Final Model - death_1year

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
Hmisc::list.tree(params)
```

```
##  params = list 5 (952 bytes)
## .  max_auc_loss = double 1= 0.01
## .  outcome_column = character 1= death_1year
## .  k = double 1= 10
## .  grid_size = double 1= 50
## .  repeats = double 1= 2
```

Minutes to run: 0.001

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
```

Minutes to run: 0.002

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))
```

Minutes to run: 0.021

```
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)
```

Minutes to run: 0.001

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)
```

Minutes to run: 0.004

```
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))

features = base::intersect(eligible_features, features_list)
```

Minutes to run: 0.001

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 Minutes to run: 0

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)
```

Minutes to run: 0.002

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]
    )
  )
}
```

Minutes to run: 0

```
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: 0.813"
```

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

```
## [1] "Full Model Test AUC: 0.821"
```

Minutes to run: 0.392

Features with zero importance on the initial model:

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

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

1. procedure_type_1
2. dialysis_t0
3. antiviral
4. vni
5. transplante_cardiaco
6. cir_toracica
7. transfusao
8. tilt_teste
9. histopatologia_qtde
10. angio_tc
11. angiografia
12. aortografia
13. pet_ct
14. bic Minutes to run: 0

```
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: 0.812"
```

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

```
## [1] "Trimmed Model Test AUC: 0.821"
```

Minutes to run: 0.362

```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) {
  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] "68"          "-6e-04"      "-0.0022"     "reop_type_1"
## [1] "67"         "-0.0011"    "-5e-04"     "endoscopia"
## [1] "66"       "-9e-04"   "2e-04"    "insulina"
## [1] "65"      "-0.0012" "-3e-04"  "holter"
## [1] "64"     "-9e-04" "3e-04"  "copd"
## [1] "63"         "4e-04"       "0.0012"      "flebografia"
## [1] "62"                    "0.0021"                 "0.0017"                  "cateter_venoso_central"
## [1] "61"           "0.0014"        "-7e-04"        "teste_esforco"
## [1] "60"         "0.0014"      "1e-04"       "ressonancia"
## [1] "59"     "0.0016" "1e-04"  "icp"
## [1] "58"                "0.0012"             "-4e-04"             "antiplaquetario_ev"
## [1] "57"       "2e-04"    "-0.001"   "digoxina"
## [1] "56"            "0.0018"         "0.0016"         "cardiac_arrest"
## [1] "55"          "0.0025"       "7e-04"        "hemodialysis"
## [1] "54"      "0.0014"  "-0.0011" "stroke"
## [1] "53"            "0.0021"        "7e-04"         "heart_failure"
## [1] "52"                       "0.0022"                  "1e-04"                    "outros_proced_cirurgico
## [1] "51"         "0.0019"      "-3e-04"      "antifungico"
## [1] "50"      "6e-04"   "-0.0013" "sex"
## [1] "50"         "6e-04"      "0.0055"     "tomografia"
## [1] "49"           "0.0014"        "8e-04"         "interconsulta"
## [1] "48"         "0.0018"     "4e-04"      "valvopathy"
## [1] "47"                "0.0016"             "-2e-04"             "procedure_type_new"
## [1] "46"      "0.0021"  "5e-04"   "cultura"
## [1] "45"       "0"        "-0.0021"  "prior_mi"
## [1] "44"     "1e-04"  "2e-04"  "cancer"
## [1] "43"                  "-0.0031"               "-0.0032"               "admission_pre_t0_180d"
## [1] "43"      "-0.0031" "0.0023"  "aco"
## [1] "43"                   "-0.0031"               "0.0049"                  "admission_t0_emergency"
## [1] "43"          "-0.0031"      "0.0044"       "hypertension"
## [1] "43"           "-0.0031"       "0.0041"        "heart_disease"
## [1] "43"       "-0.0031"  "0.0023"   "diabetes"
## [1] "42"          "-0.0027"     "4e-04"       "cateterismo"
## [1] "42"           "-0.0027"       "0.0034"        "cintilografia"
## [1] "41"        "-0.0013"   "0.0014"    "ultrassom"
## [1] "40"          "-2e-04"       "0.0011"       "cied_final_1"
## [1] "39"           "0.0015"        "0.0017"        "renal_failure"
## [1] "38"        "0.0034"    "0.0019"    "citologia"
## [1] "37"                      "9e-04"                    "-0.0025"                  "underlying_heart_diseas
## [1] "37"             "9e-04"          "0.0028"         "ecocardiograma"
## [1] "36"                  "0.0027"             "0.0018"             "ventilacao_mecanica"
## [1] "35"     "0.002"  "-7e-04" "dva"
## [1] "35"                 "0.002"             "0.0024"             "exames_imagem_qtde"
## [1] "34"     "0.0027" "7e-04"  "af"
## [1] "33"                 "0.0041"             "0.0013"             "cied_final_group_1"
## [1] "32"     "0.0059" "0.0018" "icu_t0"
## [1] "31"                   "0.0042"                 "-0.0016"                "analises_clinicas_qtde"
## [1] "30"                  "0.0035"                "-7e-04"                "proced_invasivos_qtde"
## [1] "29"            "0.0041"        "5e-04"         "antiarritmico"
```

```
## [1] "29"              "0.0041"          "0.0028"          "insuf_cardiaca"
## [1] "28"        "0.0032"    "-8e-04"    "estatina"
## [1] "28"                "0.0032"            "0.0042"            "equipe_multiprof"
## [1] "27"             "0.0014"         "-0.0018"        "psicofarmacos"
## [1] "26"                     "-5e-04"                 "-0.0019"                "metodos_graficos_qtde"
## [1] "25"         "0.0011"     "0.0016"     "diuretico"
## [1] "24"                    "-0.0026"               "-0.0037"               "meds_antimicrobianos"
## [1] "23"                "-0.0047"           "-0.0021"           "classe_meds_qtde"
## [1] "22"                    "-0.0051"               "-4e-04"                "meds_cardiovasc_qtde"
## [1] "22"          "-0.0051"     "0.0028"      "nyha_basal"
## [1] "22"               "-0.0051"          "0.0046"           "espironolactona"
## [1] "21"        "-0.0038"   "0.0013"    "ieca_bra"
## [1] "21"                      "-0.0038"                 "0.0151"                  "admission_pre_t0_count"
## [1] "21"                "-0.0038"          "0.015"            "education_level"
## [1] "21"             "-0.0038"        "0.0064"         "vasodilatador"
## [1] "21"                   "-0.0038"              "0.0131"               "comorbidities_count"
## [1] "20"           "-0.0057"      "-0.0019"      "laboratorio"
## [1] "20"           "-0.0057"      "0.0288"       "year_adm_t0"
## [1] "20"       "-0.0057" "0.0089"  "age"
## [1] "20"             "-0.0057"        "0.0161"         "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	83	0.8133	0.0067	0.0000	0.0000
All unimportant	TRUE	69	0.8117	0.0070	0.0016	0.0016
reop_type_1	TRUE	68	0.8139	0.0063	-0.0006	-0.0022
endoscopia	TRUE	67	0.8144	0.0061	-0.0011	-0.0005
insulina	TRUE	66	0.8141	0.0067	-0.0009	0.0002
holter	TRUE	65	0.8144	0.0064	-0.0012	-0.0003
copd	TRUE	64	0.8141	0.0063	-0.0009	0.0003
flebografia	TRUE	63	0.8129	0.0072	0.0004	0.0012
cateter_venoso_central	TRUE	62	0.8112	0.0070	0.0021	0.0017
teste_esforco	TRUE	61	0.8119	0.0067	0.0014	-0.0007
ressonancia	TRUE	60	0.8118	0.0062	0.0014	0.0001
icp	TRUE	59	0.8117	0.0064	0.0016	0.0001
antiplaquetario_ev	TRUE	58	0.8121	0.0068	0.0012	-0.0004
digoxina	TRUE	57	0.8131	0.0064	0.0002	-0.0010
cardiac_arrest	TRUE	56	0.8115	0.0058	0.0018	0.0016
hemodialysis	TRUE	55	0.8107	0.0065	0.0025	0.0007
stroke	TRUE	54	0.8119	0.0062	0.0014	-0.0011
heart_failure	TRUE	53	0.8112	0.0063	0.0021	0.0007
outros_proced_cirurgicos	TRUE	52	0.8111	0.0070	0.0022	0.0001
antifungico	TRUE	51	0.8114	0.0065	0.0019	-0.0003
sex	TRUE	50	0.8127	0.0071	0.0006	-0.0013
tomografia	FALSE	49	0.8071	0.0065	0.0006	0.0055
interconsulta	TRUE	49	0.8119	0.0070	0.0014	0.0008
valvopathy	TRUE	48	0.8115	0.0062	0.0018	0.0004
procedure_type_new	TRUE	47	0.8117	0.0073	0.0016	-0.0002
cultura	TRUE	46	0.8112	0.0065	0.0021	0.0005
prior_mi	TRUE	45	0.8133	0.0077	0.0000	-0.0021
cancer	TRUE	44	0.8131	0.0064	0.0001	0.0002
admission_pre_t0_180d	TRUE	43	0.8163	0.0066	-0.0031	-0.0032
aco	FALSE	42	0.8140	0.0068	-0.0031	0.0023

Table 1: *(continued)*

Tested Feature	Dropped	Features	CV AUC	CV AUC Std Error	Total AUC Loss	Instant AUC Loss
admission_t0_emergency	FALSE	42	0.8114	0.0063	-0.0031	0.0049
hypertension	FALSE	42	0.8119	0.0063	-0.0031	0.0044
heart_disease	FALSE	42	0.8123	0.0071	-0.0031	0.0041
diabetes	FALSE	42	0.8140	0.0066	-0.0031	0.0023
cateterismo	TRUE	42	0.8160	0.0068	-0.0027	0.0004
cintilografia	FALSE	41	0.8125	0.0068	-0.0027	0.0034
ultrassom	TRUE	41	0.8146	0.0062	-0.0013	0.0014
cied_final_1	TRUE	40	0.8134	0.0064	-0.0002	0.0011
renal_failure	TRUE	39	0.8118	0.0059	0.0015	0.0017
citologia	TRUE	38	0.8098	0.0061	0.0034	0.0019
underlying_heart_disease	TRUE	37	0.8123	0.0058	0.0009	-0.0025
ecocardiograma	FALSE	36	0.8095	0.0063	0.0009	0.0028
ventilacao_mecanica	TRUE	36	0.8105	0.0061	0.0027	0.0018
dva	TRUE	35	0.8112	0.0058	0.0020	-0.0007
exames_imagem_qtde	FALSE	34	0.8088	0.0058	0.0020	0.0024
af	TRUE	34	0.8105	0.0059	0.0027	0.0007
cied_final_group_1	TRUE	33	0.8092	0.0066	0.0041	0.0013
icu_t0	TRUE	32	0.8074	0.0070	0.0059	0.0018
analises_clinicas_qtde	TRUE	31	0.8090	0.0058	0.0042	-0.0016
proced_invasivos_qtde	TRUE	30	0.8097	0.0058	0.0035	-0.0007
antiarritmico	TRUE	29	0.8092	0.0065	0.0041	0.0005
insuf_cardiaca	FALSE	28	0.8064	0.0070	0.0041	0.0028
estatina	TRUE	28	0.8100	0.0065	0.0032	-0.0008
equipe_multiprof	FALSE	27	0.8058	0.0066	0.0032	0.0042
psicofarmacos	TRUE	27	0.8118	0.0065	0.0014	-0.0018
metodos_graficos_qtde	TRUE	26	0.8138	0.0072	-0.0005	-0.0019
diuretico	TRUE	25	0.8122	0.0067	0.0011	0.0016
meds_antimicrobianos	TRUE	24	0.8158	0.0071	-0.0026	-0.0037
classe_meds_qtde	TRUE	23	0.8179	0.0062	-0.0047	-0.0021
meds_cardiovasc_qtde	TRUE	22	0.8183	0.0063	-0.0051	-0.0004
nyha_basal	FALSE	21	0.8156	0.0059	-0.0051	0.0028
espironolactona	FALSE	21	0.8138	0.0057	-0.0051	0.0046
ieca_bra	TRUE	21	0.8171	0.0067	-0.0038	0.0013
admission_pre_t0_count	FALSE	20	0.8020	0.0072	-0.0038	0.0151
education_level	FALSE	20	0.8021	0.0066	-0.0038	0.0150
vasodilatador	FALSE	20	0.8107	0.0072	-0.0038	0.0064
comorbidities_count	FALSE	20	0.8040	0.0072	-0.0038	0.0131
laboratorio	TRUE	20	0.8190	0.0076	-0.0057	-0.0019
year_adm_t0	FALSE	19	0.7902	0.0080	-0.0057	0.0288
age	FALSE	19	0.8101	0.0078	-0.0057	0.0089
hospital_stay	FALSE	19	0.8028	0.0070	-0.0057	0.0161

Minutes to run: 20.015

```
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: 0.819"
```

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

```
## [1] "Selected Model Test AUC: 0.803"
```

```
 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")
```

Minutes to run: 0.247

Hyperparameter tuning

Selected features:

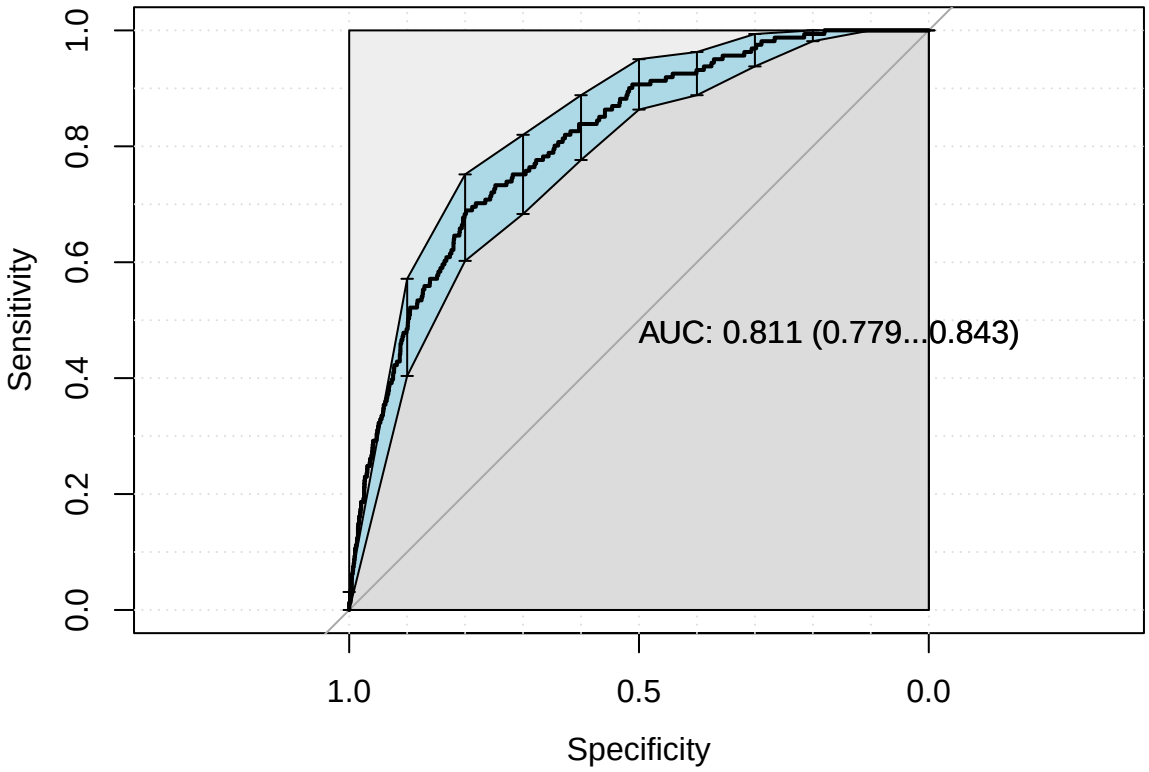
```
gluedown::md_order(selected_features, seq = TRUE, pad = TRUE)
```

1. hospital_stay
2. age
3. year_adm_t0
4. comorbidities_count
5. espironolactona
6. education_level
7. admission_pre_t0_count
8. nyha_basal
9. vasodilatador
10. equipe_multiprof
11. insuf_cardiaca
12. exames_imagem_qtde
13. ecocardiograma
14. hypertension
15. heart_disease
16. admission_t0_emergency
17. cintilografia
18. tomografia
19. diabetes
20. aco Minutes to run: 0

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(),
    sample_size = 1.0
  ) %>%
    set_engine("lightgbm",
               nthread = 8) %>%
    set_mode("classification")

  lightgbm_grid <- grid_latin_hypercube(
    trees(range = c(25L, 150L)),
    min_n(range = c(2L, 100L)),
    tree_depth(range = c(5L, 15L)),
    learn_rate(range = c(-3, -1), trans = log10_trans()),
    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) %>%
    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)
```




```
##   |
```

```
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
  )
)

# Save the final model. We need it for the calculator
lgb.save(
  parsnip::extract_fit_engine(final_lightgbm_fit),
  sprintf("./results/%s/final_model.txt", outcome_column)
)
saveRDS(final_lightgbm_fit,
        sprintf("./results/%s/final_model_wf.rds", outcome_column))
```

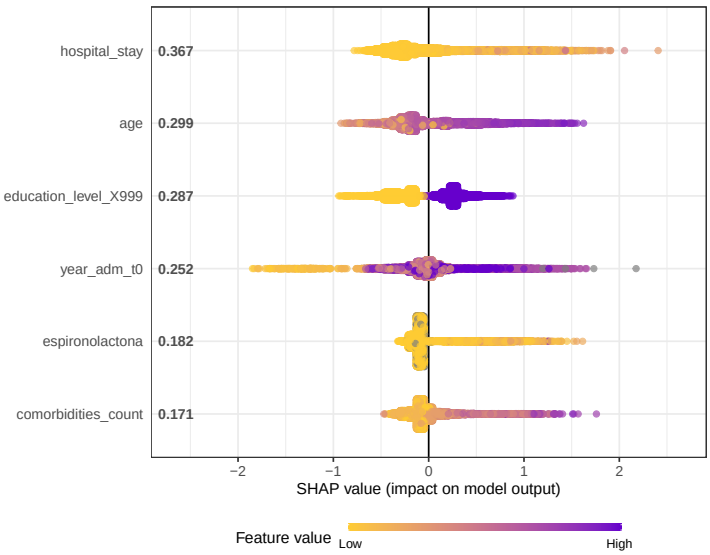
Minutes to run: 8

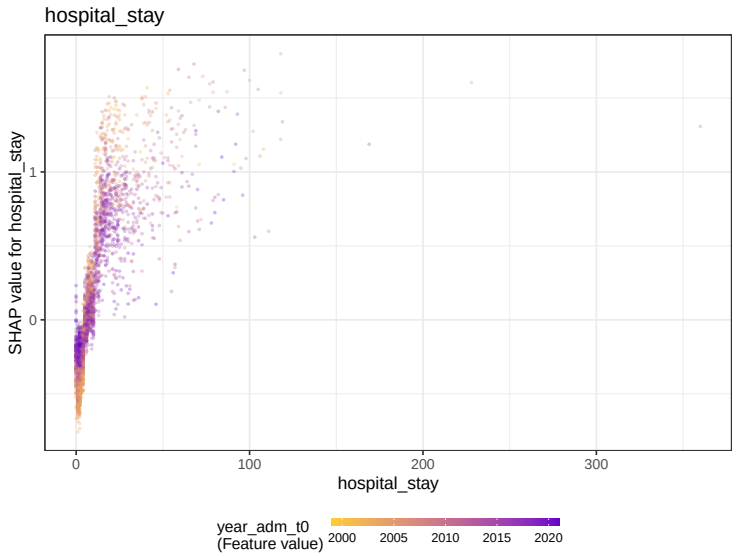
SHAP values

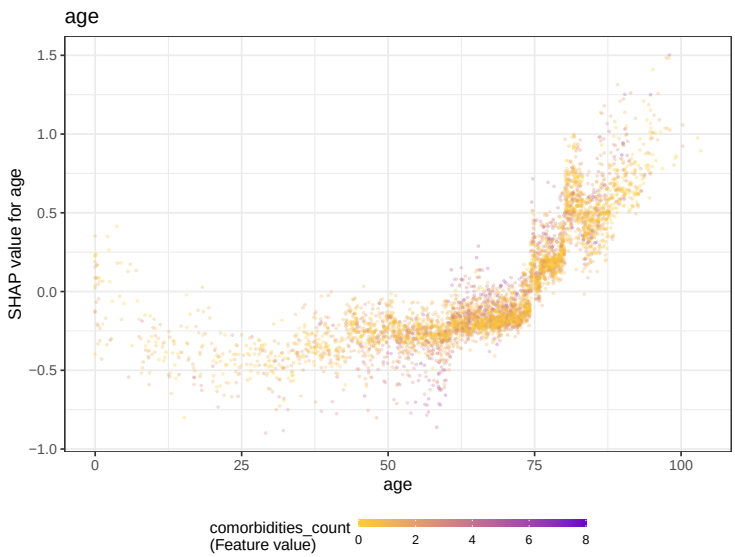
```
lightgbm_model <- parsnip::extract_fit_engine(final_lightgbm_fit)

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(6, length(selected_features))
plotted <- 0

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)) {
  p <- shap.plot.dependence(
    shap,
    x = x,
    color_feature = "auto",
    smooth = FALSE,
    jitter_width = 0.01,
    alpha = 0.3
  ) +
    labs(title = x)

  if (plotted < n_plots) {
    print(p)
    plotted <- plotted + 1
  }

  ggsave(sprintf("./auxiliar/final_model/shap_plots/%s.png", x),
         plot = p,
         dpi = 300)
}
```

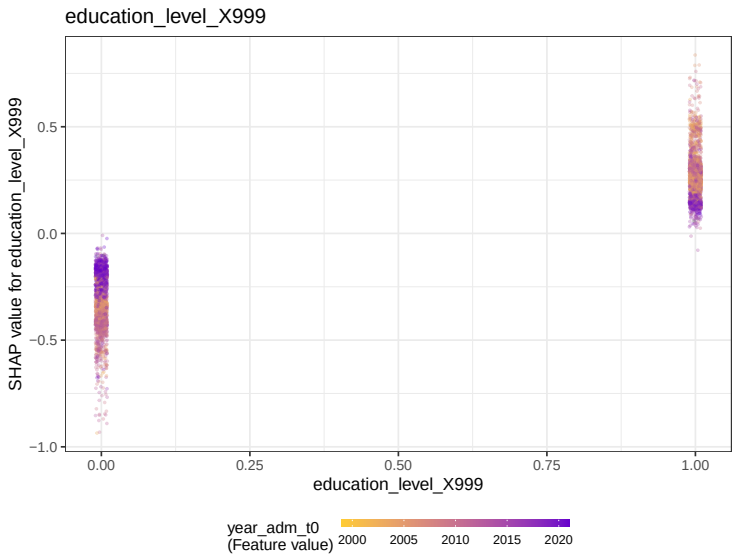
```
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image

## Warning: Removed 10 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image

## Warning: Removed 10 rows containing missing values (geom_point).

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

## Saving 6.5 x 5 in image

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

## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image

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

## Saving 6.5 x 5 in image

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

## Saving 6.5 x 5 in image

## Warning: Removed 826 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image

## Warning: Removed 1750 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image

## Warning: Removed 826 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image

## Warning: Removed 826 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image

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

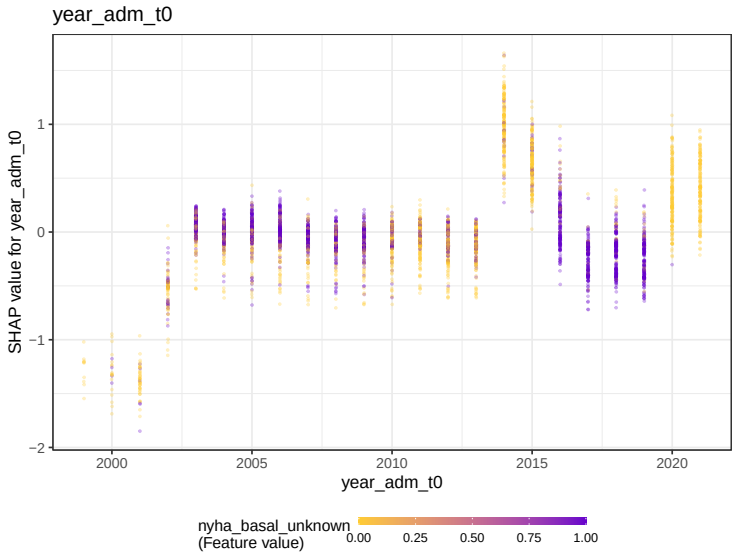
```
## Saving 6.5 x 5 in image

## Warning: Removed 826 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image

## Warning: Removed 826 rows containing missing values (geom_point).

## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
## Saving 6.5 x 5 in image
```

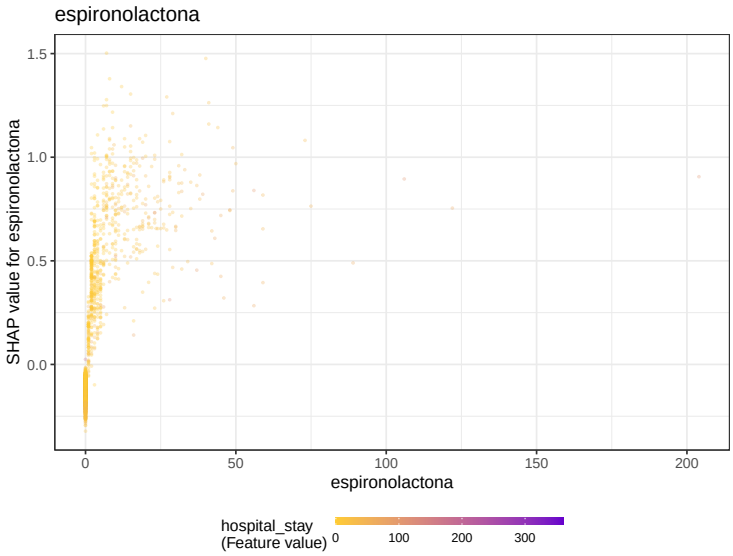


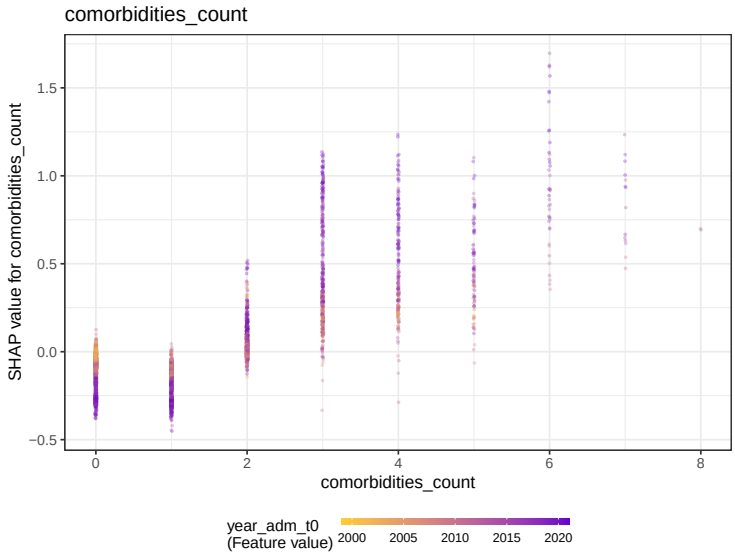












Minutes to run: 0.334

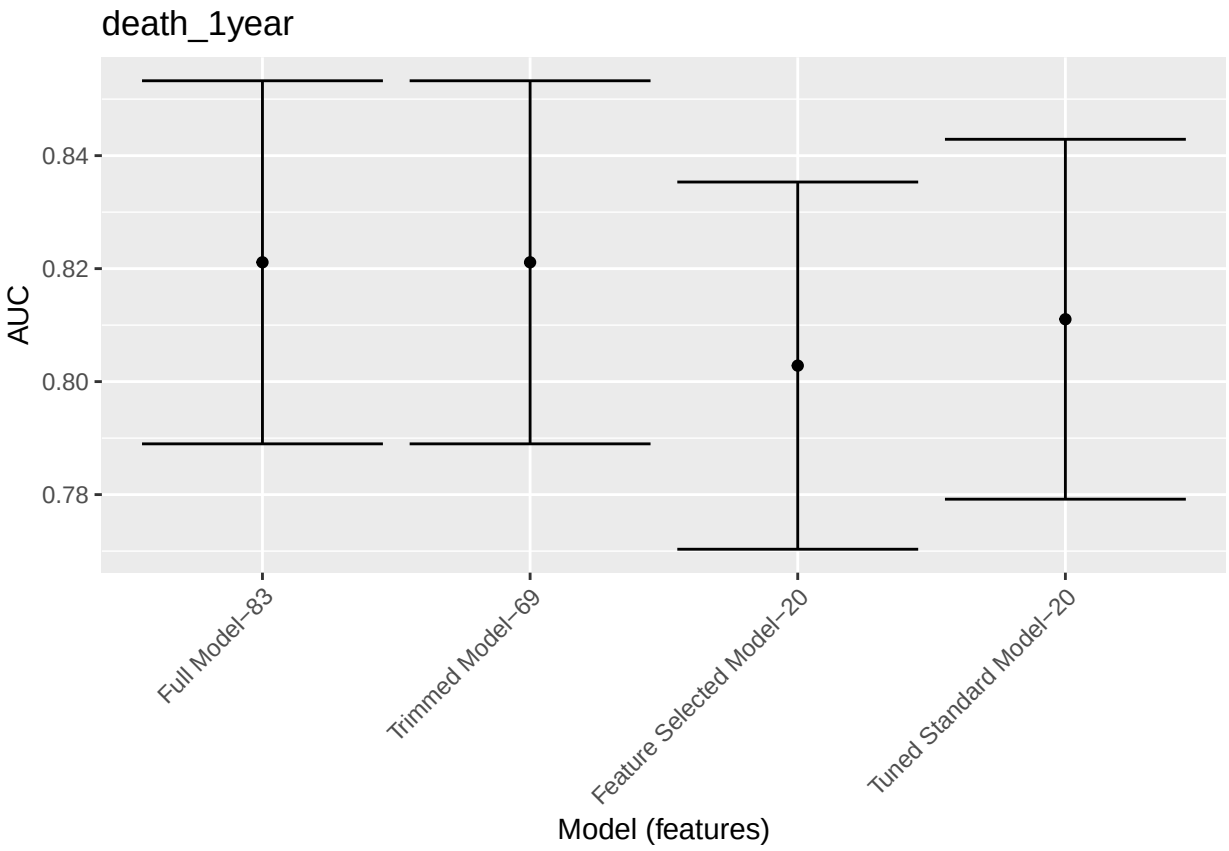
```
## $num_iterations
## [1] 101
##
## $learning_rate
## [1] 0.03979323
##
## $max_depth
## [1] 6
##
## $feature_fraction_bynode
## [1] 1
##
## $min_data_in_leaf
## [1] 25
##
## $min_gain_to_split
## [1] 0
##
## $bagging_fraction
## [1] 1
##
## $num_class
## [1] 1
##
```

```
## $objective
## [1] "binary"
##
## $num_threads
## $num_threads$num_threads
## [1] 0
##
##
## $nthread
## [1] 8
##
## $seed
## [1] 59739
##
## $deterministic
## [1] TRUE
##
## $verbose
## [1] -1
##
## $metric
## list()
##
## $interaction_constraints
## list()
##
## $feature_pre_filter
## [1] FALSE
```

Minutes to run: 0

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))
```

Minutes to run: 0.003