BST 263 Final Project: XGBoost model

Bina Choi

2023.05.07

Libraries and functions

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(xgboost)
```

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

```
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-6
```

```
library(dplyr)
library(class)
library(caret)
```

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

```
library(e1071)
library(caret)
library(ROCR)
```

Load in the data, imputation, and train/test split

```
dat <- read.csv('brfss_final.csv')
outcome <- data.frame(dat$X,dat$MICHD,dat$CVDINFR4,dat$CVDCRHD4)
outcome %>% group_by(dat.MICHD) %>% summarise(count=n())
```

```
## # A tibble: 2 x 2
##   dat.MICHD count
##       <int> <int>
## 1         1 14580
## 2         2 14580
```

```
outcome %>% group_by(dat.CVDINFR4) %>% summarise(count=n())
```

```
## # A tibble: 4 x 2
##   dat.CVDINFR4 count
##          <int> <int>
## 1            1  9188
## 2            2 19802
## 3            7   160
## 4            9    10
```

```
outcome %>% group_by(dat.CVDCRHD4) %>% summarise(count=n())
```

```
## # A tibble: 4 x 2
##   dat.CVDCRHD4 count
##          <int> <int>
## 1            1  9729
## 2            2 18874
## 3            7   550
## 4            9     7
```

```
## remove the ones that responded don't know & not sure in CVDINFR4 & CVDCRHD4
dat <- dat[-which(dat$CVDINFR4 == 7 | dat$CVDINFR4 == 9),]
dat <- dat[-which(dat$CVDCRHD4 == 7 | dat$CVDCRHD4 == 9),]

# remove columns that has only 1 value for all rows
dat <- dat[ , -which(names(dat) %in% c("MEDSHEPB","TOLDCFS", "HAVECFS", "WORKCFS"))]
```

Drop columns with more than 5% data missing, impute the rest using KNN

```r
# convert outcome variables
dat$MICHD <- factor(2-dat$MICHD)
dat$CVDINFR4 <- factor(2-dat$CVDINFR4)
dat$CVDCRHD4 <- factor(2-dat$CVDCRHD4)

# i believe X is the index column, not needed
# remove weights
dat <- dat[, !colnames(dat) %in% c('X', 'LLCPWT2', 'LLCPWT', 'CLLCPWT','STRWT','WT2RAKE')]
threshold <- .05
ncol(dat) # 190
```

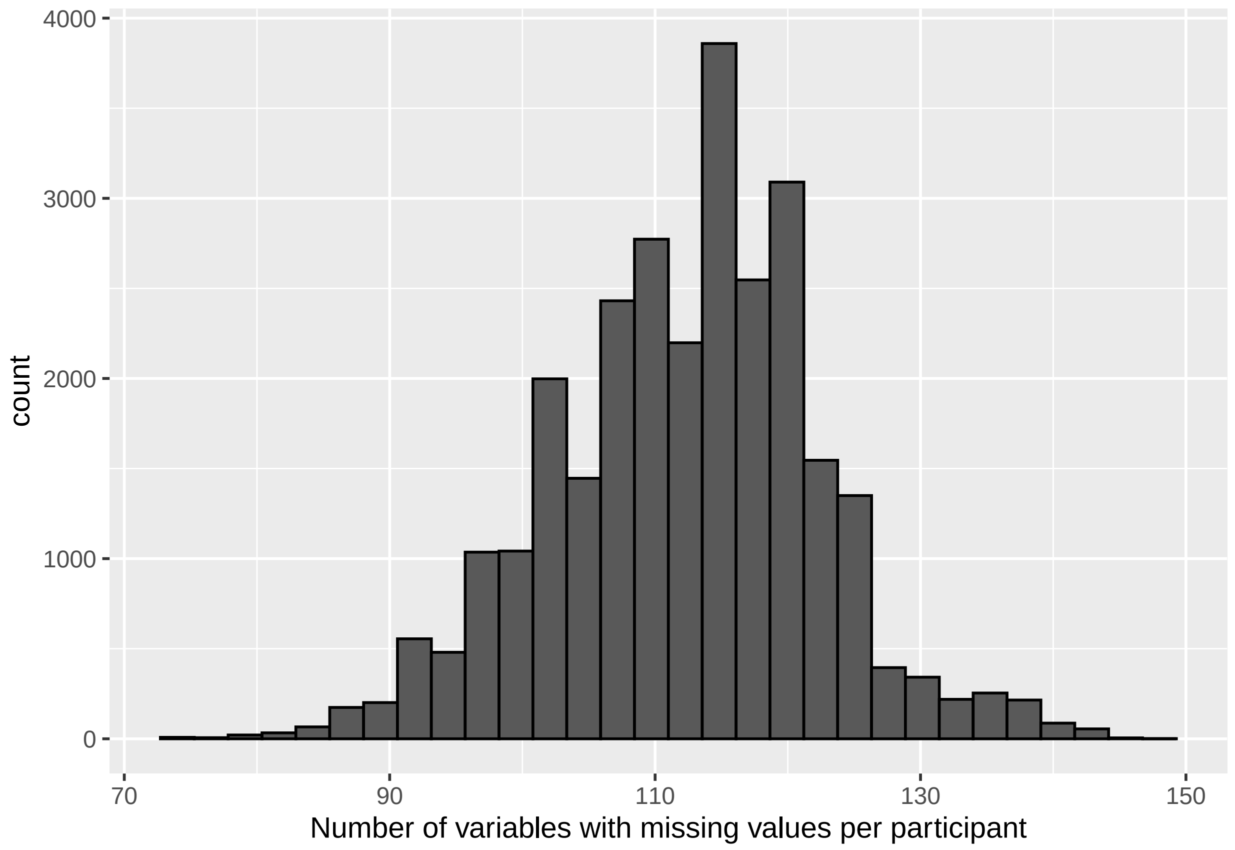
```
## [1] 190
```

```r
###
### Missing data
na_count <- data.frame(na = sapply(dat, function(y) sum(length(which(is.na(y))))))

na_count$variable <- rownames(na_count)
rownames(na_count) <- NULL

na_count %>%
  as_tibble() %>%
  select(variable, na) %>%
  filter(na != 0) %>%
  mutate(na_percent = na/28433) %>%
  arrange(desc(na))
```

```
## # A tibble: 153 x 3
##    variable    na na_percent
##    <chr>    <int>      <dbl>
##  1 HAVEHEPC 28430      1.00
##  2 SIGMTES1 28427      1.00
##  3 TRETHEPC 28426      1.00
##  4 PRIRHEPC 28424      1.00
##  5 VCLNTES1 28423      1.00
##  6 HPVADSHT 28413      0.999
##  7 SDNATES1 28406      0.999
##  8 BLDSTFIT 28396      0.999
##  9 CSRVCTL2 28355      0.997
## 10 PSATIME1 28340      0.997
## # ... with 143 more rows
```

```r
dat %>%
  mutate(na_rowise = rowSums(is.na(.))) %>%
  ggplot(aes(na_rowise)) + geom_histogram(color = "black") +
  labs(x="Number of variables with missing values per participant")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
###


dat <- dat[, colMeans(is.na(dat)) <= threshold]
ncol(dat) # 52 columns left
```

```
## [1] 52
```

```
columns_to_impute <- colnames(dat)[colSums(is.na(dat)) > 0]
columns_to_impute
```

```
##  [1] "CPDEMO1B" "VETERAN3" "EMPLOY1"  "INCOME3"  "DEAF"     "BLIND"
##  [7] "DECIDE"   "DIFFWALK" "DIFFDRES" "DIFFALON" "USENOW3"  "METSTAT"
## [13] "URBSTAT"  "MSCODE"   "DRDXAR3"
```

```
str(dat[,columns_to_impute])
```

```
## 'data.frame':    28433 obs. of  15 variables:
##  $ CPDEMO1B: int  1 1 8 1 1 8 8 1 1 2 ...
##  $ VETERAN3: int  2 2 2 2 1 2 1 2 2 2 ...
##  $ EMPLOY1 : int  8 7 2 7 7 7 7 8 7 7 ...
##  $ INCOME3 : int  77 3 99 77 7 99 5 77 5 10 ...
##  $ DEAF    : int  2 2 2 2 2 2 1 2 2 2 ...
##  $ BLIND   : int  1 2 2 2 2 2 2 2 2 2 ...
##  $ DECIDE  : int  1 2 1 2 1 2 2 2 2 2 ...
##  $ DIFFWALK: int  1 2 2 2 2 1 1 1 2 2 ...
##  $ DIFFDRES: int  2 2 2 2 2 1 2 2 2 2 ...
##  $ DIFFALON: int  1 2 2 2 2 1 1 2 2 2 ...
##  $ USENOW3 : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ METSTAT : int  1 1 1 1 1 2 1 2 1 1 ...
##  $ URBSTAT : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ MSCODE  : int  2 1 3 1 3 2 2 5 2 3 ...
##  $ DRDXAR3 : int  1 2 1 1 2 1 1 2 1 1 ...
```

```
complete_columns <- colnames(dat)[colSums(is.na(dat)) == 0 &
                                      !colnames(dat) %in% c('MICHD', 'CVDINFR4','CVDCRHD4')]

for (c in columns_to_impute) {
    col <- dat[[c]]
    scaled <- scale(dat[, complete_columns])
    knn <- knn(
        train = scaled[!is.na(col), complete_columns],
        test  = scaled[is.na(col), complete_columns],
        cl    = dat[!is.na(col), c]
        )

    dat[is.na(col), c] = knn
}

colSums(is.na(dat))
```

```
##  GENHLTH PHYSHLTH MENTHLTH PRIMINSR PERSDOC3 MEDCOST1 CHECKUP1 CVDINFR4
##        0        0        0        0        0        0        0        0
## CVDCRHD4 CVDSTRK3 CHCSCNCR CHCOCNCR CHCCOPD3 ADDEPEV3 CHCKDNY2 DIABETE4
##        0        0        0        0        0        0        0        0
##  MARITAL RENTHOM1 NUMHHOL3 CPDEMO1B VETERAN3  EMPLOY1  INCOME3     DEAF
##        0        0        0        0        0        0        0        0
##    BLIND   DECIDE DIFFWALK DIFFDRES DIFFALON  USENOW3   QSTVER  QSTLANG
##        0        0        0        0        0        0        0        0
##  METSTAT  URBSTAT   MSCODE    STSTR  RAWRAKE  DUALUSE  TOTINDA  RFHYPE6
##        0        0        0        0        0        0        0        0
##  CHOLCH3    MICHD  ASTHMS1  DRDXAR3     RACE      SEX    AGE80  CHLDCNT
##        0        0        0        0        0        0        0        0
##   EDUCAG  SMOKER3  CURECI1 DROCDY3_
##        0        0        0        0
```

```
set.seed(263)
train_index <- createDataPartition(dat$MICHD, p = 0.8, list = FALSE)
train <- dat[train_index, ]
test <- dat[-train_index, ]
```

XGBoost

In summary: 49 variables, 3 possible outcomes (MICHD, CVDINFR4, CVDCRHD4). N = 22,747 in training data, N = 5,686 in test data.

```
train_variables <- train %>%
  select(-MICHD, -CVDINFR4, -CVDCRHD4)

train_variables_matrix <- train_variables %>%
  data.matrix()
#train_outcomes <- train %>%
#  select(MICHD, CVDINFR4, CVDCRHD4) %>%
#  mutate(across(c(MICHD, CVDINFR4, CVDCRHD4), as.factor))
```

```r
train_outcomes <- train %>%
  pull(MICHD) %>%
  as.character() %>%
  as.numeric()

test_variables <- test %>%
  select(-MICHD, -CVDINFR4, -CVDCRHD4)

test_variables_matrix <- test_variables%>%
  data.matrix()

test_outcomes <- test %>%
  pull(MICHD) %>%
  as.character() %>%
  as.numeric()
```

XGBoost model with all 49 variables

```r
set.seed(43)

grid_tune <- expand.grid(
  nrounds = c(500,1000,1500), # Number of boosting rounds
  max_depth = c(1,3,5), # Max depth of tree
  eta = c(0.01, 0.1, 0.3), # Step size shrinkage to prevent overfitting
  gamma = 0, # Minimum loss reduction required to make a further partition on a leaf node of the tree
  colsample_bytree = 1,
  min_child_weight = 1,
  subsample = 1
)

train_control <- trainControl(method = "cv", # Cross validation
                              number=3, # 3 folds
                              verboseIter = TRUE,
                              allowParallel = TRUE)

xgb_tune <- train(x = train_variables,
                  y = as.factor(train_outcomes),
                  trControl = train_control,
                  tuneGrid = grid_tune,
                  method= "xgbTree",
                  verbose = TRUE)
```

```
## + Fold1: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## [14:15:11] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` inste
## [14:15:11] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` inste
## - Fold1: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## + Fold1: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## [14:15:27] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` inste
## [14:15:27] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` inste
## - Fold1: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## + Fold1: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## [14:15:54] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` inste
## [14:15:54] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold1: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:16:00] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:16:00] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold1: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:16:16] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:16:16] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold1: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:16:42] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:16:42] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold1: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:16:49] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:16:49] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold1: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:17:04] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:17:04] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold1: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:17:31] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:17:31] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:17:38] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:17:38] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:17:54] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:17:54] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:18:22] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:18:22] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:18:28] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:18:28] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:18:44] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:18:44] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:19:10] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:19:10] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:19:17] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:19:17] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:19:33] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:19:33] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:19:59] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:19:59] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:20:06] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:20:06] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:20:22] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:20:22] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:20:49] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:20:49] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:20:55] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:20:55] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:21:12] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:21:12] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:21:39] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:21:39] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:21:46] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:21:46] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:22:02] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:22:02] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:22:29] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:22:29] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## Aggregating results
## Selecting tuning parameters
## Fitting nrounds = 500, max_depth = 5, eta = 0.01, gamma = 0, colsample_bytree = 1, min_child_weight =
```

```
xgb_tune
```

```
## eXtreme Gradient Boosting
##
## 22747 samples
##    49 predictor
##     2 classes: '0', '1'
```

```
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 15164, 15165, 15165
## Resampling results across tuning parameters:
##
##   eta   max_depth  nrounds  Accuracy   Kappa
##   0.01  1           500     0.7051480  0.4099337
##   0.01  1          1000     0.7147317  0.4291049
##   0.01  1          1500     0.7164020  0.4324790
##   0.01  3           500     0.7193914  0.4391762
##   0.01  3          1000     0.7218533  0.4440565
##   0.01  3          1500     0.7214576  0.4432469
##   0.01  5           500     0.7218973  0.4442941
##   0.01  5          1000     0.7206224  0.4416680
##   0.01  5          1500     0.7194355  0.4392701
##   0.10  1           500     0.7187319  0.4371864
##   0.10  1          1000     0.7189957  0.4377392
##   0.10  1          1500     0.7193034  0.4383588
##   0.10  3           500     0.7186881  0.4376126
##   0.10  3          1000     0.7148195  0.4297398
##   0.10  3          1500     0.7095441  0.4191234
##   0.10  5           500     0.7087089  0.4176114
##   0.10  5          1000     0.7003122  0.4007361
##   0.10  5          1500     0.6956524  0.3913083
##   0.30  1           500     0.7208860  0.4415254
##   0.30  1          1000     0.7198310  0.4394442
##   0.30  1          1500     0.7193035  0.4383763
##   0.30  3           500     0.7064227  0.4129112
##   0.30  3          1000     0.6993449  0.3986811
##   0.30  3          1500     0.6906404  0.3812383
##   0.30  5           500     0.6923111  0.3846152
##   0.30  5          1000     0.6840903  0.3680191
##   0.30  5          1500     0.6818482  0.3635456
##
## Tuning parameter 'gamma' was held constant at a value of 0
## Tuning
##
## Tuning parameter 'min_child_weight' was held constant at a value of 1
##
## Tuning parameter 'subsample' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 500, max_depth = 5, eta
##  = 0.01, gamma = 0, colsample_bytree = 1, min_child_weight = 1 and subsample
##  = 1.
```

```r
xgb_best <- xgb_tune$bestTune

train_control <- trainControl(method = "none",
                              verboseIter = TRUE,
                              allowParallel = TRUE)

final_grid <- expand.grid(nrounds = xgb_best$nrounds,
                          eta = xgb_best$eta,
                            max_depth = xgb_best$max_depth,
                            gamma = xgb_best$gamma,
                            colsample_bytree = xgb_best$colsample_bytree,
                            min_child_weight = xgb_best$min_child_weight,
                            subsample = xgb_best$subsample)
xgb_model <- train(x = train_variables,
                  y = as.factor(train_outcomes),
                   trControl = train_control,
                   tuneGrid = final_grid,
                   method = "xgbTree",
                   verbose = TRUE)
```

```
## Fitting nrounds = 500, eta = 0.01, max_depth = 5, gamma = 0, colsample_bytree = 1, min_child_weight =
```

```
xgb_pred <- predict(xgb_model, test_variables)

#' Confusion Matrix
confusionMatrix(as.factor((xgb_pred)),
                as.factor(test_outcomes))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2015  677
##          1  901 2093
##
##                Accuracy : 0.7225
##                  95% CI : (0.7106, 0.7341)
##     No Information Rate : 0.5128
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4457
##
##  Mcnemar's Test P-Value : 1.98e-08
##
##             Sensitivity : 0.6910
##             Specificity : 0.7556
##          Pos Pred Value : 0.7485
##          Neg Pred Value : 0.6991
##              Prevalence : 0.5128
##          Detection Rate : 0.3544
##    Detection Prevalence : 0.4734
##       Balanced Accuracy : 0.7233
##
##        'Positive' Class : 0
##
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
```

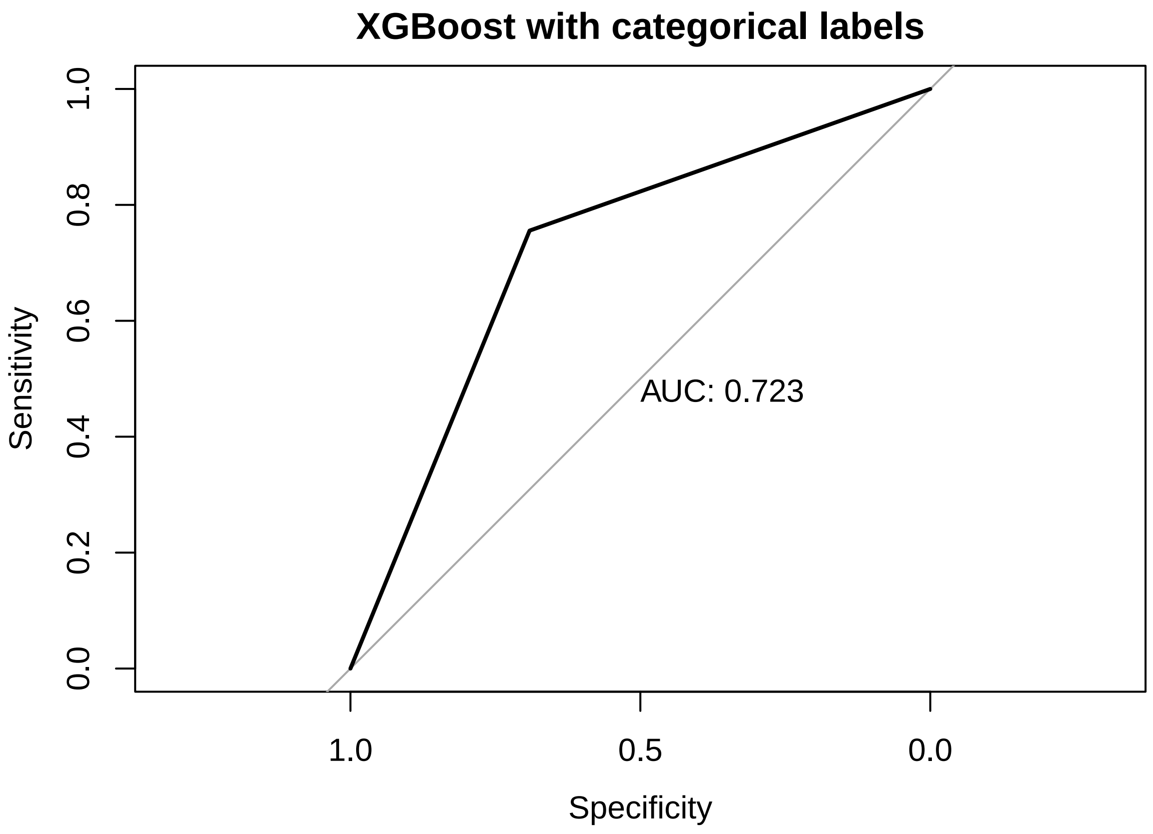
```
##
##     cov, smooth, var
```

```
roc_score <- roc(test_outcomes, as.numeric(xgb_pred))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(roc_score, print.auc = T, main = "XGBoost with categorical labels")
```



XGBoost with One-Hot encoding

49 variables become 220.

Variables to remove: QSTVER, STSTR, RAWRAKE

```
train_variables_2 <- train_variables %>%
  mutate(across(c(-PHYSHLTH, -MENTHLTH, -CPDEMO1B, -STSTR, -RAWRAKE, -AGE80, -DROCDY3_), as.character))

test_variables_2 <- test_variables %>%
  mutate(across(c(-PHYSHLTH, -MENTHLTH, -CPDEMO1B, -STSTR, -RAWRAKE, -AGE80, -DROCDY3_), as.character))

# Make one-hot encoded variables in the training set
dummy <- dummyVars(" ~ .", data=train_variables_2)
train_variables_3 <- data.frame(predict(dummy, newdata = train_variables_2)) %>%
  select(-PRIMINSR6, -CHCCOPD39, -VETERAN33, -DEAF3, -BLIND3, -DIFFWALK3) # Remove 6 one hot encoded va

# Make one-hot encoded variables in the test set
```

```
dummy <- dummyVars(" ~ .", data=test_variables_2)
test_variables_3 <- data.frame(predict(dummy, newdata = test_variables_2))

anti_join(as_tibble(colnames(train_variables_3)), as_tibble(colnames(test_variables_3)))
```

```
## Joining, by = "value"
```

```
## # A tibble: 0 x 1
## # ... with 1 variable: value <chr>
```

```
set.seed(54)

grid_tune <- expand.grid(
  nrounds = c(500,1000,1500), # Number of boosting rounds
  max_depth = c(1,3,5), # Max depth of tree
  eta = c(0.01, 0.1, 0.3), # Step size shrinkage to prevent overfitting
  gamma = 0, # Minimum loss reduction required to make a further partition on a leaf node of the tree
  colsample_bytree = 1,
  min_child_weight = 1,
  subsample = 1
)

train_control <- trainControl(method = "cv", # Cross validation
                              number=3, # 3 folds
                              verboseIter = TRUE,
                              allowParallel = TRUE)



xgb_tune_2 <- train(x = train_variables_3,
                    y = as.factor(train_outcomes),
                    trControl = train_control,
                    tuneGrid = grid_tune,
                    method= "xgbTree",
                    verbose = TRUE)
```

```
## + Fold1: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## [14:23:27] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:23:27] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## + Fold1: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## [14:24:35] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:24:35] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## + Fold1: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## [14:26:24] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:26:25] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## + Fold1: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## [14:26:51] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:26:51] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## + Fold1: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## [14:28:00] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:28:00] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nround
## + Fold1: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:29:49] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:29:49] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold1: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:30:16] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:30:16] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold1: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:31:23] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:31:23] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold1: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:33:11] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:33:11] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold1: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:33:38] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:33:38] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:34:45] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:34:45] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:36:34] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:36:34] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:37:00] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:37:00] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:38:08] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:38:08] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:39:56] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:39:56] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:40:23] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:40:23] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:41:30] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:41:30] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold2: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:43:18] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:43:18] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold2: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:43:45] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:43:45] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.01, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:44:52] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:44:52] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.01, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:46:41] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:46:41] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.01, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:47:08] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:47:08] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.10, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:48:15] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:48:15] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.10, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:50:04] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:50:04] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.10, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:50:31] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:50:31] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.30, max_depth=1, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:51:38] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:51:38] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.30, max_depth=3, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## + Fold3: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## [14:53:28] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [14:53:28] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## - Fold3: eta=0.30, max_depth=5, gamma=0, colsample_bytree=1, min_child_weight=1, subsample=1, nrounds
## Aggregating results
## Selecting tuning parameters
## Fitting nrounds = 1000, max_depth = 5, eta = 0.01, gamma = 0, colsample_bytree = 1, min_child_weight
```

```
xgb_tune_2
```

```
## eXtreme Gradient Boosting
##
## 22747 samples
##   220 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 15165, 15164, 15165
## Resampling results across tuning parameters:
##
##   eta   max_depth  nrounds  Accuracy   Kappa
##   0.01  1           500     0.7080055  0.4154891
##   0.01  1          1000     0.7156108  0.4307983
##   0.01  1          1500     0.7179408  0.4354636
##   0.01  3           500     0.7209302  0.4421745
##   0.01  3          1000     0.7210622  0.4424029
##   0.01  3          1500     0.7204028  0.4410177
##   0.01  5           500     0.7207545  0.4420623
##   0.01  5          1000     0.7213699  0.4432245
##   0.01  5          1500     0.7207104  0.4418455
##   0.10  1           500     0.7185123  0.4366778
##   0.10  1          1000     0.7179408  0.4355307
##   0.10  1          1500     0.7180288  0.4357295
##   0.10  3           500     0.7159625  0.4320790
##   0.10  3          1000     0.7136327  0.4273460
##   0.10  3          1500     0.7110830  0.4221284
##   0.10  5           500     0.7120501  0.4242057
##   0.10  5          1000     0.7072584  0.4145671
##   0.10  5          1500     0.7007520  0.4014728
##   0.30  1           500     0.7170616  0.4337889
##   0.30  1          1000     0.7158747  0.4314583
##   0.30  1          1500     0.7160506  0.4317770
##   0.30  3           500     0.7102476  0.4204758
##   0.30  3          1000     0.6985539  0.3970129
##   0.30  3          1500     0.6926189  0.3850744
##   0.30  5           500     0.6960040  0.3919000
##   0.30  5          1000     0.6885746  0.3771149
##   0.30  5          1500     0.6839585  0.3678674
##
## Tuning parameter 'gamma' was held constant at a value of 0
## Tuning
##
## Tuning parameter 'min_child_weight' was held constant at a value of 1
##
## Tuning parameter 'subsample' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 1000, max_depth = 5, eta
##  = 0.01, gamma = 0, colsample_bytree = 1, min_child_weight = 1 and subsample
##  = 1.
```

```
xgb_best_2 <- xgb_tune_2$bestTune
#Fitting nrounds = 1000, eta = 0.01, max_depth = 5, gamma = 0, colsample_bytree = 1, min_child_weight =

train_control <- trainControl(method = "none",
                              verboseIter = TRUE,
                              allowParallel = TRUE)

final_grid_2 <- expand.grid(nrounds = xgb_best_2$nrounds,
                           eta = xgb_best_2$eta,
                           max_depth = xgb_best_2$max_depth,
                           gamma = xgb_best_2$gamma,
                           colsample_bytree = xgb_best_2$colsample_bytree,
                           min_child_weight = xgb_best_2$min_child_weight,
                           subsample = xgb_best_2$subsample)

xgb_model_2 <- train(x = train_variables_3,
                  y = as.factor(train_outcomes),
                   trControl = train_control,
```

```r
                   tuneGrid = final_grid_2,
                   method = "xgbTree",
                   verbose = TRUE)
```

```
## Fitting nrounds = 1000, eta = 0.01, max_depth = 5, gamma = 0, colsample_bytree = 1, min_child_weight
```

```r
xgb_pred_2 <- predict(xgb_model_2, test_variables_3)

confusionMatrix(as.factor((xgb_pred_2)),
                as.factor(test_outcomes))
```

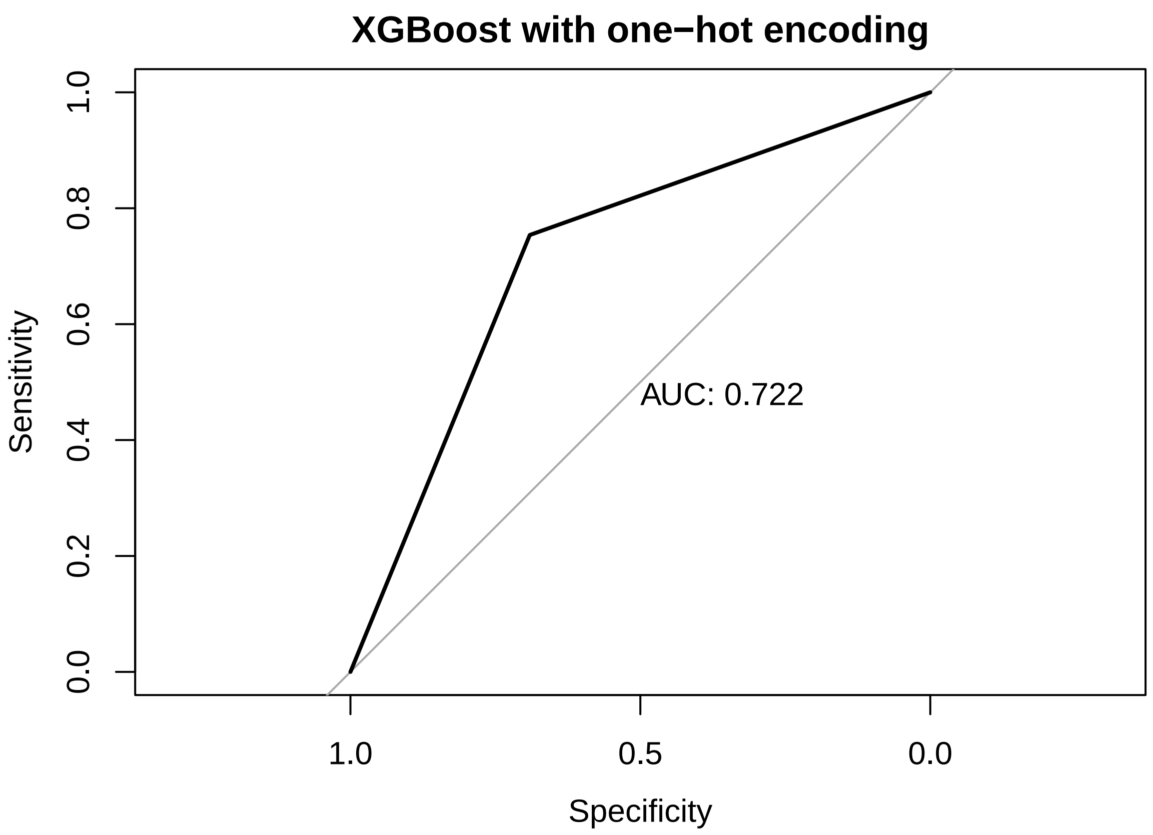
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction    0    1
##            0 2014  682
##            1  902 2088
##
##                Accuracy : 0.7214
##                  95% CI : (0.7096, 0.733)
##     No Information Rate : 0.5128
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4436
##
##  Mcnemar's Test P-Value : 3.743e-08
##
##             Sensitivity : 0.6907
##             Specificity : 0.7538
##          Pos Pred Value : 0.7470
##          Neg Pred Value : 0.6983
##              Prevalence : 0.5128
##          Detection Rate : 0.3542
##    Detection Prevalence : 0.4741
##       Balanced Accuracy : 0.7222
##
##        'Positive' Class : 0
##
```

```r
roc_score_2 <- roc(test_outcomes, as.numeric(xgb_pred_2))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
plot(roc_score_2, print.auc = T, main = "XGBoost with one-hot encoding")
```


XGBoost with one−hot encoding
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
AUC: 0.722
1.0
0.5
0.0
Specificity