SMOTE Multinomial Model Fitting

Alyzeh Jiwani

25/07/2022

We will now repeat the entire process of post feature selection and multinomial model fitting with our new, larger, datasets containing our synthesised data. From our process of feature selection that we did in our Python Notebook, we already saw a slight improvement in our decsion tree model accuracy scores. Our hope is that we will see an improvement in the effectiveness of our multinomial model as well.

```
SData_Current <- read.csv('/Users/alyzehjiwani/Downloads/Data/Final Data Used/SMOTE_data.csv')
SData_Gap_Train <- read.csv('//Users/alyzehjiwani/Downloads/Data/Final Data Used/Year_Gap_SMOTE_Train_da
SData_Gap_Test <- read.csv('/Users/alyzehjiwani/Downloads/Data/Final Data Used/Crime_Data_Year_Gap_Test
```

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 SMOTE_data, 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, Emp_Res and 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, Year, Emp_Res, Pop and Com_House

```
SData_Current <- SData_Current[,c('Com_House','Child_Care','Pop','Emp_Res','C_Rate')]
SData_Gap_Train <- SData_Gap_Train[,c('Inflation', 'Year', 'Com_House', 'Pop','Emp_Res', 'C_Rate')]
SData_Gap_Test <-SData_Gap_Test[,c('Inflation', 'Year', 'Com_House', '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
```

Working with SData_Current:

Splitting into Train/Test

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

```r
model_Scur_1 <- multinom(C_Rate~., data = SData_Current)
```

```
## # weights:  30 (20 variable)
## initial  value 2703.855693
## iter  10 value 2306.648738
## iter  20 value 1932.199874
## iter  30 value 1813.401523
## iter  40 value 1810.620305
## final  value 1810.602419
## converged
```

```r
summary(model_Scur_1)
```

```
## Call:
## multinom(formula = C_Rate ~ ., data = SData_Current)
##
## Coefficients:
##   (Intercept)  Com_House   Child_Care           Pop   Emp_Res
## 1  0.03543474 0.04476201  0.004588498 -2.449197e-05 0.1333186
## 2  0.04703063 0.04696612  0.016648530 -4.644601e-05 0.4647420
## 3  0.21698055 0.05322401  0.010265796 -8.225881e-05 0.7917630
## 4  4.43596237 0.08196770 -0.020493060 -1.165475e-03 1.3111520
##
## Std. Errors:
##    (Intercept)   Com_House  Child_Care          Pop      Emp_Res
## 1 4.870650e-05 0.007765118 0.005213027 4.482774e-06 9.722959e-05
## 2 4.710003e-05 0.007748095 0.004906921 4.700298e-06 1.122569e-04
## 3 1.039807e-04 0.007765348 0.005324381 5.053545e-06 2.214359e-04
## 4 2.098251e-04 0.018796385 0.010934331 1.276946e-04 3.343709e-04
##
## Residual Deviance: 3621.205
## AIC: 3661.205
```

```r
exp(coef(model_Scur_1))
```

```
##   (Intercept) Com_House Child_Care       Pop  Emp_Res
## 1    1.036070  1.045779  1.0045990 0.9999755 1.142614
## 2    1.048154  1.048086  1.0167879 0.9999536 1.591603
## 3    1.242320  1.054666  1.0103187 0.9999177 2.207284
## 4   84.433342  1.085421  0.9797155 0.9988352 3.710446
```

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

```r
head(round(fitted(model_Scur_1),3))
```

```
##       0     1     2     3 4
## 1 0.344 0.271 0.214 0.170 0
## 2 0.359 0.272 0.221 0.149 0
## 3 0.283 0.236 0.243 0.239 0
## 4 0.056 0.355 0.297 0.293 0
## 5 0.355 0.260 0.277 0.108 0
## 6 0.322 0.353 0.213 0.113 0
```

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

```
train$C_RatePred <- predict(model_Scur_1, newdata = train, 'class')
tab <- table(train$C_Rate, train$C_RatePred)
tab
```

```
##
##       0   1   2   3   4
##   0 173  17  32  18   0
##   1 133  32  31  35   0
##   2  82  36  43  75   0
##   3  36  41  34 105  20
##   4   0   0   0   0 236
```

Now we calculate accuracy

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

```
## [1] 49.96
```

Our accuracy is at a value of 50.21, which is higher than our original value of 42.75

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

```
test$C_RatePred <- predict(model_Scur_1, newdata = test, "class")


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

```
##
##       0   1   2   3   4
##   0  66  11  10   9   0
##   1  62  13   9  21   0
##   2  35  17  18  30   0
##   3  15   8  14  51  12
##   4   0   0   0   0 100
```

accuracy of test model

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

```
## [1] 49.5
```

Again, this is higher than our original value of 40.42

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

```
model_Sgap_1 <- multinom(C_Rate~., data = SData_Gap_Train)
```

```
## # weights:  35 (24 variable)
## initial  value 1545.060396
## iter  10 value 1521.471813
## iter  20 value 1168.729640
## iter  30 value 1016.369746
## iter  40 value 985.509795
## iter  50 value 977.818090
## iter  60 value 976.798117
## iter  70 value 975.890379
## iter  80 value 975.808460
## iter  90 value 970.056251
## iter 100 value 969.947559
## final  value 969.947559
## stopped after 100 iterations
```

```
summary(model_Sgap_1)
```

```
## Call:
## multinom(formula = C_Rate ~ ., data = SData_Gap_Train)
##
## Coefficients:
##   (Intercept)   Inflation        Year  Com_House           Pop     Emp_Res
## 1  -321.52245  -0.2324984  0.15940493 0.05404951  1.105814e-05  0.03374738
## 2    71.64339   0.1946673 -0.03582643 0.05346558 -2.522182e-05  0.56206765
## 3  -531.60875   1.2953466  0.26255469 0.06066671 -4.563481e-05  0.84901705
## 4   -50.47378 591.6783241 -0.66903247 0.09143197  4.841996e-03 -0.01320817
##
## Std. Errors:
##    (Intercept)    Inflation         Year   Com_House          Pop      Emp_Res
## 1 6.840353e-08 2.508771e-05 0.0001220574 0.011236088 1.226962e-05 1.227861e-04
## 2 6.447748e-08 1.397687e-05 0.0001239996 0.011228947 1.292896e-05 1.524275e-04
## 3 6.728749e-08 6.201546e-06 0.0001298659 0.011232439 1.363454e-05 7.097931e-05
## 4 3.983029e-06 9.227320e-06 0.0080398183 0.002075954 1.331075e-03 3.522069e-05
##
## Residual Deviance: 1939.895
## AIC: 1987.895
```

```
exp(coef(model_Sgap_1))
```

```
##     (Intercept)     Inflation      Year Com_House       Pop   Emp_Res
## 1 2.315117e-140  7.925510e-01 1.1728128  1.055537 1.0000111 1.0343233
## 2  1.301156e+31  1.214907e+00 0.9648077  1.054921 0.9999748 1.7542960
## 3 1.334295e-231  3.652262e+00 1.3002476  1.062545 0.9999544 2.3373482
## 4  1.200929e-22 9.175532e+256 0.5122039  1.095742 1.0048537 0.9868787
```

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

```
head(round(fitted(model_Sgap_1),3))
```

```
##       0     1     2     3 4
## 1 0.396 0.178 0.260 0.167 0
## 2 0.301 0.203 0.299 0.197 0
```

```
## 3 0.294 0.141 0.313 0.252 0
## 4 0.045 0.277 0.363 0.315 0
## 5 0.445 0.227 0.218 0.111 0
## 6 0.266 0.313 0.274 0.148 0
```

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

```
SData_Gap_Train$C_RatePred <- predict(model_Sgap_1, newdata = SData_Gap_Train, 'class')
tab_gap <- table(SData_Gap_Train$C_Rate, SData_Gap_Train$C_RatePred)
tab_gap
```

```
##
##       0   1   2   3   4
##   0 148  20  15   9   0
##   1 101  49  18  24   0
##   2  81  28  41  42   0
##   3  25  25  42  97   3
##   4   0   0   0   2 190
```

Now we calculate accuracy

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

```
## [1] 54.69
```

Our accuracy for this model is slightly better at 54.69

We now predict on the test dataset.

```
SData_Gap_Test$C_RatePred <- predict(model_Sgap_1, newdata = SData_Gap_Test, "class")

head(SData_Gap_Test$C_RatePred)
```

```
## [1] 3 3 3 3 3 3
## Levels: 0 1 2 3 4
```

Our values for accuracy have improved using the synthesized data.