PSTAT131 HW4

Tianyi Li

2022-05-03

```
library(tidyverse)
library(tidymodels)
library(corrr)
library(poissonreg)
library(ISLR)
library(ISLR2)
library(ggplot2)
library(yardstick)
library(rlang)
library(corrplot)
library(discrim)
library(klaR)
library(pROC)
library(knitr)
tidymodels_prefer()
```

```
titanic = read.csv('titanic.csv')
titanic$pclass <- factor(titanic$pclass)
titanic$survived <- factor(titanic$survived, ordered=TRUE, levels=c('Yes','No'))
titanic_split <- initial_split(titanic, prop = 0.80,strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
set.seed(1202)
titanic_recipe <- recipe(survived ~ pclass + sex + age + sib_sp +
                           parch + fare, data = titanic_train) %>%
  step_impute_linear(age) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(terms= ~ starts_with("sex"):fare+
                  age:fare)
titanic_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          6
##
## Operations:
##
## Linear regression imputation for age
## Dummy variables from all_nominal_predictors()
```

```
## Interactions with starts_with("sex"):fare + age:fare
```

Question 1

```r
titanic_split <- initial_split(titanic, prop = 0.80,strata = survived)
dim(titanic_train)
```

```
## [1] 712  12
```

```r
dim(titanic_test)
```

```
## [1] 179  12
```

Question 2

```r
train_folds <- vfold_cv(titanic_train, v = 10)
train_folds
```

```
## #  10-fold cross-validation
## # A tibble: 10 x 2
##    splits           id
##    <list>           <chr>
##  1 <split [640/72]> Fold01
##  2 <split [640/72]> Fold02
##  3 <split [641/71]> Fold03
##  4 <split [641/71]> Fold04
##  5 <split [641/71]> Fold05
##  6 <split [641/71]> Fold06
##  7 <split [641/71]> Fold07
##  8 <split [641/71]> Fold08
##  9 <split [641/71]> Fold09
## 10 <split [641/71]> Fold10
```

Question 3

k-fold cross-validation is a re-sampling method where a given data set is split into a K number of sections, and each section is used to test machine learning models within a limited data sample.

Because k-fold CV generates a less-biased estimate of a model, which also reduces the computation time.

If we use the entire training set, the re-sampling method would be Bootstrap.

Question 4

There are 3 models and each with 10 folds, thus 30 folds in total.

```r
log_reg = logistic_reg() %>%
        set_engine("glm") %>%
        set_mode("classification")
log_wkflow = workflow() %>%
        add_model(log_reg) %>%
        add_recipe(titanic_recipe)
log_fit = fit(log_wkflow, titanic_train)

lda_mod = discrim_linear() %>%
        set_engine("MASS") %>%
        set_mode("classification")
lda_wkflow = workflow() %>%
        add_model(lda_mod) %>%
        add_recipe(titanic_recipe)
lda_fit = fit(lda_wkflow, titanic_train)
```

```r
qda_mod = discrim_quad() %>%
        set_mode("classification") %>%
        set_engine("MASS")
qda_wkflow = workflow() %>%
        add_model(qda_mod) %>%
        add_recipe(titanic_recipe)
qda_fit = fit(qda_wkflow, titanic_train)
```

Question 5

```r
log_fit <- fit_resamples(log_wkflow,train_folds)
lda_fit <- fit_resamples(lda_wkflow,train_folds)
qda_fit <- fit_resamples(qda_wkflow,train_folds)
```

Question 6

The logistic regression model has performed the best, because it has the highest mean accuracy and a relatively low standard error.

```r
collect_metrics(log_fit)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.801    10  0.0215 Preprocessor1_Model1
## 2 roc_auc  binary     0.853    10  0.0215 Preprocessor1_Model1
```

```r
collect_metrics(lda_fit)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.788    10  0.0217 Preprocessor1_Model1
## 2 roc_auc  binary     0.854    10  0.0207 Preprocessor1_Model1
```

```r
collect_metrics(qda_fit)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.782    10  0.0154 Preprocessor1_Model1
## 2 roc_auc  binary     0.847    10  0.0210 Preprocessor1_Model1
```

Question 7

```r
log1_fit = fit(log_wkflow, titanic_train)
log1_fit
```

```
## == Workflow [trained] ==========================================================
## Preprocessor: Recipe
## Model: logistic_reg()
##
## -- Preprocessor ----------------------------------------------------------------
## 3 Recipe Steps
##
## * step_impute_linear()
## * step_dummy()
## * step_interact()
```

```
##
## -- Model ----------------------------------------------------------------------
##
## Call:  stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)
##
## Coefficients:
##     (Intercept)              age           sib_sp            parch
##       -4.692824         0.058407         0.448391         0.131479
##            fare        pclass_X2        pclass_X3         sex_male
##        0.004280         1.307633         2.453779         2.685452
## sex_male_x_fare       fare_x_age
##        0.005908        -0.000315
##
## Degrees of Freedom: 711 Total (i.e. Null);  702 Residual
## Null Deviance:       948
## Residual Deviance: 616.4     AIC: 636.4
```

Question 8

The model's testing accuracy is 0.8100559, and its average accuracy is 0.8230337

The two statistic are close to each other, while the model's testing accuracy is lower.

```
log_pred <- predict(log1_fit, new_data = titanic_test, type = "class")
bind_cols(log_pred,titanic_test$survived)
```

```
## # A tibble: 179 x 2
##    .pred_class ...2
##    <fct>       <ord>
##  1 Yes         Yes
##  2 Yes         Yes
##  3 No          No
##  4 Yes         Yes
##  5 No          No
##  6 No          No
##  7 Yes         Yes
##  8 No          No
##  9 No          No
## 10 No          No
## # ... with 169 more rows
```

```
train_acc <- augment(log1_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
train_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.808
```

```
test_acc <- augment(log1_fit, new_data = titanic_test) %>%
  accuracy(truth = survived, estimate = .pred_class)
test_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.799
```