BECOME AN AI COMPANY IN 90 DAYS

The no-bullshit guide for understanding AI, identifying opportunities, and launching your first product

KEVIN DEWALT

Produced by: Russ Rands

BECOME AN AI COMPANY IN 90 DAYS

The no-bullshit guide for understanding AI, identifying opportunities, and launching your first product

KEVIN DEWALT

Produced by: Russ Rands

ISBN 978-0-692-19233-7

Prolego, Inc.
400 West Peachtree Street NW
Suite #4 - 868
Atlanta, GA 30308

prolego.io
admin@prolego.io

Contents

Even if there were a trustworthy way to send money over the Internet–which there isn't–the network is missing a most essential ingredient of capitalism: salespeople.

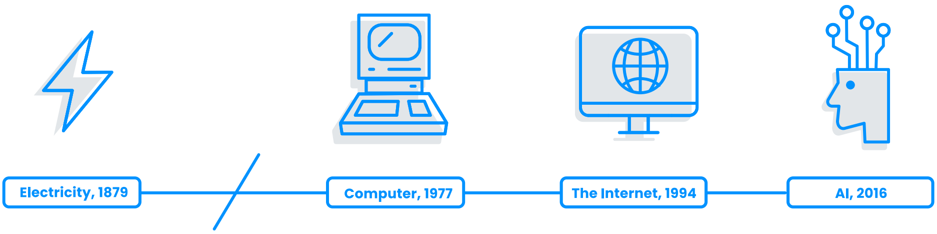
– **Clifford Stoll**

"The Internet? Bah!" *Newsweek* 1995

Introduction

THE ERA OF INTELLIGENT COMPUTING

Congratulations. You're living through a once-in-a-generation technology shift—the era of artificial intelligence, or *AI*. Like previous fundamental shifts such as electricity, the computer, or the Internet, AI will change everything.



Skeptical about such big claims? I don't blame you. Over the past few decades we've been bombarded by an endless parade of new technologies promising to "disrupt" everything and solve all our problems.

Most of these technologies never yield more than incremental benefits:

- Object-oriented programming made software only slightly more reusable.
- Social media moved us closer to only a fraction of our customers.
- NoSQL databases solved some scalability problems and created many others.

Big data, fuzzy logic, mobile, ontologies, . . . year after year the buzzwords keep coming. And most of them merely come and go.

But how about electricity? The microprocessor? Personal computers? The Internet? Enthusiasts welcomed these technologies with lots of hype, and everyone underestimated their long-term impact. Today no large-scale business can survive without all of them. These fundamental technology shifts created value by:

- **Replacing or augmenting labor** (industrial automation, appliances, robotics)
- Helping us **work more efficiently** (spreadsheets, databases, enterprise software)
- Helping us **communicate more efficiently** (phones, email, web sites)

AI, the current technology shift, creates value in an additional way: *by replacing or augmenting human thinking*.

No doubt you're already seeing examples of AI in consumer products. Facial recognition works almost flawlessly in the iPhone X. Siri and Alexa can interpret and answer simple questions, and voice-to-text on our smartphones improves daily.

But these narrow AI solutions are only the beginning. Soon business processes that involve thousands of interactions between people, and software will be replaced with AI models.

THE INTELLIGENT VACATION PLANNER

Let's consider how AI might be applied to something that's relatively easy to accomplish, like vacation planning.

Twenty years ago vacation planning was straightforward. A travel agent would ask a few questions and then present us with a handful of options. "Would you like to visit Paris, Rome, or Athens?" This simplicity came at a cost–it was relatively expensive and generic.

Today we have unlimited travel choices for nearly every budget. Unfortunately the volume of data and options creates a hassle. We have to make dozens of decisions:

- When to go
- Where to go
- Where to stay
- How much to spend
- How to get there
- What to do

Each decision impacts every other. We have nearly infinite good and bad options, and all of them can change by the minute as prices, weather, events, and schedules fluctuate. So we spend hours Googling, reading reviews, and looking at prices and weather patterns. We've exchanged simplicity for choice.

But an AI-powered vacation planner could give us both simplicity *and* choice. Such a system could ingest thousands of data sources:

- Your social media pictures and activity
- Your friends' social media activity
- Your email
- Your calendar
- The real-time pricing and schedule of every transportation resource
- The availability and cost of every lodging option
- Every local event
- The weather forecast

AI models could process these disparate data sources and suggest vacation options. We can imagine a conversational interface built into Alexa:

> Me: Alexa, can you suggest some vacation options in August?
>
> Alexa: Sure, Kevin. I'm getting some options from the Intelligent Vacation Planner. What do you think of golfing in Costa Rica, wine tasting in southern France, or relaxing in Chengdu?
>
> Me: China is too far. What are the best times for Costa Rica and France?
>
> Alexa: Anytime in August is good for Costa Rica. France is too expensive until the 25th.
>
> Me: What else can I do in Costa Rica?
>
> Alexa: Costa Rica is known for sauvignon blanc, pinot noir, and Syrah wines. You could visit some wineries.
>
> Me: Awesome. Send me three itineraries for a five-day vacation in Costa Rica. I'd like to play at least three rounds of golf and want to stay in the same hotel I had last time.

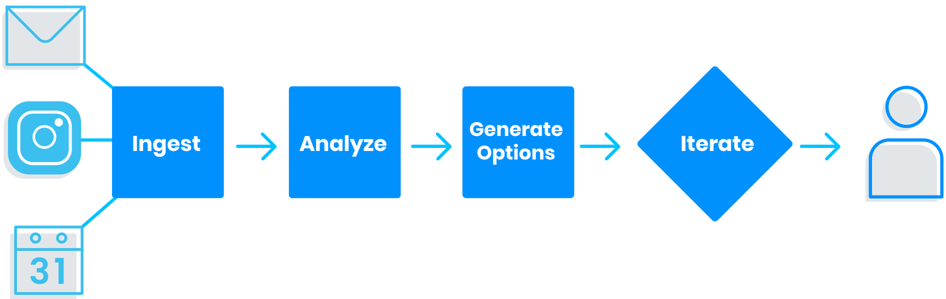
I'm thrilled to let the Intelligent Vacation Planner augment and replace my own thinking. Why should I have to comb through TripAdvisor reviews to pick a good golf course? Or use Kayak to find good flights? Airbnb to find appropriate lodging? Since there is no "best" vacation, I'd rather offload this intellectual hassle to AI so I can focus on doing client work—or playing golf.

Building the Intelligent Vacation Planner doesn't require a scientific breakthrough or the visionary leadership of Elon Musk. The fundamental AI building blocks already exist. It's only a matter of time before a company creates it. And by the end of this book you'll know how to do it.

AI IN YOUR COMPANY

To understand how you might use AI in your own business contexts, let's break down the business workflow of the Intelligent Vacation Planner:

1. Ingests data from many different data sources.
2. Analyzes complex data such as images and documents.
3. Generates the best solutions from sometimes infinite choices.
4. Provides the user a means to iterate options and make a final decision.



Many businesses have entire departments dedicated to performing similar workflow functions. Their processes can be exceedingly complex and time consuming. Consider the processes involved in some of our major industries:

- How do **financial services** generate investment recommendations for their clients?
- How do **accountants** process invoices?
- How do **hospitals** diagnose problems?
- How do **banks** review correspondence for compliance?
- How do **insurance** companies process claims?
- How do **construction** companies develop cost and schedule estimates?
- How do **governments** process tax returns?

Here's the answer: *inefficiently*.
I know because I've talked to Fortune 500 executives about applying AI to every one of these business processes. I have yet to encounter a single large company whose business won't be dramatically impacted by AI. Your company is no different.

AI EVOLUTION IN THE PATTERN OF THE INTERNET

I try to help my clients understand AI by drawing analogies with the Internet, the most recent fundamental technology shift. I agree with Chris Benson–AI in 2018 is like the Internet in 1996. Let's take a trip down memory lane and talk about the Internet's beginnings.

The current state of enterprise AI is like the Internet in 1996.

– Chris Benson,
Chief Scientist of AI at Honeywell

I was a graduate student at Stanford in the mid-1990s, the dawn of the Internet. I had been using early applications like email, FTP, Listserv, AOL, and Gopher, but none of these were indispensible yet. Then in October 1994 a classmate showed me Netscape 0.9, the first widely adopted web browser.

Like thousands of other early Internet enthusiasts, I instantly realized the world was about to change: everyone would connect through this simple computer interface. Anyone would be able to instantly find answers to questions. Commerce would move to online catalogs. Business processes between companies would become transparent and automated. Computer games would be global. Best of all, utopian democracies would emerge when everyone had access to the same information on the Internet. (Yeah, got that one wrong, didn't I? Ah, youth).

I knew a revolution was coming and I wanted to be part of it. I started telling everyone about the Internet–and quickly realized most people had no idea what I was talking about.

Most people asked me questions like:

- \> So . . . what is the Internet?
- \> Where is it?
- \> CompuServe already does all this for $10/hour–who do you pay for the Internet?
- \> What is the difference between the World Wide Web and the Internet?
- \> What is the market for it? (First meeting with a Silicon Valley investor.)

Figure 1

Most early Internet enthusiasts had the same experience. A Wired reporter registered McDonalds.com and couldn't get McDonald's corporation to take it for free.[1] Of course everything changed on August 9, 1995, when investors valued Netscape at a $3B on its IPO.

Fundamental technology shifts

Why did it take several years for most people to recognize the Internet's potential? Because fundamental technology shifts are initially abstract. It is hard to explain "everything will change" to someone who isn't trying to understand what is happening.

---

[1] www.wired.com/1994/10/mcdonalds

Figure 1: https://www.npr.org/sectionsalltechconsidered/2016/07/25/487097344/the-big-internet-brands-of-the-90s-where-are-they-now

Most new technology isn't so abstract, and it meets known needs. NoSQL, for example, is a relatively new technology that promises to overcome known problems with relational databases—that's why it isn't a fundamental shift. NoSQL also has a natural home in your infrastructure or product team.

Fundamental shifts don't necessarily solve known problems. They don't fit neatly into existing departments. And they don't replace technologies so much as they propel additional innovations. Fundamental shifts include electricity, computers, the Internet, and now AI.

That's why you're having a hard time identifying specific AI use cases for your company. The biggest opportunities for AI are not obvious. Most people give up when they don't see easy answers to hard questions. "I just don't see the value," they say.

But not you. You're reading this book because you're willing to invest the time and energy to lead your company into the AI future. Ironically, planning for a big, broad impact like AI is actually *easier* than identifying specific use cases because you can start with a blank slate.

AI's predictable path forward

Fundamental technology shifts don't happen instantly. Researchers worked on electricity, heavier-than-air flight, the microprocessor, and the Internet for decades before the technologies became practical.

AI is no different. Neural networks—the algorithms which have enabled the recent AI breakthroughs—have been in the research-and-development shop for 70 years! They only recently became useful because the right enabling technology and market forces converged.

In the following table, consider the parallels between the adoption curves of the Internet and AI.

	Internet, 1996	AI, 2018
History	Decades of research investment creates a few closed networks.	Decades of research investment leads to a few niche applications.
Awakening moment	Netscape 0.9 is released (October 1994).	Google Translate converts to AI (November 2016).[2]
Enabling tech	HTML standardization, faster modems, data compression, routers, faster computers.	Data proliferation, inexpensive storage, cloud computing, GPUs.
Impacts	Every business and person are instantly connected. Online shopping, integration between businesses, declining information power, reduced transaction friction.	Human thinking is automated and replaced by computers. Predictive models replace complex workflows. Data becomes the most valuable asset.
Human capital challenges	Leadership has no strategy. Qualified talent for development is hard to find.	Leadership has no strategy. Qualified talent for development is hard to find.
Risks	Competitors build brand awareness first. New entrants with different business models and lower cost structures steal market share. Most early bets fail.	Competitors build capabilities first. New entrants with different business models and lower cost structures steal market share. Most early bets fail.
Opportunities	New products, higher margins through direct-to-consumer engagement.	New products, higher margins through faster and more accurate decisions.
Next Steps	Set up web presence. Retool employee skills in web stack. Assign stakeholders to launch experimental initiatives based on investment trends (e-commerce, B2B transactions, online support).	Organize data assets. Retool employee skills in model development and deployment. Assign stakeholders to launch experimental initiatives based on investment trends (computer vision, NLP, automated decisions).

[2] https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html

Of course trying to predict which specific technologies or companies will emerge as winners in this technology shift is impossible. Google wasn't predictable, but the migration of advertising dollars to the web was clear from the outset. Any company with significant revenue from advertising could have begun this transition in 1996, even though the ultimate impact and timing of online advertising was unknown.

Today we're in a similar situation with AI. Venture capitalists are pouring billions into AI initiatives. Most of these will fail.[3] Most initiatives by your competitors will fail, and many of your own initial assumptions about AI will be completely wrong.

Yet there is little doubt that AI will radically change your business. You have entire departments which will transform to a workflow like that of the Intelligent Vacation Planner. You just don't know when or how.

Regardless, you can begin making small bets and taking incremental steps to prepare for AI. You're about to learn about those bets and steps.

WHAT TO EXPECT FROM THIS BOOK

So here we are. You're trying to decide whether this book is worth your time. You're skeptical because, well, most of what you read about AI is useless. Vendors are pitching you magical solutions you don't understand. The blogs, podcasts, online courses, and books are too technical. The last AI conference you attended presented nothing but fluffy concepts which didn't get you closer to your only real question: *What can we actually do with AI?*

Let's decide whether this book is worth investing your most valuable asset: your time.

---

[3] www.inc.com/john-mcdermott/report-3-out-of-4-venture-backed-start-ups-fail.html

Why should you listen to me?

I caught the AI bug as a graduate student at Stanford in the mid-1990s doing neural network research under Dr. Bernard Widrow, a legend in computer science. I was fascinated by "computers that think" even though the technology at that time fell short of even modest expectations. I decided to pursue a career in industry and had the fortune to help drive innovation in many related projects on the road to AI. I've run massive data processing applications for anomaly detection (FINRA), helped launch the first version of Palantir into the US intelligence community (In-Q-Tel), and served as CMO for a venture-backed data science startup (MadKudu).

I've also got the battle scars of many failed startups, failed investments, and failed enterprise software initiatives—experiences which taught me far more than my successes.

So what, right? My career looks like that of everyone else who's promising to help you achieve great things with AI. Smart guy, good schools, great experience . . . blah, blah, blah.

Why should you listen to me over anyone else? Because I'm investing all of my energy into bringing AI to the enterprise—**that is, I focus on solving only *your* problems.** I'm not concerned with the latest AI fads, research papers, or hot new startups. I'm not trying to impress you with how much I know by using needlessly complex jargon. I care about helping you transform your enterprise with AI.

A few years ago I realized that AI had finally matured to the point where I could start using it to solve real problems. With high hopes and big dreams, I launched an AI startup with a colleague from Apple. Here's a summary of most customer sales conversations: "Thanks Kevin, but your product sucks. But while you're here, can you explain AI to us?"

As I engaged with these companies, I started to realize that the most interesting AI problems were those that involved bringing the technology to large enterprises which had the data and business-process complexity to take advantage of AI.

Coincidentally, my friend Russ Rands was running a data science startup and had come to a similar conclusion. Russ and I decided to join forces and founded Prolego (which means *predict* in Greek) with the mission of bringing AI to Fortune 500 enterprises. We've spent the last few years helping dozens of executives in industries like banking, insurance, defense, financial services, healthcare, transportation, hospitality, automotive, and real estate.

We started recognizing similar problems across these industries:

> Confusion about basic concepts like machine learning.
> Unfamiliarity with how to value data assets and estimate data gaps.
> False assumptions about AI talent shortage and necessary skills.
> No frameworks for generalizing fundamental technologies (e.g., recurrent neural networks) into solutions.
> No frameworks for exposing the major risks and opportunities with AI projects.
> No heuristics for deciding what to buy or build.
> No means of choosing among many potential AI opportunities.
> Uncertainty about the first steps.

Simply put, these executives *needed a practical AI strategy*. We developed a process for creating one—a process I'm about to share with you.

Let’s talk about you

You’re not Google: you don’t have $1B to buy AI startups and invest in everything that sparks your interest. You’re not a startup: you can’t stand up a server tomorrow and start building whatever you want. You’re not a research organization: you don’t have the resources to push the state of the art in AI.

You work in a big, complex organization that has evolved over decades. Your technology team is still trying to finish the CRM and ERP initiatives started five years ago. Your data? Nobody can tell you where it all resides, what is in it, and who owns it. Getting anything done at your company is harder than outsiders expect. And. It. Takes. Forever.

Sound familiar? Relax. I promise you’re not alone. Your competitors are in the same position, their fancy demos and press releases notwithstanding. The CEO of that hot AI startup you met is terrified you will realize he doesn’t have anything magical you can’t build yourself.

AI is still very new, and you have time to develop your strategy and systematically test ideas. You don’t have to recruit PhDs from Stanford and MIT, pay outrageous prices to acquire AI startups, waste hours listening to vendors pitch you magical AI solutions, or invest three months learning Python so you can take Andrew Ng’s new online course. You don’t even need to hire me to build your AI strategy, because I’m about to show you how to do it yourself.

This book shows you how to **build an AI strategy** and **launch your first product**. You’ll find out how to generate AI ideas and systematically vet them for business opportunities. You’ll learn how to get your organization comfortable with experimentation and to balance competing near-term and long-term opportunities. You’ll see how to set up your AI infrastructure, build a team, and organize your data. You’ll know when and how to deploy your AI solution and how to avoid common mistakes.

AI is once-in-generation opportunity—one which will create fortunes and make careers. Why can’t it be yours?

Bullshit is unavoidable whenever circumstances require someone to talk without knowing what he is talking about.

– Harry Frankfurt,
On Bullshit, 2005

Part 1

AI FUNDAMENTALS

Every day I hear people carelessly use terms like “machine learning” and “AI.” For example, the statements “In Phase 2 we will apply machine learning to our data lake” or “Our product uses AI” are meaningless.

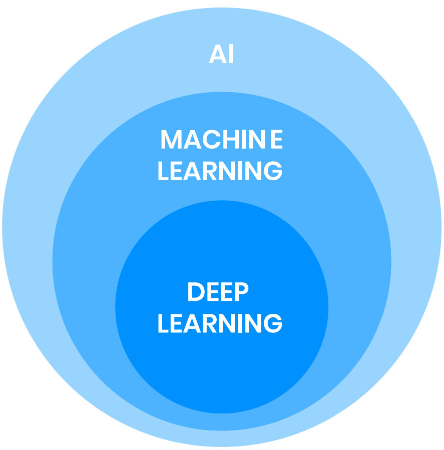
You probably find AI terminology confusing as well, and it isn’t your fault. Techno elitists and marketers deliberately use AI jargon so you’ll think they have some esoteric knowledge. Most of the time they’re just bullshitting. The true AI experts (like Jeff Dean of Google or Andrew Ng of Stanford) are able to simplify concepts considerably for their audience. You need to understand only a handful of basic AI concepts to build your strategy and communicate with your technical team.

DEFINING AI

AI just means "intelligent software." The term is about as specific as "the Internet." I use the term *AI* when speaking to broad audiences about this fundamental technology shift. Is your calculator AI? Sure, in the sense that it's programmed to "think" for you.

AI is a useful, general term for the trend of software that performs complex cognitive tasks previously done by people. That's really all there is to it.

MACHINE LEARNING AND DEEP LEARNING



Two other common terms you will encounter are *machine learning* and *deep learning*. Machine learning is a type of AI. Deep learning is a type of machine learning.

You need a basic understanding of both machine learning and deep learning to build your AI strategy. Fortunately both are simple concepts.

MACHINE LEARNING IS A DIFFERENT WAY OF BUILDING SOFTWARE

Machine learning is a technique for teaching computers how to perform specific tasks through data. Before expanding on this definition, let's first talk about how we developed software before machine learning came on the scene.

Traditional software development

Today almost all software is built without the aid of machine learning. Most developers write software that explicitly tells a computer to do something with data. For example, suppose a developer wants to detect whether a number exceeds a maximum value. The developer could write a program like this:

```
MAXIMUM = 10

if x > MAXIMUM
 print "$x is too high"
else
 print "$x is ok"
end
```

Even if you've never done any programming, you can guess what this simple program will do. It first establishes the maximum value at 10 and then checks each number to see if it is greater than 10. Here is what the output looks like:

```
 4  is ok
13  is too high
10  is ok
 9  is ok
27  is too high
-12 is ok
...
```

But how did the developer know the maximum value should be 10? Why isn't the maximum value 11? Or 9876? Or –1?

ANSWER: Because someone (analysts, customers, product managers, a specification, etc.) *told* the developer to set the maximum value at 10.

The answer didn't magically fall out of the sky. A *human being* looked at the world and made the decision to set the maximum value at 10. Almost all software at your company is made this way. Someone tells a developer what they want the computer to do, and the developer then tells the computer how to do it.

Software development with machine learning

Machine learning is a different way to create software. **In machine learning, a developer uses data to teach a model (or algorithm) how to perform a specific task.**

Developers start with examples of the task they want the computer to perform. They then systematically "teach" the algorithm how to perform the task using examples called *training data*. Let's rewrite the same program using machine learning.

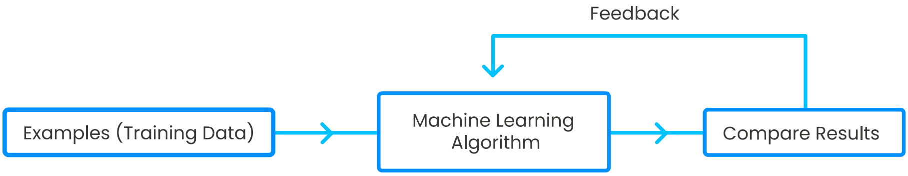
Step 1: Developer receives training data

First someone gives the developer examples which illustrate how the software should work:

Example model input	Desired model output
7	OK
11	too high
34	too high
10	OK
9	OK
17	too high
-78	OK
0	OK
1	OK
18	too high

Step 2: Developer trains an algorithm to complete the task

Next the developer uses training data to teach the algorithm to decide whether a number is too high. The developer shows each example to the machine learning algorithm. The algorithm tries to predict the results and compares its prediction with the desired result from the training examples.



For example:

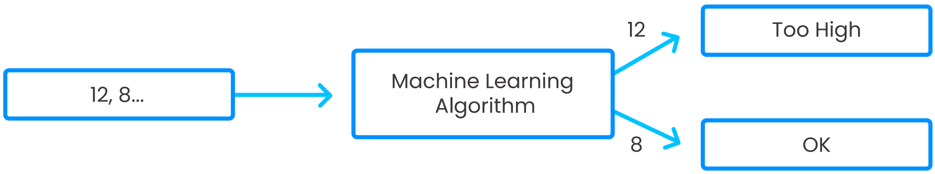
Example model input	Desired model output	Algorithm prediction	Feedback
7	OK	too high	wrong
11	too high	too high	correct

At first the algorithm doesn't work very well, but after the developer shows it enough examples it gradually gets better.

Note the key difference: the developer *doesn't know what the maximum value is*. The developer only provides examples.

Step 3: Developer deploys the software

Eventually the algorithm sees enough examples that it starts producing good results. The developer can now deploy the software.



Why use machine learning?

Machine learning looks more complex than traditional programming, doesn't it? Using traditional programming, the developer just types a few lines of code and tells the computer what to do. Machine learning requires many additional steps, such as gathering training data and using feedback to adjust the algorithm.

Machine learning isn't a good technique for solving simple problems like the one we just considered. But imagine we have a more complex problem. Suppose we have many inputs and want to decide whether a number is too high based on a lot of other factors. Normally any number greater than 10 is too high, but on Wednesdays any number greater than 11 is too high. And if the user is a child any number greater than 10 is OK, except after 5 p.m. . . .

You get the idea. In the real world, simple problems often get more complex as we try to generate more useful results. Software programs get bigger, more difficult to modify, and more expensive to maintain. In these instances machine learning can be a better solution. Consider the breakdown of software development characteristics in the following table.

	Traditional software development	Machine learning software development
Development approach	Developer explicitly tells computer what to do.	Developer trains an algorithm to perform a specific task by using examples.
Data requirements	Little or no data is required; humans generalize logic from the data.	Success is highly reliant on quantity and quality of examples.
Adaptability	Requires redesign when the environment changes.	Adapts to environment whenever it is shown new examples.

DEEP LEARNING, THE AI BREAKTHROUGH

Machine learning isn't new; banks, telecom companies, and government intelligence services have been using it for decades. But historically machine learning has been useful for solving only a few specific problems, such as detecting potential credit card fraud.

Machine learning hasn't worked well as a general solution, and it hasn't been able to deal with unstructured data such as documents, images, or video. But in the past five years, researchers have developed a new type of machine learning which can handle almost any type of data and generalize to many different problems. This advance in machine learning is called *deep learning*.

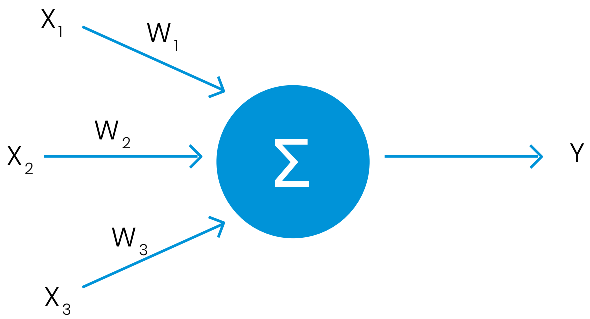
For our purposes deep learning is a type of machine learning characterized by:

> A specific machine learning algorithm called a *neural network*.
> Specialized computers called graphical processing units (GPUs).
> Very large datasets.

Let's look at each of these elements more closely.

Neural networks

You don't need to understand neural networks to build your AI strategy, but I'm including a definition here so you understand this basic building block. **A neural network is a machine learning algorithm which consists of many connected "neurons."** The concept is loosely based on early theories of how the brain works. Each neuron is an independent mathematical function which we can simply represent like this:



$$Y = W_1 * X_1 + W_2 * X_2 + W_3 * X_3$$

The neuron receives the inputs X_1, X_2, and X_3. The output is Y. The neuron generates an output by multiplying the inputs by weights, represented here by W_1, W_2, and W_3. The machine learning engineer wants to identify the weights which best predict the output based on the inputs.

Example: Predicting home prices

Let's pretend we're trying to build a neural network which predicts home prices (our output) based on three inputs:

1. The square footage of the home.
2. The number of bathrooms.
3. The average price of nearby homes.

We can think of our inputs like this:

X_1	square feet
X_2	number of bathrooms
X_3	average nearby price

Here is a sample of our training data, substituting the three inputs for X_1, X_2, and X_3:

W_1	Ft²	W_2	Number of bathrooms	W_3	Average nearby price	Price
?	2,350	?	2.5	?	715,000	560,000
?	1,750	?	1	?	275,000	195,000
?	3,778	?	4.5	?	199,000	375,000

A developer would use examples of real house sales to train this neuron to discover the values W_1, W_2, and W_3. For example:

```
W1 = 3
W2 = 4
W3 = 5
```

After training we are ready to make new predictions based on these weights:

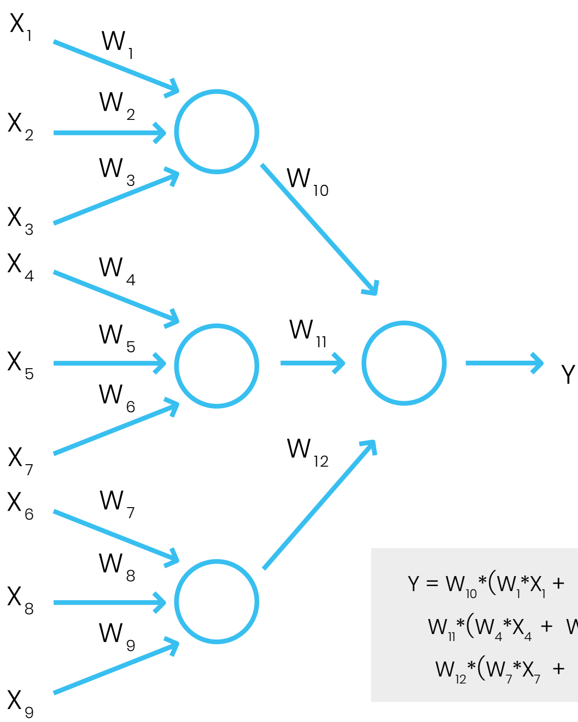
```
price = 3 *(ft²) + 4 * (no. bathrooms) + 5 * (avg.
nearby price)
```

Unfortunately this model probably isn't very good. I'm sure you can imagine many other influences on the price of a home: quality of schools, last remodel date, condition, etc.

To take advantage of these other inputs, we can design an even bigger neural network—one with many neurons and many more weights.

X_1	square feet
X_2	number of bathrooms
X_3	average nearby price
X_4	school quality score
X_5	crime frequency
X_6	years since remodel
X_7	number of garage spots
X_8	distance to grocery store
X_9	number of bedrooms

Our new larger (or deeper) neural network looks like this[4] :



$$Y = W_{10}*(W_1*X_1 + W2*X2 + W_3*X_3) + W_{11}*(W_4*X_4 + W_5*X_5 + W_6*X_6) + W_{12}*(W_7*X_7 + W_8*X_8 + W_9*X_9)$$

In our simple model we had 3 inputs and 3 weights. We now have *9 inputs* and *12 weights*. Weights W_{10}, W_{11}, and W_{12} can be used to make better

---

[4] Technical readers may observe that I've greatly oversimplified the basic multilayer perceptron neural network architecture. In practice all inputs will connect to the first layer of neurons, and neurons will have bias and activation functions. But while more technically correct, these details would distract the business reader from the key points.

pricing decisions because our model learns more complex relationships between inputs. The equation gets pretty messy, but you get the general idea: the more inputs you add, the more complex your neural networks become, and the better predictions you can make.

Practical solutions use very big neural networks. How big? Typically at least one million neurons in multiple layers. This type of very big, layered neural network architecture is what we use for "deep" learning. (You could also think of it as "big learning.")

GPUs

Unfortunately big neural networks are not easy to train—normal computers are just too slow. Researchers train neural networks fast using specialized hardware called graphical processing units, or GPUs.

Figure 2

GPUs were originally developed to efficiently render video in applications like gaming. You can buy one or more GPUs and build a deep learning server for a few hundred dollars. You can also rent them from cloud computing providers like Amazon Web Services.

Very large datasets

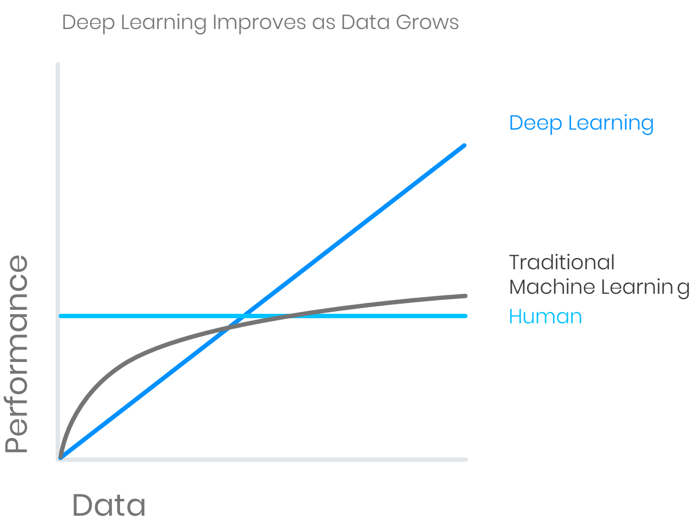
Very large neural networks require more training data than traditional machine learning algorithms. Researchers have achieved recent deep learning breakthroughs in areas like computer vision and natural-language processing by using millions of training data examples.

Figure 2: https://pxhere.com/en/photo/1387034

However, *you don't necessarily need massive amounts of training data* to use deep learning at your company. Machine learning engineers have clever techniques for getting around this obstacle, as we'll discuss later.

IMPROVING DEEP LEARNING OUTCOMES

To fully appreciate the power of deep learning, consider the following graph designed by Andrew Ng[5]:



Humans can use logic and intuition to generate good results with little data; however, our brains don't adapt well to increasing data volumes. By contrast, traditional machine learning techniques improve with larger data volumes, but outcomes eventually plateau. In deep learning neural networks require more data, but their performance continues improving as data grows.

Ng's graph compares the performance of human brains, traditional machine learning algorithms, and deep learning with respect to data. Let's consider each.

5 https://www.youtube.com/watch?v=21EiKfQYZXc

Human performance

The human brain evolved to make fast, complex decisions based on very little data. We rely on logic and intuition to make sense of the world. Show a small child a picture of a zebra and she will probably be able to identify a zebra in a completely different picture. Show a child 10 pictures of zebras and she'll be even better at identifying zebras. But show her 30 pictures of zebras and she probably won't improve much. Attempt to show her 500 pictures and you'll see performance degradation (and a temper tantrum). People can process only so many details before they feel overwhelmed.

No computer can come close to matching human performance on making complex decisions based on little data. Unfortunately human performance doesn't scale well with increasing data.

Traditional machine learning performance

Traditional machine learning techniques don't work well at low data volumes. But as data volumes increase, performance improves and eventually exceeds human performance.

Unfortunately traditional machine learning techniques don't generalize well beyond a certain level of complexity because the algorithms are designed to solve specific data problems. Adding more data results in decreasing returns.

Sometimes traditional machine learning models are the best choice for a problem. The models are simpler than deep learning models, and an engineering team can improve them faster than they can improve a deep-learning system.

Deep learning performance

Neural networks require more data and are harder to train than traditional machine learning systems. But by (1) building bigger neural networks, (2) training them on faster GPUs, and (3) adding more data, deep learning performance continues to improve. If you want to achieve state-of-the art results in many computer science problems, you will need to use deep learning.

DEEP LEARNING CHALLENGES

Deep learning is more complex than traditional machine learning:

> Deep learning tools are immature and rapidly evolving.
> Training neural networks takes specialized skills.
> Results are harder to interpret.

Nor has deep learning replaced traditional machine learning approaches. I still use traditional approaches for rapid prototyping or demonstrating fast results for clients.

Nevertheless, deep learning is becoming the dominant technique for creating AI. As infrastructure and tools have improved, the barriers to deep learning are declining. That makes deep learning more appealing to developers, who are applying it more widely. That wide application, in turn, continues to diminish the barriers, making deep learning increasingly preferable to traditional machine learning.

Deep learning often requires less initial data processing (called feature engineering) than traditional machine learning. To capitalize on this advantage, developers are beginning to replace traditional machine learning and data processing systems with deep learning. Many are reporting lower maintenance and deployment costs as a result.

KEY MACHINE LEARNING CONCEPTS

Now that you can define AI, machine learning, and deep learning, let's discuss the key business concepts for building an AI strategy.

Training data

Recall the first step in machine-learning software development: gathering training data. Your AI systems will succeed or fail based on the quality and quantity of your training data.

It's only fitting that acquiring and preparing training data can be the most expensive, highest-risk part of any AI initiative. *Data is the only long-term competitive advantage in AI systems*. Researchers are continually publishing effective AI models. Hardware is a commodity. But build the highest-quality proprietary dataset and you'll crush the competition every time.

Training data has two components: *outputs* (also called targets, labels, or dependent variables) and *inputs* (also called features or independent variables). The following table explains these two components of training data:

	Also called	Function	Strategic role	Challenge
Inputs	Features, independent variables	Used to generate outputs	Require investment	Generating enough quality inputs that are predictive of outputs
Outputs	Targets, dependent variables, labels	Results you want the AI model to produce	Create value	Identifying outputs that create business value

Outputs

Outputs are the results you want your AI system to produce. Identifying the desired output is one of the first steps in developing an AI strategy.

In practice machine learning outputs are numbers which stand for something else. In our simple machine learning example, the desired output was "OK" or "too high." We can represent "OK" with 0 and "too high" with 1 in our machine learning algorithms.

Other examples of outputs are:

- Classification or summary of a document
- Predicted future cost of a key resource
- Predicted sales demand for a product
- Identification of the presence of something (a missile, a person, etc.) in an image
- Assignment of a label to an image
- Probability of an event
- Recommendation of a product or service to an online customer
- A risk score for a cyber-intrusion event
- Identification of the spoken language in a conversation

The possibilities are endless. In Part 2 I'll share tools for identifying the most valuable outputs for your AI strategy.

Outputs usually support two types of business processes: *classification* and *regression*.

Classification processes predict whether something falls into a specific group. Examples are email spam detectors or image labelers. In our simple machine learning example, the output was a classification problem: "too high" or "OK." There are only two options in this simple classification process. Regression processes predict a quantity or value. Examples are sales forecasts or time.

Inputs

Inputs are the data the AI system uses to generate the outputs. Your company likely has enormous quantities of data. Your challenge is identifying the data's high-quality inputs for the AI models.

One of the biggest long-term AI costs is continuously building enough quality inputs. Since the world constantly changes, the value of your input data fluctuates. For example, home prices on a street may fall if a good school suddenly shuts down. You will constantly be searching for better inputs because the world always changes.

Ultimately, you will need data scientists and machine learning engineers to analyze and determine the quality of your inputs. But for the purposes of creating your AI strategy, you just need to review your data and estimate whether a particular input is potentially predictive of an output.

For example, suppose you want to predict the selling price of a home. The quality of local schools and number of bedrooms are both probably predictive of selling price. These would be high-quality inputs. The win/loss record of the city's baseball team is probably not a good predictor of house prices and thus would not be a high-quality input. Consider how inputs like these can contribute to useful outputs in your organization.

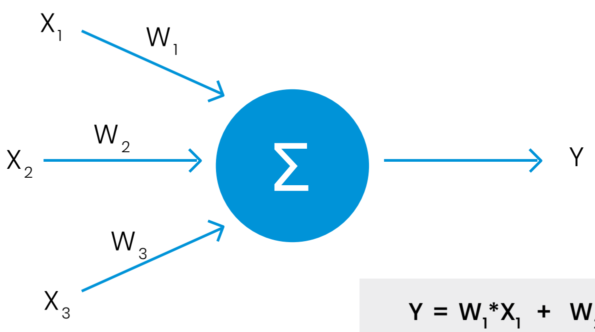
Here are some additional input examples:

- The pixels in a digital image
- The transcribed, digitized text of a conversation
- Time-stamped events from sensors
- Historical sales records
- Behavior of online customers visiting a web site
- Demographic information about a person or place

Models

The word *model* can be defined in many different ways and can be used in many different business contexts. We're not surprised when clients ask us, "What do you mean by *model*?" *Model* can refer to one part of a software application (as in model/view/controller) or to a set of business rules.

For our purposes a model (also called an algorithm) is the software which generates outputs from inputs. A neural network model starts with a set of random weights which are optimized through training. As an example, refer back to our simple neural network model:



$$Y = W_1{*}X_1 + W_2{*}X_2 + W_3{*}X_3$$

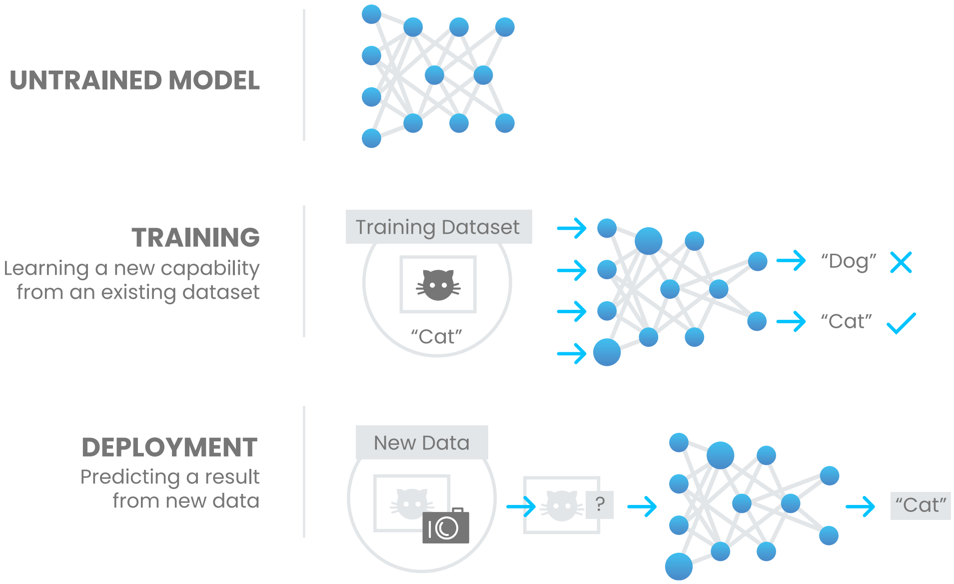
Inputs	X_1 , X_2 , X_3
Outputs	Y
Weights	W_1 , W_2 , W_3

A machine learning engineer starts with a model which has random values for the weights. She then tries to find the best values for the weights during training. Finally she saves the weights to a database so they can be loaded back into an empty model later.

There are an infinite number of machine learning models, and researchers discover new ones every day. Your AI team will need to continuously evaluate new models to see if they can give you a business advantage.

Training and deployment phases

Machine learning has two distinct phases: training and deployment (also called *inference*). During the training phase a machine learning engineer starts with an untrained model and teaches it how to generate the best outputs from a set of inputs. In the deployment phase the trained model is given new inputs and generates an output.



Many machine learning projects have great results in the training phase but fail in deployment, usually because the deployment input data is different from the training data. In Part 4 you will explore some simple techniques for preventing many of these problems.

The training phase takes longer and costs more than deployment. Machine learning engineers can spend months or longer building training data, developing models, and evaluating results.

Deploying models has other technical challenges, which we'll discuss in Part 4. Deployment challenges can also arise if models require downstream business process changes. For example, insurance companies have traditionally used simple algorithms and human labor to process claims. If an insurance company begins using more complex AI models to automatically process claims, the existing claims department will have to adapt. Will the claims department look at only complex claims? Will they look at only claims above a certain value? Or will they automatically review all claims?

PART 1 QUIZ

In Part 1 we covered the fundamental AI concepts you need to build your company's AI strategy. Before continuing to Part 2, take a few minutes to read the following questions and think through the answers. Forcing yourself to think through the answers will help your brain recall the concepts later.

1. How does machine learning differ from traditional software development?

2. What are the advantages of deep learning? What are the drawbacks?

3. What are the two components of data that are used for training? How are they different? What are other names for these terms?

Find quiz answers in Appendix 2.

“Opportunity is missed by most people because it comes dressed in overalls looking like hard work.”

– **Thomas Edison**

Part 2
DISCOVERING AI OPPORTUNITIES

Imagine you're a banker in 1896 trying to get investors excited about buying shares of 12 companies that are included in the new Dow Jones Industrial Average. Most of the companies produce valuable commodities such as cotton, sugar, tobacco, gas, iron, coal, and rubber.

Many investors are skeptical about one company in the index: Thomas Edison's company, General Electric. These investors have heard the hype about electricity and seen a few examples of electric motors and lights. But they can't understand why a company focused on electricity deserves to be listed next to cotton and sugar, commodities which dominate worldwide life and commerce. They ask, "How is the average business or family going to use electricity?"

Of course *you* can imagine how—as the world approaches a new millennium, you've been following the work of Nikola Tesla and Thomas Edison for years. You can see an inevitable new world economy emerging as companies learn how to transmit and leverage electrical power. But you can't point to power lines, radios, appliances, or factories, so how can you explain the potential of electricity to skeptical investors? How can you get them to understand the business opportunity in a technology which can change everything?

I often feel like this fictitious banker. Every time I speak to a company about AI, I am asked the same question: "What can our company do with AI?" Most people recognize that a transformational shift is coming, but they don't see a clear connection between AI capabilities and today's problems. They're looking for simple answers to complex questions.

Unfortunately, I don't have simple answers for them because:

- AI is still new.
- The technology is changing too fast.
- The successful AI projects are highly specific to each business and don't generalize well.

You can get some (mostly technical) insights from company blogs, podcasts, and conference presentations, but there are no easy answers.

Sometimes people give up when I can't provide easy answers about AI. But not you. You're reading this book because you want to be a leader in the AI revolution. You realize that fortunes and careers will be made in the next five years by those who put in the effort to connect AI technology with business needs. In Part 2 you will learn how to make this connection. I'm going to share specific tools we use every day to bridge the gap between AI capabilities and your hardest business problems.

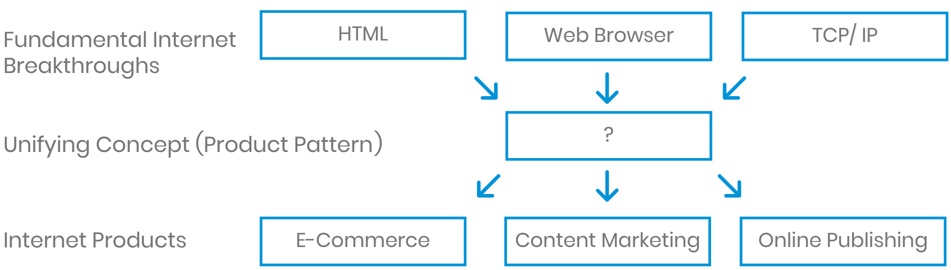
AI PRODUCT PATTERNS

Each AI breakthrough creates thousands of opportunities for new products and services. Unfortunately the connection between fundamental breakthroughs and specific solutions isn't always obvious.

For example, consider a few fundamental breakthroughs of the Internet in 1996:

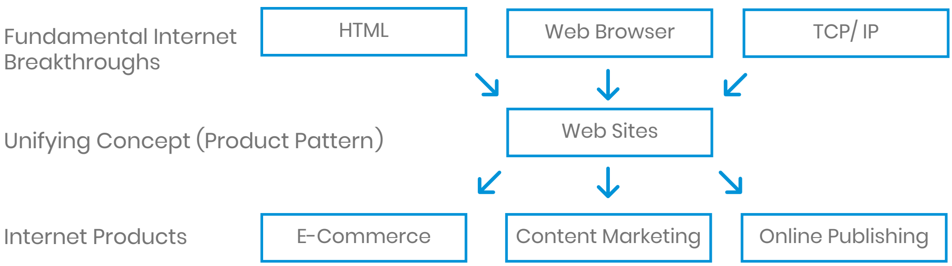
> **Internet protocol suite (TCP/IP):** The communications control standard for the Internet.
> **Hypertext Markup Language (HTML):** The standard programming language for describing the structure of a web page.
> **Web browser:** A software application such as NCSA Mosaic, Netscape, or Internet Explorer, designed for accessing information on the Internet.

Put all of these breakthroughs together, and what do you get? In 1996 that question was difficult to answer. Today we know that these were the building blocks for Internet services like e-commerce, content marketing, and online publishing.



But how did we get from HTML to an application like e-commerce? TCP/IP to online publishing? What was the unifying concept of these technologies, and why did it emerge?

The technologies emerged because whether they knew it or not, businesses that would survive past 1996 needed a way to interactively and efficiently communicate with customers in real time. This need was met by the unifying concept that was referred to as a *web site*.



Web sites, email, and instant messaging are examples of original Internet *product patterns*. Product patterns are practical applications of technology which solve recurring problems.

A product pattern enables businesses to identify workable solutions that are based on breakthrough technologies. By considering a broad product pattern, businesses of the early Internet age could have avoided the complexity of, “What can I do with HTML?” and instead asked “What can I do with a web site?”

We need similar tools to connect fundamental AI breakthroughs with workable solutions. I call these tools AI product patterns. **AI product patterns are practical applications of AI technology that solve recurring business problems.**

At present, businesses should consider four basic AI product patterns. But as researchers continue to produce breakthroughs, I expect to add more patterns to this list. Currently the AI product patterns include computer vision, natural-language processing, next-in-sequence predictions, and collaborative filters. These address problems that arise in a variety of business contexts.

Product pattern 1: Computer vision

Computer vision applications use software to generate a high-level understanding of digital images and videos. "Describe what you see" is a programming challenge that has vexed computer scientists for decades. Governments, corporations, and private investors have spent billions trying to advance the state of the art because the potential payoff is so large. Interpreting images—to accomplish tasks such as driving a vehicle, selecting the best photo for a story, and scanning the horizon for an enemy vessel—is one of the most manually intensive and expensive business processes.

Historically, computer vision solutions haven't worked very well, and few made it out of the lab.

Solving computer vision problems

In 2014 researchers began making breakthroughs in computer vision by using deep learning models called convolutional neural networks (CNNs). Not only did CNN researchers achieve far better results in annual computer vision competitions, they also published CNN models which could generalize to many other computer vision problems.

Three years later Apple released a facial recognition solution in iOS 10 using the same approach. AI researchers are now achieving better-than-human results on computer vision tasks.

"Dogs vs cats" is a classic computer vision challenge where developers attempt to build an algorithm which can automatically classify an image as containing a cat or containing a dog. As recently as five years ago, the best researchers could achieve no better than 80% accuracy. Today novice machine learning engineers can achieve better-than-human results using AI.

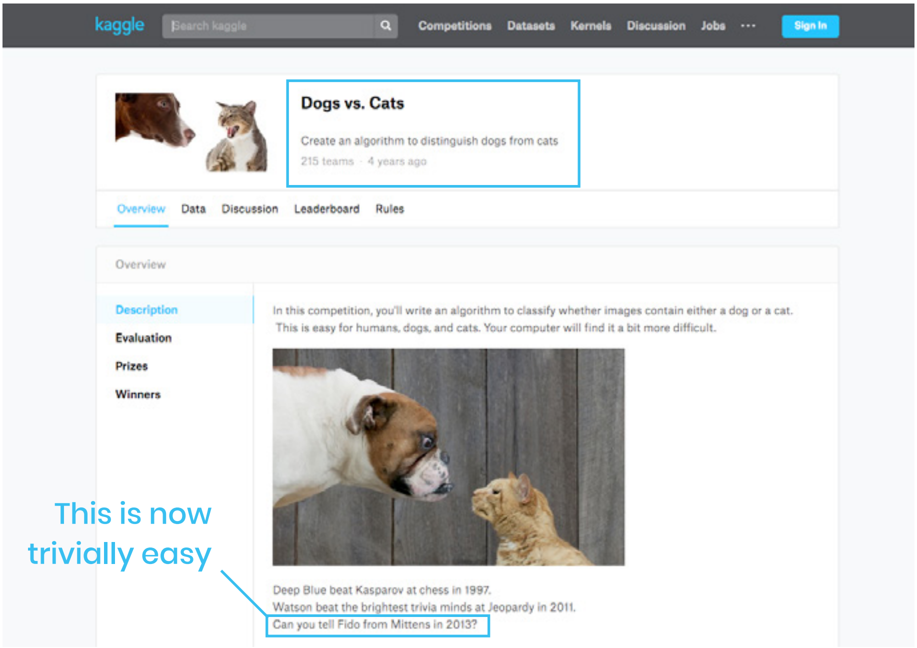


Figure 3

Computer vision applications

As advances continue, computer vision now makes significant contributions to applications like these:

- Classifying images
- Identifying an object in an image
- Retrieving a specific image from a large set
- Image restoration (e.g., removing the background noise in an image)
- Biometrics (e.g., facial recognition, walking gait recognition)

Figure 3: https://www.kaggle.com/c/dogs-vs-cats

In the following example from Google researchers, a computer vision model identified specific objects in an image and assigned confidence scores to their predictions.

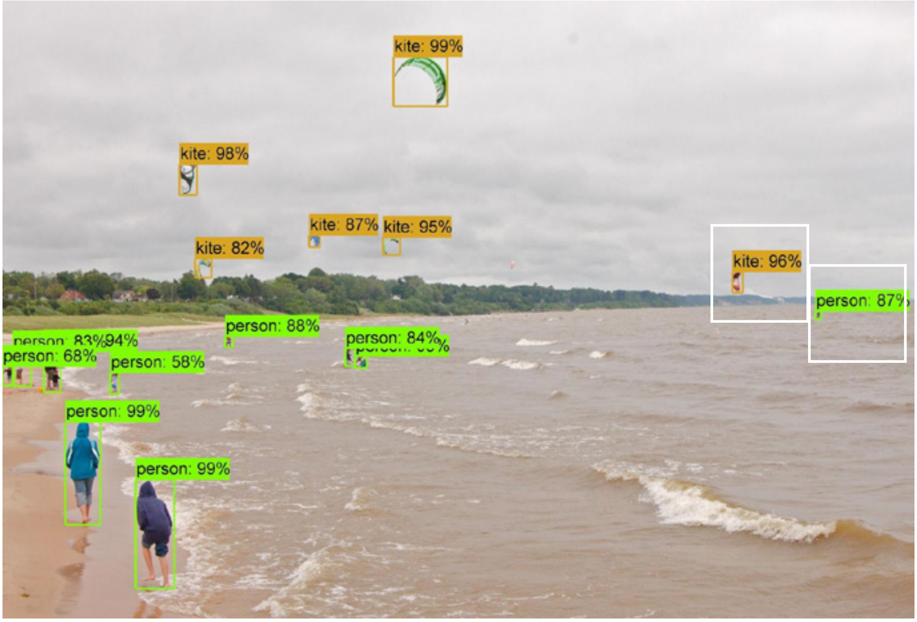


Figure 4

Take a closer look at this example. Some of the kites or people are pretty easy to identify, but how about those on the far right? Why can the computer predict with 96% certainty that the one small black blob is a kite and with 87% certainty that the other black blob is a person? *The deep learning algorithm learned that kites don't usually float on water and that people don't hover in the sky.*

Now imagine a future application which uses video, multiple cameras from different angles, sound, and multispectral (e.g., ultraviolet) images to power deep learning algorithms. The image detection capability would make Superman jealous.

Figure 4: https://research.googleblog.com/2017/11/automl-for-large-scale-image.html

AI opportunities for computer vision solutions

Computer vision is the most mature AI product pattern. Your AI strategy should start by looking for business processes where you can apply the computer vision product pattern. First identify business processes in which images are already used, and then consider processes that could benefit from the use of images.

Look for processes where images are already used

If you have business processes which depend on people looking at images, you are probably already falling behind your competitors. Are analysts reviewing satellite images? Do customers send you pictures? Do you check user identification before providing access? Do your employees take pictures while inspecting equipment?

Consider any business process where images or videos are already used. In most cases you will be able to identify the goal of the business process. This goal should align with an output. For example, suppose customers send you images of damaged products as part of a warranty claim. Claim-processing specialists review these images and approve or reject the application. The inputs to this business process are the images (and other claim data), and the output is the approve/reject decision.

Add imagery to existing processes

The proliferation of inexpensive drones, satellites, smartphones, and cameras has led to an explosion in available images. Computer vision techniques allow you to leverage these new data sources to improve existing business processes. For example, suppose you are designing the Intelligent Vacation Planner described in the introduction. You could use computer vision to ingest images in a customer's Instagram feed and predict their hobbies.

Training data for computer vision applications

Your biggest operational challenge will always be building training data for your AI algorithms. You'll want to know what to look for in your existing data and how to prepare it for use in an AI algorithm.

Inputs are the images themselves, along with any associated metadata (customer ID, timestamp, location, etc.). If your existing business process uses images, you may already be generating outputs—assigning tags, putting images into folders, circling objects, etc. If not you probably don't have data that's organized well enough to train computer vision models. To use your images for this purpose, you'll need to label them. The costs of image labeling vary depending on the domain. You can outsource simple image labeling to crowdsourcing services like Amazon Mechanical Turk for pennies per image. More complex image labeling will require the time of expert analysts.

For example this chest x-ray has been labeled by a professional radiologist who identified a region containing pneumonia. Paying skilled radiologists to label x-ray images is obviously more expensive than paying someone to determine whether a picture contains a cat.

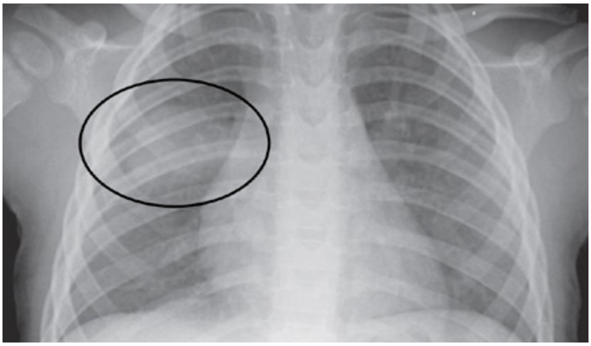


Figure 5

Unfortunately there is no easy way to know how much data you need. But data scientists have developed techniques for getting good results from ever smaller quantities of training data. The most useful technique in computer vision is transfer learning, a process that starts with a model already trained on another set of images. Other techniques are image generation, semi-supervision, and pseudo-labeling.

You will need less training data if you can start with an easy task. For instance, separating images into two categories (e.g., interesting and not interesting) takes less training data than categorizing images into 1000 categories.

Figure 5: http://www.stepwards.com/?page_id=10460

Product pattern 2: Natural-language processing

Natural-language processing (NLP) applications process and interact with human-generated (i.e., natural) language data. In 2017 you probably noticed your smartphone getting better at recognizing your speech and turning it into text. Speech recognition is an example of NLP.

NLP is another classically hard computer science problem which has vexed researchers for decades. Although AI advances in NLP lag behind those of computer vision by two or three years, researchers are making rapid progress with deep learning.

Examples of NLP solutions

NLP research continues to yield results in the following applications:

- Machine translation (e.g., Chinese to/from English)
- Speech recognition
- Speech generation
- Entity recognition/extraction
- Text generation
- Text summarization
- Text classification
- Real-time chatting (chatbots)
- Sentiment analysis

To date, deep learning has made the biggest impact on traditional NLP problems like machine translation and speech recognition. The large tech companies are investing tremendous resources building training data for mass-market services. But you don't need to invest in solutions the way they do, since you can use whatever solutions they create.

For example, in 2016 Google released a new version of Google Translate built with deep learning. Users instantly recognized a breakthrough, and many observers consider it to be the moment when practical AI arrived.[6]

Figure 6

Now, instead of building your own machine translation solution from scratch, you can just use Google Translate (APIs).

Your best bet for NLP applications

Before 2018 there were few practical applications for NLP in most enterprises, but the tools and research are advancing rapidly. Now is a great time to explore the many ways NLP will impact your company—because the potential is massive.

How many professionals spend most of their days interacting with natural-language data? Reading, summarizing, categorizing, or generating documents? Writing reports? Answering routine email? A lot.

Consider the US federal government. Seventy percent of the US federal workforce performs professional or administrative tasks, and less than 50% of these employees have a college degree. Many of these tasks will be replaced with AI-powered NLP, and you can probably find many similar applications for the technology in your company.

NLP will also increase human productivity when we offload routine activities to AI. Imagine how much time you can save when a computer can read, categorize, and summarize your email or attend a meeting and send you a summary that includes the 10 most important sentences.

Figure 6: https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html

The opportunities for NLP are endless, but you're not reading this book because you care about distant-future functionality. You want to know how NLP will impact your company in the next 18 months. Document classification is the most likely answer.

Most companies spend significant resources reading and classifying documents. For example:

> Processing forms and claims
> Prioritizing inbound sales or support inquiries
> Reviewing resumes
> Reviewing email or electronic documents for regulatory compliance
> Legal discovery
> Reading applications (e.g., college essays)

Practical, inexpensive NLP techniques for classifying documents are evolving rapidly, and you can expect many to begin hitting the enterprise playing field in 2018.[6] Begin your AI strategy by looking for places where people are currently reading and classifying documents.

Training data for NLP

Generating training data can be more challenging in NLP than in computer vision, for several reasons:

> Reading, translating, categorizing, and summarizing a document is often tedious and subjective.
> Documents inherently provide fewer inputs than images do. A page of English text contains about 1500 characters while a 500x500-pixel image contains 750,000 pixels.
> NLP models are more difficult to pretrain on other datasets.[7]

---

6 http://nlp.fast.ai/classification/2018/05/15/introducting-ulmfit.html

7 In 2018, researchers are making rapid progress developing techniques for using pretrained models in NLP tasks. See https://arxiv.org/abs/1801.06146

For these reasons you will probably want to start looking for NLP applications by considering existing business processes where outputs are already being generated. For example, if you want to automatically process forms, start by identifying outputs generated by the people who currently read and processes these forms. The inputs are the text and other data in the forms.

Product pattern 3: Next-in-sequence prediction

Our third AI product pattern, next-in-sequence prediction, does not fit into the domain of traditional computer science challenges like computer vision and NLP do. You probably won't hear about it at conferences. Few AI researchers will recognize the importance of the problems it solves, and even fewer will write papers about it. Nevertheless, it is probably the most practical, actionable product pattern because even modest results can generate clear ROI.

Next-in-sequence methods address the common business problem of predicting "next" results based on previous results in a structured dataset. Traditional machine learning has been used in next-in-sequence problems like detecting credit-card fraud for decades. Recently developers have achieved better results using deep learning techniques which can uncover complex relationships in data.

Examples of next-in-sequence applications include the following:

- Predicting future sales of an item based on past sales
- Classifying a log entry as a critical system error or a normal event
- Predicting crowd size for an event based on previous events
- Detecting fraudulent credit card transactions
- Detecting money laundering in financial transactions
- Detecting anomalous events from remote sensors
- Predicting which users will buy products or cancel subscriptions

All of these applications require a lot of structured data, and you might be surprised how much of this sort of data you already have access to.

Structured data

The term *structured data* can have many different meanings. In this context structured data refers to the kind of data that is found in a database table or a parsable format like a comma-separated values (CSV) file. This data can be either continuous or categorical.

Continuous data has (theoretically) infinite possible values. Examples are prices, task completion time, and temperature.

Categorical (or discrete) data has one of a finite set of values. Country, state, and blood type are categorical values.

For our purposes the theoretical definition of structured data isn't important. If the data looks like it belongs in a database, it is a good candidate for next-in-sequence methods. Conversely, data like text documents and user-generated string variables (e.g., tweets, web comments) is a better candidate for NLP. Images are not normally found in databases and are better candidates for computer vision methods.

High data volumes

Since most business applications already have tons of data in databases, you will encounter many opportunities for next-in-sequence approaches. Sensors, Internet of Things (IoT), log files, sales events, and online user behavior events are all good candidates for next-in-sequence applications.

Sales and Marketing Tables | Sensor Time Stamped Events | Server Logs | Online User Behavior

How to spot AI opportunities for next-in-sequence predictions

A few techniques will help you identify how to apply next-in-sequence AI to your business processes. These include reviewing your data dictionaries and key performance indicators (KPIs) and considering new data sources that could enhance predictions for your existing business processes.

Review your existing data dictionaries

Your product teams may already have documents which describe the fields in your databases. These documents are usually called data dictionaries or metadata repositories. They are designed to be read by a human being who wants to know what is in the database and what the fields mean.

You can identify opportunities for next-in-sequence predictions by simply reading the data dictionaries or having someone brief you on every field. Start with the databases currently used for reporting or metrics. You know the team that uses Tableau to produce pretty reports? Start with the database they use.

For example, consider the data dictionary for sales at Corporación Favorita,[8] Ecuador's largest grocery store[9]:

---

[8] http://www.corporacionfavorita.com

[9] Find this data dictionary at https://www.kaggle.com/c/favorita-grocery-sales-forecasting/data

Field	Definition	Example
id	Unique ID for every row in the sales database	0
date	UTC Unix timestamp of the date when the sales occurred	2013-01-01
store_nbr	Unique ID for each grocery store	25
item_nbr	Unique ID for every item sold	103775
unit_sales	Number of items sold at the store on the day of the sale	13.0
onpromotion	Boolean flag identifying whether the store was running a promotion on that item on the day of the sale	False

In this table, can you identify inputs and outputs for a next-in-sequence model? To identify an output, consider what result you might want to predict. The best candidate for helping you plan your inventory is *unit_sales*.

What plays into that prediction? The inputs here are *date, store_nbr, item_nbr,* and *onpromotion* since a model may be able to use each to predict *unit_sales*.

How about *id*? It isn't useful as an input or output.

Review your KPIs

Do your managers already use key performance indicators (KPIs) to track and manage their operations? Do they track same-store sales, the number of documents processed, or cases closed per month? These sorts of KPIs are often derived from high-volume business events, so they can make good AI outputs to help you predict future performance or operational needs.

Consider new data sources

New data sources are constantly emerging. Look for unobvious data sources from third parties which may predict future events. Examples of these sorts of data sources include weather, commodities pricing, and consumer credit scores. These data sources could be inputs for your next-in-sequence AI models.

Training data for next-in-sequence models

Finding enough training data generally isn't a problem for next-in-sequence applications. You won't need people to hand-label or generate new data, because outputs can be derived from the data. For instance, Corporación Favorita's data dictionary contains all necessary training data; in this case you would just train your model to predict past *unit_sales*.

Your challenge is picking the right data and organizing it to allow for efficient model training. Overcoming this challenge requires data analysis and engineering work.

Many output options

Next-in-sequence systems often have many potential outputs. You must identify the best candidates among them. For example, you could predict the sales volume for a particular item in a store. Or you could predict sales for the whole store or sales turnover for a region or time until the next sale.

Identifying the right output requires detailed exploration of the data and knowledge of the business processes. Ideally a cross-functional team including analysts, data scientists, programmers, and operations managers will collaboratively select the best outputs.

In practice? For me it is usually faster to build a dozen models which predict every potentially useful output and then begin testing to identify which output creates the most value.

Input feature engineering

Once you pick candidate outputs, your data scientists or machine learning engineers will have to invest time exploring candidate inputs. For example, you may have to derive new inputs such as "average time between anomalous events" for every sensor.

The process of creating new inputs is called *feature engineering*. (Recall from Part 1 that inputs are also called features.) Your data scientists will be familiar with these techniques. Feature engineering can require substantial engineering resources and could be the most expensive part of your next-in-sequence application.

Feature engineering is not normally required in many deep learning systems that use computer vision or NLP. As a result many deep learning advocates claim that feature engineering is no longer necessary for *any* deep learning application. But for next-in-sequence applications this *currently* isn't true.

For example, consider the feature engineering required for this sales database of a convenience store:

Date	Beer sales
2013-01-01	10
2013-01-02	7
2013-01-03	8
2013-01-04	49
2013-01-05	112

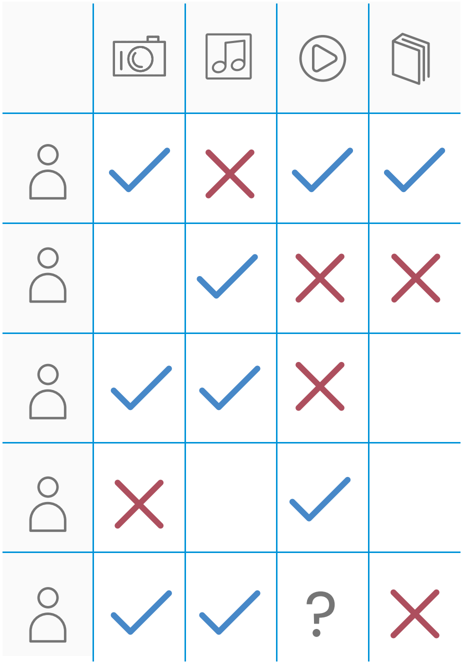
Why do beer sales vary so much by date? The answer isn't obvious from looking at this small dataset until we perform some simple feature engineering. Let's try deriving the day of the week for each date:

Date	Beer sales	Day of week
2013-01-01	10	Tuesday
2013-01-02	7	Wednesday
2013-01-03	8	Thursday
2013-01-04	49	Friday
2013-01-05	112	Saturday

Convenience store beer sales are higher on Fridays and Saturdays—not exactly an earth-shattering revelation for anyone who has visited a 7-Eleven on a Saturday afternoon.

You can imagine that other factors such as weather, holiday schedules, and paydays would all be potentially useful inputs. If your database doesn't include these inputs, you will have to buy or derive them through feature engineering.

Product pattern 3: Collaborative filtering



Our final AI product pattern currently applies to a smaller niche of applications than the previous three patterns. Collaborative filters make predictions about user behavior by collecting behavioral events for many users. Online recommendation systems are the most common example of collaborative filter applications.

Consider a service which sells digital products like pictures, music, movies, or e-books online. The service wants to boost sales by recommending the most relevant product to a user. Collaborative filters make these recommendations by comparing purchases among users who have similar interests.

My wife likes watching romantic comedies (aka romcoms) on Netflix. I hate romcoms. Whenever she turns on a romcom I throw a temper tantrum like a five-year-old (unless it stars Vince Vaughn). A collaborative filter can identify preferences like ours as data relationships and make better recommendations so we can both enjoy watching a movie together.

Other examples of collaborative-filter applications include:

- Content recommendations (e.g., Netflix movie queues)
- Digital advertising campaigns
- E-commerce product recommendations

Like next-in-sequence predictions, collaborative filters primarily use information that's already available in your databases. You will have to use similar techniques for building your training data. Traditional approaches to collaborative filtering suffer from challenges of sparse data, scalability, and adaptability. Deep learning models overcome many of these challenges.

How to spot AI opportunities for collaborative filters

When you're considering how collaborative filters can improve your business processes, look for processes which give users (or any entity) many competing choices and which record their decisions.

Training data is not normally a big challenge for collaborative filters. A more challenging problem is dealing with new users who have no history of past choices in your system.

Collaborative filters vs next-in-sequence predictions

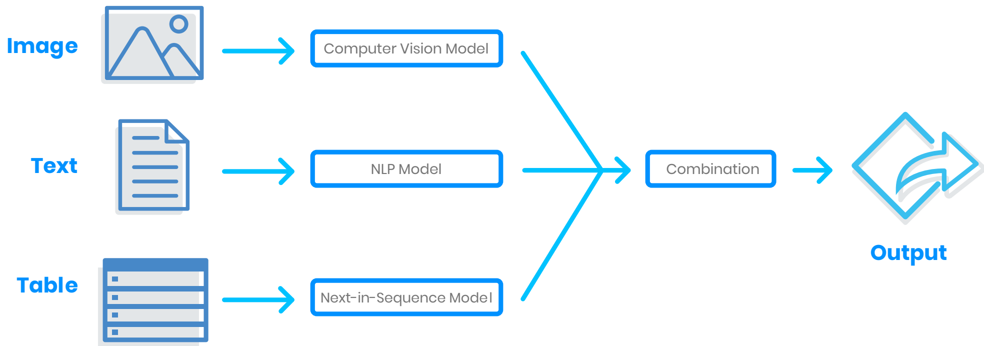
In practice, collaborative filters and next-in-sequence predictions require similar engineering efforts. The two approaches have a lot in common. In fact, either collaborative filters or next-in-sequence predictions can be used to build solutions like recommendation engines.

The distinction between the two model approaches is time—next-in-sequence prediction models make predictions based on a time series of events. For example, a content recommendation engine built with next-in-sequence models would track timestamped events of online user behavior (pages visited, past purchases, mouse movements, etc.) and make content recommendations based on that behavior. Collaborative filters make recommendations based on the behavior of other users, without regard to time.

In practice many deep learning applications are a hybrid of collaborative filters and next-in-sequence predictions.

Combining product patterns

Neural networks are incredibly flexible, and researchers are always discovering new ways to design them. Current deep learning development tools can handle multi-input and multi-output models. Thus you can combine different AI product patterns in a single system. The system outlined in the following image includes the computer vision, NLP, and next-in-sequence product patterns:

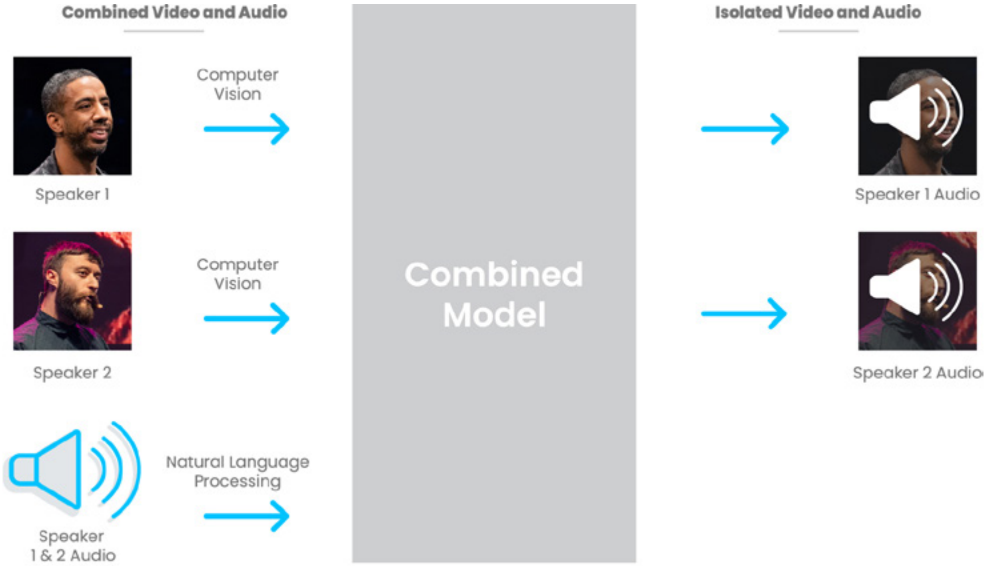


Example: Google's Looking to Listen

Google has recently demonstrated how to combine computer vision and NLP product patterns.[10] Do a quick YouTube search for "Looking to Listen" to see an example.

[10] https://ai.googleblog.com/2018/04/looking-to-listen-audio-visual-speech.html Also see the paper "Looking to Listen at the Cocktail Party," presented at SIGGRAPH 2018 (https://arxiv.org/abs/1804.03619).

In the following image of Google's Looking to Listen model, you can see that the convolutional networks used for computer vision are connected to an LSTM, a popular NLP model.



Using the computer vision and NLP product patterns, Google demonstrates how AI can mimic the "cocktail party effect": the human capacity to isolate a voice in a noisy room by looking directly at the speaker.

Applying AI product patterns to business solutions

The realm of AI research covers far more than the product patterns we've discussed here. AI researchers make daily breakthroughs, and some will lead to new practical applications. But—as is common with primary research—most discoveries will never make it out of the lab.

Colleagues may discuss the following AI applications that are on the horizon:

> Adversarial networks
> Generative models
> Unsupervised learning techniques
> Reinforcement learning

Some of these discoveries get enormous coverage in the business press. For example, Google's AlphaGo stunned the world by beating world-champion Go player Lee Sedol in 2016. Two years later I know of no single commercial application built on AlphaGo technology. Don't waste resources on these bleeding-edge AI innovations unless you have a compelling, specific business case. If you decide to pursue them anyway, you'll need to build your own AI research team to help advance the state of the art.

INSPIRATION FROM OTHER PROJECTS

The four product patterns are a great starting point for identifying AI opportunities in your organization. I use them as a first step in sorting through a client's data and business processes. After laying this foundation I try to learn how other companies are applying AI to business problems that are similar to my client's problems.

The usual resources like conferences, PR, blog posts, and job descriptions can provide valuable insights about your competitors' current AI initiatives. In addition to these, I also rely on a few online resources for inspiration. These resources include Kaggle, AngelList, and arXiv.

Kaggle

Kaggle (now owned by Google) hosts competitions in data science and machine learning. Many of the competitions are based on real data submitted by companies that are looking to crowdsource their innovation.

Kaggle is my go-to resource for preparing an AI strategy. I can learn more about a new industry in three hours on Kaggle than in three days at a conference.

Look into Kaggle to explore:

- Examples of real-world problems
- Real datasets
- Current trends in the state of the art

All of this information can inform your AI strategy, but Kaggle can be a bit intimidating if you don't have a data science background. To cut your teeth here, start by exploring Kaggle competitions and data, and read the discussions among competitors.

Explore some competitions

Start by visiting Kaggle.com/Competitions to look for problems similar to those at your company. Here are a few examples of the kind of competitions you'll find on Kaggle:

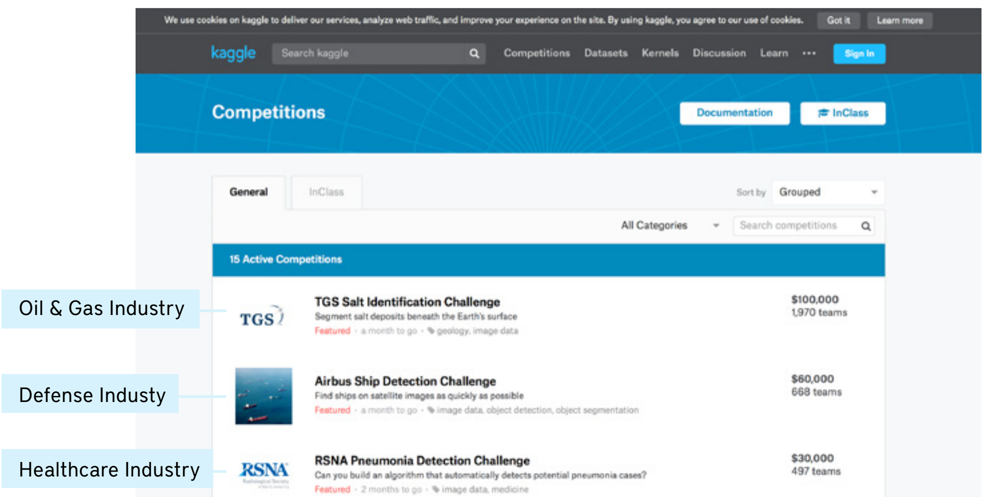


Figure 7

Figure 7: https://www.kaggle.com/competitions

Download the data

You can take a look at the competition data after you register and accept the terms of the competition. Most data is in common file formats like CSV, TXT, or JPG. Just download and open it in applications like Microsoft Excel or Word. Ask yourself whether you have similar data in your company.

Read the discussions

Kaggle competitors talk about their results, share code, and perform exploratory data analysis (EDA). Many share their results. In just a few hours of reading you can get a very good sense of what is possible.

Have your engineering team re-create the best results

If you are just setting up a new AI engineering team, you can prime their creativity and infrastructure by re-creating the Kaggle contest scenario and challenging them to achieve the same results. Look for a completed contest where the winning team published their results. The exercise will accelerate your team's progress by forcing them to get their infrastructure working and will familiarize them with best practices.

AngelList

AngelList (Angel.co) is web site where startups can connect with talent and investors. Most startups create an AngelList account, so it's a good clearinghouse for emerging technology that investors are funding. It's also a great source for learning about emerging AI ideas.

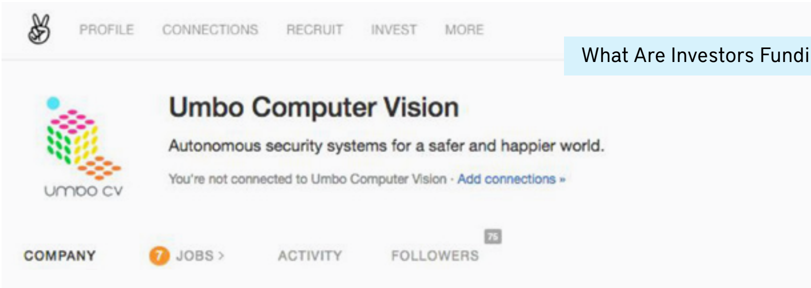


Figure 8

arXiv

You'll find an online repository of scientific papers at arXiv.org. While most of the information on arXiv is too speculative or untested for practical solutions, you can get an understanding of what problems researchers are exploring. Sometimes they reference specific datasets or results of their studies, and you might find these helpful. For example, Google's "Looking to Listen" research results are available on arXiv.

Figure 9

Figure 8: https://angel.co/umbocv
Figure 9: https://arxiv.org/abs/1804.03619

PART 2 QUIZ

1. Reread the Intelligent Vacation Planner example in the Introduction. How would you apply each of the four product patterns to this product?

2. Suppose your engineering team is looking for the most effective and fastest way to solve a problem. Would you suggest starting with Kaggle or arXiv?

Find quiz answers in Appendix 2.

“The world’s most valuable resource is no longer oil, but data.”

– ***The Economist***

Part 3

BUILDING A WINNING AI STRATEGY

You now know what AI can do and how to generate ideas for AI opportunities. With a bit of effort you will quickly discover more potential AI applications than you have the resources to pursue. But which ideas should you pursue first? Which have the biggest potential impact? The biggest risk?

Unfortunately there are no easy answers to these questions with AI products. One clear benefit of traditional software development over machine learning is predictability. You don't need to achieve fundamental breakthroughs in traditional software engineering to build a large, complex system. The path forward is clear. With traditional software development, almost all business risk is based on labor costs and market demand for your solution.

In AI software development the risk is that you don't know how well your AI models will work until you test them. You must first build a set of training data and put your data scientists to work testing the models before you know the answers to questions like these:

- How effectively does the model categorize unstructured messages?
- What percentage of objects can the model correctly identify in an image?
- How accurately can the model forecast prices?

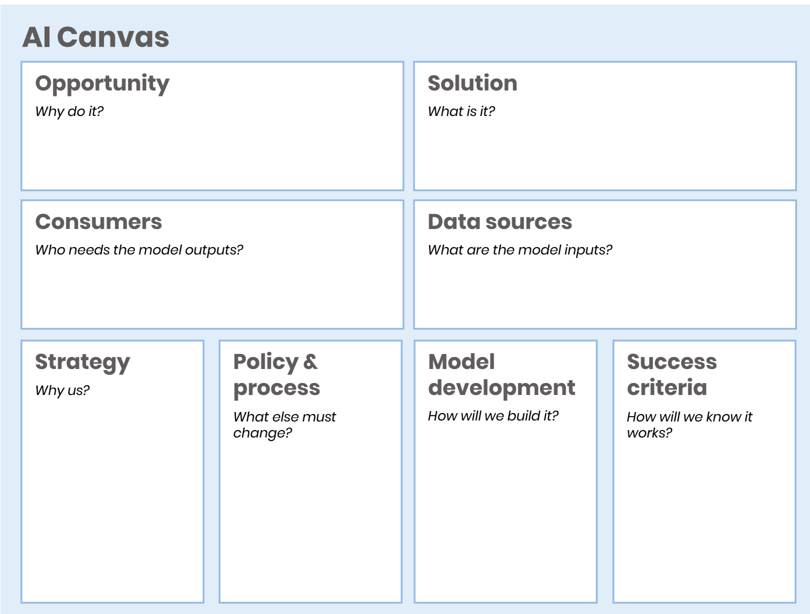
Tech giants like Amazon, Microsoft, Apple, Facebook, Baidu, and Google have mitigated the risks of AI development by throwing money at it. They race to build AI infrastructure and attempt to execute almost every idea—a strategy their leadership (and stockholders) endorse. Their business hinges on the generation of innovative algorithms, and their fundamental AI breakthroughs can return billions of dollars in revenue. For them, money truly is no object.

Well, you're not Google. Your business doesn't generate billions of dollars in free cash flow, and your (traditional) competitors are burdened by the same regulatory and infrastructure challenges as you are. Furthermore *you don't work in a culture which is comfortable with the repeated failures necessary to achieve fundamental breakthroughs*. In your field, "fail fast" isn't a sound strategy for optimizing annual bonuses.

You need tools for filtering options, placing bets, and building adequate consensus. And since some of your bets won't work, you need the top cover to change direction without fear of blame. In Part 3 you'll learn how to navigate these challenges.

USING THE AI CANVAS

Prolego created the AI Canvas to help you filter your many AI ideas and identify the best business opportunities. The AI Canvas is based on the successful Business Model Canvas[11] by Alexander Osterwalder and on the Lean Canvas[12] by my friend Ash Maurya. Organizations from part-time startups to the world's largest companies have used these to replace business plans as their strategic planning tools. Use the AI Canvas to evaluate your potential AI models.

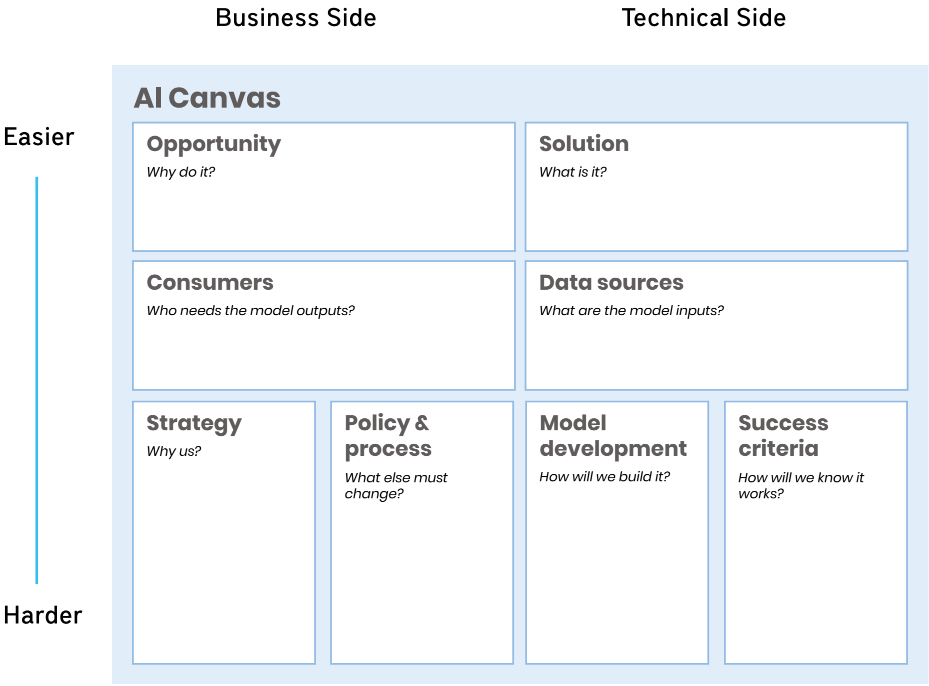


How to use the AI Canvas

With your AI product ideas in mind, start at the top of the AI Canvas and fill out each block with a few bullet points or sentences. The left side of the canvas addresses business strategy issues. The right side raises questions of technical feasibility. The issues are increasingly complex as you move from the top of the canvas to the bottom.

---

11 https://strategyzer.com/canvas/business-model-canvas
12 https://leanstack.com/leancanvas



Canvas advantages and limitations

Canvases have several advantages over business plans:

> They can be easily updated as plans change.
> They allow for efficient team planning.
> They instantly communicate key risks and opportunities for any audience.

These features make a canvas ideal for analyzing, planning, and retooling your AI strategy.

I've been using canvases for over a decade on my own startups and client initiatives. They take far less time than a massive business plan or strategy white paper (which nobody will ever read). But canvases *only simplify* the process of documenting and communicating a strategy. You'll still need to face the bigger challenge of gathering the information necessary to thoroughly explore your options. Canvases help us ask the right questions, but they don't provide the answers.

With this caveat in mind, let's explore the AI Canvas in the following example:

Automated claims processing

Let's suppose we're considering developing an AI-driven automated claims-processing system. Many industries process claims:

- Insurance companies process claims against policies.
- Governments process claims for social benefits.
- Manufacturers process claims for warranty coverage.

All of these industries follow a similar business process. Consumers submit a claim and ask for reimbursement. The organization must choose to accept or deny the claim based on the governing rules and information in the claim application.

Using the claims-processing example, let's walk through each block of the AI Canvas.

Opportunity: Why do it?

Answering the *why* for any proposed application is simple enough. We just need to provide a brief business value proposition. Common justifications for an AI opportunity are increased efficiency, faster decision cycles, staff size reductions, or increased revenue.

Example

Most claims are processed through the manual effort of multiple people who review every case before making a decision. In this system, money is sometimes lost through fraudulent claims, and honest applicants can become frustrated with slow payment cycles.

Here's what we write in the Opportunity box:

Opportunity
Why do it?

Reduced operating costs and fraud losses. Improved customer satisfaction.

Solution: What is it?

Identifying the *what* of our plan should also be relatively straightforward. We just need to describe the solution at a high level and identify any product patterns or outputs.

Example

Most claims include both structured data (such as claim numbers, dates, agents, policy numbers, and products) and natural-language text data from customers or processing agents.

Here's what we write in the Solution box:

Solution
What is it?

For every claim, automatically assign reject/accept along with a confidence score.
Models will be based on next-in-sequence and NLP product patterns.

Consumers: Who needs the model outputs?

As discussed earlier, identifying your AI model outputs is critical for building training data. It is also critical for surfacing the people and systems that will use the outputs. Will the model's output feed another system or model? Will it trigger an automated process? Will downstream business processes have to change? Do those systems have programmable interfaces to receive the outputs?

Downstream adoption is one of the biggest challenges for AI systems. Don't underestimate the challenges of getting business processes and systems to use the model outputs.

Example

Most claims departments have basic workflow management tools which generate PDFs from forms, put metadata into databases, and track cases which move through the claims process. The AI models can send results either to the workflow tools or to a new column in a database.

Building the technical interface takes no more than a few weeks of work. More challenging are the associated operational changes. How will the organization gradually transition from a fully manual human review process to one where algorithms automate claims review? Will a manager need to check each automated claim review?

For the purpose of this exercise we'll make an assumption about the organization's preferences.

Here is what we write in the Consumers box:

Consumers

Who needs the model outputs?

Results will be written to the process management system's application interface for claims. Managers will set thresholds for automated processing based on claim size and confidence scores. A percentage of claim reviews will be checked manually.

Data sources: What are the model inputs?

As emphasized previously, access to quality data is the most important factor in the success of your AI application. Identify the model's input sources in the AI Canvas's box for data sources. Your description should include any known complexities or challenges involved with the inputs.

Example

Most of the inputs are straightforward. The model will use the data that the human claims reviewers use when deciding whether to accept or reject a claim.

You can probably imagine other data sources for detecting potential fraud. Credit scores are one of these sources, and you can buy them from consumer credit services. Another possibility is the social media behavior that some consumers publicly share on sites like Instagram or Twitter.

Human analysts may not have the time (or training) to process data from many emerging data sources, but algorithms can churn through them in milliseconds. The benefits of automation may offset associated costs of the new data services.

We describe our available data in the Data Sources box:

Data Sources

Who needs the model outputs??

All available data currently used in claims processing. Forty percent of claims applications are handwritten and may need to be hand-keyed or processed through OCR. Additionally, commercially available credit scores can be used to detect fraud.

The final four boxes of the AI Canvas are more challenging and normally require some research.

Strategy: Why us?

AI technology is changing rapidly; successful new products and services may require years of ongoing investment. Nontraditional competitors such as Amazon, Google, and Apple are using AI to compete in new markets. AI startups are disrupting competition.

With so much happening it pays to ask the most important strategic question: Why us? Of course you can ask the same question about any new business opportunity, and we don't need to elaborate on the usual considerations such as brand position, market growth, core competencies, and customer relationships. Include these in the Strategy box as appropriate. Here we'll focus on the only source of long-term competitive advantage in AI: data. **Ultimately data is the only source of sustainable competitive advantage in AI.**

You will need a strategy to generate more training data through partnerships, new products, or research. Each new data source affords you the opportunity to retrain your models and build a better product.

Example

Processing claims more efficiently may or may not be a competitive advantage. In the best case the claims data will give you more information about how customers are buying and using your services—insight which could help you build and price better products.

Obviously this is true for product company warranties and insurance companies who may be able to offer lower prices or to offer purchasing incentives for customers who are unlikely to file claims. But other organizations may view claims processing only as a cost center. Governments might be happy to reduce operating costs by outsourcing the processing of unemployment claims to third parties.

For our purposes, we'll assume claims processing provides strategic value.

So in the Strategy box, we write the following:

Strategy
Why us?

Price and market our services more accurately than competitors. Fast processing of legitimate claims can be used to retain large accounts.

Policy & process: What else must change?

Your company's general counsel may have concerns with how you use your company's data. There are the obvious privacy and social implications that have been well documented in the mainstream media over the past few years. Your user and privacy agreements may need to be updated. Your security and data governance policies may need to be modified to allow new systems to access the data. These policy changes are relatively straightforward compared to a bigger challenge: data usage rights.

You and your customers may store data from third parties for which you have limited usage rights. The contracts which govern these rights can be extremely complex, so most organizations adopt draconian data governance policies to avoid legal issues. For example, legal advisors might instruct product teams to use data for only narrow business processes and to restrict access.

But these contracts were created in the era before AI. All of your data now has the potential to generate unforeseen business value. I regularly meet managers who tell me they can't pursue an AI business opportunity because their data usage rights are restricted. In these instances I escalate the issue to executives who are in the position to reinterpret or renegotiate contracts.

In the AI Canvas, data access complications are the sort of policy, security, and legal issues you should call out in the Policy & Process box.

Example

The organization's user, privacy, security, and data governance policies will need to be reviewed and updated to ensure the data can be used in automated claims processing. Customers will probably need to opt in to allow the organization to pull data from third parties.

In the Policy & Process box we write the following:

> **Policy & Process**
> *What else must change?*
>
> Review user, privacy, security, and data governance policies. Ensure marketing contracts allow us to use customer data for automated claims processing.

Model development: How will we build it?

In the Model Development box we identify any relevant insights into models or training data. For example, we would want to point out new research that makes the solution more feasible or existing data sources that would accelerate our model training. Factors that might slow the development of our model should also be considered. For instance, if images must be manually labeled or other labor-intensive work is necessary to prepare training data, those costs should be spelled out in the Model Development box.

Example

Our solution will require next-in-sequence and NLP models. For our next-in-sequence models, the structured data doesn't require any innovation. To process the natural language data in claim forms and other documents, we can use the emerging NLP document (text) classification techniques we discussed in Part 2, so in the Model Development box, we write the following:

Model Development
How will we build it?

Use best practices to process structured data for next-in-sequence predictions. Develop NLP text classification models based on emerging techniques.

Success criteria: How will we know it works?

Knowing success criteria is critical at the outset of any project. For example, you might need to hit a particular performance metric for a test dataset before you can deploy the AI solution. If you know those metrics, identify them in the AI Canvas's Success Criteria box.

Your success criteria evaluation should also identify broader business goals such as reducing headcount or increasing revenue. Key performance indicators (KPIs) and qualitative feedback might also play into your criteria for success.

Example

The operational cost-saving success metrics for our organization are straightforward: reduced labor costs and faster processing time. Fraud reduction is harder to measure since an organization doesn't have a good baseline for current fraud. We add the following success criteria to our AI Canvas:

Success Criteria
How will we know it works?

Less manual labor required to review claims. Faster average claim processing time.

Final automated claims processing canvas

Having completed all of the boxes on the AI Canvas, we can now explore the final product. This one-page canvas presents the major issues and questions for our AI strategy, and it fits neatly into an executive briefing document:

AI Canvas—Automated Claims Processing

Opportunity

Reduced operating costs and fraud losses. Improved customer satisfaction.

Solution

For every claim, automatically assign reject/accept along with a confidence score. Models will be based on next-in-sequence and NLP product patterns.

Consumers

Results will be written to the process management system's application interface for claims. Managers will set thresholds for automated processing based on claim size and confidence scores. A percentage of claim reviews will be checked manually.

Data sources

All available data currently used in claims processing. Forty percent of claims applications are handwritten and may need to be hand-keyed or processed through OCR. Additionally, commercially available credit scores can be used to detect fraud.

Strategy

Price and market our services more accurately than competitors. Fast processing of legitimate claims can be used to retain large accounts.

Policy & process

Review user, privacy, security, and data governance policies. Ensure marketing contracts allow us to use customer data for automated claims processing.

Model development

Use best practices to process structured data for next-in-sequence predictions. Develop NLP text classification models based on emerging techniques.

Success criteria

Less manual labor required to review claims. Faster average claim processing time.

A few of our next steps are obvious:

- Investigate new third-party data sources.
- Review existing legal agreements.
- Consider the strategic long-term implications.
- Develop an operational plan for using the model outputs.

BUYING VS BUILDING

Should you buy an AI solution from a vendor or build your own internal capabilities? I've been involved with buy-vs-build software decisions for my entire career. I've worked at companies that squandered millions of dollars developing an in-house, proprietary system before finally deciding to buy a product from a vendor. I've also worked at companies who lost market share by outsourcing a core competency to technology "partners."

Before I share my advice about the buy-vs-build question, let's first talk about the challenges facing AI vendors. Having founded and invested in AI companies myself, I'm quite familiar with them.

The challenge for AI vendors

Traditional software products have tremendous scaling power. A company invests a fixed set of engineering resources and then has an asset it can sell repeatedly.[13] That's why successful software products have such high profit margins. Customers benefit from this investment by getting a product for a fraction of the cost that would have been required to build it themselves.

AI products have fewer upfront engineering requirements than traditional software products do. Building a successful AI product requires three assets: models, data, and infrastructure (e.g., GPUs). Let's consider each one.

Free models

At the moment, AI researchers worldwide—including those at the largest tech giants—are racing to publish breakthroughs. Even secretive Apple publishes its AI research.[14] Why would these companies give away their new insights? For social good? Hardly.

Research is happening so fast that these companies benefit more from collaborating with the entire community than keeping discoveries to

---

[13] I'm grossly oversimplifying the challenges that face software companies, particularly the high costs of sales and marketing. But relative to other types of businesses, software requires lower recurring costs to generate recurring revenue.
[14] https://www.engadget.com/2016/12/06/apple-will-publish-ai-research

themselves. Researchers who share their work get feedback and analysis from thousands of other experts. Plus sharing helps recruit talent.

In this game, no company has a magical model—any near-term breakthrough will be discovered by another researcher in due time.

Of course training AI models still isn't easy and requires specialized talent and engineering effort. But the barriers are falling rapidly, and most enterprise problems don't necessitate fundamental research breakthroughs.

Infrastructure

Deep learning requires specialized parallel-processing hardware. Unfortunately we're at the mercy (and pricing power) of NVIDIA, the market leader in AI hardware. Many companies (including Google and Intel) are working on competing solutions, but at the moment NVIDIA is the only game in town.

For all practical purposes a vendor can't build a competitive asset with infrastructure. The AI companies that want to sell you their products don't have a hardware resource you can't easily get for yourself.

Data

Data is the most valuable asset for building an AI solution. Building a sustainable competitive advantage in AI requires an ongoing investment in better input data. AI product companies have three primary options for training the models they want to sell to you:

- Use free or purchased data.
- Build a proprietary dataset.
- Use *your* (customer) data.

Let's talk through each option.

Free data

Governments and research organizations release datasets into the public domain. AI vendors can start training their models with these datasets. Common publicly available examples are the Enron Email Dataset (email),[15] Iris dataset (structured data),[16] and ImageNet dataset (images)[17]. Anyone can download these datasets in seconds, and I frequently use them to start training my own models.

Unfortunately these datasets suffer from many limitations. They bestow no competitive advantage, and models trained on them may not produce good results when applied to customer data.

Proprietary dataset

A better source of sustainable competitive advantage is a proprietary dataset. For example, a healthcare AI company may create a proprietary dataset by hiring radiologists to hand-label MRI images. In other cases a startup will knowingly violate the usage terms of a site like LinkedIn and scrape together a dataset for training models. (Yes, this happens.)

Building a high-quality proprietary dataset takes time and money, but it can be a great asset for AI vendors.

Your own data

Often the best source of training data to solve your problems is your data. Let's consider the automated claims processing example we used to explore the AI Canvas.

Imagine an automated claims processing solution for United Services Automobile Association (USAA), the financial services company which specializes in products for military members and their families. USAA's policies, customers, business process, and decisions are optimized for its customers. If we want to automate USAA's claims processing, the best possible training data is the millions of claims USAA has already processed.

---

[15] https://www.cs.cmu.edu/~enron
[16] https://archive.ics.uci.edu/ml/datasets/iris
[17] http://www.image-net.org

A vendor which has access to USAA's claims data can build the best automated AI claims solution for USAA. But is this arrangement in USAA's best interest? It depends. USAA may be able to get a better solution by partnering with a vendor which specializes in automated claims processing. But if USAA releases its data, its competitors could hire the same AI vendor to target products to military families. Moreover, partnering with a vendor would mean that USAA has missed a chance to invest in its internal AI capabilities.

AI buy vs build questions

Now that you understand the challenges facing AI solution vendors, here are some questions to consider as you decide whether to buy or build.

Is the solution a competitive asset?

You can apply the same logic to AI solutions that you apply to any product—don't outsource solutions which are key to your success, and don't build custom products when a cheaper alternative is available from vendors. You wouldn't build a customer relationship management solution, because dozens are already available on the market. You also wouldn't outsource a custom software application which is key to retaining a competitive advantage.

Do you care if a competitor has access to your data?

The heart of your decision is your ability to control who has access to your data. Keep in mind that a solution which has an AI model trained on your data could be purchased by a competitor.

Of course you can prevent losing some of this control by specifying the vendor's data usage rights. In practice these terms are difficult to enforce and easy to work around. For example, a data scientist can glean insights from your data which can be used to generate a different dataset with similar predictive assets. This new dataset could then be used to train the vendor's models—which it can sell to your competitors. Often this isn't done maliciously; product teams are simply trying to build the best possible solution for the market. Does this matter to your company?

Does the vendor offer a unique, valuable dataset?

In some cases a vendor may have better data than you. That data would be especially valuable if it requires human labor to label or segment. In this case the vendor's asset might be far less expensive than your cost of developing a comparable dataset.

If you're skeptical, ask the vendor to set up a test that compares the vendor's data to yours. Just ensure that they can't retrain their algorithms with your data before they run the test.

Does the vendor's software solution solve additional problems?

If AI is just a part of a vendor's solution, you can evaluate it like any other software application. For example the solution may contain useful workflows or interfaces.

Case study: Expensify

Expensify is a service that automatically processes employee expense reports.[18] Employees upload receipts, and Expensify uses NLP and computer vision techniques to automatically process the receipts on behalf of its clients.

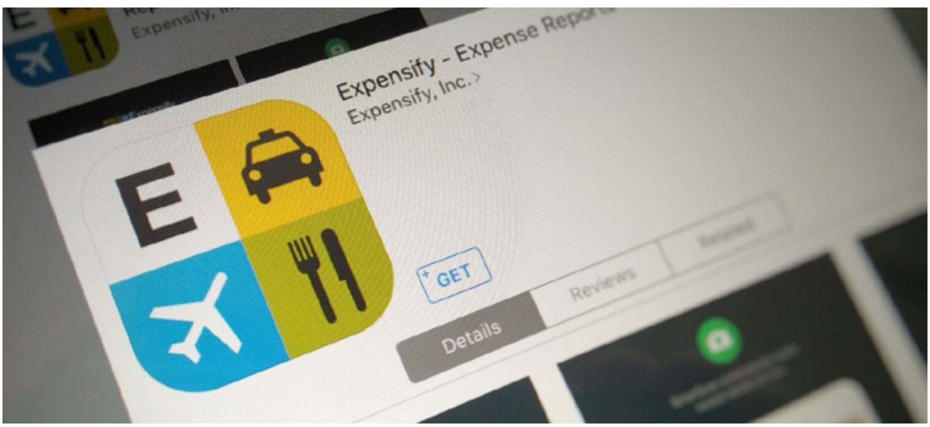


Figure 10

[18] https://www.expensify.com
Figure 10: https://venturebeat.com/2016/02/03/expensify-eyes-europe-for-growth-as-the-fintech-startup-launches-its-first-hub-outside-the-u-s/

Everyone wins in this arrangement: Customers get the solution they want, and Expensify uses the receipt data to train its algorithms. Better algorithms help employers and employees save the labor required to process receipts by hand.

Expensify is a great example of a solution you would buy instead of build.

BUILDING CONSENSUS

Fortunately you work at a company which values risk-taking. Your colleagues readily support your ideas and back you up when initiatives don't work. Your team fully embraces your ideas and gives you 100% support.

OK, enough of the fantasy. No large company is a meritocracy, and your CEO is happy with any risky investment–as long as it works.

Unfortunately AI solutions don't always work. Data science is a "science" because it requires experimentation. Models may not produce good results. If a solution produces only incremental efficiency improvements, projected cost savings might not be realized. Moreover, AI solutions often require organization-wide changes, and the AI solution team isn't always in a position to influence those changes.

Where does AI belong? With the product development department? Marketing? IT or data departments? Currently it belongs in the same place where the web site pioneering efforts of 1996 belonged: with whomever decides to pioneer it.

If you are to be the pioneer in your organization, you need to build a group of like-minded professionals to help you realize your organization's AI future. This will be your AI governance board.

The board should meet periodically to get consensus on major decisions associated with your AI initiatives:

- Which AI opportunities to pursue
- When to buy or build a solution
- When to continue, redirect, or shut down any current AI initiative

Fill your board with colleagues who will help provide top cover and will be able to make the organizational changes necessary to realize AI's potential in your business processes.

PART 3 QUIZ

1. One of your team members proposes a new AI initiative. With the AI Canvas as your guide, what questions should you ask to decide if it is an opportunity worth exploring?

2. A vendor stops by your office to pitch an AI product. What questions should you ask?

Find quiz answers in Appendix 2.

“Talent wins games.
Teamwork wins
championships.”

– **Michael Jordan**

Part 4

LAUNCHING YOUR FIRST AI PRODUCT IN 90 DAYS

So here you are–time for action! Clients don't hire Prolego to make sleek documents and pretty slides. Ultimately our success is measured by our impact. Your AI initiatives will be judged by the same ruthless standards.

In Part 4 I'll share the major steps to deploy your first AI product in 90 days. But first I'll share my point of view so you can understand my advice in context.

KNOWING YOUR REALM

I'm a startup guy. For most of my career I have been a founder, investor, or early employee in high-risk product companies. Most of these endeavors haven't worked out–that's just a reality of high stakes. It's a reality that I willingly accept because a few successes (like Palantir) make up for the rest.

As the Lean Startup methodology shows us, the best way to increase your long-term chances of success is by habitually confronting your products' highest-risk assumptions as early as possible.

AI projects fail for a lot of reasons. Here are the most common:

> The project **isn't worth pursuing.** The team incorrectly assumes an AI solution will create value. It ultimately doesn't.
> The team **doesn't have enough quality training data** to build the AI models.
> The **results are not good enough.**
> The team **can't get the product into production.**

How can you avoid failure? Pick a project worth pursuing. Know your data. Build a great product team. Get early results. Get the model into production as soon as you can. Evaluate and decide whether to continue.

How can you guarantee failure? Pick a vaguely defined project. Promise success. Tell your data scientists to work on it without clearly defining success criteria. Let them toil away without guidance or feedback. Keep promising success until . . . the CFO ultimately cancels your project.

Let's explore the steps you need to take to minimize your chance of failure and get your first AI product rolling. These steps are roughly consecutive, although you can do many of them simultaneously.

SETTING STRATEGIC ORGANIZATIONAL GOALS

Ultimately you need to make business impact: increase revenue, reduce costs, etc. Apart from those ends, you should also set organizational goals that will move you closer to becoming an AI-driven organization.

For example, communicate to your leadership that you aim to reach the following strategic milestones:

1. Set up your AI infrastructure.
2. Establish your AI governance board.
3. Review policies and make changes.
4. Create a well-organized set of training data.
5. Build a high-performing AI team.

If your first AI project utterly fails in production—perhaps for reasons outside of your control—you can point to the strategic value of hitting these organizational milestones. Your milestones prepare your organization to benefit from AI solutions and are therefore worthy goals even if they don't yield results in the near term. These objectives can help see you through the often tedious and time-consuming work of getting your project off the ground.

PICKING A PROTOTYPE

In a perfect world your company would easily see the value in AI and begin preparing for this fundamental shift. In reality you'll need to persuasively communicate the vision of what your company can do with AI before you can get any resources or support. One of the most effective ways to do that is to build a prototype.

Generating ideas

Start by generating a broad list of potential AI use cases by using the four product patterns as a guideline. Interview business leaders, analysts, or developers and ask for ideas. Use the resources discussed in Part 2 for inspiration.

I use a simple spreadsheet (Google Sheets or Microsoft Excel) to organize ideas. For example:

AI idea	Concept	Training data	Impact
Automated claims processing	Automatically accept or deny customer claims. Start with obvious claims and gradually increase automation.	Use existing claims data and results of claims processing. Potentially use other data sources from third parties. Find out how many claim forms are handwritten.	Medium

Don't do a lot of targeted research at this stage—just get all of the general ideas on the table. I usually try to generate about 20 potential use cases in a week or two.

Evaluating the promising opportunities

After completing the table, I select opportunities according to following criteria:

- Do we have readily accessible initial training data?
- Does leadership believe the project could have big impact?
- Do we have cooperation from key technical or operational people?

You might also have organization-specific criteria to consider. Usually a handful of the potential opportunities meet the core criteria.

For each of the best opportunities, complete an AI Canvas. Review policies, data dictionaries, and existing operational workflows so you can identify the key challenges. Have your data scientists review arXiv for relevant papers.

I usually spend a few weeks on this process, depending on the availability of key resources to help me answer strategic, data, and policy questions. If I can't get an answer in a few weeks, I flag it as an open issue on the canvas, and I move on.

Rank-ordering the canvases

Start your first AI governance board meeting with a bang. Open the discussion by presenting your canvases. Have each board member independently rank the canvases and pick your first AI project.

Your governance board may be reluctant to choose because of incomplete information. No matter how thorough your planning, you will have a lot of open questions. Accept the ambiguity and press forward. Then identify the best AI Canvases and turn them over to your AI product team.

BUILDING YOUR AI PRODUCT TEAM

Before we discuss your founding AI team, let's be clear on what you're trying to do. Your founding AI team needs to accomplish the following tasks:

1. Work through and resolve the open questions on your AI Canvas.
2. Identify your necessary data assets.
3. Stand up your AI infrastructure.
4. Build a training dataset.
5. Identify candidate models.
6. Train the models and evaluate results.
7. Review results with the consumers of the model's output.
8. Provide stakeholder updates.
9. Keep improving models until results are good enough.
10. Build any data engineering software necessary to put models into production.
11. Put the models into production so they can start generating results on new data.
12. Build any interfaces necessary to allow consumers to use the system in production.
13. Improve your data processing infrastructure.
14. Continue improving results by adding new data sources and fine-tuning models.

You may not understand every step in this process, but you get the idea: you need a product team. Although your team doesn't have to comprise of lofty-credentialed researchers, team members do need to be capable of hard, tedious work which requires extensive communication with many parts of the company.

You need a team which will run through walls to get your first release into production. The two critical roles for this team are (1) AI product managers and (2) machine learning engineers. As you begin deploying solutions, you will also need (3) data engineers and perhaps other specialists. Let's talk about your AI team's members.

AI product managers

Take a closer look at the 14 tasks your AI product team needs to complete. How many involve communication, coordination, and planning? Almost all of them. The product manager will drive this process. The ideal candidate knows how AI systems work, knows your data architecture, and is good at building consensus and making tradeoffs.

I evaluate potential candidates by asking myself, "Could this person design a Kaggle competition entry based on our AI project?" Visit Kaggle, review the active competitions, and think about the work required to organize a competition. The competition's requirements and assets show what an AI product manager would need to do to build a similar AI solution:

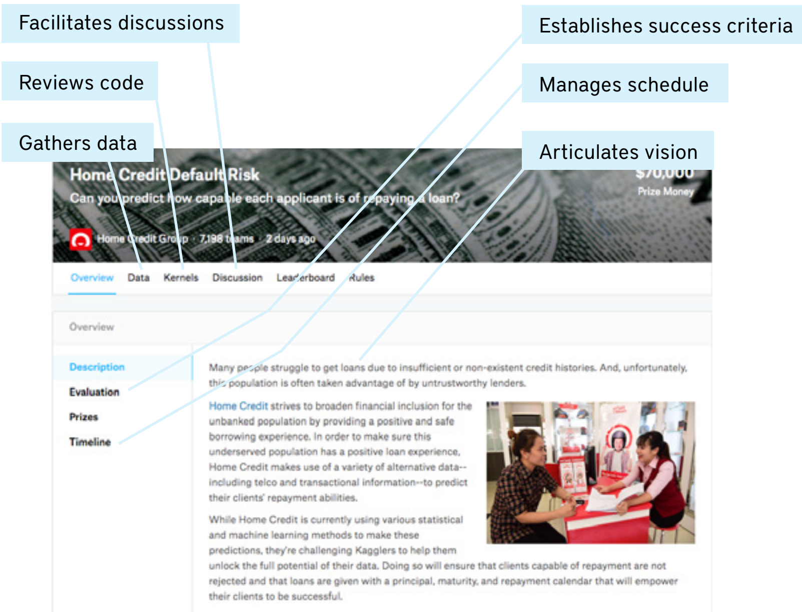


Figure 11

Look for AI product manager candidates with a math/computer science/statistics background, good project management skills, and tenacity. Be wary of product managers who come from a design background; they may not have the data and statistics background necessary to lead the team.

Machine learning engineers

Your AI product manager will remove obstacles and gather resources so the technical team can begin building your models and preparing to deploy solutions into production—an activity I call machine learning engineering. While the AI product manager is the most critical team member for the nontechnical side of your project, machine learning engineers are the most critical team members for the technical side.

Contrary to popular myth, there is no greater shortage of machine learning engineers than any other type of engineer. Not a lot of people can do machine learning engineering, but there is also not yet great demand for this skill.

Figure 11: https://www.kaggle.com/c/home-credit-default-risk

So why the obsession with AI talent shortage? In recent years the large tech companies have been rapaciously buying AI startups and gutting entire research labs in a race to gather talent.[19] These companies need the very best talent to win the AI race.

But you aren't doing cutting-edge AI research—you're just trying to apply current AI technology to your business challenges. So all you need is a team which can quickly learn the AI engineering best practices and put them to work on your data. This is what professional software engineers do.

I try to hire experienced software engineers who can do machine learning. A software engineer with five years' experience can learn how to use modern frameworks and to build basic solutions with about 6 to 12 months of study. Some return to graduate school, some take a temporary sabbatical from work, and others are self-taught in their evenings and weekends. It doesn't really matter how they got the skills as long as they can demonstrate proficiency.

In addition to the technical skills (which are beyond the scope of this book), here are some key attributes of the right machine learning engineers for your team:

- **Continuous learner.** AI is getting easier, but it is still technically challenging. Libraries are not well documented, and new techniques evolve constantly. AI is vastly harder than building an iPhone or web application. Look for the engineer who is already working on AI projects on the side.
- **Proven history of shipping products.** You need an action-oriented technical team, the kind of engineers who prioritize results. You want an engineer who will try to get a "good enough" model into production today rather than spend a few weeks searching for "the best" model.

---

[19] For example, see https://www.theverge.com/transportation/2015/5/19/8622831/uber-self-driving-cars-carnegie-mellon-poached

- **Very smart.** You don't need a team of Stanford PhD researchers, but that doesn't mean AI is easy. Despite the unsubstantiated vendor claims that AI development is "democratic," making AI do anything that's even trivial is still hard. You need some big brains.

Here are warning signs of an engineer who won't be a good fit for your team:

- Asks for training
- Has a fancy degree but is not currently working on anything
- Can't describe a solution they deployed
- Spends more time reading papers than writing code
- Uses AI buzzwords endlessly

When machine learning engineer candidates have any of these attributes, spend a little more time examining their qualifications to decide whether they are a good fit for your team.

Data engineers

As you move from prototyping to deployment, you will need software engineers who can manage data processing, pipelines, and jobs associated with running AI models in production. I call these data engineers.

Data engineers need to understand how AI works but don't have to be skilled at building models. They need to be good programmers who can build workflows so that the models all run correctly and that outputs are written to the interfaces where consumers can use them.

Hire experienced software engineers who have a history of building and supporting complex data processing systems.

They should be proficient at the following:

- Writing modular, maintainable software
- Building unit and integration tests
- Continuous integration and continuous deployment
- Maintaining and monitoring systems

Your company probably has people who manage ETL (extract, transfer, and load) systems to ensure data gets processed and stored in your data warehouse. These people usually have the skills to configure and operate workflow processes but may not have the software engineering skills necessary for your AI team.

You need roughly one data engineer for every machine learning engineer on your team. Unfortunately good data engineers who can build machine learning pipelines are extremely hard to find. You will be competing with other companies for talent from a very small pool.

Don't give up on your talent search. While your machine learning engineers can probably handle simple deployments on your first release, soon you will need dedicated data engineers to monitor and run everything in production.

Other specialists

If you're working with exceptionally large datasets, you may need an engineer who can optimize your hardware. They might need to configure clusters of GPUs, optimize cache, or allocate memory and storage, for example.

If you're pushing the cutting edge in a domain like computer vision, robotics, or NLP, you may need to bring on researchers who can keep you ahead of the competition.

Hire these sorts of specialists as you need them.

Data scientists

Do you need data scientists? It depends on your needs and their skills. The term "data scientist" has various meanings, so you'll need to understand exactly what they do to determine whether they could serve a useful role on your team. Here are some tips for evaluating data scientists for fit within your AI product team.

Data scientists who are business analysts

Some data scientists specialize in building reports and graphs in applications like Tableau. At a tech company this role is usually referred to as a business analyst. These specialists may have deep expertise in your business processes and data and may have good relationships with key stakeholders. They do little hands-on programming but help others make business decisions with data. These individuals are potential AI product manager candidates.

Data scientists who are statisticians

Some data scientists have a strong background in statistics and are good at creating hypotheses, testing assumptions, and drawing conclusions from historical data. They often focus on marketing and sales challenges such as identifying the right region or customer set for a targeted campaign. They often work with statistical packages like SAS, R, or (sometimes) Python.

These data scientists usually provide answers or rules which other engineering teams deploy into products. They are usually not experienced at building models which will be directly deployed in production. If they show enthusiasm and aptitude for the latter, they are potential machine learning engineers.

Data scientists who are engineers

Some data scientists are machine learning engineers. They have statistical skills and know how to build and deploy machine learning models. This is true in most tech companies. You want them on your AI product team.

SETTING UP YOUR INFRASTRUCTURE

You've identified your first AI pilot project and you're beginning to build your AI product team. Now you need to provide infrastructure so they can begin working. Your team will be able to handle this step, but you'll want to know the basic setup for modeling and testing your models. We call the setup an AI Sandbox.

Your machine learning engineers will use an AI Sandbox to build and evaluate models. AI tools and hardware are evolving rapidly, so it's impossible to say exactly what infrastructure you'll end up with, but here are some general guidelines.

The basics

An AI Sandbox requires a server, storage, and a graphical processing unit (GPU).

Here is a good starting point for your first AI Sandbox:

- A server with a fast, multi-core CPU
- At least 64 GB of RAM
- At least 500 GB of SSD storage
- At least 2 TB of HDD storage
- A GPU from NVIDIA with at least 8 GB of RAM

With these basics, even a top-of-the-line system will cost less than $5000.

Starting small

Don't start by buying $200K of hardware from NVIDIA unless you have a clear reason for it. For example, start with one GPU for each machine learning engineer and add additional GPUs when your team needs to train models faster.

Using what is popular

Your team will work faster if they can use the most popular tools and hardware. As long as they're using common tools, when they get stuck (this happens every day) they can search online for solutions.

Unfortunately the tools and hardware that your company's decision makers deem adequate might not be adequate for your AI Sandbox. You may have to go to battle for your team to get permission for the necessary infrastructure.

Here are the most popular tools for AI teams:

- Ubuntu Linux
- NVIDIA GPUs
- Python 3.6
- Keras, Tensorflow, or Pytorch
- Jupyter notebook

These tools will ease troubleshooting in your AI system.

Skipping the "magical" AI tools

You're going to have a parade of product companies telling you that their AI platform will magically solve your AI problems. If you haven't already, you'll learn their buzzwords by heart: Automated workflow! Pretrained models! Automated model tuning! Seamless collaboration!

Most of these vendors are selling solutions for small problems. You would be wise to wait until you actually *have* a real problem—one that *you* understand—before buying anything. In the meantime just use the open-source tools like the rest of us do.

For the moment your two biggest challenges are (1) knowing what problem to solve with AI and (2) having enough quality data to do it. Product companies can't help you with those challenges.

Giving your team control

Your team needs complete control (what we call "root access" in Linux) over the AI Sandbox. The machine learning engineers need the ability to update tools without waiting for "approval." They need to be able to download untested beta code and see if it works.

AI tools libraries change rapidly, and sometimes your team will need to update them *multiple times per day*. Waiting for approval will slow your team's progress to a crawl.

Cloud vs stand-alone servers

Product teams worldwide have spent the past decade migrating their systems to the cloud to reduce hardware maintenance costs and add resources on demand. You can get the same benefits by putting your AI Sandbox on the cloud, but doing so is not necessarily the best option.

Here are a few considerations:

- Stand-alone servers take more upfront work to configure. However, I find them to be more efficient than cloud-based servers because I don't have to deal with lag between my computer and the cloud.
- NVIDIA charges higher prices for cloud GPUs than it charges for stand-alone GPUs. I can buy faster GPUs from Best Buy for the same price that I can rent slower ones from Amazon Web Services (AWS).[20]
- If your company has restrictive policies governing data on the cloud, your machine learning team can probably build their own servers faster than you can change those policies.

Although stand-alone GPUs and cloud GPUs both have advantages and disadvantages, either option can work.

[20] https://alisha17.github.io/machine-learning/2017/12/15/benchmarks.html

BUILDING YOUR TRAINING DATA

Your project manager wants more time. Your call center manager wants more reps. Your marketing team wants a bigger budget. And your AI product team wants more training data. Nobody likes limitations, but reality is reality.

> ***We have enough training data.***
>
> – Said by no machine learning engineer, ever.

How much training data is "enough"? Although you may get better results if you have a very large dataset, you don't need Google-sized databases. Most practical AI products do not require massive amounts of data.

How much data you need is difficult to determine. If your team can't recommend a reasonable dataset size, a good rule of thumb is 50,000 labeled examples. With a dataset that size, your team can set aside 20%, or 10,000 samples, to validate their models. That is enough to detect 0.1% improvements from minor changes.

Once you set a size target for your dataset, your team will need the freedom to gather data and the discipline to put it to work fast. Here are some practical suggestions to set your team up for success.

Get support

Unfortunately you probably don't have a good inventory of your data assets. Your data has evolved over time, and you have data assets from companies you've acquired. This means your AI product team could waste *months* (seriously, this is common) trying to find the best sources of training data. When they do find good candidate datasets, the owners may be reluctant to release them to your team for many reasons. Before sending your team on a data scavenger hunt, get the support of your colleagues who own the data.

Impose time limits

Once your team starts finding data resources, give them a week or two to build their first training dataset. This time constraint will force them to make trade-offs and explore techniques like transfer learning (starting with pretrained models).

Refer to Part 2 for advice on building product-specific training data.

DRIVING FAST RESULTS

In time your team will start getting results in their AI Sandbox. They'll report that their prediction accuracy is promising, and they'll assure you that better results are on the way—they just need more time and data.

Avoid the temptation to delay deploying your models into production to give the team more testing time. As soon as your team gets decent results, feed the model a test dataset. If results are still good, deploy and iterate the model.

Proving the model with a test dataset

Your AI product team can very easily cheat or make a mistake in reporting results from their models. A team's model can get amazing results in the AI Sandbox but poor results in the production environment. The easiest way to prevent this problem is to test the team's models on a dataset they don't have. This data is called the test dataset.

A good candidate for your test dataset is your most recent operational data. If possible have a different team do the test. If the model still yields satisfactory results, you are in good shape.

Deploying and iterating

Your AI product team will initially work to build models based on historical data. But your ultimate goal is deploying the models into production so they make predictions on new data. Deploying your models has a technology component and an organizational component. Once you have a handle on these, deploy and iterate your product to make it efficient in the production environment.

Technology stack

In terms of technology, deploying AI models is similar to setting up any internal API service. Any mid-level server-side programmer should have no problem deploying your models. If you need more guidance, do a quick Google search and you'll find plenty of deployment instructions.[21]

Organizational impacts

The technological challenges for deployment will likely be much easier to solve than the organizational challenges. By the time you're ready to deploy, you should already have a basic plan and broad support in place. But your plan will still need a good guide.

Your AI product manager will have to work with the model consumers to decide where to send the model's results. Results are commonly sent to applications or database tables.

Early deployment

Compel your team to get your model into production. Your team will probably resist this request and will ask for more time to iterate the model in the AI Sandbox. But the only way to confront the major risks associated with your AI project is to get it live.

---

[21] For deployment tutorials, see https://www.analyticsvidhya.com/blog/2017/09/machine-learning-models-as-apis-using-flask and https://medium.com/@patrickmichelberger/how-to-deploy-a-serverless-machine-learning-microservice-with-aws-lambda-aws-api-gateway-and-d5b8cbead846

Expect the following:

- The models will initially not work well in production.
- Your AI product team will suddenly realize they have weeks of data engineering work to do to ensure models can reliably run in production.
- The model consumers will not change their behavior enough to actually use the results.

If you don't deploy early, you will waste time perfecting a product that's not equipped to handle real-world risks.

Improvement

You'll overcome the technical challenges of deployment the way you would overcome the challenges of any traditional software engineering project. As your AI production environment becomes more reliable, you will start identifying ways to improve the predictive power of your models.

To improve the functionality of your product, your AI product manager will have to drive the organizational changes and escalate challenges to you.

PART 4 QUIZ

1. What are the most common reasons AI projects fail?

2. Why should you set strategic organizational goals?

3. What are the two most critical roles in your first AI product team? What does each do?

4. What challenges might your team encounter when setting up the AI Sandbox?

5. What challenges might your team encounter while building training data?

Find quiz answers in Appendix 2.

"If you are not embarrassed by the first version of your product, you've launched too late."

– **Reid Hoffman**

Epilogue

I hope this book helps you begin your company's journey into AI. It isn't an easy journey to begin, so don't be discouraged.

Most AI educational resources are either too general or too technical for a corporate audience. Many of our clients have taken courses, read books and blogs, attended conferences, and listened to dozens of vendor pitches–and only found themselves more confused than ever. In this book I attempted to answer the most common questions these business leaders ask me.

Unfortunately I only scratched the surface on many complex topics. Because AI is evolving rapidly and we're getting better at solving your problems I am tempted to keep making the book "better." I want to expand the number of product patterns, add case studies, and provide more specific guidance for getting machine learning models into production. But at some point I had to stop typing and hand the copy over to Russ so he can do the harder work of getting the book produced and into your hands. So I will leave these additions for a future revision.

In the meantime, just reach out to Russ (russ@prolego.io) if you have any questions. We'll be happy to jump on a call together and talk about your situation.

Lots of people are talking about AI, but very few have successfully built and deployed AI products. We are among those few who have done it–for enterprise clients who have data, talent, and organizational challenges very similar to yours. A 30-minute call with us can save you months of headaches.

Kevin Dewalt
September 2018

ABOUT THE AUTHOR

Kevin Dewalt

Prolego founder & CEO

Kevin helps companies generate billion-dollar revenue opportunities with AI. At In-Q-Tel he helped make over $20M of venture capital investments and deployed Palantir's first intelligence product. At FINRA he ran the machine learning systems which enforce NASDAQ rules. As CMO of MadKudu he helped create Silicon Valley's premiere predictive sales platform. Kevin is also a recognized global innovation leader who launched the NSF I-CORPS class with Steve Blank at Stanford. He graduated #1 in his US Coast Guard Academy class with a degree in electrical engineering. His AI expertise began at Stanford University while completing original research under Dr. Bernard Widrow.

ABOUT THE PRODUCER

Russ Rands

Prolego founder & COO

Russ helps executives build strategies and products to take advantage of the AI revolution. Russ has spent the last 15 years leading corporate innovation teams at Fortune 1000 organizations. At Spigit, Russ led engagements in the development of enterprise-wide open innovation programs for such companies as Pfizer, Fidelity, ADP, Lowe's, and Southern Company. At CSC, Russ managed a crowdsourced innovation program, responsible for over $600M of value creation. At the Corporate Executive Board, Russ worked with chief innovation officers to build large innovation programs. Russ graduated with an M.B.A. from the George Washington University School of Business & a B.A. in English from Brigham Young University.

"I don't work on not turning AI evil today for the same reason I don't worry about the problem of overpopulation on the planet Mars."

– **Andrew Ng**

Andrew Ng would worry if "it only took 5 months to go from landing one person on Mars to Mars being overpopulated. By far the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."

– **Eliezer Yudkowski**

Appendices

APPENDIX 1:
DEALING WITH COMMON OBJECTIONS

AI is a popular topic in the mainstream media, so a lot of people know the talking points of the objectors. Here are examples of common objections you may encounter and ways to deal with them.

The “black box” objection

Many people have the impression that AI models are a “black box” which mysteriously generates results. In the AI world this concern is called the interpretability problem. People raise this objection when they lack good frameworks for evaluating AI models.

Understand model interpretability

In traditional software applications we can explain exactly why every decision is made by following the logic. These types of models are deterministic—every action has a specific cause. Lawyers and regulators like deterministic systems because they make it easy to interpret what happened. Traditional software models are highly interpretable.

AI models, conversely, are probabilistic. We don't have all of the information necessary to assign the specific reasons why a decision was made, so we have to assess a probability. AI models are thus less interpretable than traditional software models.

Probabilistic *doesn't* mean indeterministic. In other words, statements like "we have no idea why AI makes decisions" are absolutely false. For example, rolling two dice is a probabilistic model. We can't explain exactly why a 7 was rolled on a particular throw. We do know, however, that 7 occurs more often than 12, and 6.2 never occurs.

AI is more like the real world

You can't evaluate a complex probabilistic model the same way you can evaluate a traditional, deterministic software application. Probabilistic models have to be evaluated the way we evaluate complex decisions made by organizations.

For example, consider airline security. Marilyn Hartman, also known as the "serial stowaway," has sneaked onto more than 20 commercial airline flights.[22] Often without a passport or ticket, she manages to get through all airport security and to board flights.

How is this possible? What decisions lead to TSA and gate agents allowing her to pass? Hartman refuses to reveal her methods, and cameras capture only parts of her escapades, so we have few clear answers to how she does it.

---

[22] https://en.wikipedia.org/wiki/Marilyn_Hartman

Like systems in the airline industry, AI systems are extremely complex. We don't know exactly why every event occurs, but we have tools for evaluating the system's decisions.

The black-box objection most frequently arises in highly regulated industries like banking. After you identify the sticking points, addressing them might entail educating legal advisors on AI or demonstrating the AI system's usefulness by training a simpler model on the same dataset.

Educate your audience about AI

As you now know, AI isn't a newfangled, untested tool. Researchers have been studying neural networks since the 1950s. Once your audience understands the magnitude of AI's potential, they'll be willing to work to get approval to use it.

Describe to your audience what you're creating, how you're building the training data, and what happens with the results. Have your team show the results of your test datasets. Listen, explain, and build trust.

Interpret with simpler models

A demonstration is often an effective way to show the potential of your AI solution. In many cases you may be able to train a traditional machine learning model (random-forest models are a common choice) on the same dataset. For instance, you can identify the inputs that are most relevant for making an output prediction. While the results probably won't be as good as they would if you were using your neural network, the simplified model can help regulators understand how the AI version will work.

Social and economic fallout

AI will change our society in ways we cannot even imagine. Understandably people are concerned about inevitable downsides such as job losses, dangerous AI applications, and algorithms which capture undesired human biases.

Nobody knows the ultimate human costs of AI, and the experts debate the issue. They even debate about debating. Andrew Ng argues that fearing AI is like worrying about overpopulation on Mars.[23] Eliezer Yudkowsky claims this type of complacency poses enormous risks for humanity's survival.[24]

Both Ng and Yudkowsky make good arguments and know far more about AI than I do—but all they have are opinions. Since the real experts can't agree, *I don't feel compelled to take a position* about the societal impact of AI. I simply acknowledge these concerns and agree.

Every significant technological advance in history has had a human cost, and even if we haven't always overcome the costs, we have learned to deal with them.

---

23 https://www.theregister.co.uk/2015/03/19/andrew_ng_baidu_ai
24 https://samharris.org/podcasts/116-ai-racing-toward-brink

APPENDIX 2: QUIZ ANSWERS

Part 1

1. How does machine learning differ from traditional software development?

In traditional software a developer writes programs which explicitly tell a computer how to perform a task. In machine learning a developer uses examples (called training data) to teach an algorithm how to perform a task.

In traditional software a developer starts by asking, "What do you want the computer to do?" In machine learning a developer starts by asking, "Where is my training data?"

2. What are the advantages of deep learning? What are the drawbacks?

Deep learning produces better results when trained with large amounts of data. It also delivers state-of-the-art results with unstructured data like documents and images.

Deep learning also requires more specialized skills and hardware, as well as more time. Because of this, although the barriers to deep learning are falling, traditional machine learning algorithms are still a better choice in many instances.

3. What are the two components of data that are used for training? How are they different? What are the other names for these terms?

Inputs and outputs. *Outputs* are the results you want your AI system to produce. Identifying the desired output is one of the first steps in developing an AI strategy. *Inputs* are the data the AI system uses to generate the outputs. Your challenge is identifying enough quality inputs for the AI models.

	Also called	Function	Strategic role	Challenge
Inputs	Features, independent variables, X	Used to generate outputs	Require investment	Generating enough quality inputs that are predictive of outputs
Outputs	Targets, dependent variables, labels, Y	Results you want the AI model to produce	Create value	Identifying outputs that create business value

Statisticians normally refer to inputs as independent variables and outputs as dependent variables. Machine learning engineers and data scientists refer to inputs as features and refer to outputs as targets or labels. Design documents frequently list inputs as X and list outputs as Y. You need to be comfortable using all of these terms.

Part 2

1. Reread the Intelligent Vacation Planner example in the introduction. How would you apply each of the four product patterns to this product?

There are many ways to apply the four product patterns. Here are a few possible ideas:

> Computer vision could filter images from social media to identify hobbies and activity preferences. In my case you're likely to see a lot of pictures of golf and restaurants and no images from the beach or ocean—good proxies for my interests.

> NLP would be great for reading text, reviews, email, social media discussions, etc., to make recommendations on locations and activities.

> The next-in-sequence product pattern could be used to predict future prices or availability.

> Collaborative filters could be used to make recommendations based on what like-minded travelers enjoyed.

2. Suppose your engineering team is looking for the most effective and fastest way to solve a problem. Would you suggest starting with Kaggle or arXiv?

Usually Kaggle is a better source for the most effective and fastest ways to solve the kind of problems encountered in the enterprise. Kaggle competitors usually achieve better results than researchers because thousands of people are competing to solve the same problem. While academics value the creativity and intellectual rigor found in research papers, Kaggle competitors care about results. This doesn't mean Kaggle competitors are "the best" available talent. I often discover high-ranked Kaggle competitors who don't understand the fundamental principles in their code. Some get good results by copying other entries and improving their results through trial and error.

The arXiv web site is most useful for teams who are pushing the cutting edge of technology in a particular domain.

Part 3

1. One of your team members proposes a new AI initiative. What questions should you ask to decide if it is an opportunity worth exploring?

The AI Canvas poses all of the major questions. Good starting topics to address are data sources, consumers, model development (specifically product patterns), and strategy.

2. A vendor stops by your office to pitch an AI product. What questions should you ask?

Spend most of your time asking about training data—what data they used, how they acquired it, etc. Ask how they will use your data in their solution. Ask how they know their solution will work on your data. If they mention proprietary models, ask them what they have published. If the answer is "We're using AI; I'll have to ask the tech team for details," end the meeting. They're just wasting your time.

Part 4

1. What are the most common reasons AI projects fail?
Projects fail when they don't create business value, they lack sufficient quality training data, their results are poor, or the project team can't get working models into production.

2. Why should you set strategic organizational goals?
Getting your first AI project off the ground takes a lot of tedious work, which sometimes doesn't look like progress at all. Your milestones prepare your organization to benefit from future AI solutions and are therefore worthy goals even if they don't yield results in the near term. Don't set yourself up for disappointment by focusing only on immediate impact.

3. What are the two most critical roles in your first AI product team? What does each do?
(1) An *AI product manager* helps gather training data, prepare the organization for the model outputs, and update legal policies. This person basically sets up all of the non-technical requirements to get the first model into production. (2) Your *machine learning engineer* cleans the data, builds models, evaluates results, and deploys the first solution.

4. What challenges might your team encounter when setting up the AI Sandbox?
They might need hardware and tools which your organization hasn't approved. You'll need to help them obtain the resources they need.

5. What challenges might your team encounter while building training data?
They are likely to have difficulty finding the data, evaluating the data, and getting permission to use it.

You're trying to decide whether this book is worth your time. You're skeptical because, well, most of what you read about AI is useless. Vendors are pitching you magical solutions you don't understand. The blogs, podcasts, online courses, and books are too technical. The last AI conference you attended presented nothing but fluffy concepts which didn't get you closer to your only real question: *What can we actually do with AI?*

Our corporate clients have the same frustrations, and in this book we answer the most common questions these business leaders ask us. Lots of people are talking about AI, but very few have successfully built and deployed AI products at large Fortune 500 companies. We are among those few who have done it—for enterprise clients who have data, talent, and organizational challenges very similar to yours.

But every company is different and no book can address all of your challenges. Just reach out to us if you have any questions. We'll be happy to jump on a call together and talk about your situation.

Kevin Dewalt and Russ Rands
Founders, Prolego

Contact us at russ@prolego.io
or at (703) 795-1528

$79.00
ISBN 978-0-692-19233-7
57900>
9 780692 192337