Sistemas expertos

Nombre: Rafael Angamarca

Fecha: 09/07/2021

La similitud coseno es una **medida de la similitud existente entre dos vectores** en un espacio que posee un producto interior con el que se evalúa el valor del coseno del ángulo comprendido entre ellos. Esta función trigonométrica proporciona un valor igual a 1 si el ángulo comprendido es cero, es decir si ambos vectores apuntan a un mismo lugar

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

jupyter Deberes 2 segundo Interciclo (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Markdown

UNIVERSIDAD POLITECNICA SALESIANA

Nombre: Rafael Angamarca.

Similitud de Coseno

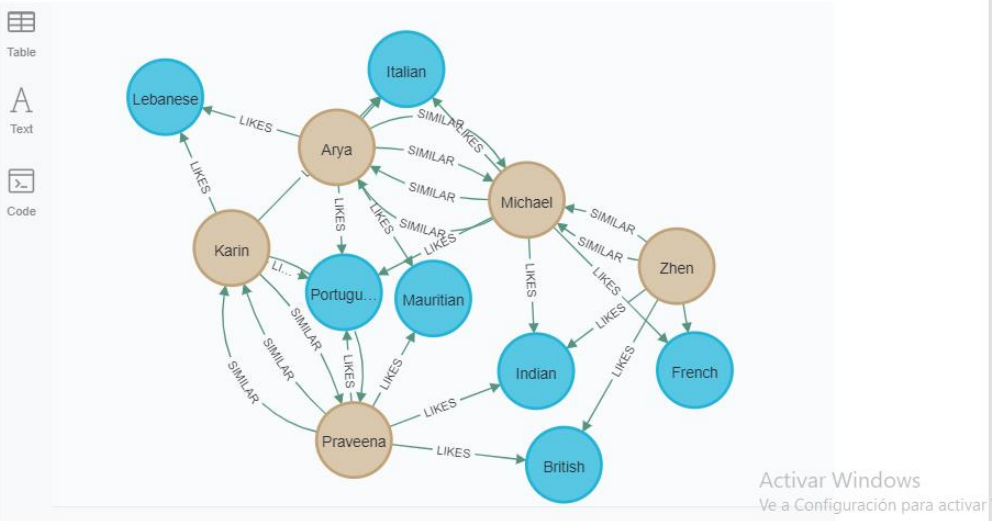
In [3]:
```
from neo4j import GraphDatabase
username = ('neo4j')
password = ('cuenca')
uri = "bolt://localhost:7687"
driver = GraphDatabase.driver(uri, auth=(username, password))
session2= driver.session()
cocina =session2.run("CREATE (french:Cuisine {name:'French'}), (italian:Cuisine {name:'Italian'}),  (indian:Cuisine {name:'Indian
persona = session2.run("CREATE (zhen:Person {name: 'Zhen'}), (praveena:Person {name: 'Praveena'}), (michael:Person {name:'Michael
relacion = session2.run("CREATE (praveena)-[:LIKES {score: 9}]->(indian),  (praveena)-[:LIKES {score: 7}]->(portuguese), (praveer
relacion2= session2.run ("CREATE (zhen)-[:LIKES {score: 10}]->(french), (zhen)-[:LIKES {score: 6}]->(indian), (zhen)-[:LIKES {sco
relacion3= session2.run (" CREATE (michael)-[:LIKES {score: 8}]->(french), (michael)-[:LIKES {score: 7}]->(italian), (michael)-[:
relacion4 = session2.run (" CREATE (arya)-[:LIKES {score: 10}]->(lebanese),  (arya)-[:LIKES {score: 10}]->(italian),  (arya)-[:LI
relacion5 = session2.run (" CREATE (karin)-[:LIKES {score: 9}]->(lebanese), (karin)-[:LIKES {score: 7}]->(italian), (karin)-[:LIK

session2.close()
driver.close()
```


In [ ]:
```
    #obtener similitud entre los gustos de la persona
gustos = session2.run(" MATCH (p:Person), (c:Cuisine)OPTIONAL MATCH (p)-[likes:LIKES]->(c) WITH {item:id(p), weights: collect(coa
print (gustos)
for c in gustos:
    print (c, 'ln')
print (' ')
print (' ')
print (' ')
similitudGustos = session2.run (" MATCH (p:Person), (c:Cuisine)  OPTIONAL MATCH (p)-[likes:LIKES]->(c) WITH {item:id(p), weights:
for c2 in similitudGustos:
    print (c2, "\n")


totalsimilitud= session2.run(" MATCH (p:Person), (c:Cuisine)  OPTIONAL MATCH (p)-[likes:LIKES]->(c)  WITH {item:id(p), weights: c
for re in   totalsimilitud:
    print (re, "\n")
```

Arbol de nodos



Bibliografia

https://neo4j.com/docs/graph-algorithms/current/labs-algorithms/cosine/