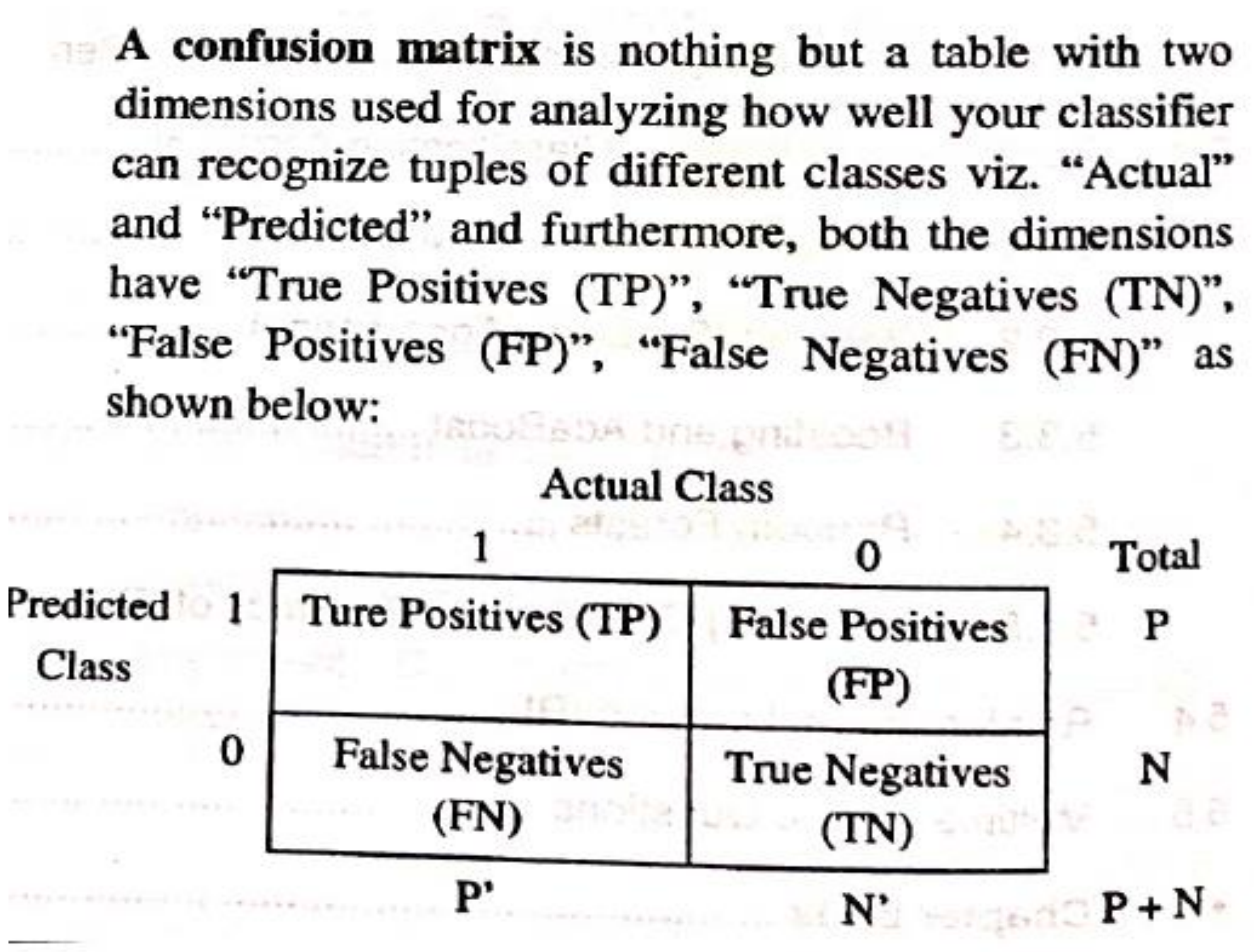
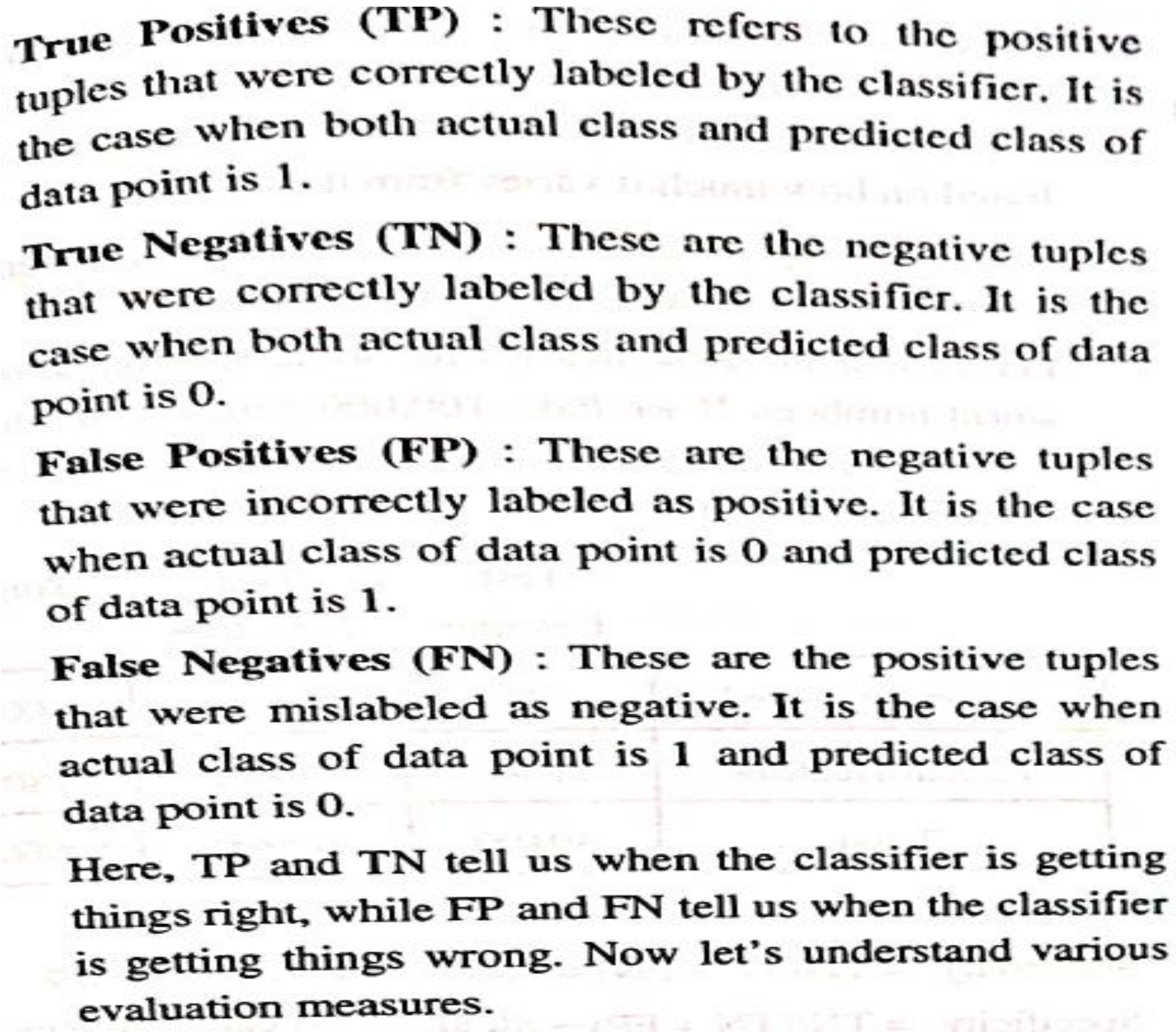


A confusion matrix is nothing but a table with two dimensions used for analyzing how well your classifier can recognize tuples of different classes viz. "Actual" and "Predicted" and furthermore, both the dimensions have "True Positives (TP)", "True Negatives (TN)", "False Positives (FP)", "False Negatives (FN)" as shown below:

		Actual Class		
		1	0	Total
Predicted Class	1	Ture Positives (TP)	False Positives (FP)	P
	0	False Negatives (FN)	True Negatives (TN)	N
		P'	N'	P + N



True Positives (TP) : These refers to the positive tuples that were correctly labeled by the classifier. It is the case when both actual class and predicted class of data point is 1.

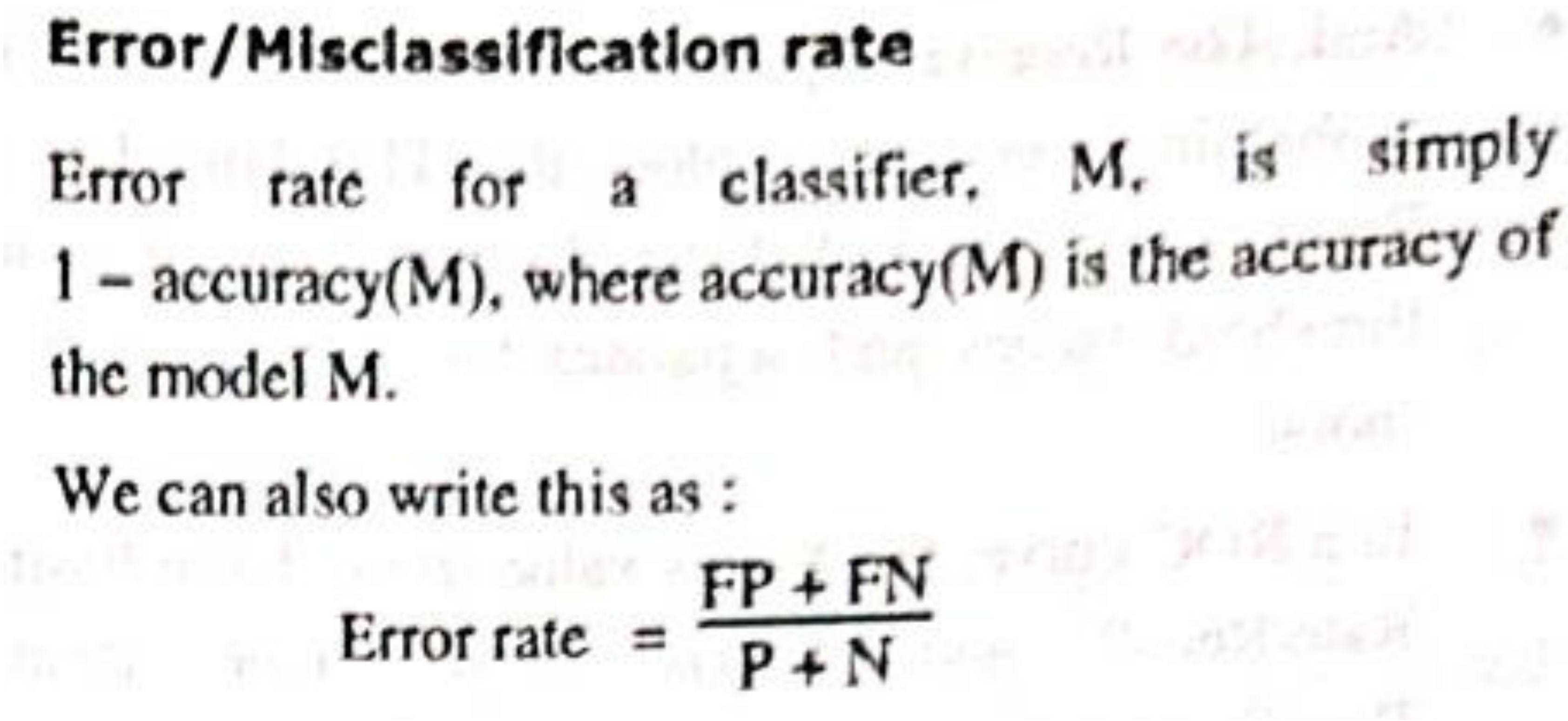
True Negatives (TN) : These are the negative tuples that were correctly labeled by the classifier. It is the case when both actual class and predicted class of data point is 0.

False Positives (FP) : These are the negative tuples that were incorrectly labeled as positive. It is the case when actual class of data point is 0 and predicted class of data point is 1.

False Negatives (FN) : These are the positive tuples that were mislabeled as negative. It is the case when actual class of data point is 1 and predicted class of data point is 0.

Here, TP and TN tell us when the classifier is getting things right, while FP and FN tell us when the classifier is getting things wrong. Now let's understand various evaluation measures.

- **There are 4 terms you should keep in mind:**
- **True Positives:** It is the case where we predicted Yes and the real output was also yes.
- **True Negatives:** It is the case where we predicted No and the real output was also No.
- **False Positives:** It is the case where we predicted Yes but it was actually No.
- **False Negatives:** It is the case where we predicted No but it was actually Yes.

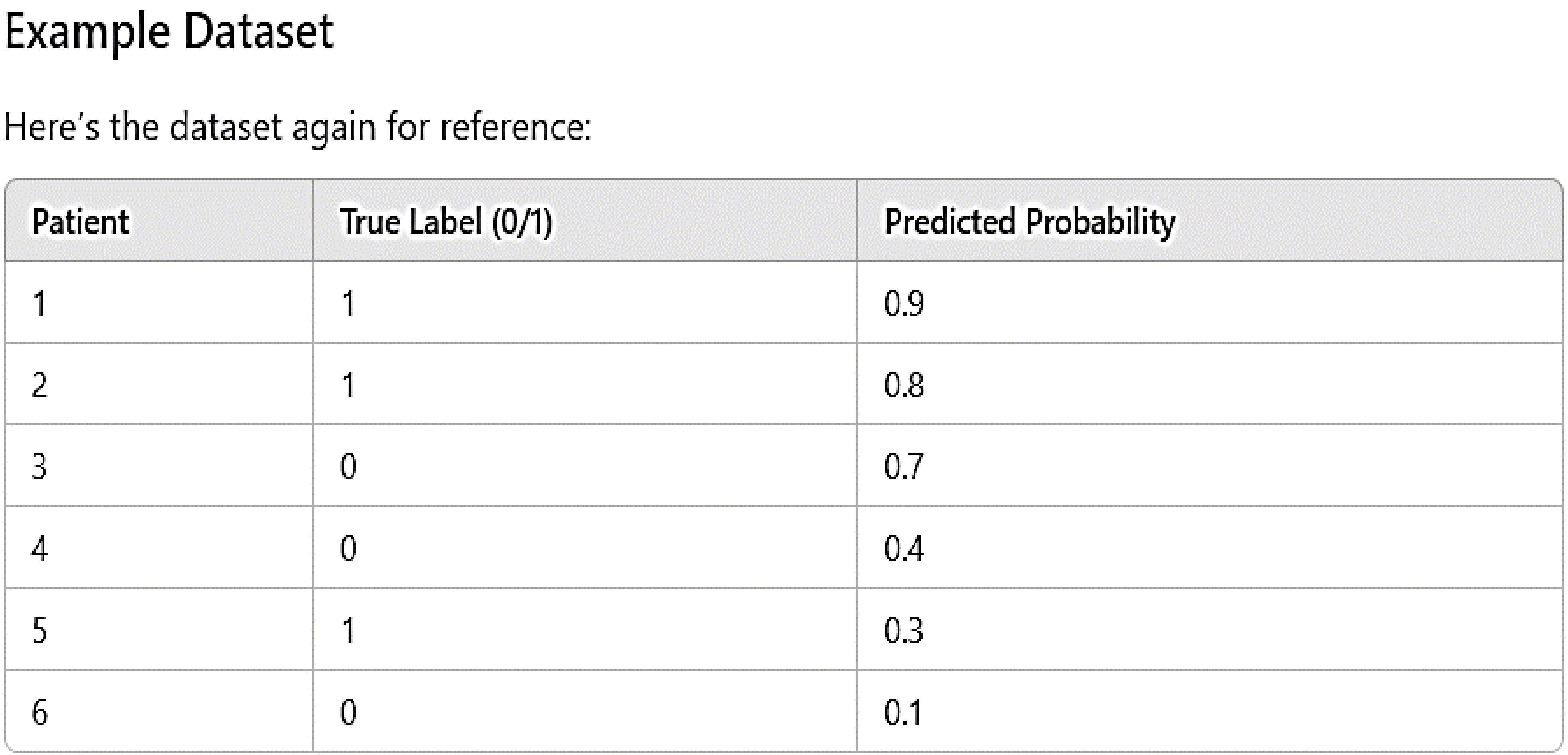


Error/Misclassification rate

Error rate for a classifier, M, is simply 1 – accuracy(M), where accuracy(M) is the accuracy of the model M.

We can also write this as :

$$\text{Error rate} = \frac{FP + FN}{P + N}$$

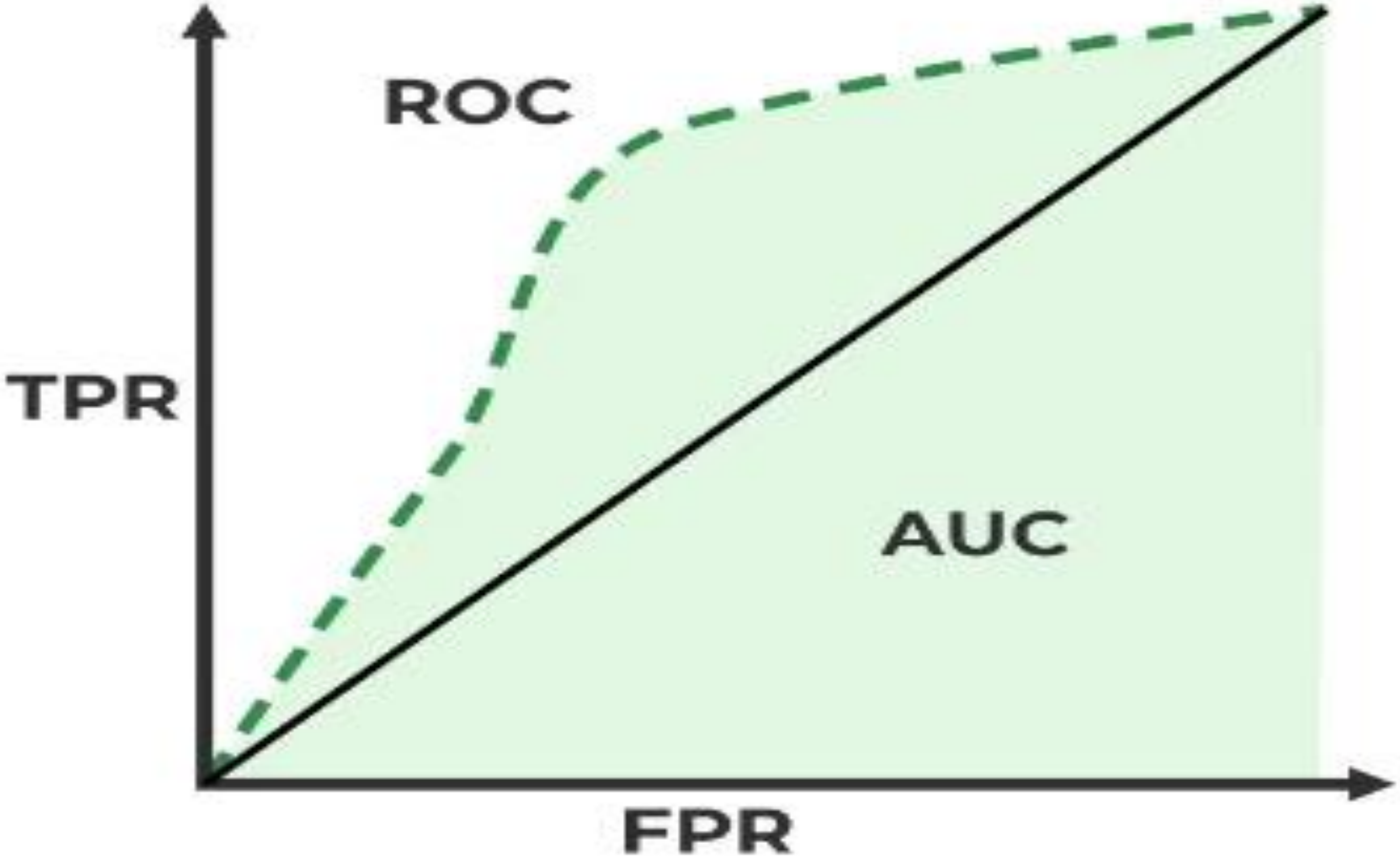


Example Dataset

Here’s the dataset again for reference:

Patient	True Label (0/1)	Predicted Probability
1	1	0.9
2	1	0.8
3	0	0.7
4	0	0.4
5	1	0.3
6	0	0.1

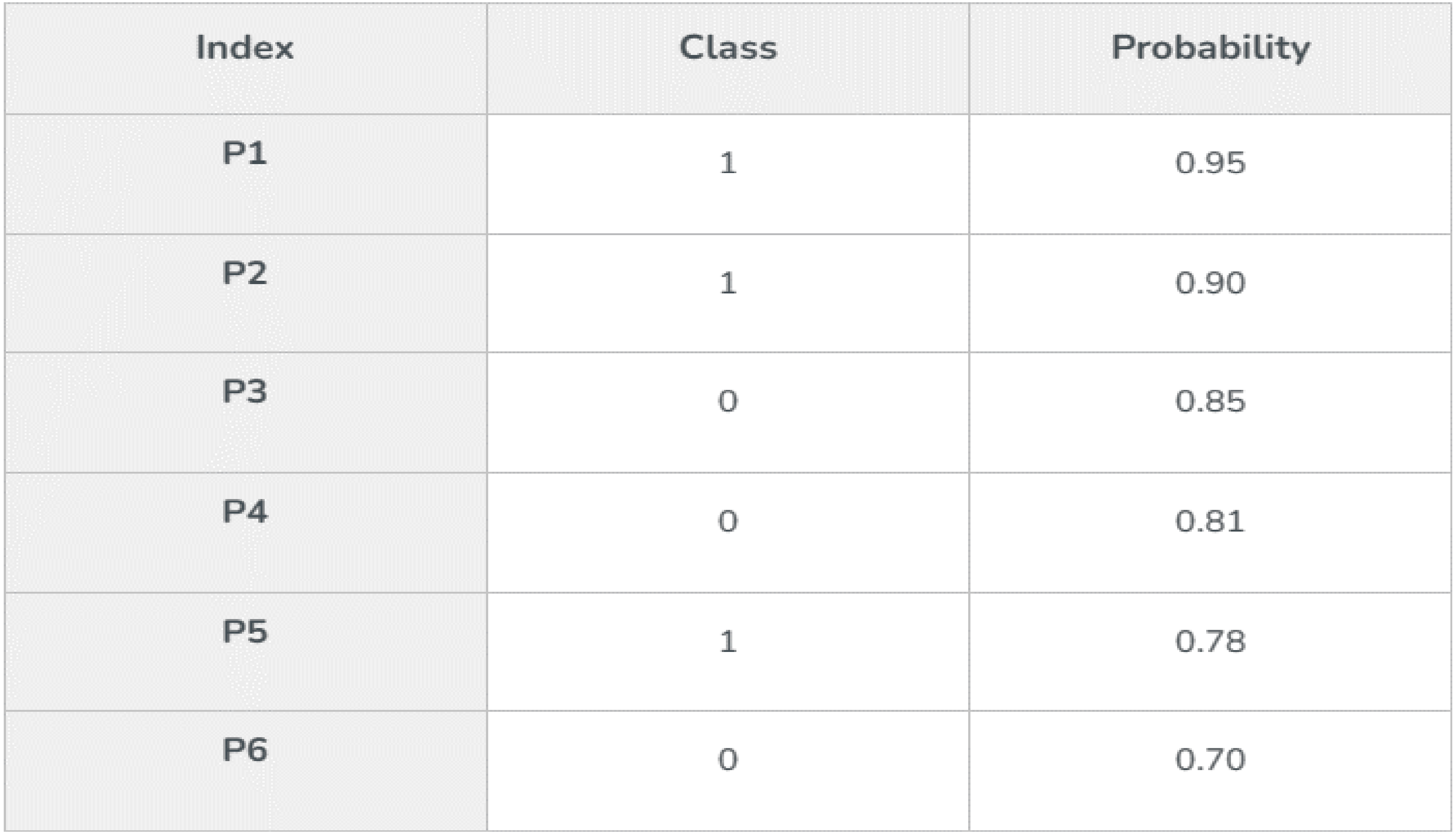
- **What is the AUC-ROC curve?**
- The AUC-ROC curve, or Area Under the Receiver Operating Characteristic curve, is a graphical representation of the performance of a binary classification model at various classification thresholds. It is commonly used in machine learning to assess the ability of a model to distinguish between two classes, typically the positive class (e.g., presence of a disease) and the negative class (e.g., absence of a disease).
- **Receiver Operating Characteristics (ROC) Curve**
- ROC stands for Receiver Operating Characteristics, and the ROC curve is the graphical representation of the effectiveness of the binary classification model. It plots the true positive rate (TPR) vs the false positive rate (FPR) at different classification thresholds.
- **Area Under Curve (AUC) Curve:**
- AUC stands for the Area Under the Curve, and the AUC curve represents the area under the ROC curve. It measures the overall performance of the binary classification model. As both TPR and FPR range between 0 to 1, So, the area will always lie between 0 and 1, and A greater value of AUC denotes better model performance. Our main goal is to maximize this area in order to have the highest TPR and lowest FPR at the given threshold. The AUC measures the probability that the model will assign a randomly chosen positive instance a higher predicted probability compared to a randomly chosen negative instance



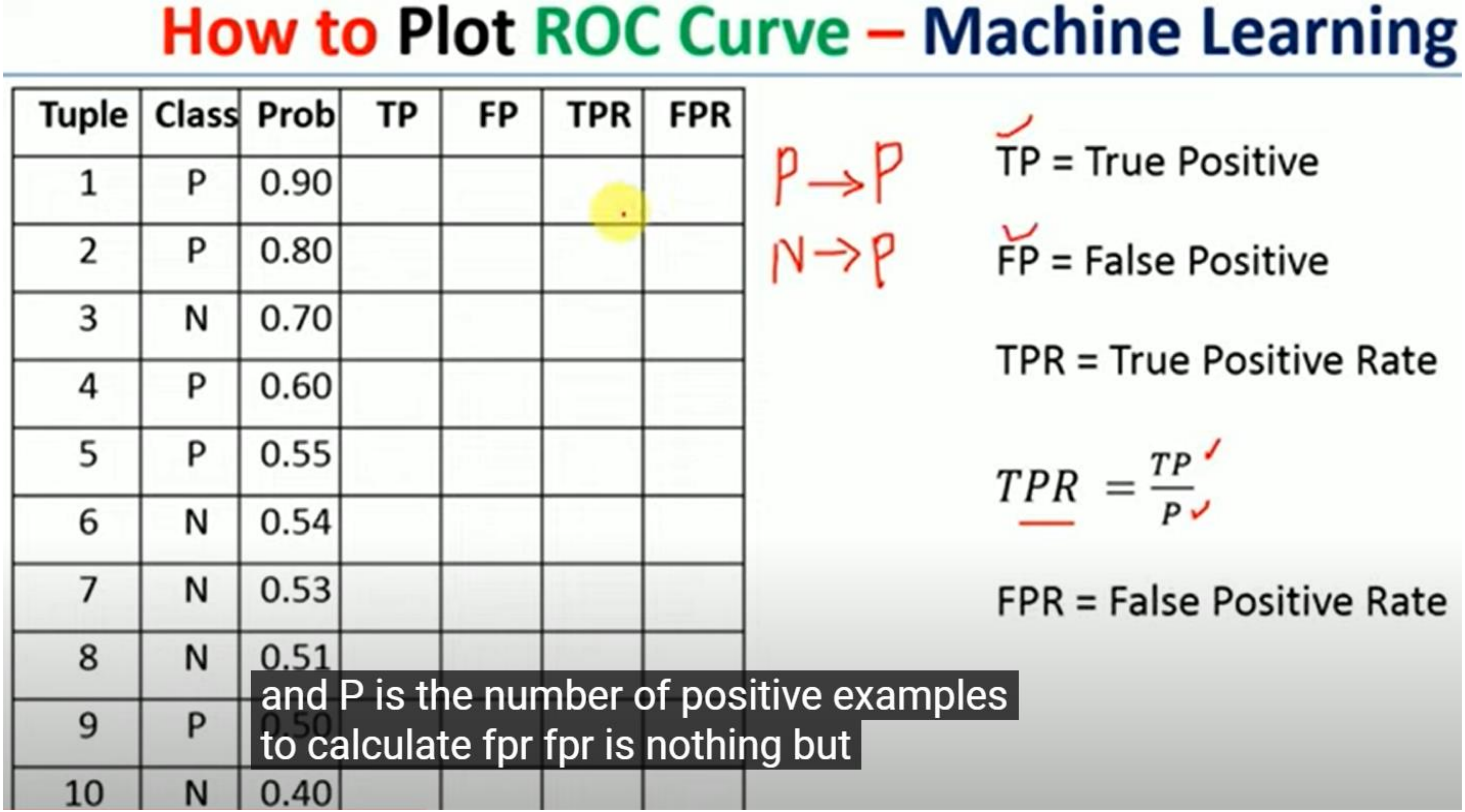
ROC-AUC Classification Evaluation Metric

- The gray dashed line represents the “Worst case” scenario, where the model’s predictions i.e TPR are FPR are same. This diagonal line is considered the worst-case scenario, indicating an equal likelihood of false positives and false negatives.
- As points deviate from the random guess line towards the upper-left corner, the model’s performance improves.
- The Area Under the Curve (AUC) is a quantitative measure of the model’s discriminative ability. A higher AUC value, closer to 1.0, indicates superior performance. The best possible AUC value is 1.0, corresponding to a model that achieves 100% sensitivity and 100% specificity.

- **Relationship between Sensitivity, Specificity, FPR, and Threshold.**
- **Sensitivity and Specificity:**
- **Inverse Relationship:** sensitivity and specificity have an inverse relationship. When one increases, the other tends to decrease. This reflects the inherent trade-off between true positive and true negative rates.
- **Tuning via Threshold:** By adjusting the threshold value, we can control the balance between sensitivity and specificity. Lower thresholds lead to higher sensitivity (more true positives) at the expense of specificity (more false positives). Conversely, raising the threshold boosts specificity (fewer false positives) but sacrifices sensitivity (more false negatives).
- **Threshold and False Positive Rate (FPR):**
- **FPR and Specificity Connection:** False Positive Rate (FPR) is simply the complement of specificity (FPR = 1 – specificity). This signifies the direct relationship between them: higher specificity translates to lower FPR, and vice versa.
- **FPR Changes with TPR:** Similarly, as you observed, the True Positive Rate (TPR) and FPR are also linked. An increase in TPR (more true positives) generally leads to a rise in FPR (more false positives). Conversely, a drop in TPR (fewer true positives) results in a decline in FPR (fewer false positives)



Index	Class	Probability
P1	1	0.95
P2	1	0.90
P3	0	0.85
P4	0	0.81
P5	1	0.78
P6	0	0.70



How to Plot ROC Curve – Machine Learning

Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.90				
2	P	0.80				
3	N	0.70				
4	P	0.60				
5	P	0.55				
6	N	0.54				
7	N	0.53				
8	N	0.51				
9	P	0.50				
10	N	0.40				

P → P

N → P

TP = True Positive

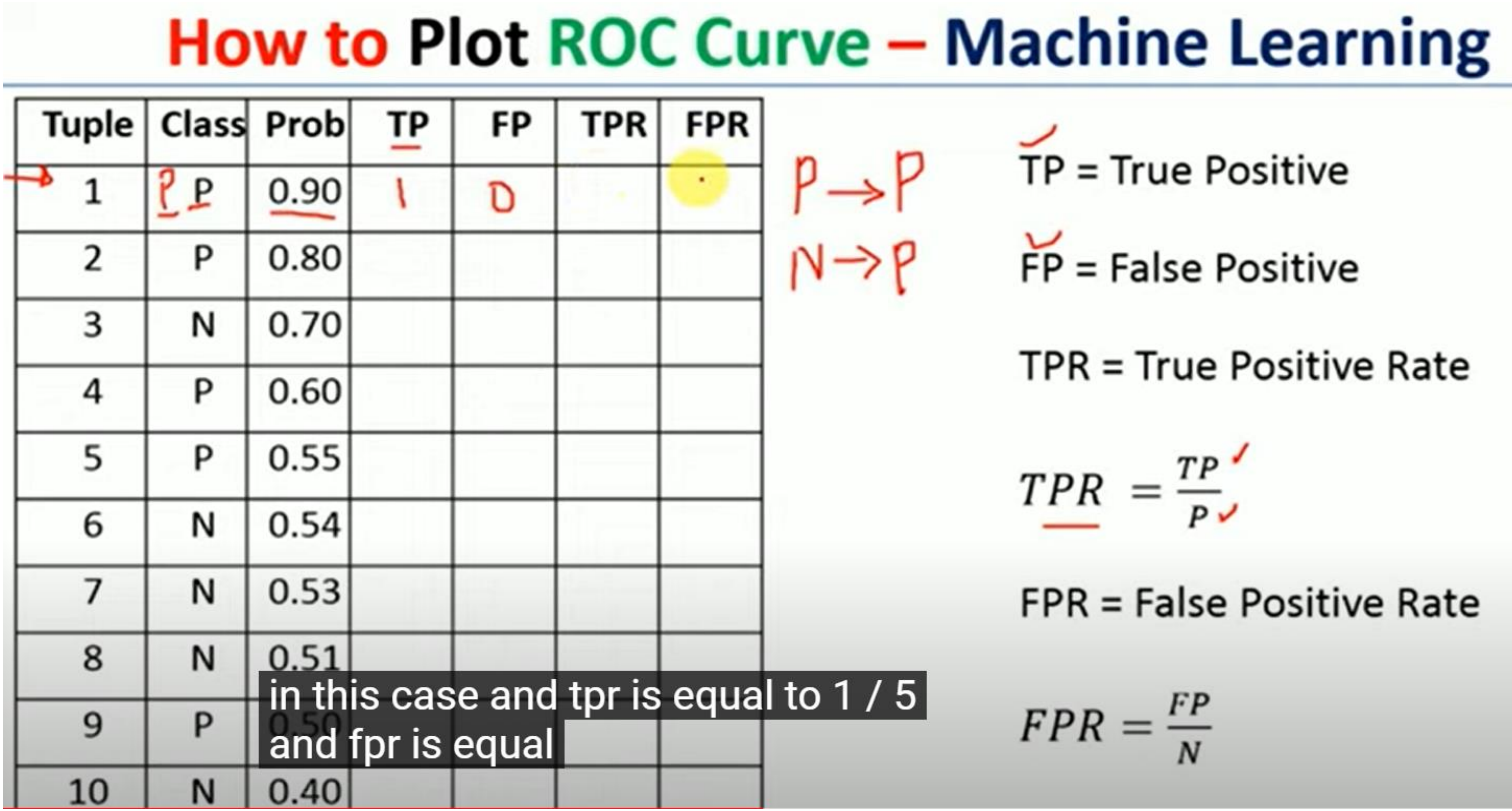
FP = False Positive

TPR = True Positive Rate

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

FPR = False Positive Rate

and P is the number of positive examples
to calculate fpr fpr is nothing but



How to Plot ROC Curve – Machine Learning

Tuple	Class	Prob	TP	FP	TPR	FPR
→ 1	P P	0.90	1	0		
2	P	0.80				
3	N	0.70				
4	P	0.60				
5	P	0.55				
6	N	0.54				
7	N	0.53				
8	N	0.51				
9	P	0.50				
10	N	0.40				

$P \rightarrow P$

$N \rightarrow P$

✓ TP = True Positive

✓ FP = False Positive

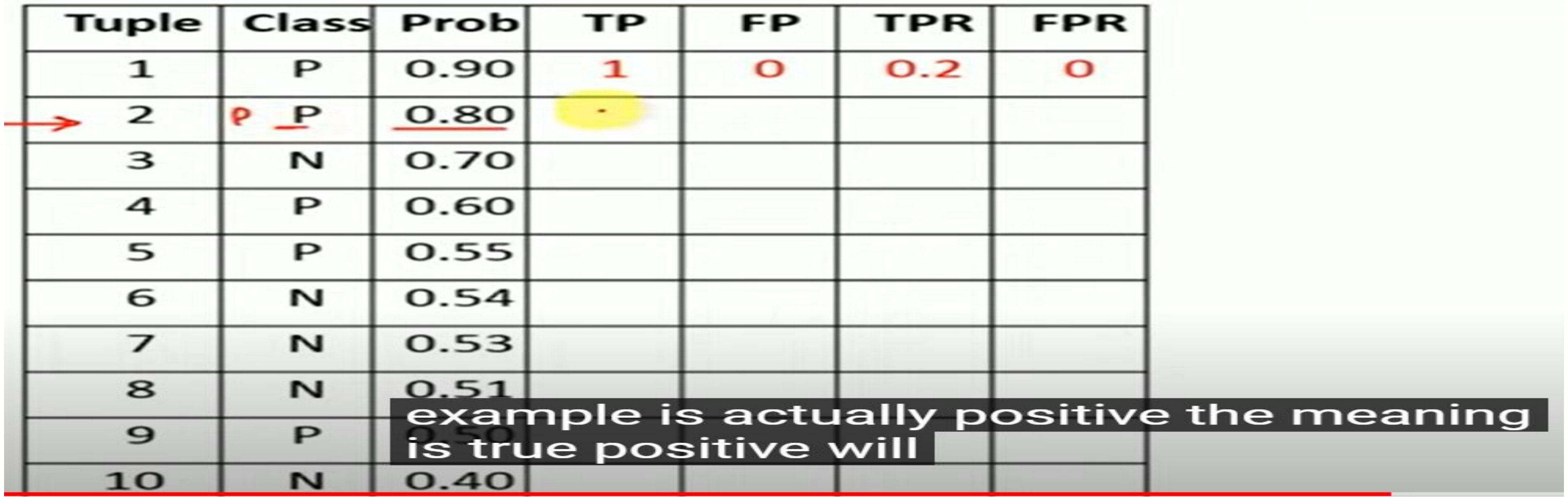
TPR = True Positive Rate

$$\underline{TPR} = \frac{TP✓}{P✓}$$

FPR = False Positive Rate

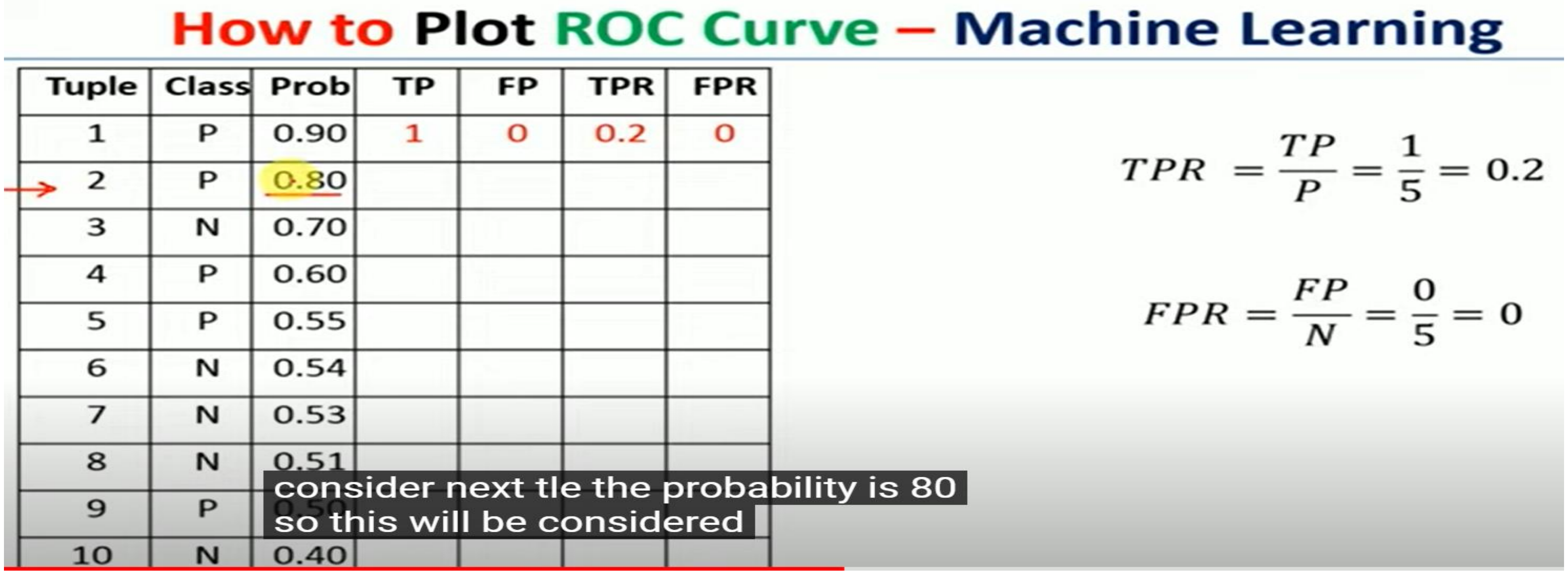
$$FPR = \frac{FP}{N}$$

in this case and tpr is equal to 1 / 5
and fpr is equal



Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.90	1	0	0.2	0
→ 2	p P	0.80				
3	N	0.70				
4	P	0.60				
5	P	0.55				
6	N	0.54				
7	N	0.53				
8	N	0.51				
9	P	0.50				
10	N	0.40				

example is actually positive the meaning
is true positive will



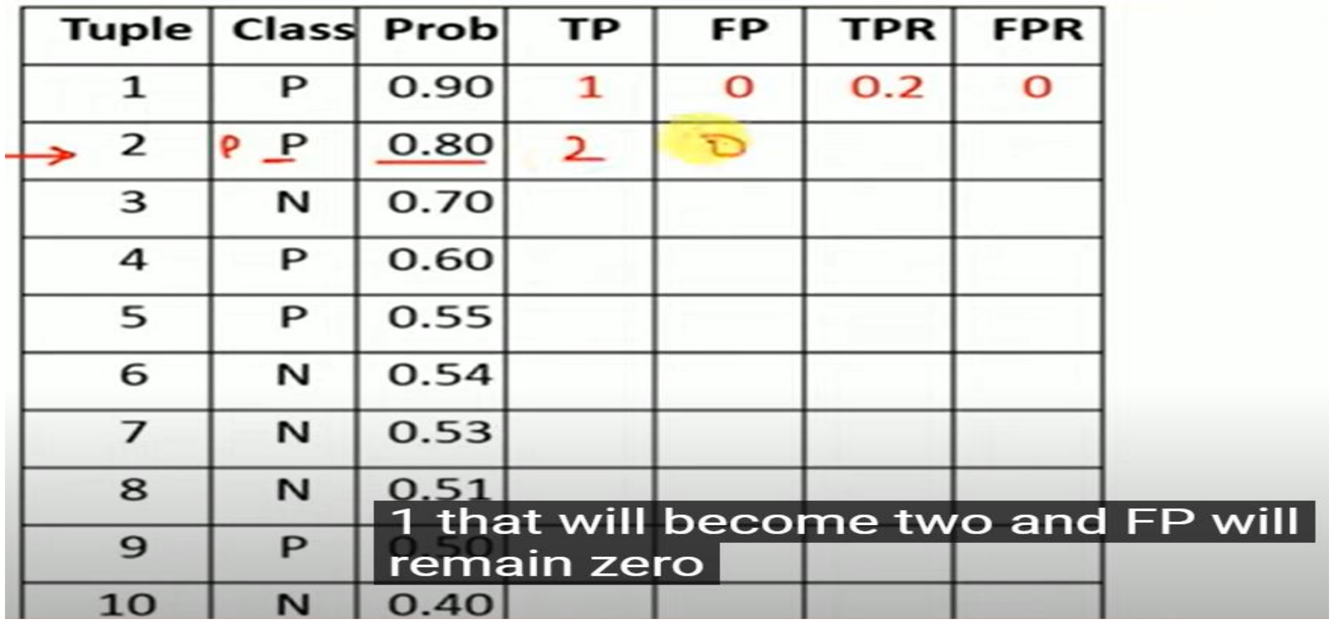
How to Plot ROC Curve – Machine Learning

Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.90	1	0	0.2	0
→ 2	P	0.80				
3	N	0.70				
4	P	0.60				
5	P	0.55				
6	N	0.54				
7	N	0.53				
8	N	0.51				
9	P	0.50				
10	N	0.40				

$$TPR = \frac{TP}{P} = \frac{1}{5} = 0.2$$

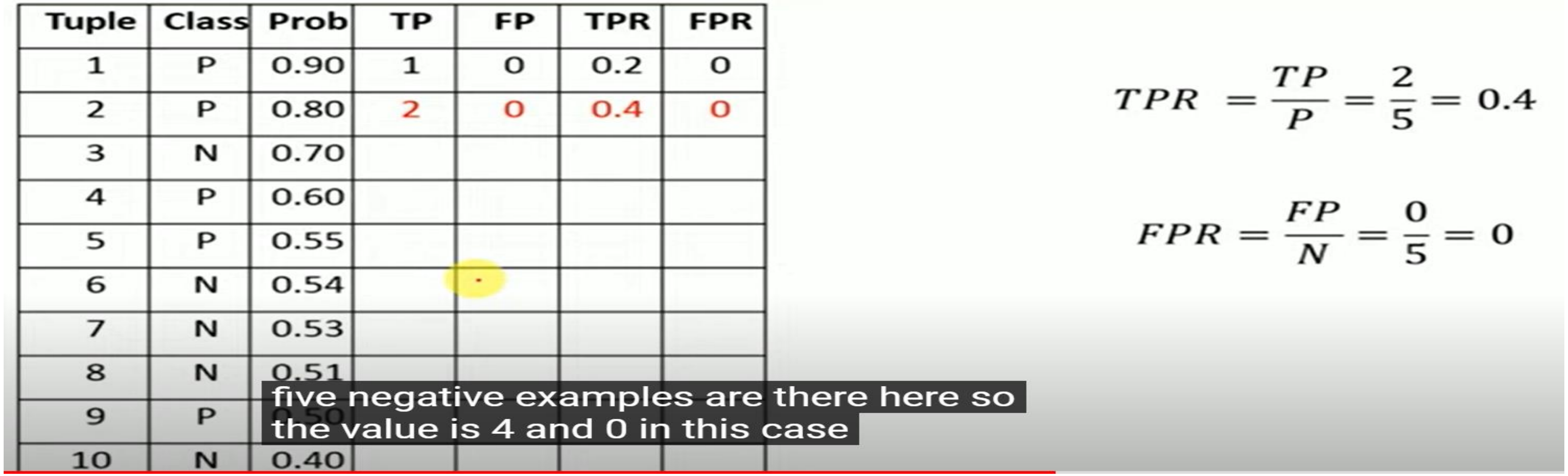
$$FPR = \frac{FP}{N} = \frac{0}{5} = 0$$

consider next tle the probability is 80
so this will be considered



Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.90	1	0	0.2	0
2	p P	0.80	2	0		
3	N	0.70				
4	P	0.60				
5	P	0.55				
6	N	0.54				
7	N	0.53				
8	N	0.51				
9	P	0.50				
10	N	0.40				

1 that will become two and FP will remain zero

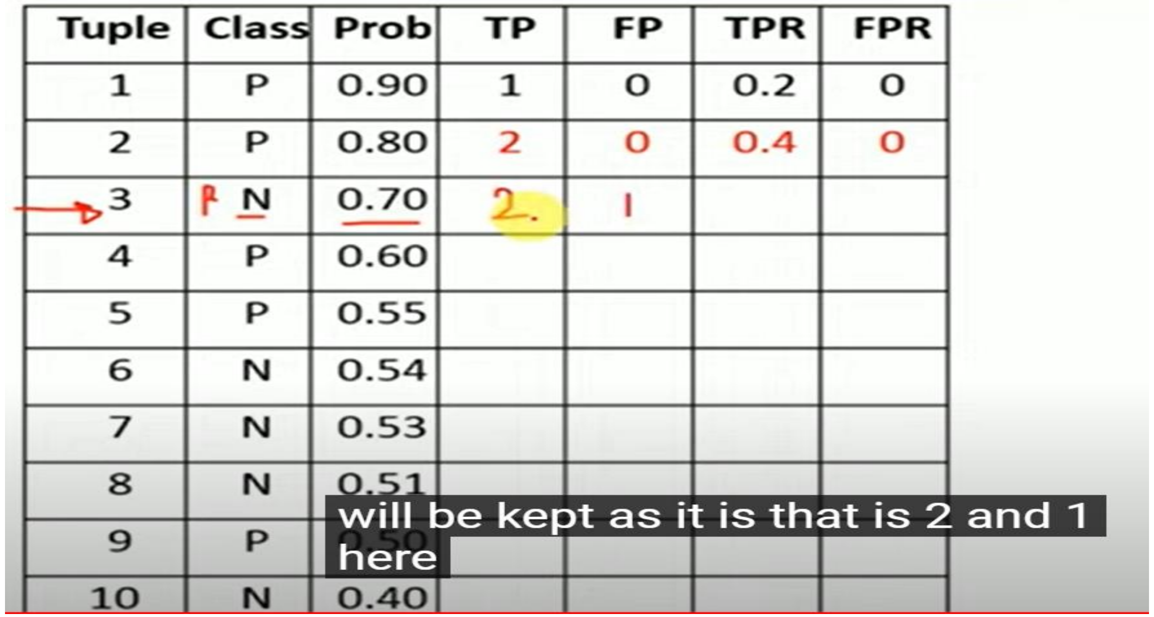


Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.90	1	0	0.2	0
2	P	0.80	2	0	0.4	0
3	N	0.70				
4	P	0.60				
5	P	0.55				
6	N	0.54				
7	N	0.53				
8	N	0.51				
9	P	0.50				
10	N	0.40				

$$TPR = \frac{TP}{P} = \frac{2}{5} = 0.4$$

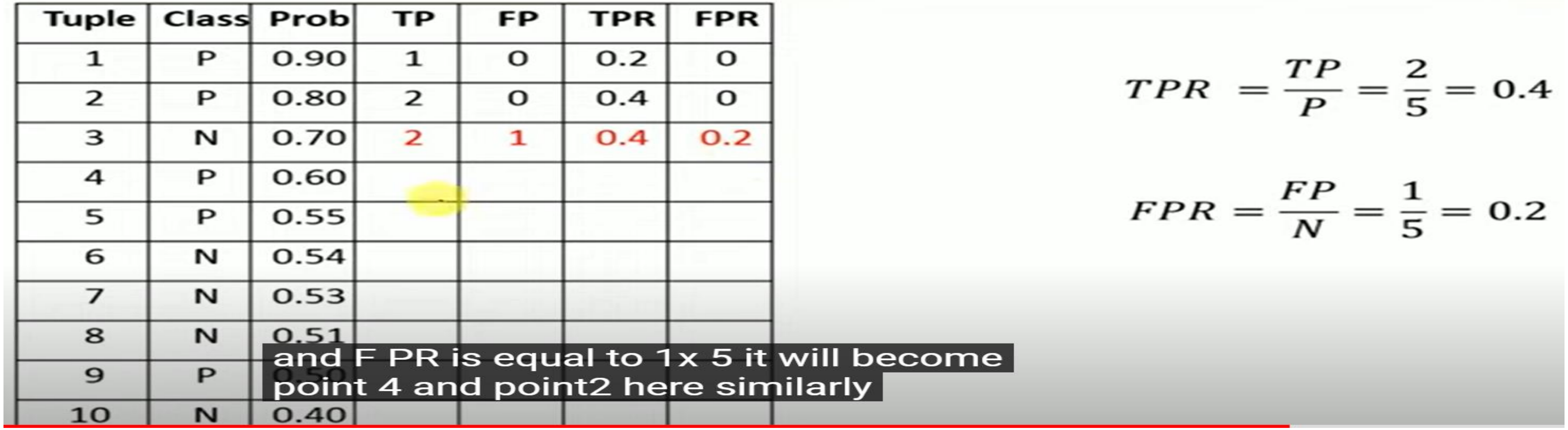
$$FPR = \frac{FP}{N} = \frac{0}{5} = 0$$

five negative examples are there here so the value is 4 and 0 in this case



Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.90	1	0	0.2	0
2	P	0.80	2	0	0.4	0
3	P N	0.70	2.	1		
4	P	0.60				
5	P	0.55				
6	N	0.54				
7	N	0.53				
8	N	0.51				
9	P	0.50				
10	N	0.40				

will be kept as it is that is 2 and 1 here



Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.90	1	0	0.2	0
2	P	0.80	2	0	0.4	0
3	N	0.70	2	1	0.4	0.2
4	P	0.60				
5	P	0.55				
6	N	0.54				
7	N	0.53				
8	N	0.51				
9	P	0.50				
10	N	0.40				

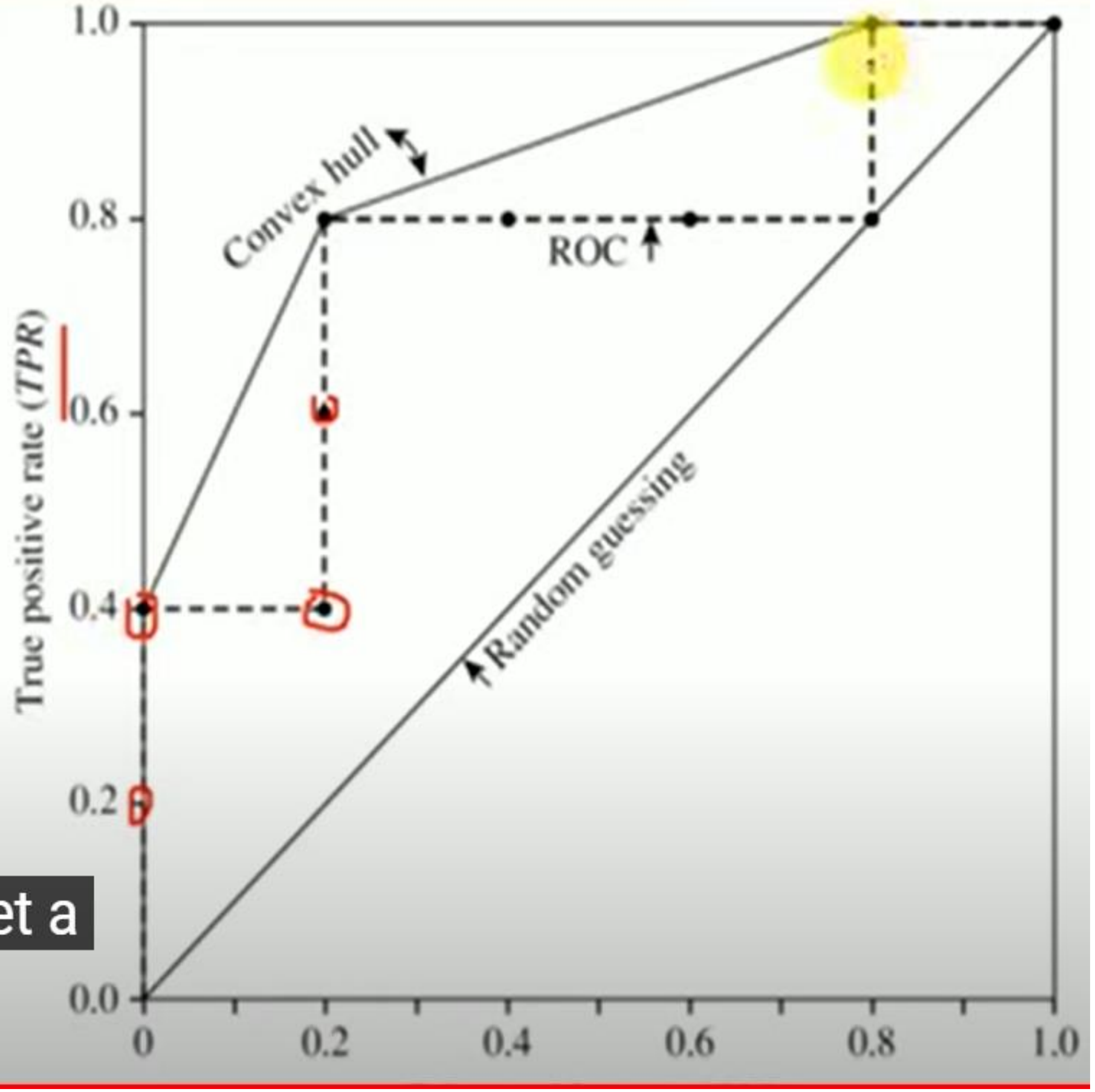
$$TPR = \frac{TP}{P} = \frac{2}{5} = 0.4$$

$$FPR = \frac{FP}{N} = \frac{1}{5} = 0.2$$

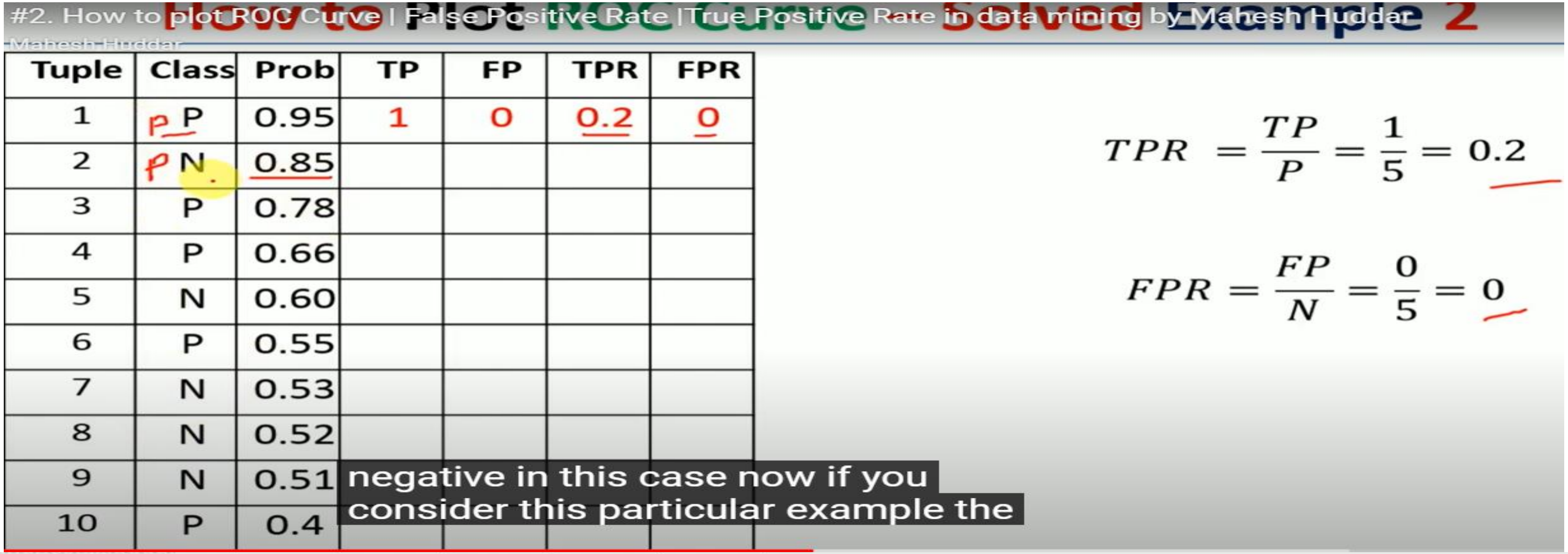
and F PR is equal to 1x 5 it will become point 4 and point2 here similarly

How to Plot Roc Curve – Machine Learning

Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.90	1	0	0.2	0
2	P	0.80	2	0	0.4	0
3	N	0.70	2	1	0.4	0.2
4	P	0.60	3	1	0.6	0.2
5	P	0.55	4	1	0.8	0.2
6	N	0.54	4	2	0.8	0.4
7	N	0.53	4	3	0.8	0.6
8	N	0.51	4	4	0.8	0.8
9	P	0.50	5	4	1.0	0.8
10	N	0.40	5	5	1.0	1.0



these part of things you will get a final r c



#2. How to plot ROC Curve | False Positive Rate |True Positive Rate in data mining by Mahesh Huddar

Mahesh Huddar

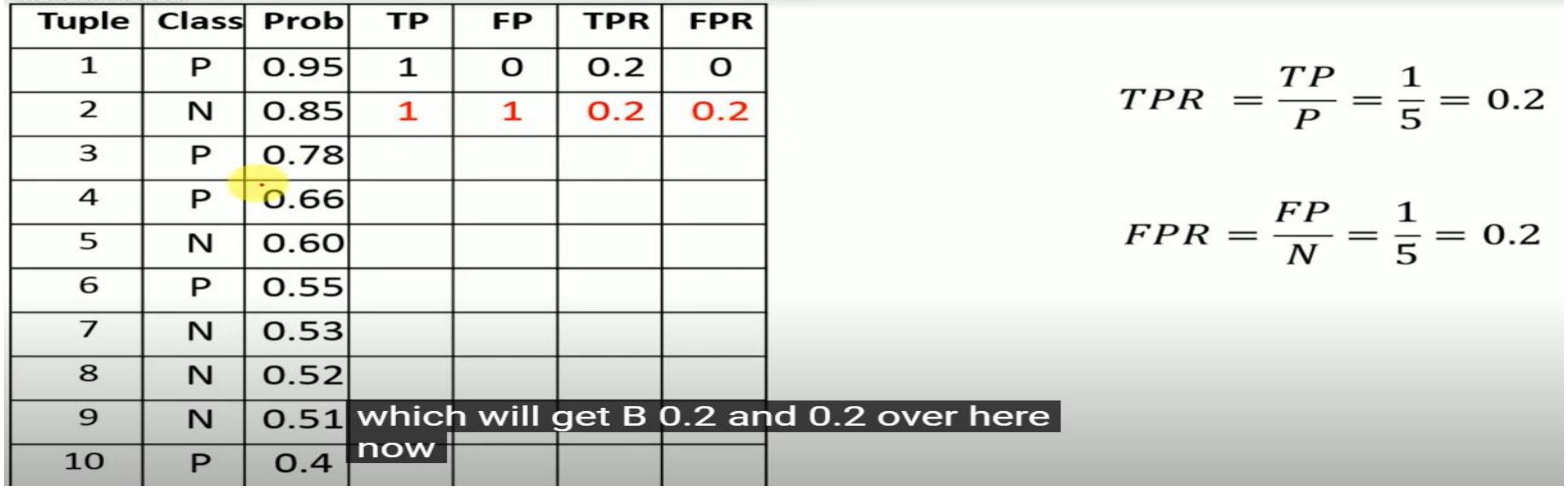
How to Plot ROC Curve Solved Example 2

Tuple	Class	Prob	TP	FP	TPR	FPR
1	p P	0.95	1	0	0.2	0
2	p N	0.85				
3	P	0.78				
4	P	0.66				
5	N	0.60				
6	P	0.55				
7	N	0.53				
8	N	0.52				
9	N	0.51				
10	P	0.4				

$$TPR = \frac{TP}{P} = \frac{1}{5} = 0.2$$

$$FPR = \frac{FP}{N} = \frac{0}{5} = 0$$

negative in this case now if you consider this particular example the

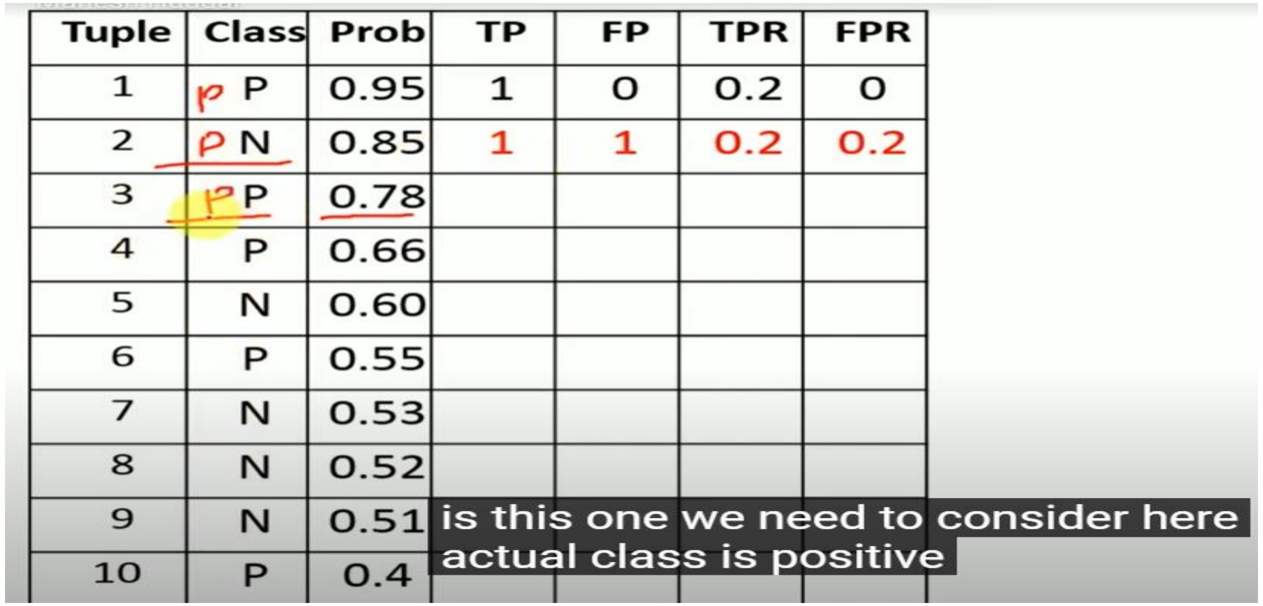


Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.95	1	0	0.2	0
2	N	0.85	1	1	0.2	0.2
3	P	0.78				
4	P	0.66				
5	N	0.60				
6	P	0.55				
7	N	0.53				
8	N	0.52				
9	N	0.51				
10	P	0.4				

$$TPR = \frac{TP}{P} = \frac{1}{5} = 0.2$$

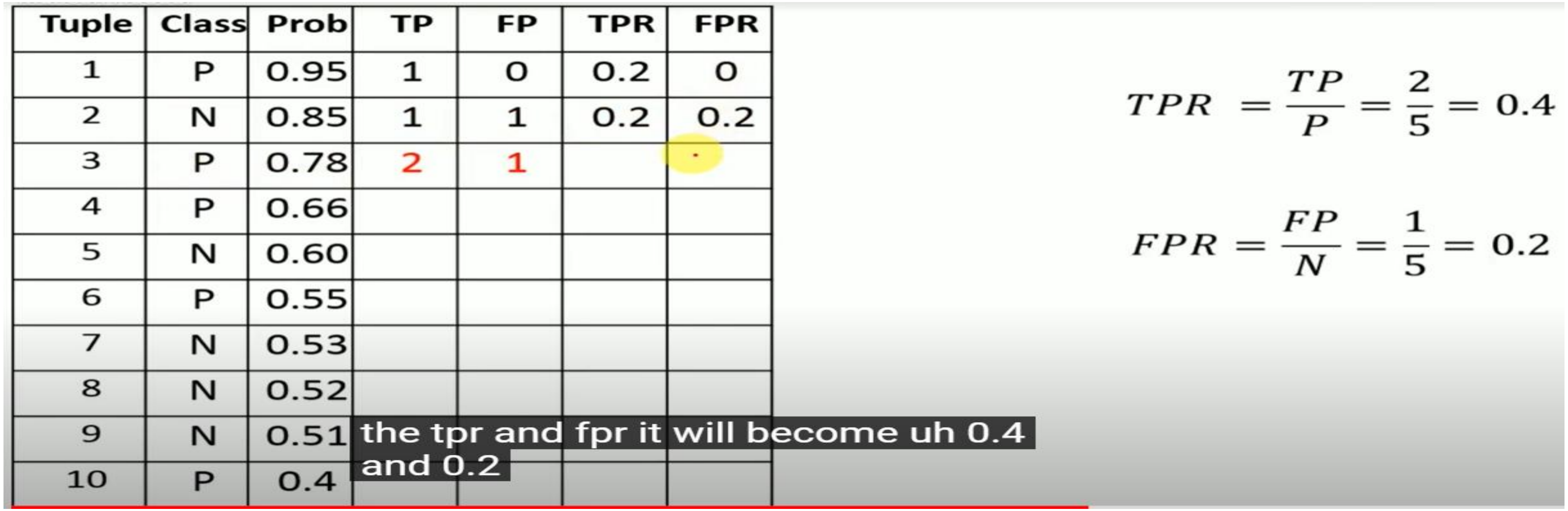
$$FPR = \frac{FP}{N} = \frac{1}{5} = 0.2$$

which will get B 0.2 and 0.2 over here now



Tuple	Class	Prob	TP	FP	TPR	FPR
1	p P	0.95	1	0	0.2	0
2	P N	0.85	1	1	0.2	0.2
3	P P	0.78				
4	P	0.66				
5	N	0.60				
6	P	0.55				
7	N	0.53				
8	N	0.52				
9	N	0.51				
10	P	0.4				

is this one we need to consider here
actual class is positive

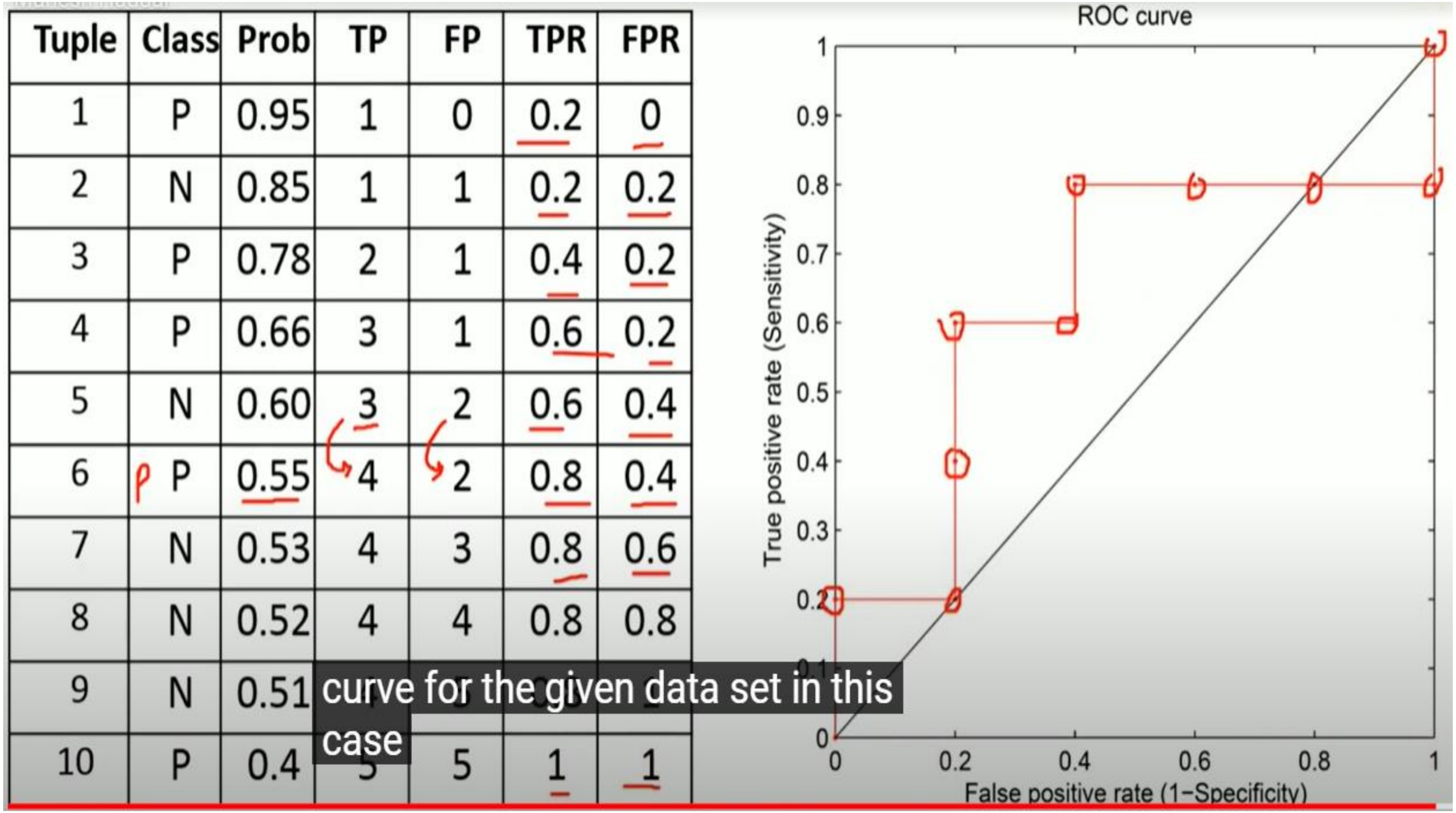


Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.95	1	0	0.2	0
2	N	0.85	1	1	0.2	0.2
3	P	0.78	2	1		
4	P	0.66				
5	N	0.60				
6	P	0.55				
7	N	0.53				
8	N	0.52				
9	N	0.51				
10	P	0.4				

$$TPR = \frac{TP}{P} = \frac{2}{5} = 0.4$$

$$FPR = \frac{FP}{N} = \frac{1}{5} = 0.2$$

the tpr and fpr it will become uh 0.4
and 0.2

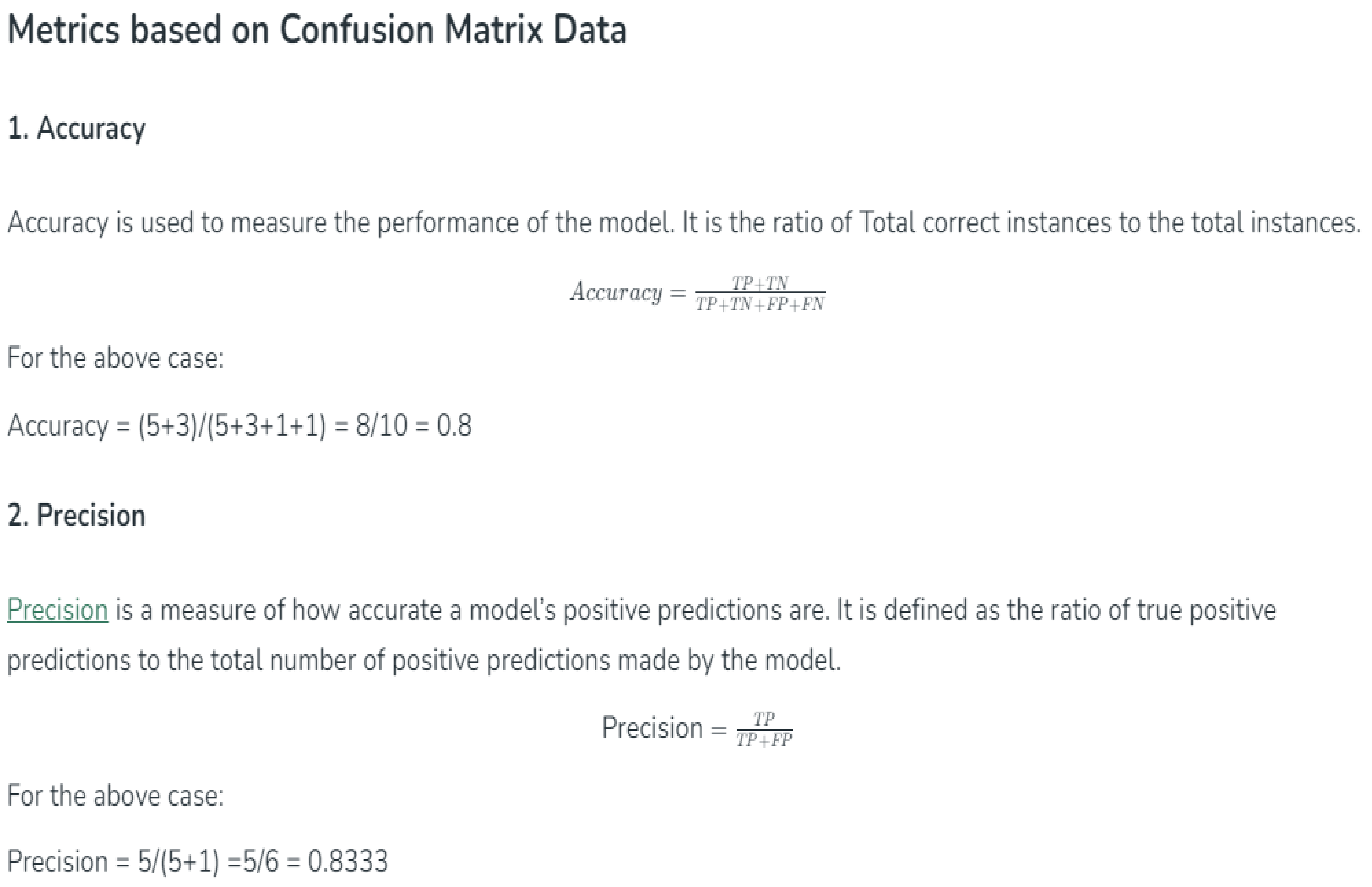


Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.95	1	0	0.2	0
2	N	0.85	1	1	0.2	0.2
3	P	0.78	2	1	0.4	0.2
4	P	0.66	3	1	0.6	0.2
5	N	0.60	3	2	0.6	0.4
6	p P	0.55	4	2	0.8	0.4
7	N	0.53	4	3	0.8	0.6
8	N	0.52	4	4	0.8	0.8
9	N	0.51	[illegible]	[illegible]	[illegible]	[illegible]
10	P	0.4	5	5	1	1

curve for the given data set in this case

ROC curve
True positive rate (Sensitivity)
False positive rate (1−Specificity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1

- **What is a Confusion Matrix?**
- A **confusion matrix** is a matrix that summarizes the performance of a machine learning model on a set of test data. It is a means of displaying the number of accurate and inaccurate instances based on the model’s predictions. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance.
- The matrix displays the number of instances produced by the model on the test data.
- **True Positive (TP):** The model correctly predicted a positive outcome (the actual outcome was positive).
- **True Negative (TN):** The model correctly predicted a negative outcome (the actual outcome was negative).
- **False Positive (FP):** The model incorrectly predicted a positive outcome (the actual outcome was negative). Also known as a Type I error.
- **False Negative (FN):** The model incorrectly predicted a negative outcome (the actual outcome was positive). Also known as a Type II error.
- **Why do we need a Confusion Matrix?**
- When assessing a classification model’s performance, a confusion matrix is essential. It offers a thorough analysis of true positive, true negative, false positive, and false negative predictions, facilitating a more profound comprehension of a model’s **recall, accuracy, precision,** and overall effectiveness in class distinction. When there is an uneven class distribution in a dataset, this matrix is especially helpful in evaluating a model’s performance beyond basic accuracy metrics.



Metrics based on Confusion Matrix Data

1. Accuracy

Accuracy is used to measure the performance of the model. It is the ratio of Total correct instances to the total instances.

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

For the above case:

Accuracy = (5+3)/(5+3+1+1) = 8/10 = 0.8

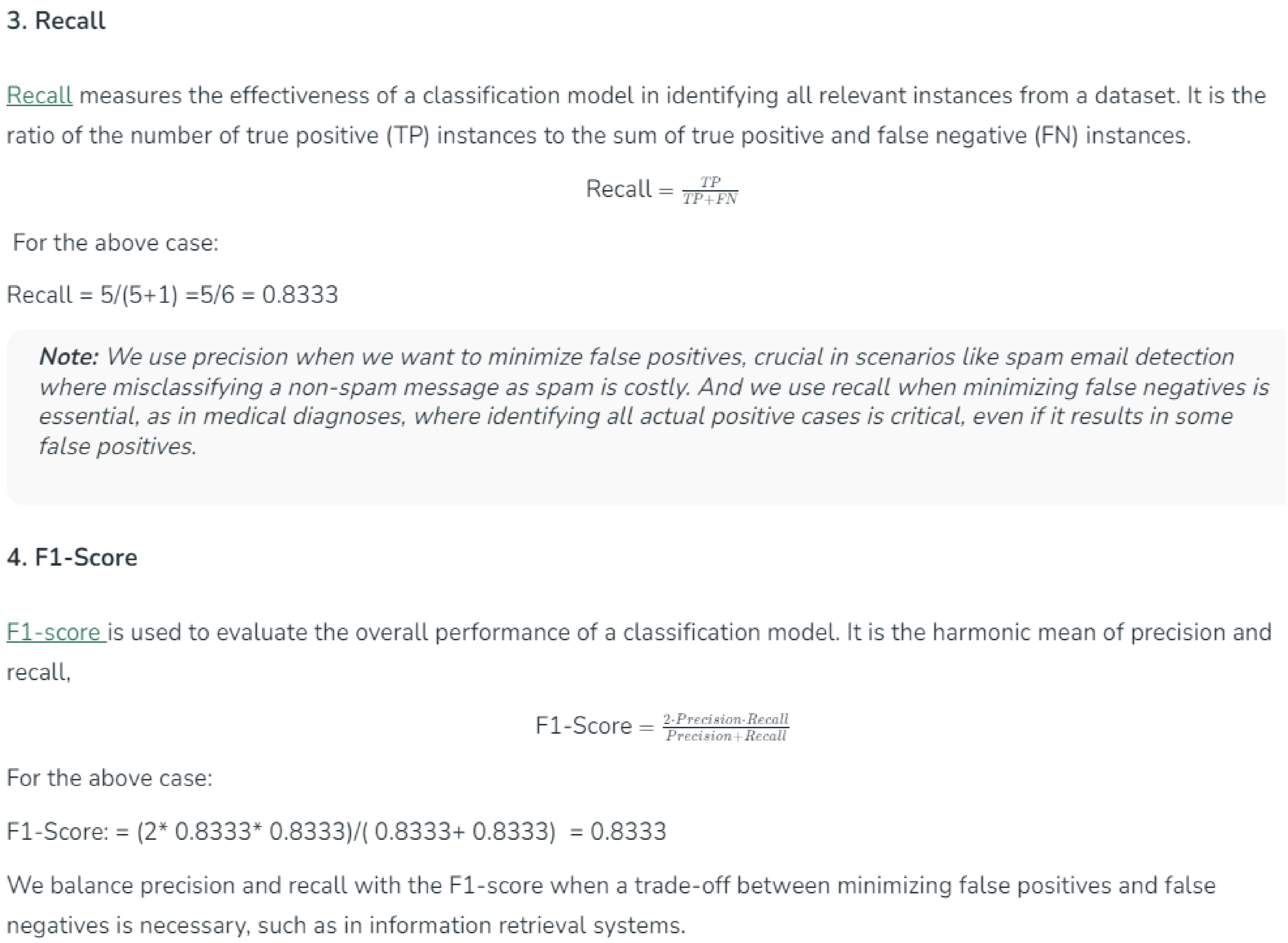
2. Precision

Precision is a measure of how accurate a model's positive predictions are. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model.

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

For the above case:

Precision = 5/(5+1) =5/6 = 0.8333



3. Recall

Recall measures the effectiveness of a classification model in identifying all relevant instances from a dataset. It is the ratio of the number of true positive (TP) instances to the sum of true positive and false negative (FN) instances.

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

For the above case:

Recall = 5/(5+1) =5/6 = 0.8333

> ***Note:*** *We use precision when we want to minimize false positives, crucial in scenarios like spam email detection where misclassifying a non-spam message as spam is costly. And we use recall when minimizing false negatives is essential, as in medical diagnoses, where identifying all actual positive cases is critical, even if it results in some false positives.*

4. F1-Score

F1-score is used to evaluate the overall performance of a classification model. It is the harmonic mean of precision and recall,

$$\text{F1-Score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

For the above case:

F1-Score: = (2* 0.8333* 0.8333)/(0.8333+ 0.8333) = 0.8333

We balance precision and recall with the F1-score when a trade-off between minimizing false positives and false negatives is necessary, such as in information retrieval systems.

5. Specificity

Specificity is another important metric in the evaluation of classification models, particularly in binary classification. It measures the ability of a model to correctly identify negative instances. Specificity is also known as the True Negative Rate. Formula is given by:

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

For example,

Specificity=3/(1+3)=3/4=0.75

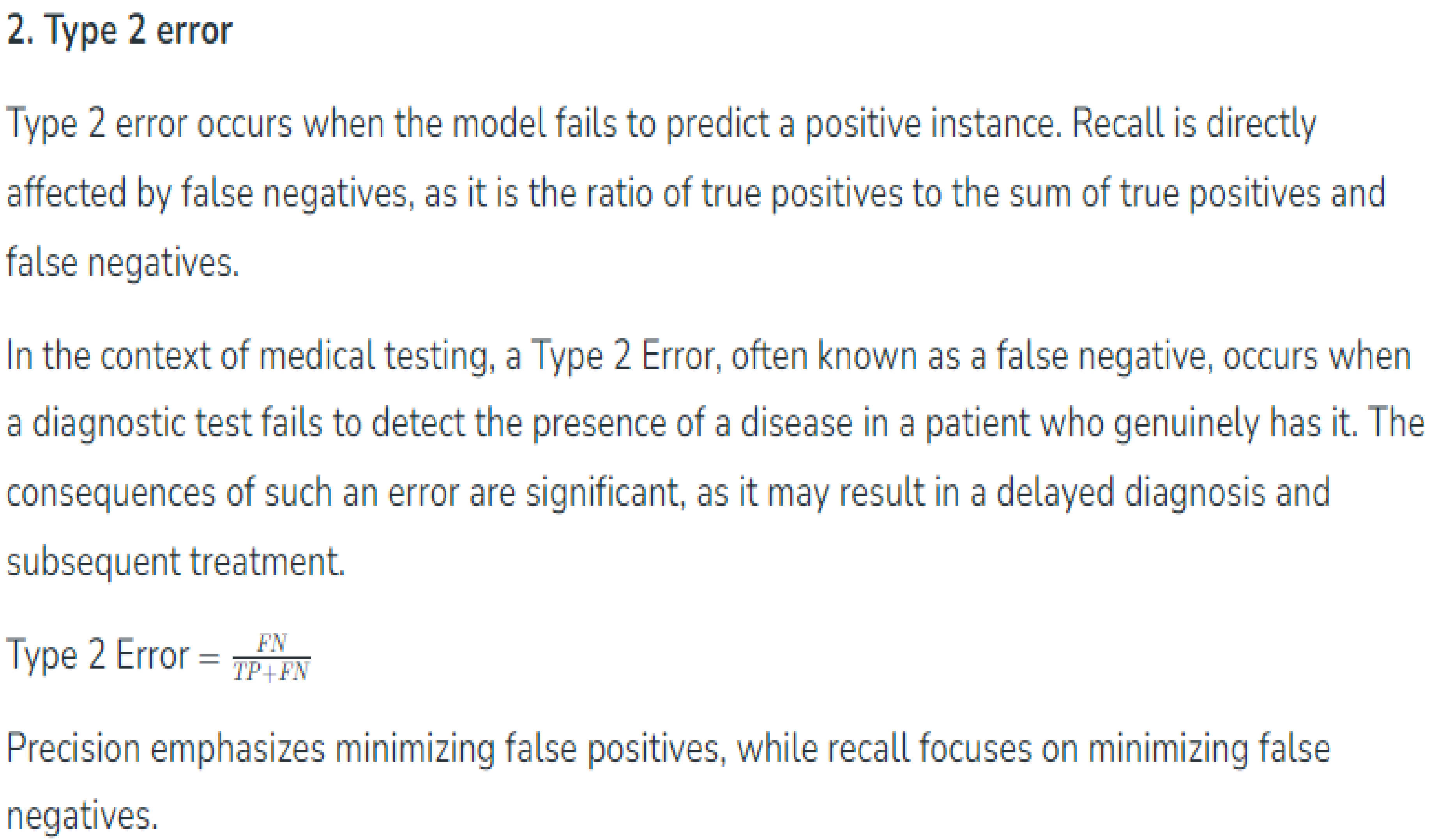
6. Type 1 and Type 2 error

1. Type 1 error

Type 1 error occurs when the model predicts a positive instance, but it is actually negative. Precision is affected by false positives, as it is the ratio of true positives to the sum of true positives and false positives.

Type 1 Error $= \frac{FP}{TN+FP}$

For example, in a courtroom scenario, a Type 1 Error, often referred to as a false positive, occurs when the court mistakenly convicts an individual as guilty when, in truth, they are innocent of the alleged crime. This grave error can have profound consequences, leading to the wrongful punishment of an innocent person who did not commit the offense in question. Preventing Type 1 Errors in legal proceedings is paramount to ensuring that justice is accurately served and innocent individuals are protected from unwarranted harm and punishment.



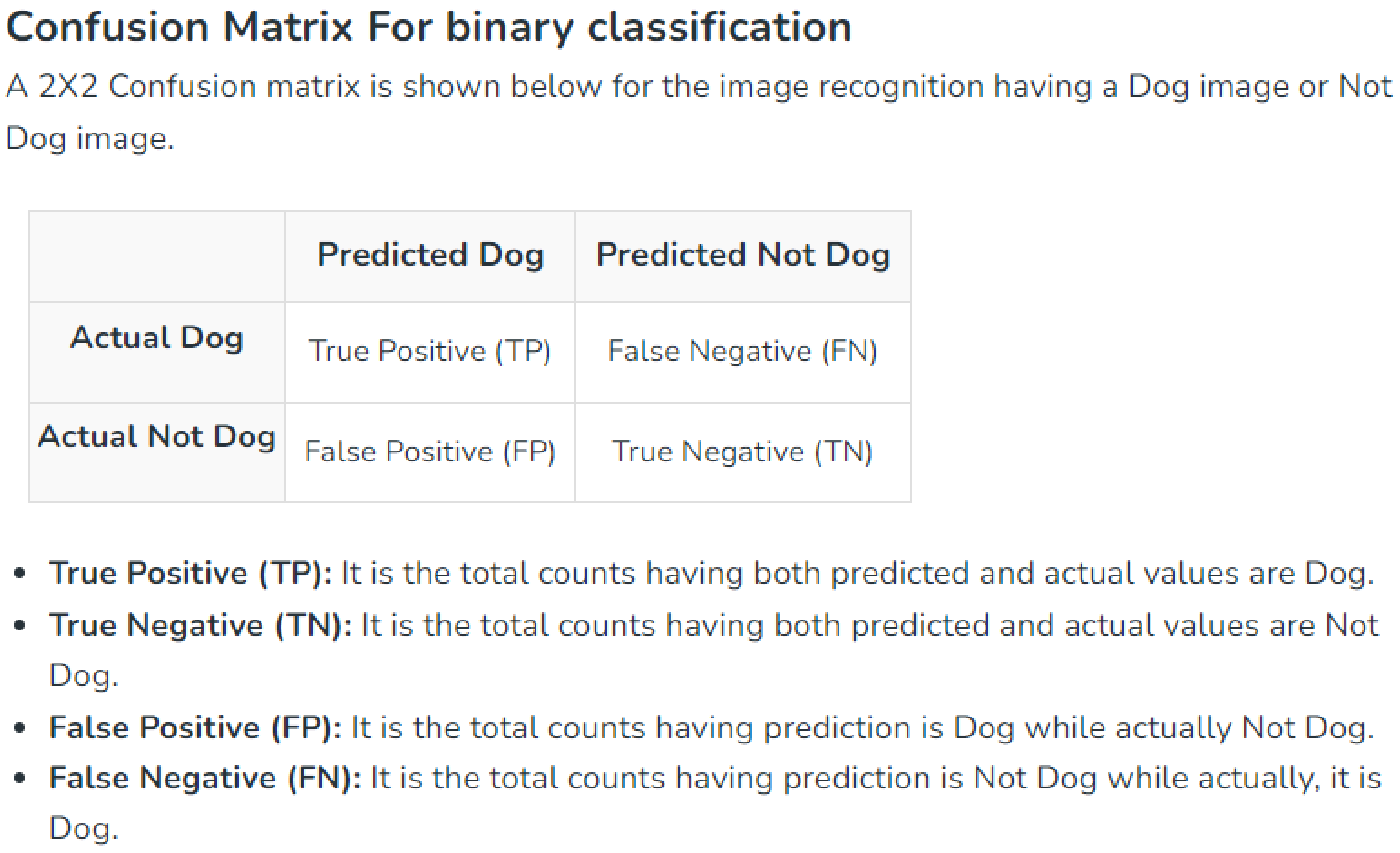
2. Type 2 error

Type 2 error occurs when the model fails to predict a positive instance. Recall is directly affected by false negatives, as it is the ratio of true positives to the sum of true positives and false negatives.

In the context of medical testing, a Type 2 Error, often known as a false negative, occurs when a diagnostic test fails to detect the presence of a disease in a patient who genuinely has it. The consequences of such an error are significant, as it may result in a delayed diagnosis and subsequent treatment.

Type 2 Error $= \frac{FN}{TP+FN}$

Precision emphasizes minimizing false positives, while recall focuses on minimizing false negatives.

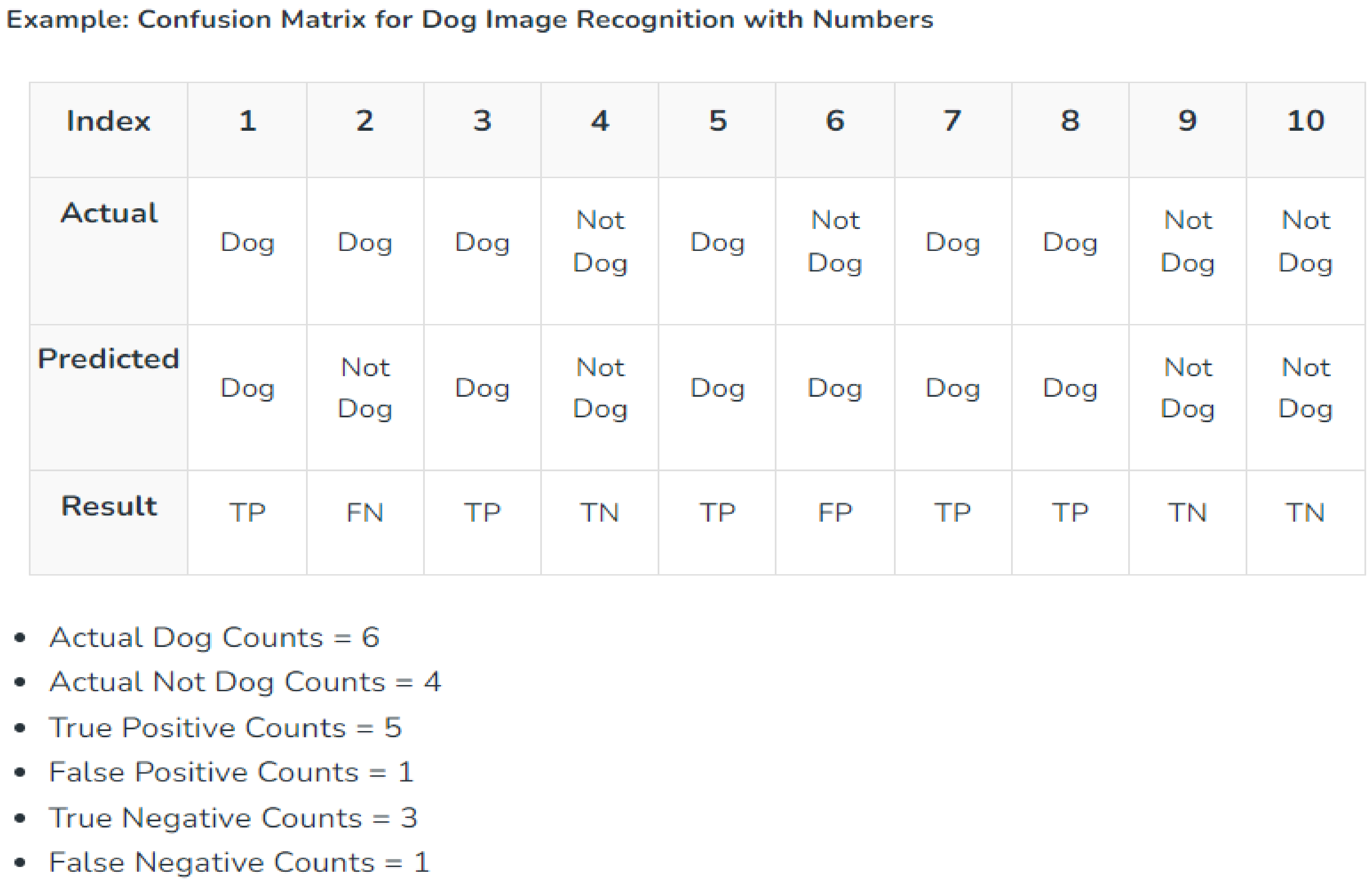


Confusion Matrix For binary classification

A 2X2 Confusion matrix is shown below for the image recognition having a Dog image or Not Dog image.

	Predicted Dog	Predicted Not Dog
Actual Dog	True Positive (TP)	False Negative (FN)
Actual Not Dog	False Positive (FP)	True Negative (TN)

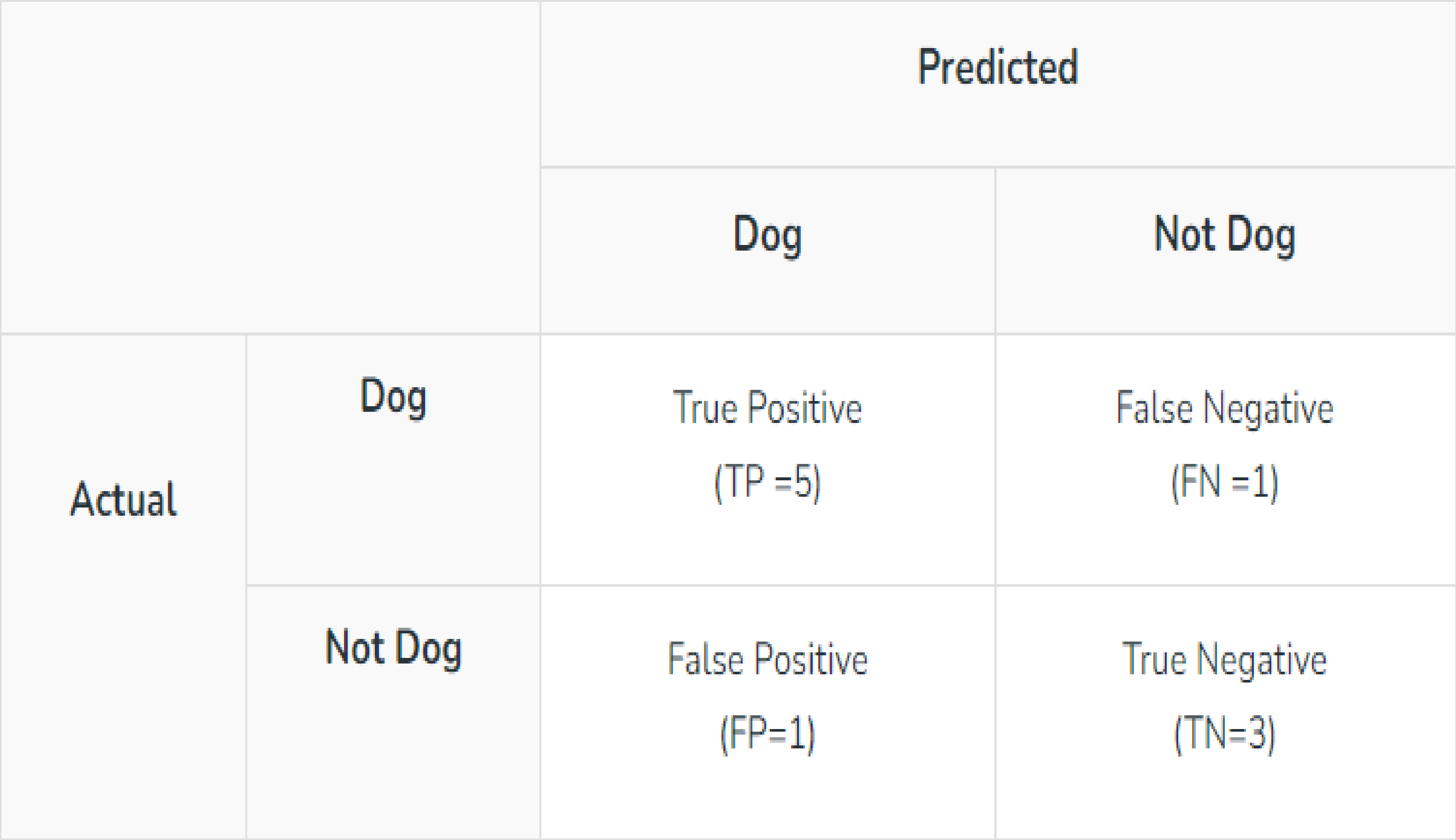
- **True Positive (TP):** It is the total counts having both predicted and actual values are Dog.
- **True Negative (TN):** It is the total counts having both predicted and actual values are Not Dog.
- **False Positive (FP):** It is the total counts having prediction is Dog while actually Not Dog.
- **False Negative (FN):** It is the total counts having prediction is Not Dog while actually, it is Dog.



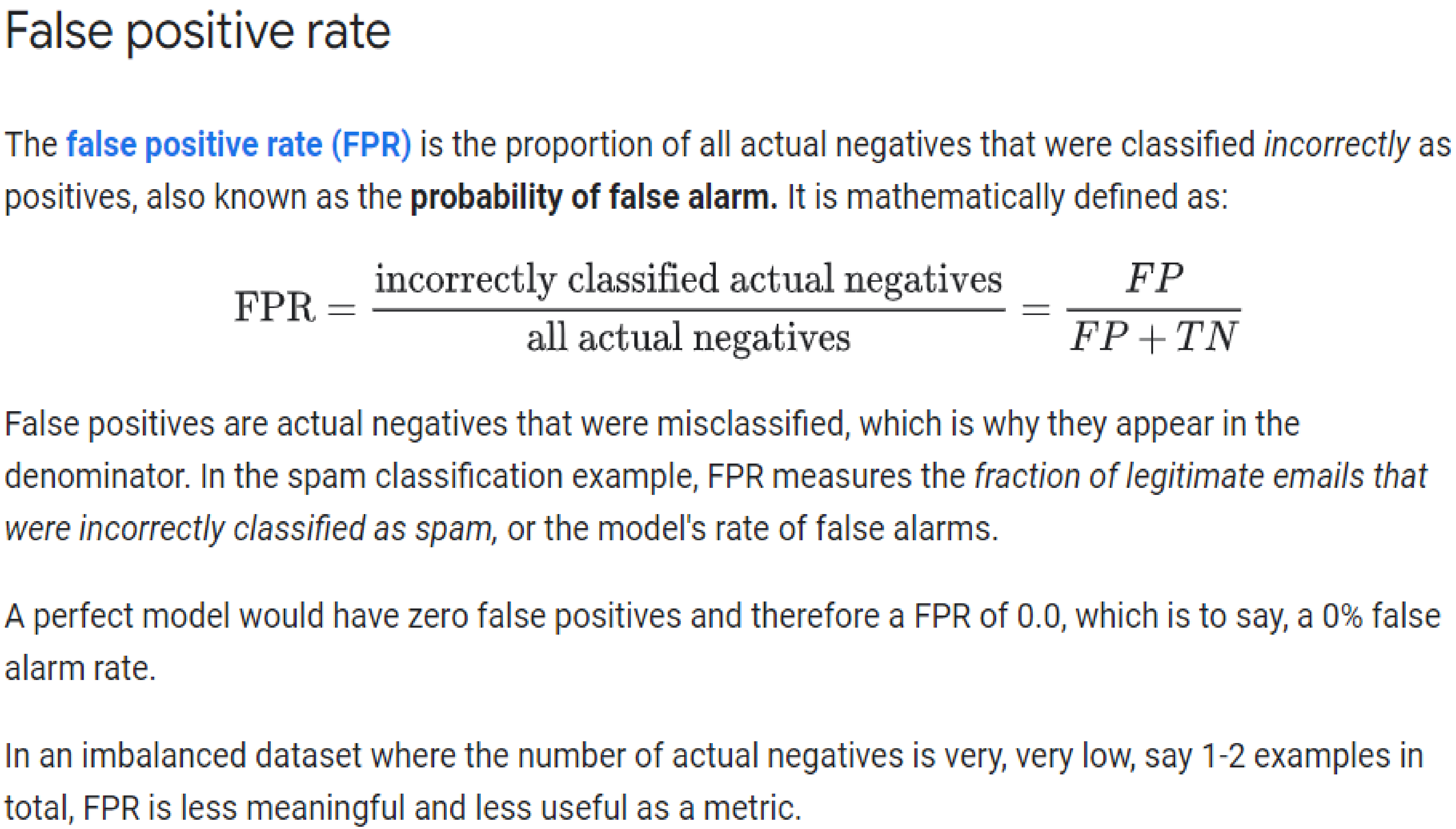
Example: Confusion Matrix for Dog Image Recognition with Numbers

Index	1	2	3	4	5	6	7	8	9	10
Actual	Dog	Dog	Dog	Not Dog	Dog	Not Dog	Dog	Dog	Not Dog	Not Dog
Predicted	Dog	Not Dog	Dog	Not Dog	Dog	Dog	Dog	Dog	Not Dog	Not Dog
Result	TP	FN	TP	TN	TP	FP	TP	TP	TN	TN

- Actual Dog Counts = 6
- Actual Not Dog Counts = 4
- True Positive Counts = 5
- False Positive Counts = 1
- True Negative Counts = 3
- False Negative Counts = 1



		Predicted	
		Dog	Not Dog
Actual	Dog	True Positive (TP =5)	False Negative (FN =1)
	Not Dog	False Positive (FP=1)	True Negative (TN=3)



False positive rate

The **false positive rate (FPR)** is the proportion of all actual negatives that were classified *incorrectly* as positives, also known as the **probability of false alarm.** It is mathematically defined as:

$$\text{FPR} = \frac{\text{incorrectly classified actual negatives}}{\text{all actual negatives}} = \frac{FP}{FP+TN}$$

False positives are actual negatives that were misclassified, which is why they appear in the denominator. In the spam classification example, FPR measures the *fraction of legitimate emails that were incorrectly classified as spam*, or the model's rate of false alarms.

A perfect model would have zero false positives and therefore a FPR of 0.0, which is to say, a 0% false alarm rate.

In an imbalanced dataset where the number of actual negatives is very, very low, say 1-2 examples in total, FPR is less meaningful and less useful as a metric.

- **True Labels (y_true)**: [1, 0, 1, 1, 0, 0, 1, 0]
- **Predicted Probabilities (y_scores)**: [0.9, 0.1, 0.8, 0.4, 0.2, 0.3, 0.85, 0.05]
- **Threshold**: 0.5

Step 1: Convert Probabilities to Predictions

Using the threshold, convert the predicted probabilities to binary predictions:

- Predictions (y_pred):
  - If probability ≥ 0.5, predict 1
  - If probability < 0.5, predict 0

From the given probabilities:

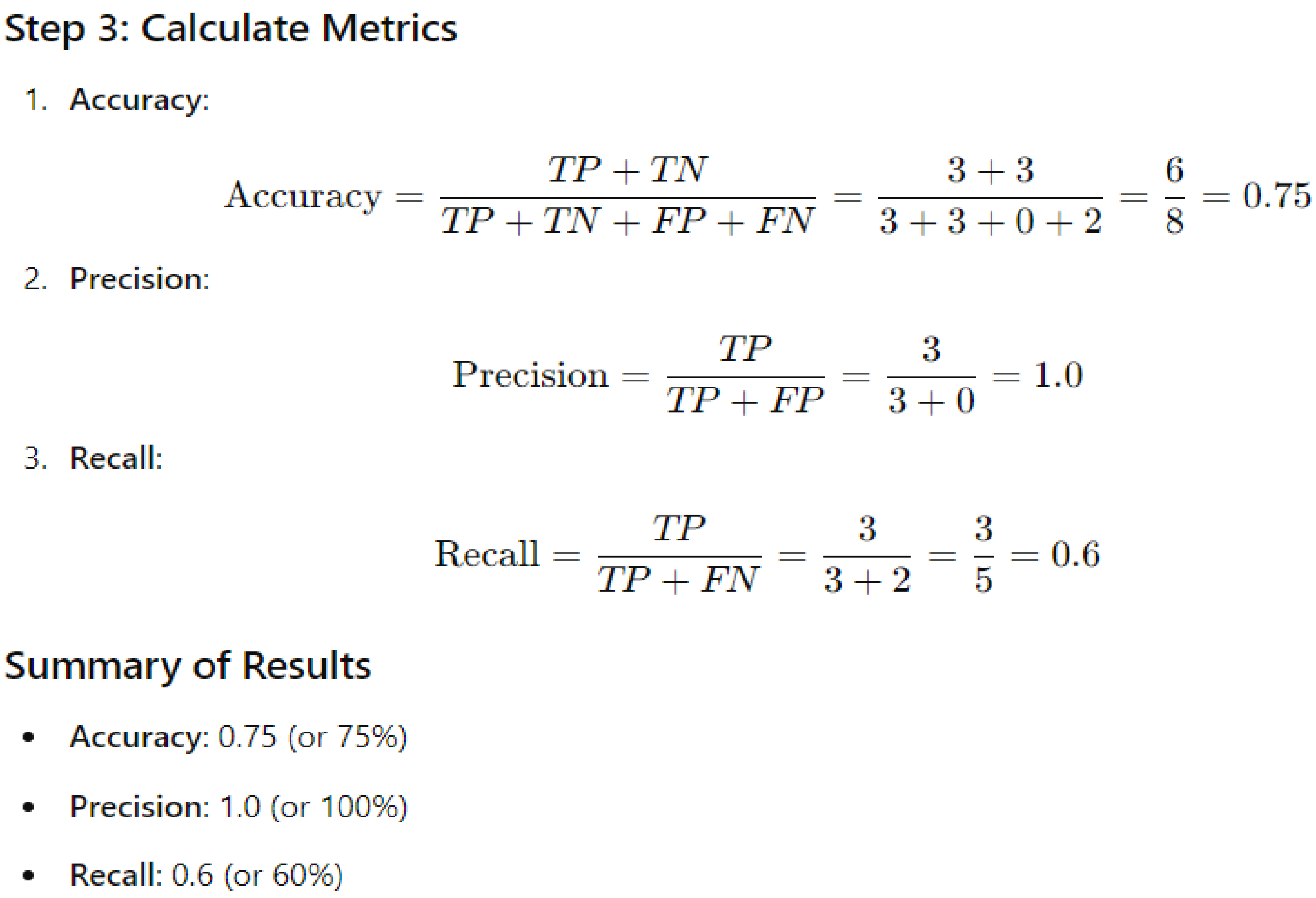
- y_pred = [1, 0, 1, 0, 0, 0, 1, 0]

Step 2: Calculate True Positives (TP), False Positives (FP), True Negatives (TN), False Negatives (FN)

- **True Positives (TP)**: Correctly predicted positive cases.
- **False Positives (FP)**: Incorrectly predicted positive cases.
- **True Negatives (TN)**: Correctly predicted negative cases.
- **False Negatives (FN)**: Incorrectly predicted negative cases.

Using y_true and y_pred:

- TP = 3 (indices 0, 2, 6)
- FP = 0
- TN = 3 (indices 1, 4, 5)
- FN = 2 (indices 3, 7)



Step 3: Calculate Metrics

1. **Accuracy:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{3+3}{3+3+0+2} = \frac{6}{8} = 0.75$$

2. **Precision:**

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{3}{3+0} = 1.0$$

3. **Recall:**

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{3}{3+2} = \frac{3}{5} = 0.6$$

Summary of Results

- **Accuracy**: 0.75 (or 75%)
- **Precision**: 1.0 (or 100%)
- **Recall**: 0.6 (or 60%)