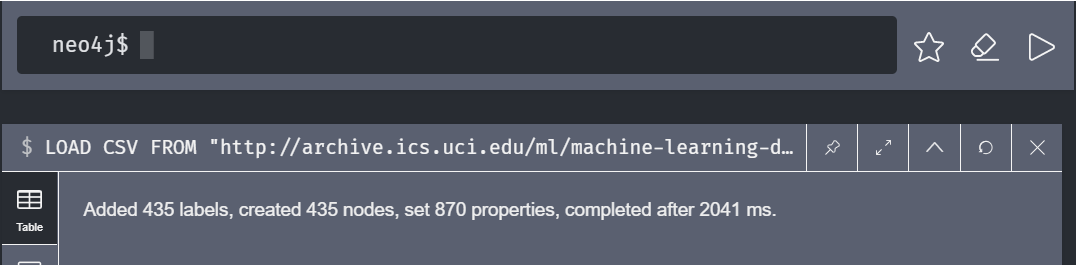
Creación de los nodos que corresponderían a los congresistas y como propiedades los votos.

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..];

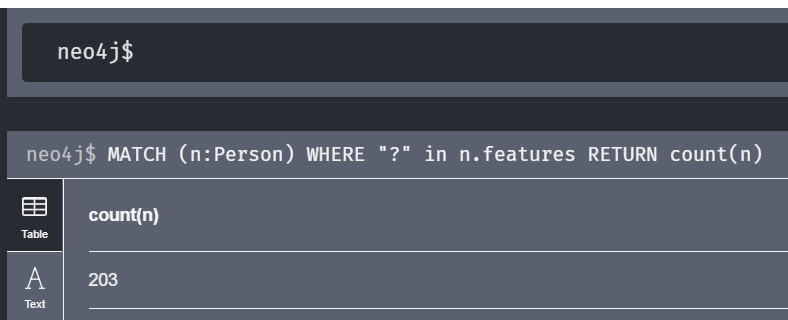


Con la siguiente consulta match se obtiene el número de congresistas que al menos no ha realizado un voto.

MATCH (n:Person)

WHERE "?" in n.features

RETURN count(n)



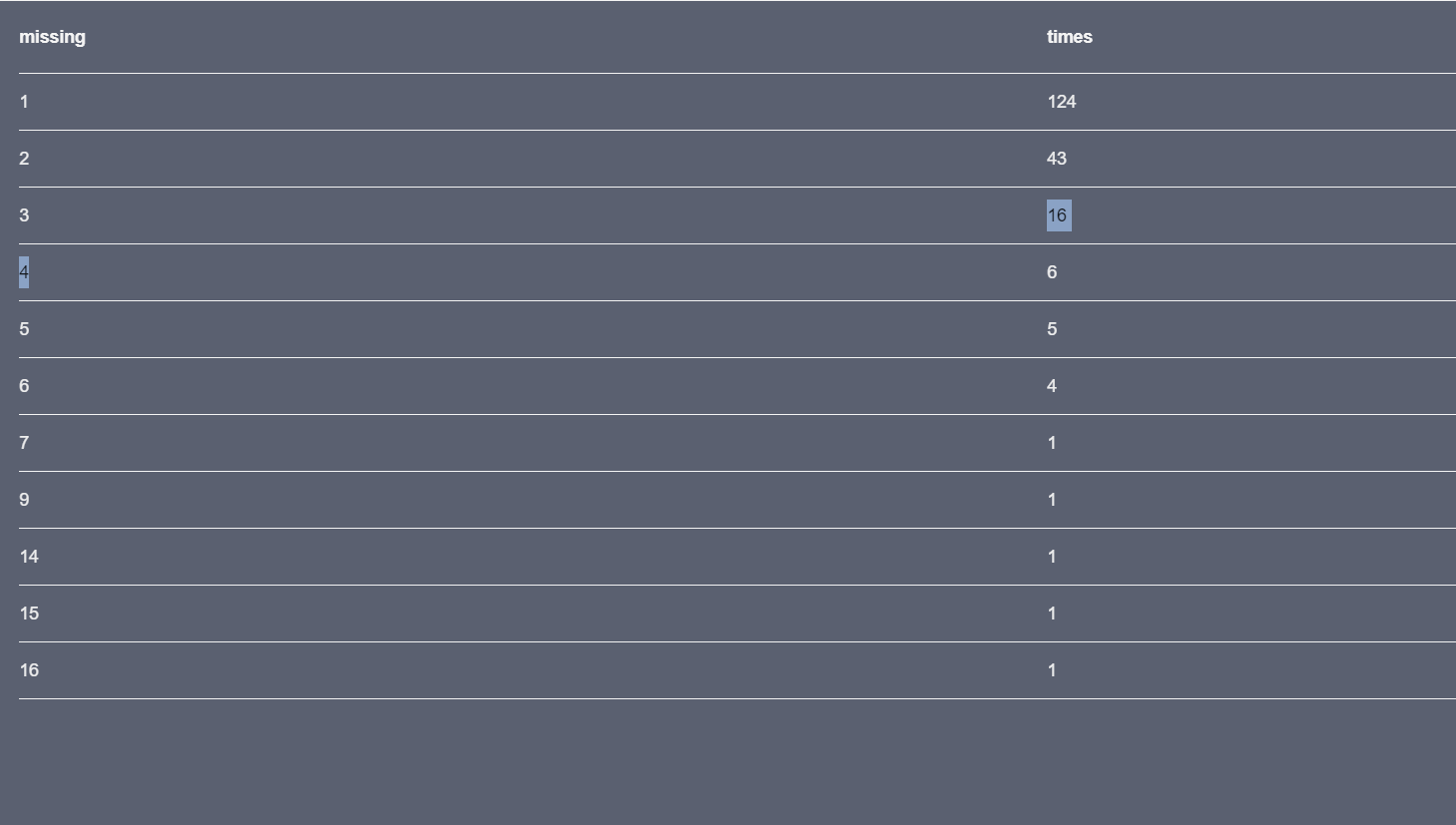
Ahora se consulta para determinar cuantos son los votos faltantes por cada congresista.

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



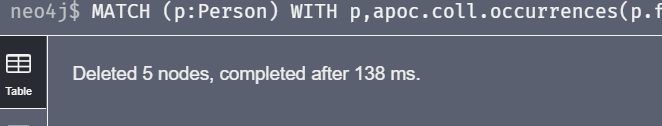
Ahora se excluirá a ciertos congresistas los que mas votos faltantes tienen.

MATCH (p:Person)

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

WHERE missing > 6

DELETE p

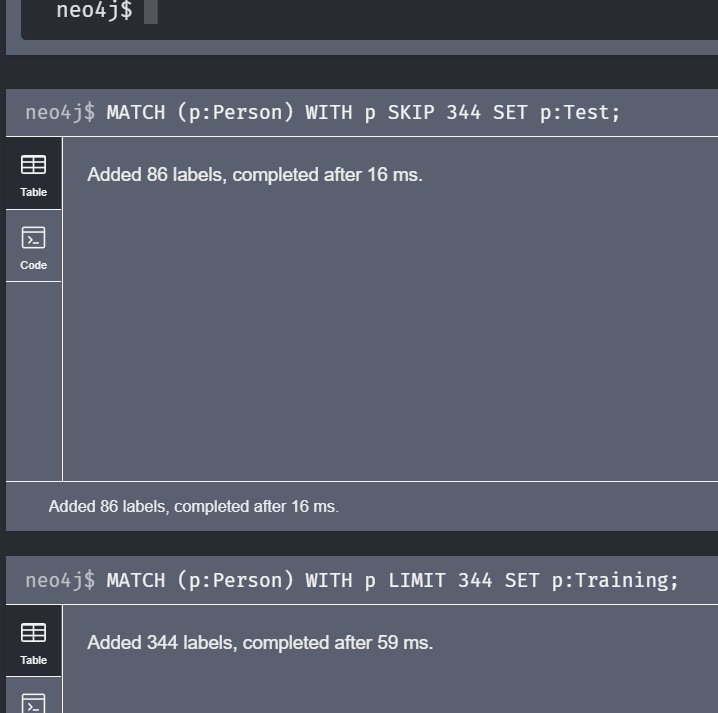


Entrenamiento de los datos

MATCH (p:Person)

WITH p LIMIT 344

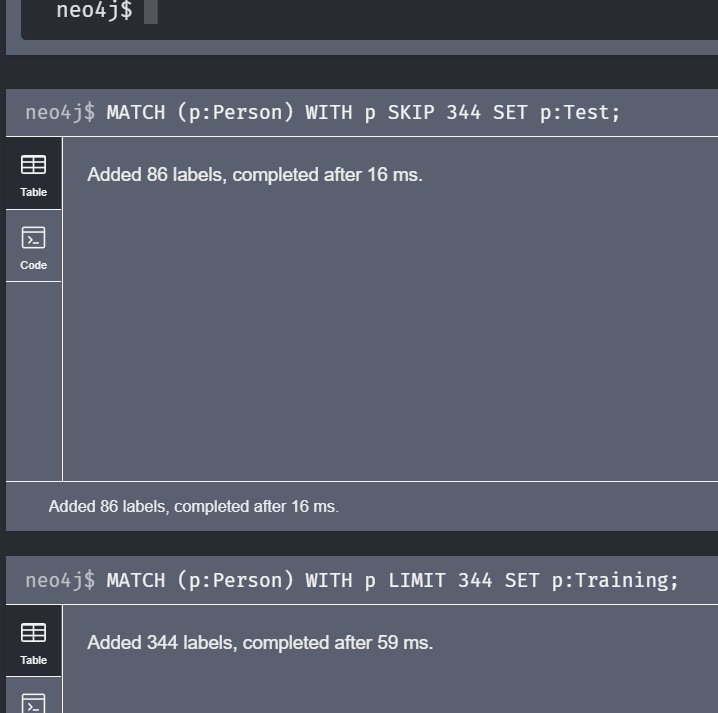
SET p:Training;



MATCH (p:Person)

WITH p SKIP 344

SET p:Test;



Como hay tres grupos de datos debido a los 3 posibles votos (favor, en contra, sin voto)

Mapearemos estos datos.

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



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

