Final Model - death_30days

Eduardo Yuki Yada

Global parameters

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

```
##  params = list 5 (952 bytes)
## .  max_auc_loss = double 1= 0.01
## .  outcome_column = character 1= death_30days
## .  k = double 1= 10
## .  grid_size = double 1= 50
## .  repeats = double 1= 2
```

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
```

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
```

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

```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)

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

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

Eligible features

```r
cat_features_list = read_yaml(sprintf(
  "./auxiliar/significant_columns/categorical_%s.yaml",
  outcome_column
))

num_features_list = read_yaml(sprintf(
  "./auxiliar/significant_columns/numerical_%s.yaml",
  outcome_column
))

features_list = c(cat_features_list, num_features_list)
```

```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)
```

Starting features:

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

1. sex
2. age
3. education_level
4. underlying_heart_disease
5. heart_disease
6. nyha_basal
7. hypertension
8. prior_mi
9. heart_failure
10. af
11. valvopathy
12. diabetes
13. renal_failure
14. hemodialysis
15. comorbidities_count
16. procedure_type_1
17. reop_type_1
18. procedure_type_new
19. cied_final_1
20. cied_final_group_1
21. admission_pre_t0_count
22. admission_pre_t0_180d
23. year_adm_t0
24. icu_t0
25. antiarritmico
26. antihipertensivo
27. betabloqueador
28. dva
29. diuretico
30. vasodilatador
31. espironolactona
32. antiplaquetario_ev
33. insulina
34. psicofarmacos
35. antifungico
36. classe_meds_qtde
37. meds_cardiovasc_qtde
38. meds_antimicrobianos
39. vni
40. ventilacao_mecanica
41. intervencao_cv
42. cateter_venoso_central
43. proced_invasivos_qtde
44. transfusao
45. interconsulta
46. equipe_multiprof
47. holter
48. metodos_graficos_qtde
49. laboratorio
50. cultura
51. analises_clinicas_qtde
52. citologia
53. histopatologia_qtde
54. angio_tc
55. angiografia
56. cintilografia
57. ecocardiograma
58. flebografia
59. ultrassom

60. tomografia
61. ressonancia
62. exames_imagem_qtde
63. bic
64. 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 <- read_yaml(
  sprintf(
    "./auxiliar/model_selection/hyperparameters/%s.yaml",
    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: 0.727"
```

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

```
## [1] "Full Model Test AUC: 0.750"
```

Features with zero importance on the initial model:

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

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

1. prior_mi
2. valvopathy
3. hemodialysis
4. antiplaquetario_ev
5. intervencao_cv
6. cateter_venoso_central
7. transfusao
8. cintilografia

```
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: 0.730"
```

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

```
## [1] "Trimmed Model Test AUC: 0.750"
```

```r
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] "55"      "-0.0035" "0"       "vni"
## [1] "54"                  "-0.0031"             "4e-04"
## [4] "histopatologia_qtde"
## [1] "53"        "-0.0074"   "-0.0044"   "ultrassom"
## [1] "52"          "-0.0074"     "1e-04"       "antifungico"
## [1] "52"                  "-0.0074"            "0.0033"
## [4] "cied_final_group_1"
## [1] "51"       "-0.0098"  "-0.0024"  "angio_tc"
## [1] "50"                 "-0.0115"            "-0.0017"
## [4] "procedure_type_new"
## [1] "49"            "-0.0114"       "1e-04"         "heart_failure"
## [1] "48"          "-0.01"       "0.0013"      "reop_type_1"
## [1] "47"                  "-0.0112"             "-0.0011"
## [4] "ventilacao_mecanica"
## [1] "46"          "-0.0152"     "-0.004"      "flebografia"
## [1] "46"                    "-0.0152"               "0.0182"
## [4] "admission_pre_t0_180d"
## [1] "45"            "-0.0157"       "-5e-04"        "heart_disease"
## [1] "44"       "-0.0145"  "0.0012"   "insulina"
## [1] "44"        "-0.0145"   "0.0066"    "citologia"
## [1] "44"      "-0.0145" "0.0027"  "sex"
## [1] "43"       "-0.0158"  "-0.0013"  "diabetes"
## [1] "43"             "-0.0158"        "0.0071"         "ecocardiograma"
## [1] "43"             "-0.0158"        "0.0051"         "betabloqueador"
## [1] "43"               "-0.0158"          "0.0029"           "procedure_type_1"
## [1] "43"      "-0.0158" "0.0162"  "af"
## [1] "43"         "-0.0158"    "0.0099"     "tomografia"
## [1] "43"      "-0.0158" "0.0072"  "cultura"
## [1] "43"                    "-0.0158"               "0.002"
## [4] "proced_invasivos_qtde"
## [1] "42"          "-0.017"      "-0.0012"     "angiografia"
## [1] "42"            "-0.017"        "0.0041"        "interconsulta"
## [1] "41"               "-0.0189"          "-0.0019"          "antihipertensivo"
## [1] "41"      "-0.0189" "0.0047"  "dva"
## [1] "41"          "-0.0189"     "0.0243"      "ressonancia"
## [1] "41"            "-0.0189"       "0.0363"        "renal_failure"
## [1] "41"           "-0.0189"      "0.013"        "hypertension"
## [1] "41"           "-0.0189"      "0.0049"       "cied_final_1"
## [1] "41"               "-0.0189"          "0.0028"           "classe_meds_qtde"
## [1] "40"                  "-0.0183"             "6e-04"
## [4] "comorbidities_count"
## [1] "40"        "-0.0183"   "0.015"     "diuretico"
## [1] "40"      "-0.0183" "0.008"   "holter"
## [1] "39"                 "-0.0203"            "-0.0019"
## [4] "exames_imagem_qtde"
```

```
## [1] "38"            "-0.0238"       "-0.0035"       "antiarritmico"
## [1] "38"          "-0.0238"    "0.022"      "nyha_basal"
## [1] "38"               "-0.0238"          "0.004"            "equipe_multiprof"
## [1] "38"      "-0.0238" "0.0139"  "bic"
## [1] "38"                     "-0.0238"               "0.005"
## [4] "analises_clinicas_qtde"
## [1] "37"                      "-0.0259"
## [3] "-0.0021"                 "underlying_heart_disease"
## [1] "36"              "-0.024"          "0.0018"          "espironolactona"
## [1] "36"                   "-0.024"               "0.0028"
## [4] "meds_cardiovasc_qtde"
## [1] "36"            "-0.024"        "0.0162"        "vasodilatador"
## [1] "36"              "-0.024"          "0.0262"          "education_level"
## [1] "35"          "-0.0263"     "-0.0022"     "laboratorio"
## [1] "35"          "-0.0263"     "0.0055"      "year_adm_t0"
## [1] "34"                    "-0.0265"             "-2e-04"
## [4] "meds_antimicrobianos"
## [1] "33"            "-0.0254"       "0.0011"        "psicofarmacos"
## [1] "32"      "-0.0412" "-0.0158" "icu_t0"
## [1] "32"                     "-0.0412"               "0.041"
## [4] "admission_pre_t0_count"
## [1] "32"                    "-0.0412"               "0.002"
## [4] "metodos_graficos_qtde"
## [1] "32"            "-0.0412"       "0.0083"        "hospital_stay"
## [1] "31"      "-0.0571" "-0.0159" "age"
```

```
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	64	0.7267	0.0320	0.0000	0.0000
All unimportant	TRUE	56	0.7302	0.0286	-0.0035	-0.0035
vni	TRUE	55	0.7302	0.0286	-0.0035	0.0000
histopatologia_qtde	TRUE	54	0.7298	0.0305	-0.0031	0.0004
ultrassom	TRUE	53	0.7341	0.0281	-0.0074	-0.0044
antifungico	TRUE	52	0.7341	0.0284	-0.0074	0.0001
cied_final_group_1	FALSE	51	0.7308	0.0307	-0.0074	0.0033
angio_tc	TRUE	51	0.7365	0.0280	-0.0098	-0.0024
procedure_type_new	TRUE	50	0.7382	0.0274	-0.0115	-0.0017
heart_failure	TRUE	49	0.7381	0.0293	-0.0114	0.0001
reop_type_1	TRUE	48	0.7367	0.0279	-0.0100	0.0013
ventilacao_mecanica	TRUE	47	0.7379	0.0281	-0.0112	-0.0011
flebografia	TRUE	46	0.7419	0.0270	-0.0152	-0.0040
admission_pre_t0_180d	FALSE	45	0.7237	0.0360	-0.0152	0.0182
heart_disease	TRUE	45	0.7424	0.0269	-0.0157	-0.0005
insulina	TRUE	44	0.7412	0.0277	-0.0145	0.0012
citologia	FALSE	43	0.7346	0.0315	-0.0145	0.0066
sex	FALSE	43	0.7386	0.0261	-0.0145	0.0027
diabetes	TRUE	43	0.7425	0.0268	-0.0158	-0.0013
ecocardiograma	FALSE	42	0.7354	0.0283	-0.0158	0.0071
betabloqueador	FALSE	42	0.7375	0.0279	-0.0158	0.0051
procedure_type_1	FALSE	42	0.7397	0.0250	-0.0158	0.0029
af	FALSE	42	0.7263	0.0263	-0.0158	0.0162
tomografia	FALSE	42	0.7326	0.0315	-0.0158	0.0099
cultura	FALSE	42	0.7353	0.0274	-0.0158	0.0072
proced_invasivos_qtde	FALSE	42	0.7405	0.0248	-0.0158	0.0020

Table 1: *(continued)*

Tested Feature	Dropped	Features	CV AUC	CV AUC Std Error	Total AUC Loss	Instant AUC Loss
angiografia	TRUE	42	0.7437	0.0252	-0.0170	-0.0012
interconsulta	FALSE	41	0.7396	0.0251	-0.0170	0.0041
antihipertensivo	TRUE	41	0.7456	0.0248	-0.0189	-0.0019
dva	FALSE	40	0.7409	0.0235	-0.0189	0.0047
ressonancia	FALSE	40	0.7213	0.0362	-0.0189	0.0243
renal_failure	FALSE	40	0.7093	0.0377	-0.0189	0.0363
hypertension	FALSE	40	0.7326	0.0358	-0.0189	0.0130
cied_final_1	FALSE	40	0.7407	0.0264	-0.0189	0.0049
classe_meds_qtde	FALSE	40	0.7428	0.0231	-0.0189	0.0028
comorbidities_count	TRUE	40	0.7451	0.0245	-0.0183	0.0006
diuretico	FALSE	39	0.7301	0.0346	-0.0183	0.0150
holter	FALSE	39	0.7370	0.0275	-0.0183	0.0080
exames_imagem_qtde	TRUE	39	0.7470	0.0279	-0.0203	-0.0019
antiarritmico	TRUE	38	0.7505	0.0254	-0.0238	-0.0035
nyha_basal	FALSE	37	0.7284	0.0391	-0.0238	0.0220
equipe_multiprof	FALSE	37	0.7464	0.0338	-0.0238	0.0040
bic	FALSE	37	0.7366	0.0263	-0.0238	0.0139
analises_clinicas_qtde	FALSE	37	0.7454	0.0285	-0.0238	0.0050
underlying_heart_disease	TRUE	37	0.7526	0.0249	-0.0259	-0.0021
espironolactona	TRUE	36	0.7508	0.0286	-0.0240	0.0018
meds_cardiovasc_qtde	FALSE	35	0.7480	0.0246	-0.0240	0.0028
vasodilatador	FALSE	35	0.7345	0.0333	-0.0240	0.0162
education_level	FALSE	35	0.7245	0.0303	-0.0240	0.0262
laboratorio	TRUE	35	0.7530	0.0259	-0.0263	-0.0022
year_adm_t0	FALSE	34	0.7475	0.0227	-0.0263	0.0055
meds_antimicrobianos	TRUE	34	0.7532	0.0287	-0.0265	-0.0002
psicofarmacos	TRUE	33	0.7521	0.0299	-0.0254	0.0011
icu_t0	TRUE	32	0.7679	0.0265	-0.0412	-0.0158
admission_pre_t0_count	FALSE	31	0.7269	0.0245	-0.0412	0.0410
metodos_graficos_qtde	FALSE	31	0.7659	0.0237	-0.0412	0.0020
hospital_stay	FALSE	31	0.7596	0.0252	-0.0412	0.0083
age	TRUE	31	0.7838	0.0211	-0.0571	-0.0159

```
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)
```

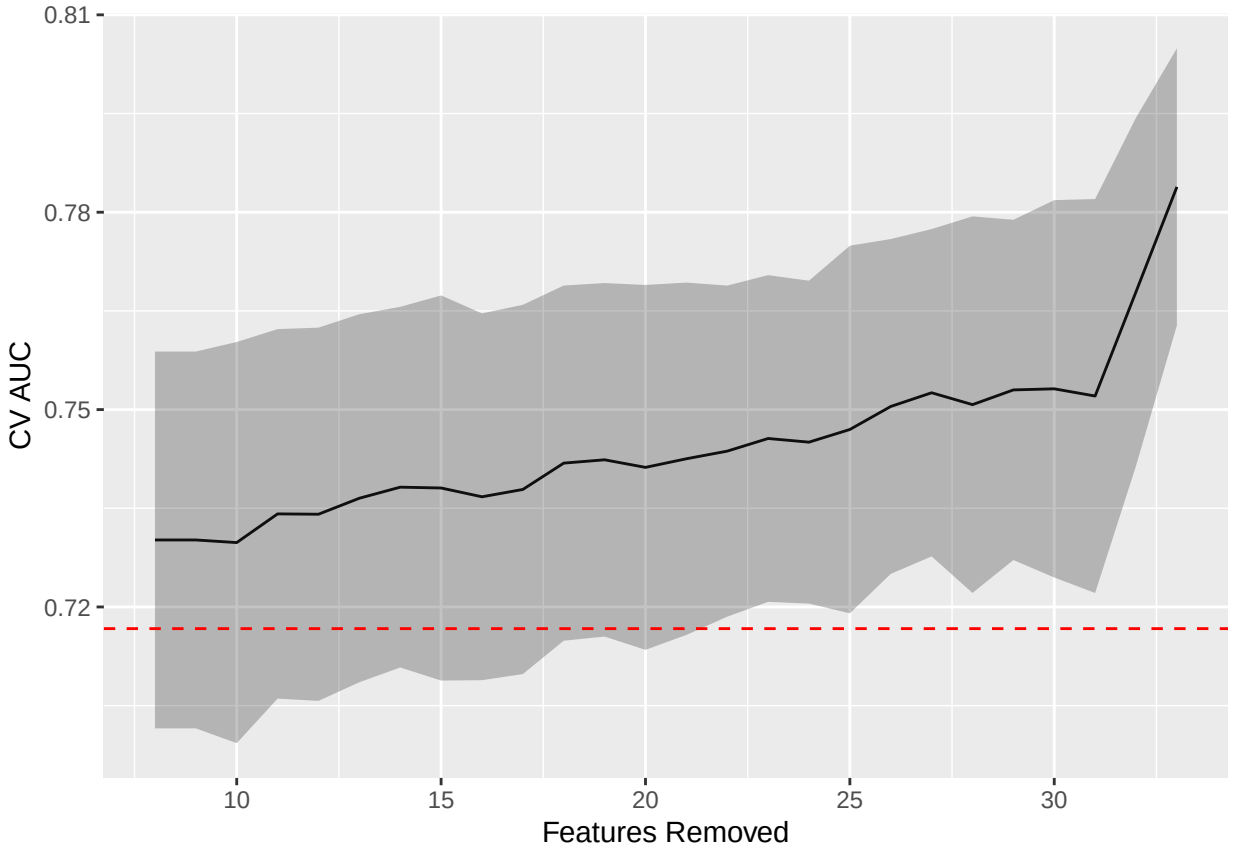
```
## [1] "Selected Model CV Train AUC: 0.784"
```

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

```
## [1] "Selected Model Test AUC: 0.727"
```

```
 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)
```

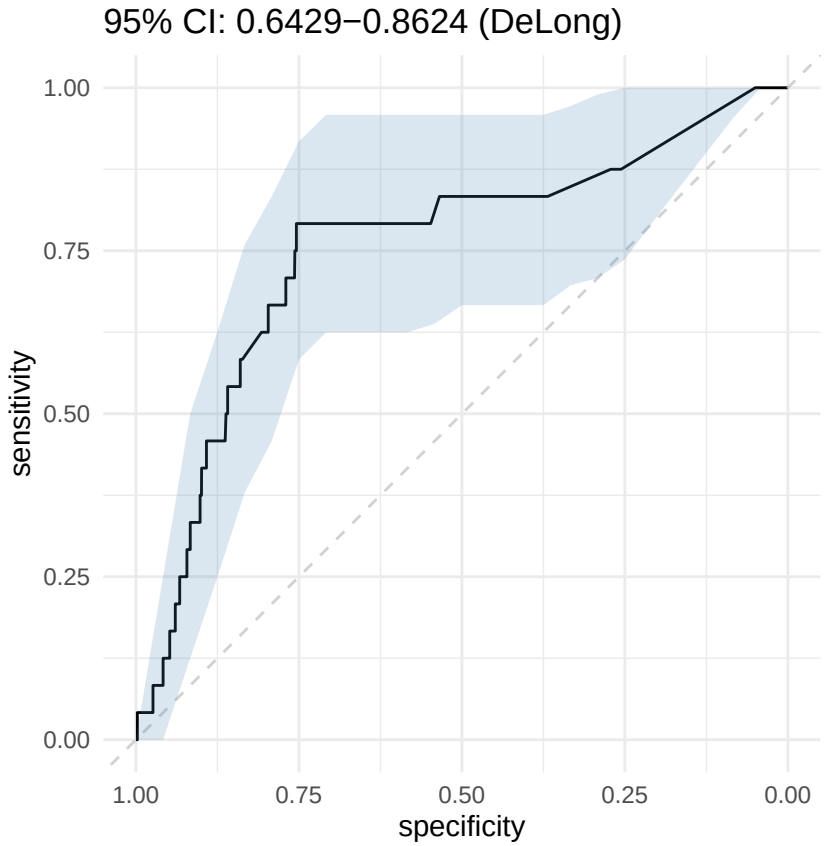
1. hospital_stay
2. admission_pre_t0_count
3. year_adm_t0
4. metodos_graficos_qtde
5. education_level
6. vasodilatador
7. analises_clinicas_qtde
8. equipe_multiprof
9. meds_cardiovasc_qtde
10. bic
11. nyha_basal
12. holter
13. hypertension
14. citologia
15. cied_final_1
16. dva
17. classe_meds_qtde
18. ressonancia

19. renal_failure
20. diuretico
21. interconsulta
22. cultura
23. betabloqueador
24. af
25. proced_invasivos_qtde
26. tomografia
27. sex
28. procedure_type_1
29. ecocardiograma
30. admission_pre_t0_180d
31. cied_final_group_1

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)

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

```r
  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: 0.00"
## Confusion Matrix and Statistics
##
##     reference
## data    0    1
```

```
##    0 3548    5
##    1 1158   19
##
##                Accuracy : 0.7541
##                  95% CI : (0.7416, 0.7663)
##     No Information Rate : 0.9949
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0219
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.75393
##             Specificity : 0.79167
##          Pos Pred Value : 0.99859
##          Neg Pred Value : 0.01614
##              Prevalence : 0.99493
##          Detection Rate : 0.75011
##    Detection Prevalence : 0.75116
##       Balanced Accuracy : 0.77280
##
##        'Positive' Class : 0
##
```

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

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

# 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))
```

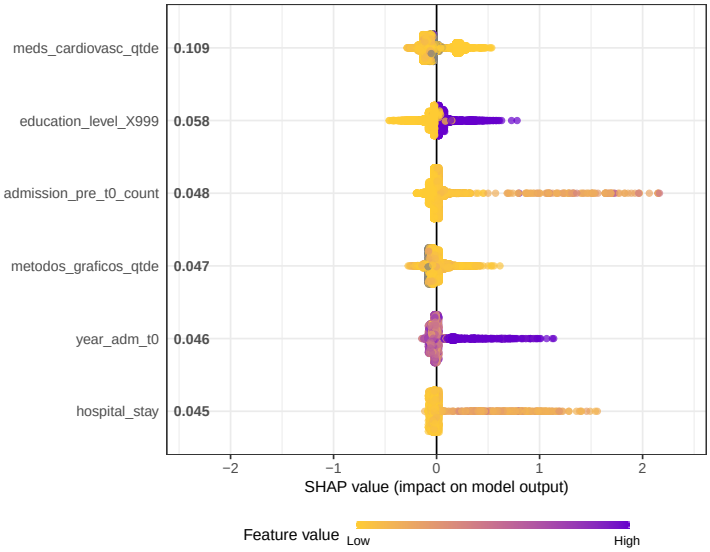
SHAP values

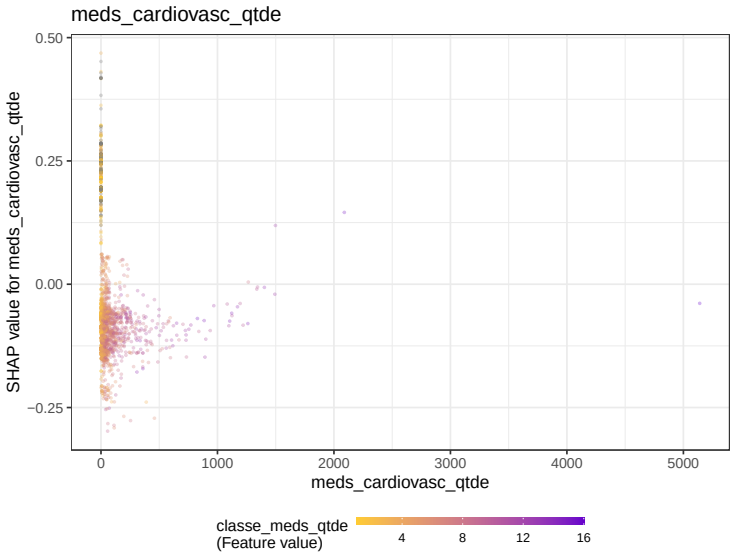
```r
lightgbm_model <- parsnip::extract_fit_engine(final_lightgbm_fit)

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))
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)
```

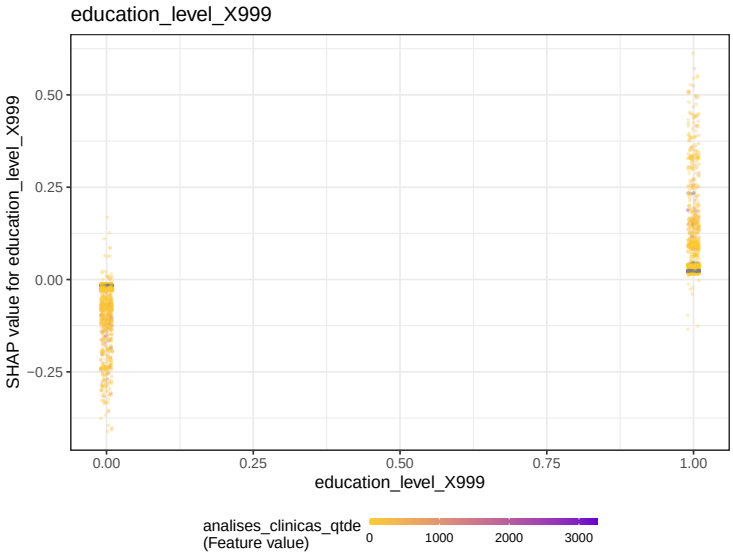
```r
for (x in shap.importance(shap, names_only = TRUE)) {
  p <- shap.plot.dependence(
    shap,
    x = x,
    color_feature = "auto",
    smooth = FALSE,
    jitter_width = 0.01,
    alpha = 0.3
  ) +
    labs(title = x)

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

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

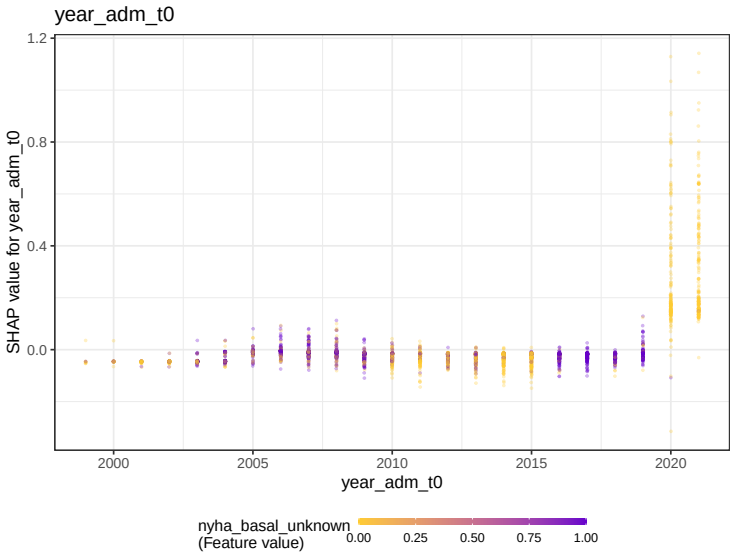
```
## Warning: Removed 1055 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 1055 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 7 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 7 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 1055 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 1055 rows containing missing values (‘geom_point()‘).
## Warning: Removed 822 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 822 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 822 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Warning: Removed 822 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 1468 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Warning: Removed 822 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
## Warning: Removed 822 rows containing missing values (‘geom_point()‘).
```

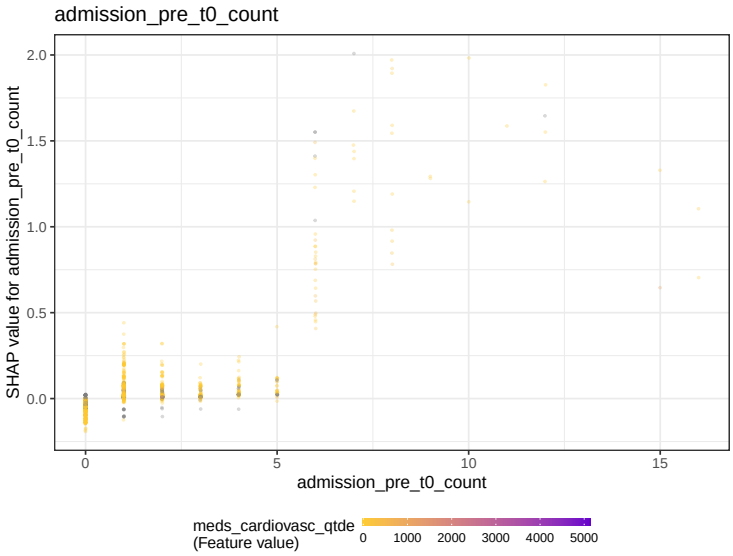
```
## Saving 6.5 x 5 in image
## Warning: Removed 822 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 822 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 822 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Warning: Removed 1055 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 1055 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 822 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 822 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Warning: Removed 1077 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Warning: Removed 1055 rows containing missing values (‘geom_point()‘).
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Warning: Removed 822 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
## 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
```

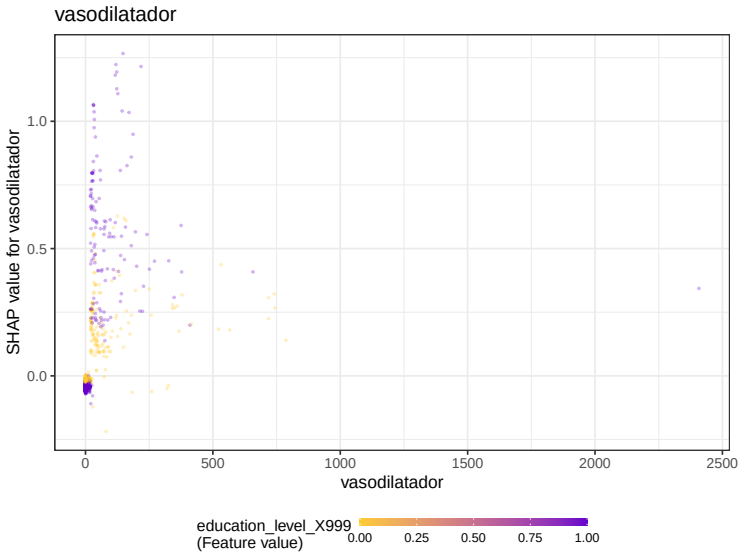

meds_cardiovasc_qtde
0.109
education_level_X999
0.058
admission_pre_t0_count
0.048
metodos_graficos_qtde
0.047
year_adm_t0
0.046
hospital_stay
0.045
SHAP value (impact on model output)
Feature value
Low
High

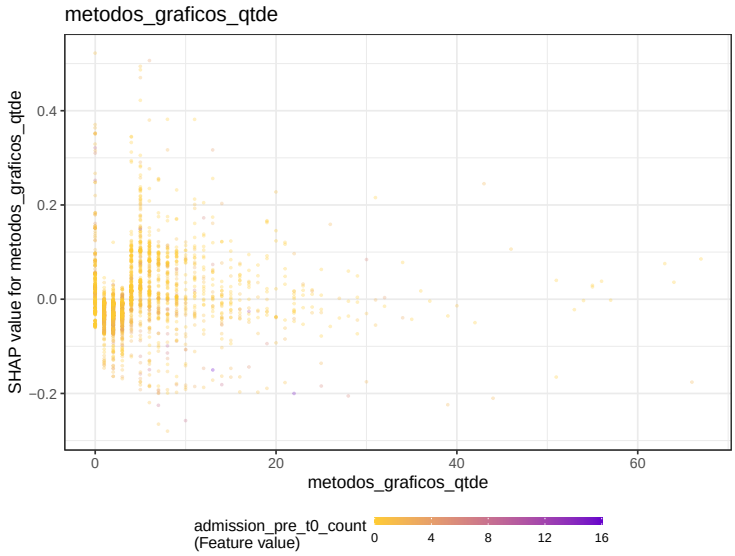

meds_cardiovasc_qtde
SHAP value for meds_cardiovasc_qtde
meds_cardiovasc_qtde
classe_meds_qtde
(Feature value)


education_level_X999
SHAP value for education_level_X999
education_level_X999
analises_clinicas_qtde
(Feature value)


year_adm_t0
SHAP value for year_adm_t0
year_adm_t0
nyha_basal_unknown
(Feature value)


admission_pre_t0_count
SHAP value for admission_pre_t0_count
admission_pre_t0_count
meds_cardiovasc_qtde
(Feature value)


vasodilatador
SHAP value for vasodilatador
vasodilatador
education_level_X999
(Feature value)

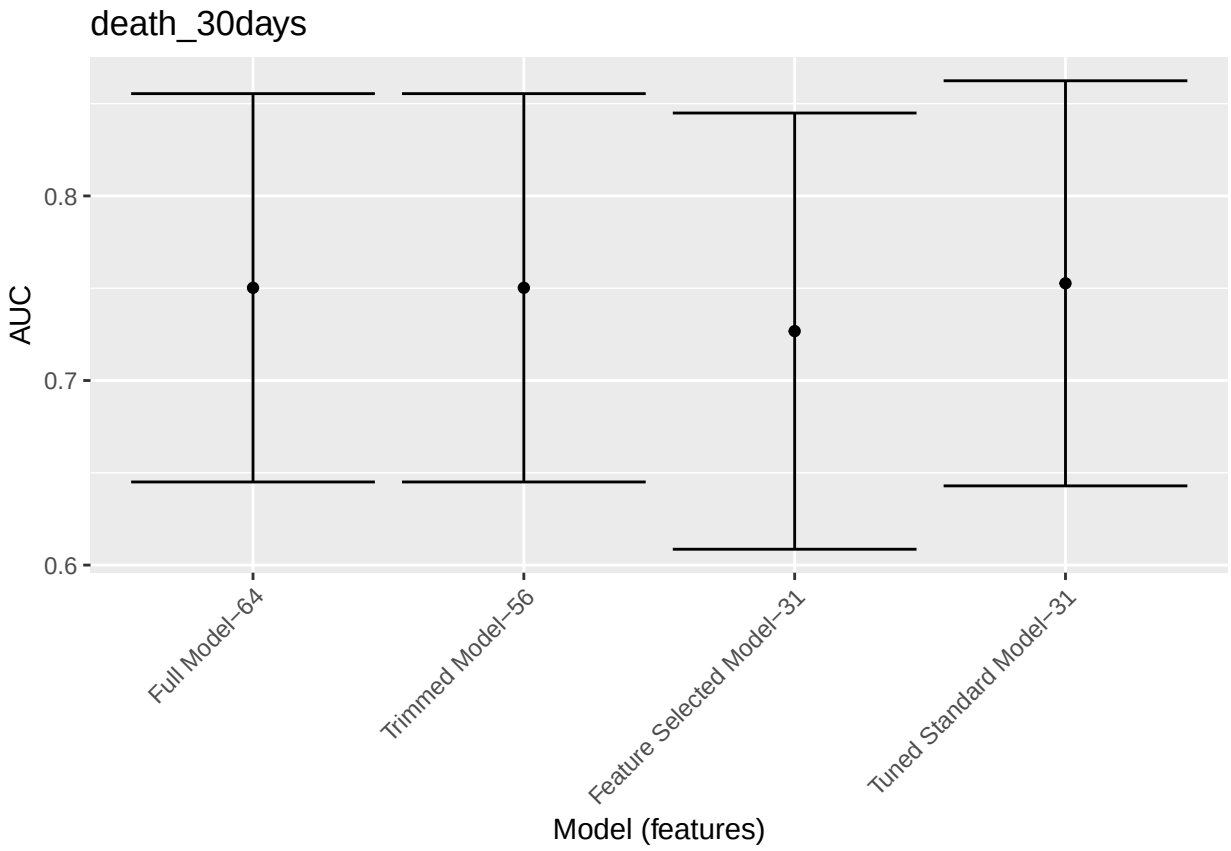



```
## $num_iterations
## [1] 54
##
## $learning_rate
## [1] 0.009830575
##
## $max_depth
## [1] 6
##
## $feature_fraction_bynode
## [1] 1
##
## $min_data_in_leaf
## [1] 38
##
## $min_gain_to_split
## [1] 0
##
## $bagging_fraction
## [1] 1
##
## $num_class
## [1] 1
##
## $objective
## [1] "binary"
##
## $num_threads
## $num_threads$num_threads
## [1] 0
##
##
## $nthread
## [1] 8
##
## $seed
## [1] 76620
##
## $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))
```




```
write_csv(df_auc, sprintf("./auxiliar/final_model/performance/%s.csv", outcome_column))
```