

Welcome to AI for everyone. AI is changing the way we work and live and this nontechnical course will teach you how to navigate the rise of AI. Whether you want to know what's behind the buzzwords or whether you want to perhaps use AI yourself either in a personal context or in a corporation or other organization, this course will teach you how. If you want to understand how AI is affecting society, and how you can navigate that, you also learn that from this course.

In this first week, we'll start by cutting through the hype and giving you a realistic view of what AI really is. Let's get started. You've probably seen news articles about how much value AI is creating. According to a study by McKinsey Global Institute, AI is estimated to create an additional 13 trillion US dollars of value annually by the year 2030.

Even though AI is already creating tremendous amounts of value into software industry, a lot of the value to be created in a future lies outside the software industry. In sectors such as retail, travel, transportation, automotive, materials, manufacturing and so on. I should have a hard time thinking of an industry that I don't think AI will have a huge impact on in the next several years. My friends and I used to challenge each other to name an industry where we don't think AI will have a huge impact. My best example was the hairdressing industry because we know how to use AI robotics to automate hairdressing. But, I once said this on stage and one of my friends who is a robotics professor was in the audience that day, and she actually stood up and she looked at me in the eye and she said, "You know Andrew, most people's hairstyles, I couldn't get a robot to cut their hair that way." But she looked at me and said, "Your hairstyle Andrew, that a robot can do."

There is a lot of excitement but also a lot of unnecessary hype about AI. One of the reasons for this is because AI is actually two separate ideas. Almost all the progress we are seeing in the AI today is artificial narrow intelligence. These are AIs that do one thing such as a smart speaker or a self-driving car or AI to do web search or AI applications in farming or in a factory. These types of AI are one trick ponies but when you find the appropriate trick, this can be incredibly valuable. Unfortunately, AI also refers to a second concept of AGI or artificial general intelligence. That is the goal to build AI. They can do anything a human can do or maybe even be superintelligence and do even more things than any human can. I'm seeing tons of progress in ANI, artificial narrow intelligence and almost no progress to what AGI or artificial general intelligence. Both of these are worthy goals and unfortunately the rapid progress in ANI which is incredibly valuable, that has caused people to conclude that there's a lot of progress in AI, which is true. But that has caused people to falsely think that there might be a lot of progress in AGI as well which is leading to some irrational fears about evil clever robots coming over to take over humanity anytime now. I think AGI is an exciting goal for researchers to work on, but it'll take most for technological breakthroughs before we get there and it may be decades or hundreds of years or even thousands of years away. Given how far away AGI is, I think there is no need to unduly worry about it.

In this week, you will learn what ANI can do and how to apply them to your problems. Later in this, course you'll also see some case studies of how ANI, this one trick ponies can be used to build really valuable applications such as smart speakers and self-driving cars. In this week, you will learn why this AI. You may have heard of machine learning and the next video will teach you what is machine learning. You also learn what is data and what types of data are valuable but also what does the data are not valuable. You learn what it is that makes a company an AI company or an AI first company so that perhaps you can start thinking if there are ways to improve your company or other organizations ability to use AI and importantly, you also learned this week what machine learning can and cannot do. In our society, newspapers as well as research papers tend to talk only about the success stories of machine learning and AI and we hardly ever see any failure stories because they just aren't as interesting to report on. But for you to have a realistic view of what AI and what machine learning can or cannot do, I think is important that you see examples of both so that you can make more accurate judgements about what you may and maybe should not try to

use these technologies for. Finally, a lot of the recent rise of, machine learning has been driven through the rise of Deep Learning. Sometimes also called Neural Networks.

In the final two optional videos of this week, you can also see an intuitive explanation of deep learning so that you will better understand what they can do particularly for a set of narrow ANI tasks. So, that's what you learn this week and by the end of this week, you have a sense of AI technologies and what they can and cannot do.

In the second week, you'll learn how these AI technologies can be used to build valuable projects. You learn what it feels like to build an AI project as well as what you should do to make sure you select projects that are technically feasible as well as valuable to you or your business or other organization. After learning what it takes to build AI projects,

In the third week you'll learn how to build AI in your company. In particular, if you want to take a few steps toward making your company good at AI, you see the AI transformation playbook and learn how to build AI teams and also build complex AI products. Finally, AI is having a huge impact on society. In a fourth and final week, you'll learn about how AI systems can be biased and how to diminish or eliminate such biases. You also learn how AI is affecting developing economies and how AI is affecting jobs and be better able to navigate this rise of AI for yourself and for your organization. By the end of this four recourse, you'll be more knowledgeable and better qualified than even the CEOs of most large companies in terms of your understanding of AI technology as well as your ability to help yourself or your company or other organization navigate the rise of AI as I hope that after this course, you'll be in a position to provide leadership to others as well as they navigate these issues. Now, one of the major technologies driving the recent rise of AI is Machine Learning. But what is Machine Learning? Let's take a look in the next video.

The rise of AI has been largely driven by one tool in AI called machine learning. In this video, you'll learn what is machine learning, so that by the end, you hope we will start thinking how machine learning might be applied to your company or to your industry.

The most commonly used type of machine learning is a type of AI that learns A to B, or input to output mappings. This is called supervised learning.

Let's see some examples. If the input A is an email and the output B one is email spam or not, zero one. Then this is the core piece of AI used to build a spam filter. Or if the input is an audio clip, and the AI's job is to output the text transcript, then this is speech recognition. More examples, if you want to input English and have it output a different language, Chinese, Spanish, something else, then this is machine translation. Or the most lucrative form of supervised learning, of this type of machine learning maybe be online advertising, where all the large online ad platforms have a piece of AI that inputs some information about an ad, and some information about you, and tries to figure out, will you click on this ad or not? By showing you the ads you're most likely to click on, this turns out to be very lucrative. Maybe not the most inspiring application, but certainly having a huge economic impact today. Or if you want to build a self-driving car, one of the key pieces of AI is in the AI that takes as input an image, and some information from their radar, or from other sensors, and output the position of other cars, so your self-driving car can avoid the other cars. Or in manufacturing. I've actually done a lot of work in manufacturing where you take as input a picture of something you've just manufactured, such as a picture of a cell phone coming off the assembly line. This is a picture of a phone, not a picture taken by a phone, and you want to output, is there a scratch, or is there a dent, or some other defects on this thing you've just manufactured? And this is visual inspection which is helping manufacturers to reduce or prevent defects in the things that they're making. This set of AI called supervised learning, just learns input to output, or A to B mappings. On one hand, input to output, A to B it seems quite limiting. But when you find a right application scenario, this can be incredibly valuable. Now, the idea of supervised learning has been around for many decades. But it's really taken off in the last few years. Why is this? Well, my friends asked me, "Hey Andrew, why is supervised learning taking off now?" There's a picture I draw for them. I want to show you this picture now, and you may be able to draw this picture for others that ask you the same question as well. Let's say on the horizontal axis you plot the amount of data you have for a task. So, for speech recognition, this might be the amount of audio data and transcripts you have. In lot of industries, the amount of data you have access to has really grown over the last couple of decades. Thanks to the rise of the Internet, the rise of computers. A lot of what used to be say pieces of paper, are now instead recorded on a digital computer. So, we've just been getting more and more data. Now, let's say on the vertical axis you plot the performance of an AI system. It turns out that if you use a traditional AI system, then the performance would grow like this, that as you feed in more data is performance gets a bit better. But beyond a certain point it did not get that much better. So it's as if your speech recognition system did not get that much more accurate, or your online advertising system didn't get that much more accurate that's showing the most relevant ads, even as you show the more data. AI has really taken off recently due to the rise of neural networks and deep learning. I'll define these terms more precise in later video, so don't worry too much about what it means for now. But with modern AI, with neural networks and deep learning, what we saw was that, if you train a small neural network, then the performance looks like this, where as you feed them more data, performance keeps getting better for much longer. If you train a even slightly larger neural network, say medium-sized neural net, then the performance may look like that. If you train a very large neural network, then the performance just keeps on getting better and better. For applications like speech recognition, online advertising, building self-driving car, where having a high-performance, highly accurate, say speech recognition system is important, enable these AI systems get much better, and make speech recognition products much more acceptable to users, much more valuable to companies and to users. Now, a few couple of implications of this figure. If you want the best possible levels of performance, your performance to be up here, to hit this level of performance, then you need two things:

One is, it really helps to have a lot of data. So that's why sometimes you hear about big data. Having more data almost always helps.

The second thing is, you want to be able to train a very large neural network.

So, the rise of fast computers, including Moore's law, but also the rise of specialized processors such as graphics processing units or GPUs, which you'll hear more about in a later video, has enabled many companies, not just a giant tech companies, but many many other companies to be able to train large neural nets on a large enough amount of data in order to get very good performance and drive business value.

The most important idea in AI has been machine learning, has basically supervised learning, which means A to B, or input to output mappings. What enables it to work really well is data.

In the next video, let's take a look at what is the data and what data you might already have? And how to think about feeding this into AI systems. Let's go on to the next video.

You may have heard that data is really important for building AI systems. But, what is data really?

Let's look at an example of a table of data which we also call a dataset. If you're trying to figure out how to price houses that you're trying to buy or sell, you might collect a dataset like this, and this can be just a spreadsheet, like a MS excel spreadsheet of data where one column is the size of the house, say in square feet or square meters, and the second column is the price of the house. So, if you're trying to build a AI system or Machine Learning system to help you set prices for houses or figure out if a house is priced appropriately, you might decide that the size of the house is A and the price of the house is B, and have an AI system learn this input to output or A to B mapping. Now, rather than just pricing a house based on their size, you might say, "Well, let's also collect data on the number of bedrooms of this house." In that case, A can be both of these first two columns, and B can be just the price of the house. So, given that table of data, given the dataset, it's actually up to you, up to your business use case to decide what is A and what is B. Data is often unique to your business, and this is an example of a dataset that a rural state agency might have that they tried to help price houses. It's up to you to decide what is A and what is B, and how to choose these definitions of A and B to make it valuable for your business.

As another example, if you have a certain budget and you want to decide what is the size of house you can afford, then you might decide that the input A is how much does someone spend and B is just the size of the house in square feet, and that would be a totally different choice of A and B that tells you, given a certain budget, what's the size of the house you should be maybe looking at.

Here's another example of a dataset. Let's say that you want to build a AI system to recognize cats in pictures. I'm not sure why you might want to do that, but maybe the fun mobile app, and you want to tag all the pictures of cats. So, you might collect a dataset where the input A is a set of different images and the output B are labels that says, "First picture is a cat, that's not a cat. That's a cat, that's not a cat" and have an AI input a picture A and output B is it the cats or not so you can tag all the cat pictures on your photo feed or your mobile app. In Machine Learning tradition, there's actually a lot of cats in Machine Learning. I think some of this started when I was leaving the Google Brain team and we published the results with somewhat infamous Google cat, where an AI system learn to detect cats from watching YouTube videos. But since then, there's been a tradition of using cats as a running example when talking about Machine Learning with apologies to all the dog lovers out there. I love dogs too. So, data is important. But how do you get data? How do you acquire data? Well, one way to get data is manual labeling. For example, you might collect a set of pictures like these over here, and then you might either yourself or have someone else go through these pictures and label each of them. So, the first one is a cat, second one is not a cat, third one is a cat, fourth one is not a cat. By manually labeling each of these images, you now have a dataset for building a cat detector. To do that, you actually need more than four pictures. You might need hundreds of thousands of pictures but manual labeling is a tried and true way of getting a dataset where you have both A and B. Another way to get a dataset is from observing user behaviors or other types of behaviors.

So, for example, let's say you run a website that sells things online. So, an e-commerce or an electronic commerce website where you offer things to users at different prices, and you can just observe if they buy your product or not. So, just through the act of either buying or not buying your product, you may be able to collected a data set like this, where you can store the user ID, the time the user visited your website, the price you offer the product to the users as well as whether or not they purchased it. So, just by using your website, users can generate this data from you. This was an example of observing user behaviors.

We can also observe behaviors of other things such as machines. If you run a large machine in a factory and you want to predict if a machine is about to fail or have a fault, then just by observing the behavior of a machine, you can then record a dataset like this. There's a machine ID, there's a temperature of the machine, there's a pressure within the machine, and then did the machine fail or not. If your application is

prevent the maintenance, say you want to figure out if a machine is about to fail, then you could for example, choose this as the input A and choose that as the output B to try to figure out if a machine is about to fail in which case you might do preventative maintenance on the machine.

The third and very common way of acquiring data is to download it from a website or to get it from a partner. Thanks to the open internet, there's just so many, there's as that you can download for free, ranging from computer vision or image datasets, to self-driving car datasets, to speech recognition datasets, to medical imaging data sets to many many more. So, if your application needs a type of data, you just download off the web keeping in mind licensing and copyright, then that could be a great way to get started on the application. Finally, if you're working with a partner, say you're working with a factory, then they may already have collected a big dataset, machines, and temperatures, and pressure into the machines fail not that they could give to you. Data is important, but there's also little bit over-hyped and sometimes misused.

Let me just describe to you two of the most common misuses or the bad ways of thinking about data. When I speak of seals of large companies, a few of them have even said to me, "Hey Andrew, give me three years to build up my IT team, we're collecting so much data. Then after three years, I'll have this perfect dataset, and then we'll do AI then." It turns out that's a really bad strategy. Instead, what I recommend to every company, is once you've started collecting some data, go ahead and start showing it or feeding it to an AI team. Because often, the AI team can give feedback to your IT team on what types of data to collect and what types of IT infrastructure to keep on building. For example, maybe an AI team can look at your factory data and say, "Hey. You know what? If you can collect data from this big manufacturing machine, not just once every ten minutes, but instead once every one minute, then we could do a much better job building a preventative maintenance systems for you." So, there's often this interplay of this back and forth between IT and AI teams, and my advise is usually try to get feedback from AI earlier, because it can help you guide the development of your IT infrastructure.

Second, misuse of data. Unfortunately, I've seen some CEOs read about the importance of the trend in use, and then say, "Hey, I have so much data. Surely, an AI team can make it valuable." Unfortunately, this doesn't always work out. More data is usually better than less data, but I wouldn't take it for granted that just because you have many terabytes or gigabytes of data, that an AI team can actually make that valuable. So, my advice is don't throw data in a AI team and assume it will be valuable. In fact, in one extreme case, I saw one company go and acquire a whole string of other companies in medicine, on the thesis, on the hypothesis that their data would be very valuable. Now, a couple years later, as far as I know the engineers have not yet figured out how to take all this data and actually create value out of it. So, sometimes it works and sometimes it doesn't. But, they will not over-invest in just acquiring data for the sake of data until unless you're also getting an AI team to take a look at it. Because, they can help guide you to think through what is the data that is actually the most valuable.

Finally, data is messy. You may have heard the phrase garbage in garbage out, and if you have bad data, then the AI will learn inaccurate things. Here are some examples of data problems. Let's say you have this data sets of size of houses, number of bedrooms, and the price. You can have incorrect labels or just incorrect data. For example, this house is probably not going to sell for \$0.1 just for one dollar. Or, data can also have missing values such as we have here a whole bunch of unknown values. So, your AI team will need to figure out how to clean up the data or how to deal with these incorrect labels and all missing values. There are also multiple types of data. For example, sometimes you hear about images, audio, and text. These are types of data that humans find it very easy to interpret. There's a term for this. This is called unstructured data, and there's a certain types of AI techniques that could work with images to recognize cats or audios to recognize speech or texts, or understand that email is spam. Then, there are also datasets like the one on the right. This is an example of structured data. That basically means data that lives in a giant spreadsheet, and the techniques for dealing with unstructured data are little bit different than the

techniques for dealing with structured data. But AI techniques can work very well for both of these types of data, unstructured data and structured data.

In this video, you learned what is data and you also saw how not to misuse data, for example by over-investing in an IT infrastructure in the hope that it will be useful for AI in the future, but we're actually checking that they're really will be useful for the AI applications you want to build. Finally, you saw data is messy. But a good AI team would be the help you deal with all of these problems.

Now, AI has a complicated terminology when people throw around terms like AI, Machine Learning, Data Science. What I want to do in the next video is share with you what these terms actually mean, so that you'd be able to confidently and accurately talk about these concepts with others. Let's go on to the next video.

You might have heard terminology from AI, such as machine learning or data science or neural networks or deep learning. What do these terms mean? In this video, you'll see what is this terminology of the most important concepts of AI, so that you will speak with others about it and start thinking how these things could apply in your business. Let's get started.

Let's say you have a housing dataset like this with the size of the house, number of bedrooms, number of bathrooms, whether the house is newly renovated as was the price. If you want to build a mobile app to help people price houses, so this would be the input A, and this would be the output B. Then, this would be a machine-learning system, and particular would be one of those machine learning systems that learns inputs to outputs, or A to B mappings. So, machine learning often results in a running AI system. So, it's a piece of software that anytime of day, anytime of night you can automatically input A these properties of house and output B. So, if you have an AI system running, serving dozens or hundreds of thousands of millions of users, that's usually a machine-learning system.

In contrast, here's something else you might want to do, which is to have a team analyze your dataset in order to gain insights. So, a team might come up with a conclusion like, "Hey, did you know if you have two houses of a similar size, they've a similar square footage, if the house has three bedrooms, then they cost a lot more than the house of two bedrooms, even if the square for this is the same." Or, "Did you know that newly renovated homes have a 15% premium, and this can help you make decisions such as, given a similar square footage, do you want to build a two bedroom or three bedroom size in order to maximize value?" Or, "Is it worth an investment to renovate a home in the hope that the renovation increases the price you can sell a house for?" So, these would be examples of data science projects, where the output of a data science project is a set of insights that can help you make business decisions, such as what type of house to build or whether to invest in renovation.

The boundaries between these two terms, machine learning and data science are actually little bit buzzy, and these terms are not used consistently even in industry today. But what I'm giving here is maybe the most commonly used definitions of these terms, but you will not find universal adherence to these definitions. To formalize these two notions a bit more,

Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed. This is a definition by Arthur Samuel many decades ago. Arthur Samuel was one of the pioneers of machine learning, who was famous for building a checkers playing program. They could play checkers, even better than he himself, the inventor could play the game. So, a machine learning project will often results in a piece of software that runs, that outputs B given A.

In contrast, data science is the size of extracting knowledge and insights from data. So, the output of a data science project is often a slide deck, the PowerPoint presentation that summarizes conclusions for executives to take business actions or that summarizes conclusions for a product team to decide how to improve a website.

Let me give an example of machine learning versus data science in the online advertising industry. Today, to launch our platforms, all have a piece of AI that quickly tells them what's the ad you are most likely to click on. So, that's a machine learning system. This turns out to be incredibly lucrative AI system to inputs enrich about you and about the ad and outputs whether you click on this or not. These systems are running 24-7. These are machine learning systems that drive our gravity for these companies, such as a piece of software that runs.

In contrast, I have also done data science projects in the online advertising industry. If analyzing data tells you, for example, that the travel industry is not buying a lot of ads, but if you send more salespeople to sell ads to travel companies, you could convince them to use more advertising, then that would be an example of a data science project and the data science conclusion the results and the executives deciding to ask a

sales team to spend more time reaching out to the travel industry. So, even in one company, you may have different machine learning and data science projects, both of which can be incredibly valuable.

You have also heard of deep learning. So, what is deep learning? Let's say you want to predict housing prices, you want to price houses. So, you will have an input that tells you the size of the house, number of bedrooms, number of bathrooms and whether it's newly renovated. One of the most effective ways to price houses, given this input A would be to feed it to this thing here in order to have it output the price. This big thing in the middle is called a neural network, and sometimes we also called an artificial neural network. That's to distinguish it from the neural network that is in your brain. So, the human brain is made up of neurons. So, when we say artificial neural network, that's just to emphasize that this is not the biological brain, but this is a piece of software. What a neural network does, or an artificial neural network does is takes this input A, which is all of these four things, and then output B, which is the estimated price of the house.

Now, in a later optional video this week, I'll show you more, what this artificial neural network really is. But all of human cognition is made up of neurons in your brain passing electrical impulses, passing little messages each other. When we draw a picture of an artificial neural network, there's a very loose analogy to the brain. These little circles are called artificial neurons, or just neurons for short. That also passes neurons to each other. This big artificial neural network is just a big mathematical equation that tells it given the inputs A, how do you compute the price B. In case it seems like there a lot of details here, don't worry about it. We'll talk more about these details later. But the key takeaways are that a neural network is a very effective technique for learning A to B or input-output mappings.

Today, the terms neural network and deep learning are used almost interchangeably, they mean essentially the same thing. Many decades ago, this type of software was called a neural network. But in recent years, we found that deep learning was just a much better sounding brand, and so that for better or worse is a term that's been taken off recently. So, what do neural networks or artificial neural networks have to do with the brain? It turns out almost nothing. Neural networks were originally inspired by the brain, but the details of how they work are almost completely unrelated to how biological brains work. So, I choose very courses today about making any analogies between artificial neural networks and the biological brain, even though there was some loose inspiration there. So, AI has many different tools.

In this video, you learned about what are machine learning and data science, and also what is deep learning, and what's a neural network. You might also hear in the media other buzzwords like unsupervised learning, reinforcement learning, graphical models, planning, knowledge graph, and so on. You don't need to know what all of these other terms mean, but these are just other tools to getting AI systems to make computers act intelligently. I'll try to give you a sense of what some of these terms mean in later videos as well. But the most important tools that I hope you know about are machine learning and data science as well as deep learning and neural networks, which are a very powerful way to do machine learning, and sometimes data science.

If we were to draw a Venn diagram showing how all these concepts put together, this is what it may look like. AI is this huge set of tools for making computers behave intelligently. Off AI, the biggest subset is prairie tools from machine learning, but AI does have other tools than machine learning, such as some of these buzzwords, are listed at the bottom. The part of machine learning that's most important these days is neural networks or deep learning, which is a very powerful set of tools for carrying out supervised learning or A to B mappings as well as some other things. But there are also other machine learning tools that are not just deep learning tools. So, how does data science fit into this picture? There is inconsistency in how the terminology is used. Some people will tell you data science is a subset of AI. Some people will tell you AI is a subset of data science. So, it depends on who you ask. But I would say that data science is maybe a cross-cutting subset of all of these tools that uses many tools from AI machine learning and deep learning,

but has some other separate tools as well that solves a very set of important problems in driving business insights.

In this video, you saw what is machine learning, what is data science, and what is deep learning and neural networks. I hope this gives you a sense of the most common and important terminology using AI, and you can start thinking about how these things might apply to your company. Now, what does it mean for a company to be good at AI? Let's talk about that in the next video.

What makes a company good at AI? Perhaps even more importantly, what will it take for your company to become great at using AI? I had previously led the Google brain team, and Baidu's AI group, which I respectively helped Google and Baidu become great AI companies. So, what can you do for your company? This is the lesson I had learned from the rise of the Internet that I think will be relevant to how all of us navigate the rise of AI. Let's take a look. A lesson we learned from the rise of the Internet was that, if you take your favorite shopping mall. So, my wife and I sometimes shop at Stanford shopping center and you build a website for the Shopping mall. Maybe sell things on the website, that by itself does not turn the shopping mall into an internet company. In fact, a few years ago I was speaking with the CEO of a large retail company who said to me, "Hey Andrew, I have a website, I sell things in the website." Amazon has a website, Amazon sells things on website is the same thing. But of course it wasn't, in the shopping mall with a website isn't the same thing as a first-class internet company. So, what is it that defines an internet company if it isn't just whether or not you sell things on the website? I think an Internet company is a company that does the thing that internet let you do really well. For example, we engage and pervasive AB testing. Meaning we routinely threw up two different versions of website and see which one works better because we can. So, we learn much faster. Whereas in a traditional shopping mall, very difficult to have two shopping malls in two parallel universes and you can only maybe change things around every quarter or every six months. Internet company is since a very short iteration times. You can ship a new product every week or maybe even every day because you can whereas a shopping mall can be redesigned and we are protected only every several months. Internet companies also tend to push decision-making down from the CEO to the engineers and to other specialized roles such that the product managers. This is in contrast to a traditional shopping mall. We can maybe have the CEO just decide all the key decisions and then just everyone does what the CEO says. It turns out that traditional model doesn't work in the internet era because only the engineers and other specialized roles like product managers know enough about the technology and the product and the users to make great decisions. So, these are some of the things that internet companies do in order to make sure they do the things that the internet doesn't do really well. This is a lesson we learned from the internet era. How about the AI era? I think that today, you can take any company and have it use a few neural networks or few deep learning algorithms. That by itself does not turn the accompany into an AI company.

Instead, what makes a great AI company, sometimes an AI first company is, are you doing the things that AI lets you do really well? For example, AI companies are very good at strategic data acquisition. This is why any of the large consumer tech companies may have three products that do not monetize and it allows them to acquire data that they can monetize elsewhere. Serve less strategy teams where we would deliberately launch products that do not make any money just for the sake of data acquisition. Thinking through how to get data is a key part of the great AI companies. AI companies sends a unified data warehouses. If you have 50 different databases or 50 different data warehouses under the control of 50 different Vice-Presidents, then there will be impossible for an engineer to get the data into one place so that they can connect the dots and spot the patterns. So, many great AI companies have preemptively invested in bringing the data together into single data warehouse to increase the odds that the teams can connect the dots. Subject of course to privacy guarantees and also to data regulations such as GDPR in Europe.

AI companies are very good at spotting automation opportunities. We're very good at saying, Oh! Let's insert the supervised learning algorithm and have an ATP mapping here so that we don't have to have people do these tasks instead we can automate it. AI companies also have many new roles such as the MLE or Machine Learning Engineer and new ways of dividing up tasks among different members of a team. So, for a company to become good at AI means, architecting for company to do the things that AI makes it possible to do really well.

Now, for a company that become good at AI does require a process. In fact, 10 years ago, Google and Baidu as well as companies like Facebook and Microsoft that I was not a part of, we're not great AI companies the way they are today. So, how can a company become good at AI? It turns out that becoming good at AI is not a mysterious magical process. Instead there is a systematic process through which many companies, almost any big company can become good at AI. This is the five-step AI transformation playbook that I recommend to companies that want to become effective at using AI. I'll give a brief overview of the playbook here and they're going to detail in a later week.

Step one is to execute pilot projects to gain momentum. So, just to a few small projects to get a better sense of what AI can or cannot do and get a better sense of what doing an AI project feels like. This you could do in house or you can also do with an outsource team.

But eventually, you then need to do step two which is the building in house AI team and provide broad AI training, not just to the engineers but also to the managers, division leaders and executives and how they think about AI. After doing this or as you're doing this, you have a better sense of what AI is and then is important for many companies to develop an AI strategy. Finally, to align internal and external communications so that all your stakeholders from employees, customers and investors are aligns with how your company is navigating the rise of AI. AI has created tremendous value in the software industry and will continue to do so. It will also create tremendous value outside the software industry. If you can help your company become good at AI, I hope you can play a leading role in creating a lot of this value.

It is video you saw what is it that makes a company a good AI company and also briefly the AI transformation playbook which I'll go into much greater detail on in a later week as a road-map for helping companies become great at AI. If you're interested, there is also published online an AI transformation playbook that goes into these five steps in greater detail for you see more of these in the later weeks as well.

Now, one of the challenges of doing AI projects such as the pilot projects in step one is understanding what AI can and cannot do. In the next video, I want to show you and give you some examples of what AI can and cannot do, to help you better select projects AI that there may be effective for your company. That's gone to the next video.

In this video and the next video, I hope to help you develop intuition about what AI can and cannot do. In practice, before I commit to a specific AI project, I'll usually have either myself or engineers do technical diligence on the project to make sure that it is feasible. This means: looking at the data, look at the input, and output A and B, and just thinking through if this is something AI can really do. What I've seen unfortunately is that some CEOs can have an inflated expectation of AI and can ask engineers to do things that today's AI just cannot do. One of the challenges is that the media, as well as the academic literature, tends to only report on positive results or success stories using AI, and we see a string of success stories and no failure stories, people sometimes think AI can do everything. Unfortunately, that's just not true. So, what I want to do in this and in the next video, is to show you a few examples of what today's AI technology can do, but also what it cannot do, and I hope that this will help you, hone your intuition about what might be more or less promising projects to select for your company.

Previously, you saw this list of AI applications from spam filtering to speech recognition, to machine translation, and so on. One imperfect rule of thumb you can use to decide what supervised learning may or may not be able to do is that, pretty much anything you could do with a second of thought, we can probably now or soon automate using supervised learning, using this input-output mapping.

So for example, in order to determine the position of other cars, that's something that you can do with less than a second. In order to tell if a phone is scratched, you can look at it and you can tell in less than a second. In order to understand or at least transcribe what was said, it doesn't take that many seconds of thought. While this is an imperfect rule of thumb, it maybe gives you a way to quickly think of some examples of tasks that AI systems can do.

Whereas in contrast, something that AI today cannot do would be: to analyze a market and write a 50 page report, a human cannot write a 50 page mark of analysis report in a second, and it's very difficult, at least I don't know. I don't think any team in the world today knows how to get an AI system to do market research and run an extended market report either. I've found out one of the best ways to hone intuition is to look at concrete examples. So, let's take a look at a specific example, relating to customer support automation.

Let's see a random website that sells things, so an e-commerce company, and you have a customer support division that gets an email like this, "The toy arrived two days late, so I wasn't able to give it to my niece for her birthday. Can I return it?" If what you want is an AI system that looks at this and decides this is a refund request, so let me route it to my refund department, then I will say, you have a good chance of building an AI system to do that. The AI system would take as input, the customer text, what the customer emails you, and it would output, is this a refund requests or is this a shipping problem, or is it the other request, in order to route this email to the most appropriate parts of your customer support center. So, the input A is the text and the output B is one of these three outcomes, is it a refund or a shipping problem, or shipping query, or is it a different requests. So, this is something that AI today can do.

Here's something that AI today cannot do which is if you want the AI to input an email and automatically generate, it responds like, "Oh, sorry to hear that. I hope you're niece had a good birthday. Yes, we can help with, and so on." So, for an AI to output a complicated piece of text like this today is very difficult by today's standards of AI and in fact to even empathize about the birthday of your niece, that is very difficult to do for every single possible type of email you might receive. Now, what would happen if you were to use a machine learning tool like a deep learning algorithm to try to do this anyway. So, let's say you tried to get an AI system to input the user's email, and output a two to the three paragraph, empathetic and appropriate response. Let's say that you have a modest-sized dataset like a 1,000 examples of user emails and appropriate responses. It turns out if you run an AI system on this type of data, on a small dataset like 1,000 examples, this may be the performance you get, which is if a user emails, "My box was damaged," they'll say, "Thank you for your email," and it says, "Where do I write a review?", "Thank you email." "What's the return policy?", "Thank you for your email." But the problem with building this type of AI is that

with just a 1,000 examples, there's just not enough data for an AI system to learn how to write to the three paragraph, appropriate and empathetic responses. So, you may end up just generating the same very simple response like, "Thank you for your email," no matter what the customer is sending you. Another thing that could go wrong, another way for an AI system to fail is if it generates gibberish such as: "When is my box arriving," and it says, "Thank, yes, now your," gibberish. This is a hard enough problem that even with 10,000 or a 100,000 email examples, I don't know if that would be enough data for an AI system to do this well. The rules for what AI can and cannot do are not hardened first and I usually end up having to ask engineering teams to sometimes spend a few weeks doing deep technical diligence to decide for myself if a project is feasible.

But to hone your intuitions to help you quickly filter feasible or not feasible projects, here are a couple of other rules of thumb about what makes a machine learning problem easier or more likely to be feasible.

One, learning a simple concept is more likely to be feasible. Well, what does a simple concept mean? There's no formal definition of that but it is something that takes you less than a second of mental thought or a very small number of seconds of mental thought to come up with a conclusion then that would lean to whether it being a simple concept. So, you're looking outside the window of a self-driving car to spot the other cars that would be a relatively simple concept. Whereas how to write an empathetic response, so a complicated user complaints, that would be less of a simple concept.

Second, a machine learning problem is more likely to be feasible if you have lots of data available. Here, our data means both the input A and the output B, that you want the AI system to have in your A to B, input to output mapping. So for example, in the customer support application, the input A would be examples of emails from customers and B could be labeling each of these customer emails as to whether it's a refund requests or a shipping query, or some other problem, one of three outcomes. Then if you have thousands of emails with both A and B, then the odds of you building a machine learning system to do that would be pretty good. AI is the new electricity and it's transforming every industry, but it's also not magic and it can't do everything under the sun.

I hope that this video started to help you hone your intuitions about what it can and cannot do, and increase the odds of your selecting feasible and valuable projects for maybe your teams to try working on. In order to help you continue developing your intuition, I would like to show you more examples of what AI can and cannot do. Let's go into the next video.

One of the challenges of becoming good at recognizing what AI can and cannot do is that it does take seeing a few examples of concrete successes and failures of AI. If you work on an average of say, one new AI project a year, then to see three examples would take you three years of work experience and that's just a long time. What I hope to do, both in the previous video and in this video is to quickly show you a few examples of AI successes and failures, or what it can and cannot do so that in a much shorter time, you can see multiple concrete examples to help hone your intuition and select valuable projects. So, let's take a look at a few more examples.

Let's say you're building a self-driving car, here's something that AI can do pretty well, which is to take a picture of what's in front of your car and maybe just using a camera, maybe using other senses as well such as radar or lidar. Then to figure out, what is the position, or where are the other cars. So, this would be an AI where the input A, is a picture of what's in front of your car, or maybe both a picture as well as radar and other sensor readings. The output B is, where are the other cars? Today, the self-driving car industry has figured out how to collect enough data and has pretty good algorithms for doing this reasonably well. So, that's what the AI today can do.

Here's an example of something that today's AI cannot do, or at least would be very difficult using today's AI, which is to input a picture and output the intention of whatever the human is trying to gesture at your car. So, here's a construction worker holding out a hand to ask your car to stop. Here's a hitchhiker trying to wave a car over. Here is a bicyclist raising the left-hand to indicate that they want to turn left. So, if you were to try to build a system to learn the A to B mapping, where the input A is a short video of our human gesturing at your car, and the output B is, what's the intention or what does this person want, that today is very difficult to do. Part of the problem is that the number of ways people gesture at you is very, very large. Imagine all the hand gestures someone could conceivably use asking you to slow down or go, or stop. The number of ways that people could gesture at you is just very, very large. So, it's difficult to collect enough data from enough thousands or tens of thousands of different people gesturing at you, and all of these different ways to capture the richness of human gestures. So, learning from a video to what this person wants, it's actually a somewhat complicated concept. In fact, even people have a hard time figuring out sometimes what someone waving at your car wants. Then second, because this is a safety critical application, you would want an AI that is extremely accurate in terms of figuring out, does a construction worker want you to stop, or does he or she wants you to go? And that makes it harder for an AI system as well. So, today if you collect just say, 10,000 pictures of other cars, many teams would build an AI system that at least has a basic capability at detecting other cars. In contrast, even if you collect pictures or videos of 10,000 people, it's quite hard to track down 10,000 people waving at your car. Even with that data set, I think it's quite hard today to build an AI system to recognize humans intentions from their gestures at the very high level of accuracy needed in order to drive safely around these people. So, that's why today, many self-driving car teams have some components for detecting other cars, and they do rely on that technology to drive safely. But very few self-driving car teams are trying to count on the AI system to recognize a huge diversity of human gestures and counting just on that to drive safely around people.

Let's look at one more example. Say you want to build an AI system to look at X-ray images and diagnose pneumonia. So, all of these are chest X-rays. So, the input A could be the X-ray image and the output B can be the diagnosis. Does this patient have pneumonia or not? So, that's something that AI can do. Something that AI cannot do would be to diagnose pneumonia from 10 images of a medical textbook chapter explaining pneumonia. A human can look at a small set of images, maybe just a few dozen images, and reads a few paragraphs from medical textbook and start to get a sense. But actually don't know, given a medical textbook, what is A and what is B? Or how to really pose this as an AI problems like know how to write a piece of software to solve, if all you have is just 10 images and a few paragraphs of text that explain what pneumonia in a chest X-ray looks like. Whereas a young medical doctor might learn quite well reading

a medical textbook at just looking at maybe dozens of images. In contrast, an AI system isn't really able to do that today.

To summarize, here are some of the strengths and weaknesses of machine learning.

Machine learning tends to work well when you're trying to learn a simple concept, such as something that you could do with less than a second of mental thought, and when there's lots of data available.

Machine learning tends to work poorly when you're trying to learn a complex concept from small amounts of data. A second underappreciated weakness of AI is that it tends to do poorly when it's asked to perform on new types of data that's different than the data it has seen in your data set.

Let me explain with an example. Say you built a supervised learning system that uses A to B to learn to diagnose pneumonia from images like these. These are well pretty high quality chest X-ray images. But now, let's say you take this AI system and apply it at a different hospital or different medical center, where maybe the X-ray technician somehow strangely had the patients always lie at an angle or sometimes there are these defects. Not sure if you can see the lost structures in the image. These low other objects lying on top of the patients. If the AI system has learned from data like that on your left, maybe taken from a high-quality medical center, and you take this AI system and apply it to a different medical center that generates images like those on the right, then it's performance will be quite poor as well. A good AI team would be able to ameliorate, or to reduce some of these problems, but doing this is not that easy. This is one of the things that AI is actually much weaker than humans. If a human has learned from images on the left, they're much more likely to be able to adapt to images like those on the right as they figure out that the patient is just lying on an angle. But then AI system can be much less robust than human doctors in generalizing or figuring out what to do with new types of data like these. I hope these examples are helping you hone your intuitions about what AI can and cannot do.

In case the boundary between what it can or cannot do still seems fuzzy to you, don't worry. It is completely normal, completely okay. In fact even today, I still can't look at a project and immediately tell is something that's feasible or not. I often still need weeks or small numbers of weeks of technical diligence before forming strong conviction about whether something is feasible or not. But I hope that these examples can at least help you start imagining some things in your company that might be feasible and might be worth exploring more. The next two videos after this are optional and are a non-technical description of what are neural networks and what is deep learning. Please feel free to watch those. Then next week, we'll go much more deeply into the process of what building an AI project would look like. I look forward to seeing you next week.

The terms deep learning and neural network are used almost interchangeably in AI. And even though they're great for machine learning, there's also been a bit of hype and bit of mystique about them. This video will demystify deep learning, so that you have a sense of what deep learning and neural networks really are.

Let's use an example from demand prediction. Let's say you run a website that sells t-shirts. And you want to know, based on how you price the t-shirts, how many units you expect to sell, how many t-shirts you expect to sell. You might then create a dataset like this, where the higher the price of the t-shirt, the lower the demand. So you might fit a straight line to this data, showing that as the price goes up, the demand goes down. Now demand can never go below zero, so maybe you say that the demand will flatten out at zero, and beyond a certain point you expect pretty much no one to buy any t-shirts. It turns out this blue line is maybe the simplest possible neural network. You have as input the price, A, and you want it to output the estimated demand, B. So the way you would draw this as a neural network is that the price will be input to this little round thing there, and this little round thing outputs the estimated demand. In the terminology of AI, this little round thing here is called a neuron, or sometimes it's called an artificial neuron, and all it does is compute this blue curve that I've drawn here on the left. This is maybe the simplest possible neural network with a single artificial neuron, that just inputs the price and outputs the estimated demand. If you think of this orange circle, this artificial neuron as a little Lego brick, all that a neural network is, if you take a lot of these Lego bricks and stack them on top of each other until you get a big power, a big network of these neurons.

Let's look at a more complex example. Suppose that instead of knowing only the price of the t-shirts, you also have the shipping costs that the customers will have to pay to get the t-shirts. Maybe you spend more or less on marketing in a given week, and you can also make the t-shirt out of a thick, heavy, expensive cotton or a much cheaper, more lightweight material. These are some of the factors that you think will affect the demand for your t-shirts. Let's see what a more complex neural network might look like. You know that your consumers care a lot about affordability. So let's say you have one neuron, and let me draw this one in blue, whose job it is to estimate the affordability of the t-shirts. And so affordability is mainly a function of the price of the shirts and of the shipping cost. A second thing though affecting demand for your t-shirts is awareness. How much are consumers aware that you're selling this t-shirt? So the main thing that affects awareness, is going to be your marketing. So let me draw here a second artificial neuron that inputs your marketing budget, how much you spend on marketing, and outputs how aware are consumers of your t-shirt. Finally, the perceived quality of your product will also affect demand, and perceived quality would be affected by marketing. The marketing tries to convince people this is a high quality t-shirt, and sometimes the price of something also affects perceived quality. So I'm going to draw here a third artificial neuron that inputs price, marketing and material, and tries to estimate the perceived quality of your t-shirts. Finally, now that the earlier neurons, these three blue neurons, have figured out how affordable, how much consumer awareness and what's the perceived quality, you can then have one more neuron over here that takes as input these three factors and outputs the estimated demand. So this is a neural network, and its job is to learn to map from these four inputs, that's the input A, to the output B, to demand. So it learns this input output or A to B mapping. This is a fairly small neural network with just four artificial neurons.

In practice, neural networks used today are much larger, with easily thousands, tens of thousands or even much larger than that numbers of neurons. Now, there's just one final detail of this description that I want to clean up, which is that in the way I've described the neural network, it was as if you had to figure out that the key factors are affordability, awareness and perceived quality. One of the wonderful things about using neural networks is that to train a neural network, in other words, to build a machine learning system using neural network, all you have to do is give it the input A and the output B. And it figures out all of the things in the middle by itself. So to build a neural network, what you would do is feed it lots of data, or the

input A, and have a neural network that just looks like this, with a few blue neurons feeding to a yellow open neuron. And then you have to give it data with the demand B as well. And it's the software's job to figure out what these blue neurons should be computing, so that it can completely automatically learn the most accurate possible function mapping from the input A to the output B. And it turns out that if you give this enough data and train a neural network that is big enough, this can do an incredible good job mapping from inputs A to outputs B. So that's a neural network, is a group of artificial neurons each of which computes a relatively simple function. But when you stack enough of them together like Lego bricks, they can compute incredibly complicated functions that give you very accurate mappings from the input A to the output B.

Now, in this video you saw an example of neural networks applied to demand prediction. Let's go on to the next video to see a more complex example of neural networks applied to face recognition.

In the last video, you saw how a neural network can be applied to demand prediction, but how can the new network look at the picture and figure out what's in the picture? Or listen to an audio clip and understand what is said in an audio clip? Let's take a look at a more complex example of applying a neural network to face recognition.

Say you want to build a system to recognize people from pictures, how can a piece of software look at this picture and figure out the identity of the person in it? Let's zoom in to a little square like that to better understand how a computer sees pictures. Where you and I see a human eye, a computer instead sees that, and sees this grid of pixel brightness values that tells it, for each of the pixels in the image, how bright is that pixel. If it were a black and white or grayscale image, then each pixel would correspond to a single number telling you how bright is that pixel. If it's a color image, then each pixel will actually have three numbers, corresponding to how bright are the red, green, and blue elements of that pixel. So, the neural networks job is to take as input a lot of numbers like these and tell you the name of the person in the picture.

In the last video, you saw how a neural network can take as input four numbers corresponding to the price, shipping costs, amounts of marketing, and cloth material of a T-shirt and output demand. In this example, the neural network just has to input a lot more numbers corresponding to all of the pixel brightness values of this picture. If the resolution of this picture is 1000 pixels by 1000 pixels, then that's a million pixels. So, if it were a black and white or grayscale image, this neural network was take as input a million numbers corresponding to the brightness of all one million pixels in this image or if was a color image it would take as input three million numbers corresponding to the red, green, and blue values of each of these one million pixels in this image. Similar to before, you will have many many of these artificial neurons computing various values, and it's not your job to figure out what these neurons should compute. The neural network will figured out by itself. Typically, when you give it an image, the neurons in the earlier parts of the neural network will learn to detect edges in pictures and then lobe and later learn to detect parts of objects. So, they learn to detect eyes and noses and the shape of cheeks and the shape of mouths, and then the later neurons, further to the right, will learn to detect different shapes of faces and it will finally, put all this together to output the identity of the person in the image. Again, part of the magic of neural networks is that you don't really need to worry about what it is doing in the middle. All you need to do is give it a lot of data of pictures like this, A, as well as the correct identity B and the learning algorithm will figure out by itself what each of these neurons in the middle should be computing.

Congratulations on finishing all the videos for this week. You now know how machine learning and data science work. I look forward to seeing you in next week's videos, as well where you'll learn how to build your own machine learning or data science project. See you next week.

Welcome back. Last week you learned about the basics of AI and machine learning technology. But how do you use this technology in a project, either if you want to do a project in your proverbial garage or if you want to do your project in a bigger company, or maybe even have something that builds up to align with your corporate strategy? Let's take a look.

First, in this week you learn what is the workflow of an AI project. Different projects have different steps. So, just as a birthday party has a sequence of predictable steps. First, you figure out the guest list, and you find a venue, then you order the birthday cake, and send invites, and so on. So, too does an AI project have a sequence of predictable steps. So, through this, you learn what is the workflow of an AI project, you learn what it feels like to be working on an AI project.

Second, how do you select an AI project? Seems there are a lot of things you could do. The second thing you learned this week is a framework for brainstorming and selecting potentially promising projects to work on, either by yourself or with a few friends, or as part of a bigger company effort.

Finally, you also learn how to organize the data as well as the team, which again could be just you are a few friends or much bigger corporate team. We learn how to organize the data and team for executing on an AI project. By the end of the week, you know what it feels like and how to build your own AI project, and maybe you'll be able to start exploring with some of your friends promising ideas to try. Let's go on to the next video.

Machine learning algorithms can learn input to output or A to B mappings. So, how do you build a machine learning project?

In this video, you'll learn what is the workflow of machine learning projects. Let's take a look. As a running example, I'm going to use speech recognition. So, some of you may have an Amazon Echo or Google Home or Apple Siri device or a Baidu DuerOS device in your homes. Some years back, I've done some work on Google's speech recognition system that also led Baidu's DuerOS project. Today, I actually have a Amazon Echo in my kitchen. So, every time I'm boiling an egg I will say, "Alexa, set timer for three minutes," and then it lets me know when the three minutes are up and my eggs are ready. So, how do you build a speech recognition system that can recognize when you say, "Alexa," or "Hey, Google," or "Hey, Siri," or "Hello, Baidu"? Let's go through the key steps of a machine learning project. Just for simplicity, I'm going to use Amazon Echo or detecting the Alexa keywords as this running example.

If you want to build an AI system or build a machine learning system to figure out when a user has said the word Alexa, the first step is to collect data. So, that means, you would go around and get some people to say the word "Alexa" for you and you record the audio of that. You'll also get a bunch of people to say other words like "Hello," or say lots of other words and record the audio of that as well. Having collected a lot of audio data, a lot of these audio clips of people saying either "Alexa" or saying other things, step two is to then train the model. This means you will use a machine learning algorithm to learn an input to output or A to B mapping, where the input A would be an audio clip. In the case of the first audio clip above, hopefully, it will tell you that the user said "Alexa," and in the case of audio clip two, shown on the right, hopefully, the system will learn to recognize that the user has said "Hello." Whenever an AI team starts to train the model, meaning to learn the A to B or input-output mapping, what happens, pretty much every time, is the first attempt doesn't work well. So invariably, the team will need to try many times or in AI, we call this iterate many times. You have to iterate many times until, hopefully, the model looks like is good enough. The third step is to then actually deploy the model. What that means is you put this AI software into an actual smart speaker and ship it to either a small group of test users or to a large group of users.

What happens in a lot of AI products is that when you ship it, you see that it starts getting new data and it may not work as well as you had initially hoped. So, for example, I am from the UK. So, I'm going to pick on the British. But let's say you had trained your speech recognition system on American-accented speakers and you then ship this smart speaker to the UK and you start having British-accented people say "Alexa." They may find that it doesn't recognize the speech as well as you had hoped. When that happens, hopefully, you can get data back of cases such as maybe British-accented speakers was not working as well as you're hoping, and then use this data to maintain and to update the model.

So, to summarize, the key steps of a machine learning project are to collect data, to train the model, the A to B mapping, and then to deploy the model. Throughout these steps, there is often a lot of iteration, meaning fine-tuning or adapting the model to work better or getting data back even after you've shipped it to, hopefully, make the product better, which may or may not be possible depending on whether you're able to get data back.

Let's look at these three steps and see how they apply on a different project on building a key component of a self-driving car. So, remember the key steps, you collect data, you train model, deploy model, since we'll revisit these steps on the next slide.

Let's say you're building a self-driving car. One of the key components of a self-driving car is a machine learning algorithm that takes as input, say a picture, of what's in front of your car and tells you where are the other cars. So, what's the first step of building this machine learning system? Hopefully, you remember from the last slide that the first step was to collect data. So, if your goal is to have a machine learning algorithm that can take as input an image and output the position of other cars, the data you would need to collect would be both images as well as position of other cars that you want the AI system to output. So,

let's say you start off with a few pictures like this. These are the inputs A to the machine learning algorithm. You need to also tell it what is the output B you would want. So, for each of these pictures, you would draw a rectangle around the cars in the picture that you wanted to detect. On this slide, I'm hand drawing these rectangles, but in practice, you will use some software that lets you draw perfect rectangles rather than these hand-drawn ones. Then, having created this dataset, what was the second step? Hope you remember that the second step was to train the model. Now, invariably, when your AI engineers start training a model, they'll find, initially, that it doesn't work that well. For example, given this picture, maybe the software, the first few tries, thinks that that is a car. It's only by iterating many times that you, hopefully, get a better result when figuring out that that is where the car actually is. Finally, what was the third step? It was to deploy the model. Of course, in the self-driving world, it's important to treat safety as number one, and deploy model or to test the model only in ways that can preserve safety. But when you put the software in cars on the road, you may find that there are new types of vehicles, say golf carts, that the software isn't detecting very well. So, you get data back, say, pictures of these golf carts, using new data to maintain and update the model so that, hopefully, you can have your AI software continually get better and better to the point where you end up with a software that can do a pretty good job detecting other cars from pictures like these.

In this video, you learned what are the key steps of a machine learning project, which are to collect data, to train the model, and then to deploy the model. Next, let's take a look at what are the key steps or what is a workflow of a data science project. Let's go on to the next video.

Unlike a machine learning project, the output of a data science project is often a set of actionable insights, a set of insights that may cause you to do things differently. So, data science projects have a different workflow than machine learning projects. Let's take a look at one of the steps of a data science project.

As our running example, let's say you want to optimize a sales funnel. Say you run a e-commerce or a online shopping website that sells coffee mugs and so for a user to buy a coffee mug from you, there's a sequence of steps they'll usually follow. First, they'll visit your website and take a look at the different coffee mugs on offer, then eventually, they have to get to a product page, and then they'll have to put it into their shopping cart, and go to the shopping cart page, and then they'll finally have to check out. So, if you want to optimize the sales funnel to make sure that as many people as possible get through all of these steps, how can you use data science to help with this problem? Let's look at the key steps of a data science project. The first step is to collect data. So, on a website like the one we saw, you may have a data set that stores when different users go to different web pages. In this simple example, I'm assuming that you can figure out the country that the users are coming from, for example, by looking at their computers' address, called an IP address, and figuring out what is the country from which they're originating. But in practice, you can usually get quite a bit more data about users than just what country they're from. The second step is to then analyze the data. Your data science team may have a lot of ideas about what is affecting the performance of your sales funnel. For example, they may think that overseas customers are scared off by the international shipping costs which is why a lot of people go to the checkout page but don't actually check out. If that's true then you might think about whether to put part of shipping costs into the actual product costs or your data science team may think there are blips in the data whenever there's a holiday. Maybe more people will shop around the holidays because they're buying gifts or maybe fewer people will shop around the holidays because they're staying home rather than sometimes shopping from their work computers. In some countries, there may be time-of-day blips where in countries that observe a siesta, so a time of rest like an afternoon rest, there may be fewer shoppers online and so your sales may go down. They may then suggest that you should spend fewer advertising dollars during the period of siesta because fewer people will go online to buy at that time. So, a good data science team may have many ideas and so they try many ideas or will say iterate many times to get good insights. Finally, the data science team will distill these insights down to a smaller number of hypotheses about ideas of what could be going well and what could be going poorly as well as a smaller number of suggested actions such as incorporating shipping costs into the product costs rather than having it as a separate line item. When you take some of these suggested actions and deploy these changes to your website, you then start to get new data back as users behave differently now that you advertise differently at the time of siesta or have a different check out policy. Then your data science team can continue to collect data and we analyze the new data periodically to see if they can come up with even better hypotheses or even better actions over time.

So the key steps of a data science project are to collect the data, to analyze the data, and then to suggest hypotheses and actions, and then to continue to get the data back and reanalyze the data periodically.

Let's take this framework and apply it to a new problem, to optimizing a manufacturing line. So we'll take these three steps and use them on the next slide as well. Let's say you run a factory that's manufacturing thousands of coffee mugs a month for sale and you want to optimize the manufacturing line. So, these are the key steps in manufacturing coffee mugs. Step one is to mix the clay, so make sure the appropriate amount of water is added. Step two is take this clay and to shape the mugs. Then you have to add the glaze, so add the coloring, a protective cover. Then you have to heat this mug and we call that firing the kiln. Finally, you would inspect the mug to make sure there aren't dents in the mug and it isn't cracked before you ship it to customers. So, a common problem in manufacturing is to optimize the yield of this manufacturing line to make sure that as few damaged coffee mugs get produced as possible because those are coffee mugs you have to throw away, resulting in time and material waste.

What's the first step of a data science project? I hope you remember from the last slide that the first step is to collect data. So for example, you may save data about the different batches of clay that you've mixed, such as who supplied the clay and how long did you mix it, or maybe how much moisture was in the clay, how much water did you add. You might also collect data about the different batches of mugs you made. So how much humidity was in that batch? What was the temperature in the kiln and how long did you fire it in the kiln? Given all this data you would then ask the data science team to analyze the data and they would, as before, iterate many times to get good insights. So, they may find that, for example, that whenever the humidity is too low and the kiln temperature is too hot that there are cracks in the mug or they may find out that because it's warmer in the afternoon that you need to adjust the humidity and temperature depending on the time of day. Based on the insights from your data science team you get suggestions for hypotheses and actions on how to change the operations and manufacturing line in order to improve the productivity of the line. When you deploy the changes, you then get new data back that you can reanalyze periodically so they can keep on optimizing the performance of your manufacturing line.

To summarize, the key steps of a data science project are to collect the data, to analyze the data, and then to suggest hypotheses and actions. In this video and the last video you saw some examples of machine learning projects and data science projects. It turns out that machine learning and data science are affecting almost every single job function. What I want to do in the next video is show you how these ideas are affecting many job functions, including perhaps yours and certainly that of many of your colleagues. Let's go on to the next video.

Data is transforming many different job functions, whether you work in recruiting or sales or marketing or manufacturing or agriculture, data is probably transforming your job function. What's happened in the last few decades is the digitization of our society. So, rather than handing out papers surveys like these, surveys are more likely to be done in digital format or doctors still write some handwritten notes but doctors handwritten note is increasingly likely to be a digital record and so to this in just about every single job function. This availability of data means that there's a good chance that your job function could be helped with tools like data science or machine learning.

Let's take a look. In this video, I want to run through many different job functions and discuss how data science and machine learning can or will impact these different types of jobs. Let's start with sales. You've already seen in the last video how data science can be used to optimize a sales funnel. How about machine learning?

If you're a salesperson you may have a set of leads about different people that you could reach out to to convince them to buy something from your company. Machine learning can help you prioritize these leads. So, you might want to prioritize calling up the CEO of a large company rather than the intern at a much smaller company and this type of automated leads sorting is making salespeople more efficient.

Let's look at more examples. Let say you're manufacturing line manager. You've already seen how data science can help you optimize a manufacturing line. How about machine learning? One of the steps of this manufacturing process is the final inspection. In fact today, in many factories there can be hundreds or thousands of people using the human eye to check over objects, maybe coffee mugs, maybe other things to see if they're scratches or dents and that's called inspection. So, machine learning can take this input, a dataset like this, and learn to automatically figure out if a coffee mug is defective or not. By automatically finding scratches or dents, it can reduce labor costs and also improve quality in your factory. This type of automated visual inspection is one of the technologies that I think will have a big impact on manufacturing. This is something I've been working on myself as well.

Let's see more examples. How about recruiting? When recruiting someone to join your company, there may be a pretty predictable sequence of steps where your recruiter or someone else would send an email to a candidate and then you'd have a phone call with them, bring them on-site for an interview and then extend an offer and maybe close the offer. Similar to how data science can be used to optimize a sales funnel, recruiting can also use data science to optimize a recruiting funnel and in fact many recruiting organizations are doing so today. For example, if you find that hardly anyone is making it from phone screen step to the on-site interviews step then you may conclude that maybe too many people are getting into the phone screen stage or maybe the people doing the phone screen are just being too tough and you should let more people get to the onsite interview stage. This type of data science is already having an impact on recruiting. What about machine learning projects? Well, one of the steps of recruiting is to screen a lot of resumes to decide who to reach out to you. So, you may have to look at one resume and say, "Yes, let's email them", look at a different one to say, "No, let us not move ahead with this candidate." Machine learning is starting to make its way into automated resume screening.

Does raise important ethical questions such as making sure that your AI software does not exhibit undesirable forms of bias and treats people fairly, but machine learning is starting to make inroads into this and I hope can do so while making sure that the systems are ethical and fair. In the final week of this AI For Everyone course, you'll also learn more about the issues of fairness and ethics in AI.

What if you work in marketing? One of the common ways to optimize the performance in website is called AB testing, in which you launch two versions of website. Here version A has a red button, version B has a green button and you measure which websites causes people to click through more. So with this type of data, a data science team can help you gain insights and suggests hypotheses or actions for optimizing your website. How about machine learning and marketing? Today a lot of websites will give customized product

recommendations to show you the things you're most likely to want to buy and this actually significantly increases sales on these websites. For example, a clothing website after seeing the way I shop after while, will hopefully just recommend blue shirts to me because that's frankly pretty much the only type shirt I ever buy, but maybe other customers will have more diverse and more interesting recommendations than mine. But today these customized product recommendations actually drive a large percentage of sales on many large online e-commerce websites.

One last example from a totally different sector. Let's say you work in agriculture. Maybe you're a farmer working on the light industrial farm, how can data science help you? Today farmers are already using data science for crop analytics, where you can take data on the soil conditions, the weather conditions, the presence of different crops in the market and have data science teams make recommendations to what to plant, when to plant so as to improve use while maintaining the condition of the soil on your farm. This type of data science is and will play a bigger and bigger role in agriculture. Let's also look at the machine learning example. I think one of the most exciting changes to agriculture is precision agriculture. Here's a picture that I took on a farm with my cell phone. On the upper right is a cotton plants and shown in the middle is a weed. With machine learning, we're starting to see products that can go onto the farms, take a picture like this and spray a weed killer in a very precise way just onto the weeds so that it gets rid of the weed but without having to spray an excessive amounts of weed killers. This type of machine learning technology is both helping farmers increase crop yields while also helping to preserve the environment.

In this video, you saw how all of these job functions, everything from sales, recruiting to marketing to manufacturing to farming agriculture, how all of these job functions are being affected by data, by data science and machine learning. It seems like there's a lot of different things you could do with AI. But how do you actually select a promising project to work on? Let's talk about that in the next video.

If you want to try your hand at an AI project, how do you select a worthwhile project to work on? Don't expect an idea to naturally come overnight. Sometimes it happens, but sometimes it also takes a few days or maybe a few weeks to come up with a worthy idea to pursue.

In this video, you see a framework for brainstorming potentially exciting AI projects to pursue. Let's say you want to build an AI project for your business. You've already seen that AI can't do everything, and so there's going to be a certain set of things that is what AI can do. So let's let the circle represent the set of things that AI can do. Now, there's also going to be a certain set of things that is valuable for your business. So let's let this second circle represent a set of things that are valuable for your business. What you would like to do is try to select projects that are at the intersection of these two sets, so you select projects hopefully that are both feasible, that can be done with AI, and that are also valuable for your business. So AI experts will tend to have a good sense of what is and what isn't in the set on the left. And domain experts, experts in your business, be it sales and marketing, or agriculture or something else, will have the best sense of what is actually valuable for your business.

So when brainstorming projects that AI can do and are valuable for your business, I will often bring together a team comprising both people knowledgeable of AI, as well as experts in your business area to brainstorm together. So that together they can try to identify projects at the intersection of both of these two sets. So sometimes we also call these cross-functional teams, and that just means a team that includes both AI experts, as well as domain experts, meaning experts in your area of business. When brainstorming projects, there's a framework that I've used with a lot of companies that I found to be useful. So let me share with you three principles or three ideas for how you can have a team brainstorm projects.

First, even though there's been a lot of press coverage about AI automating jobs away, and this is an important societal issue that needs to be addressed, when thinking about concrete AI projects, I find it much more useful to think about automating tasks rather than automating jobs. Take call center operations, there are a lot of tasks that happen in a call center. Ranging from people picking up the phone to answering phone calls to replying to emails, to taking specific actions, such as issuing a refund on behalf of a customer request. But of all of these tasks that employees in a call center do, there may be one, call routing or email routing, that maybe particularly amenable to machine learning automation. And it's by looking at all these tasks that the group of employees do and selecting one that may allow you to select the most fruitful project for automation in the near term.

Let's look at another example, the job of a radiologist. There's been a lot of press about how AI may automate radiologists' jobs, but radiologists actually do a lot of things. They read x-rays, that's really important, but they also engage in their own continuing education. They consult with other doctors, they may mentor younger doctors, some of them also consult directly with patients. And so it's by looking at all of these tasks that a radiologist does that you may identify one of them, let's say AI assistance or AI automation for reading x-rays, that allows you to select the most fruitful projects to work on. So what I would recommend is, if you look at your business, think about the tasks that people do, to see if you can identify just one of them, or just a couple of them, that may be automatable using machine learning. When I'm meeting CEOs of large companies to brainstorm AI projects for the company, a common question I'll also ask is, what are the main drivers of business value? And sometimes finding AI solutions or data science solutions to augment this can be very valuable.

Finally, a third question that I've asked that sometimes led to valuable project ideas is, what are the main pain points in your business? Some of them could be solved with AI, some of them can't be solved with AI. But by understanding the main pain points in the business, that can create a useful starting point for brainstorming AI projects as well.

I have one last piece of advice for brainstorming AI projects, which is that you can make progress even without big data, even without tons of data. Now don't get me wrong, having more data almost never

hurts, other than maybe needing to pay a bit more for disk space or network bandwidth to transmit and store the data, having more data almost always is only helpful. And I love having lots of data. It is also true that data makes some on businesses, like web search, defensible.

Web search is a long tail business, meaning that there are a lot of very, very rare web queries. And so seeing what people click on when they search on all of these rare web queries does help the leading web search engines have a much better search experience. So big data is great when you can get it, but I think big data also sometime over-hyped, and even with a small dataset, you can still often make progress. Here's an example, let's say you're building a automated visual inspection system for the coffee mug. So you want to automatically detect that the coffee mug on the right is defective. Well, if you had a million pictures of good coffee mugs and defective coffee mugs, it'd be great to have that many examples of pictures of good and bad coffee mugs to feed your AI system. But I hope you have not manufactured 1 million defective coffee mugs, because that feels like a very expensive thing to have to throw away. So sometimes with as few as 100, or maybe 1,000, or sometimes maybe as few as 10, you may be able to get started on the machine learning project. The amount of data you need is very problem dependent, and speaking with an AI engineer or AI expert would help you get better sense. There are some problems where 1,000 images may not be enough, where you do need big data to get good performance. But my advice is, don't give up just because you don't have a lot of data to start off with. And you can often still make progress, even with a small dataset.

In this video, you saw a brainstorming framework, and a set of criteria for trying to come up with projects that hopefully can be doable with AI, and are also valuable for your business. Now, having brainstormed a list of projects, how do you select one or select a small handful to actually commit to and work on? Let's talk about that in the next video.

Maybe you have a lot of ideas for possible AI projects to work on. But before committing to one, how do you make sure that this really is a worthwhile project? If it's a quick project that might take you just a few days maybe just jump in right away and see if it works or not, but some AI projects may take many months to execute.

In this video, I want to step you through the process that I use to double-check if a project is worth that many months of effort. Let's take a look. Before committing to a big AI project, I will usually conduct due diligence on it. Due diligence has a specific meaning in the legal world. But informally, it just means that you want to spend some time to make sure what you hope is true really is true. You've already seen how the best AI projects are ones that are feasible. So, it's something that AI can do, as well as valuable. We really want to choose projects to that the intersection of these two sets. So, to make sure a project is feasible, I will usually go through technical diligence, and make sure that the project is valuable, I will usually go through a business diligence process. Let me tell you more about these two steps.

Technical diligence is the process of making sure that the AI system you hope to build really is doable, really is feasible. So, you might talk to AI experts about whether or not the AI system can actually meet the desired level of performance. For example, if you are hoping to build a speech system that is 95 percent accurate, consulting of AI experts or perhaps reading some of the trade literature can give you a sense of whether this is doable or not. Or if you want a system to inspect coffee mugs in a factory and you need your system to be 99 percent accurate. Again, is this actually doable with today's technology? A second important question for technical diligence is how much data is needed to get to this desired level of performance, and do you have a way to get that much data. Third, would be engineering timeline to try to figure out how long it will take and how many people it will take to build a system that you would like to have built.

In addition to technical diligence, I will often also conduct business diligence to make sure that the project you envision really is valuable for the business. So, a lot of AI projects will drive value through lowering costs. For example, by automating a few tasks or by squeezing more efficiency onto the system. A lot of AI systems can also increase revenue. For example, driving more people to check out in your shopping cart or you may be building an AI system to help you launch a new product or a new line of business. So, business diligence is the process of thinking through carefully for the AI system that you're building such as a speech recognition system that's 95 percent accurate or a visual inspection system that's 99.9 percent accurate, would allow you to achieve your business goals. Whether your business goal is to improve your current business or to even create brand new businesses in your company. When conducting business diligence, I'll often end up building spreadsheet financial models to estimate the value quantitatively such as estimate how many dollars are actually saved or what do we think is a reasonable assumption in terms of entries revenue, and to model out the economics associated with a project before committing to many months of effort on a project. Although not explicitly listed on this slide, one thing I hope you also consider doing as a third type of diligence which is ethical diligence. I think there are a lot of things that AI can do that will even make a lot of money, but that may not make society better off. So, in addition to technical diligence and business diligence, I hope you also conduct ethical diligence and make sure that whatever you're doing is actually making humanity and making society better off. We'll also talk more about this in the last week of this course as well.

As you're planning out your AI project, you also have to decide do you want to build or buy? This is an age old question in the IT world and we're facing this question in AI as well. For example, hardly any companies build their own computers these days. They buy someone else's computers and hardly any companies build their own Wi-Fi routers, just buy a commercial Wi-Fi router. How about machine learning and data science? Machine learning projects can be in-house or outsource. I've seen both of these models used successfully. Sometimes if you outsource a machine learning project, you can have access much more quickly to talent and get going faster on a project. It is nice if eventually you build your own in-house AI team and can also

do these projects in house. You'll hear more about this when we talk about AI translation playbook in greater detail next week. Unlike machine learning projects though, data science projects are more commonly done in-house. They're not impossible to outsource, you can sometimes outsource them, but what I've seen is that data science projects are often so closely tied to your business that it takes very deep day-to-day knowledge about your business to do the best data science projects. So, just as a percentage, as a fraction, I see data science projects in-house more than machine learning projects.

Finally, in every industry some things will be industry standard and you should avoid building those. A common answer to the build versus buy question was, build the things that are going to be quite specialized to you or completely specialized to you or they'll allow you to build a unique defensible advantage, but the things that will be industry standard probably some other company will build and it'll be more efficient for you to just buy it rather than build it in-house.

One of my teams have a really poetic phrase which is, "Don't sprint in front of a train," and what that means is, if this is a train running on a railway tracks and that's the small chimney with the puff of smoke. What you don't want to do is to be the person or the engineer trying to sprint faster and faster ahead of the train. The train is the industry standard solution, and so, if there's a company maybe a startup, maybe a big company or maybe an open-source effort that is building an industry standard solution, then you may want to avoid trying to run faster and faster to keep ahead of the train. Because even though you could sprint faster in the short term, eventually the train will catch up and crash someone trying to sprint in front of a train. So, when there's a massive force of an industry standard solution that is been built, you might be better off just embracing an industry standard or embracing someone else's platform rather than trying to do everything in-house. We all live in a world of limited resources, limited time, limited data, limited engineering resources, and so, I hope you can focus those resources on the projects with our most unique and will make the biggest difference to your company. Through the process of technical diligence as well as business diligence, I hope you can start to identify projects that are potentially valuable or that seem promising for your business. If the project is a big company, maybe it'll take many months to do. It's not unusual for me to spend even a few weeks conducting this type of diligence before committing to a project. Now, say you found a few promising projects, how do you engage with an AI team? How do you work with an AI team to try to get these projects done? Let's talk about that in the next video.

Say you found an exciting project that you want to try to execute on, how do you work with an AI team on this project?

In this video, you learn how AI teams think about data and therefore how you can interact with AI teams to help them succeed on a project. Now, there is one caveat which is, whether you have a cool idea but you don't have access to an AI team, you don't have any access to any AI engineers. Fortunately in today's world, if either yourself or you can encourage some of your engineering friends to take an online course or two on machine learning or deep learning that often will give them enough knowledge to get going and make a start of an attempt, make a reasonable attempt on these types of projects. So, let's talk about how you can work with an AI team.

First, it really helps your AI team if you can specify an acceptance criteria for the project. I've done a lot of work in automated vision inspection. So, I'm going to use that as a running example in these few slides. Let's say your goal is to detect defects in coffee mugs with at least 95% accuracy. So, that can be your acceptance criteria for this project. But 95% accuracy, how do you measure accuracy? One of the things that AI team would need is a dataset on which to measure the accuracy. So, dataset is just a set of pictures like these together with the labels, with the design output B that the first two coffee mugs are okay and the third one is defective. So, as part of your specification for the acceptance criteria you should make sure that the AI team has a dataset on which to measure the performance so that they can know if they've achieved 95% accuracy. The formal term for this dataset is called a test set. The test set may not need to be too big, maybe a 1,000 pictures will be just fine for this example. But, if you consult with an AI expert they can give you a better sense of how big the test set needs to be for them to be able to evaluate whether or not they're getting to 95% accuracy. One novel part of AI systems is that their performance is usually specified in a statistical way. So, rather than asking for an AI system that just does something perfectly you see very often that we want an AI system that performs at a certain percentage accuracy like this example here. So, when specifying your acceptance criteria think of whether your acceptance criteria needs to be specified in a statistical way where you specify on average how often it does or what percent of time it has to get the right answer.

Let's dive more deeply into the concept of a test set. This is how AI teams think about data. AI teams group data into two main datasets. The first called the training set and the second called the test set which we've already talked a bit about. The training set is just a set of pictures together with labels showing whether each of these pictures is of a coffee mug that is okay or defective. So, the training set gives examples of both the input A the pictures of the coffee mugs as well as the desired output B whether it's okay or defective. So, given this training set what a machine learning algorithm will do is learn in other words computes or figure out some mapping from A to B so that you now have a piece of software that can take as input the input A and try to figure out what is the appropriate output B. So, the training set is the input to the machine learning software that lets it figure out what is this A to B mapping. The second dataset that an AI team will use is the test set. As you've seen this is just another set of images that's different from the training sets also with the labels provided. The way an AI team will evaluate their learning algorithms performance is to give the images into the test set to the AI software and see what the AI software outputs. For example, if on this three test set images the AI software outputs okay for this, okay for this, and also okay for this, then we will say that they got two out of three examples right and so that's a 66.7% accuracy. In this figure the training set and test set above have only three pictures, in practice both of these datasets would be much bigger of course. You find that for most problems the training set is much bigger than the test set. But you can talk to AI engineers to find out how much data they need for a given problem.

Finally, for technical reasons some AI teams will need not just one but two different test sets. If you hear AI teams talk about development or dev or validation tests that's the second test set that they're using. The reasons why they need two test set is quite technical and beyond the scope of this course, but if an AI team asks you for two different test sets it's quite reasonable to try to provide that to them.

Before wrapping up this video one pitfall I want to urge you to avoid is expecting a 100% accuracy from your AI software. Here's what I mean, let's say this is your test set which you've already seen on the last slide. But, let me add a few more examples to this test set. Here are some of the reasons it may not be possible for a piece of AI software to be a 100% accurate.

First, machine learning technology today despite being very powerful still has limitations and they just can't do everything. So, you may be working on a problem that it's just very difficult even for today's machine learning technology. Second, insufficient data. If you don't have enough data specifically if you don't have enough training data for the AI software to learn from it may be very difficult to get a very high level of accuracy. Third, data is messy and sometimes data can be mislabeled. For example, this green coffee mug here looks perfectly okay to me, so, the label of it being a defect looks like an incorrect label and that would hurt the performance of your AI software. Data can also be ambiguous. For example, it looks like this coffee mug has a small scratch over there and it's a pretty small scratch, so, maybe we will think of this though okay. But maybe this should actually have been a defect or maybe even different experts won't agree if this particular coffee mug is okay and should pass the inspection step. Some of these problems can be ameliorated.

For example, if you don't have enough data maybe you can try to collect more data and more data will often help. Or you can also try to clean up mislabeled data or try to get your factories experts to come to better agreement about these ambiguous labels. So, there are ways to try to make these things better, but, a lot of AI systems are incredibly valuable even without achieving a 100% accuracy. So, I will urge you to discuss with your AI engineers what is a reasonable level of accuracy to try to accomplish? Then try to find something that passes both technical diligence as well as business diligence without necessarily needing a 100% accuracy.

Congratulations on finishing all the videos for this week. You now know what it feels like and what it takes to build an AI project. I hope you e start brainstorming and exploring some ideas. There is one more optional video describing some of the technical tools that AI teams use that you can watch if you wish. But either way I look forward to seeing you next week, where you'll learn how AI projects fit in the context of a bigger company. Look forward to seeing you next week.

When you work with AI teams, you may hear them refer to the tools that they're using to build these AI systems. In this video, I want to share with you some details and names of the most commonly used AI tools, so that you'd be able to better understand what these AI engineers are doing. We're fortunate that the AI world today is very open, and many teams will openly share ideas with each other. There are great machine learning open source frameworks that many teams will use to build their systems. So, if you hear of any of these:

TensorFlow, PyTorch, Keras, MXNet, CNTK, Caffe, PaddlePaddle, Scikit-learn, R or Weka, all of these are open source machine learning frameworks that help AI teams be much more efficient in terms of writing software. Along with AI technology breakthroughs are also published freely on the Internet on this website called Arxiv. Spelled like this. I hope that other academic communities also freely share their research since I've seen firsthand how much this accelerates progress in the whole field of AI.

Finally, many teams will also share their code freely on the Internet, most commonly on a website called GitHub. This has become the de facto repository for open source software in AI and in other sectors in AI. And by using appropriately licensed open-source software, many teams can get going much faster than if they had to build everything from scratch. So, for example, if I search online for face recognition software on GitHub, you might find a web page like this. And if you scroll down, this actually has a pretty good, very readable description of software that is made available on this website for recognizing people's faces, and even finding parts of people's faces. There's just a ton of software that is freely downloadable for doing all sorts of things on the Internet, and just double-check the license, or AI team will double-check the license before using it in a product of course, but a lot of this software is open source, or is otherwise very permissively licensed for anyone to use.

Although GitHub is a technical website built for engineers, if you want you should feel free to play around GitHub and see what are the types of AI software people have released online as well.

In addition to these open source technical tools, you often also hear AI engineers talk about CPUs and GPUs. Here's what these terms mean. A CPU is the computer processor in your computer, whether it's your desktop, your laptop, or a computer server off in the Cloud. CPU stands for the central processing unit, and CPUs are made by Intel, and AMD, and a few other companies. This does a lot of the computation in your computer. GPU stands for graphics processing unit. Historically, the GPU was made to process pictures. So, if you play a video game, it's probably a GPU that is drawing the fancy graphics. But what we found several years ago was that the hardware that was originally built for processing graphics turns out to be very, very powerful for building very large neural networks. So very large deep learning algorithms. Given the need to build very large deep learning or very large neural network systems, the AI community has had this insatiable hunger for more and more computational power to train bigger and bigger neural networks. And GPUs have proved to be a fantastic fit to this type of computation that we need to have done to train very large neural networks. So, that's why GPUs are playing a big role in the rise of deep learning. And video is a company that's been selling many GPUs.

But other companies including Qualcomm, as well as Google making their own CPUs are increasingly making specialized hardware for powering these very large neural networks. Finally, you might hear about Cloud versus On-premises, or for short, On-prem deployments. Cloud deployments refer to if you rent compute servers such as from Amazon's AWS, or Microsoft's Azure, or Google's GCP in order to use someone else's service to do your computation. Whereas, an On-prem deployment means buying your own compute servers and running the service locally in your own company. A detailed exploration of the pros and cons of these two options is beyond the scope of this video.

A lot of the world is moving to Cloud deployments. Whether you search online you find many articles talking about the pros and cons of Cloud versus On-prem deployments. There is one last term you might hear about, which is Edge deployments. If you are building a self-driving car, there's not enough time to

send data from a self-driving car to a Cloud server to decide if you should stop the car or not, and then send that message back to the self-driving car. So, the computation has to happen usually in a computer right there inside the car. That's called an edge deployment, where you put a processor right where the data is collected so that you can process the data and make a decision very quickly without needing to transmit the data over the Internet to be processed somewhere else. If you look at some of the smart speakers in your home as well, this too is an edge deployment where some, not all, but some of the speech recognition task is done by a processor that is built in right there into this smart speaker that is inside your home. The main advantage of Edge deployment is it can increase response time of the system, and also reduce the amount of data you need to send over the network. But there are many pros and cons as well about Edge versus Cloud versus On-prem deployments that you can also search online to read more about.

Thanks to finishing this optional video on the technical tools that AI engineers use. Hopefully, when you hear them refer to some of these tools, you'll start to have a better sense of what they mean. I look forward to seeing you next week.

Welcome back. In the last two weeks you learned what is AI, and how to build an AI project. This week we'll look at the projects we talked about already, and talk about how projects fit in the context of a company. Whether it's a for-profit, non-profit, or even within a government entity. For the sake of concreteness, I'll talk about building AI for companies which will equally apply, no matter what type of organization you're in. Now, in case some of what you hear this week sounds like CEO level talk, don't be intimidated. This is actually useful for everyone to know, the ones that hope, your company will help your organization improve using AI. And it does take maybe two or three years for a company to become good at AI. To embrace not just one AI project but a sequence of valuable AI projects and become much more effective. But what I hope to do this week is help you paint a vision for where an organization can go over a longer period of time like that. But also end the week with very concrete short steps you can take right away. So let's get started. In detail, these are the topics you'll see this week.

First is case studies of complex AI products. So rather than a single machine learning or data science module like what you saw last week. This week you'll see how multiple modules can come together to build a much more complex AI product such as a smart speaker or a self driving car. You also learn what are the major roles in an AI team. So if ever you think your company can build a large AI team with maybe many dozens or maybe even hundreds of people, what are the things these people will do? We'll start to paint a roadmap for what building an AI team might look like. Third you also learn about the AI transformation playbook of how to help your company become good at AI. Where it goes beyond doing one or two valuable projects, but about making a whole company good at AI and hopefully much more effective and valuable as a result. Finally, whereas some of these steps may take a small number of years to complete.

We'll wrap this week's videos with concrete suggestions on how you can take your first step, right away, toward building AI in your company. Beyond these major topics, we'll also have a couple of optional videos at the end, where you see a survey of major AI application areas and techniques. So, after this week's videos, I hope you have a good vision of how you can help your company use AI and also have first steps you can take right away. Let's get started by going on to the next video.

What does it feel like to work on a complex AI product, where isn't just using a single machine learning algorithm to map from A to B, but that learning algorithm is part of a bigger more complex project or product. I want to start it this week with two case studies of building complex AI products.

First, building a smart speaker so that you can start to understand what it might feel like to maybe someday work on a complex AI product within your own company. Let's get started. Smart speakers and voice activated devices like these are taking the world by storm and if you don't already have one in your home, maybe you'll buy one someday. I'd like to go through a case study of how you would write AI software to get a smart speaker to respond to a verbal command such as "Hey device, tell me a joke." For this example, rather than using Alexa, or Okay Google, or Hey Siri, or Hello Baidu as the wake word or trigger word, to be more device agnostic, I'm just going to use "Hey device" as the trigger word or wake word to wake up, say, a smart speaker and let's say you wanted to tell you a joke. So how do you build a piece of AI software to understand a command like this and executes on it? These are the steps needed to process the command. There can be four steps.

Step one is the trigger word or the wake word detection. The speaker uses a machine learning algorithm to input the audio clip and output. Did they just hear the wake word or the trigger word "Hey Device," so outputs 0 or one and once it hears the trigger word or wake word,

once it hears "Hey device," it then has to perform step two, which is speech recognition. So what the software has to do is take the audio of what came after "Hey device" and map that to "Tell me a joke." This is also done with machine learning. Whereas the first step here used an A to B mapping to just tell it that it heard the trigger word, this uses a different A to B mapping to map the audio to a text transcript of what you just said, which in this case is the four words: tell me a joke. Now, the algorithm has to figure out what you actually want by saying these four words.

And so the third step is intent recognition. That means to take what have you said and to figure out what you actually wanted to do. So today's smart speakers have a limited set of commands such as they can tell a joke or they can tell the time. So you can say "Hey device what time is it." They can play music. They can sometimes help you make a phone call. They can tell you what's the weather, "Hey device, what's the weather tomorrow?" So, what intent recognition does is take the four words, the speech recognition's output, and use another piece of AI software, another A to B mapping, that inputs those four words and outputs which of these five or other intents do you have. So in this implementation of a machine learning algorithm, the input A is the text transcript "tell me a joke" and the output B is which of these five types of commands did the user just utter. Of course, your smart speaker may be able to understand even more commands than just these five, in which case B would be whichever of the five or 20 or 100 commands your smart speaker knows how to execute. No matter how you ask the smart speaker to tell you a joke, hopefully, the intent recognition components will recognize your intent correctly. So that you can also say not just "Hey device, tell me a joke," but also "Hey device, do you know any good jokes" or "Hey device, tell me something funny." Turns out, there are lot of ways for you to ask a smart speaker for a joke and a well-designed intent recognition system should recognize most of them.

Finally, now that your smart speaker has figured out that you really, really want to hear a joke, the last step is that there will be a software engineer that has written a piece of code to randomly select a joke and to play the joke back through the speaker. In other words, they'll execute a joke. Just for the record, my favorite joke is why are there so many shocking results in AI, because AI is the new electricity. Shocking electricity, get it? Hope you enjoyed that.

So, all right, and in seriousness, you can think of the four steps of the algorithm as these four steps, where the first step is trigger word detection, second step, speech recognition, then intent recognition, and then finally, execution of what are the command the user asked the smart speaker to execute. So the process of having four steps in an AI system like this or multiple steps, this is sometimes called an AI pipeline, where

you have multiple AI components. Yes, it's possible to have machine learning components which process data one step after another and it would not be unusual to have, say, four different teams in the company where each team focuses on one of the components of this AI pipeline. That's how we often organize projects within a large company.

Let's now look at a more complex example. One of you issue a more complex commands like "Hey device, set timer for 10 minutes." These are the steps needed to process the command. First step, same as before, is trigger word detection. So input an audio and just let me know when someone said the trigger word hey device. Then speech recognition, where you input the rest of the audio and transcribe the rest of the sound, the rest of the audio, "set timer for 10 minutes." And now, intent recognition has to input that text and output that your intent is that you want to set a timer. One difference between "set timer for 10 minutes" compared to the earlier example of "tell me a joke" is that you need to know how long to actually set the timer for. So in the execution step, you actually need to do two things. One is extract the duration. That means, look at the text, set timer for 10 minutes and pull out the phrase that tells you how long to actually set the timer for. And so if the user were to say "Hey device, let me know when 10 minutes is up," then this extract duration step would have to pull out, again, the 10 minute phrase right there. And of course, there are lots of ways to ask for a 10-minute timer. You can also say, "let me know when 10 minutes are up" or "set an alarm for 10 minutes from now" and hopefully, the intent recognition and extract duration components will both be robust enough to recognize that all of these are different ways of asking for a 10-minute timer. Finally, to execute the command, there should be a specialized software component in the smart speaker that can start a timer with a set duration. And after it has extracted your intent and the duration, it would just start the timer with that duration, so that the alarm goes off at the end of 10 minutes.

Today's smart speakers have many functions. Other than the two we've talked about of telling jokes and setting a timer, here are some other functions that many smart speakers today can execute. And the key steps of executing these commands are trigger word or the wake word detection, speech recognition to transcribe the text in the command, intent recognition to figure out which of these functions or which of these commands you want to execute, and then a specialized program to execute whichever command you uttered. One of the challenges of the smart speaker world is that, if you want your smart speaker to have this many different functions, say, 20 different functions, then you do need software engineering teams to write 20 specialized pieces of software. One to play music, one to set the volume, one to make calls, one to ask for the current time, one to convert units like from teaspoons to tablespoons, or to answer very simple questions, and on and on. So it's actually quite a lot of work to write all of these specialized programs to execute all the different commands you might want to execute in step four. And smart speakers today actually do so many things that is difficult for many users to keep straight in their heads exactly what they can and cannot do. So many smart speaker companies have been investing a lot in user training to try to let users know what are the things that smart speakers can do. Because on one hand, they can't do everything. There are lot of things you can't ask smart speakers to do such as please call all of my three friends and see when all of them are able to meet for dinner. So it's been an ongoing processes of smart speaker companies to explain to users what they can and cannot do. Nonetheless, with what they can do using voice to command, these speakers is making life much more convenient for many people. I hope this video gave you a sense of what it takes to build a complex AI product such as a smart speaker. In order to help you better understand how these complex products work, let's go on to see a second case study of how to piece together multiple AI components to build a self-driving car. Let's go on to the next video.

One of the most exciting products of the AI era is the self-driving car. Self-driving cars are also one of the most mysterious things you hear about in AI these days. In this video what I want to do is share with you a somewhat simplified description of a self-driving car so that you understand how you can piece together multiple AI components in order to build these amazing things.

Let's get started. These are the key steps for deciding how to drive your self-driving car. The car will take as input various sensors such as pictures of what's in front of the car or to the sides or behind, as well as maybe radar or Lidar meaning laser sensor readings. Given these inputs of a picture and maybe other things, it then has to detect other cars. So given that, hopefully, you'll figure out there's a car there, as well as where are the pedestrians, because we want to avoid other cars as well as avoid pedestrians. Both car detection and pedestrian detection can be done with machine learning using input/output or A to B mappings that takes as input the picture and maybe radar and Lidar, sends the inputs and tells us where are the other cars and pedestrians. Finally, now that you know where are the other cars and where are the pedestrians, you can then feed this information into another specialized piece of software, it's called a motion planning piece of software that plans the motion or plans the path that you want your car to take, so that you can make progress to your destination while avoiding any collisions. Once you've planned out the motion for your car, you can then translate that into specific steering angle for your steering wheel and acceleration and brake commands, so how much to step on the gas pedal, and how much to brake in order to get your car to move at the desired angle as well as speed.

Let's look at the three key steps of car detection, pedestrian detection, and motion planning in more detail. Car detection uses supervised learning. So, you've already seen how a learning algorithm can take as input pictures like these and output the detected cars. For most self-driving cars rather than using only a front-facing camera, so a camera that looks forward, also often use cameras that look to the left, to the right as well as to the back so it can detect cars not just to the front but all around it. This is usually done using not just cameras but other sensors as well such as radar and Lidar. Next is pedestrian detection, and using a pretty similar type of sensors as well as techniques, self-driving cars can detect pedestrians. Finally, I briefly mentioned a motion planning step. So, what is that?

Here's an example. Let's say you're driving your car and there's this light blue car in front of you. The motion planning software's job is to tell you what is the path, shown here in red, you should drive in order to follow the road and not have an accident. So the motion planning software's job is to output the path as well as speed at which you should drive your car in order to follow the road, and the speed should be set so that you don't run into the other car, but you also drive at a reasonable speed on this road.

Here's another example. If there's this gray car parked on the right side of the road, so you want to overtake this stopped car, then the motion planning software's job is to plot a path like that to let you veer a little bit to the left and safely overtake a stopped car. So far I've given a rather simplified description of self-driving as comprising mainly these three components.

Let's look at a bit more detail of how an actual self-driving car might work. This is a picture you've seen so far. Input image, radar, or Lidar, sensor readings into car detection and pedestrian detection, and that is then fed to motion planning to help you select your path and speed. Now in a real self-driving car, you would usually use more than just cameras, radar, and Lidar. Most self-driving cars today will also use GPS to sense its position as well as accelerometers, sometimes called an IMU, this means accelerometers, and gyroscopes as well as a map because we know that cars are more likely to be found on a road, pedestrians are more likely to be found on sidewalks, although they are sometimes found on the road as well. All this is usually additional information that's fed in to detect cars and pedestrians as well as other objects we'll talk about in a second. Rather than just detecting cars and pedestrians, in order to drive safely you also need to know where these cars and pedestrians are going in the future. So, another common component of self-driving cars is trajectory prediction, where there's another AI component that tells you not just the cars and

pedestrians you found, but also where they're likely to go in the next few seconds, so you can avoid them even as they're moving. To drive safely requires more than just navigating other cars and pedestrians. You also need to know where are the lanes so you might detect lane markings. If there's a traffic light you also need to figure out where's the traffic light, and is it showing a red, yellow, or green signal. Sometimes there are other obstacles, such as unexpected traffic cones or maybe there's a flock of geese walking in front of your car. That needs to be detected as well so that your car can avoid even other obstacles than cars and pedestrians. On a large self-driving car team, it would not be that unexpected to have a team or maybe a few people working on each of the boxes shown here in red, and it's by building all of these components and putting them together that you can build a self-driving car. As you can tell both from this rather complex example of an AI pipeline, as well as the early example of the four-step AI pipeline for the smart speaker, sometimes it takes a team to build all of these different components of a complex AI product.

What I'd like to do in the next video is share with you what are the key roles in large AI teams. If you're either a one-person or small AI team now, that's okay, but I want you to have a vision of what building a large AI team, maybe in the distant future, might look like. Let's go on to the next video.

You saw from the last two videos that some AI products may require a large AI team, maybe you have a 100 engineers or sometimes many more than a 100 to build. What I would like to do in this video is share with you the typical roles and responsibilities of a large AI team like this so you can better understand the types of work needed to build these complex AI products. Now, even if you will be working in a much smaller team, maybe a one or two or five person team for the foreseeable future, I hope this video will still be useful to you because I hope it'll give you a sense of the different types of work that an AI team might need to do even if you end up executing on this type of work with a much smaller team. One caveat, because AI is evolving so fast, the job titles and various responsibilities are not yet a 100 percent defined, and they are a little bit different across different companies. So, your company may use their job titles differently than what I'm presenting here, but I want to share a view how job titles are often used in many companies, so that if you're someday building up your own AI team, we hear about these roles you would have at least some deeper understanding of what these job titles mean. So, let's get started.

Many AI teams will have Software Engineers in them. So, for example, for the smart speaker we needed to write specialized software to execute on the joke or to set a timer or to answer questions about today's weather. So, those are traditional software engineering tasks. Or you're building a self-driving car to make sure that your self-driving car software is reliable and doesn't crash. These are software engineering tasks. So, it's not uncommon for AI teams to have enlarged fractions sometimes 50 percent, sometimes much much much more than 50 percent of Software Engineers in them.

The second common role is the Machine Learning Engineer. So, Machine Learning Engineer might write the software responsible for generating the A to B mapping or for building other machine learning algorithms needed for your product. So, they might gather the data of pictures of cars and positions of cars, train a neural network or train a deep learning algorithm and work iteratively to make sure that the learning algorithm is giving accurate outputs.

Another role that you sometimes hear about is the Machine Learning Researcher. The typical row of the Machine Learning Researcher is to extend the state of the art in machine learning. Machine learning and AI more broadly are still advancing rapidly. So, many companies for profit and non-profit, more have Machine Learning Researchers responsible for extending the state-of-the-art. Some Machine Learning Researchers will publish papers, but many companies also have Machine Learning Researchers that do research, but are less focused on publishing.

There's one other job title that's sort of in-between these two which is the Applied Machine Learning Scientists, which live somewhere between Machine Learning Engineer and Machine Learning Researcher. The Machine Learning Scientists kind of does a bit of both. They are often responsible for going to the academic literature or the research literature and finding the steady VR techniques and finding ways to adapt them to the problem they are facing such as how to take the most cutting edge, trigger where the wicker detection algorithm and adapt that to your smart speaker.

Let's look at some more of those. Today, there are a lot of Data Scientists working in industries. The role of Data Scientist is not very well defined and the meaning is still evolving today. I think one of the primary responsibilities of Data Scientists is to examine data and provide insights, as well as to make presentations to teams or the executives in order to help drive business decision-making. There also Data Scientists today doing other tasks. So, there are also Data Scientists today whose work looks more like the Machine Learning Engineers, they are described on the previous slide. The meaning of this job title is still evolving today.

With the rise of big data, there are also more and more Data Engineers whose main role is to help you organize your data, meaning, to make sure that your data is saved and is easily accessible, secure and cost-effective way. So, why is saving data such a big deal? Can't you just save them in a hard disk and be done with it. In some companies, data volumes have gotten so big. There's actually quite a lot of work to manage

them. To give you a sense of the scale, in computer science one MB stands for one megabytes and so a typical song on your music player like a typical MP3 file may be a few megabytes, say five megabytes would be a non unusual MP3 file size. A 1,000 megabytes is called a gigabyte. A million megabytes is called a terabyte and a billion megabytes is called a petabyte. With today's hard disk sizes, saving a few megabytes is not a big deal. It's just like a mere MP3 file, but storing a 1,000 megabytes, also called a gigabyte, starts to be a bit slower. A typical hour-long movie that you stream over the internet maybe above gigabytes. So, that's quite a lot of data.

To give you a sense of the scale, a self-driving car may collect multiple gigabytes of information every single minute of operations. So, it's as if every minute the self-driving car generates enough data to store multiple hour-long movies. So, self-driving cars actually generate lot of data and saving the data for many days or weeks or months or years of operation starts to require serious data engineering. A terabyte is 1,000 times bigger than that and a petabyte is yet another 1,000 times bigger than that of that teams that were responsible for saving several petabytes of information per day, but other than pretty large internet companies is not that common for a company to generate multiple petabytes of information per day. As you move down this scale to bigger and bigger datasets, it becomes harder and harder to make sure that data is stored in a easy accessible, secure and cost-effective way, which is why Data Engineers become more and more important.

Finally, you'll also hear some people referred to AI Product Managers whose job is to help decide what to build. In other words, they help to figure out what's feasible and valuable. Traditional Product Manager's job was already to decide what to build as well as sometimes some other roles, but the AI Product Manager now has to do this in the AI era and they're needing new skill sets to figure out what's feasible and valuable in light of what AI can and cannot do today. Because the field of AI is still evolving, none of these job titles are completely nailed down in the stone and different companies will use these job titles somewhat differently. But I hope this gives you a sense of some of the different types of work needed to build very complex AI products as well as where some of the job titles are evolving.

To finish though, I want to re-emphasize that you can get started with a small team. You don't need a 100 people to do most AI projects. So, whether you just have one Software Engineer working with you, or just a single Machine Learning Engineer, or just a single Data Scientists, or maybe nobody, but yourself, if either you or an engineer working with you has taken some online courses on machine learning or deep learning or data science, that's often enough for you by yourself or for you and an engineer to start looking at some smaller volumes of data, start drawing some conclusions or start trading some machine learning models to get going. So, even though I've tried to paint here a vision for what a large AI team might look like, even if you have only a small AI team, could be nobody by yourself, I would still encourage you to get started and start exploring what the projects you could do.

In this video, you saw what an AI team might look like, but when you look at a bigger company, an AI team doesn't live in isolation. So, how does an AI team fit into a bigger company to help the whole company become good at AI? You might remember that in week one I briefly alluded to an AI transformation playbook, which is a roadmap for you to help a company, to help maybe a great company become great at AI. Now that you've learned what is AI, how to do AI projects and even what AI teams in companies and the competency AI projects and coms may look like, let's return to their AI transmission playbook and go much deeper into the individual steps of the playbook so that you can understand what it takes to help a company over maybe a small number of years become good at AI and hopefully become much more valuable and much more effective along the way. Let's go into the AI transmission playbook in the next video.

How can you help your company become good at AI? Based on my experience, starting leading the Google Brain Team as well as Baidu's AI group which were respectively the leading forces for helping Google and Baidu become good and deeper in AI. I've spent a lot of time thinking about what it takes to help a great company become a great AI company and I wound up writing an AI transformation playbook to help other countries on this journey.

In this video, I'd like to share with you the details of the AI transformation playbook so that you can better understand what it might take for your company to become good at AI. In case some this seems like things that only CEO's need to know, I think that that's not the case, it's useful for everyone in the company to understand whether it might take for your work to impact not just a few projects but maybe have a bigger impact on the company as a whole.

Let's get started. Here are the five steps of the AI transformation playbook. We'll dive into greater detail in a little bit but briefly,

step one is for your company to execute pilot projects to gain momentum. Start to know what it feels like to work on AI projects. Step two, is to build an in-house AI team. Step three, is to provide broad AI training, not just the engineers but to many levels within a company including executives. Step four, is to develop your AI strategy and step five, is to develop internal and external communications about your company and AI.

The way your company will execute the steps may not be totally sequential and so the different steps may overlap. But this numbering gives a maybe rough sense of the order in which I think you could do these steps.

In this video, we will go in greater depth on the first three of these three steps and the next video we'll cover steps four and five.

Let's start with step one, executing pilot projects to gain momentum. If you want your company to gain momentum with AI, the most important consideration for the initial project or projects, is for them to be successful rather than necessarily be the most valuable. For example, when I was leading the Google Brain Team, there was still a lot of skepticism at that time about deep learning. So, my first internal customer was Google's speech recognition team and speech recognition is nice to have this is useful but it's actually not the most important or valuable project for the company's bottom lines. It's not as valuable as, for example web search or online appetizing. But by having my team make the Google Speech team more successful, it started the fly wheel turning helps it get momentum because then the peers and the other teams that are brothers and sisters the speech team started to see my team make the speech team more successful and so they also started to gain favor in AI and wanted to work with us.

So, my second internal customer was the Google Maps Team to improving the quality of the maps data using deep learning and with the first second successes. I then started other conversations with the online advertising team for example. So, when selecting your initial project, try to pick something that you think has a good chance of success. They can start to fly wheel turning even if it may not be the most valuable project that you could eventually do for the company. Because the goal of the first few projects is just to gain momentum is also now you see if you can pick something that can show traction within six to 12 months. So you can start to fly wheel turning quickly. Finally, your first one or two pilot projects can be either in-house or outsource. If you do not yet have a large in-house AI team, might be possible, might even be advisable to outsource some or all of your first couple of AI projects in order to get more expertise in house and to let you start building that momentum faster. Now, beyond a certain point, you will need your own in-house AI team to execute a long-term sequence of maybe many dozens of AI projects. So, step two is to build an in-house AI team. A lot of companies are organized like this, where, there's a CEO and multiple business units which I'm going to abbreviate with BU that reports up to the CEO. So, what I

recommend for most companies is to build a centralized AI team and then to take the talent in the matrix organization and two matrix them into these different business units to support their work.

Why essentialize the AI? Let's take an example. Maybe this unit is your gift card business unit, and the BU leader may be great at whatever he or she does. They may be greater than gift card business. But unless he or she is knowledgeable AI and knows how to build retain and managing AI team, it may be very difficult for that business unit leader to hire and retain an appropriately manage their own AI talent. So, in that case, I think you also successfully much higher if you find that AI team leader that can be responsible for consistent company wide standards for recruiting, retention. Have essentialized AI team to give the team the community to talk to each other about how AI applies to your business radical. It may be more efficient to take the AI talent in your centralized AI unit and matrix them into the gift card business unit so that your AI talent can work together with the gift card domain experts in order to develop interesting AI projects together.

One other responsibility for the AI unit, is to build a company wide platforms. If there are software platforms or other tools or data infrastructure that could be useful for the whole company, then a single business unit may not have the resources or the incentive to build these company-wide platforms and resources that can support the whole company but essentialized AI team maybe help built these company wide tools or platforms that can help multiple business units.

Finally, this new AI business unit can be under the CTO, the CIO, the chief data officer or the Chief Digital Officer or it could also be under a new chief AI officer. The CAIO chief AI officer is a role that I'm seeing more and more often in different companies but if some other senior executive has the right skill set, they could also manage the AI unit.

Finally, one last recommendation, which is that, I think it is hopeful if to get the AI units started the company or the CEO provides funding to build at the AI unit rather than required AI unit to get funding from the business units. Eventually, after the initial investment and after the initial ramp up, the AI unit will have to show his value that is creating for the business units but having CEO inject funding at the outset so they can get going, will often help you get that initial momentum much faster.

In addition to building an in-house AI team, I also recommend that you provide broad AI training. Now, for accompany that become good at AI, is not just that you need engineers to know AI, you need multiple people at multiple levels of the company to understand how AI interacts with their roles. For example, for executives and senior business leaders, I recommend that they learn what AI can do for your enterprise, that they learned the basics of these of AI strategy and they learned enough about AI to make resource allocation decisions. So, how much training should executives senior business leaders receive? I think that numbers of hours of training is not a very good way to measure training but with that caveat, I think you can deliver a lot of this training with maybe four hours or so of training. Leaders of divisions work on AI projects also need to know how to interact in their role with AI. I think these leaders would need to understand how to set project directions. So how the conduct technical and business diligence how to make resource allocation decisions at the division level as well as how to track and monitor progress of AI projects. So, this have a training I think would take at least 12 hours. Although, again number of hours is not a great metric for tracking how much are they earning.

Finally, many companies are hiring AI talent from outside but I will also not underestimate the importance and the impact of training of your existing engineering workforce with AI skills for a software engineer to become proficient at AI does take a while so plan for maybe at least a 100 hours of training. But I'm seeing many companies provide training to help engineers learn to build and ship AI software to gather and managed data as was helped them become effective at executing on specific AI projects. The world today does not have nearly enough AI engineers and so in-house training is a key part of many companies building up of their in-house AI capabilities.

Finally, how do you get all this training done? Thanks to the rise of online digital content, ranging from of course, online courses to also books and YouTube videos and blog posts. There is a lot of great content online about all of these subjects and I think a good CLO should work of experts to curate this type of content and motivated teams to complete these learning activities rather than necessarily create contents which is much more expensive thing to do. So steps one to three of the AI transmission favor.

Okay, I hope that your company will be able to start to execute an initial projects, build the team, provide the training and really start to get a lot of momentum going in terms of helping you accompany you become more valuable or more effective using AI. Looking at the broader picture, AI also affects company strategy and how you align different stakeholders including investors, employees, customers with this transformation as a company.

Let's go on to the next video to talk about AI strategy.

In the last video, you learned how to execute pilot projects to gain momentum for the in-house AI team and provide broad AI training. But you want your business, not just gain momentum in the short-term using AI, but in the long term be a very valuable and maybe even defensible business. What can you do?

Let's talk about AI strategy as well as perhaps important for some companies, internal and external communications relative to AI. To recap, this is the five-step AI transformation playbook, and in this video, we'll dive more deeply into these final two steps.

Step four of the AI transmission playbook is to develop an AI strategy, which, I hope for you may mean to leverage AI to create an advantage specific to your industry sector. One unusual part of this playbook is that developing the AI strategy is step four not step one. When I shared this with many CEOs consistent request of please feedback ago was, can you please put the strategy as step one? Because I want to figure out what is my company strategy, then I want to find the resources, and then execute on the strategy. But I've found that companies that tried to define the strategy as step one, before getting your feet wet, before trying out AI knowing what a feasibility AI project. Companies like that tend to end up with sometimes very academic strategies that are sometimes not true to life. So, for example, I've seen some CEOs copy and paste newspaper headlines into this strategy.

We read that data is important, he say, "My strategy is to focus on collecting a lot of data, but for your company, that data may or may not be valuable, and may or may not be a good strategy for your company. So, I tend to recommend to companies to start the other steps first, execute the pilot projects. Start building a little bit of a team. Start providing some training, so that only after you understand AI and understand how it may apply to your business, that you then formulate your strategy. I think this will work much better for your company than if you tried to formulate an AI strategy, before your company including specifically the executive team has some slightly deeper understanding of what AI can and cannot do for your industry sector. In addition, you might consider designing a strategy that is aligned with the virtuous cycle of AI. Let me illustrate that with an example from web search.

One of the reasons that web search is a very defensible business, meaning is very difficult for new entrants to compete with the incumbents with the existing large web search engines, is this: If a company has a better product, maybe a slightly better product, then that web search engine can acquire more users. Having more users means that you collect more data because you get to observe what different users click on when they search for different terms, and that data can be fed into an AI engine to produce an even better product. So, this means that the company with somewhat better product, ends up with even more users, ends up with even more data, and does an even better product with this link being created by modern AI technology. It makes it very difficult for a new entrant to break into this self-reinforcing positive feedback loop, called the virtuous cycle of AI.

Fortunately though, this virtuous cycle of AI can be used by smaller teams entering new verticals as well. So, I think today is very difficult to build a new web search engine to compete with Google, or Baidu, or Bing, or Yandex. But if you are entering a new vertical, a new application area where there isn't a entrenched incumbent, then you might really develop a strategy that lets you be the one to take advantage of this virtuous cycle.

Let me illustrate with an example. There is a company called Blue River that was acquired by John Deere for over US\$300 million, and Blue River makes agricultural technology using AI. So, what they did was build these machines that would be towed behind a tractor, in a big agricultural fields. This machine would take pictures of crops and figure out which is a crop and which is a weed, and use precision AI to kill off just the weeds, but not the crop. So, I knew some of the founders of Blue River while they were Stanford students are taking my class. So, to get the project started, they actually just use strap in his and sweat, they use their personal cameras and went out to a bunch of farms, and took a lot of pictures of crops in these agricultural fields. So, they started to collect pictures of heads of cabbage and weeds around the cabbage.

Once they had enough data, starts off with a small data set, they could train a basic product. The first product, frankly wasn't that great. It was trained on a small data set, but it worked well enough to start to convince some farmers, some users to start to use their product, to tow this machine behind the tractor, in order to start killing weeds for the farmers. Once this thing was running around the farms through the process of taking pictures of heads of cabbage and killing off weeds, they naturally acquired more and more data. Over the next few years, what they did was they were able to enter this positive feedback loop, where having more data allows you to have a better product. Having a better product allows you to convince more farmers to use it. Having farmers use it allows you to collect more data. Over several years, entering a virtuous cycle [inaudible] , can allow you to collect a huge data asset that then makes your business quite defensible.

In fact, at the time of acquisition, I'm pretty sure that they had a much bigger data asset of pictures of heads of cabbage lying on a field than even the large tech companies had, and does actually makes the business relatively defensible from even the large tech companies that have another web search data, but do not have nearly as many pictures as this company does of heads of cabbage lying in the agricultural fields.

One more piece of advice. A lot of people think that some of the large tech companies are great at AI, and I think that's true. Some of the largest tech companies are very good at AI, but this doesn't mean you need to or should try to compete with these large tech companies on AI in general because lot of AI needs to be specialized or verticalized for your industry sector. So, for most companies to be in your best interest to build AI specialized for your industry, and to do great work in AI for your application areas, rather than try to compete or feel like you need to compete left and right with the large tech companies on AI over the place which just isn't true for most companies. Other elements of an AI strategy. We are going to live in an AI power world and the right strategy can help your company navigate these changes much more effectively.

You should also consider creating a data strategy. Leading AI companies are very good at strategic data acquisition. For example, some of the large consumer facing AI companies will launch services, like a free email service, or a free photo-sharing service, or many other free services that do not monetize, but allows them to collect data in all sorts of ways that lets them learn more about you, so they can serve you more rather than adds, and thereby monetize their data in a way that is quite different than direct monetization about that product. The way you acquire data is very different depending on your industry vertical, but I have been involved in what feels like these multi-year chess games, where other corporate competitors and I are playing multi-year games to see who can acquire the most strategic data assets. You might also consider building a unified data warehouse. If you have 50 different data warehouses under the control of 50 different vice presidents, then is almost impossible for an AI engineer or for a piece of AI software to pull together all of this data in order to connect the dots.

For example, if the data warehouse for manufacturing is in a totally different place than the data warehouse for customer complaints, then how can an AI engineer pull together this data to figure out, whether the things that might happen in manufacturing, that causes you to ship a faulty cell phone, that causes a customer to complain two months later. So, a lot of leading AI companies have put a lot of upfront effort into pulling the data into a single data warehouse because this increases the odds that a engineer or a piece of software, can connect the dots and spot the patterns between how a elevated temperature in manufacturing today may result in a faulty device that leads to a customer complaint two months in the future, thus letting you go back to improve your manufacturing processes. There are many examples of this in multiple industries. You can also use AI to create network effects and platform advantages. In industries with winner take all dynamics, AI can be a huge accelerator.

For example, take the ride-sharing or the ride-hailing business. Today, companies like Uber, and Lyft, and Ola, and DiDi, and Grab seemed like they have relatively defensible businesses because they are platforms that connect drivers with passengers, and is quite difficult for a new entrant to accumulate both a large rider audience and a large passenger audience at the same time. Social media platforms like Twitter and Facebook are also very defensible because they are very strong network effects where having a lot of people on one platform makes that platform more attractive to other people. So, it's very difficult for a new entrant to break in. If you are working in a business with these types of winner take all dynamics or winner take most dynamics, then if AI can be used to help you we're growing faster. For example, with a celebrating user acquisition, then that can pass translates into a much bigger chance that your company will be the one to succeed in this business vertical. Strategy is very comfy and industry and situation specific. So, it's hard to give strategy advisers completely general to every single company. But I hope that these principles give you a framework for thinking about what might be some key elements of an AI strategy for your company. Now, AI can also fit into more traditional strategy frameworks. For example, Michael Porter, many years ago have written about low cost and high value strategies. If your company has a low-cost strategy, then perhaps AI can be used to reduce costs for your business or, if your company has a high value strategy to deliver really, really valuable products with a higher cost, then you might use AI to focus on increasing the value of your products. So, AI capabilities can also help argument existing elements of a broader corporate strategy. Lastly, as you're building these valuable and defensible businesses, I hope that you also build only businesses that make people better off. AI is a superpower. This is a very powerful thing that you can do to build a great AI company, and so I hope that whatever you do, you do this only in ways that make humanity better off. The final step of the AI transmission playbook is to develop internal and external communications. AI can change a company and its products, and its important to communicate appropriately with the relevant stakeholders about this. For example, this may include investor relations to make sure that your investors can value your company appropriately as an AI company. Investor relations may also includes government relations. For example, AI is entering health care, which is a highly regulated industry because government has a legitimate need to protect patients, and so for AI to affect these highly regulated industries, I think is important for companies to communicate with government, and to work collaboratively with them in public-private partnerships to make sure that AI solutions bring people the benefits it can, while also making sure that governments can protect consumers and protect patients. So, this would be true for health care or be true for self-driving cars, it would be true for finance and many other AI industry verticals. If your products change, then consumer or user education will be important. AI talent is very scarce into this world and so, if you are able to showcase some of your initial successes that could really help with talent and recruiting. Finally, internal communications is also important if you're making a shift in your company, then many people internally may have worries, some legitimate and some less rational about AI and internal communications, so reassure people where appropriate can only be helpful. With these five steps, I hope it gives you a vision for how you might be the hope a company become good at AI. If you're interested in reading the detailed AI transmission playbook, you can also download it from this landing AI website. I hope you enjoyed these two videos on the AI transmission playbook. I've seen companies become much more value and much more effective by embracing and become good at AI, and I hope these ideas they hope you take a first step toward helping your company good at AI. Having said that, I have also seen many common pitfalls, the companies run into when trying to implement AI across the enterprise.

Let's take a look at some of these common pitfalls in the next video so that hopefully, you can avoid them. Lets go on to the next video.

I hope you'll be able to use AI to build exciting and valuable projects either for yourself or for your company and make life better both for yourself and for others. Along the way, I hope you also manage to avoid some of the pitfalls I've seen some AI teams fall into.

Let's go over a five don'ts and does for if you're trying to build AI for your company.

First one, don't expect AI to solve everything. You already know that AI can do a lot but there's also lots AI cannot do. Instead, you should be realistic about what AI can or cannot do, given the limitations of technology, data, and engineering resources. That's why I think technical diligence in addition to business diligence is important for selecting feasible and valuable AI projects.

Second, don't just hire two or three machine learning engineers and count solely on them to come up with use cases for your company. Machine learning engineers are a scarce resource but you should instead air the engineer talents with business talent and work cross-functionally to find feasible and valuable projects. Is often the combination of the machine-learning talents worked to business talent that can select the most valuable and feasible projects.

Third, don't expect AI project to work the first time. As you've already seen, AI development is often an iterative process so should plan for it through an iterative process with multiple attempts needed to succeed.

Fourth, don't expect traditional planning processes to apply without changes. Instead, you should work with the AI team to establish timeline estimates, milestones, KPIs or metrics that do make sense. The types of timeline estimates, milestones, and KPIs or metrics associated with AI projects are a bit different than the same things associated with non AI projects. So, hopefully working with some individuals knowledge about AI can help you come up with better ways of planning AI projects.

Finally, don't think you need superstar AI engineers before you can do anything. Instead, keep building the team and get going with a team you have realizing that there are many AI engineers in the world today including many that have learned primarily from online courses. They can do a great job building valuable and feasible projects.

If you can avoid these AI pitfalls, you already be ahead of the game compared to many other companies. The important thing is to get started. You're second AI project would be better than your first. Your third AI project would better than your second. So, the important thing is to get started and to attempt your first AI project.

In the final video for this week, I want to share with you some concrete first steps you can take in AI. Let's go on to the next video.

This week, you saw some examples of what it's like to build a complex AI product, like a smart speaker, or self-driving car. You also learned about the roles and responsibilities of large AI teams, and maybe what it's like to build a large AI team, and saw the AI transformation playbook for helping a great company become a great AI company. In case some of this seems daunting, because some of these will take maybe two or three years to execute. In case any of these seems daunting, it's okay because the more important thing is that you're able to take the first step.

In fact, by taking this course, you've already taken a great first step. So, I hope that after this course you will take also an equally good second step. So, in this video, I want to share with you some concrete suggestions for the next step you can take toward AI for yourself or for your company. Here are some initial steps I would urge you to take. Rather than going it alone, consider having friends in your company or personal friends outside work to learn about AI with you. This could mean, asking them to take this course with you or after you, or starting a reading group to read some books or other materials about AI. With what you've learned in this course, you will also, especially if you have engineering friends, be able to start brainstorming projects. No project is too small and it's better to start small and succeed, than to start too big and not succeed. Many projects can be done just by you or perhaps by you and a friend. If you and or a friend take an online course on machine learning, that's enough know-how for you to get started on many potentially very valuable AI projects. In a company, you might also hire a few machine learning or data science people to help out, in addition to providing in-house training to develop the in-house talent. When you're ready to go bigger, you might also try to have your company hire or appoint an AI leader, such as a VP of AI or a Chief AI Officer. But maybe you don't need a very senior AI leader before hiring just a few machine learning or data scientists people to get going more quickly. Finally, I've spoken with a lot of CEOs and Boards about AI transformations. If you want your company to become greater AI, you would also consider trying to discuss with your CEO, the possibility of trying to execute an AI transformation. I think the key question to ask your CEO or your Board is, will your company be much more valuable and/or much more effective if it was great at AI. And if you and they think the answer is yes, that might be a good reason for the company to try to execute an AI transformation. The different items on this list have varying levels of difficulty to execute, but I hope you will start with what you can and then grow your AI efforts from there. I see many people, some technical many non-technical, help their companies learn about AI and start to use it effectively. After these videos, you now have the concrete tools to do the same. So, I hope you'll take advantage of them and help your company, help yourself, and help others as well. Finally, we have two more optional videos for this week on a survey of major AI application areas, as well as major AI techniques.

So, if you've ever wondered, what do the terms Computer Vision and Natural Language Processing mean? Or, what is Reinforcement Learning? Or, what is Unsupervised Learning? Please take a look at these videos, since we'll teach you these application areas, as well as technologies in the next two videos. We made these videos optional because they are a little bit more technical, but after watching them, you will be able to better communicate with AI engineers. So, I hope you take a look. Either way, thank you for watching all of these videos this week, and look forward to seeing you in next week's videos.

AI today is being successfully applied to image and video data, to language data, to speech data, to many other areas.

In this video, you see a survey of AI applied to these different application areas and I hope that this may spark off some ideas of how you might be able to use these techniques someday for your own projects as well. Let's take a look.

One of the major successes of deep learning has been Computer Vision. Let's take a look at some examples of computer vision applications. Image classification and object recognition refer to taking as input a picture like that and telling us what is in this picture. In this case, it'd be a cat. Rather than just recognizing cats, I've seen AI algorithms able to recognize specific types of flowers, AI able to recognize specific types of food and the ability to take as input a picture and classify it into what type of object, and this is being used in all applications.

One specific type of image classification that has had a lot of traction is face recognition. This is how face recognition systems today work. A user might register one or more pictures of their face to show the AI what they look like. Given a new image, the AI system can then say is this the same person? Is this you? Or is this a different person so that it can decide a decision, unlock the door or unlock the cell phone, unlocked the laptop or something else based on the identity of the person.

Of course I hope face recognition will only be used in ways that respect individuals privacy, we'll talk more about AI in society next week as well. A different type of computer vision algorithm is called object detection. So, rather than just tried to classify or recognize an object, you're trying to detect if the object appears. For example, in building a self-driving car, we've seen how an AI system can take as input a picture like this and not just tell us yes or no, is there a car. Yes or no, is there pedestrian but actually tells us the position of the cars as well as the positions of the pedestrians in this image, and object detection algorithm can also take as input a picture like that and just say, no I'm not finding any cars or any pedestrians in that image. So rather than taking a picture and labeling the whole image which is image classification, instead an object detection algorithm will take us input an image and tell us where in the picture different objects are as was what are the types of those objects.

Image segmentation takes this one step further. Given an image like this, an image segmentation algorithm we output, where it tells us not just where the cars and pedestrians but tells us for every single pixel, is this pixel part of this car or is this pixel part of a pedestrian. So it doesn't just draw rectangles around the objects and detects, instead it draws very precise boundaries around the objects that it finds. So, in reading x-rays for example, it would be an image segmentation algorithm that could look at an x-ray scan or some other image of a human body and carefully segment out, where's the liver or where's the heart or where is the bone in this image.

Computer vision can also deal with video and one application of that is tracking. In this example, rather than just detecting the runners in this video, it is also tracking in a video whether runners are moving over time. So, those little tails below the red boxes show how the algorithm is tracking different people running across several seconds in the video. So, the ability to track people and cars and maybe other moving objects in a video helps a computer figure out where things are going.

If you're using a video camera to track wildlife for example, say birds flying around, a tracking algorithm will also be the helper track individual birds flying across the frames of your video. These are some of the major areas of computer vision and perhaps some of them will be useful for your projects.

AI and deep learning specifically is also making a lot of progress in Natural Language Processing. Natural Language Processing or NLP refers to AI understanding natural language, meaning the language that you and I might use to communicate with each other.

One example is text classification where the job of the AI is to input a piece of texts such as an email and tell us what is the cause or what is the category of this email, such as a spam or non-spam email. There are also websites that would inputs a product description. For example, you might write, I have a secondhand cellphone for sale and automatically figure out what is the product category in which the list is product. So, that would go under cellphones or electronics or if you write, I have a new t-shirt to sell then it would list it automatically under clothing.

One type of text classification that has had a lot of attention is sentiment recognition. For example, a sentiment recognition algorithm can take as input a review like this of a restaurant, the food was good and automatically tries to tell us how many stars this review might get. The food was good as a pretty good review maybe that's four over five-star review. Whereas if someone writes service was horrible, then the sentiment recognition algorithm should be able to tell us that this corresponds maybe to a one-star review.

A second type of NLP or Natural Language Processing is information retrieval. Web search is perhaps the best known example of information retrieval where you type in the text query and you want the AI to help you find relevant documents. Many corporations will also have internal information retrieval systems where you might have an interface to help you search just within your company's set of documents for something relevant to a query that you might enter.

Name entity recognition is another natural language processing technology. Let's illustrate it with an example. Say you have this sentence and you want to find all the people names in the sentence. So, Queen Elizabeth the second is a person, Sir Paul McCartney as a person. So, the sentence Queen Elizabeth, the second night to Paul McCartney for a service of music at the Buckingham Palace, it would be a name entity recognition system to confine all the people's names in the sentence like this. If you want to find all the location names, all the place names in a sentence like that, a named entity recognition system can also do so. Name entity recognition systems can also automatically extract names of companies, phone numbers, names of countries, and so, if you have a large document collection and you want to find automatically all the company names, or all the company names the occur together or all the people's names, then a name entity recognition system would be the tool you could use to do that.

Another major AI application area is machine translation. So, for example, if you see this sentence in Japanese, AI [inaudible]. Then hopefully a machine translation system can input that and output the translation AI is in the electricity.

The four items on this slide:

text classification, information retrieval, name entity recognition, and machine translation, are four major categories of useful NLP applications.

If you work with an NLP team you may also hear them talk about parsing and part of speech tagging technologies. Let me tell you what these are. Let's take the example sentence, "The cat on the mat". A part-of-speech tagging algorithm will go through all the words and tell you which of these words are nouns, which of these words are verbs, and so on. For example, in the English language cat and mat in the sentence are nouns. So, the part of speech tagger we'll label these two words as nouns. According to the theory of English language, the word the is a determiner. Don't worry if you've never heard of a determiner before, this is a word from the theory of English language, and the word on is a preposition. So, part of speech tagger will label these words like that.

Well, why do you care? If you're building a sentence classifier for restaurant reviews, then a part-of-speech tagging algorithm would be able to tell you which are the nouns, which are the verbs, which are the adjectives, which are the adverbs, and so on, and therefore, help your AI system figure out which of the words to pay more attention to. For example, you should probably pay more attention to the nouns since

those seem like important words. Maybe the verbs. Certainly the adjectives, words like good, bad, delicious are adjectives, and your AI system may learn to ignore the determiners. Words like the which may matter less in terms of how a user is actually feeling about the restaurant. A part of speech tagging system is usually not a final application. You hardly ever wake up in the morning and think, "Boy, I wish I could get all the words in my sentence tag." There's often an important pre-processing step. There's often an important intermediate step in a longer AI pipeline, where the first step is part-of-speech tagging, or parsing, which we'll talk about in a second, and then the later steps are an application like sentence classification, or machine translation, or web search.

Now, what is a parser? Given these five words, a parser helps group the words together into phrases. For example, the cat is a phrase, and the mat is a phrase. So, a parser will draw these lines on top of the words to say, those words go together. On the mat is another phrase.

Finally, the two phrases, the cat, as well as on the mat, these two phrases are then combined to form the overall sentence. So, this thing that I drew on top with the sentence tells you what words go with what words, and how the different words relate to each other. While a parsing algorithm is also another final end-user product, it's often a commonly used step to help other AI algorithms. That's how classify tags are translated, and so on.

Modern AI, specifically deep learning has also completely transformed how software processes audio data such as speech. How is speech represented in a computer? This is an audio waveform of one of my friends saying the phrase machine learning. The x-axis here is time, and the vertical axis is what the microphone is recording. What the microphone is recording is little variations, very rapid variations in air pressure which your ear and your brain then interpret as sound. This plot shows as a function of time, the horizontal axis, how the air pressure is changing very rapidly in response to someone say the word machine learning. The problem of speech recognition, also known as speech-to-text, is the problem of taking as inputs a plot like this, and figuring out what were the words that someone said. A lot of speech recognition's recent progress has been due to deep learning.

One particular type of speech recognition is trigger word detection or wakeword detection. You saw this in the earlier video with having an AI system detect a trigger word or the wakeword such as Alexa, or Hey Google, or Hey device. Speaker ID is a specialized speech problem where the task is to listen to someone speak and figure out the identity of the speaker. Just as face recognition helps verify your identity by taking a picture, speaker ID can also help verify your identity by listening to you speak.

Finally, speech synthesis, also called text-to-speech or TTS is also getting a lot of traction. Text-to-speech is a problem of inputting a sentence written in text and turning that into an audio file. Interestingly, whereas, text-to-speech is often abbreviated TTS, I don't often see speech-to-text abbreviated STT. One quick example.

Let's take the sentence, "The quick brown fox jumps over the lazy dog." This is a fun sentence that you often see NLP people use because this sentence contains every single letter from A to Z. So, that's ABC all the way up to X, Y, and Z. You can check all 26 letters appear in this sentence. Some letters appear more than once. If you parse this sentence into a TTS system, then you might get an audio upwards like this, The quick brown fox jumps over the lazy dog. Modern TTS systems are increasingly sounding more and more natural and human-like.

AI is also applied to many applications in robotics and you've already seen one example in the self-driving car. In robotics, the term perception means figuring out what's in the world around you based on the senses you have, be it cameras, or radar, or lidar. Shown on the right is the 3D laser scan or the lidar scan of a self-driving car as well as the vehicles that this self-driving car the middle has detected in the vicinity of your car. Motion planning refers to finding a path for your robot to follow. So, if your car wants to make a

left turn, the motion planner might plan a path as well as a speed for the car to make a left turn that way. Finally, control refers to sending commands to the motors such as your steering wheel motor, as well as your gas pedal, and brake motors in order to make the car smoothly follow the path that you want.

On this slide, I'll focus on the software and the AI aspects of robotics. Of course, there's also a lot of important work being done to build hardware for robotics as well. But a lot of the work AI on perception, motion planning, and control has focused on the software rather than the hardware of robotics. In addition to these major application areas, machine learning is also very broadly used.

The examples you've seen in this video relate mainly to unstructured data such as images, audio, and text. Machine learning is applied at least as much to structured data, and that means these tables of data some of which you saw in the earlier videos. But because unstructured data such as images is so easy for humans to understand, there's something very universal, very easy for any person to understand and empathize with when we talk about an AI system that recognizes a cat. So, the popular press tends to cover AI progress on unstructured data much more than it does AI on structured data. Structured data also tends to be more specific to a single company. So, it's harder for people to write about or understand, but AI on structured data, or machine learning on structured data is creating tremendous economic value today as well as AI on unstructured data. I hope this survey of AI application areas gives you a sense that the wide range of data that AI is successfully applied to today, and maybe this even inspire you to think of how some of these application areas may be useful for your own projects.

Now, so far the one AI technique we've spent the most time talking about is supervised learning. That means learning inputs, output, or A to B mappings from labeled data where you give the AI system both A and B. But that's not the only AI technique out there. In fact, the term supervised learning almost invites the question of what is unsupervised learning, or you might also have heard from media articles, from the news about reinforcement learning. So, what are all these other techniques?

In the next video, the final optional video for this week, we'll do a survey of AI techniques, and I hope that through that maybe you'll see if some of these other AI techniques and supervised learning could be useful for your projects as well. Let's go on to the final optional video for the week.

There are a lot of AI and machine learning techniques today. And while supervised learning, that is learning A to B mappings, is the most valuable one, at least economically today, there are many other techniques that are worth knowing about. Let's take a look.

The best known example of unsupervised learning is clustering, here's an example. Let's say you run a grocery store that specializes in selling potato chips. And you collect data on different customers, and keep track of how many different packets of potato chips a single customer buys, as well as what's the average price per package that person paid for their potato chips. So you sell some low end, cheaper potato chips, as well as some high end, more expensive packets of potato chips. And different people may buy different numbers of potato chip packets, in a typical trip to your grocery store. Given data like this, a clustering algorithm will say that it looks like you have two clusters in your data. Some of your customers tend to buy relatively inexpensive potato chips, but buy a lot of packets. If your grocery store is near a college campus, for example, you may find a lot of college students that are buying cheaper potato chips packets, but they sure buy a lot of them. And there's a second cluster in this data of a different group of shoppers that buy fewer packets of potato chips, but buy more expensive packets. A clustering algorithm looks at data like this, and automatically groups the data into two clusters or more clusters, and is commonly used for market segmentation. And will help you discover things like that if you have a college student audience that buys a certain type of potato chips, and a working professional audience that buys fewer potato chips but is willing to pay more. And this can help you market differently to these market segments.

The reason this is called unsupervised learning is the following. Whereas supervised learning algorithms run an A to B mapping, and you have to tell the algorithm what is the output B that you want, an unsupervised learning algorithm doesn't tell the AI system exactly what it wants. Instead it gives the AI system a bunch of data such as this customer data, and it tells the AI to find something interesting in the data, find something meaningful in the data. And in this case, a clustering algorithm doesn't know in advance that there's a college student demographic and a working professional demographic. Instead, it just tries to find what are the different market segments without being told in advance what they are. So unsupervised learning algorithms, given data without any specific design output labels, without the target label B, can automatically find something interesting about the data.

One example of unsupervised learning that I have worked on was the slightly infamous Google cat. In this project, I had my team run a unsupervised learning algorithm on a very large set of YouTube videos, and we asked the algorithm, tell us what you find in YouTube videos. And one of the many things that are found in YouTube videos was cats, because somewhat stereotypically, YouTube apparently has a lot of cat videos. But it was a remarkable result that without telling it in advance that it should find cats, the AI system, the unsupervised learning algorithm, was able to discover the concept of a cat all by itself. Just by watching a lot of YouTube videos and discovering that, boy, there are a lot of cats in YouTube videos. It's hard to visualize exactly what an AI algorithm is thinking sometimes, but this picture on the right is a visualization of the concept of the cat that the system had learned. Even though supervised learning is an incredibly valuable and powerful technique,

one of the criticisms of supervised learning is that it just needs so much labeled data. For example, if you're trying to use supervised learning to get the AI system to recognize coffee mugs, then you may give it 1000 pictures of coffee mug, or 10,000 pictures of mug coffee mug. And that's just a lot of picture of coffee mugs coffee mug we will be giving our AI systems. For those of you that are parents, I can almost guarantee to you that no parent on this planet, no matter how loving and caring, has ever pointed out 10,000 unique coffee mugs to their children, to try to teach the children what is a coffee mug. So, AI systems today require much more labeled data to learn than when a human child or than would most animals. Which is why AI researchers hold a lot of hope out for unsupervised learning as way, maybe in the future, for AI to learn much more effectively in a more human like way, and more biological like way for much less labeled data. Now, we have pretty much no idea how the biological brain works, and so to realize this vision, we'll take

major breakthroughs in AI than none of us know yet today how to realize. But many of us hold a lot of hope for the future of unsupervised learning. Having said that, unsupervised learning is valuable today. There are some specific applications and natural language processing, for example, where unsupervised learning helps the quality of web search quite a bit, for example. But the value today of unsupervised learning is so a lot smaller than the value created through supervised learning.

Another important AI technique is transfer learning. Let's look an example. Let's say you bought a self driving car, and you've trained your AI system to detect cars. But you didn't deploy your vehicle to a new city and somehow this new city has a lot of golf carts travelling round, and so you need to also build a golf cart detection system. You may have your car detection system with a lot of images, say 100,000 images, but in this new city where you just start operating, you may have a much smaller number of images of golf carts. Transfer learning is the technology

that lets you from a task A, such as car detection, and

use the knowledge to help you on a different task B,

such as golf cart detection. Where transfer learning really shine

is if having learn from a very large dataset of car detection, task A, you can

now do pretty well on golf cart detection, even though you have a much

smaller golf cart dataset. Because some of the knowledge it

has learned from the first task, of what the vehicles look like, what the

wheels look like, how the vehicles move. Maybe that will be useful also for

golf cart detection. Transfer learning doesn't

get a lot of press, but it is one of the very

valuable techniques in AI today. And for example, many computer vision

systems are built using transfer learning, and this makes a big difference

to their performance. You may also have heard of a technique

called reinforcement learning. So, what is reinforcement learning? Let me illustrate with another example.

This is a picture of the Stanford

autonomous helicopter. So it's instrumented with GPS,

accelerometers, and a compass, so it always knows where it is. And let's say you want to write

a program to make it fly by itself. It's hard to used supervised learning

input/output, A to B mappings, because it's very difficult to

specify what is the optimal way. What is the best way to fly the helicopter

when it is in a certain, given position. Reinforcement learning

offers a different solution. I think of reinforcement learning

as similar to how you might train a pet dog to behave. My family, when I was growing up,

had a pet dog. So, how do you train

a dog to behave itself? Well, we let the dog do whatever it wanted to do, and then whenever it behaved well we would praise it. You go, good dog, and whenever it does something bad you would go, bad dog. And overtime it learns to do more of the, good dog, things, and fear of the bad dog things. Reinforcement learning takes the same principle and applies it to a helicopter or two other things. So, we would have the helicopter flying around in a simulator so it could crash without hurting anyone. But we would let the AI fly the helicopter however once, and whenever it flew the helicopter well, we will go, good helicopter. And when if it crash, we go bad helicopter. And then there was the AI's job to learn how to fly the helicopter to get more of the good helicopter rewards, and fewer of the bad helicopter negative rewards. More formally, a reinforcement learning algorithm uses a reward signal, to tell the AI when it's doing well or poorly. This means that whenever it's doing well, you give it a large positive number to give it a large positive reward. And whenever it's doing really bad job, you send it a negative number to give it a negative reward. And it's the AI's job to the automatically learn to behave so as to maximize the rewards. So, good dog responds to giving a positive number, and bad dog or bad helicopter, corresponds to you giving a negative number. And that the AI will learn to get more of the behaviors that results in large positive numbers, or in large positive rewards. Let me show you a video of the Stanford autonomous helicopter after we did this. This is a video of the helicopter flying under reinforcements learning control. I was the cameraman that day, and when you zoom out the camera, you see the trees pointing the sky. So, we actually gave it a reward signal which rewarded the helicopter flying upside down. And using reinforcement learning, we built one of the most capable autonomous helicopters in the world. In addition to robotic control, reinforcement learning has also had a lot of traction in playing games, games such as Othello or checkers or chess or Go. You might have heard of AlphaGo, which did a very good job of playing Go using reinforcement learning. And reinforcement learning has also been

very effective at playing video games. One of the weaknesses of reinforcement learning algorithms is that they can require a huge amount of data. So, if you are playing a video game, a reinforcement learning algorithm can play, essentially, an infinite number of video games. Because it's just a computer playing a computer game, and get a huge amount of data to learn how to behave better. Or for playing games like checkers, or other games. It can play a lot of games against itself, and for free, get a huge amount of data to feed in its reinforcement learning algorithm. In the case of the autonomous helicopter, we had simulator for the helicopter, so it could fly in simulation for a long time to figure out what works and what doesn't work for flying a helicopter. There so a lot of exciting research work being done to make reinforcement learning work, even for settings where you may not have an accurate simulator. Whereas harder to get this huge amounts of data. Despite the huge amount of media attention on reinforcement learning, at least today it is creating significantly less economic value than supervised learning. But there may be breakthroughs in the future that could change that. And AI is advancing so rapidly that all of us certainly hope that there will be breakthroughs in all of these areas that we're talking about. GANs, or generative adversarial networks, are another exciting new AI technique. They were created by my former student Ian Goodfellow. GANs are very good at synthesizing new images from scratch. Let me show you a video generated by a team from NVIDIA, that used GANs to synthesize pictures of celebrities. And these are all pictures of people that had never existed before. But by learning what's celebrities look like from a databases celebrity images, it's able to synthesize all these brand new pictures. There's exciting work by different things right now on applying GANs to the entertainment industry. Everything ranging from computer graphics to computer games, to media, and to just making up new contents like this from scratch. Finally, the knowledge

graph is another important AI technique that I think it's very underrated. If you do a search on Google of Leonardo da Vinci, you might find this set of results with this panel on the right of information about da Vinci. If you do a search on Ada Lovelace, you'll also similarly find a panel of additional information on the right. This information is drawn from a Knowledge Graph, which basically means a database that lists people and key information about these people. Such as their birthday, when they pass away, their bio, and other properties of these individuals. Today different companies have built knowledge graphs of many different types of things not just people. But also they built these data bases of movies, of celebrities, of hotels, of airports, of scenic attractions, and on and on and on. For example, a Knowledge Graph with hotel information may have a big database of hotels as well as key information about these hotels. So that if you look them up on the map you can find the right information relatively quickly. The term Knowledge Graph was initially popularized by Google, but this concept has spread to many other companies. Interestingly, even though Knowledge Graphs are creating a lot of economic value for multiple large companies at this point, this is one subject that is relatively little-studied in academia. And so, the number of research papers you see on Knowledge Graphs, seems to be disproportionately small, relative to the actual economic impact today. But, depending on what industry vertical you work in, perhaps some of the techniques for building Knowledge Graphs, will also be useful for building a big database of information about something relevant to your company. In this video you learned about unsupervised learning, transfer learning, reinforcement learning, GANs, and knowledge graphs. It seems like a lot, doesn't it? I hope that some of these ideas will also be useful to your projects, and that knowing what these algorithms are, will make it easier for you to fruitful discussions with AI engineers. In this week we've talk a lot about how AI can affect companies. Maybe how you could use AI to affect your company. AI is also having a huge

impact on society. So, how can we understand the impact that AI is having on society, as well as make sure that we do ethical things? And that we use AI only to help people and make people better off? And next week we'll talk about AI and society. Thanks for sticking with me up to this point, I look forward to seeing you in the final week of videos for this course.