

Práctica 0 - Aprendizaje Automático

Introducción a Python, Matplotlib y NumPy

Este trabajo, cuya puntuación máxima será de 5 puntos (1.25 pts cada uno de los cuatro ejercicios), pretende servir como práctica y refuerzo de los conocimientos adquiridos en las primeras sesiones prácticas, en donde se han introducido los principios de Python, Matplotlib y NumPy.

Límite de entrega: **13 de Marzo de 2022 a las 23:59 (PRADO)**

Materiales a entregar: un único fichero Python (.py o .ipynb). No es necesario entregar ninguna memoria o informe explicativo. El código debe estar bien comentado.

1. Ejercicio 1

- Leer la base de datos de Iris que hay en scikit-learn. Véase https://scikit-learn.org/stable/datasets/toy_dataset.html. El [conjunto de datos de flor Iris](#), introducido por Ronald Fisher en 1936, es clásico en aprendizaje automático.
- Obtener las características (datos de entrada X) y la clase (y).
- Quedarse con las características primera y tercera. Recuerdese que en Python los índices comienzan en 0.
- Visualizar con un Scatter Plot los datos, coloreando cada clase con un color diferente e indicando con una leyenda la clase a la que corresponde cada color. Más concretamente, el resultado debe ser el que se muestra en la Figura 3. Esta es la única salida que proporciona este ejercicio. No se debe mostrar ninguna otra información por terminal.

2. Ejercicio 2

- Separar en training (80 % de los datos) y test (20 %) aleatoriamente, conservando la proporción de elementos en cada clase tanto en training como en test. Con esto se pretende evitar que haya clases infra-representadas en entrenamiento o test. Con “aleatoriamente” nos referimos a que no sería válido escoger como conjunto de entrenamiento el 80 % inicial de ejemplos de cada clase y como conjunto de test el 20 % final. Es decir, los ejemplos deben desordenarse con respecto al orden original. En la implementación de este ejercicio no se puede recurrir a funciones como *train_test_split* de scikit-learn o similares.
- Se debe imprimir por terminal el número resultante de ejemplos de cada clase, tanto en entrenamiento como en test, así como las clases de cada uno de los ejemplos de entrenamiento y test. En la Figura 2 se muestra el tipo de salida que se espera obtener.

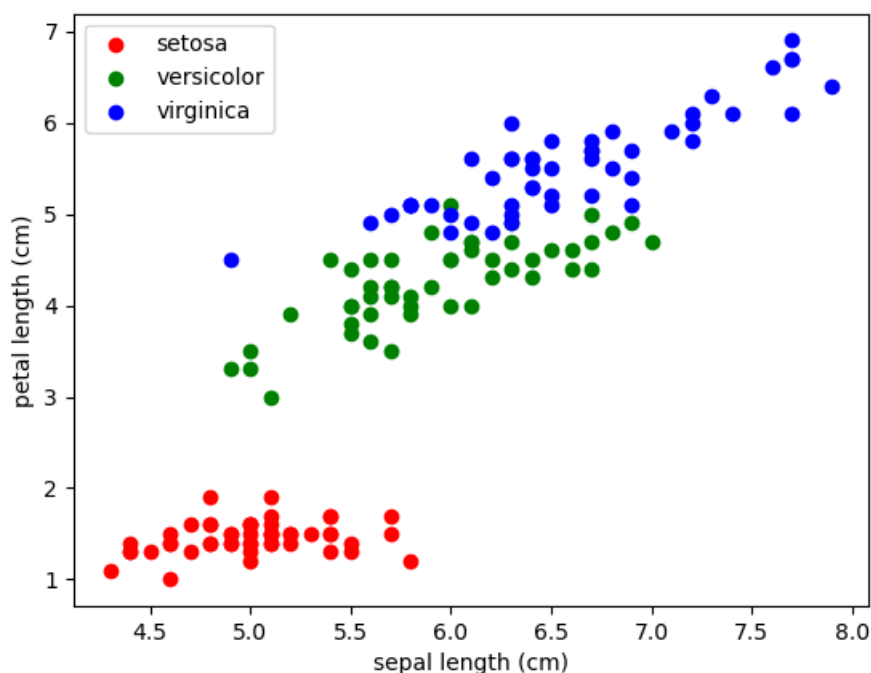


Figura 1: Scatter Plot que se debe calcular y mostrar en el Ejercicio 1.

```

--- Clase setosa ---
Ejemplos train: 40
Ejemplos test: 10
--- Clase versicolor ---
Ejemplos train: 40
Ejemplos test: 10
--- Clase virginica ---
Ejemplos train: 40
Ejemplos test: 10
Clase de los ejemplos de entrenamiento:
[0. 0. 2. 0. 2. 0. 2. 0. 1. 2. 0. 1. 0. 2. 2. 2. 0. 2. 1. 1. 2. 0. 0. 1.
 2. 1. 1. 0. 0. 0. 1. 2. 2. 2. 0. 0. 1. 2. 1. 0. 1. 0. 2. 1. 0. 2. 2. 1.
 0. 1. 0. 0. 2. 1. 1. 1. 1. 1. 0. 1. 1. 2. 2. 1. 0. 1. 2. 0. 2. 0. 0. 0.
 0. 2. 2. 1. 2. 1. 1. 1. 2. 2. 2. 1. 1. 1. 0. 0. 2. 2. 2. 2. 2. 1. 0. 2.
 1. 1. 0. 0. 1. 0. 0. 1. 2. 0. 2. 2. 0. 1. 2. 1. 0. 1. 0. 1. 0. 1. 2.]
Clase de los ejemplos de test:
[1. 2. 2. 2. 0. 1. 0. 2. 1. 1. 2. 0. 1. 2. 2. 0. 1. 1. 1. 2. 0. 1. 2. 0.
 0. 0. 1. 0. 2. 0.]

```

Figura 2: Salida que se debe mostrar por terminal para el Ejercicio 2. El orden exacto de los patrones de entrenamiento y test no tiene por qué coincidir con el de este ejemplo.

3. Ejercicio 3

- Obtener 100 valores equiespaciados entre 0 y 4π
- Obtener el valor de $10^{-5} \cdot \sinh(x)$, $\cos(x)$ y $\tanh(2 \cdot \sin(x) - 4 \cdot \cos(x))$ para los 100 valores anteriormente calculados.
- Visualizar las tres curvas simultáneamente en el mismo plot (con líneas discontinuas en

verde, negro y rojo) e incluir la leyenda correspondiente. Más concretamente, el resultado debe ser el que se muestra en la Figura 3. Esta es la única salida que proporciona este ejercicio. No se debe mostrar ninguna otra información por terminal.

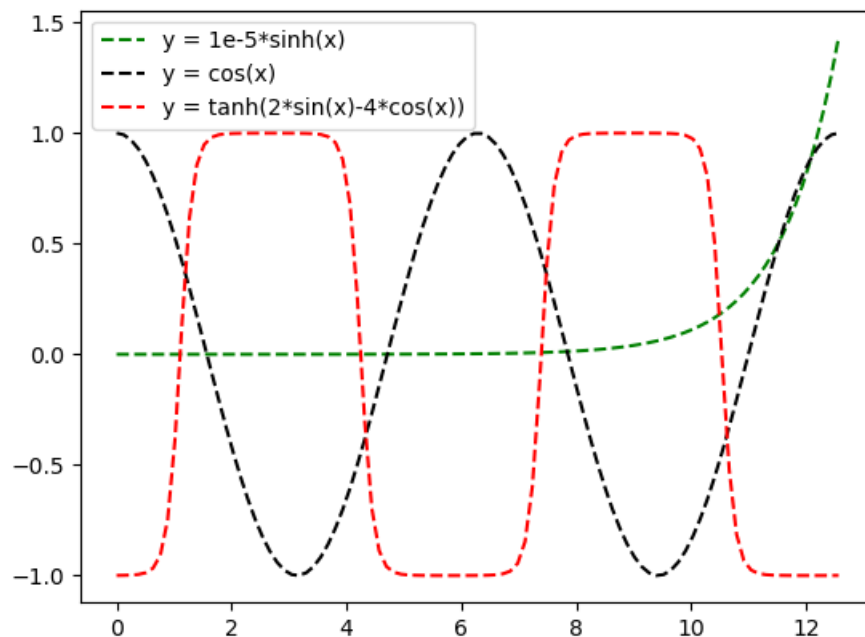
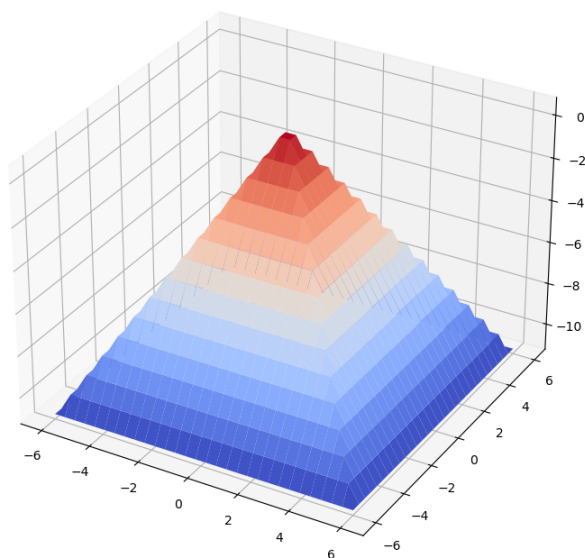


Figura 3: Gráfica que se debe calcular y mostrar en el Ejercicio 3.

4. Ejercicio 4

- Mostrar dos funciones 3D dentro de la misma figura. Las funciones a mostrar son las siguientes:
 - $f(x, y) = 1 - |x + y| - |y - x|$
 - $f(x, y) = x \cdot y \cdot e^{(-x^2 - y^2)}$
- Como se indicaba, ambas funciones deben mostrarse dentro de la misma figura/ventana como *surface plots*. Más concretamente, el resultado debe ser exactamente el que se muestra en la Figura 4 (tanto a nivel de título de las gráficas, como de rango de valores en los ejes y colores empleados). Esta es la única salida que proporciona este ejercicio. No se debe mostrar ninguna otra información por terminal. El siguiente enlace puede servir de ayuda y referencia a la hora de realizar este ejercicio: <https://matplotlib.org/stable/gallery/mplot3d/subplot3d.html>. De cara a mostrar ecuaciones matemáticas en figuras de Matplotlib usando LaTeX la siguiente referencia puede ser de utilidad: https://matplotlib.org/stable/gallery/text_labels_and_annotations/tex_demo.html.

Pirámide



$x \cdot y \cdot e^{(-x^2 - y^2)}$

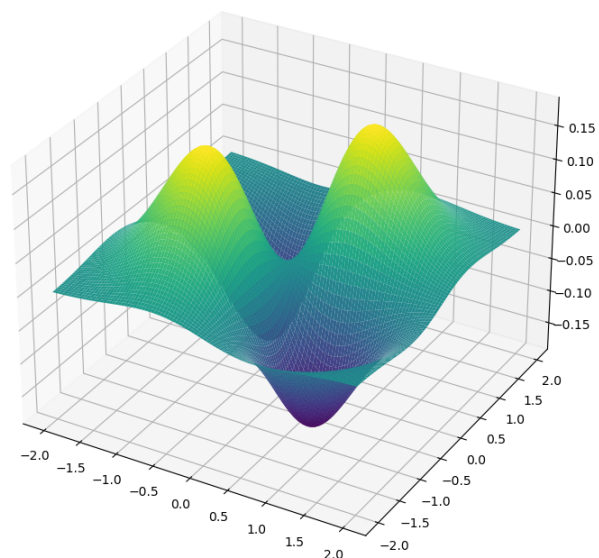


Figura 4: Gráficas que se deben calcular y mostrar, dentro de la misma Figura, en el Ejercicio 4.