P8106 Data ScienceII Homework2

Yueran Zhang(yz4188)

2023-03-05

In this exercise, we build nonlinear models using the "College" data. The dataset contains statistics for 565 US Colleges from a previous issue of US News and World Report. The response variable is the out-of-state tuition (Outstate).Partition the dataset into two parts: training data (80%) and test data (20%).

R Package

```
library(caret)
library(splines)
library(mgcv)
library(pdp)
library(earth)
library(tidyverse)
library(ggplot2)
library(gridExtra)
```

Import Dataset

```
set.seed(123)

# Load dataset + clean data
College = read.csv("/Users/yueranzhang/Desktop/DSII/HW2/DataSet/College.csv")[-1] %>%
janitor::clean_names() %>%
na.omit()

# Data Partition
RowTrain <- createDataPartition(y = College$outstate,
                                p = 0.8,
                                list = FALSE)

train_data <- College[RowTrain,]
test_data <- College[-RowTrain,]


# matrix of predictors
x <- model.matrix(outstate ~. , train_data) [,-1]
# vector of response
y <- train_data$outstate
```

Question A

Fit smoothing spline models using perc.alumni as the only predictor of Outstate for a range of degrees of freedom, as well as the degree of freedom obtained by generalized cross-validation, and plot the resulting fits. Describe the results obtained.

```
set.seed(123)

perc_alumni.grid <- seq(from =0, to = 70, by = 1)
fit.ss <- smooth.spline(train_data$perc_alumni, train_data$outstate,cv = TRUE)
```

```
## Warning in smooth.spline(train_data$perc_alumni, train_data$outstate, cv =
## TRUE): cross-validation with non-unique 'x' values seems doubtful
```

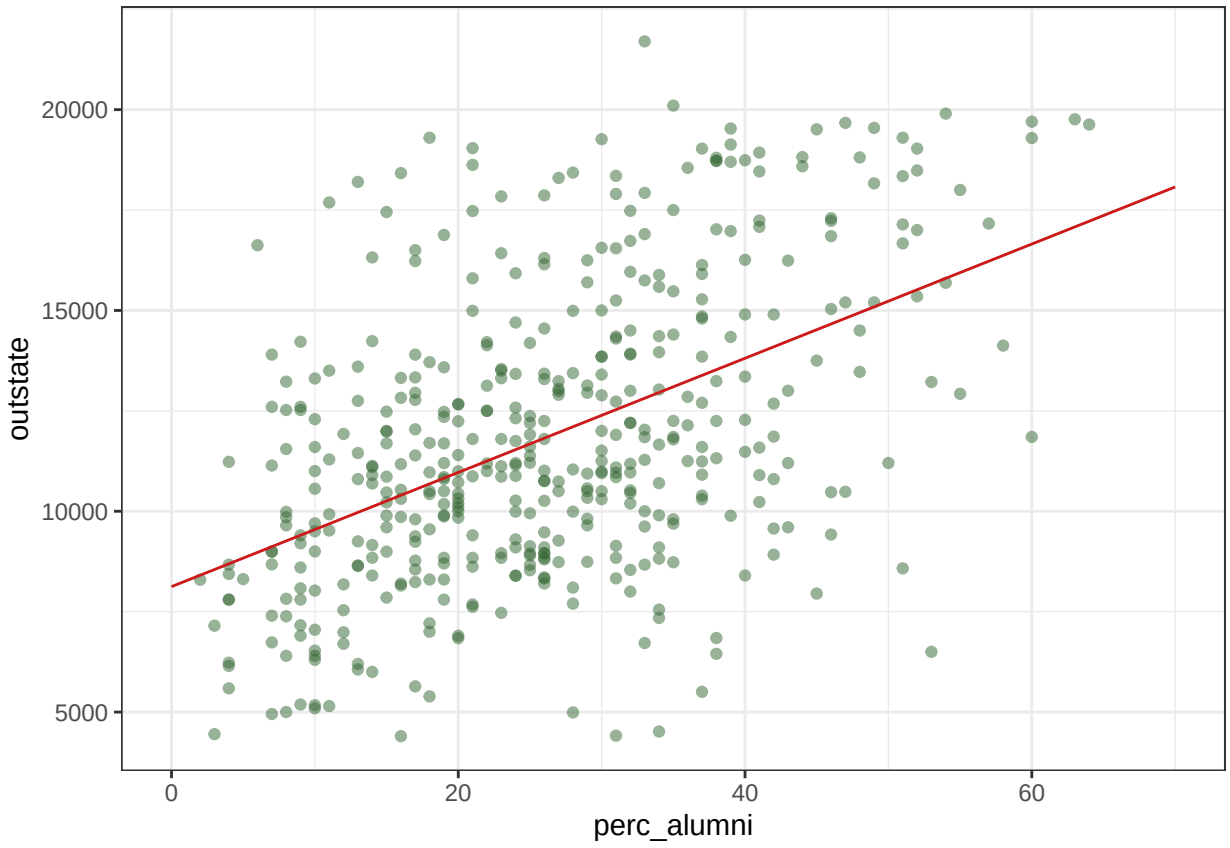
```
fit.ss$df
```

```
## [1] 2.00025
```

```
fit.ss$lambda
```

```
## [1] 2477.12
```

```
# plot the fit

pred.ss <- predict(fit.ss,
                   x =perc_alumni.grid)

pred.ss.df <- data.frame(pred = pred.ss$y,
                         perc_alumni = perc_alumni.grid)

p <- ggplot(data = train_data, aes(x = perc_alumni, y = outstate)) +
     geom_point(color = rgb(.2, .4, .2, .5))

p +
geom_line(aes(x = perc_alumni.grid, y = pred), data = pred.ss.df,
          color = rgb(.8, .1, .1, 1)) + theme_bw()
```



- The degree of freedom that we obtained by generalized cross-validation is 2.0002496. From the above plot, the smoothing spline is nearly to a linear line, or that's to say this model fits the data quite well.
- It is noticed that as we get the λ value is 2477.120298, which is pretty large, so for the function estimate f_λ is essentially constrained to have a zero penalty, and it is forced to be smoother.

Question B

Fit a generalized additive model (GAM) using all the predictors. Does your GAM model include all the predictors? Plot the results and explain your findings. Report the test error.

```
set.seed(123)

ctrl <- trainControl(method = "cv", number = 10)

gam.fit <- train(x, y,
                 method = "gam",
                 tuneGrid = data.frame(method = "GCV.Cp", select = TRUE),
                 trControl = ctrl)
```

```
## Warning: model fit failed for Fold06: method=GCV.Cp, select=TRUE Error in magic(G$y, G$X, msp, G$S, G
##   magic, the gcv/ubre optimizer, failed to converge after 400 iterations.
```

```
## Warning: model fit failed for Fold08: method=GCV.Cp, select=TRUE Error in magic(G$y, G$X, msp, G$S, G
##   magic, the gcv/ubre optimizer, failed to converge after 400 iterations.

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
gam.fit
```

```
## Generalized Additive Model using Splines
##
## 453 samples
##  16 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 408, 408, 407, 409, 407, 409, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1865.194  0.7511248  1440.444
##
## Tuning parameter 'select' was held constant at a value of TRUE
## Tuning
##  parameter 'method' was held constant at a value of GCV.Cp
```

```
gam.fit$finalModel
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ s(perc_alumni) + s(terminal) + s(books) + s(ph_d) +
##     s(grad_rate) + s(top10perc) + s(top25perc) + s(s_f_ratio) +
##     s(personal) + s(p_undergrad) + s(enroll) + s(room_board) +
##     s(accept) + s(f_undergrad) + s(apps) + s(expend)
##
## Estimated degrees of freedom:
## 2.760 0.384 6.116 0.185 3.276 6.828 0.000
## 2.901 5.326 0.000 1.837 6.711 4.295 5.234
## 3.693 5.341  total = 55.89
##
## GCV score: 2822251
```

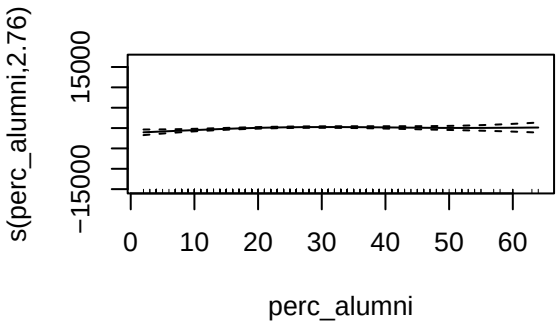
- From the above output, there are 16 predictors, which means the GAM model include all the predictors.

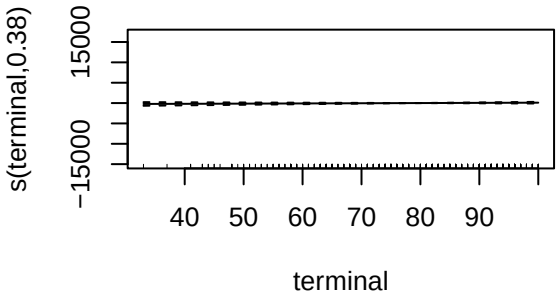
```
set.seed(123)

summary(gam.fit)
```

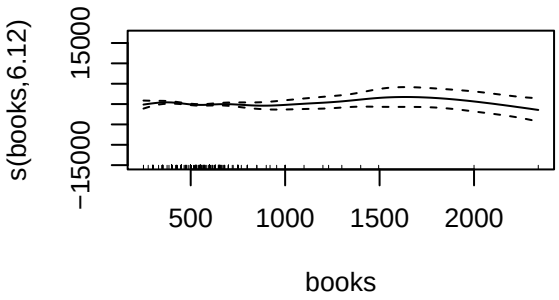
```
##
```

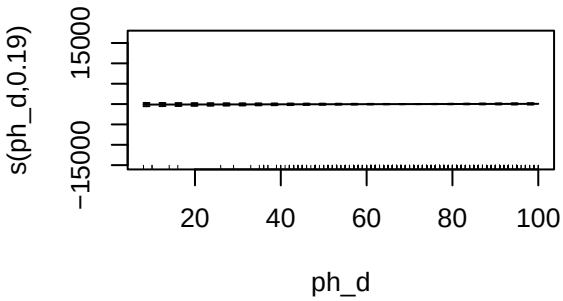
```
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ s(perc_alumni) + s(terminal) + s(books) + s(ph_d) +
##     s(grad_rate) + s(top10perc) + s(top25perc) + s(s_f_ratio) +
##     s(personal) + s(p_undergrad) + s(enroll) + s(room_board) +
##     s(accept) + s(f_undergrad) + s(apps) + s(expend)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11845.7       73.9   160.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                       edf Ref.df      F  p-value
## s(perc_alumni) 2.760e+00      9  1.422  0.00168 **
## s(terminal)    3.844e-01      9  0.103  0.09711 .
## s(books)       6.116e+00      9  1.248  0.06505 .
## s(ph_d)        1.854e-01      9  0.043  0.11295
## s(grad_rate)   3.276e+00      9  1.640  0.00109 **
## s(top10perc)   6.828e+00      9  1.347  0.06675 .
## s(top25perc)   1.156e-06      9  0.000  0.61319
## s(s_f_ratio)   2.901e+00      9  0.423  0.23151
## s(personal)    5.326e+00      9  1.501  0.01463 *
## s(p_undergrad) 8.836e-07      9  0.000  0.79884
## s(enroll)      1.837e+00      9  1.847 1.10e-05 ***
## s(room_board)  6.711e+00      9  7.519  < 2e-16 ***
## s(accept)      4.295e+00      9  2.464 5.34e-06 ***
## s(f_undergrad) 5.234e+00      9  1.646  0.00142 **
## s(apps)        3.693e+00      9  0.930  0.02295 *
## s(expend)      5.341e+00      9 18.357  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.815   Deviance explained = 83.7%
## GCV = 2.8223e+06  Scale est. = 2.4741e+06  n = 453
```

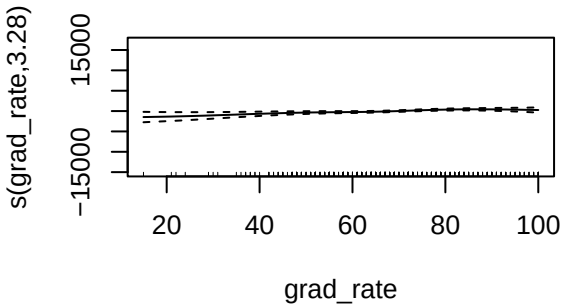
```r
# Plot of each predictor versus the outcome variable (outstate)
plot(gam.fit$finalModel, pages = 4)
```

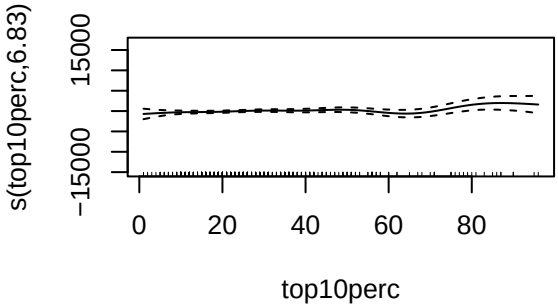

s(perc_alumni,2.76)
−15000
15000
0
10
20
30
40
50
60
perc_alumni

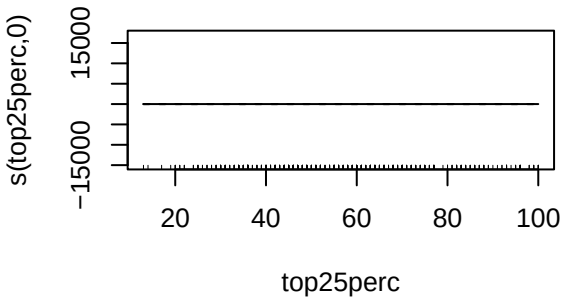

s(terminal,0.38)
−15000
15000
40
50
60
70
80
90
terminal

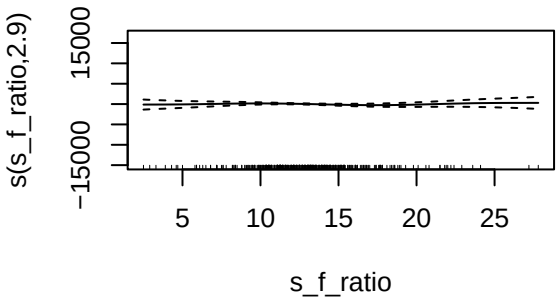

s(books,6.12)
−15000
15000
500
1000
1500
2000
books

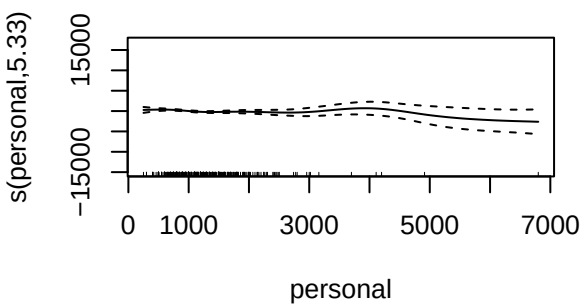

s(ph_d,0.19)
−15000
15000
20
40
60
80
100
ph_d

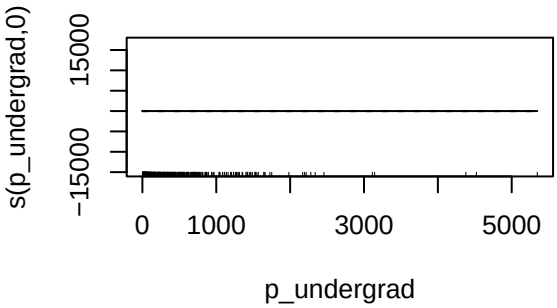

s(grad_rate,3.28)
15000
−15000
20
40
60
80
100
grad_rate

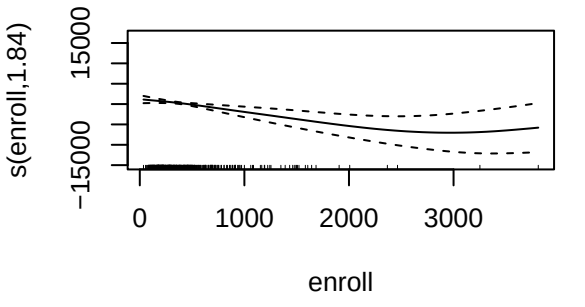

s(top10perc,6.83)
15000
−15000
0
20
40
60
80
top10perc

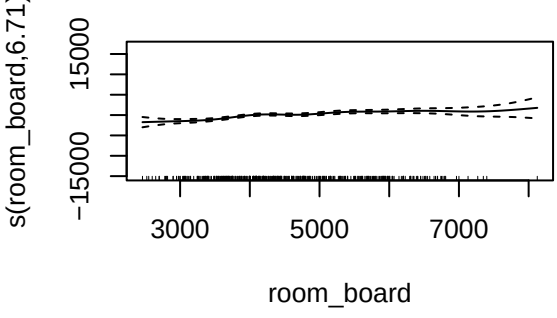

s(top25perc,0)
15000
−15000
20
40
60
80
100
top25perc

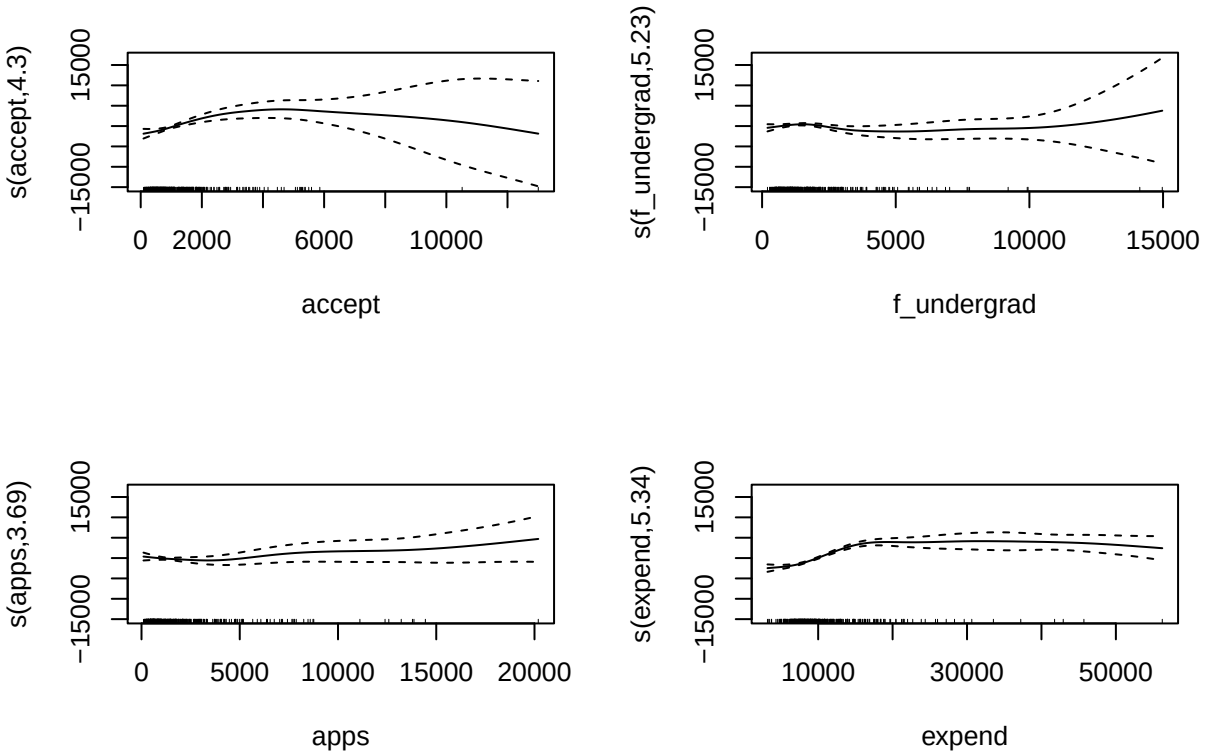

s(s_f_ratio,2.9)
15000
−15000
5
10
15
20
25
s_f_ratio


s(personal,5.33)
15000
−15000
0
1000
3000
5000
7000
personal


s(p_undergrad,0)
15000
−15000
0
1000
3000
5000
p_undergrad


s(enroll,1.84)
15000
−15000
0
1000
2000
3000
enroll


s(room_board,6.71)
15000
−15000
3000
5000
7000
room_board



- According to p-value, for some predictors, there is not sufficient evidence in the data to conclude they are significant association with the outcome variable outstate at the 5% significance level, like **terminal**, **ph_d**,**top25perc**, and **p_undergrad**. However, some of the predictors seems to have linear relationship with the model, such as **grad_rate** and **personal**.
- The Deviance explained by the model is 83.7%, and the adjusted R-Square is 81.5%, that showing a high level of correlation. The GAM model fits the data pretty well.

```
set.seed(123)

# Test Error(MSE)
testdata_x = test_data %>%
select(-outstate)

gam.pred <- predict(gam.fit, newdata = testdata_x)

mean((gam.pred - test_data$outstate)^2)
```

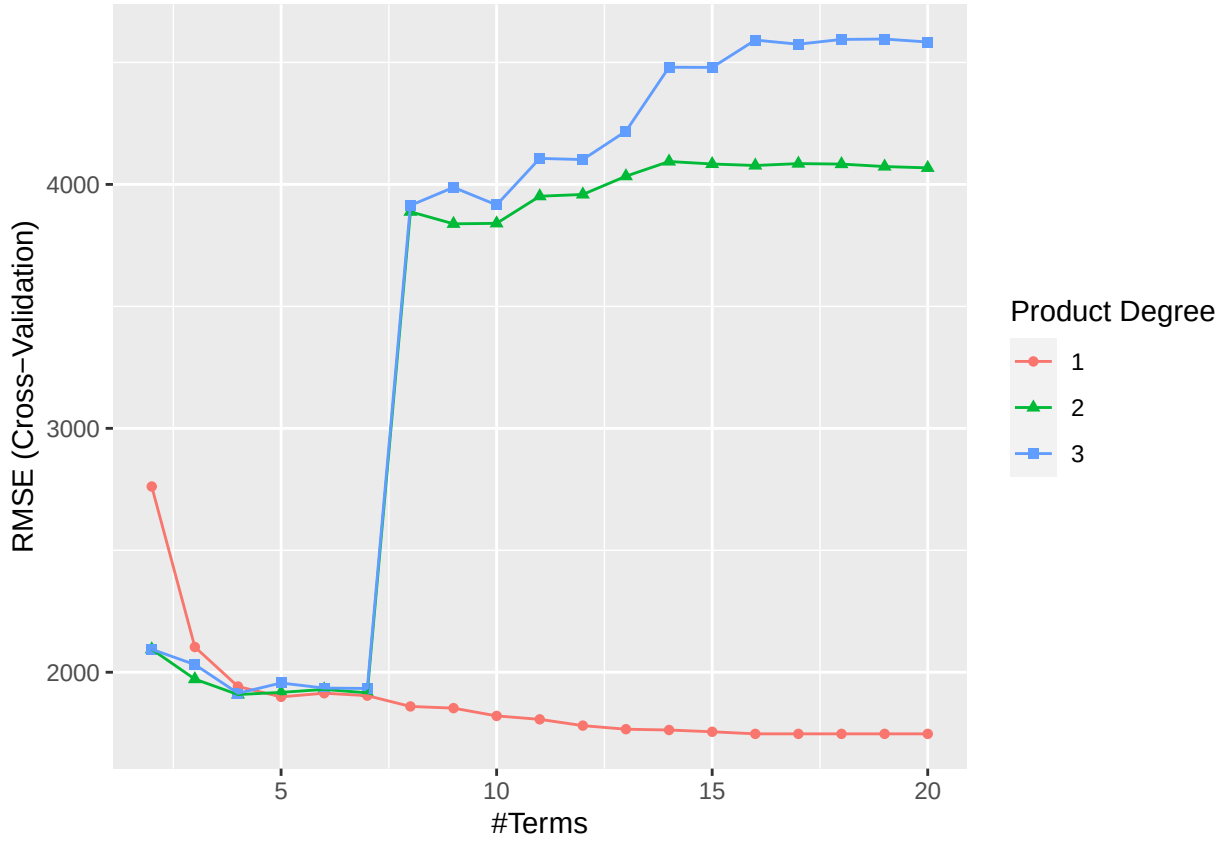
```
## [1] 9736424
```

- The test error of the GAM model is 9.7364235×10^6.

Question C

Train a multivariate adaptive regression spline (MARS) model using all the predictors. Report the final model. Present the partial dependence plot of an arbitrary predictor in your final model. Report the test error.

```
mars_grid <- expand.grid(degree = 1:3,
                         nprune = 2:20)

set.seed(123)
mars.fit <- train(x, y,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl)

# Plot of gird tuning
ggplot(mars.fit)
```




```
# final model
mars.fit$bestTune
```

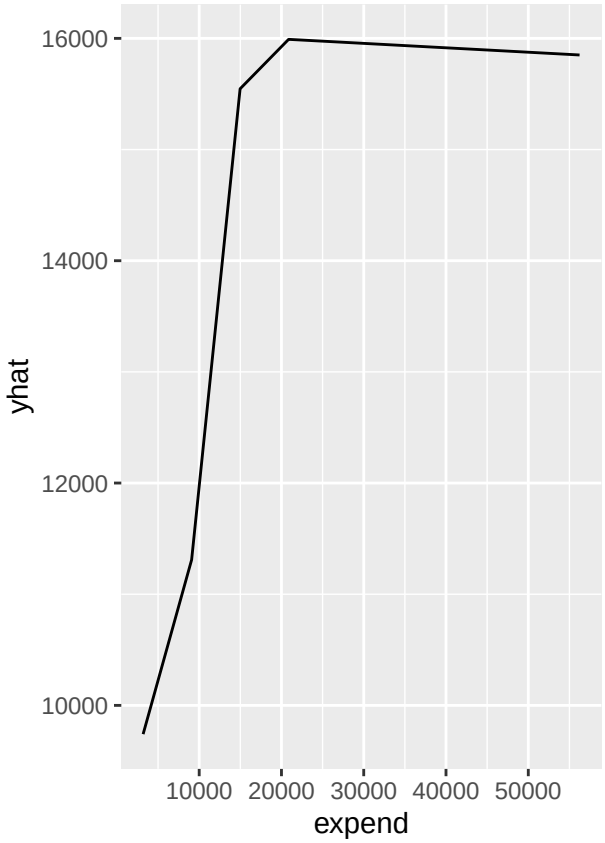
```
##    nprune degree
## 15     16      1
```

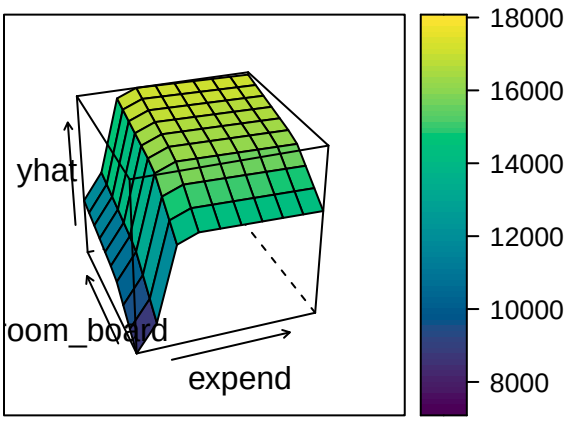
```r
# coefficient of the MARS model
coef(mars.fit$finalModel)
```

```
##         (Intercept)     h(expend-15622)  h(room_board-4460)  h(4460-room_board)
##       10684.3159852          -0.7227653           0.3113264          -1.1274658
##     h(79-grad_rate)    h(1300-personal) h(f_undergrad-1350) h(1350-f_undergrad)
##         -28.9480175           1.0471977          -0.4456624          -1.2719896
##        h(apps-2694)   h(21-perc_alumni)      h(expend-6898)       h(862-enroll)
##           0.3774909         -87.2568633           0.7187916           4.9263485
##      h(2165-accept)
##          -2.0063276
```

- From the above output, the final model has 16 coefficients, with degree of freedom is 1.
- We observed that variables such as **grad_rate** , **f_undergrad** and **perc_alumni** with larger absolute value, which means these variable may more likely to change the mean in the response given a one unit change in these predictor.

```r
# Plot of expend variable
p1 <- pdp::partial(mars.fit, pred.var = c("expend"), grid.resolution = 10) %>% autoplot()

# Plot of expend and room board variable
p2 <- pdp::partial(mars.fit, pred.var = c("expend", "room_board"),
                   grid.resolution = 10) %>%
      pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE,
                       screen = list(z = 20, x = -60))

grid.arrange(p1, p2, ncol = 2)
```





- Here are two PDPs (Partial Dependence Plot) plots, the left one is with the variable **expend**, and the right side is for **expend** and **room_board**.

```
set.seed(123)

# Test Error(MSE)
testdata_x = test_data %>%
select(-outstate)

mars.pred <- predict(mars.fit, newdata = testdata_x)

mean((mars.pred  - test_data$outstate)^2)
```

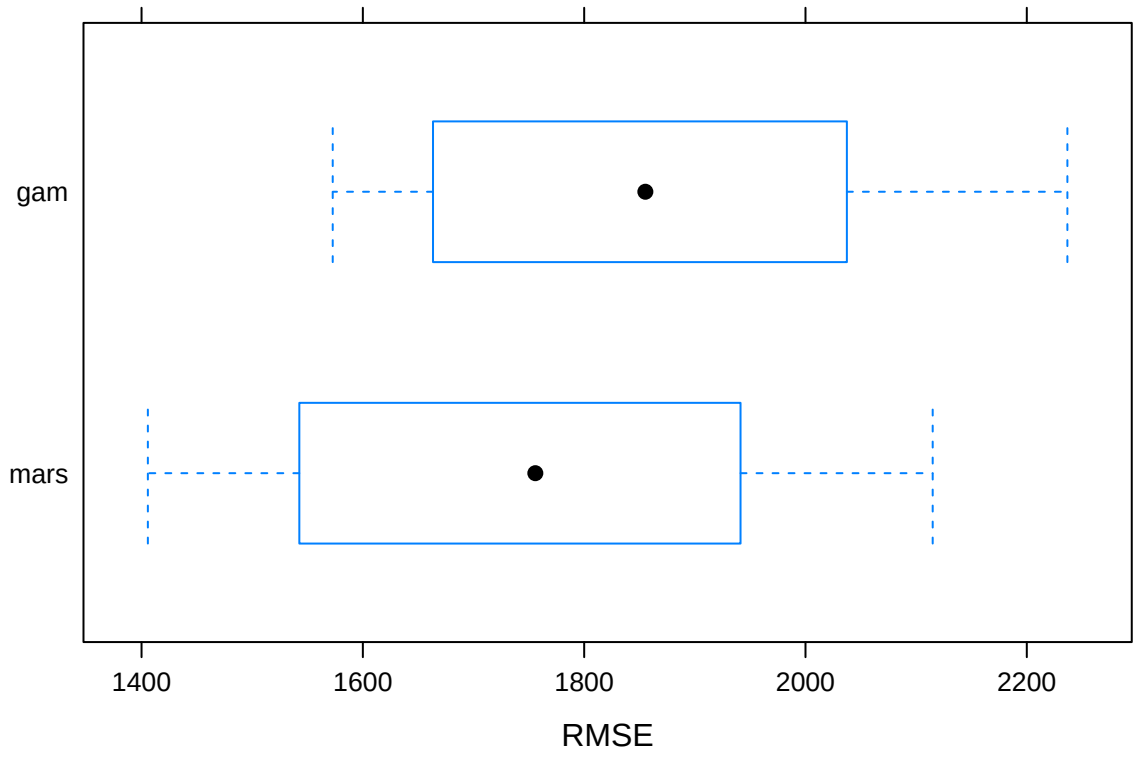
```
## [1] 2979788
```

- The test error for MARS model 2.9797881×10^6.

Question D

In this data example, do you prefer the use of MARS model over a linear model when predicting the out-of-state tuition? Why? For general applications, do you think MARS is a better approach compared to a linear model?

```
set.seed(123)

bwplot(resamples(list(mars = mars.fit,
                      gam = gam.fit)), metric = "RMSE")
```



- As for response predicting model, we would choose the model with the lowest error on predicting the test set. From the above questions, we see that Test Error of GAM model is 9.7364235×10^6 and Test Error of MARS model is 2.9797881×10^6. MARS model got the smaller test error.Besides, from the above plot for RMSE, which is the root of MSE. We prefer the model with lower values of RMSE since this indicate better fit. As we can see that MARS model is with lower values. In this circumstances, I would prefer the use of MARS model over a linear model when predicting the out-of-state tuition.
- For general applications, I believe MARS model would be a better approach compared to a linear model. The pros of linear regression is its simplicity, as it assumes a linear relationship between inputs and outputs. The interaction between metrics in the real-world is often non-linear, which means that simple linear regression cannot always give us a good approximation of outputs given the inputs. MARS is to imagine it as an ensemble of linear functions joined together by one or more hinge functions. What is more, if we go up in dimensions and build and compare models using 2 independent variables, multiple

linear regression creates a prediction plane that looks like a flat sheet of paper. Meanwhile, MARS takes that sheet of paper and folds it in a few places using hinge functions, enabling a better fit to the data.