
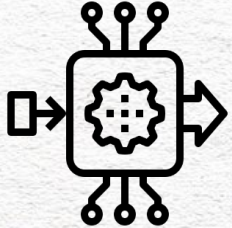



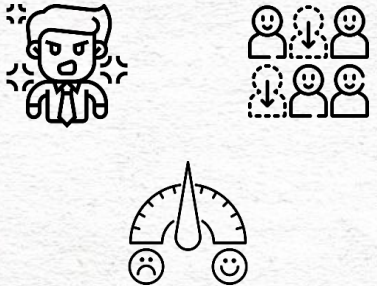



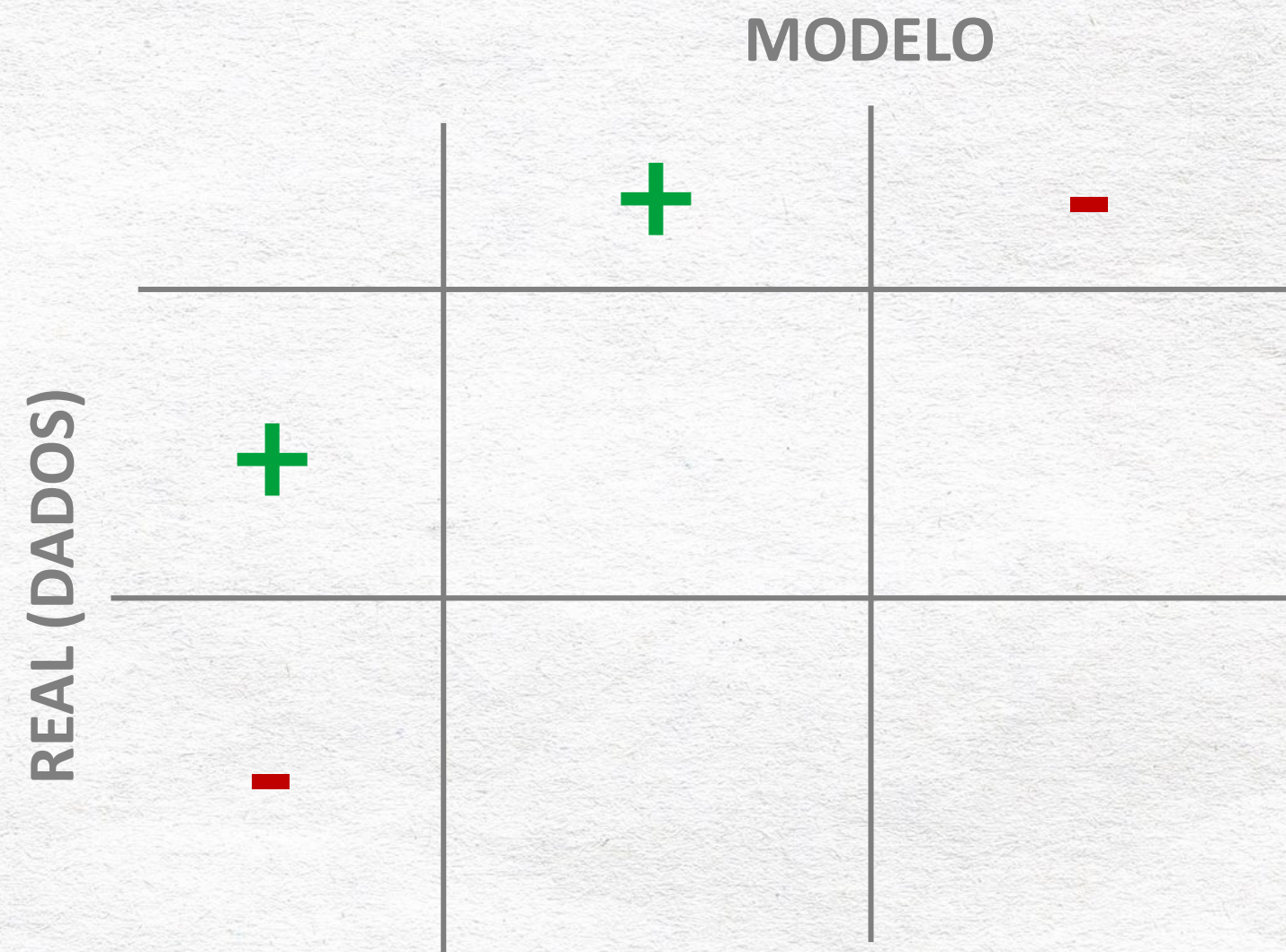
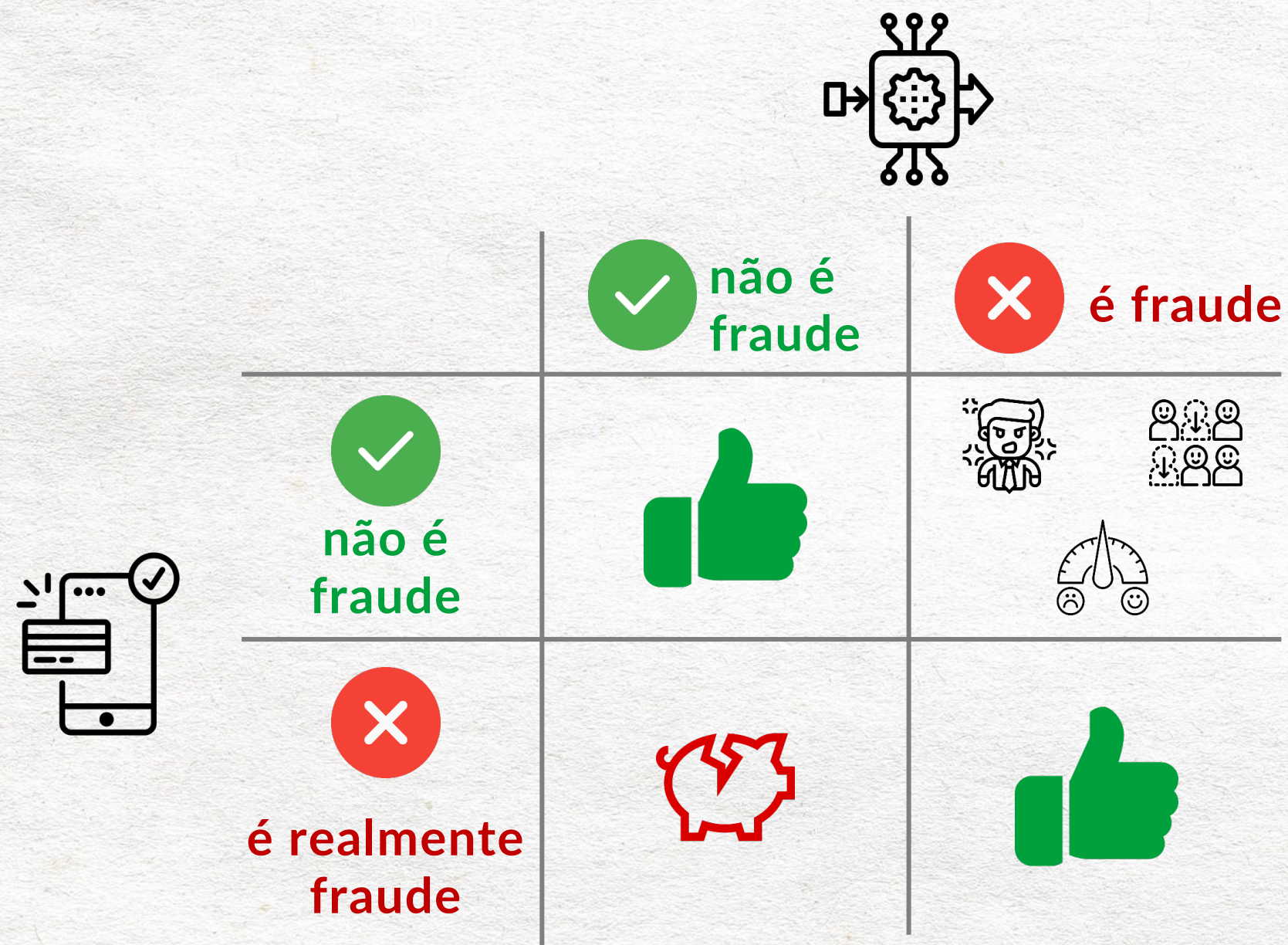


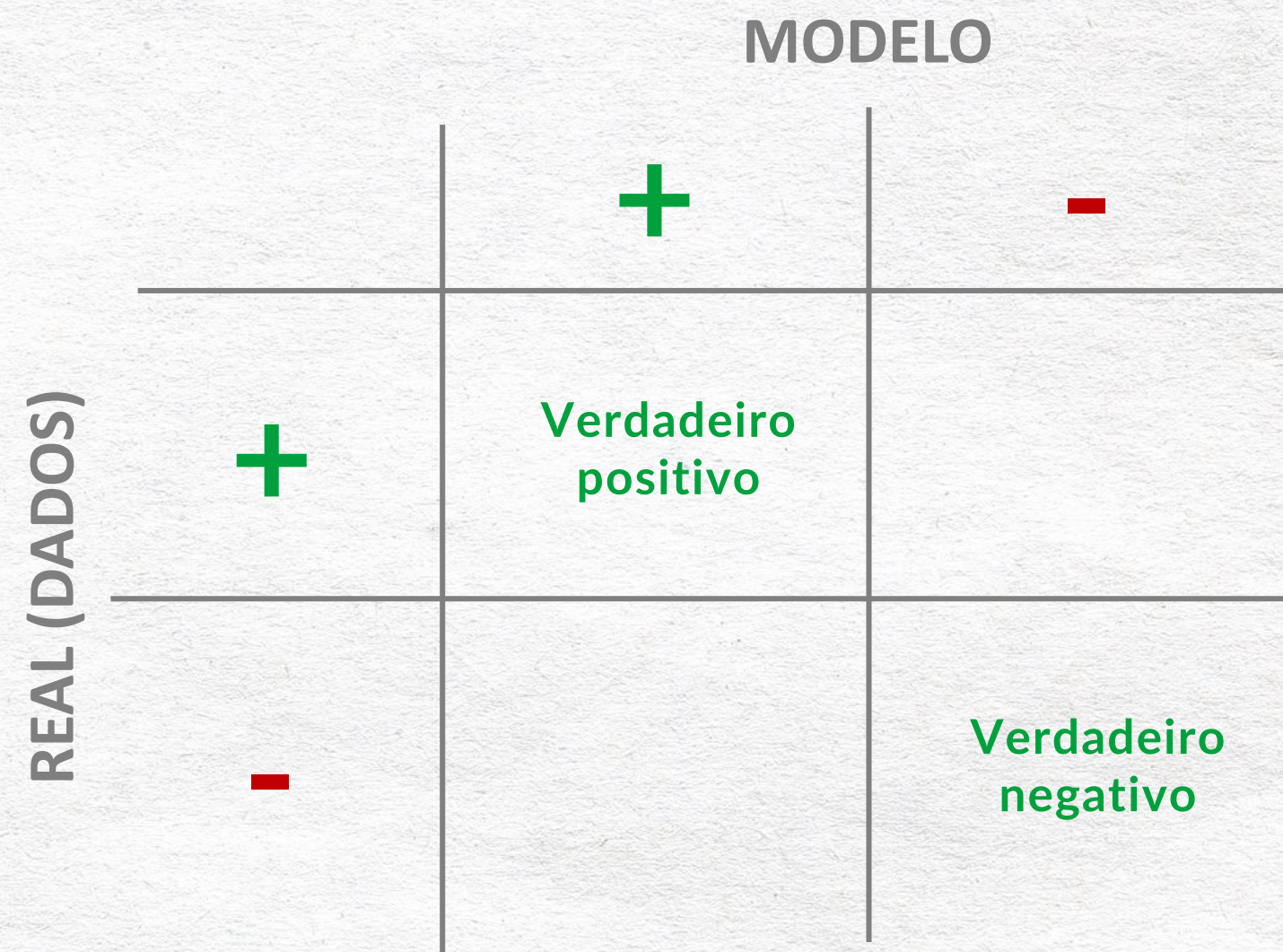
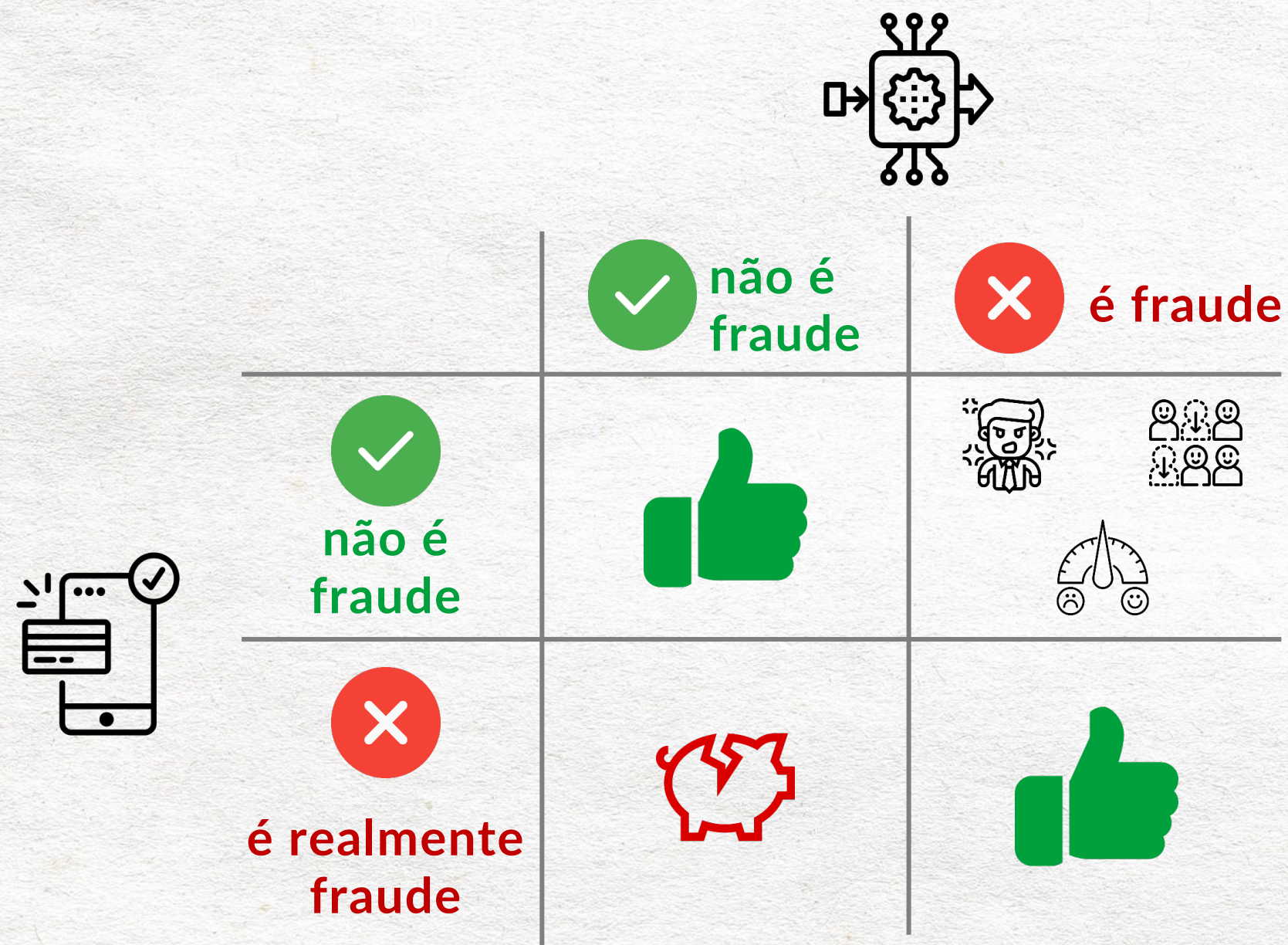
AVALIANDO O MODELO DE CLASSIFICAÇÃO

		
	 não é fraude	 é fraude
		
 é realmente fraude		

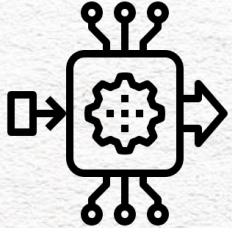

AVALIANDO O MODELO DE CLASSIFICAÇÃO













AVALIANDO O MODELO DE CLASSIFICAÇÃO







AVALIANDO O MODELO DE CLASSIFICAÇÃO

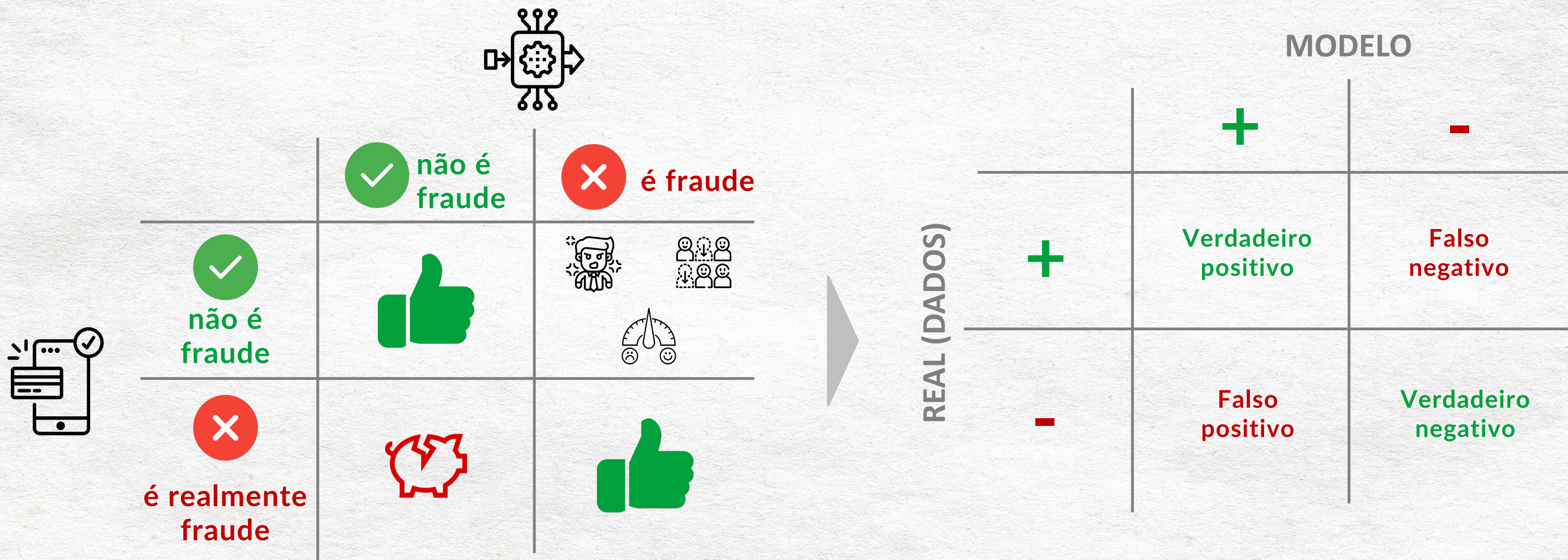


	 não é fraude	 é fraude
 não é fraude		  
 é realmente fraude		



		MODELO	
			
REAL (DADOS)		Verdadeiro positivo	Falso negativo
			Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO



AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+','+','+']
y_pred = ['-','-','+','+','-','-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

array([[1, 2],
       [1, 2]], dtype=int64)
```

		MODELO	
		+	-
REAL (DADOS)	+	+ Verdadeiro positivo	- Falso negativo
	-	- Falso positivo	- Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+','+','+']
y_pred = ['-','-','+','+','-','-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

array([[1, 2],
       [1, 2]], dtype=int64)
```

		MODELO	
		+	-
REAL (DADOS)	+	Verdadeiro positivo	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+', '+', '+']
y_pred = ['-','-','+', '+', '-','-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

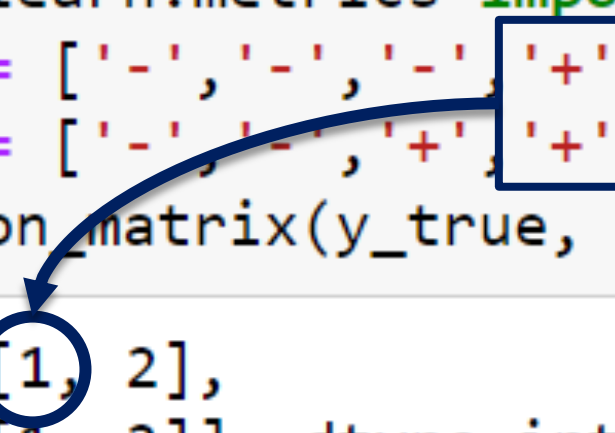
array([[1, 2],
       [1, 2]], dtype=int64)
```

		MODELO	
		+	-
REAL (DADOS)	+	Verdadeiro positivo	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+', '+', '+']
y_pred = ['-','-','+', '+', '-', '-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

array([[1, 2],
       [1, 2]], dtype=int64)
```



		MODELO	
		+	-
REAL (DADOS)	+	Verdadeiro positivo	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+','+','+']
y_pred = ['-','-','+','+','-','-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

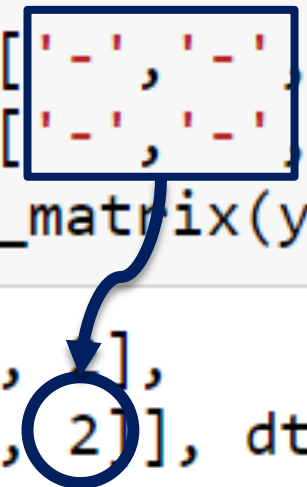
array([[1, 2],
       [1, 2]], dtype=int64)
```

		MODELO	
		+	-
REAL (DADOS)	+	Verdadeiro positivo	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-', '-', '-', '+', '+', '+']
y_pred = ['-', '-', '+', '+', '-', '-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

array([[1, 1],
       [1, 2]], dtype=int64)
```



		MODELO	
		+	-
REAL (DADOS)	+	Verdadeiro positivo	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+','+','+']
y_pred = ['-','-','+','+','-','-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

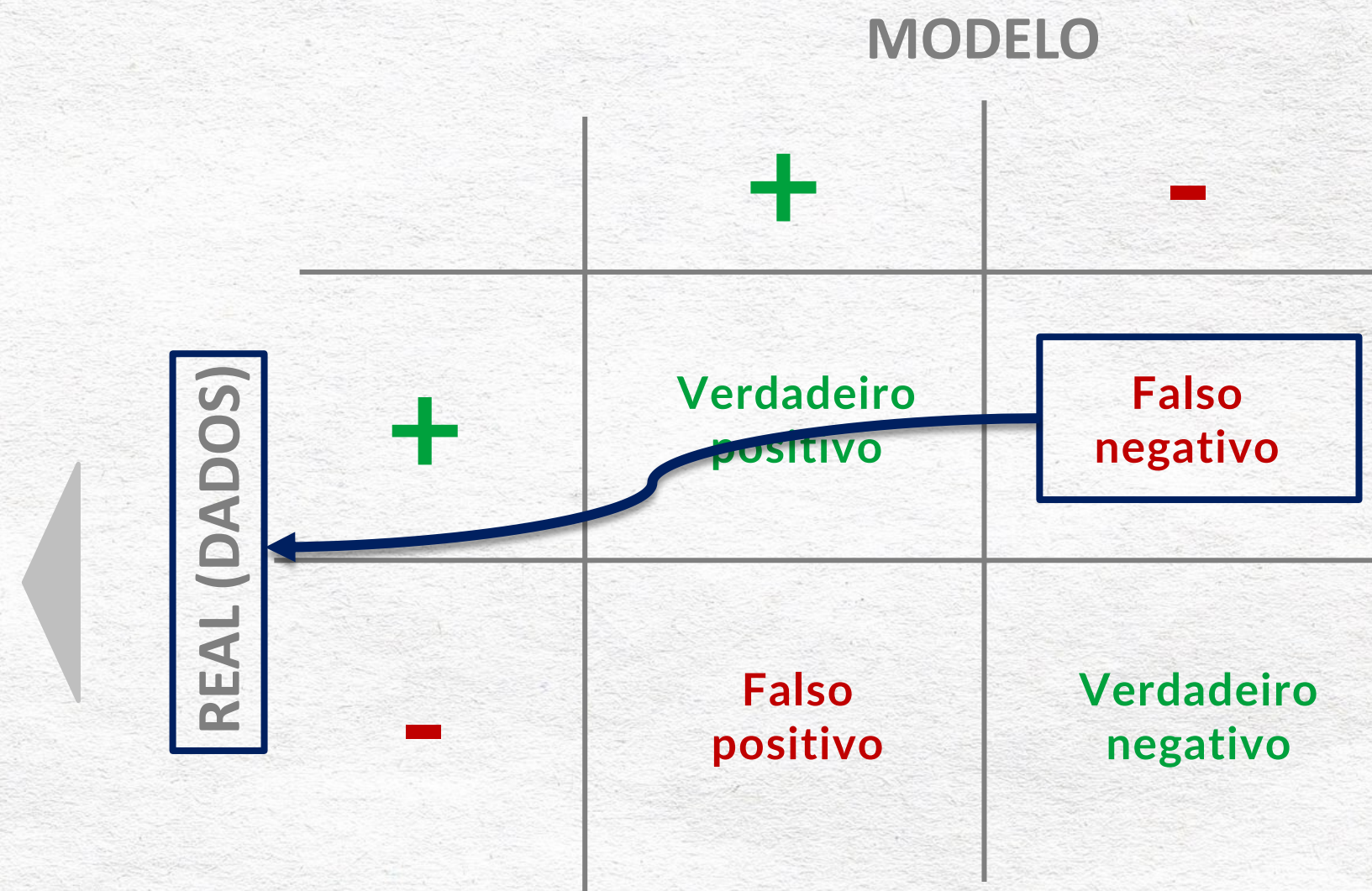
array([[1, 2],
       [1, 2]], dtype=int64)
```

		MODELO	
		+	-
REAL (DADOS)	+	Verdadeiro positivo	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+', '+', '+']
y_pred = ['-','-','+', '+', '-', '-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

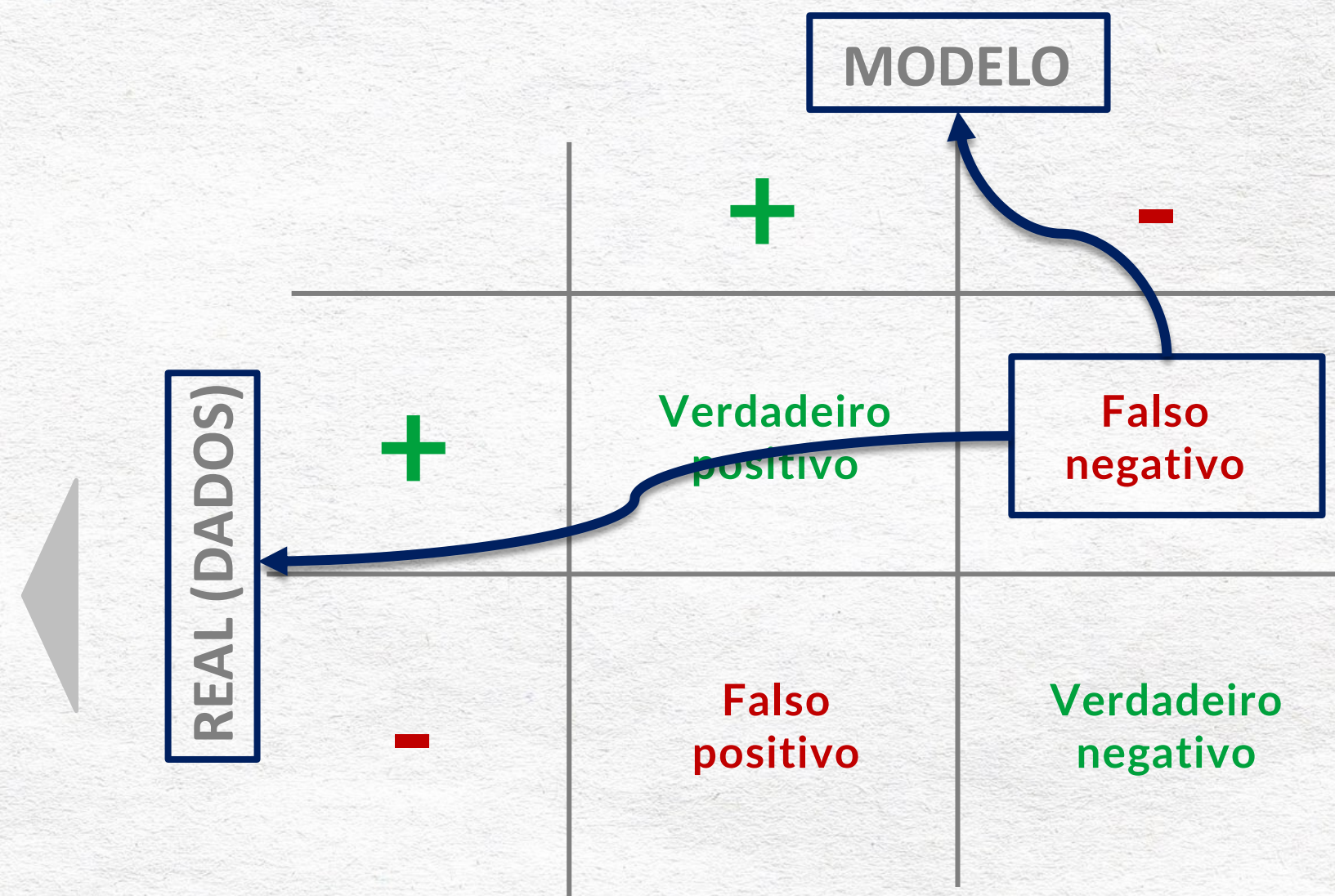
array([[1, 2],
       [1, 2]], dtype=int64)
```



AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+','+','+']
y_pred = ['-','-','+','+','-','-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

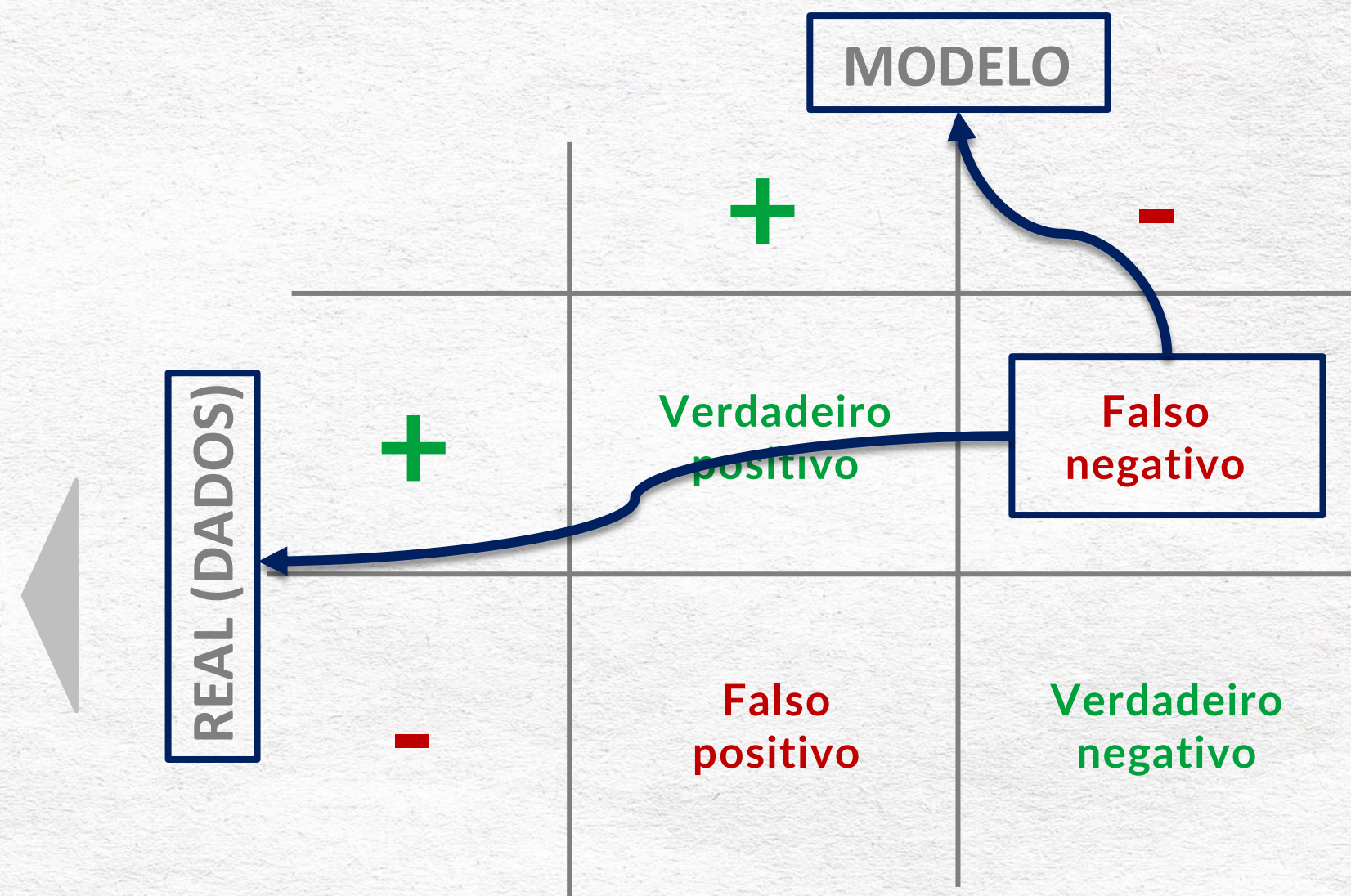

array([[1, 2],
       [1, 2]], dtype=int64)
```



AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+','+','+']
y_pred = ['-','-','+','+','-', '-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

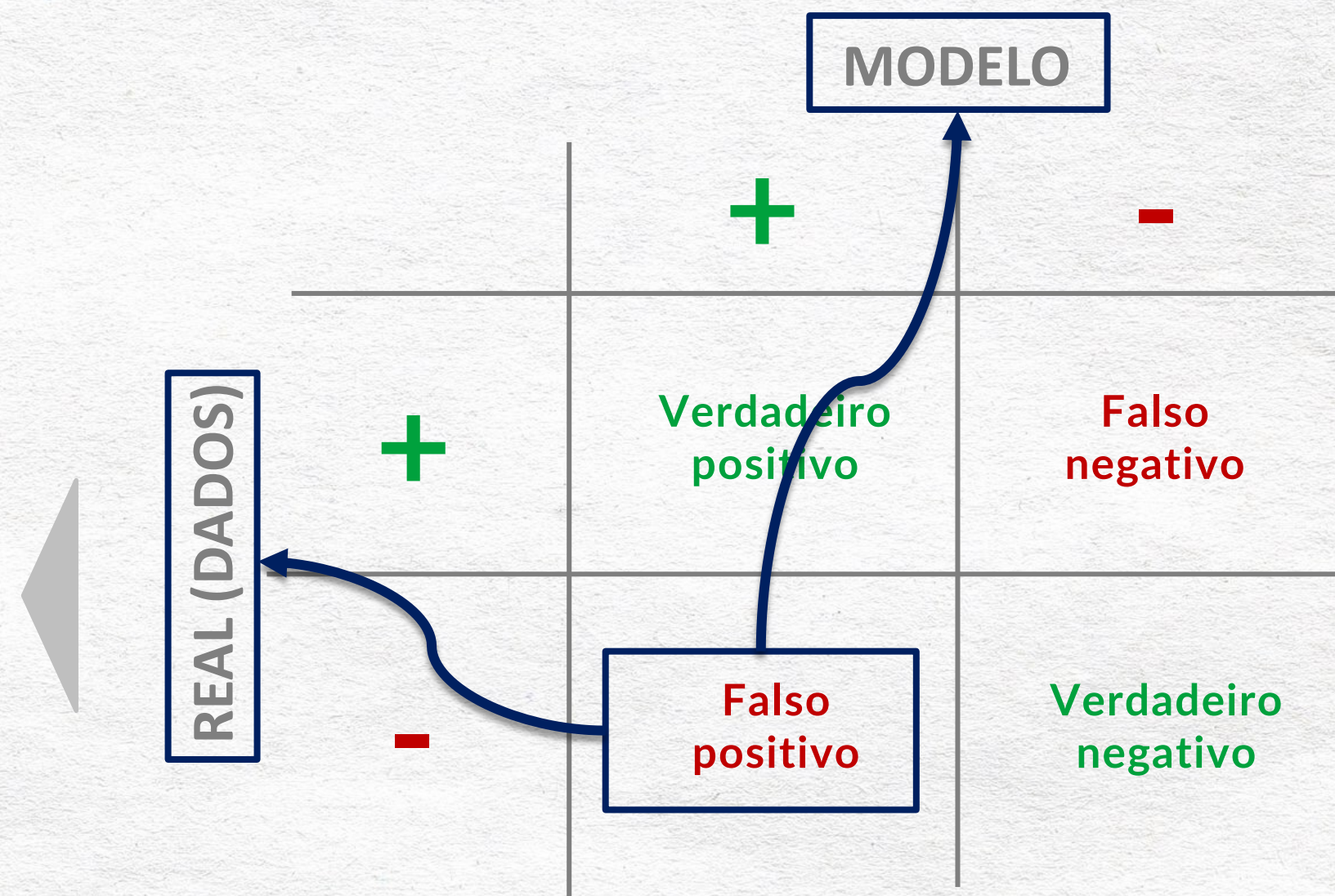
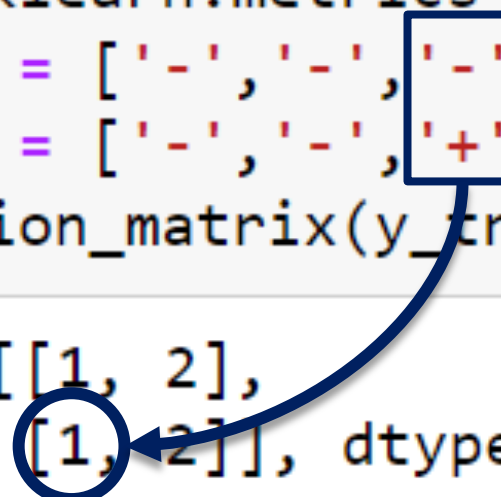
array([[1, 2],
       [1, 2]], dtype=int64)
```



AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-', '+', '+', '+']
y_pred = ['-','-','+', '+', '-', '-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

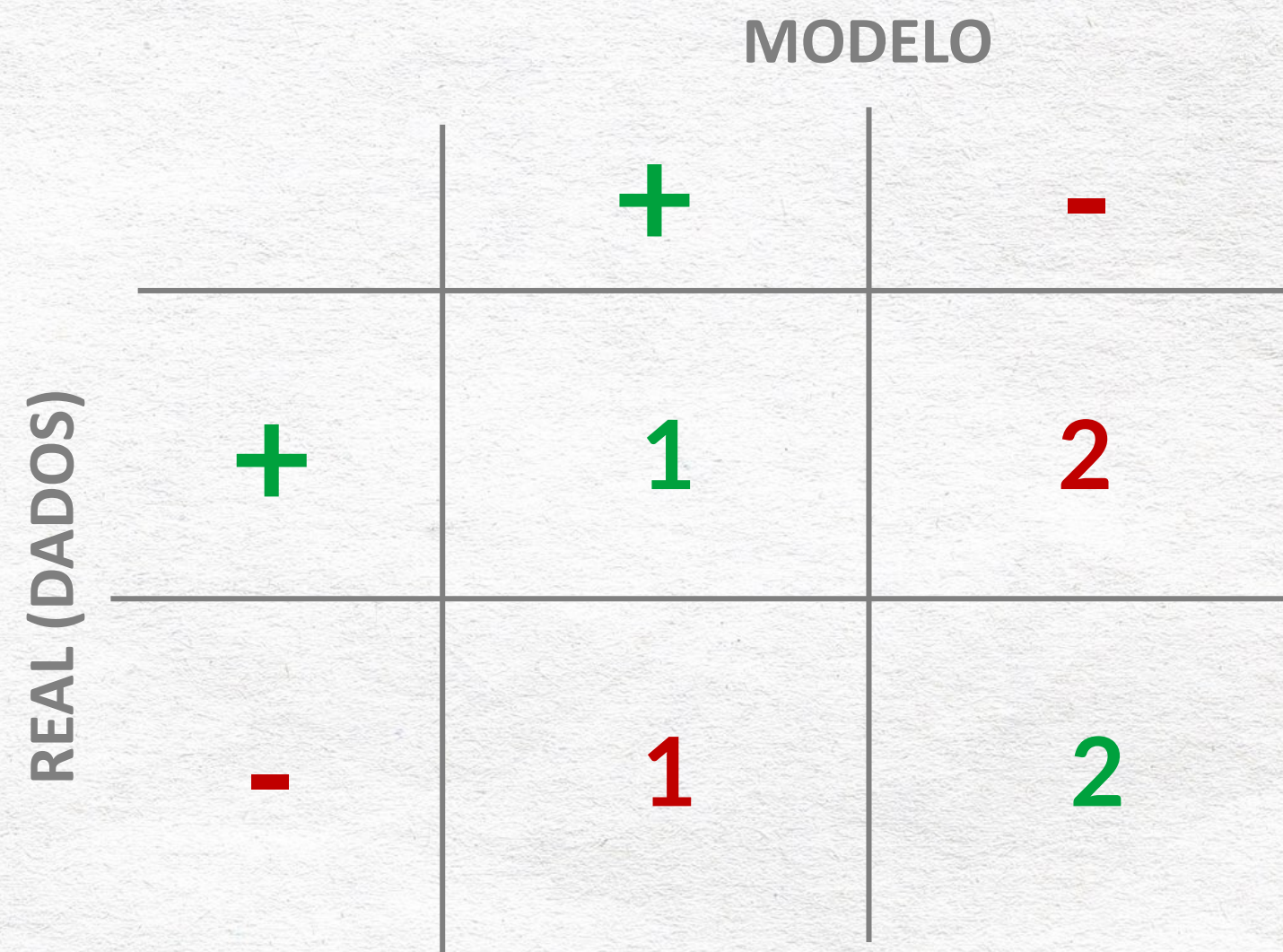
array([[1, 2],
       [1, 2]], dtype=int64)
```



AVALIANDO O MODELO DE CLASSIFICAÇÃO

```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+','+','+']
y_pred = ['-','-','+','+','-','-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

array([[1, 2],
       [1, 2]], dtype=int64)
```



		MODELO	
		+	-
REAL (DADOS)	+	1	2
	-	1	2

AVALIANDO O MODELO DE CLASSIFICAÇÃO

REAL (DADOS)	MODELO	
	-	+
-	-	+
+	+	



REAL (DADOS)	MODELO	
	+	-
+	Verdadeiro positivo	Falso negativo
-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

		MODELO	
REAL (DADOS)		-	+
	-	Verdadeiro	
	+	Verdadeiro	



		MODELO	
REAL (DADOS)		+	-
	+	Verdadeiro positivo	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

		MODELO	
		-	+
REAL (DADOS)	-	-	Verdadeiro negativo
	+	+	Verdadeiro positivo



		MODELO	
		+	-
REAL (DADOS)	+	+	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

		MODELO	
		-	+
REAL (DADOS)	-	Verdadeiro negativo	Falso
	+	Falso	Verdadeiro positivo



		MODELO	
		+	-
REAL (DADOS)	+	Verdadeiro positivo	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

		MODELO	
		-	+
REAL (DADOS)	-	Verdadeiro negativo	Falso positivo
	+	Falso	Verdadeiro positivo



		MODELO	
		+	-
REAL (DADOS)	+	Verdadeiro positivo	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

		MODELO	
		-	+
REAL (DADOS)	-	Verdadeiro negativo	Falso positivo
	+	Falso negativo	Verdadeiro positivo



		MODELO	
		+	-
REAL (DADOS)	+	Verdadeiro positivo	Falso negativo
	-	Falso positivo	Verdadeiro negativo

AVALIANDO O MODELO DE CLASSIFICAÇÃO

REAL (DADOS)	MODELO	
	0	1
0	Verdadeiro negativo	Falso positivo

```
# Dados
y_true = [0,0,0,1,1,1]
y_pred = [0,0,1,1,0,0]

confusion_matrix(y_true, y_pred)

array([[2, 1],
       [2, 1]], dtype=int64)
```



REAL (DADOS)	MODELO	
	+	-
+	Verdadeiro positivo	Falso negativo

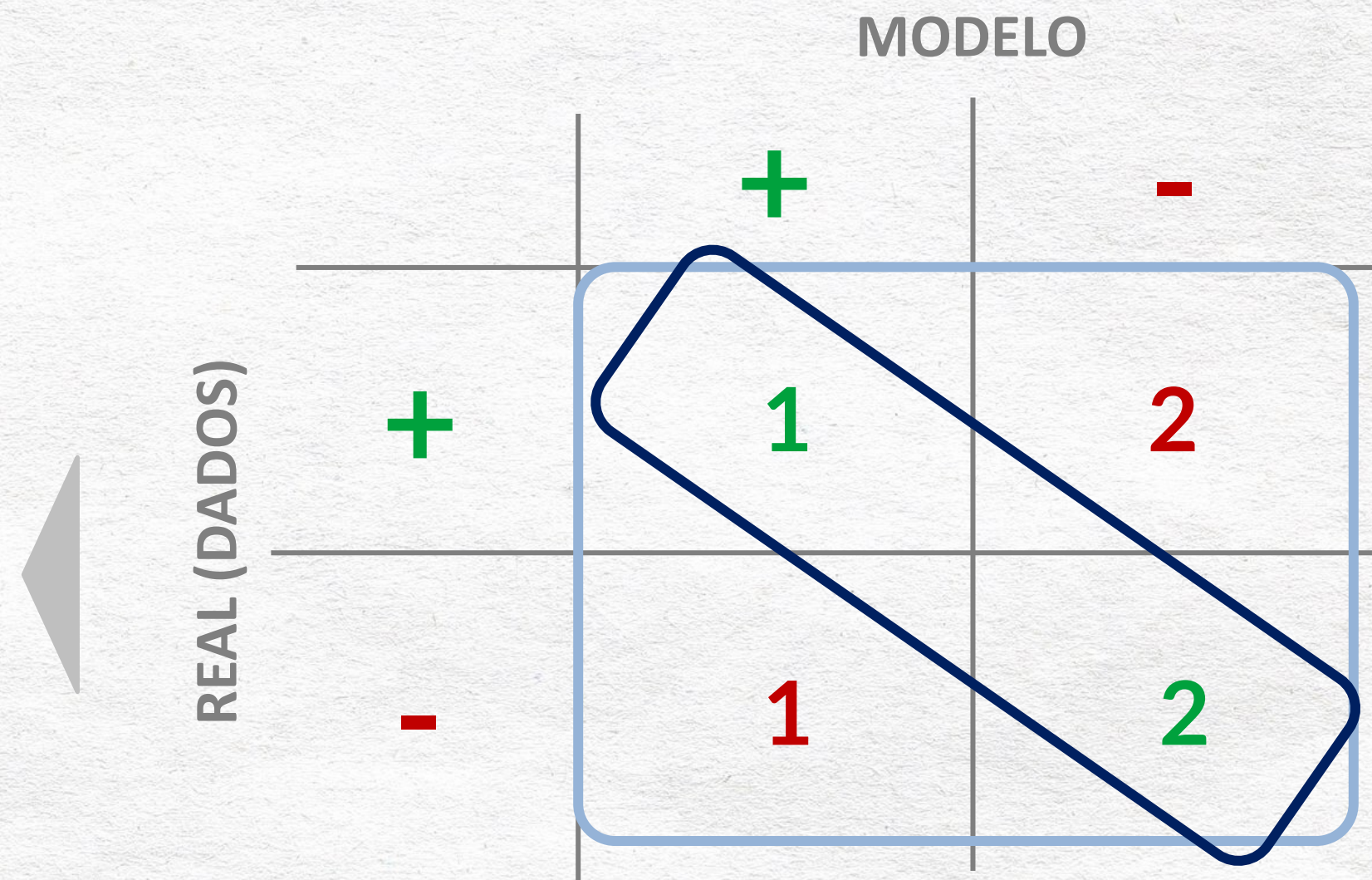
```
from sklearn.metrics import confusion_matrix
y_true = ['-','-','-','+','+','+']
y_pred = ['-','-','+','+','-','-']
confusion_matrix(y_true, y_pred, labels=['+', '-'])

array([[1, 2],
       [1, 2]], dtype=int64)
```


AVALIANDO O MODELO DE CLASSIFICAÇÃO

ACURÁCIA:

$$\text{Acurácia} = \frac{\text{Verdadeiro Positivo} + \text{Verdadeiro Negativo}}{\text{Total}}$$

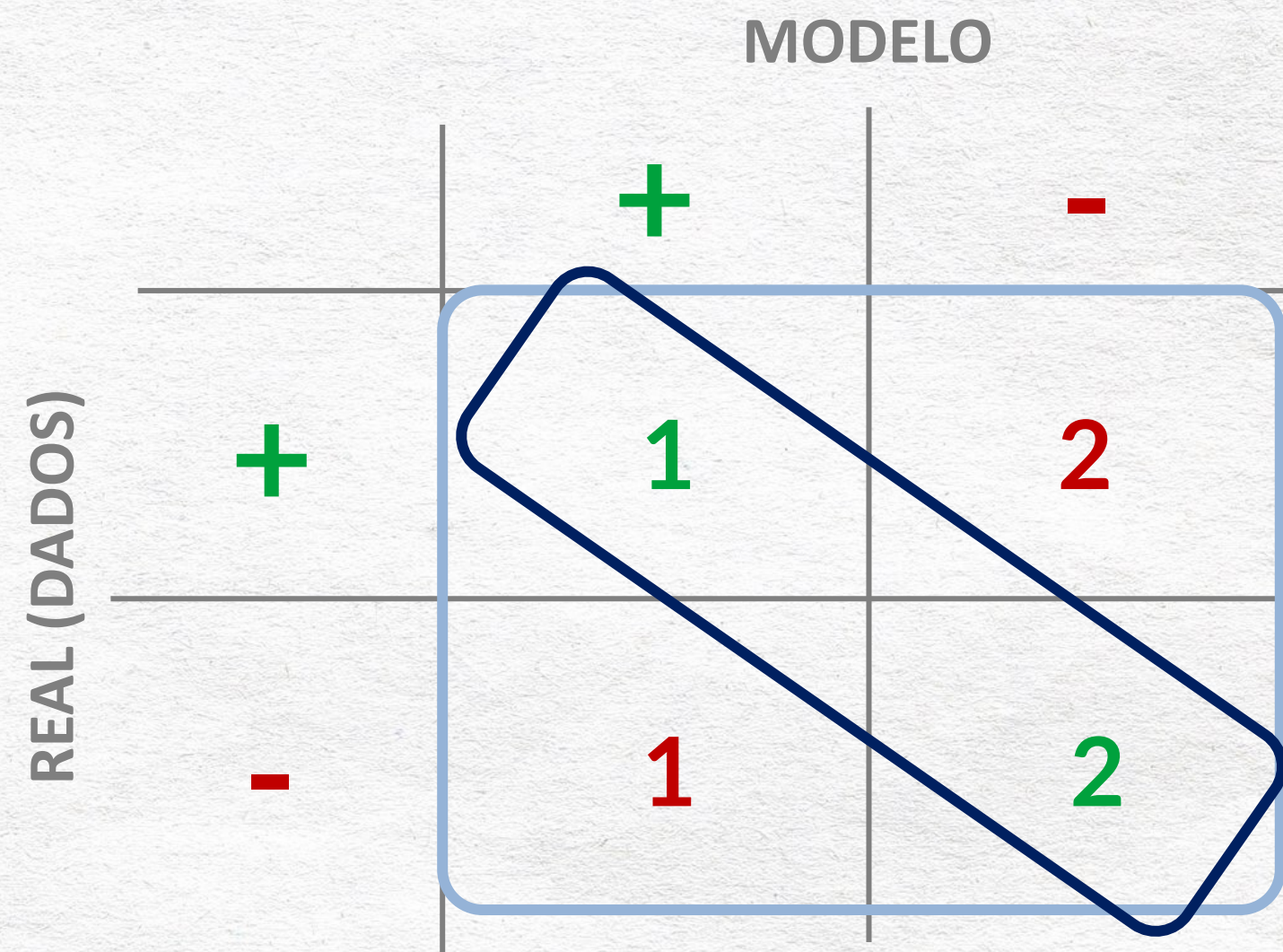


AVALIANDO O MODELO DE CLASSIFICAÇÃO

ACURÁCIA:

$$Acurácia = \frac{\text{Verdadeiro Positivo} + \text{Verdadeiro Negativo}}{\text{Total}}$$

$$Acurácia = \frac{1 + 2}{1 + 2 + 1 + 2} = 50\%$$



AVALIANDO O MODELO DE CLASSIFICAÇÃO

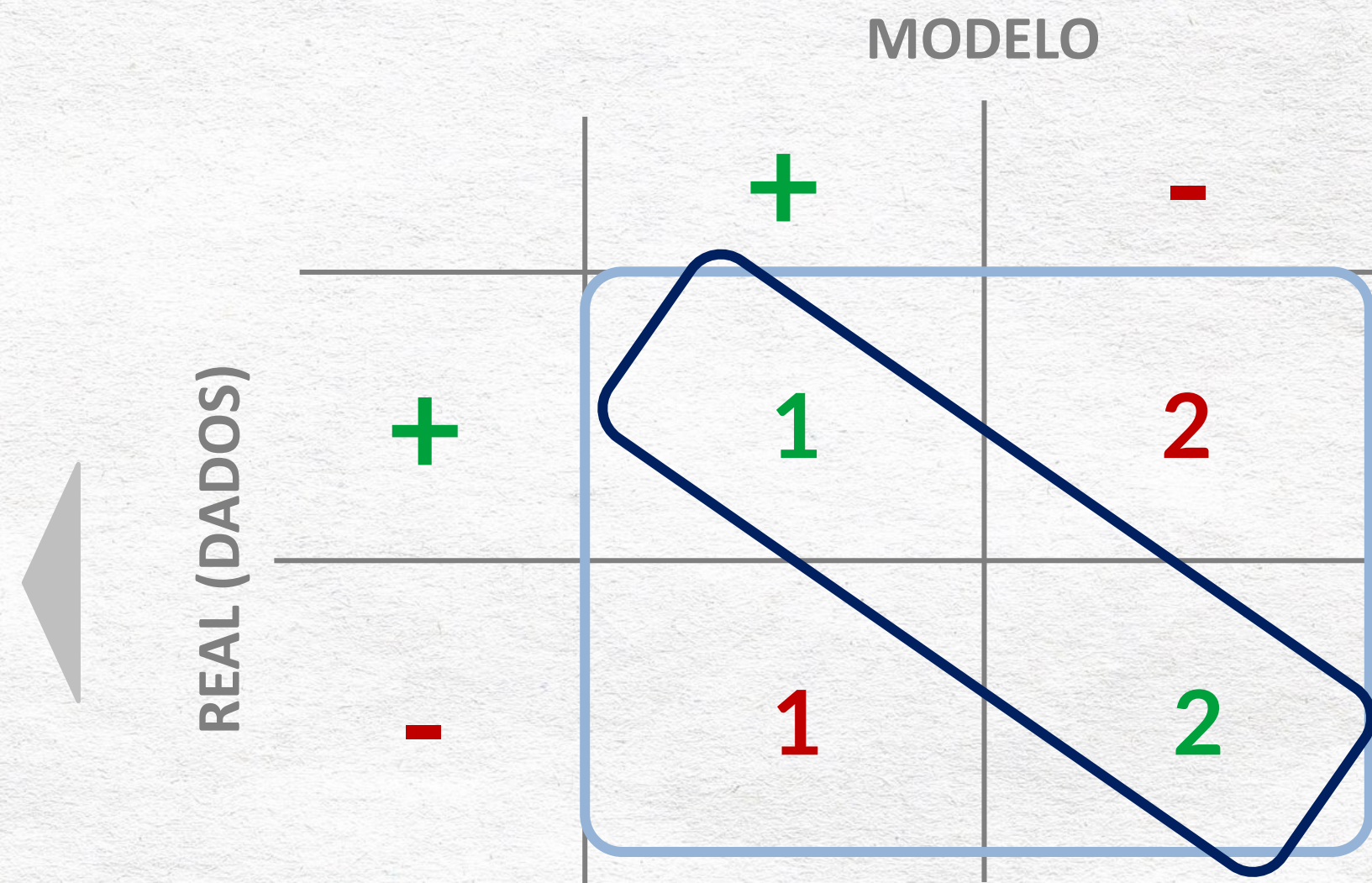
ACURÁCIA:

$$Acurácia = \frac{Verdadeiro\ Positivo + Verdadeiro\ Negativo}{Total}$$

$$Acurácia = \frac{1 + 2}{1 + 2 + 1 + 2} = 50\%$$

```
from sklearn.metrics import accuracy_score  
accuracy_score(y_true, y_pred)
```

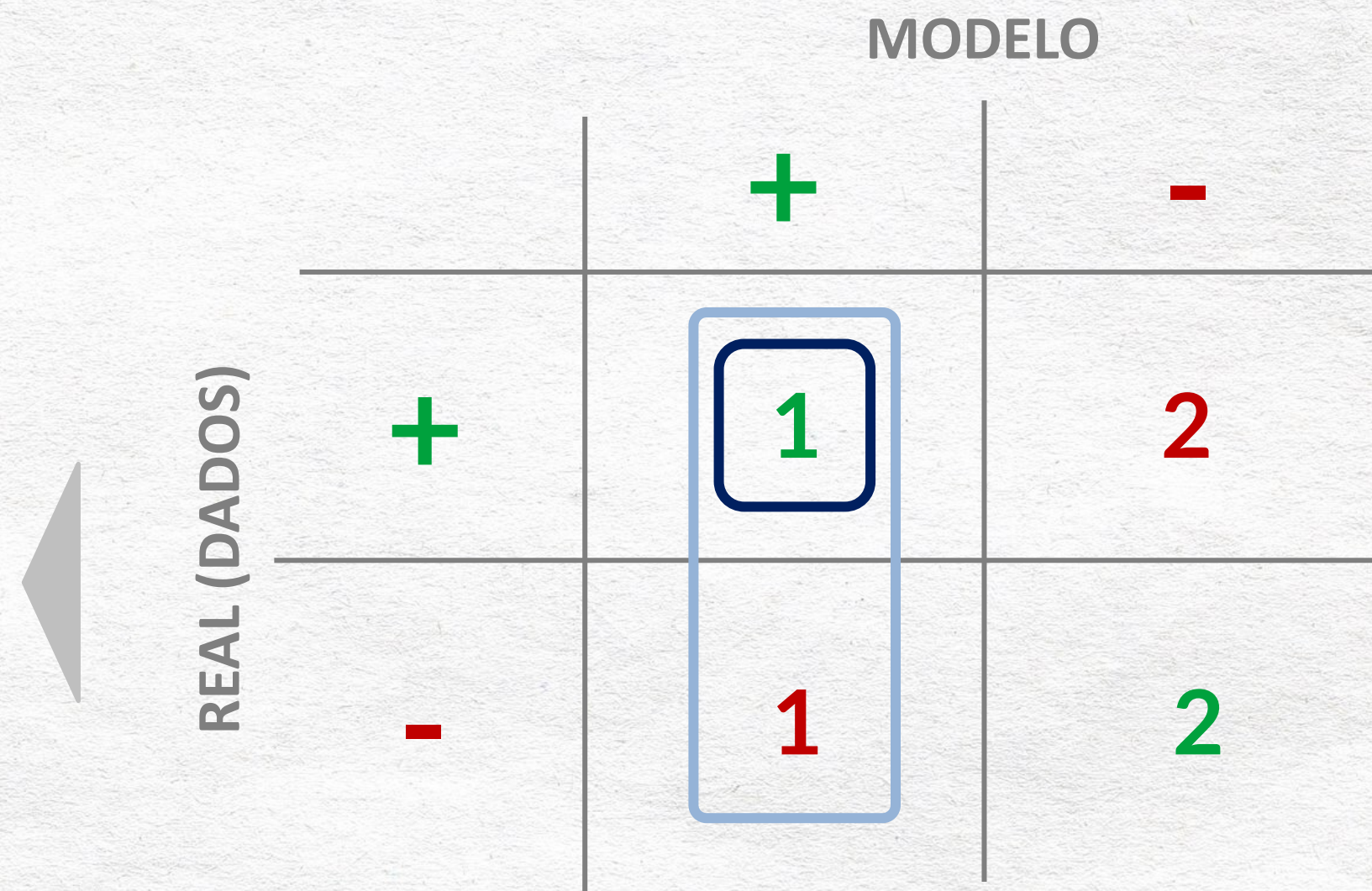
0.5



AVALIANDO O MODELO DE CLASSIFICAÇÃO

PRECISÃO:

$$\text{Precisão} = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo} + \text{Falso Positivo}}$$



A confusion matrix diagram illustrating the calculation of Precision. The matrix is a 2x2 grid with 'REAL (DADOS)' on the vertical axis and 'MODELO' on the horizontal axis. The cells contain the following values: Top-Left (True Positive) is a green '+'; Top-Right (False Positive) is a red '-'; Bottom-Left (False Negative) is a red '-'; Bottom-Right (True Negative) is a green '2'. A blue box highlights the first column (Positive class), containing a green '1' (True Positive) and a red '1' (False Positive). The numbers are color-coded to match the signs: green for positive and red for negative.

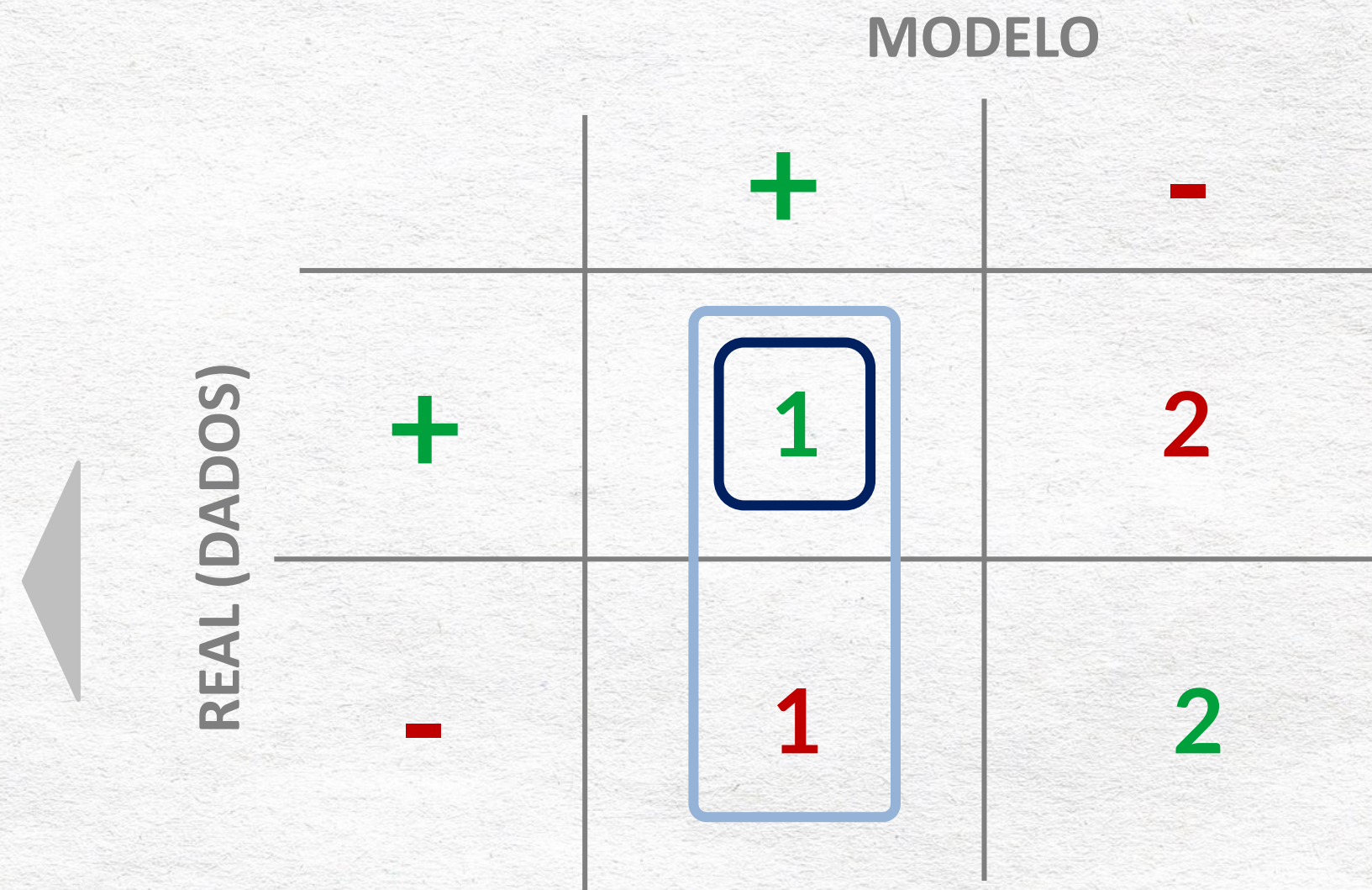
	MODELO	
REAL (DADOS)	+	-
	+	-
	1	2
	-	2

AVALIANDO O MODELO DE CLASSIFICAÇÃO

PRECISÃO:

$$Precisão = \frac{Verdadeiro\ Positivo}{Verdadeiro\ Positivo + Falso\ Positivo}$$

$$Precisão = \frac{1}{1 + 1} = 50\%$$



A confusion matrix diagram illustrating the calculation of Precision. The matrix is a 2x2 grid with 'REAL (DADOS)' on the vertical axis and 'MODELO' on the horizontal axis. The cells contain the following values: Top-Left (True Positive) is a green '+'; Top-Right (False Positive) is a red '-'; Bottom-Left (False Negative) is a red '-'; Bottom-Right (True Negative) is a green '2'. A blue box highlights the first column, which contains the True Positive (green '+') and False Positive (red '-') cells. The True Positive cell contains a green '1', and the False Positive cell contains a red '1'. A large grey arrow points from the matrix towards the precision formula on the left.

	MODELO	
REAL (DADOS)	+	-
	<div>+</div> <div>1</div>	<div>-</div> <div>2</div>

AVALIANDO O MODELO DE CLASSIFICAÇÃO

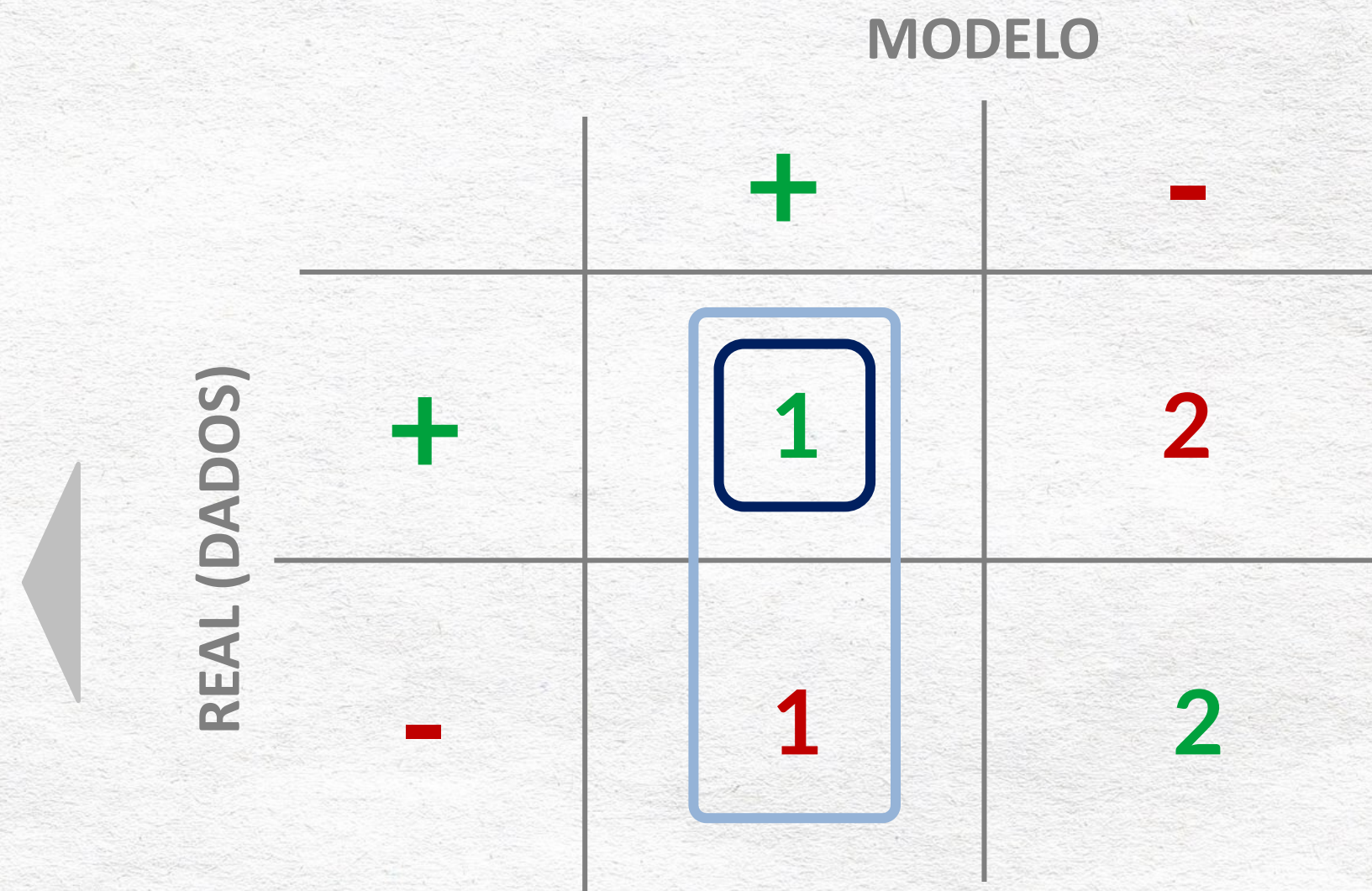
PRECISÃO:

$$\text{Precisão} = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo} + \text{Falso Positivo}}$$

$$\text{Precisão} = \frac{1}{1 + 1} = 50\%$$

```
from sklearn.metrics import precision_score  
precision_score(y_true, y_pred, pos_label='+')
```

0.5



A confusion matrix diagram illustrating classification results. The vertical axis is labeled 'REAL (DADOS)' and the horizontal axis is labeled 'MODELO'. The matrix is divided into four quadrants by a horizontal and a vertical line. The top-left quadrant contains a green '+' sign. The top-right quadrant contains a red '-' sign. The bottom-left quadrant contains a green '+' sign. The bottom-right quadrant contains a red '-' sign. In the center of the matrix, there are two overlapping boxes: a blue box containing the number '1' and a red box containing the number '2'. The blue box is positioned over the top-left and bottom-left quadrants, while the red box is positioned over the top-right and bottom-right quadrants.

	+	-
+	1	2
-	1	2

AVALIANDO O MODELO DE CLASSIFICAÇÃO

PRECISÃO:

$$\text{Precisão} = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo} + \text{Falso Positivo}}$$

$$\text{Precisão} = \frac{1}{1 + 1} = 50\%$$

```
from sklearn.metrics import precision_score  
precision_score(y_true, y_pred, pos_label='+')
```

0.5

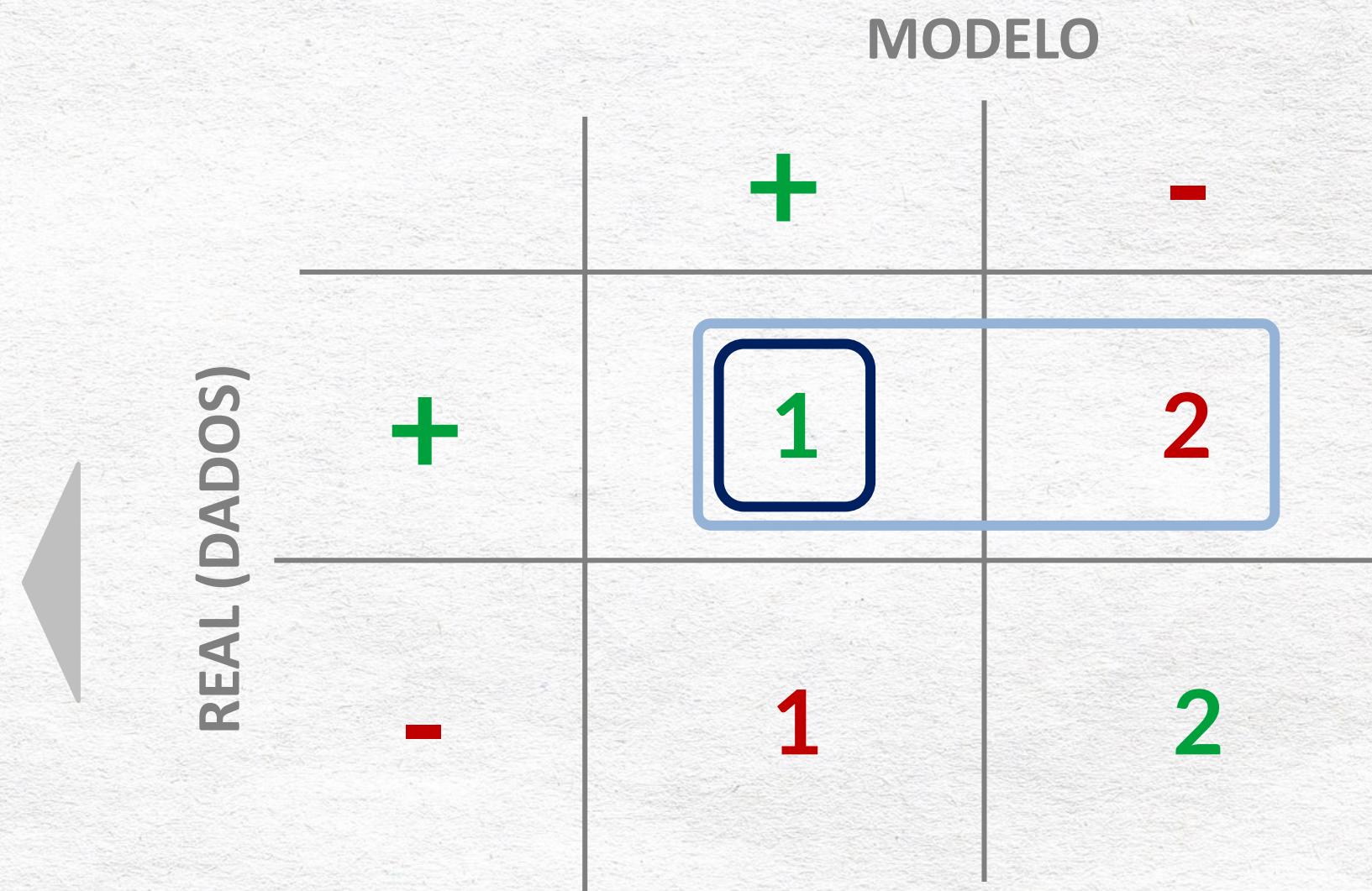
		MODELO	
		+	-
REAL (DADOS)	+	1	2
	-	1	2

Quero evitar ao máximo falsos positivos!
(classificar um cliente como bom pagador sem ele realmente ser)

AVALIANDO O MODELO DE CLASSIFICAÇÃO

RECALL:

$$Recall = \frac{Verdadeiro\ Positivo}{Verdadeiro\ Positivo + Falso\ Negativo}$$



A confusion matrix diagram illustrating the calculation of Recall. The vertical axis is labeled 'REAL (DADOS)' and the horizontal axis is labeled 'MODELO'. The matrix is divided into four quadrants by a horizontal and a vertical line. The top-left quadrant contains a green '+' sign. The top-right quadrant contains a red '-' sign. The bottom-left quadrant contains a red '-' sign. The bottom-right quadrant contains a green '+' sign. A blue box highlights the top-right quadrant, which contains a green '1' and a red '2'. A grey arrow points from the text 'Recall = Verdadeiro Positivo / (Verdadeiro Positivo + Falso Negativo)' to the blue box.

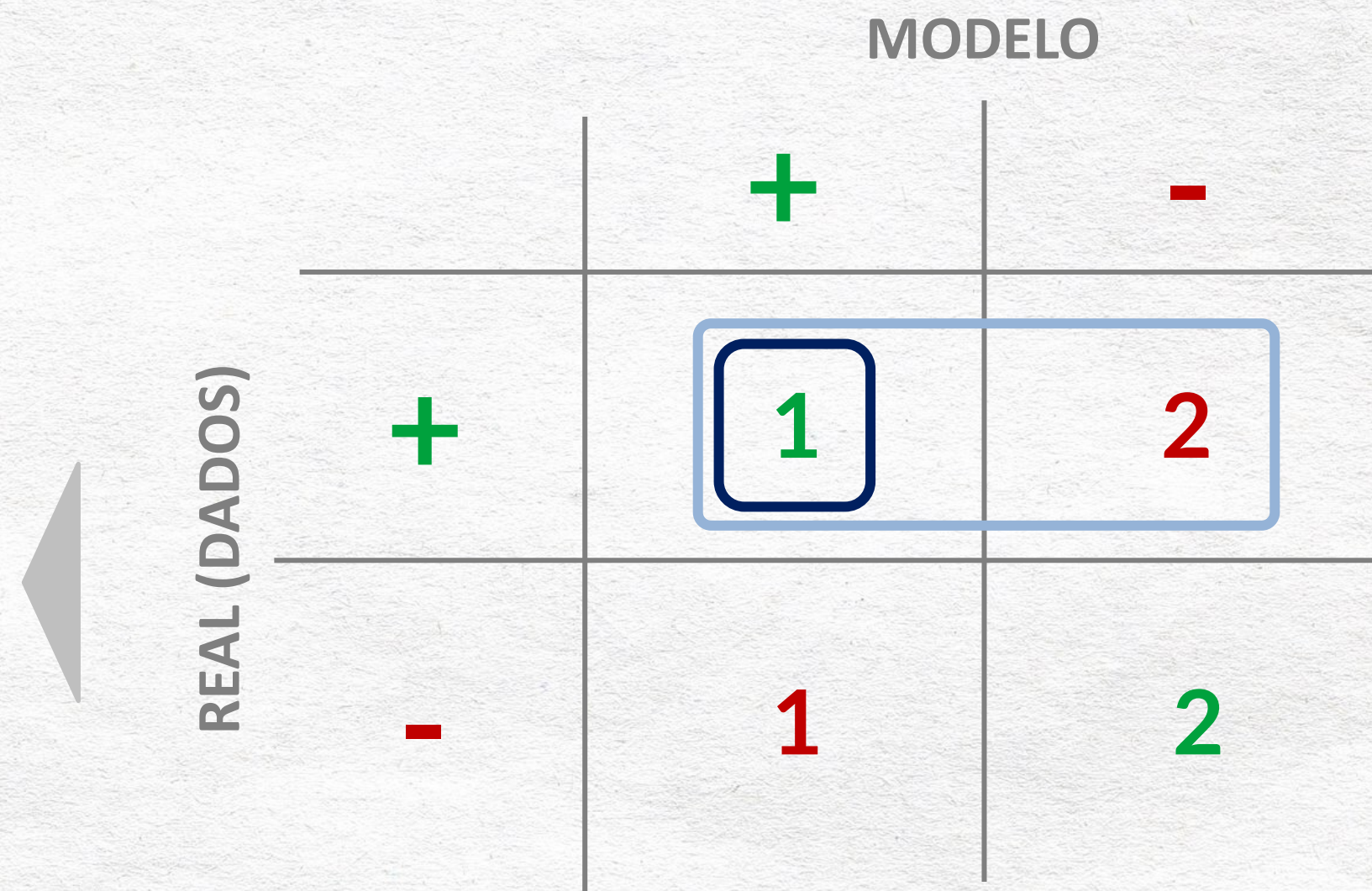
	+	-
+	1	2
-	1	2

AVALIANDO O MODELO DE CLASSIFICAÇÃO

RECALL:

$$Recall = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo} + \text{Falso Negativo}}$$

$$Recall = \frac{1}{1 + 2} = 33\%$$



A confusion matrix diagram illustrating the calculation of Recall. The matrix is a 2x2 grid with 'REAL (DADOS)' on the vertical axis and 'MODELO' on the horizontal axis. The cells contain the following values:

	+	-
+	1	2
-	1	2

The cell containing '1' (True Positive) is highlighted with a blue box. The cell containing '2' (False Negative) is also highlighted with a blue box. The cell containing '1' (False Positive) is highlighted with a red box. The cell containing '2' (True Negative) is highlighted with a green box.

AVALIANDO O MODELO DE CLASSIFICAÇÃO

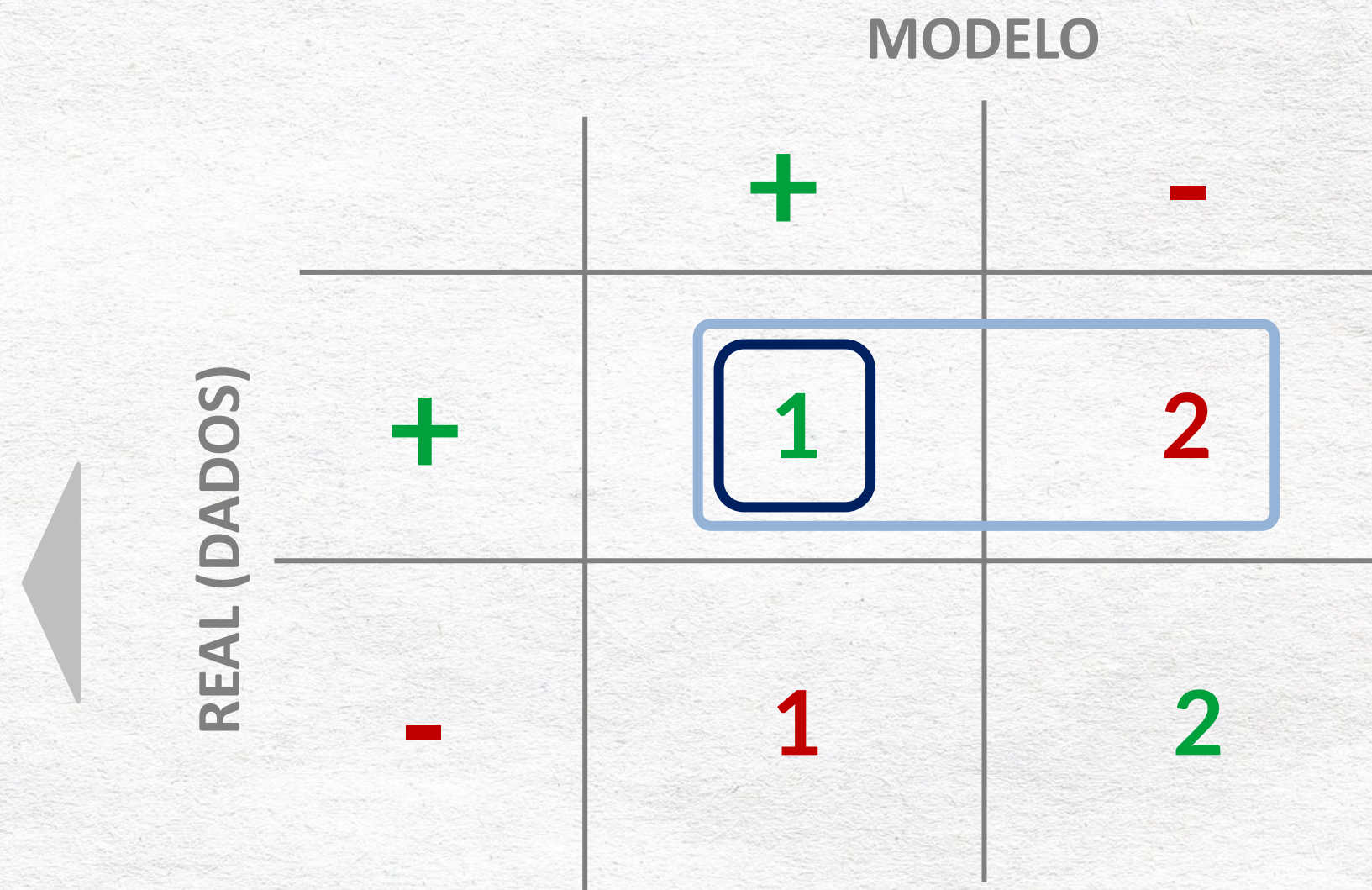
RECALL:

$$Recall = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo} + \text{Falso Negativo}}$$

$$Recall = \frac{1}{1 + 2} = 33\%$$

```
from sklearn.metrics import recall_score  
recall_score(y_true, y_pred, pos_label='+')
```

```
0.3333333333333333
```



A confusion matrix diagram illustrating classification results. The vertical axis is labeled 'REAL (DADOS)' and the horizontal axis is labeled 'MODELO'. The matrix is divided into four quadrants by a horizontal and a vertical line. The top-left quadrant contains a green '+' sign. The top-right quadrant contains a red '-' sign. The bottom-left quadrant contains a red '1'. The bottom-right quadrant contains a green '2'. A blue box highlights the top-right quadrant, which contains a green '1' and a red '2'.

	REAL (+)	REAL (-)
MODELO (+)	1	2
MODELO (-)	1	2

AVALIANDO O MODELO DE CLASSIFICAÇÃO

RECALL:

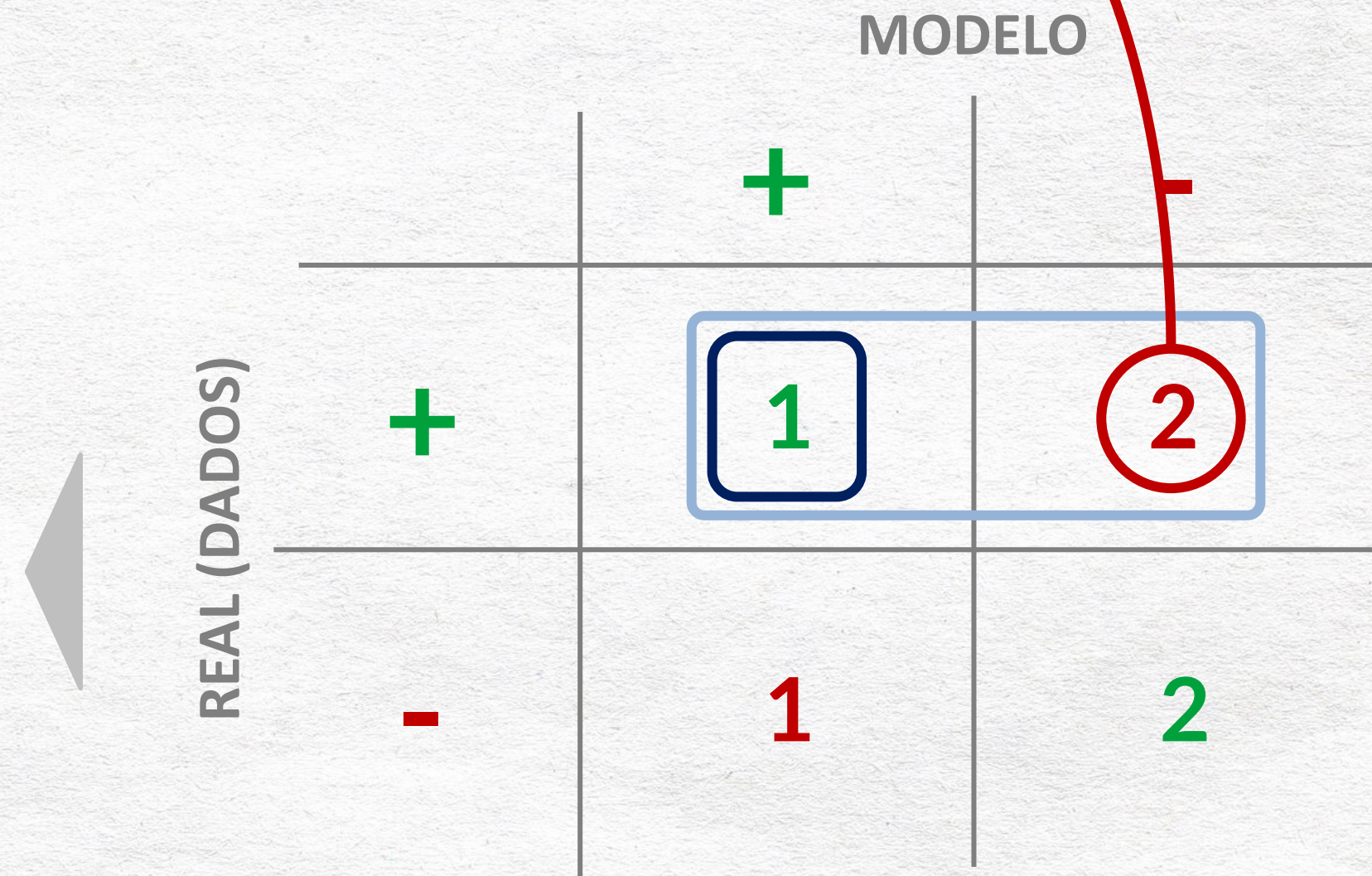
$$Recall = \frac{Verdadeiro\ Positivo}{Verdadeiro\ Positivo + Falso\ Negativo}$$

$$Recall = \frac{1}{1 + 2} = 33\%$$

```
from sklearn.metrics import recall_score  
recall_score(y_true, y_pred, pos_label='+')
```

```
0.3333333333333333
```

Quero evitar ao máximo falsos negativos!
(falar que uma transação não é fraude
sendo que na verdade ela é)



AVALIANDO O MODELO DE CLASSIFICAÇÃO

AVALIANDO FRAUDES!

Um recall de 100% é bom?

		MODELO		
		+	-	
REAL (DADOS)	+	1	2	RECALL
	-	1	2	
		PRECISÃO		

AVALIANDO O MODELO DE CLASSIFICAÇÃO

AVALIANDO FRAUDES!

Um recall de 100% é bom?

```
# Dados
```

```
y_true = [0,0,0,1,1,1]
```

```
y_pred = [1,1,1,1,1,1]
```

```
from sklearn.metrics import confusion_matrix  
confusion_matrix(y_true,y_pred,labels=[1,0])
```

```
array([[3, 0],  
       [3, 0]], dtype=int64)
```

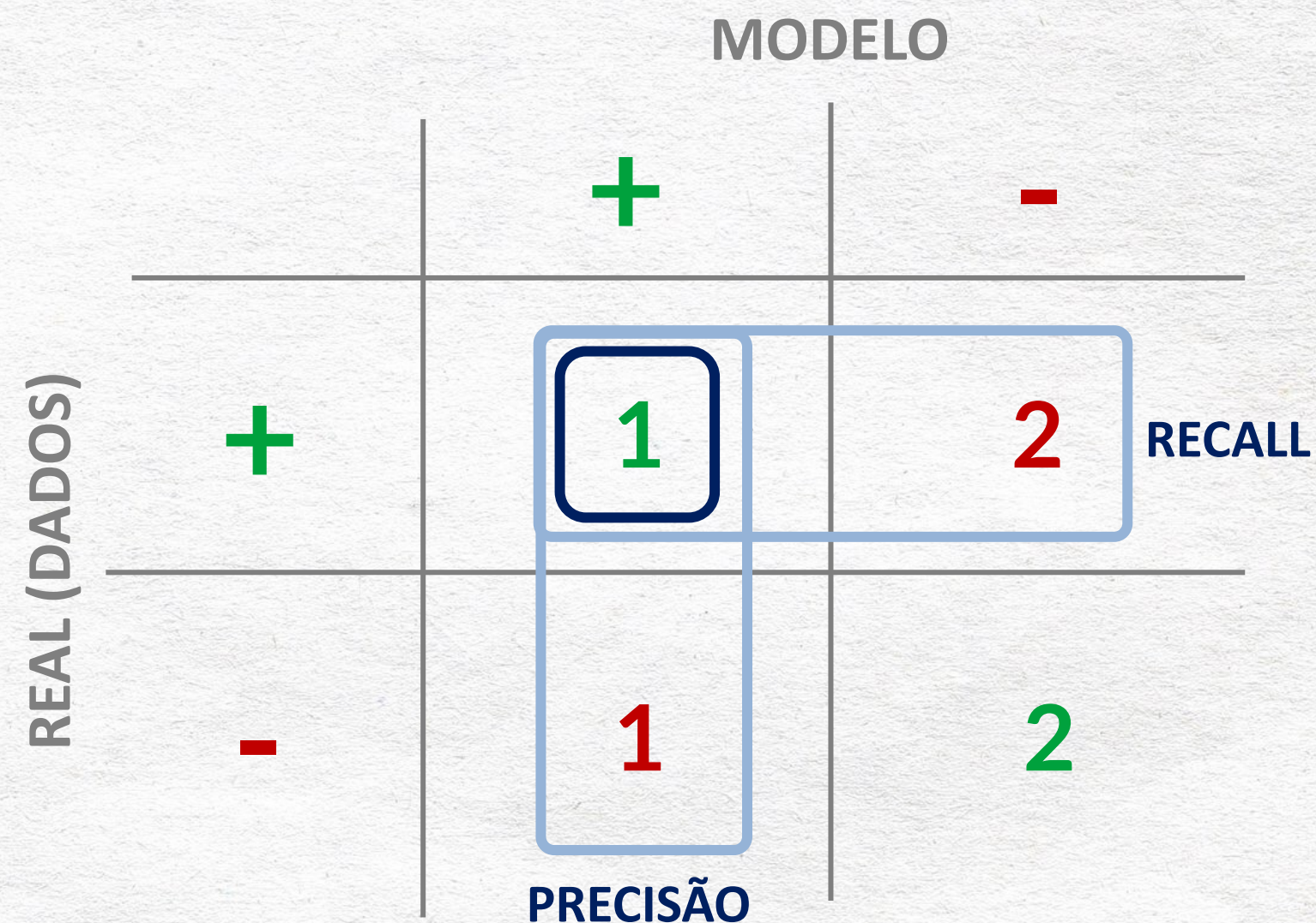
```
from sklearn.metrics import precision_score  
precision_score(y_true,y_pred)
```

```
0.5
```

```
from sklearn.metrics import recall_score  
recall_score(y_true,y_pred)
```

```
1.0
```

1: é fraude
0: não é fraude



AVALIANDO O MODELO DE CLASSIFICAÇÃO

AVALIANDO FRAUDES!

Um recall de 100% é bom?

```
# Dados
y_true = [0,0,0,1,1,1]
y_pred = [1,1,1,1,1,1]

from sklearn.metrics import confusion_matrix
confusion_matrix(y_true,y_pred,labels=[1,0])

array([[3, 0],
       [3, 0]], dtype=int64)

from sklearn.metrics import precision_score
precision_score(y_true,y_pred)

0.5

from sklearn.metrics import recall_score
recall_score(y_true,y_pred)

1.0
```

1: é fraude
0: não é fraude

Diagrama de Confusão Matrix (2x2) com eixos REAL (DADOS) e MODELO.

	+	-	
+	3	0	RECALL
-	3	0	
	PRECISÃO		

Detalhes do Diagrama:

- O eixo vertical (REAL (DADOS)) e o eixo horizontal (MODELO) definem a estrutura da matriz.
- Os sinais de "+" e "-" representam as classes de fraude e não-fraude, respectivamente.
- Os valores na diagonal (3 e 3) representam as classificações corretas.
- Os valores fora da diagonal (0 e 0) representam as classificações incorretas.
- Os retângulos azuis destacam os valores 3 na diagonal, que correspondem às métricas de precisão e recall.

AVALIANDO O MODELO DE CLASSIFICAÇÃO

AVALIANDO FRAUDES!

Uma acurácia de 90% é boa?

AVALIANDO O MODELO DE CLASSIFICAÇÃO

AVALIANDO FRAUDES!

Uma acurácia de 90% é boa?

```
# Dados
y_true = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1]
y_pred = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
```

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_true,y_pred,labels=[1,0])
```

```
array([[ 0,  2],
       [ 0, 18]], dtype=int64)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_true,y_pred)
```

0.9

1: é fraude

0: não é fraude

AVALIANDO O MODELO DE CLASSIFICAÇÃO

AVALIANDO FRAUDES!

Uma função para calcular a precisão:

```
# Dados  
y_true  
y_pred
```

```
from sklearn.metrics import accuracy_score
```

```
array
```

```
from sklearn.metrics import accuracy_score  
accuracy_score(y_true, y_pred)
```

0.9

1: é fraude

0: não é fraude

MODELO

2

18

PRECISÃO

CUIDADO! Essas métricas de avaliação não possuem nenhum valor sozinhas (e até podem levar a conclusões erradas). Aqui percebemos claramente a importância de sermos cientistas de dados impressionadores!