DataChallenge2 | Mario Saraiva (mhs2195)

Mario Saraiva (mhs2195)

3/20/2018

Contents

GR5069 - TOPICS IN APPLIED DATA SCIENCE FOR SOCIAL SCIENTISTS - Spring 2018

March 20, 2018

```
options(java.parameters = "-Xmx5g")
```

Data Challenge 2

Q.1

`Is there an association between Poverty/development levels, Lethality index, and number of deaths per conflict? I hypothesis that there is a difference between Perfect Lethality rate between poor and non-poor states in Mexico.`

Lasso

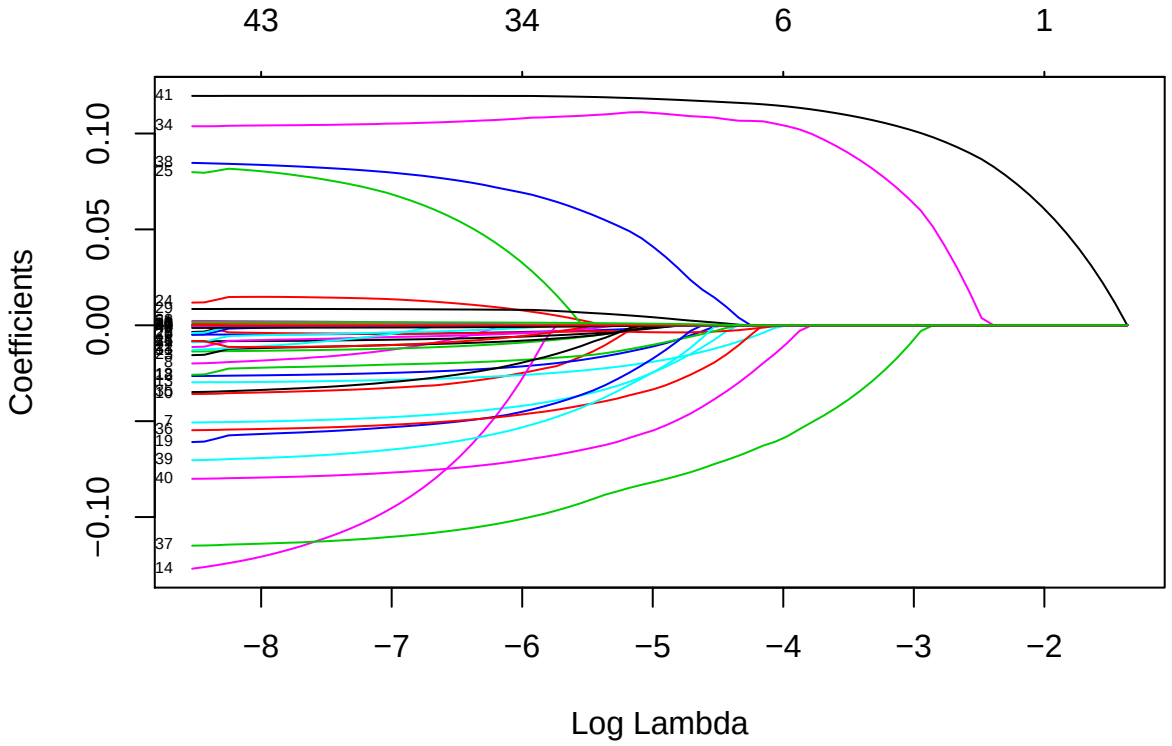
Lasso regression is run on perfect_lethality against all numeric variables in our dataset:

Interpreting the Lasso Graph:

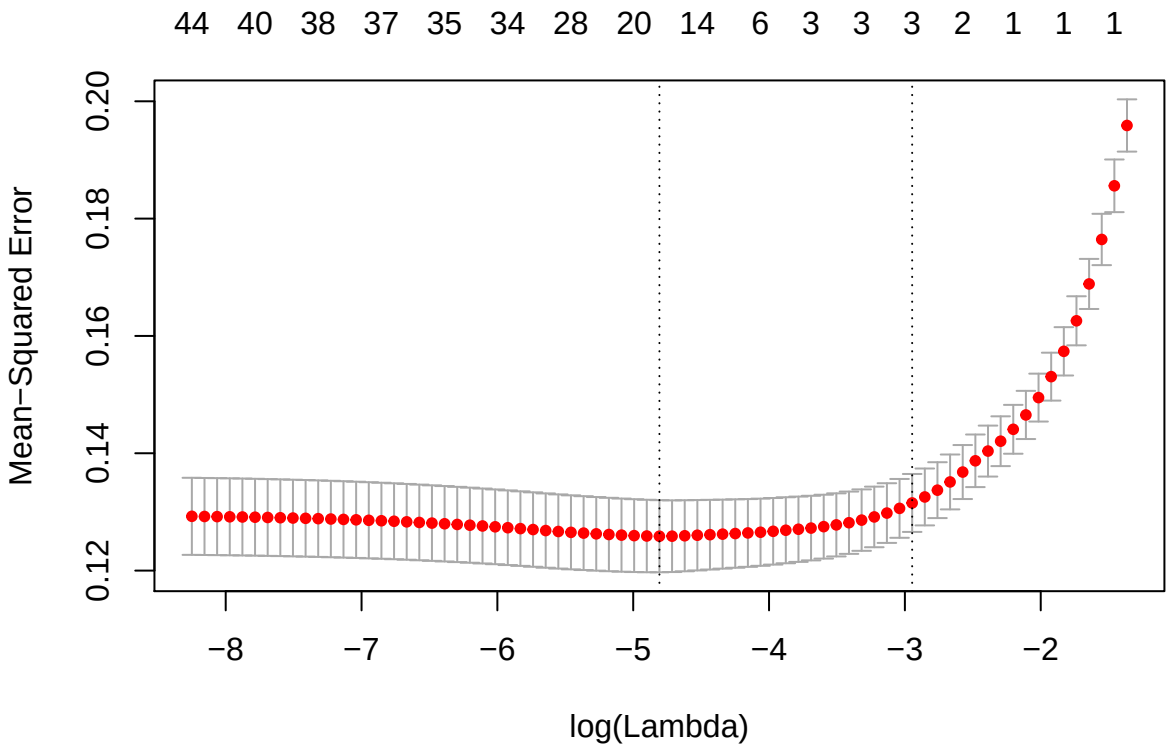
The colored line represents the value taken by a different coefficient in model 1. Lambda is the weight given to the regularization term (the L1 norm), so as lambda approaches zero, the loss function of your model approaches the OLS loss function. As lambda becomes larger, the regularization term has a greater effect and you will see fewer variables in your model. Our ideal lambda is the one that minimizes the residuals. In this case, a log of lambda of approximately -5 seems to be appropriate, as seen in the subsequent graph with MSE.

```
#Remove variables not suited for Lasso:
# [5] "state"
# [6] "state_abbr"
# [8] "municipality"
# [45] "source"
# [46] "organized_crime_lethality"
# [47] "army_lethality"
# [48] "navy_lethality"
# [49] "federal_police_lethality"
# [54] "organized_crime_NewIndex"
# [55] "army_NewIndex"
# [56] "navy_NewIndex"
# [57] "federal_police_NewIndex"
# [59] "category"
```

```
# [60] "global_id"
# [64] "Level.Social.Lag.2005"


### We first set up x and y
slim.dta.pov <- dta.pov[,-c(1,2,5,6,8,45,46,47,48,49,54,55,56,57,59,60,62,64)]

slim.dta.pov <- na.omit(slim.dta.pov)

set.seed(12345) # = Seed for replication

x <- model.matrix(perfect_lethality ~ ., data = slim.dta.pov)[,-46]
y <- slim.dta.pov$perfect_lethality

### We then fit a Lasso regression model (alpha = 1)
fit.lasso <- glmnet(x, y, alpha = 1, family = "gaussian")
plot(fit.lasso, xvar = "lambda", label = TRUE)
```




```
### Now we cross-validate
cv.lasso <- cv.glmnet(x, y)
plot(cv.lasso)
```



Cross-Validation plot suggests that the model works best when it has approximately 5 predictors. We can use cross-validation to extract coefficients that collectively minimize mean squared error.

Below are the 18 predictors of Perfect Lethality that collectively minimize mean squared error. Out of 70 variables, only 18 seem to be ideal candidates for predictors of perfect lethality.

Num	Variable	Lasso coefficient
1	detained	-0.0003073193
2	military_dead	-0.0184931882
3	ministerial_police_dead	-0.0074495572
4	municipal_police_dead	-0.0166922238
5	civilian_dead	-0.0001098928
6	total_people_wounded	-0.0037244742
7	military_wounded	-0.0074350113
8	navy_wounded	-0.0081752818
9	small_arms_seized	0.0030058344
10	army	0.1096297637
11	ministerial_police	-0.0286733220
12	municipal_police	-0.0784595966
13	navy	0.0298730962
14	other	-0.0166464452
15	state_police	-0.0492937858
16	organized_crime_lethality_diff	0.1175320252
17	Vulnerables.SocialCare.Pct.2010	-0.0000926747
18	Access.Health.Services.pct	0.0006595007

Regressing perfect lethality on significant predictors

```
### Regression with all variables
model.1 <- glm(perfect_lethality ~ detained + military_dead + ministerial_police_dead + municipal_polic

summary(model.1)
```

```
##
## Call:
## glm(formula = perfect_lethality ~ detained + military_dead +
##     ministerial_police_dead + municipal_police_dead + civilian_dead +
##     total_people_wounded + military_wounded + navy_wounded +
##     small_arms_seized + army + ministerial_police + municipal_police +
##     navy + other + state_police + organized_crime_lethality_diff +
##     Vulnerables.SocialCare.Pct.2010 + Access.Health.Services.pct,
##     family = binomial(link = "logit"), data = dta.pov)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4904  -0.3963  -0.2872   0.0062   3.2782
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -2.615077   0.277719  -9.416  < 2e-16 ***
## detained                         0.001155   0.025093   0.046   0.9633
## military_dead                   -0.625257   0.355562  -1.759   0.0787 .
## ministerial_police_dead         -0.412567   0.339042  -1.217   0.2237
## municipal_police_dead           -0.058408   0.183148  -0.319   0.7498
## civilian_dead                   -0.080882   0.149684  -0.540   0.5890
## total_people_wounded            -0.541004   0.069383  -7.797 6.32e-15 ***
## military_wounded                 0.227644   0.141696   1.607   0.1082
## navy_wounded                    -0.184586   0.523060  -0.353   0.7242
## small_arms_seized                0.026687   0.031137   0.857   0.3914
## army                             0.660978   0.141499   4.671 2.99e-06 ***
## ministerial_police               0.264904   0.228145   1.161   0.2456
## municipal_police                -0.116540   0.183220  -0.636   0.5247
## navy                             0.456907   0.357637   1.278   0.2014
## other                           -0.525305   0.406856  -1.291   0.1967
## state_police                    -0.015580   0.226983  -0.069   0.9453
## organized_crime_lethality_diff   2.541381   0.092066  27.604  < 2e-16 ***
## Vulnerables.SocialCare.Pct.2010 -0.004308   0.006822  -0.631   0.5277
## Access.Health.Services.pct       0.009524   0.005412   1.760   0.0785 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5045.4  on 4333  degrees of freedom
## Residual deviance: 2089.4  on 4315  degrees of freedom
## AIC: 2127.4
##
## Number of Fisher Scoring iterations: 25
```

Interpretation:

Only the statistically significant coefficients will be addressed.

It is the estimated amount by which the log odds of leaves.presence would increase if Area were one unit higher.

1. Total_people_wounded: On average, a one unit increase in the total people wounded ratio is associated with a **decrease** in the log odds **of 0.541004** of having a perfect lethality event.
2. Army: On average, the presence of the army is associated with a **increase** in the log odds **0.660978** of having a perfect lethality event.
3. Organized_crime_lethality_diff: On average, a one unit increase in the difference in organized crime lethality is associated with a **increase** in the log odds ___of 2.541381___of having a perfect lethality event.

The first three coefficients are statistically significant at the 0.000 level. In other words, the results we see are unlikely to be caused by luck. The following to coefficients are fairly weak statistically speaking but they present substantive insights.

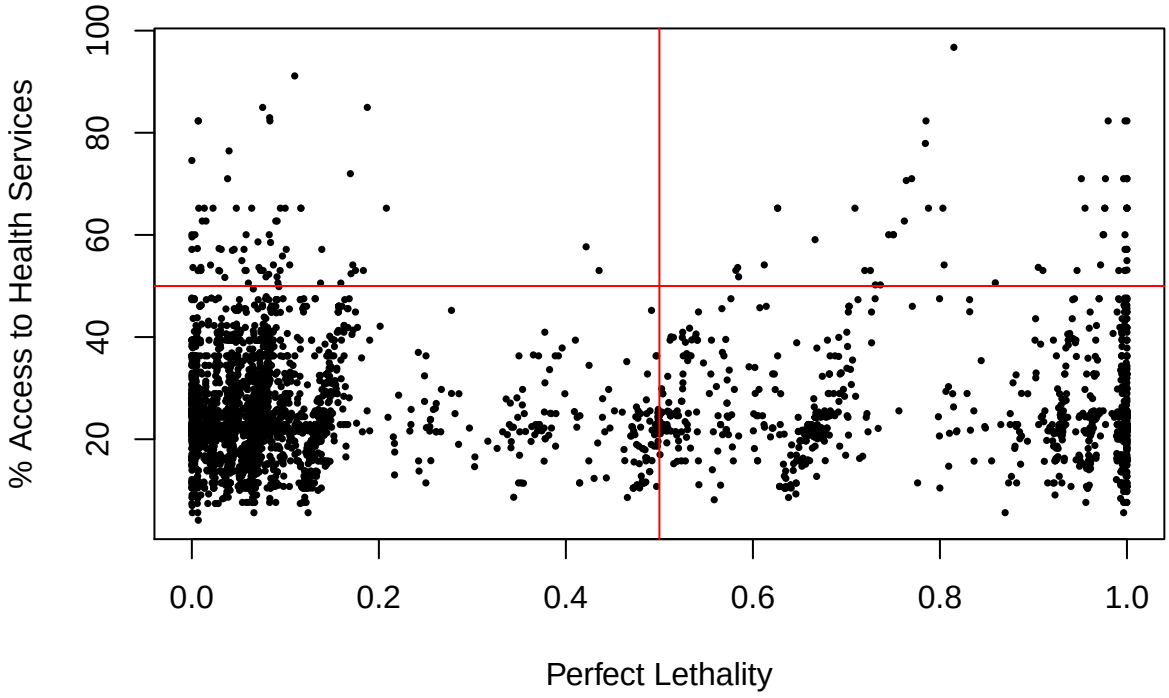
4. Military_dead: On average, each additional death in the military is associated with a **decrease** in the log odds **of 0.6252574** of having a perfect lethality event.
5. Access.Health.Services.pct: On average, each additional death in the military is associated with a **increase** in the log odds **of 0.009524** of having a perfect lethality event.

Visualization

Graph 3 illustrates a substantive (although not surprising) finding. Municipalities that have precarious access to health services (less than 50%) have a higher log odds of having a perfect lethality event, as seen by the red horizontal line.

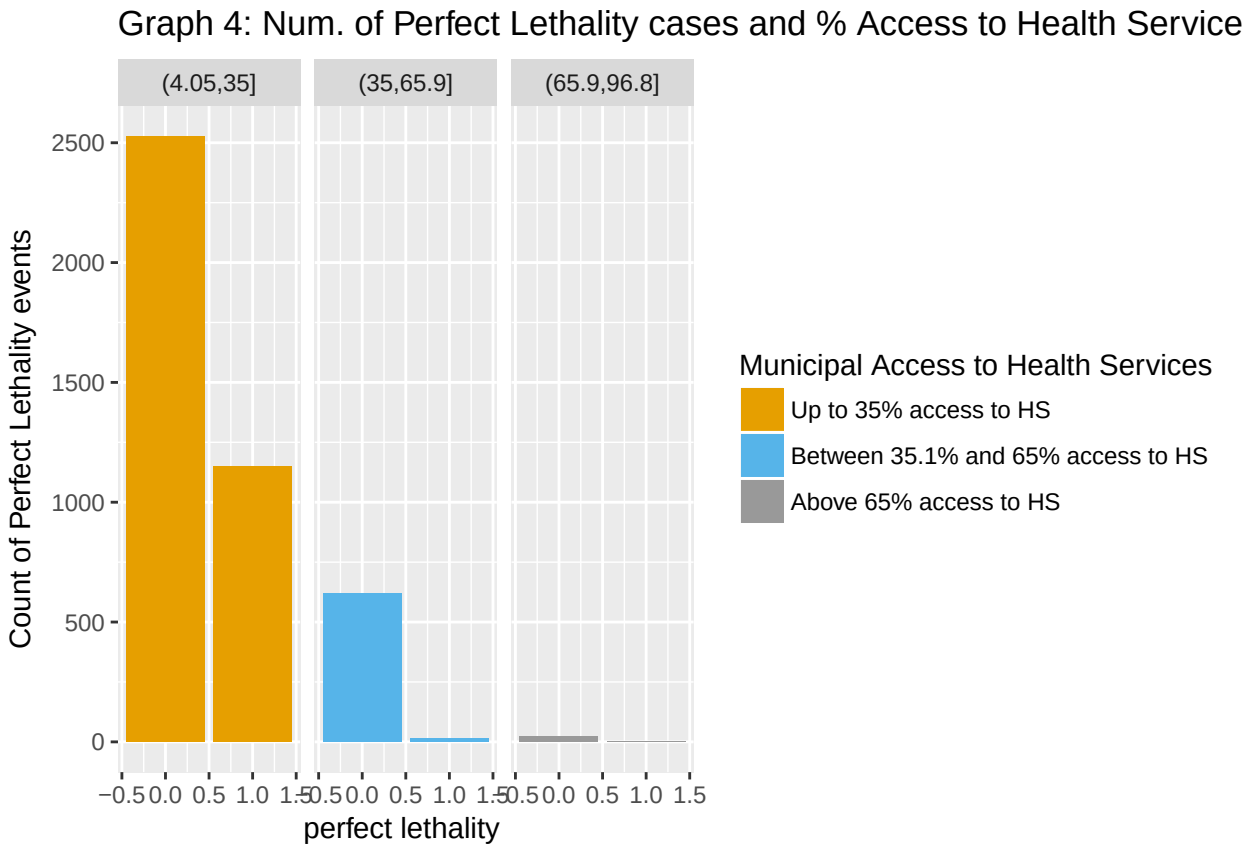
```
set.seed(12345)
ypredict <- predict(model.1, type = "response")

plot(ypredict, dta.pov$Access.Health.Services.pct, pch = 16, xlab="Perfect Lethality", ylab="% Access t
abline(h = 50, v = .5, col = "red")
```

Graph 3: Perfect Lethality events and Municipal Access to Health Serv



Graph 4 further supports our finding from the logit regression.

```
plot.gg <- ggplot(dta.pov)

plot.gg + geom_bar(aes(dta.pov$perfect_lethality, fill = cut(Access.Health.Services.pct,3)))  + ggtitle
                        name="Municipal Access to Health Services",
                        labels=c("Up to 35% access to HS", "Between 35.1% and 65% access to HS", "Above
```



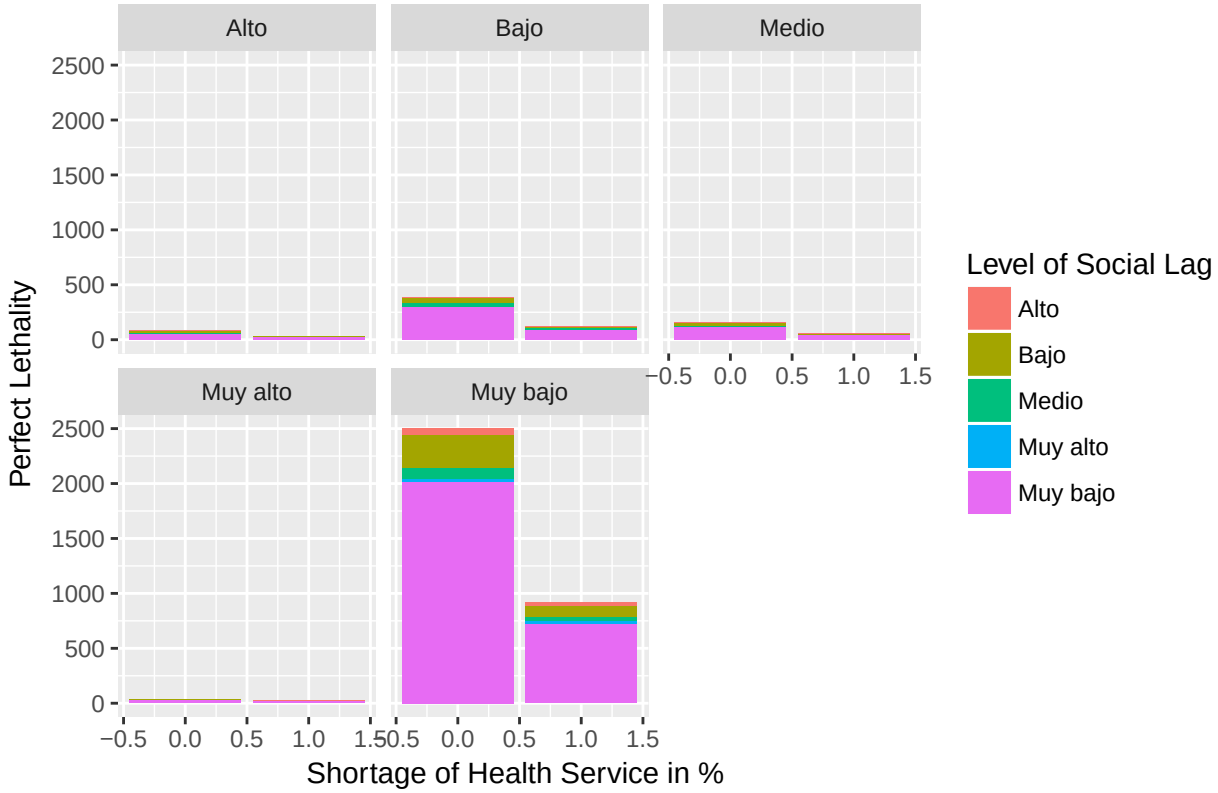
Graph 5 explores the relationship between the index for social development (Índice de Rezago Social [1] [2]) with perfect lethality events. Once again, we see that the lower levels of social development are associated with higher perfect lethality cases.

```
plot.gg + geom_bar(aes(dta.pov$perfect_lethality, fill = dta.pov$Level.Social.Lag.2005))  + ggtitle("Gra
```

[1]'Given that the General Law of Social Development establishes that the measurement of poverty must consider the multidimensional nature of poverty, CONEVAL constructed the Social Lag Index, incorporating indicators of education, access to health services, basic services, quality and spaces in the home, and assets in the home. The Social Recession Index is a weighted measure that summarizes four indicators of social deprivation (education, health, basic services and spaces in housing) in a single index that aims to order observation units according to their social needs.

[2]Source: https://www.coneval.org.mx/Medicion/IRS/Paginas/Que-es-el-indice-de-rezago-social.aspx

Graph 5: Num. of Perfect Lethality cases and Poverty levels




```
# + scale_fill_manual( values=c("#E69F00", "#56B4E9", "#999999"),
#                      name="Poverty Level",
#                      labels=c("Up to 36% are poor", "Between 36.1% and 67% are poor", "Above 67% ar
```

Discussion

Insight:

My initial hypothesis was confirmed and significant evidence was found supporting that there is a statistically significant difference between the perfect lethality in municipalities with low levels of socioeconomic development to those with high development.

Limitations:

Although our model seems to capture a significant portion of the variance in the dataset, it is important to remember that the dataset itself is not representative of the reality and it does not capture context.

Q.2

`Are we able to predict if an event had perfect lethality from the coeeficients not relating to death from in Model 1?`

Predictions with Lasso Coef.

Num	Variable

Num	Variable
9	detained
11	military_dead
16	ministerial_police_dead
17	municipal_police_dead
20	civilian_dead
21	total_people_wounded
22	military_wounded
23	navy_wounded
24	small_arms_seized
38	army
40	ministerial_police
41	municipal_police
42	navy
43	other
44	state_police
46	organized_crime_lethality_diff
68	Vulnerables.SocialCare.Pct.2010
70	Access.Health.Services.pct

```
#Choose only Lasso coefficients
set.seed(11101)
dataset1 <- dta.pov[,c(1,9,11,16,17,20,21,22,23,24,38,40,41,42,43,44,68,70)]

dataset1[1] <- as.factor(dta.pov$perfect_lethality)
colnames(dataset1)[1] <- "perfect_lethality"

aggr(dataset1, plot = FALSE, col = c('navyblue','red'), numbers = TRUE, sortVars = TRUE, labels = names
```

```
##
##  Missings in variables:
## [1] Variable Count
## <0 rows> (or 0-length row.names)
```

We slit training and testing at 3/4 and 1/4 respectively.

```
in_train <- createDataPartition(y = dataset1$perfect_lethality, p = 3 / 4, times = 1, list = FALSE)
training1 <- dataset1[ in_train, ]
testing1  <- dataset1[-in_train, ]

#Compare
dim(training1)
```

```
## [1] 3251   18
```

```
dim(testing1)
```

```
## [1] 1083   18
```

```
options(java.parameters = "-Xmx5g")
set_bart_machine_num_cores(parallel::detectCores())
```

```
## bartMachine now using 4 cores.
```

```
training1 <- as.data.frame(training1)
#Bart machine for classification then y must be a factor otherwise it will assume regression.
yvar <- training1$perfect_lethality
bart <- bartMachine(training1[,2:18], yvar, mem_cache_for_speed = FALSE)
```

```
## bartMachine initializing with 50 trees...
## bartMachine vars checked...
## bartMachine java init...
## bartMachine factors created...
## bartMachine before preprocess...
## bartMachine after preprocess... 18 total features...
## bartMachine sigsq estimated...
## bartMachine training data finalized...
## Now building bartMachine for classification ...
## evaluating in sample data...done
```

```
bart
```

```
## bartMachine v1.2.3 for classification
##
## training data n = 3251 and p = 17
## built in 18.4 secs on 4 cores, 50 trees, 250 burn-in and 1000 post. samples
##
## confusion matrix:
##
##            predicted 0 predicted 1 model errors
## actual 0      2203.000     174.000        0.073
## actual 1       597.000     277.000        0.683
## use errors       0.213       0.386        0.237
```

```
testing1 <- data.frame(testing1)
pred1 <- predict(bart, testing1[,2:18], type = "class")

table(pred1)
```

```
## pred1
##   0   1
## 927 156
```

```
confusionMatrix(pred1, testing1$perfect_lethality)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 726 201
##          1  66  90
##
##                Accuracy : 0.7535
##                  95% CI : (0.7267, 0.7789)
##     No Information Rate : 0.7313
##     P-Value [Acc > NIR] : 0.05264
##
##                   Kappa : 0.2648
##  Mcnemar's Test P-Value : 2.391e-16
```

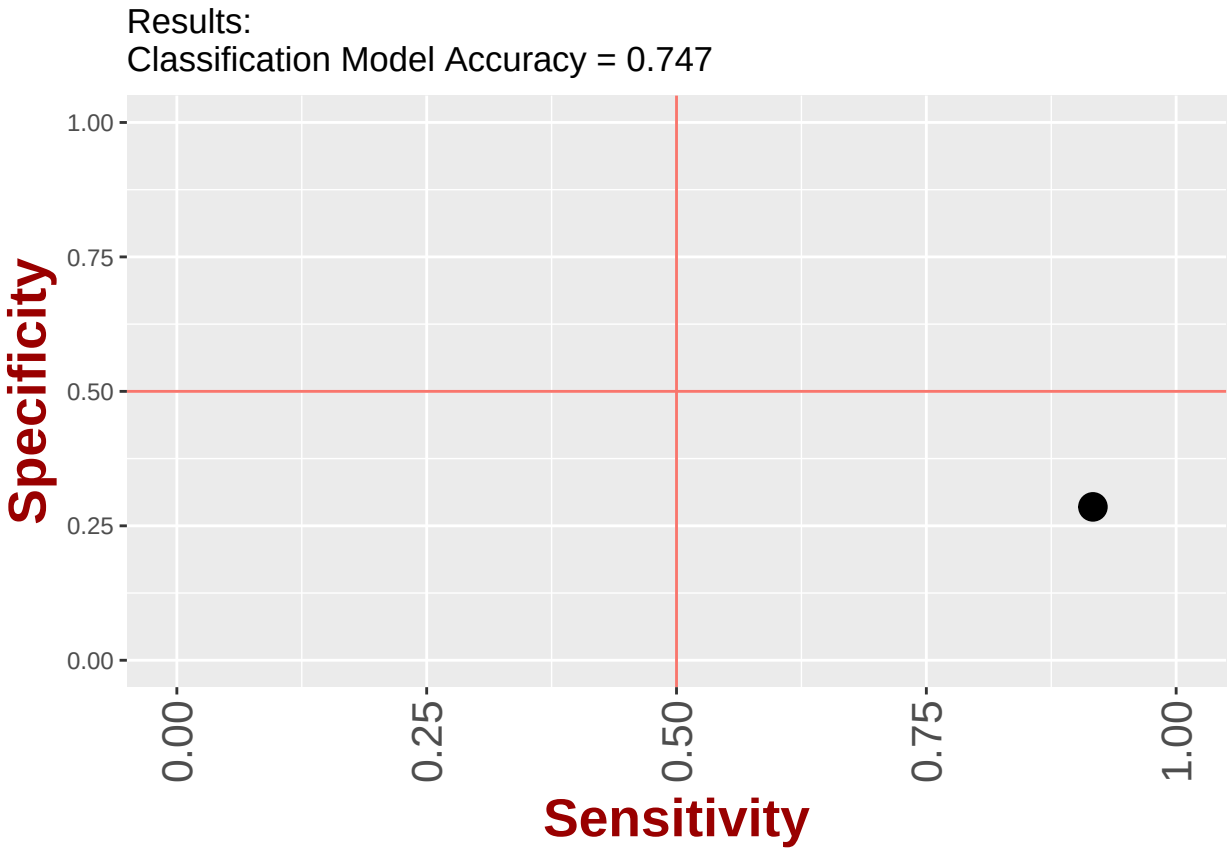
```
##
##             Sensitivity : 0.9167
##             Specificity : 0.3093
##          Pos Pred Value : 0.7832
##          Neg Pred Value : 0.5769
##              Prevalence : 0.7313
##          Detection Rate : 0.6704
##    Detection Prevalence : 0.8560
##       Balanced Accuracy : 0.6130
##
##        'Positive' Class : 0
##
```

```
df <- as.data.frame(matrix(c(0.9167,0.2852), nrow = 1, ncol = 2))

#colnames(df)[1] <- "Results"
colnames(df)[1] <- "Sensitivity"
colnames(df)[2] <- "Specificity"
```

Discussion (Insights and Limitations)

The predictors found in lasso were not able to generate good classifications when used with BartMachine. The accuracy of predictions was approximately 75%. The model does fairly well classifying non-perfect lethality events (Sensitivity of 0.9116) but does poorly when trying to classify perfect lethality events (specificity of 0.3230).

```
ggplot(df) + geom_point(aes(Sensitivity, Specificity, cex = 30)) + ylim(0:1) +xlim(0:1) + geom_vline(aes
           axis.text.x  = element_text(angle=90, vjust=0.5, size=16), axis.title.y = element_text(face=
```



Conclusion

It seems from both the glm and Bartmachine models that there are extreme cases of violence that are easier to predict if they will have a perfect lethality rate or not. However, there are some characteristics of events that makes it hard to classify, probably these events have many homogeneous characteristics. The limitations of this project are that it is still unclear which characteristics are better at predicting perfect lethality.
