

Slide 14 - R Output

Elvan Ceyhan

11/21/2024

```
#Set Working Directory to Source File Location  
library("rstudioapi") # Load rstudioapi package  
#setwd(dirname(getActiveDocumentContext()$path)) # Set working directory to source file location  
#getwd()
```

Some required packages:

```
# Load packages  
library(rstan)  
library(mvtnorm) # For multivariate normal sampling  
library(ggplot2)
```

Naive Bayes Classification using Penguin Data

```
# Load packages  
library(bayesrules)  
library(tidyverse)  
library(e1071)  
library(janitor)  
  
# Load penguin dataset  
data(penguins_bayes) # Load dataset from bayesrules package  
penguins <- penguins_bayes  
  
# Summarize species counts in the sample  
penguins %>%  
  tabyl(species) %>%  
  adorn_totals("row") # Tidy summary with row total
```

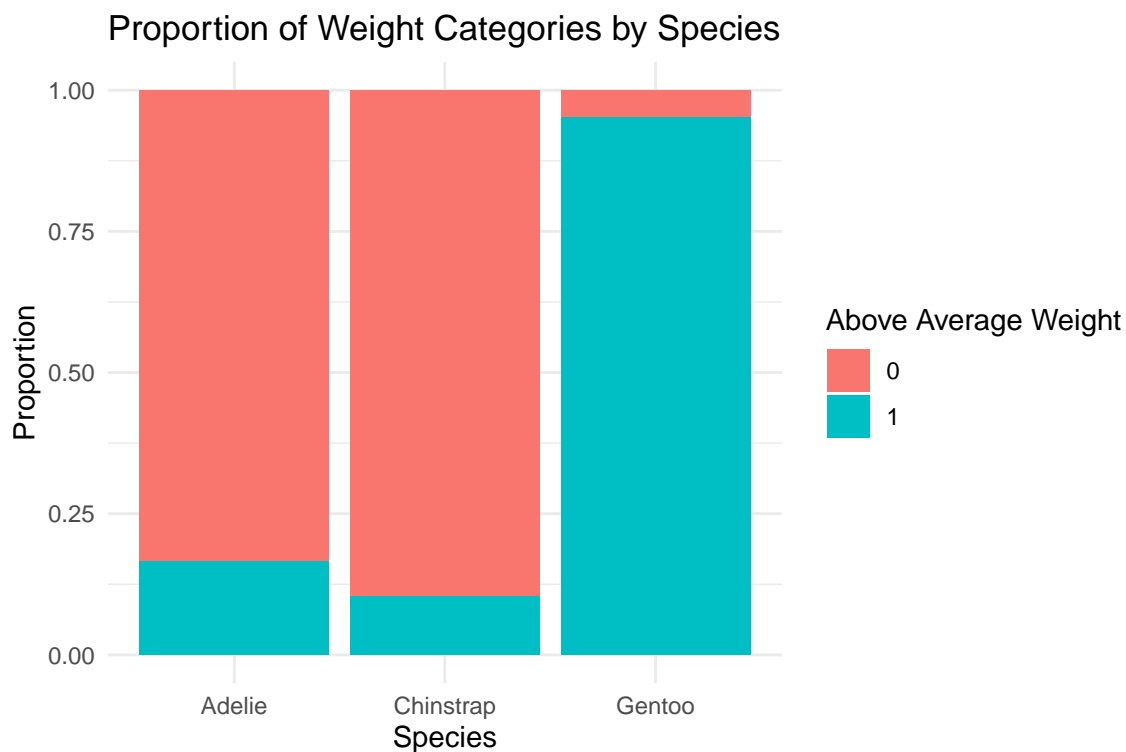
```
##   species    n  percent  
##   Adelie  152 0.4418605  
## Chinstrap   68 0.1976744  
##   Gentoo  124 0.3604651  
##      Total  344 1.0000000
```

```
# Base R equivalent for species counts  
table(penguins$species) # Simple table for species count
```

```
##
##      Adelie Chinstrap   Gentoo
##      152      68      124
```

Naive Bayes Classification with One Categorical Predictor

```
# Stacked bar plot showing species distribution by weight category
penguins %>%
  drop_na(above_average_weight) %>% # Remove rows with missing weight data
  ggplot(aes(fill = above_average_weight, x = species)) +
  geom_bar(position = "fill") +
  labs(
    title = "Proportion of Weight Categories by Species",
    x = "Species",
    y = "Proportion",
    fill = "Above Average Weight"
  ) +
  theme_minimal()
```



```
# Cross-tabulation of species by weight categories with totals
penguins %>%
  select(species, above_average_weight) %>% # Select relevant columns
  drop_na() %>% # Remove rows with missing data
  tabyl(species, above_average_weight) %>%
  adorn_totals(c("row", "col")) # Add row and column totals
```

```
##      species    0    1 Total
```

```
##      Adelie 126  25  151
## Chinstrap  61   7   68
##      Gentoo   6 117  123
##      Total 193 149  342
```

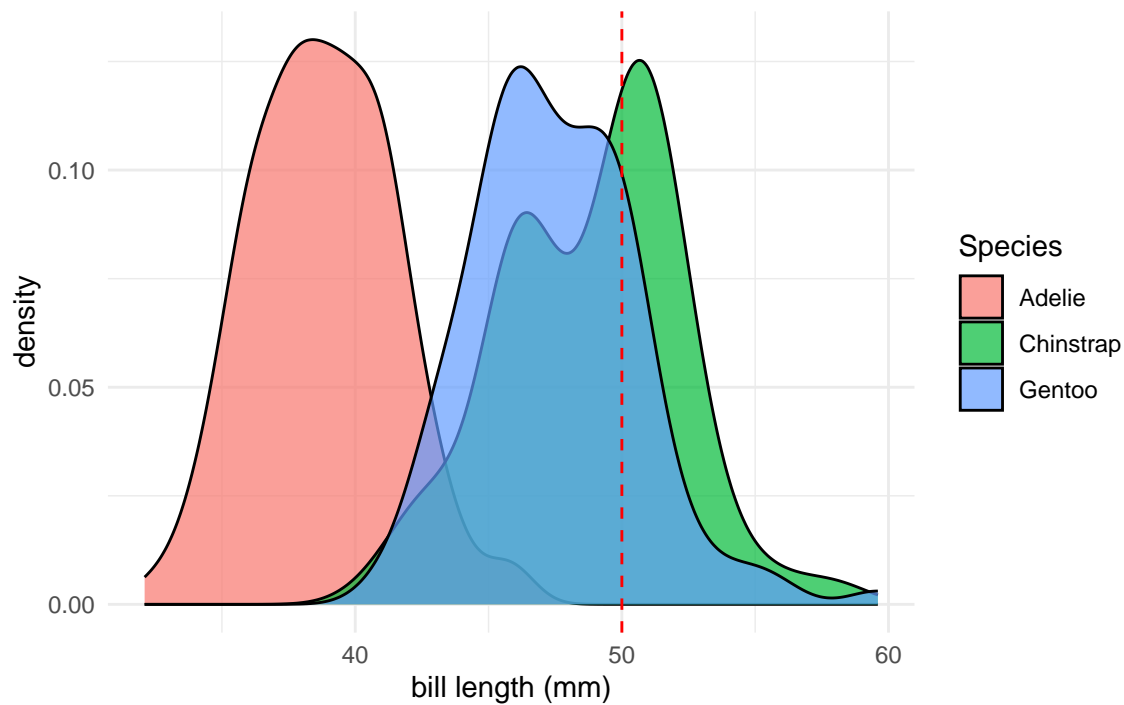
```
# Base R equivalent for the same cross-tabulation
addmargins(table(penguins$species, penguins$above_average_weight)) # Adds margins (totals)
```

```
##
##              0   1 Sum
## Adelie      126  25 151
## Chinstrap    61   7  68
## Gentoo        6 117 123
## Sum          193 149 342
```

The conditional distributions of bill length

```
# Density plot of bill length for each species with a reference line at 50mm
penguins %>%
  ggplot(aes(x = bill_length_mm, fill = species)) +
  geom_density(alpha = 0.7) + # Overlapping density curves with transparency
  geom_vline(xintercept = 50, linetype = "dashed", color = "red") + # Reference line at 50mm
  labs(
    title = "Conditional Distribution of Bill Length by Species",
    x = "bill length (mm)",
    y = "density",
    fill = "Species"
  ) +
  theme_minimal()
```

Conditional Distribution of Bill Length by Species



```
# Calculate sample means and standard deviations of bill length for each species
penguins %>%
  group_by(species) %>%
  summarize(
    mean_bill_length = mean(bill_length_mm, na.rm = TRUE), # Exclude missing values
    sd_bill_length = sd(bill_length_mm, na.rm = TRUE) # Exclude missing values
  ) %>%
  arrange(desc(mean_bill_length)) # Order by mean bill length
```

```
## # A tibble: 3 x 3
##   species mean_bill_length sd_bill_length
##   <fct>         <dbl>         <dbl>
## 1 Chinstrap      48.8           3.34
## 2 Gentoo         47.5           3.08
## 3 Adelie         38.8           2.66
```

```
# Likelihood calculations for flipper length (X3 = 195) for each species
# Using normal density function (dnorm)
```

```
# Species A (Adelie): mean = 190, sd = 6.54
L_A <- dnorm(195, mean = 190, sd = 6.54)
print(paste("L(y = A | x3 = 195):", round(L_A, 6)))
```

```
## [1] "L(y = A | x3 = 195): 0.045542"
```

```
# Species C (Chinstrap): mean = 196, sd = 7.13
L_C <- dnorm(195, mean = 196, sd = 7.13)
print(paste("L(y = C | x3 = 195):", round(L_C, 6)))
```

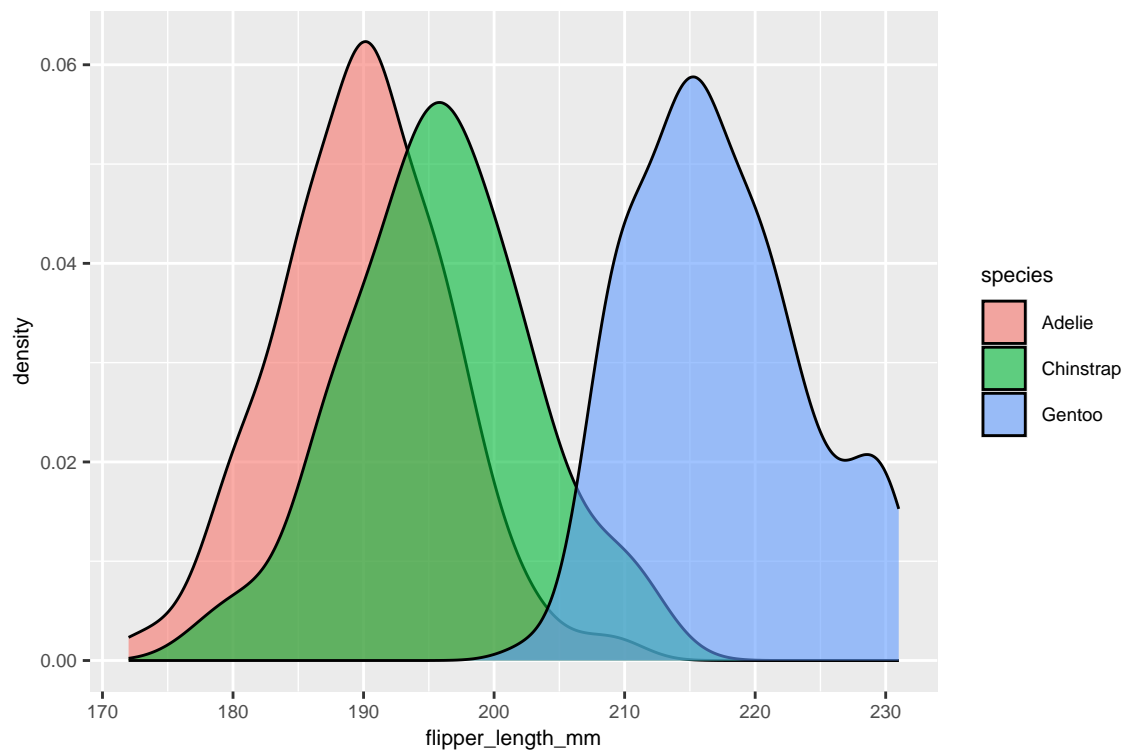
```
## [1] "L(y = C | x3 = 195): 0.055405"
```

```
# Species G (Gentoo): mean = 217, sd = 6.48
L_G <- dnorm(195, mean = 217, sd = 6.48)
print(paste("L(y = G | x3 = 195):", round(L_G, 6)))
```

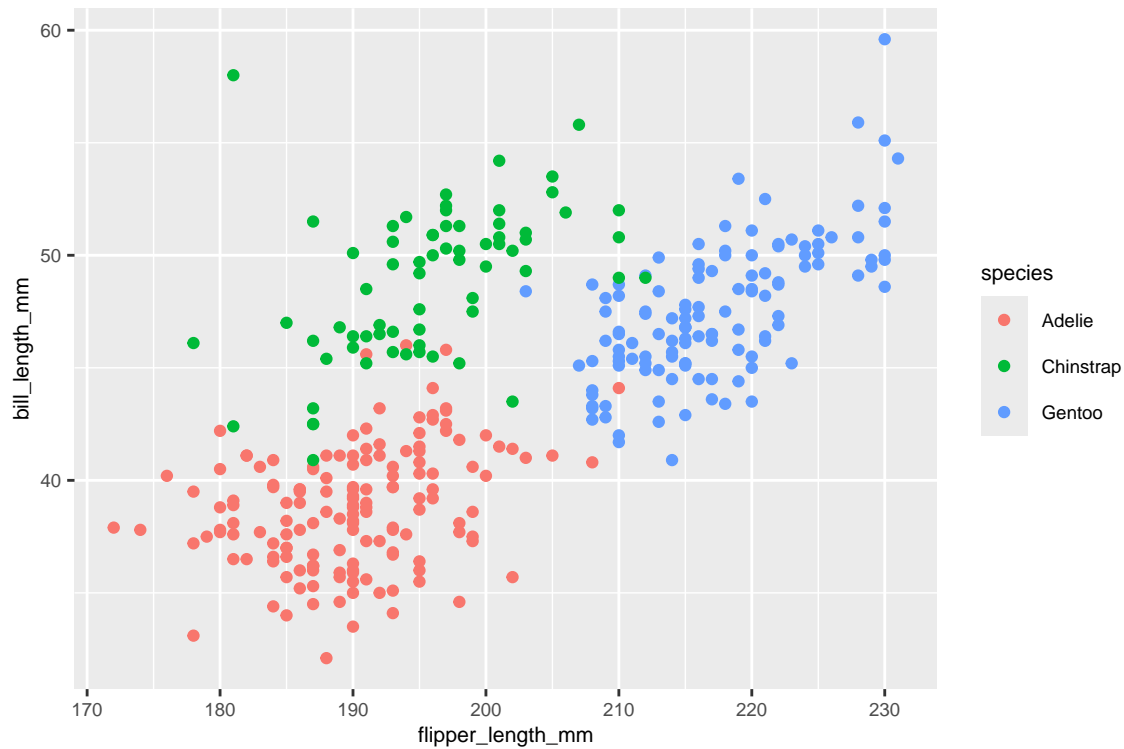
```
## [1] "L(y = G | x3 = 195): 0.000193"
```

Naive Bayes Classification with Two Predictors

```
#### Density Plots for Flipper Lengths and Scatterplot for Two Predictors
# Density plots for flipper lengths
ggplot(penguins, aes(x = flipper_length_mm, fill = species)) +
  geom_density(alpha = 0.6)
```



```
# Scatterplot for bill length vs flipper length
ggplot(penguins,
  aes(x = flipper_length_mm, y = bill_length_mm, color = species)) +
  geom_point()
```



```
#### Calculate Sample Mean and Standard Deviation for Flipper Lengths
# Sample mean and sd for flipper length
penguins %>%
  group_by(species) %>%
  summarize(mean = mean(flipper_length_mm, na.rm = TRUE),
            sd = sd(flipper_length_mm, na.rm = TRUE))
```

```
## # A tibble: 3 x 3
##   species    mean    sd
##   <fct>    <dbl> <dbl>
## 1 Adelie   190.   6.54
## 2 Chinstrap 196.   7.13
## 3 Gentoo   217.   6.48
```

```
#### Evaluate Likelihood of Observing Specific Flipper Length
# Likelihoods for flipper length
dnorm(195, mean = 190, sd = 6.54) # Adelie
```

```
## [1] 0.04554175
```

```
dnorm(195, mean = 196, sd = 7.13) # Chinstrap
```

```
## [1] 0.05540502
```

```
dnorm(195, mean = 217, sd = 6.48) # Gentoo
```

```
## [1] 0.0001933746
```

Naive Bayes Classification using the naiveBayes function in e1071 package

We build the following Naive Bayes Models:

- Model 1: Using above_average_weight as predictor
- Model 2: Using bill length as predictor
- Model 3: Using bill length and flipper length as predictors
- Model 4: Including weight information in addition to bill and flipper lengths

```
# Load necessary library
library(e1071)

# Building Naive Bayes Models

# Model 1: Using above_average_weight as predictor
naive_mod1 <- naiveBayes(species ~ above_average_weight, data = penguins)

# Model 2: Using bill length as predictor
naive_mod2 <- naiveBayes(species ~ bill_length_mm, data = penguins)

# Model 3: Using bill length and flipper length as predictors
naive_mod3 <- naiveBayes(species ~ bill_length_mm + flipper_length_mm, data = penguins)

# Model 4: Including weight information in addition to bill and flipper lengths
naive_mod4 <- naiveBayes(species ~ above_average_weight + bill_length_mm + flipper_length_mm, data = penguins)

# Making Predictions for a New Observation

# Define a new penguin with specific measurements
our_penguin <- data.frame(above_average_weight = 'O', bill_length_mm = 50, flipper_length_mm = 195)

# Predictions using Model 1 (above_average_weight only)
cat("Posterior probabilities (Model 1):\n")
```

```
## Posterior probabilities (Model 1):
```

```
print(predict(naive_mod1, newdata = our_penguin, type = "raw"))
```

```
##           Adelie Chinstrap      Gentoo
## [1,] 0.6541796 0.3146224 0.03119806
```

```
cat("Predicted class (Model 2):\n")
```

```
## Predicted class (Model 2):
```

```
print(predict(naive_mod2, newdata = our_penguin, type = "raw"))
```

```
##           Adelie Chinstrap      Gentoo
## [1,] 0.0001690279 0.3978306 0.6020004
```

```
print(predict(naive_mod2, newdata = our_penguin))
```

```
## [1] Gentoo
## Levels: Adelie Chinstrap Gentoo
```

```
# Predictions using Model 3 (bill length + flipper length)
cat("\nPosterior probabilities (Model 3):\n")
```

```
##
## Posterior probabilities (Model 3):
```

```
print(predict(naive_mod3, newdata = our_penguin, type = "raw"))
```

```
##           Adelie Chinstrap           Gentoo
## [1,] 0.0003445688 0.9948681 0.004787365
```

```
cat("Predicted class (Model 3):\n")
```

```
## Predicted class (Model 3):
```

```
print(predict(naive_mod3, newdata = our_penguin))
```

```
## [1] Chinstrap
## Levels: Adelie Chinstrap Gentoo
```

```
# Predictions using Model 4 (above_average_weight + bill length + flipper length)
cat("\nPosterior probabilities (Model 4):\n")
```

```
##
## Posterior probabilities (Model 4):
```

```
print(predict(naive_mod4, newdata = our_penguin, type = "raw"))
```

```
##           Adelie Chinstrap           Gentoo
## [1,] 0.0003219805 0.9994165 0.0002615187
```

```
cat("Predicted class (Model 4):\n")
```

```
## Predicted class (Model 4):
```

```
print(predict(naive_mod3, newdata = our_penguin))
```

```
## [1] Chinstrap
## Levels: Adelie Chinstrap Gentoo
```



```

# In-sample Predictions for All Penguins

# Generate predicted classifications for the entire sample using all models
penguins <- penguins %>%
  mutate(class_1 = predict(naive_mod1, newdata = .),
         class_2 = predict(naive_mod2, newdata = .),
         class_3 = predict(naive_mod3, newdata = .),
         class_4 = predict(naive_mod4, newdata = .))

# Results and Summary

cat("\nIn-sample classifications have been added as new columns:\n")

##
## In-sample classifications have been added as new columns:

print(head(penguins[, c("species", "class_1", "class_2", "class_3", "class_4")]))

## # A tibble: 6 x 5
##   species class_1 class_2 class_3 class_4
##   <fct>   <fct>   <fct>   <fct>   <fct>
## 1 Adelie Adelie Adelie Adelie Adelie
## 2 Adelie Adelie Adelie Adelie Adelie
## 3 Adelie Adelie Adelie Adelie Adelie
## 4 Adelie Adelie Adelie Adelie Adelie
## 5 Adelie Adelie Adelie Adelie Adelie
## 6 Adelie Adelie Adelie Adelie Adelie

```

Confusion Matrices for In-sample Predictions

```

# Function to generate and format confusion matrices
generate_confusion_matrix <- function(data, species_col, predicted_col) {
  data %>%
    tabyl(!!sym(species_col), !!sym(predicted_col)) %>%
    adorn_percentages("row") %>%
    adorn_pct_formatting(digits = 2) %>%
    adorn_ns()
}

# Confusion matrix for naive_mod1
cat("\nConfusion Matrix for naive_mod1:\n")

##
## Confusion Matrix for naive_mod1:

print(generate_confusion_matrix(penguins, "species", "class_1"))

##   species      Adelie Chinstrap      Gentoo
##   Adelie 83.55% (127) 0.00% (0) 16.45% (25)
##   Chinstrap 89.71% (61) 0.00% (0) 10.29% (7)
##   Gentoo 5.65% (7) 0.00% (0) 94.35% (117)

```

```
# Confusion matrix for naive_mod2
cat("\nConfusion Matrix for naive_mod2:\n")

##
## Confusion Matrix for naive_mod2:

print(generate_confusion_matrix(penguins, "species", "class_2"))

##      species      Adelie Chinstrap      Gentoo
##      Adelie 95.39% (145) 0.00% (0) 4.61% (7)
##      Chinstrap 5.88% (4) 8.82% (6) 85.29% (58)
##      Gentoo 6.45% (8) 4.84% (6) 88.71% (110)
```

```
# Confusion matrix for naive_mod3
cat("\nConfusion Matrix for naive_mod3:\n")

##
## Confusion Matrix for naive_mod3:

print(generate_confusion_matrix(penguins, "species", "class_3"))

##      species      Adelie Chinstrap      Gentoo
##      Adelie 96.05% (146) 2.63% (4) 1.32% (2)
##      Chinstrap 7.35% (5) 86.76% (59) 5.88% (4)
##      Gentoo 0.81% (1) 0.81% (1) 98.39% (122)
```

```
# Confusion matrix for naive_mod4
cat("\nConfusion Matrix for naive_mod4:\n")

##
## Confusion Matrix for naive_mod4:

print(generate_confusion_matrix(penguins, "species", "class_4"))

##      species      Adelie Chinstrap      Gentoo
##      Adelie 96.05% (146) 2.63% (4) 1.32% (2)
##      Chinstrap 7.35% (5) 86.76% (59) 5.88% (4)
##      Gentoo 0.81% (1) 4.03% (5) 95.16% (118)
```

Cross-Validation for Classification Accuracy

```
# Perform cross-validation for each model
cat("\nCross-validation classification accuracy (k=5):\n")

##
## Cross-validation classification accuracy (k=5):
```

```
# Cross-validation for naive_mod1
CV_mod1 <- naive_classification_summary_cv(
  model = naive_mod1, data = penguins, y = "species", k = 5)
cat("naive_mod1 Cross-validation Accuracy:\n")
```

```
## naive_mod1 Cross-validation Accuracy:
```

```
CV_mod1$cv
```

```
##   species      Adelie Chinstrap      Gentoo
##   Adelie 83.55% (127) 0.00% (0) 16.45% (25)
##   Chinstrap 89.71% (61) 0.00% (0) 10.29% (7)
##   Gentoo 5.65% (7) 0.00% (0) 94.35% (117)
```

```
# Cross-validation for naive_mod2
CV_mod2 <- naive_classification_summary_cv(
  model = naive_mod2, data = penguins, y = "species", k = 5)
cat("naive_mod2 Cross-validation Accuracy:\n")
```

```
## naive_mod2 Cross-validation Accuracy:
```

```
CV_mod2$cv
```

```
##   species      Adelie Chinstrap      Gentoo
##   Adelie 95.39% (145) 0.00% (0) 4.61% (7)
##   Chinstrap 5.88% (4) 1.47% (1) 92.65% (63)
##   Gentoo 6.45% (8) 4.84% (6) 88.71% (110)
```

```
# Cross-validation for naive_mod3
CV_mod3 <- naive_classification_summary_cv(
  model = naive_mod3, data = penguins, y = "species", k = 5)
cat("naive_mod3 Cross-validation Accuracy:\n")
```

```
## naive_mod3 Cross-validation Accuracy:
```

```
CV_mod3$cv
```

```
##   species      Adelie Chinstrap      Gentoo
##   Adelie 96.05% (146) 2.63% (4) 1.32% (2)
##   Chinstrap 7.35% (5) 85.29% (58) 7.35% (5)
##   Gentoo 0.81% (1) 0.81% (1) 98.39% (122)
```

```
# Cross-validation for naive_mod3
CV_mod4 <- naive_classification_summary_cv(
  model = naive_mod4, data = penguins, y = "species", k = 5)
cat("naive_mod4 Cross-validation Accuracy:\n")
```

```
## naive_mod4 Cross-validation Accuracy:
```

```
CV_mod4$cv
```

```
##      species      Adelie  Chinstrap      Gentoo
##      Adelie 95.39% (145)  3.29% (5)  1.32% (2)
## Chinstrap  7.35% (5) 86.76% (59)  5.88% (4)
##      Gentoo 0.81% (1) 4.03% (5) 95.16% (118)
```