SPS_Data607_Week2_2B

David Chen

Evaluating Classification Model Performance

Analyze the performance of a binary classification model and develop intuition for how probability thresholds affect model evaluation metrics.

Approach

I need to understand what a classification problem is and how classification is used. then apply classification performance metrics to evaluate whether the model performs well or poorly. According to the PDF and LLM there are a few common algorithms and evaluation metrics to use. like Accuracy, precision and recall etc.

Dataset

https://raw.githubusercontent.com/acatlin/data/refs/heads/master/penguin_predictions.csv

The dataset contains three columns:

```
.pred_female - Model-predicted probability that the observation belongs to the "female" clas

.pred_class - Predicted class label (1 if .pred_female > 0.5, otherwise 0)

sex - Actual class label used during model training
```

Running Code

```r
library(dplyr)
```

```
Attaching package: 'dplyr'

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

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.1     v stringr   1.6.0
v lubridate 1.9.4     v tibble    3.3.1
v purrr     1.2.1     v tidyr     1.3.2
v readr     2.1.6

-- 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 becor
```

```r
mydata<- read.csv("https://raw.githubusercontent.com/acatlin/data/refs/heads/master/penguin_p
#print(mydata)
head(mydata)
```

```
  .pred_female .pred_class    sex
1    0.9921746      female female
2    0.9542394      female female
3    0.9847350      female female
4    0.1870206        male female
5    0.9947012      female female
6    0.9999891      female female
```

```r
glimpse(mydata)
```

```
Rows: 93
Columns: 3
$ .pred_female <dbl> 0.99217462, 0.95423945, 0.98473504, 0.18702056, 0.9947012~
$ .pred_class  <chr> "female", "female", "female", "male", "female", "female",~
$ sex          <chr> "female", "female", "female", "female", "female", "female~
```

```r
summary(mydata)
```

```
  .pred_female       .pred_class            sex
 Min.   :0.0000000   Length:93          Length:93
 1st Qu.:0.0003508   Class :character   Class :character
 Median :0.1098907   Mode  :character   Mode  :character
 Mean   :0.4351396
 3rd Qu.:0.9921746
 Max.   :1.0000000
```

Grouping by the sex and count

```r
#Getting each count by sex
mydata %>%
  group_by(sex) %>%
  summarise(total_count = n())
```

```
# A tibble: 2 x 2
  sex    total_count
  <chr>        <int>
1 female          39
2 male            54
```

Task 1 Null Error Rate

```r
tbl <- table(mydata$sex)
tbl
```

```
female   male
    39     54
```

```r
majority_class <- names(tbl)[which.max(tbl)]
majority_count <-max(tbl)
majority_class
```

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

```r
majority_count
```

```
[1] 54
```

```r
null_error_rate <- 1 - max(tbl) / sum(tbl)
null_error_rate
```

```
[1] 0.4193548
```

Then we will get the Null Error Rate is **0.4193548**

Task 2 Confusion Matrices at Multiple Thresholds

Our data set has predicted class label and actual class label.

the predicted label is looking for female, so TP will if predicted is female and the actual label is female. then follow the concept TP,FP,TN,FN to do calculation.

```r
TP <- sum(mydata$sex == "female" & mydata$.pred_class == "female")
FP <- sum(mydata$sex == "male" & mydata$.pred_class == "female")
TN <- sum(mydata$sex == "male" & mydata$.pred_class == "male")
FN <- sum(mydata$sex == "female" & mydata$.pred_class == "male")

TP; FP; TN; FN
```

```
[1] 36
```

```
[1] 3
```

```
[1] 51
```

```
[1] 3
```

now moving to different probability thresholds to compute 0.2 / 0.5 / 0.8 into 3 confusion matrices.

```
mydata$pred_0.2 <- ifelse(mydata$.pred_female > 0.2,"female","male")
mydata$pred_0.5 <- ifelse(mydata$.pred_female > 0.5,"female","male")
mydata$pred_0.8 <- ifelse(mydata$.pred_female > 0.8,"female","male")
#print(mydata)
```

```
TP <- sum(mydata$sex == "female" & mydata$pred_0.2 == "female")
FP <- sum(mydata$sex == "male" & mydata$pred_0.2 == "female")
TN <- sum(mydata$sex == "male" & mydata$pred_0.2 == "male")
FN <- sum(mydata$sex == "female" & mydata$pred_0.2 == "male")
TP; FP; TN; FN
```

```
[1] 37

[1] 6

[1] 48

[1] 2
```

```
TP <- sum(mydata$sex == "female" & mydata$pred_0.5 == "female")
FP <- sum(mydata$sex == "male" & mydata$pred_0.5 == "female")
TN <- sum(mydata$sex == "male" & mydata$pred_0.5 == "male")
FN <- sum(mydata$sex == "female" & mydata$pred_0.5 == "male")
TP; FP; TN; FN
```

```
[1] 36

[1] 3

[1] 51

[1] 3
```

```r
TP <- sum(mydata$sex == "female" & mydata$pred_0.8 == "female")
FP <- sum(mydata$sex == "male" & mydata$pred_0.8 == "female")
TN <- sum(mydata$sex == "male" & mydata$pred_0.8 == "male")
FN <- sum(mydata$sex == "female" & mydata$pred_0.8 == "male")
TP; FP; TN; FN
```

```
[1] 36
```

```
[1] 2
```

```
[1] 52
```

```
[1] 3
```

```r
table(mydata$sex, mydata$pred_0.2)
```

```
         female male
  female     37    2
  male        6   48
```

```r
table(mydata$sex, mydata$pred_0.5)
```

```
         female male
  female     36    3
  male        3   51
```

```r
table(mydata$sex, mydata$pred_0.8)
```

```
         female male
  female     36    3
  male        2   52
```

Task 3 Metrics table

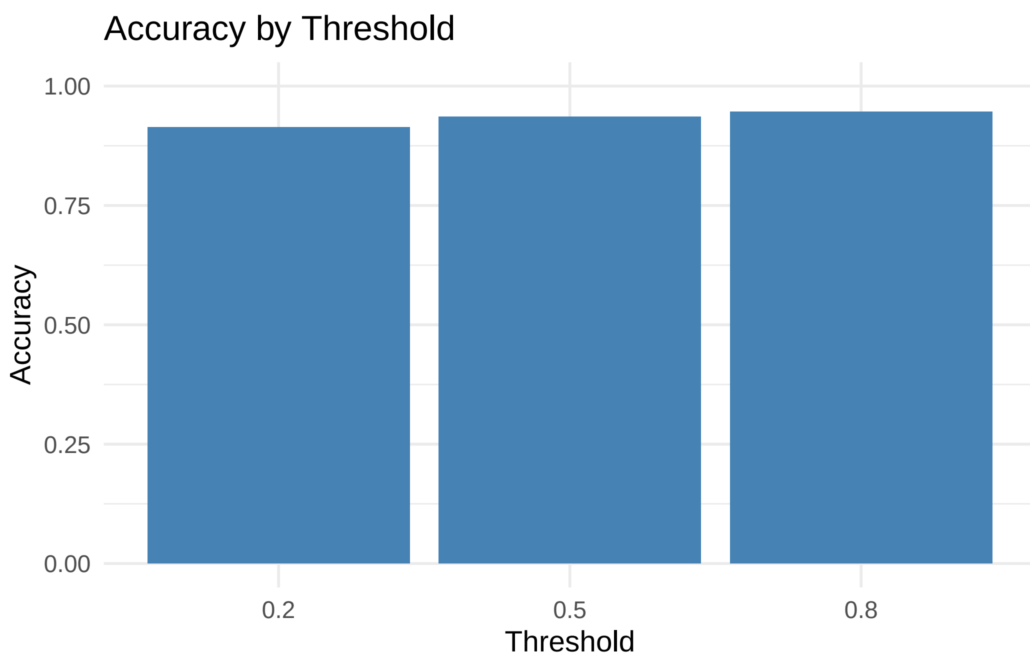
It all depends on which class we choose as the positive class. If we swap it (Female = positive), then TP/FP/TN/FN swap meaning too.

```
# Thresholds to test
thresholds <- c(0.2, 0.5, 0.8)

# Prepare empty data frame to store metrics
metrics <- data.frame(
  Threshold = thresholds,
  TP = NA, FP = NA, TN = NA, FN = NA,
  Accuracy = NA,
  Precision = NA,
  Recall = NA
)

# Loop over thresholds
for (i in seq_along(thresholds)) {
  thresh <- thresholds[i]

  # Convert probabilities to predicted labels based on threshold
  pred <- ifelse(mydata$.pred_female > thresh, "female", "male")

  # Confusion matrix
  cm <- table(mydata$sex, pred)

  # Extract TP, FP, TN, FN (make sure table has all levels)
#  TP <- ifelse("male" %in% rownames(cm) & "male" %in% colnames(cm), cm["male","male"], 0)
#  FP <- ifelse("female" %in% rownames(cm) & "male" %in% colnames(cm), cm["female","male"], 
#  TN <- ifelse("female" %in% rownames(cm) & "female" %in% colnames(cm), cm["female","female"
#  FN <- ifelse("male" %in% rownames(cm) & "female" %in% colnames(cm), cm["male","female"], 
# as LLMs mentioned it all depends on which class you choose as the positive class. If you sw

  TP <- ifelse("female" %in% rownames(cm) & "female" %in% colnames(cm), cm["female","female"]
  FP <- ifelse("male" %in% rownames(cm) & "female" %in% colnames(cm), cm["male","female"], 0)

  TN <- ifelse("male" %in% rownames(cm) & "male" %in% colnames(cm), cm["male","male"], 0)
  FN <- ifelse("female" %in% rownames(cm) & "male" %in% colnames(cm), cm["female","male"], 0)


  # Store confusion metrics
  metrics$TP[i] <- TP
```

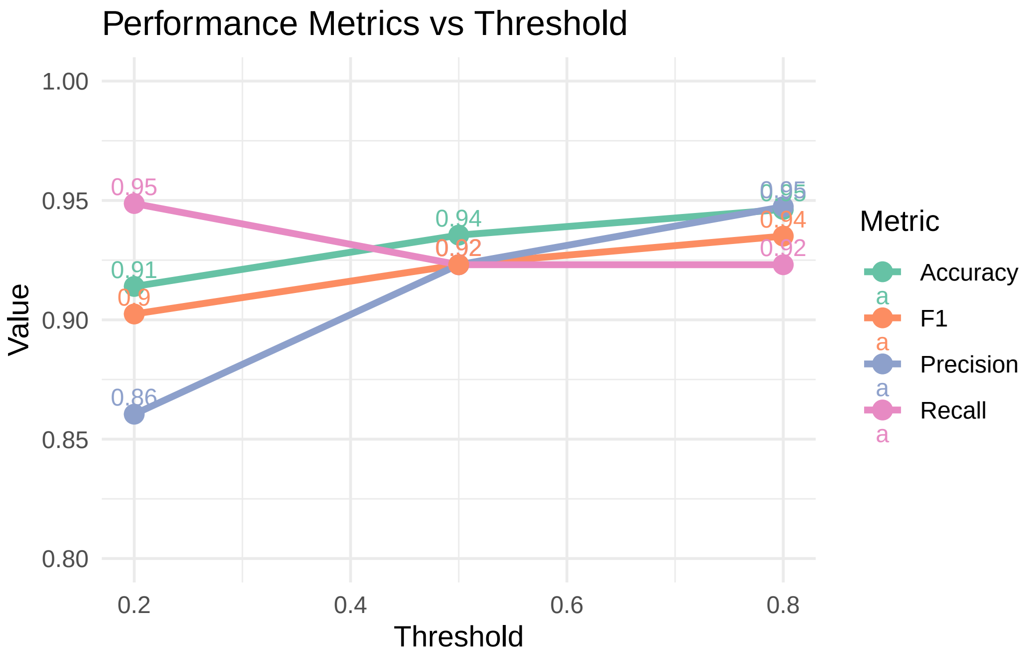
```r
  metrics$FP[i] <- FP
  metrics$TN[i] <- TN
  metrics$FN[i] <- FN

  # Compute performance metrics
  metrics$Accuracy[i]  <- (TP + TN) / (TP + TN + FP + FN)
  metrics$Precision[i] <- ifelse((TP + FP) > 0, TP / (TP + FP), NA)
  metrics$Recall[i]    <- ifelse((TP + FN) > 0, TP / (TP + FN), NA)
  # Add f1
  metrics$F1[i] <- 2* metrics$Precision[i]*metrics$Recall[i] /(metrics$Precision[i]+metrics$R
}

# Show final table
metrics
```

```
  Threshold TP FP TN FN  Accuracy Precision    Recall        F1
1       0.2 37  6 48  2 0.9139785 0.8604651 0.9487179 0.9024390
2       0.5 36  3 51  3 0.9354839 0.9230769 0.9230769 0.9230769
3       0.8 36  2 52  3 0.9462366 0.9473684 0.9230769 0.9350649
```

```r
ggplot(metrics, aes(x = factor(Threshold), y = Accuracy)) +
  geom_col(fill = "steelblue") +

  labs(
    title = "Accuracy by Threshold",
    x = "Threshold",
    y = "Accuracy"
  ) +
  ylim(0, 1) +
  theme_minimal()
```




```r
metrics_long <- metrics %>%
  pivot_longer(cols = c(Accuracy, Precision, Recall, F1),
               names_to = "Metric",
               values_to = "Value")

ggplot(metrics_long, aes(x = Threshold, y = Value, color = Metric)) +
  geom_line(linewidth  = 1.2) +
  geom_point(size = 3) +
  geom_text(aes(label = round(Value, 2)),   # show values rounded to 2 decimals
            vjust = -0.5,                   # position above the point
            size = 3)+
  labs(
    title = "Performance Metrics vs Threshold",
    x = "Threshold",
    y = "Value"
  ) +
   coord_cartesian(ylim = c(0.8, 1)) +  # ylim(0,1) +
  theme_minimal() +
  scale_color_brewer(palette = "Set2")
```



Task 4 Threshold Use cases

Threshold selection is a key decision because a lower threshold will classify more predictions as positive, while a higher threshold will classify fewer predictions as positive.

For example, in a company benefits policy, a lower threshold allows more employees to qualify for benefits. Conversely, in the hiring process, the company may increase the threshold to filter out more candidates.