PSTAT131HW03

Yifei Zhang

2022-04-17

Contents

```r
library(ggplot2)
library(tidyverse)
library(tidymodels)
library(corrplot)
library(ggthemes)
library(discrim)
library(poissonreg)
library(corrr)
library(klaR)
titanic <- read.csv("titanic.csv")
```

```r
titanic $ survived <- as_factor(titanic $ survived)

titanic $ pclass <- as_factor(titanic $ pclass)

titanic %>%
  head()
```

```
##   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
```

```
titanic %>%
  arrange(survived) %>%
  head() # I dont know how to make "Yes" come first
```

```
##   passenger_id survived pclass                           name  sex age sib_sp
## 1            1       No      3        Braund, Mr. Owen Harris male  22      1
## 2            5       No      3       Allen, Mr. William Henry male  35      0
## 3            6       No      3               Moran, Mr. James male  NA      0
## 4            7       No      1        McCarthy, Mr. Timothy J male  54      0
## 5            8       No      3 Palsson, Master. Gosta Leonard male   2      3
## 6           13       No      3 Saundercock, Mr. William Henry male  20      0
##   parch    ticket    fare cabin embarked
## 1     0 A/5 21171  7.2500  <NA>        S
## 2     0    373450  8.0500  <NA>        S
## 3     0    330877  8.4583  <NA>        Q
## 4     0     17463 51.8625   E46        S
## 5     1    349909 21.0750  <NA>        S
## 6     0 A/5. 2151  8.0500  <NA>        S
```

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.

The training and testing data sets seem to have the appropriate number of observations. First of all we have a lot of missing data in our dataset especially for the cabin and age section which I personally think are very important for our analysis, secondly in the ticket section the records are in combination of characters and numbers and that can be hard to analyze. We also have variables that are likely to correlate to each other.

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

It is a good idea to use stratified sampling for this data because we have a variety of attributes, make all the subgroups present in our sample and our result will be more accurate.We will be able to see fit better for the general population on their likelihood to survive on the sinking titanic.

```r
set.seed(1010)

titanic_split <- initial_split(titanic, prop = 0.70,
                               strata = survived)

titanic_train <- training(titanic_split)

titanic_test <- testing(titanic_split)

count(titanic_train, "passenger_id") # see how many rows
```

```
##   "passenger_id"   n
## 1   passenger_id 623
```

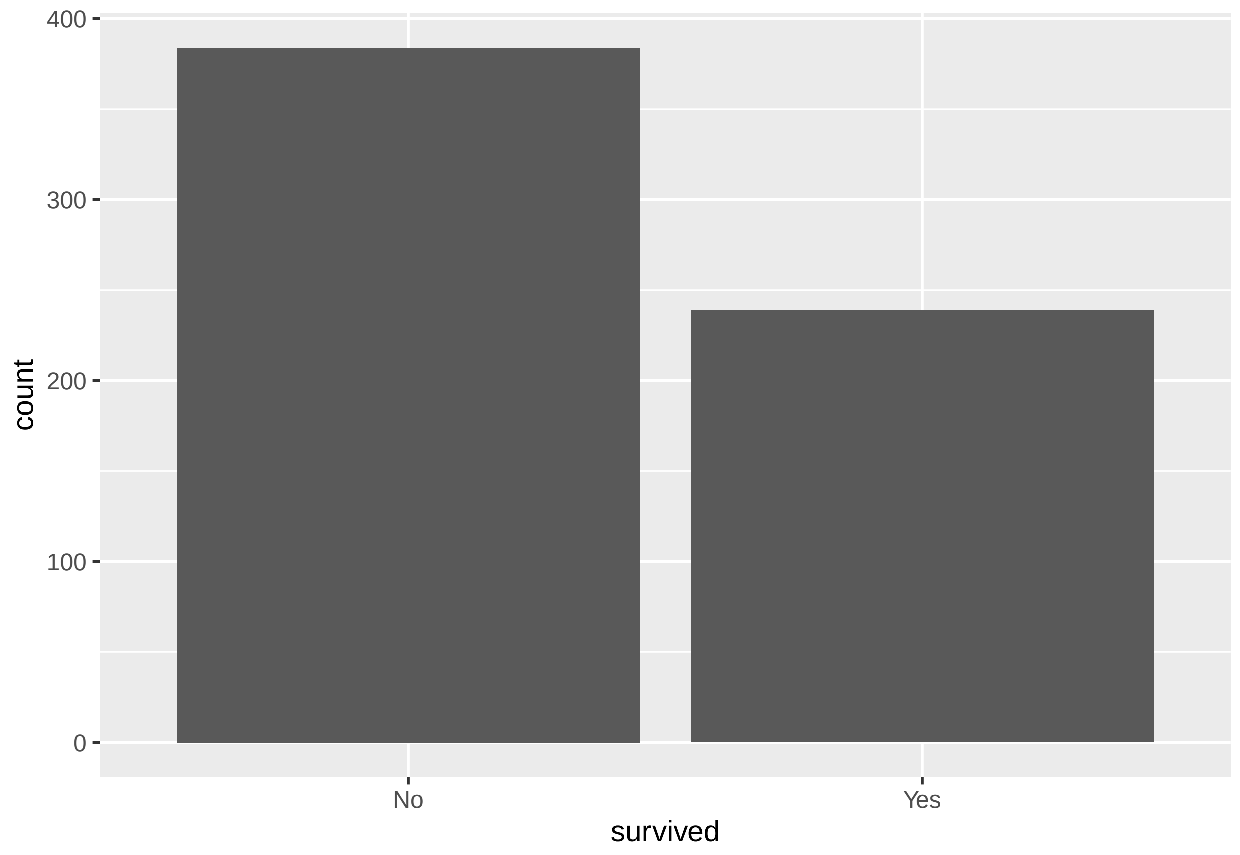
```r
count(titanic_test, "passenger_id") # see how many rows
```

```
##   "passenger_id"   n
## 1   passenger_id 268
```

Question 2

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

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



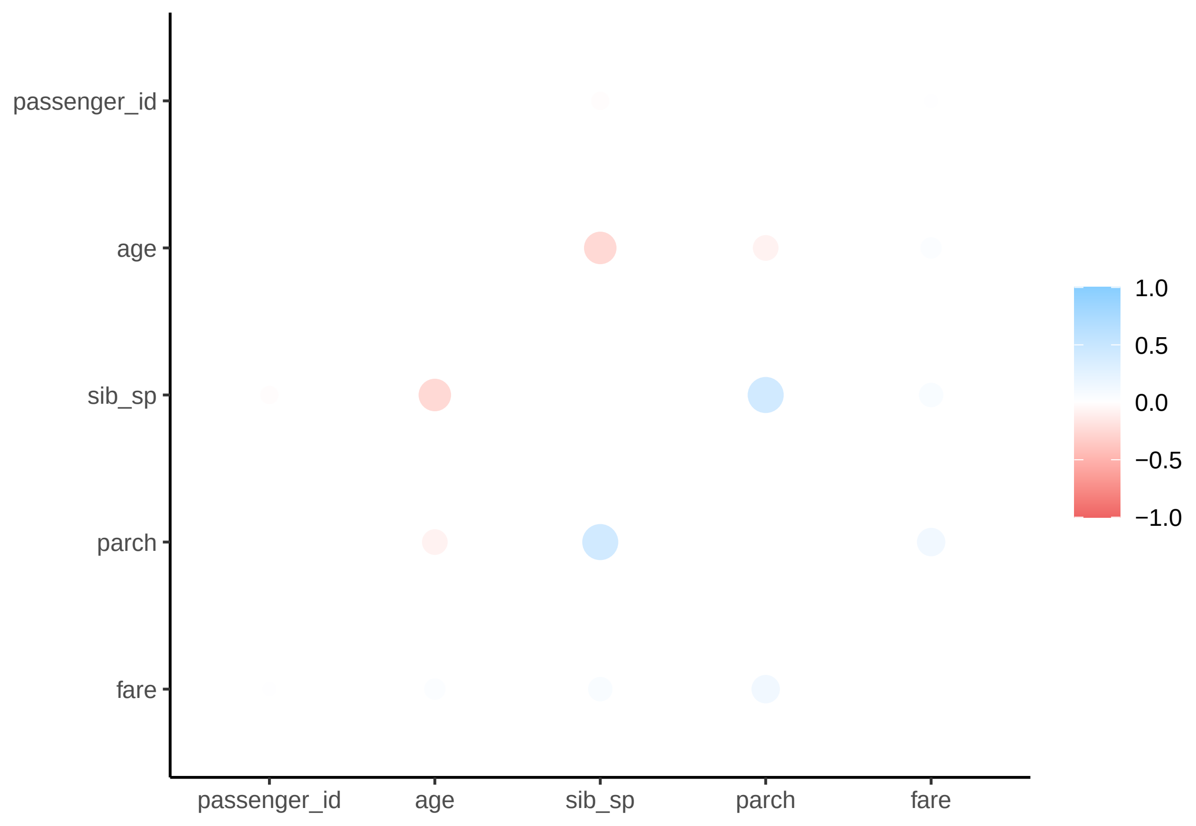
Question 3

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?

The numeric variables are passenger_id, age, sib_sp, parch, fare, although pclass seems like one of them, pclass is a factor and not numeric. A parttern I see is that a lot of them are not correlated at all. The number of siblings/ spouses aboard the Titanic has a relatively strong positive correlation with the number of parents/ children aboard the Titanic. The number of parents/ children aboard the Titanic and fare has a slightly positive correlation. The number of siblings/ spouses aboard the Titanic and fare also has a slightly positive correlation. The number of siblings/ spouses aboard the Titanic has a relatively strong negative correlation with age. There is a slightly negative correlation between the number of parents/ children aboard the Titanic and age.

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

rplot(cor_titanic_train)
```



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_train_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_train_recipe
```

```
## Data 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
```

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_train_recipe)

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)      4.88      0.717       6.81   9.62e-12
##  2 age             -0.0730    0.0142     -5.15   2.56e- 7
##  3 sib_sp          -0.517     0.140      -3.70   2.12e- 4
##  4 parch           -0.150     0.155      -0.964  3.35e- 1
##  5 fare             0.000186  0.0112      0.0166 9.87e- 1
##  6 pclass_X2       -1.06      0.385      -2.76   5.82e- 3
##  7 pclass_X3       -2.58      0.405      -6.36   2.05e-10
##  8 sex_male        -2.54      0.327      -7.77   7.84e-15
##  9 sex_male_x_fare -0.0142    0.00924    -1.53   1.25e- 1
## 10 age_x_fare       0.000440  0.000204    2.16   3.11e- 2
```

Question 6

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

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

lda_wkflow <- workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(titanic_train_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.

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

qda_wkflow <- workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(titanic_train_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`.

```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_train_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?

According to our results so far, it looks like the logistic regression model has the highest accuracy on the training data with an accuracy thats roughly 83.14607 %.

```r
log_fit_pred <- predict(log_fit, new_data = titanic_train, type = "prob")
lda_fit_pred <- predict(lda_fit, new_data = titanic_train, type = "prob")
qda_fit_pred <- predict(qda_fit, new_data = titanic_train, type = "prob")
nb_fit_pred <- predict(nb_fit, new_data = titanic_train, type = "prob")
```

```r
bind_cols(log_fit_pred, lda_fit_pred, qda_fit_pred, nb_fit_pred)
```

```
## # A tibble: 623 x 8
##    .pred_No...1 .pred_Yes...2 .pred_No...3 .pred_Yes...4 .pred_No...5
##           <dbl>         <dbl>        <dbl>         <dbl>        <dbl>
##  1        0.941        0.0586        0.969        0.0309        0.998
##  2        0.913        0.0866        0.954        0.0455        0.996
##  3        0.749        0.251         0.815        0.185         0.963
##  4        0.913        0.0866        0.948        0.0517        1.00
##  5        0.850        0.150         0.920        0.0796        0.992
##  6        0.952        0.0484        0.969        0.0310        1.00
##  7        0.775        0.225         0.843        0.157         0.988
##  8        0.476        0.524         0.359        0.641         0.999
##  9        0.913        0.0865        0.955        0.0454        0.996
## 10        0.617        0.383         0.704        0.296         0.889
## # ... with 613 more rows, and 3 more variables: .pred_Yes...6 <dbl>,
## #   .pred_No...7 <dbl>, .pred_Yes...8 <dbl>
```

```r
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)
```

```r
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         0.831
```

```r
lda_acc
```

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

```
qda_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.811
```

```
nb_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.795
```

Question 10

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

The model with the highest training accuracy got a 84.35754% accuracy 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?

The model did pretty well, the model worked better on the training data than the test data which are 83.14607% to 79.10448%. They are pretty close. I think the reason why the accuracy is higher for the training data than the test data maybe because of over fitting. We had more data to work with in the training set. And that is okay, we should get higher accuracy on the training data anyway.

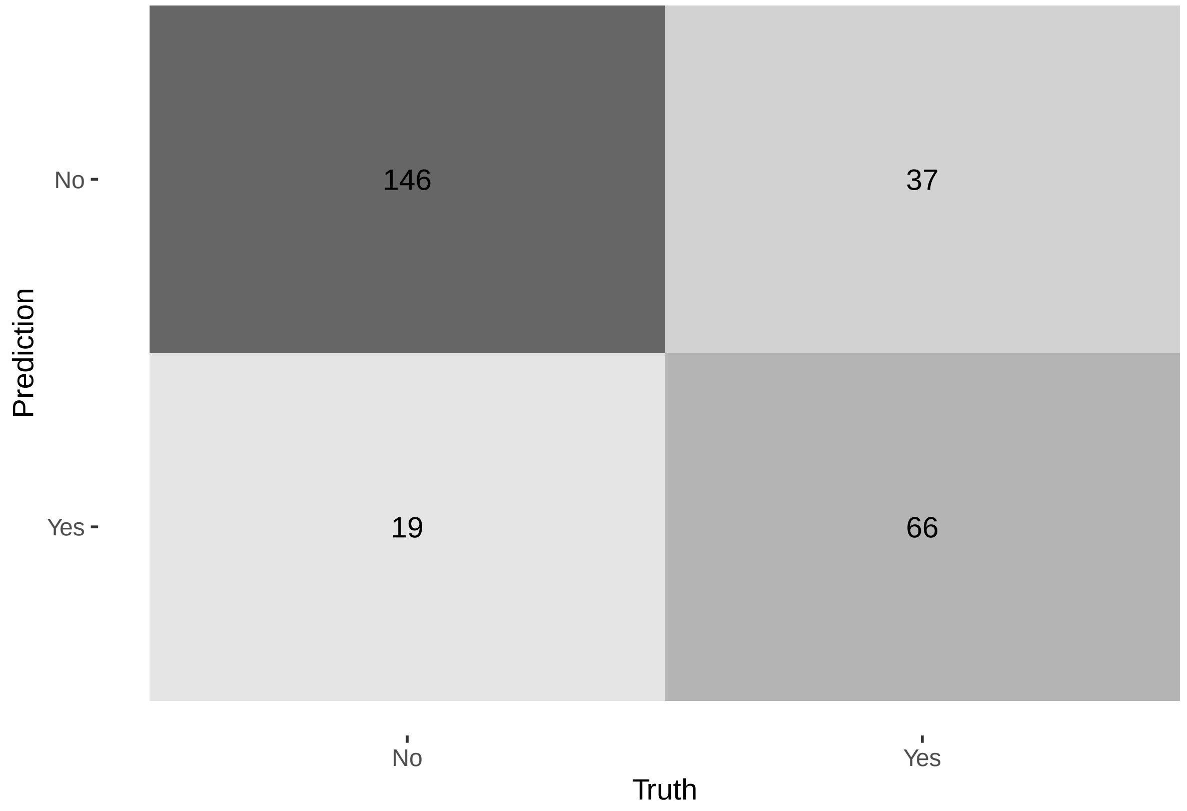
```
predict(log_fit, new_data = titanic_test, type = "prob")
```

```
## # A tibble: 268 x 2
##    .pred_No .pred_Yes
##       <dbl>     <dbl>
##  1   0.916     0.0842
##  2   0.458     0.542
##  3   0.0778    0.922
##  4   0.201     0.799
##  5   0.986     0.0141
##  6   0.209     0.791
##  7   0.556     0.444
##  8   0.362     0.638
##  9   0.915     0.0849
## 10   0.913     0.0866
## # ... with 258 more rows
```

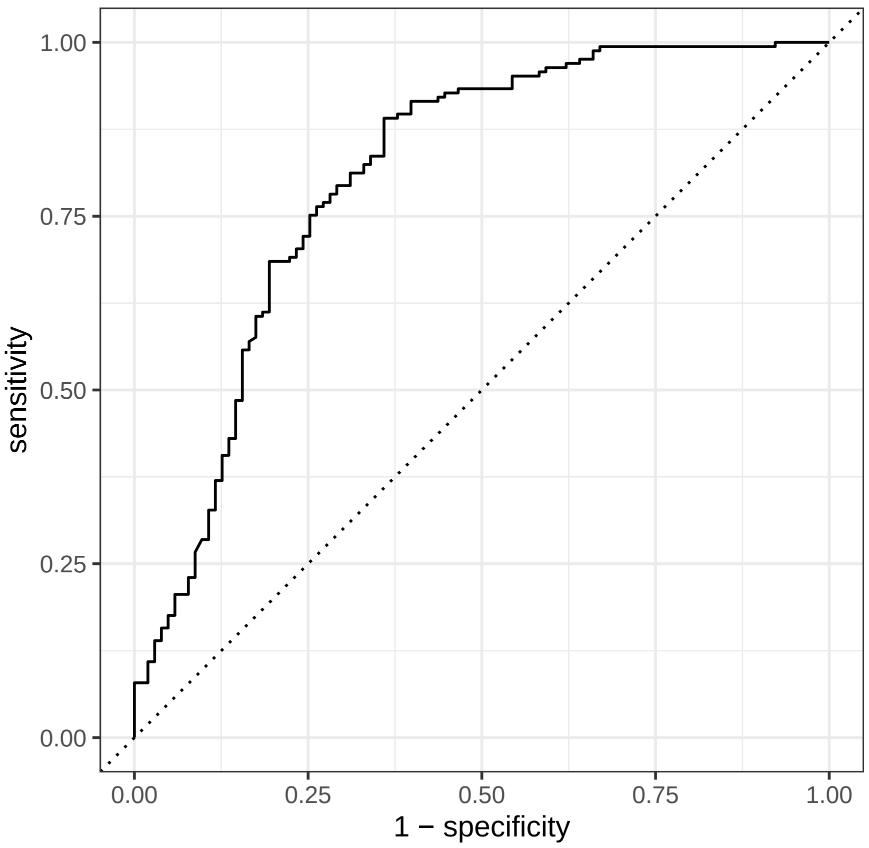
```r
log_reg_acc <- augment(log_fit, new_data = titanic_test) %>%
  accuracy(truth = survived, estimate = .pred_class)
log_reg_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.791
```

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



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


1.00
0.75
0.50
0.25
0.00
sensitivity
0.00
0.25
0.50
0.75
1.00
1 − specificity