Homework 3

PSTAT 131/231

Classification

For this assignment, we will be 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.

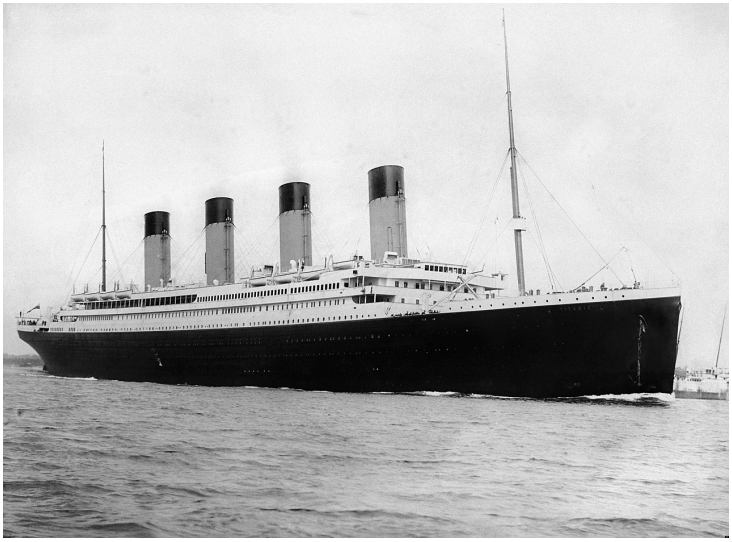


Figure 1: Fig. 1: RMS Titanic departing Southampton on April 10, 1912.

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.

```
library(tidymodels)
library(tidyverse)
library(discrim)
library(klaR)
library(corrr)
titanic <- read.csv("data/titanic.csv")
titanic <- titanic %>%
  mutate(pclass = factor(pclass), survived = factor(survived, levels = c("Yes", "No")))
```

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. Take a look at the training data and note any potential issues, such as missing data.

Why is it a good idea to use stratified sampling for this data?

```
set.seed(2022)

titanic_split <- initial_split(titanic, prop = 0.70,
                               strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)

nrow(titanic_train)
```

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

```
nrow(titanic_test)
```

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

```
head(titanic_train)
```

```
##    passenger_id survived pclass                           name  sex age sib_sp
## 1             1       No      3        Braund, Mr. Owen Harris male  22      1
## 6             6       No      3               Moran, Mr. James male  NA      0
## 7             7       No      1        McCarthy, Mr. Timothy J male  54      0
## 8             8       No      3 Palsson, Master. Gosta Leonard male   2      3
## 13           13       No      3 Saundercock, Mr. William Henry male  20      0
## 14           14       No      3    Andersson, Mr. Anders Johan male  39      1
##    parch    ticket    fare cabin embarked
## 1      0 A/5 21171  7.2500  <NA>        S
## 6      0    330877  8.4583  <NA>        Q
## 7      0     17463 51.8625   E46        S
## 8      1    349909 21.0750  <NA>        S
## 13     0 A/5. 2151  8.0500  <NA>        S
## 14     5    347082 31.2750  <NA>        S
```

```
sum(is.na(titanic_train))
```

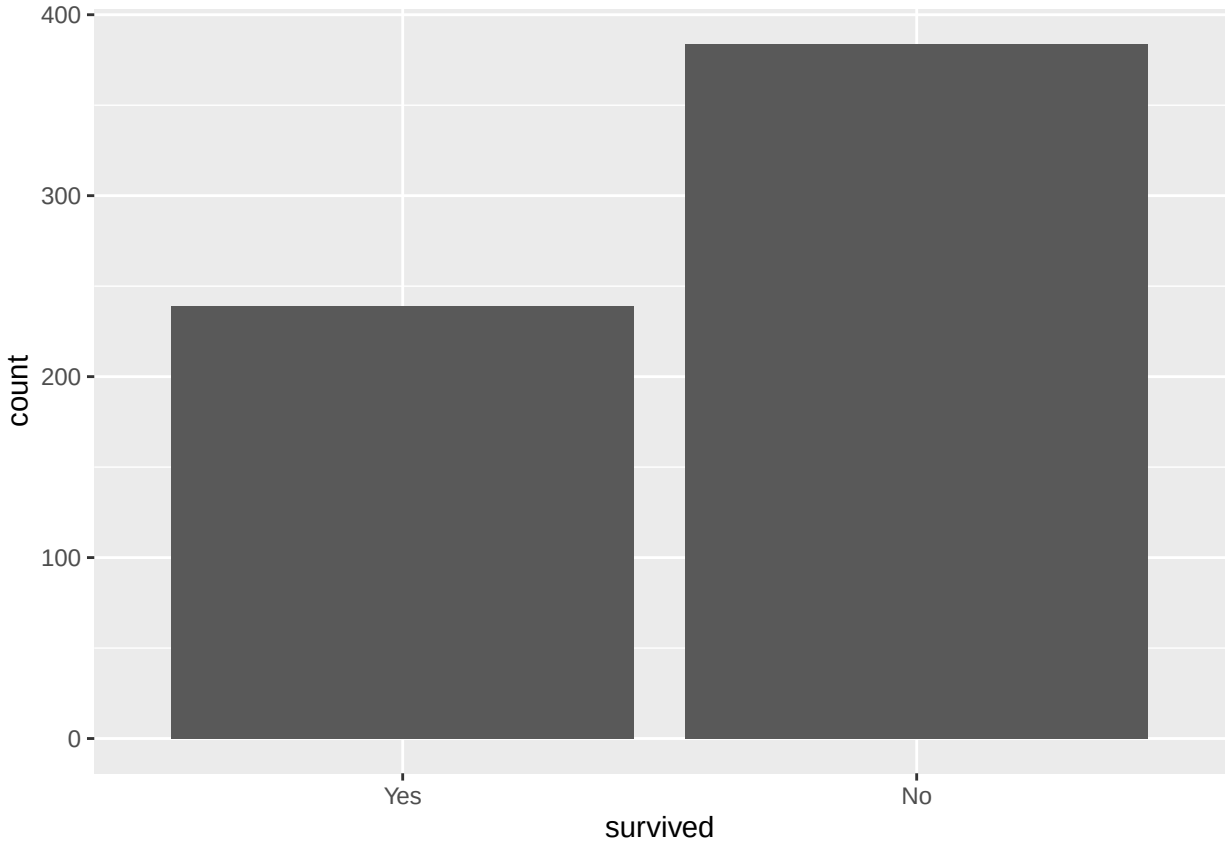
```
## [1] 617
```

Solution: There are lots of missing data for age. It is a good idea to use stratified sampling for this data because the proportion of survival and died are different.

Question 2

Using the **training** data set, explore/describe the distribution of the outcome variable `survived`.

```
titanic_train %>%
  ggplot(aes(x = survived)) +
  geom_bar()
```



Solution: The number of non-survived is larger than the number of survived.

Question 3

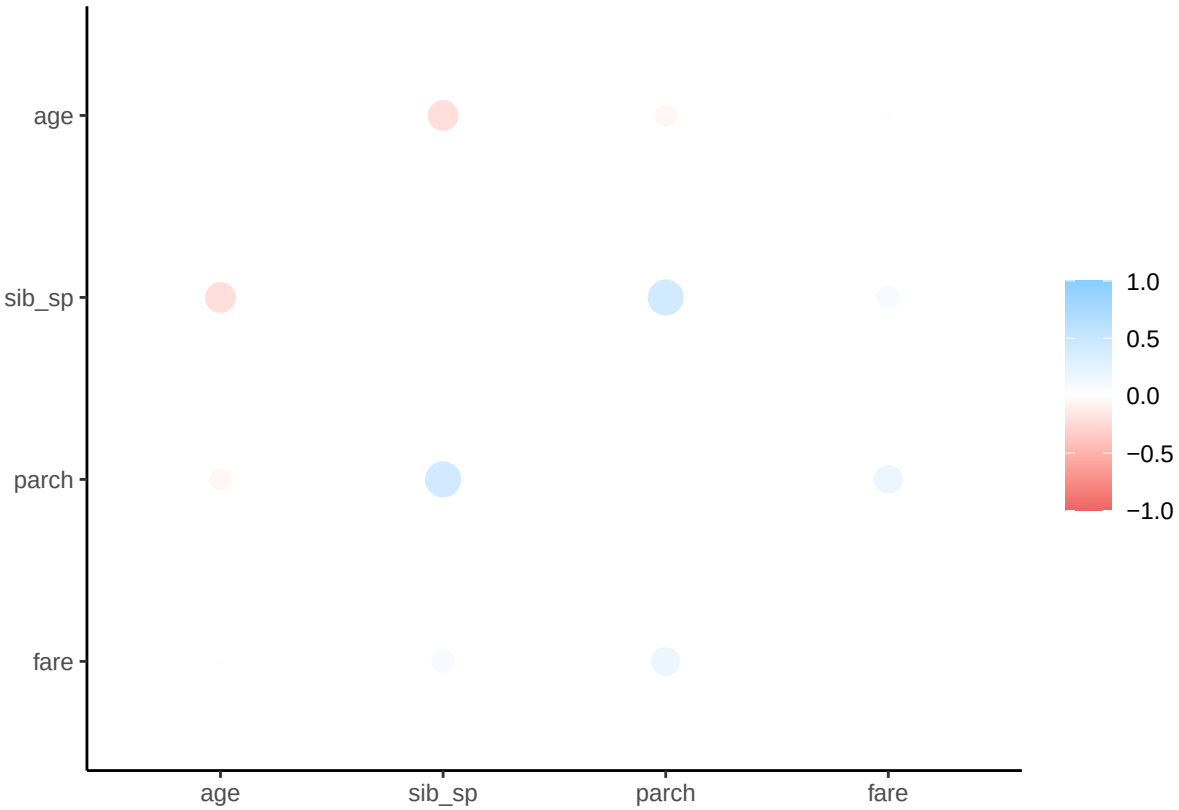
Using the **training** data set, create a correlation matrix of all continuous variables. Create a visualization of the matrix, and describe any patterns you see. Are any predictors correlated with each other? Which ones, and in which direction?

```
cor_titanic_train <- titanic_train %>%
  dplyr::select(c(age, sib_sp, parch, fare)) %>%
  correlate()


cor_titanic_train
```

```
## # A tibble: 4 x 5
##   term      age sib_sp  parch   fare
##   <chr>   <dbl>  <dbl>  <dbl>  <dbl>
## 1 age    NA     -0.309 -0.176  0.101
## 2 sib_sp -0.309 NA      0.414  0.196
## 3 parch  -0.176  0.414 NA      0.275
## 4 fare    0.101  0.196  0.275 NA
```

```
rplot(cor_titanic_train)
```



Solution: There is no obvious patterns and there are no predictors cooelated with each other.

Question 4

Using the **training** data, create a recipe predicting the outcome variable `survived`. Include the following predictors: ticket class, sex, age, number of siblings or spouses aboard, number of parents or children aboard, and passenger fare.

Recall that there were missing values for `age`. To deal with this, add an imputation step using `step_impute_linear()`. Next, use `step_dummy()` to **dummy** encode categorical predictors. Finally, include interactions between:

- Sex and passenger fare, and
- Age and passenger fare.

You'll need to investigate the `tidymodels` documentation to find the appropriate step functions to use.

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

Question 5

Specify a **logistic regression** model for classification using the `"glm"` engine. Then create a workflow. Add your model and the appropriate recipe. Finally, use `fit()` to apply your workflow to the **training** data.

Hint: Make sure to store the results of `fit()`. You'll need them later on.

```
log_reg <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

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

log_fit <- fit(log_wkflow, titanic_train)
```

Question 6

Repeat Question 5, but this time specify a linear discriminant analysis model for classification using the `"MASS"` engine.

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

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

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

Question 7

Repeat Question 5, but this time specify a quadratic discriminant analysis model for classification using the `"MASS"` engine.

```
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)
```

Question 8

Repeat Question 5, but this time specify a naive Bayes model for classification using the `"klaR"` engine. Set the `usekernel` argument to `FALSE`.

```
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)
```

Question 9

Now you've fit four different models to your training data.

Use `predict()` and `bind_cols()` to generate predictions using each of these 4 models and your **training** data. Then use the *accuracy* metric to assess the performance of each of the four models.

Which model achieved the highest accuracy on the training data?

```
log_pre <- predict(log_fit, new_data = titanic_train, type = "prob")
lda_pre <- predict(lda_fit, new_data = titanic_train, type = "prob")
qda_pre <- predict(qda_fit, new_data = titanic_train, type = "prob")
nb_pre <- predict(nb_fit, new_data = titanic_train, type = "prob")

pre <- bind_cols(log_pre, lda_pre, qda_pre, nb_pre)
```

```
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)

accuracies <- c(log_reg_acc$.estimate, lda_acc$.estimate,
                nb_acc$.estimate, qda_acc$.estimate)
models <- c("Logistic Regression", "LDA", "Naive Bayes", "QDA")
results <- tibble(accuracies = accuracies, models = models)
results %>%
  arrange(-accuracies)
```

```
## # A tibble: 4 x 2
##   accuracies models
##        <dbl> <chr>
## 1      0.823 Logistic Regression
## 2      0.803 LDA
## 3      0.798 QDA
## 4      0.764 Naive Bayes
```

Solution: The logistic regression model achieved the highest accuracy on the training data.

Question 10

Fit the model with the highest training accuracy to the **testing** data. Report the accuracy of the model on the **testing** data.

Again using the **testing** data, create a confusion matrix and visualize it. Plot an ROC curve and calculate the area under it (AUC).

How did the model perform? Compare its training and testing accuracies. If the values differ, why do you think this is so?

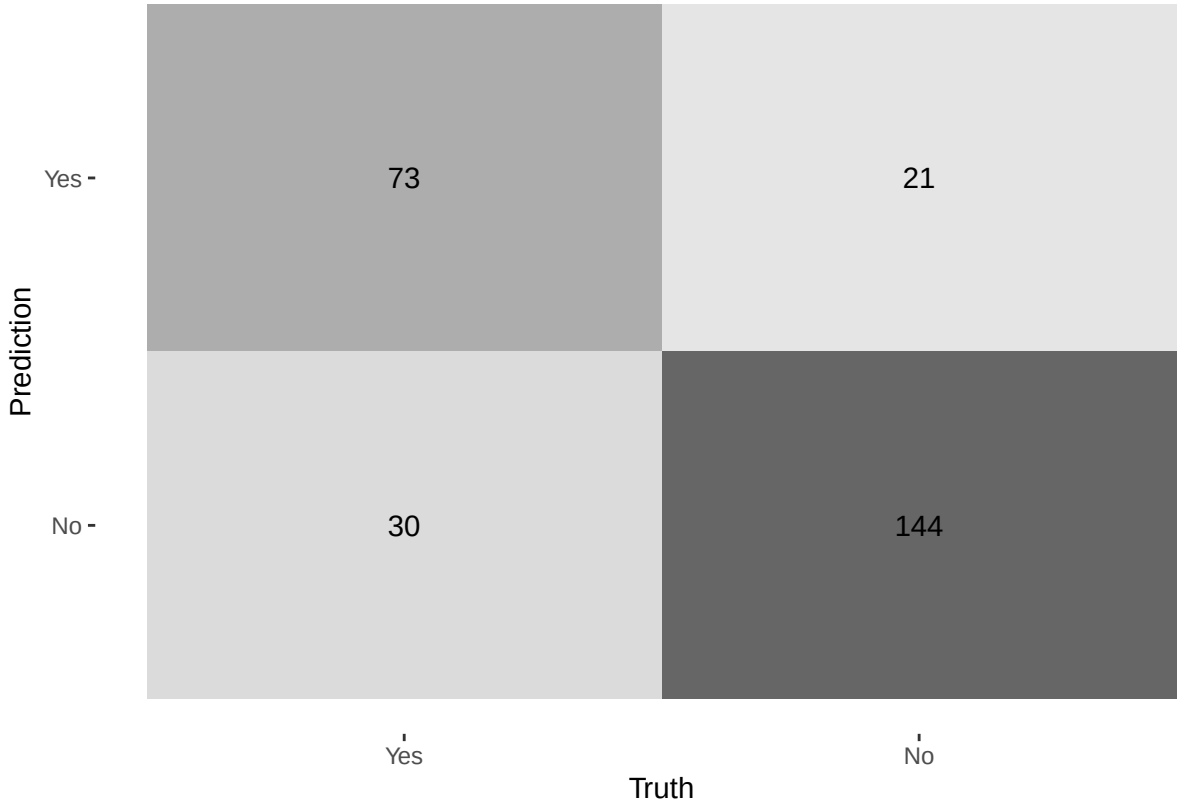
```
augment(log_fit, new_data = titanic_test) %>%
  accuracy(truth = survived, estimate = .pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.810
```

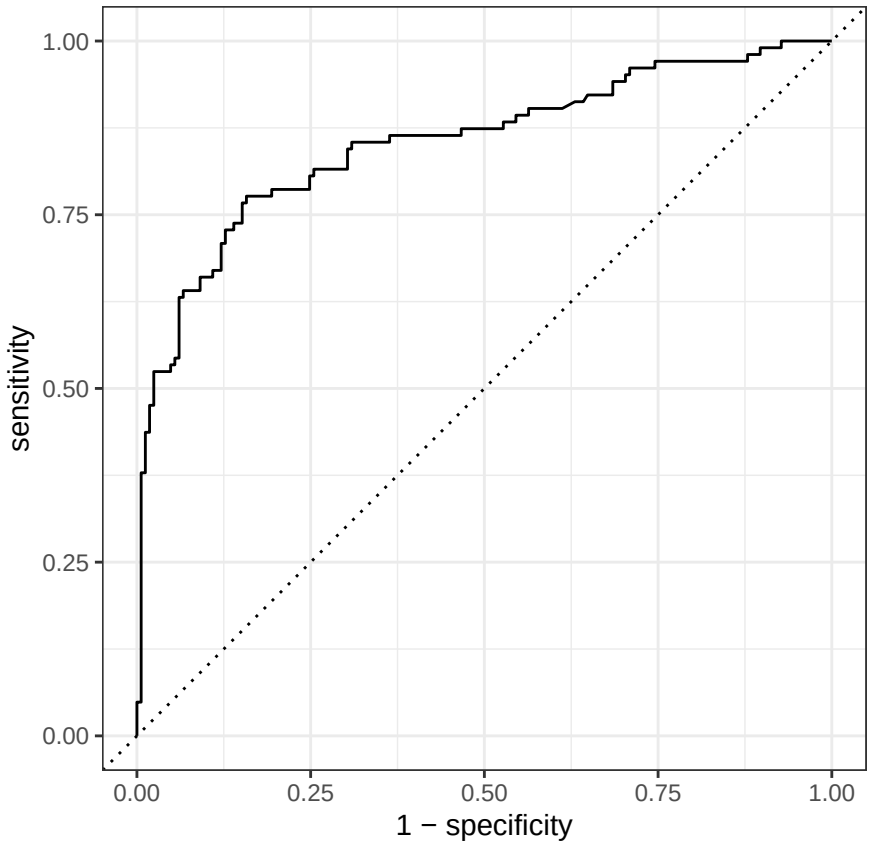
```
augment(log_fit, new_data = titanic_test) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction Yes  No
##        Yes  73  21
##        No   30 144
```

```
augment(log_fit, new_data = titanic_test) %>%
  conf_mat(truth = survived, estimate = .pred_class) %>%
  autoplot(type = "heatmap")
```




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




```
augment(log_fit, new_data = titanic_test) %>%
  roc_auc(survived, .pred_Yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.853
```

Solution: The accuracy of the model on the testing data is 0.8097. AUC is 0.8534. The model performs well, the training and testing accuracies are almost same, which means we reasonablely fit the classificaion model.