

**Université de Carthage**  
**Ecole Supérieure de la Statistique et de l'Analyse de l'Information**

**Examen de Data Mining**

**3<sup>ème</sup> année du cycle de formation d'ingénieurs**

Durée de l'épreuve : 1 heure 30 - Documents non autorisés

Nombre de pages : 3 - Date de l'épreuve : 9 janvier 2020

**Exercice 1 :** On considère le tableau de données ci-dessous contenant les valeurs observées de deux variables quantitatives  $X^1$  et  $X^2$ , et d'une variable qualitative  $Y$  possédant les deux modalités notées A et B, sur un échantillon  $I$  de huit individus notés  $P_1, \dots, P_8$ . Chaque individu est muni du poids  $1/8$ .

	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$
$x^1$	4	3	1	0	4	3	5	4
$x^2$	5	4	2	1	4	3	3	2
$y$	A	A	A	A	B	B	B	B

Par la suite, on applique différentes méthodes de classification supervisée à ces données afin d'expliquer  $Y$  en fonction de  $X^1$  et  $X^2$ . Pour cela, on utilise les commandes du logiciel R. On note  $g$ ,  $g_A$ ,  $g_B$  les centres de gravité respectifs du nuage  $I$  et des classes  $I_A = \{P_1, P_2, P_3, P_4\}$  et  $I_B = \{P_5, P_6, P_7, P_8\}$ .

**A- ANALYSE FACTORIELLE DISCRIMINANTE**

1- **(1.5 pts)** Calculer les centres de gravité  $g$ ,  $g_A$ ,  $g_B$ .

$g = (3, 3)$ ,  $g_A = (2, 3)$ ,  $g_B = (4, 3)$

On effectue l'AFD linéaire du tableau de données.

2- **(1 pt)** Expliquer pourquoi il n'existe qu'un seul axe factoriel discriminant (non trivial).

L'AFD consiste à réaliser l'ACP du nuage des centres de gravités qui contient deux points distincts  $g_A$  et  $g_B$ . Par conséquent il n'existe qu'un seul facteur discriminant non trivial.

3 - **(1 pt)** Quelle est la commande de R qui permet d'appliquer une AFD linéaire aux données. On notera "don" le data.frame dans lequel sont enregistrées les données. On précisera les arguments nécessaires pour cette fonction.

Il s'agit de `lda(Y ~., data=don)`

4- **(1.5 pts)** Sachant que le facteur discriminant a pour coordonnées :

X1 1.279204

X2 -1.066004

Indiquer les scores des 2 centres de gravité.

Le score  $z(P)$  de tout point est donné par :

$$z(x_1, x_2) = b' * (x - g) = 1.279204 * x_1 - 1.066004 * x_2 - 1.279204 * 3 + 1.066004 * 3.$$

D'où  $Z(g_A) = -1.27$  et  $Z(g_B) = 1.27$ .

5- **(2 pts)** Les probabilités *a posteriori* et les scores des individus données par l'AFD linéaire du tableau de données sont donnés ci-dessous :

```
$posterior
      A      B
P1 0.898604854 0.10139515
P2 0.938616893 0.06138311
P3 0.978504501 0.02149550
P4 0.987428061 0.01257194
P5 0.366919631 0.63308037
P6 0.500000000 0.50000000
P7 0.001434570 0.99856543
P8 0.002472623 0.99752738
```

```
$x
      LD1
P1 -0.8528029
P2 -1.0660036
P3 -1.4924050
P4 -1.7056057
P5  0.2132007
P6  0.0000000
P7  2.5584086
P8  2.3452079
```

Déterminer de deux manières différentes la classe d'affectation de chacun des individus. On affecte à la classe *A* si le score est négatif (manière 1) (resp. si la probabilité *a posteriori* est inférieure à 0.5 (manière 2)) d'où les affectations AAAAB?BB. L'individu *P6* de score nul (probabilité =0.5) est affecté au hasard.

## B- ARBRE DE DÉCISION

On considère les résultats de la classification supervisée réalisée à l'aide de l'arbre de décision. L'arbre obtenu est présenté ci-dessous :

```
> arbre.full <- rpart(Y ~ ., data = donn, minsplit =3, method = "class")
> print(arbre.full)
n= 8
```

```
node), split, n, loss, yval, (yprob)
      * denotes terminal node
```

```
1) root 8 4 A (0.5000000 0.5000000)
  2) X1< 2 2 0 A (1.0000000 0.0000000) *
  3) X1>=2 6 2 B (0.3333333 0.6666667)
    6) X2>=3.5 3 1 A (0.6666667 0.3333333) *
```

7) `X2< 3.5 3 0 B (0.0000000 1.0000000) *`

6- (1 pt) Commenter la ligne de commande qui a permis d'obtenir cet arbre.

Il s'agit d'effectuer un arbre de classification (`class`) sur les données (`don`) avec un `minsplit` égale à 3 ce qui veut dire qu'un noeud est segmenté tant que son effectif est supérieur à 3

7- (2 + 2 = 4pts) Déterminer la classe prédite de chacun des huit individus et en déduire la matrice de confusion.

Les affectations sont les suivantes : AAAAABBB

A B

La matrice de confusion est donnée par : A 4 0

B 1 3

8- (2 pts) On considère la courbe ROC évaluant le modèle de l'arbre de décision obtenu pour prédire qu'un objet appartient à la classe B. Donner les coordonnées de trois points qui sont situés sur cette courbe.

A partir de la matrice de confusion de la question 7, on obtient le point de la courbe ROC qui correspond à une sensibilité =  $3/4$  et à une spécificité = 1. Les 3 points sont donc les suivants : (0, 0), (0,  $3/4$ ) et (1, 1).

**Exercice 2 :** Le traitement du cancer de la prostate change si le cancer a atteint ou non les noeuds lymphatiques entourant la prostate. Pour éviter une investigation lourde un certain nombre de variables sont considérées comme explicatives de la variable  $Y$  :  $Y = 0$  si le cancer n'a pas atteint le réseau lymphatique et  $Y = 1$  sinon. Le but de cette étude est donc d'expliquer et de prédire  $Y$  par les variables suivantes :

- `age` : âge du patient au moment du diagnostic ;
- `acide` : le niveau d'acide phosphate sérique ;
- `rayonx` : le résultat d'une analyse par rayon X, 0= négatif et 1= positif ;
- `taille` : la taille de la tumeur, 0= petite et 1= grande ;
- `grade` : l'état de la tumeur déterminé par biopsie, 0= moyen et 1= grave ;
- `log.acid` : le logarithme népérien du niveau d'acidité ;

On dispose d'une base de données constituée de 53 individus. Chacun des 53 individus est décrit par les 6 variables prédictives présentées ci-dessus ainsi que par sa valeur sur la variable  $Y$ . Par la suite, on applique la méthode SVM (Support Vector Machine) de classification supervisée à ces données afin d'expliquer  $Y$  par l'ensemble des variables explicatives que l'on note  $X$ . Pour cela, on utilise les commandes du logiciel *Python*.

On a appliqué le scripte suivant :

```
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.25,random_state=0)
```

```
from sklearn import svm
```

```
svc = svm.SVC(C=1.0, kernel='rbf',  
              degree= 3, gamma=1.0,
```

```

coef0=0.0, shrinking=True,
probability=True,tol=0.001,
cache_size=200, class_weight=None,
verbose=False,max_iter= -1,
random_state=None)

```

```

from sklearn.preprocessing import StandardScaler

```

```

scaler = StandardScaler()
scaler.fit(X,Y)
x_train_scaled = scaler.transform(X_train)
svc.fit(x_train_scaled,Y_train)

```

1- (1 + 1 + 1 = 3pts) Commenter les lignes de commandes précédentes.

a) La commande relative au tirage de l'échantillon d'apprentissage (75%) et de l'échantillon test (25%) par la fonction `train_test_split`. b) Exécution de la fonction `svc = svm.SVC` avec les principaux paramètres de la méthode SVM qui sont le soft margin parameter  $C = 1$ , le kernel *rbf* et son paramètre  $\gamma = 1$ . c) La troisième partie est dédiée à la standardisation des données d'apprentissage puis l'application de la fonction `svc` aux données standardisées.

2- (1.5 + 1.5 = 3pts) Afin de déterminer le meilleur modèle, en terme d'erreur de prédiction, obtenu à partir de la méthode SVM, sur quels paramètres de la fonction `svm.SVC` devrions nous agir? Quelle fonction *Python* permettrait d'obtenir les meilleures valeurs de ces paramètres. On expliquera le principe de cette fonction.

Il s'agit d'agir sur les 3 paramètres évoqués plus haut de la fonction `svc = svm.SVC`. La fonction à utiliser étant la fonction `sklearn.model_selection.GridSearchCV` : Exhaustive search over specified parameter values for an estimator.