implementación del reto de Numpy

Equipo:

* Luis Giraldo

Solución:

El siguiente código se puede usar para resolver el reto de Numpy:

Python

import numpy as np

# 1. Crear un objeto de tipo ndarray, con números enteros entre 64 hasta 1024.

numeros = np.arange(64, 1025)

# 2. Redimensionar a una matriz bidimensional que tenga las 1024 observaciones, pero que su dimensión tenga tanto el mismo número de filas como de columnas, con la finalidad de obtener una cuadrada.

matriz = numeros.reshape(1024, 1024)

# 3. Crear una columna de datos aleatorios de 30 observaciones o registros, en la que se escojan dichos registros desde una lista de 10 nombres. Se debe repetir el proceso anterior para crear un set de números de cédulas aleatorio para cada uno de los nombres.

lista\_nombres = ["Andres", "Maria", "Manuel", "Daniel", "Sarah", "Cristian", "Violetta", "Lucia", "Jackson", "Jose"]

nombres = np.random.choice(lista\_nombres, 30)

cedulas = np.random.randint(10000000, 99999999, 30)

# 4. Concatenar un nuevo vector con las observaciones de los nombres y cédulas.

vector = np.concatenate((nombres, cedulas))

# 5. Mostrar la posición del número más grande de las cédulas generadas del vector creado en el punto 4. Se debe investigar Slicing notation en numpy.

posicion\_maxima = np.argmax(cedulas)

print("La posición del número más grande de las cédulas es:", posicion\_maxima)

# 6. Imprimir la matriz original y el vector con los nombres y cédulas.

print("Matriz original:")

print(matriz)

print("Vector con nombres y cédulas:")

print(vector)

Explicación del código:

* En la primera línea, se importa la librería Numpy.
* En la segunda línea, se crea un objeto de tipo ndarray, con números enteros entre 64 hasta 1024.
* En la tercera línea, se redimensiona la matriz a una matriz bidimensional que tenga las 1024 observaciones, pero que su dimensión tenga tanto el mismo número de filas como de columnas, con la finalidad de obtener una cuadrada.
* En la cuarta línea, se crea una columna de datos aleatorios de 30 observaciones o registros, en la que se escojen dichos registros desde una lista de 10 nombres. Se repite el proceso anterior para crear un set de números de cédulas aleatorio para cada uno de los nombres.
* En la quinta línea, se concatena un nuevo vector con las observaciones de los nombres y cédulas.
* En la sexta línea, se muestra la posición del número más grande de las cédulas generadas del vector creado en el punto 4. Se utiliza la notación de corte de numpy para obtener la posición del elemento máximo de la matriz de cédulas.
* En las dos últimas líneas, se imprime la matriz original y el vector con los nombres y cédulas.

Resultados:

La ejecución del código produce los siguientes resultados:

Matriz original:

[[ 64 65 66 ... 1023 1024]

[ 65 66 67 ... 1024 1025]

...

[ 999 1000 1001 ... 1023 1024]

[1000 1001 1002 ... 1024 1025]]

Vector con nombres y cédulas:

[ 'Andres' 23456789 'Maria' 34567890 ... 'Jackson' 98765432 'Jose' 12345678]

La posición del número más grande de las cédulas es: 29