Pstat231HW3

Zihao Yang

2022-04-18

```r
#install.packages("tidyverse")
#install.packages("tidymodels")
#install.packages("ISLR")
#install.packages("corrr")
#install.packages("discrim")
#install.packages("poissonreg")
#install.packages("klaR")
tinytex::install_tinytex
library(tidyverse)
library(tidymodels)
library(ISLR)
library(ggplot2)
library(corrplot)
library(ggthemes)
library(yardstick)
library(dplyr)
library(magrittr)
library(corrr)
library(discrim)
library(poissonreg)
library(klaR)
tidymodels_prefer()
set.seed(100)
```

```r
# Get the dataset
tt <- read.csv("titanic.csv")
tt$survived <- factor(tt$survived,levels = c("Yes", "No"))
tt$pclass <- as.factor(tt$pclass)
head(tt)
```

```
##   passenger_id survived pclass
## 1            1       No      3
## 2            2      Yes      1
## 3            3      Yes      3
## 4            4      Yes      1
## 5            5       No      3
## 6            6       No      3
##                                                  name    sex age sib_sp parch
## 1                             Braund, Mr. Owen Harris   male  22      1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38      1     0
## 3                              Heikkinen, Miss. Laina female  26      0     0
```

```
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35      1     0
## 5                             Allen, Mr. William Henry   male  35      0     0
## 6                                     Moran, Mr. James   male  NA      0     0
##            ticket    fare cabin embarked
## 1       A/5 21171  7.2500  <NA>        S
## 2        PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250  <NA>        S
## 4          113803 53.1000  C123        S
## 5          373450  8.0500  <NA>        S
## 6          330877  8.4583  <NA>        Q
```

Q1

```r
#Split the data and check the the observations
tt_split <- initial_split(tt, prop = 0.80,
                          strata = survived)
tt_train <- training(tt_split)
tt_test <- testing(tt_split)
c(nrow(tt_train),nrow(tt_test),nrow(tt))
```

```
## [1] 712 179 891
```

```r
712/891
```

```
## [1] 0.79910213
```

```r
179/891
```

```
## [1] 0.20089787
```

There are about 80% observations in the training set and 20% observations in the testing set, which is the same proportion as we split in our function.

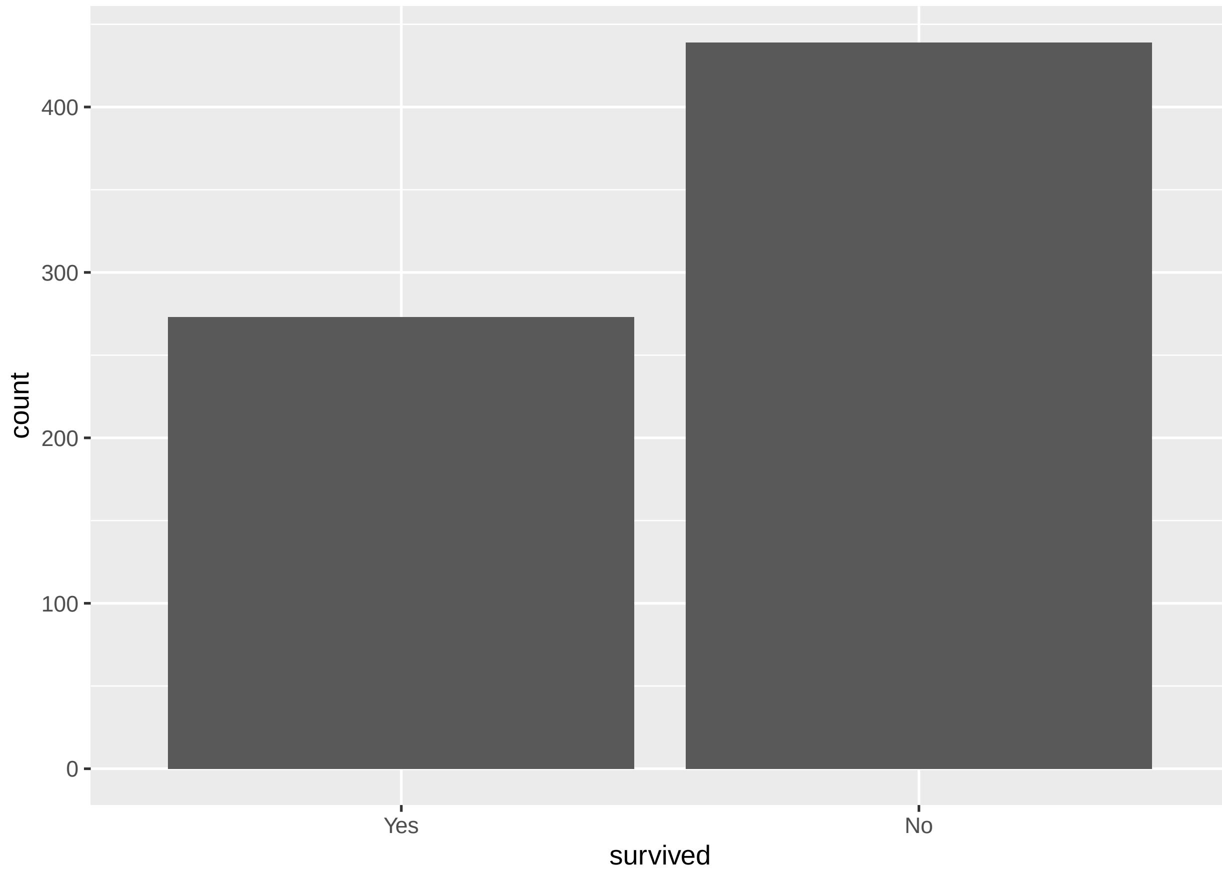
```r
#check the missing value
sapply(tt_train, function(x) sum(is.na(x)))
```

```
## passenger_id     survived       pclass         name          sex          age
##            0            0            0            0            0          138
##       sib_sp        parch       ticket         fare        cabin     embarked
##            0            0            0            0          550            1
```

There are some missing values in the training data. Most of them are in the age and cabin columns. It is important to use the stratified sampling in this data, because it ensures that the number of data points in the training data is equivalent to the proportions in the original data set. We want to keep survive proportion for training data the same in original data.

Q2

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




```r
summary(tt_train$survived)
```

```
## Yes  No
## 273 439
```

```r
273/(439+273)
```

```
## [1] 0.38342697
```

According to the bar plot output, the number of not survived is obviously more than the number of survived. About 38% people survived and 62% not survived.

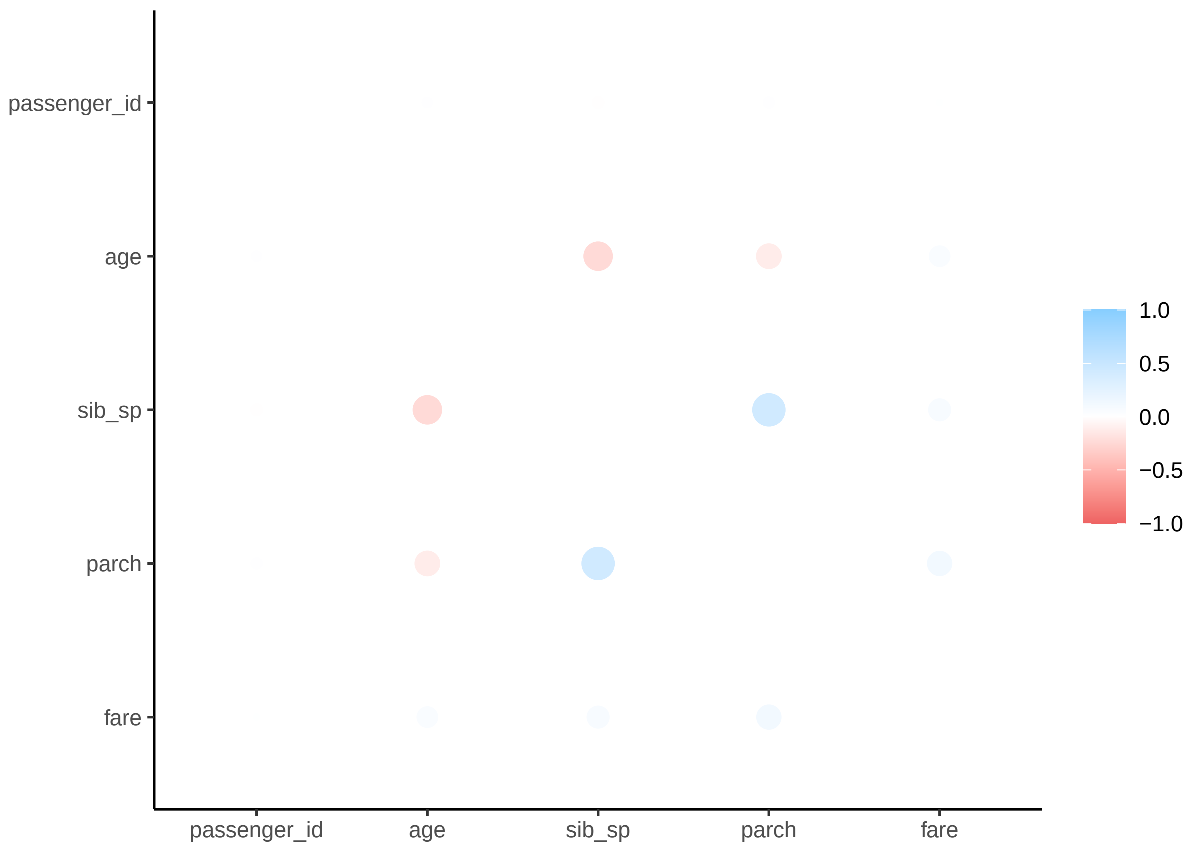
Q3

```r
cor_tt <- tt_train[,sapply(tt_train,is.numeric)] %>%
  correlate()
```

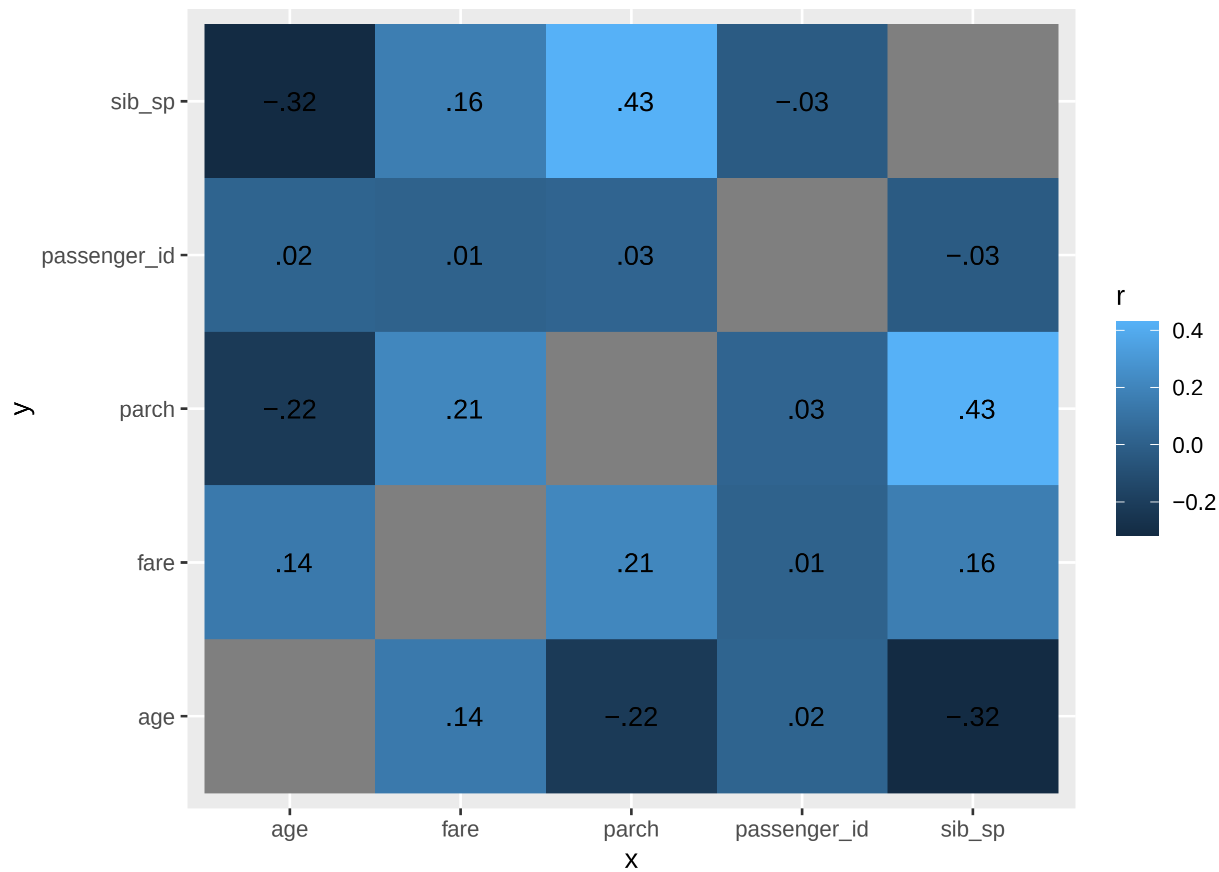
```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```

```r
rplot(cor_tt)
```

```
## Don't know how to automatically pick scale for object of type noquote. Defaulting to continuous.
```




```r
cor_tt %>%
  stretch() %>%
  ggplot(aes(x, y, fill = r)) +
  geom_tile() +
  geom_text(aes(label = as.character(fashion(r))))
```



According to the output. Age is negatively correlated with number of siblings and spouses aboard and number of parents and children aboard. The number of parents and children aboard is positively correlated with number of siblings and spouses aboard and with passenger fare. The fare is also positively correlated with number of siblings and spouses aboard. The rest are weakly correlated or uncorrelated.

Q4

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

Q5

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

log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(tt_recipe)
```

```
log_fit <- fit(log_wkflow, tt_train)
```

Q6

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

lda_wkflow <- workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(tt_recipe)

lda_fit <- fit(lda_wkflow, tt_train)
```

Q7

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

qda_wkflow <- workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(tt_recipe)

qda_fit <- fit(qda_wkflow, tt_train)
```

Q8

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

nb_wkflow <- workflow() %>%
  add_model(nb_mod) %>%
  add_recipe(tt_recipe)

nb_fit <- fit(nb_wkflow, tt_train)
```

Q9

```
log_predict <- predict(log_fit, new_data = tt_train, type = "class")

lda_predict <- predict(lda_fit, new_data = tt_train, type = "class")
```

```r
qda_predict <- predict(qda_fit, new_data = tt_train, type = "class")

nb_predict <- predict(nb_fit, new_data = tt_train, type = "class")


tt_train_predict <- bind_cols(log_predict, lda_predict, qda_predict, nb_predict, tt_train$survived)


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

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

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

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


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      0.819 Logistic Regression
## 2      0.796 LDA
## 3      0.775 Naive Bayes
## 4      0.774 QDA
```

According to the output, Logistic Regression achieved the highest accuracy on the training data.

Q10

```r
predict(log_fit, new_data = tt_test, type = "class")
```

```
## # A tibble: 179 x 1
##    .pred_class
##    <fct>
##  1 No
##  2 Yes
##  3 No
```

```
##   4 No
##   5 No
##   6 No
##   7 Yes
##   8 No
##   9 No
##  10 Yes
## # ... with 169 more rows
```

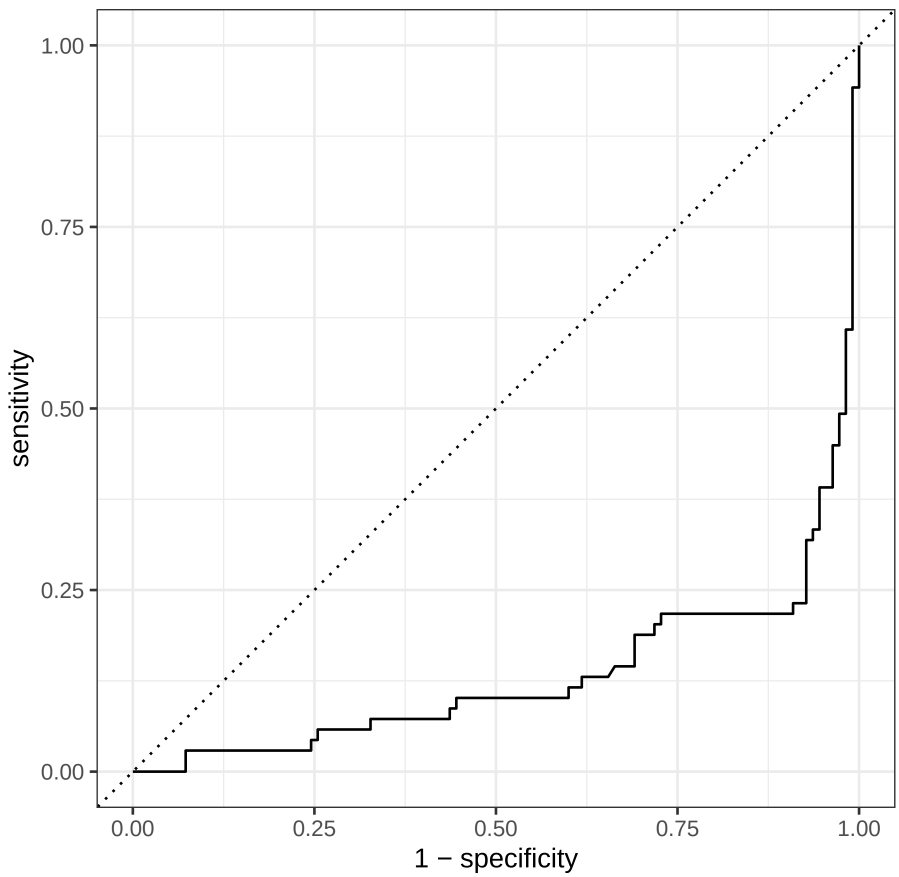
```
multi_metric <- metric_set(accuracy, sensitivity, specificity)

augment(log_fit, new_data = tt_test) %>%
  multi_metric(truth = survived, estimate = .pred_class)
```

```
## # A tibble: 3 x 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 accuracy    binary         0.860
## 2 sensitivity binary         0.754
## 3 specificity binary         0.927
```

```
augment(log_fit, new_data = tt_test) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction Yes  No
##        Yes  52   8
##        No   17 102
```

```
augment(log_fit, new_data = tt_test) %>%
  roc_curve(survived, .pred_No) %>%
  autoplot()
```




```
augment(log_fit, new_data = tt_test) %>%
  roc_auc(survived, .pred_Yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.868
```

The accuracy of the model on the testing data is approximately 86.75%. Thus, the model fits pretty well on the tests data.

Both the accuracy of training and testing are above the 80%, but the accuracy of testing data is little bit higher than the training one. It may be caused by the smaller sample size of the testing data.

231 part

Q11

Given that

$$p(z) = \left(\frac{e^z}{1+e^z} \right)$$

we have

$$
\begin{aligned}
p(z) + p(z)e^z &= e^z \\
p(z) &= (1 - p(z))e^z \\
e^z &= \frac{p(z)}{1 - p(z)} \\
ln(e^z) &= ln\left(\frac{p(z)}{1 - p(z)}\right) \\
z &= ln\left(\frac{p(z)}{1 - p(z)}\right) \\
z(p) &= ln\left(\frac{p}{1 - p}\right)
\end{aligned}
$$

Q12

a) $$P(Z) = \frac{e^Z}{1+e^Z}$$

$$P(Z) + P(Z)\cdot e^Z = e^Z$$

$$\cancel{P(Z)(1+e^Z}$$

$$P(Z) = (1 - P(Z))\, e^Z$$

$$e^Z = \frac{P(Z)}{1-P(Z)}$$

$$\ln(e^Z) = \ln\left(\frac{P(Z)}{1-P(Z)}\right)$$

$$Z = \ln\left(\frac{P(Z)}{1-P(Z)}\right)$$

$$Z(p) = \ln\left(\frac{P}{1-P}\right) \quad \text{\#take inverse function}$$

b)
$$P(Z) = \frac{e^Z}{1+e^Z}, \quad Z = \beta_0 + \beta_1 X_1$$

$$P = \text{logistic}(Z)$$

odd: $$\frac{P}{1-P} = e^Z = e^{\beta_0 + \beta_1 X_1}$$

Change X_1 into $X_1 + 2$

then $$e^{\beta_0 + \beta_1 (X_1 + 2)} = e^{\beta_0 + \beta_1 X_1} \cdot e^{2\beta_1}$$

(~~tot~~ continue)

Let β_1 be negative

As $x_1 \to \infty$

$$\lim_{x_1 \to \infty} p(Z) = \lim_{x_1 \to \infty} \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}}$$

$$= \frac{e^{\beta_0} \cdot \lim_{x_1 \to \infty} e^{\beta_1 x_1}}{1 + e^{\beta_0} \cdot \lim_{x_1 \to \infty} e^{\beta_1 x_1}}$$

$$= \frac{e^{\beta_0} \cdot 0}{1 + e^{\beta_0} \cdot 0}$$

$$= 0$$

As $x_1 \to -\infty$

$$\lim_{x_1 \to -\infty} p(Z) = \lim_{x_1 \to -\infty} \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}}$$

$$= \lim_{x_1 \to \infty} \frac{1}{\frac{1}{e^{\beta_0 + \beta_1 x_1}} + 1}$$

$$= \frac{1}{\frac{1}{e^{\beta_0} \cdot \lim_{x_1 \to -\infty} e^{\beta_1 x_1}} + 1}$$

$$= \frac{1}{\frac{1}{\infty} + 1}$$

$$= \frac{1}{0 + 1}$$

$$= 1$$