# Final Model - readmission_1year

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
```

## 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
```

## 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))
```

```
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)
```

## 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)
```

```
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))

if (is.null(features_list)) {
  features = eligible_features
} else {
  features = base::intersect(eligible_features, features_list)
}
```

Starting features:

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

1. sex
2. age
3. race
4. education_level
5. patient_state
6. underlying_heart_disease
7. heart_disease
8. nyha_basal
9. prior_mi
10. heart_failure
11. af

12. cardiac_arrest
13. transplant
14. valvopathy
15. endocardites
16. diabetes
17. renal_failure
18. hemodialysis
19. copd
20. comorbidities_count
21. procedure_type_1
22. reop_type_1
23. procedure_type_new
24. cied_final_1
25. cied_final_group_1
26. admission_pre_t0_count
27. admission_pre_t0_180d
28. year_adm_t0
29. icu_t0
30. dialysis_t0
31. admission_t0_emergency
32. aco
33. antiarritmico
34. betabloqueador
35. ieca_bra
36. dva
37. digoxina
38. estatina
39. diuretico
40. vasodilatador
41. insuf_cardiaca
42. espironolactona
43. bloq_calcio
44. antiplaquetario_ev
45. insulina
46. anticonvulsivante
47. psicofarmacos
48. antifungico
49. antiviral
50. antiretroviral
51. classe_meds_qtde
52. meds_cardiovasc_qtde
53. meds_antimicrobianos
54. ventilacao_mecanica
55. cec
56. transplante_cardiaco
57. cir_toracica
58. outros_proced_cirurgicos
59. icp
60. intervencao_cv
61. angioplastia
62. cateterismo
63. eletrofisiologia
64. cateter_venoso_central
65. proced_invasivos_qtde
66. cve_desf
67. transfusao
68. interconsulta
69. equipe_multiprof
70. holter
71. teste_esforco
72. espiro_ergoespiro

73. tilt_teste
74. metodos_graficos_qtde
75. laboratorio
76. cultura
77. analises_clinicas_qtde
78. citologia
79. biopsia
80. histopatologia_qtde
81. angio_rm
82. angio_tc
83. aortografia
84. arteriografia
85. cintilografia
86. ecocardiograma
87. endoscopia
88. flebografia
89. pet_ct
90. ultrassom
91. tomografia
92. ressonancia
93. exames_imagem_qtde
94. dieta_parenteral
95. bic
96. mpp
97. hospital_stay

## 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)
```

## 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]
    )
  )
}
```

```
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: 0.722"
```

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

```
## [1] "Full Model Test AUC: 0.718"
```

```
# full_model$importance %>%
#   filter(str_detect(Feature, 'education'))
#
# full_model$importance %>%
#   filter(str_detect(Feature, 'education')) %>%
#   summarise(across(where(is.numeric), ~ sum(.x, na.rm = TRUE)))
#
# full_model$importance %>%
#   separate(Feature, c("Feature","value"), "__") %>%
#   group_by(Feature) %>%
#   summarise(Gain = sum(Gain),
#             Cover = sum(Cover),
#             Frequency = sum(Frequency))
```

Features with zero importance on the initial model:

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

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

1. af
2. cardiac_arrest
3. transplant
4. valvopathy
5. diabetes
6. copd
7. dialysis_t0
8. aco
9. betabloqueador
10. antiplaquetario_ev
11. anticonvulsivante
12. antifungico
13. antiviral
14. antiretroviral
15. ventilacao_mecanica
16. cec
17. transplante_cardiaco
18. cir_toracica
19. icp
20. intervencao_cv
21. angioplastia
22. cateterismo
23. eletrofisiologia
24. cve_desf
25. transfusao
26. teste_esforco
27. espiro_ergoespiro
28. tilt_teste
29. laboratorio
30. cultura
31. citologia
32. angio_rm
33. angio_tc
34. aortografia
35. arteriografia
36. cintilografia
37. endoscopia
38. pet_ct
39. tomografia
40. ressonancia
41. dieta_parenteral

42. bic
43. mpp

```
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: 0.723"
```

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

```
## [1] "Trimmed Model Test AUC: 0.718"
```

```
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] "53"          "-7e-04"      "0"           "reop_type_1"
## [1] "52"     "-9e-04" "-2e-04" "sex"
## [1] "51"       "-7e-04"    "2e-04"    "prior_mi"
## [1] "50"       "-0.001"    "-4e-04"    "insulina"
## [1] "49"                      "-9e-04"                  "2e-04"                   "cateter_venoso_central"
## [1] "48"             "-7e-04"          "2e-04"           "insuf_cardiaca"
## [1] "47"            "-7e-04"         "0"              "renal_failure"
## [1] "46"     "-8e-04" "-1e-04" "holter"
## [1] "45"                     "-9e-04"                  "-1e-04"                 "proced_invasivos_qtde"
## [1] "44"            "-8e-04"         "1e-04"          "interconsulta"
## [1] "43"       "-9e-04"    "-2e-04"    "estatina"
## [1] "42"      "-6e-04"  "3e-04"    "biopsia"
## [1] "41"                        "-9e-04"                    "-3e-04"
## [4] "outros_proced_cirurgicos"
## [1] "40"            "-9e-04"        "0"             "hemodialysis"
## [1] "39"           "-0.0011"      "-3e-04"       "flebografia"
## [1] "38"                     "-0.0013"                 "-2e-04"                  "analises_clinicas_qtde"
## [1] "37"                  "-0.0011"            "2e-04"              "exames_imagem_qtde"
## [1] "36"             "-0.0014"        "-3e-04"         "heart_disease"
## [1] "35"              "-0.0015"         "-1e-04"          "ecocardiograma"
## [1] "34"       "-0.0013" "2e-04"    "race"
## [1] "33"            "-0.0012"       "1e-04"         "endocardites"
## [1] "32"         "-9e-04"     "3e-04"      "ultrassom"
## [1] "31"                        "-0.0011"                   "-2e-04"
## [4] "underlying_heart_disease"
## [1] "30"     "-0.001" "1e-04"  "dva"
## [1] "29"                      "-0.0014"                 "-4e-04"                  "admission_t0_emergency"
## [1] "28"             "-0.0012"        "2e-04"          "heart_failure"
## [1] "27"                   "-3e-04"               "9e-04"                "comorbidities_count"
## [1] "26"             "-3e-04"         "0"              "psicofarmacos"
## [1] "25"        "-6e-04"    "-3e-04"    "digoxina"
## [1] "24"                "-4e-04"            "2e-04"             "procedure_type_1"
## [1] "23"          "2e-04"       "6e-04"       "nyha_basal"
## [1] "22"        "-4e-04"    "-6e-04"    "ieca_bra"
## [1] "21"                    "-1e-04"               "3e-04"                "histopatologia_qtde"
## [1] "20"                "-4e-04"            "-3e-04"            "equipe_multiprof"
## [1] "19"         "0"          "4e-04"      "diuretico"
```

```
## [1] "18"          "-2e-04"      "-2e-04"      "bloq_calcio"
## [1] "17"           "7e-04"        "9e-04"        "cied_final_1"
## [1] "16"                    "0.0015"               "8e-04"                 "metodos_graficos_qtde"
## [1] "15"            "0.0032"        "0.0017"        "patient_state"
## [1] "14"              "0.0042"          "0.001"           "espironolactona"
## [1] "13"                 "0.0044"             "2e-04"             "cied_final_group_1"
## [1] "12"                   "0.0052"               "7e-04"               "meds_antimicrobianos"
## [1] "11"     "0.0058" "7e-04"  "icu_t0"
## [1] "10"              "0.0056"          "-2e-04"          "education_level"
## [1] "9"             "0.0065"        "9e-04"         "vasodilatador"
## [1] "8"                    "0.0063"                "-2e-04"                "admission_pre_t0_180d"
## [1] "8"                  "0.0063"             "0.0033"             "procedure_type_new"
## [1] "7"      "0.006"  "-3e-04" "age"
## [1] "7"             "0.006"         "0.0041"        "antiarritmico"
## [1] "6"                "0.0066"           "6e-04"            "classe_meds_qtde"
## [1] "5"                    "0.0074"               "8e-04"                "meds_cardiovasc_qtde"
## [1] "5"           "0.0074"      "0.0084"      "year_adm_t0"
## [1] "5"                     "0.0074"                 "0.0263"                 "admission_pre_t0_count"
## [1] "5"             "0.0074"        "0.0256"        "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 | 97 | 0.7222 | 0.0039 | 0.0000 | 0.0000 |
| All unimportant | TRUE | 54 | 0.7229 | 0.0037 | -0.0007 | -0.0007 |
| reop_type_1 | TRUE | 53 | 0.7229 | 0.0037 | -0.0007 | 0.0000 |
| sex | TRUE | 52 | 0.7231 | 0.0038 | -0.0009 | -0.0002 |
| prior_mi | TRUE | 51 | 0.7229 | 0.0038 | -0.0007 | 0.0002 |
| insulina | TRUE | 50 | 0.7233 | 0.0038 | -0.0010 | -0.0004 |
| cateter_venoso_central | TRUE | 49 | 0.7231 | 0.0037 | -0.0009 | 0.0002 |
| insuf_cardiaca | TRUE | 48 | 0.7229 | 0.0037 | -0.0007 | 0.0002 |
| renal_failure | TRUE | 47 | 0.7229 | 0.0037 | -0.0007 | 0.0000 |
| holter | TRUE | 46 | 0.7230 | 0.0037 | -0.0008 | -0.0001 |
| proced_invasivos_qtde | TRUE | 45 | 0.7231 | 0.0037 | -0.0009 | -0.0001 |
| interconsulta | TRUE | 44 | 0.7230 | 0.0037 | -0.0008 | 0.0001 |
| estatina | TRUE | 43 | 0.7232 | 0.0037 | -0.0009 | -0.0002 |
| biopsia | TRUE | 42 | 0.7228 | 0.0036 | -0.0006 | 0.0003 |
| outros_proced_cirurgicos | TRUE | 41 | 0.7231 | 0.0037 | -0.0009 | -0.0003 |
| hemodialysis | TRUE | 40 | 0.7231 | 0.0037 | -0.0009 | 0.0000 |
| flebografia | TRUE | 39 | 0.7233 | 0.0037 | -0.0011 | -0.0003 |
| analises_clinicas_qtde | TRUE | 38 | 0.7235 | 0.0037 | -0.0013 | -0.0002 |
| exames_imagem_qtde | TRUE | 37 | 0.7233 | 0.0036 | -0.0011 | 0.0002 |
| heart_disease | TRUE | 36 | 0.7236 | 0.0037 | -0.0014 | -0.0003 |
| ecocardiograma | TRUE | 35 | 0.7237 | 0.0037 | -0.0015 | -0.0001 |
| race | TRUE | 34 | 0.7235 | 0.0036 | -0.0013 | 0.0002 |
| endocardites | TRUE | 33 | 0.7234 | 0.0037 | -0.0012 | 0.0001 |
| ultrassom | TRUE | 32 | 0.7231 | 0.0037 | -0.0009 | 0.0003 |
| underlying_heart_disease | TRUE | 31 | 0.7234 | 0.0036 | -0.0011 | -0.0002 |
| dva | TRUE | 30 | 0.7232 | 0.0036 | -0.0010 | 0.0001 |
| admission_t0_emergency | TRUE | 29 | 0.7236 | 0.0037 | -0.0014 | -0.0004 |
| heart_failure | TRUE | 28 | 0.7234 | 0.0036 | -0.0012 | 0.0002 |
| comorbidities_count | TRUE | 27 | 0.7225 | 0.0037 | -0.0003 | 0.0009 |
| psicofarmacos | TRUE | 26 | 0.7225 | 0.0037 | -0.0003 | 0.0000 |
| digoxina | TRUE | 25 | 0.7228 | 0.0038 | -0.0006 | -0.0003 |
| procedure_type_1 | TRUE | 24 | 0.7226 | 0.0037 | -0.0004 | 0.0002 |

Table 1: *(continued)*

| Tested Feature | Dropped | Features | CV AUC | CV AUC Std Error | Total AUC Loss | Instant AUC Loss |
|---|---|---|---|---|---|---|
| nyha_basal | TRUE | 23 | 0.7220 | 0.0036 | 0.0002 | 0.0006 |
| ieca_bra | TRUE | 22 | 0.7226 | 0.0037 | -0.0004 | -0.0006 |
| histopatologia_qtde | TRUE | 21 | 0.7224 | 0.0037 | -0.0001 | 0.0003 |
| equipe_multiprof | TRUE | 20 | 0.7226 | 0.0037 | -0.0004 | -0.0003 |
| diuretico | TRUE | 19 | 0.7222 | 0.0035 | 0.0000 | 0.0004 |
| bloq_calcio | TRUE | 18 | 0.7224 | 0.0036 | -0.0002 | -0.0002 |
| cied_final_1 | TRUE | 17 | 0.7215 | 0.0037 | 0.0007 | 0.0009 |
| metodos_graficos_qtde | TRUE | 16 | 0.7207 | 0.0034 | 0.0015 | 0.0008 |
| patient_state | TRUE | 15 | 0.7190 | 0.0035 | 0.0032 | 0.0017 |
| espironolactona | TRUE | 14 | 0.7180 | 0.0034 | 0.0042 | 0.0010 |
| cied_final_group_1 | TRUE | 13 | 0.7178 | 0.0032 | 0.0044 | 0.0002 |
| meds_antimicrobianos | TRUE | 12 | 0.7170 | 0.0035 | 0.0052 | 0.0007 |
| icu_t0 | TRUE | 11 | 0.7164 | 0.0035 | 0.0058 | 0.0007 |
| education_level | TRUE | 10 | 0.7166 | 0.0035 | 0.0056 | -0.0002 |
| vasodilatador | TRUE | 9 | 0.7157 | 0.0035 | 0.0065 | 0.0009 |
| admission_pre_t0_180d | TRUE | 8 | 0.7159 | 0.0034 | 0.0063 | -0.0002 |
| procedure_type_new | FALSE | 7 | 0.7126 | 0.0026 | 0.0063 | 0.0033 |
| age | TRUE | 7 | 0.7162 | 0.0033 | 0.0060 | -0.0003 |
| antiarritmico | FALSE | 6 | 0.7121 | 0.0028 | 0.0060 | 0.0041 |
| classe_meds_qtde | TRUE | 6 | 0.7156 | 0.0034 | 0.0066 | 0.0006 |
| meds_cardiovasc_qtde | TRUE | 5 | 0.7148 | 0.0040 | 0.0074 | 0.0008 |
| year_adm_t0 | FALSE | 4 | 0.7063 | 0.0045 | 0.0074 | 0.0084 |
| admission_pre_t0_count | FALSE | 4 | 0.6885 | 0.0034 | 0.0074 | 0.0263 |
| hospital_stay | FALSE | 4 | 0.6892 | 0.0062 | 0.0074 | 0.0256 |

```
if (exists('last_feature_dropped')) {
  selected_features <- c(current_features, last_feature_droped)
} else {
  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: 0.715"
```

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

```
## [1] "Selected Model Test AUC: 0.705"
```

```
 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")
```


# Hyperparameter tuning

Selected features:

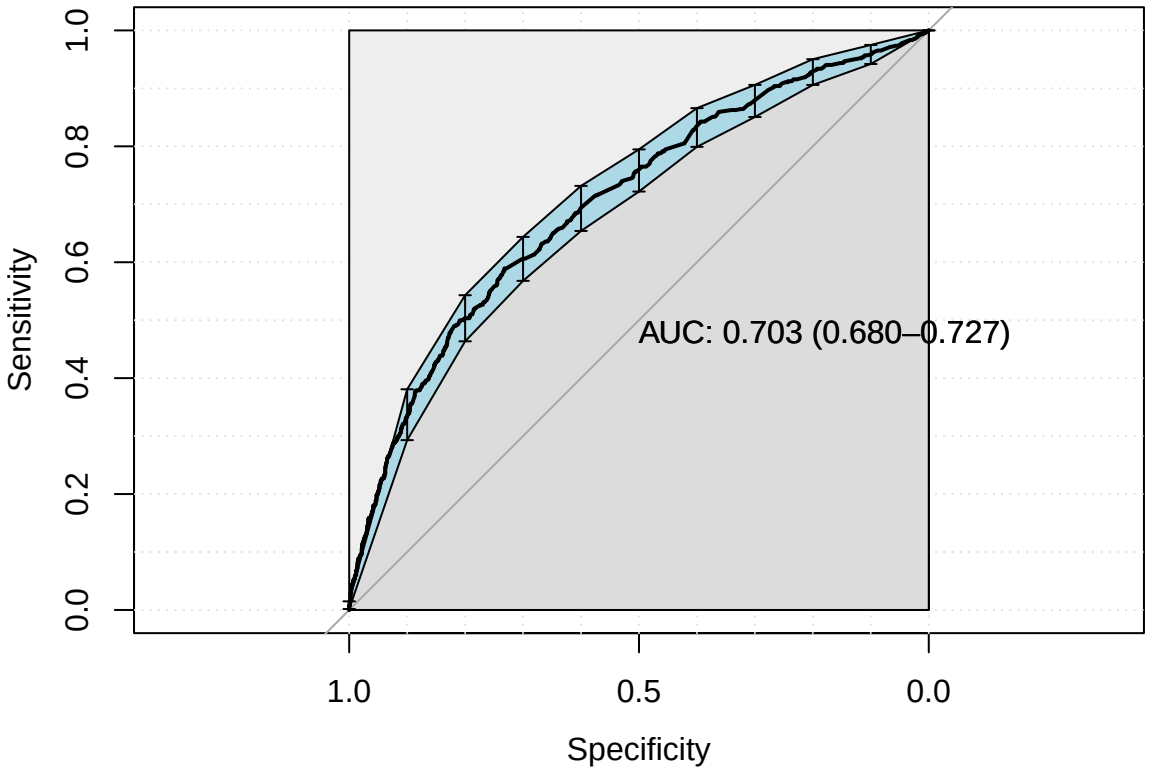
```
gluedown::md_order(selected_features, seq = TRUE, pad = TRUE)
```

1. hospital_stay
2. admission_pre_t0_count
3. year_adm_t0
4. antiarritmico
5. procedure_type_new

## 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(),
    # loss_reduction = tune(),
    sample_size = 1.0
  ) %>%
    set_engine("lightgbm") %>%
    set_mode("classification")

  lightgbm_grid <- grid_latin_hypercube(
    trees(range = c(50L, 300L)),
    min_n(),
    tree_depth(),
    learn_rate(range = c(0.01, 0.2), trans = NULL),
    # loss_reduction(),
    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)
```



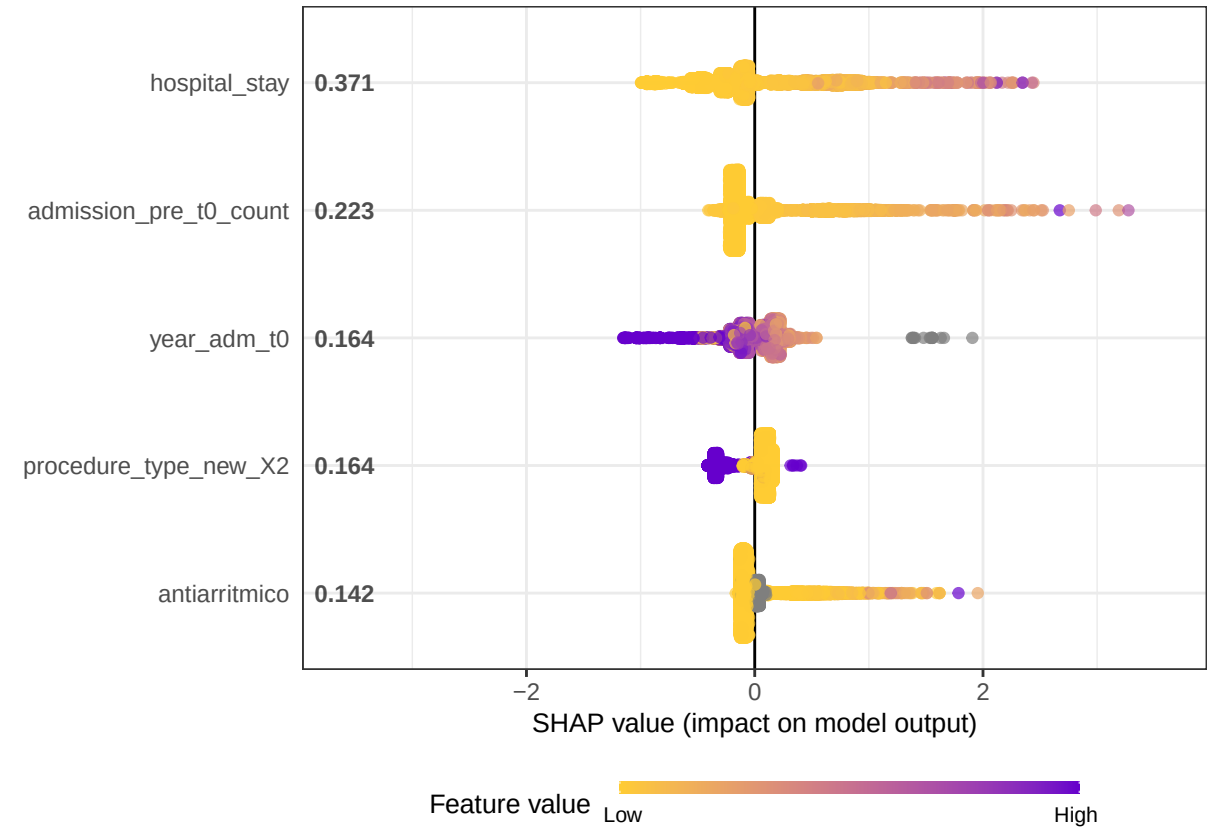

```
## [1] "Optimal Threshold: 0.12"
## Confusion Matrix and Statistics
##
##     reference
## data    0    1
##    0 3022  248
##    1 1105  356
##
##                Accuracy : 0.714
##                  95% CI : (0.7009, 0.7269)
##     No Information Rate : 0.8723
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2003
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7323
##             Specificity : 0.5894
##          Pos Pred Value : 0.9242
##          Neg Pred Value : 0.2437
##              Prevalence : 0.8723
##          Detection Rate : 0.6388
##    Detection Prevalence : 0.6912
##       Balanced Accuracy : 0.6608
##
##        'Positive' Class : 0
##
```

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

saveRDS(
  lightgbm_parameters,
```

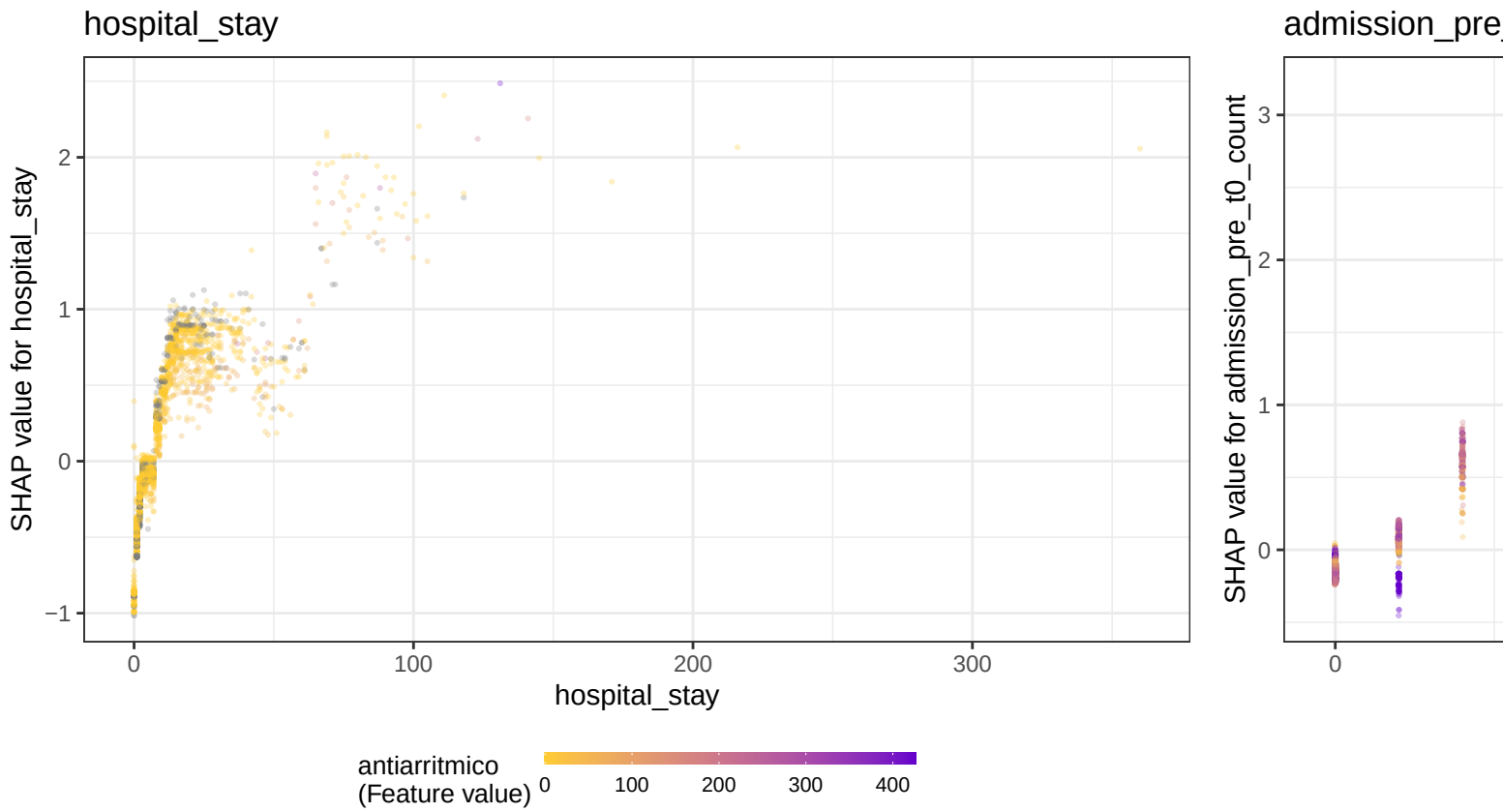
```
  file = sprintf(
    "./auxiliar/final_model/hyperparameters/lightgbm_%s.rds",
    outcome_column
  )
)

# 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))
```

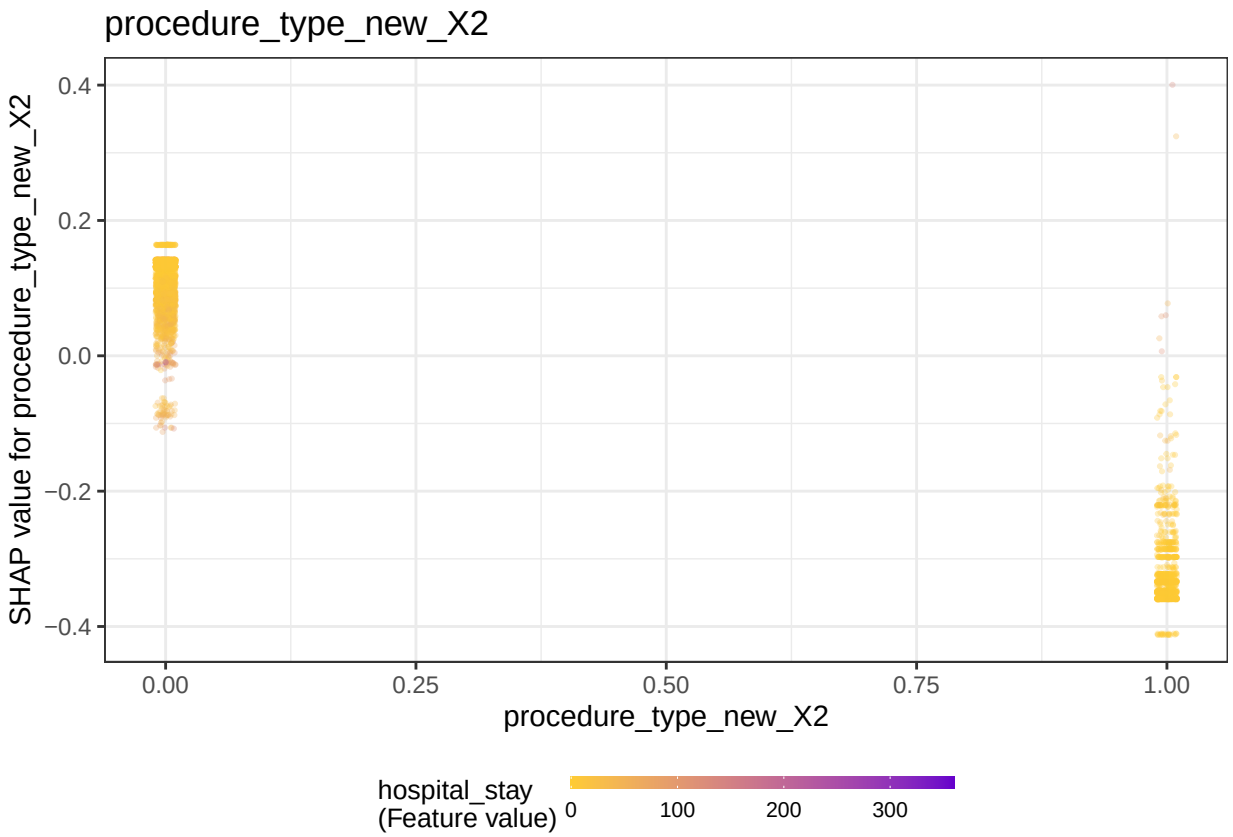
## SHAP values

```
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(5, length(selected_features))

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

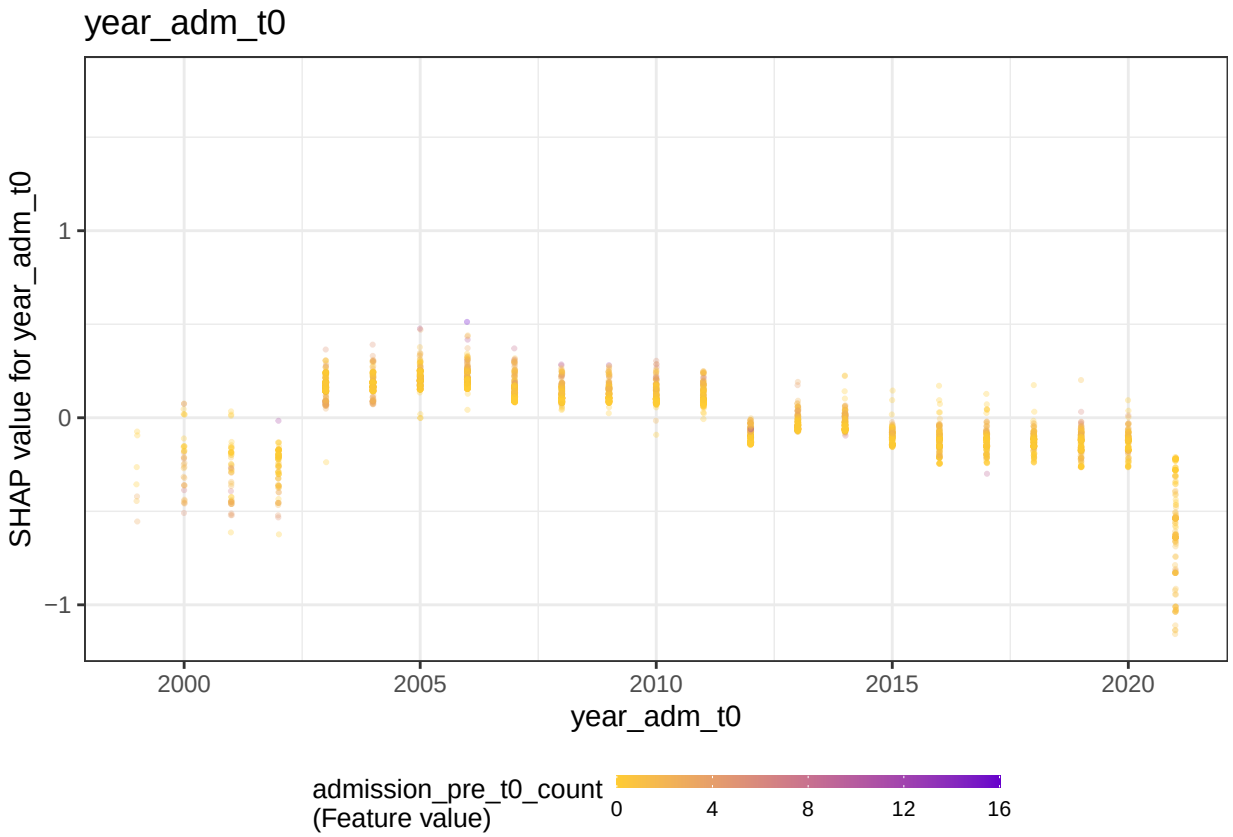


```r
shap <- shap.prep(lightgbm_model, X_train = matrix_test)

for (x in shap.importance(shap, names_only = TRUE)[1:n_plots]) {
  p <- shap.plot.dependence(
    shap,
    x = x,
    color_feature = "auto",
    smooth = FALSE,
    jitter_width = 0.01,
    alpha = 0.3
  ) +
    labs(title = x)
  print(p)
}
```

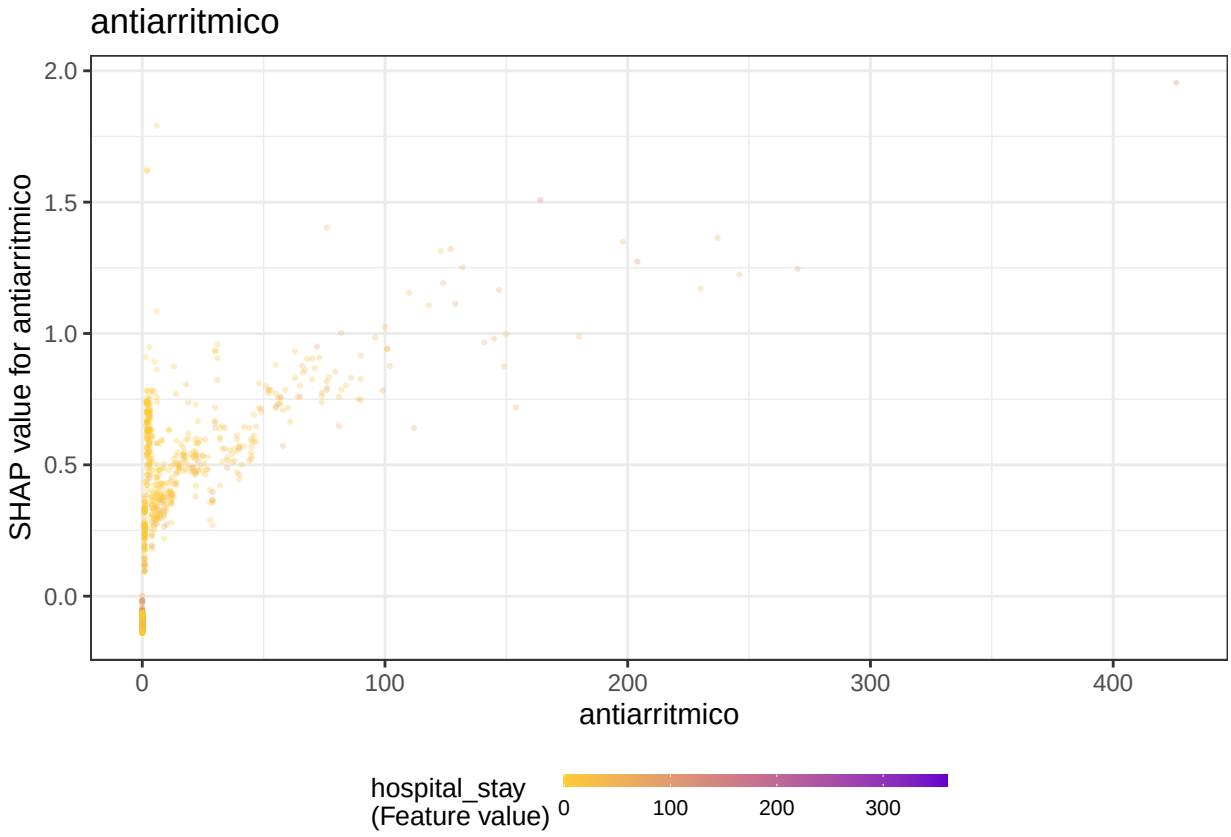





```
## Warning: Removed 5 rows containing missing values (geom_point).
```




```
## Warning: Removed 1050 rows containing missing values (geom_point).
```

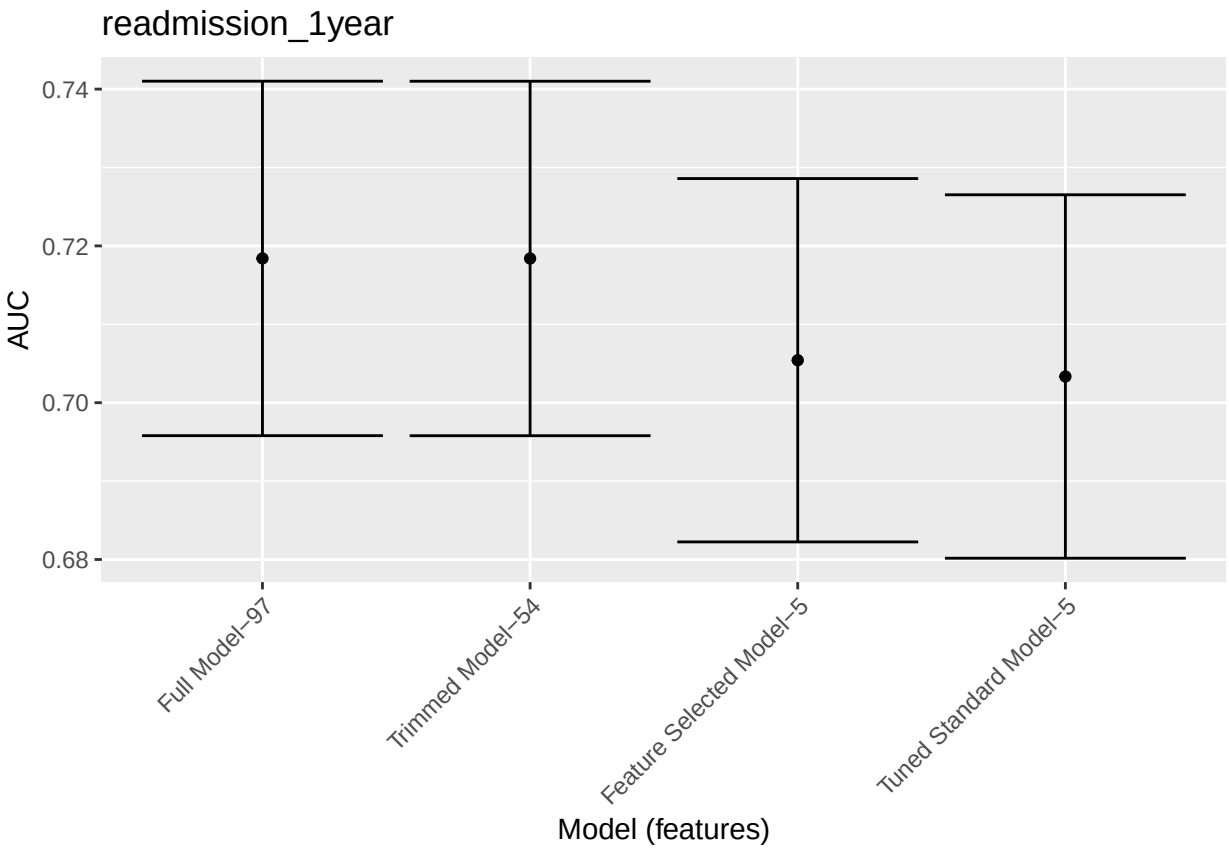



```
## $num_iterations
## [1] 75
##
## $learning_rate
## [1] 0.1087132
##
## $max_depth
## [1] 3
##
## $feature_fraction
## [1] 1
##
## $min_data_in_leaf
## [1] 7
##
## $min_gain_to_split
## [1] 0
##
## $bagging_fraction
## [1] 1
##
## $num_class
## [1] 1
##
## $objective
## [1] "binary"
##
## $num_threads
## $num_threads$num_threads
## [1] 0
##
##
## $seed
```

```
## [1] 11224
##
## $deterministic
## [1] TRUE
##
## $verbose
## [1] -1
##
## $metric
## list()
##
## $interaction_constraints
## list()
##
## $feature_pre_filter
## [1] FALSE
```

## 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))
```