Finalmodel\_OD

library(keras)  
library(tensorflow)  
library(rlang)  
library(tidyverse)

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --  
## v dplyr 1.1.2 v readr 2.1.4  
## v forcats 1.0.0 v stringr 1.5.0  
## v ggplot2 3.4.2 v tibble 3.2.1  
## v lubridate 1.9.2 v tidyr 1.3.0  
## v purrr 1.0.1   
## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x purrr::%@%() masks rlang::%@%()  
## x dplyr::filter() masks stats::filter()  
## x purrr::flatten() masks rlang::flatten()  
## x purrr::flatten\_chr() masks rlang::flatten\_chr()  
## x purrr::flatten\_dbl() masks rlang::flatten\_dbl()  
## x purrr::flatten\_int() masks rlang::flatten\_int()  
## x purrr::flatten\_lgl() masks rlang::flatten\_lgl()  
## x purrr::flatten\_raw() masks rlang::flatten\_raw()  
## x purrr::invoke() masks rlang::invoke()  
## x dplyr::lag() masks stats::lag()  
## x purrr::splice() masks rlang::splice()  
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(dplyr)  
library(gapminder)  
library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift  
##   
## The following object is masked from 'package:tensorflow':  
##   
## train

library(xgboost)

##   
## Attaching package: 'xgboost'  
##   
## The following object is masked from 'package:dplyr':  
##   
## slice

library(forcats)  
library(mice)

##   
## Attaching package: 'mice'  
##   
## The following object is masked from 'package:stats':  
##   
## filter  
##   
## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(vctrs)

##   
## Attaching package: 'vctrs'  
##   
## The following object is masked from 'package:dplyr':  
##   
## data\_frame  
##   
## The following object is masked from 'package:tibble':  
##   
## data\_frame

library(naniar)  
library(yardstick)

## For binary classification, the first factor level is assumed to be the event.  
## Use the argument `event\_level = "second"` to alter this as needed.  
##   
## Attaching package: 'yardstick'  
##   
## The following objects are masked from 'package:caret':  
##   
## precision, recall, sensitivity, specificity  
##   
## The following object is masked from 'package:readr':  
##   
## spec  
##   
## The following object is masked from 'package:keras':  
##   
## get\_weights

library(forcats)  
library(Boruta)  
library(ROSE)

## Loaded ROSE 0.0-4

library(imbalance)  
library(rnn)  
library(ROCR)  
library(SHAPforxgboost)  
library(randomForest)

## randomForest 4.6-14  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine  
##   
## The following object is masked from 'package:ggplot2':  
##   
## margin

library(mlbench)  
library(ggplot2)  
library(data.table)

##   
## Attaching package: 'data.table'  
##   
## The following objects are masked from 'package:lubridate':  
##   
## hour, isoweek, mday, minute, month, quarter, second, wday, week,  
## yday, year  
##   
## The following objects are masked from 'package:dplyr':  
##   
## between, first, last  
##   
## The following object is masked from 'package:purrr':  
##   
## transpose  
##   
## The following object is masked from 'package:rlang':  
##   
## :=

library(here)

## here() starts at R:/atai-22-075/Andy's Thesis/MLoverdose in general/GeneralodML

library(pROC)

## Type 'citation("pROC")' for a citation.  
##   
## Attaching package: 'pROC'  
##   
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(ROCR)

inferencedataset1 <- read\_csv("U:/Andy's Thesis/MLoverdose in general/GeneralodML/inferencedataset1.csv" , show\_col\_types = FALSE)

inferencedataset2 <- inferencedataset1[sample(which(inferencedataset1$od\_case != "1"), 36679), ]  
table(inferencedataset2$od\_case)

##   
## 0   
## 36679

inferencedataset3 <- inferencedataset1 %>%   
 filter(od\_case == 1)  
  
table(inferencedataset3$od\_case)

##   
## 1   
## 36679

full <- rbind(inferencedataset2, inferencedataset3)

full <- full %>%   
 mutate(od\_case = recode(od\_case, "0" = "No", "1" = "Yes"))  
  
  
  
full = full %>% mutate\_if(is.character, as.factor)  
  
dim(full)

## [1] 73358 49

nrow(full)

## [1] 73358

table(full$od\_case)

##   
## No Yes   
## 36679 36679

class(full$od\_case)

## [1] "factor"

grid\_default <- expand.grid(  
 .mtry = c(1:7),  
 .splitrule="gini",  
 .min.node.size=c(1)  
)  
  
train\_control <- caret::trainControl(  
 method = "cv",  
 number = 10,  
 search = "grid",  
 summaryFunction = twoClassSummary,  
 returnResamp = "none",  
 classProbs = TRUE,  
 savePredictions = TRUE,  
 verboseIter = FALSE,  
 allowParallel = TRUE  
)

rf1 = caret::train(od\_case ~ .,   
 data = full,  
 trControl = train\_control,  
 method = "ranger",  
 tuneGrid = grid\_default,  
 verbose = TRUE  
 )

## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not  
## in the result set. ROC will be used instead.

print(rf1)

## Random Forest   
##   
## 73358 samples  
## 48 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 66022, 66022, 66022, 66023, 66023, 66022, ...   
## Resampling results across tuning parameters:  
##   
## mtry ROC Sens Spec   
## 1 0.9020121 0.9211537 0.7477034  
## 2 0.9136307 0.9066768 0.8069468  
## 3 0.9155685 0.8983069 0.8196788  
## 4 0.9159830 0.8970255 0.8218871  
## 5 0.9154943 0.8974890 0.8216418  
## 6 0.9149067 0.8981433 0.8211783  
## 7 0.9142224 0.8981706 0.8206875  
##   
## Tuning parameter 'splitrule' was held constant at a value of gini  
##   
## Tuning parameter 'min.node.size' was held constant at a value of 1  
## ROC was used to select the optimal model using the largest value.  
## The final values used for the model were mtry = 4, splitrule = gini  
## and min.node.size = 1.

confusionMatrix(rf1)

## Cross-Validated (10 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction No Yes  
## No 44.9 8.9  
## Yes 5.1 41.1  
##   
## Accuracy (average) : 0.8595

confusionMatrix(rf1$pred$pred, rf1$pred$obs)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 231700 49160  
## Yes 25053 207593  
##   
## Accuracy : 0.8555   
## 95% CI : (0.8545, 0.8564)  
## No Information Rate : 0.5   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.711   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9024   
## Specificity : 0.8085   
## Pos Pred Value : 0.8250   
## Neg Pred Value : 0.8923   
## Prevalence : 0.5000   
## Detection Rate : 0.4512   
## Detection Prevalence : 0.5469   
## Balanced Accuracy : 0.8555   
##   
## 'Positive' Class : No   
##

# saveRDS(rf1, file = "RF\_OD")