lab 2 samir

Samir Datta

October 20, 2017

```
library(ggplot2)
library(ordinal)
```

```
## Warning: package 'ordinal' was built under R version 3.4.2
```

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

```
## Warning: package 'caret' was built under R version 3.4.2
```

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

```
labdata <- read.csv('C:/Users/Samir/Documents/MIDS/StatsF17/lab 2/lab2data.csv')
```

Feature enginerring

```
#split major into STEM and non-STEM
labdata$MajorType <- ifelse(labdata$Major=='Biology'|
                              labdata$Major=='Economics'|
                              labdata$Major=='Psychology'|
                              labdata$Major=='Physics'|
                              labdata$Major=='Chemistry'|
                              labdata$Major=='Mathematics'|
                              labdata$Major=='General Science-Chemistry'|
                              labdata$Major=='Economics-Business'|
                              labdata$Major=='General Science-Chemistry'|
                              labdata$Major=='Sociology-Anthropology'|
                              labdata$Major=='General Science-Psycho'|
                              labdata$Major=='General Science-Math'|
                              labdata$Major=='General Science-Biology'|
                              labdata$Major=='Computer Science'|
                              labdata$Major=='General Science'|
                              labdata$Major=='Mathematics-Physics'|
                              labdata$Major=='Economics-Regional Stds.'|
                              labdata$Major=='Zoology'|
                              labdata$Major=='Engineering'|
                              labdata$Major=='Sociology'|
                              labdata$Major=='Anthropology'|
                              labdata$Major=='General Science-Physics',
                            "STEM", "Non-STEM")

#create variable nextDegreeType to categorize the most common next degrees
labdata$NextDegreeType <- ifelse(labdata$Next.Degree=='JD', 'JD',
                          ifelse(labdata$Next.Degree=='MA', 'MA',
                          ifelse(labdata$Next.Degree=='PHD', 'PHD',
                         ifelse(labdata$Next.Degree=='NDA', 'NDA',
                                ifelse(labdata$Next.Degree=='MS', 'MS',
```

```
                                    ifelse(labdata$Next.Degree=='MD', 'MD',
                                           ifelse(labdata$Next.Degree=='MBA', 'MBA',
        ifelse(labdata$Next.Degree=='NONE', 'NONE', 'Other'))))))))

#create simpler variable to represent if someone has an advanced degree or not
labdata$NextDegreeBinary <- ifelse(labdata$Next.Degree=='NONE', 0, 1)


#create buckets for all years
labdata$FY16cat <- cut(labdata$FY16Giving, c(0,1,100,250,500,200000), right=F)
labdata$FY15cat <- cut(labdata$FY15Giving, c(0,1,100,250,500,200000), right=F)
labdata$FY14cat <- cut(labdata$FY14Giving, c(0,1,100,250,500,200000), right=F)
labdata$FY13cat <- cut(labdata$FY13Giving, c(0,1,100,250,500,200000), right=F)
labdata$FY12cat <- cut(labdata$FY12Giving, c(0,1,100,250,500,200000), right=F)

#turn class year into years since grad to make interpretaion easier
labdata$YearsSinceGrad <- 2017 - labdata$Class.Year


#loop through data to get each person's mean donation over the past 4 years
#and how many of the past years they've donated
labdata$meandonation <- NA
labdata$nPastYears <- NA
for (i in c(1:1000)){
  labdata[i,]$meandonation<-mean(c(labdata[i,]$FY12Giving,
                                   labdata[i,]$FY13Giving,
                                   labdata[i,]$FY14Giving,
                                   labdata[i,]$FY15Giving))

  labdata[i,]$nPastYears <- sum(c(labdata[i,]$FY12Giving>0,
                                  labdata[i,]$FY13Giving>0,
                                  labdata[i,]$FY14Giving>0,
                                  labdata[i,]$FY15Giving>0))
}

#binary variable - have they donated before or not?
labdata$past_binary <- ifelse(labdata$meandonation == 0,0,1)

#did they donate last year?
labdata$donatedLastYear <- ifelse(labdata$FY15Giving==0, 0, 1)

#how many of the past 4 years, consecutively, did they donate?
#note that this will give a 0 for those that donated 2012-2014 but NOT 2015
#since we're asking for consecutive years
labdata$nPastConsecutiveYears <- ifelse(labdata$FY15Giving==0, 0,
                                        ifelse(labdata$FY14Giving==0,1,
                                               ifelse(labdata$FY13Giving==0,2,
                                                      ifelse(labdata$FY12Giving==0,3,4))))

#did they give in 2015 or not?
labdata$gaveLastYear <- ifelse(labdata$FY15Giving==0,0,1)
```

```
labdata$donatedLastYear <- ifelse(labdata$FY15Giving==0, 0, 1)
```

```
labdata$nPastConsecutiveYears <- ifelse(labdata$FY15Giving==0, 0,
                                        ifelse(labdata$FY14Giving==0,1,
                                               ifelse(labdata$FY13Giving==0,2,
                                                      ifelse(labdata$FY12Giving==0,3,4))))
labdata$gaveLastYear <- ifelse(labdata$FY15Giving==0,0,1)
```

Model

So far my favorite model has been: FY16cat ~ AttendenceEvent + YearsSinceGrad + Gender+NextDegreeBinary+log(meandonation+1)*gaveLastYear

Ordinal:

```
clm.out <- clm(FY16cat ~ AttendenceEvent + YearsSinceGrad  +
                 Gender+NextDegreeBinary+log(meandonation+1)*gaveLastYear,
               data=labdata)
summary(clm.out)
```

```
## formula:
## FY16cat ~ AttendenceEvent + YearsSinceGrad + Gender + NextDegreeBinary + log(meandonation + 1) * gave
## data:    labdata
##
##  link  threshold nobs logLik  AIC     niter max.grad cond.H
##  logit flexible  1000 -794.52 1611.05 6(0)  4.70e-12 4.6e+04
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## AttendenceEvent                    0.174681   0.165499   1.055  0.29121
## YearsSinceGrad                    -0.013243   0.006159  -2.150  0.03154
## GenderM                            0.131992   0.146901   0.899  0.36891
## NextDegreeBinary                   0.364203   0.160444   2.270  0.02321
## log(meandonation + 1)              0.603935   0.071804   8.411  < 2e-16
## gaveLastYear                      -1.339036   0.421328  -3.178  0.00148
## log(meandonation + 1):gaveLastYear 0.724417   0.116498   6.218 5.03e-10
##
## AttendenceEvent
## YearsSinceGrad                    *
## GenderM
## NextDegreeBinary                  *
## log(meandonation + 1)             ***
## gaveLastYear                      **
## log(meandonation + 1):gaveLastYear ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                       Estimate Std. Error z value
## [0,1)|[1,100)           2.7879     0.2651   10.52
## [1,100)|[100,250)       4.3856     0.2904   15.10
## [100,250)|[250,500)     6.4668     0.3308   19.55
## [250,500)|[500,2e+05)   7.4793     0.3597   20.79
```

Multinomial:

```
mn.out <- multinom(FY16cat ~ AttendenceEvent + YearsSinceGrad  +
                     Gender+NextDegreeBinary+log(meandonation+1)*gaveLastYear,
                   data=labdata)
```

```
## # weights:  45 (32 variable)
## initial  value 1609.437912
## iter  10 value 1078.754175
## iter  20 value 799.465551
## iter  30 value 754.847860
## iter  40 value 751.841785
## iter  50 value 751.809924
## final  value 751.809860
## converged
```

```
summary(mn.out)
```

```
## Call:
## multinom(formula = FY16cat ~ AttendenceEvent + YearsSinceGrad +
##     Gender + NextDegreeBinary + log(meandonation + 1) * gaveLastYear,
##     data = labdata)
##
## Coefficients:
##             (Intercept) AttendenceEvent YearsSinceGrad     GenderM
## [1,100)       -2.525139     -0.04886314   -0.025399784 -0.43923603
## [100,250)     -5.226885      0.37216524   -0.001742145  0.43794705
## [250,500)     -7.564692      1.20721004    0.006009606  0.34016953
## [500,2e+05)   -8.814570      1.08997744   -0.015713861  0.06415283
##             NextDegreeBinary log(meandonation + 1) gaveLastYear
## [1,100)            0.7792853             0.4008516    3.2631417
## [100,250)          0.2771968             0.7726935   -0.2797615
## [250,500)         -0.1180912             0.7245348   -3.7489255
## [500,2e+05)        1.2987126             1.0510589   -8.8325912
##             log(meandonation + 1):gaveLastYear
## [1,100)                            -0.5382772
## [100,250)                           0.4084793
## [250,500)                           1.2525979
## [500,2e+05)                         2.0273536
##
## Std. Errors:
##             (Intercept) AttendenceEvent YearsSinceGrad   GenderM
## [1,100)       0.3085684       0.2151112    0.008652386 0.2035329
## [100,250)     0.5764664       0.2724300    0.010028237 0.2433898
## [250,500)     1.3882672       0.5483083    0.016782169 0.3973434
## [500,2e+05)   1.7275961       0.5636070    0.019144986 0.4323607
##             NextDegreeBinary log(meandonation + 1) gaveLastYear
## [1,100)            0.2226642             0.08256367    0.5480447
## [100,250)          0.2611452             0.13318727    0.8879789
## [250,500)          0.4141892             0.33741807    1.8335387
## [500,2e+05)        0.5489396             0.37139492    2.2408315
##             log(meandonation + 1):gaveLastYear
## [1,100)                             0.1608997
## [100,250)                           0.2194408
## [250,500)                           0.4346482
## [500,2e+05)                         0.4806594
```

```
##
## Residual Deviance: 1503.62
## AIC: 1567.62
```

Predictions

Multinomial:

```
randomRows <- sample(1000, 750, replace=FALSE)
train <- labdata[randomRows,]
test <- labdata[-randomRows,]

mn.out <- multinom(FY16cat ~ AttendenceEvent + YearsSinceGrad  +
                     Gender+NextDegreeBinary+log(meandonation+1)*gaveLastYear,
                   data=train)
```

```
## # weights:  45 (32 variable)
## initial  value 1207.078434
## iter  10 value 734.323399
## iter  20 value 572.028652
## iter  30 value 541.714837
## iter  40 value 538.973367
## iter  50 value 538.590234
## final  value 538.589560
## converged
```

```
preds <- predict(mn.out, test, type="class")
confusionMatrix(preds, test$FY16cat)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)         116      29         9         0           2
##   [1,100)         7      13         4         0           0
##   [100,250)      10       1        24        13           8
##   [250,500)       0       0         0         0           0
##   [500,2e+05)     1       0         1         0          12
##
## Overall Statistics
##
##                Accuracy : 0.66
##                  95% CI : (0.5976, 0.7185)
##     No Information Rate : 0.536
##     P-Value [Acc > NIR] : 4.639e-05
##
##                   Kappa : 0.4427
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: [0,1) Class: [1,100) Class: [100,250)
## Sensitivity                0.8657         0.3023           0.6316
## Specificity                0.6552         0.9469           0.8491
## Pos Pred Value             0.7436         0.5417           0.4286
## Neg Pred Value             0.8085         0.8673           0.9278
```

```
## Prevalence                 0.5360         0.1720           0.1520
## Detection Rate             0.4640         0.0520           0.0960
## Detection Prevalence       0.6240         0.0960           0.2240
## Balanced Accuracy          0.7604         0.6246           0.7403
##                      Class: [250,500) Class: [500,2e+05)
## Sensitivity                     0.000             0.5455
## Specificity                     1.000             0.9912
## Pos Pred Value                    NaN             0.8571
## Neg Pred Value                  0.948             0.9576
## Prevalence                      0.052             0.0880
## Detection Rate                  0.000             0.0480
## Detection Prevalence            0.000             0.0560
## Balanced Accuracy               0.500             0.7683
```

Ordinal:

```
clm.out <- clm(FY16cat ~ AttendenceEvent + YearsSinceGrad  +
                 Gender+NextDegreeBinary+log(meandonation+1)*gaveLastYear,
                   data=train)

preds <- predict(clm.out, test, type="class")
confusionMatrix(preds$fit, test$FY16cat)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)         117      32         8         0           1
##   [1,100)         4       6         6         0           1
##   [100,250)      12       5        23        13           7
##   [250,500)       0       0         0         0           0
##   [500,2e+05)     1       0         1         0          13
##
## Overall Statistics
##
##                Accuracy : 0.636
##                  95% CI : (0.573, 0.6957)
##     No Information Rate : 0.536
##     P-Value [Acc > NIR] : 0.0008746
##
##                   Kappa : 0.4011
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: [0,1) Class: [1,100) Class: [100,250)
## Sensitivity                0.8731         0.1395           0.6053
## Specificity                0.6466         0.9469           0.8255
## Pos Pred Value             0.7405         0.3529           0.3833
## Neg Pred Value             0.8152         0.8412           0.9211
## Prevalence                 0.5360         0.1720           0.1520
## Detection Rate             0.4680         0.0240           0.0920
## Detection Prevalence       0.6320         0.0680           0.2400
## Balanced Accuracy          0.7598         0.5432           0.7154
##                      Class: [250,500) Class: [500,2e+05)
```

```
## Sensitivity                 0.000            0.5909
## Specificity                 1.000            0.9912
## Pos Pred Value                NaN            0.8667
## Neg Pred Value              0.948            0.9617
## Prevalence                  0.052            0.0880
## Detection Rate              0.000            0.0520
## Detection Prevalence        0.000            0.0600
## Balanced Accuracy           0.500            0.7911
```