```
MATCH     (b:Person)-[r:RATED]->(m:Movie), (b)-[s:SIMILARITY]-(a:Person {name:'Zoltan Varju'})
WHERE     NOT((a)-[:RATED]->(m))
WITH      m, s.similarity AS similarity, r.rating AS rating
ORDER BY m.name, similarity DESC
WITH      m.name AS movie, COLLECT(rating)[0..3] AS ratings
WITH      movie, REDUCE(s = 0, i IN ratings | s + i)*1.0 / LENGTH(ratings) AS reco
ORDER BY reco DESC
RETURN    movie AS Movie, reco AS Recommendation
```

neo4j$ LOAD CSV FROM " http://archive.ics.uci.edu/ml/machine-lear...

Table

Added 435 labels, created 435 nodes, set 870 properties, completed after 1830 ms.

Code

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

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

Table

count(n)

A
Text

203

Code

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

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

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

```
neo4j$ MATCH (p:Person) WITH p,apoc.coll.occurrences(p.features,'...
```
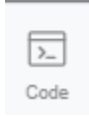
Deleted 5 nodes, completed after 71 ms.

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

```
neo4j$ MATCH (p:Person) WITH p LIMIT 344 SET p:Training;
```
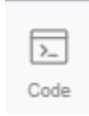
Added 344 labels, completed after 98 ms.

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

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

Added 86 labels, completed after 21 ms.

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

```
neo4j$ MATCH (n:Person) UNWIND n.features as feature WITH n,colle...
```

Set 430 properties, completed after 159 ms.

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;

neo4j$ MATCH (test:Test) WITH test,test.feature_vector as feature_vector CALL a...

correct_predictions	total_predictions	ratio
77	86	0.8953488372093024

Started streaming 1 records after 1 ms and completed after 577 ms.