Final Model - readmission_180d

Eduardo Yuki Yada

Global parameters

```
k <- 5 # Number of folds for cross validation
grid_size <- 30 # Number of parameter combination to tune on each model
max_auc_loss <- 0.01 # Max accepted loss of AUC for reducing num of features
```

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. education_level
4. patient_state
5. underlying_heart_disease
6. heart_disease
7. nyha_basal
8. prior_mi
9. heart_failure
10. af
11. cardiac_arrest
12. transplant

13. valvopathy
14. endocardites
15. diabetes
16. renal_failure
17. hemodialysis
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. icu_t0
28. dialysis_t0
29. n_procedure_t0
30. admission_t0_emergency
31. aco
32. antiarritmico
33. betabloqueador
34. ieca_bra
35. dva
36. digoxina
37. estatina
38. diuretico
39. vasodilatador
40. insuf_cardiaca
41. espironolactona
42. bloq_calcio
43. antiplaquetario_ev
44. insulina
45. anticonvulsivante
46. psicofarmacos
47. antifungico
48. antiviral
49. antiretroviral
50. classe_meds_qtde
51. meds_cardiovasc_qtde
52. meds_antimicrobianos
53. vni
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. arteriografia
84. cintilografia
85. ecocardiograma
86. endoscopia
87. flebografia
88. pet_ct
89. ultrassom
90. tomografia
91. ressonancia
92. exames_imagem_qtde
93. bic
94. mpp
95. 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))
```

Feature Selection

```
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")

  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)

  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: 0.720"
```

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

```
## [1] "Full Model Test AUC: 0.700"
```

Features with zero importance on the initial model:

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

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

1. hemodialysis
2. antiretroviral
3. transplante_cardiaco
4. cir_toracica
5. angioplastia
6. transfusao
7. tilt_teste
8. angio_rm
9. arteriografia
10. pet_ct

```
trimmed_features <- full_model$importance$Feature
hyperparameters$mtry <- min(hyperparameters$mtry, length(trimmed_features))
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: 0.719"
```

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

```
## [1] "Trimmed Model Test AUC: 0.701"
```

```
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) {
  current_least_important <-
    tail(setdiff(current_model$importance$Feature, whitelist), 1)
  test_features <-
    setdiff(current_features, current_least_important)
  hyperparameters$mtry <-
    min(hyperparameters$mtry, length(test_features))
  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
  current_auc_loss <- full_model$cv_auc - current_model$cv_auc

  if (instant_auc_loss < max_auc_loss / 5 &
      current_auc_loss < max_auc_loss) {
    dropped <- TRUE
    current_features <- test_features
  } 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] "84"      "-0.0015" "-0.003"  "cec"
## [1] "83"        "0"         "0.0015"    "citologia"
## [1] "82"            "-7e-04"        "-7e-04"        "renal_failure"
## [1] "81"           "-0.0034"      "-0.0027"      "endocardites"
## [1] "81"            "-3e-04"        "0.0031"        "teste_esforco"
## [1] "81"          "-7e-04"      "0.0027"      "dialysis_t0"
## [1] "81"            "-3e-04"        "0.0031"        "heart_disease"
## [1] "80"         "-0.0018"    "0.0016"     "transplant"
## [1] "79"         "0"          "0.0018"     "valvopathy"
## [1] "78"             "-0.0018"        "-0.0018"        "intervencao_cv"
## [1] "77"       "-6e-04"   "0.0012"   "angio_tc"
## [1] "76"      "-0.0024" "-0.0018" "vni"
## [1] "75"          "-0.0017"     "7e-04"       "antifungico"
## [1] "74"       "-0.004"   "-0.0023"  "cve_desf"
## [1] "74"                "-9e-04"           "0.003"             "espiro_ergoespiro"
## [1] "74"       "-0.0014"  "0.0025"   "prior_mi"
## [1] "74"       "-2e-04"   "0.0038"   "insulina"
## [1] "73"         "-0.0025"    "0.0014"     "tomografia"
## [1] "72"          "-0.0033"     "-8e-04"      "cateterismo"
## [1] "72"     "4e-04"  "0.0038" "icp"
## [1] "71"      "-0.0017" "0.0016"  "mpp"
## [1] "70"                  "-7e-04"             "0.0011"              "ventilacao_mecanica"
## [1] "69"          "-0.0022"     "-0.0016"     "ressonancia"
## [1] "68"                     "-0.003"                 "-8e-04"                 "cateter_venoso_central"
## [1] "67"            "-0.0011"       "0.002"         "cintilografia"
## [1] "66"      "-1e-04"  "0.001"   "biopsia"
## [1] "65"        "-9e-04"    "-8e-04"    "antiviral"
## [1] "64"        "-2e-04"    "7e-04"     "ultrassom"
## [1] "63"      "-6e-04"  "-4e-04"  "cultura"
## [1] "62"               "-8e-04"           "-2e-04"           "procedure_type_1"
## [1] "61"               "-6e-04"           "1e-04"            "eletrofisiologia"
## [1] "60"            "-6e-04"        "0"             "heart_failure"
## [1] "59"          "-9e-04"      "-3e-04"      "bloq_calcio"
## [1] "58"                 "-0.0022"            "-0.0013"            "antiplaquetario_ev"
## [1] "57"                       "-0.0014"                  "9e-04"
## [4] "outros_proced_cirurgicos"
## [1] "57"       "0.0013"   "0.0027"   "diabetes"
## [1] "56"             "-2e-04"         "0.0011"         "cardiac_arrest"
## [1] "55"          "-0.0041"     "-0.0039"     "flebografia"
## [1] "55"     "-4e-04" "0.0037" "aco"
## [1] "55"      "-0.0016" "0.0025"  "copd"
## [1] "55"            "-0.0014"       "0.0027"        "interconsulta"
```

```
## [1] "54"                  "-0.0023"            "0.0018"             "cied_final_group_1"
## [1] "53"              "-0.002"          "2e-04"           "n_procedure_t0"
## [1] "52"                  "-0.0021"            "-1e-04"             "procedure_type_new"
## [1] "51"          "-0.0011"    "0.001"      "endoscopia"
## [1] "50"      "-1e-04" "0.001"  "holter"
## [1] "49"                      "-2e-04"                   "-1e-04"
## [4] "underlying_heart_disease"
## [1] "48"      "-4e-04" "-2e-04" "af"
## [1] "47"                      "-2e-04"                 "2e-04"                  "analises_clinicas_qtde"
## [1] "46"        "-0.0013"  "-0.0011"  "digoxina"
## [1] "45"                      "0"                      "0.0012"                 "admission_t0_emergency"
## [1] "44"          "8e-04"       "9e-04"      "nyha_basal"
## [1] "43"      "6e-04"  "-3e-04" "bic"
## [1] "42"                 "3e-04"              "-3e-04"            "anticonvulsivante"
## [1] "41"      "0.0014" "0.0011" "sex"
## [1] "40"              "0.002"            "6e-04"           "education_level"
## [1] "40"             "0.0061"         "0.0041"         "patient_state"
## [1] "39"              "0.002"            "0"                "betabloqueador"
## [1] "38"              "0.0025"           "5e-04"            "ecocardiograma"
## [1] "37"              "0.0033"           "8e-04"            "insuf_cardiaca"
## [1] "36"                     "0.0043"                 "0.0011"                 "admission_pre_t0_180d"
## [1] "36"           "0.0081"       "0.0037"       "reop_type_1"
## [1] "35"      "0.004"  "-3e-04" "dva"
## [1] "34"                   "0.0018"               "-0.0022"              "comorbidities_count"
## [1] "33"                     "0.0032"                 "0.0013"                 "proced_invasivos_qtde"
## [1] "32"                   "0.0048"               "0.0017"               "histopatologia_qtde"
## [1] "32"            "0.0102"       "0.0054"       "cied_final_1"
```

```
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	95	0.7204	0.0137	0.0000	0.0000
All unimportant	TRUE	85	0.7189	0.0134	0.0015	0.0015
cec	TRUE	84	0.7219	0.0135	-0.0015	-0.0030
citologia	TRUE	83	0.7204	0.0140	0.0000	0.0015
renal_failure	TRUE	82	0.7211	0.0133	-0.0007	-0.0007
endocardites	TRUE	81	0.7239	0.0136	-0.0034	-0.0027
teste_esforco	FALSE	80	0.7207	0.0124	-0.0003	0.0031
dialysis_t0	FALSE	80	0.7211	0.0138	-0.0007	0.0027
heart_disease	FALSE	80	0.7207	0.0137	-0.0003	0.0031
transplant	TRUE	80	0.7222	0.0137	-0.0018	0.0016
valvopathy	TRUE	79	0.7205	0.0133	0.0000	0.0018
intervencao_cv	TRUE	78	0.7223	0.0120	-0.0018	-0.0018
angio_tc	TRUE	77	0.7211	0.0134	-0.0006	0.0012
vni	TRUE	76	0.7229	0.0130	-0.0024	-0.0018
antifungico	TRUE	75	0.7221	0.0139	-0.0017	0.0007
cve_desf	TRUE	74	0.7244	0.0137	-0.0040	-0.0023
espiro_ergoespiro	FALSE	73	0.7214	0.0125	-0.0009	0.0030
prior_mi	FALSE	73	0.7219	0.0130	-0.0014	0.0025
insulina	FALSE	73	0.7206	0.0130	-0.0002	0.0038
tomografia	TRUE	73	0.7230	0.0130	-0.0025	0.0014
cateterismo	TRUE	72	0.7238	0.0128	-0.0033	-0.0008
icp	FALSE	71	0.7200	0.0133	0.0004	0.0038
mpp	TRUE	71	0.7222	0.0140	-0.0017	0.0016
ventilacao_mecanica	TRUE	70	0.7211	0.0135	-0.0007	0.0011
ressonancia	TRUE	69	0.7227	0.0135	-0.0022	-0.0016

Table 1: *(continued)*

Tested Feature	Dropped	Features	CV AUC	CV AUC Std Error	Total AUC Loss	Instant AUC Loss
cateter_venoso_central	TRUE	68	0.7235	0.0130	-0.0030	-0.0008
cintilografia	TRUE	67	0.7215	0.0133	-0.0011	0.0020
biopsia	TRUE	66	0.7205	0.0133	-0.0001	0.0010
antiviral	TRUE	65	0.7213	0.0130	-0.0009	-0.0008
ultrassom	TRUE	64	0.7206	0.0142	-0.0002	0.0007
cultura	TRUE	63	0.7210	0.0137	-0.0006	-0.0004
procedure_type_1	TRUE	62	0.7212	0.0130	-0.0008	-0.0002
eletrofisiologia	TRUE	61	0.7211	0.0127	-0.0006	0.0001
heart_failure	TRUE	60	0.7211	0.0124	-0.0006	0.0000
bloq_calcio	TRUE	59	0.7214	0.0125	-0.0009	-0.0003
antiplaquetario_ev	TRUE	58	0.7227	0.0130	-0.0022	-0.0013
outros_proced_cirurgicos	TRUE	57	0.7218	0.0129	-0.0014	0.0009
diabetes	FALSE	56	0.7191	0.0123	0.0013	0.0027
cardiac_arrest	TRUE	56	0.7207	0.0119	-0.0002	0.0011
flebografia	TRUE	55	0.7245	0.0117	-0.0041	-0.0039
aco	FALSE	54	0.7208	0.0121	-0.0004	0.0037
copd	FALSE	54	0.7220	0.0127	-0.0016	0.0025
interconsulta	FALSE	54	0.7218	0.0124	-0.0014	0.0027
cied_final_group_1	TRUE	54	0.7227	0.0135	-0.0023	0.0018
n_procedure_t0	TRUE	53	0.7225	0.0117	-0.0020	0.0002
procedure_type_new	TRUE	52	0.7225	0.0120	-0.0021	-0.0001
endoscopia	TRUE	51	0.7215	0.0114	-0.0011	0.0010
holter	TRUE	50	0.7205	0.0126	-0.0001	0.0010
underlying_heart_disease	TRUE	49	0.7206	0.0116	-0.0002	-0.0001
af	TRUE	48	0.7208	0.0109	-0.0004	-0.0002
analises_clinicas_qtde	TRUE	47	0.7206	0.0125	-0.0002	0.0002
digoxina	TRUE	46	0.7217	0.0113	-0.0013	-0.0011
admission_t0_emergency	TRUE	45	0.7205	0.0120	0.0000	0.0012
nyha_basal	TRUE	44	0.7196	0.0111	0.0008	0.0009
bic	TRUE	43	0.7199	0.0119	0.0006	-0.0003
anticonvulsivante	TRUE	42	0.7201	0.0122	0.0003	-0.0003
sex	TRUE	41	0.7191	0.0128	0.0014	0.0011
education_level	TRUE	40	0.7185	0.0108	0.0020	0.0006
patient_state	FALSE	39	0.7143	0.0093	0.0061	0.0041
betabloqueador	TRUE	39	0.7185	0.0113	0.0020	0.0000
ecocardiograma	TRUE	38	0.7179	0.0115	0.0025	0.0005
insuf_cardiaca	TRUE	37	0.7172	0.0110	0.0033	0.0008
admission_pre_t0_180d	TRUE	36	0.7161	0.0103	0.0043	0.0011
reop_type_1	FALSE	35	0.7124	0.0118	0.0081	0.0037
dva	TRUE	35	0.7164	0.0117	0.0040	-0.0003
comorbidities_count	TRUE	34	0.7186	0.0105	0.0018	-0.0022
proced_invasivos_qtde	TRUE	33	0.7173	0.0101	0.0032	0.0013
histopatologia_qtde	TRUE	32	0.7156	0.0115	0.0048	0.0017
cied_final_1	FALSE	31	0.7102	0.0115	0.0102	0.0054

```
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)
```

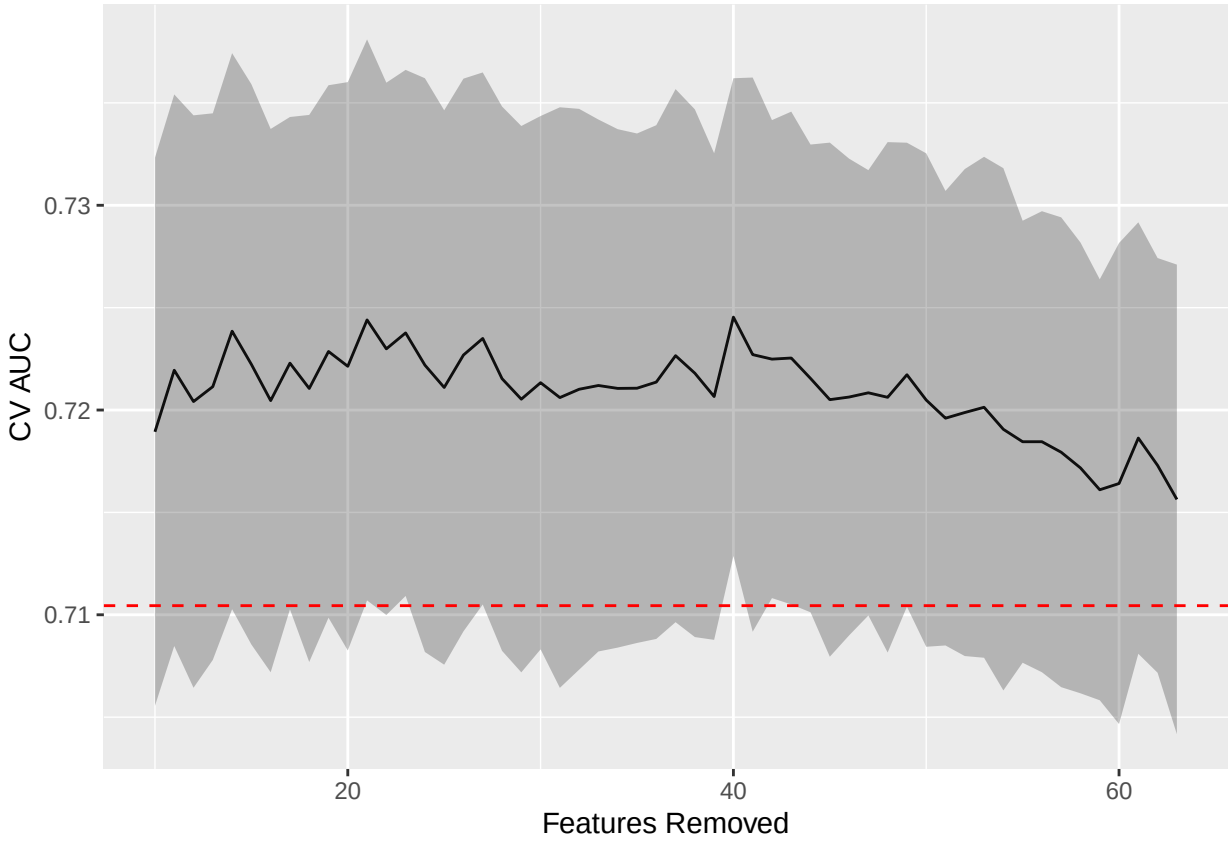
```
close(con)

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: 0.717"
```

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

```
## [1] "Selected Model Test AUC: 0.681"
```

```
 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)
```

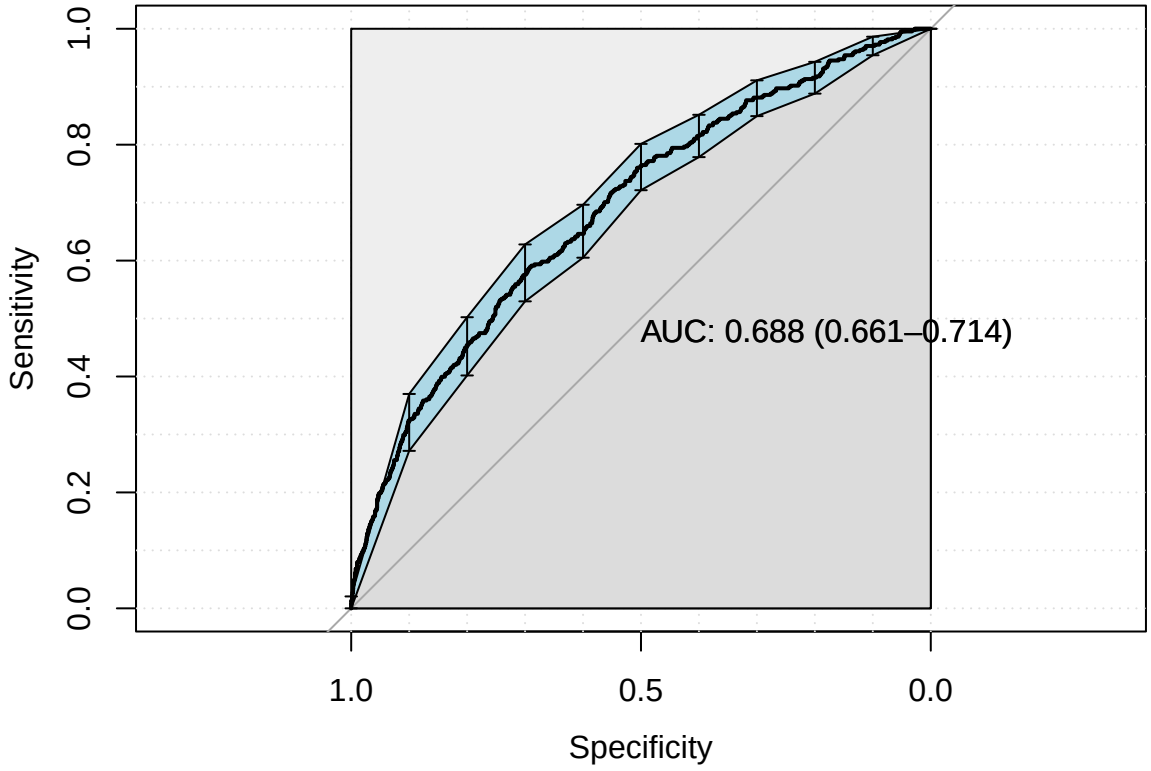
1. hospital_stay
2. age
3. meds_cardiovasc_qtde
4. admission_pre_t0_count
5. icu_t0
6. laboratorio
7. ieca_bra
8. classe_meds_qtde
9. meds_antimicrobianos
10. vasodilatador
11. metodos_graficos_qtde
12. antiarritmico
13. exames_imagem_qtde
14. diuretico
15. psicofarmacos
16. equipe_multiprof
17. estatina
18. cied_final_1
19. reop_type_1
20. espironolactona
21. patient_state
22. aco
23. interconsulta
24. diabetes
25. copd
26. insulina
27. prior_mi
28. icp
29. espiro_ergoespiro
30. heart_disease
31. teste_esforco
32. dialysis_t0

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_smote_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()) %>%
  step_impute_mean(all_numeric_predictors()) %>%
  step_smote(!!sym(outcome_column))

lightgbm_upsample_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()) %>%
  step_upsample(!!sym(outcome_column))
```

```r
lightgbm_tuning <- function(recipe) {

  lightgbm_spec <- boost_tree(
    mtry = tune(),
    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(
    mtry(range = c(1L, length(selected_features))),
    trees(range = c(100L, 300L)),
    min_n(),
    tree_depth(),
    learn_rate(),
    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")

  lightgbm_tune %>%
    collect_metrics() %>%
    filter(.metric == "roc_auc") %>%
    select(mean, mtry:tree_depth) %>%
    pivot_longer(mtry:tree_depth,
                 values_to = "value",
                 names_to = "parameter"
    ) %>%
    ggplot(aes(value, mean, color = parameter)) +
    geom_point(alpha = 0.8, show.legend = FALSE) +
    facet_wrap(~parameter, scales = "free_x") +
    labs(x = NULL, y = "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, mtry, min_n, tree_depth, learn_rate, loss_reduction) %>%
    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)
```



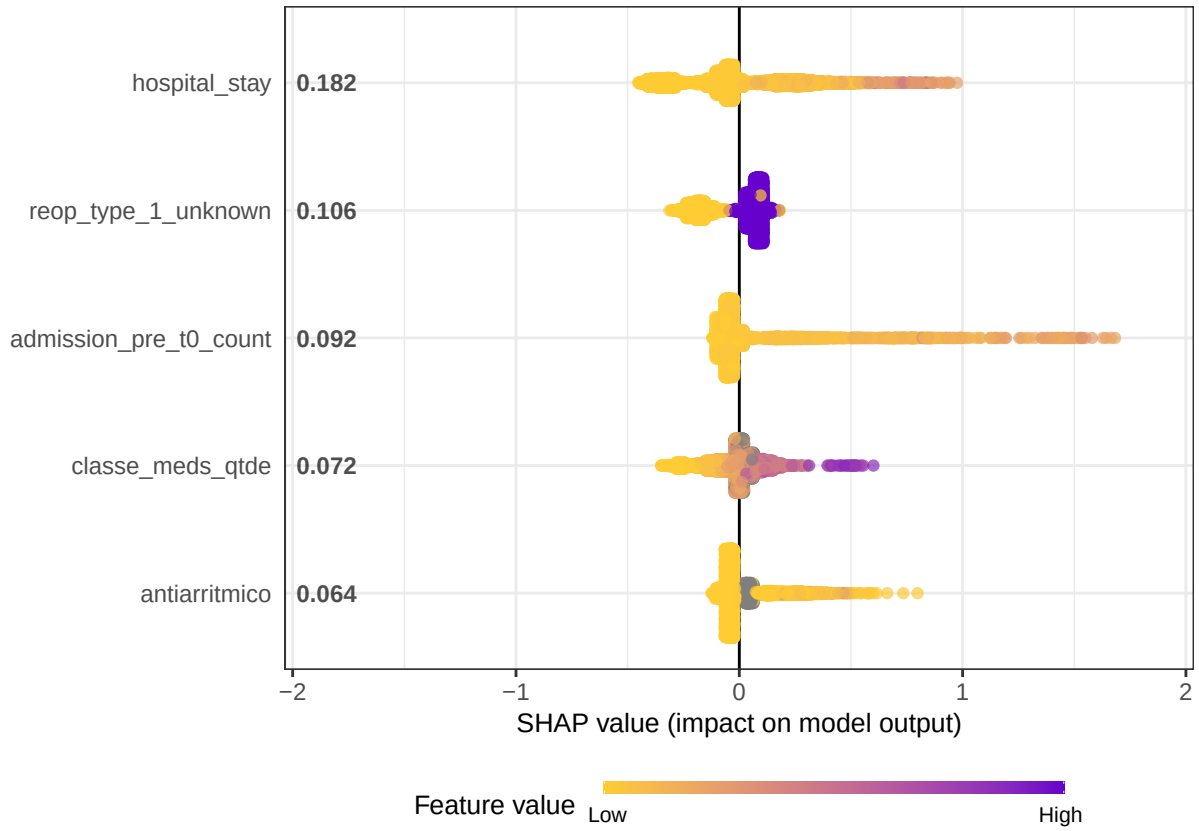

```
## [1] "Optimal Threshold: 0.09"
## Confusion Matrix and Statistics
##
##     reference
## data    0    1
##    0 2969  180
##    1 1323  258
##
##                Accuracy : 0.6822
##                  95% CI : (0.6688, 0.6955)
##     No Information Rate : 0.9074
##     P-Value [Acc > NIR] : 1
##
```

```
##                  Kappa : 0.1293
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6918
##             Specificity : 0.5890
##          Pos Pred Value : 0.9428
##          Neg Pred Value : 0.1632
##              Prevalence : 0.9074
##          Detection Rate : 0.6277
##    Detection Prevalence : 0.6658
##       Balanced Accuracy : 0.6404
##
##        'Positive' Class : 0
##
```

```
# smote_results <- lightgbm_tuning(lightgbm_smote_recipe)
# upsample_results <- lightgbm_tuning(lightgbm_upsample_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
#   )
# )
```

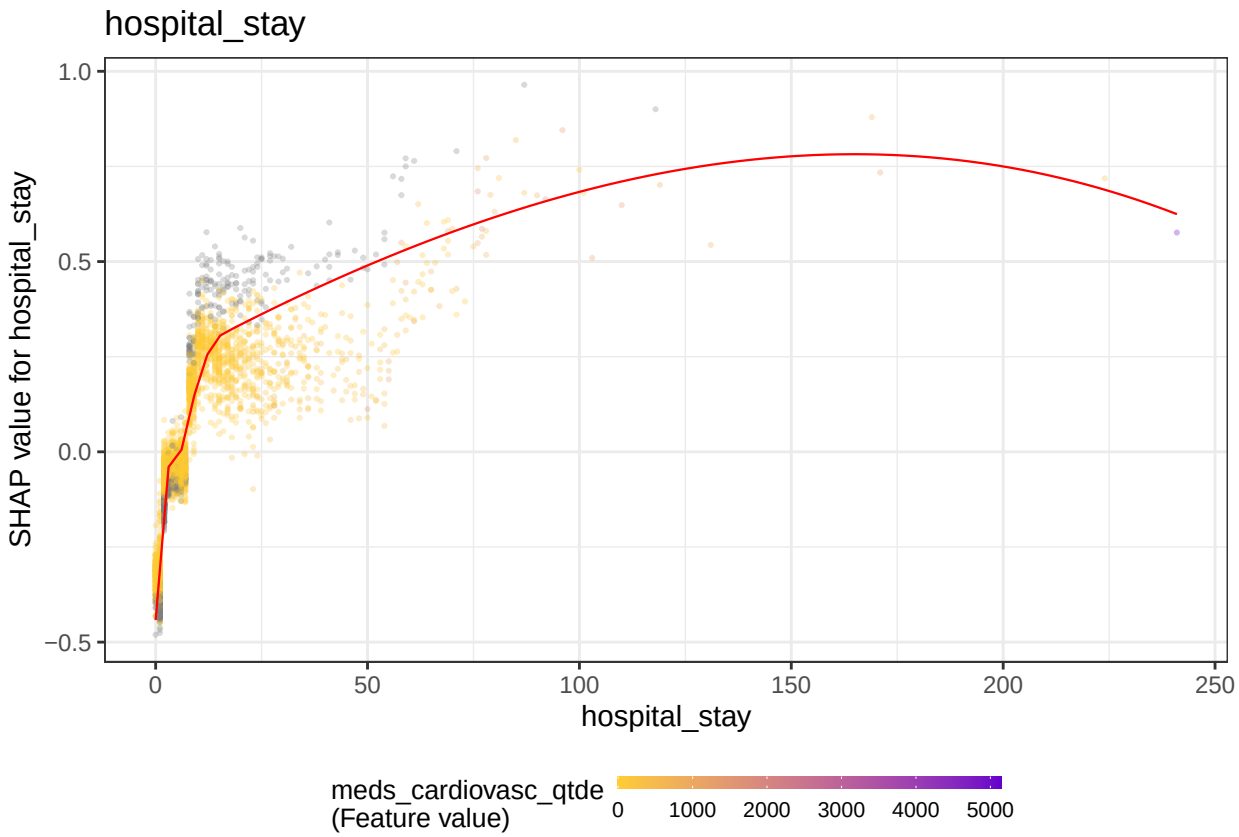
SHAP values

```
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)
```

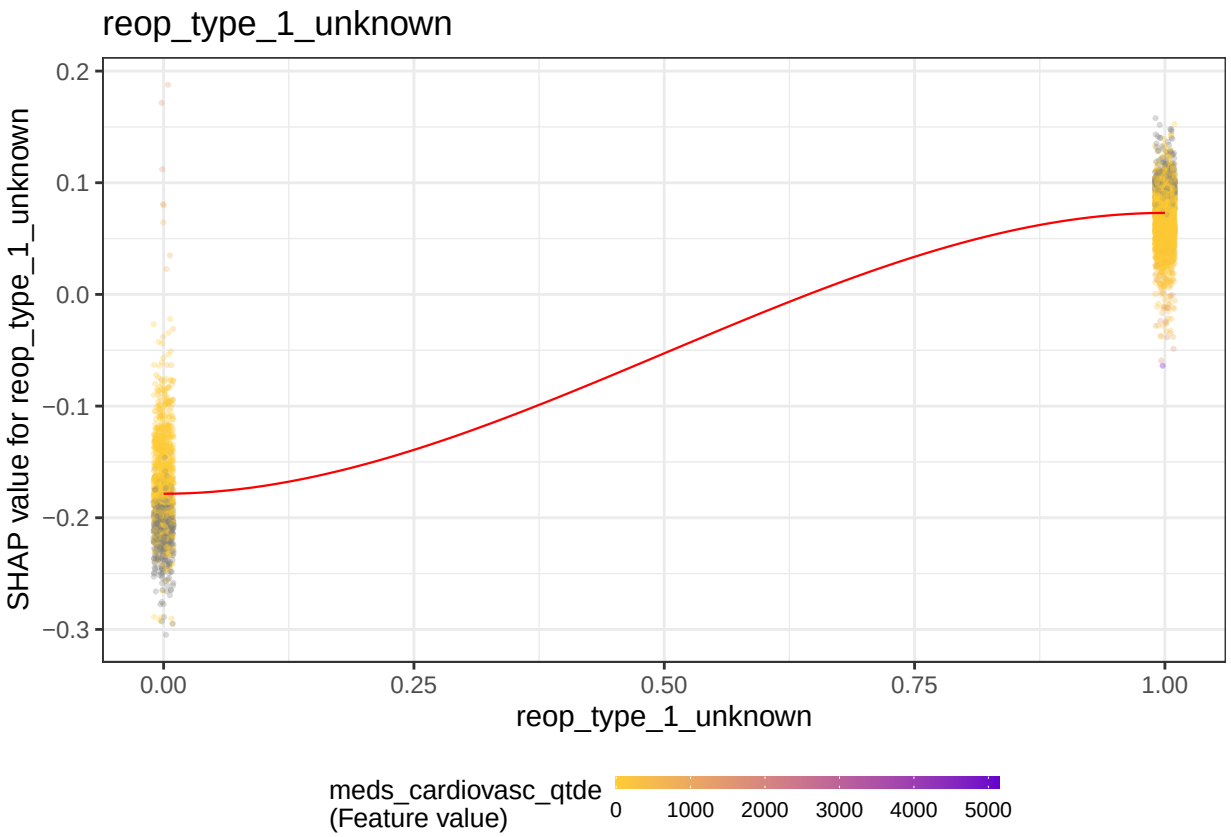



```
shap <- shap.prep(lightgbm_model, X_train = matrix_test)

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

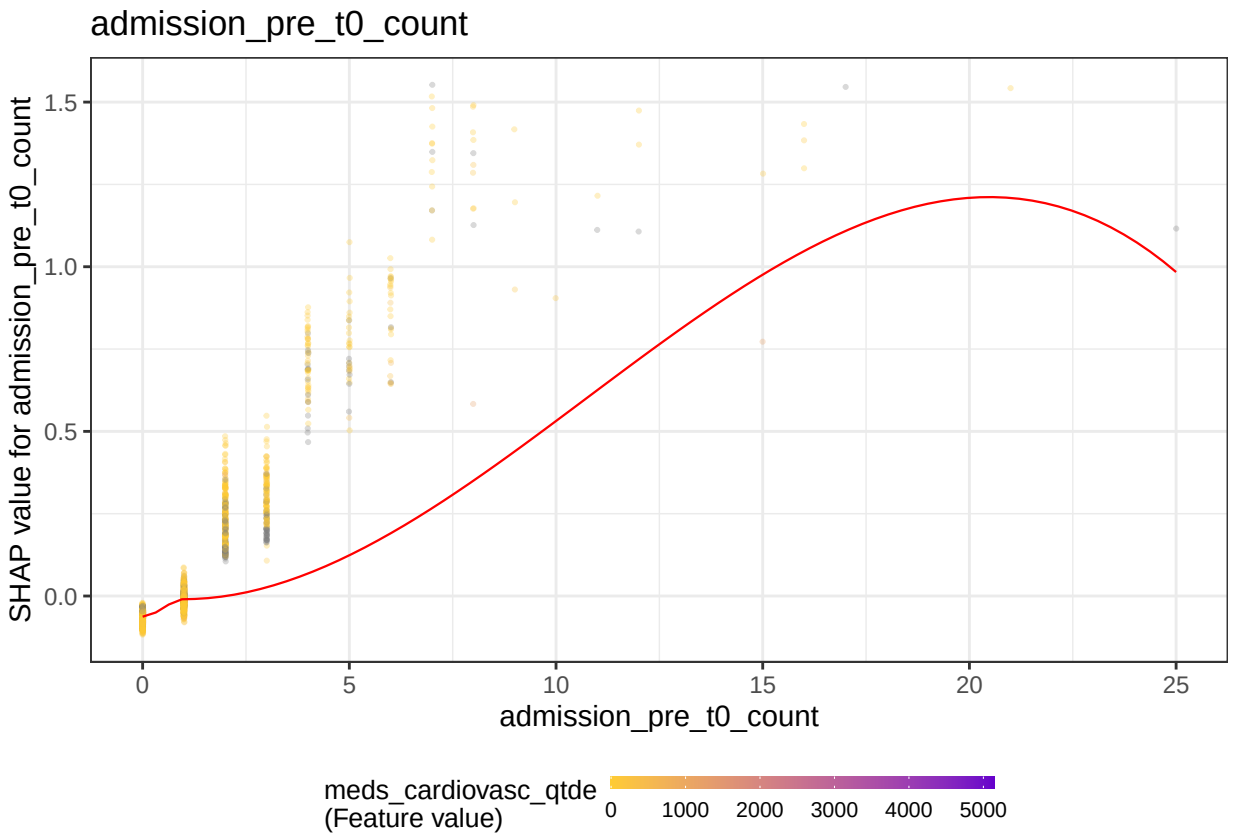
```
## `geom_smooth()` using formula 'y ~ x'
```



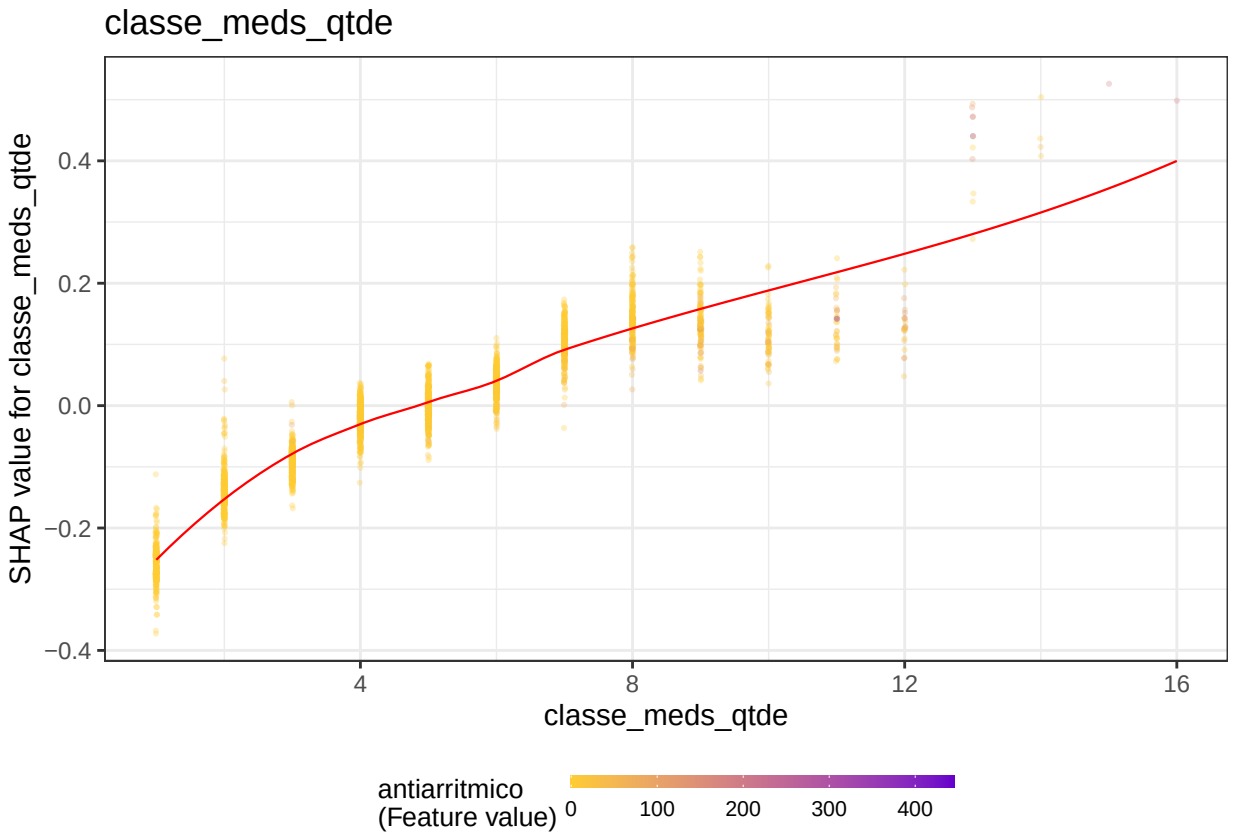

```
## `geom_smooth()` using formula 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : pseudoinverse used at
## -0.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : neighborhood radius
## 1.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : reciprocal condition
## number 5.0206e-29

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : There are other near
## singularities as well. 1.01
```




```
## `geom_smooth()` using formula 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : pseudoinverse used at
## -0.125

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : neighborhood radius
## 1.125

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : reciprocal condition
## number 2.1864e-28

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : There are other near
## singularities as well. 1
```




```
## `geom_smooth()` using formula 'y ~ x'
## Warning: Removed 1472 rows containing non-finite values (stat_smooth).
## Warning: Removed 1472 rows containing missing values (geom_point).
```




```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 1064 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : at -2.225
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : radius 4.9506
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : all data on boundary
## of neighborhood. make span bigger
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : pseudoinverse used at
## -2.225
```

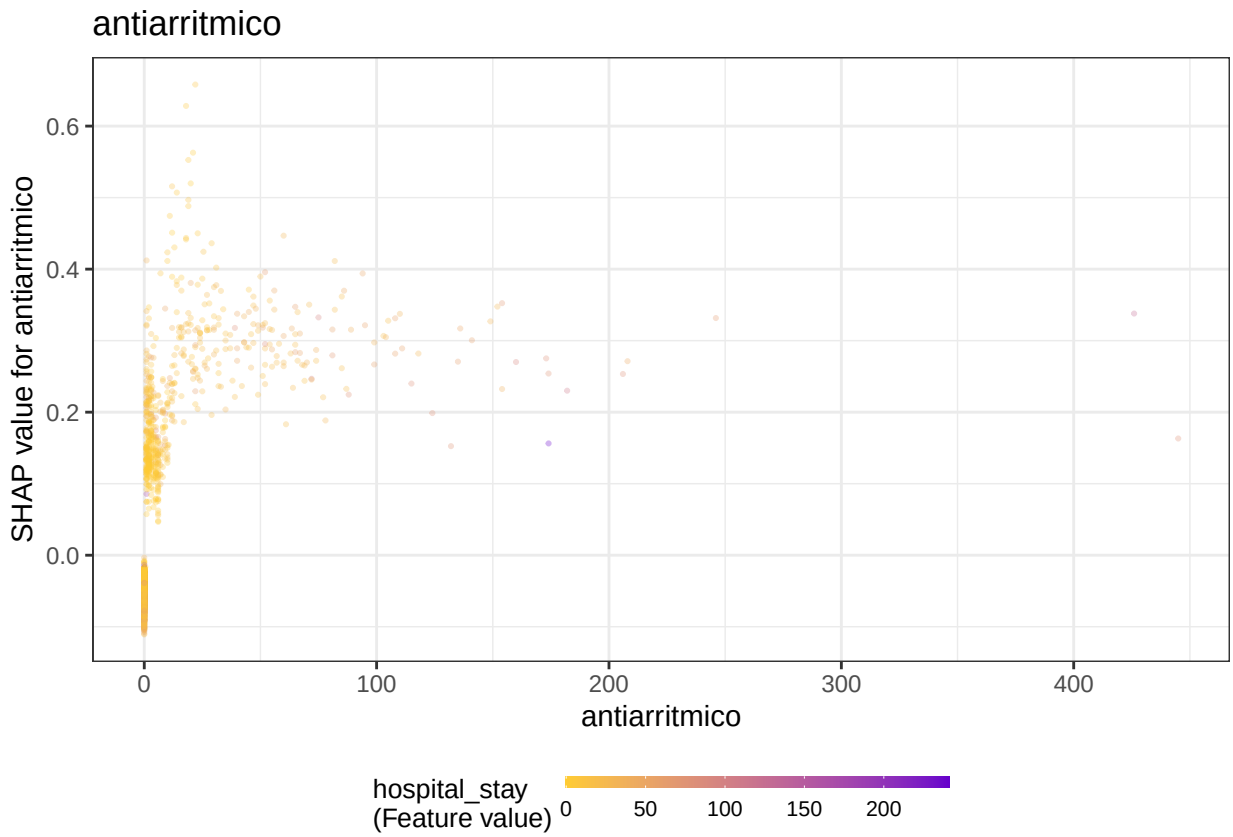
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : neighborhood radius
## 2.225
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : reciprocal condition
## number 1
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : zero-width
## neighborhood. make span bigger
```

```
## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 5)
```

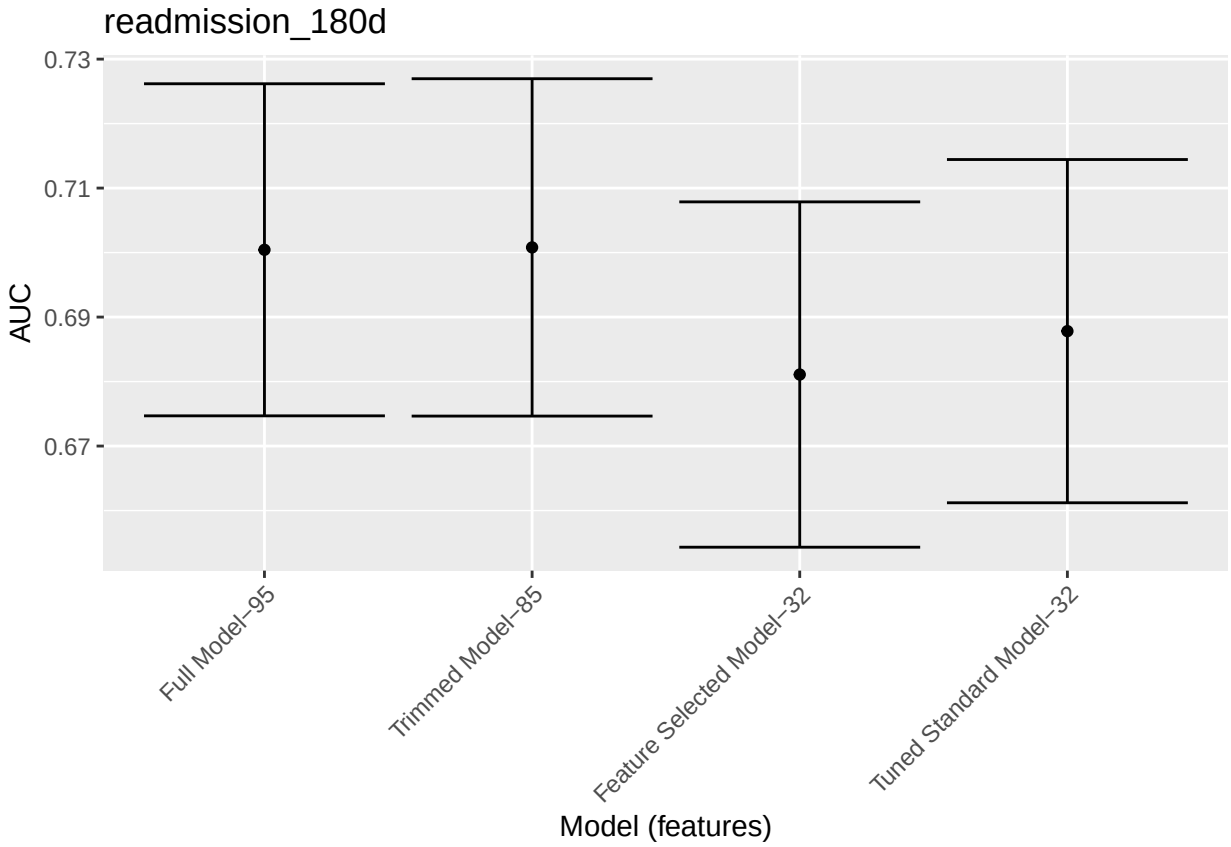
```
## Warning: Removed 1064 rows containing missing values (geom_point).
```



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$auc_upper, length(s
  # 'Tuned Smote Model', smote_results$auc, smote_results$auc_lower, smote_results$auc_upper, length(selected_fe
  # 'Tuned Upsample Model', upsample_results$auc, upsample_results$auc_lower, upsample_results$auc_upper, length
) %>%
```

```
  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))
```