library(openintro)

library(tidyverse)

library(rsample)

library(doParallel)

library(caret)

library(iml)

library(patchwork)

library(haven)

library(foreign)

zz <- file.path("U:","My Documents")

Aloc <- file.path(zz,"A.sav")

A <- read.spss(Aloc)

A =na.omit(A)

EE <- A[c("T", "EE", "IS","SYO", "COG", "DIS","Age")] %>% na.omit

EE\_split <- initial\_split(EE, prop = 0.2)

EE\_train <- training(EE\_split)

EE\_test <- testing(EE\_split)

cl <- makePSOCKcluster(detectCores())

registerDoParallel(cl)

modelNnet <- caret::train(

EE~ .,

data = EE\_train,

method = "nnet",

preProcess = c("center", "scale"),

metric = "MAE",

trControl = trainControl(method = "cv",number=10),

tuneLength = 10,

linout = TRUE,

trace = FALSE)

modelNnet$results %>% arrange(MAE) %>% round(5) %>% slice(1:10)

modelNnet$bestTune

EE\_sd <- EE\_train %>% summarise(sd(EE)) %>% round(2)

(summarise(modelNnet$results, min(MAE)) / EE\_sd) %>% round(2) %>%

str\_c("MAE / EE\_SD: ", .) %>% print

predictor <- Predictor$new(modelNnet, data = EE\_train)

imp <- FeatureImp$new(predictor, loss = "mae", compare = "ratio")

plot(imp)

imp$results

pdp <- FeatureEffects$new(predictor,

features = predictor$model$coefnames[1:6],

method = "pdp+ice")

plot(pdp)

min\_val <- predictor$model$trainingData[,predictor$model$coefnames] %>%

summarize\_each(min) %>% as.numeric

for(i in 1:length(min\_val)){pdp <- FeatureEffects$new(predictor,

feature = predictor$model$coefnames[i],

method = "pdp+ice",center.at = min\_val[i])$plot()

plot(pdp)}

ale <- FeatureEffects$new(predictor,

features = predictor$model$coefnames[1:6],

method = "ale")

plot(ale)

interact <- Interaction$new(predictor)

plot(interact)

shapley\_f <- function(x.interest){

shapley <- Shapley$new(predictor,

x.interest = x.interest)

plot(shapley) +

theme(axis.text = element\_text(size = 10))}

shapley\_f(x.interest = EE\_train[1,]) + shapley\_f(x.interest = EE\_train[2,]) +

shapley\_f(x.interest = EE\_train[3,]) + shapley\_f(x.interest = EE\_train[4,])

lime\_f <- function(x.interest){

lime <- LocalModel$new(predictor,

x.interest = x.interest,

k = ncol(EE\_train) - 1)

plot(lime)}

lime\_f(x.interest = EE\_train[1,]) + lime\_f(x.interest = EE\_train[2,]) +

lime\_f(x.interest = EE\_train[3,]) + lime\_f(x.interest = EE\_train[4,])