Zero to AI

A nontechnical, hype-free guide
to prospering in the AI era

Gianluca Mauro and Nicolò Valigi

Manning
Shelter Island

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Development editor: Lesley Trites
Technical development editor: Danny Vinson
Review editor: Ivan Martinović
Production editor: Deirdre Hiam
Copy editor: Sharon Wilkey
Proofreader: Melody Dolab
Technical proofreader: Andrew Harmor
Typesetter: Gordan Salinovic
Cover designer: Marija Tudor

ISBN 9781617296062
Printed in the United States of America

brief contents

contents

acknowledgments

Writing this book has been hard. Very hard. In a world where the internet encourages people to favor quantity over quality and push content fast and often, we decided to take the opposite route. We invested thousands of hours to put together what we believe is the best nontechnical guide for people to understand and start using AI.

It was a gigantic effort, and we'd like to thank the people who helped us in this journey.

Thanks to the Fulbright BEST commission, which offered both of us the scholarship that allowed us to meet and spend six unforgettable months in Silicon Valley.

We'd like to thank everyone at Manning who provided invaluable guidance (and sometimes just plain hand holding) during our first adventure as authors. Specifically, thanks to development editor Lesley Trites, review editor Ivan Martinović, technical development editor Danny Vinson, technical proofreader Andrew Harmor, project editor Deirdre Hiam, copy editor Sharon Wilkey, and proofreader Melody Dolab.

Thanks also to all our reviewers: Alain Couniot, Alan Matcham, Antonio Moro, Arun Pavuri, Clemens Baader, Davide Cadamuro, Francesco Catarinozzi, Harro Lissenberg, James Gray, John Montgomery, Lisa Örtegren, Marc Falk, Marianna Ricci, Mario Grassi, Michael B Marcus, Michael Modenese, Neil Croll, Nelson Wong, Nick Vazquez, Peter Hampton, Peter Trydeman, Peter White, Ravi Sajnani, Ritwik Dubey, Shiloh Morris, Sune Lomholt, Todd Wade, Tomasz Struminski, and Vitosh Doynov. Your feedback has been of immense help. We sincerely believe that without you, we wouldn't be as proud of this work as we are now. And apologies if our first drafts left something to be desired.

From Gianluca: Thank you, Sofia, for being my partner and my friend. Your relentless passion for life inspires me to live mine to its fullest, joyful and humbled by knowing I have a special place in your heart. Grazie alla mia famiglia. Sapere di poter contare sul vostro supporto e amore incondizionato è il mio porto sicuro dalla quale far partire ogni mia avventura, passata, presente e futura.

From Nicolò: Thank you, Angela, for being the complementary half of my personality. Your energy, passion, and dedication have filled my life with optimism and challenge, inspiring me to become a better person than I could ever hope to be. Grazie alla mia famiglia per essere stata una roccaforte di stabilità, un esempio di amore, e un fuoriclasse di pazienza con il sottoscritto.

about this book

Zero to AI is designed to help you understand how you can use AI in organizations small and large, for-profit and nonprofit. We want this to be a one-stop solution and leave you with the confidence you need to start using AI in your organization. To help you get there, we divided the book into two parts with two different goals:

- Part 1 of the book is about the technology, its core principles, and how companies have used it to build amazing products. At the end of this part, you'll understand what AI can and can't do, and know the vocabulary to communicate effectively with technical people.
- Part 2 focuses on organizations and value creation. We'll share with you the strategies we use in our consulting practice to select, design, and build successful AI products.

Who should read this book

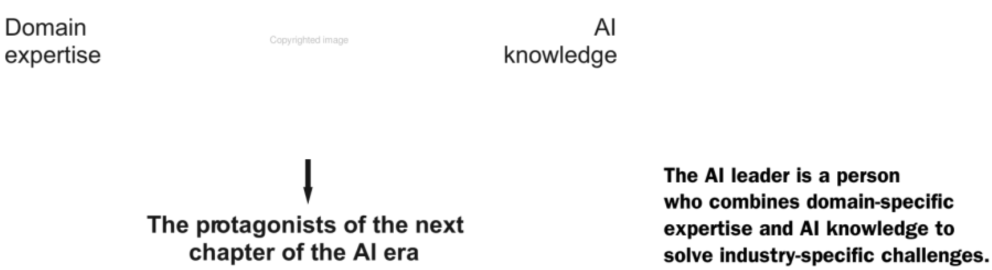
Three groups of people benefited from the recent spate of improvements in AI: tech entrepreneurs, the venture capitalists who showered them with money, and the few sought-after AI experts who saw their salaries balloon to seven figures. If we had started writing five years ago, we would have written a technical handbook with these people in mind.

Today, we think that techies have had their share of success, and it's time to let the next class of professionals join the AI revolution. The protagonists of the upcoming chapter of the AI era won't be interested in building AI applications for the sake of technological progress. They're not computer science or mathematics gurus.

They're experts in a specific industry who want to use AI as a tool for solving real-world problems.

Some of these future protagonists work for large corporations. It doesn't matter if their business card says *CEO*, *manager*, or *intern*. What matters is their drive to strengthen their careers by helping their organization be competitive in these fast-changing times. Others work for smaller companies and want to see them grow and create new products and services. Yet others are entrepreneurs, looking for the "next big thing" to build. And let's not forget about students and fresh graduates who want to develop unique skills.

In our experience as consultants and engineers, we met many such people who aspired to be AI leaders (see figure 1). We did our best to give them what they needed: a clear understanding of what AI is, what it can do, and how it can be used to bring value to their organizations. We wrote this book because we want you to join the crew of revolutionaries.



How this book is organized

Because you'll need two fundamental sets of skills to bring AI into your organization, this book is also divided into two parts covering 10 chapters.

Part 1 is about *understanding* AI. Because modern AI is built on data, each chapter in this section introduces you to different types of data and the AI tools best suited for each:

- Chapter 1 is a brief introduction to the history of AI and the innovations that sparked the AI revolution of the 2010s.
- Chapter 2 is about the data produced by core business operations, and how AI can be used to build unique products and services on top of it.
- Chapter 3 takes a deeper look at AI applications for sales and marketing.
- Chapter 4 introduces AI models that can understand, produce, and transform media such as images, video, and audio.
- Chapter 5 presents AI algorithms for understanding and producing written text.
- Chapter 6 demonstrates models for recommending personalized content to humans.

Part 2 is about *building* AI. This part is meant to be your guide to design and build new projects within your organization:

- Chapter 7 describes a framework for identifying opportunities for AI in your organization and selecting the best ones.
- Chapter 8 discusses the challenges involved in building AI projects, from collecting the right data to recruiting an effective team.
- Chapter 9 is about implementation strategy. It presents the trade-off involved in building or buying technologies and a lean approach to minimizing risk. It also covers strategies for managing AI projects and incrementally improving them.
- The book ends with chapter 10, which offers a final broad view of how AI can impact society.

We recommend reading the chapters in order, as they all build on each other to offer you a full understanding.

liveBook discussion forum

Purchase of *Zero to AI* includes free access to a private web forum run by Manning Publications, where you can make comments about the book, ask technical questions, and receive help from the author and from other users. To access the forum, go to https://livebook.manning.com/book/zero-to-ai/discussion. You can also learn more about Manning's forums and the rules of conduct at https://livebook.manning.com/#!/discussion.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers, and between readers and the authors, can take place. It is not a commitment to any specific amount of participation on the part of the authors, whose contribution to the forum remains voluntary (and unpaid). We suggest you try asking the authors some challenging questions lest their interest stray! The forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

Other online resources

After finishing this book, you might want to continue learning about two main areas. You may want to deepen your knowledge about the technical aspects of AI that we cover in part 1 and start building some AI projects. In this case, you can choose from a variety of online courses and materials. Two of the most widely known are Andrew Ng's Machine Learning and Deep Learning courses, available on Coursera. Both include programming assignments and will give you a strong foundation in the math and implementation issues behind many well-known algorithms. Several universities also offer some of their ML courses online, complete with video lectures and homework assignments. We can recommend Stanford's CS231 course about Deep Learning for Computer Vision applications (covered in chapter 4) and CS224N about Deep

1 An introduction to artificial intelligence

This chapter covers

- Gaining perspective about the history of artificial intelligence
- Understanding machine learning and its relationship to AI
- Exploring the drivers of the explosion in AI applications

Artificial intelligence (AI) is not a new technology. For decades, computer scientists have tried different approaches to reach the holy grail of computing: intelligent machines. While we are still far away from replicating the wonders of the human brain, AI applications have started to fill our daily lives and power our electronic devices, from smartphones to home alarm systems.

Why this seemingly sudden explosion? This chapter will answer this question by teaching you about modern AI—including the core principles behind it, and how and why we got to where we are now.

1.1 The path to modern AI

As humans, we've always tried to find ways to understand the world around us and bend nature to meet our goals. To do so, we have always relied on external tools that amplify our brain's capabilities.

The abacus was probably the first such tool, invented about 5,000 to 6,000 years ago to help people make calculations. Although it's still used in schools to help children visualize simple mathematical operations, it doesn't really save us from the labor of actually performing them. We had to wait until the 1960s for the first machines that could add and subtract numbers automatically. Computers have come a long way since then, but deep down their capability has still been pretty simple: executing calculations exactly as some (expert) human has instructed them to do. There's little "intelligence" in them.

The two words *artificial* and *intelligence* were first put together on August 31, 1955, when professor John McCarthy from Dartmouth College, together with M.L Minsky from Harvard University, N. Rochester from IBM, and C. E. Shannon from Bell Telephone Laboratories, asked the Rockefeller Foundation to fund a summer of research on artificial intelligence. Their proposal stated the following:

> *We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. . . . An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.*

The researchers knew that tackling intelligence as a whole was too tough of a challenge, both because of technical limitations *and* the inherent complexity of the task. Instead of solving the broad concept of intelligence, they decided to focus on subproblems, like language. Later, these applications would be called *narrow AI*. An artificial intelligence capable of matching or surpassing human capabilities would instead be called *general AI*. In other words:

- *General AI* (or *strong AI*)—An artificial intelligence program capable of tackling every kind of task it's presented. This is similar to an extremely resourceful human, and you can think of it as the robot from *The Terminator* (or, hopefully, a more peaceful version of it).
- *Narrow AI*—An artificial intelligence program capable of solving a single, well-defined task. It can be broad (recognizing objects from pictures) or extremely specific (predicting which customers who bought product A are more likely to purchase product B as well). This means one task at a time, and not any other: an AI that recognizes cats in images can't translate English to Italian, and vice versa.

General AI is still far away: researchers still don't know when we'll finally get it. Some argue that we'll never get there. Even though general AI is still a distant, fuzzy dream, this is what many people have in mind when AI is mentioned in the news. If you were one of those people, and are now disappointed that general AI is not here yet, don't despair. Narrow AI applications are still capable of creating immense value. For example, AI that can detect lung cancer is a narrow application but nevertheless extremely useful.

The results of the Dartmouth research summer of 1956 were so interesting that they sparked a wave of excitement and hope among the participants. The enthusiasm of the scientists spread to the US government, which started heavily funding research on a specific application: English/Russian translation. Finding trustworthy Russian translators must not have been easy in the midst of the Cold War.

After the first few years of work, a government committee produced the infamous 1966 Automatic Language Processing Advisory Committee (ALPAC) report. The document featured the opinions of many researchers about the state of AI research. Most were not very positive:

> *Early machine translations of simple or selected text . . . were as deceptively encouraging as "machine translations" of general scientific text have been uniformly discouraging. . . . No one can guarantee, of course, that we will not suddenly or at least quickly attain machine translation, but we feel that this is very unlikely.*
>
> *. . . there is no immediate or predictable prospect of useful machine translation.*

The ALPAC report marks the beginning of a period called the *first AI winter*: public funding for AI research stopped, excitement cooled, and researchers focused their work on other fields.

Interest in AI faded until the 1980s, when private companies such as IBM and Xerox started investing in a new AI spring. New hopes were fueled by a technology called *expert systems*: computer programs that encode the knowledge of a human expert in a certain field in the form of precise, *if-then* rules. An example will help you understand how expert systems were designed to work.

Suppose you want to build an AI system that can stand in for a gastroenterologist. This is how you do it with an expert system: you ask a doctor to describe with extreme precision how they make decisions about patients. You then ask a programmer to painstakingly transform the doctor's knowledge and diagnosis flow to if-then rules that can be understood and executed by a computer. An extremely simplified version would look something like this:

> *If the patient has a stomachache and the body temperature is high, then the patient has the flu.*
>
> *If the patient has a stomachache and has eaten expired food, then the patient has food poisoning.*

And so on. Once the doctor's knowledge is encoded into the software and a patient comes in, the software follows the same decision path as the doctor and (hopefully) comes up with the same diagnosis. This approach has several problems:

- *Poor adaptability*—The only way for the software to improve is to go back to the drawing board with a computer scientist and the expert (in this case, the doctor).
- *Extreme brittleness*—The system will fail in situations that weren't part of the original design. What if a patient has a stomachache but normal body temperature, and hasn't eaten spoiled food?
- *Tough to maintain*—The complexity of such a system is huge. When thousands of rules are put together, improving it or changing it is incredibly complicated, slow, and expensive. Have you ever worked with a huge Microsoft Excel sheet and struggled to find the root cause of a mistake? Imagine an Excel sheet 100 times bigger.

Expert systems were a commercial failure. By the end of the 1980s, many of the companies that were developing them went out of business, marking the beginning of the *second AI winter*. It wasn't until the early 2000s that the next generation of AI successes came along, fueled by an old idea that became new again: machine learning.

1.2 The engine of the AI revolution: machine learning

The first definition of *machine learning* dates back to 1959, from American AI pioneer Arthur Samuel:

> *Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.*

The key elements here are *learning* and *without being explicitly programmed*. Let's focus on the latter first. Explicitly programming a computer means defining the rules and instructions it must follow to perform a specific task. This is what software engineers do when they write software that handles your everyday tasks like doing taxes or filling out spreadsheets.

People without programming experience often feel like software engineers are powerful creatures who can bend machines to their will. Unfortunately, things are not always that easy. Try to think about the various decisions you make as you perform some trivial actions: can you explain the process you follow to recognize your friends when you see them? All the split-second decisions you make while driving? Can you list all the English grammar rules you apply as you talk? If you can't precisely explain how you do something, there's no chance that you can instruct a computer to do it.

Samuel proposed to replace "instructing computers" with "giving them the ability to learn." If you think about it, learning instead of following instructions is (coincidentally?) what human beings do all the time. Our mothers and fathers don't teach us their native tongue by giving us grammar books at the tender age of one. They just speak to us naturally, and we learn from their example, applying thousands of grammar rules without even knowing it. In fact, our brain is capable of automatically extracting rules way before it becomes capable of rationally understanding grammar at school! Even for us humans, it looks like learning rules from examples can be easier than being told about them.

temporary label to a piece of software that does something cool and surprising, until we get used to it." We don't know about you, but that just doesn't feel like a satisfying definition.

We hope we have convinced you that it is extremely hard to find a definition that makes everyone happy and can be valid as technology evolves. With the AI effect in mind, we decided to avoid a narrow definition of AI that rewards "flashy" applications just to ditch them once the hype is gone. We embrace a broader definition that includes less flashy applications. This is our definition of AI:

> *Software that solves a problem without explicit human instruction.*

As you can see, our definition focuses on the outcome of the technology rather than the specific techniques used to build it. Some people will not agree with it, because it's almost equivalent to what we said about machine learning earlier in the chapter. The truth is, learning *is* an intelligent trait, and while ML is just a tool, it is *the* tool behind 99% of the successful applications we happen to call AI today. This may change in the future, but we don't see any new approaches on the horizon that hold the same promise as ML. This is why every AI application we'll cover in this book is based on ML: it's simply the most accurate picture of the AI landscape of today and the near future.

We now have a clear view of what ML is, a working definition of modern AI, and some perspective about how these terms evolved. We are just missing the third buzzword you've probably heard about: data science.

Data science (*DS*) is a broad, multidisciplinary field that uses scientific methods and processes to analyze data and extract insights. ML techniques are some of the tools in the DS toolbox. In practice, when people refer to a *data science project*, they often mean something *static*: extracting insights from data and presenting them as a presentation or report. On the other hand, AI is more commonly used in the context of live software.

For instance, analyzing traffic data to design a new urban plan for a city to minimize congestion likely falls into the realm of data science. However, if you use the same data to control traffic in real time and direct cars through less-congested routes, most people would say the project is about AI. In the first case, the output of your project is a report, and in the second, it's "live" software that runs 24/7. Keep in mind that this division is mostly conventional: there really are no hard-and-fast rules about what's AI and what's data science. Table 1.1 summarizes the differences as we see them.

Table 1.1 The main differences between AI and data science

Artificial intelligence	Data science
Automates tasks or predicts future events based on data.	Produces insights based on data.
Is commonly used "live": it continuously elaborates new data and produces answers.	Is commonly "one-off": it produces some insights that inform decisions.
It commonly has the form of software.	It commonly has the form of a presentation or report.

Hopefully, these sections helped demystify some commonly misunderstood terms and created context for these technologies. Now you can start learning the core principles of AI, what you can potentially do with it, and how to bring this transformative technology into your organization. In the next section, we'll explain the steps of this journey and how this book guides you through them.

1.4 Our teaching method

If you want to productively use AI in your work life, it's paramount that you understand its nuts and bolts first. We noticed that nontechnical people who approach AI without a solid understanding of its principles often end up dreaming about projects that are simply impossible to build, or miss low-hanging fruit that could be easily tackled. After the first part of the book, you'll know all the AI principles you need to avoid these dead ends and get the best out of the technology.

Even after just this first chapter, you already understand that virtually all modern AI applications rely on machine learning, and machine learning is all about learning from data. This is why we used data as your guide to understanding AI. Each chapter of the first part of the book focuses on one specific kind of data, showing you how to spot it in your organization, what you can do with it, and how it fits into the world of AI.

Each chapter in part 1 uses a toy example to introduce the ML concepts you need. We found this to be the most efficient way to teach ML concepts that would otherwise be too dry and abstract. We didn't dig deep into technological aspects for two simple reasons:

- Technology changes so rapidly that implementation details would soon become obsolete.
- Simply put, you don't need it. Unless you want to pivot your career to writing code, we believe there's more value in adding AI to your wealth of knowledge and letting someone else practically implement your vision in computer terms.

This doesn't mean that we'll completely spare you from technicalities. From our experience as engineers, we know that it can be difficult for your technical team to communicate with people without the smallest bit of technical understanding. We don't want them to have trouble talking to *you*, so we made sure that you'll be learning the most important technical aspects of AI tools. Even as you leave them in the hands of your team, knowing about them will help you plan and manage the efforts.

Each chapter includes one or more real-world business cases about companies that achieved extraordinary results. To the extent that we mention specific companies, products, or services, keep in mind that we do so because we want you to develop awareness, but you shouldn't feel limited to them in any way. We have no affiliation or stake in any of the companies in the case studies; it just so happens that they're building great products we can all learn from.

When presenting cases, we followed a methodology inspired by the Harvard Business School case method: we'll first present the case in the most neutral way possible, and ask you open-ended questions at the end. Right after that, we include our thoughts about these questions and prompts for further discussion. We recommend you don't read these answers right away, but rather try thinking about how *you* would answer based on your knowledge and what you've read in the case, and only then read our take. Be aware that there's no unique solution to the questions we asked: if you found an interesting take on the cases that we didn't include in the answers, good job! This means you've learned what you needed and are able to extract insights on your own (so if that happens, we reached our goal with this book as well).

Summary

- AI has had a long history of successes and failures that dates back to the 1950s.
- General AI is the pipe dream of having an all-knowing machine. All AI applications we have today are instead narrow; they focus on specific tasks.
- Machine learning is the prevalent way to implement AI today, and is based on letting machines learn autonomously from data.
- Data science is related to AI and ML, but focuses more on extracting insights than persistent intelligence.

Part 1

Understanding AI

This part of the book focuses on the core principles of modern artificial intelligence. By the end of part 1, you'll know what AI can do for the different kinds of data your organization deals with. Perhaps more important, you'll also become familiar with what AI cannot do yet.

Today's AI revolution is based on training computers to learn from data, and that's why we have decided to organize this book based on the various shapes and forms that data can take. Every chapter focuses on a specific kind of data and uses a simplified example to help you learn key concepts about AI. At the end of each chapter, you'll find case studies from real companies that have used the technologies and data we talked about to achieve astonishing results.

engine of the organization that successful projects are almost guaranteed to make a dramatic impact.

To whet your appetite, let's briefly mention the two case studies you'll find at the end of this chapter. The first one is about an initiative of Jim Gao, an ex-Google employee in charge of operations at its data centers. He looked at the data collected from the large air-conditioning systems used to cool Google's gigantic computers and thought about using machine learning to optimize their consumption. The result was a 40% bill cut for the tech giant. The second case is about Square, a payment services company based in San Francisco. Square had been processing credit card payments for small businesses and realized that all the data it was collecting could be used to offer customized and low-risk loans to small businesses. By using machine learning models, Square created an entirely new business line and tackled the blue ocean of small business loans. The quality of its service is unmatched: by automating the lender-vetting process, Square can deposit loans to a customer's bank account just one business day after the request is submitted. As a result, Square loaned more than $3 billion in four years, with exceptionally low delinquency rates of 4%. By the end of this chapter, you'll know how these companies made these changes possible.

2.1 Unleashing AI on core business data

We define *core business data* as "data with a direct impact on the top or bottom line of the organization." Core data looks very different depending on what your organization does: cart history for an e-commerce operation, physical measurements for an engineering organization, and patient behavior for a health-care company. Regardless of its form, core data is valuable because it describes events and patterns that have a direct impact on the organization's performance, and it's easy to attach a monetary value to it.

We already mentioned the two case studies that await you at the end of this chapter. Let's look at why the data used by the two companies is core business data:

- A company like Google relies on massive data centers to offer its services (process web searches, store photos, route emails, and so forth). Probably the only variable cost for Google is the energy spent to keep the data centers' computers cool. Therefore, data about its cooling plants is core to Google's business, as it correlates directly to one of its main costs.
- Square's core product is its point-of-sale (POS) solution. Through that product, Square processes all the payments going to its customers. Because Square's mission is empowering small businesses, its transaction data is strictly linked to that vision and is valuable for its customers.

A good way to look at the value of data is to think about a metric we'll call the *dollar density of data*: how much the data influences the top or bottom line of the organization. Core business data has a high dollar density: each e-commerce order, job lead, or financial transaction has a direct impact on your top or bottom line. As you move away from the core value proposition of your organization, the dollar density of the data you collect decreases accordingly. Recording visits to your website can be valuable, but not nearly as much as tracking orders coming in. For a hospital, the call center data could be useful, but not nearly as much as patient records. This is why we have decided to start the book with the highest-dollar-density data of all: your core business data.

Often we see that the core business data takes a structured form, just like the tidy rows and columns of a Microsoft Excel spreadsheet. In engineering terms, we call this type of data *structured*. Other examples of structured data are weather reports, measurements of physical processes, financial transactions, most markets, and supply chain and warehousing metrics. As a rule of thumb, anything that you can load in Excel is likely to fall within the structured data umbrella. Other types of data are harder to fit into neat Excel columns: think about pictures, voice recordings, or text in a book.

It's important to understand that the same information can exist in both structured and unstructured forms. For example, consider the following way of recording a medical diagnosis:

> *The patient Gianluca Mauro has a severe inflammation of the shoulder joint; the therapy is to take two pills of Cortisone per day for five days.*

The same information can be recorded in a structured way, as in table 2.1.

Table 2.1 A medical diagnosis in structured form

Patient	Diagnosis	Area	Medication	Frequency (times/day)	Therapy length (days)
Gianluca Mauro	Severe inflammation	Shoulder joint	Cortisone	2	5

The information is the same—but in the first case, it's represented in an unstructured way (text), whereas in the second, it's structured. Structured data is much easier for computers to process than unstructured data. Later chapters cover AI techniques for unstructured data, including images and written language. For now, let's stick to structured data, happily knowing that most core business data falls into this category.

2.2 Using AI with core business data

Now that you know how to find and recognize core business data, let's look at what AI can do with it. Because we believe in the teaching power of stories, we developed a simplified example that's going to keep you company throughout this first part of the book.

FutureHouse is a fictional business that operates an online real estate marketplace where homeowners can advertise their homes for sale and hopefully attract interested buyers. FutureHouse has always prioritized customer service and employs agents who can offer their support to help sellers assess their house price. Buyers use the site to look for the house of their dreams.

We chose this example because it will be familiar to many readers, while giving us the opportunity to explore many aspects of the AI landscape. A nice bonus is that the housing market is a typical example in the ML literature. Should you decide to delve deeper into the technicalities of ML and maybe write your own code, you'll find plenty of references on the web.

We'll start by explaining a bit about how housing markets work, and then we'll introduce an application of ML to this business. We're going to use this description as an excuse to introduce key terminology and concepts, and conclude by extending these concepts to other, more general applications.

2.2.1 *The real estate marketplace example*

Real estate agents are at the center of housing markets. They connect sellers and buyers and help them figure out the right price for their property. Figure 2.1 shows how the typical home-sale transaction unfolds under the careful watch of a realtor:

1. A customer comes in looking to sell their house. Before they put it on the market, they want a professional opinion about how much it's worth.
2. The realtor checks out the house and sets a price based on its square footage, age, included appliances, other offers in the neighborhood, and so on.
3. The house gets listed.
4. Potential buyers find the listing and come in for an open house event. One lucky buyer eventually gets to buy the property.

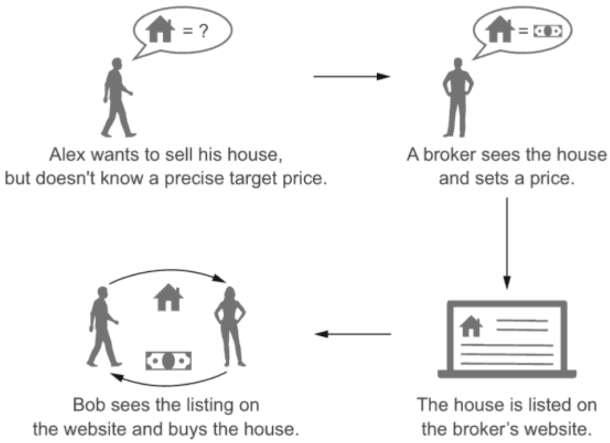


Figure 2.1 The process of listing and selling a home on the market

FutureHouse gets to collect data at each step of this process. These are its main data assets:

- Pictures of all homes for sale
- Descriptions of the homes (square footage, location, and so forth)
- Web visit and click history for all users of the website

- Historical records of properties sold in the region
- Reviews of the homes left by people who visited them
- Transcripts of negotiations between agent and buyer
- Newspaper articles about upcoming construction in the region

All these data sources don't hold the same value; their dollar density is different. For instance, the newspaper articles are a very general dataset that every other company has access to. Therefore, while it might be valuable, it's not core to FutureHouse. Website visits, images, and reviews may be important because they're specific and unique to the business, and therefore their dollar density is higher than that of newspaper articles. Still, they don't have a direct impact on the success of the business. The historical records of houses sold is much more connected to the business's performance than all the other data sources, as the traditional business model of realtors is a success fee per house sold: this is your core business data.

Let's assume we want to leverage this data to build an AI-powered property evaluation tool that automatically predicts the best price for homes listed on the platform. Price estimation is usually performed by an agent who has experience with the area and similar homes. They'll have overseen the sale of many properties in the past, developing deep intuition about what the fair price of the property would be. This is the task that we want AI to take over.

Let's try to come up with a more exhaustive list of the factors an agent might consider before suggesting a sale price:

- Square footage
- Number of floors
- Number of bedrooms
- Number of bathrooms
- Year of construction
- Does it have a pool?
- Does it have a garage?
- Energy efficiency
- Location
- Quality of the neighborhood
- Public transportation

If you ask how the agent came up with the price, they'll probably say something like "by experience." They might refer to standard guidelines like the average price per square foot in that neighborhood, but it's hard for them to articulate how other factors affected their decision. If you can't explain your thought process, you can't even code it into a computer program.

An agent learns how to value homes by looking at the selling price of many properties over time. The main idea behind ML is that a computer can do the same: it can learn how to value a home from the historical data of previously sold ones, feeding it with both their characteristics (size, neighborhood, and so forth) and the final sale price.

This AI would allow you to have the same expertise as an experienced broker, at the same speed, scale, availability, and cost of software, allowing any home seller to get an estimate of their property's value in a split second. This marriage of data availability and business value is the holy grail of AI-based innovation. We have identified a clear value proposition, backed by the right type of data to build it.

2.2.2 Adding AI capabilities to FutureHouse

One of the goals of this book is for you to learn how to talk to data scientists and engineers, so this is the paragraph where you become familiar with the lingo and technical details. At this stage in the ideation process, we have decided that we want to build an automated house-price-prediction engine. We believe it will help optimize our internal processes and offer value to our customers, as they'll be able to get a quote for their home directly online. We also have a record of transactions in the region, including some important features of the house (for example, square footage) and the sales price. If you brought all of this to an engineer, here's how the hypothetical conversation would go (keywords are highlighted in italics):

> You: We'd like to add an automatic price predictor to our real estate listings website. Can you help us with that?
>
> Engineer: Nice—this is a standard problem in machine learning. There are some well-known *machine learning algorithms* that I'm pretty confident can produce a good *model*. What *inputs* do we have?
>
> You: We spoke with the agents, and they usually factor in the square footage, number of bathrooms, distance to public transport, and things like that. Here's a complete list.
>
> Engineer: Sure, the list of *features* seems like a good start. And what *target* do you need from the model? Just the expected property price?
>
> You: Correct, that's all we need. The price will help us offer real-time quotes on our website for free.
>
> Engineer: Cool. What data do you have available for this?
>
> You: We have records for all house transactions in the region for the past decade.
>
> Engineer: Great! Do those records include all the *features* you mentioned earlier?
>
> You: Yes, most of them. We surely have the square footage and stuff like that, but you might have to figure out something for the public transport.
>
> Engineer: OK, I can work with that. Do you also have *labels* for all of them?
>
> You: Yes, the records include the final sale price of each property.
>
> Engineer: Amazing. How many *examples* are we talking about?
>
> You: We have around 200,000 transactions for the past five years.
>
> Engineer: Fantastic! That should be enough to *train our model*.

We've introduced quite a few abstractions so far, so let's stop for a second and recap:

- *Machine learning algorithm*—A technology that allows computers to learn from data
- *Features*—A set of characteristics of an object that the algorithm can learn from
- *Label*—The output or target we want the algorithm to predict
- *Training*—A phase in which the machine learning algorithm is fed with past examples and learns from them
- *Model*—The output of the training phase—a self-contained computer program that can predict a label, given a set of features
- *Inference*—The phase in which the model is used with new examples

The great thing about all these abstractions is that we can now apply them to other problems. For example, say we want to predict the price of used cars rather than homes. Can you think of what features you would use? Car model and mileage probably are good guesses, as is the maintenance history. But you can have even more imagination: think about government subsidies for low-pollution vehicles, the number of Google searches for those models, and more. More generally, all problems for which we want to predict a value (which we don't know yet) based on a set of information we do have are examples of *supervised learning*.

Supervised learning is the area of machine learning that has the most applications in industry and research today. Our example so far belongs to this subset: it uses a set of techniques that allow machines to learn a mapping between a set of information called *features* and a target value called a *label*. The beauty of supervised learning, and of machine learning in general, is that the computer will automatically find this formula, no matter how complex it is. A successful model will make predictions that are consistent with the labels, thus transferring the experience embedded in the dataset to new cases. Figure 2.4 shows how features and labels interact in supervised learning applications.

By now, you have learned the basic concept behind supervised learning. As you can see, it's not rocket science. It's actually simple to grasp. The complexity is in the inner workings: the *machine learning algorithms*, the engines that allow computers to perform the learning. In this book, we won't dig deep into these algorithms unless we think it

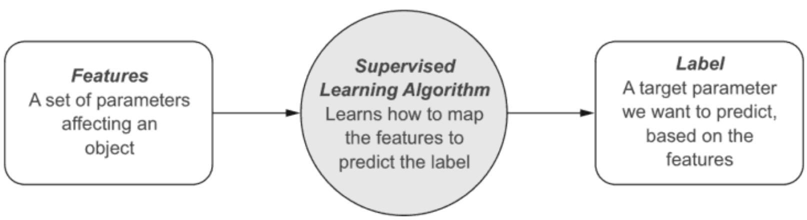


Figure 2.4 The core concept of supervised learning: finding a mapping between a set of features and a label

can be interesting and useful for you. With today's tools and services, we believe that you can generate value for your organization by knowing just enough to come up with AI products or services and passing the ball to ML engineers when it's time. Now you're starting to know how to do both, so high five!

Even if the previous concepts seem simple, don't underestimate your newfound knowledge: billions of dollars have been produced by companies that used the simple techniques you just learned. Why is machine learning so powerful, then? This is the topic of the next section.

2.2.3 The machine learning advantage

In the previous section, we had a pleasant conversation with an imaginary engineer, and we parted ways knowing that they would be building a model to predict house prices based on some of its features, such as the square footage. Scrap the price-prediction problem for a second, and imagine that we want a calculator for the property tax that the buyer would have to pay. If that were the case, our conversation with the engineer would have been much shorter:

You: We'd like to add a property tax calculator to our website. Can you help?

Engineer: Cool. What's the relevant section in the tax code?

You: You can refer to 4123 and 564b.

Engineer: Great, that's all I need.

The main difference between building a price predictor and a tax calculator is that the latter is governed by precise rules already expressed in mathematical form. Even the most junior of accountants can apply the mathematical formulas that determine property tax and apply them to new properties, with little need for experience. The same goes for physics: ever since Isaac Newton discovered the laws of dynamics in the 1600s, we have been able to predict the motions of bodies and build rockets to send people to the moon. Anytime we have situations like these, mathematics and conventional computer science will do just fine: we can just translate the rules and formulas into computer code, and the result will be 100% accurate.

This is not the case for the price-prediction example: the relationship between the features of the house and its sale price is fuzzy, unclear, and definitely not written down in a rule book. There's no way even the most experienced agent can write down computer-friendly rules for valuing properties, and no way a programmer can translate them into software.

In the past century, humanity used computers to realize amazing goals, even going as far as launching rockets into space and orchestrating billions of financial transactions every day. And yet, our most sophisticated robots still can't cross a road as safely as a six-year-old kid can.

In fact, computer science has been stuck on many important problems for a long time because they were just too complex to be understood analytically. Translating poetry into foreign languages, driving cars, or answering the phone are all tasks that