HW7 - STAT 4510/7510

Yang, Anton – #14405729

Instructions: Please list your name and student number clearly. In order to receive credit for a problem, your solution must show sufficient detail so that the grader can determine how you obtained your answer.

Submit a single pdf generated using R Markdown. All R code should be included, as well as all output produced. Upload your work to the Canvas course site.

Problem 1

In this assignment we will consider the `Boston` data set from the `MASS` library. You can use the built-in help to view descriptions of the different variables. We will be considering the regression problem of predicting median home value `medv`.

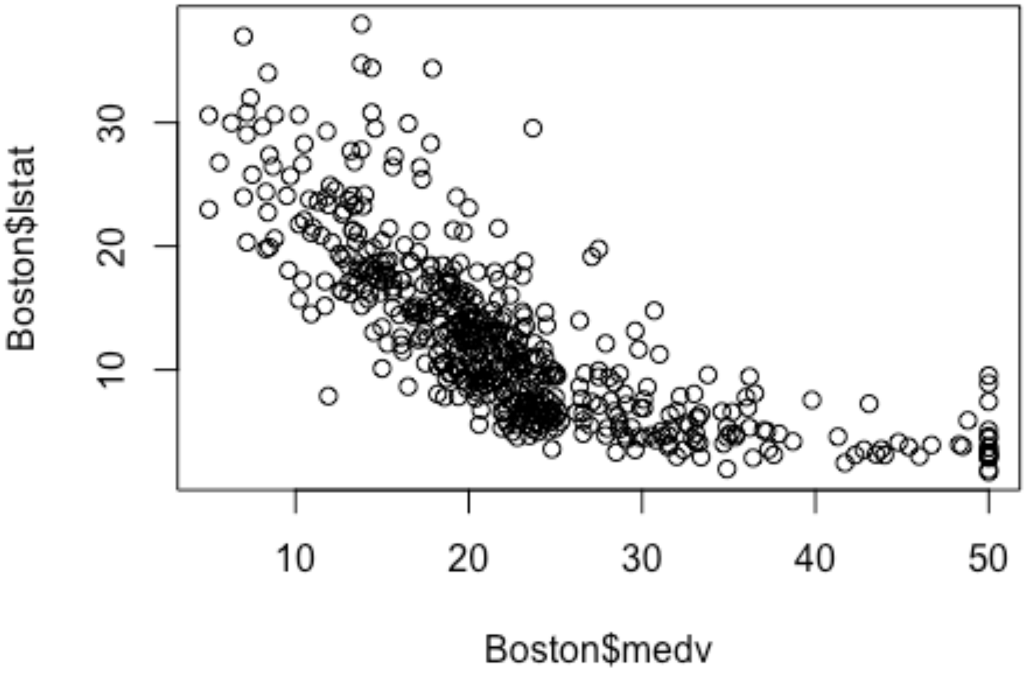
a) Load in the data and produce a scatterplot showing `medv` vs `lstat`. Does the relationship between these variables appear to be linear or nonlinear?

```
library(MASS)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

data<-Boston
plot(Boston$medv, Boston$lstat)
```



According to the scatterplot, the relationship between medv and lstat appear to be nonlinear shaped.

b) Use 5-fold cross-validation to fit polynomial models (with `medv` as a function of `lstat`) of degree 1 through 10. What degree polynomial provides the best fit?

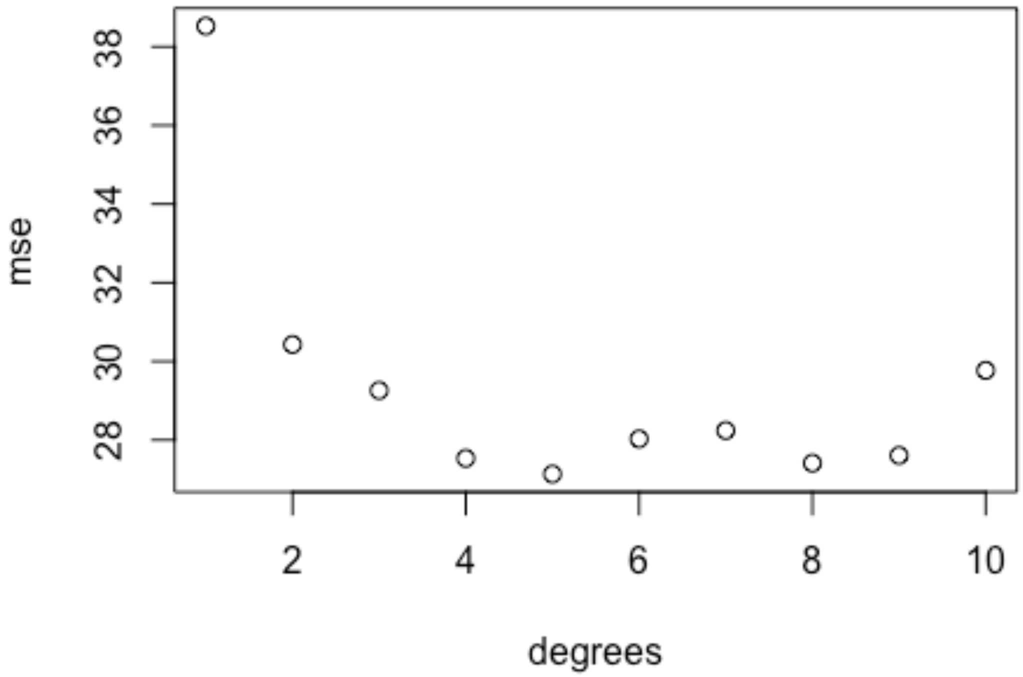
```
set.seed(1)
train_control <- trainControl(method = "cv", number = 5)

degrees <- 1:10

mse <- numeric(length(degrees))

for (degree in degrees) {
  current_formula <- as.formula(paste("medv ~ poly(lstat,", degree, ")", sep
= ""))
  model <- train(current_formula, data = Boston, method = "lm", trControl =
train_control)
  mse[degree] <- mean(model$results$RMSE^2)
}

plot(degrees, mse)
```



```
best_index<-which.min(mse)
best_degree<-degrees[best_index]

print(paste("Best MSE value:", mse[best_index]))

## [1] "Best MSE value: 27.1335642257893"

print(paste("Corresponding polynomial degrees:", best_degree))

## [1] "Corresponding polynomial degrees: 5"
```

According to the result, the best degree is the degree of 5 which has a MSE of 27.1335642257893 This means that the degree of 5 is the best fit of the data without overfitting.

c) Use the `bs()` function to fit a model (with `medv` as a function of `lstat`) using 5-fold cross-validation to select the best value of `df` (use a range of 1 through 20). What value provides the best fit (based on test MSE)?

```
library(splines)
set.seed(1)
df_values<-1:20

train_control<-trainControl(method="cv", number = 5)
```

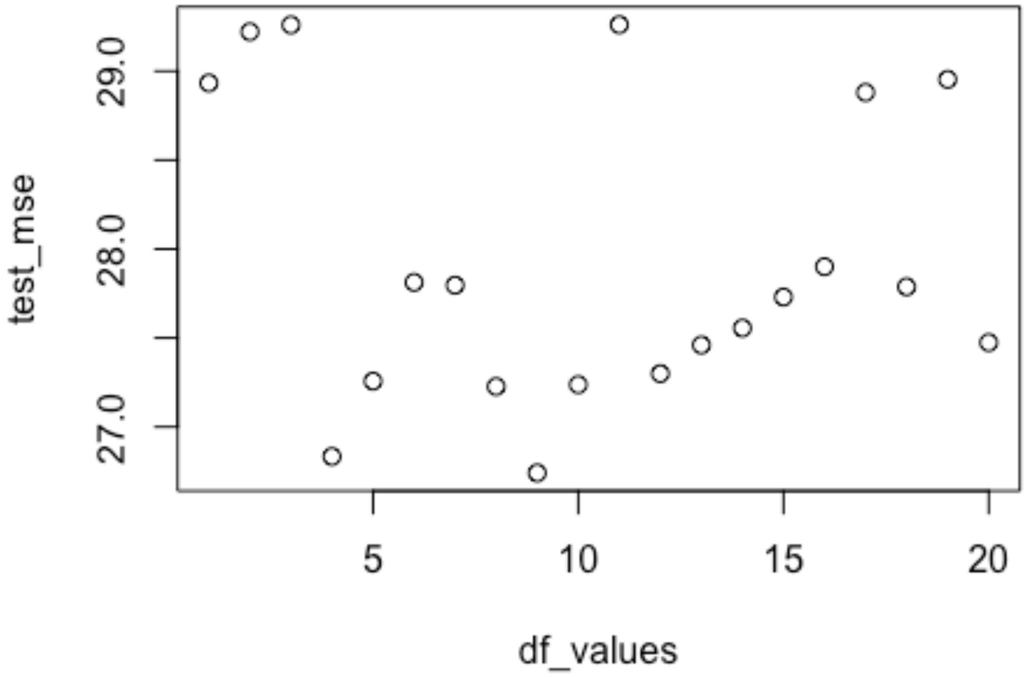
```
test_mse<-numeric(length(df_values))

for (i in seq_along(df_values)) {
  formula <- as.formula(paste("medv ~ bs(lstat, df=", df_values[i], ")", sep
= ""))
  model <- train(formula, data = data, method = "lm", trControl =
train_control)
  test_mse[i] <- mean(model$results$RMSE^2)
}

## Warning in bs(lstat, df = 1): 'df' was too small; have used 3

## Warning in bs(lstat, df = 2): 'df' was too small; have used 3

plot(df_values,test_mse)
```



```
best_index<-which.min(test_mse)
best_df<-df_values[best_index]

print(paste("Best MSE value:", test_mse[best_index]))

## [1] "Best MSE value: 26.7411982656196"
```

```
print(paste("Corresponding degree of freedom (df):", best_df))

## [1] "Corresponding degree of freedom (df): 9"
```

We can see that the best df value is 9 with a MSE of 26.7411982656196. We can see that this model is better than the best polynomial degree model.

d) Repeat part c), using smoothing splines. (Recall that `smooth.splines()` has a built-in cv option that you can use.)

```
set.seed(1)
model<-smooth.spline(data$lstat, data$medv, cv=TRUE)

## Warning in smooth.spline(data$lstat, data$medv, cv = TRUE): cross-
validation
## with non-unique 'x' values seems doubtful

mse<-model$cv.crit


print(paste("Best MSE value:", model$cv.crit))

## [1] "Best MSE value: 27.3821343446285"

print(paste("Corresponding degree of freedom (df):", model$df))

## [1] "Corresponding degree of freedom (df): 11.3741976699527"
```

We can see that by the smooth spline, the best df value is 11.3742 and MSE is 27.3821343446285. This is slightly worse than basis spline and best polynomial order model.

e) Split the data into 80%/20% training and test sets. Fit a GAM on the training data using `medv` as the response and `lstat`, `rm`, and `ptratio` as predictors. The exact model is up to you, just explain in your comments what model you are using. (You may wish to use a combination of splines and linear variables in your model.) Use `summary()` to provide a summary of the model fit. Find the test MSE for this model, and comment on your results.

```
library(gam)

## Loading required package: foreach

## Loaded gam 1.22-3

set.seed(1)
split<-sample(1:nrow(data), size = 0.8*nrow(data))
training_set<-data[split,]
test_set<-data[-split,]


model<-gam(medv~s(lstat)+s(rm)+ptratio, data=training_set)
summary(model)
```

```
##
## Call: gam(formula = medv ~ s(lstat) + s(rm) + ptratio, data =
training_set)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2277  -2.2292  -0.3807   1.8411  28.2950
##
## (Dispersion Parameter for gaussian family taken to be 17.4572)
##
##     Null Deviance: 36363.33 on 403 degrees of freedom
## Residual Deviance: 6878.153 on 393.9998 degrees of freedom
## AIC: 2313.718
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##            Df  Sum Sq Mean Sq  F value    Pr(>F)
## s(lstat)    1 21716.3 21716.3 1243.970 < 2.2e-16 ***
## s(rm)       1  2415.7  2415.7  138.376 < 2.2e-16 ***
## ptratio     1   268.3   268.3   15.367 0.0001044 ***
## Residuals 394  6878.2    17.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##             Npar Df Npar F     Pr(F)
## (Intercept)
## s(lstat)          3 15.010 2.863e-09 ***
## s(rm)             3 57.208 < 2.2e-16 ***
## ptratio
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
predictions<-predict(model,newdata=test_set)
test_mse<-mean((test_set$medv - predictions)^2)
print(test_mse)
```

```
## [1] 22.29639
```

For this model, I applied smooth function for the predictor variable lstat and rm. I let ptratio to be included as a linear term. This model includes both linear and non-linear components. This allows the capture potential non-linearities in the relationships between the predictors and the response variable while also allowing for linear effects where appropriate. According to the summary, s(lstat) has a F-statistics of 15.010 and p-value of near 0, which means it is highly significant. We can also see that s(rm) is also highly significant. This means that both of these variables are relevant on predicting response variable. We can see that the MSE is 22.29639, which means that the predictions are on average 22.29639 squared distance away from actual value.