

Sensitivity Analysis: XGBoost

Mingcheng Hu

Table of contents

Load Data	2
XGBoost	6
Data Preparation	6
Hyperparameter Tuning	7
Variable Selection	12
Cross Validation to Select the Best Number of Features	15
Model Fitting	18

```
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/youukias/BIOS-Material/BIOS992/utils/csv_utils.r")
# * Don't use setwd() for Quarto documents!
# setwd("/work/users/y/u/youukias/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, "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: full"
```

```
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    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"	

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

```
# * 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",
    "0 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))
```

```

# * 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

```

# 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_valid <- list()  
error_indices <- c()  
for (i in seq_along(tuning_results)) {  
  if (!inherits(tuning_results[[i]], "try-error")) {  
    tuning_results_valid[[length(tuning_results_valid) + 1]] <-  
    ↪ tuning_results[[i]]  
  } else {  
    error_indices <- c(error_indices, i)  
    cat("Error in trial", i, ":", as.character(tuning_results[[i]]),  
    ↪ "\n")  
  }  
}
```

```
Error in trial 1 : Error in concordance.index(pred, val_data$time, val_data$event) :  
  NA/NaN/Inf in foreign function call (arg 3)
```

```
Error in trial 3 : Error in concordance.index(pred, val_data$time, val_data$event) :  
  NA/NaN/Inf in foreign function call (arg 3)
```

```
Error in trial 5 : Error in concordance.index(pred, val_data$time, val_data$event) :  
  NA/NaN/Inf in foreign function call (arg 3)
```

```
Error in trial 7 : Error in concordance.index(pred, val_data$time, val_data$event) :  
  NA/NaN/Inf in foreign function call (arg 3)
```

```
Error in trial 9 : Error in concordance.index(pred, val_data$time, val_data$event) :  
  NA/NaN/Inf in foreign function call (arg 3)
```

```
Error in trial 11 : Error in concordance.index(pred, val_data$time, val_data$event) :  
  NA/NaN/Inf in foreign function call (arg 3)
```

Error in trial 13 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 15 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 17 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 19 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 21 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 23 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 25 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 27 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 29 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 31 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 33 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 35 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 37 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 39 : Error in concordance.index(pred, val_data\$time, val_data\$event) :
NA/NaN/Inf in foreign function call (arg 3)

Error in trial 41 : Error in concordance.index(pred, val_data\$time, val_data\$event) :

```

NA/NaN/Inf in foreign function call (arg 3)

Error in trial 43 : Error in concordance.index(pred, val_data$time, val_data$event) :
  NA/NaN/Inf in foreign function call (arg 3)

Error in trial 45 : Error in concordance.index(pred, val_data$time, val_data$event) :
  NA/NaN/Inf in foreign function call (arg 3)

Error in trial 47 : Error in concordance.index(pred, val_data$time, val_data$event) :
  NA/NaN/Inf in foreign function call (arg 3)

Error in trial 49 : Error in concordance.index(pred, val_data$time, val_data$event) :
  NA/NaN/Inf in foreign function call (arg 3)

```

```

tuning_results <- bind_rows(tuning_results_valid) # 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] 0.09279684
```

```
$max_depth
[1] 3
```

```
$min_child_weight  
[1] 0.001588788
```

```
$reg_alpha  
[1] 7.276133
```

```
$reg_lambda  
[1] 0.01915083
```

```
$aft_loss_distribution_scale  
[1] 0.7105462
```

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:27.620389 val-aft-nloglik:27.627846  
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:27.610732 val-aft-nloglik:27.615595  
[3] train-aft-nloglik:27.589430 val-aft-nloglik:27.593596  
[4] train-aft-nloglik:27.550827 val-aft-nloglik:27.551687  
[5] train-aft-nloglik:27.471049 val-aft-nloglik:27.460399  
[6] train-aft-nloglik:27.415618 val-aft-nloglik:27.404261
```

[7] train-aft-nloglik:26.199515 val-aft-nloglik:26.185999
 [8] train-aft-nloglik:22.142086 val-aft-nloglik:22.126276
 [9] train-aft-nloglik:18.649892 val-aft-nloglik:18.637067
 [10] train-aft-nloglik:15.757417 val-aft-nloglik:15.747177
 [11] train-aft-nloglik:13.365124 val-aft-nloglik:13.358380
 [12] train-aft-nloglik:11.384777 val-aft-nloglik:11.380110
 [13] train-aft-nloglik:9.744235 val-aft-nloglik:9.742342
 [14] train-aft-nloglik:8.383599 val-aft-nloglik:8.383941
 [15] train-aft-nloglik:7.253993 val-aft-nloglik:7.256695
 [16] train-aft-nloglik:6.314925 val-aft-nloglik:6.319984
 [17] train-aft-nloglik:5.533501 val-aft-nloglik:5.541208
 [18] train-aft-nloglik:4.882264 val-aft-nloglik:4.892199
 [19] train-aft-nloglik:4.338861 val-aft-nloglik:4.350265
 [20] train-aft-nloglik:3.884676 val-aft-nloglik:3.897758
 [21] train-aft-nloglik:3.504664 val-aft-nloglik:3.518913
 [22] train-aft-nloglik:3.186104 val-aft-nloglik:3.201635
 [23] train-aft-nloglik:2.918799 val-aft-nloglik:2.935354
 [24] train-aft-nloglik:2.694178 val-aft-nloglik:2.711730
 [25] train-aft-nloglik:2.505226 val-aft-nloglik:2.523904
 [26] train-aft-nloglik:2.345949 val-aft-nloglik:2.365780
 [27] train-aft-nloglik:2.211463 val-aft-nloglik:2.232224
 [28] train-aft-nloglik:2.097854 val-aft-nloglik:2.119747
 [29] train-aft-nloglik:2.001855 val-aft-nloglik:2.024661
 [30] train-aft-nloglik:1.920594 val-aft-nloglik:1.944106
 [31] train-aft-nloglik:1.851791 val-aft-nloglik:1.876144
 [32] train-aft-nloglik:1.793440 val-aft-nloglik:1.818100
 [33] train-aft-nloglik:1.743978 val-aft-nloglik:1.769169
 [34] train-aft-nloglik:1.701996 val-aft-nloglik:1.727793
 [35] train-aft-nloglik:1.666296 val-aft-nloglik:1.692686
 [36] train-aft-nloglik:1.635910 val-aft-nloglik:1.662731
 [37] train-aft-nloglik:1.609998 val-aft-nloglik:1.637623
 [38] train-aft-nloglik:1.588025 val-aft-nloglik:1.616217
 [39] train-aft-nloglik:1.569287 val-aft-nloglik:1.597968
 [40] train-aft-nloglik:1.553292 val-aft-nloglik:1.582583
 [41] train-aft-nloglik:1.539673 val-aft-nloglik:1.569453
 [42] train-aft-nloglik:1.528060 val-aft-nloglik:1.558320
 [43] train-aft-nloglik:1.518219 val-aft-nloglik:1.548973
 [44] train-aft-nloglik:1.509741 val-aft-nloglik:1.541108
 [45] train-aft-nloglik:1.502483 val-aft-nloglik:1.534345
 [46] train-aft-nloglik:1.496188 val-aft-nloglik:1.528659
 [47] train-aft-nloglik:1.490861 val-aft-nloglik:1.523669
 [48] train-aft-nloglik:1.486097 val-aft-nloglik:1.519834
 [49] train-aft-nloglik:1.482151 val-aft-nloglik:1.516262

[50]	train-aft-nloglik:1.478626	val-aft-nloglik:1.513169
[51]	train-aft-nloglik:1.475635	val-aft-nloglik:1.510558
[52]	train-aft-nloglik:1.473013	val-aft-nloglik:1.508334
[53]	train-aft-nloglik:1.470731	val-aft-nloglik:1.506551
[54]	train-aft-nloglik:1.468776	val-aft-nloglik:1.505112
[55]	train-aft-nloglik:1.467027	val-aft-nloglik:1.504025
[56]	train-aft-nloglik:1.465480	val-aft-nloglik:1.502938
[57]	train-aft-nloglik:1.464069	val-aft-nloglik:1.501736
[58]	train-aft-nloglik:1.462854	val-aft-nloglik:1.500752
[59]	train-aft-nloglik:1.461790	val-aft-nloglik:1.499890
[60]	train-aft-nloglik:1.460689	val-aft-nloglik:1.499255
[61]	train-aft-nloglik:1.459806	val-aft-nloglik:1.498939
[62]	train-aft-nloglik:1.459089	val-aft-nloglik:1.498630
[63]	train-aft-nloglik:1.458356	val-aft-nloglik:1.498075
[64]	train-aft-nloglik:1.457626	val-aft-nloglik:1.497691
[65]	train-aft-nloglik:1.456979	val-aft-nloglik:1.497582
[66]	train-aft-nloglik:1.456270	val-aft-nloglik:1.497285
[67]	train-aft-nloglik:1.455702	val-aft-nloglik:1.497229
[68]	train-aft-nloglik:1.455145	val-aft-nloglik:1.496852
[69]	train-aft-nloglik:1.454653	val-aft-nloglik:1.496703
[70]	train-aft-nloglik:1.454084	val-aft-nloglik:1.496594
[71]	train-aft-nloglik:1.453574	val-aft-nloglik:1.496202
[72]	train-aft-nloglik:1.453118	val-aft-nloglik:1.496123
[73]	train-aft-nloglik:1.452658	val-aft-nloglik:1.495993
[74]	train-aft-nloglik:1.452198	val-aft-nloglik:1.495731
[75]	train-aft-nloglik:1.451639	val-aft-nloglik:1.495663
[76]	train-aft-nloglik:1.451274	val-aft-nloglik:1.495437
[77]	train-aft-nloglik:1.450852	val-aft-nloglik:1.495320
[78]	train-aft-nloglik:1.450374	val-aft-nloglik:1.495161
[79]	train-aft-nloglik:1.449970	val-aft-nloglik:1.495053
[80]	train-aft-nloglik:1.449634	val-aft-nloglik:1.494864
[81]	train-aft-nloglik:1.449229	val-aft-nloglik:1.494804
[82]	train-aft-nloglik:1.448948	val-aft-nloglik:1.494650
[83]	train-aft-nloglik:1.448561	val-aft-nloglik:1.494457
[84]	train-aft-nloglik:1.448180	val-aft-nloglik:1.494513
[85]	train-aft-nloglik:1.447790	val-aft-nloglik:1.494443
[86]	train-aft-nloglik:1.447423	val-aft-nloglik:1.494624
[87]	train-aft-nloglik:1.447034	val-aft-nloglik:1.494644
[88]	train-aft-nloglik:1.446747	val-aft-nloglik:1.494644
[89]	train-aft-nloglik:1.446378	val-aft-nloglik:1.494599
[90]	train-aft-nloglik:1.446009	val-aft-nloglik:1.494658
[91]	train-aft-nloglik:1.445654	val-aft-nloglik:1.494686
[92]	train-aft-nloglik:1.445394	val-aft-nloglik:1.494461

```

[93]   train-aft-nloglik:1.445067   val-aft-nloglik:1.494441
[94]   train-aft-nloglik:1.444852   val-aft-nloglik:1.494433
[95]   train-aft-nloglik:1.444495   val-aft-nloglik:1.494338
[96]   train-aft-nloglik:1.444229   val-aft-nloglik:1.494249
[97]   train-aft-nloglik:1.443878   val-aft-nloglik:1.494307
[98]   train-aft-nloglik:1.443471   val-aft-nloglik:1.494244
[99]   train-aft-nloglik:1.443077   val-aft-nloglik:1.494197
[100]  train-aft-nloglik:1.442721   val-aft-nloglik:1.493949
[101]  train-aft-nloglik:1.442401   val-aft-nloglik:1.493959
[102]  train-aft-nloglik:1.442130   val-aft-nloglik:1.494078
[103]  train-aft-nloglik:1.441871   val-aft-nloglik:1.493941
[104]  train-aft-nloglik:1.441585   val-aft-nloglik:1.494056
[105]  train-aft-nloglik:1.441252   val-aft-nloglik:1.494220
[106]  train-aft-nloglik:1.440946   val-aft-nloglik:1.494269
[107]  train-aft-nloglik:1.440468   val-aft-nloglik:1.494256
[108]  train-aft-nloglik:1.440100   val-aft-nloglik:1.494389
[109]  train-aft-nloglik:1.439790   val-aft-nloglik:1.494358
[110]  train-aft-nloglik:1.439579   val-aft-nloglik:1.494414
[111]  train-aft-nloglik:1.439412   val-aft-nloglik:1.494562
[112]  train-aft-nloglik:1.439111   val-aft-nloglik:1.494689
[113]  train-aft-nloglik:1.438754   val-aft-nloglik:1.494503
Stopping. Best iteration:
[103]   train-aft-nloglik:1.441871   val-aft-nloglik:1.493941

```

```

# 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 = 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

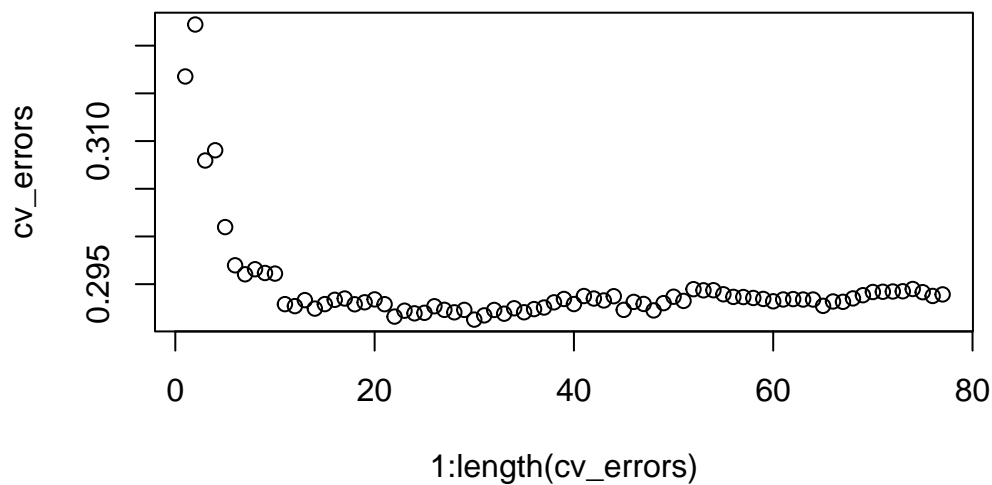
# Use C-index to measure the performance of the model
1 - concordance.index(pred, val_y_fold$time,
  ↪ val_y_fold$event)$c.index
})
print(mean(fold_errors))
mean(fold_errors)
}, title = "Cross Validation to Select the Best Number of Features")

```

```

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 30"
```

Model Fitting

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

```

[1] train-aft-nloglik:27.620389 val-aft-nloglik:27.627846
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:27.610732 val-aft-nloglik:27.615595
[3] train-aft-nloglik:27.589430 val-aft-nloglik:27.593596
[4] train-aft-nloglik:27.550827 val-aft-nloglik:27.551687
[5] train-aft-nloglik:27.471049 val-aft-nloglik:27.460399
[6] train-aft-nloglik:27.415618 val-aft-nloglik:27.404261
[7] train-aft-nloglik:26.199515 val-aft-nloglik:26.185999
[8] train-aft-nloglik:22.142086 val-aft-nloglik:22.126276
[9] train-aft-nloglik:18.649892 val-aft-nloglik:18.637067
[10] train-aft-nloglik:15.757417 val-aft-nloglik:15.747177
[11] train-aft-nloglik:13.365135 val-aft-nloglik:13.358051
[12] train-aft-nloglik:11.384795 val-aft-nloglik:11.380771
[13] train-aft-nloglik:9.744252 val-aft-nloglik:9.742650
[14] train-aft-nloglik:8.383610 val-aft-nloglik:8.384244
[15] train-aft-nloglik:7.254011 val-aft-nloglik:7.256866
[16] train-aft-nloglik:6.314953 val-aft-nloglik:6.319978
[17] train-aft-nloglik:5.533564 val-aft-nloglik:5.541123
[18] train-aft-nloglik:4.882353 val-aft-nloglik:4.891571
[19] train-aft-nloglik:4.338895 val-aft-nloglik:4.349788
[20] train-aft-nloglik:3.884776 val-aft-nloglik:3.897329
[21] train-aft-nloglik:3.504716 val-aft-nloglik:3.518838
[22] train-aft-nloglik:3.186202 val-aft-nloglik:3.201501
[23] train-aft-nloglik:2.919061 val-aft-nloglik:2.935822
[24] train-aft-nloglik:2.694427 val-aft-nloglik:2.712138
[25] train-aft-nloglik:2.505386 val-aft-nloglik:2.524211
[26] train-aft-nloglik:2.346109 val-aft-nloglik:2.365688
[27] train-aft-nloglik:2.211651 val-aft-nloglik:2.232407
[28] train-aft-nloglik:2.098145 val-aft-nloglik:2.119814
[29] train-aft-nloglik:2.002151 val-aft-nloglik:2.024684
[30] train-aft-nloglik:1.920893 val-aft-nloglik:1.944410
[31] train-aft-nloglik:1.852114 val-aft-nloglik:1.876199
[32] train-aft-nloglik:1.793695 val-aft-nloglik:1.818162
[33] train-aft-nloglik:1.744214 val-aft-nloglik:1.769315

```

[34]	train-aft-nloglik:1.702241	val-aft-nloglik:1.727859
[35]	train-aft-nloglik:1.666491	val-aft-nloglik:1.692794
[36]	train-aft-nloglik:1.636080	val-aft-nloglik:1.662796
[37]	train-aft-nloglik:1.610287	val-aft-nloglik:1.637557
[38]	train-aft-nloglik:1.588313	val-aft-nloglik:1.616265
[39]	train-aft-nloglik:1.569571	val-aft-nloglik:1.598125
[40]	train-aft-nloglik:1.553653	val-aft-nloglik:1.582660
[41]	train-aft-nloglik:1.540080	val-aft-nloglik:1.569779
[42]	train-aft-nloglik:1.528416	val-aft-nloglik:1.558675
[43]	train-aft-nloglik:1.518537	val-aft-nloglik:1.549281
[44]	train-aft-nloglik:1.509919	val-aft-nloglik:1.541480
[45]	train-aft-nloglik:1.502595	val-aft-nloglik:1.534756
[46]	train-aft-nloglik:1.496389	val-aft-nloglik:1.528986
[47]	train-aft-nloglik:1.491061	val-aft-nloglik:1.524044
[48]	train-aft-nloglik:1.486356	val-aft-nloglik:1.519843
[49]	train-aft-nloglik:1.482394	val-aft-nloglik:1.516084
[50]	train-aft-nloglik:1.478895	val-aft-nloglik:1.512878
[51]	train-aft-nloglik:1.475901	val-aft-nloglik:1.510303
[52]	train-aft-nloglik:1.473219	val-aft-nloglik:1.508376
[53]	train-aft-nloglik:1.470910	val-aft-nloglik:1.506707
[54]	train-aft-nloglik:1.468915	val-aft-nloglik:1.505018
[55]	train-aft-nloglik:1.467218	val-aft-nloglik:1.503540
[56]	train-aft-nloglik:1.465655	val-aft-nloglik:1.502676
[57]	train-aft-nloglik:1.464290	val-aft-nloglik:1.501759
[58]	train-aft-nloglik:1.463052	val-aft-nloglik:1.500836
[59]	train-aft-nloglik:1.461980	val-aft-nloglik:1.500324
[60]	train-aft-nloglik:1.461039	val-aft-nloglik:1.499712
[61]	train-aft-nloglik:1.460165	val-aft-nloglik:1.499284
[62]	train-aft-nloglik:1.459347	val-aft-nloglik:1.498820
[63]	train-aft-nloglik:1.458560	val-aft-nloglik:1.498710
[64]	train-aft-nloglik:1.457906	val-aft-nloglik:1.498505
[65]	train-aft-nloglik:1.457298	val-aft-nloglik:1.497936
[66]	train-aft-nloglik:1.456663	val-aft-nloglik:1.497659
[67]	train-aft-nloglik:1.456138	val-aft-nloglik:1.497577
[68]	train-aft-nloglik:1.455553	val-aft-nloglik:1.497570
[69]	train-aft-nloglik:1.455001	val-aft-nloglik:1.497393
[70]	train-aft-nloglik:1.454567	val-aft-nloglik:1.497302
[71]	train-aft-nloglik:1.454103	val-aft-nloglik:1.497212
[72]	train-aft-nloglik:1.453658	val-aft-nloglik:1.497014
[73]	train-aft-nloglik:1.453131	val-aft-nloglik:1.496940
[74]	train-aft-nloglik:1.452591	val-aft-nloglik:1.496947
[75]	train-aft-nloglik:1.452160	val-aft-nloglik:1.496911
[76]	train-aft-nloglik:1.451752	val-aft-nloglik:1.496798

[77]	train-aft-nloglik:1.451400	val-aft-nloglik:1.496609
[78]	train-aft-nloglik:1.451030	val-aft-nloglik:1.496393
[79]	train-aft-nloglik:1.450686	val-aft-nloglik:1.496517
[80]	train-aft-nloglik:1.450374	val-aft-nloglik:1.496342
[81]	train-aft-nloglik:1.450101	val-aft-nloglik:1.496208
[82]	train-aft-nloglik:1.449666	val-aft-nloglik:1.496267
[83]	train-aft-nloglik:1.449304	val-aft-nloglik:1.496330
[84]	train-aft-nloglik:1.449016	val-aft-nloglik:1.496145
[85]	train-aft-nloglik:1.448649	val-aft-nloglik:1.496020
[86]	train-aft-nloglik:1.448378	val-aft-nloglik:1.495873
[87]	train-aft-nloglik:1.448099	val-aft-nloglik:1.495796
[88]	train-aft-nloglik:1.447739	val-aft-nloglik:1.495808
[89]	train-aft-nloglik:1.447479	val-aft-nloglik:1.495746
[90]	train-aft-nloglik:1.447146	val-aft-nloglik:1.495821
[91]	train-aft-nloglik:1.446880	val-aft-nloglik:1.495720
[92]	train-aft-nloglik:1.446642	val-aft-nloglik:1.495625
[93]	train-aft-nloglik:1.446344	val-aft-nloglik:1.495669
[94]	train-aft-nloglik:1.446104	val-aft-nloglik:1.495481
[95]	train-aft-nloglik:1.445862	val-aft-nloglik:1.495407
[96]	train-aft-nloglik:1.445521	val-aft-nloglik:1.495496
[97]	train-aft-nloglik:1.445174	val-aft-nloglik:1.495425
[98]	train-aft-nloglik:1.444901	val-aft-nloglik:1.495425
[99]	train-aft-nloglik:1.444655	val-aft-nloglik:1.495318
[100]	train-aft-nloglik:1.444375	val-aft-nloglik:1.495235
[101]	train-aft-nloglik:1.444082	val-aft-nloglik:1.495244
[102]	train-aft-nloglik:1.443751	val-aft-nloglik:1.495298
[103]	train-aft-nloglik:1.443369	val-aft-nloglik:1.495120
[104]	train-aft-nloglik:1.443031	val-aft-nloglik:1.495065
[105]	train-aft-nloglik:1.442772	val-aft-nloglik:1.495180
[106]	train-aft-nloglik:1.442534	val-aft-nloglik:1.495379
[107]	train-aft-nloglik:1.442292	val-aft-nloglik:1.495348
[108]	train-aft-nloglik:1.442079	val-aft-nloglik:1.495300
[109]	train-aft-nloglik:1.441932	val-aft-nloglik:1.495281
[110]	train-aft-nloglik:1.441738	val-aft-nloglik:1.495201
[111]	train-aft-nloglik:1.441535	val-aft-nloglik:1.495227
[112]	train-aft-nloglik:1.441262	val-aft-nloglik:1.495141
[113]	train-aft-nloglik:1.440977	val-aft-nloglik:1.495009
[114]	train-aft-nloglik:1.440689	val-aft-nloglik:1.495038
[115]	train-aft-nloglik:1.440413	val-aft-nloglik:1.495134
[116]	train-aft-nloglik:1.440126	val-aft-nloglik:1.495213
[117]	train-aft-nloglik:1.439903	val-aft-nloglik:1.495186
[118]	train-aft-nloglik:1.439659	val-aft-nloglik:1.495060
[119]	train-aft-nloglik:1.439441	val-aft-nloglik:1.495089

```

[120] train-aft-nloglik:1.439314 val-aft-nloglik:1.495165
[121] train-aft-nloglik:1.439092 val-aft-nloglik:1.495187
[122] train-aft-nloglik:1.438803 val-aft-nloglik:1.495163
[123] train-aft-nloglik:1.438549 val-aft-nloglik:1.495086
Stopping. Best iteration:
[113] train-aft-nloglik:1.440977 val-aft-nloglik:1.495009

```

```

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

```

```

[1] train-aft-nloglik:27.620389 val-aft-nloglik:27.627846
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:27.610732 val-aft-nloglik:27.615595
[3] train-aft-nloglik:27.589430 val-aft-nloglik:27.593596
[4] train-aft-nloglik:27.550827 val-aft-nloglik:27.551687
[5] train-aft-nloglik:27.471049 val-aft-nloglik:27.460399
[6] train-aft-nloglik:27.415618 val-aft-nloglik:27.404261

```

[7] train-aft-nloglik:26.199515 val-aft-nloglik:26.185999
 [8] train-aft-nloglik:22.142086 val-aft-nloglik:22.126276
 [9] train-aft-nloglik:18.649892 val-aft-nloglik:18.637067
 [10] train-aft-nloglik:15.757417 val-aft-nloglik:15.747177
 [11] train-aft-nloglik:13.365124 val-aft-nloglik:13.358380
 [12] train-aft-nloglik:11.384777 val-aft-nloglik:11.380110
 [13] train-aft-nloglik:9.744235 val-aft-nloglik:9.742342
 [14] train-aft-nloglik:8.383599 val-aft-nloglik:8.383941
 [15] train-aft-nloglik:7.253993 val-aft-nloglik:7.256695
 [16] train-aft-nloglik:6.314925 val-aft-nloglik:6.319984
 [17] train-aft-nloglik:5.533501 val-aft-nloglik:5.541208
 [18] train-aft-nloglik:4.882264 val-aft-nloglik:4.892199
 [19] train-aft-nloglik:4.338861 val-aft-nloglik:4.350265
 [20] train-aft-nloglik:3.884676 val-aft-nloglik:3.897758
 [21] train-aft-nloglik:3.504664 val-aft-nloglik:3.518913
 [22] train-aft-nloglik:3.186104 val-aft-nloglik:3.201635
 [23] train-aft-nloglik:2.918799 val-aft-nloglik:2.935354
 [24] train-aft-nloglik:2.694178 val-aft-nloglik:2.711730
 [25] train-aft-nloglik:2.505226 val-aft-nloglik:2.523904
 [26] train-aft-nloglik:2.345949 val-aft-nloglik:2.365780
 [27] train-aft-nloglik:2.211463 val-aft-nloglik:2.232224
 [28] train-aft-nloglik:2.097854 val-aft-nloglik:2.119747
 [29] train-aft-nloglik:2.001855 val-aft-nloglik:2.024661
 [30] train-aft-nloglik:1.920594 val-aft-nloglik:1.944106
 [31] train-aft-nloglik:1.851791 val-aft-nloglik:1.876144
 [32] train-aft-nloglik:1.793440 val-aft-nloglik:1.818100
 [33] train-aft-nloglik:1.743978 val-aft-nloglik:1.769169
 [34] train-aft-nloglik:1.701996 val-aft-nloglik:1.727793
 [35] train-aft-nloglik:1.666296 val-aft-nloglik:1.692686
 [36] train-aft-nloglik:1.635910 val-aft-nloglik:1.662731
 [37] train-aft-nloglik:1.609998 val-aft-nloglik:1.637623
 [38] train-aft-nloglik:1.588025 val-aft-nloglik:1.616217
 [39] train-aft-nloglik:1.569287 val-aft-nloglik:1.597968
 [40] train-aft-nloglik:1.553292 val-aft-nloglik:1.582583
 [41] train-aft-nloglik:1.539673 val-aft-nloglik:1.569453
 [42] train-aft-nloglik:1.528060 val-aft-nloglik:1.558320
 [43] train-aft-nloglik:1.518219 val-aft-nloglik:1.548973
 [44] train-aft-nloglik:1.509741 val-aft-nloglik:1.541108
 [45] train-aft-nloglik:1.502483 val-aft-nloglik:1.534345
 [46] train-aft-nloglik:1.496188 val-aft-nloglik:1.528659
 [47] train-aft-nloglik:1.490861 val-aft-nloglik:1.523669
 [48] train-aft-nloglik:1.486097 val-aft-nloglik:1.519834
 [49] train-aft-nloglik:1.482151 val-aft-nloglik:1.516262

[50]	train-aft-nloglik:1.478626	val-aft-nloglik:1.513169
[51]	train-aft-nloglik:1.475635	val-aft-nloglik:1.510558
[52]	train-aft-nloglik:1.473013	val-aft-nloglik:1.508334
[53]	train-aft-nloglik:1.470731	val-aft-nloglik:1.506551
[54]	train-aft-nloglik:1.468776	val-aft-nloglik:1.505112
[55]	train-aft-nloglik:1.467027	val-aft-nloglik:1.504025
[56]	train-aft-nloglik:1.465480	val-aft-nloglik:1.502938
[57]	train-aft-nloglik:1.464069	val-aft-nloglik:1.501736
[58]	train-aft-nloglik:1.462854	val-aft-nloglik:1.500752
[59]	train-aft-nloglik:1.461790	val-aft-nloglik:1.499890
[60]	train-aft-nloglik:1.460689	val-aft-nloglik:1.499255
[61]	train-aft-nloglik:1.459806	val-aft-nloglik:1.498939
[62]	train-aft-nloglik:1.459089	val-aft-nloglik:1.498630
[63]	train-aft-nloglik:1.458356	val-aft-nloglik:1.498075
[64]	train-aft-nloglik:1.457626	val-aft-nloglik:1.497691
[65]	train-aft-nloglik:1.456979	val-aft-nloglik:1.497582
[66]	train-aft-nloglik:1.456270	val-aft-nloglik:1.497285
[67]	train-aft-nloglik:1.455702	val-aft-nloglik:1.497229
[68]	train-aft-nloglik:1.455145	val-aft-nloglik:1.496852
[69]	train-aft-nloglik:1.454653	val-aft-nloglik:1.496703
[70]	train-aft-nloglik:1.454084	val-aft-nloglik:1.496594
[71]	train-aft-nloglik:1.453574	val-aft-nloglik:1.496202
[72]	train-aft-nloglik:1.453118	val-aft-nloglik:1.496123
[73]	train-aft-nloglik:1.452658	val-aft-nloglik:1.495993
[74]	train-aft-nloglik:1.452198	val-aft-nloglik:1.495731
[75]	train-aft-nloglik:1.451639	val-aft-nloglik:1.495663
[76]	train-aft-nloglik:1.451274	val-aft-nloglik:1.495437
[77]	train-aft-nloglik:1.450852	val-aft-nloglik:1.495320
[78]	train-aft-nloglik:1.450374	val-aft-nloglik:1.495161
[79]	train-aft-nloglik:1.449970	val-aft-nloglik:1.495053
[80]	train-aft-nloglik:1.449634	val-aft-nloglik:1.494864
[81]	train-aft-nloglik:1.449229	val-aft-nloglik:1.494804
[82]	train-aft-nloglik:1.448948	val-aft-nloglik:1.494650
[83]	train-aft-nloglik:1.448561	val-aft-nloglik:1.494457
[84]	train-aft-nloglik:1.448180	val-aft-nloglik:1.494513
[85]	train-aft-nloglik:1.447790	val-aft-nloglik:1.494443
[86]	train-aft-nloglik:1.447423	val-aft-nloglik:1.494624
[87]	train-aft-nloglik:1.447034	val-aft-nloglik:1.494644
[88]	train-aft-nloglik:1.446747	val-aft-nloglik:1.494644
[89]	train-aft-nloglik:1.446378	val-aft-nloglik:1.494599
[90]	train-aft-nloglik:1.446009	val-aft-nloglik:1.494658
[91]	train-aft-nloglik:1.445654	val-aft-nloglik:1.494686
[92]	train-aft-nloglik:1.445394	val-aft-nloglik:1.494461

[93]	train-aft-nloglik:1.445067	val-aft-nloglik:1.494441
[94]	train-aft-nloglik:1.444852	val-aft-nloglik:1.494433
[95]	train-aft-nloglik:1.444495	val-aft-nloglik:1.494338
[96]	train-aft-nloglik:1.444229	val-aft-nloglik:1.494249
[97]	train-aft-nloglik:1.443878	val-aft-nloglik:1.494307
[98]	train-aft-nloglik:1.443471	val-aft-nloglik:1.494244
[99]	train-aft-nloglik:1.443077	val-aft-nloglik:1.494197
[100]	train-aft-nloglik:1.442721	val-aft-nloglik:1.493949
[101]	train-aft-nloglik:1.442401	val-aft-nloglik:1.493959
[102]	train-aft-nloglik:1.442130	val-aft-nloglik:1.494078
[103]	train-aft-nloglik:1.441871	val-aft-nloglik:1.493941
[104]	train-aft-nloglik:1.441585	val-aft-nloglik:1.494056
[105]	train-aft-nloglik:1.441252	val-aft-nloglik:1.494220
[106]	train-aft-nloglik:1.440946	val-aft-nloglik:1.494269
[107]	train-aft-nloglik:1.440468	val-aft-nloglik:1.494256
[108]	train-aft-nloglik:1.440100	val-aft-nloglik:1.494389
[109]	train-aft-nloglik:1.439790	val-aft-nloglik:1.494358
[110]	train-aft-nloglik:1.439579	val-aft-nloglik:1.494414
[111]	train-aft-nloglik:1.439412	val-aft-nloglik:1.494562
[112]	train-aft-nloglik:1.439111	val-aft-nloglik:1.494689
[113]	train-aft-nloglik:1.438754	val-aft-nloglik:1.494503
Stopping. Best iteration:		
[103]	train-aft-nloglik:1.441871	val-aft-nloglik:1.493941