Penguin

Josh Kong

12/2/2020

Objective: Create a classification model that predicts the sex of a penguin based on certain features.

Loading the data and neccesary packages in

```
penguins <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/
```

```
## Parsed with column specification:
## cols(
##   species = col_character(),
##   island = col_character(),
##   bill_length_mm = col_double(),
##   bill_depth_mm = col_double(),
##   flipper_length_mm = col_double(),
##   body_mass_g = col_double(),
##   sex = col_character(),
##   year = col_double()
## )
```

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------ tidyverse
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts --------------------------------------------------------------------- tidyverse_confl
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
## -- Attaching packages ------------------------------------------------------------------ tidymodels
```

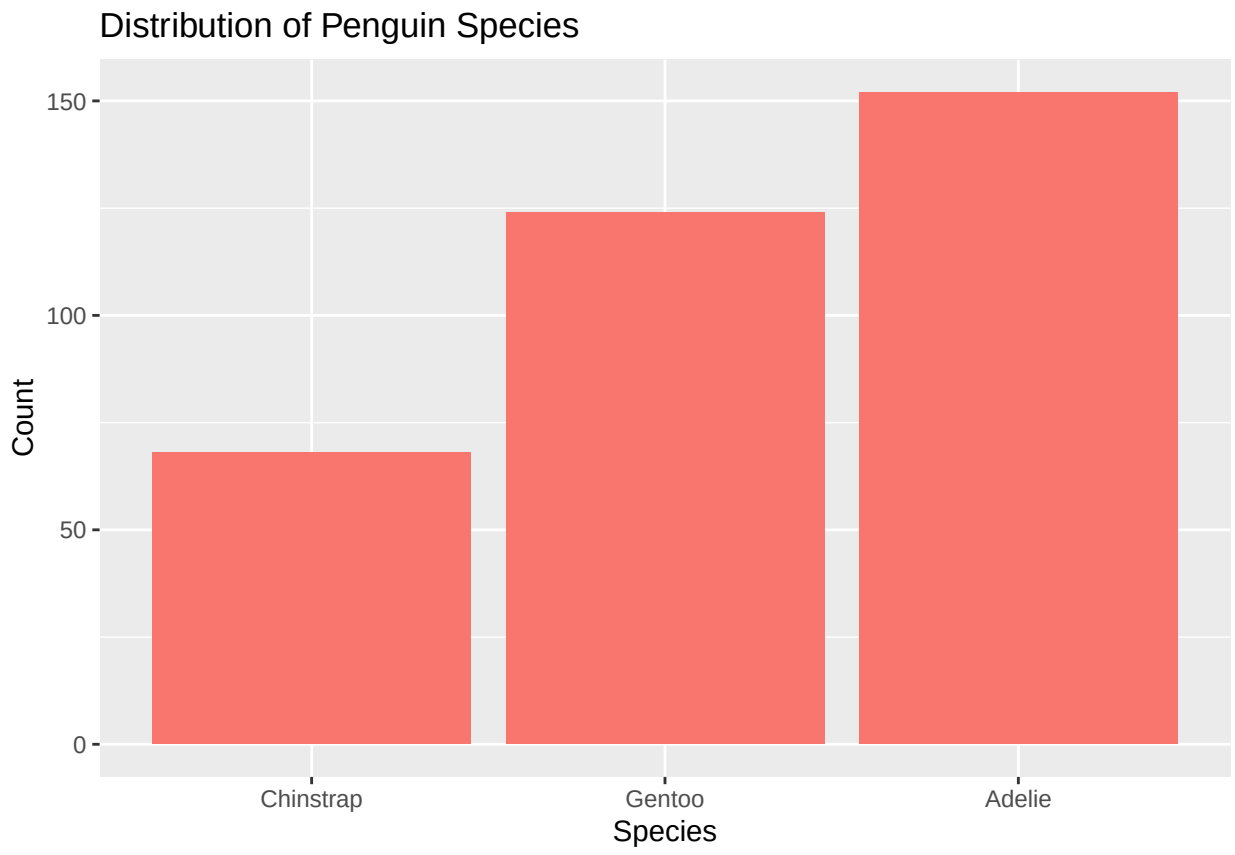
```
## v broom     0.7.0      v recipes   0.1.13
## v dials     0.0.8      v rsample   0.0.7
## v infer     0.5.3      v tune      0.1.1
## v modeldata 0.0.2      v workflows 0.1.3
## v parsnip   0.1.3      v yardstick 0.0.7

## -- Conflicts ------------------------------------------------------------------------ tidymodels_confl
## 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()
```

Data exploration

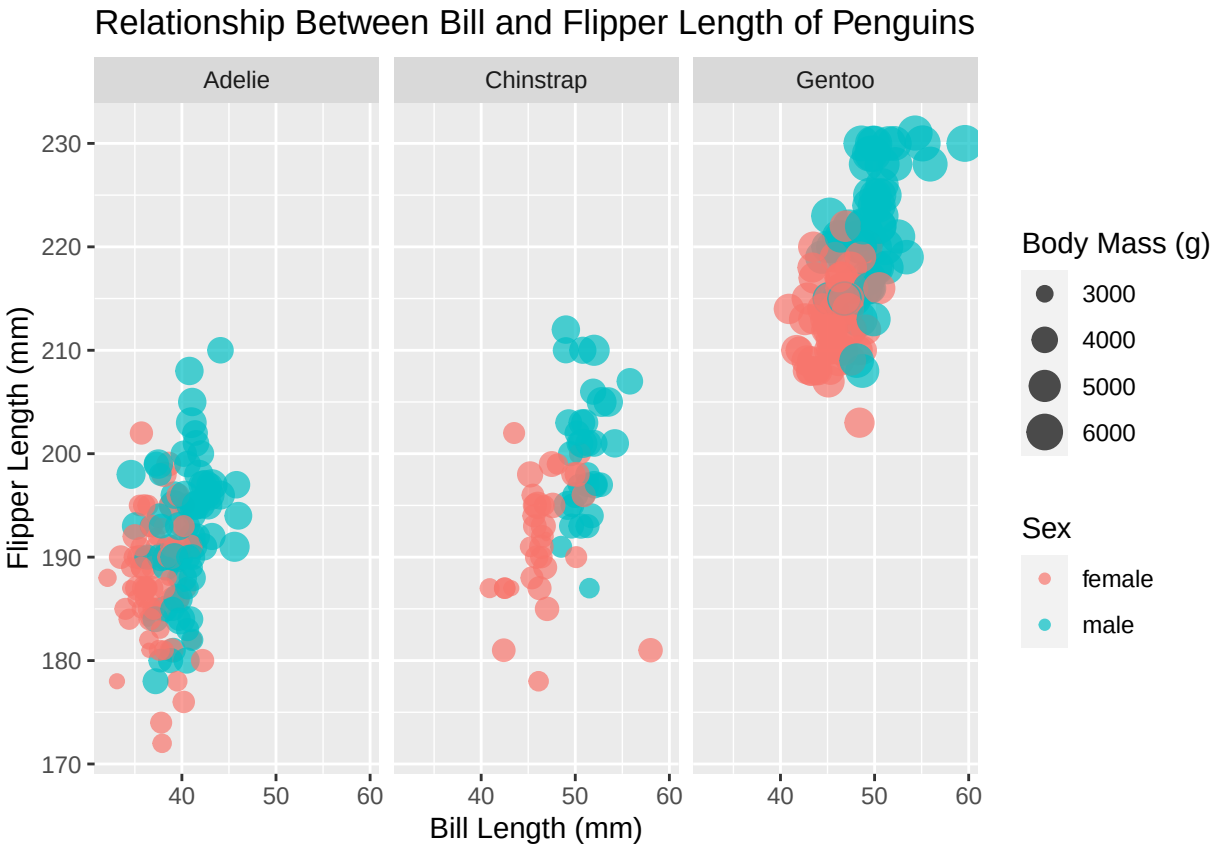
Looking at the distribution of different penguin species

```
penguins %>%
  count(species) %>%
  mutate(species = fct_reorder(species, n)) %>%
  ggplot(aes(species, n, fill = "blue")) +
  geom_col() +
  theme(legend.position = "none") +
  labs(x = "Species", y = "Count", title = "Distribution of Penguin Species")
```



Taking a look at the relationship between bill length and flipper length of penguins of different sex and different species

```
penguins %>%
  filter(!is.na(sex)) %>%
  ggplot(aes(bill_length_mm, flipper_length_mm, color = sex, size = body_mass_g)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~species) +
  labs(x = "Bill Length (mm)", y = "Flipper Length (mm)", title = "Relationship Between Bill and Flippe
```



Selecting the data we want for our model

```
penguins_df <- penguins %>%
  filter(!is.na(sex)) %>%
  select(-year, -island)

#changing our character columns into factors
penguins_df <- penguins_df %>%
  mutate_if(is.character, factor)
```

Model Building

Splitting the data

```
set.seed(123)
penguin_split <- initial_split(penguins_df, strata = sex)
penguin_train <- training(penguin_split)
penguin_test <- testing(penguin_split)
```

Creating a recipe

```
penguins_df
```

```
## # A tibble: 333 x 6
##    species bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex
##    <fct>            <dbl>         <dbl>             <dbl>       <dbl> <fct>
##  1 Adelie            39.1          18.7               181        3750 male
##  2 Adelie            39.5          17.4               186        3800 female
##  3 Adelie            40.3          18                 195        3250 female
##  4 Adelie            36.7          19.3               193        3450 female
##  5 Adelie            39.3          20.6               190        3650 male
##  6 Adelie            38.9          17.8               181        3625 female
##  7 Adelie            39.2          19.6               195        4675 male
##  8 Adelie            41.1          17.6               182        3200 female
##  9 Adelie            38.6          21.2               191        3800 male
## 10 Adelie            34.6          21.1               198        4400 male
## # ... with 323 more rows
```

```
penguin_rec <- recipe(sex ~ ., data = penguin_train) %>%
  step_dummy(species,one_hot = TRUE)   #making species into dummy variables.
```

KNN Model

Building the Model

```
knn_spec <- nearest_neighbor(neighbors = 9) %>%
  set_mode("classification") %>%
  set_engine("kknn")

knn_wf <- workflow() %>%
  add_recipe(penguin_rec) %>%
  add_model(knn_spec)

knn_fit <- knn_wf %>%
  fit(penguin_train)
```

Evaluating the Model

```
pred_knn <- predict(knn_fit, penguin_test)
knn_conf <- table(pred_knn$.pred_class, penguin_test$sex); knn_conf
```

```
##
##          female male
##   female     39    5
##   male        2   37
```

```
knn_acc <- (knn_conf[1,1] + knn_conf[2,2]) / sum(knn_conf); knn_acc
```

```
## [1] 0.9156627
```

```
paste0("Got an accuracy of ",round(knn_acc,2),"% using KNN.")
```

```
## [1] "Got an accuracy of 0.92% using KNN."
```

Random Forest Model

Building the model

```
set.seed(123)
rf_spec <- rand_forest(trees= 1000, mtry = 4) %>%
  set_mode("classification") %>%
  set_engine("ranger")

rf_wf <- workflow() %>%
  add_recipe(penguin_rec) %>%
  add_model(rf_spec)

rf_fit <- rf_wf %>%
  fit(penguin_train)
```

Evaluating the Model

```
pred_rf <- predict(rf_fit, penguin_test)
rf_conf <- table(pred_rf$.pred_class, penguin_test$sex); rf_conf
```

```
##
##          female male
##   female     39    2
##   male        2   40
```

```
rf_acc <- (rf_conf[1,1] + rf_conf[2,2]) / sum(rf_conf); rf_acc
```

```
## [1] 0.9518072
```

```
paste0("Got an accuracy of ",round(rf_acc,2),"% using a random forest model.")
```

```
## [1] "Got an accuracy of 0.95% using a random forest model."
```

Boosted Tree Model

Building the Model

```
set.seed(234)
xgb_spec <- boost_tree(trees= 1000, mtry = 7) %>%
  set_mode("classification") %>%
  set_engine("xgboost")

xgb_wf <- workflow() %>%
  add_recipe(penguin_rec) %>%
  add_model(xgb_spec)

xgb_fit <- xgb_wf %>%
  fit(penguin_train)
```

Evaluating the Model

```
pred_xgb <- predict(xgb_fit, penguin_test)
xgb_conf <- table(pred_xgb$.pred_class, penguin_test$sex); xgb_conf
```

```
##
##          female male
##   female     39    3
##   male        2   39
```

```
xgb_acc <- (xgb_conf[1,1] + xgb_conf[2,2]) / sum(xgb_conf); xgb_acc
```

```
## [1] 0.939759
```

```
paste0("Got an accuracy of ",round(xgb_acc,2),"% using a boosted tree model.")
```

```
## [1] "Got an accuracy of 0.94% using a boosted tree model."
```

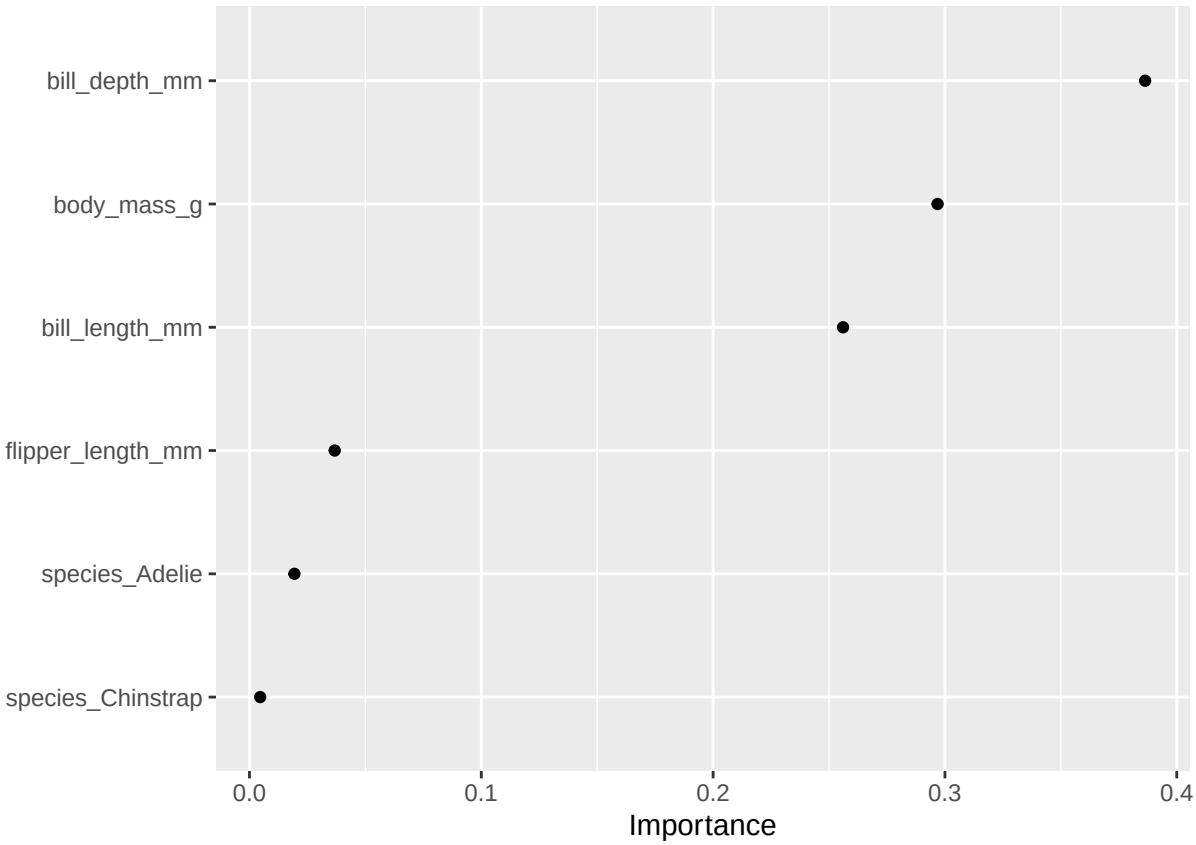
```
#which variables contributed most to the classfication of the sex of penguins?
library(vip)
```

```
##
## Attaching package: 'vip'

## The following object is masked from 'package:utils':
##
##     vi
```

```
xgb_fit %>%
  pull_workflow_fit() %>%
  vip(geom="point")
```

```
## Warning: ‘as.tibble()‘ is deprecated as of tibble 2.0.0.
## Please use ‘as_tibble()‘ instead.
## The signature and semantics have changed, see ‘?as_tibble‘.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_warnings()‘ to see where this warning was generated.
```



The two most important variables in this classfication model were the bill depth of the penguin and the body mass of the penguin.

Conclusion

I used 3 classification machine learning models (K-Nearest Neighbor, Random Forest Classifier, Boosted Tree Classifier) to predict the sex of a penguin based on certain features. Using a random forest classifier, I was able to predict the sex of a penguin with an accuracy rate of ~95%.

My final conclusion is, the feature that strongly defines a penguin's sex is the bill depth and the body mass.