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
Hmisc::list.tree(params)
```

```
##  params = list 5 (968 bytes)
## .  max_auc_loss = double 1= 0.01
## .  outcome_column = character 1= readmission_1year
## .  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
predict <- stats::predict
```

Minutes to run: 0.011

Loading data

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

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

```r
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: 0.006

```r
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

```r
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

```r
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. 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 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: 0.720"
```

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

```
## [1] "Full Model Test AUC: 0.714"
```

Minutes to run: 0.798

Features with zero importance on the initial model:

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

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

1. dialysis_t0
2. antiplaquetario_ev
3. antiviral
4. antiretroviral
5. cec
6. transplante_cardiaco
7. cir_toracica
8. intervencao_cv
9. angioplastia
10. transfusao
11. teste_esforco
12. tilt_teste
13. citologia
14. angio_rm
15. aortografia
16. pet_ct
17. dieta_parenteral
18. mpp 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: 0.719"
```

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

```
## [1] "Trimmed Model Test AUC: 0.714"
```

Minutes to run: 0.756

```
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()
```

```r
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] "78"      "0"       "-2e-04"  "biopsia"
## [1] "77"               "0"                "0"               "procedure_type_1"
## [1] "76"                "-2e-04"            "-1e-04"            "anticonvulsivante"
## [1] "75"         "2e-04"      "3e-04"      "transplant"
## [1] "74"          "-6e-04"      "-8e-04"      "antifungico"
## [1] "73"          "-3e-04"      "2e-04"       "cateterismo"
## [1] "72"            "-0.0011"       "-8e-04"        "renal_failure"
## [1] "71"       "-9e-04"   "2e-04"    "angio_tc"
## [1] "70"     "-8e-04" "1e-04"  "copd"
## [1] "69"      "-0.0012" "-4e-04"  "icp"
## [1] "68"          "-0.0013"     "-1e-04"      "ressonancia"
## [1] "67"       "2e-04"    "0.0015"   "prior_mi"
## [1] "66"             "-9e-04"         "-0.0011"        "cardiac_arrest"
## [1] "65"                "-3e-04"            "6e-04"             "espiro_ergoespiro"
## [1] "64"                      "-0.0011"                 "-8e-04"                    "outros_proced_cirurgico
## [1] "63"            "-0.0014"       "-3e-04"        "arteriografia"
## [1] "62"     "-0.001" "4e-04"  "aco"
## [1] "61"          "-0.0011"     "-1e-04"      "reop_type_1"
## [1] "60"       "-0.0011"  "-1e-04"   "cve_desf"
## [1] "59"       "-0.0016"  "-4e-04"   "diabetes"
## [1] "58"           "-0.0014"      "2e-04"        "hemodialysis"
## [1] "57"          "-0.0012"     "2e-04"       "flebografia"
## [1] "56"         "-0.0018"    "-7e-04"     "tomografia"
## [1] "55"      "-0.0013" "5e-04"   "holter"
## [1] "54"      "-0.0014" "-1e-04"  "cultura"
## [1] "53"               "-0.0025"          "-0.0011"          "eletrofisiologia"
## [1] "52"             "-0.0011"       "0.0014"        "heart_disease"
## [1] "51"             "-9e-04"        "2e-04"         "cintilografia"
## [1] "50"      "-0.0018" "-9e-04"  "af"
## [1] "49"                     "-0.0012"               "6e-04"                   "cateter_venoso_central"
## [1] "48"         "-0.0021"    "-0.001"     "valvopathy"
## [1] "47"       "-5e-04"   "0.0016"   "insulina"
## [1] "46"      "-0.0011" "-6e-04"  "race"
## [1] "45"      "-0.0016" "-4e-04"  "bic"
## [1] "44"         "-0.0015"    "1e-04"      "endoscopia"
## [1] "43"           "-0.0016"      "-1e-04"       "endocardites"
## [1] "42"                 "-0.001"             "6e-04"               "ventilacao_mecanica"
## [1] "41"                    "-0.0015"               "-6e-04"                "analises_clinicas_qtde"
## [1] "40"                   "-0.002"               "-5e-04"               "proced_invasivos_qtde"
## [1] "39"        "-0.0016"   "4e-04"     "ultrassom"
## [1] "38"            "-0.0018"       "-2e-04"        "heart_failure"
## [1] "37"            "-0.0019"       "-1e-04"        "interconsulta"
## [1] "36"             "-0.0025"        "-6e-04"         "ecocardiograma"
## [1] "35"      "-0.0027" "-2e-04"  "sex"
## [1] "34"       "-0.0016"  "0.0011"   "digoxina"
## [1] "33"                 "-8e-04"             "8e-04"               "histopatologia_qtde"
## [1] "32"            "-5e-04"         "3e-04"          "betabloqueador"
## [1] "31"                    "-5e-04"                 "0"                       "admission_t0_emergency"
## [1] "30"                "-1e-04"             "5e-04"              "cied_final_group_1"
## [1] "29"          "-4e-04"      "-3e-04"      "bloq_calcio"
## [1] "28"            "0.0012"        "0.0016"        "patient_state"
## [1] "27"      "-1e-04"  "-0.0013" "dva"
## [1] "26"             "-2e-04"         "0"              "insuf_cardiaca"
## [1] "25"                 "0.0011"             "0.0013"             "exames_imagem_qtde"
## [1] "24"            "2e-04"         "-9e-04"        "psicofarmacos"
## [1] "23"       "-8e-04"   "-0.001"   "estatina"
```

```
## [1] "22"                  "-9e-04"             "-1e-04"             "comorbidities_count"
## [1] "22"                    "-9e-04"               "0.0028"               "admission_pre_t0_180d"
## [1] "22"          "-9e-04"     "0.0021"     "nyha_basal"
## [1] "22"            "-9e-04"       "0.0034"       "cied_final_1"
## [1] "21"                "7e-04"            "0.0016"           "equipe_multiprof"
## [1] "20"                      "0.0014"                   "7e-04"                    "underlying_heart_diseas
## [1] "19"                "0.0016"          "2e-04"           "education_level"
## [1] "19"                   "0.0016"             "0.0024"             "procedure_type_new"
## [1] "18"                "0.0023"          "6e-04"           "espironolactona"
## [1] "17"                     "0.0031"                "9e-04"                "metodos_graficos_qtde"
## [1] "16"         "0.004"     "9e-04"     "diuretico"
## [1] "15"             "0.0059"        "0.0018"        "antiarritmico"
## [1] "14"           "0.0043"      "-0.0015"     "laboratorio"
## [1] "14"     "0.0043" "0.0028" "icu_t0"
## [1] "13"       "0.0045"   "2e-04"    "ieca_bra"
## [1] "13"             "0.0045"        "0.0023"        "vasodilatador"
## [1] "13"                   "0.0045"               "0.0026"               "meds_antimicrobianos"
## [1] "12"                "0.0041"           "-4e-04"           "classe_meds_qtde"
## [1] "12"           "0.0041"      "0.0084"      "year_adm_t0"
## [1] "11"                   "0.0057"               "0.0016"               "meds_cardiovasc_qtde"
## [1] "11"                     "0.0057"                "0.0157"                "admission_pre_t0_count"
## [1] "10"     "0.005"  "-7e-04" "age"
## [1] "10"             "0.005"         "0.0134"        "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	0.7196	0.0053	0.0000	0.0000
All unimportant	TRUE	79	0.7194	0.0054	0.0002	0.0002
biopsia	TRUE	78	0.7196	0.0054	0.0000	-0.0002
procedure_type_1	TRUE	77	0.7196	0.0054	0.0000	0.0000
anticonvulsivante	TRUE	76	0.7197	0.0052	-0.0002	-0.0001
transplant	TRUE	75	0.7194	0.0052	0.0002	0.0003
antifungico	TRUE	74	0.7201	0.0055	-0.0006	-0.0008
cateterismo	TRUE	73	0.7199	0.0054	-0.0003	0.0002
renal_failure	TRUE	72	0.7207	0.0053	-0.0011	-0.0008
angio_tc	TRUE	71	0.7205	0.0054	-0.0009	0.0002
copd	TRUE	70	0.7204	0.0055	-0.0008	0.0001
icp	TRUE	69	0.7208	0.0054	-0.0012	-0.0004
ressonancia	TRUE	68	0.7209	0.0057	-0.0013	-0.0001
prior_mi	TRUE	67	0.7194	0.0053	0.0002	0.0015
cardiac_arrest	TRUE	66	0.7205	0.0055	-0.0009	-0.0011
espiro_ergoespiro	TRUE	65	0.7199	0.0055	-0.0003	0.0006
outros_proced_cirurgicos	TRUE	64	0.7207	0.0057	-0.0011	-0.0008
arteriografia	TRUE	63	0.7209	0.0056	-0.0014	-0.0003
aco	TRUE	62	0.7206	0.0056	-0.0010	0.0004
reop_type_1	TRUE	61	0.7206	0.0056	-0.0011	-0.0001
cve_desf	TRUE	60	0.7207	0.0055	-0.0011	-0.0001
diabetes	TRUE	59	0.7211	0.0054	-0.0016	-0.0004
hemodialysis	TRUE	58	0.7209	0.0054	-0.0014	0.0002
flebografia	TRUE	57	0.7207	0.0054	-0.0012	0.0002
tomografia	TRUE	56	0.7214	0.0054	-0.0018	-0.0007
holter	TRUE	55	0.7209	0.0055	-0.0013	0.0005
cultura	TRUE	54	0.7209	0.0053	-0.0014	-0.0001
eletrofisiologia	TRUE	53	0.7221	0.0055	-0.0025	-0.0011

Table 1: *(continued)*

Tested Feature	Dropped	Features	CV AUC	CV AUC Std Error	Total AUC Loss	Instant AUC Loss
heart_disease	TRUE	52	0.7207	0.0055	-0.0011	0.0014
cintilografia	TRUE	51	0.7204	0.0056	-0.0009	0.0002
af	TRUE	50	0.7214	0.0054	-0.0018	-0.0009
cateter_venoso_central	TRUE	49	0.7207	0.0053	-0.0012	0.0006
valvopathy	TRUE	48	0.7217	0.0055	-0.0021	-0.0010
insulina	TRUE	47	0.7201	0.0056	-0.0005	0.0016
race	TRUE	46	0.7207	0.0055	-0.0011	-0.0006
bic	TRUE	45	0.7211	0.0056	-0.0016	-0.0004
endoscopia	TRUE	44	0.7211	0.0054	-0.0015	0.0001
endocardites	TRUE	43	0.7212	0.0055	-0.0016	-0.0001
ventilacao_mecanica	TRUE	42	0.7205	0.0056	-0.0010	0.0006
analises_clinicas_qtde	TRUE	41	0.7211	0.0055	-0.0015	-0.0006
proced_invasivos_qtde	TRUE	40	0.7216	0.0054	-0.0020	-0.0005
ultrassom	TRUE	39	0.7212	0.0056	-0.0016	0.0004
heart_failure	TRUE	38	0.7214	0.0054	-0.0018	-0.0002
interconsulta	TRUE	37	0.7215	0.0056	-0.0019	-0.0001
ecocardiograma	TRUE	36	0.7221	0.0055	-0.0025	-0.0006
sex	TRUE	35	0.7223	0.0057	-0.0027	-0.0002
digoxina	TRUE	34	0.7212	0.0055	-0.0016	0.0011
histopatologia_qtde	TRUE	33	0.7204	0.0054	-0.0008	0.0008
betabloqueador	TRUE	32	0.7201	0.0056	-0.0005	0.0003
admission_t0_emergency	TRUE	31	0.7201	0.0053	-0.0005	0.0000
cied_final_group_1	TRUE	30	0.7196	0.0056	-0.0001	0.0005
bloq_calcio	TRUE	29	0.7200	0.0053	-0.0004	-0.0003
patient_state	TRUE	28	0.7184	0.0053	0.0012	0.0016
dva	TRUE	27	0.7197	0.0054	-0.0001	-0.0013
insuf_cardiaca	TRUE	26	0.7197	0.0053	-0.0002	0.0000
exames_imagem_qtde	TRUE	25	0.7185	0.0055	0.0011	0.0013
psicofarmacos	TRUE	24	0.7193	0.0056	0.0002	-0.0009
estatina	TRUE	23	0.7203	0.0053	-0.0008	-0.0010
comorbidities_count	TRUE	22	0.7204	0.0055	-0.0009	-0.0001
admission_pre_t0_180d	FALSE	21	0.7176	0.0053	-0.0009	0.0028
nyha_basal	FALSE	21	0.7184	0.0048	-0.0009	0.0021
cied_final_1	FALSE	21	0.7171	0.0054	-0.0009	0.0034
equipe_multiprof	TRUE	21	0.7188	0.0055	0.0007	0.0016
underlying_heart_disease	TRUE	20	0.7181	0.0055	0.0014	0.0007
education_level	TRUE	19	0.7179	0.0054	0.0016	0.0002
procedure_type_new	FALSE	18	0.7156	0.0058	0.0016	0.0024
espironolactona	TRUE	18	0.7173	0.0054	0.0023	0.0006
metodos_graficos_qtde	TRUE	17	0.7164	0.0055	0.0031	0.0009
diuretico	TRUE	16	0.7156	0.0053	0.0040	0.0009
antiarritmico	TRUE	15	0.7137	0.0052	0.0059	0.0018
laboratorio	TRUE	14	0.7153	0.0049	0.0043	-0.0015
icu_t0	FALSE	13	0.7124	0.0051	0.0043	0.0028
ieca_bra	TRUE	13	0.7151	0.0045	0.0045	0.0002
vasodilatador	FALSE	12	0.7128	0.0043	0.0045	0.0023
meds_antimicrobianos	FALSE	12	0.7125	0.0047	0.0045	0.0026
classe_meds_qtde	TRUE	12	0.7154	0.0049	0.0041	-0.0004
year_adm_t0	FALSE	11	0.7070	0.0045	0.0041	0.0084
meds_cardiovasc_qtde	TRUE	11	0.7139	0.0043	0.0057	0.0016
admission_pre_t0_count	FALSE	10	0.6982	0.0049	0.0057	0.0157
age	TRUE	10	0.7146	0.0051	0.0050	-0.0007
hospital_stay	FALSE	9	0.7011	0.0057	0.0050	0.0134

Minutes to run: 51.631

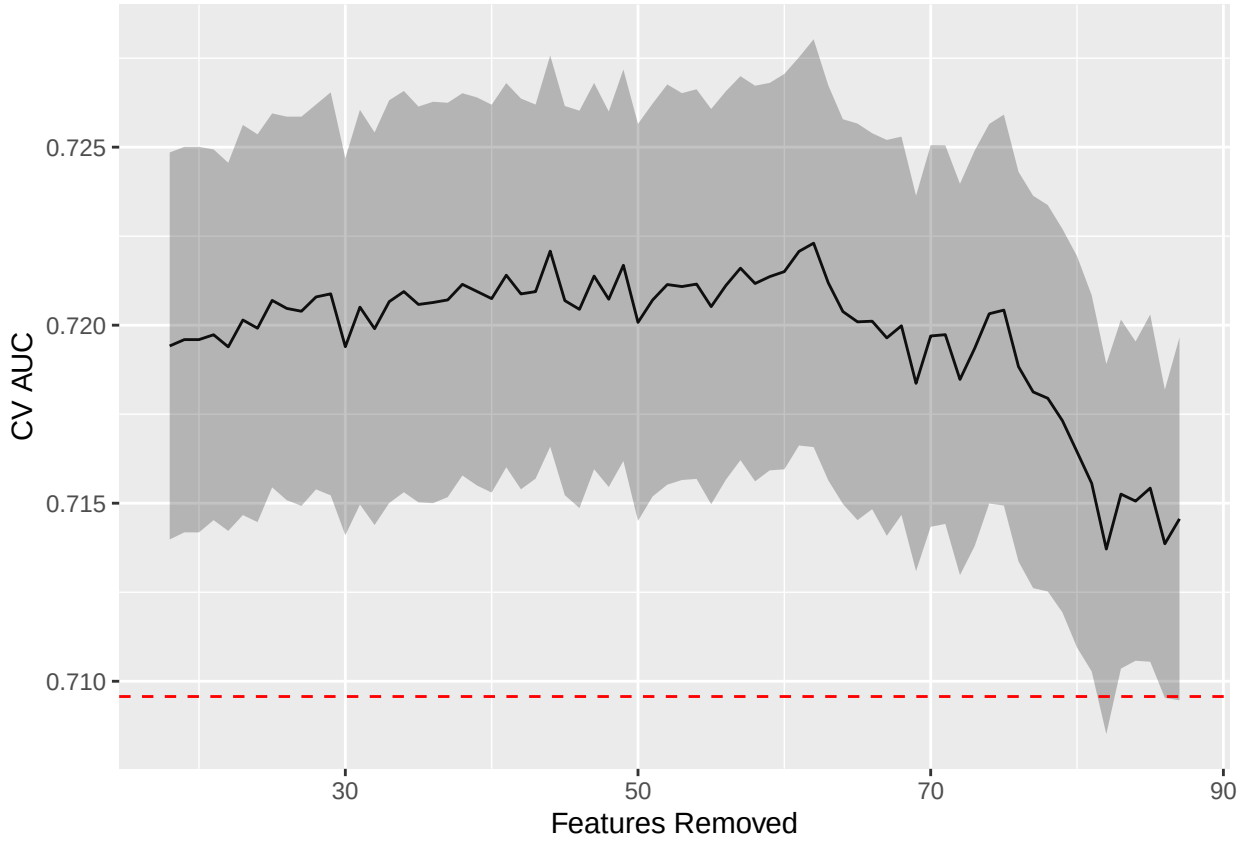
```
selected_features <- current_features

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: 0.711"
```

```
 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: 0.567

Hyperparameter tuning

Selected features:

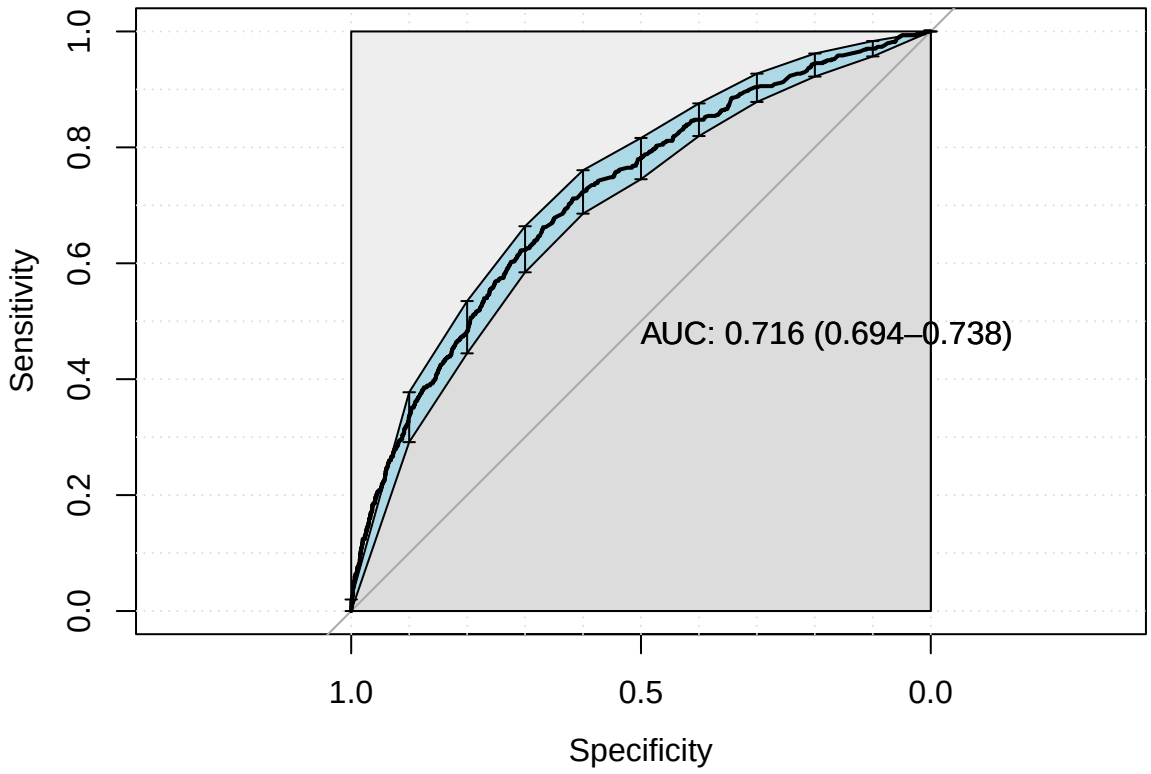
```
gluedown::md_order(selected_features, seq = TRUE, pad = TRUE)
```

1. hospital_stay
2. admission_pre_t0_count
3. year_adm_t0
4. meds_antimicrobianos
5. vasodilatador
6. icu_t0
7. procedure_type_new
8. nyha_basal
9. admission_pre_t0_180d
10. cied_final_1 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(2L, 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)
```

```r
  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: 0.11"
## Confusion Matrix and Statistics
```

```
##
##     reference
## data    0    1
##    0 2761  204
##    1 1366  400
##
##                Accuracy : 0.6681
##                  95% CI : (0.6545, 0.6816)
##     No Information Rate : 0.8723
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1819
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6690
##             Specificity : 0.6623
##          Pos Pred Value : 0.9312
##          Neg Pred Value : 0.2265
##              Prevalence : 0.8723
##          Detection Rate : 0.5836
##    Detection Prevalence : 0.6267
##       Balanced Accuracy : 0.6656
##
##        'Positive' Class : 0
##
```

```r
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))
```

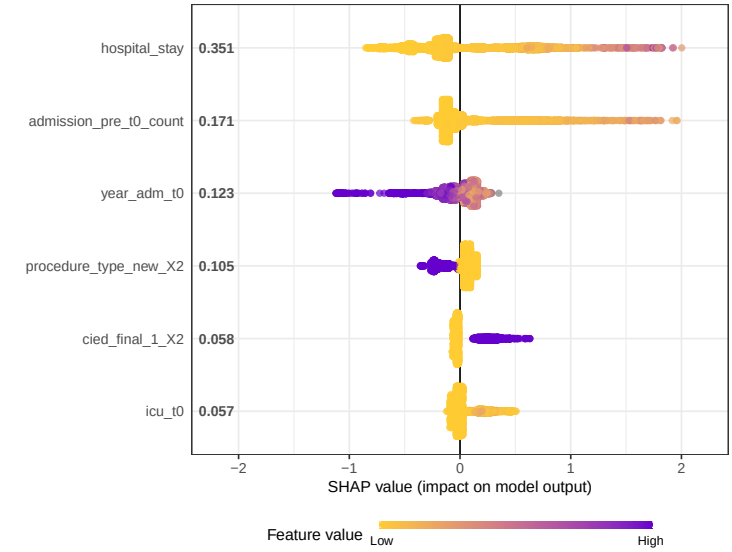
Minutes to run: 13.321

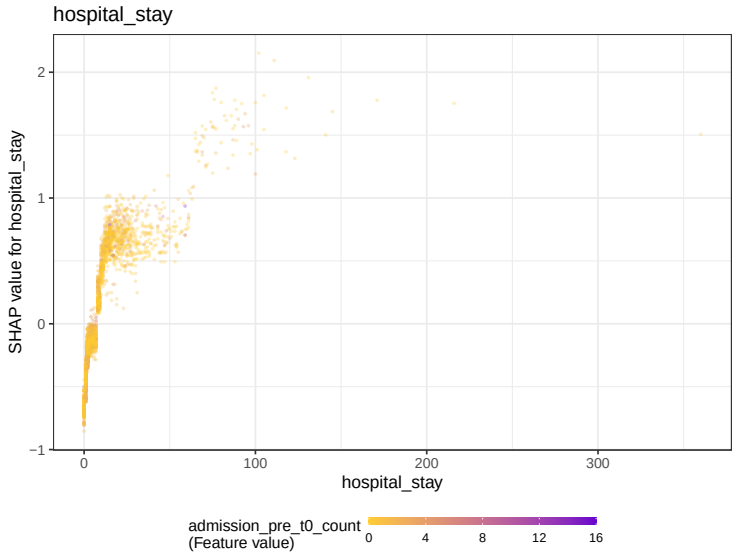
SHAP values

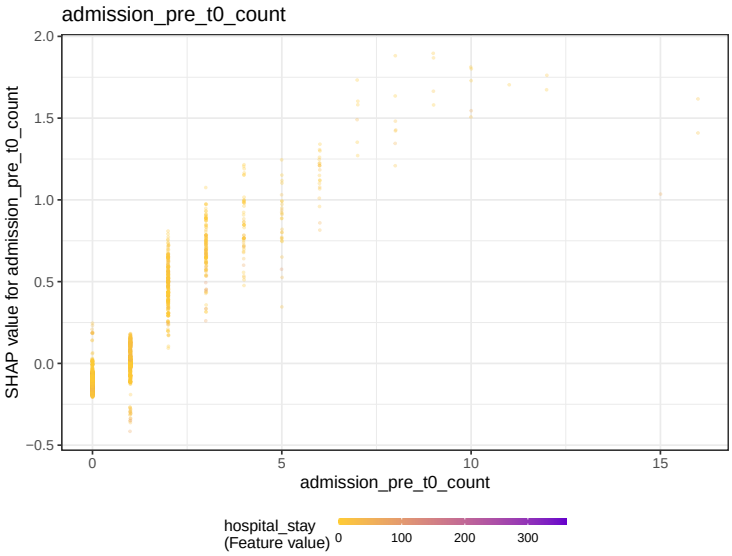
```r
lightgbm_model <- parsnip::extract_fit_engine(final_lightgbm_fit)

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))
```

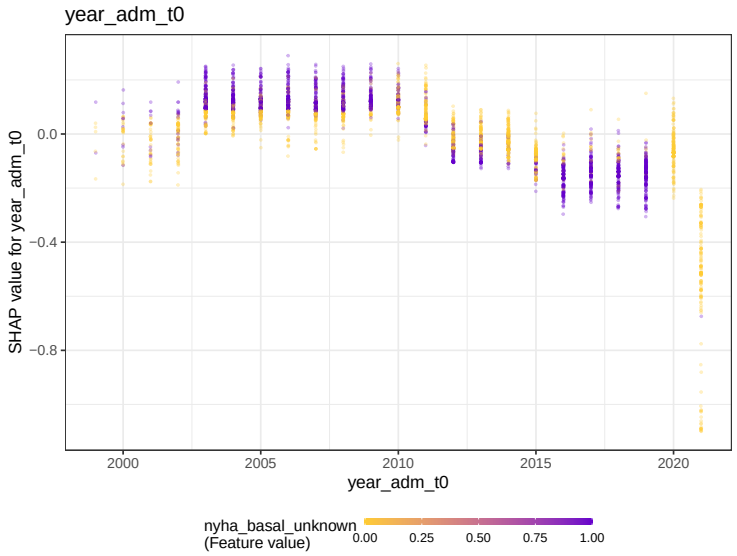
```r
plotted <- 0

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)
}
```

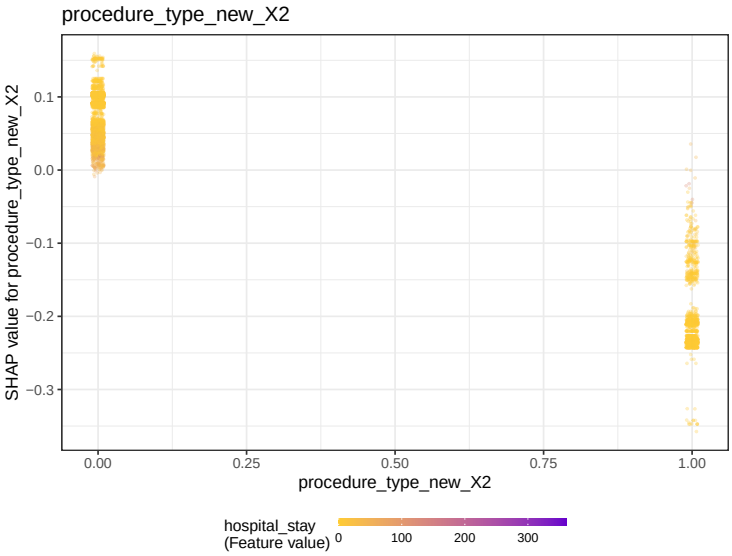
```
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Warning: Removed 5 rows containing missing values (geom_point).
## Saving 6.5 x 5 in image
## Warning: Removed 5 rows containing missing values (geom_point).
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Warning: Removed 1050 rows containing missing values (geom_point).
## Saving 6.5 x 5 in image
## Warning: Removed 1050 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
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
```

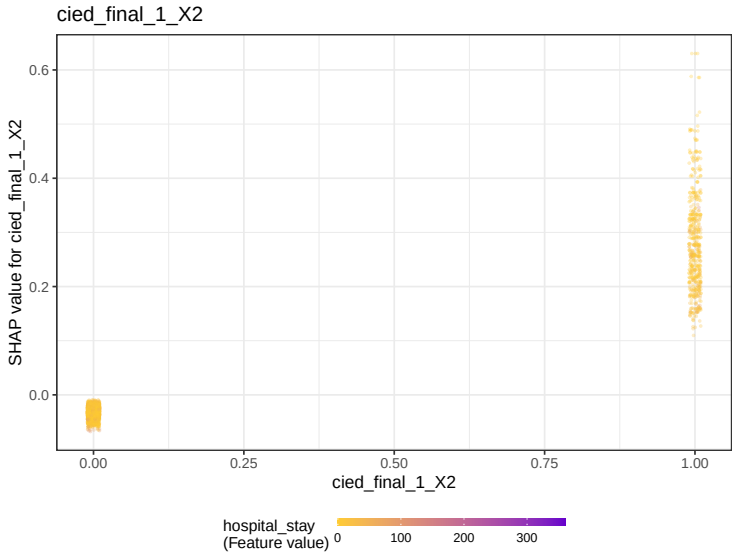

hospital_stay 0.351
admission_pre_t0_count 0.171
year_adm_t0 0.123
procedure_type_new_X2 0.105
cied_final_1_X2 0.058
icu_t0 0.057
SHAP value (impact on model output)
Feature value
Low
High

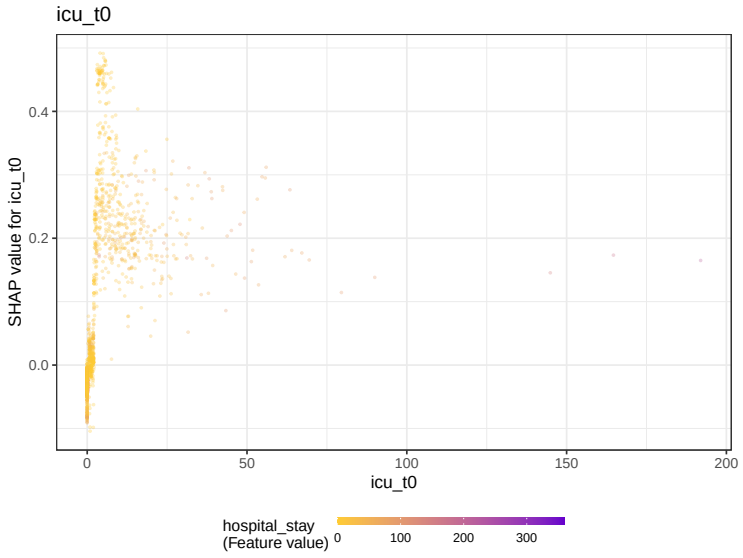

hospital_stay
SHAP value for hospital_stay
hospital_stay
admission_pre_t0_count
(Feature value)


admission_pre_t0_count
SHAP value for admission_pre_t0_count
admission_pre_t0_count
hospital_stay
(Feature value)


year_adm_t0
SHAP value for year_adm_t0
year_adm_t0
nyha_basal_unknown
(Feature value)


procedure_type_new_X2
SHAP value for procedure_type_new_X2
procedure_type_new_X2
hospital_stay
(Feature value)


cied_final_1_X2
SHAP value for cied_final_1_X2
cied_final_1_X2
hospital_stay
(Feature value)



Minutes to run: 0.225

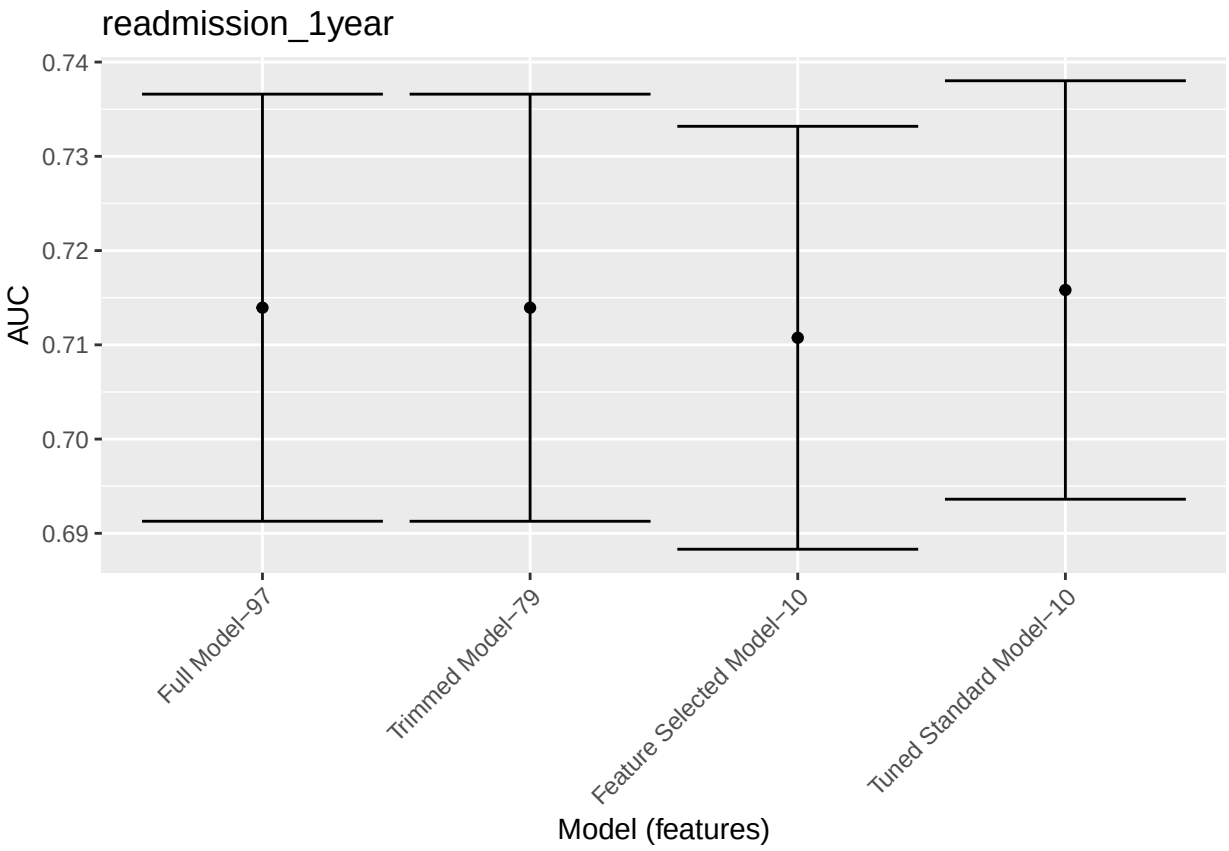
```
## $num_iterations
## [1] 79
##
## $learning_rate
## [1] 0.05136484
##
## $max_depth
## [1] 4
##
## $feature_fraction_bynode
## [1] 1
##
## $min_data_in_leaf
## [1] 42
##
## $min_gain_to_split
## [1] 0
##
## $bagging_fraction
## [1] 1
##
## $num_class
## [1] 1
##
## $objective
## [1] "binary"
##
## $num_threads
## $num_threads$num_threads
## [1] 0
##
##
## $nthread
## [1] 8
##
## $seed
## [1] 59992
##
## $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