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: 0.799"
```

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

```
## [1] "Full Model Test AUC: 0.798"
```

Features with zero importance on the initial model:

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

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

1. heart_disease
2. hemodialysis
3. stroke
4. antiplaquetario_ev
5. ventilacao_mecanica
6. transplante_cardiaco
7. angioplastia
8. transfusao
9. holter
10. tilt_teste
11. cultura
12. histopatologia_qtde
13. angio_tc
14. angiografia
15. cintilografia
16. flebografia
17. 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: 0.798"
```

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

```
## [1] "Trimmed Model Test AUC: 0.797"
```

```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) {
  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] "63"                       "6e-04"                    "-6e-04"
## [4] "outros_proced_cirurgicos"
## [1] "62"          "-1e-04"     "-7e-04"     "endoscopia"
## [1] "61"           "4e-04"       "4e-04"       "antifungico"
## [1] "60"        "8e-04"    "5e-04"    "insulina"
## [1] "59"        "8e-04"    "0"        "digoxina"
## [1] "58"               "7e-04"            "-1e-04"           "eletrofisiologia"
## [1] "57"     "1e-04"  "-6e-04" "af"
## [1] "56"     "0"      "-1e-04" "copd"
## [1] "55"                    "1e-04"                "0"                      "admission_pre_t0_180d"
## [1] "54"            "-4e-04"        "-4e-04"        "teste_esforco"
## [1] "53"                    "-1e-04"                "3e-04"                  "admission_t0_emergency"
## [1] "52"          "-1e-04"      "-1e-04"      "ressonancia"
## [1] "51"             "2e-04"          "4e-04"          "ecocardiograma"
## [1] "50"                    "-6e-04"                "-9e-04"                 "analises_clinicas_qtde"
## [1] "49"     "-3e-04" "3e-04"  "icp"
## [1] "48"                   "-0.0017"               "-0.0014"               "proced_invasivos_qtde"
## [1] "47"         "-0.0017"    "0"          "tomografia"
## [1] "46"     "-0.002" "-2e-04" "bic"
## [1] "45"                "-0.002"            "0"                  "anticonvulsivante"
## [1] "44"          "-0.002"      "0"           "reop_type_1"
## [1] "43"                "-0.0022"            "-2e-04"             "procedure_type_new"
## [1] "42"               "-0.0021"          "1e-04"            "procedure_type_1"
## [1] "41"           "-0.0023"      "-2e-04"       "hypertension"
## [1] "40"          "-0.0022"     "1e-04"       "dialysis_t0"
## [1] "39"          "-0.0022"     "1e-04"       "cateterismo"
## [1] "38"      "-0.0031" "-9e-04"  "dva"
## [1] "37"                "-0.0034"            "-3e-04"             "exames_imagem_qtde"
## [1] "36"        "-0.0033"   "1e-04"     "citologia"
## [1] "35"       "-0.0027"  "5e-04"    "prior_mi"
## [1] "34"                    "-0.0034"               "-6e-04"                 "cateter_venoso_central"
## [1] "33"      "-0.0033" "1e-04"   "sex"
## [1] "32"      "-0.0022" "0.0011"  "race"
## [1] "31"       "-0.0016"  "6e-04"    "diabetes"
## [1] "30"       "-0.0019"  "-3e-04"   "estatina"
## [1] "29"            "-0.0014"       "5e-04"         "renal_failure"
## [1] "28"      "-0.0014" "-1e-04"  "aco"
## [1] "27"        "-0.0016"   "-1e-04"    "ultrassom"
## [1] "26"               "-0.0014"          "2e-04"            "equipe_multiprof"
## [1] "25"         "-0.0025"    "-0.0011"    "valvopathy"
## [1] "24"            "-0.0025"       "0"             "psicofarmacos"
## [1] "23"           "-0.0018"      "7e-04"        "cied_final_1"
## [1] "22"                        "-0.0023"                  "-5e-04"
## [4] "underlying_heart_disease"
## [1] "21"            "-0.0013"       "9e-04"         "antiarritmico"
## [1] "20"      "-0.0024" "-0.0011" "icu_t0"
## [1] "19"                    "-0.0019"               "6e-04"                  "metodos_graficos_qtde"
## [1] "18"                   "-0.0024"              "-5e-04"               "meds_cardiovasc_qtde"
## [1] "17"            "-0.0026"       "-3e-04"        "heart_failure"
## [1] "16"             "-0.0027"        "0"             "cardiac_arrest"
```

```
## [1] "15"                 "-0.0013"           "0.0014"            "cied_final_group_1"
## [1] "14"            "-0.0011"       "2e-04"         "vasodilatador"
## [1] "13"                  "-0.0023"             "-0.0013"             "meds_antimicrobianos"
## [1] "12"          "-0.0031"     "-8e-04"      "laboratorio"
## [1] "11"             "-0.0035"        "-4e-04"         "insuf_cardiaca"
## [1] "10"         "-0.0024"    "0.0011"    "diuretico"
## [1] "10"          "0.0083"     "0.0107"     "nyha_basal"
## [1] "9"         "-0.002"    "4e-04"    "ieca_bra"
## [1] "9"               "0.0095"          "0.0115"          "education_level"
## [1] "9"                  "0.0033"             "0.0053"             "comorbidities_count"
## [1] "8"                "-0.0044"          "-0.0023"          "classe_meds_qtde"
## [1] "8"                     "0.0146"                "0.0189"                "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	0.7995	0.0069	0.0000	0.0000
All unimportant	TRUE	64	0.7982	0.0065	0.0013	0.0013
outros_proced_cirurgicos	TRUE	63	0.7988	0.0069	0.0006	-0.0006
endoscopia	TRUE	62	0.7995	0.0066	-0.0001	-0.0007
antifungico	TRUE	61	0.7991	0.0065	0.0004	0.0004
insulina	TRUE	60	0.7987	0.0067	0.0008	0.0005
digoxina	TRUE	59	0.7987	0.0066	0.0008	0.0000
eletrofisiologia	TRUE	58	0.7988	0.0064	0.0007	-0.0001
af	TRUE	57	0.7994	0.0066	0.0001	-0.0006
copd	TRUE	56	0.7995	0.0066	0.0000	-0.0001
admission_pre_t0_180d	TRUE	55	0.7994	0.0066	0.0001	0.0000
teste_esforco	TRUE	54	0.7998	0.0067	-0.0004	-0.0004
admission_t0_emergency	TRUE	53	0.7996	0.0067	-0.0001	0.0003
ressonancia	TRUE	52	0.7996	0.0066	-0.0001	-0.0001
ecocardiograma	TRUE	51	0.7993	0.0068	0.0002	0.0004
analises_clinicas_qtde	TRUE	50	0.8001	0.0065	-0.0006	-0.0009
icp	TRUE	49	0.7998	0.0065	-0.0003	0.0003
proced_invasivos_qtde	TRUE	48	0.8012	0.0056	-0.0017	-0.0014
tomografia	TRUE	47	0.8012	0.0057	-0.0017	0.0000
bic	TRUE	46	0.8014	0.0059	-0.0020	-0.0002
anticonvulsivante	TRUE	45	0.8014	0.0058	-0.0020	0.0000
reop_type_1	TRUE	44	0.8014	0.0058	-0.0020	0.0000
procedure_type_new	TRUE	43	0.8016	0.0061	-0.0022	-0.0002
procedure_type_1	TRUE	42	0.8015	0.0059	-0.0021	0.0001
hypertension	TRUE	41	0.8018	0.0061	-0.0023	-0.0002
dialysis_t0	TRUE	40	0.8017	0.0057	-0.0022	0.0001
cateterismo	TRUE	39	0.8016	0.0060	-0.0022	0.0001
dva	TRUE	38	0.8025	0.0063	-0.0031	-0.0009
exames_imagem_qtde	TRUE	37	0.8029	0.0064	-0.0034	-0.0003
citologia	TRUE	36	0.8027	0.0065	-0.0033	0.0001
prior_mi	TRUE	35	0.8022	0.0063	-0.0027	0.0005
cateter_venoso_central	TRUE	34	0.8028	0.0064	-0.0034	-0.0006
sex	TRUE	33	0.8027	0.0071	-0.0033	0.0001
race	TRUE	32	0.8017	0.0070	-0.0022	0.0011
diabetes	TRUE	31	0.8011	0.0066	-0.0016	0.0006
estatina	TRUE	30	0.8014	0.0068	-0.0019	-0.0003
renal_failure	TRUE	29	0.8009	0.0067	-0.0014	0.0005
aco	TRUE	28	0.8009	0.0063	-0.0014	-0.0001

Table 1: *(continued)*

Tested Feature	Dropped	Features	CV AUC	CV AUC Std Error	Total AUC Loss	Instant AUC Loss
ultrassom	TRUE	27	0.8011	0.0062	-0.0016	-0.0001
equipe_multiprof	TRUE	26	0.8009	0.0067	-0.0014	0.0002
valvopathy	TRUE	25	0.8020	0.0066	-0.0025	-0.0011
psicofarmacos	TRUE	24	0.8020	0.0057	-0.0025	0.0000
cied_final_1	TRUE	23	0.8013	0.0075	-0.0018	0.0007
underlying_heart_disease	TRUE	22	0.8017	0.0066	-0.0023	-0.0005
antiarritmico	TRUE	21	0.8008	0.0076	-0.0013	0.0009
icu_t0	TRUE	20	0.8019	0.0075	-0.0024	-0.0011
metodos_graficos_qtde	TRUE	19	0.8014	0.0071	-0.0019	0.0006
meds_cardiovasc_qtde	TRUE	18	0.8018	0.0072	-0.0024	-0.0005
heart_failure	TRUE	17	0.8021	0.0072	-0.0026	-0.0003
cardiac_arrest	TRUE	16	0.8021	0.0076	-0.0027	0.0000
cied_final_group_1	TRUE	15	0.8007	0.0069	-0.0013	0.0014
vasodilatador	TRUE	14	0.8005	0.0066	-0.0011	0.0002
meds_antimicrobianos	TRUE	13	0.8018	0.0060	-0.0023	-0.0013
laboratorio	TRUE	12	0.8026	0.0063	-0.0031	-0.0008
insuf_cardiaca	TRUE	11	0.8030	0.0065	-0.0035	-0.0004
diuretico	TRUE	10	0.8019	0.0067	-0.0024	0.0011
nyha_basal	FALSE	9	0.7912	0.0091	0.0083	0.0107
ieca_bra	TRUE	9	0.8015	0.0060	-0.0020	0.0004
education_level	FALSE	8	0.7899	0.0050	0.0095	0.0115
comorbidities_count	FALSE	8	0.7962	0.0072	0.0033	0.0053
classe_meds_qtde	TRUE	8	0.8038	0.0082	-0.0044	-0.0023
admission_pre_t0_count	FALSE	7	0.7849	0.0109	0.0146	0.0189

```
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: 0.803"
```

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

```
## [1] "Selected Model Test AUC: 0.784"
```

```
 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)
```

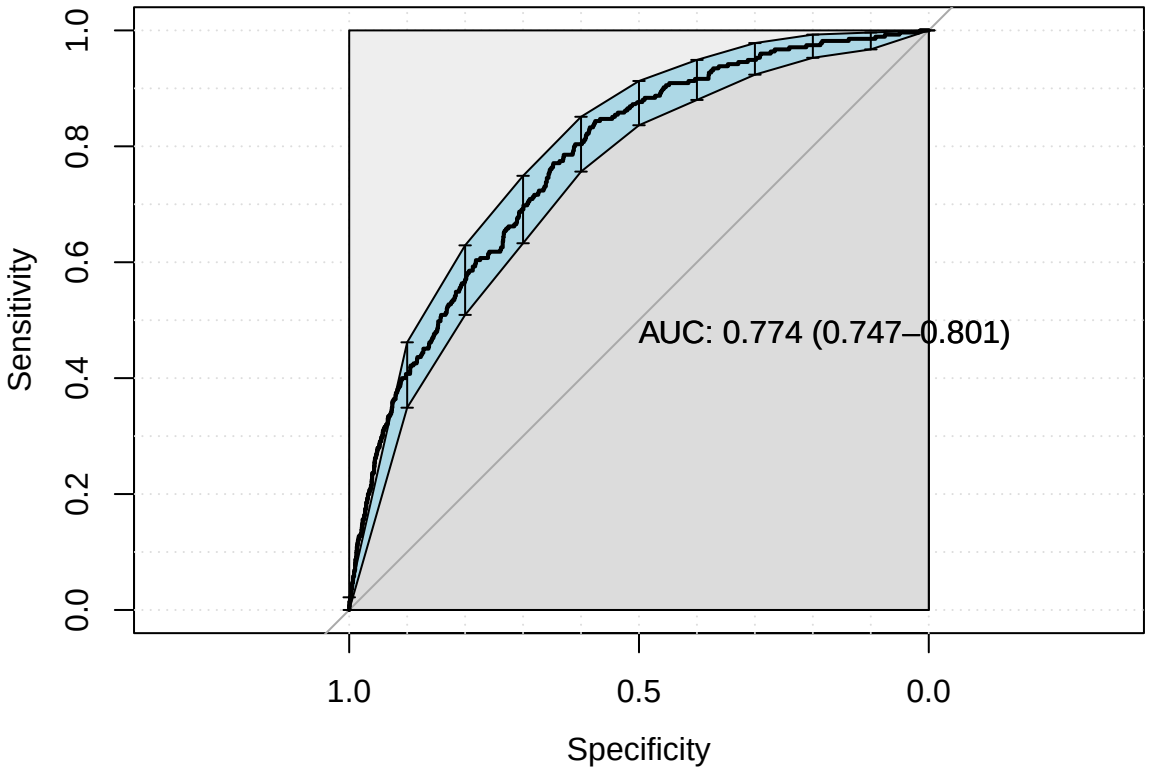
1. hospital_stay
2. year_adm_t0
3. admission_pre_t0_count
4. age
5. espironolactona
6. education_level
7. comorbidities_count
8. nyha_basal

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)
```



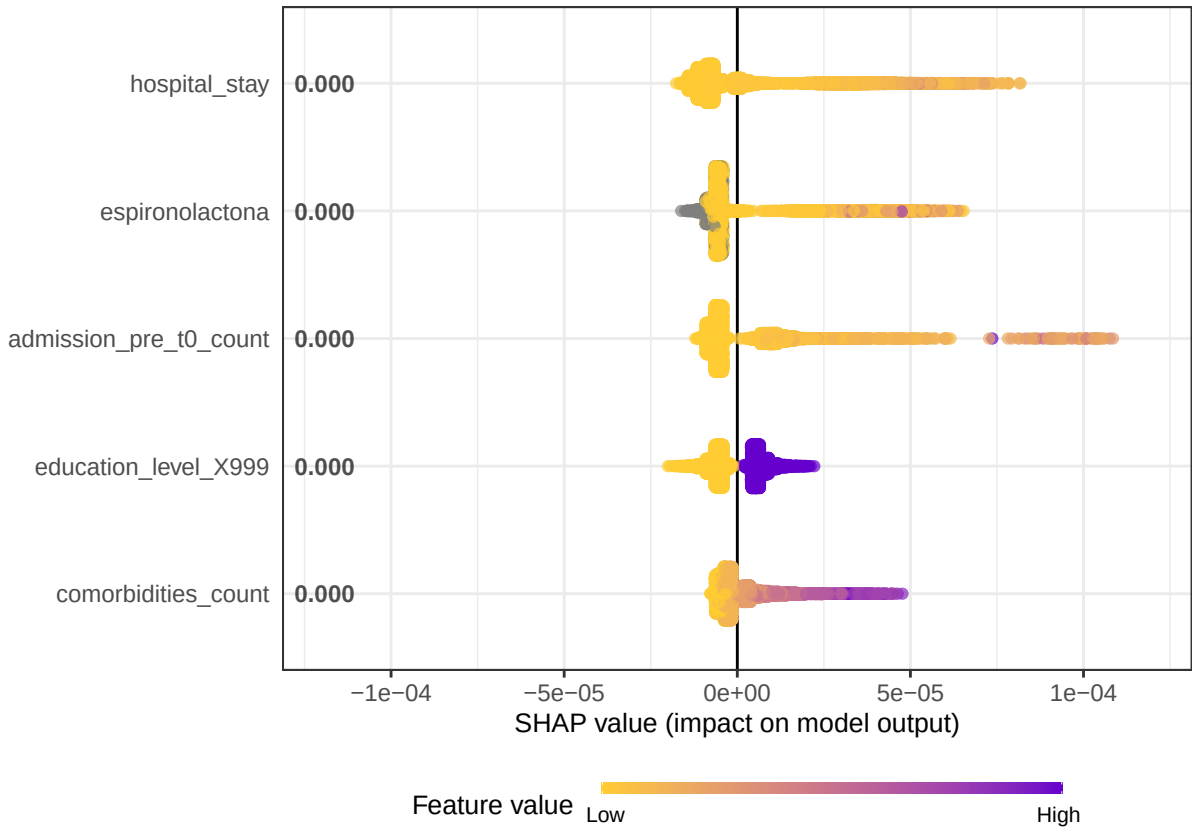

```
## [1] "Optimal Threshold: 0.06"
## Confusion Matrix and Statistics
##
##     reference
## data    0    1
##    0 2564   43
##    1 1891  232
##
##                Accuracy : 0.5911
##                  95% CI : (0.577, 0.6052)
##     No Information Rate : 0.9419
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1009
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.5755
##             Specificity : 0.8436
##          Pos Pred Value : 0.9835
##          Neg Pred Value : 0.1093
##              Prevalence : 0.9419
##          Detection Rate : 0.5421
##    Detection Prevalence : 0.5512
##       Balanced Accuracy : 0.7096
##
##        'Positive' Class : 0
##
```

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

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

```
# saveRDS(
#   lightgbm_parameters,
#   file = sprintf(
#     "./auxiliar/final_model/hyperparameters/lightgbm_%s.rds",
#     outcome_column
#   )
# )
```

SHAP values

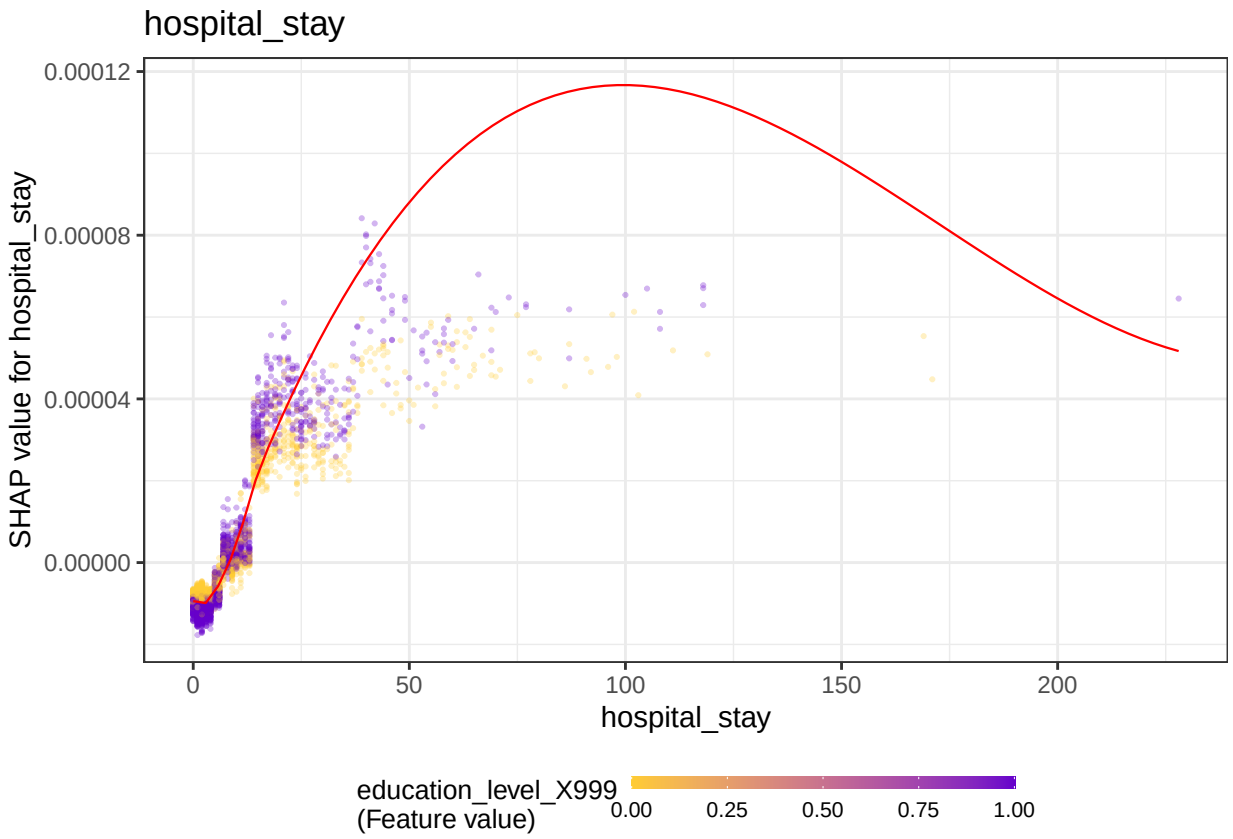
```
lightgbm_model <- parsnip::extract_fit_engine(final_lightgbm_fit)

trained_rec <- prep(lightgbm_recipe, training = df_train)

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

matrix_train <- as.matrix(df_train_baked %>% select(-all_of(outcome_column)))
matrix_test <- as.matrix(df_test_baked %>% select(-all_of(outcome_column)))

n_plots <- min(5, length(selected_features))

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



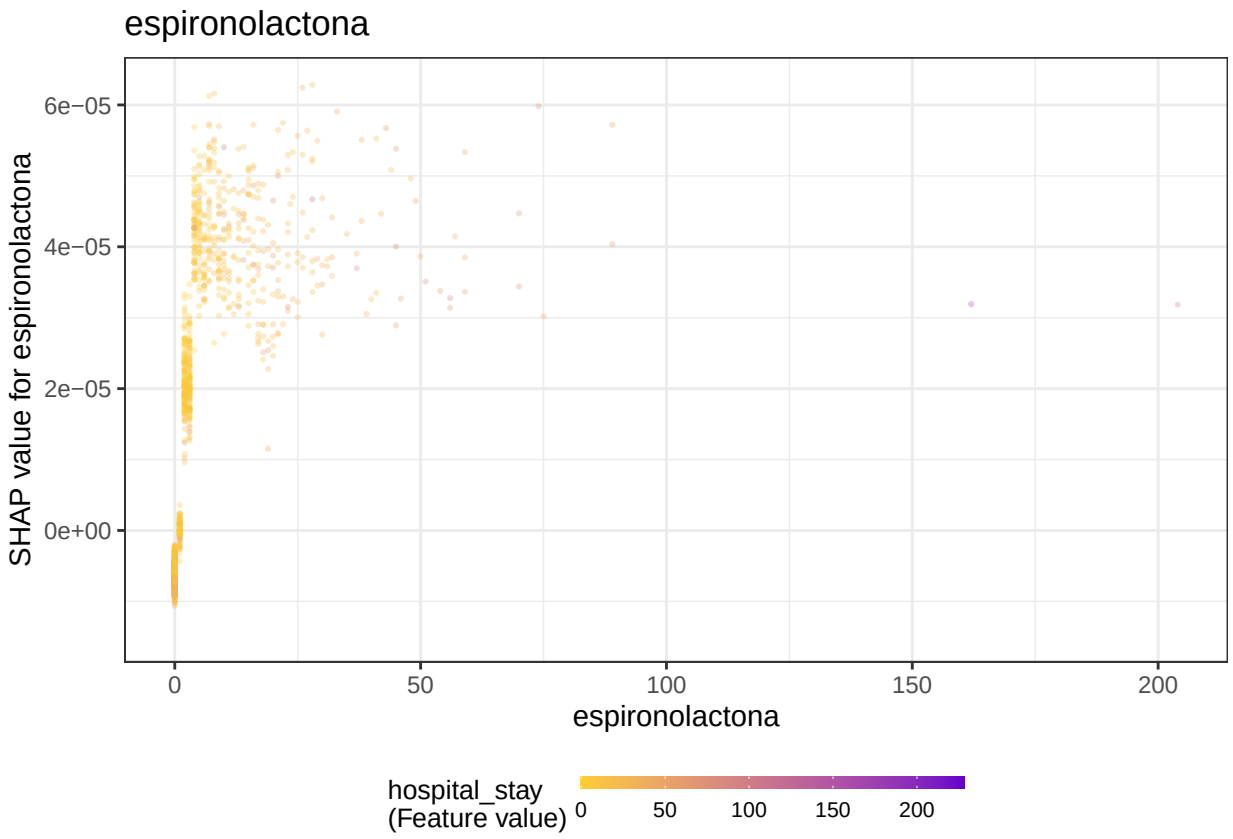

```
shap <- shap.prep(lightgbm_model, X_train = matrix_test)

for (x in shap.importance(shap, names_only = TRUE)[1:n_plots]) {
  p <- shap.plot.dependence(
    shap,
    x = x,
    color_feature = "auto",
```

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

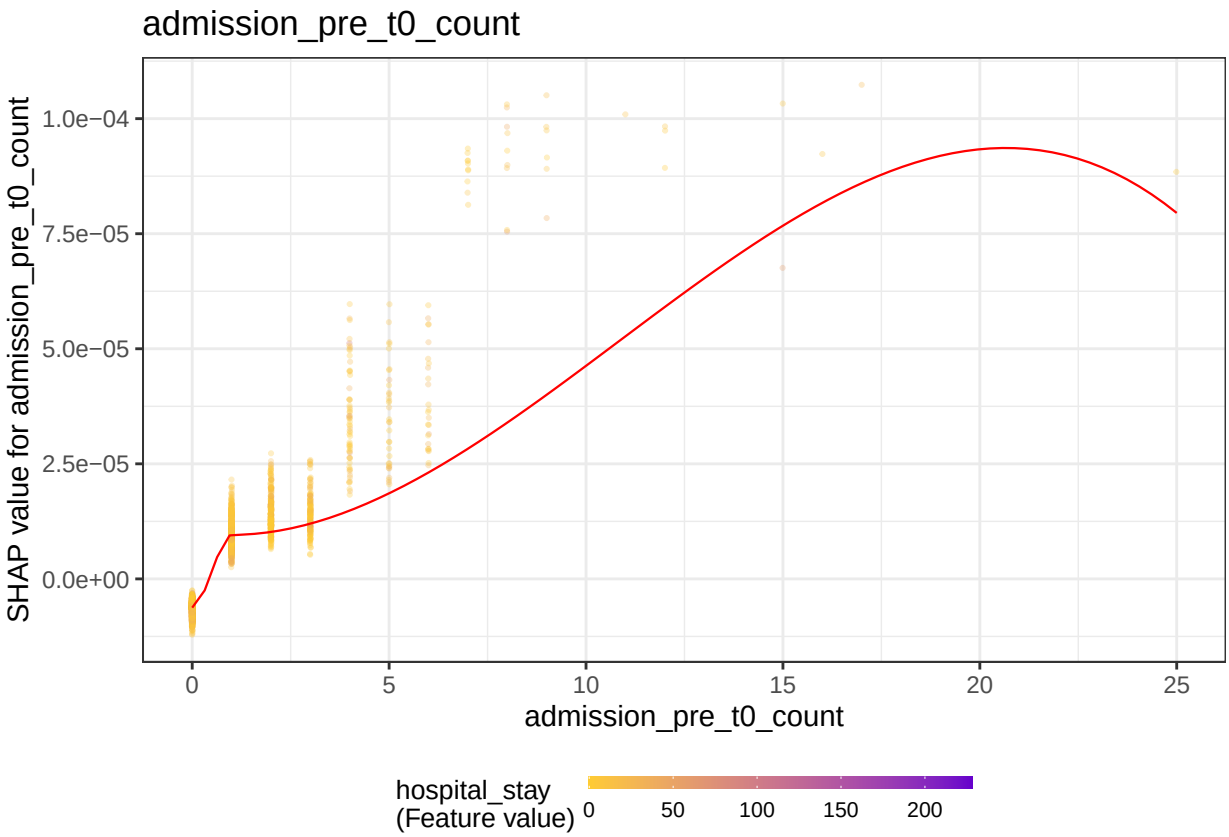
```
## `geom_smooth()` using formula 'y ~ x'
```



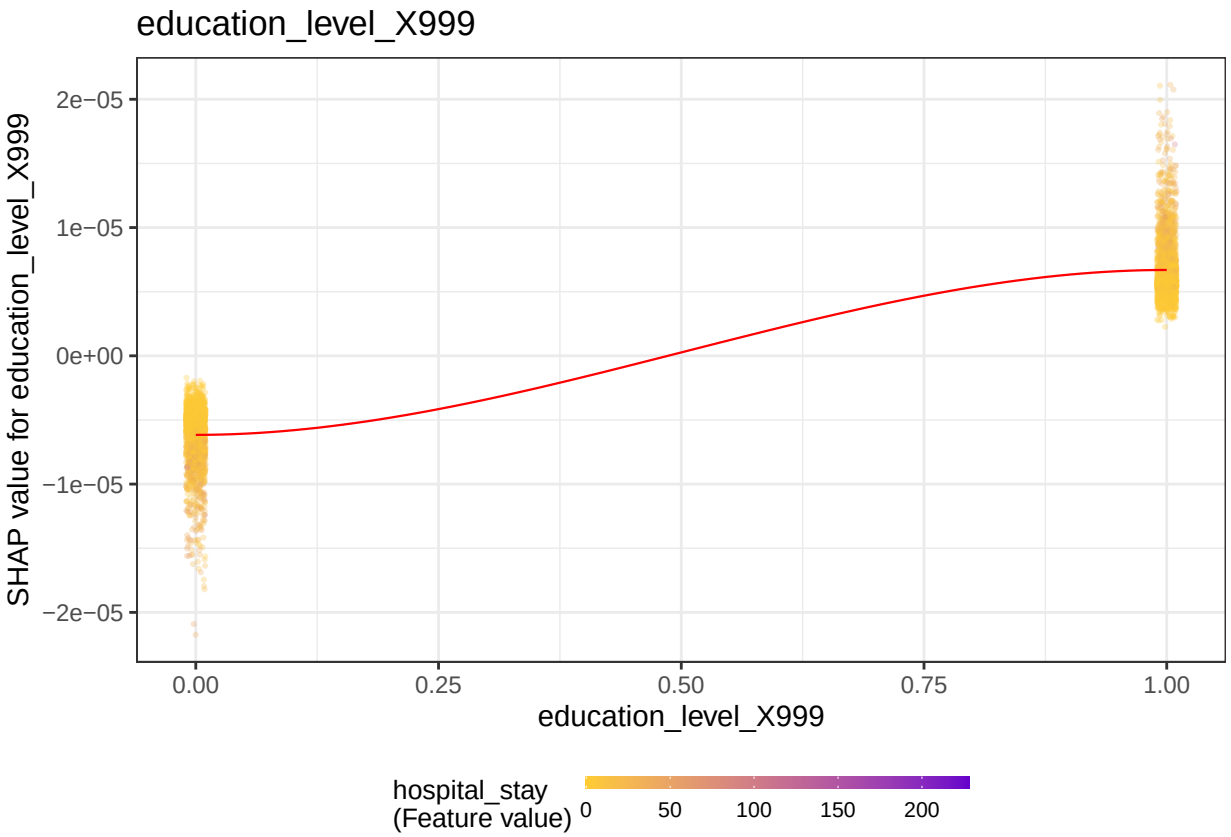

```
## `geom_smooth()` using formula 'y ~ x'
## Warning: Removed 1070 rows containing non-finite values (stat_smooth).
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : at -1.02
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : radius 1.0404
## 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
## -1.02
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : neighborhood radius
## 1.02
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : reciprocal condition
## number 1
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : zero-width
## neighborhood. make span bigger
## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 5)
## Warning: Removed 1070 rows containing missing values (geom_point).
```



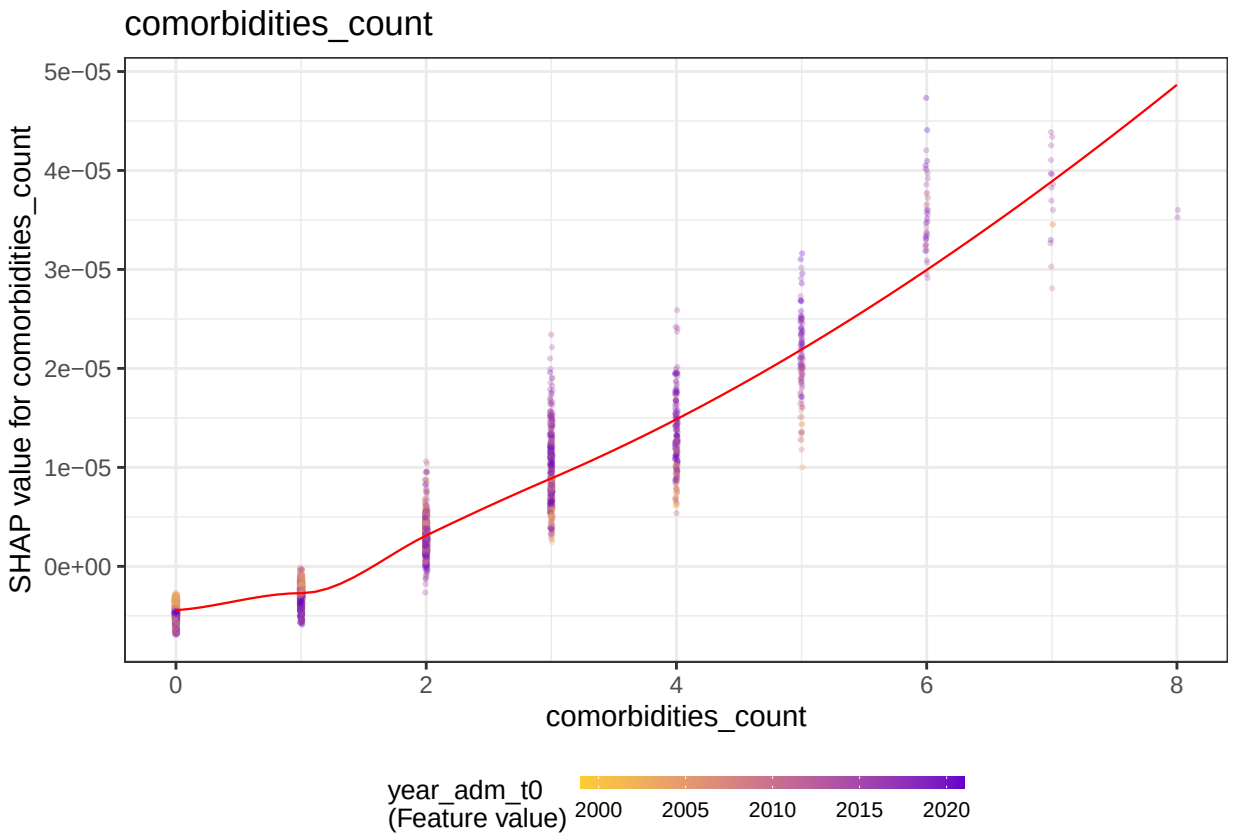

```
## `geom_smooth()` using formula 'y ~ x'

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

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

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : reciprocal condition
## number 1.3776e-27

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




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

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

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

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : reciprocal condition
## number 2.1666e-28

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

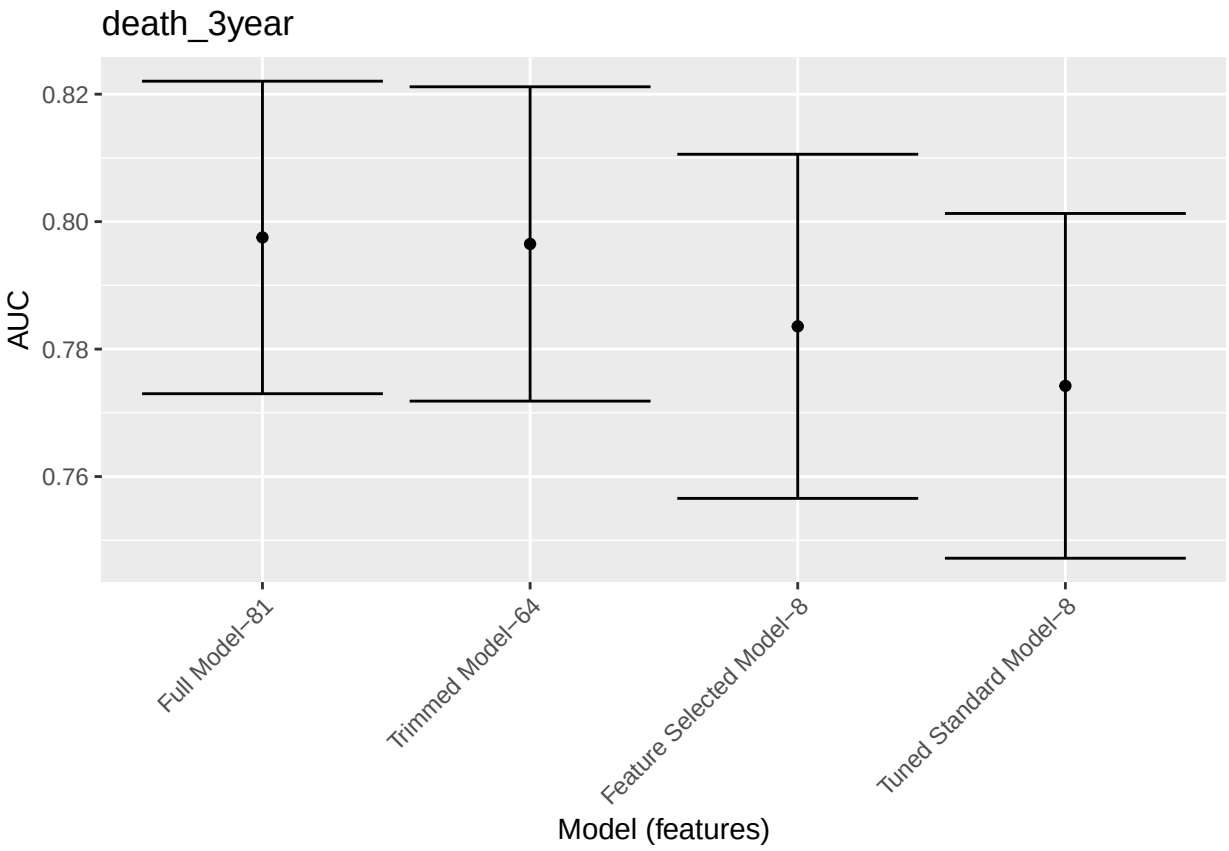



```
## `geom_smooth()` using formula 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : pseudoinverse used at
## 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : neighborhood radius 2

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : reciprocal condition
## number 4.2007e-15

## 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
  '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))
```