WK5_HW5

Anon Skywalker

9/21/2020

Question 8.1

Working as a Clinical Data Analyst at a hospital, COVID-19 cases detected in our emergency department would benefit from using linear regression. Using linear regression could help predict and map the trend of cases as we have been seeing over the last many months in our ED and help us plan for staffing. some predictors that might help in plotting this out are day of the week (people tend to get tested on weekends), population density (of the county the hospital is in), tests performed in a given day, test type, and state cases (people tend to get tested as severity of cases in the state rises).

Question 8.2

To begin with this question, we must first get a better understanding of the data that we are using. The resource: http://www.statsci.org/data/general/uscrime.html (http://www.statsci.org/data/general/uscrime.html). Here we can see the predictors and get a better understanding as to how they impact the crime rate in the US. Additionally, this resource actually provides us with some analysis already done at the bottom of the report. The important takeaways that I have factored into this report are that:

1. Only one of Po1 and Po2, and only one of U1 and U2, remain in the final regression, because of high collinearity.
2. Crime is negatively associated with probability of imprisonment.

Our task is to create a linear regression model, which is a statistical model that will analyze the relationship between a response variable (Y) and one or more independent variables and their effect on the dependent/response variable. The below code section involves the breakdown of applying the built in lm() function in R to build a default regression model that fits the data given ALL predictors as a *baseline. The end of the report will involve using the improved model*.

```
lm.default <- lm(Crime~., data = crime.data)
summary(lm.default)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = crime.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
```

We can then plug the model into the built in AIC() function in R of which provides us with the “goodness of fit” for our model.

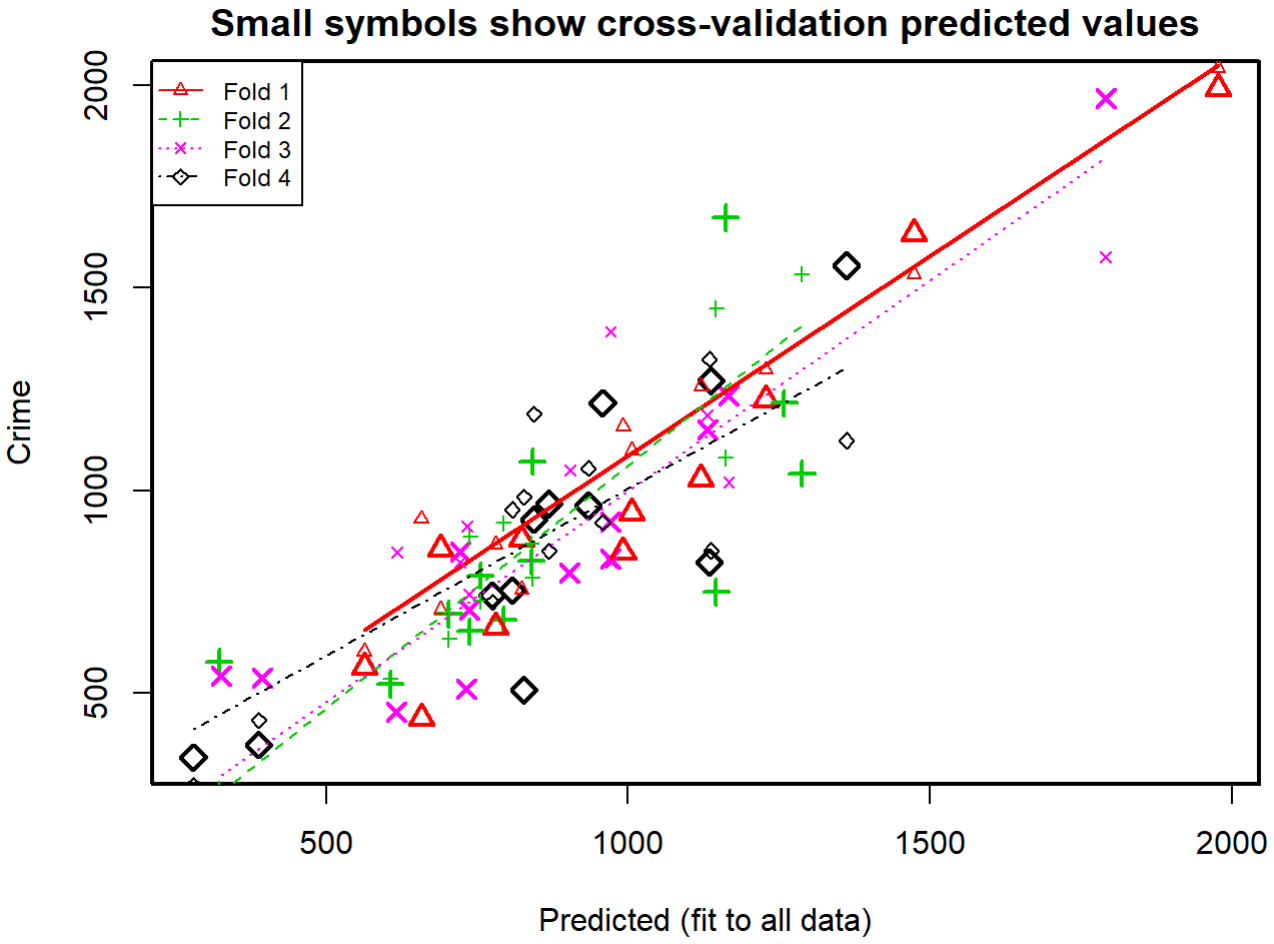
```
#Lower value the better
AIC(lm.default)
```

```
## [1] 650.0291
```

To go even further, we can use cross validation and a manual r-squared calculation to get how close our data points are to the fitted line for the model. As the given adjustedUsing four folds in the cv.lm() function, this came out to 0.353 in which a higher value is better (> 0.70 is excellent and values can be between 0 and 1).

```
lm.default.model <- cv.lm(crime.data, lm.default, m = 4)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## M          1   55084   55084    1.26  0.2702
## So         1   15370   15370    0.35  0.5575
## Ed         1  905668  905668   20.72 7.7e-05 ***
## Po1        1 3076033 3076033   70.38 1.8e-09 ***
## Po2        1  153024  153024    3.50  0.0708 .
## LF         1   61134   61134    1.40  0.2459
## M.F        1  111000  111000    2.54  0.1212
## Pop        1   42649   42649    0.98  0.3309
## NW         1   14197   14197    0.32  0.5728
## U1         1    7065    7065    0.16  0.6904
## U2         1  269663  269663    6.17  0.0186 *
## Wealth     1   34748   34748    0.79  0.3795
## Ineq       1  547423  547423   12.52  0.0013 **
## Prob       1  222620  222620    5.09  0.0312 *
## Time       1   10304   10304    0.24  0.6307
## Residuals 31 1354946   43708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##
## fold 1
## Observations in test set: 11
##                2   9   14   16     20   22     26    38  41   44   47
## Predicted   1474 689  780 1006 1227.8  657 1977.4 562.7 824 1121  992
## cvpred      1535 706  867 1100 1298.9  931 2043.3 602.8 757 1257 1159
## Crime       1635 856  664  946 1225.0  439 1993.0 566.0 880 1030  849
## CV residual  100 150 -203 -154  -73.9 -492  -50.3 -36.8 123 -227 -310
##
## Sum of squares = 512057    Mean square = 46551    n = 11
##
## fold 2
## Observations in test set: 12
##                 1   3    6   11   19    25      28   29  30   33   35    39
## Predicted   755.0 322  793 1161 1146 605.9 1258.48 1287 703  841  738 839.3
## cvpred      727.7 265  920 1082 1449 535.1 1219.78 1534 634  784  886 868.7
## Crime       791.0 578  682 1674  750 523.0 1216.00 1043 696 1072  653 826.0
## CV residual  63.3 313 -238  592 -699 -12.1   -3.78 -491  62  288 -233 -42.7
##
## Sum of squares = 1382466    Mean square = 115205    n = 12
##
## fold 3
## Observations in test set: 12
##                4    5    10  12   13   15  17    34   37     40    42   45
## Predicted   1791 1167 736.5 722  733  903 393 971.5  971 1131.5 326.3  617
## cvpred      1576 1021 745.1 824  912 1050 103 823.4 1392 1186.8 -85.5  848
## Crime       1969 1234 705.0 849  511  798 539 923.0  831 1151.0 542.0  455
## CV residual  393  213 -40.1  25 -401 -252 436  99.6 -561  -35.8 627.5 -393
##
## Sum of squares = 1491541    Mean square = 124295    n = 12
##
## fold 4
## Observations in test set: 12
##                  7    8   18    21   23  24    27    31   32   36   43   46
## Predicted    934.2 1362  844 774.9  958 869 279.5 388.0  808 1138 1134  827
## cvpred      1055.1 1123 1189 725.3  922 851 272.7 433.1  953  852 1324  984
## Crime        963.0 1555  929 742.0 1216 968 342.0 373.0  754 1272  823  508
## CV residual  -92.1  432 -260  16.7  294 117  69.3 -60.1 -199  420 -501 -476
##
## Sum of squares = 1065774    Mean square = 88814    n = 12
##
## Overall (Sum over all 12 folds)
##    ms
## 94720
```

```
#94720 derived from the cv.lm() model return
sse <- 94720*nrow(crime.data)
sst<- sum((crime.data$Crime - mean(crime.data$Crime))^2)
r.squared <- 1 - sse/sst
r.squared
```

```
## [1] 0.353
```

Coefficients for the DEFAULT model:

```
lm.default$coefficients
```

```
## (Intercept)           M          So          Ed         Po1         Po2
##   -5.98e+03    8.78e+01   -3.80e+00    1.88e+02    1.93e+02   -1.09e+02
##          LF         M.F         Pop          NW          U1          U2
##   -6.64e+02    1.74e+01   -7.33e-01    4.20e+00   -5.83e+03    1.68e+02
##      Wealth        Ineq        Prob        Time
##    9.62e-02    7.07e+01   -4.86e+03   -3.48e+00
```

We can then use our default model to predict using the predict() function for use with our test.data point to get an idea of the estimated crime using our default model. This provides us with a value of 155 for crime, which would be the smallest crime value recorded when assessing the current crime data where 342 is the lowest crime value. This tells us that we are likely running into problems with our model while using it to predict the test point, where the number of predictors is likely the factor throwing off our model (Overfitting).

```
#test point from the hw
test.data <- data.frame(M = 14.0, So = 0, Ed = 10.0, Po1 = 12.0, Po2 = 15.5,
                        LF = 0.640, M.F = 94.0, Pop = 150, NW = 1.1, U1 = 0.120,
                        U2 = 3.6, Wealth = 3200, Ineq = 20.1, Prob = 0.040, Time = 39.0)

predict(lm.default, test.data)
```

```
##   1
## 155
```

Next, we want to improve on this model. We can do this using the understanding that a low p-value indicates a high significance for a given predictor variable. We can then remove the predictor variables that are not significant to crime based on this methodology. After removing a few of the variables, I arrived at using M, Ed, Po1, U2, Prob, and Ineq for my model.

```
lm.improv <- lm(Crime~ M+Ed+Po1+U2+Ineq+Prob, data = crime.data)
summary(lm.improv)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -470.7  -78.4  -19.7  133.1  556.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5040.5      899.8   -5.60  1.7e-06 ***
## M              105.0       33.3    3.15   0.0031 **
## Ed             196.5       44.8    4.39  8.1e-05 ***
## Po1            115.0       13.8    8.36  2.6e-10 ***
## U2              89.4       40.9    2.18   0.0348 *
## Ineq            67.7       13.9    4.85  1.9e-05 ***
## Prob         -3801.8     1528.1   -2.49   0.0171 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201 on 40 degrees of freedom
## Multiple R-squared:  0.766,  Adjusted R-squared:  0.731
## F-statistic: 21.8 on 6 and 40 DF,  p-value: 3.42e-11
```

This provided a AIC of 640, 10 lower than the default lm.

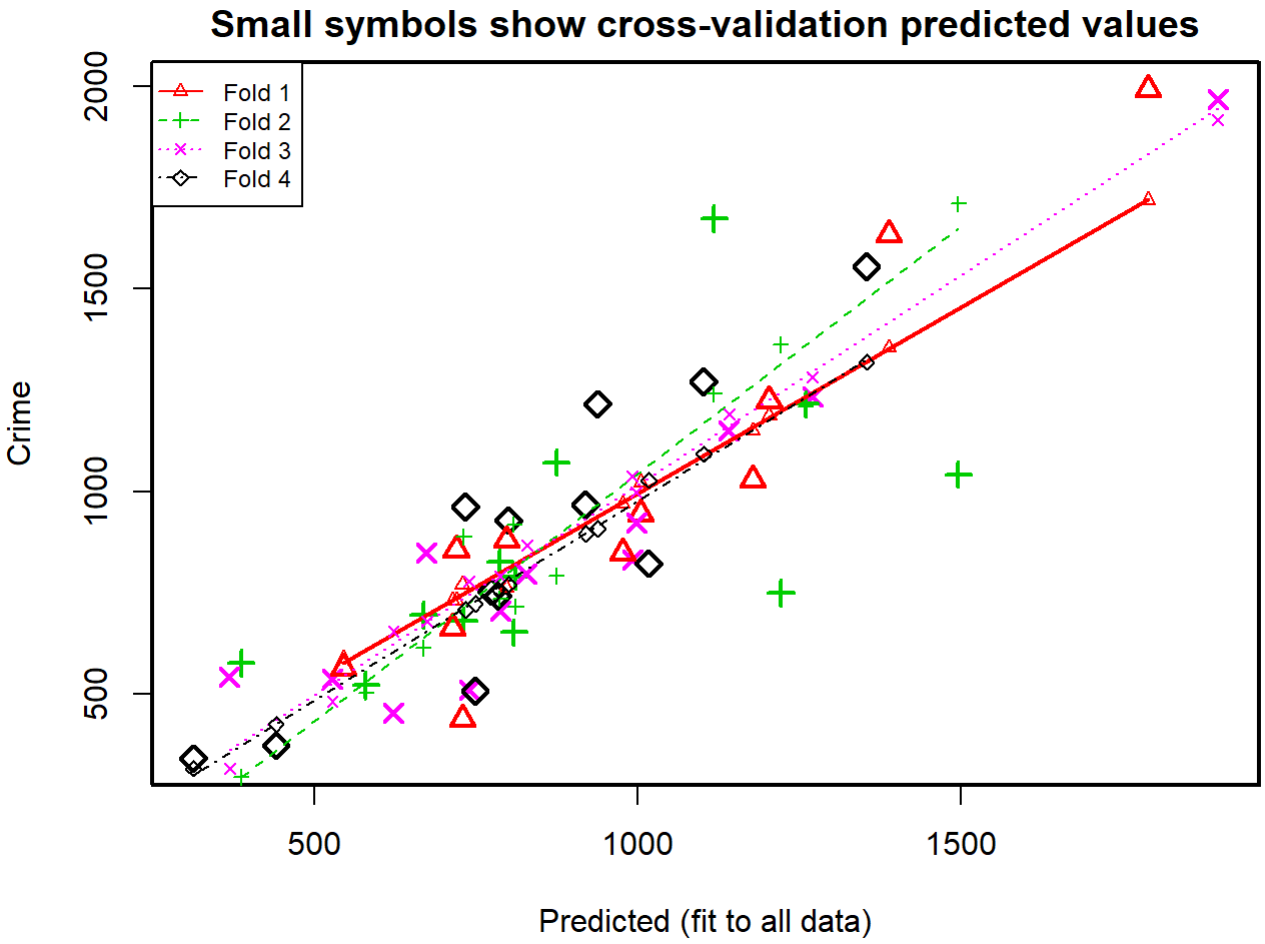
```
AIC(lm.improv)
```

```
## [1] 640
```

Using a similar method, we can use cross validation using the cv.calculate for the manual r-squared. This r-squared value came out to be 0.671, beating our default model. We can also take a closer look at our fitted lines in the plot displayed below, indicating a better fit than the previous plot for our default model.

```
lm.improv.model <- cv.lm(crime.data, lm.improv, m = 4)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## M          1   55084   55084    1.37 0.24914
## Ed         1  725967  725967   18.02 0.00013 ***
## Po1        1 3173852 3173852   78.80 5.3e-11 ***
## U2         1  217386  217386    5.40 0.02534 *
## Ineq       1  848273  848273   21.06 4.3e-05 ***
## Prob       1  249308  249308    6.19 0.01711 *
## Residuals 40 1611057   40276
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Small symbols show cross-validation predicted values
Fold 1
Fold 2
Fold 3
Fold 4
Crime
2000
1500
1000
500
500
1000
1500
Predicted (fit to all data)

```
##
## fold 1
## Observations in test set: 11
##                2   9    14     16     20   22   26    38  41   44   47
## Predicted   1388 719 713.6 1004.4 1203.0  728 1789 544.4 796 1178  976
## cvpred      1355 731 731.1 1023.2 1187.6  771 1720 588.4 763 1150  970
## Crime       1635 856 664.0  946.0 1225.0  439 1993 566.0 880 1030  849
## CV residual  280 125 -67.1  -77.2   37.4 -332  273 -22.4 117 -120 -121
##
## Sum of squares = 334042    Mean square = 30367    n = 11
##
## fold 2
## Observations in test set: 12
##                 1   3    6   11   19    25     28   29    30   33   35    39
## Predicted   810.8 386  730 1118 1221 579.1 1259.0 1495 668.0  874  808 786.7
## cvpred      716.9 296  888 1241 1363 504.3 1208.7 1711 614.2  792  919 736.6
## Crime       791.0 578  682 1674  750 523.0 1216.0 1043 696.0 1072  653 826.0
## CV residual  74.1 282 -206  433 -613  18.7    7.3 -668  81.8  280 -266  89.4
##
## Sum of squares = 1300449    Mean square = 108371    n = 12
##
## fold 3
## Observations in test set: 12
##                  4      5    10  12   13  15    17    34   37     40  42   45
## Predicted   1897.2 1269.8 787.3 673  739 828 527.4 997.5  992 1140.8 369  622
## cvpred      1916.6 1282.8 791.8 680  778 867 483.3 998.2 1037 1190.7 317  656
## Crime       1969.0 1234.0 705.0 849  511 798 539.0 923.0  831 1151.0 542  455
## CV residual   52.4  -48.8 -86.8 169 -267 -69  55.7 -75.2 -206  -39.7 225 -201
##
## Sum of squares = 261503    Mean square = 21792    n = 12
##
## fold 4
## Observations in test set: 12
##                7    8  18  21   23    24    27  31  32   36   43   46
## Predicted   733 1354 800 783  938 919.4 312.2 440 774 1102 1017  748
## cvpred      708 1319 771 759  909 896.3 316.2 426 740 1093 1027  723
## Crime       963 1555 929 742 1216 968.0 342.0 373 754 1272  823  508
## CV residual 255  236 158 -17  307  71.7  25.8 -53  14  179 -204 -215
##
## Sum of squares = 369549    Mean square = 30796    n = 12
##
## Overall (Sum over all 12 folds)
##    ms
## 48203
```

```
#48203 derived from the cv.lm() model return
sse <- 48203*nrow(crime.data)
sst <- sum((crime.data$Crime - mean(crime.data$Crime))^2)
r.squared <- 1 - sse/sst
r.squared
```

```
## [1] 0.671
```

Here are the coefficients for the improved model:

```
lm.improv$coefficients
```

```
## (Intercept)           M          Ed         Po1          U2        Ineq
##     -5040.5       105.0       196.5       115.0        89.4        67.7
##        Prob
##     -3801.8
```

The prediction for this model with the test point provided a crime value of 1304, which seems to be more accurate considering the range of the data set for the crime column (highest in the data set for crime: 1993). We can conclude that this model better fits the data due to our superior value for AIC and r-squared as well as the more reasonable prediction for our test.data point.

```
predict(lm.improv, test.data)
```

```
##    1
## 1304
```