Stat 148 Project

SUPER BAYES

12/15/2021

#Set working Directory  
getwd()

## [1] "D:/me/Documents/UPLB/1st Semester A.Y. 2021-2022/Stat 148/Project"

setwd("D:/me/Documents/UPLB/1st Semester A.Y. 2021-2022/Stat 148/Project")  
  
#Import  
YAF.raw <- read\_excel("YAFS4-ER\_Final.xlsx")

summary(YAF.raw)

## Bar\_Strat Age I\_Age Contraceptive   
## Min. :1.000 Min. :15.00 Min. :13.00 Min. :0.0000   
## 1st Qu.:2.000 1st Qu.:19.00 1st Qu.:20.00 1st Qu.:0.0000   
## Median :2.000 Median :21.00 Median :23.00 Median :0.0000   
## Mean :1.778 Mean :20.85 Mean :22.74 Mean :0.4412   
## 3rd Qu.:2.000 3rd Qu.:23.00 3rd Qu.:25.00 3rd Qu.:1.0000   
## Max. :2.000 Max. :25.00 Max. :98.00 Max. :1.0000   
## Mar\_Stat Educ Pov\_Stat   
## Min. :1.000 Min. :1.000 Min. :0.0000   
## 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:0.0000   
## Median :3.000 Median :3.000 Median :0.0000   
## Mean :2.399 Mean :2.654 Mean :0.3687   
## 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.:1.0000   
## Max. :4.000 Max. :4.000 Max. :1.0000

## Pre-processing

# preparing the inputs  
y <- YAF.raw$Contraceptive

#Convert the Categorical variables into a factor  
YAF.raw$Contraceptive <- as.factor(YAF.raw$Contraceptive)  
YAF.raw$Mar\_Stat <- as.factor(YAF.raw$Mar\_Stat)  
YAF.raw$Educ<- as.factor(YAF.raw$Educ)  
YAF.raw$Pov\_Stat <- as.factor(YAF.raw$Pov\_Stat )

## Model Building

#Bayesian Logistic Regression Model  
  
prior\_dist <- student\_t(df = 7, location = 0, scale = 2.5)  
bayes\_mod <- stan\_glm(Contraceptive ~ Bar\_Strat + Age + I\_Age + Mar\_Stat + Educ + Pov\_Stat, data = YAF.raw,  
 family = binomial(link = "logit"),   
 prior = prior\_dist, prior\_intercept = prior\_dist,  
 seed = 15689)

##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 0 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 1: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 1: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 1: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 1: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 1: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 1: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 1: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 1: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 1: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 1: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 1:   
## Chain 1: Elapsed Time: 2.605 seconds (Warm-up)  
## Chain 1: 2.127 seconds (Sampling)  
## Chain 1: 4.732 seconds (Total)  
## Chain 1:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).  
## Chain 2:   
## Chain 2: Gradient evaluation took 0 seconds  
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 2: Adjust your expectations accordingly!  
## Chain 2:   
## Chain 2:   
## Chain 2: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 2: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 2: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 2: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 2: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 2: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 2: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 2: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 2: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 2: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 2:   
## Chain 2: Elapsed Time: 2.886 seconds (Warm-up)  
## Chain 2: 2.342 seconds (Sampling)  
## Chain 2: 5.228 seconds (Total)  
## Chain 2:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).  
## Chain 3:   
## Chain 3: Gradient evaluation took 0 seconds  
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 3: Adjust your expectations accordingly!  
## Chain 3:   
## Chain 3:   
## Chain 3: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 3: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 3: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 3: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 3: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 3: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 3: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 3: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 3: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 3: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 3: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 3: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 3:   
## Chain 3: Elapsed Time: 2.84 seconds (Warm-up)  
## Chain 3: 2.138 seconds (Sampling)  
## Chain 3: 4.978 seconds (Total)  
## Chain 3:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).  
## Chain 4:   
## Chain 4: Gradient evaluation took 0.001 seconds  
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Chain 4: Adjust your expectations accordingly!  
## Chain 4:   
## Chain 4:   
## Chain 4: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 4: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 4: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 4: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 4: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 4: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 4: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 4: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 4: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 4: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 4: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 4:   
## Chain 4: Elapsed Time: 2.921 seconds (Warm-up)  
## Chain 4: 2.235 seconds (Sampling)  
## Chain 4: 5.156 seconds (Total)  
## Chain 4:

bayes\_mod

## stan\_glm  
## family: binomial [logit]  
## formula: Contraceptive ~ Bar\_Strat + Age + I\_Age + Mar\_Stat + Educ + Pov\_Stat  
## observations: 4019  
## predictors: 11  
## ------  
## Median MAD\_SD  
## (Intercept) -3.4 0.4   
## Bar\_Strat -0.3 0.1   
## Age 0.1 0.0   
## I\_Age 0.0 0.0   
## Mar\_Stat2 0.9 0.1   
## Mar\_Stat3 0.7 0.1   
## Mar\_Stat4 0.0 0.2   
## Educ2 0.3 0.1   
## Educ3 0.5 0.1   
## Educ4 0.7 0.1   
## Pov\_Stat1 -0.2 0.1   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

posterior\_interval(bayes\_mod, prob = 0.95)

## 2.5% 97.5%  
## (Intercept) -4.174113774 -2.50239902  
## Bar\_Strat -0.470949911 -0.14765291  
## Age 0.083755632 0.14669015  
## I\_Age -0.009594525 0.03277254  
## Mar\_Stat2 0.674620262 1.09605765  
## Mar\_Stat3 0.549184975 0.94408203  
## Mar\_Stat4 -0.421033942 0.36203591  
## Educ2 0.121076436 0.55386778  
## Educ3 0.271933164 0.70046594  
## Educ4 0.463868563 0.94988912  
## Pov\_Stat1 -0.365563772 -0.06347516

summary(residuals(bayes\_mod))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.728482 -0.431374 -0.256146 0.002466 0.495752 0.862694

#We can extract corresponding posterior median estimates using ‘coef’ function and to get a sense for the uncertainty in our estimates we can use the posterior\_interval function to get Bayesian uncertainty intervals. The uncertainty intervals are computed by finding the relevant quantiles of the draws from the posterior distribution. For example, to compute median and 90% intervals we use:  
  
round(coef(bayes\_mod), 2)

## (Intercept) Bar\_Strat Age I\_Age Mar\_Stat2 Mar\_Stat3   
## -3.35 -0.31 0.12 0.01 0.88 0.74   
## Mar\_Stat4 Educ2 Educ3 Educ4 Pov\_Stat1   
## -0.03 0.33 0.48 0.70 -0.22

round(posterior\_interval(bayes\_mod, prob = 0.9), 2)

## 5% 95%  
## (Intercept) -4.05 -2.64  
## Bar\_Strat -0.44 -0.18  
## Age 0.09 0.14  
## I\_Age -0.01 0.03  
## Mar\_Stat2 0.70 1.06  
## Mar\_Stat3 0.58 0.91  
## Mar\_Stat4 -0.36 0.29  
## Educ2 0.15 0.52  
## Educ3 0.31 0.67  
## Educ4 0.50 0.91  
## Pov\_Stat1 -0.34 -0.09

## Model Assessment

set.seed(15689)  
index <- createDataPartition(YAF.raw$Contraceptive,p = 0.7,list = F)  
train <- YAF.raw[index,]  
test <- YAF.raw[-index,]

## *Leave-one-out cross-validation*

#rstanarm supports loo package which implements fast Pareto smoothed leave-one-out cross-validation (PSIS-LOO) (Vehtari, Gelman and Gabry, 2017) to compute expected log predictive density (elpd):  
  
  
(loo1 <- loo(bayes\_mod, save\_psis = TRUE))

##   
## Computed from 4000 by 4019 log-likelihood matrix  
##   
## Estimate SE  
## elpd\_loo -2646.9 16.7  
## p\_loo 10.9 0.3  
## looic 5293.8 33.4  
## ------  
## Monte Carlo SE of elpd\_loo is 0.1.  
##   
## All Pareto k estimates are good (k < 0.5).  
## See help('pareto-k-diagnostic') for details.

#Above we see that PSIS-LOO result is reliable as all Pareto k estimates are small (k< 0.5) (Vehtari, Gelman and Gabry, 2017; Vehtari et al., 2019).

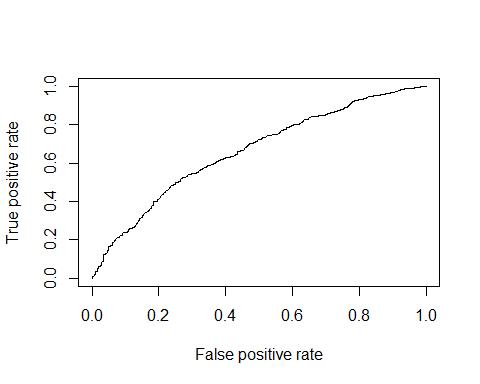
#Confusion Matrix  
  
library(caret)  
prob <- predict(bayes\_mod, newdata=test, type="response")   
pred <- ifelse(prob > 0.5, "1", "0")  
  
  
#Confusion matrix and Statistics  
accuracy <- table(pred, test$Contraceptive)  
accuracy

##   
## pred 0 1  
## 0 509 272  
## 1 164 259

sum(diag(accuracy))/sum(accuracy)

## [1] 0.6378738

#Area under ROC  
  
library(ROCR)  
prob <- predict(bayes\_mod , newdata=test , type="response")  
pred <- prediction(prob, test$Contraceptive)  
perf <- performance(pred, 'tpr', 'fpr')  
plot(perf)



auc <- performance(pred, measure = "auc") #Area Under ROC   
auc <- auc@y.values[[1]]  
auc

## [1] 0.6617921