Supervised learning

Classification

- Example
- Confusion matrix
- Classification metrics
- Trade-off
- ROC and AUC

Classification example

				TRUE VALUE ACTUAL VALUE GROUND TRUTH	*E.G. LOGISTIC REGRESSION MODEL*		
COUNT	**F1**	**F2**	**FN**	**DIAGNOSIS (POSITIVE/NEGATIVE)**	***DIAGNOSIS (POSITIVE/NEGATIVE)***		
1				+	+	TP	TRATAMIENTO->SALVO
2				+	+	TP	TRATAMIENTO->SALVO
3				+	+	TP	TRATAMIENTO->SALVO
4				+	+	TP	TRATAMIENTO->SALVO
5				-	+	FP	*TRATAMIENTO->$$$*
6				-	+	FP	*TRATAMIENTO->$$$*
7				-	+	FP	*TRATAMIENTO->$$$*
8				+	+	TP	TRATAMIENTO->SALVO
9				+	+	TP	TRATAMIENTO->SALVO
10				+	-	FN	**NO TRATAMIENTO->MUERE**
11				+	-	FN	**NO TRATAMIENTO->MUERE**
12				-	+	FP	*TRATAMIENTO->$$$*
13				-	+	FP	*TRATAMIENTO->$$$*
14				-	-	TN	NO TRAMIENTO->NO GASTA
15				-	-	TN	NO TRAMIENTO->NO GASTA
16				-	-	TN	NO TRAMIENTO->NO GASTA
17				-	-	TN	NO TRAMIENTO->NO GASTA
18				-	-	TN	NO TRAMIENTO->NO GASTA
19				-	-	TN	NO TRAMIENTO->NO GASTA
20				-	-	TN	NO TRAMIENTO->NO GASTA
21				-	-	TN	NO TRAMIENTO->NO GASTA
22				-	-	TN	NO TRAMIENTO->NO GASTA
23				-	-	TN	NO TRAMIENTO->NO GASTA
24				-	-	TN	NO TRAMIENTO->NO GASTA
25				-	-	TN	NO TRAMIENTO->NO GASTA
26				-	-	TN	NO TRAMIENTO->NO GASTA
27				-	-	TN	NO TRAMIENTO->NO GASTA
28				-	-	TN	NO TRAMIENTO->NO GASTA
29				-	-	TN	NO TRAMIENTO->NO GASTA
30				-	-	TN	NO TRAMIENTO->NO GASTA
31				-	-	TN	NO TRAMIENTO->NO GASTA
32				-	-	TN	NO TRAMIENTO->NO GASTA
33				-	-	TN	NO TRAMIENTO->NO GASTA
34				-	-	TN	NO TRAMIENTO->NO GASTA

Classification example

				TRUE VALUE ACTUAL VALUE GROUND TRUTH	*E.G. LOGISTIC REGRESSION MODEL*		
COUNT	**F1**	**F2**	**FN**	**DIAGNOSIS (POSITIVE/NEGATIVE)**	***DIAGNOSIS (POSITIVE/NEGATIVE)***		
1				+	+	TP	TRATAMIENTO->SALVO
2				+	+	TP	TRATAMIENTO->SALVO
3				+	+	TP	TRATAMIENTO->SALVO
4				+	+	TP	TRATAMIENTO->SALVO
5				-	+	FP	*TRATAMIENTO->$$$*
6				-	+	FP	*TRATAMIENTO->$$$*
7				-	+	FP	*TRATAMIENTO->$$$*
8				+	+	TP	TRATAMIENTO->SALVO
9				+	+	TP	TRATAMIENTO->SALVO
10				+	-	FN	**NO TRATAMIENTO->MUERE**
11				+	-	FN	**NO TRATAMIENTO->MUERE**
12				-	+	FP	*TRATAMIENTO->$$$*
13				-	+	FP	*TRATAMIENTO->$$$*
14				-	-	TN	NO TRAMIENTO->NO GASTA
15				-	-	TN	NO TRAMIENTO->NO GASTA
16				-	-	TN	NO TRAMIENTO->NO GASTA
17				-	-	TN	NO TRAMIENTO->NO GASTA
18				-	-	TN	NO TRAMIENTO->NO GASTA
19				-	-	TN	NO TRAMIENTO->NO GASTA
20				-	-	TN	NO TRAMIENTO->NO GASTA
21				-	-	TN	NO TRAMIENTO->NO GASTA
22				-	-	TN	NO TRAMIENTO->NO GASTA
23				-	-	TN	NO TRAMIENTO->NO GASTA
24				-	-	TN	NO TRAMIENTO->NO GASTA
25				-	-	TN	NO TRAMIENTO->NO GASTA
26				-	-	TN	NO TRAMIENTO->NO GASTA
27				-	-	TN	NO TRAMIENTO->NO GASTA
28				-	-	TN	NO TRAMIENTO->NO GASTA
29				-	-	TN	NO TRAMIENTO->NO GASTA
30				-	-	TN	NO TRAMIENTO->NO GASTA
31				-	-	TN	NO TRAMIENTO->NO GASTA
32				-	-	TN	NO TRAMIENTO->NO GASTA
33				-	-	TN	NO TRAMIENTO->NO GASTA
34				-	-	TN	NO TRAMIENTO->NO GASTA

Predicted values

Actual values		Positive	Negative
	Positive	**TP** **6**	**FN** **2**
	Negative	**FP** **5**	**TN** **21**

Confusion matrix

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

- It is a technique for visualizing the performance of a classification model
- The confusion matrix decomposes predictions into several categories of interest

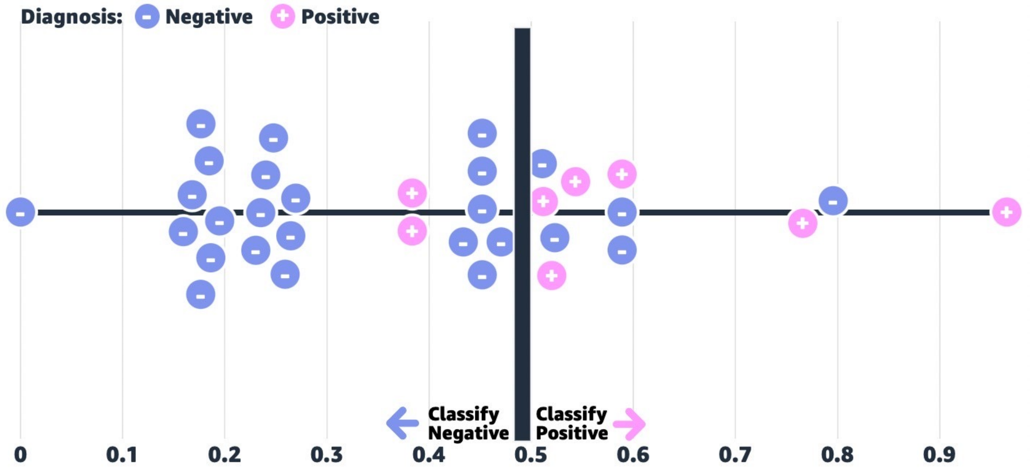
TP	True Positive: predicted positive when the actual is positive
TN	True Negative: predicted negative when the actual is negative
FN	False Negative: predicted negative when the actual is positive
FP	False Positive: predicted positive when the actual is negative

Classification example

- Image that we've trained a binary classification model to diagnose an individual as having cancer or not
  - Our model will output a probability for each individual of having cancer, and we'll compare this probability to the value of our classification threshold to determine whether or not an individual has cancer or not.
  - The *classification threshold* is just a value we use to translate our probabilities into binary outputs.

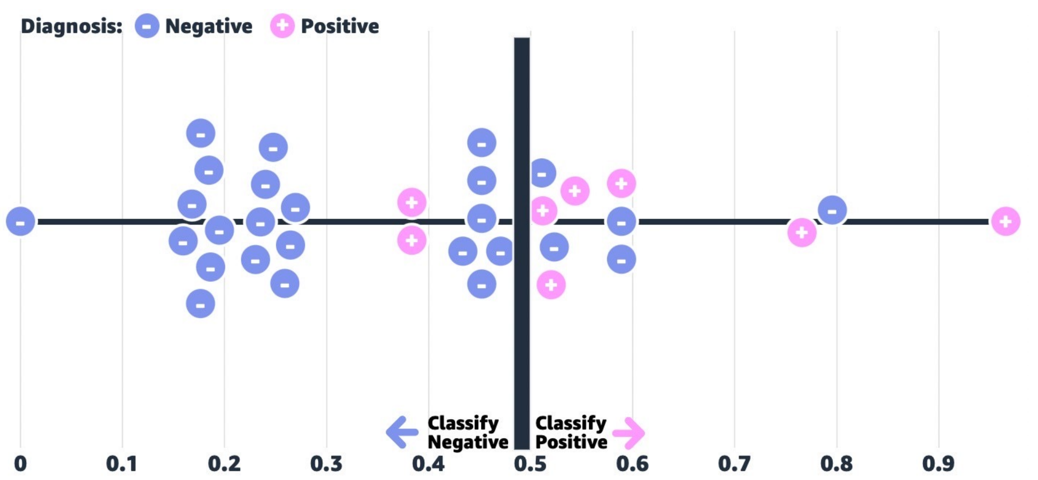
Classification example

If our *classification threshold is 0.5*, we'll classify any patient with a probability greater than 0.5 as being cancer-positive, and any patient with a probability less than 0.5 as being cancer-free



Classification example

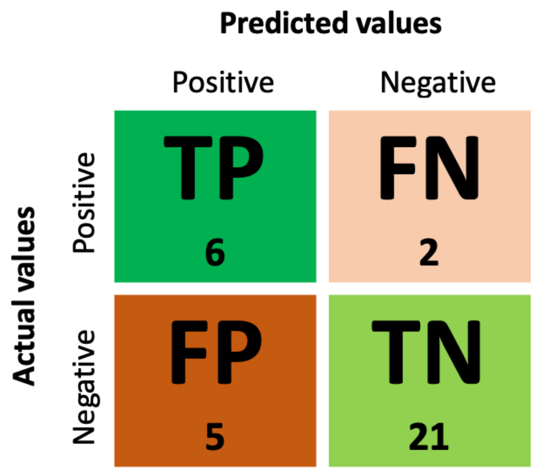
If our *classification threshold is 0.5*, we'll classify any patient with a probability greater than 0.5 as being cancer-positive, and any patient with a probability less than 0.5 as being cancer-free



		Predicted values	
		Positive	Negative
Actual values	Positive	TP 6	FN 2
	Negative	FP 5	TN 21

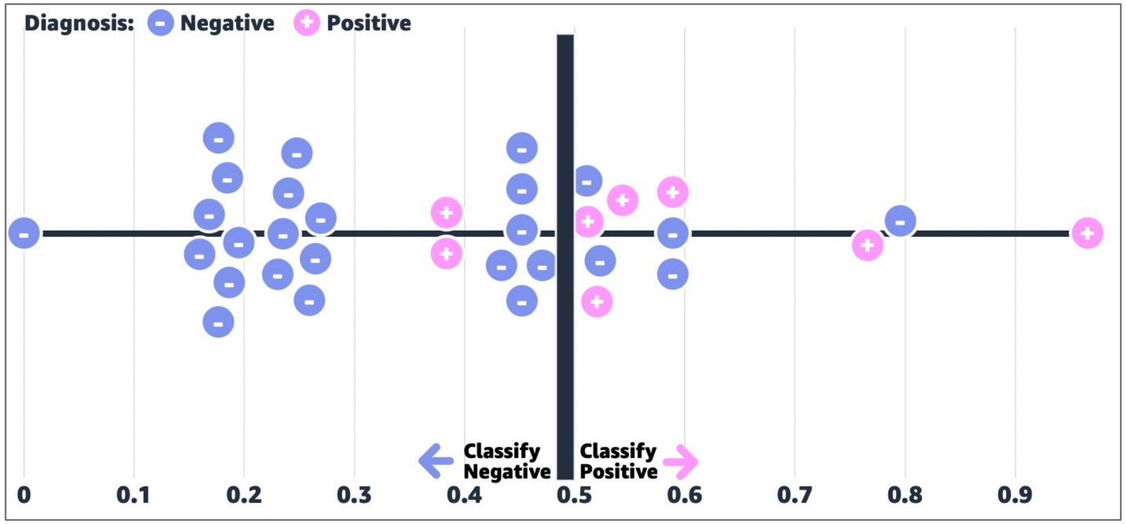
Evaluation Metrics in Classification Models

- Accuracy
- Precision
- True Positive Rate = Recall = Sensitivity
- False Positive Rate = 1-Sensitivity
- True Negative Rate = Specificity



Accuracy

The percent (ratio) of cases classified correctly



$$(bad) \leq accuracy \leq 1(good)$$

Predicted values

Actual values	Positive	Negative
Positive	TP 6	FN 2
Negative	FP 5	TN 21

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

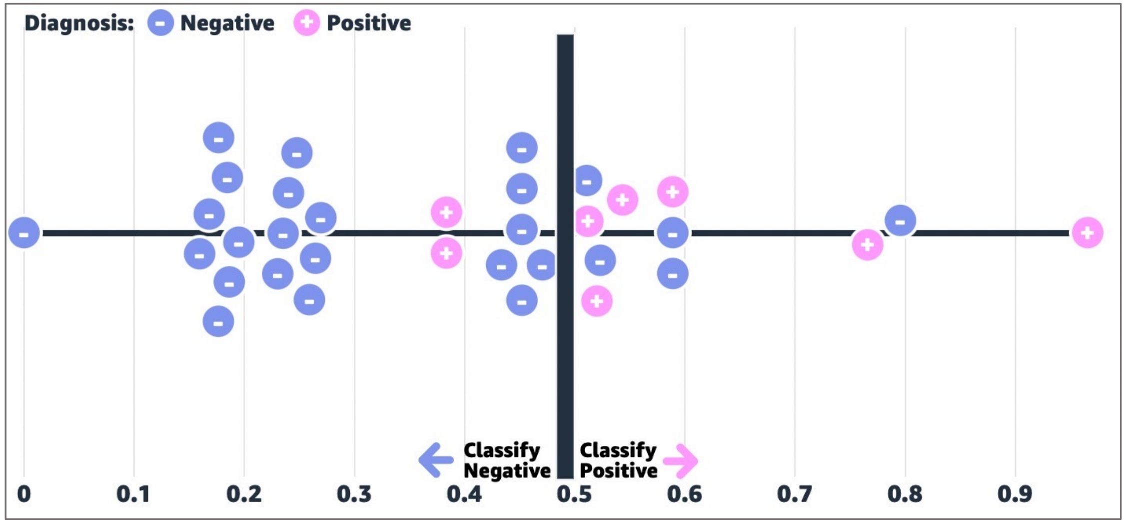
$$Accuracy = \frac{6 + 21}{6 + 5 + 2 + 21}$$

$$Accuracy \approx 79\%$$

Accuracy

High accuracy paradox: Accuracy is misleading when dealing with <u>imbalance</u> dataset (e.g., our data has three times more negative examples than positive, so few TP and many TN)
High accuracy even when few TP

$Accuracy \approx 79\%$



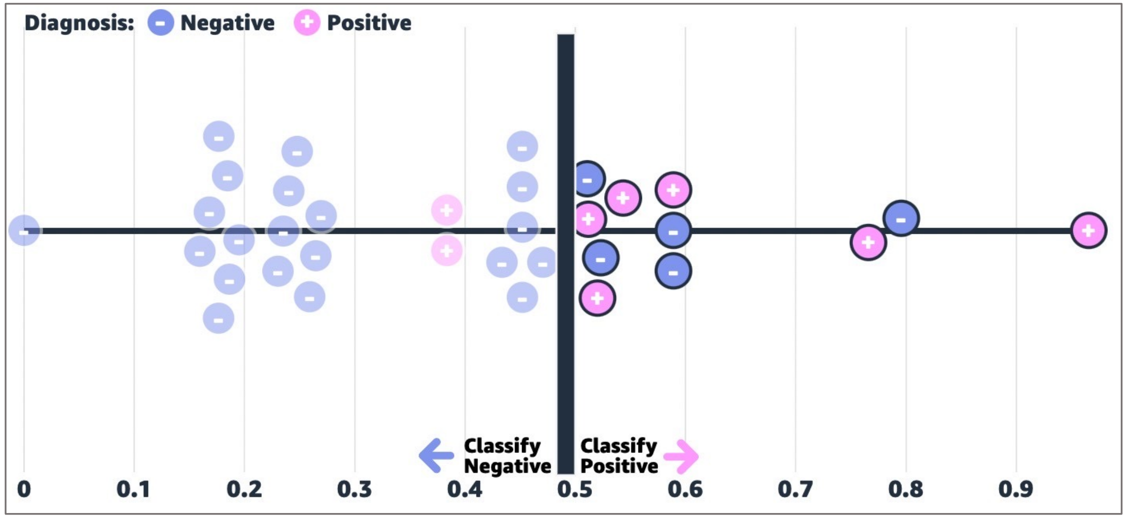
Predicted values

Actual values		Positive	Negative
	Positive	**TP** 6	**FN** 2
	Negative	**FP** 5	**TN** 21

Precision (Positive Predicted Values)

It is the ratio of correctly predicted positive classes to *all items <u>predicted as positive</u>*

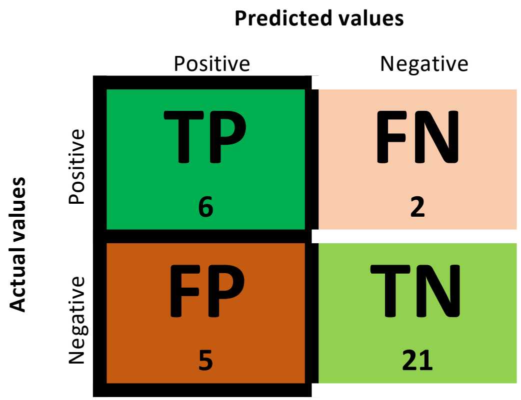
$$(bad) \leq precision \leq 1(good)$$



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

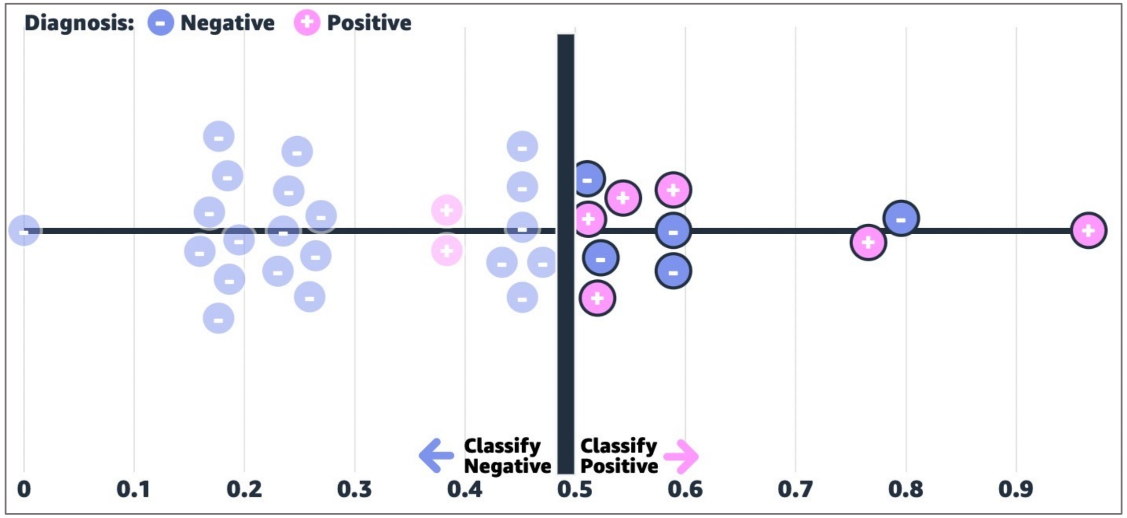
$$Precision = \frac{6}{6 + 5}$$

$$Precision \approx 55\%$$



Precision (Positive Predicted Values)

- It tells us how correct, or *precise*, are our model's positive predictions.
- It is important when we believe False Positives **FP** are more important than False Negatives **FN** (e.g. spam detection)

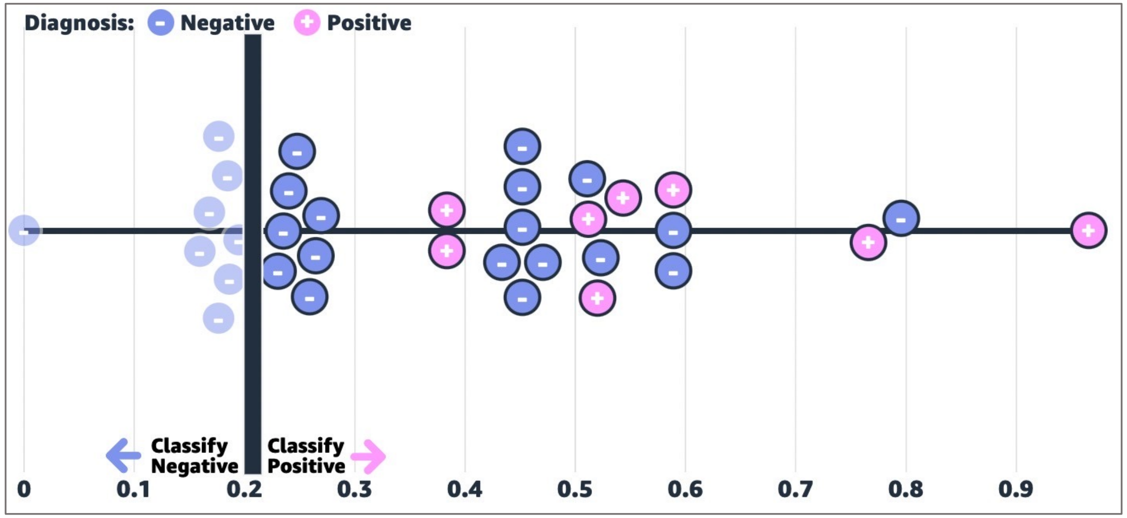


Predicted values

Actual values	Positive (predicted)	Negative (predicted)
Positive	**TP** 6	**FN** 2
Negative	**FP** 5	**TN** 21

Precision (Positive Predicted Values)

Precision worsens with the increase of False Positives **FP**



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

$$Precision = \frac{8}{8 + 18}$$

$$Precision \approx 31\%$$

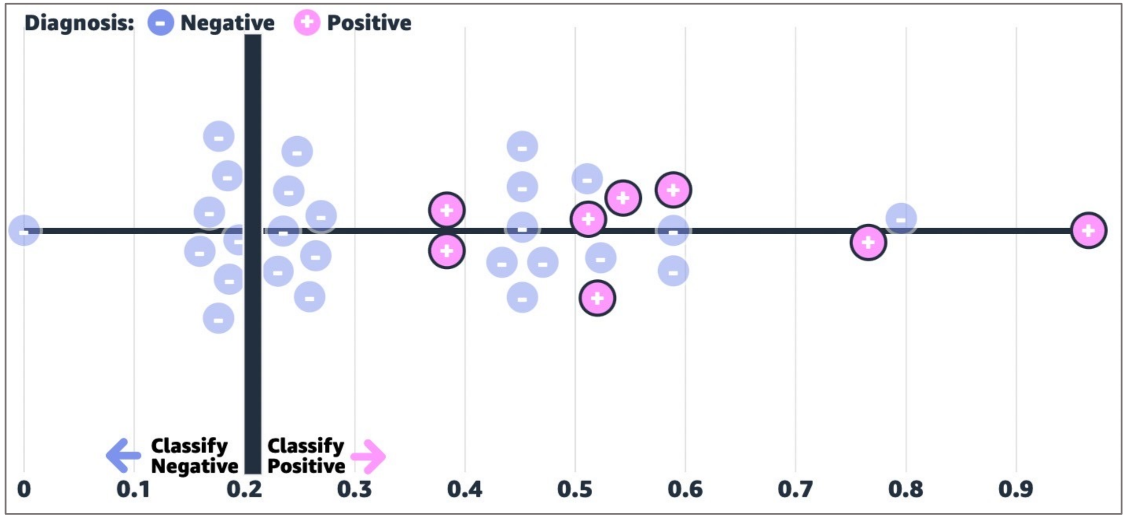
Predicted values

Actual values		Positive	Negative
	Positive	**TP** 8	**FN** 0
	Negative	**FP** 18	**TN** 8

Recall (Sensitivity or True Positive Rate)

It is the ratio of correctly predicted positive classes to *all items <u>that are actually positive</u>*

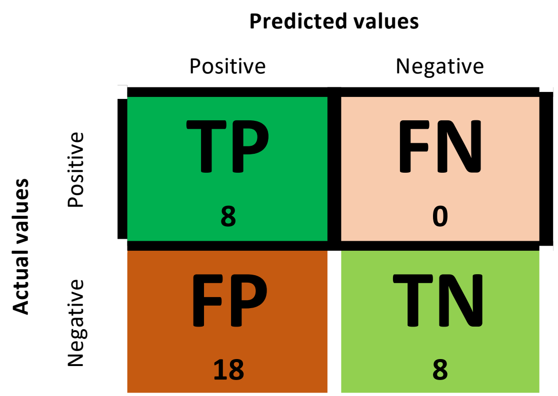
$$(bad) \leq recall \leq 1(good)$$



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

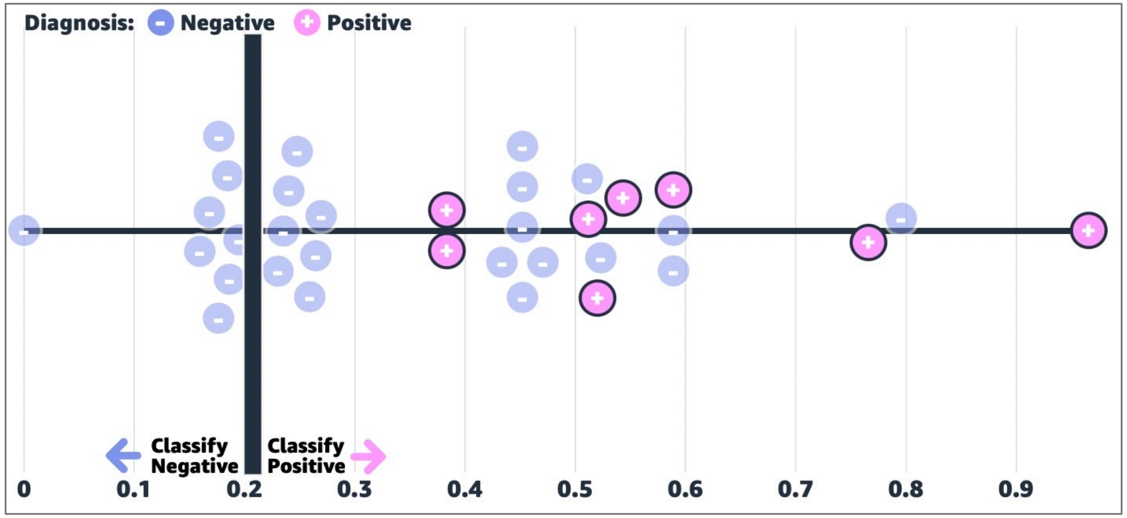
$$Recall = \frac{8}{8 + 0}$$

$$Recall \approx 100\%$$



Recall (Sensitivity or True Positive Rate)

- It measures how many of the actual positive instances we were able to correctly predict (or *recall*).
- It is important when we believe False Negatives **FN** are more important than False Positives **FP** (e.g. our problem of cancer detection).



Predicted values

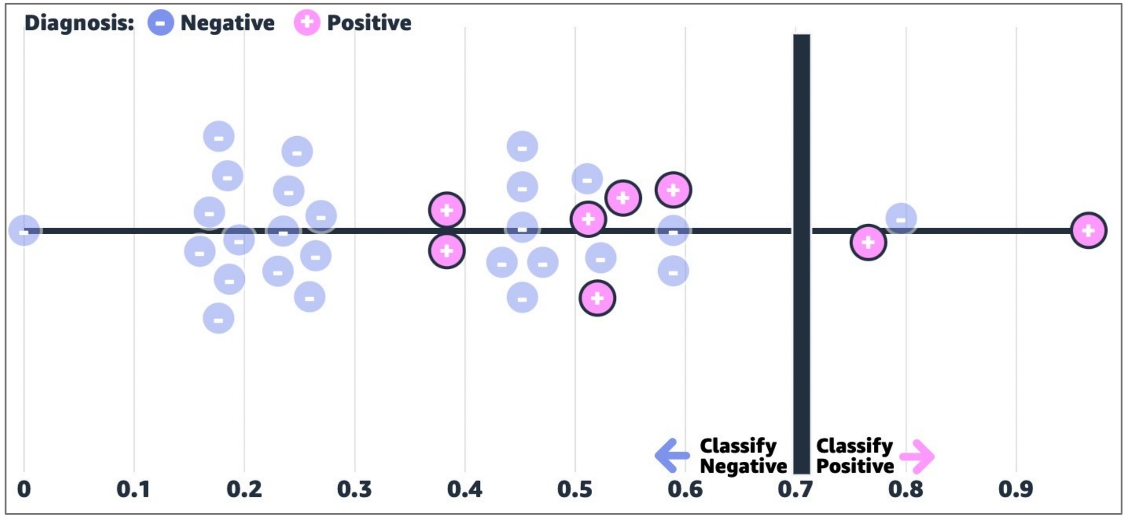
Actual values	Positive	Negative
Positive	**TP** 8	**FN** 0
Negative	**FP** 18	**TN** 8

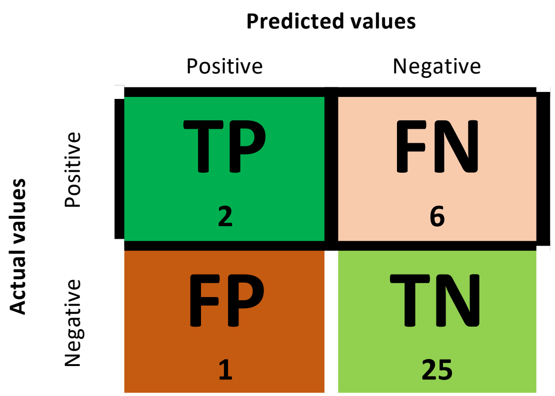
Recall (Sensitivity or True Positive Rate)

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

$$Recall = \frac{2}{2 + 6}$$

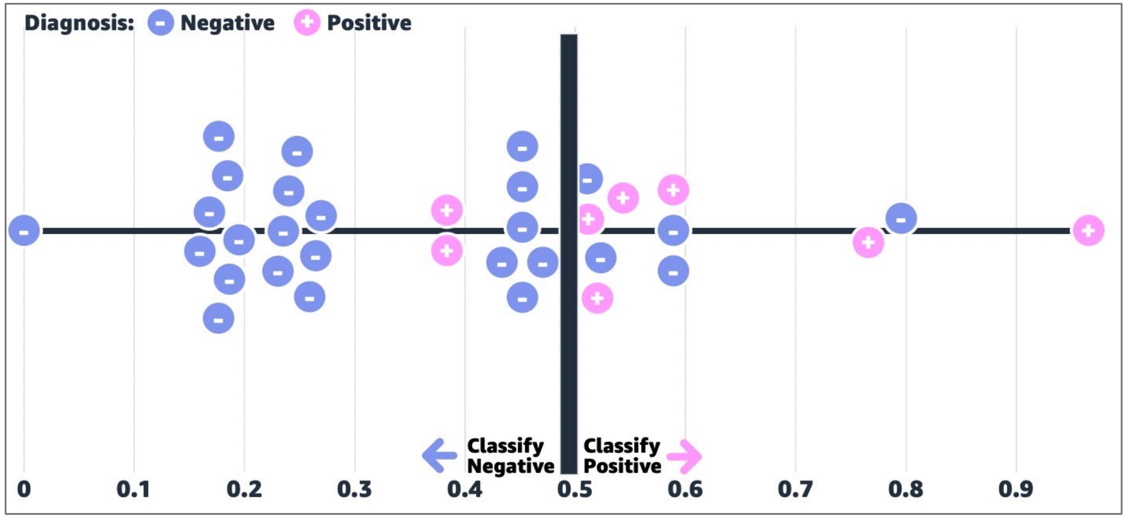
$$Recall \approx 25\%$$





Tradeoff

Ideally, our model would have both perfect precision *and* perfect recall. However, in practice there often exists a tradeoff between the two

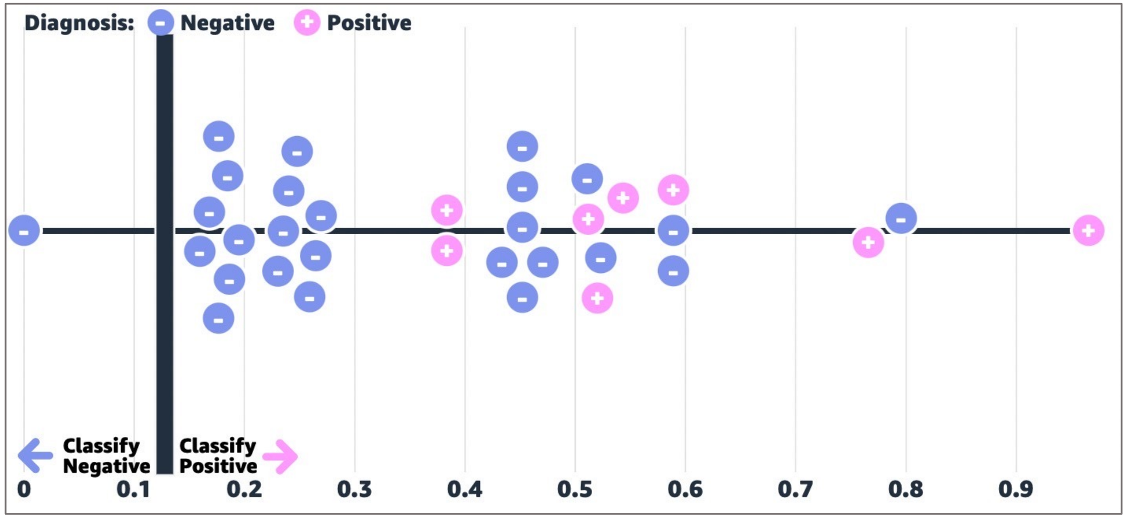


Precision: 55%
Recall: 75%

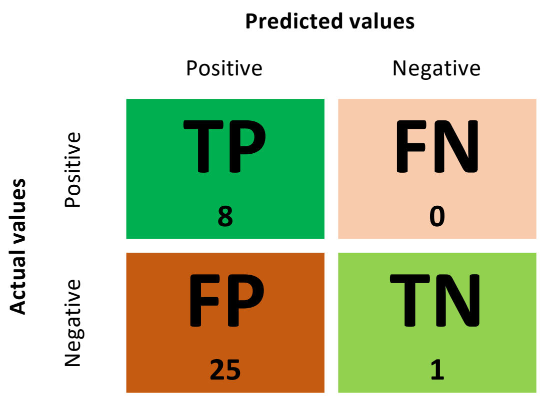
		Predicted values: Positive	Predicted values: Negative
Actual values	Positive	TP 6	FN 2
	Negative	FP 5	TN 21

Tradeoff

Ideally, our model would have both perfect precision *and* perfect recall. However, in practice there often exists a tradeoff between the two

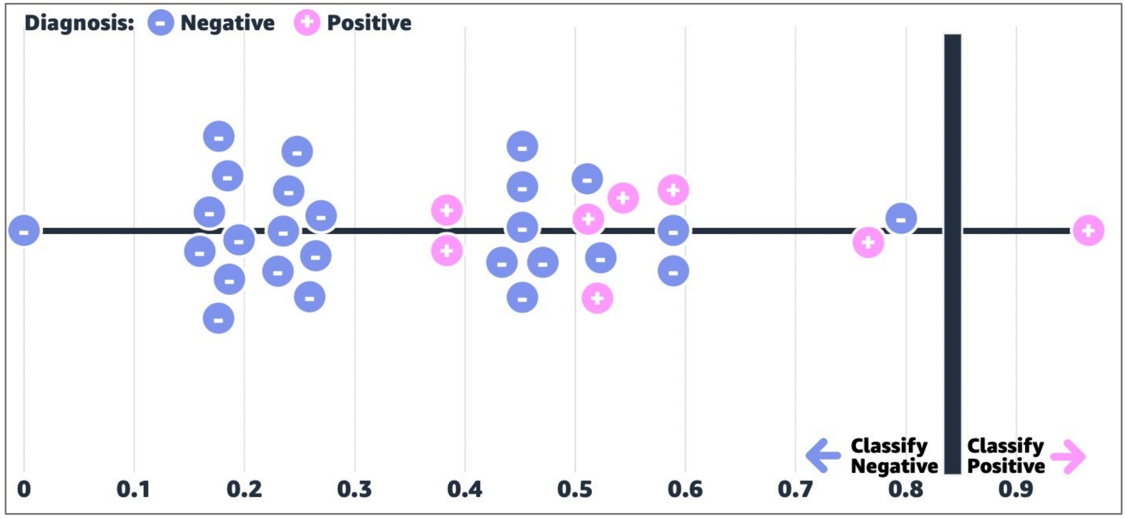


Precision: 24%
Recall: 100%



Tradeoff

Ideally, our model would have both perfect precision *and* perfect recall. However, in practice there often exists a tradeoff between the two



Precision: 100%
Recall: 13%

Predicted values

Actual values		Positive	Negative
	Positive	**TP** 1	**FN** 7
	Negative	**FP** 0	**TN** 26

Tradeoff

It's important to understand the problem that you're trying to solve and any inherent consequences of favoring False Positives **FP** over False Negatives **FN** (or vice versa).

In our example:

- A **model with high recall** will identify most people that have cancer (*true positives*, potentially saving lives. However, this comes at the cost of false positives, —misdiagnosing healthy people as sick—which can lead to unnecessary and harmful treatments like chemotherapy.
- A **model optimized for precision** produces highly confident predictions (i.e., if the model says someone has cancer, it is likely true). But it may miss some actual cancer cases (false negatives), which could result in undiagnosed patients and potentially fatal outcomes.
- Since **false negatives** in this context can lead to death, the classification threshold should likely be adjusted to **maximize recall**, even at the expense of precision.

F1-Score (F-Measure)

- It is a single performance metric that takes both **precision** and **recall** into account.
- It gives equal importance to precision and recall
- It's calculated by taking the harmonic mean of the two metrics

$$(bad) \leq F1 \leq 1(good)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- **F1** low when one or both of the Precision and Recall are low
- **F1** high when both Precision and Recall are high
- **F1** is a great way to compare the performance of multiple classifiers. When choosing between multiple models, all with varying values of precision and/or recall, it may be used to determine which one produces the 'best' results for the problem at hand. For this reason, it's often used in practice as a metric by which to rank models by performance

F1-Score (F-Measure)

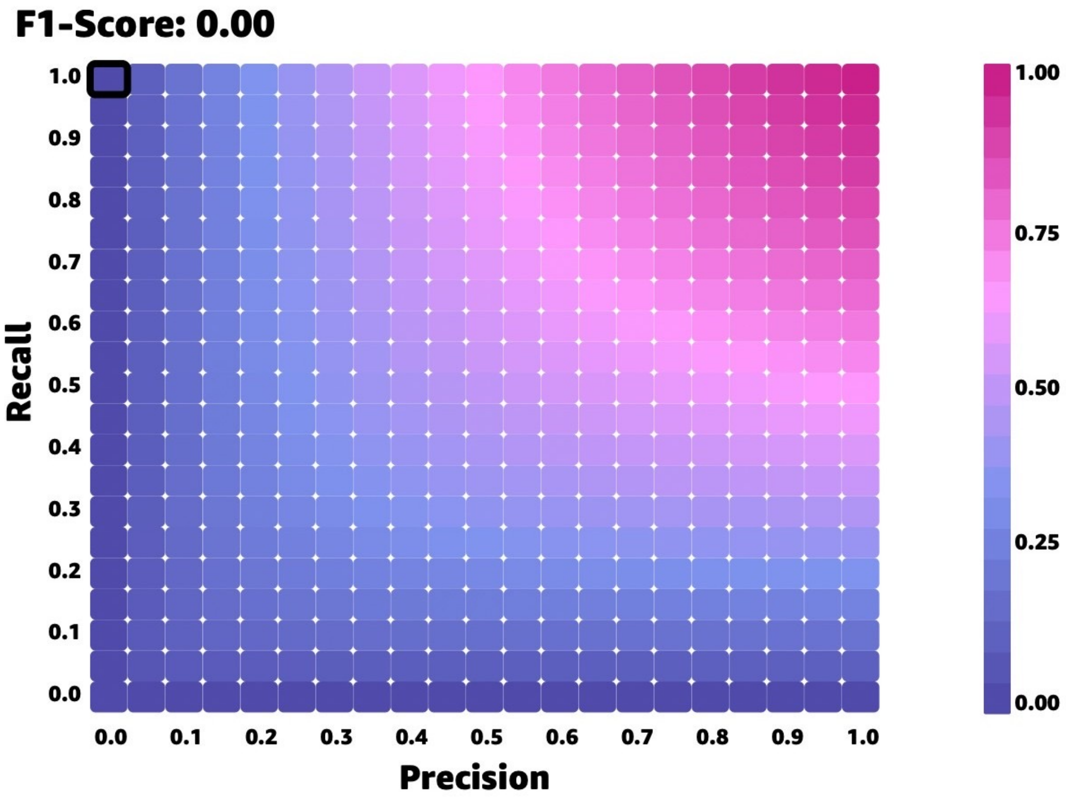
Precision=***0.00***

Recall=***1.00***

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$F1 = 2 * \frac{0.00 * 1.00}{0.00 + 1.00}$$

$$F1 = 0.00$$



F1-Score (F-Measure)

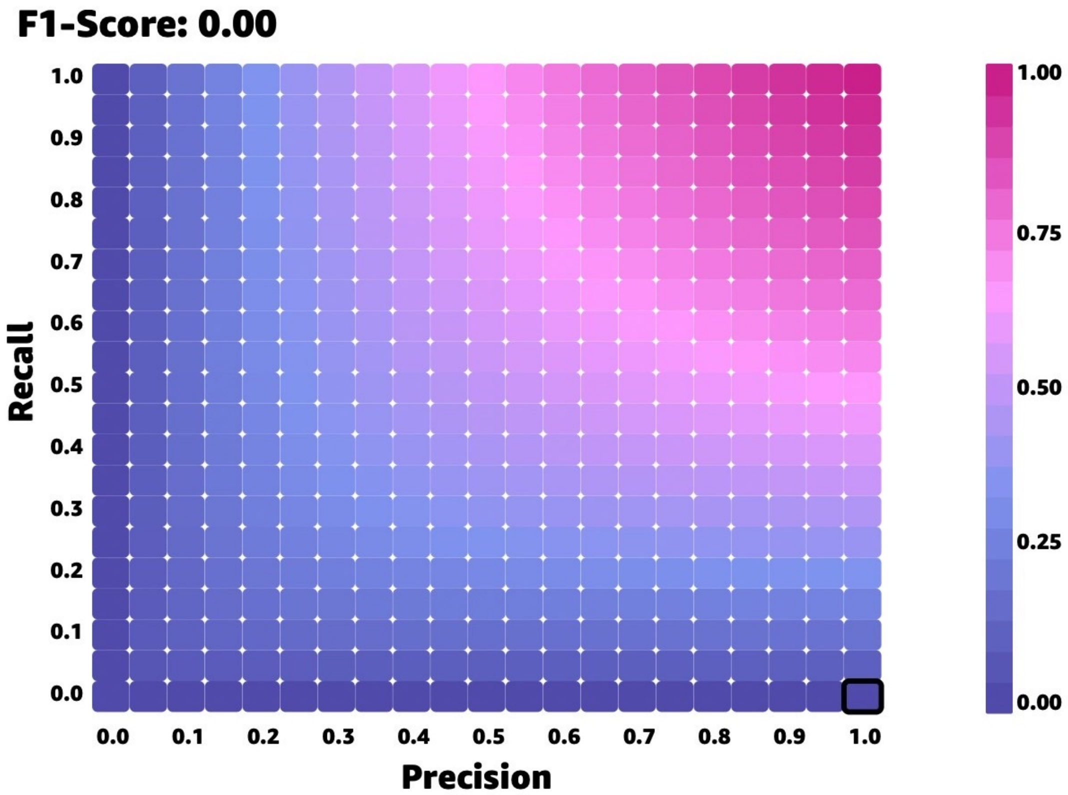
*Precision=**1.00***

*Recall=**0.00***

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$F1 = 2 * \frac{1.00 * 0.00}{1.00 + 0.00}$$

$$F1 = 0.00$$



F1-Score (F-Measure)

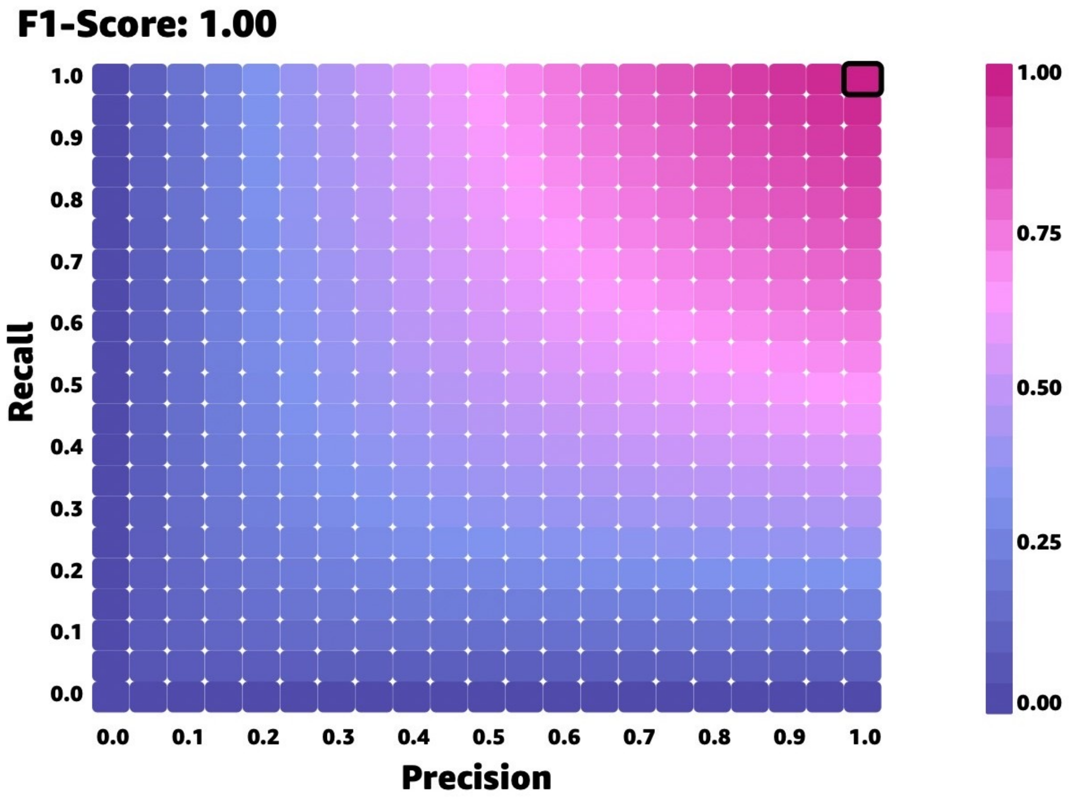
Precision=***1.00***

Recall=***1.00***

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$F1 = 2 * \frac{1.00 * 1.00}{1.00 + 1.00}$$

$$F1 = 1.00$$



F1-Score (F-Measure)

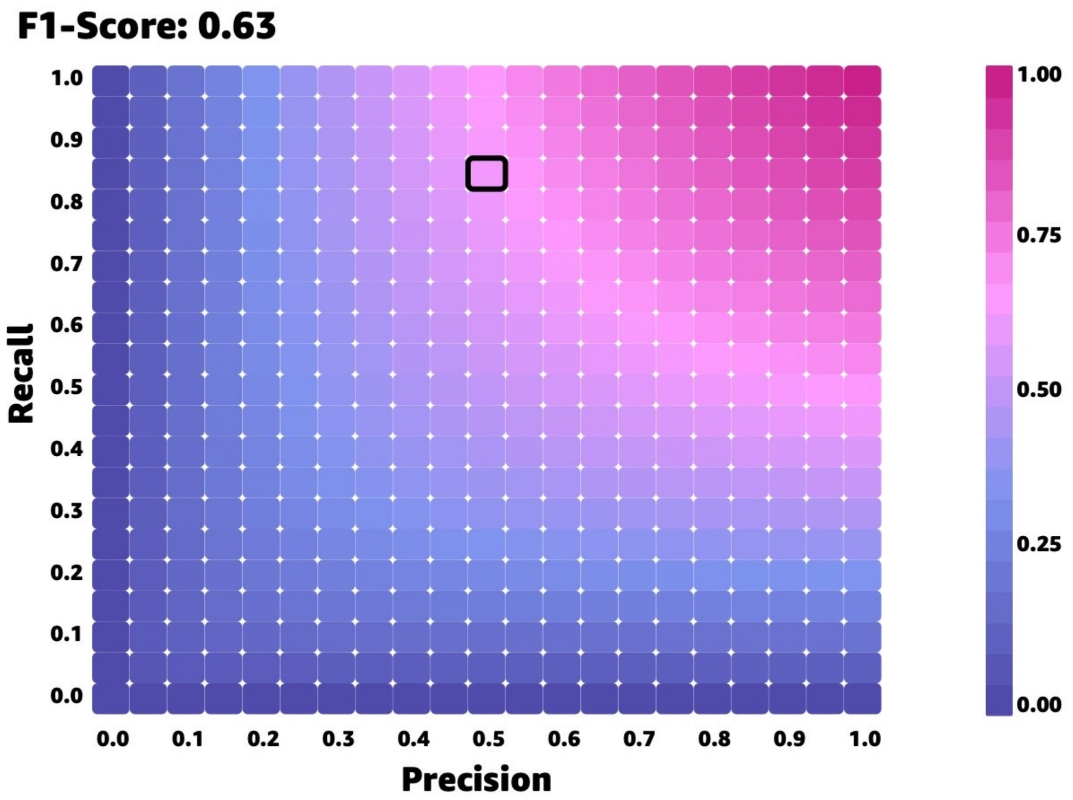
In our example:

*Since False Negatives **FN** result in death, our classification threshold would likely be set to optimize recall over precision*

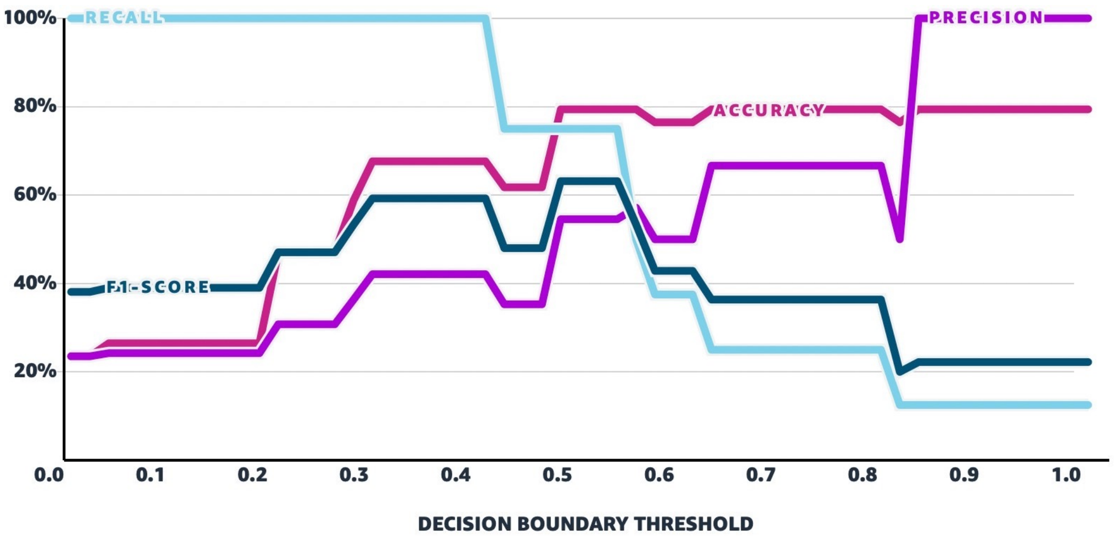
$Precision$ = 0.50

$Recall$ = 0.85

F1= 0.63



In our example



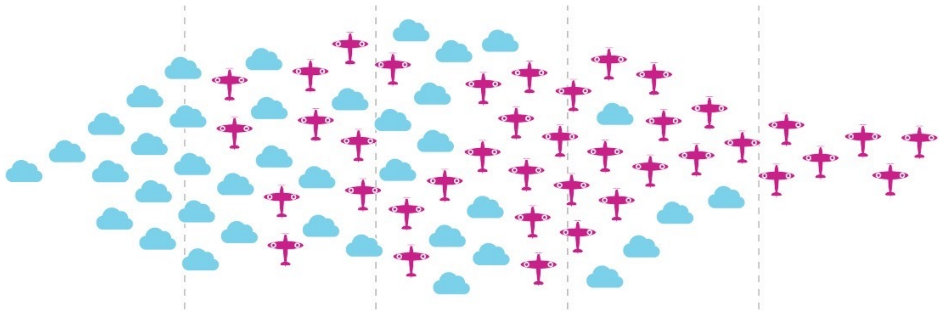
- **Recall**: lower classification thresholds yield perfect recall at the cost of low precision
- **Precision**: higher classification thresholds yield perfect precision at the cost of low recall
- **F1-score**: F1-Score is maximized when both Precision and Recall perform well relatively close to each other, and is low otherwise
- **Accuracy**: Accuracy is also its highest at the maximum point for the F1-Score, but note that it barely changes thereafter

ROC and AUC

History:

ROC curves were first employed during World War 2 to analyze radar signals:

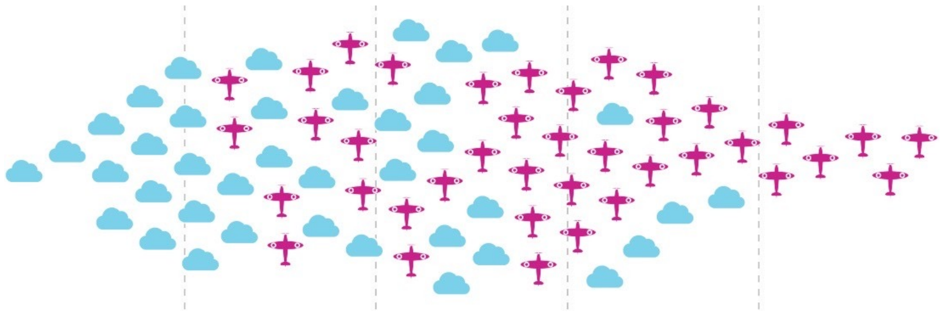
- After missing the Japanese aircraft that carried out the attack on Pearl Harbor, the US wanted their radar receiver operators to better identify aircraft from *signal noise* (e.g. clouds).



ROC and AUC

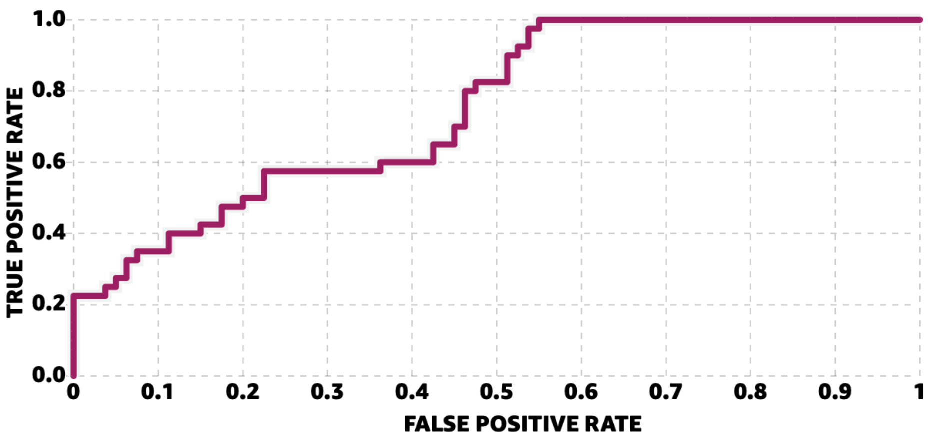
History:

- The operator's ability to identify as many true positives as possible while minimizing false positives was named the *Receiver Operating Characteristic*, and the curve analyzing their predictive abilities was called the ROC Curve.
- ROC curves are used in a number of contexts, including clinical settings (to assess the diagnostic accuracy of a test) and machine learning.



Receiver Operating Characteristic Curve (ROC)

- ROC Curves analyze the predictive power of a classifier
- ROC Curves provide a visual way to observe how changes in our model's classification thresholds affect our model's performance
- The curves allow us to select for classification thresholds that allow our model to identify as many **true positives** as possible while minimizing **false positives**.



Receiver Operating Characteristic Curve (ROC)

True-Positives Rate (TPR or Sensitivity)

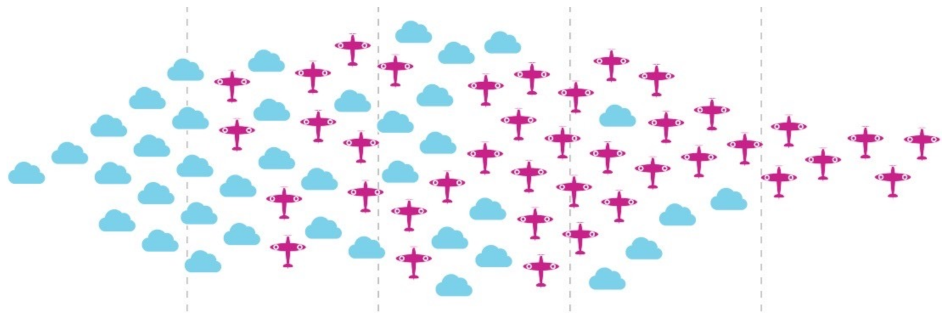
The probability that a positive sample is correctly predicted in the positive class.

- E.g., the percentage of radar signals predicted to be airplanes that actually are airplanes

False-Positives Rate (FPR or Specificity)

The probability that a negative sample is incorrectly predicted in the positive class (1 - Sensitivity)

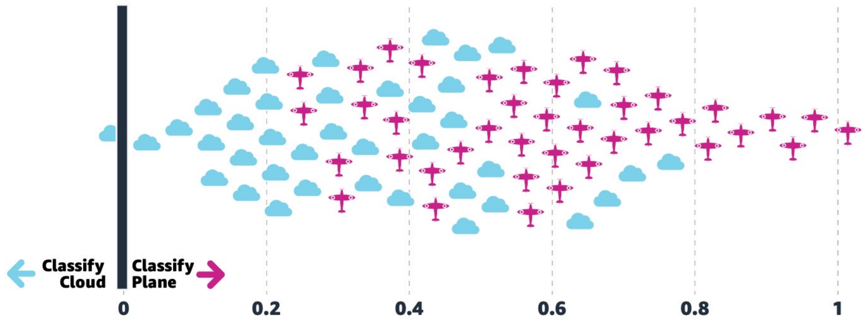
- E.g., the percentage of radar signals predicted to be airplanes that actually are *not* airplanes



Receiver Operating Characteristic Curve (ROC)

Our First Threshold

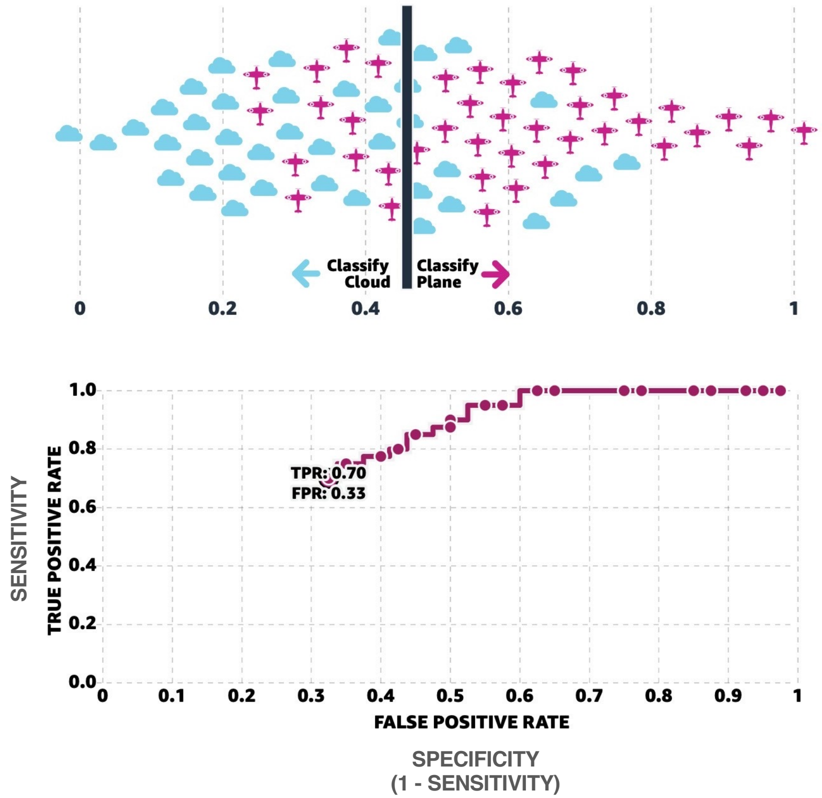
- We'll start with our model's classification threshold at 0, so anything with a probability greater-than-or-equal-to zero of being an airplane, we'll classify as an airplane.
    - everything will be classified as an airplane!
- While this model will correctly classify every airplane as an airplane (yielding a perfect **TPR=1**), it will also incorrectly classify every radar noise as an airplane (giving us the worst possible **FPR=1**).



Receiver Operating Characteristic Curve (ROC)

Some New Thresholds

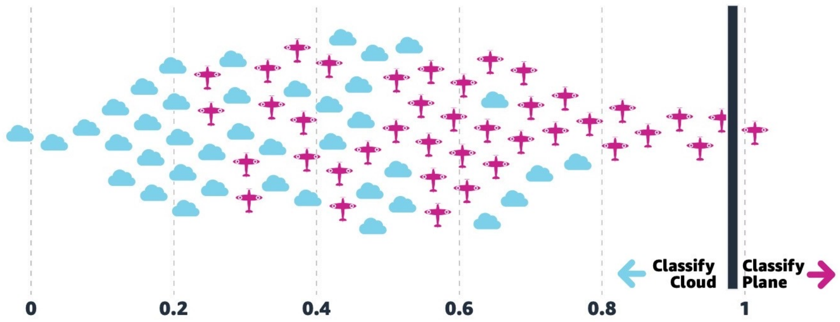
- We'll move our threshold more and more to the right, increasing the threshold at which we classify a radar signal as an airplane. To assess the performance, we calculate the TPR and FPR for each new threshold choice, and plot them below.

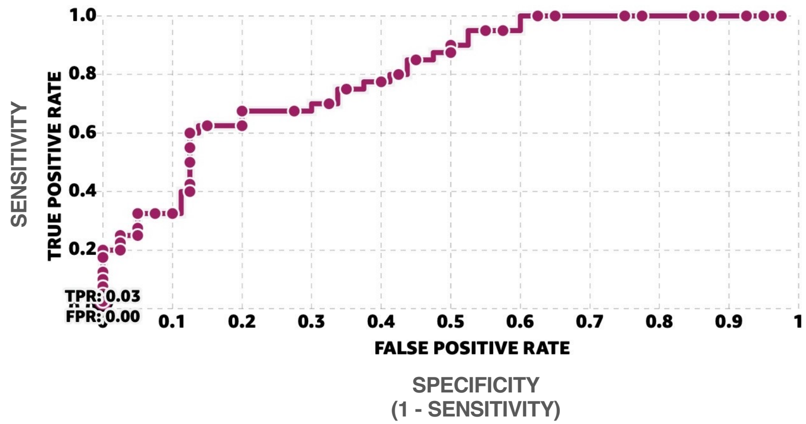


Receiver Operating Characteristic Curve (ROC)

And More Thresholds

- Recall that our goal is to find the classification threshold that best maximizes true positives while minimizing false positives.
  - To find that threshold, we'll have to try all possible values for our threshold!
  - Continue increasing our threshold until we can't any further (i.e. moving it all the way to the right), logging each TPR and FPR along the way.



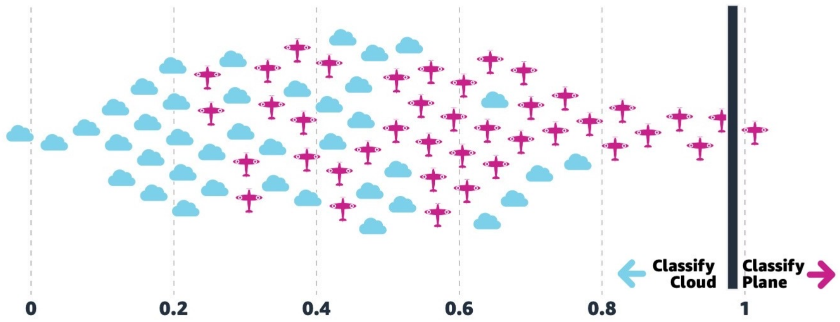


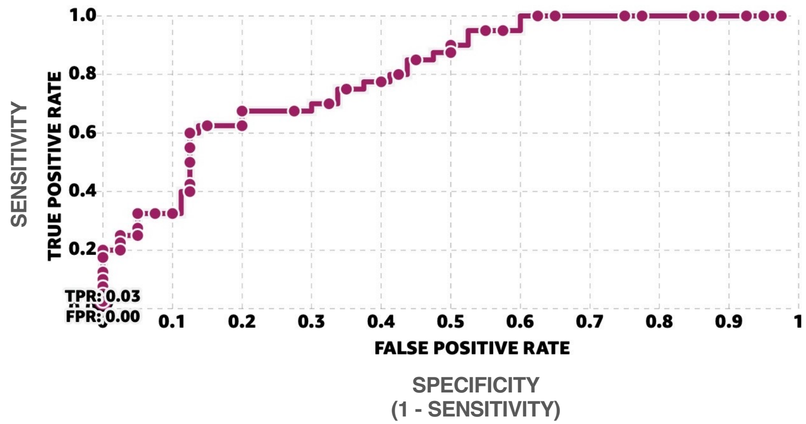
Receiver Operating Characteristic Curve (ROC)

The ROC gives us a convenient visual of the performance of our classifier.

- It allows us to understand how that performance changes as a function of the model's classification threshold.

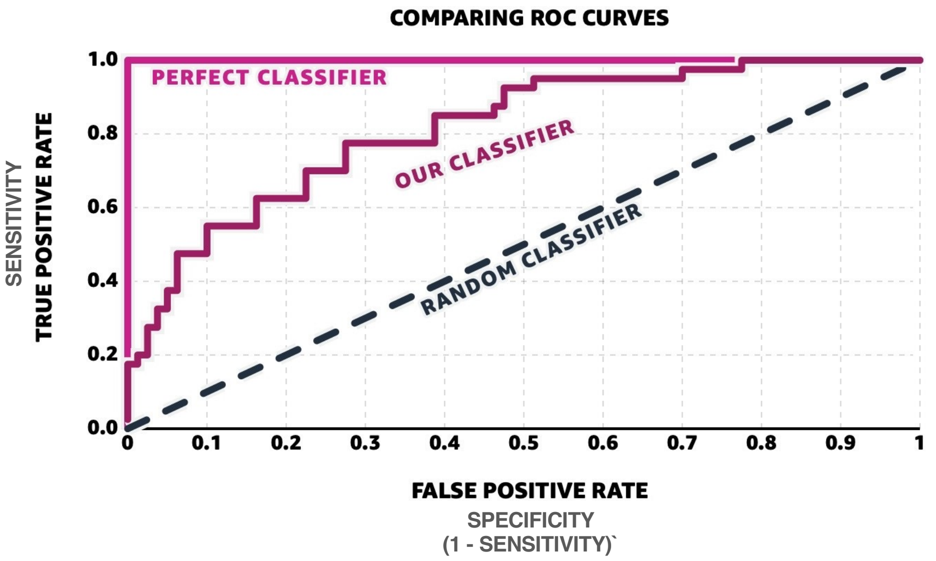
Which threshold would you choose?





Receiver Operating Characteristic Curve (ROC)

- It gives us an overview of our model's performance
  - **Perfect classifiers**: hugs along the outer-left and top of the chart. Classifiers will always have a **TPR=1**, regardless of the **FPR**
  - **Diagonal line**: implies **TPR=FRP** for every classification threshold (the classifier is just making random guesses)
- It gives us an easy visual to compare the performance of different classifiers to one another
  - Curves that fall above the ROC Curve of a random classifier (the diagonal line) are good or decent.
  - **Better**: Classifiers closer to the curve of the elusive perfect classifier
  - **Worse**: Anything below the diagonal line has worse performance than random guessing



Area Under the ROC Curve (AUC, AUROC)

- The AUC is the area under the ROC Curve
- It is a measure of performance of our classifier, independent of the threshold chosen

$$(bad) \leq AUC \leq 1(good)$$

$AUC = 1;\ perfect\ classifier$

$AUC = 0.5;\ random\ classifier$

$AUC < 0.5;\ poor\ performance$

$0.5 \leq AUC \leq 1;\ good\ performance$