GAM

2024-03-21

Contents


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

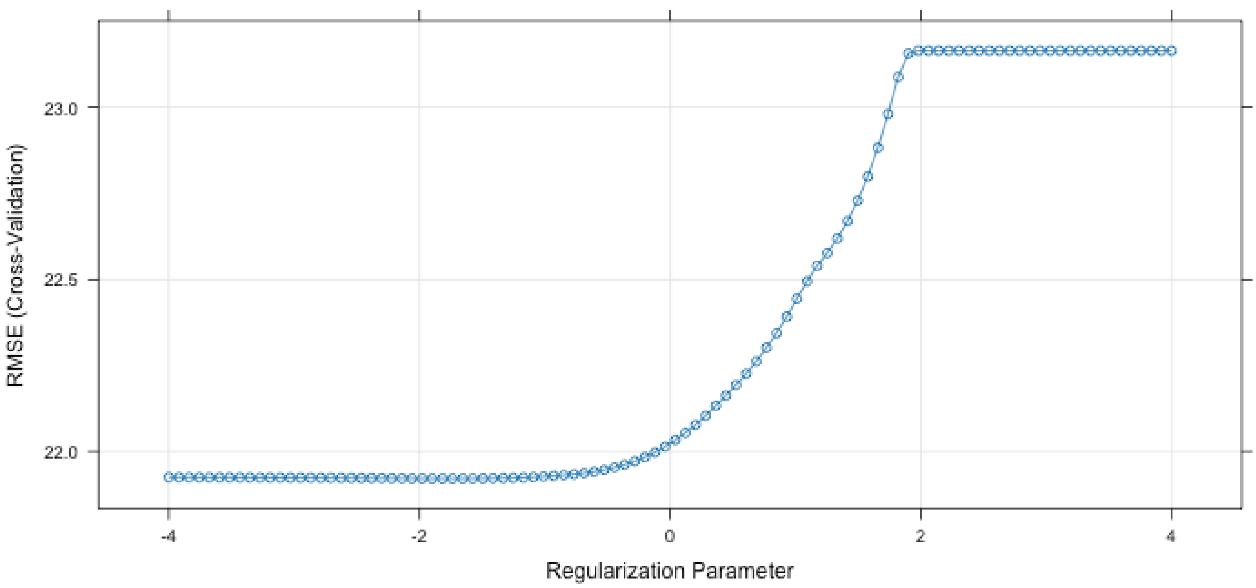
Load Data and Package

```
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))
```

```r
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))

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()
```

LASSO

```r
set.seed(2716)
lasso_model =  train(x = x_train, y = y_train,  method = "glmnet",
                     tuneGrid = expand.grid(alpha = 1,
                                            lambda = exp(seq(4, -4, length = 100))),
                     trControl = ctrl1,
                     preProcess = c("center", "scale"))
plot(lasso_model, xTrans = log)
```

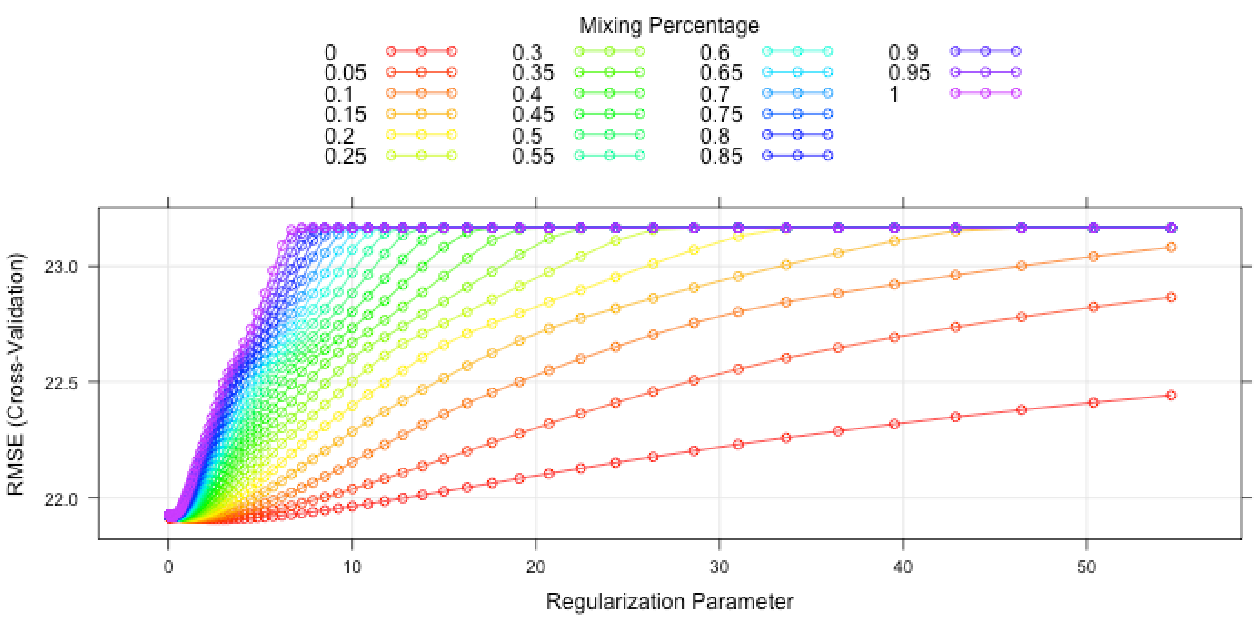



```r
# Get the index of the model with the lowest RMSE
best_model_index <- which.min(lasso_model$results$RMSE)

# Get the coefficients of the optimal model
optimal_model_coeffs <- coef(lasso_model$finalModel, s = lasso_model$results$lambda[best_model_index])

# Print the coefficients
print(optimal_model_coeffs)
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
```

```
## (Intercept)  45.30034033
## age           0.67090363
## gender       -2.60992261
## race         -0.05594452
## smoking       1.12285164
## bmi           6.19544334
## hypertension  2.30516545
## diabetes     -1.32273041
## SBP           0.39829875
## LDL          -0.42481562
## vaccine      -6.08848135
## severity      7.32419171
## study         .

lasso_pred = predict(lasso_model, x_test)
lasso_mse = mean((y_test - lasso_pred) ^ 2)
```

Elastic Net Model

```
set.seed(2716)
enet_model = train(x = x_train, y = y_train,  method = "glmnet",
                     tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                            lambda = exp(seq(4, -4, length = 100))),
                     trControl = ctrl1,
                     preProcess = c("center", "scale"))

myCol = rainbow(25)
myPar =
  list(superpose.symbol = list(col = myCol),
       superpose.line = list(col = myCol))
plot(enet_model, par.settings = myPar)
```

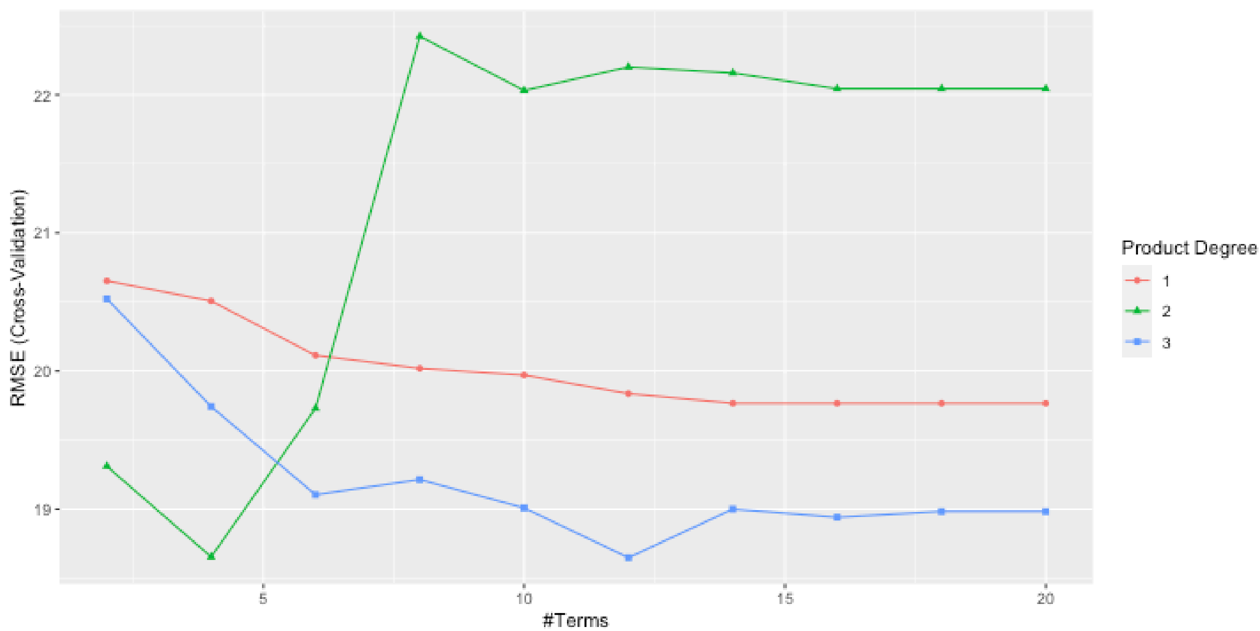


```r
# Get the index of the model with the lowest RMSE
best_model_index <- which.min(enet_model$results$RMSE)

# Get the coefficients of the optimal model
optimal_model_coeffs <- coef(enet_model$finalModel, s = enet_model$results$lambda[best_model_index])

# Print the coefficients
print(optimal_model_coeffs)
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept)  45.3328887
## age           0.7663697
## gender       -2.6692465
## race         -0.2138549
## smoking       1.1614323
## bmi           5.7103159
## hypertension  2.1099815
## diabetes     -1.6276258
## SBP           0.5771050
## LDL          -0.5794348
## vaccine      -5.8106118
## severity      7.0806029
## study         .

enet_pred = predict(enet_model, x_test)
enet_mse = mean((y_test - enet_pred) ^ 2)
```

PLS

MARS

```r
set.seed(2716)
mars_grid = expand.grid(degree = 1 : 3, nprune = seq(2, 20, by = 2))
mars_model = train(x = x_train, y = y_train, method = "earth",
                   tuneGrid = mars_grid, trControl = ctrl1)
ggplot(mars_model)
```




```
summary(mars_model$finalModel)
## Call: earth(x=data.frame[2402,12], y=c(29,34,41,50,3...), keepxy=TRUE,
##             degree=3, nprune=12)
##
##                                    coefficients
## (Intercept)                          -8.0457447
## gender1                              -3.3771125
## vaccine1                             -5.6486741
## h(1-smoking)                         -3.3000670
## h(bmi-23.8)                           7.4964488
## h(31-bmi)                             7.2296297
## h(bmi-23.8) * severity1               1.3291655
## h(bmi-31) * studyB                  -18.3936203
## smoking * h(bmi-31) * studyB         12.0251817
## h(age-62) * h(bmi-31) * studyB        4.6574999
## h(bmi-31) * h(112-LDL) * studyB       1.3157487
## h(bmi-31) * h(LDL-87) * studyB        0.8267105
##
## Selected 12 of 18 terms, and 8 of 12 predictors (nprune=12)
## Termination condition: Reached nk 25
## Importance: bmi, studyB, LDL, age, vaccine1, smoking, severity1, gender1, ...
## Number of terms at each degree of interaction: 1 5 2 4
## GCV 290.5539    RSS 681447.1    GRSq 0.4803533    RSq 0.4921888
## Coefficient of the MARS model
coef(mars_model$finalModel)
##                      (Intercept)                        h(31-bmi)
##                       -8.0457447                        7.2296297
##               h(bmi-31) * studyB  h(age-62) * h(bmi-31) * studyB
##                      -18.3936203                        4.6574999
##                      h(bmi-23.8) h(bmi-31) * h(112-LDL) * studyB
##                        7.4964488                        1.3157487
##                         vaccine1    smoking * h(bmi-31) * studyB
##                       -5.6486741                       12.0251817
##          h(bmi-23.8) * severity1  h(bmi-31) * h(LDL-87) * studyB
```

```
##                        1.3291655                              0.8267105
##                     h(1-smoking)                               gender1
##                       -3.3000670                             -3.3771125

# Get the index of the model with the lowest RMSE
best_model_index <- which.min(mars_model$results$RMSE)

# Get the coefficients of the optimal model
optimal_model_coeffs <- coef(mars_model$finalModel, s = mars_model$results$lambda[best_model_index])

# Print the coefficients
print(optimal_model_coeffs)
##                      (Intercept)                             h(31-bmi)
##                       -8.0457447                             7.2296297
##              h(bmi-31) * studyB  h(age-62) * h(bmi-31) * studyB
##                      -18.3936203                             4.6574999
##                      h(bmi-23.8) h(bmi-31) * h(112-LDL) * studyB
##                        7.4964488                             1.3157487
##                         vaccine1    smoking * h(bmi-31) * studyB
##                       -5.6486741                            12.0251817
##         h(bmi-23.8) * severity1  h(bmi-31) * h(LDL-87) * studyB
##                        1.3291655                             0.8267105
##                     h(1-smoking)                               gender1
##                       -3.3000670                            -3.3771125

mars_pred = predict(mars_model, newdata = x_test)
mars_mse = mean((mars_pred - y_test) ^ 2)
```

GAM

```
set.seed(2716)
gam_model = train(x = x_train,
                  y = y_train,
                  method = "gam",
                 #metric = "RMSE", by default
                  trControl = ctrl1)

summary(gam_model$finalModel)
##
## 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

# Calculate test RMSE of optimal model
gam_pred = predict(gam_model, x_test)

gam_mse = mean((gam_pred - y_test) ^ 2)
```

```
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))
```

```r
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

```
set.seed(2716)

# 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)
#rf_model  <- train(x = x_train,
#                   y = y_train,
#                   method = "rf",
#                   trControl = ctrl1,
#                   tuneGrid = tunegrid)

#model.rf <- train(x = x_train,
#                   y = y_train,
#                  method = "rf",
#                   #metric = "RMSE", by default
#                   trControl = trainControl(method = "cv", number = 10))


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

print(rf_model)
## Random Forest
##
## 2402 samples
##   12 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 2162, 2162, 2161, 2162, 2162, 2162, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
```

```
##   1     21.45597  0.3157134  12.80560
##   2     19.98460  0.3365832  12.22872
##   3     19.43467  0.3505647  12.09322
##   4     19.22088  0.3640621  12.08736
##   5     19.07825  0.3711168  12.08982
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 5.


rf_pred = predict(rf_model, x_test)

rf_mse = mean((rf_pred - y_test) ^ 2)
```

Resample

```
resamp =
  resamples(list(lasso = lasso_model,
                 gam = gam_model,
                 enet = enet_model,
               #  pls = pls.fit,
                 mars = mars_model,
               rf = rf_model))
summary(resamp)
##
## Call:
## summary.resamples(object = resamp)
##
## Models: lasso, gam, enet, mars, rf
## Number of resamples: 10
##
## MAE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lasso 12.19308 12.62708 12.94659 13.27005 13.62913 15.90894    0
## gam   11.57691 11.73191 12.43424 12.60549 12.98565 14.49104    0
## enet  12.15942 12.60133 12.83413 13.20336 13.55724 15.85049    0
## mars  10.89088 11.75502 11.98479 12.05695 12.47158 13.64296    0
## rf    11.05159 11.72795 12.10203 12.08982 12.36082 13.39686    0
##
## RMSE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lasso 16.23379 18.23425 20.92631 21.92098 25.14080 29.09945    0
## gam   16.08372 17.27037 19.24321 19.59045 21.38481 24.92036    0
## enet  16.19213 18.18064 20.91186 21.90863 25.15289 29.17824    0
## mars  14.64730 17.01084 18.42199 18.64837 20.45147 22.81323    0
## rf    15.20140 18.28549 19.30096 19.07825 20.28879 22.17660    0
##
## Rsquared
##             Min.    1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lasso 0.04823526 0.09324113 0.1280890 0.1129572 0.1339400 0.1422747    0
## gam   0.13502675 0.19667929 0.2719358 0.2929828 0.3944992 0.4462167    0
## enet  0.05391096 0.09335045 0.1278573 0.1133733 0.1349297 0.1408855    0
```

```
## mars  0.13026481 0.24311171 0.3585692 0.3627122 0.5139298 0.5734756    0
## rf    0.13928475 0.26906316 0.3805867 0.3711168 0.4843315 0.5384843    0
bwplot(resamp, metric = "RMSE")
```