CORRIGÉ DE L'EXAMEN DE DM - SESSION DE MAI 2024

Partie 1

1) Remplacer les " ?" de la commande `step` par les paramètres adéquats pour effectuer la régression logistique pas à pas.
`step(modele1, dir="backward")`

2) Expliquer le principe de la sélection pas à pas utilisée ci-dessus.
Le principe de la sélection pas à pas backward consiste à partit avec un modèle contenant l'ensemble des varaibles puis à retirer à chaque étape la variable dont le retrait minimise le plus l'AIC. C'est ainsi que dans notre cas le modèle commence par supprimer la varaible age, puis grade pour ne garder que les variables acide, taille, log.acid et rayonx.

3) Comparer les résultats du modele2 à ceux du modele1.
Alors que le modele2 utilise les variables acide, taille, log.acid et rayonx (cette dernière variable étant associée à l'AIC le plus faible), on constate que même si le modele1 utilise toutes les variables, les résultats du test de Wald montrent que ce sont les 4 variables du modele2 qui sont les plus significatives, rayonx ayant la p-value plus faible.

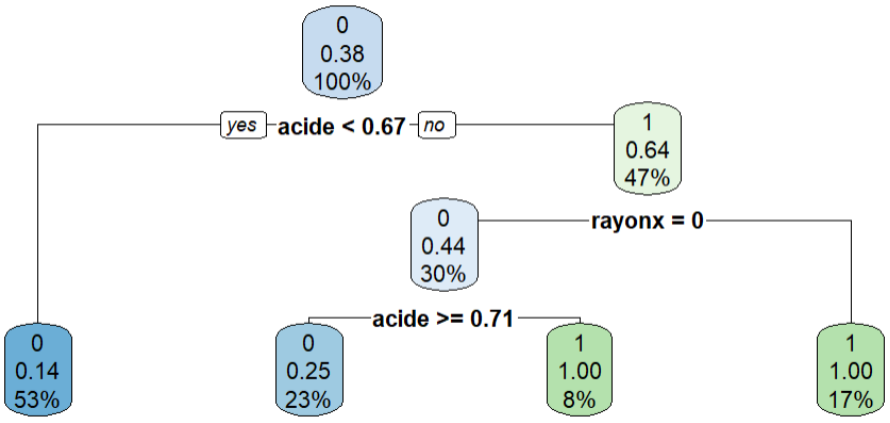
4) A partir des résultats des modèles 1 et 2, déterminer la classe d'affectation de l'individu 1 sachant qu'il a les caractéristiques suivantes : `age=66 ; acide=0.48 ; rayonx= 0 ; taille= 0 ; grade = 0 ; log.acid= -0.73`.

1. Avec le modele1 : $P(Y = 1) = exp(a)/1 + exp(a) = 0.02\ (< 0.5)$, avec $a = 10.087 - 0.043 * 66 - 0.48 * 8.48 - 0.73 * 9.609 = -3.83$. D'où Y=0.
2. Avec le modele2 : $P(Y = 1) = exp(b)/1 + exp(b) = 0.035\ (< 0.5)$, avec $b = -3.307$. D'où Y=0.

Partie 2

Afin d'expliquer Y, nous avons aussi effectué un arbre de classification sur le logiciel R. Les résultats sont présentés ci-dessous :

```
> modele3 <- rpart(Y~ ., data = cancer_prostate, method = "class",minsplit=5)
> printcp(modele3)

Classification tree:
rpart(formula = Y ~ ., data = cancer_prostate, method = "class",
    minsplit = 5)

Variables actually used in tree construction:
[1] acide  age    rayonx

Root node error: 20/53 = 0.37736

n= 53

     CP nsplit rel error xerror    xstd
```




```
1 0.350      0      1.00   1.00 0.17644
2 0.150      1      0.65   1.00 0.17644
3 0.050      3      0.35   0.65 0.15662
4 0.025      5      0.25   0.85 0.16991
5 0.010      7      0.20   0.85 0.16991
```

5) A partir de ces résultats, donner la commande qui permet d'obtenir l'arbre optimal à partir de l'arbre du modele3.

Pour avoir l'arbre optimal, on consulte la cptable pour voir l'intervalle du CP qui correpond au xerror + xstd le plus faible, ici]0.05, 0.15]. D'où la commande est : prune(modele3, cp=0.1).

6) On considère l'arbre donné par la figure ci-dessus. Commenter cet arbre puis donner les règles qui en découlent.

On constate que l'on a 4 noeuds terminaux. Le premier contenant 53% des individus et contenant 14% des positifs (Y=1) ; ce noeud est donc classé négatif (Y=0)... D'autre part les 4 règles sont les suivantes :

1. Si acide <0.67 alors Y=0
2. Si acide >0.67 et rayonx = 1 alors Y=1
3. Si 0.67 < acide < 0.71 et rayonx = 0 alors Y=1
4. Si acide > 0.71 et rayonx = 0 alors Y=0

7) A partir de cet arbre, déterminer la classe d'affectation de l'individu 1 de la question 4).

Cet individu vérifie la règle 1, il est donc classé Y=0.

Partie 3

Afin d'expliquer Y, nous avons enfin effectué un *random forest* sur le logiciel R. Les résultats sont présentés ci-dessous :

```
> modele5 <- randomForest(Y~.,data=cancer_prostate, mtry= 3,ntree=500)
> modele5$confusion
   0  1 class.error
0 26  7   0.2121212
1  9 11   0.4500000
```

```
> imp <- importance(modele5)
> imp
         MeanDecreaseGini
age              3.710594
acide            6.488794
rayonx           3.852525
taille           2.078109
grade            2.005427
log.acid         6.238672
```

8) Expliquer le lien entre le choix de la valeur du paramètre `mtry` et le taux d'erreur réel du modèle donné par le *random forest*.
Voir cours sur les Random forest (slide p. 18).

9) Expliquer comment a-t-on obtenu la matrice de confusion donnée par `modele5$confusion` ?
Cette matrice de confusion a été obtenue en utilisant le principe de l'erreur Out Of Bag (cf. slide 21 du cours de RF). L'avantage de la procédure Out Of Bag (OOB) est qu'elle ne nécessite pas de découper l'échantillon en échantillon d'apprentissage et échantillon test. Elle utilise le fait que chaque arbre est construit sur un échantillon et que, par conséquent, il n'utilise pas toutes les observations de la base.
Etant donné une observation i, on désigne par $\mathcal{I}_i$ l'ensemble des arbres de la forêt qui ne contiennent pas cette observation dans leur échantillon bootstrap. La prévision s'obtient en faisant voter les arbres de $\mathcal{I}_i$ à la majorité.

10) Commenter les résultats de `importance(modele5)`.
Ici on utilise un principe analogue à celui de l'OOBk donnant l'importance des variables dans la RF : visiblement c'est la variable acide qui est la plus imortante avec une MeanDecreaseGini de 6.48 suivi de log.acid....