

Build Survival Model: XGBoost

Mingcheng Hu

Table of contents

Load Data	2
XGBoost	5
Data Preparation	5
Hyperparameter Tuning	6
Variable Selection	10
Cross Validation to Select the Best Number of Features	13
Model Fitting	15

```
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

```
# 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] 0.1047715
```

```
$max_depth
```

```
[1] 8
```

```
$min_child_weight
```

```
[1] 0.540415
```

```
$reg_alpha
```

```
[1] 48.51929
```

```
$reg_lambda
```

```
[1] 0.7476557
```

```
$aft_loss_distribution_scale
```

```
[1] 1.735978
```

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:14.061364 val-aft-nloglik:14.074756
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:11.701796 val-aft-nloglik:11.716477
[3] train-aft-nloglik:9.796189  val-aft-nloglik:9.812065
[4] train-aft-nloglik:8.255162  val-aft-nloglik:8.272145
[5] train-aft-nloglik:7.007131  val-aft-nloglik:7.025147
[6] train-aft-nloglik:5.994775  val-aft-nloglik:6.013753
[7] train-aft-nloglik:5.172181  val-aft-nloglik:5.192060
[8] train-aft-nloglik:4.502575  val-aft-nloglik:4.523299
[9] train-aft-nloglik:3.956487  val-aft-nloglik:3.978006
[10]  train-aft-nloglik:3.510292  val-aft-nloglik:3.532559
[11]  train-aft-nloglik:3.145027  val-aft-nloglik:3.167986
[12]  train-aft-nloglik:2.845464  val-aft-nloglik:2.869125
[13]  train-aft-nloglik:2.599352  val-aft-nloglik:2.623650
[14]  train-aft-nloglik:2.396823  val-aft-nloglik:2.421746
[15]  train-aft-nloglik:2.229909  val-aft-nloglik:2.255460
[16]  train-aft-nloglik:2.092158  val-aft-nloglik:2.118356
[17]  train-aft-nloglik:1.978319  val-aft-nloglik:2.005010
[18]  train-aft-nloglik:1.884152  val-aft-nloglik:1.911441
[19]  train-aft-nloglik:1.806186  val-aft-nloglik:1.833996
[20]  train-aft-nloglik:1.741616  val-aft-nloglik:1.770017
[21]  train-aft-nloglik:1.688104  val-aft-nloglik:1.717089
[22]  train-aft-nloglik:1.643761  val-aft-nloglik:1.673135
[23]  train-aft-nloglik:1.607020  val-aft-nloglik:1.636912
[24]  train-aft-nloglik:1.576557  val-aft-nloglik:1.606895
[25]  train-aft-nloglik:1.551337  val-aft-nloglik:1.582016
[26]  train-aft-nloglik:1.530438  val-aft-nloglik:1.561517
[27]  train-aft-nloglik:1.513149  val-aft-nloglik:1.544610
[28]  train-aft-nloglik:1.498837  val-aft-nloglik:1.530628
[29]  train-aft-nloglik:1.487012  val-aft-nloglik:1.519184
[30]  train-aft-nloglik:1.477203  val-aft-nloglik:1.509629

```

[31]	train-aft-nloglik:1.469096	val-aft-nloglik:1.501914
[32]	train-aft-nloglik:1.462321	val-aft-nloglik:1.495458
[33]	train-aft-nloglik:1.456733	val-aft-nloglik:1.490101
[34]	train-aft-nloglik:1.452125	val-aft-nloglik:1.485747
[35]	train-aft-nloglik:1.447956	val-aft-nloglik:1.481956
[36]	train-aft-nloglik:1.444627	val-aft-nloglik:1.478845
[37]	train-aft-nloglik:1.441715	val-aft-nloglik:1.476167
[38]	train-aft-nloglik:1.439404	val-aft-nloglik:1.474002
[39]	train-aft-nloglik:1.437350	val-aft-nloglik:1.472202
[40]	train-aft-nloglik:1.435689	val-aft-nloglik:1.470781
[41]	train-aft-nloglik:1.434194	val-aft-nloglik:1.469603
[42]	train-aft-nloglik:1.432850	val-aft-nloglik:1.468681
[43]	train-aft-nloglik:1.431784	val-aft-nloglik:1.467669
[44]	train-aft-nloglik:1.430797	val-aft-nloglik:1.466878
[45]	train-aft-nloglik:1.429995	val-aft-nloglik:1.466127
[46]	train-aft-nloglik:1.429182	val-aft-nloglik:1.465633
[47]	train-aft-nloglik:1.428596	val-aft-nloglik:1.465334
[48]	train-aft-nloglik:1.428039	val-aft-nloglik:1.464928
[49]	train-aft-nloglik:1.427448	val-aft-nloglik:1.464566
[50]	train-aft-nloglik:1.426956	val-aft-nloglik:1.464252
[51]	train-aft-nloglik:1.426389	val-aft-nloglik:1.464007
[52]	train-aft-nloglik:1.425920	val-aft-nloglik:1.463837
[53]	train-aft-nloglik:1.425452	val-aft-nloglik:1.463641
[54]	train-aft-nloglik:1.425086	val-aft-nloglik:1.463459
[55]	train-aft-nloglik:1.424740	val-aft-nloglik:1.463331
[56]	train-aft-nloglik:1.424290	val-aft-nloglik:1.463165
[57]	train-aft-nloglik:1.423940	val-aft-nloglik:1.463018
[58]	train-aft-nloglik:1.423551	val-aft-nloglik:1.462956
[59]	train-aft-nloglik:1.423169	val-aft-nloglik:1.462933
[60]	train-aft-nloglik:1.422933	val-aft-nloglik:1.462917
[61]	train-aft-nloglik:1.422637	val-aft-nloglik:1.462919
[62]	train-aft-nloglik:1.422318	val-aft-nloglik:1.462899
[63]	train-aft-nloglik:1.421998	val-aft-nloglik:1.462752
[64]	train-aft-nloglik:1.421589	val-aft-nloglik:1.462650
[65]	train-aft-nloglik:1.421451	val-aft-nloglik:1.462591
[66]	train-aft-nloglik:1.421184	val-aft-nloglik:1.462553
[67]	train-aft-nloglik:1.420875	val-aft-nloglik:1.462545
[68]	train-aft-nloglik:1.420625	val-aft-nloglik:1.462554
[69]	train-aft-nloglik:1.420506	val-aft-nloglik:1.462477
[70]	train-aft-nloglik:1.420161	val-aft-nloglik:1.462472
[71]	train-aft-nloglik:1.419940	val-aft-nloglik:1.462466
[72]	train-aft-nloglik:1.419632	val-aft-nloglik:1.462432
[73]	train-aft-nloglik:1.419394	val-aft-nloglik:1.462385

```

[74]   train-aft-nloglik:1.419201   val-aft-nloglik:1.462435
[75]   train-aft-nloglik:1.419044   val-aft-nloglik:1.462447
[76]   train-aft-nloglik:1.418747   val-aft-nloglik:1.462377
[77]   train-aft-nloglik:1.418488   val-aft-nloglik:1.462404
[78]   train-aft-nloglik:1.418303   val-aft-nloglik:1.462429
[79]   train-aft-nloglik:1.418055   val-aft-nloglik:1.462480
[80]   train-aft-nloglik:1.417834   val-aft-nloglik:1.462468
[81]   train-aft-nloglik:1.417645   val-aft-nloglik:1.462522
[82]   train-aft-nloglik:1.417346   val-aft-nloglik:1.462480
[83]   train-aft-nloglik:1.417212   val-aft-nloglik:1.462479
[84]   train-aft-nloglik:1.416996   val-aft-nloglik:1.462441
[85]   train-aft-nloglik:1.416792   val-aft-nloglik:1.462415
[86]   train-aft-nloglik:1.416629   val-aft-nloglik:1.462452
Stopping. Best iteration:
[76]   train-aft-nloglik:1.418747   val-aft-nloglik:1.462377

```

```

# 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 <- 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
})

```

```

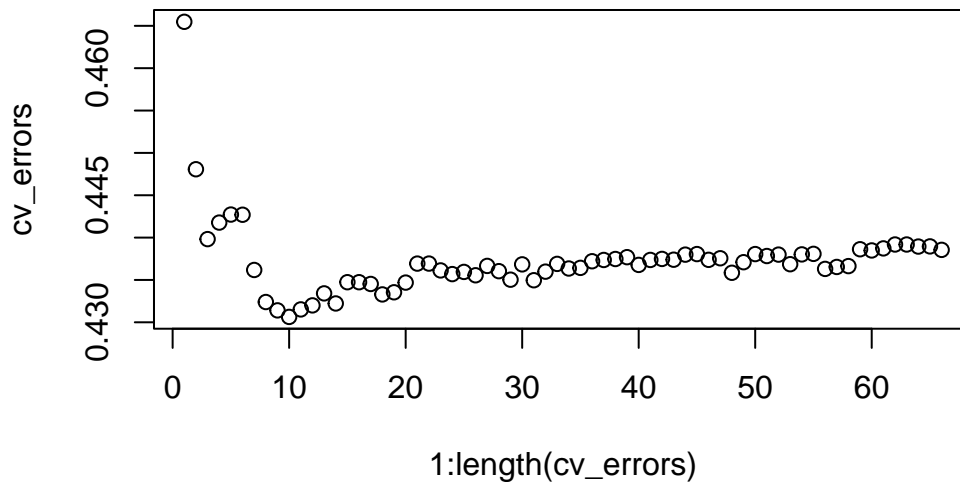
    print(mean(fold_errors))
    mean(fold_errors)
  }, title = "Cross Validation to Select the Best Number of Features")

```

```

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 10"

```

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:14.061364 val-aft-nloglik:14.074756
```

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:11.701796 val-aft-nloglik:11.716477
```

```
[3] train-aft-nloglik:9.796189 val-aft-nloglik:9.812065
```

```
[4] train-aft-nloglik:8.255162 val-aft-nloglik:8.272145
```

```
[5] train-aft-nloglik:7.007131 val-aft-nloglik:7.025147
```

```
[6] train-aft-nloglik:5.994775 val-aft-nloglik:6.013753
```

```
[7] train-aft-nloglik:5.172181 val-aft-nloglik:5.192060
```

```
[8] train-aft-nloglik:4.502575 val-aft-nloglik:4.523299
```

```
[9] train-aft-nloglik:3.956487 val-aft-nloglik:3.978006
```

```
[10] train-aft-nloglik:3.510292 val-aft-nloglik:3.532559
```

```
[11] train-aft-nloglik:3.145027 val-aft-nloglik:3.167986
```

```
[12] train-aft-nloglik:2.845465 val-aft-nloglik:2.869083
```

```
[13] train-aft-nloglik:2.599357 val-aft-nloglik:2.623614
```


[14]	train-aft-nloglik:2.396833	val-aft-nloglik:2.421682
[15]	train-aft-nloglik:2.229926	val-aft-nloglik:2.255353
[16]	train-aft-nloglik:2.092189	val-aft-nloglik:2.118162
[17]	train-aft-nloglik:1.978371	val-aft-nloglik:2.004828
[18]	train-aft-nloglik:1.884229	val-aft-nloglik:1.911196
[19]	train-aft-nloglik:1.806294	val-aft-nloglik:1.833725
[20]	train-aft-nloglik:1.741759	val-aft-nloglik:1.769627
[21]	train-aft-nloglik:1.688297	val-aft-nloglik:1.716533
[22]	train-aft-nloglik:1.644001	val-aft-nloglik:1.672659
[23]	train-aft-nloglik:1.607315	val-aft-nloglik:1.636319
[24]	train-aft-nloglik:1.576907	val-aft-nloglik:1.606333
[25]	train-aft-nloglik:1.551741	val-aft-nloglik:1.581524
[26]	train-aft-nloglik:1.530923	val-aft-nloglik:1.561049
[27]	train-aft-nloglik:1.513707	val-aft-nloglik:1.544183
[28]	train-aft-nloglik:1.499479	val-aft-nloglik:1.530288
[29]	train-aft-nloglik:1.487695	val-aft-nloglik:1.518673
[30]	train-aft-nloglik:1.477976	val-aft-nloglik:1.509287
[31]	train-aft-nloglik:1.469943	val-aft-nloglik:1.501611
[32]	train-aft-nloglik:1.463291	val-aft-nloglik:1.495184
[33]	train-aft-nloglik:1.457711	val-aft-nloglik:1.489885
[34]	train-aft-nloglik:1.452944	val-aft-nloglik:1.485501
[35]	train-aft-nloglik:1.448922	val-aft-nloglik:1.481759
[36]	train-aft-nloglik:1.445608	val-aft-nloglik:1.478721
[37]	train-aft-nloglik:1.442857	val-aft-nloglik:1.476279
[38]	train-aft-nloglik:1.440535	val-aft-nloglik:1.474216
[39]	train-aft-nloglik:1.438636	val-aft-nloglik:1.472445
[40]	train-aft-nloglik:1.436973	val-aft-nloglik:1.470988
[41]	train-aft-nloglik:1.435507	val-aft-nloglik:1.469881
[42]	train-aft-nloglik:1.434315	val-aft-nloglik:1.468793
[43]	train-aft-nloglik:1.433308	val-aft-nloglik:1.467879
[44]	train-aft-nloglik:1.432403	val-aft-nloglik:1.467286
[45]	train-aft-nloglik:1.431652	val-aft-nloglik:1.466601
[46]	train-aft-nloglik:1.430961	val-aft-nloglik:1.466039
[47]	train-aft-nloglik:1.430459	val-aft-nloglik:1.465717
[48]	train-aft-nloglik:1.429968	val-aft-nloglik:1.465438
[49]	train-aft-nloglik:1.429526	val-aft-nloglik:1.465055
[50]	train-aft-nloglik:1.429112	val-aft-nloglik:1.464873
[51]	train-aft-nloglik:1.428801	val-aft-nloglik:1.464667
[52]	train-aft-nloglik:1.428515	val-aft-nloglik:1.464500
[53]	train-aft-nloglik:1.428210	val-aft-nloglik:1.464282
[54]	train-aft-nloglik:1.427968	val-aft-nloglik:1.464115
[55]	train-aft-nloglik:1.427687	val-aft-nloglik:1.463946
[56]	train-aft-nloglik:1.427454	val-aft-nloglik:1.463843

[57]	train-aft-nloglik:1.427257	val-aft-nloglik:1.463838
[58]	train-aft-nloglik:1.427033	val-aft-nloglik:1.463722
[59]	train-aft-nloglik:1.426854	val-aft-nloglik:1.463643
[60]	train-aft-nloglik:1.426654	val-aft-nloglik:1.463579
[61]	train-aft-nloglik:1.426492	val-aft-nloglik:1.463495
[62]	train-aft-nloglik:1.426298	val-aft-nloglik:1.463457
[63]	train-aft-nloglik:1.426126	val-aft-nloglik:1.463405
[64]	train-aft-nloglik:1.426002	val-aft-nloglik:1.463391
[65]	train-aft-nloglik:1.425862	val-aft-nloglik:1.463345
[66]	train-aft-nloglik:1.425794	val-aft-nloglik:1.463313
[67]	train-aft-nloglik:1.425652	val-aft-nloglik:1.463292
[68]	train-aft-nloglik:1.425513	val-aft-nloglik:1.463258
[69]	train-aft-nloglik:1.425464	val-aft-nloglik:1.463259
[70]	train-aft-nloglik:1.425415	val-aft-nloglik:1.463240
[71]	train-aft-nloglik:1.425283	val-aft-nloglik:1.463219
[72]	train-aft-nloglik:1.425158	val-aft-nloglik:1.463194
[73]	train-aft-nloglik:1.425071	val-aft-nloglik:1.463161
[74]	train-aft-nloglik:1.425014	val-aft-nloglik:1.463183
[75]	train-aft-nloglik:1.424975	val-aft-nloglik:1.463180
[76]	train-aft-nloglik:1.424893	val-aft-nloglik:1.463187
[77]	train-aft-nloglik:1.424845	val-aft-nloglik:1.463196
[78]	train-aft-nloglik:1.424745	val-aft-nloglik:1.463148
[79]	train-aft-nloglik:1.424699	val-aft-nloglik:1.463165
[80]	train-aft-nloglik:1.424599	val-aft-nloglik:1.463172
[81]	train-aft-nloglik:1.424562	val-aft-nloglik:1.463174
[82]	train-aft-nloglik:1.424525	val-aft-nloglik:1.463163
[83]	train-aft-nloglik:1.424431	val-aft-nloglik:1.463157
[84]	train-aft-nloglik:1.424402	val-aft-nloglik:1.463156
[85]	train-aft-nloglik:1.424271	val-aft-nloglik:1.463181
[86]	train-aft-nloglik:1.424185	val-aft-nloglik:1.463178
[87]	train-aft-nloglik:1.424097	val-aft-nloglik:1.463165
[88]	train-aft-nloglik:1.424028	val-aft-nloglik:1.463137
[89]	train-aft-nloglik:1.424006	val-aft-nloglik:1.463133
[90]	train-aft-nloglik:1.423928	val-aft-nloglik:1.463140
[91]	train-aft-nloglik:1.423840	val-aft-nloglik:1.463175
[92]	train-aft-nloglik:1.423811	val-aft-nloglik:1.463180
[93]	train-aft-nloglik:1.423780	val-aft-nloglik:1.463177
[94]	train-aft-nloglik:1.423712	val-aft-nloglik:1.463171
[95]	train-aft-nloglik:1.423632	val-aft-nloglik:1.463201
[96]	train-aft-nloglik:1.423597	val-aft-nloglik:1.463204
[97]	train-aft-nloglik:1.423576	val-aft-nloglik:1.463204
[98]	train-aft-nloglik:1.423576	val-aft-nloglik:1.463204
[99]	train-aft-nloglik:1.423576	val-aft-nloglik:1.463204

Stopping. Best iteration:

```
[89]    train-aft-nloglik:1.424006   val-aft-nloglik:1.463133
```

```
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:14.061364 val-aft-nloglik:14.074756
```

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:11.701796 val-aft-nloglik:11.716477  
[3] train-aft-nloglik:9.796189  val-aft-nloglik:9.812065  
[4] train-aft-nloglik:8.255162  val-aft-nloglik:8.272145  
[5] train-aft-nloglik:7.007131  val-aft-nloglik:7.025147  
[6] train-aft-nloglik:5.994775  val-aft-nloglik:6.013753  
[7] train-aft-nloglik:5.172181  val-aft-nloglik:5.192060  
[8] train-aft-nloglik:4.502575  val-aft-nloglik:4.523299  
[9] train-aft-nloglik:3.956487  val-aft-nloglik:3.978006  
[10] train-aft-nloglik:3.510292  val-aft-nloglik:3.532559  
[11] train-aft-nloglik:3.145027  val-aft-nloglik:3.167986  
[12] train-aft-nloglik:2.845464  val-aft-nloglik:2.869125  
[13] train-aft-nloglik:2.599352  val-aft-nloglik:2.623650  
[14] train-aft-nloglik:2.396823  val-aft-nloglik:2.421746  
[15] train-aft-nloglik:2.229909  val-aft-nloglik:2.255460
```

[16]	train-aft-nloglik:2.092158	val-aft-nloglik:2.118356
[17]	train-aft-nloglik:1.978319	val-aft-nloglik:2.005010
[18]	train-aft-nloglik:1.884152	val-aft-nloglik:1.911441
[19]	train-aft-nloglik:1.806186	val-aft-nloglik:1.833996
[20]	train-aft-nloglik:1.741616	val-aft-nloglik:1.770017
[21]	train-aft-nloglik:1.688104	val-aft-nloglik:1.717089
[22]	train-aft-nloglik:1.643761	val-aft-nloglik:1.673135
[23]	train-aft-nloglik:1.607020	val-aft-nloglik:1.636912
[24]	train-aft-nloglik:1.576557	val-aft-nloglik:1.606895
[25]	train-aft-nloglik:1.551337	val-aft-nloglik:1.582016
[26]	train-aft-nloglik:1.530438	val-aft-nloglik:1.561517
[27]	train-aft-nloglik:1.513149	val-aft-nloglik:1.544610
[28]	train-aft-nloglik:1.498837	val-aft-nloglik:1.530628
[29]	train-aft-nloglik:1.487012	val-aft-nloglik:1.519184
[30]	train-aft-nloglik:1.477203	val-aft-nloglik:1.509629
[31]	train-aft-nloglik:1.469096	val-aft-nloglik:1.501914
[32]	train-aft-nloglik:1.462321	val-aft-nloglik:1.495458
[33]	train-aft-nloglik:1.456733	val-aft-nloglik:1.490101
[34]	train-aft-nloglik:1.452125	val-aft-nloglik:1.485747
[35]	train-aft-nloglik:1.447956	val-aft-nloglik:1.481956
[36]	train-aft-nloglik:1.444627	val-aft-nloglik:1.478845
[37]	train-aft-nloglik:1.441715	val-aft-nloglik:1.476167
[38]	train-aft-nloglik:1.439404	val-aft-nloglik:1.474002
[39]	train-aft-nloglik:1.437350	val-aft-nloglik:1.472202
[40]	train-aft-nloglik:1.435689	val-aft-nloglik:1.470781
[41]	train-aft-nloglik:1.434194	val-aft-nloglik:1.469603
[42]	train-aft-nloglik:1.432850	val-aft-nloglik:1.468681
[43]	train-aft-nloglik:1.431784	val-aft-nloglik:1.467669
[44]	train-aft-nloglik:1.430797	val-aft-nloglik:1.466878
[45]	train-aft-nloglik:1.429995	val-aft-nloglik:1.466127
[46]	train-aft-nloglik:1.429182	val-aft-nloglik:1.465633
[47]	train-aft-nloglik:1.428596	val-aft-nloglik:1.465334
[48]	train-aft-nloglik:1.428039	val-aft-nloglik:1.464928
[49]	train-aft-nloglik:1.427448	val-aft-nloglik:1.464566
[50]	train-aft-nloglik:1.426956	val-aft-nloglik:1.464252
[51]	train-aft-nloglik:1.426389	val-aft-nloglik:1.464007
[52]	train-aft-nloglik:1.425920	val-aft-nloglik:1.463837
[53]	train-aft-nloglik:1.425452	val-aft-nloglik:1.463641
[54]	train-aft-nloglik:1.425086	val-aft-nloglik:1.463459
[55]	train-aft-nloglik:1.424740	val-aft-nloglik:1.463331
[56]	train-aft-nloglik:1.424290	val-aft-nloglik:1.463165
[57]	train-aft-nloglik:1.423940	val-aft-nloglik:1.463018
[58]	train-aft-nloglik:1.423551	val-aft-nloglik:1.462956

```

[59] train-aft-nloglik:1.423169 val-aft-nloglik:1.462933
[60] train-aft-nloglik:1.422933 val-aft-nloglik:1.462917
[61] train-aft-nloglik:1.422637 val-aft-nloglik:1.462919
[62] train-aft-nloglik:1.422318 val-aft-nloglik:1.462899
[63] train-aft-nloglik:1.421998 val-aft-nloglik:1.462752
[64] train-aft-nloglik:1.421589 val-aft-nloglik:1.462650
[65] train-aft-nloglik:1.421451 val-aft-nloglik:1.462591
[66] train-aft-nloglik:1.421184 val-aft-nloglik:1.462553
[67] train-aft-nloglik:1.420875 val-aft-nloglik:1.462545
[68] train-aft-nloglik:1.420625 val-aft-nloglik:1.462554
[69] train-aft-nloglik:1.420506 val-aft-nloglik:1.462477
[70] train-aft-nloglik:1.420161 val-aft-nloglik:1.462472
[71] train-aft-nloglik:1.419940 val-aft-nloglik:1.462466
[72] train-aft-nloglik:1.419632 val-aft-nloglik:1.462432
[73] train-aft-nloglik:1.419394 val-aft-nloglik:1.462385
[74] train-aft-nloglik:1.419201 val-aft-nloglik:1.462435
[75] train-aft-nloglik:1.419044 val-aft-nloglik:1.462447
[76] train-aft-nloglik:1.418747 val-aft-nloglik:1.462377
[77] train-aft-nloglik:1.418488 val-aft-nloglik:1.462404
[78] train-aft-nloglik:1.418303 val-aft-nloglik:1.462429
[79] train-aft-nloglik:1.418055 val-aft-nloglik:1.462480
[80] train-aft-nloglik:1.417834 val-aft-nloglik:1.462468
[81] train-aft-nloglik:1.417645 val-aft-nloglik:1.462522
[82] train-aft-nloglik:1.417346 val-aft-nloglik:1.462480
[83] train-aft-nloglik:1.417212 val-aft-nloglik:1.462479
[84] train-aft-nloglik:1.416996 val-aft-nloglik:1.462441
[85] train-aft-nloglik:1.416792 val-aft-nloglik:1.462415
[86] train-aft-nloglik:1.416629 val-aft-nloglik:1.462452
Stopping. Best iteration:
[76] train-aft-nloglik:1.418747 val-aft-nloglik:1.462377

```

```
# SHAP?
```