Homework 6 - Predictive Modeling in Finance and Insurance

Dennis Goldenberg

2024-02-27

```r
library(MASS)
library(ggplot2)
library(leaps)
Boston$chas <- factor(Boston$chas)
```

1. Model Selection

a. Best Subset Selection

I perform the selection as intended:

```r
bestSubset <- leaps::regsubsets(medv ~., data = Boston, method = "exhaustive",
                                nvmax = dim(Boston)[2] - 1)
summary(bestSubset)$outmat
```

```
##           crim zn  indus chas1 nox rm  age dis rad tax ptratio black lstat
## 1  ( 1 )  " "  " " " "   " "   " " " " " " " " " " " " " "     " "   "*"  
## 2  ( 1 )  " "  " " " "   " "   " " "*" " " " " " " " " " "     " "   "*"  
## 3  ( 1 )  " "  " " " "   " "   " " "*" " " " " " " " " "*"     " "   "*"  
## 4  ( 1 )  " "  " " " "   " "   " " "*" " " "*" " " " " "*"     " "   "*"  
## 5  ( 1 )  " "  " " " "   " "   "*" "*" " " "*" " " " " "*"     " "   "*"  
## 6  ( 1 )  " "  " " " "   "*"   "*" "*" " " "*" " " " " "*"     " "   "*"  
## 7  ( 1 )  " "  " " " "   "*"   "*" "*" " " "*" " " " " "*"     "*"   "*"  
## 8  ( 1 )  " "  "*" " "   "*"   "*" "*" " " "*" " " " " "*"     "*"   "*"  
## 9  ( 1 )  "*"  " " " "   "*"   "*" "*" " " "*" "*" " " "*"     "*"   "*"  
## 10  ( 1 ) "*"  "*" " "   " "   "*" "*" " " "*" "*" "*" "*"     "*"   "*"  
## 11  ( 1 ) "*"  "*" " "   "*"   "*" "*" " " "*" "*" "*" "*"     "*"   "*"  
## 12  ( 1 ) "*"  "*" "*"   "*"   "*" "*" " " "*" "*" "*" "*"     "*"   "*"  
## 13  ( 1 ) "*"  "*" "*"   "*"   "*" "*" "*" "*" "*" "*" "*"     "*"   "*"  
```

So , the variables were 1. lstat 2. rm 3. ptratio 4. dis 5. nox 6. chas. I show the c_p, BIC, and R^2:

```r
eStatdf<-data.frame(cbind(1:6, summary(bestSubset)$cp[1:6],
         summary(bestSubset)$bic[1:6],summary(bestSubset)$adjr2[1:6]))
colnames(eStatdf) <- c("Model #", "Cp", "BIC", "Adj. R Squared")
eStatdf
```

```
##   Model #        Cp       BIC Adj. R Squared
## 1       1 362.75295 -385.0521      0.5432418
## 2       2 185.64743 -496.2582      0.6371245
## 3       3 111.64889 -549.4767      0.6767036
## 4       4  91.48526 -561.9884      0.6878351
## 5       5  59.75364 -585.6823      0.7051702
## 6       6  47.17537 -592.9553      0.7123567
```

b. Forward and backward selection

I repeat the procedure for a, but doing forward and backward selection, and show the first 6 variables selected in each case in data frame format:

```
forSubset <- leaps::regsubsets(medv ~., data = Boston, method = "forward",
                               nvmax = dim(Boston)[2] - 1)
backSubset <- leaps::regsubsets(medv ~., data = Boston, method = "backward",
                                nvmax = dim(Boston)[2] - 1)
summary(forSubset)$outmat
```

```
##           crim zn  indus chas1 nox rm  age dis rad tax ptratio black lstat
## 1  ( 1 )  " "  " " " "   " "   " " " " " " " " " " " " " "     " "   "*"
## 2  ( 1 )  " "  " " " "   " "   " " "*" " " " " " " " " " "     " "   "*"
## 3  ( 1 )  " "  " " " "   " "   " " "*" " " " " " " " " "*"     " "   "*"
## 4  ( 1 )  " "  " " " "   " "   " " "*" " " "*" " " " " "*"     " "   "*"
## 5  ( 1 )  " "  " " " "   " "   "*" "*" " " "*" " " " " "*"     " "   "*"
## 6  ( 1 )  " "  " " " "   "*"   "*" "*" " " "*" " " " " "*"     " "   "*"
## 7  ( 1 )  " "  " " " "   "*"   "*" "*" " " "*" " " " " "*"     "*"   "*"
## 8  ( 1 )  " "  "*" " "   "*"   "*" "*" " " "*" " " " " "*"     "*"   "*"
## 9  ( 1 )  "*"  "*" " "   "*"   "*" "*" " " "*" " " " " "*"     "*"   "*"
## 10  ( 1 ) "*"  "*" " "   "*"   "*" "*" " " "*" "*" " " "*"     "*"   "*"
## 11  ( 1 ) "*"  "*" " "   "*"   "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 12  ( 1 ) "*"  "*" "*"   "*"   "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 13  ( 1 ) "*"  "*" "*"   "*"   "*" "*" "*" "*" "*" "*" "*"     "*"   "*"
```

```
summary(backSubset)$outmat
```

```
##           crim zn  indus chas1 nox rm  age dis rad tax ptratio black lstat
## 1  ( 1 )  " "  " " " "   " "   " " " " " " " " " " " " " "     " "   "*"
## 2  ( 1 )  " "  " " " "   " "   " " "*" " " " " " " " " " "     " "   "*"
## 3  ( 1 )  " "  " " " "   " "   " " "*" " " " " " " " " "*"     " "   "*"
## 4  ( 1 )  " "  " " " "   " "   " " "*" " " "*" " " " " "*"     " "   "*"
## 5  ( 1 )  " "  " " " "   " "   "*" "*" " " "*" " " " " "*"     " "   "*"
## 6  ( 1 )  " "  " " " "   " "   "*" "*" " " "*" " " " " "*"     "*"   "*"
## 7  ( 1 )  " "  " " " "   " "   "*" "*" " " "*" "*" " " "*"     "*"   "*"
## 8  ( 1 )  "*"  " " " "   " "   "*" "*" " " "*" "*" " " "*"     "*"   "*"
## 9  ( 1 )  "*"  " " " "   " "   "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 10  ( 1 ) "*"  "*" " "   " "   "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 11  ( 1 ) "*"  "*" " "   "*"   "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 12  ( 1 ) "*"  "*" "*"   "*"   "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 13  ( 1 ) "*"  "*" "*"   "*"   "*" "*" "*" "*" "*" "*" "*"     "*"   "*"
```

```
featSelect<-data.frame(1:6,cbind(c("lstat", "rm", "ptratio","dis","nox","chas"),
                              c("lstat", "rm", "ptratio","dis","nox","black")))
colnames(featSelect) <- c("Model Number", "Var. forward", "Var. backward")
featSelect
```

```
##   Model Number Var. forward Var. backward
## 1            1        lstat         lstat
## 2            2           rm            rm
## 3            3      ptratio       ptratio
## 4            4          dis           dis
## 5            5          nox           nox
## 6            6         chas         black
```

c. Comparing Variable selections

The best Subset selection and forward selection algorithms selected the same 6 variables, and in the same order. The backward selection algorithm matched the other two up until model 6, where the 6th variable selected was black as opposed to chas. I compare the coefficients from the different models

```
BestFowModel <- lm("medv ~ lstat + rm + ptratio + dis + nox + chas",
                   data = Boston)
backModel <- lm("medv ~ lstat + rm + ptratio + dis + nox + black",
                   data = Boston)
print("Coefficients for Best Subset and forward model:")
```

```
## [1] "Coefficients for Best Subset and forward model:"
```

```
summary(BestFowModel)$coefficients
```

```
##                Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)  36.9226340 4.55908556   8.098693 4.291836e-15
## lstat        -0.5698442 0.04744883 -12.009657 2.305468e-29
## rm            4.1118117 0.40721667  10.097356 6.144302e-22
## ptratio      -1.0027463 0.11273664  -8.894591 1.078984e-17
## dis          -1.1445857 0.16671617  -6.865475 1.975595e-11
## nox         -18.7404327 3.22732486  -5.806801 1.134454e-08
## chas1         3.2443048 0.88324944   3.673147 2.654731e-04
```

```
print("Coefficients for Backward Model:")
```

```
## [1] "Coefficients for Backward Model:"
```

```
summary(backModel)$coefficients
```

```
##                 Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)  30.516970426 4.959607224   6.153102 1.560882e-09
## lstat        -0.545496912 0.048414974 -11.267111 2.165763e-26
## rm            4.354807129 0.410753352  10.602000 8.019446e-24
## ptratio      -1.012059411 0.112597327  -8.988308 5.194370e-18
## dis          -1.159602736 0.166618639  -6.959622 1.077921e-11
## nox         -15.842368174 3.278907022  -4.831600 1.805153e-06
## black         0.009577916 0.002677202   3.577584 3.806043e-04
```

The coefficients for the first 5 variables are of the same sign, all significant even at an $\alpha = .01$ significance level, and all of similar size. The 6th variable in the Best Subset or forward case is chas, which has the same sign and same significance as black,with the difference in estimate for coefficients attributable to the difference in scale. Altogether, best subset, forward selection, and backward selection produce very similar models when $k = 6$.

1d. Model Selection via cross-validation

i. Creating 10 folds through sampling, and Matrix for Results

I orchestrate what is desired, including making a subset of only the variables for ease in generation of MSE:

```
set.seed(1)
dataCV <- subset.data.frame(Boston, select = c("medv","lstat", "rm","ptratio",
                                               "dis", "nox","chas"))
obs <-1:dim(dataCV)[1]
samples <- list()
for(i in 1:10){
  if(i < 10){
```

```r
  samples[[i]] <- sample(obs,size=(as.integer(0.1 * dim(dataCV)[1]) + (i %%2)))
  obs <- setdiff(obs, unlist(samples))
  } else {
    samples[[i]] <- setdiff(obs, samples)
  }
}
resultsMatrix <- matrix(nrow = 6, ncol = 10)
rownames(resultsMatrix) = c("Model 1", "Model 2", "Model 3",
                            "Model 4", "Model 5", "Model 6")
```

ii. Training on 9 test sets, testing on final set

I train each of the models 10 times, leaving one set of points out for testing, and then predict on that last set. The test MSE for each case is shown with the row indicating the model, and the column indicating the test set.

```r
for (i in 1:10){
  for (j in 2:7){
    tempTrain <- dataCV[setdiff(1:506, samples[[i]]),1:j]
    tempTest <- dataCV[samples[[i]],1:j]
    tempMod <- lm("medv ~.", data = tempTrain)
    predictTemp <- predict(tempMod, tempTest)
    SSE <- sum(unname(predictTemp) - tempTest$medv)^2
    resultsMatrix[j - 1,i] <- SSE/(dim(tempTest)[1])
  }
}
print(resultsMatrix)
```

```
##              [,1]      [,2]        [,3]      [,4]      [,5]         [,6]
## Model 1 111.00650  13.04895  0.04854855 4.7873412 12.028912 1.864733e+01
## Model 2  70.73342  66.15611  0.18961511 0.2463749 39.168297 1.615226e-01
## Model 3  37.05265  51.86702  3.27453047 0.2705134  9.043447 5.078187e-01
## Model 4  53.81366  76.90127  1.39916956 0.2934775 27.547151 1.247094e+00
## Model 5  67.23151  66.67588 10.50302515 5.9725256 15.429756 8.812132e-04
## Model 6  71.72734 116.35896  7.59243692 7.4141596 19.402252 1.616269e+00
##              [,7]      [,8]     [,9]      [,10]
## Model 1 16.566405 26.695639 265.8201 12.4974706
## Model 2 28.629657  8.139520 186.4930  0.3748765
## Model 3 26.758044  1.424391 183.3662  0.7889736
## Model 4 27.531975  3.532213 214.7002  1.3474224
## Model 5  8.219110  3.401536 164.5363  0.9820933
## Model 6  3.652887 10.738473 167.7706  6.5224374
```

iii. Computing MSE

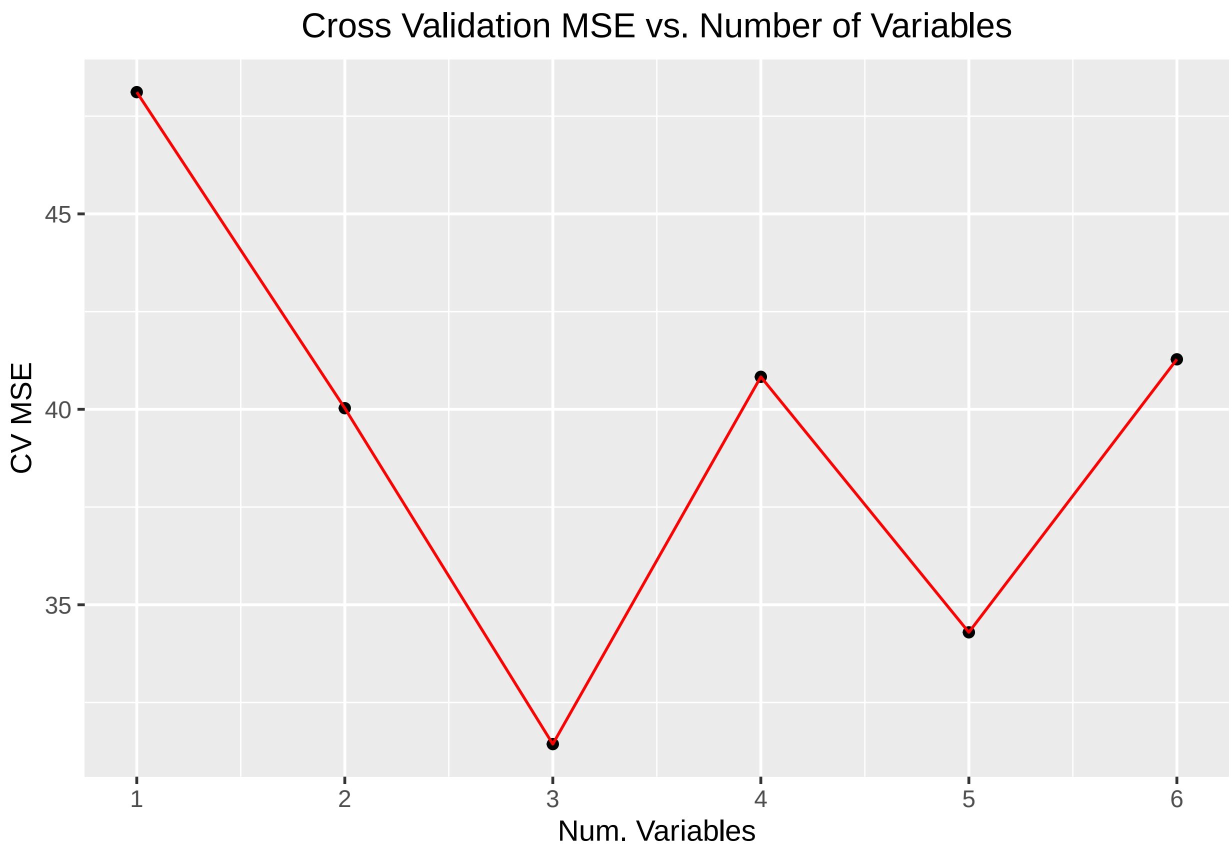
I use the apply function to get the CV MSE by taking the mean of each row.

```r
CV_MSE <- apply(X = resultsMatrix, MARGIN = 1, FUN = 'mean')
CV_MSE
```

```
##  Model 1  Model 2  Model 3  Model 4  Model 5  Model 6
## 48.11472 40.02923 31.43536 40.83136 34.29526 41.27958
```

I plot it against the number of variables used:

```r
cvdframe <- data.frame(cbind(1:6, unname(CV_MSE)))
colnames(cvdframe) <- c("Num. Variables", "CV MSE")
ggplot(data = cvdframe) + geom_point(aes(x = `Num. Variables`, y = `CV MSE`)) +
  geom_line(aes(x = `Num. Variables`, y = `CV MSE`), color = 'red') +
  labs(title = "Cross Validation MSE vs. Number of Variables") +
  scale_x_continuous(breaks=seq(1,6, by = 1)) +
  theme(plot.title = element_text(hjust = 0.5))
```



The model that was selected was the one with three variables, with the variables acquired from best subset selection being lstat, rm, and ptratio.

iv. Showing coefficients

I show the coefficients of the best model below:

```r
bestModel <- lm("medv ~ lstat + rm + ptratio", data = Boston)
bestModel$coefficients
```

```
## (Intercept)       lstat          rm     ptratio
##  18.5671115  -0.5718057   4.5154209  -0.9307226
```

2. Feature Selection and Model Selection

a. Subset selection

i. Forward, 2

The first variable that forward subset selection selects is x_1, as it has the lowest SSE of all predictors. It can then choose to add one of the other variables. The model with x_1 and x_3 has a lower SSE, so this model chooses x_1 and x_3.

ii. Backward, 1

The backward algorithm starts with model 8. Since removing variable x_1 creates the smallest increase in SSE, this one is removed first. Then, backward subset selection has 2 choices: remove x_2 and leave x_3, or visa versa. The SSE of the model with x_3 only is smaller than the one with , so x_3 is the variable selected.

iii. Backward, 2

In the problem above, the first variable was removed was x_1, so the algorithm arrived at the model with x_2 and x_3; thus, these 2 are selected.

b. Model Selection

Note that, from the full model, we know that:

$$s^2_{\text{full}} = \frac{SSR_{\text{full}}}{N-k} = \frac{3.05}{10-3} = \frac{3.05}{7}$$

And, from the null model, $SST = 25$

Using this value, I calculate all of the test stats for each of the models:

```
SSE <- c(25,9.5, 18,15,8.25,6.25,5.06,3.05)
SST <- 25
k <- c(0,1,1,1,2,2,2,3)
s2full = 3.05/7
Cp <- round((1/10)*(SSE + 2 *(k + 1)*s2full),3)
AIC <- round((1/s2full)*(SSE + 2 *(k + 1)*s2full),3)
BIC <- round((1/s2full)*(SSE + (log(10) *(k + 1)*s2full)),3)
r2adj <- round(1 - ((SSE/SST)*(10 - 1)/(10 - k - 1)),3)
sumFrame <- data.frame(cbind(1:8, SSE, Cp, AIC, BIC, r2adj))
colnames(sumFrame) <- c("Model", "SSE", "Cp", "AIC", "BIC", "R^2adj")
sumFrame
```

```
##   Model   SSE    Cp    AIC    BIC R^2adj
## 1     1 25.00 2.587 59.377 59.680  0.000
## 2     2  9.50 1.124 25.803 26.408  0.573
## 3     3 18.00 1.974 45.311 45.917  0.190
## 4     4 15.00 1.674 38.426 39.031  0.325
## 5     5  8.25 1.086 24.934 25.842  0.576
## 6     6  6.25 0.886 20.344 21.252  0.679
## 7     7  5.06 0.767 17.613 18.521  0.740
## 8     8  3.05 0.654 15.000 16.210  0.817
```

i. Best Model, Best Subset, Mallows Cp

For Best subset, Models 1, 2, 7, and 8 are considered (they have the lowest SSE for their fixed number of predictors). **Model 8** is the one selected of those by C_p, having lowest of these 4.

ii. Best Model, Forward, BIC

For the forward algorithm, Models 1, 2, 6, and then 8 are considered. Under BIC, **Model 8** is selected, having the lowest of these 4.

iii. Best Model, Backward, R^2adj

For the backward algorithm, models 8, 7, 4, and then 1 are considered. Under R^2_{adj}, **model 8** is selected, having the highest of these 4.