ws4\_model

April 26, 2019

library(readr)  
library(ggplot2)  
library(caret)  
library(effects)

## Original model

logit4 <- glm(H\_viol\_2017 ~ TREATED\_SOURCE + PTC\_CWS + PTC\_NTNCWS + Other + MR,   
 data=ws4\_P, family="binomial")  
summary(logit4)

##   
## Call:  
## glm(formula = H\_viol\_2017 ~ TREATED\_SOURCE + PTC\_CWS + PTC\_NTNCWS +   
## Other + MR, family = "binomial", data = ws4\_P)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.7224 -0.3146 -0.2326 -0.2239 2.7197   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.67337 0.02344 -156.715 < 2e-16 \*\*\*  
## TREATED\_SOURCE 0.07656 0.02869 2.669 0.007615 \*\*   
## PTC\_CWS 0.69189 0.03022 22.892 < 2e-16 \*\*\*  
## PTC\_NTNCWS 0.18070 0.04740 3.812 0.000138 \*\*\*  
## Other 0.83396 0.05173 16.120 < 2e-16 \*\*\*  
## MR 0.86080 0.04470 19.257 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 47911 on 146350 degrees of freedom  
## Residual deviance: 46349 on 146345 degrees of freedom  
## AIC: 46361  
##   
## Number of Fisher Scoring iterations: 6

# odds ratios and CI's  
exp(coef(logit4))

## (Intercept) TREATED\_SOURCE PTC\_CWS PTC\_NTNCWS Other   
## 0.02539069 1.07956376 1.99748950 1.19805060 2.30241551   
## MR   
## 2.36504250

exp(confint(logit4))

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 0.02424363 0.02657685  
## TREATED\_SOURCE 1.02045490 1.14191392  
## PTC\_CWS 1.88270364 2.11952003  
## PTC\_NTNCWS 1.09093218 1.31374642  
## Other 2.07865196 2.54604231  
## MR 2.16529572 2.58001445

* just re-creating the summary output

# Estimate  
Estimate <- coef(logit4)  
Estimate

## (Intercept) TREATED\_SOURCE PTC\_CWS PTC\_NTNCWS Other   
## -3.67337255 0.07655703 0.69189114 0.18069573 0.83395880   
## MR   
## 0.86079599

# Std. Error  
Std\_Error <- coef(summary(logit4))[, "Std. Error"]  
Std\_Error

## (Intercept) TREATED\_SOURCE PTC\_CWS PTC\_NTNCWS Other   
## 0.02343976 0.02868716 0.03022354 0.04740321 0.05173415   
## MR   
## 0.04470005

# z value  
z\_value <- Estimate/Std\_Error  
z\_value

## (Intercept) TREATED\_SOURCE PTC\_CWS PTC\_NTNCWS Other   
## -156.715462 2.668686 22.892462 3.811888 16.120082   
## MR   
## 19.257160

# Pr(>|z|)  
P\_value <- round(2\*pnorm(abs(z\_value), lower.tail = F), 6)  
P\_value

## (Intercept) TREATED\_SOURCE PTC\_CWS PTC\_NTNCWS Other   
## 0.000000 0.007615 0.000000 0.000138 0.000000   
## MR   
## 0.000000

* manual calculation of an odds ratio with 95% CI

cat("2.5%", unname(exp(Estimate[4] - 1.96\*Std\_Error[4])), "\n\n")

## 2.5% 1.091754

exp(Estimate[4])

## PTC\_NTNCWS   
## 1.198051

cat("\n97.5%", unname(exp(Estimate[4] + 1.96\*Std\_Error[4])))

##   
## 97.5% 1.314697

## Some additional ways to look at model

# not required, but I like to convert outcome to a labelled factor variable for clarity in caret functions, and also the predictors for the effects summaries and plots  
  
ws4\_P$H\_viol\_2017 <- factor(ws4\_P$H\_viol\_2017, levels = c(0,1), labels = c("HB\_N", "HB\_Y"))  
  
ws4\_P[, c(4, 8:11)] <- lapply(ws4\_P[, c(4, 8:11)], as.factor)  
  
prop.table(table(ws4\_P$H\_viol\_2017))

##   
## HB\_N HB\_Y   
## 0.96133269 0.03866731

# get same as original model  
  
logit4 <- glm(H\_viol\_2017 ~ TREATED\_SOURCE + PTC\_CWS + PTC\_NTNCWS + Other + MR,   
 data=ws4\_P, family="binomial")  
  
summary(logit4)

##   
## Call:  
## glm(formula = H\_viol\_2017 ~ TREATED\_SOURCE + PTC\_CWS + PTC\_NTNCWS +   
## Other + MR, family = "binomial", data = ws4\_P)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.7224 -0.3146 -0.2326 -0.2239 2.7197   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.67337 0.02344 -156.715 < 2e-16 \*\*\*  
## TREATED\_SOURCE1 0.07656 0.02869 2.669 0.007615 \*\*   
## PTC\_CWS1 0.69189 0.03022 22.892 < 2e-16 \*\*\*  
## PTC\_NTNCWS1 0.18070 0.04740 3.812 0.000138 \*\*\*  
## Other1 0.83396 0.05173 16.120 < 2e-16 \*\*\*  
## MR1 0.86080 0.04470 19.257 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 47911 on 146350 degrees of freedom  
## Residual deviance: 46349 on 146345 degrees of freedom  
## AIC: 46361  
##   
## Number of Fisher Scoring iterations: 6

## Effects

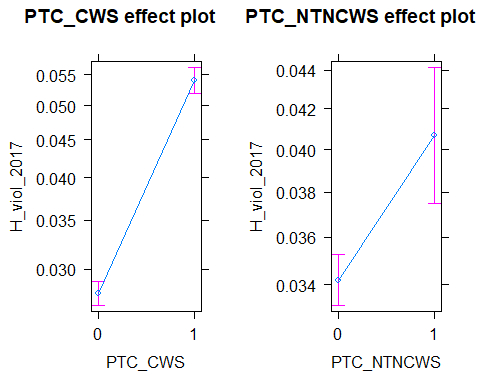
eff\_logit4 <- allEffects(logit4)  
  
eff\_logit4

## model: H\_viol\_2017 ~ TREATED\_SOURCE + PTC\_CWS + PTC\_NTNCWS + Other +   
## MR  
##   
## TREATED\_SOURCE effect  
## TREATED\_SOURCE  
## 0 1   
## 0.03401800 0.03662547   
##   
## PTC\_CWS effect  
## PTC\_CWS  
## 0 1   
## 0.02779963 0.05403119   
##   
## PTC\_NTNCWS effect  
## PTC\_NTNCWS  
## 0 1   
## 0.03418437 0.04067920   
##   
## Other effect  
## Other  
## 0 1   
## 0.03397869 0.07491763   
##   
## MR effect  
## MR  
## 0 1   
## 0.03350375 0.07577241

# more detail on "CWS"  
as.data.frame(eff\_logit4[[2]])

## PTC\_CWS fit se lower upper  
## 1 0 0.02779963 0.0005421252 0.02675658 0.02888213  
## 2 1 0.05403119 0.0010954178 0.05192400 0.05621883

# as factors instead of dummy 0,1 variables get error bars  
plot(eff\_logit4[2:3])



## Accuracy assessment train/test data split

set.seed(147)  
index <- createDataPartition(ws4\_P$H\_viol\_2017, p = 0.8, list = FALSE)  
  
train <- ws4\_P[index, ]  
test <- ws4\_P[-index, ]  
  
  
fitControl <- trainControl(  
 method = "none",   
 savePredictions = TRUE  
)  
  
logit4train <- train(H\_viol\_2017 ~ TREATED\_SOURCE + PTC\_CWS + PTC\_NTNCWS + Other + MR,  
 data=train, method="glm", family=binomial(), trControl=fitControl)  
  
  
logit4test\_pred <- predict(logit4train, test)  
confusionMatrix(logit4test\_pred, test$H\_viol\_2017, positive="HB\_Y")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction HB\_N HB\_Y  
## HB\_N 28138 1131  
## HB\_Y 0 0  
##   
## Accuracy : 0.9614   
## 95% CI : (0.9591, 0.9635)  
## No Information Rate : 0.9614   
## P-Value [Acc > NIR] : 0.5079   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.00000   
## Specificity : 1.00000   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.96136   
## Prevalence : 0.03864   
## Detection Rate : 0.00000   
## Detection Prevalence : 0.00000   
## Balanced Accuracy : 0.50000   
##   
## 'Positive' Class : HB\_Y   
##

* High accuracy because of high % correctly predicted with no HB Violation in 2017, not to say that the predictors aren’t meaningful. Could explore using different probability thresholds, e.g., to maximize specificity and sensitivity.

train$prediction <- predict(logit4train, newdata = train, type = "prob")  
  
print.data.frame(head(train))

## X1 WATER\_SYSTEM.PWSID H\_viol\_2017 TREATED\_SOURCE ZIP\_CODE5 LAT  
## 1 1 010106001 HB\_N 0 06339 41.44948  
## 2 2 010109005 HB\_N 0 06382 41.45720  
## 3 3 010307001 HB\_N 0 02535 41.34558  
## 4 4 010502002 HB\_N 0 02813 41.38526  
## 5 7 020000004 HB\_N 0 14779 42.16851  
## 6 8 020000005 HB\_N 0 13655 44.98184  
## LON PTC\_CWS PTC\_NTNCWS Other MR P prediction.HB\_N  
## 1 -71.98233 1 0 0 0 0.04826953 0.9511825  
## 2 -72.11459 1 0 0 0 0.04826953 0.9511825  
## 3 -70.75145 1 0 0 0 0.04826953 0.9511825  
## 4 -71.66813 0 1 0 0 0.02952132 0.9702282  
## 5 -78.72970 1 0 0 0 0.04826953 0.9511825  
## 6 -74.67374 1 0 0 0 0.04826953 0.9511825  
## prediction.HB\_Y  
## 1 0.04881746  
## 2 0.04881746  
## 3 0.04881746  
## 4 0.02977181  
## 5 0.04881746  
## 6 0.04881746

something adapted from:  
<http://ethen8181.github.io/machine-learning/unbalanced/unbalanced.html>

ggplot(train, aes(prediction$HB\_Y, color = H\_viol\_2017 ) ) +   
geom\_density( size = 1 ) +  
ggtitle( "Training Set's Predicted Score" )

