

# Reporte de Proyecto 1

Alumno: Gael Peña Fonseca

Preguntas:

1) ¿Cuál es la página web más importante según el algoritmo si se inicia con el vector  $(1/n, \dots, 1/n)$ ?

Para contestar esta pregunta observemos lo que nos imprime al usar

```
def calcular_PR(matriz, Epsilon=1e-6):

    n = len(matriz)
    pi = np.ones((1, n)) / n
    pi_1 = []

    for j in range(n):

        valor_columna = sum(pi[0][k] * matriz[k][j] for k in range(n))
        pi_1.append(valor_columna)
    pi_1 = np.array([pi_1])
    while np.linalg.norm(pi - pi_1) > Epsilon:
        pi = pi_1
        SSJ_pi_1 = []
        for j in range(n):
            #Calcular el valor para la columna j
            valor_columna = sum(pi[0][k] * matriz[k][j] for k in range(n))
            SSJ_pi_1.append(valor_columna)

        pi_1 = np.array([SSJ_pi_1])
    return np.array(pi[0])

calcular_PR(M)
```

Esto nos imprime la siguiente matriz:

[7.33320314e-07, 1.01764169e-06, 6.33367242e-07, 1.36085707e-06,

5.34701306e-07, 8.20661192e-07, 1.37005441e-06, 8.33927232e-07,  
1.10169011e-06, 9.60685274e-07, 5.04500778e-07, 5.24041574e-07,  
7.54179322e-07, 7.57679334e-07, 9.73391912e-07, 3.60576923e-02,  
3.60576923e-02, 3.60576923e-02, 3.60576923e-02, 4.80769231e-02,  
3.60576923e-02, 3.60576923e-02, 3.60576923e-02, 4.80769231e-02,  
3.60576923e-02, 6.15371735e-01]

Donde 6.15371735e-01 representa el numero más grande y no solo eso, sino también hace referencia al índice 26 cuya página web asociada es [deferalreserve.gob](http://deferalreserve.gob)

2) ¿La página de la pregunta 1 sigue siendo la más importante?

En este caso ya no porque solo está contando las que tienen terminación .ru, la cual es [vedomosti.ru](http://vedomosti.ru) cuyo valor asociado es 0.12500005

3) Utilice los valores 0.5, 0.85, 1 para d y reporte los

resultados. De los tres valores anteriores para d, ¿cuál considera que es el mejor (según los ordenamientos obtenidos) y por qué?

Esto lo podemos analizar por lo que nos imprime:

```
def PR_d(matriz, d, Epsilon_tercios=1e-6):  
  
    n = len(matriz)  
    u = np.ones((n, n)) / n  
    matriz_mplus = d * matriz + (1 - d) * u  
  
    return calcular_PR(matriz_mplus, Epsilon_tercios)  
  
valores_d = [0.5, 0.85, 1]  
  
resultados_d = {}  
  
for d in valores_d:  
    resultados_d[d] = PR_d(matriz_kaioken, d)  
    print("PageRank con", d)  
    print(resultados_d[d])
```

Es decir:

PageRank con 0.5

```
[0.03276549 0.03755851 0.03342875 0.04702054 0.03173843 0.03527269  
0.04338471 0.03600769 0.0395377 0.03503242 0.0298904 0.03074748  
0.03496692 0.03363201 0.04040186 0.03719881 0.03783815 0.03708623  
0.03722273 0.04384511 0.03722273 0.03705013 0.0363509 0.04385451  
0.03694609 0.07399901]
```

PageRank con 0.85

```
[0.02375518 0.03083679 0.02284086 0.041152 0.02072416 0.02682135  
0.03861176 0.02747867 0.0330701 0.02855831 0.01910243 0.01997687  
0.02556454 0.02504939 0.03183435 0.03643144 0.03676753 0.03632114  
0.03648717 0.04684119 0.03648717 0.03626792 0.03593937 0.04688206  
0.0361904 0.20000787]
```

PageRank con 1

```
[7.33320314e-07 1.01764169e-06 6.33367242e-07 1.36085707e-06  
5.34701306e-07 8.20661192e-07 1.37005441e-06 8.33927232e-07  
1.10169011e-06 9.60685274e-07 5.04500778e-07 5.24041574e-07  
7.54179322e-07 7.57679334e-07 9.73391912e-07 3.60576923e-02  
3.60576923e-02 3.60576923e-02 3.60576923e-02 4.80769231e-02  
3.60576923e-02 3.60576923e-02 3.60576923e-02 4.80769231e-02  
3.60576923e-02 6.15371735e-01]
```

El valor d que nos conviene es .85 porque el valor más grande empieza en la posición de los decimos, a diferencia de 1 y 0.5 que no sería posible observarlo