Homework 3

Contents

Classification

```
library(tidyverse)
library(tidymodels)
library(ggplot2)
library(corrplot)
library(corrr)
library(ggthemes)
library(discrim)
library(poissonreg)
library(klaR) # for naive bayes
tidymodels_prefer()
```

```
titanic <- read.csv("homework-3/data/titanic.csv", stringsAsFactors = TRUE)
```

Question 1:

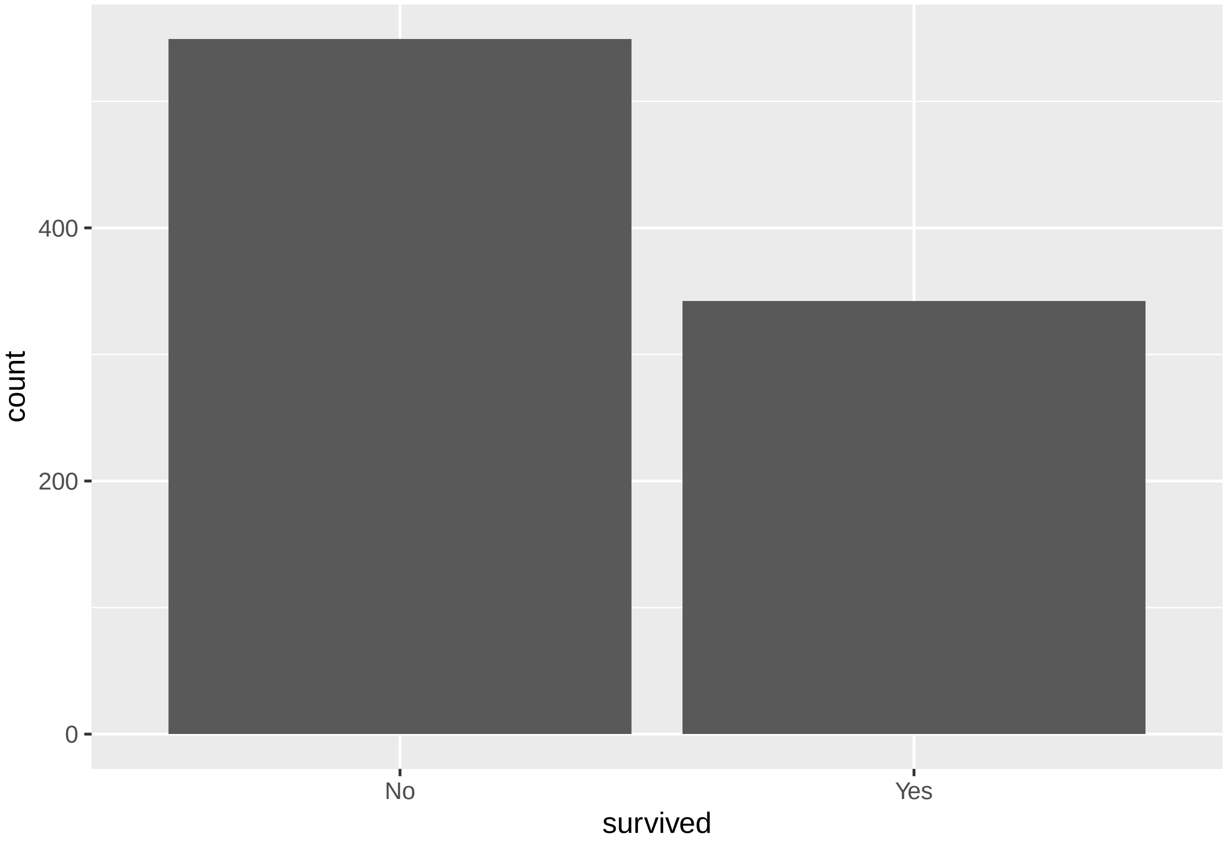
```
set.seed(0714)

titanic_split <- initial_split(titanic, prop=0.70,
                               strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
```

For the training data set, there are potential issues with the variable cabin, because there seems to be observations that have missing data. Likewise can be seen for the variable age as well. The other variable with two missing values in the observations is embarked.

It is a good idea to use stratified sampling, because it ensures that the population is being represented as accurately as possible, so if there are missing values, the stratified sampling should be able to represent the ratio of missing values as the ones seen in the population.

Question 2:

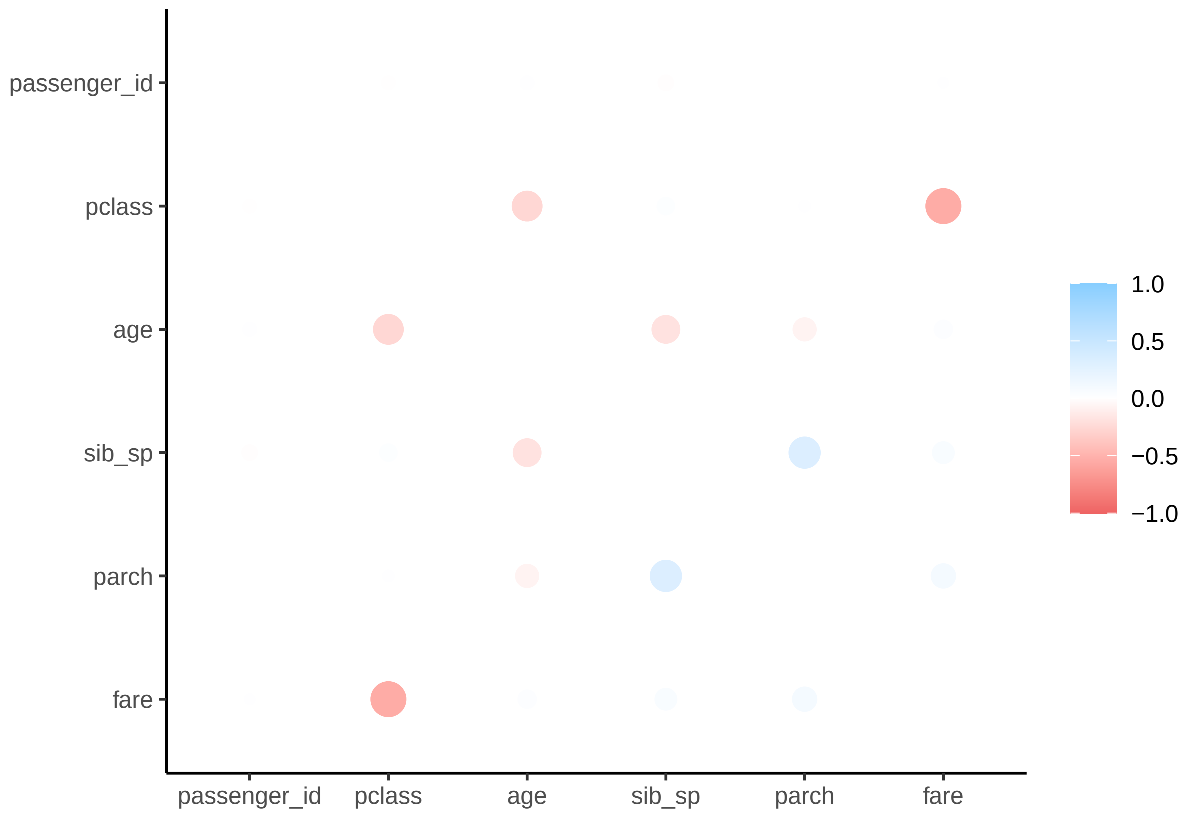
```
titanic %>%
  ggplot(aes(x = survived)) +
  geom_bar()
```



Based on the graph, it seems like there are more people that did not survive than those that did. Because the outcome is only yes or no, the distribution follows a Bernoulli distribution.

Question 3:

```
#Correlation
cor_titanic <- titanic %>%
  select(where(is.numeric)) %>%
  correlate()
rplot(cor_titanic)
```



There are a couple of variables that are correlated with each other. The following are negatively correlated with each other:

- pclass and fare
- pclass and age
- sib_sp and age
- parch and age

The following are positively correlated with each other:

- sib_sp and parch
- fare and parch

Question 4:

```
titanic_recipe <- recipe(survived ~ pclass + sex + age + sib_sp
                         + parch + fare, data = titanic_train) %>%
  step_impute_linear(age) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(~ starts_with("sex"):fare) %>%
  step_interact(~ age:fare)
```

Question 5:

```
#Logistic Regression Model
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)
```

Question 6:

```
#Linear Discriminant Analysis Model
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:

```
#Quadratic Discriminant Analysis Model
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:

```
#Naive-Bayes Model
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:

```
#Logistic Regression Performance
log_regt_perf <- predict(log_fit, new_data = titanic_train, type = "prob")

log_regt_perf <- bind_cols(log_regt_perf, titanic_train)

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


#LDA Performance
ldat_perf <- predict(lda_fit, new_data = titanic_train, type = "prob")

ldat_perf <- bind_cols(ldat_perf, titanic_train)

lda_acc <- augment(lda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)


#QDA Performance
qdat_perf <- predict(qda_fit, new_data = titanic_train, type = "prob")

qdat_perf <- bind_cols(qdat_perf, titanic_train)

qda_acc <- augment(qda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)


#NB Performance
nbt_perf <- predict(nb_fit, new_data = titanic_train, type = "prob")

nbt_perf <- bind_cols(nbt_perf, titanic_train)

nb_acc <- augment(nb_fit, new_data = titanic_train)%>%
  accuracy(truth = survived, estimate = .pred_class)

#Comparing Model Perfomance
accuracies <- c(log_reg_acc$.estimate, lda_acc$.estimate,
                nb_acc$.estimate, qda_acc$.estimate)
models <- c("Logistic Regression", "LDA", "Naive Bayes", "QDA")
results <- tibble(accuracies = accuracies, models = models)
results %>%
  arrange(-accuracies)
```

```
## # A tibble: 4 x 2
##   accuracies models
##        <dbl> <chr>
## 1      0.795 Logistic Regression
## 2      0.788 LDA
## 3      0.783 QDA
## 4      0.762 Naive Bayes
```

The model that achieved the highest accuracy was the logistic regression model with 79.5%

Question 10:

```
#Fit into testing data
predict(log_fit, new_data = titanic_test, type = "prob")
```

```
## # A tibble: 268 x 2
##    .pred_No .pred_Yes
##       <dbl>     <dbl>
##  1   0.0895     0.911
##  2   0.885      0.115
##  3   0.358      0.642
##  4   0.769      0.231
##  5   0.442      0.558
##  6   0.465      0.535
##  7   0.885      0.115
##  8   0.563      0.437
##  9   0.0526     0.947
## 10   0.547      0.453
## # ... with 258 more rows
```

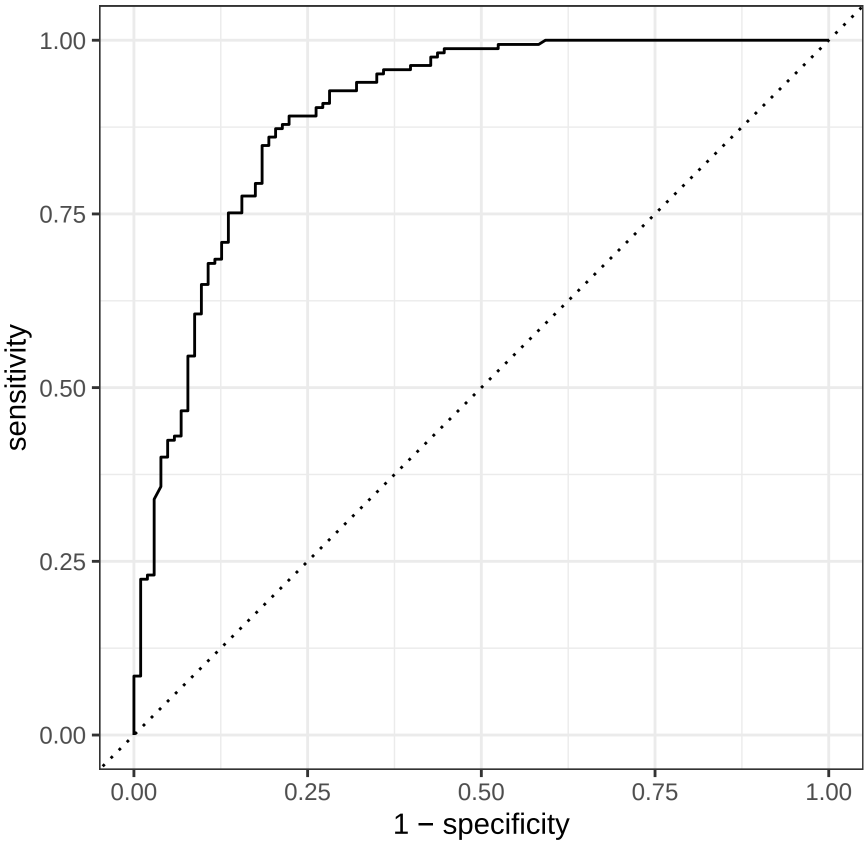
```
#Testing Accuracy
multi_metric <- metric_set(accuracy, sensitivity, specificity)
augment(log_fit, new_data = titanic_test) %>%
  multi_metric(truth = survived, estimate = .pred_class)
```

```
## # A tibble: 3 x 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 accuracy    binary         0.836
## 2 sensitivity binary         0.909
## 3 specificity binary         0.718
```

```
#Confusion matrix on the testing data
augment(log_fit, new_data = titanic_test) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction  No Yes
##        No  150  29
##        Yes  15  74
```

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



```r
#Find the AUC
library(pROC)
library(yardstick)
augment(log_fit, new_data = titanic_test) %>%
  roc_auc(survived, .pred_No)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.895
```

The model performed rather well. The testing accuracy score was 83.6%, while the training accuracy score was 79.5%, so it seemed to fit the testing data better than the training data. These values might differ because there are less observations in the testing dataset than the training dataset.