Classification

Homework 3

Code

PSTAT 131/231

Classification

For this assignment, we will be working with part of a Kaggle data set (https://www.kaggle.com/c/titanic/overview) 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 (https://en.wikipedia.org/wiki/Titanic).

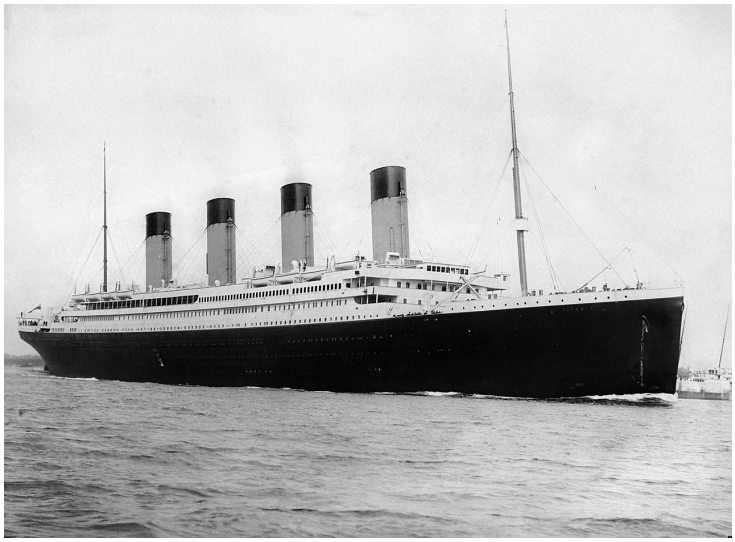


Fig. 1: RMS Titanic departing Southampton on April 10, 1912.

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.

Hide

```
library(tidyverse)
library(tidymodels)
library(corrr)

library(ISLR) # For the Smarket data set
library(ISLR2) # For the Bikeshare data set
library(discrim)
library(poissonreg)

library(klaR) # for naive bayes
tidymodels_prefer()
library(pROC)
```

Hide

```
titanic <- read.csv('titanic.csv')
head(titanic)
```

```
##   passenger_id survived pclass
## 1            1       No      3
## 2            2      Yes      1
## 3            3      Yes      3
## 4            4      Yes      1
## 5            5       No      3
## 6            6       No      3
##                                                 name    sex age sib_sp parch
## 1                             Braund, Mr. Owen Harris   male  22      1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38      1     0
## 3                              Heikkinen, Miss. Laina female  26      0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35      1     0
## 5                            Allen, Mr. William Henry   male  35      0     0
## 6                                    Moran, Mr. James   male  NA      0     0
##             ticket    fare cabin embarked
## 1        A/5 21171  7.2500  <NA>        S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250  <NA>        S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500  <NA>        S
## 6           330877  8.4583  <NA>        Q
```

Hide

```
titanic$pclass <- factor(titanic$pclass)
titanic$survived <- factor(titanic$survived, ordered=TRUE, levels=c('Yes','No'))

head(titanic)
```

```
##   passenger_id survived pclass
## 1            1       No      3
## 2            2      Yes      1
## 3            3      Yes      3
## 4            4      Yes      1
## 5            5       No      3
## 6            6       No      3
##                                                 name    sex age sib_sp parch
## 1                             Braund, Mr. Owen Harris   male  22      1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38      1     0
## 3                              Heikkinen, Miss. Laina female  26      0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35      1     0
## 5                            Allen, Mr. William Henry   male  35      0     0
## 6                                    Moran, Mr. James   male  NA      0     0
##             ticket    fare cabin embarked
## 1        A/5 21171  7.2500  <NA>        S
## 2         PC 17599 71.2833   C85        C
## 3 STON/02. 3101282  7.9250  <NA>        S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500  <NA>        S
## 6           330877  8.4583  <NA>        Q
```

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?

Hide

```
set.seed(3435)

titanic_split <- initial_split(titanic, prop = 0.80,
                               strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
head(titanic_train)
```

```
##    passenger_id survived pclass                                     name    sex age
## 1             1       No      3                  Braund, Mr. Owen Harris   male  22
## 5             5       No      3                 Allen, Mr. William Henry   male  35
## 7             7       No      1                  McCarthy, Mr. Timothy J   male  54
## 8             8       No      3           Palsson, Master. Gosta Leonard   male   2
## 14           14       No      3              Andersson, Mr. Anders Johan   male  39
## 15           15       No      3 Vestrom, Miss. Hulda Amanda Adolfina female  14
##    sib_sp parch    ticket    fare cabin embarked
## 1       1     0 A/5 21171  7.2500  <NA>        S
## 5       0     0    373450  8.0500  <NA>        S
## 7       0     0     17463 51.8625   E46        S
## 8       3     1    349909 21.0750  <NA>        S
## 14      1     5    347082 31.2750  <NA>        S
## 15      0     0    350406  7.8542  <NA>        S
```

Hide

```
nrow(titanic)
```

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

Hide

```
nrow(titanic_train)
```

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

Hide

```
nrow(titanic_test)
```

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

There are missing values for age.

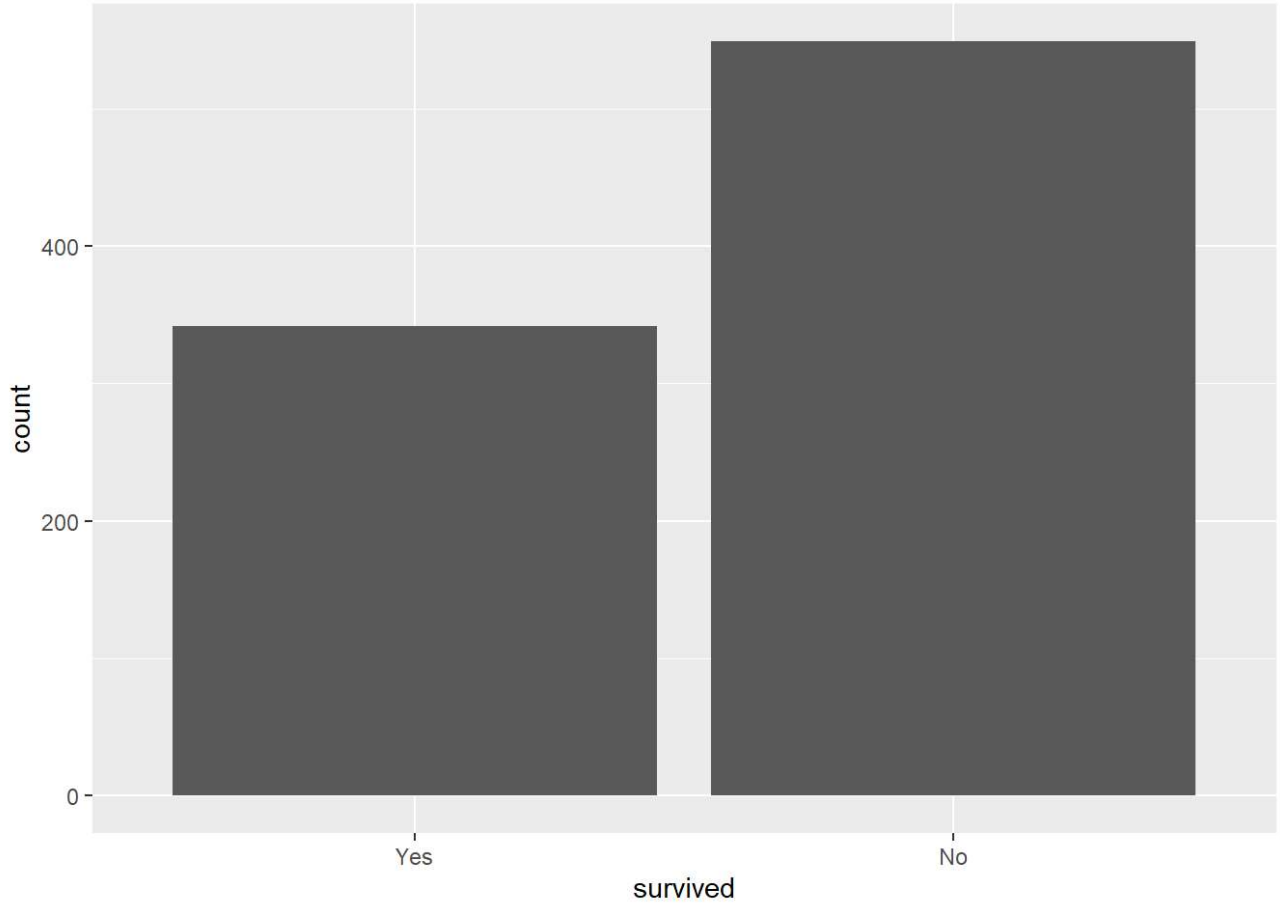
We could notice by the bar plot that the number of people survived is much smaller than those did not. Thus, stratified sampling could help avoid that the sample is mostly drew from group of “No.” Stratified sampling could present subgroups of people according to strata, making the sample representative as a whole.

Question 2

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

Hide

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



The number of people who did not survived is about 1.5 times of the number of people who survived.

Question 3

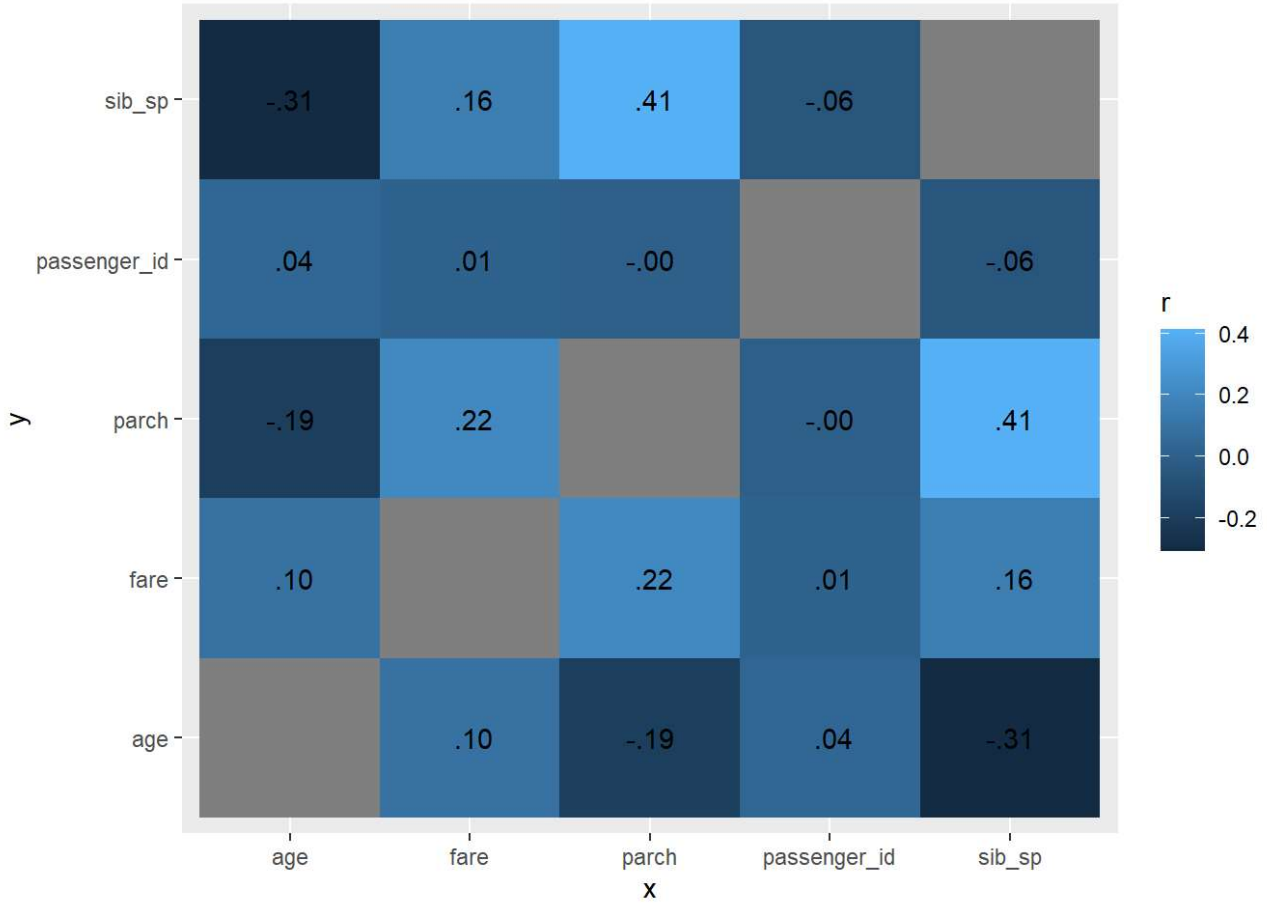
Using the **training** data set, create a correlation matrix of all continuous variables. Create a visualization of the matrix, and describe any patterns you see. Are any predictors correlated with each other? Which ones, and in which direction?

Hide

```
titanic2 = select_if(titanic,is.numeric)
cor_titanic <- titanic2 %>%
  correlate()
```

Hide

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



"sib_sp"(number of siblings / spouses aboard the Titanic) is positively correlated with "parch"(number of parents / children aboard the Titanic). "parch" has a positive correlation with "fare." "sib_sp" has positive correlation with "fare." "sib_sp" has a negative correlation with "age." "Parch" has a negative correlation with "age."

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.

Hide

```
titanic_recipe <- recipe(survived~pclass+sex+age+sib_sp+parch+fare,data=titanic_train) %
>%
  step_impute_linear(age) %>% step_dummy(all_nominal_predictors())

titanic_recipe
```

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

Hide

```
titanic_interact <- titanic_recipe %>% step_interact(terms = ~ starts_with('sex'):fare+age:fare)
titanic_interact %>% prep() %>% bake(titanic_train)
```

```
## # A tibble: 712 x 10
##      age sib_sp parch  fare survived pclass_X2 pclass_X3 sex_male
##    <dbl>  <int> <int> <dbl> <ord>        <dbl>     <dbl>    <dbl>
##  1    22      1     0  7.25 No               0         1        1
##  2    35      0     0  8.05 No               0         1        1
##  3    54      0     0 51.9  No               0         0        1
##  4     2      3     1 21.1  No               0         1        1
##  5    39      1     5 31.3  No               0         1        1
##  6    14      0     0  7.85 No               0         1        0
##  7     2      4     1 29.1  No               0         1        1
##  8    31      1     0 18    No               0         1        0
##  9    35      0     0 26    No               1         0        1
## 10     8      3     1 21.1  No               0         1        0
## # ... with 702 more rows, and 2 more variables: sex_male_x_fare <dbl>,
## #   fare_x_age <dbl>
```

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.

Hide

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

Hide

```
log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(titanic_interact)

log_fit <- fit(log_wkflow, titanic_train)

log_fit %>% tidy
```

```
## # A tibble: 10 x 5
##    term             estimate std.error statistic  p.value
##    <chr>               <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)     -4.34      0.651       -6.66  2.72e-11
##  2 age              0.0606    0.0128       4.75  1.99e- 6
##  3 sib_sp           0.436     0.129        3.37  7.57e- 4
##  4 parch            0.280     0.151        1.85  6.40e- 2
##  5 fare            -0.00639   0.0109      -0.587 5.57e- 1
##  6 pclass_X2        1.16      0.343        3.39  6.92e- 4
##  7 pclass_X3        2.33      0.361        6.45  1.15e-10
##  8 sex_male         2.37      0.297        8.00  1.29e-15
##  9 sex_male_x_fare  0.0136    0.00859      1.59  1.13e- 1
## 10 fare_x_age      -0.000281  0.000190    -1.48  1.39e- 1
```

Question 6

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

Hide

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

lda_wkflow <- workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(titanic_interact)
```

Hide

```
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.

Hide

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

qda_wkflow <- workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(titanic_interact)

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`.

Hide

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

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

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?

Hide

```
log_pre <-predict(log_fit, new_data = titanic_train, type = "class")
log_reg_acc <- augment(log_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         0.816
```

Hide

```
lda_pre <-predict(lda_fit, new_data = titanic_train, type = "class")
lda_acc <- augment(lda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
lda_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.806
```

Hide

```
qda_pre <- predict(qda_fit, new_data = titanic_train, type = "class")
qda_acc <- augment(qda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
qda_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.787
```

Hide

```
nb_pre <-predict(nb_fit, new_data = titanic_train, type = "class")
nb_acc <- augment(nb_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
nb_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.778
```

Hide

```
bind_cols(log_pre,lda_pre,qda_pre,nb_pre,titanic_train$survived)
```

```
## # A tibble: 712 x 5
##    .pred_class...1 .pred_class...2 .pred_class...3 .pred_class...4 ...5
##    <fct>           <fct>           <fct>           <fct>           <ord>
##  1 No              No              No              No              No
##  2 No              No              No              No              No
##  3 No              No              No              No              No
##  4 No              No              No              No              No
##  5 No              No              No              No              No
##  6 Yes             Yes             Yes             No              No
##  7 No              No              No              No              No
##  8 No              Yes             No              No              No
##  9 No              No              No              No              No
## 10 Yes             Yes             No              No              No
## # ... with 702 more rows
```

Hide

```
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      0.816 Logistic Regression
## 2      0.806 LDA
## 3      0.787 QDA
## 4      0.778 Naive Bayes
```

Logistic model has the highest accuracy.

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?

Hide

```
predict(log_fit, new_data = titanic_test, type = "class")
```

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

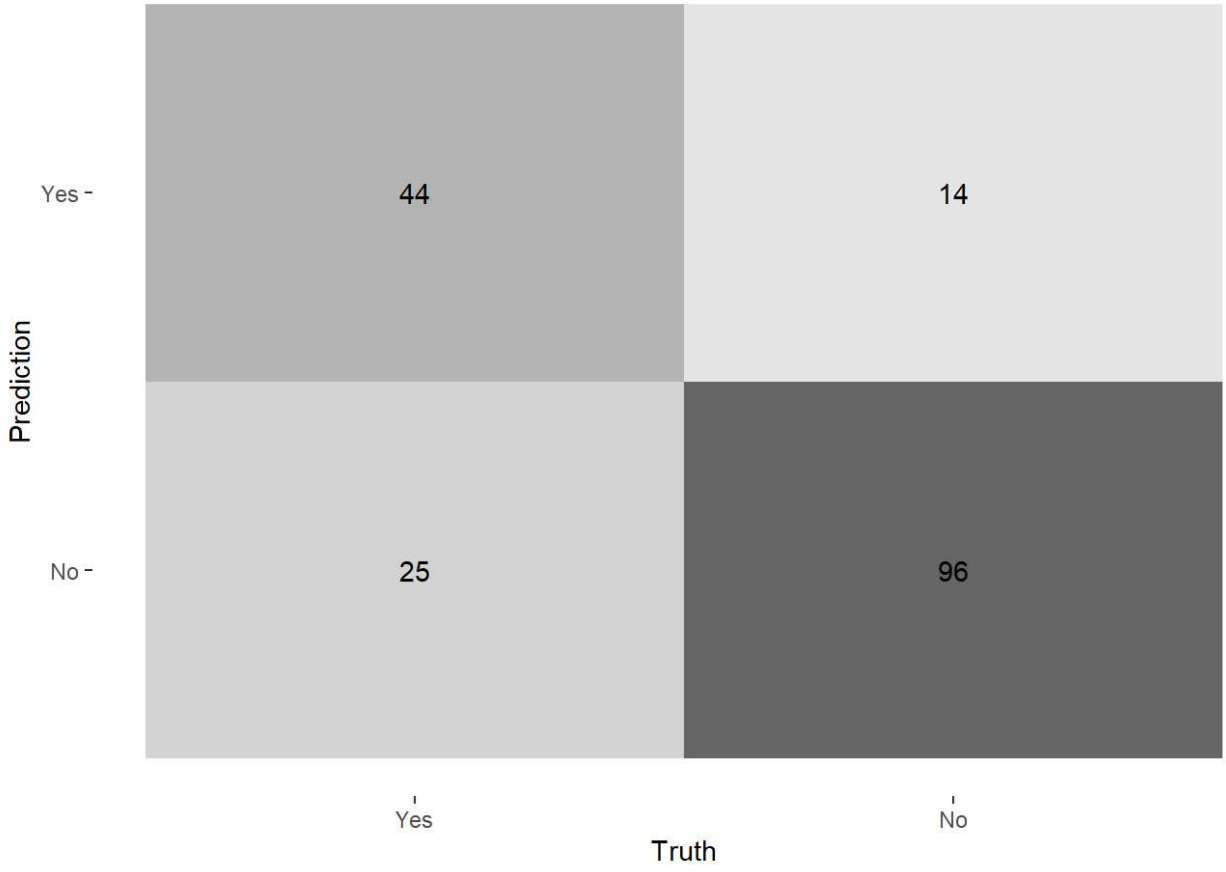
Hide

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

```
##           Truth
## Prediction Yes No
##        Yes  44 14
##        No   25 96
```

Hide

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



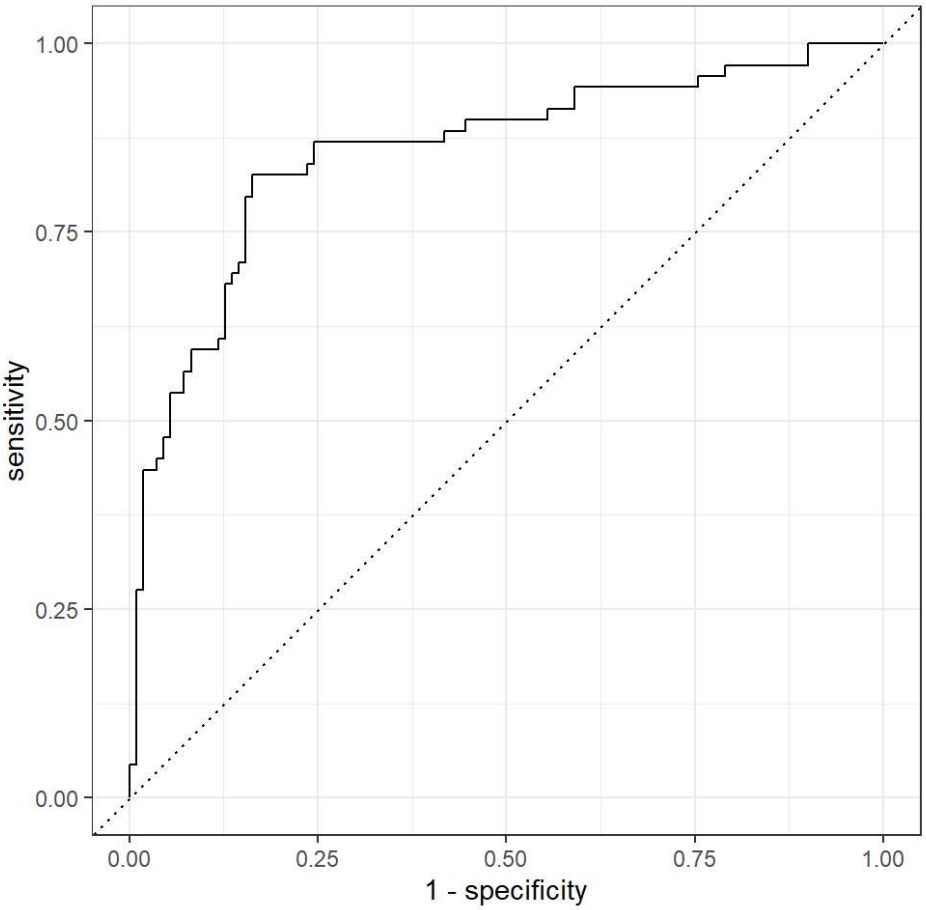
Hide

```
test_accu <- augment(log_fit, new_data = titanic_test) %>% accuracy(truth=survived,estima
te=.pred_class)
test_accu
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.782
```

Hide

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



Hide

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

```
## Area under the curve: 0.8556
```

The model performs relatively well. The area under ROC curve is 0.8556, which is close to 1. The training accuracy is 0.816 and the test accuracy is 0.782. The test accuracy is a little smaller than the training accuracy. It is reasonable since the model is fitted by the training data set, so it reflects the training data more precisely.

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:

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

Proof

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

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

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

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

$$e^z = \frac{p}{1-p}$$

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

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 β_1 is negative. What value does p approach as x_1 approaches ∞? What value does p approach as x_1 approaches $-\infty$?

$$z' = \beta_0 + \beta_1(x_1 + 2)$$

$$p = logstic(z)$$

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

$$\frac{p}{1-p} = e^z$$

$$\frac{p}{1-p} = e^{z'} = e^{\beta_0+\beta_1(x_1+2)} = e^{\beta_0+\beta_1 x_1}e^{2\beta_1}$$

The odds of the outcome increase $e^{2\beta_1}$ times when x_1 increase by two.

If β_1 is negative and x_1 approaches ∞, then $e^{-\infty} = 0$. Therefore, p approaches 0.
If β_1 is negative and x_1 approaches $-\infty$, then $e^{\infty} = \infty$. Both denominator and numerator approaches ∞. Therefore, p approaches 1.