Prevalence in Practice & ROC and Precision-Recall Curves

Machine Learning - Sections 2.1.4 & 2.1.5

Marc Omar Haddad

20 January, 2020

2.1.4: Prevalence Matters in Practice

An algorithm that yields a high Sensitivity or Specificity is **not** necessarily useful in practice when **prevalence** is close to either 0 or 1.

For example:

- A doctor that specializes in a rare disease is interested in developing an algorithm that can predict who has the disease. Assume we are tasked with creating the algorithm from data provided by the doctor.
- We create an algorithm with **high Sensitivity** which means that *if a patient has the disease, our algorithm is likely to predict its existence correctly.*
- We also come to the worrying conclusion that the data set provided by the doctor indicates that 50% of patients have the, supposedly rare, disease: $Pr(\hat{Y} = 1) = \frac{1}{2}$.
- The doctor then explains that what is actually important is the **Precision** of the test: $Pr(Y = 1 \mid \hat{Y} = 1)$.
- Using Bayes' theorem, we can actually show a link between both measures:

$$Pr(Y = 1 \mid \hat{Y} = 1) = Pr(\hat{Y} = 1 \mid Y = 1)\frac{Pr(Y = 1)}{Pr(\hat{Y} = 1)}$$

- The doctor knows that the **Prevalence** is actually 5 in 1000 which means that **actual outcome** $Pr(Y = 1)$ is $\frac{1}{200}$.
- Remembering that our **predicted outcome** $Pr(\hat{Y} = 1)$ is $\frac{1}{2}$, we find that the final fraction in our Bayesian equation above $\frac{Pr(Y=1)}{Pr(\hat{Y}=1)}$ is $\frac{\frac{1}{200}}{\frac{1}{2}} = \frac{1}{100}$.
- Keeping in mind that $Pr(\hat{Y} = 1 \mid Y = 1)$ is a value between 0 and 1: our **Prevalence** of $\frac{1}{100}$ means that our **Precision** $Pr(Y = 1 \mid \hat{Y} = 1)$ is going to be **less than** 0.01.
- Thus, our algorithm is useless to the doctor.

2.1.5: ROC and Precision-Recall Curves

We used **accuracy** and **F1 Score** to compare our competing algorithms of guessing vs. height cutoff. We found that our height cutoff algorithm clearly outperformed the other. However, we neglected to test *other* values within our guessing algorithm (we used $p = 0.5$). Setting our p to 0.9, we get a vastly better result with just guessing due to the fact that our data set is biased:

```
p = 0.9
y_hat = sample(c("Male", "Female"), length(test_index),
               replace = TRUE, prob = c(p, 1-p)) %>%
  factor(levels = levels(test_set$sex))
mean(y_hat == test_set$sex)
```

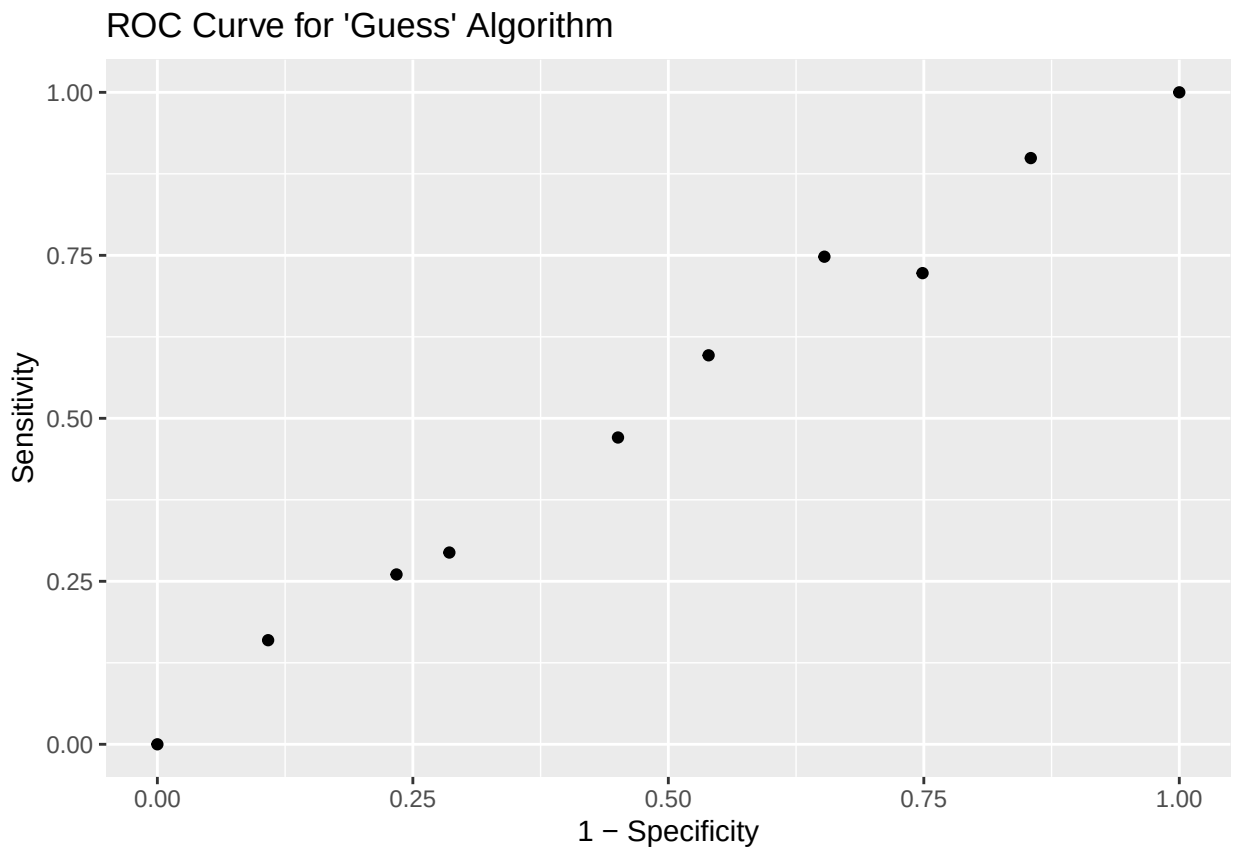
```
## [1] 0.691
```

However, even though we obtained a relatively high accuracy, it was obtained at the cost of lower Sensitivity. The curves below will help us compare the effects of varying values on different algorithms and visually aid us in assessing which one is best.

Receiver Operating Characteristic (ROC) Curve

The ROC curve plots **Sensitivity (TPR)** versus 1 **- Specificity**, also known as the **False Positive Rate (FPR)**.

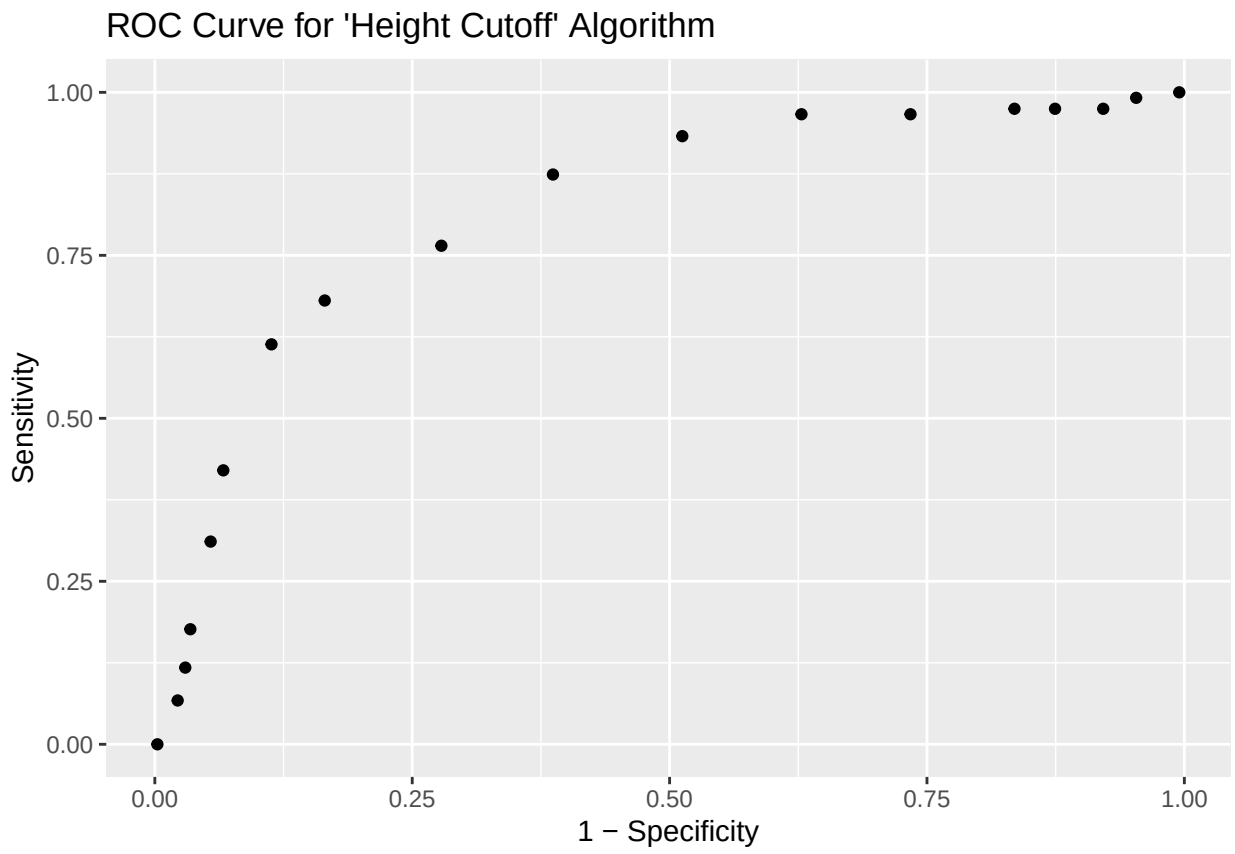
```
set.seed(2, sample.kind = "Rounding")
p_seq = seq(0, 1, length.out = 10) # Sequence of p's for testing

guessing = map_df(p_seq, function(x) {
  y_hat = sample(c("Male", "Female"), length(test_index),
                 replace = TRUE, prob = c(x, 1 - x)) %>%
    factor(levels = levels(test_set$sex))
  list(method = "Guessing",
       # True Positive Rate AKA Sensitivity
       TPR = sensitivity(y_hat, test_set$sex),
       # False Positive Rate AKA 1 - Specificity
       FPR = 1 - specificity(y_hat, test_set$sex))
})
guessing %>% ggplot(aes(FPR, TPR)) +
  geom_point() +
  xlab("1 - Specificity") +
  ylab("Sensitivity") +
  ggtitle("ROC Curve for 'Guess' Algorithm")
```



The ROC Curve for guessing is always similar to the above plot. A perfect algorithm would immediately go up to 1 (perfect Sensitivity) and stay there for all values of Specificity.

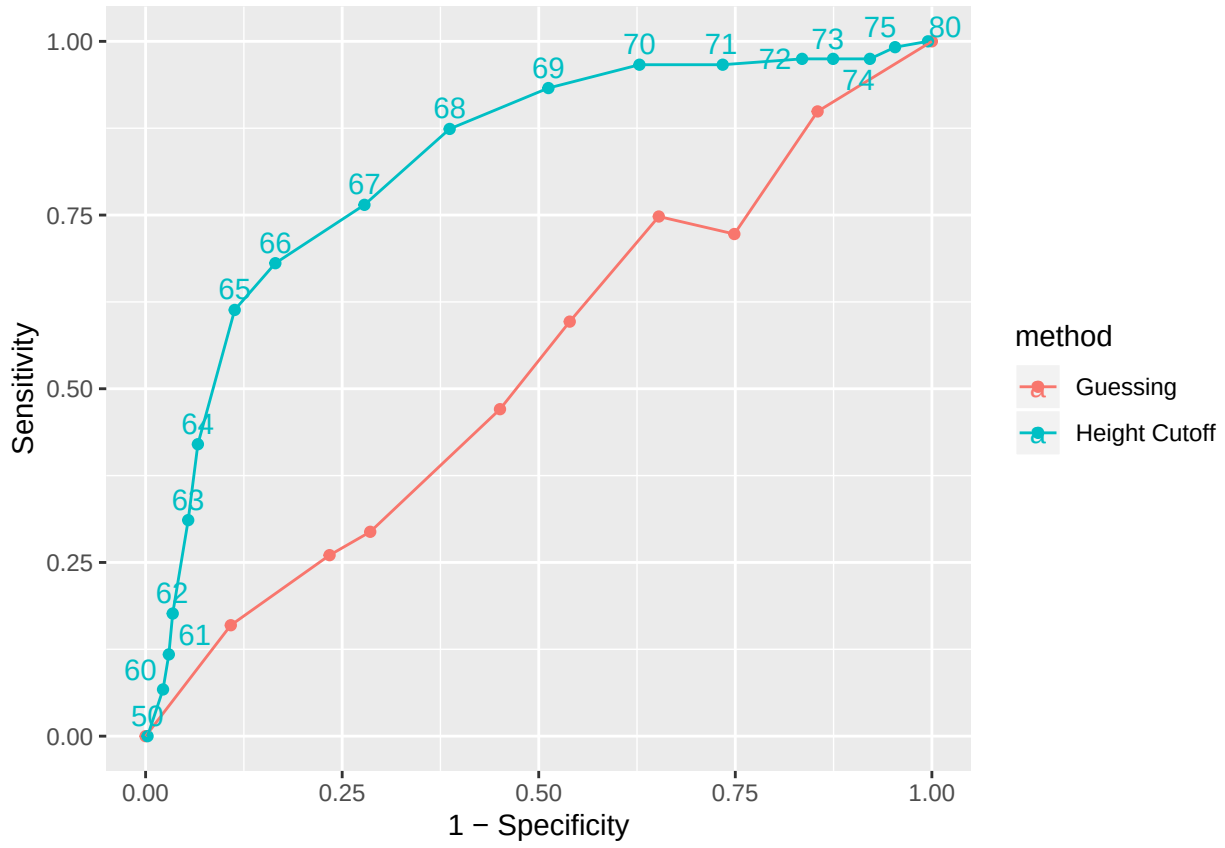
Similarly, we can construct an ROC Curve for our "Height cutoff" algorithm:

```
cutoffs = c(50, seq(60, 75), 80)
height_cutoff = map_df(cutoffs, function(x) {
  y_hat = ifelse(test_set$height > x, "Male", "Female") %>%
    factor(levels = levels(test_set$sex))
  list(method = "Height Cutoff",
       TPR = sensitivity(y_hat, test_set$sex),
       FPR = 1 - specificity(y_hat, test_set$sex),
       cutoff = x # We'll be using this later for labeling
       )
})
height_cutoff %>% ggplot(aes(FPR, TPR)) +
  geom_point() +
  xlab("1 - Specificity") +
  ylab("Sensitivity") +
  ggtitle("ROC Curve for 'Height Cutoff' Algorithm")
```



Plotting both curves together reveals a significant difference between the two algorithms:

```
bind_rows(guessing, height_cutoff) %>% # Combine results of both algorithms
  ggplot(aes(FPR, TPR, color = method, label = cutoff)) +
  geom_point() +
  geom_line() +
  xlab("1 - Specificity") +
  ylab("Sensitivity") +
  geom_text_repel(nudge_x = 0, nudge_y = 0.01)
```



By plotting both curves together, we can see that the Sensitivity of our Height Cutoff algorithm outperforms the Guessing algorithm for almost all values of Specificity. (Note: it is good practice to label the cutoff values)

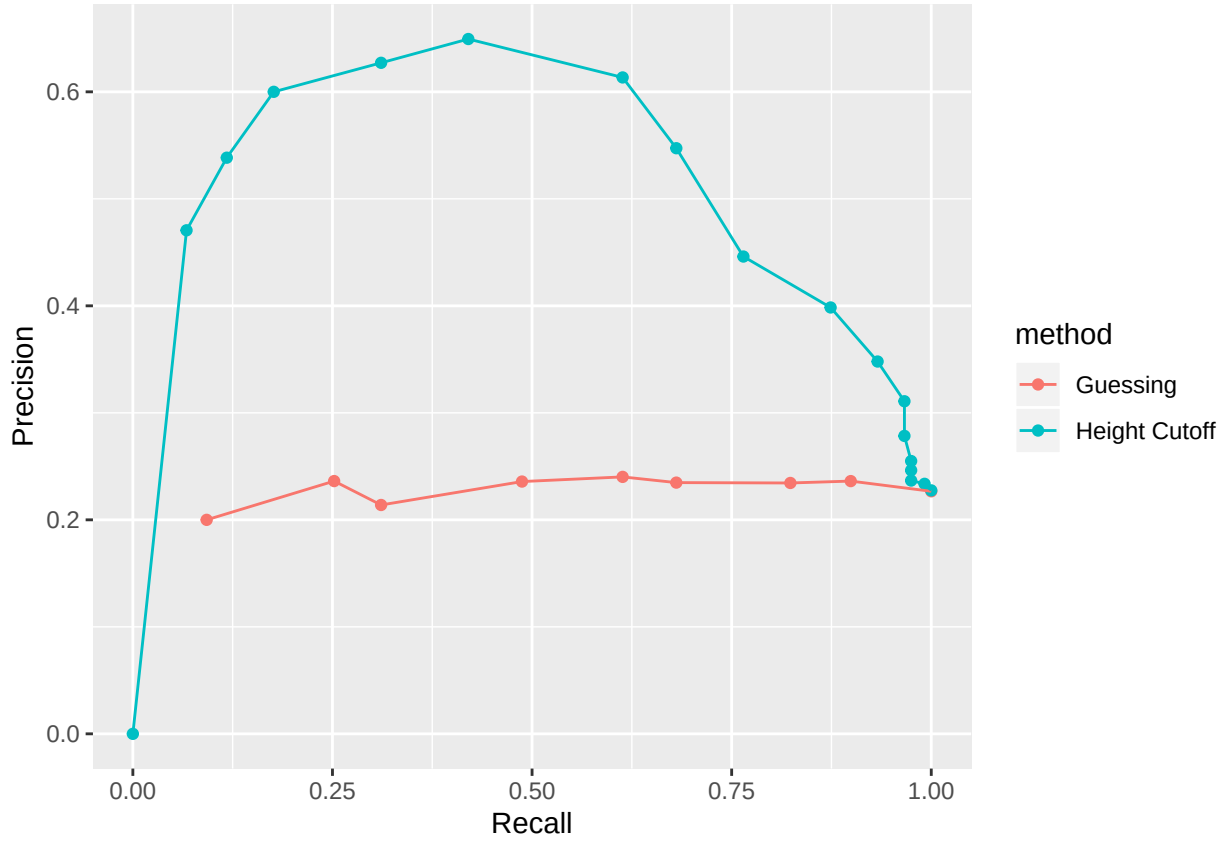
The main weakness of the ROC Curve is that it does not account for prevalence and, as such, is incapable of indicating its effect on the data.

Precision-Recall Curve

The **Precision-Recall Curve** is best used in cases in which prevalence matters.

```
# Guessing Algorithm Results
guessing = map_df(p_seq, function(x) {
  y_hat = sample(c("Male", "Female"), length(test_index), replace = TRUE, prob = c(x, 1 - x)) %>%
    factor(levels = levels(test_set$sex))
  list(method = "Guessing",
       precision = precision(y_hat, test_set$sex),
       TPR = sensitivity(y_hat, test_set$sex))
})

# Height Cutoff Algorithm Results
height_cutoff = map_df(cutoffs, function(x) {
  y_hat = ifelse(test_set$height > x, "Male", "Female") %>%
    factor(levels = levels(test_set$sex))
  list(method = "Height Cutoff",
       TPR = sensitivity(y_hat, test_set$sex),
```

```
      precision = precision(y_hat, test_set$sex))
})

# Combine both results for plotting
bind_rows(guessing, height_cutoff) %>%
  ggplot(aes(TPR, precision, color = method)) +
  geom_point() +
  geom_line() +
  xlab("Recall") + # AKA Sensitivity
  ylab("Precision")
```



From the above plot we can clearly see that precision for our "Guessing" algorithm is low due to the fact that prevalence is low.

Finally, changing what we designated to be the "Positive" value in our data set ("Male" instead of "Female") **does not** affect ROC Curves but **does** affect our Precision-Recall Curves.