Homework 4

PSTAT 131/231

Contents

Resampling

For this assignment, we will continue working with part of a Kaggle data set that was the subject of a machine learning competition and is often used for practicing ML models. The goal is classification; specifically, to predict which passengers would survive the Titanic shipwreck.

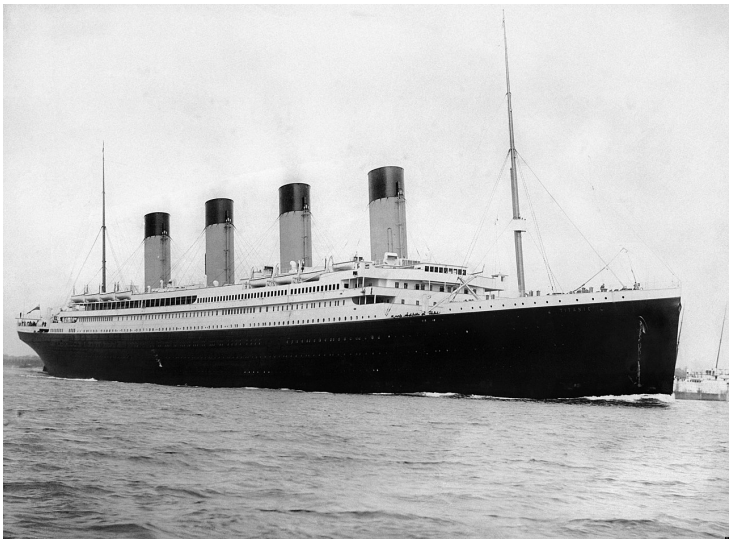

Figure 1: Fig. 1: RMS Titanic departing Southampton on April 10, 1912.

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`!

```
library(tidyverse)
library(tidymodels)
library(ggplot2)
library(corrplot)
```

```
library(discrim)
library(klaR)
library(caret)
library(pROC)
tidymodels_prefer()
set.seed(0213)

ti = read.csv(file = "data/titanic.csv") %>% mutate(survived = factor(survived, levels = c("Yes", "No")))
```

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.

```
ti_split = ti %>%
 initial_split(strata = survived, prop = 0.7)
ti_train = training(ti_split)
ti_test = testing(ti_split)
dim(ti_train) #623 observations is around 0.7 of the data set.
```

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

```
dim(ti_test)
```

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

Question 2

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

```
ti_folds = vfold_cv(ti_train, strata = survived,
 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?

Answer k-fold cross-validation is a method that divide data to k folds of individual mini testing set. When we apply the individual testing sets, other data are used for training. We should use this since we sometimes have observations in one class much higher or lower than other classes, and we use cross validation methods to estimate the test error associated with a model to evaluate performance. When we used the entire training set, it is a 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.

```
ti_recipe = recipe(survived ~ pclass + sex + age + sib_sp + parch + fare, ti_train) %>%
  step_impute_linear(age, impute_with = imp_vars(sib_sp)) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(~ starts_with("sex"):age + age:fare)

log_reg = logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

log_wflow = workflow() %>%
  add_model(log_reg) %>%
  add_recipe(ti_recipe)

ldisc = discrim_linear() %>%
  set_mode("classification") %>%
  set_engine("MASS")

ldisc_wflow = workflow() %>%
  add_model(ldisc) %>%
  add_recipe(ti_recipe)

quad = discrim_quad() %>%
 set_mode("classification") %>%
 set_engine("MASS")

quad_wflow = workflow() %>%
 add_model(quad) %>%
 add_recipe(ti_recipe)
```

10 folds for each type of model, which brings a total of 30 models ### Question 5

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

IMPORTANT: *Some models may take a while to run – anywhere from 3 to 10 minutes. You should NOT re-run these models each time you knit. Instead, run them once, using an R script, and store your results; look into the use of loading and saving. You should still include the code to run them when you knit, but set* `eval = FALSE` *in the code chunks.*

```
log_fit = fit_resamples(resamples = ti_folds,log_wflow, control = control_resamples(save_pred = TRUE))
ldisc_fit = fit_resamples(resamples = ti_folds,ldisc_wflow, control = control_resamples(save_pred = TRU
quad_fit = fit_resamples(resamples = ti_folds, quad_wflow, control = control_resamples(save_pred = TRUE
```

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_fit)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.792    10  0.0232 Preprocessor1_Model1
## 2 roc_auc  binary     0.848    10  0.0217 Preprocessor1_Model1
```

```
collect_metrics(ldisc_fit)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.789    10  0.0228 Preprocessor1_Model1
## 2 roc_auc  binary     0.846    10  0.0216 Preprocessor1_Model1
```

```
collect_metrics(quad_fit)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.769    10  0.0159 Preprocessor1_Model1
## 2 roc_auc  binary     0.820    10  0.0219 Preprocessor1_Model1
```

logistic regression model performed the best since it has the highest mean accuracy and is less than one standard error away from linear discriminant analysis model

Question 7

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

```
log_fit_a = fit(log_wflow, ti_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.

```
log_test_a = fit(log_wflow, ti_test)
predict = predict(log_test_a, new_data = ti_test, type = "class") %>%
  bind_cols(ti_test %>% select(survived)) %>%
  accuracy(truth = survived, estimate = .pred_class)
predict
```

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

Required for 231 Students

Consider the following intercept-only model, with $\epsilon \sim N(0, \sigma^2)$:

$$Y = \beta + \epsilon$$

where β is the parameter that we want to estimate. Suppose that we have n observations of the response, i.e. $y_1, ..., y_n$, with uncorrelated errors.

Question 9

Derive the least-squares estimate of β.

Question 10

Suppose that we perform leave-one-out cross-validation (LOOCV). Recall that, in LOOCV, we divide the data into n folds. What is the covariance between $\hat{\beta}^{(1)}$, or the least-squares estimator of β that we obtain by taking the first fold as a training set, and $\hat{\beta}^{(2)}$, the least-squares estimator of β that we obtain by taking the second fold as a training set?