

Build Survival Model: XGBoost

Mingcheng Hu

Table of contents

Load Data	2
XGBoost	4
Data Preparation	4
Hyperparameter Tuning	5
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$adjust_type, "minimal") #
  ↳ options: "minimal", "partial", "full"
impute_type <- ifelse(exists("params"), params$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: minimal"
```

```
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    48
```

```
colnames(data)
```

```

[1] "event"           "time"
[3] "HRV_SD1"         "HRV_SD2"
[5] "HRV_SD1SD2"      "HRV_S"
[7] "HRV_CSI"         "HRV_CVI"
[9] "HRV_CSI_Modified" "HRV_PIP"
[11] "HRV_IALS"        "HRV_PSS"
[13] "HRV_PAS"         "HRV_GI"
[15] "HRV_SI"          "HRV_AI"
[17] "HRV_PI"          "HRV_C1d"
[19] "HRV_C1a"         "HRV_SD1d"
[21] "HRV_SD1a"        "HRV_C2d"
[23] "HRV_C2a"         "HRV_SD2d"
[25] "HRV_SD2a"        "HRV_Cd"
[27] "HRV_Ca"          "HRV_SDNNd"
[29] "HRV_SDNNa"       "HRV_ApEn"
[31] "HRV_ShanEn"      "HRV_FuzzyEn"
[33] "HRV_MSEn"        "HRV_CMSEn"
[35] "HRV_RCMSEn"      "HRV_CD"
[37] "HRV_HFD"         "HRV_KFD"
[39] "HRV_LZC"         "HRV_DFA_alpha1"
[41] "HRV_MFDFA_alpha1_Width" "HRV_MFDFA_alpha1_Peak"
[43] "HRV_MFDFA_alpha1_Mean"  "HRV_MFDFA_alpha1_Max"

```

```
[45] "HRV_MFDFA_alpha1_Delta"      "HRV_MFDFA_alpha1_Asymmetry"
[47] "HRV_MFDFA_alpha1_Fluctuation" "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 = 1000,
    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] 0.06626034

```

```

$max_depth
[1] 7

```

```
$min_child_weight  
[1] 52.77876
```

```
$reg_alpha  
[1] 0.00161188
```

```
$reg_lambda  
[1] 1.074741
```

```
$aft_loss_distribution_scale  
[1] 1.57089
```

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 = 1000,  
  early_stopping_rounds = 10,  
  watchlist = list(train = dtrain, val = dval)  
)
```

```
[1] train-aft-nloglik:17.780534 val-aft-nloglik:17.791474  
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:15.791891 val-aft-nloglik:15.803801  
[3] train-aft-nloglik:14.051916 val-aft-nloglik:14.064747  
[4] train-aft-nloglik:12.528906 val-aft-nloglik:12.542613  
[5] train-aft-nloglik:11.195230 val-aft-nloglik:11.209770  
[6] train-aft-nloglik:10.026806 val-aft-nloglik:10.042141
```


[7]	train-aft-nloglik:9.002649	val-aft-nloglik:9.018741
[8]	train-aft-nloglik:8.104468	val-aft-nloglik:8.121284
[9]	train-aft-nloglik:7.316325	val-aft-nloglik:7.333833
[10]	train-aft-nloglik:6.624332	val-aft-nloglik:6.642500
[11]	train-aft-nloglik:6.016382	val-aft-nloglik:6.035183
[12]	train-aft-nloglik:5.481928	val-aft-nloglik:5.501335
[13]	train-aft-nloglik:5.011772	val-aft-nloglik:5.031762
[14]	train-aft-nloglik:4.597901	val-aft-nloglik:4.618449
[15]	train-aft-nloglik:4.233325	val-aft-nloglik:4.254411
[16]	train-aft-nloglik:3.911947	val-aft-nloglik:3.933590
[17]	train-aft-nloglik:3.628446	val-aft-nloglik:3.650619
[18]	train-aft-nloglik:3.378183	val-aft-nloglik:3.400895
[19]	train-aft-nloglik:3.157111	val-aft-nloglik:3.180300
[20]	train-aft-nloglik:2.961688	val-aft-nloglik:2.985329
[21]	train-aft-nloglik:2.788832	val-aft-nloglik:2.812928
[22]	train-aft-nloglik:2.635831	val-aft-nloglik:2.660305
[23]	train-aft-nloglik:2.500327	val-aft-nloglik:2.525173
[24]	train-aft-nloglik:2.380239	val-aft-nloglik:2.405482
[25]	train-aft-nloglik:2.273724	val-aft-nloglik:2.299463
[26]	train-aft-nloglik:2.179243	val-aft-nloglik:2.205300
[27]	train-aft-nloglik:2.095348	val-aft-nloglik:2.121883
[28]	train-aft-nloglik:2.020852	val-aft-nloglik:2.047743
[29]	train-aft-nloglik:1.954619	val-aft-nloglik:1.981961
[30]	train-aft-nloglik:1.895796	val-aft-nloglik:1.923622
[31]	train-aft-nloglik:1.843413	val-aft-nloglik:1.871682
[32]	train-aft-nloglik:1.796780	val-aft-nloglik:1.825535
[33]	train-aft-nloglik:1.755316	val-aft-nloglik:1.784474
[34]	train-aft-nloglik:1.718370	val-aft-nloglik:1.747980
[35]	train-aft-nloglik:1.685421	val-aft-nloglik:1.715607
[36]	train-aft-nloglik:1.656152	val-aft-nloglik:1.686879
[37]	train-aft-nloglik:1.629914	val-aft-nloglik:1.661317
[38]	train-aft-nloglik:1.606442	val-aft-nloglik:1.638562
[39]	train-aft-nloglik:1.585665	val-aft-nloglik:1.618350
[40]	train-aft-nloglik:1.567183	val-aft-nloglik:1.600400
[41]	train-aft-nloglik:1.550570	val-aft-nloglik:1.584496
[42]	train-aft-nloglik:1.535856	val-aft-nloglik:1.570263
[43]	train-aft-nloglik:1.522640	val-aft-nloglik:1.557635
[44]	train-aft-nloglik:1.510762	val-aft-nloglik:1.546419
[45]	train-aft-nloglik:1.500359	val-aft-nloglik:1.536519
[46]	train-aft-nloglik:1.491006	val-aft-nloglik:1.527771
[47]	train-aft-nloglik:1.482731	val-aft-nloglik:1.520075
[48]	train-aft-nloglik:1.475246	val-aft-nloglik:1.513251
[49]	train-aft-nloglik:1.468573	val-aft-nloglik:1.507236

[50]	train-aft-nloglik:1.462707	val-aft-nloglik:1.501866
[51]	train-aft-nloglik:1.457337	val-aft-nloglik:1.497174
[52]	train-aft-nloglik:1.452422	val-aft-nloglik:1.492922
[53]	train-aft-nloglik:1.448012	val-aft-nloglik:1.489267
[54]	train-aft-nloglik:1.444251	val-aft-nloglik:1.485976
[55]	train-aft-nloglik:1.440740	val-aft-nloglik:1.483181
[56]	train-aft-nloglik:1.437486	val-aft-nloglik:1.480523
[57]	train-aft-nloglik:1.434607	val-aft-nloglik:1.478319
[58]	train-aft-nloglik:1.432182	val-aft-nloglik:1.476330
[59]	train-aft-nloglik:1.429686	val-aft-nloglik:1.474493
[60]	train-aft-nloglik:1.427706	val-aft-nloglik:1.473013
[61]	train-aft-nloglik:1.425826	val-aft-nloglik:1.471582
[62]	train-aft-nloglik:1.424287	val-aft-nloglik:1.470419
[63]	train-aft-nloglik:1.422807	val-aft-nloglik:1.469344
[64]	train-aft-nloglik:1.421498	val-aft-nloglik:1.468415
[65]	train-aft-nloglik:1.420218	val-aft-nloglik:1.467604
[66]	train-aft-nloglik:1.418959	val-aft-nloglik:1.466958
[67]	train-aft-nloglik:1.418015	val-aft-nloglik:1.466389
[68]	train-aft-nloglik:1.417169	val-aft-nloglik:1.465921
[69]	train-aft-nloglik:1.416316	val-aft-nloglik:1.465506
[70]	train-aft-nloglik:1.415578	val-aft-nloglik:1.465197
[71]	train-aft-nloglik:1.414927	val-aft-nloglik:1.464961
[72]	train-aft-nloglik:1.413828	val-aft-nloglik:1.464631
[73]	train-aft-nloglik:1.412967	val-aft-nloglik:1.464448
[74]	train-aft-nloglik:1.412046	val-aft-nloglik:1.464125
[75]	train-aft-nloglik:1.411272	val-aft-nloglik:1.464043
[76]	train-aft-nloglik:1.410832	val-aft-nloglik:1.463930
[77]	train-aft-nloglik:1.410372	val-aft-nloglik:1.463827
[78]	train-aft-nloglik:1.409732	val-aft-nloglik:1.463607
[79]	train-aft-nloglik:1.409311	val-aft-nloglik:1.463505
[80]	train-aft-nloglik:1.408803	val-aft-nloglik:1.463366
[81]	train-aft-nloglik:1.408155	val-aft-nloglik:1.463365
[82]	train-aft-nloglik:1.407601	val-aft-nloglik:1.463318
[83]	train-aft-nloglik:1.407210	val-aft-nloglik:1.463267
[84]	train-aft-nloglik:1.406821	val-aft-nloglik:1.463211
[85]	train-aft-nloglik:1.406515	val-aft-nloglik:1.463163
[86]	train-aft-nloglik:1.406057	val-aft-nloglik:1.463180
[87]	train-aft-nloglik:1.405862	val-aft-nloglik:1.463137
[88]	train-aft-nloglik:1.405605	val-aft-nloglik:1.463179
[89]	train-aft-nloglik:1.405389	val-aft-nloglik:1.463238
[90]	train-aft-nloglik:1.405144	val-aft-nloglik:1.463321
[91]	train-aft-nloglik:1.404976	val-aft-nloglik:1.463337
[92]	train-aft-nloglik:1.404598	val-aft-nloglik:1.463273

```

[93]   train-aft-nloglik:1.404446   val-aft-nloglik:1.463237
[94]   train-aft-nloglik:1.404229   val-aft-nloglik:1.463173
[95]   train-aft-nloglik:1.404071   val-aft-nloglik:1.463162
[96]   train-aft-nloglik:1.403936   val-aft-nloglik:1.463282
[97]   train-aft-nloglik:1.403797   val-aft-nloglik:1.463303
Stopping. Best iteration:
[87]   train-aft-nloglik:1.405862   val-aft-nloglik:1.463137

```

```

# 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 = 1000,
        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

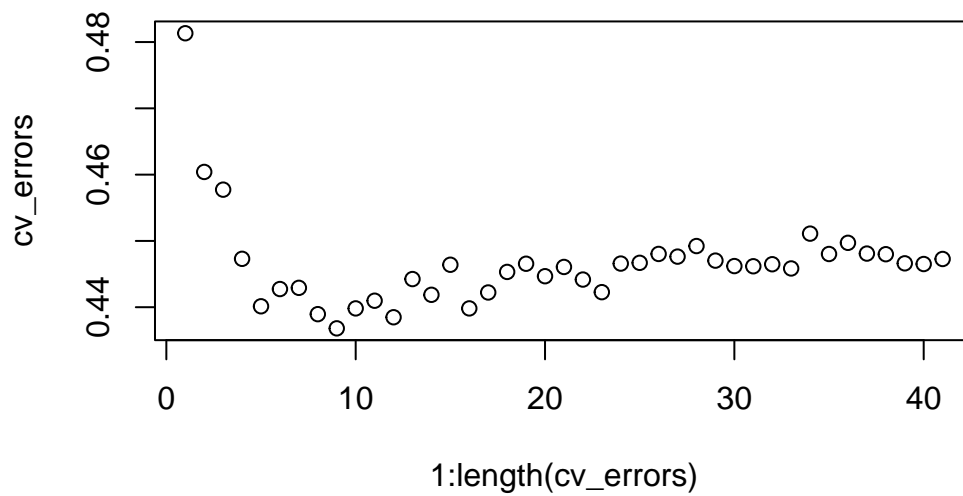
    # Use C-index to measure the performance of the model
    1 - concordance.index(pred, val_y_fold$time,
        ↪ val_y_fold$event)$c.index
    })
    print(mean(fold_errors))
    mean(fold_errors)
}, title = "Cross Validation to Select the Best Number of Features")

```

```

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 9"
```

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 = 1000,
  early_stopping_rounds = 10,
  watchlist = list(train = dtrain_selected, val = dval_selected)
)
```

[1] train-aft-nloglik:17.780534 val-aft-nloglik:17.791474

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:15.791891 val-aft-nloglik:15.803801

[3] train-aft-nloglik:14.051916 val-aft-nloglik:14.064747

[4] train-aft-nloglik:12.528906 val-aft-nloglik:12.542613

[5] train-aft-nloglik:11.195230 val-aft-nloglik:11.209770

[6] train-aft-nloglik:10.026806 val-aft-nloglik:10.042141

[7] train-aft-nloglik:9.002649 val-aft-nloglik:9.018741

[8] train-aft-nloglik:8.104468 val-aft-nloglik:8.121284

[9] train-aft-nloglik:7.316325 val-aft-nloglik:7.333833

[10] train-aft-nloglik:6.624332 val-aft-nloglik:6.642500

[11] train-aft-nloglik:6.016382 val-aft-nloglik:6.035183

[12] train-aft-nloglik:5.481928 val-aft-nloglik:5.501335

[13] train-aft-nloglik:5.011772 val-aft-nloglik:5.031762

[14] train-aft-nloglik:4.597901 val-aft-nloglik:4.618449

[15] train-aft-nloglik:4.233325 val-aft-nloglik:4.254411

[16] train-aft-nloglik:3.911951 val-aft-nloglik:3.933553

[17] train-aft-nloglik:3.628453 val-aft-nloglik:3.650625

[18] train-aft-nloglik:3.378196 val-aft-nloglik:3.400881

[19] train-aft-nloglik:3.157126 val-aft-nloglik:3.180285

[20] train-aft-nloglik:2.961708 val-aft-nloglik:2.985323

[21] train-aft-nloglik:2.788853 val-aft-nloglik:2.812928

[22] train-aft-nloglik:2.635854 val-aft-nloglik:2.660339

[23] train-aft-nloglik:2.500352 val-aft-nloglik:2.525213

[24]	train-aft-nloglik:2.380271	val-aft-nloglik:2.405492
[25]	train-aft-nloglik:2.273803	val-aft-nloglik:2.299368
[26]	train-aft-nloglik:2.179336	val-aft-nloglik:2.205298
[27]	train-aft-nloglik:2.095486	val-aft-nloglik:2.121806
[28]	train-aft-nloglik:2.021018	val-aft-nloglik:2.047680
[29]	train-aft-nloglik:1.954850	val-aft-nloglik:1.981935
[30]	train-aft-nloglik:1.896047	val-aft-nloglik:1.923562
[31]	train-aft-nloglik:1.843771	val-aft-nloglik:1.871675
[32]	train-aft-nloglik:1.797265	val-aft-nloglik:1.825671
[33]	train-aft-nloglik:1.755851	val-aft-nloglik:1.784838
[34]	train-aft-nloglik:1.719027	val-aft-nloglik:1.748436
[35]	train-aft-nloglik:1.686173	val-aft-nloglik:1.716173
[36]	train-aft-nloglik:1.656929	val-aft-nloglik:1.687469
[37]	train-aft-nloglik:1.630867	val-aft-nloglik:1.661871
[38]	train-aft-nloglik:1.607707	val-aft-nloglik:1.639157
[39]	train-aft-nloglik:1.587003	val-aft-nloglik:1.619006
[40]	train-aft-nloglik:1.568551	val-aft-nloglik:1.601043
[41]	train-aft-nloglik:1.552196	val-aft-nloglik:1.585109
[42]	train-aft-nloglik:1.537673	val-aft-nloglik:1.570989
[43]	train-aft-nloglik:1.524602	val-aft-nloglik:1.558430
[44]	train-aft-nloglik:1.513017	val-aft-nloglik:1.547262
[45]	train-aft-nloglik:1.502620	val-aft-nloglik:1.537305
[46]	train-aft-nloglik:1.493393	val-aft-nloglik:1.528591
[47]	train-aft-nloglik:1.485147	val-aft-nloglik:1.520825
[48]	train-aft-nloglik:1.477847	val-aft-nloglik:1.513944
[49]	train-aft-nloglik:1.471179	val-aft-nloglik:1.507833
[50]	train-aft-nloglik:1.465227	val-aft-nloglik:1.502522
[51]	train-aft-nloglik:1.460083	val-aft-nloglik:1.497781
[52]	train-aft-nloglik:1.455323	val-aft-nloglik:1.493583
[53]	train-aft-nloglik:1.450955	val-aft-nloglik:1.489898
[54]	train-aft-nloglik:1.447314	val-aft-nloglik:1.486654
[55]	train-aft-nloglik:1.443897	val-aft-nloglik:1.483843
[56]	train-aft-nloglik:1.441060	val-aft-nloglik:1.481293
[57]	train-aft-nloglik:1.438376	val-aft-nloglik:1.479090
[58]	train-aft-nloglik:1.436047	val-aft-nloglik:1.477189
[59]	train-aft-nloglik:1.434001	val-aft-nloglik:1.475499
[60]	train-aft-nloglik:1.431914	val-aft-nloglik:1.473994
[61]	train-aft-nloglik:1.430005	val-aft-nloglik:1.472687
[62]	train-aft-nloglik:1.428358	val-aft-nloglik:1.471493
[63]	train-aft-nloglik:1.426968	val-aft-nloglik:1.470438
[64]	train-aft-nloglik:1.425701	val-aft-nloglik:1.469496
[65]	train-aft-nloglik:1.424459	val-aft-nloglik:1.468707
[66]	train-aft-nloglik:1.423222	val-aft-nloglik:1.468013

[67]	train-aft-nloglik:1.422270	val-aft-nloglik:1.467383
[68]	train-aft-nloglik:1.421375	val-aft-nloglik:1.466866
[69]	train-aft-nloglik:1.420474	val-aft-nloglik:1.466434
[70]	train-aft-nloglik:1.419735	val-aft-nloglik:1.466114
[71]	train-aft-nloglik:1.419070	val-aft-nloglik:1.465740
[72]	train-aft-nloglik:1.418551	val-aft-nloglik:1.465472
[73]	train-aft-nloglik:1.417942	val-aft-nloglik:1.465169
[74]	train-aft-nloglik:1.417413	val-aft-nloglik:1.464916
[75]	train-aft-nloglik:1.416782	val-aft-nloglik:1.464769
[76]	train-aft-nloglik:1.416450	val-aft-nloglik:1.464609
[77]	train-aft-nloglik:1.416122	val-aft-nloglik:1.464531
[78]	train-aft-nloglik:1.415714	val-aft-nloglik:1.464470
[79]	train-aft-nloglik:1.415475	val-aft-nloglik:1.464440
[80]	train-aft-nloglik:1.415154	val-aft-nloglik:1.464400
[81]	train-aft-nloglik:1.414904	val-aft-nloglik:1.464345
[82]	train-aft-nloglik:1.414701	val-aft-nloglik:1.464337
[83]	train-aft-nloglik:1.414397	val-aft-nloglik:1.464322
[84]	train-aft-nloglik:1.414151	val-aft-nloglik:1.464329
[85]	train-aft-nloglik:1.413948	val-aft-nloglik:1.464222
[86]	train-aft-nloglik:1.413691	val-aft-nloglik:1.464132
[87]	train-aft-nloglik:1.413594	val-aft-nloglik:1.464128
[88]	train-aft-nloglik:1.413413	val-aft-nloglik:1.464122
[89]	train-aft-nloglik:1.413269	val-aft-nloglik:1.464107
[90]	train-aft-nloglik:1.413129	val-aft-nloglik:1.464129
[91]	train-aft-nloglik:1.412821	val-aft-nloglik:1.464147
[92]	train-aft-nloglik:1.412543	val-aft-nloglik:1.464129
[93]	train-aft-nloglik:1.412397	val-aft-nloglik:1.464129
[94]	train-aft-nloglik:1.411903	val-aft-nloglik:1.464109
[95]	train-aft-nloglik:1.411482	val-aft-nloglik:1.464056
[96]	train-aft-nloglik:1.411389	val-aft-nloglik:1.464054
[97]	train-aft-nloglik:1.411259	val-aft-nloglik:1.464066
[98]	train-aft-nloglik:1.411014	val-aft-nloglik:1.464026
[99]	train-aft-nloglik:1.410839	val-aft-nloglik:1.464047
[100]	train-aft-nloglik:1.410736	val-aft-nloglik:1.464089
[101]	train-aft-nloglik:1.410608	val-aft-nloglik:1.464127
[102]	train-aft-nloglik:1.410495	val-aft-nloglik:1.464152
[103]	train-aft-nloglik:1.410412	val-aft-nloglik:1.464192
[104]	train-aft-nloglik:1.410204	val-aft-nloglik:1.464159
[105]	train-aft-nloglik:1.410145	val-aft-nloglik:1.464179
[106]	train-aft-nloglik:1.409999	val-aft-nloglik:1.464238
[107]	train-aft-nloglik:1.409801	val-aft-nloglik:1.464358
[108]	train-aft-nloglik:1.409640	val-aft-nloglik:1.464316

Stopping. Best iteration:


```
[98]    train-aft-nloglik:1.411014  val-aft-nloglik:1.464026
```

```
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 = 1000,  
  early_stopping_rounds = 10,  
  watchlist = list(train = dtrain_full, val = dval)  
)
```

```
[1] train-aft-nloglik:17.780534 val-aft-nloglik:17.791474
```

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:15.791891 val-aft-nloglik:15.803801  
[3] train-aft-nloglik:14.051916 val-aft-nloglik:14.064747  
[4] train-aft-nloglik:12.528906 val-aft-nloglik:12.542613  
[5] train-aft-nloglik:11.195230 val-aft-nloglik:11.209770  
[6] train-aft-nloglik:10.026806 val-aft-nloglik:10.042141  
[7] train-aft-nloglik:9.002649  val-aft-nloglik:9.018741  
[8] train-aft-nloglik:8.104468  val-aft-nloglik:8.121284  
[9] train-aft-nloglik:7.316325  val-aft-nloglik:7.333833  
[10]  train-aft-nloglik:6.624332  val-aft-nloglik:6.642500  
[11]  train-aft-nloglik:6.016382  val-aft-nloglik:6.035183  
[12]  train-aft-nloglik:5.481928  val-aft-nloglik:5.501335  
[13]  train-aft-nloglik:5.011772  val-aft-nloglik:5.031762  
[14]  train-aft-nloglik:4.597901  val-aft-nloglik:4.618449  
[15]  train-aft-nloglik:4.233325  val-aft-nloglik:4.254411  
[16]  train-aft-nloglik:3.911947  val-aft-nloglik:3.933590
```

[17]	train-aft-nloglik:3.628446	val-aft-nloglik:3.650619
[18]	train-aft-nloglik:3.378183	val-aft-nloglik:3.400895
[19]	train-aft-nloglik:3.157111	val-aft-nloglik:3.180300
[20]	train-aft-nloglik:2.961688	val-aft-nloglik:2.985329
[21]	train-aft-nloglik:2.788832	val-aft-nloglik:2.812928
[22]	train-aft-nloglik:2.635831	val-aft-nloglik:2.660305
[23]	train-aft-nloglik:2.500327	val-aft-nloglik:2.525173
[24]	train-aft-nloglik:2.380239	val-aft-nloglik:2.405482
[25]	train-aft-nloglik:2.273724	val-aft-nloglik:2.299463
[26]	train-aft-nloglik:2.179243	val-aft-nloglik:2.205300
[27]	train-aft-nloglik:2.095348	val-aft-nloglik:2.121883
[28]	train-aft-nloglik:2.020852	val-aft-nloglik:2.047743
[29]	train-aft-nloglik:1.954619	val-aft-nloglik:1.981961
[30]	train-aft-nloglik:1.895796	val-aft-nloglik:1.923622
[31]	train-aft-nloglik:1.843413	val-aft-nloglik:1.871682
[32]	train-aft-nloglik:1.796780	val-aft-nloglik:1.825535
[33]	train-aft-nloglik:1.755316	val-aft-nloglik:1.784474
[34]	train-aft-nloglik:1.718370	val-aft-nloglik:1.747980
[35]	train-aft-nloglik:1.685421	val-aft-nloglik:1.715607
[36]	train-aft-nloglik:1.656152	val-aft-nloglik:1.686879
[37]	train-aft-nloglik:1.629914	val-aft-nloglik:1.661317
[38]	train-aft-nloglik:1.606442	val-aft-nloglik:1.638562
[39]	train-aft-nloglik:1.585665	val-aft-nloglik:1.618350
[40]	train-aft-nloglik:1.567183	val-aft-nloglik:1.600400
[41]	train-aft-nloglik:1.550570	val-aft-nloglik:1.584496
[42]	train-aft-nloglik:1.535856	val-aft-nloglik:1.570263
[43]	train-aft-nloglik:1.522640	val-aft-nloglik:1.557635
[44]	train-aft-nloglik:1.510762	val-aft-nloglik:1.546419
[45]	train-aft-nloglik:1.500359	val-aft-nloglik:1.536519
[46]	train-aft-nloglik:1.491006	val-aft-nloglik:1.527771
[47]	train-aft-nloglik:1.482731	val-aft-nloglik:1.520075
[48]	train-aft-nloglik:1.475246	val-aft-nloglik:1.513251
[49]	train-aft-nloglik:1.468573	val-aft-nloglik:1.507236
[50]	train-aft-nloglik:1.462707	val-aft-nloglik:1.501866
[51]	train-aft-nloglik:1.457337	val-aft-nloglik:1.497174
[52]	train-aft-nloglik:1.452422	val-aft-nloglik:1.492922
[53]	train-aft-nloglik:1.448012	val-aft-nloglik:1.489267
[54]	train-aft-nloglik:1.444251	val-aft-nloglik:1.485976
[55]	train-aft-nloglik:1.440740	val-aft-nloglik:1.483181
[56]	train-aft-nloglik:1.437486	val-aft-nloglik:1.480523
[57]	train-aft-nloglik:1.434607	val-aft-nloglik:1.478319
[58]	train-aft-nloglik:1.432182	val-aft-nloglik:1.476330
[59]	train-aft-nloglik:1.429686	val-aft-nloglik:1.474493

[60]	train-aft-nloglik:1.427706	val-aft-nloglik:1.473013
[61]	train-aft-nloglik:1.425826	val-aft-nloglik:1.471582
[62]	train-aft-nloglik:1.424287	val-aft-nloglik:1.470419
[63]	train-aft-nloglik:1.422807	val-aft-nloglik:1.469344
[64]	train-aft-nloglik:1.421498	val-aft-nloglik:1.468415
[65]	train-aft-nloglik:1.420218	val-aft-nloglik:1.467604
[66]	train-aft-nloglik:1.418959	val-aft-nloglik:1.466958
[67]	train-aft-nloglik:1.418015	val-aft-nloglik:1.466389
[68]	train-aft-nloglik:1.417169	val-aft-nloglik:1.465921
[69]	train-aft-nloglik:1.416316	val-aft-nloglik:1.465506
[70]	train-aft-nloglik:1.415578	val-aft-nloglik:1.465197
[71]	train-aft-nloglik:1.414927	val-aft-nloglik:1.464961
[72]	train-aft-nloglik:1.413828	val-aft-nloglik:1.464631
[73]	train-aft-nloglik:1.412967	val-aft-nloglik:1.464448
[74]	train-aft-nloglik:1.412046	val-aft-nloglik:1.464125
[75]	train-aft-nloglik:1.411272	val-aft-nloglik:1.464043
[76]	train-aft-nloglik:1.410832	val-aft-nloglik:1.463930
[77]	train-aft-nloglik:1.410372	val-aft-nloglik:1.463827
[78]	train-aft-nloglik:1.409732	val-aft-nloglik:1.463607
[79]	train-aft-nloglik:1.409311	val-aft-nloglik:1.463505
[80]	train-aft-nloglik:1.408803	val-aft-nloglik:1.463366
[81]	train-aft-nloglik:1.408155	val-aft-nloglik:1.463365
[82]	train-aft-nloglik:1.407601	val-aft-nloglik:1.463318
[83]	train-aft-nloglik:1.407210	val-aft-nloglik:1.463267
[84]	train-aft-nloglik:1.406821	val-aft-nloglik:1.463211
[85]	train-aft-nloglik:1.406515	val-aft-nloglik:1.463163
[86]	train-aft-nloglik:1.406057	val-aft-nloglik:1.463180
[87]	train-aft-nloglik:1.405862	val-aft-nloglik:1.463137
[88]	train-aft-nloglik:1.405605	val-aft-nloglik:1.463179
[89]	train-aft-nloglik:1.405389	val-aft-nloglik:1.463238
[90]	train-aft-nloglik:1.405144	val-aft-nloglik:1.463321
[91]	train-aft-nloglik:1.404976	val-aft-nloglik:1.463337
[92]	train-aft-nloglik:1.404598	val-aft-nloglik:1.463273
[93]	train-aft-nloglik:1.404446	val-aft-nloglik:1.463237
[94]	train-aft-nloglik:1.404229	val-aft-nloglik:1.463173
[95]	train-aft-nloglik:1.404071	val-aft-nloglik:1.463162
[96]	train-aft-nloglik:1.403936	val-aft-nloglik:1.463282
[97]	train-aft-nloglik:1.403797	val-aft-nloglik:1.463303
Stopping. Best iteration:		
[87]	train-aft-nloglik:1.405862	val-aft-nloglik:1.463137

SHAP?