The #1 Platform for Connected Data

WHITE PAPER

Artificial Intelligence & Graph Technology

Enhancing AI with Context & Connections

Amy E. Hodler, Mark Needham & Jake Graham

The #1 Platform for Connected Data

White Paper

TABLE OF CONTENTS

Artificial Intelligence & Graph Technology

Enhancing AI with Context & Connections

Amy E. Hodler, Mark Needham & Jake Graham

The idea of artificial intelligence (AI) has a long history. Loosely defined, AI is a solution or set of tools to solve problems in ways that mimic human intelligence. Usually its most practical goal is to make predictions – either classifying things (such as adding a label) or predicting a value (such as the next number expected in a series).

In a larger sense, AI is categorized as either **narrow** or **general**. Narrow AI is focused on performing one task very well, such as image recognition. More general AI includes multiple abilities around planning, language comprehension, object recognition, learning or problem solving. AI solutions today mostly fall into the narrow AI category, but they are becoming broader in their applicability to novel situations and therefore more powerful over time.

One way to make AI applications more widely capable is to provide them with context, surrounding them with related information to use in solving the problem at hand.

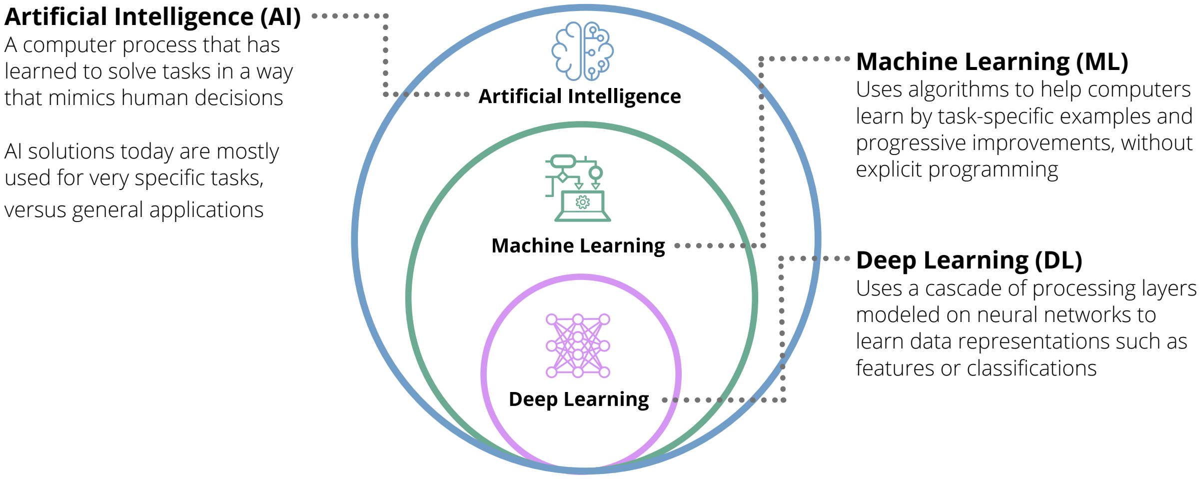
Consider the case of self-driving cars. Teaching autonomous vehicles to drive in rainy conditions is difficult because there is so much variability in rainy conditions (think rain on a sunny day, on a cloudy day, with sunlight coming from the left or right, windy rain, wintry mix, etc.).

If the autonomous vehicle's AI needs to see every possible combination of light and weather conditions, it would be impossible to train it for all possible situations. But if the AI is supplied with connected, contextual information (rain *and* night, night ***and*** temperature), it is possible to combine information from multiple contexts and infer the next action to take (like slowing down or turning on headlights).

Graph technology connects data and defines relationships. By enhancing AI with related context, graph technology offers an effective means to empower the development of sophisticated AI applications.

What Is Artificial Intelligence?

There are three major categories of AI technology that each solve problems in different ways. Artificial intelligence is an umbrella term that includes the subset of machine learning (ML) and deep learning (DL).



Artificial intelligence consists of several subsets of technologies that each solve problems in different ways.

The "learning" part of machine learning means that the algorithms iterate to optimize an objective function, like error or loss reduction. Machine learning is dynamic, with the ability to modify itself when presented with more data.

AI is a computer process that has learned to accomplish tasks in a way that mimics human decisions. Note that this does not require there to be actual intelligence. It does, however, leave the door open for many ways to perform tasks that are characteristic of human intelligence.

AI is the solution goal, and machine learning is essentially a method for achieving it.

Machine learning uses algorithms to learn by specific examples and progressive improvements, without explicit programming. "Training" an AI involves providing a lot of data to an algorithm to enable it to learn how to process that information. The "learning" part of machine learning means that the algorithms iterate to optimize an objective function, like error or loss reduction. Machine learning is dynamic, with the ability to modify itself when presented with more data.

Deep learning is a subset of machine learning that uses multiple layers to cascade learning and work with hierarchical abstractions. The "deep" part of deep learning refers to multiple hidden layers of abstraction. These layers enable feature hierarchies such as adding shape, size and smell to a fruit category.

Since AI aims to make choices the same way people do, it needs to look into the most important class of information people use to make decisions: context. AI needs context in order to mimic human intelligence.

The Importance of Context for AI

Context is crucial to decision making, for humans as well as artificial intelligence. Adults make tens of thousands of decisions every day (some say around 35,000), and most are dependent upon our surrounding circumstances or perspectives.

If we're making travel arrangements, our decisions vary significantly depending on whether the travel is for work, pleasure or with others. In language, meaning is highly dependent on the situation as well as who uses a phrase and their intonation. For example, a person who says "Get out!" may be expressing a friendly note of surprise or demanding that someone leave the room.

Let's say we're trying to solve a real-world problem: making a decision that requires a human to have the right contextual, relevant information and trying to automate or streamline that process in some way.

Humans use contextual learning to figure out what's important in a situation and how to apply that to new situations. For artificial intelligence to make decisions closer to the way humans do, it needs to incorporate a lot of context. Without peripheral and related information, AI requires more exhaustive training, more prescriptive rules and more specific applications.

4 Ways Graphs Provide Context

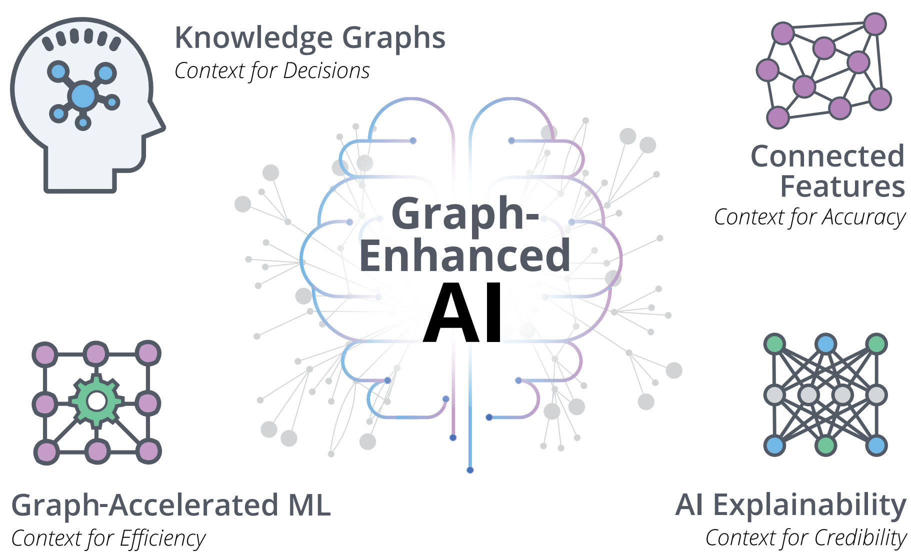
There are at least four main areas where graphs provide context for AI, which we'll detail in the following sections throughout the rest of this white paper

First is knowledge graphs, which provide context for decision support (e.g., for call center staff or support engineers) and help ensure that answers are appropriate to the situation (e.g., autonomous vehicles in rainy driving conditions).

Second, graphs offer greater efficiency of processing, and thus graph accelerated machine learning uses graphs to optimize models and speed up processes.

Third, connected feature extraction analyzes data to identify the most predictive elements within data. Basing a predictive model on strong characteristics found in the data improves accuracy.

Fourth and finally, graphs offer a way to provide transparency into the way AI makes decisions. This area is called AI explainability.



Knowledge Graphs: Context for Decisions

One of the AI areas that's moved into production fastest is decision support. Let's say we're trying to solve a real-world problem: making a decision that requires a human to have the right contextual, relevant information and trying to automate or streamline that process in some way.

CONTEXT-RICH KNOWLEDGE GRAPHS

Internal knowledge documents & files, with metadata tagging

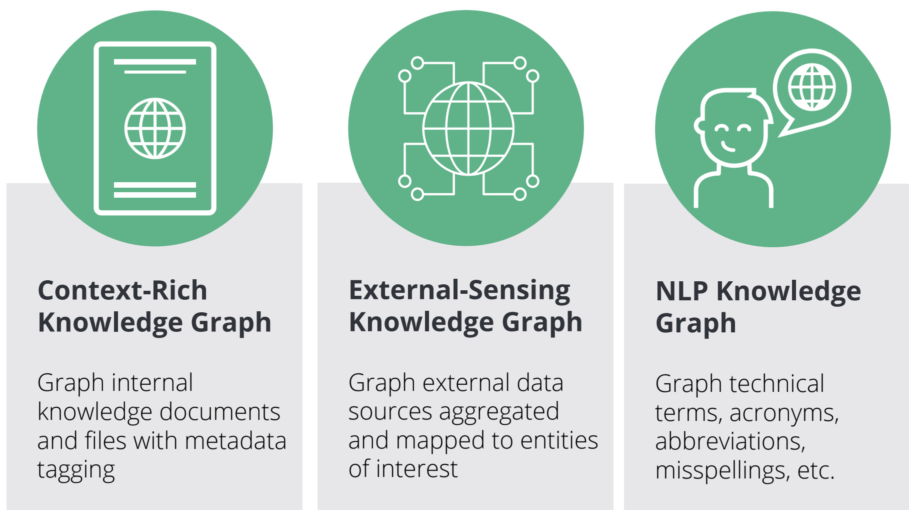
Examples:

- Search
- Customer support
- Document classification

Knowledge graphs offer a way to streamline workflows, automate responses and scale intelligent decisions. At a high level, knowledge graphs are interlinked sets of facts that describe real-world entities, facts or things and their interrelations in a human understandable form. Unlike a simple knowledge base with flat structures and static content, a knowledge graph acquires and integrates adjacent information using data relationships to derive new knowledge.

Here are some of the key characteristics of knowledge graphs:

- A knowledge graph needs to be connected around relevant attributes. Since not all data is knowledge, we're looking for pertinent information that's contextually related.
- A knowledge graph is dynamic in that the graph itself understands what connects entities, eliminating the need to program every new piece of information manually. A knowledge graph is able to make appropriate associations across attributes that are important to us because we've already programmed them in.
- A knowledge graph needs to be understandable. Sometimes we say it's semantic because the knowledge itself tells us what it is. The intelligent metadata helps us traverse the graph to find answers to specific problems, even when we don't know exactly how to ask for it.
- In practice, a knowledge graph usually contains heterogeneous data types. It combines and uncovers connections across silos of information.



Knowledge graphs fall into three categories related to the types of knowledge sought and the data used

There are three primary categories of knowledge graphs in the market today: context-rich, external-sensing and natural language processing (NLP).

Context-Rich Knowledge Graphs

A context-rich knowledge graph addresses the fact that simple keyword document searches or identifying the importance of a single word doesn't work well for a large corpus of heterogeneous knowledge. A knowledge graph enables us to incorporate the context of internal documents and files with metadata tagging. Connecting this information in a graph enables us to traverse that knowledge much faster.

EXTERNAL-SENSING KNOWLEDGE GRAPHS

External data source aggregation mapped to entities of interest

Examples

- Supply chain/compliance risk
- Market activity aggregation
- Sales opportunities

THOMSON REUTERS

pitney bowes

NLP KNOWLEDGE GRAPHS

Graph technical terms, acronyms, abbreviations, misspellings, etc.

Examples

- Improved search
- Chatbot implementation
- Improved classification

ebay

CAT

The most familiar use case for a context-rich knowledge graph is Google's search engine, but documentation classification and customer support are also common applications. For example, if we capture tens of thousands of complex technical support issues a year, being able to show a technician the most similar problem we've seen, how it was solved and the associated documents all greatly accelerate resolution.

A context-rich knowledge graph works well for organizations that have a great deal of knowledge captured in the form of documents. The knowledge graph helps to fill in the gap between having information (data collection) and being able to find and apply that information (data connection). One example is NASA's Lessons Learned database, which captures 50 years of knowledge about past missions and projects.

External-Sensing Knowledge Graphs

An external-sensing knowledge graph aggregates external data sources and maps them to internal entities of interest. For example, in evaluating supply chain risk, we may want to look at all our suppliers, all of the places they manufacture and all of our supply lines to analyze disruption risk. It is then possible to consider how a natural disaster in a specific location might impact the supply chain and to identify similar suppliers in different locations.

In general, we need to be able to incorporate an enormous amount of information from the market: sense the information, determine what's contextually relevant, and present it to the right person. Beyond supply chain monitoring, external insight sensing is used for analyzing compliance risk, the impact of market activity and sales opportunities.

For instance, Thomson Reuters (now Refinitiv) has a knowledge graph feed of financial content, enabling organizations to connect external and internal knowledge and make the best possible financial decisions quickly, often before the broader market has time to react.

Natural Language Processing Knowledge Graphs

Natural language processing (NLP) knowledge graphs incorporate the complexity and nuances of human language. NLP knowledge graphs require understanding a company's specific technical terms, product names, industry acronyms, part numbers and even common misspellings. This is where analysts create a knowledge graph to map meaning and build an ontology, which in turn improves search and delivers more relevant results.

Heavy equipment manufacturer Caterpillar uses NLP knowledge graphs to power natural language search and to extract meaning from thousands of warranty documents. Another example is the eBay App for Google Assistant, which uses all three types of knowledge graphs (context-rich, external-sensing and NLP) in order to guide shoppers to the perfect item.

Many of the implementations of AI applications using graphs today leverage knowledge graphs. The rest of this paper explores other areas where graph technology holds promise for AI applications.

Graph-Accelerated Machine Learning: Context for Efficiency

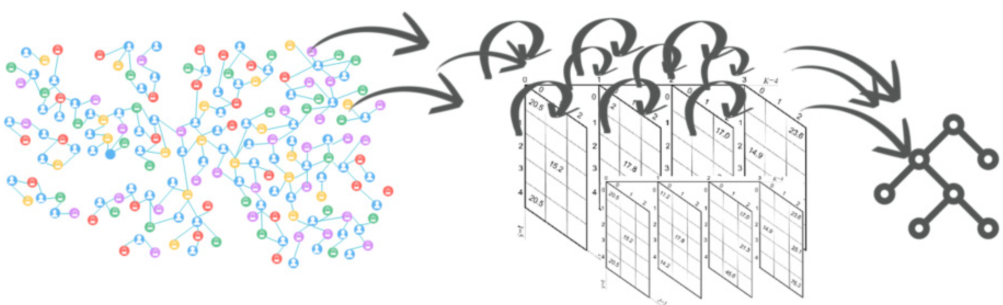
Current machine learning methods often rely on data stored in tables. Machine learning on such data is resource-intensive at best. More than half of enterprise CIOs surveyed state that iterative model training is one of their greatest challenges in taking AI projects from concept into production.

Graphs provide context for improved efficiency because data is connected, enabling relationships of numerous degrees of separation to be traversed and analyzed quickly at scale.

Graphs provide context for improved efficiency for machine learning algorithms because data is *already* connected in the graph model, enabling relationships of numerous degrees of separation to be traversed and analyzed quickly at scale. Thus the name **graph-accelerated machine learning.**

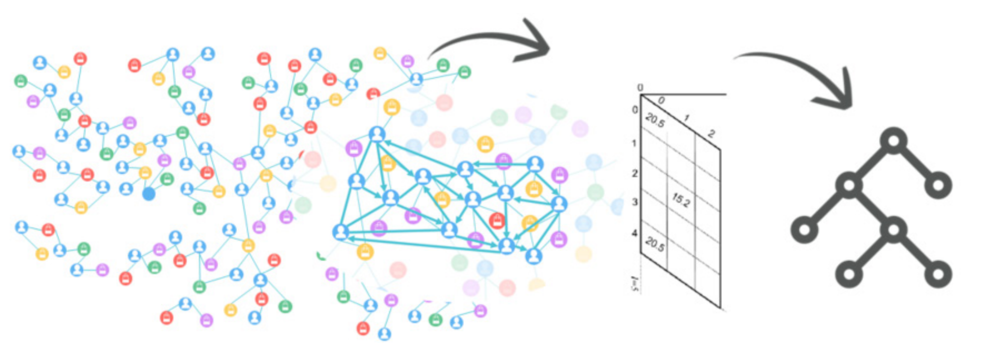
Humans naturally connect related information. As an example, consider how people think when asked, "What does this picture of a dog remind you of?" A human wouldn't need to run an intensive program, such as a nearest neighbor classifier, to compare that dog to all possible objects. We would quickly identify it with mammals – not humans or inanimate objects – and then classify it as a dog.

When data is stored as tables, it takes numerous iterations to connect it. For example, filtering processes are inefficient when they manifest relationships as table JOINs that bog down data pipelines. Data science practices such as collaborative filtering tend to require many JOINs as the result of multiple tables, indexes and lookup requirements.



For many machine learning systems, real-world connected data is stored as tables, then iteratively connected to produce decision trees.

Scale is another issue in machine learning efficiency. Machine learning algorithms may run a calculation against all the data. To avoid this, analysts may create subsets of data manually. These approaches tend to slow iterations down because they are either computationally intensive or require human involvement. A simple graph query accelerates this process by returning a subgraph containing only the needed data.



Training on connected data stored in a graph database is far more efficient. Graph filtering is quite effective, especially compared to typical manual sub-setting or statistical inference.

Using graphs, we quickly extract predicitve features and reshape the data to be usable in a machine learning pipeline. For example, from a graph we extract the relevant subset of the data (for example a strongly connected group) into a tabular format for model building.

"Increasingly we're learning that you can make better predictions about people by getting all the information from their friends and their friends' friends than you can from the information you have about the person themselves."

- James Fowler, *Connected*

Connected Features: Context for Accuracy

Relationships are often the strongest predictors of behavior.

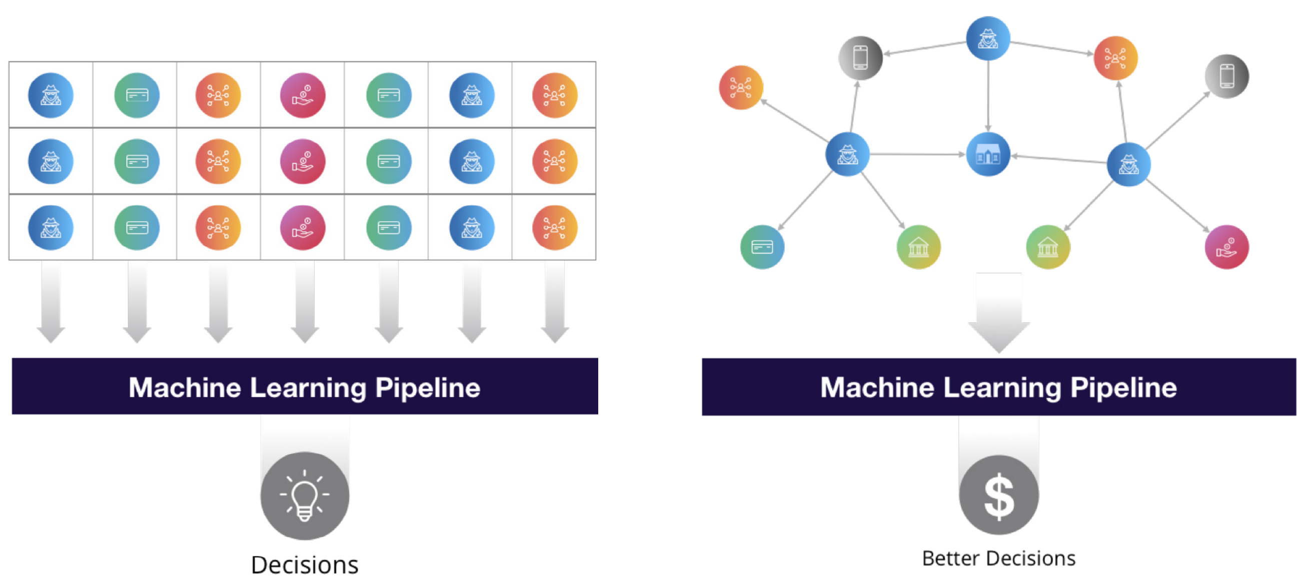
For example, studies show that your larger friend network is a better indicator of whether you will vote than even your immediate friendships (in this case, friends of friends have more influence than immediate friends). Connected features are connection-related metrics about our graph, such as the number of relationships going into or out of nodes, a count of potential triangles or neighbors in common.

Current machine learning methods typically rely on input data built from tables. This often means trying to abstract, simplify and – sometimes – entirely leaving out a lot of predictive relationships and contextual data. With connected data and relationships stored as graphs, it is straightforward to extract connected features and more easily incorporate all of this important information.

Connected features are used in many industries and have been particularly helpful for investigating financial crimes like fraud and money laundering. In these scenarios, criminals often try to hide activities through multiple layers of obfuscation and network relationships. Traditional methods may be unable to detect such behavior, and this is where graph extracted features excel.

There are a few different methods for using connected features. In the last section we covered feature extraction to reformat our data and even more valuable is feature engineering that combines and processes data to create *new*, more meaningful features.

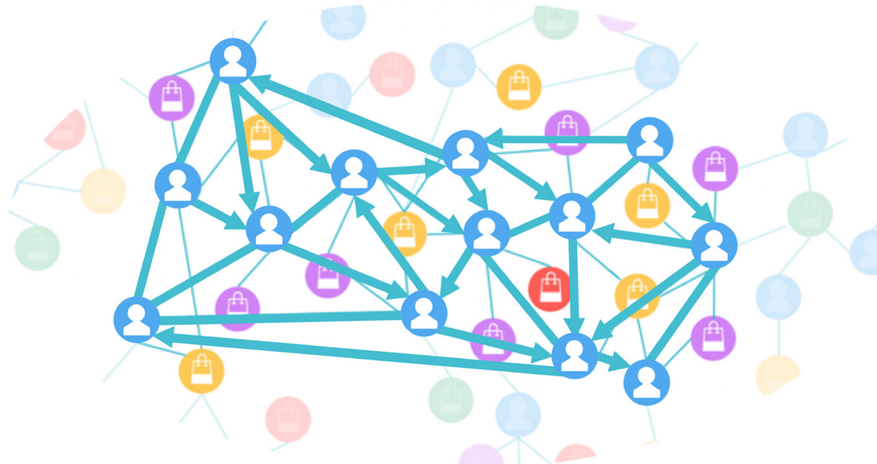
Connected feature engineering might involve simple queries or graph algorithms. When we know exactly what we're trying to find, such as identifying how many known fraudsters are in somebody's network, a specific query serves us well.



Traditional methods based on "flat data" simplify, or leave out entirely, predictive relationships. Graphs add highly predictive features to models, adding accuracy without altering current workflows.

DID YOU KNOW?

The Neo4j Graph Platform includes more than 35 graph algorithms for exploring and analyzing connected data. Register here for a free O'Reilly book on graph algorithms.



Graph community detection algorithms like the Louvain algorithm identify tight communities and relationship hierarchies for creating scored features for improving machine learning predictions.

However, we should use graph algorithms to find features where we know the *general structure* we want but not the exact pattern. For instance, graph algorithms simplify finding anomalies of tight communities that might be fraud rings or money laundering networks. We could then score nodes in our tightly-nit community and extract that information for training a machine learning model.

Finally, we use graph algorithms for feature selection to reduce the number of features used in a model to a relevant subset. For example, we might use algorithms like PageRank to find the features with the most influence, such as determining which attributes are most predictive of fraud. This helps eliminate less-important features and reduce overfitting, which causes models to be overtuned to their training data.

Using connected features maximizes the predictive power of our model while increasing how broadly the solution can be applied.

AI Explainability: Context for Credibility

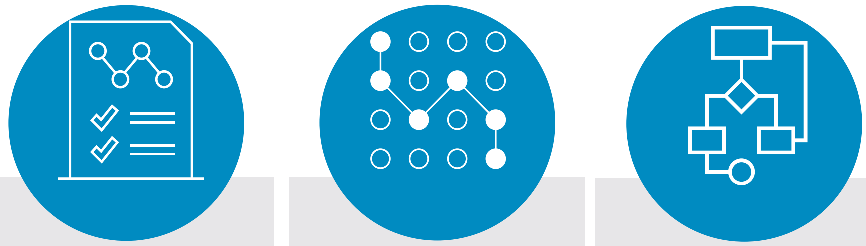
One challenge in AI adoption is understanding how an AI solution made a particular decision. The area of **AI explainability** is still emerging, but there's considerable research that suggests graphs make AI predictions easier to trace and explain.

This ability is crucial for long-term AI adoption because, in many industries, such as healthcare, credit risk scoring and criminal justice, we must be able to explain how and why decisions are made by AI. This is where graphs may add context for credibility.

There are many examples of ML and deep learning providing incorrect answers. For example, classifiers can make associations that lead to miscategorization, such as classifying a dog as a wolf. It is sometimes a significant challenge to understand what led an AI solution to make that decision.

Graphs tackle the explainable data issue fairly easily, using data lineage methods employed by most of the top financial institutions today. This requires storing our data as a graph but provides the ability to track how data is changed, where data is used and who used what data.

There are three categories of explainability that relate to the type of questions we're asking.



Explainable Data	**Explainable Predictions**	**Explainable Algorithms**
What data was used to train the model and why?	What features and weights were used for this particular prediction?	What are the individual layers and the thresholds used for a prediction?

Questions around AI explainability help us understand how data, predictions and algorithms influence decisions.

Explainable data means that we know what data was used to train our model and why. Unfortunately, this isn't as straightforward as we might think. If we consider a large cloud service provider or a company like Facebook with an enormous amount of data, it's difficult to know what exact data was used to inform its algorithms.

Graphs tackle the explainable data issue fairly easily, using data lineage methods employed by most of the top financial institutions today. This requires storing our data as a graph but provides the ability to track how data is changed, where data is used and who used what data.

Another area with significant potential is research into **explainable predictions**. This is where we want to know what features and weights were used for a particular prediction. There's active research into using graphs for more explainable predictions.

For example, if we associate nodes in a neural network to a labeled knowledge graph, when a neural network uses a node, we'll have insight into all the node's associated data from the knowledge graph. This allows us to traverse through the activated nodes and infer an explanation from the surrounding data.

Finally, **explainable algorithms** enable us to understand which individual layers and thresholds lead to a prediction. We are years from solutions in this area, but there is promising research that includes constructing a tensor in a graph with weighted linear relationships. Early signs indicate we may be able to determine explanations and coefficients at each layer.

The Neo4j Graph Platform empowers you to build intelligent applications faster, enabling you to find new markets, delight customers, improve outcomes and address your most pressing business challenges.

Conclusion

In this white paper, we've considered four ways graphs add context for artificial intelligence: ***context for decisions*** with knowledge graphs, ***context for efficiency*** with graph accelerated ML, ***context for accuracy*** with connected feature extraction, and ***context for credibility*** with AI explainability.

AI and machine learning hold great potential. Graphs unlock that potential. That's because graph technology incorporates context and connections that make AI more broadly applicable.

If you're building an AI solution, you need graph technology to give it contextual power. The Neo4j Graph Platform empowers you to build intelligent applications faster, enabling you to find new markets, delight customers, improve outcomes and address your most pressing business challenges.