Model Selection Practice using the BodyFat Data

Prof K

11/12/2020

```
library(faraway)
library(tidyverse)
library(dplyr)
library(glmnet)
theme_set(theme_classic())
```

The goal of the exercise is to practice the methods we have been learning.

Data

The dataset fat is available in the library(faraway). You may have to install this library. Before installing read https://hpcsupport.atlassian.net/servicedesk/customer/portal/3/topic/9e49af72-b170-4f7e-ba9c-0a2cfef45cd6/article/562429973

The data set contains several physical measurements of 252 males. Most of the variables can be measured with a scale or tape measure. Can they be used to predict the percentage of body fat? If so, this offers an easy alternative to an underwater weighing technique.

Data frame with 252 observations on the following 19 variables.

The data were supplied by Dr. A. Garth Fisher, Human Performance Research Center, Brigham Young University, Provo, Utah 84602, who gave permission to freely distribute the data and use them for non-commercial purposes.

Variables:

- brozek – Percent body fat using Brozek's equation, 457/Density - 414.2
- siri – Percent body fat using Siri's equation, 495/Density - 450
- density – Density (gm/cm^2)
- age – Age (yrs)
- weight – Weight (lbs)
- height – Height (inches)
- adipos – BMI Adiposity index = Weight/Height^2 (kg/m^2)
- free – Fat Free Weight = (1 - fraction of body fat) * Weight, using Brozek's formula (lbs)
- neck – Neck circumference (cm)
- chest – Chest circumference (cm)
- abdom – Abdomen circumference (cm) "at the umbilicus and level with the iliac crest"
- hip – Hip circumference (cm)
- dthigh – Thigh circumference (cm)
- knee – Knee circumference (cm)
- ankle – Ankle circumference (cm)
- biceps – Extended biceps circumference (cm)
- forearm – Forearm circumference (cm)
- wrist – Wrist circumference (cm) "distal to the styloid processes"

Task 0 - Data Prep and Division of Data into Test and Train

With the fat dataset in the library(faraway), we want to fit a linear model to predict body fat (variable brozek) using the other variables available, except for siri (another way of computing body fat), density (it is used in the brozek and siri formulas) and free (it is computed using brozek formula)

```
library(leaps)   #function to search for the best model
library(faraway) #has the dataset fat
set.seed(1212)
fat = fat %>%
dplyr::select(brozek,age,weight,height,adipos, neck , chest,
                              abdom ,hip, thigh , knee ,
                              ankle ,biceps,
                              forearm , wrist)

train = fat %>%
  sample_frac(0.67)

test = fat %>%
  setdiff(train)
```

Following the examples we did in class carry out the following.

Task 1: OLS

Task 2: Best Subsets

Task 3: Best Subsets with CV

In Task 3, do Best Subset with Cross-Validation (CV) on the training data. Report the number of predictors selected and the predictors. Report the test MSE obtained.

Since there is no predict function for regsubsets we use the function from the class notes:

```
predict.regsubsets = function(object,newdata,id,...){
      form = as.formula(object$call[[2]]) # Extract the formula used when we called regsubsets()
      mat = model.matrix(form,newdata)    # Build the model matrix
      coefi = coef(object,id=id)          # Extract the coefficiants of the ith model
      xvars = names(coefi)                # Pull out the names of the predictors used in the ith model
      mat[,xvars]%*%coefi               # Make predictions using matrix multiplication
}
```

```
k = 14        # number of folds
set.seed(1)   # set the random seed so we all get the same results

# Assign each observation to a single fold
folds = sample(1:k, nrow(train), replace = TRUE)

# Create a matrix to store the results of our upcoming calculations,
#we use 14 because we have 14 possible predictors (10 variables 1 with 3 levels)
cv_errors = matrix(NA, k, 14, dimnames = list(NULL, paste(1:14)))

# For each fold
for(j in 1:k){
  # Fit the model with each subset of predictors on the training part of the fold
  best.fit=regsubsets(brozek~.,data=train[folds!=j,], nvmax=14)
  # For each subset
```

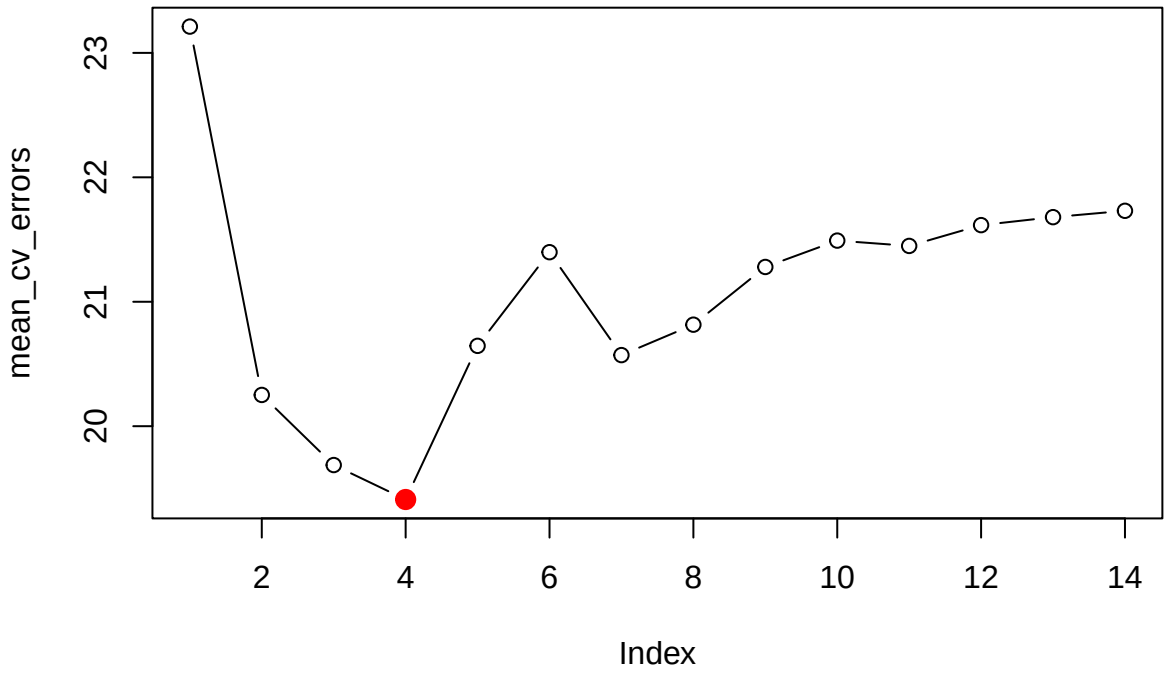
```
  for(i in 1:14){
    # Predict on the hold out part of the fold for that subset
    pred=predict(best.fit, train[folds==j,],id=i)
    # Get the mean squared error for the model trained on the fold with the subset
    cv_errors[j,i]=mean((train$brozek[folds==j]-pred)^2)
  }
}


# Take the mean of over all folds for each model size
mean_cv_errors = apply(cv_errors, 2, mean)
mean_cv_errors
```

```
##        1        2        3        4        5        6        7        8
## 23.21111 20.25116 19.68788 19.41109 20.64572 21.39842 20.57135 20.81581
##        9       10       11       12       13       14
## 21.27949 21.49160 21.44874 21.61583 21.67991 21.73098
```

```
# Find the model size with the smallest cross-validation error
min = which.min(mean_cv_errors)
min
```

```
## 4
## 4
```

```
# Plot the cross-validation error for each model size, highlight the min
plot(mean_cv_errors, type='b')
points(min, mean_cv_errors[min][1], col = "red", cex = 2, pch = 20)
```




```
reg_best = regsubsets(brozek~., data = train, nvmax = 4)
coef(reg_best, 4)
```

```
## (Intercept)      weight       abdom     forearm       wrist
## -28.9720272  -0.1239790   0.9054636   0.4283119  -1.4124922
```

```r
#pred=predict(best.fit, train[folds==j,],id=i)
predCV_best <- predict(reg_best, test, id=4)
MSEcv_best=mean((test$brozek - predCV_best)^2)  #this is test MSE
MSEcv_best
```

```
## [1] 14.37278
```