Final_Emote_prediction_SVM_time_domain_test22222.R

mac

Wed May 1 22:09:18 2019

```
library(tidyverse)
library(caret)
#install.packages("DAAG")
library(DAAG)


mydat <- read.csv(here('data', 'Preprocessing_data_outputs', 'Paper', 'data_out.csv'))%>%
  filter(threshold == 250 & winds == 2)%>%
  select(-winds, - threshold, -index)
 dim(mydat)
```

```
## [1] 2153   15
```

```
#mydat2 <- mydat[ complete.cases(mydat$SDNN) , ]

mydat1 <- mydat[ complete.cases(mydat) , ]
rmind <- which((mydat1$Stress>=5) & (mydat1$Y == "Control"))
mydat2 <- mydat1[-rmind, ]
dim(mydat2)
```

```
## [1] 129  15
```

```
#142
str(mydat2)
```

```
## 'data.frame':    129 obs. of  15 variables:
##  $ ID     : int  201 201 201 201 201 202 202 207 207 207 ...
##  $ when   : Factor w/ 1901 levels "","1/1/18 1:25",..: 37 48 81 89 878 1112 1148 999 1008 1031 ...
##  $ Event  : Factor w/ 1913 levels "","1/1/18 1:25",..: 37 48 81 89 883 1122 1154 1005 1014 1037 ...
##  $ Stress : int  2 1 3 2 2 2 1 1 1 4 ...
##  $ Y      : Factor w/ 3 levels "","Control","Episode": 2 2 3 2 3 2 3 2 2 3 ...
##  $ SDNN   : num  56.7 27.5 47.8 55.2 47 ...
##  $ SDANN  : num  30.82 9.6 13.64 5.58 19.47 ...
##  $ SDNNIDX: num  45.9 29 40.1 46.8 43.6 ...
##  $ pNN50  : num  19.4 11.4 12.1 19.3 13.6 ...
##  $ SDSD   : num  48.8 38 38 55.9 39 ...
##  $ rMSSD  : num  48.8 38 38 55.9 39 ...
##  $ IRRR   : num  78.1 31.3 62.5 78.1 62.5 ...
##  $ MADRR  : num  31.3 15.6 15.6 31.3 15.6 ...
##  $ TINN   : num  143 63.8 109 134.5 109.1 ...
##  $ HRVi   : num  9.15 4.09 6.98 8.61 6.98 ...
```

```
table(mydat2$Y)
```

```
##
##         Control Episode
##       0     110      19
```

```
#    Control Episode
#     0     110      19
```

```r
#creat within subject z scores
mydat3 <- mydat2
temp =  unique(mydat2$ID)
m = length(temp)
for (i in 1:m){
tempdat <- mydat2[which(mydat2$ID == temp[i]), ]
mydat3[which(mydat2$ID == temp[i]), 6:15] <- scale(tempdat[ , 6:15])
}
mydat3 <- mydat3[complete.cases(mydat3), ]
table(mydat3$Y)
```

```
##
##         Control Episode
##       0     107      19
```

```r
#         Control Episode
#      0      107      19

#dataset = mydat2
#folder = subdata$Fold3

library(e1071)
# make predictions
library(ROSE)
library(caret)
library(caretEnsemble)
#mydat2, subdata$Fold1
modelfit <- function(dataset, folder){
data <- dataset[-folder, ]
mytest <- dataset[folder, ]
##Generate balanced dataset
train.m.bal<-ovun.sample(Y ~ ., data=data, method="both",p=0.6, seed=1342)$data
names(train.m.bal)
table(train.m.bal$Y)
xtrain <- train.m.bal[, 6:15]
xtrain <- as.matrix(xtrain)
ytrain <- train.m.bal$Y
#levels(ytrain) <- list(no="0", yes="1")
#y.train <- as.numeric(train.m.bal$Y)
#y.train[which(train.m.bal$Y == "Control")] <- 0
#y.train[which(train.m.bal$Y == "Episode")] <- 1
table(ytrain) # 0 is control, 1 is episode
ytrain <- as.factor(ytrain)
xtrain <- data.frame(xtrain)
datatrain <- cbind(ytrain, xtrain)
#datatrain$ytrain <- as.factor(datatrain$ytrain)
#control <- trainControl(method="repeatedcv", number=5, repeats=3)
fit<- train(ytrain ~ ., data= datatrain,method="svmPoly")
svmFit <- train(ytrain ~ ., data=datatrain, method="svmPoly", trControl = trainControl( classProbs=T))

#svmFit <- train(ytrain ~ ., data=datatrain, method="svmPoly", trControl = trainControl(method = "repea


#fit2<- train(as.factor(y.train) ~ ., data= data.train,method="gbm")
```

```
#models <- list(svm =fit)
#predict(fit, newdata= x.test, type = "prob")
#varImp(fit)
xtest <- mytest[, 6:15]
xtest <- data.frame(xtest)
ytest <- as.numeric(mytest$Y)-1
table(ytest)
yhat6 = predict(fit, xtest)
table(yhat6)
yhat6 <- as.numeric(yhat6)
yhat6 <- as.factor(yhat6)
ytest <-as.factor(ytest)
temp <- caret::confusionMatrix(yhat6,ytest)


yhat62 = predict(svmFit, xtest)
table(yhat62)
yhat62 <- as.numeric(yhat62)
yhat62 <- as.factor(yhat62)
temp2 = caret::confusionMatrix(yhat62, ytest)

#predict(fit, testX = xtest, type="prob")

#predict(svmFit, testX = xtest, type="prob")
#extractProb(models, x.test)


#result <- c(temp$overall[1], temp$byClass[1:2]) #<-can change threshold if you want

result <- c(temp$overall[1], temp$byClass[1:2],temp2$overall[1], temp2$byClass[1:2]) #<-can change thre
return(result)
}

#make predictions
#involve random sampling, need to set the seed

set.seed(9)
nfolds=4
subdata<-createFolds(mydat3$Y, nfolds)
t1 <- modelfit(mydat3, subdata$Fold1)
t1
```

```
##    Accuracy Sensitivity Specificity    Accuracy Sensitivity Specificity
##   0.5937500   0.6296296   0.4000000   0.5937500   0.6296296   0.4000000
```

```
t2 <- modelfit(mydat3, subdata$Fold2)
```

```
## maximum number of iterations reached 0.0001524629 -0.0001523796maximum number of iterations reached
```

```
t2
```

```
##    Accuracy Sensitivity Specificity    Accuracy Sensitivity Specificity
##   0.7096774   0.6923077   0.8000000   0.3870968   0.3076923   0.8000000
```

```
t3 <- modelfit(mydat3, subdata$Fold3)
```

```
## maximum number of iterations reached 0.002030985 -0.002025171maximum number of iterations reached 0.
```

```
t3
```

```
##    Accuracy Sensitivity Specificity    Accuracy Sensitivity Specificity
##   0.5483871   0.4814815   1.0000000   0.6451613   0.5925926   1.0000000
```

```
t4 <- modelfit(mydat3, subdata$Fold4)
```

```
## maximum number of iterations reached 0.0001085587 -0.0001082045maximum number of iterations reached 1
```

```
t4
```

```
##    Accuracy Sensitivity Specificity    Accuracy Sensitivity Specificity
##   0.6875000   0.7037037   0.6000000   0.7187500   0.7407407   0.6000000
```

```
colMeans(rbind(t1,t2,t3,t4))
```

```
##    Accuracy Sensitivity Specificity    Accuracy Sensitivity Specificity
##   0.6348286   0.6267806   0.7000000   0.5861895   0.5676638   0.7000000
```