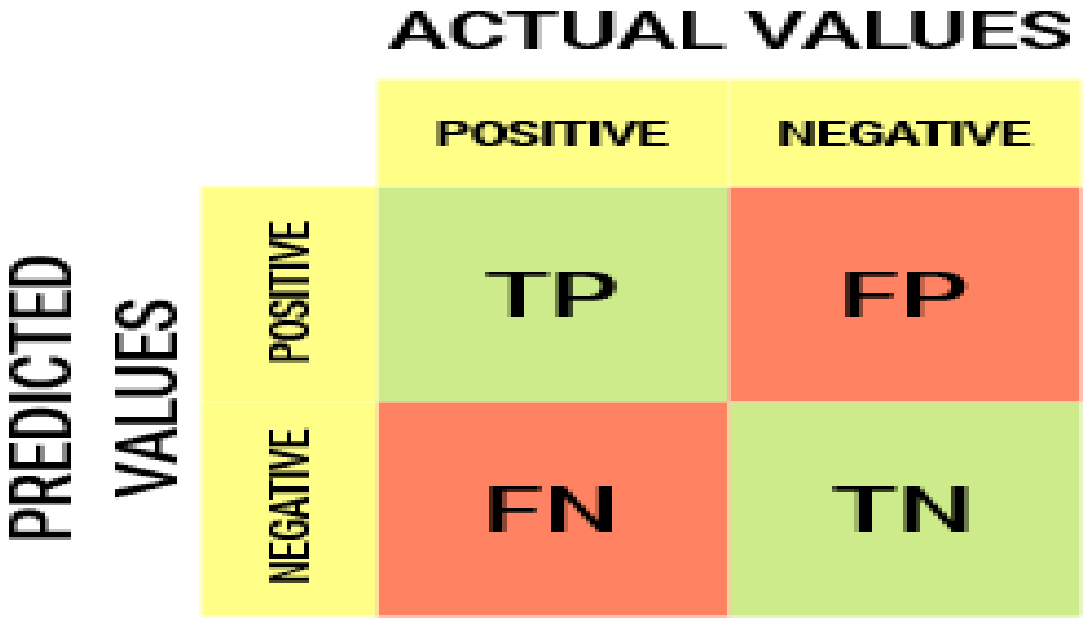
Confusion Matrix

What is a Confusion Matrix?

- A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the 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 x 2 matrix as shown below with 4 values



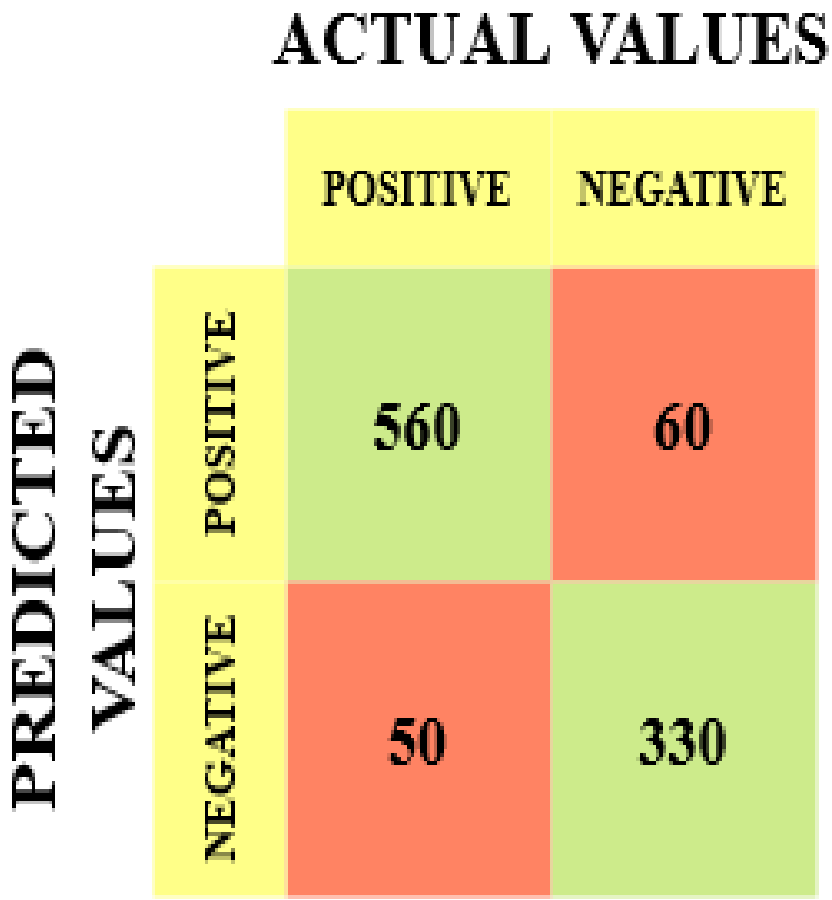
- Let's decipher the matrix:
- The target variable has two values: **Positive** or **Negative**
- The **columns** represent the **actual values** of the target variable
- The **rows** represent the **predicted values** of the target variable

Understanding True Positive, True Negative, False Positive and False Negative in a Confusion Matrix

- **True Positive (TP)**
  - The predicted value matches the actual value
  - The actual value was positive and the model predicted a positive value
- **True Negative (TN)**
  - The predicted value matches the actual value
  - The actual value was negative and the model predicted a negative value
- **False Positive (FP) – Type 1 error**
  - The predicted value was falsely predicted
  - The actual value was negative but the model predicted a positive value
  - Also known as the **Type 1 error**
- **False Negative (FN) – Type 2 error**
  - The predicted value was falsely predicted
  - The actual value was positive but the model predicted a negative value
  - Also known as the **Type 2 error**

Example

- Suppose we had a classification dataset with 1000 data points. We fit a classifier on it and get the below confusion matrix:
- The different values of the Confusion matrix would be as follows:
- **True Positive (TP)** = 560; meaning 560 positive class data points were correctly classified by the model
- **True Negative (TN)** = 330; meaning 330 negative class data points were correctly classified by the model
- **False Positive (FP)** = 60; meaning 60 negative class data points were incorrectly classified as belonging to the positive class by the model
- **False Negative (FN**) = 50; meaning 50 positive class data points were incorrectly classified as belonging to the negative class by the model



Why Do We Need a Confusion Matrix?

- 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 do we need a confusion matrix when we have our all-weather friend – Accuracy? Well, let’s see where accuracy falters.

- Our dataset is an example of an **imbalanced dataset**.
- There are 947 data points for the negative class and 3 data points for the positive class.
- This is how we'll calculate the accuracy:

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

- 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	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN

96%! Not bad!

The total outcome values are:

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

So, the accuracy for our model turns out to be: $$Accuracy = \frac{30 + 930}{30 + 30 + 930 + 10} = 0.96$$

96%! Not bad!

- But it is giving 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 is predicting 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 correctly predicted cases, 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

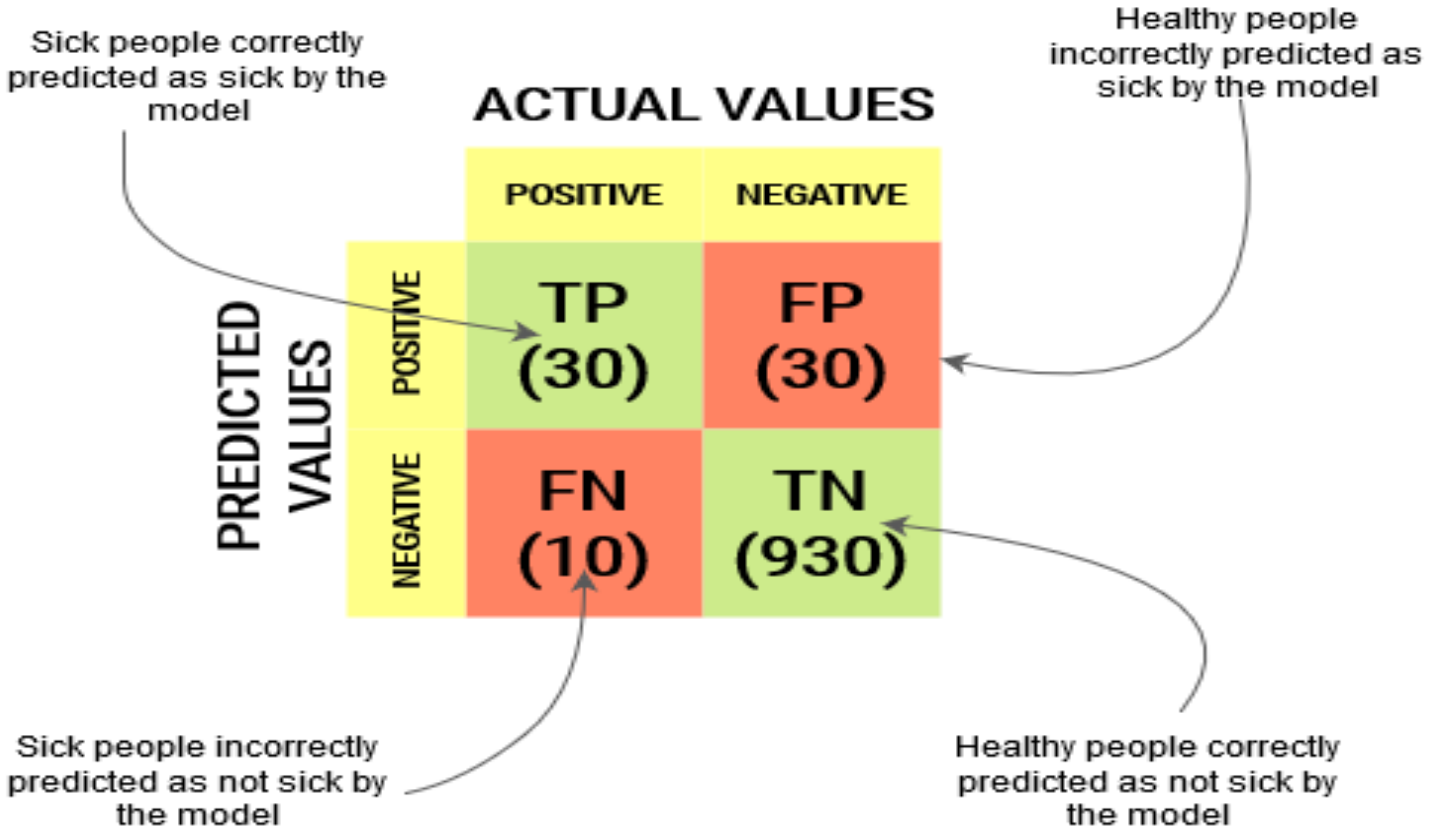
Precision vs. Recall

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



- 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.
- 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

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

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

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

- 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

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:

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

- **F1-score is a harmonic mean of Precision and Recall**, and so it gives a combined idea about these two metrics. It is maximum when Precision is equal to Recall