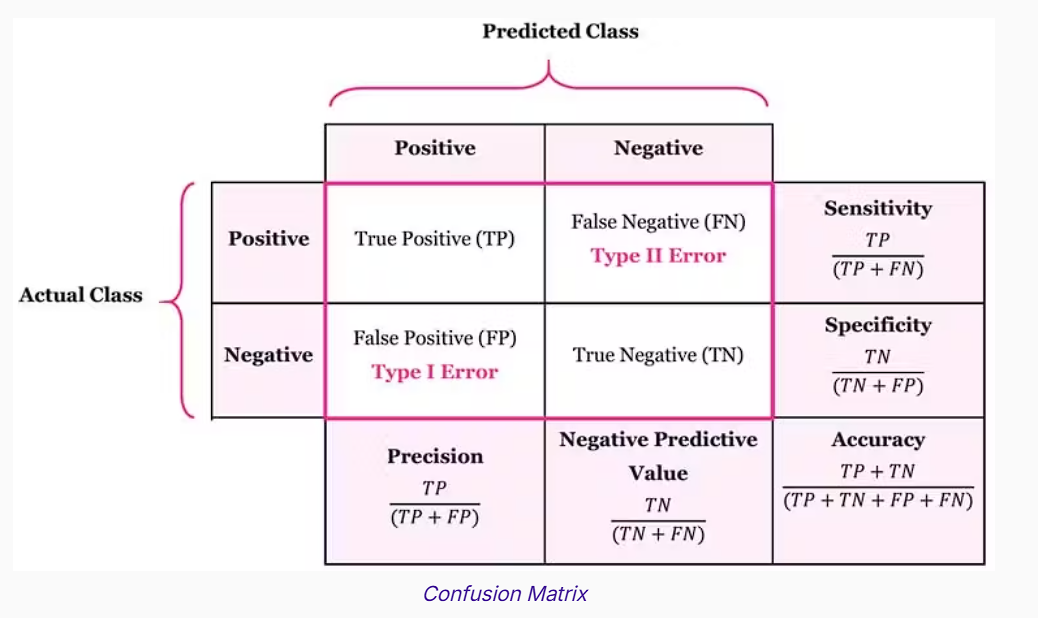
# Confusion Matrix

**Definition :**

A confusion matrix is a performance evaluation tool used in machine learning that summarizes the performance of a classification model by tabulating true positive, true negative, false positive, and false negative predictions. It helps assess the accuracy and effectiveness of the model's predictions. The matrix is based on the concepts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). It provides a granular view of a model's performance across different classes.



Recall

Precision

Precision

In any project there will be some certain set of ways to measure the performance of any machine runtime.

Mainly in Machine Learning, Data Science, and system security we have various performance measures which are being used to evaluate accuracy, efficiency and effectiveness.

# Precision :

 Precision evaluates the proportion of true positive predictions among all positive predictions (TP / (TP + FP)). This metric is crucial when the cost of false positives is high.

* Measures how many predicted **positive cases correctly.**
* Useful in fraud detection, medical diagnosis..etc…,

*Precision =*

# Recall (Sensitivity or True Positive Rate):

Recall measures the ratio of true positive predictions to the actual number of positive instances (TP / (TP + FN)). This metric is significant when missing positive instances is costly.

* Measures how well the model identifies **actual positive cases** are….
* Important in span detection and safety applications.

*Recall =*

# Accuracy :

Accuracy quantifies the ratio of correct predictions (TP and TN) to the total number of predictions. While informative, this metric can be misleading when classes are imbalanced.

* Overall correctness of the model.
* Useful when data is balanced

*Accuracy =*

# F1 Score :

The F1-Score strikes a balance between precision and recall, making it useful when both false positives and false negatives carry similar importance.

* Harmonic mean of precision and recall
* Used when there’s an imbalance between classes

F1 Score = 2 x

**True Positives (TP)**: Instances where the model correctly predicts a positive class when it is indeed positive. Consider a cancer diagnostic model: a true positive would occur when the model correctly identifies a patient with cancer as having the disease. TP is a vital measure of the model's ability to recognize positive instances accurately.

**True Negatives (TN)**: Instances where the model correctly predicts a negative class when it is indeed negative. Continuing the medical analogy, a true negative would be when the model correctly identifies a healthy patient as not having the disease. TN reflects the model's proficiency in recognizing negative instances.

**False Positives (FP)**: Instances where the model incorrectly predicts a positive class when it should have been negative. In the medical scenario, a false positive would mean the model wrongly indicates a patient has the disease when they are, in fact, healthy. FP illustrates instances where the model exhibits overconfidence in predicting positive outcomes.

**False Negatives (FN):** Instances where the model incorrectly predicts a negative class when it should have been positive. In the medical context, a false negative would be when the model fails to detect a disease in a patient who actually has it. FN highlights situations where the model fails to capture actual positive instances.

**Miss classification rate :**

**= FP + FN / total (TP + FP + TN + FN)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | prediction | |
| **positive +** | **negative -** |
| Actual | **positive +** | TP =95 | FN = 5 |
| **negative -** | FP =5 | TN = 45 |
|  |  | yes predictions = 100 | no predictions = 50 |
|  |  |  |  |
|  | lets take sample data set of 150 in which 100 are true and 50 are false | | |
|  |

precision

= TP / Total positive predictions(TP + FP)

= 100/150

= 66.66 %

# Recall / sensitivity:

= TP / TP + FN

=positive rate / total actual positive (actual + predictions)

= 100/ (100 + 10)

= 90.09%

# F1 score :

= 2\*(66.66 \* 75 / 66.66 + 75) = 70.5 %

Miss classification rate = FP + FN / total

=10+40 /200

=25%

# Accuracy:

= TP + TN / total set ( TP + FP +TN + FN)

=140/200

= 70%

outliers

**MAE: mean absolute error**

A = [10, 20, 30]

P= [ 11 , 23 , 35]

E= [-1, -3 , -5]

= |10-11| + |20-23| + |30-35| / 3

= 9/3

=3

**MSE : mean squared error**

A = [10, 20, 30]

P= [ 11 , 23 , 35]

E= [-1, -3 , -5]

**= 1 + 9 + 25 /3**

**=11.666**

**RMSE: root mean squared error**

RMSE =

from sklearn.metrics import confusion\_matrix

import numpy as np

def compute\_metrics(y\_true, y\_pred):

    # Compute confusion matrix

    TN, FP, FN, TP = confusion\_matrix(y\_true, y\_pred).ravel()

    # Accuracy

    accuracy = (TP + TN) / (TP + TN + FP + FN)

    # Precision (PPV)

    precision = TP / (TP + FP) if (TP + FP) != 0 else 0

    # Recall (Sensitivity, TPR)

    recall = TP / (TP + FN) if (TP + FN) != 0 else 0

    # Specificity (TNR)

    specificity = TN / (TN + FP) if (TN + FP) != 0 else 0

    # F1 Score

    f1\_score = 2 \* (precision \* recall) / (precision + recall) if (precision + recall) != 0 else 0

  # False Positive Rate (FPR)

    fpr = FP / (FP + TN) if (FP + TN) != 0 else 0

    # False Negative Rate (FNR)

    fnr = FN / (FN + TP) if (FN + TP) != 0 else 0

    # False Discovery Rate (FDR)

    fdr = FP / (FP + TP) if (FP + TP) != 0 else 0

    # False Omission Rate (FOR)

    for\_rate = FN / (FN + TN) if (FN + TN) != 0 else 0

    # Prevalence

    prevalence = (TP + FN) / (TP + TN + FP + FN)

    # Balanced Accuracy

    balanced\_accuracy = (recall + specificity) / 2

    # Matthews Correlation Coefficient (MCC)

    mcc\_numerator = (TP \* TN) - (FP \* FN)

    mcc\_denominator = np.sqrt((TP + FP) \* (TP + FN) \* (TN + FP) \* (TN + FN))

    mcc = mcc\_numerator / mcc\_denominator if mcc\_denominator != 0 else 0

    # Fowlkes-Mallows Index (FMI)

    fmi = np.sqrt(precision \* recall)

    return {

        "Accuracy": accuracy,

        "Precision (PPV)": precision,

        "Recall (Sensitivity, TPR)": recall,

        "Specificity (TNR)": specificity,

        "F1 Score": f1\_score,

        "False Positive Rate (FPR)": fpr,

        "False Negative Rate (FNR)": fnr,

        "False Discovery Rate (FDR)": fdr,

        "False Omission Rate (FOR)": for\_rate,

        "Prevalence": prevalence,

        "Balanced Accuracy": balanced\_accuracy,

        "Matthews Correlation Coefficient (MCC)": mcc,

        "Fowlkes-Mallows Index (FMI)": fmi

    }

# Example usage

y\_true = [1, 0, 1, 1, 0, 1, 0, 0, 1, 0]  # Actual labels

y\_pred = [1, 0, 1, 0, 0, 1, 1, 0, 1, 0]  # Predicted labels

metrics = compute\_metrics(y\_true, y\_pred)

for metric, value in metrics.items():

    print(f"{metric}: {value:.4f}")