PSTAT 131 Homework 3

Luke Fields (8385924)

April 15, 2022

Below are the packages and libraries we are using in this assignment.

```
library(corrplot)
library(discrim)
library(poissonreg)
library(corrr)
library(klaR) # for naive bayes
library(knitr)
library(MASS)
library(tidyverse)
library(tidymodels)
library(ggplot2)
library("dplyr")
library("yardstick")
tidymodels_prefer()
titanic <- read_csv("titanic.csv")
# set global chunk options: images will be 7x5 inches
knitr::opts_chunk$set(
    echo = TRUE,
    fig.height = 5,
    fig.width = 7,
    tidy = TRUE,
    tidy.opts = list(width.cutoff = 60)
)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
options(digits = 4)



## indents are for indenting r code as formatted text
## They may need to be adjusted depending on your OS
# if your output looks odd, increase or decrease indent
indent1 = '    '
indent2 = '        '
indent3 = '            '
```

Before we begin working with our model, we will factorize the survived and pclass variables first, making sure that "Yes" is the first level in our data set.

```
set.seed(912)
dim(titanic)
```

```
## [1] 891  12
```

```
survived_levels <- c("Yes", "No")
titanic$survived <- as.factor(titanic$survived)
titanic$survived <- relevel(titanic$survived, "Yes")
titanic$pclass <- as.factor(titanic$pclass)
titanic
```

```
## # A tibble: 891 x 12
##    passenger_id survived pclass name       sex     age sib_sp parch ticket  fare
##           <dbl> <fct>    <fct>  <chr>      <chr> <dbl>  <dbl> <dbl> <chr>  <dbl>
##  1            1 No       3      Braund, M~ male     22      1     0 A/5 2~  7.25
##  2            2 Yes      1      Cumings, ~ fema~    38      1     0 PC 17~ 71.3
##  3            3 Yes      3      Heikkinen~ fema~    26      0     0 STON/~  7.92
##  4            4 Yes      1      Futrelle,~ fema~    35      1     0 113803 53.1
##  5            5 No       3      Allen, Mr~ male     35      0     0 373450  8.05
##  6            6 No       3      Moran, Mr~ male     NA      0     0 330877  8.46
##  7            7 No       1      McCarthy,~ male     54      0     0 17463  51.9
##  8            8 No       3      Palsson, ~ male      2      3     1 349909 21.1
##  9            9 Yes      3      Johnson, ~ fema~    27      0     2 347742 11.1
## 10           10 Yes      2      Nasser, M~ fema~    14      1     0 237736 30.1
## # ... with 881 more rows, and 2 more variables: cabin <chr>, embarked <chr>
```

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(912)
titanic_split <- initial_split(titanic,
                               prop = 0.7, strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
no_of_titanic_rows <- nrow(titanic)
no_of_train_rows <- nrow(titanic_train)
no_of_test_rows <- nrow(titanic_test)
no_of_missing_train <- colSums(is.na(titanic_train))
titanic_train
```

```
## # A tibble: 623 x 12
##    passenger_id survived pclass name      sex     age sib_sp parch ticket   fare
##           <dbl> <fct>    <fct>  <chr>     <chr> <dbl>  <dbl> <dbl> <chr>   <dbl>
##  1            1 No       3      Braund, ~ male     22      1     0 A/5 2~   7.25
##  2            5 No       3      Allen, M~ male     35      0     0 373450   8.05
##  3            8 No       3      Palsson,~ male      2      3     1 349909  21.1
##  4           13 No       3      Saunderc~ male     20      0     0 A/5. ~   8.05
##  5           19 No       3      Vander P~ fema~    31      1     0 345763  18
##  6           21 No       2      Fynney, ~ male     35      0     0 239865  26
##  7           25 No       3      Palsson,~ fema~     8      3     1 349909  21.1
##  8           27 No       3      Emir, Mr~ male     NA      0     0 2631      7.22
##  9           28 No       1      Fortune,~ male     19      3     2 19950   263
## 10           31 No       1      Uruchurt~ male     40      0     0 PC 17~  27.7
## # ... with 613 more rows, and 2 more variables: cabin <chr>, embarked <chr>
```

Above is what our training dataset looks like.

```
no_of_titanic_rows
```

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

```
no_of_train_rows
```

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

```
no_of_test_rows
```

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

The previous three rows are the amount of observations in the titanic data set, as well as the train and test sets we created.

```
no_of_train_rows / no_of_titanic_rows
```

```
## [1] 0.6992
```

```
no_of_test_rows / no_of_titanic_rows
```

```
## [1] 0.3008
```

The above two rows give us the proportion of our training and test sets compared to our original titanic data set.

```
no_of_missing_train
```

```
## passenger_id     survived       pclass         name          sex          age 
##            0            0            0            0            0          121 
##       sib_sp        parch       ticket         fare        cabin     embarked 
##            0            0            0            0          489            0
```

We can see the number of missing values in our training set above.

After performing a 70/30 train/test split, we see that there are 623 (69.9% of our original titanic data) and 268 (30.1% of our original titanic data) observations in the training data set and test datas et, respectively, so it is verified that the training and testing sets have the correct dimension. Iwthin our training dataset, there are missing values in age and cabin, where age has about 20% of its values being missing, and cabin having nearly 75% of its values missing. We want to use stratified sampling when we want to understand the relationship between two types of variables, in this case, survived or not survived. Our sample is able to be divided into different subgroups, so stratified sampling is a good idea in this case.

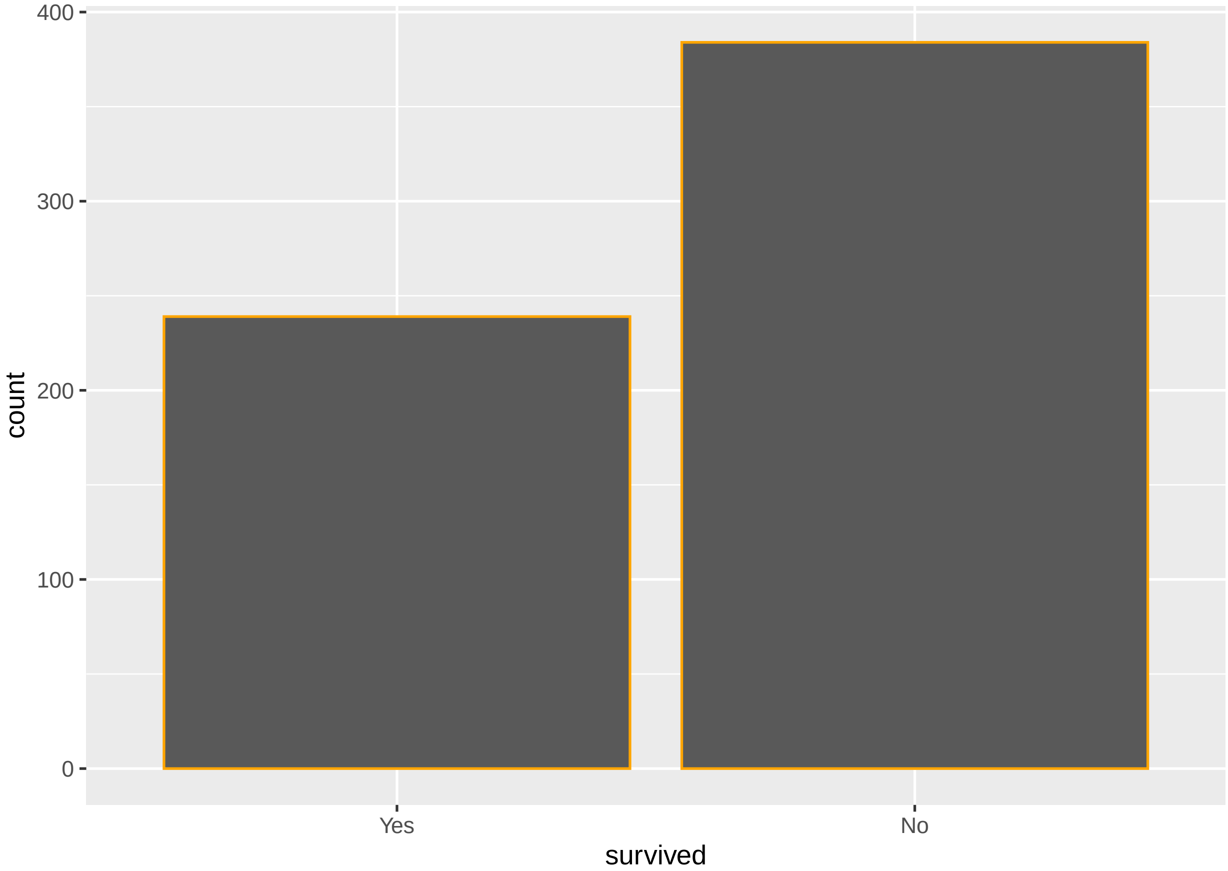
Question 2: Using the training data set, explore/describe the distribution of the outcome variable survived.

```
survived_bar <- titanic_train %>%
  ggplot(aes(x = survived)) +
  geom_bar(color = "orange")

survived_density <- titanic_train %>%
  ggplot(aes(x = survived)) +
  geom_density(color = "orange")

survived_box <- titanic_train %>%
  ggplot(aes(x = survived)) +
  geom_boxplot(color = "orange")

survived_bar
```



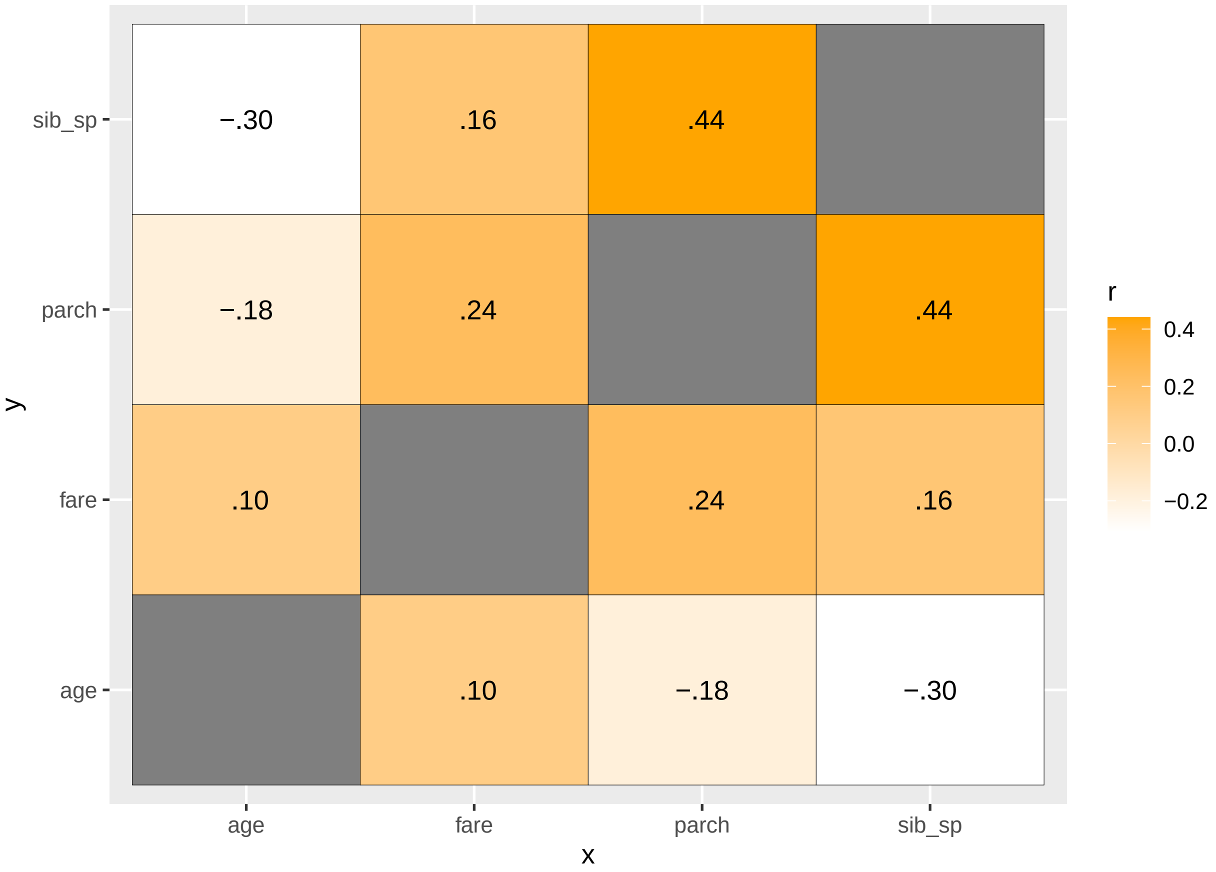
Looking at the training dataset, we can see that majority of the people aboard the titanic did not survive. It looks like a little less than 250 people in our training data set survived, while just under 400 died.

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 <- titanic_train %>%
  select("age", "fare", "sib_sp", "parch") %>%
  correlate() %>%
  stretch() %>%
  ggplot(aes(x, y, fill = r)) + geom_tile() +
  geom_tile(color = "black") +
  scale_fill_gradient(low = "white", high = "orange") +
  geom_text(aes(label = as.character(fashion(r))))
```

```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```

```
cor_titanic
```



There is not much correlation going on with any of the continuous predictor variables, which is kind of surprising. The continuous variables in this dataset are age, fare, number of siblings / spouses aboard, and number of parents / children aboard, as these are all numeric, measurable variables. The amount of siblings /spouses aboard and the number of parents / children aboard are slightly positively correlated with a correlation factor of 0.44, which makes sense as families most likely traveled together if they had the ability to.

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.

```
titanic_recipe <-
  recipe(survived ~ pclass + sex + age +
           sib_sp + parch + fare, data = titanic_train) %>%
  step_impute_linear(age, impute_with = imp_vars(all_predictors())) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(terms = ~sex_male:fare) %>%
  step_interact(terms = ~age:fare)

titanic_recipe
```

```
## 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
```

Here we created a recipe for our models to use in the rest of this assignment, attempting to predict survival of a person based on their ticket class, sex, age, number of siblings or spouses aboard, number of parents or children aboard, and passenger fare. We used the step_impute_linear function to impute missing values for age using a linear model predictor for each of the NA values. We also use step_dummy to create dummy variables for our categorical predictors, and step_interact to create interaction terms between sex and fare as well as age and 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.

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

titanic_log_workflow <- workflow() %>%
  add_model(titanic_log_reg) %>%
  add_recipe(titanic_recipe)

titanic_log_fit <- fit(titanic_log_workflow, titanic_train)
```

Here we applied a workflow to our titanic training data for a logistic regression model.

Question 6: Repeat Question 5, but this time specify a linear discriminant analysis model for classification using the “MASS” engine.

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

titanic_lda_workflow <- workflow() %>%
  add_model(titanic_lda_mod) %>%
  add_recipe(titanic_recipe)

titanic_lda_fit <- fit(titanic_lda_workflow, titanic_train)
```

Here we applied a workflow to our titanic training data for a linear discriminant analysis model.

Question 7: Repeat Question 5, but this time specify a quadratic discriminant analysis model for classification using the “MASS” engine.

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

titanic_qda_workflow <- workflow() %>%
  add_model(titanic_lda_mod) %>%
  add_recipe(titanic_recipe)

titanic_qda_fit <- fit(titanic_qda_workflow, titanic_train)
```

Here we applied a workflow to our titanic training data for a quadratic discriminant analysis model.

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

titanic_nb_workflow <- workflow() %>%
  add_model(titanic_nb_mod) %>%
  add_recipe(titanic_recipe)

titanic_nb_fit <- fit(titanic_nb_workflow, titanic_train)
```

Here we applied a workflow to our titanic training data for a naive Bayes model.

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_log_reg_pred <- predict(titanic_log_fit,
                           new_data = titanic_train)
titanic_lda_pred <- predict(titanic_lda_fit,
                           new_data = titanic_train)
titanic_qda_pred <- predict(titanic_qda_fit,
                           new_data = titanic_train)
titanic_nb_pred <- predict(titanic_nb_fit,
                           new_data = titanic_train)

titanic_train_pred <- bind_cols(titanic_log_reg_pred,
                                titanic_lda_pred,
                                titanic_qda_pred,
                                titanic_nb_pred,
                                titanic_train$survived)
```

```
## New names:
## * .pred_class -> .pred_class...1
## * .pred_class -> .pred_class...2
## * .pred_class -> .pred_class...3
## * .pred_class -> .pred_class...4
## * ‘‘ -> ...5
```

```
names(titanic_train_pred) <- c("Log Reg Survived", "LDA Survived", "QDA Survived", "Naive Bayes Survive
titanic_train_pred
```

```
## # A tibble: 623 x 5
##    ‘Log Reg Survived‘ ‘LDA Survived‘ ‘QDA Survived‘ ‘Naive Bayes Survived‘
##    <fct>              <fct>          <fct>          <fct>
##  1 No                 No             No             No
##  2 No                 No             No             No
##  3 No                 No             No             No
##  4 No                 No             No             No
##  5 No                 Yes            Yes            No
##  6 No                 No             No             No
##  7 Yes                Yes            Yes            No
##  8 No                 No             No             No
##  9 No                 No             No             Yes
## 10 No                 No             No             No
## # ... with 613 more rows, and 1 more variable: ‘Actually Survived‘ <fct>
```

Above is our data frame that contains the predictions for survived by model. The first four columns are the models we just fit (Logistic Regression, LDA, QDA, and Naive Bayes, in that order), and the last column is the actual outcome from our training data set.

```
titanic_log_reg_acc <- augment(titanic_log_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
```

```
titanic_lda_acc <- augment(titanic_lda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
titanic_qda_acc <- augment(titanic_qda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
titanic_nb_acc <- augment(titanic_nb_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)


titanic_accuracies <- c(titanic_log_reg_acc$.estimate,
                        titanic_lda_acc$.estimate,
                        titanic_qda_acc$.estimate,
                        titanic_nb_acc$.estimate)
models <- c("Log Reg", "LDA", "QDA", "Naive Bayes")
results <- tibble(accuracies = titanic_accuracies, models = models)
results %>%
  arrange(-titanic_accuracies)
```

```
## # A tibble: 4 x 2
##   accuracies models
##        <dbl> <chr>
## 1      0.822 Log Reg
## 2      0.801 LDA
## 3      0.801 QDA
## 4      0.780 Naive Bayes
```

This is a table that describes the accuracy of each of our four models (Logistic Regression, LDA, QDA, and Naive Bayes, in that order) in terms of correctly predicting whether someone survived or not. Logistic Regression had the highest accuracy in predicting survival, with a 0.8218 accuracy rate in its predictions, so we will use that on our test set.

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?

```
titanic_test_pred <- predict(titanic_log_fit,
                             new_data = titanic_test,
                             type = "prob") %>%
  bind_cols(titanic_test %>% select(survived))

titanic_confus_mat <- augment(titanic_log_fit, new_data = titanic_test) %>%
  conf_mat(truth = survived, estimate = .pred_class)

titanic_test_acc <- augment(titanic_log_fit, new_data = titanic_test) %>%
  accuracy(truth = survived, estimate = .pred_class)

titanic_roc_plot <- augment(titanic_log_fit, new_data = titanic_test) %>%
  roc_curve(survived, .pred_Yes) %>%
  autoplot()
```

```
names(titanic_test_pred) <- c("Survived_Prediction_Probability", "Did_Not_Survive_Prediction_Probability
titanic_test_pred
```

```
## # A tibble: 268 x 3
##    Survived_Prediction_Probability Did_Not_Survive_Prediction_~ Actually_Surviv~
##                              <dbl>                        <dbl> <fct>
##  1                          0.954                        0.0456 Yes
##  2                          0.104                        0.896  No
##  3                          0.281                        0.719  No
##  4                          0.0259                       0.974  No
##  5                          0.759                        0.241  No
##  6                          0.0495                       0.951  No
##  7                          0.623                        0.377  Yes
##  8                          0.103                        0.897  No
##  9                          0.488                        0.512  No
## 10                          0.103                        0.897  Yes
## # ... with 258 more rows
```

This is the "prediction of class", or prediction of whether or not someone survived or not based on our predictor variables for all 268 observations in our test set. The "Survived Prediction Probability" column is what our model predicts is the probability of a passenger surviving, the "Did Not Survive Prediction Probability" column is what our model predicts is the probability of a passenger not surviving, and the "Actually Survived?" column is whether or not the passenger actually survived.

```
titanic_confus_mat
```

```
##           Truth
## Prediction Yes  No
##        Yes  75  26
##        No   28 139
```

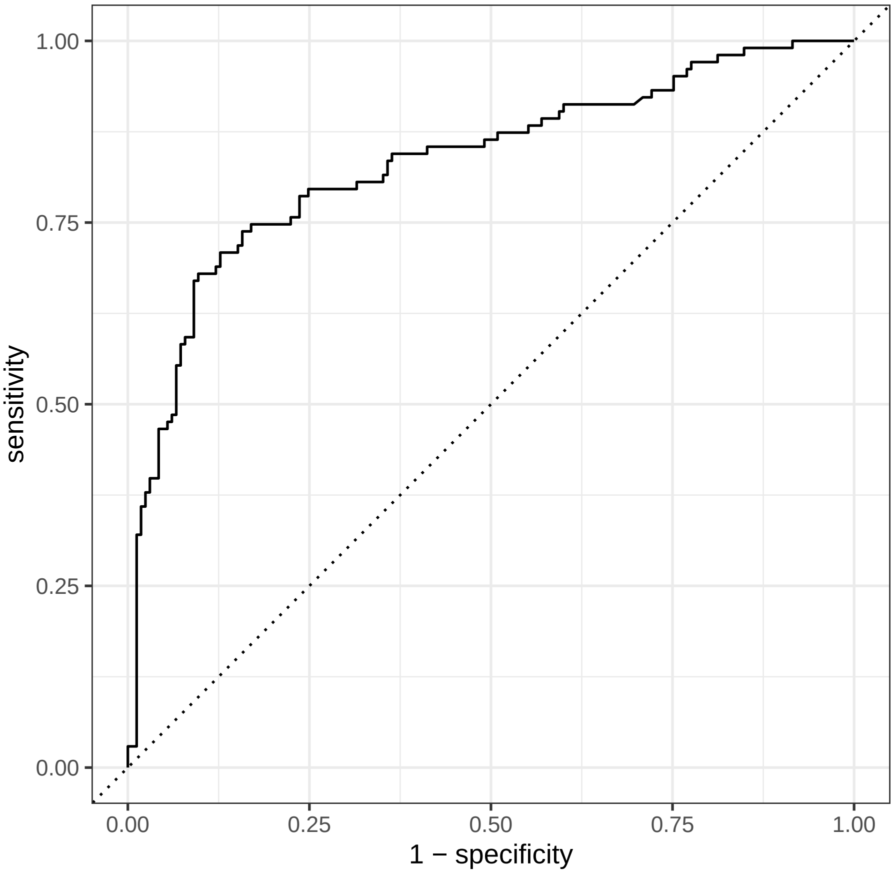
The confusion matrix for the logistic regression model being applied to our test set shows that only 28 of the 103 people to of survived were predicted to die, and only 26 of the 165 people that died were predicted to survive. In other words, only 54 of the 268 predictions our model produced were incorrect. Nice!

```
titanic_test_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.799
```

This gives us the proportion for the 54 of 268 number we just discovered in the previous text, so roughly 80% of our predictions were correct using the logistic regression model.

```
titanic_roc_plot
```



```
titanic_roc_auc <- titanic_test_pred %>%
  roc_auc(Actually_Survived, Survived_Prediction_Probability)
```

Above. is the ROC curve for survival through our test set Below is the ROC curve's area under the curve, which is 0.8324, close to our accuracy estimate.

```
titanic_roc_auc$.estimate
```

```
## [1] 0.8324
```

Below, we compare the difference between our training and test accuracy.

```
titanic_log_reg_acc$.estimate
```

```
## [1] 0.8218
```

```
titanic_test_acc$.estimate
```

```
## [1] 0.7985
```

In conclusion, our model performed quite well. We had fairly high accuracy ratings for both training and testing, hovering around 80% for each. Our training model had slightly higher accuracy, but this is probably due to the model learning how to perform on the larger training set. Regardless, having our accuracy rates that close in percentage is quite splendid. For our final logistic regression model to correctly predict whether a passenger survived the titanic wreck 80% of the time, we should be excited about the results.

END OF HOMEWORK 3