AN EXCERPT FROM

PORTFOLIO PENGUIN

THE AI-FIRST COMPANY

HOW TO COMPETE AND WIN WITH ARTIFICIAL INTELLIGENCE

ASH FONTANA

PORTFOLIO | PENGUIN

INTRODUCTION

The animal kingdom is full of creatures with better senses than us humans. We can't see like owls or smell like dogs or hear like whales, nor can we do arithmetic faster than a computer—a creature of our own creation. But we are great at gathering a lot of *information* across our senses and processing it in parallel to learn fast. This—learning fast—is the definition of intelligence.

Now artificial intelligence (AI) can help us learn faster by gathering valuable *data*, processing it into information, and spreading it across networks. Companies that build AI increase their own intelligence. That's what this book is all about: building AI-First companies with extraordinary capabilities of gathering, processing, and communicating information to where it's needed, when it's needed—to learn faster than the competition.

AI-First companies were the first—and are still the only—trillion-dollar companies, and they will dominate more industries more definitively than ever before.

This is your guide to building an AI-First company.

THE NEW TOOLS

We like to make new tools and, by doing so, power through our natural limits. Charles Babbage, who arguably invented the first mechanical computer, said, “It is not a bad description of man to describe him as a tool-making animal.” We use our intelligence to learn how things work and build tools to make them work better—turning the gradual evolution of man into the fast revolution of man plus machine.

The first wave of tools, during the Stone Age, at least 2.6 million years ago, brought physical leverage. Think rope traps and spears. These allowed us to go beyond our immediate, physical reach to gather more food than we could with our bare hands. The industrial era of the eighteenth and nineteenth centuries brought us increasingly complicated sets of tools that worked in harmony to plow fields and mill steel. However, this physical leverage and scale were limited by our intellectual capacity.

The second wave brought intellectual leverage. The printing press distributed information, but then computers allowed us to gather more information and perform computations beyond our ordinary intellectual reach. We learned how to move information around, throw it into a calculator, quickly send it to different places, and generate multiple insights. However, this intellectual leverage was limited by how much information we could acquire about the future.

Artificial intelligence brings the third wave of tools that give us decision-making leverage. These tools are affording us an entirely new form of intelligence that gathers, processes, and communicates information to make better decisions. We’re learning better and faster as we see these decisions play out.

Let’s thread a needle: in the first wave, it was possible to stitch fabrics faster than we could weave by hand using looms full of little levers. Physical leverage. In the second wave, computers turned drawings—visual information—into patterns for the loom to automatically weave. Informational leverage. The third wave changes the game: computers scan photos posted on social media, figure out consumer trends, draw up new styles and turn drawings into patterns for the loom. New styles hit the stores right as they become fashionable. Temporal leverage.

Whereas earlier tools gave us physical and intellectual leverage, AI provides decision-making leverage that fulfills two important imperatives. The first applies to all of society: getting more out of our limited resources. The second applies to businesses: building a competitive advantage. Whereas land, labor, and capital were the resources leveraged to compete in the last millennium, information is now the resource to leverage. Whereas the leading companies of yesterday excelled at managing land, labor, and capital, tomorrow's leaders will excel at managing information.

There are many books about using land, labor, and capital to win at business. This book is about using information to win.

THE AI-FIRST CENTURY

The AI-First Century started in 1950 with the development of AI and will finish around 2050 with AI being fully deployed across all industries, so it's not too late for you to get started on applying AI in your industry.

The First Half of the AI-First Century: 1950–2000

If the first fifty years, from 1950 to 2000, were about getting AI to work in the lab, then the next fifty are about getting AI to work for people, business, and society. Understanding how it took the better part of a century to get to this point makes recent innovations seem even more remarkable and highlights our readiness to integrate AI with business strategy. We've thought AI was just around the corner for a long time, but it turns out that it was just getting started, and we are only now bringing working AIs into the real world.

Here is the story of the people who built AI, bit by bit, over the last half century. They are the ones who got us here, and you are the one who will take their work forward over the next half century with the help of this book.

Theoretical Foundations

The foundational thinking about AI happened in the 1950s. Individuals experimented, thought partners collaborated on equations, and groups of great thinkers formed to define artificial intelligence.

Mathematician Warren McCulloch and neurologist Walter Pitts wrote the first equations representing what happens in our minds. McCulloch was educated on the streets of Detroit; Pitts, at an East Coast private school. Both were founding members of the *cybernetics* movement: the science of control and communication in machines and living things. Walter was particularly prodigious, invited to study at England's Cambridge University when he was just twelve after mailing corrections to the great British philosopher and mathematician Bertrand Russell's *Principia Mathematica*, a three-volume work on the foundations of mathematics. He ran away from home when he was fifteen to visit Russell at the University of Chicago and never saw his family again.

Even though Warren was twenty-four years older than Walter, they spent a lot of time together across societies formed to understand the human mind and through institutions such as Massachusetts Institute of Technology (MIT). They also drank a lot of whiskey together. Their partnership yielded an important model known as the *threshold logic unit (TLU)*: a mathematical model of a human brain cell, or *neuron*, that explains how the brain computes things. This model of the atomic unit of our own intelligence provided the starting point for creating artificial intelligence.

Neuroscientists, computer programmers, and researchers from a wide range of disciplines formed groups to discuss how our brain works, and how to represent its functionality—first on paper and then in a computer program. These discussions borrowed concepts from logic, computation, neuroscience, communication, and much more. One group at Dartmouth College coined the term *artificial intelligence* in 1956; another at Cornell University created a *perceptron* algorithm that improved the ability of the earlier algorithms to mimic human neurons; and one at Stanford University joined these neurons together in an early, small version of an *artificial neural network*.

The fifties also brought us the first consideration of how to use AI outside the lab, with the great Alan Turing coming up with a test for intelligence based on reaching a human-level quality of conversation. All the while, Pitts was studying frogs. Among his discoveries: the fact that different parts of a nervous system each carries out a degree of computation. The decade was foundational for AI theory, but the practice was just getting started.

Progressing to Practice

In the 1960s, research on thinking progressed to sensing and speaking. Researchers at MIT, the US Defense Department's Defense Advanced Research Projects Agency (DARPA), and International Business Machines (IBM) studied how neurons move when sensing and reacting to stimuli in the environment, and they wrote computer programs to understand natural language, vision, and reasoning. Around this time, computers learned to play chess—an enduring fascination for the rest of the century.

Government funding for the field was cut in the seventies because developing a general purpose AI was deemed to be an intractable problem, prone to combinatorial explosion. This was more of a political than technological problem: the field overpromised and under-delivered. Corporations picked up the slack, developing the first robots, speech recognition systems, and language translators. Corporations anticipated AI intersecting with another trend: *process automation*. This trend was based on a factory management system developed in the late nineteenth century by an engineer named Frederick W. Taylor. The system increased efficiency by breaking down each step of a manufacturing process into specialized tasks, repeatedly performed by the same person. The trend continued into the twentieth century as specialized tasks could be written in computer code, and then repeatedly performed by computers. At this time, companies such as IBM and Systems Applications and Products in Data Processing (SAP), the German software corporation founded by a former IBM employee in 1972, started making lots of money selling computers to large manufacturing companies that would, for example, log inventory as it moved through a production process. The US military took this a step further to improve logistics: for example, using AI to allocate critical supplies across bases during the 1991 Gulf War against Iraq. Intelligent systems formed, bit by bit, as both companies and governments used AI to automate manufacturing processes and optimize supply chains.

Investment led to innovations in cars (1995), chess (1997), and cute dogs (1999). These breakthroughs were remarkable because they brought AI into the public consciousness, applying it, respectively, to a robot that changed its behavior based on what it could see; the first car to drive itself a thousand miles; the first chess program to beat a world champion; and a "pet" robot that responded to human expressions.

This period resembles what's happening today, twenty years later: evidence that AI works in the real world—both for the sake of entertainment and to make businesses more efficient—catalyzes investment. However, the investment in AI following the remarkable developments in the nineties didn't lead to many companies adopting AI. The military, IBM, and Sony were among the only entities with enough data, talent, and computing power to build AI at this stage.

These companies had a tremendous head start but were limited by the computing power and networking technology then available. Decades of research led to programmable artificial neurons. However, these programmable artificial neurons were singular and therefore had limited functionality: data in, compute something, data out. Stringing neurons together into large networks required prohibitive amounts of computational resources, which made the approach rather impractical. Architecting and running such a system remained a daunting challenge.

AI researchers made breakthroughs in stringing neurons together in a network at the start of the millennium. The Canadian computer scientist Yoshua Bengio devised a language model based on a neural network that figured out the next best word to use among all the available words in a language based on where that word usually appeared with respect to other words. Geoffrey Hinton, a British-born computer scientist and psychologist, developed a neural network that linked many *layers* of neurons together, the precursor to *deep learning*. Importantly, researchers worked to get these neural networks running efficiently on the available computer chips, settling on the chips used for computer graphics because they are particularly good at running many numerical computations in parallel. The result was a trainable neural network: programmable neurons, connected in a weblike network, passing the computations onto another web sitting below it—all computed on a chip that could perform the necessary operations on a reasonable timescale: mere days instead of months.

Computers Get Cheap

As tends to happen in the world of technology, an innovation in one field sparked innovation in another. The computer science of distributed systems enabled us to build very big, powerful, and cheap computers. This development was the first key enabler of AI. Large technology companies such as IBM and Microsoft gave their customers access to cloud computing clusters, and computing power became cheap as those clusters scaled up. The vast majority of people in the developed world got a computer to carry around. The data created while using these computers and the Internet was the second key enabler of AI. Researchers, equipped with more computing power and data, took a renewed interest in AI. These neural networks became deployable enough to embed in everyday products, from the keyboard on mobile phones, to shopping websites, to devices that sit in our homes such as the Amazon Alexa or Apple HomePod, ready for our every command.

I started working with AI-based technology around the time that researchers began taking an interest in it again, just after the establishment of cloud computing, when lots of data came into the world. Realizing just how much new data was flooding into the world and the challenges of making sense of it all, I started a company that helped the biggest travel companies in the world organize all the data that consumers generated with their phones and on social media. I was in awe of both the progress made and the novelty of deep neural networks. This was a technology that would change *everything* and was well worth spending decades understanding. Indeed, I joined the first investment fund completely focused on AI, Zetta Venture Partners, which was launched in 2013—the same year a whole *zettabyte* of data (10^21 bytes, or 1 trillion gigabytes) went across the Internet for the first time. We've reviewed tens of thousands of pitches to create AI-First companies, backed pioneering companies, and were the largest investor in the biggest community of *machine learning (ML) engineers* in the world: Kaggle. My job, as a venture capital investor, is to predict as best I can when the frontier of technology meets reality. I've bet my career that, for AI, that time is now. Zetta has since backed the first generation of companies applying AI to the real world, learning lessons as the "rubber hit the road." This book shares those lessons with you.

The Second Half of the AI-First Century: 2000–2050

The last few centuries have shown that a field moves forward when it intersects with another: philosophy and mathematics intersected to generate physics; when industrial engineering and electrical engineering intersected, factories leaped forward in terms of efficiency. Today artificial intelligence is intersecting with *distributed systems*, allowing us to run *machine learning (ML) models* over lots of data, on powerful computers. This book will provide best practices around investing in infrastructure and people that properly understand both intersecting technologies.

It's not too late to build an AI-First company. The rising tide of technology can help you catch up fast, whether you're steering a small boat or a large boat. Contemporary methods efficiently prepare data, let you do a lot with small amounts of data, synthetically generate data, efficiently label data, automatically build ML models, and integrate them with existing software. We will learn about many of these methods in this book.

THE AI-FIRST COMPANY

AI-First companies put AI to work, prioritizing it within real budgets and time constraints. AI-First companies make short-term trade-offs to build intelligence in order to gain a long-term advantage over their competitors.

Over these pages, you will learn answers to the most fundamental questions about running an AI-First company. From the starting points of experimentation and team building to the tactics of implementation, measuring value, and weighing long-term strategic advantage, you will learn what it takes to capture data, mold it for models, and learn faster than the competition.

This book fills the gap between theoretical writing about the potential of AI and technical writing about how to implement AI, offering readers an executable guide to applying AI to business problems. It is designed to be helpful for readers with different degrees of expertise. It offers a primer on AI, a glossary of key terms, and explanations of the component technologies to provide common understanding. The point is not to make you a technical expert but to give you a foundation for better business decision-making. Simply put, this book is about how to use AI to win at business. The question isn't whether you *can* contribute in the AI-First Century but whether you want to.

THE EIGHT-PART FRAMEWORK

The framework is organized into eight chapters, each one building upon the one before it.

1. We define *data learning effects (DLEs)*, the new type of competitive advantage that arises as we move from human

to machine intelligence. This new concept builds upon three existing concepts of competitive advantages: network, scale, and *learning effects*.

2. We show you how to get started today using the concept of *Lean AI*, a methodology for building AIs to test with customers.

3. Now that you've tested AI with customers, we flesh out your strategy for getting more of the data that models need.

4. We give you a plan for building an AI-First team that processes data.

5. With more data, you can build better models to compound the value of that data.

6. Moving from experimentation to production involves deeper integration with your customers' systems and constant management of the models. We give you frameworks for successful customer implementations and a systematic approach to managing AI models.

7. With everything in motion, we have to make sure AI delivers results. We give you qualitative and quantitative ways to measure the models, and the success of your AI-First company.

8. We explore how to aggregate data learning effects with other competitive advantages to build a company that beats *incumbents* and is defensible for decades to come.

	CHAPTER	LESSON
Strategy *What creates a competitive advantage?*	1. Defining Data Learning Effects	• Intelligence is determined by how fast you learn, and you'll learn faster using machines. • The automatic compounding of information is a data learning effect. • Data learning effects = economies of scale to data + data processing capabilities + data network effects. • Data learning effects compound faster than any other form of competitive advantage. • *Data network effects* are where each incremental data point adds more information to a user of the network than the last data point.
Tactics *Where do I start?*	2. Lean AI	• Anyone, at any company, can start building AI. • Lean AI is a process to build an AI-First product.
Capital *How do I allocate resources?*	3. Getting the Data 4. AI-First Teams 5. Making the Models	• The most significant source of data for AI-First companies is from the customers they're serving. • Data coalitions create a unique data asset for both vendors and customers. • AI-First companies need a diverse group of people to manage different technologies. • AI-First companies embed AI-First teams everywhere.

	CHAPTER	LESSON
Metrics *What should happen?*	6. Managing the Models 7. Measuring the Loop	• The world is always changing, so your models will too. • Model management is not code management. • The goal of tracking every version of a model is reproducibility. • The motion of a data learning effect is looping. • *Data learning effects* automatically generate assets, capabilities, and information. • Companies need new methods to properly account for the cost of delivering an AI-First product.
More *What's the second act?*	8. Aggregating Advantages	• *Vertical integration* gets more data, revenue, and profit. • Aggregating data creates new products. • Strategic data management can lead to customer lock-in. • Increase compatibility or create an ecosystem of third-party developers to contribute data to the ecosystem around a product. • Intelligent systems can analyze large amounts of product usage data to personalize products at scale. • *Intelligent applications* borrow from and then subordinate *legacy applications*.

Whether you're a new online retailer, own a family business, or manage a Fortune 500 company, you'll find what you need to compete and win with AI in this book.

Now, let's start building!

THE POWERS OF DATA LEARNING EFFECTS

Winners Take All with Data Learning Effects

Markets tip to a single technology when there are both high economies of scale and demand for variety. Economies of scale happen when cost decreases, on a marginal basis, as production increases. Variety is customer demand for lots of different features or products. Referring to the bottom-right quadrant of the table below, customers want variety when shopping online, and operating warehouses gets cheaper as they get bigger, so Amazon dominates e-commerce. Referring to the bottom-left quadrant of the table below, we want variety in our news sources, but journalists can write only so many articles per day—they don't scale well—so there are a few large newspapers.

WINNERS	LOW ECONOMIES OF SCALE	HIGH ECONOMIES OF SCALE
LOW DEMAND FOR VARIETY	Many	Few
HIGH DEMAND FOR VARIETY	Few	One

Only one company accumulates the most resources (like investment, ecosystem participants, among others) in a market, and the company with the most resources is the only one that can invest in building a wide variety of features for that market's customers. The company with the most scale can offer the most variety, so it dominates the market after this tipping point.

DLEs can have high economies of *scale* to data because they need lots of data to get started. DLEs create a wide *variety* of predictions because models are constantly generating different predictions based on feedback data. Further, customers of AI-First products inherently have a high demand for variety because each customer needs a model that can make predictions specific to their business. DLEs thus tip markets in favor of one winner.

Data Learning Effects Make Products More Useful

Data learning effects tend to work behind the scenes rather than front and center. Some products manifest their utility in the foreground and some in the background. Manifesting utility in the foreground means that the increasing utility is obvious to the end user, who sees that adding some data generates a more accurate prediction for them or immediately triggers a new insight. For example, adding their own sales data allows them to immediately see their sales benchmarked against their competitors' sales. Manifesting utility in the background means that increasing utility is not obvious to the user. The user doesn't see that adding some data generates a better prediction for them, and they don't see any information they don't already have.

One product that gets better in the foreground thanks to DLEs is Square Capital, the division of Square that lends money to customers of the company's point-of-sale (POS) systems (their cash registers). Merchants, such as restaurant owners, add data from their POS systems and get loans based on how much money they're making, almost immediately. Square is able to offer them that loan by comparing the data uploaded by the merchant to other merchants that previously received loans from Square. The user (the merchant) sees an immediate benefit of adding data (the offer of a loan) because the product utilizes a DLE to run a predictive system that delivers that product: an interest rate based on the prediction that the merchant will pay back the loan. You don't get a loan if you don't add data, and you qualify for a loan only because your data can be compared with other data to generate a prediction. This product is based on DLEs and gets better in the foreground.

One product that gets better in the background thanks to DLEs is Cloudflare, a website performance and security product. Customers such as news websites add a Cloudflare data collection mechanism to the network, which serves up page requests to the website's viewers and provides protection against requests from bad actors. Cloudflare offers that protection by comparing which requests were bad for other Cloudflare customers, with "bad"' meaning a request that tries to swamp the site with traffic in an attempt to shut it down (a *denial-of-service*, or DoS, attack) or otherwise exploit a security hole. The customer (the website owner) does not see an alert to deny a particularly dangerous request immediately after adding the Cloudflare data collection mechanism, but the product is constantly learning and delivering alerts to potentially bad requests. You don't receive these alerts if you don't allow Cloudflare to see your network requests. The product is based on DLEs and gets better in the background.

Data Learning Effects Compound Faster Than Network Effects

DLEs compound fast when they get data from customers. The fuel for the network effect is data, and customers provide this fuel for the system that ultimately benefits them.

Products with entry-level data network effects often get data customers, sometimes called a *give-to-get* model. Every customer contributes data that may make the product more useful, attracting more customers, and so on. High-quality data makes the product better, and customers can contribute some of that data.

Products with next-level data network effects often get data from customers that provide data as part of using the product, sometimes called *feedback data*, but—crucially—that data goes into a model that compounds the value of this data by transforming it into a prediction.

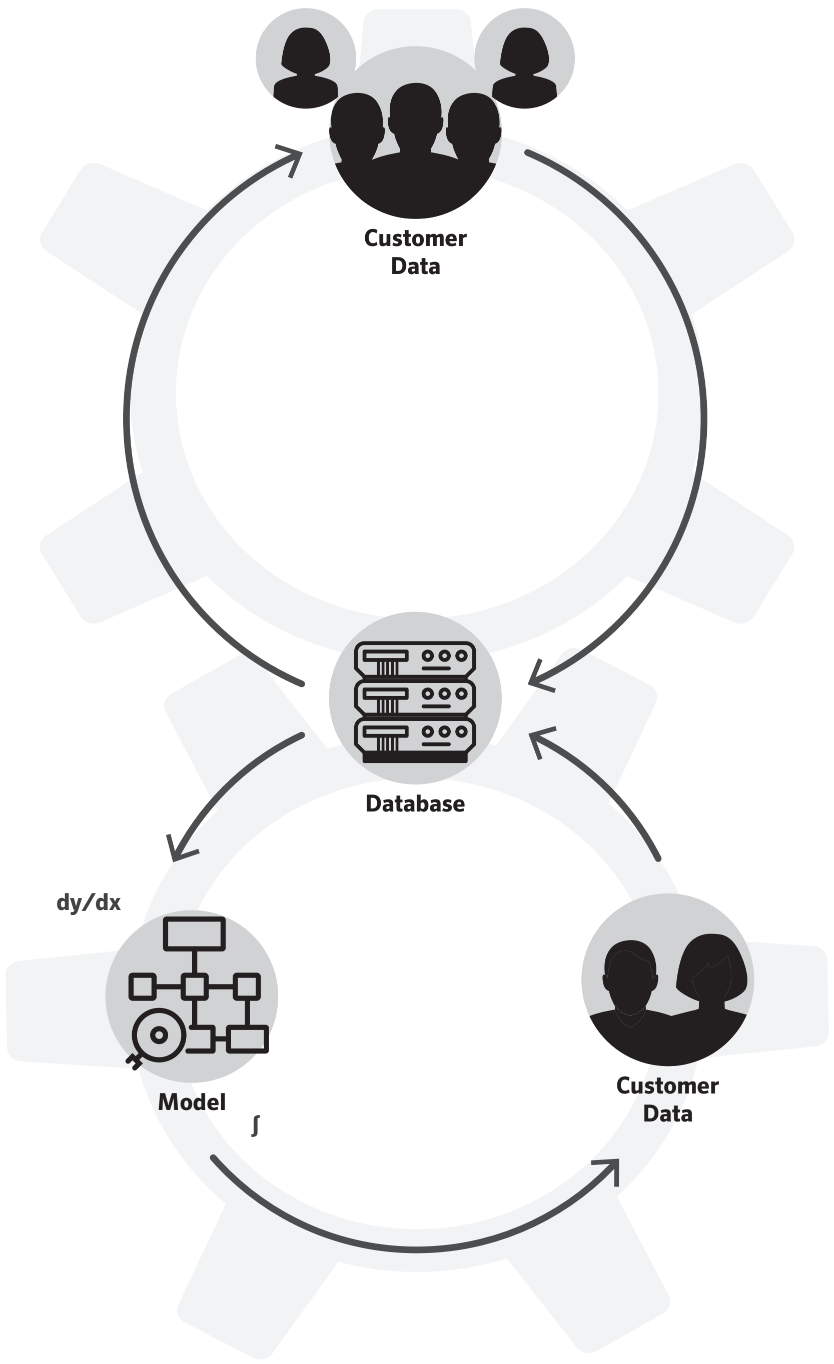
For example, customers submit data every time they get a credit score, and the credit scoring agencies use the data to provide more accurate scores (and then sell it to third parties). That's an entry-level data network effect. The next level would be a lending application where a customer provides ongoing access to their spending in order to receive loan offers. The company offering the loan can then predict if and when it may be paid back, decide on the customer's creditworthiness, and, accordingly, issue a bigger loan or cut them off. The difference between the two levels is the feedback data and the predictive model.

One Flywheel Kicks Off Another

Automating this process of collecting data and generating information is the start of the next level, when the shared brain starts comparing what it learns to other things it learns. This is the "flywheel" that kicks one network effect into another, more powerful network effect.

Amazon made this happen. First, it gathered a great deal of data on products and helped customers make better buying decisions by putting all of that data in the product listings, providing comparison tables with structured product information. More information meant better comparisons and decisions. Then Amazon invested in a team to build machine learned search and recommendation systems: A9. This team effectively got that product data and matched it with purchase data to learn which products customers want to buy so that Amazon could recommend similar products to those customers in listing pages and search results. Gathering a lot of data started the entry-level network effect: Amazon was the most useful shopping website to consumers because it had the most product information. Learning over that data kicked off the next-level network effect: Amazon is the most useful shopping website to consumers because it offers the best recommendations and has the best search experience.

ONE FLYWHEEL KICKS OFF ANOTHER



Data Learning Effects Drive Cost Leadership

DLEs automatically enable businesses to reduce the cost of making products while also increasing their value for customers. This allows for *cost leadership* by either charging less or charging the same but providing more value.

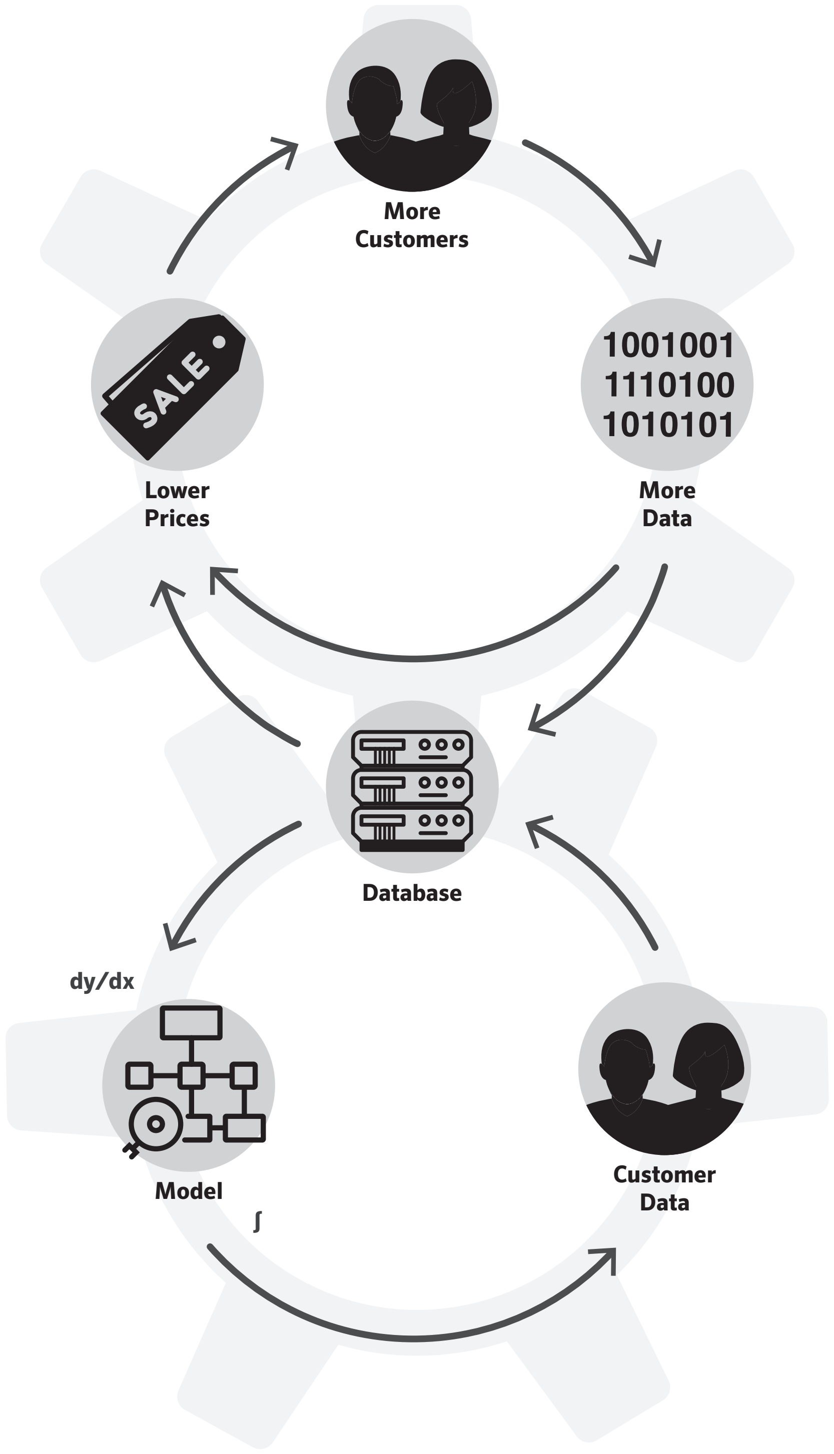
The initial cost of making products with DLEs amortizes over each additional customer. Building intelligent systems is expensive: buying, collecting, cleaning, and storing data. Hiring data scientists and ML engineers is also expensive. Usually a company has to spend a lot of money before customers see value in the predictions. However, expenses reduce once customers start using the product and contributing data. Customers might label or store their own data, or they might reconfigure the model's features through an interface. Customers effectively take on the cost of gathering and processing data through feedback systems, reducing production and maintenance costs, and allowing for cost leadership.

DLEs make products more valuable. More data empowers customers to make more accurate predictions and better decisions. Improved performance of the underlying models generates value for customers, meaning a higher return on their investments. The denominator (price) stays the same as value goes up.

$$\frac{\text{Revenues from Investment} - \text{Cost of Investment}}{\text{Cost of Investment}} \text{ x } 100 = ROI\ (\%)$$

Incidentally, cost leadership may help to quickly build a DLE. Cost leadership is a strategy to attract more customers, who, in turn, generate more data. This strategy may be deliberately employed to the point of unprofitability for a limited period of time to accumulate the critical mass of data required to improve an AI's accuracy so that it is useful to customers.

DATA LEARNING EFFECTS DRIVE COST LEADERSHIP



Data Learning Effects and Price Optimization

DLEs allow for more accurate pricing. Well-priced products sell more, and more customers gather more data, powering the DLE. Cost leadership is different from *price optimization*, and better pricing can boost DLEs.

Pricing is an information game that is won by figuring out exactly what someone will pay for something. Predictive systems are used in pricing experiments to generate test prices. One way to run a pricing experiment is to pick a price based on experience, observations, guesses, or other factors. Another way is to use data from previous experiments to predict what is probably an acceptable price. This second approach is more likely to find the optimal price.

E-commerce websites often personalize pricing. The price one shopper sees for a product is often different from the price another sees on the same website, or the price a shopper sees for a product on his phone may be different from the price for the same product when seen on his laptop. This is because the AI constantly runs experiments to see what customers may pay for something based on who they are (if logged in to a profile with previous purchases and demographic information), where they live, what they just clicked on, etc. Sophisticated shopping websites use hundreds of data points on each individual, from sets of data about customer preferences, and billions of data points on different groups of people to price products. These websites often have promotions that might even lose money—selling the product below cost—but which ultimately generate useful data about what customers might be willing to pay.

The *manual personalization process* may involve taking an observation, using it to inform the next price in an experiment, running the next experiment, making more observations, and so on. The *automated personalization process* may entail using ML by feeding data into a system that generates a price to test based on the gradient of the curve it thinks will arrive at the optimal price.

Airlines developed some of the first price optimization systems, commonly known as *yield management systems*, that priced each seat based on variables such as: the number of seats on the plane, the cost of flying that plane on the given route, the seat position, the expected demand given the time of year, and so on. These systems rely on the mathematical concept of linear optimization, the basis of some ML systems. Today these systems price seats, change the prices minute by minute, and distribute those prices to the myriad places to buy plane tickets.

There are also examples of predictive pricing for software products sold to businesses. The challenge is that there's typically less data available on what business customers will pay for a product because they make fewer bigger purchasing decisions and because their behavior when making those decisions isn't as easily observed as consumers browsing a shopping website.

Better pricing leads to more profit and boosting DLEs *if* profits are invested back into DLEs.

- Better pricing yields profit to invest in ML *research and development (R & D)*.
- Better pricing attracts customers, thus more data (at no cost), increasing profit, allowing further investment in R & D.
- Better pricing means less spending on sales and marketing, increasing profit, allowing further investment in R & D.

LIMITATIONS OF DATA LEARNING EFFECTS

There are limits to network effects, calling into question the assumption that utility constantly increases with the size of the network. Sometimes the limitation is external; for example, with regulatory action against the network owner as the product overlaying the network reaches a monopolistic position in the market. Other times the limitation is internal; say, the technology that serves the product around the network breaks at a certain scale. And sometimes the product generating the network effect just doesn't become more useful beyond a certain network size; for instance, your 510th friend on Facebook isn't as nice to have as your 50th friend on Facebook; you just don't engage with them, or the product, as much once the network grows beyond a certain size.

DLEs face limits, too. Sometimes the limits are external, as in the case of regulatory action against the owner of the data by the government to break up the owner's monopolistic position in the market. Sometimes it is internal; for example, if the data becomes too expensive to store and manage. This can even create a situation of decreasing returns to scale of data if adding a data point increases extraction, transformation, preparation, cleaning, loading, and storage costs. For example, adding more data to a customer relationship management system about each customer's preferences creates the need for more fine-grained labels to meaningfully classify the data. Adding data can also increase the required number of output dimensions, in other words, what the prediction must display. Sometimes the product generating the network effect doesn't have marginal utility with the addition of more information; for instance the three hundredth review of a coffee shop on Google Maps or Yelp isn't as useful as the third review; the aggregate score is more solid, but no one reads that three hundredth review. More data isn't always better.

DLEs have limits by virtue of acquiring new data, increasing the cost of building the product, or failing to generate new information. The value of getting more data is a technical and practical consideration: the extra data must make the model more accurate, and that increased accuracy must be useful to customers.

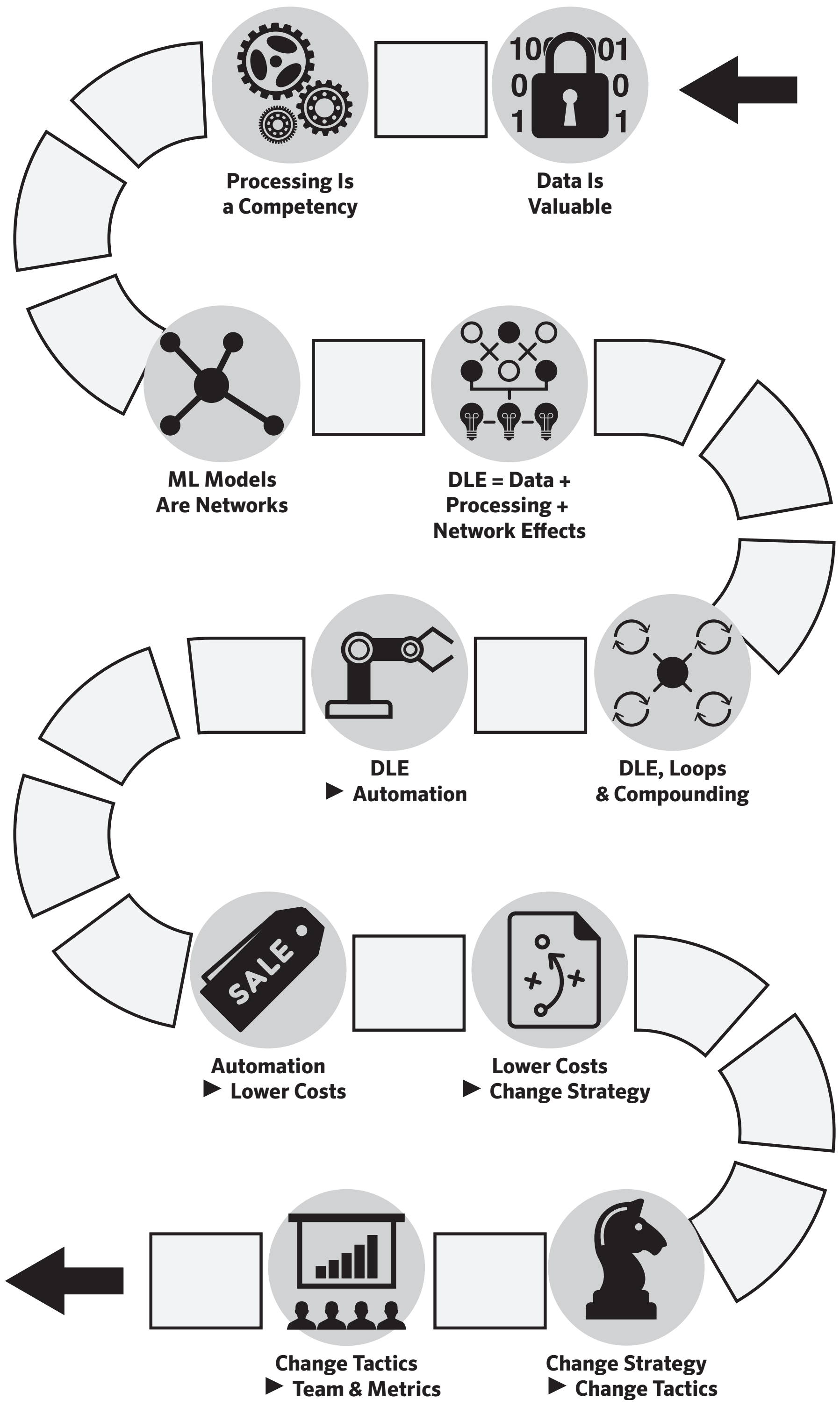
CONCLUSION: NOW WHAT?

We have vocabulary to describe how companies collectively and quickly learn from data. These companies make the leap from verbally to digitally distributing information, from manually to automatically sharing information, from learning in our own minds to learning across minds, from learning on wetware to learning on hardware, and from learning on one node to learning across nodes in a network. These are AI-First companies.

I've found that network effects just don't happen in AI-First companies the way they do in social networks: starting because one user ropes in others and escalating from there. Data network effects start only after working hard to gather and process data. That sometimes takes years and doesn't happen just by making the right technology choices. Highly capable people who know what data is where and why it's relevant to solving particular problems are crucial. What I've seen companies learn as a result of this hard work is powerful. When the DLE kicks in, not only do they attract more customers, but also they procure *proprietary information* about businesses that can be used to figure out how to automate core processes. This automation makes their customers' businesses more profitable, and this is what makes AI-First products stick.

The following path maps out the learning journey of the AI-First company. First, we learn about how to obtain and value data, then how to build an AI-First team to process that data, and then how to build the models that use that data to improve the product. This gets us a compounding source of competitive advantage in DLEs that can lead to a significant degree of automation. The question then becomes what to do with the cost savings that come from automation, which we cover in the final chapter of the book, "Aggregating Advantages."

THE DATA LEARNING EFFECTS JOURNEY


Data Is Valuable
Processing Is a Competency
ML Models Are Networks
DLE = Data + Processing + Network Effects
DLE, Loops & Compounding
DLE ▶ Automation
SALE
Automation ▶ Lower Costs
Lower Costs ▶ Change Strategy
Change Strategy ▶ Change Tactics
Change Tactics ▶ Team & Metrics

PLAYBOOK

- **The accumulation of information from data that automatically compounds is a data learning effect.** The steps to building a DLE are: (1) capturing a critical mass of data; (2) developing capabilities to process that data into information; and (3) feeding that information into a computer that runs calculations over data, learning from new data points.
- **Data generates marginal output when combined with data processing capabilities and data network effects.** Data learning effects articulate the value chain around data.
- **Data learning effects are unique.** They start with a supply-side competitive advantage that kicks off a demand-side competitive advantage and combines privileged access to a resource with capabilities to transform that resource into something valuable.
- **Data network effects are not like normal network effects.** The difference between a network effect and a data network effect is what's added to the network. With a network effect, edges are functional and communicate. With a data network effect, edges are informational and calculate.
- **Data scale effects are not just economies of scale to data.** More data doesn't automatically lead to a data network effect, but it may lead to a scale effect, whereby costs of processing or acquiring data decrease as scale increases.
- **There are two types of data network effects: entry and next level.** The entry level sees the addition of data producing a positive effect, but the next level is when the addition of data plus AI causes the positive effect. The first are easier to build, but the second grow faster.

- **Next-level data network effects generate data for free.** Entry-level effects need data from outside the network. Next-level effects generate data from inside the network (feedback data) leading to a high rate of compounding the value of data assets.
- **Better intelligence forms from self-generating data networks.** Entry-level data network effects augment our own intelligence. Next-level data network effects enable faster learning.
- **Winners take all with data learning effects.** AI-First products have high economies of scale to data and high demand for variety between customers such that these products can dominate an industry after a tipping point.
- **Data learning effects make products cheaper.** DLEs allow for cost leadership by reducing the cost of acquiring and processing data while delivering more value to customers.
- **Data learning effects make products easier to price.** Build a separate, intelligent system to run price experiments and optimize pricing across customer segments.
- **Data learning effects have limitations.** DLEs are limited by scaling limits of surrounding technology, data storage costs, lack of marginal utility to customers, and regulatory action.

PRE-ORDER NOW

amazon

Get it on Apple Books

BARNES&NOBLE

audible an amazon company

Bookshop

Porchlight

For orders of 25+ copies

ashfontana

PORTFOLIO PENGUIN