Pipelines

1 Mar 2023

Table of contents

Materials from Modeling in R and Python

- `tidymodels`: meta-package for 'tidy' modeling in R
- `scikit-learn`: modeling in Python

`parsnip`

- `parsnip` package (CART → `caret` → `parsnip`)
- unify modeling interfaces (`lm`, `glmnet`, `randomForest`, etc etc etc)
- model (algorithm), mode (classification/regression), engine (implementation/package)

`rsample`

- resampling, cross-validation, bootstrapping, holdout sets …
- train/test split (`initial_split()`/`training()`/`testing()`)
- cross-validation (`vfold_cv()`), bootstrap (`bootstrap`)
- blocked/grouped methods! `group_vfold_cv`, `group_bootstraps()`

`recipes`

- feature engineering
- preprocessing (centering/scaling, imputation, dimension reduction, etc.)

more

- `workflows`: bundle preprocessing/modeling/post-processing
- `tune`: hyperparameter tuning
- `yardstick`: assessment

example

```r
library(tidyverse)
library(tidymodels)
historical <- (read_csv("../code/historical_baseball.csv")
    |> mutate(across(inducted, ~fct_rev(factor(.))))
    |> filter(ab > 250)
)
data_split <- initial_split(historical, prop = 2/3, strata = inducted)
train_data <- training(data_split)
testing_data <- testing(data_split)
```

```r
b_recipe  <- (
    recipe(inducted ~ ., data = train_data)
    |> step_rm("last_year")
    |> update_role(player_id, new_role = "ID") ## not predictor or outcome
    |> step_center(all_numeric())
    |> step_scale(all_numeric())
    |> step_nzv(all_numeric())
    |> prep()
)
```

```r
data(iris)
svm.model <- e1071::svm(Species ~ ., data = iris, probability = TRUE)
```

```r
pred <- predict(svm.model, iris, probability = TRUE)
prob <- attr(pred, "probabilities")
spmat <- matrix(0, ncol = length(levels(iris$Species)),
                nrow = length(iris$Species))
spmat[cbind(1:nrow(prob), as.numeric(iris$Species))] <- 1
-1 * mean(sapply(1:nrow(prob),
                 \(i) dmultinom(x = spmat[i,], size = 1, prob[i,], log = TRUE)))
```

```
[1] 0.07936639
```

```r
yardstick::mn_log_loss_vec(truth = iris$Species, estimate = prob)
```

```
[1] 0.07936639
```

```r
b_recipe |> bake(train_data) |> rsample::vfold_cv()
```

```
#  10-fold cross-validation
# A tibble: 10 x 2
   splits             id
   <list>             <chr>
 1 <split [1598/178]> Fold01
 2 <split [1598/178]> Fold02
 3 <split [1598/178]> Fold03
 4 <split [1598/178]> Fold04
 5 <split [1598/178]> Fold05
 6 <split [1598/178]> Fold06
 7 <split [1599/177]> Fold07
 8 <split [1599/177]> Fold08
 9 <split [1599/177]> Fold09
10 <split [1599/177]> Fold10
```

```r
lr_mod <-
    logistic_reg(mode = "classification", penalty = tune(), mixture = 1) %>%
    set_engine(engine = "glmnet")
```

```r
tt <- tune_grid(
     object = lr_mod,
     preprocessor = b_recipe,
     resamples = vfold_cv(train_data),
     metrics   = metric_set(mn_log_loss)
)
```