PSTAT 131 Homework 2

Sunrise Gao

10/14/2022

Contents

Linear Regression

For this lab, we will be working with a data set from the UCI (University of California, Irvine) Machine Learning repository (see website here). The full data set consists of $4,177$ observations of abalone in Tasmania. (Fun fact: Tasmania supplies about 25% of the yearly world abalone harvest.)

The age of an abalone is typically determined by cutting the shell open and counting the number of rings with a microscope. The purpose of this data set is to determine whether abalone age (**number of rings + 1.5**) can be accurately predicted using other, easier-to-obtain information about the abalone.

The full abalone data set is located in the `\data` subdirectory. Read it into R using `read_csv()`. Take a moment to read through the codebook (`abalone_codebook.txt`) and familiarize yourself with the variable definitions.

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

```
library(ggplot2)
library(tidyverse)
library(tidymodels)
library(corrplot)
library(ggthemes)
tidymodels_prefer()
tidymodels_packages()
```

```
##  [1] "broom"        "cli"          "conflicted"   "dials"        "dplyr"
##  [6] "ggplot2"      "hardhat"      "infer"        "modeldata"    "parsnip"
## [11] "purrr"        "recipes"      "rlang"        "rsample"      "rstudioapi"
## [16] "tibble"       "tidyr"        "tune"         "workflows"    "workflowsets"
## [21] "yardstick"    "tidymodels"
```

```
abalone <- read.csv("abalone.csv")
head(abalone)
```

```
##   type longest_shell diameter height whole_weight shucked_weight viscera_weight
## 1    M         0.455    0.365  0.095       0.5140         0.2245         0.1010
## 2    M         0.350    0.265  0.090       0.2255         0.0995         0.0485
## 3    F         0.530    0.420  0.135       0.6770         0.2565         0.1415
## 4    M         0.440    0.365  0.125       0.5160         0.2155         0.1140
## 5    I         0.330    0.255  0.080       0.2050         0.0895         0.0395
## 6    I         0.425    0.300  0.095       0.3515         0.1410         0.0775
##   shell_weight rings
## 1        0.150    15
## 2        0.070     7
## 3        0.210     9
## 4        0.155    10
## 5        0.055     7
## 6        0.120     8
```

Question 1

Your goal is to predict abalone age, which is calculated as the number of rings plus 1.5. Notice there currently is no `age` variable in the data set. Add `age` to the data set.

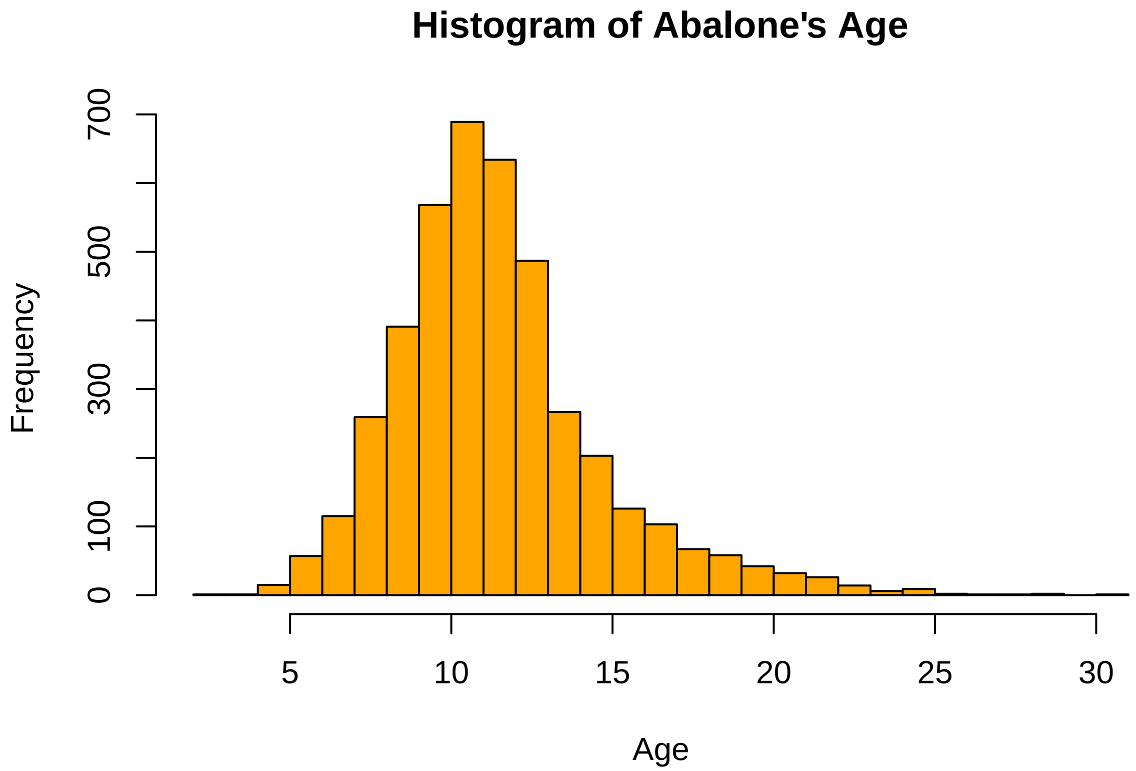
Assess and describe the distribution of `age`.

```r
abalone_new <- abalone %>%
  mutate(age = rings + 1.5)

head(abalone_new)
```

```
##   type longest_shell diameter height whole_weight shucked_weight viscera_weight
## 1    M         0.455    0.365  0.095       0.5140         0.2245         0.1010
## 2    M         0.350    0.265  0.090       0.2255         0.0995         0.0485
## 3    F         0.530    0.420  0.135       0.6770         0.2565         0.1415
## 4    M         0.440    0.365  0.125       0.5160         0.2155         0.1140
## 5    I         0.330    0.255  0.080       0.2050         0.0895         0.0395
## 6    I         0.425    0.300  0.095       0.3515         0.1410         0.0775
##   shell_weight rings  age
## 1        0.150    15 16.5
## 2        0.070     7  8.5
## 3        0.210     9 10.5
## 4        0.155    10 11.5
## 5        0.055     7  8.5
## 6        0.120     8  9.5
```

```r
hist(abalone_new$age, xlab = "Age",breaks =30, main = "Histogram of Abalone's Age", col = 'orange')
```



As the result we got above, we can see that the age of abalone is not distributed evenly. The shape of the distribution is skewed to the left with a clear mode around age of 11, and most of abalone's age are around 8 to 13 and there is an outlier at around 30.

Question 2

Split the abalone data into a training set and a testing set. Use stratified sampling. You should decide on appropriate percentages for splitting the data.

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

```
set.seed(2000)

abalone_split <- initial_split(abalone_new, prop = 0.80, strata = age)
abalone_train <- training(abalone_split)
abalone_test <- testing(abalone_split)

head(abalone_train)
```

```
##    type longest_shell diameter height whole_weight shucked_weight
## 2     M         0.350    0.265  0.090       0.2255         0.0995
## 5     I         0.330    0.255  0.080       0.2050         0.0895
## 36    M         0.465    0.355  0.105       0.4795         0.2270
## 38    F         0.450    0.355  0.105       0.5225         0.2370
## 43    I         0.240    0.175  0.045       0.0700         0.0315
## 45    I         0.210    0.150  0.050       0.0420         0.0175
##    viscera_weight shell_weight rings age
## 2          0.0485        0.070     7 8.5
## 5          0.0395        0.055     7 8.5
## 36         0.1240        0.125     8 9.5
## 38         0.1165        0.145     8 9.5
## 43         0.0235        0.020     5 6.5
## 45         0.0125        0.015     4 5.5
```

```
head(abalone_test)
```

```
##    type longest_shell diameter height whole_weight shucked_weight
## 6     I         0.425    0.300  0.095       0.3515         0.1410
## 11    F         0.525    0.380  0.140       0.6065         0.1940
## 17    I         0.355    0.280  0.085       0.2905         0.0950
## 19    M         0.365    0.295  0.080       0.2555         0.0970
## 21    M         0.355    0.280  0.095       0.2455         0.0955
## 24    F         0.550    0.415  0.135       0.7635         0.3180
##    viscera_weight shell_weight rings  age
## 6          0.0775        0.120     8  9.5
## 11         0.1475        0.210    14 15.5
## 17         0.0395        0.115     7  8.5
## 19         0.0430        0.100     7  8.5
## 21         0.0620        0.075    11 12.5
## 24         0.2100        0.200     9 10.5
```

Question 3

Using the **training** data, create a recipe predicting the outcome variable, `age`, with all other predictor variables. Note that you should not include `rings` to predict `age`. Explain why you shouldn't use `rings` to predict `age`.

Steps for your recipe:

1. dummy code any categorical predictors
2. create interactions between
   - `type` and `shucked_weight`,
   - `longest_shell` and `diameter`,
   - `shucked_weight` and `shell_weight`
3. center all predictors, and
4. scale all predictors.

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

```
abalone_recipe <- abalone_train %>%
  recipe(age ~ type +longest_shell + diameter + height +
           whole_weight + shucked_weight + viscera_weight +
           shell_weight) %>%
  step_dummy(all_nominal_predictors())  %>%
  step_interact(terms = ~ starts_with("type"):shucked_weight+
                  longest_shell:diameter+
                  shucked_weight:shell_weight) %>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors())
```

Question 4

Create and store a linear regression object using the `"lm"` engine.

```
lm_model <- linear_reg() %>%
  set_engine("lm")
```

Question 5

Now:

1. set up an empty workflow,
2. add the model you created in Question 4, and
3. add the recipe that you created in Question 3.

```
lm_wflow <- workflow() %>%
  add_model(lm_model) %>%
  add_recipe(abalone_recipe)
```

Question 6

Use your `fit()` object to predict the age of a hypothetical female abalone with longest_shell = 0.50, diameter = 0.10, height = 0.30, whole_weight = 4, shucked_weight = 1, viscera_weight = 2, shell_weight = 1.

```
#fitting the model
lm_fit <- fit(lm_wflow, abalone_train)
```

```
#predicting
abalone_hypo <- data.frame(type="F",longest_shell = 0.50,
                           diameter = 0.10,
                           height = 0.30,
                           whole_weight = 4,
                           shucked_weight = 1,
                           viscera_weight = 2,
                           shell_weight = 1)
predict(lm_fit,new_data = abalone_hypo)
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  24.8
```

Question 7

Now you want to assess your model's performance. To do this, use the `yardstick` package:

1. Create a metric set that includes R^2, RMSE (root mean squared error), and MAE (mean absolute error).

```
abalone_metrics <- metric_set(rsq, rmse, mae)
```

2. Use `predict()` and `bind_cols()` to create a tibble of your model's predicted values from the **training data** along with the actual observed ages (these are needed to assess your model's performance).

```
abalone_train_res <- predict(lm_fit, new_data = abalone_train %>% select(-age))


abalone_train_res <- bind_cols(abalone_train_res, abalone_train %>% select(age))
abalone_train_res %>%
  head()
```

```
## # A tibble: 6 x 2
##   .pred   age
##   <dbl> <dbl>
## 1  9.48   8.5
## 2  8.10   8.5
## 3 10.1    9.5
## 4 11.0    9.5
## 5  6.30   6.5
## 6  5.94   5.5
```

3. Finally, apply your metric set to the tibble, report the results, and interpret the R^2 value.

```
abalone_metrics(abalone_train_res, truth = age, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.559
## 2 rmse    standard       2.13
## 3 mae     standard       1.53
```

The R^2 represents how well the regression model fits the observed data. As the we getting rsq = 0.5567203, the model is not fitting the observed data well.
The root mean squared error is the standard deviation of the residuals (prediction errors), As we are predicting the age of abalone, a 2.1412871 of rmse is a bit to big as a prediction error.
mae: The mean absolute error indicates that the magnitude of difference between the prediction of an observation and the true value of that observation is 1.5456314.