GAM

2024-03-21

Contents

```
knitr::opts_chunk$set(
  collapse = TRUE,
  warning = FALSE,
  message = FALSE,
  fig.dim = c(10, 5),
  fig.format = "png")
```

Load Data and Package

```
library(tidyverse)
library(caret)
## Load the the training/test set & control method

# Load the training and test sets
train_data <- read.csv("./Data/train_data.csv")
test_data <- read.csv("./Data/test_data.csv")

# Load the control method
ctrl1 <- readRDS("./Data/train_control.rds")

# change variables to be factors again
train_data <- train_data %>%
  mutate(gender = as_factor(gender),
         diabetes = as_factor(diabetes),
         hypertension = as_factor(hypertension),
         vaccine = as_factor(vaccine),
         severity = as_factor(severity))

test_data <- test_data %>%
  mutate(gender = as_factor(gender),
         diabetes = as_factor(diabetes),
         hypertension = as_factor(hypertension),
         vaccine = as_factor(vaccine),
         severity = as_factor(severity))
```

GAM

```
set.seed(1)

x_train = train_data |>  select(-recovery_time)
y_train = train_data|> select(recovery_time) |>pull()

x_test = test_data |>  select(-recovery_time)
y_test = test_data|> select(recovery_time) |>pull()

gam_model = train(x = x_train,
                  y = y_train,
                  method = "gam",
                 #metric = "RMSE", by default
                  trControl = trainControl(method = "cv", number = 10))

gam_model_final = gam_model$finalModel

summary(gam_model_final)
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender + hypertension + diabetes + vaccine + severity +
##     study + smoking + race + s(age) + s(SBP) + s(LDL) + s(bmi)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    43.0689     1.1322  38.039  < 2e-16 ***
## gender1        -3.6317     0.7936  -4.576 4.97e-06 ***
## hypertension1   3.2473     0.7998   4.060 5.06e-05 ***
## diabetes1      -1.2603     1.1111  -1.134 0.256797
## vaccine1       -6.3587     0.8101  -7.849 6.26e-15 ***
## severity1       8.1497     1.2720   6.407 1.78e-10 ***
## studyB          4.6462     0.8468   5.487 4.52e-08 ***
## smoking         1.9839     0.5779   3.433 0.000607 ***
## race           -0.1192     0.3699  -0.322 0.747330
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df       F p-value
## s(age) 3.908e-07      9   0.000   0.510
## s(SBP) 1.737e-06      9   0.000   0.395
## s(LDL) 3.093e-01      9   0.049   0.231
## s(bmi) 8.115e+00      9 109.190  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.327   Deviance explained = 33.2%
## GCV = 378.61  Scale est. = 375.87    n = 2402
```

```r
# Calculate test RMSE of optimal model
test_predictions = predict(gam_model_final, x_test)

gam_test_RMSE = sqrt(mean((y_test - test_predictions)^2))
gam_test_RMSE
## [1] 18.51986
```

```r
set.seed(1)

x_train_A = train_data |> filter(study == "A") |> select(-recovery_time)
y_train_A = train_data|> filter(study == "A") |> select(recovery_time) |>pull()

x_test = test_data |> filter(study == "A") |> select(-recovery_time)
y_test = test_data |> filter(study == "A") |> select(recovery_time)|>pull()

model.gam.a <- train(x = x_train_A,
                   y = y_train_A,
                   method = "gam",
                   #metric = "RMSE", by default
                   trControl = trainControl(method = "cv", number = 10))


ma_gam = model.gam.a$finalModel


x_train_B = train_data |> filter(study == "B") |> select(-recovery_time)
y_train_B = train_data|> filter(study == "B") |> select(recovery_time) |>pull()

x_test_B = test_data |> filter(study == "B") |> select(-recovery_time)
y_test_B = test_data |> filter(study == "B") |> select(recovery_time)|>pull()

model.gam.b <- train(x = x_train_B,
                   y = y_train_B,
                   method = "gam",
                   #metric = "RMSE", by default
                   trControl = trainControl(method = "cv", number = 10))


mb_gam = model.gam.b$finalModel
summary(ma_gam)
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender + hypertension + diabetes + vaccine + severity +
##     smoking + race + s(age) + s(SBP) + s(LDL) + s(bmi)
##
## Parametric coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    42.5323     0.6386  66.602  < 2e-16 ***
## gender1        -2.7039     0.4601  -5.877 5.07e-09 ***
## hypertension1   2.3612     0.4619   5.112 3.57e-07 ***
## diabetes1      -0.7718     0.6338  -1.218 0.223492
## vaccine1       -4.1436     0.4711  -8.796  < 2e-16 ***
## severity1       2.7942     0.7438   3.757 0.000178 ***
## smoking         1.5442     0.3321   4.649 3.60e-06 ***
## race           -0.2680     0.2171  -1.235 0.217161
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df      F p-value
## s(age) 2.019e-02      9  0.001   0.441
## s(SBP) 4.837e-07      9  0.000   0.949
## s(LDL) 4.203e-07      9  0.000   0.732
## s(bmi) 5.188e+00      9 64.234  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.317   Deviance explained = 32.2%
## GCV = 86.117  Scale est. = 85.415    n = 1620
summary(mb_gam)
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender + hypertension + diabetes + vaccine + severity +
##     smoking + race + s(age) + s(SBP) + s(LDL) + s(bmi)
##
## Parametric coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     48.53846    2.97490  16.316  < 2e-16 ***
## gender1         -4.59808    1.95838  -2.348  0.01914 *
## hypertension1    5.08990    3.22311   1.579  0.11471
## diabetes1       -2.45506    2.83527  -0.866  0.38682
## vaccine1       -11.36520    1.99270  -5.703 1.68e-08 ***
## severity1       15.30840    3.08683   4.959 8.74e-07 ***
## smoking          4.52580    1.45352   3.114  0.00192 **
## race            -0.05194    0.89224  -0.058  0.95359
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df      F p-value
## s(age) 5.556  6.632  1.930  0.0784 .
## s(SBP) 1.123  1.235  0.028  0.9652
## s(LDL) 1.000  1.000  2.620  0.1060
## s(bmi) 8.789  8.987 74.371  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.496   Deviance explained = 51.1%
## GCV = 749.82  Scale est. = 726.36    n = 782
```

Random Forest

```r
set.seed(1)

# long time for model training
# save the model in model file
control  <- trainControl(method="repeatedcv", number=10, repeats=3)
tunegrid <- expand.grid(mtry = 1:5)
#model.rf  <- train(x = x_train,
#                   y = y_train,
#                   trControl = control,
#                   tuneGrid = tunegrid)
#print(model.rf)
#model.rf <- train(x = x_train,
#                   y = y_train,
#                  method = "rf",
#                   #metric = "RMSE", by default
#                   trControl = trainControl(method = "cv", number = 10))


#saveRDS(model.rf, file = "./Model/model_rf.rds")
rf_model = readRDS("./Model/model_rf.rds")

rf_model_final = rf_model$finalModel
#plot(model.rf$finalModel)
```

```r
train_predictions =  predict(rf_model_final,x_train)

rf_train_RMSE = sqrt(mean((y_train - train_predictions)^2))
rf_train_RMSE

# Calculate test RMSE of optimal model
test_predictions = predict(rf_model_final, x_test)

rf_test_RMSE = sqrt(mean((y_test - test_predictions)^2))
rf_test_RMSE
```