

Reto 1

1. Enunciado

El objetivo de esta práctica es construir un clasificador lineal mediante regresión logística capaz de separar imágenes del MNIST del 3 de y del 7.

Para ello hay que construir dos características que se extraigan de la imagen o que se obtengan mediante ingeniería de características. Después se utilizarán esas dos únicas características para entrenar el clasificador.

Está prohibido:




- Utilizar transformaciones de características predefinidas; por ejemplo como las que se pueden encontrar en `sklearn.preprocessing` [!\[\]\(b1b781be830eb908d845c527ab08d5f8_img.jpg\)](#)
- Utilizar técnicas de reducción de la dimensionalidad como PCA, LLE, Isomap, t-SNE, ... (ya las utilizaremos más adelante)
- Entrenar el clasificador con más de dos características o con dos características que no sean las elegidas finalmente. Eso no significa que no se puedan probar varias, pero al final hay que indicar claramente cuales son las dos elegidas.
- Entrenar un clasificador que no sea lineal. Además se debe utilizar la regresión logística y no otro, como SVM por ejemplo.

2. Condiciones de entrega

- El equipo debe estar formado por 2 alumnos.
- Todos tendrán acceso al mismo conjunto de datos, que consiste en dos ficheros CSV para entrenar, “*1000_tres.csv*” y “*1000_siete.csv*” y un fichero para la competición, “*reto1_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: “**Reto1_Ypred.csv**”
 2. Formato: Una etiqueta por línea, guardada como un entero

Por ejemplo, si en el código las etiquetas generadas para el conjunto de datos de la competición se han almacenado en la variable `y_pred`, las siguientes instrucciones generan un fichero CSV que cumple los requisitos:

```
>>> import numpy as np
>>> np.savetxt('reto1_Ypred.csv', y_pred, fmt='%i', delimiter=',')
```

Reto1_Y.csv	Reto1_Y.csv	Reto1_Y.csv
<pre> 3 7 7 7 3 7 3 3 : </pre> 	<pre> 3, 7, 7, 7, 3, 7 ... </pre> 	<pre> 3.0 7.0 7.0 7.0 3.0 7.0 3.0 3.0 : </pre> 

- 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 11 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

- Para el reto se proporcionan tres ficheros CSV: `1000_tres.csv`, `1000_siete.csv`, `reto1_X.csv`
- Todos utilizan la coma como separador y contienen cifras del MNIST.
- Cada fila tiene 784 números enteros positivos entre 0 y 255 que se corresponden con la intensidad de luz en cada pixel 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()

```