131-hw2

Isha Gokhale

2022-10-13

(1)

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 1.0.0 --
## v broom        1.0.1     v rsample      1.1.0
## v dials        1.0.0     v tune         1.0.1
## v infer        1.0.3     v workflows    1.1.0
## v modeldata    1.0.1     v workflowsets 1.0.0
## v parsnip      1.0.2     v yardstick    1.1.0
## v recipes      1.0.1
## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Dig deeper into tidy modeling with R at https://www.tmwr.org
```

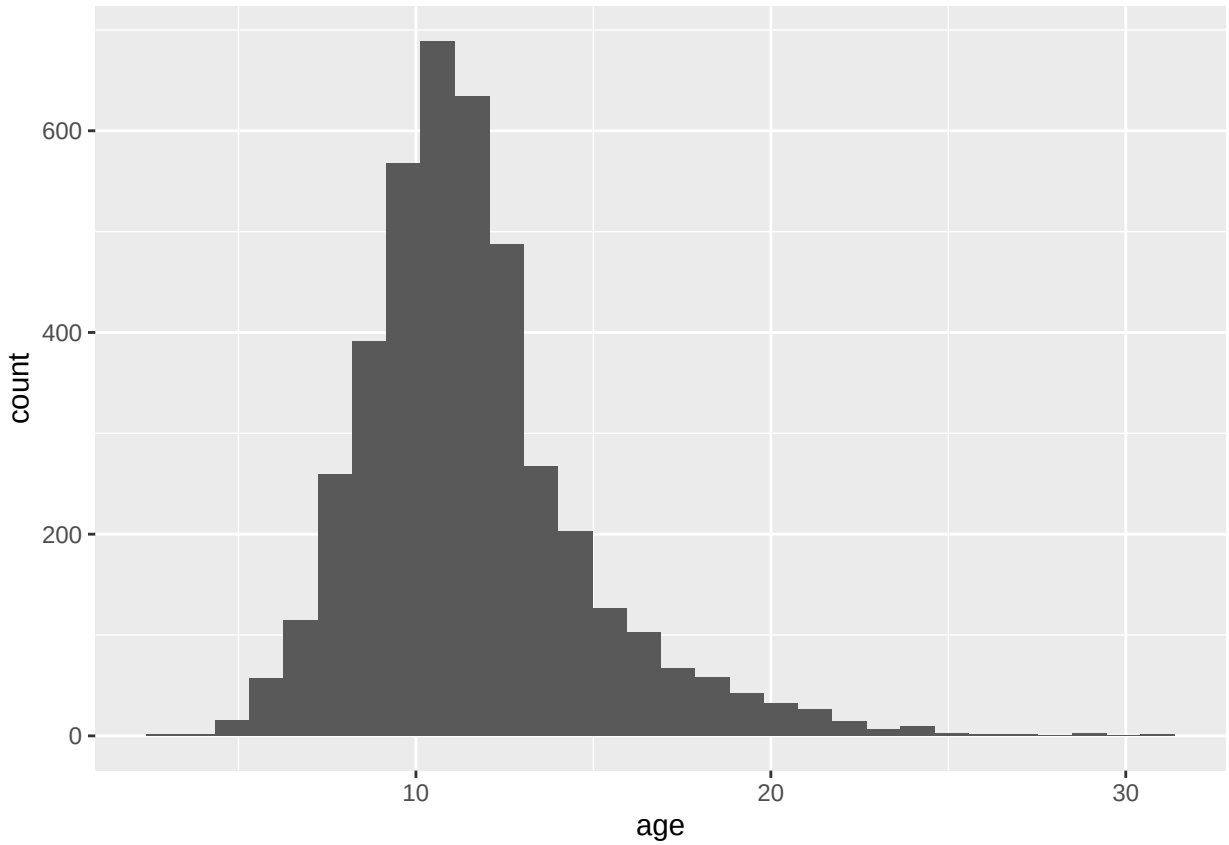
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggthemes)
tidymodels_prefer()
```

```
df = read.csv('~/Downloads/hw2/data/abalone.csv')
df$age <- df$rings+1.5
ggplot(df, aes(x=age)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can see that the distribution of the age variable is right skewed, meaning majority of the observations lie towards the left. The mean age seems to be around 10.

(2)

```
set.seed(3435)
df_split <- initial_split(df, prop = 0.80,strata = age)
df_train <- training(df_split)
df_test <- testing(df_split)
```

(3)

```
df_recipe <- recipe(age ~ type+longest_shell + diameter+ height + whole_weight + shucked_weight+ viscera
step_dummy(all_nominal_predictors()) %>%
step_interact(terms = ~longest_shell:diameter) %>%
step_interact(terms = ~starts_with('type'):shucked_weight) %>%
step_interact(terms = ~shucked_weight:shell_weight) %>%
step_zv(all_predictors()) %>%
step_normalize(all_numeric(), -all_outcomes()) %>%
prep(verbose = TRUE, log_changes = TRUE)
```

```
## oper 1 step dummy [training]
## step_dummy (dummy_Es02q):
##  new (2): type_I, type_M
##  removed (1): type
##
```

```
## oper 2 step interact [training]
## step_interact (interact_mwFDe):
##  new (1): longest_shell_x_diameter
##
## oper 3 step interact [training]
## step_interact (interact_Wl0Dt):
##  new (2): type_I_x_shucked_weight, type_M_x_shucked_weight
##
## oper 4 step interact [training]
## step_interact (interact_cD3Mp):
##  new (1): shucked_weight_x_shell_weight
##
## oper 5 step zv [training]
## step_zv (zv_Qev3K): same number of columns
##
## oper 6 step normalize [training]
## step_normalize (normalize_PAdDJ): same number of columns
##
## The retained training set is ~ 0.36 Mb  in memory.
```

We can leave out the rings variable because the age variable is directly derived from the ring variable as age = rings + 1.5, which is linearly dependent. Therefore, all the other predictors would not be useful as rings would give us a perfect outcome each time.

(4)

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

(5)

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

```
lm_fit <- fit(lm_wflow, df_train)

lm_fit %>%
  # This returns the parsnip object:
  extract_fit_parsnip() %>%
  # Now tidy the linear model object:
  tidy()
```

```
## # A tibble: 14 x 5
##    term                        estimate std.error statistic  p.value
##    <chr>                          <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)                  11.4       0.0375  305.     0
##  2 longest_shell                 0.591     0.286     2.07   3.86e- 2
##  3 diameter                      2.06      0.313     6.61   4.59e-11
##  4 height                        0.236     0.0696    3.39   7.10e- 4
##  5 whole_weight                  4.29      0.387    11.1    4.66e-28
##  6 shucked_weight               -4.06      0.250   -16.2    5.35e-57
##  7 viscera_weight               -0.792     0.158    -5.00   6.12e- 7
##  8 shell_weight                  1.74      0.212     8.20   3.32e-16
##  9 type_I                       -0.942     0.117    -8.07   9.36e-16
```

```
## 10 type_M                         -0.239      0.104    -2.29   2.21e- 2
## 11 longest_shell_x_diameter       -2.75       0.396    -6.95   4.32e-12
## 12 type_I_x_shucked_weight         0.525      0.0876    5.99   2.26e- 9
## 13 type_M_x_shucked_weight         0.293      0.109     2.68   7.41e- 3
## 14 shucked_weight_x_shell_weight -0.00330    0.205    -0.0161 9.87e- 1
```

(6)

```
new_data <- data.frame(longest_shell = 0.50, diameter = 0.10, height = 0.30, whole_weight = 4, shucked_
```

```
lm_fit
```

```
## == Workflow [trained] ==========================================================
## Preprocessor: Recipe
## Model: linear_reg()
##
## -- Preprocessor ----------------------------------------------------------------
## 6 Recipe Steps
##
## * step_dummy()
## * step_interact()
## * step_interact()
## * step_interact()
## * step_zv()
## * step_normalize()
##
## -- Model -----------------------------------------------------------------------
##
## Call:
## stats::lm(formula = ..y ~ ., data = data)
##
## Coefficients:
##                   (Intercept)                  longest_shell
##                     11.423353                       0.591227
##                      diameter                         height
##                      2.064936                       0.235918
##                  whole_weight                 shucked_weight
##                      4.288839                      -4.061214
##                viscera_weight                   shell_weight
##                     -0.791796                       1.735662
##                        type_I                         type_M
##                     -0.942109                      -0.238578
##      longest_shell_x_diameter       type_I_x_shucked_weight
##                     -2.753237                       0.524826
##       type_M_x_shucked_weight shucked_weight_x_shell_weight
##                      0.293330                      -0.003297
```

```
pred <- predict(lm_fit, new_data = new_data)
pred
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  23.7
```

The hypothetical age of this abalone would be around 23.7 years.

(7)

```
df_train_res <- predict(lm_fit, new_data = df_train %>% select(-age))
df_train_res %>%
  head()
```

```
## # A tibble: 6 x 1
##   .pred
##   <dbl>
## 1  8.03
## 2  9.68
## 3 10.4
## 4 10.1
## 5 10.9
## 6  6.26
```

```
df_train_res <- bind_cols(df_train_res, df_train %>% select(age))
df_train_res %>%
  head()
```

```
## # A tibble: 6 x 2
##   .pred   age
##   <dbl> <dbl>
## 1  8.03   8.5
## 2  9.68   8.5
## 3 10.4    8.5
## 4 10.1    9.5
## 5 10.9    9.5
## 6  6.26   6.5
```

```
rmse(df_train_res, truth = age, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        2.16
```

```
df_metrics <- metric_set(rmse, rsq, mae)
df_metrics(df_train_res, truth = age,
               estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.16
## 2 rsq     standard       0.551
## 3 mae     standard       1.55
```

We can see that the r squared is .5513. This means that our predictors have a moderate effect on the age. The predictors explain around 55% of the variability seen in age can be explained by our predictors in our model. From the predictions we can see that they were pretty close to the actual age, however, two of the predictions were over a year off when the age was actually 8.5. Overall, the predictions were pretty accurate. The RMSE is 2.16.