Build Survival Model: XGBoost

Mingcheng Hu

Table of contents

Load Data **2**

XGBoost **6**

Data Preparation 6
Hyperparameter Tuning 7
Variable Selection 10
Cross Validation to Select the Best Number of Features 10
Model Fitting 13

```
library(tidyverse)
library(survival)
library(xgboost)
library(caret)
library(survcomp)
library(parallel)
library(mcprogress) # wrap mclapply with progress bar.
library(kableExtra) # include knitr automatically
library(mlr3)  # hyperparameter tuning
library(mlr3tuning)
library(paradox)

source("/work/users/y/u/yuukias/BIOS-Material/BIOS992/utils/csv_utils.r")
# * Don't use setwd() for Quarto documents!
# setwd("/work/users/y/u/yuukias/BIOS-Material/BIOS992/data")

adjust_type <- ifelse(exists("params"), params$adjust_type, "full") #
 ↪  options: "minimal", "partial", "full"
impute_type <- ifelse(exists("params"), params$impute_type, "imputed") #
 ↪  options: "unimputed", "imputed"
```

```r
include_statin <- ifelse(exists("params"), params$include_statin, "no") #
 ↪  options: "yes", "no"

# hyperparameter tuning trials
n_trials <- 50   # * It is recommended to set it to n_params * (8~10)
n_folds <- 10
set.seed(1234)
```

```r
# string of parameters
adjust_type_str <- switch(adjust_type,
    minimal = "minimal",
    partial = "partial",
    full = "full"
)
print(paste0("Model Adjustment Type: ", adjust_type_str))
```

```
[1] "Model Adjustment Type: full"
```

```r
impute_type_str <- switch(impute_type,
    unimputed = "unimputed",
    imputed = "imputed"
)
print(paste0("Data Imputation Type: ", impute_type_str))
```

```
[1] "Data Imputation Type: imputed"
```

Load Data

```r
if (include_statin == "yes") {
    data_train <-
 ↪  read.csv(paste0("/work/users/y/u/yuukias/BIOS-Material/BIOS992/data/train_data_",
 ↪  impute_type_str, "_statin.csv"),
        header = TRUE
    )
} else {
    data_train <-
 ↪  read.csv(paste0("/work/users/y/u/yuukias/BIOS-Material/BIOS992/data/train_data_",
 ↪  impute_type_str, ".csv"),
```

```
        header = TRUE
    )
}

data_train <- data_train[, -1] # the first column is the index generated by
 ↪  sklearn
(dim(data_train))
```

```
[1] 28127   100
```

```
data <- select_subset(data_train, type = adjust_type)
(dim(data))
```

```
[1] 28127    89
```

```
colnames(data)
```

```
 [1] "event"                      "time"
 [3] "age"                        "sex"
 [5] "ethnicity"                  "BMI"
 [7] "smoking"                    "diabetes"
 [9] "systolic_bp"                "hypertension_treatment"
[11] "total_chol"                 "hdl_chol"
[13] "education"                  "activity"
[15] "max_workload"               "max_heart_rate"
[17] "HRV_MeanNN"                 "HRV_SDNN"
[19] "HRV_RMSSD"                  "HRV_SDSD"
[21] "HRV_CVNN"                   "HRV_CVSD"
[23] "HRV_MedianNN"               "HRV_MadNN"
[25] "HRV_MCVNN"                  "HRV_IQRNN"
[27] "HRV_SDRMSSD"                "HRV_Prc20NN"
[29] "HRV_Prc80NN"                "HRV_pNN50"
[31] "HRV_pNN20"                  "HRV_MinNN"
[33] "HRV_MaxNN"                  "HRV_HTI"
[35] "HRV_TINN"                   "HRV_LF"
[37] "HRV_HF"                     "HRV_VHF"
[39] "HRV_TP"                     "HRV_LFHF"
[41] "HRV_LFn"                    "HRV_HFn"
[43] "HRV_LnHF"                   "HRV_SD1"
```

```
[45] "HRV_SD2"                      "HRV_SD1SD2"
[47] "HRV_S"                        "HRV_CSI"
[49] "HRV_CVI"                      "HRV_CSI_Modified"
[51] "HRV_PIP"                      "HRV_IALS"
[53] "HRV_PSS"                      "HRV_PAS"
[55] "HRV_GI"                       "HRV_SI"
[57] "HRV_AI"                       "HRV_PI"
[59] "HRV_C1d"                      "HRV_C1a"
[61] "HRV_SD1d"                     "HRV_SD1a"
[63] "HRV_C2d"                      "HRV_C2a"
[65] "HRV_SD2d"                     "HRV_SD2a"
[67] "HRV_Cd"                       "HRV_Ca"
[69] "HRV_SDNNd"                    "HRV_SDNNa"
[71] "HRV_ApEn"                     "HRV_ShanEn"
[73] "HRV_FuzzyEn"                  "HRV_MSEn"
[75] "HRV_CMSEn"                    "HRV_RCMSEn"
[77] "HRV_CD"                       "HRV_HFD"
[79] "HRV_KFD"                      "HRV_LZC"
[81] "HRV_DFA_alpha1"               "HRV_MFDFA_alpha1_Width"
[83] "HRV_MFDFA_alpha1_Peak"        "HRV_MFDFA_alpha1_Mean"
[85] "HRV_MFDFA_alpha1_Max"         "HRV_MFDFA_alpha1_Delta"
[87] "HRV_MFDFA_alpha1_Asymmetry"   "HRV_MFDFA_alpha1_Fluctuation"
[89] "HRV_MFDFA_alpha1_Increment"
```

```r
data <- tibble::as_tibble(data)
```

```r
# * There are some imputed ethnicity set to "e". We will exclude them at this
 ↪  time.
data <- data %>%
    filter(ethnicity != "e")

# * We also need to manually relevel the categorical variables
data <- data %>%
    mutate(
        # Set "Never" (0) as baseline for smoking
        smoking = factor(smoking,
            levels = c("0", "1", "2", "-3"),
            labels = c("Never", "Previous", "Current", "Prefer not to
             ↪  answer")
        ),

        # Set "No" (0) as baseline for diabetes
```

```
        diabetes = factor(diabetes,
            levels = c("0", "1", "-1", "-3"),
            labels = c("No", "Yes", "Do not know", "Prefer not to answer")
        ),

        # Ensure other categorical variables are properly factored
        ethnicity = factor(ethnicity,
            levels = c("1", "2", "3", "4", "5", "6"),
            labels = c("White", "Mixed", "Asian/Asian British", "Black/Black
             ↪  British", "Chinese", "Other")
        ),
        education = factor(education,
            levels = c("1", "2", "3", "4", "5", "6", "-7", "-3"),
            labels = c(
                "College/University degree", "A levels/AS levels",
                "O levels/GCSEs", "CSEs", "NVQ/HND/HNC",
                "Other professional", "None of the above",
                "Prefer not to answer"
            )
        ),
        activity = factor(activity,
            levels = c("0", "1", "2"),
            labels = c("Low", "Moderate", "High")
        ),
        sex = factor(sex,
            levels = c("0", "1"),
            labels = c("Female", "Male")
        ),
        hypertension_treatment = factor(hypertension_treatment,
            levels = c("0", "1"),
            labels = c("No", "Yes")
        )
    )
```

```
# * It is very hard to compare the HR as different predictors are on
 ↪  different magnitudes, so we need to normalize them.
time_col <- data$time
event_col <- data$event
data <- data %>%
    select(-c(time, event)) %>%
    mutate(across(where(is.numeric), scale)) %>%
```

```
    mutate(
        time = time_col,
        event = event_col
    )
```

Note now the interpretation of HR is different! For example, if HR=1.16 for the predictor in the univariate model fitted using scaled data, it means that each standard deviation increase is associated with 16% higher risk of event.

```
# For XGBoost model, we create a validation set for early stopping.
set.seed(1234)
train_index <- createDataPartition(
    data$event, # stratify by event
    p = 0.8,
    list = FALSE
)

train_data <- data[train_index, ]
val_data <- data[-train_index, ]
```

XGBoost

Data Preparation

XGBoost does not support categorical variables. We need to convert them to dummy variables using `model.matrix`.

```
total_x <- data %>% select(-c(time, event))
total_x_xgb <- model.frame(~ . - 1, data = total_x, na.action = na.pass)
total_x_xgb <- model.matrix(~ . - 1, data = total_x_xgb)
total_y <- data %>% select(time, event)
# * Note format of label should be different when using Cox model and AFT
 ↪  model.
# define For uncensored labels, use [a,a]
# define For right-censored labels, use [a,Inf]
total_y_lower_bound <- data$time
total_y_upper_bound <- ifelse(data$event == 1, data$time, Inf)

train_x <- train_data %>% select(-c(time, event))
```

```r
# * To avoid deleting rows, we need to set na.action to na.pass for
 ↪  model.matrix
train_x_xgb <- model.frame(~ . - 1, data = train_x, na.action = na.pass)
train_x_xgb <- model.matrix(~ . - 1, data = train_x_xgb)
train_y_lower_bound <- train_data$time
train_y_upper_bound <- ifelse(train_data$event == 1, train_data$time, Inf)
dtrain <- xgb.DMatrix(
    data = train_x_xgb,
    label_lower_bound = train_y_lower_bound,
    label_upper_bound = train_y_upper_bound
)

val_x <- val_data %>% select(-c(time, event))
val_x_xgb <- model.frame(~ . - 1, data = val_x, na.action = na.pass)
val_x_xgb <- model.matrix(~ . - 1, data = val_x_xgb)
val_y_lower_bound <- val_data$time
val_y_upper_bound <- ifelse(val_data$event == 1, val_data$time, Inf)
dval <- xgb.DMatrix(
    data = val_x_xgb,
    label_lower_bound = val_y_lower_bound,
    label_upper_bound = val_y_upper_bound
)
```

Hyperparameter Tuning

```r
# Ref Barnwal, A., Cho ,Hyunsu, & and Hocking, T. (2022). Survival Regression
 ↪  with Accelerated Failure Time Model in XGBoost. Journal of Computational
 ↪  and Graphical Statistics, 31(4), 1292-1302.
 ↪  https://doi.org/10.1080/10618600.2022.2067548
param_set <- ParamSet$new(params = list(
    learning_rate = p_dbl(
        lower = log10(0.001),
        upper = log10(1.0),
        trafo = function(x) 10^x
    ),
    max_depth = p_int(
        lower = 2,
        upper = 10
    ),
    min_child_weight = p_dbl(
```

```
        lower = log10(0.001),
        upper = log10(100.0),
        trafo = function(x) 10^x
    ),
    reg_alpha = p_dbl(
        lower = log10(0.001),
        upper = log10(100.0),
        trafo = function(x) 10^x
    ),
    reg_lambda = p_dbl(
        lower = log10(0.001),
        upper = log10(100.0),
        trafo = function(x) 10^x
    ),
    aft_loss_distribution_scale = p_dbl(
        lower = 0.5,
        upper = 2.0
    )
))
```

```
tune_xgb <- function(params_trial) {
    model <- xgb.train(
        params = c(
            list(
                objective = "survival:aft",
                eval_metric = "aft-nloglik",
                aft_loss_distribution = "normal"
            ),
            params_trial
        ),
        data = dtrain,
        nrounds = 500,
        early_stopping_rounds = 10,
        watchlist = list(train = dtrain, val = dval),
        verbose = 0
    )
    pred <- predict(model, dval)
    pred <- -pred
    # return(list(score = min(model$evaluation_log$val_aft_nloglik)))
    return(list(score = concordance.index(pred, val_data$time,
     ↪  val_data$event)$c.index))
```

```
}
```

```
tuning_results <- pmclapply(1:n_trials, function(i) {
    params_trial <- generate_design_random(param_set, n = 1)$data
    params_trial <- param_set$trafo(params_trial)
    score <- tune_xgb(params_trial)
    return(data.frame(trial = i, score = score$score, params = params_trial))
}, title = "Tuning XGBoost hyperparameters")
```

```
tuning_results <- bind_rows(tuning_results)  # convert list of lists to a
 ↪  data frame
tuning_results_best <- tuning_results[which.max(tuning_results$score), ]


model_params <- list(
    learning_rate = tuning_results_best$params.learning_rate,
    max_depth = tuning_results_best$params.max_depth,
    min_child_weight = tuning_results_best$params.min_child_weight,
    reg_alpha = tuning_results_best$params.reg_alpha,
    reg_lambda = tuning_results_best$params.reg_lambda,
    aft_loss_distribution_scale =
     ↪  tuning_results_best$params.aft_loss_distribution_scale
)
print("Best hyperparameters:")
```

```
[1] "Best hyperparameters:"
```

```
print(model_params)
```

```
$learning_rate
[1] 0.06457443

$max_depth
[1] 3

$min_child_weight
[1] 0.009989202

$reg_alpha
```

```
[1] 0.007727406

$reg_lambda
[1] 0.04588311

$aft_loss_distribution_scale
[1] 1.863505
```

Variable Selection

```
# * As mentioned in the paper, we use AFT model instead of Cox model.
xgb_var_select <- xgb.train(
    params = c(
        list(
            objective = "survival:aft",
            eval_metric = "aft-nloglik",
            aft_loss_distribution = "normal"
        ),
        model_params
    ),
    data = dtrain,
    nrounds = 100,
    early_stopping_rounds = 10,
    watchlist = list(train = dtrain, val = dval)
)
```

```
# Sort descendingly using gain
xgb_importance <- xgb.importance(model = xgb_var_select)
# Other attributes: Gain, Cover, Frequency
vars_ranked <- xgb_importance$Feature
```

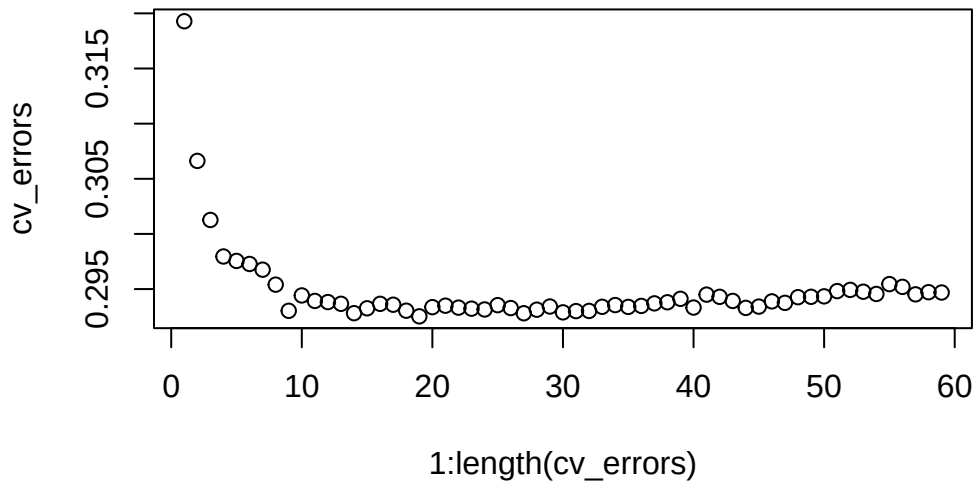
Cross Validation to Select the Best Number of Features

```
# * xgb.cv is not available for AFT model.
set.seed(1234)
folds <- createFolds(data$event, k = n_folds)

cv_errors <- pmclapply(seq(1, length(vars_ranked), by = 1),
 ↪  function(num_vars) {
```

```
    selected_vars <- vars_ranked[1:num_vars]
    fold_errors <- sapply(folds, function(fold_idx) {
        # * We take all training data and validation data and then split them
        ↪  into folds.
        # train_x_fold <- total_x[-fold_idx, selected_vars, drop = FALSE]
        # train_x_fold <- model.frame(~ . - 1, data = train_x_fold, na.action
        ↪  = na.pass)
        # train_x_fold <- model.matrix(~ . - 1, data = train_x_fold)
        train_x_fold <- total_x_xgb[-fold_idx, selected_vars, drop = FALSE]
        train_y_lower_fold <- total_y_lower_bound[-fold_idx]
        train_y_upper_fold <- total_y_upper_bound[-fold_idx]

        # val_x_fold <- total_x[fold_idx, selected_vars, drop = FALSE]
        # val_x_fold <- model.frame(~ . - 1, data = val_x_fold, na.action =
        ↪  na.pass)
        # val_x_fold <- model.matrix(~ . - 1, data = val_x_fold)
        val_x_fold <- total_x_xgb[fold_idx, selected_vars, drop = FALSE]
        val_y_lower_fold <- total_y_lower_bound[fold_idx]
        val_y_upper_fold <- total_y_upper_bound[fold_idx]
        val_y_fold <- total_y[fold_idx, ]  # for C-index calculation

        dtrain_fold <- xgb.DMatrix(
            data = train_x_fold,
            label_lower_bound = train_y_lower_fold,
            # label_upper_bound = train_y_upper_fold
            label_upper_bound = train_y_lower_fold
        )

        dval_fold <- xgb.DMatrix(
            data = val_x_fold,
            label_lower_bound = val_y_lower_fold,
            # label_upper_bound = val_y_upper_fold
            label_upper_bound = val_y_lower_fold
        )

        model <- xgb.train(
            params = c(
                list(
                    objective = "survival:aft",
                    eval_metric = "aft-nloglik",
                    aft_loss_distribution = "normal"
                ),
```

```
                model_params
            ),
            data = dtrain_fold,
            nrounds = 500,
            early_stopping_rounds = 10,
            watchlist = list(train = dtrain_fold, val = dval_fold),
            verbose = 0
        )
        # * It outputs the estimated survival time. We need to convert it to
        ↪  risk.
        pred <- predict(model, dval_fold)
        pred <- -pred

        # Use C-index to measure the performance of the model
        1 - concordance.index(pred, val_y_fold$time,
        ↪  val_y_fold$event)$c.index
    })
    print(mean(fold_errors))
    mean(fold_errors)
}, title = "Cross Validation to Select the Best Number of Features")
```

```
cv_errors <- as.numeric(cv_errors)
plot(1:length(cv_errors), cv_errors)
```




```r
best_num_vars <- which.min(cv_errors)
vars_selected <- vars_ranked[1:best_num_vars]
```

```r
print(paste0("The best number of features to retain is ", best_num_vars))
```

```
[1] "The best number of features to retain is 19"
```

Model Fitting

```r
train_x_selected <- train_x_xgb[, vars_selected]
# No need to change the label
train_y_lower_bound_selected <- train_y_lower_bound
train_y_upper_bound_selected <- train_y_upper_bound

dtrain_selected <- xgb.DMatrix(
    data = train_x_selected,
    label_lower_bound = train_y_lower_bound_selected,
    label_upper_bound = train_y_upper_bound_selected
)
```

```
val_x_selected <- val_x_xgb[, vars_selected]
val_y_lower_bound_selected <- val_y_lower_bound
val_y_upper_bound_selected <- val_y_upper_bound
dval_selected <- xgb.DMatrix(
    data = val_x_selected,
    label_lower_bound = val_y_lower_bound_selected,
    label_upper_bound = val_y_upper_bound_selected
)


xgb_model <- xgb.train(
    params = c(
        list(
            objective = "survival:aft",
            eval_metric = "aft-nloglik",
            aft_loss_distribution = "normal"
        ),
        model_params
    ),
    data = dtrain_selected,
    nrounds = 100,
    early_stopping_rounds = 10,
    watchlist = list(train = dtrain_selected, val = dval_selected)
)
```

```
[1] train-aft-nloglik:13.529111 val-aft-nloglik:13.534252
Multiple eval metrics are present. Will use val_aft_nloglik for early stopping.
Will train until val_aft_nloglik hasn't improved in 10 rounds.

[2] train-aft-nloglik:12.107670 val-aft-nloglik:12.113716
[3] train-aft-nloglik:10.858367 val-aft-nloglik:10.865424
[4] train-aft-nloglik:9.759843  val-aft-nloglik:9.767852
[5] train-aft-nloglik:8.793434  val-aft-nloglik:8.802304
[6] train-aft-nloglik:7.942785  val-aft-nloglik:7.952680
[7] train-aft-nloglik:7.193633  val-aft-nloglik:7.204162
[8] train-aft-nloglik:6.533491  val-aft-nloglik:6.544646
[9] train-aft-nloglik:5.951415  val-aft-nloglik:5.963313
[10]    train-aft-nloglik:5.437851  val-aft-nloglik:5.450427
[11]    train-aft-nloglik:4.984449  val-aft-nloglik:4.997488
[12]    train-aft-nloglik:4.583877  val-aft-nloglik:4.597448
[13]    train-aft-nloglik:4.229737  val-aft-nloglik:4.243848
```

```
[14]    train-aft-nloglik:3.916447  val-aft-nloglik:3.930971
[15]    train-aft-nloglik:3.639075  val-aft-nloglik:3.654072
[16]    train-aft-nloglik:3.393352  val-aft-nloglik:3.408862
[17]    train-aft-nloglik:3.175513  val-aft-nloglik:3.191317
[18]    train-aft-nloglik:2.982253  val-aft-nloglik:2.998425
[19]    train-aft-nloglik:2.810681  val-aft-nloglik:2.827220
[20]    train-aft-nloglik:2.658246  val-aft-nloglik:2.675197
[21]    train-aft-nloglik:2.522750  val-aft-nloglik:2.540063
[22]    train-aft-nloglik:2.402222  val-aft-nloglik:2.419862
[23]    train-aft-nloglik:2.294960  val-aft-nloglik:2.312895
[24]    train-aft-nloglik:2.199424  val-aft-nloglik:2.217830
[25]    train-aft-nloglik:2.114297  val-aft-nloglik:2.132959
[26]    train-aft-nloglik:2.038413  val-aft-nloglik:2.057415
[27]    train-aft-nloglik:1.970742  val-aft-nloglik:1.989969
[28]    train-aft-nloglik:1.910345  val-aft-nloglik:1.929906
[29]    train-aft-nloglik:1.856433  val-aft-nloglik:1.876222
[30]    train-aft-nloglik:1.808301  val-aft-nloglik:1.828304
[31]    train-aft-nloglik:1.765297  val-aft-nloglik:1.785566
[32]    train-aft-nloglik:1.726866  val-aft-nloglik:1.747403
[33]    train-aft-nloglik:1.692515  val-aft-nloglik:1.713429
[34]    train-aft-nloglik:1.661801  val-aft-nloglik:1.682950
[35]    train-aft-nloglik:1.634356  val-aft-nloglik:1.655673
[36]    train-aft-nloglik:1.609793  val-aft-nloglik:1.631367
[37]    train-aft-nloglik:1.587830  val-aft-nloglik:1.609652
[38]    train-aft-nloglik:1.568184  val-aft-nloglik:1.590272
[39]    train-aft-nloglik:1.550600  val-aft-nloglik:1.572896
[40]    train-aft-nloglik:1.534872  val-aft-nloglik:1.557367
[41]    train-aft-nloglik:1.520815  val-aft-nloglik:1.543520
[42]    train-aft-nloglik:1.508224  val-aft-nloglik:1.531105
[43]    train-aft-nloglik:1.496966  val-aft-nloglik:1.520052
[44]    train-aft-nloglik:1.486845  val-aft-nloglik:1.510121
[45]    train-aft-nloglik:1.477822  val-aft-nloglik:1.501319
[46]    train-aft-nloglik:1.469722  val-aft-nloglik:1.493410
[47]    train-aft-nloglik:1.462486  val-aft-nloglik:1.486340
[48]    train-aft-nloglik:1.455998  val-aft-nloglik:1.480020
[49]    train-aft-nloglik:1.450209  val-aft-nloglik:1.474386
[50]    train-aft-nloglik:1.445020  val-aft-nloglik:1.469438
[51]    train-aft-nloglik:1.440365  val-aft-nloglik:1.464905
[52]    train-aft-nloglik:1.436198  val-aft-nloglik:1.460946
[53]    train-aft-nloglik:1.432448  val-aft-nloglik:1.457365
[54]    train-aft-nloglik:1.429092  val-aft-nloglik:1.454222
[55]    train-aft-nloglik:1.426034  val-aft-nloglik:1.451447
[56]    train-aft-nloglik:1.423338  val-aft-nloglik:1.448909
```

```
[57]    train-aft-nloglik:1.420862  val-aft-nloglik:1.446735
[58]    train-aft-nloglik:1.418673  val-aft-nloglik:1.444735
[59]    train-aft-nloglik:1.416711  val-aft-nloglik:1.442993
[60]    train-aft-nloglik:1.414935  val-aft-nloglik:1.441556
[61]    train-aft-nloglik:1.413335  val-aft-nloglik:1.440128
[62]    train-aft-nloglik:1.411895  val-aft-nloglik:1.438884
[63]    train-aft-nloglik:1.410604  val-aft-nloglik:1.437749
[64]    train-aft-nloglik:1.409432  val-aft-nloglik:1.436671
[65]    train-aft-nloglik:1.408387  val-aft-nloglik:1.435775
[66]    train-aft-nloglik:1.407395  val-aft-nloglik:1.435006
[67]    train-aft-nloglik:1.406475  val-aft-nloglik:1.434373
[68]    train-aft-nloglik:1.405653  val-aft-nloglik:1.433760
[69]    train-aft-nloglik:1.404924  val-aft-nloglik:1.433205
[70]    train-aft-nloglik:1.404262  val-aft-nloglik:1.432690
[71]    train-aft-nloglik:1.403653  val-aft-nloglik:1.432214
[72]    train-aft-nloglik:1.403083  val-aft-nloglik:1.431795
[73]    train-aft-nloglik:1.402530  val-aft-nloglik:1.431588
[74]    train-aft-nloglik:1.402052  val-aft-nloglik:1.431201
[75]    train-aft-nloglik:1.401614  val-aft-nloglik:1.430944
[76]    train-aft-nloglik:1.401205  val-aft-nloglik:1.430721
[77]    train-aft-nloglik:1.400805  val-aft-nloglik:1.430476
[78]    train-aft-nloglik:1.400442  val-aft-nloglik:1.430281
[79]    train-aft-nloglik:1.400093  val-aft-nloglik:1.430039
[80]    train-aft-nloglik:1.399767  val-aft-nloglik:1.429874
[81]    train-aft-nloglik:1.399437  val-aft-nloglik:1.429736
[82]    train-aft-nloglik:1.399181  val-aft-nloglik:1.429575
[83]    train-aft-nloglik:1.398910  val-aft-nloglik:1.429391
[84]    train-aft-nloglik:1.398610  val-aft-nloglik:1.429324
[85]    train-aft-nloglik:1.398369  val-aft-nloglik:1.429263
[86]    train-aft-nloglik:1.398132  val-aft-nloglik:1.429168
[87]    train-aft-nloglik:1.397924  val-aft-nloglik:1.429008
[88]    train-aft-nloglik:1.397706  val-aft-nloglik:1.428882
[89]    train-aft-nloglik:1.397521  val-aft-nloglik:1.428758
[90]    train-aft-nloglik:1.397288  val-aft-nloglik:1.428715
[91]    train-aft-nloglik:1.397096  val-aft-nloglik:1.428594
[92]    train-aft-nloglik:1.396939  val-aft-nloglik:1.428504
[93]    train-aft-nloglik:1.396774  val-aft-nloglik:1.428405
[94]    train-aft-nloglik:1.396584  val-aft-nloglik:1.428386
[95]    train-aft-nloglik:1.396439  val-aft-nloglik:1.428369
[96]    train-aft-nloglik:1.396306  val-aft-nloglik:1.428315
[97]    train-aft-nloglik:1.396144  val-aft-nloglik:1.428288
[98]    train-aft-nloglik:1.396003  val-aft-nloglik:1.428294
[99]    train-aft-nloglik:1.395838  val-aft-nloglik:1.428237
```

```
[100]   train-aft-nloglik:1.395698  val-aft-nloglik:1.428162
```

```r
train_x_full <- model.frame(~ . - 1, data = train_x, na.action = na.pass)
train_x_full <- model.matrix(~ . - 1, data = train_x_full)
train_y_lower_bound_full <- train_y_lower_bound
train_y_upper_bound_full <- train_y_upper_bound
dtrain_full <- xgb.DMatrix(
    data = train_x_full,
    label_lower_bound = train_y_lower_bound_full,
    label_upper_bound = train_y_upper_bound_full
)

# We also fit the full model
xgb_model_full <- xgb.train(
    params = c(
        list(
            objective = "survival:aft",
            eval_metric = "aft-nloglik",
            aft_loss_distribution = "normal"
        ),
        model_params
    ),
    data = dtrain_full,
    nrounds = 100,
    early_stopping_rounds = 10,
    watchlist = list(train = dtrain_full, val = dval)
)
```

```
[1] train-aft-nloglik:13.529111 val-aft-nloglik:13.534252
Multiple eval metrics are present. Will use val_aft_nloglik for early stopping.
Will train until val_aft_nloglik hasn't improved in 10 rounds.

[2] train-aft-nloglik:12.107670 val-aft-nloglik:12.113716
[3] train-aft-nloglik:10.858367 val-aft-nloglik:10.865424
[4] train-aft-nloglik:9.759843  val-aft-nloglik:9.767852
[5] train-aft-nloglik:8.793434  val-aft-nloglik:8.802304
[6] train-aft-nloglik:7.942785  val-aft-nloglik:7.952680
[7] train-aft-nloglik:7.193633  val-aft-nloglik:7.204162
[8] train-aft-nloglik:6.533491  val-aft-nloglik:6.544646
[9] train-aft-nloglik:5.951415  val-aft-nloglik:5.963313
[10]    train-aft-nloglik:5.437851  val-aft-nloglik:5.450427
[11]    train-aft-nloglik:4.984449  val-aft-nloglik:4.997488
```

```
[12]    train-aft-nloglik:4.583877  val-aft-nloglik:4.597448
[13]    train-aft-nloglik:4.229737  val-aft-nloglik:4.243848
[14]    train-aft-nloglik:3.916447  val-aft-nloglik:3.930971
[15]    train-aft-nloglik:3.639075  val-aft-nloglik:3.654072
[16]    train-aft-nloglik:3.393352  val-aft-nloglik:3.408862
[17]    train-aft-nloglik:3.175513  val-aft-nloglik:3.191317
[18]    train-aft-nloglik:2.982253  val-aft-nloglik:2.998425
[19]    train-aft-nloglik:2.810681  val-aft-nloglik:2.827220
[20]    train-aft-nloglik:2.658240  val-aft-nloglik:2.675195
[21]    train-aft-nloglik:2.522743  val-aft-nloglik:2.540061
[22]    train-aft-nloglik:2.402215  val-aft-nloglik:2.419860
[23]    train-aft-nloglik:2.294953  val-aft-nloglik:2.312894
[24]    train-aft-nloglik:2.199417  val-aft-nloglik:2.217829
[25]    train-aft-nloglik:2.114290  val-aft-nloglik:2.132959
[26]    train-aft-nloglik:2.038405  val-aft-nloglik:2.057415
[27]    train-aft-nloglik:1.970734  val-aft-nloglik:1.989970
[28]    train-aft-nloglik:1.910336  val-aft-nloglik:1.929906
[29]    train-aft-nloglik:1.856423  val-aft-nloglik:1.876222
[30]    train-aft-nloglik:1.808292  val-aft-nloglik:1.828305
[31]    train-aft-nloglik:1.765287  val-aft-nloglik:1.785597
[32]    train-aft-nloglik:1.726853  val-aft-nloglik:1.747517
[33]    train-aft-nloglik:1.692506  val-aft-nloglik:1.713406
[34]    train-aft-nloglik:1.661798  val-aft-nloglik:1.682957
[35]    train-aft-nloglik:1.634340  val-aft-nloglik:1.655716
[36]    train-aft-nloglik:1.609790  val-aft-nloglik:1.631441
[37]    train-aft-nloglik:1.587824  val-aft-nloglik:1.609679
[38]    train-aft-nloglik:1.568166  val-aft-nloglik:1.590195
[39]    train-aft-nloglik:1.550600  val-aft-nloglik:1.572852
[40]    train-aft-nloglik:1.534879  val-aft-nloglik:1.557270
[41]    train-aft-nloglik:1.520802  val-aft-nloglik:1.543442
[42]    train-aft-nloglik:1.508208  val-aft-nloglik:1.531052
[43]    train-aft-nloglik:1.496943  val-aft-nloglik:1.519995
[44]    train-aft-nloglik:1.486867  val-aft-nloglik:1.510131
[45]    train-aft-nloglik:1.477826  val-aft-nloglik:1.501307
[46]    train-aft-nloglik:1.469733  val-aft-nloglik:1.493423
[47]    train-aft-nloglik:1.462471  val-aft-nloglik:1.486350
[48]    train-aft-nloglik:1.455985  val-aft-nloglik:1.480042
[49]    train-aft-nloglik:1.450188  val-aft-nloglik:1.474451
[50]    train-aft-nloglik:1.445003  val-aft-nloglik:1.469444
[51]    train-aft-nloglik:1.440342  val-aft-nloglik:1.465039
[52]    train-aft-nloglik:1.436167  val-aft-nloglik:1.460997
[53]    train-aft-nloglik:1.432426  val-aft-nloglik:1.457508
[54]    train-aft-nloglik:1.429053  val-aft-nloglik:1.454297
```

```
[55]    train-aft-nloglik:1.426032  val-aft-nloglik:1.451430
[56]    train-aft-nloglik:1.423354  val-aft-nloglik:1.448907
[57]    train-aft-nloglik:1.420911  val-aft-nloglik:1.446637
[58]    train-aft-nloglik:1.418732  val-aft-nloglik:1.444701
[59]    train-aft-nloglik:1.416740  val-aft-nloglik:1.442897
[60]    train-aft-nloglik:1.414977  val-aft-nloglik:1.441254
[61]    train-aft-nloglik:1.413297  val-aft-nloglik:1.439971
[62]    train-aft-nloglik:1.411838  val-aft-nloglik:1.438708
[63]    train-aft-nloglik:1.410531  val-aft-nloglik:1.437583
[64]    train-aft-nloglik:1.409316  val-aft-nloglik:1.436690
[65]    train-aft-nloglik:1.408207  val-aft-nloglik:1.435807
[66]    train-aft-nloglik:1.407172  val-aft-nloglik:1.435127
[67]    train-aft-nloglik:1.406289  val-aft-nloglik:1.434351
[68]    train-aft-nloglik:1.405466  val-aft-nloglik:1.433739
[69]    train-aft-nloglik:1.404686  val-aft-nloglik:1.433310
[70]    train-aft-nloglik:1.403996  val-aft-nloglik:1.432835
[71]    train-aft-nloglik:1.403369  val-aft-nloglik:1.432391
[72]    train-aft-nloglik:1.402784  val-aft-nloglik:1.431876
[73]    train-aft-nloglik:1.402225  val-aft-nloglik:1.431593
[74]    train-aft-nloglik:1.401738  val-aft-nloglik:1.431300
[75]    train-aft-nloglik:1.401286  val-aft-nloglik:1.431025
[76]    train-aft-nloglik:1.400865  val-aft-nloglik:1.430740
[77]    train-aft-nloglik:1.400452  val-aft-nloglik:1.430642
[78]    train-aft-nloglik:1.400063  val-aft-nloglik:1.430431
[79]    train-aft-nloglik:1.399734  val-aft-nloglik:1.430156
[80]    train-aft-nloglik:1.399408  val-aft-nloglik:1.429913
[81]    train-aft-nloglik:1.399061  val-aft-nloglik:1.429710
[82]    train-aft-nloglik:1.398736  val-aft-nloglik:1.429680
[83]    train-aft-nloglik:1.398460  val-aft-nloglik:1.429571
[84]    train-aft-nloglik:1.398213  val-aft-nloglik:1.429462
[85]    train-aft-nloglik:1.397947  val-aft-nloglik:1.429285
[86]    train-aft-nloglik:1.397683  val-aft-nloglik:1.429176
[87]    train-aft-nloglik:1.397443  val-aft-nloglik:1.429101
[88]    train-aft-nloglik:1.397217  val-aft-nloglik:1.429057
[89]    train-aft-nloglik:1.396999  val-aft-nloglik:1.428943
[90]    train-aft-nloglik:1.396786  val-aft-nloglik:1.428856
[91]    train-aft-nloglik:1.396549  val-aft-nloglik:1.428822
[92]    train-aft-nloglik:1.396360  val-aft-nloglik:1.428734
[93]    train-aft-nloglik:1.396185  val-aft-nloglik:1.428603
[94]    train-aft-nloglik:1.395936  val-aft-nloglik:1.428581
[95]    train-aft-nloglik:1.395760  val-aft-nloglik:1.428503
[96]    train-aft-nloglik:1.395566  val-aft-nloglik:1.428387
[97]    train-aft-nloglik:1.395354  val-aft-nloglik:1.428314
```

```
[98]    train-aft-nloglik:1.395203  val-aft-nloglik:1.428234
[99]    train-aft-nloglik:1.394994  val-aft-nloglik:1.428224
[100]   train-aft-nloglik:1.394824  val-aft-nloglik:1.428167
```

```
# SHAP?
```