Finn Stack 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.

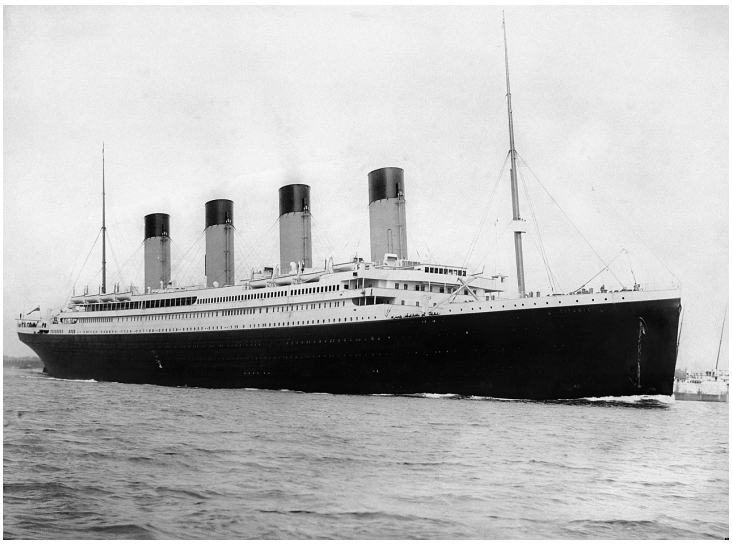

Figure 1: Fig. 1: RMS Titanic departing Southampton on April 10, 1912.

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`!

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. *Importing packages from HW #3*

```
library(klaR) # for naive bayes
library(tidyverse)
library(tidymodels)
library(corrplot)
library(discrim)
library(poissonreg)
library(corrr)
tidymodels_prefer()
```

Importing the data:

```
titanic <- read.csv(file = "/Users/finnianstack/Desktop/titanic.csv") %>%
  mutate(survived = factor(survived, levels = c("Yes", "No")),
         pclass = factor(pclass))
```

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.

```
set.seed(1234)

titanic_split <- initial_split(titanic, prop = 0.80, strata = survived)

titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
```

Creating a recipe based off of the previous homework.

```
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)
```

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?

Simply put, K-fold cross-validation is used to estimate the performance of the model on new data. We use this to estimate how accuract our model is on data not used in the training model. 'K' represents the number of groups the data should be split into. In our case, the number of groups is 10.

We use this method of cross-validation because it ensures a less-biased model. This occurs because every observation from the original dataset has equal chance of appearing in the training and test set. (cited: https://machinelearningmastery.com/k-fold-cross-validation/)

If we use the entire training set, we would be using the training-test-validation 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.

Since there are 10 folds for each of the three models, there are 30 models total.

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(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.)*

```
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     0.805    10 0.0105  Preprocessor1_Model1
## 2 roc_auc  binary     0.853    10 0.00885 Preprocessor1_Model1
```

```
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     0.801    10 0.00924 Preprocessor1_Model1
## 2 roc_auc  binary     0.852    10 0.00882 Preprocessor1_Model1
```

```
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     0.784    10  0.0147 Preprocessor1_Model1
## 2 roc_auc  binary     0.849    10  0.0101 Preprocessor1_Model1
```

The fitted model that performed the best was the log fit because it has the highest accuracy.

Question 7

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

```
titanic_log_entire <- 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.

```
log_fit_test <- fit(titanic_log_entire, titanic_test)

predict(log_fit_test, new_data = titanic_test, type = "class") %>%
  bind_cols(titanic_test %>% select(survived)) %>%
  accuracy(truth = survived, estimate = .pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.804
```

We can see from this test that our model's testing accuracy is about 80%. This is very good considering the average accuracy accross the 10-fold models was 79.6%. Overall, this is a pretty good model.