Understanding the Confusion Matrix in Machine Learning

We can come up with multiple models for a particular problem, hence to select the best out of it we have many performance measures to judge upon. One of which is "CONFUSION MATRIX" which is used for checking the performance of a model based on any classification algorithm.

A confusion matrix is a table that summarizes the performance of a classification model by showing the counts of true positive, true negative, false positive, and false negative predictions. It helps in understanding the types of errors a model makes, beyond just overall accuracy, and is crucial for evaluating model performance, especially in imbalanced datasets or when different types of errors have different consequences

What is Confusion Matrix?

Confusion matrix is a simple table used to measure how well a classification model is performing. It compares the predictions made by the model with the actual results and shows where the model was right or wrong. This helps you understand where the model is making mistakes so you can improve it. It breaks down the predictions into four categories:

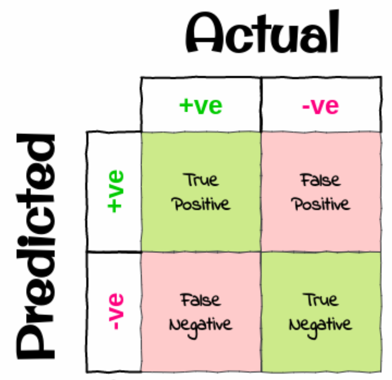
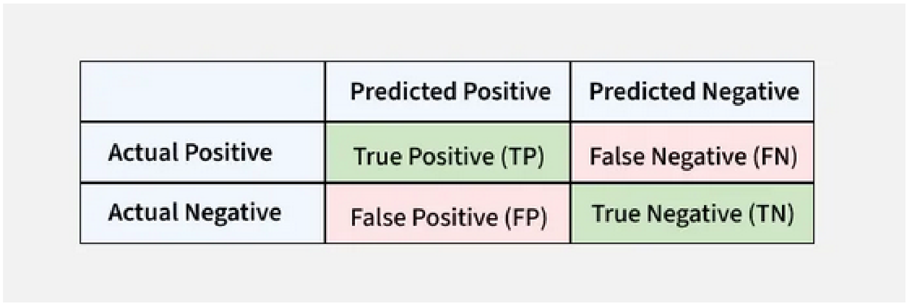


Fig 1: general representation of a confusion matrix



	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

- **True Positive (TP):** The model correctly predicted a positive outcome i.e the actual outcome was positive.
- **True Negative (TN):** The model correctly predicted a negative outcome i.e the actual outcome was negative.
- **False Positive (FP):** The model incorrectly predicted a positive outcome i.e the actual outcome was negative. It is also known as a Type I error.
- **False Negative (FN):** The model incorrectly predicted a negative outcome i.e the actual outcome was positive. It is also known as a Type II error.

It also helps calculate key measures like **accuracy**, **precision** and **recall** which give a better idea of performance especially when the data is imbalanced.

Metrics based on Confusion Matrix Data

1. Accuracy

Accuracy shows how many predictions the model got right out of all the predictions. It gives idea of overall performance but it can be misleading when one class is more dominant over the other. For example, a model that predicts the majority class correctly most of the time might have high accuracy but still fail to capture important details about other classes. It can be calculated using the below formula:

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

2. Precision

Precision focus on the quality of the model's positive predictions. It tells us how many of the "positive" predictions were actually correct. It is important in situations where false positives need to be minimized such as detecting spam emails or fraud. The formula of precision is:

$$\boxed{\text{Precision} = \frac{TP}{TP+FP}}$$

3. Recall

Recall measures how good the model is at predicting positives. It shows the proportion of true positives detected out of all the actual positive instances. High recall is essential when missing positive cases has significant consequences like in medical tests.

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

4. F1-Score

F1-score combines precision and recall into a single metric to balance their trade-off. It provides a better sense of a model's overall performance particularly for imbalanced datasets. It is helpful when both false positives and false negatives are important though it assumes precision and recall are equally important but in some situations one might matter more than the other.

$$\text{F1-Score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

5. Specificity

Specificity is another important metric in the evaluation of classification models particularly in binary classification. It measures the ability of a model to correctly identify negative instances. Specificity is also known as the True Negative Rate Formula is given by:

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

6. Type 1 and Type 2 error

Type 1 and Type 2 error are:

- **Type 1 error**: It occurs when the model incorrectly predicts a positive instance but the actual instance is negative. This is also known as a **false positive**. Type 1 Errors affect the **precision** of a model which measures the accuracy of positive predictions.

$$\text{Type 1 Error} = \frac{\text{FP}}{\text{FP}+\text{TN}}$$

- **Type 2 error**: This occurs when the model fails to predict a positive instance even though it is actually positive. This is also known as a **false negative**. Type 2 Errors impact the **recall** of a model which measures how well the model identifies all actual positive cases.

$$\boxed{\text{Type 2 Error} = \frac{FN}{TP+FN}}$$

Example: A diagnostic test is used to detect a particular disease in patients.

- **Type 1 Error (False Positive):** This occurs when the test predicts a patient has the disease (positive result) but the patient is actually healthy (negative case).

- **Type 2 Error (False Negative):** This occurs when the test predicts the patient is healthy (negative result) but the patient actually has the disease (positive case).

Confusion Matrix For Binary Classification

A 2x2 Confusion matrix is shown below for the image recognition having a Dog image or Not Dog image:

	Predicted	**Predicted**
Actual	True Positive (TP)	False Negative (FN)
Actual	False Positive (FP)	True Negative (TN)

- **True Positive (TP):** It is the total counts having both predicted and actual values are Dog.

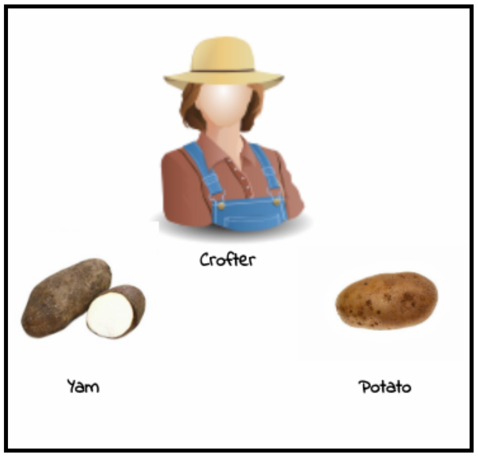
- **True Negative (TN):** It is the total counts having both predicted and actual values are Not Dog.

- **False Positive (FP):** It is the total counts having prediction is Dog while actually Not Dog.

- **False Negative (FN):** It is the total counts having prediction is Not Dog while actually, it is Dog.

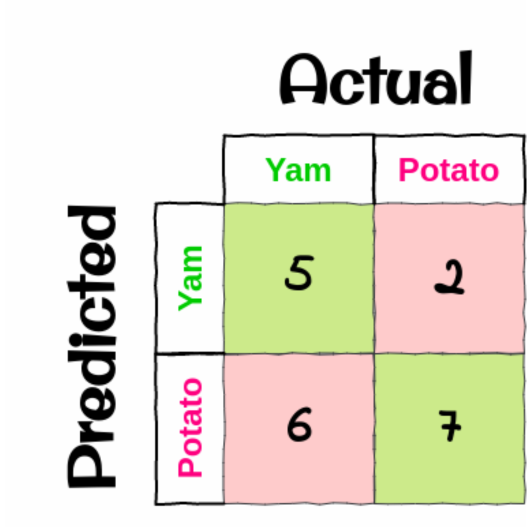
EXAMPLE

Let's understand how a confusion matrix works and how it looks with the help of an example. Let's imagine we are to build a classification model for a crofter who grows two types of crops viz. 1. Yam 2. Potato



So the task of the classifier that we have built is to classify both the crops correctly so the crofter can maximize his profit by selling the crops at their right market price.

So the crofter sends 20 samples to us for checking how the classifier performs, out of these 20 samples 11 samples are of Yam and 9 samples are of Potatoes.



		Actual	
		Yam	Potato
Predicted	Yam	5	2
	Potato	6	7

Fig 2: Classifier's performance

Fig 2 gives us the confusion matrix for our classifier that we built for the crofter, which basically tells us that the columns are our actual classes(crops here) and the rows are our predicted classes(crops here) predicted by the classifier, it also tells us that:

Out of 11 Yam samples, 5 were correctly classified to be Yam and 6 were incorrectly classified to be Potato. Similarly out of 9 Potato samples, 2 were incorrectly classified to be Yam and 7 were correctly classified to be Potato.

That means out of 20 samples 12 were correctly classified(sum of green boxes in fig 2) and 8 were incorrectly classified(sum of red boxes in fig 2). Also from fig 2 we get to know that the classifier is performing really bad for rightly classifying Yam but is doing a faily good job for rightly classifying Potatoes.

So, if we are to define a confusion matrix in simple words then we can say that:

"Confusion Matrix helps us to know the number of correct and incorrect classifications for each class."

Recall

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

So recall is the ratio of TP/(TP+FN) where TP is the number of True Positives and FN is the number of False Negatives.

In other words, it lets us know the number of correct classifications of a particular class over the total number of samples of that class.

Let's understand it with the crofter's example taking "Yam" as the crop that we are interested in:

so if we calculate the recall be placing the values from fig2 to the formula we got:

recall = 5/(5+6) = 5/11 = 0.454 = 45.4%,

where TP = 5 and FN = 6

As we saw that the recall percentage was quite low that shows that our classifier is not classifying Yam correctly most of the time and often mistaking it to be a potato.

Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

So precision is the ratio of TP/(TP+FP) where TP is the number of True Positives and FP is the number of False Positives.

In other words, it lets us know the number of correct classifications of a particular class over the total number of predictions for that class.

Let's understand it by again taking the crofter's example, we still consider "Yam" as the crop that we are interested in:

so if we calculate the precision be placing the values from fig2 to the formula we got:

precision = 5/(5+2) = 5/7 = 0.714 = 71.4%,

where TP = 5 and FP = 2

So from the above calculation, we got the insight that the classifier 5 times correctly predicted the Yam sample to be the "Yam" but 2 times wrongly predicted a Potato sample to be a "Yam". In other words, the significantly lower values of FP in comparison to TP, the higher the precision of the classifier will be.

F1- score

$$\textbf{F1-Score} = \frac{2 \text{ x (precision x recall)}}{\text{precision + recall}}$$

$$\textbf{F1-Score} = \frac{\text{TP}}{\text{TP} + \left[\frac{\text{FN + FP}}{2}\right]}$$

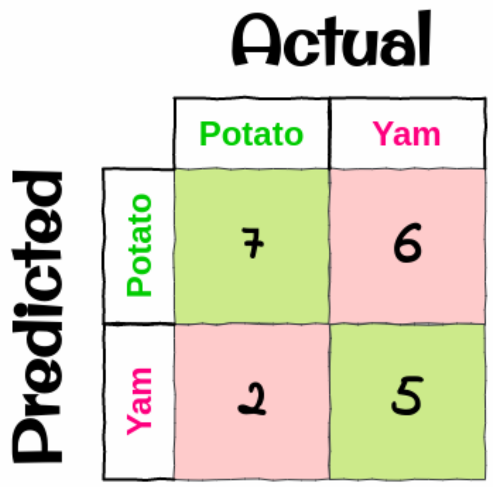
The F1-score is a measure of a model's overall performance on a particular dataset. The F1-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall.

So if we calculate F1-score from the above-calculated precision and recall we have it as below:

F1-score = (2 x (0.454 x 0.714)) / (0.454 + 0.714) = 0.648/1.168 = 0.554

This tells us that the performance of the classifier that we did build for the crofter is not performing that very great in terms of correctly classifying the Yam crop.

We can similarly find the recall, precision, and f1-score for potatoes too. ***By considering them as our positive class or the class of interest.*** For that, we would just have to swap the columns of the actual classes and will also have to swap the rows of the predicted classes in fig2. Then the confusion matrix will look as in fig3.



		Actual	
		Potato	Yam
Predicted	Potato	7	6
	Yam	2	5

Fig3.

Recall = 77.7%

Precision = 53.8%

F1-score = 0.635