Variable selection using the Boston Housing Data set

Symon Kimitei

4/20/2020

R Markdown

R Markdown

```
#install.packages("readxl")
library(readxl)
#install.packages("dplyr")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#install.packages("ggplot2")
library(ggplot2)
library(lattice)
#library(rpart)
library(Matrix)
#install.packages("kableExtra")
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
#install.packages("plotly")
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
#install.packages("reshape2")
library(reshape2)

#install.packages("caret")
library(caret)

#install.packages("car")
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(dplyr)
#install.packages("car")
library(carData)

#install.packages("caTools") # For sample.split

library(caTools)

#install.packages("performance")  # for VIF, checkmulticollinearity, etc
library(performance)

# install.packages("modelr")  # package for mse, rmse, etc
library(modelr)
```

```
##
## Attaching package: 'modelr'
```

```
## The following objects are masked from 'package:performance':
##
##     mse, rmse
```

```
setwd("C:/Users/kimit/OneDrive/Desktop/r_code/boston")
boston<-read.csv("boston.csv", header = T, sep = ",", row.names =1)
#head(boston)
boston$ID <- 1:nrow(boston)
#boston<-boston[!duplicated(boston$ID), ]



library(dplyr) # from version 1.0.0

col_idx <- grep("ID", names(boston))
boston <- boston[, c(col_idx, (1:ncol(boston))[-col_idx])]
names(boston)
```

```
##  [1] "ID"      "crim"    "zn"      "indus"   "chas"    "nox"     "rm"
##  [8] "age"     "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"
## [15] "medv"
```

```
head(boston)
```

```
##   ID    crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1  1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2  2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3  3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4  4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5  5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6  6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

```
# Display the class of the R object housing.df
glimpse(boston)
```

```
## Rows: 506
## Columns: 15
## $ ID      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,~
## $ crim    <dbl> 0.00632, 0.02731, 0.02729, 0.03237, 0.06905, 0.02985, 0.08829,~
## $ zn      <dbl> 18.0, 0.0, 0.0, 0.0, 0.0, 0.0, 12.5, 12.5, 12.5, 12.5, 12.5, 1~
## $ indus   <dbl> 2.31, 7.07, 7.07, 2.18, 2.18, 2.18, 7.87, 7.87, 7.87, 7.87, 7.~
## $ chas    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ nox     <dbl> 0.538, 0.469, 0.469, 0.458, 0.458, 0.458, 0.524, 0.524, 0.524,~
## $ rm      <dbl> 6.575, 6.421, 7.185, 6.998, 7.147, 6.430, 6.012, 6.172, 5.631,~
## $ age     <dbl> 65.2, 78.9, 61.1, 45.8, 54.2, 58.7, 66.6, 96.1, 100.0, 85.9, 9~
## $ dis     <dbl> 4.0900, 4.9671, 4.9671, 6.0622, 6.0622, 6.0622, 5.5605, 5.9505~
## $ rad     <int> 1, 2, 2, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 4,~
## $ tax     <int> 296, 242, 242, 222, 222, 222, 311, 311, 311, 311, 311, 311, 31~
## $ ptratio <dbl> 15.3, 17.8, 17.8, 18.7, 18.7, 18.7, 15.2, 15.2, 15.2, 15.2, 15~
## $ black   <dbl> 396.90, 396.90, 392.83, 394.63, 396.90, 394.12, 395.60, 396.90~
## $ lstat   <dbl> 4.98, 9.14, 4.03, 2.94, 5.33, 5.21, 12.43, 19.15, 29.93, 17.10~
## $ medv    <dbl> 24.0, 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9, 15~
```

```
#set a seed
set.seed(2021)
split <- sample.split(boston,SplitRatio =0.80)
train <- subset(boston,split==TRUE)
test <- subset(boston,split==FALSE)

dim(train)
```

```
## [1] 406  15
```

```
# DEVELOPING THE HOUSING PRICE PREDICTION MODEL
#=================================================================
# crim - per capita crime rate by town
# zn - proportion of residential land zoned for lots over 25,000 sq.ft
# indus - proportion of non-retail business acres per town
# chas - Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
# nox - nitric oxides concentration (parts per 10 million)
# rm - average number of rooms per dwelling
# age - proportion of owner-occupied units built prior to 1940
# dis - weighted distances to five Boston employment centres
# rad - index of accessibility to radial highways
# tax - full-value property-tax rate per USD 10,000
# ptratio- pupil-teacher ratio by town
# black - the proportion of blacks by town
# lstat - percentage of lower status of the population
# medv - median home value in various neighborhoods(median value of owner-occupied homes in USD
 1000)

# Fit the model by expressing all the parameters as follows:

#model <- lm(medv ~ crim + zn + indus+ chas + nox + rm + tax + age + dis
#            + rad + tax + ptratio+ black + lstat , data = train)

# OR
train$ID<-NULL
model <- lm(medv ~., data = train)
summary(model)
```

```
##
## Call:
## lm(formula = medv ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.0865  -2.8566  -0.6432   1.9900  27.4513
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.106108   5.507795   6.374 5.18e-10 ***
## crim         -0.135962   0.034398  -3.953 9.17e-05 ***
## zn            0.032381   0.015236   2.125  0.03419 *
## indus         0.009620   0.066167   0.145  0.88447
## chas          2.792161   0.982623   2.842  0.00472 **
## nox         -16.909701   4.257663  -3.972 8.50e-05 ***
## rm            4.134143   0.443788   9.316  < 2e-16 ***
## age          -0.005917   0.014523  -0.407  0.68393
## dis          -1.375061   0.216335  -6.356 5.75e-10 ***
## rad           0.299541   0.075060   3.991 7.86e-05 ***
## tax          -0.012324   0.004188  -2.942  0.00345 **
## ptratio      -0.968265   0.146633  -6.603 1.31e-10 ***
## black         0.007072   0.002880   2.455  0.01450 *
## lstat        -0.489655   0.055095  -8.887  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.709 on 392 degrees of freedom
## Multiple R-squared:  0.7432, Adjusted R-squared:  0.7347
## F-statistic: 87.28 on 13 and 392 DF,  p-value: < 2.2e-16
```

Display the VIF values for each predictor

```
all_vifs <- car::vif(model)
print(all_vifs)
```

```
##     crim       zn    indus     chas      nox       rm      age      dis
## 1.687150 2.273767 3.771835 1.097340 4.339054 1.819928 3.000492 3.802198
##      rad      tax  ptratio    black    lstat
## 7.805695 9.145866 1.820140 1.321306 2.714435
```

Plot VIF values for each predictor by using the Performance package. All the VIF values are less than 10.

Therefore it is Ok to proceed without centering the model. One reviewing the p-values from the summary table,

it is evident that the p-values for indus and age predictors are not significant since the p-values>0.05

The process of automatic variable selection should identitifes these as variables that need to be removed.

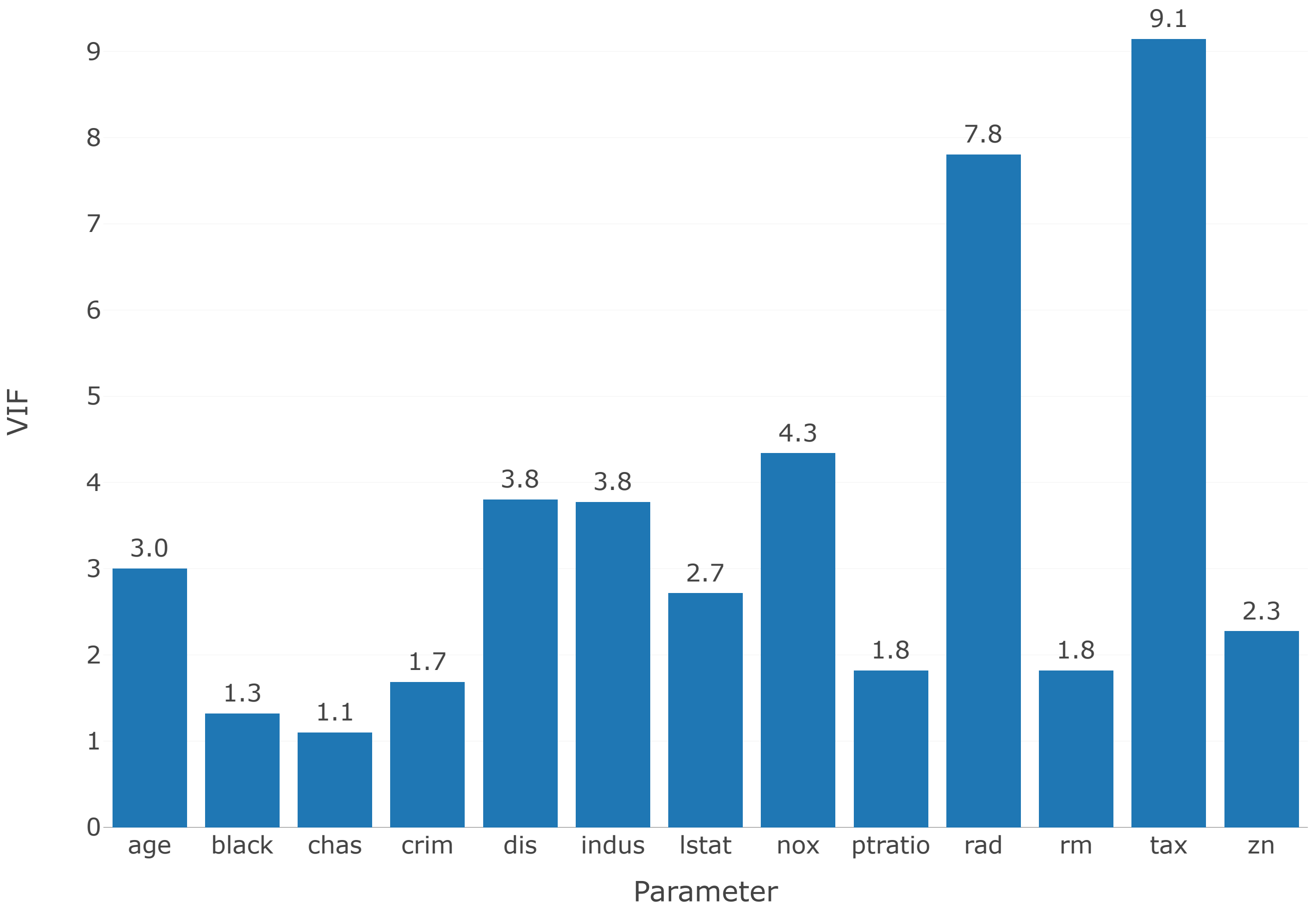
```
vif_dat <- check_collinearity(model)
vif_dat$Parameter
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"    
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"
```

```
vif_dat$Parameter<-c("crim","zn","indus","chas","nox","rm","age","dis","rad","tax","ptratio","bl
ack","lstat")


fig <- plot_ly(vif_dat, type='bar', x = ~Parameter, y = ~VIF, text = ~Parameter, name="",
               hovertemplate = paste('VIF: %{y}', ''),
               texttemplate = '%{y:.2s}', textposition = 'outside')

fig <- fig %>% layout(uniformtext=list(minsize=8, mode='hide'))
fig
```



Print the metrics of the model before applying forward, backward and stepwise selection

```
data.frame(
  Rsq = rsquare(model, data = train),
  RMSE = rmse(model,data=train),
  MAE= mae(model, data = train),
  AIC = AIC(model),
  BIC= BIC(model),
  AdRsq=summary(model)$adj.r.squared
)
```

```
##         Rsq     RMSE      MAE      AIC     BIC     AdRsq
## 1 0.7432179 4.627379 3.304193 2426.154 2486.25 0.7347022
```

Store

```
library(broom)
```

```
##
## Attaching package: 'broom'
```

```
## The following object is masked from 'package:modelr':
##
##     bootstrap
```

```
results1<-glance(model)
write.csv(results1,"metrics2.csv")
```

```
# Apply Forward Selection: Starting with the intercept-only model before adding predictors to th
e model and evaluating their usefulness
model1 <- lm(medv ~1, data = train)
model_forwd<-step(model1,direction="forward",scope=formula(model))
```

```
## Start:  AIC=1797.94
## medv ~ 1
## 
##           Df Sum of Sq   RSS    AIC
## + lstat    1   18058.4 15797 1490.5
## + rm       1   16438.0 17418 1530.1
## + ptratio  1    8966.8 24889 1675.0
## + indus    1    7792.1 26064 1693.8
## + tax      1    7242.7 26613 1702.2
## + nox      1    5954.5 27901 1721.4
## + rad      1    4894.9 28961 1736.5
## + crim     1    4835.2 29021 1737.4
## + age      1    4769.0 29087 1738.3
## + zn       1    4589.0 29267 1740.8
## + black    1    3390.6 30465 1757.1
## + dis      1    2060.2 31796 1774.5
## + chas     1    1249.9 32606 1784.7
## <none>                 33856 1797.9
## 
## Step:  AIC=1490.46
## medv ~ lstat
## 
##           Df Sum of Sq   RSS    AIC
## + rm       1    3683.0 12114 1384.7
## + ptratio  1    2294.8 13502 1428.7
## + chas     1     629.4 15168 1476.0
## + dis      1     491.8 15306 1479.6
## + tax      1     342.8 15454 1483.6
## + crim     1     278.2 15519 1485.2
## + age      1     182.3 15615 1487.8
## + black    1     160.5 15637 1488.3
## + zn       1     137.8 15660 1488.9
## + indus    1     101.9 15695 1489.8
## + rad      1      89.4 15708 1490.2
## <none>                 15797 1490.5
## + nox      1       0.1 15797 1492.5
## 
## Step:  AIC=1384.69
## medv ~ lstat + rm
## 
##           Df Sum of Sq   RSS    AIC
## + ptratio  1   1419.01 10695 1336.1
## + crim     1    493.43 11621 1369.8
## + tax      1    450.04 11664 1371.3
## + chas     1    396.44 11718 1373.2
## + black    1    320.44 11794 1375.8
## + rad      1    256.38 11858 1378.0
## + dis      1    218.02 11896 1379.3
## <none>                 12114 1384.7
## + indus    1     54.94 12059 1384.8
## + zn       1     37.54 12077 1385.4
## + nox      1     22.37 12092 1385.9
## + age      1      1.47 12113 1386.6
```

```
##
## Step:  AIC=1336.11
## medv ~ lstat + rm + ptratio
##
##         Df Sum of Sq   RSS    AIC
## + dis    1    332.75 10363 1325.3
## + crim   1    256.98 10438 1328.2
## + black  1    242.17 10453 1328.8
## + chas   1    241.62 10454 1328.8
## + tax    1     82.38 10613 1335.0
## <none>               10695 1336.1
## + nox    1     29.43 10666 1337.0
## + zn     1     25.91 10669 1337.1
## + age    1     19.59 10676 1337.4
## + rad    1      3.25 10692 1338.0
## + indus  1      0.17 10695 1338.1
##
## Step:  AIC=1325.28
## medv ~ lstat + rm + ptratio + dis
##
##         Df Sum of Sq     RSS    AIC
## + nox    1    600.48  9762.1 1303.0
## + crim   1    398.36  9964.2 1311.4
## + black  1    328.14 10034.4 1314.2
## + tax    1    277.64 10084.9 1316.2
## + indus  1    173.84 10188.7 1320.4
## + chas   1    161.61 10201.0 1320.9
## + age    1     84.01 10278.6 1324.0
## + rad    1     64.44 10298.1 1324.8
## <none>               10362.6 1325.3
## + zn     1     48.35 10314.2 1325.4
##
## Step:  AIC=1303.04
## medv ~ lstat + rm + ptratio + dis + nox
##
##         Df Sum of Sq    RSS    AIC
## + crim   1   279.502 9482.6 1293.2
## + chas   1   246.003 9516.1 1294.7
## + black  1   189.498 9572.6 1297.1
## <none>               9762.1 1303.0
## + zn     1    45.145 9716.9 1303.2
## + tax    1    33.699 9728.4 1303.6
## + indus  1     9.930 9752.2 1304.6
## + age    1     8.942 9753.1 1304.7
## + rad    1     7.620 9754.5 1304.7
##
## Step:  AIC=1293.25
## medv ~ lstat + rm + ptratio + dis + nox + crim
##
##         Df Sum of Sq    RSS    AIC
## + chas   1   213.975 9268.6 1286.0
## + rad    1   132.796 9349.8 1289.5
## + black  1   109.255 9373.3 1290.5
## + zn     1    91.451 9391.1 1291.3
## <none>                9482.6 1293.2
## + age    1    15.013 9467.6 1294.6
## + indus  1     7.030 9475.6 1295.0
## + tax    1     0.335 9482.2 1295.2
## 
## Step:  AIC=1285.98
## medv ~ lstat + rm + ptratio + dis + nox + crim + chas
## 
##         Df Sum of Sq    RSS    AIC
## + rad    1   139.475 9129.1 1281.8
## + black  1    95.480 9173.1 1283.8
## + zn     1    93.751 9174.9 1283.8
## <none>                9268.6 1286.0
## + indus  1    15.418 9253.2 1287.3
## + age    1    15.251 9253.4 1287.3
## + tax    1     1.613 9267.0 1287.9
## 
## Step:  AIC=1281.83
## medv ~ lstat + rm + ptratio + dis + nox + crim + chas + rad
## 
##         Df Sum of Sq    RSS    AIC
## + tax    1   191.958 8937.2 1275.2
## + black  1   146.694 8982.4 1277.2
## + zn     1    64.046 9065.1 1281.0
## <none>                9129.1 1281.8
## + indus  1    32.332 9096.8 1282.4
## + age    1     7.428 9121.7 1283.5
## 
## Step:  AIC=1275.2
## medv ~ lstat + rm + ptratio + dis + nox + crim + chas + rad + 
##     tax
## 
##         Df Sum of Sq    RSS    AIC
## + black  1   132.379 8804.8 1271.1
## + zn     1   108.129 8829.0 1272.3
## <none>                8937.2 1275.2
## + age    1     7.227 8929.9 1276.9
## + indus  1     0.112 8937.1 1277.2
## 
## Step:  AIC=1271.14
## medv ~ lstat + rm + ptratio + dis + nox + crim + chas + rad + 
##     tax + black
## 
##         Df Sum of Sq    RSS    AIC
## + zn     1   107.071 8697.7 1268.2
## <none>                8804.8 1271.1
## + age    1    11.095 8793.7 1272.6
## + indus  1     0.002 8804.8 1273.1
## 
## Step:  AIC=1268.17
## medv ~ lstat + rm + ptratio + dis + nox + crim + chas + rad + 
##     tax + black + zn
## 
##         Df Sum of Sq    RSS    AIC
## <none>                8697.7 1268.2
## + age    1    3.7227 8694.0 1270.0
## + indus  1    0.5104 8697.2 1270.2
```

Find the AdjR^2, AIC and BIC values for model with AIC=1230.54

```
fwd_model1<-lm(medv~lstat+rm+ptratio+chas+black+dis+nox+zn+crim+rad+tax,data=train)
summary(fwd_model1)
```

```
## 
## Call:
## lm(formula = medv ~ lstat + rm + ptratio + chas + black + dis + 
##     nox + zn + crim + rad + tax, data = train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -16.011  -2.882  -0.618   2.005  27.248 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  35.205808   5.476369   6.429 3.73e-10 ***
## lstat        -0.496549   0.051280  -9.683  < 2e-16 ***
## rm            4.089675   0.430791   9.493  < 2e-16 ***
## ptratio      -0.968272   0.144662  -6.693 7.52e-11 ***
## chas          2.810076   0.973183   2.888  0.00410 ** 
## black         0.006995   0.002868   2.439  0.01517 *  
## dis          -1.356644   0.199839  -6.789 4.18e-11 ***
## nox         -17.190211   3.977753  -4.322 1.96e-05 ***
## zn            0.033071   0.015016   2.202  0.02822 *  
## crim         -0.136355   0.034294  -3.976 8.34e-05 ***
## rad           0.298813   0.072496   4.122 4.59e-05 ***
## tax          -0.012128   0.003841  -3.158  0.00171 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4.698 on 394 degrees of freedom
## Multiple R-squared:  0.7431, Adjusted R-squared:  0.7359 
## F-statistic: 103.6 on 11 and 394 DF,  p-value: < 2.2e-16
```

```
extractAIC(fwd_model1)
```

```
## [1]   12.000 1268.172
```

```
BIC<-AIC(fwd_model1,k = log(length(fwd_model1)))
BIC
```

```
## [1] 2428.654
```

Find the AdjR^2, AIC and BIC values for model with AIC=1250.73

```
fwd_model2<-lm(medv~lstat+rm+ptratio+chas+black+dis+nox+zn+crim+rad,data=train)
summary(fwd_model2)
```

```
##
## Call:
## lm(formula = medv ~ lstat + rm + ptratio + chas + black + dis +
##     nox + zn + crim + rad, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.6017  -3.0956  -0.5284   1.8962  27.2336
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.295423   5.504305   6.049 3.38e-09 ***
## lstat        -0.502395   0.051825  -9.694  < 2e-16 ***
## rm            4.266562   0.431956   9.877  < 2e-16 ***
## ptratio      -1.017854   0.145431  -6.999 1.11e-11 ***
## chas          2.966703   0.982892   3.018 0.002706 **
## black         0.007400   0.002897   2.554 0.011027 *
## dis          -1.311487   0.201577  -6.506 2.34e-10 ***
## nox         -20.155407   3.908964  -5.156 3.99e-07 ***
## zn            0.025346   0.014983   1.692 0.091503 .
## crim         -0.134801   0.034678  -3.887 0.000119 ***
## rad           0.117158   0.044612   2.626 0.008972 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.752 on 395 degrees of freedom
## Multiple R-squared:  0.7366, Adjusted R-squared:  0.7299
## F-statistic: 110.5 on 10 and 395 DF,  p-value: < 2.2e-16
```

```
extractAIC(fwd_model2)
```

```
## [1]   11.000 1276.319
```

```
BIC<-AIC(fwd_model2,k = log(length(fwd_model2)))
BIC
```

```
## [1] 2436.316
```

Find the AdjR^2, AIC and BIC values for model with AIC=1254.16

```
fwd_model3<-lm(medv~lstat+rm+ptratio+chas+black+dis+nox+zn+crim,data=train)
summary(fwd_model3)
```

```
##
## Call:
## lm(formula = medv ~ lstat + rm + ptratio + chas + black + dis +
##     nox + zn + crim, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1127  -2.9186  -0.6485   1.8329  28.5382
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  28.514769   5.233090   5.449 8.92e-08 ***
## lstat        -0.502393   0.052210  -9.623  < 2e-16 ***
## rm            4.399933   0.432142  10.182  < 2e-16 ***
## ptratio      -0.851232   0.131831  -6.457 3.13e-10 ***
## chas          2.940555   0.990132   2.970  0.00316 **
## black         0.006039   0.002872   2.103  0.03611 *
## dis          -1.311922   0.203072  -6.460 3.07e-10 ***
## nox         -16.047964   3.608986  -4.447 1.13e-05 ***
## zn            0.031127   0.014930   2.085  0.03772 *
## crim         -0.098285   0.032004  -3.071  0.00228 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.787 on 396 degrees of freedom
## Multiple R-squared:  0.732,  Adjusted R-squared:  0.7259
## F-statistic: 120.2 on 9 and 396 DF,  p-value: < 2.2e-16
```

```
extractAIC(fwd_model3)
```

```
## [1]   10.000 1281.346
```

```
BIC<-AIC(fwd_model3,k = log(length(fwd_model3)))
BIC
```

```
## [1] 2440.858
```

We could also search through the possible models in a backwards fashion using BIC.

To do so, we again use the step() function. In Backward selection, R labels BIC as AIC

```
# Apply Backward Selection: Starting with the full model before removing some of the predictors
 that are not useful.
model2 <- lm(medv ~., data = train)
n = length(resid(model2))
model_back_bic = step(model2, direction = "backward")
```

```
## Start:  AIC=1271.98
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + 
##     tax + ptratio + black + lstat
## 
##           Df Sum of Sq     RSS    AIC
## - indus    1      0.47  8694.0 1270.0
## - age      1      3.68  8697.2 1270.2
## <none>                  8693.5 1272.0
## - zn       1    100.16  8793.7 1274.6
## - black    1    133.72  8827.2 1276.2
## - chas     1    179.07  8872.6 1278.2
## - tax      1    192.02  8885.5 1278.8
## - crim     1    346.49  9040.0 1285.8
## - nox      1    349.82  9043.3 1286.0
## - rad      1    353.19  9046.7 1286.1
## - dis      1    895.99  9589.5 1309.8
## - ptratio  1    967.02  9660.6 1312.8
## - lstat    1   1751.70 10445.2 1344.5
## - rm       1   1924.55 10618.1 1351.2
## 
## Step:  AIC=1270
## medv ~ crim + zn + chas + nox + rm + age + dis + rad + tax + 
##     ptratio + black + lstat
## 
##           Df Sum of Sq     RSS    AIC
## - age      1      3.72  8697.7 1268.2
## <none>                  8694.0 1270.0
## - zn       1     99.70  8793.7 1272.6
## - black    1    133.41  8827.4 1274.2
## - chas     1    183.97  8878.0 1276.5
## - tax      1    218.36  8912.4 1278.1
## - crim     1    347.75  9041.8 1283.9
## - nox      1    365.93  9059.9 1284.7
## - rad      1    368.70  9062.7 1284.9
## - dis      1    960.69  9654.7 1310.5
## - ptratio  1    980.38  9674.4 1311.4
## - lstat    1   1766.03 10460.0 1343.1
## - rm       1   1935.33 10629.3 1349.6
## 
## Step:  AIC=1268.17
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + 
##     black + lstat
## 
##           Df Sum of Sq     RSS    AIC
## <none>                  8697.7 1268.2
## - zn       1    107.07  8804.8 1271.1
## - black    1    131.32  8829.0 1272.3
## - chas     1    184.06  8881.8 1274.7
## - tax      1    220.11  8917.8 1276.3
## - crim     1    348.98  9046.7 1282.1
## - rad      1    375.04  9072.8 1283.3
## - nox      1    412.28  9110.0 1285.0
## - ptratio  1    989.00  9686.7 1309.9
## - dis      1   1017.37  9715.1 1311.1
## - rm       1   1989.55 10687.3 1349.8
## - lstat    1   2069.82 10767.5 1352.8
```

```
bwd_model1<-lm(medv~crim+zn+chas+nox+rm+dis+rad+tax+ptratio+black+lstat,data=train)
summary(bwd_model1)
```

```
## 
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad + 
##     tax + ptratio + black + lstat, data = train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -16.011  -2.882  -0.618   2.005  27.248 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  35.205808   5.476369   6.429 3.73e-10 ***
## crim         -0.136355   0.034294  -3.976 8.34e-05 ***
## zn            0.033071   0.015016   2.202  0.02822 *  
## chas          2.810076   0.973183   2.888  0.00410 ** 
## nox         -17.190211   3.977753  -4.322 1.96e-05 ***
## rm            4.089675   0.430791   9.493  < 2e-16 ***
## dis          -1.356644   0.199839  -6.789 4.18e-11 ***
## rad           0.298813   0.072496   4.122 4.59e-05 ***
## tax          -0.012128   0.003841  -3.158  0.00171 ** 
## ptratio      -0.968272   0.144662  -6.693 7.52e-11 ***
## black         0.006995   0.002868   2.439  0.01517 *  
## lstat        -0.496549   0.051280  -9.683  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4.698 on 394 degrees of freedom
## Multiple R-squared:  0.7431, Adjusted R-squared:  0.7359 
## F-statistic: 103.6 on 11 and 394 DF,  p-value: < 2.2e-16
```

```
extractAIC(bwd_model1)
```

```
## [1]   12.000 1268.172
```

```
BIC<-AIC(bwd_model1,k = log(length(bwd_model1)))
BIC
```

```
## [1] 2428.654
```

```
bwd_model2<-lm(medv~crim+zn+chas+nox+rm+dis+rad+tax+ptratio+black+lstat+age,data=train)
summary(bwd_model2)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + black + lstat + age, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.0846  -2.8477  -0.6268   1.9915  27.4597
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.063468   5.493129   6.383 4.89e-10 ***
## crim         -0.136132   0.034335  -3.965 8.73e-05 ***
## zn            0.032216   0.015176   2.123  0.03439 *
## chas          2.809418   0.974213   2.884  0.00415 **
## nox         -16.756436   4.119968  -4.067 5.75e-05 ***
## rm            4.128225   0.441367   9.353  < 2e-16 ***
## dis          -1.382579   0.209803  -6.590 1.42e-10 ***
## rad           0.296891   0.072724   4.082 5.40e-05 ***
## tax          -0.012085   0.003846  -3.142  0.00181 **
## ptratio      -0.965271   0.144999  -6.657 9.42e-11 ***
## black         0.007062   0.002876   2.456  0.01449 *
## lstat        -0.488792   0.054706  -8.935  < 2e-16 ***
## age          -0.005949   0.014503  -0.410  0.68187
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.703 on 393 degrees of freedom
## Multiple R-squared:  0.7432, Adjusted R-squared:  0.7354
## F-statistic: 94.78 on 12 and 393 DF,  p-value: < 2.2e-16
```

```
extractAIC(bwd_model2)
```

```
## [1]   13.000 1269.998
```

```
BIC<-AIC(bwd_model2,k = log(length(bwd_model2)))
BIC
```

```
## [1] 2430.965
```

```
bwd_model3<-lm(medv~crim+zn+chas+nox+rm+dis+rad+tax+ptratio+black+lstat+age+indus,data=train)
summary(bwd_model3)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + black + lstat + age + indus, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.0865  -2.8566  -0.6432   1.9900  27.4513
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.106108   5.507795   6.374 5.18e-10 ***
## crim         -0.135962   0.034398  -3.953 9.17e-05 ***
## zn            0.032381   0.015236   2.125  0.03419 *
## chas          2.792161   0.982623   2.842  0.00472 **
## nox         -16.909701   4.257663  -3.972 8.50e-05 ***
## rm            4.134143   0.443788   9.316  < 2e-16 ***
## dis          -1.375061   0.216335  -6.356 5.75e-10 ***
## rad           0.299541   0.075060   3.991 7.86e-05 ***
## tax          -0.012324   0.004188  -2.942  0.00345 **
## ptratio      -0.968265   0.146633  -6.603 1.31e-10 ***
## black         0.007072   0.002880   2.455  0.01450 *
## lstat        -0.489655   0.055095  -8.887  < 2e-16 ***
## age          -0.005917   0.014523  -0.407  0.68393
## indus         0.009620   0.066167   0.145  0.88447
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.709 on 392 degrees of freedom
## Multiple R-squared:  0.7432, Adjusted R-squared:  0.7347
## F-statistic: 87.28 on 13 and 392 DF,  p-value: < 2.2e-16
```

```
extractAIC(bwd_model3)
```

```
## [1]   14.000 1271.976
```

```
BIC<-AIC(bwd_model3,k = log(length(bwd_model3)))
BIC
```

```
## [1] 2433.428
```

```
# Apply Forward/Backward Selection
fullmodel<- lm(medv ~., data = train)
nullmodel<-lm(medv ~1, data = train)

model_step <- step(fullmodel, scope=list(lower=nullmodel, upper=fullmodel), direction='both')
```

```
## Start:  AIC=1271.98
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + 
##     tax + ptratio + black + lstat
## 
##           Df Sum of Sq     RSS    AIC
## - indus    1      0.47  8694.0 1270.0
## - age      1      3.68  8697.2 1270.2
## <none>                  8693.5 1272.0
## - zn       1    100.16  8793.7 1274.6
## - black    1    133.72  8827.2 1276.2
## - chas     1    179.07  8872.6 1278.2
## - tax      1    192.02  8885.5 1278.8
## - crim     1    346.49  9040.0 1285.8
## - nox      1    349.82  9043.3 1286.0
## - rad      1    353.19  9046.7 1286.1
## - dis      1    895.99  9589.5 1309.8
## - ptratio  1    967.02  9660.6 1312.8
## - lstat    1   1751.70 10445.2 1344.5
## - rm       1   1924.55 10618.1 1351.2
## 
## Step:  AIC=1270
## medv ~ crim + zn + chas + nox + rm + age + dis + rad + tax + 
##     ptratio + black + lstat
## 
##           Df Sum of Sq     RSS    AIC
## - age      1      3.72  8697.7 1268.2
## <none>                  8694.0 1270.0
## + indus    1      0.47  8693.5 1272.0
## - zn       1     99.70  8793.7 1272.6
## - black    1    133.41  8827.4 1274.2
## - chas     1    183.97  8878.0 1276.5
## - tax      1    218.36  8912.4 1278.1
## - crim     1    347.75  9041.8 1283.9
## - nox      1    365.93  9059.9 1284.7
## - rad      1    368.70  9062.7 1284.9
## - dis      1    960.69  9654.7 1310.5
## - ptratio  1    980.38  9674.4 1311.4
## - lstat    1   1766.03 10460.0 1343.1
## - rm       1   1935.33 10629.3 1349.6
## 
## Step:  AIC=1268.17
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + 
##     black + lstat
## 
##           Df Sum of Sq     RSS    AIC
## <none>                  8697.7 1268.2
## + age      1      3.72  8694.0 1270.0
## + indus    1      0.51  8697.2 1270.2
## - zn       1    107.07  8804.8 1271.1
## - black    1    131.32  8829.0 1272.3
## - chas     1    184.06  8881.8 1274.7
## - tax      1    220.11  8917.8 1276.3
## - crim     1    348.98  9046.7 1282.1
## - rad      1    375.04  9072.8 1283.3
## - nox      1    412.28  9110.0 1285.0
## - ptratio  1    989.00  9686.7 1309.9
## - dis      1   1017.37  9715.1 1311.1
## - rm       1   1989.55 10687.3 1349.8
## - lstat    1   2069.82 10767.5 1352.8
```

```
step_model1<-lm(medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + black + lstat,d
ata=train)
summary(step_model1)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + black + lstat, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.011  -2.882  -0.618   2.005  27.248
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.205808   5.476369   6.429 3.73e-10 ***
## crim         -0.136355   0.034294  -3.976 8.34e-05 ***
## zn            0.033071   0.015016   2.202  0.02822 *
## chas          2.810076   0.973183   2.888  0.00410 **
## nox         -17.190211   3.977753  -4.322 1.96e-05 ***
## rm            4.089675   0.430791   9.493  < 2e-16 ***
## dis          -1.356644   0.199839  -6.789 4.18e-11 ***
## rad           0.298813   0.072496   4.122 4.59e-05 ***
## tax          -0.012128   0.003841  -3.158  0.00171 **
## ptratio      -0.968272   0.144662  -6.693 7.52e-11 ***
## black         0.006995   0.002868   2.439  0.01517 *
## lstat        -0.496549   0.051280  -9.683  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.698 on 394 degrees of freedom
## Multiple R-squared:  0.7431, Adjusted R-squared:  0.7359
## F-statistic: 103.6 on 11 and 394 DF,  p-value: < 2.2e-16
```

```
extractAIC(step_model1)
```

```
## [1]   12.000 1268.172
```

```
BIC<-AIC(step_model1,k = log(length(step_model1)))
BIC
```

```
## [1] 2428.654
```

```
step_model2<-lm(medv~crim + zn + chas + nox + rm + age + dis + rad + tax + ptratio + black + lst
at,data=train)
summary(step_model2)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + age + dis +
##     rad + tax + ptratio + black + lstat, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.0846  -2.8477  -0.6268   1.9915  27.4597
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.063468   5.493129   6.383 4.89e-10 ***
## crim         -0.136132   0.034335  -3.965 8.73e-05 ***
## zn            0.032216   0.015176   2.123  0.03439 *
## chas          2.809418   0.974213   2.884  0.00415 **
## nox         -16.756436   4.119968  -4.067 5.75e-05 ***
## rm            4.128225   0.441367   9.353  < 2e-16 ***
## age          -0.005949   0.014503  -0.410  0.68187
## dis          -1.382579   0.209803  -6.590 1.42e-10 ***
## rad           0.296891   0.072724   4.082 5.40e-05 ***
## tax          -0.012085   0.003846  -3.142  0.00181 **
## ptratio      -0.965271   0.144999  -6.657 9.42e-11 ***
## black         0.007062   0.002876   2.456  0.01449 *
## lstat        -0.488792   0.054706  -8.935  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.703 on 393 degrees of freedom
## Multiple R-squared:  0.7432, Adjusted R-squared:  0.7354
## F-statistic: 94.78 on 12 and 393 DF,  p-value: < 2.2e-16
```

```
extractAIC(step_model2)
```

```
## [1]   13.000 1269.998
```

```
BIC<-AIC(step_model2,k = log(length(step_model2)))
BIC
```

```
## [1] 2430.965
```

```
step_model3<-lm(medv~crim+zn+chas+nox+rm+dis+rad+tax+ptratio+black+lstat+age+indus,data=train)
summary(step_model3)
```

```
## 
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad + 
##     tax + ptratio + black + lstat + age + indus, data = train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -16.0865  -2.8566  -0.6432   1.9900  27.4513 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  35.106108   5.507795   6.374 5.18e-10 ***
## crim         -0.135962   0.034398  -3.953 9.17e-05 ***
## zn            0.032381   0.015236   2.125  0.03419 *  
## chas          2.792161   0.982623   2.842  0.00472 ** 
## nox         -16.909701   4.257663  -3.972 8.50e-05 ***
## rm            4.134143   0.443788   9.316  < 2e-16 ***
## dis          -1.375061   0.216335  -6.356 5.75e-10 ***
## rad           0.299541   0.075060   3.991 7.86e-05 ***
## tax          -0.012324   0.004188  -2.942  0.00345 ** 
## ptratio      -0.968265   0.146633  -6.603 1.31e-10 ***
## black         0.007072   0.002880   2.455  0.01450 *  
## lstat        -0.489655   0.055095  -8.887  < 2e-16 ***
## age          -0.005917   0.014523  -0.407  0.68393    
## indus         0.009620   0.066167   0.145  0.88447    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4.709 on 392 degrees of freedom
## Multiple R-squared:  0.7432, Adjusted R-squared:  0.7347 
## F-statistic: 87.28 on 13 and 392 DF,  p-value: < 2.2e-16
```

```
extractAIC(step_model3)
```

```
## [1]   14.000 1271.976
```

```
BIC<-AIC(step_model3,k = log(length(step_model3)))
BIC
```

```
## [1] 2433.428
```

Applying the different models in the table to the test dataset and recording the AdjR^2, AIC and BIC values

Model #1

```
test_model1<-lm(medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + black + lstat,d
ata=test)
summary(test_model1)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + black + lstat, data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.6552  -2.5115  -0.3456   1.7558  21.1772
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  45.592525  15.024417   3.035  0.00317 **
## crim          0.159856   0.109201   1.464  0.14679
## zn            0.098383   0.031427   3.131  0.00237 **
## chas          2.918574   1.830592   1.594  0.11445
## nox         -18.638555   7.741063  -2.408  0.01814 *
## rm            2.297517   1.277328   1.799  0.07550 .
## dis          -2.270725   0.495176  -4.586 1.49e-05 ***
## rad           0.252202   0.140468   1.795  0.07602 .
## tax          -0.008897   0.007314  -1.216  0.22709
## ptratio      -0.884744   0.283290  -3.123  0.00242 **
## black         0.019720   0.007529   2.619  0.01038 *
## lstat        -0.796152   0.134892  -5.902 6.57e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.676 on 88 degrees of freedom
## Multiple R-squared:  0.7825, Adjusted R-squared:  0.7553
## F-statistic: 28.77 on 11 and 88 DF,  p-value: < 2.2e-16
```

```
extractAIC(test_model1)
```

```
## [1]  12.0000 319.7257
```

```
BIC<-AIC(test_model1,k = log(length(test_model1)))
BIC
```

```
## [1] 611.8172
```

Applying the different models in the table to the test dataset and recording the AdjR^2, AIC and BIC values

Model #2

```
test_model2<-lm(medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + black + lstat+a
ge+indus,data=test)
summary(test_model2)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + black + lstat + age + indus, data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.8004  -2.1362  -0.3765   1.8118  20.3313
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  48.505757  15.272954   3.176  0.00207 **
## crim          0.165655   0.109541   1.512  0.13414
## zn            0.107133   0.033078   3.239  0.00171 **
## chas          2.267514   1.911536   1.186  0.23880
## nox         -24.107429   8.894119  -2.710  0.00811 **
## rm            2.144457   1.316060   1.629  0.10687
## dis          -2.107512   0.515816  -4.086 9.83e-05 ***
## rad           0.319734   0.156299   2.046  0.04385 *
## tax          -0.013609   0.009545  -1.426  0.15754
## ptratio      -0.943392   0.288561  -3.269  0.00155 **
## black         0.019355   0.007629   2.537  0.01298 *
## lstat        -0.835637   0.142884  -5.848 8.74e-08 ***
## age           0.032122   0.032251   0.996  0.32205
## indus         0.115501   0.170192   0.679  0.49918
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.687 on 86 degrees of freedom
## Multiple R-squared:  0.7865, Adjusted R-squared:  0.7542
## F-statistic: 24.37 on 13 and 86 DF,  p-value: < 2.2e-16
```

```
extractAIC(test_model2)
```

```
## [1]  14.000 321.863
```

```
AIC(test_model2,k = log(length(test_model2)))
```

```
## [1] 614.9243
```

Model #3

```
test_model3<-lm(medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + black + lstat+a
ge,data=test)
summary(test_model3)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + black + lstat + age, data = test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.726  -2.299  -0.282   1.831  20.393
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  48.329003  15.223318   3.175  0.00207 **
## crim          0.164709   0.109193   1.508  0.13507
## zn            0.100390   0.031453   3.192  0.00197 **
## chas          2.360216   1.900729   1.242  0.21767
## nox         -22.176534   8.400589  -2.640  0.00983 **
## rm            2.026920   1.300563   1.558  0.12275
## dis          -2.120928   0.513837  -4.128 8.38e-05 ***
## rad           0.276207   0.142091   1.944  0.05514 .
## tax          -0.009477   0.007327  -1.293  0.19932
## ptratio      -0.939983   0.287622  -3.268  0.00155 **
## black         0.018836   0.007567   2.489  0.01470 *
## lstat        -0.844073   0.141900  -5.948 5.53e-08 ***
## age           0.034480   0.031964   1.079  0.28369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.672 on 87 degrees of freedom
## Multiple R-squared:  0.7853, Adjusted R-squared:  0.7557
## F-statistic: 26.52 on 12 and 87 DF,  p-value: < 2.2e-16
```

```
extractAIC(test_model3)
```

```
## [1]  13.0000 320.3971
```

```
BIC<-AIC(test_model3,k = log(length(test_model3)))
BIC
```

```
## [1] 612.9735
```