Homework 2

PSTAT 131/231

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.

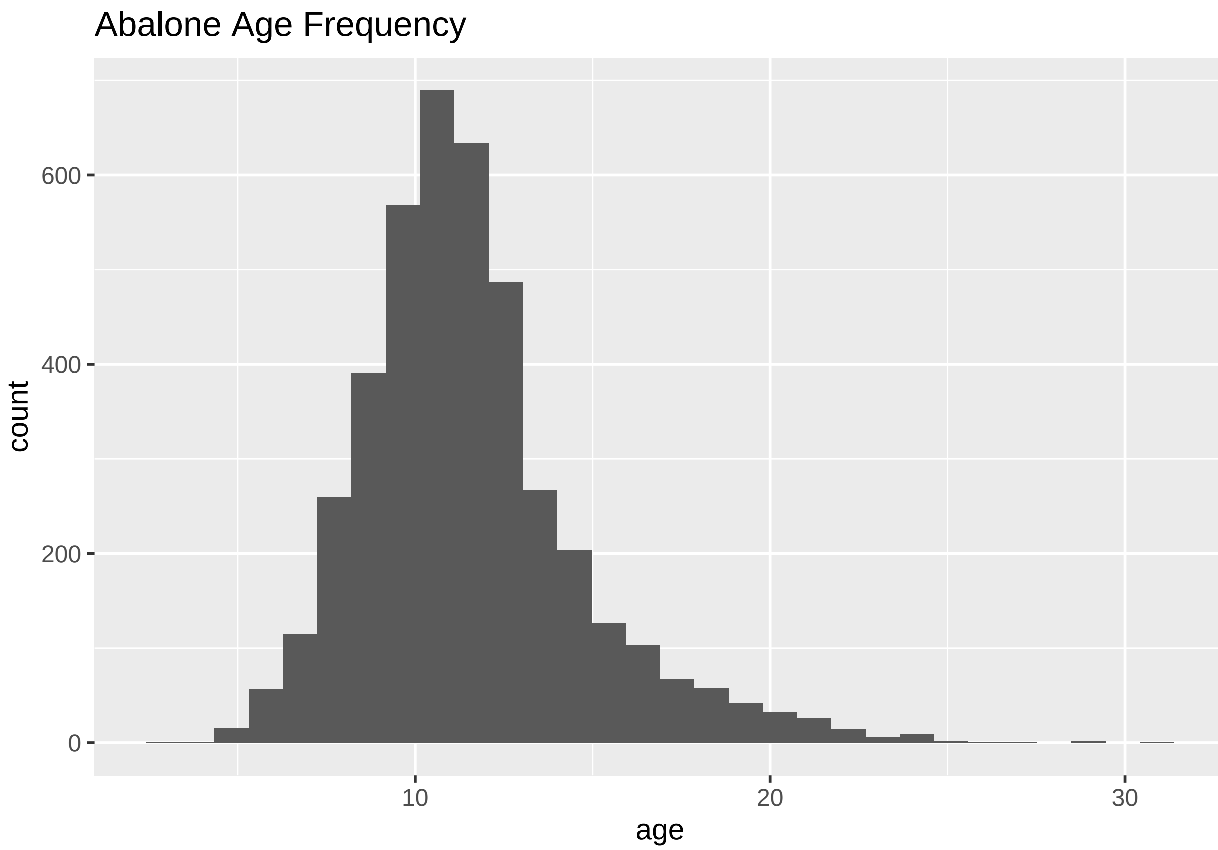
Make sure you load the `tidyverse` and `tidymodels`!

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`.

```
abalone <- read_csv(file = "data/abalone.csv")
abalone <- abalone %>%
  mutate(age = rings+1.5)
abalone %>%
  ggplot(aes(x = age)) +
  geom_histogram()+
  ggtitle("Abalone Age Frequency")
```



The distribution is roughly bell shaped, but it does have a right skew. The mode age seems to be around 11 years of age and most abalone are under 15 years of age.

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(42069) # for reproducibility

# get stratified train/testing sets
abalone_split <- initial_split(abalone, prop = 0.80,
                               strata = age)
abalone_train <- training(abalone_split)
abaone_test <- testing(abalone_split)
```

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.

We shouldn't use `rings` because our response variable `age`, is a linear transformation of `rings`.

```
abalone_recipe <- recipe(age ~ ., data = abalone_train%>%select(-rings)) %>%
  step_dummy(all_nominal_predictors())%>%
  step_interact(terms = ~ starts_with("type"):shucked_weight + longest_shell:diameter + shucked_weight:s
  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.

```
lm_fit <- fit(lm_wflow, abalone_train%>%select(-rings))
new_abalone <- data.frame(type="F", longest_shell = 0.50, diameter = 0.10, height = 0.30, whole_weight =
pred <- predict(lm_fit, new_data = new_abalone)
pred
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  24.0
```

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).
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).

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

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

abalone_metrics <- metric_set(rmse, rsq, mae)
abalone_metrics(abalone_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       0.553
## 3 mae     standard       1.55
```

We have that $R^2 = 0.5533$ which means that about 55% of the variability in the response variable (`age`) is explained by our model.

Required for 231 Students

In lecture, we presented the general bias-variance tradeoff, which takes the form:

$$E[(y_0 - \hat{f}(x_0))^2] = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

where the underlying model $Y = f(X) + \epsilon$ satisfies the following:

- ϵ is a zero-mean random noise term and X is non-random (all randomness in Y comes from ϵ);
- (x_0, y_0) represents a test observation, independent of the training set, drawn from the same model;
- $\hat{f}(.)$ is the estimate of f obtained from the training set.

Question 8 Which term(s) in the bias-variance tradeoff above represent the reproducible error? Which term(s) represent the irreducible error?

Answer: The reducible error is represented by $\mathrm{Var}(\hat{f}(x_0)) + \mathrm{Bias}^2(\hat{f}(x_0))$ and the irreducible error is represented by $\mathrm{Var}(\epsilon)$.

Question 9 Using the bias-variance tradeoff above, demonstrate that the expected test error is always at least as large as the irreducible error.

Answer: Note that variance is inherently a nonnegative quantity, and squared bias is also nonnegative. Hence, we see that the expected test MSE can never lie below $\mathrm{Var}(\epsilon)$, the irreducible error.

Question 10 Prove the bias-variance tradeoff.

Hints:

- use the definition of $Bias(\hat{f}(x_0)) = E[\hat{f}(x_0)] - f(x_0)$;
- reorganize terms in the expected test error by adding and subtracting $E[\hat{f}(x_0)]$

Answer: We have that

$$
\begin{aligned}
E[(y_0 - \hat{f}(x_0))^2] &= E[((f(x_0) + \epsilon) - \hat{f}(x_0))^2] \\
&= E[((f(x_0) - \hat{f}(x_0) + \epsilon)^2] \\
&= E[(f(x_0) - \hat{f}(x_0))^2] + \underbrace{E[(f(x_0) - \hat{f}(x_0))\epsilon]}_{=E[(f(x_0)-\hat{f}(x_0))]E[\epsilon]=0} + \underbrace{E[\epsilon^2]}_{=\mathrm{Var}(\epsilon)} \\
&= E[f^2(x_0) - 2f(x_0)\hat{f}(x_0) + \hat{f}^2(x_0)] + \mathrm{Var}(\epsilon) \\
&= f^2(x_0) - 2f(x_0)E[\hat{f}(x_0)] \qquad\qquad + E[\hat{f}^2(x_0)] \qquad\qquad + \mathrm{Var}(\epsilon) \\
&= \underbrace{f^2(x_0) - 2f(x_0)E[\hat{f}(x_0)] + E[\hat{f}(x_0)]^2}_{(E[\hat{f}(x_0)]-f(x_0))^2} + E[\hat{f}^2(x_0)] - E[\hat{f}(x_0)]^2 + \mathrm{Var}(\epsilon) \\
&= \mathrm{Bias}^2(\hat{f}(x_0)) + \mathrm{Var}(\hat{f}(x_0)) + \mathrm{Var}(\epsilon)
\end{aligned}
$$