Multivariate Linear Regression

Nail Bed Images

Nov 14, 2022

Function

```
# function to calculate model fit statistics
calc_model_stats <- function(x) {
  glance(extract_fit_parsnip(x)) |>
    select(adj.r.squared, AIC, BIC)
}
```

Packages

```
library(tidyverse)
library(tidymodels)
library(knitr)
```

Load data

```
nail_data <- read_csv("data/nail_data.csv") |>
  rename(concentration = `Concentration (g/dL)`)

glimpse(nail_data)
```

```
Rows: 72
Columns: 15
$ Image_URL     <chr> "https://storage.labelbox.com/cklyhytg2ulao0774uvyw0poy%~
$ xmin          <dbl> 248, 340, 469, 644, 347, 511, 640, 744, 201, 322, 458, 6~
```

```
$ xmax          <dbl> 292, 396, 527, 696, 385, 553, 682, 774, 248, 376, 509, 6~
$ ymin          <dbl> 537, 465, 414, 426, 398, 413, 472, 532, 767, 796, 700, 4~
$ ymax          <dbl> 582, 523, 474, 482, 461, 472, 522, 570, 803, 838, 726, 5~
$ Mean_H        <dbl> 8.1922, 13.0864, 12.3233, 13.1862, 11.3486, 11.9142, 10.~
$ Mean_S        <dbl> 0.2842, 0.2684, 0.2863, 0.2962, 0.2670, 0.2686, 0.2695, ~
$ Mean_V        <dbl> 0.7824, 0.8003, 0.7863, 0.7895, 0.7871, 0.7820, 0.7820, ~
$ Mean_L        <dbl> 68.7689, 68.3792, 68.0705, 68.5229, 68.4438, 68.2958, 67~
$ Mean_A        <dbl> 15.4397, 13.9356, 15.3458, 14.7066, 14.6354, 14.0090, 14~
$ Mean_B        <dbl> 10.6895, 12.0016, 11.8946, 14.3816, 11.9706, 11.9289, 11~
$ Mean_Prop_R   <dbl> 0.3999, 0.3980, 0.3943, 0.3994, 0.3972, 0.3997, 0.3977, ~
$ Mean_Prop_G   <dbl> 0.3077, 0.3125, 0.3147, 0.3140, 0.3113, 0.3115, 0.3107, ~
$ Mean_Prop_B   <dbl> 0.2925, 0.2895, 0.2910, 0.2866, 0.2915, 0.2888, 0.2916, ~
$ concentration <dbl> 9.5, 9.5, 9.5, 9.5, 9.5, 9.5, 9.5, 9.5, 9.2, 9.2, 9.2, 9~
```

Split data into training and testing

Split your data into testing and training sets.

```
set.seed(123)
nail_split <- initial_split(nail_data)
nail_train <- training(nail_split)
nail_test <- testing(nail_split)
```

Specify model

```
nail_spec <- linear_reg() |>
  set_engine("lm")

nail_spec
```

```
Linear Regression Model Specification (regression)

Computational engine: lm
```

Create recipe

```
nail_rec <- recipe(concentration ~ ., data = nail_train) |>
  update_role(Image_URL, new_role = "id") |>
  step_rm(xmin, xmax, ymin, ymax) |>
  step_dummy(all_nominal_predictors()) |>
  step_zv(all_predictors())

nail_rec
```

```
Recipe

Inputs:

      role #variables
        id          1
   outcome          1
 predictor         13

Operations:

Variables removed xmin, xmax, ymin, ymax
Dummy variables from all_nominal_predictors()
Zero variance filter on all_predictors()
```

Create workflow

```
nail_wflow <- workflow() |>
  add_model(nail_spec) |>
  add_recipe(nail_rec)

nail_wflow
```

```
== Workflow ====================================================================
Preprocessor: Recipe
Model: linear_reg()

-- Preprocessor ----------------------------------------------------------------
3 Recipe Steps
```

```
* step_rm()
* step_dummy()
* step_zv()

-- Model ---------------------------------------------------------------------
Linear Regression Model Specification (regression)

Computational engine: lm
```

Cross validation

Create folds

Create 10-folds.

```
# make 10 folds
set.seed(1)
folds <- vfold_cv(nail_train, v = 10)
folds
```

```
#  10-fold cross-validation
# A tibble: 10 x 2
   splits         id
   <list>         <chr>
 1 <split [48/6]> Fold01
 2 <split [48/6]> Fold02
 3 <split [48/6]> Fold03
 4 <split [48/6]> Fold04
 5 <split [49/5]> Fold05
 6 <split [49/5]> Fold06
 7 <split [49/5]> Fold07
 8 <split [49/5]> Fold08
 9 <split [49/5]> Fold09
10 <split [49/5]> Fold10
```

Conduct cross validation

Conduct cross validation on the 10 folds.

```r
set.seed(456)
# Fit model and calculate statistics for each fold
nail_fit_rs <- nail_wflow |>
  fit_resamples(resamples = folds,
                control = control_resamples(extract = calc_model_stats))
```

Summarize assessment CV metrics

Summarize assessment metrics from your CV resamples.

```r
collect_metrics(nail_fit_rs, summarize = FALSE) # summarize = FALSE to see individualized
```

```
# A tibble: 20 x 5
   id     .metric .estimator .estimate .config
   <chr>  <chr>   <chr>          <dbl> <chr>
 1 Fold01 rmse    standard     1.44    Preprocessor1_Model1
 2 Fold01 rsq     standard     0.00399 Preprocessor1_Model1
 3 Fold02 rmse    standard     1.12    Preprocessor1_Model1
 4 Fold02 rsq     standard     0.485   Preprocessor1_Model1
 5 Fold03 rmse    standard     1.47    Preprocessor1_Model1
 6 Fold03 rsq     standard     0.0835  Preprocessor1_Model1
 7 Fold04 rmse    standard     1.23    Preprocessor1_Model1
 8 Fold04 rsq     standard     0.129   Preprocessor1_Model1
 9 Fold05 rmse    standard     1.10    Preprocessor1_Model1
10 Fold05 rsq     standard     0.721   Preprocessor1_Model1
11 Fold06 rmse    standard     1.59    Preprocessor1_Model1
12 Fold06 rsq     standard     0.0332  Preprocessor1_Model1
13 Fold07 rmse    standard     1.32    Preprocessor1_Model1
14 Fold07 rsq     standard     0.258   Preprocessor1_Model1
15 Fold08 rmse    standard     0.730   Preprocessor1_Model1
16 Fold08 rsq     standard     0.444   Preprocessor1_Model1
17 Fold09 rmse    standard     2.31    Preprocessor1_Model1
18 Fold09 rsq     standard     0.0938  Preprocessor1_Model1
19 Fold10 rmse    standard     1.13    Preprocessor1_Model1
20 Fold10 rsq     standard     0.525   Preprocessor1_Model1
```

Summarize model fit CV metrics

```
map_df(nail_fit_rs$.extracts, ~ .x[[1]][[1]]) |> #model stats are in .extracts column
  summarise(mean_adj_rsq = mean(adj.r.squared), # avg_stats computed over 10 folds
            mean_aic = mean(AIC),
            mean_bic = mean(BIC))
```

```
# A tibble: 1 x 3
  mean_adj_rsq mean_aic mean_bic
         <dbl>    <dbl>    <dbl>
1        0.364     162.     183.
```

```
nail_fit_train <- nail_wflow |>
      fit(data = nail_train)

tidy(nail_fit_train) |>
  kable(digits=3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	1356.390	3815.071	0.356	0.724
Mean_H	0.004	0.005	0.741	0.463
Mean_S	31.719	18.789	1.688	0.098
Mean_V	16.880	25.667	0.658	0.514
Mean_L	-0.126	0.245	-0.512	0.611
Mean_A	-0.435	0.272	-1.596	0.118
Mean_B	-0.257	0.328	-0.783	0.438
Mean_Prop_R	-1306.425	3818.067	-0.342	0.734
Mean_Prop_G	-1435.883	3818.314	-0.376	0.709
Mean_Prop_B	-1319.198	3813.765	-0.346	0.731

```
nail_train_pred <- predict(nail_fit_train, nail_train) |>
        bind_cols(nail_train)

rmse_train <- rmse(nail_train_pred, truth = concentration, estimate = .pred)

nail_test_pred <- predict(nail_fit_train, nail_test) |>
        bind_cols(nail_test)

rmse_test <- rmse(nail_test_pred, truth = concentration, estimate = .pred)
```

```r
model_tibble <- tibble(RMSE_train=rmse_train$.estimate, RMSE_test=rmse_test$.estimate)

model_tibble |>
        kable()
```

RMSE_train	RMSE_test
1.041681	1.445858