Homework 4

PSTAT 131/231

Contents

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

```
## [1] 891  12
```

```
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)
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 the process of splitting the the observations into k folds with one set being the validation set and the other sets used to fit the data. This process is repeated k times with a different fold as the validation set each time. K-fold cross validation allows us to maximize our data in order to estimate how well a model will perform. If we were to use the entire training set, a smaller amount of data is used to fit the model and we're prone to a highly variable validation test error rate. This process is known as 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.

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.

```
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(~ starts_with("sex"):age + age:fare)



log_reg <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")
log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(titanic_recipe)



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

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



qda_mod <- discrim_quad() %>%
  set_mode("classification") %>%
```

```
  set_engine("MASS")

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

There are 10 folds for each type of model, and therefore, there will be 30 models fit.

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_wkflow_rs <-
  log_wkflow %>%
  fit_resamples(titanic_folds)

lda_wkflow_rs <-
  lda_wkflow %>%
  fit_resamples(titanic_folds)

qda_wkflow_rs <-
  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_wkflow_rs)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.820    10 0.0117  Preprocessor1_Model1
## 2 roc_auc  binary     0.867    10 0.00963 Preprocessor1_Model1
```

```
collect_metrics(lda_wkflow_rs)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.810    10 0.0123  Preprocessor1_Model1
## 2 roc_auc  binary     0.867    10 0.00987 Preprocessor1_Model1
```

```
collect_metrics(qda_wkflow_rs)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.800    10  0.0155 Preprocessor1_Model1
## 2 roc_auc  binary     0.855    10  0.0135 Preprocessor1_Model1
```

The logistic model was the best performing by a slight margin since it had a mean accuracy of around 82% and a standard error of .012, both of which performed better than the other models.

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)
log_fit
```

```
## == Workflow [trained] ==========================================================
## Preprocessor: Recipe
## Model: logistic_reg()
##
## -- Preprocessor ----------------------------------------------------------------
## 3 Recipe Steps
##
## * step_impute_linear()
## * step_dummy()
## * step_interact()
##
## -- Model -----------------------------------------------------------------------
##
## Call:  stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)
##
## Coefficients:
##    (Intercept)             age          sib_sp           parch            fare
##     -3.9477224       0.0189737       0.3594724       0.1890924       0.0114843
##      pclass_X2       pclass_X3        sex_male  sex_male_x_age      age_x_fare
##      1.6336088       2.7957394       0.8918174       0.0735529      -0.0004366
##
## Degrees of Freedom: 711 Total (i.e. Null);  702 Residual
## Null Deviance:       948
## Residual Deviance: 589.1     AIC: 609.1
```

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

The model accuracy is around 82.1% which is slightly higher than the average across all folds, which varied from 80% to 82%. This means the folded models performed relatively well, very close to our model's testing accuracy.