

Operations Practice

Operationalizing machine learning in processes

Machine learning shows tremendous potential for increasing process efficiency. But generating real, lasting value requires more than just the best algorithms.

by Rohit Panikkar, Tamim Saleh, Maxime Szybowski, and Rob Whiteman



As organizations look to modernize and optimize processes, machine learning (ML) is an increasingly powerful tool to drive automation. Unlike basic, rule-based automation—which is typically used for standardized, predictable processes—ML can handle more complex processes and learn over time, leading to greater improvements in accuracy and efficiency.

But a lot of companies are stuck in the pilot stage; they may have developed a few discrete use cases, but they struggle to apply ML more broadly or take advantage of its most advanced forms. A recent McKinsey Global Survey, for example, found that only about 15 percent of respondents have successfully scaled automation across multiple parts of the business. And only 36 percent of respondents said that ML algorithms had been deployed beyond the pilot stage.

A central challenge is that institutional knowledge about a given process is rarely codified in full, and many decisions are not easily distilled into simple rule sets. In addition, many sources of information critical to scaling ML are either too high-

level or too technical to be actionable (see sidebar “A glossary of machine-learning terminology”). This leaves leaders with little guidance on how to steer teams through the adoption of ML algorithms.

The value at stake is significant. By building ML into processes, leading organizations are increasing process efficiency by 30 percent or more while also increasing revenues by 5 to 10 percent. At one healthcare company, a predictive model classifying claims across different risk classes increased the number of claims paid automatically by 30 percent, decreasing manual effort by one-quarter. In addition, organizations can develop scalable and resilient processes that will unlock value for years to come.

Four steps to turn ML into impact

ML technology and relevant use cases are evolving quickly, and leaders can become overwhelmed by the pace of change. To cut through the complexity, the most advanced organizations are applying a four-step approach to operationalize ML in processes.

A glossary of machine-learning terminology

DevOps: A set of practices that combine software development and IT operations. Because DevOps is based on continuous integration and continuous deployment, the implementation process is much faster and more agile than the traditional software-delivery life cycle.

Labeled data: A data set with clear parameters that distinguish specific attributes, used to train a machine-learning (ML) model. For example, if a company wanted to train an ML algorithm to

distinguish cats from dogs, it would show two collections of images and clearly delineate which are cats and which are dogs. From that baseline, the algorithm would be able to accurately categorize subsequent images.

Machine learning: Advanced algorithms that can “learn” from data without relying on rules-based programming.

MLOps: The application of DevOps concepts to operationalize machine learning.

Probabilistic: An automation solution that uses statistical functions to predict output based on trained behavior (“If A, then most probably B”).

Rule-based automation: A traditional approach to automation that relies on rules-based algorithms to predictable situations (“If A, then B”).

Step 1. Create economies of scale and skill

Because processes often span multiple business units, individual teams often focus on using ML to automate only steps they control. That, we find, is usually a mistake. Having different groups of people around the organization work on projects in isolation—and not across the entire process—dilutes the overall business case for ML and spreads precious resources too thinly. Siloed efforts are difficult to scale beyond a proof of concept, and critical aspects of implementation—such as model integration and data governance—are easily overlooked.

Rather than seeking to apply ML to individual steps in a process, companies can design processes that are more automated end to end. This approach capitalizes on synergies among elements that are consistent across multiple steps, such as the types of inputs, review protocols, controls, processing, and documentation. Each of these elements represents potential use cases for ML-based solutions.

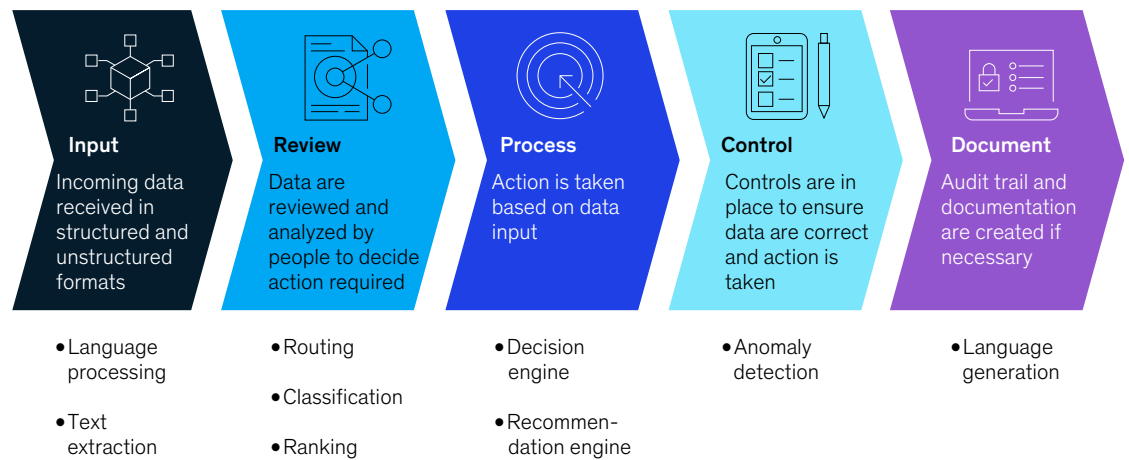
For example, several functions may struggle with processing documents (such as invoices, claims, contracts) or detecting anomalies during review processes. Because many of these use cases have similarities, organizations can group them together as “archetype use cases” and apply ML to them en masse. Exhibit 1 shows nine typical ML archetype use cases that make up a standard process.

Bundling automation initiatives in this way has several advantages. It generates a more attractive return on investment for ML development. It also allows the implementation team to reuse knowledge gained from one initiative to refine another. As a result, organizations can make faster progress in developing capabilities and scaling initiatives: one discovered that several initiatives were based on the same natural-language-processing technology, allowing it to save time in future development of similar solutions.

Exhibit 1

Nine machine-learning archetypes can be used to redesign processes across an organization.

A typical transactional process can benefit from nine machine-learning applications (not exhaustive)



Step 2. Assess capability needs and development methods

The archetype use cases described in the first step can guide decisions about the capabilities a company will need. For example, companies that focus on improving controls will need to build capabilities for anomaly detection. Companies struggling to migrate to digital channels may focus more heavily on language processing and text extraction.

As for *how* to build the required ML models, there are three primary options. Companies can:

- *build fully tailored models* internally, devoting significant time and capital on bespoke solutions that will meet their unique needs;
- *take advantage of platform-based solutions* using low- and no-code approaches; or




- *purchase point solutions* for specific use cases, which is easier and faster but requires trade-offs.

Exhibit 2 shows a list of the advantages and disadvantages of each approach.

Deciding among these options requires assessing a number of interrelated factors, including whether a particular set of data can be used in multiple areas and how ML models fit into broader efforts to automate processes. Applying ML in a basic transactional process—as in many back-office functions in banking—is a good way to make initial progress on automation, but it will likely not produce a sustainable competitive advantage. In this context, it is probably best to use platform-based solutions that leverage the capabilities of existing systems.

Exhibit 2

Machine-learning models can be built in three different ways depending on client context and situation.

Machine-learning models by client type			
	 Fully tailored	 Platform-based	 Point solutions
Description	Develop the solution entirely in-house	Code-free solution reduces need for data scientists	Solution is built to work for certain situations
	<ul style="list-style-type: none"> • Models are built from scratch using third-party libraries from major software vendors 	<ul style="list-style-type: none"> • Leverage existing platforms that facilitate creation of models 	<ul style="list-style-type: none"> • Solution has been built by someone for a specific use case
Pros	<ul style="list-style-type: none"> • Strengthens internal capabilities • Full IP protection and ownership • End solution tailored to specific needs and owned by business 	<ul style="list-style-type: none"> • Reduced need for data-science knowledge • Rapid time to market • Great for cost reduction 	<ul style="list-style-type: none"> • Very quick time to market • Requires limited capabilities to deploy
Cons	<ul style="list-style-type: none"> • Significant time to market • Expensive for development and maintenance • Requires strong internal capabilities 	<ul style="list-style-type: none"> • Solution not fully tailored to processes • Partial ownership of solution potentially reducing competitive advantage 	<ul style="list-style-type: none"> • Can become expensive long term • Nonreusable for other uses • No ownership of solution
Effort to deploy	High	Medium	Low

Step 3. Give models ‘on the job’ training

Operationalizing ML is data-centric—the main challenge isn’t identifying a sequence of steps to automate but finding quality data that the underlying algorithms can analyze and learn from. This can often be a question of data management and quality—for example, when companies have multiple legacy systems and data are not rigorously cleaned and maintained across the organization.

However, even if a company has high-quality data, it may not be able to use the data to train the ML model, particularly during the early stages of model design. Typically, deployments span three distinct, and sequential, environments: the developer environment, where systems are built and can be easily modified; a test environment (also known as user-acceptance testing, or UAT), where users can test system functionalities but the system can’t be modified; and, finally, the production environment, where the system is live and available at scale to end users.

Even though ML models can be trained in any of these environments, the production environment is generally optimal because it uses real-world data (Exhibit 3). However, not all data can be used in all three environments, particularly in highly regulated industries or those with significant privacy concerns.

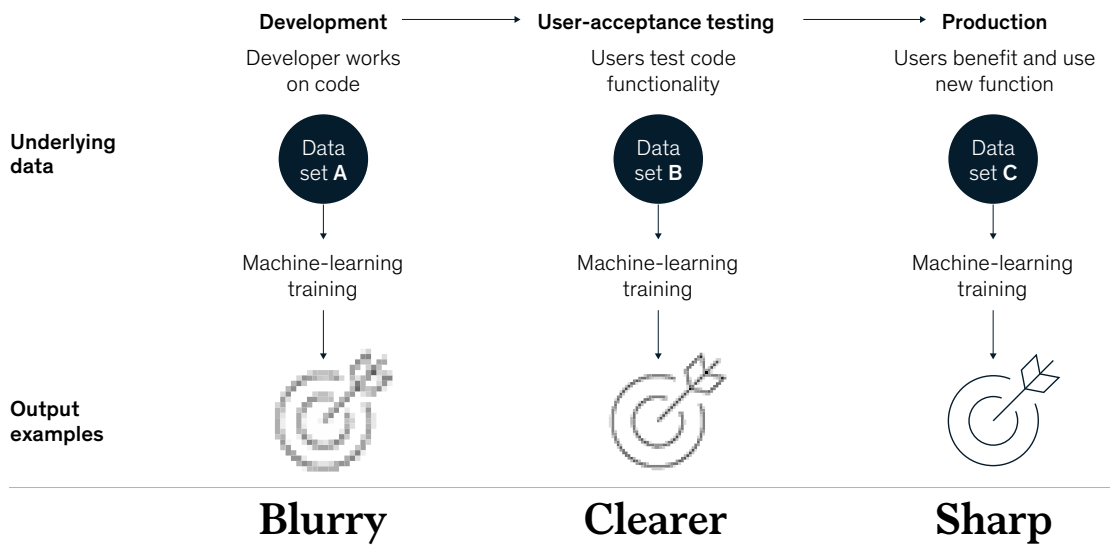
In a bank, for example, regulatory requirements mean that developers can’t “play around” in the development environment. At the same time, models won’t function properly if they’re trained on incorrect or artificial data. Even in industries subject to less stringent regulation, leaders have understandable concerns about letting an algorithm make decisions without human oversight.

To deal with this challenge, some leading organizations design the process in a way that allows a human review of ML model outputs (see sidebar “Data options for training a machine-learning model”). The model-development team sets a threshold of certainty for each decision and enables the machine to handle the process with

Exhibit 3

Matching the right data set to the right production stage is critical for successful deployment of machine learning.

Performance of machine-learning models will vary based on data set used



Data options for training a machine-learning model

Companies can choose among several data-management approaches to training machine-learning (ML) models, bearing in mind the need to start from the best available labeled data and comply with applicable regulatory and privacy standards.

Human in the loop: In situations where the data set is available only in the production environment (often for legal reasons) or data quality is sparse, the delivery team will want to gradually create the outputs via manual processing and use those to train and iteratively improve the ML model.

Standard deployment: If high-quality data sets can be found in both test and production environments, the company

can simply follow a standard sequence in training, testing, and deploying the ML model.

Set up an artificial production environment: If a data set is available for the production environment, companies can create a simulated, preproduction environment that uses the data for training purposes without live systems used by end users.

Anonymize the production data set: In some cases—often because of legal constraints—the production data set must be anonymized before being moved to a training environment (for example, customer names removed).

Replicate the production data set in the DEV/UAT environments: In some cases, the correct production data set is available and can be safely moved to a separate environment (DEV/UAT) to train the model.

Fully create a training set in DEV/UAT: If there are no correct data available in the different IT environments, a new, separate training data set needs to be created by the end users for the ML model.

Use an alternative data set with similar features: Rather than creating a data set from scratch, the team can find an alternative with similar features and behavior of the production data set.

full autonomy in any situation that exceeds that threshold. This human-in-the-loop approach gradually enabled a healthcare company to raise the accuracy of its model so that within three months, the proportion of cases resolved via straight-through processing rose from less than 40 percent to more than 80 percent.

Step 4. Standardize ML projects for deployment and scalability

Innovation—in applying ML or just about any other endeavor—requires experimentation. When researchers experiment, they have protocols in place to ensure that experiments can be reproduced and interpreted, and that failures can be explained. The same logic should be applied to ML. An organization should accumulate knowledge even when experiments fail.

The right guidance is usually specific to a particular organization, but best practices such as MLOps can help guide any organization through the process. MLOps refers to DevOps—the combination of software development and IT operations—as applied to machine learning and artificial intelligence. The approach aims to shorten the analytics development life cycle and increase model stability by automating repeatable steps in the workflows of software practitioners (including data engineers and data scientists).

Although MLOps practices can vary significantly, they typically involve a set of standardized and repeatable steps to help scale ML implementation across the enterprise, and they address all components needed to deliver successful models (Exhibits 4 and 5).

Exhibit 4

Achieving scale requires a standardized and repeatable approach to machine-learning operationalization.

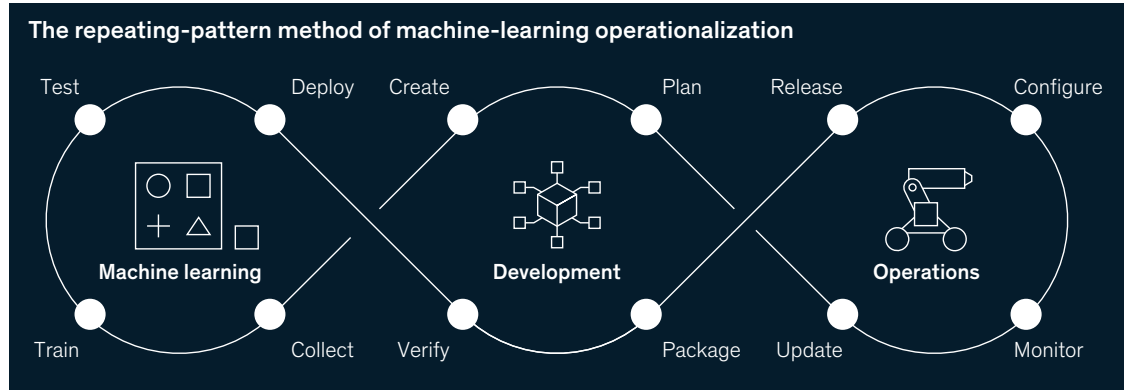
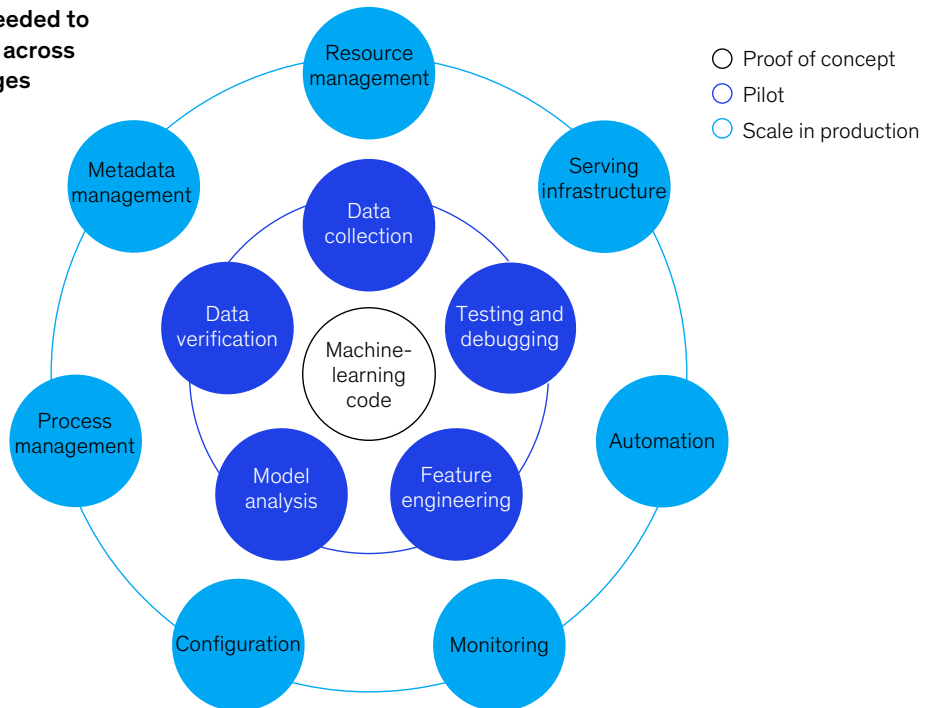


Exhibit 5

Machine-learning operations covers all components needed to deliver models.

Components needed to deliver models, across production stages



While standardizing delivery is helpful, organizations also need to address the people component—by assembling dedicated, cross-functional teams to embed ML into daily operations. Modifying organization structures and building new capabilities are both critical for large-scale adoption. The healthcare company built an ML model to screen up to 400,000 candidates each year. This meant recruiters no longer needed to sort through piles of applications, but it also required new capabilities to interpret model outputs and train the model over time on complex cases.

Adopting the right mindsets

ML has become an essential tool for companies to automate processes, and many companies are seeking to adopt algorithms widely. Yet the journey is difficult. The right mindsets matter.

The more data, the better. Unlike rule-based automation, which is highly centered around processes, ML is data-centric. A common refrain is that the three most important elements required for success are data, data, and more data.

Plan before doing. Excitement over ML's promise can cause leaders to launch too many initiatives at once, spreading resources too thin. Because the ML journey contains so many challenges, it is essential to break it down into manageable steps. Think about archetypical use cases, development methods, and understand which capabilities are needed and how to scale them.

Think end to end. Asking managers of siloed functions to develop individual use cases can leave value on the table. It's important to reimagine entire processes from beginning to end, breaking apart the way work is done today and redesigning the process in a way that's more conducive to how machines and people work together.

There is a clear opportunity to use ML to automate processes, but companies can't apply the approaches of the past. Instead, the four-step approach outlined here provides a road map for operationalizing ML at scale.

Rohit Panikkar and **Rob Whiteman** are partners in McKinsey's Chicago office, **Tamim Saleh** is a senior partner in the London office, and **Maxime Szybowski** is a consultant in the Zurich office.

Designed by McKinsey Global Publishing
Copyright © 2021 McKinsey & Company. All rights reserved.