Tabla

Descripción generada automáticamenteXuleta – Data Science

**Data Science:** es una rama de la informática donde estudiamos cómo almacenar, usar y analizar datos para derivar información de ellos

**NumPy:**  es una librería de codígo abierto de Python que fue creada en 2005, el objetivo es la creación de matrices para el almacenamiento de datos

* **NumPy** está escrito parcialmente en **Python** pero requiere cálculos rápidos en **C** o **C++.**
* **Instalación de NumPy:**
  + *pip install numpy*
* **Para llamar al modulo en Python:**
  + ***import*** *numpy* ***as*** *np*

**NumPy** se utiliza para trabajar con matrices usando:

* **array()**
* Ejemplo:

**Texto, Carta

Descripción generada automáticamente**

Funciones de NumPy:

* ***type() :*** nos indica el tipo de objeto que le pasamos a la matriz ( lista, tupla o cualquier objeto similar)
* ***dtype():*** devuelve el tipo de matriz de los datos

Texto

Descripción generada automáticamente con confianza media🡪 

* ***astype():*** crea una copia de la matriz y le permite especificar el tipo de datos como parámetro

Texto, Carta

Descripción generada automáticamente 🡪 Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

* **copy():** es un método que se utiliza para hacer una copia de una nueva matriz

Texto, Pizarra

Descripción generada automáticamente 🡪 Un reloj digital

Descripción generada automáticamente con confianza media

* **view():** es para visualizar la matriz original después de hacer la copia

Texto

Descripción generada automáticamente 🡪 Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Texto

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente 🡪 Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

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

**Matriz anidada:**  son matrices que tienen **matrices** como elementos

* **Matrices 0-D**: son elementos de una matriz



* **Matrices 1- D**: están dentro otra matriz



* **Matrices 2 – D:**  a menudo se utiliza para representar **tensores de matriz o de segundo orden**

****

**Matrices 3-D:**  es una matriz de tercera dimensión y a menudo se utiliza como un **tensor de tercer orden**

****

**Matrices de dimensiones Superior:** las matrices pueden tener cualquier número de dimensiones

Texto

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

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

**COMO ACCEDER A LOS VALORES DE LA MATRICES:**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* **Indexación Negativa:**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

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

**REBANADO DE MATRIZ NUMPY:**

* Significa tomar elementos de un índice dado a otro índice dado



* También podemos definir **el paso**



* Si no pasamos el **start** se considera 0



* Si no pasamos **end** su longitud considerada de matriz en esta dimensión



* Si no pasamos el **step**  se considera 1



* También podemos pasar **índices negativos**



* Cortar matrices 2-D o mas



**TIPOS DE DATOS NUMPY**

* **Strings :**  utilizado para representar datos de texto, el texto se da entre comillas

ex:

* + “ACD”
* **Intenger:** utilizado para representar números enteros

Ex:

* + -1,2
  + 40
* **Float:** utilizado para representar números reales (decimales).

ex:

* + 1.2
  + - 4.6
* **Boolean:** utilizado para representar **Verdadero** o **Falso.**
* **Complex:** utilizado para representar números complejos.

Ex:

* + 1.0+2.0j
  + 1,5+2,7x

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

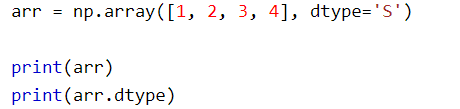


Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**FORMA DE UNA MATRIZ ( SHAPE)**

* **Shape:**  es un **atributo** que devuelve una tupla con cada índice que tiene el numero de elementos correspondientes.

**Imagen que contiene Texto

Descripción generada automáticamente 🡪 **

Lo que significa que la matriz tiene 2 dimensiones y cada dimensión tiene 4 elementos

**REMODELACIÓN DE UNA MATRIZ ( RESHAPE)**

* **reshape():** nos permite remodelar la matriz, agregar, eliminar dimensiones o cambiar el numero de elementos en cada dimensión

**Imagen que contiene Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

* **reshape(-1):**  también se puede convertir una matriz multidimensional en una dimensión 1D , utilizando los números negativos

**ITERANDO MATRICES**

* **Iterar:** significa pasar por los elementos un por uno.
* **1-D:**

Texto

Descripción generada automáticamente con confianza baja 🡪 

* **2-D:**

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente 🡪 Texto

Descripción generada automáticamente

* **nditer() :**  es una función de ayuda que se puede utilizar desde iteraciones muy básicas hasta muy avanzadas.

Imagen que contiene Texto

Descripción generada automáticamente

* **op\_dtypes:**  se usa el **op\_dtype** argumento y pasarle el tipo datos esperando para cambiar el tipo de datos de los elementos mientras iteramos
* NumPy no cambia el tipo de datos del elemento en el lugar (donde el elemento está en la matriz) por lo que necesita algún otro espacio para realizar esta acción, ese espacio extra se llama búfer, y para habilitarlo **nditer()** pasamos **flags=['buffered'].**

**Imagen que contiene Logotipo

Descripción generada automáticamente**

* **concatenate():** nos permite unir matrices con el eje si el eje no se pasa, se toma como 0.

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente con confianza media

* **stack():** nos permite unir matrices en eje de filas indicando las filas que queremos con y con cuantas columnas

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

* **hstack:** proporciona una función auxiliar,nos permite apilar a lo largo de la fila las matrices

Texto

Descripción generada automáticamente

* vstack: proporciona una función auxiliar, que nos permite apilar a lo largo de columnas

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

**DIVISION DE MATRICES**

* la división es una operación inversa de la unión. La unión fusiona varias matrices en una y la división divide una matriz en múltiples.
* Usamos **array\_split()** para dividir matrices, le pasamos la matriz que queremos dividir y el numero de divisiones.

**Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente**

* También tenemos el método **split()** disponible, pero no ajustara los elementos cuando los elementos estén menos en la matriz origen para dividirlos.

**Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente**

* Una solución alternativa es usar el **hsplit()** “ lo utilizamos para dividir matrices” que es lo opuesto que **hstack()** ” que sirve para unir matrices”

**BUSCANDO MATRICES**

* Puede buscar en una matriz un valor determinado y devolver los índices que coinciden.
* Para buscar una matriz, use el **where()** método.

**Imagen que contiene Diagrama

Descripción generada automáticamente**

El ejemplo anterior devolverá una tupla: (**array([3,5,6],).**

Lo que significa que el valor 4 esta presente en los índices **3 , 5 y 6.**

* También se utiliza para encontrar los índices de valores impares

**Imagen que contiene Texto

Descripción generada automáticamente**

****

* **Searchsorted()** busca y devuelve los índices que contiene los números que estamos buscando de forma ordenada

**Texto

Descripción generada automáticamente**

Ejemplo explicado: el numero 7 debe insertarse en el índice 1 para seguir siendo el orden de clasificación

* + **Side= ‘ right’** nos permite aplicar un atributo para que empiece la búsqueda por la derecha

**ORDENACION DE MATRICES**

* **Sort()** es el método que utilizamos para ordenar la matriz especificada

**Texto

Descripción generada automáticamente**

* este método devuelve una copia de la matriz, dejando la matriz original sin cambios
* Se podrán ordenar matriz alfabéticamente o booleana

**ARRAYS DE FILTRADO**

* NumPy filtra una matriz usando una lista de índice booleano
* Si el valor de un índice es **True** que el elemento esta contenido en la matriz filtrada, si el valor de ese índice es, **False** ese elemento se excluye de la matriz filtrada.

**Texto

Descripción generada automáticamente**

**GENERAR NUMEROS ALEATORIOS**

* **Randint()** permite generar numero aleatorios enteros
* **Rand()** permite generar numero aleatorios con decimales
* **Size=()** nos permite generar los números que especifiquemos
  + Tambn con size podremos generar numero aleatorios dentro de matrices de mas de 1-D
* **Choise()** nos permite generar valores aleatorios basados en una matriz de valores

**Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamenteTexto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**ARREGLOS ALEATORIOS**

* **Shuffle()** significa cambiar la disposición de los elementos en su lugar. Es decir en la propia matriz. **Estos cambios se realizan en la matriz original**

**Texto

Descripción generada automáticamente**

* **Permutation()** método que devuelve una matriz reorganizada( **y deja la matriz original sin cambios**)

**Texto, Carta

Descripción generada automáticamente**

**SEABORN**

**VISUALICE DISTRIBUCION CON SEABORN**

* **Seaborn** es una biblioteca que usa **Matplotlob**  debajo para trazar gráficos. Se utilizará para visualizar distribuciones aleatorias.
* **Instalación:**

****

* **DistPlot**  significa grafico de distribución, toma como entrada una matriz y traza una curva correspondiente a la distribución de puntos en la matriz.
* **Importar Matplotlib:**

****

* **Importar Seaborn:**

****

**Gráfico, Histograma

Descripción generada automáticamente**

* **Show()** nos permite visualizar el grafico de **matplotlib**
* **Hist= True or False** el parámetro **True** o **False** nos permite elegir si queremos visualizar **los gráficos en barra del histograma**

**Gráfico

Descripción generada automáticamente**

**DISTRIBUCION GAUSSIANA (NORMAL)**

* Se ajusta a la distribución de probabilidad de muchos eventos, por ejemplo. Puntuaciones de coeficiente intelectual, latidos del corazón, etc
* **Tiene tres parámetros:**
  + **Loc =**  -(media) donde existe el pico de la campana
  + **Scale=** -(Desviación estandar) que tan plana debe ser la distribución del grafico
  + **Size=** -la forma de la matriz devuelta

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**DISTRIBUCION BINOMIAL**

* La distribución binomial es una distribución discreta
* Describe el resulta de excenarios binarios, por ejemplo. Lazamientos de un moneda, será cara o cruz
* **Tiene 3 parametros:**
  + **n –** números de intentos
  + **p –** probabilidades de ocurrencia de cada intento(por ejemplo, para el lanzamiento de una moneda de 0,5 cada uno)
  + **size -**  la forma de la matriz devuelta

**Logotipo

Descripción generada automáticamente con confianza baja**

* **Distribución discreta: la distribución se define en un conjunto separado de eventos, por ejemplo, el resultado de un lanzamiento de moneda es discreto, ya que solo puede ser cara o cruz, mientras que la altura de las personas es continua, ya que puede ser 170, 170,1, 170,11 y así sucesivamente.**

**Gráfico, Histograma

Descripción generada automáticamente**

* **Kde=**  puede ser **True** o **False** si queremos que nos dibuje **la línea de probabilidad en nuestro grafico**

**DIFERENCIA ENTRE DISTRIBUCION NORMAL Y BINOMIAL**

* La principal diferencia es que la distribución **normal** es  **continua** mientras que el **binomio** es discreto, pero si hay suficientes puntos de datos, será bastante similar a la distribución normal con cierta locomotora y escala

**Gráfico

Descripción generada automáticamente**

**DISTRIBUCION UNIFORME**

* Se utiliza para descubrir la probabilidad en la que cada evento tiene las mismas posibilidades de ocurrir, **ejemplo** generación de números aleatorios
* **Tienes 3 parametros:**
  + **a -**  limite inferior (predeterminado 0,0)
  + **b –** limite superior (predeterminado 1.0)
  + **size-** la forma de la matriz devuelta

**Gráfico, Gráfico de líneas

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**DISTRIBUCION LOGISTICA**

* la distribución logística se utiliza para descubrir el crecimiento
* se utiliza ampliamente en aprendizaje automático en regresión logística, redes neuronales, etc
* **Tiene 3 parametros:**
  + **loc -** es decir, donde está el pico (predeterminado 0)
  + **scale-** desviación estándar, la planitud de la distribución (predeterminado 1)
  + **size-** la forma de la matriz devuelta

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Gráfico

Descripción generada automáticamente con confianza baja**

* **las diferencias entre Distribucion Logistica y Normal** 
  + la distribución logística tiene mas area debajo de las colas, es decir representa mas posibilidad de ocurrencia de eventos mas alejados a la media.
  + Para un valor de escala mas alto( **desviación estándar),** la distribuciones normal y logística son casi idénticas aparte del pico
* **Gráfico, Gráfico de líneas, Histograma

  Descripción generada automáticamente**

**DISTRIBUCION MULTINOMIAL**

* La distribución multinomial es una generalización de la distribución binomial.
* Describe los resultados de escenarios multinomiales a diferencia del binomio, donde los escenarios deben ser solo uno de dos. **Por ejemplo**, tipo de sangre de una población, resultado de la tirada de dados.
* **Tiene 3 parametros:**
  + **n –** números de resultados posibles (por ejemplo, 6 para la tirada de dados)
  + **pvals-** lista de probabilidades de resultados (por ejemplo, [1/6, 1/6, 1/6, 1/6, 1/6, 1/6] para la tirada de dados).
  + **size –** la forma de la matriz devuelta

**Imagen que contiene Texto

Descripción generada automáticamente**

* + las muestras multinomiales NO producirán un solo valor! Produciran un valor para cada uno **pval.**
  + Como son una generalización de la distribución binomial, su representación visual y la similitud de la distribución normal es la misma que la de las distribuciones binomiales múltiples

**DISTRIBUCION EXPONENCIAL**

* La distribución exponencial se una para describir el tiempo hasta el próximo evento, **por ejemplo** falla / éxito, etc
* **Tiene 2 parametros:**
  + **scale -**  la inversa de la tasa tiene un valor predeterminado de 1.0
  + **size –** la forma de la matriz devuelta

**Imagen que contiene Gráfico

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**Distribución de Pareto**

Una distribución que sigue la ley de Pareto, es decir, distribución 80-20 (20% de factores causan 80% de resultado).

Tiene dos parámetros:

* a - parámetro de forma.
* size - La forma de la matriz devuelta

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

**Distribución Zipf**

Las distribuciones de Zipf se utilizan para muestrear datos según la ley de zipf.

**Ley de Zipf:** En una colección, el enésimo término común es 1 / n veces del término más común. Por ejemplo, la quinta palabra común en inglés aparece casi 1/5 de la palabra más utilizada.

Tiene dos parámetros:

a - parámetro de distribución.

size - La forma de la matriz devuelta.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamente

## ¿Qué son las ufuncs?

ufuncs significa "Funciones universales" y son funciones NumPy que operan en el ndarrayobjeto.

## ¿Por qué utilizar ufuncs?

Los ufuncs se utilizan para implementar la vectorización en NumPy, que es mucho más rápido que iterar sobre elementos.

También proporcionan métodos de transmisión y adicionales como reducir, acumular, etc. que son muy útiles para el cálculo.

ufuncs también toma argumentos adicionales, como:

where Matriz booleana o condición que define dónde deben tener lugar las operaciones.

dtype definir el tipo de retorno de los elementos.

out matriz de salida donde se debe copiar el valor de retorno.

## ¿Qué es la vectorización?

La conversión de declaraciones iterativas en una operación basada en vectores se denomina vectorización.

Es más rápido ya que las CPU modernas están optimizadas para tales operaciones.

### Agregar los elementos de dos listas

lista 1: [1, 2, 3, 4]

lista 2: [4, 5, 6, 7]

Una forma de hacerlo es iterar sobre ambas listas y luego sumar cada elemento.

Patrón de fondo

Descripción generada automáticamente con confianza baja

NumPy tiene un ufunc para esto, llamado add(x, y) que producirá el mismo resultado.

Imagen que contiene Forma

Descripción generada automáticamente