

# Final Model - death\_30days

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_30days
## . k = double 1= 10
## . grid_size = double 1= 50
## . repeats = double 1= 2
```

## 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
predict <- stats::predict
```

## 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))

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. valvopathy  
12. diabetes  
13. renal\_failure  
14. hemodialysis  
15. cancer  
16. comorbidities\_count  
17. procedure\_type\_1  
18. reop\_type\_1  
19. procedure\_type\_new  
20. cied\_final\_1  
21. cied\_final\_group\_1  
22. admission\_pre\_t0\_count  
23. admission\_pre\_t0\_180d  
24. year\_adm\_t0  
25. icu\_t0  
26. antiarritmico  
27. antihipertensivo  
28. betabloqueador  
29. dva  
30. diuretico  
31. vasodilatador  
32. espironolactona  
33. antiplaquetario\_ev  
34. insulina  
35. psicofarmacos  
36. antifungico  
37. classe\_meds\_qtde  
38. meds\_cardiovasc\_qtde  
39. meds\_antimicrobianos  
40. vni  
41. ventilacao\_mecanica  
42. intervencao\_cv  
43. cateter\_venoso\_central  
44. proced\_invasivos\_qtde  
45. transfusao  
46. interconsulta  
47. equipe\_multiprof  
48. holter  
49. metodos\_graficos\_qtde  
50. laboratorio  
51. cultura  
52. analises\_clinicas\_qtde  
53. citologia  
54. histopatologia\_qtde  
55. angio\_tc  
56. angiografia  
57. cintilografia  
58. ecocardiograma  
59. flebografia  
60. ultrassom  
61. tomografia  
62. ressonancia  
63. exames\_imagem\_qtde  
64. bic  
65. 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: 0.735"
sprintf('Full Model Test AUC: %.3f' ,full_model$auc)

## [1] "Full Model Test AUC: 0.760"

Features with zero importance on the initial model:

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

unimportant_features %>%
  gluedown::md_order()

1. prior_mi
2. valvopathy
3. hemodialysis
4. procedure_type_new
5. antiplaquetario_ev
6. antifungico
7. intervencao_cv
8. transfusao
9. cintilografia

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: 0.724"
sprintf('Trimmed Model Test AUC: %.3f' ,trimmed_model$auc)

## [1] "Trimmed Model Test AUC: 0.760"

```

```

selection_results <- tibble::tribble(
  ~`Tested Feature`, ~`Dropped`, ~`Number of Features`, ~`CV AUC`, ~`CV AUC Std Error`, ~`Total AUC Loss`, ~`Instant AUC Loss`,
  '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,
      `Instant AUC Loss` = instant_auc_loss)
}

```

```

`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] "64"      "0"       "prior_mi"
## [1] "63"      "0.0011"   "0.001"   "valvopathy"
## [1] "62"      "0.0011"   "0"        "hemodialysis"
## [1] "61"      "-3e-04"   "-0.0014"  "procedure_type_new"
## [1] "60"      "-3e-04"   "0"        "antiplaquetario_ev"
## [1] "59"      "-3e-04"   "0"        "antifungico"
## [1] "58"      "-3e-04"   "0"        "intervencao_cv"
## [1] "57"      "-3e-04"   "0"        "transfusao"
## [1] "56"      "-5e-04"   "-3e-04"   "cintilografia"
## [1] "55"      "-0.0039"  "-0.0034"  "histopatologia_qtde"
## [1] "54"      "-0.0083"  "-0.0044"  "ultrassom"
## [1] "53"      "-0.0083"  "0"        "cateter Venoso Central"
## [1] "52"      "-0.0082"  "0"        "cancer"
## [1] "52"      "-0.0082"  "0.0215"   "admission_pre_t0_180d"
## [1] "51"      "-0.0064"  "0.0019"   "angio_tc"
## [1] "50"      "-0.0128"  "-0.0065"  "angiografia"
## [1] "49"      "-0.0128"  "0"        "vni"
## [1] "49"      "-0.0128"  "0.0021"   "ventilacao_mecanica"
## [1] "48"      "-0.0135"  "-7e-04"   "heart_failure"
## [1] "48"      "-0.0135"  "0.0076"   "cied_final_group_1"
## [1] "47"      "-0.0127"  "8e-04"   "reop_type_1"
## [1] "46"      "-0.0142"  "-0.0015"  "sex"
## [1] "45"      "-0.0123"  "0.0019"   "flebografia"
## [1] "45"      "-0.0123"  "0.0024"   "citologia"
## [1] "44"      "-0.014"   "-0.0018"  "heart_disease"
## [1] "44"      "-0.014"   "0.0052"   "insulina"
## [1] "44"      "-0.014"   "0.0056"   "diabetes"
## [1] "43"      "-0.0212"  "-0.0072"  "ecocardiograma"
## [1] "43"      "-0.0212"  "0.0107"   "tomografia"
## [1] "43"      "-0.0212"  "0.0084"   "af"
## [1] "43"      "-0.0212"  "0.0065"   "dva"
## [1] "43"      "-0.0212"  "0.0077"   "antihipertensivo"
## [1] "43"      "-0.0212"  "0.0097"   "proced_invasivos_qtde"
## [1] "43"      "-0.0212"  "0.0102"   "betabloqueador"
## [1] "43"      "-0.0212"  "0.0046"   "procedure_type_1"
## [1] "43"      "-0.0212"  "0.0065"   "cultura"
## [1] "43"      "-0.0212"  "0.0023"   "diuretico"
## [1] "43"      "-0.0212"  "0.0068"   "interconsulta"
## [1] "43"      "-0.0212"  "0.0032"   "hypertension"
## [1] "43"      "-0.0212"  "0.0263"   "ressonancia"
## [1] "43"      "-0.0212"  "0.0504"   "renal_failure"
## [1] "43"      "-0.0212"  "0.009"    "cied_final_1"
## [1] "43"      "-0.0212"  "0.0202"   "classe_meds_qtde"
## [1] "43"      "-0.0212"  "0.0036"   "comorbidities_count"
## [1] "43"      "-0.0212"  "0.0282"   "nyha Basal"
## [1] "43"      "-0.0212"  "0.0142"   "holter"
## [1] "43"      "-0.0212"  "0.0095"   "analises_clinicas_qtde"
## [1] "43"      "-0.0212"  "0.0074"   "antiarritmico"
## [1] "43"      "-0.0212"  "0.0032"   "equipe_multiprof"

```

```

## [1] "42"           "-0.0204"          "8e-04"           "underlying_heart_diseas
## [1] "41"           "-0.0242" "bic"      "0.0017"          "exames_imagem_qtde"
## [1] "40"           "-0.0225"          "0.0017"          "0.0294"          "meds_cardiovasc_qtde"
## [1] "40"           "-0.0225"          "0.005"           "laboratorio"
## [1] "39"           "-0.0375"          "-0.0149"          "espironolactona"
## [1] "39"           "-0.0375"          "0.048"           "vasodilatador"
## [1] "39"           "-0.0375"          "0.0413"          "education_level"
## [1] "39"           "-0.0375"          "0.0174"          "meds_antimicrobianos"
## [1] "39"           "-0.0375"          "0.0352"          "year_adm_t0"
## [1] "39"           "-0.0375"          "0.0345"          "metodos_graficos_qtde"
## [1] "39"           "-0.0375"          "0.0095"          "icu_t0"
## [1] "39"           "-0.0375"          "0.044"           "admission_pre_t0_count"
## [1] "39"           "-0.0375"          "0.021"           "psicofarmacos"
## [1] "39"           "-0.0375"          "0.0478"          "hospital_stay"
## [1] "39"           "-0.0375"          "0.0166"          "age"

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	65	0.7346	0.0285	0.0000	0.0000
prior_mi	TRUE	64	0.7345	0.0285	0.0000	0.0000
valvopathy	TRUE	63	0.7335	0.0283	0.0011	0.0010
hemodialysis	TRUE	62	0.7335	0.0283	0.0011	0.0000
procedure_type_new	TRUE	61	0.7349	0.0280	-0.0003	-0.0014
antiplaquetario_ev	TRUE	60	0.7349	0.0280	-0.0003	0.0000
antifungico	TRUE	59	0.7349	0.0281	-0.0003	0.0000
intervencao_cv	TRUE	58	0.7349	0.0281	-0.0003	0.0000
transfusao	TRUE	57	0.7349	0.0281	-0.0003	0.0000
cintilografia	TRUE	56	0.7351	0.0281	-0.0005	-0.0003
histopatologia_qtde	TRUE	55	0.7385	0.0261	-0.0039	-0.0034
ultrassom	TRUE	54	0.7429	0.0260	-0.0083	-0.0044
cateter Venoso_Central	TRUE	53	0.7429	0.0260	-0.0083	0.0000
cancer	TRUE	52	0.7428	0.0260	-0.0082	0.0000
admission_pre_t0_180d	FALSE	51	0.7213	0.0402	-0.0082	0.0215
angio_tc	TRUE	51	0.7410	0.0261	-0.0064	0.0019
angiografia	TRUE	50	0.7474	0.0239	-0.0128	-0.0065
vni	TRUE	49	0.7474	0.0239	-0.0128	0.0000
ventilacao_mecanica	FALSE	48	0.7453	0.0237	-0.0128	0.0021
heart_failure	TRUE	48	0.7481	0.0235	-0.0135	-0.0007
cied_final_group_1	FALSE	47	0.7405	0.0252	-0.0135	0.0076
reop_type_1	TRUE	47	0.7473	0.0227	-0.0127	0.0008
sex	TRUE	46	0.7488	0.0230	-0.0142	-0.0015
flebografia	TRUE	45	0.7469	0.0235	-0.0123	0.0019
citologia	FALSE	44	0.7445	0.0234	-0.0123	0.0024
heart_disease	TRUE	44	0.7486	0.0232	-0.0140	-0.0018
insulina	FALSE	43	0.7434	0.0235	-0.0140	0.0052
diabetes	FALSE	43	0.7431	0.0220	-0.0140	0.0056
ecocardiograma	TRUE	43	0.7558	0.0222	-0.0212	-0.0072
tomografia	FALSE	42	0.7452	0.0254	-0.0212	0.0107
af	FALSE	42	0.7474	0.0230	-0.0212	0.0084
dva	FALSE	42	0.7493	0.0230	-0.0212	0.0065
antihipertensivo	FALSE	42	0.7482	0.0248	-0.0212	0.0077
proced_invasivos_qtde	FALSE	42	0.7461	0.0232	-0.0212	0.0097
betabloqueador	FALSE	42	0.7456	0.0223	-0.0212	0.0102

Table 1: (continued)

Tested Feature	Dropped	Features	CV AUC	CV AUC Std Error	Total AUC Loss	Instant AUC Loss
procedure_type_1	FALSE	42	0.7513	0.0217	-0.0212	0.0046
cultura	FALSE	42	0.7494	0.0234	-0.0212	0.0065
diuretico	FALSE	42	0.7536	0.0230	-0.0212	0.0023
interconsulta	FALSE	42	0.7490	0.0246	-0.0212	0.0068
hypertension	FALSE	42	0.7526	0.0239	-0.0212	0.0032
ressonancia	FALSE	42	0.7295	0.0375	-0.0212	0.0263
renal_failure	FALSE	42	0.7055	0.0399	-0.0212	0.0504
cied_final_1	FALSE	42	0.7468	0.0252	-0.0212	0.0090
classe_meds_qtde	FALSE	42	0.7356	0.0261	-0.0212	0.0202
comorbidities_count	FALSE	42	0.7522	0.0239	-0.0212	0.0036
nyha_basal	FALSE	42	0.7277	0.0392	-0.0212	0.0282
holter	FALSE	42	0.7416	0.0242	-0.0212	0.0142
analises_clinicas_qtde	FALSE	42	0.7463	0.0226	-0.0212	0.0095
antiarritmico	FALSE	42	0.7485	0.0234	-0.0212	0.0074
equipe_multiprof	FALSE	42	0.7526	0.0230	-0.0212	0.0032
underlying_heart_disease	TRUE	42	0.7550	0.0244	-0.0204	0.0008
bic	TRUE	41	0.7588	0.0221	-0.0242	-0.0038
exames_imagem_qtde	TRUE	40	0.7571	0.0246	-0.0225	0.0017
meds_cardiovasc_qtde	FALSE	39	0.7277	0.0274	-0.0225	0.0294
laboratorio	FALSE	39	0.7521	0.0286	-0.0225	0.0050
espironolactona	TRUE	39	0.7720	0.0232	-0.0375	-0.0149
vasodilatador	FALSE	38	0.7240	0.0274	-0.0375	0.0480
education_level	FALSE	38	0.7308	0.0298	-0.0375	0.0413
meds_antimicrobianos	FALSE	38	0.7546	0.0251	-0.0375	0.0174
year_adm_t0	FALSE	38	0.7368	0.0232	-0.0375	0.0352
metodos_graficos_qtde	FALSE	38	0.7375	0.0325	-0.0375	0.0345
icu_t0	FALSE	38	0.7625	0.0243	-0.0375	0.0095
admission_pre_t0_count	FALSE	38	0.7281	0.0293	-0.0375	0.0440
psicofarmacos	FALSE	38	0.7510	0.0268	-0.0375	0.0210
hospital_stay	FALSE	38	0.7242	0.0323	-0.0375	0.0478
age	FALSE	38	0.7554	0.0328	-0.0375	0.0166

```

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: 0.772"
sprintf('Selected Model Test AUC: %.3f', feature_selected_model$auc)

## [1] "Selected Model Test AUC: 0.776"

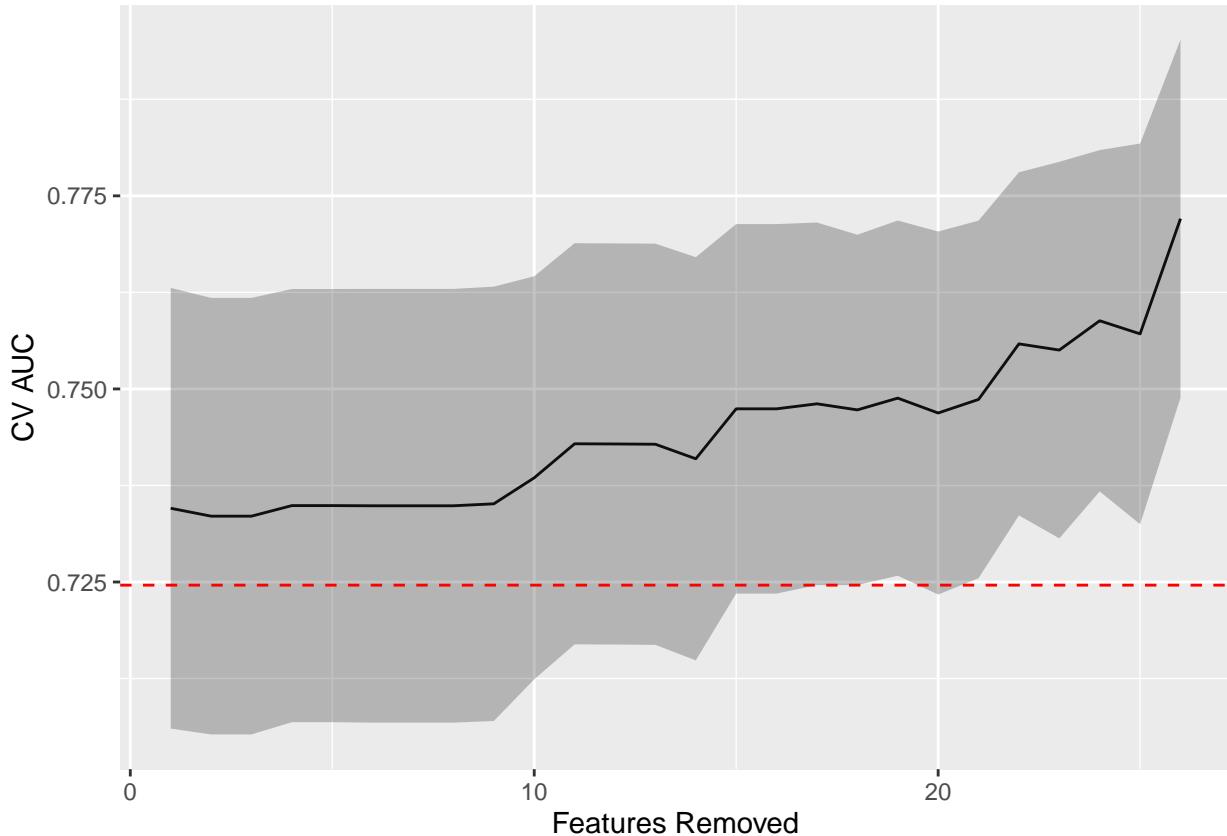
selection_results <- selection_results %>%
  filter(`Number of Features` < length(features)) %>%
  mutate(`Features Removed` = length(features) - `Number of Features`,
         `CV AUC Low` = `CV AUC` - `CV AUC Std Error`,
         `CV AUC High` = `CV AUC` + `CV AUC Std Error`)

```

```

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. education\_level
3. nyha\_basal
4. hypertension
5. af
6. diabetes
7. renal\_failure
8. comorbidities\_count
9. procedure\_type\_1
10. cied\_final\_1
11. cied\_final\_group\_1
12. admission\_pre\_t0\_count
13. admission\_pre\_t0\_180d
14. year\_adm\_t0
15. icu\_t0
16. antiaritmico
17. antihipertensivo
18. betabloqueador

19. dva  
 20. diuretico  
 21. vasodilatador  
 22. insulina  
 23. psicofarmacos  
 24. classe\_meds\_qtd  
 25. meds\_cardiovasc\_qtd  
 26. meds\_antimicrobianos  
 27. ventilacao\_mecanica  
 28. proced\_invasivos\_qtd  
 29. interconsulta  
 30. equipe\_multiprof  
 31. holter  
 32. metodos\_graficos\_qtd  
 33. laboratorio  
 34. cultura  
 35. analises\_clinicas\_qtd  
 36. citologia  
 37. tomografia  
 38. ressonancia  
 39. hospital\_stay

## 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(2L, 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,
              tuning_length = 5)
  )
}

```

```

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)

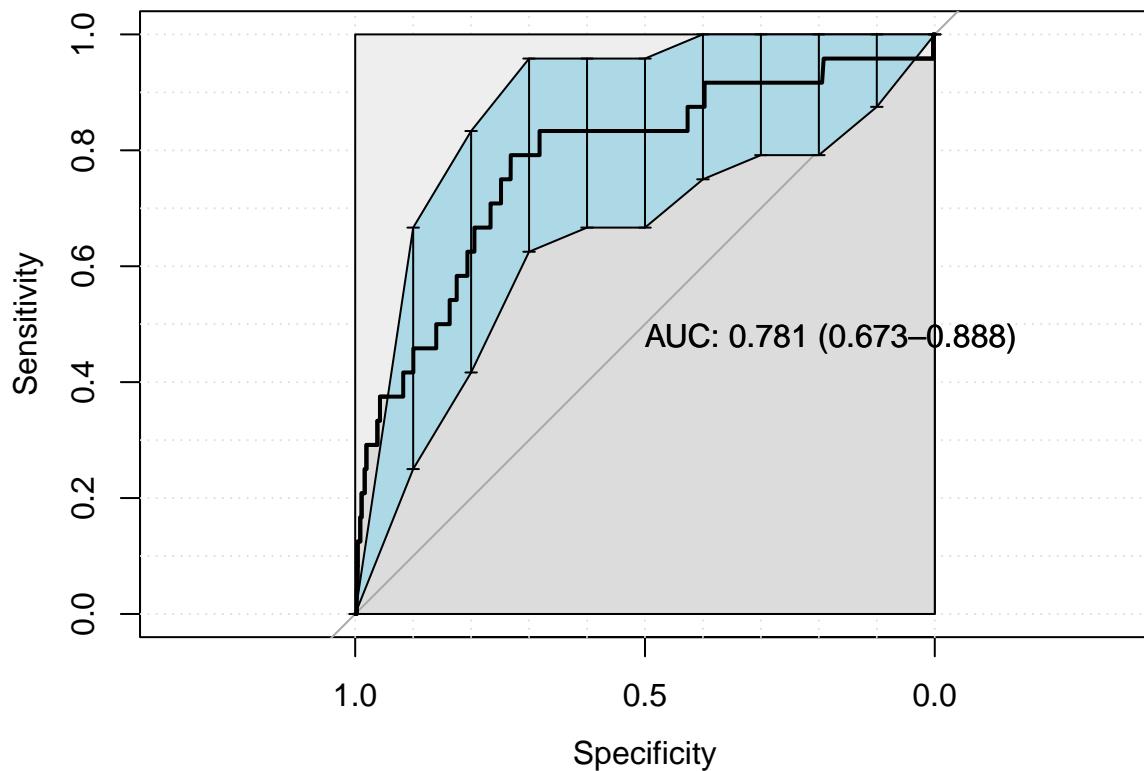
lightgbm_auc <- validation(final_lightgbm_fit, df_test)

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)

```



```

## [1] "Optimal Threshold: 0.00"
## Confusion Matrix and Statistics
##
##      reference
## data      0     1
##   0 3443     5
##   1 1263    19
##
##                  Accuracy : 0.7319
##                     95% CI : (0.7191, 0.7445)
##      No Information Rate : 0.9949
##      P-Value [Acc > NIR] : 1
##
##                  Kappa : 0.0193
##
## Mcnemar's Test P-Value : <2e-16
##
##      Sensitivity : 0.73162
##      Specificity : 0.79167
##      Pos Pred Value : 0.99855
##      Neg Pred Value : 0.01482
##      Prevalence : 0.99493
##      Detection Rate : 0.72791
##      Detection Prevalence : 0.72896
##      Balanced Accuracy : 0.76164
##
##      '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
)
)

# 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))

```

## 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)

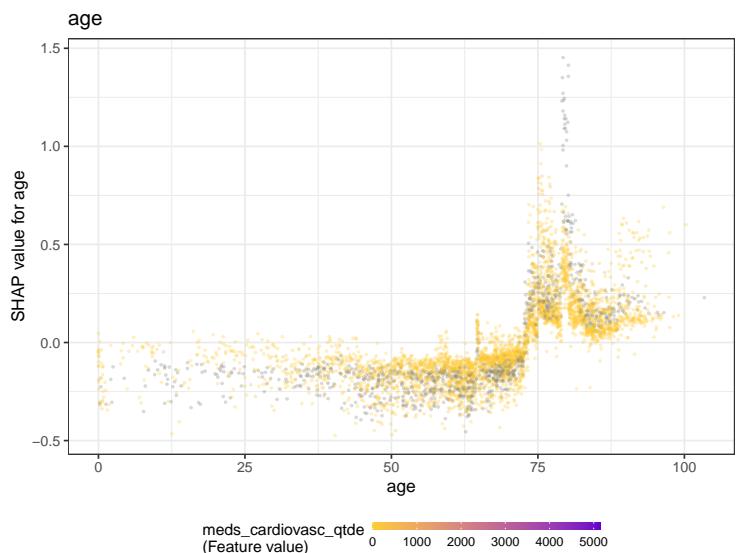
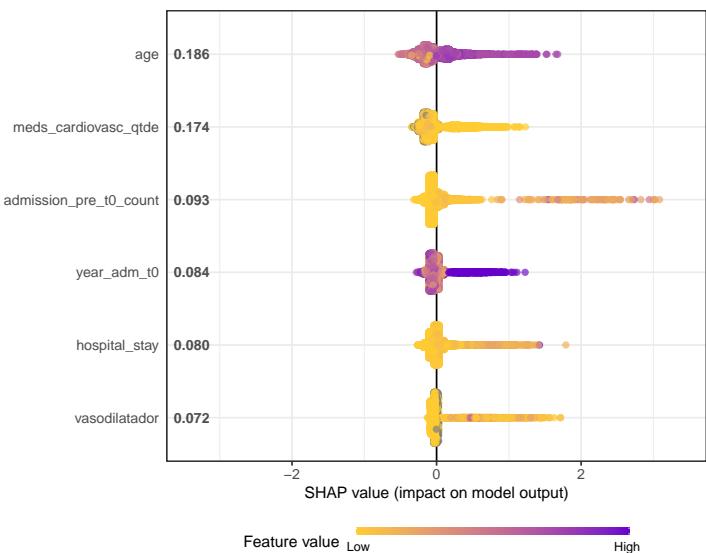
  if (plotted < n_plots) {
    print(p)
    plotted <- plotted + 1
  }
}

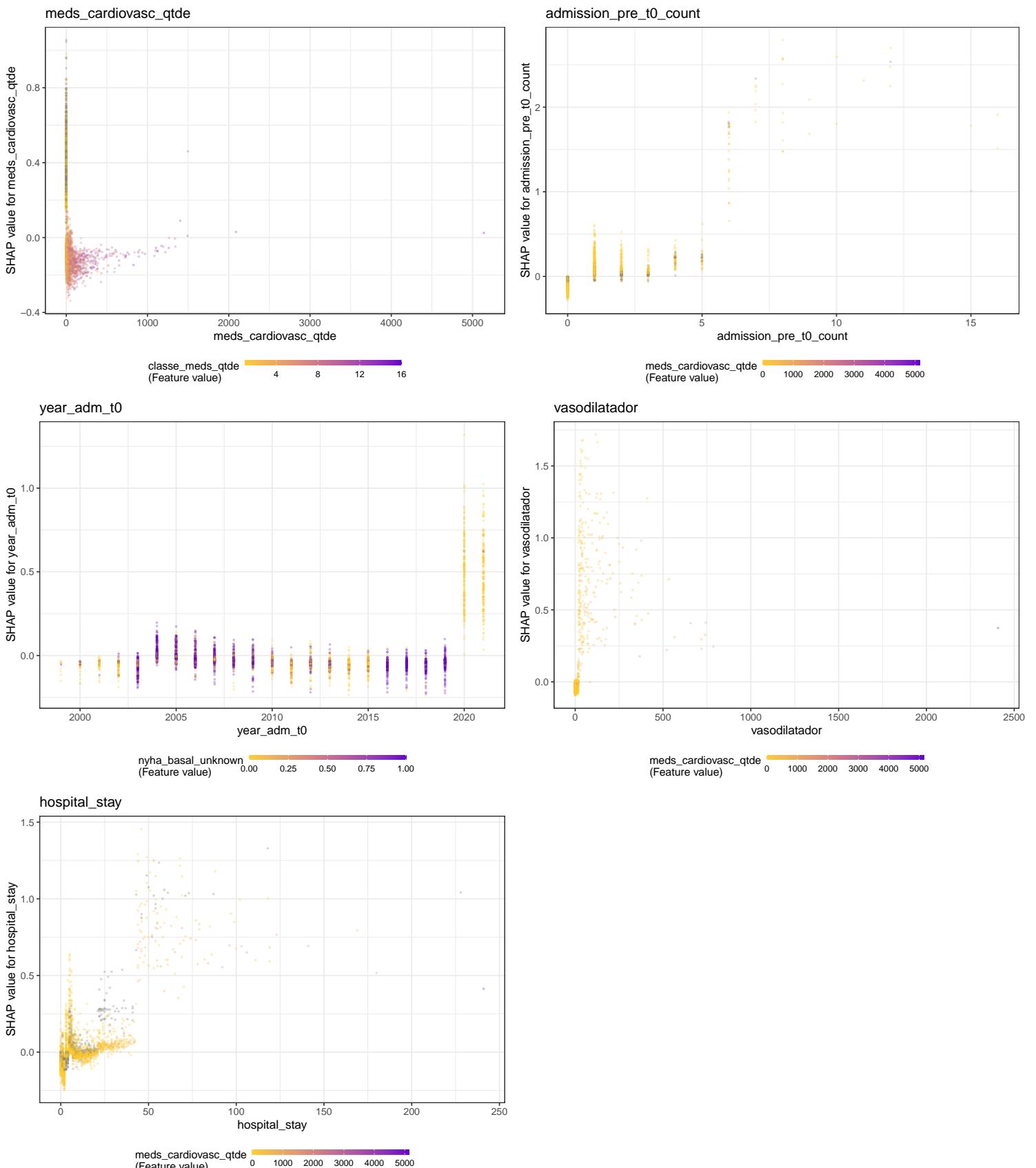
ggsave(sprintf("./auxiliar/final_model/shap_plots/%s.png", x),
       plot = p,
       dpi = 300)
}

## Saving 6.5 x 5 in image
## Warning: Removed 1055 rows containing missing values (geom_point).
## Saving 6.5 x 5 in image

```

```
## Warning: Removed 1055 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Warning: Removed 7 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Warning: Removed 7 rows containing missing values (geom_point).  
## Removed 1055 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Warning: Removed 1055 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Saving 6.5 x 5 in image  
## Warning: Removed 822 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Saving 6.5 x 5 in image  
## Warning: Removed 1055 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Warning: Removed 1468 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Warning: Removed 822 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Saving 6.5 x 5 in image  
## Warning: Removed 1055 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Warning: Removed 822 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Saving 6.5 x 5 in image  
## Warning: Removed 822 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Saving 6.5 x 5 in image  
## Warning: Removed 822 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Saving 6.5 x 5 in image  
## Warning: Removed 822 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Warning: Removed 1055 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Saving 6.5 x 5 in image  
## Warning: Removed 822 rows containing missing values (geom_point).  
## Saving 6.5 x 5 in image  
## Saving 6.5 x 5 in image  
## Warning: Removed 1055 rows containing missing values (geom_point).
```





```

## $num_iterations
## [1] 43
##
## $learning_rate
## [1] 0.03044482
##
## $max_depth

```

```

## [1] 15
##
## $feature_fraction_bynode
## [1] 1
##
## $min_data_in_leaf
## [1] 94
##
## $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] 77864
##
## $deterministic
## [1] TRUE
##
## $verbose
## [1] -1
##
## $metric
## list()
##
## $interaction_constraints
## list()
##
## $feature_pre_filter
## [1] FALSE

```

## Models Comparison

```

df_auc <- tribble::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$auc_upper,
  'Tuned Standard Model', standard_results$auc, standard_results$auc_lower, standard_results$auc_upper, length(standard_features)
) %>%
  mutate(Target = outcome_column,
        `Model (features)` = fct_reorder(paste0(Model, " - ", Features), -Features))

df_auc %>%
  ggplot(aes(

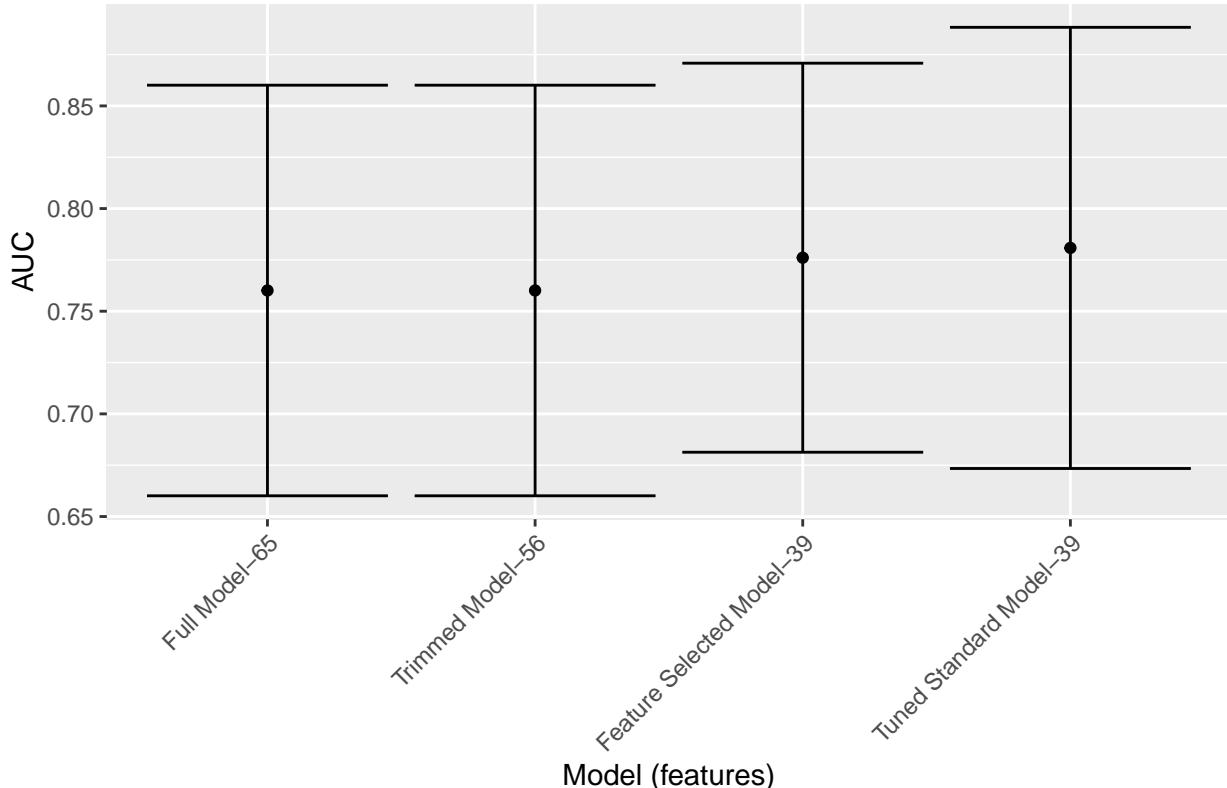
```

```

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))

```

death\_30days



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