131 Homework 4

Tonia Wu

Resampling

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.

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 <- titanic %>%
  initial_split(strata = survived, prop = 0.8)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
dim(titanic_train)
```

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

```
dim(titanic_test)
```

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

The testing set has 179 observations, and the training set has 712, which is almost 80% of the full data set of 891 observations.

Question 2

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

```
# create recipe
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(~ starts_with("sex"):age + age:fare)
```

```
# k-fold cross-validation
titanic_folds <- vfold_cv(titanic_train, v = 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?

k-fold cross-validation is used to evaluate model performance. It involves randomizing the dataset, splitting it into k groups, and taking out one group at a time as a test data set. Generally the results are less biased than if we simply split the data with train and test. If we did use the entire training set, it becomes the validation set approach.

Question 4

Set up workflows for 3 models:

1. 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)
```

2. 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)
```

3. 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)
```

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 have three different types of models. Since we fit three models for each of the 10 folds, we will have 30 fitted models.

Question 5

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

```
log_fold <-  log_wkflow %>%
  fit_resamples(titanic_folds)

lda_fold <- lda_wkflow %>%
  fit_resamples(titanic_folds)
```

```
## Warning: package 'MASS' was built under R version 4.1.2
```

```
qda_fold <- 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.)*

```
collect_metrics(log_fold)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.799    10  0.0148 Preprocessor1_Model1
## 2 roc_auc  binary     0.844    10  0.0149 Preprocessor1_Model1
```

```
collect_metrics(lda_fold)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.791    10  0.0187 Preprocessor1_Model1
## 2 roc_auc  binary     0.845    10  0.0153 Preprocessor1_Model1
```

```
collect_metrics(qda_fold)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.782    10  0.0159 Preprocessor1_Model1
## 2 roc_auc  binary     0.829    10  0.0153 Preprocessor1_Model1
```

The logistic regression model had the highest mean accuracy and the lowest standard error of all the models, so we will now use it to fit the training dataset.

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_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.

```
predict(log_fit, new_data = titanic_test) %>% bind_cols(titanic_test%>% dplyr::select(survived)) %>%
accuracy(truth=survived, estimate = .pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.832
```

The testing accuracy, 0.799, is slightly lower than 0.832, the average accuracy across folds. This could be the result of variance in the points selected for the testing sets; in this case, the folded data performed slightly better.