Final Model - death_3year

Eduardo Yuki Yada

Global parameters

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

Imports

```
library(tidyverse)
library(yaml)
library(tidymodels)
library(usemodels)
library(vip)
library(kableExtra)
library(SHAPforxgboost)
library(xgboost)
library(Matrix)
library(mltools)
library(bonsai)
library(lightgbm)
library(pROC)
library(caret)
library(themis)

source("aux_functions.R")

select <- dplyr::select
```

Loading data

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

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

outcome_column <- params$outcome_column
features_list <- params$features_list

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

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

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

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

Eligible features

```
cat_features_list = readRDS(sprintf(
  "./auxiliar/significant_columns/categorical_%s.rds",
  outcome_column
))

num_features_list = readRDS(sprintf(
  "./auxiliar/significant_columns/numerical_%s.rds",
  outcome_column
))

features_list = c(cat_features_list, num_features_list)
```

```
eligible_columns <- df_names %>%
  filter(momento.aquisicao == 'Admissão t0') %>%
  .$variable.name

exception_columns <- c('death_intraop', 'death_intraop_1', 'disch_outcomes_t0')

correlated_columns = c('year_procedure_1', # com year_adm_t0
                       'age_surgery_1', # com age
                       'admission_t0', # com admission_pre_t0_count
                       'atb', # com meds_antimicrobianos
                       'classe_meds_cardio_qtde', # com classe_meds_qtde
                       'suporte_hemod', # com proced_invasivos_qtde,
                       'radiografia', # com exames_imagem_qtde
                       'ecg' # com metodos_graficos_qtde
                       )

eligible_features <- eligible_columns %>%
  base::intersect(c(columns_list$categorical_columns, columns_list$numerical_columns)) %>%
  setdiff(c(exception_columns, correlated_columns))

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

Starting features:

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

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

13. valvopathy
14. diabetes
15. renal_failure
16. hemodialysis
17. stroke
18. copd
19. comorbidities_count
20. procedure_type_1
21. reop_type_1
22. procedure_type_new
23. cied_final_1
24. cied_final_group_1
25. admission_pre_t0_count
26. admission_pre_t0_180d
27. year_adm_t0
28. icu_t0
29. dialysis_t0
30. admission_t0_emergency
31. aco
32. antiarritmico
33. ieca_bra
34. dva
35. digoxina
36. estatina
37. diuretico
38. vasodilatador
39. insuf_cardiaca
40. espironolactona
41. antiplaquetario_ev
42. insulina
43. anticonvulsivante
44. psicofarmacos
45. antifungico
46. classe_meds_qtde
47. meds_cardiovasc_qtde
48. meds_antimicrobianos
49. ventilacao_mecanica
50. transplante_cardiaco
51. outros_proced_cirurgicos
52. icp
53. angioplastia
54. cateterismo
55. eletrofisiologia
56. cateter_venoso_central
57. proced_invasivos_qtde
58. transfusao
59. equipe_multiprof
60. holter
61. teste_esforco
62. tilt_teste
63. metodos_graficos_qtde
64. laboratorio
65. cultura
66. analises_clinicas_qtde
67. citologia
68. histopatologia_qtde
69. angio_tc
70. angiografia
71. cintilografia
72. ecocardiograma
73. endoscopia

74. flebografia
75. pet_ct
76. ultrassom
77. tomografia
78. ressonancia
79. exames_imagem_qtde
80. bic
81. 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: 0.780"
```

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

```
## [1] "Full Model Test AUC: 0.802"
```

Features with zero importance on the initial model:

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

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

1. hemodialysis
2. angioplastia
3. tilt_teste
4. histopatologia_qtde
5. angiografia

```
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: 0.784"
```

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

```
## [1] "Trimmed Model Test AUC: 0.801"
```

```
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)
  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] "76"         "-0.0015"    "0.0023"     "transfusao"
## [1] "76"                  "-3e-04"               "0.0036"               "transplante_cardiaco"
## [1] "75"      "-0.0042" "-4e-04"  "stroke"
## [1] "74"      "-0.0027" "0.0015"  "pet_ct"
## [1] "73"            "-0.0049"       "-0.0022"       "heart_disease"
## [1] "73"            "-0.0021"       "0.0028"        "teste_esforco"
## [1] "73"                     "-0.0019"               "0.0029"                "cateter_venoso_central"
## [1] "72"       "-0.0047"  "1e-04"    "angio_tc"
## [1] "71"                "-0.0045"            "2e-04"              "antiplaquetario_ev"
## [1] "70"          "-0.0054"     "-9e-04"      "flebografia"
## [1] "69"                "-0.0035"            "0.0019"             "procedure_type_new"
## [1] "68"        "-0.0055"   "-0.002"    "citologia"
## [1] "68"              "-0.003"           "0.0025"           "eletrofisiologia"
## [1] "68"      "-0.0017" "0.0039"  "copd"
## [1] "68"         "-0.0013"    "0.0042"     "endoscopia"
## [1] "67"          "-0.0047"     "8e-04"       "dialysis_t0"
## [1] "67"            "-8e-04"        "0.0039"        "cintilografia"
## [1] "67"      "-0.0023" "0.0024"  "holter"
## [1] "66"             "-0.0044"        "3e-04"          "cardiac_arrest"
## [1] "65"     "-0.003" "0.0014" "icp"
## [1] "65"          "-1e-04"      "0.0029"      "antifungico"
## [1] "64"              "-0.0015"          "0.0015"           "procedure_type_1"
## [1] "63"          "-7e-04"      "8e-04"       "ressonancia"
## [1] "62"          "-0.003"      "-0.0023"     "cateterismo"
## [1] "62"                       "-4e-04"                   "0.0026"                   "outros_proced_cirurgicc
## [1] "62"           "5e-04"        "0.0035"       "cied_final_1"
## [1] "61"                  "-0.005"              "-0.002"              "ventilacao_mecanica"
## [1] "61"       "-5e-04"   "0.0044"   "insulina"
## [1] "61"       "-2e-04"   "0.0047"   "prior_mi"
## [1] "61"     "0.0035" "0.0084" "bic"
## [1] "61"      "-0.0022" "0.0027"  "af"
## [1] "61"            "-2e-04"        "0.0048"        "heart_failure"
## [1] "61"               "-0.0021"           "0.0029"            "anticonvulsivante"
## [1] "61"      "-0.0011" "0.0038"  "aco"
## [1] "61"            "2e-04"          "0.0051"         "ecocardiograma"
## [1] "61"         "-7e-04"     "0.0042"     "valvopathy"
## [1] "61"      "-4e-04"  "0.0046"  "cultura"
## [1] "61"           "0.0019"       "0.0068"       "hypertension"
## [1] "61"       "-0.0018"  "0.0031"   "diabetes"
## [1] "61"     "0.0029" "0.0079" "race"
## [1] "61"       "-0.0019"  "0.0031"   "digoxina"
## [1] "61"                   "0.0025"               "0.0074"               "admission_pre_t0_180d"
## [1] "61"        "-0.0025"   "0.0025"    "ultrassom"
## [1] "60"                   "-0.0043"              "7e-04"                "proced_invasivos_qtde"
## [1] "60"     "3e-04"  "0.0045" "sex"
## [1] "60"                   "-0.002"                 "0.0022"                "admission_t0_emergency"
## [1] "60"         "0.0011"     "0.0053"     "tomografia"
## [1] "60"            "0.0039"        "0.0081"        "renal_failure"
## [1] "60"                     "5e-04"                    "0.0048"                   "underlying_heart_diseas
## [1] "60"                "3e-04"             "0.0046"            "cied_final_group_1"
## [1] "59"          "-0.0034"     "8e-04"       "reop_type_1"
## [1] "59"     "-7e-04" "0.0027" "dva"
## [1] "59"                   "0.0021"                 "0.0055"                "analises_clinicas_qtde"
## [1] "59"            "0.0011"        "0.0045"        "antiarritmico"
## [1] "59"         "0.0052"     "0.0086"     "nyha_basal"
## [1] "59"       "-7e-04"   "0.0028"   "estatina"
```

```
## [1] "58"                  "-0.002"            "0.0014"            "exames_imagem_qtde"
## [1] "58"                "0.0023"          "0.0043"          "equipe_multiprof"
## [1] "57"                   "-0.0039"             "-0.0019"             "meds_antimicrobianos"
## [1] "56"      "-0.0022" "0.0017"  "icu_t0"
## [1] "55"          "-3e-04"      "0.0019"      "laboratorio"
## [1] "55"              "0.0069"          "0.0071"          "education_level"
## [1] "54"                   "-2e-04"               "1e-04"                 "metodos_graficos_qtde"
## [1] "53"            "-0.0014"       "-0.0012"       "psicofarmacos"
## [1] "53"            "7e-04"         "0.0021"        "vasodilatador"
## [1] "53"              "0.0023"          "0.0037"          "classe_meds_qtde"
## [1] "53"             "0.0026"         "0.004"          "insuf_cardiaca"
## [1] "53"                  "0.0052"             "0.0066"             "comorbidities_count"
## [1] "52"        "-6e-04"    "8e-04"     "diuretico"
## [1] "51"                   "-0.002"               "-0.0014"              "meds_cardiovasc_qtde"
## [1] "51"       "0.0059"   "0.0079"   "ieca_bra"
## [1] "50"              "-0.0031"         "-0.0011"         "espironolactona"
## [1] "50"                    "0.0117"                "0.0148"                 "admission_pre_t0_count"
```

```
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	81	0.7801	0.0031	0.0000	0.0000
All unimportant	TRUE	76	0.7840	0.0046	-0.0038	-0.0038
transfusao	FALSE	75	0.7817	0.0041	-0.0015	0.0023
transplante_cardiaco	FALSE	75	0.7804	0.0029	-0.0003	0.0036
stroke	TRUE	75	0.7843	0.0029	-0.0042	-0.0004
pet_ct	TRUE	74	0.7828	0.0034	-0.0027	0.0015
heart_disease	TRUE	73	0.7850	0.0027	-0.0049	-0.0022
teste_esforco	FALSE	72	0.7823	0.0036	-0.0021	0.0028
cateter_venoso_central	FALSE	72	0.7821	0.0027	-0.0019	0.0029
angio_tc	TRUE	72	0.7849	0.0044	-0.0047	0.0001
antiplaquetario_ev	TRUE	71	0.7846	0.0030	-0.0045	0.0002
flebografia	TRUE	70	0.7856	0.0027	-0.0054	-0.0009
procedure_type_new	TRUE	69	0.7837	0.0029	-0.0035	0.0019
citologia	TRUE	68	0.7857	0.0037	-0.0055	-0.0020
eletrofisiologia	FALSE	67	0.7831	0.0009	-0.0030	0.0025
copd	FALSE	67	0.7818	0.0046	-0.0017	0.0039
endoscopia	FALSE	67	0.7815	0.0037	-0.0013	0.0042
dialysis_t0	TRUE	67	0.7848	0.0020	-0.0047	0.0008
cintilografia	FALSE	66	0.7810	0.0037	-0.0008	0.0039
holter	FALSE	66	0.7824	0.0023	-0.0023	0.0024
cardiac_arrest	TRUE	66	0.7846	0.0028	-0.0044	0.0003
icp	TRUE	65	0.7832	0.0038	-0.0030	0.0014
antifungico	FALSE	64	0.7803	0.0024	-0.0001	0.0029
procedure_type_1	TRUE	64	0.7816	0.0029	-0.0015	0.0015
ressonancia	TRUE	63	0.7808	0.0015	-0.0007	0.0008
cateterismo	TRUE	62	0.7831	0.0036	-0.0030	-0.0023
outros_proced_cirurgicos	FALSE	61	0.7805	0.0018	-0.0004	0.0026
cied_final_1	FALSE	61	0.7796	0.0029	0.0005	0.0035
ventilacao_mecanica	TRUE	61	0.7851	0.0057	-0.0050	-0.0020
insulina	FALSE	60	0.7807	0.0024	-0.0005	0.0044
prior_mi	FALSE	60	0.7804	0.0033	-0.0002	0.0047
bic	FALSE	60	0.7767	0.0013	0.0035	0.0084
af	FALSE	60	0.7824	0.0018	-0.0022	0.0027
heart_failure	FALSE	60	0.7803	0.0026	-0.0002	0.0048

Table 1: *(continued)*

Tested Feature	Dropped	Features	CV AUC	CV AUC Std Error	Total AUC Loss	Instant AUC Loss
anticonvulsivante	FALSE	60	0.7822	0.0019	-0.0021	0.0029
aco	FALSE	60	0.7813	0.0015	-0.0011	0.0038
ecocardiograma	FALSE	60	0.7800	0.0029	0.0002	0.0051
valvopathy	FALSE	60	0.7809	0.0030	-0.0007	0.0042
cultura	FALSE	60	0.7805	0.0032	-0.0004	0.0046
hypertension	FALSE	60	0.7783	0.0008	0.0019	0.0068
diabetes	FALSE	60	0.7820	0.0023	-0.0018	0.0031
race	FALSE	60	0.7772	0.0024	0.0029	0.0079
digoxina	FALSE	60	0.7820	0.0016	-0.0019	0.0031
admission_pre_t0_180d	FALSE	60	0.7777	0.0036	0.0025	0.0074
ultrassom	FALSE	60	0.7827	0.0022	-0.0025	0.0025
proced_invasivos_qtde	TRUE	60	0.7844	0.0027	-0.0043	0.0007
sex	FALSE	59	0.7799	0.0024	0.0003	0.0045
admission_t0_emergency	FALSE	59	0.7822	0.0037	-0.0020	0.0022
tomografia	FALSE	59	0.7791	0.0028	0.0011	0.0053
renal_failure	FALSE	59	0.7763	0.0033	0.0039	0.0081
underlying_heart_disease	FALSE	59	0.7796	0.0029	0.0005	0.0048
cied_final_group_1	FALSE	59	0.7798	0.0016	0.0003	0.0046
reop_type_1	TRUE	59	0.7836	0.0013	-0.0034	0.0008
dva	FALSE	58	0.7809	0.0038	-0.0007	0.0027
analises_clinicas_qtde	FALSE	58	0.7780	0.0016	0.0021	0.0055
antiarritmico	FALSE	58	0.7791	0.0029	0.0011	0.0045
nyha_basal	FALSE	58	0.7749	0.0041	0.0052	0.0086
estatina	FALSE	58	0.7808	0.0048	-0.0007	0.0028
exames_imagem_qtde	TRUE	58	0.7821	0.0017	-0.0020	0.0014
equipe_multiprof	FALSE	57	0.7778	0.0037	0.0023	0.0043
meds_antimicrobianos	TRUE	57	0.7840	0.0027	-0.0039	-0.0019
icu_t0	TRUE	56	0.7823	0.0037	-0.0022	0.0017
laboratorio	TRUE	55	0.7804	0.0017	-0.0003	0.0019
education_level	FALSE	54	0.7733	0.0034	0.0069	0.0071
metodos_graficos_qtde	TRUE	54	0.7803	0.0027	-0.0002	0.0001
psicofarmacos	TRUE	53	0.7816	0.0030	-0.0014	-0.0012
vasodilatador	FALSE	52	0.7795	0.0041	0.0007	0.0021
classe_meds_qtde	FALSE	52	0.7779	0.0040	0.0023	0.0037
insuf_cardiaca	FALSE	52	0.7776	0.0018	0.0026	0.0040
comorbidities_count	FALSE	52	0.7750	0.0032	0.0052	0.0066
diuretico	TRUE	52	0.7808	0.0037	-0.0006	0.0008
meds_cardiovasc_qtde	TRUE	51	0.7821	0.0041	-0.0020	-0.0014
ieca_bra	FALSE	50	0.7743	0.0040	0.0059	0.0079
espironolactona	TRUE	50	0.7832	0.0054	-0.0031	-0.0011
admission_pre_t0_count	FALSE	49	0.7685	0.0072	0.0117	0.0148

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

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

feature_selected_model <- model_fit_wf(df_train, selected_features,
                                       outcome_column, hyperparameters)
```

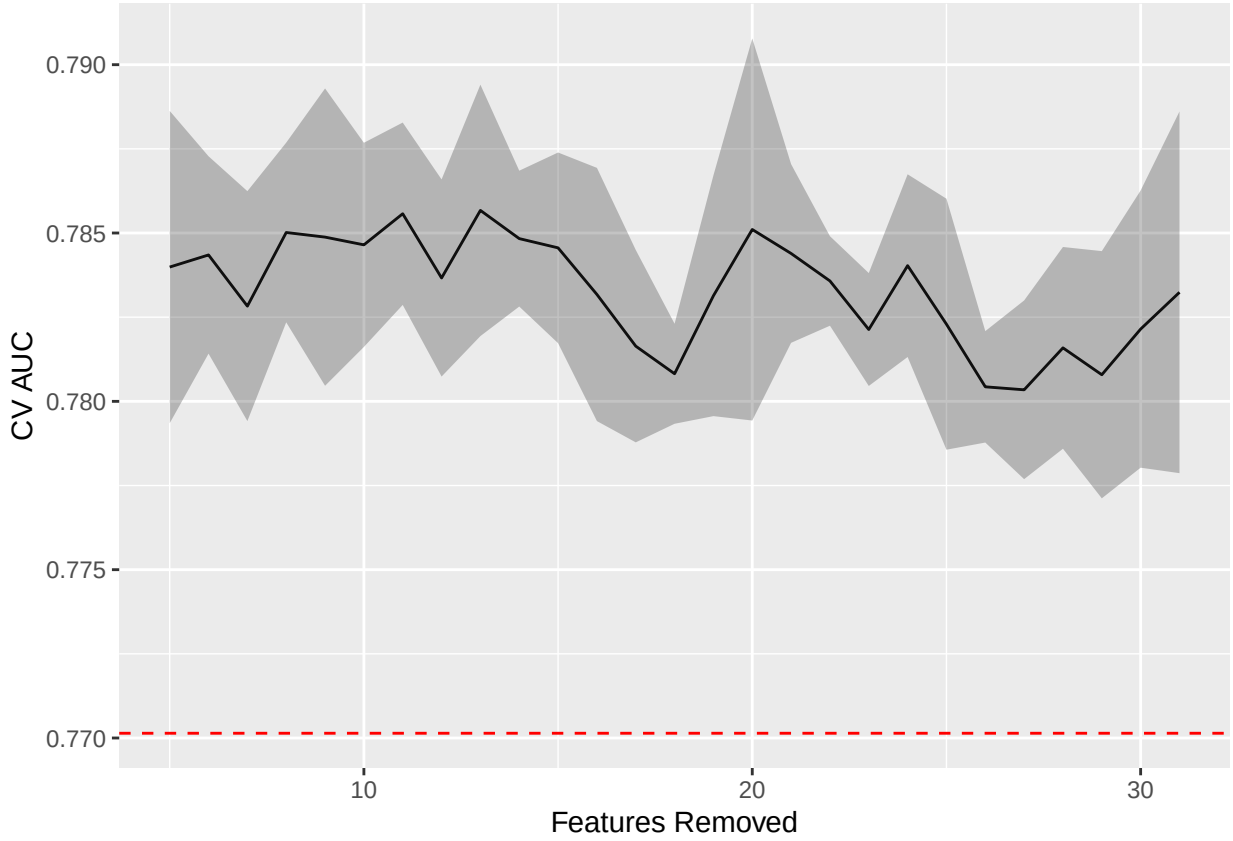
```
sprintf('Selected Model CV Train AUC: %.3f', feature_selected_model$cv_auc)
```

```
## [1] "Selected Model CV Train AUC: 0.780"
```

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

```
## [1] "Selected Model Test AUC: 0.787"
```

```
 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. age
2. hospital_stay
3. year_adm_t0
4. admission_pre_t0_count
5. ieca_bra

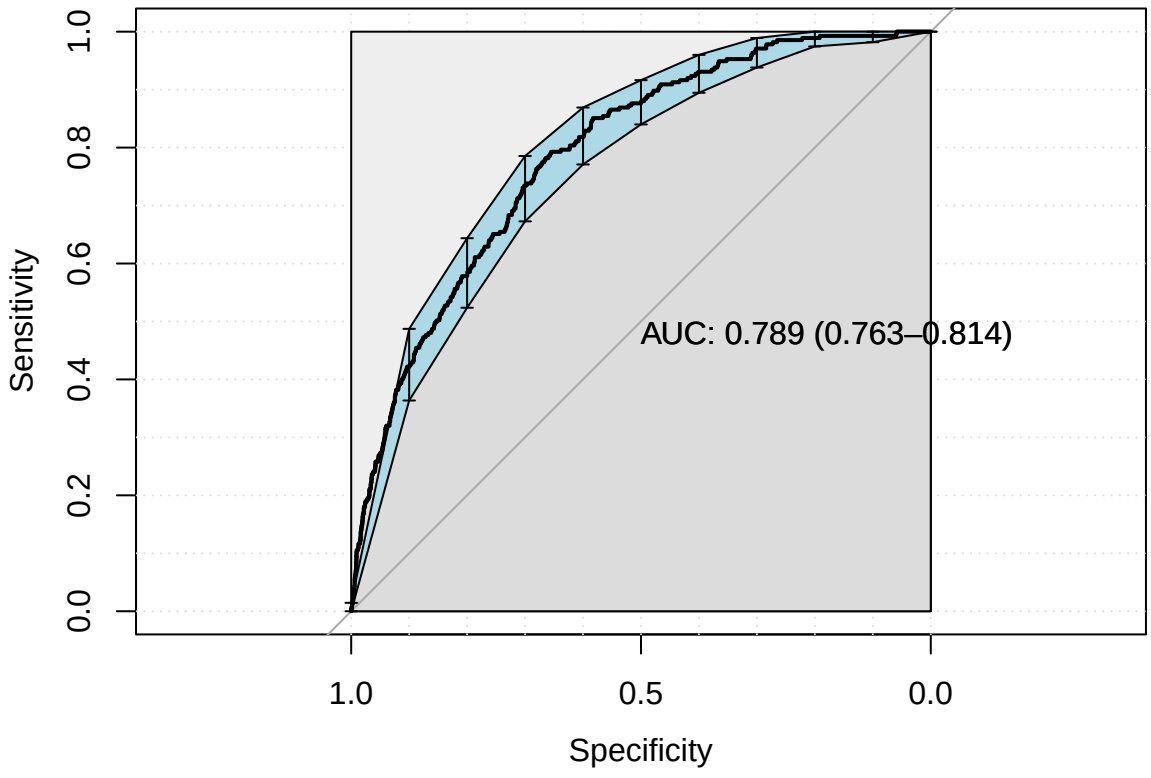
6. classe_meds_qtde
7. comorbidities_count
8. insuf_cardiaca
9. education_level
10. vasodilatador
11. estatina
12. equipe_multiprof
13. analises_clinicas_qtde
14. antiarritmico
15. nyha_basal
16. dva
17. admission_t0_emergency
18. cied_final_group_1
19. renal_failure
20. valvopathy
21. hypertension
22. admission_pre_t0_180d
23. race
24. underlying_heart_disease
25. sex
26. ultrassom
27. diabetes
28. digoxina
29. ecocardiograma
30. aco
31. cied_final_1
32. tomografia
33. cultura
34. af
35. heart_failure
36. bic
37. prior_mi
38. anticonvulsivante
39. outros_proced_cirurgicos
40. insulina
41. antifungico
42. endoscopia
43. copd
44. holter
45. eletrofisiologia
46. cintilografia
47. cateter_venoso_central
48. teste_esforco
49. transfusao
50. transplante_cardiaco

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(50L, 300L)),
    min_n(),
    tree_depth(),
    learn_rate(range = c(0.01, 0.3), trans = NULL),
    loss_reduction(),
    size = grid_size
  )

  lightgbm_workflow <-
    workflow() %>%
    add_recipe(recipe) %>%
    add_model(lightgbm_spec)

  lightgbm_tune <-
    lightgbm_workflow %>%
    tune_grid(resamples = df_folds,
              grid = lightgbm_grid)

  lightgbm_tune %>%
    show_best("roc_auc") %>%
    niceFormatting(digits = 5, label = 4)

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

  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)
```

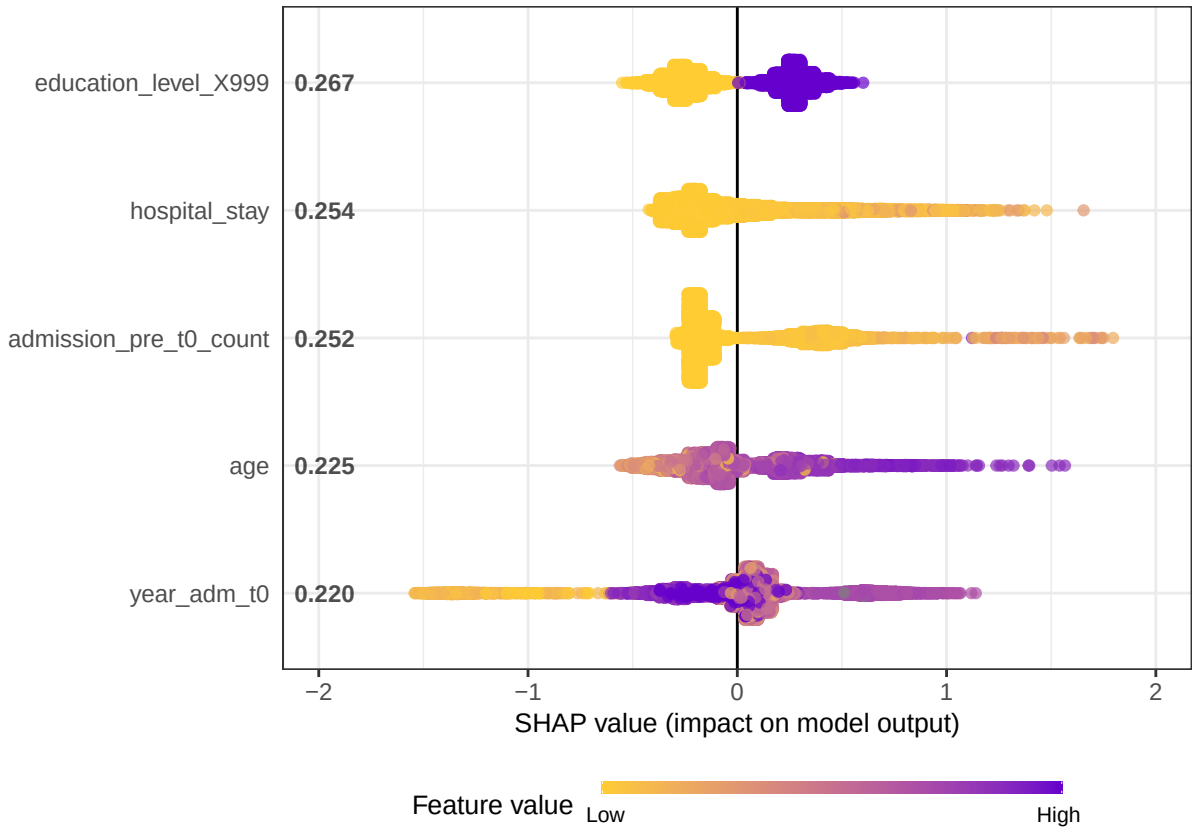


```
## [1] "Optimal Threshold: 0.04"
## Confusion Matrix and Statistics
##
##     reference
## data    0    1
##    0 2919   57
##    1 1536  218
##
##                Accuracy : 0.6632
##                  95% CI : (0.6495, 0.6767)
##     No Information Rate : 0.9419
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1271
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6552
##             Specificity : 0.7927
##          Pos Pred Value : 0.9808
##          Neg Pred Value : 0.1243
##              Prevalence : 0.9419
##          Detection Rate : 0.6171
##    Detection Prevalence : 0.6292
##       Balanced Accuracy : 0.7240
##
##        'Positive' Class : 0
##
```

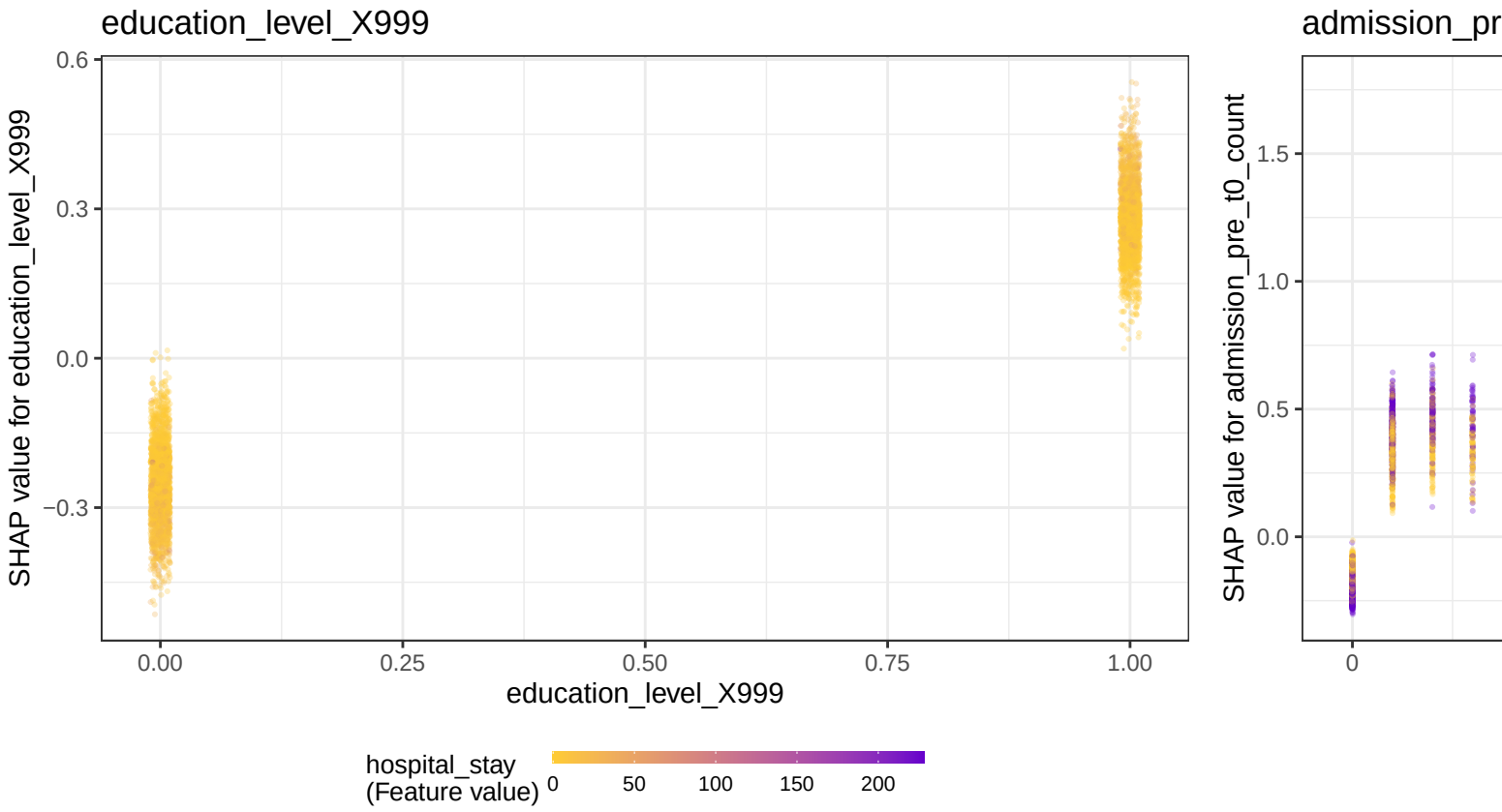
```r
# smote_results <- lightgbm_tuning(lightgbm_smote_recipe)
# upsample_results <- lightgbm_tuning(lightgbm_upsample_recipe)

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
  )
)
```

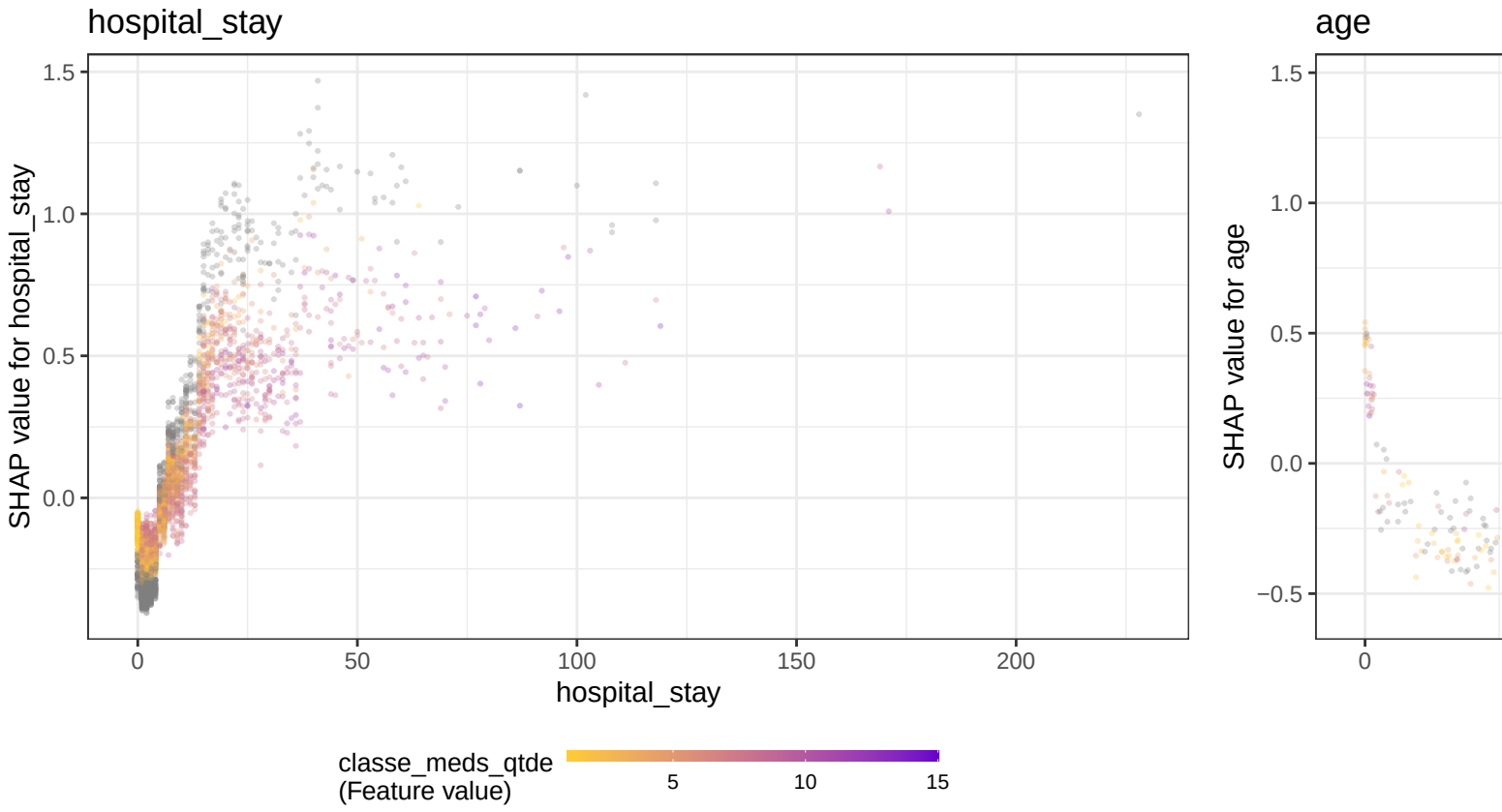
SHAP values

```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(5, length(selected_features))

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

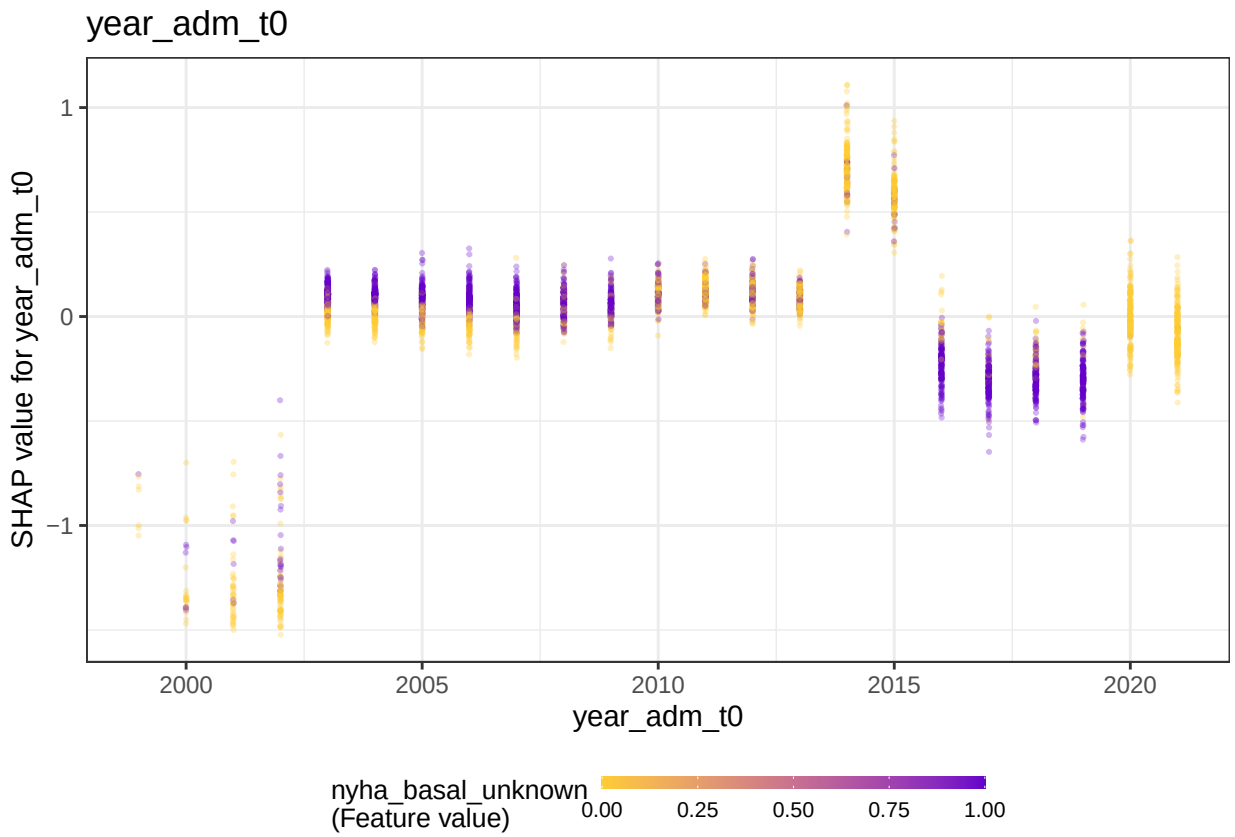



```
shap <- shap.prep(lightgbm_model, X_train = matrix_test)

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






```
## Warning: Removed 6 rows containing missing values (geom_point).
```

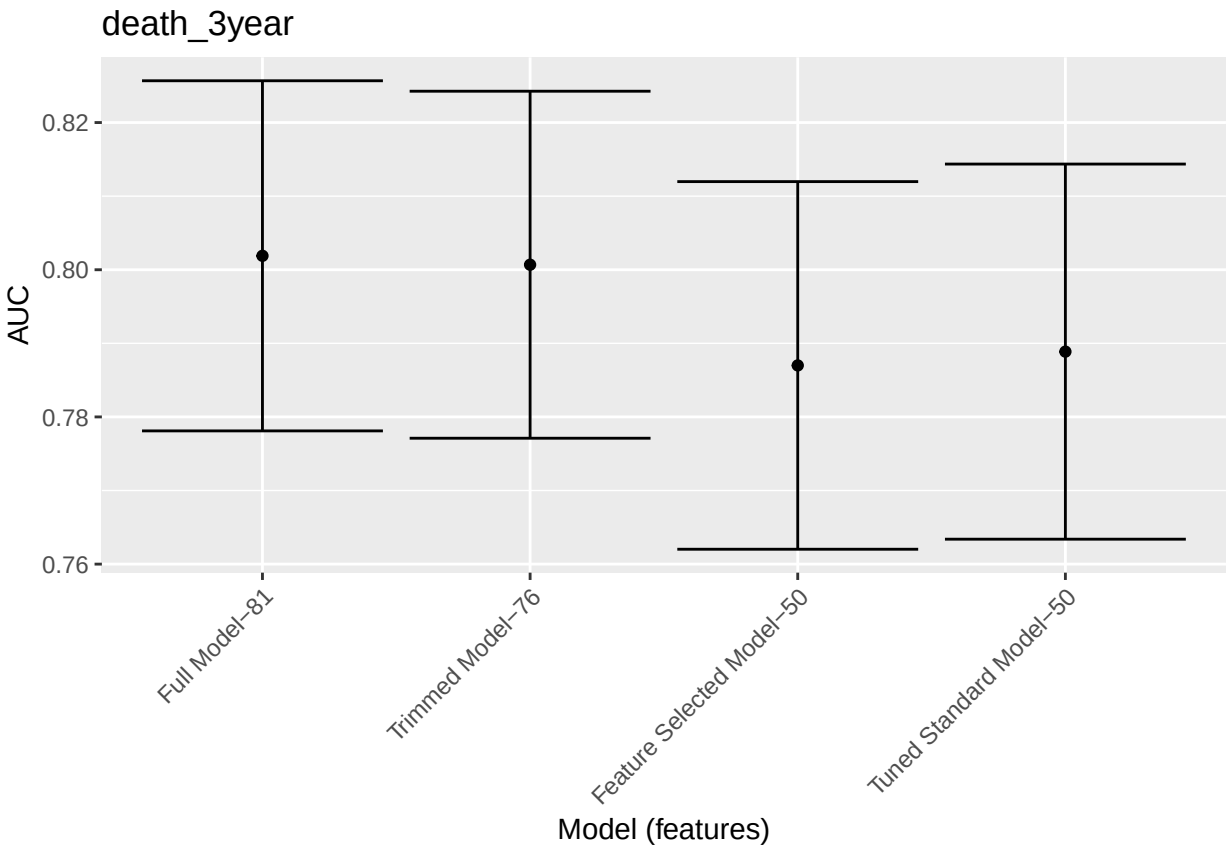



```
## $num_iterations
## [1] 129
##
## $learning_rate
## [1] 0.07676426
##
## $max_depth
## [1] 4
##
## $feature_fraction
## [1] 0.3561644
##
## $min_data_in_leaf
## [1] 30
##
## $min_gain_to_split
## [1] 1.351057e-10
##
## $bagging_fraction
## [1] 1
##
## $num_class
## [1] 1
##
## $objective
## [1] "binary"
##
## $num_threads
## $num_threads$num_threads
## [1] 0
##
##
## $seed
```

```
## [1] 78745
##
## $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$auc_upper, length(s
  # 'Tuned Smote Model', smote_results$auc, smote_results$auc_lower, smote_results$auc_upper, length(selected_fe
  # 'Tuned Upsample Model', upsample_results$auc, upsample_results$auc_lower, upsample_results$auc_upper, length
) %>%
  mutate(Target = outcome_column,
         `Model (features)` = fct_reorder(paste0(Model, "-", Features), -Features))

df_auc %>%
  ggplot(aes(
    x = `Model (features)`,
    y = AUC,
    ymin = `Lower Limit`,
    ymax = `Upper Limit`
  )) +
  geom_point() +
  geom_errorbar() +
  labs(title = outcome_column) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




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