Predicting Passenger Survival of the Titanic Part 2

```
#load packages
library(tidyverse)
library(tidymodels)
library(discrim)
set.seed(202204) # set seed for reproducibility
```

In this mini-assignment, we continue our quest on understanding categorized data using non-regression classifiers. Today, we focus on one of the simplest classification models called the Naive Bayes.[1]

Using the `titanic` data set. Our goal is to understand the data utilizing the Naive Bayes - 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))
titanic <- titanic %>%
  filter(!embarked == "?") %>%
  mutate(embarked = factor(embarked))
```

Below we define the model equation in terms of variables.

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

```
# 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()
```

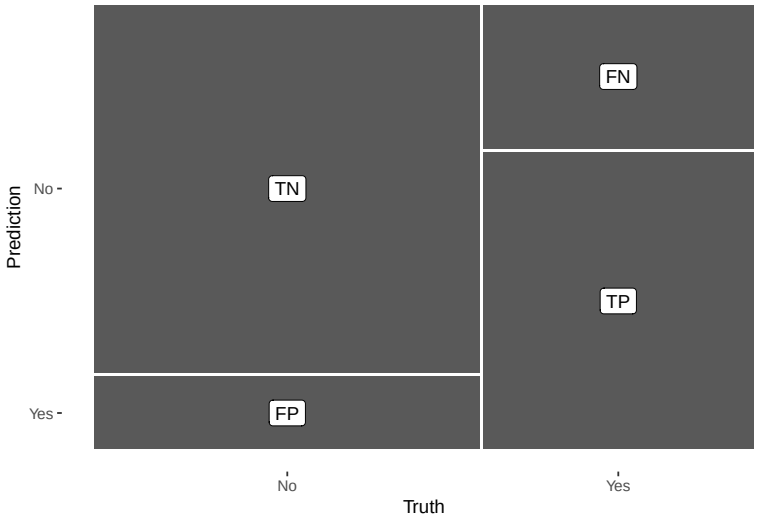
[1] Credit to Simon Ahn for this mini-assignment.

Naive Bayes

We are using a naive bayes classifier to predict passenger survival. The major assumption about the data when using the naive bayes is that all input variables are linearly independent, meaning a variable is unrelated to the presence of any other variable. This is why it's called "naive".

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

# create confusion matrix
conf_matrix <- conf_mat(pred, truth = survived, estimate = survived_nb)
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_nb_accuracy <- accuracy(pred, survived, survived_nb)
training_error <- 1 - mod_nb_accuracy[3]
print(training_error)
```

```
##   .estimate
## 1 0.2314149
```

You will notice that the naive bayes classifier performed a little bit worse than last time when we use the decision tree model. Here, we attempt to understand why.

1. **Which variables are dependent?** Use the `table()` function to create pairwise variable comparisons. If a variable is shown as numerical, use the `case_when()` function to convert it into a categorical variable with 3-4 levels. Since the `titanic` data has six variables, then you should end up with 15 tables. Use the chi-square test of independence to check each pair of variables whether they are independent or not. You can use the `chisq.test()` function to compute the p-value for each table. Finally, create a visualization where the x-axis is the variable pairs, and the y-axis is the p-values. Hint: Use a for-loop to achieve the first part and a bar plot for the visualization. Provide an explanation on why the naive bayes classifier performed worse than the decision tree model. Do you think adding more variable makes the model better?

```
### [BEGIN] - WORK ON YOUR DATA WRANGLING AND GGPLOT PIPELINE HERE
variable_names <- names(titanic) # get names of variables
all_pairs <- combn(variable_names,2) # get all pairwise combinations
### [END] - WORK ON YOUR DATA WRANGLING AND GGPLOT PIPELINE HERE
```

2. **Visualizing the Titanic data in a different way.** Create a visualization using the `geom_count()` function that shows the y-axis as the `survived` variable with all variables in the data. You may need to convert the numerical variables into a 3-4 level variable using the `case_when()` function. Use the parameter `position = position_jitter(width = 0, height = 0.1)` within the `geom_count()` function. Adjust the labels, scales, and colors accordingly.

```
### [BEGIN] - WORK ON YOUR GGPLOT PIPELINE HERE
p <- ggplot(data = titanic, aes(x = fare, y = survived)) +
  geom_count(
    aes(color = sex, shape = embarked)
  ) +
  facet_wrap(~pclass)
### [END] - WORK ON YOUR GGPLOT PIPELINE HERE
p
```