



Prueba 2

17/01/2021

Jonathan Atancuri

Objetivo:

- Consolidar los conocimientos adquiridos en clase de los sistemas expertos basados en casos utilizando Neo4J.

Enunciado:

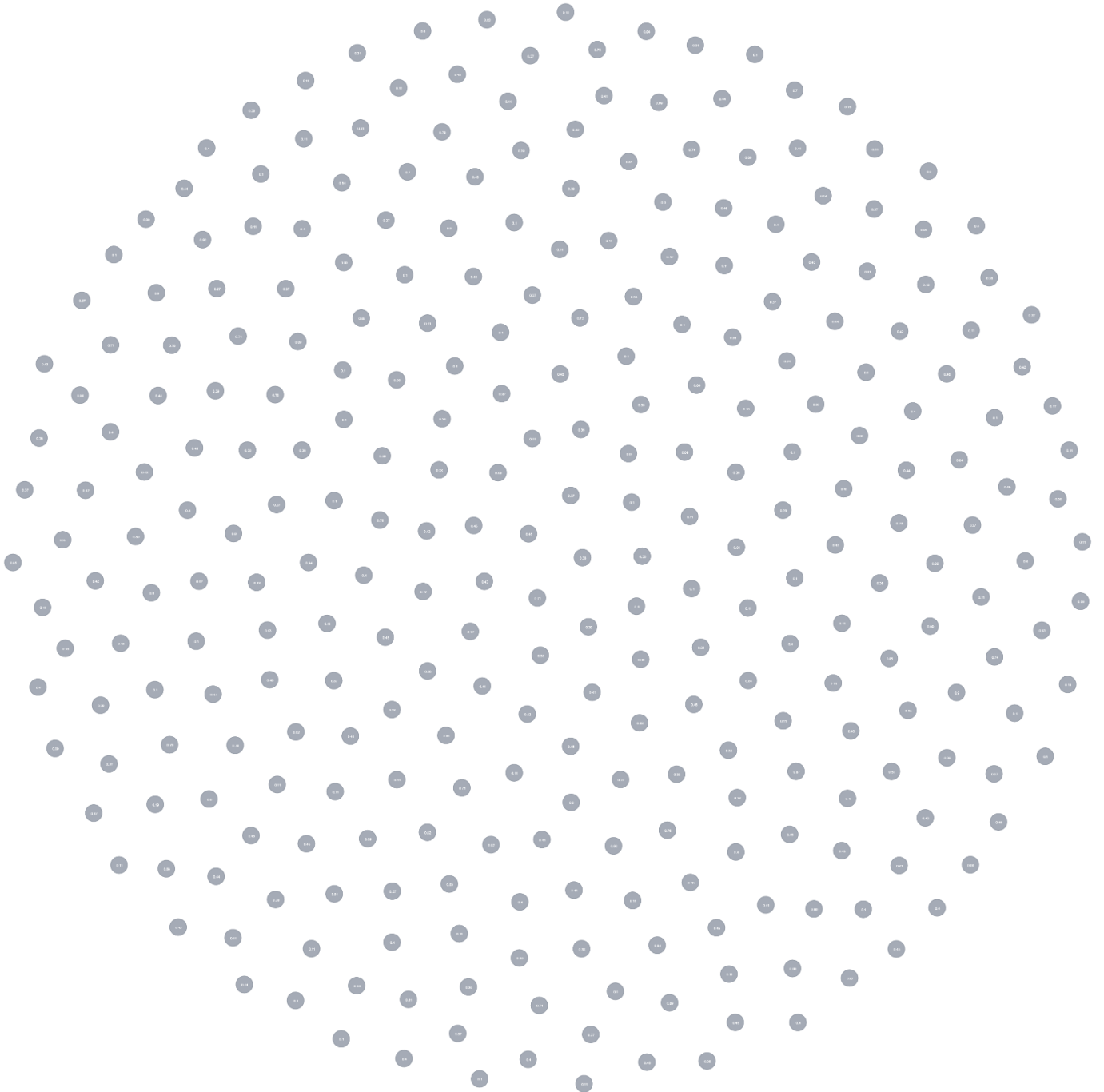
- Diseñe y desarrolle un algoritmo Knn en Neo4j para:
 - **Fila B - 1:** Este es un conjunto de datos de empleados en una empresa y el resultado es estudiar sobre la deserción de los empleados, para ello se debe descargar los datos del siguiente link: <http://smalldatabrains.com/wp-content/uploads/2018/03/data.csv> [2].
- Ingresar cada uno de los datos en un nodo y obtener el grado de similitud se recomienda utilizar la distancia Euclidiana o Person, una vez obtenido la similitud ingresar datos de prueba para validar (Máximo 3 datos).
- Generar otro entorno en donde solo ingrese el 70% de los datos y validar con el 30%.
- Agregar el grafico con los nodos conformados.
- El proceso de programación desarrollado deberá considerar los siguientes aspectos:
 - Se deberá tener un archivo que tenga todos los procesos o código de búsqueda y datos de Neo4j (<https://neo4j.com/docs/labs/apoc/current/export/cypher/>).

Creación de los nodos a la base de graphos NEO4J , el archivo de los datos es un archivo csv



Prueba 2

17/01/2021



La siguiente declaración crear el grafico y lo almacenara en el catalogo de gráficos:



Prueba 2

17/01/2021

```
CALL gds.graph.create(  
  'Empleados',  
  {  
    Datos: {  
      label: 'Datos',  
      properties: 'nivel_de_satisfacción'  
    }  
  },  
  '*',  
);
```

neo4j\$ CALL gds.graph.create('Empleados', { Datos: { label: 'Datos', properties: 'nivel_de_satisfacción' } }, '*');

	nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	createMillis
1	{ "Datos": { "properties": { "nivel_de_satisfacción": { "property": "nivel_de_satisfacción", "defaultValue": null } }, "label": "Datos" } }	{ "_ALL_": { "orientation": "NATURAL", "aggregation": "DEFAULT", "type": "*", "properties": { } } }	"Empleados"	14999	0	93

Started streaming 1 records after 7 ms and completed after 3586 ms.

En el algoritmo de knn se usa para encontrar la similitud

```
CALL gds.beta.knn.stream('Empleados', {  
  topK: 1,  
  nodeWeightProperty: 'nivel_de_satisfacción',  
  // The following parameters are set to produce a deterministic result  
  randomSeed: 42,  
  concurrency: 1,  
  sampleRate: 1.0,  
  deltaThreshold: 0.0  
})  
YIELD node1, node2, similarity  
RETURN gds.util.asNode(node1).empleado AS Empleado1, gds.util.asNode(node2).empleado AS Empleado2, similarity  
ORDER BY similarity DESCENDING, Empleado1, Empleado2
```

Resultados de similitud obtenidos con el algoritmo KNN



Prueba 2

17/01/2021

```
neo4j$ CALL gds.beta.knn.stream('Empleados', { topK: 1, nodeWeightProperty: 'nivel_de_satisfacción', // The following paramet...
```

	Empleado1	Empleado2	similarity
1	"Empleado10"	"Empleado12306"	1.0
2	"Empleado100"	"Empleado1188"	1.0
3	"Empleado1000"	"Empleado1381"	1.0
4	"Empleado10000"	"Empleado11849"	1.0
5	"Empleado10001"	"Empleado8076"	1.0
6	"Empleado10002"	"Empleado11675"	1.0
7	"Empleado10003"	"Empleado11675"	1.0

Creemos de nuevo el grapho para obtener una relación entre los 3 nodos diferentes

```
CALL gds.graph.create(
  'Empleados',
  {
    Datos: {
      label: 'Datos',
      properties: 'nivel_de_satisfacción'
    }
  },
  '*
');
```

```
1 CALL gds.beta.knn.stream('Empleados', {
2   topK: 1,
3   nodeWeightProperty: 'nivel_de_satisfacción',
4   // The following parameters are set to produce a deterministic result
5   randomSeed: 42,
6   concurrency: 1,
7   sampleRate: 1.0,
8   deltaThreshold: 0.0
9 })
10 YIELD node1, node2, similarity
11 RETURN gds.util.asNode(node1).empleado AS Empleado1, gds.util.asNode(node2).empleado AS Empleado2, similarity
12 ORDER BY similarity DESCENDING, Empleado1, Empleado2
```

Generar otro entorno en donde solo ingrese el 70% de los datos y validar con el 30%.



Sistemas Expertos

Tema: Basados en casos.

Prueba 2

17/01/2021

```
import csv
import pandas as pd
from neomodel import StructuredNode, StringProperty, RelationshipTo, RelationshipFrom, config, IntegerProperty, FloatProperty
config.DATABASE_URL = 'bolt://neo4j:ddd@localhost:7687'

df = pd.read_csv("data2.csv", sep=';')
lista = [list(row) for row in df.values]

class Datos(StructuredNode):
    empleado = StringProperty(index=True)
    nivel_de_satisfacción = FloatProperty(index=True)
    ultima_evaluacion = FloatProperty(index=True)
    proyecto_de_número = FloatProperty(index=True)
    promedio_de_horas_mensuales = FloatProperty(index=True)
    tiempo_dedicado_a_la_compañía = FloatProperty(index=True)
    accidente_laboral = FloatProperty(index=True)
    promoción_últimos_5_años = FloatProperty(index=True)
    izquierda = FloatProperty(index=True)

f=1
s="Empleado"

for x in lista:
    f=f+1
    t=s+str(f)
    datos=Datos(
        empleado = t,
        nivel_de_satisfacción = (x[0]),
        ultima_evaluacion = (x[1]),
        proyecto_de_número = (x[2]),
        promedio_de_horas_mensuales = (x[3]),
        tiempo_dedicado_a_la_compañía = (x[4]),
        accidente_laboral = (x[5]),
        promoción_últimos_5_años = (x[6]),
        izquierda = (x[7]),
    ).save()
```

Match de los 70% de los nodos

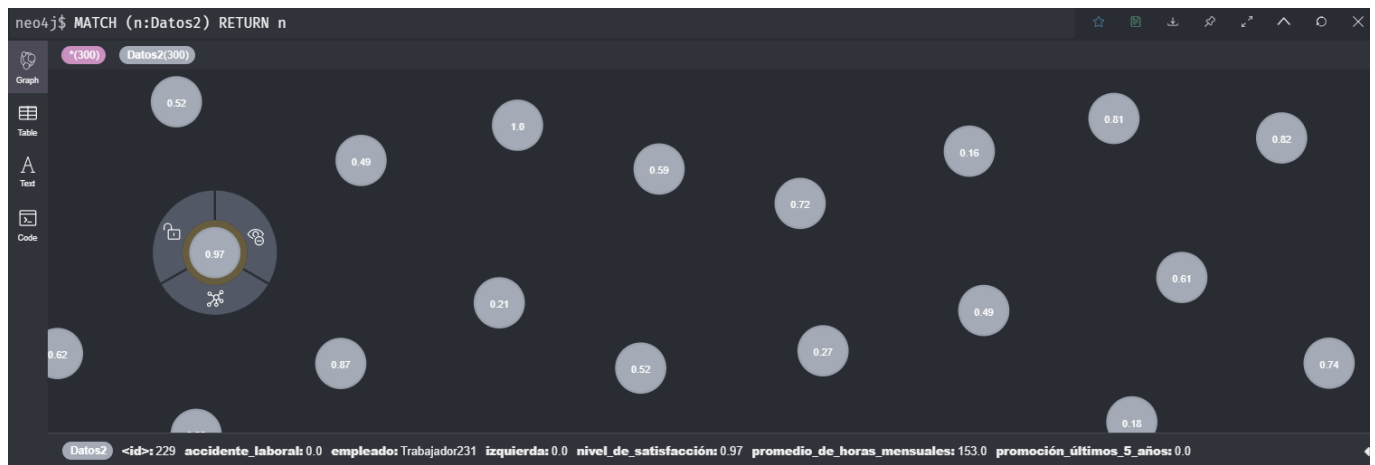


Creación de los 30% de los nodos



Prueba 2

17/01/2021



Algoritmo para la comparación del 30% y el 70%

```
1 CALL gds.graph.create(  
2   'Evaluacion',  
3   {  
4     Datos: {  
5       label: 'Datos',  
6       properties: 'ultima_evaluacion'  
7     }  
8   },  
9   '*'  
10 );
```

neo4j\$ CALL gds.beta.knn.stream('Empleados', { topK: 1, nodeWeightProperty: 'nivel_de_satisfacción', // The following paramet...

	Empleado1	Empleado2	similarity
1	"Empleado10"	"Empleado12306"	1.0
2	"Empleado100"	"Empleado1188"	1.0
3	"Empleado1000"	"Empleado1381"	1.0
4	"Empleado10000"	"Empleado11849"	1.0
5	"Empleado10001"	"Empleado8076"	1.0
6	"Empleado10002"	"Empleado11675"	1.0
7	"Empleado10003"	"Empleado11675"	1.0

Bibliografía

- [1] <https://tech-cookbook.com/2019/11/11/python-machine-learning-knn-example-from-csv-data/>
- [2] <https://smalldatabrains.com/python-knn/>
- [3] <https://www.kaggle.com/cengizeralp/practice-1-gender-prediction-with-knn/data>



Sistemas Expertos
Tema: Basados en casos.

Prueba 2

17/01/2021