**Faculté d'Informatique** **SII 2021/2022**
**Département** Informatique **Module** AARN

# Projet de fin de semestre

**Introduction**

De nombreux services de messagerie fournissent aujourd'hui des filtres anti-spam capables de classer les e-mails dans les spams et les non-spams avec une grande précision. Pour ce projet, l'objectif est d'utiliser les notions apprises durant ce semestre pour résoudre un problème de sciences de données. Les étudiants en groupe entre 1 et 4 étudiants devront implémenter un détecteur de Spam en apprenant sur un ensemble de données.

Le classificateur devra classer si un e-mail donné, x, en spam (y = 1) ou non-spam (y = 0).

Chaque email devra être converti en un vecteur de caractéristiques $x \in R^n$. L'ensemble de données utilisé est SpamAssassin Public Corpus[1].2 Pour les besoins de ce projet, il est préférable d'utiliser uniquement le corps de l'e-mail (après nettoyage).

**Étapes à réaliser :**

**Étape 1 : préparation des données**

Avant de commencer une tâche d'apprentissage automatique, il est généralement judicieux de jeter un œil à des exemples de l'ensemble de données. La Figure 1 montre un exemple d'e-mail contenant une URL, une adresse e-mail (à la fin), des chiffres et des montants en dollar. Bien que de nombreux e-mails contiennent des types d'entités similaires (par exemple, numéros, autres URL ou autres adresses e-mail), les entités spécifiques (par exemple, l'URL spécifique ou un montant spécifique) sera différent dans presque chaque e-mail. Par conséquent, une méthode souvent employée dans le traitement des e-mails consiste à « normaliser » ces valeurs, afin que toutes les URL soient traitées de la même manière, tous les nombres sont traités de la même manière, etc. Par exemple, nous pourrions remplacer chaque URL dans l'email avec la chaîne unique "httpaddr" pour indiquer qu'une URL était présente.

[1] https://spamassassin.apache.org/old/publiccorpus/

```
> Anyone knows how much it costs to host a web portal ?
>
Well, it depends on how many visitors youre expecting.  This can be
anywhere from less than 10 bucks a month to a couple of $100.  You
should checkout http://www.rackspace.com/ or perhaps Amazon EC2 if
youre running something big..

To unsubscribe yourself from this mailing list, send an email to:
groupname-unsubscribe@egroups.com
```

*Figure 1. Exemple d'un email*

Cela a pour effet de laisser le classificateur de spam prendre une décision de classification en fonction de la présence ou non d'une URL, plutôt que d'une URL spécifique étant présente. Cela améliore généralement les performances d'un classificateur de spam, étant donné que les spammeurs randomisent souvent les URL, et donc les chances de voir n'importe quel URL particulière à nouveau dans un nouveau spam est très petite.

A cet effet, la normalisation ou nettoyage qui doit être implémentée dans les étapes suivantes.

**Nettoyage d'emails**

**Minuscule** : l'intégralité de l'e-mail devra être convertie en minuscules.

**Suppression de balises HTML** : Toutes les balises HTML devront être supprimées des e-mails. De nombreux e-mails sont souvent accompagnés d'un formatage HTML ; toutes les Balises HTML devront être supprimées, de sorte que seul à garder uniquement le contenu de l'email.

**Normalisation des URL** : Toutes les URL devront être remplacées par le texte « httpaddr ».

**Normalisation des adresses e-mail** : toutes les adresses e-mail devront être remplacées avec le texte "emailaddr".

**Normalisation des nombres** : Tous les nombres devront être remplacés par le texte "nombre".

**Normalisation des dollars** : Tous les signes dollar ($)devront être remplacés par le texte "dollar".

**Radicalisation de mots** : Les mots devront être réduits à leur forme radicale. Par exemple, "discount", "discounts", "discounted" et "discounting" devront être tous remplacé par " discount", et "include", "includes", "included", et "ncluded" devront être tous remplacés par « includ ».

**Suppression des non-mots** : les non-mots et la ponctuation devront être supprimés. Tous les espaces blancs (onglets, nouvelles lignes, espaces) devront être remplacés par un seul espace.

Le résultat de ces étapes de prétraitement est illustré à la Figure 2.

```
anyon know how much it cost to host a web portal well it depend on how
mani visitor your expect thi can be anywher from less than number buck
a month to a coupl of dollarnumb you should checkout httpaddr or perhap
amazon ecnumb if your run someth big to unsubscrib yourself from thi
mail list send an email to emailaddr
```

*Figure 2. Email après nettoyage*

### 2.1.1 Construction du vocabulaire

Après le prétraitement des e-mails, une liste de mots sera pour chaque e-mail. L'étape suivante consiste à choisir les mots que nous aimerions utiliser dans notre classificateur et que nous voudrions laisser de côté. Pour ce projet, il est possible de choisir uniquement les mots les plus fréquents (la liste de vocabulaire). Puisque les mots qui se produisent rarement dans l'ensemble de formation ne sont que dans quelques e-mails, ils peuvent provoquer un sur-apprentissage.

La liste complète du vocabulaire devra être sauvegardée dans un fichier, exemple vocab.txt.

Dans cette liste de vocabulaire seulement les mots qui apparaissent au moins K fois dans le corpus de spam devront être gardés. K devra être choisi empiriquement. En pratique, une liste de vocabulaire avec environ 10 000 à 50 000 mots sont souvent utilisés.

Une fois ayant obtenu la liste de vocabulaire, il sera possible de mapper chaque mot dans l'email prétraité à son index dans une liste d'index de mots (qui contient l'index du mot dans la liste de vocabulaire).

Ceci est fait en cherchant le mot dans le vocabulaire liste vocabList et trouver si le mot existe. Si oui, il devra être ajouté dans la variable index des mots. Si le mot n'existe pas, et n'est donc pas dans le vocabulaire, le mot devra être ignoré.

### 2.2 Extraction de caractéristiques

L'extraction de fonctionnalités devra convertir chaque e-mail en un vecteur dans $R^n$. Pour ce projet, nous utiliserons n = # mots de vocabulaire liste.

Il existe deux manières de représenter le vecteur caractéristique, une représentation binaire et une représentation par comptage.

**Représentation binaire des caractéristiques** : la caractéristique $x_i \in \{0, 1\}$ d'un e-mail correspond à

savoir si le i-ème mot du dictionnaire apparaît dans l'e-mail. Autrement dit, $x_i = 1$ si le i-ième mot est dans l'e-mail et $x_i = 0$ si le i-ème mot n'est pas présent dans l'e-mail.

**Représentation des caractéristiques par comptage** : la caractéristique $x_i \in \{0,.., M\}$ d'un e-mail correspond au nombre d'apparitions du i-ème mot du dictionnaire dans l'e-mail.

**Étape 2 : Classification**

Une fois les vecteurs caractéristiques obtenus ; il est possible d'utiliser tous les classifieurs appris durant ce semestre :

Une comparaison des classifieurs devra être faite et les résultats discutés.

Une modularité du code est exigée.

L'utilisation des implémentations existantes (librairies) d'algorithmes de classification est permise, toutefois il est important de justifier le choix des libraires ainsi que des paramètres.

L'utilisation d'algorithmes d'apprentissage profond pour cette tâche est grandement appréciée et une comparaison entre les approches classiques et celles de l'apprentissage profond est encouragée.

Les étudiants ayant obtenus de bons résultats, sont encouragés à participer au challenge :
CERIST - NLP Challenge 2022

**Consignes :**

Les livrables du projet sont :

- Code source du projet.
- Rapport de projet : décrivant et justifiant les choix des approches et des libraires utilisées ainsi qu'une analyse (synthèse) des résultats obtenus.
- Photos des étudiants.

Le délai de remise du projet sera au plus tard 5 jours avant les délibérations.