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.

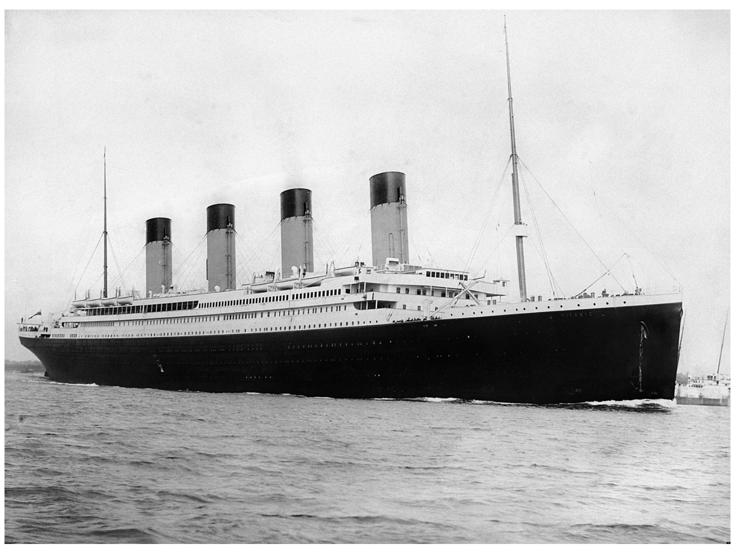


Figure 1: Fig. 1: RMS Titanic departing Southampton on April 10, 1912.

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`!

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

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

```
library(poissonreg)
library(corrr)
library(klaR) # for naive bayes
library(forcats)
library(corrplot)
library(pROC)
tidymodels_prefer()
```

```
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
```

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?

```
titanic$survived <- as.factor(titanic$survived)
titanic$survived <- ordered(titanic$survived, levels = c("Yes", "No"))
titanic$pclass <- as.factor(titanic$pclass)
```

```
set.seed(2022)
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 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
```

```
head(titanic_test)
```

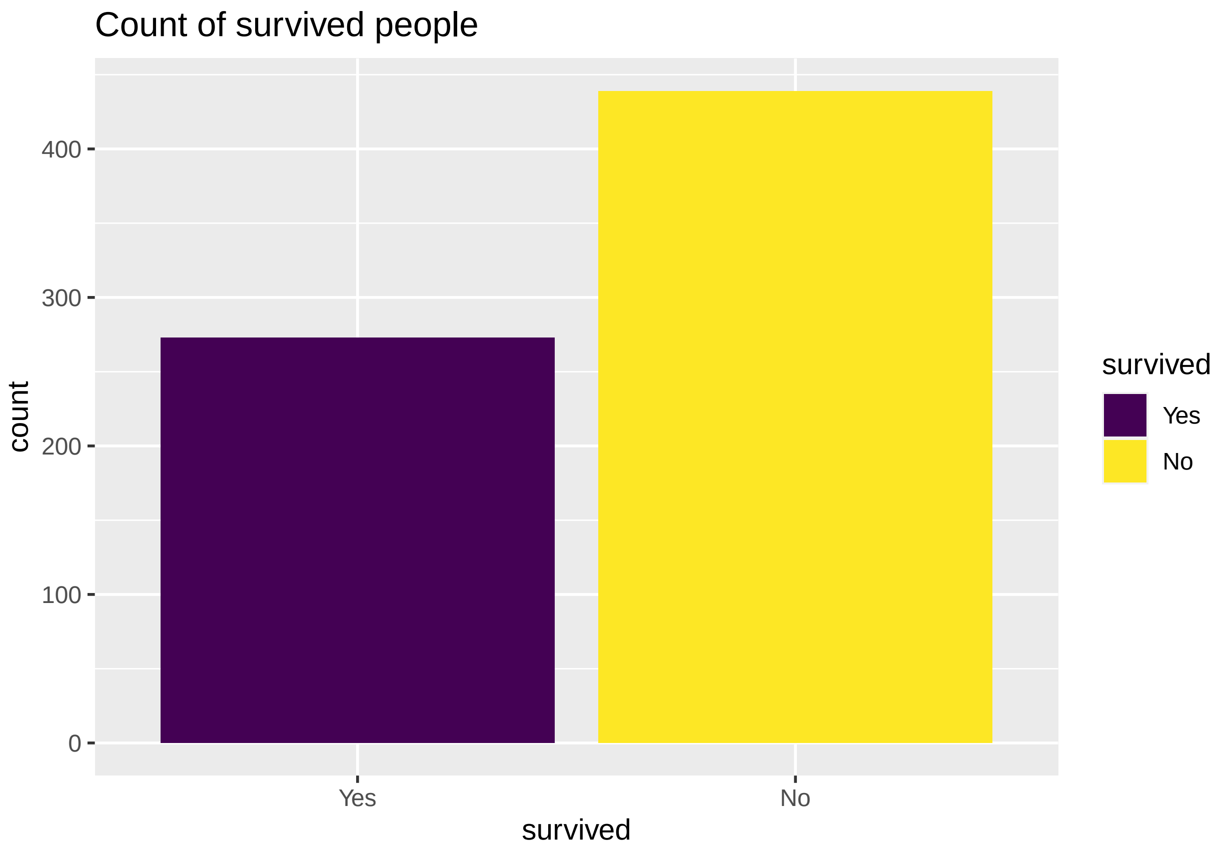
```
##    passenger_id survived pclass
## 5             5       No      3
## 9             9      Yes      3
## 28           28       No      1
## 39           39       No      3
## 49           49       No      3
## 50           50       No      3
##                                                  name    sex age sib_sp parch
## 5                             Allen, Mr. William Henry   male  35      0     0
## 9  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) female  27      0     2
## 28                      Fortune, Mr. Charles Alexander   male  19      3     2
## 39                  Vander Planke, Miss. Augusta Maria female  18      2     0
## 49                                 Samaan, Mr. Youssef   male  NA      2     0
## 50     Arnold-Franchi, Mrs. Josef (Josefine Franchi) female  18      1     0
##    ticket     fare       cabin embarked
## 5  373450   8.0500        <NA>        S
## 9  347742  11.1333        <NA>        S
## 28  19950 263.0000 C23 C25 C27        S
## 39 345764  18.0000        <NA>        S
## 49   2662  21.6792        <NA>        C
## 50 349237  17.8000        <NA>        S
```

From the notice that the age, cabin have missing value and the ticket has different format.
Our goal is predicting the survived people, so we should stratify survived people from different class, sex,age,etc.

Question 2

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

```
titanic_train %>%
  ggplot(aes(x = survived,fill=survived)) +
  geom_bar() +
  ggtitle("Count of survived people")
```

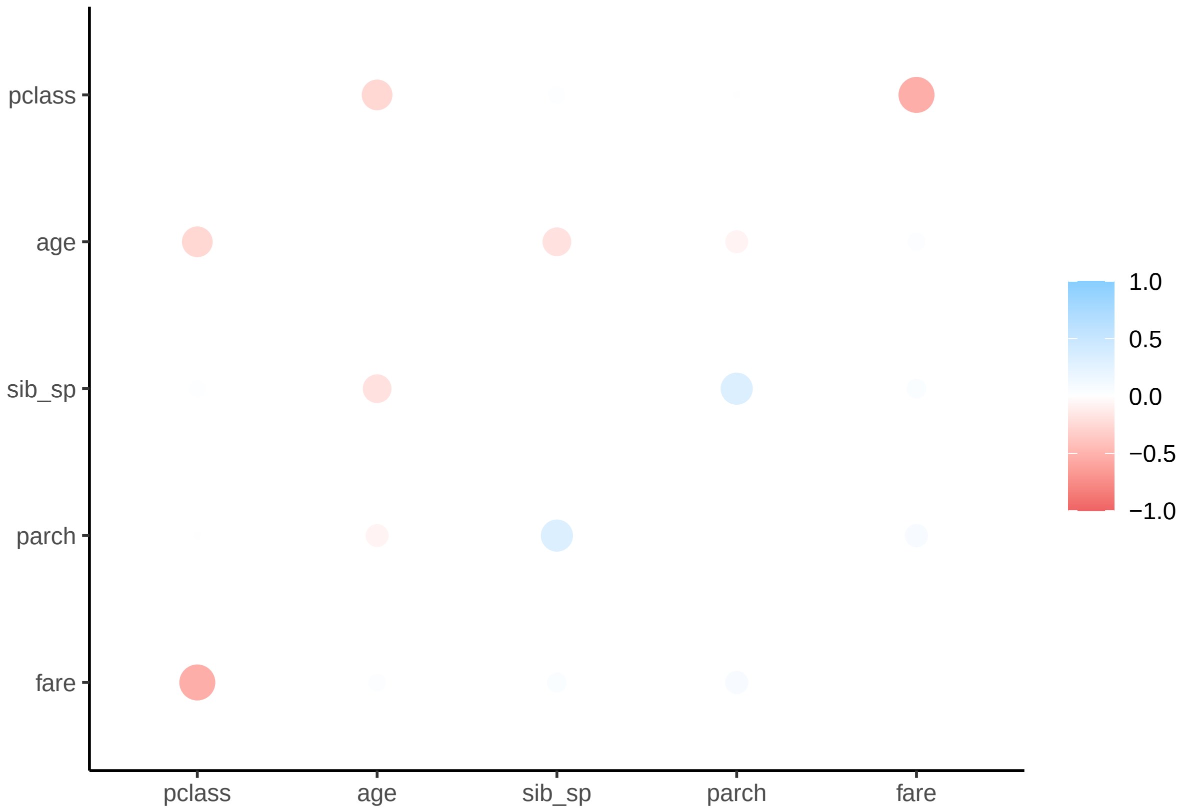


The distribution of the outcome is uneven distribute and the number of no survived people is much more than survived people.

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 %>%
  select( -sex,-passenger_id, -name, -cabin, -ticket,-embarked,-survived) %>%
  mutate(pclass = as.integer(pclass)) %>%
  correlate(use = "pairwise.complete.obs", method = "pearson")
rplot(cor_titanic_train)
```



We want to look for the bright, large circles which shows the strong correlations. The size and shading depends on the absolute values of the coefficients; color depends on direction.
* survived positive correlate to sex, pclass
* pclass negative correlate to fare, age
* age negative correlate to sib_sp
* sib_sp positive correlate to parch
* parch negative correlate to sex, 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.

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

```
  step_interact(terms = ~ starts_with("sex"):fare +
                  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?

```
titanic_train_logistic <- predict(log_fit, new_data = titanic_train, type = "prob")
log_acc <- augment(log_fit, new_data = titanic_train)%>%
  accuracy(truth = survived, estimate = .pred_class)
```

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

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

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

```
titanic_train_predictions <- bind_cols(titanic_train_logistic,
                               titanic_train_lda,titanic_train_qda,titanic_train_nb)
titanic_train_predictions %>%
  head()
```

```
## # A tibble: 6 x 8
##   .pred_Yes...1 .pred_No...2 .pred_Yes...3 .pred_No...4 .pred_Yes...5
##           <dbl>        <dbl>         <dbl>        <dbl>         <dbl>
## 1        0.0949        0.905        0.0580        0.942     0.00443
## 2        0.108         0.892        0.0627        0.937     0.00427
## 3        0.279         0.721        0.231         0.769     0.0398
```

```
## 4        0.0803        0.920        0.0583        0.942     0.0000309
## 5        0.166         0.834        0.0971        0.903     0.00698
## 6        0.0167        0.983        0.0110        0.989     0.00160
## # ... with 3 more variables: .pred_No...6 <dbl>, .pred_Yes...7 <dbl>,
## #   .pred_No...8 <dbl>
```

```
accuracies <- c(log_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      0.813 Logistic Regression
## 2      0.796 LDA
## 3      0.774 QDA
## 4      0.768 Naive Bayes
```

Question 10

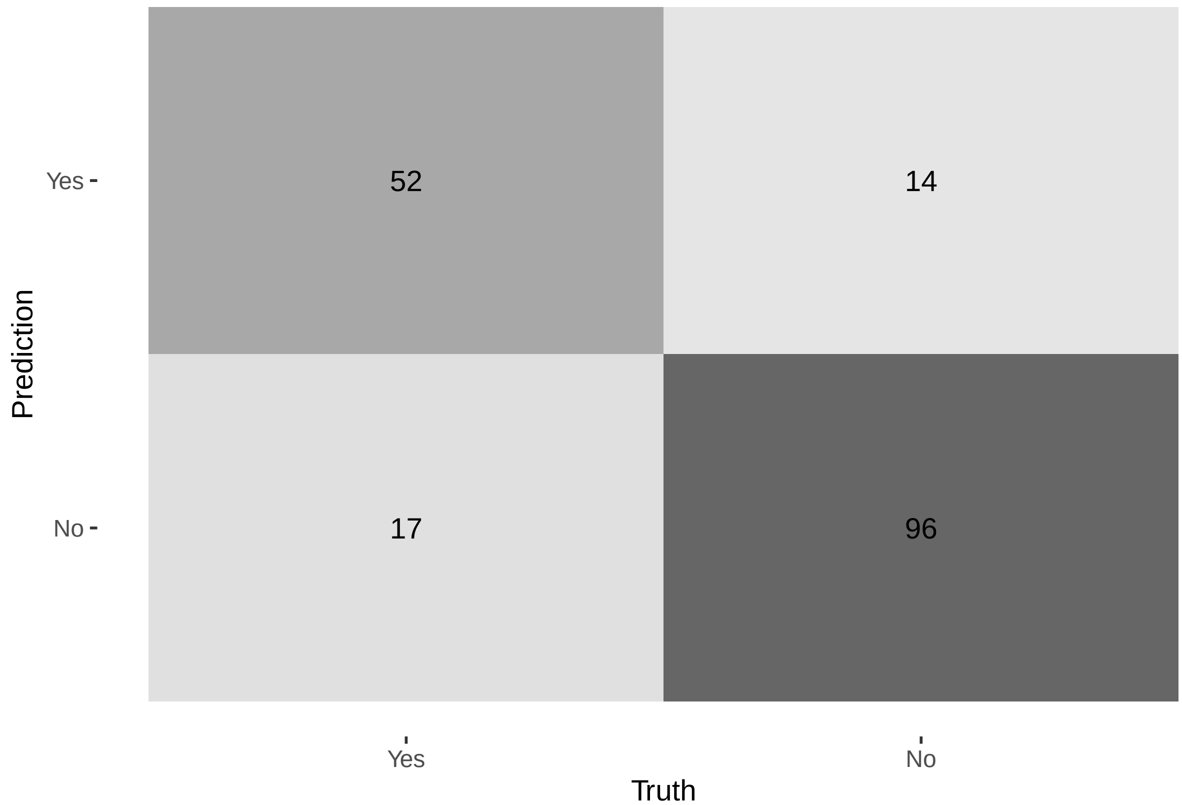
Fit the model with the highest training accuracy to the **testing** data. Report the accuracy of the model on the **testing** data.

```
log_test <- fit(log_wkflow, titanic_test)
predict(log_test, new_data = titanic_test, type = "class") %>%
  bind_cols(titanic_test %>% select(survived)) %>%
  accuracy(truth = survived, estimate = .pred_class)
```

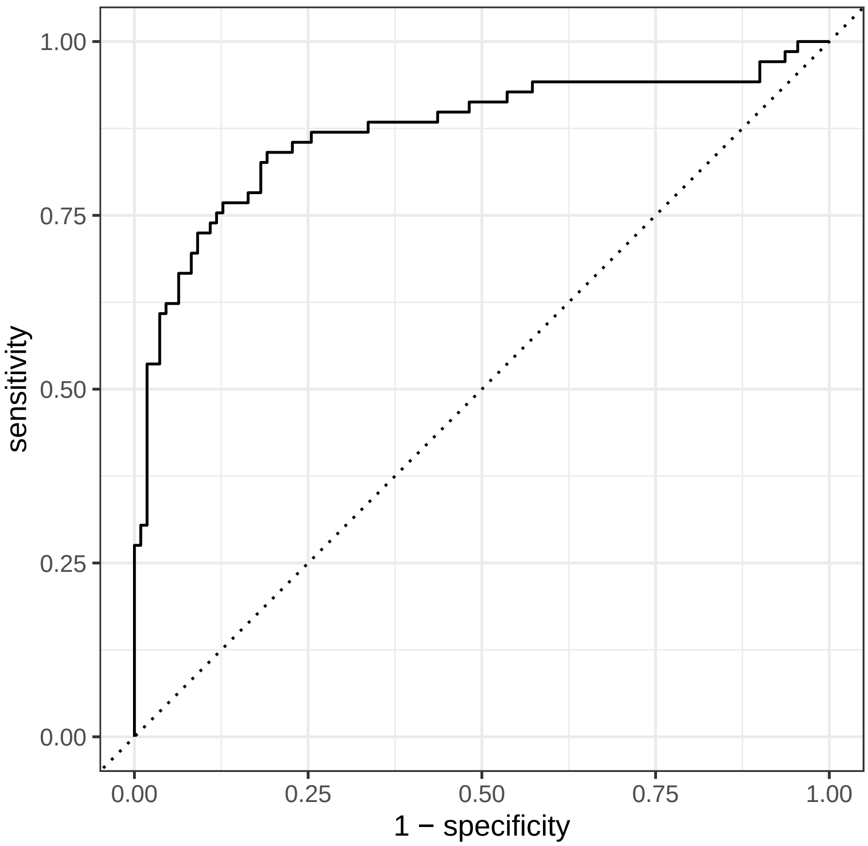
```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.827
```

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

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




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




```
# Calculate AUC
augment(log_test, new_data = titanic_test) %>%
  roc_auc(survived, .pred_Yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.872
```

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

The auc is 0.8715 which means the model perform not bad.

The accurcies of training and testing value are 0.81 and 0.804 which is pretty close, and since we optimized the training model, so the training accuracies is higher.