Predicting Passenger Survival of the Titanic

```
#load packages
library(tidyverse)
library(tidymodels)
library(rpart)
library(rpart.plot)
library(partykit)
set.seed(202204) # set seed for reproducibility
```

When RMS Titanic first set sail on April 15, 1912, they were first considered unsinkable. But ,as we all know, they collided an iceberg and sank with most of the passengers in them. There weren't enough lifeboats for everyone on board, resulting in the death of 1502 out of 2224 passengers and crew. In this assignment, we want to analyze and predict which factors affects passenger's surviving rate by looking at different factors that we know about the passengers on board titanic.[1]

Using the `titanic` data set. Our goal is to understand the data utilizing predictive models such as the decision tree model - with the dependent variable as `survived` (survived: Yes, No) and independent variable as `pclass`, `sex`, `age`, `fare`, and `embarked`.

```
# load titanic data
titanic <- read_csv("titanic.csv") %>%
  mutate_if(is.character, factor) %>%
  mutate_if(is.numeric, round, digits = 2) %>%
  select(-c(name, home.dest, sibsp,parch))
```

Below we define the model equation in terms of variables.

```
form <- as.formula(
  "survived ~ pclass + sex + age + fare + embarked")
```

Next, we split the data set into training and test set. The training set is for making the model learn while the test set is for testing the model on its predictions after training.

```
# split the data into a training and test set.
train_prop <- 0.80 # let 80% of rows be the training set

# Here we are using a simple random sampling of the data
n <- nrow(titanic)
titanic_initial <- titanic %>%
  initial_split(prop = train_prop)
titanic_train <- titanic_initial %>% training()
titanic_test <- titanic_initial %>% testing()
```

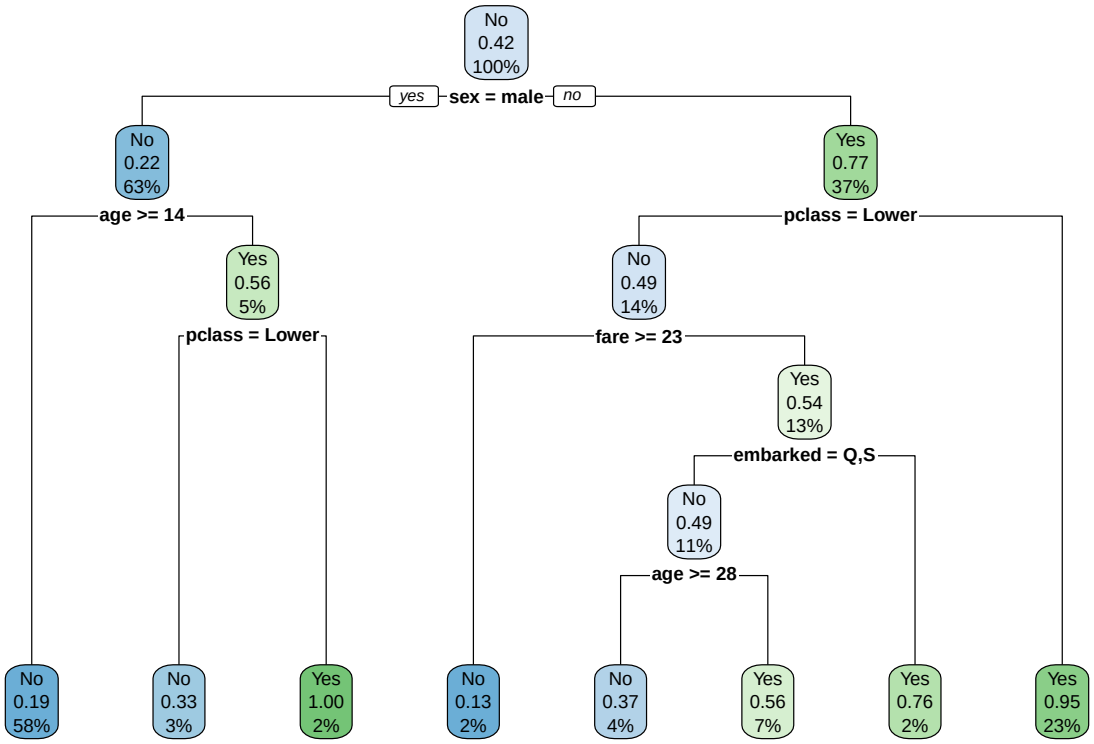
[1] Credit to Simon Ahn for this mini-assignment.

Decision Trees

We are using a decision tree model to predict passenger survival.

```
# Model decision tree
mod_tree <- decision_tree(mode = "classification") %>%
  set_engine("rpart") %>%
  fit(form, data = titanic_train)
```

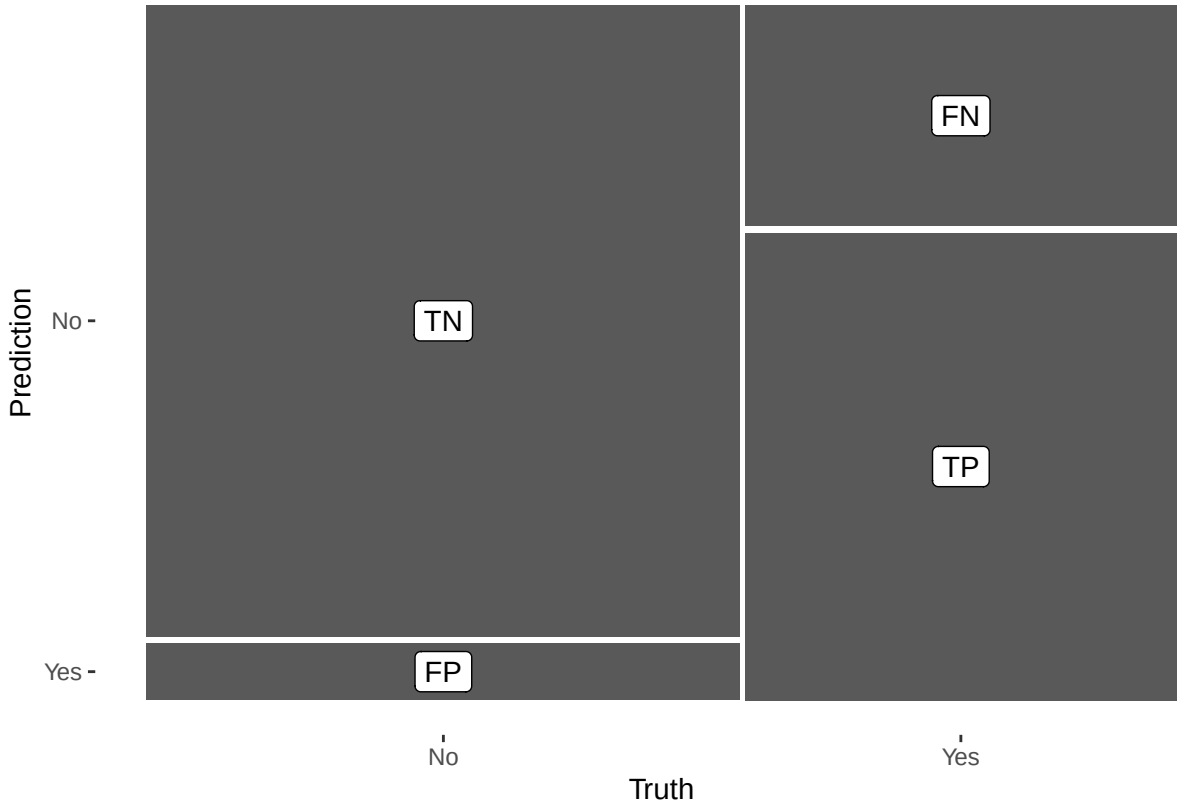
```
rpart.plot(mod_tree$fit, roundint=FALSE)
```



We evaluate the model on how accurate it is on predicting passenger survival. Below compute the confusion matrix of the model and the computed accuracy of the model predictions using the training set.

```
# prediction on training data
pred <- titanic_train %>%
  select(survived) %>%
  bind_cols(
    predict(mod_tree,
            new_data = titanic_train,
            type = "class")) %>%
  rename(survived_tree = .pred_class)

# create confusion matrix
conf_matrix <- conf_mat(pred, truth = survived, estimate = survived_tree)
autoplot(conf_matrix) +
  geom_label(
    aes(
      x = (xmax + xmin) / 2,
      y = (ymax + ymin) / 2,
      label = c("TN", "FP", "FN", "TP")
    )
  )
```




```
#find the error rate
mod_tree_accuracy <- accuracy(pred, survived, survived_tree)
training_error <- 1 - mod_tree_accuracy[3]
print(training_error)
```

```
##   .estimate
## 1 0.1830144
```

1. **Testing the decision tree model.** Using the above example code, create a confusion matrix and compute the accuracy of the model using the test set. Note that the testing set are with data points that the model has not seen before. What are your thoughts on the accuracy of the testing set predictions vs the training set predictions?

```
### [BEGIN] - WORK ON YOUR MODEL EVALUATION HERE

### [END] - WORK ON YOUR MODEL EVALUATION HERE
```

2. **Visualizing the Titanic data set.** Visualize the Titanic data set using all variables so that you can compare the training set and testing set. Provide a paragraph of your observations.

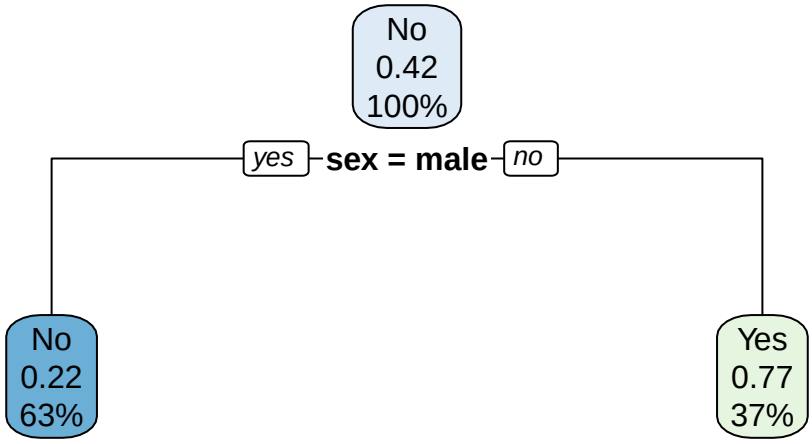
```
### [BEGIN] - WORK ON YOUR DATA WRANGLING AND GGPLOT PIPELINE HERE

### [END] - WORK ON YOUR DATA WRANGLING AND GGPLOT PIPELINE HERE
```

3. **(BONUS) Pruning to the best fit decision tree model.** Use the series of R code blocks below to find a best decision tree model that maximizes the accuracy of the model.

```
### [BEGIN] - WORK ON DATA WRANGLING AND MODIFY MODEL HERE
# pruning to best fit the decision tree and plot the pruned tree
mod_tree_p <- decision_tree(mode = "classification") %>%
  set_engine("rpart", control = rpart.control(cp = 0.36526)) %>%
  fit(form, data = titanic_train)
### [END] - WORK ON DATA WRANGLING AND MODIFY MODEL HERE
```

```
rpart.plot(mod_tree_p$fit, roundint=FALSE)
```



Using the above code, use values from 0 to 1 of the `cp` argument in the `rpart.control()` function and plot the resulting accuracy. Describe your observations. Hint: Use a for-loop.

Below uses a random forest method to a classification model. Try different values of the `mtry` and `trees` parameters within the `rand_forest` function. What happens to the error rate of the model?

```
### [BEGIN] - MODIFY R CODE HERE
# use random forest method
titanic_rf <- rand_forest(
  mode = "classification",
  mtry = 2,
  trees = 100) %>%
  set_engine("randomForest") %>%
  fit(form, data = titanic_train)

# make predictions
pred <- pred %>%
  bind_cols(
    predict(titanic_rf, new_data = titanic_train, type = "class")
  ) %>%
  rename(survived_rf = .pred_class)

# create confusion matrix
conf_mat(pred, survived, survived_rf)
```

```
##           Truth
## Prediction  No Yes
##        No  470  78
##        Yes  14 274
```

```r
# check accuracy
rf_accuracy <- accuracy(pred, survived, survived_rf)
tree_training_error<- 1 - accuracy(pred, survived, survived_rf)[3]
### [BEGIN] - MODIFY R CODE HERE
print(tree_training_error)
```

```
##   .estimate
## 1 0.1100478
```