REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
NISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité :Génie Informatique (GI)

Par :

BOUSSAHA MOHAMED
BOUDIAF MOURAD

Sur le thème

Moodle forum text mining

Soutenu publiquement le 03/10 /2021 à Tiaret devant le jury composé de :

Mr. MERATI Medjeded	MCA	Université de Tiaret	Président
Mr.TALBI Omar	MCB	Université de Tiaret	Encadreur
Mr.OUARED Abdelkader	MCB	Université de Tiaret	Examinateur

2020-2021

DEDICACE

Je dédie ce mémoire à :

Mes chers parents, que nulle dédicace ne puisse exprimer mes sincères sentiments, pour leur aide, leur encouragement et leur patience illimitée.

Mes chers frères et sœurs, Mes chers amis (e), qui sans leur encouragement ce travail n'aura jamais vu le jour.

Et à toute ma famille, et à tous ceux que j'aime.

A ma deuxième famille et ma deuxième école, École de l'Union des étudiants libres généraux

Remerciements

Nous remercions Dieu Tout-Puissant qui nous a permis de terminer cet humble travail

Nous adressons nos sincères remerciements et notre gratitude au Dr Talbi Omar, qui nous a accompagnés de ses conseils et de Ses astuces, et qui a été comme un père pour nous dans notre cheminement vers la réalisation de ce travail

Mes remerciements vont aussi à tous les membres du jury qui ont accepté de lire et d'évaluer

Ce travail.

Nous remercions également nos familles pour leur soutien et leurs encouragements tout au long de notre parcours universitaire depuis le début jusqu'à la fin de notre note de fin d'études.

Nous tenons à remercier tous ceux qui nous ont soutenus et encouragés à mener à bien ce travail, que ce soit de près ou de loin.

Résumé

Les forums de discussion en ligne forment des communautés de personnes qui apprennent les unes des autres, que ces forums soient synchrones ou asynchrones, ils génèrent beaucoup de données, et comme ces données ne cessent de croître, il devient difficile de gérer et d'analyser cela ce qui met l'enseignant dans un position difficile en termes de temps et d'efforts.

Ce travail présente une étude visant à faciliter le travail de l'enseignant en analysant les messages des étudiants et en sachant lesquels d'entre eux sont liés ou non au sujet abordé par l'enseignant. Ce travail commence par la collecte des données représentées dans les messages « textes » des étudiants, puis nous les organisons. Et ce, en nettoyant le texte après quoi des algorithmes de classification lui sont appliqués. Finalement le résultat est une machine capable de classer les messages, qu'ils soient liés ou nonau sujet.

Mots-clés :

exploitation de texte, Moodle, Cours moodle, Forum moodle, Exploration de données.

Abstract :

Online discussion forums form communities of people who learn from each other, whether these forums are synchronous or asynchronous, they generate a lot of data, and as this data continues to grow, it becomes difficult to manage and analyze this which puts the teacher in difficult position in terms of time and effort .

This work presents a study aimed at facilitating the teacher’s work by analyzing the students messages and knowing which of them respond to the topic raised by the teacher, as this work begins with collecting data represented in the students’ messages “text”, then we organize and clean the text well after that we apply classification algorithms to it , and finally the result is a machine capable of classifying messages whether they are related to the topic or not.

Mots-clés :

Text mining, Moodle, Cours moodle, Forum moodle, Data mining.

Liste de Figure

Sommaire

Introduction générale

La diffusion massive des technologies de l'information et de la communication dans le monde de l'éducation, en particulier dans l'enseignement supérieur, a conduit à l'émergence d'une nouvelle forme d'apprentissage appelée e-learning et en raison de cette nouvelle éducation de nombreuses universités à travers le monde ont lancé des stratégies pour soutenir l'utilisation des technologies de l'information et de la communication dans l'éducation en développant des plateformes de formation en ligne telles qu'un système de gestion de l'apprentissage tel que Moodle.

Les activités pédagogiques pour les enseignants et les étudiants menées sur la plate-forme Moodle génèrent des empreintes numériques qui constituent une source de données importante dans le domaine de Educational data mining (EDM), Les techniques EDM ont été utilisées avec succès pour améliorer l'apprentissage des élèves et aider les enseignants à améliorer le processus d'apprentissage.

Problématique

Bien que les techniques de l'EDM aient contribué à améliorer le processus d'apprentissage, elles n'explorent pas pleinement toutes les ressources pédagogiques disponibles. Par exemple, il est courant d'avoir des activités d'apprentissage de type Forum qui pourraient être utilisées pour estimer l'efficacité des stratégies pédagogiques, mesurer la motivation des étudiants, étudier leur comportement, etc.

Le Forum est sûrement l'une des activités les plus importantes d'un Cours : il établit et stimule l'interaction de l'enseignant avec les étudiants ainsi que l'interaction entre pairs (entre étudiants), favorisant ainsi un apprentissage actif. Ces discussions en ligne sont souvent désorganisées et confuses, à cause de leur volume conséquent et du développement fréquent de nombreux fils de discussion et de conversations parallèles rendant ainsi leur traitement manuel quasi impossible par l'enseignant.

Objectifs

Dans ce PFE nous nous intéressons aux techniques de Text Mining (TM) qui pourraient être adoptées dans une activité Forum d'un Cours Moodle.

Nous procéderons au traitement de volumes conséquents de contenus textuels générés par ce type d'activité en extrayant les principales caractéristiques et tendances que nous utiliserons dans le processus de l'EDM, ceci d'une part. D'autre part cette technique va permettre également à l'enseignant d'automatiser le traitement des données contenues dans le Forum de son Cours et d'obtenir rapidement des éléments lui permettant d'apprécier la motivation et l'engagement des étudiants ainsi que l'évaluation de son enseignement.

Organisation du mémoire

Ce mémoire est composé de quatre chapitres organisés comme suit :

- Le chapitre I aborde les notions de base, à savoir, le e-learning ,les plateformes de formation dans l'enseignement supérieur, plus expressément Moodle, l'activité d'apprentissage de type forum, d'autres définitions concernant le Data Mining et le Text Mining ainsi que les étapes permettant de les mettre en œuvre y seront abordées.
- Le chapitre II se veut un état de l'art sur les travaux réalisés sur les activités de type forum dans la plateforme de formation Moodle. Nous en avons étudié trois, ceux de *Dringus & Ellis*, de *Azevedo- Reategui* and *Behar*et de *Cristóbal Romero*et ses collègues.
- Le chapitre III présente notre contribution, en expliquant le processus que nous avons retenu à travers ses différentes étapes.
- Le chapitre IV présente l'implémentation de notre solution et les outils qui nous ont permis d'y parvenir.

Chapitre 1 : Background

Introduction

Au cours des dernières décennies, la qualité de l'enseignement supérieur s'est améliorée grâce à l'utilisation généralisée du e-learning qui s'appuie fortement sur Internet pour faciliter l'accès à diverses ressources et services, ainsi que les échanges et la coopération à distance.

Cela a conduit à la création de nombreuses plateformes de formation à distance, telle que Moodle que nous avons retenue dans le cadre de ce PFE car il s'agit d'une plateforme de e-learning open source largement utilisée dans les universités.

Moodle propose des activités d'apprentissage collaboratives où les étudiants en interagissant construisent ensemble leurs savoir, le forum en est une. C'est un lieu pour présenter des idées et en discuter avec tous les participants au cours.

Cela se traduit par beaucoup de données dont l'évaluation par l'enseignant nécessite beaucoup d'efforts et de temps. Afin d'y remédier, des techniques d'EDM et de Text Mining ont été proposées par des chercheurs.

Dans ce chapitre nous allons aborder les notions de base, à savoir, le e-learning , les plateformes de formation dans l'enseignement supérieur, plus expressément Moodle, l'activité d'apprentissage de type forum. Par la suite nous présenterons les notions d'EDM et de Text Mining ainsi que les étapes permettant de les mettre en œuvre.

I. Notion de base

Dans cette partie nous aborderons le e-learning dans l'enseignement supérieur, les Plateformes de formation à distance, Moodle et l'activité Forum.

1.1 E-learning dans l'enseignement supérieur

Le mot e-learning est apparu au début des années 2000 dans la formation professionnelle de *Betty Roberts*, où la formation en ligne et est l'un des domaines de recherche les plus prometteurs [01].

Selon la définition de la Commission européenne, e-learning est : « *l'utilisation des nouvelles technologies multimédias et d'Internet afin d'améliorer la qualité de l'apprentissage en facilitant l'accès aux ressources et aux services, ainsi que les échanges et la coopération à distance* ».

Auparavant, le e-learning était mal accepté du fait qu'on lui reprochait le manque de l'élément humain nécessaire à l'apprentissage. Cependant, avec l'avancement rapide de la technologie et l'avancement des systèmes d'apprentissage, les Smartphones, tablettes, etc., ont acquis une place importante dans le processus d'enseignement et d'apprentissage. Cela a

conduit à son acceptation par les masses, et en conséquence des plates-formes d'apprentissage à distance ont été créées pour aider l'éducation.

1.2 Plateformes de formation à distance

Les plateformes d'enseignement à distance offrent aux étudiants un moyen facile d'accéder aux ressources pédagogiques et de les utiliser via Internet, tout en facilitant la gestion du cours pour les enseignants et les formateurs. Elles aident les étudiants, les enseignants et l'administration dans le processus d'enseignement-apprentissage. La plupart des plateformes d'enseignement à distance sont gratuites et beaucoup sont disponibles en plusieurs langues.

Ces plateformes proposent des solutions de cours en distanciel. Ce sont des outils qui favorisent la cohésion de la classe en créant un moment dans la journée où tous les étudiants peuvent se retrouver et échanger. De plus, ils permettent aux étudiants de ne pas rester isolés, de maintenir une dynamique de groupe et d'entretenir le lien qu'ils ont avec leurs enseignants.

Ce type de formation permet aux apprenants de pouvoir se former à distance, quand ils le souhaitent et d'avoir accès à différents contenus pédagogiques 24h/24, 7J/7 et peu importe le lieu où ils se trouvent et ce à leur rythme.

1.3 Moodle

Le terme « Moodle » était à l'origine un acronyme pour "*Modular Object- Oriented Dynamic Learning Environnement*". Toute personne utilisant Moodle est un "Moodleur".

Moodle est une plate-forme e-learning Open Source, suivant la licence GPL (*General Public Licence*), existant en plus de 60 langues et largement utilisée dans les Hautes Ecoles européennes [02]**.**

Cette plateforme de formation a été développée en Australie. Elle résulte de l'effort de développement *Martin Dougiamas*, un ancien de la plateforme WebCT, qui insatisfait par sa structure de fonctionnement a décidé de produire une plateforme reproduisant les fonctionnalités de WebCT tout en les améliorant. Le développement en code source libre permettait, par ailleurs, une plus grande flexibilité d'adaptation et la possibilité de collaboration. Moodle a été un succès puisque la communauté des utilisateurs est devenue très importante [03].

Cette plateforme permet la mise en place de cours en ligne et de sites web. C'est un projet bénéficiant d'un développement actif et conçu pour favoriser un cadre de formation socio constructiviste [04]. Moodle présente de nombreuses caractéristiques : multilinguisme, forums, gestionnaire de ressources, tests et plusieurs modules (activités d'apprentissage) prêtes à l'emploi (Devoirs, Chat, Sondage, Glossaires, Journal, Etiquettes, Leçons, Wiki).

Des filtres permettent également d'utiliser facilement des fichiers multimédias ou des expressions mathématiques au sein de ses pages.

Elle permet également de créer, par l'intermédiaire du réseau (web), des interactions entre des enseignants, des apprenants, et des ressources pédagogiques.Moodle est une plate-forme gratuite, modifiable, stable et robuste mais simple d'utilisation pour un professeur novice en e-learning.

Nous nous intéressons dans ce PFE aux activités d'apprentissage de type Forum faisant partie d'un Cours Moodle.

La figure 01 ci-dessous est donnée pour une meilleure compréhension d'un Cours au seins de la plateforme de formation Moodle.

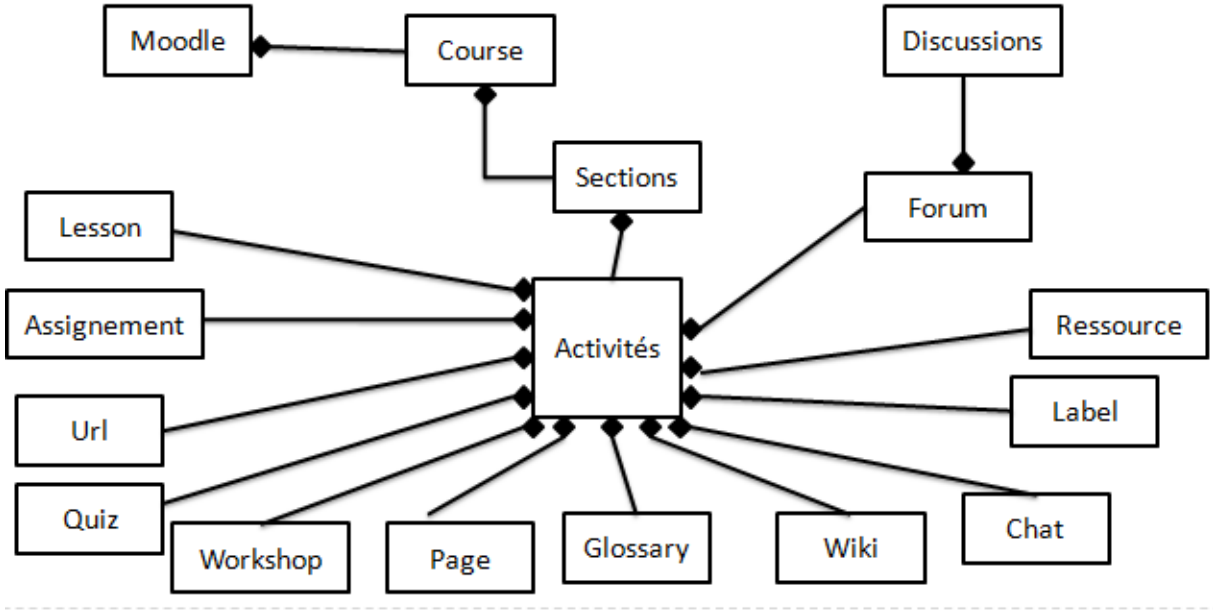

Figure 01: Modèle de d'un Cours au sein de Moodle.

1.4 Forum de discussion dans Moodle

Le forum est l'un des modules les plus importants de Moodle, il permet aux étudiants et enseignants d'échanger des opinions et des commentaires sur différents sujets reliés au cours.

Les forums peuvent être structurés de différentes manières. Ils peuvent permettre l'évaluation des messages par les pairs. Un même forum peut contenir plusieurs sujets de discussion et il est même possible de joindre des fichiers aux messages publiés dans un forum.

Généralement, les messages d'un forum sont affichés dans le forum et sont envoyées par courriel aux utilisateurs abonnés au forum.

La figure 02 ci-dessous est donnée pour une meilleure compréhension de l'activité Forum.

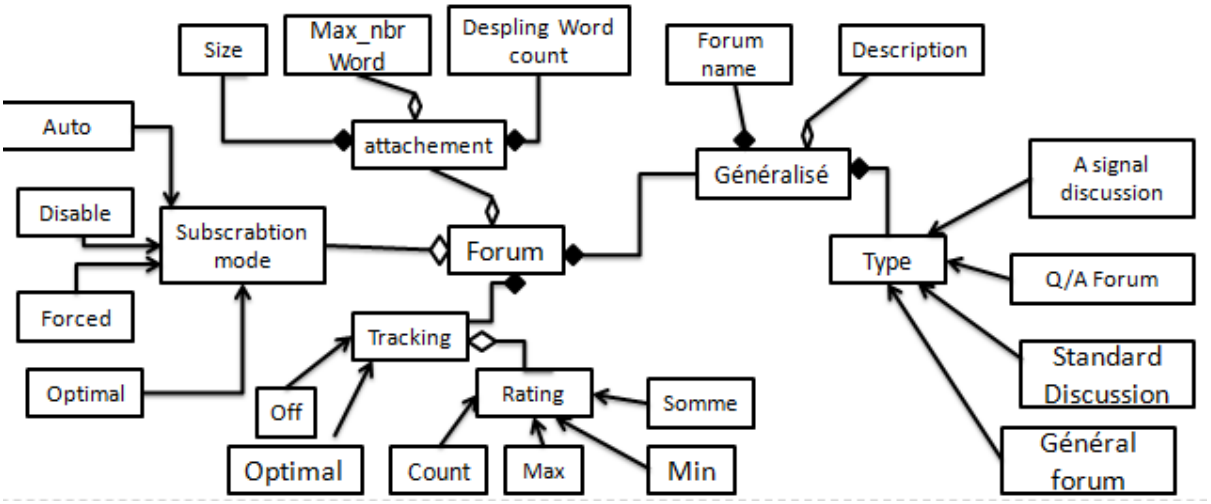

Figure 2 :Modèle de l'activité Forum au sein de Moodle.

II. Data Mining vs Text Mining:

Dans cette partie, nous passons en revue le Data Mining et ses étapes, puis le Text Mining et ses étapes.

Introduction

L'analyse d'une grande quantité de données est difficile et demande beaucoup de temps et d'efforts, il est donc très important de comprendre ces données. Comme l'a dit *Pyatsky Shapiro*, "*Tant que le monde continuera à produire des données de toutes sortes à un rythme toujours croissant, la demande d'exploration de données continuera de croître*". Par conséquent, la recherche de données est devenue une nécessité. Les universités évoluent dans un environnement complexe et avec un développement technologique rapide et un équipement informatique bon marché, cela a conduit à l'accumulation rapide de la quantité de données stockées dans les bases de données éducatives. Les outils, méthodes et techniques d'exploration de données nous permettent d'analyser ces données et de trouver des modèles et des informations cachées. Des techniques de Text Mining ont également été utilisées pour faciliter l'extraction de données en cas de problèmes avec les textes.

2.1 Data science

La *data science* combine plusieurs domaines, dont les statistiques, les méthodes scientifiques, l'Intelligence Artificielle (IA) et l'analyse des données, pour extraire de la valeur des données. Les personnes qui pratiquent la science des données sont appelées « *data scientists* » et combinent un éventail de compétences pour analyser les données collectées sur le web, les Smartphones, les capteurs et d'autres sources afin d'en tirer des informations exploitables.

La data science englobe la préparation des données pour l'analyse, y compris le nettoyage, l'agrégation et la manipulation des données pour effectuer une analyse avancée des

données. Les applications analytiques et les data scientistes peuvent ensuite examiner les résultats pour découvrir des modèles et permettre aux chefs d'entreprise de tirer des conclusions éclairées [06].

2.2 Data Mining

Dans cette section nous abordons le concept Data Mining en citant quelques définitions, sur quels types de données s'applique-t-il, la démarche de Data Mining et l'apprentissage automatique.

2.2.1 Définitions

Selon le Groupe Gartner, le Data Mining appelé aussi fouille de données est le processus de découverte de nouvelles corrélations, modèles et tendances en analysant une grande quantité de données, en utilisant les technologies de reconnaissance des formes ainsi que d'autres techniques statistiques et mathématiques[07].

Ils existent d'autres définitions, nous en citons deux :

Le Data Mining est l'analyse de grands ensembles de données observationnelles pour découvrir des nouvelles relations entre elles et de les reformuler afin de les rendre plus utilisables de la part de ses propriétaires [08].

Le Data Mining, traduit en Français par la Fouille de Données est l'ensemble des méthodes et techniques destinées à l'exploration et l'analyse de grandes bases de données informatiques, de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles des associations, des tendances inconnues ou cachées, des structures particulières restituant l'essentiel de l'information utile tout en réduisant la quantité de données permettant d'étayer les prises de décision [09].

2.2.2 Data Mining sur quels types de données ?

Le Data Mining n'est pas spécifique à un type de médias ou de données. Il est applicable à n'importe quel type d'information. Le Data Mining est utilisé et étudié pour les Bases de Données incluant les Bases de Données relationnelles et les Bases de Donnée Orientées-Objets, les entrepôts de données, en anglais *data Warehouse*, les Bases de Données transactionnelles, les supports de données non structurés et Semi-structurés comme le World Wide Web, les Bases de Données Avancés comme les Bases de Données spatiales, les Bases de Données multimédia, les Bases de données de séries temporelles et les Bases de Données textuelles et même fichiers plats.

2.2.3 Démarche de fouille de données :

Le processus d'exploration de données est divisé en deux parties, à savoir :

Le prétraitement des données et l'exploration de données.

Le prétraitement des données implique le nettoyage des données, l'intégration des données, la réduction des données et la transformation des données.

L'exploration de données effectue le traitement proprement dit des données, l'évaluation des modèles et la représentation des connaissances des donnée tel qu'illustré par la figure03.Cette dernière représente les étapes de processus de Data Mining, de la collecte de données à la génération de connaissances.

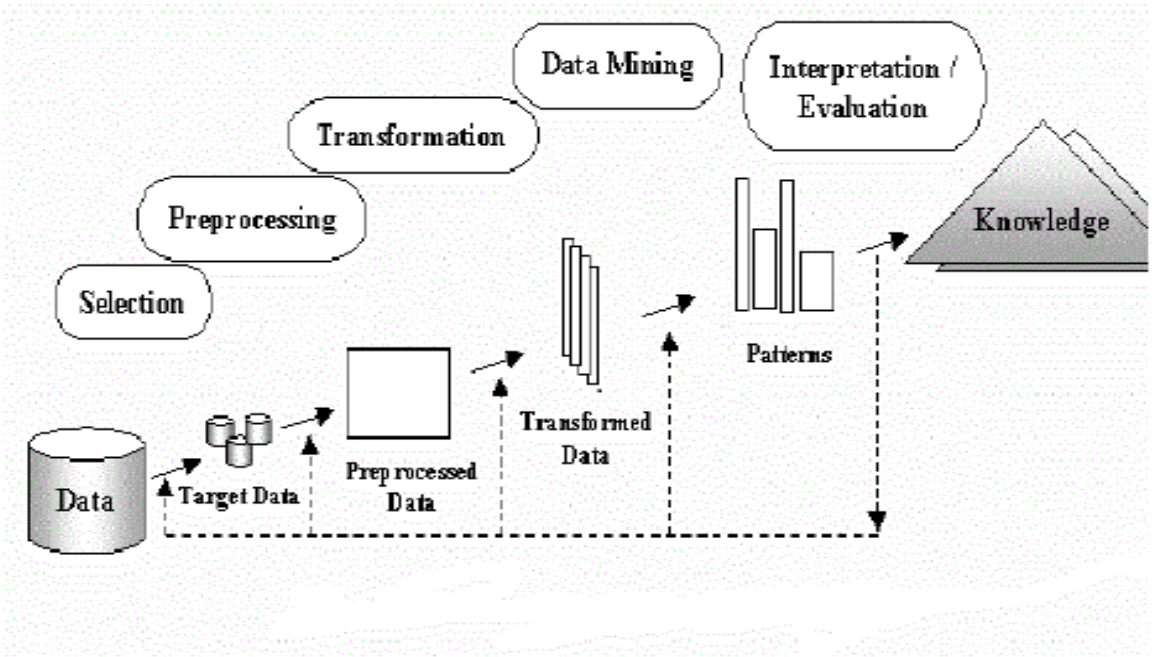

Figure 03 : Le processus de Data Mining [10]

De nombreux facteurs déterminent l'utilité des données, tels que l'exactitude, l'exhaustivité, la cohérence et l'actualité.

Les données doivent être de qualité pour qu'elles répondent à l'objectif visé. Ainsi, le prétraitement est crucial dans le processus d'exploration de données. Les principales étapes du prétraitement des données sont expliquées ci-dessous :

a) Collecte des données «*Selection*»

La combinaison de plusieurs sources de données, souvent hétérogènes, dans une base de données [11] [12].

b) Nettoyage des données « *Data Cleaning* »:

Les équipes de nettoyage des données doivent d'abord nettoyer toutes les données de processus afin qu'elles soient alignées sur les normes de l'industrie. Les données sales ou incomplètes entrainent de mauvaises informations et des défaillances système qui coutent du temps et de l'argent. Toutes les données impures seront supprimées des données acquises par l'organisation. Différentes méthodes de prétraitement et de nettoyage des données sont utilisées en fonction des ressources disponibles. Par exemple, des valeurs manquantes sont remplies manuellement ou bien la moyenne des autres données est utilisée pour remplir une valeur probable. Les équipes utiliseront également des méthodes de *Binning* pour supprimer les données bruyantes, identifier les valeurs aberrantes et résoudre les incohérences [13].

c) L'intégration

Lorsque les Data Miner combinent différents ensembles de données et sources pour effectuer des analyses, ils l'appellent intégration de données. Il s'agit de l'une des meilleures techniques d'extraction pour rationaliser l'ensemble du processus d'extraction, de transformation et de chargement.

De nombreux spécialistes effectuent un nettoyage supplémentaire des données dans différentes bases de données au cours de cette étape. Cela élimine davantage les informations incohérentes et garantit la qualité des données afin de répondre aux exigences de l'entreprise. Les spécialistes utiliseront des outils d'exploration de données tels que Microsoft SQL pour intégrer les données [13].

d) Réduction

Ce processus extrait l'information pertinente pour l'analyse des données et l'évaluation des tendances. Une petite taille des données est prise tout en conservant leur intégrité pendant la réduction des données. Les équipes peuvent utiliser des réseaux neuronaux ou d'autres formes d'apprentissage automatique au cours de ce processus minier. Les stratégies peuvent inclure les réductions dimensionnelles, les réductions de la numérosité ou la compression des données. En termes de réductions dimensionnelle, les ingénieurs réduisent la quantité d'attributs dans les données d'analyse. Dans les réductions de la numérosité, les équipes remplacent la quantité initiale de données par une plus petite quantité de données. Dans la compression des données, les ingénieurs fournissent une généralisation compressée des données collectées [13].

e) Transformation

Dans ce processus standard, les ingénieurs transforment les données en une forme acceptable pour les harmoniser avec les objectifs d'exploitation minière. Ils consolident les données de préparation pour optimiser les processus d'exploration de données et faciliter le discernement des modèles dans l'ensemble de données final.
La transformation des données englobe la cartographie des données et d'autres techniques de science des données. Les stratégies comprennent le lissage ou l'élimination du bruit des

données. D'autres techniques populaires incluent l'agrégation, la normalisation ou la discrétisation [13].

f) Exploration « *Data Mining* »

L'application de quelques algorithmes du Data Mining sur les données produites par l'étape précédente (*Knowledge Discovery in Data bases*, ou KDD) [12] [14].

g) Évaluation

C'est l'étape où on apporte un aperçu du monde réel. Les spécialistes identifient tous les modèles utiles qui peuvent générer des connaissances commerciales. Ils utiliseront leurs modèles, leurs données historiques et leurs informations en temps réel pour en savoir plus sur les clients, les employés et les ventes. Les équipes résument également les données d'information ou utilisent des techniques d'exploration de données de visualisation pour les rendre plus faciles à comprendre [13].

h) Représentation :

La représentation des connaissances est une étape où des outils de visualisation des données et de représentation des connaissances sont utilisés pour représenter les données extraites. Les données sont visualisées sous forme de rapports, de tableaux, etc.

2.2.4 Fouille de données et apprentissage automatique :

L'apprentissage automatique est une tentative de comprendre et reproduire cette faculté d'apprentissage dans des systèmes artificiels. Il s'agit, très schématiquement, de concevoir des algorithmes capables, à partir d'un nombre important d'exemples (les données correspondant à l'expérience passée), d'en assimiler la nature afin de pouvoir appliquer ce qu'ils ont ainsi appris aux cas futurs [15].

La figure 04 représente le schéma des types de l'apprentissage automatique.

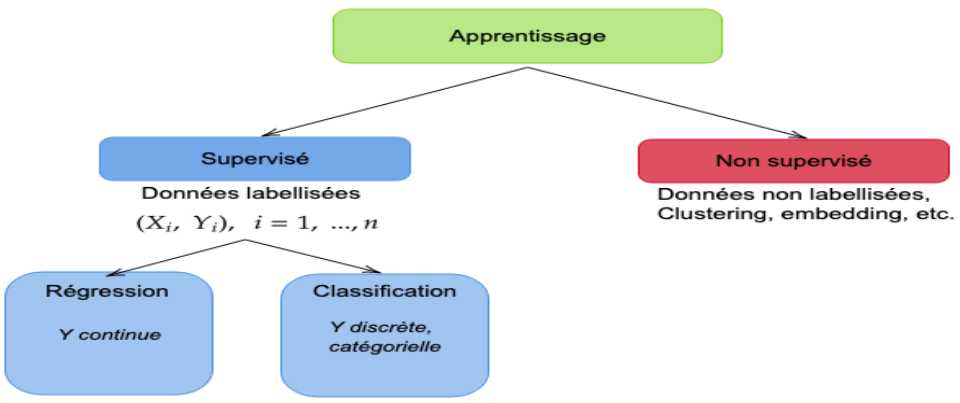

Figure 04: Les deux types de l'apprentissage

Il existe deux types de L'apprentissage automatique :

Apprentissage supervisé et Apprentissage non-supervisé.

- **Apprentissage Supervisé**

Si les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classement ; on parle alors d'apprentissage supervisé (ou d'analyse discriminante) :

1. **Classification**

Les méthodes de classification s'appliquent lorsque l'ensemble des valeurs résultats est discret. Ceci revient à attribuer une classe (aussi appelée étiquette ou label) pour chaque valeur d'entrée. Les techniques de classification peuvent être basées sur des hypothèses probabilistes (exemple, naïf bayésien), des notions de proximité (exemple, k plus proches voisins) ou des recherches dans des espaces d'hypothèses (exemple, arbres de décisions) [16].

Le choix de la technique convenable est important ; il faut pouvoir choisir la méthode la plus adapté qui sera capable de séparer au mieux les données d'apprentissage.

2. **Régression**

Les méthodes de régression s'appliquent lorsque le résultat que l'on cherche à estimer est une valeur continue. En ML, la régression est un outil important de l'apprentissage supervisé pour la modélisation et l'analyse des données. Elle est notamment utilisée en statistique et en économie [17]. Il y a deux types de régression :

Régression linéaire et **Régression logistique**

- **Apprentissage non supervisé :**

Dans l'apprentissage non supervisé il n'y a pas de valeurs de sortie, il s'agit de trouver des structures cachées à partir d'un ensemble de données qui doivent être regroupé d'où le terme « clustering ». Le but de ce type d'apprentissage est de séparer les données en groupes ou en catégories

.

- **Clustering** :

Clustering est une technique d'apprentissage automatique non supervisé, utilisé pour le regroupement des données non étiquetées dans de nombreux domaines. Si on dispose d'un nombre fini de points de données et on cherche à les classer dans des groupe de sorte que chaque groupe contient des points de données ayant des propriétés et/ou caractéristiques similaires. Le problème principal qui se pose dans ces algorithmes c'est le choix des propriétés à prendre en compte au cours du regroupement [18].

III Text Mining

Dans cette section nous abordons la notion de Text Mining.

3.1 Définitions

Le Text Mining communément appelé Text Data Mining [21] ou découverte de connaissances à partir de données textuelles fait généralement référence au processus d'extraction de modèles d'intérêt ou de connaissances auparavant inconnues [20]. Le Text Mining fait partie intégrante de la science agrégée en science des données, donc en intelligence artificielle.

C'est un ensemble de méthodes d'analyse linguistique, de techniques et d'outils utilisés pour analyser et traiter des données textuelles. Qui sont pour la plupart des données non structurées et non référencées dans une base de données. Les machines ne peuvent donc pas interpréter ces données. Il existe différents types de données textuelles : textes dactylographiés, mots, e-mails, PowerPoint, etc.

C'est une extension pour extraire et explorer des données ou des connaissances à partir de bases de données structurées [19] [22].

3.2 Présentation du texte :

La représentation de texte est également une étape importante dans le processus de classification de texte car les algorithmes d'apprentissage ne sont pas capables de traiter directement le texte, plus précisément les données non structurées telles que les images, les sons, les textes et les vidéos.

Ces termes sont les unités minimales qui composent un texte, et peuvent être plus ou moins complexes : chaînes de caractères, mots, racines de mots, groupes de mots (concept) ou expression.

Cette étape consiste généralement à représenter chaque document par un vecteur, où les termes donnés dans le texte par exemple sont des composantes de ce vecteur, à ces termes on associe des poids pour rendre chaque vecteur exploitable par des algorithmes d'apprentissage, et enfin réduire la dimensionnalité.

✓ Représentation en « sac de mots » « bag of words»

Le modèle Bag Of Words (BOW) est une représentation simplificatrice utilisée dans le traitement du langage naturel et la recherche d'informations (RI). Dans ce modèle, un texte est représenté comme une collection non ordonnée de ses mots, sans tenir compte de la grammaire et même de l'ordre des mots.

En cas de classification de texte, un mot dans un document se voit attribuer un poids en fonction de sa fréquence dans le document et de sa fréquence entre les différents documents. Les mots avec leurs poids forment BOW. Ce travail introduit une nouvelle fonctionnalité au modèle BOW. Récemment, différents domaines se sont montrés très intéressants pour tirer parti de cette technique BOW vers une analyse de texte efficace.

Le modèle BOW est couramment utilisé dans les méthodes de classification de documents, où la (fréquence d'occurrence de chaque mot est utilisée comme caractéristique pour l'apprentissage d'un classificateur [23] [24] [25].

Cette représentation des textes exclut toute analyse grammaticale et toute idée de l'espace entre les mots : c'est pourquoi cette représentation est appelée le "sac de mots"

Ces descripteurs ont un réel privilège d'avoir un sens explicite, cependant la représentation "sac de mots" présente plusieurs anomalies :

Le problème des mots composés tels que : Arc-en-ciel, peut-être et problème Abréviations telles que : TIC, TAL et TM.

Présence parmi les descripteurs de mots outils qui constituent une grande partie des mots du texte, mais qui contiennent peu d'informations utiles pour la classification du texte [26].

Distinguer des mots d'une même famille à cause de leur différence morphologique (ex : une fois, fois, femme, quoi) est généralement un handicap, car chaque mot a une fréquence très basse alors que les regrouper permettra d'avoir des fréquences importantes et de réduire le phénomène d'imprécision de fréquence. [26]

La représentation du sac de mots est un regroupement lâche de tous les mots du document "Sac de mots" sans tenir compte des groupes et de l'ordre des mots dans la phrase qui entraîne une perte de sémantique du texte. [26]

✓ Représentation des textes par des phrases

Un certain nombre de chercheurs suggèrent d'utiliser des phrases comme unité de représentation au lieu de Mots Malgré la simplicité d'utiliser les mots comme unité de représentation.

Les phrases ont l'avantage de retenir des informations sur la position du mot dans la phrase. Il a un degré d'ambiguïté inférieur à celui de ses mots constitutifs, et a également l'avantage de conserver des informations sur la position du mot dans la phrase. Les phrases sont plus utiles que les mots seuls, donc les phrases sont plus utiles que les mots seuls.

Étant donné que l'utilisation de « sac de phrases » pose un problème de taille (pour le nombre de mots, il existe probablement n combinaisons de longueur k).

Pour y remédier, nous ne considérons pas toutes les séquences possibles mais essayons de sélectionner des phrases sélectionnées, avec une préférence pour les phrases riches en sens dans la phrase.

- ✓ **Représentation des textes avec des racines lexicales(*Stemming*)**

Dans la description du modèle précédent, chaque flexion d'un mot est considérée comme un descripteur différent; en particulier, les différentes formes un verbe sont autant de mots.

Par exemple, les mots (Entre),(entrent),(entré) sont considérés comme des descripteurs différents alors qu'il s'agit de trois formes conjuguées du même verbe (Entrer) Pour remédier à ce problème, il faut de considérer uniquement la racine des mots plutôt que les mots entiers (on parle de stem en anglais).

- ✓ **Représentation des textes avec des lemmes (lemmatisation) :**

La lemmatisation consiste à utiliser l'analyse grammaticale afin de remplacer les verbes par leur forme infinitive et les noms par leur forme au singulier.

La lemmatisation est donc plus compliquée à mettre en œuvre que la recherche de racines puisqu'elle nécessite une analyse grammaticale des textes. L'objectif de la lemmatisation est d'associer à chaque mot, une entrée dans le lexique.

D'une manière générale, on définit un lexique comme un ensemble de lemmes, ce qui correspond plus ou moins à un dictionnaire. Par exemple le lemme de "mangeaient" est "manger". Depuis la fin des années 80, les lemmatiseurs sont capables d'associer à chaque mot d'un texte son lemme grâce à un étiquetage morpho syntaxique (nom, verbe, adjectif, etc.)[27].

- ✓ **Représentation par n-grammes :**

La notion de n-grammes et plus particulièrement bi-grammes et trigrammes (c'est -à-dire avec respectivement n=2 et n=3). La notion de n-grammes introduit par (Shannon, 1948) dans le cadre de systèmes de prédiction de caractères en fonction des autres caractères précédemment entrés. La notion n-gramme de X définit comme une séquence de n X consécutifs. X peut alors être un caractère ou bien un mot.

La construction de n-grammes de caractères et de mots par la notion de déplacement de fenêtre. Ce déplacement se fait par étape, une étape étant soit un caractère ou bien un mot. Les caractères (ou mots) contenus dans la fenêtre ainsi définie constituent les descripteurs d'un corpus.

3.3 Prétraitement :

La méthode de prétraitement joue un rôle très important dans les techniques et les applications de Text Mining. C'est la première étape du processus de Text Mining. La méthode de prétraitement joue un rôle très important dans les techniques et les applications de Text Mining.

C'est la première étape du processus de Text Mining. Il se compose de quatre étapes (Fig. 05).

La figure 05 représente le schéma des étapes de prétraitement de texte.

Figure 05: Text Mining Pre-processingSteps.

1) Case folding

Dans un document qui utilise des majuscules ou autres, il n'y a parfois pas de points communs, cela peut être dû à des erreurs d'écriture. Dans le prétraitement de texte, le processus de pliage de la casse vise à changer toutes les lettres d'un document texte en lettres minuscules.

2) Tokenisation

Un document texte se compose d'un ensemble de phrases, le processus de tokenisation divise le document en plusieurs parties de mots appelés jetons [28].

3) LES STOP WORDS

Les mots vides sont une division du langage naturel. Le motif pour lequel les mots vides doivent être supprimés d'un texte est qu'ils rendent le texte plus lourd et moins important pour les analystes. La suppression des mots vides réduit la dimensionnalité de l'espace des termes.

Les mots les plus courants dans les documents texte sont les articles, les prépositions et les pronoms, etc. qui ne donnent pas le sens des documents. Ces mots sont traités comme des mots vides. Exemple pour les mots vides : le, dans, un, un, avec, etc. Les mots vides sont supprimés des documents car ces mots ne sont pas mesurés en tant que mots-clés dans les applications d'exploration de texte [29].

4) Méthodes de suppression des mots vides

Quatre types de méthodes de suppression de mots vides sont suivis, les méthodes sont utilisées pour supprimer les mots vides des fichiers [29].

a) La méthode classique

La méthode classique est basée sur la suppression des mots vides obtenus à partir de listes précompilées [30].

b) Méthodes basées sur la loi de Zipf (Z-Méthode)

En plus de la liste de mots vides classique, nous utilisons trois méthodes de création de mots vides déplacées par la loi de Zipf, dont : la suppression des mots les plus fréquents (TF-High) et supprimer les mots qui apparaissent une fois, c'est-à-dire les mots singleton (TF1). Nous envisageons également de supprimer les mots à faible fréquence de document inverse (IDF) [30] [31].

C) La méthode d'information mutuelle (MI)

La méthode d'information mutuelle (MI) est une méthode supervisée qui fonctionne en calculant l'information mutuelle entre un terme donné et une classe de document (par exemple, positif, négatif), fournissant une suggestion de la quantité d'informations que le terme peut parler d'une classe donnée. Une faible information mutuelle suggère que le terme a un faible pouvoir de discrimination et par conséquent, il devrait être supprimé [32] [33].

d) Échantillonnage aléatoire basé sur le terme (TBRS) :

Cette méthode a été proposée pour la première fois par Lo et al. (2005) pour détecter manuellement les mots vides des documents Web.

Cette méthode fonctionne en itérant sur des morceaux de données séparés qui sont sélectionnés au hasard. Il classe ensuite les termes de chaque bloc en fonction de leurs valeurs de format à l'aide de la mesure de divergence de *Kullback-Leibler*, comme indiqué dans l'équation 1.

$$dx\,(t) = Px\,(t).\log 2\,(Px\,(t)/p\,(t)\quad (1)$$

Où Px (t) est la fréquence de terme normalisée de a, terme t dans une masse x, et P(t) est la fréquence de terme normalisée de t dans l'ensemble de la collection.

La liste d'arrêt finale est ensuite construite en prenant le moins de termes informatifs dans tous les morceaux et en supprimant toutes les duplications possibles [32].

5) Stemming

Cette méthode est utilisée pour identifier la racine/la racine d'un mot. Par exemple, les mots connectés, connecté, connexion, connexions peuvent tous être dérivés du mot « connecter » [34].

Le but de cette méthode est de supprimer divers suffixes, de réduire le nombre de mots, d'avoir des radicaux correspondants avec précision, de gagner du temps et de l'espace mémoire. Dans le radicalisme, la traduction des formes morphologiques d'un mot vers sa racine est effectuée en supposant que chacune est sémantiquement liée.

Deux points sont pris en compte lors de l'utilisation d'un *stemmer* :

- Les mots qui n'ont pas le même sens doivent être séparés
- Les formes morphologiques d'un mot sont supposées avoir la même signification de base et doivent donc être mappées sur le même radical.

Ces deux règles sont considérées satisfaisantes et suffisantes dans les applications de Text Mining ou de traitement du langage. Le stemming est généralement considéré comme un dispositif d'amélioration du rappel. Pour les langues à morphologie relativement simple, le pouvoir de racinisation est moindre que pour celles à morphologie plus complexe. La plupart des expériences de racinisation réalisées jusqu'à présent sont en anglais et dans d'autres langues d'Europe occidentale.

3.4 Word embedding :

a. Définitions et généralités :

Il s'agit d'un ensemble de techniques d'apprentissage automatique qui visent à représenter des mots ou des phrases de texte par des vecteurs de nombres réels. Dans cette représentation, les mots qui apparaissent dans des contextes similaires ont des vecteurs correspondants relativement proches. Cette technique est basée sur l'hypothèse qui veut que les mots apparaissant dans des contextes similaires ont des significations apparentées [35].

La technique de Word Embeding diminue la dimension de la représentation des mots en comparaison d'un modèle vectoriel en facilitant ainsi les taches d'apprentissage impliquant ces mots.

Le Word Embeding caractérise chaque mot par un ou plusieurs vecteurs denses de faible dimension ayant des éléments réels. Il existe plusieurs approches de word embeding. Les premiers remontent aux années 1960 et reposent sur des méthodes de réduction de

dimensionnalité. Plus récemment de nouvelles techniques basées sur des modèles probabilistes et des réseaux de neurones comme Word2Vec sont utilisées [35].

b. Word2Vec

Le modèle *Word2Vec* un modèle de représentation distribué des peu profond (par apport a un réseau de neurone traditionnel) ou le principe est de générer des vecteurs dimensionnels a partir d'un apprentissage sur des donnes d'entrée non étiquetées. En fait, il prédit les mots en fonction de leur contexte en utilisant l'un des deux modèles neuronaux distincts : CBOW et Skip-Gram.

➢ CBOW

Le Modèle *CBOW* [35] prédit un mot courant basé sur son contexte. Le contexte corresponde a un certain nombre de mots voisins a gauche et a droite de mot. Dans le modèle CBOW trois couches sont utilisées comme ne po vous le voir sur la figure :

- La couche d'entrée correspond au contexte.
- La couche cachée correspond a la projection de chaque mot de la couche d'entée dans la matrice de poids.
- La couche de sortie.

La dernière étape de ce modèle est les comparaisons entre le mot entrée et lui-même afin de corriger sa représentation en fonction de la propagation arrière du gradient d'erreur. Ainsi, Skip-gram cherche la prédiction d'un mot donnée au lieu de la prédiction de contexte d'un mot sachant son contexte comme CBOW.[36]

La dernière étape de ce modèle est les comparaisons entre sa sortie et chaque mot de contexte afin de corriger sa représentation en fonction de la propagation arrière du gradient d'erreur.

La figure 06 représente le schéma de modèle CBOW.

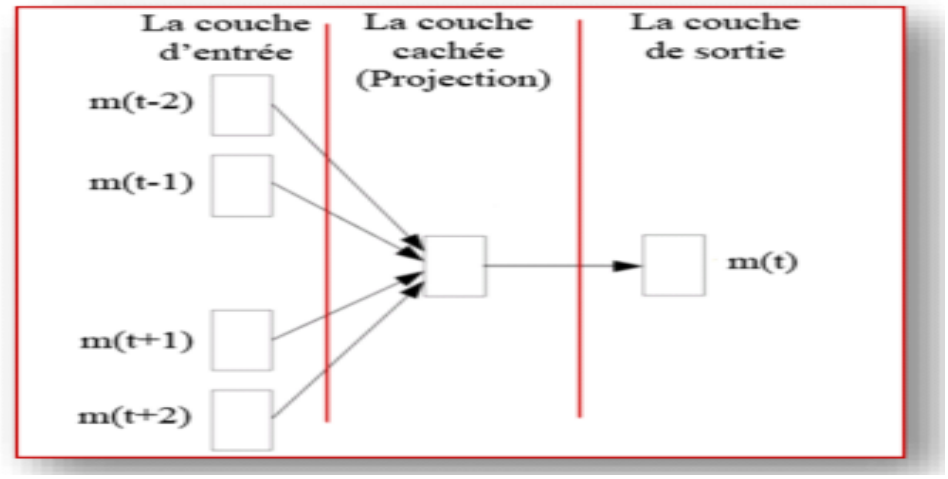

Figure 06:schéma de modèle CBOW

- **Skip Gram «SG»**

Skip_Gram est le contraire de modèle CBOW, En effet. La couche d'entrée correspond au mot cible. La couche de sortie correspond au contexte comme on peut le voir sur figure. Ainsi, Skip gram cherche la prédiction d'un mot donnée au lieu de la prédiction de contexte d'un mot sachant son contexte comme CBOW. La dernière étape de ce modèle est les comparaisons entre sa sortie et chaque mot de contexte afin de corriger sa représentation en fonction de la propagation arrière du gradient d'erreur [36].

La figure 07 représente le schéma de modèle Skip_Gram.

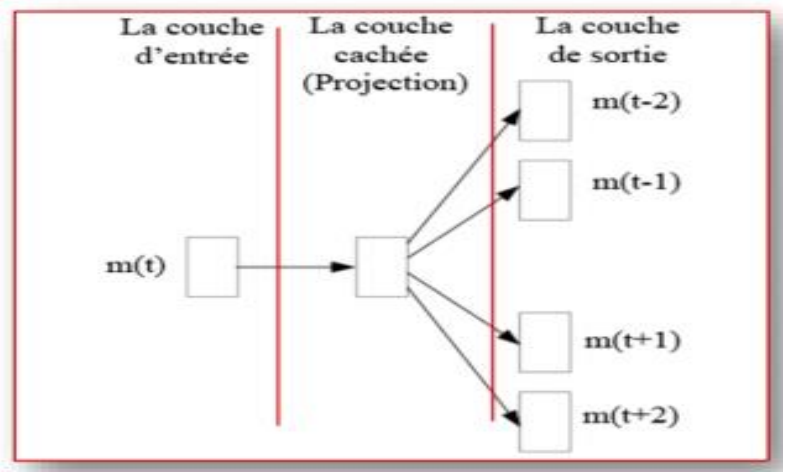

Figure 07: schéma de modèle Skip-Gram

3.5 Réduction de dimensionnalité

a) Pourquoi réduire ?

Trouver une solution pour résoudre le problème de la classification automatique, c'est aussi trouver des solutions à la difficulté du traitement automatique du langage naturel, en raison du vocabulaire large et passionnant des textes, qui peut être un obstacle à l'application d'algorithmes plus complexes, car l'utilisation ces vocabulaires directement et la création d'une fonctionnalité pour chaque mot qu'il contient conduit à une haute dimensionnalité du vecteur spatial.

Chaque texte sera représenté par un vecteur contenant de nombreux termes tels que les mots du vocabulaire. Traiter ce vecteur peut prendre beaucoup de temps de calcul et beaucoup

d'espace mémoire et peut nous empêcher d'utiliser des algorithmes de classification plus complexes et l'utilisation de tous ces mots peuvent affecter négativement l'exactitude de la classification..

En fait, beaucoup de mots n'ont pas de sens. Aussi, si un mot est présent dans plusieurs documents, il ne sera pas possible de déterminer si le texte le contenant appartient à l'une ou l'autre catégorie. Ainsi, il est nécessaire de réduire et de choisir les descripteurs les plus appropriés (ceux qui assureront les meilleures performances du classifieur), qui seront utilisés comme vecteurs d'entrée avant de pouvoir utiliser le modèle d'apprentissage. [16]

Les principaux objectifs de la réduction de dimensionnalité sont :

- Faciliter la visualisation et la compréhension des données
- Réduire l'espace de stockage requis
- Réduire l'apprentissage et l'utilisation du temps
- Déterminer les facteurs pertinents

b) Nombre de descripteurs enregistrés

C'est pourquoi nous cherchons à supprimer des termes de la représentation des textes, sachant que chaque suppression d'un terme entraîne une perte d'information ; Le bon compromis doit être trouvé, d'une part, entre la nécessité de réduire l'espace des descripteurs tout en minimisant les redondances potentielles, et d'autre part, la nécessité de conserver suffisamment d'informations.

c) Méthodes de sélection d'un descripteur

Il existe plusieurs méthodes ou techniques de sélection de termes :

Pour chaque terme qui représente sa signification pour le document dans lequel il apparaît ou pour l'ensemble du groupe, puis pour identifier les termes les plus importants, ces techniques ont été développées principalement pour réduire le vocabulaire. Il se divise en deux grandes familles ; Sélection d'attributs et extraction de caractéristiques.

1. ***Sélection de fonctionnalité*** Cette technique reprend les attributs d'origine (les mots dans notre cas) et ne garde que ceux qui sont utiles à la classification, en fonction bien entendu d'une fonction d'évaluation particulière.

1. ***Extraire les attributs*** Contrairement à la sélection, cette méthode crée de nouveaux attributs à partir des attributs de départ et effectue soit des agrégations, soit des transformations.

3.6 Schéma de pondération

a. La Pondération

La pondération des termes est une mesure statistique, le principe de pondération s'appuie sur l'observation suivante (Rij, 1979) (Sal &McG, 1983) : « *la fréquence d'apparition des mots dans les textes en langage naturel est significative de l'importance de ces mots dans le seul but de représenter le contenu de ces textes* »1.

L'intérêt de cette pondération est mieux exploiter l'information contenue dans le document pour améliorer les performances d'un système de classification de textes.

Il y a beaucoup de méthodes pour calculer le poids sachant que, pour chaque terme, il est possible de calculer non seulement sa fréquence dans le corpus, mais aussi le nombre de documents contenant ce terme.

b. Formules de pondération

Il y a trois étapes pour juger les mots

- ✓ **Termfrequency (TF)**

- le terme qui apparait plusieurs fois dans un document est plus important qu'un terme qui apparaît une seule fois

- W (j i) = Fréquence du terme t(i) dans le document d(i)

$$TF\ (i,j)= W(i,g)/|d(j) \qquad (1)$$

- ✓ **Inverse document frequency (IDF)**

- Un terme qui apparaît dans peu de documents est un meilleur discriminant qu'un terme qui apparaît dans tous les documents :

- df (i)= nombre de documents contenant le terme t(i).

- d= nombre de documents du corpus.

$$IDF = \log(d/df(j))$$

- ✓ **TF-IDF:**

-TF-IDF signifie Term Frequency x Inverse Document Frequency:

« *Mesure l'importance d'un terme dans un document relativement à l'ensemble des documents* ».

- **Vecteur TF-IDF**

L'idée de base est de représenter les documents par des vecteurs et de mesurer la proximité entre documents par l'angle entre les vecteurs, cet angle étant donc supposé représenter une distance sémantique.

Le principe est de coder chaque élément du sac de mot par un scalaire (nombre) appelé TFIDF (présenté précédemment) pour donner un aspect mathématique aux documents textes.

Cette représentation donne plus de poids aux termes qui apparaissent avec une haute fréquence dans peu de documents. L'idée est que de tels mots aident à discriminer entre textes ayant différent sujet. [37]

c. Modèles de représentation de document

Afin de rendre le document utilisable par des algorithmes d'apprentissage, le document est représenté dans une première étape par un vecteur, et ses composants, par exemple des mots contenus dans le texte. Une collection de textes peut être ainsi représentée par une matrice dont les lignes sont les termes qui apparaissent au moins une fois et les colonnes sont les documents de cette collection [38].

Il existe de nombreux modèles de représentation de texte, tels que le modèle probabiliste, le modèle séquentiel, et le modèle le plus couramment utilisé est le modèle vectoriel.

Le modèle vectoriel permet une analyse très efficace de grandes collections de documents. Il a une structure de données simple, sans utiliser aucune information sémantique explicite. Un grand nombre de chercheurs dans le domaine ont choisi d'utiliser une représentation vectorielle dans laquelle chaque texte est représenté par un vecteur de n termes pondérés.

1. Représentation binaire

C'est la représentation la plus simple et la plus ancienne, qui ne concerne que la présence ou l'absence d'un terme dans le texte, et consiste à utiliser une pondération binaire : 1 si le terme est présent une ou plusieurs fois dans le document, 0 sinon.

Comme cette méthode de représentation ne fournit pas les informations nécessaires ni sur les occurrences d'un terme dans le document, qui pourraient être des informations importantes pour le processus de classification, ni sur la longueur du texte, elle n'est pas très utile.

2. Représentation fréquentielle:

Cette représentation présenter le texte sous forme de vecteur dont les éléments renseignent non seulement sur la présence ou l'absence d'un terme comme dans un vecteur binaire mais aussi informe sur le nombre de présences du terme dans le texte (fréquence d'apparition des termes).

Conclusion

Dans cette section, notre travaille été consacré au Data Mining vs Text Mining. Concernant le Data Mining, nous avons présenté :-les différentes démarches de fouille de données, - sur quels type de données nous appliquant les techniques de Data Mining et- la relation entre Data Mining et l'apprentissage automatique.

En ce qui concerne le Text Mining, nous avons présenté :-les différentes méthodes de représentation de texte, - les méthodes de prétraitement de texte et - les méthodes les plus utilisé pour pondérer les termes. Nous avons également abordé la technique de Word Embeding et les principales méthodes de réduction de la dimension.

Comme nous l'avons déjà souligné, l'étape de représentation de texte est très importante pour la catégorisation de texte ainsi que pour avoir de bons résultats.

Chapitre 02 : L'état de l'art

Introduction

Dans ce chapitre nous allons étudier quelques travaux connexes au thème de notre PFE, à savoir : Exploiter les messages postés dans les forums de discussion au sein de la plateforme Moodle dans le but d'aider l'enseignant et l'apprenant dans le processus d'enseignement-apprentissage. Pour ce faire, nous avons retenu trois travaux de la littérature. Le premier travail, celui de « **Dringus& Ellis** »dont l'intérêt est de montrer qu'une simple fouille de données peut produire des informations utiles pour l'enseignant. Le deuxième travail de « **Azevedo- Reategui and Behar »** aborde le calcul de l'importance objective des messages des étudiants par rapport au sujet évoqué par l'enseignant, le dernier travail. Quant à celui de « **Cristóbal Romero_ Manuel-Ignacio Lopez_ Jose-María Luna and Sebastián Ventura** », il traite de la prédiction précoce concernant la réussite ou l'échec de l'étudiant. Tous ces éléments contribuent à aider l'enseignant à évaluer l'élève.

I. Travaux de *Dringus& Ellis*

Cet article écrit par les auteurs en**2005**. Dans cet article, les auteurs discutent de la manière dont le processus manuel d'évaluation des forums de discussion et les fils de discussion peut être simplifié en intégrant des concepts d'exploration de données avec des critères d'évaluation et une sélection d'indicateurs. L'intérêt de ces travaux est de montrer que des opérations fouille de données simples peuvent produire des informations utiles pour l'enseignant telles que des données concernant : le temps, la vitesse et les séquences d'échange des contributions. L'un des objectifs des auteurs est d'identifier des indicateurs de coparticipation à utiliser pour évaluer les progrès et les performances des étudiants dans les discussions en ligne.

Les auteurs ont choisi des indicateurs provisoires, tels que le temps, la vitesse et la séquence, pour fournir une démonstration directe des concepts du Data Mining.

Dans cet article, les auteurs utilisent le terme « indicateurs de participation » pour décrire des « données » qui peuvent être utilisées pour représenter les différentes manières dont les forums de discussion interdépendants peuvent être évalués quantitativement et qualitativement.

Les auteurs ont tenté de répertorier certaines des structures de participation et des indicateurs communs qui ont été identifiés dans la littérature, pour être utilisés comme exemples d'exploration de données soit manuellement par l'enseignant, soit à travers l'ensemble de données des forums dont nous mentionnons :

- Niveau de discussion - élevé, progressif, faible (Jarvilla & Hakkinen, (2003).
- Type de message: messages, contributions, réponses et publications publiques.
- Degré d'interaction (Wentling& Johnson, 1999).

• Contributions précoces, intermédiaires, tardives ou de dernier moment.

• Aptitudes à la pensée critique - Créativité, résolution de problèmes, intuition et perspicacité.

• Enquête pratique - initiation à l'événement, exploration, intégration, décision.

• Détermination par l'enseignant de l'intervalle de temps direct par rapport à la période de continuum latent le temps.

• Pour les réponses, le délai entre la publication initiale et la réponse.

• Mots-clés significatifs et pertinents, expressions utilisées pour rester sur le sujet.

• Transitions/discussions en cours L'intervalle de temps « d'attente ».

• Cote de facilitation - hautement facilitant, informatif, utile, non facilitant.

• Temps entre les rendez-vous.

La discussion sur les indications de participation a révélé la nécessité d'identifier des indications clés bien définies qui sont utiles et extractibles dans l'évaluation des forums. Sans quoi les enseignants ne disposent pas d'un terrain d'entente ou d'un ensemble standard de critères permettant de mener une évaluation de manière cohérente. Les auteurs ont également souligné qu'une stratégie d'exploration de données peut être efficace, avec différents indicateurs de participation, et avec n'importe quelle rubrique ou modèle d'évaluation.

Pour ce faire ils ont utilisé un Toolkit, programmé pour soutenir le modèle ou le modèle d'évaluation.

II. Travaux de Azevedo- Reategui and Behar

Ce travail a été fait par " ***Azevedo- Reategui and Behar***" en **2011** alors qu'il travaille sur le forum où il a pris les messages écrits par les étudiants dans le forum et a utilisé du Text Mining sur ces messages dans le but d'arriver à une conclusion, à savoir, si ces messages étaient pertinents ou non.

L'importance de l'objectivité signifie « si la publication écrite par l'étudiant est pertinente ou non ».

L'importance objective est calculée en comparant le graphique extrait de la thèse avec le graphique du sujet traité, et la comparaison se fait en connaissant le nombre, la fréquence et la répartition des mots dans la publication, s'ils sont présents dans le sujet de discussion. Plus le nombre de mots est élevé, plus la pertinence thématique est élevée, ce qui signifie que la publication écrite par l'étudiant est pertinente par rapport au sujet de discussion, il a été mentionné qu'il a essayé un programme.

Au cours du processus d'analyse des messages dans un forum, ce logiciel collecte les messages de chaque auteur et calcule le nombre total de messages individuels et le nombre de messages considérés comme pertinents ou non pertinents pour le sujet.

Ce programme suit les étapes suivantes Pour calculer l'importance d'un objectif :

1 Le graphique est créé à partir du texte de référence de la discussion, où les mots vides, les prépositions et les adverbes sont supprimés, les mots les plus présents dans le texte sont sélectionnés et les mots les plus pertinents sont représentés dans l'extrait de texte.

2 Créez un graphique pour chaque message.

3 Calculer l'importance de l'objectif de la manière suivante :

Le programme analyse le graphe du texte de référence et le graphe du message :

En analysant les sommets d'équivalence des deux graphes, et par équivalence on entend les sommets, c'est-à-dire s'ils fournissent un contenu similaire, c'est-à-dire s'ils contiennent les mêmes mots, ou s'ils sont réduits à la même racine, ou s'ils ont les mêmes synonymes.

Utilisez une formule qui prend en compte trois côtés, à savoir le nombre de sommets dans les deux graphes, la distance entre eux dans le même graphe et leur poids dans le graphe correspondant.

Cela permet à l'enseignant d'orienter son soutien vers les apprenants. En analysant la pertinence thématique des textes. L'enseignant dispose d'informations qui peuvent l'aider à noter les apports des étudiants :-Dirigez son soutien vers les étudiants, - Surveiller les contributions des étudiants, - Faire des diagnostics sur les étudiants, - Vérifier les contributions textuelles qui nécessitent une intervention.

Ainsi, il est possible d'identifier les apprenants les moins contributeurs et de leur apporter ensuite plus de soutien.

Aussi, grâce à ces informations, les enseignants peuvent également motiver les étudiants qui postent des messages plus pertinents à interagir avec ceux qui écrivent moins de messages.

III. Travaux de Cristóbal Romero et collègues

Ce travail a été effectué par " **Cristóbal Romero et collègues**" en **2013.** En travaillant sur les Forum au sein de la plateforme Moodle, ces auteurs répondent à une question importante : Vous pouvez prédire le succès ou l'échec des étudiants des dés et sauter les lettres qu'il a écrites.

Pour le savoir, des indicateurs quantitatifs et qualitatifs ont été utilisés pour participer à des forums de discussion en ce qui concerne le nombre et la qualité des contributions ainsi

que sur l'exploration de données pour découvrir et construire des modèles de discussion sur des données de forum alternatives.

Les chercheurs permettent également l'utilisation de l'information quantitative et qualitative du "nombre de messages envoyés par l'étudiant avec le nombre de messages lus, le nombre de mots et le nombre total de messages que l'étudiant passe dans le Forum. « et les réseaux sociaux existants en utilisant la classification des algorithmes traditionnels d'exploration de données » avec des règles d'association.

Ce travail est divisé en 4étapes :

1) Collecte des données du Forum dans Moodle "Données quantitatives, qualitatives et sociales"
2) Données avant traitement "Sélectionner les cas, les caractéristiques et la conversion de données"
3) Utilisation des techniques de "classification et de compilation" pour explorer le texte.
4) Interprétation des modèles obtenus

Les modèles obtenus peuvent être utiles à l'enseignant parce qu'ils : - Prédisent avec succès l'échec ou la réussite des étudiants, - Aident les étudiants susceptibles d'échouer en l'accompagnant.

Le résultat a été divisé en deux sections :

Modèles blancs et Modèles noirs.

Les modèles noirs ont un haut degré de précision prédictive, mais leur interprétation des résultats varie. Les modèles blancs, tels que les arbres décisionnels basés sur des règles blanches, sont plus utiles car ils fournissent un ensemble simple de catégories de "cas"faciles à comprendre permettant de poser des questions sur les résultats obtenus.

L'une des meilleures techniques présentées dans cette étude est l'utilisation de trois algorithmes de classification (SMO, BayesNet et NaiveBayesSimple).

Ces méthodes ont fourni des résultats bons et semi-précis dans la prédiction des performances des étudiants.

Cela nous ramène à la réponse à la question la plus importante de cette étude : est-il possible de prédire les performances des étudiants ? Autre question : y a-t-il une relation directe entre le nombre de post d'étudiants sur le forum et l'amélioration de ses performances et de ses résultats ?

Oui, il est possible de prédire la performance d'un étudiant tôt, s'il échoue ou réussit. Mais cette prédiction n'est pas très précise.

Les étudiants qui ne passent pas de temps dans le forum, cela se reflète souvent négativement, comme souvent ils ne seront pas capables d'écrire des messages bons et

pertinents Pertinence par rapport au sujet, et cela diminue la quantité et la qualité des indicateurs.

Conclusion :

L'étude de ces trois travaux issus de la littérature nous ont permis de mieux comprendre les techniques de Data Mining et de Text Mining utilisées dans les forums de discussion au sein des plateformes de formation.

Nous pouvons ainsi conclure qu'il est possible de prédire tôt l'échec ou la réussite de l'étudiant. Cela, en commençant par le processus d'excavation du texte et d'extraction de messages pertinents au cours et en lui appliquant des techniques d'exploration de données. Cela dans le but d'aider l'enseignant à évaluer le niveau des élèves et à apporter son soutien à chaque étudiant que le système aura prédit en situation d'échec.

Chapitre 03 : Solution proposée

Introduction :

Dans cette partie nous présentons le travail que nous avons fait, comment nous avons procédé et les étapes que nous avons suivies. Et ce, en commençant par la collecte des messages postés par les étudiants dans les forums jusqu'aux résultats obtenus en utilisant des algorithmes de classification comme il est expliqué dans les sections suivantes.

I. Vue globale de notre solution :

Nous présentons cette vue globale à travers un processus comportant quatre étapes. La première étape (1) à considérer dans l'élaboration de notre solution consiste en la collecte de tous les messages écrits par les étudiants dans un espace 'Cours' Moodle, plus précisément dans les activités forum qu'ils soient liés au sujet ou non comme l'aurait fait l'enseignant et de les mettre un fichier à l'allure d'un texte qui deviendra par la suite notre dataset. Dans la deuxième étape (2) nous nettoyons et organisons le texte. Cela nous permet de travailler plus facilement avec. Dans la troisième étape (3) nous appliquons des algorithmes de classification pour aider la machine à apprendre, cette étape est une étape d'entrainement. Finalement, dans la quatrième étape (4) nous insérons de nouveaux messages pour vérifier si la machine est capable de les classer avec précision, c'est l'étape test. La figure 08 montre le fonctionnement et la séquence des étapes.

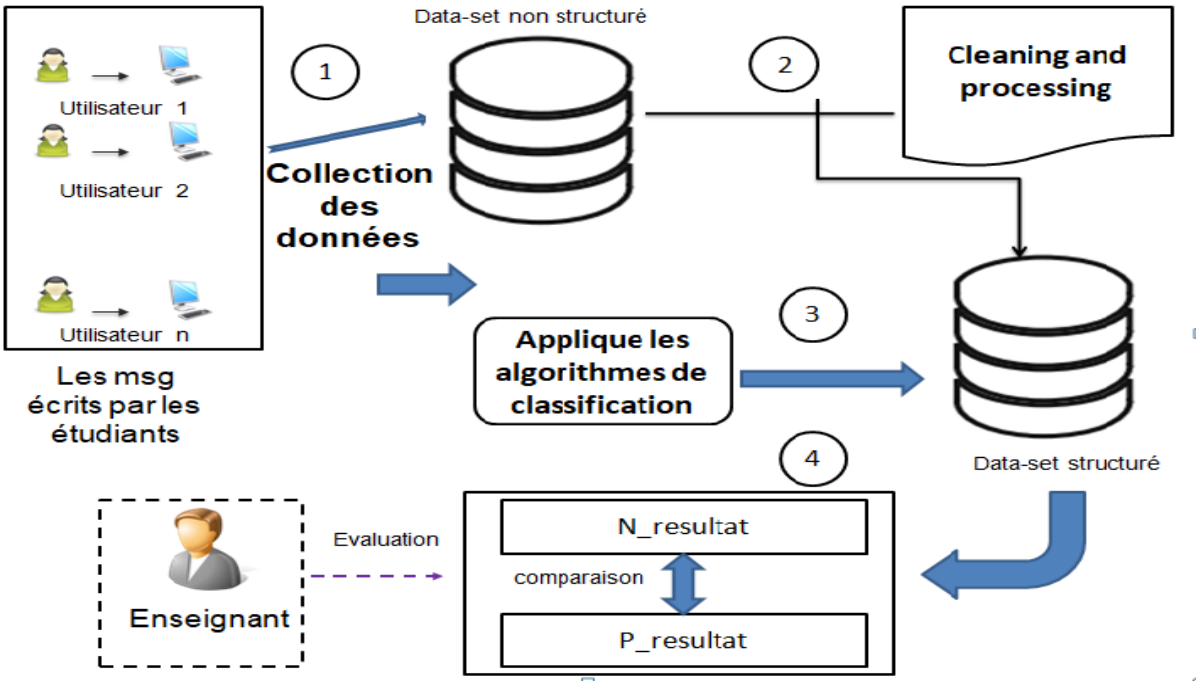

Figure 08:schéma de la vue globale

II. Formalisation de notre problème :

Comme illustré par la figure 7, notre problème se résume comme suit : Nous disposons de données que nous collectons des « messages d'étudiants » puis nous y appliquons du Text Mining dans l'espoir d'obtenir un résultat acceptable, qui est « une machine capable de classer les messages, à savoir : "Pertinents" ou "Non-pertinents". Dans notre contexte, nous entendons par Pertinence en anglais "*Relevance*" l'adéquation ou l'appropriation du message posté par rapport au Cours. En d'autres termes, le message posté est-il pertinent par rapport au cours ?

La figure 09 montre la formalisation de notre problème :Au fur et à mesure que l'étudiant saisit le message et le message passe par des étapes et à la fin nous obtenons un résultat représenté dans « le msg et pertinent ou non ».

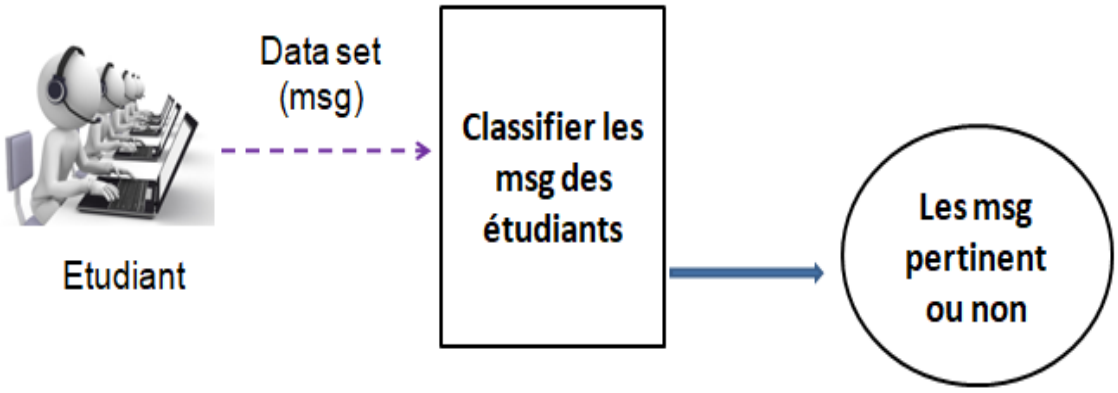

Figure 09:schéma de la notre problème

III. Les étapes de notre solution :

La solution que nous avons envisagée est constituée de quatre étapes, comme le montre la figure 10.

La figure 10 représentes les déférent étapes «collection des données, Text proccessing, apprentissage, test » pour obtenus un bonne résultat.

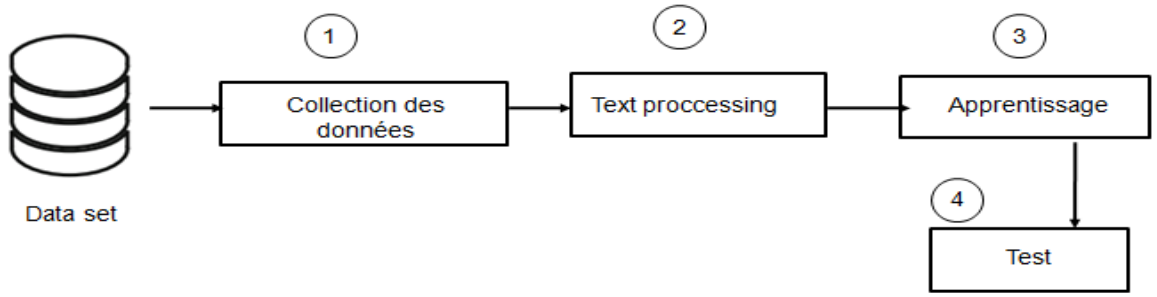

Figure 10: présentation de l'étape de notre solution

a. Collecte des messages :

Cette étape est la première étape où nous collectons les messages que les étudiants écrivent dans le forum Moodle et les mettons dans un fichier comme le montre la figure 11.

La figure 11 représente le Phase de collection des données.

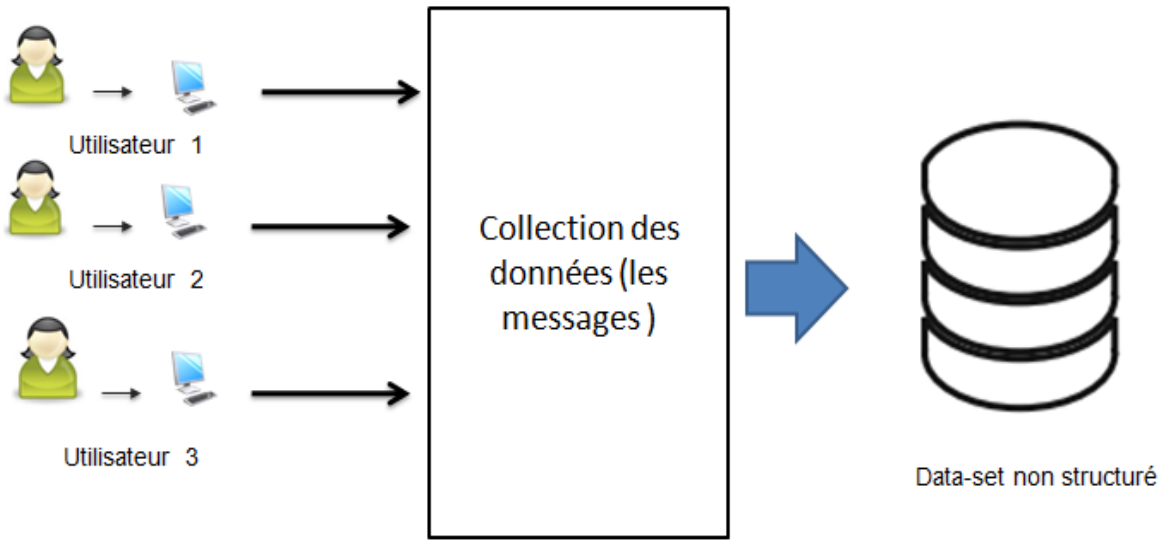

Figure 11:étape 1 collection des messages

b. Text processing

La deuxième étape est considérée comme l'étape la plus importante, car nous nettoyons le texte et l'organisons de façon qu'il soit prêt au traitement, cela nous aide dans la troisième étape de notre solution. Pour ce faire, nous appliquons des opérations au texte en nous débarrassant des mots qui ne nous profitent pas et en retournant les mots à leur origine jusqu'au dernier ... comme le montre la figure 12.

La figure 12 représente les défirent procédure pour passé a text non structuré a text structuré

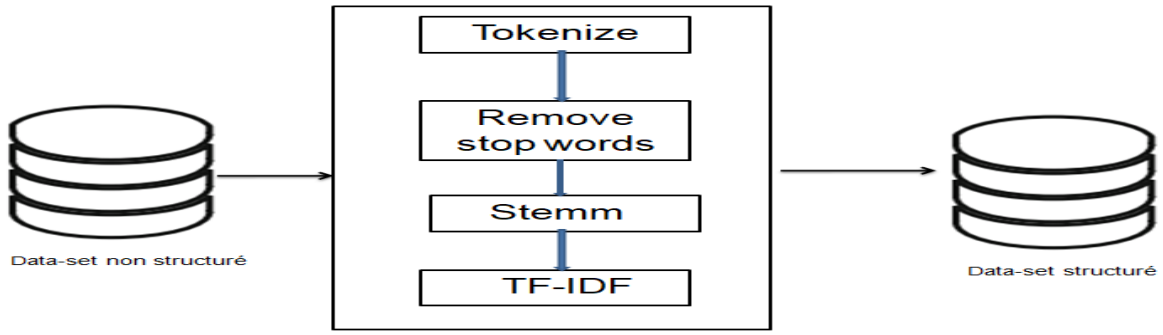

Figure 12:étape 2 Text processing

c. Classification :

Après la deuxième étape, nous passons à la troisième étape, où nous appliquons des algorithmes de classification au text structuré. Nous avons implémenté des algorithmes représentés par "" et nous surveillons les résultats de chaque algorithme jusqu'à ce que nous concluons lequel donne un bon résultat.

La figure 13 représenté le travail de classification.

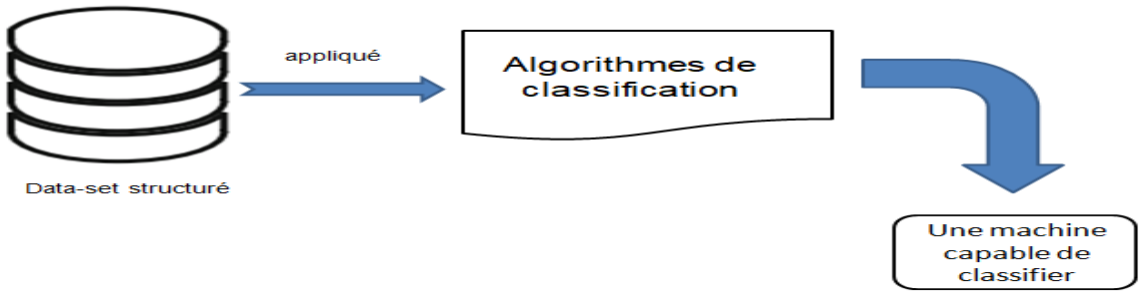

Figure 13: étape 3 classification des messages

d. Test :

Cette étape est considérée comme étant la dernière, Après avoir obtenu les résultats de la troisième étape, nous disposons d'une machine capable de classer les messages s'ils sont pertinents ou non. Nous entrons des messages dont nous savons à l'avance s'ils sont pertinents ou non. Et nous surveillons la classification de la machine, quelle est la proportion des résultats de la machine avec les résultats que nous avons précédemment, comme le montre la figure. 14

La figure 14 illustre comment nous avons fait le test et la comparaison entre le résultat précédent et le nouveau résultat.

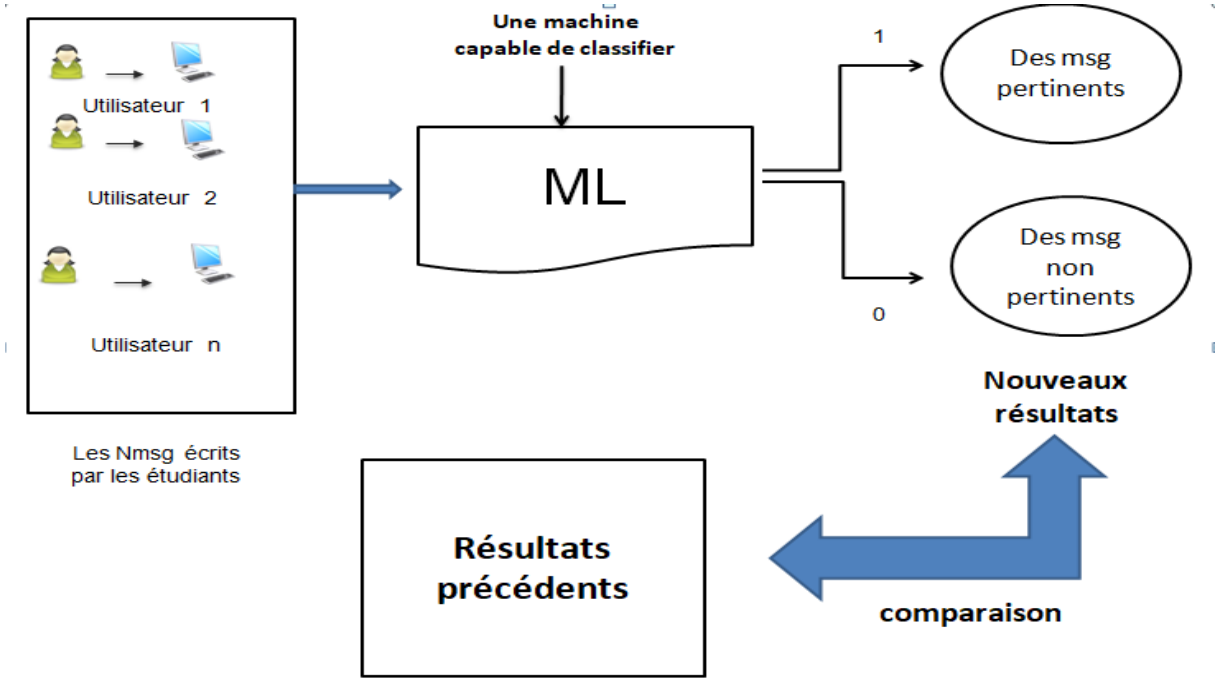

Figure 14: étape 4 Test

Conclusion :

Ce travail, nous a permis d'obtenir plusieurs résultats qui nous semblent plutôt acceptables. La deuxième étape est très importante pour l'expérimentation, car c'est la base et le pilier du Text Mining.

En effet, si nous parvenons à obtenir un texte bien organisé et exempt de défauts, la troisième étape sera très efficace et donnera des résultats assez précis. Nous avons appliqué trois algorithmes, puis nous avons retenu celui qui donne de bons résultats.

Cela nous conduit à une machine capable de déterminer si le message est pertinent ou non.

Chapitre 04 : Implémentation

Introduction :

Dans ce chapitre, nous allons présenter notre application qui a été élaborée conformément aux trois étapes suivantes :

1) Nous avons utilisé le dataset "csv_brutes.csv", qui contient les messages que les étudiants ont écrit en répondant à la question d'un enseignant.

2) Nous avons mis en place des procédures "tokenezation, stemm, lemm et remove stop_words " pour nettoyer et organiser le text obtenu dans la première étape.

3) Dans cette dernière étape, nous avons appliqué les algorithmes de classification au texte extrait de l'étape 2 et nous avons examiné les résultats produits par les algorithmes afin d'en retenir le meilleur.

I. Ressources utilisées

Dans cette section, nous allons présenter les différents outils que nous avons utilisés pour la réalisation de notre solution.

Python

Afin de développer notre application, nous avons travaillé avec Python, un langage de programmation open source parmi les langages les plus utilisés par les informaticiens. Il permet aux développeurs de se concentrer sur ce qu'ils font plutôt que sur comment ils le font, leur permettant ainsi de développer du code plus rapidement comparativement aux autres langages de programmation tel que Java.

Dans Python nous avons importé quelques librairies qui sont :

Librairies pour structurer le dataset :

- **import nltk**
- **import numpy as np**
- **import pandas as Pd**
- **from collections import Counter**
- **from nltk.tokenize import word_tokenize.**
- **From nltk.corpus import stopwords**
- **From nltk.stem import WordNetLemmatizer**
- **From nltk.stem import PorterStemmer**

Librairies pour faire le text classification :

- **from sklearn.model_selection import train_test_split**
- **from sklearn.feature_extraction.text import TfidfTransformer**
- **from sklearn.svm import LinearSVC**
- **from sklearn import metrics**
- **from sklearn.naive_bayes import MultinomialNB**
- **from sklearn.svm import SVC**

Librairies pour dessiner des graphes :

- **from wordcloud import WordCloud**
- **import matplotlib.pyplot as plt**

II. application

Ci-dessous le programme source de notre application :

La figure 15 représente le Téléchargement des bibliothèques, l'affichage du fichier «*csv_brutes.csv* » contenant notre dataset.de notre travaille et la somme des valeurs de la colonne *label.*

```
Entrée [1]: import nltk
            from nltk.collocations import *

Entrée [2]: import numpy as np
            import pandas as pd

Entrée [3]: import pandas as pd
            train = pd.read_csv('CSV_brutes.csv')
            train.head()
```

Out[3]:

	ID	POST	label
0	0	What didn't you understand in the lesson of an...	1
1	1	i understood everything except the example 01	1
2	2	he memorised the answers of the past exams	0
3	3	i hate school	0
4	4	she didn't see the lesson of the android	1

```
Entrée [4]: train['label'].value_counts()

Out[4]: 0    126
        1     82
        Name: label, dtype: int64
```


Figure 15: import libraires

La figure 16 représente un graphe de la somme des valeurs de « *label* ». La couleur bleue représente la valeur 0 (les messages Non Pertinents) et la couleur orange représente la valeur 1 (les messages Pertinents).

.

```
Entrée [5]: from collections import Counter
            import matplotlib.pyplot as plt
            count_Class=pd.value_counts(train["label"], sort= True)
            count_Class.plot(kind= 'bar', color= ["blue", "orange"])
            plt.title('Bar chart')
            plt.show()
```

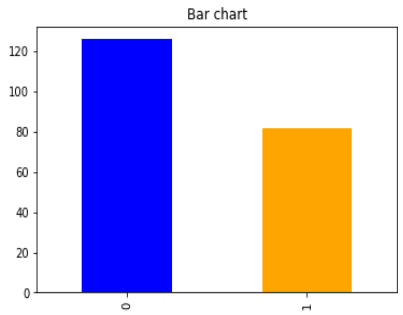

Figure 16:affichage des donnes en graphes

La figure 17 représente les messages seuls que nous avons extraits et qu'ensuite, nous avons divisés en mot sen utilisant la Procédure « *split()* »

```
Entrée [6]: text=train['POST']
            text=' '.join([w for w in text])
            text.split()
```

```
Out[6]: ['What',
         "didn't",
         'you',
         'understand',
         'in',
         'the',
         'lesson',
         'of',
         'android',
         'studio',
         '?',
         'i',
         'understood',
         'everything',
         'except',
         'the',
         'example',
         '01',
         'he',
```

Figure 17:converting donnée to text

La figure 18 représente à peu près le même travail que le précédent, cette fois nous avons devisé le texte en mots en utilisant la Procédure « *tokenize* ».

```
Entrée [7]: from nltk.tokenize import word_tokenize
            word_tokens= word_tokenize(text.lower())
            print(word_tokens)
```

```
['what', 'did', "n't", 'you', 'understand', 'in', 'the', 'lesson', 'of', 'android', 'studio', '?', 'i', 'understood', 'everythi
ng', 'except', 'the', 'example', '01', 'he', 'memorised', 'the', 'answers', 'of', 'the', 'past', 'exams', 'i', 'hate', 'schoo
l', 'she', 'did', "n't", 'see', 'the', 'lesson', 'of', 'the', 'android', 'he', 'studied', 'only', 'the', 'lesson', 'without',
'tp', 'is', 'there', 'tp', 'in', 'exam', '?', 'yes', ',', 'there', 'is', 'what', "'s", 'the', 'point', 'of', 'studding', 'andro
id', 'in', 'the', 'futur', '?', 'have', 'the', 'ability', 'to', 'build', 'application', 'mobile', 'with', 'android', 'studio',
'studied', 'the', 'answers', 'of', 'the', 'past', 'exams', 'that', "'s", 'not', 'a', 'question', ',', 'everything', 'we', 'stud
y', 'have', 'a', 'value', 'we', 'did', "n't", 'understand', 'how', 'on.setclick', '(', ')', 'work', '?', 'get', 'an', 'answer',
'from', 'the', 'application', 'your', 'answer', 'is', 'wrong', 'you', 'can', 'find', 'the', 'answer', 'in', 'the', 'lesson', 't
he', 'answer', 'is', 'not', 'complet', 'how', 'do', 'we', 'find', 'view', 'by', 'id', '?', 'who', 'have', 'a', 'right', 'answe
r', 'tell', 'us', ',', 'do', "n't", 'let', 'us', 'walk', 'in', 'the', 'wrong', 'way', 'find', 'widget', 'fils', 'by', 'id', 'wh
at', 'are', 'you', 'talking', 'about', 'what', 'lesson', '?', 'what', 'does', 'it', 'mean', 'layout', '?', 'it', "'s", 'an', 'i
nterface', 'graphique', 'what', 'does', 'it', 'mean', 'type', 'face', '?', 'it', "'s", 'noyeau', 'edit', 'text', '?', 'get', 'y
our', 'text', 'hhhhhhhhhhhh', 'can', 'the', 'android', 'write', 'all', 'the', 'word', 'in', 'capital', 'letters', '?', 'when',
'we', 'choose', 'capital', 'put', 'only', 'the', 'first', 'letter', 'in', 'capital', 'wrong', '!', 'your', 'answer', 'is', 'rig
ht', 'what', "'s", 'the', 'difference', 'between', 'android', 'num', 'and', 'android', 'digit', '?', 'in', 'digit', 'accept',
'only', 'the', 'numbers', 'and', 'espace', ',', 'but', 'in', 'num', 'accept', 'letters', 'and', 'numbers', 'without', 'espace',
'i', "'m", 'not', 'sure', 'the', 'professor', 'will', 'come', 'and', 'answer', 'you', 'can', 'we', 'find', 'all', 'the', 'answe
rs', 'in', 'the', 'lesson', '?', 'the', 'professor', 'told', 'you', 'to', 'make', 'your', 'own', 'resarchs', 'deamn', ',', 'i',
'can', 'take', 'any', 'more', 'in', 'this', 'university', 'get', 'lost', 'hhhhhhhhhhhh', 'cut', 'the', 'crap', 'we', "'re", 's
tudding', 'the', 'android', 'apps', 'toast', ',', 'what', 'does', 'it', 'do', '?', 'give', 'me', 'a', 'second', 'is', 'there',
'a', 'way', 'to', 'make', 'textview', 'the', 'n', 'you', 'delete', 'it', '?', 'you', 'choose', 'the', 'long', 'way', ',', 'toas
t', 'is', 'better', 'you', 'have', 'a', 'point', 'friends', ',', 'you', 'will', 'find', 'everything', 'in', 'the', 'lesson', 'n
o', '!', 'you', 'wo', "n't", 'get', 'it', 'all', 'the', 'time', 'what', "'s", 'the', 'difference', 'between', 'match', 'paren
t', 'and', 'wrap', 'content', '?', 'id', 'button', '?', 'who', 'know', 'the', 'answer', 'please', ',', 'the', 'exam', 'of', 'an
droid', 'when', '?', 'next', 'week', '!', 'i', "'m", 'going', 'to', 'teach', 'you', 'how', 'to', 'wtite', 'an', 'app', 'with',
'android', 'studio', 'i', "'ll", 'give', 'you', 'a', 'solution', 'of', 'an', 'exam', 'the', 'lesson', 'is', 'very', 'difficult',
'then', 'the', 'tp', 'we', 'have', 'a', 'problam', 'to', 'install', 'android', 'studio', 'in', 'the', 'pc', 'it', 'takes', 'a
t', 'least', '3', 'weeks', 'to', 'understand', 'it', 'i', "'m", 'so', 'tired', 'of', 'everything', 'that', 'make', 'two', 'hhhh
```


Figure 18:tokenize text to words

La figure 19 montre le nettoyage du texte que nous avons effectué en utilisant une fonction pour éliminer les {,’ ?§/. ?%* :!*$.

```
Entrée [8]: def remove_punct(txt):
             return [word for word in txt if word.isalpha()]
            sent = remove_punct(word_tokens)
            print(sent)
```

```
['what', 'did', 'you', 'understand', 'in', 'the', 'lesson', 'of', 'android', 'studio', 'i', 'understood', 'everything', 'exce
pt', 'the', 'example', 'he', 'memorised', 'the', 'answers', 'of', 'the', 'past', 'exams', 'i', 'hate', 'school', 'she', 'di
d', 'see', 'the', 'lesson', 'of', 'the', 'android', 'he', 'studied', 'only', 'the', 'lesson', 'without', 'tp', 'is', 'there',
'tp', 'in', 'exam', 'yes', 'there', 'is', 'what', 'the', 'point', 'of', 'studding', 'android', 'in', 'the', 'futur', 'have',
'the', 'ability', 'to', 'build', 'application', 'mobile', 'with', 'android', 'studio', 'studied', 'the', 'answers', 'of', 'th
e', 'past', 'exams', 'that', 'not', 'a', 'question', 'everything', 'we', 'study', 'have', 'a', 'value', 'we', 'did', 'underst
and', 'how', 'work', 'get', 'an', 'answer', 'from', 'the', 'application', 'your', 'answer', 'is', 'wrong', 'you', 'can', 'fin
d', 'the', 'answer', 'in', 'the', 'lesson', 'the', 'answer', 'is', 'not', 'complet', 'how', 'do', 'we', 'find', 'view', 'by',
'id', 'who', 'have', 'a', 'right', 'answer', 'tell', 'us', 'do', 'let', 'us', 'walk', 'in', 'the', 'wrong', 'way', 'find', 'w
idget', 'fils', 'by', 'id', 'what', 'are', 'you', 'talking', 'about', 'what', 'lesson', 'what', 'does', 'it', 'mean', 'layou
t', 'it', 'an', 'interface', 'graphique', 'what', 'does', 'it', 'mean', 'type', 'face', 'it', 'noyeau', 'edit', 'text', 'ge
t', 'your', 'text', 'hhhhhhhhhhhh', 'can', 'the', 'android', 'write', 'all', 'the', 'word', 'in', 'capital', 'letters', 'whe
n', 'we', 'choose', 'capital', 'put', 'only', 'the', 'first', 'letter', 'in', 'capital', 'wrong', 'your', 'answer', 'is', 'ri
ght', 'what', 'the', 'difference', 'between', 'android', 'num', 'and', 'android', 'digit', 'in', 'digit', 'accept', 'only',
'the', 'numbers', 'and', 'espace', 'but', 'in', 'num', 'accept', 'letters', 'and', 'numbers', 'without', 'espace', 'i', 'no
t', 'sure', 'the', 'professor', 'will', 'come', 'and', 'answer', 'you', 'can', 'we', 'find', 'all', 'the', 'answers', 'in',
'the', 'lesson', 'the', 'professor', 'told', 'you', 'to', 'make', 'your', 'own', 'resarchs', 'deamn', 'i', 'can', 'take', 'an
y', 'more', 'in', 'this', 'university', 'get', 'lost', 'hhhhhhhhhhhh', 'cut', 'the', 'crap', 'we', 'studding', 'the', 'andro
id', 'apps', 'toast', 'what', 'does', 'it', 'do', 'give', 'me', 'a', 'second', 'is', 'there', 'a', 'way', 'to', 'make', 'text
```

Figure 19: text cleaning

La figure 20 montre l'utilisation de la Procédure « stop_words ».

Entrée [9]:
```
from nltk.corpus import stopwords
stop_words= set(stopwords.words('english'))
stopwords = [s.lower() for s in stop_words]
filtered_list = [ w for w in sent if w not in stopwords and len(w)>2]
print (filtered_list)
```

```
['understand', 'lesson', 'android', 'studio', 'understood', 'everything', 'except', 'example', 'memorised', 'answers', 'past',
'exams', 'hate', 'school', 'see', 'lesson', 'android', 'studied', 'lesson', 'without', 'exam', 'yes', 'point', 'studding', 'and
roid', 'futur', 'ability', 'build', 'application', 'mobile', 'android', 'studio', 'studied', 'answers', 'past', 'exams', 'quest
ion', 'everything', 'study', 'value', 'understand', 'work', 'get', 'answer', 'application', 'answer', 'wrong', 'find', 'answe
r', 'lesson', 'answer', 'complet', 'find', 'view', 'right', 'answer', 'tell', 'let', 'walk', 'wrong', 'way', 'find', 'widget',
'fils', 'talking', 'lesson', 'mean', 'layout', 'interface', 'graphique', 'mean', 'type', 'face', 'noyeau', 'edit', 'text', 'ge
t', 'text', 'hhhhhhhhhhhh', 'android', 'write', 'word', 'capital', 'letters', 'choose', 'capital', 'put', 'first', 'letter',
'capital', 'wrong', 'answer', 'right', 'difference', 'android', 'num', 'android', 'digit', 'digit', 'accept', 'numbers', 'espac
e', 'num', 'accept', 'letters', 'numbers', 'without', 'espace', 'sure', 'professor', 'come', 'answer', 'find', 'answers', 'less
on', 'professor', 'told', 'make', 'resarchs', 'deamn', 'take', 'university', 'get', 'lost', 'hhhhhhhhhhhh', 'cut', 'crap', 'st
udding', 'android', 'apps', 'toast', 'give', 'second', 'way', 'make', 'textview', 'delete', 'choose', 'long', 'way', 'toast',
'better', 'point', 'friends', 'find', 'everything', 'lesson', 'get', 'time', 'difference', 'match', 'parent', 'wrap', 'conten
t', 'button', 'know', 'answer', 'please', 'exam', 'android', 'next', 'week', 'going', 'teach', 'wtite', 'app', 'android', 'stud
io', 'give', 'solution', 'exam', 'lesson', 'diffulct', 'problam', 'install', 'android', 'studio', 'takes', 'least', 'weeks', 'u
nderstand', 'tired', 'everything', 'make', 'two', 'hhhhh', 'khnows', 'android', 'studio', 'sites', 'web', 'follow', 'udemy', 'c
oursera', 'diffirence', 'height', 'understand', 'des', 'security', 'help', 'understnad', 'des', 'finished', 'security', 'let',
'come', 'back', 'android', 'value', 'xml', 'android', 'une', 'interface', 'grafique', 'get', 'welcome', 'get', 'data', 'base',
'android', 'talking', 'android', 'studio', 'general', 'give', 'diffinition', 'activity', 'still', 'studing', 'something', 'dat
e', 'exam', 'uknolodge', 'hhhhhhhhh', 'activity', 'graphique', 'deffinitions', 'lesson', 'find', 'resume', 'find', 'everythin
g', 'resume', 'thanks', 'idea', 'another', 'ideas', 'find', 'lesson', 'write', 'onkeylistner', 'work', 'resume', 'made', 'giv
e', 'email', 'yes', 'give', 'email', 'please', 'gmail', 'com', 'thanks', 'please', 'send', 'lesson', 'android', 'soon', 'possib
le', 'good', 'studant', 'keep', 'helping', 'friends', 'resume', 'double', 'lesson', 'hhhh', 'study', 'get', 'lost', 'understan
d', 'need', 'write', 'onlongclick', 'view', 'onstart', 'starts', 'activity', 'demarrer', 'one', 'activity', 'six', 'remember',
'tell', 'names', 'activitys', 'khnow', 'try', 'resume', 'khonw', 'ondestroy', 'onpause', 'ondestroy', 'finish', 'activity', 'on
pause', 'exist', 'first', 'interface', 'find', 'rest', 'tell', 'alright', 'probleme', 'onstart', 'ondestroy', 'onpause', 'onres
```

Figure 20 : remove stop words

La figure 21 montre l'utilisation de la procédure « *lemmetizer* »pour Renvoyer les mots à leur origine, par exemple, renvoyer les verbes à l'origine du mot.

Entrée [10]:
```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(word, pos = "v") for word in filtered_list]
print(lemmatized_words)
```

```
['understand', 'lesson', 'android', 'studio', 'understand', 'everything', 'except', 'example', 'memorise', 'answer', 'past', 'e
xams', 'hate', 'school', 'see', 'lesson', 'android', 'study', 'lesson', 'without', 'exam', 'yes', 'point', 'stud', 'android',
'futur', 'ability', 'build', 'application', 'mobile', 'android', 'studio', 'study', 'answer', 'past', 'exams', 'question', 'eve
rything', 'study', 'value', 'understand', 'work', 'get', 'answer', 'application', 'answer', 'wrong', 'find', 'answer', 'lesso
n', 'answer', 'complet', 'find', 'view', 'right', 'answer', 'tell', 'let', 'walk', 'wrong', 'way', 'find', 'widget', 'fils', 't
alk', 'lesson', 'mean', 'layout', 'interface', 'graphique', 'mean', 'type', 'face', 'noyeau', 'edit', 'text', 'get', 'text', 'h
hhhhhhhhhhh', 'android', 'write', 'word', 'capital', 'letter', 'choose', 'capital', 'put', 'first', 'letter', 'capital', 'wron
g', 'answer', 'right', 'difference', 'android', 'num', 'android', 'digit', 'digit', 'accept', 'number', 'espace', 'num', 'accep
t', 'letter', 'number', 'without', 'espace', 'sure', 'professor', 'come', 'answer', 'find', 'answer', 'lesson', 'professor', 't
ell', 'make', 'resarchs', 'deamn', 'take', 'university', 'get', 'lose', 'hhhhhhhhhhhh', 'cut', 'crap', 'stud', 'android', 'app
s', 'toast', 'give', 'second', 'way', 'make', 'textview', 'delete', 'choose', 'long', 'way', 'toast', 'better', 'point', 'frien
ds', 'find', 'everything', 'lesson', 'get', 'time', 'difference', 'match', 'parent', 'wrap', 'content', 'button', 'know', 'answ
er', 'please', 'exam', 'android', 'next', 'week', 'go', 'teach', 'wtite', 'app', 'android', 'studio', 'give', 'solution', 'exa
m', 'lesson', 'diffulct', 'problam', 'install', 'android', 'studio', 'take', 'least', 'weeks', 'understand', 'tire', 'everythin
g', 'make', 'two', 'hhhhh', 'khnows', 'android', 'studio', 'sit', 'web', 'follow', 'udemy', 'coursera', 'diffirence', 'height',
'understand', 'des', 'security', 'help', 'understnad', 'des', 'finish', 'security', 'let', 'come', 'back', 'android', 'value',
'xml', 'android', 'une', 'interface', 'grafique', 'get', 'welcome', 'get', 'data', 'base', 'android', 'talk', 'android', 'studi
o', 'general', 'give', 'diffinition', 'activity', 'still', 'stud', 'something', 'date', 'exam', 'uknolodge', 'hhhhhhhhh', 'acti
vity', 'graphique', 'deffinitions', 'lesson', 'find', 'resume', 'find', 'everything', 'resume', 'thank', 'idea', 'another', 'id
eas', 'find', 'lesson', 'write', 'onkeylistner', 'work', 'resume', 'make', 'give', 'email', 'yes', 'give', 'email', 'please',
'gmail', 'com', 'thank', 'please', 'send', 'lesson', 'android', 'soon', 'possible', 'good', 'studant', 'keep', 'help', 'friend
s', 'resume', 'double', 'lesson', 'hhhh', 'study', 'get', 'lose', 'understand', 'need', 'write', 'onlongclick', 'view', 'onstar
t', 'start', 'activity', 'demarrer', 'one', 'activity', 'six', 'remember', 'tell', 'name', 'activitys', 'khnow', 'try', 'resum
e', 'khonw', 'ondestroy', 'onpause', 'ondestroy', 'finish', 'activity', 'onpause', 'exist', 'first', 'interface', 'find', 'res
t', 'tell', 'alright', 'probleme', 'onstart', 'ondestroy', 'onpause', 'onresume', 'onstop', 'onrestart', 'dive', 'definitions',
'last', 'three', 'onresume', 'comeback', 'interface', 'onstop', 'visible', 'onrestart', 'make', 'interface', 'visivle', 'else',
```

Figure 21:lemmetize words

La figure 22 montre l'utilisation de la Procédure « *porter stemmer* »pour remettre les mots à leurs racines..

```
Entrée [11]: from nltk.stem import PorterStemmer
             from nltk.stem import SnowballStemmer
             stemmer_ps = PorterStemmer()
             stemmed_words_ps = [stemmer_ps.stem(word) for word in lemmatized_words]
             print(stemmed_words_ps)
```

```
['understand', 'lesson', 'android', 'studio', 'understand', 'everyth', 'except', 'exampl', 'memoris', 'answer', 'past', 'exa
m', 'hate', 'school', 'see', 'lesson', 'android', 'studi', 'lesson', 'without', 'exam', 'ye', 'point', 'stud', 'android', 'fu
tur', 'abil', 'build', 'applic', 'mobil', 'android', 'studio', 'studi', 'answer', 'past', 'exam', 'question', 'everyth', 'stu
di', 'valu', 'understand', 'work', 'get', 'answer', 'applic', 'answer', 'wrong', 'find', 'answer', 'lesson', 'answer', 'compl
et', 'find', 'view', 'right', 'answer', 'tell', 'let', 'walk', 'wrong', 'way', 'find', 'widget', 'fil', 'talk', 'lesson', 'me
an', 'layout', 'interfac', 'graphiqu', 'mean', 'type', 'face', 'noyeau', 'edit', 'text', 'get', 'text', 'hhhhhhhhhhhh', 'and
roid', 'write', 'word', 'capit', 'letter', 'choos', 'capit', 'put', 'first', 'letter', 'capit', 'wrong', 'answer', 'right',
'differ', 'android', 'num', 'android', 'digit', 'digit', 'accept', 'number', 'espac', 'num', 'accept', 'letter', 'number', 'w
ithout', 'espac', 'sure', 'professor', 'come', 'answer', 'find', 'answer', 'lesson', 'professor', 'tell', 'make', 'resarch',
'deamn', 'take', 'univers', 'get', 'lose', 'hhhhhhhhhhhh', 'cut', 'crap', 'stud', 'android', 'app', 'toast', 'give', 'secon
d', 'way', 'make', 'textview', 'delet', 'choos', 'long', 'way', 'toast', 'better', 'point', 'friend', 'find', 'everyth', 'les
son', 'get', 'time', 'differ', 'match', 'parent', 'wrap', 'content', 'button', 'know', 'answer', 'pleas', 'exam', 'android',
'next', 'week', 'go', 'teach', 'wtite', 'app', 'android', 'studio', 'give', 'solut', 'exam', 'lesson', 'diffulct', 'problam',
'instal', 'android', 'studio', 'take', 'least', 'week', 'understand', 'tire', 'everyth', 'make', 'two', 'hhhhh', 'khnow', 'an
droid', 'studio', 'sit', 'web', 'follow', 'udemi', 'coursera', 'diffir', 'height', 'understand', 'de', 'secur', 'help', 'unde
rstnad', 'de', 'finish', 'secur', 'let', 'come', 'back', 'android', 'valu', 'xml', 'android', 'une', 'interfac', 'grafiqu',
'get', 'welcom', 'get', 'data', 'base', 'android', 'talk', 'android', 'studio', 'gener', 'give', 'diffinit', 'activ', 'stil
l', 'stud', 'someth', 'date', 'exam', 'uknolodg', 'hhhhhhhhh', 'activ', 'graphiqu', 'deffinit', 'lesson', 'find', 'resum', 'f
ind', 'everyth', 'resum', 'thank', 'idea', 'anoth', 'idea', 'find', 'lesson', 'write', 'onkeylistn', 'work', 'resum', 'make',
```

Figure22: stemm words

La figure 23 montre l'obtention d'un graphe représentant les 25 mots les plus utilisés.

```
Entrée [12]: from collections import Counter
             import seaborn as sns
             all_list = stemmed_words_ps
             all_counts = Counter(all_list)
             all_words = [word[0] for word in all_counts.most_common(25)]
             all_counts = [word[1] for word in all_counts.most_common(25)]
             plt.style.use('ggplot')
             plt.figure(figsize=(15, 12))
             sns.barplot(y=all_words, x=all_counts)
             plt.title('most word');
```

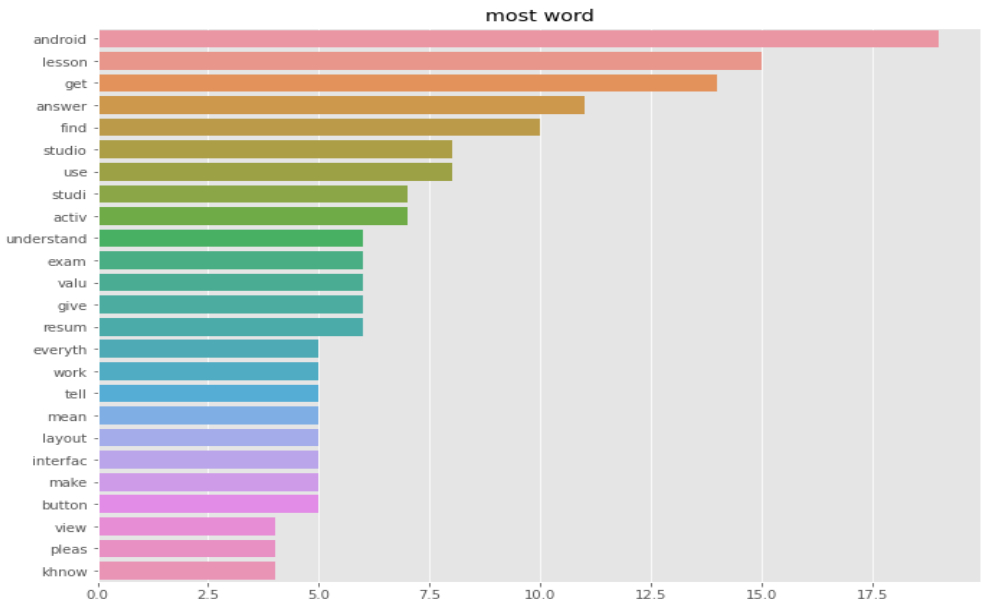

Figure 23: Most Common words

La figure 24 montre comment nous avons divisé les données en deux parties, la première partie contient « 70 %» des données initiales, et permet d'entraîner la machine à la classification. La deuxième partie« 30 %» de données restantes est utilisée plus tard pour voir la capacité de la machine à classer correctement.

```
Entrée [13]: from sklearn.model_selection import train_test_split

             X = train['POST']
             y = train['label']
             X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42)

Entrée [14]: y_train.value_counts()

Out[14]: 0    87
         1    58
         Name: label, dtype: int64

Entrée [15]: y_test.value_counts()

Out[15]: 0    39
         1    24
         Name: label, dtype: int64

Entrée [16]: from sklearn.feature_extraction.text import CountVectorizer
             count_vect = CountVectorizer()

             X_train_counts = count_vect.fit_transform(X_train)
             X_train_counts.shape

Out[16]: (145, 327)
```

Figure 24: split data to train and test

La figure 25montre l'utilisation de la Procédure de « *TF-IDF* » permettant de représenter les mots avec des vecteurs. Pour ce faire, nous avons utilisé le premier algorithme de classification, à savoir : *linearSvc()*et on a affiché par la suite ses résultats.

```
Entrée [17]: from sklearn.feature_extraction.text import TfidfTransformer
             tfidf_transformer = TfidfTransformer()

             X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
             X_train_tfidf.shape

Out[17]: (145, 327)

Entrée [18]: from sklearn.feature_extraction.text import TfidfVectorizer
             vectorizer = TfidfVectorizer()

             X_train_tfidf = vectorizer.fit_transform(X_train) # remember to use the original X_train set
             X_train_tfidf.shape

Out[18]: (145, 327)

Entrée [19]: from sklearn.svm import LinearSVC
             clf = LinearSVC()
             clf.fit(X_train_tfidf,y_train)

Out[19]: LinearSVC()

Entrée [20]: from sklearn.pipeline import Pipeline
             text_clf = Pipeline([('tfidf', TfidfVectorizer()),
                                  ('clf', LinearSVC()),])
             text_clf.fit(X_train, y_train)

             predictions = text_clf.predict(X_test)

Entrée [21]: from sklearn import metrics
             print(metrics.confusion_matrix(y_test,predictions))

             [[25 14]
              [ 5 19]]

Entrée [22]: print(metrics.classification_report(y_test,predictions))

                          precision    recall  f1-score   support

                       0       0.83      0.64      0.72        39
                       1       0.58      0.79      0.67        24

                accuracy                           0.70        63
               macro avg       0.70      0.72      0.70        63
            weighted avg       0.74      0.70      0.70        63

Entrée [23]: print(metrics.accuracy_score(y_test,predictions))

             0.6984126984126984
```

Figure 25: apply linearSvc and result

La figure 26 représenté l'utilisation du deuxième algorithme de classification « *naive-bayes : MultinomailNB()* » et l'afficher de ses résultats.

```
Entrée [24]: from sklearn.naive_bayes import MultinomialNB
             nb_model = MultinomialNB()
             nb_model.fit(X_train_tfidf, y_train)

  Out[24]: MultinomialNB()

Entrée [25]: from sklearn.pipeline import Pipeline
             text_clf1 = Pipeline([('tfidf', TfidfVectorizer()),
                                   ('nb_model', MultinomialNB()),])
             text_clf1.fit(X_train, y_train)
             predictions1 = text_clf1.predict(X_test)

Entrée [26]: from sklearn import metrics
             print(metrics.confusion_matrix(y_test,predictions1))

             [[27 12]
              [10 14]]

Entrée [27]: print(metrics.classification_report(y_test,predictions1))

                          precision    recall  f1-score   support

                       0       0.73      0.69      0.71        39
                       1       0.54      0.58      0.56        24

                accuracy                           0.65        63
               macro avg       0.63      0.64      0.64        63
            weighted avg       0.66      0.65      0.65        63

Entrée [28]: print(metrics.accuracy_score(y_test,predictions1))

             0.6507936507936508
```

Figure 26: apply naive_bayes and result

La figure 27 présente l'utilisation du deuxième algorithme de classification « *naive-bayes : SVC ()* » et l'affichage de ses résultats.

```
Entrée [29]: from sklearn.svm import SVC
             svc_model = SVC(gamma='auto')
             svc_model.fit(X_train_tfidf,y_train)

  Out[29]: SVC(gamma='auto')

Entrée [30]: from sklearn.pipeline import Pipeline
             text_clf2 = Pipeline([('tfidf', TfidfVectorizer()),
                                   ('svc_model', SVC(gamma='auto')),])
             text_clf2.fit(X_train,y_train)
             predictions2 = text_clf2.predict(X_test)

Entrée [31]: from sklearn import metrics
             print(metrics.confusion_matrix(y_test,predictions2))

             [[39  0]
              [24  0]]

Entrée [32]: print(metrics.classification_report(y_test,predictions2))

                          precision    recall  f1-score   support

                       0       0.62      1.00      0.76        39
                       1       0.00      0.00      0.00        24

                accuracy                           0.62        63
               macro avg       0.31      0.50      0.38        63
            weighted avg       0.38      0.62      0.47        63

Entrée [33]: print(metrics.accuracy_score(y_test,predictions2))

             0.6190476190476191
```

Figure 27: apply SVC and result

La figure 28donne une vue en 'Nuage de mots' postés dans le forum et ce en utilisant la procédure « *wordcloud* »

```
Entrée [34]: from wordcloud import WordCloud
             import matplotlib.pyplot as plt
             text=' '.join(all_list)
             wordcloud = WordCloud(width=800, height=500,
                                   random_state=21, max_font_size=110).generate(text)
             plt.figure(figsize=(15, 12))
             plt.imshow(wordcloud, interpolation="bilinear")
             plt.axis('off');
             plt.show()
```

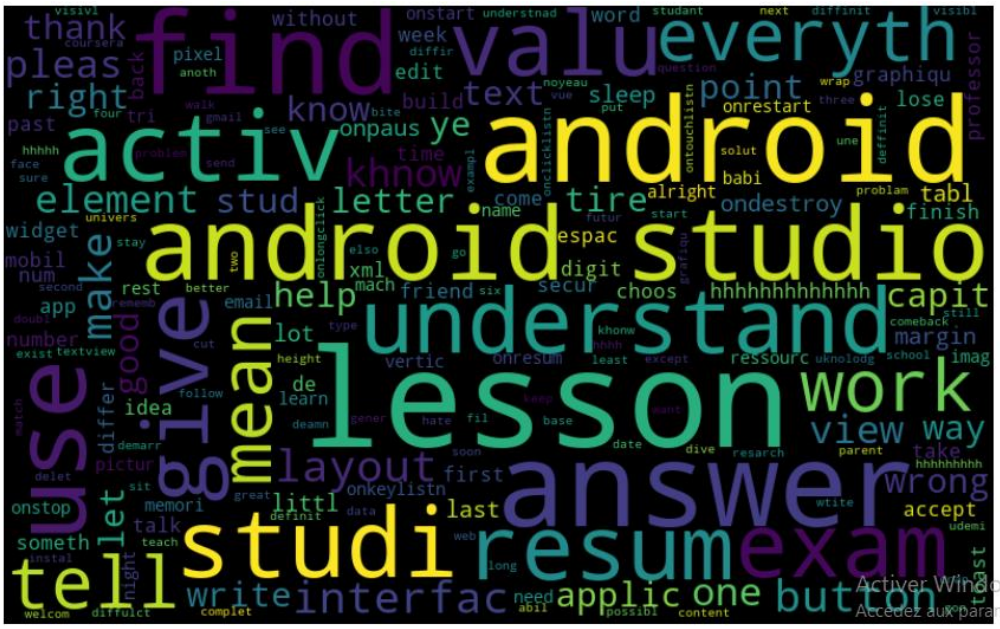

Figure 28: wordCloud for most common words

La figure 29 montre l'utilisation d'un graphe permettant de représenter le message « pertinent = p, non-pertinent = nonp » en utilisant un nouveau message afin de savoir s si la machine est capable de classifier ses messages correctement.

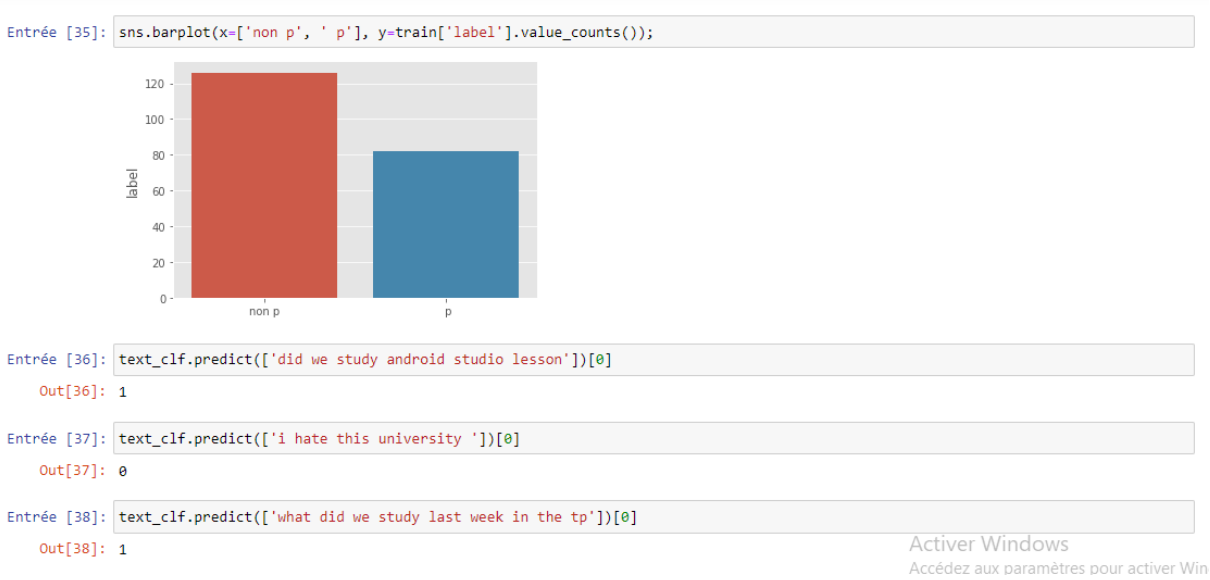

Figure 29:testing for classification

Conclusion :

Dans l'implémentation de notre solution, nous avons utilisé trois techniques pour classer les messages issus du texte préalablement nettoyé et organisé, à savoir : (1) LinearSVC, (2) naive_bayes :MultinomialNB et (3) SVC .

Le tableau représenter la comparaison entre 03 algorithmes de classification

« p représente pertinent =1 » et « non p représente non pertinent = 0 »

Classification algorithme	**Matrices Confusion**	**P Non p**	**précision**	**recall**	**f1-score**	**support**	**accuracy**
LinearSVC ()	25 14 5 19	0 1	0.83 0.59	0.64 0.73	0.72 0.67	39 24	0.698
naive_bayes : MultinomialNB ()	27 12 10 14	0 1	0.73 0.54	0.69 0.58	0.71 0.56	39 24	0.650
SVC ()	39 0 24 0	0 1	0.62 0.00	1.00 0.00	0.76 0.00	39 24	0.619

Tableau 1 : Représentation des résultats

L’application de l’algorithme "*LinearSVC*" a donné les meilleurs résultats avec un pourcentage avoisinant "9.68" où sur 39 non-texte messages la capacité de La machine a reconnu 25 messages appelés «True positive » et un taux d'erreur de 14 messages appelés «False positive », quant aux messages associés estimés à 24 messages appelés « True négative » 19 messages ont été identifiés et un taux d'erreur de 5 messages appelés «False négative ».

Ceci est une expérience un peu moyenne mais qui nous a permis d’appréhender le domaine de l’IA, notamment le Data Mining et le Text Mining.

Conclusion générale

Dans ce mémoire, nous avons parlé de l'enseignement à distance, en tant que référence au développement des plateformes d'enseignement à distance qui sont utilisées par les universités du monde entier. Nous nous sommes focalisés par la suite à l'activité d'apprentissage de type Forum au sein de la plateforme de formation à distance Moodle. Cette activité collaborative fournit un espace d'interaction entre les enseignants et les étudiants ou entre les étudiants eux-mêmes. Les résultats de ces interactions sont représentés dans les messages postés dans les forums de discussion.

Notre projet consiste à concevoir et construire une application basée sur des techniques de Text Mining afin de classer les messages écrits par les étudiants en messages pertinents ou non, en d'autres termes distinguer les messages qui sont liés au cours Cours de ceux qui ne le sont pas.

Nous avons divisé notre travail en 4 étapes :

- Dans une première étape, nous avons discuté des définitions de l'e-learning, des plateformes d'apprentissage à distance, de la plateforme Moodle et du forum, puis nous avons parlé des techniques de Data Mining, des étapes et des types, puis nous avons abordé les techniques de Text Mining et tout ce qui s'y rapporte.
- Dans la deuxième étape, nous avons étudié trois travaux dans le domaine du Text Mining et le Data Mining, dont l'objectif commun était de prédire les étudiants susceptibles d'échouer et par voie de conséquence permettre à l'enseignant de les aider à étudier.
- Dans la troisième étape, nous avons présenté la conception de notre solution à travers une présentation séquentielle du travail que nous allons faire en utilisant des des formulaires.
- Dans la quatrième étape, nous avons présenté l'application que nous avons faite afin de classer les messages. Nous avons utilisé trois algorithmes qui nous ont permis d'obtenir des résultats..

De ces résultats modestes, nous concluons que dans notre travail, le meilleur algorithme était « *linearsvc* » qui nous permis d'obtenir un résultat de « 68.9% ».

De là, nous avons une machine qui est capable de classer les messages écrits par les étudiants, à savoir s'ils sont pertinents au Cours fourni par l'enseignant ou non.

Comme perspective, nous envisageons d'approfondir nos connaissances dans les techniques de Text Mining et d'en tirer profit dans leurs applications aux activités de type Forums afin de contribuer plus efficacement à l'amélioration de l'apprentissage des étudiants.

Bibliographie

[01]**Jean-Pierre ROBERT**, 2008 : Dictionnaire pratique de didactique de FLE,p.198.

[02] Direction des Systèmes d'Information, « *Fiche Pratique Moodle*» université de technologie Compiègne, Année 2004.

[03] Pierre-Léonard Harvey, « *les plates-formes d'apprentissage en ligne* », Université du Québec à Montréal, Année 2003.

[04] Team Cyberlearn, « *COURS MOODLE*», centree-learning hes-so

cyberlearn, Année 2007.

[05] The Gartner Group, www.gartner.com.

[06] https://www.oracle.com/dz/data-science/what-is-data-science/

[07] D. HAND, H. MANNILA et P. SMYTH, Principles of Data Mining , MIT Press, Cambridge, MA, 2001.

[08] P. CABENA, P. HADJINIAN, R. STADLER, J. VERHEES et A.

[09] ZANASI, Discovering Data Mining : From Concept to Implementation, Prentice Hall, Upper Saddle River, NJ, 1998

[10] figure 03 : https://jafwin.com/2019/07/05/lessentiel-a-savoir-sur-le-data-mining/

[11] O. R. ZAÏANE, Principles of Knowledge Discovery in Databases, CMPUT690, University of Alberta, 1999.

[12] Ph. PREUX, Fouille de données : Notes de cours, Université de Lille 3

.[13] https://zipreporting.com/fr/data-mining/data-mining-process.html

[14] G. DONG, J. PEI, Sequence Data Mining , Springer Edition, 2007.

[15]P.Vincent, « Modèles à noyaux à structure locale », Thèse de Phd en informatique , Université de Montréal,2003.

[16] Mokhtar Taffar. « Initialisation à l'apprentissage automatique ». Université de Jijel, Département Informatique, Algérie. 201

[17] https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

[18]https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68 L'un des algorithmes de clustering les plus utilisés est le « K-Means ».

[19] Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From Data Mining to knowledge discovery: An Overview. In Advances in Knowledge Discovery and Data Mining , U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds., MIT Press,Cambridge, Mass., 1-36.

[20] Feldman, R. & Dagan, I. (1995) Knowledge discovery in textual databases (KDT). Inproceedings of the First International Conference on Knowledge Discovery and DataMining (KDD-95), Montreal, Canada, August 20-21, AAAI Press, 112-117.

[21] Hearst, M. A. (1997) Text Data Mining : Issues, techniques, and the relationship toinformation access. Presentation notes for UW/MS workshop on Data Mining , July 1997.

[22] Simoudis, E. (1996). Reality check for Data Mining . IEEE Expert, 11(5).

[23] Dani Yogatama and Noah A. Smith, (2014), Making the Most of Bag of Words: Sentence Regularization with Alternating

Direction Method of Multipliers, Proceedings of the 31st International Conference on Machine Learning, Beijing, China, JMLR: W&CP, volume 32

[24] Cordelia Schmid, Bag-of-features for category classification, www.cs.umd.edu/~djacobs/CMSC426/BagofWords.pdf.

[25] Jialu Liu, Image Retrieval based on Bag-of-Words model, jialu.cs.illinois.edu/technical_notes/CBIR_BOW.pdf

[26] H. Matallah, « Classification Automatique de Textes Approche Orientée Agent ». Mémoire de magister, Département de l'informatique, université d'AboubekrBelkaid-Tlemcen, Février 2011.

[27] S. Jaillet, « Catégorisation Automatique De Documents » LIRMM UMR 5506, 161 rue Ada, 34392 Montpellier Cedex 5 – France URL, 2004.

[28] Gupta G 2015 Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example) Int. J. Comput. Appl. 24–6

[29] M.F. Porter, An Algorithm for Suffix Stripping,Program, vol. 14, no. 3, pp. 130-137, 1980.

[30] Ms. Anjali Ganesh Jivani, A Comparative Study of Stemming Algorithms, Anjali Ganesh Jivani et al, Int. J. Comp. Tech. Appl., Vol 2 (6), 1930-1938, ISSN:2229-6093

[31] Deepika Sharma, Stemming Algorithms, A Comparative Study and their Analysis, International Journal of Applied Information Systems (IJAIS) –ISSN : 2249-0868, Foundation of Computer Science FCS, New York, USA, Volume 4– No.3, September 2012 – www.ijais.org

[32] Ms. Anjali Ganesh Jivani, A Comparative Study of Stemming Algorithms, Anjali Ganesh Jivani et al, Int. J. Comp. Tech. Appl., Vol 2 (6), 1930-1938, ISSN:2229-6093.

[33] Deepika Sharma, Stemming Algorithms, A Comparative Study and their Analysis, International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868, Foundation of Computer Science FCS, New York, USA, Volume 4– No.3, September 2012 – www.ijais.org.

[34] C.Ramasubramanian and R.Ramya, Effective PreProcessing Activities in Text Mining using Improved Porter‟s Stemming Algorithm, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 12, December 2013, ISSN (Online) : 2278-1021.

[35] Efficient Estimation of Word representation in. Vector space. Tomas Mikolov.gool Inc. Mountain View CA.

[36] MNaili, A.H.Chaibi, dan H.H.BenGhezala.”comparative study of Word EmbedingMethodes in Topic segmentations,” procedia computer scence.2017.p.p.340-349.vol.12.

[37] H. Dahmani, « Classification des documents médicaux basée sur le Text Mining » Mémoire de Master, Département de l’informatique, Université de saâddahlab blida, 2012.

[38] R. Jalam, « Apprentissage Automatique Et Catégorisation De Textes Multilingues », 2003.