Class Imbalance Problem

BY

DR. ANUPAM GHOSH

What is the Class Imbalance Problem?

- It is the problem in machine learning where the total number of a class of data (positive) is far less than the total number of another class of data (negative).
- This problem is extremely common in practice and can be observed in various disciplines including fraud detection, anomaly detection, medical diagnosis, oil spillage detection, facial recognition
- Predictive accuracy works fine, when the classes are balanced; That is, every class in the data set are equally important. In fact, data sets with imbalanced class distributions are quite common in many real life applications
- When the classifier classified a test data set with imbalanced class distributions then, predictive accuracy on its own is not a reliable indicator of a classifier's effectiveness.
- Thus, when the classifier classified a test data set with imbalanced class distributions, then predictive accuracy on its own is not a reliable indicator of a classifier's effectiveness. This necessitates an alternative metrics to judge the classifier.

- A confusion matrix for a two classes (+, -) is shown below.

	C1	C2
C1	True positive	False negative
C2	False positive	True negative

	+	-
+	++	+-
-	-+	--

- There are four quadrants in the confusion matrix, which are symbolized as below.
  - True Positive (TP: f_{++}) : The number of instances that were positive (+) and correctly classified as positive (+v).
  - False Negative (FN: f_{+-}): The number of instances that were positive (+) and incorrectly classified as negative (-). It is also known as **Type 2 Error**.
  - False Positive (FP: f_{-+}): The number of instances that were negative (-) and incorrectly classified as (+). This also known as **Type 1 Error**.
  - True Negative (TN: f_{--}): The number of instances that were negative (-) and correctly classified as (-).

- N_p = TP (f_{++}) + FN (f_{+-})

  = is the total number of positive instances.

- $\mathrm{N_n}$ = FP(f_{-+}) + Tn(f_{--})

  = is the total number of negative instances.

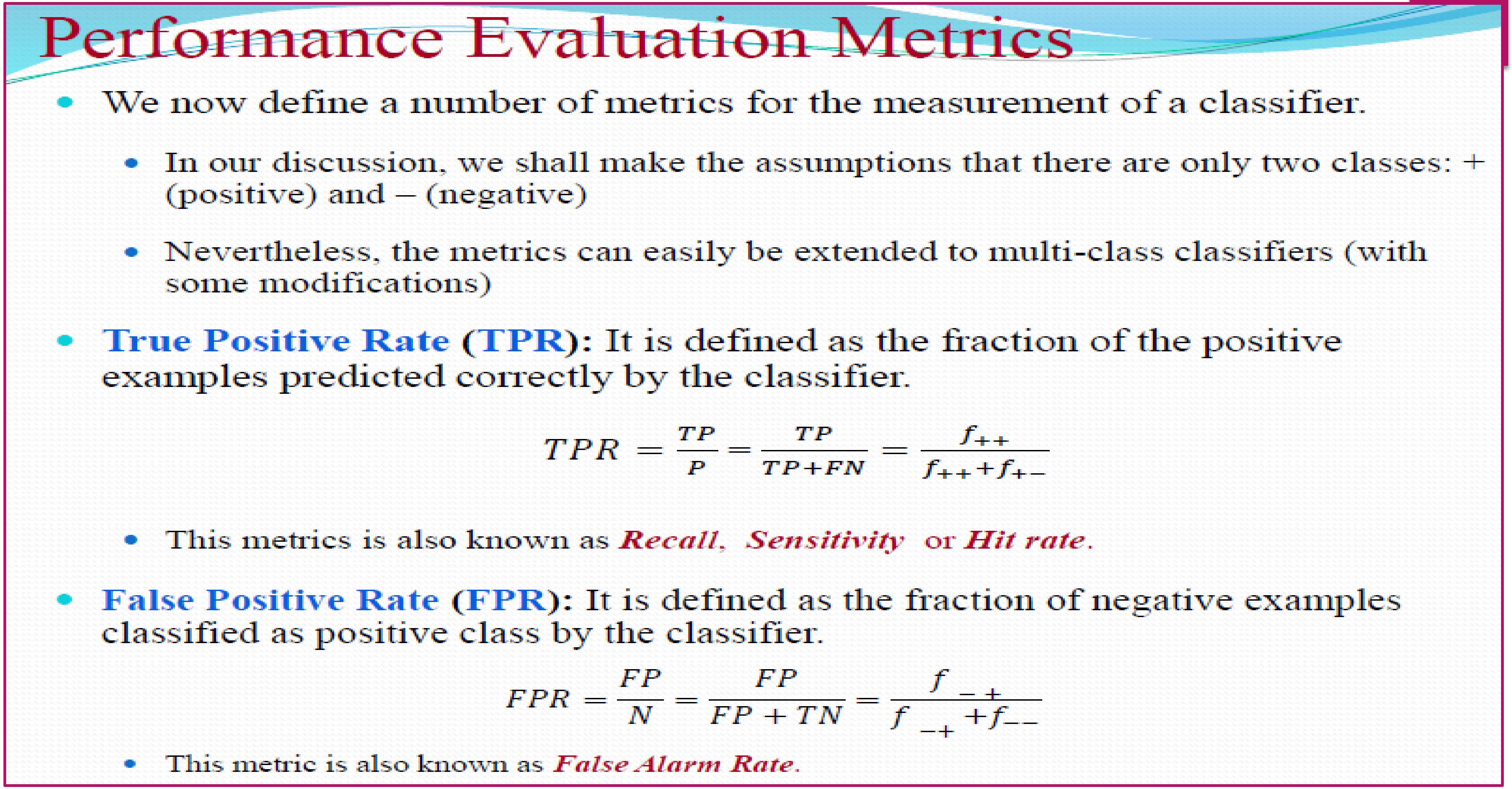
- $N = N_p + N_n$

  = is the total number of instances.

- (TP + TN) denotes the number of correct classification

- (FP + FN) denotes the number of errors in classification.

- For a perfect classifier FP = FN = 0, that is, there would be no Type 1 or Type 2 errors.



Performance Evaluation Metrics

- We now define a number of metrics for the measurement of a classifier.
  - In our discussion, we shall make the assumptions that there are only two classes: + (positive) and – (negative)
  - Nevertheless, the metrics can easily be extended to multi-class classifiers (with some modifications)
- **True Positive Rate (TPR):** It is defined as the fraction of the positive examples predicted correctly by the classifier.

$$TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = \frac{f_{++}}{f_{++}+f_{+-}}$$

  - This metrics is also known as ***Recall***, ***Sensitivity*** or ***Hit rate***.
- **False Positive Rate (FPR):** It is defined as the fraction of negative examples classified as positive class by the classifier.

$$FPR = \frac{FP}{N} = \frac{FP}{FP+TN} = \frac{f_{-+}}{f_{-+}+f_{--}}$$

  - This metric is also known as ***False Alarm Rate***.

- **Positive Predictive Value (PPV):** It is defined as the fraction of the positive examples classified as positive that are really positive

$$PPV = \frac{TP}{TP + FP} = \frac{f_{++}}{f_{++} + f_{-+}}$$

  - It is also known as *Precision*.

- **F$_1$ Score (F$_1$):** Recall (r) and Precision (p) are two widely used metrics employed in analysis, where detection of one of the classes is considered more significant than the others.

  - It is defined in terms of (r or TPR) and (p or PPV) as follows.

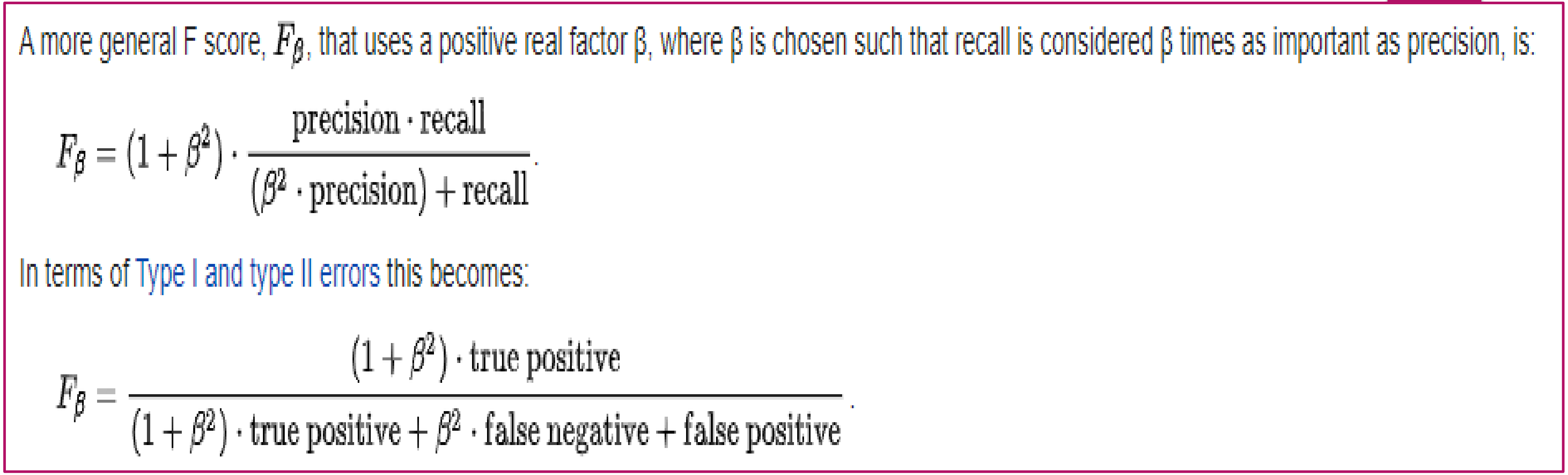
$$F_1 = \frac{2r.p}{r+p} = \frac{2TP}{2TP + FP + FN}$$

$$= \frac{2f_{++}}{2f_{++} + f_{\mp} + f_{+-}} = \frac{2}{\frac{1}{r} + \frac{1}{p}}$$

Note

- F$_1$ represents the harmonic mean between recall and precision

- High value of F$_1$ score ensures that both Precision and Recall are reasonably high.



A more general F score, F_β, that uses a positive real factor β, where β is chosen such that recall is considered β times as important as precision, is:

$$F_\beta = (1+\beta^2)\cdot\frac{\mathrm{precision}\cdot\mathrm{recall}}{(\beta^2\cdot\mathrm{precision})+\mathrm{recall}}.$$

In terms of Type I and type II errors this becomes:

$$F_\beta = \frac{(1+\beta^2)\cdot\mathrm{true\ positive}}{(1+\beta^2)\cdot\mathrm{true\ positive}+\beta^2\cdot\mathrm{false\ negative}+\mathrm{false\ positive}}.$$

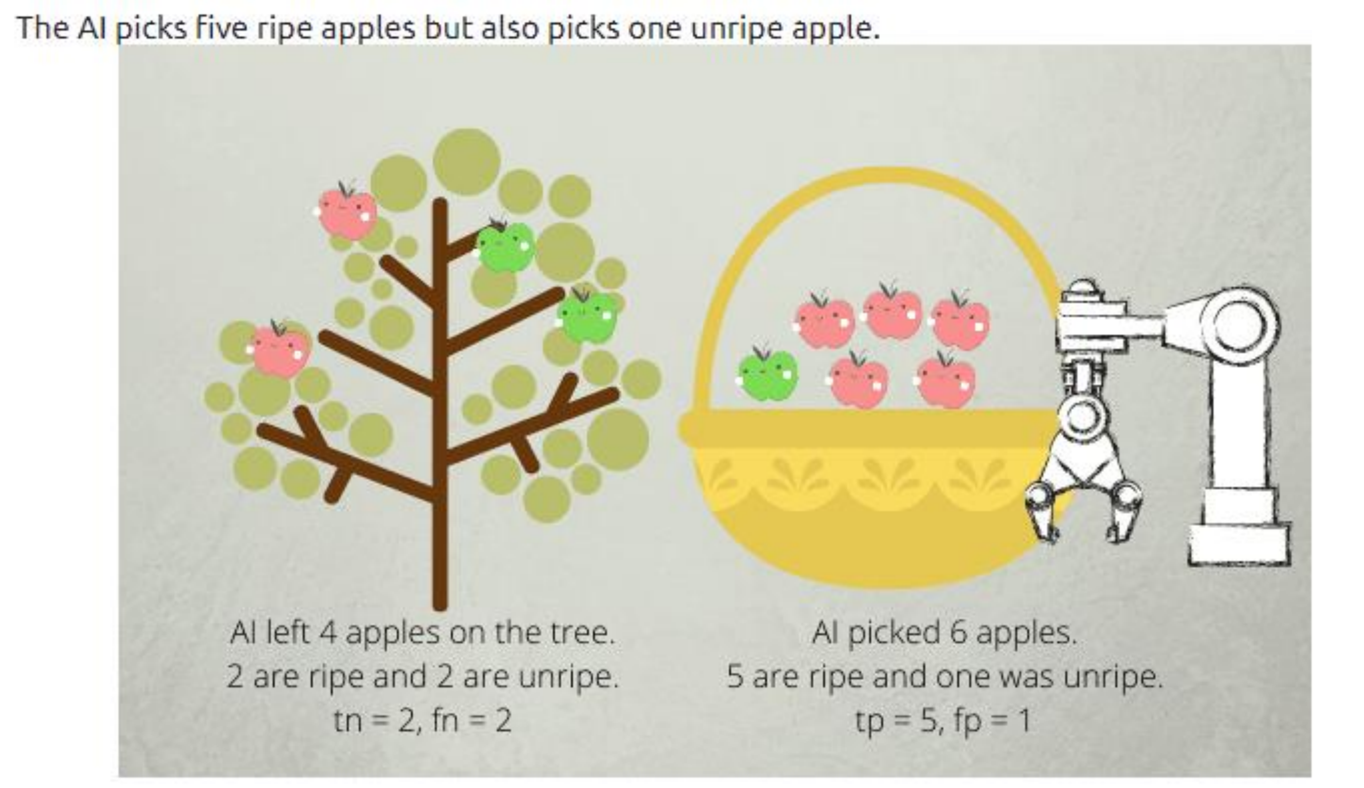
A factor indicating how much more important recall is than precision. For example, if we consider recall to be twice as important as precision, we can set β to 2. The standard F-score is equivalent to setting β to one.

Example

Example Calculation of F-score #1: Basic F-score

Let us imagine we have a tree with ten apples on it. Seven are ripe and three are still unripe, but we do not know which one is which. We have an AI which is trained to recognize which apples are ripe for picking, and pick all the ripe apples and no unripe apples. We would like to calculate the F-score, and we consider both precision and recall to be equally important, so we will set β to 1 and use the F1-score.

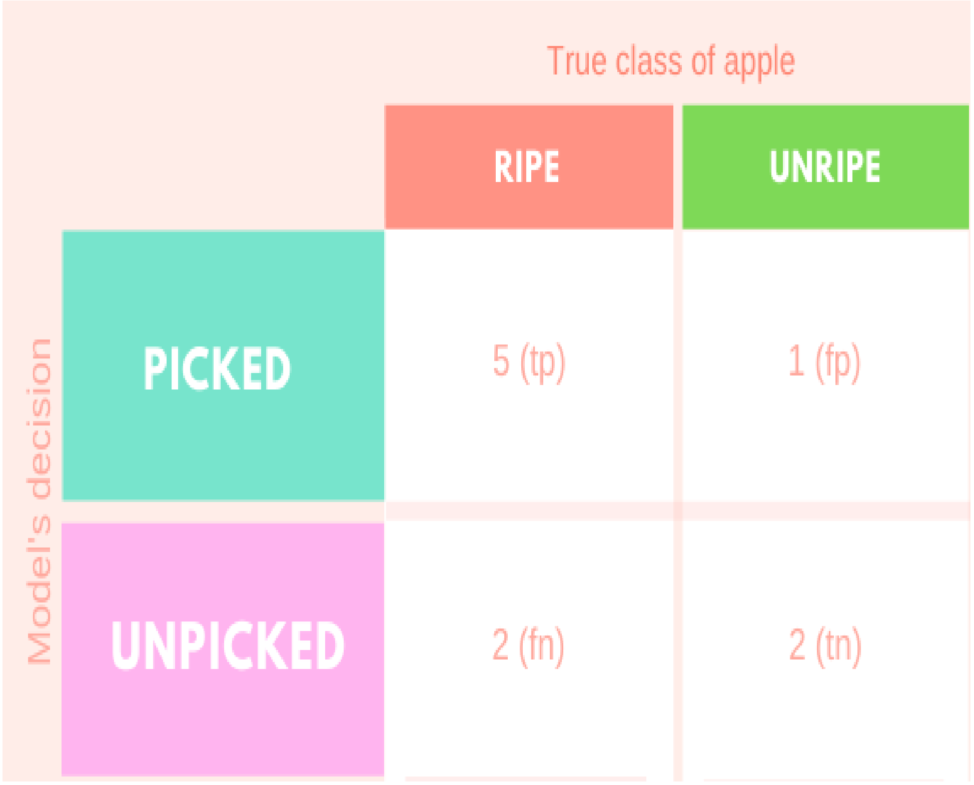
The AI picks five ripe apples but also picks one unripe apple.



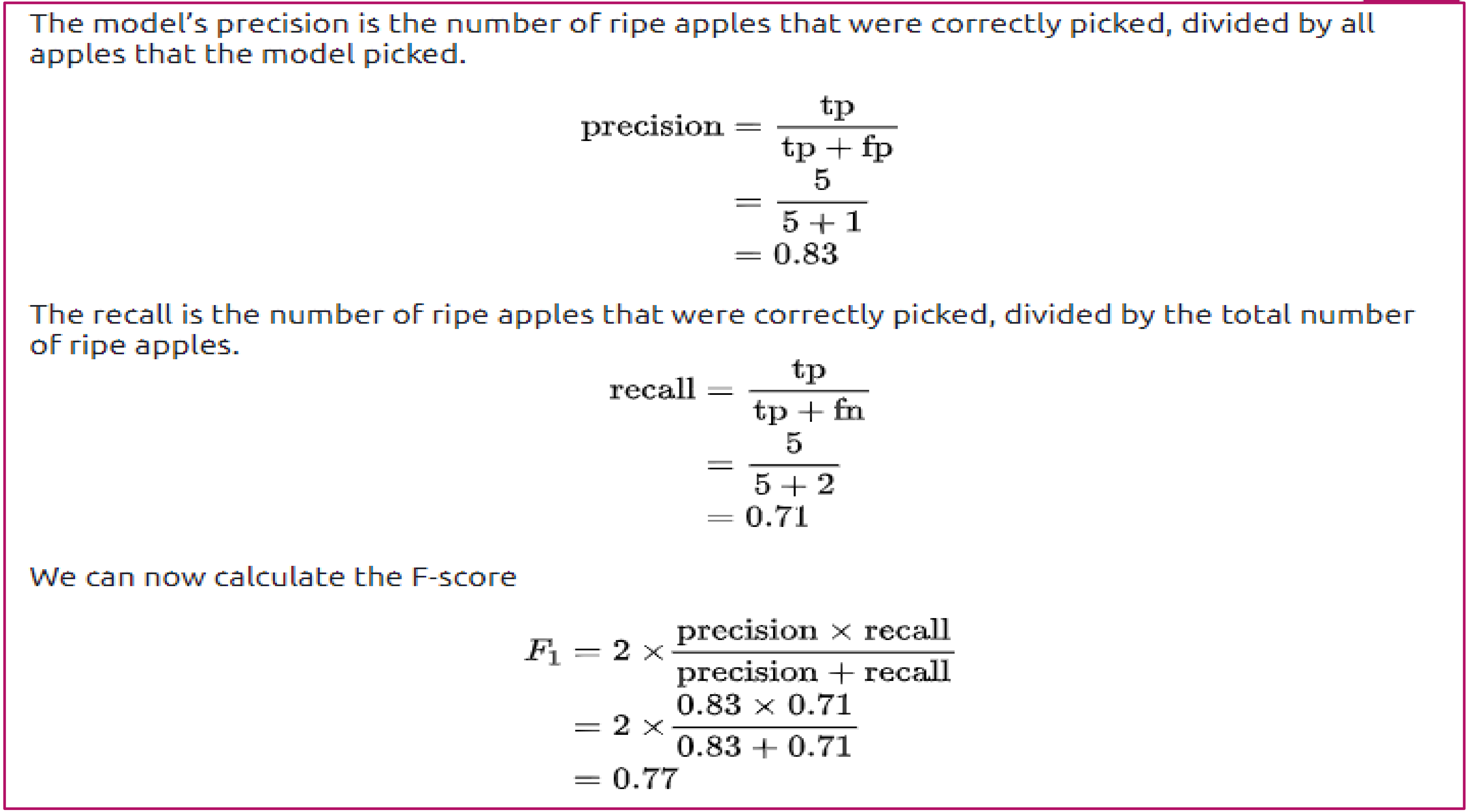
The AI picks five ripe apples but also picks one unripe apple.

AI left 4 apples on the tree.
2 are ripe and 2 are unripe.
tn = 2, fn = 2

AI picked 6 apples.
5 are ripe and one was unripe.
tp = 5, fp = 1



		True class of apple	
		RIPE	UNRIPE
Model's decision	PICKED	5 (tp)	1 (fp)
	UNPICKED	2 (fn)	2 (tn)



The model's precision is the number of ripe apples that were correctly picked, divided by all apples that the model picked.

$$\begin{aligned}\text{precision} &= \frac{\text{tp}}{\text{tp} + \text{fp}} \\ &= \frac{5}{5+1} \\ &= 0.83\end{aligned}$$

The recall is the number of ripe apples that were correctly picked, divided by the total number of ripe apples.

$$\begin{aligned}\text{recall} &= \frac{\text{tp}}{\text{tp} + \text{fn}} \\ &= \frac{5}{5+2} \\ &= 0.71\end{aligned}$$

We can now calculate the F-score

$$\begin{aligned}F_1 &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \\ &= 2 \times \frac{0.83 \times 0.71}{0.83 + 0.71} \\ &= 0.77\end{aligned}$$

Predictive Accuracy (ε)

- It is defined as the fraction of the number of examples that are correctly classified by the classifier to the total number of instances.

$$\varepsilon = \frac{TP + TN}{P + N}$$

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

$$= \frac{f_{++} + f_{--}}{f_{++} + f_{+-} + f_{\mp} + f_{--}}$$

Error Rate ($\bar{\varepsilon}$)

- The error rate $\bar{\varepsilon}$ is defined as the fraction of the examples that are incorrectly classified.

$$\bar{\varepsilon} = \frac{FP + FN}{P + N}$$

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

$$= \frac{f_{+-} + f_{-+}}{f_{++} + f_{+-} + f_{-+} + f_{--}}$$

Note

$$\bar{\varepsilon} = 1 - \varepsilon.$$

Accuracy, Sensitivity and Specificity

- Predictive accuracy (ε) can be expressed in terms of sensitivity and specificity.
- We can write

$$\varepsilon = \frac{TP + TN}{TP + FP + FN + TN}$$

$$= \frac{TP + TN}{P + N}$$

$$\varepsilon = \frac{TP}{P} \times \frac{P}{P + N} + \frac{TN}{N} \times \frac{N}{P + N}$$

Thus, $\varepsilon = \text{Sensitivity} \times \frac{P}{P+N} + \text{Specificity} \times \frac{N}{P+N}$

Analysis with Performance Measurement Metrics

- Based on the various performance metrics, we can characterize a classifier.
- We do it in terms of TPR, FPR, Precision and Recall and Accuracy
- **Case 1: Perfect Classifier**

When every instance is correctly classified, it is called the perfect classifier. In this case, $TP = P$, $TN = N$ and CM is

$$TPR = \frac{P}{P} = 1$$

$$FPR = \frac{0}{N} = 0$$

$$Precision = \frac{P}{P} = 1$$

$$F_1\ Score = \frac{2\times 1}{1+1} = 1$$

$$Accuracy = \frac{P+N}{P+N} = 1$$

		Predicted Class	
		+	-
Actual class	+	P	0
	-	0	N

- **Case 2: Worst Classifier**

When every instance is wrongly classified, it is called the worst classifier. In this case, $TP = 0$, $TN = 0$ and the CM is

$$TPR = \frac{0}{P} = 0$$

$$FPR = \frac{N}{N} = 1$$

$$Precision = \frac{0}{N} = 0$$

$$F_1\ Score = \text{Not applicable}$$

$$\text{as } Recall + Precision = 0$$

$$\text{Accuracy} = \frac{0}{P+N} = 0$$

		Predicted Class	
		+	-
Actual class	+	0	P
	-	N	0

- **Case 3: Ultra-Liberal Classifier**

The classifier always predicts the + class correctly. Here, the False Negative (FN) and True Negative (TN) are zero. The CM is

$$TPR = \frac{P}{P} = 1$$

$$FPR = \frac{N}{N} = 1$$

$$Precision = \frac{P}{P+N}$$

$$F_1\ Score = \frac{2P}{2P+N}$$

$$\text{Accuracy} = \frac{P}{P+N} = 0$$

		Predicted Class	
		+	-
Actual class	+	P	0
	-	N	0

- **Case 4: Ultra-Conservative Classifier**

This classifier always predicts the - class correctly. Here, the False Negative (FN) and True Negative (TN) are zero. The CM is

$$TPR = \frac{0}{P} = 0$$

$$FPR = \frac{0}{N} = 0$$

$$Precision = \text{Not applicable}$$

$$(\text{as } TP + FP = 0)$$

$$F_1\ Score = \text{Not applicable}$$

$$\text{Accuracy} = \frac{N}{P+N} = 0$$

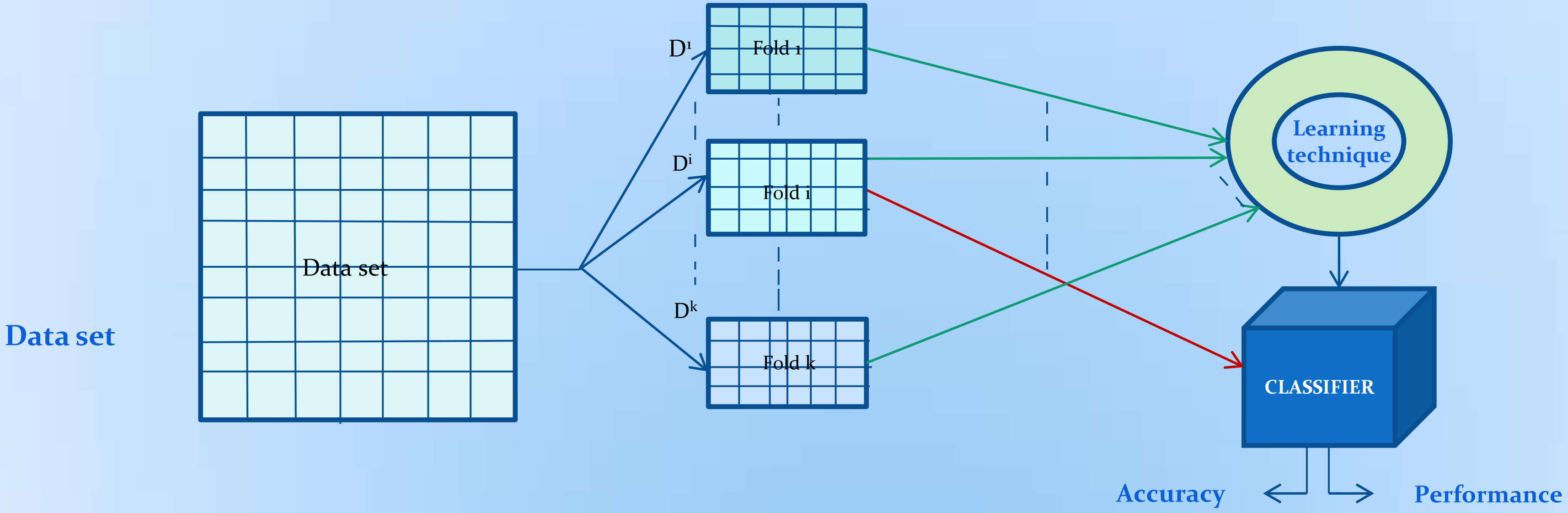
		Predicted Class	
		+	-
Actual class	+	0	p
	-	0	N

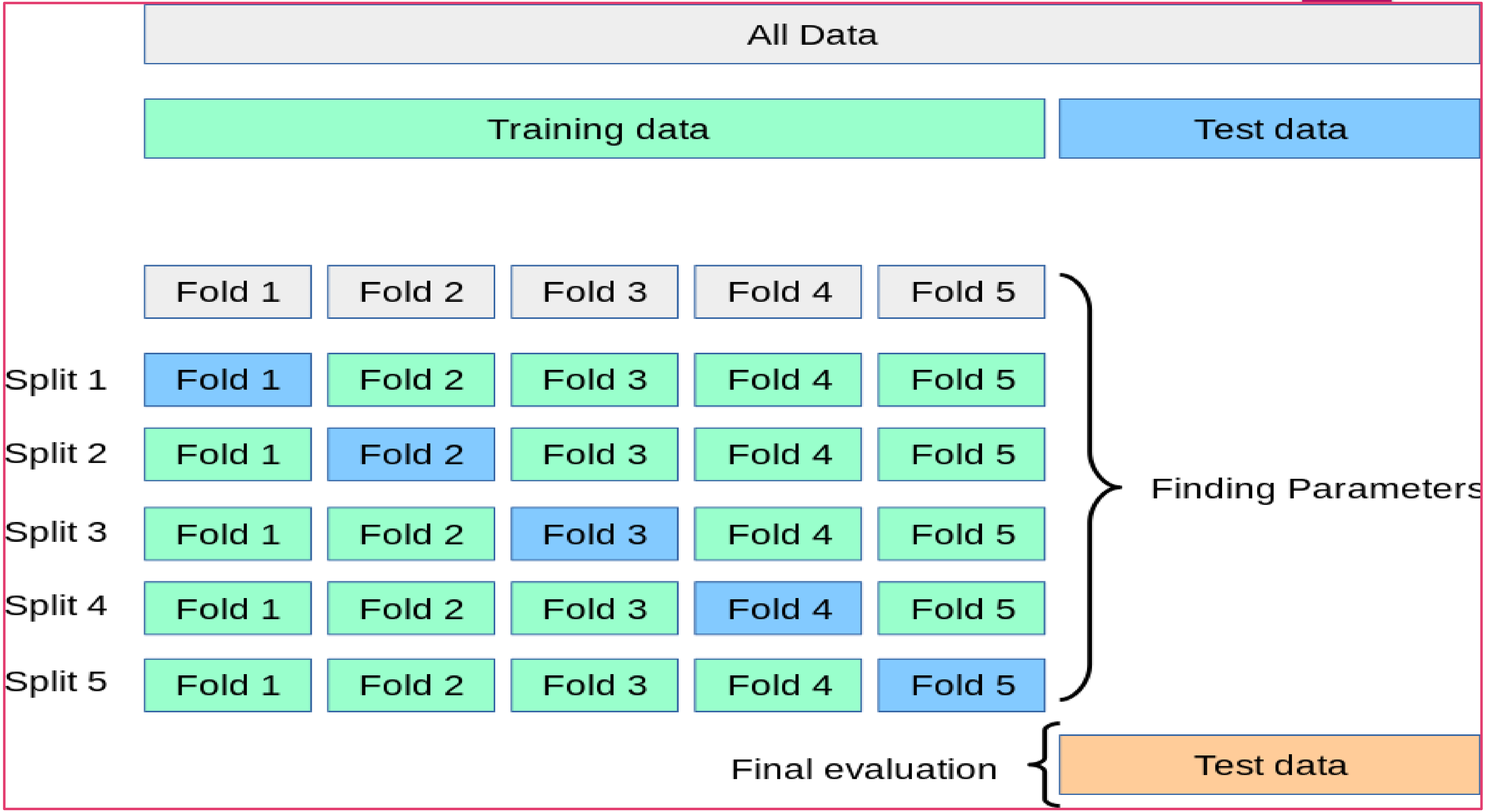
k-fold Cross-Validation

- Dataset consisting of *N* tuples is divided into *k* (usually, 5 or 10) equal, mutually exclusive parts or folds $(D_1, D_2, \ldots., D_k)$, and if *N* is not divisible by *k*, then the last part will have fewer tuples than other (*k*-1) parts.

- A series of *k* runs is carried out with this decomposition, and in *i*$^{\text{th}}$ iteration D_i is used as test data and other folds as training data
  - Thus, each tuple is used same number of times for training and once for testing.

- Overall estimate is taken as the average of estimates obtained from each iteration.





All Data

Training data

Test data

Fold 1 Fold 2 Fold 3 Fold 4 Fold 5

Split 1 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5

Split 2 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5

Split 3 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5

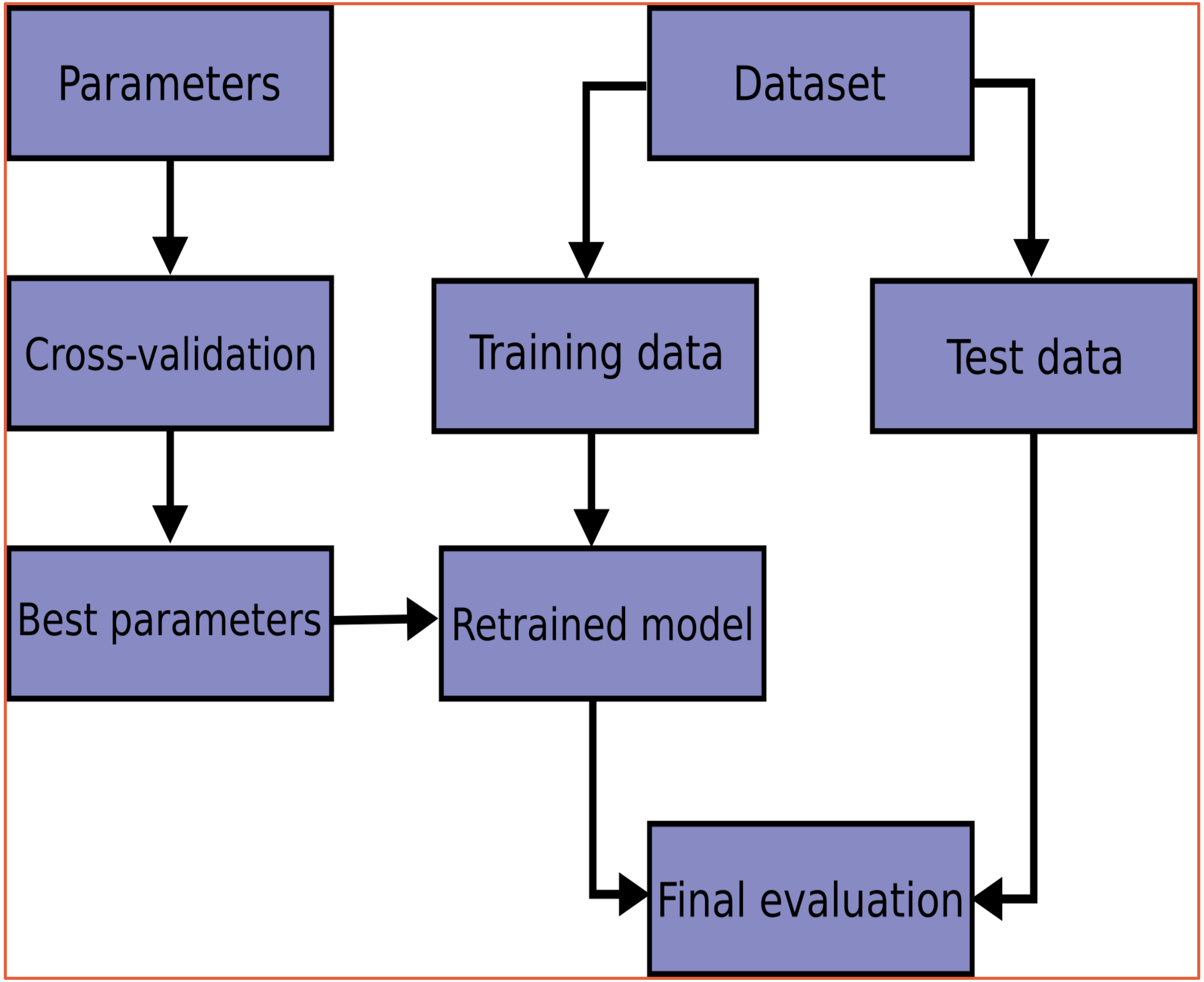
Split 4 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5

Split 5 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5

Finding Parameters

Final evaluation

Test data



A model is trained using of the folds as training data; the resulting model is validated on the remaining part of the data (i.e., it is used as a test set to compute a performance measure such as accuracy).

Statistical Estimation using Confidence Level

Example: True accuracy from observed accuracy

A classifier is tested with a test set of size 100. Classifier predicts 80 test tuples correctly. We are to calculate the following.

a) Observed accuracy

b) Mean error rate

c) Standard error

d) True accuracy with confidence level 0.95.

Solution:

a) The observed accuracy($\in$) = 80/100 = 0.80 So error (p) = 0.2

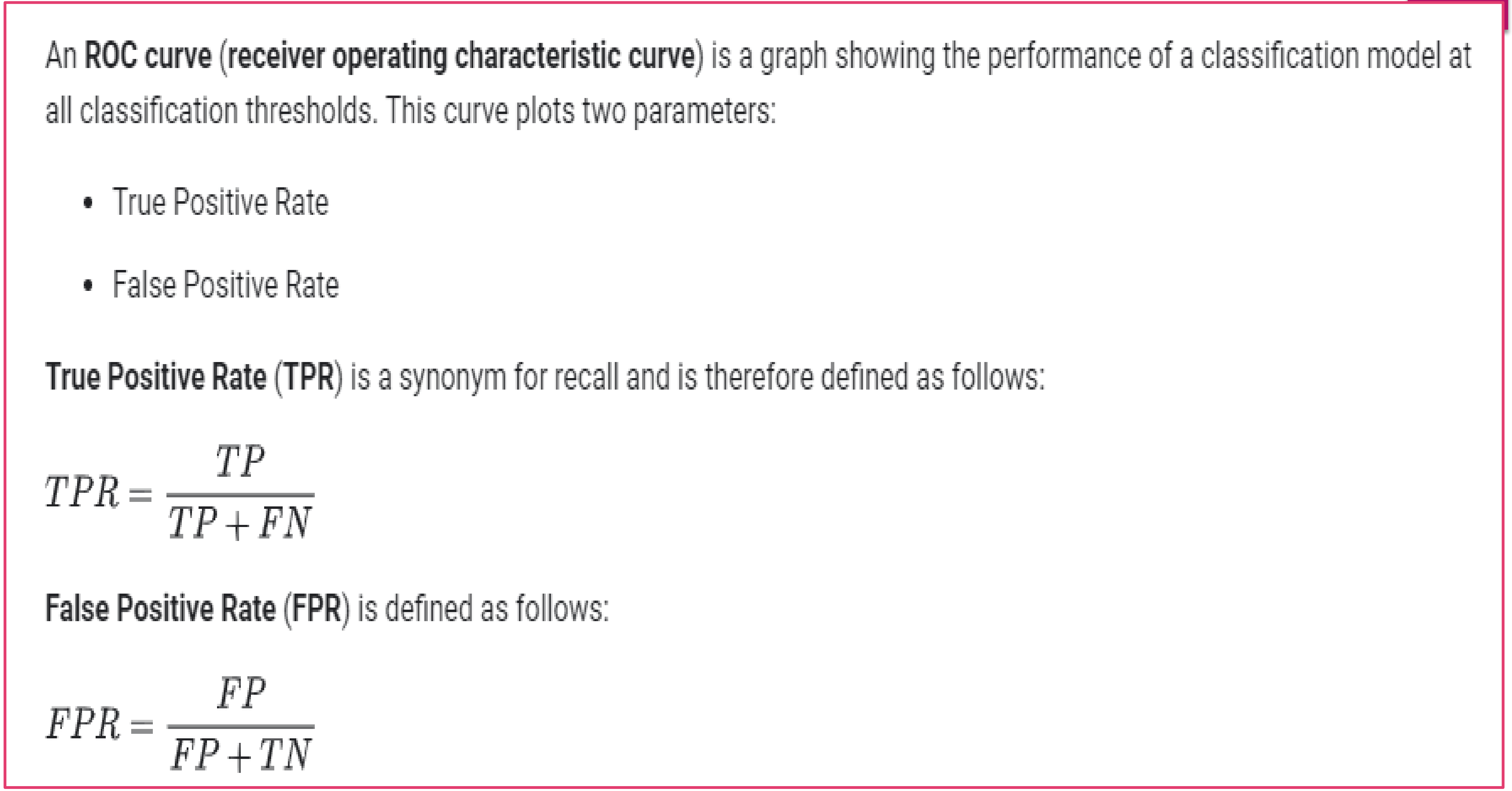
b) Mean error rate = $p \times N$ = 0.2×N = 20

c) Standard error rate (σ) = $\sqrt{\in (1 - \in)/N} = \sqrt{\frac{0.8 \times 0.2}{100}} = 0.04$

d) $\widetilde{\in} = \in \pm \tau_{\alpha} \times \sqrt{\in (1 - \in)/N}$ = 0.8±0.04×1.96 = 0.7216 with τ_{α}=1.96 and α = 0.95.

AUC-ROC curve

- The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the ‘signal’ from the ‘noise’. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.
- The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.



An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

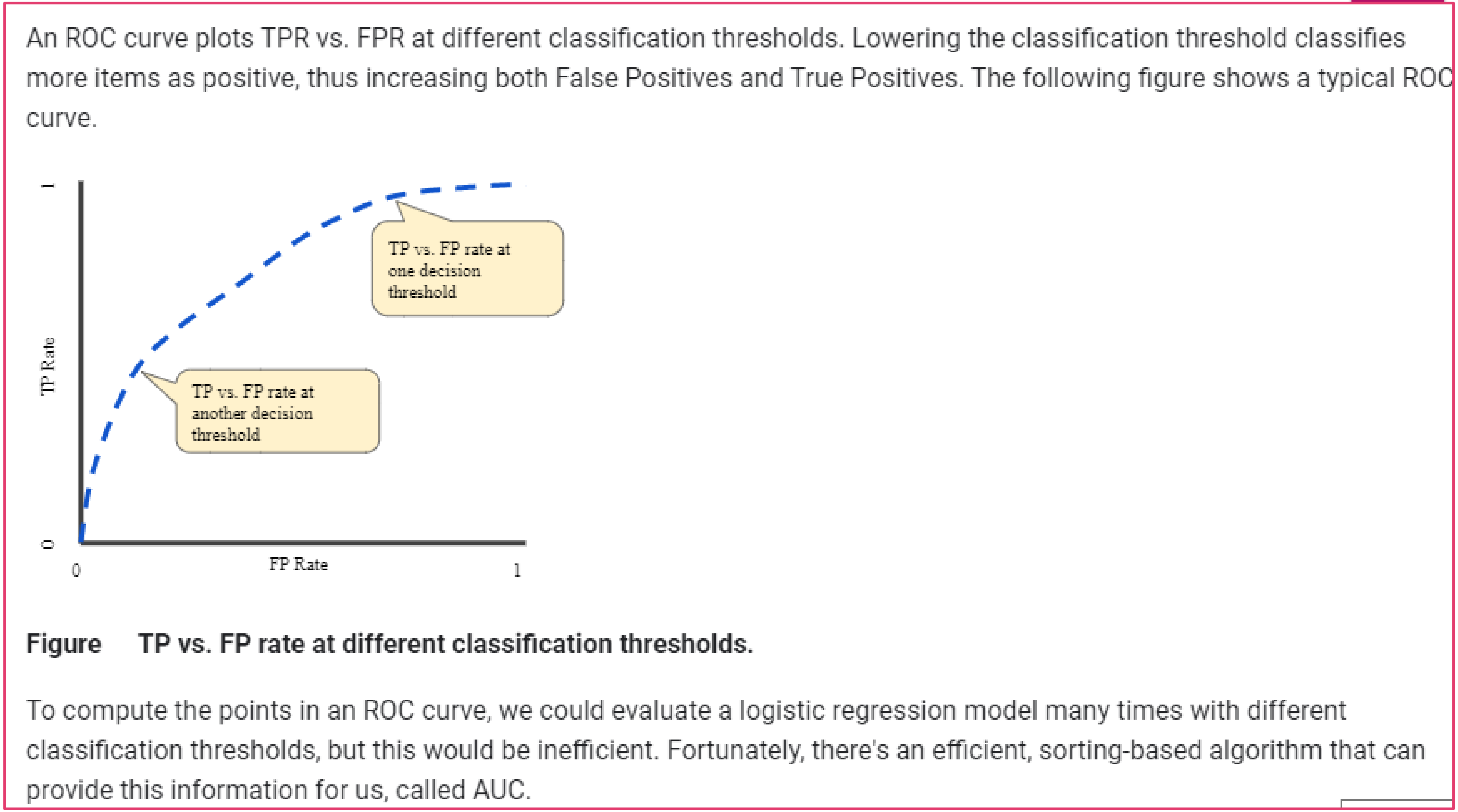
- True Positive Rate
- False Positive Rate

True Positive Rate (**TPR**) is a synonym for recall and is therefore defined as follows:

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

False Positive Rate (**FPR**) is defined as follows:

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

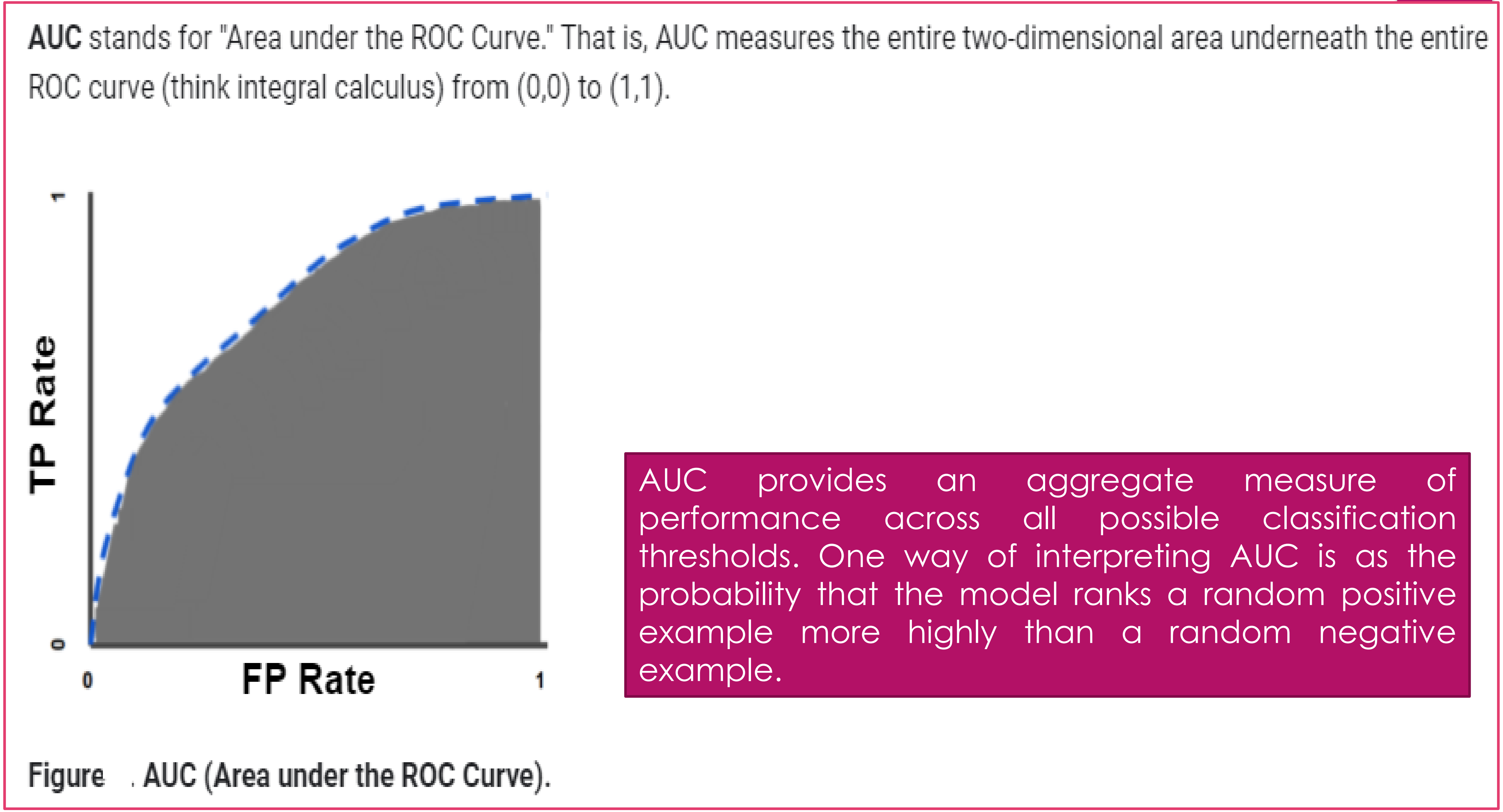


An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.

1
TP Rate
0
0
FP Rate
1
TP vs. FP rate at one decision threshold
TP vs. FP rate at another decision threshold

Figure TP vs. FP rate at different classification thresholds.

To compute the points in an ROC curve, we could evaluate a logistic regression model many times with different classification thresholds, but this would be inefficient. Fortunately, there's an efficient, sorting-based algorithm that can provide this information for us, called AUC.



AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).

Figure . AUC (Area under the ROC Curve).

AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example.

- When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives.

- When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values. This is so because the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives.

- When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is predicting random class or constant class for all the data points.

It will be continued