Homework 4

PSTAT 131 John Wei

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.

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(tidymodels)
library(tidyverse)
library(ISLR)
library(ISLR2)
library(discrim)
library(poissonreg)
library(corrr)
library(corrplot)
library(klaR)
library(pROC)
library(tinytex)
set.seed(4167)
```

```
titanic <- read_csv("titanic.csv")
```

```
titanic
```

```
## # A tibble: 891 x 12
##    passenger_id survived pclass name       sex     age sib_sp parch ticket  fare
##           <dbl> <chr>     <dbl> <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$survived <- factor(titanic$survived)
titanic$pclass <- factor(titanic$pclass)
titanic <- titanic %>% arrange(desc(survived))
```

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

```
## [1] 712  12 179  12
```

891 total observations, split 80-20.

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?

In question 2 the training set is divided into 10 groups of similar size. This lets us measure model performance without needing to predict the entire training set. If the entire training set was used we would then be using bootstrapping.

Question 4

Set up workflows for 3 models:

```
titanic_recipe <- recipe(survived ~ pclass + sex + age +
                           sib_sp + parch + fare, 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)
```

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.

10 folds for each model; 30 total models will be fitted to data.

Question 5

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

```
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.

```
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     0.802    10  0.0160 Preprocessor1_Model1
## 2 roc_auc  binary     0.856    10  0.0160 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     0.796    10  0.0206 Preprocessor1_Model1
## 2 roc_auc  binary     0.852    10  0.0169 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     0.777    10  0.0151 Preprocessor1_Model1
## 2 roc_auc  binary     0.833    10  0.0193 Preprocessor1_Model1
```

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.)*

Log model performed best as it had the highest mean accuracy and closest to lowest standard error.

Question 7

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

```
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)     3.52      0.595        5.91  3.38e- 9
##  2 age            -0.0150    0.0145      -1.04  3.00e- 1
##  3 sib_sp         -0.334     0.122       -2.73  6.27e- 3
##  4 parch          -0.121     0.126       -0.956 3.39e- 1
##  5 fare           -0.0113    0.00618     -1.83  6.79e- 2
##  6 pclass_X2      -1.30      0.355       -3.66  2.54e- 4
##  7 pclass_X3      -2.42      0.347       -6.98  2.90e-12
##  8 sex_male       -1.13      0.509       -2.22  2.66e- 2
##  9 sex_male_x_age -0.0597    0.0173      -3.46  5.48e- 4
## 10 age_x_fare      0.000393  0.000172     2.28  2.26e- 2
```

Question 8

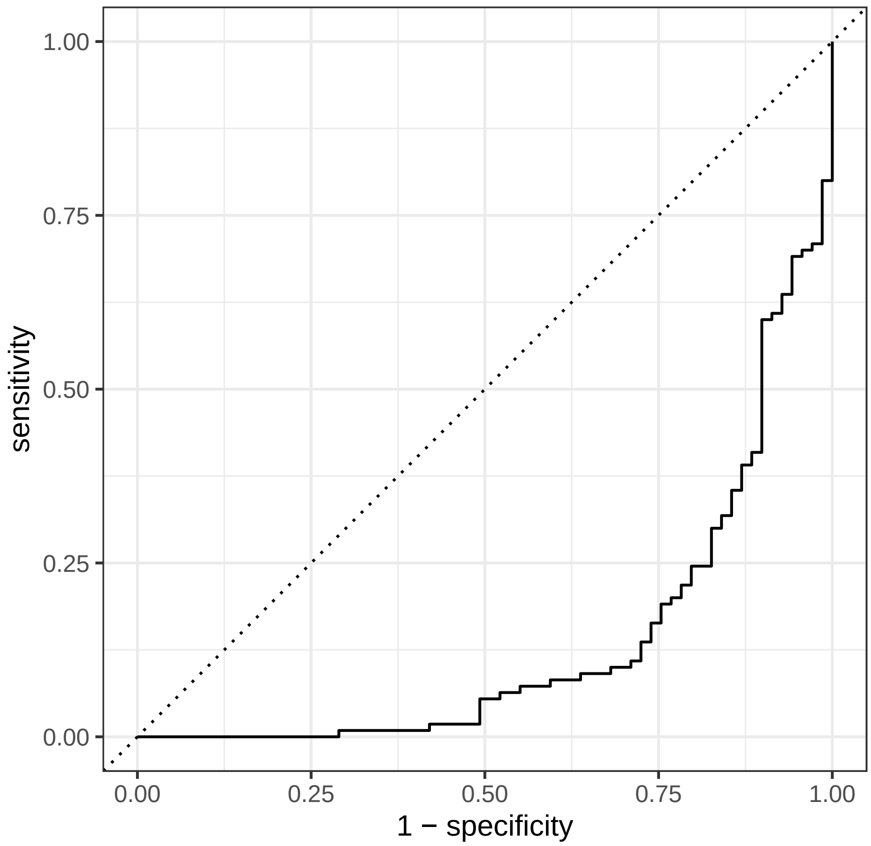
Finally, with your fitted model, use `predict()`, `bind_cols()`, and `accuracy()` to assess your model's performance on the testing data!

Compare your model's testing accuracy to its average accuracy across folds. Describe what you see.

```
predict(log_fit1, new_data = titanic_test, type = "class") %>%
  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         0.799
```

```
augment(log_fit1, new_data = titanic_test) %>%
  roc_curve(survived, .pred_Yes) %>%
  autoplot()
```



The model did well; the testing accuracy was similar to the average accuracy (higher than the other models) and the area under the curve is large.