Final Model - death_1year

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. education_level
4. underlying_heart_disease
5. heart_disease
6. nyha_basal
7. hypertension
8. prior_mi
9. heart_failure
10. af
11. cardiac_arrest
12. valvopathy

13. diabetes
14. renal_failure
15. hemodialysis
16. stroke
17. copd
18. cancer
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. psicofarmacos
44. antifungico
45. antiviral
46. classe_meds_qtde
47. meds_cardiovasc_qtde
48. meds_antimicrobianos
49. vni
50. ventilacao_mecanica
51. transplante_cardiaco
52. cir_toracica
53. outros_proced_cirurgicos
54. icp
55. cateterismo
56. cateter_venoso_central
57. proced_invasivos_qtde
58. transfusao
59. interconsulta
60. equipe_multiprof
61. holter
62. teste_esforco
63. tilt_teste
64. metodos_graficos_qtde
65. laboratorio
66. cultura
67. analises_clinicas_qtde
68. citologia
69. histopatologia_qtde
70. angio_tc
71. angiografia
72. aortografia
73. cintilografia

74. ecocardiograma
75. endoscopia
76. flebografia
77. pet_ct
78. ultrassom
79. tomografia
80. ressonancia
81. exames_imagem_qtde
82. bic
83. 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: 0.806"
```

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

```
## [1] "Full Model Test AUC: 0.810"
```

Features with zero importance on the initial model:

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

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

1. cardiac_arrest
2. vni
3. cir_toracica
4. transfusao
5. teste_esforco
6. aortografia
7. 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: 0.805"
```

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

```
## [1] "Trimmed Model Test AUC: 0.814"
```

```
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
)
```

```r
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] "75"         "-0.0011"    "-0.0023"    "endoscopia"
## [1] "74"         "-0.0012"    "-1e-04"     "tilt_teste"
## [1] "73"       "-7e-04"   "6e-04"    "angio_tc"
## [1] "72"                        "2e-04"                    "9e-04"
## [4] "underlying_heart_disease"
## [1] "71"               "3e-04"            "1e-04"            "procedure_type_1"
## [1] "70"          "-0.0012"     "-0.0015"     "dialysis_t0"
## [1] "69"        "-0.0015"   "-3e-04"    "antiviral"
## [1] "68"           "-9e-04"        "6e-04"         "heart_disease"
## [1] "68"          "0.0018"      "0.0028"      "antifungico"
## [1] "67"      "-0.0014" "-5e-04"  "icp"
## [1] "66"     "-0.001" "4e-04"  "holter"
## [1] "65"                "-0.0024"            "-0.0013"            "antiplaquetario_ev"
## [1] "64"          "-0.002"      "4e-04"       "ressonancia"
## [1] "63"                  "-4e-04"              "0.0016"               "transplante_cardiaco"
## [1] "62"                 "-0.004"              "-0.0036"             "histopatologia_qtde"
## [1] "62"          "-0.0019"     "0.0021"      "flebografia"
## [1] "61"          "-0.0023"     "0.0016"      "angiografia"
## [1] "61"                     "9e-04"                  "0.0032"                 "cateter_venoso_central"
## [1] "60"           "-0.0025"      "-2e-04"       "cied_final_1"
## [1] "60"     "0.0024" "0.0049" "bic"
## [1] "59"         "-0.003"     "-5e-04"     "nyha_basal"
## [1] "59"       "-1e-04"   "0.0028"   "insulina"
## [1] "59"                "5e-04"              "0.0034"             "procedure_type_new"
## [1] "58"       "-0.0026"  "4e-04"    "prior_mi"
## [1] "58"     "-1e-04" "0.0025" "copd"
## [1] "57"           "-0.0017"      "9e-04"        "hemodialysis"
## [1] "56"            "-8e-04"        "9e-04"         "heart_failure"
## [1] "55"     "4e-04"  "0.0013" "aco"
## [1] "54"       "-9e-04"   "-0.0014"  "digoxina"
## [1] "53"     "5e-04"  "0.0014" "stroke"
## [1] "52"            "0.0019"        "0.0014"        "interconsulta"
## [1] "51"      "1e-04"   "-0.0018" "cultura"
## [1] "51"     "0.0032" "0.0031" "sex"
## [1] "50"          "4e-04"       "3e-04"       "reop_type_1"
## [1] "49"     "0.0017" "0.0013" "cancer"
## [1] "48"                       "-0.002"                  "-0.0037"
## [4] "outros_proced_cirurgicos"
## [1] "48"            "0.0013"        "0.0033"        "renal_failure"
## [1] "47"            "-0.002"        "-1e-04"        "cintilografia"
## [1] "47"       "0.0029"   "0.005"    "diabetes"
## [1] "46"          "-0.001"      "0.001"       "cateterismo"
## [1] "45"           "-0.0021"      "-0.0011"      "hypertension"
## [1] "44"                    "-0.0021"               "0"                     "admission_pre_t0_180d"
## [1] "43"        "-5e-04"    "0.0016"    "ultrassom"
## [1] "42"                  "0"                   "5e-04"               "ventilacao_mecanica"
## [1] "41"         "-5e-04"     "-5e-04"     "valvopathy"
## [1] "40"             "8e-04"          "0.0013"         "ecocardiograma"
## [1] "39"         "3e-04"      "-5e-04"     "tomografia"
## [1] "39"                      "0.0043"                 "0.004"                  "admission_t0_emergency"
## [1] "39"        "0.0032"    "0.0029"    "citologia"
## [1] "38"     "0.002"  "0.0017" "af"
## [1] "37"                    "0.0033"                "0.0013"                "proced_invasivos_qtde"
## [1] "36"     "0.0038" "4e-04"  "dva"
## [1] "36"                 "0.0069"             "0.0031"             "cied_final_group_1"
## [1] "35"                 "0.0044"             "6e-04"              "exames_imagem_qtde"
## [1] "34"            "0.0061"        "0.0017"        "antiarritmico"
## [1] "33"             "0.0048"         "-0.0013"        "insuf_cardiaca"
```

```
## [1] "33"              "0.0113"          "0.0065"          "education_level"
```

```
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	83	0.8060	0.0128	0.0000	0.0000
All unimportant	TRUE	76	0.8049	0.0109	0.0011	0.0011
endoscopia	TRUE	75	0.8072	0.0097	-0.0011	-0.0023
tilt_teste	TRUE	74	0.8073	0.0096	-0.0012	-0.0001
angio_tc	TRUE	73	0.8067	0.0105	-0.0007	0.0006
underlying_heart_disease	TRUE	72	0.8058	0.0113	0.0002	0.0009
procedure_type_1	TRUE	71	0.8057	0.0104	0.0003	0.0001
dialysis_t0	TRUE	70	0.8072	0.0111	-0.0012	-0.0015
antiviral	TRUE	69	0.8075	0.0109	-0.0015	-0.0003
heart_disease	TRUE	68	0.8069	0.0107	-0.0009	0.0006
antifungico	FALSE	67	0.8042	0.0121	0.0018	0.0028
icp	TRUE	67	0.8074	0.0103	-0.0014	-0.0005
holter	TRUE	66	0.8070	0.0103	-0.0010	0.0004
antiplaquetario_ev	TRUE	65	0.8084	0.0112	-0.0024	-0.0013
ressonancia	TRUE	64	0.8080	0.0123	-0.0020	0.0004
transplante_cardiaco	TRUE	63	0.8064	0.0110	-0.0004	0.0016
histopatologia_qtde	TRUE	62	0.8100	0.0117	-0.0040	-0.0036
flebografia	FALSE	61	0.8079	0.0118	-0.0019	0.0021
angiografia	TRUE	61	0.8083	0.0113	-0.0023	0.0016
cateter_venoso_central	FALSE	60	0.8051	0.0106	0.0009	0.0032
cied_final_1	TRUE	60	0.8085	0.0095	-0.0025	-0.0002
bic	FALSE	59	0.8036	0.0104	0.0024	0.0049
nyha_basal	TRUE	59	0.8090	0.0102	-0.0030	-0.0005
insulina	FALSE	58	0.8061	0.0125	-0.0001	0.0028
procedure_type_new	FALSE	58	0.8055	0.0115	0.0005	0.0034
prior_mi	TRUE	58	0.8086	0.0091	-0.0026	0.0004
copd	FALSE	57	0.8061	0.0117	-0.0001	0.0025
hemodialysis	TRUE	57	0.8077	0.0116	-0.0017	0.0009
heart_failure	TRUE	56	0.8069	0.0102	-0.0008	0.0009
aco	TRUE	55	0.8056	0.0116	0.0004	0.0013
digoxina	TRUE	54	0.8070	0.0111	-0.0009	-0.0014
stroke	TRUE	53	0.8055	0.0117	0.0005	0.0014
interconsulta	TRUE	52	0.8041	0.0093	0.0019	0.0014
cultura	TRUE	51	0.8059	0.0107	0.0001	-0.0018
sex	FALSE	50	0.8029	0.0111	0.0032	0.0031
reop_type_1	TRUE	50	0.8056	0.0096	0.0004	0.0003
cancer	TRUE	49	0.8043	0.0113	0.0017	0.0013
outros_proced_cirurgicos	TRUE	48	0.8080	0.0113	-0.0020	-0.0037
renal_failure	FALSE	47	0.8047	0.0120	0.0013	0.0033
cintilografia	TRUE	47	0.8081	0.0103	-0.0020	-0.0001
diabetes	FALSE	46	0.8031	0.0100	0.0029	0.0050
cateterismo	TRUE	46	0.8070	0.0118	-0.0010	0.0010
hypertension	TRUE	45	0.8081	0.0095	-0.0021	-0.0011
admission_pre_t0_180d	TRUE	44	0.8081	0.0101	-0.0021	0.0000
ultrassom	TRUE	43	0.8065	0.0101	-0.0005	0.0016
ventilacao_mecanica	TRUE	42	0.8060	0.0097	0.0000	0.0005
valvopathy	TRUE	41	0.8065	0.0098	-0.0005	-0.0005
ecocardiograma	TRUE	40	0.8052	0.0089	0.0008	0.0013
tomografia	TRUE	39	0.8057	0.0096	0.0003	-0.0005

Table 1: *(continued)*

Tested Feature	Dropped	Features	CV AUC	CV AUC Std Error	Total AUC Loss	Instant AUC Loss
admission_t0_emergency	FALSE	38	0.8017	0.0103	0.0043	0.0040
citologia	FALSE	38	0.8028	0.0092	0.0032	0.0029
af	TRUE	38	0.8040	0.0103	0.0020	0.0017
proced_invasivos_qtde	TRUE	37	0.8027	0.0091	0.0033	0.0013
dva	TRUE	36	0.8022	0.0096	0.0038	0.0004
cied_final_group_1	FALSE	35	0.7991	0.0102	0.0069	0.0031
exames_imagem_qtde	TRUE	35	0.8016	0.0105	0.0044	0.0006
antiarritmico	TRUE	34	0.7999	0.0088	0.0061	0.0017
insuf_cardiaca	TRUE	33	0.8012	0.0089	0.0048	-0.0013
education_level	FALSE	32	0.7947	0.0095	0.0113	0.0065

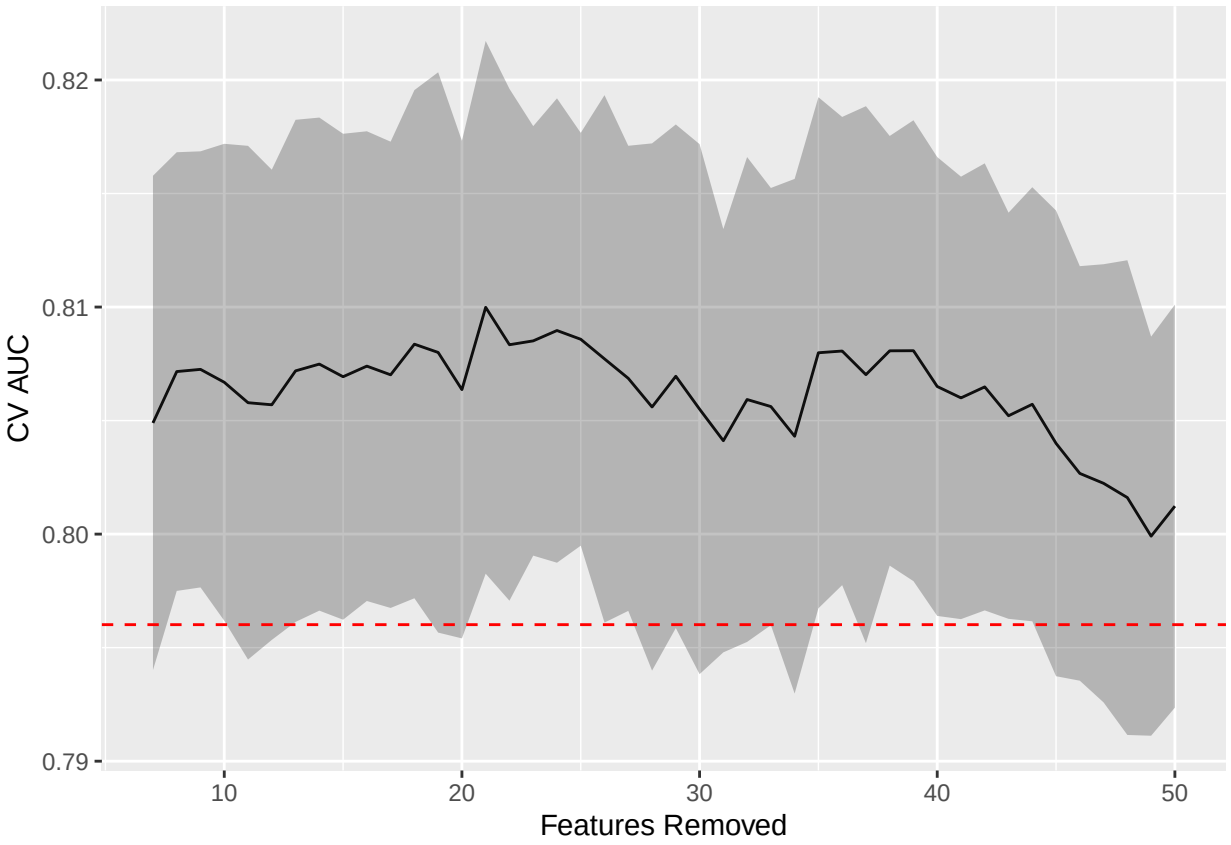
```
if (exists('last_feature_dropped')) {
  selected_features <- c(current_features, last_feature_droped)
} else {
  selected_features <- current_features
}

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: 0.800"
```

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

```
## [1] "Selected Model Test AUC: 0.800"
```

```
 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)
```

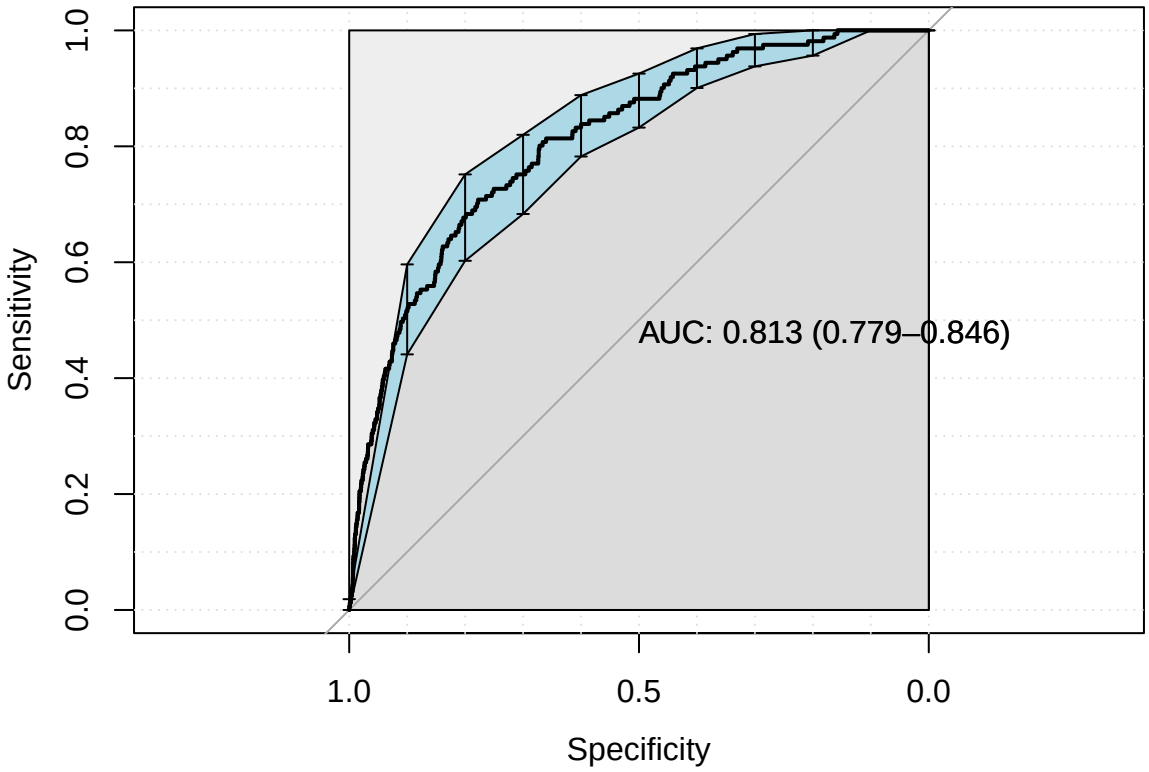
1. age
2. hospital_stay
3. year_adm_t0
4. comorbidities_count
5. admission_pre_t0_count
6. vasodilatador
7. laboratorio
8. analises_clinicas_qtde
9. espironolactona
10. metodos_graficos_qtde
11. meds_cardiovasc_qtde
12. classe_meds_qtde
13. psicofarmacos
14. meds_antimicrobianos
15. ieca_bra
16. icu_t0
17. diuretico
18. estatina
19. equipe_multiprof
20. education_level
21. cied_final_group_1
22. admission_t0_emergency
23. citologia
24. diabetes
25. renal_failure
26. sex
27. procedure_type_new

28. copd
29. bic
30. insulina
31. cateter_venoso_central
32. antifungico
33. flebografia

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)
```



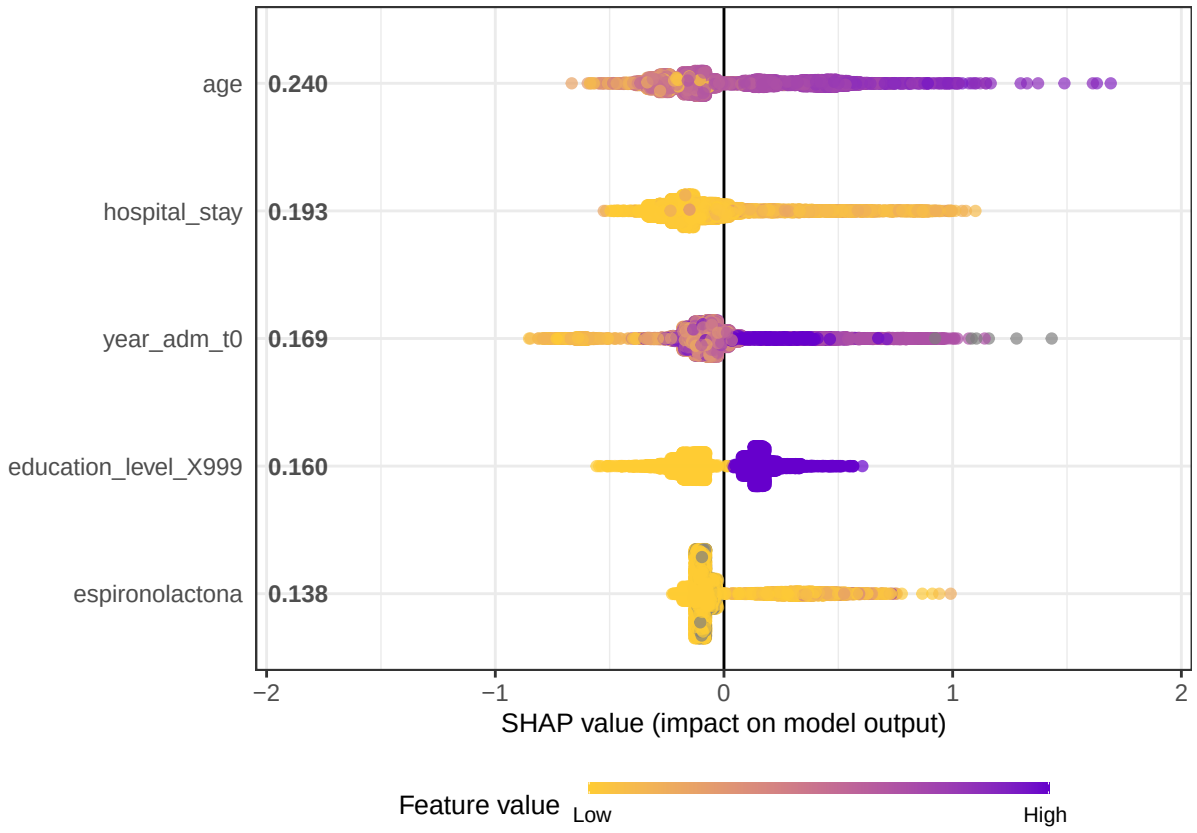

```
## [1] "Optimal Threshold: 0.03"
## Confusion Matrix and Statistics
##
##     reference
## data    0    1
##    0 3553   47
##    1 1016  114
##
##                Accuracy : 0.7753
##                  95% CI : (0.7631, 0.7871)
##     No Information Rate : 0.966
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1244
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7776
##             Specificity : 0.7081
##          Pos Pred Value : 0.9869
##          Neg Pred Value : 0.1009
##              Prevalence : 0.9660
##          Detection Rate : 0.7512
##    Detection Prevalence : 0.7611
##       Balanced Accuracy : 0.7429
##
##        '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

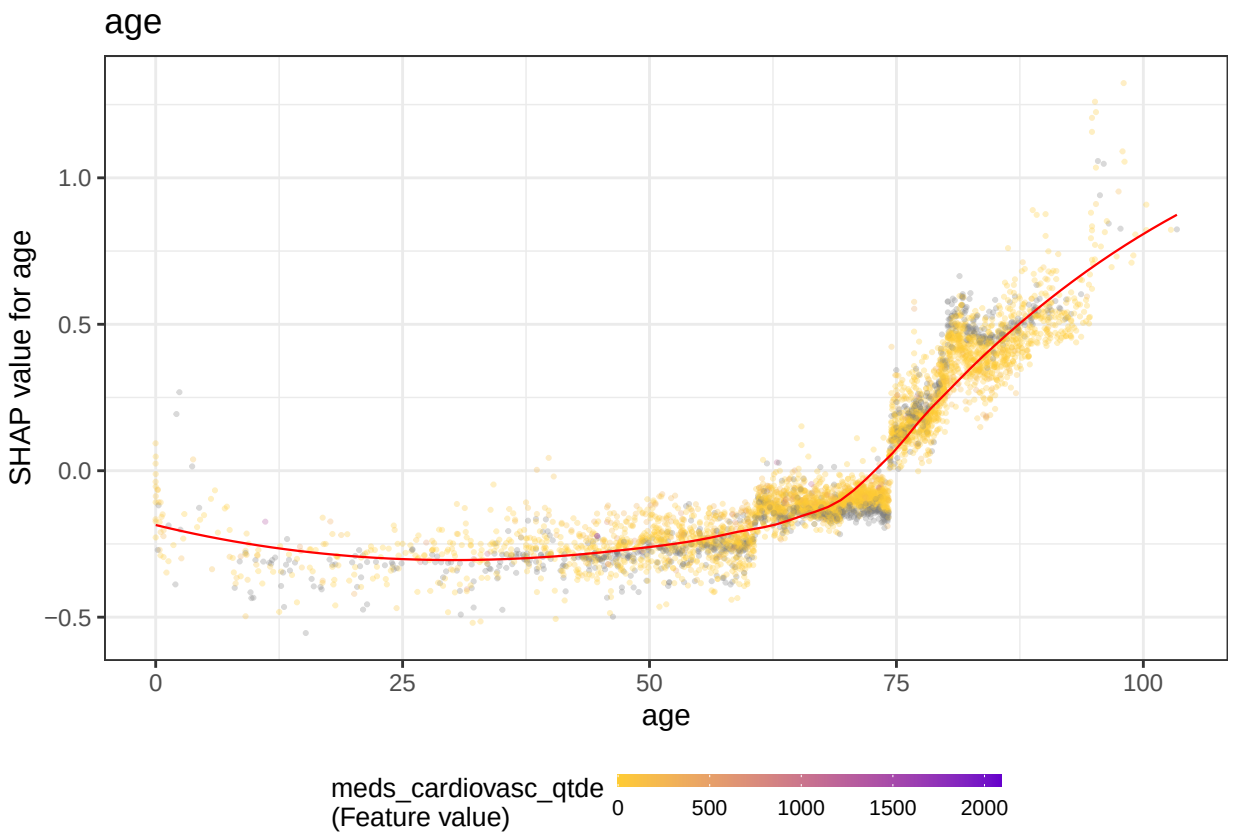
```
lightgbm_model <- parsnip::extract_fit_engine(final_lightgbm_fit)

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)
```



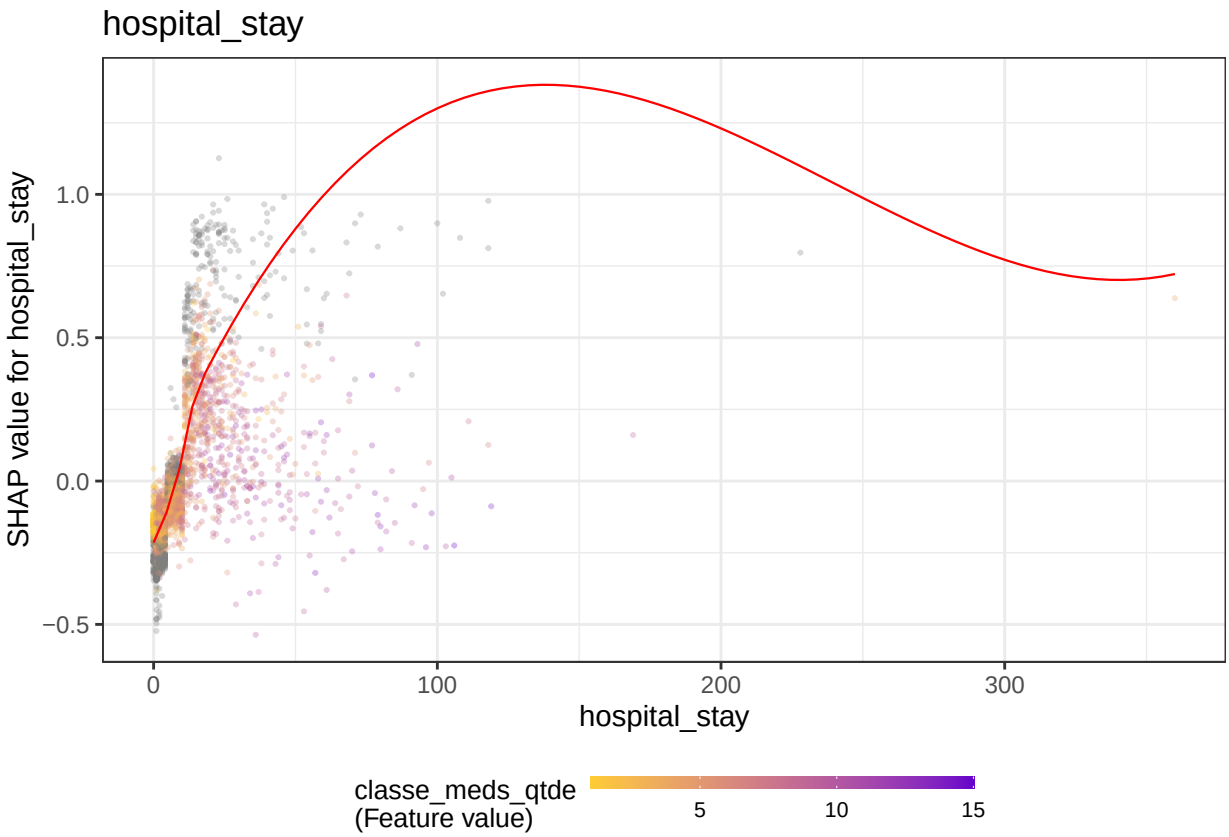

```
shap <- shap.prep(lightgbm_model, X_train = matrix_test)

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)
}
```

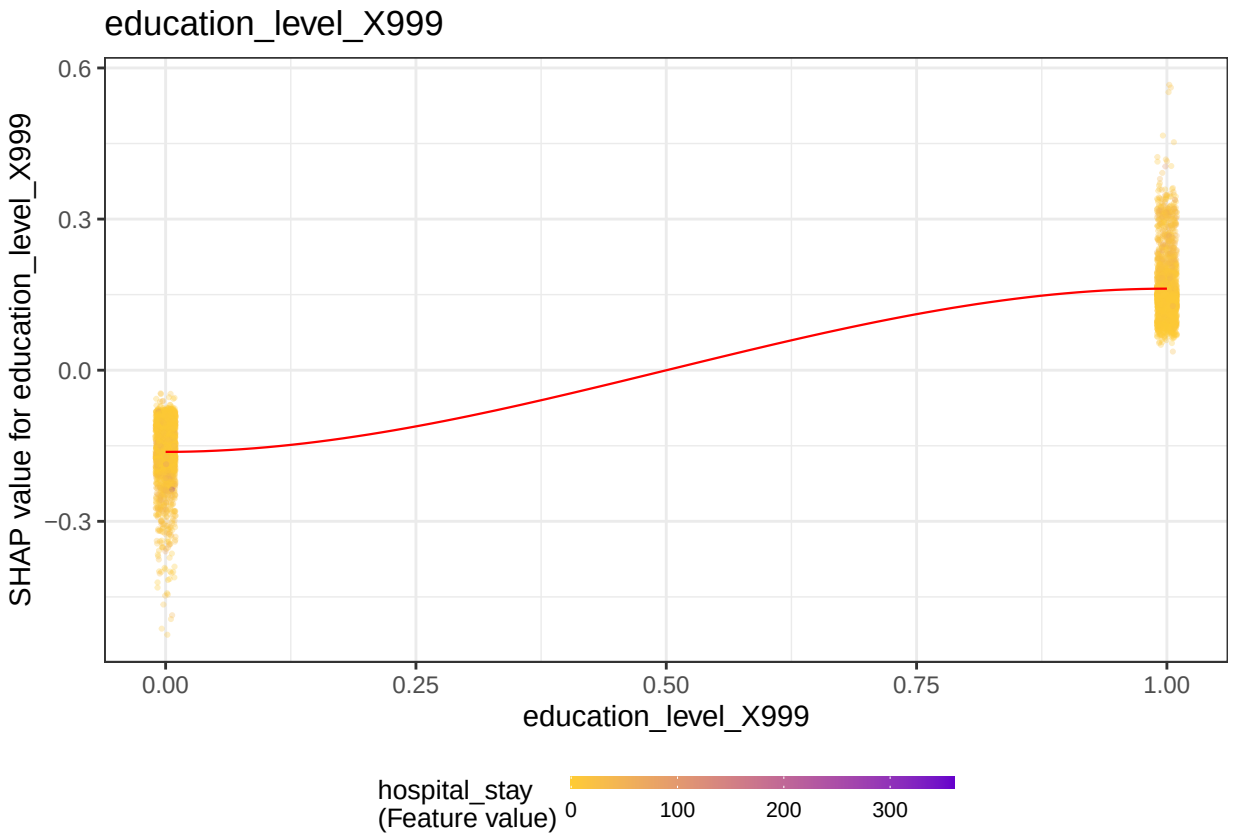
```
## `geom_smooth()` using formula 'y ~ x'
```



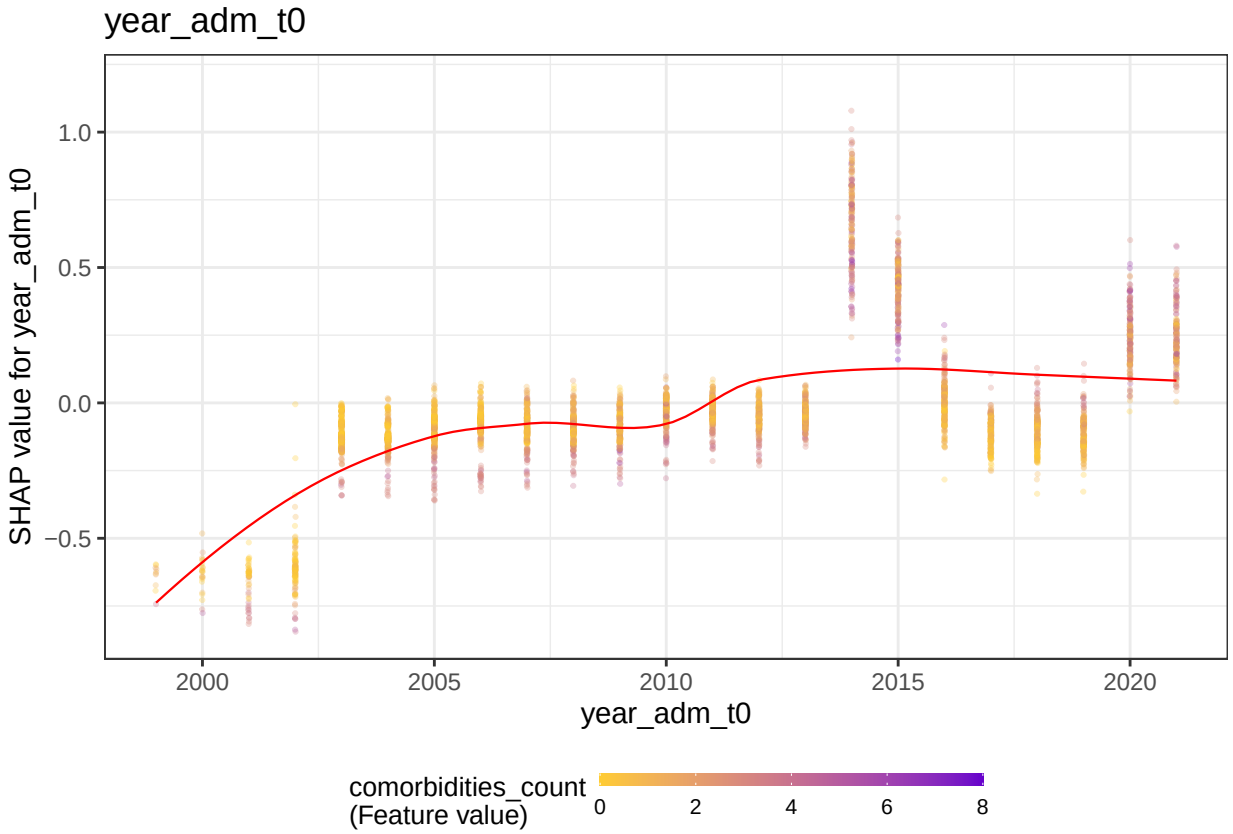

```
## `geom_smooth()` using formula 'y ~ x'
```




```
## `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 4.7919e-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: Removed 10 rows containing non-finite values (stat_smooth).
## Warning: Removed 10 rows containing missing values (geom_point).
```




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

```
## Warning: Removed 1044 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
```

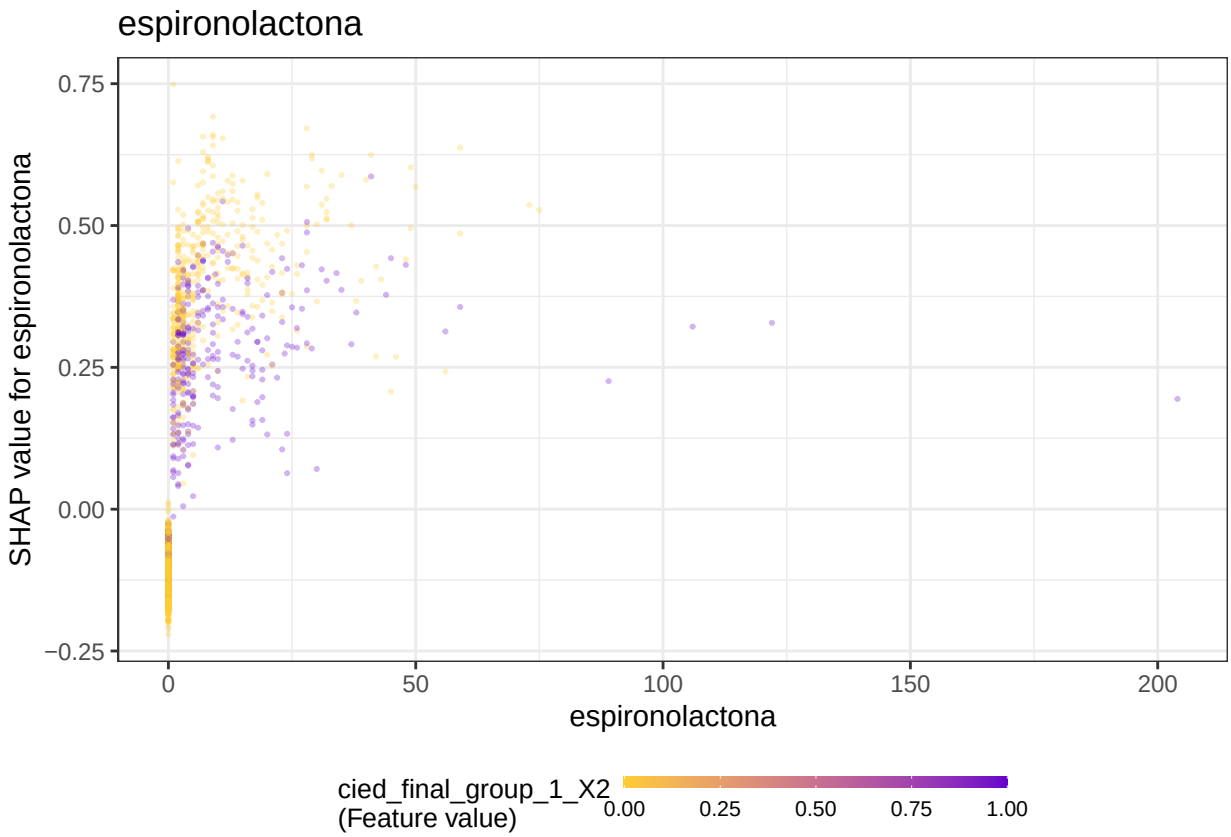
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, : neighborhood radius
## 1.02
```

```
## 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)
```

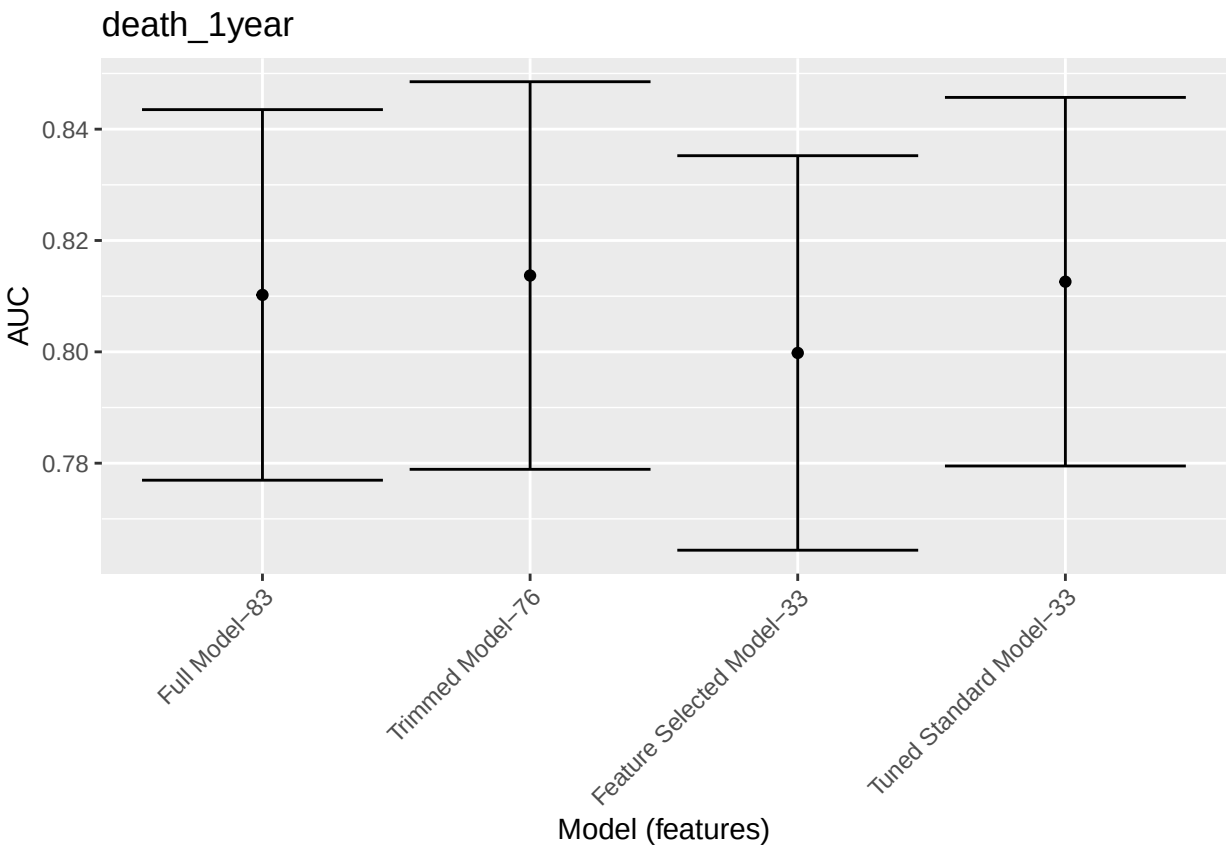
```
## Warning: Removed 1044 rows containing missing values (geom_point).
```



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))
```