Dara Pak Master 1 OIVM

TP 3 : Classification binaire avec approches probabilistes

Introduction

Dans notre TP on peut considérer ham et spam en valeur numérique 0 et 1.

VP (Vrais Positifs) : Le nombre de prédictions correctes pour les échantillons positifs.

FN (Faux Négatifs) : Le nombre d'échantillons positifs que le modèle n'a pas correctement identifiés.

FP (Faux positifs) : Nombre d'échantillons négatifs 0(ham) classés à tort comme positifs 1(spam).

Exercice 1

Question 5 :

Écriture mathématique	Définitions
Accuracy: $$Accuracy = \frac{\text{Vrais Positifs (VP)} + Vrai\ négatifs(VN)}{\text{Total}}$$	
Recall: $$Rappel(recall) = \frac{\text{Vrais Positifs (VP)}}{\text{Vrais positifs(VP)} + Faux\ Négatfis(FN)}$$	Un score recall près de 1 meilleur est la détection car elle serait à 100% de précision. Le recall est particulièrement important lorsqu'on manque d'échantillons, comme dans les diagnostics médicaux ou la détection de fraude.
F1_Score: $$F1 = 2 * \frac{\text{Vrais Positifs (VP)}}{\text{Vrais positifs(VP)} + Faux\ Positifs\ (FP) + Faux\ Négatfis(FN)}$$	Le F1-score atteint sa valeur maximale de 1 (100%) lorsque la précision et le rappel sont parfaits. Un F1-score faible indique soit une mauvaise précision, soit un faible rappel, ou les deux
Macro Avg: $$macro\ avg = \frac{\sum_{i=1}^{N} Mi}{\text{N}}$$	C'est une moyenne arithmétique des métrique (précision, rappel, F1_score) N = le nombre de classe, dans notre cas cela correspond à v1 et v2 Mi = la métrique (précision, rapport ou F1_score) pour la classe i (1 et 2 dans notre cas)

Weighted avg : $$weighted\ avg = \frac{\sum_{i=1}^{N} Si.Mi}{\sum_{i=1}^{N} Si}$$	C'est une moyenne pondérée par le support des métrique (précision, rappel, F1_score) N = le nombre de classe, dans notre cas cela correspond à v1 et v2 Mi = la métrique (précision, rapport ou F1_score) pour la classe i (1 et 2 dans notre cas) Si = Support (nombre de d'échantillon) de la classe i (v1 ou v2)

En supplément on peut utiliser la métrique **hamming_loss** qui permet de calculer la proportion de class mal classifiés, en prenant en compte chaque classe individuellement.

Question 6 :

L'intérêt d'un modèle génératif par rapport aux autres modèles dans notre cas est qu'il est le plus adapté pour les données textuelles vectorisées (CountVectorizer) en matrice, est Bayes et le plus simple et rapide lorsque l'on a beaucoup de caractères comme nous avons ici.

Exercice 2

Question 6 :

Si l'on regarde en le rapport de classification du modèle complément Naïve Bayes dans la Figure 1 : tableau répertoriant les rapports de classification par modèle, nous avons :

La colonne réalité :	Ligne de prediction :
Represente les classes du csv réel Negative : 0 (classe de ham) Positive : 1 (classe spam)	Represente les prediction du modele Negative : 0 (le modele predit le nombre de classe de ham) Positive : 1 (le modele predit le nombre de classe spam)

A partir de toutes ces valeurs on peut calculer nos différentes métriques : Accuracy, Precision, recall, f1_score, heureusement il existe des fonctions dans scikit-learn pour faire les différents calculs.

Exercise 3:

Exercice 4 : Comparaison des performances

Question 1 : table de comparaison

Générative : Naïves bayes :
Matrice de confusion :

		Réalité	
		Négative : 0	Positive : 1
Prédiction	Négative : 0	957	8
	Positive : 1	12	138

Rapport de classification:

	Metric precision	recall	f1-score	support
0	0.99	0.99	0.99	965
1	0.95	0.92	0.93	150
Accuracy			0.98	1115
Macro avg	0.97	0.96	0.96	1115
weighted avg	0.98	0.98	0.98	1115

Complément : Naïves Bayes
Matrice de confusion :

		Réalité	
		Négative : 0	Positive : 1
Prédiction	Négative : 0	929 (VN)	36(FN)
	Positive : 1	9(FP)	141(VP)

Rapport de classification

	Metric precision	recall	f1-score	support
0	0.99	0.96	0.98	965
1	0.80	0.94	0.86	150
Accuracy			0.98	1115
Macro avg	0.89	0.95	0.92	1115
weighted avg	0.96	0.96	0.96	1115

Regression logistique:
Matrice de confusion:

		Réalité	
		Negative: 0	**Positive: 1**
Prediction	**Negative: 0**	965	0
	Positive: 1	22	128

Rapport de classification:

	Metric precision	**recall**	**f1-score**	**support**
0	**0.98**	**1**	**0.99**	**965**
1	**0.80**	**0.85**	**0.92**	**150**
Accuracy			**0.98**	**1115**
Macro avg	**0.99**	**0.93**	**0.95**	**1115**
weighted avg	**0.98**	**0.98**	**0.98**	**1115**

Figure 1 : tableau répertoriant les rapports de classification par modèle

Question 3: Comparison

Accuracy	**Analyses**
Generative Bayes	Précision élevée pour les spams (0.95) et les non-spams.

Complement Bayes	Précision plus faible pour les spams (0.80), mais meilleure que la Régression Logistique.
Regression logistique	Précision de 0.80 pour les spams.
Choix : Naïves Bayes est le meilleur en termes de précisions pour minimiser les faux positif (erreurs les ham)	

Recall	**Analyses**
Generative Bayes	Meilleur rappel pour les spams (0.92), ce qui signifie qu'il manque peu de spams
Complement Bayes	Meilleur rappel (0.94), légèrement supérieur au Naïve Bayes standard
Regression logistique	Appel plus faible (0.85) pour les spams, ce qui signifie qu'il manque davantage de spams par rapport au 2 autres modèles
Choix : Complément naives bayes est légèrement le 1er en termes de recall	

F1_Score	**Analyses**
Generative Bayes	Très bon score F1-score pour les spams (0.93) et les non-spams (0.99), elle est bien équilibrée
Complement Bayes	F1-score plus faible pour les spams (0.86), malgré un bon call, malheureusement l'accuracy était plus base
Regression logistique	Compromis moyen en spam et ham.
Choix : Naives Bayes est le plus équilibre en termes de F1_score	

Discussion :

Analyse temps d'entrainement :

Generative Bayes	0.196854829788208 secondes
Complement Bayes	0.19664859771728516 secondes
Regression logistique	0.7287561893463135 secondes

En termes de temps d'entrainement on a un très léger avantage du modèle compléments naïves Bayes

Analyse complexité :

Generative Bayes	O(n)
Complement Bayes	
Regression logistique	

Pertinence de chaque approche :

Naïves Bayes est le plus idéale pour les données de type textuelle, lorsque les classes sont équilibrées.

Complément Naïves Bayes : Plus adapté pour les classes déséquilibrées ce qui peut être plus utiles pour la détection d'anomalies.

Régression logistique : utiles pour une meilleure flexibilité dans la séparation des classes, peut donc être utiliser dans des csv plus brouillons.

Conclusion :

Pour des problèmes de type textuels simples et rapide. Pour un dataset déséquilibre il vaut mieux choisir Complément naïves bayes.

Pour des données complexes il vaut mieux utiliser la régression logistique car plus robuste au bruit comme nous l'avons vue lors des précédents TP.