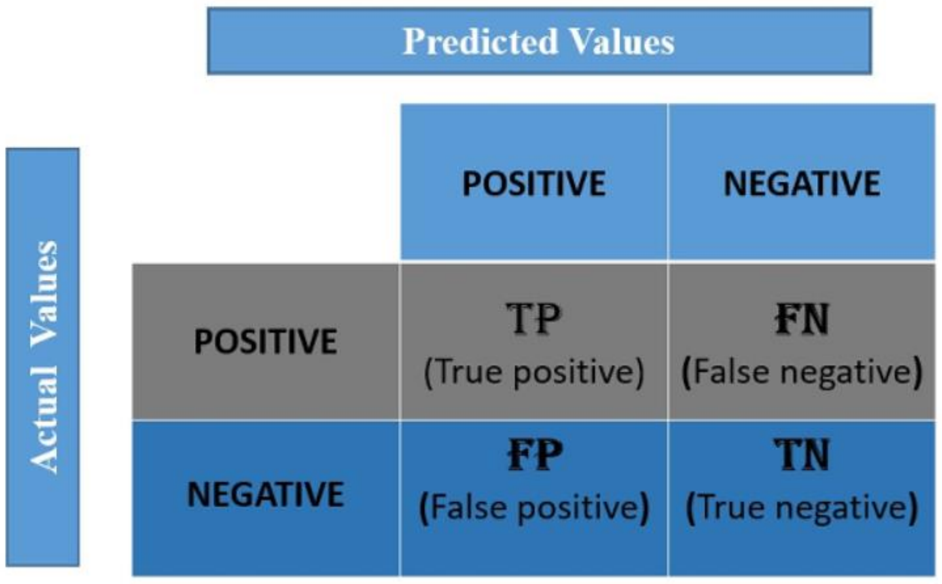
What is a Confusion Matrix?

Confusion Matrix is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



The confusion matrix depicts four possible outcomes:

1. True Positive (TP) - A **true positive** is an outcome where the model *correctly* predicts the *positive* class.
2. False Positive (FP) - A **false positive** is an outcome where the model *incorrectly* predicts the *positive* class.
3. False Negative (FN) - A **false negative** is an outcome where the model *incorrectly* predicts the *negative* class.
4. True Negative (TN) - A **true negative** is an outcome where the model *correctly* predicts the *negative* class.

Confusion Matrix for Binary Classification

Let us see how to construct a confusion matrix and understand its terminologies. Consider, we have to model a classifier that classifies 2 kinds of fruits. We have 2 types of fruits – apples and grapes – and we want our machine-learning model to identify or classify the given fruit as an apple or grape.

So we take 15 samples of 2 fruits, out of which 8 belong to Apples, and 7 belong to the Grapes class. Class is nothing but the output, in this example, we have 2 output classes – Apples and Grapes. We will represent Apple as 1 and Grape as 0 class.

The actual class for 8 apples and 7 grapes can be represented as:

Actual = [1,1,1,1,1,1,1,1,0,0,0,0,0,0,0]

The classifier model predicts 1 for Apple and 0 for grape.

Assume that the classifier takes all 15 inputs and makes the following predictions:

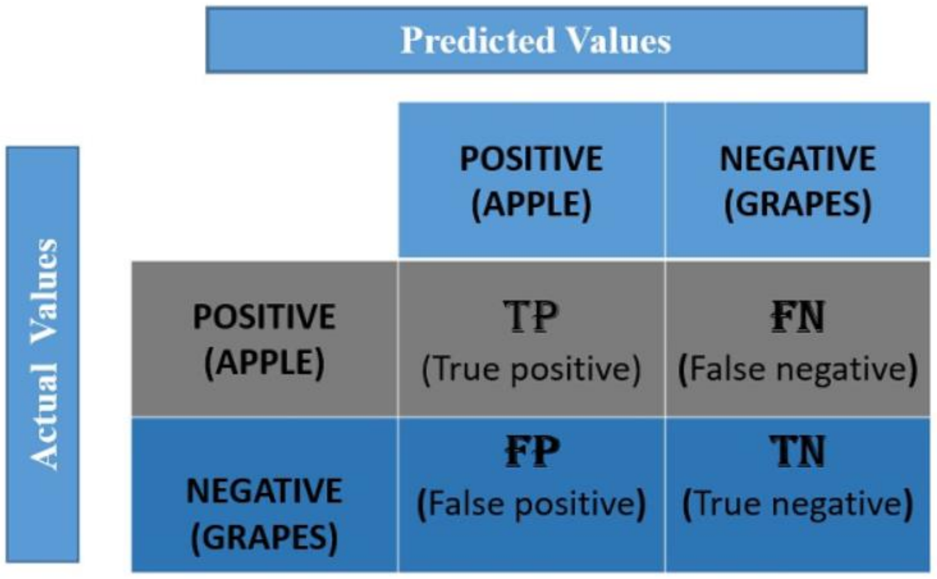
· Out of 8 apples, it will classify 5 correctly as apples and wrongly predict 3 as grapes.

· Out of 7 grapes, it will classify 5 correctly as grapes and wrongly predicts 2 as apples.

The prediction of the classifier may be as follows:

Prediction = [1,0,0,0,1,1,1,1,0,0,0,0,0,1,1]

The confusion matrix for this example can be visualized below.



For our example, the positive value is Apple, and the negative value is Grapes.

True Positive:

It means the actual value and also the predicted values are the same. In our case, the actual value is also an apple, and the model prediction is also an apple. If you observe the TP cell, the positive value is the same for Actual and predicted.

False Negative:

This means the actual value is positive. In our case, it is apple, but the model has predicted it as negative, i.e., grapes. So the model has given the wrong prediction. It was supposed to give a positive (apple), but it has given a negative (grape). So whatever the negative output we got is false; hence the name False Negative.

False Positive:

This means the actual value is negative. In our case, it is grapes, but the model has predicted it as positive, i.e., apple. So the model has given the wrong prediction. It was supposed to give a negative (grape), but it has given a positive (apple), so whatever the positive output we got is false, hence the name False Positive.

True Negative:

It means the actual value and also the predicted values are the same. In our case, the actual values are grapes, and the prediction is also Grapes.

The values for the above example are:

TP=5, FN=3, FP=2, TN=5.

Confusion Matrix for Multi-Class Classification

In the multi-class classification problem, we won't get TP, TN, FP, and FN values directly as in the binary classification problem.

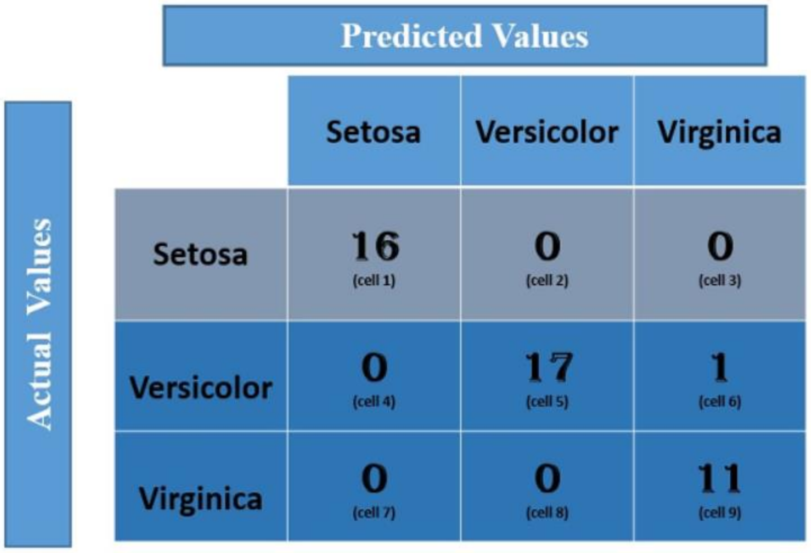
Let's try to understand the confusion matrix for 3 classes and the confusion matrix for multiple classes with a popular dataset – the IRIS DATASET.

The dataset has 3 flowers as outputs or classes, Versicolor, Virginia, and Setosa.

How to Calculate FN, FP, TN, and TP Values?

As discussed earlier, FN: The False-negative value for a class will be the sum of values of corresponding rows except for the TP value. FP: The False-positive value for a class will be the sum of values of the corresponding column except for the TP value. TN: The True-negative value for a class will be the sum of the values of all columns and rows except the values of that class that we are calculating the values for. And TP: the True-positive value is where the actual value and predicted value are the same.

The confusion matrix for the IRIS dataset is as below:



Actual Values \ Predicted Values	Setosa	Versicolor	Virginica
Setosa	16 (cell 1)	0 (cell 2)	0 (cell 3)
Versicolor	0 (cell 4)	17 (cell 5)	1 (cell 6)
Virginica	0 (cell 7)	0 (cell 8)	11 (cell 9)

Let us calculate the TP, TN, FP, and FN values for the class **Setosa** using the above tricks:

TP: The actual value and predicted value should be the same. So concerning Setosa class, the value of cell 1 is the TP value.

FN: The sum of values of corresponding rows except for the TP value

FN = (cell 2 + cell3)

= (0 + 0)

= 0

FP: The sum of values of the corresponding column except for the TP value.

FP = (cell 4 + cell 7)

= (0 + 0)

= 0

TN: The sum of values of all columns and rows except the values of that class that we are calculating the values for.

TN = (cell 5 + cell 6 + cell 8 + cell 9)

= 17 + 1 +0 + 11

= 29

Similarly, for the **Versicolor** class, the values/metrics are calculated as below:

TP: 17 (cell 5)

FN: 0 + 1 = 1 (cell 4 +cell 6)

FP: 0 + 0 = 0 (cell 2 + cell 8)

TN: 16 +0 +0 + 11 =27 (cell 1 + cell 3 + cell 7 + cell 9).

Why Use Confusion Matrix?

The confusion Matrix allows us to measure F1-Score, Recall and Precision, which, along with Accuracy and the AUC-ROC curve, are the metrics used to measure the performance of ML models.

Accuracy

Accuracy represents the number of correctly classified data instances over the total number of data instances.

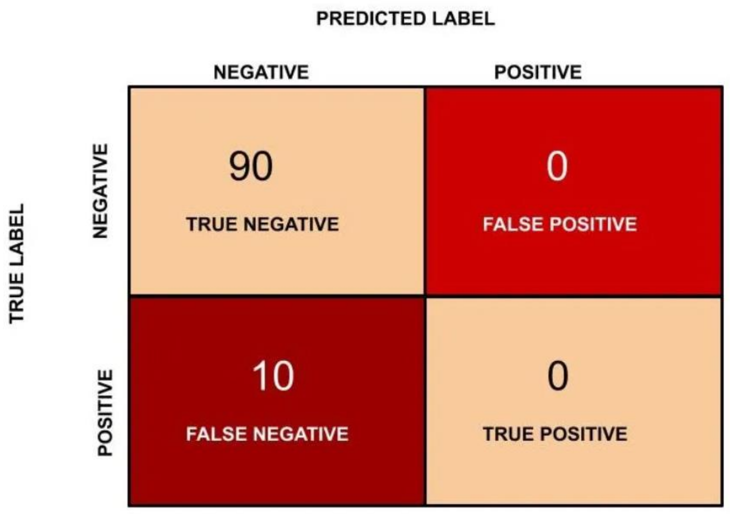
$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

Is accuracy the best measure?

Accuracy may not be a good measure if the dataset is not balanced (imbalanced). We will explain this with an example.

Consider the following scenario: There are 90 people who are healthy (negative) and 10 people who have some disease (positive). Now let's say our machine learning model perfectly classified the 90 people as healthy but it also classified the unhealthy people as healthy. What will happen in this scenario? Let us see the confusion matrix and find out the accuracy?

In this example, *TN* = 90, *FP* = 0, *FN* = 10 and *TP* = 0. The confusion matrix is as follows.


PREDICTED LABEL
NEGATIVE
POSITIVE
TRUE LABEL
NEGATIVE
POSITIVE
90
TRUE NEGATIVE
0
FALSE POSITIVE
10
FALSE NEGATIVE
0
TRUE POSITIVE

Accuracy in this case will be (90 + 0)/100 = 0.9 and in percentage the accuracy is 90 %.

Is there anything fishy?

The accuracy, in this case, is 90 % but this model is very poor because all the 10 people who are unhealthy are classified as healthy. By this example what we are trying to say is that accuracy is not a good metric when the data set is imbalanced. Using accuracy in such scenarios can result in misleading interpretation of results.

Let us move ahead with other metrics – Precision, Recall and F1-Score.

Precision

Precision measures the accuracy of positive predictions.

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

Precision should ideally be 1 (high) for a good classifier. *Precision* becomes 1 only when the numerator and denominator are equal i.e. *TP = TP +FP*, this also means *FP* is zero. As *FP* increases the value of denominator becomes greater than the numerator and *precision* value decreases (which we don't want).

So for example, *precision* = 90 / (90+ 0) = 1

Recall

Recall measures the completeness of positive predictions.

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

Recall should ideally be 1 (high) for a good classifier. *Recall* becomes 1 only when the numerator and denominator are equal i.e. *TP = TP +FN*, this also means *FN* is zero. As *FN* increases the value of denominator becomes greater than the numerator and *recall* value decreases (which we don't want).

So for example, *Recall* = 90 / (90+ 10) = 0.90

F1-Score

The F1-Score combines the precision and recall scores of a model.

So ideally in a good classifier, we want both *precision* and *recall* to be one which also means *FP* and *FN* are zero. Therefore we need a metric that takes into account both *precision* and *recall*. *F1-score* is a metric which takes into account both *precision* and *recall* and is defined as follows:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

F1 Score becomes 1 only when *precision* and *recall* are both 1. *F1 score* becomes high only when both *precision* and *recall* are high. *F1 score* is the harmonic mean of *precision* and *recall* and is a better measure than *accuracy*.

In our example, *F1 Score* = 2 * (1.0 * 0.9) / (1.0 + 0.9) = 0.94.