Pstat 131 Homework 4

Yu Tian

Spring 2022-05-03

Resampling

```
# Read the titanic data set into R using read_csv()
titanic <- read_csv(file = "titanic.csv") %>%
  mutate(survived = factor(survived, levels = c("Yes", "No")),
         pclass = factor(pclass))
titanic %>% head()
```

View Titanic Date

```
## # A tibble: 6 x 12
##   passenger_id survived pclass name  sex     age sib_sp parch ticket  fare cabin
##          <dbl> <fct>    <fct>  <chr> <chr> <dbl>  <dbl> <dbl> <chr>  <dbl> <chr>
## 1            1 No       3      Brau~ male     22      1     0 A/5 2~  7.25 <NA>
## 2            2 Yes      1      Cumi~ fema~    38      1     0 PC 17~ 71.3  C85
## 3            3 Yes      3      Heik~ fema~    26      0     0 STON/~  7.92 <NA>
## 4            4 Yes      1      Futr~ fema~    35      1     0 113803 53.1  C123
## 5            5 No       3      Alle~ male     35      0     0 373450  8.05 <NA>
## 6            6 No       3      Mora~ male     NA      0     0 330877  8.46 <NA>
## # ... with 1 more variable: embarked <chr>
```

Question 1

Split the data, stratifying on the outcome variable, survived. You should choose the proportions to split the data into. Verify that the training and testing data sets have the appropriate number of observations.

Answer Q1

```
# set a seed
set.seed(0623)

# split the titanic data into a training set and a testing set.
titanic_split <- initial_split(titanic, prop = 0.80, strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
dim(titanic)
```

```
## [1] 891  12
```

```
dim(titanic_train)
```

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

```
dim(titanic_test)
```

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

```
# Verify the training and testing data sets have the appropriate number of observations
# the number of observations for all data
a <- nrow(titanic)
a
```

```
## [1] 891
```

```
# the number of observations for training data
b <- nrow(titanic_train)
b
```

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

```
# the number of observations for test data
c <- nrow(titanic_test)
c
```

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

```
# the percentage of observations for training data
b/a
```

```
## [1] 0.7991021
```

```
# the percentage of observations for test data
c/a
```

```
## [1] 0.2008979
```

The probability of training data observations is 0.7991021, which is almost equal to prob=0.80, so the training and testing data sets have the appropriate number of observations

```
# create a recipe identical to the recipe you used in Homework 3
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"):age + 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"):age + age:fare
```

Question 2

Fold the training data. Use k-fold cross-validation, with k=10.

Answer Q2

```
titanic_folds <- vfold_cv(titanic_train, k = 10)
titanic_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

In your own words, explain what we are doing in Question 2. What is k-fold cross-validation? Why should we use it, rather than simply fitting and testing models on the entire training set? If we did use the entire training set, what resampling method would that be?

Answer Q3

(part of words cited from lecture slides)

In Question 2, we are trying to use k-fold cross-validation and divide the testing data into 10 groups of roughly equal size to prepare for the later fitting and prediction process.

k-fold cross-validation is one kind of resampling method. For each model, this method will randomly divide the observation data into k groups of roughly equal sizes, which are folds. This method will hold out the 1st fold as the validation set to be evaluated. Then the remaining k-1 folds will be analyzed to fit the model. The final estimate of model will get by the average of k results.

Compared with simply fitting and testing models on the entire training set, it will avoid the over optimistic estimate based on the all training data and overestimate of the testing data.

If we use the entire training set, resampling method would be validation set approach.

Question 4

Set up workflows for 3 models:

A logistic regression with the glm engine; A linear discriminant analysis with the MASS engine; A quadratic discriminant analysis with the MASS engine. How many models, total, across all folds, will you be fitting to the data? To answer, think about how many folds there are, and how many models you'll fit to each fold.

Answer Q4

```
# set up workflows for a logistic regression with the glm engine
log_reg <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")
```

```
log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(titanic_recipe)
```

```
# set up workflows for a linear discriminant analysis with the MASS engine
lda_mod <- discrim_linear() %>%
  set_mode("classification") %>%
  set_engine("MASS")

lda_wkflow <- workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(titanic_recipe)
```

```
# set up workflows for a quadratic discriminant analysis with the MASS engine
qda_mod <- discrim_quad() %>%
  set_mode("classification") %>%
  set_engine("MASS")

qda_wkflow <- workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(titanic_recipe)
```

30 models in total, across all folds, will be fitting to the data. There are 10 folds and 3 models I will fit to each fold, so total number is 3*10=30.

Question 5

Fit each of the models created in Question 4 to the folded data.

Answer Q5

```
# fit the logistic regression model
log_fit <- log_wkflow %>%
  fit_resamples(titanic_folds)

# fit the linear discriminant analysis model
lda_fit <- lda_wkflow %>%
  fit_resamples(titanic_folds)

#fit the quadratic discriminant analysis model
qda_fit <- qda_wkflow %>%
  fit_resamples(titanic_folds)
```

Question 6

Use collect_metrics() to print the mean and standard errors of the performance metric accuracy across all folds for each of the four models.

Decide which of the 3 fitted models has performed the best. Explain why. (Note: You should consider both the mean accuracy and its standard error.)

Answer Q6

```
# Use collect_metrics() to print the mean and standard errors of the performance metric accuracy
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     0.820    10 0.00988 Preprocessor1_Model1
## 2 roc_auc  binary     0.871    10 0.0188  Preprocessor1_Model1
```

```
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     0.809    10  0.0143 Preprocessor1_Model1
## 2 roc_auc  binary     0.867    10  0.0187 Preprocessor1_Model1
```

```
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     0.803    10  0.0155 Preprocessor1_Model1
## 2 roc_auc  binary     0.853    10  0.0148 Preprocessor1_Model1
```

The logistic regression model performs the best, since it has highest mean and lowest standard deviation of the performance metric accuracy across all folds.

Question 7

Now that you've chosen a model, fit your chosen model to the entire training dataset (not to the folds).

Answer Q7

```
log_fit_entire <- fit(log_wkflow, titanic_train)
```

Question 8

Finally, with your fitted model, use predict(), bind_cols(), and accuracy() to assess your model's performance on the testing data!

Compare your model's testing accuracy to its average accuracy across folds. Describe what you see.

Answer Q8

```
# use predict(), bind_cols(), and accuracy()to assess the model's performance on the testing data
predict(log_fit_entire, new_data = titanic_test, type = "prob")
```

```
## # A tibble: 179 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1    0.567     0.433
##  2    0.861     0.139
##  3    0.626     0.374
##  4    0.667     0.333
##  5    0.538     0.462
##  6    0.107     0.893
##  7    0.305     0.695
##  8    0.144     0.856
##  9    0.639     0.361
```

```
## 10    0.0874    0.913
## # ... with 169 more rows
```

```
log_reg_acc_test <-
  predict(log_fit_entire, new_data = titanic_test) %>%
  bind_cols(titanic_test %>% select(survived)) %>%
  bind_cols(predict(log_fit_entire, titanic_test, type = "prob")) %>%
  accuracy(truth = survived, estimate = .pred_class)
log_reg_acc_test
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.793
```

The model's testing accuracy is 0.7932961, and the accuracy across folds is 0.82. Thus, the testing accuracy is close to the average accuracy across folds, so the k-fold cross-validation method fits well.