Filter
Quick links
Prerequisites
Exercises
Help Center
ML models
Linear regression (70 min)
Logistic regression (35 min)
Classification (70 min)
Introduction (3 mins)
Thresholds and the confusion matrix (12 min)
Accuracy, recall, precision, and related metrics (15 min)
ROC and AUC (10 min)
Prediction bias (3 min)
Multi-class classification (2 min)
Programming exercise (15 min)
Test your knowledge (10 min)
What's next
Data
Working with numerical data (85 min)
Working with categorical data (50 min)
Datasets, generalization, and overfitting (105 min)

Thresholds and the confusion matrix

Send feedback

On this page
Confusion matrix
Effect of threshold on true and false positives and negatives
Check your understanding

Let's say you have a logistic regression model for spam-email detection that predicts a value between 0 and 1, representing the probability that a given email is spam. A prediction of 0.50 signifies a 50% likelihood that the email is spam, a prediction of 0.75 signifies a 75% likelihood that the email is spam, and so on.

You'd like to deploy this model in an email application to filter spam into a separate mail folder. But to do so, you need to convert the model's raw numerical output (e.g., `0.75`) into one of two categories: "spam" or "not spam."

To make this conversion, you choose a threshold probability, called a **classification threshold**. Examples with a probability above the threshold value are then assigned to the **positive class**, the class you are testing for (here, `spam`). Examples with a lower probability are assigned to the **negative class**, the alternative class (here, `not spam`).

Click here for more details on the classification threshold

You may be wondering: what happens if the predicted score is equal to the classification threshold (for instance, a score of 0.5 where the classification threshold is also 0.5)? Handling for this case depends on the particular implementation chosen for the classification model. The Keras library predicts the negative class if the score and threshold are equal, but other tools/frameworks may handle this case differently.

Suppose the model scores one email as 0.99, predicting that email has a 99% chance of being spam, and another email as 0.51, predicting it has a 51% chance of being spam. If you set the classification threshold to 0.5, the model will classify both emails as spam. If you set the threshold to 0.95, only the email scoring 0.99 will be classified as spam.

While 0.5 might seem like an intuitive threshold, it's not a good idea if the cost of one type of wrong classification is greater than the other, or if the classes are imbalanced. If only 0.01% of emails are spam, or if misfiling legitimate emails is worse than letting spam into the inbox, labeling anything the model considers at least 50% likely to be spam as spam produces undesirable results.

Confusion matrix

The probability score is not reality, or **ground truth**. There are four possible outcomes for each output from a binary classifier. For the spam classifier example, if you lay out the ground truth as columns and the model's prediction as rows, the following table, called a **confusion matrix**, is the result:

	Actual positive	Actual negative
Predicted positive	**True positive (TP)**: A spam email correctly classified as a spam email. These are the spam messages automatically sent to the spam folder.	**False positive (FP)**: A not-spam email misclassified as spam. These are the legitimate emails that wind up in the spam folder.
Predicted negative	**False negative (FN)**: A spam email misclassified as not-spam. These are spam emails that aren't caught by the spam filter and make their way into the inbox.	**True negative (TN)**: A not-spam email correctly classified as not-spam. These are the legitimate emails that are sent directly to the inbox.

Notice that the total in each row gives all predicted positives (TP + FP) and all predicted negatives (FN + TN), regardless of validity. The total in each column, meanwhile, gives all real positives (TP + FN) and all real negatives (FP + TN) regardless of model classification.

When the total of actual positives is not close to the total of actual negatives, the dataset is **imbalanced**. An instance of an imbalanced dataset might be a set of thousands of photos of clouds, where the rare cloud type you are interested in, say, volutus clouds, only appears a few times.

Effect of threshold on true and false positives and negatives

Different thresholds usually result in different numbers of true and false positives and true and false negatives. The following video explains why this is the case.

Try changing the threshold yourself.

This widget includes three toy datasets:

- **Separated**, where positive examples and negative examples are generally well differentiated, with most positive examples having higher scores than negative examples.
- **Unseparated**, where many positive examples have lower scores than negative examples, and many negative examples have higher scores than positive examples.
- **Imbalanced**, containing only a few examples of the positive class.

Check your understanding

1. Imagine a phishing or malware classification model where phishing and malware websites are in the class labeled 1 (true) and harmless websites are in the class labeled 0 (false). This model mistakenly classifies a legitimate website as malware. What is this called?

- A true positive ☐
- A true negative ☐
- A false positive ✓

  A negative example (legitimate site) has been wrongly classified as a positive example (malware site).

  Correct answer.
- A false negative ☐

2. In general, what happens to the number of false positives when the classification threshold increases? What about true positives? Experiment with the slider above.

- Both true and false positives increase. ☐
- True positives increase. False positives decrease. ☐
- Both true and false positives decrease. ✓

  As the threshold increases, the model will likely predict fewer positives overall, both true and false. A spam classifier with a threshold of .9999 will only label an email as spam if it considers the classification to be at least 99.99% likely, which means it is highly unlikely to mislabel a legitimate email, but also likely to miss actual spam email.

  Correct answer.

3. In general, what happens to the number of false negatives when the classification threshold increases? What about true negatives? Experiment with the slider above.

- Both true and false negatives decrease. ☐
- True negatives increase. False negatives decrease. ☐
- Both true and false negatives increase. ✓

  As the threshold increases, the model will likely predict more negatives overall, both true and false. At a very high threshold, almost all emails, both spam and not-spam, will be classified as not-spam.

  Correct answer.

Key terms:

- Binary classification
- Class-imbalanced dataset
- Classification threshold
- Confusion matrix
- Ground truth
- Negative class
- Positive class

Help Center

Previous
Introduction (3 mins)

Next
Accuracy, recall, precision, and related metrics (15 min)

Was this helpful?

Send feedback

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 License, and code samples are licensed under the Apache 2.0 License. For details, see the Google Developers Site Policies. Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-05-22 UTC.