

# Fundamentos de Ciencias de Datos

Semana 10 - Resumen Clasificador MNIST

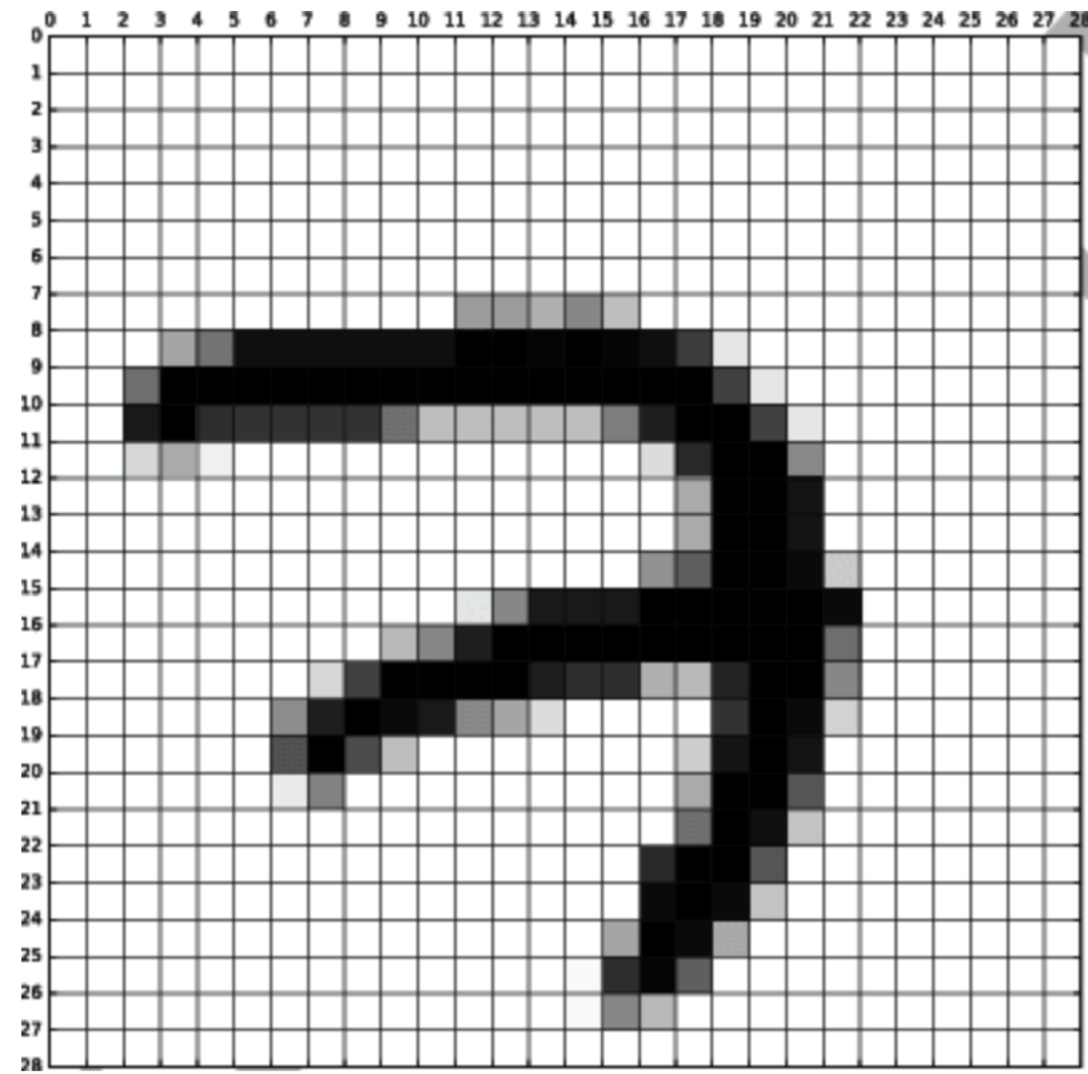
# Dataset MNIST

Recordemos que el *dataset* MNIST se compone de 70,000 imágenes de dígitos escritos a mano

Cada imagen es de 28x28 pixeles

Cada pixel va entre el 0 (completamente blanco) al 255 (completamente negro)

# Dataset MNIST



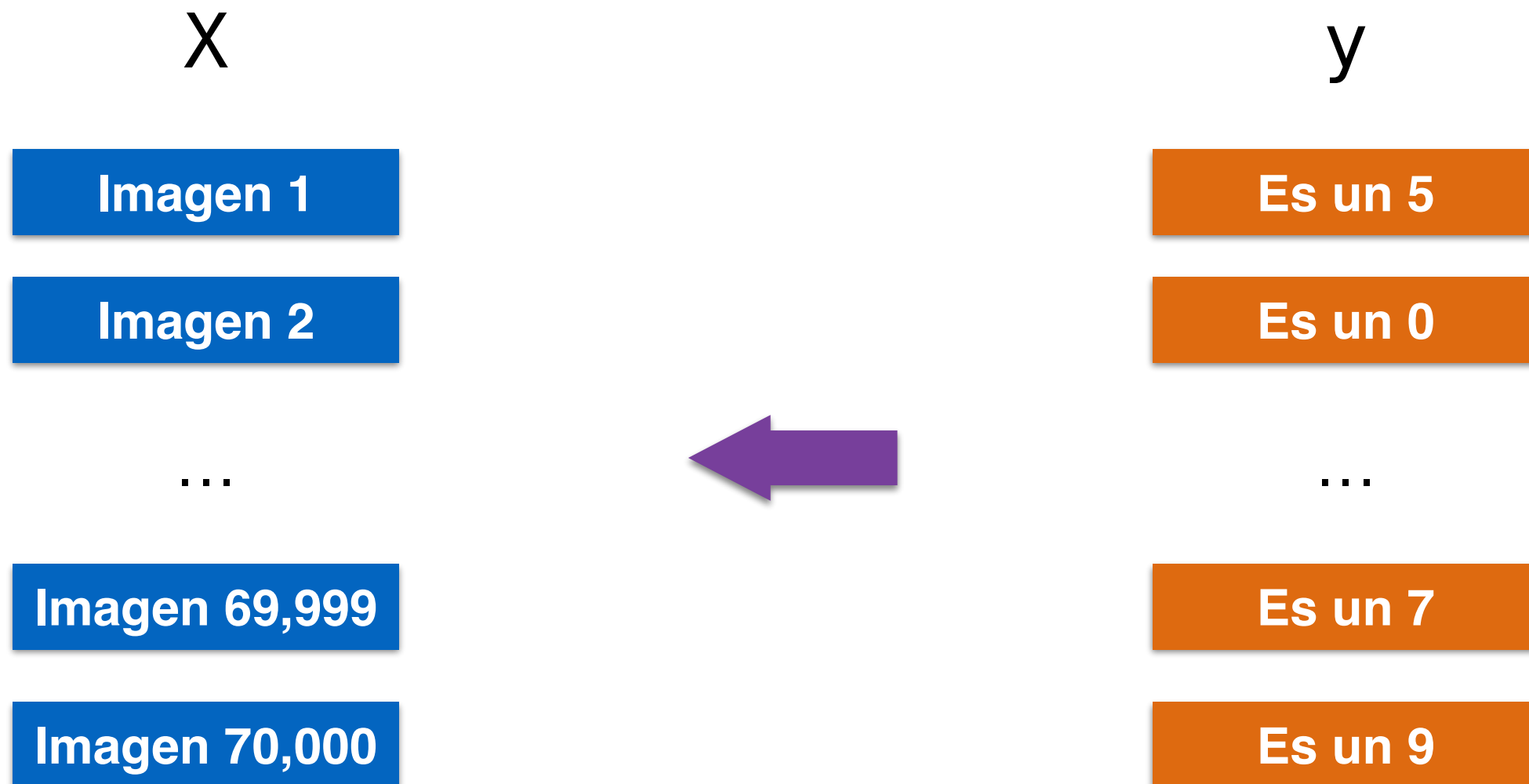
# Dataset MNIST

Recordemos que en Python, cada imagen se representa como una fila de largo 784 (recordemos que  $28 \times 28 = 784$ )

Por lo tanto, tendremos un DataFrame de 70,000 filas y 784 columnas

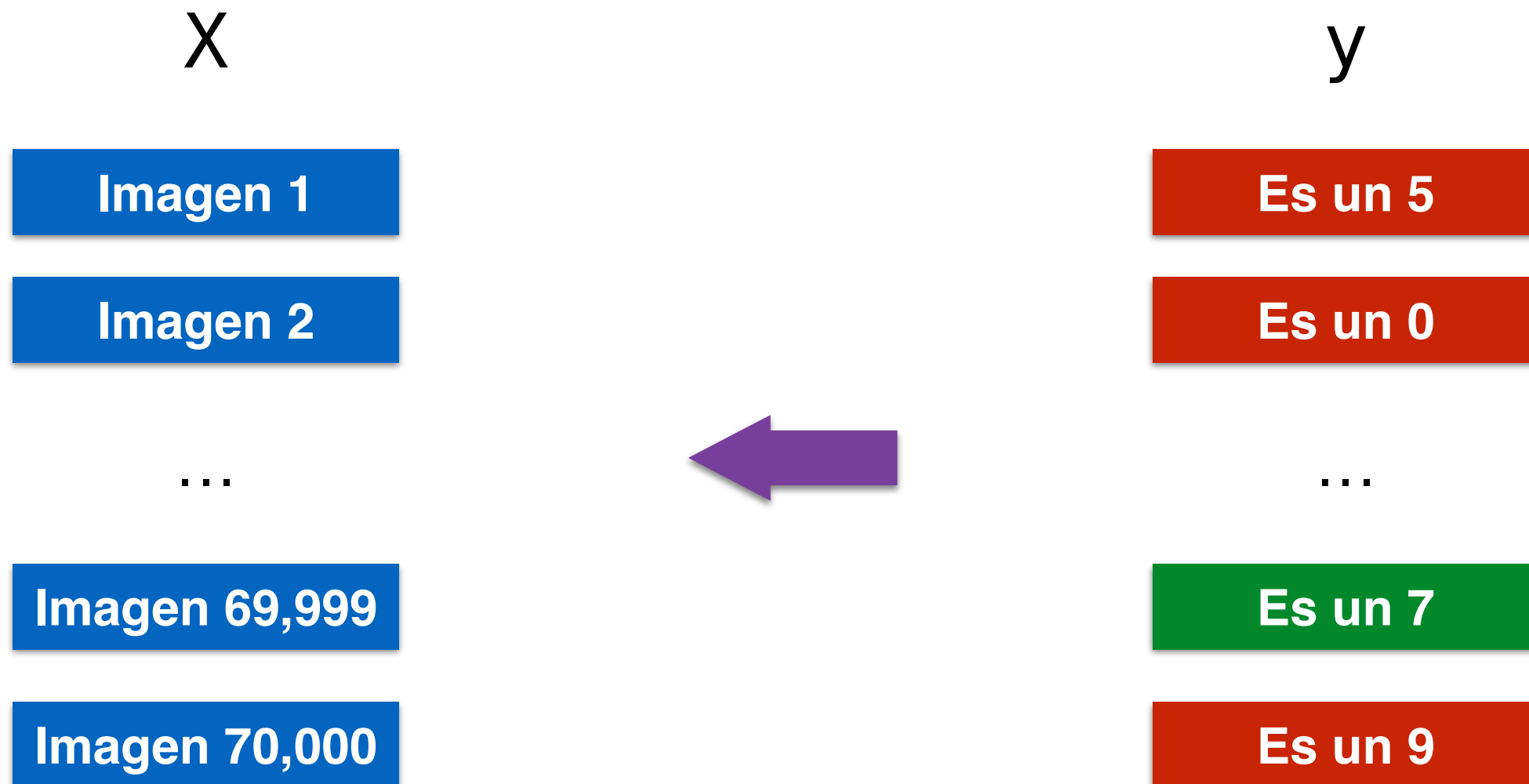
# Dataset MNIST

Además cada imagen viene con la etiqueta que le corresponde



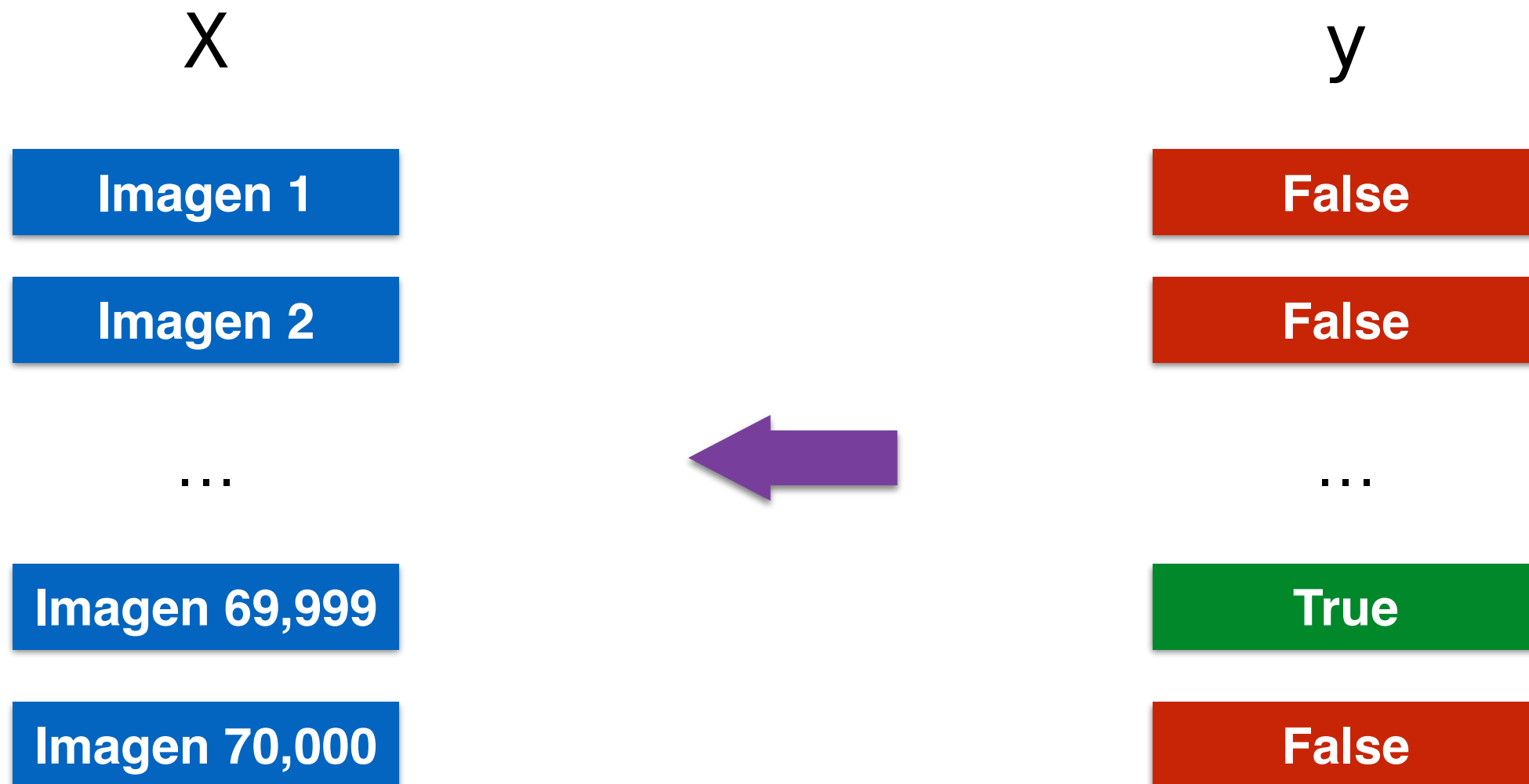
# Clasificador de 7s

Vamos a entrenar un clasificador binario que nos diga si una imagen es un 7 o no



# Clasificador de 7s

Convertimos  $y$  a un arreglo booleano



# Clasificador de 7s

Y ahora tenemos que dividir nuestros datos en entrenamiento y prueba

El *dataset* de Scikit Learn ya viene ordenado: las primeras 60,000 imágenes son los datos de entrenamiento y las últimas 10,000 son datos de prueba

Recordemos que en general esta división se hace con la función `test_train_split`



# Clasificador de 7s

Ojo, cuando el *dataset* no viene preparado, la división entre entrenamiento y prueba se hace con la función `test_train_split` (ver clase de regresión lineal)

# Clasificador de 7s

## Datos de entrenamiento

$X_{\text{train}}$

$y_{\text{train}}$

Imagen 1

False

Imagen 2

False

...

...

Imagen 59,999

False

Imagen 60,000

True



## Datos de prueba

$X_{\text{test}}$

$y_{\text{test}}$

Imagen 60,001

True

Imagen 60,002

False

...

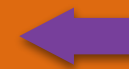
...

Imagen 69,999

True

Imagen 70,000

False



# Clasificador de 7s

Importamos el modelo `SGDClassifier` de Scikit Learn y le pasamos los ejemplos de entrenamiento (60,000 imágenes) con sus respuestas

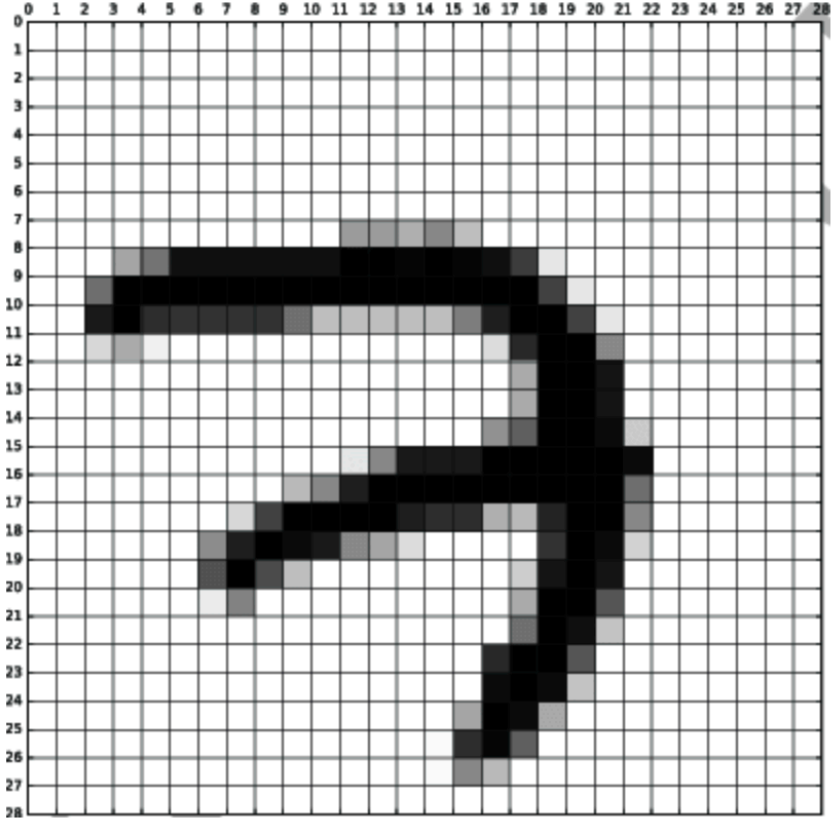
# Clasificador de 7s

```
SGDClassifier.fit(  
    X_train, y_train  
    Imagen 1, False  
    Imagen 2, False  
    ...  
    Imagen 59,999, False  
    Imagen 60,000, True  
)
```

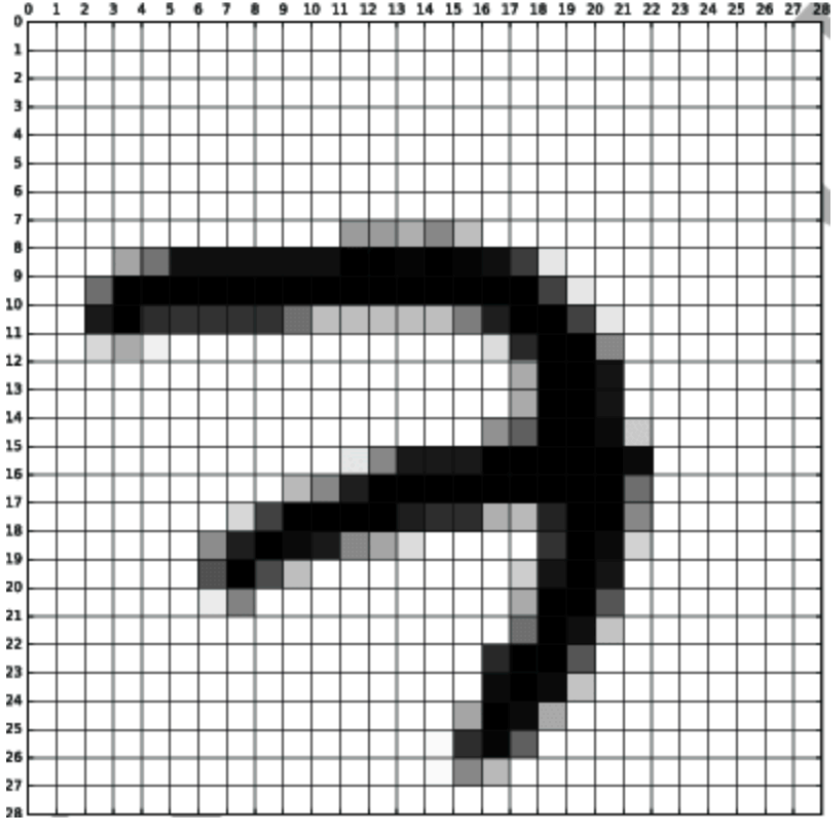
# Clasificador de 7s

Con esto el modelo es capaz de predecir nuevas imágenes

# Clasificador de 7s

`SGDClassifier.predict(`  `)`

# Clasificador de 7s

`SGDClassifier.predict(`  `)`

Según lo que  
aprendí es un 7

# Clasificador de 7s

También podemos predecir sobre todos los datos de prueba

```
SGDClassifier.predict(  
    X_test  
    Imagen 60,001  
    Imagen 60,002  
    ...  
    Imagen 69,999  
    Imagen 70,000  
)
```



# Clasificador de 7s

Y obtenemos una lista con las respuestas

**X\_test**

**Imagen 60,001**

**Imagen 60,002**

...

**Imagen 69,999**

**Imagen 70,000**

**y\_pred**

**Respuesta img 60,001: True**

**Respuesta img 60,002: False**

...

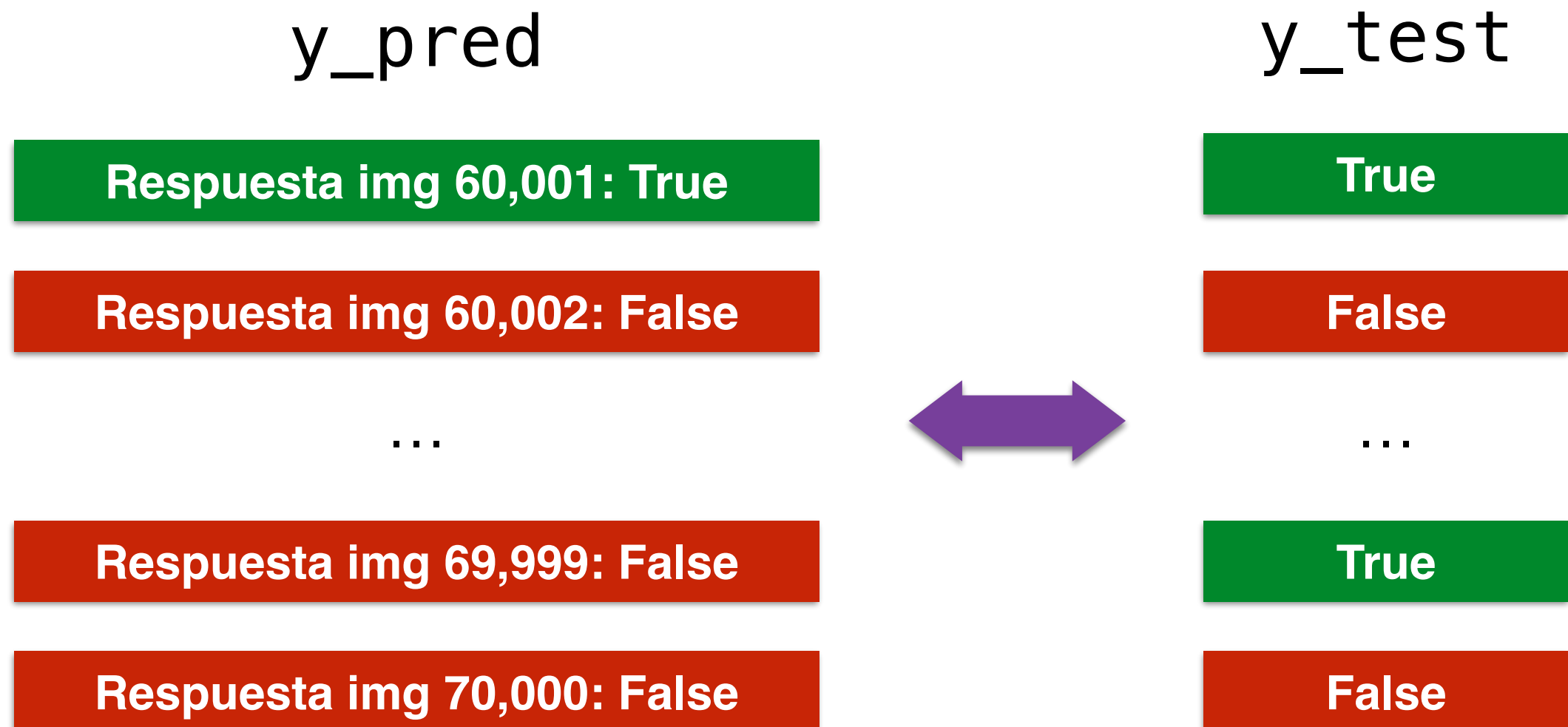
**Respuesta img 69,999: False**

**Respuesta img 70,000: False**



# Clasificador de 7s

Y como tenemos las **respuestas reales** para esas imágenes, podemos ver cómo se comportó nuestro clasificador



# Clasificador de 7s

Aquí calculamos el porcentaje de respuestas correctas (*accuracy*), que recordemos que nos dio entre 96% - 97%

# Fundamentos de Ciencias de Datos

Semana 10 - Resumen Clasificador MNIST