CS 109B, Spring 2017, Homework 2: Generalized Additive Models

Problem 1: Heart Disease Diagnosis

In this problem, the task is to build a model that can diagnose heart disease for a patient presented with chest pain. The data set is provided in the files `dataset_1_train.txt` and `dataset_1_test.txt`, and contains 6 predictors for each patient, along with the diagnosis from a medical professional.

- By visual inspection, do you find that the predictors are good indicators of heart disease in a patient?
- Apply the generalized additive model (GAM) method to fit a binary classification model to the training set and report its classification accuracy on the test set. You may use a smoothing spline basis function wherever relevant, with the smoothing parameter tuned using cross-validation on the training set. Would you be able to apply the smoothing spline basis to categorical predictors? Is there a difference in the way you would handle categorical attributes in `R` compared to `sklearn` in `Python`?
- Plot the smooth of each predictor for the fitted GAM. By visual inspection, do you find any benefit in modeling the numerical predictors using smoothing splines?
- Using a likelihood ratio test, compare the fitted GAM with the following models: (i) a GAM with only the intercept term; (ii) a GAM with only categorical predictors; and (iii) a GAM with all predictors entered linearly.

Hints: You may use the function `gam` in the `gam` library to fit GAM with binary responses. Do not forget to set the attribute `family = binomial(link="logit")`. The `plot` function can be used to visualize the local models fitted by GAM on each predictor. You may use the `anova` function (with attribute `test="Chi"`) to compare two models using a likelihood ratio test.

You may use the following sample code for cross-validation:

```r
library(boot)

# Function to compute k-fold cross-validation accuracy for a given classification model
cv_accuracy = function(model, data, k) {
  # Input:
  #   'model' - a fitted classification model
  #   'data' - data frame with training data set used to fit the model
  #   'k' - number of folds for CV
  # Output:
  #   'cv_accuracy' - cross-validation accuracy for the model

  acc <- 1 - cv.glm(data, model, K = k)$delta[1]
  return(acc)
}
```

Solution:

```r
#load libraries
library(ggplot2)
library(gridExtra)
library(productplots)
library(gam)
```

Load train and test datasets

```r
# load train set
train = read.csv("datasets/dataset_1_train.txt", header=TRUE)
cat("Train data size:", dim(train), "\n")
head(train)

# load test set
test = read.csv("datasets/dataset_1_test.txt", header=TRUE)
cat("\nTest data size:", dim(test), "\n")
head(test)
cat("\n")

#Dataset structure
str(train)
```

```
## Train data size: 210 7
##   Age Sex    ChestPain RestBP ExAng       Thal HeartDisease
## 1  67   1 asymptomatic    160     1     normal          Yes
## 2  37   1   nonanginal    130     0     normal           No
## 3  59   1   nonanginal    126     0      fixed          Yes
## 4  54   1   nonanginal    150     0 reversable           No
## 5  58   0 asymptomatic    100     0     normal           No
## 6  50   0   nontypical    120     0     normal           No
##
## Test data size: 91 7
##   Age Sex    ChestPain RestBP ExAng       Thal HeartDisease
## 1  63   1      typical    145     0      fixed           No
## 2  67   1 asymptomatic    160     1     normal          Yes
## 3  67   1 asymptomatic    120     1 reversable          Yes
## 4  56   1   nontypical    120     0     normal           No
## 5  56   1   nonanginal    130     1      fixed          Yes
## 6  48   1   nontypical    110     0 reversable          Yes
##
## 'data.frame':    210 obs. of  7 variables:
##  $ Age         : int  67 37 59 54 58 50 52 54 57 57 ...
##  $ Sex         : int  1 1 1 1 0 0 1 0 1 1 ...
##  $ ChestPain   : Factor w/ 4 levels "asymptomatic",..: 1 2 2 2 1 3 4 2 2 1 ...
##  $ RestBP      : int  160 130 126 150 100 120 118 108 150 132 ...
##  $ ExAng       : int  1 0 0 0 0 0 0 0 0 1 ...
##  $ Thal        : Factor w/ 3 levels "fixed","normal",..: 2 2 1 3 2 2 1 2 3 3 ...
##  $ HeartDisease: Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
```

```r
library(boot)

# Function to compute k-fold cross-validation accuracy for a given classification model
cv_accuracy = function(model, data, k) {
  # Input:
  #   'model' - a fitted classification model
  #   'data' - data frame with training data set used to fit the model
  #   'k' - number of folds for CV
  # Output:
  #   'cv_accuracy' - cross-validation accuracy for the model

  acc <- 1 - cv.glm(data, model, K = k)$delta[1]
  return(acc)
}


classification_accuracy = function(true_val,predicted) {
  # Input:
  #   'true_val' - Actual value (truth)
  #   'predicted' - Predicted probabilites by model
  # Output:
  #   classfication accuracy
  y = true_val=='Yes'
  y_ = (predicted>0.5)
  return (mean(y == y_))
}
```

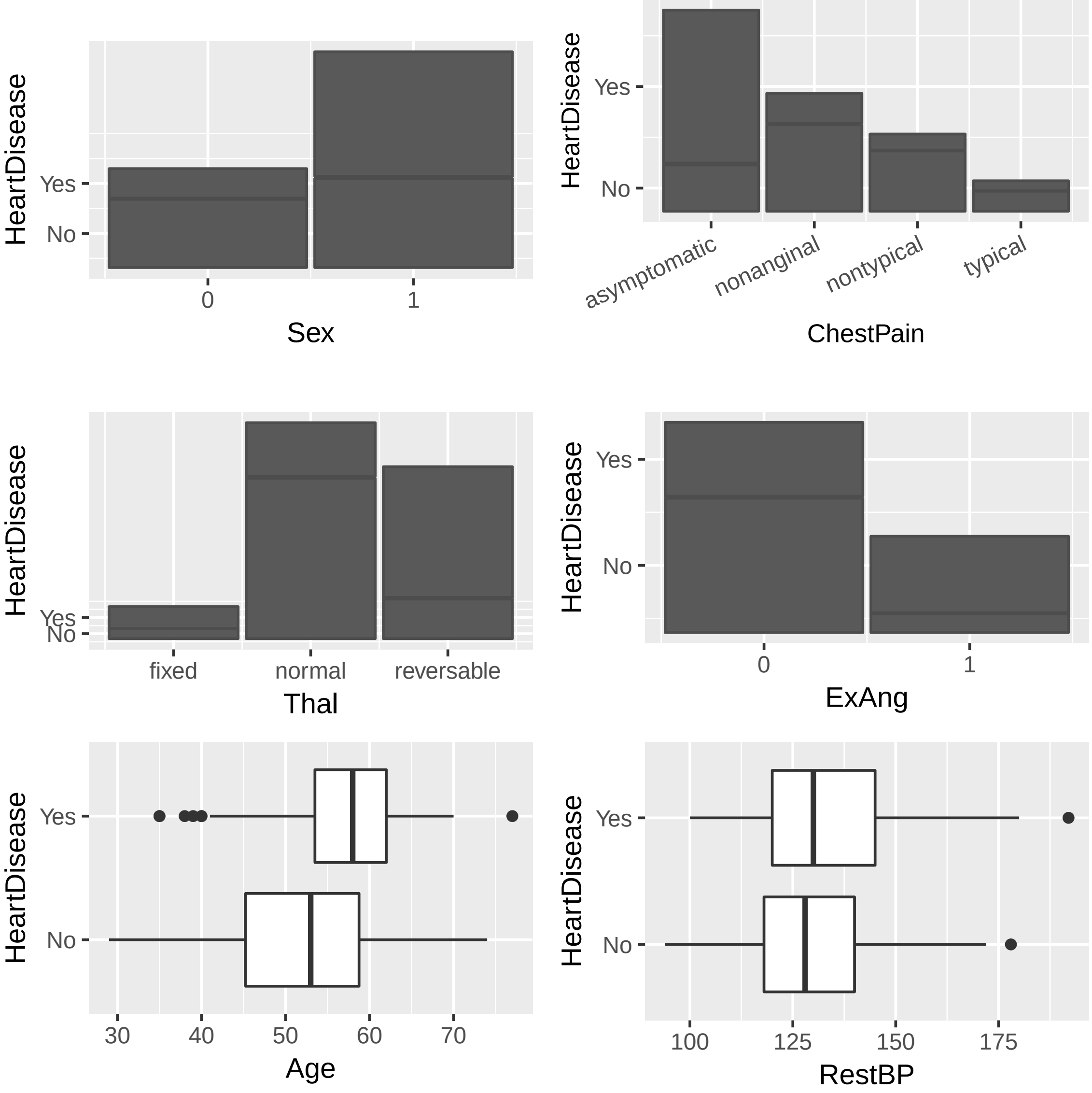
```
table(train$HeartDisease) #Check how many patients with or without HeartDisease
```

```
##
##  No Yes
## 106 104
```

By visual inspection, do you find that the predictors are good indicators of heart disease in a patient?

```
p1 = prodplot(train, ~ HeartDisease + Sex, c("vspine", "hbar")) + ggtitle("")
#Observation: HeartDisease is highest when sex = 1 as compared to sex=0

p2 = prodplot(train, ~ HeartDisease + ChestPain, c("vspine", "hbar")) +
      theme(axis.text.x = element_text(angle = 25, hjust = 1),
            axis.title=element_text(size=10))
#Observation: HeartDisease is highest, when ChestPain = asymptomatic

p3 = prodplot(train, ~ HeartDisease + Thal, c("vspine", "hbar")) + ggtitle("") +
      theme(plot.title = element_text(hjust = 0.5))
#Observation: HeartDisease is not common when Thal=normal

p4 = prodplot(train, ~ HeartDisease + ExAng, c("vspine", "hbar")) + ggtitle("") +
      theme(plot.title = element_text(hjust = 0.5))
#Observation: Patients with ExAng=1, HeartDisease is higher.

p5 = ggplot(train, aes(x = HeartDisease, y = Age)) +
        geom_boxplot() + coord_flip()
#Observation: The median age is higher for patients with HeartDisease

p6 = ggplot(train, aes(x = HeartDisease, y = RestBP)) +
        geom_boxplot() + coord_flip()
```

```r
#Observation: RestBP is slightly higher for patients with HeartDisease.

grid.arrange(p1,p2,p3,p4,p5,p6,nrow=3,ncol=2)
```



Observation: Starting from top-left:
(1) HeartDisease vs Sex - HeartDisease is highest when sex = 1 as compared to sex=0.
(2) HeartDisease vs ChestPain - HeartDisease is highest, when ChestPain = asymptomatic.
(3) HeartDisease vs Thal - HeartDisease is not common when Thal=normal.
(4) HeartDisease vs ExAng - HeartDisease is higher for Patients with ExAng=1.
(5) HeartDisease vs Age - The median age is higher for patients with HeartDisease.
(6) HeartDisease vs RestBP - RestBP is also slightly higher for patients with HeartDisease.

Overall, the predictors seem to be good indicators of predicting heart disease in patients.

Apply the generalized additive model (GAM) method to fit a binary classification model to the training set and report its classification accuracy on the test set. You may use a smoothing spline basis function wherever relevant, with the smoothing parameter tuned using cross-validation on the training set. Would you be able to apply the smoothing spline basis to

categorical predictors? Is there a difference in the way you would handle categorical attributes in `R` compared to `sklearn` in `Python`?

Function to create GAM model

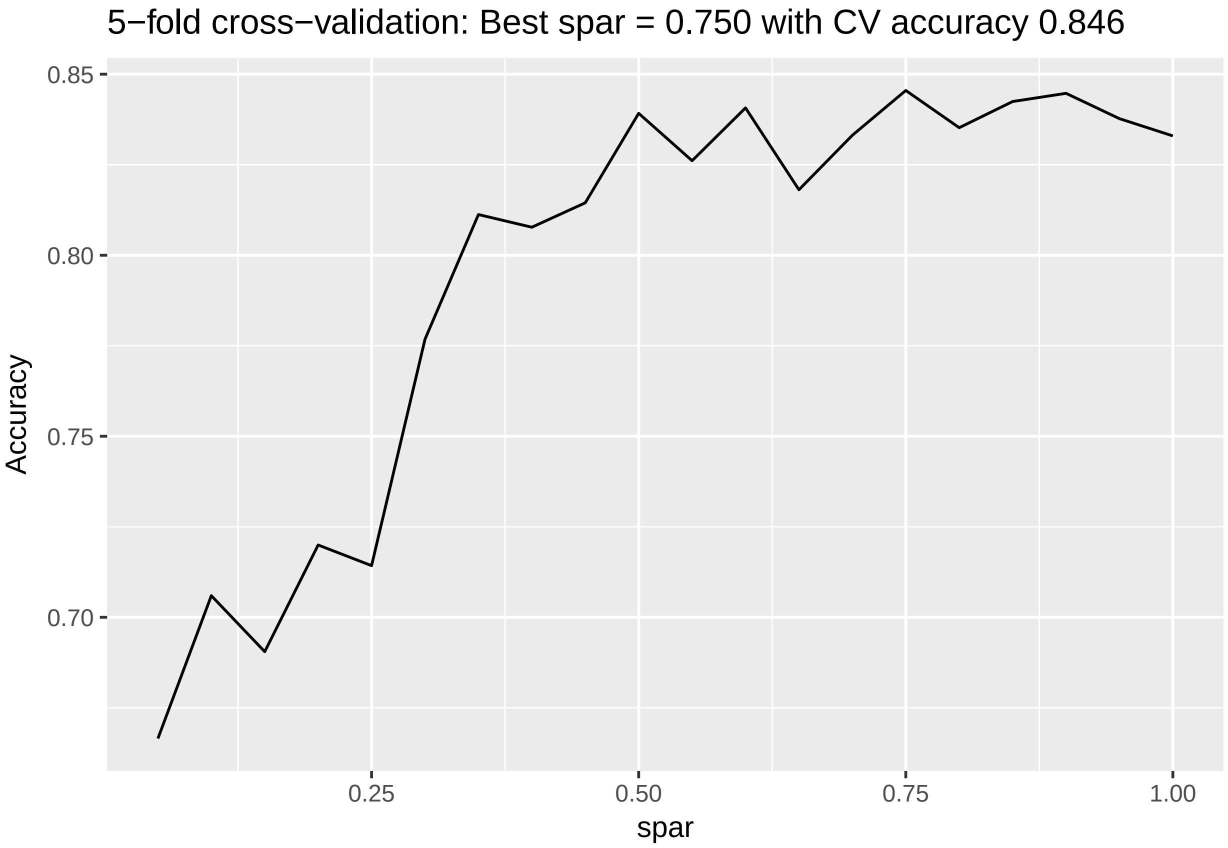
```
fit_gam_s = function(train, test, spar_val, disp) {
  # Input:
  #   Training dataframe: 'train',
  #   Test dataframe: 'test',
  #   Tuning parameter spar: 'spar_val'
  #   Boolean value to decide what will be return value: 'disp'
  # Output:
  #   if 'disp' is true function returns GAM model else function returns GAM test accuracy

  gam_formula = as.formula(paste0("HeartDisease ~ s(RestBP,spar = ",spar_val,") +
                                  s(Age,spar = ",spar_val,") + ChestPain + factor(Sex) + Thal +
                                  factor(ExAng)"))

  model.gam <- gam(gam_formula, data=train,family=binomial(link = "logit"))

  preds = predict(model.gam, newdata=test, type="response")
  gam_testaccuracy = classification_accuracy(test$HeartDisease,preds)

  preds = predict(model.gam, newdata=train, type="response")
  gam_trainaccuracy = classification_accuracy(train$HeartDisease,preds)

  if(disp==TRUE){
    cat(sprintf("GAM with smoothing spline (spar = %.2f): Train R^2: %.3f,
                Test R^2: %.3f\n", spar_val, gam_trainaccuracy, gam_testaccuracy))
    return(model.gam)
  }
  else{
    return(gam_testaccuracy)
  }
}
```

```
#Let's explore few spar values, to check how it affects the classification accuracy
acc1 = fit_gam_s(train,test,0.25,FALSE)
acc2 = fit_gam_s(train,test,0.5,FALSE)
acc3 = fit_gam_s(train,test,0.75,FALSE)
acc4 = fit_gam_s(train,test,0.95,FALSE)

cat("Classification accuracy, spar = 0.25:", acc1)
cat("\nClassification accuracy, spar = 0.5:", acc2)
cat("\nClassification accuracy, spar = 0.75:", acc3)
cat("\nClassification accuracy, spar = 0.95:", acc4)
```

```
## Classification accuracy, spar = 0.25: 0.8461538
## Classification accuracy, spar = 0.5: 0.8241758
## Classification accuracy, spar = 0.75: 0.8021978
## Classification accuracy, spar = 0.95: 0.8131868
```

Cross validation for tuning spar parameter

```
spars = seq(0.05, 1, 0.05)
res = rep(NA, length(spars))
set.seed(109)

for (i in 1:length(spars)) {
  gam_formula = as.formula(paste0("HeartDisease ~ s(RestBP,spar = ",spars[i],") +
                                  s(Age,spar = ",spars[i],") + ChestPain + factor(Sex) + Thal +
                                  factor(ExAng)"))
  model.gam <- gam(gam_formula, data=train,family=binomial(link = "logit"))

  res[i] = cv_accuracy(model.gam,train,5)  #5 fold cross-validation
}

# Find spar with highest CV accuracy
best_spar = which(res==max(res))
title_str = sprintf("5-fold cross-validation: Best spar = %.3f with CV accuracy %.3f",
                    spars[best_spar], res[best_spar])

# Plot - Classification accuracy as a function of Spar values
ggplot() +
  geom_line(aes(x=spars,y=res)) +
  labs(x="spar" , y = "Accuracy" ,title=title_str )
```

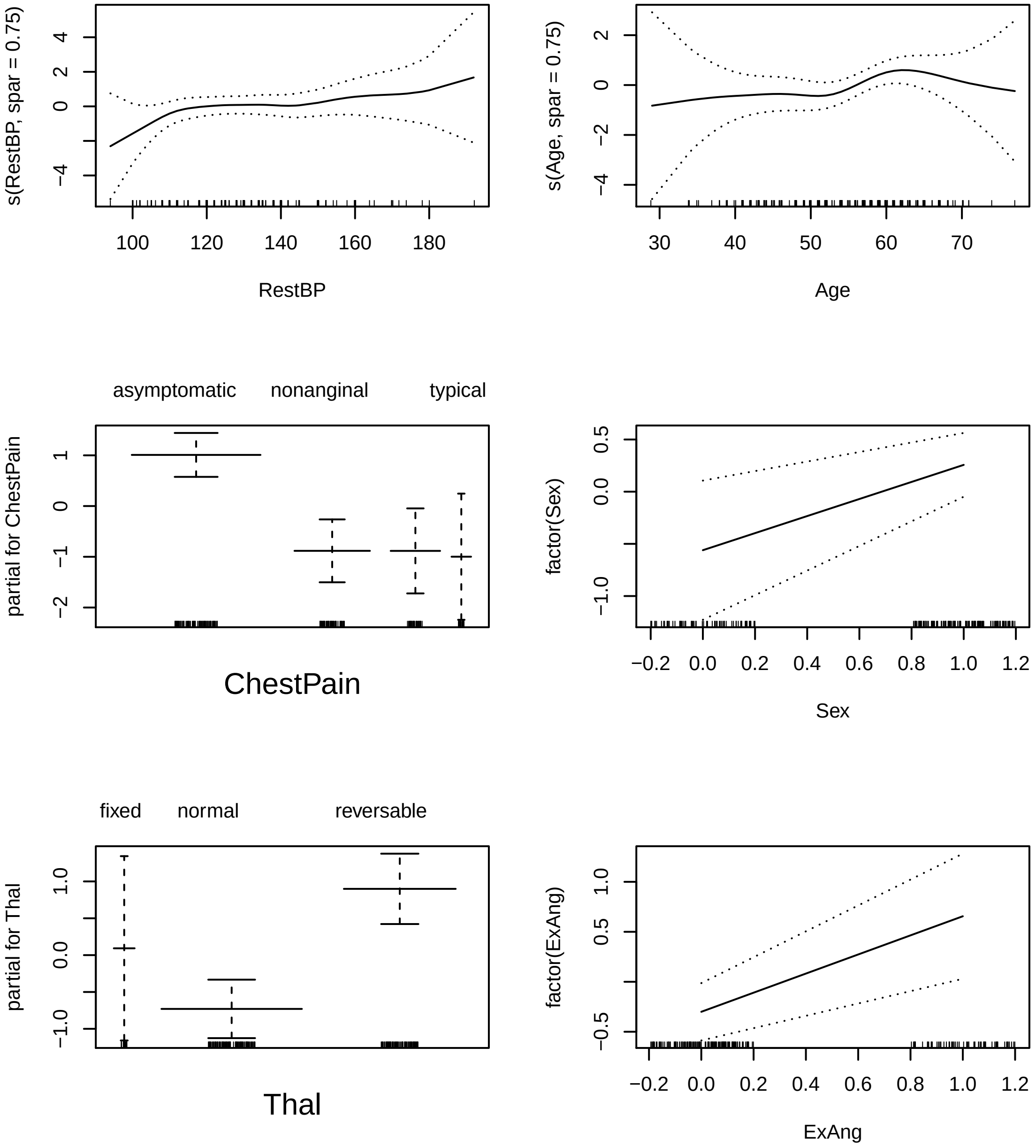


Observation: Smoothing spline basis cannot be applied to categorical predictors.
In `sklearn` in `Python` we have to convert categorical data to numeric to be able to create any models.

Plot the smooth of each predictor for the fitted GAM. By visual inspection, do you find any benefit in modeling the numerical predictors using smoothing splines?

Fit GAM model with best spar value and plot

```
model.gam = fit_gam_s(train,test,spars[best_spar],TRUE)
par(mfrow=c(3,2))
plot(model.gam, se = TRUE)
```



```
par(mfrow=c(1,1))
```

```
## GAM with smoothing spline (spar = 0.75): Train R^2: 0.829,
##                   Test R^2: 0.802
```

Observation: Based on the plots we conclude that smoothing splines are beneficial for numerical predictors (RestBP and Age).

Using a likelihood ratio test, compare the fitted GAM with the following models: (i) a GAM with only the intercept term; (ii) a GAM with only categorical predictors; and (iii) a GAM with all predictors entered linearly.

(i) a GAM with only the intercept term

```
#gam with intercept term
gam_formula = as.formula(paste0("HeartDisease ~ 1"))
model.gam1 <- gam(gam_formula, data=train,family=binomial(link = "logit"))

preds = predict(model.gam1, newdata=test, type="response")
gam_testaccuracy1 = classification_accuracy(test$HeartDisease,preds)
```

(ii) GAM with only categorical predictors

```
gam_formula = as.formula(paste0("HeartDisease ~ Sex + ChestPain + Thal + ExAng"))
model.gam2 <- gam(gam_formula, data=train,family=binomial(link = "logit"))

preds = predict(model.gam2, newdata=test, type="response")
gam_testaccuracy2 = classification_accuracy(test$HeartDisease,preds)
```

(iii) GAM with all predictors entered linearly.

```
gam_formula = as.formula(paste0("HeartDisease ~ Sex + ChestPain + Thal + ExAng + Age +
                                RestBP"))
model.gam3 <- gam(gam_formula, data=train,family=binomial(link = "logit"))

preds = predict(model.gam3, newdata=test, type="response")
gam_testaccuracy3 = classification_accuracy(test$HeartDisease,preds)

cat("Summary of models:")
cat("\nGAM model with intercept only:",gam_testaccuracy1)
cat("\nGAM model with only categorical predictors:", gam_testaccuracy2)
cat("\nGAM model with all predictors entered linearly:", gam_testaccuracy3)
```

```
## Summary of models:
## GAM model with intercept only: 0.6043956
## GAM model with only categorical predictors: 0.8461538
## GAM model with all predictors entered linearly: 0.8131868
```

Likelihood test to compare against previous GAM model

```
anova(model.gam1, model.gam, test="Chi") #Comparison with only intercept term
```

```
## Analysis of Deviance Table
##
## Model 1: HeartDisease ~ 1
## Model 2: HeartDisease ~ s(RestBP, spar = 0.75) + s(Age, spar = 0.75) +
##     ChestPain + factor(Sex) + Thal + factor(ExAng)
##   Resid. Df Resid. Dev     Df Deviance  Pr(>Chi)
## 1    209.00     291.10
## 2    193.83     173.27 15.167   117.83 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model.gam2, model.gam, test="Chi") #Comparison with only categorical predictors
```

```
## Analysis of Deviance Table
##
## Model 1: HeartDisease ~ Sex + ChestPain + Thal + ExAng
## Model 2: HeartDisease ~ s(RestBP, spar = 0.75) + s(Age, spar = 0.75) +
##     ChestPain + factor(Sex) + Thal + factor(ExAng)
##   Resid. Df Resid. Dev     Df Deviance Pr(>Chi)
## 1    202.00     189.75
## 2    193.83     173.27 8.1673   16.486  0.03903 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model.gam3, model.gam, test="Chi") #Comparison with all predictors entered linearly
```

```
## Analysis of Deviance Table
##
## Model 1: HeartDisease ~ Sex + ChestPain + Thal + ExAng + Age + RestBP
## Model 2: HeartDisease ~ s(RestBP, spar = 0.75) + s(Age, spar = 0.75) +
##     ChestPain + factor(Sex) + Thal + factor(ExAng)
##   Resid. Df Resid. Dev     Df Deviance Pr(>Chi)
## 1    200.00     181.62
## 2    193.83     173.27 6.1673   8.3513   0.2276
```

Observation: - Model fitted with only intercept is worse at a significance level of 0.001
- Model fitted with only categorical attributes is also worse at a significance level of 0.05
- The difference in performance between the two models (predictors entered linearly and model.gam) is not statistically significant.