# Post feature Selection

Alyzeh Jiwani

27/06/2022

## Post Feature Selection

From our analysis using decision trees and permutation fetaure importance in python, we are able to now remove some features that are unlikely to contribute to our to our models.

For df_new, where the values for a ll the features and our response variable crime_rate are taken in the same year, we decided to keep the following features: Com_House, Child_Care, Pop

For df_2, where values for features are taken three years prior to values for our response variable crime_rate, the features we have decided to keep are: Inflation, Recreation, Com_House, Pop,Emp_Res and Child_Care

```
Data_Current <- read.csv('/Users/alyzehjiwani/Downloads/Data/Final Data Used/Crime_Data.csv')
Data_Current <- Data_Current[,c('Com_House','Child_Care','Pop','C_Rate')]
Data_Gap_Train <- read.csv('//Users/alyzehjiwani/Downloads/Data/Final Data Used/Crime_Data_Year_Gap_Trai
Data_Gap_Test <- read.csv('//Users/alyzehjiwani/Downloads/Data/Final Data Used/Crime_Data_Year_Gap_Test
Data_Gap_Train <- Data_Gap_Train[,c('Inflation','Recreation','Com_House','Child_Care','Pop','Emp_Res','C
Data_Gap_Test <- Data_Gap_Test[,c('Inflation','Recreation','Com_House','Child_Care','Pop','Emp_Res')]
```

We will now try to fit the data to a multinomial regression model

```
library(nnet)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

Splitting into Train/Test

```
index <- createDataPartition(Data_Current$C_Rate, p = 0.7, list = FALSE)
train <- Data_Current[index,]
test <- Data_Current[-index,]
```

```
model_cur_1 <- multinom(C_Rate~., data = Data_Current)
```

```
## # weights:  25 (16 variable)
## initial  value 1802.570462
## iter  10 value 1631.449756
## iter  20 value 1428.371682
## iter  30 value 1380.289944
```

```
## iter  40 value 1379.955843
## iter  50 value 1379.890921
## final  value 1379.875847
## converged
```

```
summary(model_cur_1)
```

```
## Call:
## multinom(formula = C_Rate ~ ., data = Data_Current)
##
## Coefficients:
##           (Intercept)    Com_House   Child_Care           Pop
## Low         0.1417881 -0.009605888 -0.002917352  1.537124e-06
## Medium      0.2372870 -0.006698838  0.008215376 -6.143905e-07
## Very High   3.7080610  0.030277578 -0.024117034 -8.310479e-04
## Very Low   -0.7734812 -0.051137604 -0.008382748  2.029466e-05
##
## Std. Errors:
##            (Intercept)   Com_House  Child_Care          Pop
## Low       6.310972e-05 0.003474224 0.005148986 4.660281e-06
## Medium    8.510603e-05 0.003005985 0.004527637 4.526159e-06
## Very High 2.349552e-04 0.016378640 0.010831335 9.470085e-05
## Very Low  8.717877e-05 0.011527159 0.007461553 6.219032e-06
##
## Residual Deviance: 2759.752
## AIC: 2791.752
```

```
exp(coef(model_cur_1))
```

```
##           (Intercept) Com_House Child_Care       Pop
## Low          1.152332 0.9904401  0.9970869 1.0000015
## Medium       1.267805 0.9933235  1.0082492 0.9999994
## Very High   40.774667 1.0307406  0.9761715 0.9991693
## Very Low     0.461404 0.9501479  0.9916523 1.0000203
```

These are the probabilities of neighbourhoods being having a particular crime rate level

```
head(round(fitted(model_cur_1),3))
```

```
##    High   Low Medium Very High Very Low
## 1 0.249 0.292  0.314     0.001    0.144
## 2 0.250 0.288  0.304     0.000    0.158
## 3 0.247 0.291  0.310     0.000    0.152
## 4 0.369 0.274  0.337     0.001    0.019
## 5 0.229 0.239  0.418     0.000    0.114
## 6 0.284 0.294  0.321     0.000    0.101
```

We now want to see what the accuracy of the model is.

```r
train$C_RatePred <- predict(model_cur_1, newdata = train, 'class')
tab <- table(train$C_Rate, train$C_RatePred)
tab
```

```
##
##              High Low Medium Very High Very Low
##   High         42   1    135        18        0
##   Low          16   1    176         0        3
##   Medium       24   0    212         0        0
##   Very High     0   0      1        78        0
##   Very Low      4   1     74         0        0
```

Now we calculate accuracy

```r
round((sum(diag(tab))/sum(tab))*100,2)
```

```
## [1] 42.37
```

Our accuracy is at a value of 41.98.

We now predict on the test dataset and see our classification table

```r
test$C_RatePred <- predict(model_cur_1, newdata = test, "class")


tab_test <- table(test$C_Rate, test$C_RatePred)
tab_test
```

```
##
##              High Low Medium Very High Very Low
##   High         16   1     57        10        0
##   Low           3   1     80         0        0
##   Medium       12   0     88         0        0
##   Very High     0   0      0        33        0
##   Very Low      2   0     31         0        0
```

accuracy of train model

```r
round((sum(diag(tab_test))/sum(tab_test))*100,2)
```

```
## [1] 41.32
```

We now repeat this for the data where values of our features are taken three years prior to the values of our response variable.

```r
model_gap_1 <- multinom(C_Rate~., data = Data_Gap_Train)
```

```
## # weights:  40 (28 variable)
## initial  value 1126.606539
## iter  10 value 1079.962845
```

```
## iter  20 value 864.260205
## iter  30 value 770.228815
## iter  40 value 716.371036
## iter  50 value 705.530032
## iter  60 value 702.512986
## iter  70 value 702.492899
## iter  80 value 702.238433
## final  value 702.228725
## converged
```

```
summary(model_gap_1)
```

```
## Call:
## multinom(formula = C_Rate ~ ., data = Data_Gap_Train)
##
## Coefficients:
##           (Intercept)  Inflation Recreation    Com_House   Child_Care
## Low          3.150807  -2.002632 -0.5985208 -0.002496771 -0.011897973
## Medium       2.720133  -1.497873 -0.5127378 -0.002773046 -0.004998913
## Very High -1216.077553 512.015922  1.1156268 -0.020437910  0.022570692
## Very Low     1.813895  -1.566388 -0.6712453 -0.033266857 -0.003974328
##                   Pop    Emp_Res
## Low       5.838618e-05 -0.5978852
## Medium    3.072436e-05 -0.2388855
## Very High 4.458361e-03 -1.1500153
## Very Low  5.497935e-05 -0.8552482
##
## Std. Errors:
##           (Intercept)    Inflation   Recreation    Com_House   Child_Care
## Low       5.030706e-05 9.001385e-05 6.409508e-05 4.757071e-03 0.0068568324
## Medium    6.224132e-05 1.185701e-04 8.166292e-05 4.087756e-03 0.0060243150
## Very High 4.885472e-07 1.140393e-06 4.209951e-06 7.361959e-05 0.0002965789
## Very Low  9.513441e-05 1.739427e-04 2.248637e-04 1.536962e-02 0.0085714111
##                   Pop      Emp_Res
## Low       7.196787e-06 4.801052e-05
## Medium    6.592175e-06 1.279104e-04
## Very High 5.233246e-05 5.374379e-06
## Very Low  1.027359e-05 1.083508e-04
##
## Residual Deviance: 1404.457
## AIC: 1460.457
```

```
exp(coef(model_gap_1))
```

```
##           (Intercept)     Inflation Recreation Com_House Child_Care      Pop
## Low         23.354900  1.349795e-01  0.5496241 0.9975063  0.9881725 1.000058
## Medium      15.182334  2.236052e-01  0.5988538 0.9972308  0.9950136 1.000031
## Very High    0.000000 2.321076e+222  3.0514802 0.9797695  1.0228273 1.004468
## Very Low     6.134294  2.087981e-01  0.5110717 0.9672804  0.9960336 1.000055
##             Emp_Res
## Low       0.5499735
## Medium    0.7875050
## Very High 0.3166319
## Very Low  0.4251776
```

These are the probabilities of neighbourhoods being having a particular crime rate level

```
head(round(fitted(model_gap_1),3))
```

```
##    High   Low Medium Very High Very Low
## 1 0.265 0.260  0.325         0    0.151
## 2 0.239 0.295  0.355         0    0.112
## 3 0.334 0.216  0.354         0    0.096
## 4 0.305 0.303  0.351         0    0.040
## 5 0.227 0.253  0.316         0    0.204
## 6 0.176 0.375  0.318         0    0.131
```

We now want to see what the accuracy of the model is.

```
Data_Gap_Train$C_RatePred <- predict(model_gap_1, newdata = Data_Gap_Train, 'class')
tab_gap <- table(Data_Gap_Train$C_Rate, Data_Gap_Train$C_RatePred)
tab_gap
```

```
##
##              High Low Medium Very High Very Low
##   High        104  16     54         3        0
##   Low          21  77     64         0        0
##   Medium       47  57     88         0        0
##   Very High     1   0      0       111        0
##   Very Low      5  26     26         0        0
```

Now we calculate accuracy

```
round((sum(diag(tab_gap))/sum(tab_gap))*100,2)
```

```
## [1] 54.29
```

Our accuracy for this model is slightly better at 52.93

We now predict on the test dataset. We cannot make a classification table as we have no actual values to compare the predicted values to.

```
head(predict(model_gap_1, newdata = Data_Gap_Test, "class"))
```

```
## [1] Medium High   High   High   High   High
## Levels: High Low Medium Very High Very Low
```

Our values for accuracy are fairly low. This is likely to be primarily due to our lack of data points but can also be a result of class imbalance in the dataset (This would be tough to fix as we cannot change the number of high/low/etc level crime rate neighbourhoods). It is also likely that our inability to source demographic information for each data point is a large contributing factor. While we can see that the features we have chosen here are relevant to our response variable, it can be hypothesised that demographic information is a key element in predicting the crime rates of neighbourhoods, especially features that are poverty indicators such as average household size, income, unemployment rate, etc. Without these, our data is lacking important context.