# Final Model - death_3year

Eduardo Yuki Yada

## Global parameters

```
k <- params$k # Number of folds for cross validation
grid_size <- params$grid_size # Number of parameter combination to tune on each model
max_auc_loss <- params$max_auc_loss # Max accepted loss of AUC for reducing num of features
repeats <- params$repeats
Hmisc::list.tree(params)
```

```
##  params = list 5 (952 bytes)
## .  max_auc_loss = double 1= 0.01
## .  outcome_column = character 1= death_3year
## .  k = double 1= 10
## .  grid_size = double 1= 50
## .  repeats = double 1= 2
```

Minutes to run: 0

## Imports

```
library(tidyverse)
library(yaml)
library(tidymodels)
library(usemodels)
library(vip)
library(kableExtra)
library(SHAPforxgboost)
library(xgboost)
library(Matrix)
library(mltools)
library(bonsai)
library(lightgbm)
library(pROC)
library(caret)
library(themis)

source("aux_functions.R")

select <- dplyr::select
```

Minutes to run: 0.001

## Loading data

```
load('dataset/processed_data.RData')
load('dataset/processed_dictionary.RData')

columns_list <- yaml.load_file("./auxiliar/columns_list.yaml")

outcome_column <- params$outcome_column
features_list <- params$features_list

df[columns_list$outcome_columns] <- lapply(df[columns_list$outcome_columns], factor)
df <- mutate(df, across(where(is.character), as.factor))
```

Minutes to run: 0.007

```
dir.create(file.path("./auxiliar/final_model/hyperparameters/"),
           showWarnings = FALSE,
           recursive = TRUE)

dir.create(file.path("./auxiliar/final_model/performance/"),
           showWarnings = FALSE,
           recursive = TRUE)

dir.create(file.path("./auxiliar/final_model/selected_features/"),
           showWarnings = FALSE,
           recursive = TRUE)
```

Minutes to run: 0

## Eligible features

```
cat_features_list = readRDS(sprintf(
  "./auxiliar/significant_columns/categorical_%s.rds",
  outcome_column
))

num_features_list = readRDS(sprintf(
  "./auxiliar/significant_columns/numerical_%s.rds",
  outcome_column
))

features_list = c(cat_features_list, num_features_list)
```

Minutes to run: 0

```
eligible_columns <- df_names %>%
  filter(momento.aquisicao == 'Admissão t0') %>%
  .$variable.name

exception_columns <- c('death_intraop', 'death_intraop_1', 'disch_outcomes_t0')

correlated_columns = c('year_procedure_1', # com year_adm_t0
                       'age_surgery_1', # com age
                       'admission_t0', # com admission_pre_t0_count
                       'atb', # com meds_antimicrobianos
                       'classe_meds_cardio_qtde', # com classe_meds_qtde
                       'suporte_hemod', # com proced_invasivos_qtde,
                       'radiografia', # com exames_imagem_qtde
```

```
                        'ecg' # com metodos_graficos_qtde
                        )

eligible_features <- eligible_columns %>%
  base::intersect(c(columns_list$categorical_columns, columns_list$numerical_columns)) %>%
  setdiff(c(exception_columns, correlated_columns))

features = base::intersect(eligible_features, features_list)
```

Minutes to run: 0

Starting features:

```
gluedown::md_order(features, seq = TRUE, pad = TRUE)
```

1. sex
2. age
3. race
4. education_level
5. underlying_heart_disease
6. heart_disease
7. nyha_basal
8. hypertension
9. prior_mi
10. heart_failure
11. af
12. cardiac_arrest
13. valvopathy
14. diabetes
15. renal_failure
16. hemodialysis
17. stroke
18. copd
19. comorbidities_count
20. procedure_type_1
21. reop_type_1
22. procedure_type_new
23. cied_final_1
24. cied_final_group_1
25. admission_pre_t0_count
26. admission_pre_t0_180d
27. year_adm_t0
28. icu_t0
29. dialysis_t0
30. admission_t0_emergency
31. aco
32. antiarritmico
33. ieca_bra
34. dva
35. digoxina
36. estatina
37. diuretico
38. vasodilatador
39. insuf_cardiaca
40. espironolactona
41. antiplaquetario_ev
42. insulina
43. anticonvulsivante
44. psicofarmacos
45. antifungico
46. classe_meds_qtde

47. meds_cardiovasc_qtde
48. meds_antimicrobianos
49. ventilacao_mecanica
50. transplante_cardiaco
51. outros_proced_cirurgicos
52. icp
53. angioplastia
54. cateterismo
55. eletrofisiologia
56. cateter_venoso_central
57. proced_invasivos_qtde
58. transfusao
59. equipe_multiprof
60. holter
61. teste_esforco
62. tilt_teste
63. metodos_graficos_qtde
64. laboratorio
65. cultura
66. analises_clinicas_qtde
67. citologia
68. histopatologia_qtde
69. angio_tc
70. angiografia
71. cintilografia
72. ecocardiograma
73. endoscopia
74. flebografia
75. pet_ct
76. ultrassom
77. tomografia
78. ressonancia
79. exames_imagem_qtde
80. bic
81. hospital_stay Minutes to run: 0

## Train test split (70%/30%)

```
set.seed(42)

if (outcome_column == 'readmission_30d') {
  df_split <- readRDS("dataset/split_object.rds")
} else {
  df_split <- initial_split(df, prop = .7, strata = all_of(outcome_column))
}

df_train <- training(df_split) %>% select(all_of(c(features, outcome_column)))
df_test <- testing(df_split) %>% select(all_of(c(features, outcome_column)))

df_folds <- vfold_cv(df_train, v = k,
                     strata = all_of(outcome_column),
                     repeats = repeats)
```

Minutes to run: 0.001

## Feature Selection

```
custom_dummy_names <- function(var, lvl, ordinal = FALSE) {
  dummy_names(var, lvl, ordinal = FALSE, sep = "__")
}

model_fit_wf <- function(df_train, features, outcome_column, hyperparameters){
  model_recipe <-
    recipe(formula = sprintf("%s ~ .", outcome_column) %>% as.formula,
           data = df_train %>% select(all_of(c(features, outcome_column)))) %>%
    step_novel(all_nominal_predictors()) %>%
    step_unknown(all_nominal_predictors()) %>%
    step_other(all_nominal_predictors(), threshold = 0.05, other = ".merged") %>%
    step_dummy(all_nominal_predictors(), naming = custom_dummy_names)

  model_spec <-
    do.call(boost_tree, hyperparameters) %>%
    set_engine("lightgbm") %>%
    set_mode("classification")

  model_workflow <-
    workflow() %>%
    add_recipe(model_recipe) %>%
    add_model(model_spec)

  model_fit_rs <- model_workflow %>%
    fit_resamples(df_folds)

  model_fit <- model_workflow %>%
    fit(df_train)

  model_auc <- validation(model_fit, df_test, plot = F)

  raw_model <- parsnip::extract_fit_engine(model_fit)

  feature_importance <- lgb.importance(raw_model, percentage = TRUE) %>%
    separate(Feature, c("Feature","value"), "__", fill = 'right') %>%
    group_by(Feature) %>%
    summarise(Gain = sum(Gain),
              Cover = sum(Cover),
              Frequency = sum(Frequency)) %>%
    ungroup() %>%
    arrange(desc(Gain))

  cv_results <- collect_metrics(model_fit_rs) %>% filter(.metric == 'roc_auc')

  return(
    list(
      cv_auc =  cv_results$mean,
      cv_auc_std_err = cv_results$std_err,
      importance = feature_importance,
      auc = as.numeric(model_auc$auc),
      auc_lower = model_auc$ci[1],
      auc_upper = model_auc$ci[3]
    )
  )
}
```

Minutes to run: 0

```
hyperparameters <- readRDS(
  sprintf(
    "./auxiliar/model_selection/hyperparameters/lightgbm_%s.rds",
    outcome_column
  )
)

hyperparameters$sample_size <- 1

full_model <- model_fit_wf(df_train, features, outcome_column, hyperparameters)

sprintf('Full Model CV Train AUC: %.3f' ,full_model$cv_auc)
```

```
## [1] "Full Model CV Train AUC: 0.799"
```

```
sprintf('Full Model Test AUC: %.3f' ,full_model$auc)
```

```
## [1] "Full Model Test AUC: 0.805"
```

Minutes to run: 0.43

Features with zero importance on the initial model:

```
unimportant_features <- setdiff(features, full_model$importance$Feature)

unimportant_features %>%
  gluedown::md_order()
```

1. hemodialysis
2. transplante_cardiaco
3. angioplastia
4. transfusao
5. tilt_teste
6. histopatologia_qtde
7. angio_tc
8. angiografia
9. pet_ct Minutes to run: 0

```
trimmed_features <- full_model$importance$Feature
trimmed_model <- model_fit_wf(df_train, trimmed_features,
                              outcome_column, hyperparameters)

sprintf('Trimmed Model CV Train AUC: %.3f' ,trimmed_model$cv_auc)
```

```
## [1] "Trimmed Model CV Train AUC: 0.800"
```

```
sprintf('Trimmed Model Test AUC: %.3f' ,trimmed_model$auc)
```

```
## [1] "Trimmed Model Test AUC: 0.805"
```

Minutes to run: 0.419

```
selection_results <- tibble::tribble(
  ~`Tested Feature`, ~`Dropped`, ~`Number of Features`, ~`CV AUC`, ~`CV AUC Std Error`, ~`Total AUC Loss`, ~`Ins
  'None', TRUE, length(features), full_model$cv_auc, full_model$cv_auc_std_err, 0, 0
)
```

```
whitelist <- c()

if (full_model$cv_auc - trimmed_model$cv_auc < max_auc_loss) {
  current_features <- trimmed_features
  current_model <- trimmed_model
  current_auc_loss <- full_model$cv_auc - current_model$cv_auc
  instant_auc_loss <- full_model$cv_auc - current_model$cv_auc

  selection_results <- selection_results %>%
    add_row(`Tested Feature` = 'All unimportant',
            `Dropped` = TRUE,
            `Number of Features` = length(trimmed_features),
            `CV AUC` = current_model$cv_auc,
            `CV AUC Std Error` = current_model$cv_auc_std_err,
            `Total AUC Loss` = current_auc_loss,
            `Instant AUC Loss` = instant_auc_loss)
} else {
  current_features <- features
  current_model <- full_model
  current_auc_loss <- 0
}

while (current_auc_loss < max_auc_loss & mean(current_features %in% whitelist) < 1) {
  zero_importance_features <-
    setdiff(current_features, current_model$importance$Feature) %>%
    setdiff(whitelist)
  if (length(zero_importance_features) > 0) {
    current_least_important <- zero_importance_features[1]
  } else {
    current_least_important <-
      tail(setdiff(current_model$importance$Feature, whitelist), 1)
  }
  test_features <-
    setdiff(current_features, current_least_important)
  current_model <-
    model_fit_wf(df_train, test_features, outcome_column, hyperparameters)
  instant_auc_loss <-
    tail(selection_results %>% filter(Dropped) %>% .$`CV AUC`, n = 1) - current_model$cv_auc

  if (instant_auc_loss < max_auc_loss / 5 &
      current_auc_loss < max_auc_loss) {
    dropped <- TRUE
    current_features <- test_features
    current_auc_loss <- full_model$cv_auc - current_model$cv_auc
  } else {
    dropped <- FALSE
    whitelist <- c(whitelist, current_least_important)
  }

  selection_results <- selection_results %>%
    add_row(
      `Tested Feature` = current_least_important,
      `Dropped` = dropped,
      `Number of Features` = length(test_features),
      `CV AUC` = current_model$cv_auc,
      `CV AUC Std Error` = current_model$cv_auc_std_err,
      `Total AUC Loss` = current_auc_loss,
      `Instant AUC Loss` = instant_auc_loss
    )

  print(c(
```

```
    length(current_features),
    round(current_auc_loss, 4),
    round(instant_auc_loss, 4),
    current_least_important
  ))
}
```

```
## [1] "71"                 "-7e-04"             "0"                  "antiplaquetario_ev"
## [1] "70"               "-1e-04"           "6e-04"            "eletrofisiologia"
## [1] "69"         "-6e-04"     "-5e-04"     "endoscopia"
## [1] "68"     "-1e-04" "5e-04"  "icp"
## [1] "67"             "-5e-04"         "-4e-04"         "cardiac_arrest"
## [1] "66"           "-2e-04"        "3e-04"         "teste_esforco"
## [1] "65"          "-9e-04"      "-7e-04"      "flebografia"
## [1] "64"          "-9e-04"      "0"           "reop_type_1"
## [1] "63"            "-5e-04"        "4e-04"         "cintilografia"
## [1] "62"          "-7e-04"      "-2e-04"      "dialysis_t0"
## [1] "61"                     "-3e-04"                 "5e-04"                  "cateter_venoso_central"
## [1] "60"       "3e-04"    "6e-04"    "insulina"
## [1] "59"        "1e-04"     "-2e-04"    "citologia"
## [1] "58"    "4e-04" "3e-04" "copd"
## [1] "57"          "-0.001"      "-0.0014"     "ressonancia"
## [1] "56"               "-0.0023"           "-0.0014"           "anticonvulsivante"
## [1] "55"     "-8e-04" "0.0015" "stroke"
## [1] "54"                     "-6e-04"                   "3e-04"                    "outros_proced_cirurgico
## [1] "53"     "1e-04"  "6e-04"  "holter"
## [1] "52"          "-6e-04"      "-7e-04"      "cateterismo"
## [1] "51"                "-0.0014"            "-8e-04"             "procedure_type_new"
## [1] "50"          "-0.0027"     "-0.0013"     "antifungico"
## [1] "49"       "-0.0017"  "0.001"    "prior_mi"
## [1] "49"            "-0.0017"       "0.0025"        "heart_failure"
## [1] "49"        "-0.0017"   "0.0021"    "ultrassom"
## [1] "48"                  "-0.0017"             "0"                    "ventilacao_mecanica"
## [1] "47"     "-8e-04" "9e-04"  "aco"
## [1] "46"      "-0.0016" "-7e-04"  "dva"
## [1] "45"       "-0.0014"  "2e-04"    "diabetes"
## [1] "44"                    "-0.002"                 "-6e-04"                 "analises_clinicas_qtde"
## [1] "43"            "-0.0013"       "7e-04"         "heart_disease"
## [1] "42"     "-4e-04" "9e-04"  "bic"
## [1] "41"                   "-0.0028"               "-0.0024"               "proced_invasivos_qtde"
## [1] "40"                    "-0.0014"                 "0.0015"                 "admission_t0_emergency"
## [1] "39"             "-0.0019"        "-5e-04"         "ecocardiograma"
## [1] "38"           "-8e-04"       "0.001"        "hypertension"
## [1] "37"     "-2e-04" "6e-04"  "af"
## [1] "36"      "-0.0021" "-0.0019" "sex"
## [1] "35"       "-0.0018"  "3e-04"    "digoxina"
## [1] "34"               "-8e-04"           "0.001"            "procedure_type_1"
## [1] "33"                   "-0.0029"               "-0.0021"               "admission_pre_t0_180d"
## [1] "32"           "-0.0013"      "0.0016"       "cied_final_1"
## [1] "31"         "-9e-04"     "3e-04"      "tomografia"
## [1] "30"         "-0.0028"    "-0.0018"    "valvopathy"
## [1] "29"      "-0.0016" "0.0012"  "cultura"
## [1] "28"     "-3e-04" "0.0013" "race"
## [1] "27"            "-8e-04"        "-5e-04"        "renal_failure"
## [1] "26"                 "-9e-04"             "-1e-04"             "cied_final_group_1"
## [1] "25"       "-0.0019"  "-0.001"   "estatina"
## [1] "24"                     "-0.0016"                  "3e-04"                    "underlying_heart_diseas
## [1] "23"           "-0.0014"       "2e-04"         "antiarritmico"
## [1] "23"                 "-0.0014"           "0.0024"            "exames_imagem_qtde"
## [1] "22"                    "-0.0014"               "1e-04"                 "metodos_graficos_qtde"
## [1] "22"               "-0.0014"          "0.0021"           "equipe_multiprof"
```

```
## [1] "21"                    "-0.0015"              "-1e-04"               "meds_antimicrobianos"
## [1] "21"             "-0.0015"       "0.0038"        "vasodilatador"
## [1] "21"       "-0.0015" "0.0025"  "icu_t0"
## [1] "21"                "-0.0015"          "0.0028"           "classe_meds_qtde"
## [1] "20"              "5e-04"           "0.002"          "insuf_cardiaca"
## [1] "19"              "0.0017"         "0.0012"         "psicofarmacos"
## [1] "18"         "0.0019"    "2e-04"     "diuretico"
## [1] "18"          "0.0019"     "0.0059"     "nyha_basal"
## [1] "17"           "-0.0014"     "-0.0033"     "laboratorio"
## [1] "16"        "-2e-04"   "0.0012"   "ieca_bra"
## [1] "16"                   "-2e-04"              "0.0033"              "comorbidities_count"
## [1] "16"               "-2e-04"          "0.0053"          "education_level"
## [1] "15"                    "0"                      "2e-04"                "meds_cardiovasc_qtde"
## [1] "15"               "0"                "0.0029"          "espironolactona"
## [1] "15"                       "0"                         "0.0203"                  "admission_pre_t0_count"
## [1] "15"    "0"     "0.005" "age"
## [1] "15"           "0"            "0.0178"       "year_adm_t0"
## [1] "15"             "0"              "0.0101"         "hospital_stay"
```

```
selection_results %>%
  rename(Features = `Number of Features`) %>%
  niceFormatting(digits = 4, label = 1)
```

Table 1:

| Tested Feature | Dropped | Features | CV AUC | CV AUC Std Error | Total AUC Loss | Instant AUC Loss |
|---|---|---|---|---|---|---|
| None | TRUE | 81 | 0.7991 | 0.0062 | 0.0000 | 0.0000 |
| All unimportant | TRUE | 72 | 0.7998 | 0.0058 | -0.0007 | -0.0007 |
| antiplaquetario_ev | TRUE | 71 | 0.7998 | 0.0057 | -0.0007 | 0.0000 |
| eletrofisiologia | TRUE | 70 | 0.7992 | 0.0057 | -0.0001 | 0.0006 |
| endoscopia | TRUE | 69 | 0.7998 | 0.0058 | -0.0006 | -0.0005 |
| icp | TRUE | 68 | 0.7992 | 0.0056 | -0.0001 | 0.0005 |
| cardiac_arrest | TRUE | 67 | 0.7996 | 0.0059 | -0.0005 | -0.0004 |
| teste_esforco | TRUE | 66 | 0.7994 | 0.0058 | -0.0002 | 0.0003 |
| flebografia | TRUE | 65 | 0.8000 | 0.0060 | -0.0009 | -0.0007 |
| reop_type_1 | TRUE | 64 | 0.8000 | 0.0060 | -0.0009 | 0.0000 |
| cintilografia | TRUE | 63 | 0.7997 | 0.0059 | -0.0005 | 0.0004 |
| dialysis_t0 | TRUE | 62 | 0.7999 | 0.0061 | -0.0007 | -0.0002 |
| cateter_venoso_central | TRUE | 61 | 0.7994 | 0.0063 | -0.0003 | 0.0005 |
| insulina | TRUE | 60 | 0.7988 | 0.0062 | 0.0003 | 0.0006 |
| citologia | TRUE | 59 | 0.7990 | 0.0060 | 0.0001 | -0.0002 |
| copd | TRUE | 58 | 0.7987 | 0.0061 | 0.0004 | 0.0003 |
| ressonancia | TRUE | 57 | 0.8001 | 0.0059 | -0.0010 | -0.0014 |
| anticonvulsivante | TRUE | 56 | 0.8015 | 0.0061 | -0.0023 | -0.0014 |
| stroke | TRUE | 55 | 0.8000 | 0.0061 | -0.0008 | 0.0015 |
| outros_proced_cirurgicos | TRUE | 54 | 0.7997 | 0.0059 | -0.0006 | 0.0003 |
| holter | TRUE | 53 | 0.7991 | 0.0063 | 0.0001 | 0.0006 |
| cateterismo | TRUE | 52 | 0.7997 | 0.0063 | -0.0006 | -0.0007 |
| procedure_type_new | TRUE | 51 | 0.8005 | 0.0063 | -0.0014 | -0.0008 |
| antifungico | TRUE | 50 | 0.8018 | 0.0062 | -0.0027 | -0.0013 |
| prior_mi | TRUE | 49 | 0.8008 | 0.0063 | -0.0017 | 0.0010 |
| heart_failure | FALSE | 48 | 0.7983 | 0.0064 | -0.0017 | 0.0025 |
| ultrassom | FALSE | 48 | 0.7987 | 0.0060 | -0.0017 | 0.0021 |
| ventilacao_mecanica | TRUE | 48 | 0.8008 | 0.0060 | -0.0017 | 0.0000 |
| aco | TRUE | 47 | 0.8000 | 0.0058 | -0.0008 | 0.0009 |
| dva | TRUE | 46 | 0.8007 | 0.0062 | -0.0016 | -0.0007 |
| diabetes | TRUE | 45 | 0.8005 | 0.0062 | -0.0014 | 0.0002 |

Table 1: *(continued)*

| Tested Feature | Dropped | Features | CV AUC | CV AUC Std Error | Total AUC Loss | Instant AUC Loss |
|---|---|---|---|---|---|---|
| analises_clinicas_qtde | TRUE | 44 | 0.8011 | 0.0063 | -0.0020 | -0.0006 |
| heart_disease | TRUE | 43 | 0.8005 | 0.0067 | -0.0013 | 0.0007 |
| bic | TRUE | 42 | 0.7996 | 0.0066 | -0.0004 | 0.0009 |
| proced_invasivos_qtde | TRUE | 41 | 0.8020 | 0.0062 | -0.0028 | -0.0024 |
| admission_t0_emergency | TRUE | 40 | 0.8005 | 0.0063 | -0.0014 | 0.0015 |
| ecocardiograma | TRUE | 39 | 0.8010 | 0.0059 | -0.0019 | -0.0005 |
| hypertension | TRUE | 38 | 0.8000 | 0.0064 | -0.0008 | 0.0010 |
| af | TRUE | 37 | 0.7993 | 0.0061 | -0.0002 | 0.0006 |
| sex | TRUE | 36 | 0.8012 | 0.0060 | -0.0021 | -0.0019 |
| digoxina | TRUE | 35 | 0.8010 | 0.0059 | -0.0018 | 0.0003 |
| procedure_type_1 | TRUE | 34 | 0.7999 | 0.0063 | -0.0008 | 0.0010 |
| admission_pre_t0_180d | TRUE | 33 | 0.8020 | 0.0060 | -0.0029 | -0.0021 |
| cied_final_1 | TRUE | 32 | 0.8004 | 0.0059 | -0.0013 | 0.0016 |
| tomografia | TRUE | 31 | 0.8001 | 0.0062 | -0.0009 | 0.0003 |
| valvopathy | TRUE | 30 | 0.8019 | 0.0056 | -0.0028 | -0.0018 |
| cultura | TRUE | 29 | 0.8007 | 0.0056 | -0.0016 | 0.0012 |
| race | TRUE | 28 | 0.7994 | 0.0059 | -0.0003 | 0.0013 |
| renal_failure | TRUE | 27 | 0.7999 | 0.0061 | -0.0008 | -0.0005 |
| cied_final_group_1 | TRUE | 26 | 0.8000 | 0.0052 | -0.0009 | -0.0001 |
| estatina | TRUE | 25 | 0.8011 | 0.0054 | -0.0019 | -0.0010 |
| underlying_heart_disease | TRUE | 24 | 0.8008 | 0.0058 | -0.0016 | 0.0003 |
| antiarritmico | TRUE | 23 | 0.8006 | 0.0057 | -0.0014 | 0.0002 |
| exames_imagem_qtde | FALSE | 22 | 0.7982 | 0.0055 | -0.0014 | 0.0024 |
| metodos_graficos_qtde | TRUE | 22 | 0.8005 | 0.0051 | -0.0014 | 0.0001 |
| equipe_multiprof | FALSE | 21 | 0.7984 | 0.0054 | -0.0014 | 0.0021 |
| meds_antimicrobianos | TRUE | 21 | 0.8006 | 0.0051 | -0.0015 | -0.0001 |
| vasodilatador | FALSE | 20 | 0.7968 | 0.0051 | -0.0015 | 0.0038 |
| icu_t0 | FALSE | 20 | 0.7981 | 0.0053 | -0.0015 | 0.0025 |
| classe_meds_qtde | FALSE | 20 | 0.7978 | 0.0054 | -0.0015 | 0.0028 |
| insuf_cardiaca | TRUE | 20 | 0.7986 | 0.0056 | 0.0005 | 0.0020 |
| psicofarmacos | TRUE | 19 | 0.7975 | 0.0056 | 0.0017 | 0.0012 |
| diuretico | TRUE | 18 | 0.7973 | 0.0057 | 0.0019 | 0.0002 |
| nyha_basal | FALSE | 17 | 0.7914 | 0.0056 | 0.0019 | 0.0059 |
| laboratorio | TRUE | 17 | 0.8005 | 0.0055 | -0.0014 | -0.0033 |
| ieca_bra | TRUE | 16 | 0.7994 | 0.0056 | -0.0002 | 0.0012 |
| comorbidities_count | FALSE | 15 | 0.7961 | 0.0061 | -0.0002 | 0.0033 |
| education_level | FALSE | 15 | 0.7941 | 0.0064 | -0.0002 | 0.0053 |
| meds_cardiovasc_qtde | TRUE | 15 | 0.7991 | 0.0057 | 0.0000 | 0.0002 |
| espironolactona | FALSE | 14 | 0.7962 | 0.0052 | 0.0000 | 0.0029 |
| admission_pre_t0_count | FALSE | 14 | 0.7788 | 0.0056 | 0.0000 | 0.0203 |
| age | FALSE | 14 | 0.7941 | 0.0058 | 0.0000 | 0.0050 |
| year_adm_t0 | FALSE | 14 | 0.7813 | 0.0054 | 0.0000 | 0.0178 |
| hospital_stay | FALSE | 14 | 0.7891 | 0.0055 | 0.0000 | 0.0101 |

Minutes to run: 24.266

```
selected_features <- current_features

con <- file(sprintf('./auxiliar/final_model/selected_features/%s.yaml', outcome_column), "w")
write_yaml(selected_features, con)
close(con)

feature_selected_model <- model_fit_wf(df_train, selected_features,
                                       outcome_column, hyperparameters)
```

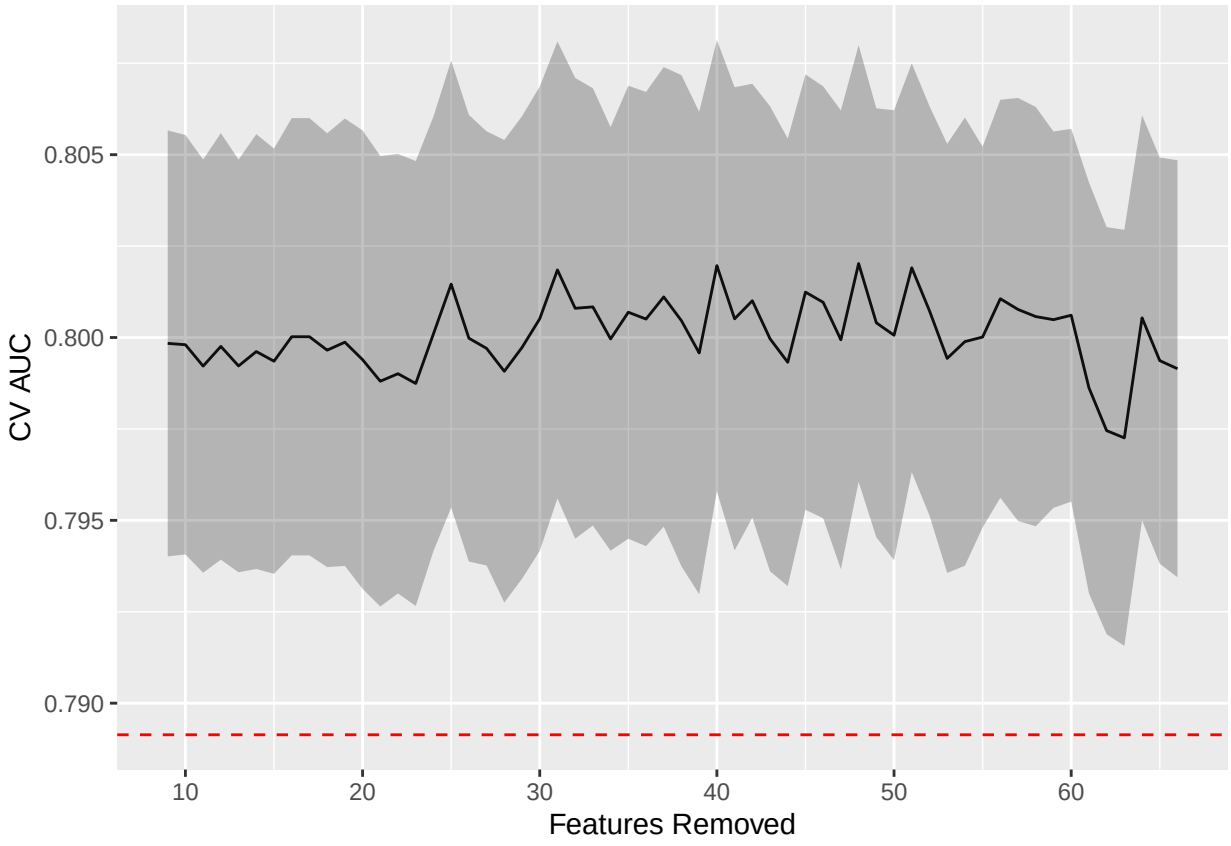
```
sprintf('Selected Model CV Train AUC: %.3f', feature_selected_model$cv_auc)
```

```
## [1] "Selected Model CV Train AUC: 0.799"
```

```
sprintf('Selected Model Test AUC: %.3f', feature_selected_model$auc)
```

```
## [1] "Selected Model Test AUC: 0.796"
```

```
 selection_results <- selection_results %>%
  filter(`Number of Features` < length(features)) %>%
  mutate(`Features Removed` = length(features) - `Number of Features`,
         `CV AUC Low` = `CV AUC` - `CV AUC Std Error`,
         `CV AUC High` = `CV AUC` + `CV AUC Std Error`)

selection_results %>%
  filter(Dropped) %>%
  ggplot(aes(x = `Features Removed`, y = `CV AUC`,
             ymin = `CV AUC Low`, ymax = `CV AUC High`)) +
  geom_line() +
  geom_ribbon(alpha = .3) +
  geom_hline(yintercept = full_model$cv_auc - max_auc_loss,
             linetype = "dashed", color = "red")
```




Minutes to run: 0.264

## Hyperparameter tuning

Selected features:

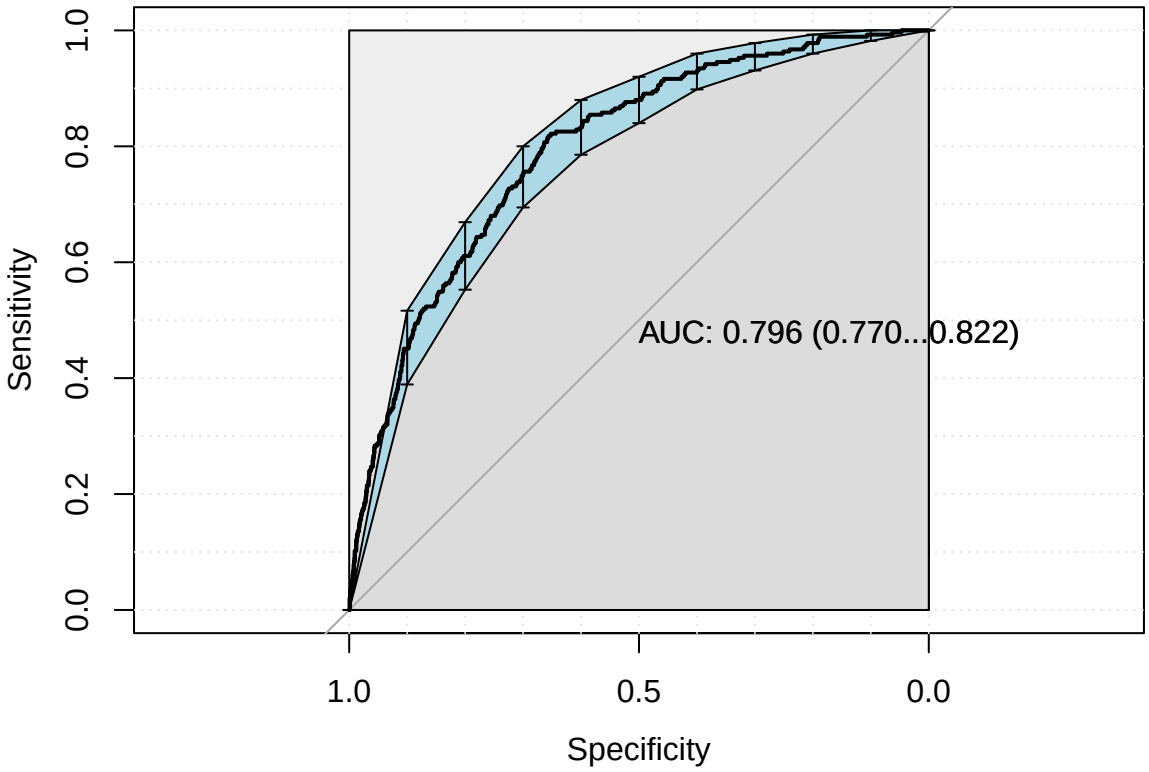
```
gluedown::md_order(selected_features, seq = TRUE, pad = TRUE)
```

1. hospital_stay
2. age
3. year_adm_t0
4. admission_pre_t0_count
5. espironolactona
6. education_level
7. comorbidities_count
8. nyha_basal
9. vasodilatador
10. classe_meds_qtde
11. icu_t0
12. exames_imagem_qtde
13. equipe_multiprof
14. ultrassom
15. heart_failure Minutes to run: 0

## Standard

```
lightgbm_recipe <-
  recipe(formula = sprintf("%s ~ .", outcome_column) %>% as.formula,
         data = df_train %>% select(all_of(c(selected_features, outcome_column)))) %>%
  step_novel(all_nominal_predictors()) %>%
  step_unknown(all_nominal_predictors()) %>%
  step_other(all_nominal_predictors(), threshold = 0.05, other = ".merged") %>%
  step_dummy(all_nominal_predictors())

lightgbm_tuning <- function(recipe) {

  lightgbm_spec <- boost_tree(
    trees = tune(),
    min_n = tune(),
    tree_depth = tune(),
    learn_rate = tune(),
    sample_size = 1.0
  ) %>%
    set_engine("lightgbm",
               nthread = 8) %>%
    set_mode("classification")

  lightgbm_grid <- grid_latin_hypercube(
    trees(range = c(25L, 150L)),
    min_n(range = c(2L, 100L)),
    tree_depth(range = c(5L, 15L)),
    learn_rate(range = c(-3, -1), trans = log10_trans()),
    size = grid_size
  )

  lightgbm_workflow <-
    workflow() %>%
    add_recipe(recipe) %>%
    add_model(lightgbm_spec)

  lightgbm_tune <-
    lightgbm_workflow %>%
    tune_grid(resamples = df_folds,
              grid = lightgbm_grid)
```

```
  lightgbm_tune %>%
    show_best("roc_auc") %>%
    niceFormatting(digits = 5, label = 4)

  best_lightgbm <- lightgbm_tune %>%
    select_best("roc_auc")

  autoplot(lightgbm_tune, metric = "roc_auc")

  final_lightgbm_workflow <-
    lightgbm_workflow %>%
    finalize_workflow(best_lightgbm)

  last_lightgbm_fit <-
    final_lightgbm_workflow %>%
    last_fit(df_split)

  final_lightgbm_fit <- extract_workflow(last_lightgbm_fit)

  lightgbm_auc <- validation(final_lightgbm_fit, df_test)

  lightgbm_parameters <- lightgbm_tune %>%
    show_best("roc_auc", n = 1) %>%
    select(trees, min_n, tree_depth, learn_rate) %>%
    as.list

  return(list(auc = as.numeric(lightgbm_auc$auc),
              auc_lower = lightgbm_auc$ci[1],
              auc_upper = lightgbm_auc$ci[3],
              parameters = lightgbm_parameters,
              fit = final_lightgbm_fit))
}

standard_results <- lightgbm_tuning(lightgbm_recipe)
```




```
##   |
```

```
final_lightgbm_fit <- standard_results$fit
lightgbm_parameters <- standard_results$parameters
```

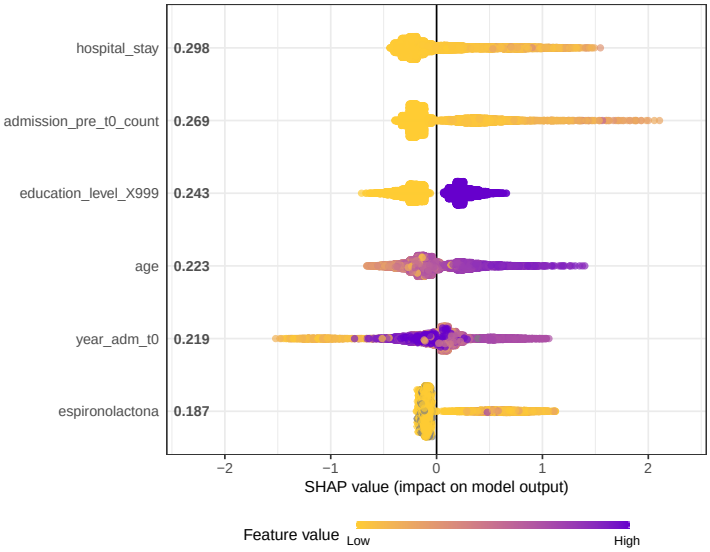
```
saveRDS(
  lightgbm_parameters,
  file = sprintf(
    "./auxiliar/final_model/hyperparameters/lightgbm_%s.rds",
    outcome_column
  )
)

# Save the final model. We need it for the calculator
lgb.save(
  parsnip::extract_fit_engine(final_lightgbm_fit),
  sprintf("./results/%s/final_model.txt", outcome_column)
)
saveRDS(final_lightgbm_fit,
        sprintf("./results/%s/final_model_wf.rds", outcome_column))
```

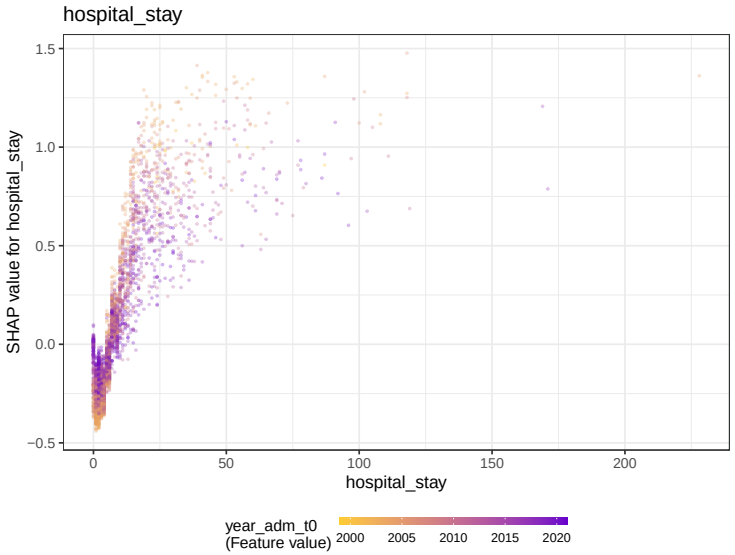
Minutes to run: 7.725

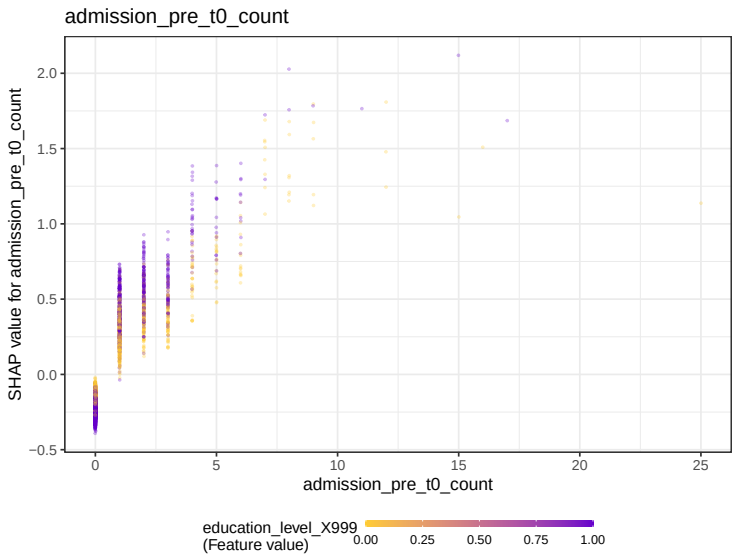
## SHAP values

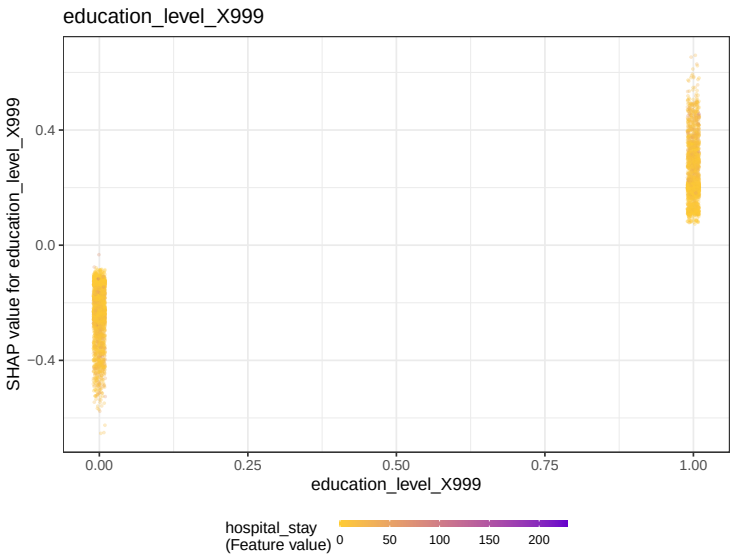
```
lightgbm_model <- parsnip::extract_fit_engine(final_lightgbm_fit)

trained_rec <- prep(lightgbm_recipe, training = df_train)

df_train_baked <- bake(trained_rec, new_data = df_train)
df_test_baked <- bake(trained_rec, new_data = df_test)

matrix_train <- as.matrix(df_train_baked %>% select(-all_of(outcome_column)))
matrix_test <- as.matrix(df_test_baked %>% select(-all_of(outcome_column)))

n_plots <- min(6, length(selected_features))
plotted <- 0

shap.plot.summary.wrap1(model = lightgbm_model, X = matrix_train,
                        top_n = n_plots, dilute = F)

shap <- shap.prep(lightgbm_model, X_train = matrix_test)

for (x in shap.importance(shap, names_only = TRUE)) {
  p <- shap.plot.dependence(
    shap,
    x = x,
    color_feature = "auto",
    smooth = FALSE,
    jitter_width = 0.01,
    alpha = 0.3
  ) +
    labs(title = x)

  if (plotted < n_plots) {
    print(p)
    plotted <- plotted + 1
  }

  ggsave(sprintf("./auxiliar/final_model/shap_plots/%s.png", x),
         plot = p,
         dpi = 300)
}
```

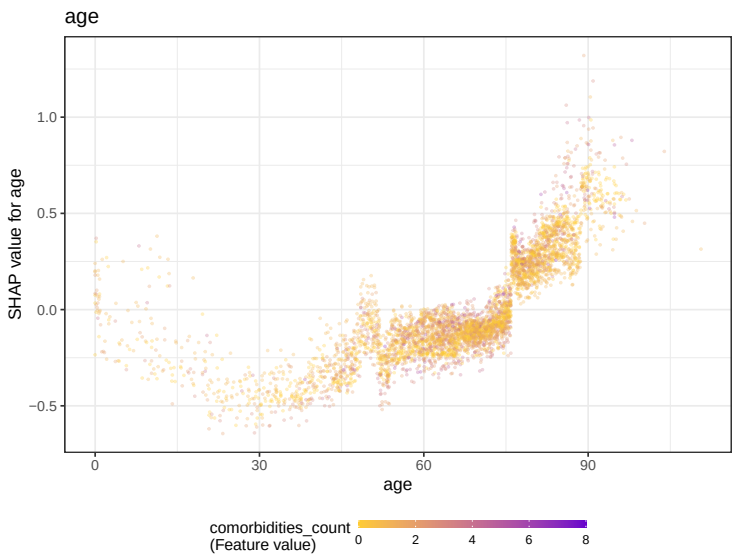
```
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image

## Warning: Removed 6 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image

## Warning: Removed 6 rows containing missing values (geom_point).

## Warning: Removed 1070 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image

## Warning: Removed 1070 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image

## Warning: Removed 1484 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image

## Warning: Removed 1070 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image

## Warning: Removed 835 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image

## Warning: Removed 835 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image

## Warning: Removed 835 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
```

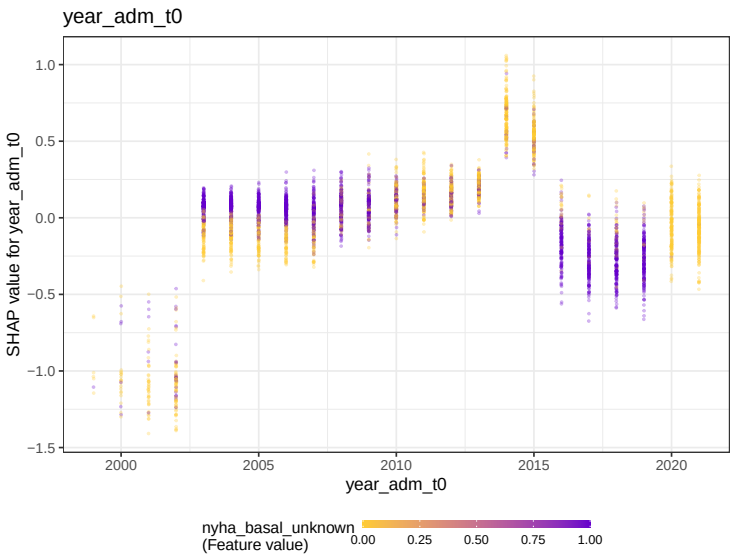

hospital_stay 0.298
admission_pre_t0_count 0.269
education_level_X999 0.243
age 0.223
year_adm_t0 0.219
espironolactona 0.187
SHAP value (impact on model output)
Feature value
Low
High

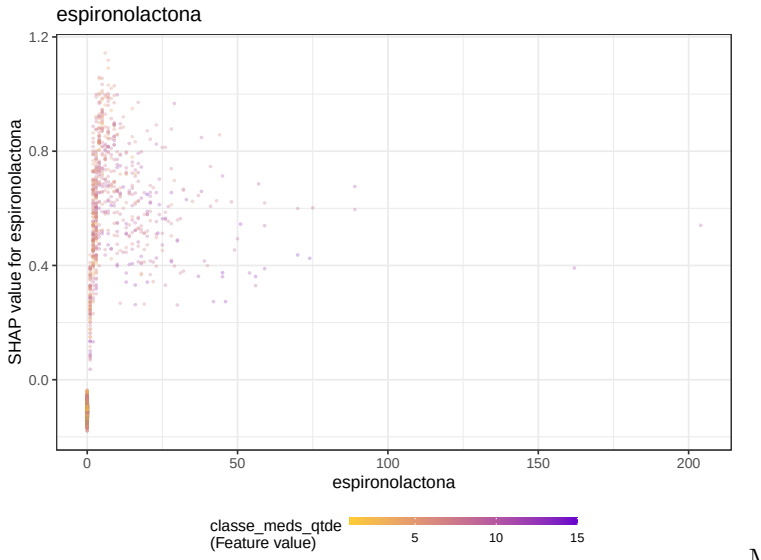

hospital_stay
SHAP value for hospital_stay
hospital_stay
year_adm_t0
(Feature value)


admission_pre_t0_count
SHAP value for admission_pre_t0_count
admission_pre_t0_count
education_level_X999
(Feature value)


education_level_X999
SHAP value for education_level_X999
education_level_X999
hospital_stay
(Feature value)


age
SHAP value for age
age
comorbidities_count
(Feature value)


year_adm_t0
SHAP value for year_adm_t0
year_adm_t0
nyha_basal_unknown
(Feature value)




Minutes to run: 0.255

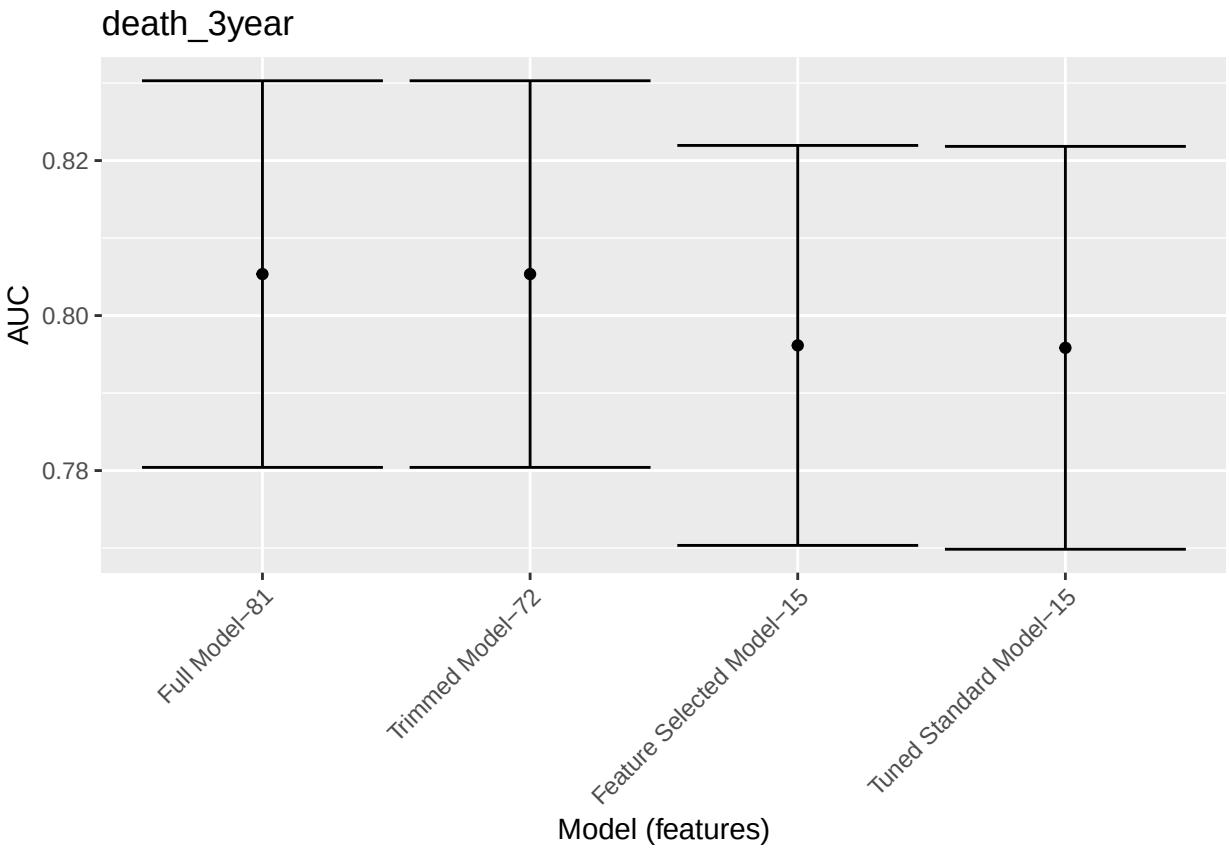
```
## $num_iterations
## [1] 98
##
## $learning_rate
## [1] 0.03495387
##
## $max_depth
## [1] 6
##
## $feature_fraction_bynode
## [1] 1
##
## $min_data_in_leaf
## [1] 94
##
## $min_gain_to_split
## [1] 0
##
## $bagging_fraction
## [1] 1
##
## $num_class
## [1] 1
##
## $objective
## [1] "binary"
##
## $num_threads
## $num_threads$num_threads
## [1] 0
##
##
## $nthread
## [1] 8
##
## $seed
## [1] 90694
##
## $deterministic
## [1] TRUE
##
## $verbose
```

```
## [1] -1
##
## $metric
## list()
##
## $interaction_constraints
## list()
##
## $feature_pre_filter
## [1] FALSE
```

Minutes to run: 0

## Models Comparison

```
df_auc <- tibble::tribble(
  ~Model, ~`AUC`, ~`Lower Limit`, ~`Upper Limit`, ~`Features`,
  'Full Model', full_model$auc, full_model$auc_lower, full_model$auc_upper, length(features),
  'Trimmed Model', trimmed_model$auc, trimmed_model$auc_lower, trimmed_model$auc_upper, length(trimmed_features)
  'Feature Selected Model', feature_selected_model$auc, feature_selected_model$auc_lower, feature_selected_model
  'Tuned Standard Model', standard_results$auc, standard_results$auc_lower, standard_results$auc_upper, length(s
) %>%
  mutate(Target = outcome_column,
         `Model (features)` = fct_reorder(paste0(Model, "-", Features), -Features))

df_auc %>%
  ggplot(aes(
    x = `Model (features)`,
    y = AUC,
    ymin = `Lower Limit`,
    ymax = `Upper Limit`
  )) +
  geom_point() +
  geom_errorbar() +
  labs(title = outcome_column) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
saveRDS(df_auc, sprintf("./auxiliar/final_model/performance/%s.RData", outcome_column))
```

Minutes to run: 0.002