The utility of analysing the response to initial prone-positioning in predicting ICU mortality using Logistic Regression with k-fold cross-validation

David M. Hannon

Description of dataset

There are a total of 169 columns in the dataset.

```
colnames(pre_post_changes)
```

```
  [1] "patient_id"
  [2] "proning_session"
  [3] "time_supine"
  [4] "sa_o2_systemic_supine"
  [5] "ph_abg_supine"
  [6] "pa_o2_supine"
  [7] "pa_co2_supine"
  [8] "bicarbonate_abg_a_supine"
  [9] "lactate_abg_supine"
 [10] "base_excess_vt_supine"
 [11] "potassium_abg_supine"
 [12] "sodium_abg_supine"
 [13] "ionised_calcium_abg_supine"
 [14] "anion_gap_abg_supine"
 [15] "glucose_abg_supine"
 [16] "total_haemoglobin_supine"
 [17] "fi_o2_supine"
 [18] "end_tidal_co2_marquette_supine"
 [19] "peep_supine"
 [20] "resp_rate_supine"
 [21] "heart_rate_supine"
 [22] "arterial_pressure_systolic_supine"
 [23] "white_cell_count_supine"
 [24] "neutrophils_supine"
 [25] "lymphocytes_supine"
 [26] "c_reactive_protein_supine"
 [27] "urea_supine"
 [28] "pcre_supine"
```

```
[29] "gfr_supine"
[30] "haematocrit_supine"
[31] "platelet_count_supine"
[32] "albumin_supine"
[33] "aa_gradient_p_aco2_supine"
[34] "aa_gradient_paco2_supine"
[35] "plateau_airway_pressure_d_supine"
[36] "resistance_d_supine"
[37] "minute_volume_coalesced_supine"
[38] "mean_airway_pressure_coalesced_supine"
[39] "peak_pressure_coalesced_supine"
[40] "hydrogen_ion_abg_nmolL_supine"
[41] "predicted_weight"
[42] "oxy_factor_supine"
[43] "ventilatory_ratio_supine"
[44] "pf_ratio_supine"
[45] "time_prone"
[46] "sa_o2_systemic_prone"
[47] "ph_abg_prone"
[48] "pa_o2_prone"
[49] "pa_co2_prone"
[50] "bicarbonate_abg_a_prone"
[51] "lactate_abg_prone"
[52] "base_excess_vt_prone"
[53] "potassium_abg_prone"
[54] "sodium_abg_prone"
[55] "ionised_calcium_abg_prone"
[56] "anion_gap_abg_prone"
[57] "glucose_abg_prone"
[58] "fi_o2_prone"
[59] "end_tidal_co2_marquette_prone"
[60] "peep_prone"
[61] "resp_rate_prone"
[62] "heart_rate_prone"
[63] "arterial_pressure_systolic_prone"
[64] "aa_gradient_p_aco2_prone"
[65] "aa_gradient_paco2_prone"
[66] "minute_volume_coalesced_prone"
[67] "mean_airway_pressure_coalesced_prone"
[68] "peak_pressure_coalesced_prone"
[69] "hydrogen_ion_abg_nmolL_prone"
[70] "oxy_factor_prone"
[71] "ventilatory_ratio_prone"
[72] "pf_ratio_prone"
[73] "time_supine_post"
[74] "sa_o2_systemic_supine_post"
[75] "ph_abg_supine_post"
```

```
 [76] "pa_o2_supine_post"
 [77] "pa_co2_supine_post"
 [78] "bicarbonate_abg_a_supine_post"
 [79] "lactate_abg_supine_post"
 [80] "base_excess_vt_supine_post"
 [81] "potassium_abg_supine_post"
 [82] "sodium_abg_supine_post"
 [83] "ionised_calcium_abg_supine_post"
 [84] "anion_gap_abg_supine_post"
 [85] "glucose_abg_supine_post"
 [86] "total_haemoglobin_supine_post"
 [87] "fi_o2_supine_post"
 [88] "end_tidal_co2_marquette_supine_post"
 [89] "peep_supine_post"
 [90] "resp_rate_supine_post"
 [91] "heart_rate_supine_post"
 [92] "arterial_pressure_systolic_supine_post"
 [93] "aa_gradient_p_aco2_supine_post"
 [94] "aa_gradient_paco2_supine_post"
 [95] "minute_volume_coalesced_supine_post"
 [96] "mean_airway_pressure_coalesced_supine_post"
 [97] "peak_pressure_coalesced_supine_post"
 [98] "hydrogen_ion_abg_nmolL_supine_post"
 [99] "oxy_factor_supine_post"
[100] "ventilatory_ratio_supine_post"
[101] "pf_ratio_supine_post"
[102] "gender"
[103] "age_years"
[104] "height_cm"
[105] "weight_kg"
[106] "ards_type"
[107] "ards_risk_factor"
[108] "pathogenic_factor"
[109] "adm_location"
[110] "los_days"
[111] "apache_ii"
[112] "total_proning_sessions"
[113] "ventilator"
[114] "outcome"
[115] "bmi"
[116] "time_between_abg"
[117] "sa_o2_change_absolute"
[118] "sa_o2_retain_absolute"
[119] "ph_change_absolute"
[120] "ph_retain_absolute"
[121] "hydrogen_ion_absolute"
[122] "hydrogen_ion_retain_absolute"
```

```
[123] "pa_o2_change_absolute"
[124] "pa_o2_retain_absolute"
[125] "pa_co2_change_absolute"
[126] "pa_co2_retain_absolute"
[127] "bicarbonate_change_absolute"
[128] "bicarbonate_retain_absolute"
[129] "lactate_abg_change_absolute"
[130] "lactate_abg_retain_absolute"
[131] "base_excess_change_absolute"
[132] "base_excess_retain_absolute"
[133] "potassium_abg_change_absolute"
[134] "potassium_abg_retain_absolute"
[135] "sodium_abg_change_absolute"
[136] "sodium_abg_retain_absolute"
[137] "ionised_calcium_abg_change_absolute"
[138] "ionised_calcium_abg_retain_absolute"
[139] "anion_gap_change_absolute"
[140] "anion_gap_retain_absolute"
[141] "glucose_change_absolute"
[142] "glucose_retain_absolute"
[143] "fi_o2_change_absolute"
[144] "fi_o2_retain_absolute"
[145] "et_co2_change_absolute"
[146] "et_co2_retain_absolute"
[147] "peep_change_absolute"
[148] "peep_retain_absolute"
[149] "resp_rate_change_absolute"
[150] "resp_rate_retain_absolute"
[151] "mean_airway_pressure_change_absolute"
[152] "mean_airway_pressure_retain_absolute"
[153] "peak_pressure_change_absolute"
[154] "peak_pressure_retain_absolute"
[155] "pfr_change_absolute"
[156] "pfr_retain_absolute"
[157] "oxy_factor_change_absolute"
[158] "oxy_factor_retain_absolute"
[159] "vent_ratio_change_absolute"
[160] "vent_ratio_retain_absolute"
[161] "aa_p_aco2_change_absolute"
[162] "aa_p_aco2_retain_absolute"
[163] "aa_paco2_change_absolute"
[164] "aa_paco2_retain_absolute"
[165] "minute_volume_change_absolute"
[166] "minute_volume_retain_absolute"
[167] "die_in_72"
[168] "die_in_120"
[169] "die_in_168"
```

Building the model

Create folds

Uses `rsample` package.

```
data_folds <- vfold_cv(data = pre_post_refined, v = folds, strata = outcome)
```

Prepare data

Uses `recipes` package. We must now create a recipe for the LR process. We will leave imputation to later, as it will be needed in each fold twice (training and testing data). This recipe:

- removes variables with > 5% data missing
- removes all numeric variables with a correlation > 0.8
- removes any variables possessing a single value
- creates dummy variables via one-hot encoding for all variables that are factors

Imputation of NA values (via bagged trees) will be done during the model fitting process.

```
lr_rec <- recipe(pre_post_refined, formula =  ~ .) %>%
  step_rm(fi_o2_change_absolute,
          resp_rate_supine,
          resp_rate_change_absolute,
          heart_rate_supine,
          arterial_pressure_systolic_supine) %>%
  step_filter_missing(all_predictors(), threshold = 0.05) %>% # leave only vars with <5% NA
  step_corr(all_numeric_predictors(), threshold = 0.8) %>%
  step_zv(all_predictors()) %>%
  step_dummy(all_nominal_predictors()) %>%
  prep()
```

Model fitting

```
# initiate empty table that will store performance metrics
lr_performance <- tibble(fold = 0,
                         accuracy = 0,
                         sensitivity = 0,
                         specificity = 0,
                         f_measure = 0,
                         kappa = 0)

# run a for loop that will fit the model to each of the folds
for (i in 1:folds) {
  data_lr_train <- data_folds$splits[[i]] %>% analysis()
  data_lr_train <- bake(object = lr_rec, new_data = data_lr_train)
```

```
lr_rec_2 <- recipe(data_lr_train, formula = ~ .) %>%
  step_impute_bag(all_predictors()) %>%
  prep()

data_lr_train <- bake(object = lr_rec_2,
                      new_data = data_lr_train) %>%
mutate(gender_m = as.factor(gender_m),
       ards_type_Covid_19 = as.factor(ards_type_Covid_19),
       ards_type_ARDSexp = as.factor(ards_type_ARDSexp),
       ards_type_Unknown = as.factor(ards_type_Unknown),
       outcome_rip = as.factor(outcome_rip))

lr_model <- glm(data = data_lr_train,
                formula = outcome_rip ~ .,
                family = 'binomial')

data_lr_test <- data_folds$splits[[i]] %>% assessment()
data_lr_test <- bake(object = lr_rec, new_data = data_lr_test)

data_lr_test <- bake(object = lr_rec_2,
                     new_data = data_lr_test) %>%
mutate(gender_m = as.factor(gender_m),
       ards_type_Covid_19 = as.factor(ards_type_Covid_19),
       ards_type_ARDSexp = as.factor(ards_type_ARDSexp),
       ards_type_Unknown = as.factor(ards_type_Unknown),
       outcome_rip = as.factor(outcome_rip))

threshold <- 0.5

data_lr_test$rip_odds <- predict(object = lr_model,
                                 newdata = data_lr_test,
                                 type = 'response')

data_lr_test$rip_odds <- as.numeric(data_lr_test$rip_odds)

data_lr_test$pred_rip <- if_else(condition = data_lr_test$rip_odds > threshold,
                                 true = 1,
                                 false = 0)

data_lr_test <- mutate(data_lr_test, pred_rip = as.factor(pred_rip))

acc_lr <- accuracy(data = data_lr_test, truth = outcome_rip, estimate = pred_rip)
sens_lr <- sensitivity(data = data_lr_test, truth = outcome_rip, estimate = pred_rip)
spec_lr <- specificity(data = data_lr_test, truth = outcome_rip, estimate = pred_rip)
fmeas_lr <- f_meas(data = data_lr_test, truth = outcome_rip, estimate = pred_rip)
```

```
  kappa_lr <- kap(data = data_lr_test, truth = outcome_rip, estimate = pred_rip)

  lr_performance<- add_row(lr_performance,
                           fold = i,
                           accuracy = round(acc_lr$.estimate, 2),
                           sensitivity = round(sens_lr$.estimate, 2),
                           specificity = round(spec_lr$.estimate, 2),
                           f_measure = round(fmeas_lr$.estimate, 2),
                           kappa = round(kappa_lr$.estimate, 2))

}

lr_performance <- lr_performance[2:(folds + 1), ]

lr_summary <- summarise(lr_performance,
                        accuracy_mean = mean(accuracy),
                        sensitivity_mean = mean(sensitivity),
                        specificity = mean(specificity),
                        f_meas_mean = mean(f_measure),
                        kappa_mean = mean(kappa))


# tidy
rm(acc_lr,
   auroc_lr,
   data_lr_test,
   data_lr_train,
   fmeas_lr,
   kappa_lr,
   lr_rec,
   lr_rec_2,
   sens_lr,
   spec_lr,
   i,
   threshold)
```

The resulting averaged metrics of the logistic regression model are given below.

```
# A tibble: 1 x 5
  accuracy_mean sensitivity_mean specificity f_meas_mean kappa_mean
          <dbl>            <dbl>       <dbl>       <dbl>      <dbl>
1         0.718            0.772       0.658       0.742      0.432
```