Business Analytics - ETC3250 2017 - Lab 4 solutions

Souhaib Ben Taieb

16 August 2017

Assignment - Question 1

Using the bias-variance decomposition, show that $E[(y - \hat{f}(x_0))^2]$ is minimum when $\hat{f}(x_0) = E[y|x = x_0]$. What is this minimum value?

Replacing $\hat{f}(x_0)$ by $E[y|x = x_0]$ in the bias-variance decomposition gives $E[(y - \hat{f}(x_0))^2] = \sigma^2$ which is the irreducible error (and the minimum value).

Assignment - Question 2

Write code that includes:

1. fitting your final model above;

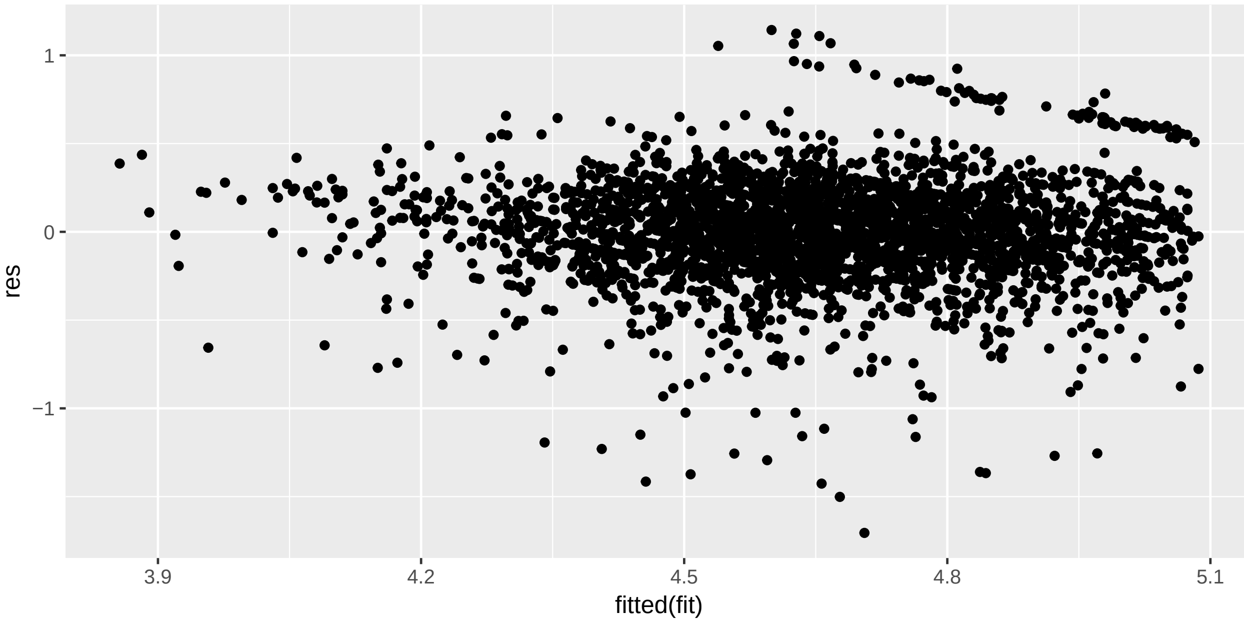
```
library(ISLR)
library(splines)
library(ggplot2)
library(gridExtra)

set.seed(1986)
idx <- sample(1:nrow(Wage), size=2000)
train <- Wage[idx,]
test <- Wage[-idx,]

testMSE <- numeric(20)
for (i in 1:20) {
  fit <- lm(log(wage) ~ year + ns(age, df = i) + education +
              race + jobclass + health + maritl, data=train)
  testMSE[i] <- mean((test$wage - exp(predict(fit, newdata = test)))^2)
}
best_df <- which.min(testMSE)
fit <- lm(log(wage) ~ year + ns(age, df = best_df) + education +
              race + jobclass + health + maritl, data=Wage)
```

2. summary statistics for the residuals;

```
res <- residuals(fit)
summary(res)
#     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
# -1.70600 -0.15170  0.01261  0.00000  0.16570  1.14300
```

3. a plot of the residuals against the fitted values.

```r
qplot(fitted(fit), res)
```



Assignment - Question 3

Do the exercise 7 in Section 2.4 of ISLR.

(a) 3, 2, 3.162, 2.236, 1.414, 1.732

(b) Y = GREEN

(c) Y = RED

(d) Small since we will need more flexibility.

Assignment - Question 4

We want to predict whether a given car gets high or low gas mileage based on a set of features describing the car. We will use the *Auto* dataset in library ISLR.

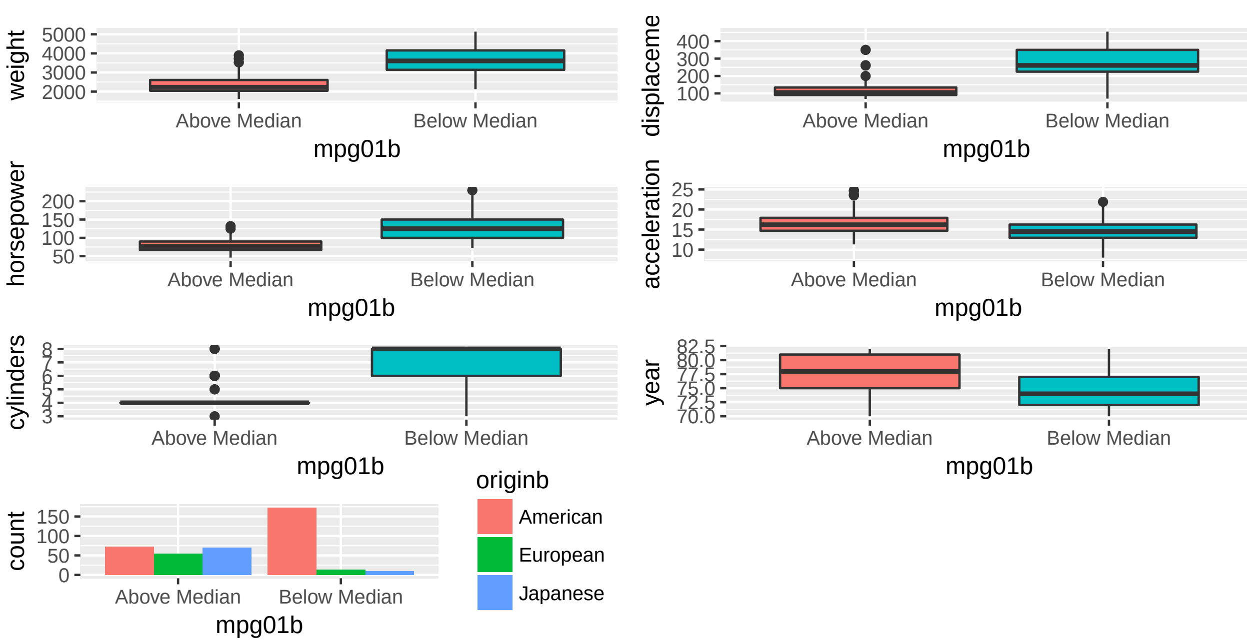
1. Create a binary variable, *mpg01*, that contains a 1 if *mpg* contains a value above its median, and a 0 if *mpg* contains a value below its median. Add the variable *mpg01* to the data.frame *Auto*

```r
DT <- Auto
median_mpg <- median(DT$mpg)
DT$mpg01 <- ifelse(DT$mpg > median_mpg, 1, 0)
```

```
DT$mpg01b <- ifelse(DT$mpg01, "Above Median", "Below Median")
DT$originb[Auto$origin == 1] <- "American"
DT$originb[Auto$origin == 2] <- "European"
DT$originb[Auto$origin == 3] <- "Japanese"

DT$originb <- factor(DT$originb)
DT$mpg01b <- factor(DT$mpg01b)
```

2. Explore the data graphically in order to investigate the association between *mpg01* and the other features. Which of the other features seem most likely to be useful in predicting *mpg01*? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

```
variables <- c("weight", "displacement", "horsepower", "acceleration", "cylinders", "year", "originb")
plist <- vector("list", length(variables))
for(i in seq_along(variables)){
  plist[[i]] <- ggplot(DT, aes_string(x = "mpg01b", y = variables[i], fill = "mpg01b")) +
                geom_boxplot() + theme(legend.position = "none")
  if(variables[i] == "originb"){
    plist[[i]] <- ggplot(DT, aes(mpg01b)) + geom_bar(aes(fill = originb), position = "dodge")
  }
}


library(gridExtra)
n <- length(plist)
nCol <- floor(sqrt(n))
do.call("grid.arrange", c(plist, ncol=nCol))
```



3. Split the data into a training set and a test set.
4. Perform KNN on the training data, with several values of K, in order to predict *mpg01*. Use only the variables that seemed most associated with *mpg01*. Plot the training and testing errors a function of $1/k$. Compare which value of K seems to perform the best when using training or testing errors?

```r
library(class)
set.seed(1986)
idtrain <- sample(1:nrow(DT), size=300)
variables <- c("weight", "displacement")  # OK to use other variables. This is just an example.
Xtrain <- DT[idtrain, variables]
Xtest <- DT[-idtrain, variables]
Ytrain <- DT[idtrain, "mpg01"]
Ytest <- DT[-idtrain, "mpg01"]

error_rate_train <- error_rate_test <- numeric(30)
K <- 30
for(k in seq(K)){
  knn.pred <- knn(train = Xtrain, test = Xtrain, cl = Ytrain, k = k)
  error_rate_train[k] <- sum(knn.pred != Ytrain)

  knn.pred <- knn(train = Xtrain, test = Xtest, cl = Ytrain, k = k)
  error_rate_test[k] <- sum(knn.pred != Ytest)
}

E <- data.frame(kinv = 1/seq(K), error_rate_train, error_rate_test)

ggplot(data = E) +
  geom_line(aes(kinv, error_rate_train), col = "red") +
  geom_line(aes(kinv, error_rate_test), col = "blue") + ylab('Error rate') + xlab('1/K')
```