EXAM 1 Q2

Trey Davidson

2024-10-12

### Part A

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

data <- read.table("~/Desktop/HW/EXAM1/T7\_1\_SEISHU.DAT", header = FALSE)  
colnames(data)[colnames(data) == "V1"] <- "taste"  
  
# Sets x and y's for model  
x <- data[, 2:10]   
y <- data$taste  
  
# Creates df from y and x data  
model\_data <- data.frame(taste = y, x)  
  
# Uses train\_control to eaisly run Leave one out cross validation   
train\_control <- trainControl(method = "LOOCV")  
  
  
model <- train(taste ~ ., data = model\_data, method = "lm", trControl = train\_control)  
  
# PUlls RMSE from results  
model$results$RMSE

## [1] 0.3727178

### B.

library(leaps)  
  
data <- read.table("~/Desktop/HW/EXAM1/T7\_1\_SEISHU.DAT", header = FALSE)  
colnames(data)[1] <- "taste"  
  
x <- data[, 2:10]   
y <- data$taste  
  
model\_data <- data.frame(taste = y, x)  
  
best\_subset <- regsubsets(taste ~ ., data = model\_data, nvmax = 9)  
  
summary\_best <- summary(best\_subset)  
  
cp\_values <- summary\_best$cp  
  
min\_cp\_index <- which.min(cp\_values)  
best\_cp\_value <- cp\_values[min\_cp\_index]  
best\_model\_size <- min\_cp\_index  
  
cat("Best model size (p):", best\_model\_size, "\n")

## Best model size (p): 4

selected\_vars <- names(coef(best\_subset, best\_model\_size))[-1]  
  
# This creates a formula that then can be plugged into lm model  
lm\_form <- as.formula(paste("taste ~", paste(selected\_vars, collapse = " + ")))  
  
linnear\_model <- lm(lm\_form, data = data)  
  
summary(linnear\_model)

##   
## Call:  
## lm(formula = lm\_form, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.63019 -0.16020 -0.01038 0.25439 0.44255   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -7.92871 2.53089 -3.133 0.004379 \*\*   
## V2 0.54515 0.12910 4.223 0.000279 \*\*\*  
## V3 1.50919 0.46942 3.215 0.003582 \*\*   
## V6 0.08081 0.03450 2.342 0.027419 \*   
## V7 0.55210 0.22529 2.451 0.021601 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2977 on 25 degrees of freedom  
## Multiple R-squared: 0.5174, Adjusted R-squared: 0.4401   
## F-statistic: 6.7 on 4 and 25 DF, p-value: 0.0008287

R sqaured = 0.5174 Yes all varibales in final model are statstical siginfinct V2 = odor V3 = PH V6 = Sake meter V7 = direct reucing sugar

### C.

library(leaps)  
# Sets up data and defines V1 as TASTE  
data <- read.table("~/Desktop/HW/EXAM1/T7\_1\_SEISHU.DAT", header = FALSE)  
colnames(data)[1] <- "taste"  
y <- data$taste  
x <- data[, 2:10]  
model\_data <- data.frame(y, x)  
  
  
loocv\_errors <- numeric(6)  
  
# k is the amount of varibles that dataset will hold.   
for (k in 3:9) {  
 # This finds the model with the best number of varibles for any given varible  
 best\_subset <- regsubsets(y ~ ., data = model\_data, nbest = 1, nvmax = k)  
 best\_models <- summary(best\_subset)  
   
 # Initialize a vector to hold the test errors for the best k-variable model  
 test\_errors <- numeric(nrow(model\_data))  
   
 # Perform LOOCV  
 for (i in 1:nrow(model\_data)) {  
 train\_data <- model\_data[-i, ]  
 test\_data <- model\_data[i, , drop = FALSE]  
   
   
 selected\_vars <- names(coef(best\_subset, k))  
 model\_formula <- as.formula(paste("y ~", paste(selected\_vars[-1], collapse = "+")))  
   
 fit <- lm(model\_formula, data = train\_data)  
   
 prediction <- predict(fit, newdata = test\_data)  
   
 # Calculate squared error  
 test\_errors[i] <- (test\_data$y - prediction)^2  
 }  
   
 # Calculate LOOCV error for the current model size  
 loocv\_errors[k-3] <- mean(test\_errors)  
}  
  
# This prints the test errors at every varible level from 3-9  
loocv\_errors

## [1] 0.09881545 0.10103897 0.10431705 0.11408484 0.12119578 0.13891859

#### D.

1. Yes they do line up. the second one in the vector has the lowest error, and part b gave out 4 varibles. It does make sense if the model is truely predictive. As Mallow’s CP and LOOCV emppthise difernt things, but a good model should minmize both.