SDM_Assignment4_1

Sri Balaji Muruganandam

14/11/2021

1) For the Boston data in the ISLR2 package:

> library(ISLR2)

> data(Boston)

> ?Boston

Using best subset regression analysis fit models for “medv” (median value of owner-occupied homes in $1000s). Perform model selection using the AIC, BIC, five-and tenfold cross-validation, and bootstrap .632 estimates of prediction error. Comment on your results and the differences in the selected model.

```
rm(list = ls())
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.1.1
```

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.1.1
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.1
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

```
## Loading required package: lattice
```

```
data(Boston)
dim(Boston)
```

```
## [1] 506  13
```

Splitting the data into training and the test data

```
set.seed(23)
random_index = sample(c(1:nrow(Boston)), size = round(8/10 * nrow(Boston)), replace = FALSE)
train_data <- Boston[random_index,]
test_data <- Boston[-random_index,]

dim(train_data)
```

```
## [1] 405  13
```

```
dim(test_data)
```

```
## [1] 101  13
```

Performing Exhaustive Selection

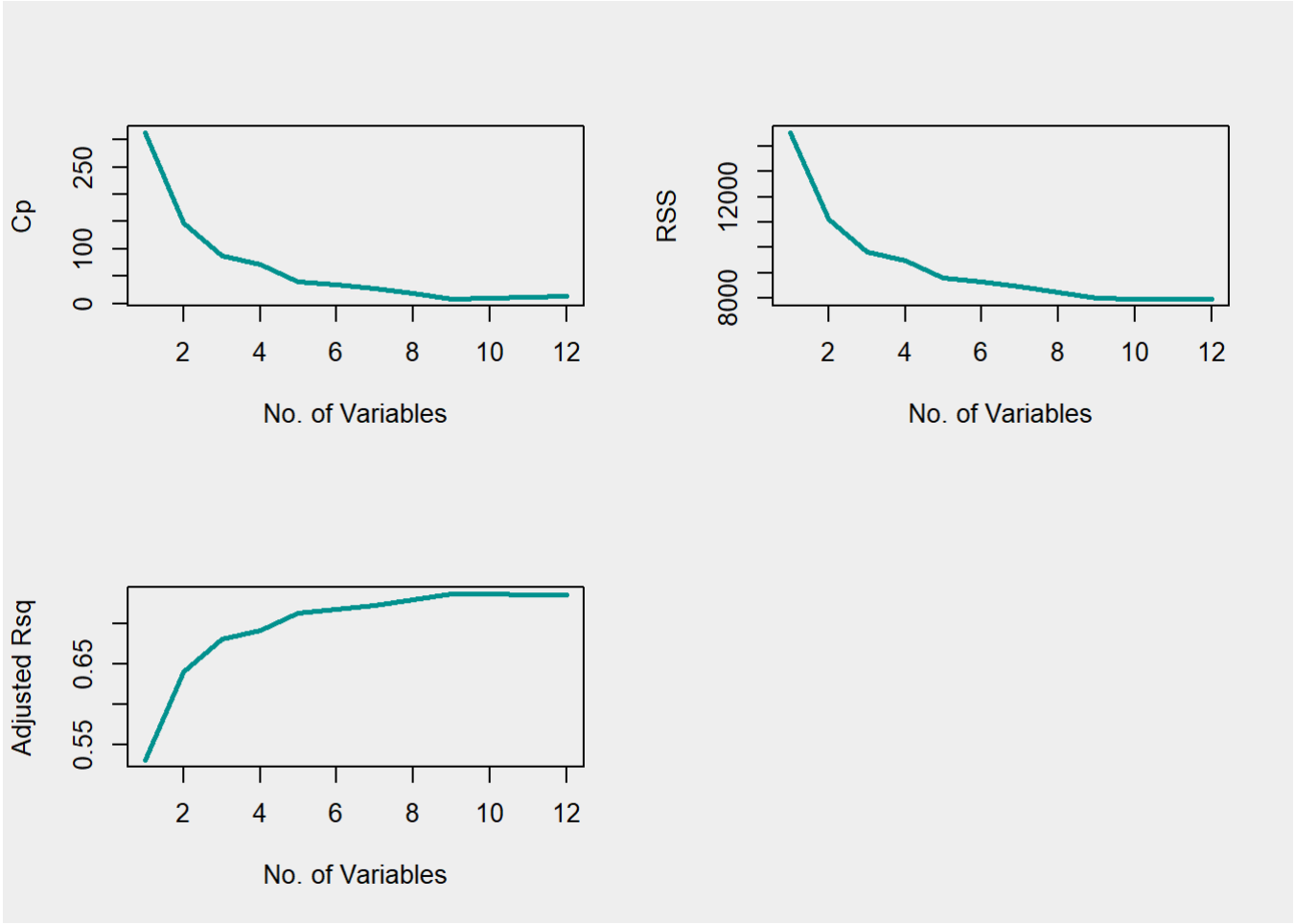
Defining the model

```
ex_subset <- regsubsets(medv~., data = train_data, nbest = 1, nvmax = 15, method = "exhaustiv
e")
ex_summary <- summary(ex_subset)
names(ex_summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

Plotting Exhaustive Selection Measurements

```
par(mfrow = c(2,2),bg = '#EEEEEE')
plot(ex_summary$cp, xlab = "No. of Variables", ylab = "Cp", type = "l",col = "#00918E",lwd =
2.5)
plot(ex_summary$rss, xlab = "No. of Variables", ylab = "RSS", type = "l",col = "#00918E",lwd
 = 2.5)
plot(ex_summary$adjr2, xlab = "No. of Variables", ylab = "Adjusted Rsq", type = "l",col = "#0
0918E",lwd = 2.5)
```



Finding the optimal model measures selection

```
which(ex_summary$cp == min(ex_summary$cp))
```

```
## [1] 9
```

```
which(ex_summary$bic == min(ex_summary$bic))
```

```
## [1] 9
```

```
which(ex_summary$rss == min(ex_summary$rss))
```

```
## [1] 12
```

```
which(ex_summary$adjr2 == max(ex_summary$adjr2))
```

```
## [1] 9
```

```
print(min(ex_summary$rss))
```

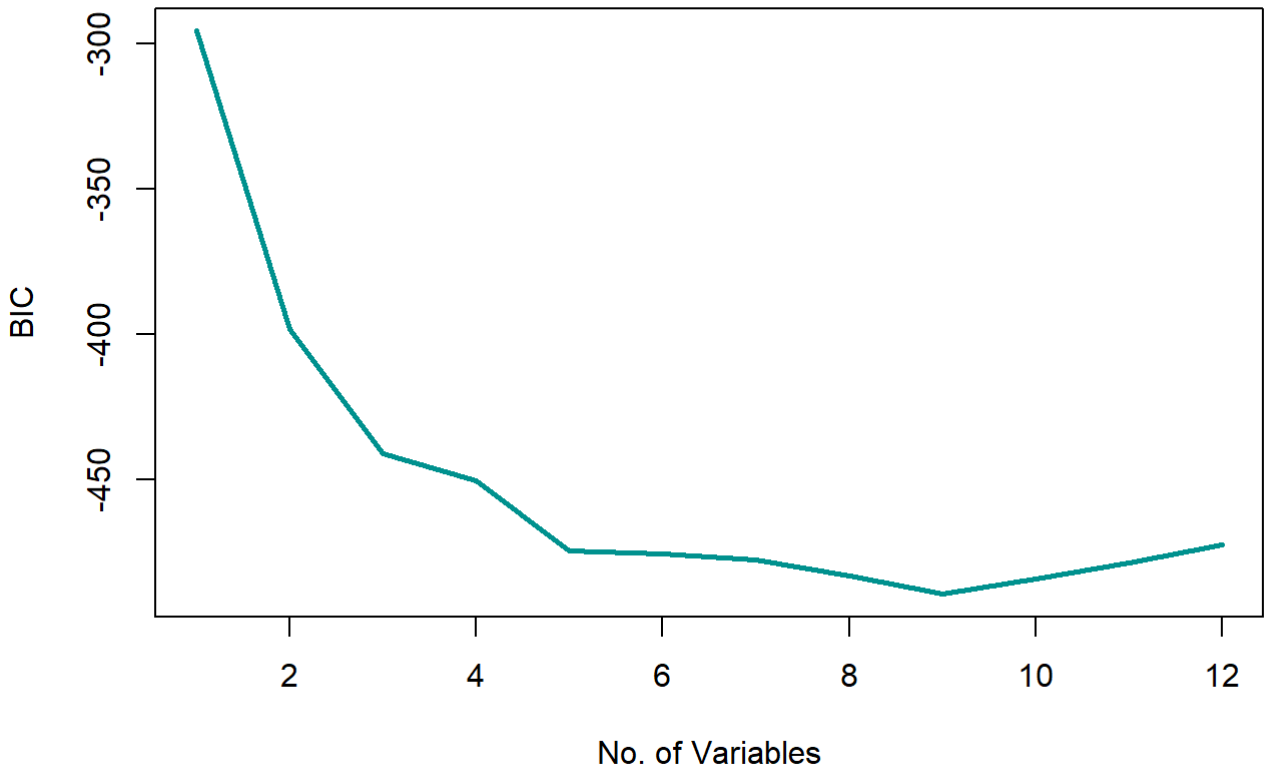
```
## [1] 7972.052
```

```
print(max(ex_summary$adjr2))
```

```
## [1] 0.7365197
```

Performing model selection using BIC

```
plot(ex_summary$bic, xlab = "No. of Variables", ylab = "BIC", type = "l",col = "#00918E",lwd
 = 2.5)
```




```
which(ex_summary$bic == min(ex_summary$bic))
```

```
## [1] 9
```

Using BIC, the data with 9 predictors is found to be the best among others

Performing 5 fold cross validation

```
set.seed(23)
five_fold = trainControl(method = "cv", number = 5)
five_fold_fit  = train(medv~., data = Boston, method = "lm", trControl = five_fold)
summary(five_fold_fit)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1304  -2.7673  -0.5814   1.9414  26.2526
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.617270   4.936039   8.431 3.79e-16 ***
## crim         -0.121389   0.033000  -3.678 0.000261 ***
## zn            0.046963   0.013879   3.384 0.000772 ***
## indus         0.013468   0.062145   0.217 0.828520
## chas          2.839993   0.870007   3.264 0.001173 **
## nox         -18.758022   3.851355  -4.870 1.50e-06 ***
## rm            3.658119   0.420246   8.705  < 2e-16 ***
## age           0.003611   0.013329   0.271 0.786595
## dis          -1.490754   0.201623  -7.394 6.17e-13 ***
## rad           0.289405   0.066908   4.325 1.84e-05 ***
## tax          -0.012682   0.003801  -3.337 0.000912 ***
## ptratio      -0.937533   0.132206  -7.091 4.63e-12 ***
## lstat        -0.552019   0.050659 -10.897  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.798 on 493 degrees of freedom
## Multiple R-squared:  0.7343, Adjusted R-squared:  0.7278
## F-statistic: 113.5 on 12 and 493 DF,  p-value: < 2.2e-16
```

Performing 10 fold cross validation

```
set.seed(23)
five_fold = trainControl(method = "cv", number = 10)
five_fold_fit  = train(medv~., data = Boston, method = "lm", trControl = five_fold)
summary(five_fold_fit)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1304  -2.7673  -0.5814   1.9414  26.2526
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.617270   4.936039   8.431 3.79e-16 ***
## crim         -0.121389   0.033000  -3.678 0.000261 ***
## zn            0.046963   0.013879   3.384 0.000772 ***
## indus         0.013468   0.062145   0.217 0.828520
## chas          2.839993   0.870007   3.264 0.001173 **
## nox         -18.758022   3.851355  -4.870 1.50e-06 ***
## rm            3.658119   0.420246   8.705  < 2e-16 ***
## age           0.003611   0.013329   0.271 0.786595
## dis          -1.490754   0.201623  -7.394 6.17e-13 ***
## rad           0.289405   0.066908   4.325 1.84e-05 ***
## tax          -0.012682   0.003801  -3.337 0.000912 ***
## ptratio      -0.937533   0.132206  -7.091 4.63e-12 ***
## lstat        -0.552019   0.050659 -10.897  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.798 on 493 degrees of freedom
## Multiple R-squared:  0.7343, Adjusted R-squared:  0.7278
## F-statistic: 113.5 on 12 and 493 DF,  p-value: < 2.2e-16
```