CONFUSION MATRIX:

STRAIGHTFORWARD TO

PERFORMANCE EVALUATION

JINYOON KIM

PENNSTATE HARRISBURG

01/24/2024

TABLE OF CONTENTS

TABLE OF CONTENTS 2

Why use confusion matrix? 3

What is confusion matrix? 4

Basic Confusion Matrix Example (Binary Classification) 6

Evaluation Scores 8

Confusion matrix on multi-class classification 13

Confusion matrix on data imbalance 15

Reference 16

Why use confusion matrix?

The confusion matrix stands as a primary tool of machine learning models, especially in the realm of classification tasks. This tool not only presents a clear depiction of the accuracy of a model by displaying the number of correct and incorrect predictions but also delves deeper, revealing the nature and specifics of these predictions. Particularly in binary classification, it distinguishes between false positives and false negatives, the different types of errors which we will discuss further in the next section.

Beyond its essential role in depicting model accuracy, the confusion matrix forms the foundation for calculating more advanced performance metrics such as precision, recall, and the F1-score. These renowned evaluation metrics based on the results from the confusion matrix provide a higher dimension of perspective on a model's performance, effectively showcasing its strengths as well as its weaknesses.

In multi-class classification scenarios, the confusion matrix reveals its true potential by presenting the inter-class dynamics of model predictions. It identifies not only the overall performance but also how and where the model confuses different classes, offering invaluable insights. This information is crucial for guiding the model improvement process, whether through enriching the training data, refining features, or adjusting the model architecture. In essence, the confusion matrix transcends its role as a mere performance metric; it serves as a comprehensive diagnostic tool to improve the model, which makes it one of the most prominent evaluation frameworks in the modern machine learning field.

What is confusion matrix?

The most fundamental form of a confusion matrix arises from the performance evaluation of a classification problem, where the output can belong to two or more classes. It is a table that presents four distinct combinations of predicted and actual values, offering a comprehensive overview of the model's classification accuracy. This model is very useful for measuring metrics such as Recall, Precision, Specificity, Accuracy, and even AUC-ROC curves, which are representative of evaluation metrics in machine learning models. We will discuss these metrics in more detail later on. For now, we will focus on how the confusion matrix operates.

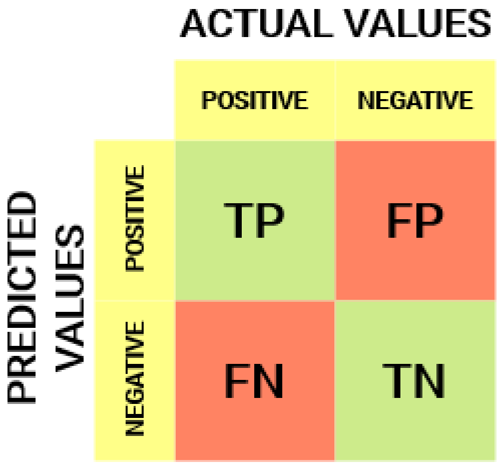
Let’s start with the most basic model that the confusion matrix can be used for: a binary classification task. When we mention binary classification, we consider situations where there is a “True” or “False” outcome that needs to be classified using the model. What we do with these is determine how accurately the model predicts the correct solution by comparing its prediction results with the actual results. Then, there can be four cases:

- Where prediction of the model is True and the actual result is True (True Positive, TP)

- Where prediction of the model is True but the actual result is False (True Negative, TN)

- Where prediction of the model False and the actual result is False (True Negative, TP)

- Where the prediction of the model is False but the actual result is True (False Negative, FN)



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

Figure 1. Binary Classification Confusion Matrix

Now, we understand there can be four cases. Figure 1 showcases these four different cases in four distinct sections, where the horizontal side represents the "Actual Result" as true or false, while the vertical side represents the "Predicted Values" as true or false. Each section counts and records the number of predictions that fall under their respective categories while the model evaluation is conducted on a set of data.

Basic Confusion Matrix Example (Binary Classification)

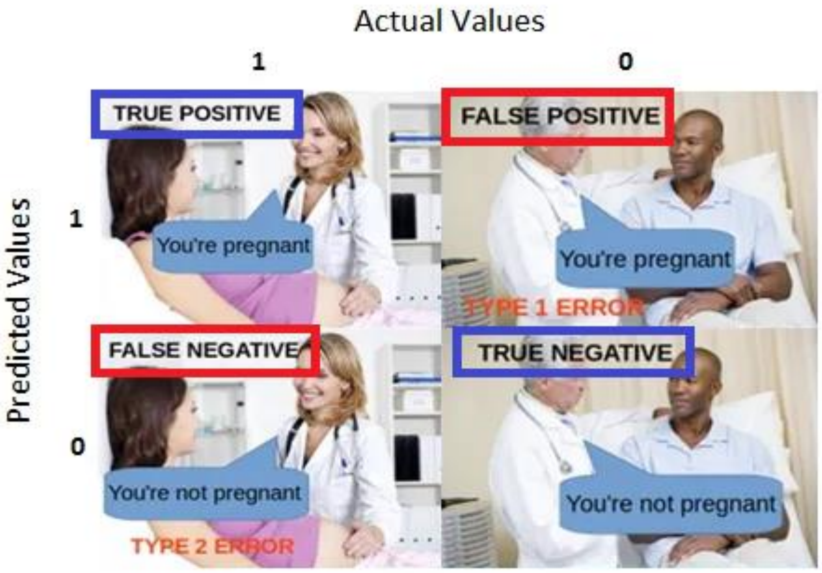


Figure 2. Confusion matrix example

Figure 2 illustrates a simple example of a confusion matrix using a pregnancy analogy for a binary classification task, which helps us understand the concept better. Let's suppose you are a doctor who needs to diagnose whether a patient is pregnant. As is evident, a woman can be pregnant while a man cannot.

If you predict that a woman is pregnant, and she indeed is, then it is a True Positive. If you predict that a man is not pregnant, and he indeed is not, then it is a True Negative. However, if you predict that a man is pregnant but he actually is not, then this is a False Positive, also known as **"Type 1 Error"**. Conversely, if you predict that a woman is not pregnant but she actually is, then this is a False Negative, also known as **"Type 2 Error"**.

If, for instance, we diagnose 1000 patients as evaluation data points, these patients would be distributed among these four categories. The count of patients falling into each category would then be represented by the respective numbers of TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives).

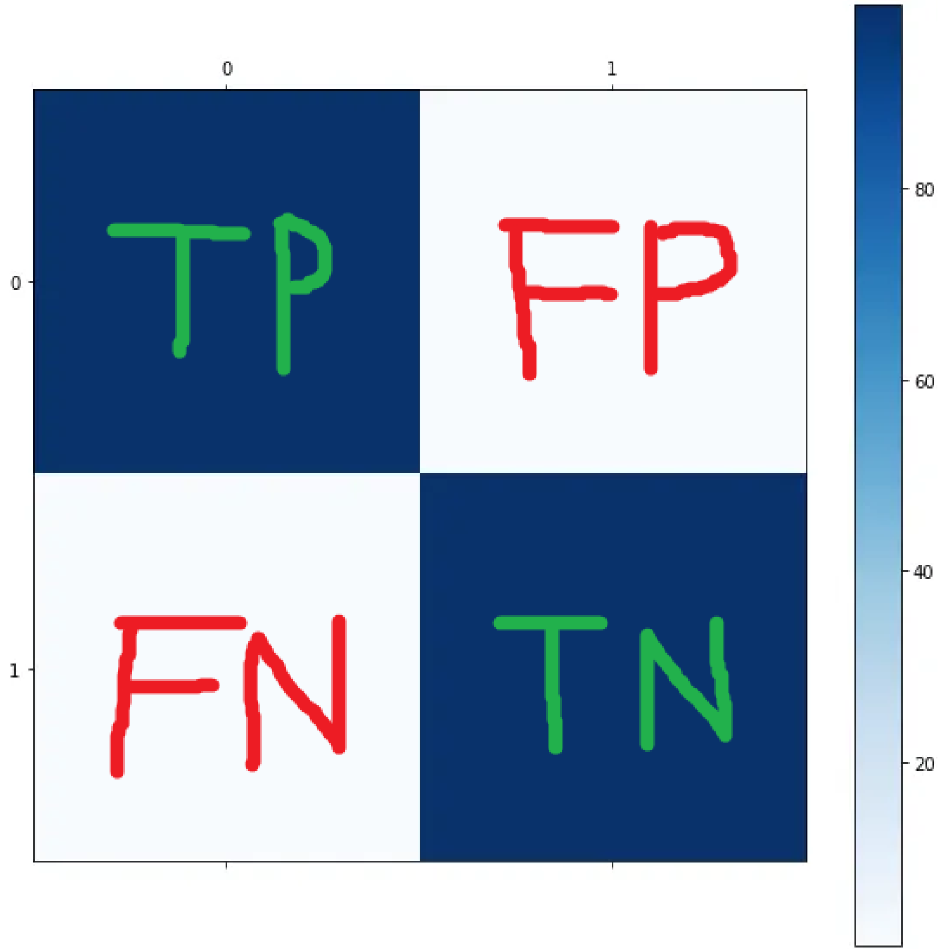


Figure 3. How implementation of confusion matrix would look like

The implementation of this confusion matrix is depicted in Figure 3, which is commonly represented using a heatmap. The heatmap indicates the **number of data points that fall into each category during evaluation**. The deeper the blue, the greater the number of data points in that category; conversely, the whiter the area, the fewer the data points. Since most of the data is concentrated in the True Positive and True Negative cells, we can infer that **the model evaluated using this confusion matrix is highly accurate**. If the model were not accurate, the diagonal cells representing True Positives and True Negatives would appear whiter, indicating fewer correct predictions, while the False Positive and False Negative cells would become bluer, indicating a higher number of incorrect predictions. This type of confusion matrix implementation is a standard practice in the evaluation of models for classification tasks which you will get to see a lot from now on.

Evaluation Scores

Now we understand how the confusion matrix works with basics. It is time for us to look over how the common evaluation metrics are derived from this confusion matrix. Let's start straightforwardly by examining how each evaluation metric is composed of the prediction types made by the confusion matrix.

- Accuracy

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

This metric measures how often the classifier makes correct predictions. It considers all classes (positive and negative) and evaluates how many of them were predicted correctly. Essentially, it's the ratio of the number of correct predictions to the total number of predictions made.

While accuracy is a clear and straightforward metric, it has its limitations, particularly in cases where the **dataset is imbalanced**.

***What is imbalanced dataset?** An imbalanced dataset is one where the number of observations in each class is not equally distributed. It's skewed, meaning specific classes have significantly more data than others.

For instance, let's revisit the pregnancy diagnosis example, where we are developing a model to predict pregnancy among patients, with only 1% of the patients actually being pregnant. Out of 1000 individuals, only 10 are pregnant.

A naive model might predict that no one is pregnant. This model would be correct 990 times and wrong only 10 times, failing to identify the 10 pregnant individuals.

$$\frac{990TN + 0TP}{990TN + 0TP + 10FN + 0FP} = \frac{990}{1000} = 99$$

Despite its terrible identifying quality, this model would still achieve 99% accuracy according to the formula! Clearly, this is not the result we desire. To address this fallacy and gain a more comprehensive perspective, we need to adopt additional metrics such as Precision, Recall, and the F-1 Score.

- Precision

Now, due to the limitations of accuracy as a metric, the dual concepts of Precision and Recall comes into play.

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

Precision informs us about the **proportion of correctly predicted positive cases that are indeed positive**, helping to determine the reliability of our model. High precision indicates that the model has a low rate of false positives. In other words, when a model with high precision predicts a case as positive, you can trust it with confidence. However, it does **not indicate how well the model is at detecting all the positive instances**.

This evaluation metric is particularly important in scenarios where the consequences of a false positive are severe. For example, in email spam detection, a model with high precision is essential, as you want to avoid any situation where even a single crucial email is mistakenly marked as spam.

- Recall

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

In contrast, the Recall metric concentrates on the actual **positive instances**, addressing the question: "**Of all the actual positive instances, how many did the model successfully identify**?" High recall indicates that the model is proficient at detecting positive instances. However, it does not account for the number of negative instances that were incorrectly labeled as positive (false positives).

Recall becomes critical in scenarios where the consequences of a false negative are severe. For instance, in medical diagnostics, raising a false alarm may be acceptable, but it's imperative that actual positive cases are not overlooked!

Precision vs Recall the Trade off. There is often trade-off between precision and recall while training the model. Improving one metric typically reduces the other and vice versa. This can be adjusted depending on the needs using different strategies and one common method is: F1 score.

- F1 Score

$$\text{F1 Score} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

It is challenging to compare two models when one exhibits low precision and high recall, or vice versa. Therefore, to facilitate comparison, we use the F1 Score. The F1 Score effectively measures both Recall and Precision simultaneously. It employs the Harmonic Mean instead of the Arithmetic Mean, which penalizes extreme values more severely and tends towards the smaller of the two elements. This approach ensures a more balanced evaluation of the two metrics.

Consequently, **if either precision or recall is low, the F1 Score will also be low**. The F1 Score ranges from 0 to 1 and reaches its maximum value when **Precision equals Recall**. Thus, the F1 Score inherently strives to **balance precision and recall**, existing as a singular metric that encloses the equilibrium between these two metrics.

- AUC-ROC Curves

As we might have learned from the concepts of classification models, particularly binary classifiers, these models output a **probability score**. This score indicates the likelihood of an instance belonging to the positive class; otherwise, it belongs to the negative class. The score typically ranges from 0 to 1.

The threshold is a value within this range that acts as a decision boundary. If the model's output score for an instance is above the threshold, the instance is classified as the positive class. If it's below the threshold, the instance is classified as the negative class. The choice of threshold affects how the model's predictions are categorized into positive and negative, which in turn affects metrics like precision, recall, and the F1 Score.

The AUC-ROC curve is a performance measurement for classification problems that offers a distinct perspective from the F1 Score for this threshold perspective. While the F1 Score provides a single metric based on a specific threshold, enclosing the balance between precision and recall, the AUC-ROC curve offers a comprehensive view of a model's performance across all possible thresholds.

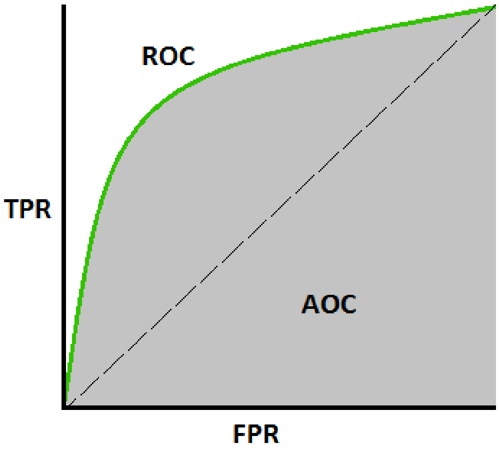


Figure 4. AUC-ROC curve illustration

$$\text{True Positive Rate} = \text{Recall} = \frac{TP}{TP+FN}$$

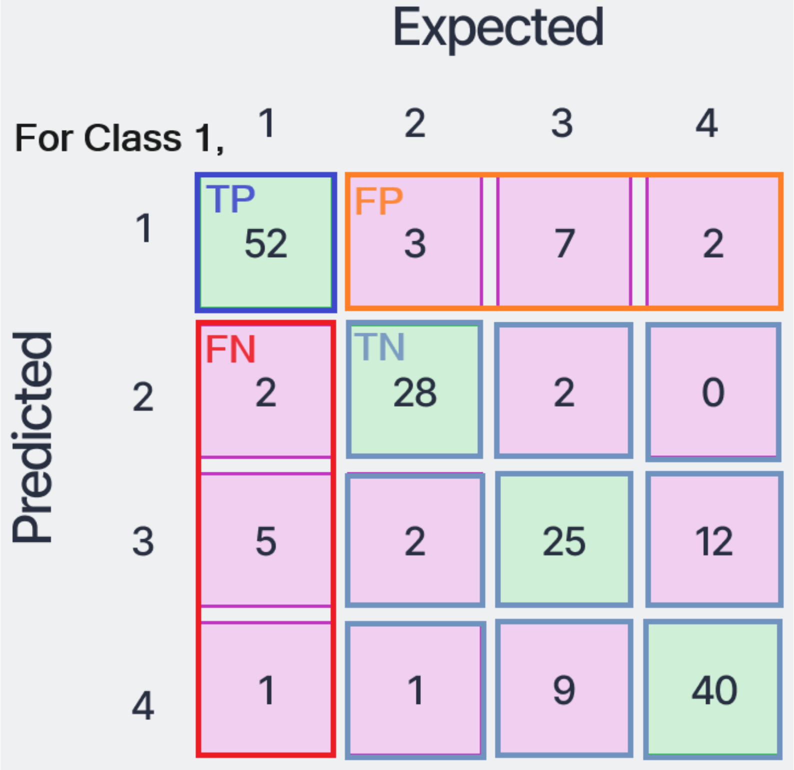
$$\text{False Positive Rate} = \frac{FP}{FP+TN}$$

ROC (Receiver Operating Characteristic) Curve plots the **True Positive Rate (TPR = Recall)** on the Y-Axis and **False Positive Rate (FPR)** on the X-axis for different threshold values. AUC (Area under the ROC Curve) represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is **predicting positive as positive and negative as negative**.

While the F1 Score is useful for getting a quick understanding of model performance at a particular threshold, the AUC-ROC offers a more nuanced interpretation, showing how well the model can separate the positive and negative classes over the entire range of thresholds. It's particularly useful when the optimal threshold is not known or when you want a performance metric that is robust to changes in the class distribution.

The AUC is a metric that provides a singular, comprehensive value representing the performance of a classification model over all possible thresholds. By condensing the information from the ROC curve into a single scalar, the AUC facilitates easy comparison between models, offers robustness against threshold variability, and remains invariant to class distribution in the dataset. Especially useful in comparing models and evaluating performance on imbalanced datasets, the AUC serves as an aggregate measure of a model's ability to distinguish between classes, with a higher AUC indicating superior model performance.

Confusion matrix on multi-class classification



			Expected		
	For Class 1,	1	2	3	4
Predicted	1	TP 52	FP 3	7	2
	2	FN 2	TN 28	2	0
	3	5	2	25	12
	4	1	1	9	40

Figure 5. multi-class classification confusion matrix

When classification tasks extend to the multiclass level, the size of the confusion matrix expands from a 2x2 to an NxN dimension, where N represents the number of classes. A key point is that True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are **calculated separately for each class**. These values change depending on the **class under consideration**

and can then be averaged to obtain the overall score. This concept is illustrated in the confusion matrix provided in Figure 4.

True Positives (TP): These are the diagonal elements and vary depending on the class perspective. For example, there are 52 TPs for class 1, 28 for class 2, 25 for class 3, and 40 for class 4. TPs indicate the instances that were correctly predicted for each respective class.

True Negatives (TN): For a given class, TNs consist of all the correct predictions that are not associated with that class. For instance, for class 1, TNs would be the sum of all cells not in row 1 or column 1. When calculating the TN of class 2, TNs shift to become the sum of all cells not in row 2 or column 2, and so on for classes 3 and beyond.

False Positives (FP): For a given class, FPs encompass all instances that were incorrectly predicted to belong to that class. For class 1, this would be the sum of all cells in row 1, excluding the TP cell for class 1. This sum changes to include all cells in row 2, excluding the TP cell for class 2 (located in the second row and second column), and so on for subsequent classes.

False Negatives (FN): For a given class, FNs are the instances that truly belong to that class but were predicted to be something else. For class 1, FNs would be the sum of all cells in column 1, excluding the TP cell for class 1. This sum changes to include all cells in column 2, excluding the TP cell for class 2, and continues in this pattern for each class.

After determining the TP, TN, FP, and FN for each class (1, 2, 3, ..., N), we calculate the precision, recall, and F1 score for each. These can then be averaged or combined in another preferred manner to produce the overall precision, recall, and F1 score.

Further Confusion Matrix Sample Code and the impact of the data imbalance

Click Here to Access the Colab Notebook.

Reference

- Narkhede, S. (2018, May 9). Understanding confusion matrix. *Medium*. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

- Bhandari, A. (2024, January 11). Understanding & interpreting confusion matrix in machine learning (Updated 2024). *Analytics Vidhya.* https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

- Shaffi, A. (2021, March 15). Building a confusion matrix from scratch. *Medium*. https://medium.datadriveninvestor.com/building-a-confusion-matrix-from-scratch-85a8bfb97626

- Narkhede, S. (2018, June 26). Understanding AUC - ROC curve. *Medium*. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

- Kundu, R. (2022, September 13). Confusion matrix: How to use it & interpret results [Examples]. *V7 Labs*. https://www.v7labs.com/blog/confusion-matrix-guide

- https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/

- Bharathi. (2023, December 28). Latest guide on confusion matrix for multi-class classification. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/