Homework 4

PSTAT 131/231

Contents

```
library(tidyverse)
library(tidymodels)
library(ISLR)
library(ISLR2)
library(discrim)
```

Resampling

Load the data from `data/titanic.csv` into *R* and familiarize yourself with the variables it contains using the codebook (`data/titanic_codebook.txt`).

Notice that `survived` and `pclass` should be changed to factors. When changing `survived` to a factor, you may want to reorder the factor so that *"Yes"* is the first level.

Make sure you load the `tidyverse` and `tidymodels`!

Remember that you'll need to set a seed at the beginning of the document to reproduce your results.

Create a recipe for this dataset **identical** to the recipe you used in Homework 3.

```
set.seed(1234)
titanic <- read.csv("C:/Users/rocke/Downloads/homework-3/homework-3/data/titanic.csv")  # reading in ti
titanic$pclass <- factor(titanic$pclass)  # changing pclass into a factor
titanic$survived <- factor(titanic$survived, levels = c("Yes", "No"))  # changing survived into a facto
```

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.

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

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

```
dim(titanic_test)
```

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

Question 2

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

```
titanic_folds <- vfold_cv(titanic_train, v = 10)
```

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?

In question 2, we are splitting our training data into 10 folds of roughly equal size. K-fold cross-validation is a resampling method that allows us to get more robust estimates of model performance. It is when you divide the data into k folds, hold out one fold, fit the model to the rest of the folds, calculate accuracy or MSE on the held out fold, and repeat this process k times. We should use it rather than simply fitting and testing models on the entire training set because k-fold cross-validation will give a more accurate measure of model performance (less biased). If we used the entire training set, we would be using the validation set approach.

Question 4

Set up workflows for 3 models:

1. A logistic regression with the `glm` engine;
2. A linear discriminant analysis with the `MASS` engine;
3. A quadratic discriminant analysis with the `MASS` engine.

```
titanic_recipe <- recipe(survived ~ pclass + sex + age + sib_sp + parch + fare,
                         data = titanic_train) %>%
  step_impute_linear(age, impute_with = imp_vars(sib_sp)) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(terms = ~starts_with("sex"):fare) %>%
  step_interact(terms = ~ age:fare)

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

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

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

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


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


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

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.

I would be fitting 30 models (10 + 10 + 10) total. This is because for each model type, I would be fitting 10 models.

Question 5

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

```
log_folded <- log_titanic_workflow %>% fit_resamples(titanic_folds)

lda_folded <- lda_titanic_workflow %>% fit_resamples(titanic_folds)

qda_folded <- qda_titanic_workflow %>% 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.)*

```
collect_metrics(log_folded)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.802    10  0.0214 Preprocessor1_Model1
## 2 roc_auc  binary     0.852    10  0.0180 Preprocessor1_Model1
```

```
collect_metrics(lda_folded)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.795    10  0.0204 Preprocessor1_Model1
## 2 roc_auc  binary     0.853    10  0.0176 Preprocessor1_Model1
```

```
collect_metrics(qda_folded)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.784    10  0.0175 Preprocessor1_Model1
## 2 roc_auc  binary     0.848    10  0.0218 Preprocessor1_Model1
```

I think the model that performed the best is the logistic regression model due to it having the highest accuracy and similar standard errors to the other models. Additionally, the standard error is quite small, so I have no doubts that its performance is near the calculated accuracy.

Question 7

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

```
log_fit <- fit(log_titanic_workflow, 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!

```
log_acc <- predict(log_fit, new_data = titanic_train, type = "class") %>%
  bind_cols(titanic_train %>% dplyr::select(survived)) %>%
  accuracy(truth = survived, estimate = .pred_class)

log_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.813
```

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

My model's testing accuracy is 0.8132, while its average accuracy across the folds is 0.802. I see that the testing accuracy lies within one standard error of the average accuracy, which indicates that the k-fold cross validation method worked quite well!