

## PRÁCTICA: BASADOS EN INSTANCIAS

En el fichero denominado “digits\_from\_5\_to\_9.csv” se encuentra una tabla con filas de tamaño 28x28, que almacena el valor de cada uno de los pixels de un dígito manuscrito entre 5 y 9. Hay una columna adicional, que tiene el entero correspondiente al dígito. Cada uno de los mencionados píxeles viene codificado por un entero entre 0 y 255 de acuerdo a una escala de grises. Este fichero tiene una primera línea con la cabecera de las columnas.

Antes de seguir adelante, deberá separar un conjunto de muestras para aprendizaje y otro para test basándose en el método de resorte tomando la razón habitual de  $\frac{2}{3}$  y  $\frac{1}{3}$ .

A continuación se harán varios experimentos de clasificación aplicando el método de los k vecinos más próximos, que en python se lleva a cabo con:

- `sklearn.neighbors.KneighborsClassifier`

De entre sus argumentos, aparte del número k, aparece otro denominado “algorithm”. Aquí hay que usar el denominado “brute”, ya que corresponde con el más simple descrito en la clase de teoría.

### PARTE I:

Variando el k entre 1 y 7, ejecutar el algoritmo y probarlo con los datos de test para cada pareja de dígitos, para obtener los dos que mejor se distinguen con este método. Por tanto, hay que obtener esta pareja ganadora para cada k. Si siempre saliese la misma combinación de dígitos, extraer el k óptimo.

### PARTE II:

Lo mismo que lo anterior, sólo que los píxeles hay que reducirlos a bits, esto es, poner uno si el pixel no era cero y dejar el cero, si ya lo tenía.

### ENTREGA:

Un fichero jupyter-notebook/lab, con todo el código y las explicaciones pertinentes para seguir los pasos de esta práctica.