[SQUEAKING]

[RUSTLING]

[CLICKING]

SPEAKER 2: Before proceeding, please view the following video-- Mashable, "Google's AI Assistant Can Now Make Real Phone Calls."

BRANDON LESHCHINSKIY: Hi, everyone. My name is Brandon Leshchinskiy. Thank you so much for having me. What we just saw was a very good example of, I think, what most people would call AI. Now, let me ask, what is AI?

SPEAKER 1: Answers from participants:

SPEAKER 2: Artificial intelligence is something that can learn. It's more than a program that follows rules. It's able to adapt.

SPEAKER 1: Something to make our lives easier.

SPEAKER 2: A machine that simulates human decision-making and understanding to a degree that it's indistinguishable from a human being.

BRANDON LESHCHINSKIY: So already, we're seeing one of the issues here, is that "AI" is a very overloaded term. I think most typical definitions are closest to that last one, that artificial intelligence is when a machine or a computer can exhibit some signs of human-like intelligence. Somebody mentioned learning. That would be a subset of artificial intelligence, which we'll get into.

But in any case, part of the point here is that AI is a very overloaded term that can mean a lot of different things to a lot of different people. And the joke in the AI community is that it's AI until we figure out how to do it, and then after that, it's just coding.

Like I said, thank you so much for having me. My name is Brandon. I'm a recent graduate of MIT. And in today's workshop, we'll cover the basics of artificial intelligence. We'll talk about machine learning. We'll train an algorithm together. And I'll leave you with a couple of questions and a word of caution.

Before we get into all of that, I want to start with the main takeaway, which is, with the right data and the right model, artificial intelligence can solve many problems. But choosing the right problem, finding the right data, and training the right model can be difficult. So if there's nothing else that you remember from today's workshop, this slide is the key takeaway.

And like I said, we'll cover artificial intelligence. We will see a few different examples of tools that can be used to implement artificial intelligence, including something called machine learning. We'll talk about different kinds of machine learning and different problems that can be solved by machine learning. And let me just say, since, probably, a lot of you are already wondering, what is deep learning?

Deep learning is either a subset or a subset of a subset of machine learning. So for those of you who are familiar with neural networks or deep learning, that is a tool that might be used in machine learning problems. We won't talk about that in any depth at all today, but I hope that by the end of the workshop, you guys will have the fundamentals necessary to dig into deep learning at a conceptual level if you want to do that some other time.

All that being said, let's get into basics of AI. So AI can be general or it can be narrow. When we talk about general artificial intelligence, it's when a machine can learn any task that a human can perform. This does not exist. There are no Terminator robots. There are no-- there is no such thing as general AI. And experts vary widely in terms of their predictions on when this may actually happen. So there are some people who say, oh, 10, 20 years, there are some people who say not in our lifetimes, and some people who say, not ever. So really, really broad range of guesses.

In any case, most AI-- sorry, all of the AI that exists today and most of the AI that you hear people talking about is narrow. And narrow AI is when a computer exhibits intelligence at one task. To be clear, you can do pretty impressive things with narrow AI. A self-driving car for example, has one narrow AI system that does vision. It looks at the road and interprets things. There's another AI system that does steering, that controls the car. There's another AI system that does route planning-- that says, on a broad level, where are we going, how do we get there?

And when you stitch all of those together, you can actually create a very impressive system. But you cannot sit this car down and play chess with it. This car will not learn how to paint. This car will not learn how to speak Russian. It's doing a specific set of tasks, each of which has been trained separately. And that is narrow AI. So from now on, when I say AI, I'm talking about narrow artificial intelligence. Any questions so far?

SPEAKER 2: Question from participant:

SPEAKER 1: What category do you fall in with regard to general AI? When do you think it will be possible?

BRANDON LESHCHINSKIY: I don't think it's going to happen for a while-- like many, many decades. But I also think that it's definitely going to happen at some point. Many decades to maybe centuries. I don't know. I Think technological progress is hard to guess at. So I think it's a fool's errand. Anyway, so what does narrow AI actually do? Well, typical tasks, as we've seen with the self-driving car example, include vision, language processing, and planning. These are just a couple of examples.

So vision is going to be understanding or creating images or video. Language processing-- understanding or creating speech or text. That's going to be, like, Siri or Amazon Alexa. And planning. Lots of different kinds of planning. You can do route planning, motion planning, task planning. All of these things fall under the category of narrow AI tasks that you can train an algorithm to do.

As far as how we actually build AI, there's a lot of different ways. Two classical ways to implement artificial intelligence-- and I remember learning about this and being like, wow, this is not intelligence at all. This is super dumb. Are expert systems and tree search. So in an expert system, it's just a person writing a list of rules. If this, then that. If this, then that. This can be super useful for certain problems.

But a lot of the things that we actually think of as really straightforward turn out to be quite difficult. So for example, we can identify a table like that-- very, very easily. We know what the table is. But I challenge you to write a list of rules that can define a table. What color is it? What's it made of? How many legs does it have? What shape is it? So it turns out that even something like "what is a table" becomes really, really challenging to answer using an expert system.

So these can be really helpful. And there are some simple ways for their other-- there are some problems where heuristics are actually the best way to do it. Because it's interpretable, you know what's going to happen, and you don't need a neural network for every problem. But this definitely can't solve every problem [INAUDIBLE].

Another kind of algorithm people might use in classical AI, it's called a tree search. You start in your current state. That's state A. And you enumerate all of the possible options from there. And then you just repeat that for each state until you find the state where you want to end up in-- your goal state. Now, obvious problem here-- a lot of options.

A lot of options when you're operating in the real world. So how do you actually enumerate all of them? And sometimes, it can even be difficult to define a goal state. When I'm playing chess, I know there's a goal state of win-- checkmate. But in real-life environments, it's not so simple. So again, can be really, really helpful. And this is how the chess computers beat Garry Kasparov. But also limited.

So now, we see these are just two ways to implement AI. They're not that smart. Like, most of us could have come up with these. If we were just told, like, come up with a way to solve this problem, we probably would have figured out that a list of rules or enumerating all of the options-- yeah, that makes sense.

So let's now move into machine learning, which is a subset of artificial intelligence that allows a machine to learn automatically from data. And this is what somebody else mentioned. So this is where there's not a programmer who's sitting there explicitly programming or changing the code, changing the model. It's like oh, well, if this happens, then you should actually do that. It is learning automatically by looking at data. And I'll say more about learning in just a second.

But let me just first answer the question of, when do we use machine learning? So machine learning can perform many different tasks. Three of the most common are classification, clustering, and regression. Classification means, here's some input data. I'm going to give you an output of a class. So maybe it's an image, and I tell you what kind of animal it is. Maybe it's a list of information about somebody and I tell you whether they're a threat or not. You give me an input, I give you a class.

Clustering is, again, you give me some kind of input, and I'm going to find clusters. I'm going to find things that are similar. Maybe I'm clustering different groups of animals or other similar groups of animals together. Maybe it's a similar gene sequences. Oh, these look related somehow. So clustering can be really useful, especially in genetics, from what I've heard-- not that I've ever done that.

And then another kind of problem is regression. And regression is, again, you get input data. But instead of putting a class, like "dog" or "safe," you put a numerical prediction. So maybe you're predicting home price based on square footage. Maybe you're predicting crime rate based on location. And in this example-- let me just ask you as a question-- what's the model that we're using? What's the model the machine-- technically, it's a machine learning model. Or, you can call it that. What's the model that we've trained in this example? What's actually doing the prediction?

SPEAKER 1: Answer from participant.

SPEAKER 2: The red line?

BRANDON LESHCHINSKIY: Exactly. It's a line. And I just want to say this, because I think a lot of people think models are like really complicated. Sometimes, the model is just a line. Like, this is a model. I give it some kind of input, and it gives me an output. This is a model. You can also use a curve. You can use a neural network. It's way more complicated-- a lot more operations there. But these are all models.

So like I said, machine learning can perform many tasks. Classification, clustering and regression are the most commonly performed. And now, I want to talk a little bit about what it actually means to learn something. What does it mean for a machine to learn? Are there any questions up until this point?

SPEAKER 2: Question from participant:

SPEAKER 1: Where does a convolutional network fall in the scheme of things?

BRANDON LESHCHINSKIY: So convolutional neural networks are a very specific tool that are used usually in computer vision. That is a model. And we will-- let me get back to convolutional neural networks after this section. I'll talk briefly about them. So three types of learning, guys. There's supervised, unsupervised, and reinforcement learning.

When we talk about supervised learning, it's really like studying for a test. I'm going to give you the example. Sorry, I'm going to give you the questions and the answers in advance. You're going to study. Them you're going to figure out what the broad patterns are. And then I'm going to test you. This is often-- doesn't have to be, but it's often applied to classification problems.

Now, the key thing here is that, in supervised learning, you need to have the answers in advance. You need to have labeled data, in other words. I need to-- it's not enough for me to just show the model a bunch of images. The images have to be labeled as cat, cow, giraffe, dog, whatever. Threat, not threat. There have to be answers. Otherwise, it'd be like giving you flashcards with only one side. And you're turning it over, and you're like, I have no idea. So that's what supervised learning is.

In unsupervised learning, it's more-- you don't have those answers. You don't have those labels. It's more like a baby playing. You're just discovering patterns. This is often used in clustering. I don't know, a priori, what the right clusters are going to look like. I have no idea what they should look like. All I know is, oh, these looks something-- they look somewhat similar. These look somewhat similar, and they look kind of different from each other. But I have no external source of information as far as what kind of class this belongs to. Does it make sense? So that's unsupervised learning.

And then the last kind of learning that we'll talk about is reinforcement learning. Reinforcement learning is, again, you don't really have a right answer, and you're not really playing either. It's something separate. You are aiming to reward some kind of behavior or some kind of solution to a problem. It's like a rat in a maze. You give it cheese, and it discovers winning strategies on its own.

So you don't have the right answer. You just know that you need the thing to get out of the maze. This is often used in robotics, for example, with training a robotic arm, or maybe training an agent in a game. So supervised, unsupervised, reinforcement learning. These are the three ways to learn in machine learning. Any questions about these three things?

SPEAKER 1: Question from participant:

SPEAKER 2: Is reinforcement learning where we know what we wanted to do and technology figures that avenues with parameters to get there?

BRANDON LESHCHINSKIY: Exactly. So you might, for example, say, win this game, and over the course of a million simulations, the thing learns, oh, probabilistically, this has a good chance of winning the game. This does not let me do this thing. Now, let me just actually add about reinforcement learning-- all of these types of learnings have pros and cons, of course. So with supervised learning, the trade-off is, you need just, like, a stupid amount of labeled data. And then that takes forever. That can be very, very expensive to collect and label data.

With unsupervised learning, you don't necessarily know what the right answer is. So it can help you discover patterns, but it may not be as powerful as supervised learning. And with reinforcement learning, the real trick is, how good is your simulation? Because you're not going to take a real car and do reinforcement learning with a real car. It's going to take you a million iterations. You're going to have to go through a million cars before it learns how to drive without crashing.

When you simulate it, then the question is, how good is your simulation? Will it actually apply to the real world? Are you capturing all of the noise, all of the disturbances, all of the possibilities? Are you capturing that biker who cuts you off, or the wind, or the reflection of the sun on a white sign? So there are pros and cons to each of these.

So now, what we're going to do is, we're actually going to train our own algorithm. And again, I want to reiterate the main point of this presentation before we get into that algorithm training. With the right data and the right model, AI can solve many problems. Choosing the right problem, finding the right data, training the right model are all potentially difficult.

So let's do this. We're going to walk through the seven steps of training an algorithm. Step one is, define a problem. Now, here, I've defined it for us. I've said, we're going to decide whether a picture is a cat or a dog. So first, let me ask you, what kind of machine learning problem is this?

SPEAKER 1: Answer from participant:

SPEAKER 2: Classification.

BRANDON LESHCHINSKIY: Cool. Everybody see why? I'm getting some kind of input. I'm getting an image. And I'm telling you what class it is, what class it belongs to. It's either a cat or a dog. Now, in this case, the work of problem identification has been done for us, and we're assuming that classifying between cats and dogs is actually closing an important gap.

However, let me just say, problem identification is super important. There are a lot of problems that cannot be solved by AI, period. If you have issues with diversity in your unit, machine learning is not the answer. But as with any new technology, it's really seductive. It's really sexy. Everybody wants to be like, oh, we're using AI. So I just want to make clear that it's important to be discerning with where you even apply AI. And that comes into that very first step of, what's the problem that we're trying to solve? Define the problem.

Step two is, cool, I have a problem. Find data. What kind of data do I need in order to solve this problem? So let's say I wanted to train an algorithm that can classify between cats and dogs. What kind of data should I look at?

SPEAKER 2: Answer from participant:

SPEAKER 1: It's going to be a lot of visual data, lots of images to show cats can look all of these different ways, dogs can look all of these different ways.

BRANDON LESHCHINSKIY: So something like this. Exactly. Now, this data has labels. So what kind of learning are we able to do as a result?

SPEAKER 2: Answer from participant.

SPEAKER 1: Supervised learning.

BRANDON LESHCHINSKIY: Yes. Because we have the right answer in advance. So we can train it and say, ah, this is a cat. This is a dog. Now, if we didn't have labels, what our algorithm might do is cluster them together and say, oh, there's this type of thing and there's this type of thing, and they look different. But there's no external reference. The thing has no idea what a cat or a dog is. It's just saying, these two things look different. Everybody see the difference there between supervised and unsupervised? So this way, we actually can say, this is a cat. This is a dog.

SPEAKER 1: Question from participant:

SPEAKER 1: Is this how Google and other companies have been developing their technology, using humans to develop data pools?

BRANDON LESHCHINSKIY: Yeah, there's-- I don't if you guys-- if anybody here is familiar with*XKCD.* It's an online webcomic. There's one where it's like, it's a Google sign-in thing, and it's like, please find the picture with the stop sign. Next frame-- please do it quickly. Our car is approaching the intersection. So yes, this is absolutely them collecting training data by having humans label it.

So the next step after we've found data is clean data. And I know this feels like, maybe, extraneous. But for anybody here who has worked with data before, you get this is, like, 80% of data science and 80% of AI. Cleaning data is the biggest pain, but it's most of the work. It's because data doesn't just exist in this nice way where you can show it to a machine learning model and it starts learning.

Like, data exists in some weird directory on the internet where you have to download it, you have to reorganize it, you have to name it properly, you have to make sure that there are no errors there, you have to make sure that you can access it quickly, blah, blah, blah. So cleaning data is a huge part of what it means to be a data scientist and to do machine learning or to do AI.

And that's why if you have already clean data, oh, my god, it's like a data scientist's paradise, because they don't have to do that work. They can just go and download stuff and train models, and that's the fun part. So getting good, clean data is a huge boon for any kind of AI development. For us, we're going to say that this is now clean.

Step four is, choose a model. I'm going to show you guys a model. Now, don't get attached to this model. It's not a good model. It's not a helpful model in the real world. It's a model that we're going to use to illustrate a few key concepts. So this is our model. Dogs always blank, sometimes blank. Cats-- always, sometimes. This model is interpretable. It means that it may not be good, we'll find. But at the very least, we'll be able to know what it's doing. It's not going to always be true. But this is the model we're going to use.

So now, let's train it. Let's see what it means for our model to learn something. We're going to show the model inputs, and we're going to ask it, what is this thing? If it answers them correctly, model's great. Doesn't need to be updated. If it gets it wrong, maybe we can improve the model somehow. So at the very beginning, we're going to show this picture.

Now, the model has never seen anything at this point. It's a totally blank slate. It's just going to make a random guess. So obviously, this is a cat. We know that. If the model guessed correctly, there would be no update to the model. As far as the model is concerned, I'm 100% accurate. Why would I change?

Well, let's say that the model gets it wrong. Let's say the model guessed dog. Then in that case, we would go ahead and update the model. So again, we have, model now knows it's a cat. It's going to update itself. So what should the model now say about cats? What does the model-- based on this example, what does the model now know about cats?

SPEAKER 2: Answers from participants:

SPEAKER 1: Are you looking for things like pointy ears, pink nose, round eyes?

SPEAKER 2: I feel like there's not even enough data to say that yet. With such a small training set, all the model knows is that a cat looks like that.

BRANDON LESHCHINSKIY: Ah, OK. So let me ask-- so good intuition. So we heard pointy ears, multicolor, laying on a bed, maybe, paws. In our model-- let me update it-- are we going to say always have those things-- cats always have those things, or cats sometimes have those things? Do cats always have pointed ears and are on a bed and have a pink nose, or is it sometimes? What do we think?

SPEAKER 1: Answer from participant:

SPEAKER 2: Sometimes, definitely.

BRANDON LESHCHINSKIY: OK. Any different opinions? So here's the thing. Corey knows what cats are. Our model is dumb. Our model knows nothing about the world. It knows nothing about cats. As far as the model is concerned, as somebody else just mentioned, this is the only cat in the world. This is what every cat looks like. Every cat is this cat, and this cat is every cat.

So from the model's perspective, cats always look exactly like this thing. It's the only example it's ever seen of a cat. What do you want? This is what cats look like. So the model is purely statistical. It has no idea what a cat is conceptually. It just says, statistically speaking, 100% of cats look exactly like this cat.

Now, we show it a second input example. OK. Now, at this point, our model's looking at this image. It's saying, well, all of the cats that I've seen look like that other cat, and this does not look like that other cat, so this must be a dog. I haven't seen a dog yet. This is probably a dog. OK. Turns out, model's wrong. Model has to get updated. So at this point, what is our model going to-- so tell me about this cat. What is the model going to know about cats after this example?

SPEAKER 1: Answers from participants:

SPEAKER 2: They're mean-looking. Cats can be super fluffy and have yellow eyes.

SPEAKER 1: Their pupils can be small or big.

BRANDON LESHCHINSKIY: Good. OK. So now, let me ask, are we going to put this in sometimes or always, these examples we just got?

SPEAKER 2: Answer from participant:

SPEAKER 1: Sometimes.

BRANDON LESHCHINSKIY: Some of them are going to be sometimes, right? So cats still have two eyes, always-- so far. All cats have two eyes. All cats have a nose that looks roughly like this. Sometimes, it's a pink nose. Sometimes, it's a black nose. So now, we are fleshing out our model a little bit more and we're seeing, oh, these are things that cats always have and these are things that cats sometimes have. Does this make sense? OK.

All of the things that we just wrote down are called features-- the number of eyes, the shape of the ears, the size of the pupils. These are all features. Features are what allow us-- or allow the model, rather-- to make a decision. So for example, if I am looking-- I'm just walking around, trying to assess safety, which is, I think, a thing that people kind of do instinctually. There are features that I'm looking for.

Like, is the person that's approaching me holding any weapons? What's their posture like? What's their facial expression like? I'm not looking at the color of their hair. Like, that's not a relevant feature in that particular example. So in this case, all of the things that we're pulling out to say-- to identify, this is a cat or a dog, those are what we call features. Now, let me ask you a somewhat tricky question. When a computer sees an image-- let's say it's a 256 by 256-pixel image-- how many features does a computer see?

SPEAKER 1: Answer from participant:

SPEAKER 2: Is it 256, because that's how many pixels there are?

BRANDON LESHCHINSKIY: So it's 256 squared pixels. And that's how many features there are. The computer is looking at it. It's not looking at eyes. It's not looking at ears. It's not looking at notes. It's looking at, there's a pixel here, and there's a whole bunch more throughout the rest of the image. It's looking at an array of data.

Now, what a neural network does, and what a convolutional neural network does in particular, is the image gets passed-- so, whoever asked about convolutional neural networks, if you weren't paying attention, now's a good chance to tune in. What a neural network does-- a convolutional neural network-- takes an image, says 256 by 256 pixels. Passes it through the first layer of the neural network. Starts extracting features. With that first layer, maybe it's looking at lines, or edges, or color gradients.

Passes it through to the next layer in the neural network, and it combines those lines into shapes. Maybe now, it's looking for circles, or triangles, or whatever. Next part of the neural network. It gets passed through again. So it's processing it, processing it, making it more-- like, making the features-- automatically extracting features at each step, and making the features more and more sophisticated. So now, it's looking at noses, and eyes, and now finally, at the very end, it's looking at a whole face or a whole body.

So that's what a convolutional neural network does. But it starts, with the model's perspective, as 256 by 256 features. And what a neural network is sometimes really, really good at is extracting the features that are relevant, combining them, and processing them until they're more and more sophisticated. And now, you have facial recognition, for example. There's, like, some kind of eye/nose/mouth structure that it's looking for. Oh, this is your face. I know who this is. So that's what a convolutional neural network does.

But I just want to be clear, again, models are dumb. They're not looking at this from the perspective of, I know what a nose or a cat or a dog is. They're looking at it from the perspective of, I see an array of data. And sometimes, a neural network, a model, can interpret it and create these features that are more sophisticated. I'll also say, these models can be very brittle.

So there was a paper written a few years ago where they basically tricked the neural network, and by changing just a few pixels or adding some noise, it still looks like a turtle to humans-- we still say it's a turtle-- but the model looks at it and says, oh, is a gun, or oh, this is a rooster. And they could change anything into anything, as far as the model was concerned.

So then you can train your model on that data and you can make the model more robust to it. So you end up with this kind of adversarial play similar to other defensive operations, I assume. But at any rate, that's a bit of a tangent. Any questions so far on training? We're going to do a dog example really quick. OK.

So next example-- this thing. So again, we're going to assume the model gets it wrong. So there's an update. We're going to describe this dog. It's black. It's got floppy ears. It's got brown eyes. It's got a pink tongue, four legs. Where's this going, always or sometimes? What do you think? Always. This is the only dog that exists in the whole world. This is what every dog looks like. Every dog is this dog.

Now, when we see-- oh, yeah, sorry, dog, as we know. Now, when we see this guy, the model is going to update. And now we have that change, where it's, OK, there's two eyes still, so every dog has two eyes. Until we see a dog with only one eye. And now, the model is like, OK, well, it looks like most of the time, dogs have two eyes, and sometimes, they have one. Yeah, statistical model. But at this point, 100% of dogs look exactly like this dog.

So that's how we train a model. We now have a model of what cats and dogs look like, or do, or are. So now, we go and we test the model. And it's really, really critical, when we test the model, we use data that the model has not seen before. If I give you the exam in advance, then you can just memorize the questions and the answers. That's not really helpful for me to determine whether you've actually learned the material.

I want to give you some sample questions, sample answers, and then on the test, I'll give you different ones. Because I want to make sure that you can generalize to other problems, to other kinds of questions. If you just memorize the solutions, it's not helpful for me to understand whether you've learned it.

So when you're testing the model, it is absolutely critical that you're using data that the model has not yet seen. Now, in this case, what we're finding is, this thing is black. It has two eyes. Does have pointy ears. But the thing is-- like, and our model has never seen a black cat before. It's seen a black dog, though. So now, our model is confused. What is? What am I referring to? Anybody know what this problem is?

SPEAKER 1: Answer from participant:

SPEAKER 2: Boundary conditions?

BRANDON LESHCHINSKIY: Not quite. This is a problem of bias. This is a problem of bias in your data set. We don't have any black cats. So our model is not great at recognizing black cats. Now, you can imagine, in medical contexts, this becomes really problematic, because we're training data on mostly white people and mostly men, for example. And that doesn't help for not-white people and for not-men.

Because the model is going to learn, as we saw, statistics-- models are dumb. Model's going to learn a whole bunch of things that are true about white people and men, or about cats that are not black, and then we're presented with this other thing-- this other type of thing.

The model has no idea what to do with it, and just applies the same assumptions to this other thing. So what's the lesson here? Make sure that your training data is representative. And be very intentional about the way that you are collecting training data to make sure that it reflects the use case. This goes back to problem definition. What problem are we trying to solve? Who is our end user? What are we hoping to accomplish?

All of these things that have to happen in good engineering. Yeah, this is a great example. So if we use AI to define what makes a good pilot, the AI is going to say, oh, probably white guys. White guys are good pilots. I can't solve this problem for you guys, but I can make you aware of it so that you're asking the right questions. And of course, the model will eventually know that this is a cat, hopefully.

So now, step seven. Deploy the model. You can do this on Amazon Web Services. You can make it publicly accessible. You can put it on a secure hard drive somewhere. However you guys are going to use it, that will dictate how to deploy it. So everything flows from problem definition. You can put it on a chip.

But to review, we have, define a problem, find data, clean data, choose a model, train the model, test the model-- whoopsies-- and then deploy the model. Questions on this process?

SPEAKER 1: Question from participant:

SPEAKER 2: Is this the typical approach regardless of the model that is used?

BRANDON LESHCHINSKIY: I think so. Implicitly or explicitly, this is generally the process. Sorry, go ahead, Jennifer.

SPEAKER 2: Question from participant:

SPEAKER 1: So if you showed the model a guinea pig, it would try to classify it as a dog or cat because those are the options we had presented?

BRANDON LESHCHINSKIY: Correct. So that's-- so that's step four. What model are we using? And the model architecture might limit you. There's only two outputs, either cat or dog. You don't have a third output, you need a third-- you need to make space for that.

SPEAKER 1: Question from participant:

SPEAKER 2: I'm working on a master's degree, and I completed a class in data analytics. I was working with R, and one of the questions I had at the end of the class was, is there an AI program or progress towards AI that clean data?

BRANDON LESHCHINSKIY: There are-- I think they're working on it. There's, like, AutoML tools which are trying to democratize AI, where-- but even then, my guess is that you would have to input the data in a way that it finds useful, and then it takes it from there. I don't know about cleaning the data. I really don't know. Any other questions before we move on to summary and conclusion? OK.

So to conclude, we've covered an intro to artificial intelligence. We've seen, step by step, how to train an algorithm. And to close, I want to leave you all with four questions and a word of caution. Four questions you should ask about any AI that you encounter, whether you're trying to assess it or use it. Four questions you should always ask. Number one: What is the goal? Problem definition, right? What is the goal? What problem are you solving? What are its inputs and its outputs? What does the thing do? That's number one.

Number two: What is the training data? Where did it come from? How was it prepared? How did you make sure-- did you check for this bias or that bias? How did you make sure that it's balanced? Number three: What kind of model did you use? Why? Is it interpretable? Should it be? How did you pick this model?

And number four: What's the accuracy when you test it on new data, when you test it on testing data-- not training data. When you give it a whole new set of data that it's never seen, what is its accuracy? The reinforcement learning version of this would be, what happens when you test it in a different simulation or in the real world? But something-- there has to be some question about, how well does this actually do when it's given a problem that it has not had the opportunity to memorize? So goal, training data, model, accuracy. If you keep these four questions in mind, you'll be better able to assess any AI system you encounter.

And now, let me close with a word of caution. Developing an AI model can be very difficult, can also be very valuable, and it can also be very seductive. It is a task. If the task does not belong to a strategy in support of certain objectives that measure a mission that reflect values, it is not a good use of time, even if it is really cool.

So when I say goal, I'm talking, again, both at a low level-- inputs and outputs of this model-- but also at a high level. What are we trying to do? And it's important that we constantly reevaluate the low-level goals with respect to the high-level problem. We want to situate the model, and the training, and the data within a broader problem-solving context.

If I were looking at the Air Force and trying to figure out, where would AI-- or where to implement AI, I would start here. I would start with identifying, what are the values? What's the mission, objectives, strategy, tasks? Kind of as a tree. And I would identify problems. What are the gaps between our aspirations and our reality? Then I would think critically about which gaps artificial intelligence actually could fill and whether those gaps would be partially or completely filled.

If you have data that's unused, that presents an opportunity. If you have tasks that require a lot of people, that presents an opportunity. If you have tasks that are complex or require a lot of thinking and processing data, those are problems where AI can help. If you have a problem with drug overdoses or with suicides, AI is probably not going to help. And in fact, I've heard of cases where people collect data, and they do all of this quantitative analysis, but this is fundamentally a human problem. And so it can be actually quite distracting.

So my point is simply that, at its best, AI can make the Air Force more efficient, more effective. At its worst, it can distract us from actually solving difficult, challenging human and cultural problems. So it's really important to be discerning about what kind of problem you're trying to solve and where AI can help.

But at this point, I hope, and I think that you have the tools to prevent that from happening, and to harness this very powerful new technology in support of building a more effective Air Force and pursuing America's interests. So thank you all so much. I'm going to stick around for questions, and I will leave you with this conclusion slide.