3i005

Gosse-dumesnil.tony

February 2018

1 Exercice 2

1.1

On associe le label +1 à l'email(c'est un spam) si la probabilité que la email soit un spam sachant sa description x est plus grande que la probabilité qu'il ne soit pas un spam sachant sa desciption x, sinon on associe le label -1 à l'email(ce n'est pas un spam).

On a
$<=> P(Y = +1|X = x) > P(Y = -1|X = x)$
$<=> \frac{P(X=x|Y=+1)P(Y=+1)}{P(X=x)} > \frac{P(X=x|Y=-1)P(Y=-1)}{P(X=x)}$
$<=> P(X = x|Y = +1)P(Y = +1) > P(X = x|Y = -1)P(Y = -1)$

On a autant de chance de tomber sur un spam ou un non spam Donc on peut retirer $P(Y = +1)$ et $P(Y = -1)$ dans l'inéquation. De même pour $P(X = x)$ qui est présent des deux côtés de l'inéquation.

donc on a
$P(X = x|Y = +1) > P(X = x|Y = -1)$

1.2

Il n'est pas raisonnable de considérer tous les cas possibles, on aurait des cas avec une probabilité nulle puisque on se base sur les données recueillies pour estimer les nouvelles. Il suffit de tomber sur un cas que l'on n'a pas dans notre dataset d'entraînement.

Il parait raisonnable de ranger les données par paquets de 20, 10 c'est sur-évaluer, plus de 20 serait sous-évaluer

1.3

apprend_modele(spam, non_spam) pour chaque fourchette de longueurs attribue une probabilité d'être un spam en utilisant la formule suivante:
$P(X = x|Y = +1) = \frac{P(Y=+1|X=x)P(X=x)}{P(Y=+1)}$

or:

$P(X = x) = \frac{nombred'emaildelongueurx}{nombretotald'email}$

predict: pour chaque email de longueur x il regarde si la probabilité d'être un spam est plus élevée que de ne pas l'être en utilisant le résultat de apprend_modele.

La probabilité de l'erreur se défini par $1 - P(f(x) = y)$ On a $P(f(x) \neq y) \approx 0.4125$

On en déduis que la classification par taille de mail garantie mal le type des mails que nous recevons.

2 Exercice 3

2.1

La probabilité $P(X = x|Y = +1)$ représente la probabilité que le vecteur binaire ait les valeurs du vecteur x sachant que c'est un spam. Calculer la distribution $P(X = x/Y = +1)$ avec x un vecteur binaire $\{x_1, x_2, ..., x_d\} \in \{0, 1\}^d$ est raisonnable car pour la probabilité $P(Y = +1|X = x)$ serait très faible du au nombre de configurations possible du vecteur.

Pour un dictionnaire de 1000 mots nous devons calculer 2*1000 paramètres pour caractériser la distribution car nous avons besoins du nombre de spams qui contient le mot et du nombre total de mails qui contient ce mot.

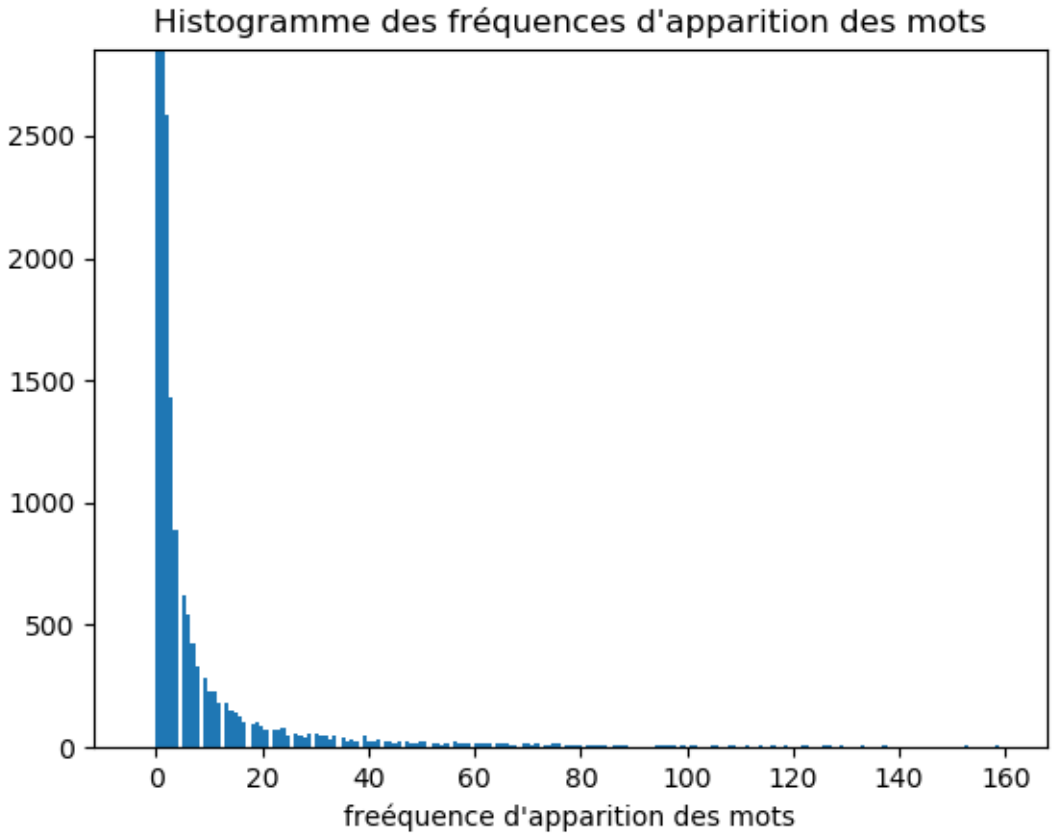
On a $P(X = x/Y = +1) = P(X_1 = x_1, X_2 = x_2, ..., X_d = x_d/Y = +1)$

Or les variables aléatoires $X_1, X_2, ...X_n$ sont considérées comme indépendantes donc

$P(X = x|Y = +1) = P(X_1 = x_1|Y = +1) * ... * P(X_d = x_d|Y = +1)$
$P(X = x|Y = +1) = \prod_{i=1}^{d} P(X_i = x_i|Y = +1)$

$P(X = xi|Y = +1) =$ Nombre de spam qui contiennent xi/ Nombre total de mails qui contiennent xi

2.2



Si on retire tout les mots qui ont une fréquence d'apparition au dessus de 20 on obtient un haut pourcentage de réussite avec $P(f(x) = y) \approx 0.82431$ mais le problème est que le temps d'éxecution est très long du au grand nombres de mots dans le dictionnaire. C'est du surapprentissage ! Nous décidons donc de réduire considérablement la taille du dictionnaire en retirant tout les dont la fréquence d'apparition en dessous de A avec $A > 40$.

2.3

Passer au logarithme les probabilités à calculer est possible car on a $P \in [0, 1]$ et $ln(P) \in [-\infty, 0]$ et pour $P \in [0, 1]$ et $P \in [0, 1]$ si $P1 < P2$ alors $ln(P1) < ln(P2)$ car ln est strictement croissante et $ln(P1 * P2) = ln(P1) + ln(P2) \in [-\infty, 0]$

On obtient un pourcentage de valeurs bien classées ou probabilité de réussite pour $A = 60$ $P(f(x) = y) \approx 0.8$

On en deduit que la classification sémantique prédit très bien le type du mail que nous recevons.