2019-2020

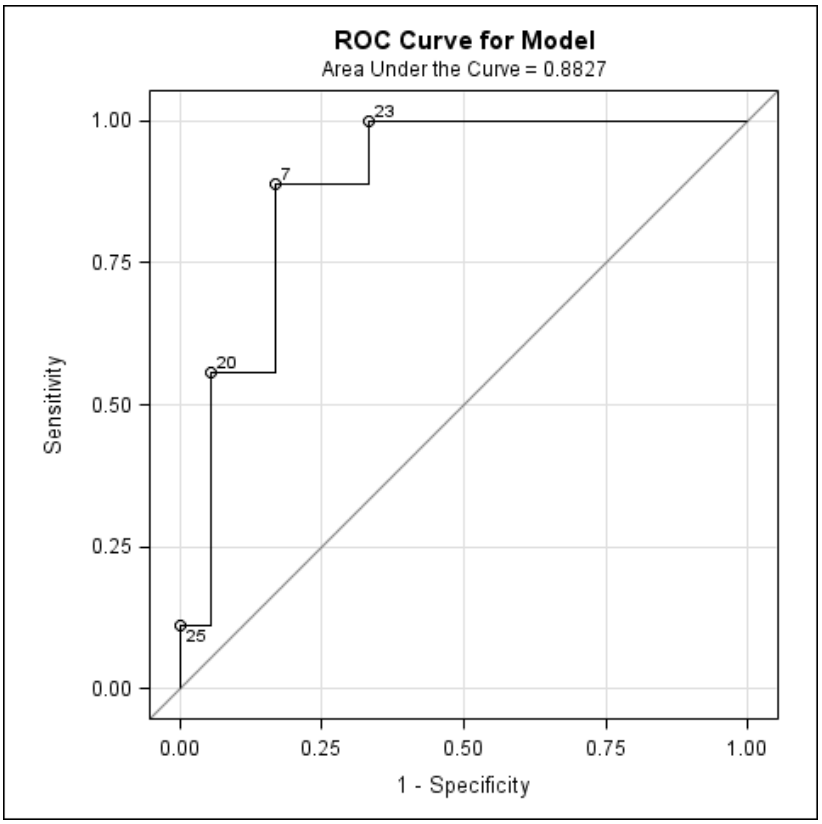
Apprentissage supervisé

TD/TP 3 – COMPARAISON DE METHODES
octobre 2019

L'objectif de ce TD/TP est d'apporter des éléments de comparaison entre les méthodes d'apprentissage supervisé que sont l'analyse discriminante, les arbres de décision, le classifieur bayésien naïf et les k plus proches voisins.

PARTIE 1 : partie TD

1. Vous trouverez ci-après, la courbe ROC obtenue avec un certain modèle. Ajouter sur le même graphique la courbe qui correspondrait d'une part à un modèle complètement au hasard et, d'autre part, ce qui correspondrait au modèle parfait. Quels sont les points les plus intéressants sur cette courbe ? Que représente l'aire sous la courbe ?
2. Comment se traduit avec les courbes ROC le fait qu'un modèle M1 sera toujours meilleur que M2 ? Qu'est-ce que l'enveloppe convexe dans la sélection de modèles ?



3. Courbe Lift

Soit une variable à expliquer binaire Y à deux modalités (Y=1 ou Y=0) que l'on souhaite prévoir grâce à p variables X1 à Xp. On considère que l'événement qui nous intéresse est Y=1.
A quoi peut correspondre la variable score du tableau ci-dessous ? Qu'est-ce que la courbe Lift ? Comment pourrait-on la construire pour le modèle correspondant au score calculé ci-dessous ?

Ce tableau est à votre disposition dans moodle : *Lift élèves.xls*

Obs	Y	score	% population	??? modèle considéré	??? modèle aléatoire	??? modèle parfait
1	1	0,92825	4%			
2	0	0,82762	7%			
3	**1**	**0,74119**	**11%**			
4	1	0,73182	15%			
5	**1**	**0,72418**	**19%**			
6	1	0,63537	22%			
7	0	0,63155	26%			
8	**0**	**0,61763**	**30%**			
9	1	0,49700	33%			
10	1	0,46686	37%			
11	**1**	**0,37062**	**41%**			
12	0	0,31244	44%			
13	0	0,29258	48%			
14	**0**	**0,26446**	**52%**			
15	1	0,22945	56%			
16	0	0,21983	59%			
17	0	0,17818	63%			
18	0	0,13490	67%			
19	**0**	**0,12012**	**70%**			
20	0	0,05337	74%			
21	0	0,01124	78%			
22	**0**	**0,00639**	**81%**			
23	0	0,00241	85%			
24	**0**	**0,00102**	**89%**			
25	0	0,00099	93%			
26	0	0,00040	96%			
27	**0**	**0,00011**	**100%**			

PARTIE 2 : partie TP avec mise en œuvre sous R

L'objectif de cette partie du TP est la détection des spams dans les courriels reçus par un individu : de construire un filtre qui permette de distinguer automatiquement les spams des messages pertinents en vue de leur non-acheminement voire de leur destruction.

Les spams recouvrent des publicités « non sollicitées» pour des produits, des sites Web, des astuces pour gagner de l'argent facilement, des invitations à faire suivre un message à un certain nombre de personnes (« chaines ») et nombreux autres messages indésirables qui polluent les boites à lettres électroniques. On peut noter que le thème des spams reçus par un individu dépend fortement de son réseau d'interlocuteurs, de ses centres d'intérêt, de son activité professionnelle, etc … Il est donc possible de personnaliser un détecteur de spams.

La base de données a été constituée par un ingénieur de Hewlett-Packard (George Forman, Palo Alto, USA) qui a rassemblé un certain nombre de ses courriels pertinents (non – spam) et d'autres considérés comme indésirables (spam) en juin et juillet 1999. Le filtre à spams que nous allons constituer est donc personnalisé car l'apprentissage est fait sur les messages qu'il reçoit, les caractéristiques des messages reçus reflètent les messages reçus par cette personne.

La base de données contient 4601 messages, dont 1813 spams (39.4%).
57 variables permettent de caractériser chaque message, elles correspondent aux caractères, mots qu'il contient et surtout à la fréquence avec laquelle ceux-ci apparaissent.

- 48 variables continues [0-100] sont du type « fréquence_du_mot_MOT » : pourcentage de mots du courriel égaux à MOT = 100*(nombre de fois où le mot MOT apparait dans l'e-mail / nombre total de mots dans le courriel). Un MOT est ici une suite de caractères alpha-numériques bornée par des caractères non alpha-numériques ou des blancs. Les fréquences relatives de 48 mots particuliers sont calculées pour chaque courriel
- 6 variables continues [0-100] sont du type « fréquence_du_caractère_CARACTERE » : la même définition que précédemment, avec non pas des mots, mais des caractères particuliers comme « ! », par exemple.

Liste des mots et caractères retenus, dans l'ordre du fichier

MOTS *				CARACTERES
V1 = make	V14 = report	V27 = George	V40 = direct	V49 = ;
V2 = address	V15 = addresses	V28 = 650	V41 = cs	V50 = (
V3 = all	V16 = free	V29 = Lab	V42 = meeting	V51 = [
V4 = 3d	V17 = business	V30 = labs	V43 = original	V52 = !
V5 = our	V18 = email	V31 = telnet	V44 = project	V53 = $
V6 = over	V19 = you	V32 = 857	V45 = re	V54 = #
V7 = remove	V20 = credit	V33 = data	V46 = edu	
V8 = internet	V21 = your	V34 = 415	V47 = table	
V9 = order	V22 = font	V35 = 85	V48 = conference	
V10 = mail	V23 = 000	V36 = technology		
V11 = receive	V24 = money	V37 = 1999		
V12 = will	V25 = hp	V38 = parts		
V13 = people	V26 = hpl	V39 = pm		

* Certains mots sont directement liés au concepteur de la base :
George Forman (gforman at nospam hpl.hp.com) 650-857-7835 , June-July 1999

- 1 variable continue positive égale à la longueur moyenne des séquences non-interrompues de lettres capitales dans le message (V55)
- 1 variable continue positive égale à la plus grande longueur de toutes les séquences non-interrompues de lettres capitales (V56)
- 1 variable continue égale à la somme des longueurs de toutes les séquences non-interrompues de lettres capitales (=nombre total de lettres capitales dans le message) (V57)

- **La variable cible est le statut du message : is_spam=0 (non-spam) ou is_spam =1 (spam)**

Mettre en œuvre les méthodes suivantes : l'analyse discriminante, les arbres de décision, le classifieur bayésien naïf et les k plus proches voisins. Comparer notamment à l'aide des courbes ROC et Lift les performances de ces méthodes sur ce jeu de données.

Le fichier de données à utiliser pour le TP est la table ***spambase*** de la librairie ***nutshell***

```
library(nutshell)
data(spambase)
```