# ISYE 6501 Week 8 Homework

2023-10-10

## Question 11.1

**Using the crime data set uscrime.txt from Questions 8.2, 9.1, and 10.1, build a regression model using: 1) Stepwise regression, 2) Lasso, and 3) Elastic net.**

**1) Stepwise regression**

I fit a linear regression model using all of the data in the crime_data data frame. Then, I used the step() function to perform variable selection on my model. The hyperparameter direction="both" tells the function to perform both forward selection and backward elimination. The resulting model uses the predictors M, Ed, Po1, M.F, U1, U2, Ineq, and Prob.

```
#load the data
crime_data <- read.table("http://www.statsci.org/data/general/uscrime.txt", header=TRUE)

#fit linear regression model
lm_model <- lm(Crime ~ ., data=crime_data)

#perform stepwise regression using forward selection and backward elimination
stepwise_model <- step(lm_model, direction="both")
```

```
## Start:  AIC=514.65
## Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 +
##     U2 + Wealth + Ineq + Prob + Time
##
##          Df Sum of Sq     RSS    AIC
## - So      1        29 1354974 512.65
## - LF      1      8917 1363862 512.96
## - Time    1     10304 1365250 513.00
## - Pop     1     14122 1369068 513.14
## - NW      1     18395 1373341 513.28
## - M.F     1     31967 1386913 513.74
## - Wealth  1     37613 1392558 513.94
## - Po2     1     37919 1392865 513.95
## <none>                1354946 514.65
## - U1      1     83722 1438668 515.47
## - Po1     1    144306 1499252 517.41
## - U2      1    181536 1536482 518.56
## - M       1    193770 1548716 518.93
## - Prob    1    199538 1554484 519.11
## - Ed      1    402117 1757063 524.86
## - Ineq    1    423031 1777977 525.42
##
## Step:  AIC=512.65
```

```
## Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
##     Wealth + Ineq + Prob + Time
##
##          Df Sum of Sq     RSS    AIC
## - Time    1     10341 1365315 511.01
## - LF      1     10878 1365852 511.03
## - Pop     1     14127 1369101 511.14
## - NW      1     21626 1376600 511.39
## - M.F     1     32449 1387423 511.76
## - Po2     1     37954 1392929 511.95
## - Wealth  1     39223 1394197 511.99
## <none>                1354974 512.65
## - U1      1     96420 1451395 513.88
## + So      1        29 1354946 514.65
## - Po1     1    144302 1499277 515.41
## - U2      1    189859 1544834 516.81
## - M       1    195084 1550059 516.97
## - Prob    1    204463 1559437 517.26
## - Ed      1    403140 1758114 522.89
## - Ineq    1    488834 1843808 525.13
##
## Step:  AIC=511.01
## Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
##     Wealth + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - LF      1     10533 1375848 509.37
## - NW      1     15482 1380797 509.54
## - Pop     1     21846 1387161 509.75
## - Po2     1     28932 1394247 509.99
## - Wealth  1     36070 1401385 510.23
## - M.F     1     41784 1407099 510.42
## <none>                1365315 511.01
## - U1      1     91420 1456735 512.05
## + Time    1     10341 1354974 512.65
## + So      1        65 1365250 513.00
## - Po1     1    134137 1499452 513.41
## - U2      1    184143 1549458 514.95
## - M       1    186110 1551425 515.01
## - Prob    1    237493 1602808 516.54
## - Ed      1    409448 1774763 521.33
## - Ineq    1    502909 1868224 523.75
##
## Step:  AIC=509.37
## Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + NW + U1 + U2 + Wealth +
##     Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - NW      1     11675 1387523 507.77
## - Po2     1     21418 1397266 508.09
## - Pop     1     27803 1403651 508.31
## - M.F     1     31252 1407100 508.42
## - Wealth  1     35035 1410883 508.55
## <none>                1375848 509.37
```

```
## - U1      1     80954 1456802 510.06
## + LF      1     10533 1365315 511.01
## + Time    1      9996 1365852 511.03
## + So      1      3046 1372802 511.26
## - Po1     1    123896 1499744 511.42
## - U2      1    190746 1566594 513.47
## - M       1    217716 1593564 514.27
## - Prob    1    226971 1602819 514.54
## - Ed      1    413254 1789103 519.71
## - Ineq    1    500944 1876792 521.96
##
## Step:  AIC=507.77
## Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + U1 + U2 + Wealth + Ineq +
##     Prob
##
##          Df Sum of Sq     RSS    AIC
## - Po2     1     16706 1404229 506.33
## - Pop     1     25793 1413315 506.63
## - M.F     1     26785 1414308 506.66
## - Wealth  1     31551 1419073 506.82
## <none>                1387523 507.77
## - U1      1     83881 1471404 508.52
## + NW      1     11675 1375848 509.37
## + So      1      7207 1380316 509.52
## + LF      1      6726 1380797 509.54
## + Time    1      4534 1382989 509.61
## - Po1     1    118348 1505871 509.61
## - U2      1    201453 1588976 512.14
## - Prob    1    216760 1604282 512.59
## - M       1    309214 1696737 515.22
## - Ed      1    402754 1790276 517.74
## - Ineq    1    589736 1977259 522.41
##
## Step:  AIC=506.33
## Crime ~ M + Ed + Po1 + M.F + Pop + U1 + U2 + Wealth + Ineq +
##     Prob
##
##          Df Sum of Sq     RSS    AIC
## - Pop     1     22345 1426575 505.07
## - Wealth  1     32142 1436371 505.39
## - M.F     1     36808 1441037 505.54
## <none>                1404229 506.33
## - U1      1     86373 1490602 507.13
## + Po2     1     16706 1387523 507.77
## + NW      1      6963 1397266 508.09
## + So      1      3807 1400422 508.20
## + LF      1      1986 1402243 508.26
## + Time    1       575 1403654 508.31
## - U2      1    205814 1610043 510.76
## - Prob    1    218607 1622836 511.13
## - M       1    307001 1711230 513.62
## - Ed      1    389502 1793731 515.83
## - Ineq    1    608627 2012856 521.25
## - Po1     1   1050202 2454432 530.57
```

```
##
## Step:  AIC=505.07
## Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Wealth + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - Wealth  1     26493 1453068 503.93
## <none>                1426575 505.07
## - M.F     1     84491 1511065 505.77
## - U1      1     99463 1526037 506.24
## + Pop     1     22345 1404229 506.33
## + Po2     1     13259 1413315 506.63
## + NW      1      5927 1420648 506.87
## + So      1      5724 1420851 506.88
## + LF      1      5176 1421398 506.90
## + Time    1      3913 1422661 506.94
## - Prob    1    198571 1625145 509.20
## - U2      1    208880 1635455 509.49
## - M       1    320926 1747501 512.61
## - Ed      1    386773 1813348 514.35
## - Ineq    1    594779 2021354 519.45
## - Po1     1   1127277 2553852 530.44
##
## Step:  AIC=503.93
## Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## <none>                1453068 503.93
## + Wealth  1     26493 1426575 505.07
## - M.F     1    103159 1556227 505.16
## + Pop     1     16697 1436371 505.39
## + Po2     1     14148 1438919 505.47
## + So      1      9329 1443739 505.63
## + LF      1      4374 1448694 505.79
## + NW      1      3799 1449269 505.81
## + Time    1      2293 1450775 505.86
## - U1      1    127044 1580112 505.87
## - Prob    1    247978 1701046 509.34
## - U2      1    255443 1708511 509.55
## - M       1    296790 1749858 510.67
## - Ed      1    445788 1898855 514.51
## - Ineq    1    738244 2191312 521.24
## - Po1     1   1672038 3125105 537.93
```

To evaluate how "good" my model is, I calculated its r-squared value. The model has an r-squared value of 78.9%, indicating that it accounts for a majority of the model's variability.

```r
#make predictions
prediction <- predict(stepwise_model, newdata=crime_data)

#calculate R2
SSR <- sum((prediction - crime_data$Crime)^2)
SST <- sum((crime_data$Crime - mean(crime_data$Crime))^2)
R2 <- 1 - SSR/SST
R2
```

```
## [1] 0.7888268
```

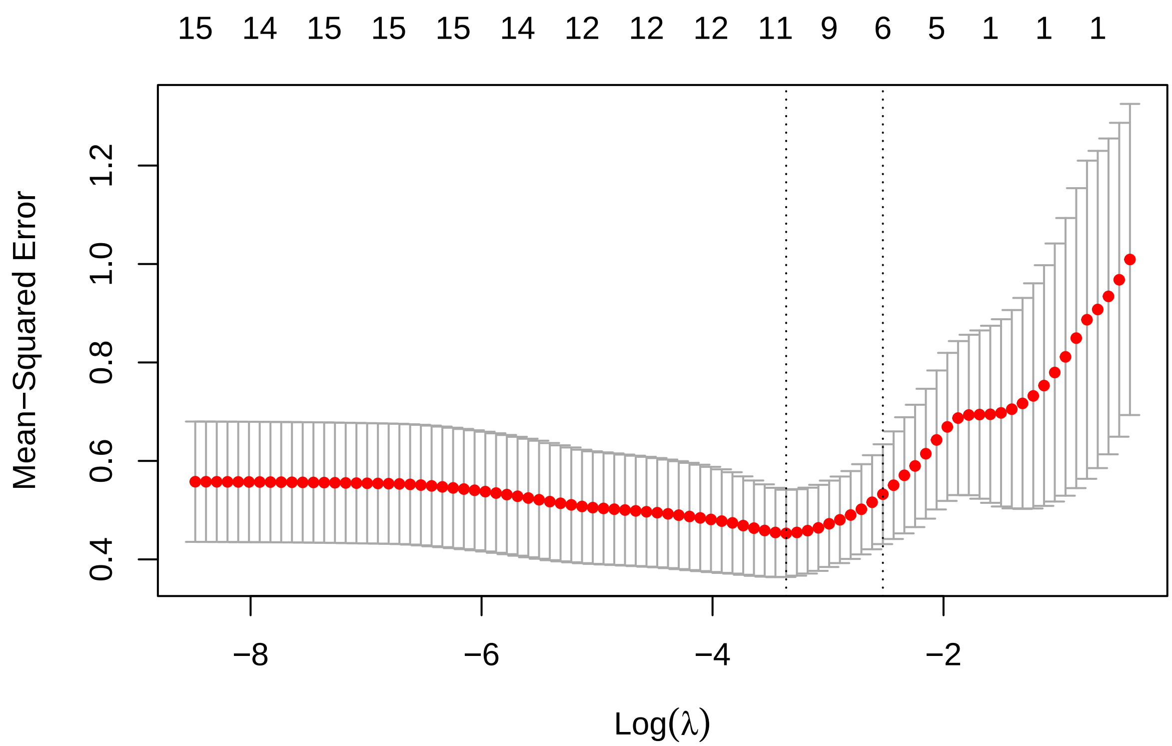
### 2) Lasso

Before fitting a lasso regression model, I reformatted my predictors into a matrix and scaled the data to ensure the lasso constraint will have the desired effect. Then, I used the cv.glmnet() function to find the best value of lambda. Lambda is the regularization parameter that trades off between model complexity and accuracy.

```
#create a matrix of predictors
scaled_crime_data <- scale(crime_data)
predictors <- as.matrix(scaled_crime_data[,1:ncol(scaled_crime_data)-1])
response <- scaled_crime_data[,ncol(scaled_crime_data)]

#use cv to find best parameter of lambda
cv_result <- cv.glmnet(predictors, response, alpha=1)

#plot results
plot(cv_result)
```




```
#get best lambda
best_lambda <- cv_result$lambda.min
best_lambda
```

```
## [1] 0.03465288
```

I also extracted the cv model's coefficients. As you can see below, M, Po1, M.F, Ineq, and Prob were selected. When compared to the stepwise model above, the lasso model uses fewer predictors.

```
#extract model coefficients
lasso_coeff <- coef(cv_result)
lasso_coeff
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept) -3.821380e-16
## M            1.157874e-01
## So           .
## Ed           1.884051e-04
## Po1          7.311501e-01
## Po2          .
## LF           .
## M.F          1.403030e-01
## Pop          .
## NW           .
## U1           .
## U2           .
## Wealth       .
## Ineq         2.130764e-01
## Prob        -1.342423e-01
## Time         .
```

Next, I fit a lasso model using the optimal value of lambda.

```
#fit LASSO regression model
lasso_model <- glmnet(predictors, response, alpha=1, lambda=best_lambda)
summary(lasso_model)
```

```
##           Length Class     Mode
## a0         1     -none-    numeric
## beta      15     dgCMatrix S4
## df         1     -none-    numeric
## dim        2     -none-    numeric
## lambda     1     -none-    numeric
## dev.ratio  1     -none-    numeric
## nulldev    1     -none-    numeric
## npasses    1     -none-    numeric
## jerr       1     -none-    numeric
## offset     1     -none-    logical
## call       5     -none-    call
## nobs       1     -none-    numeric
```

To evaluate my model's fit, I calculated its r-squared value. This model has a slightly lower r-squared than the stepwise model.

```
#make predictions
prediction <- predict(lasso_model, newx=predictors)
```

```r
#calculate R2
SSR <- sum((prediction - response)^2)
SST <- sum((response - mean(response))^2)
R2 <- 1 - SSR/SST
R2
```

```
## [1] 0.7515535
```

### 3) Elastic Net

Elastic net uses a combination of lasso and ridge regression. Thus, it's alpha parameter will be somewhere between 0 (ridge) and 1 (lasso). For this model, I followed the same steps as above, but I used a for loop to test out the model with different parameters of alpha.

As you can see in the code block below, r-squared is highest when alpha = 0.4. This means the model relies more heavily on ridge regression than on lasso.

```r
for(a in seq(0.1, 0.9, by=0.1)){

  #use cv to find best parameter of lambda
  cv_result <- cv.glmnet(predictors, response, alpha=a)

  #get best lambda
  best_lambda <- cv_result$lambda.min

  #fit elastic net model with optimal lambda and chosen alpha
  elastic_net_model <- glmnet(predictors, response, alpha=a, lambda = best_lambda)
  summary(elastic_net_model)

  #make predictions
  prediction <- predict(elastic_net_model, newx=predictors)
  prediction

  #calculate R2
  SSR <- sum((prediction - response)^2)
  SST <- sum((response - mean(response))^2)
  R2 <- 1 - SSR/SST
  print(paste0("alpha = ", a, " :", round(R2, 3)))
}
```

```
## [1] "alpha = 0.1 :0.768"
## [1] "alpha = 0.2 :0.773"
## [1] "alpha = 0.3 :0.783"
## [1] "alpha = 0.4 :0.795"
## [1] "alpha = 0.5 :0.749"
## [1] "alpha = 0.6 :0.774"
## [1] "alpha = 0.7 :0.79"
## [1] "alpha = 0.8 :0.778"
## [1] "alpha = 0.9 :0.743"
```