Gartner.
Technical Professional Advice

Preparing and Architecting for Machine Learning

Published: 17 January 2017 **ID:** G00317328

Analyst(s): Carlton E. Sapp

As machine learning gains traction in digital businesses, technical professionals must explore and embrace it as a tool for creating operational efficiencies. This primer discusses the benefits and pitfalls of machine learning, the requirements of its architecture, and how to get started.

Key Findings

- Machine learning (ML) — a subset of artificial intelligence (AI) — is more than a technique for analyzing data. It's a system that is fueled by data, with the ability to learn and improve by using algorithms that provide new insights without being explicitly programmed to do so.
- Preparing data for ML pipelines is challenging when end-to-end data and analytic architectures are not refined to interoperate with underlying analytic platforms.
- ML is best-suited for dealing with big data. Organizations overwhelmed with data are using multiple ML frameworks to increase operational efficiencies and achieve greater business agility.
- Technical professionals are using machine learning to add elements of intelligence to software development and IT operations (DevOps) to gain operational efficiencies.
- The ML compute and storage cluster — which is the heart of the ML system — will vary based on learning method, learning application and need for automation.

Recommendations

To modernize your organization's business intelligence and analytics capabilities to support machine learning:

- Update the data organization layer in end-to-end analytics architectures to support data preparation for ML algorithms.
- Incorporate a development life cycle that supports learning models when the organization plans to aggressively build custom ML algorithms and applications.

- Choose an ML platform that supports and interoperates with multiple ML frameworks when the organization plans to leverage service providers or commercial off-the-shelf solutions. As AI and ML gain momentum, more frameworks will be packaged with solutions and service providers.
- Focus on storage and compute clusters to support machine learning capabilities. Choose the public cloud when you don't have the appropriate staff for engineering infrastructures for ML. The cloud is a great place for designing ML capabilities because of its elastic capabilities for scaling algorithms.

Table of Contents

List of Tables

List of Figures

Analysis

As technical professionals, much of what you do with your systems involves data. You strive to gain insight from that data in order to gain knowledge or a better understanding of your systems and their behavior, which allows you to make informed decisions. However, how you use data and extract a better understanding of systems can determine the level of success and competitive advantage.

The capability to transform data into actionable insight is the key to a competitive advantage for any organization. But the ability to autonomously learn and evolve as new data is introduced — without explicitly programming to do so — is the holy grail of business intelligence. That's what machine learning offers: a capability that accelerates data-driven insights and knowledge acquisition.

Machine learning has been around for decades. But due to the pervasiveness of data (from the Internet of Things [IoT], social media and mobile devices) and the seemingly infinite scalability of cloud-based compute power, ML has grabbed the center stage of business intelligence. Understanding and sophistication of the algorithms have expanded as well. While many ML algorithms have been around for years, the ability to apply complex mathematical calculations to data, and process them more quickly than ever before, is a recent development. This trend — in addition to the growing access to high volumes of data, more compute power and publicly visible success stories — is driving growing interest in exploiting ML to gain competitive advantages in business.

Information is being collected and generated from more sources than ever before, including sensors at the edge of IoT systems, social media, mobile devices, the web and traditional business data stores. Many organizations don't have the resources to derive all the business value they could from this mountain of information. Because ML can analyze data and derive predictions and inferences on its own, without the need for significant advance programming, this is opening up new opportunities to exploit the latent value in business data and gain a competitive edge.

> "An organization's ability to learn, and translate that learning into action rapidly, is the ultimate competitive advantage." — Jack Welch, former CEO of GE

The capability to transform learned data into business insight and action, extremely rapidly, is a disruptive one that can provide any organization with a competitive edge. This is why Gartner recommends that technical professionals adopt ML techniques as part of their personal "tradecraft," which will improve their ability to support digital business efforts — as well as tackle data management and operations challenges that arise within IT. See "Top Skills for IT's Future: Cloud, Analytics, Mobility and Security" for additional information on critical skills needed for IT professionals.

Gartner recommends that technical professionals engaged in data management and digital business take proactive steps now to gain knowledge and experience in ML, rather than waiting for business leaders to demand it and then having to play catch-up. This initiative analysis, aimed at the architects of digital business, discusses the ML technology basics, benefits and pitfalls, and how to get started. It answers the following questions:

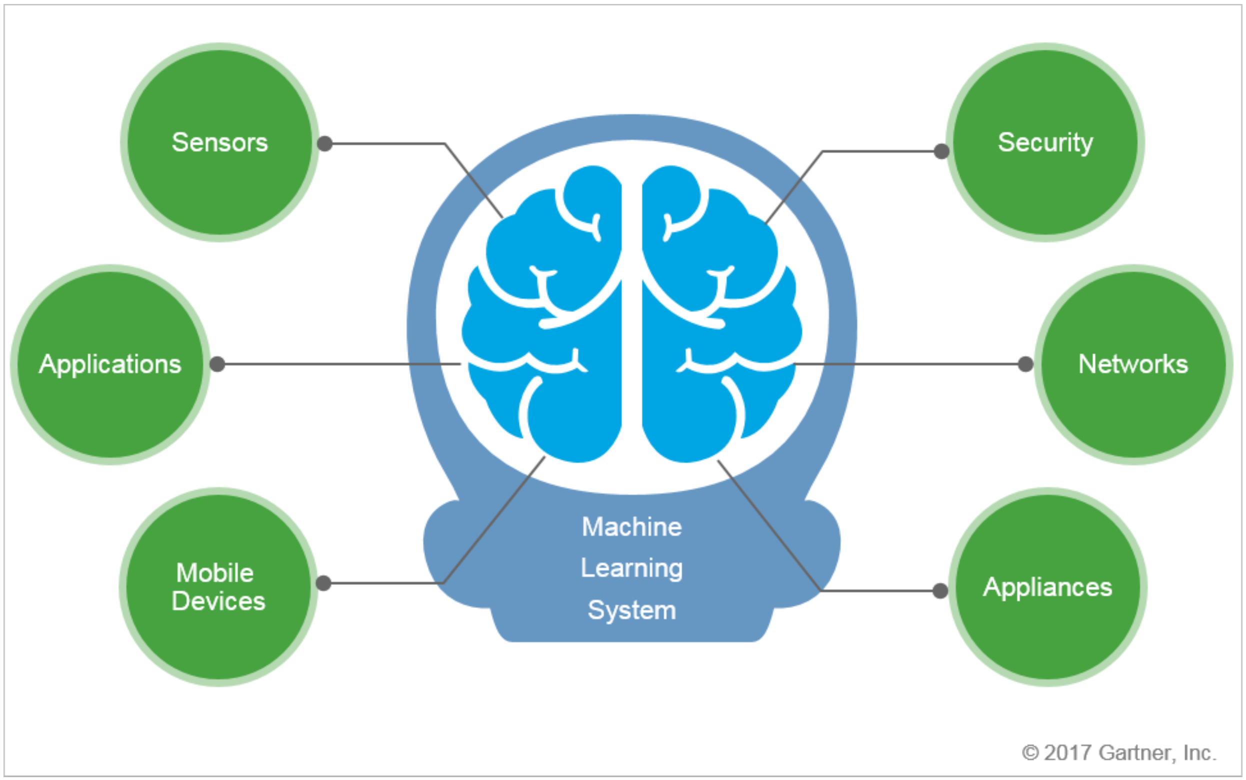
- What is machine learning?
- What business value does ML provide?
- What are the basics of the architecture, process and skills needed for ML?
- What steps should be taken to get started in ML?

What Is Machine Learning?

> Machine learning is not only for data scientists; it is tradecraft for digital architects.

ML is a type of data analysis technology that extracts knowledge without being explicitly programmed to do so. Data from a variety of potential sources (such as applications, sensors, networks, devices and appliances) is fed to the machine learning system, which uses that data, fit to algorithms, to build its own logic and to solve a problem or derive some insight (see Figure 1).

Figure 1. Learning From Data Without Being Explicitly Programmed



© 2017 Gartner, Inc.

Source: Gartner (January 2017)

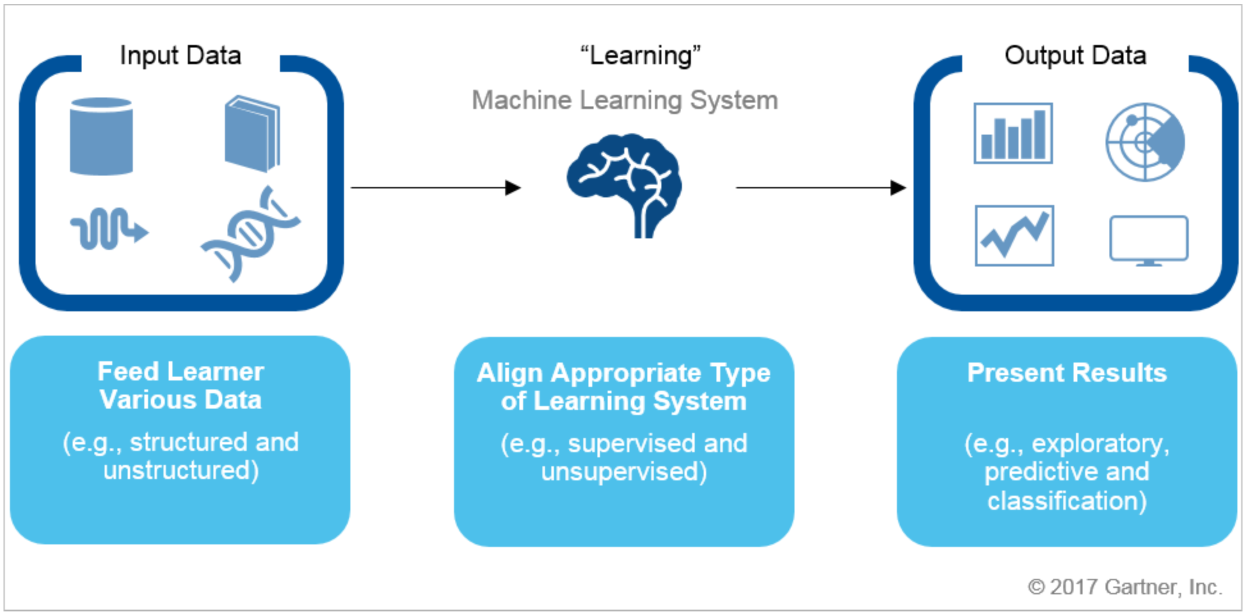
Formally defined, machine learning is a technical discipline that aims to extract knowledge or patterns from a series of observations (see "Hype Cycle for Emerging Technologies, 2016"). Depending on the type of observations provided, it can be split into three major subdisciplines:

- *Supervised learning,* where observations contain input/output pairs (aka labeled data): These sample pairs are used to "train" the machine learning system to recognize certain rules for correlating inputs to outputs. Examples include types of ML that are trained to recognize a shape based on a series of shapes in pictures.
- *Unsupervised learning,* where those labels are omitted: In this form of ML, rather than being "trained" with sample data, the machine learning system finds structures and patterns in the data on its own. Examples include types of ML that recognize patterns in attributes from input data that can be used to make a prediction or classify an object.
- *Reinforcement learning,* where evaluations are given about how good or bad a certain situation is: Examples include types of ML that enable computers to learn to play games or drive vehicles.

ML is a discipline that evolved from artificial intelligence, but it focuses more on cognitive learning capabilities. AI has many other aspects that attempt to model human function and intelligence (such as problem solving). However, ML is a subset technology specific to the use of data to simulate human learning. A key aspect of ML that makes it particularly appealing in terms of business value is that it does not require as much explicit programming in advance to gain intelligent insight because of its ability to use learning algorithms that simulate some human learning capabilities. Once data is acquired and prepared for ML, and algorithms are selected, modeled and evaluated, the learning system proceeds through learning iterations on its own to uncover latent business value from data. Although ML does not require much advance programming, it typically does require large amounts of raw data to work from — as well as high computing power on the execution platform to perform the computations needed to "learn." Note that there will still be a need to for programming the application of machine learning, especially as it is applied to automation.

The concept of ML is relatively simple (see Figure 2).

Figure 2. The Basics of Machine Learning Technology


Input Data
"Learning"
Machine Learning System
Output Data
Feed Learner Various Data
(e.g., structured and unstructured)
Align Appropriate Type of Learning System
(e.g., supervised and unsupervised)
Present Results
(e.g., exploratory, predictive and classification)
© 2017 Gartner, Inc.

Source: Gartner (January 2017)

The basics of ML involve input data, the learning process itself and output data:

- **Input data:** A wide variety of data can be used as input for ML purposes. This data could come from a variety of sources, such as enterprise systems, mainframe databases or IoT edge devices, and may be structured or unstructured in nature. Very high volumes of data are often fed into machine learners because more data often yields more insights. This is exacerbated by the digital business era, in which sources and volumes of information are exploding.
- **Learning:** Typically, the ML used for business purposes is either supervised or unsupervised in nature. Within these categories, however, there many different types of algorithms and ML routines, which can be used to accomplish different goals. Additionally, there are often different learning methods, such as "eager" and "lazy" learning methods. These methods govern how to process training data, and that governance will determine compute and storage requirements:

  - *Eager learning methods* evaluate training data and "eagerly" begin computing (for example, classification) before receiving new (test) data. They generally depend more on upfront evaluation of training data in order to compute (that is, predict) without the need for new data. As a result, eager learning methods tend to spend more time on processing the training data.
  - *Lazy learning methods* delay processing and data evaluation until new test data is provided — hence the term "lazy" or "lazy evaluation." As a result, lazy learning methods are often case-based, spending less time on the training data and more time on predicting.
- **Output data:** ML can be used to deliver results that are either predictive (that is, providing forecasts) or prescriptive (that is, suggesting recommended actions). The results can also deliver outputs that classify information or highlight areas for exploration. This output data may be stored for analysis, delivered as reports or fed as input into other enterprise applications or systems.

Table 1 provides descriptions of a few of the more common types of ML used in business, and it lists examples of the types of business applications they can be used to solve.

Table 1. Examples of Types of Machine Learning Algorithms/Problem Solving Approaches

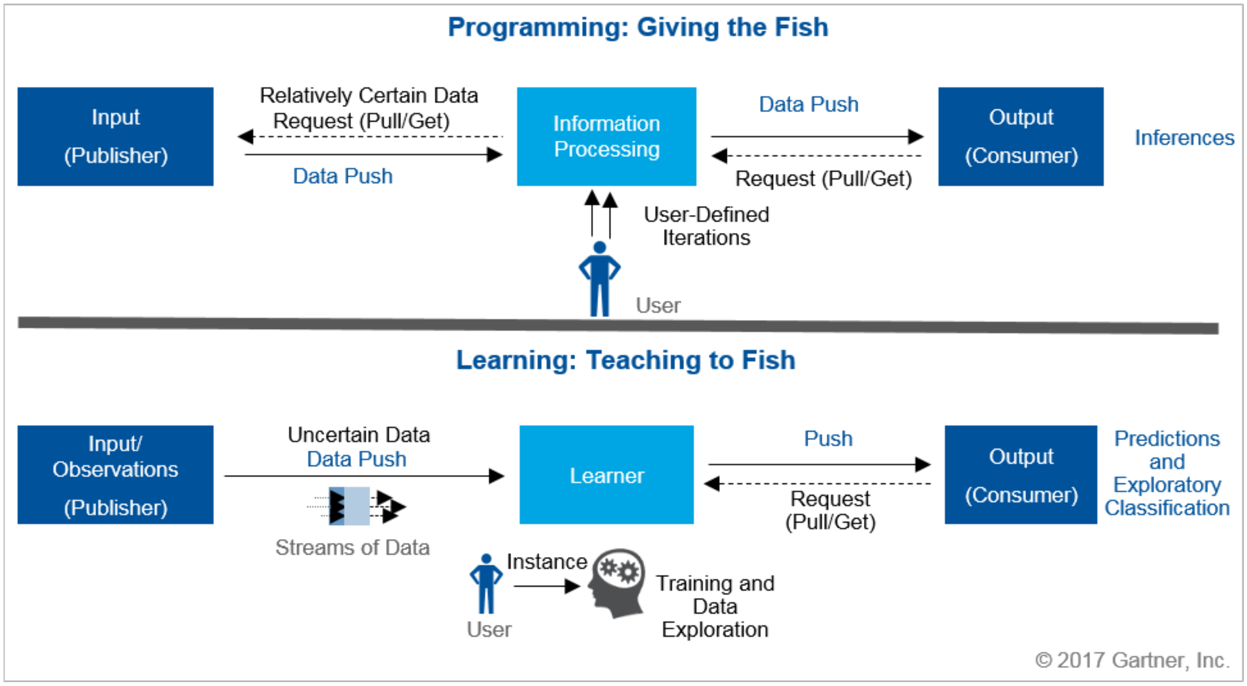
Type	Model/ Algorithm or Task	Description	Usage Examples in Business
Supervised	Neural network	Computations are structured in terms of interconnected groups, much like the neurons in a brain. Neural networks are used to model complex relationships between inputs and outputs to find patterns in data or to capture a statistical structure among variables with unknown relationships. They may also be used to discover unknown inputs (unsupervised).	■ Predicting financial results ■ Fraud detection
Supervised	Classification and/or regression	Computations are structured in terms of categorized outputs or observations based on defined classifications. Classification models are used to predict new outputs based on classification rules. Regression models are generally used to predict outputs from training data.	■ Spam filtering ■ Fraud detection
Supervised	Decision tree	Computations are particular representations of possible solutions to a decision based on certain conditions. Decision trees are great for building classification models because they can decompose datasets into smaller, more manageable subsets.	■ Risk assessment ■ Threat management systems ■ Any optimization problem where an exhaustive search is not feasible
Unsupervised	Cluster analysis	Computations are structured in terms of groups of input data (clusters) based on how similar they are to one another. Cluster analysis is heavily used to solve exploratory challenges where little is known about the data.	■ Financial transactions ■ Streaming analytics in IoT ■ Underwriting in insurance
Unsupervised	Pattern recognition	Computations are used to provide a description or label to input data, such as in classification. Each input is evaluated and matched based on a pattern identified. Pattern recognition can be used for supervised learning as well.	■ Spam detection ■ Biometrics ■ Identity management
Unsupervised	Association rule learning	Computations are rule-based in order to determine the relationship between different types of input or variables and to make predictions.	■ Security and intrusion detection ■ Bioinformatics

Type	Model/ Algorithm or Task	Description	Usage Examples in Business
			▪ Manufacturing and assembly

Source: Gartner (January 2017)

ML is about the process of programming to learn instead of programming for a single output. ML's main advantage over explicit programming can be summed up in the proverb: "Give a man a fish and you feed him for a day; teach a man to fish and you feed him for a lifetime" (see Figure 3). Programming a system to deliver value is like providing the system with the fish in the sense that the steps needed to solve the problem are explicitly provided to the system for each new problem that needs solving. In ML, once the system is provided with the right data and algorithms, it can "fish for itself" by performing its own logical processes to derive business value and insights. In fact, in many cases today, ML is already infused in the business in many packaged applications or commercial off-the-shelf (COTS) products.

Figure 3. Programming vs. Learning



Source: Gartner (January 2017)

Deep learning is a type of machine learning that is based on algorithms with extensive connections or layers between inputs and outputs. ML neural nets have inputs (variables), hidden layers (functions that compute the output) and output (results). In a simple example, imagine an ML neural net to detect a dog. This seemingly simple example may include a tail detector, ear detector, hair detector and so on. These detectors are combined into layers that contribute to detecting a dog. The more "detectors" you have, the deeper the neural net.

What Business Trends and Benefits Are Driving Machine Learning?

Business use of ML is gaining momentum due to the increasing pervasiveness of the technology and the rising discovery of business benefits that can be derived from its use. The data-rich nature that underpins a digital business, along with other big data sources and trends, has also been a major driver.

The huge masses of data that are now being collected from IoT sensors and other new information sources are overwhelming the abilities of businesses to interpret them and derive value and insights from them. Because ML can relatively quickly and efficiently sift through and interpret these mountains of data, many businesses are seizing the opportunity to uncover latent insights that could deliver a competitive edge. Gaining such an edge — with the speed and efficiency that ML offers — is particularly critical in the digital business world, where key data-driven insight obtained by one player in the market can drive competitors out of business.

ML is particularly well-suited to gaining a competitive edge in digital business because it offers:

- **Speed** to support faster compute calculations and decision making. Under the right sets of conditions, ML can be used to deliver valuable business insights more quickly and efficiently than many other analytics techniques because there's no need to program every possible scenario.
- **Power** to process and analyze large volumes of data. ML can use higher volumes of data than those traditional techniques, and it has the potential to perform much more powerful analytics.
- **Efficiency** to generate more models, more accurately, than traditional analytical and programming approaches. It offers efficiency in enabling models and insights to be generated without human resources or coding.
- **Intelligence** through the ability to learn autonomously and uncover latent insights.

Examples of How Machine Learning Can Deliver Value to Organizations

Many types of organizations have gained business value from machine learning. The types of ML that can be applied to create this value are as varied as the types of value they can generate. Listed below are some examples — but the possibilities are virtually limitless, if you have data that can be tapped into:

- **Customer Relationship Management:** A telecom organization looking to reduce call center and mailing costs is utilizing ML to predict and improve customer satisfaction by optimizing workforce management, and the organization is also using it to estimate the probability that customers will respond to new offers.
- **Failure Prediction for Preventive Maintenance:** One utility company applied ML to internal and external datasets in order to predict and respond to gas leaks in uncertain environmental

conditions. This analysis helped the organization better understand the flow of gas in its systems and where gas was either leaking or being stolen. This contributed to a reduction in lost and unaccounted-for gas.

- **Finance and Hedge Fund Portfolio Pricing:** One hedge fund reaped significant benefits by using ML to price financial portfolios in overly aggressive markets. By discovering latent features from the data in its portfolios, the firm was able to adjust pricing to maximize profit and boost revenue.
- **State Government and Litigation Case Management:** A prosecution office used a neural net to assess cases and determine the probability of a guilty verdict if the case went to trial. This enabled the district attorney to better predict which cases to pursue, not to pursue or offer to settle out of court.
- **Workforce Management:** An energy company is using ML to optimize the management and productivity of its work fleets, allowing the company to predict where its workers are likely to be most needed.

Machine Learning Can Also Provide Process Benefits for IT Organizations

> Artificial intelligence/ML and DevOps are mutually inclusive. DevOps can leverage ML, and ML can leverage the principles of DevOps.

In addition to uncovering new benefits for improve competitiveness, ML can also directly benefit IT professionals by helping to improve IT-related processes and functions. Gartner is seeing growth in using ML in applications for IT areas such as network threat monitoring, as well as data management and analysis. This can help IT make better decisions with regard to how to improve speed, power, efficiency and intelligence in the enterprise's networks and data platforms.

In data management, for example, ML could be applied to learn the most reliable sources of data given certain scenarios. In this sense, Gartner believes that ML is particularly valuable as "tradecraft" for digital architects, as well as other IT professionals focused on data management and analytics.

Other areas in which ML can benefit IT include:

- **Security Operations:** ML is being used extensively in threat management systems to predict or detect network intrusions.
- **IT Call Centers:** IT organizations are reducing or eliminating call centers by using ML to process and interpret incoming help desk calls and accurately route them to the right person for problem resolution within a single call.

- **DevOps:** IT organizations are using ML to optimize software deployment strategies — and to reduce the failure rate of new releases in software delivery life cycles — by analyzing log files and predicting server failures and outages. There are DevOps opportunities for developing custom ML and AI applications, as well as opportunities to leverage ML and AI as a tool for improving operational efficiencies within the IT organization.
- **Project Management:** Organizations are using decision trees and other ML models to optimize project schedules and to deliver cost estimates and rough orders of magnitude. Historic data describing project forecasts and actuals for time and effort can produce more accurate estimates, making it less likely that project managers will be pressured into unrealistic delivery commitments and thus saving the expense of remedial activity.

Business Strengths and Challenges of Machine Learning

To better understand how ML may benefit their own organization — and to weigh this against the potential costs and downsides of using it — technical professionals need to understand the major strengths and challenges of ML when applied to the business domain.

Strengths of ML in business include:

- **It offers speed, power, efficiency and intelligence.** ML can deliver valuable business insights more quickly and efficiently than traditional data analysis techniques because there's no need to program every possible scenario or require a human to be part of the process — taking people out of the process. Because ML can process higher volumes of data, it also has the potential to perform much more powerful analytics. ML's intelligence, provided by its ability to learn autonomously, can be used to uncover latent insights.
- **It is becoming more pervasive.** Due to higher volumes of data collected by increasingly ubiquitous computing devices and systems — and new ML-oriented analytic offerings and services taking off in the market from cloud data providers and vendors — ML can now be applied to a variety of data sources. It can also solve problems under a variety of contexts.
- **It can increase the capability to achieve business goals.** ML can be used to add unique functionalities to enterprise systems that may otherwise be too difficult to program. For example, ML is increasingly being used to solve large-scale process improvement initiatives that support business objectives. Many organizations are replacing programs, such as Six Sigma, with ML algorithms that learn to enhance processes. Programming for such a capability is not trivial.
- **It can handle nonspecific and unexpected situations.** When organizations are uncertain about the value or insights inherent in their data — or are confronted with new information they don't know how to interpret — ML can help discover business value where they may not have been able to before.

Challenges related to the use of ML in business include:

- **It requires considerable data and compute power.** Because ML applies analytics to such large amounts of data, and runs such sophisticated algorithms, it typically requires high levels of compute performance and advanced data management capabilities. Organizations will need to invest in infrastructure to handle it, or gain access to it through the on-demand services of external providers, such as big data analytics cloud providers.

- **It requires knowledgeable data science specialists or teams.** Typically, organizations that are successful with ML have one or more data scientists on staff who are knowledgeable about ML. However, this is not always a prerequisite. Technical professionals who understand ML basics may be able to buy algorithms from the market that will meet their needs without the aid of in-house data scientists. That course is more challenging, however.
- **It adds complexity to the organization's data integration strategy.** Machine learning feeds off of large amounts of raw data, which often come from various sources. This brings a demand for advanced data integration tools and infrastructure, which must be addressed in a thorough data integration strategy.
- **Learning ML algorithms is challenging without an advanced math background.** Many of the analytical models or algorithms involved are based on mathematical concepts, such as linear algebra and advanced statistical analysis. Technical professionals who can use off-the-shelf algorithms — or rely on the expertise of in-house data scientists — won't need this knowledge. However, if they to try modify existing algorithms or try to experiment with them on their own, it will be difficult without a mathematics background.
- **The context of data often changes.** For example, financial or demographic information used at the start of an ML analysis become less relevant before that analysis is complete, due to changes in external factors that have occurred in the meantime. Also, when data is pulled from one context and integrated with data from another, this may have ramifications on its accuracy or validity. Addressing these issues with advance data practices can be a challenge.
- **Algorithmic bias, privacy and ethical concerns may be overlooked.** It's easy to lose sight of privacy and ethical issues in our journey for exploration and discovery. If ML insights involve too much personal information, for example, this can raise moral or legal red flags. A rising concern with ML includes determining if the predictions or outputs are discriminating based on data. How do you prevent machine learners from discriminating based on private or sensitive information? Gartner recommends performing feature analysis on processed data as one way to combat privacy and ethical concerns, and to ensure features used in training or models do not contribute to discriminating outputs (see the Feature Analysis section). Another method for combating privacy and ethical concerns is to promote diversity in team composition. Diversity among data science teams help ensure that ML algorithms used to recognize objects, people or speech may offer greater inclusion of a multicultural environment.

How Should IT Prepare for Machine Learning?

The benefits that ML can provide will likely drive interest from business leaders. If the IT organization is proactive about planning and is preparing the IT environment for ML now, it will be better-positioned to deliver benefits. To be ready, technical professionals should start by planning in areas related to:

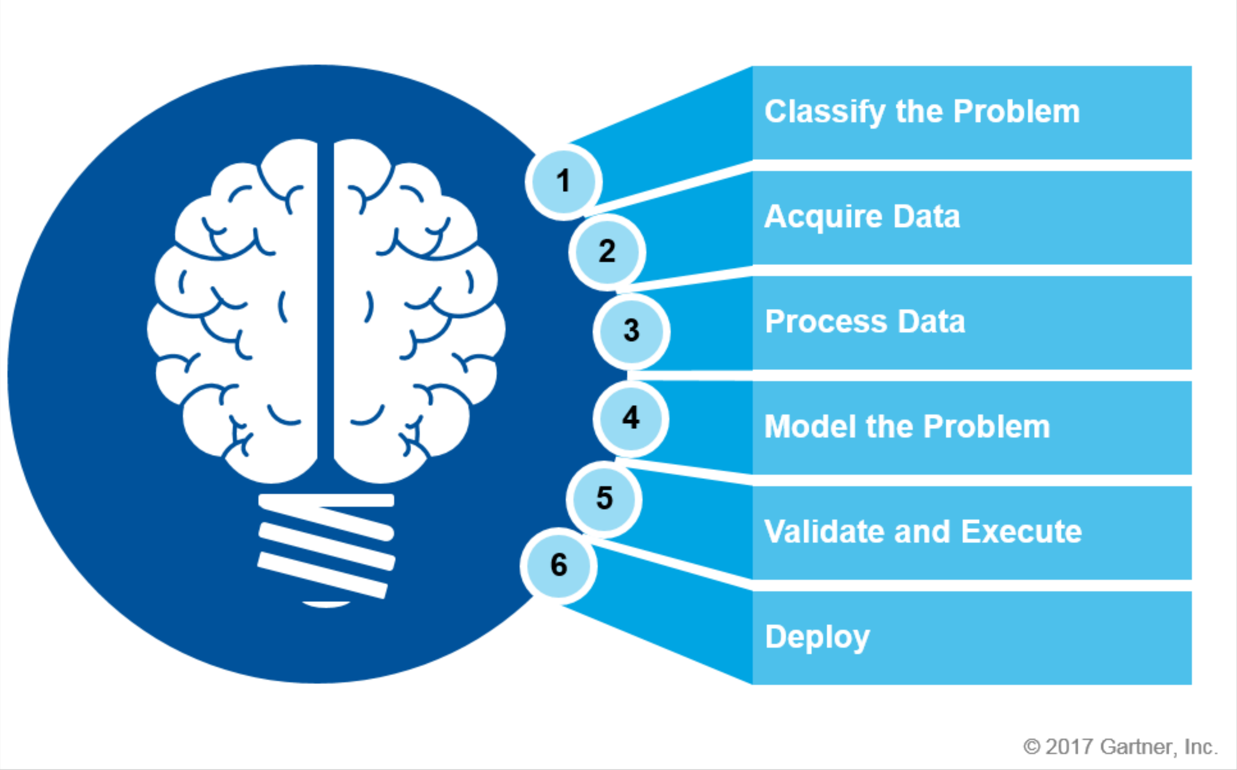
- The ML process

- ML technical architecture
- Required skills

Learn the Stages of the Machine Learning Process

The first key step in preparing to explore and exploit ML is to understand the basic stages involved (see Figure 4).

Figure 4. Stages of the Machine Learning Process

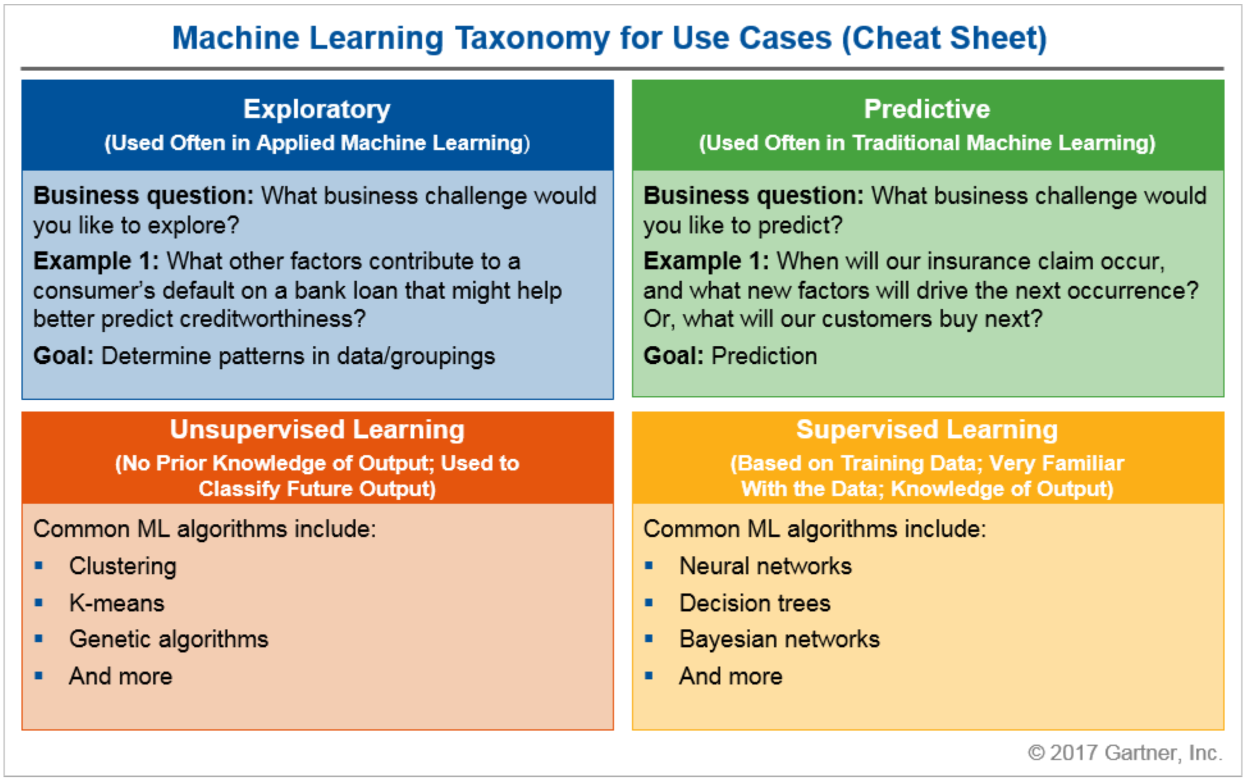


Source: Gartner (January 2017)

- **Classify the problem.** Build your problem taxonomy that describes how to classify the problem or business question to solve. The "cheat sheet" shown in Figure 5 provides a sample taxonomy for classifying problems or business challenges to be solved by ML. (Note: Figure 5 can be downloaded from the toolbar displayed in the left margin of this web page.)
- **Acquire data.** Identify where the data exists to support the problem you're trying to solve. Data used in ML can come from a variety of sources, such as ERP systems, IoT edge devices or mainframe data. The data used may be structured (such as NoSQL database records) or unstructured (such as emails).
- **Process data.** Identify how to prepare data for ML execution. Steps here include data transformation, normalization and cleansing, as well as the selection of training sets (for supervised learning).

- **Model the problem.** Determine the ML algorithms to be used for training or clustering. A range of algorithms can be acquired and extended to suit different purposes.
- **Validate and execute.** Validate results, determine the platform to execute models and algorithms, and then execute the ML routines. The execution process will likely comprise many cycles of running the ML routine and tuning and refining results.
- **Deploy.** Finally, the output of the ML process is deployed to provide some form of business value. This value may come in the form of data that will inform decisions, feed applications or systems, or be stored for future analysis. Depending on the type of ML routines executed, the output may also take the form of new models or routines that may supplement existing systems or applications (such as predictive models). Whatever the form of the results, this phase entails determining where and how to deploy them for consumption and decision making.

Figure 5. Machine Learning Guide



Machine Learning Taxonomy for Use Cases (Cheat Sheet)

Exploratory
(Used Often in Applied Machine Learning)

Business question: What business challenge would you like to explore?

Example 1: What other factors contribute to a consumer's default on a bank loan that might help better predict creditworthiness?

Goal: Determine patterns in data/groupings

Predictive
(Used Often in Traditional Machine Learning)

Business question: What business challenge would you like to predict?

Example 1: When will our insurance claim occur, and what new factors will drive the next occurrence? Or, what will our customers buy next?

Goal: Prediction

Unsupervised Learning
(No Prior Knowledge of Output; Used to Classify Future Output)

Common ML algorithms include:

- Clustering
- K-means
- Genetic algorithms
- And more

Supervised Learning
(Based on Training Data; Very Familiar With the Data; Knowledge of Output)

Common ML algorithms include:

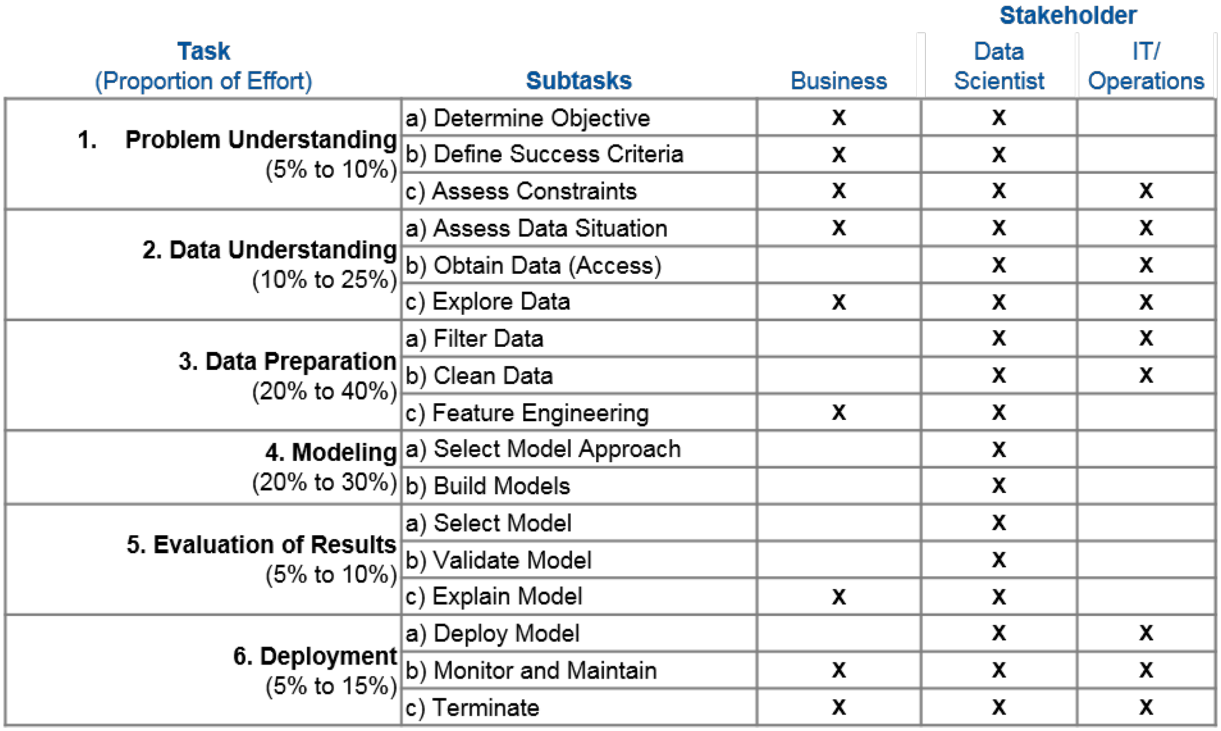
- Neural networks
- Decision trees
- Bayesian networks
- And more

© 2017 Gartner, Inc.

Source: Gartner (January 2017)

The stages of ML often overlaps with the data science life cycle. Figure 6 describes the data science life cycle, with a responsibility assignment matrix associated with it.

Figure 6. Data Science Life Cycle

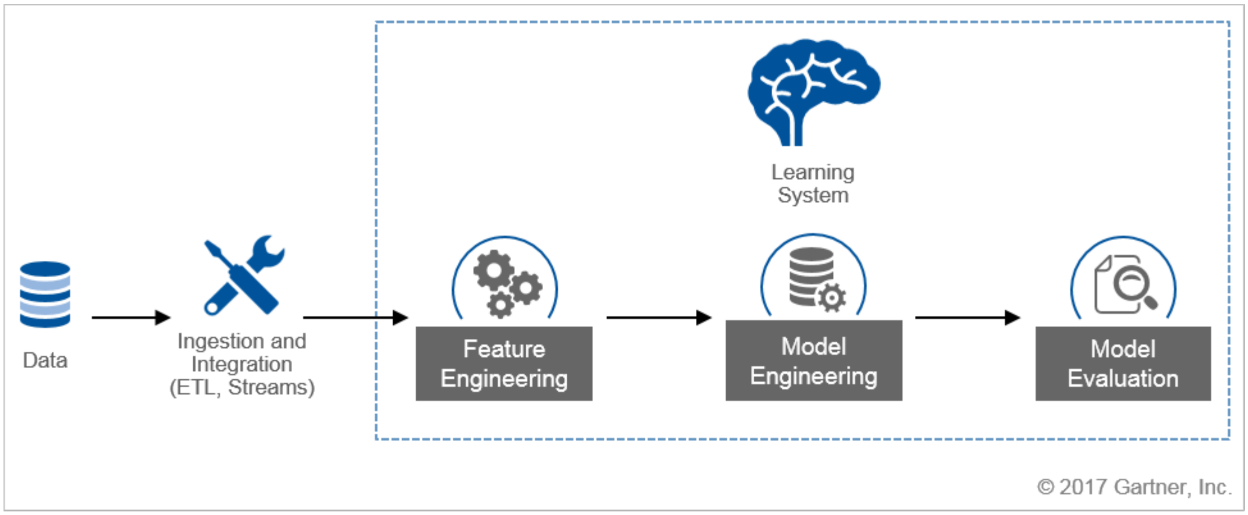


		Stakeholder		
Task (Proportion of Effort)	**Subtasks**	Business	Data Scientist	IT/ Operations
1. Problem Understanding (5% to 10%)	a) Determine Objective	x	x	
	b) Define Success Criteria	x	x	
	c) Assess Constraints	x	x	x
2. Data Understanding (10% to 25%)	a) Assess Data Situation	x	x	x
	b) Obtain Data (Access)		x	x
	c) Explore Data	x	x	x
3. Data Preparation (20% to 40%)	a) Filter Data		x	x
	b) Clean Data		x	x
	c) Feature Engineering	x	x	
4. Modeling (20% to 30%)	a) Select Model Approach		x	
	b) Build Models		x	
5. Evaluation of Results (5% to 10%)	a) Select Model		x	
	b) Validate Model		x	
	c) Explain Model	x	x	
6. Deployment (5% to 15%)	a) Deploy Model		x	x
	b) Monitor and Maintain	x	x	x
	c) Terminate	x	x	x

Source: Gartner (January 2017)

Building and operating an end-to-end machine learning system requires stakeholders made up of business subject matter experts, data scientists and IT operations personnel (see Figure 6). However, data preparation, modeling and evaluation often strain the data scientist, typically resulting in a slow, inconsistent and largely manual process for developing ML models. ML uses complex models and algorithms to aid in predictions and simulations. An expansion of the existing data life cycle is needed to improve efficiencies related to developing ML models needed for machine learning. Model development differs from traditional software development because of the requirements to monitor and tune ML models in short iterations. A typical life cycle for developing learning systems is summarized in Figure 7.

Figure 7. Life Cycle for Developing Learning Systems

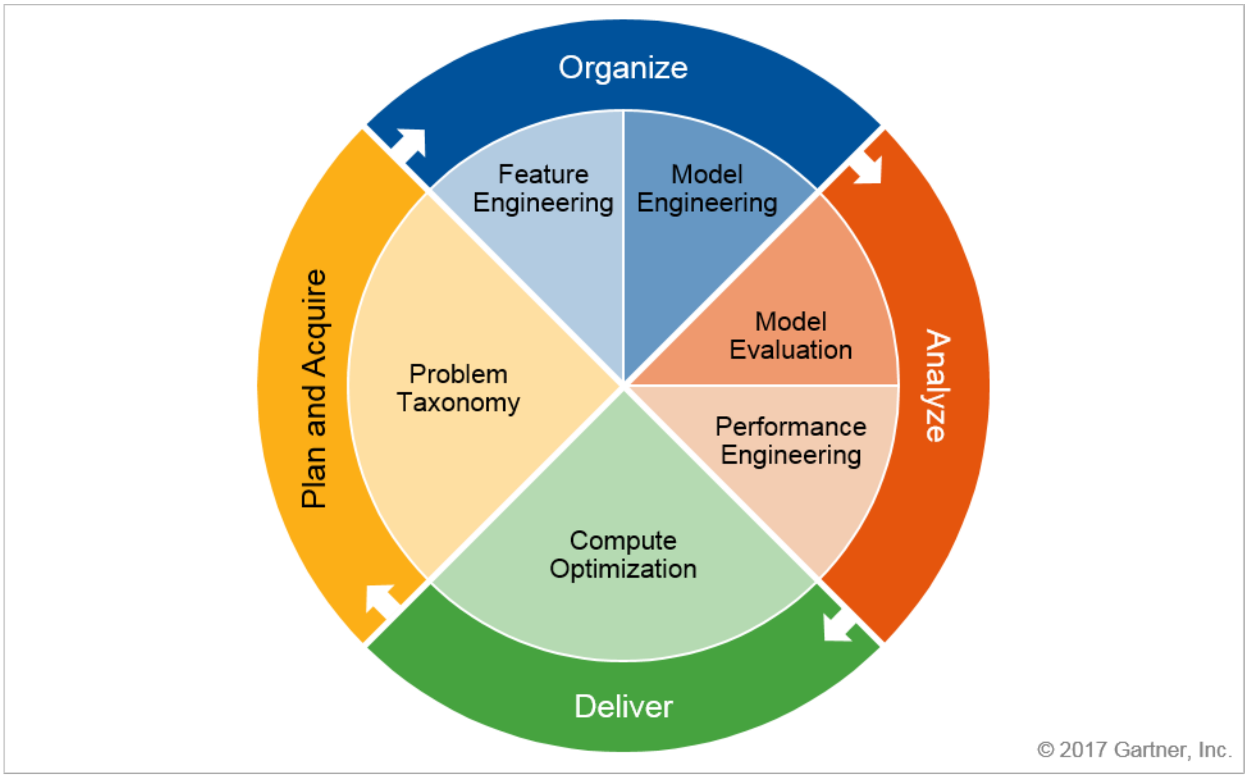


Source: Gartner (January 2017)

Understand the Model Development Life Cycle Needed for Machine Learning

When planning to aggressively build custom ML algorithms and applications, organizations must develop a life cycle for machine learning to support the highly iterative building, testing and deployment of ML models. The process for planning, creating, testing and deploying ML systems is similar to any other application development life cycle. However, a slightly adapted life cycle is needed in order to focus more on ML model evaluation and tuning. Figure 8 describes the adapted model development life cycle needed for ML. The following ML modeling process will guide technical professionals in implementing a continuous model deployment and control framework for automating the process of developing, testing, deploying and monitoring ML models.

Figure 8. Continuous ML Model and Control Framework



Source: Gartner (January 2017)

The adapted life cycle offers the same tasks as traditional data and analytic services, with the addition of subtasks that enable ML capabilities. Gartner recommends incorporating these subtasks into your data and analytics programs. A description of the updated processes of Figure 8 is described in the Understanding the Basic Architecture Needed for Machine Learning section of this document.

Development life cycles must support:

- Collaboration for heterogeneous teams and technologies
- Monitoring of ML models with statistical analysis capabilities
- Reusability of ML models for rapid development
- Interoperability between different analytic platforms and ML frameworks

Understand the Basic Architecture Needed for Machine Learning

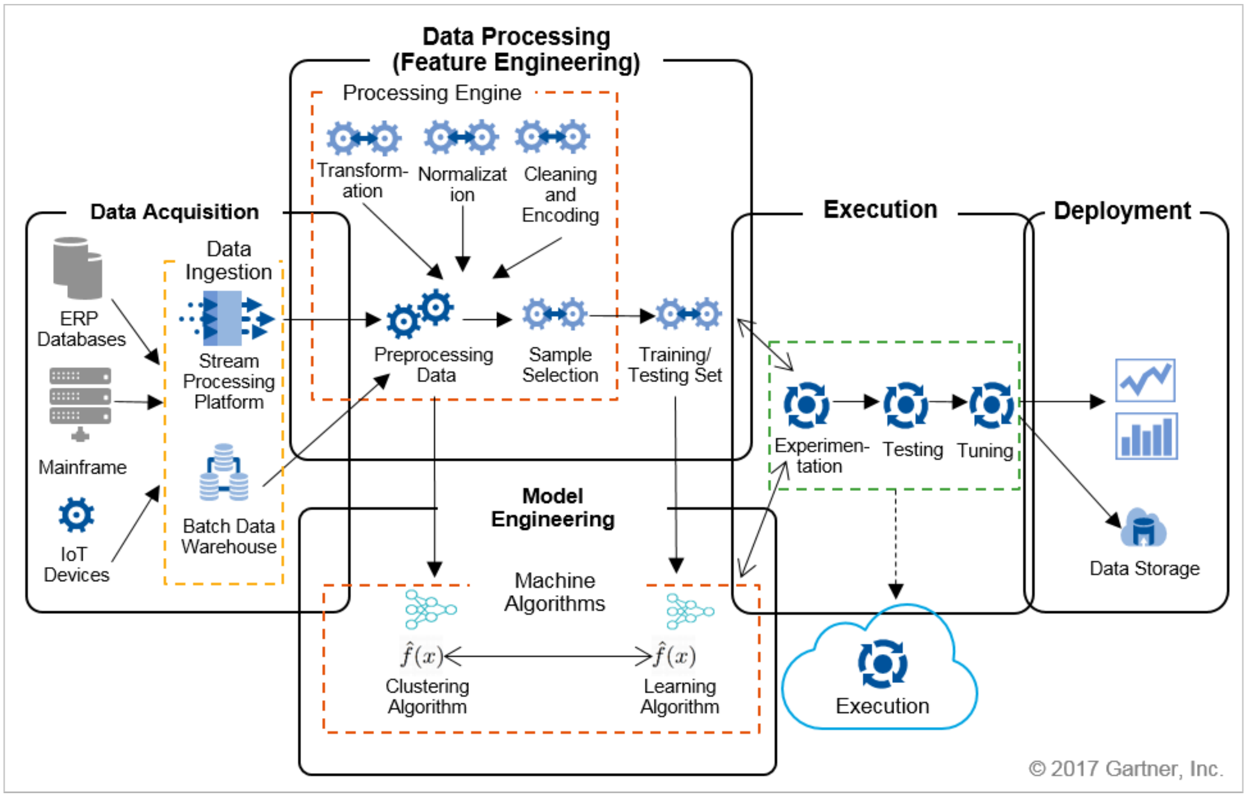
The stages discussed in Figure 4 map to the basic architecture that will be needed to execute ML in the enterprise. This architecture differs in many ways from the architectures used for traditional data processing and analytics functions in enterprises. ML architecture needs to be more flexible to accommodate the elastic learning patterns of the ML process and the large and varying volumes of data and processing power involved.

Many of the underlying infrastructure elements used are highly likely to be cloud-based ones. The cloud is an excellent match for most ML applications due to the elasticity it offers to scale processing and handle high data volumes as needed.

Figure 9 shows Gartner's suggested reference architecture for ML. It covers the following infrastructure areas for functions needed to execute the ML process:

- **Data acquisition,** where data is collected, prepared and forwarded for processing
- **Data processing,** where steps such as preprocessing, sample selection and the training of datasets take place, in preparation for execution of the ML routines:
    - **Feature analysis or feature engineering** (a subset of the data processing component), where features that describe the structures inherent in your data are analyzed and selected
- **Data modeling or model engineering,** which includes the data model designs and machine algorithms used in ML data processing (including clustering and training algorithms):
    - Model fitting, where a set of training data is assigned to a model in order to make reliable predictions on new or untrained data
    - Model evaluation, where models are evaluated based on performance and efficacy
- **Execution,** the environment where the processed and trained data is forwarded for use in the execution of ML routines (such as experimentation, testing and tuning)
- **Deployment,** where business-usable results of the ML process — such as models or insights — are deployed to enterprise applications, systems or data stores (for example, for reporting)

Figure 9. Machine Learning Architecture



Source: Gartner (January 2017)

Note that these portions of the architecture map to many of the ML phases discussed in the previous section — for example, acquiring, processing and modeling data, and then executing ML routines and deploying the results.

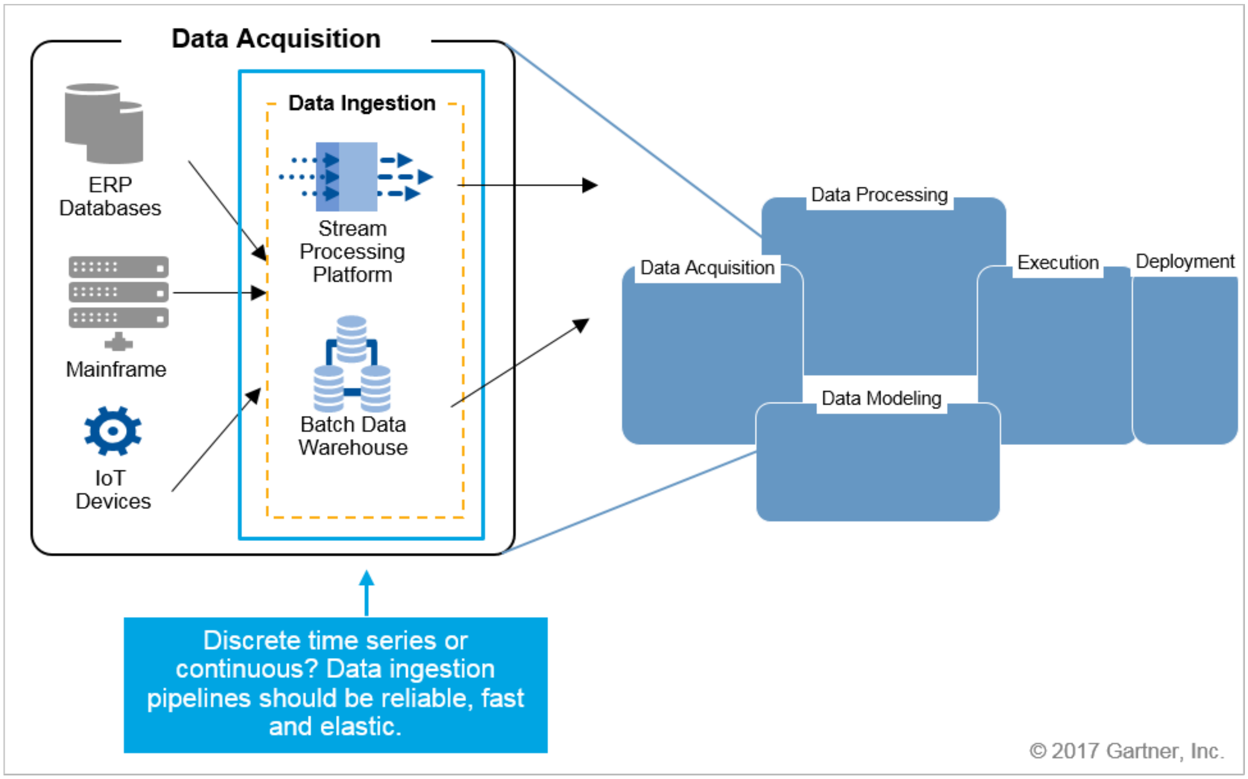
A full-blown enterprise ML architecture, containing all the features above, probably won't be necessary when just starting out with ML. Instead, IT organizations are likely to build up to this architecture as they gain more experience with, and exploit more uses cases for, ML over time. Early on, practitioners may purchase a small-scale ML platform to suit a specific use case, and purchase off-the-shelf tools to support a smaller-scale, "ML Lite" (lightweight) architecture suited for early use cases. Over time, they can iteratively build, scale and unify this into an "ML Enterprise" architecture that can efficiently support multiple use cases and more-mature ML efforts.

The following sections explore the different major components of this architecture in more detail.

Data Acquisition

In the data acquisition component of the ML architecture, data is collected from a variety of sources and prepared for ingestion for ML data processing platform (see Figure 10). This component of the architecture is important because ML often begins with the collection of high volumes of data from a variety of potential sources, such as ERP databases, mainframes or instrumented devices that are part of an IoT system. This portion of the architecture contains the elements needed to ensure that the ingestion of ML data is reliable, fast and elastic.

Figure 10. ML Architecture: Data Acquisition



Source: Gartner (January 2017)

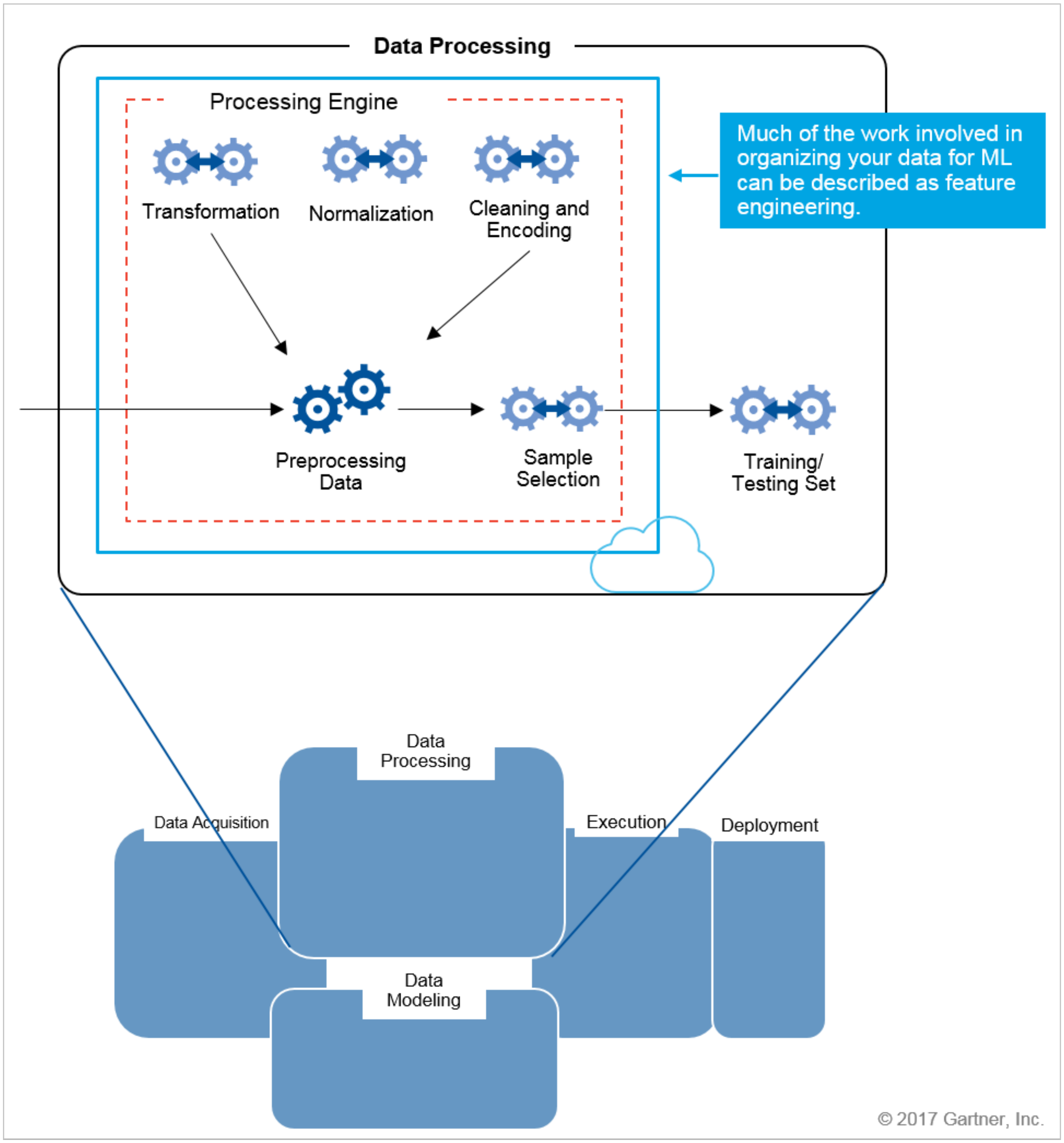
How this data is handled prior to ingestion depends on whether data is coming in discrete chunks or a continuous flow. Discrete data may be stored and forwarded via a batch data warehouse. If continuously streaming data is used — especially if large, erratic streams of data are feeding the ML process (from IoT systems, for example) — a stream processing platform may be needed here. This stream-processing capability may be needed to screen out data not needed for processing, to store some in the data warehouse for future reporting, or to pass a portion along if it's needed for immediate processing. Many cloud platform providers offer stream processing engines that perform this type of in-stream analytics.

Gartner tip: Look for tools that support batch and real-time data ingestion strategies in order to leverage data in motion for ML processing.

Data Processing/Integration

The data processing portion of the architecture is where ingested data is forwarded for the advance integration and processing steps needed to prepare the data for ML execution (see Figure 11). This may include modules to perform any upfront data transformation, normalization, cleaning and encoding steps that are necessary. In addition, if supervised learning is being used, data will need to have sample selection steps performed to prepare sets of data for training.

Figure 11. ML Architecture: Data Processing



Source: Gartner (January 2017)

There are several key considerations that may influence choices in this portion of the architecture. For example, will the data be processed in transit or at rest? If continuous processing or high-throughput computing is needed here, you may choose to implement a Lambda architecture.[1] Additionally, you may choose to use in-memory processing for high-speed processing. Other choices for integrating data in this layer may include whether to use a stand-alone data integration application or an integration platform as a service (iPaaS) offering (such as Dell Boomi).

Feature Analysis

Much of the data ingested for processing may include features (aka variables) that are redundant or irrelevant. Therefore, technical professionals must enable the ability to select and analyze a subset of the data in order to reduce training time, or to simplify the model. In many cases, feature analysis is a part of sample selection. However, it is important to highlight this subcomponent in order to filter data that may violate privacy conditions or promote unethical predictions. To combat privacy and ethical concerns, users should focus on removing features from being used in the model.

Note that it is a good principle to extract as much data as possible from sources when they are available. This is because it is difficult to predict which data fields are useful. Obtaining copies of production data sources can be difficult and subject to stringent change control. Therefore, it is better to obtain a superset of the data that is available and then restrict the data actually used in the model through the use of filtering or database views. If during development it becomes clear that further data fields are needed, then it is possible to simply relax the filtering or view criteria, and the extra data is immediately available. Storage is inexpensive, and this makes the process much more agile.

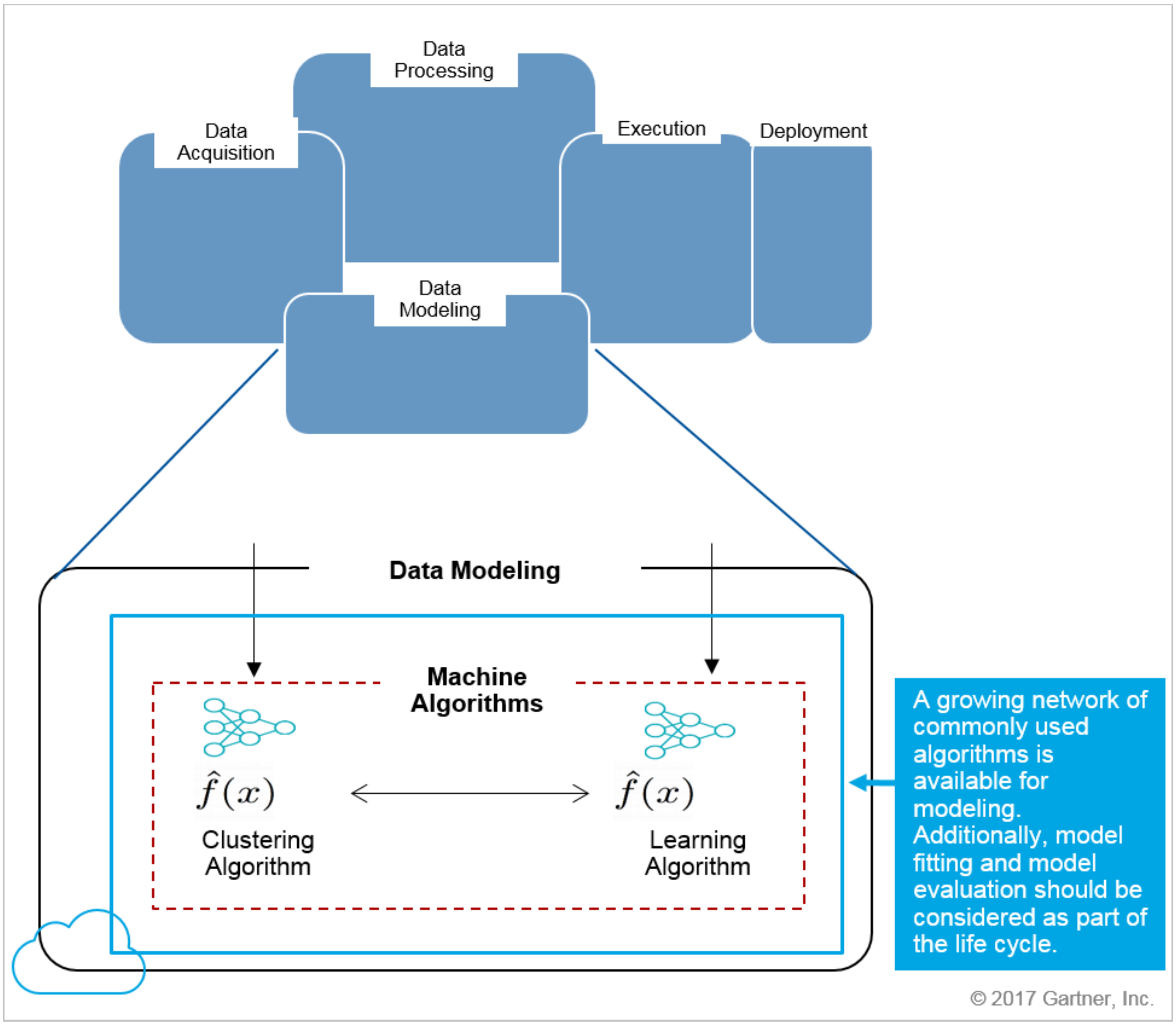
Self-service data preparation tools are often used to perform feature analysis and selection. (See "Embrace Self-Service Data Preparation Tools for Agility, but Govern to Avoid Data Chaos.")

Gartner tip: Look for tools that support self-service data preparation in order to provide data science teams and developers with the ability to manipulate data to support ML algorithms or models. Governance will play a major role in the component of your architecture. Additionally, consider securing the learning and classification parts of your architecture to ensure privacy or ethical considerations aren't breached through adversarial ML.

Data Modeling

The modeling portion of the architecture is where algorithms are selected and adapted to address the problem that will be examined in the execution phase (see Figure 12). For example, if the learning application will involve cluster analysis, data clustering algorithms will be part of the ML data model used here. If the learning to be performed is supervised, data training algorithms will be involved as well.

Figure 12. ML Architecture: Data Modeling



A growing network of commonly used algorithms is available for modeling. Additionally, model fitting and model evaluation should be considered as part of the life cycle.

Source: Gartner (January 2017)

Note that algorithms do not need to be built from scratch by your data science team. Many useful libraries of extensible algorithms are available in the market, and they can be extended and adapted for your own use. When starting out with ML, experience can be gained by obtaining a few common algorithms — either supervised or unsupervised — from the marketplace and deploying them in the cloud with some data to perform experiments. These may uncover promising avenues for future business value, and eventually be expanded into formal ML deployments.

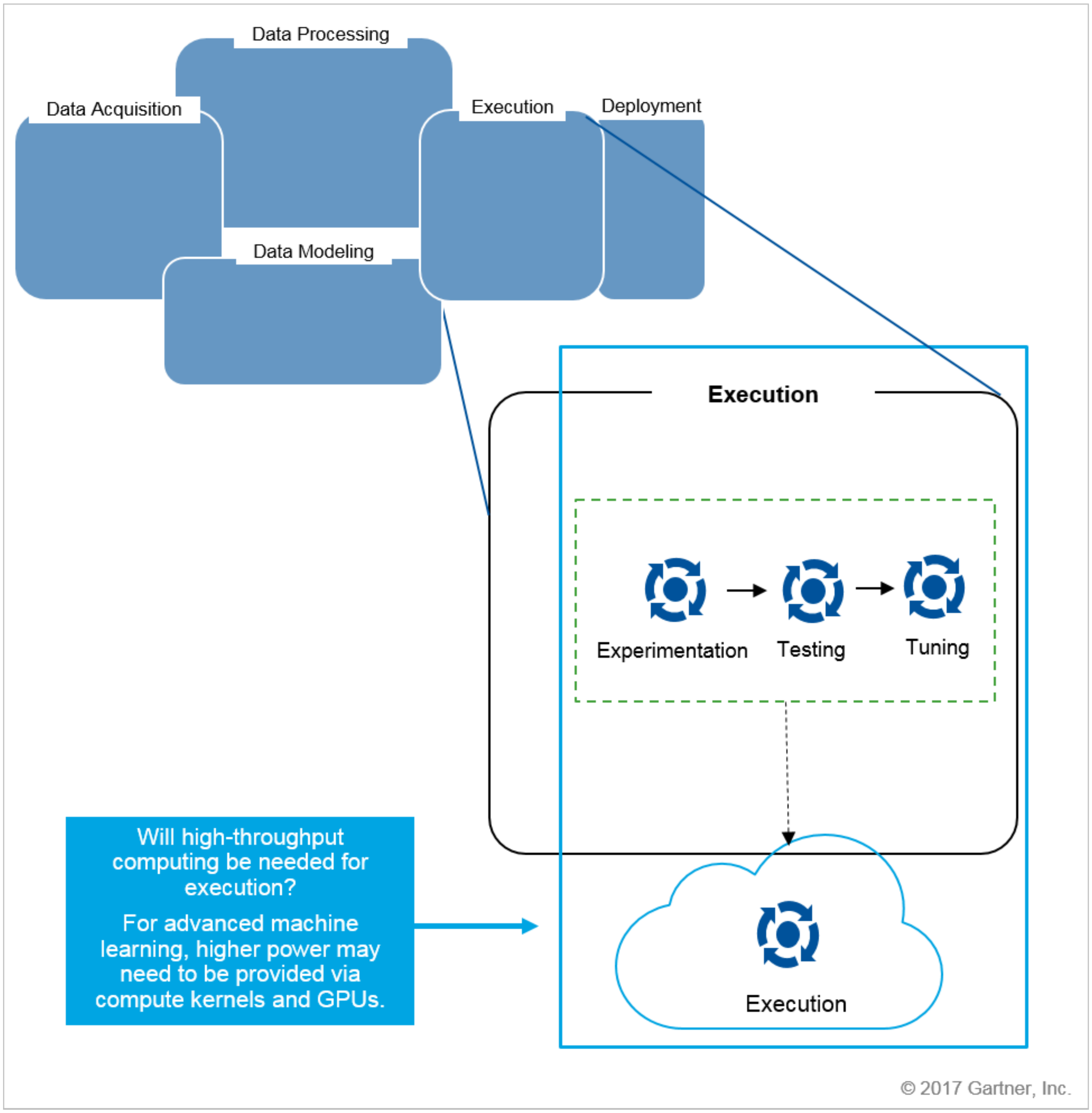
Gartner tip: Consider ML toolkits instead of developing algorithms from scratch. Additionally, consider key-value databases to store metadata associated with the ML models. For example, a key-value store may be used to store semantic or context-dependent information about the ML models or algorithms.

Execution

> ML algorithms can be highly nondeterministic, and they may yield unexpected behaviors, depending on training and data preparation. Technical professionals must design for nondeterminism by enabling elastic compute environments. Consider the public cloud as one of those elastic environments.

Once the data is prepared and algorithms have been modeled to solve a specific business problem, the stage is set for ML routines to be run on the execution portion of the architecture. The ML routine will execute repeatedly — as cycles of experimentation, testing and tuning are performed to optimize the performance of the algorithms and refine the results — in preparation for the deployment of those results for consumption or decision making (see Figure 13).

Figure 13. ML Architecture: Execution


Data Processing
Data Acquisition
Execution
Deployment
Data Modeling
Execution
Experimentation
Testing
Tuning
Will high-throughput computing be needed for execution?
For advanced machine learning, higher power may need to be provided via compute kernels and GPUs.
Execution
© 2017 Gartner, Inc.

Source: Gartner (January 2017)

One key consideration in this area is the amount of processing power that will be needed to effectively execute ML routines — whether that infrastructure is hosted on-premises or obtained as a service from a cloud provider. Depending on how advanced the ML routines are, the performance needed here may be significant. For example, for a relatively simple neural net with only four or five inputs (or "features") in it, the processing could be handled using a regular CPU on a desktop server or laptop computer. However, a net that has numerous features — designed to perform advanced, "deep learning" routines — will likely need high-throughput computing power on the execution platform in the form of high-performance computing (HPC) clusters, or compute kernels executing on high-powered graphics processing units (GPUs). Nvidia products are an example of an infrastructure designed for efficiently executing ML algorithms using GPUs. Nvidia's parallel computing platform, together with its Cuda application programming interface (API), is arguably the leader in platforms that support the efficient execution of machine learning algorithms, given that it provides the muscle to scale architectures to support ML. In addition, cloud providers are now making available machines with GPUs, so it is not necessary to have your own on-premises system.

ML algorithms can be highly nondeterministic, meaning the algorithms may execute differently and yield different performance results and output depending on unexpected variable behavior, streaming data pipelines and feature evaluation. An important consideration here is to examine infrastructures that scale automatically to avoid disruption or excessive latency due to throttling techniques. A suitable environment for running ML is in the cloud. Cloud environments are highly elastic, which may save on the overhead of engineering solutions on-premises.

Data science teams will often look to test and debug their ML models or algorithms prior to deployment. Testing and debugging ML is slightly different compared with the traditional methods of testing and debugging software applications. This is because ML developers often experiment with data that is different from live data in order to operationalize models earlier in the life cycle. In addition, testing ML is typically multidimensional — that is, developers must test for data, test for proper model fit, and test for proper execution. This can be nontrivial. To combat this challenge, Gartner recommends designing testing environments that mimic production as closely as possible.

> Testing and debugging ML focuses on three components of the architecture: testing for data, appropriate fit of the ML model and proper execution.

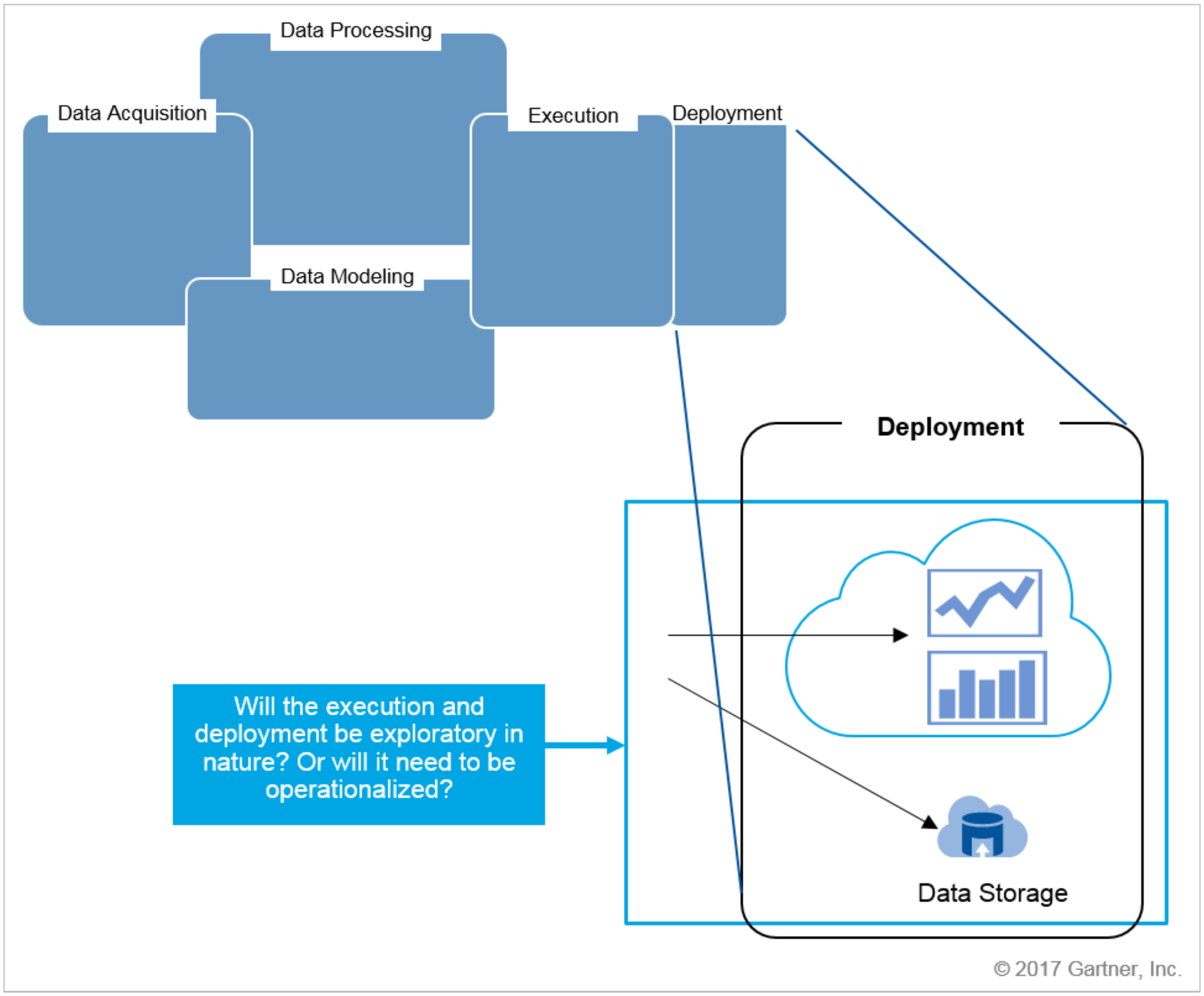
Gartner tip: Consider tools that offer monitoring and execution of ML experiments, collaboration, and code reuse. It is essential to view the performance of different ML experiments toward optimization.

Deployment

ML output is similar to any other software application output, and it can be persisted to storage, file, memory or application or looped back to the processing component to be reprocessed. In many cases, ML output is persisted to dashboards that alert a decision maker of a recommended course of action. When operationalizing ML programs, note that the learner becomes an analytics program, similar to any other analytic program you might run in production. In production, the machine learning system becomes an advanced nondeterministic query that relies on compute power for execution.

Understand that the deployment of the resulting information, tools or new functionality generated by the machine learning routine will vary depending on what type of ML is being used, and what value it is intended to generate. Deployed outputs could take the form of reported insights, new models to supplement data analytics applications, or information to be stored or fed into other systems (see Figure 14).

Figure 14. ML Architecture: Deployment



Source: Gartner (January 2017)

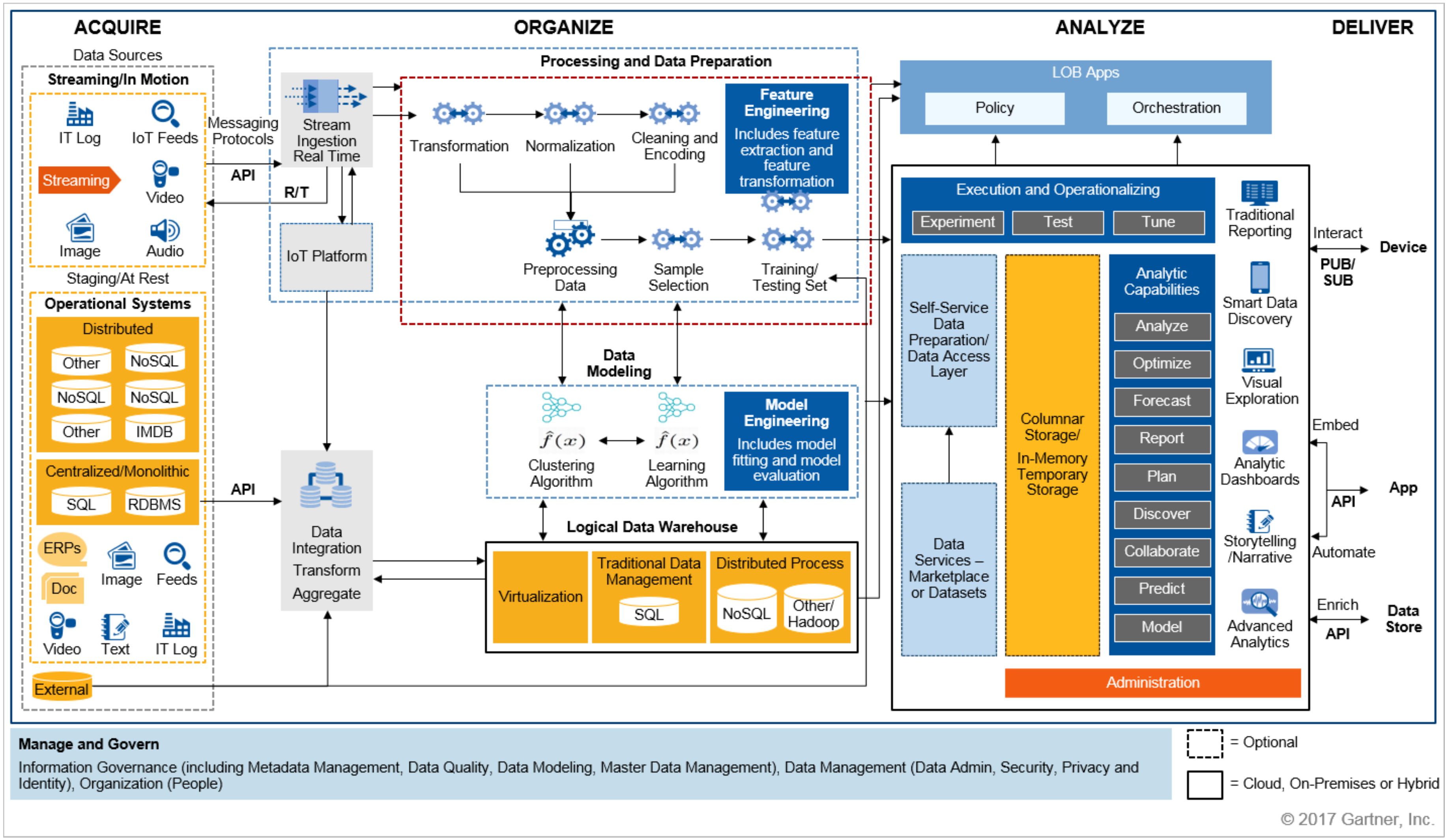
An important consideration here is whether this portion of the architecture will need to be operationalized. This is unlikely to be the case if the ML routine is exploratory in nature because the nature of the end results — and how they are deployed — is less likely to be able to be planned or predicted in advance. However, for nonexploratory ML routines, operationalization of the execution and deployment phases may need to be planned for in the architecture.

Gartner tip: Develop a process that seamlessly moves ML experiments into production. This can be done through traditional application methods or through COTS software products. Traditionally, a challenge in deployment has been that the languages needed to operationalize models have been different from those that have been used to develop them. Keep that in mind as you procure COTS to operationalize ML programs.

A Comprehensive End-to-End Architecture

To support ML applications, technical professionals must envision a revitalized data and analytics end-to-end architecture that incorporates diverse data, models and algorithms and can deliver analytics anywhere (see "2017 Planning Guide for Data and Analytics"). Figure 15 shows Gartner's four stage architecture that includes ML capabilities.

Figure 15. End-to-End ML and Analytics Architecture



LOB = line of business
pub/sub = publication and subscribe

Source: Gartner (January 2017)

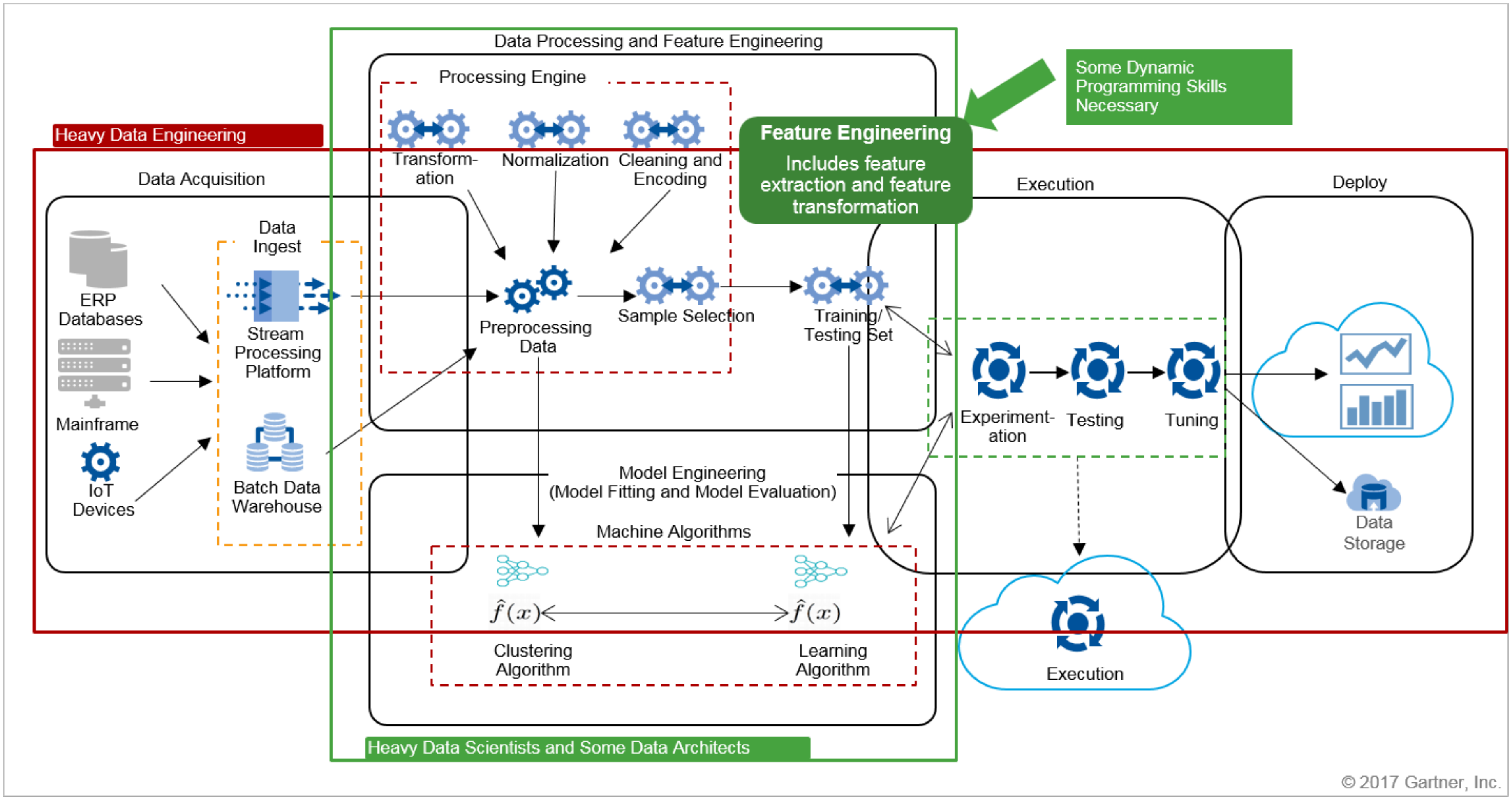
Understand What Skills Will Be Needed for Machine Learning

The successful deployment of ML initiatives will require specialized skills, some of which will likely already be present in the organization and many of which may not. Technical professionals interested in pursuing ML will want to hone their skills in the areas that will be in demand for these initiatives. Further, those interested in promoting and advocating the pursuit of ML initiatives within their organizations may want to influence their IT leaders to invest in building such skills.

If the organization already has a data science team, many of the skills it is likely to possess will help. However, it may need to be enhanced with even more specialized data science skills specific to ML, such as feature engineering and feature extraction. Data architects and data engineering skills will also be needed. Where these skills will be needed will vary in different areas of the ML architecture (see Figure 16). For example:

- **Heavy data engineering skills** will likely be needed for the processes moving horizontally across the ML architecture diagram — that is, from data acquisition through execution and deployment. In these areas, data engineers — along with system engineers — will be needed to ensure successful results in areas such as ingesting data, understanding throughput and payload, and optimizing execution performance:
    - Dynamic programming skills will likely be needed for building applications that include ML. Common dynamic programming languages include Python, R, Java and matrix laboratory (MATLAB). Gartner recommends building in this area to support operationalizing ML applications.
- **Heavy data science and some data architecture skills** will likely be needed in the center of the process — for the data processing and data modeling phases that precede the execution of ML routines. In these areas, specialized coding will be needed, and feature engineering will be important. Here, data scientists will need a data processing integration architecture and platform to process data and deploy their models. Data architects will play a role in these areas as well.

Figure 16. The Skills Needed for Machine Learning Will Vary by Area



Source: Gartner (January 2017)

Steps to Get Started With Machine Learning

Because of the potential business value offered by ML, business leaders in your organization may soon call on the IT organization to support ML initiatives. Will yours be ready? Rather than waiting until the business demands action to begin taking steps to support ML initiatives, technical professionals should get out in front of this trend now. They should start by investigating the technology, identifying value opportunities and trying to launch their first ML solutions to gain experience and demonstrate value.

The steps outlined in the sections below compose an action plan for technical professionals who want to push ahead with ML efforts and get the ball rolling within their own organizations.

Learn About and Experiment With ML Concepts and Technology

Contrary to popular belief, you don't need a Ph.D. to get started developing with ML. It helps to have a background in qualitative and quantitative analysis (often learned in the Ph.D. process) in order to understand how to examine the impact of independent and dependent variables. However, that is not a requirement. The first important step is to learn as much as possible about ML technology, and begin experimenting with the technology to gain your first experience about how ML solutions operate. Recommended learning and experimentation steps include:

- **Participate in online courses.** An abundance of online training is available. Good places to start include the "Machine Learning" course offered by ML pioneer Andrew Ng, as well as "Intro to Machine Learning" offered by Udacity, an online university.
- **Pick a simple algorithm to study.** Your goal is to get a brief understanding of what an algorithm looks like. Many toolkits are available for users to experiment with. Scikit-learn, for example, is an excellent source for understanding ML models and algorithms using dynamic programming languages, such as Python.
- **Experiment with ML technology in the cloud.** Try conducting an experiment in the cloud now by using a cloud-based offering such as Amazon Machine Learning or Microsoft Azure Machine Learning. To learn more about how ML algorithms behave and execute at runtime, monitor these experimental projects with a service such as Amazon CloudWatch.

Work Closely With Data Science Teams and Business Users to Identify a Use Case

Once you've gained some knowledge of the basics, the next step is to identify a good business challenge in your organization that could be solved using ML; kicking off an initial small project will help you to gain experience and demonstrate value. If your organization has a data science team, contacting this group is a great place to start. Find out how much the team knows about ML, how much it is interested in pursuing it, and whether it can help you to identify any good use cases in the business for the first ML solution you build. Consider providing these data scientists with a few ML algorithms they can play with and extend, to gauge their interest and engage them in the technology. They can contribute their ML knowledge, and you can help them get the data they need. You can also check their assumptions about the quality and suitability of that data. Success in this area is a team sport.

Beyond data science teams, you should also investigate whether there are areas of your business organization where ML may offer benefits, and whether leaders of those units may be interested in teaming up with you on pilot projects. If there are any business organizations with which the IT organization has collaborated particularly well in the past — or for which tools or analysis built from ML routines could offer some promising potential — those business groups are promising starting points for your outreach efforts.

Build a Use Case in the Cloud

Once you've identified a good first business challenge to solve with ML, build a small pilot project and conceptual architecture around that use case. Gartner recommends using cloud technologies for the components of that architecture because the cloud offers the flexibility to scale your efforts and the elasticity suitable for ML's often variable data streams and volumes.

Pointers for constructing the initial solution architecture include:

- Invest in a small ML platform first, based on and designed to support your initial use case, by using off-the-shelf tools that support basic ML (that is, start with an "ML Lite" approach). When moving to the cloud, be mindful of the cost models associated with the investment. ML can grow to rapidly absorb compute power, and this is something that might get expensive fast. Look for cost models that separate storage from compute power.
- Make data acquisition a priority. Consider where you will obtain the data and how it will be collected. Build for flexible streams of data (real time and batch), and consider whether ML gateways will be needed to handle the data ingestion.
- Plan to devote significant upfront work to the steps needed to prepare the data for consumption in the data processing phase. Focus on data quality, and remember that you may not need all of the data collected.

Iteratively Expand Your ML Platform and Services Over Time

Avoid attempting to deploy too much technology, or handle too many ML services, too quickly. Instead, after you gain experience and incremental success by executing a few small-scale use cases, you can build out your experience and architecture iteratively, over time.

As you build, position ML as a service offered by the IT organization. As each use case adds to your experience — and to your learners' intelligence — you can build out your taxonomy of ML services. The more ML solutions and services you gain experience with, the more competitive advantage your organization can gain.

In conjunction with expanding services, iteratively evolve and expand your ML platform as well. Initial, smaller, "ML Lite" platforms built around your first use cases can by unified into a larger, more sophisticated "ML Enterprise" platform that can support multiple use cases across your organization.

Recommendations

Gartner recommends that technical professionals take the following steps to prepare and initiate ML capabilities:

- **Build a taxonomy for classifying the problems or challenges to be solved by ML.** The cheat sheet shown in Figure 5 offers a good template for starting this technique. ML algorithms can be overwhelming because there are many to choose from. Organizations often spend too much time debugging models that don't fit the data, business problem or challenge they are trying to address. Start by categorizing to help reduce capabilities and to avoid overwhelming users.
- **Evaluate self-service platforms that support data preparation and applied machine learning**. For example, C3 IoT (formerly C3 Energy) offers a platform product for self-service ML called C3 Ex Machina. This tool provides a designer interface that aids developers in building ML applications. C3 IoT also offers a significant ML toolkit for data science teams to explore.
  - Note: There are a variety of ML platforms that support proprietary deep learning frameworks, but don't support common frameworks offered by the open-source community (such as Google TensorFlow, Caffe, Torch, Deeplearning4j and so on). Gartner recommends evaluating self-service ML platforms against their capability to interoperate with multiple deep learning frameworks.
- **Offer ML as a toolkit to data scientists rather than allowing them to build their own customized algorithms.** There are extensive toolkits available, and they will likely support your use case or business challenge. Developing customized algorithms can be a nontrivial undertaking and can expand your architecture with unconventional integration to third-party tools. Gartner recommends offering toolkits to be exploited by data science teams to avoid potential integration challenges.
- **Use the public cloud to start your initiative because it can elastically scale to accommodate any requirement.** Amazon, Microsoft, IBM, Google and many other cloud providers offer ML capabilities that can be leveraged to achieve self-service capabilities. However, Gartner recommends exploring the capability to interoperate with multiple ML frameworks and toolkits in order to design an open architecture.

Conclusion

Machine learning is about acquiring knowledge through data, and it differs from traditional applications or programs that generate statistics or engineering output. ML technologies offer the benefits of speed, power, efficiency and intelligence through learning without having to explicitly program these characteristics into an application. In other words, ML enables us to program *how* we make decisions instead of programming those decision. This offers many opportunities for developers and data science teams to enhance product offerings, customer relationships, marketing and advertising, process improvement, and much more.

ML is the next generation of analytic tradecraft for digital business architects. These architects should build for ML by understanding the business problem to be solved, acquiring data, processing data, modeling, executing and deploying. They should start with a business challenge, build sample programs, and execute use cases iteratively. They should also use the cloud because of its elasticity. Infrastructures that support ML must be elastic to account for the variations in data throughput and context. Public cloud providers offer suitable ML toolkits and compute instances to support self-service ML, if required.

Gartner Recommended Reading

Some documents may not be available as part of your current Gartner subscription.

"2017 Planning Guide for Data and Analytics"

"Doing Machine Learning Without Hiring Data Scientists"

"Machine Learning Drives Digital Business"

"Embrace Self-Service Data Preparation Tools for Agility, but Govern to Avoid Data Chaos"

Evidence

[1] The Lambda architecture was defined by Nathan Mars in the book "Big Data: Principles and Best Practices of Scalable Real-Time Data Systems" and consists of a batch subsystem holding raw or relatively unchanged data and a real-time subsystem for streaming or real-time query. These combine to support any requirement for query and analysis, no matter what the latency required.

GARTNER HEADQUARTERS

Corporate Headquarters
56 Top Gallant Road
Stamford, CT 06902-7700
USA
+1 203 964 0096

Regional Headquarters
AUSTRALIA
BRAZIL
JAPAN
UNITED KINGDOM

For a complete list of worldwide locations,
visit http://www.gartner.com/technology/about.jsp