NOMBRE:

ALEX BENAVIDEZ

CARRERA:

INGENIERIA EN SISTEMAS

MATERIA:

SISTEMAS EXPERTOS

PROFESOR:

ING DIEGO QUISIS

FECHA:

22/05/2020

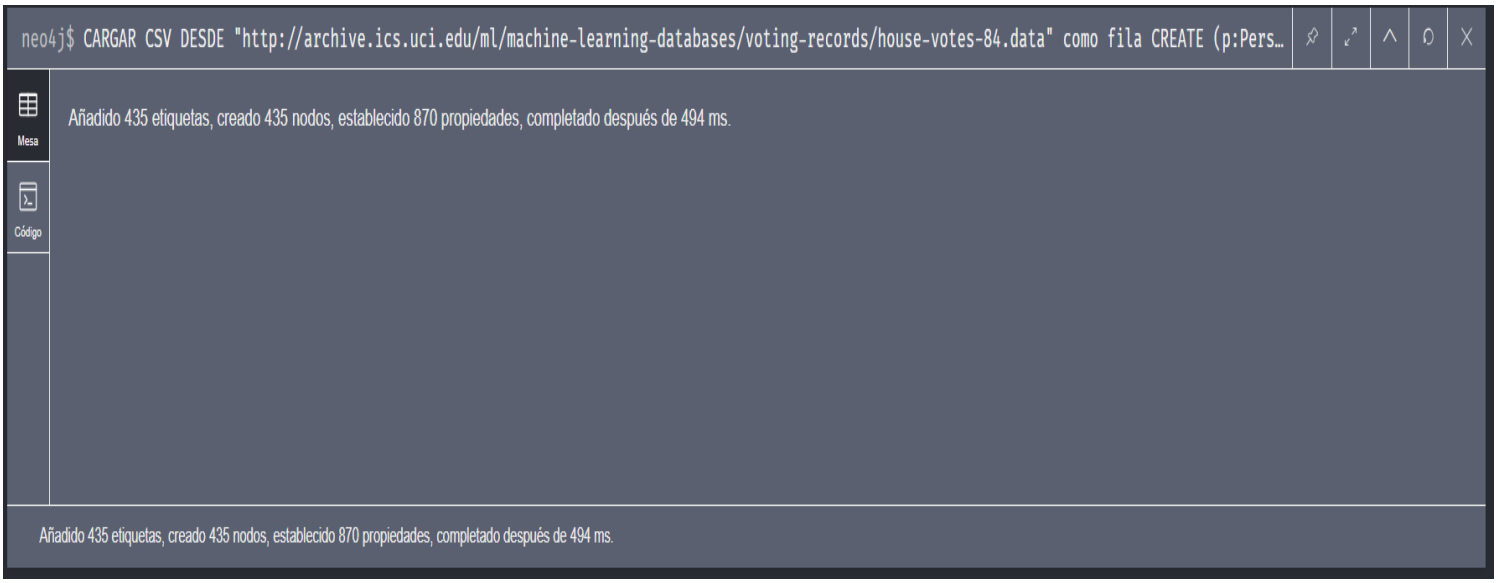
1. Creacion de los nodos

```
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. Vamos a ver cuántos miembros del congreso tienen al menos un voto perdido.

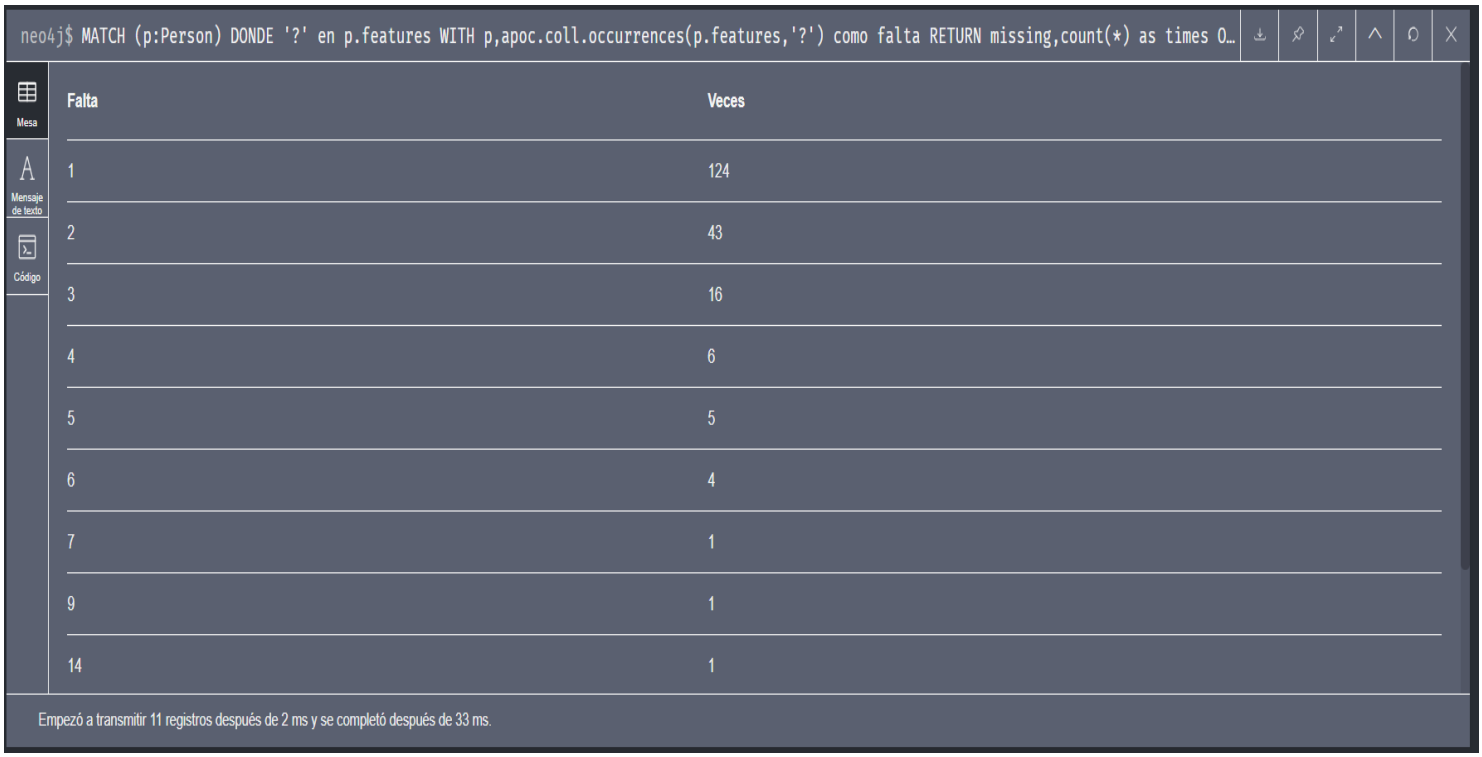
```
MATCH (n:Person)

WHERE "?" in n.features

RETURN count(n)
```

3. **Vistazo de la distribución de los 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
```



Falta	Veces
1	124
2	43
3	16
4	6
5	5
6	4
7	1
9	1
14	1

Empezó a transmitir 11 registros después de 2 ms y se completó después de 33 ms.

4. **Tres miembros casi nunca votaron (14,15,16 votos perdidos) y dos de ellos (7,8 votos faltantes) tienen más del 50% de votos que faltan. Los excluiremos de nuestro análisis posterior para tratar de reducir el ruido.**

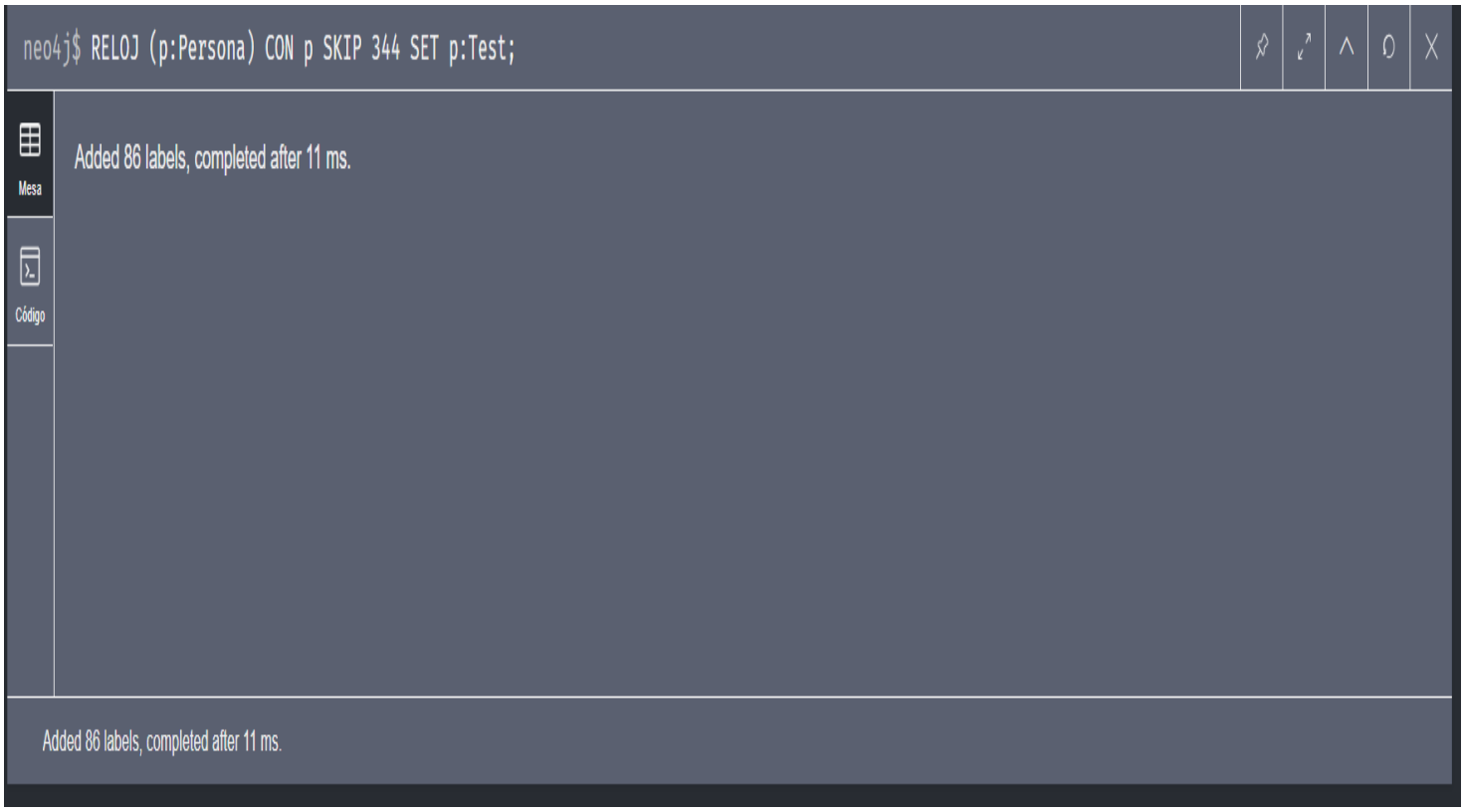
```
MATCH (p:Person)
WITH p,apoc.coll.occurrences(p.features,'?') as missing
WHERE missing > 6
DELETE p
```

5. **Marcar datos de entrenamiento**

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

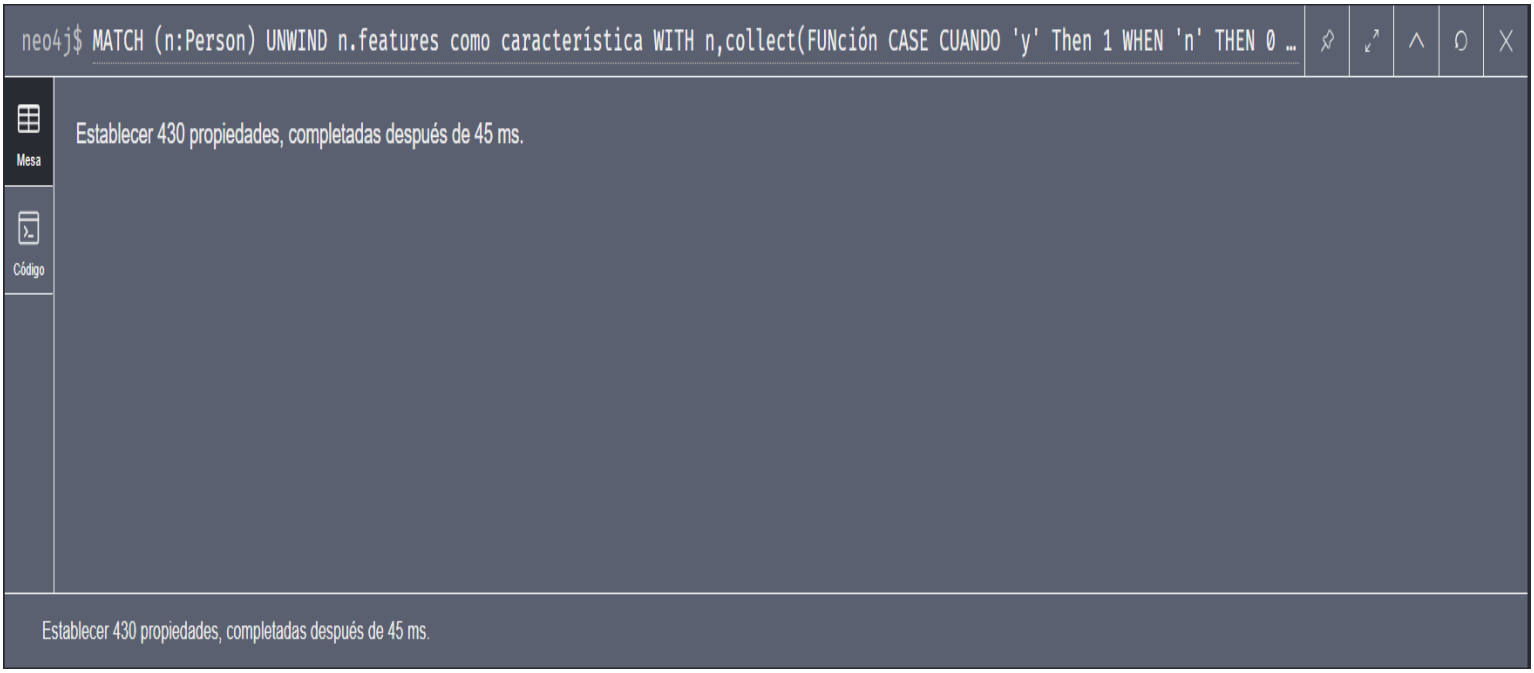
6. **Marcar datos de prueba**

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



7. **Transformar a vector de entidades.**

```
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. **Calcular la Distancia Euclideana entre nodos**

```
MATCH (test:Test)
WITH test,test.feature_vector as feature_vector
CALL apoc.cypher.run('MATCH (training:Training)
     WITH
training,gds.alpha.similarity.euclideanDistance($featu
re_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;
```


neo4j$ MATCH (test:Test) WITH test,test.feature_vector as feature_vector CALL apoc.cypher.run('MATCH (training:Training…
Mesa
Mensaje de texto
Código
correct_predictions
total_predictions
Cociente
78
86
0.906976744186046
Empezó a transmitir 1 registros después de 74 ms y se completó después de 631 ms.