# Classification: choosing the threshhold

Recall our methodology for Classification:

- Compute a "score" that our example is in each of the target classes
- Construct a probability distribution (over the target classes) from the scores
  - convert the per class score into a probability via the sigmoid/softmax function
- Compare the probability to a threshhold

$$\hat{p} = \sigma(\mathbf{\Theta}^T \mathbf{x})$$

where  $\sigma$ , the logistic function, is:

Convert  $\hat{p}^{(i)}$  into Classification prediction  $\hat{y}^{(i)}$ 

$$\hat{y}^{(i)} = \begin{cases} 0 & \text{if } \hat{p}^{(i)} < 0.5 & \text{Negative} \\ 1 & \text{if } \hat{p}^{(i)} \ge 0.5 & \text{Positive} \end{cases}$$

But does the threshhold *need* to be 0.5?

We will motivate other choices for the threshhold.

Let's examine our predictions at a fine granularity via the following table

- the row labels correspond to the predicted class
- the column labels correspond to the target (actual) class

**P** N**P** TP FP**N** FN TN

#### The correct predictions

- True Positives (TP) are test examples predicted as Positive that were in fact Positive
- True Positives (TN) are test examples predicted as Negative that were in fact Negative

#### The incorrect predictions

- False Positives (FP) are test examples predicted as Positive that were in fact Negative
- False Positives (FN) are test examples predicted as Negative that were in fact Positive

Unconditional Accuracy can thus be written as

Accuracy = 
$$\frac{TP + TN}{TP + FP + TN + FN}$$

We can also define some conditional Accuracy measures

### Recall

• Conditioned on Positive test examples

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

- The fraction of Positive examples that were correctly classified
- Also goes by the names: True Positive Rate (TPR), Sensitivity

We can affect the prediction of Positive/Negative by varying the choice of Threshhold.

We can increase the number of Positive predictions by lowering the threshhold

- this will increase TP
  - degenerate case: always predict Positive!
  - increase Recall by increasing numerator
- but also increase FP
  - which *does not* appear in denominator

Why would we want to increase Recall (at the potential cost of decreased unconditional Accuracy)?

It depends on your task.

Consider a diagnostic test for an extremely dangerous, infectious disease

- It might very important to have high Recall (catch truly infected patients)
- Even at the expense of incorrectly labelling some healthy patients as infected

## **Specificity**

• conditioned on Negative examples

Specificity = 
$$\frac{TN}{TN + FP}$$

- The fraction of Negative examples that were correctly classified
- Also goes by the name: True Negative Rate (TNR)

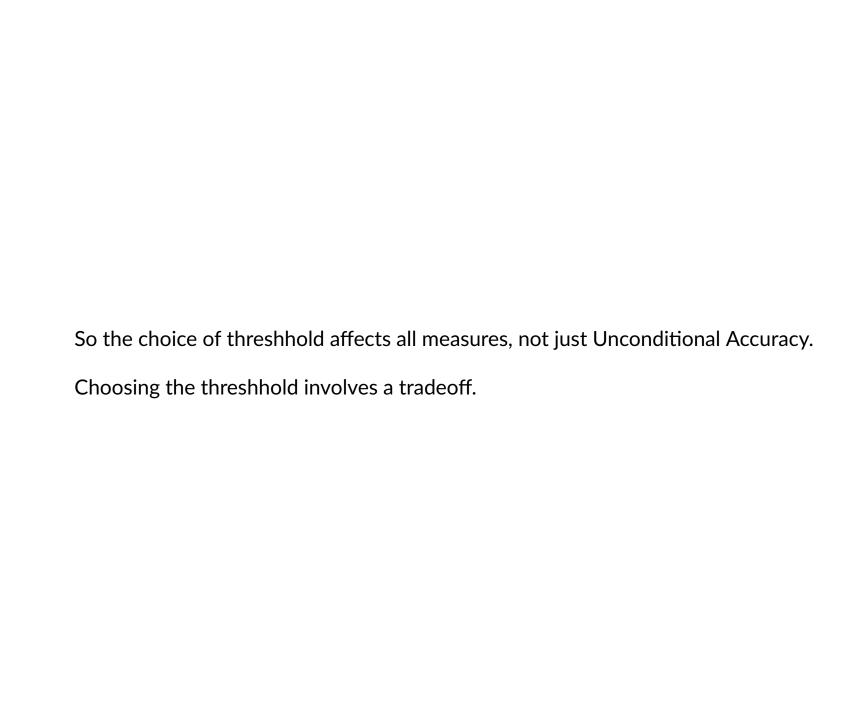
By raising the threshhold, we can increase the number of Negative prediction.

Why would we want to increase Specificity (potentially decreasing unconditional Accuracy)?

by increasing the False Negatives (FN)

Consider a diagnostic test for a mild, non-infectious disease

- A Positive prediction might entail an expensive/painful remedy, which we want to avoid
- Even at the expense of incorrectly labelling some non-healthy patients as healthy



```
In [4]: print("Done")
```

Done