DATA 621—Assignment no. 2

Critical Thinking Group 2

October 10, 2019

Contents

Executive Overview

We explore metrics to evaluate the performance of classification algorithms, including custom R functions, on a real dataset:

- Accuracy
- Classification error rate
- Precision
- Sensitivity
- F1 score
- ROC curve

We also examine the functionality of the `caret` and `pROC` packages.

2. Confusion matrix

3. Accuracy

4. Classification error rate

5. Precision

Precision measures how reliable a model's positive predictions are, i.e., $P(Positive \mid model = Positive)$. A poor (low) precision score indicates many of the positive predictions are likely to be *false* positives. For instance, if you have 100 positive (spam) predictions from a spam e-mail detection model, how many of them are actually positive (spam)? Thus the formula is:

$$Precision = \frac{TP}{TP + FP}$$

R function:

```
calc_precision <- function(actual, predicted, pos=1) {
    if (length(unique(actual)) != 2 | length(unique(predicted)) != 2) {
        stop('Actual and predicted vectors may only have two distinct values')
    }

    # set pos/neg to TRUE/FALSE as conveniance
    actual <- ifelse(actual == pos, TRUE, FALSE)
    predicted <- ifelse(predicted == pos, TRUE, FALSE)

    conf_matrix <- as.data.frame(table(actual, predicted)) %>%
        arrange(actual, predicted)

    TP <- conf_matrix$Freq[4]
    FP <- conf_matrix$Freq[2]

    return( TP / (TP + FP) )

}
```

Applied to data set:

```
calc_precision(df$class, df$scored.class)
```

```
## [1] 0.84375
```

Thus the model that scored this dataset performs fairly reliabily on positive cases.

6. Sensitivity

Also called recall, this measures how often a model will return a positive result given a positive example. In a way, it is the reverse of precision, i.e., sensitivity is $P(model = Positive \mid Positive)$. For instance, if a spam e-mail detection model was given 100 spam e-mails, how many will the model label as spam (positive)? The resulting number is sensitivity. It is calculated by:

$$Sensitivity = \frac{TP}{TP + FN}$$

where a score of 1 indicates perfect sensitivity, the event where there are no false negatives.

Function to calculate sensitivity:

```
calc_sensitivity <- function(actual, predicted, pos=1) {
    if (length(unique(actual)) != 2 | length(unique(predicted)) != 2) {
        stop('Actual and predicted vectors may only have two distinct values')
    }

    # set pos/neg to TRUE/FALSE as conveniance
    actual <- ifelse(actual == pos, TRUE, FALSE)
    predicted <- ifelse(predicted == pos, TRUE, FALSE)

    conf_matrix <- as.data.frame(table(actual, predicted)) %>%
        arrange(actual, predicted)

    TP <- conf_matrix$Freq[4]
    FN <- conf_matrix$Freq[3]

    return( TP / (TP + FN) )

}
```

Applied to dataframe:

```
calc_sensitivity(df$class, df$scored.class)
```

```
## [1] 0.4736842
```

7. Specificity

8. F1 score

The F1 score is an attempt to balance sensitivity and precision, perhaps in cases where costs of false negative and false positive predictions are comparable. The formula is:

$$F1 = 2 \cdot \frac{Precision \cdot Sensitivity}{Precision + Sensitivity}$$

R function:

```
calc_f1 <- function(actual, predicted, pos=1) {
    if (length(unique(actual)) != 2 | length(unique(predicted)) != 2) {
        stop('Actual and predicted vectors may only have two distinct values')
    }

    # set pos/neg to TRUE/FALSE as conveniance
    actual <- ifelse(actual == pos, TRUE, FALSE)
    predicted <- ifelse(predicted == pos, TRUE, FALSE)
```

```
    # truncating in this annoying manner to avoid package conflicts with caret
    precis <- calc_precision(actual, predicted, pos=pos)
    sensit <- calc_sensitivity(actual, predicted, pos=pos)

    return( 2 * ( (precis * sensit) / (precis + sensit) ) )
}
```

The F1 score for the scored dataset is:

```
calc_f1(df$class, df$scored.class)
```

```
## [1] 0.6067416
```

9. F1 score: Bounds

10. ROC curve

11. In Practice

12. caret package

13. pROC package