Data Science Task

Task: Providing Authors with Co-author suggestions

SUBMISSION BY :

ANJALI KEDIA

anjali.kedia2021@vitstudent.ac.in

+91 9560427614

Part 1: Unveiling Patterns (Exploring the Landscape of Author Collaborations)

Tool : NEO4J DESKTOP 1.5.8

Exploratory Data Analysis

Table of Contents:

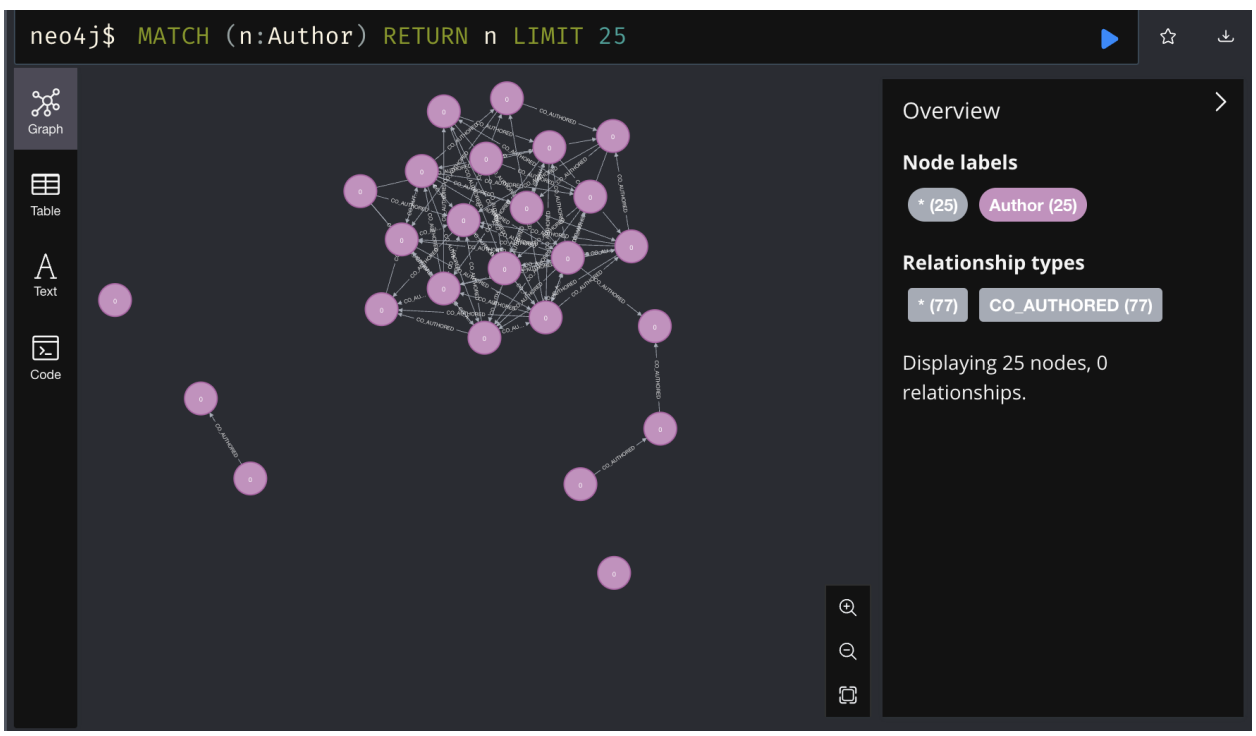
I begin with visualizing the information we currently have, and analyzing the data we are working with:

All the EDA approaches have been provided step by step below until the point where we download the csv which I have used for the second task.

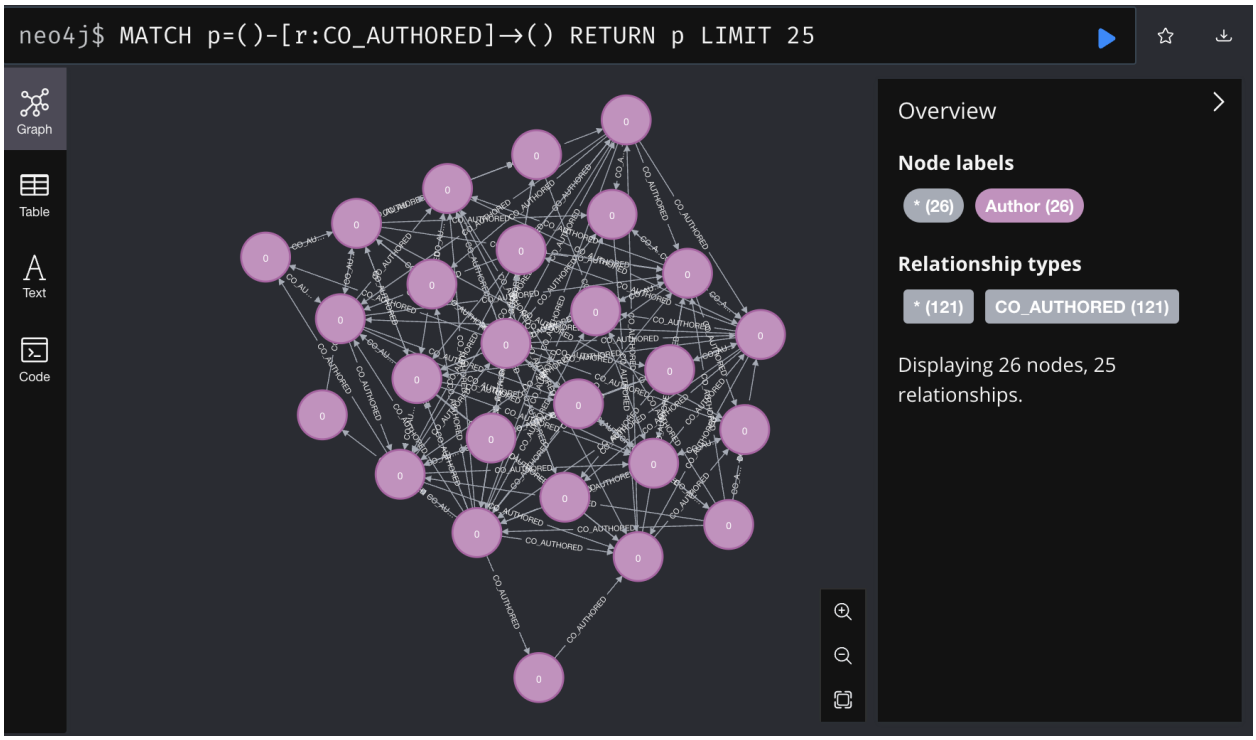
1. VISUALISATION:

   Visualizations can provide immediate insights into the data's structure, revealing clusters of nodes, central nodes, and other important network properties. This aids in understanding the data's overall topology and identifying potential anomalies or patterns.

   MATCH (n:Author) RETURN n LIMIT 25



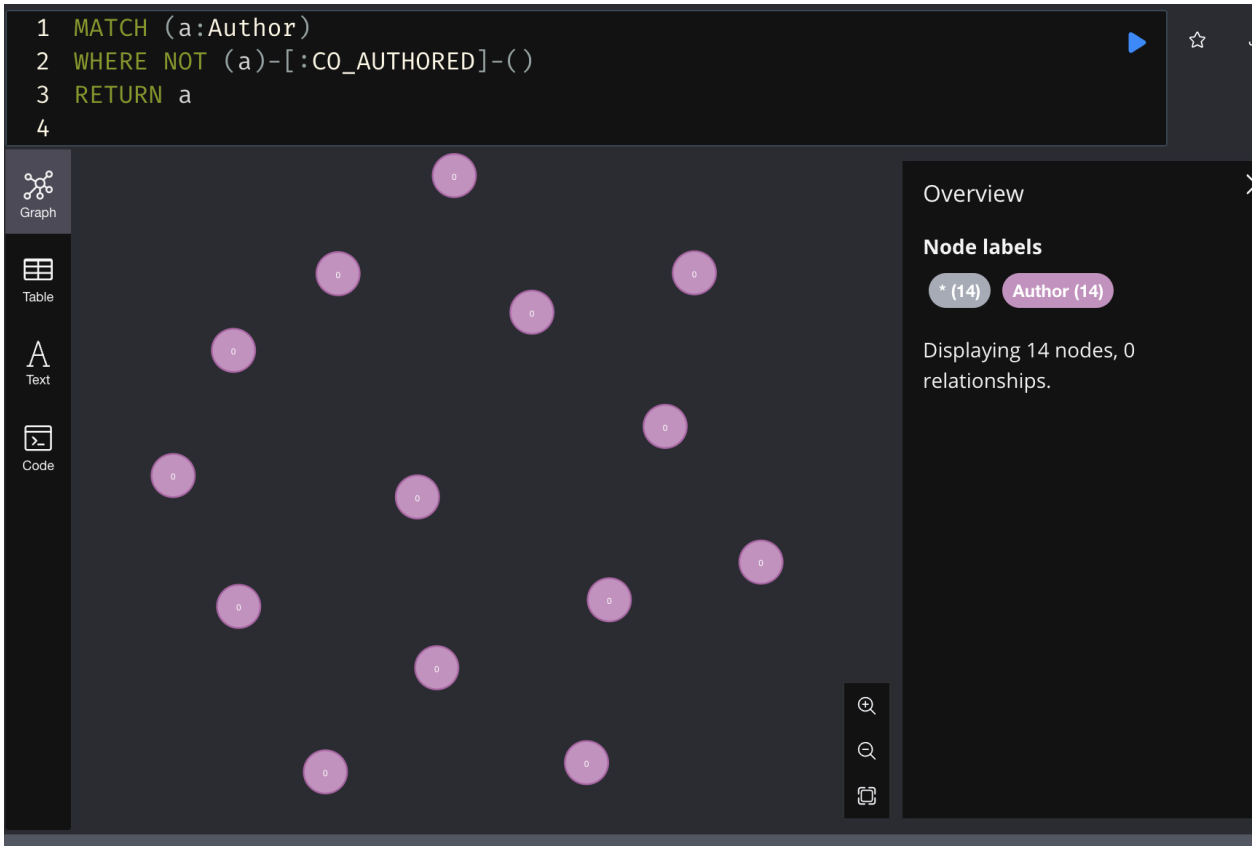
```
MATCH p=()-[r:CO_AUTHORED]->() RETURN p LIMIT 25
```



After looking at the existing relationships and authors, I was curious if there are authors who have had no co-authors yet.

<u>Cypher query to check if there are authors that never had co authors</u>

```
->  MATCH (a:Author) WHERE NOT (a)-[:CO_AUTHORED]-() RETURN a
```



Now that we know the basics of the data we are working with, we move on with

2. **UNDERSTANDING THE DATA:**

   A deep understanding of the data is essential for formulating meaningful research questions and hypotheses. It helps analysts identify potential biases, outliers, or data quality issues that might affect the analysis and interpretation of results.

1. **<u>TOTAL NUMBER OF AUTHORS</u>**

```
neo4j$ MATCH (a:Author) RETURN COUNT(a) AS totalAuthors
```

	totalAuthors
1	347

2. **<u>Total Number of Co-Authorship Relationships:</u>**

```
neo4j$ MATCH ()-[:CO_AUTHORED]-() RETURN COUNT(*) AS totalCoAuthorships
```

	totalCoAuthorships
1	5038

Started streaming 1 records in less than 1 ms and completed after 5 ms.

I will later use this which I create tensors to check the uniformity of sizes of tensors created.

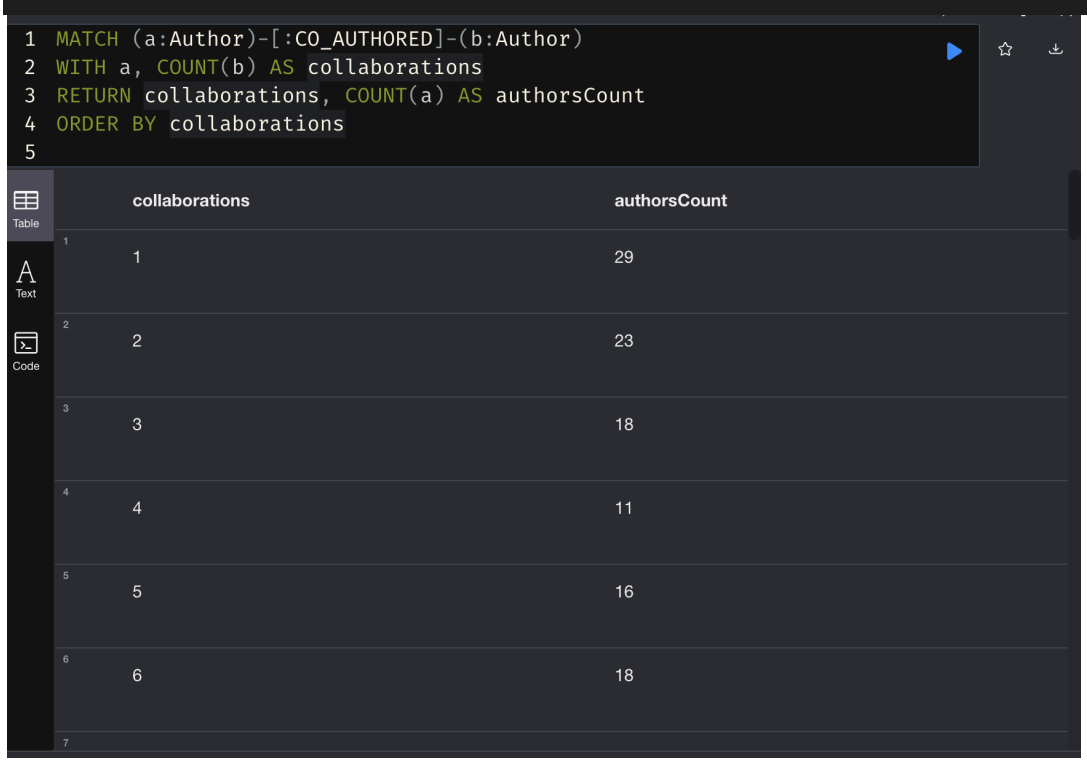
3. **Exploring Collaboration Patterns**

Understanding collaboration patterns can be crucial in various domains, such as academia, business, and social networks. It can help uncover opportunities for collaboration, evaluate the effectiveness of collaboration strategies, and detect changes in collaborative behavior over time.

Now, let's explore the collaboration patterns among authors. We can start by looking at the distribution of co-authorship relationships:

Distribution of Co-Authorships (Number of Collaborations per Author):

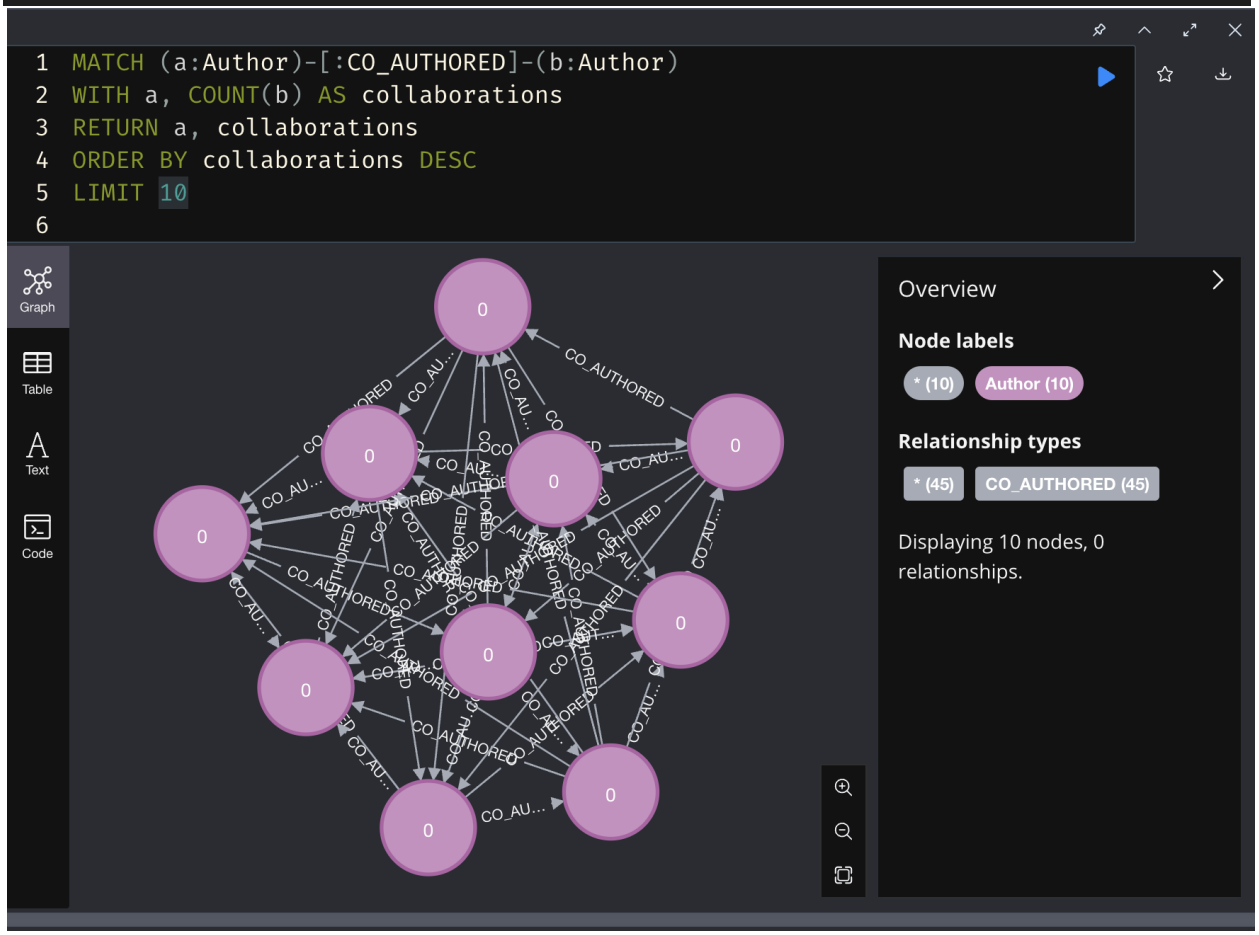
```
MATCH (a:Author)-[:CO_AUTHORED]-(b:Author)
WITH a, COUNT(b) AS collaborations
RETURN collaborations, COUNT(a) AS authorsCount
ORDER BY collaborations
```



	collaborations	authorsCount
1	1	29
2	2	23
3	3	18
4	4	11
5	5	16
6	6	18
7		

Authors with the Most Collaborations (Top N authors with the most collaborations):

```
MATCH (a:Author)-[:CO_AUTHORED]-(b:Author)
WITH a, COUNT(b) AS collaborations
RETURN a, collaborations
ORDER BY collaborations DESC
LIMIT 10
```



I list down all these authors, so that later when I evaluate working of my model, I can cross check if the predicted co-authors, do not always come from this list:

AuthorIDs:

authorID_5f9c4_ab08c_ac745_7e911_1a30e

authorID_f10d9_1a759_6bf5a_67735_79ff1

authorID_3635a_91e3d_a857f_7847f_68185

authorID_3635a_91e3d_a857f_7847f_68185

authorID_d6e5a_20b30_f8721_6b2c7_58f5e

authorID_6f4b6_61212_5fb3a_0daec_d2799

authorID_49d18_0ecf5_61328_19571_bf39d

authorID_7688b_6ef52_55596_2d008_fff89

authorID_1be00_34108_2e25c_4e251_ca671

authorID_27d71_9c754_aacd4_92a6d_c8a1b

4. Network Analysis

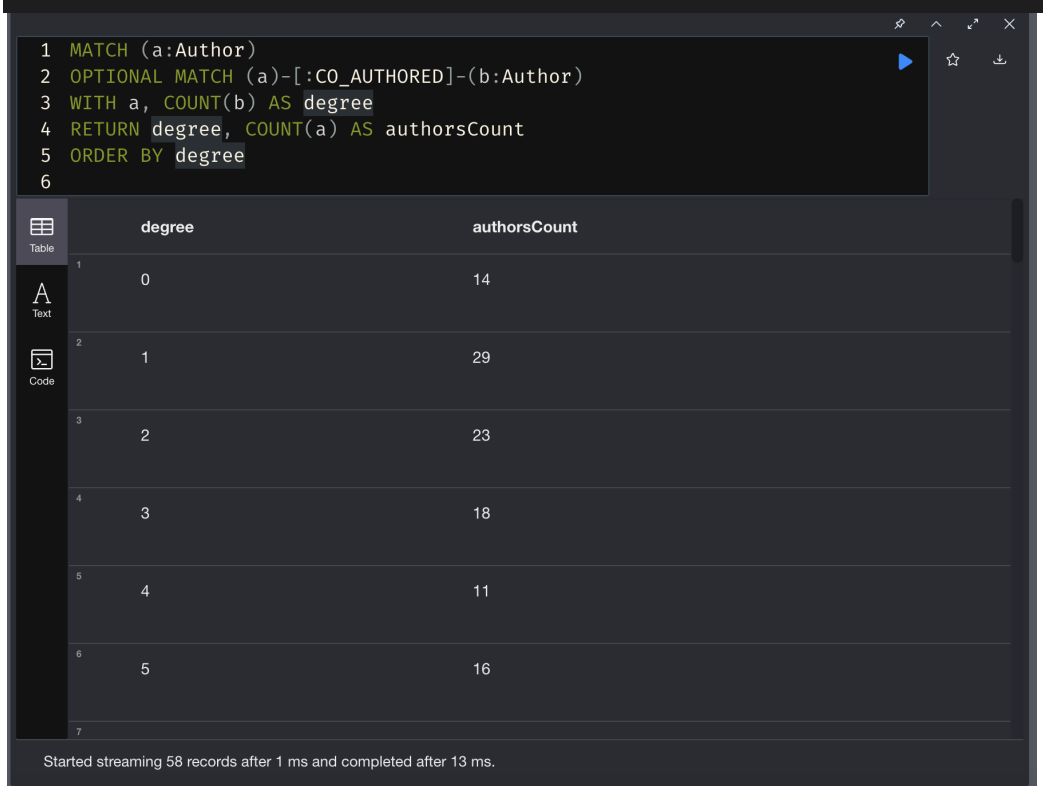
Network analysis provides a quantitative basis for understanding the

roles and significance of nodes and edges within the graph. It uncovers hidden patterns, identifies key players, and supports hypothesis testing by quantifying network properties.

Let's perform some basic network analysis to understand the structure of the collaboration network:

Degree Distribution (Number of Collaborators per Author):

```
MATCH (a:Author)
OPTIONAL MATCH (a)-[:CO_AUTHORED]-(b:Author)
WITH a, COUNT(b) AS degree
RETURN degree, COUNT(a) AS authorsCount
ORDER BY degree
```



List of pairs of author IDs where each pair represents two authors who have co-authored at least one publication together. This kind of query is useful for identifying collaboration networks among authors in a graph database.

```
MATCH (a:Author)-[:CO_AUTHORED]-(b:Author)
RETURN ID(a) AS source, ID(b) AS target
```

Table | Text | Code

	source	target
126	105	3
127	90	3
128	65	3
129	171	3
130	60	3
131	111	3
132		

Started streaming 5038 records in less than 1 ms and completed in less than 1 ms, displaying first 1000 rows.

Checking if we can find and extract node features

```
MATCH (a:Author)-[:CO_AUTHORED]-(b:Author)
RETURN ID(a) AS source, ID(b) AS target, a.features AS sourceFeatures,
b.features AS targetFeatures
```

```
MATCH (a:Author)-[:CO_AUTHORED]-(b:Author)
RETURN ID(a) AS source, ID(b) AS target, a.features AS sourceFeatures,
b.features AS targetFeatures
```

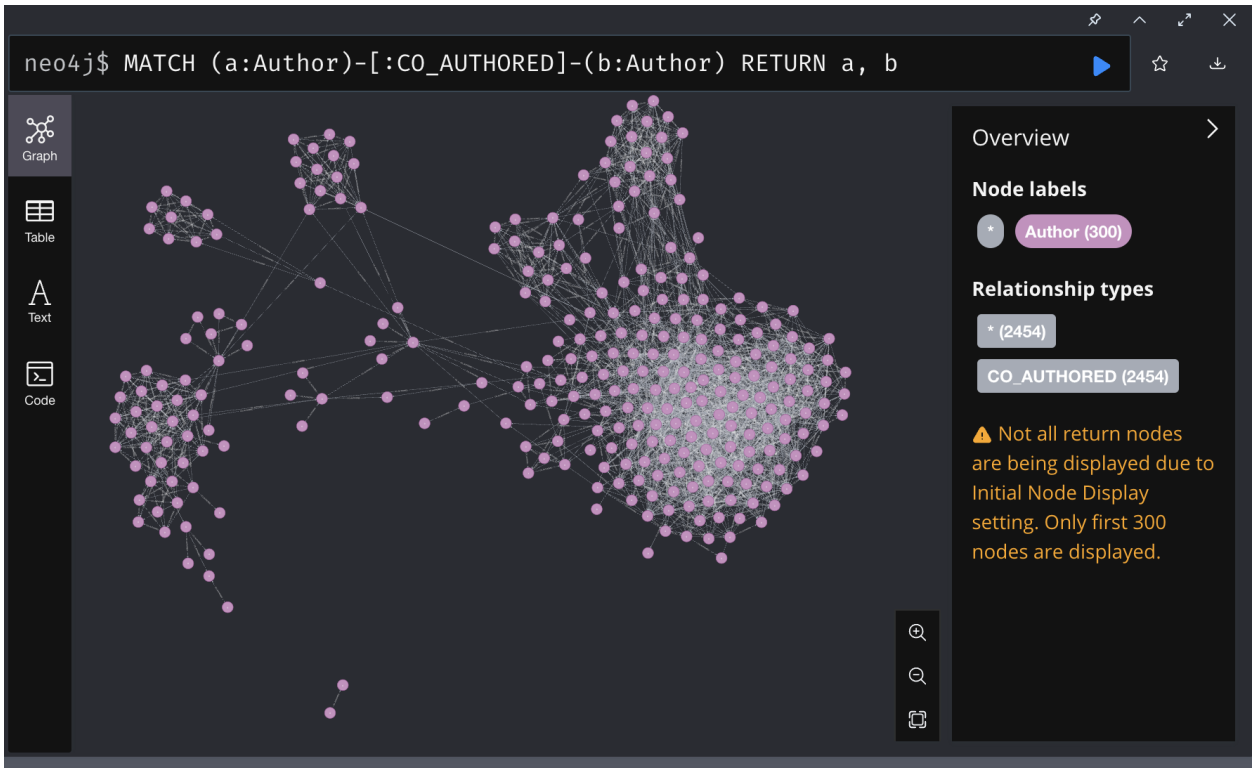
	source	target	sourceFeatures	targetFeatures
21	159	0	*null*	*null*
22	224	0	*null*	*null*
23	63	0	*null*	*null*
24	108	0	*null*	*null*
25	137	0	*null*	*null*
26	7	0	*null*	*null*
27	60	0	*null*	*null*

Started streaming 5038 records in less than 1 ms and completed after 1 ms, displaying first 1000 rows.

As we observe and as given on the given information, confirm that Each node or author has 224 features associated with it, which has been anonymized.

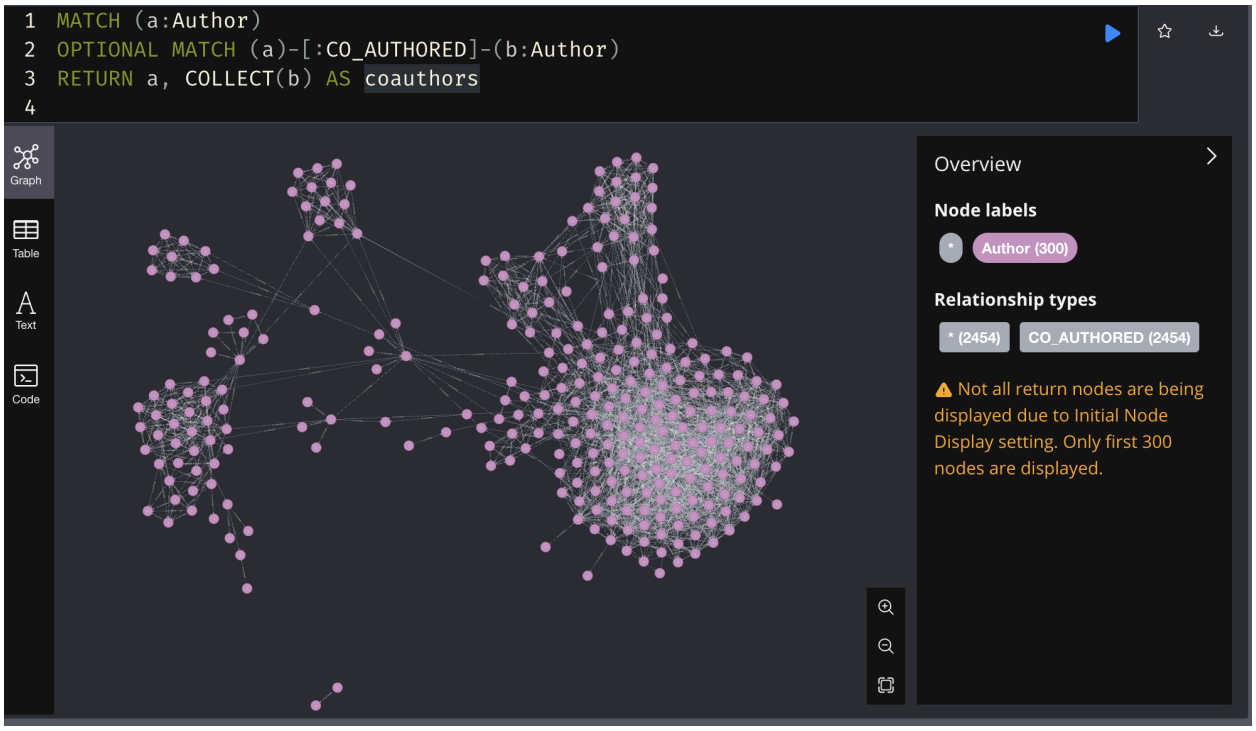
5. Query for CSV export

First, I tried a query that did not consider the authors that did not have co-authors earlier.



Finally I use the query below and export the data in table format to CSV for further use of GNN libraries.

```
MATCH (a:Author)
OPTIONAL MATCH (a)-[:CO_AUTHORED]-(b:Author)
RETURN a, COLLECT(b) AS coauthors
```



You can refer to how I proceed with the GNN Model creation and prediction following this link: https://docs.google.com/document/d/10AJrBY4tYDY6kR3r_sNLUXEk7394B3RCUSZSMk4FbU4/edit?usp=sharing

Or, you may refer to the repo, where I have uploaded the documentation for that.

Thank you for your patience, and an opportunity to explore the world of graph neural networks! :)