**Jorge Adrian Padilla Velasco**

**A00570894**

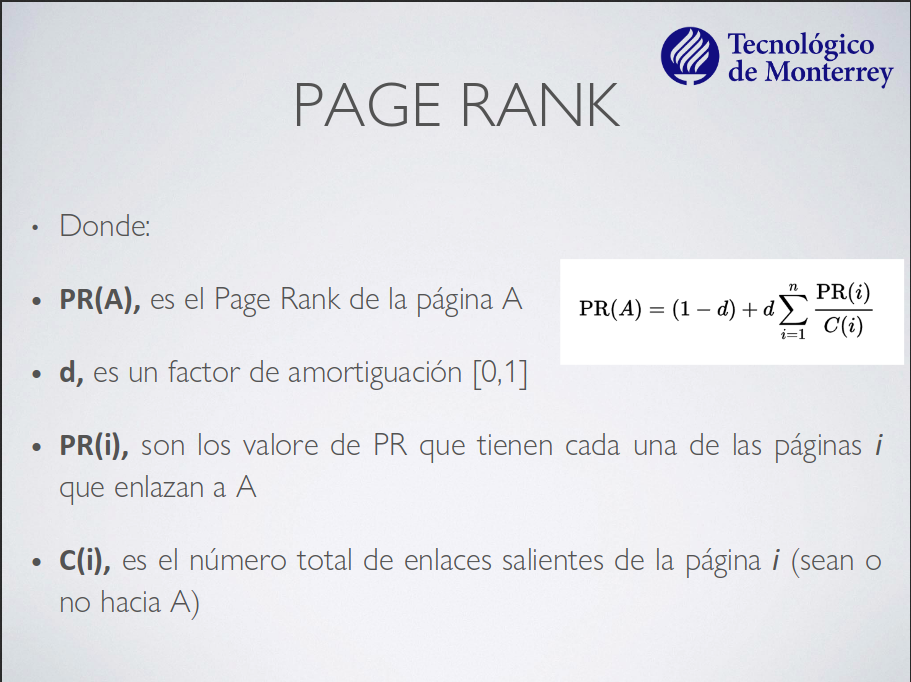
**9 de octubre de 2018**

**Práctica 2.3**

**Introducción**

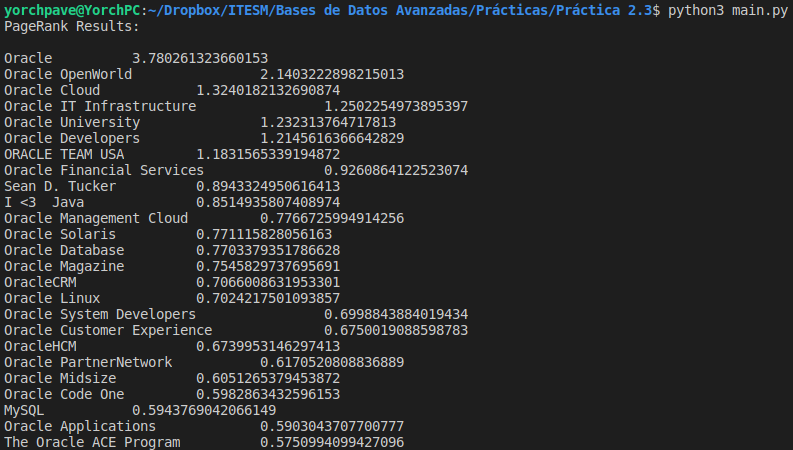
PageRank es una marca registrada y patentada por Google el 9 de enero de 1999 que ampara una familia de algoritmos utilizados para asignar de forma numérica la relevancia de los documentos (o páginas web) indexados por un motor de búsqueda.

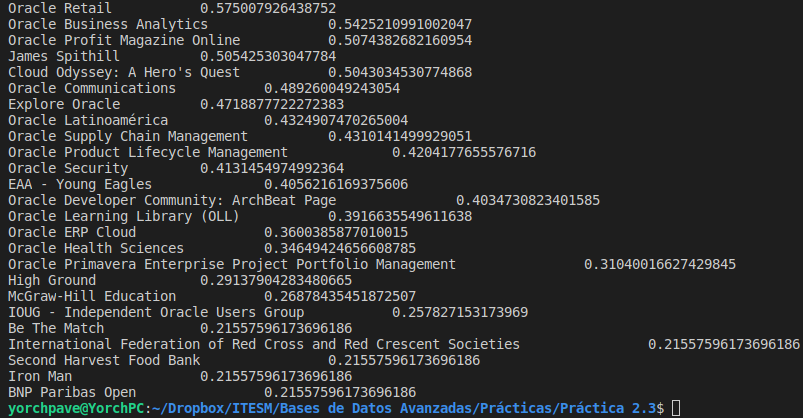
**Modelo matemático**



**Desarrollo**

Al implementar el algoritmo (se muestra a continuación), tomé el factor de amortiguación default, es decir, d = 0.85. Tras evaluar los archivos CSV de Oracle (nodos y aristas) con el código, obtuve los siguientes resultados:





**Código:**

**Graph.py**

class Node(object):

def \_\_init\_\_(self, id, name, pageRank = 0):

self.id = id

self.name = name

self.adj = {}

self.invAdj = {}

self.connected = 0

self.pageRank = pageRank

def addConnection(self, node, cost):

self.adj[node.id] = [cost, node]

node.invAdj[self.id] = self

self.connected += 1

def updatePageRank(self, d = 0.85):

accum = 0

for node in self.invAdj.values():

accum += node.pageRank / node.connected

self.pageRank = (1 - d) + (d) \* accum

class Graph (object):

def \_\_init\_\_(self, name="Graph"):

self.name = name

self.size = 0

self.nodes = {}

def insertNode(self, node):

self.nodes[node.id] = node

self.size += 1

def pageRank(self, iterations = 0):

keys = self.nodes.keys()

for \_ in range(iterations):

for key in keys:

self.nodes[key].updatePageRank(d=0.85)

pageRankDictionary = {}

for key in keys:

pageRankDictionary[self.nodes[key].name] = self.nodes[key].pageRank

keyValue = zip(pageRankDictionary.keys(), pageRankDictionary.values())

return dict(sorted(keyValue, key = lambda x: x[1], reverse=True))

def fromCSV(self, pathToNodes, pathToEdges):

import pandas as pd

csvNodes = pd.read\_csv(pathToNodes)

for i, row in csvNodes.iterrows():

self.insertNode(Node(row['id'], row['label']))

csvEdges = pd.read\_csv(pathToEdges)

for i, row in csvEdges.iterrows():

self.nodes[row['Source']].addConnection(self.nodes[row['Target']], 1)

**main.py**

from Graph import \*

if \_\_name\_\_ == '\_\_main\_\_':

myGraph = Graph("Pages")

myGraph.fromCSV("Oracle\_Nodes.csv", "Oracle\_Edges.csv")

print("PageRank Results: \n")

pageRank = myGraph.pageRank(50)

for key, value in zip(pageRank.keys(), pageRank.values()):

print(key + "\t\t" + str(value))

del myGraph

**¿Cuál es el nodo más influyente?**

Como podemos ver, Oracle es el nodo más influyente con un pagerank PR = 3.78. Esto quiere decir que es el nodo más importante del grafo porque Oracle apunta a cierto número de páginas importantes y, al mismo tiempo, páginas importantes o relevantes apuntan a Oracle.

**¿Cuál es el nodo menos influyente?**

Se puede decir que existe un quíntuple empate entre las páginas:

1. Be The Match
2. International Federation of Red Cross and Red Crescent Societies
3. Second Harvest Food Bank
4. Iron Man
5. BNP Paribas Open

con un PageRank PR = 0.21. Estas fueron las páginas menos importantes ya que, a pesar de apuntar a páginas relevantes, estas no son apuntadas de vuelta por páginas relevantes.

**Conclusiones**

Analizando los resultados (imágenes anexas), me llamó la atención que los nodos más influyentes fueran de la misma empresa de Oracle. Aunque por una parte, esto es de esperarse ya que Oracle es una empresa altamente reconocida a nivel Mundial, por lo que muchas páginas relevantes apuntan a las diferentes páginas de Oracle. Mientras que, los nodos menos influyentes, no tienen nada que ver con Oracle y no son apuntadas por páginas relevantes, razón por lo cual su influencia es mínima.

**Referencias**

<https://es.wikipedia.org/wiki/PageRank>

PAGE RANK M. en C. Rodolfo Rubén Álvarez González