

Build Survival Model: XGBoost

Mingcheng Hu

Table of contents

Load Data	2
XGBoost	6
Data Preparation	6
Hyperparameter Tuning	7
Variable Selection	10
Cross Validation to Select the Best Number of Features	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/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, "imputed") #
  ↪ 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: imputed"
```

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_SDS"
[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 <- 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] 0.1483341
```

```
$max_depth
```

```
[1] 2
```

```
$min_child_weight
```

```
[1] 0.0106544
```

```
$reg_alpha
```

```
[1] 0.5021755
```

```
$reg_lambda
```

```
[1] 0.03295546
```

```
$aft_loss_distribution_scale
```

```
[1] 1.353221
```

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:19.456536 val-aft-nloglik:19.454136
```

Multiple eval metrics are present. Will use val_aft_nloglik for early stopping.
Will train until val_aft_nloglik hasn't improved in 10 rounds.

```
[2] train-aft-nloglik:14.740032 val-aft-nloglik:14.741504
```

```
[3] train-aft-nloglik:11.288312 val-aft-nloglik:11.292653
```

```
[4] train-aft-nloglik:8.755920 val-aft-nloglik:8.762477
```

```
[5] train-aft-nloglik:6.892467 val-aft-nloglik:6.901157
```

```
[6] train-aft-nloglik:5.516612 val-aft-nloglik:5.526814
```

```
[7] train-aft-nloglik:4.496918 val-aft-nloglik:4.508555
```

```
[8] train-aft-nloglik:3.738140 val-aft-nloglik:3.750936
```

```
[9] train-aft-nloglik:3.171110 val-aft-nloglik:3.185393
```

```
[10] train-aft-nloglik:2.745627 val-aft-nloglik:2.760977
```

```
[11] train-aft-nloglik:2.425058 val-aft-nloglik:2.441205
```

[12]	train-aft-nloglik:2.182621	val-aft-nloglik:2.199456
[13]	train-aft-nloglik:1.998640	val-aft-nloglik:2.016236
[14]	train-aft-nloglik:1.858651	val-aft-nloglik:1.877058
[15]	train-aft-nloglik:1.751850	val-aft-nloglik:1.770742
[16]	train-aft-nloglik:1.670195	val-aft-nloglik:1.689899
[17]	train-aft-nloglik:1.607729	val-aft-nloglik:1.627811
[18]	train-aft-nloglik:1.559862	val-aft-nloglik:1.580412
[19]	train-aft-nloglik:1.523131	val-aft-nloglik:1.544007
[20]	train-aft-nloglik:1.495023	val-aft-nloglik:1.516294
[21]	train-aft-nloglik:1.473387	val-aft-nloglik:1.495321
[22]	train-aft-nloglik:1.456849	val-aft-nloglik:1.479087
[23]	train-aft-nloglik:1.444138	val-aft-nloglik:1.466730
[24]	train-aft-nloglik:1.434349	val-aft-nloglik:1.457239
[25]	train-aft-nloglik:1.426725	val-aft-nloglik:1.450230
[26]	train-aft-nloglik:1.420900	val-aft-nloglik:1.444661
[27]	train-aft-nloglik:1.416386	val-aft-nloglik:1.440517
[28]	train-aft-nloglik:1.412829	val-aft-nloglik:1.437138
[29]	train-aft-nloglik:1.410045	val-aft-nloglik:1.434815
[30]	train-aft-nloglik:1.407825	val-aft-nloglik:1.432811
[31]	train-aft-nloglik:1.406048	val-aft-nloglik:1.431320
[32]	train-aft-nloglik:1.404615	val-aft-nloglik:1.430107
[33]	train-aft-nloglik:1.403509	val-aft-nloglik:1.429166
[34]	train-aft-nloglik:1.402506	val-aft-nloglik:1.428375
[35]	train-aft-nloglik:1.401687	val-aft-nloglik:1.427915
[36]	train-aft-nloglik:1.401034	val-aft-nloglik:1.427366
[37]	train-aft-nloglik:1.400454	val-aft-nloglik:1.427042
[38]	train-aft-nloglik:1.399973	val-aft-nloglik:1.426741
[39]	train-aft-nloglik:1.399488	val-aft-nloglik:1.426585
[40]	train-aft-nloglik:1.399077	val-aft-nloglik:1.426303
[41]	train-aft-nloglik:1.398684	val-aft-nloglik:1.426062
[42]	train-aft-nloglik:1.398299	val-aft-nloglik:1.425867
[43]	train-aft-nloglik:1.397994	val-aft-nloglik:1.425591
[44]	train-aft-nloglik:1.397690	val-aft-nloglik:1.425316
[45]	train-aft-nloglik:1.397393	val-aft-nloglik:1.425161
[46]	train-aft-nloglik:1.397079	val-aft-nloglik:1.424953
[47]	train-aft-nloglik:1.396799	val-aft-nloglik:1.424867
[48]	train-aft-nloglik:1.396572	val-aft-nloglik:1.424709
[49]	train-aft-nloglik:1.396338	val-aft-nloglik:1.424693
[50]	train-aft-nloglik:1.396072	val-aft-nloglik:1.424536
[51]	train-aft-nloglik:1.395842	val-aft-nloglik:1.424458
[52]	train-aft-nloglik:1.395670	val-aft-nloglik:1.424343
[53]	train-aft-nloglik:1.395433	val-aft-nloglik:1.424226
[54]	train-aft-nloglik:1.395237	val-aft-nloglik:1.424114

```

[55] train-aft-nloglik:1.395024 val-aft-nloglik:1.424089
[56] train-aft-nloglik:1.394764 val-aft-nloglik:1.424051
[57] train-aft-nloglik:1.394587 val-aft-nloglik:1.423994
[58] train-aft-nloglik:1.394414 val-aft-nloglik:1.423863
[59] train-aft-nloglik:1.394201 val-aft-nloglik:1.423743
[60] train-aft-nloglik:1.394040 val-aft-nloglik:1.423735
[61] train-aft-nloglik:1.393831 val-aft-nloglik:1.423768
[62] train-aft-nloglik:1.393659 val-aft-nloglik:1.423881
[63] train-aft-nloglik:1.393479 val-aft-nloglik:1.423784
[64] train-aft-nloglik:1.393348 val-aft-nloglik:1.423754
[65] train-aft-nloglik:1.393211 val-aft-nloglik:1.423806
[66] train-aft-nloglik:1.393041 val-aft-nloglik:1.423663
[67] train-aft-nloglik:1.392873 val-aft-nloglik:1.423561
[68] train-aft-nloglik:1.392716 val-aft-nloglik:1.423459
[69] train-aft-nloglik:1.392593 val-aft-nloglik:1.423440
[70] train-aft-nloglik:1.392444 val-aft-nloglik:1.423326
[71] train-aft-nloglik:1.392289 val-aft-nloglik:1.423259
[72] train-aft-nloglik:1.392125 val-aft-nloglik:1.423144
[73] train-aft-nloglik:1.392034 val-aft-nloglik:1.422975
[74] train-aft-nloglik:1.391814 val-aft-nloglik:1.422907
[75] train-aft-nloglik:1.391634 val-aft-nloglik:1.422822
[76] train-aft-nloglik:1.391541 val-aft-nloglik:1.422775
[77] train-aft-nloglik:1.391369 val-aft-nloglik:1.422736
[78] train-aft-nloglik:1.391246 val-aft-nloglik:1.422834
[79] train-aft-nloglik:1.391106 val-aft-nloglik:1.422741
[80] train-aft-nloglik:1.390999 val-aft-nloglik:1.422734
[81] train-aft-nloglik:1.390853 val-aft-nloglik:1.422831
[82] train-aft-nloglik:1.390711 val-aft-nloglik:1.422907
[83] train-aft-nloglik:1.390559 val-aft-nloglik:1.422887
[84] train-aft-nloglik:1.390436 val-aft-nloglik:1.422874
[85] train-aft-nloglik:1.390330 val-aft-nloglik:1.422920
[86] train-aft-nloglik:1.390200 val-aft-nloglik:1.422984
[87] train-aft-nloglik:1.390051 val-aft-nloglik:1.422897
[88] train-aft-nloglik:1.389913 val-aft-nloglik:1.423015
[89] train-aft-nloglik:1.389756 val-aft-nloglik:1.423023
[90] train-aft-nloglik:1.389606 val-aft-nloglik:1.422914
Stopping. Best iteration:
[80] train-aft-nloglik:1.390999 val-aft-nloglik:1.422734

```

```

# 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

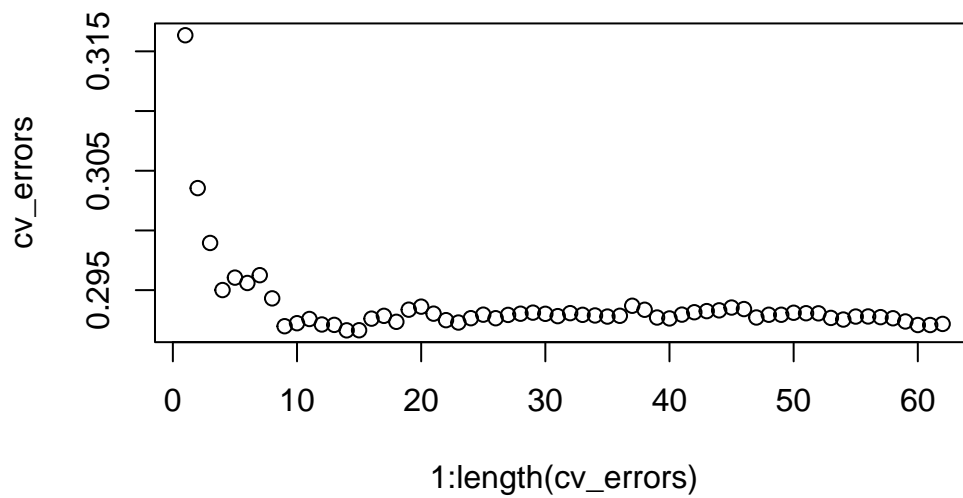
    # Use C-index to measure the performance of the model
    1 - concordance.index(pred, val_y_fold$time,
        ↪ val_y_fold$event)$c.index
    })
    print(mean(fold_errors))
    mean(fold_errors)
}, title = "Cross Validation to Select the Best Number of Features")

```

```

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 14"
```

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:19.456536 val-aft-nloglik:19.454136
 Multiple eval metrics are present. Will use val_aft_nloglik for early stopping.
 Will train until val_aft_nloglik hasn't improved in 10 rounds.

```

[2] train-aft-nloglik:14.740032 val-aft-nloglik:14.741504
[3] train-aft-nloglik:11.288312 val-aft-nloglik:11.292653
[4] train-aft-nloglik:8.755920 val-aft-nloglik:8.762477
[5] train-aft-nloglik:6.892467 val-aft-nloglik:6.901157
[6] train-aft-nloglik:5.516612 val-aft-nloglik:5.526814
[7] train-aft-nloglik:4.496918 val-aft-nloglik:4.508555
[8] train-aft-nloglik:3.738140 val-aft-nloglik:3.750936
[9] train-aft-nloglik:3.171110 val-aft-nloglik:3.185393
[10] train-aft-nloglik:2.745627 val-aft-nloglik:2.760977
[11] train-aft-nloglik:2.425058 val-aft-nloglik:2.441205
[12] train-aft-nloglik:2.182621 val-aft-nloglik:2.199456
[13] train-aft-nloglik:1.998640 val-aft-nloglik:2.016236

```


[14]	train-aft-nloglik:1.858651	val-aft-nloglik:1.877058
[15]	train-aft-nloglik:1.751850	val-aft-nloglik:1.770742
[16]	train-aft-nloglik:1.670195	val-aft-nloglik:1.689899
[17]	train-aft-nloglik:1.607729	val-aft-nloglik:1.627811
[18]	train-aft-nloglik:1.559862	val-aft-nloglik:1.580412
[19]	train-aft-nloglik:1.523131	val-aft-nloglik:1.544007
[20]	train-aft-nloglik:1.495023	val-aft-nloglik:1.516294
[21]	train-aft-nloglik:1.473387	val-aft-nloglik:1.495321
[22]	train-aft-nloglik:1.456849	val-aft-nloglik:1.479087
[23]	train-aft-nloglik:1.444138	val-aft-nloglik:1.466730
[24]	train-aft-nloglik:1.434349	val-aft-nloglik:1.457239
[25]	train-aft-nloglik:1.426725	val-aft-nloglik:1.450230
[26]	train-aft-nloglik:1.420900	val-aft-nloglik:1.444661
[27]	train-aft-nloglik:1.416410	val-aft-nloglik:1.440613
[28]	train-aft-nloglik:1.412856	val-aft-nloglik:1.437238
[29]	train-aft-nloglik:1.410071	val-aft-nloglik:1.434917
[30]	train-aft-nloglik:1.407859	val-aft-nloglik:1.433010
[31]	train-aft-nloglik:1.406090	val-aft-nloglik:1.431523
[32]	train-aft-nloglik:1.404652	val-aft-nloglik:1.430304
[33]	train-aft-nloglik:1.403548	val-aft-nloglik:1.429363
[34]	train-aft-nloglik:1.402546	val-aft-nloglik:1.428563
[35]	train-aft-nloglik:1.401726	val-aft-nloglik:1.428104
[36]	train-aft-nloglik:1.401073	val-aft-nloglik:1.427557
[37]	train-aft-nloglik:1.400500	val-aft-nloglik:1.427113
[38]	train-aft-nloglik:1.400045	val-aft-nloglik:1.426720
[39]	train-aft-nloglik:1.399557	val-aft-nloglik:1.426563
[40]	train-aft-nloglik:1.399153	val-aft-nloglik:1.426376
[41]	train-aft-nloglik:1.398802	val-aft-nloglik:1.426142
[42]	train-aft-nloglik:1.398425	val-aft-nloglik:1.425974
[43]	train-aft-nloglik:1.398082	val-aft-nloglik:1.425732
[44]	train-aft-nloglik:1.397786	val-aft-nloglik:1.425594
[45]	train-aft-nloglik:1.397492	val-aft-nloglik:1.425361
[46]	train-aft-nloglik:1.397249	val-aft-nloglik:1.425288
[47]	train-aft-nloglik:1.397023	val-aft-nloglik:1.425242
[48]	train-aft-nloglik:1.396770	val-aft-nloglik:1.425140
[49]	train-aft-nloglik:1.396530	val-aft-nloglik:1.425021
[50]	train-aft-nloglik:1.396306	val-aft-nloglik:1.424859
[51]	train-aft-nloglik:1.396125	val-aft-nloglik:1.424756
[52]	train-aft-nloglik:1.395915	val-aft-nloglik:1.424611
[53]	train-aft-nloglik:1.395741	val-aft-nloglik:1.424464
[54]	train-aft-nloglik:1.395564	val-aft-nloglik:1.424453
[55]	train-aft-nloglik:1.395397	val-aft-nloglik:1.424421
[56]	train-aft-nloglik:1.395203	val-aft-nloglik:1.424296

```

[57] train-aft-nloglik:1.394960 val-aft-nloglik:1.424318
[58] train-aft-nloglik:1.394796 val-aft-nloglik:1.424204
[59] train-aft-nloglik:1.394662 val-aft-nloglik:1.424035
[60] train-aft-nloglik:1.394514 val-aft-nloglik:1.424041
[61] train-aft-nloglik:1.394309 val-aft-nloglik:1.424076
[62] train-aft-nloglik:1.394158 val-aft-nloglik:1.423958
[63] train-aft-nloglik:1.394012 val-aft-nloglik:1.424030
[64] train-aft-nloglik:1.393924 val-aft-nloglik:1.424040
[65] train-aft-nloglik:1.393809 val-aft-nloglik:1.424051
[66] train-aft-nloglik:1.393679 val-aft-nloglik:1.424073
[67] train-aft-nloglik:1.393560 val-aft-nloglik:1.424069
[68] train-aft-nloglik:1.393466 val-aft-nloglik:1.424065
[69] train-aft-nloglik:1.393367 val-aft-nloglik:1.423931
[70] train-aft-nloglik:1.393297 val-aft-nloglik:1.423986
[71] train-aft-nloglik:1.393186 val-aft-nloglik:1.423900
[72] train-aft-nloglik:1.393061 val-aft-nloglik:1.423936
[73] train-aft-nloglik:1.392922 val-aft-nloglik:1.424009
[74] train-aft-nloglik:1.392758 val-aft-nloglik:1.423966
[75] train-aft-nloglik:1.392667 val-aft-nloglik:1.424012
[76] train-aft-nloglik:1.392571 val-aft-nloglik:1.423969
[77] train-aft-nloglik:1.392485 val-aft-nloglik:1.423973
[78] train-aft-nloglik:1.392385 val-aft-nloglik:1.424045
[79] train-aft-nloglik:1.392276 val-aft-nloglik:1.423984
[80] train-aft-nloglik:1.392179 val-aft-nloglik:1.423985
[81] train-aft-nloglik:1.392100 val-aft-nloglik:1.424012
Stopping. Best iteration:
[71] train-aft-nloglik:1.393186 val-aft-nloglik:1.423900

```

```

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:19.456536 val-aft-nloglik:19.454136
Multiple eval metrics are present. Will use val_aft_nloglik for early stopping.
Will train until val_aft_nloglik hasn't improved in 10 rounds.

```

[2] train-aft-nloglik:14.740032 val-aft-nloglik:14.741504
[3] train-aft-nloglik:11.288312 val-aft-nloglik:11.292653
[4] train-aft-nloglik:8.755920 val-aft-nloglik:8.762477
[5] train-aft-nloglik:6.892467 val-aft-nloglik:6.901157
[6] train-aft-nloglik:5.516612 val-aft-nloglik:5.526814
[7] train-aft-nloglik:4.496918 val-aft-nloglik:4.508555
[8] train-aft-nloglik:3.738140 val-aft-nloglik:3.750936
[9] train-aft-nloglik:3.171110 val-aft-nloglik:3.185393
[10] train-aft-nloglik:2.745627 val-aft-nloglik:2.760977
[11] train-aft-nloglik:2.425058 val-aft-nloglik:2.441205
[12] train-aft-nloglik:2.182621 val-aft-nloglik:2.199456
[13] train-aft-nloglik:1.998640 val-aft-nloglik:2.016236
[14] train-aft-nloglik:1.858651 val-aft-nloglik:1.877058
[15] train-aft-nloglik:1.751850 val-aft-nloglik:1.770742
[16] train-aft-nloglik:1.670195 val-aft-nloglik:1.689899
[17] train-aft-nloglik:1.607729 val-aft-nloglik:1.627811
[18] train-aft-nloglik:1.559862 val-aft-nloglik:1.580412
[19] train-aft-nloglik:1.523131 val-aft-nloglik:1.544007
[20] train-aft-nloglik:1.495023 val-aft-nloglik:1.516294
[21] train-aft-nloglik:1.473387 val-aft-nloglik:1.495321
[22] train-aft-nloglik:1.456849 val-aft-nloglik:1.479087
[23] train-aft-nloglik:1.444138 val-aft-nloglik:1.466730
[24] train-aft-nloglik:1.434349 val-aft-nloglik:1.457239
[25] train-aft-nloglik:1.426725 val-aft-nloglik:1.450230
[26] train-aft-nloglik:1.420900 val-aft-nloglik:1.444661

```

[27]	train-aft-nloglik:1.416386	val-aft-nloglik:1.440517
[28]	train-aft-nloglik:1.412829	val-aft-nloglik:1.437138
[29]	train-aft-nloglik:1.410045	val-aft-nloglik:1.434815
[30]	train-aft-nloglik:1.407825	val-aft-nloglik:1.432811
[31]	train-aft-nloglik:1.406048	val-aft-nloglik:1.431320
[32]	train-aft-nloglik:1.404615	val-aft-nloglik:1.430107
[33]	train-aft-nloglik:1.403509	val-aft-nloglik:1.429166
[34]	train-aft-nloglik:1.402506	val-aft-nloglik:1.428375
[35]	train-aft-nloglik:1.401687	val-aft-nloglik:1.427915
[36]	train-aft-nloglik:1.401034	val-aft-nloglik:1.427366
[37]	train-aft-nloglik:1.400454	val-aft-nloglik:1.427042
[38]	train-aft-nloglik:1.399973	val-aft-nloglik:1.426741
[39]	train-aft-nloglik:1.399488	val-aft-nloglik:1.426585
[40]	train-aft-nloglik:1.399077	val-aft-nloglik:1.426303
[41]	train-aft-nloglik:1.398684	val-aft-nloglik:1.426062
[42]	train-aft-nloglik:1.398299	val-aft-nloglik:1.425867
[43]	train-aft-nloglik:1.397994	val-aft-nloglik:1.425591
[44]	train-aft-nloglik:1.397690	val-aft-nloglik:1.425316
[45]	train-aft-nloglik:1.397393	val-aft-nloglik:1.425161
[46]	train-aft-nloglik:1.397079	val-aft-nloglik:1.424953
[47]	train-aft-nloglik:1.396799	val-aft-nloglik:1.424867
[48]	train-aft-nloglik:1.396572	val-aft-nloglik:1.424709
[49]	train-aft-nloglik:1.396338	val-aft-nloglik:1.424693
[50]	train-aft-nloglik:1.396072	val-aft-nloglik:1.424536
[51]	train-aft-nloglik:1.395842	val-aft-nloglik:1.424458
[52]	train-aft-nloglik:1.395670	val-aft-nloglik:1.424343
[53]	train-aft-nloglik:1.395433	val-aft-nloglik:1.424226
[54]	train-aft-nloglik:1.395237	val-aft-nloglik:1.424114
[55]	train-aft-nloglik:1.395024	val-aft-nloglik:1.424089
[56]	train-aft-nloglik:1.394764	val-aft-nloglik:1.424051
[57]	train-aft-nloglik:1.394587	val-aft-nloglik:1.423994
[58]	train-aft-nloglik:1.394414	val-aft-nloglik:1.423863
[59]	train-aft-nloglik:1.394201	val-aft-nloglik:1.423743
[60]	train-aft-nloglik:1.394040	val-aft-nloglik:1.423735
[61]	train-aft-nloglik:1.393831	val-aft-nloglik:1.423768
[62]	train-aft-nloglik:1.393659	val-aft-nloglik:1.423881
[63]	train-aft-nloglik:1.393479	val-aft-nloglik:1.423784
[64]	train-aft-nloglik:1.393348	val-aft-nloglik:1.423754
[65]	train-aft-nloglik:1.393211	val-aft-nloglik:1.423806
[66]	train-aft-nloglik:1.393041	val-aft-nloglik:1.423663
[67]	train-aft-nloglik:1.392873	val-aft-nloglik:1.423561
[68]	train-aft-nloglik:1.392716	val-aft-nloglik:1.423459
[69]	train-aft-nloglik:1.392593	val-aft-nloglik:1.423440

[70]	train-aft-nloglik:1.392444	val-aft-nloglik:1.423326
[71]	train-aft-nloglik:1.392289	val-aft-nloglik:1.423259
[72]	train-aft-nloglik:1.392125	val-aft-nloglik:1.423144
[73]	train-aft-nloglik:1.392034	val-aft-nloglik:1.422975
[74]	train-aft-nloglik:1.391814	val-aft-nloglik:1.422907
[75]	train-aft-nloglik:1.391634	val-aft-nloglik:1.422822
[76]	train-aft-nloglik:1.391541	val-aft-nloglik:1.422775
[77]	train-aft-nloglik:1.391369	val-aft-nloglik:1.422736
[78]	train-aft-nloglik:1.391246	val-aft-nloglik:1.422834
[79]	train-aft-nloglik:1.391106	val-aft-nloglik:1.422741
[80]	train-aft-nloglik:1.390999	val-aft-nloglik:1.422734
[81]	train-aft-nloglik:1.390853	val-aft-nloglik:1.422831
[82]	train-aft-nloglik:1.390711	val-aft-nloglik:1.422907
[83]	train-aft-nloglik:1.390559	val-aft-nloglik:1.422887
[84]	train-aft-nloglik:1.390436	val-aft-nloglik:1.422874
[85]	train-aft-nloglik:1.390330	val-aft-nloglik:1.422920
[86]	train-aft-nloglik:1.390200	val-aft-nloglik:1.422984
[87]	train-aft-nloglik:1.390051	val-aft-nloglik:1.422897
[88]	train-aft-nloglik:1.389913	val-aft-nloglik:1.423015
[89]	train-aft-nloglik:1.389756	val-aft-nloglik:1.423023
[90]	train-aft-nloglik:1.389606	val-aft-nloglik:1.422914

Stopping. Best iteration:

[80]	train-aft-nloglik:1.390999	val-aft-nloglik:1.422734
------	----------------------------	--------------------------