الجمهورية الجزائرية الديموقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la Matière

Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Sciences et technologie de l'information et de la communication

Thème :

Un schéma de pondération et de sélection des termes pertinents basé sur la distribution des fréquences des documents et des termes entre catégories

---

Encadré par :

Dr. Farek Lazhar

Présenté par :

Siafa Aya

Juin 2022

RÉSUMÉ

La sélection des features joue un rôle important dans la catégorisation du texte. Elle s'est avérée être un moyen efficace et efficient de préparer des données de grande dimension pour l'exploration de données et la classification des textes. Parmi les métriques de sélection les plus répandues, nous trouvons : le gain Information (IG), l'information mutuelle (MI), chi-square (Chi2)et fréquence des document (DF) qui utilise la distribution de la fréquence des documents pour calculer la pertinence des mots par rapport à la variable classe, sans considérer la distribution des fréquences des mots intra-documents.
Notre principale contribution consiste à proposer une nouvelle approche appelée (TFDF) de sélection de features basée sur la fréquence des termes et la fréquence des documents au niveau de la catégorie. Dans les expériences, notre méthode proposée est comparée avec les métriques existantes comme IG, MI, Chi2 et DF. Les classifieurs utilisés pour tester les performances des métriques de sélection sont Support Vector Machine (SVM) et Naive Bayes (NB), les plus performants à l'heure actuel.
Les résultats expérimentaux montrent que notre méthode proposée est supérieure aux résultats des métriques existantes dans la littérature.
Mots clés : sélection, terme, classification, texte, fréquence des termes, fréquence des documents.

ABSTRACT

Feature selection plays an important role in text categorization. It has proven to be an effective and efficient way to prepare high dimensional data for data mining and text classification. Among the most popular selection metrics, we find : Gain Information (GI), Mutual Information (MI), Chi-square (Chi2) and Document Frequency (DF) which uses the document frequency distribution to compute the relevance of words to the class variable, without considering the intra-document word frequency distribution.
Our main contribution is to propose a new approach called (TFDF) feature selection based on term frequency and document frequency at the class level. In the experiments, our proposed method is compared with existing metrics such as GI, MI, Chi2 and DF. The classifiers used to test the performance of the selection metrics are Support Vector Machine (SVM) and Naive Bayes (NB), which are the best performing ones at present.
Experimental results show that our proposed method is superior to the results of existing metrics in the literature.

Keywords : selection, term, classification, text, term frequency, document frequency.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE

1. Problématique

Dans le domaine des sciences de l'information, la classification des textes fait référence au processus de classification automatique des ensembles de données de textes, et les classificateurs de texte automatiques peuvent être utilisés pour résoudre plusieurs problèmes du monde réel et déterminer à quelle classe appartient un document donné. Le but de la catégorisation automatique de textes est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu. La catégorisation de textes est le processus qui consiste à assigner une ou plusieurs catégories parmi une liste prédéfinie à un document. L'objectif du processus est d'être capable d'effectuer automatiquement les classes d'un ensemble de nouveaux textes. Dans ce processus, un algorithme est d'abord conçu, puis il est entraîné avec un ensemble de features (termes) spécifiques. Une fois entraîné, l'algorithme est utilisé pour étiqueter de nouveaux textes. Ces derniers sont différents des textes utilisés lors de l'entraînement. L'algorithme est évalué sur le taux d'erreur de classification obtenu lors de la phase d'apprentissage et lors de la phase de test. Lors de l'entraînement de l'algorithme de classification, la phase d'extraction des features utilisées pour l'apprentissage est cruciale. Les features extraites du texte sont généralement extraites d'espaces vectoriels de grande dimension. Le choix de la taille des données d'apprentissage et des tests est très important. Si le classifieur est alimenté par un petit nombre de documents afin de réaliser l'apprentissage, il ne peut pas acquérir des connaissances importantes pour classer les données de test correctement. Par ailleurs, si les données d'apprentissage sont trop importantes par rapport aux données de test, elle conduit à un problème appelé « Sur-apprentissage » (Overfitting) [42]. Ces problèmes peuvent éliminer l'importance de classification de textes.
La sélection des features (eng. Feature Selection – FS) est l'une des techniques les plus couramment utilisées et les plus importantes dans le prétraitement des données et est devenue une partie intégrante du processus d'apprentissage automatique, également connu sous le nom de sélection de variables, sélection d'attributs ou sélection de sous-ensembles variables dans l'apprentissage automatique et les statistiques. C'est le processus de détection des features pertinents et de suppression des features non pertinents, redondants ou bruyants. Ce processus accélère les algorithmes d'exploration de données, améliore la précision des prédictions et augmente la compréhensibilité.

Les méthodes de FS ont reçu beaucoup d'attention de la communauté de classification de textes en raison de leur capacité à améliorer l'efficacité de calcul [35]

, comme la fréquence des documents (DF), le gain Information (IG), l'information mutuelle (MI),Chi-square (Ch2) etc. qui ont prouvé leur efficacité en améliorant les performances des modèles construits tout en minimisant les coûts et le temps d'apprentissage. Mais, malheureusement ces métriques ne considèrent que la fréquence des documents dans les lesquels appartient le terme en ignorant le nombre d'occurrences des termes dans les documents (Term-Frequency), sachant qu'un terme qui apparait fréquemment à l'intérieur des documents semble plus pertinent pour la discrimination de la classe cible qu'un terme pondéré en fonction du nombre de documents auquel il appartient.
En combinant le Term-Frequency (TF) et le Document-Frequency (DF), et en vue d'améliorer les performances des algorithmes de classification, nous proposons dans ce travail, une méthode de sélection qui permet de pondérer les termes en fonction de leurs distribution inter-documents et intra-documents.

2. Organisation du Mémoire

Notre mémoire est organisé en quatre chapitres comme suit :

Chapitre 1 : Dans ce premier chapitre, nous introduisons un aperçu général sur la sélection des features, l'objectif, ainsi que le processus général de la sélection et ses différentes étapes. En outre, nous décrivons quelques méthodes et métriques de sélection des features.

Chapitre 2 : Dans ce chapitre, nous montrons d'abord une définition de la classification des textes, ainsi les étapes de processus de classification, les types de classification, par la suite les algorithmes de classification, et enfin nous citons les mesures d'évaluation de classification.

Chapitre 3 : Dans ce chapitre, nous présentons l'approche proposée et la conception détaillée du système.

Chapitre 4 : Dans ce chapitre, nous présentons les outils et les techniques utilisées et les détails d'implémentation de notre méthode, les détails des tests et les principaux résultats obtenus.

Conclusion générale : Nous concluons notre travail par une conclusion générale.

CHAPITRE 1

SÉLECTION DE FEATURES POUR LA CLASSIFICATION DE TEXTES : ÉTAT DE L'ART

1.1 Introduction

En classification des textes, la sélection des features (en. Feature Selection) est le processus de sélection d'un sous-ensemble de termes qui apparaissent dans l'ensemble d'apprentissage et l'utilisation de ce sous-ensemble comme espace représentatif pour la classification de texte. La sélection des caractéristiques a deux objectifs principaux : Premièrement, cela rend l'apprentissage et l'application des classificateurs plus efficaces en réduisant la taille du vocabulaire. Deuxièmement, la sélection des caractéristiques améliore généralement la précision de la classification en supprimant les caractéristiques indésirables. Un feature indésirable est un feature lorsqu'il est ajouté à la représentation des documents, augmente les erreurs de classification sur les nouvelles données.
Dans ce chapitre, nous allons présenter les principes généraux, les objectifs de sélection des features ainsi que les métriques les plus largement utilisées dans la littérature.

1.2 Sélection de Features : Définition et Objectifs

La sélection des features est une méthode de réduction de la dimensionnalité utilisée en apprentissage automatique. Elle consiste à trouver un sous-ensemble de variables pertinentes toute en minimisant la perte d'information venant de la suppression de toutes les autres variables.
La sélection des features est généralement définie comme un processus de recherche permettant de trouver un sous-ensemble « pertinent » de features parmi celles de l'ensemble de départ. La notion de pertinence d'un sous-ensemble de features dépend toujours des objectifs et des critères du système [16].
comme montré sur la Figure 1.1. le processus général de sélection de features fonctionne comme suit : étant donné un ensemble de dimension n, il faut sélectionner le sous-ensemble de dimension m tel que m < n, conduisant au taux d'erreur le plus faible. [25, 26].

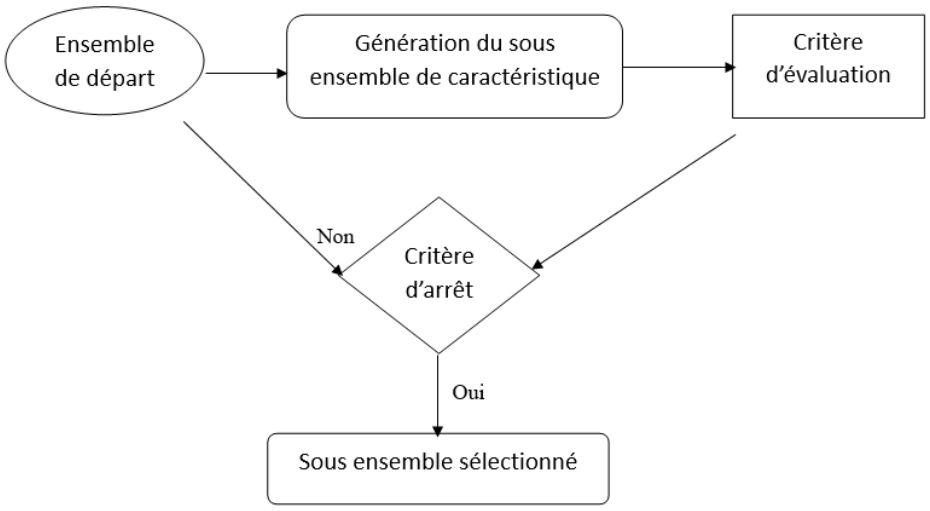


FIGURE 1.1 – Procédure générale d'un algorithme de sélection de caractéristiques.

Les objectifs de sélection de caractéristiques sont les suivantes :

- Minimiser le taux d'erreur de classification.
- Minimiser le nombre de features.
- Identifier les features pertinente.
- Bases d'apprentissage et de test réduites.
- Réduire la taille de l'espace d'entrée en éliminant les informations non pertinentes et redondantes.
- Réduire le temps d'apprentissage.
- Améliorer la vitesse de la classification.

1.3 Approches de sélection des features

Les approches de sélection des features utilisées en classification des données peuvent être classées en quatre catégories : l'approche par filtre, l'approche par enveloppe, l'approche intégrée et l'approche hybride.

1.3.1 L'approche par filtre

La méthode de filtrage vise à sélectionner le sous-ensemble de variables le plus pertinent sans utilisation d'algorithmes d'apprentissage.
Cette approche est très efficace. Cependant, elle ne prend pas en compte le biais et les heuristiques des algorithmes d'apprentissage. Ainsi, elle peut manquer des feature pertinents pour l'algorithme d'apprentissage cible. Un algorithme de filtrage se compose généralement de deux étapes : Dans la première étape, les features sont classées en fonction de certains critères. Dans la deuxième étape, les features les mieux classés sont choisis. De nombreux critères de classement des métriques de sélection sont proposés : la capacité de séparer efficacement les échantillons de différentes classes

en tenant compte de la variance entre les classes et au sein de la classe, la dépendance entre l'entité et l'étiquette de la classe, la capacité de préserver la structure multiple, l'information mutuelle entre les features, etc.

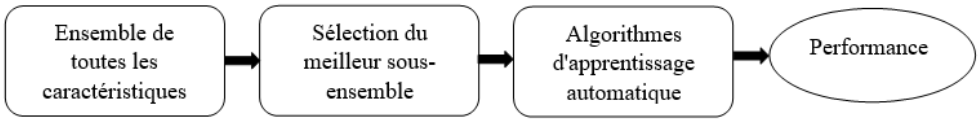


FIGURE 1.2 – L'approche Filtre.

1.3.2 L'approche par enveloppe

L'inconvénient majeur de l'approche par filtre est qu'elle ignore totalement les effets du sous-ensemble de features sélectionné sur les performances de l'algorithme de clustering ou de classification. Le sous-ensemble optimal de features devrait dépendre des biais et des heuristiques spécifiques des algorithmes d'apprentissage. Sur la base de cette hypothèse, les modèles enveloppe utilisent un algorithme d'apprentissage spécifique pour évaluer la qualité des features sélectionnés. Le composant de recherche de features produira un ensemble basé sur certaines stratégies de recherche. Le composant d'évaluation de features utilisera ensuite l'algorithme d'apprentissage prédéfini pour évaluer les performances, qui seront renvoyées au composant de recherche de features pour la prochaine itération de la sélection des sous-ensembles de features. L'ensemble de features avec les meilleures performances sera choisi comme ensemble final.
L'avantage de cette approche est l'amélioration de la précision de classification et la simplicité conceptuelle par contre elle est lente.
Les modèles Enveloppe obtiennent de meilleures estimations de précision prédictive, car ils prennent en compte les biais des algorithmes d'apprentissage. Cependant, les modèles enveloppe sont très coûteux en temps de calcul.

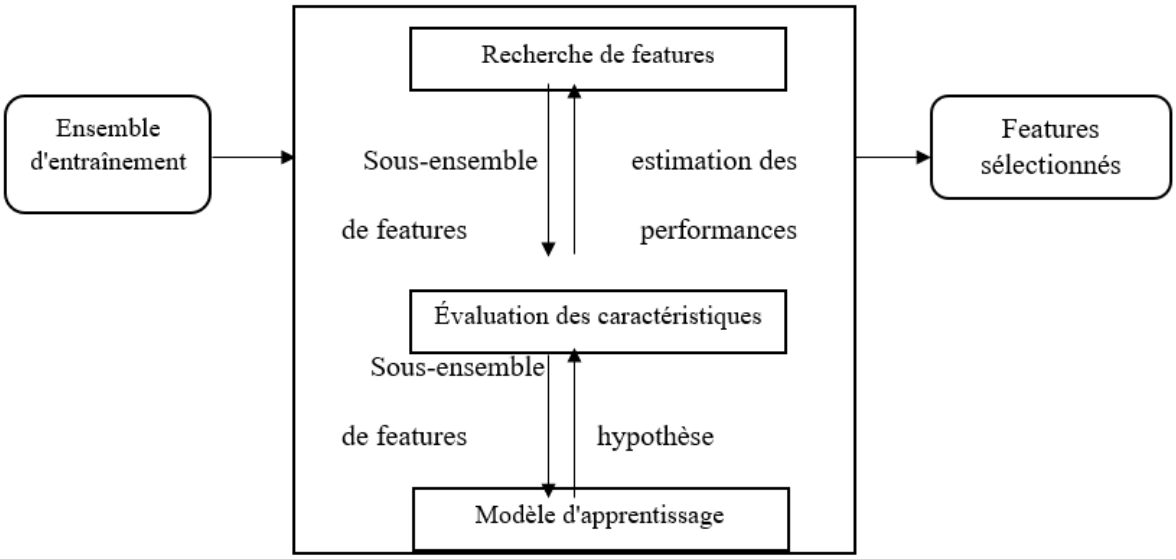


FIGURE 1.3 – Principe de l'approche Enveloppe.

1.3.3 Approche intégrée (embarquée)

Les modèles intégrés sont un compromis entre l'approche par filtre et par enveloppe en intégrant la sélection des features dans la construction du modèle. Ainsi, les modèles intégrés tirent parti à la fois des modèles de filtre et des modèles d'enveloppe :
1) exécuter les modèles d'apprentissage plusieurs fois pour évaluer les features.
2) ils incluent l'interaction avec le modèle d'apprentissage, sélectionner des features pendant le processus de construction du modèle pour effectuer une sélection sans évaluation supplémentaire des features.

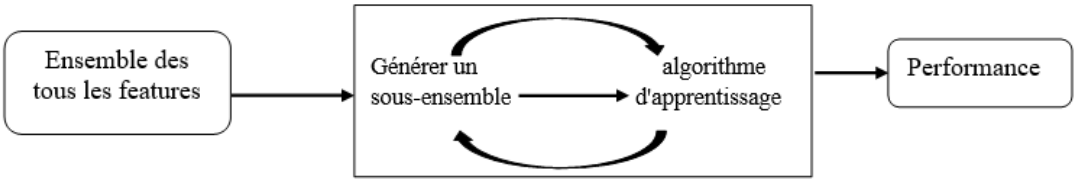


FIGURE 1.4 – Principe de l'approche intégrée.

1.3.4 L'approche hybride

Cette approche est la combinaison entre Filtre et Enveloppe et elle est adoptée dans l'espoir de regrouper les avantages des deux méthodes. Elle consiste à sélectionner les features pertinents (éliminer les features redondants) tout en améliorant la performance du système.

Pour les méthodes hybrides, le processus de sélection des features est effectué conjointement au processus de classification. Une fonction d'évaluation de type 'filtre' est tout d'abord utilisée pour présélectionner les sous-ensembles de features les plus discriminants tout en faisons une bonne discrimination entre les classes, puis le

taux moyen de bonne reconnaissance obtenus après sélection est comparé avec ceux obtenues avant sélection, afin de déterminer le sous-ensemble final [11].

1.4 Métriques de sélection pour la classification de textes

Il existe des techniques qui ont prouvé leur efficacité de la sélection des features en classification de textes en trouvant des bons résultats dans le corpus textuel comme :

1.4.1 Fréquence des documents (en. Document Frequency - DF)

DF est le nombre de documents dans lesquels un terme apparaît dans un ensemble de données. C'est le critère le plus simple pour la sélection des termes et il s'adapte facilement à un grand ensemble de données avec une complexité de calcul linéaire. Il s'agit d'une méthode de sélection de simple mais efficace pour la catégorisation de texte [45].

1.4.2 Gain d'Information (IG)

Information Gain (IG) proposé par Quinlan en 1986 est une méthode supervisée utilisé comme critère de sélection dans le domaine de l'apprentissage automatique [45].
Le gain d'information est capable de détecter la ou les features possédant le plus d'informations, en fonction d'une classe spécifique. Il est dérivé de l'entropie, L'entropie est une mesure de l'incertitude d'une classe en utilisant la probabilité d'un certain événement ou attribut. Étant donné une variable aléatoire discrète X, son entropie est donnée par l'équation 1.

$$H(X) = \sum_{xX} P(x) log(P(x))$$

(1.1)

Où $P(x)$ est la probabilité d'une observation x.
Soit m le nombre de classes, Cj le j-ème classe et p est les probabilité d'un document.
Le gain d'information du terme t est défini comme :

$$IG(C,t) = H(C) - H(C/t)$$
$$= - \sum_{j=1}^{m} p(c_j) log(p(c_j)) + p(t) \sum_{j=1}^{m} p(c_j/t) log(p(c_j/t))$$
$$+ p(t\ ) \sum_{j=1}^{m} p(c_j/t) log(p(c_j/t))$$

(1.2)

1.4.3 Information Mutuelle (IM)

IM est une mesure statistique de la quantité d'informations qu'une variable aléatoire possède sur une autre variable. IM peut être utilisée pour identifier les features pertinents pour une catégorie particulière [45] .
MI mesure l'association entre un terme t_i et une catégorie cible cj ayant des probabilités $P(t_i)$ et $P(c_j)$ respectivement, Elle est calculée par la formule suivante :

$$MI(t_i, c_j) = \frac{logp(t_i, c_j)}{(p(t_i)p(c_j))}$$

(1.3)

S'il existe une forte association entre le terme t_i et la classe c_j, alors $MI(t_i, c_j)$ est élevé, ce qui peut être interprété comme la probabilité conjointe $p(t_i, c_j)$ est supérieure au produit des probabilités marginales $p(t_i)$ et $p(c_j,)$. Inversement, s'il existe une association faible entre le terme t_i et la catégorie c_j, alors $MI(t_i, c_j)$ est une association faible, qui peut être interprétée comme $p(t_i, c_j)$ inférieur à $p(t_i)p(c_j)$. Si t_i et c_j sont complètement indépendants, $MI(t_i, c_j)$ est nul, ce qui signifie $p(t_i, c_j) = P(t_i)P(c_j)$ donc $MI(t_i, c_j) = log(1) = 0$.

1.4.4 Gini Index (GI)

GI à l'origine utilisé dans les algorithmes d'arbre de décision mais ont proposé la méthode améliorée de Gini Index pour l'appliquer directement à la sélection de features des données textuelles.[41, 40].Gini index est une mesure statistique utilisée pour quantifier si le feature est capable de séparer les instances de différentes classes [21].
L'indice de Gini est une méthode qui mesure la pureté des features par rapport aux classes [39]. La pureté fait référence au niveau de discrimination des features qui distinguent les classes possibles [23]. Pour un terme t, l'indice de Gini est calculé comme suit :

$$GI(t) = \sum_{j=0}^{m} p(tc_j)^2 p(c_j|t)$$

(1.4)

Où, m est le nombre de classes, $P(t|c_j)$ est la probabilité du terme t étant donnée la classe c_j , $P(c_j|t)$ est la probabilité de classe c_j étant donné le terme t.

1.4.5 Chi-square (Ch2)

Chi-square est une méthode supervisée de sélection de features, utilisée pour tester l'indépendance de deux variables statistiques (en trouvant des valeurs de test basées sur la relation entre le terme t et la variable cible c), par calcul de la corrélation du terme t avec la classe C_i . Le score du Ch2 utilise le test d'indépendance pour

évaluer si le feature est indépendant de de la classe [30].
Le score du Ch2 pour un terme quelconque est calculé comme suit :

$$X^2(T_k, C_i) = N\frac{(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)}$$

(1.5)

où : N est le nombre total des documents dans le corpus, A est le nombre de documents dans la classe c_i contenant le terme T_k ; B est le nombre de documents contenant le terme T_k dans d'autres classes ; C est le nombre de documents de la classe ci qui ne contiennent pas le terme T_k ; D est le nombre de documents qui ne contiennent pas le terme T_k dans d'autres classes [12].

1.5 Conclusion

La sélection des features est un domaine de recherche qui a engendré de nombreuses études et de nouvelles méthodes, c'est une étape très importante dans la création des modèles de Machine Learning. Cela peut accélérer le temps d'apprentissage, rendre les modèles plus simples et plus faciles à déboguer.
Dans ce chapitre, nous avons décrit le processus, les méthodes et les métriques de sélection de features pour la classification des textes, ainsi l'importance de la sélection des features pour améliorer les performances d'un algorithme de classification.
Dans le chapitre suivant, nous allons présenter un état de l'art sur la classification de texte qui a une relation avec les méthodes de sélection de features qui atténuent les problèmes clés des procédures de classification car elles sont utilisées pour améliorer la précision de la classification, réduire la dimensionnalité des données et supprimer les données non pertinentes.

CHAPITRE 2
CLASSIFICATION SUPERVISÉE DES TEXTES

2.1 Introduction :

L'apprentissage automatique supervisé est l'un des problèmes les plus étudiés en matière de classification de texte. La classification de texte est le problème de l'attribution automatique de catégories prédéfinies à des documents en texte libre. Comme de plus en plus d'informations textuelles sont disponibles en ligne, une récupération efficace est difficile sans une indexation et un résumé approprié du contenu du document. La classification des documents est la solution à ce problème. Ces dernières années, de plus en plus de méthodes de classification statistique et de techniques d'apprentissage automatique ont été appliquées à la classification des textes. La classification de texte devient une technologie clé pour traiter et organiser un grand nombre de documents.
Dans ce chapitre nous présentons d'abord une définition de la classification supervisée des textes, ainsi le processus de classification, les types de classification, par la suite les algorithmes de classification et enfin nous citons les modèles d'évaluation des classifications.

2.2 Définitions

La classification de textes est une tâche générique qui consiste à regrouper de manière automatisée des documents qui se ressemblent suivant certains critères à savoir les critères observables tels que le type du document, l'année, la discipline, l'édition, etc., ou le critère du contenu, et à assigner une ou plusieurs catégories, parmi une liste prédéfinie, ou non à un document. La classification de textes est définie comme une opération qui identifie des classes d'équivalence entre des segments de textes en tenant compte de leur contenu informationnel [32] .
La catégorisation (classification) de textes est le processus qui consiste à assigner une ou plusieurs catégories parmi une liste prédéfinie à un document. L'objectif du processus est d'être capable d'effectuer automatiquement les classes d'un ensemble de nouveaux textes. La catégorisation de documents consiste à apprendre, à partir d'exemples caractérisant des classes thématiques, un ensemble de descripteurs discriminants pour permettre de ranger un document donné dans la (ou les) classe(s)

correspondant à son contenu [15] .

2.3 Processus de classification

Il s'agit de créer un classificateur à base d'un corpus de documents étiquetés (généralement à la main), ce corpus est partitionné en deux ensembles : l'ensemble d'apprentissage et celui de test. Cela consiste, en premier temps, à entrainer ce classificateur avec l'ensemble d'apprentissage, et une fois appris, nous testerons son efficacité avec l'ensemble test. Dans certain cas, nous terminerons ce processus par une étape de validation du classificateur avec un ensemble de nouveau documents. Le schéma ci-dessous illustre parfaitement ces différentes étapes [43].

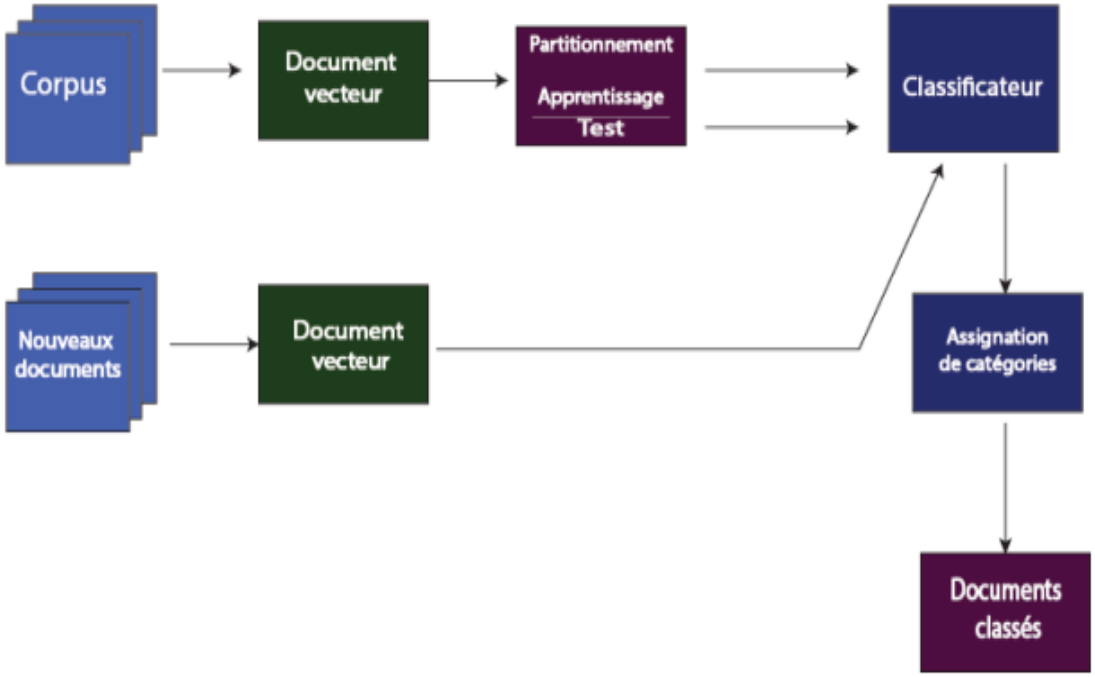


FIGURE 2.1 – Processus de classification de documents[43].

2.4 Algorithmes de classification

Parmi les algorithmes d'apprentissage les plus couramment utilisés pour étudier le problème de classification de texte sont les suivantes : Naïve Bayes (NB), Support Vector Machines (SVM), Arbres de Décision, Réseaux de Neurones, k-Nearest Neighbors (KNN). Les algorithmes les plus utilisés sont NB et SVM car ils sont reconnus comme ayant de bons résultats dans la tâche de classification des textes [35].

2.4.1 Naïve Bayes (NB)

Naïve Bayes (NB) est une approche probabiliste basée sur le théorème de bayes avec une forte indépendance (naïve), il suppose l'indépendance des attributs et la

prévision du futur à partir du passé et des connaissances à priori.
NB est populaire dans la classification de texte en raison de son efficacité de calcul et de ses performances prédictives et sa simplicité. Mais il est très sensible à la sélection des features (termes). La méthode de classification naïve bayésienne est un algorithme d'apprentissage supervisé qui permet de classifier un ensemble d'observations selon des règles déterminées par l'algorithme lui-même.

Le classificateur NB est un classificateur de probabilité commune et son principe principal est le principe de Bayes de l'hypothèse d'indépendance [46]. Il est implémenté en Java sur la base de son principe. Son avantage est que le principe de mise en œuvre est simple, il suffit de faire quelques calculs de probabilité. De plus, si les hypothèses d'indépendance mutuelle sont vraies, NB est plus rapide que de nombreux modèles de classification. Même si les hypothèses ne sont pas vraies, le classificateur NB a généralement un bon effet dans l'utilisation réelle. Il a généralement une efficacité de classification stable [48].

Notons $P(c_i)$ la probabilité a priori d'une classe c_i , $P(x)$ la probabilité d'observer un vecteur-feature x et $P(x|c_i)$ la probabilité d'observer le vecteur x sachant que la classe est c_i . La règle de Bayes permet alors de calculer la probabilité a posteriori de la classe c_i quand x est observé [16] :

$$p_{(c_i|x)} = \frac{P(xc_i)P(c_i)}{\sum_j P(xc_j)P(c_j)} \tag{2.1}$$

Dans la pratique, puisque le dénominateur de la formule de Bayes ne dépend pas de C_i , nous ne nous intéressons qu'au numérateur. Les probabilités $P(Ci)$ de chaque classe ainsi que les distributions $P(x|C_i)$ doivent être préalablement estimées à partir d'un échantillon d'apprentissage. Le vecteur x est assigné à la classe Ci si [16] :

$$\forall j \neq i, P(C_i|x) > P(C_j|x) \tag{2.2}$$

L'analyse discriminante se présente comme un cas particulier de l'approche Bayésienne. Dans ce cas, les données d'apprentissage sont modélisées par des distributions Gaussiennes. Sur la base des paramètres estimés, des fonctions discriminantes sont construites permettant de classer tout vecteur de features [16] .

2.4.2 Support Vector Machine (SVM)

SVM est un algorithme d'apprentissage utilisé en Machine Learning pour résoudre des problèmes de classification supervisée.
Le classificateur SVM est un modèle de classification à deux classes dont le modèle de base est défini comme un classificateur linéaire avec le plus grand intervalle dans l'espace des features. SVM présente les avantages d'une haute précision et d'une bonne garantie théorique de sur-ajustement. SVM fournit un moyen d'éviter la complexité de l'espace de grande dimension. Il utilise directement la fonction noyau de cet espace et la méthode de résolution dans le cas de la séparabilité linéaire pour résoudre le problème de décision de l'espace de grande dimension correspondant, ce qui assure l'effet de classification dans l'espace de grande dimension [1].

Le SVM est un classificateur dit linéaire, ça veut dire que, dans le cas parfait, les données (document texte dans notre cas) doivent être linéairement séparables. Ainsi notre corpus est représenté comme étant un espace vectoriel, où chaque document texte est représenté par un point dans ce dernier. La problématique maintenant est de trouver le meilleur séparateur (ligne, plan ou hyperplan) qui partage notre corpus en deux catégories. L'espace entre ces deux catégories est appelé marge, qui est définie par les points (Vecteurs de Support) les plus proches du séparateur, de part et d'autre. Le but étant essentiellement de maximiser cette marge. Le classificateur se généralise bien avec les nouvelles données. C'est un classificateur qui se généralise très bien avec les nouvelles données, mais le temps d'apprentissage est très élevé [33].

Hyperplan : est un séparateur d'objet de classes, A partir de ce concept, on peut dire que de nombreux hyperplans peuvent évidemment être trouvés, mais une Machine à Vecteurs de Support est un hyperplan avec la plus petite distance des échantillons d'apprentissage est le plus grand, cet hyperplan est appelé l'hyperplan optimal, et la distance est appelée marge.

Vecteur Support : ce sont les points qui déterminent l'hyperplan tels qu'ils soient les plus proches du séparateur.
La figure illustre un schéma représentatif de ces notions [36] :

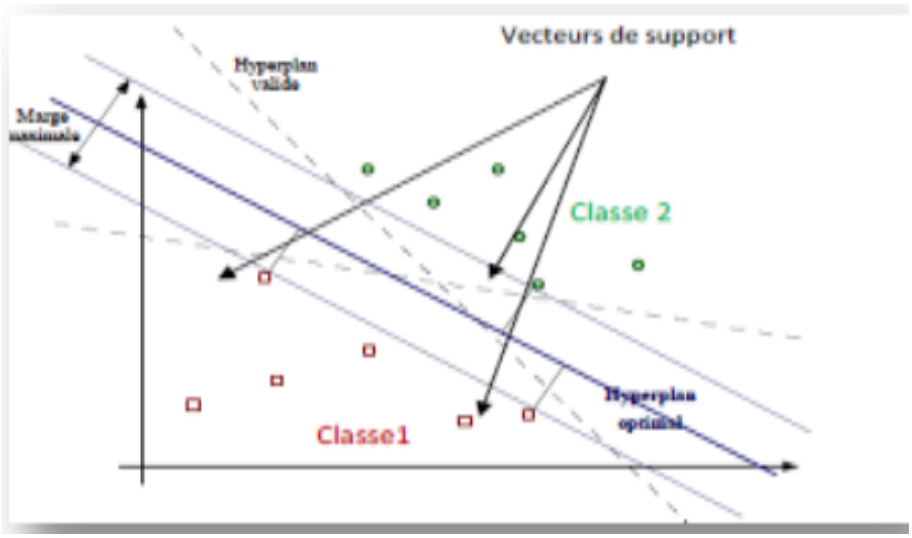


FIGURE 2.2 – Les vecteurs à support [36].

Le principe des SVMs est de réduire le problème de classification ou de discrimination à un hyperplan (espace des features), où les données sont divisées en plusieurs catégories avec des frontières aussi éloignées que possible des points de données (ou "marges maximales") qui veut dire la distance du point le plus proche de l'hyperplan. Les SVMs sont appréciées pour leur simplicité d'usage.

Les SVM représentent plusieurs avantages, notamment ceux-ci :
— Elles ont une base théorique solide [2].
— Les SVM sont efficaces dans les espaces de grande dimension [24].
— Différentes fonctions noyau peuvent être spécifiées [24].

Malgré leurs performances, les SVM représentent aussi des faiblesses, notamment celles-ci :

— Elles utilisent des fonctions mathématiques complexes pour la classification [24].
— Les SVMs demandent un temps énorme durant les phases de test [2].

2.4.3 Arbres de Décision (AD)

L'arbre de décision (AD) est une classe d'algorithmes d'apprentissage, un arbre est un graphe non orienté, acyclique et connexe. Il se compose d'un ensemble des nœuds et se divise en trois catégories : Nœud racine, Nœuds internes et les feuilles (les nœuds terminaux). Ils apprennent à partir d'observations qu'on appelle des exemples (attributs et classes associées), ils structurent les données sous la forme d'une séquence de décisions, par une représentation hiérarchique pour but de distinguer les similitudes et les différences entre les attributs des exemples du jeu de données [22].

Les arbres de décision sont un outil d'apprentissage automatique basé sur la structure d'arbre. L'avantage majeur de ce type est qu'il est très semblable au raisonnement humain, de ce fait il est très facile d'expliquer les résultats obtenus et son comportement. La construction de l'arbre est basée sur un algorithme dans lequel on utilise des mesures statistiques pour faire des séparations (split) entre les données à travers un un attribut sélectionné. Particulièrement, dans chaque itération et pour un ensemble de données, on calcule une mesure pour chaque variable, et la meilleure parmi eux sera choisie comme Nœud. Les différentes valeurs figurées dans cette variable seront les branches directes du nœud en cours. Chaque chemin produit par une branche nous ramène vers un nouveau sous-ensemble de données. Le critère d'arrêt est le suivant : Si un sous-ensemble donné dans lequel on trouve que toutes les instances appartiennent à la même classe alors on va créer une feuille d'arbre, et la fin du processus pour ce chemin. Pour le cas contraire, un sous-ensemble avec des instances de différentes étiquettes, le processus de construction du nœud est repris. Les algorithmes ID3, C4.5 et CART sont les plus reconnus parmi d'autres [13].

Les étapes d'apprentissage d'un arbre de décision sont les suivantes :

— Choix d'une variable de segmentation
— Traitement des variables continues
— Définir la bonne taille de l'arbre
— Décision
— Fusion des sommets lors de la segmentation

2.4.4 Réseaux de neurones

Un réseau de neurones est une modélisation mathématique du cerveau humain. C'est un ensemble de neurones interconnectés entre eux permettant la résolution de problèmes complexes tels que la reconnaissance des formes ou le traitement du langage naturel, grâce à l'ajustement des coefficients de pondération dans une phase d'apprentissage. Il est calqué sur le paradigme du cerveau humain dont il démultiplie la puissance, sans lui ressembler tout à fait puisqu'il est dépourvu d'émotions

[1].
En général, l'architecture de base se compose de trois types de couche neuronale : les couches d'entrée, les couches cachées et les couches de sortie.
Un réseau se distingue en général par le type de neurone formel qu'il utilise, la règle d'apprentissage qui le décrit et l'architecture définissant les interconnections entre neurones. Le neurone formel, qui peut être considéré comme une modélisation élémentaire de neurone réel, est un automate possédant n entrées réelles z, ..., zn et dont le traitement consiste à affecter à sa sortie O le résultat d'une fonction d'activation f de la somme pondérée de ses entrées.

$$O = f(\sum_{i=1}^{n} w_i z_i) = f(net_i)$$

(2.3)

Où les w; sont les coefficients ou poids synaptiques associés aux entrées z_i. La fonction d'activation f est généralement de type sigmoïde. La règle d'apprentissage tente de déterminer les valeurs (optimales) des coefficients synaptiques à l'aide d'exemples permettant de minimiser une fonction de coût d'erreur définie entre la sortie effective du réseau et la sortie désirée [14].

2.4.5 K-Nearest Neighbors (KNN)

K plus proche voisin est une classification basée sur des exemples où un document non vu est classé dans la catégorie de la majorité des k documents les plus similaires. La similitude entre deux documents peut être mesurée par la distance euclidienne de n features correspondants. Tous les voisins peuvent être traités de manière égale avec un poids correspondant à leur distance par rapport aux documents catégorisés [44].Dans les cas où plusieurs de ces k plus proches voisins appartiennent à la même catégorie, leurs poids sont additionnés, utilisant ainsi la somme pondérée finale comme score de probabilité pour cette catégorie.

Notons par $X_p = (n_p1, n_p2, \ldots, n_pN)$ le vecteur de feature de l'entité p, avec N le nombre de features, et par p et q deux entités à comparer.

Les distances suivantes sont usuellement employées par les classificateurs KNN :

Distance Euclidienne : D (X_P, X_q) $= \sqrt{\sum_{i=1}^{N}(x_{p_i} - x_{q_i})^2}$

Distance de Manhattan : D (X_P, X_q) $= \sum_{i=1}^{N}(|x_{p_i} - x_{q_i}|)$

Distance de Minkowski : D (X_P, X_q) $= (\sum_{i=1}^{N}(x_{pi} - x_{qi})^r)^{1/r}$

Distance de Tchebychev : D (X_P, X_q) $= max_{N,i+1}(|x_{p_i} - x_{q_i}|)$

Le pseudo-algorithme suivant décrit le principe de fonction de K-NN :

Pour déterminer les documents les plus proches, toutes les mesures de similarité ou distance peuvent être utiles. Notamment, la distance Euclidienne, Manhattan et

Algorithme 1 Algorithme KNN (DATA, k, distance)

1: Charger les données.
2: Initialiser la valeur de k.
3: Parcourir et calculer la distance entre le point teste et l'ensemble des points d'apprentissage.
4: Trier dans une liste les documents (points) d'apprentissage en ordre croissant en fonction des valeurs de distance.
5: Obtenir le top k documents à partir de notre liste triée.
6: Déterminer la classe la plus fréquente de ces documents.
7: Retourner la classe trouvée.
8: Fin d'algorithme.

la similarité Cosinus parmi d'autres. La construction d'un classificateur K-NN peut contenir dans son expérimentation un ensemble de validation afin de déterminer le seuil K qui indique combien de documents d'entrainements sur lesquelles la prise de décision sera établie. Par ailleurs, les expériences ont montré que l'augmentation de K n'augmente pas forcément la performance du classificateur KNN [13].

2.5 Évaluation des modèles de classification

Certains principes d'évaluation sont couramment utilisés dans le domaine de classification de texte. Deux types de métriques différents couramment utilisés lors de l'évaluation des classificateurs supervisés sont les métriques macro-moyennes et les métriques micro-moyennes. Ces deux mesures consistent généralement en des scores de précision, de rappel et d'une combinaison des deux appelée score F. Les métriques macro-moyennes sont calculées sur une base de classes et ne prennent donc pas en compte la taille de la classe. Les métriques micro-moyennes, d'autre part, sont calculées sur la base d'un document, ce qui signifie que les classes plus grandes auront un impact plus important sur les métriques résultantes que les classes plus petites[20].

2.5.1 Matrice de confusion

La matrice de confusion (en. Confusion Matrix) est un outil qui mesure les performances d'un modèle de classification a deux classes ou plus, qui permet de confronter les valeurs observées avec celles de la prédiction et sert à vérifier le bon classement des documents.
Pour une classification binaire, la matrice de confusion a la forme suivante :

		Classes prédites	
		Classe 1	Classe 2
Classes réelles	Classe 1	VP	FN
	Classe 2	FP	VN

TABLE 2.1 – Matrice de Confusion.

On définit les :

— **VP** : vrai positif (true positif) : les cas où les prédictions sont positives, et la valeur réelle est effectivement positive .

— **VN** : vrai négatif (true négatif) : les cas où les prédictions sont négatives, et la valeur réelle est effectivement négative .

— **FN** : faux négatif (false négatif) : les cas où les prédictions sont négatives, et la valeur réelle est effectivement positive .

— **FP** : faux positif (false positif) : les cas où les prédictions sont positives, et ou la valeur réelle est effectivement négative .

2.5.2 Accuracy

C'est le pourcentage de bonne prédiction, est une mesure permettant d'évaluer les modèles de classification basée sur la matrice de confusion, elle mesure le taux de prédictions correcte sur l'ensemble des individus.

$$\textbf{accuracy} = \frac{\textbf{vrai positif +vrai negatif}}{\textbf{total}} \tag{2.4}$$

2.5.3 Précision

C'est également appelée valeur prédictive positive (taux de prédictions correctes), c'est le rapport entre les observations positives (prédite) et le nombre total d'observations positives prédites. Il mesure la capacité du modèle à ne pas faire d'erreur lors de prédiction positives.

$$\textbf{précision} = \frac{\textbf{vrai positif}}{\textbf{vrai positif+faux positif}} \tag{2.5}$$

2.5.4 Rappel

Également appelé le taux de vrai positif (sensibilité ou sensitivité), c'est le rapport entre les observations positives (prédites) et toutes les observations de la classe réelle.

$$\textbf{rappel} = \frac{\textbf{vrai positif}}{\textbf{vrai positif+faux negatif}} \tag{2.6}$$

2.5.5 F1-score

C'est la moyenne harmonique de la précision et du rappel. Par conséquent, ce score prend en compte à la fois les faux positifs et les faux négatifs. Intuitivement, ce

n'est pas aussi facile à comprendre que la précision, mais F1 est généralement plus utile que la précision, surtout si vous avez une distribution de classe inégale. La précision fonctionne mieux si les faux positifs et les faux négatifs ont un coût similaire. Si le coût des faux positifs et des faux négatifs est très différent, il vaut mieux regarder à la fois Précision et Rappel.

$$\textbf{F1-score} = \mathbf{2} \times \frac{\textbf{precision} \times \textbf{rappel}}{\textbf{precision} + \textbf{rappel}} \tag{2.7}$$

2.6 Types de classification supervisée

Comme nous l'avons déjà mentionné dans ce chapitre, la classification de texte est le problème pour déterminer à quelle classe un document donnée appartient. Cependant, un document peut appartenir à plusieurs catégories à la fois, c'est ce qu'on appelle la classification multi-labels.

2.6.1 Classification binaire

La classification binaire est la forme de classification la plus répandue, où la sortie est limitée à deux classes. Par exemple, dans le diagnostic médical, un classificateur binaire pour une maladie spécifique pourrait prendre en compte les symptômes d'un patient et prédire si le patient est en bonne santé ou à une maladie. Les résultats possibles du diagnostic sont positifs et négatifs.
Le résultat de la classification est un score (probabilité d'appartenance à l'une des deux classes) entre 0 et 1, donc on doit choisir un seuil de décision pour classer en classe 0 ou 1.

2.6.2 Classification Multi-labels

Les approches de classification multi-label se divisent en trois grandes familles selon :
1) Méthodes de transformation : elles transforment le problème d'apprentissage multi-labels en plusieurs problèmes de classification ou régression mono-label [31].
2) Méthodes adaptées : elles utilisent des algorithmes d'apprentissage mono-label pour les adapter au cas multi-labels [31].
3) Méthodes ensemble :elles intègrent des ensembles de classifieurs [31].

2.6.3 Classification multi-classes

La classification multi-classes est un problème de classification avec plus de deux classes. La classification multi classe suppose que chaque catégorie est affectée à une classe. Il existe de nombreuses façons de résoudre ce problème. Nous pouvons utiliser un classificateur binaire pour résoudre un problème de classification multiple.
Les métriques standard utilisées dans le modèle multi-classes sont les mêmes que celles utilisées dans le cas de la classification binaire.

2.7 Méthodes de pondération

Pour toute représentation de textes, il faut choisir une façon efficace de coder les termes d'un vecteur dans lequel un document est représenté.

2.7.1 Sac à mots (Bag of Words)

L'un des types les plus simples de modèles d'extraction de caractéristiques s'appelle Bag of Words. Le nom Bag of Words fait référence au fait que ce modèle ne tient pas compte de l'ordre des mots. Au lieu de cela, on peut imaginer que chaque mot est mis dans un sac, où l'ordre des mots se perd. Bien qu'il existe quelques variantes différentes de ce modèle, la plus courante consiste simplement à compter le nombre d'occurrences de chaque mot dans un document et à conserver le résultat dans un vecteur. De cette façon, les fréquences des termes restent intactes, bien que la grammaire et l'ordre soient perdus [19].

2.7.2 Term Frequency Inverse Document Frequency (TF-IDF)

La pondération TF-IDF acronyme de (Term Frequency-Inverse Document Frequency) est la formule la plus répandue dans le codage de textes. Elle est issue du monde de la recherche d'information [37].

Le codage TF-IDF prend en compte deux critères importants à la fois pour un terme : le premier (TF Term Frequency) exprime l'importance locale tandis que le deuxième (IDF : Inverse Document Frequency) exprime l'importance globale, qui est l'inverse du nombre des documents dans lesquelles le terme est apparu. Autrement dit, la pondération TF-IDF tient en compte aussi bien des termes importants dans les documents par la pondération TF que des termes importants par rapport à la collection par la pondération IDF [13].
TF-IDF consiste en deux étapes, calculant d'abord la fréquence du terme (TF), puis calculant la fréquence inverse du document (IDF). Il existe plusieurs variantes de ces deux parties. Une variante de TF fonctionne en calculant d'abord combien de fois un terme apparaît dans un document, comme vous le faites pour le vecteur de comptage. Le raisonnement ici est que les mots qui apparaissent fréquemment dans un document sont probablement plus importants que les mots qui n'apparaissent pas fréquemment. Le résultat est ensuite normalisé en le divisant par le nombre de mots dans l'ensemble du document. Cette normalisation est effectuée afin d'éviter un biais vers des documents plus longs, de sorte que nous obtenions la fréquence à laquelle le terme apparaît et pas seulement le décompte brut du terme [29].

$$tf_{t,d} = \frac{n_{t,d}}{\sum_k n_{k,d}} \tag{2.8}$$

Où $n_{t,d}$ est le nombre de fois que le terme t apparaît dans le document d, et $n_{k,d}$ est le nombre d'occurrences de chaque terme dans le document d. Pour calculer la partie IDF de la formule, une variante consiste à prendre le nombre total de documents

dans le corpus et à le diviser par le nombre de documents où le terme apparaît. La partie IDF de la formule agit comme une forme d'attribution de poids, donnant plus de poids aux termes importants et moins de poids aux termes insignifiants.

$$idf_t = log\frac{|D|}{|D_t|}$$

(2.9)

L'une des faiblesses de cette méthode, et des méthodes similaires du sac de mots en général, est qu'elle ne tient pas compte du contexte et des synonymes. Il ne tiendrait pas compte, par exemple, du fait que les termes « prêtre » et « révérend » se réfèrent à un sujet similaire. Elle ne tiendrait pas non plus compte du fait que le terme « Apple » pourrait désigner soit la société Apple Inc., soit le fruit selon le contexte [19].

TF représente la proportion du terme ti dans la catégorie c_k, et IDF représente la fréquence inverse des documents. Plus le nombre de textes contenant le terme t_i est petit, plus IDF est grand. Cela indique que le terme t_i a une meilleure capacité de discrimination, et vice versa. La formule de calcul de TFIDF [47] est la suivante :

$$TFIDF = tf_{i,k} \times log(\frac{t_P + f_n + f_P + t_n}{t_P + f_P + 1})$$

(2.10)

Où $tf_{i,k}$ représente la proportion du terme t_i dans la catégorie c_k.

2.8 conclusion :

La classification supervisée des textes a parcouru un long chemin ces dernières années. L'utilisation croissante de données textuelle a conduit à la nécessite d'une classification automatique des textes.

Dans ce chapitre, nous avons aperçu général sur la classification des textes et les notions connexes à cette tâche qu'on devrait connaitre due à son lien avec la sélection des features et la méthode proposée que nous allons présenter dans le chapitre suivant.

CHAPITRE 3

MÉTHODE PROPOSÉE DE SÉLECTION DES FEATURES POUR LA CLASSIFICATION DES TEXTES

3.1 Introduction :

La sélection des features demeure une technique efficace et efficiente de catégorisation de texte. La sélection des features est une étape importante pour la classification des textes, elle utilise les métriques de sélection comme Chi-Square test (Chi2), Document-Frequency (DF), Information Gain (IG), Mutual Information (MI), etc., sont des méthodes statistiques qui emploient seulement la fréquence des documents du terme pour calculer son score. Cependant, l'évaluation de l'importance du terme en introduisant son fréquence (Term-frequency) à l'intérieur du document peut conduire à une estimation plus juste de sa capacité de discrimination qu'une estimation basée sur la fréquence des documents seulement.

Par conséquent, dans ce travail, nous proposons une méthode de sélection des features (termes) au niveau des catégories, où chaque terme est pondéré en fonction de son DF et TF pour chaque catégorie, puis un score final est calculé pour toutes les catégories.

3.2 Inconvénients des métriques de sélection

Les méthodes de sélection des features pour la classification des textes, peuvent être regroupées en deux catégories principales : les méthodes basées sur la fréquence des documents (DF) et les méthodes basées sur la fréquence des termes (TF).

Les méthodes de sélection des features, telles que Chi2, IG, MI, etc. ignorent la fréquence du terme dans le document et comptent que les documents dans lesquels le terme apparaît. C'est l'inconvénient majeur que les métriques courantes ne prennent pas en compte le TF en considérant que le DF, où le terme qui apparaît plus d'une fois (même à haute fréquence) dans un document est le même qui apparaît une fois dans un document.

3.3 Utilité de fréquence des termes (Term-Frequency)

La fréquence des termes a attiré plus d'attention dans la sélection des features. La fréquence du terme est une mesure de la fréquence d'occurrences du terme dans un document. Le calcul le plus simple consiste simplement à compter le nombre de fois qu'un mot apparaît. Cependant, il existe des moyens de modifier cette valeur en fonction de la longueur du document ou de la fréquence des mots les plus courants dans le document.
Parmi les métriques de sélection basées sur TF, nous trouvons celle proposée par[38], définie comme suit :

$$w_(u,k) = \frac{p(t_u|c=k)}{\sum_{j=1}^{k} p(t_u|c=j)}$$

(3.1)

Où $P(t_u|c=k)$ est la probabilité conditionnelle du terme t étant donné une étiquette de classe k.

La fréquence d'un terme (TF) est le pourcentage d'apparition de ce terme dans un document.
Soit D : un document et T : un terme, la fréquence du terme T dans le document D, que l'on note par $TF_{(t,d)}$, est calculée par cette formule :

$$TF_{t,d} = \frac{n_{(t,d)}}{\sum_{terme} n_{terme,d}}$$

(3.2)

Où :
$n_{terme,d}$: le nombre d'occurrences du terme T dans le document D.
$\sum_{terme} n_{terme,d}$: La somme des occurrences de tous les termes qui apparaissent dans le document D [28].

Un document avec 10 occurrences du terme est plus pertinent qu'un document avec une seule occurrence du terme.
La plupart des algorithmes de sélection de features tiennent pleinement compte de la fréquence des documents. Mais ils ignorent l'influence de la fréquence du terme et les interactions entre la fréquence des documents et la fréquence du mot.

La fréquence des documents fait référence au nombre de documents contenant le terme, qui peut être au niveau de la catégorie ou au niveau de l'ensemble de données. La fréquence des mots fait référence au nombre de termes, qui peut être divisé en fréquence au niveau du document, fréquence au niveau de la catégorie et fréquence au niveau de l'ensemble de données.

La fréquence du terme au niveau de la catégorie et la fréquence du terme au niveau de l'ensemble de données peuvent évidemment influencer le jugement des informations de de la catégorie, tandis que la fréquence du terme au niveau du document

peut exprimer la contribution réelle d'un mot dans le document.

Par exemple, deux mots dans un document ont des fréquences de terme de 2 et 20, mais ils ont les mêmes fréquences de document de 1. Ces deux mots ont la même fréquence de document mais leurs contributions au document sont significativement différentes.

3.4 Méthode proposée

Notre travail vise à proposer un schéma de pondération et de sélection des terme pertinents en se basant sur la fréquence des documents et les termes entre les catégories.
Nous pensons que « les apparitions multiples d'un terme dans un document sont plus importantes que les apparitions uniques », c'est-à- dire « apparaître une fois dans un document » est différent de « apparaître au moins deux fois dans un document ».

3.4.1 Prétraitement des textes

Dans tout processus de classification de texte, l'étape de prétraitement joue un rôle décisif sur la performance du classifieur. Cela dit, un bon nettoyage peut améliorer les résultats indépendamment de la possibilité d'appliquer le classificateur, et vice-versa. Le nettoyage peut supprimer tous les bruits, les mots inutiles, les mots qui contribuent négativement dans le contexte de la classification et sa contribution au texte, les balises html, les caractères spéciaux, les stop-words, les chiffres, etc.

Le but de prétraitement des textes est de représenter chaque document comme un vecteur de features qui divise le texte en mots individuels. Les documents texte sont formés comme des transactions.

Malgré la suppression des features non informatifs, la dimensionnalité de l'espace représentatif peut encore être trop élevée [27].

3.4.2 Construction du vocabulaire

Chaque texte dans l'ensemble d'apprentissage est représenté comme un vecteur sous la forme (x, c), où $x \in R^n$, et c est l'étiquette de la classe. Chaque dimension de cet espace représente un feature unique de ce vecteur et son poids qui est calculé par la fréquence d'occurrence de chaque features dans ce document texte. Cette étude représentera chaque vecteur de document d comme $d = (w_1, w_2, ..., w_n)$.
Où w_i est le poids du ième terme du document d. Cette représentation est appelée représentation de données ou modèle d'espace vectoriel.

Par exemple dans la phrase " I am a student at Guelma university ", le vocabulaire extrait de cette phrase est la liste des mots V= [I, am, a, student, at, Guelma, university], mais après prétraitement la liste devient V= [student, Guelma, university] et cela après élimination des stop-words comme I, am, a, at.

3.4.3 Fréquence des termes et fréquence des documents

Pour chaque catégorie, nous calculons le nombre d'occurrences TF : term-frequency) de chaque mot du vocabulaire, et son DF (document-frequency) avec :
La fréquence d'un terme est le nombre de fois qu'un terme apparaît dans un document. La fréquence d'un terme t est calculée comme suit :

$$TF_{t,d} = \frac{\text{nombre de dans d}}{\text{nombre de mots dans d}} \tag{3.3}$$

La fréquence des documents, c'est le nombre total de documents contenant le terme dans le corpus, Document Frequency (DF) d'un terme t est calculée comme suit :

$$DF_t = \text{Nombre de document contenant le terme t}$$

(3.4)

Cela mesure l'importance du document dans l'ensemble du corpus, ceci est très similaire à TF. La seule différence est que TF est le compteur de fréquence pour un terme t dans le document d, et DF est le nombre d'occurrences du terme t dans l'ensemble de documents N [3].
Par exemple, si on a une catégorie (c) qui contient 3 documents et nous cherchons si un terme t apparait ou non dans chaque document, avec :
1er document : TF= 3
2ème document : TF= 1
3ème document : TF=2
DF (t, d) =3
TF (t, c) =6
Le 1er document a une grande importance par rapport au 2ème et 3ème.

3.4.4 Schéma de pondération

La fréquence d'apparition d'un terme dans une classe est un bon indicateur de l'importance de ce terme, plus un terme est fréquent dans un document plus il est important dans la description de ce document. Les termes importants doivent avoir un poids fort.

Pour calculer le poids (score) d'un document :

Pour chaque catégorie on doit calculer le poids $W(t, C)$. Par exemple, pour deux catégories C_1, C_2, on a :

Pour C_1 : $W(T, C_1) = TF(T, C_1)DF(T, C_1)$

Pour C_2 : $W(T, C_2) = TF(T, C_2)DF(T, C_2)$

Le poids général : $W(T) = W(T, C_1) - W(T, C_2)$

Pour que la valeur soit toujours positive (on a utilisé la valeur absolue) :

$$W(T) = |W(T, C_1) - W(T, C_2)| \tag{3.5}$$

Pour limiter les valeurs de grandes espaces, on a employé le logarithme (log) aux formules :

$$W(T) = Log|W(T, C_1) - W(T, C_2)| \tag{3.6}$$

Au cas où la différence entre les deux catégories sera nulle, le Logarithme est indéfini à 0 pour cela on a ajouté 1 au cas où la différence donne un résultat nul :

$$W(T) = Log|W(T, C_1) - W(T, C_2) + 1| \tag{3.7}$$

L'algorithme suivant résume les étapes de pondération :

Algorithme 2 schéma de pondération

Entrées: $V = \{mot\ uniques\}; C = \{c_1, c_2, \ldots ...c_n\};$
Sorties: liste des mots pondérés;

Pour $c_j \in c$ **Faire**
  Pour tout c_j**dans**c **Faire**
    Pour tout t**dans**v **Faire**
      Calculer :
      $DF \leftarrow DF(t, c_j)$
      $TF \leftarrow TF(t, c_j)$
      $W(t, c_j) = TF(T, c_j) \times (DF(T, c_j)$
    Fin Pour
  $W(T) = Log|W(T, C_1) - W(T, C_2) + 1|$
  Fin Pour
Fin Pour

3.5 Conclusion

Dans ce chapitre, nous avons défini les étapes de notre approche proposée qui est un schéma de pondération basé sur la distribution des fréquences des documents et des termes entre catégories, nous combinons la fréquence des documents avec la fréquence des termes. Avec cette approche les métriques de sélection de features définies avec la fréquence des termes deviennent plus performantes que celles définies avec la fréquence des documents.

Le chapitre suivant est dédié à l'implémentation de notre projet en procédant à des expérimentations et des discussions.

CHAPITRE 4
EXPÉRIMENTATION ET EVALUATION DES RÉSULTATS

4.1 Introduction

Nous avons présenté dans le chapitre précédent la conception détaillée de notre méthode de sélection des features pour la classification des textes. L'objectif de ce chapitre est de présenter les étapes de l'implémentation de la méthode proposée, nous commençons par décrire l'environnement de développement, les différentes bibliothèques utilisées pour mettre en œuvre la méthode proposée, ainsi que l'explication de l'application et toutes ses fonctionnalités. La performance de la méthode proposée en termes du score F1 est comparée avec celles des méthodes : IG, MI, CH2, et DF.

4.2 Langages et outils de développement

Pour l'implémentation de notre projet, nous avons utilisé Python version 3.8.5 comme langage de programmation et Jupyter Notebook comme environnement de développement intégré (IDE).

python : Python est un langage de programmation libre de haut niveau, créé par Guido van Rossum, sa première publication a été en 1991. Python est un langage interprété, c'est-à-dire qu'il peut être exécuté sans compilation. Il est caractérisé par un système de typage dynamique, une gestion de mémoire automatique et une bibliothèque multifonctionnelle complète. Ce langage prend en charge plusieurs paradigmes de programmation, et il est adaptable pour tous les systèmes d'exploitation. Python est un langage très approprié pour les apprenants débutants, mais il est aussi très motivant pour les utilisateurs expérimentés [17].Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données et dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages [4].

FIGURE 4.1 – Logo du langage de programmation Python.

Jupyter Notebook : Jupyter Notebook est un notebook de calcul open source, est l'application Web originale pour la création et le partage de documents informatiques. Il offre une expérience simple, rationalisée et centrée sur les documents.
Jupyter est une communauté de passionnés de données qui croient en la puissance des outils et des normes ouverts pour l'éducation, la recherche et l'analyse de données, Jupyter prend en charge plus de 40 langages de programmation, dont Python, R, Julia et Scala. Le code peut produire une sortie riche et interactive : HTML, images, vidéos, Latex et types MIME personnalisés [5].

FIGURE 4.2 – Logo du Jupyter Notebook.

4.2.1 Les Bibliothèques Python utilisées

Scikit-learn : (Sklearn) est la bibliothèque la plus utile et la plus robuste pour l'apprentissage automatique. Il fournit une sélection d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression, le regroupement et la réduction de la dimensionnalité via une interface de cohérence en Python. Cette bibliothèque, qui est en grande partie écrite en Python, est construite sur NumPy, SciPy et Matplotlib [6].

Matplotlib : est une bibliothèque complète pour la création de visualisations statiques, animées et interactives en Python [7].

Numpy : La bibliothèque NumPy permet d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux. NumPy propose des fonctions mathématiques complètes, des générateurs de nombres aléatoires, des routines d'algèbre linéaire, des transformées de Fourier. La syntaxe de haut niveau de

NumPy le rend accessible et productif pour les programmeurs de tous horizons ou niveaux d'expérience [8].

Pandas : est un outil d'analyse et de manipulation de données open source rapide, puissant, flexible et facile à utiliser, construit sur le langage de programmation Python [9].

NLTK (Natural Language Toolkit) : est une plate-forme leader pour la création de programmes Python et pour travailler avec des données de langage humain. Il fournit des interfaces faciles à utiliser à plus de 50 corpus et ressources lexicales telles que WordNet, ainsi qu'une suite de bibliothèques de traitement de texte pour la classification, la tokenisation, la radicalisation, etc [10].

4.3 Démarche expérimentale

4.3.1 Présentation des Datasets

Nous avons utilisé trois datasets publiquement disponibles pour évaluer notre approche :

Movie Reviews : contient 1 000 critiques positives et 1 000 critiques négatives, collectées sur imdb.com, et utilisées pour l'analyse des sentiments. La première version a été publiée en 2002 et la version mise à jour a été publiée en 2004, appelée "polarity dataset v2.0" [34].

	text	category
0	plot : two teen couples go to a church party ,...	0
1	the happy bastard ' s quick movie review damn ...	0
2	it is movies like these that make a jaded movi...	0
3	" quest for camelot " is warner bros . ' first...	0
4	synopsis : a mentally unstable man undergoing ...	0
...	...	...
1995	wow ! what a movie . it ' s everything a movie...	1
1996	richard gere can be a commanding actor , but h...	1
1997	glory -- starring matthew broderick , denzel w...	1
1998	steven spielberg ' s second epic film on world...	1
1999	truman (" true - man ") burbank is the perfe...	1

2000 rows × 2 columns

FIGURE 4.3 – Le dataset Movie Reviews .

BBC Text : est constituée d'ensembles de données d'articles d'actualités, provenant de BBC News, fournis à titre de référence pour la recherche en apprentissage automatique. Les données d'origine sont traitées pour former un fichier csv unique pour une utilisation facile, le titre de l'actualité et le nom du fichier texte associé sont conservés avec le contenu de l'actualité et sa catégorie. Il se compose de 2225 documents du site Web d'actualités de BBC correspondant à des articles dans 5 domaines

thématiques (business, entrainement, politique, sport, technologie). La figure suivante présente le dataset BBC après avoir choisi seulement les deux catégories : tech et sport.

		text	category
category			
sport	**0**	mourinho escape fa charge bos mourinho face fo...	0
	1	stevens england england bath prop stevens star...	0
	2	fume robinson blast official england coach rob...	0
	3	rover reject ferguson bid blackburn reject bid...	0
	4	mourinho plot impressive win fulham confirm po...	0
...	**...**	...	...
tech	**95**	game learn play god game player control virtua...	1
	96	text message aid disaster recovery text messag...	1
	97	uk broadband speed injection broadband rapid r...	1
	98	blog mainstream web log blog estimate web numb...	1
	99	solution net security fear fake bank mail phis...	1

200 rows × 2 columns

FIGURE 4.4 – Le dataset BBC Text .

Twitter US Airline Sentiment : L'ensemble de données contient environ 15 000 tweets, collectés à partir de février 2015 sur diverses critiques de compagnies aériennes. Chaque avis est étiqueté comme positif, négatif ou neutre [18].

1) Prétraitement des données :

Le prétraitement des données est le processus de transformation des données brutes dans un format compréhensible. C'est aussi une étape importante dans l'exploration de données car nous ne pouvons pas travailler avec des données brutes. La qualité des données doit être vérifiée avant d'appliquer des algorithmes d'apprentissage automatique ou d'exploration de données.
Les opérations de prétraitement qui nous affections sur les trois datasets sont :

1. **Convertir tous les textes en minuscules**
2. **Lemmatisation**
3. **Supprimer les éventuelles balises html**
4. **Supprimer des chiffres**
5. **Supprimer les caractères spéciaux**
6. **Supprimer les mots de moins de 3 caractères**
7. **Supprimer les espaces supplémentaires entre les mots**
8. **Supprimer les espaces gauches et droit**
9. **Supprimer les mots vides**

Les figures ci-dessous montrent les trois datasets après prétraitement.

	text	category	text_clean
0	plot : two teen couples go to a church party ,...	0	plot two teen couple church party drink drive ...
1	the happy bastard ' s quick movie review damn ...	0	happy bastard quick movie review damn bug get ...
2	it is movies like these that make a jaded movi...	0	movie like make jaded movie viewer thankful in...
3	" quest for camelot " is warner bros . ' first...	0	quest camelot warner bros first feature length...
4	synopsis : a mentally unstable man undergoing ...	0	synopsis mentally unstable man undergo psychot...
...	...	...	...
1995	wow ! what a movie . it ' s everything a movie...	1	wow movie everything movie funny dramatic inte...
1996	richard gere can be a commanding actor , but h...	1	richard gere commanding actor always great fil...
1997	glory -- starring matthew broderick , denzel w...	1	glory star matthew broderick denzel washington...
1998	steven spielberg ' s second epic film on world...	1	steven spielberg second epic film world war un...
1999	truman (" true - man ") burbank is the perfe...	1	truman true man burbank perfect name jim carre...

2000 rows × 3 columns

FIGURE 4.5 – Movie Reviews après prétraitement.

		text	category
category			
sport	**0**	fa probe crowd trouble fa action trouble mar w...	0
	1	umaga ready fearsome lion black captain umaga ...	0
	2	ferguson rue failure cut gap bos sir ferguson ...	0
	3	eyeing world gold olympic success determine ba...	0
	4	martinez vinci challenge veteran spaniard mart...	0
...	**...**	...	...
tech	**145**	musical future phone analyst thompson future h...	1
	146	video phone deaf people deaf people prefer com...	1
	147	jeeves join web log market jeeves buy blogline...	1
	148	movie body hit peer peer net movie industry st...	1
	149	tough rule ringtone seller firm flout rule rin...	1

300 rows × 2 columns

FIGURE 4.6 – BBC Text après prétraitement.

	Unnamed: 0	text	category
0	1	virginamerica plus add commercial experience t...	1
1	3	virginamerica really aggressive blast obnoxiou...	0
2	4	virginamerica really big bad thing	0
3	5	virginamerica seriously would pay flight seat ...	0
4	6	virginamerica yes nearly every time fly ear wo...	1
...	...	...	...
11536	14633	americanair flight cancel flightled leave tomo...	0
11537	14634	americanair right cue delays	0
11538	14635	americanair thank get different flight chicago	1
11539	14636	americanair leave minute late flight warning c...	0
11540	14638	americanair money change flight answer phone s...	0

11541 rows × 3 columns

FIGURE 4.7 – Twitter US Airline Sentiment après prétraitement .

4.3.2 Implémentation de TFDF (Term Frequency - Document Frequency)

Nous avons implémenté notre méthode TFDF, et comme mentionné dans le chapitre précédent, nous avons fait une combinaison entre le TF et DF, premièrement nous avons calculé à chaque terme son score en utilisant le schéma de pondération proposée, après, nous avons fait une classification en utilisant les features sélectionnés par TFDF et cela pour tester sa performance par rapport aux autres métriques utilisées.

Les figures ci-dessous montrent un aperçu sur les scores affectés par TFDF pour les trois datasets.

```
{'mulan': 4.554199073851086,
 'flynt': 4.357183547799859,
 'lebowski': 3.9899582307621175,
 'seagal': 3.604538305680185,
 'homer': 3.49902963089919,
 'jawbreaker': 3.3705233744773406,
 'webb': 3.3705233744773406,
 'jude': 3.3705233744773406,
 'bulworth': 3.359284342214698,
 'benigni': 3.3356547723673566,
 'winslet': 3.3356547723673566,
 'magoo': 3.3356547723673566,
 'hudson': 3.2995337278856547,
 'redford': 3.2995337278856547,
 'jakob': 3.2620674234509743,
 'pleasantville': 3.1826727762042397,
 'skinhead': 3.1826727762042397,
 'nello': 3.1404983965137254,
 'whale': 3.1034059013776494,
```

FIGURE 4.8 – Un aperçu sur les termes pondérés de Movie Reviews.

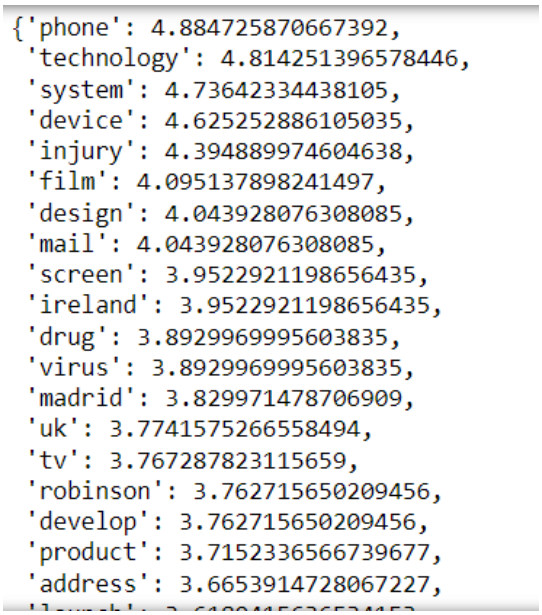


FIGURE 4.9 – Un aperçu sur les termes pondérés de BBC Text.

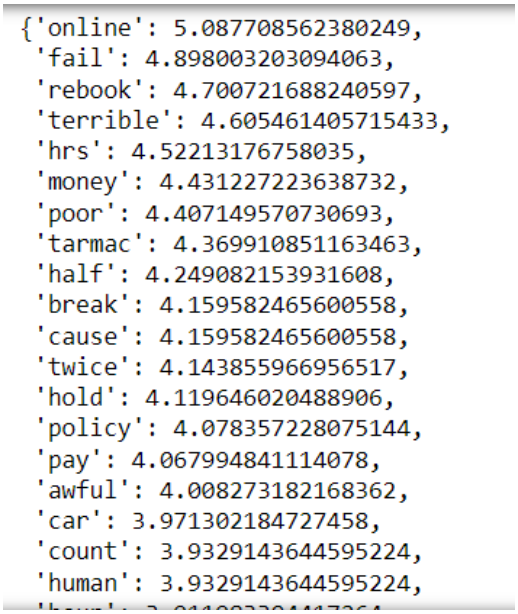


FIGURE 4.10 – Un aperçu sur les termes pondérés de Twitter US Airline Sentiment.

4.3.3 Classification

Pour évaluer l'efficacité de la méthode proposée de sélection des features, les algorithmes de classification doivent être entraînés et testés à l'aide de différents ensembles de données. Les algorithmes les plus utilisés sont Support Vector Machine (SVM) et Naive Bayes (NB) car ils sont reconnus comme ayant de bons résultats dans la tâche de classification des textes [35].

Pour cela, nous avons effectué une classification sur les trois datasets avec les deux algorithmes de classification SVM et NB avec les métriques de sélection de features les plus courantes (IG, MI, CH2, et DF).

Notre but des expérimentations est de tester la performance de notre approche TFDF (Terme Frequency-Document Frequency) et la comparer avec les méthodes de sélection IG, MI, CH2, et DF en termes des score F1 de SVM et NB.

Dans cette section sont présentés les différents résultats des classifications appliquées sur les trois datasets. La présentation commence par les résultats de classification sans sélection de features, puis avec les différentes méthodes de sélection enfin notre approche proposé TFDF.

1) Classification sans sélection de features
Dans le Tableau 4.4, les résultats de classification sans sélection des features sont présentés en termes des meilleurs score F1 obtenus par SVM et NB.

Dataset	F1 score	
	SVM	NB
Movie reviews	0.818182	0.795455
BBC text	0.984848	0.984848
Twitters US Airline Sentiment	0.906275	0.841166

TABLE 4.1 – Résultats de classification sans sélection de features.

2) Classification avec sélection de features

Parmi les méthodes de sélection de features, nous avons choisi quatre métriques les plus utilisées Information Gain (IG), Mutuel Information (MI), CH-Square (CH2) et la fréquence du document (DF) pour faire la classification avec les mêmes classifiers SVM et NB.

Les résultats de classification obtenus par SVM et NB pour ces métriques sont comparés avec ceux obtenus pour notre méthode TFDF.
Pour le dataset Movie Reviews, et d'après les résultats du tableau 4.2, notre méthode TFDF est significativement meilleure en fonction de du score F1et le nombre de features. Sachant que ces quatre métriques que nous comparons avec la méthode TFDF sont très compétitives. Les résultats obtenus démontrent clairement l'efficacité de notre méthode TFDF.

DATASET	**Movie Reviews**			
ALGORITHME	**SVM**		NB	
	Nb. Features	**Score F1**	**Nb. Features**	**Score F1**
IG	700	0.865152	900	0.851515
MI	21800	0.836364	29400	0.800000
CH2	1100	0.887879	6100	0.887879
DF	2500	0.836364	1800	0.807576
TFDF	4600	**0.880303**	4600	**0.890909**

TABLE 4.2 – Résultats de classification pour le dataset Movie Reviews après sélection de features

Notons que les résultats montrés dans le tableau 4.3 indiquent que notre méthode TFDF proposé dépasse considérablement toutes les autres métriques en termes de F1 et le nombre optimal de features sélectionnés.

Pour le dataset Dataset BBC Text, les résultats montrés sur le tableau 4.2 évaluent généralement l'efficacité de la méthode proposée par rapport aux autres métriques de sélections.

DATASET	**BBC Text**			
ALGORITHME	**SVM**		**NB**	
	Nb. Features	**Score F1**	**Nb. Features**	**Score F1**
IG	200	0.989899	200	0.969697
MI	100	0.989899	500	0.989899
CH2	700	1.000000	200	0.989899
DF	100	0.989899	300	0.969697
TFDF	600	**0.979798**	500	**0.969697**

TABLE 4.3 – Résultats de classification pour le dataset BBC Text après sélection de features

Le F1 score de notre méthode est relativement inférieur aux autres métriques par rapport aux autres métriques, mais si nous comparons le nombre de features sélectionné nous assurons que notre approche est meilleure que les autres en termes de sélection optimale du sous-ensemble de features pertinents.

Les mêmes expérimentations sont effectuées aussi pour le dataset Twitter US Airline Sentiment, où les résultats sont montrés sur le Tableau 4.4.

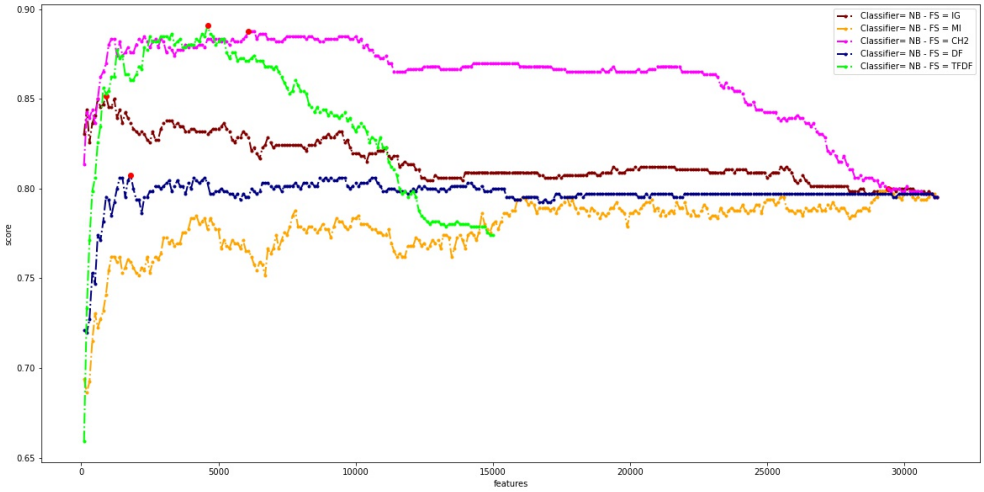
DATASET	**BBC Text**			
ALGORITHME	**SVM**		**NB**	
	Nb. Features	**Score F1**	**Nb. Features**	**Score F1**
IG	2500	0.918614	800	0.912313
MI	9300	0.907587	900	0.875033
CH2	230	0.923602	700	0.911000
DF	6700	0.907587	1100	0.891048
TFDF	9300	**0.910213**	2800	**0.842216**

TABLE 4.4 – Résultats de classification pour le dataset Twitter US Airline Sentiment après sélection de features.

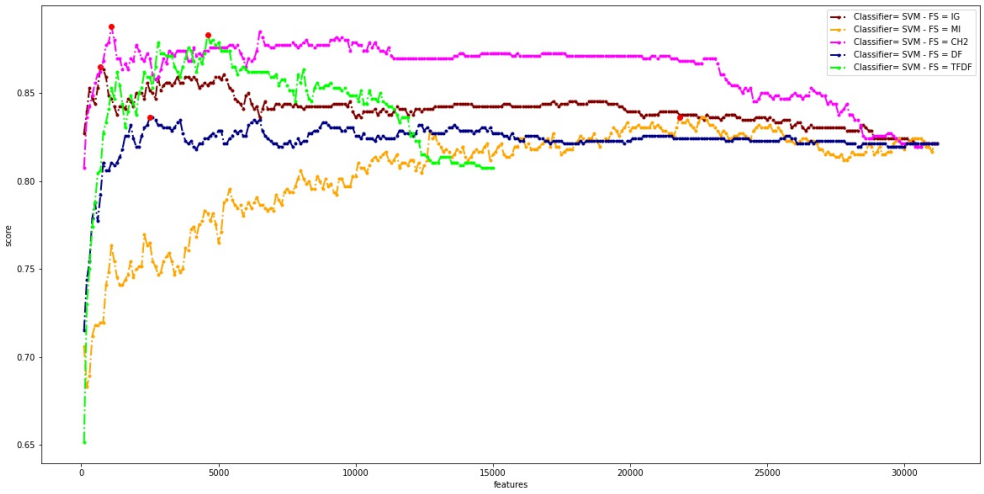
A partir de ce tableau, nous remarquons que le score de la méthode proposé TFDF est inférieur un peu par rapport à IG et CH2, c'est normal parce que les textes (tweets) sont courts d'où le TF sera petit mais cela n'indique pas que notre approche n'est pas performante parce que le score de chaque feature est calculé en fonction de TF et DF à la fois d'où certains features pertinents ne seront pas considérés.

3)Discussion des résultats
Après avoir pondéré les features par chaque métrique : IG, MI, CH2, DF et la méthode proposée TFDF, nous avons trié les features selon leurs scores de plus petit au plus grand.
Pour chaque dataset, nous avons effectué une classification en utilisant les deux algorithmes SVM et NB, où seuillage de 100 features a été utilisé, c'est-à-dire à chaque itération nous ajoutons 100 features à l'ensemble des features déjà sélectionné, puis nous calculons le score F1 correspondant.

Les figures suivantes montrent les scores F1 obtenus pour chaque dataset :

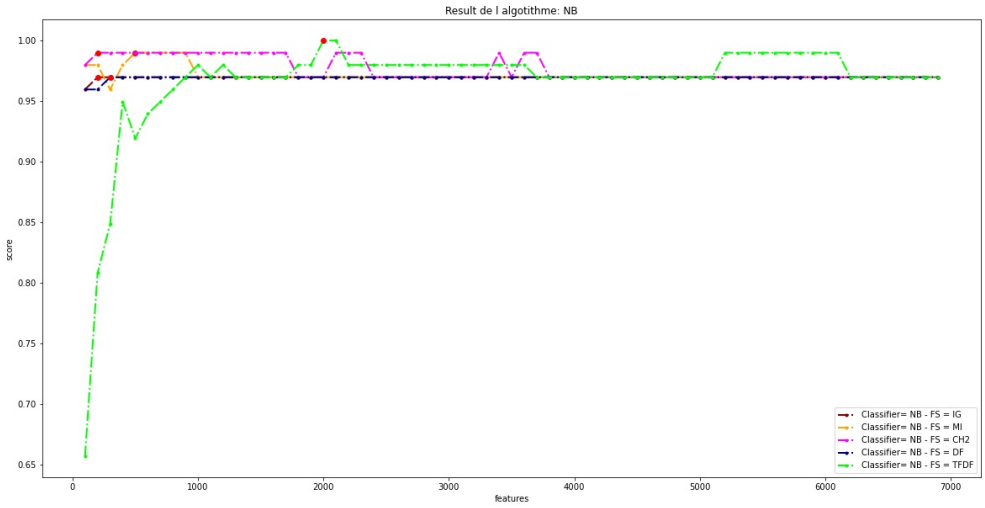


(a)

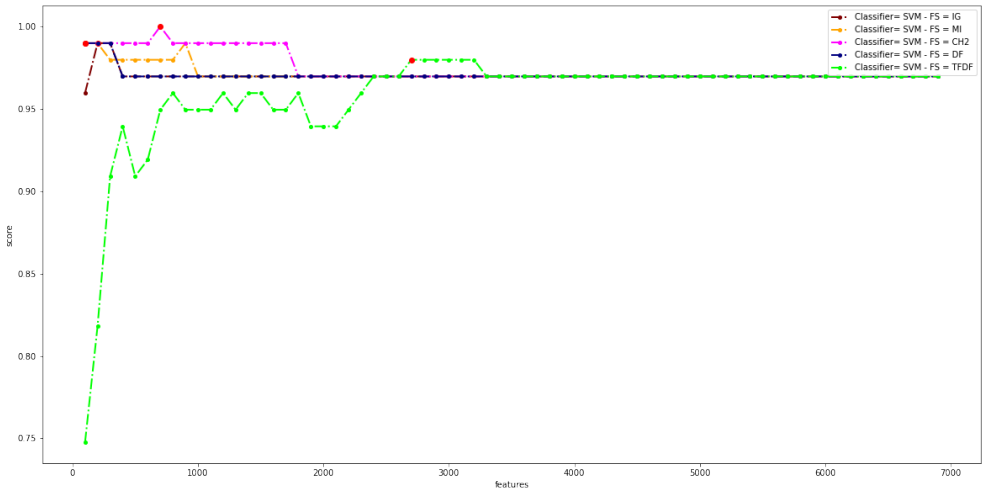


(b)

FIGURE 4.11 – Scores F1 pour Movie Reviews (a) NB, (b) SVM.

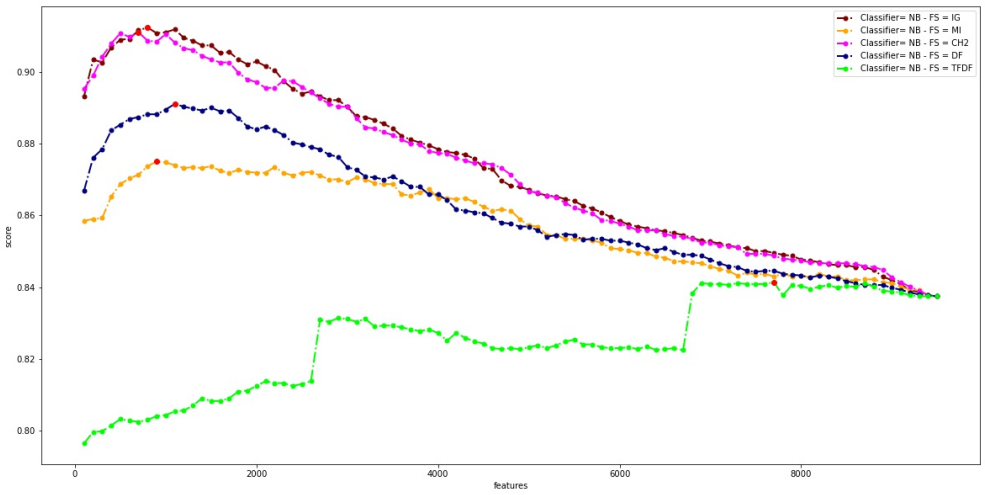


(a)

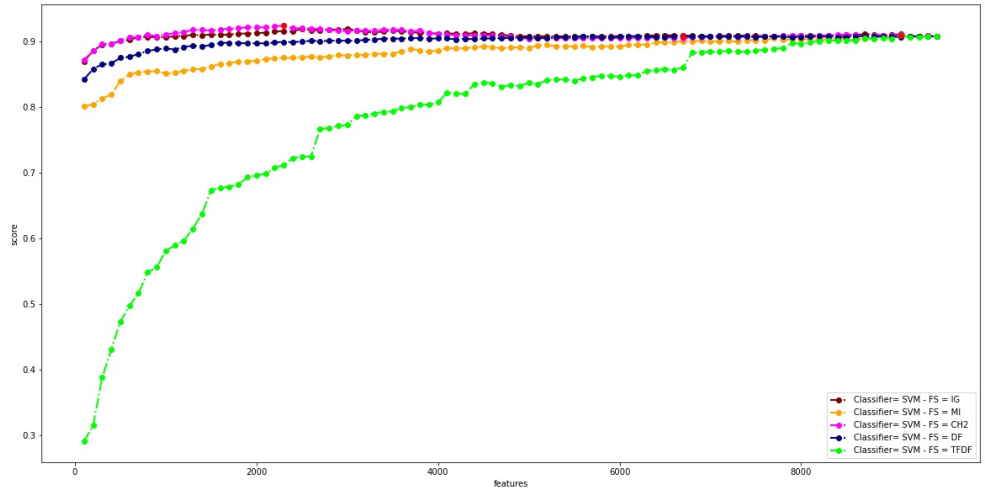


(b)

FIGURE 4.12 – Scores F1 pour BBC Text (a) NB, (b) SVM.



(a)



(b)

FIGURE 4.13 – Scores F1 pour Tweets US Airline Sentiment (a) NB, (b) SVM.

En termes des scores F1 obtenus, et le nombres de features sélectionnés, les expérimentations ont montré l'efficacité de notre méthode TFDF comparées aux méthodes de sélection des données catégoriales le plus répandues IG, MI, CH2 et DF.

— Le TF (Term-Frequency) joue un rôle important dans la détermination de la capacité de discrimination des termes, où un terme qui apparait fréquemment à l'intérieur des documents et distribué entre plusieurs documents (DF) est considéré comme plus pertinent par rapport aux termes qui sont pondérés par leurs DFs seulement.

— La taille du document à une influence sur la capacité de discrimination des termes qui apparait fréquemment ou rarement, par exemple dans des documents longs comme le cas du dataset Movie Reviews, notre méthode TFDF est plus performante que les autres méthodes, tandis que pour le dataset Tweets US Airline Sentiment, notre méthode n'as pas prouvé efficacement sa performance à cause de la petite taille des messages Tweeter (Tweets) où le TF calculé aura une influence sur les scores des termes.

4.4 Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de notre méthode TFDF (Term Frequency - Document Frequency) où les résultats obtenus ont montré son efficacité comparée aux méthodes de sélection des features IG, MI, CH2 et DF.

Notre méthode de sélection de features TFDF est une méthode statistique qui combine le TF (Terme-Frequency) et le DF(Document-Frequency) pour mesurer l'importance des termes.

En utilisant les deux algorithmes les plus efficaces pour la classification des textes, les résultats obtenus montrent la performance de notre méthode, où la considération du TF dans l'évaluation des termes a prouvé son importance, contrairement aux autres méthodes statistiques qui se basent seulement sur le DF pour évaluer les termes.

CONCLUSION GÉNÉRALE

La classification de textes peut être utilisée pour gérer plusieurs problèmes du monde réel. Un problème majeur avec la classification ou le regroupement de textes est la haute dimensionnalité des features (termes), qui peut être résolu en utilisant les métriques de sélection qui permettent d'évaluer l'importance des termes puis filtrer ceux qui sont plus pertinents pour le processus de classification.
Les méthodes de sélection des features ont reçu beaucoup d'attention de la communauté de classification des textes, pour réduire la dimensionnalité, supprimer les features non pertinents et augmenter la précision de l'apprentissage.
Parmi les méthodes les plus populaires, nous trouvons : Information Gain (IG), Mutual Information (MI), Chi-square (Chi2), et Document-Frequency (DF) qui sont des métriques statistiques basée sur la distribution probabiliste des mots entre les documents. En d'autres termes, elles utilisent le nombre de documents dans lesquels le mot appartient ou n'appartient pas pour calculer son importance. L'inconvénient majeur de ces métriques est que le nombre d'apparitions dans le même document (TF : Term-Frequency) n'est pas considéré ce qui peut réduire l'importance des termes qui apparient fréquemment à l'intérieur des documents.
Dans ce travail, nous avons proposé une méthode de sélection des features pour la classification des textes qui permettent d'évaluer les termes au niveau de chaque catégorie (classe) en prenant en compte le DF et le TF de chaque terme, puis un score final pour toute les catégories.
Les résultats d'expérimentation ont montré l'efficacité de notre méthode en la comparant avec IG, MI, Chi2 et DF.

BIBLIOGRAPHIE

[1] https://www.futura-sciences.com/tech/definitions/informatique-reseau-neuronal601/. (dernier accès : 05 /2022.)

[2] <http://scikit-learn.org/stable/modules/svm.html>. (dernier accès : 10/06/2020.)

[3] https://towardsdatascience.com/tf-term-frequency-idf-inverse-document-frequency-from-scratch-in-python-6c2b61b78558. (dernier accès : 05 /2022.)

[4] https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/#:~:text=Qu. (dernier accès : 03/05/2022.)

[5] https://jupyter.org/. (dernier accès : 02/05/2022.)

[6] https://www.tutorialspoint.com/scikit_learn/index.htm. (dernier accès : 05/2020.)

[7] https://matplotlib.org/. (dernier accès : 05 /2022.)

[8] https://numpy.org/. (dernier accès : 05 /2022.)

[9] https://pandas.pydata.org/. (dernier accès : 05 /2022.)

[10] https://www.nltk.org/. (dernier accès : 05 /2022.)

[11] Soraya AZZOUG. « Sélection automatique des caractéristiques pour la reconnaissance des chiffres manuscrits par la méthode F-score ». Thèse de doct. Faculté d'Electronique et d'Informatique, 2013.

[12] Said BAHASSINE et al. « Feature selection using an improved Chi-square for Arabic text classification ». In : *Journal of King Saud University-Computer and Information Sciences* 32.2 (2020), p. 225-231.

[13] Mouhoub BELAZZOUG. « Apprentissage statistique pour l'extraction des relations à partir de textes ». Thèse de doct. 2021.

[14] Nadia BENAHMED. *Optimisation de réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés : Sélection et pondération des primitives par algorithmes génétiques*. École de technologie supérieure, 2002.

[15] Eric W BROWN et Herb A CHONG. « The GURU system in TREC-6 ». In : *NIST SPECIAL PUBLICATION SP* (1998), p. 535-540.

[16] Hassan CHOUAIB. « Sélection de caractéristiques : méthodes et applications ». In : *Paris Descartes University : Paris, France* (2011).

[17] Younes DERFOUFI. « Programmation en langage Python ». In : (2019).

[18] Xiaotong DUAN, Tianshu JI et Wanyi QIAN. *Twitter US Airline Recommendation Prediction*. 2016.

[19] Martin EKLUND. *Comparing Feature Extraction Methods and Effects of Pre-Processing Methods for Multi-Label Classification of Textual Data*. 2018.

[20] George FORMAN. « A pitfall and solution in multi-class feature selection for text classification ». In : *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 38.

[21] Corrado GINI. « Variabilità e mutabilità (Variability and Mutability) ». In : *Cuppini, Bologna* 156 (1912).

[22] Alain GIRARD. « Exploration d'un algorithme génétique et d'un arbre de décision à des fins de catégorisation ». Thèse de doct. Université du Québec à Trois-Rivières, 2007.

[23] Veneta HARALAMPIEVA et Gavin BROWN. « Evaluation of Mutual information versus Gini index for stable feature selection ». In : (2016).

[24] Hassane HILALI. « Application de la classification textuelle pour l'extraction des règles d'association maximales ». Thèse de doct. Université du Québec à Trois-Rivières, 2009.

[25] Anil K JAIN et Douglas ZONGKER. « Representation and recognition of handwritten digits using deformable templates ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.12 (1997), p. 1386-1390.

[26] Lakhmi C JAIN, Shing Chiang TAN et Chee Peng LIM. « An introduction to computational intelligence paradigms ». In : *Computational Intelligence Paradigms*. Springer, 2008, p. 1-23.

[27] Ammar Ismael KADHIM, Yu-N CHEAH et Nurul Hashimah AHAMED. « Text document preprocessing and dimension reduction techniques for text document clustering ». In : *2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology*. IEEE. 2014, p. 69-73.

[28] Ali LABIAD. « Sélection des mots clés basée sur la classification et l'extraction des règles d'association ». Thèse de doct. Université du Québec à Trois-Rivières, 2017.

[29] Sungjick LEE et Han-joon KIM. « News keyword extraction for topic tracking ». In : *2008 fourth international conference on networked computing and advanced information management*. T. 2. IEEE. 2008, p. 554-559.

[30] Huan LIU et Rudy SETIONO. « Chi2 : Feature selection and discretization of numeric attributes ». In : *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*. IEEE. 1995, p. 388-391.

[31] Gjorgji MADJAROV et al. « An extensive experimental comparison of methods for multi-label learning ». In : *Pattern recognition* 45.9 (2012), p. 3084-3104.

[32] Choayb OUALI. « Classification automatique de textes ». Thèse de doct. UNIVERSITE MOHAMED BOUDIAF M'SILA : FACULTE DES MATHEMATIQUES ET DE L ..., 2014.

[33] Lahlou OUCHIHA. « Classification supervisée de documents : étude comparative ». Thèse de doct. Université du Québec en Outaouais, 2016.