Confusion Matrix:

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known.

Example confusion matrix for a binary classifier

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

- There are two possible predicted classes: "yes" and "no". If we were predicting the presence of a disease, for example, "yes" would mean they have the disease, and "no" would mean they don't have the disease.
- The classifier made a total of 165 predictions (e.g., 165 patients were being tested for the presence of that disease).
- Out of those 165 cases, the classifier predicted "yes" 110 times, and "no" 55 times.
- In reality, 105 patients in the sample have the disease, and 60 patients do not.

Basic terms:

- **true positives (TP):** These are cases in which we predicted yes (they have the disease), and they do have the disease.
- **true negatives (TN):** We predicted no, and they don't have the disease.
- **false positives (FP):** We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- **false negatives (FN):** We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

Added these terms to the confusion matrix, and also added the row and column totals:

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Performance Measures:

- **Accuracy:** Overall, how often is the classifier correct?
  - **(TN+TP)/total** = (50+100)/165 = 0.91

- **Error Rate:** Overall, how often is it wrong?
  - **(FN+FP)/total** = (5+10)/165 = 0.09

- **Precision:**
  - **TP/(TP+FP)** = 100/110 = 0.90

- **Recall:** When it's actually yes, how often does it predict yes?
  - **TP/(TP+FN)** = 100/105 = 0.95

- **Specificity:** When it's actually no, how often does it predict no?
  - **TN/actual no** = 50/60 = 0.83

- **F1 Score:** Considered perfect when it’s 1, while the model is a total failure when it’s 0.
  - **2*((Precision*Recall / Precision+Recall))**

    = 2*((0.90*0.95) / (0.90+0.95))

    = 2*((0.85) / (1.85))

    = 2*0.45

    = 0.91