Performance Metrics - Classification

Classification Metrics:

1. Accuracy
2. Confusion Metrics
3. Precision & Recall
4. F1 - Score
5. ROC - AUC
6. Log Loss

Accuracy:

Dn → Training → 10 datapoints → X_train, y_train → $\boxed{\text{Algo}}$ → Model

Dn → Test

X_train → Model → y_train_pred

y_train, y_train_pred → $\boxed{\text{Eval Metrics}}$ → train_score

y_train	y_train_pred
1	1
0	1
1	0
1	1
0	0
1	1
0	0
1	1
1	1
0	1

$$\text{Accuracy} = \frac{\text{Correct Prediction}}{\text{Total datapoints}}$$

$$\text{Accuracy} = \frac{?}{?}$$

Dn ⟶ Training ⟶ 1000 datapoints → X_train, ~~y_train~~ → $\boxed{\text{Algo}}$ → Model

900 +ve   100 -ve

We know X_train has 900 +ve's & 100 -ve's

Let's say our model has following predictions:

X_train → Model → y_train_pred

Act		Pred
900 +ves	→	850 +ves, 50 -ves
100 -ves	→	10 +ves, 90 -ves

$$\text{Accuracy} = \frac{?}{?}$$

Dn ⟶ Training ⟶ 1000 datapoints → X_train, ~~y_train~~ → $\boxed{\text{Algo}}$ → Model

900 +ve   100 -ve

We know X_train has 900 +ve's & 100 -ve's

Let's say our model has following predictions:

X_train → Model → y_train_pred

Act		Pred
900 +ves	→	900 +ves, 0 -ves
100 -ves	→	100 +ves, 0 -ves

$$\text{Accuracy} = \frac{?}{?}$$

Issues with ACCURACY Metrics:

Issues with ACCURACY Metrics :

If target variable is not uniformly distributed, it can give an incorrect interpretation about models performance.
↳ i.e IMBALANCED DATA

Confusion Matrix :

$y_i \in \{0, 1\}$ OR $y_i \in \{-ve, +ve\}$

Act →

Pred ↓	1	0
1	TP	FP
0	FN	TN

$P = TP + FN$

$N = FP + TN$

Is your prediction correct? — What is predicted?

↓ T/F — ↓ P/N

Calculating Rates :

Sensitivity

$TPR = \frac{TP}{P}$

$FNR = \frac{FN}{P}$

$FPR = \frac{FP}{N}$

$TNR = \frac{TN}{N}$

Specificity

```python
cm = metrics.confusion_matrix(y_test, y_test_pred)

print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[13  0  0]
 [ 0 15  1]
 [ 0  0  9]]
```

Act →

Pred ↓	0	1
0	TNR	FNR
1	FPR	TPR

→ Much better for interpretation than having counts.

```python
actual = np.sum(cm, axis=1).reshape(-1, 1)

cmn = np.round(cm/actual, 2)

print("Normalized Confusion Matrix:")
print(cmn)
```

```
Normalized Confusion Matrix:
[[1.   0.   0.  ]
 [0.   0.94 0.06]
 [0.   0.   1.  ]]
```

Precision & Recall :

For +ve class :

Precision → Out of total +ve Prediction, how many are correctly predicted +ve.

Act →

Pred ↓	1	0
1	TP	FP

Precision → Out of total +ve Prediction, how many are correctly predicted +ve.

Recall → Out of total +ve Actuals, how many are correctly predicted +ve.

Pred ↓ / Act →	1	0
1	TP	FP
0	FN	TN

$$\boxed{\underset{\text{(on +ve class)}}{\text{Precision}} = \frac{TP}{TP+FP} \quad \& \quad \underset{\text{(on +ve class)}}{\text{Recall}} = \frac{TP}{TP+FN}}$$

Recall → Sensitivity

For -ve class ↓

Precision → Out of total -ve Prediction, how many are correctly predicted -ve.

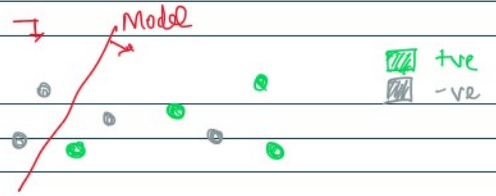
Recall → Out of total -ve Actuals, how many are correctly predicted -ve.

Pred ↓ / Act →	1	0
1	TP	FP
0	FN	TN

$$\boxed{\underset{\text{(on -ve class)}}{\text{Precision}} = \frac{TN}{TN+FN} \quad \& \quad \underset{\text{(on -ve class)}}{\text{Recall}} = \frac{TN}{TN+FP}}$$

Recall → Specificity

Example ↓



Model's Prec & Rec Calculation:

	Precision	Recall
+ve	4/6	4/4
-ve	2/2	2/4

Applications →

(a) Cancer Prediction ↓

Understand that in case of cancer prediction, it is very important to have least False Negative. A false negative case means that a cancer patient is assessed to not have the disease.

Similarly false Positive means a healthy patient diagnosed as cancer patient.

In most high risk disease detection cases (like COVID), recall is more important evaluation metric than precision.

Actual	Malignant	Benign
Pred	Benign	Malignant

Benign → False Negative

Malignant → +1
Benign → -1

(b) Spam Detection ↓

Important to minimize False Positive. Precision is more important.

Actual	Spam	Ham
Pred	Ham	Spam

Spam → +1
Ham → -1

Actual	Spam	Ham
Pred	Ham	Spam

False Positive

Spam → +1
Ham → −1

CASES →

A: Precision ↑ Recall ↑ Good Model

B: Precision ↓ Recall ↓ Bad Model

C: Precision ↑ Recall ↓

D: Precision ↓ Recall ↑

} Depends on the case study.

Examples of Case C : Spam Detection, Recommendation System, etc...

Examples of Case D : Cancer Detection, COVID Detection, Fraud Detection, etc...

F1 Score ⇁ $0 \leq F1\text{-}Score \leq 1$

F1 score is calculated by computing Harmonic Mean of Precision & Recall

$$F1\text{-}score = \left(\frac{(Precision)^{-1} + (Recall)^{-1}}{2}\right)^{-1}$$

$$\boxed{F1\text{-}score = \frac{2 * Precision * Recall}{Precision + Recall}}$$

ROC Curve & AUC ⇁ (Receiver Operating Characteristic Curve & Area under the curve)

Log-Loss ⇁ $0 \leq log\text{-}loss \leq +\infty$

$$Log\text{-}Loss = \sum_{i=1}^{n} -\left(y_i \log(\sigma(w^T x_i + w_0)) - ((1-y_i)\log(1-\sigma(w^T x_i + w_0))\right)$$