Computación Blanda - Machine learning 03

Soft Computing - Machine learning 03

Autor: Karen Posada Muñoz

*IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e: karen.posada@utp.edu.co

***Resumen*— Este documento muestra y explica el funcionamiento de una serie de códigos hechos en python, que son el abrebocas hacía el Machine learning y nos brindan unas herramientas para el manejo de estructuras de datos con las librerías numpy y scipy**

***Palabras clave—*numpy, python, scipy**

***Abstract*— This document shows and explains the operation of a series of codes made in python, which are the opening to the Machine learning and give us some tools for the management of data structures with the libraries numpy and scipy**

***Key Word*— numpy, python , scipy**

1. INTRODUCCIÓN

Dentro del Machine Learning existen diferentes herramientas que nos ayudan al manejo y procesamiento de los datos, podremos observar algunas de estas herramientas.

1. CÓDIGO

*data = np.genfromtxt("/content/drive/My Drive/web\_traffic.tsv", delimiter="\t")*

*print(data[:10], '\n')*

*print(data.shape)*

En las primeras líneas de código vamos a obtener los datos y además les haremos una exploración a los primeros 10 datos almacenados en *data, además de que conoceremos el tamaño de la matriz*

*[[1.000e+00 2.272e+03]*

*[2.000e+00 nan]*

*[3.000e+00 1.386e+03]*

*[4.000e+00 1.365e+03]*

*[5.000e+00 1.488e+03]*

*[6.000e+00 1.337e+03]*

*[7.000e+00 1.883e+03]*

*[8.000e+00 2.283e+03]*

*[9.000e+00 1.335e+03]*

*[1.000e+01 1.025e+03]]*

*(743, 2)*

*---------------------------------------*

*x = data[:,0]*

*y = data[:,1]*

*print(x, '\n') #Se muestra el vector x*

*print(y, '\n') #Se muestra el vector y*

Lo siguiente que haremos es tomar el arreglo que almacenamos en la variable *Data* y la vamos a llevar a dos vectores que denominaremos *x & y*

---------------------------------------------------------------------------

*print(x.ndim, '\n') #Imprimir la dimensión de x*

*print(y.ndim, '\n') #Imprimir la dimensión de y*

*print(x.shape, '\n')*

*print(y.shape)*

Además, con la función *ndim* podremos conocer las dimensiones de nuestros vectores, en nuestro caso sabemos que pasamos de un arreglo de dos columnas a dos vectores unidimensionales y la función *shape* nos dirá el tamaño de los dos vectores.

*1*

*1*

*(743,)*

*(743,)*

*---------------------------------------*

*print(np.sum(np.isnan(y)))*

Como pudimos observar en la primera exploración de los datos, existen unos cierta información que aparece de la forma *nan,* el cual significa *Not a Number,* con ayuda de la función *isnan* podemos averiguar cuándo un dato es nan y con la función *sum* podremos saber cual es el total de estos en el vector y.

*8*

---------------------------------------------------------------------------

*print(x.shape, '\n')*

*print(y.shape, '\n')*

*# Se eliminan los elementos nan tanto de x como de y*

*x = x[~np.isnan(y)]*

*y = y[~np.isnan(y)]*

*# Se cuenta el número de elementos tanto de x como de y*

*print(x.shape, '\n')*

*print(x.shape, '\n')*

En las primeras dos líneas de código podremos determinar cuál es el tamaño de los vectores y en las dos próximas líneas vamos a eliminar los datos nan que encontramos en el vector con la función pasada. después volveremos a imprimir los datos para que vea si se eliminaron los nana previamente identificados

*(743,)*

*(743,)*

*(735,)*

*(735,)*

*---------------------------------------*

*# Se importa la librería para graficar*

*import matplotlib.pyplot as plt*

*# Dibuja los puntos (x,y) con círculos de tamaño 10*

*plt.scatter(x, y, s=10)*

*# Títulos de la gráfica*

*plt.title("Tráfico Web en el último mes")*

*plt.xlabel("Tiempo")*

*plt.ylabel("Accesos/Hora")*

*plt.xticks([w\*7\*24 for w in range(10)],['semana %i' % w for w in range(10)])*

*plt.autoscale(tight=True)*

*# dibuja una cuadrícula punteada ligeramente opaca*

*plt.grid(True, linestyle='-', color='0.75')*

*# Muestra el gráfico*

*plt.show()*

Lo que haremos al final es importar la librería gráfica de python *Matplotlib*  para hacer una visualización de los datos,

Para este ejemplo decidimos graficar a través de puntos de tamaño 10 y le pondremos de título a la gráfica “Tráfico web en el último mes” en donde nuestra variable x representa el tiempo y la variable y los accesos por hora.

Con ayuda de la función *xticks* nos ayudará a poner una escala de semana en la variable x.

