Classification Metrics

Understanding Model Predictions

- Consider a problem of classifying emails as spam or not spam
- An ML model is built to do the classification
- Four possible outcomes

Case 1	Case 2	Case 3	Case 4
The email was not spam and the model also predicted that it would not be spam	The email was not spam but the model predicted that it would be spam	The email was spam and the model also predicted that it would be spam	The email was spam but the model predicted that it would not be spam

Confusion Matrix

- A table that helps you visualize the performance of a classification model
- Especially useful when you have a binary classification problem
    - Email: spam vs not spam
    - Loan Repayment Default: yes vs no
- Can be extended for classification task with more than two classes too

	Predicted Spam	Predicted Not Spam
Actual Spam	40	5
Actual Not Spam	5	50

Confusion Matrix

- **True Positives (TP):** The model predicted the class as positive, and it is actually positive.
- **True Negatives (TN):** The model predicted the class as negative, and it is actually negative.
- **False Positives (FP):** The model predicted the class as positive, but it is actually negative
  - Also known as a **Type I error**
- **False Negatives (FN):** The model predicted the class as negative, but it is actually positive
  - Also known as a **Type II error**

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Accuracy

- A basic measure used to evaluate the overall performance of a classification model
- Represents the ratio of correctly predicted instances to the total instances
- Example:
  - Out of 100 emails, an ML model correctly classified 90 emails as either spam or not spam
  - Accuracy = (# Correct Predictions) / (# Total Predictions) = 90/100 = 0.9 (or 90%)

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

Is Accuracy always a good metric?

- Consider a problem of identifying fraudulent transactions
- An ML model classifies all transactions as ‘Not Fraud’
- Accuracy of the model would be very high as the number of fraudulent transactions would be very low
    - If 10 out of 1000 transactions are fraudulent, the above model would have a 99% accuracy!
- But the model **missed the main point** in this case
- **We need better metrics in such cases**

Precision

- A metric used to measure the accuracy of positive predictions made by a model
- **Represents the ability of the model to reduce the number of false positives**

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

- For example, marketing campaigns require a high precision value to ensure that a large number of potential customers will interact with their survey or be interested to learn more

Recall

- A metric that measures the proportion of actual positives that are correctly identified by the model
- **Represents the ability of the model to reduce the number of false negatives**

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

- For example, algorithms used to detect credit card fraud need to have a very high recall to ensure that the maximum number of fraudulent transactions are flagged, as missing even a single fraudulent transaction would have repercussions

F1 Score

- In some business scenarios, we might need a model that has both low false positives as well as low false negatives
- In such cases, F1 score can be used

$$\text{F1 Score} = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

- For example, when detecting spam emails, we want a model to have low false positives as well as low false negatives

Happy Learning !