Google Developer Groups
Cloud • Pune

Google Cloud

Community Day 2023

Google Cloud

Community
Day 2023

Introduction to Graph Data Science

Dr. Yogesh Haribhau Kulkarni

AI Coach

Google Developer Groups
Cloud • Pune

Hello!

I am Yogesh Kulkarni

- 20+ years in CAD/Engineering software development
- Got Bachelors, Masters and Doctoral degrees in Mechanical Engineering (specialization: Geometric Modeling Algorithms).
- Currently doing Coaching in fields such as Data Science, Artificial Intelligence Machine-Deep Learning (ML/DL) and Natural Language Processing (NLP).

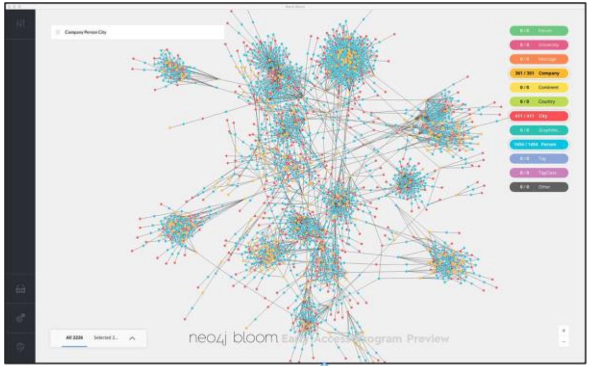
Feel free to follow me at:

- LinkedIn (www.linkedin.com/in/yogeshkulkarni/)
- Medium (yogeshharibhaukulkarni.medium.com)

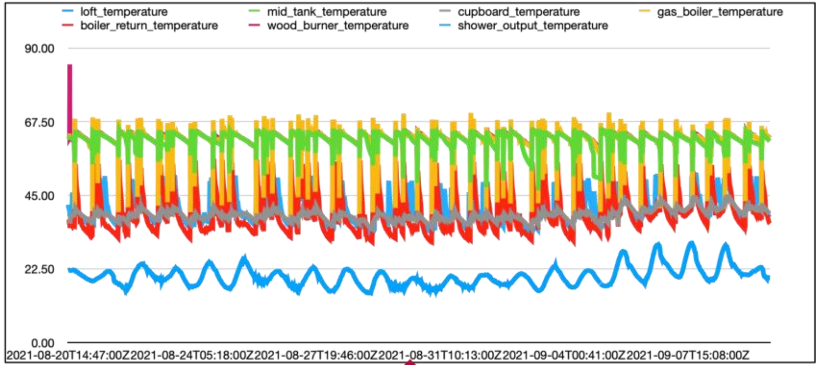
1.

What are Graphs?

First, one disclaimer

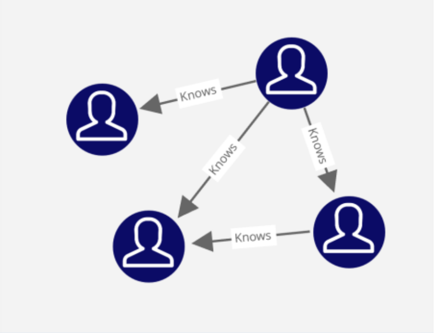


This is a graph



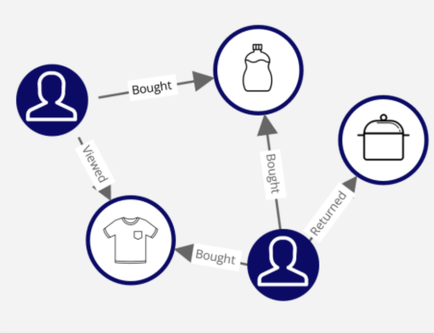
This is a chart

Everything is Naturally Connected



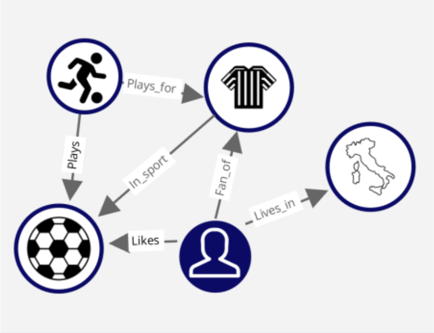
Networks of People

Employees, Customers, Suppliers, Partners, Influencers, etc.



Transaction Networks

Risk management, Supply chain, Orders, Payments, etc.

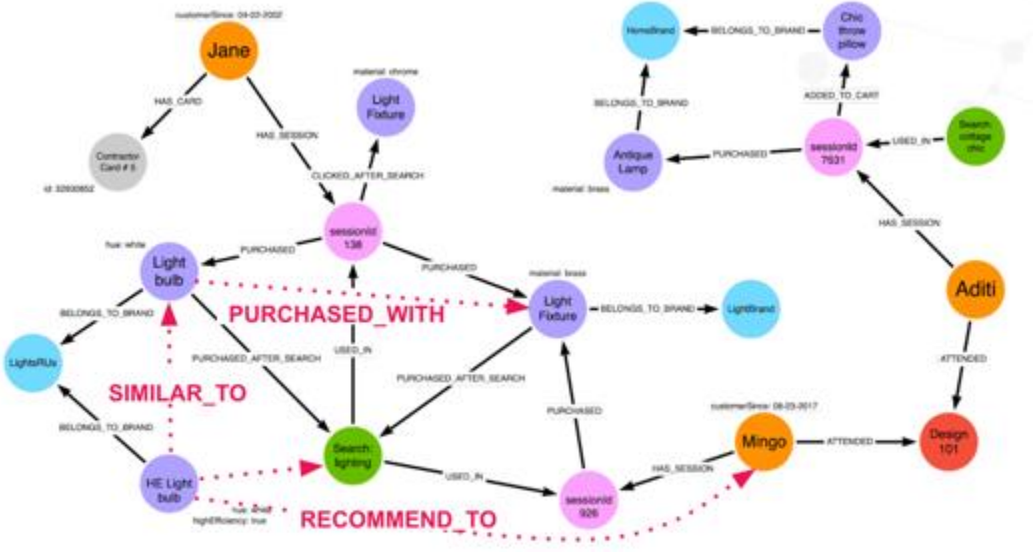


Knowledge Networks

Enterprise content, Domain specific content, eCommerce content, etc

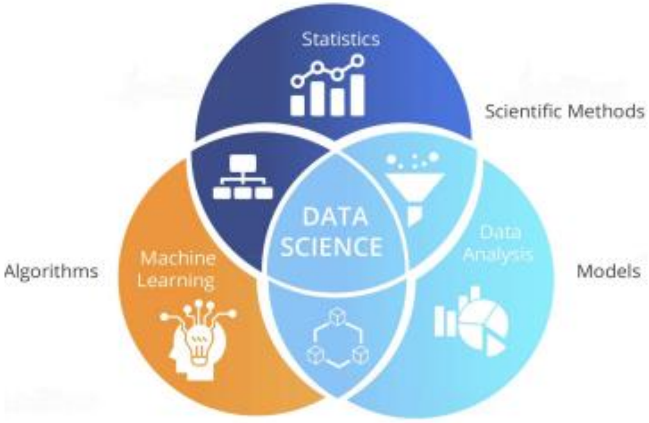
Take advantage of

- graphs to capture the network structure
- graph queries to store and retrieve relationships
- graph algorithms to infer relationships



Google Developer Groups
Cloud • Pune

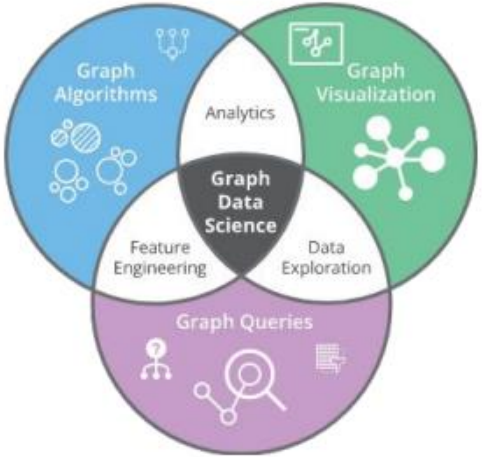
What is data science?



“Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data.” - Wikipedia

Mark Needham - Intro to Graph Data Science with Neo4j

What is GRAPH data science?

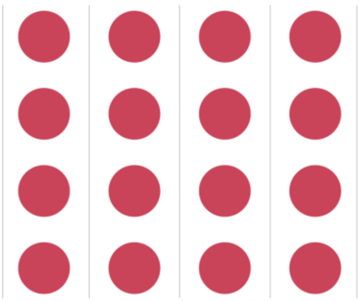


Graph Data Science is a science-driven approach to gain knowledge from the relationships and structures in data, typically to power predictions.

Data scientists use relationships to answer questions.

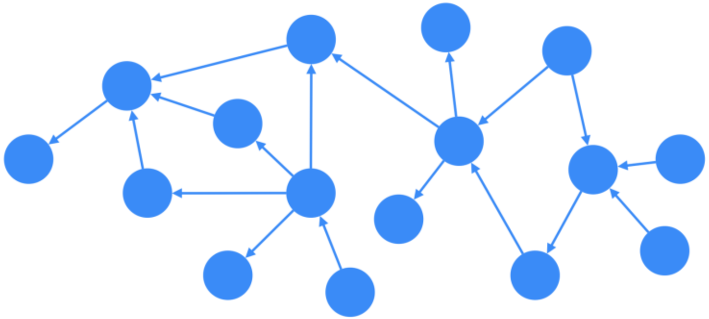
What’s different?

- Separate vs related
- Fixed vs variable size



Static approach

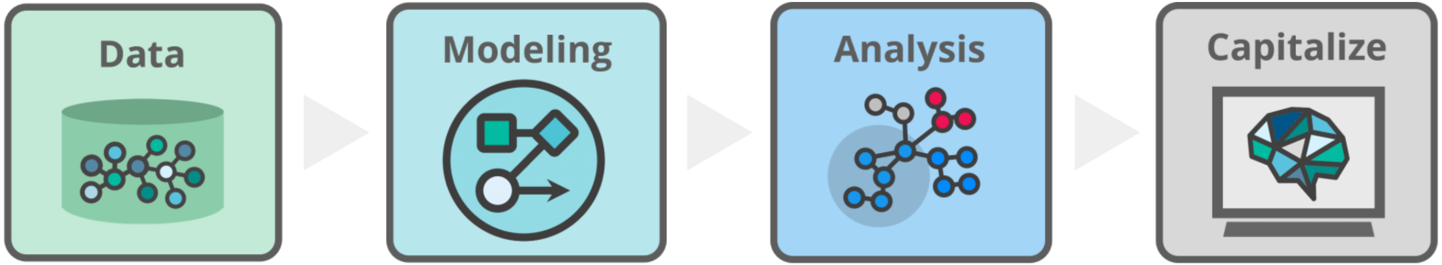
Discrete data, as represented in a relational database



Connected approach, or “graph”

Data represented as network in a “graph”-structure

Graphs enrich all stages of Data Science



Knowledge Graphs

Graph Feature Engineering and Graph ML

Graph Analytics, Investigations and Counterfactuals

Integrations and Knowledge Graphs for Heuristic AI

Graph Data Science Applications

Fraud Detection

Disambiguation & Segmentation

Personalized Recommendations

Life Sciences

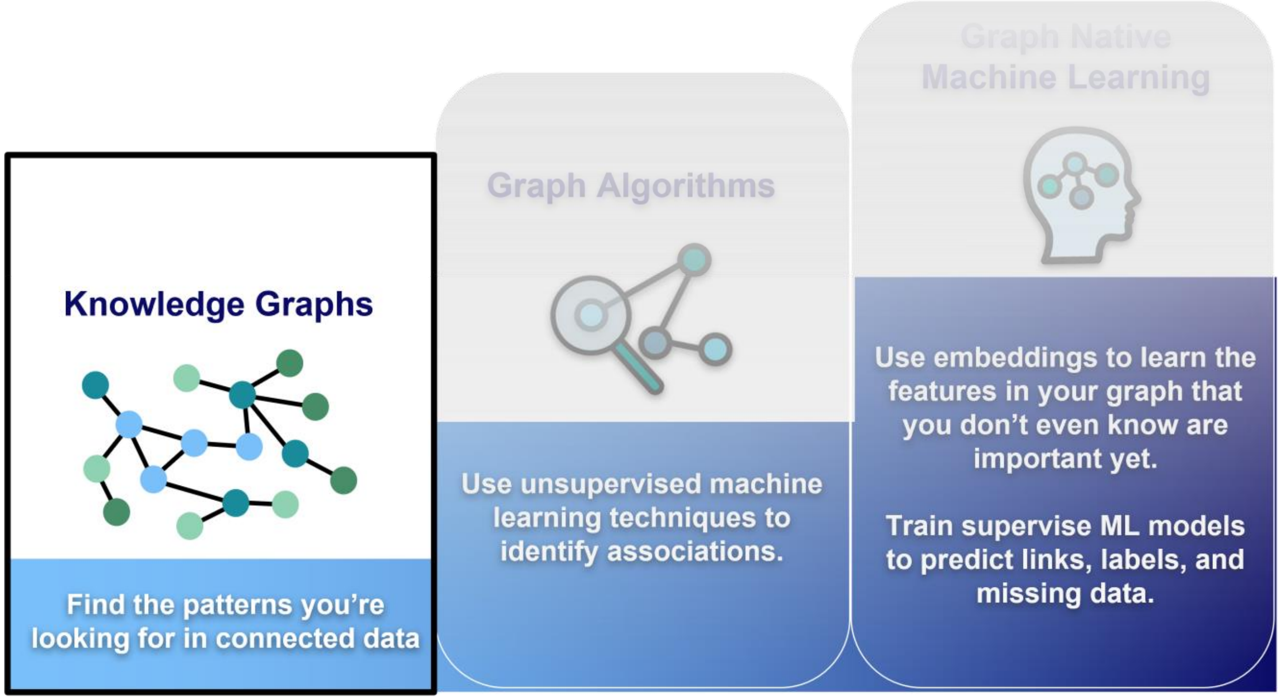
Churn Prediction

Search & Master Data Mgmt.

Predictive Maintenance

Cybersecurity

Section 1/3



Knowledge Graphs

Find the patterns you're looking for in connected data

Graph Algorithms

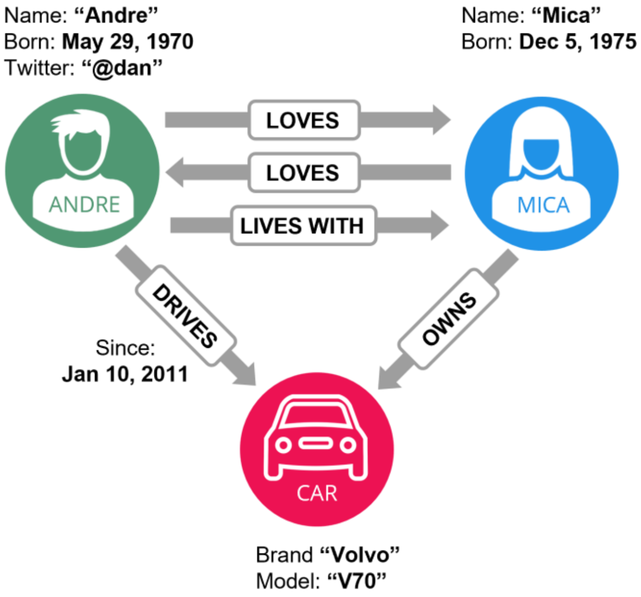
Use unsupervised machine learning techniques to identify associations.

Graph Native Machine Learning

Use embeddings to learn the features in your graph that you don't even know are important yet.

Train supervise ML models to predict links, labels, and missing data.

Knowledge Graph - Building Blocks



Node

Represents an entity in the graph

Relationship

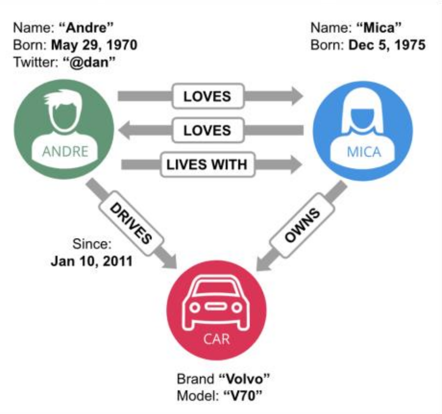
Connect nodes to each other

Property

Describes a node or relationship:
e.g. name, age, weight etc

Database, Query Language and Visualization

DATA



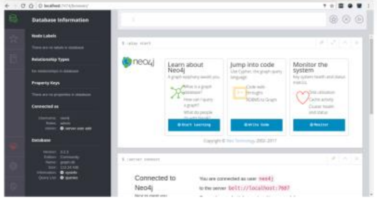
GRAPH QUERIES

```
MATCH (a)-[:KNOWS]->(b)
RETURN b.name
UNION ALL
MATCH (a)-[:LOVES]->(b)
RETURN b.name
```

CYPHER

https://neo4j.com/developer/cypher/

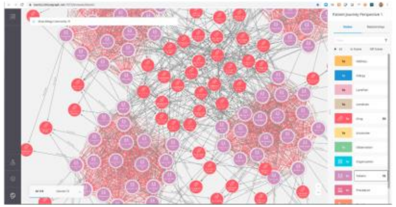
GRAPH VISUALIZATION



Connected to Neo4j

You are connected as user neo4j

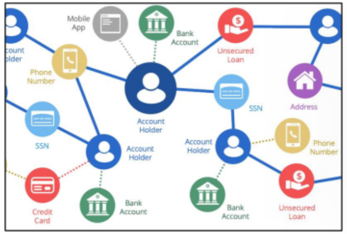
to the server bolt://localhost:7687



Google Developer Groups
Cloud • Pune

What can you do with a knowledge graph?

Financial Domain

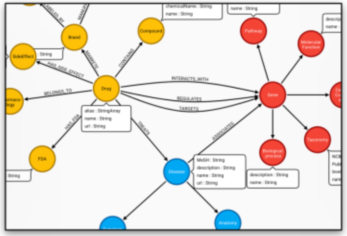


How many flagged accounts are in the applicant's network 4+ hops out?

How many login / account variables in common?

Add these metrics to your approval process

Life Sciences

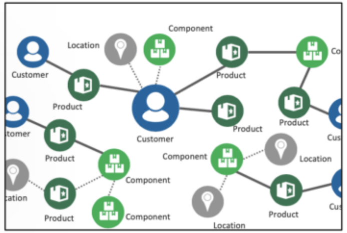


What completes the connections from genes to diseases to targets?

What genes can be reached 4+ hops out from a known drug target?

What mechanisms in common are there between two drugs?

Marketing and Recommendations



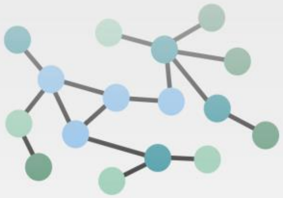
Collaborative filtering: users who bought X, also bought Y (open-ended pattern matching)

What items make you more likely to buy additional items in subsequent transactions?

Traverse hierarchies - what items are similar 4+ hops out?

Section 2/3

Knowledge Graphs



Find the patterns you're looking for in connected data

Graph Algorithms

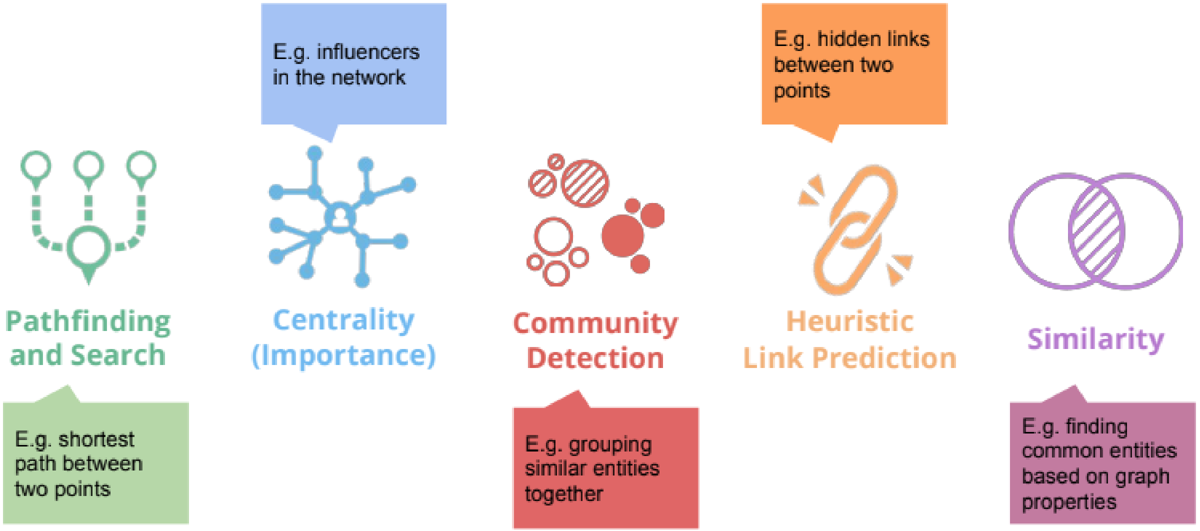
Use unsupervised machine learning techniques to identify associations.

Graph Native Machine Learning

Use embeddings to learn the features in your graph that you don't even know are important yet.

Train supervise ML models to predict links, labels, and missing data.

Graph Algorithm types



Pathfinding and Search

E.g. shortest path between two points

Centrality (Importance)

E.g. influencers in the network

Community Detection

E.g. grouping similar entities together

Heuristic Link Prediction

E.g. hidden links between two points

Similarity

E.g. finding common entities based on graph properties

Mark Needham - Intro to Graph Data Science with Neo4j

Graph Algorithms

Pathfinding & Search

- Shortest Path
- Single-Source Shortest Path
- All Pairs Shortest Path
- A* Shortest Path
- Yen's K Shortest Path
- Minimum Weight Spanning Tree
- K-Spanning Tree (MST)
- Random Walk
- Breadth & Depth First Search

Centrality & Importance

- Degree Centrality
- Closeness Centrality
- Harmonic Centrality
- Betweenness Centrality & Approx.
- PageRank
- Personalized PageRank
- ArticleRank
- Eigenvector Centrality
- Hyperlink Induced Topic Search (HITS)
- Influence Maximization (Greedy, CELF)

Community Detection

- Triangle Count
- K-Means
- Local Clustering Coefficient
- Connected Components (Union Find)
- Strongly Connected Components
- Label Propagation
- Louvain Modularity
- K-1 Coloring
- Modularity Optimization
- Speaker Listener Label Propagation

Supervised Machine Learning

- Node Classification
- Link Prediction
- Node Regression

Heuristic Link Prediction

- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors

Similarity

- Node Similarity
- K-Nearest Neighbors (KNN)
- Jaccard Similarity
- Cosine Similarity
- Pearson Similarity
- Euclidean Distance
- Approximate Nearest Neighbors (ANN)

Graph Embeddings

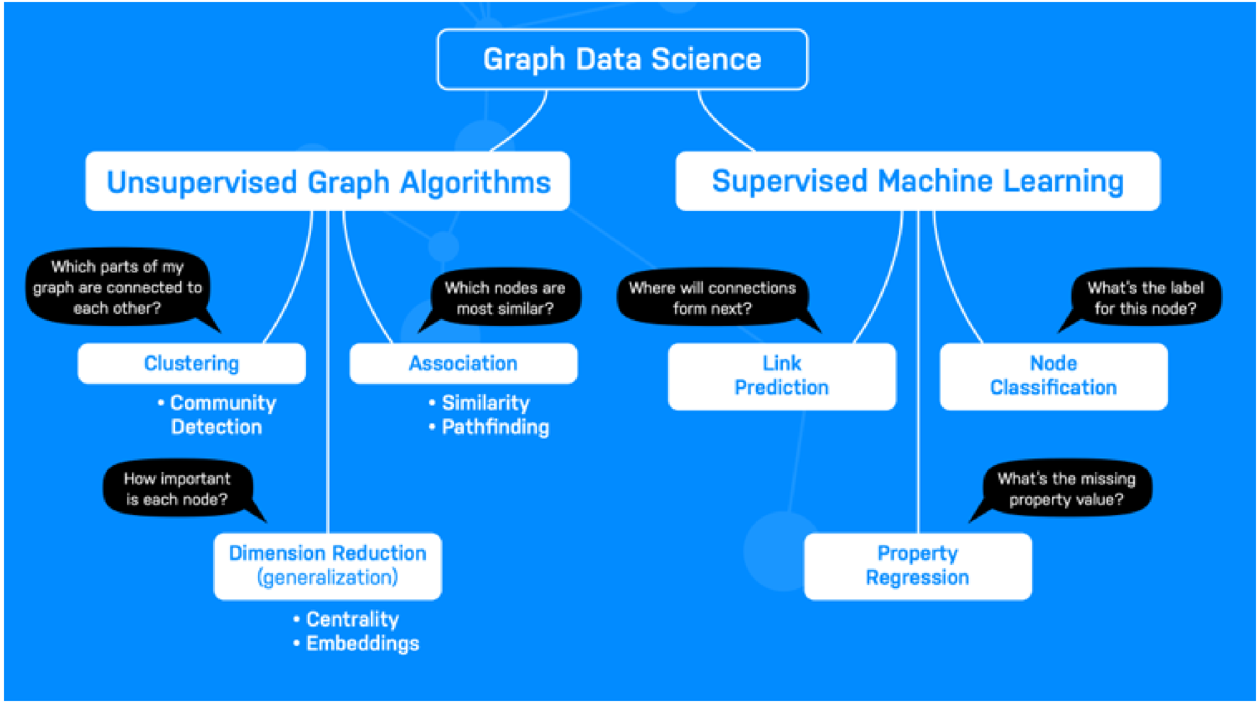
- Node2Vec
- FastRP
- GraphSAGE

… and more!

- Synthetic Graph Generation
- Scale Properties
- Collapse Paths
- One Hot Encoding
- Split Relationships
- Graph Export
- Pregel API (write your own algos)

Google Developer Groups
Cloud • Pune

What are Graph Algorithms?



Graph Data Science

Unsupervised Graph Algorithms

Which parts of my graph are connected to each other?

Clustering

- Community Detection

Which nodes are most similar?

Association

- Similarity
- Pathfinding

How important is each node?

Dimension Reduction (generalization)

- Centrality
- Embeddings

Supervised Machine Learning

Where will connections form next?

Link Prediction

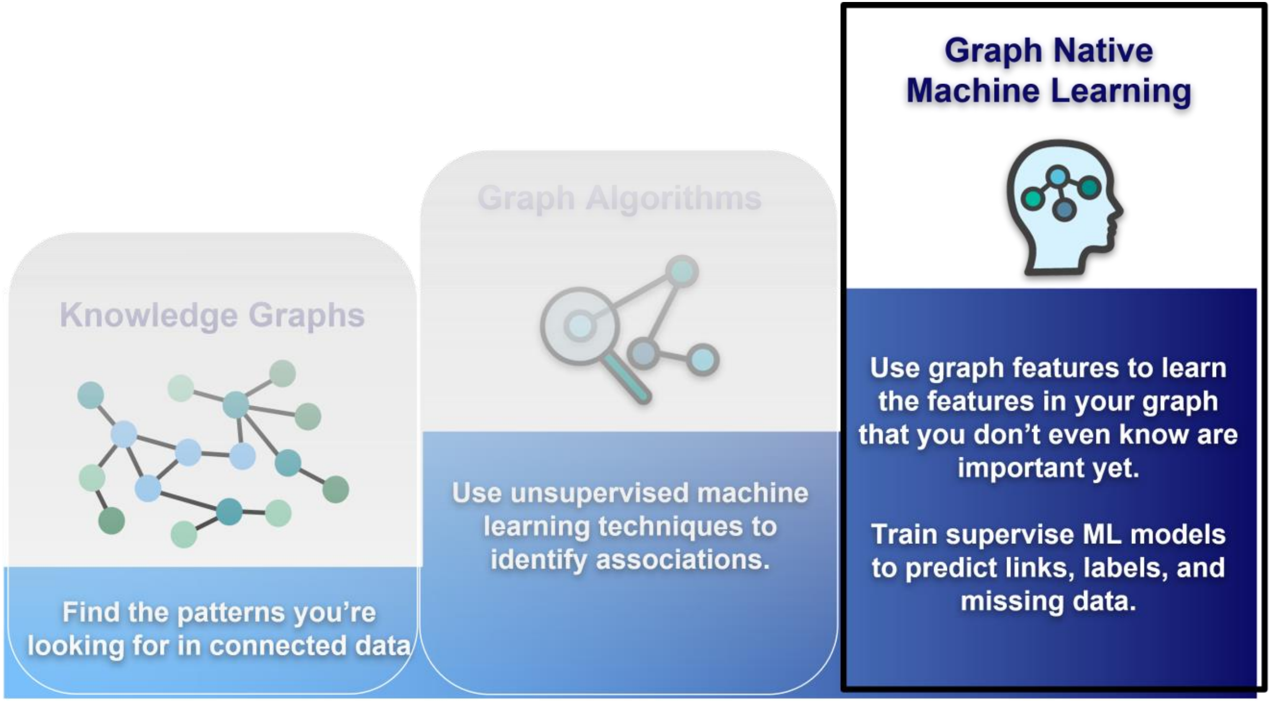
What's the label for this node?

Node Classification

What's the missing property value?

Property Regression

Section 3/3



Knowledge Graphs

Find the patterns you're looking for in connected data

Graph Algorithms

Use unsupervised machine learning techniques to identify associations.

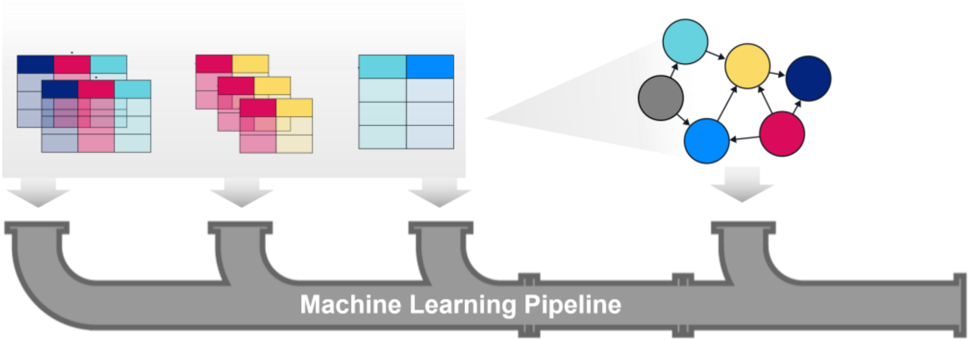
Graph Native Machine Learning

Use graph features to learn the features in your graph that you don't even know are important yet.

Train supervise ML models to predict links, labels, and missing data.

Relationship-Driven AI

- Traditional ML ignore network structure because it’s difficult to extract
- Use the right data structures to store and retrieve relationships
- Add relationships to AI/ML pipelines to make them contextual and to unlock otherwise unattainable predictions



Graph Feature Engineering

Queries

Human-crafted query, human-readable result

```
MATCH (p1:Person)-[:ENEMY]->(:Person)<-[:ENEMY]-(p2:PERSON)
MERGE (p1)-[:FRIEND]->(p2)
```

Algorithms

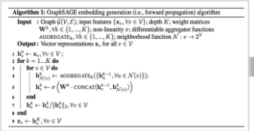
Predefined formula, human-readable result

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}$$

```
PageRank(Emil)    = 13.25
PageRank(Amy)     =  4.83
PageRank(Alicia)  =  4.75
```

Embeddings

AI-learned formula, machine-readable result



```
Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm
Input  : Graph $\mathcal{G}(\mathcal{V},\mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth $K$; weight matrices
         $\mathbf{W}^k, \forall k \in \{1,...,K\}$; non-linearity $\sigma$; differentiable aggregator functions
         $\text{AGGREGATE}_k, \forall k \in \{1,...,K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$
Output : Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$
$\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$ ;
for $k = 1...K$ do
    for $v \in \mathcal{V}$ do
        $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$;
        $\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\right)$
    end
    $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$
end
$\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$
```

```
Node2Vec(Emil)   =[5.4 5.1 2.4 4.5 3.1]
Node2Vec(Amy)    =[2.8 1.8 7.2 0.9 3.0]
Node2Vec(Alicia)=[1.4 5.2 4.4 3.9 3.2]
```

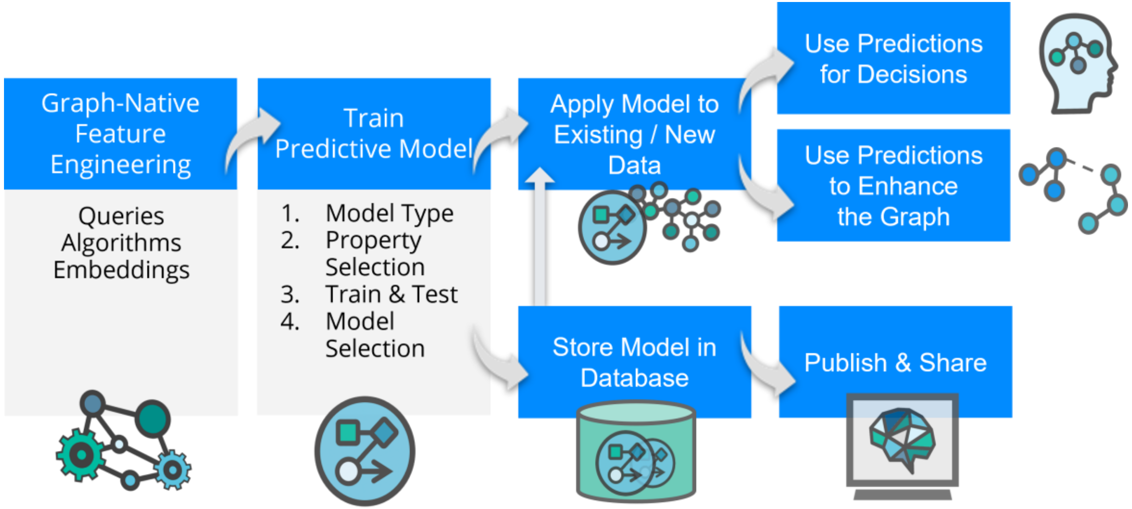
Machine Learning Workflows

Train ML models based on results

$f(x)$

Google Developer Groups
Cloud • Pune

Graph Machine Learning



Graph-Native Feature Engineering

Queries
Algorithms
Embeddings

Train Predictive Model

1. Model Type
2. Property Selection
3. Train & Test
4. Model Selection

Apply Model to Existing / New Data

Store Model in Database

Use Predictions for Decisions

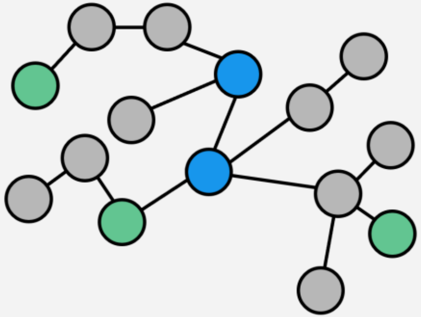
Use Predictions to Enhance the Graph

Publish & Share

The Only Completely In-Graph, ML Workflow

Google Developer Groups
Cloud • Pune

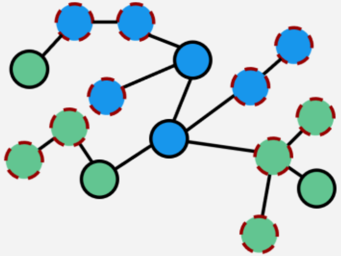
In-Graph Machine Learning



Labeled data: Pairs of nodes that are either linked or not
Features: Pre-existing attributes, algorithms (pageRank), embedding

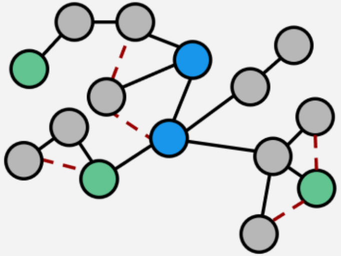
$f(x)$

Node classification: “What kind of node is this?”



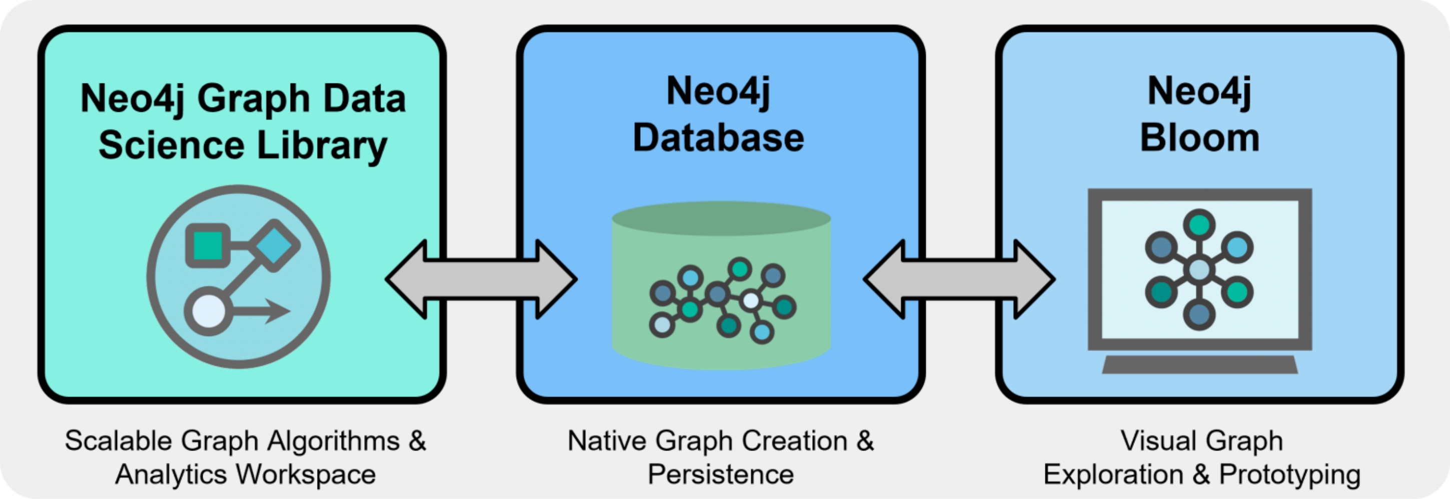
$f(x)$

Link prediction: “Should there be a relationship between these nodes?”

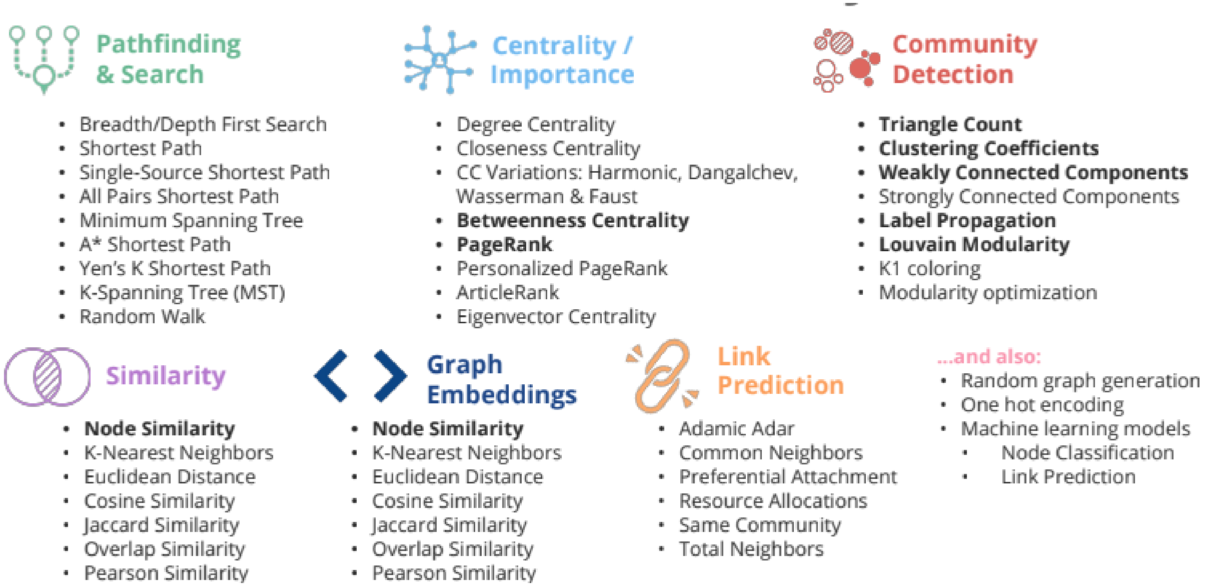


Google Developer Groups
Cloud • Pune

Neo4j Graph Data Science Framework


Neo4j Graph Data Science Library
Neo4j Database
Neo4j Bloom
Scalable Graph Algorithms & Analytics Workspace
Native Graph Creation & Persistence
Visual Graph Exploration & Prototyping

What’s available in GDS library?



Pathfinding & Search

- Breadth/Depth First Search
- Shortest Path
- Single-Source Shortest Path
- All Pairs Shortest Path
- Minimum Spanning Tree
- A* Shortest Path
- Yen’s K Shortest Path
- K-Spanning Tree (MST)
- Random Walk

Centrality / Importance

- Degree Centrality
- Closeness Centrality
- CC Variations: Harmonic, Dangalchev, Wasserman & Faust
- **Betweenness Centrality**
- **PageRank**
- Personalized PageRank
- ArticleRank
- Eigenvector Centrality

Community Detection

- **Triangle Count**
- **Clustering Coefficients**
- **Weakly Connected Components**
- Strongly Connected Components
- **Label Propagation**
- **Louvain Modularity**
- K1 coloring
- Modularity optimization

Similarity

- **Node Similarity**
- K-Nearest Neighbors
- Euclidean Distance
- Cosine Similarity
- Jaccard Similarity
- Overlap Similarity
- Pearson Similarity

Graph Embeddings

- **Node Similarity**
- K-Nearest Neighbors
- Euclidean Distance
- Cosine Similarity
- Jaccard Similarity
- Overlap Similarity
- Pearson Similarity

Link Prediction

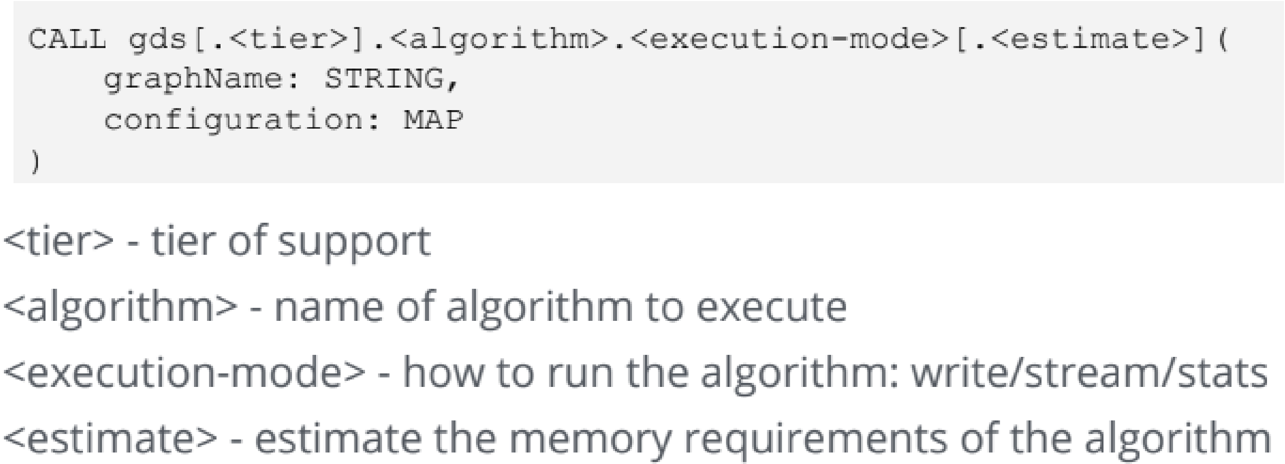
- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors

...and also:

- Random graph generation
- One hot encoding
- Machine learning models
  - Node Classification
  - Link Prediction

Mark Needham - Intro to Graph Data Science with Neo4j

GDS syntax



```
CALL gds[.<tier>].<algorithm>.<execution-mode>[.<estimate>](
    graphName: STRING,
    configuration: MAP
)
```

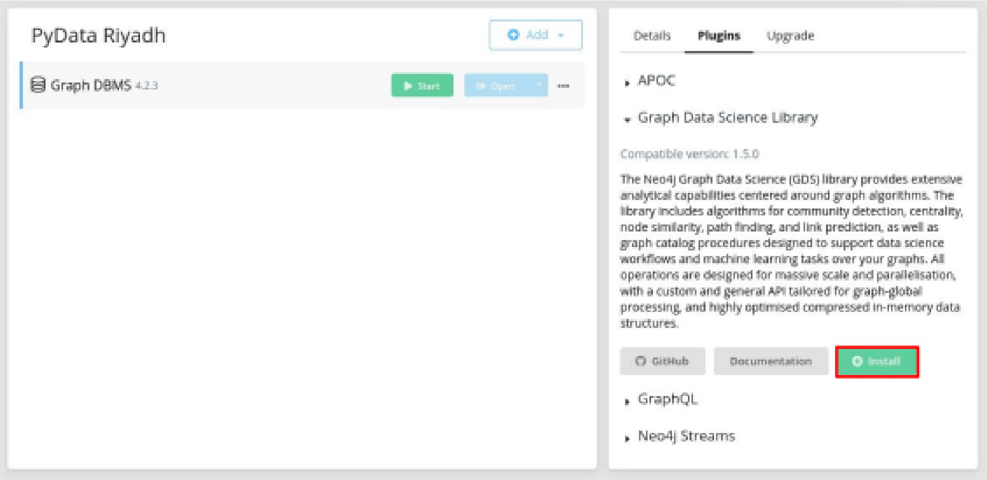
<tier> - tier of support

<algorithm> - name of algorithm to execute

<execution-mode> - how to run the algorithm: write/stream/stats

<estimate> - estimate the memory requirements of the algorithm

Where do I find the GDS library?



Compatible version: 1.5.0

The Neo4j Graph Data Science (GDS) library provides extensive analytical capabilities centered around graph algorithms. The library includes algorithms for community detection, centrality, node similarity, path finding, and link prediction, as well as graph catalog procedures designed to support data science workflows and machine learning tasks over your graphs. All operations are designed for massive scale and parallelisation, with a custom and general API tailored for graph-global processing, and highly optimised compressed in-memory data structures.

neo4j.com/developer/neo4j-desktop/

Mark Needham - Intro to Graph Data Science with Neo4j

Where do I find the GDS library?

- Download from: neo4j.com/download-center
- Documentation: neo4j.com/docs/graph-data-science/1.5

What Next?

Free online training and certification

- dev.neo4j.com/learn
- dev.neo4j.com/datasets

connect with the community:

- dev.neo4j.com/chat
- dev.neo4j.com/forum
- dev.neo4j.com/newsletter

Next developer events

- Live Streams - Weekly & Online
- Local Meetups neo4j.com/events

Get started - downloads & GitHub projects

- dev.neo4j.com/sandbox
- github.com/neo4j

Continue your graph journey
with Graph Academy

What is Graph Academy?
Free, Self-Paced, Hands-on Online Training to help you learn how to build, optimize and launch your Neo4j project, all from the Neo4j experts.

What's more?
2 free certifications designed to test you on your overall knowledge of Neo4j:

- Neo4j Graph Data Science Certification
- Neo4j Certified Professional

Interested? For more information visit:
www.graphacademy.neo4j.com

neo4j | NODES 23

Beginner to Expert. It All Connects Here.

October 26, 2023

Save the Date

Submit your talk by June 30, 2023

dev.neo4j.com/nodes23

References

- Neo4j Graph Data Science - Alison Cossette
- Neo4j supplied presentations: Graph Databases, Cypher.
- “an introduction to neo4j (graph database tutorial for beginners)" - Chris Hay
- CIS 6930 - Advanced Databases - Neo4j
- Learning Neo4j" - Wabri/LearningNeo4j - Github
- Neo4J Certification Sample Questions - https://wiki.glitchdata.com/index.php/Neo4J_Certification
- Neo4j Certified Professional: Exam Practice Tests - By Cristian Scutaru
- "Graph Data Science with Neo4j Graph Algorithms - Will Lyon" Youtube
- "Mark Needham - Intro to Graph Data Science with Neo4j" Youtube
- "THE POWER OF “GRAPH DATA SCIENCE” using Neo4j" Youtube
- "Using Neo4j Graph Data Science in Python to Improve Machine Learning Models" - Tomaz Bratanic
- "Graph Data Science for Supply Chains – Part 1" - Zach Blumenfeld

Google Developer Groups
Cloud • Pune

Thank You !

yogeshkulkarni@yahoo.com