Classification: True vs. False and Positive vs. Negative

In this section, we'll define the primary building blocks of the metrics we'll use to evaluate classification models. But first, a fable:

An Aesop's Fable: The Boy Who Cried Wolf (*compressed*)

A shepherd boy gets bored tending the town's flock. To have some fun, he cries out, "Wolf!" even though no wolf is in sight. The villagers run to protect the flock, but then get really mad when they realize the boy was playing a joke on them.

[Iterate previous paragraph *N* times.]

One night, the shepherd boy sees a real wolf approaching the flock and calls out, "Wolf!" The villagers refuse to be fooled again and stay in their houses. The hungry wolf turns the flock into lamb chops. The town goes hungry. Panic ensues.

Let's make the following definitions:

- "Wolf" is a **positive class**.
- "No wolf" is a **negative class**.

We can summarize our "wolf-prediction" model using a 2x2 confusion matrix that depicts all four possible outcomes:

True Positive (TP): Reality: A wolf threatened. Shepherd said: "Wolf." Outcome: Shepherd is a hero.	False Positive (FP): Reality: No wolf threatened. Shepherd said: "Wolf." Outcome: Villagers are angry at shepherd for waking them up.
False Negative (FN): Reality: A wolf threatened. Shepherd said: "No wolf." Outcome: The wolf ate all the sheep.	True Negative (TN): Reality: No wolf threatened. Shepherd said: "No wolf." Outcome: Everyone is fine.

A **true positive** is an outcome where the model *correctly* predicts the *positive* class. Similarly, a **true negative** is an outcome where the model *correctly* predicts the *negative* class.

A **false positive** is an outcome where the model *incorrectly* predicts the *positive* class. And a **false negative** is an outcome where the model *incorrectly* predicts the *negative* class.

In the following sections, we'll look at how to evaluate classification models using metrics derived from these four outcomes.

Classification: Accuracy

Accuracy is one metric for evaluating classification models. Informally, **accuracy** is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$\text{Accuracy} = \text{Number of correct predictions} / \text{Total number of predictions}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = TP + TN / (TP + TN + FP + FN)$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

Let's try calculating accuracy for the following model that classified 100 tumors as malignant (the positive class) or benign (the negative class):

True Positive (TP): Reality: Malignant ML model predicted: Malignant Number of TP results: 1	False Positive (FP): Reality: Benign ML model predicted: Malignant Number of FP results: 1
False Negative (FN): Reality: Malignant ML model predicted: Benign Number of FN results: 8	True Negative (TN): Reality: Benign ML model predicted: Benign Number of TN results: 90

$$\text{Accuracy} = TP + TN / (TP + TN + FP + FN) = 1 + 90 / (1 + 90 + 1 + 8) = 0.91$$

Accuracy comes out to 0.91, or 91% (91 correct predictions out of 100 total examples). That means our tumor classifier is doing a great job of identifying malignancies, right?

Actually, let's do a closer analysis of positives and negatives to gain more insight into our model's performance.

Of the 100 tumor examples, 91 are benign (90 TNs and 1 FP) and 9 are malignant (1 TP and 8 FNs).

Of the 91 benign tumors, the model correctly identifies 90 as benign. That's good. However, of the 9 malignant tumors, the model only correctly identifies 1 as malignant—a terrible outcome, as 8 out of 9 malignancies go undiagnosed!

While 91% accuracy may seem good at first glance, another tumor-classifier model that always predicts benign would achieve the exact same accuracy (91/100 correct predictions) on our examples. In other words, our model is no better than one that has zero predictive ability to distinguish malignant tumors from benign tumors.

Accuracy alone doesn't tell the full story when you're working with a **class-imbalanced data set**, like this one, where there is a significant disparity between the number of positive and negative labels.

In the next section, we'll look at two better metrics for evaluating class-imbalanced problems: precision and recall.

Classification: Precision and Recall

Precision

Precision attempts to answer the following question:

What proportion of positive identifications was actually correct?

Precision is defined as follows:

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP})$$

Note: A model that produces no false positives has a precision of 1.0.

Let's calculate precision for our ML model that analyzes tumors:

True Positives (TPs): 1	False Positives (FPs): 1
False Negatives (FNs): 8	True Negatives (TNs): 90

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP}) = 1/(1+1) = 0.5$$

Our model has a precision of 0.5—in other words, when it predicts a tumor is malignant, it is correct 50% of the time.

Recall

Recall attempts to answer the following question:

What proportion of actual positives was identified correctly?

Mathematically, recall is defined as follows:

$$\text{Recall}=\text{TP}/(\text{TP}+\text{FN})$$

Note: A model that produces no false negatives has a recall of 1.0.

Let's calculate recall for our tumor classifier:

True Positives (TPs): 1	False Positives (FPs): 1
False Negatives (FNs): 8	True Negatives (TNs): 90

$$\text{Recall}=\text{TP}/(\text{TP}+\text{FN})=1/(1+8)=0.11$$

Our model has a recall of 0.11—in other words, it correctly identifies 11% of all malignant tumors.

Precision and Recall: A Tug of War

To fully evaluate the effectiveness of a model, you must examine **both** precision and recall. Unfortunately, precision and recall are often in tension. That is, improving precision typically reduces recall and vice versa. Explore this notion by looking at the following figure, which shows 30 predictions made by an email classification model. Those to the right of the classification threshold are classified as "spam", while those to the left are classified as "not spam."

Figure 1. Classifying email messages as spam or not spam.

Let's calculate precision and recall based on the results shown in Figure 1:

True Positives (TP): 8	False Positives (FP): 2
False Negatives (FN): 3	True Negatives (TN): 17

Precision measures the percentage of **emails flagged as spam** that were correctly classified—that is, the percentage of dots to the right of the threshold line that are green in Figure 1:

$$\text{Precision}=TP/(TP+FP)=8/(8+2)=0.8$$

Recall measures the percentage of **actual spam emails** that were correctly classified—that is, the percentage of green dots that are to the right of the threshold line in Figure 1:

$$\text{Recall}=TP/(TP+FN)=8/(8+3)=0.73$$

Figure 2 illustrates the effect of increasing the classification threshold.

Figure 2. Increasing classification threshold.

The number of false positives decreases, but false negatives increase. As a result, precision increases, while recall decreases:

True Positives (TP): 7	False Positives (FP): 1
False Negatives (FN): 4	True Negatives (TN): 18

$$\text{Precision}= TP/(TP+FP)=7/(7+1)=0.88$$

$$\text{Recall}= TP/(TP+FN)=7/(7+4)=0.64$$

Conversely, Figure 3 illustrates the effect of decreasing the classification threshold

(from its original position in Figure 1).

Figure 3. Decreasing classification threshold.

False positives increase, and false negatives decrease. As a result, this time, precision decreases and recall increases:

True Positives (TP): 9	False Positives (FP): 3
False Negatives (FN): 2	True Negatives (TN): 16

$$\text{Precision} = TP/(TP+FP) = 9/(9+3) = 0.75$$

$$\text{Recall} = TP/(TP+FN) = 9/(9+2) = 0.82$$

Various metrics have been developed that rely on both precision and recall