# Homework 3

PSTAT 131/231

## Contents



### 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.

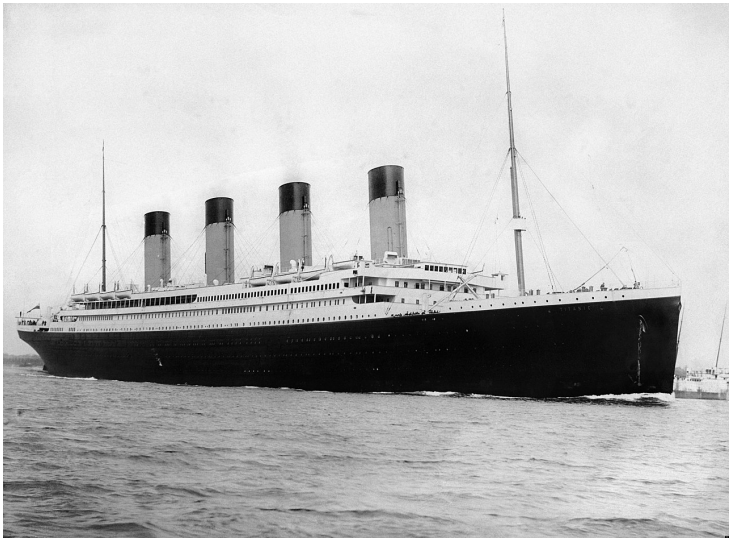


Figure 1: Fig. 1: RMS Titanic departing Southampton on April 10, 1912.

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`!

```
library(tidyverse)
library(tidymodels)
library(ggplot2)
library(corrplot)
library(discrim)
library(klaR)
```

```
library(caret)
library(pROC)
tidymodels_prefer()
set.seed(0213)

ti = read.csv(file = "data/titanic.csv") %>% mutate(survived = factor(survived, levels = c("Yes", "No"))
```

*Remember that you'll need to set a seed at the beginning of the document to reproduce your results.*

### 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?

```
ti_split = ti %>%
 initial_split(strata = survived, prop = 0.7)
ti_train = training(ti_split)
ti_test = testing(ti_split)
dim(ti_train) #623 observations is around 0.7 of the data set.
```

```
## [1] 623  12
```

```
dim(ti_test)
```

```
## [1] 268  12
```

```
ti_train %>% summary()
```

```
##   passenger_id   survived  pclass      name              sex
##  Min.   :  1.0   Yes:239   1:154   Length:623         Length:623
##  1st Qu.:232.5   No :384   2:126   Class :character   Class :character
##  Median :442.0             3:343   Mode  :character   Mode  :character
##  Mean   :449.9
##  3rd Qu.:671.5
##  Max.   :890.0
##
##       age            sib_sp           parch           ticket
##  Min.   : 0.42   Min.   :0.0000   Min.   :0.0000   Length:623
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :29.00   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :30.72   Mean   :0.4815   Mean   :0.3323
##  3rd Qu.:39.25   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.0000   Max.   :5.0000
##  NA's   :127
##       fare            cabin             embarked
##  Min.   :  0.000   Length:623         Length:623
##  1st Qu.:  7.896   Class :character   Class :character
##  Median : 13.792   Mode  :character   Mode  :character
```

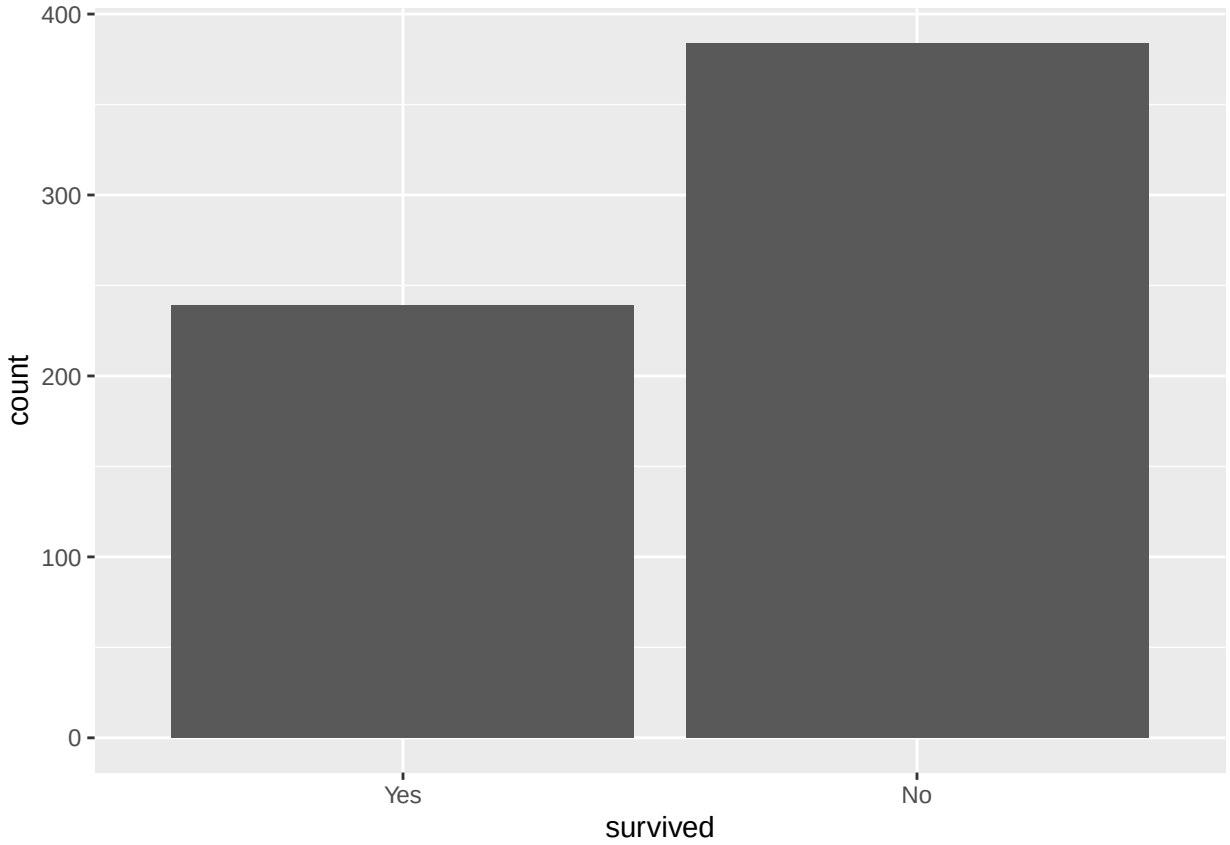
```
##  Mean   : 30.921
##  3rd Qu.: 30.598
##  Max.   :512.329
##
```

There are 114 missing datas in age

**Question 2**

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

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

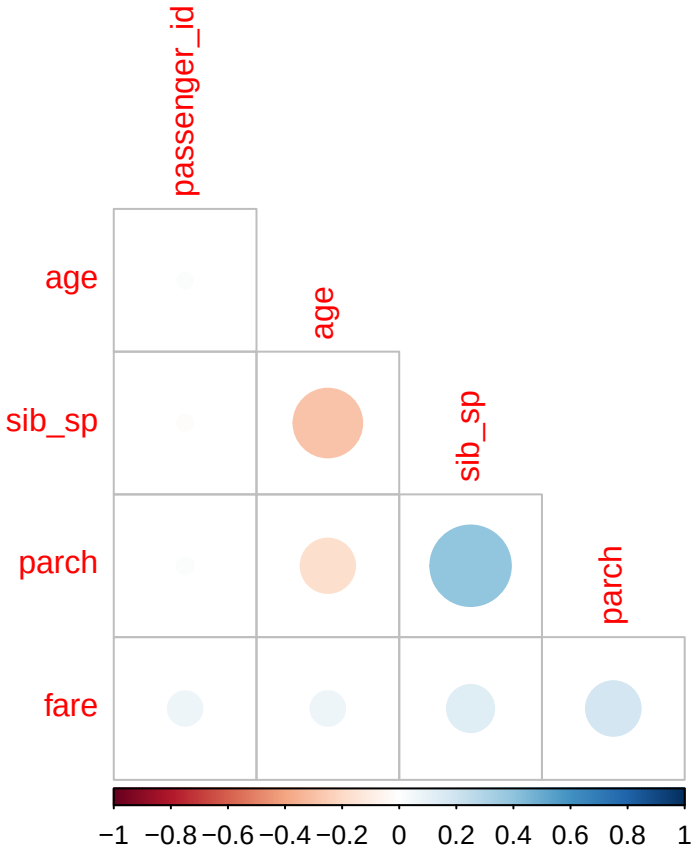



They are not very balanced

**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?

```
ti_train %>%
  select(is.numeric) %>%
  cor(use = "complete.obs") %>%
  corrplot(type = "lower", diag = FALSE)
```




Passemger_id should be ignored as it is not a variable we are analyzing on. age and sib_sp are negatively correlated - elders are less likely to have siblings on board. son_sp and parch are positively correlated - having siblings on board are more likely to have parents/ children on board, vice versa.

### 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.

```
ti_recipe = recipe(survived ~ pclass + sex + age + sib_sp + parch + fare, ti_train) %>%
  step_impute_linear(age, impute_with = imp_vars(sib_sp)) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(~ starts_with("sex"):age + 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_wflow = workflow() %>%
  add_model(log_reg) %>%
  add_recipe(ti_recipe)

log_fit = fit(log_wflow, ti_train)
```

### Question 6

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

```
ldisc = discrim_linear() %>%
  set_mode("classification") %>%
  set_engine("MASS")
ldisc_wflow = workflow() %>%
  add_model(ldisc) %>%
  add_recipe(ti_recipe)
ldisc_fit = fit(ldisc_wflow, ti_train)
```

### Question 7

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

```
quad = discrim_quad() %>%
 set_mode("classification") %>%
 set_engine("MASS")
quad_wflow = workflow() %>%
 add_model(quad) %>%
 add_recipe(ti_recipe)
quad_fit = fit(quad_wflow, ti_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 = naive_Bayes() %>%
 set_mode("classification") %>%
 set_engine("klaR") %>%
 set_args(usekernel = FALSE)
nb_wflow = workflow() %>%
 add_model(nb) %>%
 add_recipe(ti_recipe)
nb_fit = fit(nb_wflow, ti_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_perf = predict(log_fit, new_data = ti_train, type = "class") %>%
  bind_cols(ti_train %>% select(survived)) %>%
  accuracy(truth = survived, estimate = .pred_class)

ldisc_perf = predict(ldisc_fit, new_data = ti_train, type = "class") %>%
  bind_cols(ti_train %>% select(survived)) %>%
  accuracy(truth = survived, estimate = .pred_class)

quad_perf = predict(quad_fit, new_data = ti_train, type = "class") %>%
  bind_cols(ti_train %>% select(survived)) %>%
  accuracy(truth = survived, estimate = .pred_class)

nb_perf = predict(nb_fit, new_data = ti_train, type = "class") %>%
  bind_cols(ti_train %>% select(survived)) %>%
  accuracy(truth = survived, estimate = .pred_class)

results = bind_rows(log_perf, ldisc_perf, quad_perf, nb_perf) %>%
  tibble() %>%
  mutate(model = c("Logistic", "Linear Discriminant Analysis", "Quadratic Discriminant Analysis", "Naiv
  select(model, .estimate)

results
```

```
## # A tibble: 4 x 2
##   model                          .estimate
##   <chr>                              <dbl>
## 1 Logistic                           0.803
## 2 Linear Discriminant Analysis       0.793
## 3 Quadratic Discriminant Analysis    0.782
## 4 Naive Bayes                        0.782
```

Logistic regression model performed the best

**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?

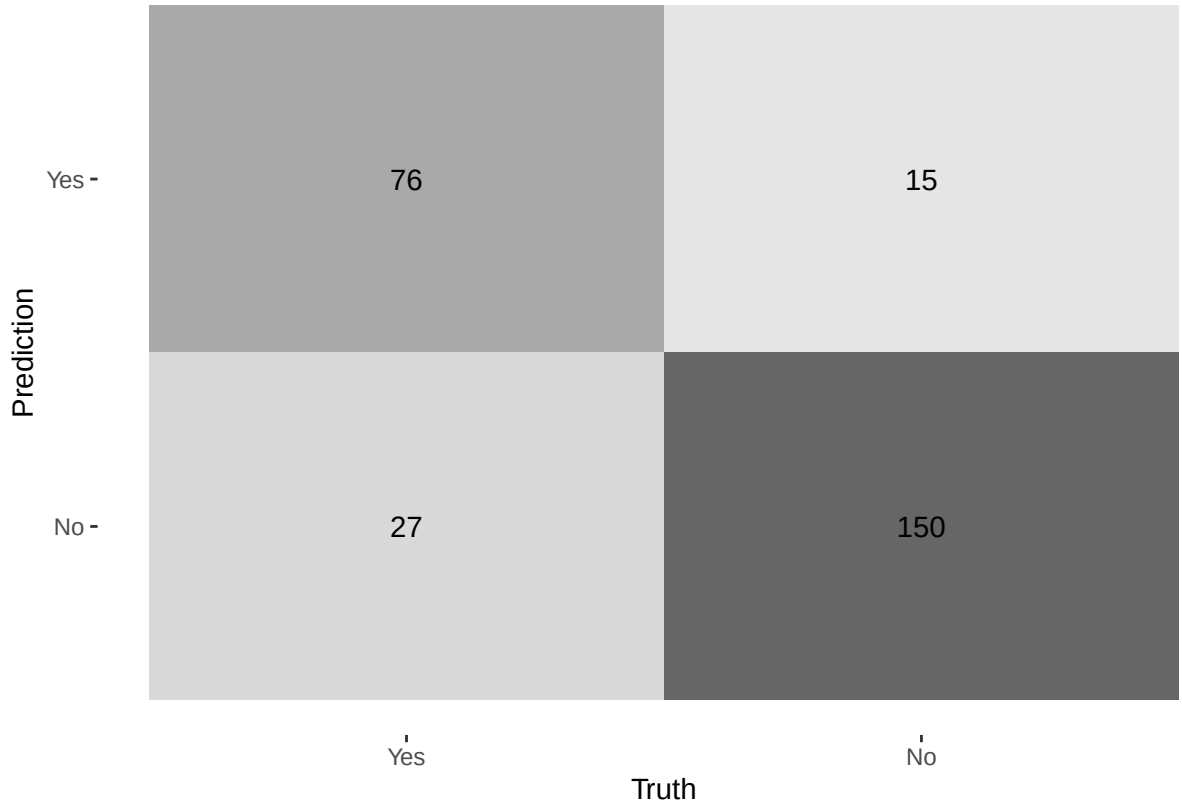
```
log_test <- fit(log_wflow, ti_test)

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

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.843
```

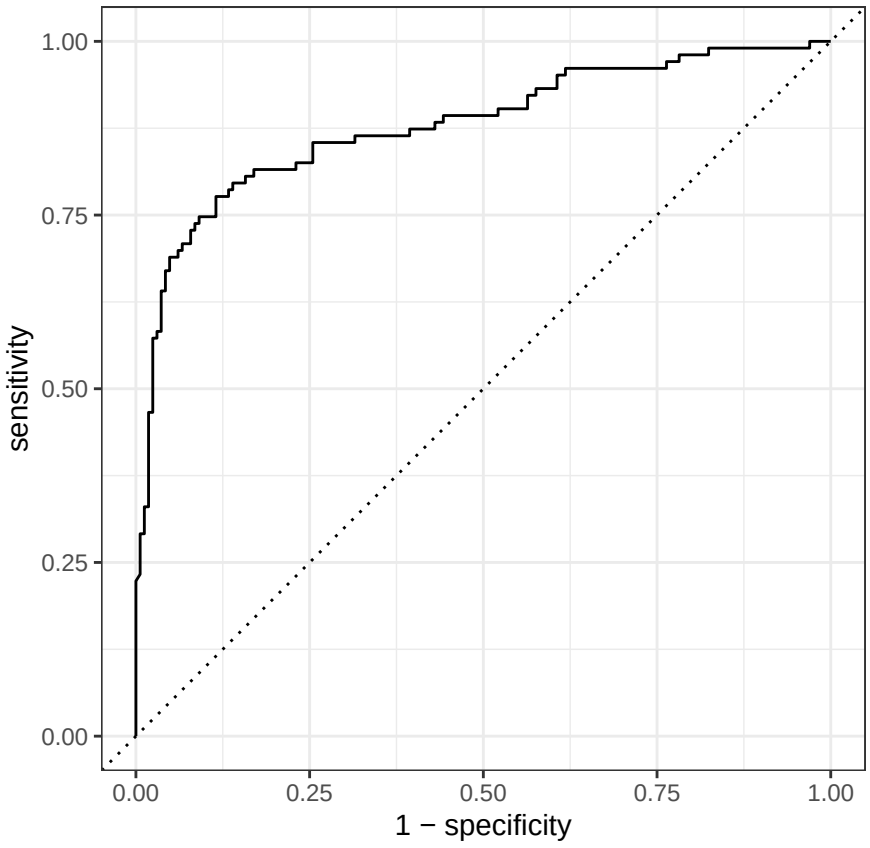
Confusion matrix:

```
log_results <- augment(log_test, new_data = ti_test)
log_results %>%
  conf_mat(truth = survived, estimate = .pred_class) %>%
  autoplot(type = "heatmap")
```




ROC:

```
log_results %>%
  roc_curve(survived, .pred_Yes) %>%
  autoplot()
```




Area under the ROC curve

```
log_results %>%
  roc_auc(survived, .pred_Yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.879
```

**Required for 231 Students**

In a binary classification problem, let $p$ represent the probability of class label 1, which implies that $1 - p$ represents the probability of class label 0. The *logistic function* (also called the "inverse logit") is the cumulative distribution function of the logistic distribution, which maps a real number $z$ to the open interval $(0, 1)$.

**Question 11**

Given that:

$$p(z) = \frac{e^z}{1 + e^z}$$

Prove that the inverse of a logistic function is indeed the *logit* function:

$$z(p) = ln\left(\frac{p}{1-p}\right)$$

**Question 12**

Assume that $z = \beta_0 + \beta_1 x_1$ and $p = logistic(z)$. How do the odds of the outcome change if you increase $x_1$ by two? Demonstrate this.

Assume now that $\beta_1$ is negative. What value does $p$ approach as $x_1$ approaches $\infty$? What value does $p$ approach as $x_1$ approaches $-\infty$?