Homework 3

PSTAT 131/231

Classification

For this assignment, we will be working with part of a Kaggle data set that was the subject of a machine learning competition and is often used for practicing ML models. The goal is classification; specifically, to predict which passengers would survive the Titanic shipwreck.

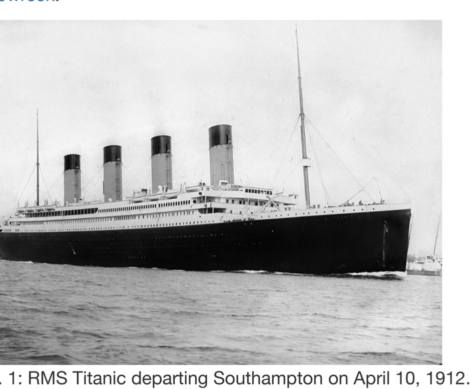


Fig. 1: RMS Titanic departing Southampton on April 10, 1912.

Load the data from `data/titanic.csv` into *R* and familiarize yourself with the variables it contains using the codebook (`data/titanic_codebook.txt`).

Notice that `survived` and `pclass` should be changed to factors. When changing `survived` to a factor, you may want to reorder the factor so that *"Yes"* is the first level.

Make sure you load the `tidyverse` and `tidymodels`!

Remember that you'll need to set a seed at the beginning of the document to reproduce your results.

```
library(tidymodels)
library(ISLR) # For the Smarket data set
library(ISLR2) # For the Bikeshare data set
library(discrim)
library(poissonreg)
library(corrr)
library(klaR) # for naive bayes
tidymodels_prefer()
setwd("~/Desktop/PSTAT 131/pstat131-hw3")
set.seed(3435)
```

```
titanic <- read.csv("data/titanic.csv")
titanic$survived <- as.factor(titanic$survived)
titanic$pclass <- as.factor(titanic$pclass)
titanic$survived <- factor(titanic$survived, levels = c("Yes","No"))
titanic %>% head()
```

```
##   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
```

Question 1

Split the data, stratifying on the outcome variable, `survived.` You should choose the proportions to split the data into. Verify that the training and testing data sets have the appropriate number of observations. Take a look at the training data and note any potential issues, such as missing data.

Why is it a good idea to use stratified sampling for this data?

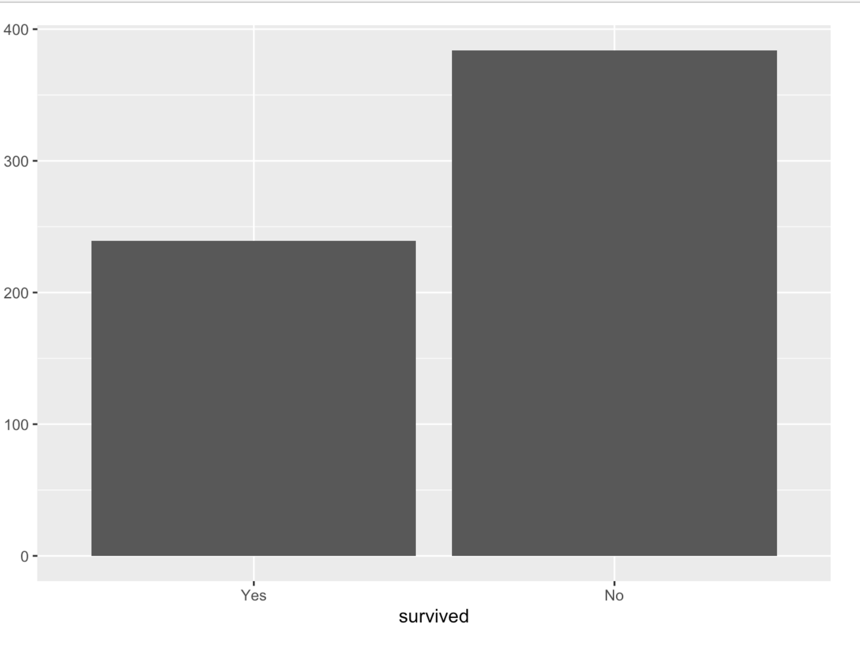
we use stratified sampling because it enables us to obtain a sample population that best represents the entire population being studied

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

Question 2

Using the **training** data set, explore/describe the distribution of the outcome variable `survived`.

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

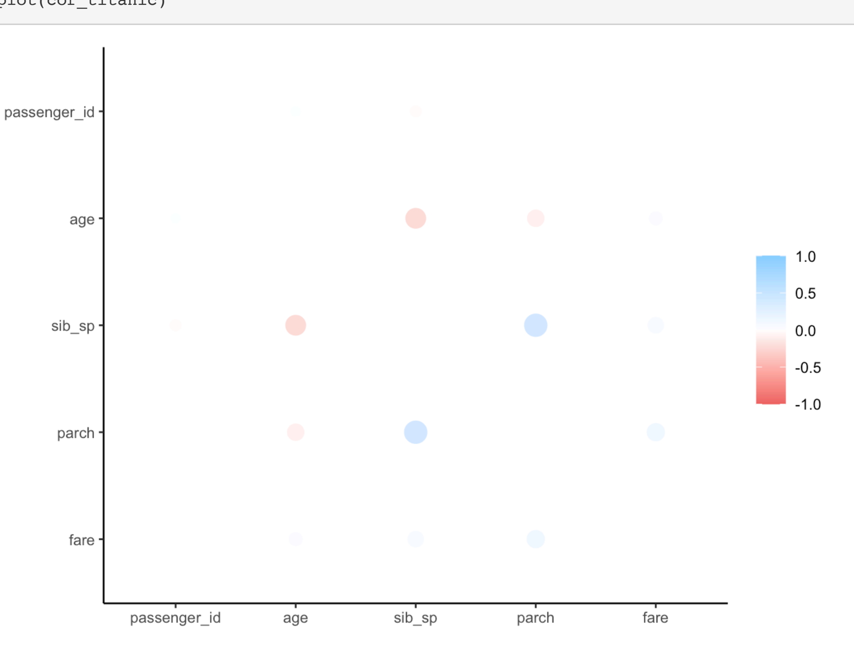


we can see from the bar plot that most of the people did not survive from the accident

Question 3

Using the **training** data set, create a correlation matrix of all continuous variables. Create a visualization of the matrix, and describe any patterns you see. Are any predictors correlated with each other? Which ones, and in which direction?

```
library("dplyr")
cor_titanic <- select_if(titanic,is.numeric) %>% correlate()
rplot(cor_titanic)
```



From the correlation matrx, I can see that sib_sp seems tonegatively corelate with age, and parch is positively corelate with sib_sp.

Question 4

Using the **training** data, create a recipe predicting the outcome variable `survived`. Include the following predictors: ticket class, sex, age, number of siblings or spouses aboard, number of parents or children aboard, and passenger fare.

Recall that there were missing values for `age`. To deal with this, add an imputation step using `step_impute_linear()`. Next, use `step_dummy()` to **dummy** encode categorical predictors. Finally, include interactions between:

- Sex and passenger fare, and
- Age and passenger fare.

You'll need to investigate the `tidymodels` documentation to find the appropriate step functions to use.

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

Question 5

Specify a **logistic regression** model for classification using the `"glm"` engine. Then create a workflow. Add your model and the appropriate recipe. Finally, use `fit()` to apply your workflow to the **training** data.

```
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 × 5
##    term              estimate std.error statistic  p.value
##    <chr>                <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)      -4.40      0.683       -6.45  1.13e-10
##  2 age               0.0629    0.0136       4.62  3.77e- 6
##  3 sib_sp            0.437     0.132        3.30  9.52e- 4
##  4 parch             0.151     0.153        0.989 3.23e- 1
##  5 fare             -0.00116   0.0107      -0.108 9.14e- 1
##  6 pclass_X2         1.25      0.363        3.46  5.48e- 4
##  7 pclass_X3         2.44      0.382        6.39  1.62e-10
##  8 sex_male          2.15      0.303        7.09  1.32e-12
##  9 sex_male_x_fare   0.0139    0.00836      1.66  9.65e- 2
## 10 fare_x_age       -0.000360  0.000206    -1.75  8.03e- 2
```

Hint: Make sure to store the results of `fit()`. You'll need them later on.

Question 6

Repeat Question 5, but this time specify a linear discriminant analysis model for classification using the `"MASS"` engine.

```
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

Repeat Question 5, but this time specify a quadratic discriminant analysis model for classification using the `"MASS"` engine.

```
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

Repeat Question 5, but this time specify a naive Bayes model for classification using the `"klaR"` engine. Set the `usekernel` argument to `FALSE`.

```
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

Now you've fit four different models to your training data.

Use `predict()` and `bind_cols()` to generate predictions using each of these 4 models and your **training** data. Then use the *accuracy* metric to assess the performance of each of the four models.

Which model achieved the highest accuracy on the training data?

```
bound_train_data = bind_cols(predict(log_fit, new_data = titanic_train, type = "prob"),
                             predict(lda_fit, new_data = titanic_train, type = "prob"),
                             predict(qda_fit, new_data = titanic_train, type = "prob"),
                             predict(nb_fit, new_data = titanic_train, type = "prob"),
                             titanic_train$survived)
colnames(bound_train_data) = c("Log fit", "LDA fit", "QDA fit",
                               "NB fit", "True")
```

The logistic model accuracy is

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

```
## # A tibble: 1 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.814
```

The linear discriminant analysis model accuracy is

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

```
## # A tibble: 1 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.793
```

The quadratic discriminant analysis model accuracy is

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

```
## # A tibble: 1 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.778
```

The native bayesian model has accuracy of

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

```
## # A tibble: 1 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.787
```

From the data above, we can see the the **logicstic model** achieve the highest accuracy of 0.814

Question 10

Fit the model with the highest training accuracy to the **testing** data. Report the accuracy of the model on the **testing** data.

```
predict(log_fit, new_data = titanic_test, type = "prob")
```

```
## # A tibble: 268 × 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1    0.933    0.0671
##  2    0.924    0.0763
##  3    0.119    0.881
##  4    0.183    0.817
##  5    0.230    0.770
##  6    0.239    0.761
##  7    0.120    0.880
##  8    0.119    0.881
##  9    0.0456   0.954
## 10    0.174    0.826
## # … with 258 more rows
```

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

augment(nb_fit, new_data = titanic_test) %>%
  multi_metric(truth = survived, estimate = .pred_class)
```

```
## # A tibble: 3 × 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 accuracy    binary         0.799
## 2 sensitivity binary         0.592
## 3 specificity binary         0.927
```

As we can see from the table, the **accuracy is 0.799 on the testing data**

Again using the **testing** data, create a confusion matrix and visualize it. Plot an ROC curve and calculate the area under it (AUC).

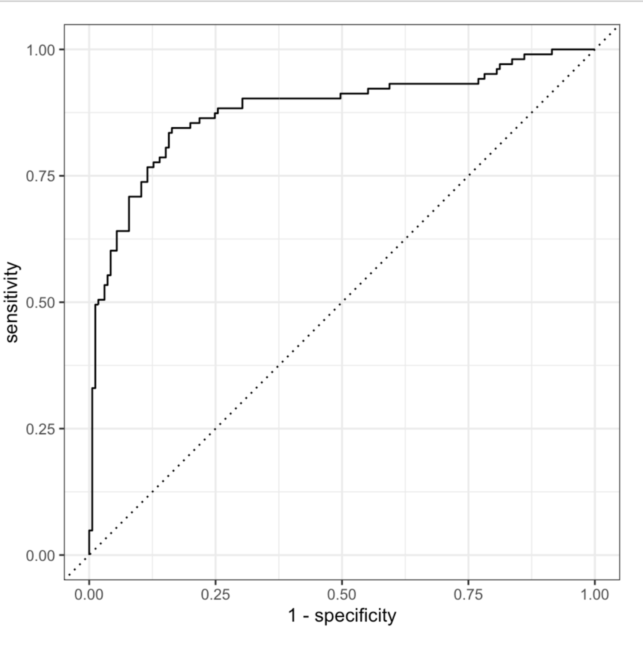
the confusion matrix is

```
augment(nb_fit, new_data =titanic_test) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction Yes  No
##        Yes  61  12
##        No   42 153
```

the ROC curve is

```
ROC <- augment(log_fit, new_data = titanic_test) %>%
  roc_curve(survived, .pred_Yes) %>%
  autoplot()
ROC
```




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

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.880
```

so the area under is 0.880

How did the model perform? Compare its training and testing accuracies. If the values differ, why do you think this is so?

As we can see from the ROC curve, the model does well in predicting the survival of titanic popultaion. The training accuracy and the testing accuracy is also close, so this is indeed a satisfactory model.

Required for 231 Students

In a binary classification problem, let p represent the probability of class label 1, which implies that $1-p$ represents the probability of class label 0. The *logistic function* (also called the "inverse logit") is the cumulative distribution function of the logistic distribution, which maps a real number *z* to the open interval $(0, 1)$.

Question 11

Given that:

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

Prove that the inverse of a logistic function is indeed the *logit* function:

$$z(p) = ln\left(\frac{p}{1-p}\right)$$

$$\begin{aligned}
\text{let } z = \text{logit}(p) &= \log\frac{p}{1-p} \\
e^z &= \frac{p}{1-p} \\
1+e^z &= \frac{1-p}{1-p} + \frac{p}{1-p} = \frac{1}{1-p} \\
\frac{1}{1+e^z} &= 1-p \\
p &= 1 - \frac{1}{1+e^z} = \frac{e^z}{1+e^z}
\end{aligned}$$

Question 12

Assume that $z = \beta_0 + \beta_1 x_1$ and $p = logistic(z)$. How do the odds of the outcome change if you increase x_1 by two? Demonstrate this.

We have

$$\log\left(\frac{p(z)}{1-p(z)}\right) = \beta_0 + \beta_1 z$$

We can see from this formula that, a two unit increase of x_1 at the right hand side increases x_1 by two unit changes the log odds by $2\beta_1$. Equivalently, it multiplies the odds by $e^{2\beta_1}$. However, because the relationship between p and z in the equation is not a straight line, β_1 does not correspond to the change in p associated with a two-unit increase in x_1. The amount that p changes due to a two-unit change in x_1 depends on the current value of x_1. But regardless of the value of z, if β_1 is positive then increasing x_1 will be associated with increasing p(X)

Assume now that β_1 is negative. What value does p approach as x_1 approaches ∞? What value does p approach as x_1 approaches $-\infty$?

If β_1 is negative then increasing x_1 will be associated with decreasing p. p approach 0 as x_1 approaches ∞, and p approach 1 as x_1 approaches $-\infty$.