131-hw-3

Tonia Wu

4/14/2022

Q1

```
# set seed
set.seed(266)

# changing survived and pclass to afctors
titanic$survived = factor(titanic$survived, levels = c("Yes","No"))
titanic$pclass = factor(titanic$pclass)
titanic$sex = factor(titanic$sex)


# split data, stratifying on survived
titanic_split <- initial_split(titanic, prop = 0.80, strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
```

Our training set has 712 observations:

```
dim(titanic_train)
```

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

Our test dataset has 179 observations:

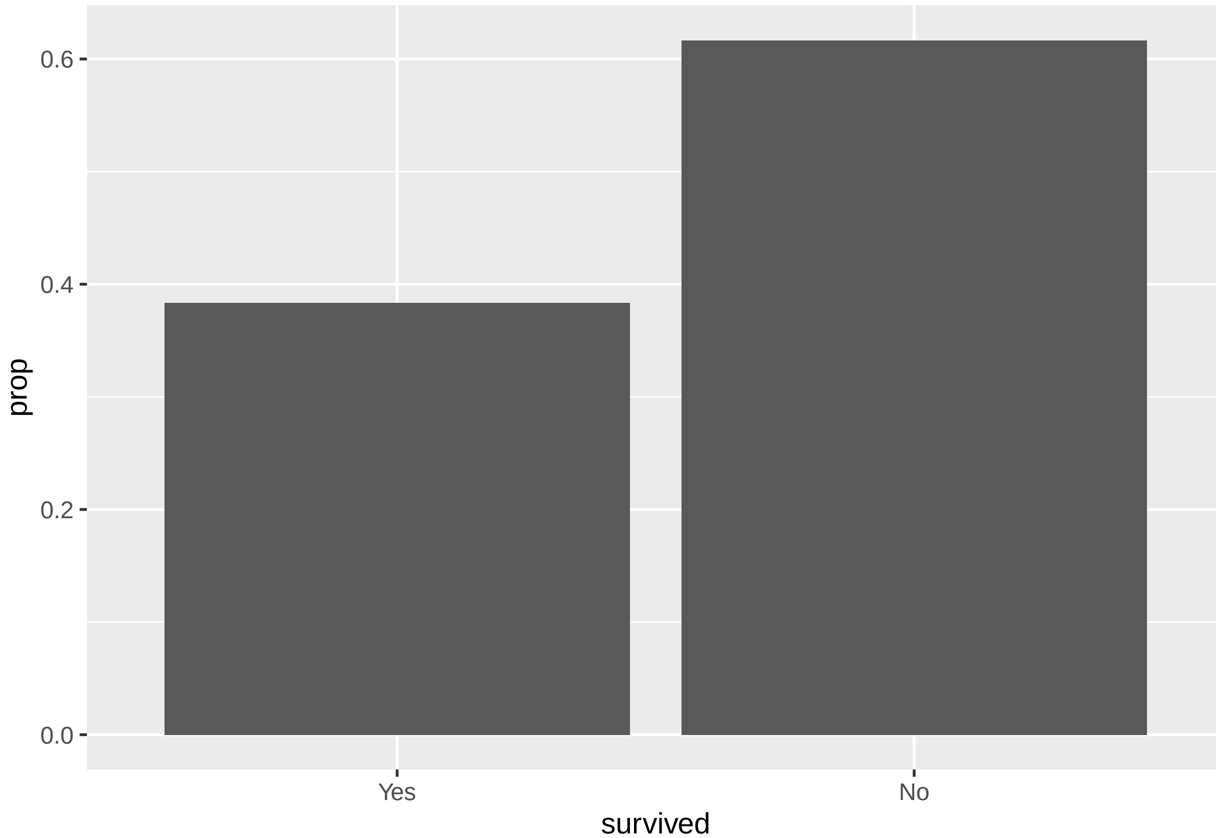
```
# get test dimensions
dim(titanic_test)
```

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

Q2

Roughly 60% of passengers did not survivee.

```
# plot of how many survived
titanic_train %>%
  ggplot(aes(x = survived)) +
  geom_bar(aes(y = ..prop.., group = 1))
```



Q3

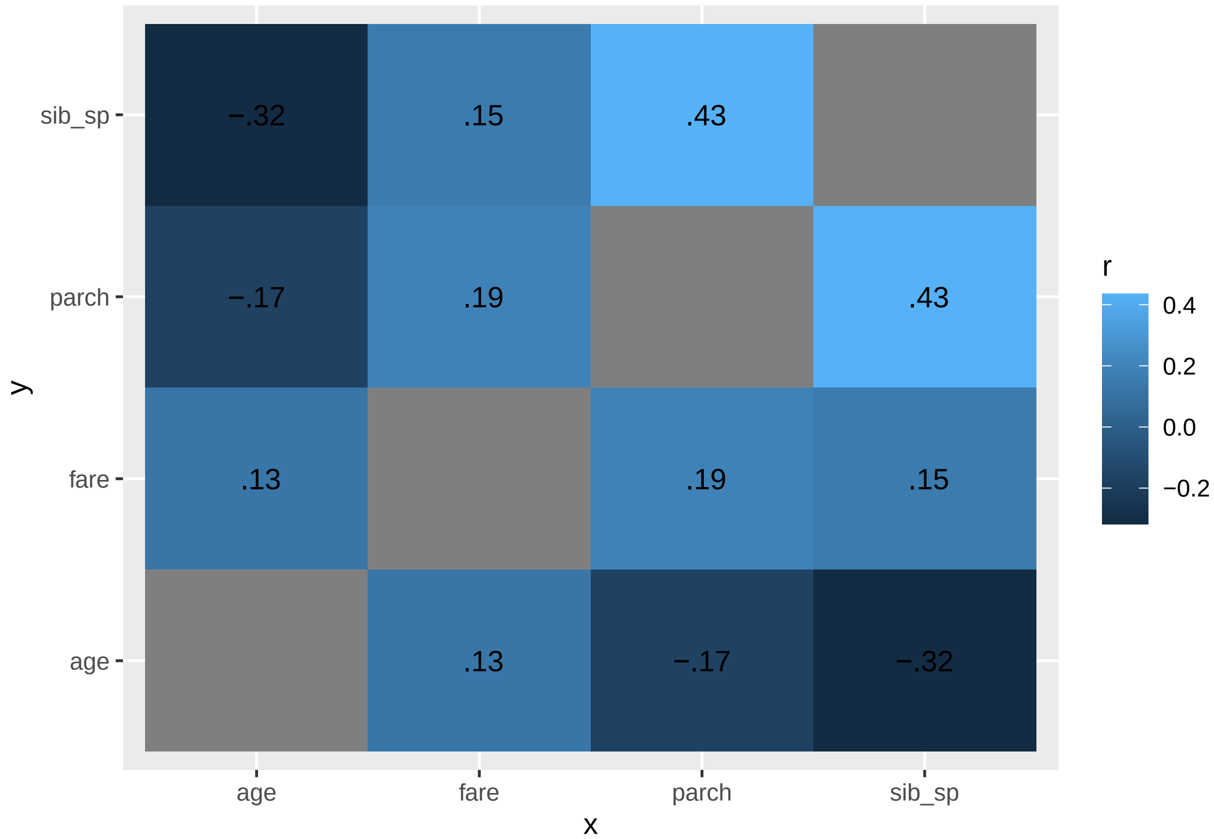
At most, there are weak to low correlations both positive and negative between the variables. Passenger id was removed because, despite being numeric, it is not meaningfully quantitative.

The strongest two correlations are 1) a low positive correlation between the number of siblings/spouses and the number of parents/children aboard and 2) a low negative correlation between number of siblings/spouses and age.

```
# calculate corr matrix
cor_plot <- titanic_train %>%
  dplyr::select(-c(survived, pclass, name, sex, cabin, ticket, embarked, passenger_id)) %>%
  correlate(use = "pairwise.complete.obs", method = "pearson")
```

```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```

```
# visualize
cor_plot %>%
  stretch() %>%
  ggplot(aes(x, y, fill = r)) +
  geom_tile() +
  geom_text(aes(label = as.character(fashion(r))))
```



Q4

```
# creating a recipe predicting survived using training data
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 = ~ sex_male : fare) %>%
  step_interact(terms = ~ age : fare)

titanic_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          6
##
## Operations:
##
## Linear regression imputation for age
## Dummy variables from all_nominal_predictors()
```

```
## Interactions with sex_male:fare
## Interactions with age:fare
```

Q5

```r
# specify model type and engine
log_reg <- logistic_reg() %>%
  set_engine('glm') %>%
  set_mode('classification')

# set up workflow
log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(titanic_recipe)

# fit model to training data
log_fit <- fit(log_wkflow, titanic_train)
```

Q6

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

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

# fit lda
lda_fit <- fit(lda_wkflow, titanic_train)
```

Q7

```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(qda_wkflow, titanic_train)
```

Q8

```r
nb_mod <- naive_Bayes() %>%
  set_mode("classification") %>%
  set_engine("klaR") %>%
  set_args(usekernel = FALSE)

nb_wkflow <- workflow() %>%
  add_model(nb_mod) %>%
  add_recipe(titanic_recipe)

nb_fit <- fit(nb_wkflow, titanic_train)
```

Q9

The log prediction performed the best on the training data.

```r
pred_models = bind_cols(predict(log_fit, titanic_test),
                    predict(lda_fit, titanic_test),
                    predict(qda_fit, titanic_test),
                    predict(nb_fit, titanic_test))
colnames(pred_models) = c("log_pred","lda_pred","qda_pred", "nb_fit")
head(pred_models)
```

```
## # A tibble: 6 x 4
##   log_pred lda_pred qda_pred nb_fit
##   <fct>    <fct>    <fct>    <fct>
## 1 Yes      Yes      No       No
## 2 Yes      Yes      Yes      Yes
## 3 No       No       No       No
## 4 Yes      Yes      Yes      Yes
## 5 No       No       No       No
## 6 No       No       No       No
```

```r
# calculate accuracies
log_reg_acc <- augment(log_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
lda_acc <- augment(lda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
qda_acc <- augment(qda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
nb_acc <- augment(nb_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)

log_reg_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.819
```

```
lda_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.805
```

```
qda_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.794
```

```
nb_acc
```

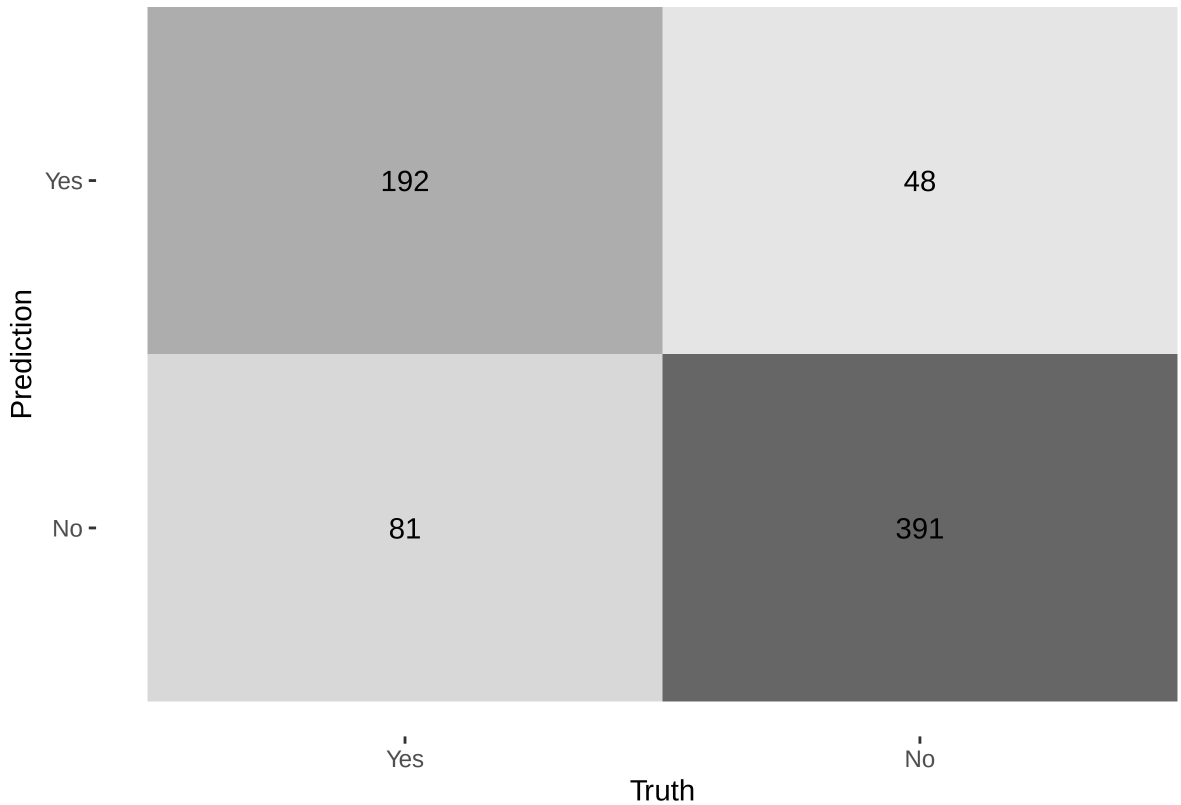
```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.770
```

Q10

```
predict(log_fit, titanic_test)
```

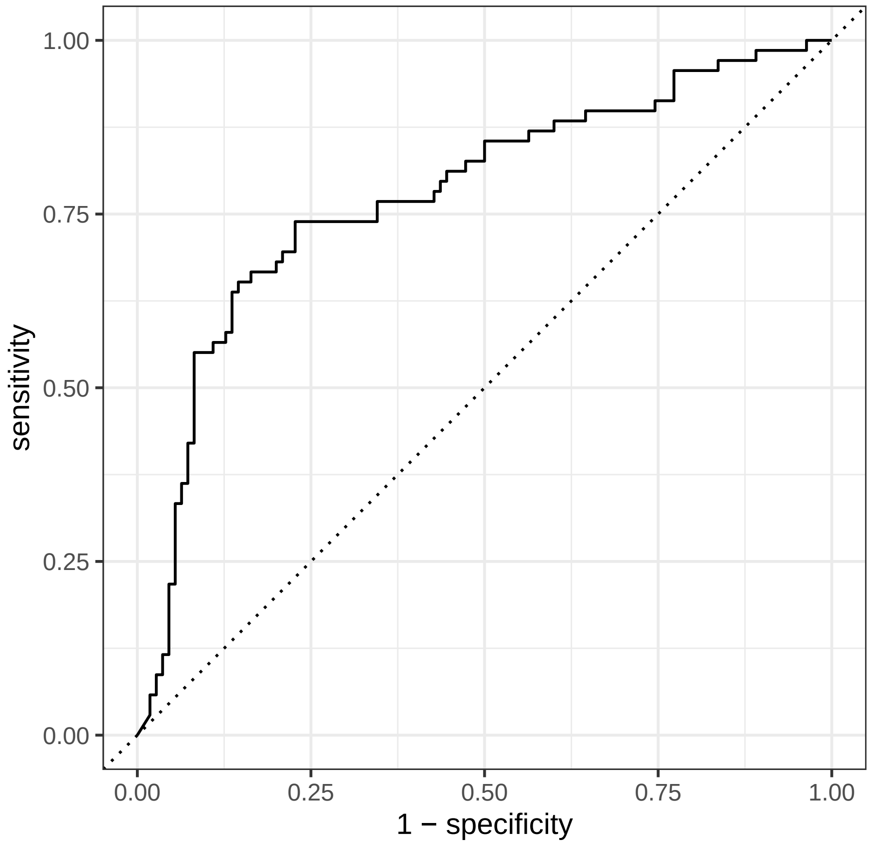
```
## # A tibble: 179 x 1
##    .pred_class
##    <fct>
##  1 Yes
##  2 Yes
##  3 No
##  4 Yes
##  5 No
##  6 No
##  7 No
##  8 No
##  9 No
## 10 Yes
## # ... with 169 more rows
```

```
# visual representation of confusion matrix
augment(log_fit, new_data = titanic_train) %>%
  conf_mat(truth = survived, estimate = .pred_class) %>%
  autoplot(type = "heatmap")
```



ROC plot:

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



The AUC is 0.7781:

```
augment(nb_fit, new_data = titanic_test) %>%
  roc(survived, .pred_Yes)
```

```
##
## Call:
## roc.data.frame(data = ., response = survived, predictor = .pred_Yes)
##
## Data: .pred_Yes in 69 controls (survived Yes) > 110 cases (survived No).
## Area under the curve: 0.7781
```