# Comparison of Generalized additive models (GAM) and Lasso regression models

The dataset contains data about prostate cancer patients with information on the size of the prostate, the age of the patient, a blood marker (lpsa) and so on. The response variable is a score (Cscore) on the progression of the cancer after detailed study of the tumor pathology.

**Tasks** - Study and describe the predictor variables. Do you see any issues that are relevant for making predictions? - Make an appropriate LASSO model, with the appropriate link and error function, and evaluate the prediction performance. - Fit a model with appropriate non-linear effects. Report a comparison of performance to LASSO and explain what you find.

```r
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.6.3
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
## Warning: package 'readr' was built under R version 3.6.3
```

```
## Warning: package 'purrr' was built under R version 3.6.3
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.6.3
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
## Loaded glmnet 4.1-1
```

```r
library(gam)
```

```
## Loading required package: splines
## Loading required package: foreach
## Warning: package 'foreach' was built under R version 3.6.3
##
## Attaching package: 'foreach'
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
## Loaded gam 1.20
```

```r
library(PerformanceAnalytics)
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 3.6.3
## Loading required package: xts
## Warning: package 'xts' was built under R version 3.6.3
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 3.6.3
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Attaching package: 'xts'
## The following objects are masked from 'package:dplyr':
##
##     first, last
##
## Attaching package: 'PerformanceAnalytics'
## The following object is masked from 'package:graphics':
##
##     legend
```

```r
library(mgcv)
```

```
## Warning: package 'mgcv' was built under R version 3.6.3
## Loading required package: nlme
## Warning: package 'nlme' was built under R version 3.6.3
```

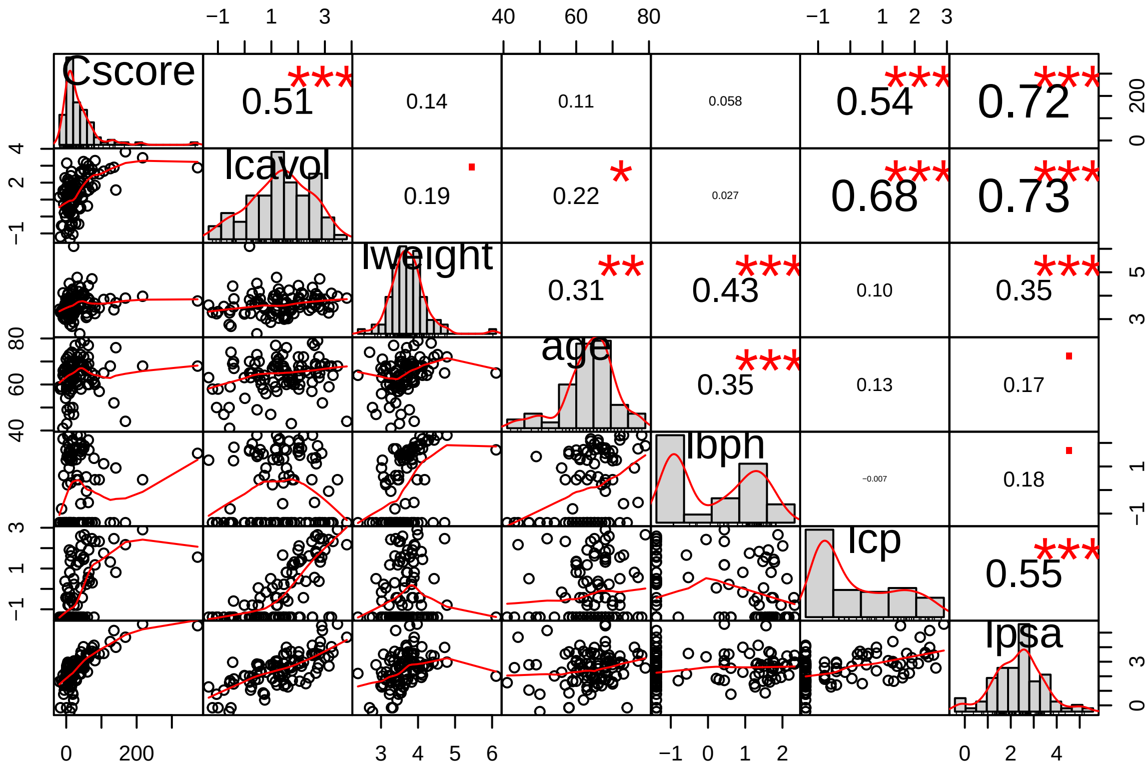
```
##
## Attaching package: 'nlme'
## The following object is masked from 'package:dplyr':
##
##     collapse
## This is mgcv 1.8-35. For overview type 'help("mgcv-package")'.
##
## Attaching package: 'mgcv'
## The following objects are masked from 'package:gam':
##
##     gam, gam.control, gam.fit, s
```

```
load("C:/Users/Nnamdi/Desktop/prostate2.Rdata")
prostate <- na.omit(prostate)
head(prostate)
```

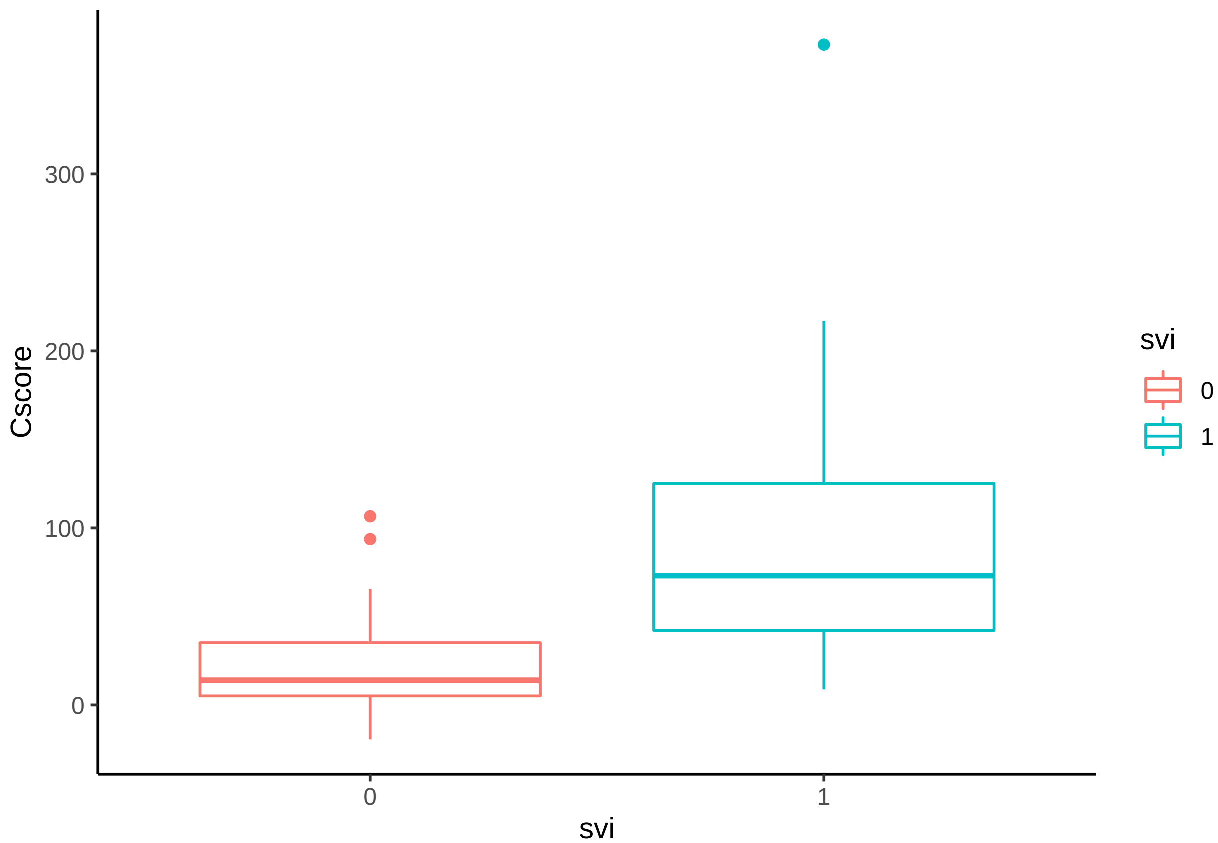
```
##       Cscore     lcavol  lweight age      lbph svi       lcp       lpsa
## 1  10.477386 -0.5798185 2.769459  50 -1.386294   0 -1.386294 -0.4307829
## 2   1.076665 -0.9942523 3.319626  58 -1.386294   0 -1.386294 -0.1625189
## 3  16.101624 -0.5108256 2.691243  74 -1.386294   0 -1.386294 -0.1625189
## 4 -16.393194 -1.2039728 3.282789  58 -1.386294   0 -1.386294 -0.1625189
## 5  21.079178  0.7514161 3.432373  62 -1.386294   0 -1.386294  0.3715636
## 6  18.862940 -1.0498221 3.228826  50 -1.386294   0 -1.386294  0.7654678
```

### Exploratory data analysis

```
prostate$svi <- as.factor(prostate$svi)
chart.Correlation(prostate[,-6], histogram=TRUE, pch=19)
```




```r
ggplot(prostate,aes(svi,Cscore,color = svi))+geom_boxplot() + theme_classic()
```




There is significant positive correlation between the response variable (Cscore) and the following predictor variables: lcavol, lpsa and lcp.

Create training and test datasets

```
set.seed(1000) # allows reproducibilty
index <- sample(1:nrow(prostate),0.8*nrow(prostate)) # use random sample (80%) as training data
train <- prostate[index,] # training dataset
test <- prostate[-index,] # test dataset
```

Preprocess data

```
cols <- c("Cscore","lcavol","lweight","age","lbph","svi","lcp","lpsa")
pre_proc <- preProcess(train[,cols], method =c("center", "scale")) ### scale numeric features
train[,cols] <- predict(pre_proc,train[,cols])
test[,cols] <- predict(pre_proc, test[,cols])
summary(train) # confirm that the mean is zero for the predictor variables
```

```
##      Cscore             lcavol             lweight               age
##  Min.   :-1.0575   Min.   :-2.38837   Min.   :-2.51138   Min.   :-2.94036
##  1st Qu.:-0.5505   1st Qu.:-0.74733   1st Qu.:-0.48839   1st Qu.:-0.45691
##  Median :-0.3066   Median : 0.04851   Median :-0.05013   Median : 0.09497
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.1858   3rd Qu.: 0.62907   3rd Qu.: 0.42861   3rd Qu.: 0.50887
##  Max.   : 6.0111   Max.   : 2.10036   Max.   : 4.78732   Max.   : 1.88857
##       lbph         svi        lcp               lpsa
##  Min.   :-1.0575   0:59   Min.   :-0.9023   Min.   :-2.7301
##  1st Qu.:-1.0575   1:18   1st Qu.:-0.9023   1st Qu.:-0.6882
##  Median : 0.2195          Median :-0.3309   Median : 0.1067
```

```
##  Mean   : 0.0000          Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.9884          3rd Qu.: 0.9564   3rd Qu.: 0.4554
##  Max.   : 1.5409          Max.   : 2.2067   Max.   : 2.7342
```

**Lasso regression model**

```
set.seed(1000)
x_train <- model.matrix(Cscore~.,train)[,-1] #predictor variables
y_train <- train$Cscore # response variables
x_test <- model.matrix(Cscore~.,test)[,-1]
y_test <- test$Cscore
# use cross validation to select the best lambda value with the lowest error
cv_lasso <- cv.glmnet(x_train, y_train, alpha = 1)
cv_lasso$lambda.min
```

```
## [1] 0.01088517
```

```
# Apply the best lambda value in the lasso model
lasso_model <- glmnet(x_train, y_train, alpha = 1, lambda = cv_lasso$lambda.min, family = gaussian(link
coef(lasso_model)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept) -0.0766801
## lcavol      -0.2067151
## lweight     -0.1102918
## age          .
## lbph         .
## svi1         0.3280204
## lcp          0.1569431
## lpsa         0.7303362
```

The lasso regression model is: *Cscore = -0.08 - 0.21(lcavol) - 0.11(lweight) + 0.32(svi[1]) + 0.16(lcp) + 0.73(lpsa) + e*

The lasso regression model shows that Cscore is decreased by 0.21 units for each unit increase in lcavol, while keeping other variables at constant.

**Predictions and performance assessment of lasso regression model**

```
predictions <- predict(lasso_model,x_test)
RMSE <- round(RMSE(predictions, y_test), digits = 2)
MSE <- round(RMSE**2, digits = 2)
print(paste("The lasso model has an RMSE value of",RMSE))
```

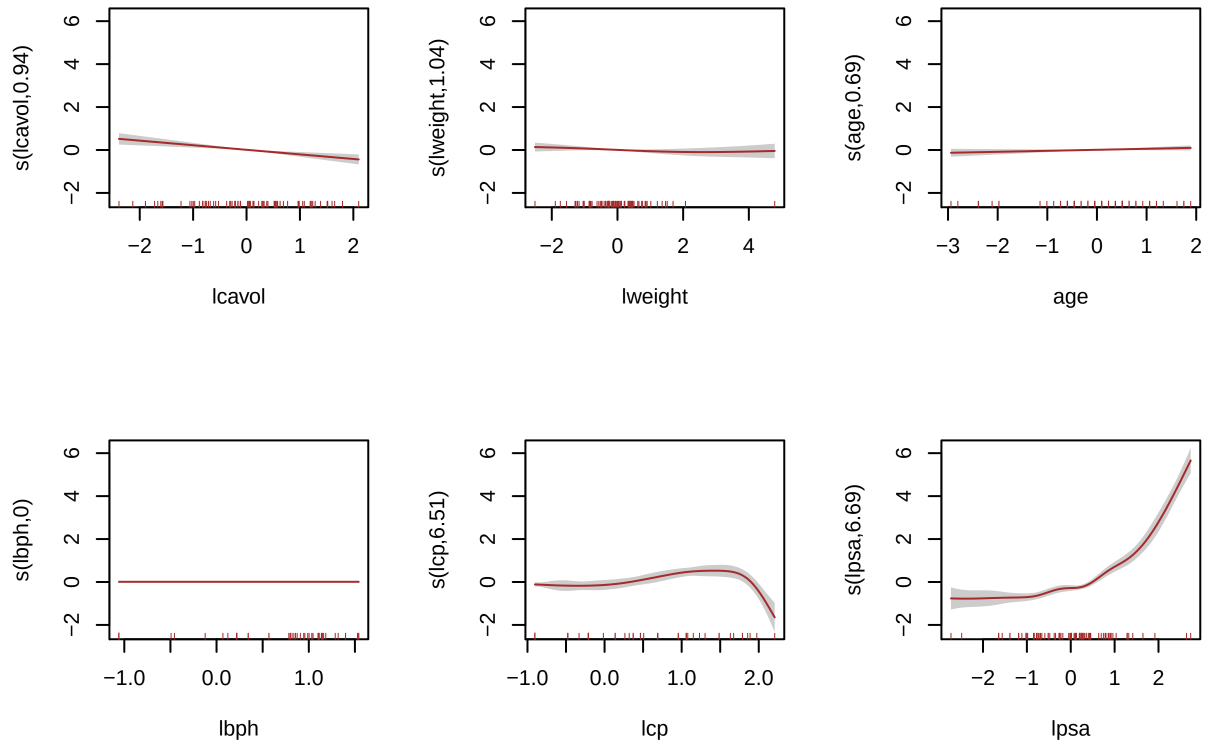
```
## [1] "The lasso model has an RMSE value of 0.47"
```

```
print(paste("The lasso model has an MSE value of",MSE))
```

```
## [1] "The lasso model has an MSE value of 0.22"
```

**Generalized additive model (GAM)**

```
gam_model <- gam(Cscore ~ s(lcavol) + s(lweight) + s(age) + s(lbph) + s(lcp) + s(lpsa) + svi, data = tra
plot(gam_model,pages = 1,rug = TRUE, shade = TRUE,shift = coef(gam_model)[1],  col = "brown")
```




For the smooth terms, if we cannot draw a horizontal line across the 955 confidenxe interval then it is significantly non-linear.

```
summary(gam_model)
```

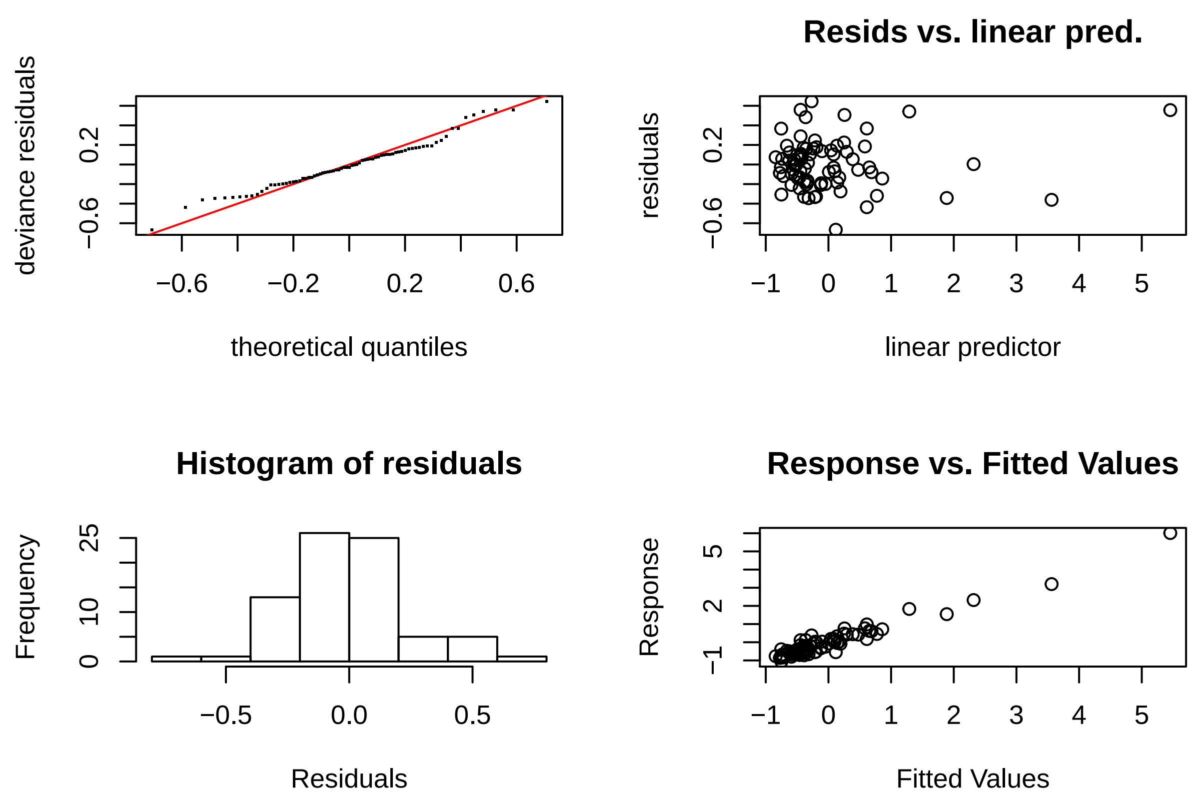
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Cscore ~ s(lcavol) + s(lweight) + s(age) + s(lbph) + s(lcp) +
##     s(lpsa) + svi
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.009101   0.044338   0.205    0.838
## svi1        -0.038932   0.129278  -0.301    0.764
##
## Approximate significance of smooth terms:
##                 edf Ref.df      F  p-value
## s(lcavol)  9.363e-01      9  1.632 0.000168 ***
## s(lweight) 1.041e+00      9  0.269 0.089933 .
## s(age)     6.903e-01      9  0.248 0.064638 .
## s(lbph)    1.480e-06      9  0.000 0.713598
## s(lcp)     6.510e+00      9  7.558  < 2e-16 ***
## s(lpsa)    6.689e+00      9 48.823  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.919   Deviance explained = 93.7%
## -REML = 44.993  Scale est. = 0.081047  n = 77
```

In the GAM model, the significant predictors of Cscore are: lcavol, lcp and lpsa. lpsa and lcp have a non-linear relationship with the response variable while lcavol has a linear relationship with the response variable.

**Gam model diagnostics**

```r
gam.check(gam_model)
```




```
##
## Method: REML   Optimizer: outer newton
## full convergence after 31 iterations.
## Gradient range [-1.670801e-05,2.264971e-05]
## (score 44.99301 & scale 0.08104652).
## Hessian positive definite, eigenvalue range [2.826096e-07,37.95906].
## Model rank =  56 / 56
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                   k'      edf k-index p-value
## s(lcavol)  9.00e+00 9.36e-01    1.23    0.97
## s(lweight) 9.00e+00 1.04e+00    1.10    0.75
## s(age)     9.00e+00 6.90e-01    1.12    0.85
## s(lbph)    9.00e+00 1.48e-06    0.82    0.05 *
```

```
## s(lcp)      9.00e+00 6.51e+00    1.05    0.61
## s(lpsa)     9.00e+00 6.69e+00    1.10    0.78
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

**Predictions and performance of the GAM model**

```
predictions <- predict(gam_model, test, type="response")
RMSE_GAM <- round(RMSE(predictions, test$Cscore),digits = 2)
MSE_GAM <- round(RMSE_GAM**2, digits = 2)
print(paste("The GAM model has an RMSE value of",RMSE_GAM))
```

```
## [1] "The GAM model has an RMSE value of 0.62"
```

```
print(paste("The GAM model has an MSE value of",MSE_GAM))
```

```
## [1] "The GAM model has an MSE value of 0.38"
```

**Comparison of Lasso and GAM models**

```
RMSE_compare <- c(RMSE,RMSE_GAM)
MSE_compare <- c(MSE,MSE_GAM)
accuracy <- data.frame(RMSE_compare,MSE_compare)
row.names(accuracy) <- c("Lasso","GAM")
accuracy
```

```
##       RMSE_compare MSE_compare
## Lasso         0.47        0.22
## GAM           0.62        0.38
```

Lasso regression model had better performance than GAM in the analysis of this dataset. This is probably because Lasso sets the irrelevant predictors to zero.