Dédicaces

Louanges à mon dieu ALLAH,

qui m'a donné la volonté et le courage pour la réalisation de ce travail.

Je dédie ce projet à mon cher père Ridha et à ma chère mère Aljia qui m'ont comblé de leur soutien et m'ont voué un amour inconditionnel. Vous êtes pour moi un exemple de courage et des sacrifices continu, que cet humble travail témoigne mon affection, mon éternel attachement et qu'il appelle sur moi votre continuelle bénédiction. Je dédie ce travail :

À ma magnifique sœur Aya

À mes frères Aymen et Achref

À tous mes amis et toute personne qui ont contribué à la réalisation de ce travail

Puisse ce projet apporter la pleine satisfaction à tous ceux qui le lisent.

Mille Merci

Hammami Mohamed

Dédicaces

Louanges à mon dieu ALLAH,

qui m'a donné la volonté et le courage pour la réalisation de ce travail.

Je dédie ce projet à :

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude,

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Allah faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutient permanent venu de toi,

Mes frères Naim et Wassim,

Mes sœur Mariem et Houda,

Mes amis proches Ala Ouni, Salem Nasraoui, Mourad Chebbi, Najeh Abbassi et Refka Bouzaien,

Et à toutes les personnes qui ont contribué à la réalisation de ce travail

Puisse ce projet apporter la pleine satisfaction à tous ceux qui le lisent.

Mille Merci

Dhifli Mohamed

Remerciements

Nous tenons à remercier tout d'abord notre encadrant Mr. Mounir Zrigui et Mr. Houssem Abdellaoui, nous serons vaniteux si nous nous devions énumérer en ces quelques lignes votre remarquables qualités humaines et professionnelles, veuillez trouver ici l'expression et le témoignage de notre gratitude ressentie. Nous tenons d'emblée à vous remercier spécialement de tout l'intérêt que vous aviez bien voulu porter à notre travail, vos conseils, vos explications, vos directions et remarques, que ce modeste travail vous honore et vous témoigne nos reconnaissances. Nous tenons aussi mercier tous les membres du jury pour avoir accepté de juger notre travail. Nous exprimons toutes nos reconnaissances et gratitudes à l'administration et à l'ensemble du corps enseignant de la faculté des sciences de Monastir, pour leurs efforts à nous garantir la continuité et l'aboutissement de ce Projet fin d'études. Enfin, nous remercions tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce modeste travail et en particulier nos familles et nos amis qui par leurs prières et leurs encouragements, nous avons pu surmonter tous les obstacles

Sommaire

Table de figures

Introduction générale

Le traitement automatique des langues naturelles est un domaine d'application d'intelligence artificielle qui s'intéresse à l'aspect de la langue humain. L'un des domaines de TALN, l'analyse des sentiments s'est manifestée vers le début des années 2000 et connait un succès grandissant dû à l'abondance des données provenant de réseaux sociaux, en particulier celles fournies par Twitter.

Dans notre projet, on avait la chance d'intégrer dans une équipe qui focalise ses recherches sur l'extraction et l'analyse des opinions pour la langue arabe. L'objectif principal de cette étude est de réaliser un extracteur qui permet à partir d'un texte donné de détecté la polarité du texte, il peut être positive, négative ou bien neutre.

Ce rapport est organisé en deux grandes parties : Cadre théorique et cadre pratique, chacune contient deux chapitres.

Le Cadre théorique :

Le premier chapitre intitulé « l'analyse des sentiments », présentera les aspects de l'intelligence artificielle et ses applications comme le traitement automatique de la langue naturel ensuite on propose une étude théorique sur l'analyse des sentiments afin d'expliquer ses approches.

Le deuxième chapitre intitulé « Description des algorithmes », dans lequel on s'intéresse aux algorithmes d'apprentissage, leur fonctionnement, leurs avantages et inconvénients et à la validation croisée.

Le Cadre pratique :

Le premier chapitre intitulé « environnement du travail» concerne la description de l'environnement de travail à savoir l'outil de développement et les Framework utilisés.

Ce rapport sera clôturé par un chapitre « réalisation », nous détaillerons les différentes étapes de réalisation de notre projet avec les résultats des tests réalisés ainsi que des captures d'écran des différentes interfaces du site web créé.

Partie I : Cadre théorique

Chapitre 1 : Analyse des sentiments

Introduction

Le concept de l'intelligence artificielle (IA) se focalise sur l'élaboration des programmes informatiques capables d'effectuer des tâches accomplies par des humains demandant un apprentissage, une organisation de la mémoire et un raisonnement. Le but est de donner des notions de rationalité, des fonctions de raisonnement et de perception. L'évolution atteinte par l'IA a engendré une prolifération dans les domaines d'application tels que le TALN. [1]

Dans ce chapitre, on s'intéressera à l'intelligence artificielle, au traitement automatique de langue naturelle et ses différents domaines d'utilisation puis à l'analyse des sentiments et ses approches.

1. Intelligence Artificielle

1.1. Définition

Définissons tout d'abord l'intelligence :

- L'intelligence est l'ensemble des facultés mentales permettent de comprendre les choses et le faits de découvrir les relations entres eux,

Définissons également Le terme artificiel :

- Le terme artificiel se rapporte à tous ce qui est n'est pas naturel, et implique généralement qui cela a été créé ou fabriqué par la main de l'homme,

En effet l'intelligence artificielle est un domaine de l'informatique qui traite de donner aux machines la capacité de ressembler à l'intelligence humaine.

Il existe deux types d'intelligence artificielle :

- Intelligence artificielle faible :

La notion d'intelligence artificielle faible constitue une approche pragmatique d'ingénieur : chercher à bâtir des systèmes de plus en plus autonomes, des algorithmes capables de résoudre des problèmes d'une certaine classe. Mais cette fois la machine simule l'intelligence, elle paraît agir comme si elle était intelligente, il s'agit donc d'un programme préalable effectué par l'homme,

- Intelligence artificielle forte :

Le concept d'intelligence artificielle forte fait indiquer à une machine capable non seulement de produire un comportement intelligent, mais d'éprouver une trace d'une réelle cognition de soi, « vrais sentiments », et une compréhension de ses propres raisonnements. [2]

1.2. Différents domaine d'application

Les réalisations actuelles de l'intelligence artificielle peuvent être regroupées en différents domaines, tel que :

- L'apprentissage automatique,
- Les systèmes d'expert,
- Le traitement automatique de la langue naturelle [3]

2. Traitement automatique de la langue naturelle

2.1. Définition

Le traitement automatique de la langue naturelle (TALN) est le domaine d'application de l'intelligence artificielle qui s'intéresse à l'étude des aspects du langage humain.

Le but du TALN est d'utiliser des règles de la langue des qu'on doit expliciter puis les représenter dans un formalisme opératoire et calculable afin de les implémenter pour développer des moyens logiciels capables de traiter automatiquement des données linguistiques. [4]

2.2. Les domaines d'utilisation

Le TALN ouvre les portes de l'innovation et les idées de l'utilisation sont plusieurs.

Le TALN s'intéresse principalement au traitement automatique de la parole (c'est la captation, la transmission, l'identification et la synthèse de la parole), à l'analyse linguistique automatique du texte (qui soumet deux branches principales qui sont la traduction automatique et la génération automatique du texte dit aussi le développement de système question-réponse) et à l'analyse des sentiments qu'on s'intéresse dans notre projet.

3. Analyse des sentiments

3.1. Définition

L'analyse des sentiments est l'extraction de l'opinion à partir d'un texte et de détecter sa polarité par rapport à un sujet, il s'agit du processus qui permet de déterminer l'avis qui se cache derrière un texte, on parle alors de polarité, elle peut être positive, neutre ou négative. C'est la partie du text mining qui essaye de définir les opinions, sentiments et attitudes présente dans un texte ou un ensemble de texte. Développée essentiellement depuis les années 2000. [5]

3.2. Retombés sociale et scientifique

Etant donné l'ampleur de l'internet, tous ces informations constituent une source de données immense, facilement accessible et sans frais, pour les chercheurs et avant tout pour l'industrie dans différents domaines. Les politologues peuvent utiliser cette information pour déterminer le candidat ou la partie politique reçoit le plus de soutient. Les sociologues peuvent estimer les besoin de la population. Les études de marché auront probablement le plus grands d'intérêt dans un système automatique et précis d'analyse des avis d'utilisateurs. Ils ont besoin de détecter la tendance actuelle et d'y répondre, ce qui rend leurs produits souhaitable pour un plus large public. Enfin, la sécurité publique peut bénéficier des applications de sécurités d'analyse et de détection des sentiments antisociaux pour la prévention des risques.

3.3. Approches

L'analyse des sentiments est un procès délicat et sensible au contexte et à l'environnement et à la langue ou on s'exprime. Il nécessite d'avoir de phrase qui exprime l'opinion de façon claire sinon l'identification des sentiments devient difficile.

L'analyse peut s'effectuer à différents niveaux :

Au niveau du document : détermine l'opinion générale de l'ensemble d'un document,

Au niveau de la phrase : détermine l'opinion générale d'une phrase (positive, négative ou neutre),

Au niveau des aspects : au lieu de déterminer les entités à analyser en fonction de critère structuraux (phrase, paragraphe, document) ces méthodes se basent sur un analyse de corrélation entre l'opinion émise et la cible de cette opinion.

Il existe deux approches d'analyse des sentiments : l'approche basé sur un lexique et l'approche basé sur l'apprentissage.

3.3.1. Approche basée sur les lexiques

Les approches lexicales utilisent des dictionnaires de mots et pour lesquels la tonalité peut être pré-codée.

Ces dictionnaires sont utilisés pour classifier des textes dont on sait qu'ils parlent de l'entité nommée qui nous intéresse. L'analyse du texte se base sur les dictionnaires marquant le sentiment positif ou négatif. Il s'agit pour la plupart de verbes et d'adjectifs, mais aussi de quelques noms communs et d'adverbes.

Limites de l'analyse des sentiments à partir des lexiques :

- Les dictionnaires affectent une tonalité positive ou négative à un mot, sans tenir compte du contexte,
- Les dictionnaires ont tendance à éliminer les mots à valence ambiguë a priori,
- Le traitement des expressions ambiguës reste à faire et demande de faire appel à d'autres principes et à d'autres techniques,
- Lorsque la négation n'est pas prise en compte (ce qui peut paraître étonnant mais qui existe encore, par exemple dans les méthodes basées sur les « sacs de mots », qui calculent des fréquences d'occurrence dans un texte), le score de polarité peut être complètement faussé. [6]

3.3.2. Approche basée sur l'apprentissage automatique

Les techniques d'apprentissage automatiques (machine learning en anglais) offrent de meilleurs résultats que les méthodes linguistiques. Ils précisent toutefois que les dictionnaires d'opinion utilisés ne sont peut-être pas optimaux.

Dans cette approche, la machine est entraînée à détecter des expressions subjectives en la faisant travailler sur un premier échantillon de données (échantillon d'apprentissage) : cet apprentissage est basé sur la détection des caractéristiques qui peuvent déterminer les aspects de polarité. Elle doit être capable de détecter ensuite ces modèles dans lui-même, voire d'en détecter de nouveaux, proches de ceux qu'elle connaît déjà. [6]

Cette approche peut être postulée par deux méthodes :

L'apprentissage non supervisé : il vise à caractériser la distribution des données, et les relations entre les variables, sans discriminer entre les variables observes et les variables à prédire.

En effet, il divise un groupe hétérogène de données, en sous-groupes de manière que les données considérées comme les plus similaires soient associées au sein d'un groupe homogène et qu'au contraire les données considérées comme différentes se retrouvent dans d'autres groupes distincts ; est de tirer de la valeur de données dans lesquelles l'attribut à prédire n'apparaît pas. [7]

On distingue 3 principales tâches réalisables de l'apprentissage non-supervisé :

1. Le partitionnement des données,
2. La détection d'éléments atypiques,
3. La reduction de dimensions.

L'apprentissage supervisé : c'est une technique d'apprentissage automatique, son but est de définir des règles de classification à partir de l'entrainement et en utilisant des variables de qualification et de quantification qui caractérisent un échantillon de données utilisé pour l'entrainement dit ensemble d'apprentissage puis utilise ces règles et un algorithme de classification pour prédire les nouvelles données. Il nécessite l'étude de la fiabilité de ces règles et les comparer avant de les appliquer sur les nouvelles entrées en utilisant un deuxième échantillon de données appelé ensemble de validation ou ensemble de teste. [8]

Conclusion

Dans ce chapitre, nous avons défini l'intelligence artificielle et le traitement automatique des langues naturelles, nous avons présenté l'utilisation TALN dans plusieurs domaines. Ensuite, on a présenté le domaine d'analyse des sentiments et on a donné les différentes approches arrivant aux différents types d'apprentissage et leur principe de fonctionnement. Dans le chapitre suivant on présentera quelques algorithmes d'apprentissage supervisé.

Chapitre 2 : Description des algorithmes

Introduction

Principalement, par apprentissage automatique, on tente de résoudre le problème de la catégorisation automatique de données et de remplir des taches qu'il est difficile ou impossible de satisfaire par les méthodes classiques.

L'apprentissage automatique a émergé dans la seconde moitié du 20ème siècle du domaine de l'intelligence artificielle et correspond à l'élaboration d'algorithmes capables d'accumuler de la connaissance et de l'intelligence à partir d'expériences, En Bref, c'est un type d'intelligence artificielle qui offre aux ordinateurs la capacité d'apprendre sans être explicitement programmé. [9]

En Effet, l'apprentissage automatique utilise plusieurs techniques permettant de construire des modèles qui peuvent être utilisés dans différentes applications, telles que la prédiction ou la classification, l'apprentissage supervisé est l'une de ces techniques.

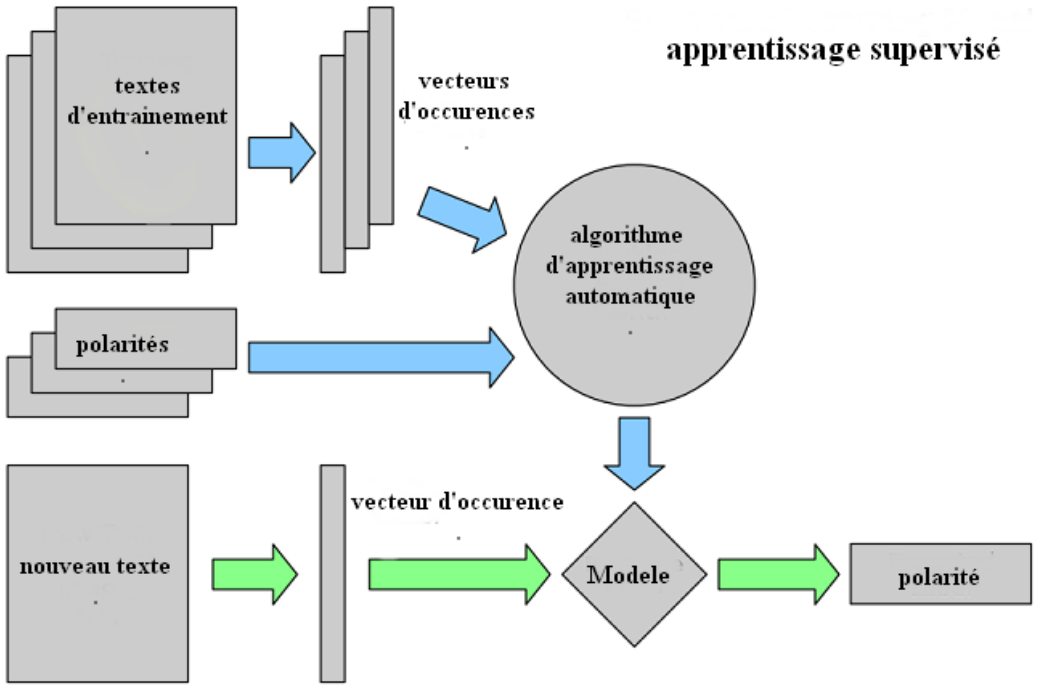

Figure 1 : Apprentissage supervisé

Dans ce cadre, plusieurs algorithmes ont étés mis au point, dans cette étude, on s'intéresse de ces algorithmes en détaillant trois d'entre eux qui sont k Nearest Neighbors, Support Vector Machine et Naïve Bayes.

1. K Nearest Neighbors (KNN)

1.1. Description

La méthode k Nearest Neighbors (KNN) est un algorithme est l'un des algorithmes d'apprentissage supervisé qui stocke tous les cas disponibles et classe de nouveaux cas à partir d'une mesure de similarité, L'algorithme KNN est donc une méthode à base de voisinage, non-paramétrique. [10]

Dans un contexte de classification d'une nouvelle observation x, l'idée est de faire voter les k plus proches voisins de cette observation. La classe de x est déterminée selon la majorité.

KNN a été utilisé dans l'estimation statistique et reconnaissance des formes déjà au début des années 1970 comme une technique non paramétrique.

1.2. Algorithme

Soit $D = \{(x', c), c \in C\}$ l'ensemble d'apprentissage.

Soit x l'exemple dont on souhaite déterminer la classe :

L'algorithme :

```
Début
    Pour chaque ((x', c) ∈ D) faire
    CalculerLaDistance dist (x, x')
     Fin
     Pour chaque {x' ∈ kppv(x)}  faire
          CompterNombreOccurrenceDeChaqueClasse()
      Fin
Fin      [11]
```

1.3. Principe

Le principe de cet algorithme de classification est très simple .On lui fournit :

Un ensemble d'apprentissage D : les données à classer et leurs polarités,

Un entier K : le nombre de voisin pris en charge par l'algorithme,

Une fonction de distance d : fonction qui va calculer la distance entre les fonctionnalités des données.

Pour tout nouveau point de test x, pour lequel il doit prendre une décision

- Chercher dans D les K points les plus proches de x au sens de la distance d
- Attribuer x à la classe qui est la plus fréquente parmi ces K voisins.

Choix de K :

Le paramètre k est alors déterminé par l'utilisateur avec $1 < K < n$ (avec $n=card(D)$) et c'est la phase la plus délicate de l'algorithme.

Si on choisit K=1 :

On obtient des frontières des classes très complexes avec variance élevée, on risque de sur-ajustement et de ne pas pouvoir standardiser les résultats obtenus

Si on choisit K=n :

On obtient réduction de bruit mais des frontières rigides ce qui rendent les frontières entre classes moins distinctes et par suite une grande probabilité d'erreur pour une nouvelle entrée.

En général, le meilleur choix de K dépend du jeu de donnée. Les grandes valeurs de K réduisent l'effet du bruit sur la classification et risque de sur-apprentissage.

Il convient donc de faire un choix de compromis entre la variabilité associée à une faible valeur de K contre un sur-lissage pour une forte valeur de K. Un bon k peut être sélectionné par diverses techniques heuristiques, par exemple, validation-croisée. Nous choisirons la valeur de k qui minime l'erreur de classification. [11]

Choix de métrique de calcul de distance :

Mesures souvent utilisées pour la distance :

La distance Euclidienne : qui calcule la racine carrée de la somme des différences entre les coordonnées de deux points sur un plan 2D :

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(xi - yi)^2}\ )$$

La distance de Minkowski : de distance générale sur un plan de dimension q.

$$D(x,y) = \sqrt[q]{\sum_{i=1}^{n}(xi - yi)^q}$$

1.4. Avantages et Inconvénients

1.3.1. Avantages

- Apprentissage rapide et incrémental,
- Méthode facile à comprendre,
- Performance asymptotique (n) près de l'optimum.

1.3.2. Inconvénients

- Prédiction lente car il faut revoir tous les exemples à chaque fois,
- Méthode gourmande en mémoire,
- Sensible aux attributs non pertinents et corrélés : bruits,
- Particulièrement vulnérable au fléau de la dimensionnalité.

2. Naive Bayes (NB)

2.1. Description

Naive Bayes est l'une des méthodes les plus simples en apprentissage supervisé. Il est basée sur l'application du théorème de Bayes avec des hypothèses d'indépendance fortes (naïves) entre les caractéristiques. Malgré cette hypothèse forte, ce classifieur s'est avéré très efficace sur de nombreuses applications réelles et est souvent utilisé sur les flux de données pour la classification supervisée. [12]

Le NB nécessite simplement en entrée l'estimation des probabilités conditionnelles par variable P. Il a été largement étudié depuis les années 1950. Il a été introduit sous un nom différent dans la communauté de récupération de texte au début des années 1960 et reste une méthode populaire (de base) pour la catégorisation du texte, le problème de juger les documents appartenant à une catégorie ou à l'autre (comme le spam ou les sports légitimes, les sports Ou politique, etc.) avec des fréquences de mots comme caractéristiques. Avec un prétraitement approprié, il est compétitif dans ce domaine avec des méthodes plus avancées, y compris les machines vectorielles de support. Il trouve également une application dans le diagnostic médical automatique.

2.2. Algorithme

TRAINNB(C ,D)

debut

V= ExtraireVocabulaire (D)

N = ConterDocs(D)

Pour c dans C faire

N_c = ConterDocsDansClasse(D,c)

Prior[c] = $\frac{N_c}{N}$

$text_c$= ConcaténerTextDeDocumentDansClasse(D,c)

Pour t dans V faire

T_{ct}= COUNTTOKENSOFTERM($text_c$, t)

Pour t dans V faire

condprob[t][c] = $\frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$

Retourner (v, prior, condprob)

fin

```
PREDICT(C, V, prior, condprob, d)
        Debut
        W = EXTRACTTOKENSFROMDOC(V, d)
        Pour c dans C faire
                Score[c] = log (prior[c])
                 Pour t dans W faire
                        Score[c] = score[c] + log(condprob[t][c])
        Retourner argmax(c ∈C) (score[c])
        fin
```

2.3. Principe

Appliquer le théorème de Bayes pour définir un algorithme de classification simple et efficace (en pratique)

Rappels sur les probabilités conjointe et conditionnelle :

$$P(X, Y) = P(Y|X)\,P(X) = P(X|Y)\,P(Y)$$

Cette double égalité est à l'origine du théorème de Bayes.

Théorème de Bayes

Transforme la probabilité a priori d'une classe Y en probabilité a posteriori à l'aide des informations contenues dans l'observation X.

Le modèle de bayes peut s'écrire comme suit :

$$P(Y|X) = \frac{p(X|Y)\,P(Y)}{p(x)}$$

Avec *P (y|x)* est la probabilité conditionnelle d'un événement x sachant qu'un autre événement y de probabilité non nulle

- *P (y|x)* croît quand *P(y)* croît : plus y est probable, plus il y a de chances qu'elle soit la classe.
- *P (y|x)* croît quand *P (x|y)* croît : si (x) arrive souvent quand y est la classe, alors il y a des chances que y soit la classe.
- *P (y|x)* décroît quand *P(x)* croît : si (x) est courant, il nous apprend peu sur y.

Le modèle Naïve Bayes consiste à faire l'hypothèse assez forte que les descripteurs xi sont indépendants conditionnellement à la classe, d'où :

$$P(X/Y) = \prod_{i=1}^{p} P(Xi|Y)$$

Donc la formule de bayes va donner :

$$P(Y|X) = \frac{\prod_{i=1}^{p} p(Xi|Y)\, p(Y)}{p(X)} = \frac{\prod_{i=1}^{p} p(Xi|Y)\, p(Y)}{\sum_{y'} P(X|Y')}$$

Le dénominateur *P(X)* sert à normaliser

$$\sum_{y'} P(y'|x) = 1$$

- Donc Le classifieur Bayésien prédit la classe *Y* pour chacune des instances tel que soit maximale la probabilité conditionnelle *P (Y |X).* L'hypothèse naïve dans un classifieur bayésien est de considérer indépendantes les variables explicatives conditionnellement aux classes. [13]

Classification bayésienne : Approche MAP (Maximum A Posteriori)

Cette approche propose de déterminer le meilleur sous-ensemble de variables en maximisant la vraisemblance, pénalisée par un a priori hiérarchique sur les paramètres de sélection du nombre de variables puis sur les sous-ensembles de variables.

Choisir y qui maximise la proba à posteriori :

- *Y= (y1,..., yk)* ensemble de classes (chacune munie d'une probabilité).
- (x) un ensemble d'attributs.
- Retourner la classe ayant la probabilité la plus forte après l'observation de (x).

Hypothèse Maximale A Posteriori: *hMAP*

$hMAP = argmax_{yk \in Y} P(x|yk) \cdot P(yk)/P(x)$

$hMAP = argmax_{yk \in Y} P(x|yk) \cdot P(ck) / P(x)$

P(x) est constant donc $hMAP = argmax_{yk\ Y} P(x|yk) \cdot P(yk)$

2.3. Avantages et Inconvénients

2.3.1. Avantages

- Relativement peu de données d'entraînement pour estimer les paramètres nécessaires à la classification, à savoir moyennes et variances des différentes variables,
- Ce classifieur est souvent utilisé, car très simple d'emploi,
- Computationnellement acceptable,
- marche bien même lorsque l'indépendance conditionnelle des variables prédictives non vérifiée.

2.3.2. Inconvénients

- très sensible à leur corrélation,
- pour les variables continues, le problème d'estimation de densité reste entier,
- Pas de construction de model,
- Génératif : les modèles génératifs fournissent de bons estimateurs lorsque le modèle est "juste", ou en terme statistique bien spécifié, ce qui signifie que le processus qui génère les données réelles

induit une distribution qui est celle du modèle génératif. Lorsque le modèle est mal spécifié (ce qui est de loin le cas le plus courant) on aura intérêt à utiliser une méthode discriminative.

3. Support Vector Machine (SVM)

3.1. Description

SVM est une technique utilisé pour l'apprentissage supervisé, elle repose sur l'idée de Vladimir Vapnik sur le développement d'une théorie statistique de l'apprentissage en 1990.

L'idée générale est de trouver un hyperplan pour classifier les données et maximiser la marge de séparation.

La marge est la distance entre la frontière de séparation et les échantillons les plus proches. Ces derniers sont appelés vecteurs de support. Ce choix est justifié par la théorie de statistique de l'apprentissage, qui montre que la frontière de séparation de marge maximale possède la plus petite capacité. Le problème est de trouver cette frontière séparatrice optimale, à partir d'un ensemble d'apprentissage. [14]

3.2. Principe

Comme on a déjà cité, SVM tente à créer l'hyperplan optimal pour pouvoir classifier les données et prédire la classe d'une nouvelle entrée.

On s'intéressera dans notre projet à la classification linéaire.

En effet, l'hyperplan est une fonction qui a, en entrée, un vecteur x qui fait correspondre à une sortie y, c'est-à-dire $y=f(x)$.
De même, dans le cas de classification linéaire, SVM construit l'hyperplan par combinaison linéaire du vecteur d'entrée $x=(x_1, x_2, \ldots, x_n)^T$:

$$h(x) = w^T x + w_0$$

$h(x)$ est appelé hyperplan séparateur, ou la séparatrice.

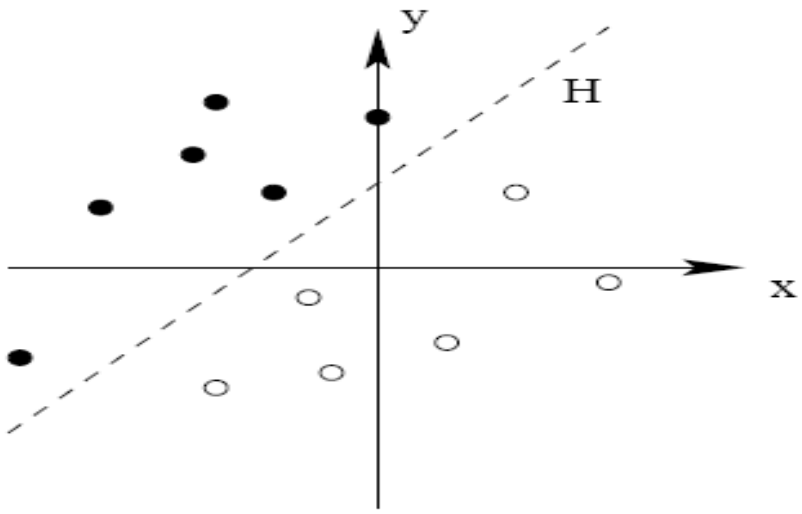

Figure 2 : Séparations d'un ensemble des points par un hyperplan

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

L'algorithme SVM décide alors que x est de classe 1 si h(x)≥0 et de classe -1 sinon.

Même dans le cas de classification linéaire, le choix de l'hyperplan séparateur n'est pas évident, il existe en effet une infinité d'hyperplans séparateurs, dont les performances en apprentissage sont identiques, c'est-à-dire que l'algorithme apprend mais prend des fausses décisions à cause du mauvais choix de l'hyperplan.

Le but est alors de construire h(x) la plus optimal par apprentissage supervisé, pour l'obtenir il faut savoir trouver la marge maximale de discrimination c'est la marge maximale entre les échantillons et l'hyperplan séparateur.

Il existe des raisons théoriques à ce choix. Vapnik a montré que la capacité des classes d'hyperplans séparateurs diminue lorsque leur marge augmente. [15]

$$Arg\ max_{w,w0} min_k\{||x - xk|| : x \in R^N, w^T x + w_0$$

Les échantillons le plus proches au hyperplan séparateur sont appelés vecteurs supports. L'hyperplan qui maximise la marge est donné par :

$$h(x) = w^T x + w_0 = 0$$

Il s'agit donc de trouver w et w_0 remplissant ces conditions, afin de déterminer l'équation de l'hyperplan séparateur :

3.3. Avantages et Inconvénients

3.3.1. Avantages

- Possibilité d'utilisation de structure de données comme les chaines de caractères et arbres comme des entrées,
- traitement des données a grandes dimensions.

3.3.2. Inconvénients

- Demande des données négatives & positives en même temps.

4. La validation croisée

La validation croisée est une méthode d'estimation de fiabilité. Il y a au moins trois techniques de validation croisée :

La première « testset validation » :

Il suffit de diviser l'échantillon en deux sous échantillons, le premier d'apprentissage (communément supérieur à 60 % de l'échantillon) et le second de test. Le modèle est bâti sur l'échantillon d'apprentissage et valide l'échantillon de test. L'erreur est estimée en calculant un test, une mesure ou un score de performance du modèle sur l'échantillon de test.

La seconde « k-fold cross-validation » :

On divise l'échantillon original en x échantillons, puis on sélectionne un des x échantillons comme ensemble de validation et les (x-1) autres échantillons constitueront l'ensemble d'apprentissage. On calcule comme dans la première méthode le score de performance. Puis on répète l'opération en sélectionnant un

autre échantillon de validation parmi les (x-1) échantillons qui n'ont pas encore été utilisés pour la validation du modèle.

L'opération se répète ainsi x fois pour qu'en fin de compte chaque sous-échantillon ait été utilisé exactement une fois comme ensemble de validation. La moyenne des x erreurs quadratiques moyennes est enfin calculée pour estimer l'erreur de prédiction.

La troisième méthode « leave-one-out cross-validation » :

C'est un cas particulier de la deuxième méthode où x=n, c'est-à-dire que l'on apprend sur (n-1) observations puis on valide le modèle sur la énième observation et l'on répète cette opération n fois. [16]

Conclusion

Dans ce chapitre on a spécifié notre travail en expliquant l'apprentissage supervisé en détaillant les trois algorithmes que nous allons utiliser, dans le chapitre suivant on va s'intéressés aux différents outils de développements utilisés pour notre application et expliquer les raisons pour lesquelles on les a choisi.

Partie II : Cadre pratique

Chapitre 1 : Environnement du travail

Introduction

Tout au long des chapitres précédents nous avons introduit le projet, énuméré les étapes nécessaires à sa mise en œuvre, étudier le traitement automatique de la langue et maîtriser les algorithmes supervisé, ce chapitre concerne la description de l'environnement de travail à savoir l'outil de développement et les Framework utilisés.

1. Outils de développement

Python

Python est un langage de programmation, dont la première version est sortie en 1991. Créé par Guido van Rossum, la Python Software Foundation**,** créée en 2001.

C'est est un langage de programmation qui vous permet de travailler plus rapidement et d'intégrer vos systèmes plus efficacement. [17]

Python est un langage à usage général, ce qui signifie qu'il peut être utilisé pour construire n'importe quoi, ce qui sera facilité avec les bons outils / bibliothèques.

Professionnellement, Python est idéal pour le développement web backend, l'analyse de données, l'intelligence artificielle et l'informatique scientifique. De nombreux développeurs ont également utilisé Python pour créer des outils de productivité, des jeux et des applications de bureau, de sorte qu'il existe de nombreuses ressources pour vous aider à apprendre à les faire également. [18]

Django

Django est un framework web haut de gamme Python qui avance une amélioration rapide et un design propre et pragmatique. Travaillé par des concepteurs expérimentés, cela sous-tend une grande partie de la

problématique de l'avancement de la web, afin que vous puissiez vous concentrer sur la composition de votre application sans avoir à redistribuer la roue. C'est gratuit et open source. [19]

Bootstrap

Bootstrap est une bibliothèque gratuite d'avancement open source pour créer des destinations Web et des applications Web. Le Framework Bootstrap est basé sur HTML, CSS et JavaScript (JS) pour rationaliser l'avancement des destinations et des applications réactives et portables. [20]

Bootstrap rend le développement web front-end plus rapide et plus facile. Il est conçu pour les personnes de tous niveaux, des dispositifs de toutes formes et des projets de toutes tailles.

Matplotlib

Matplotlib est une bibliothèque suivante de Python qui permet de distribuer des éléments de qualité dans une variété de formats imprimés et d'environnements interactifs à travers les plates-formes. Matplotlib peut être utilisé comme une partie des scripts Python, du Shell Python et IPython, du jupyter journal, des serveurs d'applications Web et de quatre packs graphiques d'interface utilisateur. [21]

Sklearn

Scikit-learn est une bibliothèque Python gratuite consacrée à l'apprentissage automatique. Il est destiné à se connecter aux bibliothèques informatiques et logiques Python NumPy et SciPy.

Des dispositifs simples et réussis pour l'exploration de données et l'analyse de données.

Il est ouvert à tous et réutilisable dans différents contextes. [22]

NumPy

NumPy est le paquet fondamental pour l'informatique scientifique avec Python. Il contient :

- Un puissant objet de tableau N-dimensionnel
- Fonctions sophistiquées (diffusion)
- Outils pour intégrer le code C / C ++ et Fortran

NumPy peut également être utilisé comme un support multidimensionnel productif d'informations non exclusives. Les types de données discrétionnaires peuvent être caractérisés. Cela permet à NumPy de se coordonner de manière constante et rapide avec un large éventail de bases de données. [23]

SQlite

SQLite est **une** bibliothèque écrite en C. SQLite est parfait pour les petits projets. Sa particularité est d'être intégrée directement à un programme et ne répond donc pas à la logique client-serveur. C'est le moteur de base de données le plus répandu sur la planète car il est intégré à de nombreux logiciels grand public comme, par exemple FireFox, Skype, Adobe, etc.

Pas du tout comme des serveurs de base de données classiques, par exemple, MySQL, sa distinction n'est pas reproduire le schéma de serveur client standard, mais plutôt d'être intégrée directement dans les projets. [24]

Certaines applications peuvent utiliser SQLite pour le stockage de données internes. Il est également possible de prototyper une application à l'aide de SQLite, puis de transmettre le code à une base de données plus grande, comme Oracle.ss.

2-Corpus de données

Le corpus donné est un ensemble de phrase (texte) arabe qui est collecté à partir du web, plus précisément à partir de Twitter. Et la polarité de chacune.

Le tableau suivant présente les caractéristiques du corpus

Nombre de phrases neutres	Nombre de phrases positives	Nombre de phrases négatives	Nombre de mots
805	777	1642	61872

Conclusion :

Dans ce chapitre nous avons présenté les différents outils de développements et les bibliothèques utilisées pour notre application afin de clarifier les raisons pour lesquelles on les a utilisés. Dans le chapitre suivant nous allons aborder la dernière partie qui représente la partie de réalisation, ou on va présenter les tests, les résultats et les comparaisons qu'on a fait ainsi que la démo du site web qu'on a créé.

Chapitre 2 : Réalisation

Introduction

Dans le chapitre précédant on a précisé les différents outils ainsi que le corpus de données utilisés pour la réalisation du projet, dans ce chapitre on présente les différents les tests, les résultats et les comparaisons qu'on a réalisé puis on va présenter les différentes interfaces du site qu'on a créé.

1. Etude et comparaison

1.1. Etude

Le but de cette partie est de tester réellement les différents algorithmes et essayer d'améliorer leurs fiabilités avant de les implémenter dans une application.

1.1.1. CountVectorizer

1.1.1.1. Définition

L'analyse de texte est un domaine d'application majeur pour les algorithmes d'apprentissage par machine. Cependant, les données brutes, une séquence de symboles ne peuvent pas être directement alimentée par les algorithmes directement, car ils prévoient des vecteurs de caractéristiques numériques avec une taille fixe plutôt que des documents de texte brut à longueur variable.

CountVectorizer est un module utilisé pour extraire des fonctionnalités dans un format supporté, il compte les occurrences des mots dans chaque document (ou phrase). [25]

1.1.1.2. Comparaison par moyenne de précision

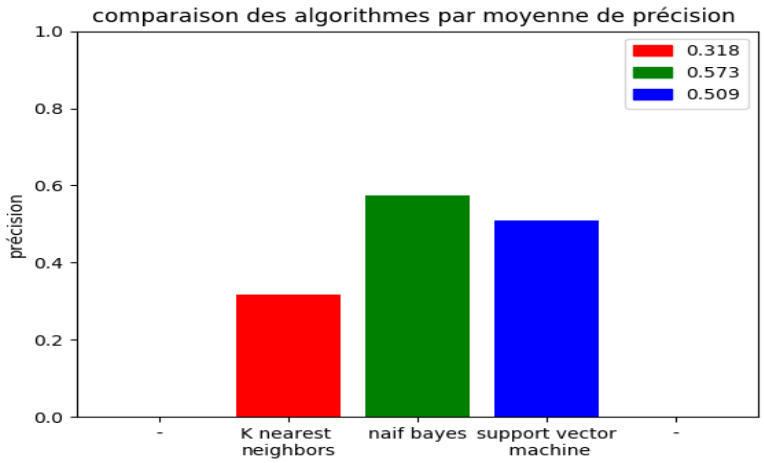

Figure 3 Countvectorizer : Comparaison par moyenne de précision

La figure 3 présente la moyenne de précision par validation croisé de chaque algorithme.

K Nearest Neighbors a un très faible score qui atteint 31,8%

Naive Bayes a un score de 57,3%

Support Vector Machine a eu le score de 50,9%

1.1.1.3- Comparaison par validation croisée

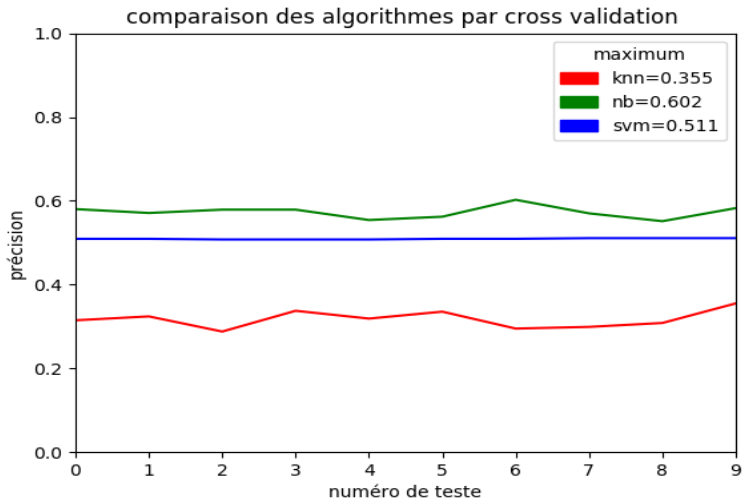

Figure 4 : Countvectorizer : Comparaison par validation croisée

La figure 4 présente la variation des scores des algorithmes à chaque test de la validation croisée.

La courbe en rouge présente les scores de knn qui sont très faible et ont un maximum de 35,5%,

La courbe en vert présente les scores de Naive Bayes de maximum 60,2%,

La courbe en bleu présente les score de SVM on remarque qu'elle est constante a un score de 51,1%.

1.1.2. TfidfVectorizer

1.1.2.1. Définition

TfidfVectorizer (Term Frequency-Inverse Document Frequency) est une méthode statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus. [26]

1.1.2.2. Comparaison par moyenne de précision

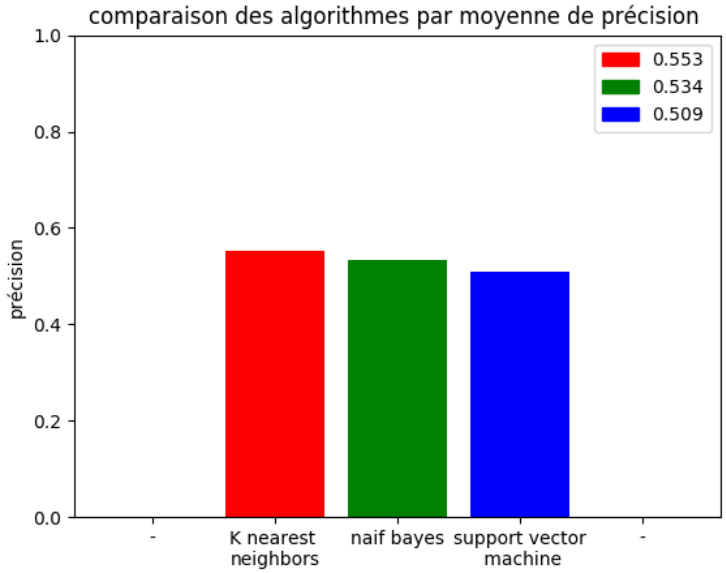

Figure 5 :Tfidf comparaison par moyenne de précision

La figure 5 présente la moyenne de précision par validation croisé de chaque algorithme.

K Nearest Neighbors a un score de 55,3%

Naive Bayes a un score de 53,4%

Support Vector Machine a eu le score de 50,9%

1.1.2.3. Comparaison par validation croisée

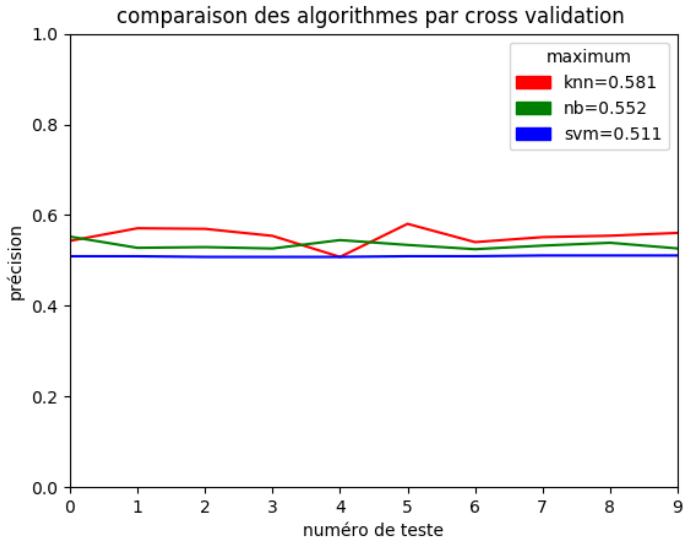

Figure 6 : TfidfVectorizer : Comparaison par validation croisée

La figure 6 présente la variation des scores des algorithmes à chaque test de la validation croisée après l'utilisation de TfidfVectorizer.

La courbe en rouge présente les scores de KNN et ont un maximum de 58,1%, les scores sont plus grande et plus stable qu'avec CountVectorizer,

La courbe en vert présente les scores de Naive Bayes de maximum 55,2%.

La courbe en bleu présente les scores de SVM on remarque qu'elle est constante a un score de 51,1%.

On remarque le croisement des scores knn et Naive Bayes

1.1.3. Optimisation

1.1.3.1. Introduction

On a essayé d'optimiser les résultats obtenus par chaque algorithme.

Tout d'abords on a varié le nombre de voisin pris en charge par l'algorithme KNN.

Ensuite on a testé les différents kernel de SVM,

Poly : le noyau polynomial représente la similarité des vecteurs (échantillons d'apprentissage) dans un espace de degré polynomial plus grand que celui des variables d'origine, ce qui permet un apprentissage de modèles non-linéaires.

Intuitivement, il ne tient pas compte uniquement des propriétés des échantillons d'entrée afin de déterminer leur similitude, mais des combinaisons aussi. [27]

Rbf : le noyau a base fonction radial est utilisé à la classification des données non linéaire il utilise une fonction radiale [28], c'est-à-dire une fonction définie sur un espace euclidien R^n dont la valeur à chaque point dépend uniquement de la distance entre ce point et l'origine [29]

Linear : le noyau linéaire utilise une fonction linéaire pour créer l'hyperplan, une fonction linéaire est écrite de la forme

$$\vec{w}.\vec{x} - b = 0$$

Et enfin on a testé notre corpus sur les deux types de Naive Bayes

Multinomial : Multinomial Naive Bayes nous permet simplement de savoir que chaque p (fi | c), p (fi | c) est une distribution multinomiale, plutôt qu'une autre distribution. Cela fonctionne bien pour les données qui peuvent facilement être transformées en compte, comme le nombre de mots dans le texte [30].

La distribution multinomiale est une généralisation de la distribution binomiale à plus de deux catégories.

Bernoulli : les caractéristiques sont des booléens indépendants (variables binaires) décrivant les entrées. Ce modèle est populaire pour les tâches de classification des documents, où les fonctions d'occurrence de terme binaire sont utilisées plutôt que des fréquences de termes. [31]

1.1.3.2. KNN : variation de k le nombre de voisins

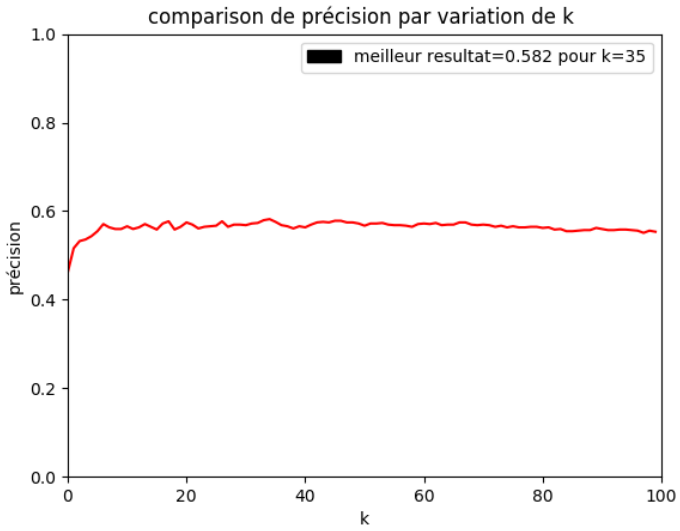

Figure 7 : Optimisation : variation de nombre de voisins

La figure 7 présente les scores de KNN en variant le nombre de voisin considérés par l'algorithme entre 1 et 100.

Le meilleur score est donné à K = 35 a score =58,2%.

1.1.3.3. SVM : changement de kernel

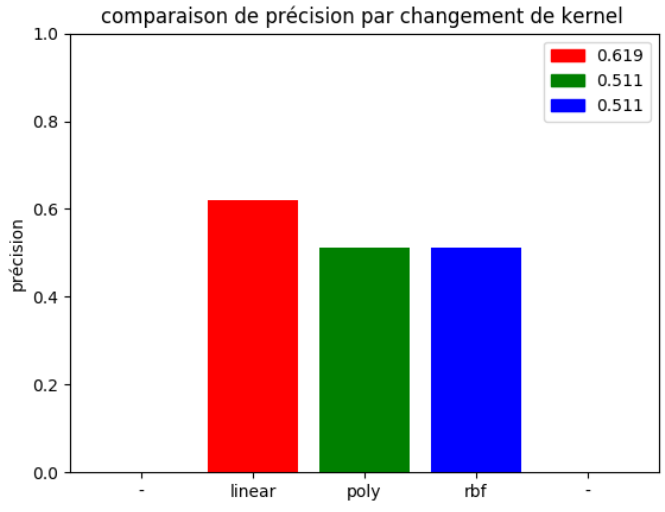

Figure 8 : Optimisation : changement de kernel

La figure 8 présente la moyenne des scores obtenus par SVM avec les différents noyaux.

SVM avec noyau linéaire a eu la moyenne la plus élevée de 61,9%.

SVM avec noyau binomial et avec noyau a base fonction radiale ont eu le même score de 51,1%.

Ceci est expliqué par la linéarité des données qui sont des textes.

1.1.3.4. NB : Multinomial et Bernoulli

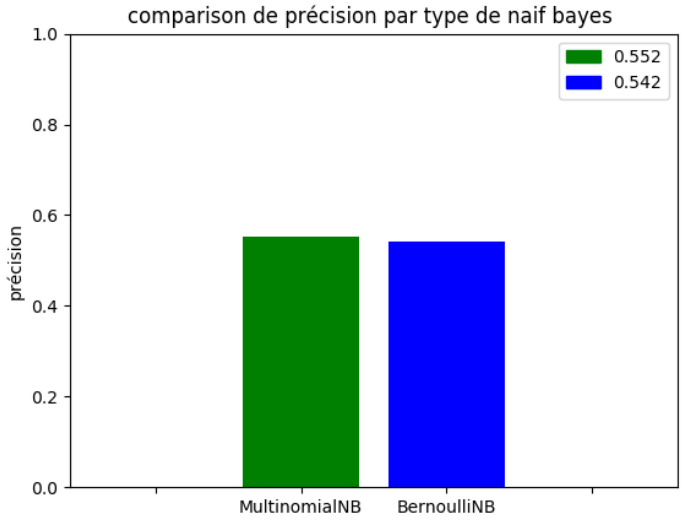

Figure 9 : Optimisation changement de type de Naive Bayes

La figure 9 présente le score moyen par validation croisée de Multinomial Naive Nayes et Bernoulli Naive Bayes.

Multinomial Naive Bayes a eu une moyenne des scores la plus élevée, elle est de 55,2%.

Bernoulli Naive Bayes a eu un score de 54,2%.

1.1.3.5. Comparaison par moyenne de précision

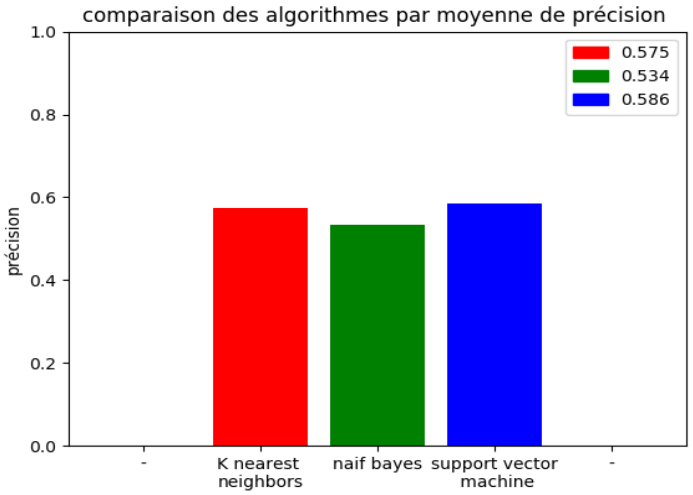

Figure 10 : Résultats de l'optimisation : comparaison par moyenne de précision

La figure 10 présente la moyenne de chacun des algorithmes après l'optimisation avec la validation croisée.

K Nearest Neighbors a une moyenne de 57,5%.

Naive Bayes a une moyenne de 53,4%.

Support Vector Machine a eu la moyenne le plus élevée de 58,6%.

1.1.3.6. Comparaison par validation croisée

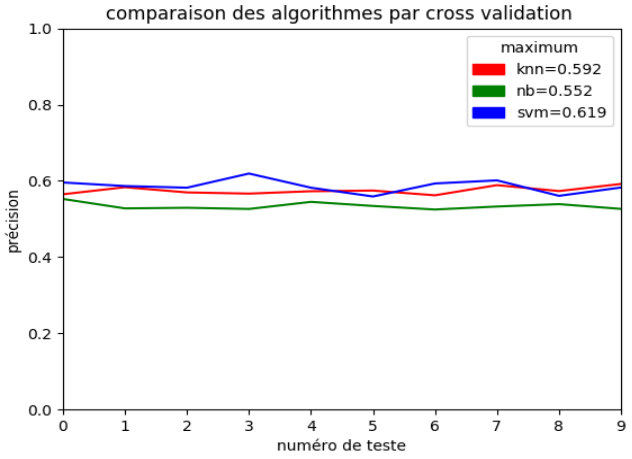

Figure 11 : Résultats de l'optimisation : comparaison par validation croisée

La figure 5 présente la variation des scores des algorithmes à chaque test de la validation croisée après l'optimisation.

La courbe en rouge présente les scores de KNN qui ont un maximum de 59,2%, les scores sont plus grande et plus stable qu'avec CountVectorizer,

La courbe en vert présente les scores de Naive Bayes de maximum 55,2%.

La courbe en bleu présente les scores de SVM on remarque qu'elle est constante a un score de 61,9%.

On remarque le croisement des scores KNN et SVM.

1.1.4. Racinisation

1.1.4.1. Définition

La racination (Stemmer) est une procédure régulière dans les applications de traitement automatique du langage naturel, elle sert à changer des mots (flexions) à leurs racines.

La base d'un mot se compare à la partie du reste du mot une fois que ses préfixes et postfixes ont été évacués, pour être spécifique à sa racine. [32]

1.1.4.2. Comparaison par moyenne de précision

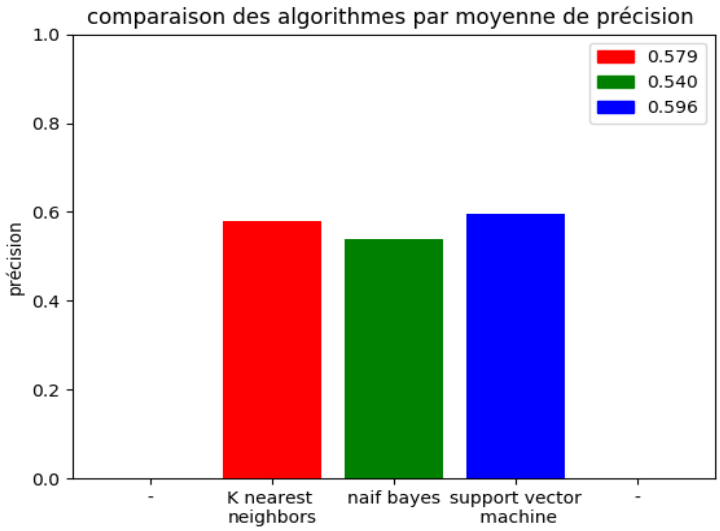

Figure 12 : Racination : comparaison par moyenne de précision

La figure 9 présente la moyenne de chacun des algorithmes après la racination des données avec la validation croisée.

K Nearest Neighbors a une moyenne de 57,9%.

Naive Bayes a une moyenne de 54%.

Support Vector Machine a eu la moyenne le plus élevée de 59,6%.

1.1.4.3. Comparaison par validation croisée

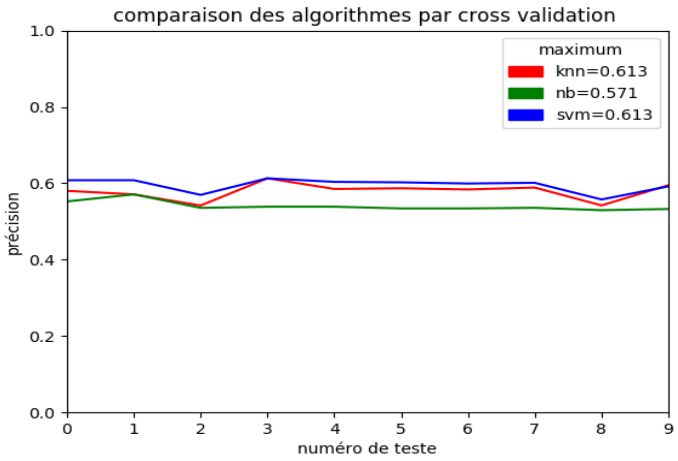

Figure 13 : Racination : comparaison par validation croisée

La figure 5 présente la variation des scores des algorithmes à chaque test de la validation croisée après la racination des données.

La courbe en rouge présente les scores de knn et ont un maximum de 61,3%, les scores sont plus grande et plus stable qu'avec CountVectorizer,

La courbe en vert présente les scores de Naive Bayes de maximum 57,1%.

La courbe en bleu présente les scores de SVM on remarque qu'elle est constante a un score de 61,3%.

1.2. Comparaison des algorithmes

Ce tableau présente les différentes moyennes de score obtenues par les trois algorithmes dans chacun des testes

	CountVectorizer	TfidfVectorizer	optimisation	Racination
KNN	31,8%	55,3%	57,5%	57,9%
NB	57,3%	53,4%	53,4%	54%
SVM	50,9%	50,9%	58,6%	59,6%

3. Démo web

3.1. Présentation

Le site web qu'on a créé a comme but de prédire la polarité d'une phrase entrée par l'utilisateur.

L'utilisateur aura la possibilité de choisir l'algorithme et de varier les paramètres.

La phrase est envoyer ensuite au serveur pour qu'il applique la prédiction de sa polarité et la renvoi à l'utilisateur dans un « modal ».

Le site est disponible en ligne sur le lien http://mohameddhifli.pythonanywhere.com

3.2. Démonstration

3.2.1. Page d'accueil

Dans la page d'accueil on a présenté notre projet en détaillant les algorithmes utilisés et les outils logiciels

Figure 14 : Page d'accueil

3.2.1. La page de l'application

Dans la page application on a donné à l'utilisateur le formulaire avec lequel il peut utiliser les fonctionnalités du site.

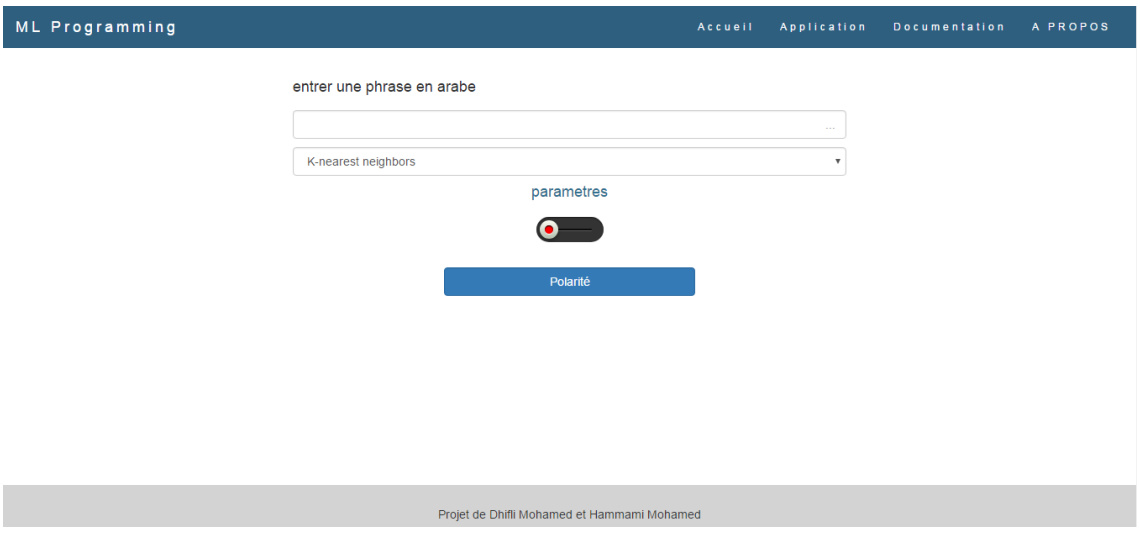

Figure 15 :La page application

- ✓ On a donné à l'utilisateur la possibilité d'utiliser la racination et de donner les paramètres des algorithmes :
- ✓ Pour KNN le changement de nombre de voisins

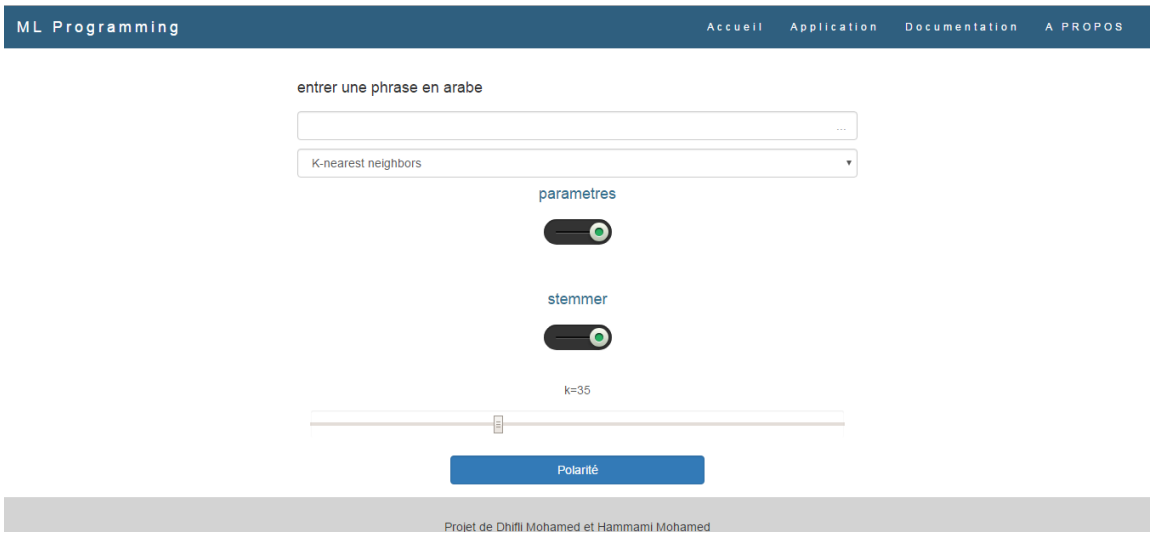

Figure 16 : Page application variation de k

- ✓ Pour Naive Bayes le changement de type (Multinomial ou Bernoulli)

Figure 17 : Page application changement de type de naive bayes

✓ Pour SVM le changement de noyau

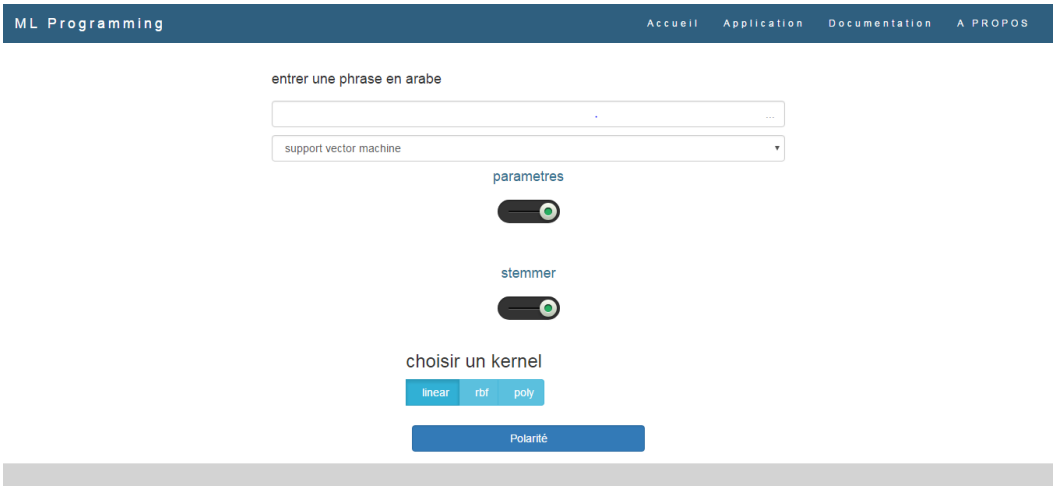

Figure 18 : Page application changement de kernel de SVM

Quand l'utilisateur entre une phrase et clique sur « polarité » le serveur reçoit une requête qui contient la phrase, l'algorithme à utiliser et ses paramètres.

✓ Le serveur applique la prédiction de la polarité et renvoi le résultat à l'utilisateur

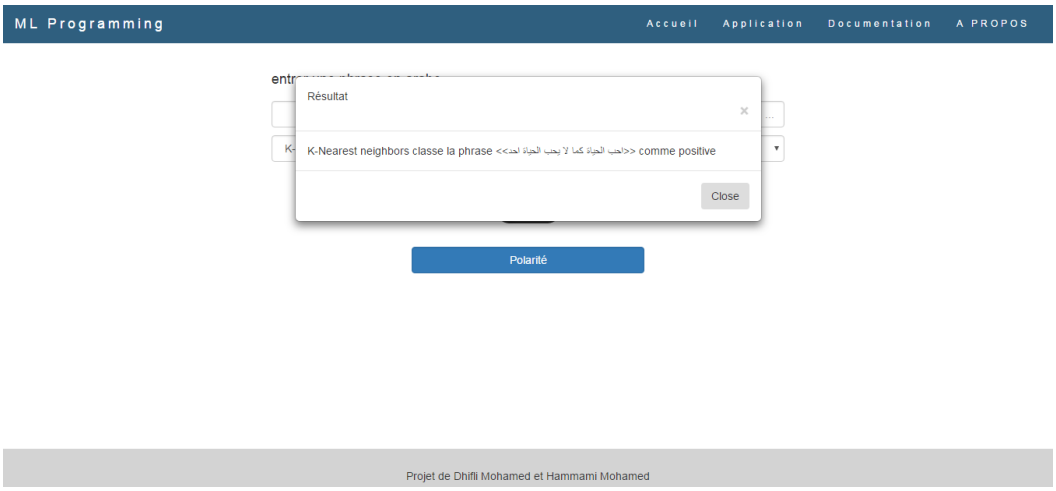

Figure 19 : Retour de resultat (polarité de la phrase)

Conclusion

Dans ce chapitre on a présenté les tests, les résultats et les comparaisons que nous avons fait ainsi que la démo du site web qu'on a créé.

Conclusion générale

Dans le cadre de notre projet de fin d'étude, nous avons eu la chance d'intégrer une unité de recherche spécialisée dans le traitement automatique des langues naturelles (TALN) et plus précisément l'Analyse des Sentiments.

Ce projet nous a fournis de consolider notre formation surtout en matière de Machine Learning et d'intelligence artificielle.

L'objectif de ce projet est de contribuer à la réalisation d'un système d'analyse de sentiment pour la langue arabe. Le travail présenté dans ce rapport, a pour vocation de servir de base à l'exploration poussée du domaine. Les techniques d'analyse des sentiments ont rendu possible de nos jours d'analyser les tendances comportementales des internautes dans plusieurs domaines.

Après avoir réalisé un état de l'art sur le domaine, nous avons fait plusieurs expérimentations et après les analyses des résultats, nous concluons que :

L'utilisation des deux algorithmes de classifications : Support Vector Machine (SVM) et K-Nearest Neighbors (KNN), pour la détection de la polarité des avis en langue arabe, donne des performances plus élevées que l'utilisation de l'algorithme Naive Bayes (NB). Nous avons atteint deux précisions qui sont très proches tout en appliquant des combinaisons différentes de prétraitement. .

La deuxième partie de notre rapport nous l'avons consacré à la réalisation d'une application web qui permet l'utilisateur de tester à lui-même notre modèle de classification de polarité sur des avis qu'il peut proposer lui-même. Pendant la réalisation, nous avons utilisé une panoplie de technologies (Python, Bootstrap, Django…).

Bibliographie

[1] «intelligence artificielle,» [En ligne]. Available: http://www.intelligenceartificielle.fr/. [Accès le 02 04 2017].

[2] «Intelligence_Artificielle,» [En ligne]. Available: https://wiki.labomedia.org/index.php/Intelligence_Artificielle. [Accès le 02 04 2017].

[3] «DomaineIA,» [En ligne]. Available: http://tpe-intelligence-artificielle-2013.e-monsite.com/pages/definition-de-l-intelligence-artificielle.html. [Accès le 02 04 2017].

[4] d. F. Fuchs & B. Habert, «le traiement automatique de la langue,» p. 7, 2004.

[5] «Analyse_de_sentiments_en_text_mining,» [En ligne]. Available: http://edutechwiki.unige.ch/fr/Analyse_de_sentiments_en_text_mining. [Accès le 14 04 2017].

[6] «opinion mining et sentiment analysis,» [En ligne]. Available: http://books.openedition.org/oep/214?lang=fr. [Accès le 02 04 2017].

[7] «Apprentissage_non_supervisé,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Apprentissage_non_supervis%C3%A9. [Accès le 14 04 2017].

[8] «apprentissage supervisé,» [En ligne]. Available: https://tel.archives-ouvertes.fr/tel-00465008/document. [Accès le 14 04 2017].

[9] H. Nonne, «Une-petite-histoire-du-Machine-Learning,» 28 10 2015. [En ligne]. Available: https://www.quantmetry.com/single-post/2015/10/28/Une-petite-histoire-du-Machine-Learning. [Accès le 02 04 2017].

[10] «K-nearest_neighbors_algorithm,» [En ligne]. Available: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm. [Accès le 15 04 2017].

[11] L. wassim, «algorithme-knn,» [En ligne]. Available: https://fr.slideshare.net/wassimlahbibi/algorithme-knn. [Accès le 9 04 2017].

[12] «Naive_Bayes_Classifier_Explained,» [En ligne]. Available: https://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_Naive_Bayes_Classifier_Explained.pdf. [Accès le 12 04 2017].

[13] «naive bayes,» [En ligne]. Available: https://www.di.ens.fr/~fbach/courses/fall2010/cours9.pdf. [Accès le 10 04 2017].

[14] «SVM,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Machine_%C3%A0_vecteurs_de_support. [Accès le 05 04 2017].

[15] «SVM,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Machine_%C3%A0_vecteurs_de_support. [Accès le 05 04 2017].

[16] H. L. Lei Tang, «Validation_crois%C3%A9e,» 07 11 2013. [En ligne]. Available: https://fr.wikipedia.org/wiki/Validation_crois%C3%A9e. [Accès le 02 04 2017].

[17] «python,» [En ligne]. Available: https://www.python.org/. [Accès le 01 04 2017].

[18] «why-learn-python,» [En ligne]. Available: http://www.bestprogramminglanguagefor.me/why-learn-python. [Accès le 05 04 2017].

[19] «django,» [En ligne]. Available: https://www.djangoproject.com/. [Accès le 17 04 2017].

[20] «bootstrap,» [En ligne]. Available: http://whatis.techtarget.com/definition/bootstrap. [Accès le 15 04 2017].

[21] «matplotlib,» [En ligne]. Available: https://matplotlib.org/. [Accès le 20 04 2017].

[22] «Scikit-learn,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Scikit-learn. [Accès le 15 04 2017].

[23] «numpy,» [En ligne]. Available: http://www.numpy.org/. [Accès le 11 04 2017].

[24] «SQLite,» [En ligne]. Available: https://fr.wikipedia.org/wiki/SQLite. [Accès le 03 04 2017].

[25] «countvectorizer,» [En ligne]. Available: http://scikit-learn.org/stable/modules/feature_extraction.html. [Accès le 20 04 2017].

[26] «TfidfVectorizer,» [En ligne]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. [Accès le 04 20 2017].

[27] «Noyau_polynomial,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Noyau_polynomial. [Accès le 25 04 2017].

[28] «fonction radiale,» [En ligne]. Available: https://en.wikipedia.org/wiki/Radial_function. [Accès le 27 04 2017].

[29] «SVM RBF,» [En ligne]. Available: https://en.wikipedia.org/wiki/Radial_basis_function_kernel. [Accès le 25 04 2017].

[30] «multinomial naive bayes,» [En ligne]. Available: http://scikit-learn.org/stable/modules/naive_bayes.html. [Accès le 20 04 2017].

[31] «Bernoulli_naive_Bayes,» [En ligne]. Available: https://en.wikipedia.org/wiki/Naive_Bayes_classifier. [Accès le 16 04 2017].

[32] «Racinisation,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Racinisation. [Accès le 2017 04 29].