# hw2_spline_GAM_MARS

Cary Ni

2023-02-26

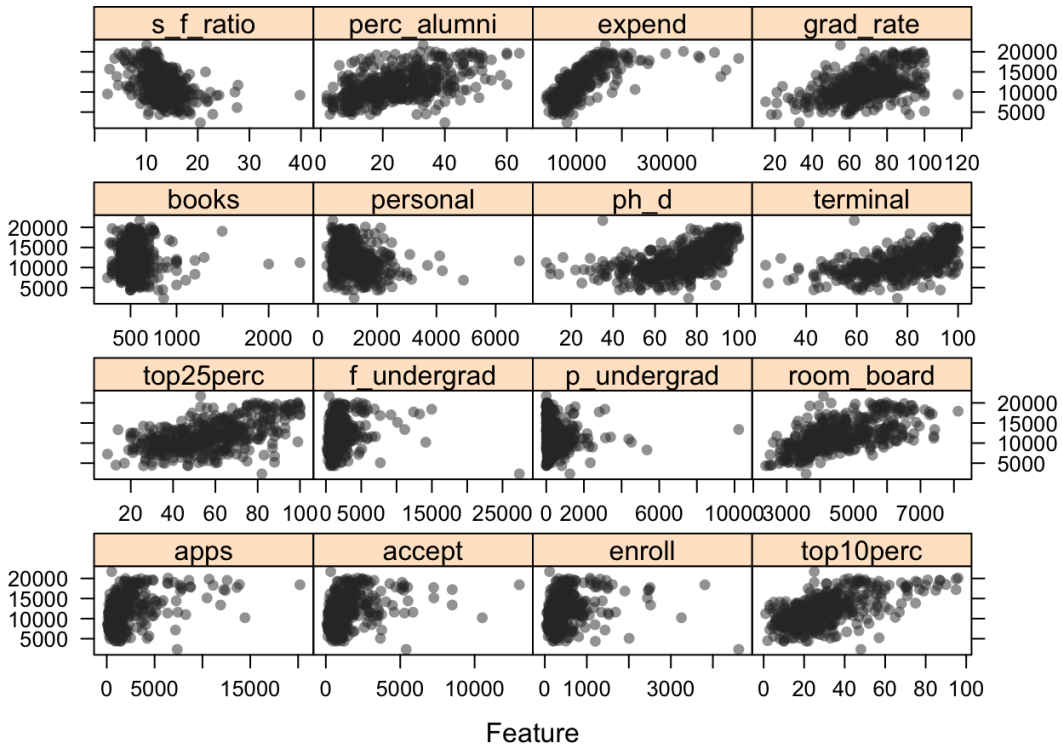
```
# load dataset
college_df = read_csv("College.csv", show_col_types = FALSE) %>%
  janitor::clean_names() %>%
  na.omit()

# data partition
set.seed(2023)
index_train = createDataPartition(y = college_df$outstate, p = 0.8, list = FALSE)
train_set = college_df[index_train, ]
test_set = college_df[-index_train, ]
pred_x = model.matrix(outstate~.-college, data = train_set)[, -1]
resp_y = train_set %>% pull(outstate)
test_x = model.matrix(outstate~.-college, data = test_set)[, -1]
test_y = test_set %>% pull(outstate)

# create a quick function to calculate test mse
get_test_mse = function(input_model, x_test, y_test) {
  predict_value = predict(input_model, newdata = x_test)
  test_mse = mean((predict_value - y_test)^2)
  return(test_mse)
}
```

# Create feature plot to examine the relationship between predictors and response variable

```
theme1 <- trellis.par.get()
theme1$plot.symbol$col = rgb(.2, .2, .2, .5)
theme1$plot.symbol$pch = 16
theme1$plot.line$col = rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
trellis.par.set(theme1)
featurePlot(x = pred_x,
            y = resp_y,
            plot = "scatter",
            layout = c(4, 4))
```




It can be seen from the plots that `s_f_ratio`, `per_alumni`, `grad_rate`, and `room_board` are most likely to be linearly correlated with the response variable `outstate` while the linear relationship is not apparent for the rest of the predictors.

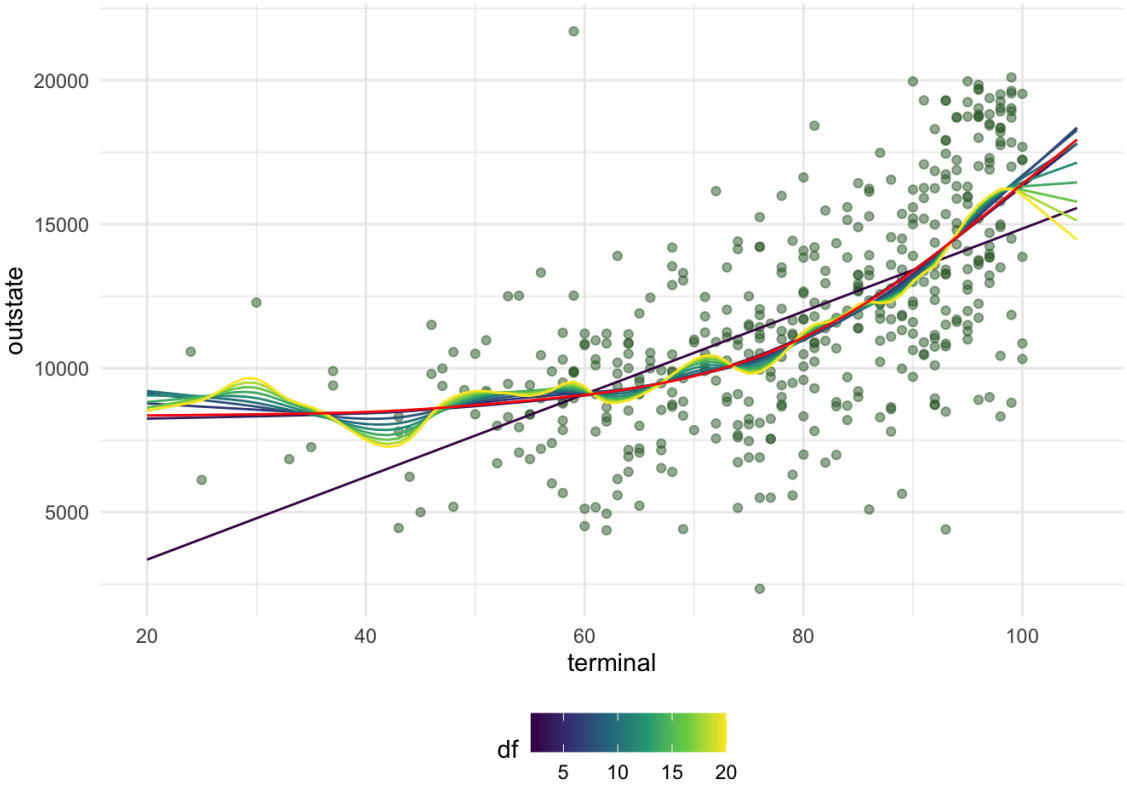
# Fit smoothing spline models

```
terminal_grid = seq(20, 105, by = 1)
# fit smoothing spline models with pre-specified df

# fit smoothing spline model with generalized cross-validation
ss_model = smooth.spline(train_set$terminal, train_set$outstate, cv = FALSE)
# show the resulting degree of freedom from gcv
ss_model$df
```

```
## [1] 4.287189
```

```
# draw the line with df from gcv
pred_gcv = predict(object = ss_model, x=terminal_grid)
post_gcv = data.frame(terminal=pred_gcv$x, outstate=pred_gcv$y, df = ss_model$df)
# fit the models with different df
model_list = list()
pred = list()
post = list()
for (i in 1:10) {
  model_list[[i]] = smooth.spline(train_set$terminal, train_set$outstate, df = 2*i)
  pred[[i]] = predict(object = model_list[[i]], x=terminal_grid)
  post[[i]] = data.frame(terminal=pred[[i]]$x, outstate=pred[[i]]$y, df = 2*i)
}
# combine the data from models
combine_df = as_tibble_col(post) %>% unnest(value)
# show the training set data points
p = ggplot(data = train_set, aes(x = terminal, y = outstate)) + geom_point(color= rgb(.2, .4, .2, .5))
# show the smoothing splines with different df, the red one is obtained from gcv
p + geom_line(aes(x = terminal, y = outstate, group = df, color = df), data = combine_df) + geom_line(aes(x
= terminal, y = outstate), color = 'red', data = post_gcv)
```




a. The plot with resulting fits with different degree of freedom shows that the smoothing spline model with small degree of freedom is more rigid and close to linear regression. As the degree of freedom increases, the smoothing spline model becomes more flexible with wiggling line that fits more observations. The model obtained from generalized cross-validation is marked red in the plot with a degree of freedom 4.287189.

# Fit Generalized Additive Models GAM

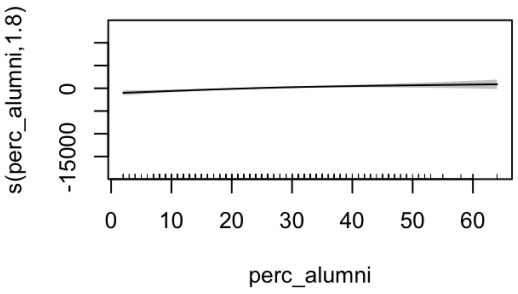
```
# set train method
ctrl_1 = trainControl(method = "cv", number = 10)
# fit gam model with all predictors
set.seed(1)
tune_Grid = data.frame(method = "GCV.Cp", select = c(TRUE, FALSE))
gam_model = train(pred_x, resp_y,
                  method = "gam",
                  trControl = ctrl_1)
gam_model$bestTune
```

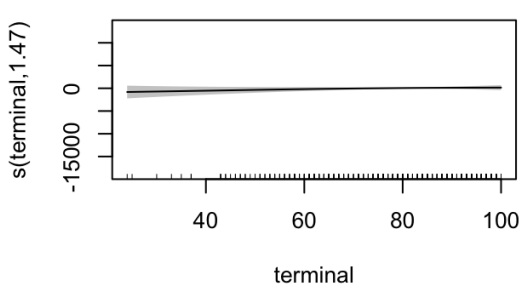
```
##   select method
## 1  FALSE GCV.Cp
```

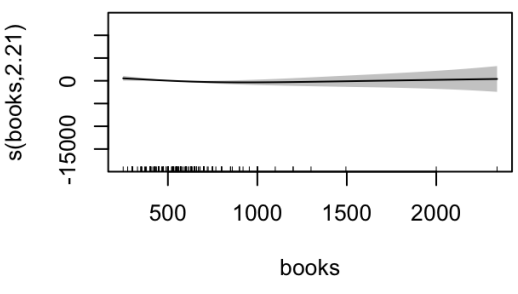
```
summary(gam_model$finalModel)
```

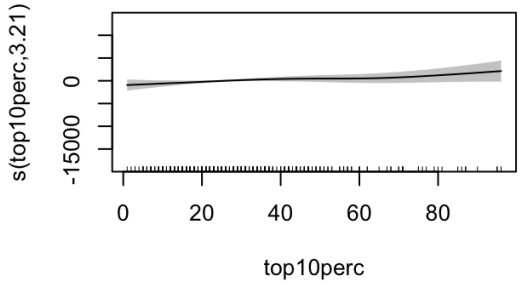
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ s(perc_alumni) + s(terminal) + s(books) + s(top10perc) +
##     s(ph_d) + s(grad_rate) + s(top25perc) + s(s_f_ratio) + s(personal) +
##     s(p_undergrad) + s(enroll) + s(room_board) + s(accept) +
##     s(f_undergrad) + s(apps) + s(expend)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11775.8       78.4   150.2   <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) 1.796  2.271  7.716 0.000348 ***
## s(terminal)    1.467  1.806  1.054 0.449896
## s(books)       2.209  2.757  2.274 0.137179
## s(top10perc)   3.212  4.092  1.400 0.241800
## s(ph_d)        5.335  6.423  1.686 0.118984
## s(grad_rate)   3.830  4.801  2.608 0.027104 *
## s(top25perc)   1.000  1.000  1.457 0.228136
## s(s_f_ratio)   3.689  4.641  1.483 0.230693
## s(personal)    1.000  1.000  4.584 0.032864 *
## s(p_undergrad) 1.000  1.000  0.130 0.719133
## s(enroll)      1.000  1.000  5.631 0.018106 *
## s(room_board)  1.998  2.540 18.963  < 2e-16 ***
## s(accept)      1.000  1.000  8.817 0.003159 **
## s(f_undergrad) 6.297  7.361  4.548  6.3e-05 ***
## s(apps)        1.000  1.000  5.880 0.015747 *
## s(expend)      5.794  6.984 16.246  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.797   Deviance explained = 81.6%
## GCV = 3.0737e+06  Scale est. = 2.7845e+06  n = 453
```

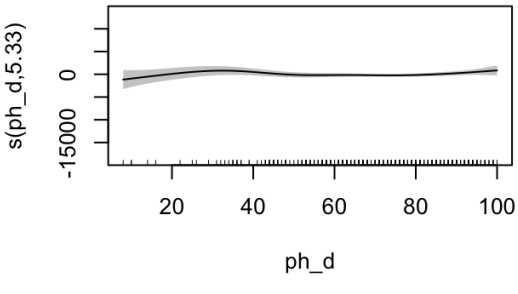
```
# plot.gam for each of the predictors
par(mfrow = c(2, 2))
plot(gam_model$finalModel, shade = TRUE)
```

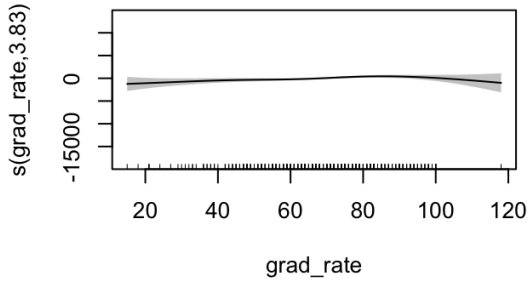

s(perc_alumni,1.8)
-15000
0
0
10
20
30
40
50
60
perc_alumni

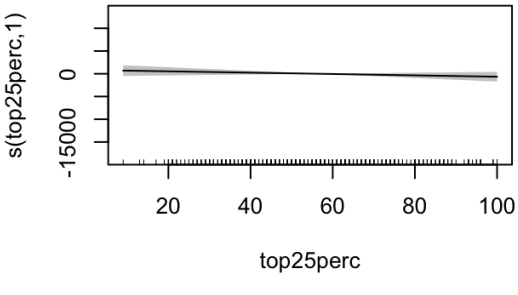

s(terminal,1.47)
-15000
0
40
60
80
100
terminal

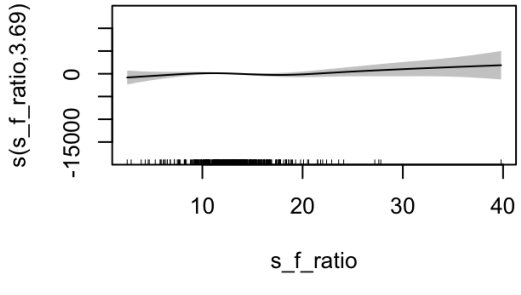

s(books,2.21)
-15000
0
500
1000
1500
2000
books

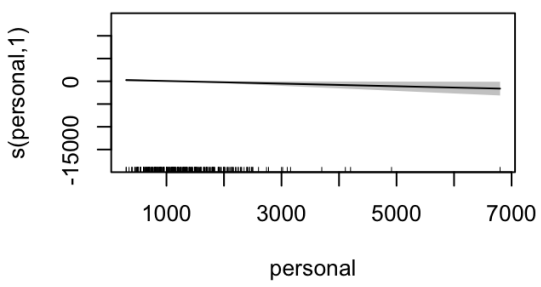

s(top10perc,3.21)
-15000
0
0
20
40
60
80
top10perc

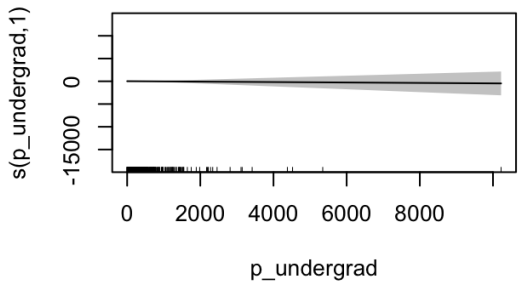

s(ph_d,5.33)
-15000
0
20
40
60
80
100
ph_d

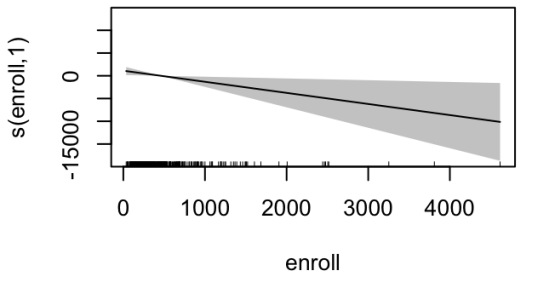

s(grad_rate,3.83)
-15000
0
20
40
60
80
100
120
grad_rate

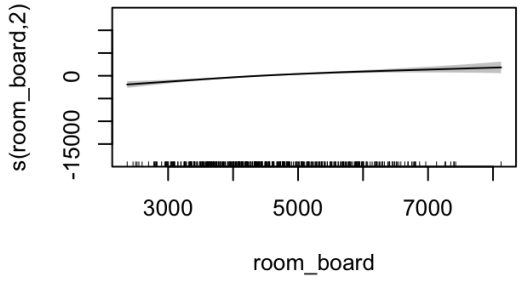

s(top25perc,1)
-15000
0
20
40
60
80
100
top25perc

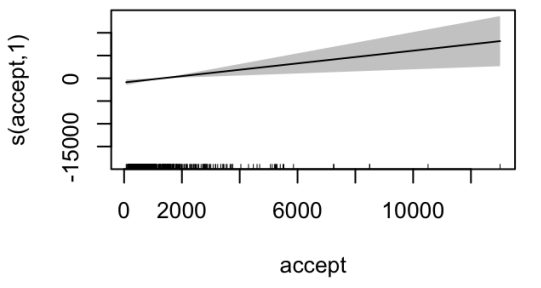

s(s_f_ratio,3.69)
-15000
0
10
20
30
40
s_f_ratio

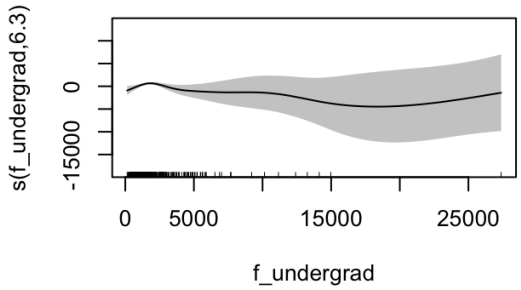


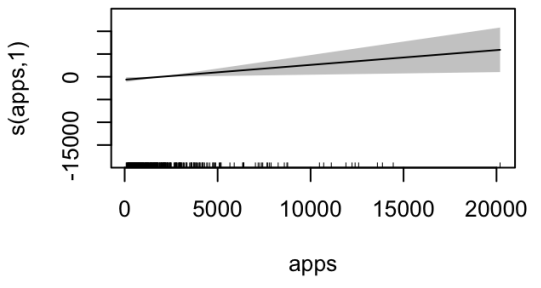



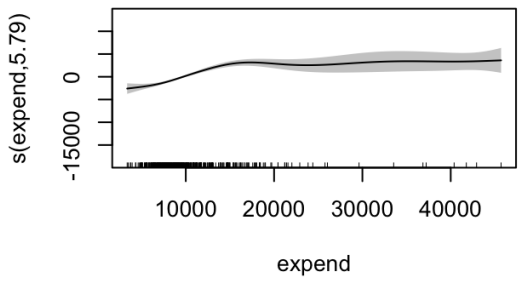



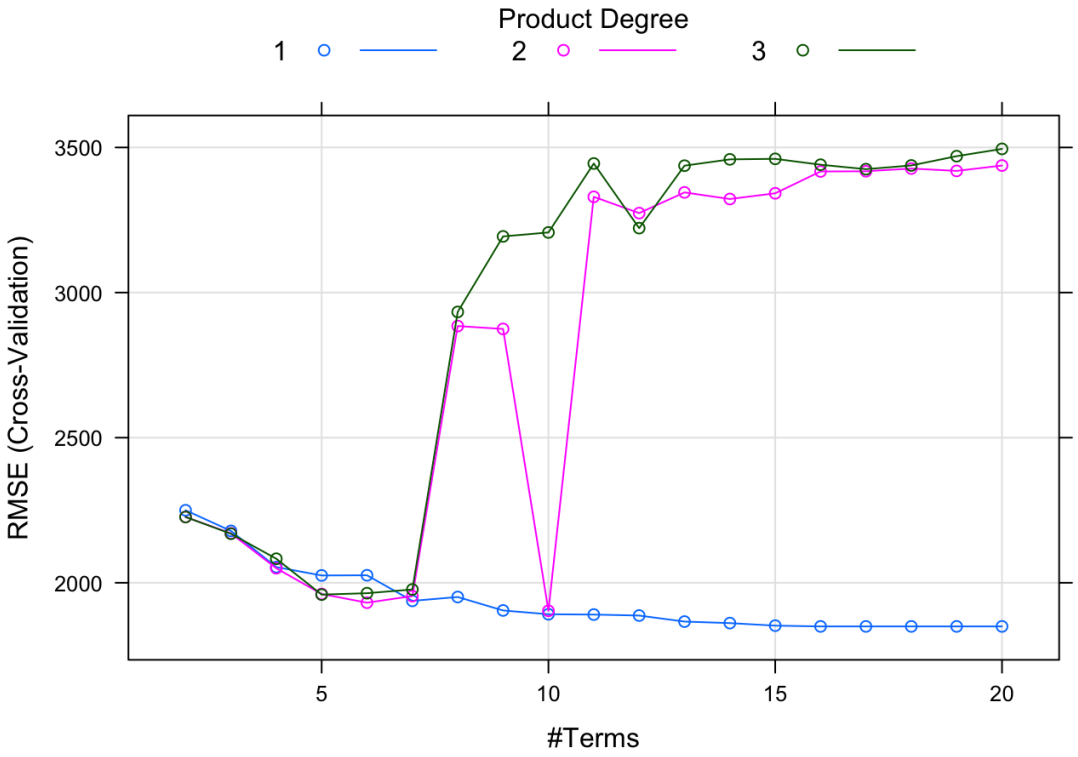

```
# get test mse for gam model
get_test_mse(gam_model, test_x, test_y)
```

```
## [1] 1922868
```

b. As shown in the plot and the summary of the generalized additive model, while all of the predictors are included, `top25perc`, `personal`, `p_undergrad`, `enroll`, `accept`, and `apps` are the predictors with the effective degree of freedom of 1, suggesting an linear correlation between those covariates and the response variable. `ph_d`, `f_undergrad`, and `expend` are the variables with largest effective degree of freedom which are over 5, indicating greatest flexibility in fitting smoothing splines. The test MSE is 1.92e6.

# Fit multivariate adaptive regression spline model (MARS)

```
set.seed(1)
# Set tuning parameters
mars_grid = expand.grid(degree = 1:3, nprune = 2:20)
# Fit MARS model
mars_model = train(pred_x, resp_y,
                   method = "earth",
                   tuneGrid = mars_grid,
                   trControl = ctrl_1)
# Plot the model
plot(mars_model)
```



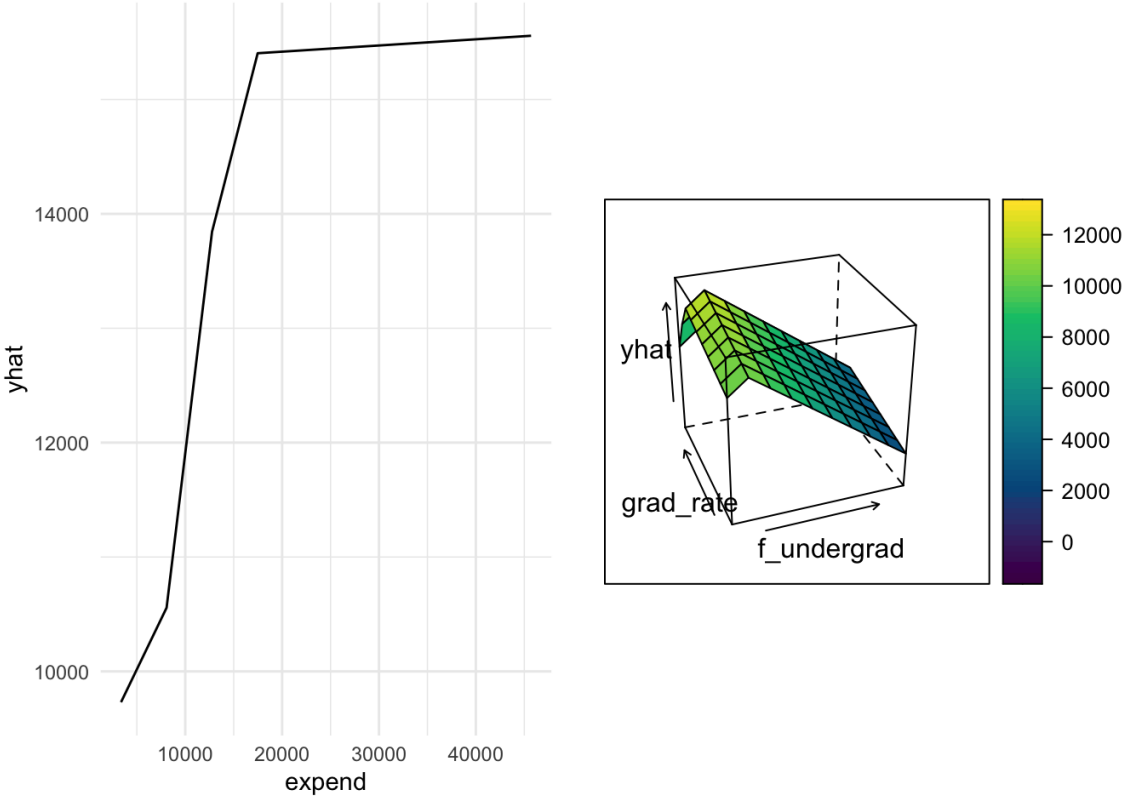

```
mars_model$bestTune
```

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

```
# Report the final model
summary(mars_model$finalModel)
```

```
## Call: earth(x=matrix[453,16], y=c(12280,12960,7...), keepxy=TRUE, degree=1,
##             nprune=16)
##
##                     coefficients
## (Intercept)            11104.8034
## h(apps-3768)               0.3937
## h(2342-accept)            -1.9205
## h(903-enroll)              4.8535
## h(1405-f_undergrad)       -1.4530
## h(f_undergrad-1405)       -0.3555
## h(4440-room_board)        -1.1100
## h(room_board-4440)         0.3591
## h(1300-personal)           1.0473
## h(ph_d-95)               336.2820
## h(22-perc_alumni)        -87.1010
## h(expend-6889)             0.6995
## h(expend-14980)           -0.6941
## h(98-grad_rate)          -21.6301
## h(grad_rate-98)         -232.0382
##
## Selected 15 of 22 terms, and 10 of 16 predictors (nprune=16)
## Termination condition: RSq changed by less than 0.001 at 22 terms
## Importance: expend, room_board, perc_alumni, accept, f_undergrad, apps, ...
## Number of terms at each degree of interaction: 1 14 (additive model)
## GCV 3066313    RSS 1216886154    GRSq 0.777412    RSq 0.8041351
```

```
# Build partial dependence plot for expend and f_undergrad & grad_rate
p1 = pdp::partial(mars_model, pred.var = c("expend"),
                  grid.resolution = 10) %>% autoplot()
p2 = pdp::partial(mars_model, pred.var = c("f_undergrad", "grad_rate"),
                  grid.resolution = 10) %>% pdp::plotPartial(
                    levelplot = FALSE, zlab = "yhat", drape = TRUE,
                    screen = list(z = 20, x = -60))
gridExtra::grid.arrange(p1, p2, ncol = 2)
```



```
# get test mse for mars model
get_test_mse(mars_model, test_x, test_y)
```

```
## [1] 1873834
```

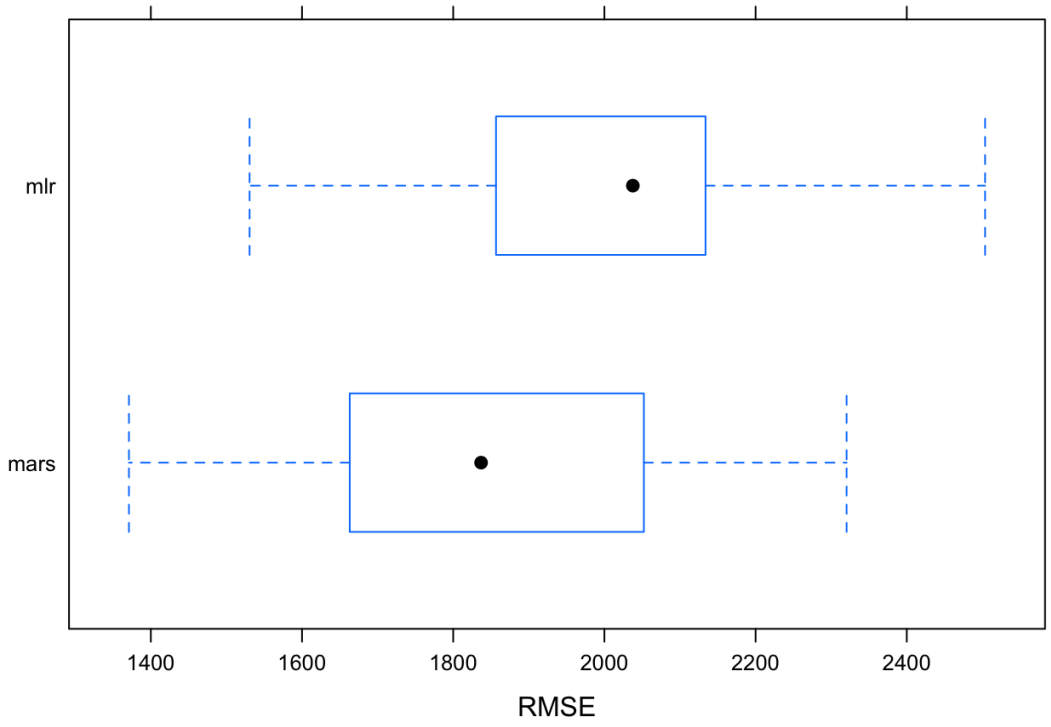
c. The plot of model selection shows that the final multivariate adaptive regression spline model uses 10 of the 16 original predictors with degree of 1 and 15 of 22 terms in total including the intercept. In the example partial dependence plots above, two knots at 6889 and 14980 can be found in `expend` , a knot at 1405 for `f_undergrad` and 98 for `grad_rate` can be seen in the three dimension plot on the right. The reported test MSE is 1.87e6.

# Compare MARS model to a linear model(without regularization)

```
# model comparison to MLR model (without regularization)
set.seed(1)
lm_model = train(pred_x, resp_y,
                 method = "lm",
                 trControl = ctrl_1)
summary(lm_model)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6153.8 -1263.1    -1.2  1330.6  9682.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 778.72839  962.22710   0.809  0.41879
## apps         -0.06346    0.13050  -0.486  0.62700
## accept        1.25500    0.21241   5.908 6.97e-09 ***
## enroll       -3.07254    1.11232  -2.762  0.00598 **
## top10perc    29.01324   15.91381   1.823  0.06897 .
## top25perc     2.38976   12.46495   0.192  0.84805
## f_undergrad  -0.03217    0.18298  -0.176  0.86053
## p_undergrad  -0.18351    0.14882  -1.233  0.21820
## room_board    0.94529    0.11228   8.419 5.53e-16 ***
## books        -0.11212    0.54877  -0.204  0.83820
## personal     -0.49356    0.15117  -3.265  0.00118 **
## ph_d          9.72973   10.43866   0.932  0.35181
## terminal     22.20171   11.80190   1.881  0.06061 .
## s_f_ratio   -20.54622   32.17627  -0.639  0.52345
## perc_alumni  44.30332    9.67794   4.578 6.14e-06 ***
## expend        0.17752    0.02935   6.048 3.15e-09 ***
## grad_rate    18.23778    6.97291   2.616  0.00922 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1984 on 436 degrees of freedom
## Multiple R-squared:  0.7239, Adjusted R-squared:  0.7138
## F-statistic: 71.45 on 16 and 436 DF,  p-value: < 2.2e-16
```

```
# compare model performance through sampling method
resamp = resamples(list(
  mars = mars_model,
  mlr = lm_model
))
# plot resampling rmse
bwplot(resamp, metric = "RMSE")
```




d. It can be seem that the multivariate adaptive regression spline (MARS) model has much lower cross-validation error than multiple linear regression (MLR) model from the boxplot above. In addition, MARS model also has fewer predictors estimated than MLR model (without regularization). Therefore, MARS model is more favored in predicting out-of-state tuition in this case study. For general application focusing on predictability instead of interpretability, MARS could not only capture the linear relationship between predictors and response variable as linear model when degree is one, but also capture the nonlinear pattern with degree larger than one. Thus, MARS is more favorable than linear model in terms of predictability in general context though linear model may be more informative because of its better interpretability.