**Created by Bradley Velasquez on Friday, November 11, 2022 at 11:49 AM**
https://umich.enterprise.slack.com/files/U02CR94U41E/F04AYP9SKRP/model_script.r

```
##############LIBRARY AND FUNCTIONS##############
#For cleaning and reading data
library(tidyverse)
library(caret)
library(themis)
library(EZtune)
library(MLmetrics)

#
# library(tidymodel)
#for plotting
theme_set(theme_bw())


#change wd, and import data
# setwd("~/Documents/Box Sync/Statistics Master/Fall 2022/STATS504/hw5")
setwd("/Users/brad/Downloads/hw5")
df <- read.csv("data/nuMoM2bsubset.csv")
# load("brad_models.RData") # Saved my global enviornment just in case...

############ Groups 4, 9, 14 outcome: dv.hypertension1

null_outcomes <-c("dv.diabetes1",
                  "dv.v1epdstotal",
                  "dv.gestweeks",
                  "dv.preeclampsia")

df <- df[,!(names(df) %in% null_outcomes)]

######## DATA CLEANING #########
test_df = df %>% summarise(across(everything(), list(min,max)))
test_df = t(test_df)

#Convert booleans to "TRUE"/"FALSE"
df$emosupport <- df$emosupport == 1
df$financialsupport <- df$financialsupport==1
df$prenatalsupport <- df$prenatalsupport == 1
df$financialsupport <- df$financialsupport ==1
df$deliverysupport <- df$deliverysupport ==1
df$exercise <- df$exercise == 1
df$dv.hypertension1 <- df$dv.hypertension1==1
df$kidney1 <- df$kidney1==1
df$lupus1 <- df$lupus1 == 1
df$collagen1 <- df$collagen1==1
df$crohns1 <- df$crohns1 == 1
df$pcos1 <- df$pcos1 == 1


#Three level factors....
df$familypreeclampsia <- as.factor(df$familypreeclampsia)
df$bornearly <- as.factor(df$bornearly)



#Higher level factors....
df$worryfambaby <- as.factor(df$worryfambaby)
```

```r
df$worryhealthcare <- as.factor(df$worryhealthcare)
df$worrysymptoms <- as.factor(df$worrysymptoms) #what are the levels for this? There should be a codebook.
df$discrimination <- as.factor(df$discrimination)
df$race <- as.factor(df$race)

# skimr::skim(df)
############SUMMARY TABLE###########
skimmed_df = skimr::skim(df)
skimmed_df$n_missing = NULL
skimmed_df$complete_rate= NULL
skimmed_df$numeric.hist = NULL
skimmed_df$factor.ordered = NULL

#For factors
skimmed_df$factor.top_counts[8:18] <- skimmed_df$logical.count[8:18]
skimmed_df$factor.top_counts[8:18] <- skimmed_df$logical.count[8:18]
skimmed_df$factor.n_unique[8:18] <- 2
skimmed_df$numeric.mean[8:18] <- skimmed_df$logical.mean[8:18]
skimmed_df$numeric.mean <- round(skimmed_df$numeric.mean, digits= 4)
skimmed_df$logical.mean <- NULL
skimmed_df$logical.count = NULL
skimmed_df$factor.top_counts[19:26] <- paste0("
(",skimmed_df$numeric.p25[19:26],",",skimmed_df$numeric.p75[19:26],")")
skimmed_df$numeric.sd <- NULL
skimmed_df$numeric.p0 <- NULL
skimmed_df$numeric.p25 <- NULL
skimmed_df$numeric.p50 <- NULL
skimmed_df$numeric.p75 <- NULL
skimmed_df$numeric.p100 <- NULL
# write.csv(skimmed_df, "data/baseline.csv")


rm(test_df, null_outcomes) #drop unused items

######IMPUTE VALUES########
colSums(df == 0)
#age
#psstotla
#ssqmean
# prepreglbs

df$age[df$age==0] <- mean(df$age[df$age!=0])
df$prepreglbs[df$prepreglbs==0] <- mean(df$prepreglbs[df$prepreglbs!=0])

########TEST TRAIN###############
set.seed(1123)
size = floor(0.3*dim(df))

id = sample(c(1:7934),replace=F, size = floor(0.3*dim(df))[1])
train<-df[-id,]
test<-df[id,]

# write.csv(test,"test_df.csv")
# write.csv(train,"train_df.csv")

#########ENCODE TRAIN#########
dmy <- dummyVars(" ~ .", data = train)
train_hot <- data.frame(predict(dmy, newdata = train))

train_hot_X = train_hot[,!(names(train_hot) %in% c("dv.hypertension1FALSE","dv.hypertension1TRUE"))]
```

```

#Outcome variable needs to have a valid name. Use make.names() or use
train_hot_y = factor(train_hot$dv.hypertension1TRUE,
                     levels = c(1,0),
                     labels = c("yes","no"))



#Combined data frame .....
train_hot_X_y<- train_hot_X
train_hot_X_y$dv.hypertension <- train_hot_y


#####UP SAMPLE TRAINING DAT########
#minority class has 50% of observations as majoirty
train_up50<- smote(train_hot_X_y, var ="dv.hypertension", over=0.5, k=10)
table(train_up50$dv.hypertension)
#majority class has 100% of observations as majority
train_up100<- smote(train_hot_X_y, var ="dv.hypertension",  over=1, k=10)
table(train_up100$dv.hypertension)

#Export
# write.csv(train_up50, "data/train_one_hot_50.csv")
# write.csv(train_up100, "data/train_one_hot_100.csv")




#########columns with no variance in training data ############
X_no_var = nearZeroVar(train_hot_X_y)
X_no_var_names = names(train_hot_X_y)[X_no_var]
X_no_var_names = X_no_var_names[-c(1,37)] #hypertension has low variance, as does race == native

#Drop columns with no variance....
train_hot_X_y <- train_hot_X_y[,!(names(train_hot_X) %in% X_no_var_names)]
train_up50 <- train_up50[,!(names(train_up50) %in% X_no_var_names)]
train_up100 <- train_up100[,!(names(train_up100) %in% X_no_var_names)]

# Check the distribution of these variables....
# train_hot_X[,X_no_var_names] %>% skimr::skim()

############DOWNSAMPLE#############
set.seed(1123)
train_hot_down = downSample(x=train_hot_X_y,
                            y=train_hot_X_y$dv.hypertension)
train_hot_down$Class <- NULL

##############ENCODE TEST##########
dmy <- dummyVars(" ~ .", data = test)
test_hot <- data.frame(predict(dmy, newdata = test))

test_hot_X = test_hot[,!(names(test_hot) %in% c("dv.hypertension1FALSE","dv.hypertension1TRUE"))]

#Outcome variable needs to have a valid name. Use make.names() or use
test_hot_y = factor(test_hot$dv.hypertension1TRUE,
                    levels = c(1,0),
                    labels = c("yes","no"))

#combine into one data frame....
test_hot_X_y = test_hot_X
```

```r
test_hot_X_y$dv.hypertension = test_hot_y

#Drop columns with no variance
test_hot_X_y <- test_hot_X_y[,!(names(test_hot_X) %in% X_no_var_names)]


########CLEAN WORKSPACE############
rm(list = c(#"test_hot_X_sub",
            "test_hot_X",
            "test_hot",
            "test_hot_y",
            # "train_hot_X_sub",
            "train_hot_X",
            "train_hot",
            "train_hot_y",
            "dmy"))

############ HYPERPARAMETER TUNING##############

models <- caret::modelLookup() #what models are in the caret package?

########## Adaboost.M1############

#!!!!!!!!!!!!!! WARNING THE FOLLOWING CHUNKS TAKE ~40 Minutes to run!!!!!!!!!!!!!!!!

fitGrid_ada <- expand.grid(mfinal = c(1,6,9,100),
                           # mfinal = (1:3)*3,
                           # maxdepth = c(1:3),
                           maxdepth = c(1,2,4),
                           coeflearn = c("Breiman"))

fitControl_ada <- trainControl(method = "repeatedcv",
                               repeats = 5,
                               classProbs = T,
                               # summaryFunction = twoClassSummary,
                               summaryFunction = prSummary)
#on up sampled

# using the adaboost.m1 package....
set.seed(1123)
start_time = Sys.time()
ada.mod <- train(x=train_hot_X_y[,-48],
                 y= train_hot_X_y$dv.hypertension,
                 method = 'AdaBoost.M1',
                 trControl = fitControl_ada,
                 tuneGrid = fitGrid_ada,
                 metric = "AUC",
                 verbose = TRUE)
total_time <- Sys.time() - start_time
total_time

# Upsampled to be 50% majority class
set.seed(1123)
start_time <- Sys.time()
ada50.mod <- train(x=train_up50[,-48],
                   y= train_up50$dv.hypertension,
                   method = 'AdaBoost.M1',
                   trControl = fitControl_ada,
                   tuneGrid = fitGrid_ada,
                   metric = "AUC",
```

```
                   verbose = TRUE)
total_time <- Sys.time() - start_time
total_time


# upsampled to Matched classes - change metric to ROC.
set.seed(1123)
total_time <- Sys.time()
ada100.mod <- train(x=train_up100[,-48],
                    y= train_up100$dv.hypertension,
                    method = 'AdaBoost.M1',
                    trControl = fitControl_ada,
                    tuneGrid = fitGrid_ada,
                    metric = "ROC",
                    verbose = TRUE)
total_time <- Sys.time() - start_time
total_time

# On downsampled - use ROC
set.seed(1123)
start_time <- Sys.time()
adadown.mod <- train(x=train_hot_down[,-48],
                     y= train_hot_down$dv.hypertension,
                     method = 'AdaBoost.M1',
                     trControl = fitControl_ada,
                     tuneGrid = fitGrid_ada,
                     metric = "ROC",
                     verbose = TRUE)
total_time <- Sys.time() - start_time
total_time


############# USING GBM PACKAGE ############3

# set.seed(1123)
# start_time <- Sys.time()
# ada.mod <- train(x=train_hot_X_y[,-48],
#                    y= train_hot_X_y$dv.hypertension,
#                    distribution = 'adaboost',
#                    method="gbm",
#                    trControl = fitControl_ada,
#                    tuneGrid = fitGrid_ada,
#                    metric = "AUC",
#                    verbose = TRUE)
# total_time <- Sys.time() - start_time
# total_time

# set.seed(1123)
# start_time <- Sys.time()
# ada50.mod <- train(x=train_up50[,-48],
#                     y= train_up50$dv.hypertension,
#                     distribution = 'adaboost',
#                     method="gbm",
#                     trControl = fitControl_ada,
#                     tuneGrid = fitGrid_ada,
#                     metric = "AUC",
#                     verbose = TRUE)
# total_time <- Sys.time() - start_time
# total_time

```

```
296
297 # set.seed(1123)
298 # total_time <- Sys.time()
299 # ada100.mod <- train(x=train_up100[,-83],
300 #                     y= train_up100$dv.hypertension,
301 #                     distribution = 'adaboost',
302 #                     method="gbm",
303 #                     trControl = fitControl_ada,
304 #                     tuneGrid = fitGrid_ada,
305 #                     metric = "ROC",
306 #                     verbose = TRUE)
307 # total_time <- Sys.time() - start_time
308 # total_time
309
310
311 #Fit downsampled data on finer grid...
312 # fitGrid_ada <-  expand.grid(interaction.depth = c(1, 3, 6, 9),
313 #                             n.trees = c(1,10,20,50,100),
314 #                             shrinkage = seq(.0005, .05,.0005),
315 #                             n.minobsinnode = 10)
316 #
317 # fitControl_ada <- trainControl(method = "repeatedcv",
318 #                                repeats = 5,
319 #                                classProbs = T,
320 #                                summaryFunction = twoClassSummary)
321 #on up sampled
322 # set.seed(1123)
323 # start_time <- Sys.time()
324 # adadown.mod <- train(x=train_hot_down[,-83],
325 #                   y= train_hot_down$dv.hypertension,
326 #                   distribution = 'adaboost',
327 #                   method="gbm",
328 #                   trControl = fitControl_ada,
329 #                   tuneGrid = fitGrid_ada,
330 #                   metric = "ROC",
331 #                   verbose = TRUE)
332 # total_time <- Sys.time() - start_time
333 # total_time
334
335
336
337 ########PREDICTION#########
338 # Class Predictions
339 test_predada <- predict(object = ada.mod,newdata = test_hot_X_y[,-48])
340 test_predada50 <- predict(object = ada50.mod,newdata = test_hot_X_y[,-48])
341 test_predada100 <- predict(object = ada100.mod,newdata = test_hot_X_y[,-48])
342 test_predadadown <- predict(object = adadown.mod,newdata = test_hot_X_y[,-48])
343
344 # Probabilities
345 test_predada_p <- predict(object = ada.mod,newdata = test_hot_X_y[,-48],type="prob")
346 test_predada50_p <- predict(object = ada50.mod,newdata = test_hot_X_y[,-48],type = "prob")
347 test_predada100_p <- predict(object = ada100.mod,newdata = test_hot_X_y[,-48],type = "prob")
348 test_predadadown_p <- predict(object = adadown.mod,newdata = test_hot_X_y[,-48],type ="prob")
349
350 ########PRAUC#######
351 ada_prauc = MLmetrics::PRAUC(test_predada_p$yes, test_hot_X_y$dv.hypertension)
352 ada_prauc
353
354 adaup_prauc = MLmetrics::PRAUC(test_predada100_p$yes, test_hot_X_y$dv.hypertension)
355 adaup_prauc
```

```
356
357 ######## AUROC #######
358 # library(pROC)
359 # ada.roc <- roc(test_hot_X_y$dv.hypertension, test_predada_p$yes)
360 # # plot(ada.roc, print.thres="best", print.thres.best.method="closest.topleft")
361 # ada50.roc <- roc(test_hot_X_y$dv.hypertension, test_predada50_p$yes)
362 # ada100.roc <- roc(test_hot_X_y$dv.hypertension, test_predada100_p$yes)
363 #
364
365 # plot(ada50.roc, print.thres="best", print.thres.best.method="closest.topleft")
366 # result.coords <- coords(ada.roc, "best", best.method="closest.topleft", ret=c("ppv","tpr"))
367 # print(result.coords)#to get threshold and accuracy
368
369
370 #########CONFUSION METRICS############
371 conf_matada = table("truth"=test_hot_X_y$dv.hypertension,"pred"= test_predada)
372 conf_matada=confusionMatrix(conf_matada, mode = "everything", positive = "yes")
373 conf_matada
374
375 conf_matada50 = table("truth"=test_hot_X_y$dv.hypertension,"pred"= test_predada50)
376 conf_matada50=confusionMatrix(conf_matada50, mode = "everything", positive = "yes")
377 conf_matada50
378
379 conf_matada100 = table("truth"=test_hot_X_y$dv.hypertension,"pred"= test_predada100) #Adaboost looks a bit bette
380 conf_matada100=confusionMatrix(conf_matada100, mode = "everything", positive ="yes")
381 conf_matada100
382
383 conf_matadadown= table("truth"=test_hot_X_y$dv.hypertension,"pred"= test_predadadown) #Adaboost looks a bit
    better
384 conf_matadadown=confusionMatrix(conf_matadadown, mode = "everything", positive ="yes")
385 conf_matadadown
386
387 metrics_df = data.frame(ada = conf_matada$byClass,
388                         ada50 = conf_matada50$byClass,
389                         ada.tune100 = conf_matada100$byClass)
390
391 # save.image(file='brad_models.RData')
392 # load("brad_svm_env.RData")
393
394 ############PLOTS ###############
395 hyp_race_p = df %>% group_by(race) %>% summarise(p = mean(dv.hypertension1)) %>%
396   ggplot(aes(x=reorder(race, -p), y = p, fill = race))+
397   geom_bar(stat='identity')+
398   scale_y_continuous(labels = scales::percent)+
399   xlab("Race")+
400   ylab("Perc. of Women w/ Hypertension")+
401   guides(fill="none")
402
403
404 ggplot(data = df, aes(x=prepreglbs, fill = race, group = race))+
405   # geom_density(alpha=0.4)+
406   geom_histogram(aes(y=stat(density)))+
407   # scale_y_continuous(labels = scales::percent)+
408   # xlab("Frequency")+
409   # scale_y_continuous(labels = percent )
410   facet_grid(rows = vars(race))
411   # ylab("Perc. w/ Hypertension")+
412   # guides(fill="none")
413
414
```

```
ggplot(data=df)+
  geom_jitter(aes(x=diastolic,
                  y = systolic,
                  col = dv.hypertension1),
              alpha=0.5)+
  facet_wrap(~dv.hypertension1)+
  scale_x_continuous(sec.axis = sec_axis(~ . ,
                                         name = "Has Hypertension",
                                         breaks = NULL,
                                         labels = NULL))+
  # scale_y_continuous(labels = scales::percent)+
  xlab("Systolic Blood Pressure At First Visit")+
  ylab("Diastolic Blood Pressure At First Visit")+
  guides(col="none")
```