REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre :…./M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Système d'Information Optimisation et Décision(SIOD)**

Classification Automatique des documents textuels

Par :

Abdelali Tamene

Soutenu le../../…. devant le jury composé de :

. . .	grade	Président
Abdelli belkacem	MCD	Rapporteur
. . .	grade	Examinateur

Année universitaire 2020-2021

TABLE DE MATIERES

Table des Figures

Dédicaces

À

mes parents,

mes frères et ma soeur,

toute la famille,

et mes amis,

je dédie ce modeste travail.

Remerciements

Je tiens premièrement à prosterner remerciant Allah le tout puissant de m'avoir donné le courage et la patience pour terminer ce travail.

Je voudrais dans un premier temps remercier mon Encadreur, Monsieur **Abdelli belkacem,** de m'avoir encadré, orienté, aidé et conseillé.

Je tiens à remercier spécialement mon ami *Masoud* pour ses conseils et ses critiques, Je remercie mes amis *Bacha et Ayoub , Yasine, et Youcef Rayeh.*

Je remercie mes très chers parents, et mes frères et ma soeur *yakine*, pour leur encouragements Enfin, Je voudrais exprimer ma reconnaissance envers mon amie *Assia* , et tout membre de ma famille m'a apporté un soutien moral tout au long de ma démarche.

J'adresse mes sincères remerciements aux membres du jury, qui ont accepté d'évaluer mon travail.

Résumé/Abstract

Résumé

Le texte est important car il contient une énorme quantité d'informations dont nous pouvons Bénéficier en l'analysant son contenu.
Le travail effectué dans le cadre de ce mémoire s'intéresse à la réalisation d'un système permet de suivre pas à pas le processus de prétraitement et de classification automatique des textes sur la base des types déjà connus.
Dans notre travail, nous utilisons une approche Machine learning pour repérer les concepts les plus représentatifs de texte.

Mots clés : Texte, Classification automatique, Machine learning.

Abstract

The text is important because it contains a huge amount of information which we can benefit by analyzing its content.
The work carried out within the framework of this thesis is concerned with the realization of a system allowing to follow step by step the process of preprocessing and automatic classification of texts on the basis of already known types.
In our work, we use a Machine learning approach to identify the most representative concepts of text.

Keywords : Text, automatic classification, Machine learning.

الملخص

النص مهم لأنه يحتوي على قدر هائل من المعلومات نستطيع الاستفادة من تحليل محتواها

يتعلق العمل المنفذ في إطار هذه الأطروحة بتحقيق نظام يسمح باتباع عملية المعالجة المسبقة خطوة بخطوة والتصنيف التلقائي للنصوص على أساس أنواع معروفة بالفعل

في عملنا ، نستخدم نهج التعلم الآلي لتحديد المفاهيم الأكثر تمثيلاً للنص

الكلمات الرئيسية : النص، التصنيف التلقائي، التعلم الآلي.

Introduction General

De nos jours, les besoins de catégorisation automatique de documents en raison de La quantité d'information accessible sur Internet, la conception et la mise en oeuvre d'outils efficaces, permettant notamment à l'utilisateur de n'avoir accès qu'à l'information qu'il juge pertinente, devient une nécessité absolue. Comme la plupart de ces outils sont destinés à être utilisés dans un cadre professionnel, les exigences de fiabilité et de convivialité sont très importantes. La grande masse de documents textuels disponibles actuellement ainsi que ses exploitations croissantes, nécessite le développement de méthodes et d'outils pour le traitement automatique du texte. Le but de Notre travail est de développer un modèle fondé sur l'apprentissage automatique pour la catégorisation de textes en utilisant la méthode de naïve bayésienne, qui a donné des résultats satisfaisants.

L'objectif de notre travail est: Comment classer des documents textuels à l'aide de L'apprentissage automatique. Nous allons d'abord calculs de fréquence d'occurrence de termes. Ensuite, Nous appliquons des algorithmes de classification. Enfin, Classer le document selon sa catégorie. La structure proposée du mémoire présentée comme suit :

- Dans **le premier chapitre** nous introduisons des notions générales sur les domaines de : Data Mining, Text Mining en donnant quelques définitions, les taches principales, les applications de chacun.
- Dans **Le deuxième chapitre**, nous détaillons L'apprentissage automatique. (définition, Les applications de l'apprentissage automatique, les avantages et les limite), nous détaillons l'apprentissage automatique pour le texte et comment ça marche, en fin nous présentons les algorithme de classification.
- **Le troisième chapitre :** Dans cette partie du mémoire, nous décrivons la conception de notre système.
- **Et le dernier chapitre** de ce mémoire est réservé aux résultats obtenus lors de l'implémentation du système que nous avons réalisé, ainsi on présente l'environnement (langages et outils) sur lequel le système sera validé et réalisé.

1 Chapitre 1: DATA MINING ET TEXT MINING

1.1 Fouille de données (Data Mining)

1.1.1 Introduction

Aujourd'hui, des milliards de données sont collectées chaque jour dans le monde. En effet, les faibles coûts des machines en termes de stockage et de puissance ont encouragé les sociétés à accumuler toujours plus d'informations. Cependant, bien que la quantité de données à traiter ne cesse d'augmenter les spécialistes dans le domaine estiment que la quantité de données collectées dans le monde double tous les 20 mois. Les entreprises étaient jusqu'alors incapables de transformer leurs données en connaissance directement utilisable.

1.1.2 Définition de fouille de données

La fouille de données ou le Data Mining consiste essentiellement à extraire de l'information d'immenses bases de données de la façon la plus automatique possible. Plus concrètement, le Data Mining est un processus de traitement informatique d'une très grande quantité de données afin de trouver des informations pertinentes, contrairement à la méthode statistique qui nécessite que l'on établisse une hypothèse de départ qu'il s'agira de vérifier. C'est, des données elles-mêmes, que se dégageront les corrélations intéressantes. Le Data Mining se situe à la croisée des statistiques, de l'intelligence artificielle et des bases de données (1).

1.1.3 Les taches de fouille de données

La liste suivante indique les tâches les plus courantes que le data mining est amené à accomplir :

1.1.3.1 La classification

Supposons qu'un décideur veuille classer ses employés par tranches de revenu, ou n'importe quelle autre caractéristique associée à cette personne, comme l'âge, le sexe et la profession, Cette tâche est une tâche de classification (2).

1.1.3.2 L'estimation

L'estimation est similaire à la classification à part que la variable de sortie est numérique plutôt que catégorique, En fonction des autres champs de l'enregistrement l'estimation consiste à compléter une valeur manquante dans un champ particulier. Par exemple on cherche à estimer la

lecture de tension systolique d'un patient dans un hôpital, en se basant sur l'âge du patient, son genre, son indice de masse corporelle et le niveau de sodium dans son sang. La relation entre la tension systolique et les autres données vont fournir un modèle d'estimation, Et par la suite nous pouvons appliquer ce modèle dans d'autres cas (3).

1.1.3.3 La prédiction

La prédiction est semblable à la classification et l'estimation, sauf que pour la prévision, les résultats se situent dans l'avenir.

Exemples de tâches de prévision appliquée au marketing : Prédire le prix d'un stock de trois mois dans le futur (2).

1.1.3.4 Le groupement par similitude

Le groupement par similitude consiste à déterminer quels attributs "vont ensemble", La tâche la plus répandue dans le monde du business, est celle appelée l'analyse d'affinité ou l'analyse du panier du marché, elle permet de rechercher des associations pour mesurer la relation entre deux ou plusieurs attributs (3).

1.1.3.5 La description

Parfois le but de la fouille est simplement de décrire se qui se passe sur une Base de Données compliquée en expliquant les relations existantes dans les données pour premier lieu comprendre le mieux possible les individus, les produit et les processus présents dans cette base. Une bonne description d'un comportement implique souvent une bonne explication de celui-ci. Dans la société Algériennes nous pouvons prendre comme exemple comment une simple description, "les femmes supportent le changement plus que les hommes", peut provoquer beaucoup d'intérêt et promouvoir les études de la part des journalistes, sociologues, économistes et les spécialistes en politiques (3).

1.1.4 Domaines d'utilisation de fouille de données

On peut résumer les champs d'application les plus importants du Data Mining dans les domaines suivants :

1.1.4.1 La gestion de relation client

C'est le domaine principal où le Data Mining a prouvé son efficacité, En effet, dans ce cas le Data Mining permet d'accroître les ventes par une meilleure connaissance de la clientèle. Dans un contexte concurrentiel de plus en plus soutenu, la capacité à conquérir et à retenir les clients repose sur une connaissance fine de leurs besoins et de leur comportement, Les objectifs des analyses en Data Mining sont multiples, tels que la fidélisation, les ventes additionnelles l'efficacité de la force de vente, la personnalisation de l'offre, le contact client, l'enquête de satisfaction des clients, …etc (4).

1.1.4.2 L'aide à la décision (Business intelligence)

C'est l'un des meilleurs facteurs d'augmentation de la productivité. Le Data Mining est incorporé dans cette activité pour mieux analyser les données, rechercher des facteurs expliquant les défauts de la production et leur qualité, et anticiper d'éventuelles réactions.

Les industriels, notamment dans les unités de production, le contrôle et la surveillance, ont toujours fait appel aux méthodes statistiques et de la modélisation (4).

1.1.4.3 La recherche scientifique et médicale

Le Data Mining fournit aux établissements hospitaliers des solutions et des services pour leur permettre de mieux connaître les comportements sanitaires et les pathologies rencontrées à savoir: Le diagnostic médical, l'état des lieux des comportements en matière de santé, l'analyse des risques sanitaires, l'étude de traitements de thérapies, ainsi que les différentes études en milieu hospitalier telle que la génomique, le code génétique, …etc (4).

1.1.4.4 Le domaine financier (Banques et Assurances)

Grâce au Data Mining, un organisme financier peut déterminer le profil exact de ces clients afin de cibler ceux de même profil (mailing). D'autres applications peuvent y avoir telles que la Gestion et calcul du risque client, l'analyse des sinistres, l'assistance au recouvrement en orientant la bonne démarche, la recherche de Fraudes, la recherche des corrélations entre les indicateurs financiers, le retour sur investissement de portefeuilles d'actions (4).

1.1.4.5 Laboratoires pharmaceutiques et cosmétiques

Le Data Mining fournit aux laboratoires pharmaceutiques et de cosmétologie des solutions et des services pour leur permettre à la fois de mieux connaître leur coeur de cible et d'améliorer les

procédés de fabrication, de s'assurer de la qualité de leurs produits et d'évaluer leur potentiel de commercialisation (4).

1.2 Fouille de textes (text mining)

1.2.1 Définition de fouille de textes

La fouille de textes est une technique permettant d'automatiser le traitement de gros volumes de contenus texte pour en extraire les principales tendances et répertorier de manière statistique les différents sujets évoqués. Les techniques de fouille de textes sont surtout utilisées pour des données déjà disponibles au format numérique Sur Internet, le fouille de textes peut être utilisé pour analyser le contenu des e-mails entrants ou les propos tenus sur des forums et médias sociaux (5).

1.2.2 Tâches principales de la fouille de textes

Dans cette section, nous allons énumérer les trois principales tâches auxquelles s'attaque la fouille de textes. Chacune de ces tâches sera un cas particulier du schéma général de la figure ci-dessous

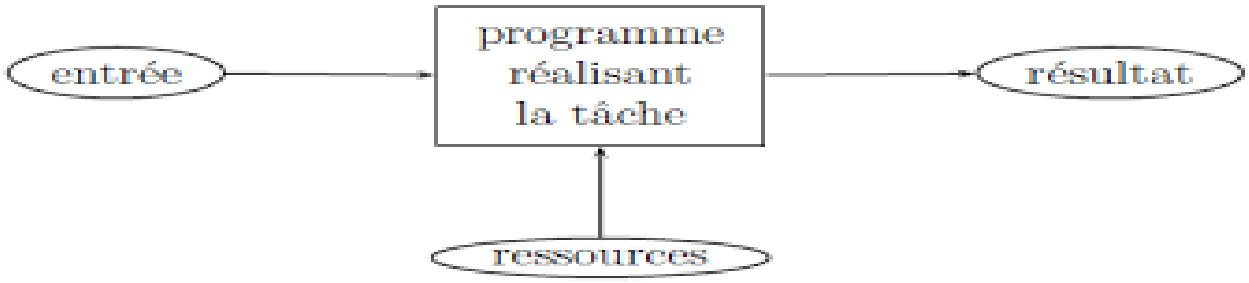


Figure 1.1 Les tâches principales de la fouille de textes

La tâche la plus "naturelle" à envisage, est :

1. **La classification de textes**. Elle consiste à ranger des textes ou des documents dans des "classes" prédéfinies (6).

2. **La recherche d'information**(ou RI) est l'autre "tâche" générale d'ors et déjà omniprésente dans nos usages quotidiens des ordinateurs. Nous la sollicitons chaque fois que nous recherchons des documents répondant à une "requête" (6).

3. **L'extraction d'information** est la dernière tâche fondamentale que nous voulons présenter ici. Comme son nom l'indique, elle se fixe comme objectif d'extraire de textes des informations factuelles précises (6).

1.2.3 Domaines d'utilisation de fouille de textes

Certains des domaines les plus importants de fouille de textes (7):

1.2.3.1 Connaître l'opinion publique

Savoir « qui pense quoi » au sujet d'une question précise est d'un grand intérêt pour les politiciens et les sociologues. Les outils de fouille de textes peuvent augmenter les résultats des élections. Les citoyens s'expriment sur le web, et la plupart de ces informations sont disponibles au public. Malheureusement, ces renseignements sont éparpillés et utiliser un moteur de recherche ne résoudra pas le problème du recouvrement. Après avoir choisi certaines sources de renseignements pertinentes sur le web, les données rassemblées pourraient être analysées pour corréler opinions impliquées, fréquence, et catégories de citoyens (7).

1.2.3.2 La recherche légale

Il n'est pas toujours facile de corréler des rapports de police, des déclarations écrites ou des actes notariés spécifiques à une affaire, avec le droit (législation, réglementation, jurisprudence). Les outils automatisés qui peuvent extraire et résumer les renseignements ont alors beaucoup d'avantages sur un moteur de recherche, car le chercheur ne peut toujours savoir les mots-clé ou la terminologie spécifique touchant les renseignements dont il a besoin (7).

1.2.3.3 Shopping

Le magasinage sur le web veut trouver le bon produit au bon prix. Les prix varient d'un site à un autre et ce n'est pas pratique de visiter et parcourir chaque site manuellement. Un site comparatif pourra se baser sur des analyses automatiques des sites vendeurs, à partir d'une liste de critères (7).

1.2.3.4 L’Analyse du sentiment

Permet de déterminer la "position" des individus étudiés à l'égard d'une marque ou d'un événement (7).

1.3 Conclusion

La fouille de données est une discipline a pour but de valoriser les bases de données. Elle offre des perspectives nouvelles pour la statistique et répond au défi du traitement des giga bases de données. Les données textuelles en format « libre » disponibles sur supports informatiques représentent environ 70% des données disponibles. La préparation des données est une étape importante, si ce n'est primordial, du processus d'extraction de connaissances à partir de données. En schématisant, il s'agit de définir au mieux les individus et la représentation utilisée pour l'apprentissage.

2 Chapitre 2 : L'apprentissage automatique et classification des textes

2.1 Introduction

Au cours des dernières années, la gestion électronique automatique des documents a été un domaine de recherche majeur en informatique. Les documents texte sont devenus le type de référentiel d'informations le plus populaire, en particulier avec la popularité croissante d'Internet et du World Wide Web (WWW). Les documents Internet et Web tels que les pages Web, les e-mails, les messages de groupes de discussion, le fil d'actualités Internet, etc. contiennent des millions ou des milliard de documents texte. Au cours des dernières décennies, les tâches de gestion de documents basées sur le contenu ont pris de l'importance dans le domaine des systèmes d'information, en raison de la disponibilité accrue des documents sous forme numérique (8).

2.2 Intelligence Artificielle

L'intelligence artificielle (IA) est une discipline scientifique étendue qui permet aux systèmes informatiques de résoudre des problèmes en émulant des processus biologiques complexes tels que l'apprentissage, le raisonnement et l'autocorrection. Cet article présente un examen complet de l'application des techniques d'IA pour améliorer les performances des systèmes et réseaux de communication optique. L'utilisation de techniques basées sur l'IA est d'abord étudiée dans des applications liées à la transmission optique, allant de la caractérisation et du fonctionnement des composants du réseau à la surveillance des performances, à l'atténuation des non-linéarités et à l'estimation de la qualité de la transmission. Ensuite, les applications liées au contrôle et à la gestion des réseaux optiques sont également examinées, y compris des sujets tels que la planification et l'exploitation des réseaux optiques dans les réseaux de transport et d'accès. Enfin, le document présente également un résumé des opportunités et des défis dans les réseaux optiques où l'IA devrait jouer un rôle clé dans un avenir proche (9).

2.3 Machine Learning

L'apprentissage automatique est une méthode d'analyse de données qui automatise la construction de modèles analytiques. C'est une branche de l'intelligence artificielle basée sur

l'idée que les systèmes peuvent apprendre des données, identifier des modèles et prendre des décisions avec une intervention humaine minimale (10).

L'apprentissage automatique En quoi consiste l'apprentissage automatique ? De manière générale, un programme informatique tente de résoudre un problème pour lequel nous avons la solution. Par exemple : calculer la moyenne générale des étudiants, classer les étudiants selon leur moyenne. . . Pour certains problèmes, nous ne connaissons pas de solution exacte et donc nous ne pouvons pas écrire de programme informatique. Par exemple : reconnaître automatiquement des chiffres écrits `a la main `a partir d'une image scannée, déterminer automatiquement une typologie des clients d'une banque, jouer automatiquement aux échecs contre un humain ou un autre programme. En revanche, pour ces problèmes il est facile d'avoir une base de données regroupant de nombreuses instances du problème considéré. L'apprentissage automatique consiste alors à programmer des algorithmes permettant d'apprendre automatiquement de données et d'expériences passées, un algorithme cherchant à résoudre au mieux un problème considéré (11).

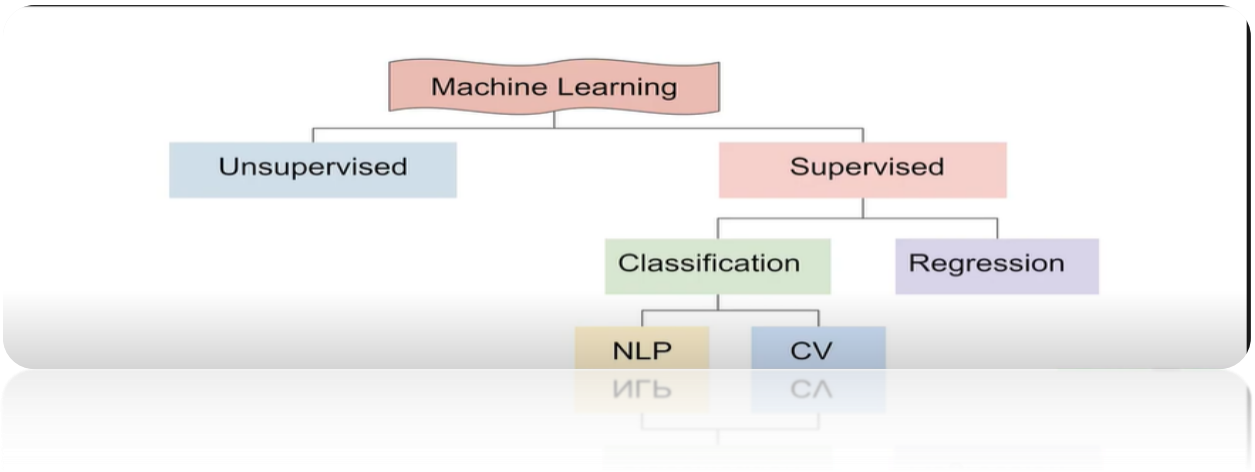


Figure 2.1 Les type de machine learning

2.4 Les type d'apprentissage

2.4.1 L'apprentissage Supervisé

L'apprentissage supervisé (en anglais : Supervised Learning) est le paradigme d'apprentissage le plus populaire en Machine Learning et en Deep Learning. Comme son nom l'indique, cela consiste à superviser l'apprentissage de la machine en lui montrant des exemples

(des données) de la tâche qu'elle doit réalisée. Les applications sont nombreuses : Reconnaissance vocale, vision par ordinateur, régressions, classification, etc. La grande majorité des problèmes de Machine Learning et de Deep Learning utilisent l'apprentissage supervisé. Il est donc essentiel de bien comprendre le fonctionnement de cette mécanique (12).

Figure 2.2 L'apprentissage Supervisé

2.4.2 L'apprentissage Non Supervisé

L'apprentissage non supervisé (Unsupervised Learning) consiste à ne disposer que de données d'entrée (X) et pas de variables de sortie correspondantes. On l'appelle apprentissage non supervisé car, contrairement à l'apprentissage supervisé , il n'y a pas de réponse correcte ni d'enseignant. Les algorithmes sont laissés à leurs propres mécanismes pour découvrir et présenter la structure intéressante des données (13).

L'objectif de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin d'en apprendre davantage sur les données (13).

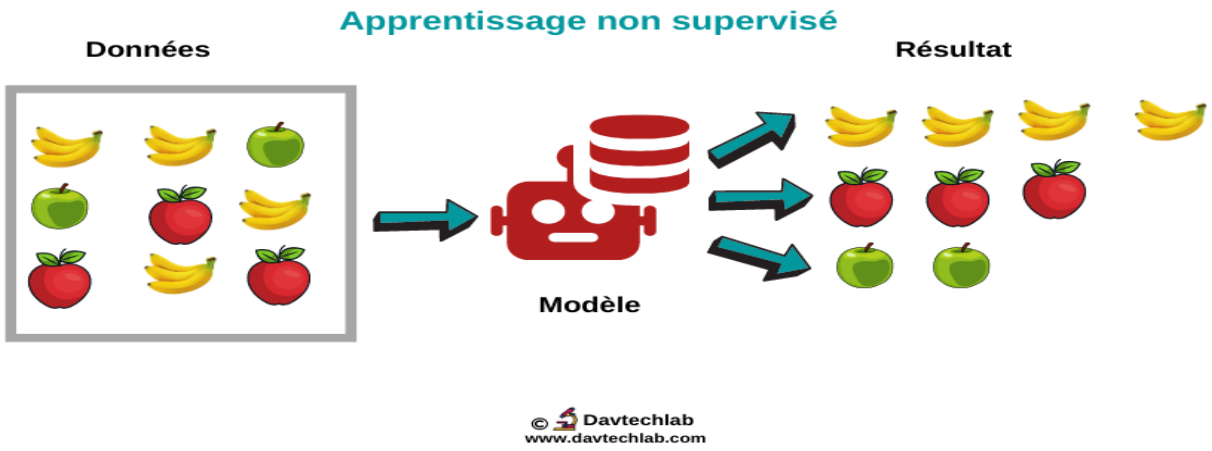


Figure 2.3 L'apprentissage Non Supervisé

2.5 Exemples d'application du Machine Learning

2.5.1 La classification de texte

Est une des tâches importantes et typiques de l'apprentissage automatique supervisé (ML). Il attribue des catégories aux documents, qui peuvent être une page Web, un livre de bibliothèque, des articles de presse, une galerie, etc. filtrage des spams, routage des e-mails, analyse des sentiments, etc (14).

2.5.2 Le e-commerce et l'exemple Amazon

Amazon le site de vente est un exemple intéressant d'application du «Machine Learning» dans l'e-commerce. Supposons par exemple que nous effectuons la recherche d'un produit sur Amazon aujourd'hui. Lorsque nous revenons un autre jour sur le site, il est capable de nous proposer des produits en rapport avec nos besoins spécifiques. Et ceci grâce à des algorithmes de «Machine Learning» qui prévoient l'évolution de nos besoins à partir de nos précédentes visites sur le site (14).

2.5.3 Prédiction des prix

L'algorithme va estimer la valeur de quelque chose (le prix d'une maison, ou les gains espérés d'une boutique …) en fonction des observations précédentes. Par exemple, estimer le prix d'une maison en fonction de sa superficie, sa localisation, possibilité de Parking ou non

etc… Ces estimations sont faites en observant d'autres produits similaires pour en tirer des conclusions (15).

2.5.4 Diagnostique médical

En se basant sur les données médicales d'un patient, l'algorithme peut diagnostiquer si le sujet est atteint d'une maladie donnée. Parfois, ces algorithmes peuvent alerter d'un incident grave de santé avant que cela n'arrive, notamment pour les crises cardiaques (15).

2.6 Les Avantages des Machine learning

2.6.1 Amélioration continue

Les algorithmes d'apprentissage automatique sont capables d'apprendre à partir des données que nous fournissons. Avec la fourniture de nouvelles données, la précision et l'efficacité du modèle s'améliorent avec la formation ultérieure (16).

2.6.2 Automatisation pour tout

Une utilité très puissante du Machine Learning est sa capacité à automatiser diverses tâches de prise de décision. Cela libère beaucoup de temps pour que les développeurs utilisent leur temps pour une utilisation plus productive. Par exemple, une utilisation courante que nous voyons dans notre vie quotidienne est l'analyse des sentiments des médias sociaux et les chatbots. Dès qu'un tweet négatif est émis concernant un produit ou un service d'une entreprise, un chatbot répond instantanément en tant que support client de premier niveau. L'apprentissage automatique change le monde avec son automatisation pour presque tout ce à quoi nous pouvons penser (16).

2.6.3 Identifier les tendances et les modèles

Cette caractéristique est évidente. Quiconque d'entre nous s'intéresse à la technologie d'apprentissage automatique sait bien comment différents algorithmes d'apprentissage supervisé, non supervisé et par renforcement peuvent être utilisés dans de nombreux problèmes de classification et de régression. Nous identifions différentes tendances et modèles avec une énorme quantité de données en utilisant cette technologie. Par exemple, Amazon analyse les habitudes d'achat et les tendances de recherche de ses clients et prédit les produits pour eux à l'aide d'algorithmes d'apprentissage automatique (16).

2.7 Limites de Machine learning

2.7.1 L'acquisition des données

Le point le plus douloureux dans le domaine de la Data Science et du Machine Learning est l'acquisition de données. De plus, la collecte de données a un coût. De plus, il se trouve que lorsque nous collectons des données à partir d'enquêtes, elles peuvent contenir un grand volume de données fausses et incorrectes. Souvent, nous sommes confrontés à une situation où nous trouvons un déséquilibre dans les données qui conduit à une mauvaise précision des modèles. Ces raisons font de l'acquisition de données un inconvénient majeur (16).

2.7.2 Choisissez des algorithmes

Nous pouvons implémenter différents algorithmes pour trouver une solution. C'est une tâche difficile d'exécuter des modèles avec différents algorithmes et de sélectionner l'algorithme le plus précis en fonction des résultats (16).

2.7.3 Consommation de temps

Les modèles d'apprentissage automatique sont capables de traiter d'énormes quantités de données. Chaque fois que la taille des données augmente, le temps d'apprentissage et de traitement des données augmente également. Parfois, cela peut également nécessiter d'autres ressources (16).

2.8 Machine Learning Pour Le Texte

2.8.1 Classification

Action de ranger par classes, par catégories des choses présentant des critères en commun. Et nous devons d’abord nous rappeler, quelle est la classification. C'est l'utilisation d'algorithmes d'apprentissage automatique pour déterminer à quelle catégorie appartient l'échantillon

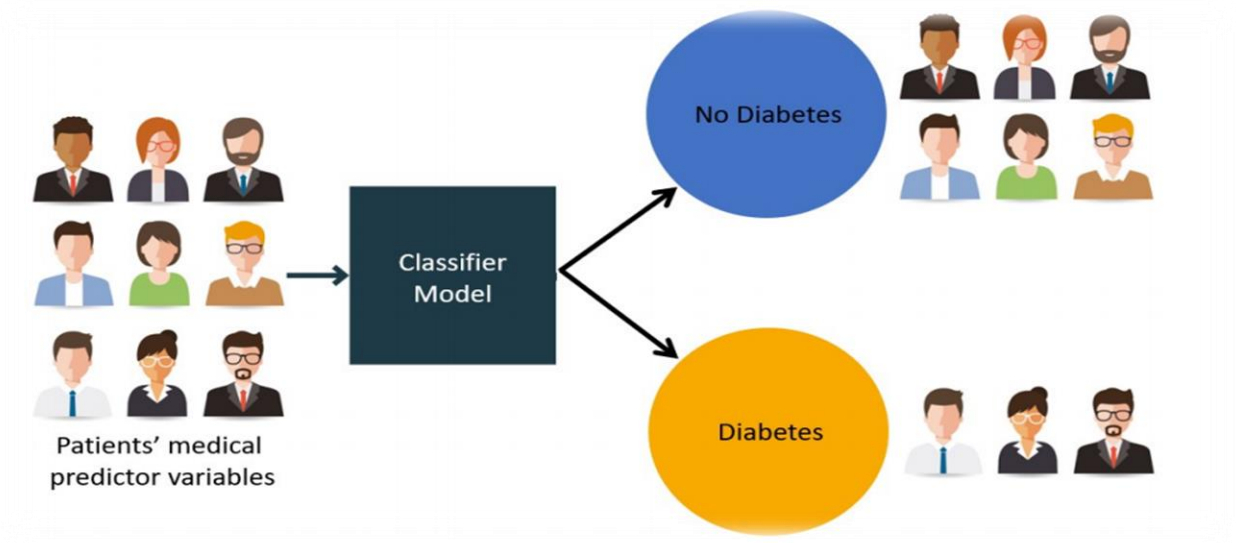


Figure 2.4 Classification

Les algorithmes ne sont pas toujours très efficaces

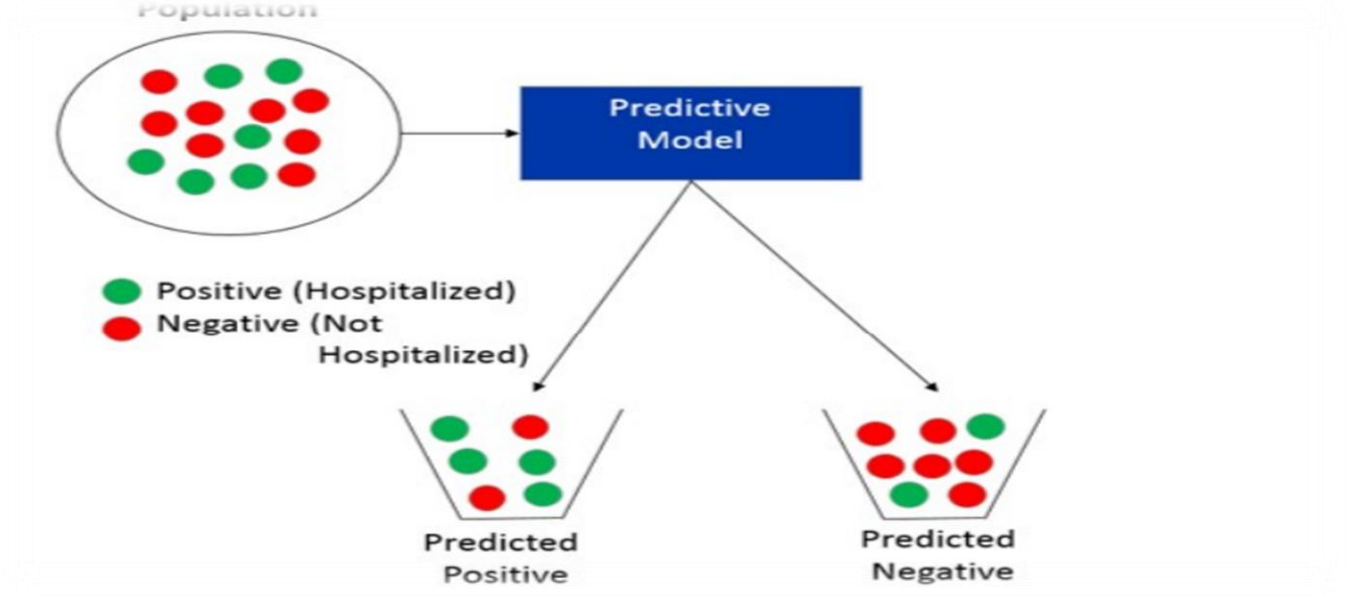


Figure 2.5 Exemple Efficacité d'algorithme

Il a deux types, la classification binaire et la classification multiple

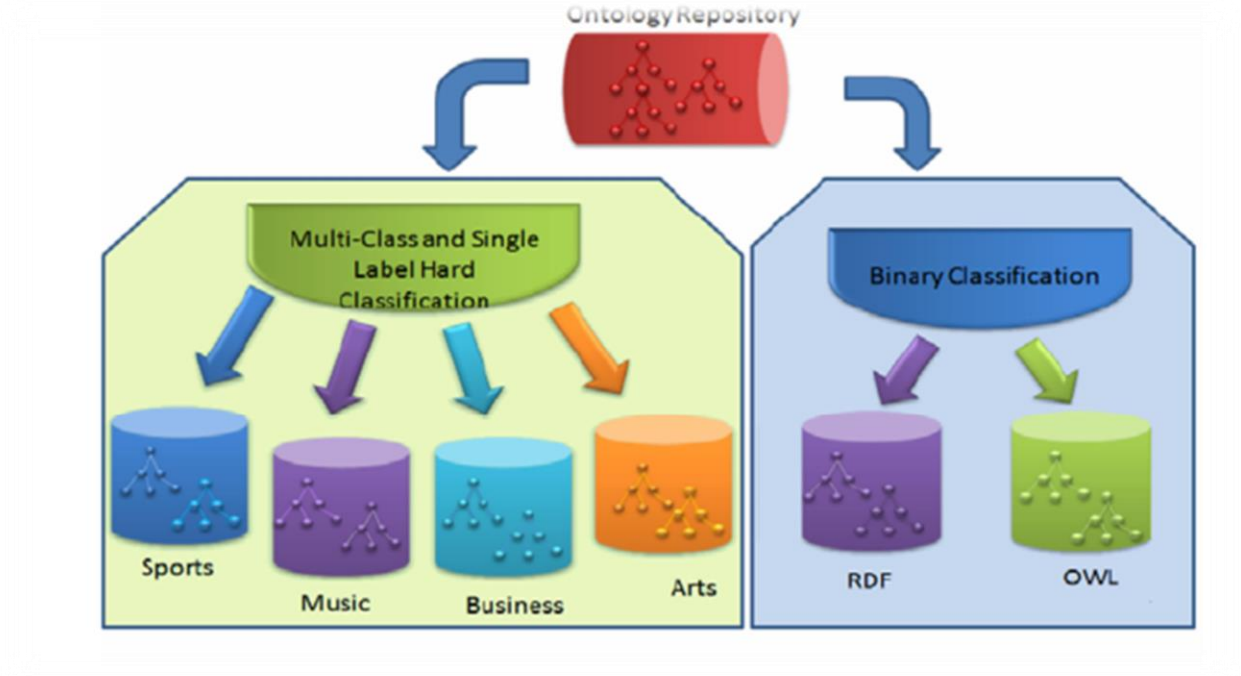


Figure 2.6 Les type de classification

Il est utilisé pour classer les images

Figure 2.7 classification des images

Maintenant, quelle est la classification des textes.

2.8.2 Classification des textes

C'est une application plus importante de l'apprentissage automatique et elle est en plus demandée. Ca veut dire utiliser des techniques pour classer automatiquement le texte dans plus d'une catégorie, en fonction de son contenu textuel. Parmi les exemples les plus connus : la classification des e-mails

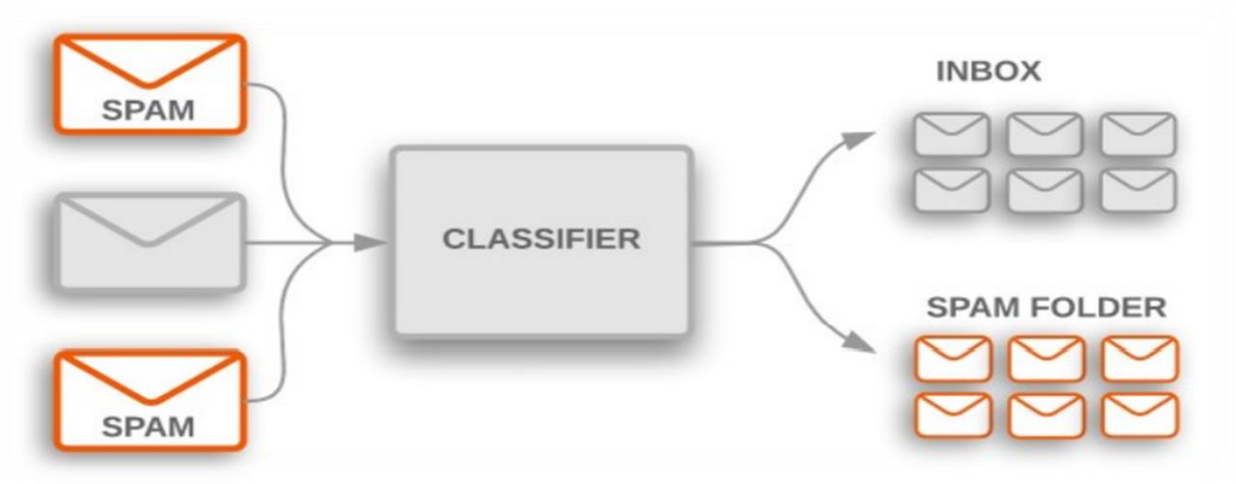


Figure 2.8 Classification des e-mail

Aussi, l'analyse sentimentale dans les textes si elle est négative ou positive

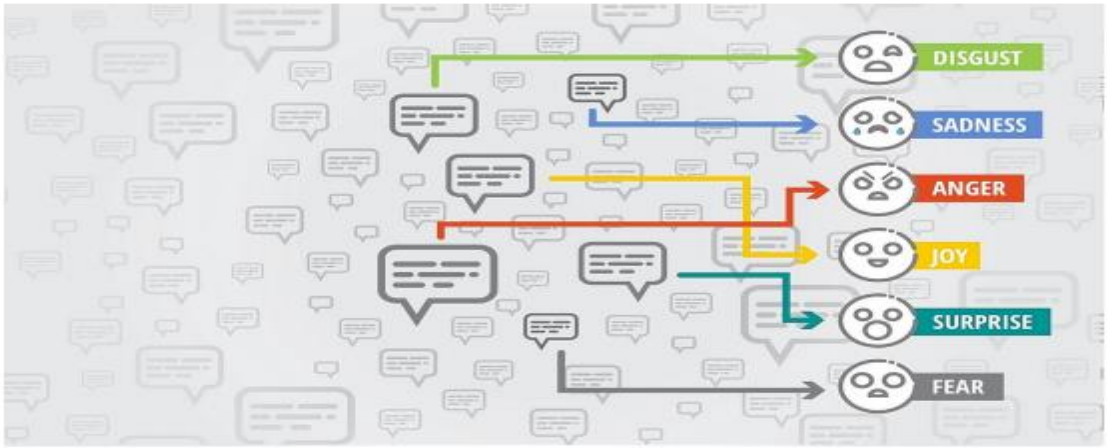


Figure 2.9 L'analyse sentimentale

Aussi, classification des textes et à quelle catégorie il appartient

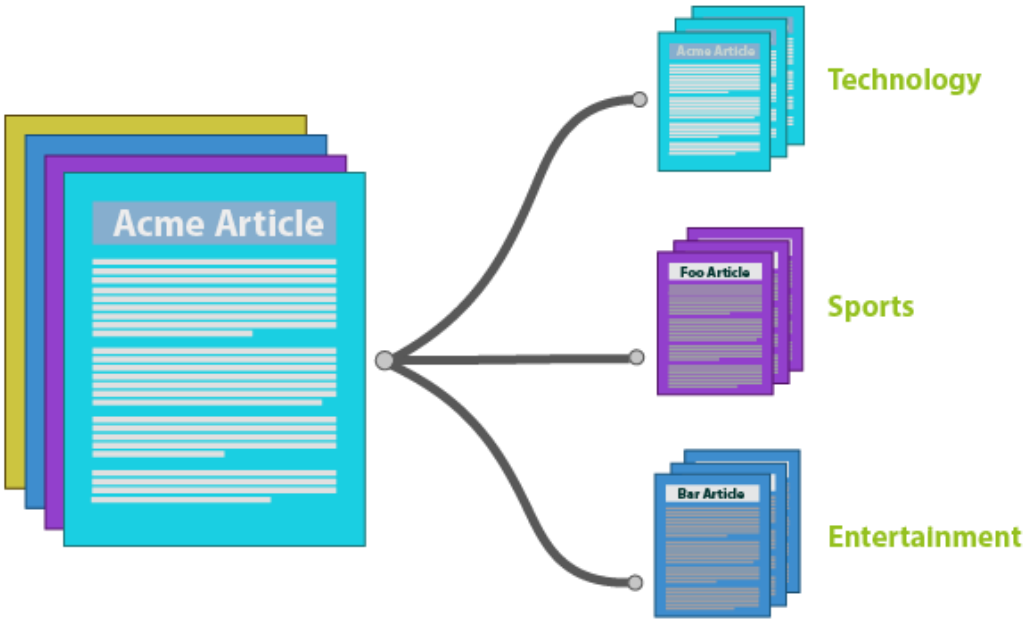


Figure 2.10 Classification des documents

2.9 Algorithmes d'apprentissage

Le choix de l'algorithme d'apprentissage dépend de l'objectif final à atteindre et de la taille du corpus qui joue un rôle très important dans ce choix. Nous avons choisi l'algorithme Naïve bayes et Régression logistique pour construire notre modèle de prédiction qui nous permet d'associer des documents à une catégorie.

2.9.1 Naïve bayésiene

L'une des applications les plus utiles de la règle de Bayes est ce que l'on appelle le classifieur bayésien naïf. La classification naïve bayésiene est un type de classification Bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses, appartenant à la famille des classifieurs Linéaires (17).

Le classifieur bayésien est une technique d'apprentissage automatique qui peut être utilisée pour séparer des objets tels que des documents textuels en deux classes ou plus. Le classifieur est formé en analysant un ensemble de données d'entraînement, pour lesquelles les classes correctes sont indiquées (18).

2.9.1.1 Description du modèle Bayésienne

Le modèle probabiliste pour un classifieur est le modèle conditionne ($\boldsymbol{C}|\boldsymbol{F1}$,……..,) où **C** est une variable de classe dépendante dont les instances ou classes sont peu nombreuses, conditionnée par plusieurs variables caractéristiques $\boldsymbol{F1}$,……..,$\boldsymbol{Fn}$. Lorsque le nombre de caractéristiques $\boldsymbol{n}$ est grand, ou lorsque ces caractéristiques peuvent prendre un grand nombre de valeurs, baser ce modèle sur des tableaux de probabilités devient impossible. Par conséquent, nous le dérivons pour qu'il soit plus facilement soluble. À l'aide du théorème de Bayes, nous écrivons (17):

$$\boldsymbol{p}(\boldsymbol{C}|\boldsymbol{F1},\ldots\ldots..,\boldsymbol{Fn}) = \frac{\mathrm{p(c)p(F1,\ldots\ldots..,Fn|c)}}{\mathrm{p(F1,\ldots\ldots..,Fn)}}$$

En pratique, seul le numérateur nous intéresse, puisque le dénominateur ne dépend pas de **C** et les valeurs des caractéristiques $\boldsymbol{Fi}$ sont données. Le dénominateur est donc en réalité constant. Le numérateur est soumis à la loi de probabilité à plusieurs variables (17).

$$\boldsymbol{p}(\boldsymbol{C},\boldsymbol{F1},\ldots\ldots..,\boldsymbol{Fn})$$

et peut être factorisé de la façon suivante, en utilisant plusieurs fois la définition de la probabilité conditionnelle (17) :

$$\boldsymbol{p}(\boldsymbol{C},\boldsymbol{F1},\ldots\ldots..,\boldsymbol{Fn})$$

$$= \boldsymbol{p}(\boldsymbol{C})\,\boldsymbol{p}(\boldsymbol{F1},\ldots\ldots..,\boldsymbol{Fn}|\mathrm{C})$$

$$= \boldsymbol{p}(\boldsymbol{C})\,\mathrm{p}(\boldsymbol{F_1}|\boldsymbol{C})\,\mathrm{p}(\boldsymbol{F_{2,\ldots\ldots\ldots,}}\boldsymbol{Fn}|\mathrm{C},\boldsymbol{F_1})$$

$$= \boldsymbol{p}(\boldsymbol{C})\,\mathrm{p}(\boldsymbol{F_1}|\boldsymbol{C})\,\mathrm{p}(\boldsymbol{F_2}|\boldsymbol{C},\boldsymbol{F_1})\,\boldsymbol{p}(\boldsymbol{F_2},\ldots\ldots..,\boldsymbol{Fn}|\mathrm{C},\boldsymbol{F_1},\boldsymbol{F_2})$$

$$= \boldsymbol{p}(\boldsymbol{C})\,\mathrm{p}(\boldsymbol{F_1}|\boldsymbol{C})\,\mathrm{p}(\boldsymbol{F_2}|\boldsymbol{C},\boldsymbol{F_1})\,\mathrm{p}(\boldsymbol{F_3}|\boldsymbol{C},\boldsymbol{F_1},\boldsymbol{F_2})\,\boldsymbol{p}(\boldsymbol{F_4},\ldots\ldots..,\boldsymbol{Fn}|\mathrm{C},\boldsymbol{F_1},\boldsymbol{F_2})$$

$= p(C)\ \mathrm{p}(F_1|C)\ \mathrm{p}(F_2|C, F_1)\ \mathrm{p}(F_3|C, F_1, F_2) \ldots\ldots\ \mathrm{p}(F_n|\mathrm{C}, F_1, F_2, F_3\ \ldots$

C'est là que nous faisons intervenir l'hypothèse naïve : si chaque Fi est indépendant des autres caractéristiques $yi \neq j$ R alors Pour tout $i \neq j$R, par conséquent la probabilité conditionnelle peut s'écrire (17) :

$$p(F_i|C, F_j) = p(F_i|C)$$

$$p(C, F_1, \ldots\ldots\ldots, F\mathfrak{y}) = p(C)\ p\ (F_1|C)\ \ p\ (F_2|C)\ \ p\ (F_3|C)\ \ \ldots$$

$$= p(C) \prod_{i=1}^{n} p(\ F_i|C)$$

Par conséquent, en tenant compte de l'hypothèse indépendance ci-dessus, la probabilité conditionnelle de la variable de classe C peut être exprimée par où

$$p(C, F1, \ldots\ldots\ldots, F\mathfrak{y}) = \frac{1}{Z}\ p\ (\mathrm{C}) \prod_{i=1}^{n} p(\ F_i|C)$$

où $\mathbf{Z}$ (appelé « évidence ») est un facteur d'échelle qui dépend uniquement de $F1, \ldots\ldots\ldots,$, à savoir une constante dans la mesure où les valeurs des variables caractéristiques sont connues.

Les modèles probabilistes ainsi décrits sont plus faciles à manipuler, puisqu'ils peuvent être factorisés par l'antérieure $\mathbf{P(C)}$ (probabilité a priori de $\mathbf{C}$) et les lois de probabilité indépendantes $\mathbf{P(F_i|C)}$. S'il existe **K** classes pour $\mathbf{C}$ et si le modèle pour chaque fonction peut être exprimé selon paramètres, alors le modèle bayésien naïf correspondant dépend de (k − 1) + n r k paramètres (17).

Dans la pratique, on observe souvent des modèles où $\mathbf{K=2}$ (classification binaire) et $\mathbf{r=1}$ (les caractéristiques sont alors des variables de Bernoulli). Dans ce cas, le nombre total de paramètres du modèle bayésien naïf ainsi décrit est de $2\mathbf{n}+1$, avec $\mathbf{n}$ le nombre de caractéristiques binaires utilisées pour la classification (17).

2.9.1.2 Les Types de Naïve Bayes Classificateur

Multinomial Naïve Bayes

Est principalement utilisé pour le problème de classification des documents, c'est-à-dire si un document appartient à la catégorie des sports, de la politique, de la technologie... etc. Les

features/prédicteurs utilisés par le classifieur sont la fréquence des mots présents dans le document (19).

Bernoulli Naïve Bayes

Est similaire aux bayes naïfs multinomiaux mais les caractéristique ne sont que sous forme binaire (19).

Gaussian Naïve Bayes

Lorsque les caractéristique prennent une valeur continue et ne sont pas discrets, on suppose que ces valeurs sont échantillonnées à partir d'une distribution gaussienne (19).

2.9.1.3 Les Avantages de Naïve Bayes Classificateur

Ce type de classification « simple » permet à l'algorithme d'apprendre rapidement. Il n'est pas nécessaire en effet de fournir un gros volume de données lors de la phase d'apprentissage. Son exécution est de plus très rapide, comparativement à d'autres méthodes autrement plus complexes mais lourdes à mettre en œuvre. Cette méthode offre ainsi des résultats très efficaces dans des domaines d'utilisation variés. C'est aujourd'hui un algorithme largement plébiscité pour les outils de Machine learning (capacité donnée aux ordinateurs d'apprendre par eux-mêmes) du fait de ses calculs de probabilités peu coûteux qui lui confèrent une grande agilité (20).

2.9.1.4 Les Inconvénients de Naïve Bayes Classificateur

Naive Bayes suppose que toutes les fonctionnalités sont indépendantes, C'est rare dans la vie réelle. ce qui limite l'application de cet algorithme dans des cas d'utilisation réels (21).

2.9.2 Régression logistique

2.9.2.1 Définition

Régression logistique est un modèle de classification linéaire qui est le pendant de la régression linéaire , quand $\boldsymbol{Y}$ ne doit prendre que deux valeurs possibles (0 ou 1). Comme le modèle est linéaire, la fonction hypothèse pourra s'écrire comme suit (22):

$$\boldsymbol{S(X^{(i)}) = \vartheta_0 + \vartheta_1 x_1 + \vartheta_2 x_2 + \cdots + + \vartheta_n x_n}$$

Avec :

1. $\mathbf{x^{(i)}}$: une observation (que ce soit du Training Set ou du Test Set), cette variable est un vecteur contenant $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$
2. $\mathbf{x^{(i)}}$: est une variable prédictive (feature) qui servira dans le calcul du modèle prédictif.
3. $\boldsymbol{\theta_i}$: est un poids/paramètre de la fonction hypothèse. Ce sont ces $\boldsymbol{\theta_i}$ qu'on cherche à calculer pour obtenir notre fonction de prédiction.
1. θ_0:est une constante nommée le bias .

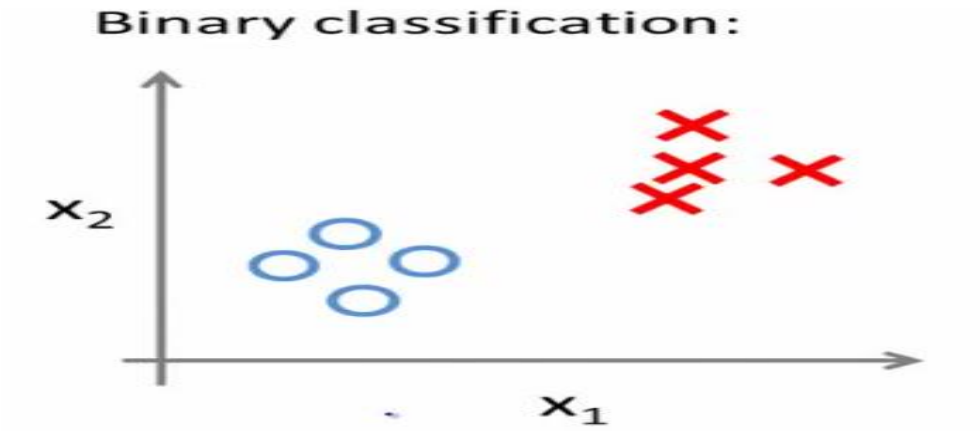


Figure 2.11 Classification binair

2.9.2.2 La fonction Logistique pour calculer la probabilité d'une classe

La fonction *score* qu'on a obtenue intègre les différentes variables prédictives (les x_i). A cette fonction, on appliquera **la fonction Logistique** Cette fonction produit des valeurs comprises entre 0 et 1 (22).

Le résultat obtenu par la fonction Logistique est interprété comme la **probabilité que l'observation X soit d'un label (étiquette) 1** (22)**.**

La fonction Logistique (autre nom pour la fonction Sigmoid), est définie comme suit :

$$Sigmoid = \frac{1}{1+e^{-x}}$$

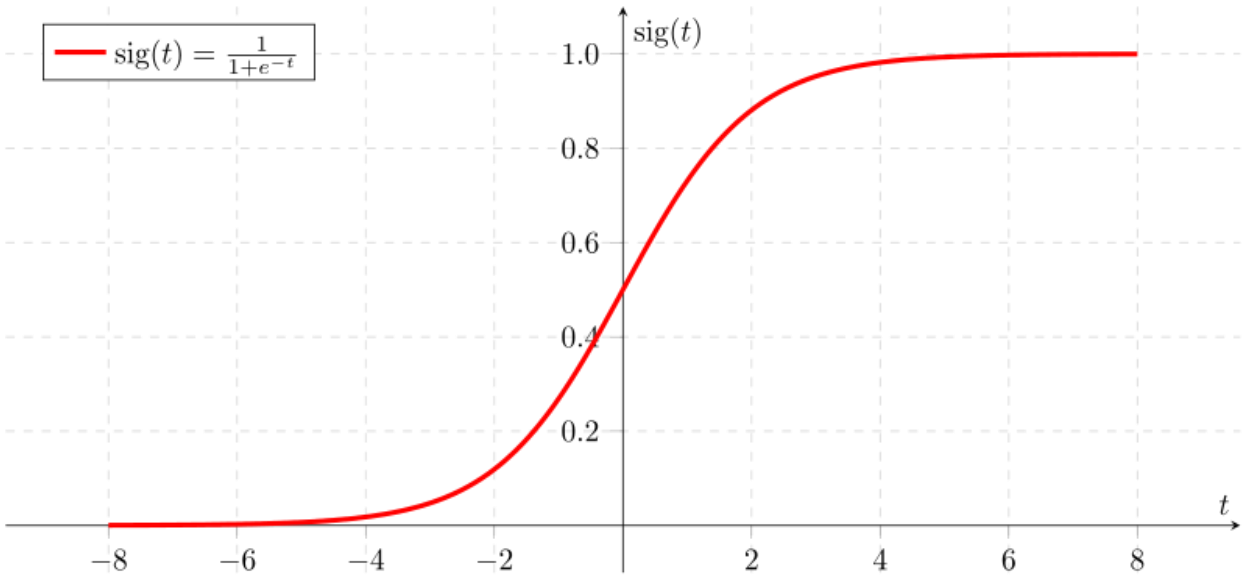


Figure 2.12 La fonction logistique

En analysant la courbe ci-dessus, On peut remarquer trois choses :

- Elle passe par l'ordonnée 0.5 quand x=0 : *Sigmoid =0.5*.
- La fonction Sigmoid asymptote à 0 et 1 (elle s'approche des ordonnées 0 et 1 mais sans les "toucher")
- On remarque que *Sigmoid(x)>0.5* quand x > 0 et Sigmoid (x) < 0.5 quand x < 0

Quand notre problème a plusieurs étiquettes possibles (par exemple classifier un article dans une catégorie (sport, politique, Tech...), on parle de Multi-class classification (Classification Multi classes). Dans ce cas, $\boldsymbol{Y} \in \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \ldots\}$. Encore une fois, on peut attribuer arbitrairement les numéros des classes aux observations du Training Set (22).

2.9.2.3 La régression logistique pour classification multi-classes

Par défaut, la régression logistique ne peut pas être utilisée pour les tâches de classification qui ont plus de deux étiquettes de classe, ce que l'on appelle la classification multi-classe.

Il nécessite une modification pour prendre en charge les problèmes de classification multi-classes. Une approche courante pour adapter la régression logistique aux problèmes de classification multi-classe consiste à diviser le problème de classification multi-classe en

plusieurs problèmes de classification binaire et à adapter un modèle de régression logistique standard à chaque sous problème (22).

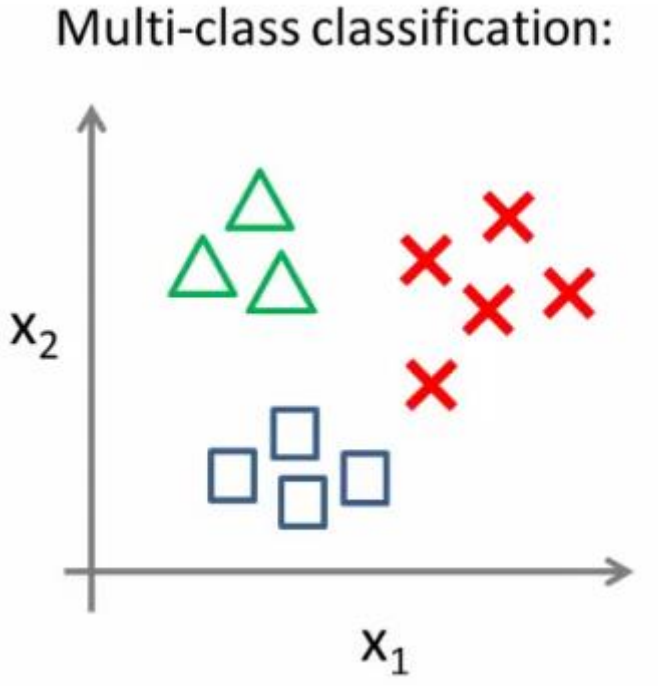


Figure 2.13 Classification multi-classe

2.10 Conclusion

La catégorisation de textes a progressé grâce à l'introduction des techniques héritées de l'apprentissage automatique qui ont amélioré très significativement les taux de bonne classification. Il reste néanmoins difficile de fournir des valeurs chiffrées sur les performances qu'un système de classification peut actuellement atteindre.

3 Chapitre 3: Conceptions du système

3.1 Introduction

Dans ce chapitre nous aborderons une description générale de notre système, en mettant en évidence son coté conceptuel du prétraitement qui constitue une étape fondamentale avant la mise en oeuvre de notre système.

3.2 Architecture générale

Notre système se base sur l'utilisation du Machine Learning pour classer les documents textuels. Le système prend en entrée une base composé de cinq catégories (Sport, Technologie, Politic, Entertainment, Business) et la transforme en une base de caractéristiques utilisable par la phase d'apprentissage. La base est subdivisée en deux partie une pour l'entraînement et l'autre pour le test. Le module d'entraînement utilise la base d'entraînement et un algorithme d'apprentissage pour fournir un modèle de décision qui est appliqué sur la base de test, Si le modèle est accepté il sera conservé et utilisé par le module d'utilisation et l'entraînement se termine.

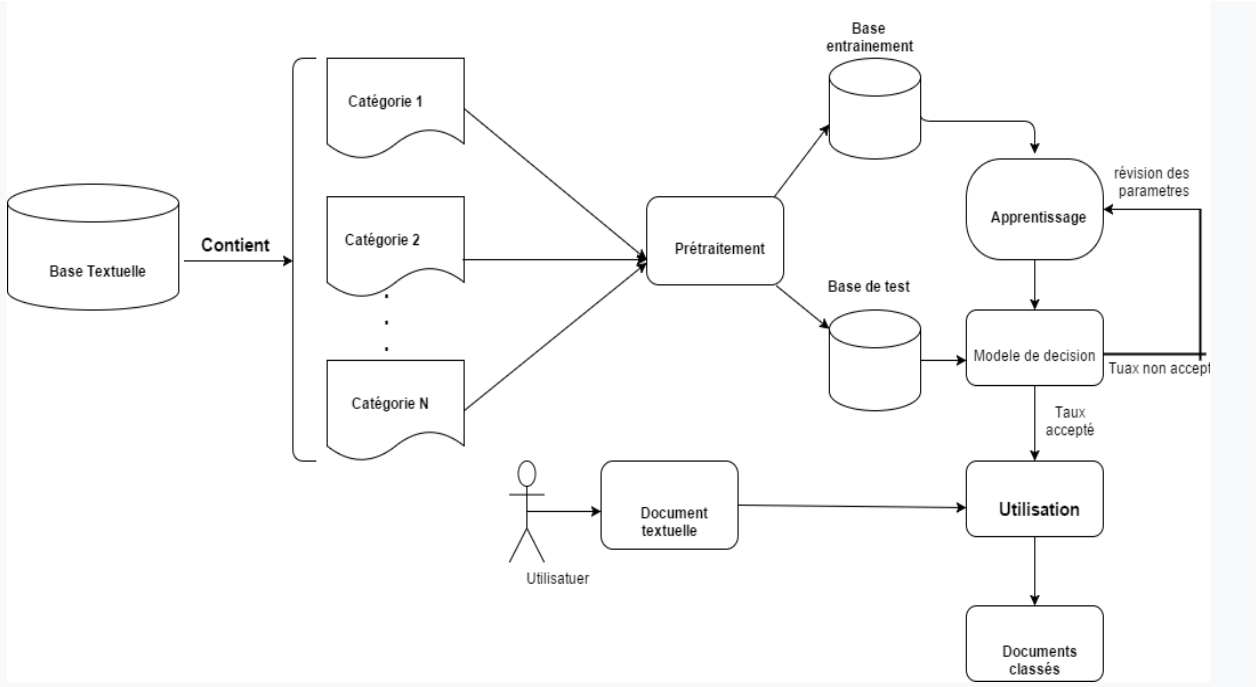


Figure 3.1 Architecture générale

Les deux acteurs de notre système sont les suivants :

Administrateur: L'administrateur de notre système assure les fonctionnalités suivantes :

1. Sélection de la base de données.
2. Divisez la base en une base d'entraînement et une base de test
3. Sélection de l'algorithme d'apprentissage.

Utilisateur: les tâches qu'un utilisateur de système peut réaliser sont :

1. Utilisé le system sur des nouvelles Textes non classé.
2. Evaluation de résultats.

3.3 Architecture détaillée

Dans ce qui suit, nous détaillons chacune des phases de notre système.

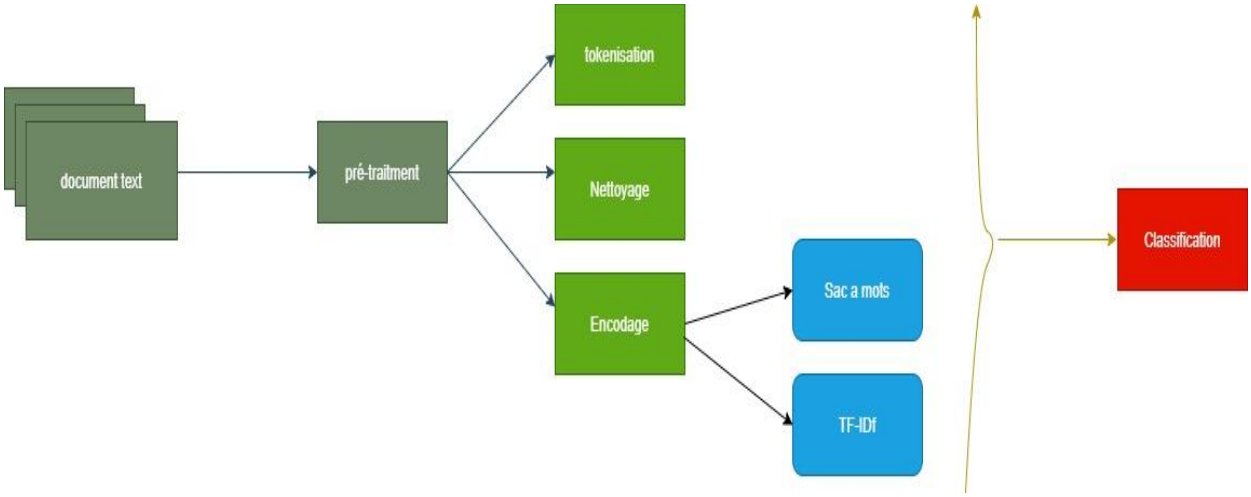


Figure 3.2 Architecture détaillée

3.3.1 Pré-traitement

Avant de pouvoir créer et entrainer un quelconque classificateur à partir d'un corpus de documents donné, nous devions impérativement transformer les documents textes en entrées valides compréhensibles par les différents algorithmes de classification. Ces entrées valides sont en fait des vecteurs ou des matrices qui définissent le poids de chaque descripteur (mot ou groupe de mots) dans chaque document texte où ils apparaissent.

Les traitements spécifiques, propre à notre cas, seront détaillés dans la deuxième partie.

3.3.1.1 Tokenization

Un token est une unité définie comme une séquence de caractères comprise entre deux séparateurs, les séparateurs étant les blancs, les signes de ponctuation et certains autres caractères comme les guillemets ou les parenthèses.

La tokenisation consiste à segmenter un document texte en tokens de mots, séquences de mots ou carrément de phrases, mais généralement elle s'opère souvent sur des mots.

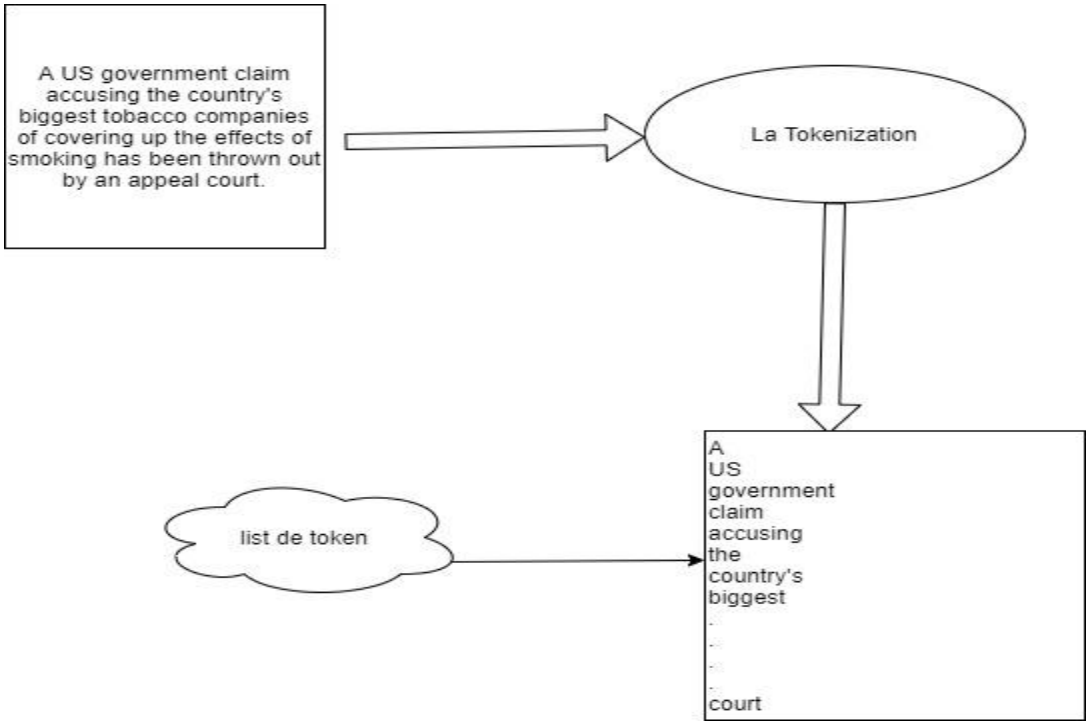


Figure 3.3 La Tokenization

3.3.1.2 Nettoyage (éliminer les mots vides)

Une fois les documents textes découpés en tokens, nous apercevons que certains de ces tokens sont présents dans tous les textes du corpus, c'est ce que nous appelons les mots vides : les articles, les prépositions, les déterminants, les adverbes... comme "la, le, dans, car," dans la langue française, et "the, and, after" dans la langue anglaise. Ils représentent 30% des mots dans un texte. La présence de ces mots n'apporte absolument aucune différence tant sur le plan sémantique que sur le plan lexical. Cela veut dire que leur présence dans tous les textes du corpus les rend non discriminants et du coup leurs utilisation pour une tâche de classification s'avère inutile. Par contre, leur suppression réduit la dimension de notre document vecteur, par conséquent, le temps de traitement et le temps d'apprentissage seront réduit considérablement.

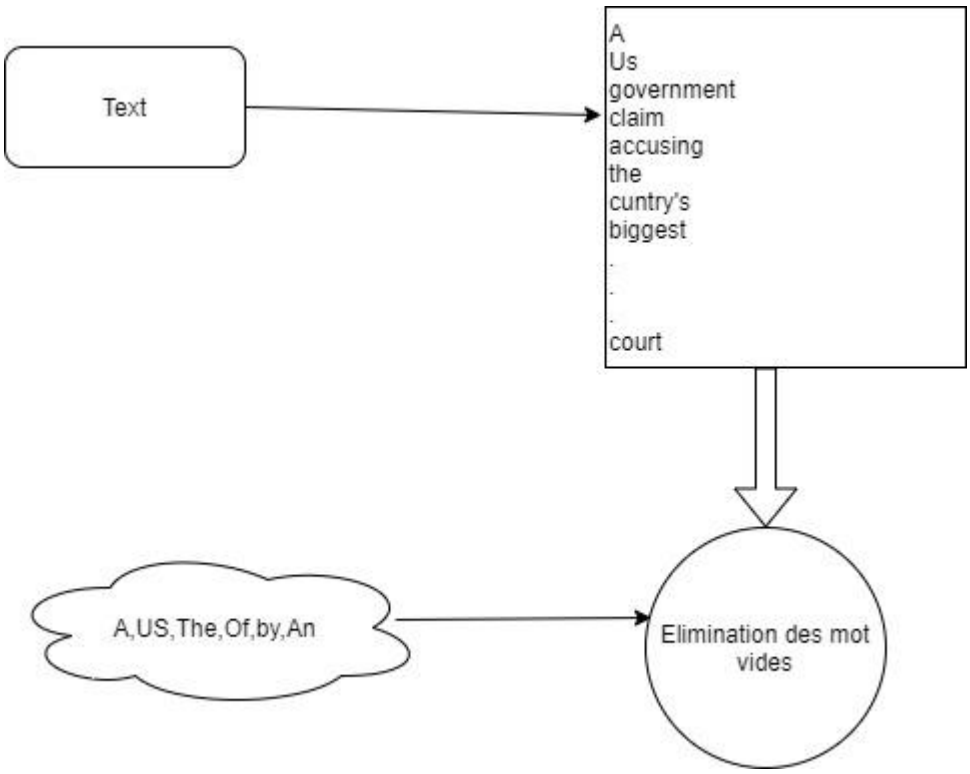


Figure 3.4 Nettoyage

3.3.1.3 Encodage

Transformer l'ensemble des mots en un vecteur numérique en passant par deux étapes:

La technique de sac à mots, puis l'application de la méthode TF-IDF sur le résultat.

Sac à mots : dans ce modèle, le texte est représenté sous forme de vecteur contenant ses mots, sans tenir compte de leur ordre, mais en gardant la multiplicité. Cette technique est principalement utilisée pour calculer différentes mesures qui caractérisent le texte, Mais le problème ici est que, un mot qui se répète dans tout le corpus ne veut pas dire qu'il est vraiment important ou il caractérise un document précis. Pour résoudre ce problème, la fréquence du terme est pondérée par l'importance du document dans le corpus. Dans notre système nous avons utilisé la méthode TF-IDF.

TF-IDF : (Term Frequency-Inverse Document Frequency) est une mesure statistique permet d'évaluer l'importance d'un terme contenu au sein d'un document, dans un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document.

TF(t) = Nombre d'apparition du terme t dans le document (n) / Nombre total de termes dans le document (k).

$$Tf = \frac{n}{k}$$

IDF = Nombre totale des documents (D)/ Nombre des documents citant ce terme (D_t).

$$IDF = \log \frac{D}{D_t}$$

TF*IDF (Term Frequency Inverse Document Frequency):

Le poids d'un terme t dans un document D

3.3.2 Apprentissage

il regroupe deux modules, l'entraînement et la validation utilisant chacun une partie de la base des caractéristiques subdivisée en deux parties, base d'entraînement et base de test. Le module d'entraînement utilise la base d'entraînement pour fournir un modèle de décision tandis que le module de validation utilise la base de test pour mesurer la performance du modèle fourni.

3.3.2.1 Entraînement

pour entraîner notre modèle, nous avons choisis l'algorithme que nous avons déjà présenté dans le chapitre précédent. Le résultat de l'entraînement est un modèle ou pattern, qui représente l'analyse des données et leur transformation en informations utiles, en établissant des relations entre elles.

3.3.2.2 Validation

consiste à mesurer la capacité du modèle a reconnaître des nouveaux exemples. Pour cela, on écarte dès le départ une partie des exemples pour les utiliser pour le test du modèle.

3.3.2.3 Les score

Les scores sont aussi importants au même titre que les autres phases du processus de construction d'un classificateur. Sinon, comment savoir si un tel classificateur est fiable, et comment savoir s'il se comporte bien avec de nouvelles données? Plusieurs mesures ou scores existent pour justement vérifier et estimer le degré de généralisation d'un classificateur

La matrice de confusion :

		Classes Prédites	
		Classe01	Classe02
Classes Réelles	Classe01	VP	FN
	Classe02	FP	VN

La matrice de confusion en text mining sert à vérifier le bon classement des documents préalablement étiquetés. En d'autres termes, elle indique si un classificateur fonctionne bien et à quel degré de fiabilité.

Chaque classe du classificateur est représentée par une colonne et une ligne. La ligne indique le nombre de documents réels appartenant à la classe (C) et la colonne indique à quel nombre de documents cette classe (C) est assignée.

VP (Vrais positifs) : Les documents appartenant à la classe 01que le classificateur a classés à la classe 01.

FP (Faux positifs) : Les documents appartenant à la classe 02 que le classificateur a classés à la classe 01.

VN (Vrais négatifs) : Les documents appartenant à la classe 02 que le classificateur a classés à la classe 02.

FN (Faux négatifs): Les documents appartenant à la classe 01 que le classificateur a classés à la classe 02.

Notons bien que c'est avec ces quatre paramètres que toutes les autres mesures sont calculées.

La précision :

C'est le rapport entre le nombre de documents correctement classés dans la classe (C) sur le nombre de document auxquels la classe (C) est assignée.

$$\text{Précision(C)} = \frac{VP}{VP+FP}$$

Le rappel :

C'est le rapport entre le nombre de documents correctement classés dans la classe (C) sur le nombre de documents appartenant à la classe (C).

$$\text{Rappel (C)} = \frac{VP}{VP+FN}$$

La F-Mesure :

La F-Mesure est un indicateur qui combine le rappel et la précision elle est donnée par la formule suivante :

$$\text{F-Mesure} = \frac{2*(\text{Précision}*\text{Rappel})}{\text{Précision}+\text{Rappel}}$$

3.3.3 Utilisation

C'est la dernière phase et la plus importante dans notre système. Après être arrivé au meilleur taux de reconnaissance, ou après avoir construit le meilleur modèle dans la phase précédente, nous devons l'utiliser sur des nouvelles informations non étiquetées, et le modèle nous permet de prédire la classe de la nouvelle document.

3.4 Conclusion

Ce chapitre a décrit la conception de notre système et il a présenté la démarche suivie dans ses différentes phases. Dans le chapitre suivant nous allons présenter l'implémentation de tous les composants et les modules de notre système.

4 Chapitre 4: Implémentation

4.1 Introduction

Ce chapitre a pour objectif de présenter l'aspect implémentation de notre application, il s'agit donc d'expliquer l'environnement matériel sur lequel notre système a été développé, les langages de programmation et les outils utilisés. Par la suite, nous allons présenter les interfaces graphiques en décrivant les différentes fonctionnalités de notre application et nous présenterons un exemple qui nous permettra d'illustrer les résultats obtenus lors de l'utilisation de notre approche.

4.2 Outils utilisés

4.2.1 Langage utilisé

Python

Python est un langage de programmation, interprété car, avant de pouvoir les exécuter, un logiciel spécialisé se charge de transformer le code du programme en langage machine, multi-paradigme et multiplateformes, est placé sous une licence libre. qui vous permet de travailler rapidement et d'intégrer les systèmes plus efficacement. Python 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. Plusieurs bibliothèques peuvent ainsi être installées pour, par exemple, développer des interfaces graphiques en Python.

Figure 4.1 PYTHON

4.2.2 Bibliothèques utilisées

Sklearn

Scikit-learn est une bibliothèque d'apprentissage automatique gratuite pour Python. Il Comporte divers algorithmes tels que multinomial naive bayes et logistique régression.

```
    if combo2.get() == "MultiNomial":
        from sklearn.naive_bayes import MultinomialNB
        model = MultinomialNB()
        history = model.fit(X_train, y_train)
        msg.showinfo('Info', 'Model Created')
        y_pred = model.predict(X_test)
        acc.config(text="Accuracy Score = " + str(accuracy_score(y_test, y_pred))[:5])
        acc1.config(text="F1 Score              = " + str(f1_score(y_test, y_pred, average='macro'))[:5])
        acc2.config(text="Recall Score      = " + str(recall_score(y_test, y_pred, average='macro'))[:5])
        y_predicting = y_pred
    if combo2.get() == "LogisticRegression":
        from sklearn.linear_model import LogisticRegression
        model = LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                                   penalty='l2', random_state=0, solver='liblinear', tol=0.0001,
                                   verbose=0, warm_start=False)
        history = model.fit(X_train, y_train)
        msg.showinfo('Info', 'Model Created')
        y_pred = model.predict(X_test)
        acc.config(text="Accuracy Score = "+str(accuracy_score(y_test, y_pred))[:5])
        acc1.config(text="F1 Score              = " + str(f1_score(y_test, y_pred, average='macro'))[:5])
        acc2.config(text="Recall Score      = " + str(recall_score(y_test, y_pred, average='macro'))[:5])
        y_predicting = y_pred
```

Figure 4.2 Les algorithme d'apprentissage

La bibliothèque Scikit-learn contient différent méthode Tel que TF-IDF.

```
def extractFeat():
    global featuress
    try:
        global vec
        if combo1.get() == "TfIdf Vectorizer":
            from sklearn.feature_extraction.text import TfidfVectorizer
            vec = TfidfVectorizer(analyzer='word', stop_words='english')
            xx=vec.fit(df.text.values)
            features = xx.transform(df.text.values)
            featuress = features.toarray()
            msg.showinfo('Info', 'Text Features Extracted')
```

Figure 4.3 Méthode de TF-IDF

Pandas

Est une bibliothèque python utilisée pour Traitement et analyse des données En particulier.

```
def getFileName():
    global df
    try:
        import pandas as pd
        fDir = path.dirname(__file__)
        fName = fd.askdirectory(parent=win, initialdir=fDir)
        random_state = 0
        data = load_files(fName, encoding="utf-8", decode_error="replace", random_state=random_state)
        df = pd.DataFrame(list(zip(data['data'], data['target'])), columns=['text', 'label'])
    except:
        msg.showerror('Error','Error While Loading Data Files')
```

Figure 4.4 Lire des fichiers texte

Tkinter

C'est une bibliothèque pour créer des interfaces graphiques pour des programmes en Python. Ils sont installés une fois l'environnement Pyhton installé.

4.3 L'environnement de développement

PyCharm

Est un environnement de développement intégré (IDE) utilisé dans la programmation informatique, spécifiquement pour le langage Python. Il est développé par la société tchèque JetBrains. Il fournit l'analyse de code, un débogueur graphique, un testeur d'unité intégré, l'intégration avec des systèmes de contrôle de version (VCS) et prend en charge le développement Web avec Django ainsi que la science des données avec Anaconda.

Figure 4.5 PyCharm

4.4 Application

4.4.1 Présentation De La Fenêtre D'application

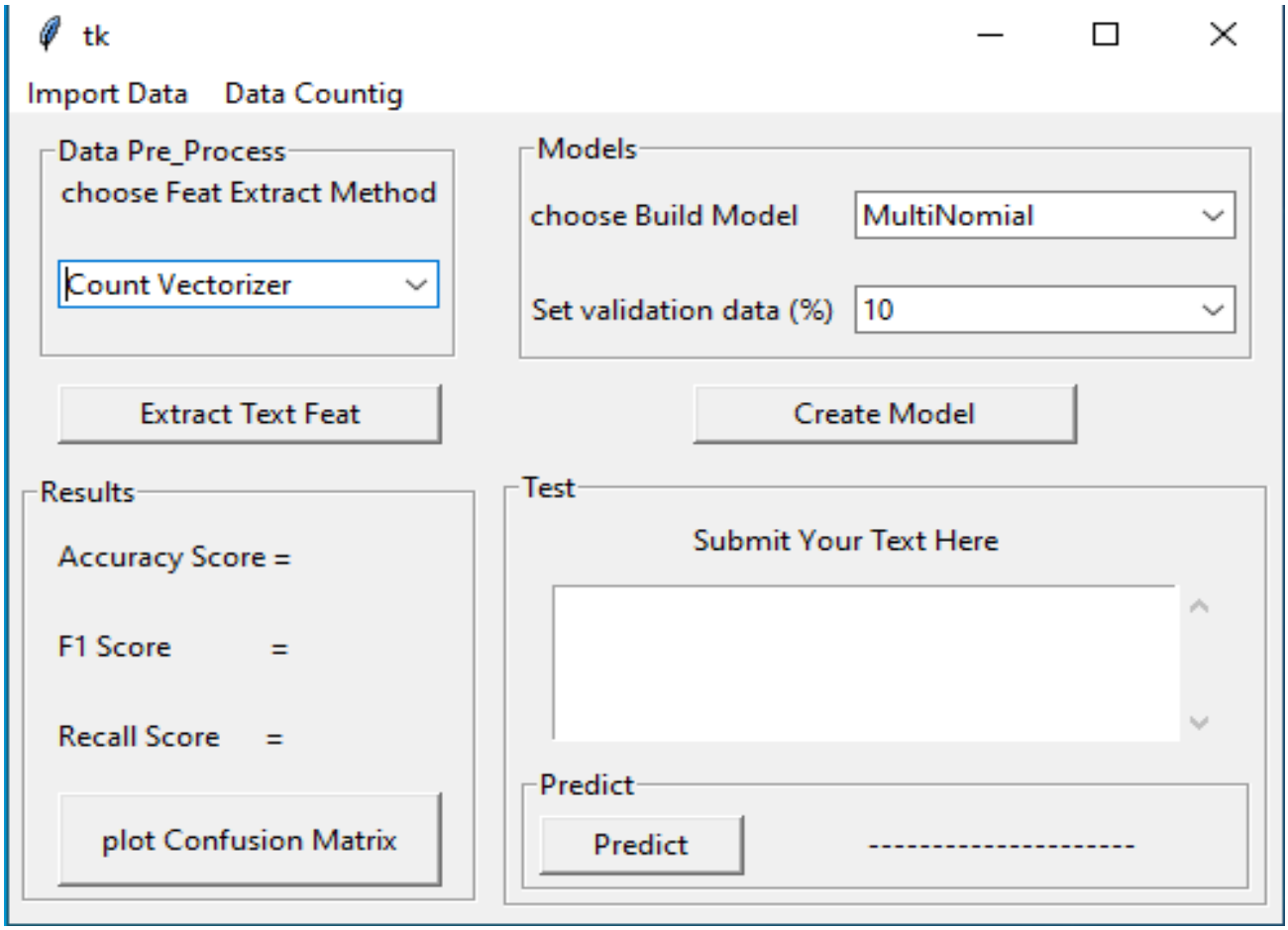


Figure 4.6 Fenêtre1 d'application

4.4.2 Les tache

Dans notre application différentes taches effectuées pour la classification a savoir: le prétraitement, l'entraînement, validation et l'utilisation.

4.4.2.1 Prétraitement

L'objectif de prétraitement est d'extraire les meilleures caractéristiques des termes et faire analyse, en fin classement de document.

Le graphe suivant Affiche la distribution des données dans les classes de base de données.

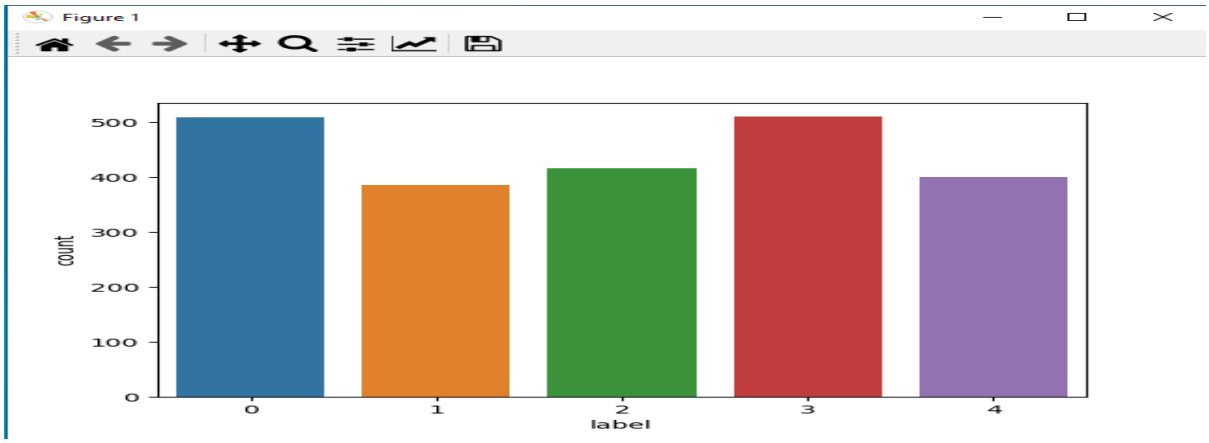


Figure 4.7 Distribution des données

Nous choisissons d'abord la base de données, puis choisissons la Méthode d'extraire les caractéristiques pour faire l'analyse.

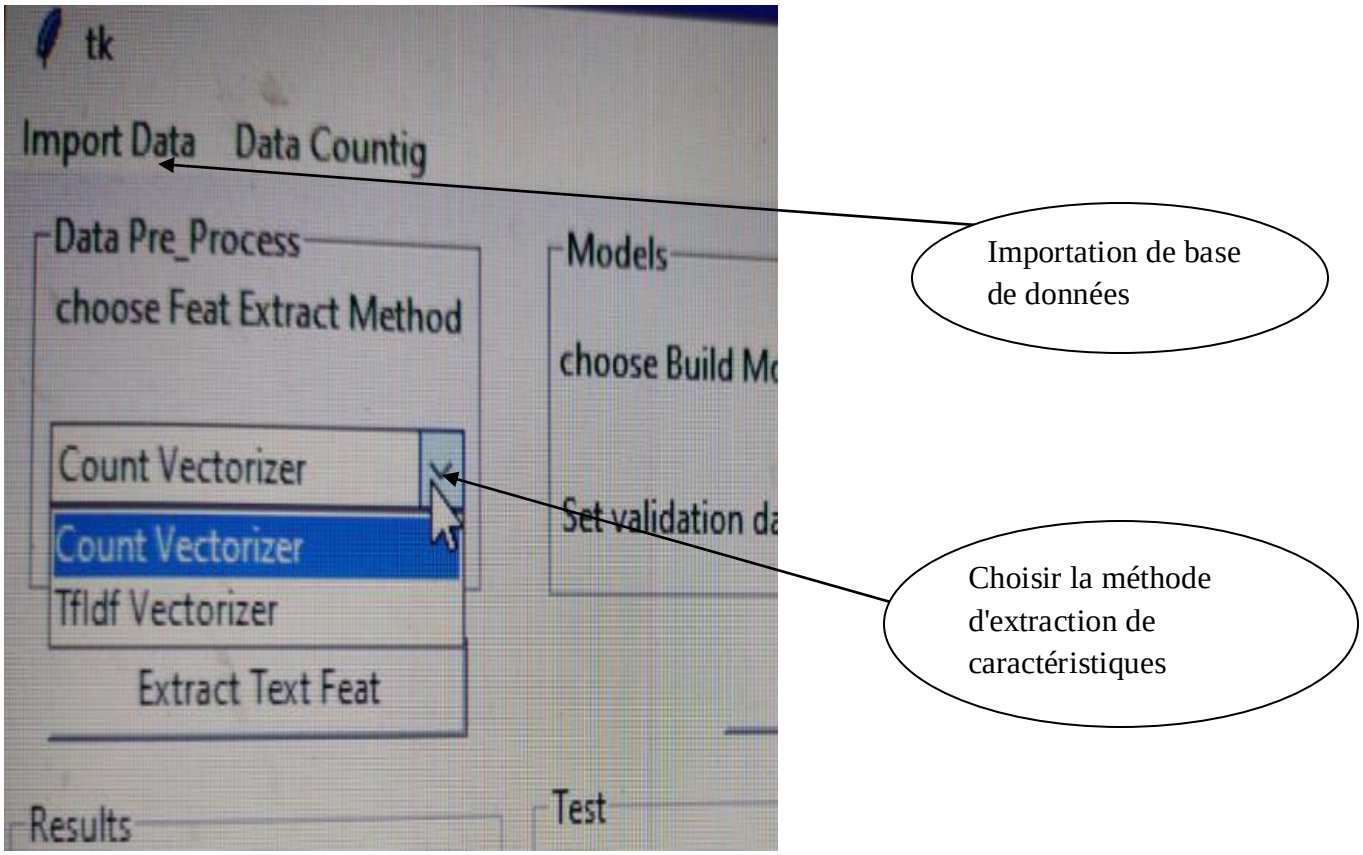


Figure 4.8 Méthode d'extraire les caractéristiques

Nous appuyons sur le bouton **(Extract text feat)** pour d'extraire les caractéristiques.

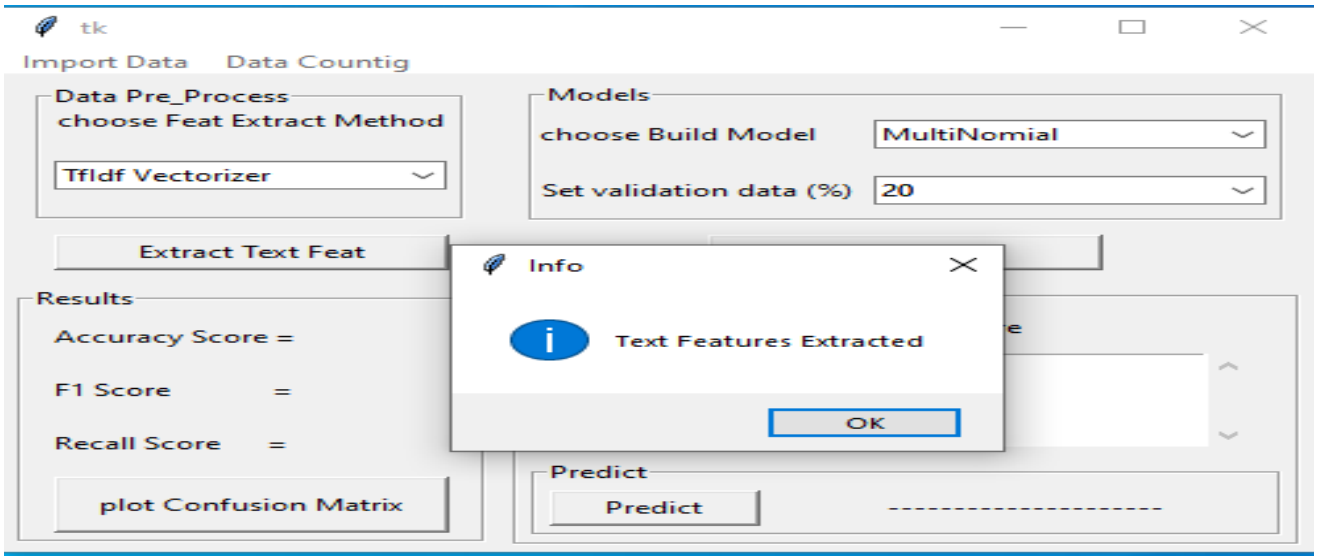


Figure 4.9 Extraire les caractéristiques

4.4.2.2 L'entraînement

C'est une étape qui permet aux utilisateurs d'entraîner les données et voir les performances du modèle (Rappel, Précision, F1 Score, Matrice de confusion).

La figure (ci-dessous) représente la sélection de l'algorithme d'apprentissage.

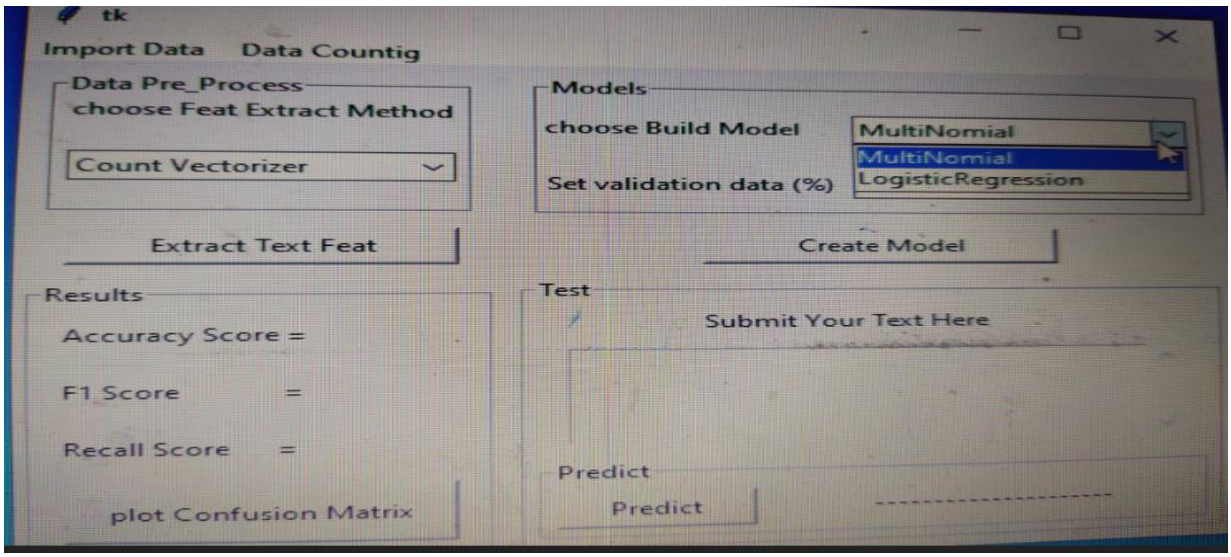


Figure 4.10 Sélection de l'algorithme d'apprentissage

Ensuite, nous appuyons sur le bouton **(Create model)** pour créer le model.

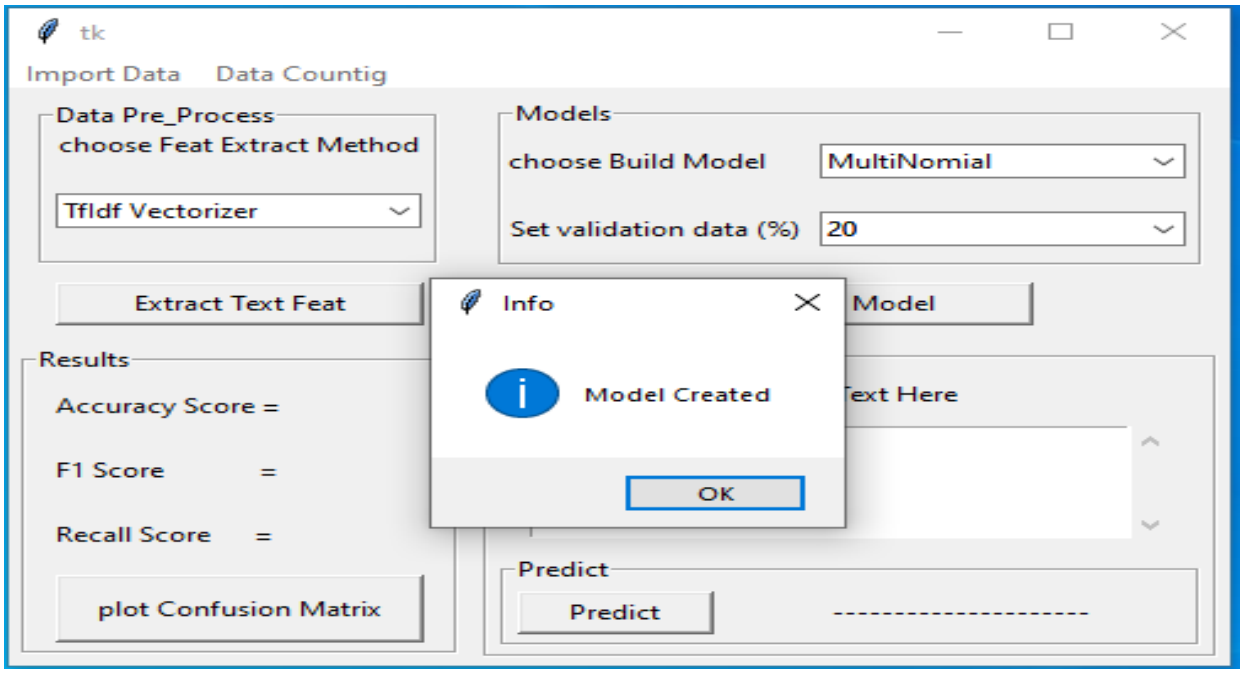


Figure 4.11 Creation de model

La figure (ci-dessous) représente les performances du modèle (Rappel, Précision, F1 Score, Matrice de confusion).

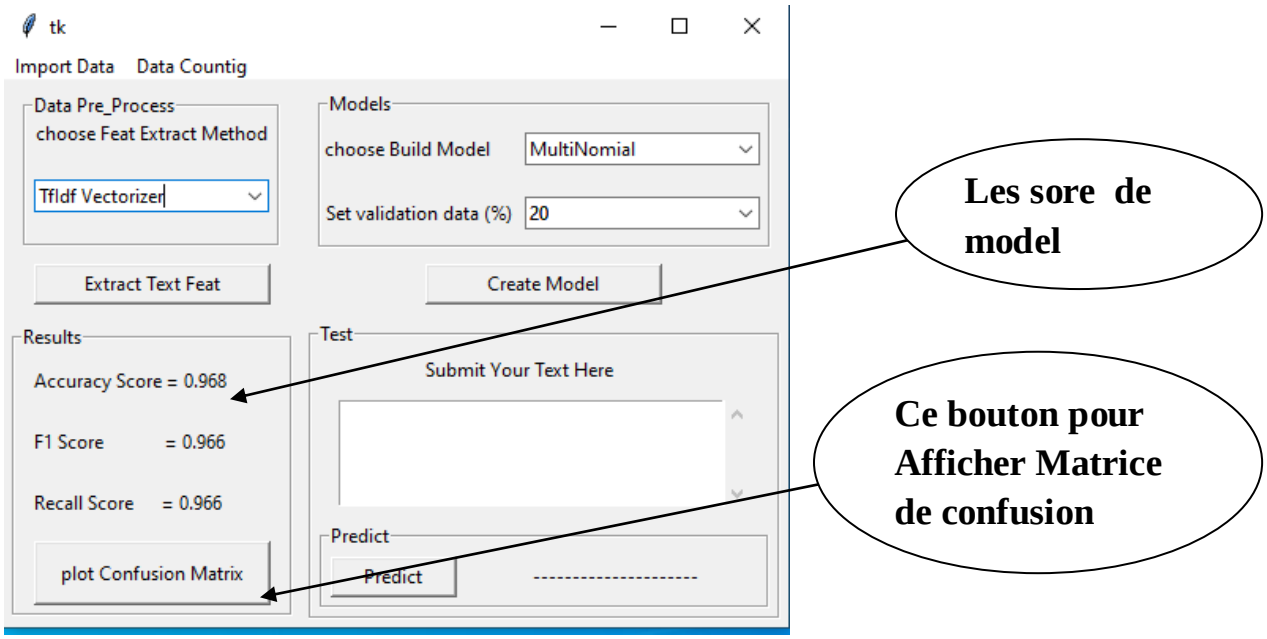


Figure 4.12 Les score de model

La figure suivante illustre la matrice de confusion

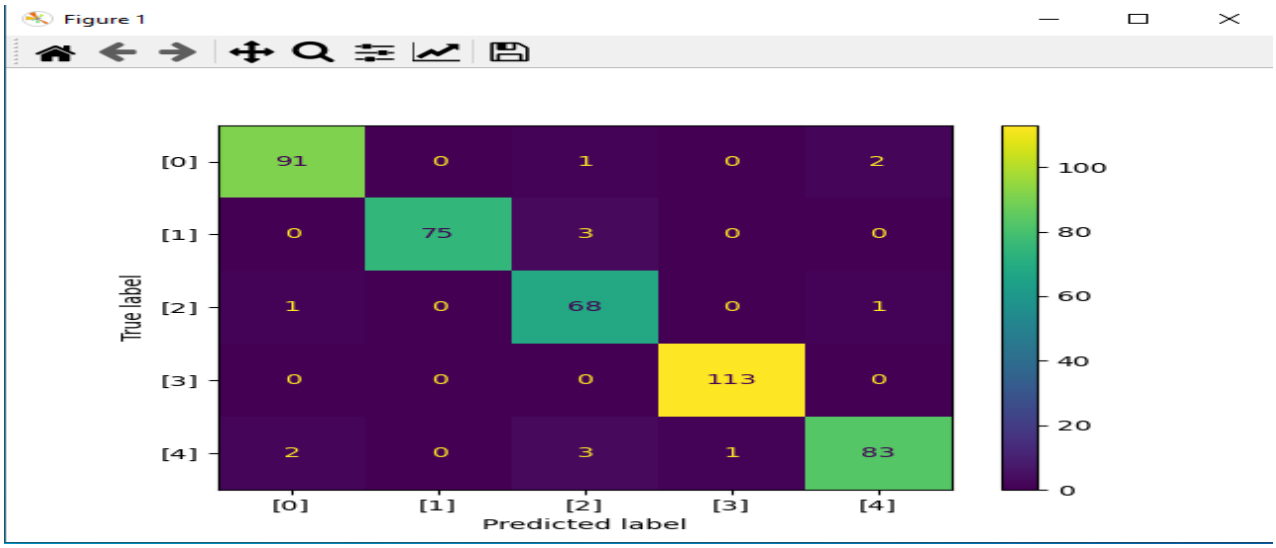


Figure 4.13 Matrice de confusion

4.4.2.3 Validation

C'est une étape qui permet aux utilisateurs de tester des modèles sur de base autre que celles utilisé lors de l'entraînement.

Dans notre application le dataset est subdivisé en deux partie, la première pour l' apprentissage, et la deuxième pour le test. généralement Nous choisissons 60% : apprentissage et 40% : test.

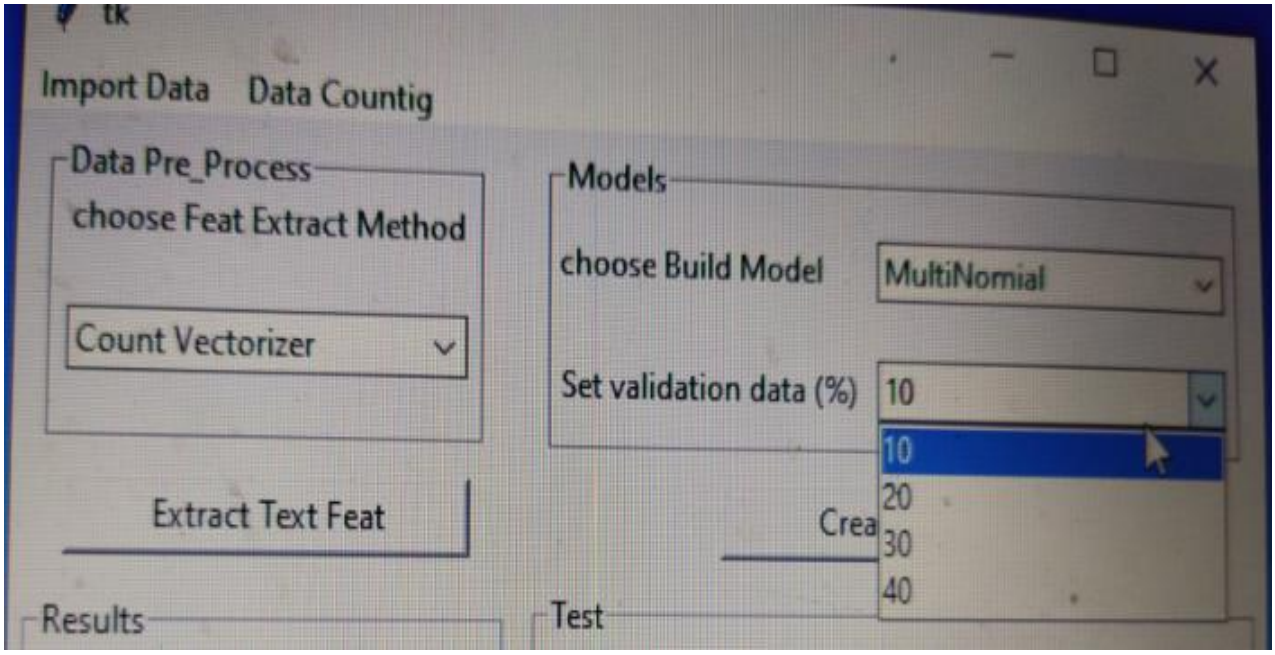


Figure 4.14 Subdivisé le dataset

4.4.2.4 Utilisation

Cette étape destinée à d'utilisateur. l'utilisateur peut entrer une nouvelle document text pour obtenir la catégorie de cette document.

Ceci est un exemple dans la figure ci-dessous

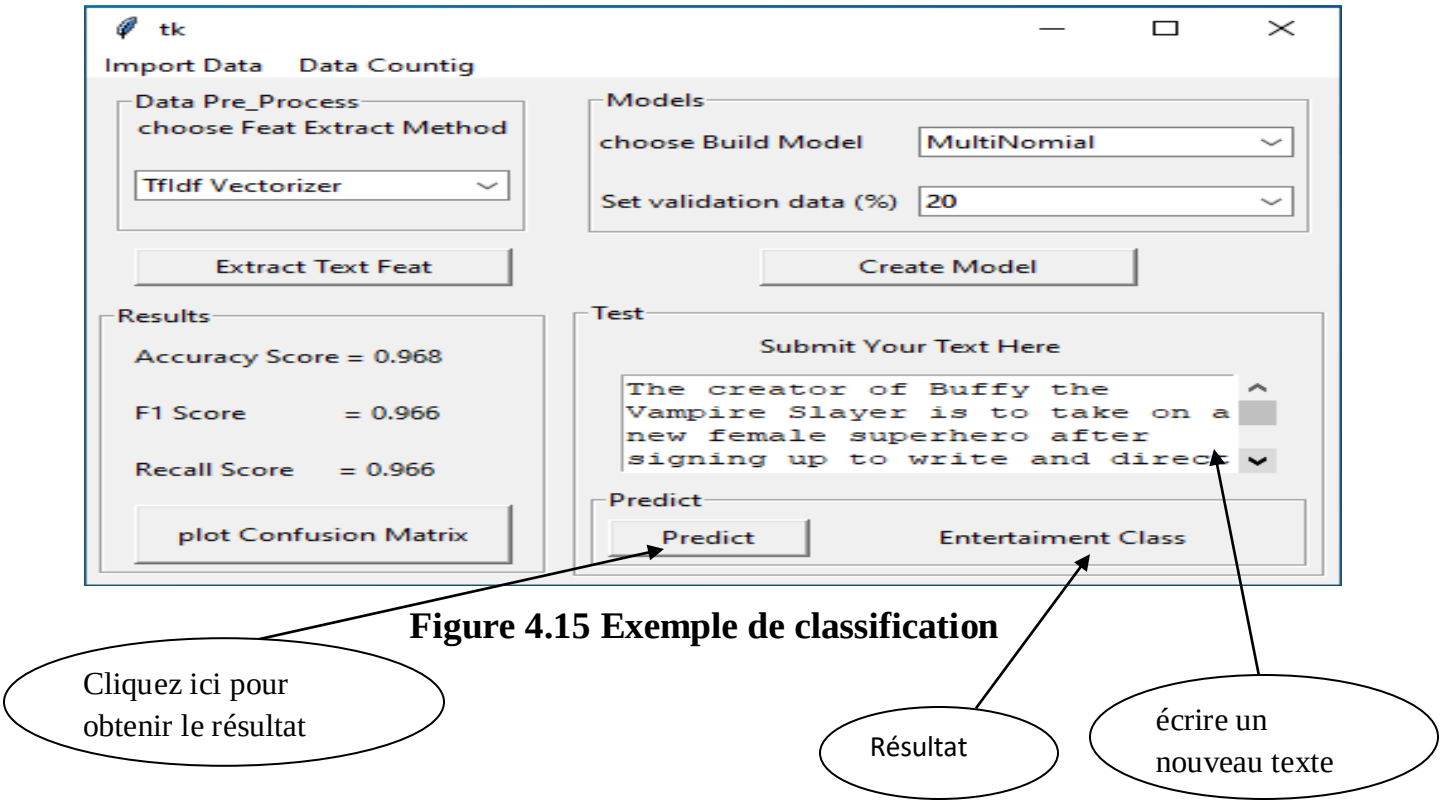


Figure 4.15 Exemple de classification

4.5 Conclusion

Dans ce chapitre, nous avons représenté l'implémentation de notre système, où nous avons montré l'environnement et les outils de développement qu'on a utilisé. Ensuite nous avons expliqué L'entraînement du modèle et les paramètres utilisé et le test, et aussi nous avons montré l'utilisation de notre modèle, comme nous avons aussi présenté les interfaces graphiques de notre système. Finalement nous avons expliqué les expérimentations et les résultats obtenues.

CONCLUSION GENERALE

Dans ce mémoire, nous nous sommes intéressés à la catégorisation des documents avec la méthode des Naïve bayes et Régression logistique. 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.

Nous avons présenté un travail regroupant le maximum de concepts et d'approches de fouille de texte et de classification nous avons présenté les différents algorithmes et montré leurs avantages et inconvénients pour la classification textuelle.

Nous avons commencé par étudier le domaine fouille de données (data mining) et fouille de texte (text mining), Nous avons ensuite conçu et implémenté une solution qui se base sur l'utilisation des techniques de nettoyage, encodage par sac à mots et TF-IDF pour le pré-traitement des textes. Ensuite l'extraction des autres caractéristiques permettant d'identifier une catégorie de document À la fin nous avons appliqué l'algorithme de naïve bayes ou Régression logistique sur notre base de caractéristiques pour construire un modèle permettant la classification des nouvelles textes.

Après l'analyse des résultats obtenus, on a pu constater que les taux de classification sont acceptables, Pour pouvoir comparer les résultats obtenus dans les différentes expérimentations, on a utilisé les mesures de performance Rappel, Précision et F1 score.

Nous estimons avoir atteint les objectifs que nous nous étions fixés, en présentant les techniques de fouille de données et de classification et En réalisant un analyseur qui prépare les données textuelles par une suite d'opérations pour être exploitable par les techniques de fouille de données.

5 Bibliography

1. B.Agard, A.Kusiak. Exploration des bases de données industrielles à l'aide du Data Mining – Perspectives. *.en 9ème colloque national AIP PRIMECA, Avril 2005*. 2005.

2. Principales tâches du data mining. https://www.petite-entreprise.net/P-2595-83-G1-principales-taches-du-data-mining.html.

3. Abdelhamid DJEFFAL. Cours Fouille de données avancée. 2018/2019. http://abdelhamid-djeffal.net/web_documents/firstfda.pdf.

4. Mr DAHMANI Djilali. MEMOIRE DE MAGISTER,Fouille des règles d'association guidée par des ontologies et des schémas de règles :Application au domaine de la production SONATRACH / AVAL. 2011.

5. Définitions du Text mining. https://touriaelouahabi.wordpress.com/text-mining/definition-du-text-mining/.

6. Gherabi Sara. MEMOIRE de fin d'étude UNIVERSITE DE M'SILA,CLASSIFICATION AUTOMATIQUE DES TEXTES ARABE. 2011.

7. Manu Konchady. Text Mining Application Programming, Charles River Media Programming series,USA. 2007.

8. HARISH, Bhat S, GURU, Devanur S, MANJUNATH, Shantharamu. Representation and Classification of Text Documents. Published By Foundation of Computer Science, 2010. https://www.ijcaonline.org/specialissues/rtippr/number2/984-107.

9. MATA, Javier, et al. Artificial intelligence (AI) methods in optical networks: A comprehensive survey. Optical switching and networking. Avr 2018. https://www.sciencedirect.com/science/article/pii/S157342771730231X.

10. Machine Learning. . *en What it is and why it matters.* Joi 15, 2021.

11. Julien Ah-Pine. Apprentissage automatique. 2019/2020. https://eric.univ-lyon2.fr/~jahpine/cours/m2_dm-ml/cm.pdf.

12. Guillaume Saint-Cirgue. ,Machine Learning,Apprentissage Supervisé. Jui 02, 2019.

13. Zakariyaa ISMAILI. apprentissage-supervise-vs-non-supervise. https://analyticsinsights.io/apprentissage-supervise-vs-non-supervise/.

14. Juanita Dagnon. Les applications du Machine Learning en Big Data. Jan 27, 2015. https://fr.blog.businessdecision.com/machine-learning/.

15. Younes Benzaki. Machine Learning applications : 10 cas d'usage pratiques,. Aou 29, 2017. https://mrmint.fr/machine-learning-applications.

16. Ivy Professional School. Advantages and Disadvantages of Machine Learning. Fév 2020. https://www.pinterest.com/pin/687713805588560689/.

17. Classification naïve bayesienne - Définition et Explications. https://www.techno-science.net/glossaire-definition/Classification-naive-bayesienne.html.

18. Classification naïve bayésienne. https://course.elementsofai.com/fr/3/3.

19. Rohith Gandhi. Naive Bayes Classifier. Mai 05, 2018. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c#:~:text=Bayes%20Theorem%3A&text=The%20assumption%20made%20here%20is,Hence%20it%20is%20called%20naive..

20. La Rédaction. Classification naïve bayésienne : définition et principaux avantages. Mai 19, 2021. https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501321-classification-naive-bayesienne-definition-et-principaux-avantages/.

21. Pavan Vadapalli. Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2021. Jan 05, 2021. https://www.upgrad.com/blog/naive-bayes-explained/.

22. Younes Benzaki. Logistic Regression pour Machine Learning – Une Introduction Simple. Sep 06, 2017. https://mrmint.fr/logistic-regression-machine-learning-introduction-

simple?fbclid=IwAR1lOZ7GwpceVvyFDHCmiNwgqL5D7mIsEMGb6Ylj39CPbsoAf-oEzGCSLq8.

Annexe

Ce code affiche le contenu des fichiers texte de la base de données et la catégorie à laquelle ils appartiennent

```
data.head()
```

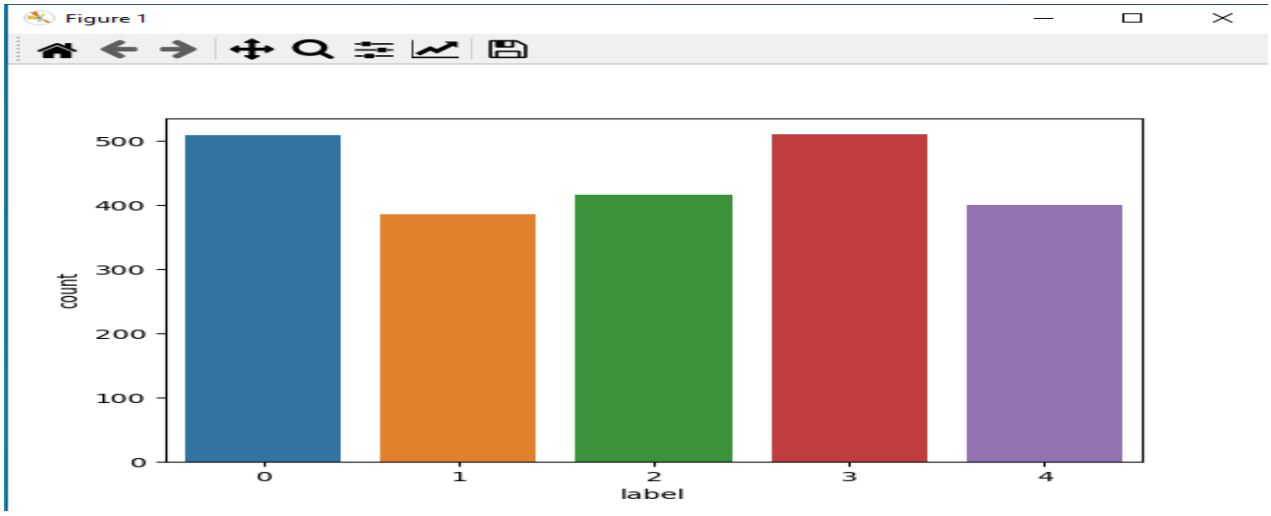
	ArticleId	Text	Category
0	1833	worldcom ex-boss launches defence lawyers defe...	business
1	154	german business confidence slides german busin...	business
2	1101	bbc poll indicates economic gloom citizens in ...	business
3	1976	lifestyle governs mobile choice faster bett...	tech
4	917	enron bosses in $168m payout eighteen former e...	business

Divisée la base en deux partie une pour l'entraînement et l'autre pour le test.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=(int(combo3.get())/100), random_state=0)
```

Le graphe suivant Affiche la distribution des données dans les classes de base de données.

```
def datacounting():
    sns.countplot(x='label', data=df)
    show()
```



Ce code pour couper le texte en token et éliminer les mots vide

```
def process_text(text)
    text = text.lower().replace('\n',' ').replace('\r','').strip()
    text = re.sub('+',' ',text)
    text = re.sub(r',[^\w\s;]','',text)

    stop_word = set(stopword.words(english))
    word_tokens = word_tokenize(text)
    filtered_sentence = [w for w in word_tokens if not w in stop_word]

    text = ' '.join(filtered_sentence)
    return text
```

Ce texte est avant et après la tokenization et éliminer les mots vide

	ArticleId	Text	Category	News_length	Text_parsed
0	1833	worldcom ex-boss launches defence lawyers defe...	business	1866	worldcom exboss launches defence lawyers defen...
1	154	german business confidence slides german busin...	business	2016	german business confidence slides german busin...
2	1101	bbc poll indicates economic gloom citizens in ...	business	3104	bbc poll indicates economic gloom citizens maj...
3	1976	lifestyle governs mobile choice faster bett...	tech	3618	lifestyle governs mobile choice faster better ...
4	917	enron bosses in $168m payout eighteen former e...	business	2190	enron bosses 168m payout eighteen former enron...

Appliquer la méthode TF-IDF et extraire les caractéristique.

```
vec = TfidfVectorizer(analyzer='word', stop_words='english')
vec.fit(df.text.values)
features = vec.transform(df.text.values)
features = features.toarray()
print(features)
```

```
[[0.         0.01584596 0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]]
```

Appliquer l'algorithme de Multinomial naïve bayes .

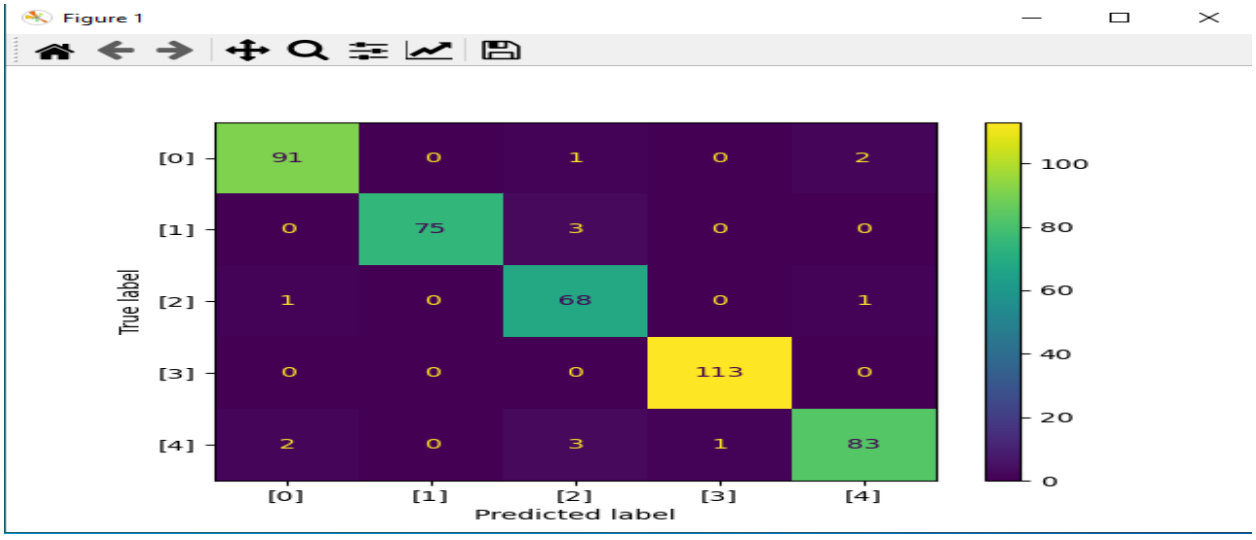
```
model = MultinomialNB()
history = model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

C'est le résultat de la prédiction où chaque nombre (0, 1, 2, 3, 4) représente une classe (politic, sport, technologie, business, Entertainment)

```
[1 2 0 0 3 2 1 0 3 3 3 2 3 1 3 2 4 2 3 0 4 0 0 4 1 3 1 1 2 2 3 1 2 2 2 3 1
 4 1 4 4 3 2 4 0 4 1 2 0 2 1 1 0 1 4 4 2 0 4 2 2 0 3 4 4 1 0 4 2 2 1 3 2 1
 0 1 3 0 1 4 0 4 3 2 0 1 4 1 3 2 1 3 3 4 1 0 3 3 0 4 2 3 0 1 2 3 0 3 2 3 2
 3 2 3 3 3 0 0 0 3 4 3 0 0 0 0 4 4 4 2 2 0 1 0 1 0 1 3 1 4 1 4 0 0 4 2 2 2
 0 0 2 0 3 4 3 2 4 0 2 0 2 0 0 4 0 3 3 4 4 2 1 1 0 2 1 4 0 3 2 3 2 0 3 3 4
 2 3 0 4 1 3 0 3 1 3 1 3 1 0 0 0 4 2 4 2 3 2 1 0 2 1 0 1 2 3 4 3 4 1 0 4 1
 3 1 3 0 3 1 3 1 0 3 1 3 0 0 3 1 2 1 0 1 2 0 2 3 2 2 2 0 1 3 2 2 2 1 2 0 4
 0 2 4 4 3 2 3 0 1 2 3 2 3 0 3 0 2 1 1 1 3 4 4 0 1 2 2 3 3 3 2 0 0 0 1 1 0
 3 2 3 4 3 1 3 2 1 3 1 4 3 1 0 4 3 3 2 3 0 2 3 2 1 3 1 4 0 1 4 2 4 0 2 4 3
 3 0 0 4 0 4 2 3 0 0 0 0 0 1 0 0 1 0 3 4 1 0 3 4 4 4 4 4 4 0 4 0 3 2 0 1 3
 4 3 4 0 2 4 1 4 3 3 4 3 3 0 1 1 4 0 2 2 0 2 0 4 2 3 0 1 1 3 4 4 4 3 4 3 1
 3 1 2 2 3 3 4 3 0 2 0 2 2 2 3 0 2 3 4 2 4 0 0 4 2 0 2 4 0 4 0 0 2 2 1 2 0
 0 2 0 0 4 0 0 3 1]
```

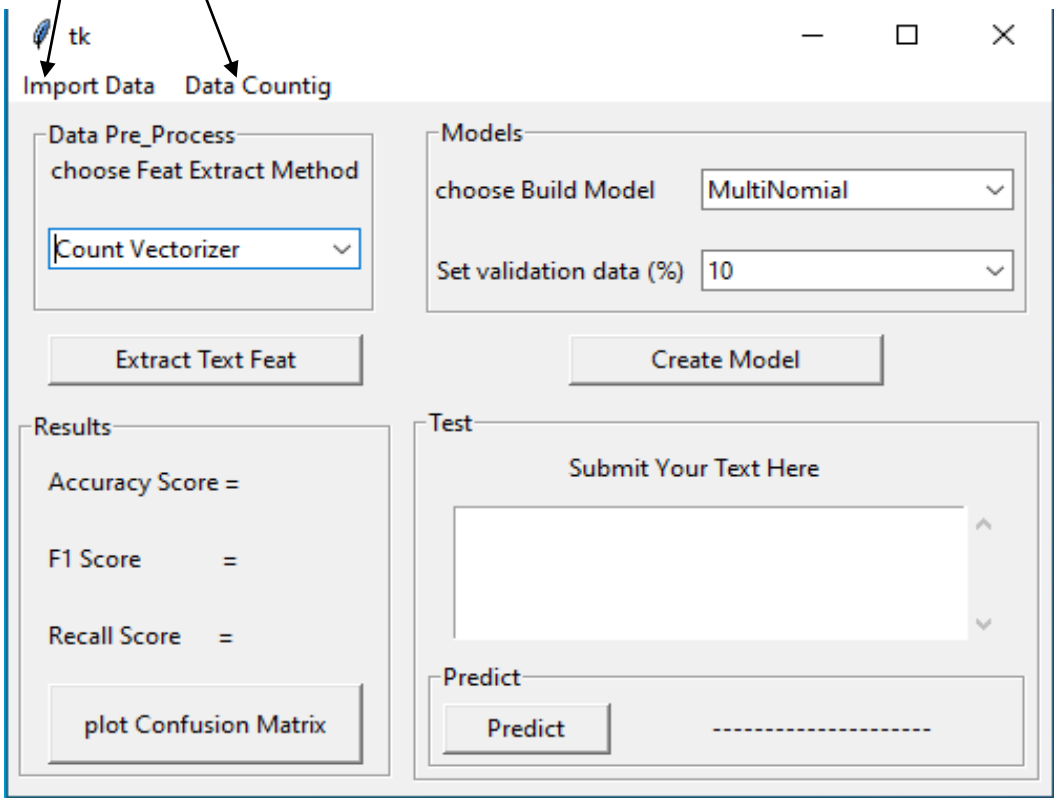
Ce Code pour construire la matrice de confusion.

```
cc = confusion_matrix(y_testing, y_predicting)
import matplotlib.pyplot as plt
disp = ConfusionMatrixDisplay(cc, display_labels=[[0], [1], [2], [3], [4]])
disp.plot(include_values=True, cmap='viridis',
                  xticks_rotation='horizontal', values_format='d', ax=None)
plt.show()
```



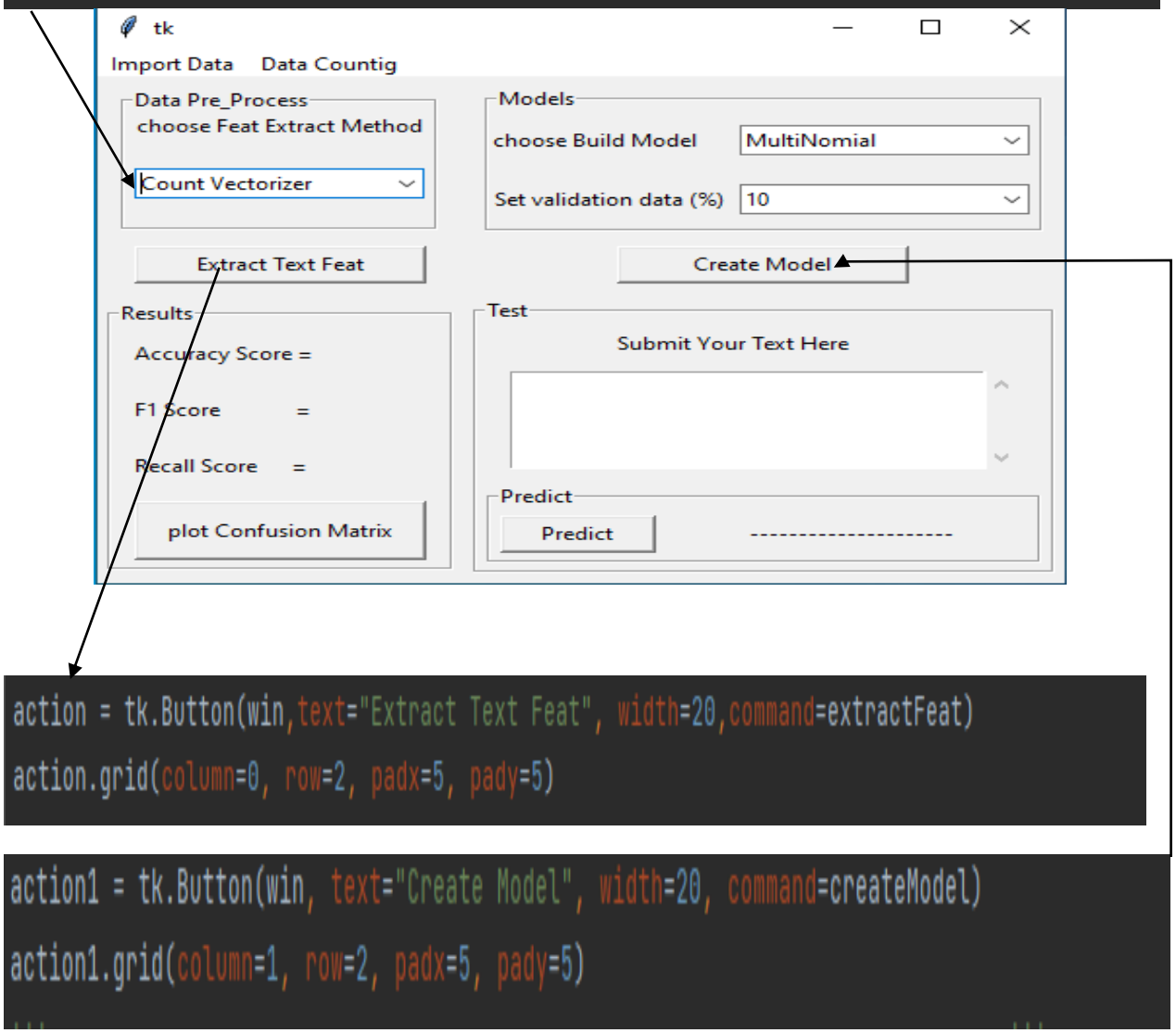
Les deux buttons suivants pour Importation de base de données et Affiche la graphe de distribution des données dans les classes de base de données.

```
win = tk.Tk()
menubar = Menu(win)
menubar.add_command(label="Import Data", command=getFileName)
menubar.add_command(label="Data Countig", command=datacounting)
win.config(menu=menubar)
```



Ce code pour Choisir la méthode d'extraction de caractéristiques.

```
combo1 = ttk.Combobox(groupbox, width=21, textvariable=number)
combo1['value'] = ('Count Vectorizer', 'TfIdf Vectorizer')
combo1.grid(column=0, row=1, padx=5, pady=20)
combo1.current(0)
```



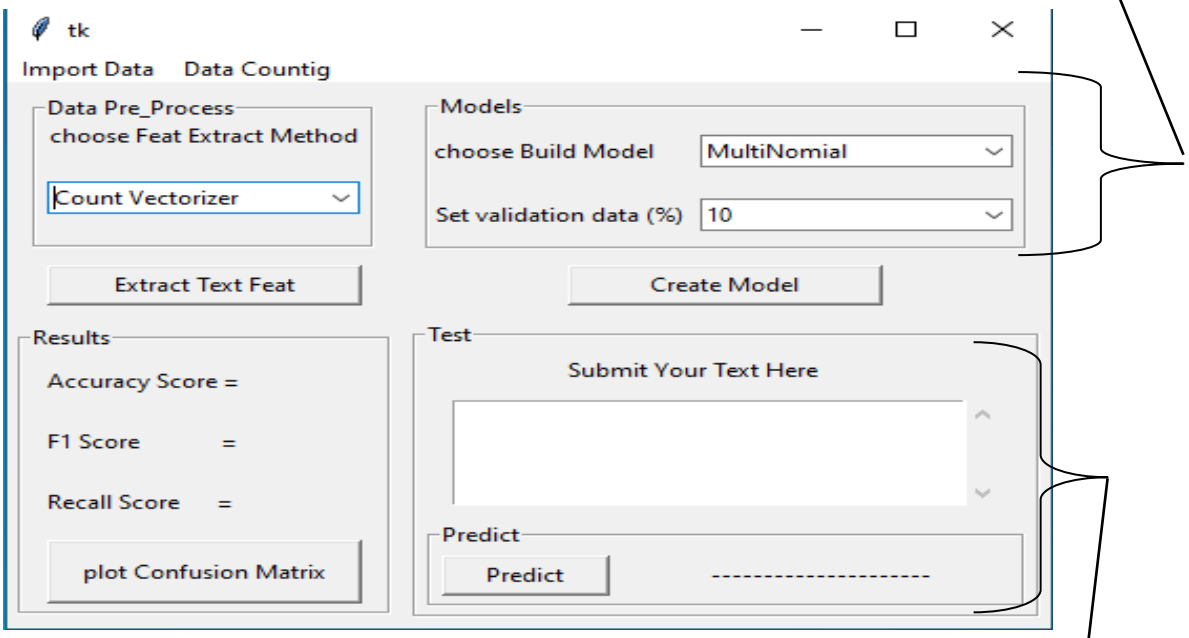

```
action = tk.Button(win,text="Extract Text Feat", width=20,command=extractFeat)
action.grid(column=0, row=2, padx=5, pady=5)
```

```
action1 = tk.Button(win, text="Create Model", width=20, command=createModel)
action1.grid(column=1, row=2, padx=5, pady=5)
```

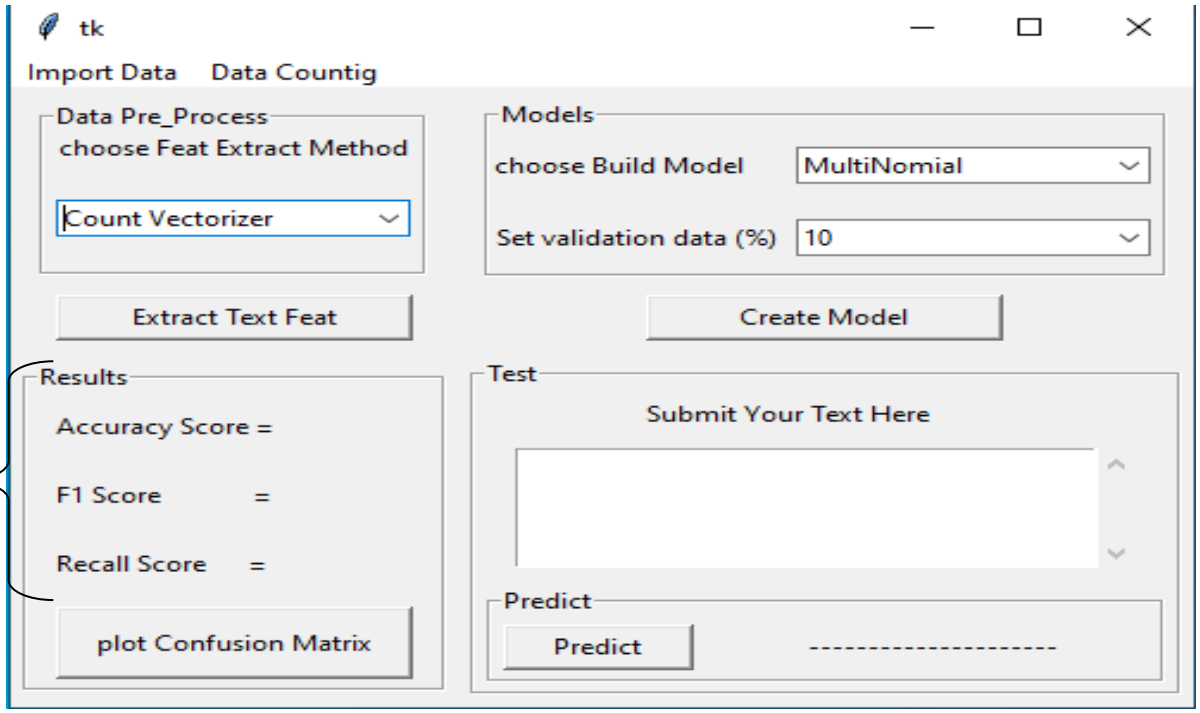
Lorsque nous cliquons sur les boutons ci-dessus, il effectue la méthode qui se trouve dans « command »

Ces codes représentent des parties de notre application

```
groupbox1 = tk.LabelFrame(win, text='Models')
groupbox1.grid(column=1, row=0, padx=5, pady=5)
ttk.Label(groupbox1, text="choose Build Model     ",).grid(column=0, row=0)
combo2 = ttk.Combobox(groupbox1, width=21, textvariable=number1)
combo2['value'] = ('MultiNomial', 'LogisticRegression')
combo2.grid(column=1, row=0,padx=5, pady=10)
combo2.current(0)
ttk.Label(groupbox1, text="Set validation data (%)",).grid(column=0, row=1)
combo3 = ttk.Combobox(groupbox1, width=21, textvariable=number2)
combo3['value'] = (10,20,30,40)
combo3.grid(column=1, row=1,padx=5, pady=10)
```

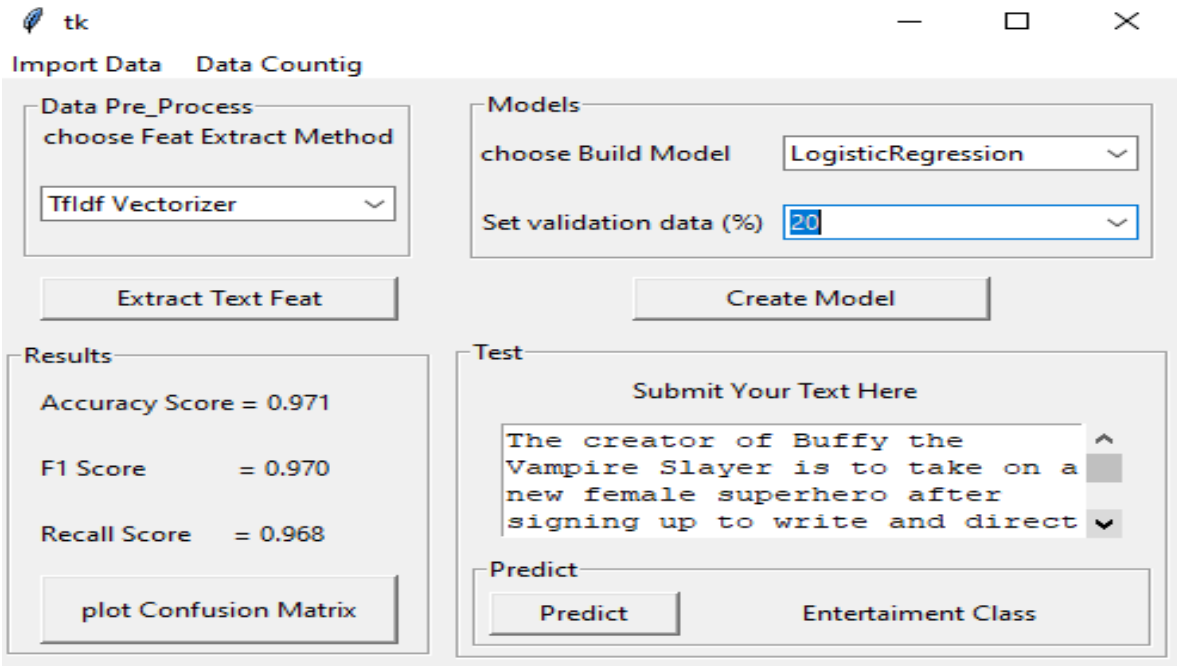



```
groupbox3 = tk.LabelFrame(win, text='Test')
groupbox3.grid(column=1, row=3, padx=5, pady=5)
ttk.Label(groupbox3,width=25, text="Submit Your Text Here",).grid(column=0, row=0, padx=5, pady=5)
scr = scrolledtext.ScrolledText(groupbox3, width=30, height=4,wrap=tk.WORD)
scr.grid(column=0, row=1, padx=5, pady=5)
groupbox4 = tk.LabelFrame(groupbox3, text='Predict')
groupbox4.grid(column=0, row=2, padx=5, pady=5)
action3 = tk.Button(groupbox4,text="Predict", width=10,command=makeprediction)
action3.grid(column=0, row=0, padx=5, pady=5)
acc3 = tk.Label(groupbox4,width=25, text="----------",)
acc3.grid(column=1, row=0, padx=5, pady=5)
```

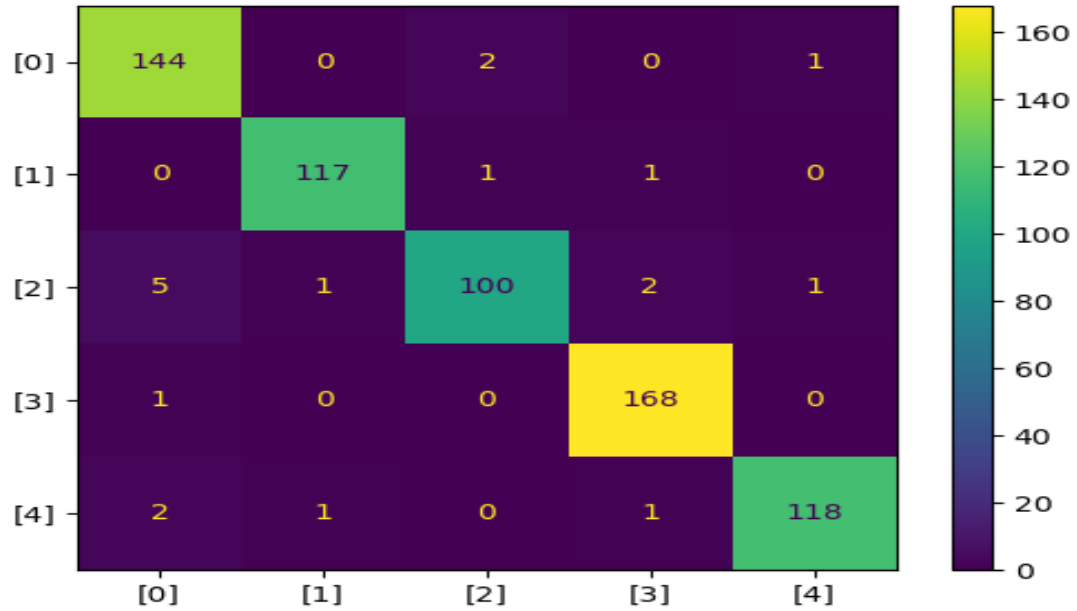



```
groupbox2 = tk.LabelFrame(win, text='Results')
groupbox2.grid(column=0, row=3, padx=5, pady=5)
acc = ttk.Label(groupbox2, width=25, text="Accuracy Score = ")
acc.grid(column=0, row=0, padx=10, pady=10)
acc1 = ttk.Label(groupbox2, width=25, text="F1 Score            = ")
acc1.grid(column=0, row=1, padx=10, pady=10)
acc2=ttk.Label(groupbox2, width=25, text="Recall Score      = ")
acc2.grid(column=0, row=2, padx=10, pady=10)
action2 = tk.Button(groupbox2,text="plot Confusion Matrix", width=20,height=2,command=displaycnf)
action2.grid(column=0, row=3, padx=5, pady=5)
```

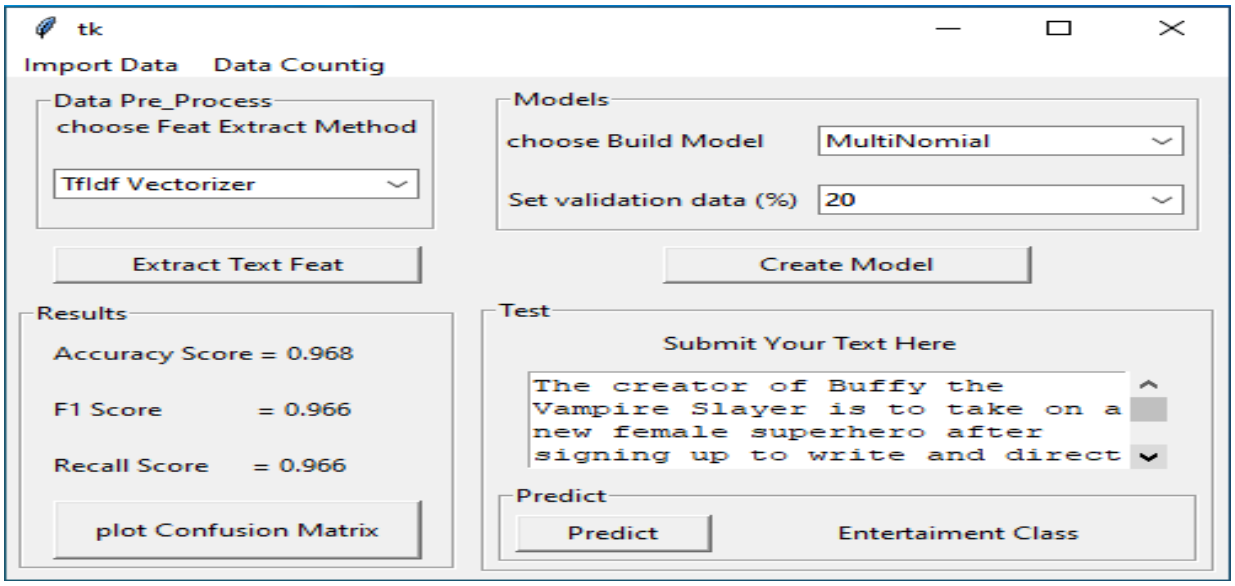
Dans la partie suivante, nous présentons deux exemples de fonctionnement de notre application Le premier est exécuter l'algorithme de naïve bayes et le deuxième est exécuter l'algorithme de Régression logistique, Et enfin, on montre la matrice de confusion de chacun



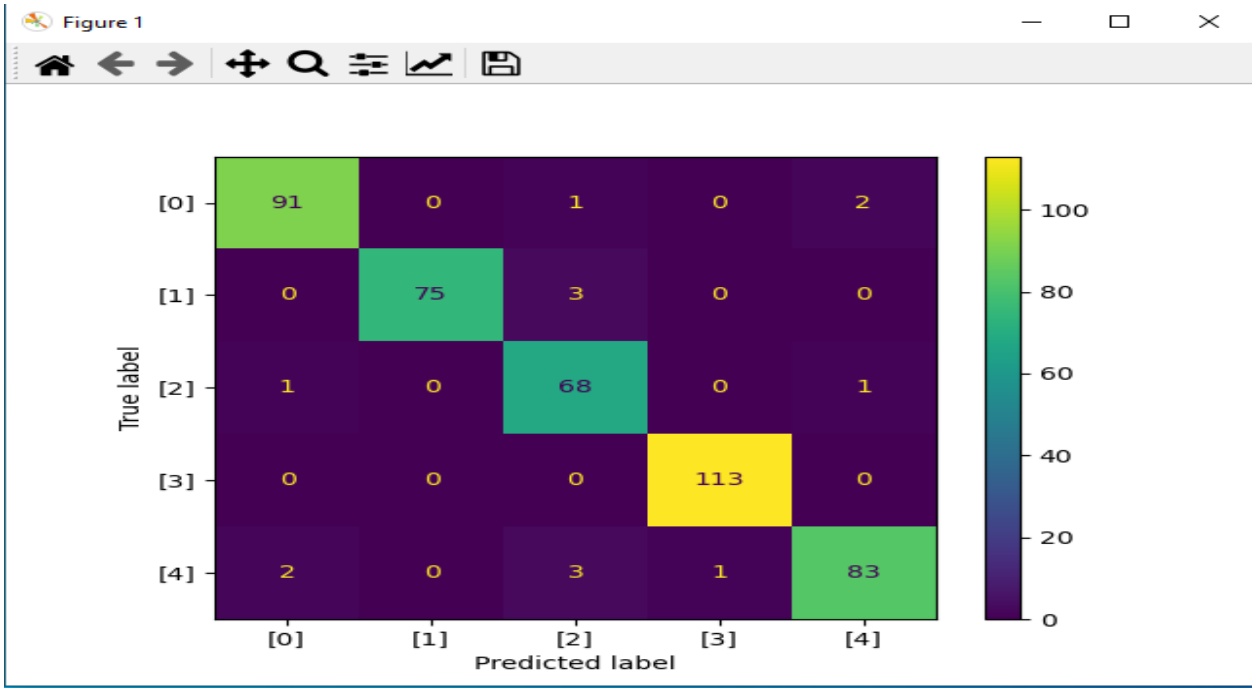
Nous remarquons que l'algorithme de régression logistique correctement prédit la classe pour le nouveau texte



La matrice de confusion de régression logistique



Nous remarquons que l'algorithme de naïve bayes correctement prédit la classe pour le nouveau texte



La matrice de confusion de naïve bayes

Notons que les deux résultats sont très similaires.