

Build Survival Model: XGBoost

Mingcheng Hu

Table of contents

Load Data	2
XGBoost	5
Data Preparation	5
Hyperparameter Tuning	6
Variable Selection	8
Cross Validation to Select the Best Number of Features	11
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/youkias/BIOS-Material/BIOS992/utils/csv_utils.r")
# * Don't use setwd() for Quarto documents!
# setwd("/work/users/y/u/youkias/BIOS-Material/BIOS992/data")

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

```
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)
```

```
# 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: partial"
```

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

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

Load Data

```
if (include_statin == "yes") {
  data_train <-
  ↪ read.csv(paste0("/work/users/y/u/youkias/BIOS-Material/BIOS992/data/train_data_",
  ↪ impute_type_str, "_statin.csv"),
    header = TRUE
  )
} else {
  data_train <-
  ↪ read.csv(paste0("/work/users/y/u/youkias/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    75
```

```
colnames(data)
```

```

[1] "event"           "time"
[3] "HRV_MeanNN"      "HRV_SDNN"
[5] "HRV_RMSSD"       "HRV_SSD"
[7] "HRV_CVNN"        "HRV_CVSD"
[9] "HRV_MedianNN"    "HRV_MadNN"
[11] "HRV_MCVNN"       "HRV_IQRNN"
[13] "HRV_SDRMSSD"     "HRV_Prc20NN"
[15] "HRV_Prc80NN"     "HRV_pNN50"
[17] "HRV_pNN20"       "HRV_MinNN"
[19] "HRV_MaxNN"       "HRV_HTI"
[21] "HRV_TINN"        "HRV_LF"
[23] "HRV_HF"          "HRV_VHF"
[25] "HRV_TP"          "HRV_LFHF"
[27] "HRV_LFn"         "HRV_HFn"
[29] "HRV_LnHF"        "HRV_SD1"
[31] "HRV_SD2"         "HRV_SD1SD2"
[33] "HRV_S"           "HRV_CSI"
[35] "HRV_CVI"         "HRV_CSI_Modified"
[37] "HRV_PIP"         "HRV_IALS"
[39] "HRV_PSS"         "HRV_PAS"
[41] "HRV_GI"          "HRV_SI"
[43] "HRV_AI"          "HRV_PI"

```

[45] "HRV_C1d"	"HRV_C1a"
[47] "HRV_SD1d"	"HRV_SD1a"
[49] "HRV_C2d"	"HRV_C2a"
[51] "HRV_SD2d"	"HRV_SD2a"
[53] "HRV_Cd"	"HRV_Ca"
[55] "HRV_SDNNd"	"HRV_SDNNa"
[57] "HRV_ApEn"	"HRV_ShanEn"
[59] "HRV_FuzzyEn"	"HRV_MSEn"
[61] "HRV_CMSEn"	"HRV_RCMSEn"
[63] "HRV_CD"	"HRV_HFD"
[65] "HRV_KFD"	"HRV_LZC"
[67] "HRV_DFA_alpha1"	"HRV_MFDFA_alpha1_Width"
[69] "HRV_MFDFA_alpha1_Peak"	"HRV_MFDFA_alpha1_Mean"
[71] "HRV_MFDFA_alpha1_Max"	"HRV_MFDFA_alpha1_Delta"
[73] "HRV_MFDFA_alpha1_Asymmetry"	"HRV_MFDFA_alpha1_Fluctuation"
[75] "HRV_MFDFA_alpha1_Increment"	

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

```
# * 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

```
total_x <- as.matrix(data %>% select(-c(time, event)))
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 <- as.matrix(train_data %>% select(-c(time, event)))
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,
  label_lower_bound = train_y_lower_bound,
  label_upper_bound = train_y_upper_bound
)

val_x <- as.matrix(val_data %>% select(-c(time, event)))
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,
  label_lower_bound = val_y_lower_bound,
  label_upper_bound = val_y_upper_bound
)
```

Hyperparameter Tuning

Since there is no

```
# 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] 0.08015368
```

```
$max_depth
```

```
[1] 8
```

```
$min_child_weight
```

```
[1] 40.66065
```

```
$reg_alpha
```

```
[1] 0.3062108
```

```
$reg_lambda
```

```
[1] 3.067648
```

```
$aft_loss_distribution_scale
```

```
[1] 1.575629
```

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)
)

```

[1] train-aft-nloglik:17.237924 val-aft-nloglik:17.249140
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:14.931642 val-aft-nloglik:14.944015
[3] train-aft-nloglik:12.971291 val-aft-nloglik:12.984750
[4] train-aft-nloglik:11.303941 val-aft-nloglik:11.318424
[5] train-aft-nloglik:9.884827  val-aft-nloglik:9.900274
[6] train-aft-nloglik:8.676091  val-aft-nloglik:8.692448
[7] train-aft-nloglik:7.645716  val-aft-nloglik:7.662934
[8] train-aft-nloglik:6.766627  val-aft-nloglik:6.784660
[9] train-aft-nloglik:6.015929  val-aft-nloglik:6.034734
[10]  train-aft-nloglik:5.374257  val-aft-nloglik:5.393796
[11]  train-aft-nloglik:4.825233  val-aft-nloglik:4.845470
[12]  train-aft-nloglik:4.355001  val-aft-nloglik:4.375902
[13]  train-aft-nloglik:3.951835  val-aft-nloglik:3.973369
[14]  train-aft-nloglik:3.605809  val-aft-nloglik:3.627949
[15]  train-aft-nloglik:3.308518  val-aft-nloglik:3.331236
[16]  train-aft-nloglik:3.052837  val-aft-nloglik:3.076108
[17]  train-aft-nloglik:2.832726  val-aft-nloglik:2.856527
[18]  train-aft-nloglik:2.643055  val-aft-nloglik:2.667365
[19]  train-aft-nloglik:2.479455  val-aft-nloglik:2.504312
[20]  train-aft-nloglik:2.338220  val-aft-nloglik:2.363578
[21]  train-aft-nloglik:2.216186  val-aft-nloglik:2.241997
[22]  train-aft-nloglik:2.110669  val-aft-nloglik:2.136917
[23]  train-aft-nloglik:2.019376  val-aft-nloglik:2.045998
[24]  train-aft-nloglik:1.940321  val-aft-nloglik:1.967250
[25]  train-aft-nloglik:1.871833  val-aft-nloglik:1.899237
[26]  train-aft-nloglik:1.812425  val-aft-nloglik:1.840351
[27]  train-aft-nloglik:1.760939  val-aft-nloglik:1.789177
[28]  train-aft-nloglik:1.716218  val-aft-nloglik:1.745041
[29]  train-aft-nloglik:1.677449  val-aft-nloglik:1.706659
[30]  train-aft-nloglik:1.643740  val-aft-nloglik:1.673399
[31]  train-aft-nloglik:1.614446  val-aft-nloglik:1.644687
[32]  train-aft-nloglik:1.588715  val-aft-nloglik:1.619942
[33]  train-aft-nloglik:1.566461  val-aft-nloglik:1.598350
[34]  train-aft-nloglik:1.546902  val-aft-nloglik:1.579422

```

[35]	train-aft-nloglik:1.529851	val-aft-nloglik:1.563185
[36]	train-aft-nloglik:1.514726	val-aft-nloglik:1.549051
[37]	train-aft-nloglik:1.501576	val-aft-nloglik:1.536884
[38]	train-aft-nloglik:1.490280	val-aft-nloglik:1.526291
[39]	train-aft-nloglik:1.480382	val-aft-nloglik:1.517176
[40]	train-aft-nloglik:1.471454	val-aft-nloglik:1.509231
[41]	train-aft-nloglik:1.463477	val-aft-nloglik:1.502377
[42]	train-aft-nloglik:1.456568	val-aft-nloglik:1.496440
[43]	train-aft-nloglik:1.450640	val-aft-nloglik:1.491256
[44]	train-aft-nloglik:1.445161	val-aft-nloglik:1.486817
[45]	train-aft-nloglik:1.440237	val-aft-nloglik:1.482935
[46]	train-aft-nloglik:1.435668	val-aft-nloglik:1.479411
[47]	train-aft-nloglik:1.432179	val-aft-nloglik:1.476618
[48]	train-aft-nloglik:1.428766	val-aft-nloglik:1.474576
[49]	train-aft-nloglik:1.425855	val-aft-nloglik:1.472625
[50]	train-aft-nloglik:1.423006	val-aft-nloglik:1.470691
[51]	train-aft-nloglik:1.420581	val-aft-nloglik:1.469320
[52]	train-aft-nloglik:1.418658	val-aft-nloglik:1.467982
[53]	train-aft-nloglik:1.416718	val-aft-nloglik:1.466714
[54]	train-aft-nloglik:1.415081	val-aft-nloglik:1.465725
[55]	train-aft-nloglik:1.413370	val-aft-nloglik:1.464926
[56]	train-aft-nloglik:1.411867	val-aft-nloglik:1.464264
[57]	train-aft-nloglik:1.410387	val-aft-nloglik:1.463849
[58]	train-aft-nloglik:1.408998	val-aft-nloglik:1.463396
[59]	train-aft-nloglik:1.407975	val-aft-nloglik:1.462962
[60]	train-aft-nloglik:1.406885	val-aft-nloglik:1.462911
[61]	train-aft-nloglik:1.405944	val-aft-nloglik:1.462575
[62]	train-aft-nloglik:1.405351	val-aft-nloglik:1.462463
[63]	train-aft-nloglik:1.404275	val-aft-nloglik:1.462273
[64]	train-aft-nloglik:1.403632	val-aft-nloglik:1.462170
[65]	train-aft-nloglik:1.402936	val-aft-nloglik:1.462136
[66]	train-aft-nloglik:1.402063	val-aft-nloglik:1.461992
[67]	train-aft-nloglik:1.401657	val-aft-nloglik:1.461998
[68]	train-aft-nloglik:1.400665	val-aft-nloglik:1.461889
[69]	train-aft-nloglik:1.399943	val-aft-nloglik:1.461801
[70]	train-aft-nloglik:1.399414	val-aft-nloglik:1.461765
[71]	train-aft-nloglik:1.398703	val-aft-nloglik:1.461806
[72]	train-aft-nloglik:1.397704	val-aft-nloglik:1.461802
[73]	train-aft-nloglik:1.397330	val-aft-nloglik:1.461938
[74]	train-aft-nloglik:1.397062	val-aft-nloglik:1.462044
[75]	train-aft-nloglik:1.396177	val-aft-nloglik:1.462106
[76]	train-aft-nloglik:1.395703	val-aft-nloglik:1.462163
[77]	train-aft-nloglik:1.395386	val-aft-nloglik:1.462366

```

[78]    train-aft-nloglik:1.394566    val-aft-nloglik:1.462212
[79]    train-aft-nloglik:1.393781    val-aft-nloglik:1.462275
[80]    train-aft-nloglik:1.393133    val-aft-nloglik:1.462353
Stopping. Best iteration:
[70]    train-aft-nloglik:1.399414    val-aft-nloglik:1.461765

```

```

# 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]
    print(paste0("Selecting ", num_vars, " variables"))
    fold_errors <- sapply(folds, function(fold_idx) {
      # * We take all training data and validation data and then split them
      ↪ into folds.
      train_x_fold <- as.matrix(total_x[-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 <- as.matrix(total_x[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

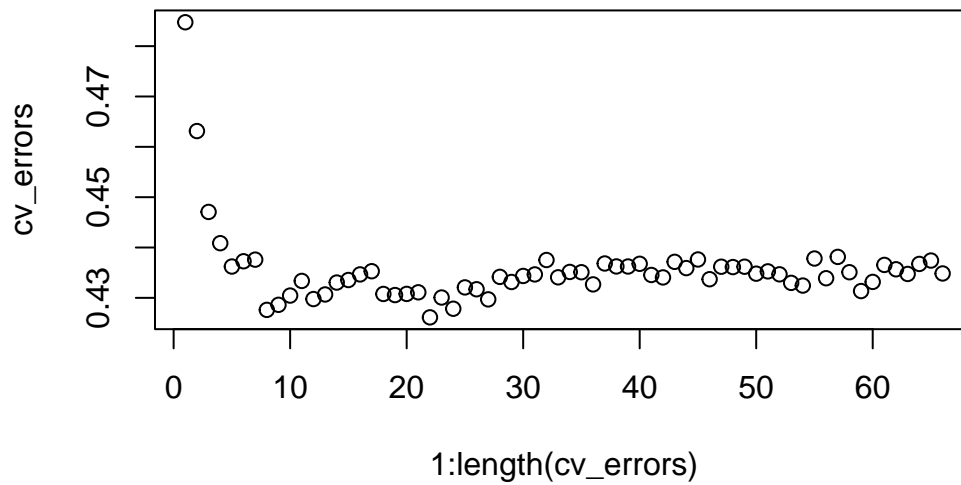
# Use C-index to measure the performance of the model
1 - concordance.index(pred, val_y_fold$time,
↪ val_y_fold$event)$c.index
})
print(mean(fold_errors))
mean(fold_errors)
}, title = "Cross Validation to Select the Best Number of Features")

```

```

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

```



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

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

```
[1] "The best number of features to retain is 22"
```

Model Fitting

```
dtrain_selected <- xgb.DMatrix(
  data = train_x[, vars_selected],
  label_lower_bound = train_y_lower_bound,
  label_upper_bound = train_y_upper_bound
)
dval_selected <- xgb.DMatrix(
  data = val_x[, vars_selected],
  label_lower_bound = val_y_lower_bound,
  label_upper_bound = val_y_upper_bound
)
```

```
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:17.237924 val-aft-nloglik:17.249140
```

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:14.931642 val-aft-nloglik:14.944015
```

```
[3] train-aft-nloglik:12.971291 val-aft-nloglik:12.984750
```

```
[4] train-aft-nloglik:11.303941 val-aft-nloglik:11.318424
```

```
[5] train-aft-nloglik:9.884827 val-aft-nloglik:9.900274
```

```
[6] train-aft-nloglik:8.676091 val-aft-nloglik:8.692448
```

```
[7] train-aft-nloglik:7.645716 val-aft-nloglik:7.662934
```

```
[8] train-aft-nloglik:6.766627 val-aft-nloglik:6.784660
```

```
[9] train-aft-nloglik:6.015929 val-aft-nloglik:6.034734
```

```
[10] train-aft-nloglik:5.374257 val-aft-nloglik:5.393796
```

```
[11] train-aft-nloglik:4.825233 val-aft-nloglik:4.845470
```

```
[12] train-aft-nloglik:4.355001 val-aft-nloglik:4.375902
```

```
[13] train-aft-nloglik:3.951835 val-aft-nloglik:3.973369
```

```
[14] train-aft-nloglik:3.605809 val-aft-nloglik:3.627949
```

```
[15] train-aft-nloglik:3.308518 val-aft-nloglik:3.331236
```

```
[16] train-aft-nloglik:3.052837 val-aft-nloglik:3.076108
```

```
[17] train-aft-nloglik:2.832726 val-aft-nloglik:2.856527
```

```
[18] train-aft-nloglik:2.643055 val-aft-nloglik:2.667365
```

```
[19] train-aft-nloglik:2.479459 val-aft-nloglik:2.504243
```

```
[20] train-aft-nloglik:2.338226 val-aft-nloglik:2.363545
```

```
[21] train-aft-nloglik:2.216192 val-aft-nloglik:2.241902
```

```
[22] train-aft-nloglik:2.110679 val-aft-nloglik:2.136824
```

```
[23] train-aft-nloglik:2.019377 val-aft-nloglik:2.046020
```

[24]	train-aft-nloglik:1.940321	val-aft-nloglik:1.967423
[25]	train-aft-nloglik:1.871847	val-aft-nloglik:1.899276
[26]	train-aft-nloglik:1.812481	val-aft-nloglik:1.840440
[27]	train-aft-nloglik:1.761025	val-aft-nloglik:1.789317
[28]	train-aft-nloglik:1.716294	val-aft-nloglik:1.745087
[29]	train-aft-nloglik:1.677491	val-aft-nloglik:1.706844
[30]	train-aft-nloglik:1.643798	val-aft-nloglik:1.673551
[31]	train-aft-nloglik:1.614515	val-aft-nloglik:1.644766
[32]	train-aft-nloglik:1.589026	val-aft-nloglik:1.619814
[33]	train-aft-nloglik:1.566716	val-aft-nloglik:1.598269
[34]	train-aft-nloglik:1.547313	val-aft-nloglik:1.579464
[35]	train-aft-nloglik:1.530385	val-aft-nloglik:1.563233
[36]	train-aft-nloglik:1.515574	val-aft-nloglik:1.549168
[37]	train-aft-nloglik:1.502769	val-aft-nloglik:1.536958
[38]	train-aft-nloglik:1.491370	val-aft-nloglik:1.526370
[39]	train-aft-nloglik:1.481464	val-aft-nloglik:1.517297
[40]	train-aft-nloglik:1.472865	val-aft-nloglik:1.509541
[41]	train-aft-nloglik:1.465445	val-aft-nloglik:1.502688
[42]	train-aft-nloglik:1.458905	val-aft-nloglik:1.496975
[43]	train-aft-nloglik:1.453120	val-aft-nloglik:1.491872
[44]	train-aft-nloglik:1.448171	val-aft-nloglik:1.487493
[45]	train-aft-nloglik:1.443306	val-aft-nloglik:1.483578
[46]	train-aft-nloglik:1.439386	val-aft-nloglik:1.480257
[47]	train-aft-nloglik:1.435910	val-aft-nloglik:1.477359
[48]	train-aft-nloglik:1.432941	val-aft-nloglik:1.474907
[49]	train-aft-nloglik:1.429906	val-aft-nloglik:1.472625
[50]	train-aft-nloglik:1.427198	val-aft-nloglik:1.470842
[51]	train-aft-nloglik:1.424686	val-aft-nloglik:1.469220
[52]	train-aft-nloglik:1.422742	val-aft-nloglik:1.467955
[53]	train-aft-nloglik:1.420575	val-aft-nloglik:1.466773
[54]	train-aft-nloglik:1.418689	val-aft-nloglik:1.465839
[55]	train-aft-nloglik:1.417103	val-aft-nloglik:1.465056
[56]	train-aft-nloglik:1.415785	val-aft-nloglik:1.464453
[57]	train-aft-nloglik:1.414313	val-aft-nloglik:1.463894
[58]	train-aft-nloglik:1.413180	val-aft-nloglik:1.463392
[59]	train-aft-nloglik:1.412177	val-aft-nloglik:1.463137
[60]	train-aft-nloglik:1.411317	val-aft-nloglik:1.462964
[61]	train-aft-nloglik:1.410374	val-aft-nloglik:1.462658
[62]	train-aft-nloglik:1.409558	val-aft-nloglik:1.462401
[63]	train-aft-nloglik:1.408667	val-aft-nloglik:1.462155
[64]	train-aft-nloglik:1.407895	val-aft-nloglik:1.462103
[65]	train-aft-nloglik:1.407085	val-aft-nloglik:1.461974
[66]	train-aft-nloglik:1.406355	val-aft-nloglik:1.461787

```

[67] train-aft-nloglik:1.405692 val-aft-nloglik:1.461703
[68] train-aft-nloglik:1.405032 val-aft-nloglik:1.461783
[69] train-aft-nloglik:1.404392 val-aft-nloglik:1.461856
[70] train-aft-nloglik:1.403906 val-aft-nloglik:1.461877
[71] train-aft-nloglik:1.402889 val-aft-nloglik:1.461972
[72] train-aft-nloglik:1.402417 val-aft-nloglik:1.461973
[73] train-aft-nloglik:1.401703 val-aft-nloglik:1.461953
[74] train-aft-nloglik:1.400710 val-aft-nloglik:1.462175
[75] train-aft-nloglik:1.400350 val-aft-nloglik:1.462117
[76] train-aft-nloglik:1.399894 val-aft-nloglik:1.462018
[77] train-aft-nloglik:1.399372 val-aft-nloglik:1.462187
Stopping. Best iteration:
[67] train-aft-nloglik:1.405692 val-aft-nloglik:1.461703

```

```

dtrain_full <- xgb.DMatrix(
  data = train_x,
  label_lower_bound = train_y_lower_bound,
  label_upper_bound = train_y_upper_bound
)
# 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:17.237924 val-aft-nloglik:17.249140
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:14.931642 val-aft-nloglik:14.944015
[3] train-aft-nloglik:12.971291 val-aft-nloglik:12.984750
[4] train-aft-nloglik:11.303941 val-aft-nloglik:11.318424

```


[5]	train-aft-nloglik:9.884827	val-aft-nloglik:9.900274
[6]	train-aft-nloglik:8.676091	val-aft-nloglik:8.692448
[7]	train-aft-nloglik:7.645716	val-aft-nloglik:7.662934
[8]	train-aft-nloglik:6.766627	val-aft-nloglik:6.784660
[9]	train-aft-nloglik:6.015929	val-aft-nloglik:6.034734
[10]	train-aft-nloglik:5.374257	val-aft-nloglik:5.393796
[11]	train-aft-nloglik:4.825233	val-aft-nloglik:4.845470
[12]	train-aft-nloglik:4.355001	val-aft-nloglik:4.375902
[13]	train-aft-nloglik:3.951835	val-aft-nloglik:3.973369
[14]	train-aft-nloglik:3.605809	val-aft-nloglik:3.627949
[15]	train-aft-nloglik:3.308518	val-aft-nloglik:3.331236
[16]	train-aft-nloglik:3.052837	val-aft-nloglik:3.076108
[17]	train-aft-nloglik:2.832726	val-aft-nloglik:2.856527
[18]	train-aft-nloglik:2.643055	val-aft-nloglik:2.667365
[19]	train-aft-nloglik:2.479455	val-aft-nloglik:2.504312
[20]	train-aft-nloglik:2.338220	val-aft-nloglik:2.363578
[21]	train-aft-nloglik:2.216186	val-aft-nloglik:2.241997
[22]	train-aft-nloglik:2.110669	val-aft-nloglik:2.136917
[23]	train-aft-nloglik:2.019376	val-aft-nloglik:2.045998
[24]	train-aft-nloglik:1.940321	val-aft-nloglik:1.967250
[25]	train-aft-nloglik:1.871833	val-aft-nloglik:1.899237
[26]	train-aft-nloglik:1.812425	val-aft-nloglik:1.840351
[27]	train-aft-nloglik:1.760939	val-aft-nloglik:1.789177
[28]	train-aft-nloglik:1.716218	val-aft-nloglik:1.745041
[29]	train-aft-nloglik:1.677449	val-aft-nloglik:1.706659
[30]	train-aft-nloglik:1.643740	val-aft-nloglik:1.673399
[31]	train-aft-nloglik:1.614446	val-aft-nloglik:1.644687
[32]	train-aft-nloglik:1.588715	val-aft-nloglik:1.619942
[33]	train-aft-nloglik:1.566461	val-aft-nloglik:1.598350
[34]	train-aft-nloglik:1.546902	val-aft-nloglik:1.579422
[35]	train-aft-nloglik:1.529851	val-aft-nloglik:1.563185
[36]	train-aft-nloglik:1.514726	val-aft-nloglik:1.549051
[37]	train-aft-nloglik:1.501576	val-aft-nloglik:1.536884
[38]	train-aft-nloglik:1.490280	val-aft-nloglik:1.526291
[39]	train-aft-nloglik:1.480382	val-aft-nloglik:1.517176
[40]	train-aft-nloglik:1.471454	val-aft-nloglik:1.509231
[41]	train-aft-nloglik:1.463477	val-aft-nloglik:1.502377
[42]	train-aft-nloglik:1.456568	val-aft-nloglik:1.496440
[43]	train-aft-nloglik:1.450640	val-aft-nloglik:1.491256
[44]	train-aft-nloglik:1.445161	val-aft-nloglik:1.486817
[45]	train-aft-nloglik:1.440237	val-aft-nloglik:1.482935
[46]	train-aft-nloglik:1.435668	val-aft-nloglik:1.479411
[47]	train-aft-nloglik:1.432179	val-aft-nloglik:1.476618

[48]	train-aft-nloglik:1.428766	val-aft-nloglik:1.474576
[49]	train-aft-nloglik:1.425855	val-aft-nloglik:1.472625
[50]	train-aft-nloglik:1.423006	val-aft-nloglik:1.470691
[51]	train-aft-nloglik:1.420581	val-aft-nloglik:1.469320
[52]	train-aft-nloglik:1.418658	val-aft-nloglik:1.467982
[53]	train-aft-nloglik:1.416718	val-aft-nloglik:1.466714
[54]	train-aft-nloglik:1.415081	val-aft-nloglik:1.465725
[55]	train-aft-nloglik:1.413370	val-aft-nloglik:1.464926
[56]	train-aft-nloglik:1.411867	val-aft-nloglik:1.464264
[57]	train-aft-nloglik:1.410387	val-aft-nloglik:1.463849
[58]	train-aft-nloglik:1.408998	val-aft-nloglik:1.463396
[59]	train-aft-nloglik:1.407975	val-aft-nloglik:1.462962
[60]	train-aft-nloglik:1.406885	val-aft-nloglik:1.462911
[61]	train-aft-nloglik:1.405944	val-aft-nloglik:1.462575
[62]	train-aft-nloglik:1.405351	val-aft-nloglik:1.462463
[63]	train-aft-nloglik:1.404275	val-aft-nloglik:1.462273
[64]	train-aft-nloglik:1.403632	val-aft-nloglik:1.462170
[65]	train-aft-nloglik:1.402936	val-aft-nloglik:1.462136
[66]	train-aft-nloglik:1.402063	val-aft-nloglik:1.461992
[67]	train-aft-nloglik:1.401657	val-aft-nloglik:1.461998
[68]	train-aft-nloglik:1.400665	val-aft-nloglik:1.461889
[69]	train-aft-nloglik:1.399943	val-aft-nloglik:1.461801
[70]	train-aft-nloglik:1.399414	val-aft-nloglik:1.461765
[71]	train-aft-nloglik:1.398703	val-aft-nloglik:1.461806
[72]	train-aft-nloglik:1.397704	val-aft-nloglik:1.461802
[73]	train-aft-nloglik:1.397330	val-aft-nloglik:1.461938
[74]	train-aft-nloglik:1.397062	val-aft-nloglik:1.462044
[75]	train-aft-nloglik:1.396177	val-aft-nloglik:1.462106
[76]	train-aft-nloglik:1.395703	val-aft-nloglik:1.462163
[77]	train-aft-nloglik:1.395386	val-aft-nloglik:1.462366
[78]	train-aft-nloglik:1.394566	val-aft-nloglik:1.462212
[79]	train-aft-nloglik:1.393781	val-aft-nloglik:1.462275
[80]	train-aft-nloglik:1.393133	val-aft-nloglik:1.462353

Stopping. Best iteration:

[70]	train-aft-nloglik:1.399414	val-aft-nloglik:1.461765
------	----------------------------	--------------------------

SHAP?