

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)



From Excel to ML, NLP, DL. Learn everything about Data Science!

[Learn More](#)

[Home](#)

[Aniruddha Bhandari](#) – Published On April 17, 2020 and Last Modified On May 2nd, 2023

[Beginner](#) [Classification](#) [Machine Learning](#) [Python](#) [Technique](#)

⌚ 12 minutes



Introduction

Confusion Matrix – Not So Confusing!

Have you been in a situation where you expected your [machine learning model](#) to perform really well, but it sputtered out a poor accuracy? You've done all the hard work – so where did the classification model go wrong? How can you correct this?

There are plenty of ways to gauge the performance of your classification model, but none have stood the test of time like the confusion matrix. It helps us evaluate how our model performed and where it went wrong and offers us guidance to correct our path. Most people, especially beginners who have just started to understand machine learning or artificial intelligence, find the concept of a confusion matrix very “confusing”. However, this is not the case.

In this tutorial, we will explore what a confusion matrix in machine learning is and how it gives a holistic view of the performance of your model. And unlike its name, you will realize that a confusion matrix is a pretty simple yet powerful concept in machine learning or deep learning. So let's unravel the mystery around the confusion matrix!

Learning Objectives

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics Vidhya, you agree to our [Privacy Policy](#) and [Terms of Use](#). [Accept](#)

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)

- [Python for Data Science \(free course!\)](#)
- [Machine Learning Starter Program](#)

Table of Contents

1. [What Is a Confusion Matrix?](#)
2. [Important Terms in a Confusion Matrix](#)
3. [Why Do We Need a Confusion Matrix?](#)
4. [Precision vs. Recall](#)
5. [Confusion Matrix Using Scikit-learn in Python](#)
6. [Confusion Matrix for Multi-Class Classification](#)
7. [Conclusion](#)
8. [Frequently Asked Questions](#)
 1. [Q1. How is accuracy calculated from a confusion matrix?](#)
 2. [Q2. How is precision calculated from a confusion matrix?](#)
 3. [Q3. What is the F1 score in a confusion matrix?](#)

What Is a Confusion Matrix?

A Confusion matrix is an $N \times N$ matrix used for evaluating the performance of a classification model, where N is the total number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

For a binary classification problem, we would have a 2×2 matrix, as shown below, with 4 values:

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Let's decipher the matrix:

- The target variable has two values: **Positive** or **Negative**
- The **columns** represent the **actual values** of the target variable
- The **rows** represent the **predicted values** of the target variable

But wait – what's TP, FP, FN, and TN here? That's the crucial part of a confusion matrix. Let's understand each term below.

Important Terms in a Confusion Matrix

True Positive (TP)

- The predicted value matches the actual value, or the predicted class matches the actual class.

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics Vidhya, you

agree to our [Privacy Policy](#) and [Terms of Use](#). [Accept](#)

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)

False Positive (FP) – Type I Error

- The predicted value was falsely predicted.
- The actual value was negative, but the model predicted a positive value.
- Also known as the type I error.

False Negative (FN) – Type II Error

- The predicted value was falsely predicted.
- The actual value was positive, but the model predicted a negative value.
- Also known as the type II error.

Let me give you an example to better understand this. Suppose we had a classification dataset with 1000 data points. We fit a classifier (say logistic regression or decision tree) on it and get the below confusion matrix:

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	560	60
	NEGATIVE	50	330

The different values of the Confusion matrix would be as follows:

- True Positive (TP) = 560, meaning the model correctly classified 560 positive class data points.
- True Negative (TN) = 330, meaning the model correctly classified 330 negative class data points.
- False Positive (FP) = 60, meaning the model incorrectly classified 60 negative class data points as belonging to the positive class.
- False Negative (FN) = 50, meaning the model incorrectly classified 50 positive class data points as belonging to the negative class.

This turned out to be a pretty decent classifier for our dataset, considering the relatively larger number of true positive and true negative values.

Remember the Type I and Type II errors. Interviewers love to ask the difference between these two! You can prepare for all this better from our Machine Learning Course Online.

Why Do We Need a Confusion Matrix?

Before we answer this question, let's think about a hypothetical classification problem.

Let's say you want to predict how many people are infected with a contagious virus in times before they show the symptoms and isolate them from the healthy population (ringing any bells, yet?). The two values for our target variable would be Sick and Not Sick.

Now, you must be wondering why we need a confusion matrix when we have our all-weather friend – Accuracy. Well, let's see where classification accuracy falters.

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)

Let's see how our model performed:

ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FN
8	0	1	FP
9	0	0	TN
10	1	0	FN
:	:	:	:
1000	0	0	TN

The total outcome values are:

$$TP = 30, TN = 930, FP = 30, FN = 10$$

So, the accuracy of our model turns out to be:

$$Accuracy = \frac{30 + 930}{30 + 30 + 930 + 10} = 0.96$$

96%! Not bad!

But it gives the wrong idea about the result. Think about it.

Our model is saying, "I can predict sick people 96% of the time". However, it is doing the opposite. It predicts the people who will not get sick with 96% accuracy while the sick are spreading the virus!

Do you think this is a correct metric for our model, given the seriousness of the issue? Shouldn't we be measuring how many positive cases we can predict correctly to arrest the spread of the contagious virus? Or maybe, out of the correct predictions, how many are positive cases to check the reliability of our model?

This is where we come across the dual concept of Precision and Recall.

Precision vs. Recall

“ Precision tells us how many of the correctly predicted cases actually turned out to be positive.

Here's how to calculate Precision:

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

This would determine whether our model is reliable or not.

“ Recall tells us how many of the actual positive cases we were able to predict correctly with our model.

And here's how we can calculate Recall:

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

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)

		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP (30)	FP (30)
	NEGATIVE	FN (10)	TN (930)

Sick people incorrectly predicted as not sick by the model Healthy people correctly predicted as not sick by the model

We can easily calculate Precision and Recall for our model by plugging in the values into the above questions:

$$\text{Precision} = \frac{30}{30 + 30} = 0.5$$

$$\text{Recall} = \frac{30}{30 + 10} = 0.75$$

50% percent of the correctly predicted cases turned out to be positive cases. Whereas 75% of the positives were successfully predicted by our model. Awesome!

C Precision is a useful metric in cases where False Positive is a higher concern than False Negatives.

Precision is important in music or video recommendation systems, e-commerce websites, etc. Wrong results could lead to customer churn and be harmful to the business.

C Recall is a useful metric in cases where False Negative trumps False Positive.

Recall is important in medical cases where it doesn't matter whether we raise a false alarm, but the actual positive cases should not go undetected!

In our example, Recall would be a better metric because we don't want to accidentally discharge an infected person and let them mix with the healthy population, thereby spreading the contagious virus. Now you can understand why accuracy was a bad metric for our model.

But there will be cases where there is no clear distinction between whether Precision is more important or Recall. What should we do in those cases? We combine them!

What Is F1-Score?

In practice, when we try to increase the precision of our model, the recall goes down, and vice-versa. The F1-score captures both the trends in a single value:

$$F1 - score = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$$

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)

result.

Confusion Matrix Using Scikit-learn in Python

You know the theory – now let's put it into practice. Let's code a confusion matrix with the [Scikit-learn \(sklearn\) library](#) in Python.

Python Code:

 @shivanshkausha/Confusion_Matrix_April-17-2020
 A Nix repl by shivanshkausha

[Open on Replit](#)

Show files

1

Run 45

Sklearn has two great functions: **confusion_matrix()** and **classification_report()**.

- Sklearn **confusion_matrix()** returns the values of the Confusion matrix. The output is, however, slightly different from what we have studied so far. It takes the rows as Actual values and the columns as Predicted values. The rest of the concept remains the same.
- Sklearn [**classification_report\(\)**](#) outputs precision, recall, and f1-score for each target class. In addition to this, it also has some extra values: **micro avg**, **macro avg**, and **weighted avg**

Mirco average is the precision/recall/f1-score calculated for all the classes.

$$\text{Micro avg Precision} = \frac{TP_1 + TP_2}{TP_1 + TP_2 + FP_1 + FP_2}$$

Macro average is the average of precision/recall/f1-score.

$$\text{Macro avg Precision} = \frac{P_1 + P_2}{2}$$

Weighted average is just the weighted average of precision/recall/f1-score.

Confusion Matrix for Multi-Class Classification

How would a confusion matrix work for a multi-class classification problem? Well, don't scratch your head! We will have a look.

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics Vidhya, you

agree to our [Privacy Policy](#) and [Terms of Use](#). [Accept](#)

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)

		ACTUAL VALUES		
		Facebook	Instagram	Snapchat
PREDICTED VALUES	Facebook	+ve 1	-ve 2	-ve 3
	Instagram	-ve 4	+ve 5	-ve 6
	Snapchat	-ve 7	-ve 8	+ve 9

The true positive, true negative, false positive, and false negative for each class would be calculated by adding the cell values as follows:

Facebook	Instagram	Snapchat
$TP = Cell_1$	$TP = Cell_5$	$TP = Cell_9$
$FP = Cell_2 + Cell_3$	$FP = Cell_4 + Cell_6$	$FP = Cell_7 + Cell_8$
$TN = Cell_5 + Cell_6 + Cell_8 + Cell_9$	$TN = Cell_1 + Cell_3 + Cell_7 + Cell_9$	$TN = Cell_1 + Cell_2 + Cell_4 + Cell_5$
$FN = Cell_4 + Cell_7$	$FN = Cell_2 + Cell_8$	$FN = Cell_3 + Cell_6$

That's it! You are ready to decipher any N x N confusion matrix!

Conclusion

The Confusion matrix is not so confusing anymore, is it? Hope this article gave you a solid base on how to interpret and use a confusion matrix for classification algorithms in machine learning. The matrix helps in understanding where the model has gone wrong and gives guidance to correct the path and it is a powerful and commonly used tool to evaluate the performance of a classification model in machine learning.

We will soon come out with an article on the AUC-ROC curve and continue our discussion there. Until next time, don't lose hope in your classification model; you just might be using the wrong evaluation metric!

Key Takeaways

- True Positive and True Negative values mean the predicted value matches the actual value.
- A Type I Error happens when the model makes an incorrect prediction, as in, the model predicted positive for an actual negative value.
- A Type II Error happens when the model makes an incorrect prediction of an actual positive value as negative.

Frequently Asked Questions

Q1. How is accuracy calculated from a confusion matrix?

A. Accuracy is calculated as the number of correct predictions divided by the total number of predictions made by the model.

Q2. How is precision calculated from a confusion matrix?

A. Precision is calculated as the number of true positives divided by the total number of positive predictions made by the model.

Q3. What is the F1 score in a confusion matrix?

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics Vidhya, you

agree to our [Privacy Policy](#) and [Terms of Use](#). [Accept](#)

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)

About the Author

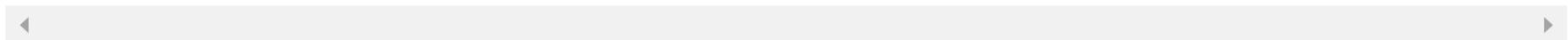


[Aniruddha Bhandari](#)

Our Top Authors



view more



Download

Analytics Vidhya App for the Latest blog/Article



Previous Post

[10 Productive Jupyter Notebook Hacks, Tips and Tricks!](#)

Next Post

[A Comprehensive Guide to 21 Popular Deep Learning Interview Questions and Answers](#)

18 thoughts on "Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)"



Punit Kumar says:

April 17, 2020 at 11:56 am

Hi Aniruddha, Thanks for writing this. Isn't the definition of FP and FN other way around? Like False Positive (FP) – The actual value was positive but we predicted a negative value Shouldn't it be reversed?

[Reply](#)



Arjun Badhan says:

April 17, 2020 at 12:29 pm

Hi Aniruddha, Thanks for the article. It is indeed informative. However, I would like to highlight something in the section with heading "Understanding True Positive, True Negative, False Positive and False Negative in a Confusion Matrix". Do you think that we might have mixed up on the second point on False Positive and False Negative.

[Reply](#)



Sagar says:

April 17, 2020 at 12:37 pm

Nice and well written article Aniruddha. Introduced confusion matrix very well for beginners. (But I think there is one minor issue you may want to correct if you also notice it is really an issue: Under Type1 and Type2 error definitions, I think you have to swap 2nd bullet points. These do not match with matrix you mentioned earlier.)

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics Vidhya, you

agree to our [Privacy Policy](#) and [Terms of Use](#). [Accept](#)

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)

[Reply](#)

Aniruddha Bhandari says:

April 17, 2020 at 1:38 pm

Hey Punit, Thanks for taking out the time to read the article and pointing out the mistake. Much appreciated! Thanks Aniruddha

[Reply](#)

Aniruddha Bhandari says:

April 17, 2020 at 1:46 pm

Hi Arjun, Glad you found it useful. And you are correct in pointing out the mix-up in the definitions. Thanks for your timely intervention 🎉 . Aniruddha

[Reply](#)

Aniruddha Bhandari says:

April 17, 2020 at 1:52 pm

Hey Sagar Really glad you liked the article! I have made the relevant changes. Thanks for the feedback! Aniruddha

[Reply](#)

Aniruddha Bhandari says:

April 17, 2020 at 1:55 pm

Hey Luciano Thanks for pointing it out. I have already made the changes. Aniruddha

[Reply](#)

Shahnawaz Sayyed says:

April 17, 2020 at 3:29 pm

Awesome!!! Very simple and crisp explaination. Thanks So much

[Reply](#)

Aniruddha Bhandari says:

April 17, 2020 at 6:15 pm

Glad to hear that you found it helpful!

[Reply](#)

Abid Saber says:

April 17, 2020 at 7:15 pm

Thanks for the tutorial, please How can we extract Tp, FP, FN, TN from the confusion matrix in the multiclass case with sklearn
thank you in advance

[Reply](#)

Aniruddha Bhandari says:

April 17, 2020 at 10:50 pm

Hi Abid If you want to calculate the values for a multiclass classification problem, you will need to calculate it manually as Sklearn returns the value of the Confusion matrix and not really the TP, FP, etc. (which was in the case of binary classification as we only had two classes). I have written the code for finding the TP, FP, etc. for one class of a multiclass classification problem [here](#). The rest I am pretty sure you will be able to manage yourself. If not, feel free to ask your doubts. Thanks Aniruddha

[Reply](#)

Ryan Tabeshi says:

April 18, 2020 at 4:00 am

Hi Puneet, In a FP, the value was predicted to be positive, but the value actually belonged to the negative class, so I think its correct, unless I'm missing something.

[Reply](#)

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)

[Reply](#)

MRB says:
June 29, 2020 at 7:01 am

Aniruddha: Too often confusion matrices deserve their unfortunate name, especially for students just starting to learn ML etc. Thanks for sharing a helpful tutorial & correcting the typo. 😊 It always helps (me, anyhow) to see some 'friendly' examples which I can follow along by myself so I'm sure others are going to find your contribution worthwhile too.

[Reply](#)

Trinkesh says:
July 20, 2020 at 8:17 am

great bro.

[Reply](#)

Prasad Karanjkar says:
September 09, 2020 at 3:01 am

Thanks Aniruddha- This was very helpful

[Reply](#)

Shailesh says:
November 19, 2022 at 10:22 pm

Yes. Interpretation need to be corrected here about FP.

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Notify me of follow-up comments by email.

Notify me of new posts by email.

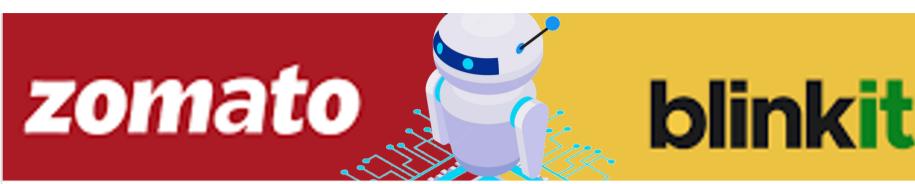
Submit

Top Resources

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics Vidhya, you

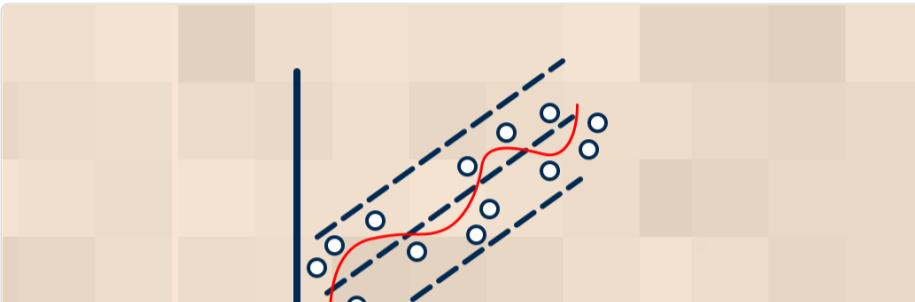
agree to our [Privacy Policy](#) and [Terms of Use](#). [Accept](#)

Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023)



[Zomato Embarks on Groundbreaking Artificial Intelligence](#)

K.sabreena - JUN 09, 2023



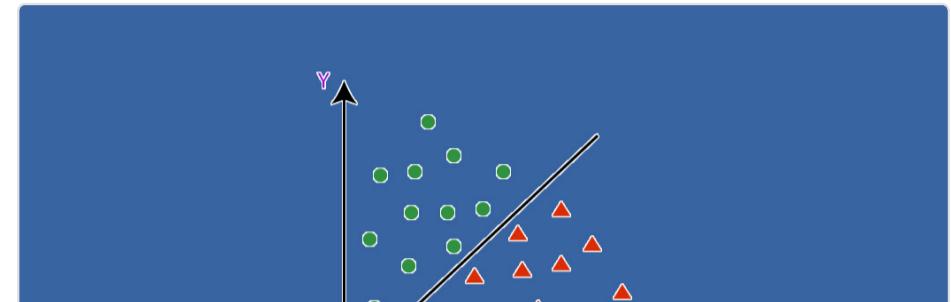
[Everything you need to Know about Linear Regression!](#)

KAVITA MALI - OCT 04, 2021



[Understand Random Forest Algorithms With Examples
\(Updated 2023\)](#)

Sruthi E R - JUN 17, 2021



[Support Vector Machine\(SVM\): A Complete guide for beginners](#)

Anshul Saini - OCT 12, 2021

Download App

[Analytics Vidhya](#)

[About Us](#)

[Our Team](#)

[Careers](#)

[Contact us](#)

[Companies](#)

[Post Jobs](#)

[Trainings](#)

[Hiring Hackathons](#)

[Advertising](#)

[Data Scientists](#)

[Blog](#)

[Hackathon](#)

[Discussions](#)

[Apply Jobs](#)

[Visit us](#)



© Copyright 2013-2023 Analytics Vidhya.

[Privacy Policy](#) [Terms of Use](#) [Refund Policy](#)

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics Vidhya, you agree to our [Privacy Policy](#) and [Terms of Use](#). [Accept](#)