Homework 3

Jules Merigot (8488256)

October 26, 2022

```
# loading the data
titanic_data <- read.csv(file = "C:/Users/jules/OneDrive/Desktop/homework-3/data/titanic.csv")
head(titanic_data)

titanic_data$survived <- factor(titanic_data$survived, labels = c("Yes", "No"))
titanic_data$pclass <- factor(titanic_data$pclass)

str(titanic_data)
```

Classification

Question 1

```
# setting the seed
set.seed(8488)

titanic_split <- initial_split(titanic_data, prop=0.60, strata=survived)

titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
```

For splitting the data, I chose a proportion of 0.60 because it allows for more training data, while retaining enough data to be tested since there is a limited amount of observations. The training data has 534 observations while the testing data has 357 observations.

```
#plot of missing values in the training data
missing_plot(titanic_train)
```


```
# the number of missing values in the training data
sum(is.na(titanic_train))
```

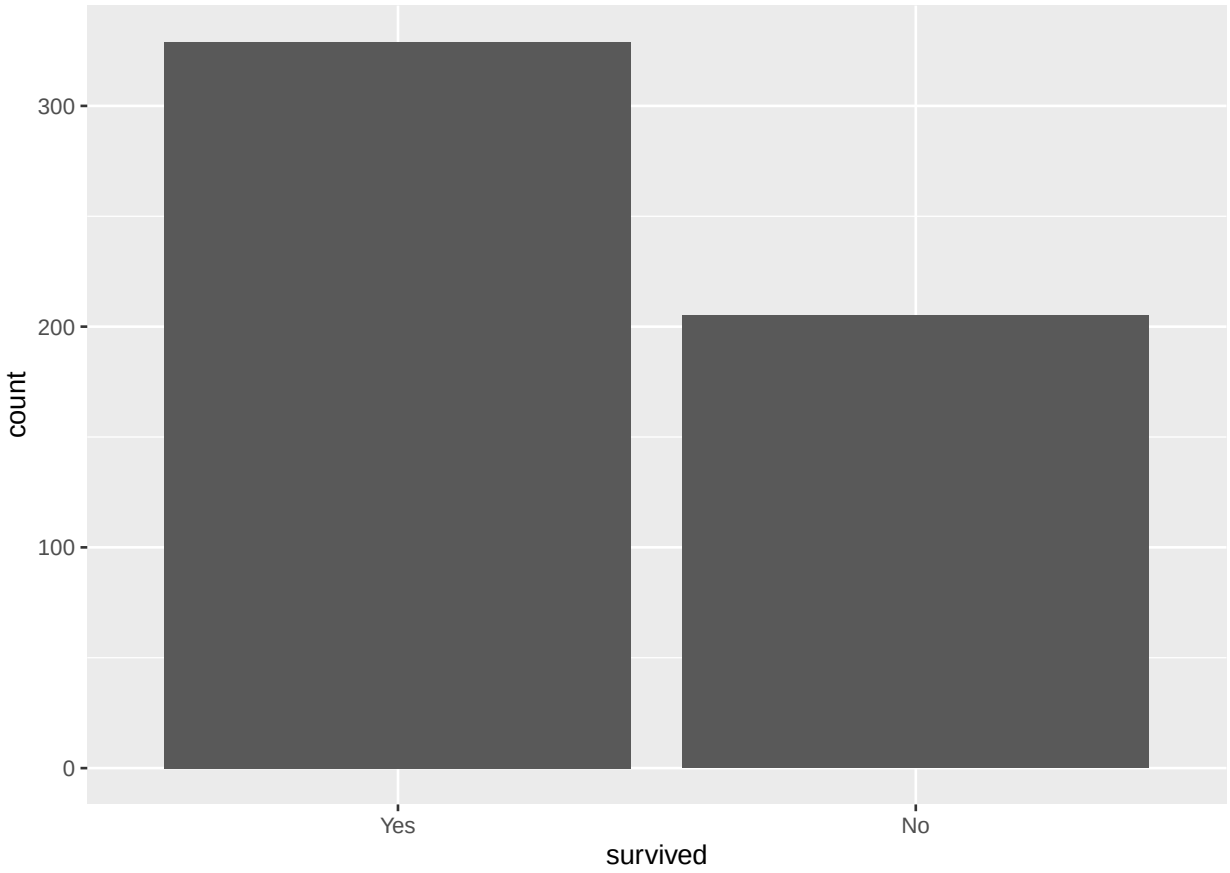
```
## [1] 530
```

There is a good amount of missing data in the training data, 530 missing data values to be exact, particularly for the *age* and *cabin* variables, as can be seen in the plot above.

We want to use stratified sampling for this data because not only does it allow for less bias, but since we have less observations than the abalone dataset for example, stratified sampling allows for more precision on a smaller dataset, and thus a more precise sample in this case.

Question 2

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



Using the above visualization of the distribution of the outcome variable *survived*, we can see that less people survived than people that perished on the Titanic. More than 300 (known) passengers lost their lives, while only a little more than 200 passengers survived.

Question 3

```
cor_titanic <- titanic_train %>%
  select(where(is.numeric)) %>%
  correlate()
```

```
## Correlation computed with
## * Method: 'pearson'
## * Missing treated using: 'pairwise.complete.obs'
```

```
rplot(cor_titanic)
```

After making the correlation matrix above, there are some clear patterns that emerge, such as most variables being slightly negatively correlated with others, with some exceptions. *parch* and *sib_sp* have a positive correlation, which means that the number of siblings/spouses of a certain passenger is positively correlated with the number of children/parents of that passenger, which makes sense. Additionally, *sib_sp* and *age* are negatively correlated, which indicates that a passenger's age is negatively correlated with the number of siblings/spouses they have. This makes sense because younger passengers will tend to travel alone, and thus are less likely to have siblings or spouses.

Question 4

```
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 + age:fare)

titanic_recipe %>% prep() %>% juice()
```

```
## # A tibble: 534 x 10
##      age sib_sp parch  fare survived pclass_X2 pclass_X3 sex_m~1 sex_m~2 fare_~3
##    <dbl>  <int> <int> <dbl> <fct>        <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
##  1  35        1     0 53.1  No               0         0       0     0     1858.
##  2  27        0     2 11.1  No               0         1       0     0      301.
##  3  14        1     0 30.1  No               1         0       0     0      421.
```

```
##  4  32.2      0     0 13    No               1         0       1    13      419.
##  5  34        0     0 13    No               1         0       1    13      442
##  6  28        0     0 35.5  No               0         0       1    35.5    994
##  7  25.8      0     0  7.75 No               0         1       0     0      200.
##  8  14        1     0 11.2  No               0         1       0     0      157.
##  9   3        1     2 41.6  No               1         0       0     0      125.
## 10  19        0     0  7.88 No               0         1       0     0      150.
## # ... with 524 more rows, and abbreviated variable names 1: sex_male,
## #   2: sex_male_x_fare, 3: fare_x_age
```

Question 5

```
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)

log_fit %>%
  tidy()
```

```
## # A tibble: 10 x 5
##    term             estimate std.error statistic  p.value
##    <chr>               <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)      5.23      0.786        6.65  2.87e-11
##  2 age             -0.0647    0.0154      -4.19  2.75e- 5
##  3 sib_sp          -0.477     0.147       -3.24  1.19e- 3
##  4 parch           -0.0180    0.166       -0.108 9.14e- 1
##  5 fare            -0.00363   0.0123      -0.295 7.68e- 1
##  6 pclass_X2       -1.34      0.433       -3.10  1.93e- 3
##  7 pclass_X3       -2.97      0.466       -6.38  1.81e-10
##  8 sex_male        -2.66      0.337       -7.88  3.36e-15
##  9 sex_male_x_fare -0.0119    0.00819     -1.46  1.45e- 1
## 10 fare_x_age       0.000259  0.000290     0.895 3.71e- 1
```

Question 6

```
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

```
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

```
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

We now generate predictions of each of the four models using our training data, and we use the accuracy metric to asses their performance.

```
library(yardstick)

log_predict <- predict(log_fit, titanic_train) %>%
  bind_cols(titanic_train$survived) %>%
  accuracy(truth = titanic_train$survived, estimate = .pred_class)
```

```
## New names:
## * `` -> `...2`
```

```
log_predict

lda_predict <- predict(lda_fit, titanic_train) %>%
  bind_cols(titanic_train$survived) %>%
  accuracy(truth = titanic_train$survived, estimate = .pred_class)
```

```
## New names:
## * `` -> `...2`
```

```
lda_predict

qda_predict <- predict(qda_fit, titanic_train) %>%
  bind_cols(titanic_train$survived) %>%
  accuracy(truth = titanic_train$survived, estimate = .pred_class)
```

```
## New names:
## * `` -> `...2`
```

```
qda_predict

nb_predict <- predict(nb_fit, titanic_train) %>%
  bind_cols(titanic_train$survived) %>%
  accuracy(truth = titanic_train$survived, estimate = .pred_class)
```

```
## New names:
## * `` -> `...2`
```

```
nb_predict
```

In order to compare the predictions and discover which model achieved the highest accuracy on the training data, we can make a table of the accuracy rates.

```
accuracies <- c(log_predict$.estimate, lda_predict$.estimate,
                qda_predict$.estimate, nb_predict$.estimate)
models <- c("Logistic Regression", "LDA", "QDA", "Naive Bayes")
results <- tibble(accuracies = accuracies, models = models)
results %>%
  arrange(-accuracies)
```

```
## # A tibble: 4 x 2
##   accuracies models
##        <dbl> <chr>
## 1      0.831 Logistic Regression
## 2      0.824 QDA
## 3      0.816 LDA
## 4      0.813 Naive Bayes
```

As can be seen in the table above, the logistic regression model achieved the highest accuracy on the training data with an accuracy of 83.15%.

Question 10