ce modèle est le meilleur pour améliorer la précision des modèles en général.

4.6.3 Discussion

Dans une première phase, l'approche suivie dans notre travail est de tester chaque technique de prétraitement individuellement sur chacun des quatre algorithmes (SVM, NB, DT et KNN), puis d'appliquer toutes les combinaisons possibles d'entre eux deux par deux et enfin, de rassembler toutes les techniques.

Dans ces expériences, nous notons que les résultats en utilisant TF-IDF pour les trois classificateurs SVM, NB et KNN étaient meilleurs que dans le cas de la représentation par Sac de mots, à l'exception du cas où l'algorithme DT est utilisé.

À partir des résultats obtenus on peut clairement constater que, le fait de retirer les mots vides avec la tokenisation augmente les performances de tous les modèles que nous avons testés. De même, les combinaisons qui incluent la suppression des mots vides avec la tokenisation sont plus performantes que celles qui ne les incluent pas. À l'exception de KNN lors de l'utilisation de la pondération TF-IDF, dont les résultats étaient meilleurs dans le cas où aucune technique de prétraitement n'est utilisée.

Pour les algorithmes, les résultats montrent clairement l'avance de SVM dans tous les cas testés, que ce soit pour le premier cas lorsque nous avons appliqué la pondération TF-IDF ou le deuxième cas lorsque nous avons utilisé la représentation par sac de mots. En revanche, les résultats du modèle DT étaient les plus faibles, car DT est influencé plus que les autres méthodes par le nombre de classes.

En se basant sur les résultats de la première phase, dans la deuxième phase, nous avons sélectionné la meilleure combinaison pour chaque algorithme et nous avons préposer d'injecter les mécanismes d'ensemble learning à ces modèles dont le but d'améliorer les résultats de ces classificateurs de base. En effet, des résultats plus précis ont été obtenus par rapport aux résultats de base, comme expliqué précédemment.

4.7 Application

4.7.1 Présentation de l'application

Cette section est destinée pour présenter notre application qui a été faite pour fournir une utilisation plus facile de nos modèles. L'interface principale de notre application est illustrée par la **Figure 4.6**.

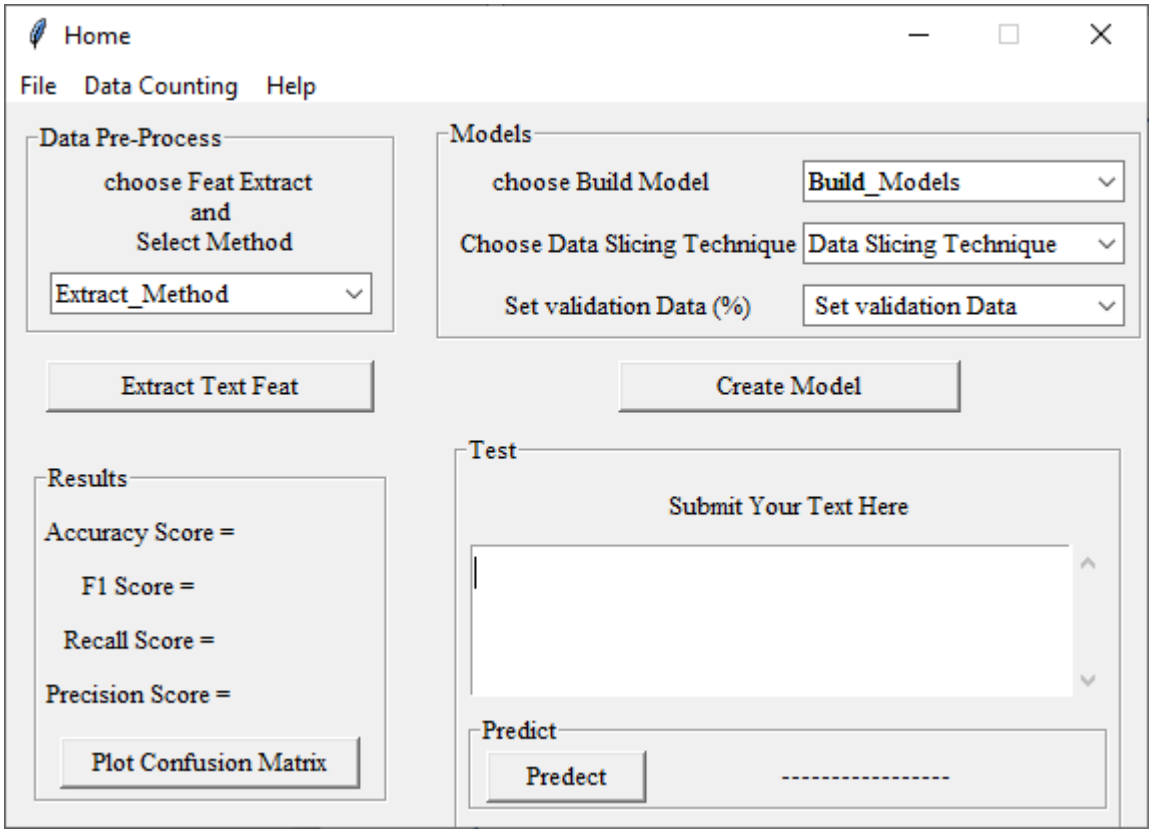

Figure 4.6 Fenêtre 1 d'application.

Dans notre application, plusieurs tâches sont effectuées pour garantir des résultats précis et fiables. Voici les principales étapes impliquées dans le processus :

1. Dans la première étape, nous sélectionnons la base de données.

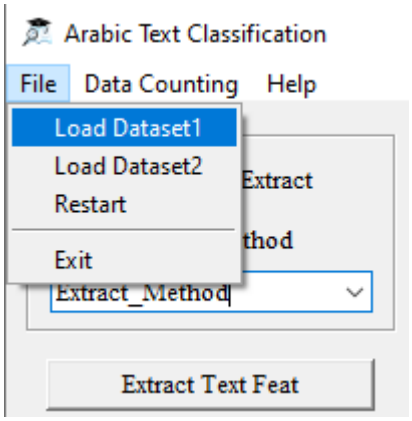

Figure 4.7 Importation de la base de données.

La figure suivante montre la distribution des données en fonction des catégories des textes de la base de données :

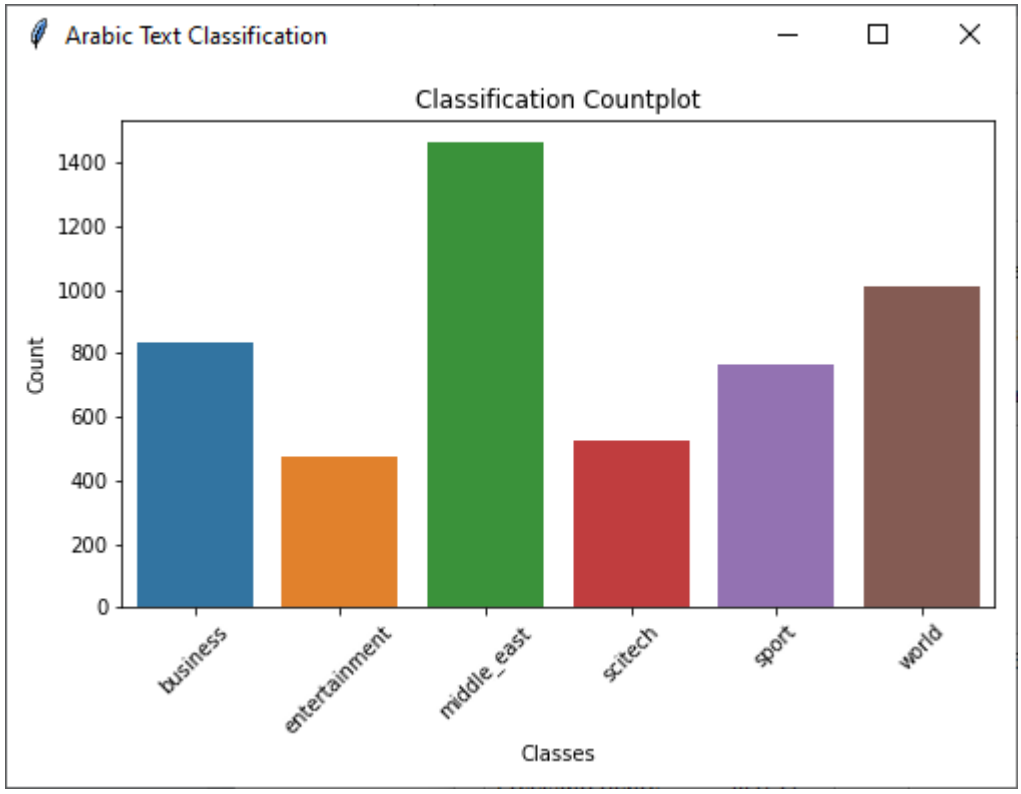

Figure 4.8 Distribution des données.

2. Nous choisissons ensuite la méthode d'extraction de caractéristiques pour convertir les textes en une représentation quantitative qui peut être utilisée par les algorithmes de classification.

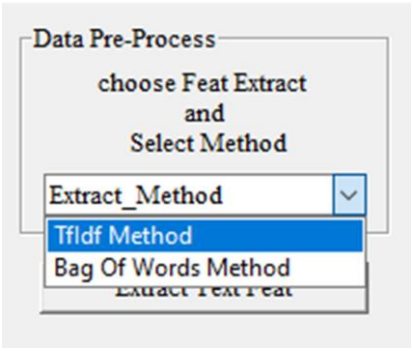

Figure 4.9 Méthodes d'extraction de caractéristiques.

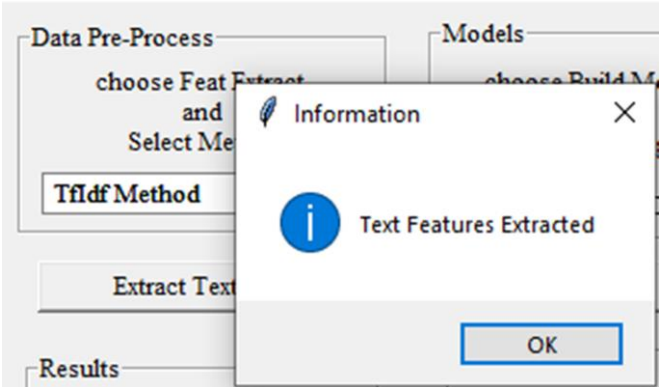

Figure 4.10 Extraire les caractéristiques.

3. L'étape suivante est de définir l'algorithme d'apprentissage et de choisir la technique de découpage des données ainsi que de diviser les données en ensembles d'entraînement et de test.

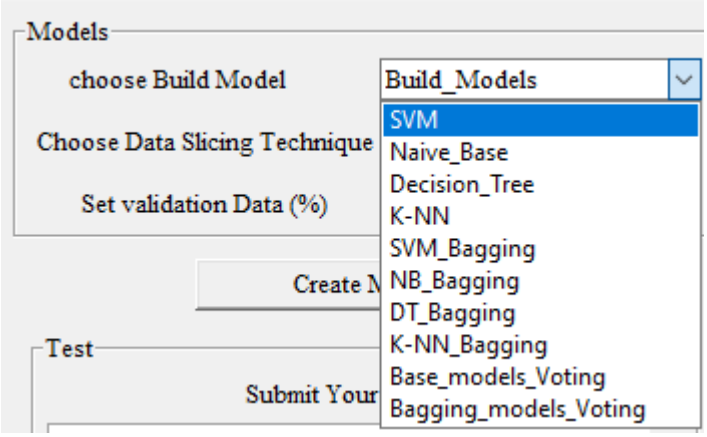

Figure 4.11 Sélection de l'algorithme d'apprentissage.

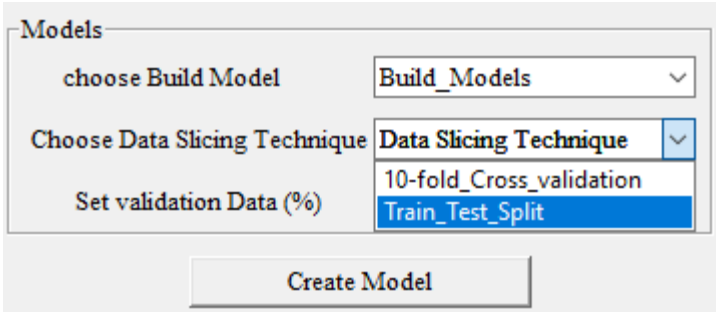

Figure 4.12 Le choix de la technique de découpage des données.

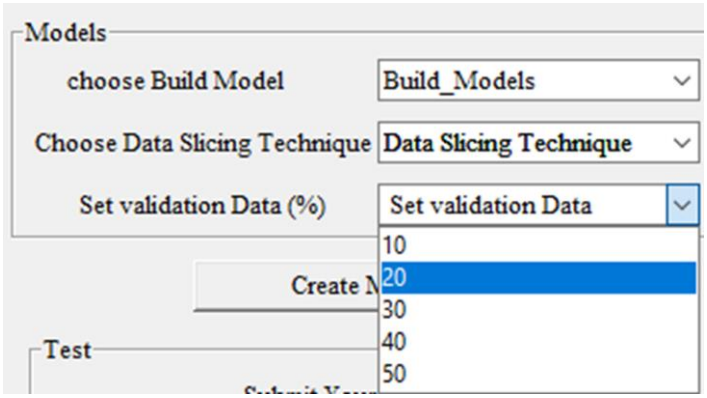

Figure 4.13 Identification des données de validation (%).

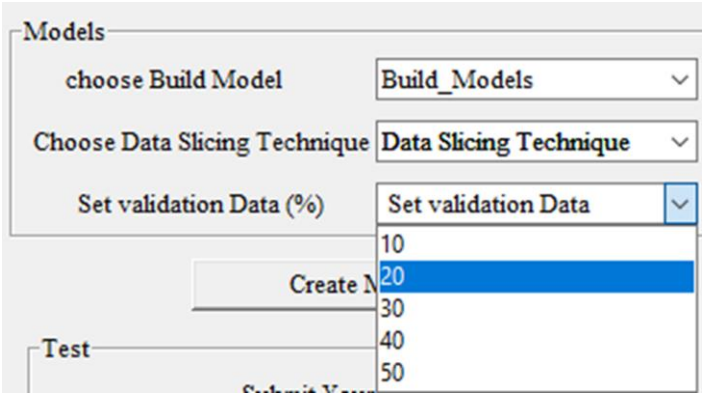

Figure 4.14 Identification des données de validation (%).

4. Après la création du modèle, on peut voir ses performances *(Accuracy, Rappel, Précision, F1 Score, Matrice de confusion)*, comme le montrent les deux figures suivantes :

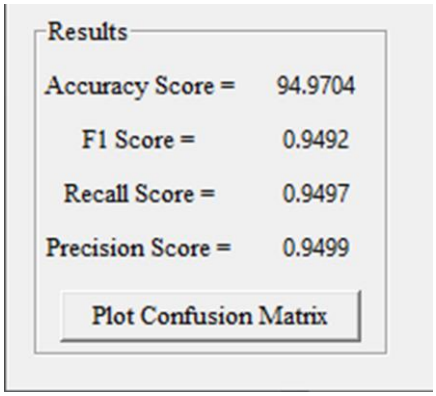

Figure 4.15 Scores du modèle.

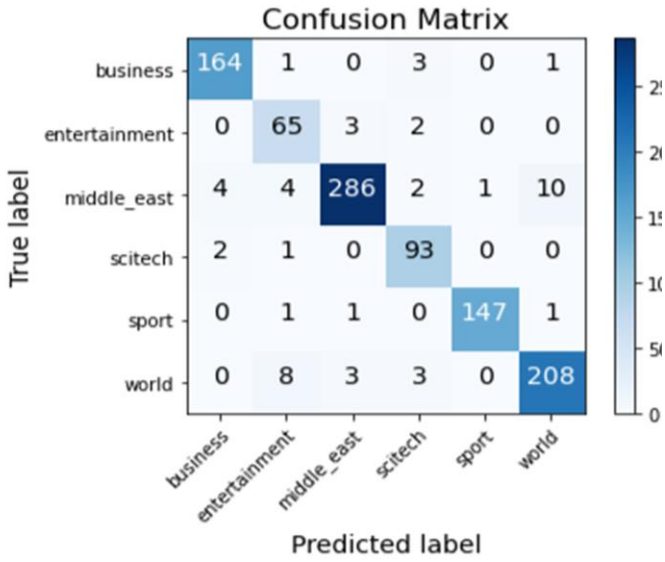

Figure 4.16 Matrice de confusion.

5. La dernière étape représente la phase de l'exploitation des modèles proposés, l'utilisateur peut entrer un nouveau texte pour obtenir la catégorie à laquelle il appartient, après avoir appliqué le prétraitement. Nous montrons un exemple dans la figure suivante :

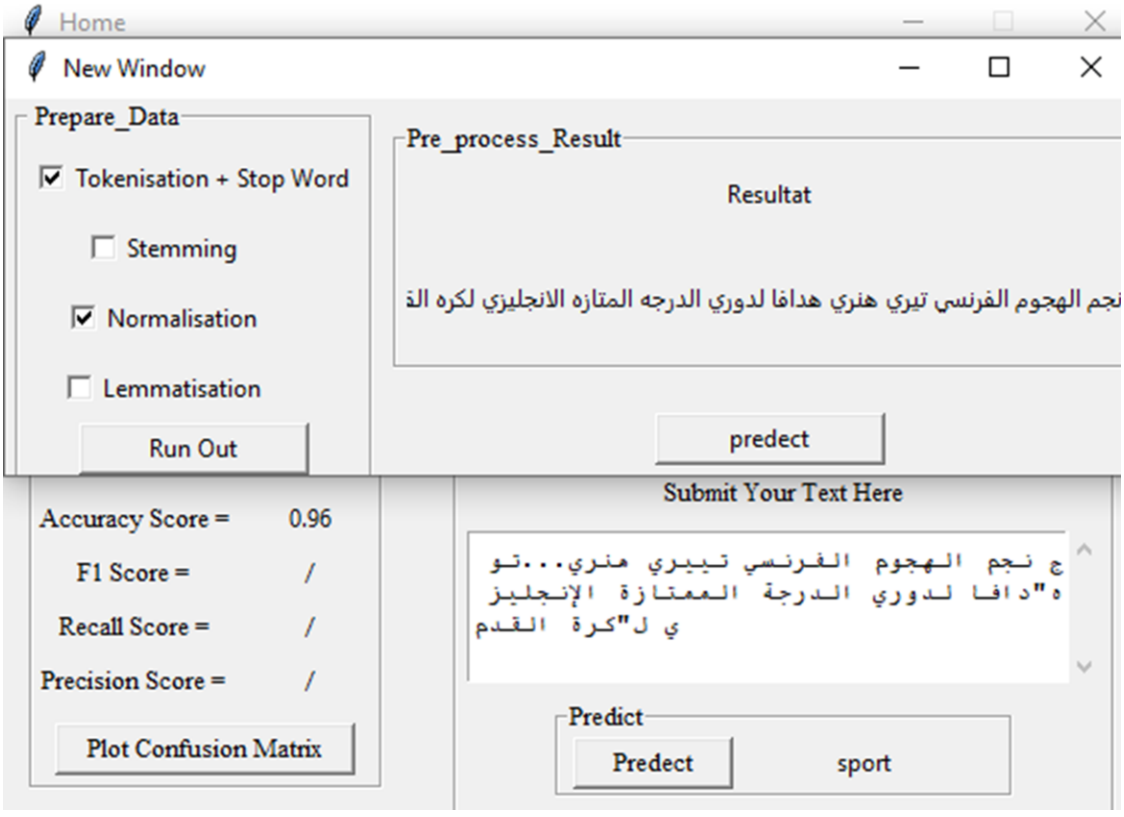

Figure 4.17 Exemple de classification.

4.8 Conclusion

Au cours de ce chapitre, nous avons présenté les outils utilisés dans la mise en œuvre de notre projet. Ensuite, nous avons expliqué la démarche suivie dans le processus de classification, qui comprenait un ensemble d'expérimentations utilisant les quatre algorithmes (SVM, NB, DT et KNN) avec ses ensembles learning de Bagging et de Voting correspondants, comme nous avons également discuté des résultats de ces expérimentations. Finalement, nous avons présenté l'interface graphique de notre application.

Conclusion générale

Dans ce mémoire, nous nous sommes intéressés à la catégorisation des textes arabe en utilisant l'apprentissage automatique. Rappelons que le but de la catégorisation est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu.

Pour atteindre cet objectif, nous avons utilisé une base de données contenant 5070 documents répartis en six catégories et une autre contenant 1200 documents répartis également en 5 catégories.

Nous avons conçu et implémenté une solution basée sur l'utilisation des techniques de Pré-traitement qui sont : la normalisation, tokenisation avec la suppression des mots vides, le stemming et lemmatisation, en plus des techniques d'extraction de caractéristiques (TF-IDF et Bag-of-Words). À la fin, nous avons appliqué les quatre algorithmes d'apprentissage supervisé les plus utilisés dans la classification des textes (SVM, NB, DT et KNN) pour construire des modèles permettant de classer de nouveaux textes. Pour une classification plus précise, nous avons utilisé les deux méthodes d'apprentissage par ensemble Bagging et Voting.

Les expérimentations réalisées ont confirmé que les techniques de prétraitement bien sélectionnées ont un grand impact sur la classification des textes arabes et ont démontré comment leur utilisation a conduit à des résultats positifs. La suppression des mots vides avec tokenisation s'est avérée être la technique la plus bénéfique, en particulier pour les algorithmes de classification qui souffrent beaucoup dans les espaces de caractéristiques de grande dimension comme les arbres de décision. Les résultats ont également prouvé l'efficacité des trois algorithmes SVM, NB et KNN dans ce cas.

Malgré ces résultats positifs, certaines limites ont été rencontrées, à savoir :

Complexité linguistique, la langue arabe est une langue complexe avec une grammaire et une syntaxe distincte. Il a également un grand nombre de formes de mots dérivés, des variations de

racine. Ces propriétés rendent la tâche de classification plus difficile car les modèles doivent être capables de comprendre ces structures linguistiques complexes pour bien fonctionner.

Temps d'entrainement, l'utilisation d'une grande base de données avec Jupyter Notebook prend parfois trop de temps pour l'entrainement.

Les résultats rapportés ici ouvrent des pistes pour de nouvelles recherches pour faire avancer l'état de la classification des textes arabes. Par conséquent, les travaux futurs consistent à :

- Améliorer la tâche de classification en se concentrant sur d'autres phases, telles que les méthodes de sélection des caractéristiques (*feature selection methods*) avec les différents algorithmes de classification.
- Exploiter autres méthodes de pré-traitement qui peuvent fournir de meilleurs résultats pour la langue arabe.
- Le domaine du *Deep learning* a connu des progrès significatifs ces dernières années, offrant de nouvelles opportunités pour résoudre des problèmes complexes tels que la classification des textes. Nous prévoyons de tirer parti des avancées des méthodes d'apprentissage profond afin d'améliorer la catégorisation des textes arabes.

Références bibliographiques

[1] R. Jalam, "Apprentissage automatique et catégorisation de textes multilingues," *PhD Tesis, Université Lumiere Lyon,* vol. 2, 2003.

[2] A. AZIZI, "CLASSIFICATION AUTOMATIQUE DE TEXTES ARABES SUPERVISEE PAR L'ONTOLOGIE LEXICALE WORDNET," FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE DEPARTEMENT D'INFORMATIQUE, 2018.

[3] A. Charif and A. E. Chenene, "Fouille de textes appliquée au Saint Coran," University of m'sila, 2022.

[4] L.SABRI, "Cours : Text et Web Mining (TMW). Partie 1," 2021.

[5] O. SBAI, M. BENPTHMAN, and A. OUAHAB, "Impact le choix du modèle de représentation de textes sur la classification de textes arabes," Université Ahmed Draya-Adrar, 2017.

[6] H. R. Mohamed and A. Lehireche, "La classification non supervisée (clustering) de documents textuels par les automates cellulaires," 2009.

[7] A. Labiad, "Sélection des mots clés basée sur la classification et l'extraction des règles d'association," Université du Québec à Trois-Rivières, 2017.

[17] A. El Kah and I. Zeroual, "The effects of pre-processing techniques on Arabic text classification," *Int. J,* vol. 10, no. 1, pp. 1-12, 2021.

[18] A. M. Al Sbou, A. Hussein, B. Talal, and R. Rashid, "A survey of arabic text classification models," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 8, no. 6, pp. 4352-4355, 2018.

[19] B. Elayeb, "Arabic Text Classification: A Literature Review," in *2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA)*, 2021: IEEE, pp. 1-8.

[20] W. Alabbas, H. M. Al-Khateeb, and A. Mansour, "Arabic text classification methods: Systematic literature review of primary studies," in *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, 2016: IEEE, pp. 361-367.

[21] D. Kılınç, "The effect of ensemble learning models on Turkish text classification," *Celal Bayar University Journal of Science,* vol. 12, no. 2, 2016.

Références webographiques

[8] "RÉDUCTION DE DIMENSIONNALITÉ." https://dataanalyticspost.com/Lexique/reduction-de-dimensionnalite/ (accessed janvier, 2023).

[9] ia-data-analytics.fr. "QU'EST-CE QUE LE MACHINE LEARNING ?" https://ia-data-analytics.fr/machine-learning/ (accessed 23 janvier, 2023).

[10] Linedata. " L'apprentissage_Supervisé." https://fr.linedata.com/quest-ce-que-lapprentissage-supervise (accessed 21 janvier, 2023).

[11] mobiskill. "Apprentissage supervisé vs apprentissage non supervisé." https://mobiskill.fr/blog/conseils-emploi-tech/apprentissage-supervise-vs-apprentissage-non-supervise/ (accessed 25 janvier, 2023).

[12] "Regression vs. Classification in Machine Learning." https://www.javatpoint.com/regression-vs-classification-in-machine-learning (accessed 01 juin, 2023).

[13] "Principaux algorithmes de classification – Partie 1." https://fr.linedata.com/principaux-algorithmes-de-classification-partie-1 (accessed 25 janvier, 2023).

[14] "Principaux algorithmes de classification – Partie 2." https://fr.linedata.com/principaux-algorithmes-de-classification-partie-2 (accessed 25 janvier, 2023).

[15] "A Gentle Introduction to Ensemble Learning Algorithms." Jason Brownlee. https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms (accessed 01 juin, 2023).

[16] "The Complete Guide to Ensemble Learning." https://www.v7labs.com/blog/ensemble-learning (accessed février, 2023).

[22] "DataScientest News." https://datascientest.com/pandas-python-data-science (accessed 02 mai, 2023).

[23] "Scikit-learn." https://fr.wikipedia.org/wiki/Scikit-learn (accessed 02 mai, 2023).

[24] "DataScientest News." https://datascientest.com/matplotlib (accessed 02 mai, 2023).

[25] "Les 10 meilleurs environnements de développement pour Python." https://thkernel.medium.com/les-10-meilleurs-environnements-de-d%C3%A9veloppement-pour-python-603f1b64c21d (accessed 02 mai, 2023).

[26] "Arabic Computational Linguistics." https://osdn.net/projects/sfnet_ar-text-mining/downloads/Arabic-Corpora/cnn-arabic-utf8.7z/ (accessed 03 mai, 2023).

[27] "Alj-News-Arabic-text-classification-dataset." https://github.com/yalhag1/Alj-News-Arabic-text-classification-dataset/blob/Dataset/Alj-News%20Arabic%20Dataset.rar (accessed 04 mai, 2023).