TP Techniques de régression et de scoring

DUFOUR Guillaume

14/01/2022

Question : Peut-on prédire si les poissons appartiennent à l'espèce étudiée en fonction de leur mensuration ?

J'ai choisi de réaliser le TP à l'aide du langage R car c'est le langage que nous avons vu en cours lors des TPs.

Import des librairies

```
library(ggplot2)
library(ggcorrplot)
library(GGally)
library(caret)
library(pROC)
```

Récupération du dataset

On récupère les données via un jeu de données au format CSV.

```
fishes = read.table("./Fish.csv", header = TRUE, sep = ";")
```

Exploration des données du dataset

```
str(fishes)
```

```
## 'data.frame':    111 obs. of  4 variables:
##  $ Species: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Weight : num  242 290 340 363 430 450 500 390 450 500 ...
##  $ Height : num  11.5 12.5 12.4 12.7 12.4 ...
##  $ Width  : num  4.02 4.31 4.7 4.46 5.13 ...
```

```
head(fishes)
```

```
##   Species Weight  Height  Width
## 1       0    242 11.5200 4.0200
```

```
## 2       0    290 12.4800 4.3056
## 3       0    340 12.3778 4.6961
## 4       0    363 12.7300 4.4555
## 5       0    430 12.4440 5.1340
## 6       0    450 13.6024 4.9274
```

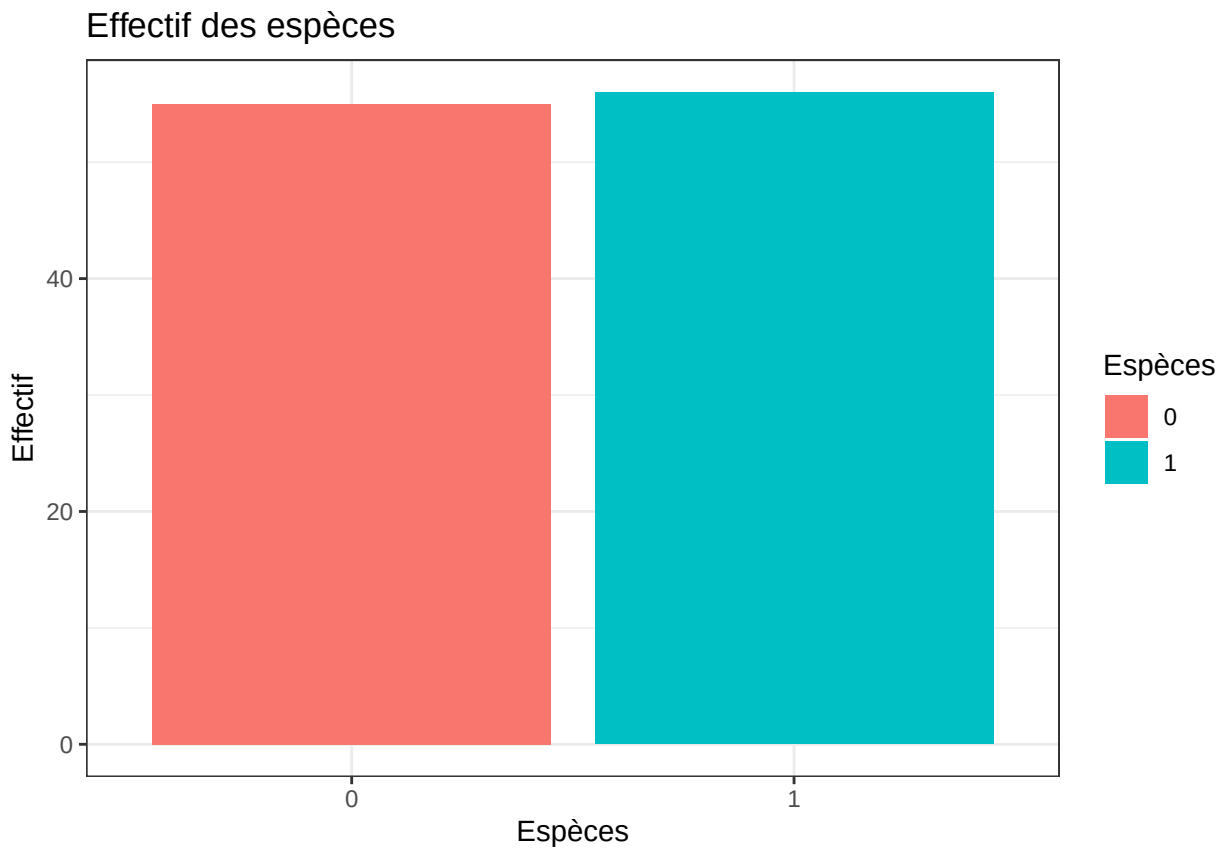
```r
summary(fishes)
```

```
##     Species           Weight           Height           Width
##  Min.   :0.0000   Min.   :   0.0   Min.   : 2.112   Min.   :1.408
##  1st Qu.:0.0000   1st Qu.: 137.5   1st Qu.: 6.192   1st Qu.:3.624
##  Median :1.0000   Median : 300.0   Median : 8.877   Median :4.566
##  Mean   :0.5045   Mean   : 415.0   Mean   : 9.960   Mean   :4.765
##  3rd Qu.:1.0000   3rd Qu.: 687.5   3rd Qu.:13.681   3rd Qu.:6.011
##  Max.   :1.0000   Max.   :1100.0   Max.   :18.957   Max.   :8.142
```

A l'aide de ces trois commandes, on peut avoir un rapide aperçu de la forme des données. On peut y voir le nombre d'observations (111) ainsi que des valeurs pour chaque colonne.
Ensuite, la commande head nous permet d'afficher quelques lignes du dataset (ici 6) ce qui nous permet de voir à quoi ressemble le dataset.
La fonction "summary" nous donne quelques chiffres sur les variables comme la moyenne mais également l'étendue des données, ce qui est intéressant à voir pour "Height", "Width" et "Weight".

Effectif des groupes

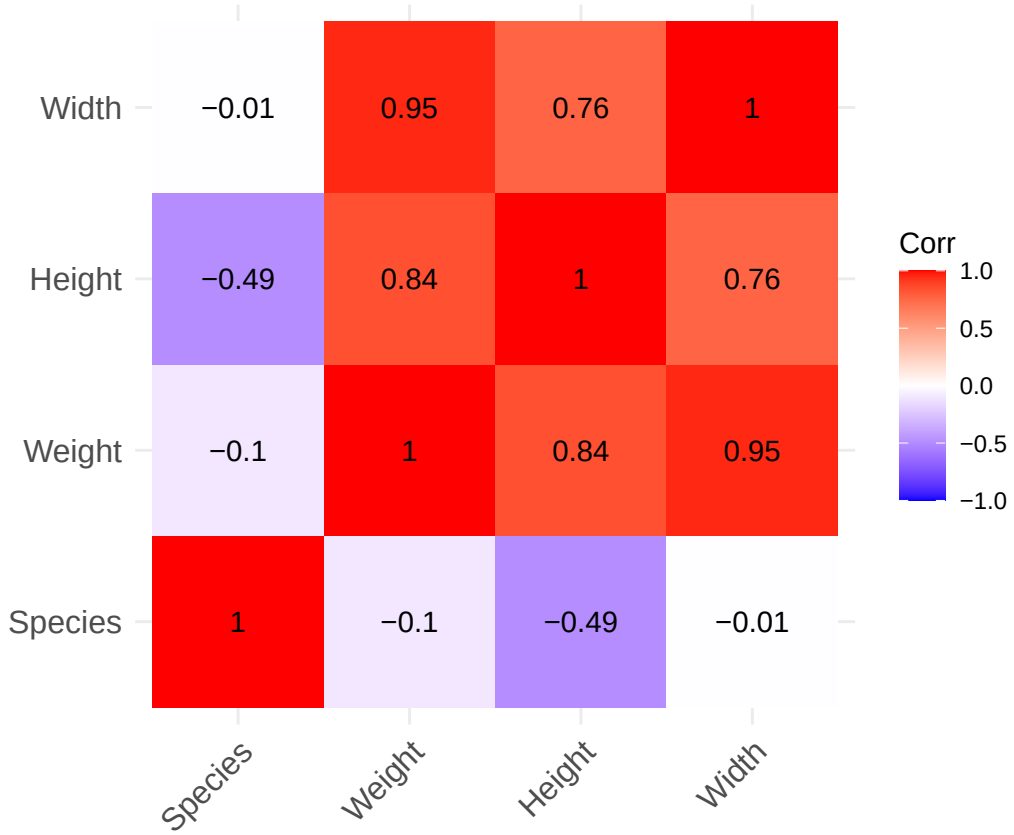
```r
#Graphique affichant les effectifs des espèces
ggplot(fishes, aes(x = as.factor(Species))) +
  geom_bar(aes(fill = as.factor(Species))) +
  xlab("Espèces") +
  ylab("Effectif") +
  labs(fill = "Espèces") +
  theme_bw() +
  scale_color_manual(values = c("#E63946", "#457B9D")) +
  ggtitle("Effectif des espèces")
```



Sur ce graphique, on peut voir les effectifs de chaque espèce. On veut voir qu'il y a très légèrement plus de poissons de l'espèce 1.

Analyse des données

```
#Graphique de corrélation des variables
ggcorrplot(cor(fishes), outline.col = NA, lab = TRUE)
```



A l'aide de ce graphique, nous pouvons voir que certaines variables sont très fortement corrélées ("Width" et "Weight" à 95 %).
"Height" et "Weight" sont également corrélées (à 84 %).
Ces corrélations semblent logiques puisque le poids très souvent relié à la taille (à la largeur ou à la hauteur).

Modèle de régression

Lors de cette étape, on sépare les données en deux échantillons :
- un est utilisé pour l'apprentissage du modèle
- le second est un échantillon utilisé pour tester les performances en prédiction du modèle (ainsi que sa capacité de généralisation)

On sépare les échantillons comme cela : 80 % pour l'échantillon d'apprentissage et 20 % pour l'échantillon de test.

```
#Taille de l'échantillon
n <- nrow(fishes)

#Indices des individus de l'échantillon d'apprentissage
train_index <- sample(x = 1:n, size = round(0.8 * n), replace = FALSE)

#Création des deux échantillons sous forme de dataset
train_data <- fishes[train_index,]
test_data <- fishes[-train_index,]
```

Training du modèle

On va ensuite chercher à prédire l'espèce du poisson étudié.

On commence par réaliser une régression backward.
On va tenter d'améliorer notre modèle en partant du modèle complet et à chaque étape, une variable du modèle sera enlevé pour trouver un modèle réduit qui représentera le mieux les données.

```
#Training du modèle
log_reg_backward <- glm(Species ~ ., data = train_data, family = "binomial")

#Selection du modèle
log_reg_backward <- step(log_reg_backward, direction = "backward")
```

```
## Start:  AIC=46.58
## Species ~ Weight + Height + Width
##
##          Df Deviance     AIC
## <none>        38.581  46.581
## - Width   1   42.040  48.040
## - Weight  1   45.066  51.066
## - Height  1  106.753 112.753
```

```
summary(log_reg_backward)
```

```
##
## Call:
## glm(formula = Species ~ Weight + Height + Width, family = "binomial",
##     data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.52934  -0.00220   0.00031   0.31414   1.65159
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.43473    5.47903   1.722  0.08507 .
## Weight       0.03032    0.01819   1.666  0.09564 .
## Height      -3.86987    1.18305  -3.271  0.00107 **
## Width        3.09975    2.01319   1.540  0.12363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 123.279  on 88  degrees of freedom
## Residual deviance:  38.581  on 85  degrees of freedom
## AIC: 46.581
##
## Number of Fisher Scoring iterations: 8
```

```r
hat_pi_backward <- predict(log_reg_backward, newdata = test_data, type = "response")
hat_y_backward <- as.integer(hat_pi_backward > 0.5)

table(hat_y_backward, test_data$Species)
```

```
##
## hat_y_backward  0  1
##              0 11  1
##              1  1  9
```

Nous allons ensuite réaliser une régression forward.
A l'inverse de la régression backward, on part d'un modèle vide et on va y ajouter les variables étape par étape jusqu'à obtenir un modèle qui représentera le mieux les données.

```r
#Training du modèle
log_reg_forward <- glm(Species ~ 1, data = train_data, family = "binomial")

#Sélection du modèle
log_reg_forward_sel <- step(
  log_reg_forward,
  direction = "forward",
  scope = list(lower=log_reg_forward, upper=~Weight+Height+Width)
)
```

```
## Start:  AIC=125.28
## Species ~ 1
##
##          Df Deviance    AIC
## + Height  1   98.817 102.82
## <none>       123.279 125.28
## + Weight  1  122.006 126.01
## + Width   1  123.258 127.26
##
## Step:  AIC=102.82
## Species ~ Height
##
##          Df Deviance     AIC
## + Weight  1   42.040  48.040
## + Width   1   45.066  51.066
## <none>       98.817 102.817
##
## Step:  AIC=48.04
## Species ~ Height + Weight
##
##         Df Deviance    AIC
## + Width  1   38.581 46.581
## <none>       42.040 48.040
##
## Step:  AIC=46.58
## Species ~ Height + Weight + Width
```

```r
summary(log_reg_forward_sel)
```

```
##
## Call:
## glm(formula = Species ~ Height + Weight + Width, family = "binomial",
##     data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.52934  -0.00220   0.00031   0.31414   1.65159
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.43473    5.47903   1.722  0.08507 .
## Height      -3.86987    1.18305  -3.271  0.00107 **
## Weight       0.03032    0.01819   1.666  0.09564 .
## Width        3.09975    2.01319   1.540  0.12363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 123.279  on 88  degrees of freedom
## Residual deviance:  38.581  on 85  degrees of freedom
## AIC: 46.581
##
## Number of Fisher Scoring iterations: 8
```

```r
hat_pi_forward <- predict(log_reg_forward_sel, newdata = test_data, type = "response")
hat_y_forward <- as.integer(hat_pi_forward > 0.5)

result <- table(hat_y_forward, test_data$Species)

result
```

```
##
## hat_y_forward  0  1
##             0 11  1
##             1  1  9
```

```r
#Calcul de l'accuracy
accuracy <- round((result[1] + result[4]) / sum(result), 4)
```

Nous voyons que dans les régressions backward et forward donnent les mêmes résultat. Dans les deux cas, le modèle est optimal lorsque les variables "Height" et "Weight" sont sélectionnées.
Nous allons donc réaliser une matrice de confusion avec une des deux sélections (forward choisi de manière arbitraire).

Matrice de confusion

```r
#Matrice de confusion
confusionMatrix(
  data = as.factor(hat_y_forward),
  reference = as.factor(test_data$Species),
  positive = "1"
)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 11  1
##          1  1  9
##
##                Accuracy : 0.9091
##                  95% CI : (0.7084, 0.9888)
##     No Information Rate : 0.5455
##     P-Value [Acc > NIR] : 0.0002906
##
##                   Kappa : 0.8167
##
##  Mcnemar's Test P-Value : 1.0000000
##
##             Sensitivity : 0.9000
##             Specificity : 0.9167
##          Pos Pred Value : 0.9000
##          Neg Pred Value : 0.9167
##              Prevalence : 0.4545
##          Detection Rate : 0.4091
##    Detection Prevalence : 0.4545
##       Balanced Accuracy : 0.9083
##
##        'Positive' Class : 1
##
```

Avec le modèle réalisé, on obtient une accuracy égale à 0.9091, ce qui est une bonne valeur (car celle-ci est proche de 1).

A l'aide de cette matrice de confusion, on obtient différentes informations :
- le nombre de vrai positif (prédit égal à 1 et réellement égal à 1) : 9 dans notre modèle
- le nombre de vrai négatif (prédit égal à 0 et réellement égal à 0) : 11 dans notre modèle
- le nombre de faux positif (prédit égal à 1 et réellement égal à 0) : 1 dans notre modèle
- le nombre de faux négatif (prédit égal à 0 et réellement égal à 1) : 1 dans notre modèle

On remarque que le modèle semble être performant car le ratio de positif est très correct et qu'on compte peu de négatif.

```r
#Calcul de l'AUC
auc(test_data$Species, hat_pi_forward)
```

```
## Area under the curve: 0.9667
```

AUC proche de 1 nous montre que nous avons un bon modèle. Il aurait été préférable d'avoir un échantillon de données plus important , ce qui nous permettrait d'avoir un échantillon de test plus important. En effet, avec la répartition choisie pour les datasets de train et de test (80 - 20 respectivement), l'échantillon de test peut être considéré comme petit. Modifier cette répartition permettrait d'avoir un échantillon de test avec plus de données mais l'accuracy serait moins bonne (environ 0.85 avec une répartition de 70 - 30).

Conclusion

Durant ce TP, nous avons réussi à trouver un modèle assez performant permettant de déterminer l'espèce de nouveaux poissons en fonction de deux variables : le poids et leur taille (Width). Pour cela, nous avons utilisé un modèle de régression logistique. A cause du faible jeu de données, la performance du modèle varie et donc la précision n'est pas forcément stable.