**Regresión Logística**

**Objetivos**

Estos ejercicios tienen como objetivo que te familiarices con la librería Keras implementando modelos simples de Regresión Logística.

**Preparación**

Desde una terminal de Anaconda ubicarse en el directorio de la práctica donde se encuentra los archivos con extensión “.ipynb” (notebooks) y escribir: “jupyter notebook”. Desde el entorno web, acceder a la dirección del servidor (generalmente <http://localhost:8888/>) para abrir los archivos.

**Ejercicio 1** Así como con la Regresión Lineal, los siguientes cuadernos te ayudarán a comprender mejor el funcionamiento de la función sigmoidea/logística, la regresión logística y la entropía cruzada, su función de error más apropiada.

***Archivos*: Funcion Logistica.ipynb**

**Regresion Logistica - Modelo.ipynb**

**Regresion Logistica - Aprendizaje.ipynb**

**Separación por hiperplanos.ipynb**

**Regresion Logistica 2D - Modelo.ipynb**

**Regresion Logistica 2D - Aprendizaje.ipynb**

**Ejercicio 2** El código de ejercicio implementa regresión logística con Keras. Si bien no hay nada que implementar, el objetivo del ejercicio es comprender cómo utilizar la API de Keras para entrenar un modelo con la función de activación de la regresión logística y la función de error asociada, así como la forma de codificar la salida.

***Archivo*: Regresion Logistica con Keras.ipynb**

**Ejercicio 3** En este ejercicio, proponemos entrenar modelos de regresión logística para varios conjuntos de datos de clasificación, y visualizar sus resultados. Los conjuntos de datos son de problemas de clasificación binaria o multiclase, y nos permitirán comprender los límites de la regresión logística (y lineal). Los conjuntos de datos *iris* y *diabetes* tienen más de 2 dimensiones de entrada por lo que no podrán ser visualizados.

Anotá los mejores resultados y cómo los obtuviste.

***Regresion Logística con Keras y varios conjuntos de datos.ipynb***

*Conjuntos de datos*:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Archivo | Variables de entrada | Cantidad de clases | Mejor accuracy | Mejor loss | Épocas | Tasa de aprendizaje |
| 2\_clases\_simple.csv |  |  |  |  |  |  |
| 6\_clases\_dificil.csv |  |  |  |  |  |  |
| circulos.csv |  |  |  |  |  |  |
| diabetes.csv |  |  |  |  |  |  |
| iris.csv |  |  |  |  |  |  |
| moons\_mis-scaled.csv |  |  |  |  |  |  |
| moons.csv |  |  |  |  |  |  |

**Ejercicio 4.** Utilizando regresión logística, clasificá a los países del conjunto de datos de expectativa de vida en base a la columna **Status**. Evaluá si hay un desbalance de clases, y cómo impacta en la clasificación. Indicá, en base a los pesos encontrados del modelo, a qué atributos se le da mayor importancia para clasificar a los ejemplos. Podés basarte en el código de ejercicios previos para entrenar el modelo.

**Archivo: who\_life\_expectancy\_no\_missing.csv**

**Ejercicio 5.** **(Opcional)** Implementa las diversas métricas de clasificación y codificaciones de la salida utilizando numpy.

**Archivo: Métricas de Clasificación.ipynb**

**Ejercicio 6.**  **(Opcional)** Utilizando regresión logística binaria (es decir, para clasificar entre 2 clases), el siguiente cuaderno clasifica ejemplos de imágenes de dígitos manuscritos. Explorá el cuaderno, observando las curvas de error y accuracy, y cambiando los dígitos a clasificar para ver cuales pueden tener problemas.

***Archivo*: Regresion Logistica Múltiple - Imágenes.ipynb**

**Ejercicio 7 (Avanzado, Opcional).** Realiza las guías 10 a 12 de [*simplenn*](https://github.com/facundoq/simplenn)