Final Model - death_180days

Eduardo Yuki Yada

Global parameters

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

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. classe_meds_qtde
46. meds_cardiovasc_qtde
47. meds_antimicrobianos
48. vni
49. ventilacao_mecanica
50. transplante_cardiaco
51. outros_proced_cirurgicos
52. icp
53. cateterismo
54. cateter_venoso_central
55. proced_invasivos_qtde
56. transfusao
57. interconsulta
58. equipe_multiprof
59. holter
60. teste_esforco
61. metodos_graficos_qtde
62. laboratorio
63. cultura
64. analises_clinicas_qtde
65. citologia
66. histopatologia_qtde
67. angiografia
68. aortografia
69. arteriografia
70. cintilografia
71. ecocardiograma
72. endoscopia

73. ultrassom
74. tomografia
75. ressonancia
76. exames_imagem_qtde
77. bic
78. hospital_stay

Train test split (70%/30%)

```
set.seed(42)

if (outcome_column == 'readmission_30d') {
  df_split <- readRDS("dataset/split_object.rds")
} else {
  df_split <- initial_split(df, prop = .7, strata = all_of(outcome_column))
}

df_train <- training(df_split) %>% select(all_of(c(features, outcome_column)))
df_test <- testing(df_split) %>% select(all_of(c(features, outcome_column)))

df_folds <- vfold_cv(df_train, v = k,
                     strata = all_of(outcome_column),
                     repeats = repeats)
```

Feature Selection

```
custom_dummy_names <- function(var, lvl, ordinal = FALSE) {
  dummy_names(var, lvl, ordinal = FALSE, sep = "__")
}

model_fit_wf <- function(df_train, features, outcome_column, hyperparameters){
  model_recipe <-
    recipe(formula = sprintf("%s ~ .", outcome_column) %>% as.formula,
           data = df_train %>% select(all_of(c(features, outcome_column)))) %>%
    step_novel(all_nominal_predictors()) %>%
    step_unknown(all_nominal_predictors()) %>%
    step_other(all_nominal_predictors(), threshold = 0.05, other = ".merged") %>%
    step_dummy(all_nominal_predictors(), naming = custom_dummy_names)

  model_spec <-
    do.call(boost_tree, hyperparameters) %>%
    set_engine("lightgbm") %>%
    set_mode("classification")

  model_workflow <-
    workflow() %>%
    add_recipe(model_recipe) %>%
    add_model(model_spec)

  model_fit_rs <- model_workflow %>%
    fit_resamples(df_folds)

  model_fit <- model_workflow %>%
    fit(df_train)

  model_auc <- validation(model_fit, df_test, plot = F)

  raw_model <- parsnip::extract_fit_engine(model_fit)
```

```
  feature_importance <- lgb.importance(raw_model, percentage = TRUE) %>%
    separate(Feature, c("Feature","value"), "__", fill = 'right') %>%
    group_by(Feature) %>%
    summarise(Gain = sum(Gain),
              Cover = sum(Cover),
              Frequency = sum(Frequency)) %>%
    ungroup() %>%
    arrange(desc(Gain))

  cv_results <- collect_metrics(model_fit_rs) %>% filter(.metric == 'roc_auc')

  return(
    list(
      cv_auc =  cv_results$mean,
      cv_auc_std_err = cv_results$std_err,
      importance = feature_importance,
      auc = as.numeric(model_auc$auc),
      auc_lower = model_auc$ci[1],
      auc_upper = model_auc$ci[3]
    )
  )
}
```

```
hyperparameters <- 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: 0.795"
```

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

```
## [1] "Full Model Test AUC: 0.838"
```

```
# full_model$importance %>%
#   filter(str_detect(Feature, 'education'))
#
# full_model$importance %>%
#   filter(str_detect(Feature, 'education')) %>%
#   summarise(across(where(is.numeric), ~ sum(.x, na.rm = TRUE)))
#
# full_model$importance %>%
#   separate(Feature, c("Feature","value"), "__") %>%
#   group_by(Feature) %>%
#   summarise(Gain = sum(Gain),
#             Cover = sum(Cover),
#             Frequency = sum(Frequency))
```

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. hypertension
3. prior_mi
4. heart_failure
5. cardiac_arrest
6. valvopathy
7. renal_failure
8. copd
9. reop_type_1
10. procedure_type_new
11. dialysis_t0
12. vni
13. transplante_cardiaco
14. outros_proced_cirurgicos
15. icp
16. cateter_venoso_central
17. transfusao
18. holter
19. histopatologia_qtde
20. angiografia
21. arteriografia

```
trimmed_features <- full_model$importance$Feature
trimmed_model <- model_fit_wf(df_train, trimmed_features,
                              outcome_column, hyperparameters)

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

```
## [1] "Trimmed Model CV Train AUC: 0.797"
```

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

```
## [1] "Trimmed Model Test AUC: 0.838"
```

```
selection_results <- tibble::tribble(
  ~`Tested Feature`, ~`Dropped`, ~`Number of Features`, ~`CV AUC`, ~`CV AUC Std Error`, ~`Total AUC Loss`, ~`Ins
  'None', TRUE, length(features), full_model$cv_auc, full_model$cv_auc_std_err, 0, 0
)

whitelist <- c()

if (full_model$cv_auc - trimmed_model$cv_auc < max_auc_loss) {
  current_features <- trimmed_features
  current_model <- trimmed_model
  current_auc_loss <- full_model$cv_auc - current_model$cv_auc
  instant_auc_loss <- full_model$cv_auc - current_model$cv_auc

  selection_results <- selection_results %>%
    add_row(`Tested Feature` = 'All unimportant',
            `Dropped` = TRUE,
            `Number of Features` = length(trimmed_features),
            `CV AUC` = current_model$cv_auc,
            `CV AUC Std Error` = current_model$cv_auc_std_err,
            `Total AUC Loss` = current_auc_loss,
            `Instant AUC Loss` = instant_auc_loss)
} else {
  current_features <- features
  current_model <- full_model
  current_auc_loss <- 0
}

while (current_auc_loss < max_auc_loss & mean(current_features %in% whitelist) < 1) {
  zero_importance_features <- setdiff(current_features, current_model$importance$Feature) %>%
```

```
    setdiff(whitelist)
  if (length(zero_importance_features) > 0) {
    current_least_important <- zero_importance_features[1]
  } else {
    current_least_important <-
      tail(setdiff(current_model$importance$Feature, whitelist), 1)
  }
  test_features <-
    setdiff(current_features, current_least_important)
  current_model <-
    model_fit_wf(df_train, test_features, outcome_column, hyperparameters)
  instant_auc_loss <-
    tail(selection_results %>% filter(Dropped) %>% .$`CV AUC`, n = 1) - current_model$cv_auc

  if (instant_auc_loss < max_auc_loss / 5 &
      current_auc_loss < max_auc_loss) {
    dropped <- TRUE
    current_features <- test_features
    current_auc_loss <- full_model$cv_auc - current_model$cv_auc
  } else {
    dropped <- FALSE
    whitelist <- c(whitelist, current_least_important)
  }

  selection_results <- selection_results %>%
    add_row(
      `Tested Feature` = current_least_important,
      `Dropped` = dropped,
      `Number of Features` = length(test_features),
      `CV AUC` = current_model$cv_auc,
      `CV AUC Std Error` = current_model$cv_auc_std_err,
      `Total AUC Loss` = current_auc_loss,
      `Instant AUC Loss` = instant_auc_loss
    )

  print(c(
    length(current_features),
    round(current_auc_loss, 4),
    round(instant_auc_loss, 4),
    current_least_important
  ))
}
```

```
## [1] "56"      "-0.0032" "-0.0011" "af"
## [1] "55"                    "-0.0022"               "0.001"
## [4] "proced_invasivos_qtde"
## [1] "54"          "-0.0023"     "-1e-04"      "aortografia"
## [1] "53"      "-0.0034" "-0.0011" "sex"
## [1] "52"       "-0.0033"  "0"        "estatina"
## [1] "51"                     "-0.004"
## [3] "-7e-04"                 "analises_clinicas_qtde"
## [1] "50"          "-0.0038"     "2e-04"       "ressonancia"
## [1] "49"        "-0.0027"   "0.0011"    "ultrassom"
## [1] "48"      "-0.0016" "0.0011"  "dva"
## [1] "47"         "-0.002"     "-4e-04"     "tomografia"
## [1] "46"       "-6e-04"   "0.0014"   "diabetes"
## [1] "45"            "0"             "6e-04"         "teste_esforco"
## [1] "44"             "-0.0018"        "-0.0018"        "ecocardiograma"
## [1] "43"               "-0.0019"          "-1e-04"
## [4] "procedure_type_1"
## [1] "42"       "-0.0018"  "1e-04"    "insulina"
```

```
## [1] "41"                 "-0.0033"            "-0.0015"
## [4] "exames_imagem_qtde"
## [1] "40"           "-0.004"       "-7e-04"       "cied_final_1"
## [1] "39"      "-0.0034" "6e-04"   "bic"
## [1] "38"          "-0.0039"     "-5e-04"      "antifungico"
## [1] "37"                     "-0.0057"
## [3] "-0.0019"                "admission_t0_emergency"
## [1] "37"                  "-0.0057"             "0.0022"
## [4] "ventilacao_mecanica"
## [1] "36"       "-0.0051"  "6e-04"    "digoxina"
## [1] "36"            "-0.0051"       "0.002"         "psicofarmacos"
## [1] "35"                       "-0.0054"
## [3] "-3e-04"                   "underlying_heart_disease"
## [1] "34"      "-0.0067" "-0.0012" "cancer"
## [1] "33"      "-0.0069" "-3e-04"  "stroke"
## [1] "32"             "-0.0069"        "0"              "insuf_cardiaca"
## [1] "31"            "-0.0065"       "4e-04"         "cintilografia"
## [1] "30"      "-0.0084" "-0.002"  "aco"
## [1] "29"                 "-0.0079"            "6e-04"
## [4] "cied_final_group_1"
## [1] "28"         "-0.006"     "0.0019"     "endoscopia"
## [1] "27"        "-0.0076"   "-0.0016"   "diuretico"
## [1] "26"           "-0.0067"      "9e-04"        "hemodialysis"
## [1] "25"               "-0.0078"          "-0.0011"
## [4] "classe_meds_qtde"
## [1] "24"      "-0.0074" "4e-04"   "cultura"
## [1] "23"                   "-0.0083"              "-9e-04"
## [4] "meds_cardiovasc_qtde"
## [1] "23"                 "-0.0083"            "0.0021"
## [4] "antiplaquetario_ev"
## [1] "23"                   "-0.0083"              "0.0021"
## [4] "meds_antimicrobianos"
## [1] "22"        "-0.0073"   "0.001"     "citologia"
## [1] "21"            "-0.0061"       "0.0012"        "interconsulta"
## [1] "20"            "-0.0055"       "6e-04"         "antiarritmico"
## [1] "20"          "-0.0055"     "0.0035"      "cateterismo"
## [1] "19"                    "-0.0045"               "9e-04"
## [4] "metodos_graficos_qtde"
## [1] "19"                    "-0.0045"               "0.0033"
## [4] "admission_pre_t0_180d"
## [1] "18"               "-0.0028"          "0.0017"
## [4] "equipe_multiprof"
## [1] "17"         "-0.0012"    "0.0016"     "nyha_basal"
## [1] "16"     "-5e-04" "7e-04"  "icu_t0"
## [1] "15"       "-0.003"   "-0.0025"  "ieca_bra"
## [1] "14"                    "-0.0022"
## [3] "8e-04"                 "admission_pre_t0_count"
## [1] "13"                  "-0.0014"             "8e-04"
## [4] "comorbidities_count"
## [1] "13"              "-0.0014"         "0.017"           "education_level"
## [1] "13"            "-0.0014"       "0.0032"        "vasodilatador"
## [1] "12"          "5e-04"       "0.0018"      "laboratorio"
## [1] "12"              "5e-04"           "0.0205"          "espironolactona"
## [1] "12"     "5e-04"  "0.0084" "age"
## [1] "12"          "5e-04"       "0.0401"      "year_adm_t0"
## [1] "12"            "5e-04"         "0.0215"        "hospital_stay"
```

```
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	78	0.7946	0.0051	0.0000	0.0000
All unimportant	TRUE	57	0.7967	0.0053	-0.0020	-0.0020
af	TRUE	56	0.7978	0.0051	-0.0032	-0.0011
proced_invasivos_qtde	TRUE	55	0.7968	0.0052	-0.0022	0.0010
aortografia	TRUE	54	0.7969	0.0052	-0.0023	-0.0001
sex	TRUE	53	0.7980	0.0052	-0.0034	-0.0011
estatina	TRUE	52	0.7980	0.0066	-0.0033	0.0000
analises_clinicas_qtde	TRUE	51	0.7986	0.0059	-0.0040	-0.0007
ressonancia	TRUE	50	0.7985	0.0064	-0.0038	0.0002
ultrassom	TRUE	49	0.7974	0.0060	-0.0027	0.0011
dva	TRUE	48	0.7963	0.0053	-0.0016	0.0011
tomografia	TRUE	47	0.7967	0.0054	-0.0020	-0.0004
diabetes	TRUE	46	0.7952	0.0057	-0.0006	0.0014
teste_esforco	TRUE	45	0.7946	0.0056	0.0000	0.0006
ecocardiograma	TRUE	44	0.7964	0.0058	-0.0018	-0.0018
procedure_type_1	TRUE	43	0.7965	0.0054	-0.0019	-0.0001
insulina	TRUE	42	0.7964	0.0053	-0.0018	0.0001
exames_imagem_qtde	TRUE	41	0.7979	0.0053	-0.0033	-0.0015
cied_final_1	TRUE	40	0.7986	0.0056	-0.0040	-0.0007
bic	TRUE	39	0.7981	0.0053	-0.0034	0.0006
antifungico	TRUE	38	0.7985	0.0056	-0.0039	-0.0005
admission_t0_emergency	TRUE	37	0.8004	0.0046	-0.0057	-0.0019
ventilacao_mecanica	FALSE	36	0.7981	0.0050	-0.0057	0.0022
digoxina	TRUE	36	0.7998	0.0056	-0.0051	0.0006
psicofarmacos	FALSE	35	0.7977	0.0054	-0.0051	0.0020
underlying_heart_disease	TRUE	35	0.8000	0.0054	-0.0054	-0.0003
cancer	TRUE	34	0.8013	0.0047	-0.0067	-0.0012
stroke	TRUE	33	0.8016	0.0045	-0.0069	-0.0003
insuf_cardiaca	TRUE	32	0.8015	0.0047	-0.0069	0.0000
cintilografia	TRUE	31	0.8011	0.0049	-0.0065	0.0004
aco	TRUE	30	0.8031	0.0042	-0.0084	-0.0020
cied_final_group_1	TRUE	29	0.8025	0.0046	-0.0079	0.0006
endoscopia	TRUE	28	0.8006	0.0041	-0.0060	0.0019
diuretico	TRUE	27	0.8023	0.0042	-0.0076	-0.0016
hemodialysis	TRUE	26	0.8013	0.0040	-0.0067	0.0009
classe_meds_qtde	TRUE	25	0.8024	0.0048	-0.0078	-0.0011
cultura	TRUE	24	0.8020	0.0051	-0.0074	0.0004
meds_cardiovasc_qtde	TRUE	23	0.8029	0.0052	-0.0083	-0.0009
antiplaquetario_ev	FALSE	22	0.8008	0.0056	-0.0083	0.0021
meds_antimicrobianos	FALSE	22	0.8008	0.0055	-0.0083	0.0021
citologia	TRUE	22	0.8019	0.0051	-0.0073	0.0010
interconsulta	TRUE	21	0.8008	0.0062	-0.0061	0.0012
antiarritmico	TRUE	20	0.8001	0.0068	-0.0055	0.0006
cateterismo	FALSE	19	0.7966	0.0067	-0.0055	0.0035
metodos_graficos_qtde	TRUE	19	0.7992	0.0071	-0.0045	0.0009
admission_pre_t0_180d	FALSE	18	0.7959	0.0071	-0.0045	0.0033
equipe_multiprof	TRUE	18	0.7974	0.0076	-0.0028	0.0017
nyha_basal	TRUE	17	0.7959	0.0071	-0.0012	0.0016
icu_t0	TRUE	16	0.7951	0.0062	-0.0005	0.0007
ieca_bra	TRUE	15	0.7976	0.0062	-0.0030	-0.0025
admission_pre_t0_count	TRUE	14	0.7968	0.0063	-0.0022	0.0008
comorbidities_count	TRUE	13	0.7960	0.0053	-0.0014	0.0008
education_level	FALSE	12	0.7790	0.0055	-0.0014	0.0170

Table 1: *(continued)*

Tested Feature	Dropped	Features	CV AUC	CV AUC Std Error	Total AUC Loss	Instant AUC Loss
vasodilatador	FALSE	12	0.7928	0.0051	-0.0014	0.0032
laboratorio	TRUE	12	0.7942	0.0055	0.0005	0.0018
espironolactona	FALSE	11	0.7736	0.0069	0.0005	0.0205
age	FALSE	11	0.7858	0.0075	0.0005	0.0084
year_adm_t0	FALSE	11	0.7541	0.0064	0.0005	0.0401
hospital_stay	FALSE	11	0.7726	0.0081	0.0005	0.0215

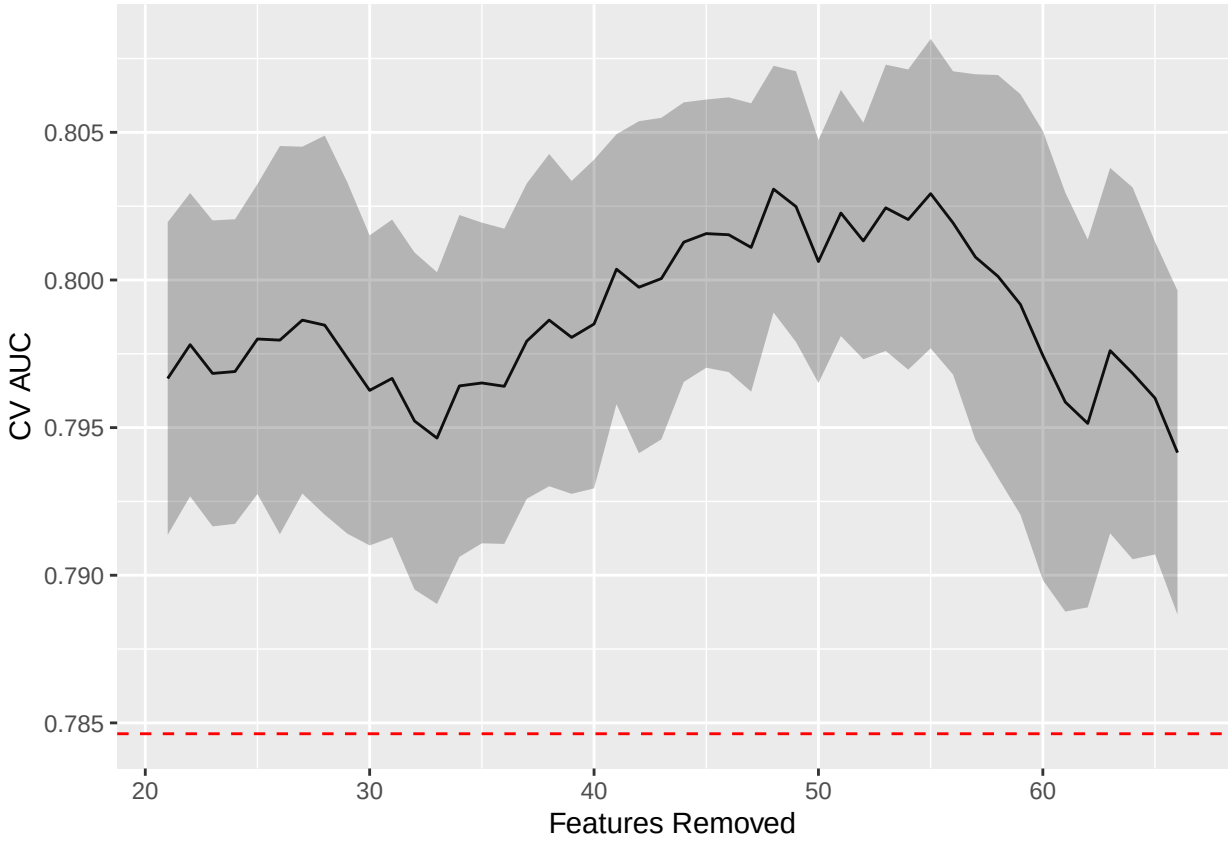
```
if (exists('last_feature_dropped')) {
  selected_features <- c(current_features, last_feature_droped)
} else {
  selected_features <- current_features
}

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: 0.794"
```

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

```
## [1] "Selected Model Test AUC: 0.818"
```

```
 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)
```

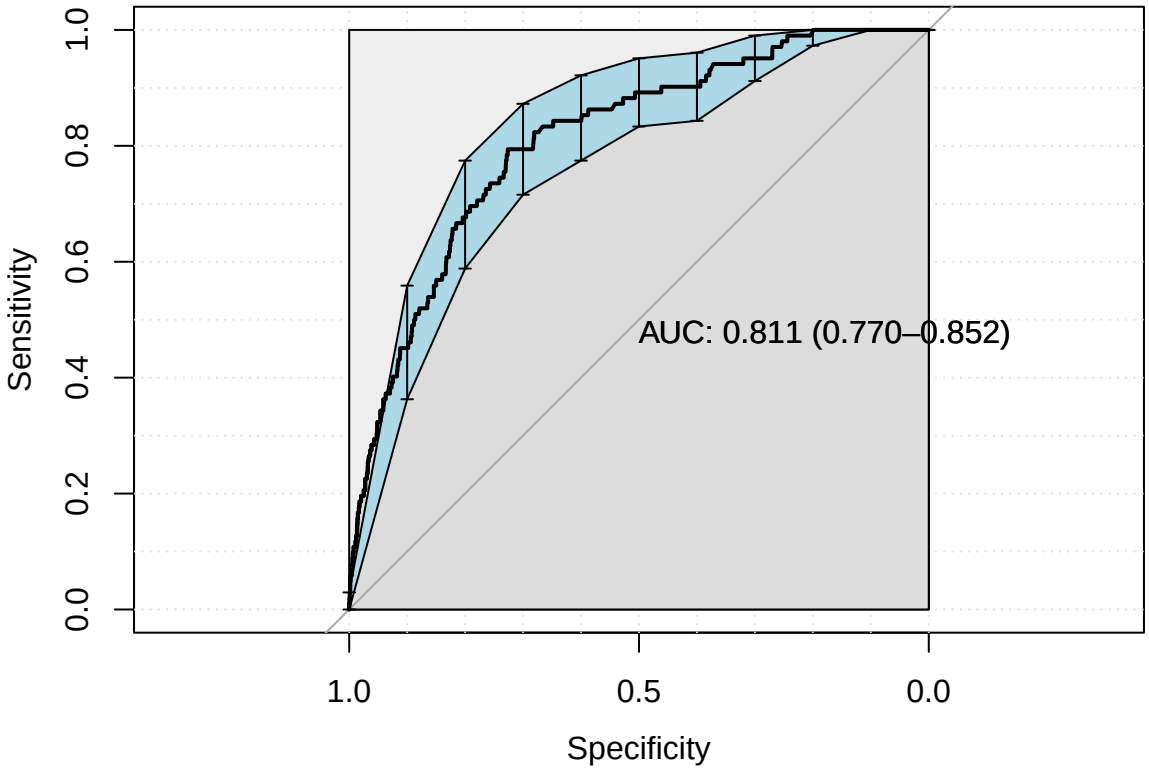
1. hospital_stay
2. age
3. year_adm_t0
4. espironolactona
5. education_level
6. vasodilatador
7. admission_pre_t0_180d
8. meds_antimicrobianos
9. cateterismo
10. antiplaquetario_ev
11. ventilacao_mecanica
12. psicofarmacos

Standard

```
lightgbm_recipe <-
  recipe(formula = sprintf("%s ~ .", outcome_column) %>% as.formula,
         data = df_train %>% select(all_of(c(selected_features, outcome_column)))) %>%
  step_novel(all_nominal_predictors()) %>%
  step_unknown(all_nominal_predictors()) %>%
  step_other(all_nominal_predictors(), threshold = 0.05, other = ".merged") %>%
  step_dummy(all_nominal_predictors())

lightgbm_tuning <- function(recipe) {

  lightgbm_spec <- boost_tree(
    trees = tune(),
```

```
    min_n = tune(),
    tree_depth = tune(),
    learn_rate = tune(),
    loss_reduction = tune(),
    sample_size = 1.0
  ) %>%
    set_engine("lightgbm") %>%
    set_mode("classification")

  lightgbm_grid <- grid_latin_hypercube(
    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")

  autoplot(lightgbm_tune, metric = "roc_auc")

  final_lightgbm_workflow <-
    lightgbm_workflow %>%
    finalize_workflow(best_lightgbm)

  last_lightgbm_fit <-
    final_lightgbm_workflow %>%
    last_fit(df_split)

  final_lightgbm_fit <- extract_workflow(last_lightgbm_fit)

  lightgbm_auc <- validation(final_lightgbm_fit, df_test)

  lightgbm_parameters <- lightgbm_tune %>%
    show_best("roc_auc", n = 1) %>%
    select(trees, min_n, tree_depth, learn_rate, 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)
```



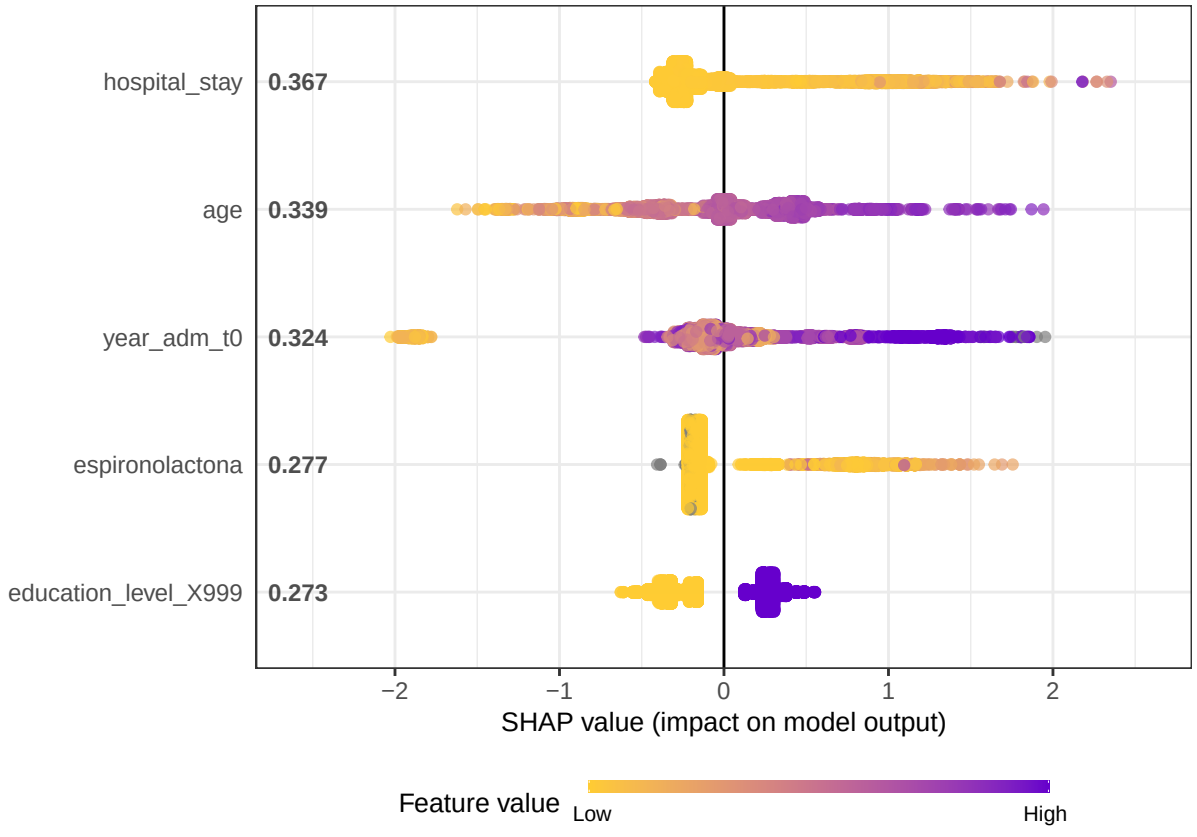

```
## [1] "Optimal Threshold: 0.02"
## Confusion Matrix and Statistics
##
##     reference
## data    0    1
##    0 3362   21
##    1 1266   81
##
##                Accuracy : 0.7279
##                  95% CI : (0.715, 0.7406)
##     No Information Rate : 0.9784
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0747
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.72645
##             Specificity : 0.79412
##          Pos Pred Value : 0.99379
##          Neg Pred Value : 0.06013
##              Prevalence : 0.97844
##          Detection Rate : 0.71078
##    Detection Prevalence : 0.71522
##       Balanced Accuracy : 0.76028
##
##        'Positive' Class : 0
##
```

```
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
  )
)
```

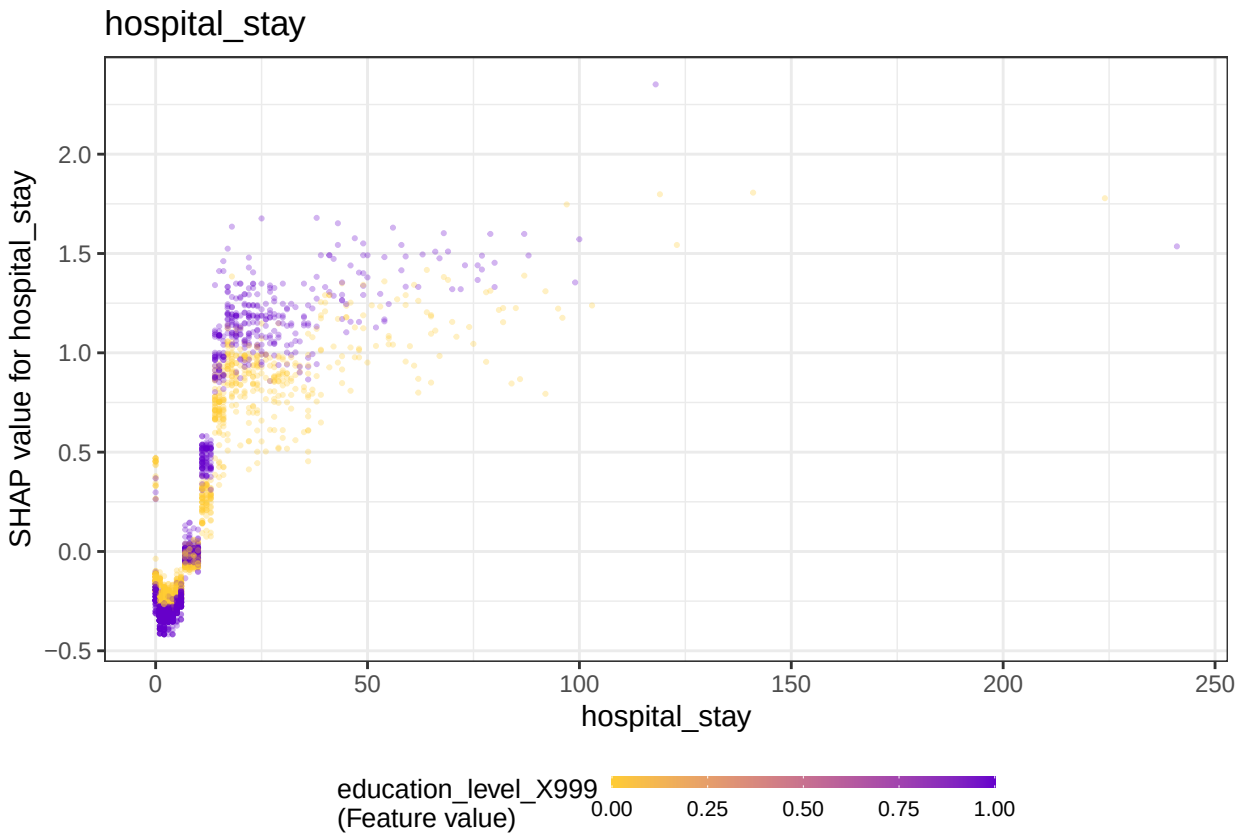
SHAP values

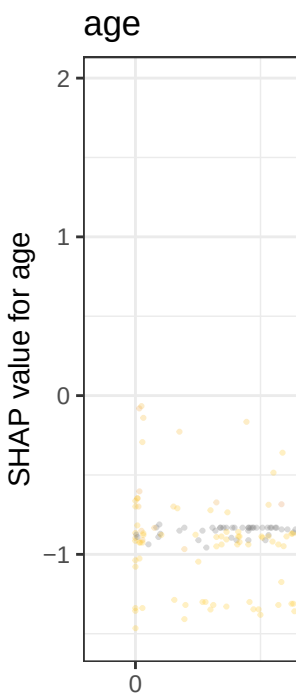
```
lightgbm_model <- parsnip::extract_fit_engine(final_lightgbm_fit)

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)
```



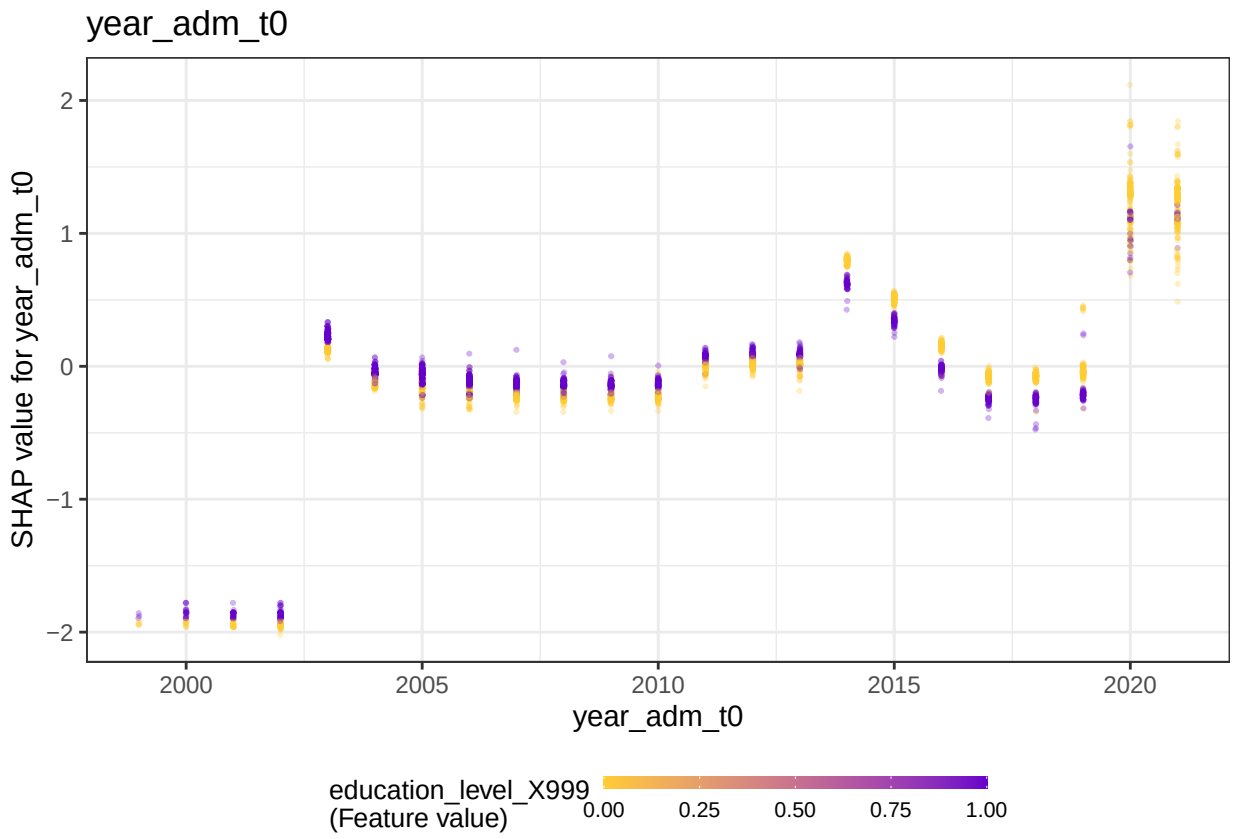

```
shap <- shap.prep(lightgbm_model, X_train = matrix_test)

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

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

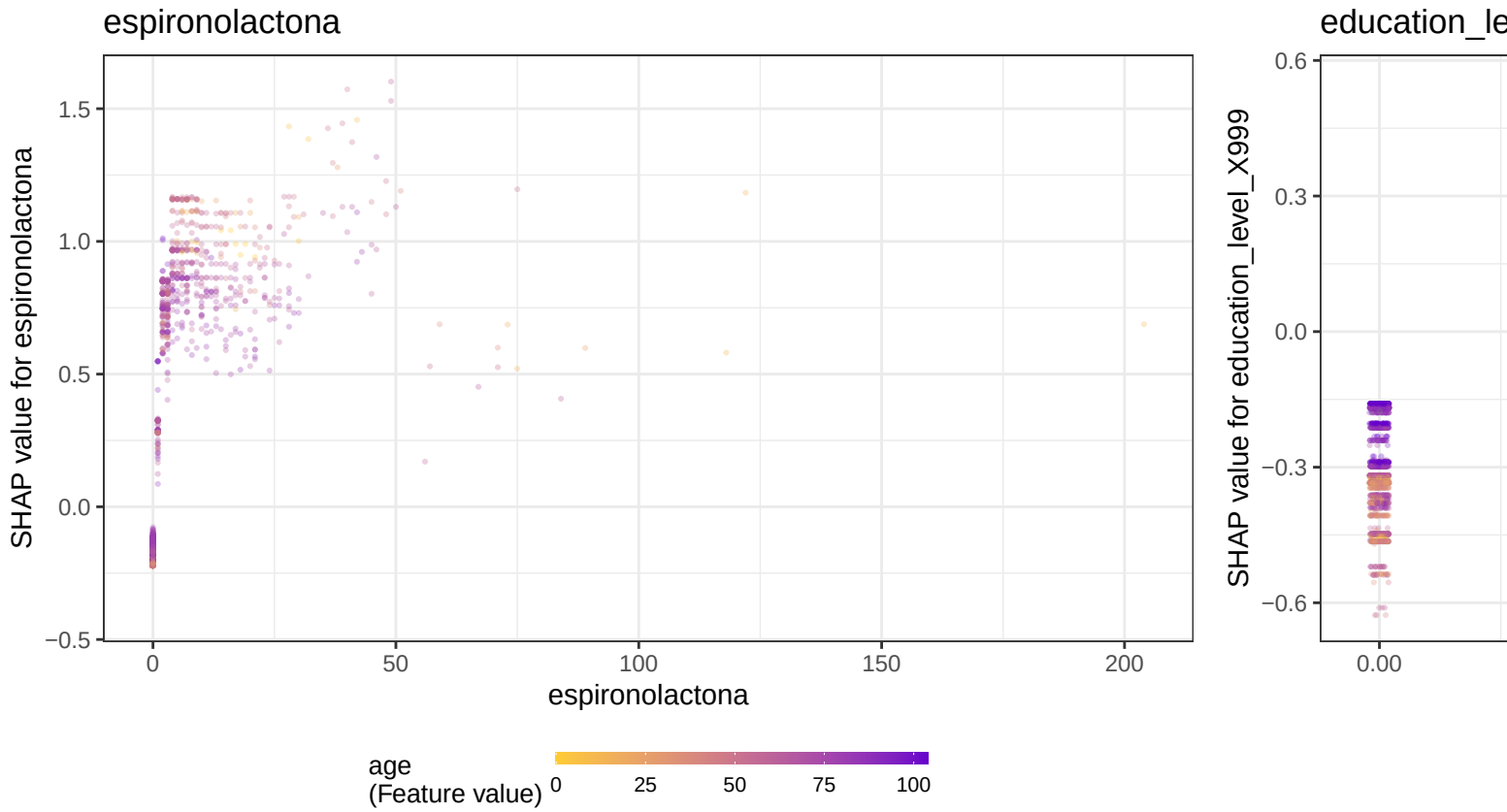





```
## Warning: Removed 7 rows containing missing values (geom_point).
```




```
## Warning: Removed 1041 rows containing missing values (geom_point).
```



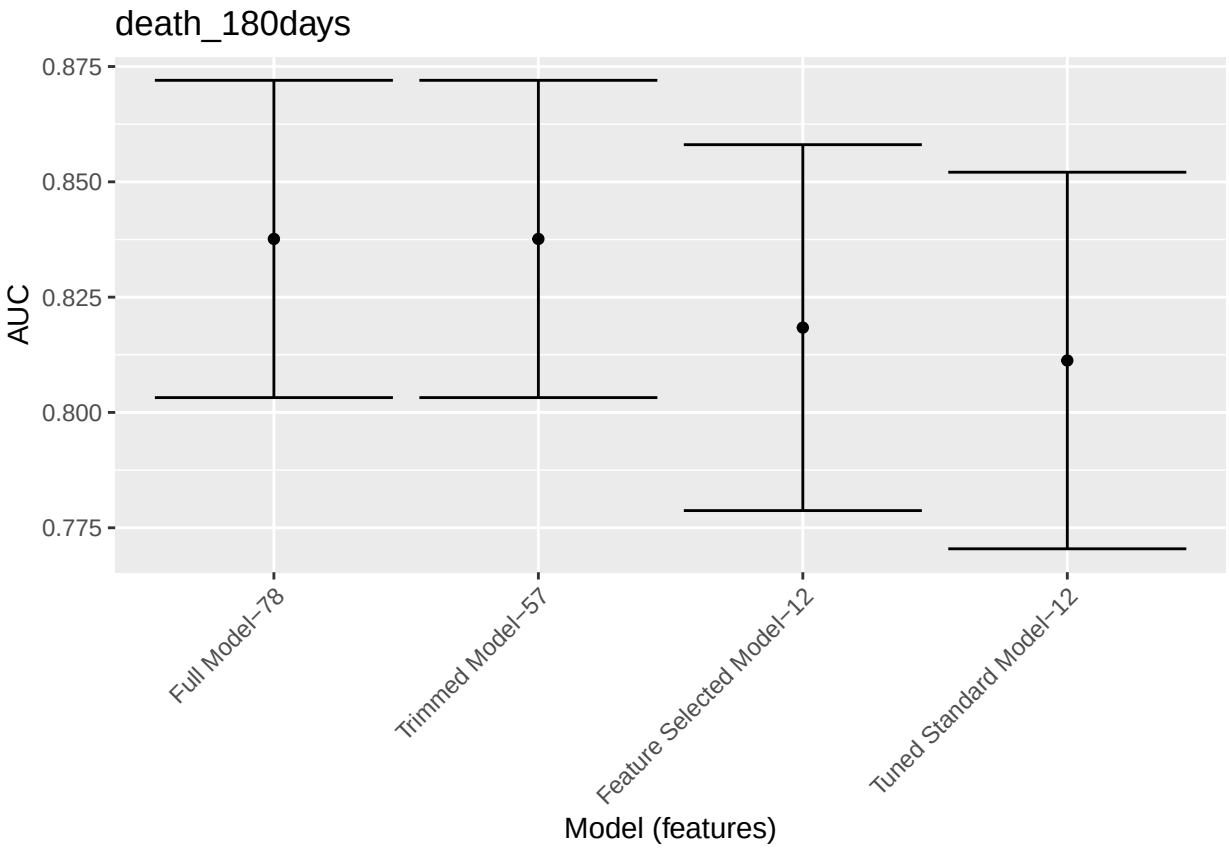

```
## $num_iterations
## [1] 210
##
## $learning_rate
```

```
## [1] 0.07831466
##
## $max_depth
## [1] 2
##
## $feature_fraction
## [1] 1
##
## $min_data_in_leaf
## [1] 37
##
## $min_gain_to_split
## [1] 5.953498e-10
##
## $bagging_fraction
## [1] 1
##
## $num_class
## [1] 1
##
## $objective
## [1] "binary"
##
## $num_threads
## $num_threads$num_threads
## [1] 0
##
##
## $seed
## [1] 17923
##
## $deterministic
## [1] TRUE
##
## $verbose
## [1] -1
##
## $metric
## list()
##
## $interaction_constraints
## list()
##
## $feature_pre_filter
## [1] FALSE
```

Models Comparison

```
df_auc <- tibble::tribble(
  ~Model, ~`AUC`, ~`Lower Limit`, ~`Upper Limit`, ~`Features`,
  'Full Model', full_model$auc, full_model$auc_lower, full_model$auc_upper, length(features),
  'Trimmed Model', trimmed_model$auc, trimmed_model$auc_lower, trimmed_model$auc_upper, length(trimmed_features)
  'Feature Selected Model', feature_selected_model$auc, feature_selected_model$auc_lower, feature_selected_model
  'Tuned Standard Model', standard_results$auc, standard_results$auc_lower, standard_results$auc_upper, length(s
) %>%
  mutate(Target = outcome_column,
         `Model (features)` = fct_reorder(paste0(Model, "-", Features), -Features))

df_auc %>%
  ggplot(aes(
```

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




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