

Biostat 203B Homework 5 / Random Forest

Due Mar 20 @ 11:59PM

AUTHOR

Ningke Zhang 705834790

1. Load libraries

```
library(tidymodels)
```

— Attaching packages — tidymodels 1.3.0 —

✓ broom	1.0.7	✓ recipes	1.1.1
✓ dials	1.4.0	✓ rsample	1.2.1
✓ dplyr	1.1.4	✓ tibble	3.2.1
✓ ggplot2	3.5.1	✓ tidyr	1.3.1
✓ infer	1.0.7	✓ tune	1.3.0
✓ modeldata	1.4.0	✓ workflows	1.2.0
✓ parsnip	1.3.1	✓ workflowsets	1.1.0
✓ purrr	1.0.4	✓ yardstick	1.3.2

— Conflicts — tidymodels_conflicts() —

```
* purrr::discard() masks scales::discard()
* dplyr::filter()   masks stats::filter()
* dplyr::lag()      masks stats::lag()
* recipes::step()   masks stats::step()
```

```
library(dplyr)
library(recipes)
library(workflows)
library(tune)
library(glmnet)
```

Loading required package: Matrix

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

expand, pack, unpack

Loaded glmnet 4.1-8

```
library(vip)
```

Attaching package: 'vip'

The following object is masked from 'package:utils':

vi

```
library(ranger)
library(future)
```

2.Data preprocessing and feature engineering.

```
# read data
mimiciv_icu_cohort <- readRDS("../hw4/mimiciv_shiny/mimic_icu_cohort.rds") |>
  select(-c(intime,
            outtime,
            admittime,
            disctime,
            deathtime,
            admit_provider_id,
            edregtime,
            edouttime,
            anchor_age,
            anchor_year,
            anchor_year_group,
            last_careunit,
            discharge_location,
            hospital_expire_flag,
            dod,
            los)
  ) |>
  mutate(los_long = as.factor(los_long)) |>
  print(width = Inf)
```

3.Data split

```
set.seed(203)

mimiciv_icu_cohort <- mimiciv_icu_cohort |>
  arrange(subject_id, hadm_id, stay_id) |>
  select(-c(subject_id, hadm_id, stay_id))
mimiciv_icu_cohort <- mimiciv_icu_cohort |> drop_na()

data_split <- initial_split(mimiciv_icu_cohort,
                           strata = "los_long",
                           prop = 0.5)

icu_other <- training(data_split)
icu_test <- testing(data_split)
```

4. Train Random Forest model.

```

#define recipe
rf_recipe <- recipe(los_long ~ ., data = icu_other) |>
  step_impute_median(all_numeric_predictors()) |>
  step_impute_mode(all_nominal_predictors()) |>
  step_novel(all_nominal_predictors()) |>
  step_unknown(all_nominal_predictors()) |>
  step_dummy(all_nominal_predictors()) |>
  step_nzv(all_predictors()) |>
  step_normalize(all_numeric_predictors(), -all_outcomes())

#define model
rf_mod <-
  rand_forest(
    mode = "classification",
    mtry = tune(),
    trees = tune(),
    min_n = tune()) |>
  set_engine("ranger", importance = "permutation")

# Define the workflow
rf_wf <- workflow() |>
  add_recipe(rf_recipe) |>
  add_model(rf_mod)

# Define the grid
rf_grid <- grid_regular(
  mtry(range = c(2, 6)),
  trees(range = c(150, 200)),
  min_n(range = c(5, 10)),
  levels = c(3, 3, 3)
)

```

5. Cross-validation

```

set.seed(203)

folds <- vfold_cv(icu_other, v = 5)
folds

#Fit cross-validation
rf_fit <- rf_wf |>
  tune_grid(
    resamples = folds,
    grid = rf_grid,
    metrics = metric_set(roc_auc, accuracy),
    control = control_grid(save_pred = TRUE, verbose = TRUE)
  )
rf_fit

```

```
#visualize CV results
rf_fit |>
  collect_metrics() |>
  filter(.metric == "roc_auc") |>
  ggplot(aes(x = trees, y = mean, color = factor(mtry), group = factor(mtry))) +
  geom_point() +
  geom_line() +
  labs(
    title = "Random Forest Cross-Validation AUC",
    x = "Number of Trees",
    y = "Cross-Validation AUC",
    color = "mtry (Number of Variables)"
  ) +
  scale_x_continuous(breaks = seq(min(rf_grid$trees),
                                  max(rf_grid$trees), by = 50)) +
  theme_minimal()
```

6. Model evaluation

```
# select the best model
best_rf <- rf_fit |> select_best(metric = "roc_auc")
print(best_rf)

# finalize the workflow/fit
final_rf_wf <- finalize_workflow(rf_wf, best_rf)

final_rf_fit <- final_rf_wf |> last_fit(data_split)

saveRDS(final_rf_fit, "final_fit_rf_lastfit.rds")

final_rf_model <- final_rf_fit |> extract_workflow() |> extract_fit_parsnip()

final_rf_model |> vip()

saveRDS(final_rf_model, "final_fit_rf.rds")
```