Sistemas Expertos - KNN

Nombre: David Egas

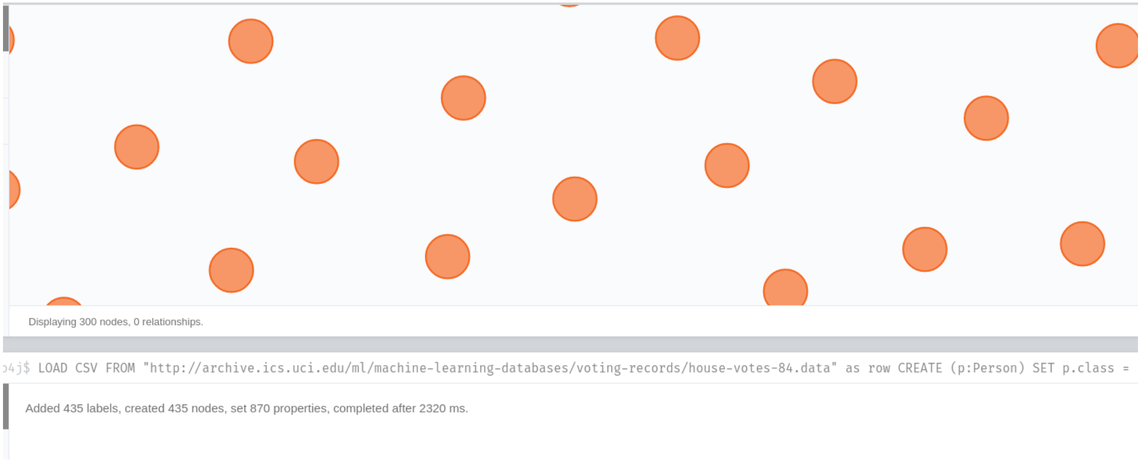
Implementar comando a comando el ejemplo de los votos para la cámara de representantes de EEUU.

1. Carga del archivo .CSV con los votos

Comando:

```
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:



Displaying 300 nodes, 0 relationships.

```
›4j$ 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 =
```

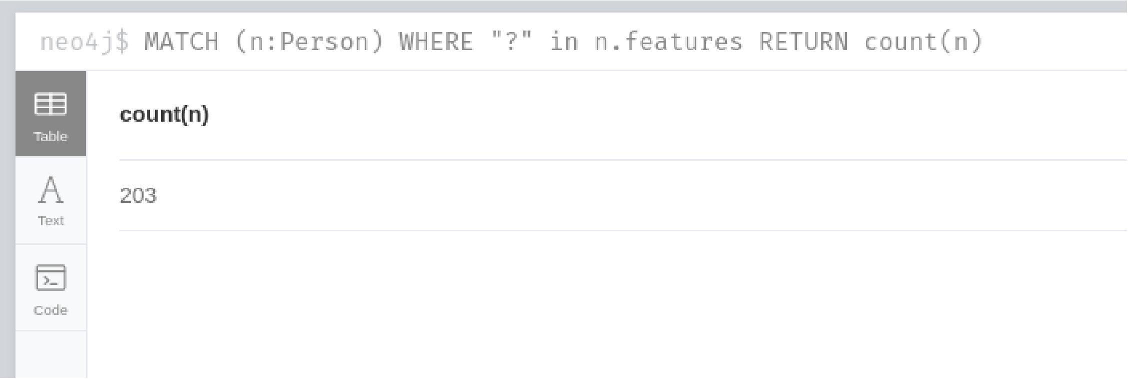
Added 435 labels, created 435 nodes, set 870 properties, completed after 2320 ms.

2. Votos faltantes

Comando:

```
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

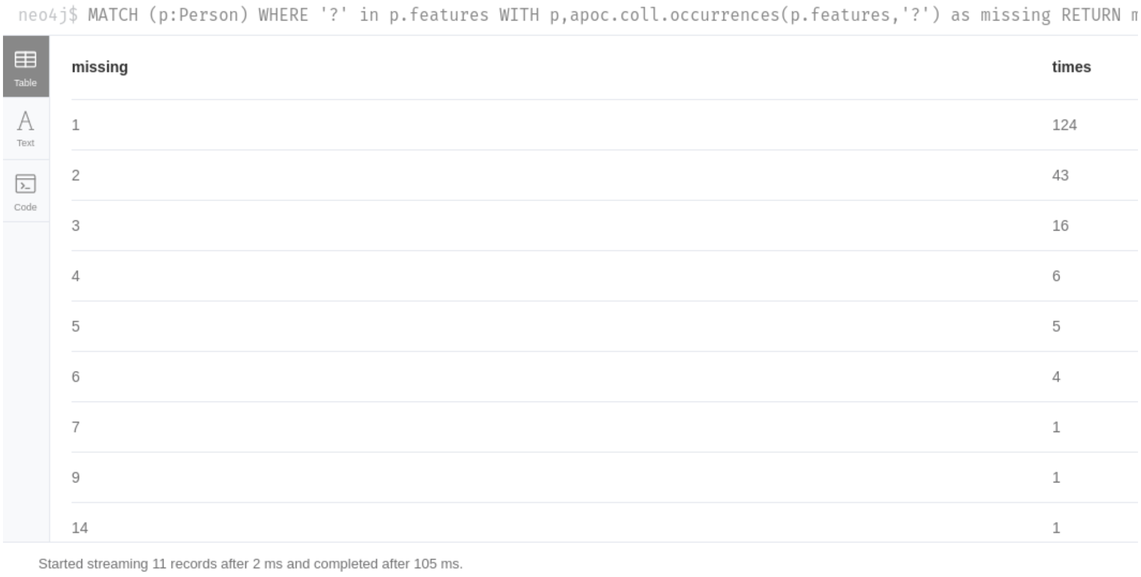
204 son los votos faltantes.

3. Votos faltantes por miembro

Comando:

```
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
```

Resultados:



```
neo4j$ MATCH (p:Person) WHERE '?' in p.features WITH p,apoc.coll.occurrences(p.features,'?') as missing RETURN m
```

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

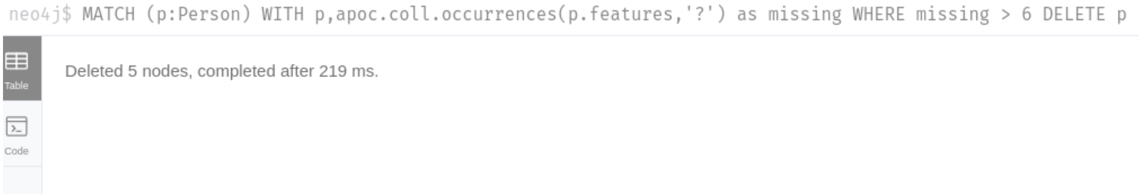
Started streaming 11 records after 2 ms and completed after 105 ms.

4. Exclusión de los miembros que casi nunca votan.

Comando:

```
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
```

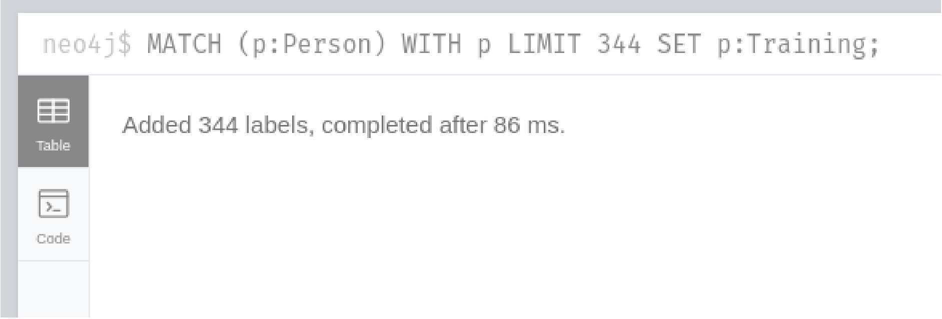
Deleted 5 nodes, completed after 219 ms.

5. Marcar los datos de entrenamiento

Comando:

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

Resultado:



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

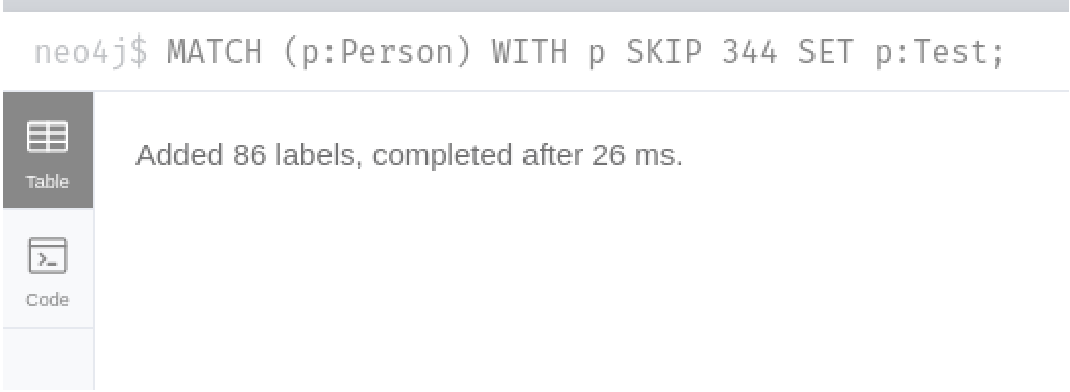
Added 344 labels, completed after 86 ms.

6. Marcar datos de prueba

Comando:

MATCH (p:Person)
WITH p SKIP 344
SET p:Test;

Resultado:



```
neo4j$ MATCH (p:Person) WITH p SKIP 344 SET p:Test;
```

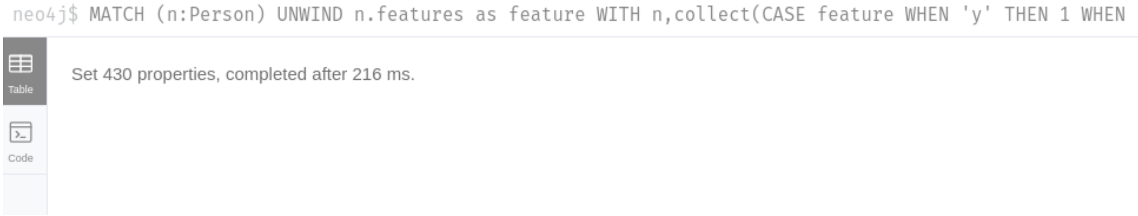
Added 86 labels, completed after 26 ms.

7. Transformar a vector de características

Comando:

```
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
```

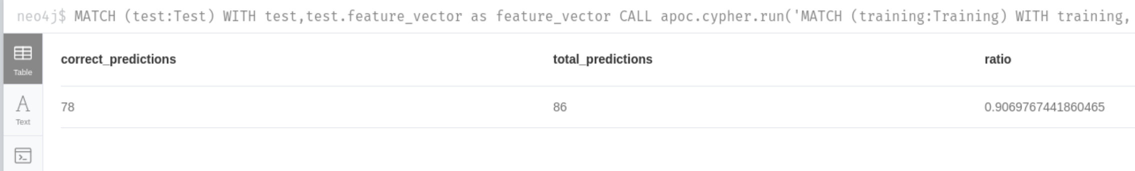
Set 430 properties, completed after 216 ms.

8. Aplicar el algoritmo clasificador KNN

Comando:

```
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) WITH training,
```

correct_predictions	total_predictions	ratio
78	86	0.9069767441860465