TP : Régression logistique

1 Données artificielles

On considère le jeu de données artificielles (cf. Table 1) caractérisés par deux variables quantitatives x_1 et x_2 et une variable qualitative y à deux modalités, notées A et B. Le poids de chaque individu est choisi égal à 1/8.

	x_1	x_2	y
P_1	-2	0	A
P_2	2	-2	B
P_3	-1	1	A
P_4	-1	-1	A
P_5	-1	-2	B
P_6	0	0	A
P_7	-1	2	B
P_8	2	2	B

Table 1

Dans ce qui suit, on cherche à expliquer y en fonction de x_1 et x_2. Pour cela on réalise différentes classifications supervisées à l'aide du logiciel R.

1.1 Création des données dans le logiciel R

```
donnee1 <- matrix(c(-2,2,-1,-1,-1,0,-1,2,0,-2,1,-1,-2,0,2,2),ncol=2,byrow=F)
donnee2 <- matrix(c("A","B","A","A","B","A","B","B"),ncol=1,byrow=F)
donn<- cbind.data.frame(donnee1,donnee2)
nomligne<-c("P1","P2","P3","P4","P5","P6","P7","P8")
nomcol<-c("X1","X2","Y")
dimnames(donn)<-list(nomligne,nomcol)
donn
```

1.2 Régression Logistique

La commande "glm" réalise une régression logistique.

1. Effectuer une régression logistique (commande "glm" avec l'argument "family=binomial") du jeu de données :

   ```
   library(MASS)
   > modele3 = glm(Y ~X1+X2, family=binomial,donn)
   > summary(modele3)
   > predict(modele3, donn)
   > predict(modele3, donn,type="response")
   ```

2. Interpréter les résultats ainsi obtenus pour le modèle de régression logistique "modele3".

1.3 Evaluation de la qualité des modèles

1. **Matrice de confusion :** Utiliser la commande "table" pour construire le tableau de classement (TC), c.-à-d. le tableau qui croise les variables classe d'appartenance (donn[,3]) et classe d'affectation (i.e. "predict(modele, type="response")>0.5" pour la régression logistique), puis déterminer le pourcentage de bien classés lors de l'estimation du modèle. :

```
(TC[1,1]+TC[2,2])/sum(TC)
```

2. **CV :** Ecrire un scripte qui effectue une validation croisée avec un échantillon d'apprentissage de 80%.
3. **Courve ROC :** Installer le package ROCR, puis tracer la courbe ROC et évaluer l'AUC. Noter que la commande "prediction" permet de calculer les paramètres de base nécessaire à la définition de la courbe ROC.

```
library(ROCR)
modele3.posterior<-predict(modele3, donn,type="response")
modele3.pred<-prediction(modele3.posterior, donn[,3])
modele3.roc<-performance(modele3.pred, "tpr","fpr")
plot(modele3.roc,colorize=TRUE,add=T) # avec "add = TRUE" pour le 2 et 3° plot
# Evaluer l'AUC :
modele3.auc<-performance(modele3.pred, "auc") ; modele3.auc@y.values[[1]]
```

2 Données réelles

On considère le jeu de données contenu dans le fichier "don0.txt".
Les données portent sur un échantillon de 100 patients pour lesquels on a relevé les mesures suivantes :

- AGE (en années) ;
- POIDS (en kg) ;
- TAILLE (en cm) ;
- ALCOOL (en nombre de verres bus) ;
- SEXE (F ou H) ;
- RONFLE (ronflement : O = ronfle ; N = ne ronfle pas) ;
- TABA (tabac : O = fumeur ; N = non fumeur).

On cherche à expliquer/prédire la variable "RONFLE" à l'aide des autres variables.

2.1 Différents modèles possibles

1. Effectuer la régression logistique avec toutes les variables.
2. Effectuer la régression logistique avec les méthodes stepwise (forward, backward et both).
3. Comparer les résultats des tests de significativité des variables.
4. Comparer les résultats des différents modèles avec les courbes ROC et Precision-Recall.
5. Calculer les odds ratio puis interpréter les.

2.2 Prédiction

Soit quatre nouveaux patients pour lesquels les valeurs des variables explicatives sont les suivantes :

AGE	POIDS	TAILLE	ALCOOL	SEXE	TABA
42	55	169	0	F	N
58	94	185	4	H	O
35	70	180	6	H	O
67	63	166	3	F	N

Pour effectuer la prédiction, il faut d'abord créer un data-frame contenant les données sur les nouveaux patients : ce data-frame doit posséder la même structure que les données initiales.

1. Construire un data.frame appelé "n_donnes" contenant ces données.
2. En utilisant la commande "predict" sur n_donnees, donner les prédictions d'appartenance de ces quatre patients aux différentes classes, selon le meilleur modèle déterminé précédemment.

2.3 Régression logistique avec python

Effectuer cette RL sur python :
`https://www.justintodata.com/logistic-regression-example-in-python/`