Comparative Analysis of Logistic and general additive model

Mohd Javed

2023-03-27

```
library(readr)
library(caret)
library(tidyverse)
n90_pol <- read_csv("n90_pol.csv")
View(n90_pol)
```

Bootstrap confidence interal

```
library(boot)
set.seed(1)

b3 <- boot(n90_pol,
  statistic = function(data, i) {
    cor(data[i, "amygdala"], data[i, "acc"], method='pearson')
  },
  R = 1000
)
b3
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = n90_pol, statistic = function(data, i) {
##     cor(data[i, "amygdala"], data[i, "acc"], method = "pearson")
## }, R = 1000)
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* -0.128484 0.004483558  0.09631828
```

```
set.seed(1)

b4 <- boot(n90_pol,
  statistic = function(data, i) {
    cor(data[i, "amygdala"], data[i, "orientation"], method='pearson')
  },
```

```
  R = 1000
)
b4
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = n90_pol, statistic = function(data, i) {
##     cor(data[i, "amygdala"], data[i, "orientation"], method = "pearson")
## }, R = 1000)
##
##
## Bootstrap Statistics :
##       original        bias    std. error
## t1* -0.2215031 -0.0004475096  0.09685546
```

```
set.seed(1)

b5 <- boot(n90_pol,
  statistic = function(data, i) {
    cor(data[i, "acc"], data[i, "orientation"], method='pearson')
  },
  R = 1000
)
b5
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = n90_pol, statistic = function(data, i) {
##     cor(data[i, "acc"], data[i, "orientation"], method = "pearson")
## }, R = 1000)
##
##
## Bootstrap Statistics :
##      original        bias    std. error
## t1* 0.2699053 -0.0001949058  0.09733553
```

Linear Model

```
mod<-lm(orientation~amygdala+acc,data=n90_pol)
stargazer::stargazer(mod,type="text")
```

```
##
## ================================================
##                         Dependent variable:
```

```
##                       ----------------------------
##                                 orientation
## ---------------------------------------------------
## amygdala                         -5.239*
##                                  (2.815)
##
## acc                              10.802**
##                                  (4.492)
##
## Constant                         3.578***
##                                  (0.091)
##
## ---------------------------------------------------
## Observations                       90
## R2                                0.108
## Adjusted R2                       0.088
## Residual Std. Error          0.859 (df = 87)
## F Statistic              5.285*** (df = 2; 87)
## ===================================================
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

Creating categorical variable

```
library(dplyr)
library(magrittr)
n90_pol<-n90_pol%>%
  mutate(orientation_cat=ifelse(orientation<=2,1,0))
```

partitioning of dataset

```
set.seed(1234)
training.individuals <- n90_pol$orientation_cat %>%
            createDataPartition(p = 0.6, list = FALSE)
train.data <- n90_pol[training.individuals, ]
test.data <- n90_pol[-training.individuals, ]
```

Training logistic model with Leave One Out Cross Validation

```
#specify that we want to use LOOCV
train_control <- trainControl(method = "LOOCV")

#train the model
model <- train(orientation_cat~acc+amygdala,data=n90_pol, method = "glm", trControl = train_control)
model
```

```
## Generalized Linear Model
##
## 90 samples
```

```
##  2 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 89, 89, 89, 89, 89, 89, ...
## Resampling results:
##
##   RMSE       Rsquared    MAE
##   0.3407546  0.07066104  0.2452126
```

```
# predicting with trained model on test data
predictions <- model %>% predict(test.data)

#
predict_bin<-ifelse(predictions<=.5,0,1)
predict_bin<-as.numeric(predict_bin)
# Model accuracy
mean(predict_bin==test.data$orientation_cat)
```

```
## [1] 0.8333333
```

```
#Confusion matrix

table(predict_bin,test.data$orientation_cat)
```

```
##
## predict_bin  0  1
##           0 30  6
```

Training logistic model with 5 fold Cross Validation

```
train_control <- trainControl(method = "cv", number = 5)

#train the model
model21 <- train(orientation_cat~acc+amygdala,data=n90_pol, method = "glm", trControl = train_control)


# predicting with trained model on test data
predictions <- model21 %>% predict(test.data)

#
predict_bin<-ifelse(predictions<=.5,0,1)
predict_bin<-as.numeric(predict_bin)
# Model accuracy
mean(predict_bin==test.data$orientation_cat)
```

```
## [1] 0.8333333
```

```r
#Confusion matrix

table(predict_bin,test.data$orientation_cat)
```

```
##
## predict_bin  0  1
##           0 30  6
```

##Training generalized additive model with Leave One Out Cross Validation

```r
#specify that we want to use LOOCV
train_control <- trainControl(method = "LOOCV")

#train the model
model212 <- train(orientation_cat~acc+amygdala,data=n90_pol, method = "gam", trControl = train_control)
model
```

```
## Generalized Linear Model
##
## 90 samples
##  2 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 89, 89, 89, 89, 89, 89, ...
## Resampling results:
##
##   RMSE       Rsquared    MAE
##   0.3407546  0.07066104  0.2452126
```

```r
# predicting with trained model on test data
predictions <- model212 %>% predict(test.data)

#
predict_bin<-ifelse(predictions<=.5,0,1)
predict_bin<-as.numeric(predict_bin)
# Model accuracy
mean(predict_bin==test.data$orientation_cat)
```

```
## [1] 0.8888889
```

```r
#Confusion matrix

table(predict_bin,test.data$orientation_cat)
```

```
##
## predict_bin  0  1
##           0 29  3
##           1  1  3
```

Training generalized additive model with 5 fold Cross Validation

```
train_control <- trainControl(method = "cv", number = 5)

#train the model
model211 <- train(orientation_cat~acc+amygdala,data=n90_pol, method = "gam", trControl = train_control)


# predicting with trained model on test data
predictions <- model211 %>% predict(test.data)

#
predict_bin<-ifelse(predictions<=.5,0,1)
predict_bin<-as.numeric(predict_bin)
# Model accuracy
mean(predict_bin==test.data$orientation_cat)
```

```
## [1] 0.8888889
```

```
#Confusion matrix

table(predict_bin,test.data$orientation_cat)
```

```
##
## predict_bin  0  1
##           0 29  3
##           1  1  3
```

Generalised aditive model has performed better than logistic model.

higher accuracy in GAM.