

INFORME

ANÁLISIS DE LAS VOTACIONES

KNN IN NEO4J

Nombre: Jordan Murillo

Fecha: Viernes, 22 de mayo de 2020

Tema: kNN Classification of members of congress using similarity algorithms in Neo4j.

Requerimientos:

- Neo4j
- Neo4j graph algorithms plugin
- Neo4j APOC plugin

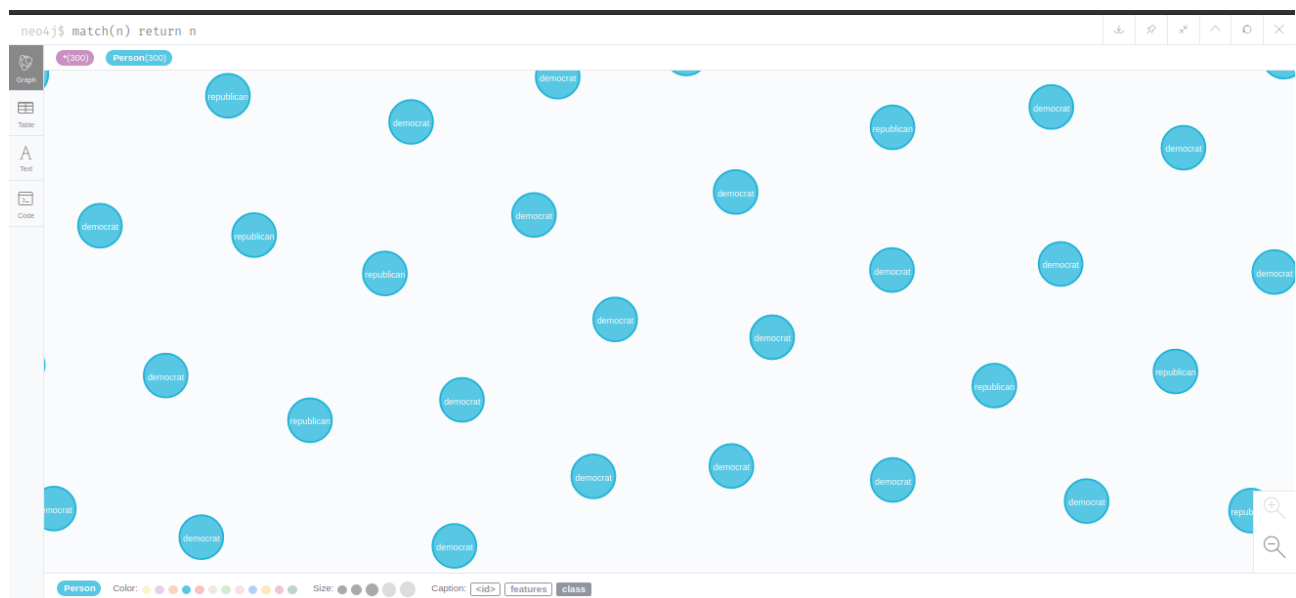
Base de Datos – Dataset

Tomaremos el archivo csv de un link de internet:

Código Neo4j

```
LOAD CSV FROM "http://archive.ics.uci.edu/ml/machine-learning-databases/voting-records/house-votes-84.data" as row
CREATE (p:Person)
SET p.class = row[0],
    p.features = row[1..];
```

Resultado



Votos Perdidos

Ver votos perdidos

MATCH (n:Person)

```
WHERE "?" in n.features  
RETURN count(n)
```

Resultado



neo4j\$ MATCH (n:Person) WHERE "?" in n.features RETURN count(n)

count(n)
203

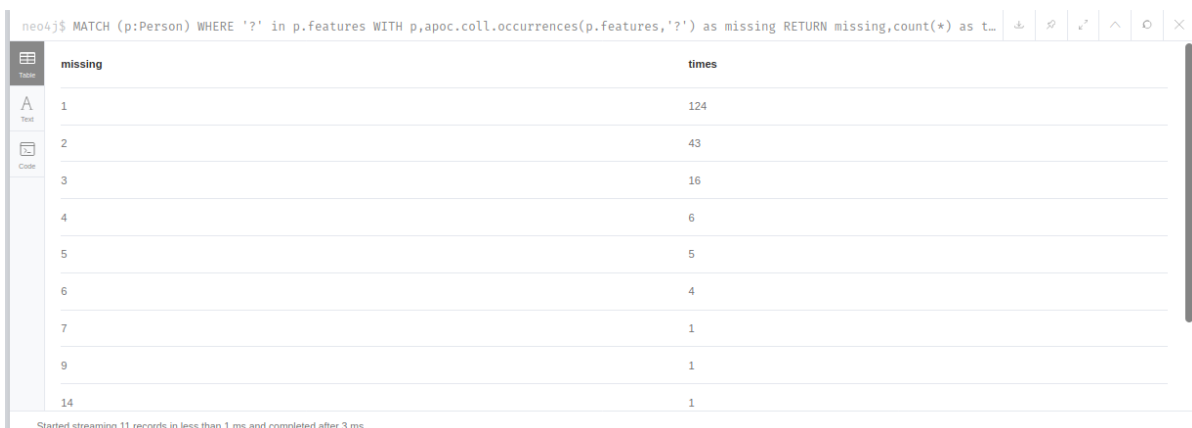
Started streaming 1 records in less than 1 ms and completed after 5 ms.

Visualizar distribución de votos perdidos por miembro

MATCH (p:Person)

```
WHERE '?' in p.features  
WITH p,apoc.coll.occurrences(p.features,'?') as missing  
RETURN missing,count(*) as times ORDER BY missing ASC
```

Resultado



neo4j\$ MATCH (p:Person) WHERE '?' in p.features WITH p,apoc.coll.occurrences(p.features,'?') as missing RETURN missing,count(*) as times

missing	times
1	124
2	43
3	16
4	6
5	5
6	4
7	1
9	1
14	1

Started streaming 11 records in less than 1 ms and completed after 3 ms.

Eliminar votos perdidos mayores a 6 por cada miembro

MATCH (p:Person)

```
WITH p,apoc.coll.occurrences(p.features,'?') as missing  
WHERE missing > 6  
DELETE p
```

Resultado

```
neo4j$ MATCH (p:Person) WITH p,apoc.coll.occurrences(p.features,'?') as missing WHERE missing > 6 DELETE p
```

Deleted 5 nodes, completed after 2 ms.

Entrenamiento y datos para pruebas

Seleccionar datos para entrenamiento que son 80% de los datos

80%=344

MATCH (p:Person)

```
WITH p LIMIT 344  
SET p:Training;
```

Resultados

```
neo4j$ MATCH (p:Person) WITH p LIMIT 344 SET p:Training;
```

Added 344 labels, completed after 3 ms.

Seleccionar datos para pruebas que son el 20%

MATCH (p:Person)

```
WITH p SKIP 344  
SET p:Test;
```

Resultados

neo4j\$ MATCH (p:Person) WITH p SKIP 344 SET p:test;		🔍	🔍	🔍	🔍	🔍
Table	Added 86 labels, completed after 1 ms.					
Code						

Added 86 labels, completed after 1 ms.

Creación de vectores para el análisis de la similitud mediante la distancia euclidiana.

Tomar en cuenta que y=1, n=0 y ?=0.5

```
MATCH (n:Person)

UNWIND n.features as feature
WITH n,collect(CASE feature WHEN 'y' THEN 1
                           WHEN 'n' THEN 0
                           ELSE 0.5 END) as feature_vector
SET n.feature_vector = feature_vector
```

Resultado

neo4j\$ MATCH (n:Person) UNWIND n.features as feature WITH n,collect(CASE feature WHEN 'y' THEN 1 WHEN 'n' THEN 0 ELSE 0.5 END) as featur...		🔍	🔍	🔍	🔍	🔍
Table	Set 430 properties, completed after 8 ms.					
Code						

Set 430 properties, completed after 8 ms.

kNN classifier algorithm

```
MATCH (test:Test)

WITH test,test.feature_vector as feature_vector

CALL apoc.cypher.run('MATCH (training:Training)

    WITH training,gds.alpha.similarity.euclideanDistance($feature_vector,
training.feature_vector) AS similarity
```

```

ORDER BY similarity ASC LIMIT 3

RETURN collect(training.class) as classes',

{feature_vector:feature_vector}) YIELD value

WITH test.class as class,
apoc.coll.sortMaps(apoc.coll.frequencies(value.classes), '^count')[-1].item as
predicted_class

WITH sum(CASE when class = predicted_class THEN 1 ELSE 0 END) as
correct_predictions, count(*) as total_predictions

RETURN correct_predictions,total_predictions, correct_predictions /
toFloat(total_predictions) as ratio

```

Resultado

neo4j\$ MATCH (test:Test) WITH test,test.feature_vector as feature_vector CALL apoc.cypher.run('MATCH (training:Training) // calculat...			
	correct_predictions	total_predictions	ratio
78	86	0.9069767441860465	

Started streaming 1 records after 14 ms and completed after 258 ms.

Conclusión

En la práctica aprendimos como usar Neo4j para aplicar algoritmos que nos ayudan a interpretar datos y dar resultados como lo haría un sistema experto de casos (CBR). Con esto aprendemos también a utilizar herramientas nuevas para hacer de manera más rápido el análisis de los datos.

Webgrafia

- <https://tbgraph.wordpress.com/2018/11/25/knn-classification-using-similarity-algorithms-in-neo4j/>