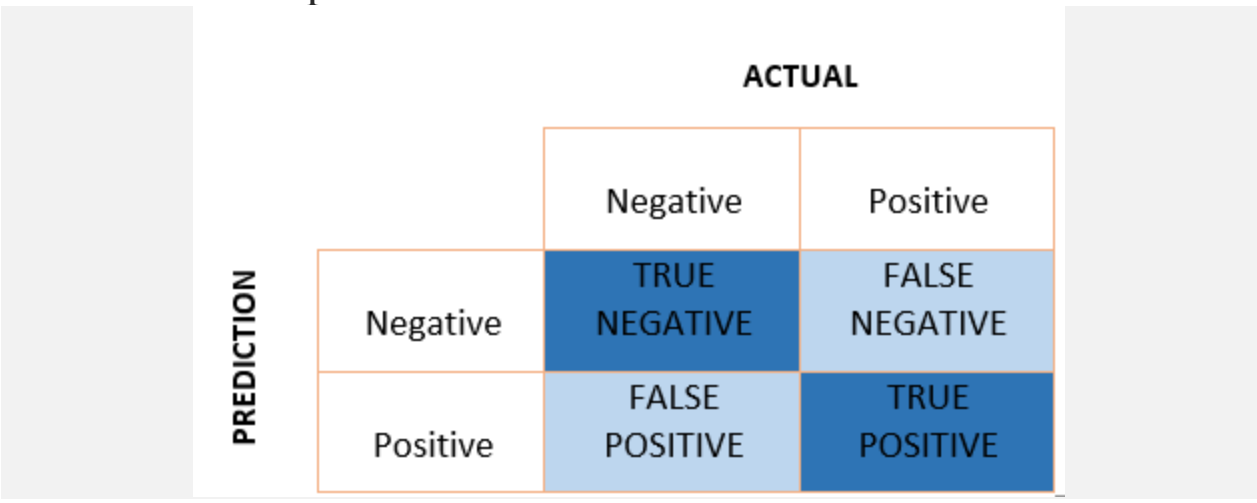
# Performance Metrics: Confusion matrix, Precision, Recall, and F1 Score

Accuracy performance metrics can be decisive when dealing with imbalanced data. In this blog, we will learn about the Confusion matrix and its associated terms, which looks confusing but are trivial. The confusion matrix, precision, recall, and F1 score gives better intuition of prediction results as compared to accuracy. To understand the concepts, we will limit this article to binary classification only.

## What is a confusion matrix?

It is a matrix of size 2×2 for binary classification with actual values on one axis and predicted on another.




Confusion Matrix

Let's understand the confusing terms in the confusion matrix: **true positive, true negative, false negative, and false positive** with an example.

## EXAMPLE

A machine learning model is trained to predict tumor in patients. The test dataset consists of 100 people.

| | | ACTUAL | |
|---|---|---|---|
| | | Negative | Positive |
| PREDICTION | Negative | 60 | 8 |
| | Positive | 22 | 10 |

Confusion Matrix for tumor detection

**True Positive (TP)** — model correctly predicts the positive class (prediction and actual both are positive). In the above example, **10 people** who have tumors are predicted positively by the model.
**True Negative (TN)** — model correctly predicts the negative class (prediction and actual both are negative). In the above example, **60 people** who don't have tumors are predicted negatively by the model.
**False Positive (FP)** — model gives the wrong prediction of the negative class (predicted-positive, actual-negative). In the above

example, **22 people** are predicted as positive of having a tumor, although they don't have a tumor. FP is also called a **TYPE I** error. **False Negative (FN)** — model wrongly predicts the positive class (predicted-negative, actual-positive). In the above example, **8 people** who have tumors are predicted as negative. FN is also called a **TYPE II** error.

With the help of these four values, we can calculate True Positive Rate (TPR), False Negative Rate (FPR), True Negative Rate (TNR), and False Negative Rate (FNR).

$$TPR = \frac{TP}{Actual\ Positive} = \frac{TP}{TP + FN}$$

$$FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP + FN}$$

$$TNR = \frac{TN}{Actual\ Negative} = \frac{TN}{TN + FP}$$

$$FPR = \frac{FP}{Actual\ Negative} = \frac{FP}{TN + FP}$$

Even if data is imbalanced, we can figure out that our model is working well or not. For that, **the values of TPR and TNR should be high, and FPR and FNR should be as low as possible.**

With the help of TP, TN, FN, and FP, other performance metrics can be calculated.

## Precision, Recall

Both precision and recall are crucial for information retrieval, where positive class mattered the most as compared to negative. **Why?**

While searching something on the web, the model does not care about something **irrelevant** and **not retrieved** (this is the true negative case). Therefore only TP, FP, FN are used in Precision and Recall.

### Precision

Out of all the positive predicted, what percentage is truly positive.

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

The precision value lies between 0 and 1.

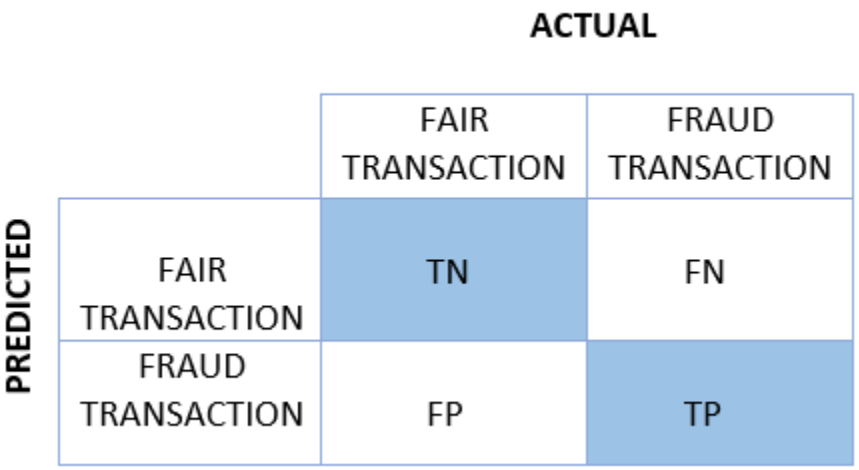
### Recall

Out of the total positive, what percentage are predicted positive. It is the same as TPR (true positive rate).

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

**How are precision and recall useful?** Let's see through examples.

## EXAMPLE 1- Credit card fraud detection

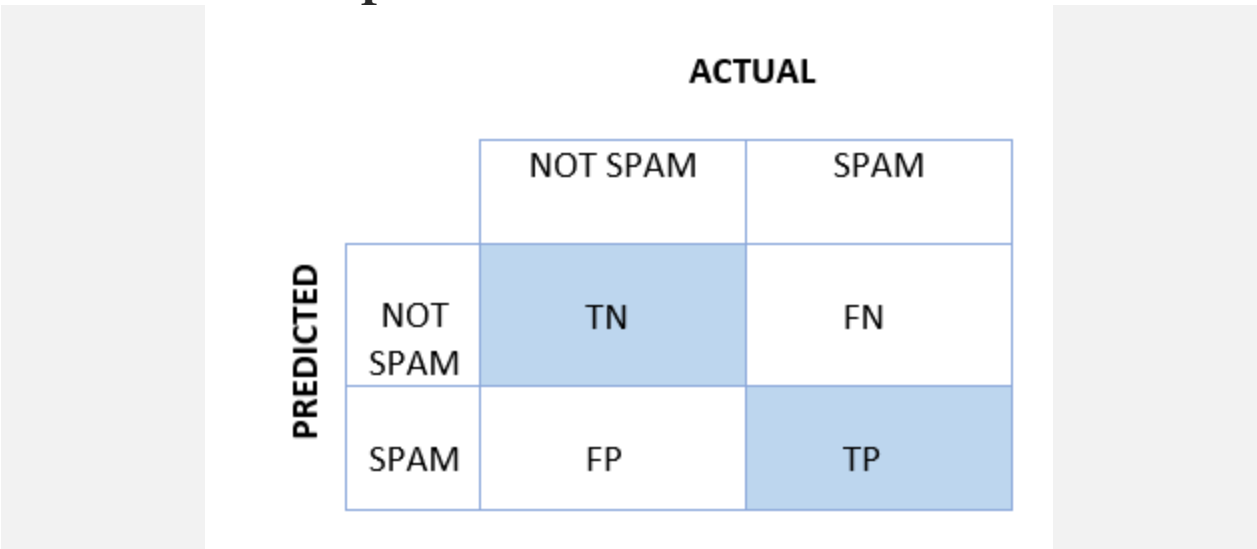



Confusion Matrix for Credit Card Fraud Detection

We do not want to **miss any fraud transactions**. Therefore, we **want False-Negative to be as low as possible.** In these situations, we can compromise with the low precision, but recall should be high. Similarly, in the medical application, we don't want to miss any patient. Therefore we focus on having a high recall.

So far, we have discussed when the recall is important than precision. **But, when is the precision more important than recall?**

## EXAMPLE 2 — Spam detection




Confusion Matrix for Spam detection

In the detection of spam mail, it is okay if any spam mail remains undetected (false negative), but what if we miss any critical mail because it is classified as spam (false positive). In this situation, **False Positive** should be as low as possible. Here, precision is more vital as compared to recall.

When comparing different models, it will be difficult to decide which is better (high precision and low recall or vice-versa). Therefore, there should be a metric that combines both of these. One such metric is the F1 score.

### F1 Score

It is the harmonic mean of precision and recall. It takes both false positive and false negatives into account. Therefore, it performs well on an imbalanced dataset.

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

F1 score gives the same weightage to recall and precision.

There is a **weighted F1 score** in which we can give different weightage to recall and precision. As discussed in the previous section, different problems give different weightage to recall and precision.

$$F_\beta = (1 + \beta^2) * \frac{(Precision * Recall)}{(\beta^2 * Precision) + Recall}$$

**Beta represents how many times recall is more important than precision**. If the recall is twice as important as precision, the value of Beta is 2.

## Conclusion

Confusion matrix, precision, recall, and F1 score provides better insights into the prediction as compared to accuracy performance metrics. Applications of precision, recall, and F1 score is in information retrieval, word segmentation, named entity recognition, and many more.

## Class Activity:

The Software quality engineering **(SQE)** team of Avanza Solution Private Limited Karachi is going to test the quality of a Machine Learning model which is being implemented for a Software Product of one of its renowned Client National Foods Private Limited Karachi. The overall performance of the application is not so good after the deployment of the first released of the software product onto the Production server therefore we need to find out the defects from this application source code. Application Source Code pertains 800 defects detected from the Source code repository. The **(SQE)** team classified the defects into "OK" and "Faulty" category. The further classification found as under the following criteria:

1) 200 faulty defects were classified as "OK"
2) 100 faulty defects were classified as "Faulty"
3) 400 non-faulty defects were classified as "OK"
4) 100 non-faulty defects were classified as "Faulty"

Perform the software quality engineering analysis on the above Source code criteria collecting from defects tracking and preventions team to find out the Quality of the Source Code:

a) Compute confusion matrix?
b) Find out the Quality factor accuracy of Client Source Code?
c) What will be the Sensitivity of the Code?
d) Calculate Precision of the Code?
e) Calculate Type I and Type II error in your calculations?
f) Calculate the metric F1-Score?