Homework 4

Yixin Wang

Contents

```
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 0.2.0 --

## v broom        0.8.0     v recipes      0.2.0
## v dials        0.1.1     v rsample      0.1.1
## v dplyr        1.0.8     v tibble       3.1.6
## v ggplot2      3.3.5     v tidyr        1.2.0
## v infer        1.0.0     v tune         0.2.0
## v modeldata    0.1.1     v workflows    0.2.6
## v parsnip      0.2.1     v workflowsets 0.2.1
## v purrr        0.3.4     v yardstick    0.0.9

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v readr   1.4.0     v forcats 0.5.1
## v stringr 1.4.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x stringr::fixed()    masks recipes::fixed()
## x dplyr::lag()        masks stats::lag()
## x readr::spec()       masks yardstick::spec()
```

```
library(discrim)
```

```
##
## Attaching package: 'discrim'
```

```
## The following object is masked from 'package:dials':
##
##     smoothness
```

```
library(poissonreg)
library(corrr)
library(klaR)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(ISLR)
library(ISLR2)
```

```
##
## Attaching package: 'ISLR2'

## The following objects are masked from 'package:ISLR':
##
##     Auto, Credit

## The following object is masked from 'package:MASS':
##
##     Boston
```

```
library(sf)
```

```
## Linking to GEOS 3.9.1, GDAL 3.4.0, PROJ 8.1.1; sf_use_s2() is TRUE
```

```
tidymodels_prefer()
setwd('/Users/galaxy/Desktop/PSTAT_131')
set.seed(3435)
```

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 <- read_csv(file = "titanic.csv") %>%
  mutate(survived = factor(survived,
                           levels = c("Yes", "No")),
         pclass = factor(pclass))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   passenger_id = col_double(),
##   survived = col_character(),
##   pclass = col_double(),
##   name = col_character(),
##   sex = col_character(),
##   age = col_double(),
##   sib_sp = col_double(),
##   parch = col_double(),
##   ticket = col_character(),
##   fare = col_double(),
##   cabin = col_character(),
##   embarked = col_character()
## )
```

```
titanic
```

```
## # A tibble: 891 x 12
##    passenger_id survived pclass name       sex     age sib_sp parch ticket  fare
##           <dbl> <fct>    <fct>  <chr>      <chr> <dbl>  <dbl> <dbl> <chr>  <dbl>
##  1            1 No       3      Braund, M~ male     22      1     0 A/5 2~  7.25
##  2            2 Yes      1      Cumings, ~ fema~    38      1     0 PC 17~ 71.3
##  3            3 Yes      3      Heikkinen~ fema~    26      0     0 STON/~  7.92
##  4            4 Yes      1      Futrelle,~ fema~    35      1     0 113803 53.1
##  5            5 No       3      Allen, Mr~ male     35      0     0 373450  8.05
##  6            6 No       3      Moran, Mr~ male     NA      0     0 330877  8.46
##  7            7 No       1      McCarthy,~ male     54      0     0 17463  51.9
##  8            8 No       3      Palsson, ~ male      2      3     1 349909 21.1
##  9            9 Yes      3      Johnson, ~ fema~    27      0     2 347742 11.1
## 10           10 Yes      2      Nasser, M~ fema~    14      1     0 237736 30.1
## # ... with 881 more rows, and 2 more variables: cabin <chr>, embarked <chr>
```

```
titanic_split <- titanic %>%
  initial_split(strata = survived, prop = 0.7)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)

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

```
dim(titanic_train)
```

```
## [1] 623  12
```

```
dim(titanic_test)
```

```
## [1] 268  12
```

Question 2

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

```
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 [560/63]> Fold01
##  2 <split [560/63]> Fold02
##  3 <split [560/63]> Fold03
##  4 <split [561/62]> Fold04
##  5 <split [561/62]> Fold05
##  6 <split [561/62]> Fold06
##  7 <split [561/62]> Fold07
##  8 <split [561/62]> Fold08
##  9 <split [561/62]> Fold09
## 10 <split [561/62]> 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?

In Question 2, we randomly partitioned the training titanic sample into 10 sub samples. k-fold cross-validation is randomly divide the data into k groups(or folds) of equal sizes. Then hold out the first fold as the validation set, and the model is fit on the remaining k-1 folds. Then repeat the process for k times. We use it because the observations are used for training and validation, and observation is used for validation once. When we use the entire training set, the method will be 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.

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.

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

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

```r
log_fit <- fit_resamples(log_wkflow, titanic_folds)
```

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

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

lda_fit <- fit_resamples(lda_wkflow, titanic_folds)
```

```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_resamples(qda_wkflow, titanic_folds)
```

There are ten folds for each model.

Question 5

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

```r
log_fit <- fit_resamples(log_wkflow, titanic_folds)
lda_fit <- fit_resamples(lda_wkflow, titanic_folds)
qda_fit <- fit_resamples(qda_wkflow, 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.)*

```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     0.809    10  0.0119 Preprocessor1_Model1
## 2 roc_auc  binary     0.844    10  0.0146 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     0.796    10  0.0125 Preprocessor1_Model1
## 2 roc_auc  binary     0.840    10  0.0155 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     0.787    10  0.0186 Preprocessor1_Model1
## 2 roc_auc  binary     0.827    10  0.0129 Preprocessor1_Model1
```

The logistic regression has performed the best, since the mean accuracy is high and standard error is low.

Question 7

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

```r
log_fit1 <- fit(log_wkflow, titanic_train)
log_fit1 %>% tidy()
```

```
## # A tibble: 10 x 5
##    term            estimate std.error statistic  p.value
##    <chr>              <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)    -4.01      0.675        -5.93 2.97e- 9
##  2 age             0.0263    0.0169        1.56 1.19e- 1
##  3 sib_sp          0.338     0.127         2.67 7.52e- 3
##  4 parch           0.177     0.148         1.19 2.33e- 1
##  5 fare            0.0116    0.00694       1.67 9.44e- 2
##  6 pclass_X2       1.63      0.394         4.13 3.61e- 5
##  7 pclass_X3       2.75      0.397         6.92 4.55e-12
##  8 sex_male        0.922     0.540         1.71 8.81e- 2
##  9 sex_male_x_age  0.0597    0.0191        3.12 1.83e- 3
## 10 age_x_fare     -0.000364  0.000189     -1.93 5.39e- 2
```

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.

```r
fit_test <- predict(log_fit1, titanic_test) %>%
  bind_cols(predict(log_fit1, titanic_test, type = "prob")) %>%
  bind_cols(titanic_test %>% select(survived))
```

```
fit_test %>%
  accuracy(truth = survived, estimate = .pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.843
```

The model's testing accuracy is close to the average accuracy across folds.