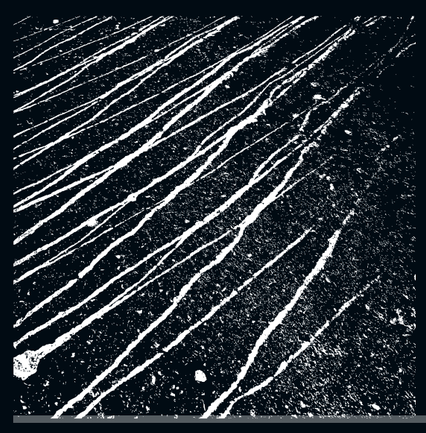
The Confusion Matrix Explained

How you can train a model that works for you.

The AI “black box”

Often AI is referred to as a “black box” because it's difficult to understand how the algorithms work. Measuring the accuracy of an model is an important step to gaining confidence in AI.



Getting clear on the maths

Machine learning models use different mathematical algorithms optimised for certain use cases. Depending on the use case you may want to use a specific algorithm, or train a model for it to perform a specific way.

Model parameters

You can modify the behaviour of AI models by changing model settings called "hyperparameters."

Classification

Machine learning models can be used to "label" data, or classify data into categories. For example, a model can predict if a manufactured part is faulty or not based on data rather than physically testing it.

Configuring hyperparameters to tune the model is key to making more accurate predictions.

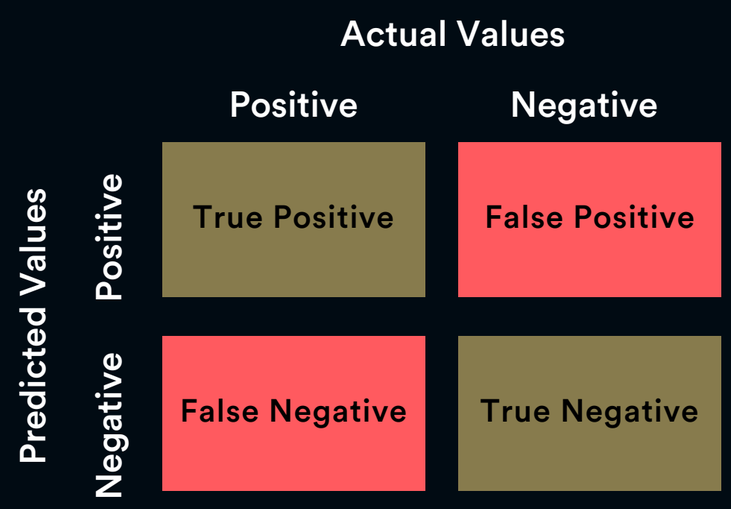
Prediction examples

For example, a model may predict that out of 100 parts, 98 parts were functional, and 2 parts were faulty. When a prediction is incorrect it might mean that there are:

False Positives - Parts that were predicted to be faulty but were actually functional.

False Negatives - Parts that were predicted to be functional but were actually faulty.

The confusion matrix



The confusion matrix is a common tool for measuring the accuracy of your classification model by comparing predicted vs actual results in a table.

See how you can use the confusion matrix to build a classification model that works for your application.

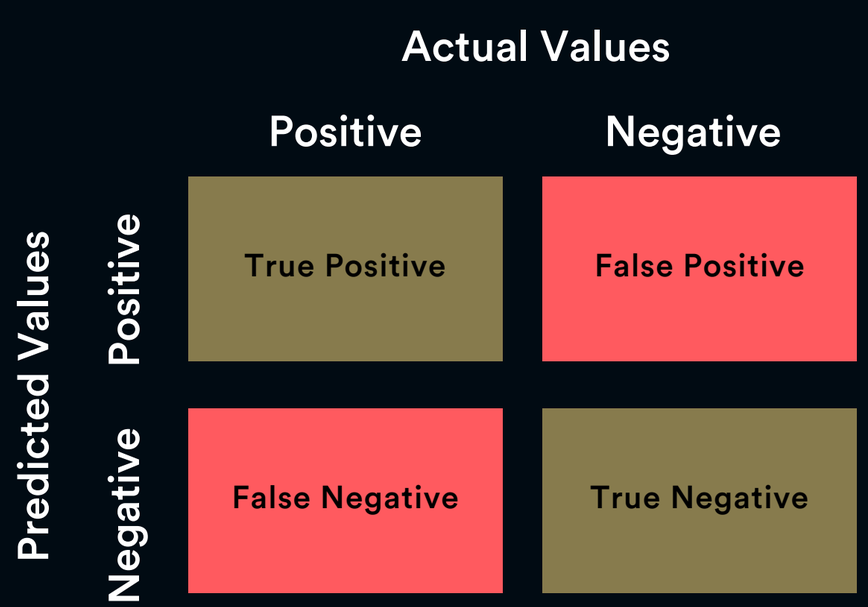
MONOLITH

The Confusion Matrix Explained

What the confusion matrix tells you:

By tuning various hyperparameters or changing the type of machine learning model — you can adjust and improve different prediction results. The confusion matrix is a simplified view to help you understand which model is best suited for your application.

The confusion matrix



Positive or negative?

How you define what is positive and negative can affect your understanding of the results from the confusion matrix.

If you are using a machine learning model to detect faulty parts, and it finds one — that is considered a “positive” result.

To properly interpret the numbers in a confusion matrix, you must clearly understand the "positive" and "negative" definitions used.

What do the results mean for me?

- True Positive - Correct prediction
- True Negative - Correct prediction
- False Positive - “False Alarm”
  - A part that was marked as faulty but was actually functional — in which case you are discarding a perfectly functional part unnecessarily because of an incorrect prediction.
- False Negative - “Miss”
  - A part marked as functional but was actually faulty — meaning a defective part is allowed to proceed, which could result in safety concerns or downtime due to product failures.

See a worked example of how an engineer would use a confusion matrix to select the best model for their application

MONOLITH

The Confusion Matrix Explained

Worked Example

Sophia, a test engineer, wants to analyse production data from manufacturing lightbulbs to predict whether they are defective or not. Using this approach, she can reduce the number of physical tests required and increase production throughput.

To do this, Sophia trains two different models (Alpha and Beta) using different hyperparameter settings to find the most accurate predictor.

To validate model predictions, Sophia compares model results against actual test results.

After physically testing the lightbulbs to validate the model, there were different results*:

True Positive - Some lightbulbs were predicted to be faulty and tests proved that they were indeed faulty
True Negative - Some lightbulbs were predicted to be functional and they were indeed functional
False Positive - Some lightbulbs were predicted to be faulty but when tested proved to be functional
False Negative - Some lightbulbs were predicted to be functional but when tested proved to be faulty

*Note: In this example “positive” refers to the presence of a faulty part

Model Alpha

Predicted Values \ Actual Values	Positive	Negative
Positive	True Positive (50)	False Positive (5)
Negative	False Negative (10)	True Negative (930)

Model Alpha here shows that it was correct for 980 (930+50) samples.

It falsely marked 5 lightbulbs as defective when they were in reality functional.

It falsely marked 10 lightbulbs as functional when they were in fact defective.

This model prioritises reducing waste (fewer false positives)

Model Beta

Predicted Values \ Actual Values	Positive	Negative
Positive	True Positive (45)	False Positive (25)
Negative	False Negative (5)	True Negative (925)

Model Beta here shows that it was correct for 970 (925+45) samples.

It falsely marked 25 lightbulbs as defective when they were in reality functional.

It falsely marked 5 lightbulbs as functional when they were in fact defective.

This model prioritises reducing potential safety risks (fewer false negatives)

Continued...

The Confusion Matrix Explained

Worked example [continued]

After reviewing the results through a confusion matrix, Sophia now has two different models she can use for evaluating parts — Alpha, which is tuned to reduce waste, and Beta, which is tuned to minimise safety risk.

For an application like manufacturing light bulbs, you might prioritise reducing waste, or minimising false positives. If a faulty light bulb slips through, you can easily replace it with a new one. However, if the algorithm is incorrectly marking bulbs as faulty when they are not, you are needlessly wasting your inventory.

For a low-cost, low-risk product like light bulbs, applying the Alpha model is more appropriate.

Therefore, model Alpha overall is better for Sophia's lightbulb use case because it reduces waste and does not introduce significant risks if a faulty bulb slips through.

Model Alpha

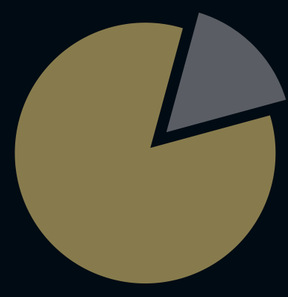
Predicted Values	Actual Values: Positive	Actual Values: Negative
Positive	True Positive (50)	False Positive (5)
Negative	False Negative (10)	True Negative (930)

Finding the balance

Different modelling algorithms and settings will classify data differently. Each are valuable depending on your priorities relative to minimising false positives or false negatives.

Other metrics

There are other metrics that you can use to measure how suitable your model is for your application.



Testing less with AI

Using machine learning models, engineers can reduce the physical testing requirements to save time, resources, and money.

Glossary/Terms/Metrics →

Confusion Matrix Glossary

Confusion Matrix: A table used to describe the performance of a classification model. It shows the actual versus predicted classifications in a 2x2 matrix for binary classification problems, but can be expanded for multi-class problems.

True Positive (TP): The number of instances correctly predicted as positive.

True Negative (TN): The number of instances correctly predicted as negative.

False Positive (FP): The number of instances incorrectly predicted as positive (also known as Type I error).

False Negative (FN): The number of instances incorrectly predicted as negative (also known as Type II error).

Accuracy: The ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances. Formula: (TP+TN)/(TP+TN+FP+FN).

Precision: The ratio of correctly predicted positive instances to the total predicted positives. Formula: TP/(TP+FP).

Recall: Also known as Sensitivity or True Positive Rate, it is the ratio of correctly predicted positive instances to all actual positives. Formula: TP/(TP+FN).

Specificity: The ratio of correctly predicted negative instances to all actual negatives. Formula: TN/(TN+FP).

F1 Score: The harmonic mean of precision and recall, providing a balance between the two. Formula: 2x(Precision×Recall)/(Precision+Recall).

Sensitivity: Another term for Recall, it measures the proportion of actual positives that are correctly identified.

Negative Predictive Value (NPV): The ratio of correctly predicted negative instances to the total predicted negatives. Formula: TN/(TN+FN).

Positive Predictive Value (PPV): Another term for Precision, it measures the proportion of positive results that are true positives.

Type I Error: The error of falsely identifying a negative instance as positive (False Positive).

Type II Error: The error of falsely identifying a positive instance as negative (False Negative).

MONOLITH

We enable engineers all over the world to:

- Reduce validation testing time and effort
- Get to market faster
- Optimise complex battery test plans
- Find more errors in your test data faster
- Forecast results to stop long-running tests early