

Practica 1- Introducción a numpy

February 9, 2021

PRÁCTICA 1: MINERÍA DE DATOS: INTRODUCCIÓN A NUMPY

0.0.1 Ejercicio 1: Implemente mediante un programa en python la asignación del reparto de escaños de una circunscripción electoral usando la ley de D'Hont. Los datos se pueden introducir por teclado o leerse desde un fichero.

```
[ ]: import numpy as np
import sys
```

0.0.2 Ejercicio 2: Implemente un programa Python que genere aleatoriamente una matriz de valores reales de un tamaño indicado por teclado. Sobre una matriz generada realice las siguientes operaciones:

2.1) Obtenga los valores máximos y mínimos de la matriz.

2.2) Use el producto escalar para obtener el ángulo formado por dos vectores fila o columna solicitados por teclado

```
[31]: import numpy as np

rows = input("Introduce número de filas")
columns = input("Introduce el número de columnas")

matriz = np.random.rand(int(rows), int(columns))
matriz = np.array(matriz)
print(matriz)

# 2.1 Valores máximos y mínimos de la matriz
print("Valor máximo de la matriz", matriz.max())
print("Valor mínimo de la matriz", matriz.min())

#2.2 Producto escalar para obtener el ángulo formados por dos vectores fila o
↳ columna solicitados
decision = input("Introduce un 0 para filas u otra cosa para columnas")
if(decision ==1):
    vector1 = input("Introduce un entero referente a la fila de la matriz")
    vector2 = input("Introduce un entero referente a la fila de la matriz")
    if(int(vector1) < matriz[0].size and int(vector2) < matriz[0].size):
```

```

        u= matriz[int(vector1)]
        v= matriz[int(vector2)]
        dotpr=np.sum(u*v)
        modu=np.sqrt(sum(u*u))
        modv=np.sqrt(sum(v*v))
        aux=dotpr/(modu*modv)
        angulo=np.arccos(aux)
        print("El angulo formado por los dos vectores es ",(angulo*180)/3.
↪141592653589793,"°")
    else:
        print("Indices no validos")
else:
    vector1 = input("Introduce un entero referente a la columna de la matriz")
    vector2 = input("Introduce un entero referente a la columna de la matriz")
    if(int(vector1) < matriz[0].size and int(vector2) < matriz[0].size):
        u= matriz[:,int(vector1)]
        v= matriz[:,int(vector2)]
        dotpr=np.sum(u*v)
        modu=np.sqrt(sum(u*u))
        modv=np.sqrt(sum(v*v))
        aux=dotpr/(modu*modv)
        angulo=np.arccos(aux)
        print("El angulo formado por los dos vectores es ",(angulo*180)/3.
↪141592653589793,"°")
    else:
        print("Indices no validos")

```

Introduce número de filas5

Introduce el número de columnas5

5

Valor máximo de la matriz 0.9921854591799903

Valor mínimo de la matriz 0.025264361999718044

Introduce un 0 para filas u otra cosa para columnas1

Introduce un entero referente a la columna de la matriz1

Introduce un entero referente a la columna de la matriz2

El angulo formado por los dos vectores es 35.36628735593343 °

0.0.3 Ejercicio 3: Implemente un programa Python que lea una matriz de número reales desde el teclado de una dimensión dada. A partir de la matriz leída debe calcular la siguiente información:

3.1) Máximo por filas y por columnas

3.2) Determinante de la matriz

3.3) Rango de la matriz

```
[65]: import numpy as np

tam = input("Introduce número de filas y columnas")

matriz = np.array([], dtype=float)
x=0
# Rellenamos la matriz
for i in range(0,int(tam)):
    aux=[]
    for j in range(0, int(tam)):
        print("Introduce el valor de ",i,",",j,":")
        value=input()
        aux.append(float(value))
    if(x==0):
        matriz=aux
        x=1
    else:
        matriz=np.vstack([matriz,aux])
print(matriz)

for i in range(np.size(matriz,1)):
    print("Valor maximo de la fila "+str(i)+": " , np.max(matriz[i,:]))
    print("Valor minimo de la fila "+str(i)+": " , np.min(matriz[i,:]))

for i in range(np.size(matriz,0)):
    print("Valor maximo de la fila "+str(i)+": " , np.max(matriz[:,i]))
    print("Valor minimo de la fila "+str(i)+": " , np.min(matriz[:,i]))

print("Determinante de la matriz: "+str(round(np.linalg.det(matriz))))
print("El rango de la matriz es ", np.linalg.matrix_rank(matriz))
```

```
Introduce número de filas y columnas5
Introduce el valor de  0 , 0 :
5
Introduce el valor de  0 , 1 :
2
Introduce el valor de  0 , 2 :
5
Introduce el valor de  0 , 3 :
4
Introduce el valor de  0 , 4 :
7
Introduce el valor de  1 , 0 :
8
Introduce el valor de  1 , 1 :
5
Introduce el valor de  1 , 2 :
```

```

6
Introduce el valor de 1 , 3 :
5
Introduce el valor de 1 , 4 :
5
Introduce el valor de 2 , 0 :
5
Introduce el valor de 2 , 1 :
5
Introduce el valor de 2 , 2 :
5
Introduce el valor de 2 , 3 :
5
Introduce el valor de 2 , 4 :
5
Introduce el valor de 3 , 0 :
5
Introduce el valor de 3 , 1 :
5
Introduce el valor de 3 , 2 :
5
Introduce el valor de 3 , 3 :
5
Introduce el valor de 3 , 4 :
5
Introduce el valor de 4 , 0 :
5
Introduce el valor de 4 , 1 :
5
Introduce el valor de 4 , 2 :
5
Introduce el valor de 4 , 3 :
5
Introduce el valor de 4 , 4 :
5
[[5. 2. 5. 4. 7.]
 [8. 5. 6. 5. 5.]
 [5. 5. 5. 5. 5.]
 [5. 5. 5. 5. 5.]
 [5. 5. 5. 5. 5.]]
Valor maximo de la fila 0: 7.0
Valor minimo de la fila 0: 2.0
Valor maximo de la fila 1: 8.0
Valor minimo de la fila 1: 5.0
Valor maximo de la fila 2: 5.0
Valor minimo de la fila 2: 5.0
Valor maximo de la fila 3: 5.0
Valor minimo de la fila 3: 5.0

```

```

Valor maximo de la fila 4: 5.0
Valor minimo de la fila 4: 5.0
Valor maximo de la fila 0: 8.0
Valor minimo de la fila 0: 5.0
Valor maximo de la fila 1: 5.0
Valor minimo de la fila 1: 2.0
Valor maximo de la fila 2: 6.0
Valor minimo de la fila 2: 5.0
Valor maximo de la fila 3: 5.0
Valor minimo de la fila 3: 4.0
Valor maximo de la fila 4: 7.0
Valor minimo de la fila 4: 5.0
Determinante de la matriz: 0.0
El rango de la matriz es 3

```

0.0.4 Ejercicio 4: Implemente un programa Python que lea una matriz de número enteros desde teclado de una dimensión dada. A partir de la matriz leída debe calcular la siguiente información:

4.1) Moda de la matriz

4.2) Media de todos los elementos de la matriz

```

[71]: import numpy as np
      from scipy import stats as st

      m = np.array([], dtype=float)
      rows = input("Introduce el número de filas")
      cols = input("Introduce el número de columnas")
      x=0

      # Rellenamos la matriz
      for i in range(0,int(rows)):
          aux=[]
          for j in range(0, int(cols)):
              print("Introduce el valor de ",i,",",j,":")
              value=input()
              aux.append(float(value))
          if(x==0):
              m =aux
              x=1
          else:
              m =np.vstack([m ,aux])
      print(m)

      print("La Moda de la matriz es: ", st.mode(m))
      print("La Media de la matriz es: ", np.average(m))

```

Introduce el número de filas2

```

Introduce el número de columnas2
Introduce el valor de  0 , 0 :
2
Introduce el valor de  0 , 1 :
2
Introduce el valor de  1 , 0 :
2
Introduce el valor de  1 , 1 :
2
[[2. 2.]
 [2. 2.]]
La Moda de la matriz es: ModeResult(mode=array([[2., 2.]]), count=array([[2,
2]]))
La Media de la matriz es:  2.0

```

0.0.5 Ejercicio 5: Implemente un programa Python que lea una matriz de números reales desde un fichero texto con formato libre. Una vez leído el programa debe obtener la inversa de la matriz y realizar un producto matricial para comprobar que el cálculo de la inversa es correcto.⁶

```

[84]: import numpy as np
import math
def MatrixFiledata(nameFile,deli):
    openFile = np.loadtxt("./Datos/"+nameFile, delimiter=deli)
    return openFile

file = input("\n Inserte el nombre del fichero: ")
delimiter = input("\n Inserte el delimitador del fichero, en nuestro caso un_
↳espacio: ")

mat = MatrixFiledata(file,delimiter)
print("\n Matrix")
print(mat)

inversa = np.linalg.inv(mat)
print("\n --> Inversa: ")
print(inversa)
print("\n --> Producto m * inversa: ")
multiplicada = np.matmul(m,inversa)
print(multiplicada)

identidad = np.identity(np.size(mat,1))
print(identidad)
if np.array_equal(identidad,multiplicada)==True:
    print("Es correcto\n")

```

Inserte el nombre del fichero: Fichero1.txt

Inserte el delimitador del fichero, en nuestro caso un espacio:

```
Matrix
[[ 3.  4. -1.]
 [ 2.  0.  1.]
 [ 1.  3. -2.]]

--> Inversa:
[[-0.6  1.   0.8]
 [ 1.  -1.  -1. ]
 [ 1.2 -1.  -1.6]]

--> Producto m * inversa:
[[ 1.00000000e+00  2.22044605e-16  0.00000000e+00]
 [ 0.00000000e+00  1.00000000e+00 -2.22044605e-16]
 [ 0.00000000e+00 -4.44089210e-16  1.00000000e+00]]
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

[]: