ANKARA UNIVERSITY

FACULTY OF ENGINEERING

COM4514 SPECIAL TOPICS II

TERM PROJECT REPORT

EFE ATEŞ

22290588

PURPOSE

Purpose of this report is to examine and learn behaviour of Neo4j computer programme under a movie database. Then we will complete this report by applying three different machine learning techniques: co-occurrence network, similarity graph, link prediction.

PROJECT STRUCTURE

```
movie_graph_analysis/
│
├── main.py
│
├── requirements.txt // for directly installing required libraries using pip command
│
└── readme.md
```

METHOD

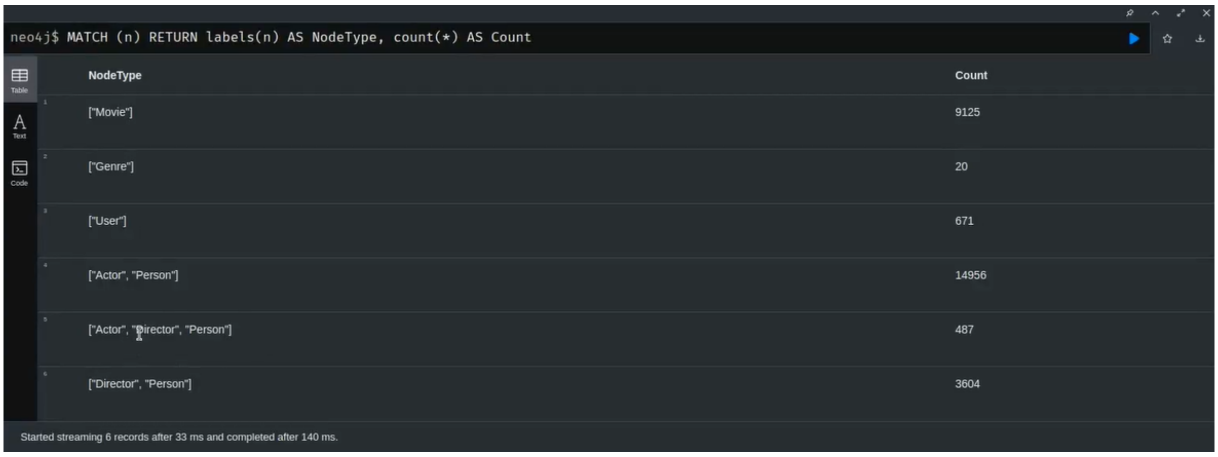
Method is to examine data using neo4j and doing technical analysis and obtaining its graph using python.

RESULTS

Graph models and queries listed below. Including key results, graphs, and findings. Or you can access the video by clicking here.

Identify node types:

Basic Query

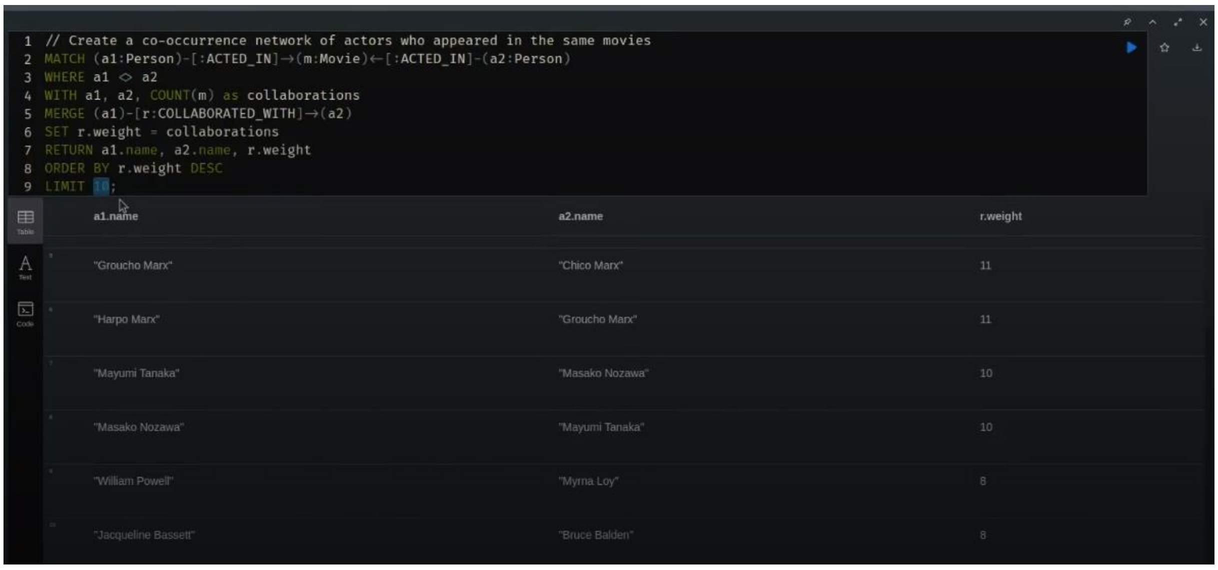


```
neo4j$ MATCH (n) RETURN labels(n) AS NodeType, count(*) AS Count
```

NodeType	Count
["Movie"]	9125
["Genre"]	20
["User"]	671
["Actor", "Person"]	14956
["Actor", "Director", "Person"]	487
["Director", "Person"]	3604

Started streaming 6 records after 33 ms and completed after 140 ms.

Co-occurrence network based on membership degree

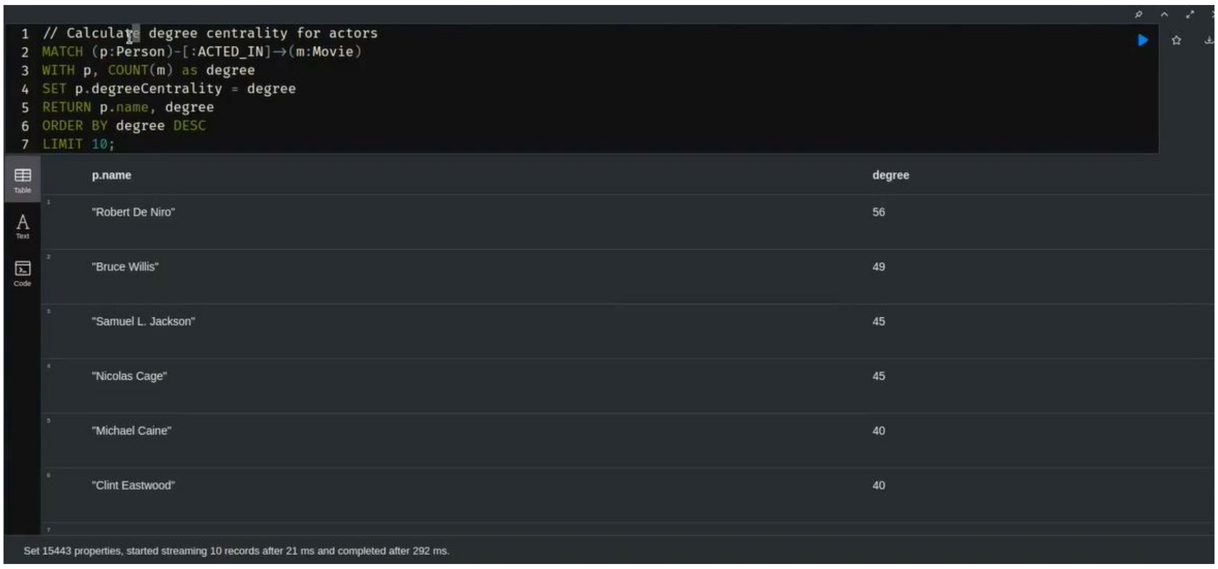


```
// Create a co-occurrence network of actors who appeared in the same movies
MATCH (a1:Person)-[:ACTED_IN]→(m:Movie)←[:ACTED_IN]-(a2:Person)
WHERE a1 <> a2
WITH a1, a2, COUNT(m) as collaborations
MERGE (a1)-[r:COLLABORATED_WITH]→(a2)
SET r.weight = collaborations
RETURN a1.name, a2.name, r.weight
ORDER BY r.weight DESC
LIMIT 10;
```

a1.name	a2.name	r.weight
"Groucho Marx"	"Chico Marx"	11
"Harpo Marx"	"Groucho Marx"	11
"Mayumi Tanaka"	"Masako Nozawa"	10
"Masako Nozawa"	"Mayumi Tanaka"	10
"William Powell"	"Myrna Loy"	8
"Jacqueline Bassett"	"Bruce Balden"	8

Perform exploratory graph analysis:

Calculate degree centrality for actors

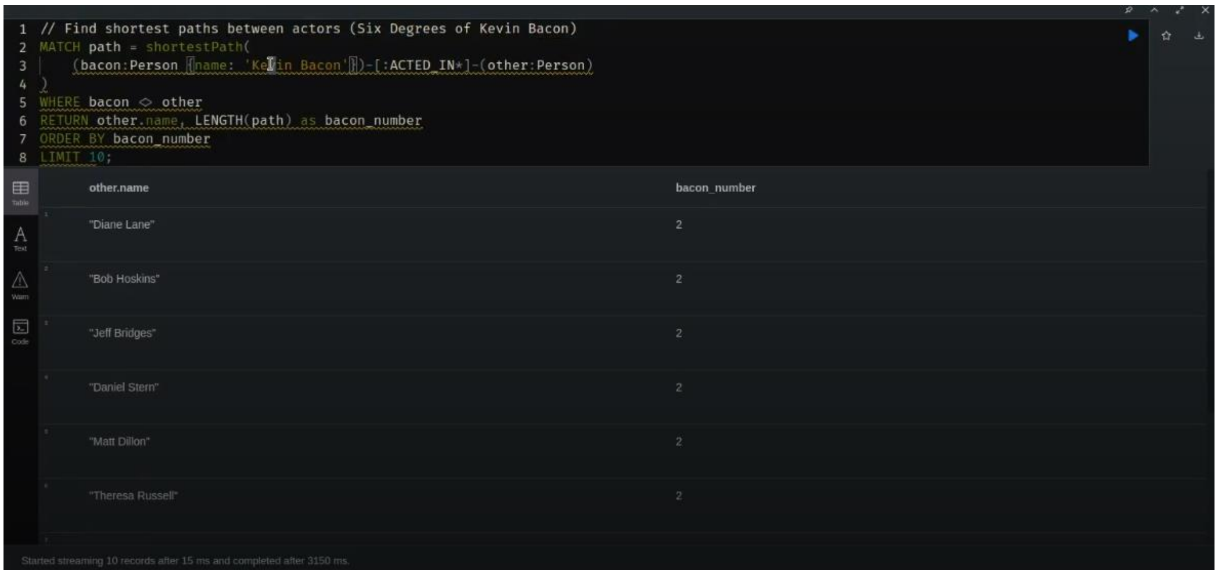


```
// Calculate degree centrality for actors
MATCH (p:Person)-[:ACTED_IN]→(m:Movie)
WITH p, COUNT(m) as degree
SET p.degreeCentrality = degree
RETURN p.name, degree
ORDER BY degree DESC
LIMIT 10;
```

	p.name	degree
1	"Robert De Niro"	56
2	"Bruce Willis"	49
3	"Samuel L. Jackson"	45
4	"Nicolas Cage"	45
5	"Michael Caine"	40
6	"Clint Eastwood"	40

```
Set 15443 properties, started streaming 10 records after 21 ms and completed after 292 ms.
```

Find the shortest paths between actors

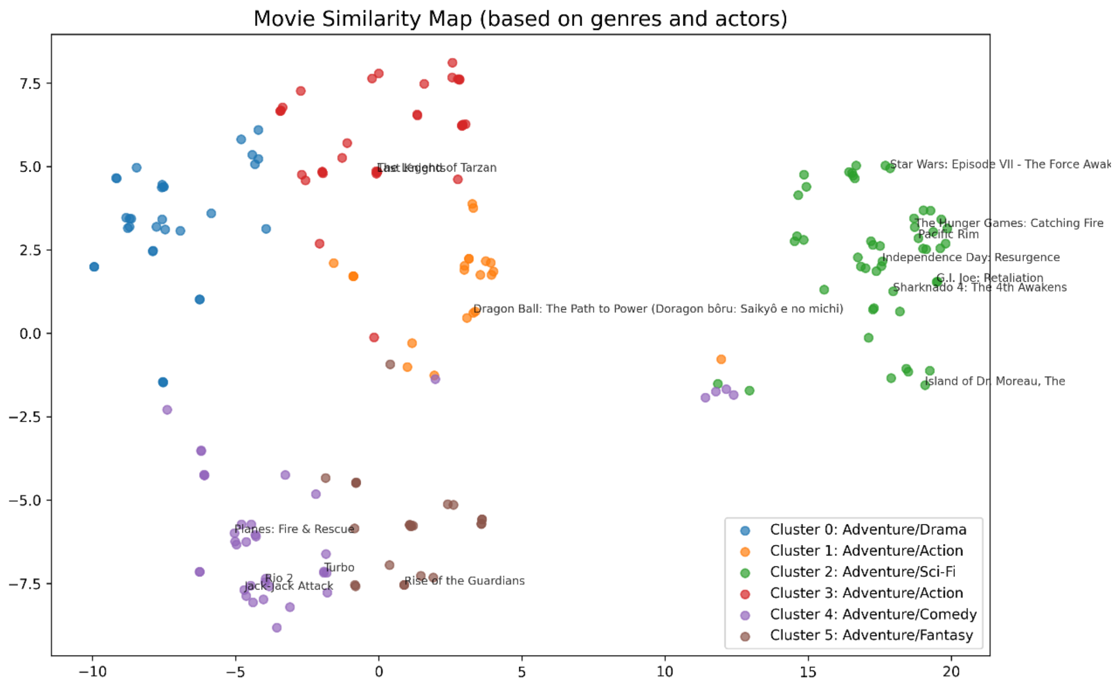


```
// Find shortest paths between actors (Six Degrees of Kevin Bacon)
MATCH path = shortestPath(
    (bacon:Person {name: 'Kevin Bacon'})-[:ACTED_IN*]-(other:Person)
)
WHERE bacon <> other
RETURN other.name, LENGTH(path) as bacon_number
ORDER BY bacon_number
LIMIT 10;
```

	other.name	bacon_number
1	"Diane Lane"	2
2	"Bob Hoskins"	2
3	"Jeff Bridges"	2
4	"Daniel Stern"	2
5	"Matt Dillon"	2
6	"Theresa Russell"	2

```
Started streaming 10 records after 15 ms and completed after 3150 ms.
```

Machine Learning side:



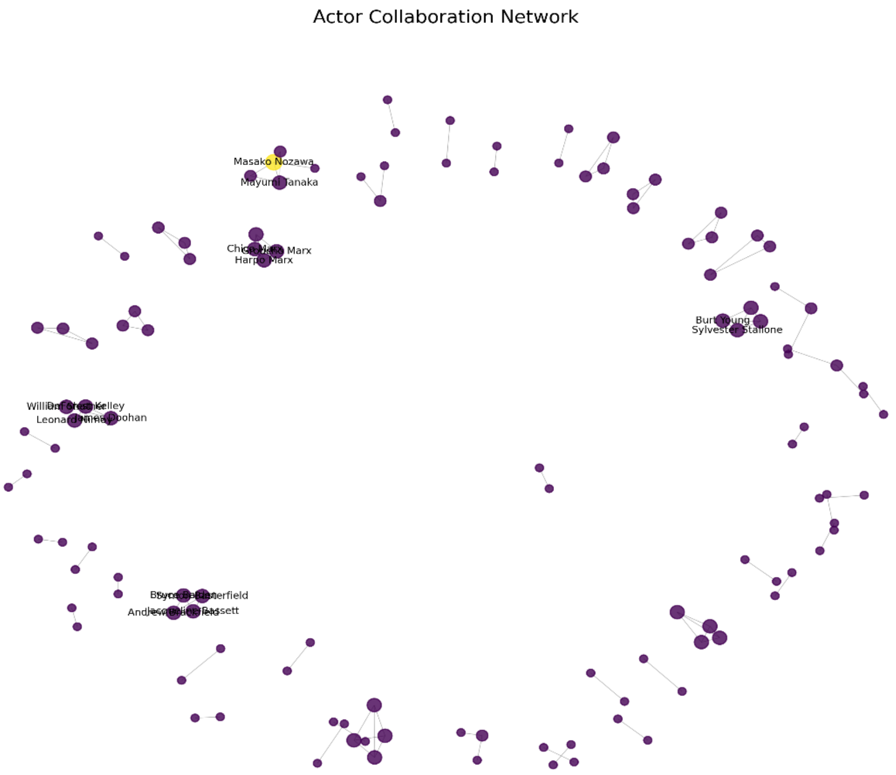
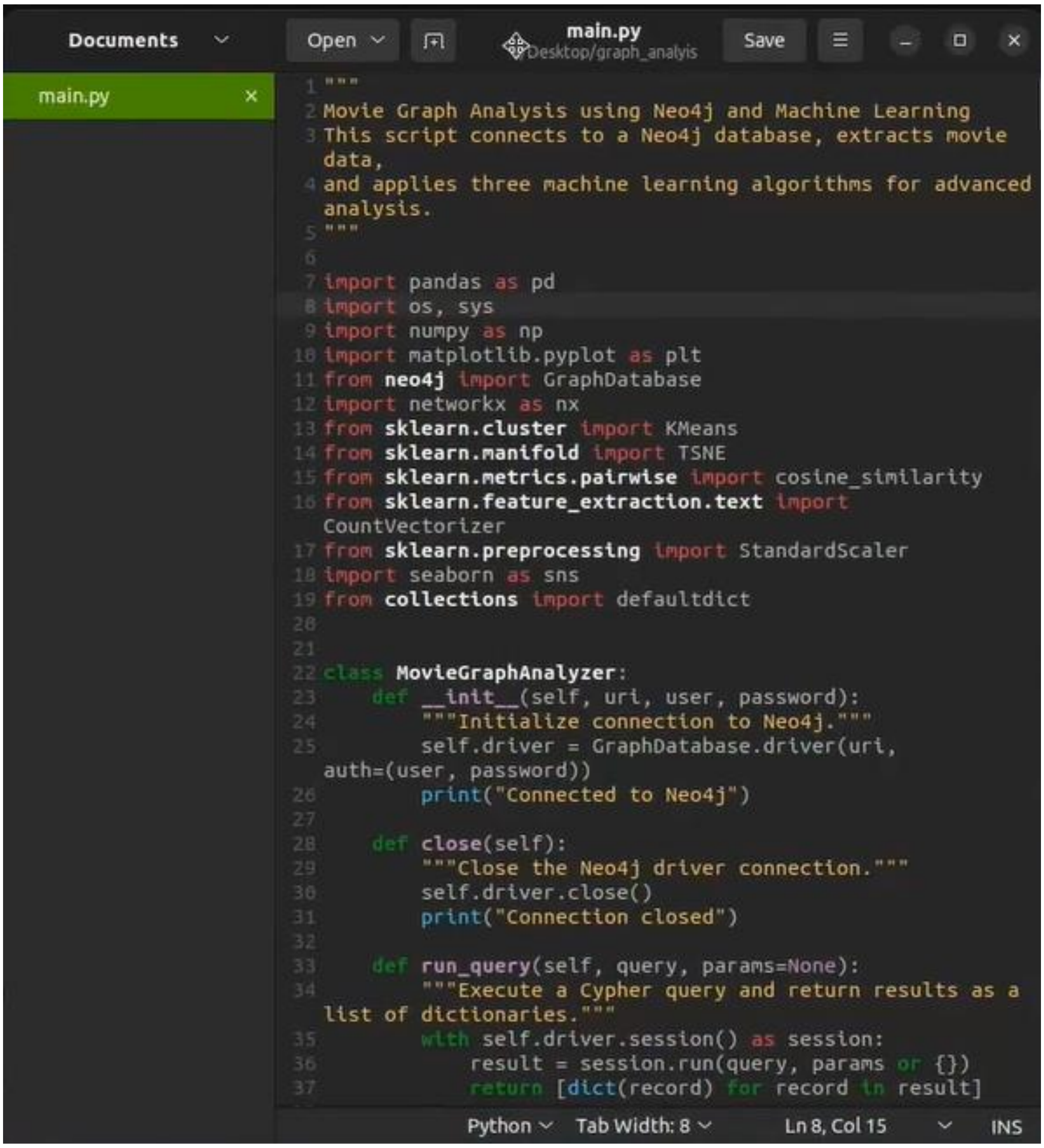


Figure that shows in order – Cluster Graph from similarity graph and co-occurrence network.

<u>Python Script:</u>

Python Script is also available on zip file. I could not obtain graph file but I obtained the result as string.



```
"""
Movie Graph Analysis using Neo4j and Machine Learning
This script connects to a Neo4j database, extracts movie data,
and applies three machine learning algorithms for advanced analysis.
"""

import pandas as pd
import os, sys
import numpy as np
import matplotlib.pyplot as plt
from neo4j import GraphDatabase
import networkx as nx
from sklearn.cluster import KMeans
from sklearn.manifold import TSNE
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import StandardScaler
import seaborn as sns
from collections import defaultdict


class MovieGraphAnalyzer:
    def __init__(self, uri, user, password):
        """Initialize connection to Neo4j."""
        self.driver = GraphDatabase.driver(uri, auth=(user, password))
        print("Connected to Neo4j")

    def close(self):
        """Close the Neo4j driver connection."""
        self.driver.close()
        print("Connection closed")

    def run_query(self, query, params=None):
        """Execute a Cypher query and return results as a list of dictionaries."""
        with self.driver.session() as session:
            result = session.run(query, params or {})
            return [dict(record) for record in result]
```

Figure that shows a snippet and start of code.

INNOVATION IDEAS

Franchise Ecosystem Analysis

By viewing film franchises as interconnected ecosystems, we can explore how narratives and character dynamics evolve across different instalments. This perspective not only highlights the relationships within the franchise but also helps us predict its long-term viability. We can assess how much the narrative relies on established characters and relationships versus its ability to adapt and incorporate new elements. This analysis sheds light on what makes a franchise successful and enduring in the ever-changing landscape of cinema.

Temporal Evolution of Cinematic Networks

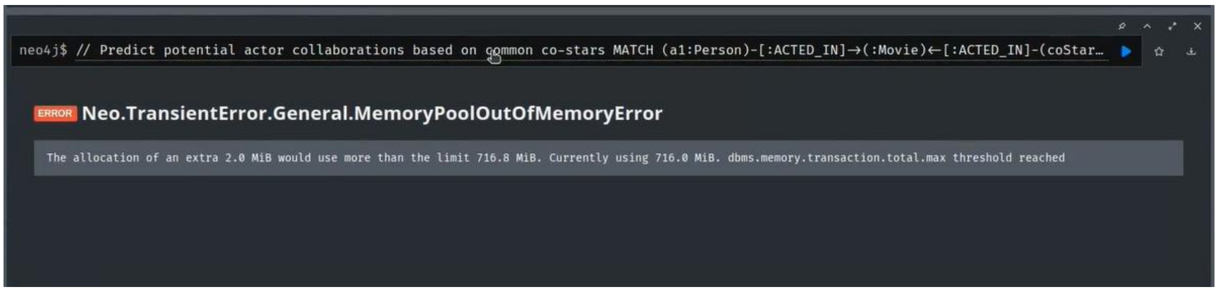
Implementing temporal analysis allows us to uncover the fascinating shifts in collaboration patterns throughout film history. As actors and directors progress in their careers, their networking behaviours often change significantly. Early-career actors might collaborate with a wide variety of partners, while seasoned professionals tend to form consistent, recurring relationships. By tracking these changes over time, we can identify pivotal moments when careers take unexpected turns, revealing insights into the complex nature of professional relationships in the film industry.

Representation and Industry Dynamics

Examining diversity metrics within the graph structure provides valuable insights into how representation has evolved over time. This analysis goes beyond mere statistics; it reveals deeper structural inequalities in who gets to collaborate with whom. By understanding these dynamics, we can see how certain groups gain access to influential networks while others remain marginalized. This exploration helps challenge existing power structures within the industry and promotes a more inclusive cinematic landscape.

DISCUSSION

Errors occurred when doing predictions for determining potential actor collaborations. It was due to lack of memory setting knowledge while determining query searches. Other mistake was for third technique script could not manage graph.



```
neo4j$ // Predict potential actor collaborations based on common co-stars MATCH (a1:Person)-[:ACTED_IN]→(:Movie)←[:ACTED_IN]-(coStar…

ERROR Neo.TransientError.General.MemoryPoolOutOfMemoryError

The allocation of an extra 2.0 MiB would use more than the limit 716.8 MiB. Currently using 716.0 MiB. dbms.memory.transaction.total.max threshold reached
```

Figure that is for first Error

CONCLUSION

Neo4j proves to be an exceptionally powerful tool for analysing complex, interconnected datasets like our movie database. The graph structure naturally accommodates the web of relationships that define the film industry, allowing us to extract meaningful patterns that would be difficult to detect using traditional relational databases. The three machine learning techniques we applied—co-occurrence networks, similarity graphs, and link prediction—each provide unique insights into various aspects of the cinematic landscape. From identifying established collaborative clusters to predicting future working relationships, these approaches demonstrate the analytical power of combining graph databases with machine learning algorithms. This exploration has implications beyond academic interest. The patterns revealed could inform casting decisions, help production companies identify promising collaborations, or enhance recommendation systems for streaming platforms. As the volume of entertainment data continues to grow, graph-based approaches like those enabled by Neo4j will become increasingly valuable for making sense of complex creative ecosystems and the relationships that drive them.

REFERENCES

1. *Com4514 Special Topics 2 lecture notes and project manual (20.05.2025 Access Date)*