Bee Colonies Lost Prediction using Linear Regression

Masardi Rachman Rosyid (29918758)

2022-06-21

Clearly state your chosen outcome and why your group chose it.

```
library(tidyverse)
library(ggplot2)
library(here)
library(class)
library(leaps)
library(jtools)
set.seed(1374372)
```

Individual component

```
colony <- read_csv(here("data/colony.csv"))
stressor <- read_csv(here("data/stressor.csv"))
combined_data <- left_join(
  x = colony,
  y = stressor,
  by = c("year", "months", "state")
) %>%
  select(
    -c(...1.x, ...1.y)
  )
```

Model Chosen and Reasoning

The data chosen specifically for this project is the linear regression, the nature of the data is understood to be not completely linear, but it is good enough, there will be an expected wide prediction interval due to this reason. The error is observed to be normally distributed and also homoscedastic which makes it more appropriate for a linear regression as opposed to regression tree due to its nature to make partitions in which could overly generalised the dependent variable.

Data Training and Accuracy Measures plus a dummy data

```
full_data <- combined_data %>%
  filter(state == "United States") %>%
```

```
  mutate(
    state = as.factor(state),
    stress_pct = replace_na(stress_pct, 0),
    stress_pct = stress_pct / 100,
    months = factor(
      x = months,
      levels = c("January-March", "April-June", "July-September", "October-December"),
      labels = c("Q1", "Q2", "Q3", "Q4")
    ),
    stressor = str_to_title(stressor),
    stressor = str_replace_all(
      string = stressor,
      pattern = c(
        "Other Pests_parasites" = "Other Pests Parasites",
        "Disesases" = "Diseases"
      )
    ),
    stressor = as.factor(stressor),
    colony_reno_pct = colony_reno_pct / 100
  ) %>%
  drop_na(colony_n) %>%
  # Replace value in all columns with 0
  replace(list = is.na(.), values = 0) %>%
  rename(
    stress_proportion = stress_pct,
    quarter = months,
    colony_reno_proportion = colony_reno_pct
  ) %>%
  select(
    -c(
      state,
      colony_max,
      colony_lost_pct,
      colony_reno_proportion
    )
  )
```

```
training_size <- 0.90
test_size <- 1 - training_size
level <- 0.95
lm_formula <- "colony_lost ~ ."

data_subsets <- full_data %>%
  mutate(
    sample_type = sample(
      x = c("train", "test"),
      size = n(),
      replace = TRUE,
      prob = c(training_size, test_size)
    )
  )

training_set <- data_subsets %>%
```

```r
  filter(sample_type == "train") %>%
  select(-sample_type)

test_set <- data_subsets %>%
  filter(sample_type == "test") %>%
  select(-sample_type)

lm_model <- lm(
  formula = lm_formula,
  data = training_set
)

train_prediction <- predict(
  object = lm_model,
  newdata = training_set,
  level = level,
  interval = "prediction"
)

train_prediction <- data.frame(
  training_set,
  train_prediction
) %>%
  mutate(
    within_coverage = colony_lost > lwr & colony_lost < upr
  )

test_prediction <- predict(
  object = lm_model,
  newdata = test_set,
  level = level,
  interval = "prediction"
)

test_prediction <- data.frame(
  test_set,
  test_prediction
) %>%
  mutate(
    within_coverage = colony_lost > lwr & colony_lost < upr
  )

train_errors <- train_prediction %>%
  summarise(
    me = mean(colony_lost - fit),
    mae = mean(abs(colony_lost - fit)),
    mse = mean((colony_lost - fit)^2),
    rmse = sqrt(mse),
    coverage = mean(within_coverage)
  )

test_errors <- test_prediction %>%
  summarise(
```

```
    me = mean(colony_lost - fit),
    mae = mean(abs(colony_lost - fit)),
    mse = mean((colony_lost - fit)^2),
    rmse = sqrt(mse),
    coverage = mean(within_coverage)
  )

compare_errors <- rbind(
  train_errors,
  test_errors
)

new_data <- data.frame(
  year = 2020,
  quarter = "Q2",
  colony_n = 3000000,
  colony_added = 50000,
  colony_reno = 250000,
  stressor = "Varroa Mites",
  stress_proportion = 0.50
)

predict(
  object = lm_model,
  newdata = new_data,
  interval = "prediction",
  level = level
)
```

```
##        fit      lwr      upr
## 1 286708.1 208979.6 364436.7
```