

Reto 2

1. Enunciado

El objetivo de esta práctica es construir un clasificador multiclase capaz de separar imágenes del MNIST del 0, del 3, del 6 y del 9.

Para ello hay que construir dos o más características que se extraigan de la imagen y después, opcionalmente, aumentar la dimensionalidad mediante transformaciones. Con el conjunto de características definitivo, se entrena un clasificador lineal.

Está permitido:

- Utilizar transformaciones de características predefinidas; por ejemplo como las que se pueden encontrar en `sklearn.preprocessing` [↗](#)

Está prohibido:

- Entrenar el clasificador con características que no sean las obtenidas en el código que se entrena, y que además se entrega. Eso no significa que no se puedan probar varias, pero al final hay que indicar claramente cuales son las dos elegidas.
- Utilizar técnicas de reducción de la dimensionalidad como PCA, LLE, Isomap, t-SNE, ... (ya las utilizaremos más adelante)
- Clasificadores no lineales (SVM, DT, RF...) y *Ensembles* (Bagging, Boosting, ...).

2. Condiciones de entrega

- El equipo debe estar formado por 2 alumnos.
- Todos tendrán acceso al mismo conjunto de datos, que consiste en cuatro ficheros CSV para entrenar, “1000_cero.csv”, “1000_tres.csv”, “1000_seis.csv” y “1000_nueve.csv” y un fichero para la competición, “reto2_X.csv”.
- La entrega debe ser un archivo comprimido ZIP que contenga:
 - un fichero .TXT con el nombre de los alumnos del grupo
 - el código de python utilizado para entrenar el clasificador y generar las etiquetas del fichero de la competición.
 - un fichero con las etiquetas generadas que debe cumplir obligatoriamente lo siguiente:
 1. Nombre: “**Reto2_Ypred.csv**”
 2. Formato: Una etiqueta por línea, guardada como un entero; como en el Reto 1.
- Se puede utilizar el código proporcionado en clase; y si se utiliza código de terceros debe estar indicado con el comentario `*** codigo de terceros ! **`
- La fecha límite para subir el fichero ZIP es el **lunes 25 de octubre a las 6:00 am**.

- Se valorará:
 - Los comentarios añadidos en el código
 - El proceso seguido con los datos para entrenar
 - Las características construidas
 - El resultado final

3. Descripción del conjunto de datos

- Todos los ficheros proporcionados utilizan la coma como separador y contienen muestras del 0, 3, 6 y 9 del MNIST.
- Cada fila tiene 784 números enteros positivos entre 0 y 255 que se corresponden con la intensidad de luz en cada píxel de un ejemplo del MNIST.
- Si, por ejemplo, se quiere mostrar el ejemplo número 15 del fichero `1000_tres.csv` se debe redimensionar a 28×28 . El siguiente código es una manera de hacerlo:

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
# Suponiendo que el fichero 1000_tres.csv está volcado en la variable X3
>>> plt.imshow(X3[15,:].reshape(28,28))
>>> plt.show()
```