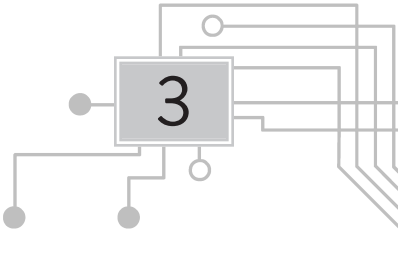


The AI Factory

Through much of history, products were painstakingly and individually crafted in artisanal workshops. That ended when the Industrial Revolution transformed the economy by spawning a scalable and repeatable approach to manufacturing. Engineers and managers became experts at understanding the processes needed for mass production and built the first generation of factories, dedicated to the continuous, low-cost production of quality goods. However, while production was industrialized, analysis and decision making remained largely traditional, idiosyncratic processes.

Now, the age of AI is manifested by companies driving another fundamental transformation. This one involves industrializing data gathering, analytics, and decision making to reinvent the core of the modern firm, in what we call the "AI factory."[1]

The AI factory is the scalable decision engine that powers the digital operating model of the twenty-first-century firm. Managerial decisions are increasingly embedded in software, which digitizes many processes that have traditionally been carried out by employees. No human auctioneer gets involved in the millions of daily search-ad auctions at Google or Baidu. Dispatchers do not decide which car is chosen on DiDi, Grab, Lyft, or Uber. Sports retailers do not set daily prices on golf apparel at Amazon. Bankers do not approve every loan at Ant Financial. Instead, these processes are digitized and enabled by an AI factory that treats decision making as an industrial process. Analytics systematically convert internal and external data into predictions, insights, and choices, which in turn guide or even automate a variety of operational actions. This

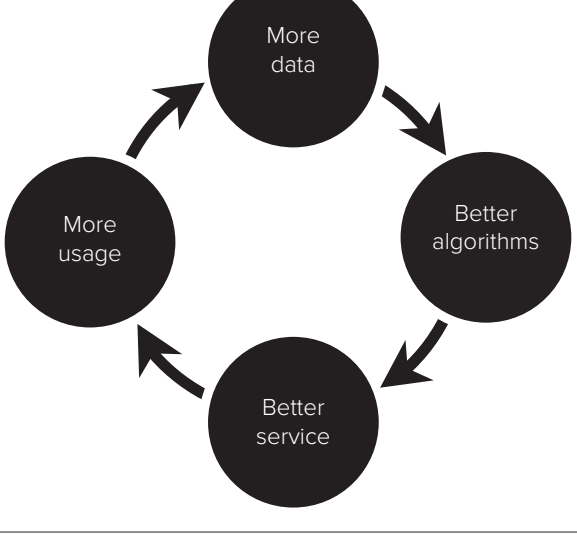
is what enables the superior scale, scope, and learning capacity of the digital firm.

Digital operating models can take various forms. In some cases, they might only manage flows of information (think Ant Financial, Google, or Facebook). In other cases, operating models guide how the company builds, delivers, or operates actual physical products (think Ocado, Amazon, or Waymo). In either case, AI factories are at the core of the model, guiding the most critical processes and operating decisions, while humans are moved to the edge, off the critical path of value delivery.

In its essence, the AI factory creates a virtuous cycle between user engagement, data collection, algorithm design, prediction, and improvement (see figure 3-1). It integrates data generated from multiple sources (internal or external to the firm) to refine and train a set of algorithms. These algorithms not only make predictions but also use the data to improve their own accuracy. The predictions then drive decisions and actions, either by informing human insights or by enabling an automated response. Hypotheses about changing

FIGURE 3-1

The AI factory's virtuous cycle



customer behavior patterns, competitive responses, and process variations are tested through rigorous experimentation protocols that enable causal identification of changes that might improve the system. Data about usage and about the accuracy and impact of the prediction outcomes is then sent back into the system for further learning and predictions. And the cycle continually repeats.

Take, for example, a search engine like Google or Bing. As soon as a user types a few letters in the search box, algorithms dynamically predict the full search term based on prior search terms and the user's past actions. These predictions are captured in a drop-down menu (the *autosuggest box*), which helps users zero in quickly on the desired search. Every user movement and every click are captured as data points, and every data point gathered improves the prediction for future searches. The more searches, the better the predictions, and the better the predictions, the more the search engine is used.

There are multiple other prediction cycles in a search engine's AI factory. During the natural search process, the search term entered by a user generates a display of organic search results, which are drawn from a previously assembled index of the web and optimized by using the outcomes (the clicks generated) of previous searches. In addition, entering the search term also starts an automated auction for the most relevant ads to match the user's intent, an auction whose results are also shaped by additional learning loops. The search-results page, which combines organic search results and relevant ads, is thus heavily influenced by data on previous search attempts. Any click on or away from the search query or search-results page provides useful data.

In addition, a product manager within the search engine operations might have some new hypothesis—for example, that showing fewer ads might improve revenues on a given page, or that highlighting search results would improve click-through rates. To provide additional fodder for improvement, these hypotheses would be loaded on the experimental machinery and tested on a statistically relevant sample of users.

Clearly there is no way all this data could be analyzed by a few analysts using manual tools, or even by casually assembled code.

The AI factory solves this problem by bringing mass production methods to data processing and analytics, thus forming the core of a digital operating model. Let's dig deeper into its nature, using Netflix to anchor the discussion.

Building and Running the AI Factory

Netflix has transformed the media landscape by harnessing the power of artificial intelligence. The core of Netflix is its AI-centric operating model: it is powered by software infrastructure that gathers data and trains and executes algorithms that influence virtually every aspect of the business, from personalizing the user experience to picking movie concepts to negotiating content agreements.

In its earliest days two decades ago, Netflix displayed movie reviews, generated recommendations based on customers' viewing histories, and shipped DVDs of new releases the day they were made available in stores. Even then, Netflix recognized the importance of using data to improve the customer experience. The company's early efforts were focused on developing a recommendation engine, which suggested movies based on a viewer's history, movie ratings, and the preferences of similar viewers.[2] Netflix not only used this data internally but also shared the reviews with movie studios. Sharing this data helped Netflix negotiate better financial terms in its partnerships with Warner Home Video and Columbia TriStar.[3]

Netflix grew rapidly, hitting eight million subscribers in 2007 when it launched its streaming service. This new offering dramatically increased the company's access to user data, which Netflix analytics teams used extensively. With its mail delivery service, Netflix could track only those titles users requested, the length of time they kept a DVD, and their rating of each title; Netflix could not monitor actual viewing behavior. With streaming, Netflix could track the full user experience—when viewers pause, rewind, or skip during a show, for example, or what device they are using. This behavioral data helped Netflix determine which movie thumbnail

image to show a viewer (yes, even these are personalized based on preferences for particular genres, actors, and other such factors), predicting their likely preferences. Through more-advanced analytics, Netflix also predicted drivers of customer loyalty. With the goal of increasing subscriber viewing time and decreasing customer churn rates, Netflix used AI to launch a function that automatically queues the next episode in a series or recommends similar movies. The customization and personalization has become pervasive. As Joris Evers, then chief of communications at Netflix, told the *New York Times* in 2013, "[T]here are 33 million different versions of Netflix," meaning that each user's Netflix experience is personalized and customized.[4]

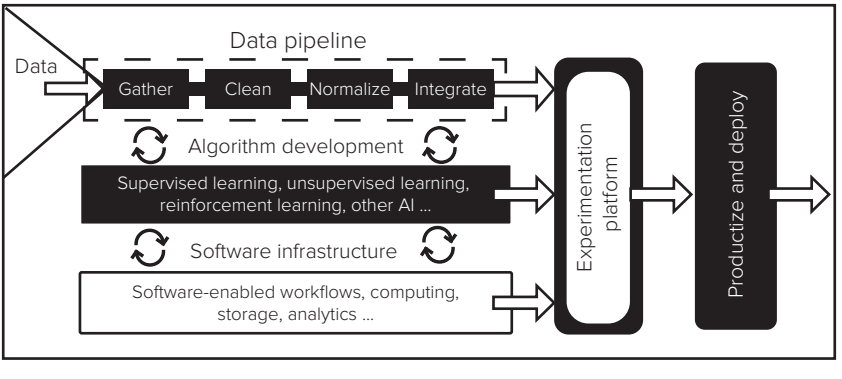
Netflix also uses data and AI algorithms to decide which content to create on its own. The company's first use of predictive analytics for this purpose was in 2013 to evaluate the potential of *House of Cards*, the fictional account of a senator's rise to the White House, in collaboration with Media Rights Capital (MRC). Cindy Holland, vice president of original content, noted in an interview, "We have projection models that help us understand, for a given idea or area, how large we think an audience size might be, given certain attributes about it. We have a construct for genres that basically gives us areas where we have a bunch of programs and others that are areas of opportunity."[5]

By 2010 Netflix was embracing the AI factory approach to systematically apply analytics and AI to the company's recommendation engine. In 2014, the company expanded the factory to improve the streaming experience by understanding user behavior, creating a personalized streaming experience for each user (based on such factors as connection speed and preferred device), and determining what movies and shows to cache on "edge servers," which are deployed closer to viewers.[6] Now Netflix has about 150 million subscribers in more than 190 countries, has amassed a content library of more than 5,500 shows, and consumes 15 percent of the global internet bandwidth.

Experience from Netflix and other leading firms underlines the importance of a few essential AI factory components (see figure 3-2):

FIGURE 3-2

AI factory components



1. ***Data pipeline:*** This process gathers, inputs, cleans, integrates, processes, and safeguards data in a systematic, sustainable, and scalable way.

2. ***Algorithm development:*** The algorithms generate predictions about future states or actions of the business. These algorithms and predictions are the beating heart of the digital firm, driving its most critical operating activities.

3. ***Experimentation platform:*** This is the mechanism through which hypotheses regarding new prediction and decision algorithms are tested to ensure that changes suggested are having the intended (causal) effect.

4. ***Software infrastructure:*** These systems embed the pipeline in a consistent and componentized software and computing infrastructure, and connect it as needed and appropriate to internal and external users.

If the data is the fuel that powers the AI factory, then infrastructure makes up the pipes that deliver the fuel, and the algorithms are the machines that do the work. The experimentation platform, in turn, controls the valves that connect new fuel, pipes, and machines to existing operational systems.

Let's look first at the data pipeline.

The Data Pipeline

Data is the essential input of the AI factory. One reason for the radical advances made by AI systems in recent years is that the velocity, volume, and variety of data available for analysis has exploded. As far back as 2012, Netflix was using a broad base of data inputs. As described by Xavier Amatriain and Justin Basilico, two Netflix engineers, on the Netflix blog, the inputs are varied.

- *We have several billion item **ratings** from members. And we receive millions of new ratings a day.*
- *We already mentioned item **popularity** as a baseline. But, there are many ways to compute popularity. We can compute it over various time ranges, for instance hourly, daily, or weekly. Or, we can group members by region or other similarity metrics and compute popularity within that group.*
- *We receive several million stream **plays** each day, which include context such as duration, time of day and device type.*
- *Our members add millions of items to their **queues** each day.*
- *Each item in our catalog has rich **metadata**: actors, director, genre, parental rating, and reviews.*
- ***Presentations**: We know what items we have recommended and where we have shown them, and can look at how that decision has affected the member's actions. We can also observe the member's interactions with the recommendations: scrolls, mouse-overs, clicks, or the time spent on a given page.*
- ***Social** data has become our latest source of personalization features; we can process what connected friends have watched or rated.*
- *Our members directly enter millions of **search terms** in the Netflix service each day.*
- *All the data we have mentioned above comes from internal sources. We can also tap into **external data** to improve our*

features. For example, we can add external item data features such as box office performance or critic reviews.

- *Of course, that is not all: there are many* ***other*** *features such as demographics, location, language, or temporal data that can be used in our predictive models.*[7]

In 2018, Netflix users had more than 5,600 movie and TV series titles to choose from. Every time users open the Netflix application on their TV, computer, phone, or tablet, the company's systems kick in to make personal recommendations and customize the interface. Virtually every aspect of a user's experience generates data, which then enables Netflix to further fine-tune the customizations it provides. (And certainly, there is much more data available now than when this post was written in 2012.) All of this data is cleaned, integrated, prepared, and used by Netflix to dynamically adapt its service to continuously improve the value it provides to its estimated 300 million users.

The depth and breadth of the Netflix data is the envy of the industry. Part of the company's data and analytics assets includes creating approximately two thousand *microclusters*, or taste communities, which connect viewers having similar tastes. Individual users can fit in to several taste communities, and they defy simple demographic profiles; a sixty-five-year-old grandmother in urban Mumbai may like the same shows as a teenager in rural Arkansas.

Netflix has "datafied" TV entertainment—a term coined by Ming Zeng, Alibaba's strategy chief and academic counsel. The idea of *datafication* refers to systematically extracting data from activities and transactions that are naturally ongoing in any business.[8] The Nest thermostat, for example, invaded a sleepy market by datafying a traditional spectrum of activities—controlling the heating, ventilation, and cooling (HVAC) systems in a home. The addition of a few electronic sensors to monitor temperature and motion in the home, along with computer-based control and Wi-Fi connectivity, enabled Nest to create a brand-new data layer that generates important new value for homeowners. The Nest device, in only a few days, can learn your habits and adjust the temperature auto-

matically in your house, participate in energy reduction programs at your nearby utility, and enable smartphone control.

Similar datafication has happened in almost every setting, from social behavior on Facebook to fitness with an Apple Watch or Fitbit, to sleep and health tracking with the Oura and Motiv rings.[9] Increasingly, as in the Netflix example, the initial process of datafication can be combined with external data sources to provide additional value to the user. The Oura ring's app, for example, combines sleep and heart rate data with the user's activity level monitored by an Apple Watch to coach the user on the level of rest and activity needed for a productive day. Ride-sharing platforms like Uber, Lyft, Grab, DiDi, and GOJEK have built a datafication layer around transportation. The combination of their applications and the smartphone infrastructure has enabled these companies to generate data at an unprecedented level about individual transportation preferences, demand and supply needs, and overall flow of traffic in and out of urban centers. Accurate, real-time data about all this has never existed until now.

Sometimes, innovation is needed to transform traditional activities into sources of useful data. Alipay and WeChat Pay have led the way in economic transactions through their extensive use of QR codes for payments. If data is not readily available or does not exist, it may be worthwhile for a company to invest in technology and services that generate the data in the first place. Even Pitney Bowes, the hundred-year-old provider of postal equipment, has built a datafication strategy around physical addresses in the United States and is augmenting the company's business model by offering data-driven Knowledge Fabric solutions to banks, insurers, social platforms, and retailers—any organization that can use address data for marketing, fraud detection, and other purposes. The company realized that it could create and capture value beyond selling postage.

Many incumbent businesses that are attempting to build AI factories find that the data they possess is fragmented, incomplete, and often siloed within divisions and disparate IT systems. Take, for example, a typical hotel stay for a business traveler. In theory, a hotel

chain should have a treasure trove of data on their customers, from home address to credit card information, to frequency of travel, airline, and mode of transportation, location of travel, class of stay, meal selections, local sightseeing favorites, and health and fitness preferences. In practice, though, the data is highly fragmented, resides in various system silos with incompatible data structures, is missing common identifiers, and may not necessarily be very accurate. Executives at many incumbent companies consistently underestimate the challenge and the urgency of the investment they face in cleaning and integrating their data across the enterprise so that they can build an effective AI factory. The first order of business facing these executives is to ensure that the appropriate investments are in place.

We emphasize that after the data is gathered, much work remains to be done in cleaning, normalizing, and integrating it. These steps are quite challenging. Data assets are most often plagued by all kinds of biases and even plain errors, and a significant investment needs to be made in ensuring that the data is checked carefully for inaccuracies and inconsistencies. Moreover, as various streams of data are integrated into a single stream to feed complex analysis, the different kinds of data must be normalized. A particular challenge is making sure that financial data is being used properly, in a way that is consistent with operational data, so that any insight that comes from analyzing the integrated dataset is accurate. For example, units should be consistent, redundancies eliminated, and variables compatible. These things often sound simple but are not, especially as the datasets reach significant size.

Algorithm Development

After the data is gathered and prepared, the tool that makes the data useful is the *algorithm*—the set of rules a machine follows to use data to make a decision, generate a prediction, or solve a particular problem.

Consider how you would analyze whether a customer is likely to leave a service like Netflix. Here the algorithm would predict

customer churn as a function of variables such as usage (frequency and intensity), satisfaction, demographics, and relationships or similarities with other users. The prediction algorithm would be tuned and calibrated with data on past customers, tested for accuracy with past data or with a controlled experiment, and deployed either as an analytical tool for managers or as a step in an operational process—for example, automatically enabling a special offer to retain vulnerable customers.

Ajay Agrawal, Josh Gans, and Avi Goldfarb of the University of Toronto note that data proliferation and advances in AI algorithms have lowered the cost of making accurate predictions, increasing the scope and intensity of the usage of prediction algorithms throughout the economy.[10] Algorithms predict which Google photos include family members or friends, what Facebook content you should read next, how much revenue to expect from giving a Walmart discount to a particular customer, or when a piece of equipment at a Ford manufacturing facility will need maintenance. These kinds of predictions are vital to the success of many organizations, and the algorithms deployed should be geared to provide consistent and robust predictions.

AI algorithms can be used for a broad variety of applications, from generating relatively simple predictions (like a sales forecast) to suggesting stocks to pick for high-frequency trading, to complex image recognition and language translation tasks that may exceed human capabilities. Some of the most complex applications, such as driving a car, use a variety of different algorithms simultaneously—for example, to identify and track cars and to route a car through heavy traffic.

Although the use of applications has exploded over the past decade, the foundations of algorithm design have been around for quite some time.[11] The conceptual and mathematical development of classic statistical models such as linear regression, clustering, or Markov chains date back more than a hundred years. Although neural networks are now generating a lot of excitement, they were initially developed in the 1960s and are only now being put to use at scale with production-ready outputs. The vast majority of production-ready and operational AI systems use one of three general approaches to

develop accurate predictions using statistical models, also known as machine learning. These are supervised learning, unsupervised learning, and reinforcement learning.

Supervised Learning

The basic goal of *supervised* machine learning algorithms is to come as close as possible to a human expert (or an accepted source of truth) in predicting an outcome. The classic case is analyzing a picture and predicting whether the subject is a cat or a dog. In this case the expert would be any human being who could label photos as images of a cat or a dog. The algorithms in this class of machine learning systems rely on an *expert-labeled* dataset of the outcome (the Y) and the potential characteristics or features (the Xs). The operationalization of the algorithm is called a *model*, which takes the general-purpose statistical approach and creates a context-specific instantiation of the prediction problem that needs to be solved.

The first step in supervised learning is to create (or acquire) a labeled dataset. For example, we might acquire a file containing thousands of pictures of cats and thousands of pictures of dogs, with each picture labeled appropriately. The data is then split between training and validation. The *training* dataset is used to determine the parameters of the model that generates the prediction of the outcome (whether a given picture depicts a cat or a dog). After the model is trained, the *validation* dataset is used to test the accuracy of the model. The model makes its predictions on the validation dataset; we can then compare these predictions to the expert predictions and thereby assess the quality of the model. Supervised machine learning algorithms can be used to predict either a binary outcome (for example, whether a picture shows a cat or a dog) or a numerical quantity (such as the sales forecast for a particular product).[12]

As we compare the algorithmic model's prediction of the outcome to the validated labeled outcomes, we can determine whether we are satisfied with the error rate between model prediction and

expert. If we are not satisfied, we can choose a different statistical approach, get more data, or work on identifying other features that may be helpful in making a more accurate prediction. The main challenge here is to keep iterating between data, features, and algorithms until we are satisfied with the error rate between the model prediction and the expert prediction.

Examples of supervised machine learning abound. Every time we label an email as spam, we help our email provider's machine learning algorithms update its models to identify the latest clever scam. Facebook's or Baidu's ability to suggest names of friends who may appear in newly uploaded pictures is based on our prior labeling of photos. Credit card companies or payment platforms decide whether to allow a transaction based on prior purchasing habits, which automatically create labeled data. A Nest thermostat's ability to change the temperature in your living room thirty minutes before you arrive home is based on autogenerated labeled data gathered from your previous arrival and departure times, as well as your prior temperature-setting habits.

Netflix uses supervised learning in a variety of scenarios. For recommendations, Netflix has used labeled datasets made up of actions and results (e.g., movies chosen and liked) by people who are deemed by the algorithm to be similar to a given user. A large dataset of user choices, calibrated by characteristics of the user and of the decision context, can lead to effective recommendations. This kind of *collaborative filtering algorithm* is used for all kinds of recommendations, including Amazon's shopping engine and Airbnb's matching engine.

Many companies may already have vast troves of algorithm-ready labeled data thanks to their investments in systems, technologies, databases, and heavyweight enterprise resource planning (ERP) installations. For example, most large insurance companies have decades of labeled data relating to property damage and could readily implement supervised machine learning models to reduce both fraud and the time it takes to process and resolve claims—especially if the company is equipped for direct photo uploads or drone-based inspection. Similarly, health-care systems are full of labeled datasets. For example, many companies are taking medical

data (such as radiology, cardiology, pathology, and EKG results) and correlating it with health diagnoses. Israel-based Zebra Medical Vision now offers technology to help radiologists make better diagnoses from X-ray, CT, and MRI scans.

Unsupervised Learning

Unlike supervised learning models, which train a system to recognize known outcomes, the primary application of *unsupervised* learning algorithms is to discover insights in data with few preconceptions or assumptions. This is what Netflix does when it discovers related groups of customers in analyzed viewing data, when it creates customer segments for marketing campaigns, or when it creates different versions of the user interface that match different usage patterns. Or think of various national security agencies and law enforcement organizations accumulating huge amounts of social media data to look for abnormal patterns and discern potential security threats. In these cases, one does not know exactly what to look for but is searching for related groups or for events that fit or don't fit established patterns.

Unlike supervised learning algorithms, where the data inputs are labeled with a given outcome, unsupervised learning algorithms aim to find "natural" groupings in the data, without labels, and uncover structures that may not be obvious to the observer. Thus the job of the algorithm is to show patterns in data, with humans (or even other algorithms) labeling the patterns or groups and deciding on potential actions. In our example of photos of cats and dogs, an unsupervised learning algorithm might find several types of groupings. Depending on how the clusters are structured, these groupings could end up separating cats and dogs, or indoor and outdoor photographs, or pictures taken during day or night, or virtually anything else. Again, an unsupervised learning algorithm does not suggest specific labels but rather establishes the most robust statistical groupings. Humans, or other algorithms, do the rest.

Unsupervised learning is useful for gaining insights from social media postings by, say, identifying customer groups and sentiment

patterns that can be used to guide product development. Attitudinal and demographic survey responses by customers can be used to create customer segments. The reasons for customer churn could also be categorized through unsupervised learning. In manufacturing settings, one could group instances of machine failure or order delay.

There are three broad types of unsupervised learning. The first relates to algorithms that *cluster* data into groups. A fashion retailer may use this approach to understand how to segment its customers based on the types of products purchased, the pricing and profitability of the items, and the various channels that brought customers to the store. More-sophisticated retailers might have additional data such as social network-based graph data (whom customers are connected to) and their social media postings. All this data then can allow the company to uncover a unique set of segments, well beyond simple demographics.

Netflix microclusters—its taste communities of members with similar movie and series preferences—is a good illustration of the power of such a tool. Cluster analysis in the form of topic modeling is used extensively to find meaning in text-based data and uncover salient topics within and across texts. The technique has been used to analyze news reports, SEC filings, investor calls, customer call center transcripts, or even chat records.

The second broad category is known as *association rule mining*. A common example is the recommendations for additional products an online shopper might want to purchase based on the current set of products in the shopping cart. Amazon has made a science of association rule mining. The algorithms look for frequency and probability of co-occurrence among any set of items and then create associations that are likely to occur between various types of products. Ocado, for example, learned from its data that there was a strong relationship between diapers and beer. New parents don't get to go out much, so recommending beer and wine to shoppers when they are purchasing diapers turned out to be profitable and also increased customer satisfaction.

The third type of unsupervised learning algorithm is *anomaly detection*. Here the algorithm simply looks at each new incoming

observation or datum and makes the judgment whether or not it fits prior patterns. If it does not fit the pattern, then the algorithm flags that item as anomalous. This type of application is often used in fraud detection in financial services, health care for a variety of patient data, and maintenance of systems and machines.

Reinforcement Learning

Although they are still relatively underdeveloped, the potential applications of *reinforcement* learning may be even more impactful than those of supervised and unsupervised learning. Rather than start with data on an expert's view of the outcome, as in supervised learning, or with a pattern-and-anomaly recognition system, as in unsupervised learning, reinforcement learning requires only a starting point and a performance function. We start somewhere and probe the space around us, using as a guide whether we have improved or worsened our position. The key trade-off is whether to spend more time *exploring* the complex world around us or *exploiting* the model we have built so far to drive decisions and actions.

Let's say we take a cable car up a tall mountain and we want to find our way down. It's a foggy day, and the mountain does not have any clearly marked paths. Because we can't see the best way down, we have to walk around and explore different options. There is a natural trade-off between the time we spend walking around getting a feel for the mountain, and the time we spend actually walking down when we believe we have found the best path. This is the trade-off between exploration and exploitation. The more time we spend exploring, the more we will be convinced we have the best way down, but if we spend too long exploring, we will have less time to exploit the information and actually walk down.

This is close to the way the Netflix algorithm personalizes movie recommendations and the visuals they are associated with.[13] The problem is a bit more complicated, because the Netflix team needs to figure out which movie selection to present and then which artwork to combine it with to maximize the match between user and recommendation. But in a way similar to our finding our way

down the mountain, Netflix spends some time exploring options, and some time exploiting the solution offered by its models. To explore visual options, Netflix systematically randomizes the visuals shown to a user, thereby exploring new possibilities and refining the prediction model. Netflix then exploits the improved model to show the user a slew of recommendations with improved visuals.

The Netflix service continues to improve dynamically by automatically cycling between periods of exploration and exploitation, a process designed to learn the most about the preferences of a complex human being and maximize user engagement over the long term. The writer of the Netflix technology blog asked in a 2017 post, "Given the enormous diversity in taste and preferences, wouldn't it be better if we could find the best artwork for each of our members to highlight the aspects of a title that are specifically relevant to them?"[14]

The Netflix challenge is a fancy variant of a common class of models used in reinforcement learning. Known as the *multiarmed bandit problem*, it is named after imagining a gambler playing different slot machines ("one-armed bandits"), each machine characterized by a different (but unknown) reward distribution. The gambler can spend more time exploring which machine seems to give the best rewards or can focus on exploiting the one machine that seems to be the best bet so far. Any deviation from the optimal path (just cranking on the best machine) is expressed as the *regret* measure. Multiarmed bandit problems are useful in the allocation of finite resources across different processes, each associated with different reward distributions. The general idea is to maximize operating performance by minimizing regret.

Multiarmed bandit problems are vitally important to the deployment of AI in operating models. As we strive to optimize and improve operating performance across processes, managing the trade-off between exploration and exploitation is fundamental. These algorithms are used extensively to manage a variety of operating workflows, from making product recommendations to setting product prices, and from planning clinical trials to selecting digital ads. They can even guide the behavior of actual agents in imagined or real worlds, from the path of Nintendo's Mario Kart video game

to the bots in Ocado's warehouses. In essence, multiarmed bandits are set up to make real operating decisions while they optimize the trade-offs between short-term impact and long-term improvement.

Reinforcement learning has captured public attention thanks to a software system called AlphaGo. Created by Google's DeepMind AI research team, AlphaGo has started to beat master players around the world at the ancient Chinese strategy game Go. Although computers have beaten humans at chess (remember Deep Blue by IBM), Go was thought to be too complicated for any program to master it. However, starting in 2016, this started to change as top Go masters kept losing to AlphaGo. These results were stunning—so much so that Kai-Fu Lee, an eminent computer scientist and technology investor, noted in his book *AI Superpowers* that the Chinese government declared its own "Sputnik moment" and made achieving world-class leadership in AI a national priority, with tremendous resources dedicated to achieving this goal.

That was before AlphaGo Zero came on to the scene and started beating AlphaGo at its own game. AlphaGo Zero uses the reinforcement learning approach: unlike prior versions of AlphaGo, wherein data from hundreds of thousands of games was used as input, the AlphaGo Zero system was essentially given the rules of the game and then asked to figure out the best approaches (the "Zero" stands for no external data). Reinforcement learning works by having a software agent interact with the environment and take actions within it to maximize a predefined reward. By giving the rules of the game or environment to the agent, the software system can quickly learn to maximize rewards and achieve superior performance. Google's DeepMind team has applied the lessons from Go to drug discovery and protein folding and has found that its system performs considerably better than the best scientists and their approaches.

The Experimentation Platform

To be reliably impactful, the wealth of predictions generated by data and algorithms in an AI factory requires careful validation. Google runs more than one hundred thousand experiments each year to test

a vast variety of potential data-driven improvements to its service. LinkedIn reportedly runs more than forty thousand experiments each year. The experimentation capacity required by digital operating models is such that traditional, ad hoc approaches to experimentation simply cannot handle the scale and impact of what is required. A state-of-the-art experimentation platform will provide the comprehensive set of technologies, tools, and methods required to do experimentation at scale.

To use an experimentation platform, potential significant changes to the business must first be formalized as a hypothesis. Each hypothesis is then typically tested as a *randomized control trial* (also known as an A/B test) in which a random sample of users is exposed to the change (known as a *treatment*) and a second random sample of users experience business as usual (the *control*). The outcomes are then compared, and if the difference between them is statistically significant the treatment is known to actually impact the outcome, instead of just being spuriously correlated. This approach ensures that any prediction being generated by algorithms actually has a *causal* effect on the outcome.

The experimentation platform is a necessary component of the AI factory. Imagine running our algorithm to predict customer churn and learning that churn correlates with a certain age group. We still do not know whether customers in that age group are more likely to churn in general, or whether they would respond positively to some kind of special offer and continue to use our service. Before offering an expensive rebate to millions of customers, it would make sense to try an A/B test on a small fraction of users and gather statistically significant evidence on what portion of customers would remain with our service *because of* that specific offer. The same kind of logic applies to a great variety of potential business improvements recommended by an AI factory at scale.

Netflix engineers and data scientists have built an extensive experimentation platform that is fully integrated within its algorithm development and execution process.[15] Every significant product change at Netflix goes through A/B testing before it becomes a standard part of the product experience. The experimentation platform is also utilized to improve video streaming and content

delivery network algorithms (the service supports hundreds of devices and a vast range of bandwidth conditions) as well as image selection, user interface changes, email campaigns, playback, and registration.

Indeed, the company tries to bring scientific rigor to all of its decision making by embracing experimentation as an integral component. The fully automated experimentation platform enables Netflix employees to run experiments at scale. The platform allows them to kick off the experiment, ensures there are no other blocking experiments or overlapping subject pools, recruits subjects from its audience, and creates reports to analyze and visualize results both during and after the experiments are completed.

Software, Connectivity, and Infrastructure

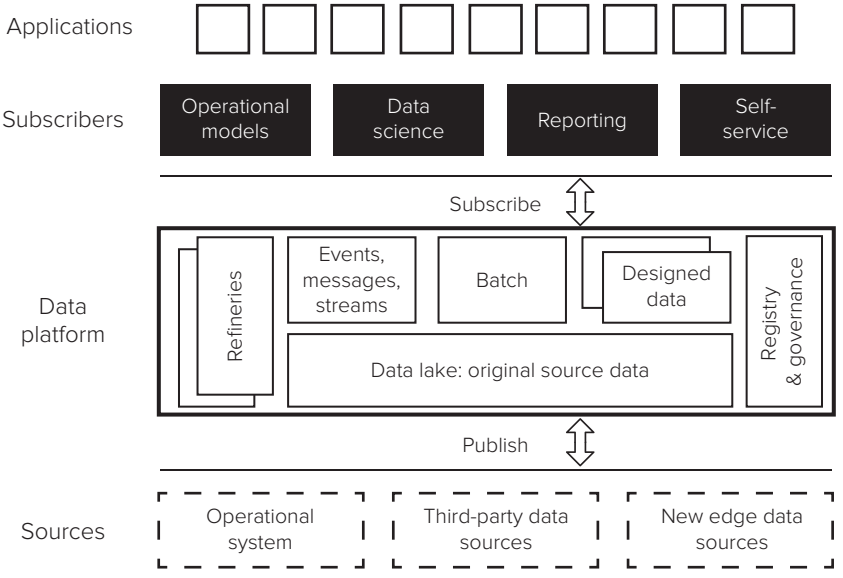
The data pipeline, the algorithm design and execution engine, and the experimentation platform should all be embedded in software infrastructure to drive the operating activities of the digital firm.

Figure 3-3 depicts an example of a state-of-the-art data platform powering an AI factory, with data flowing from bottom to top. The data platform provides a structure for software developers to build, deploy, and execute AI applications. The basic idea behind the pipeline is a *publish-subscribe* methodology for APIs (application programing interfaces). The purpose is to make clean, consistent data available to applications; think of it as something like a data supermarket.

After the data is aggregated, cleaned, refined, and processed, it is made available through consistent interfaces (the APIs), allowing applications to rapidly subscribe, sample what they need, test, and deploy. All this lets an agile development team build a new application in weeks, sometimes even days. Without these assets, a traditional IT custom-built process takes orders of magnitude more time and cost and becomes a nightmare to maintain and update. And in becoming an AI-driven company like Netflix, the idea is not to build one AI application. Rather, the idea is to build thousands,

FIGURE 3-3

A state-of-the-art data platform



Source: Keystone Strategy

of them—indeed enough to help make as many different types of predictions as possible.

Concurrent with investments in data and software are strategic investments in connectivity and infrastructure to integrate with the data platform. As we discuss in detail in the next chapter, most enterprises, even today, operate in separate silos. Even though customers view the enterprise as a unified entity, internally the systems and data across units and functions are typically fragmented, thereby preventing the aggregation of data, delaying insight generation, and making it impossible to leverage the power of analytics and AI.

Data platforms, and the organizations that work with them, should avoid siloed structures and instead should be designed in a modular fashion. The design of interfaces is critical in ensuring modularity in both code and organization. Clear interfaces therefore

allow for decentralized innovation at the module level; as long as there is a standard for the sharing of data and functionality, each module can improve its core function independently. APIs compartmentalize the innovation problem and enable independent agile teams or individual developers to focus on specific tasks without destroying the consistency of the whole.

Building a consistent (and secure!) data platform is even more important if the data is exposed to external partners. Taobao, Alibaba's online mall, is a good example, listing more than one billion items, all supplied by third-party providers. The only way for the company to satisfactorily share data with its internal and external users is through clear and secure APIs that enable the required range of functionality.

A typical internal Alibaba developer or external Taobao seller may be subscribing to more than one hundred different data platform software modules to enable them to upload inventory information, set pricing (manually or automatically), track consumer reviews, handle shipments, and the like. The development of well-designed APIs not only frees Taobao's engineers to keep developing and advancing internal systems to serve billions of users and millions of merchants but also unleashes creativity by an ecosystem of software vendors to offer a wealth of additional services.[16]

Finally, building a state-of-the-art AI factory with a well-designed data platform improves the organization's ability to focus on the crucial challenges of data governance and security. The massive amount of data that is increasingly captured from users, suppliers, partners, and employees is extremely valuable, sensitive, and private. It simply should not be stored in an ad hoc fashion. An organization needs to build a secure, centralized system for careful data security and governance, defining appropriate checks and balances on access and usage, inventorying the assets carefully, and providing all stakeholders with the necessary protection.

As part of the essential data governance challenge, carefully defining clear and secure APIs is essential to the AI factory. After all, APIs throttle the flow of data in and out of AI factory systems. Think of it as a way for the company to control all the data and functionality that it is willing to offer to internal and to external

developers. As such, APIs control access to some of the most critical and private assets within the organization. They force the company to define, ahead of time, which of these critical assets it wants to make available within the enterprise and which it may be willing to offer to anyone outside the company. The data that can flow through an API can make or break a digital company. The Cambridge Analytica scandal happened because developer and manager errors apparently caused a critical hole in the Facebook platform's graph API, allowing external application developers to access much more data than may have been originally intended by the company.

Ultimately, the data, software, and connectivity underlying an AI factory must reside within a secure, robust, and scalable computational infrastructure. Increasingly this infrastructure is on the cloud, is scalable on demand, and is built using standard off-the-shelf components and open source software. In addition, it needs to be seamlessly connected to the many individual processes and activities that constitute the company's operating model. Ultimately, these are the core digital processes that shape the delivery of value, such as creating, recommending, selecting, and delivering Netflix content, billing Netflix customers, or tracking the performance of Netflix content partners.

Building an AI Factory

You don't have to be Netflix to build an AI factory. The Laboratory of Innovation Science at Harvard (LISH), where we are faculty directors, in collaboration with colleagues from Harvard Medical School and the Dana-Farber Cancer Institute, demonstrated the development of an AI system that maps the shape of lung cancer tumors based on CT image scans. Deployed in only ten weeks and on an academic budget, the system is as good as a Harvard-trained radiation oncologist.

To develop the system, we leveraged the LISH AI factory, itself built to create a data pipeline and platform architecture for solving a variety of problems, usually with the help of crowdsourced algorithm design contests on Topcoder. LISH has partnered with leading

organizations like NASA, Harvard Medical School hospitals, Broad Institute of Harvard and MIT, and Scripps Research to take some of their toughest computational and prediction challenges.

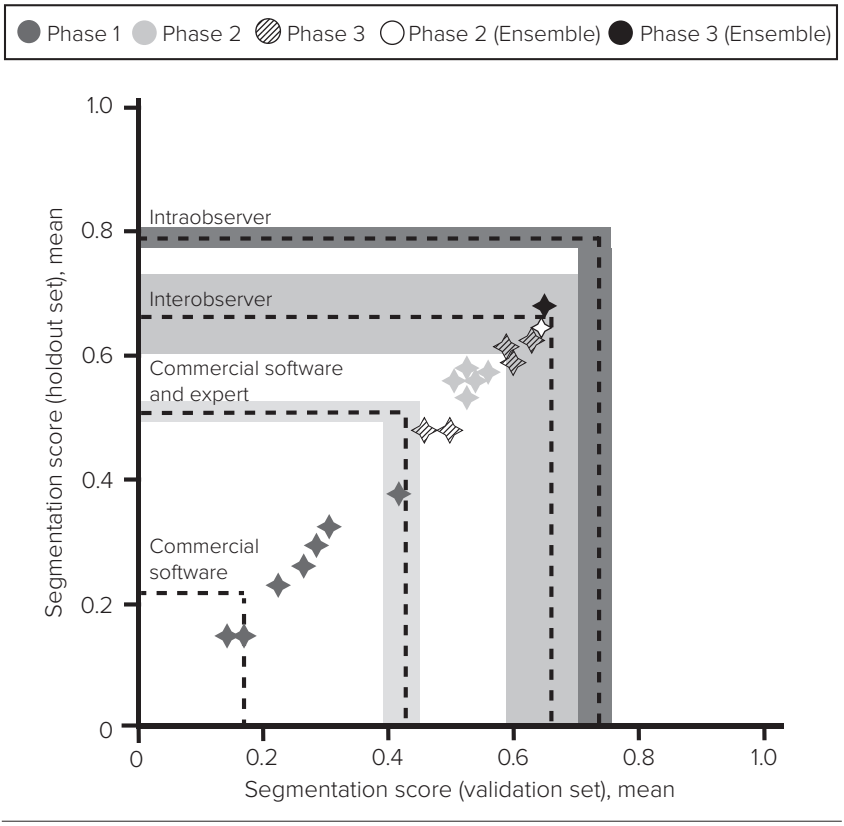
Outlining a lung cancer is critical in developing an effective therapy for patients. Oncologists therefore spend much time mapping the exact volumetric shape of any tumor that is to receive radiation therapy. Correctly outlining the tumor is particularly critical so that the therapy does not miss cancer cells or damage healthy tissue. The LISH team worked with Raymond Mak, from the Dana-Farber Cancer Institute, on the possibility of automating this task, leveraging data from 461 patients consisting of more than 77,000 CT image slices.

Using Dr. Mak's data, cleaned and prepared by our lab-based AI factory, two data scientists (physicists with no background in medical imaging) designed a series of contests to find the best algorithm to outline a tumor. We embarked on three sequential contests over ten weeks and had thirty-four contestants submit forty-five algorithms. We gave our contestants a "training" dataset consisting of scans from 229 patients, with the cancer fully outlined across the images by Mak. We held back the remaining dataset to see how accurate the algorithms would be in mimicking Mak's work.

The top five contestants used a variety of approaches, including convolutional neural networks (CNNs) and random forest algorithms. Surprisingly, none of our contest participants had any prior experience with medical imaging or cancer diagnostics. The solutions they developed involved both custom and published architectures and frameworks to perform the tasks of object detection and localization, with open source algorithms originally developed for facial detection, biomedical image segmentation, and road scene segmentation for research on autonomous vehicles. The phase 3 algorithms produced segmentations at rates between fifteen seconds and two minutes per scan—substantially faster than a human expert, who took eight minutes per scan. The ensemble of the five best algorithms performed as well as a human radiation oncologist (interobserver), and better than existing commercial software, as shown in figure 3-4.

FIGURE 3-4

Results of LISH analysis contest using data from the Dana-Farber Cancer Institute



We cite this example not only because we're proud of it but also to demonstrate that an organization doesn't have to be rich in data, IT resources, or AI talent to construct an AI factory. To create ours we tapped resources that are available to everyone. And the benefit we got from it is invaluable. We shared our findings in the *Journal of the American Medical Association Oncology*—not where you'd expect to find the work of business school faculty.[17]

We admit that it's relatively easy to tap the power of AI within a small laboratory. We did not have to deal with large, siloed organizations or complex, outdated, and mismatched IT systems. As AI

enables more of the operating processes in complex corporations, the way it is embedded and architected in the broader operating model becomes increasingly critical. This is why a firm's operating architecture has become a strategic consideration that should be thought through at the most senior levels. This is the topic of the next chapter.