131-Homework 4

Caleb Mazariegos

2022-05-06

```
# setting survived and pclass as factors, reordering survived so that "Yes" is the first level
titanic_codebook$survived <- as.factor(titanic_codebook$survived)
titanic_codebook$survived <- factor(titanic_codebook$survived, levels = c("Yes", "No"))
titanic_codebook$pclass <- as.factor(titanic_codebook$pclass)
```

Question 1

```
# Setting the seed
set.seed(3435)
titanic_split <- initial_split(titanic_codebook, prop = 0.75, strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
```

```
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) %>%
  step_interact(terms = ~ age:fare)
titanic_recipe
```

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.

```
## 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
## Interactions with age:fare
```

```
Auto_train <- training(titanic_split)
Auto_test <- testing(titanic_split)
dim(Auto_train)
```

```
## [1] 667  12
```

```
dim(Auto_test)
```

```
## [1] 224  12
```

Question 2

```
set.seed(234)
train_folds <- vfold_cv(Auto_train, v=10)
train_folds
```

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

```
## #  10-fold cross-validation
## # A tibble: 10 x 2
##    splits           id
##    <list>           <chr>
##  1 <split [600/67]> Fold01
##  2 <split [600/67]> Fold02
##  3 <split [600/67]> Fold03
##  4 <split [600/67]> Fold04
##  5 <split [600/67]> Fold05
##  6 <split [600/67]> Fold06
##  7 <split [600/67]> Fold07
##  8 <split [601/66]> Fold08
##  9 <split [601/66]> Fold09
## 10 <split [601/66]> 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? K-fold cross verification is a statistical method used to estimate the skill of machine learning models. In question 2 we are using it to find the best value of degree that yields the "closest fit". We should use k-fold cross-validation instead of fitting and testing models on the entire training set because k-fold cross validation resamples without replacement, which creates data sets that are smaller than the original data set. If we did use the entire training set, the resampling method we would use is bootstrap.

Question 4

Set up workflows for 3 models:

```
log_reg <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

log_workflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(titanic_recipe)

log_fit <- fit(log_workflow, titanic_train)

log_fit %>%
  tidy()
```

- A logistic regression with the glm engine;

```
## # A tibble: 10 x 5
##    term             estimate std.error statistic  p.value
##    <chr>               <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)     -4.02      0.644       -6.25  4.14e-10
##  2 age              0.0531    0.0126       4.21  2.54e- 5
##  3 sib_sp           0.462     0.132        3.50  4.70e- 4
##  4 parch            0.183     0.147        1.25  2.13e- 1
##  5 fare            -0.00729   0.0106      -0.689 4.91e- 1
##  6 pclass_X2        1.10      0.351        3.14  1.67e- 3
##  7 pclass_X3        2.20      0.364        6.05  1.48e- 9
##  8 sex_male         2.36      0.301        7.85  4.29e-15
##  9 sex_male_x_fare  0.0133    0.00836      1.60  1.10e- 1
## 10 age_x_fare      -0.000237  0.000190    -1.25  2.12e- 1
```

```
lda_mod <- discrim_linear() %>%
    set_mode("classification") %>%
    set_engine("MASS")


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

lda_fit <- fit(lda_wkflow, titanic_train)
```

- A linear discriminant analysis with the MASS engine;

```
qda_mod <- discrim_quad() %>%
  set_engine("MASS") %>%
  set_mode("classification")

qda_workflow <- workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(titanic_recipe)

qda_fit <- fit(qda_workflow, titanic_train)
```

- 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. We will be fitting 30 models across all folds. This is because there are 10 folds and 3 models.

Question 5

```
log_fit <-
  log_workflow %>%
  fit_resamples(train_folds)
log_fit
```

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

```
## # Resampling results
## # 10-fold cross-validation
## # A tibble: 10 x 4
##    splits           id     .metrics          .notes
##    <list>           <chr>  <list>            <list>
##  1 <split [600/67]> Fold01 <tibble [2 x 4]> <tibble [0 x 3]>
##  2 <split [600/67]> Fold02 <tibble [2 x 4]> <tibble [0 x 3]>
##  3 <split [600/67]> Fold03 <tibble [2 x 4]> <tibble [0 x 3]>
##  4 <split [600/67]> Fold04 <tibble [2 x 4]> <tibble [0 x 3]>
##  5 <split [600/67]> Fold05 <tibble [2 x 4]> <tibble [0 x 3]>
##  6 <split [600/67]> Fold06 <tibble [2 x 4]> <tibble [0 x 3]>
##  7 <split [600/67]> Fold07 <tibble [2 x 4]> <tibble [0 x 3]>
##  8 <split [601/66]> Fold08 <tibble [2 x 4]> <tibble [0 x 3]>
##  9 <split [601/66]> Fold09 <tibble [2 x 4]> <tibble [0 x 3]>
## 10 <split [601/66]> Fold10 <tibble [2 x 4]> <tibble [0 x 3]>
```

```
lda_fit <-
  lda_wkflow %>%
  fit_resamples(train_folds)
lda_fit
```

```
## # Resampling results
```

```
## # 10-fold cross-validation
## # A tibble: 10 x 4
##    splits           id     .metrics         .notes
##    <list>           <chr>  <list>           <list>
##  1 <split [600/67]> Fold01 <tibble [2 x 4]> <tibble [0 x 3]>
##  2 <split [600/67]> Fold02 <tibble [2 x 4]> <tibble [0 x 3]>
##  3 <split [600/67]> Fold03 <tibble [2 x 4]> <tibble [0 x 3]>
##  4 <split [600/67]> Fold04 <tibble [2 x 4]> <tibble [0 x 3]>
##  5 <split [600/67]> Fold05 <tibble [2 x 4]> <tibble [0 x 3]>
##  6 <split [600/67]> Fold06 <tibble [2 x 4]> <tibble [0 x 3]>
##  7 <split [600/67]> Fold07 <tibble [2 x 4]> <tibble [0 x 3]>
##  8 <split [601/66]> Fold08 <tibble [2 x 4]> <tibble [0 x 3]>
##  9 <split [601/66]> Fold09 <tibble [2 x 4]> <tibble [0 x 3]>
## 10 <split [601/66]> Fold10 <tibble [2 x 4]> <tibble [0 x 3]>
```

```
qda_fit <-
  qda_workflow %>%
  fit_resamples(train_folds)
qda_fit
```

```
## # Resampling results
## # 10-fold cross-validation
## # A tibble: 10 x 4
##    splits           id     .metrics         .notes
##    <list>           <chr>  <list>           <list>
##  1 <split [600/67]> Fold01 <tibble [2 x 4]> <tibble [0 x 3]>
##  2 <split [600/67]> Fold02 <tibble [2 x 4]> <tibble [0 x 3]>
##  3 <split [600/67]> Fold03 <tibble [2 x 4]> <tibble [0 x 3]>
##  4 <split [600/67]> Fold04 <tibble [2 x 4]> <tibble [0 x 3]>
##  5 <split [600/67]> Fold05 <tibble [2 x 4]> <tibble [0 x 3]>
##  6 <split [600/67]> Fold06 <tibble [2 x 4]> <tibble [0 x 3]>
##  7 <split [600/67]> Fold07 <tibble [2 x 4]> <tibble [0 x 3]>
##  8 <split [601/66]> Fold08 <tibble [2 x 4]> <tibble [0 x 3]>
##  9 <split [601/66]> Fold09 <tibble [2 x 4]> <tibble [0 x 3]>
## 10 <split [601/66]> Fold10 <tibble [2 x 4]> <tibble [0 x 3]>
```

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.)