Algoritmo KNN

1. DESCARGAR ARCHIVO

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..];

2. CONTAR MIEMBROS DEL CONGRESO

MATCH (n:Person)

WHERE "?" in n.features

RETURN count(n)

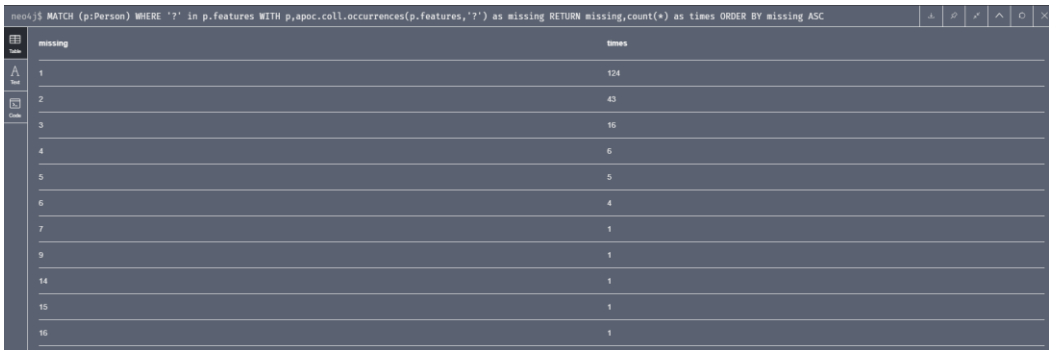
3. DISTRIBUCCION DE LOS BOTOS POR MIEMBRO

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



missing	times
1	124
2	43
3	16
4	6
5	5
6	4
7	1
9	1
14	1
15	1
16	1

4. ELIMINAR VOTOS FALTANTES

MATCH (p:Person)

WITH p,apoc.coll.occurrences(p.features,'?') as missing

WHERE missing > 6

DELETE p

5. MARCAR NODOS DE ENTRENAMIENTO

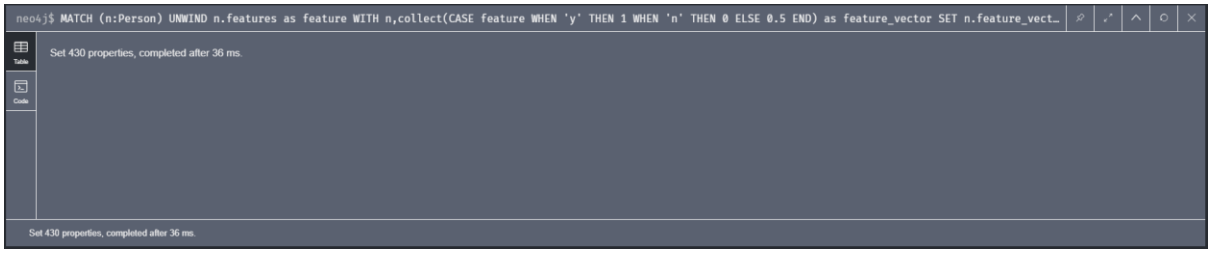
MATCH (p:Person) WITH p LIMIT 344 SET p:Training;

6. MARCAR NODOS PRUEBA

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

7. TRANSFORMAR A VECTOR

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



8. APLICAR KNN

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