Evaluating Classification Model Performance — Penguins

Sachi Kapoor

Questions

1) What is the null error rate of the dataset?
2) How do confusion matrix counts (TP, FP, TN, FN) change at thresholds 0.2, 0.5, 0.8?
3) What are the accuracy, precision, recall, and F1 for each threshold?
4) Which threshold would you choose and why?

Setup

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(janitor)
```

```
Attaching package: 'janitor'
```

```
The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```r
library(readr)
library(purrr)

# Load the penguin predictions dataset
url_csv <- "https://raw.githubusercontent.com/acatlin/data/master/penguin_predictions.csv"
peng <- read_csv(url_csv, show_col_types = FALSE) |> clean_names()

glimpse(peng)
```

```
Rows: 93
Columns: 3
$ pred_female <dbl> 0.99217462, 0.95423945, 0.98473504, 0.18702056, 0.99470123~
$ pred_class  <chr> "female", "female", "female", "male", "female", "female", ~
$ sex         <chr> "female", "female", "female", "female", "female", "female"~
```

1) Null error rate + distribution plot

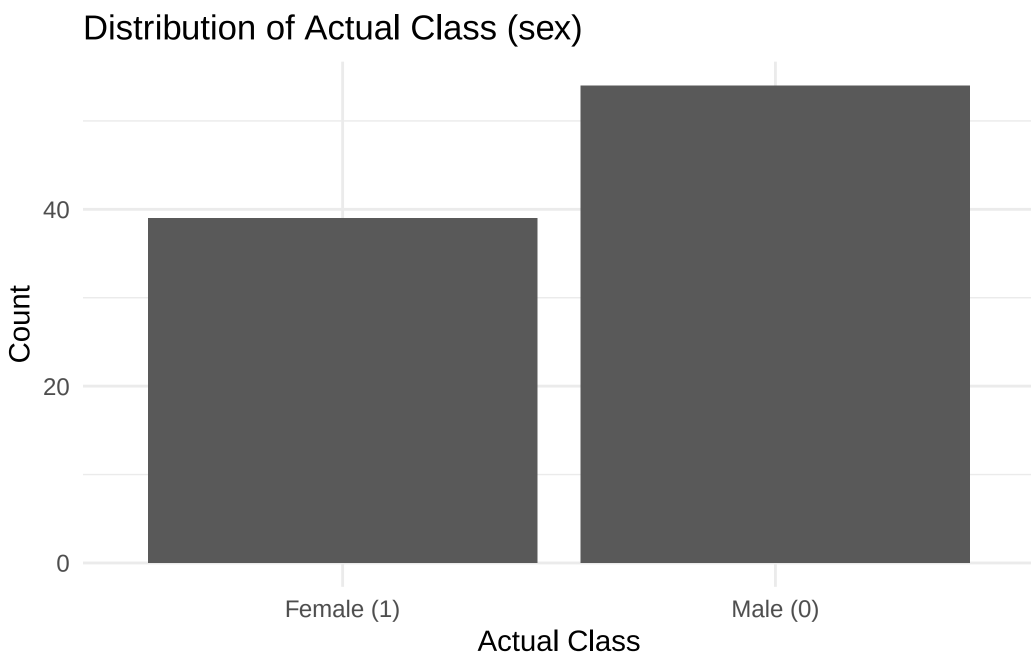
```r
# Majority class
maj_class <- peng |> count(sex) |> arrange(desc(n)) |> slice(1) |> pull(sex)

# Null error rate
n_total <- nrow(peng)
n_maj   <- sum(peng$sex == maj_class)
null_error_rate <- 1 - (n_maj / n_total)
maj_class
```

```
[1] "male"
```

```r
null_error_rate
```

```
[1] 0.4193548
```

```r
peng |>
  mutate(
    sex_clean = tolower(trimws(as.character(sex))),
    sex_label = case_when(
      sex_clean %in% c("female","f","1") ~ "Female (1)",
      sex_clean %in% c("male","m","0")   ~ "Male (0)",
      TRUE ~ "Unknown"
    )
  ) |>
  ggplot(aes(x = sex_label)) +
  geom_bar() +
  labs(title = "Distribution of Actual Class (sex)",
       x = "Actual Class", y = "Count") +
  theme_minimal()
```




```r
conf_counts <- function(df, thr){
  df2 <- df |>
    mutate(
      actual = case_when(
        is.numeric(sex) ~ as.integer(sex),
        tolower(as.character(sex)) %in% c("female","f","1") ~ 1L,
        tolower(as.character(sex)) %in% c("male","m","0") ~ 0L,
        TRUE ~ NA_integer_
```

```r
      ),
      pred = if_else(pred_female >= thr, 1L, 0L)
    )

  df2 |>
    summarize(
      TP = sum(pred == 1 & actual == 1, na.rm = TRUE),
      FP = sum(pred == 1 & actual == 0, na.rm = TRUE),
      TN = sum(pred == 0 & actual == 0, na.rm = TRUE),
      FN = sum(pred == 0 & actual == 1, na.rm = TRUE)
    )
}

ths <- c(0.2, 0.5, 0.8)
conf_list <- setNames(lapply(ths, \(t) conf_counts(peng, t)), ths)
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `actual = case_when(...)`.
Caused by warning:
! NAs introduced by coercion
There was 1 warning in `mutate()`.
i In argument: `actual = case_when(...)`.
Caused by warning:
! NAs introduced by coercion
There was 1 warning in `mutate()`.
i In argument: `actual = case_when(...)`.
Caused by warning:
! NAs introduced by coercion
```

```r
conf_list
```

```
$`0.2`
# A tibble: 1 x 4
     TP    FP    TN    FN
  <int> <int> <int> <int>
1    37     6    48     2

$`0.5`
# A tibble: 1 x 4
     TP    FP    TN    FN
  <int> <int> <int> <int>
```

```
1     36      3     51      3

$`0.8`
# A tibble: 1 x 4
     TP    FP    TN    FN
  <int> <int> <int> <int>
1    36     2    52     3
```

3) Accuracy, Precision, Recall, F1

2 * prec * rec / (prec + rec)) tibble(accuracy = acc, precision = prec, recall = rec, f1 = f1)
}

```
metrics_from_counts <- function(cc){
  TP <- cc$TP; FP <- cc$FP; TN <- cc$TN; FN <- cc$FN
  acc  <- (TP + TN) / (TP + FP + TN + FN)
  prec <- ifelse(TP + FP == 0, NA_real_, TP / (TP + FP))
  rec  <- ifelse(TP + FN == 0, NA_real_, TP / (TP + FN))
  f1   <- ifelse(is.na(prec) | is.na(rec) | (prec + rec) == 0, NA_real_,
                 2 * prec * rec / (prec + rec))
  tibble(accuracy = acc, precision = prec, recall = rec, f1 = f1)
}

metrics_tbl <-
  tibble(threshold = ths) |>
  mutate(cc = conf_list) |>
  mutate(metrics = purrr::map(cc, metrics_from_counts)) |>
  select(threshold, metrics) |>
  unnest(metrics)

metrics_tbl
```

```
# A tibble: 3 x 5
  threshold accuracy precision recall    f1
      <dbl>    <dbl>     <dbl>  <dbl> <dbl>
1       0.2    0.914     0.860  0.949 0.902
2       0.5    0.935     0.923  0.923 0.923
3       0.8    0.946     0.947  0.923 0.935
```

4) Threshold choice

- 0.2 → higher recall, more false positives (good when missing positives is costly).
- 0.5 → balanced; compare F1 if FP/FN costs are similar.

- 0.8 → higher precision, more false negatives (good when false alarms are costly).

Conclusions

- Null error rate is the baseline to beat.
- Raising the threshold lowers recall and raises precision.
- Pick the threshold by the cost of FP vs FN for the use case.