DAV_hw03_Model Selection and Model Validation

Weijun Zhu

October 27, 2019

```
library(ISLR)
library(leaps)
data(Hitters)
help(Hitters)
```

```
## starting httpd help server ... done
```

```
Hitters = na.omit(Hitters)
```

Exercise 1.

Perform stepwise regression starting with the full model using all the predictors of salary.

```
library(leaps)
regfit.full = regsubsets (Salary ~ .,Hitters)
summary(regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., Hitters)
## 19 Variables  (and intercept)
##             Forced in Forced out
## AtBat           FALSE      FALSE
## Hits            FALSE      FALSE
## HmRun           FALSE      FALSE
## Runs            FALSE      FALSE
## RBI             FALSE      FALSE
## Walks           FALSE      FALSE
## Years           FALSE      FALSE
## CAtBat          FALSE      FALSE
## CHits           FALSE      FALSE
## CHmRun          FALSE      FALSE
## CRuns           FALSE      FALSE
## CRBI            FALSE      FALSE
## CWalks          FALSE      FALSE
## LeagueN         FALSE      FALSE
## DivisionW       FALSE      FALSE
## PutOuts         FALSE      FALSE
## Assists         FALSE      FALSE
## Errors          FALSE      FALSE
## NewLeagueN      FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns
```

```
## 1  ( 1 ) " "   " "  " "   " "  " " " " " "   " "   " "    " "   " "    " "
## 2  ( 1 ) " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
## 3  ( 1 ) " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
## 4  ( 1 ) " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
## 5  ( 1 ) "*"   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
## 6  ( 1 ) "*"   "*"  " "   " "  " " " " "*"   " "   " "    " "   " "    " "
## 7  ( 1 ) " "   "*"  " "   " "  " " " " "*"   " "   "*"    "*"   "*"    " "
## 8  ( 1 ) "*"   "*"  " "   " "  " " " " "*"   " "   " "    " "   "*"    "*"
##          CRBI CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1  ( 1 ) "*"  " "    " "     " "       " "     " "     " "    " "
## 2  ( 1 ) "*"  " "    " "     " "       " "     " "     " "    " "
## 3  ( 1 ) "*"  " "    " "     " "       "*"     " "     " "    " "
## 4  ( 1 ) "*"  " "    " "     "*"       "*"     " "     " "    " "
## 5  ( 1 ) "*"  " "    " "     "*"       "*"     " "     " "    " "
## 6  ( 1 ) "*"  " "    " "     "*"       "*"     " "     " "    " "
## 7  ( 1 ) " "  " "    " "     "*"       "*"     " "     " "    " "
## 8  ( 1 ) " "  "*"    " "     "*"       "*"     " "     " "    " "
```

Exercise 2.

Compare the coefficients of the stepwise model and the full model.

```
coef(regfit.full ,6)
```

```
##  (Intercept)        AtBat         Hits        Walks         CRBI
##   91.5117981   -1.8685892    7.6043976    3.6976468    0.6430169
##    DivisionW      PutOuts
## -122.9515338    0.2643076
```

Exercise 3.

Which variable of variables did stepwise drop from the full model?

```
coef(regfit.full ,6)
```

```
##  (Intercept)        AtBat         Hits        Walks         CRBI
##   91.5117981   -1.8685892    7.6043976    3.6976468    0.6430169
##    DivisionW      PutOuts
## -122.9515338    0.2643076
```

Conclusion: The best model of 6 variables show above.

Exercise 4.

Perform a cross-validation of the stepwise model.

```
k = 3
set.seed(9)
folds = sample(1:k,nrow(Hitters),replace=TRUE)
cv.errors = matrix(NA,k,19, dimnames =list(NULL , paste(1:19)))
```

```
predict.regsubsets = function (object ,newdata ,id,...){
  form=as.formula(object$call [[2]])
  mat=model.matrix(form,newdata)
  coefi=coef(object ,id=id)
  xvars=names(coefi)
  mat[,xvars]%*%coefi
  }
```

```
for(j in 1:k){
  best.fit = regsubsets (Salary ~ .,data = Hitters[folds!=j,], nvmax=19)
  for(i in 1:19){
    pred = predict.regsubsets(best.fit ,Hitters[folds == j,],id = i)
    cv.errors[ j,i] = mean((Hitters$Salary[folds==j]-pred)^2)
  }
}
```

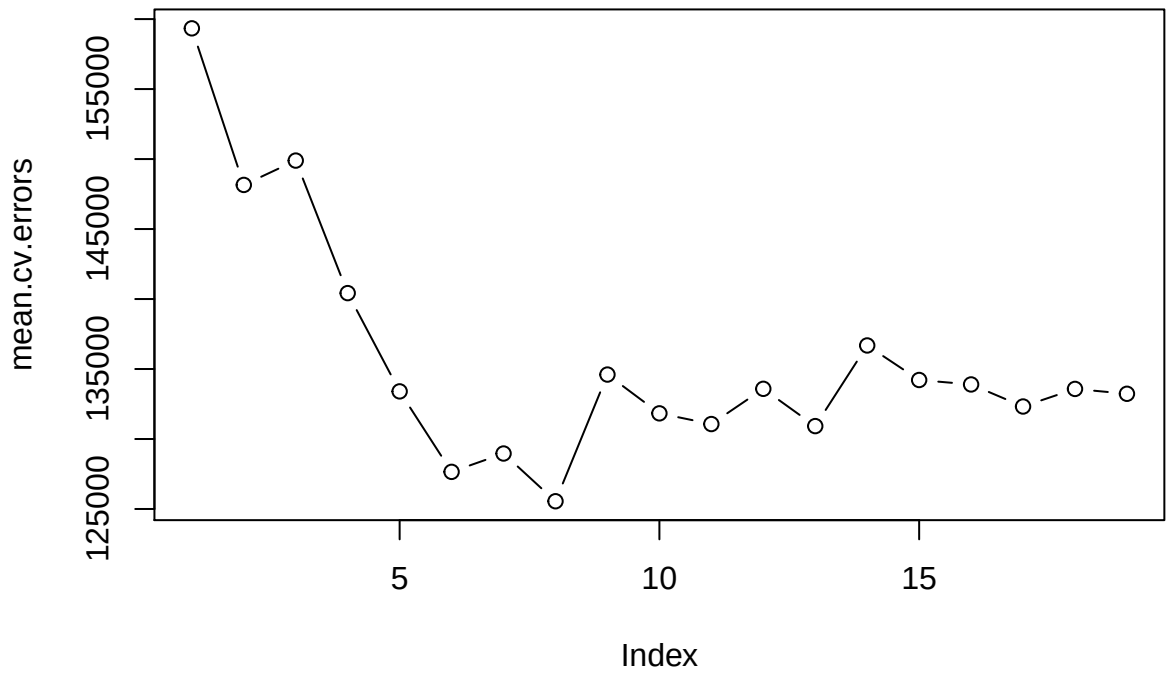
Exercise 5.

Compare the two models using the mse's from the cross-validations with number of folds equal to 3. Which model gives the better mse?

```
mean.cv.errors = apply(cv.errors,2,mean)
mean.cv.errors
```

```
##        1        2        3        4        5        6        7        8
## 159337.9 148152.5 149891.8 140422.9 133402.0 127653.6 128963.8 125553.9
##        9       10       11       12       13       14       15       16
## 134604.0 131833.3 131067.8 133584.5 130921.2 136683.2 134215.1 133896.9
##       17       18       19
## 132322.0 133577.8 133231.6
```

```
par(mfrow=c(1,1))
plot(mean.cv.errors,type = 'b')
```


mean.cv.errors
125000
135000
145000
155000
5
10
15
Index