Armando Tejeda Homework 8 10/17/22

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
3. Elastic net For Parts 2 and 3, remember to scale the data first – otherwise, the regression coefficients will be on different scales and the constraint won't have the desired effect.

For Parts 2 and 3, use the glmnet function in R.

Notes on R: • For the elastic net model, what we called λ in the videos, glmnet calls "alpha"; you can get a range of results by varying alpha from 1 (lasso) to 0 (ridge regression) [and, of course, other values of alpha in between]. • In a function call like glmnet(x,y,family="mgaussian",alpha=1) the predictors x need to be in R's matrix format, rather than data frame format. You can convert a data frame to a matrix using as.matrix – for example, x <- as.matrix(data[,1:n-1]) • Rather than specifying a value of T, glmnet returns models for a variety of values of T.

In [ ]:
```
rm(list = ls())
set.seed(1)

setwd("C:/Users/ateje/OneDrive/Desktop/VS Code Projects/GTx_MM_in_Analytics/R_projects

usCrime_data = read.table("uscrime.txt", header = TRUE)
```

Running a quick linear model to get a baseline model for this dataset as we have done before.

In [ ]:
```
usCrime_lm = lm(Crime~., data = usCrime_data)
summary(usCrime_lm,)
```

```
Call:
lm(formula = Crime ~ ., data = usCrime_data)

Residuals:
    Min      1Q  Median      3Q     Max
-395.74  -98.09   -6.69  112.99  512.67

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
M            8.783e+01  4.171e+01   2.106 0.043443 *
So          -3.803e+00  1.488e+02  -0.026 0.979765
Ed           1.883e+02  6.209e+01   3.033 0.004861 **
Po1          1.928e+02  1.061e+02   1.817 0.078892 .
Po2         -1.094e+02  1.175e+02  -0.931 0.358830
LF          -6.638e+02  1.470e+03  -0.452 0.654654
M.F          1.741e+01  2.035e+01   0.855 0.398995
Pop         -7.330e-01  1.290e+00  -0.568 0.573845
NW           4.204e+00  6.481e+00   0.649 0.521279
U1          -5.827e+03  4.210e+03  -1.384 0.176238
U2           1.678e+02  8.234e+01   2.038 0.050161 .
Wealth       9.617e-02  1.037e-01   0.928 0.360754
Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
Time        -3.479e+00  7.165e+00  -0.486 0.630708
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 209.1 on 31 degrees of freedom
Multiple R-squared:  0.8031,    Adjusted R-squared:  0.7078
F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

Baseline linear regression with R2 and adjusted R2 of 0.8031 and 0.7078 respectively. This is very good but from previous homeworks we have seen that this due to overfitting, due to the small ratio of data points to predictors.

In [ ]:
```r
library(caret)

# Doing stepwise regression.

stepwise_usCrime = train(Crime ~., data = usCrime_data,
                     method = "lmStepAIC",
                     trControl = trainControl(),
                     trace = FALSE
)

summary(stepwise_usCrime)
```

```
Call:
lm(formula = .outcome ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + 
    Prob, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-444.70 -111.07    3.03  122.15  483.30 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -6426.10    1194.61  -5.379 4.04e-06 ***
M              93.32      33.50   2.786  0.00828 ** 
Ed            180.12      52.75   3.414  0.00153 ** 
Po1           102.65      15.52   6.613 8.26e-08 ***
M.F            22.34      13.60   1.642  0.10874    
U1          -6086.63    3339.27  -1.823  0.07622 .  
U2            187.35      72.48   2.585  0.01371 *  
Ineq           61.33      13.96   4.394 8.63e-05 ***
Prob        -3796.03    1490.65  -2.547  0.01505 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 195.5 on 38 degrees of freedom
Multiple R-squared:  0.7888,    Adjusted R-squared:  0.7444 
F-statistic: 17.74 on 8 and 38 DF,  p-value: 1.159e-10
```

Stepwise regression gives R2 = 0.788 and adjusted R2 = 0.744. A little worse than a simple lm regression but likely a better model with less overfitting.

LASSO Regression next:

In [ ]:
```
# Scale the data, except for col2 and col16

scaled_usCrime_data = as.data.frame(scale(usCrime_data[,c(1,3:15)]))
scaled_usCrime_data = cbind(usCrime_data[,2],scaled_usCrime_data,usCrime_data[,16])
colnames(scaled_usCrime_data)[1] <- "So"
colnames(scaled_usCrime_data)[16] <- "Crime"
```

In [ ]:
```
library(glmnet)

# Lasso Regression.

LASSO_usCrime = cv.glmnet(x=as.matrix(scaled_usCrime_data[,-16]), y=as.matrix(scaled_u
                    alpha=1, nfolds = 5, type.measure="mse", family="gaussian")


LASSO_usCrime$lambda.min
```

11.12694331514

In [ ]:
```
coefficients(LASSO_usCrime, LASSO_usCrime$lambda.min)
```

```
16 x 1 sparse Matrix of class "dgCMatrix"
                    s1
(Intercept) 889.059998
So           47.073756
M            85.028207
Ed          124.299584
Po1         308.896175
Po2           .
LF            1.112481
M.F          52.034368
Pop           .
NW            4.437239
U1          -22.252726
U2           55.983894
Wealth        .
Ineq        180.589656
Prob        -81.811935
Time          .
```

In [ ]:
```
# Now that we have the coefficients for the best lambda value, we run a linear model.

LASSO_usCrime_lm = lm(Crime ~M+So+Ed+Po1+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob,
                      data = scaled_usCrime_data)

summary(LASSO_usCrime_lm)
```

```
Call:
lm(formula = Crime ~ M + So + Ed + Po1 + M.F + Pop + NW + U1 +
    U2 + Wealth + Ineq + Prob, data = scaled_usCrime_data)

Residuals:
    Min      1Q  Median      3Q     Max
-434.18 -107.01   18.55  115.88  470.32

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   897.29      51.91  17.286  < 2e-16 ***
M             112.71      49.35   2.284  0.02876 *
So             22.89     125.35   0.183  0.85621
Ed            195.70      62.94   3.109  0.00378 **
Po1           293.18      64.99   4.511 7.32e-05 ***
M.F            48.92      48.12   1.017  0.31656
Pop           -33.25      45.63  -0.729  0.47113
NW             19.16      57.71   0.332  0.74195
U1            -89.76      65.68  -1.367  0.18069
U2            140.78      66.77   2.108  0.04245 *
Wealth         83.30      95.53   0.872  0.38932
Ineq          285.77      85.19   3.355  0.00196 **
Prob          -92.75      41.12  -2.255  0.03065 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 202.6 on 34 degrees of freedom
Multiple R-squared:  0.7971,    Adjusted R-squared:  0.7255
F-statistic: 11.13 on 12 and 34 DF,  p-value: 1.52e-08
```

In [ ]:
```
# Further remove factors with p-values over 0.05 and run lm again.

LASSO_usCrime_lm_final = lm(Crime ~M+ Ed+ Po1+ U2+ Ineq+Prob,
                      data = scaled_usCrime_data)
```

```
summary(LASSO_usCrime_lm_final)
```

```
Call:
lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = scaled_usCrime_data)

Residuals:
    Min      1Q  Median      3Q     Max
-470.68  -78.41  -19.68  133.12  556.23

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   905.09      29.27  30.918  < 2e-16 ***
M             131.98      41.85   3.154  0.00305 **
Ed            219.79      50.07   4.390 8.07e-05 ***
Po1           341.84      40.87   8.363 2.56e-10 ***
U2             75.47      34.55   2.185  0.03483 *
Ineq          269.91      55.60   4.855 1.88e-05 ***
Prob          -86.44      34.74  -2.488  0.01711 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 200.7 on 40 degrees of freedom
Multiple R-squared:  0.7659,    Adjusted R-squared:  0.7307
F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11
```

Final results for Lasso regression are seemingly very good. Down to 6 factors with an R2 = 0.7659 and Adjusted_R2 = 0.7307.

Elastic Net next:

In [ ]:
```
# Elastic net regression is one that I dont understand very well but I came up with th
# after watching a couple of online tutorials.


# blank list
list = c()

for (i in 0:10) {
    usCrime_elastic <- cv.glmnet(x=as.matrix(scaled_usCrime_data[,-16]),y=as.matrix(sc
                                 alpha=i/10, nfolds = 5, type.measure="mse",
                                 family="gaussian")

    list = cbind(list, usCrime_elastic$glmnet.fit$dev.ratio[which(usCrime_elastic$glmr

}

# get best lambda (alpha in this equation)

lambda = (which.max(list)-1)/10

# get elastic net model
usCrime_elastic = cv.glmnet(x=as.matrix(scaled_usCrime_data[,-16]),
                            y=as.matrix(scaled_usCrime_data$Crime), alpha=0.05,
                            nfolds = 5, type.measure="mse", family="gaussian")


# linear regression
usCrime_elastic_lm = lm(Crime ~M+So+Ed+Po1+Po2+LF+M.F+NW+U1+U2+Wealth+Ineq+Prob
```

```
+Time, data = scaled_usCrime_data)

summary(usCrime_elastic_lm)
```

```
Call:
lm(formula = Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + NW +
    U1 + U2 + Wealth + Ineq + Prob + Time, data = scaled_usCrime_data)

Residuals:
    Min      1Q  Median      3Q     Max
-380.91 -101.89  -14.77  110.87  505.40

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  906.483     58.484  15.500  < 2e-16 ***
M            112.837     51.691   2.183  0.03649 *
So            -4.105    147.172  -0.028  0.97792
Ed           211.246     68.713   3.074  0.00429 **
Po1          563.337    311.541   1.808  0.07998 .
Po2         -313.824    324.701  -0.966  0.34104
LF           -31.702     58.147  -0.545  0.58939
M.F           64.479     54.722   1.178  0.24737
NW            44.572     65.892   0.676  0.50362
U1          -112.728     73.902  -1.525  0.13699
U2           143.186     68.749   2.083  0.04535 *
Wealth        87.836     98.588   0.891  0.37961
Ineq         269.086     86.824   3.099  0.00403 **
Prob        -110.457     51.117  -2.161  0.03830 *
Time         -31.582     48.772  -0.648  0.52189
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 206.8 on 32 degrees of freedom
Multiple R-squared:  0.801,     Adjusted R-squared:  0.714
F-statistic: 9.202 on 14 and 32 DF,  p-value: 1.301e-07
```

I am pretty certain this is not fully correct but is the best I could come up with for elastic net regression. R2 = 0.80 abd adjusted_R2 = 0.714.