

Final Model - readmission_30d

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. education_level
2. underlying_heart_disease
3. heart_disease
4. nyha_basal
5. prior_mi
6. heart_failure
7. transplant
8. endocardites
9. hemodialysis
10. comorbidities_count
11. procedure_type_1
12. reop_type_1

13. procedure_type_new
14. cied_final_1
15. cied_final_group_1
16. admission_pre_t0_count
17. admission_pre_t0_180d
18. icu_t0
19. dialysis_t0
20. admission_t0_emergency
21. aco
22. antiaritmico
23. betabloqueador
24. ieca_bra
25. dva
26. digoxina
27. estatina
28. diuretico
29. vasodilatador
30. insuf_cardiaca
31. espironolactona
32. bloq_calcio
33. antiplaquetario_ev
34. insulin
35. anticonvulsivante
36. psicofarmacos
37. antifungico
38. antiviral
39. classe_meds_qtd
40. meds_cardiovasc_qtd
41. meds_antimicrobianos
42. ventilacao_mecanica
43. cec
44. transplante_cardiaco
45. outros_proced_cirurgicos
46. icp
47. intervencao_cv
48. cateterismo
49. eletrofisiologia
50. cateter_venoso_central
51. proced_invasivos_qtd
52. cve_desf
53. transfusao
54. equipe_multiprof
55. holter
56. metodos_graficos_qtd
57. laboratorio
58. cultura
59. analises_clinicas_qtd
60. citologia
61. biopsia
62. histopatologia_qtd
63. angio_rm
64. angio_tc
65. cintilografia
66. ecocardiograma
67. endoscopia
68. flebografia
69. pet_ct
70. ultrassom
71. tomografia
72. ressonancia
73. exames_imagem_qtd

74. bic
75. mpp
76. 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: 0.682"
sprintf('Full Model Test AUC: %.3f' ,full_model$auc)

## [1] "Full Model Test AUC: 0.698"

Features with zero importance on the initial model:
unimportant_features <- setdiff(features, full_model$importance$Feature)

unimportant_features %>%
  gluedown::md_order()

1. transplant
2. endocardites
3. hemodialysis
4. dialysis_t0
5. antiviral
6. transplante_cardiaco
7. icp
8. intervencao_cv
9. citologia
10. endoscopia
11. 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: 0.681"
sprintf('Trimmed Model Test AUC: %.3f' ,trimmed_model$auc)

## [1] "Trimmed Model Test AUC: 0.697"

selection_results <- tibble::tribble(
  ~`Tested Feature`, ~`Dropped`, ~`Number of Features`, ~`CV AUC`, ~`CV AUC Std Error`, ~`Total AUC Loss`, ~`In
  '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] "64"           "0.0013"          "-2e-04"          "education_level"
## [1] "64"           "0.0048"          "0.0035"          "biopsia"
## [1] "63"           "0.0032"          "0.0019"          "heart_disease"
## [1] "62"           "0.0023"          "-0.001"          "cec"
## [1] "61"           "0.0022"          "0"              "transfusao"
## [1] "60"           ""                "0.0014"
## [3] "-9e-04"       ""                "cateter_venoso_central"
## [1] "59"           "3e-04"          "-0.0011"         "angio_tc"
## [1] "58"           "0.0021"          "0.0018"
## [4] "antiplaquetario_ev"
## [1] "57"           "0.0019"          "-2e-04"          "insulina"
## [1] "56"           ""                "0.002"
## [3] "0"             ""                "underlying_heart_disease"
## [1] "55"           "-0.0013"         "-0.0033"         "cve_desf"
## [1] "55"           "0.0015"          "0.0028"          "nyha_basal"
## [1] "55"           "0.0039"          "0.0053"
## [4] "eletrofisiologia"
## [1] "54"           "0"              "0.0013"          "bloq_calcio"
## [1] "54"           "0.0026"          "0.0027"          "flebografia"
## [1] "53"           "0.0017"          "0.0018"          "ressonancia"
## [1] "52"           "0.0019"          "1e-04"          "mpp"
## [1] "51"           "0.0031"          "0.0012"          "cateterismo"
## [1] "50"           "0.0022"          "-9e-04"          "cintilografia"
## [1] "49"           "0.0012"          "-0.001"
## [4] "procedure_type_1"
## [1] "48"           ""                "0.0017"          "5e-04"
## [4] "histopatologia_qtde"
## [1] "48"           "0.004"          "0.0023"          "prior_mi"
## [1] "47"           ""                "8e-04"
## [3] "-9e-04"       ""                "outros_proced_cirurgicos"
## [1] "47"           "0.0035"          "0.0027"          "heart_failure"
## [1] "46"           "-4e-04"         "-0.0012"         "cied_final_1"
## [1] "45"           "3e-04"          "7e-04"          "antifungico"
## [1] "44"           "0.0019"          "0.0016"          "cultura"
## [1] "43"           "0.0036"          "0.0017"          "tomografia"
## [1] "42"           "0.0036"          "1e-04"          "reop_type_1"
## [1] "41"           "0.0027"          "-0.001"          "betabloqueador"
## [1] "40"           ""                "0.002"          "-7e-04"
## [4] "ventilacao_mecanica"
## [1] "40"           "0.0052"          "0.0032"          "holter"
## [1] "39"           "0.0028"          "9e-04"          "ultrassom"
## [1] "38"           "0.0029"          "1e-04"          "aco"
## [1] "37"           "0.0022"          "-7e-04"          "ecocardiograma"
## [1] "37"           ""                "0.0073"          "0.0051"
## [4] "procedure_type_new"
## [1] "36"           ""                "0.0028"          "6e-04"
## [4] "cied_final_group_1"
## [1] "35"           "0.0037"          "9e-04"          "digoxina"
## [1] "34"           ""                "0.0047"          "0.001"
## [4] "admission_pre_t0_180d"
## [1] "33"           ""                "0.004"
## [3] "-7e-04"       ""                "admission_t0_emergency"
## [1] "32"           "0.0054"          "0.0013"          "espironolactona"
## [1] "31"           ""                "0.0043"          "-0.0011"
## [4] "proced_invasivos_qtde"
## [1] "30"           "0.0026"          "-0.0017"         "angio_rm"
## [1] "30"           ""                "0.005"          "0.0024"
## [4] "exames_imagem_qtde"
## [1] "29"           "0.0026"          "0"              "estatina"
## [1] "29"           ""                "0.0053"          "0.0027"
## [4] "equipe_multiprof"

```

```

## [1] "28"           "0.0028"        "2e-04"          "insuf_cardiaca"
## [1] "28"           "0.0048"        "0.002"         "dva"
## [1] "28"           ""              "0.0068"        "0.004"
## [4] "comorbidities_count"
## [1] "27"           "8e-04"         "-0.002"        "ieca_bra"
## [1] "27"           ""              "0.003"         "0.0022"
## [4] "anticonvulsivante"
## [1] "27"           ""              "0.0041"        "0.0032"
## [4] "classe_meds_qtde"
## [1] "26"           "0.0026"        "0.0018"        "psicofarmacos"
## [1] "25"           "0.0025"        "-1e-04"        "laboratorio"
## [1] "24"           "0.0039"        "0.0014"        "diuretico"
## [1] "23"           "0.0026"        "-0.0013"       "antiarritmico"
## [1] "22"           "0.0027"        "1e-04"         "bic"
## [1] "22"           ""              "0.0077"        "0.005"
## [1] "22"           ""              "0.008"         "0.0053"
## [4] "metodos_graficos_qtde"
## [1] "21"           ""              "0.004"         "0.0013"
## [4] "meds_antimicrobianos"
## [1] "21"           ""              "0.0083"
## [3] "0.0043"       "analises_clinicas_qtde"
## [1] "21"           "0.0129"        "0.009"         "icu_t0"

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	76	0.6822	0.0068	0.0000	0.0000
All unimportant	TRUE	65	0.6807	0.0082	0.0015	0.0015
education_level	TRUE	64	0.6809	0.0080	0.0013	-0.0002
biopsia	FALSE	63	0.6774	0.0064	0.0048	0.0035
heart_disease	TRUE	63	0.6790	0.0086	0.0032	0.0019
cec	TRUE	62	0.6799	0.0077	0.0023	-0.0010
transfusao	TRUE	61	0.6800	0.0086	0.0022	0.0000
cateter Venoso Central	TRUE	60	0.6808	0.0069	0.0014	-0.0009
angio_tc	TRUE	59	0.6819	0.0078	0.0003	-0.0011
antiplaquetario_ev	TRUE	58	0.6801	0.0075	0.0021	0.0018
insulina	TRUE	57	0.6803	0.0075	0.0019	-0.0002
underlying_heart_disease	TRUE	56	0.6802	0.0073	0.0020	0.0000
cve_desf	TRUE	55	0.6835	0.0082	-0.0013	-0.0033
nyha_basal	FALSE	54	0.6807	0.0073	0.0015	0.0028
eletrofisiologia	FALSE	54	0.6783	0.0082	0.0039	0.0053
bloq_calcio	TRUE	54	0.6822	0.0070	0.0000	0.0013
flebografia	FALSE	53	0.6796	0.0078	0.0026	0.0027
ressonancia	TRUE	53	0.6805	0.0074	0.0017	0.0018
mpp	TRUE	52	0.6803	0.0076	0.0019	0.0001
cateterismo	TRUE	51	0.6791	0.0080	0.0031	0.0012
cintilografia	TRUE	50	0.6800	0.0080	0.0022	-0.0009
procedure_type_1	TRUE	49	0.6810	0.0078	0.0012	-0.0010
histopatologia_qtde	TRUE	48	0.6805	0.0075	0.0017	0.0005
prior_mi	FALSE	47	0.6782	0.0072	0.0040	0.0023
outros_proced_cirurgicos	TRUE	47	0.6814	0.0067	0.0008	-0.0009
heart_failure	FALSE	46	0.6787	0.0069	0.0035	0.0027
cied_final_1	TRUE	46	0.6826	0.0077	-0.0004	-0.0012
antifungico	TRUE	45	0.6819	0.0079	0.0003	0.0007
cultura	TRUE	44	0.6803	0.0073	0.0019	0.0016

Table 1: (continued)

Tested Feature	Dropped	Features	CV AUC	CV AUC Std Error	Total AUC Loss	Instant AUC Loss
tomografia	TRUE	43	0.6786	0.0080	0.0036	0.0017
reop_type_1	TRUE	42	0.6786	0.0069	0.0036	0.0001
betabloqueador	TRUE	41	0.6795	0.0064	0.0027	-0.0010
ventilacao_mecanica	TRUE	40	0.6802	0.0068	0.0020	-0.0007
holter	FALSE	39	0.6770	0.0072	0.0052	0.0032
ultrassom	TRUE	39	0.6794	0.0073	0.0028	0.0009
aco	TRUE	38	0.6793	0.0063	0.0029	0.0001
ecocardiograma	TRUE	37	0.6800	0.0063	0.0022	-0.0007
procedure_type_new	FALSE	36	0.6749	0.0049	0.0073	0.0051
cied_final_group_1	TRUE	36	0.6794	0.0076	0.0028	0.0006
digoxina	TRUE	35	0.6785	0.0067	0.0037	0.0009
admission_pre_t0_180d	TRUE	34	0.6775	0.0073	0.0047	0.0010
admission_t0_emergency	TRUE	33	0.6782	0.0055	0.0040	-0.0007
espironolactona	TRUE	32	0.6768	0.0081	0.0054	0.0013
proced_invasivos_qtde	TRUE	31	0.6779	0.0066	0.0043	-0.0011
angio_rm	TRUE	30	0.6796	0.0069	0.0026	-0.0017
exames_imagem_qtde	FALSE	29	0.6772	0.0064	0.0050	0.0024
estatina	TRUE	29	0.6796	0.0068	0.0026	0.0000
equipe_multiprof	FALSE	28	0.6769	0.0061	0.0053	0.0027
insuf_cardiaca	TRUE	28	0.6794	0.0078	0.0028	0.0002
dva	FALSE	27	0.6774	0.0071	0.0048	0.0020
comorbidities_count	FALSE	27	0.6754	0.0090	0.0068	0.0040
ieca_bra	TRUE	27	0.6814	0.0065	0.0008	-0.0020
anticonvulsivante	FALSE	26	0.6792	0.0072	0.0030	0.0022
classe_meds_qtde	FALSE	26	0.6781	0.0074	0.0041	0.0032
psicofarmacos	TRUE	26	0.6796	0.0072	0.0026	0.0018
laboratorio	TRUE	25	0.6797	0.0078	0.0025	-0.0001
diuretico	TRUE	24	0.6783	0.0087	0.0039	0.0014
antiarritmico	TRUE	23	0.6796	0.0074	0.0026	-0.0013
bic	TRUE	22	0.6795	0.0059	0.0027	0.0001
vasodilatador	FALSE	21	0.6745	0.0064	0.0077	0.0050
metodos_graficos_qtde	FALSE	21	0.6742	0.0048	0.0080	0.0053
meds_antimicrobianos	TRUE	21	0.6782	0.0059	0.0040	0.0013
analises_clinicas_qtde	FALSE	20	0.6739	0.0060	0.0083	0.0043
icu_t0	FALSE	20	0.6693	0.0083	0.0129	0.0090

```

if (exists('last_feature_dropped')) {
  selected_features <- c(current_features, last_feature_dropped)
} 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: 0.678"

```

```

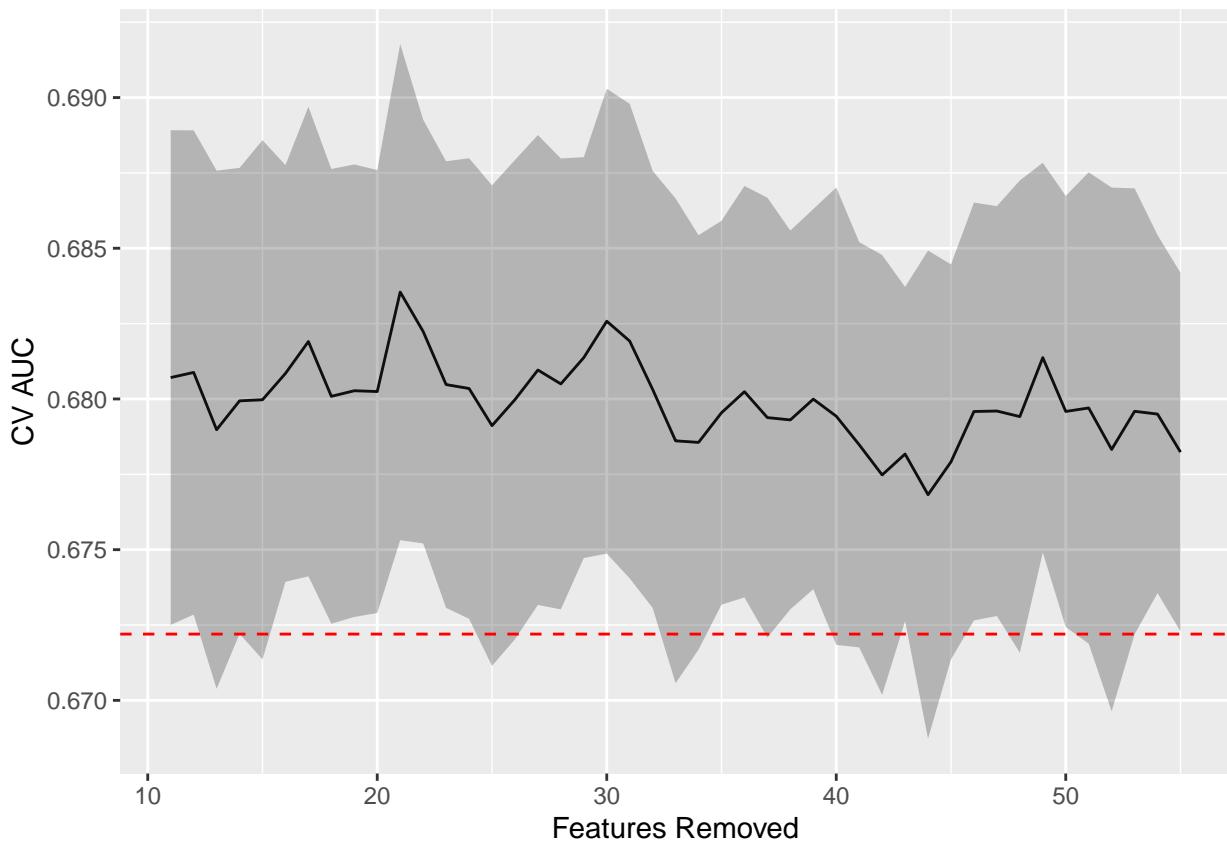
sprintf('Selected Model Test AUC: %.3f', feature_selected_model$auc)

## [1] "Selected Model Test AUC: 0.690"

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. analises_clinicas_qtde
3. vasodilatador
4. classe_meds_qtde
5. metodos_graficos_qtde
6. exames_imagem_qtde
7. icu_t0
8. meds_cardiovasc_qtde
9. admission_pre_t0_count

10. dva
11. equipe_multiprof
12. anticonvulsivante
13. holter
14. comorbidities_count
15. procedure_type_new
16. prior_mi
17. heart_failure
18. eletrofisiologia
19. biopsia
20. flebografia
21. nyha_basal

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))

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)

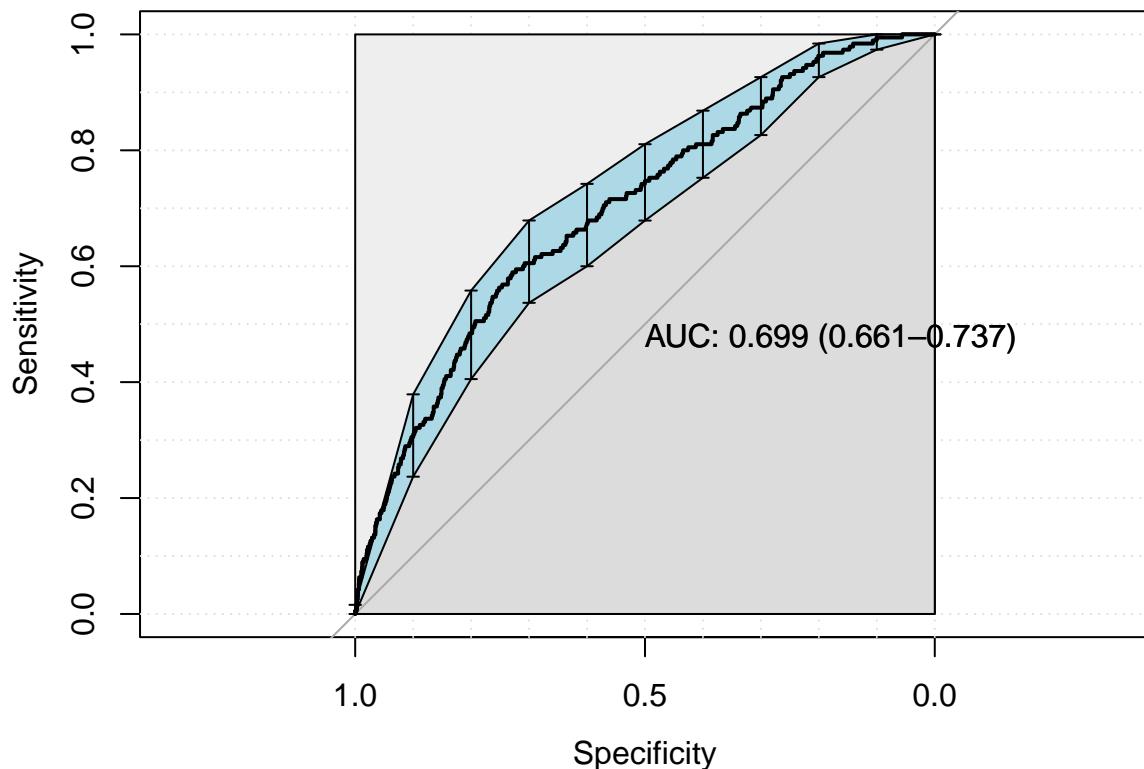
lightgbm_auc <- validation(final_lightgbm_fit, df_test)

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: 0.04"
## Confusion Matrix and Statistics
##
##      reference
## data      0      1
##   0 3309    78
##   1 1231   112
##
##                  Accuracy : 0.7233
##                         95% CI : (0.7103, 0.736)
##      No Information Rate : 0.9598
##      P-Value [Acc > NIR] : 1
##
##                  Kappa : 0.0815
##
## Mcnemar's Test P-Value : <2e-16
##
##      Sensitivity : 0.7289
##      Specificity : 0.5895
##      Pos Pred Value : 0.9770
##      Neg Pred Value : 0.0834
##      Prevalence : 0.9598
##      Detection Rate : 0.6996
##      Detection Prevalence : 0.7161
##      Balanced Accuracy : 0.6592
##
##      '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)

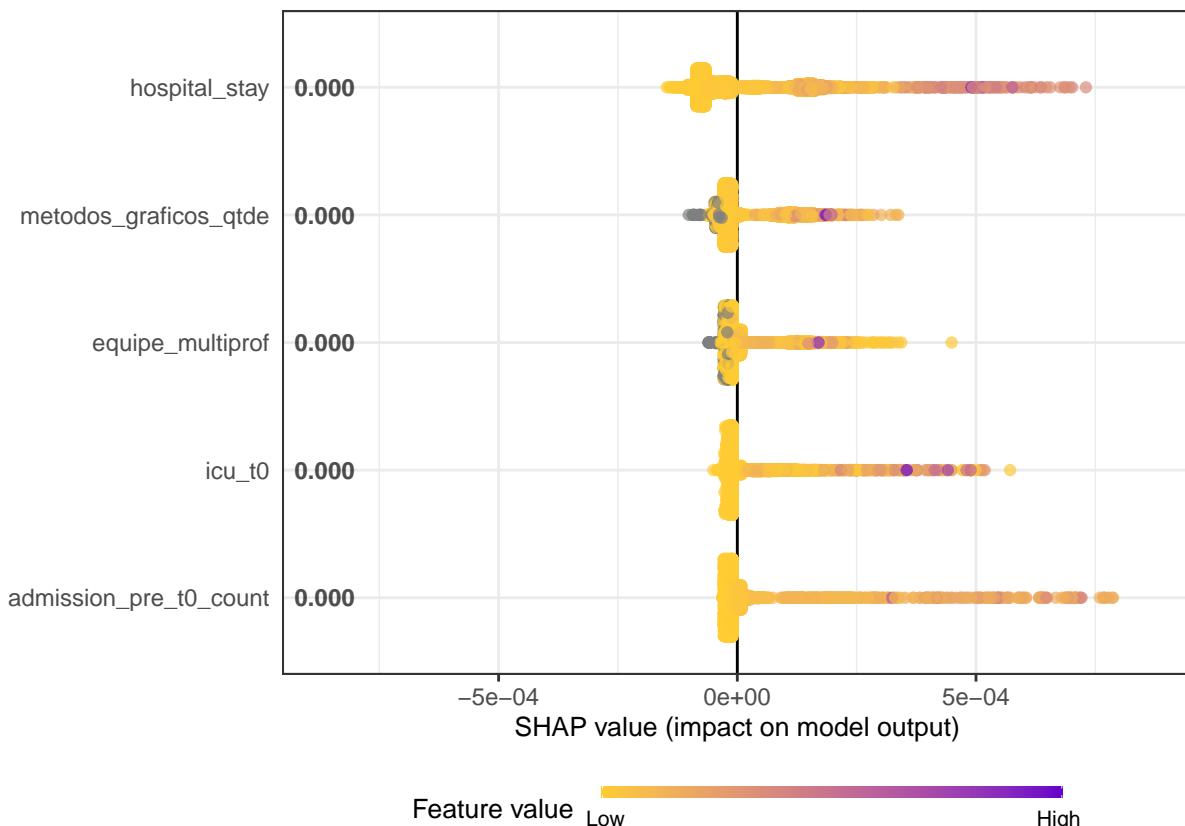
df_train_baked <- bake(trained_rec, new_data = df_train)
df_test_baked <- bake(trained_rec, new_data = df_test)

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",

```

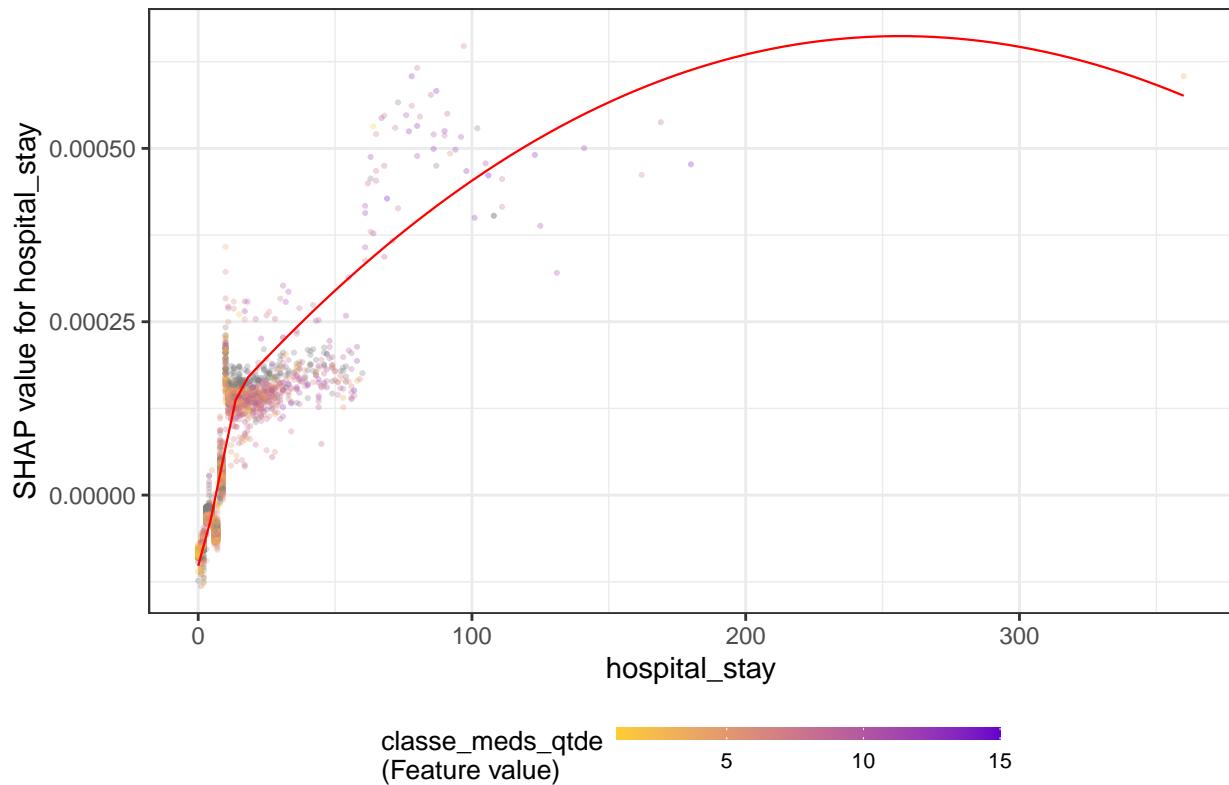
```

    smooth = TRUE,
    jitter_width = 0.01,
    alpha = 0.3
) +
  labs(title = x)
print(p)
}

```

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

hospital_stay

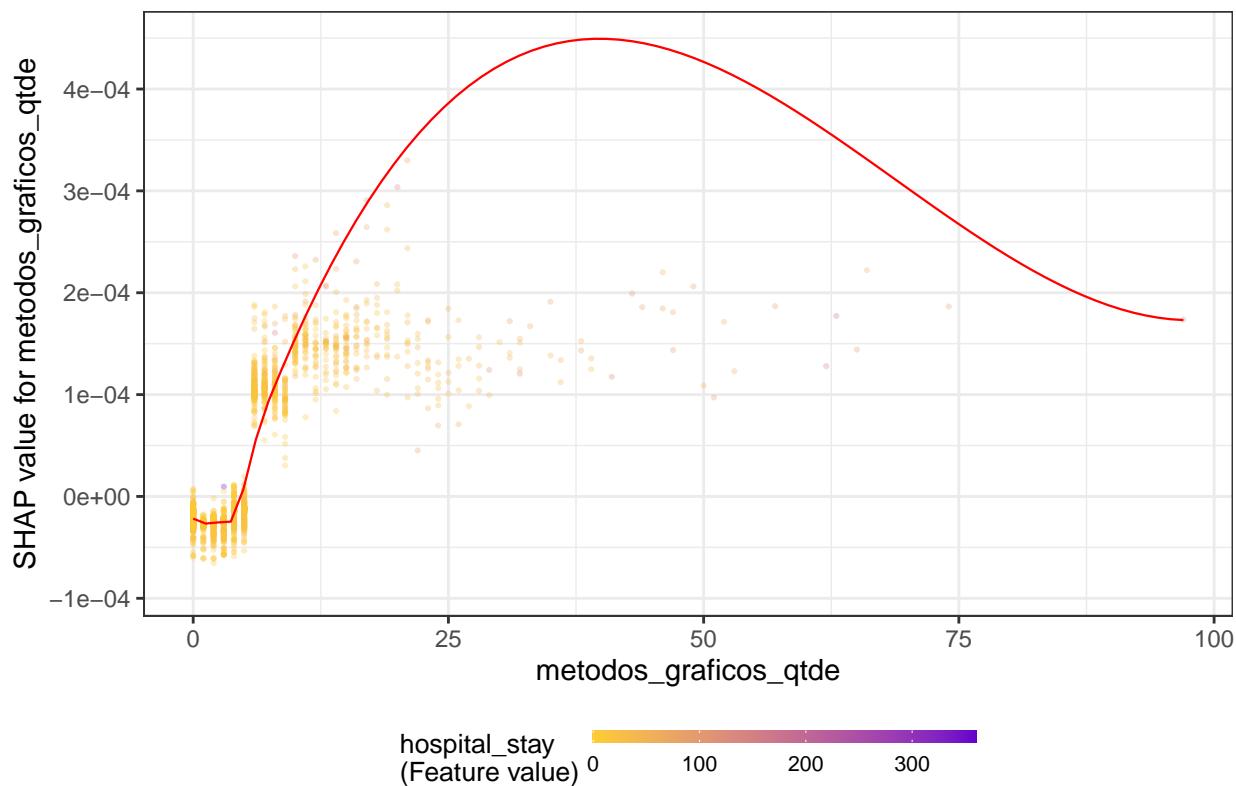


```

## `geom_smooth()` using formula 'y ~ x'
## Warning: Removed 825 rows containing non-finite values (stat_smooth).
## Warning: Removed 825 rows containing missing values (geom_point).

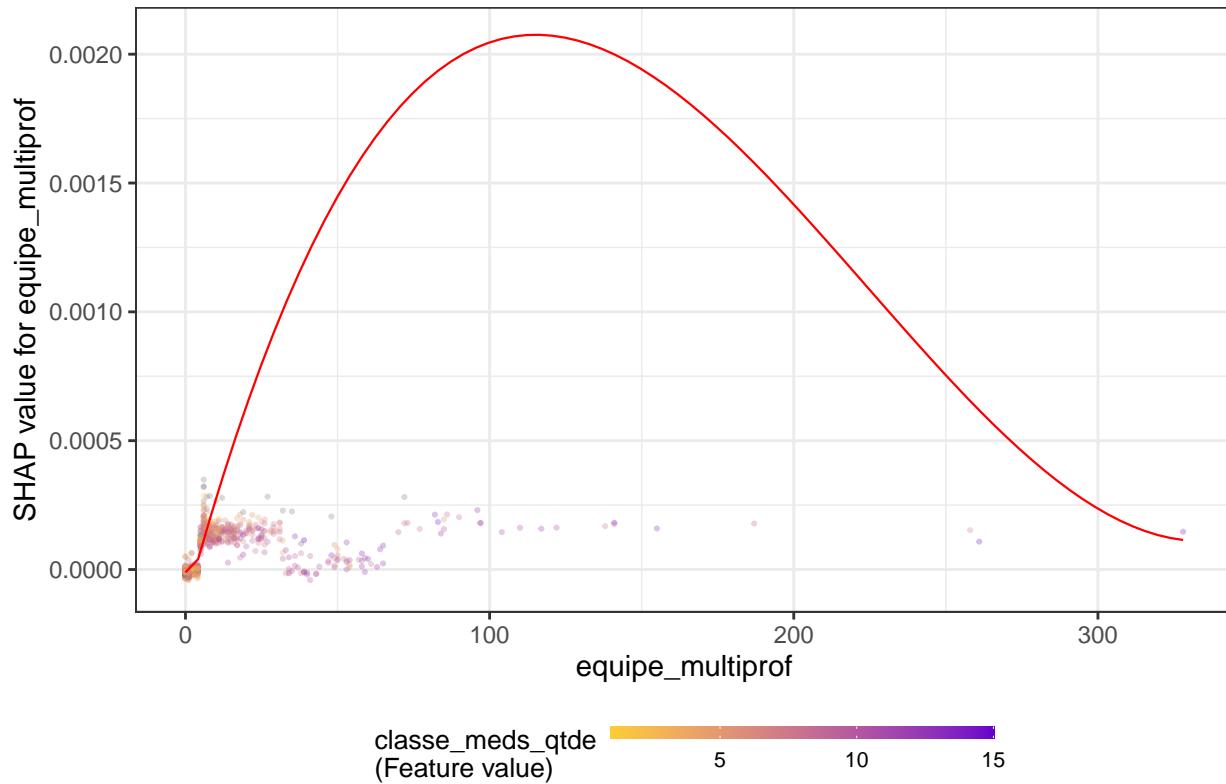
```

metodos_graficos_qtde

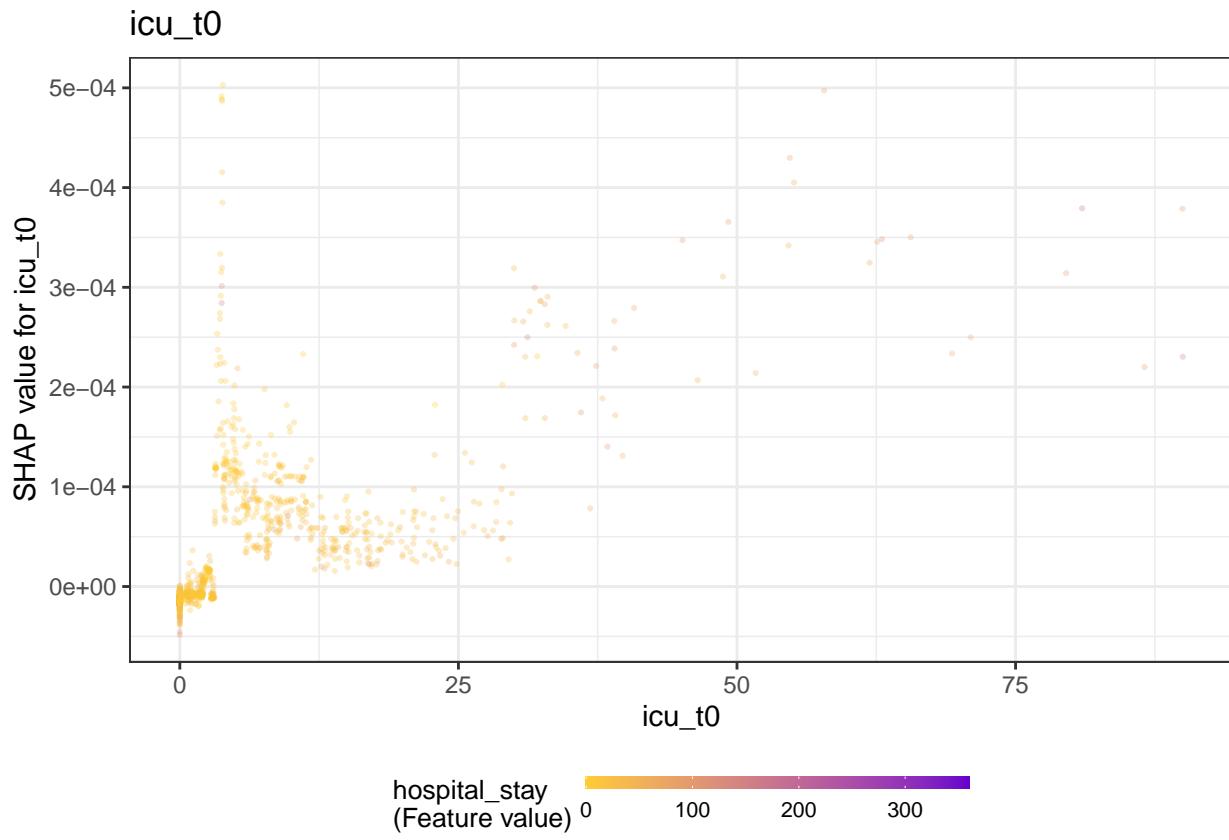


```
## `geom_smooth()` using formula 'y ~ x'  
## Warning: Removed 825 rows containing non-finite values (stat_smooth).  
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : pseudoinverse used at -1.64  
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : neighborhood radius 2.64  
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : reciprocal condition number 4.3021e-28  
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : There are other near singularities as well. 1  
## Warning: Removed 825 rows containing missing values (geom_point).
```

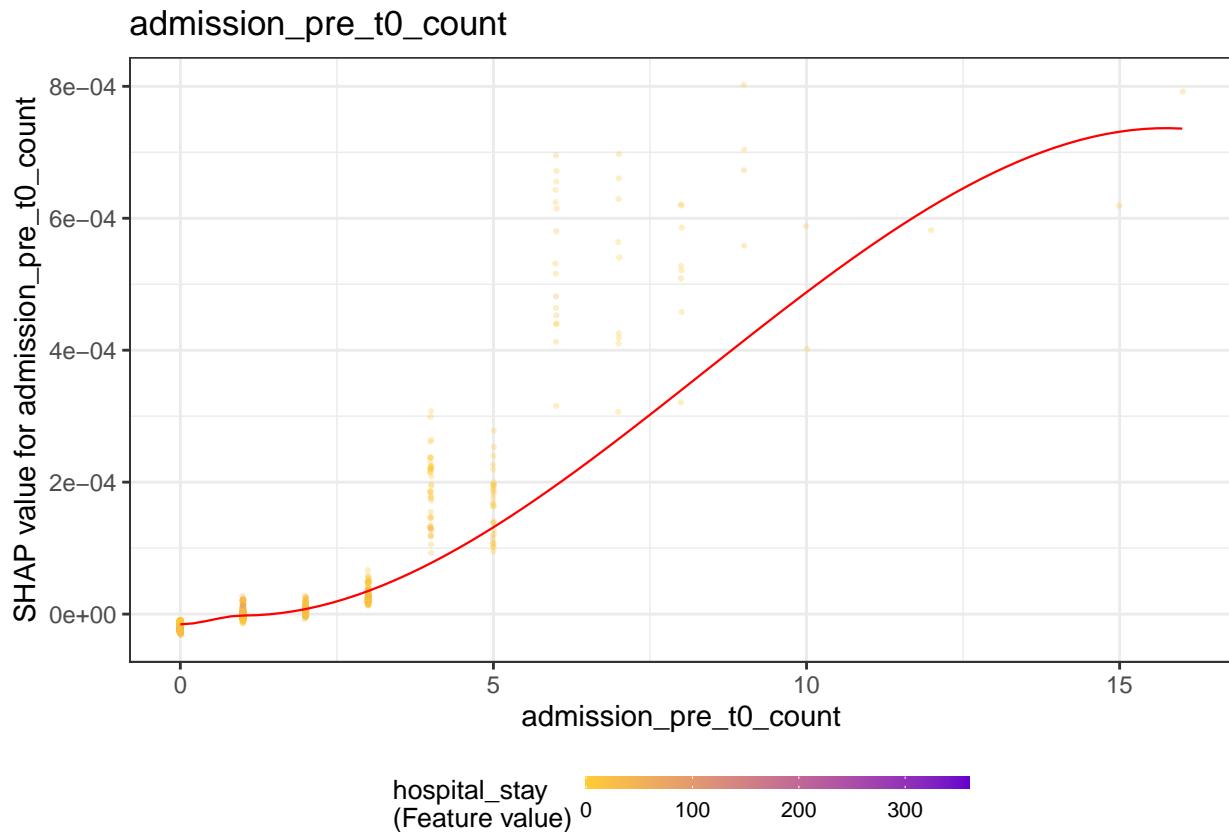
equipe_multiprof



```
## `geom_smooth()` using formula 'y ~ x'  
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : at -0.45  
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : radius 0.2025  
## 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 -0.45  
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : neighborhood radius 0.45  
## 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)
```



```
## `geom_smooth()` using formula 'y ~ x'
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.08
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.08
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1.7723e-27
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1
```



Models Comparison

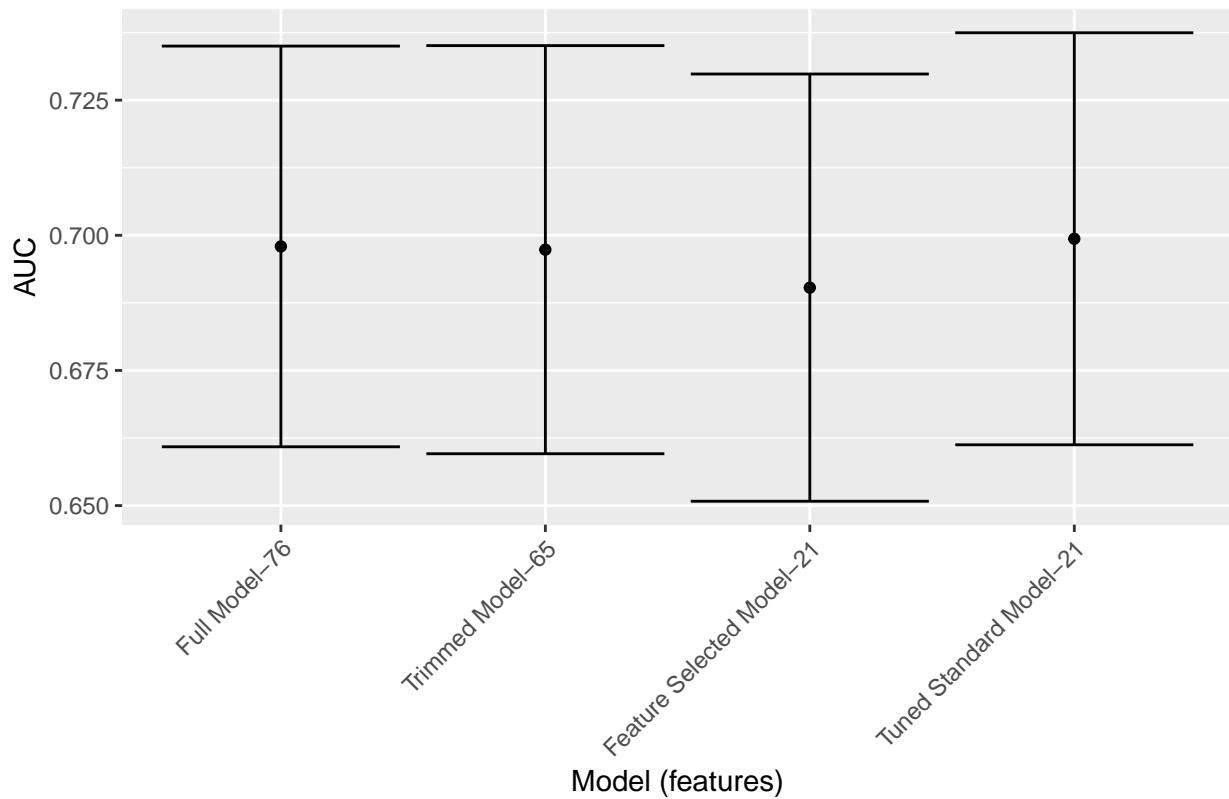
```

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$auc_upper,
  'Tuned Standard Model', standard_results$auc, standard_results$auc_lower, standard_results$auc_upper, length(selected_features),
  # 'Tuned Smote Model', smote_results$auc, smote_results$auc_lower, smote_results$auc_upper, length(selected_features),
  # 'Tuned Upsample Model', upsample_results$auc, upsample_results$auc_lower, upsample_results$auc_upper, length(selected_features)
) %>%
  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))

```

readmission_30d



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