كلية العلوم الرباط
Faculté des sciences Rabat

جامعة محمد الخامس بالرباط
Université Mohammed V de Rabat

Classificateur d'email à base de Naive Bayes

NLP Project

Mastère Spécialisé IDDLo

Encadré par : Pr .Mahmoudi

Rédigé par : ELHADAF Abdelouahab

EDDAHIR Zakaria

Table des matières

1/ Introduction

Le traitement naturel du langage, ou Natural Language Processing (NLP) en anglais, est une technologie d'intelligence artificielle visant à permettre aux ordinateurs de comprendre le langage humain.

Le NLP peut être divisé en 2 grandes parties, le NLU (Natural Language Understanding) et le NLG (Natural Language Generation):

- Le premier est toute la partie « compréhension » du texte, prendre un texte en entrée et pouvoir en ressortir des données.
- Le second, est générer du texte à partir de données, pouvoir construire des phrases cohérentes de manière automatique.

Le but de ce mini projet est d'utiliser les techniques et algorithmes du NLP pour prédire et classer un mail comme spam ou not spam. Ce travail est basé sur l'algorithme Naïve bayes.

2/ Projet : Mail clissifier

A/ Définition

Notre projet consiste en la réalisation d'un programme qui peut classifier les mails comme spam ou pas spam.

Le programme se décompose :

- Data Cleaning : il consiste à faire donner au programme une dataset de presque de 3000 mail catégorisé comme spam et non spam.

- Entrainement de l'algorithme : avec la fonction *train()* on va faire entrainer le programme avec le dataset fourni à l'avance.
- Exécution du module sur des emails : finalement la fonction *classify(email)* va permettre de classer les mails en spam ou non spam.

B/ Principe de l'algorithme Naive Bayes

Naive Bayes Classifier est un algorithme populaire en Machine Learning. C'est un algorithme du Supervised Learning utilisé pour la classification. Il est particulièrement utile pour les problématiques de classification de texte. Un exemple d'utilisation du Naive Bayes est celui du filtre anti-spam.

Regardons de plus prés comment fonctionne cet algorithme.

Le naive Bayes classifier se base sur le théorème de Bayes. Ce dernier est un classique de la théorie des probabilités. Ce théorème est fondé sur les probabilités conditionnelles.

Probabilités conditionnelles : Quelle est la probabilité qu'un événement se produise sachant qu'un autre événement s'est déjà produit.

Exemple :

Supposons qu'on ait une classe de lycéens. Soit A et B les deux événements suivants:

L'événement A : l'élève est une fille.

L'événement B : l'élève pratique l'allemand.

Quelle est la probabilité qu'on choisisse au hasard une fille pratiquant l'allemand ?

Le théorème de Bayes permet de calculer ce genre de probabilité.

Notons P la probabilité d'un événement.

P(eleve \ est \ une \ fille \ ET \ eleve \ pratique \ allemand) = P(eleve \ est \ une \ fille) * P(eleve \ pratique \ allemand \mid eleve \ est \ une \ fille)

Cela équivaut à calculer ;

P(eleve \ est \ une \ fille \ ET \ eleve \ pratique \ allemand) = P(eleve \ pratique \ allemand) * P(eleve \ est \ une \fille \mid eleve \ pratique \ allemand)

Le terme P(A | B) se lit : la probabilité que l'événement A se réalise sachant que l'événement B s'est déjà réalisé.

On appelle le terme A : l'évidence (faux ami avec le mot anglais Evidence). Le terme B s'appelle Outcome.

a/ Théorème de Bayes

Reprenons l'exemple des lycéens pratiquant l'allemand. Imaginons le jeu de données suivant :

	Fille (A)	Garçon ($\neg A$)	Totaux
Allemand (B)	10	7	17
Autre langue ($\neg B$)	4	9	13
Totaux	14	16	30

Calculons la probabilité suivante : Quelle est la probabilité qu'on tire au hasard un élève parlant Allemand sachant qu'elle est une fille ?

Selon la formule de Bayes on a :

$P(Allemand \mid Fille) = P(B \mid A) = \frac{P(B \cap A)} {P(A)} = \frac{P(A \mid B) * P(B)}{P(A)}$

P(A) est la probabilité de prendre au hasard une fille de la population des élèves de la classe. On appelle P(A) la probabilité antérieure (prior probability).

$P(A) = \frac{cardinal(A)}{cardinal(\Omega)} = \frac{14}{30} \approx 0.4666$

$P(B \cap A) = \frac{cardinal(B \cap A)}{cardinal(\Omega)} = \frac{10}{30} \approx 0.3333$

Ce qui donne :

$P(B \mid A) = \frac{\frac{10}{30}}{\frac{14}{30}} \approx \frac{0.3333}{0.4666} \approx 0.7143$

Note: le cardinal d'un ensemble est le nombre d'éléments dans ce dernier cardinal (\Omega) représente l'ensemble des lycéens de notre exemple (l'univers de probabilités)

b /Naive Bayes Classifier

Lors de l'exemple précédent, nous avons appliqué le théorème de Bayes avec une seule variable prédictive (Evidence) : A savoir le sexe de l'élève. Dans les vraies applications du Naive Bayes, on calcule le résultat (Outcome) en se basant sur plusieurs variables. L'application du théorème de Bayes sur plusieurs variables rend le calcul complexe. Pour contourner cela, une approche consiste à prendre en considération ces variables indépendamment les unes des autres. Il s'agit d'une hypothèse forte. Généralement, les variables prédictives sont liées entre elles. Le terme "naïve" vient du fait qu'on suppose cette indépendance des variables.

Naïve Bayes est particulièrement prisé dans les problèmes de classification de texte

Si on prend l'exemple de classification de mails en \{SPAM, NON SPAM\}, Naïve Bayes se basera sur la fréquence d'occurrence des mots dans le mail pour en définir

la catégorie. Lors de sa classification, l'algorithme supposera que les mots du mail "apparaissent" indépendamment les uns des autres. Évidemment, d'un point de vue linguistique et sémantique, cette supposition est fausse !

c/ Avantages et limitations du Naive Bayes Classifier

Avantages

En se basant sur l'exemple de classification des fruits, on remarque plusieurs avantages pour cet algorithme :

Le Naive Bayes Classifier est très rapide pour la classification : en effet les calculs de probabilités ne sont pas très coûteux.

La classification est possible même avec un petit jeu de données .

Inconvénients

l'algorithme Naive Bayes Classifier suppose l'indépendance des variables : C'est une hypothèse forte et qui est violée dans la majorité des cas réels.

Contre intuitivement, malgré la violation de la contrainte d'indépendance des variables, Naïve Bayes donne de bons résultats de classification. La publication de Harry Zhang donne une explication sur cette performance contre intuitive

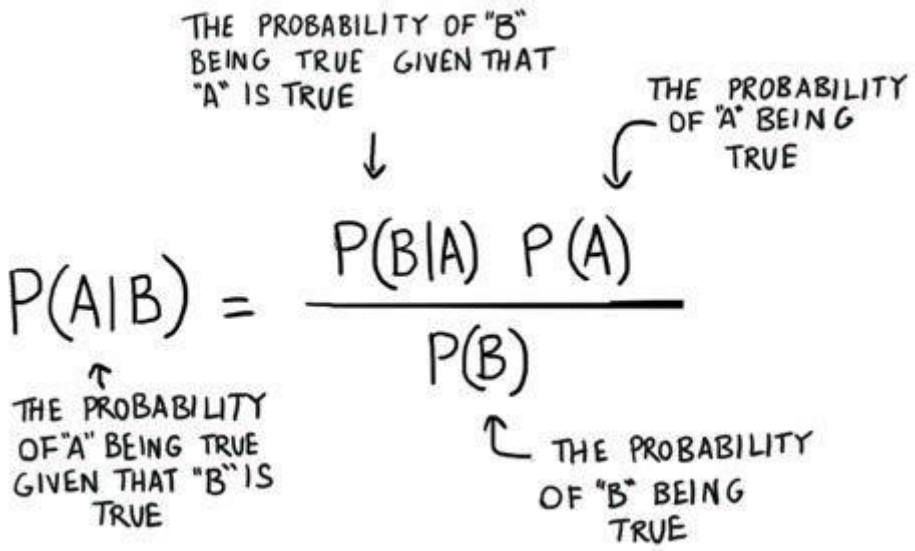
d/ Comment fonctionnent les filtres anti-spam ?

La plupart des filtres anti-spam sont basés sur le théorème bayésien. Il vous permet de trouver la probabilité d'un événement, à condition qu'un autre événement se soit produit. Ceci est symboliquement exprimé par P(A|B), c'est-à-dire la probabilité que l'événement A se produise étant donné que l'événement B s'est déjà produit. Dans notre cas, un événement est la probabilité de rencontrer un texte. c.-à-d. la condition « le texte soit un spam ».

Les raisons d'utiliser le théorème de Naive Bayes :

- Le théorème de Bayes fournit une perspective utile pour comprendre et évaluer de nombreux algorithmes d'apprentissage.
- Il calcule des probabilités explicites pour l'hypothèse et il est robuste au bruit dans les données d'entrée.

Dans la classification statistique, cela minimise la probabilité d'erreur de classification.



• P(A) - probabilité de spam

• P(B) - probabilité du texte

• P(A|B) - probabilité que le texte soit du spam

• P(B|A) - probabilité de rencontrer ce texte parmi les spams

e/ Comment compter ces probabilités ?

Le classificateur bayésien est basé sur l'hypothèse que les attributs d'une classe sont indépendants les uns des autres. En réalité, presque tous les attributs de quoi que ce soit dépendent les uns des autres, et il est très naïf de l'admettre. Mais avec la classification des spams, l'algorithme bayésien est très efficace (et ce malgré le fait que les mots dans le texte dépendent toujours les uns des autres). En effet, pour déterminer le caractère « spam » d'un texte, nous fonctionnons sur des probabilités. Et si nous croyons que ces probabilités sont indépendantes, nous avons une arme très puissante - les probabilités indépendantes peuvent être multipliées !

Pour calculer la probabilité qu'une phrase donnée soit du spam, nous calculons la probabilité de spam pour chaque mot de la phrase, puis multiplions simplement ces probabilités. Ainsi, la probabilité qu'un texte donné pour être spam est le produit de la probabilité de spam et de la probabilité de rencontrer un texte donné parmi l'ensemble de toutes les variantes de spam possibles.

Ainsi, on calcule

$$P(B|A) = \prod_i P(B_i|A)$$

- $\prod_i$ - product of several elements,
- B - one sentence,
- B_i - one word in sentense.

Et puis calculons et comparons les valeurs des expressions. Cette inégalité nous montre qu'une phrase donnée est spam :

$$P(A|B) > P(\overline{A}|B)$$

- $P(\overline{A}|B)$ - probabilité que la phrase donnée ne soit pas du spam.

Ou nous pourrions réécrire la formule :

$$P(A) \cdot P(B|A) > P(\overline{A}) \cdot P(B|\overline{A})$$

C/ Outils de travail

Anaconda est une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion

des paquets et de déploiement. Les versions de paquetages sont gérées par le système de gestion de paquets anaconda. La Distribution Anaconda est utilisée par plus de 6 millions d'utilisateurs, et il comprend plus de 250 paquets populaires en science des données adaptés pour Windows, Linux et MacOS.

Jupyter est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Julia, Python, R, Ruby ou encore Scala. Jupyter permet de réaliser des notebooks, c'est-à-dire des programmes contenant à la fois du texte en markdown et du code en Julia, Python, R... Ces notebooks sont utilisés en science des données pour explorer et analyser des données.

Un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d’exceptions.

3/ Conclusion

Dans notre projet on utiliser la partie analyse des sentiments du NLP pour trainer un model est classier les mails, puis prédire les catégories des textes fournies en input.

Ce projet était très bénéfique pour nous pour pratiquer les connaissances déjà acquises dans le cours.

Il nous a permet aussi d'étudier et d'analyser le comportement des filtres anti-spam existant dans le marché.

On veut par l'occasion remercier Mr Mahmoudi de nous avoir donner cette chance et pour tous les efforts fournis pour réussir cette étape.