Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **What it means to be AI-first**

Format: Screencast

Google Cloud

What it means to be AI-first

How Google does ML

Machine Learning on Google Cloud Platform

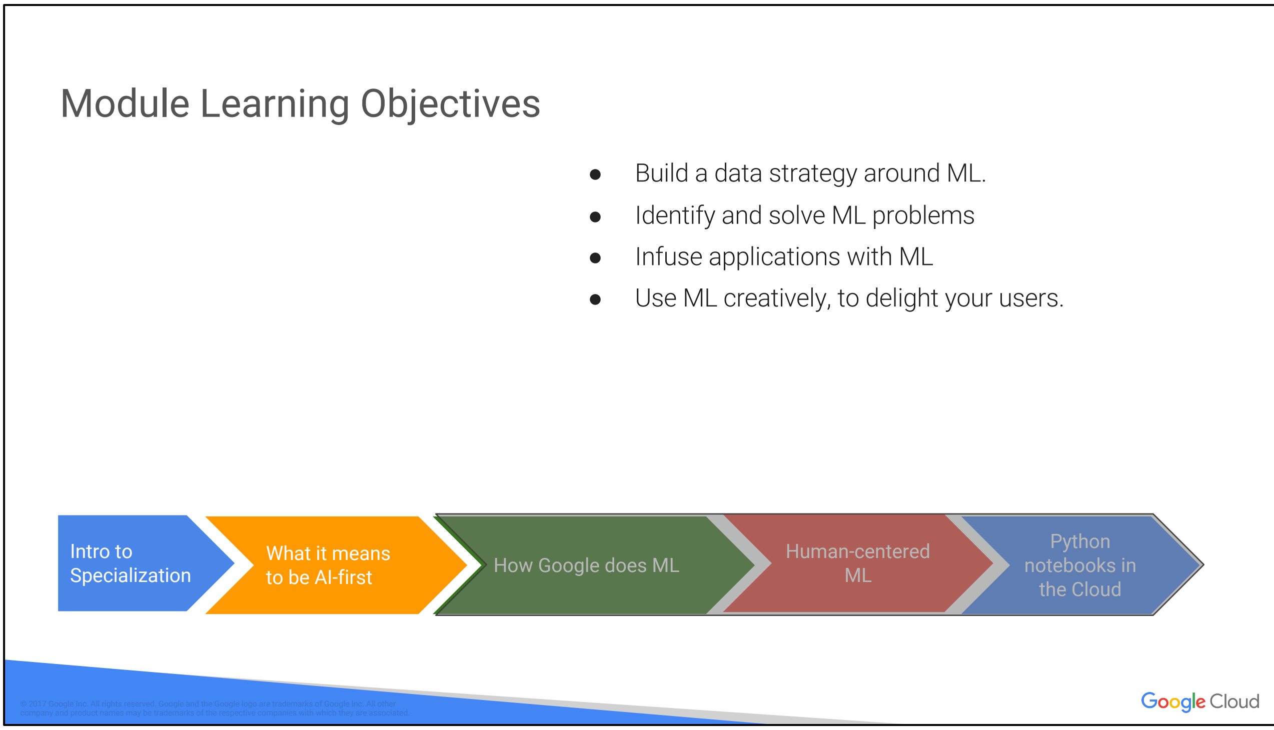
Google Cloud

(Lak)
Machine Learning has completely transformed the way we at Google build our products.
In this chapter, you will learn what we mean when we say that Google's company strategy is to be AI-first, and what that means in practice.

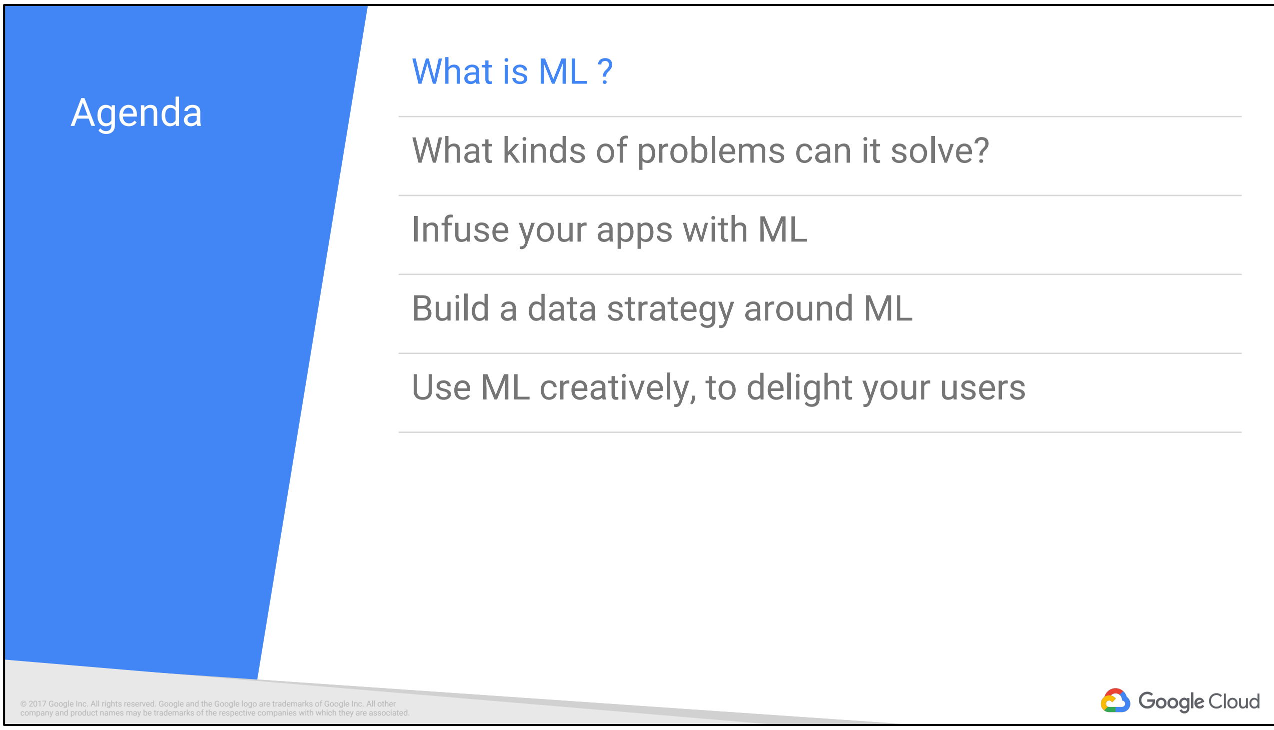
Of course, the goal here is not to just talk about Google, but about how you can apply what we have learned in our journey to using ML everywhere.

Among the things you will learn is how to take a business problem and look at it with fresh insight --
The insight will help you employ ML to help augment your users' decision making.

A few years ago, Google decided to focus on mobile, and we began to do things "mobile-first". Instead of building applications for the web, for example, and then porting them to mobile devices, we decided that we would build for mobile first. This sort of slogan serves as a rallying cry across the organization, and helps focus our collective mind on new opportunities.
It in that context that you should understand "AI-first". Of course, what this means in practice is quite different from what "mobile-first" meant in practice.
In the case of AI-first, what this means is that we consciously rethink our approach to how we build our products.

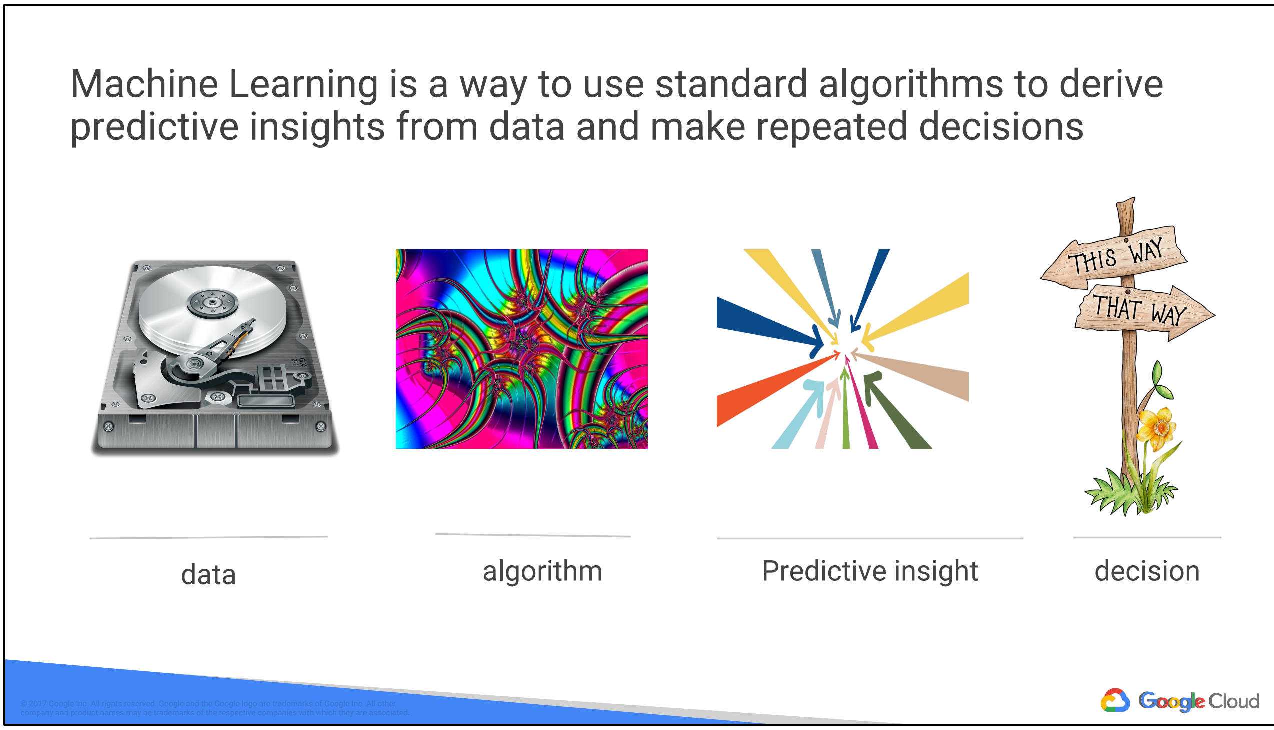


In this module, we will build a data strategy around ML, learn how to identify and solve ML problems and use ML creatively.



We will start by defining what we mean when we say machine learning.
And what does it mean to be AI-first?
This comes down to a unique answer we provide to the second question here -- what kinds of problems can ML solve?
Then, we'll look at three specific strategic things you will do: infusing applications with ML, building a data strategy, and finally about using ML creatively.

We will start by defining what machine learning is.



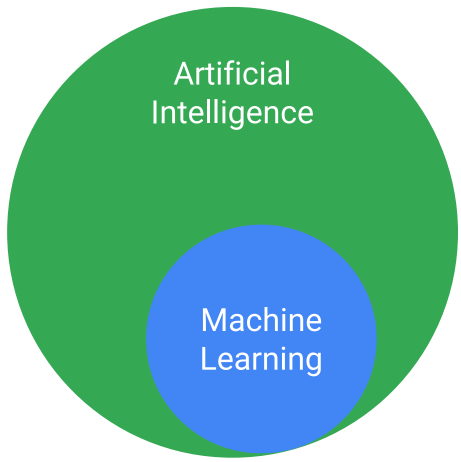
ML is a way to derive *predictive* insights from data.
We do this using algorithms that are relatively general and applicable to a wide variety of datasets.
Think back to your company. How do you use data today? Perhaps you have a dashboard that business analysts and decision makers view on a daily basis? Perhaps a report that they read on a monthly basis? That is an example of backward-looking use of data -- looking at historical data to create reports and dashboards; that tends to be what people mean when they talk about business intelligence. A lot of data analytics is backwards-looking.

Of course, the point of looking at historical data might to be make decisions. Perhaps business analysts examine the data and suggest new policies or rules? They suggest, for example, that it might be possible to raise the price of a product in a certain region. Now, the business analyst is making a predictive insight. But is that scalable? Can the business analyst make such a decision for every product in every region? Can they dynamically adjust the price every second? In order to make decisions around predictive insights repeatable, you need machine learning. You need a computer program deriving such insights.

So, machine learning is about making many predictive decisions from data. It is about scaling up business intelligence and decision making.

https://pixabay.com/en/hard-disk-storage-computer-159264/ (cc0)
https://pixabay.com/en/fractal-pattern-abstract-form-142748/ (cc0)
https://pixabay.com/en/arrows-inside-pressure-request-2029157/ (cc0)
https://pixabay.com/en/sign-direction-kids-cute-paint-2792576/ (cc0)

Artificial Intelligence is a discipline; machine learning is a specific way of solving AI problems



A very common question I get -- what's the difference between AI and machine learning?
One way to think about it is that AI is a discipline, sort of like physics. AI has to do with the theory and methods to build machines that think & act like humans.
Machine Learning is a toolset, sort of like Newton's laws of mechanics; just as you can use Newton's laws to figure out how long it will take a ball to fall to the ground if you drop it off a cliff … you can use machine learning to solve certain kinds of AI problems.

The basic difference between ML and other techniques in AI (for example, expert systems) is that in ML, machines learn. They don't start out intelligent, become intelligent.

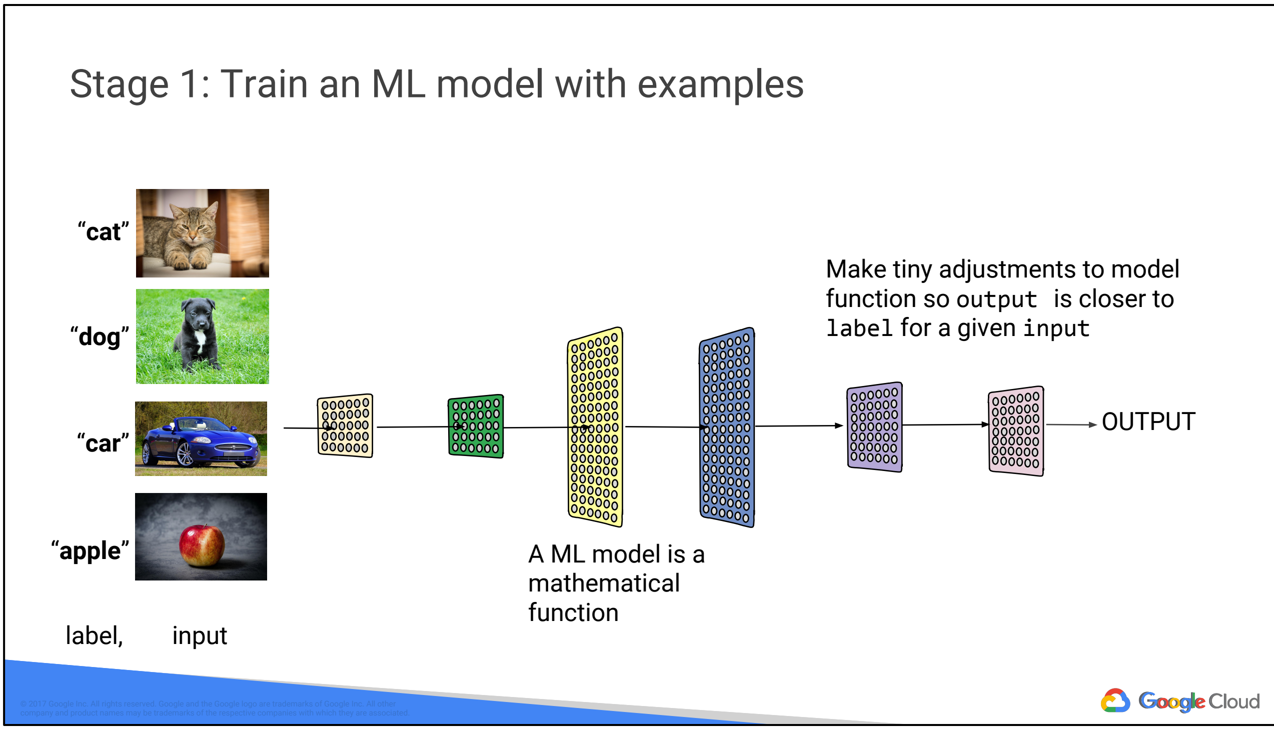
Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **Two Stages of ML**

Format: Screencast

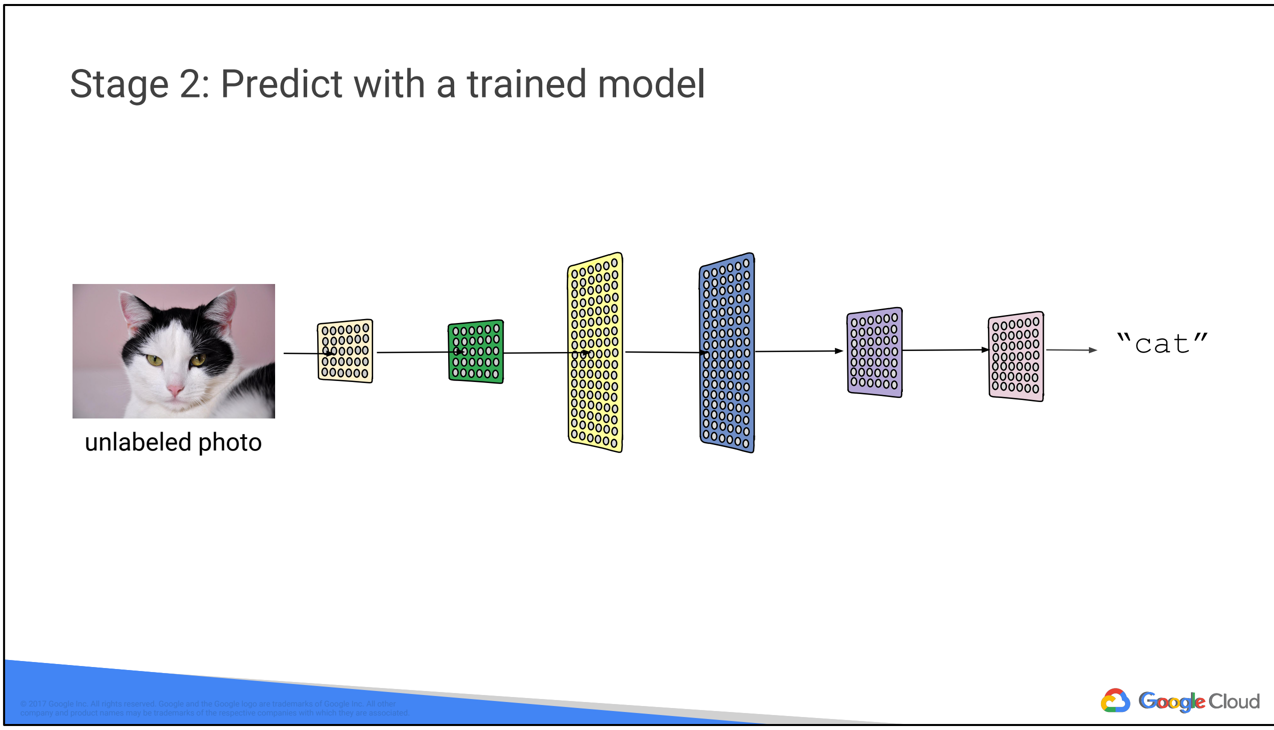
Google Cloud



(Lak)
The first stage of ML is to train an ML model with examples. The form of machine learning that we will be focused on in this specialization is called supervised learning, and in supervised learning, we start from examples.
An example consists of a label and an input. For example, suppose we want to train a machine learning model to look at images and identify what is in those images. The true answer is called the label, so "cat" for the first image and "dog" for the second image. The image itself, the pixels of the image, are the input to the model.
The model itself is a mathematical function, of a form that can be applied to a wide variety of problems.
The models used in ML have a bunch of adjustable parameters. Then when we "train" the model, what we do is to make tiny adjustments to the model so that the output of the model -- the output of the mathematical function -- is as close as possible to the true answer for a given input. Of course, we don't do this on one image at a time -- the idea is adjust the mathematical function so that overall, the outputs of the model for the set of training inputs is as close as possible to the training labels.

The key thing is that machine learning, at least ML of the form that we will consider in this course (the most mature form of ML), relies on having a dataset of labeled examples. Labeled examples -- input + the true answer.

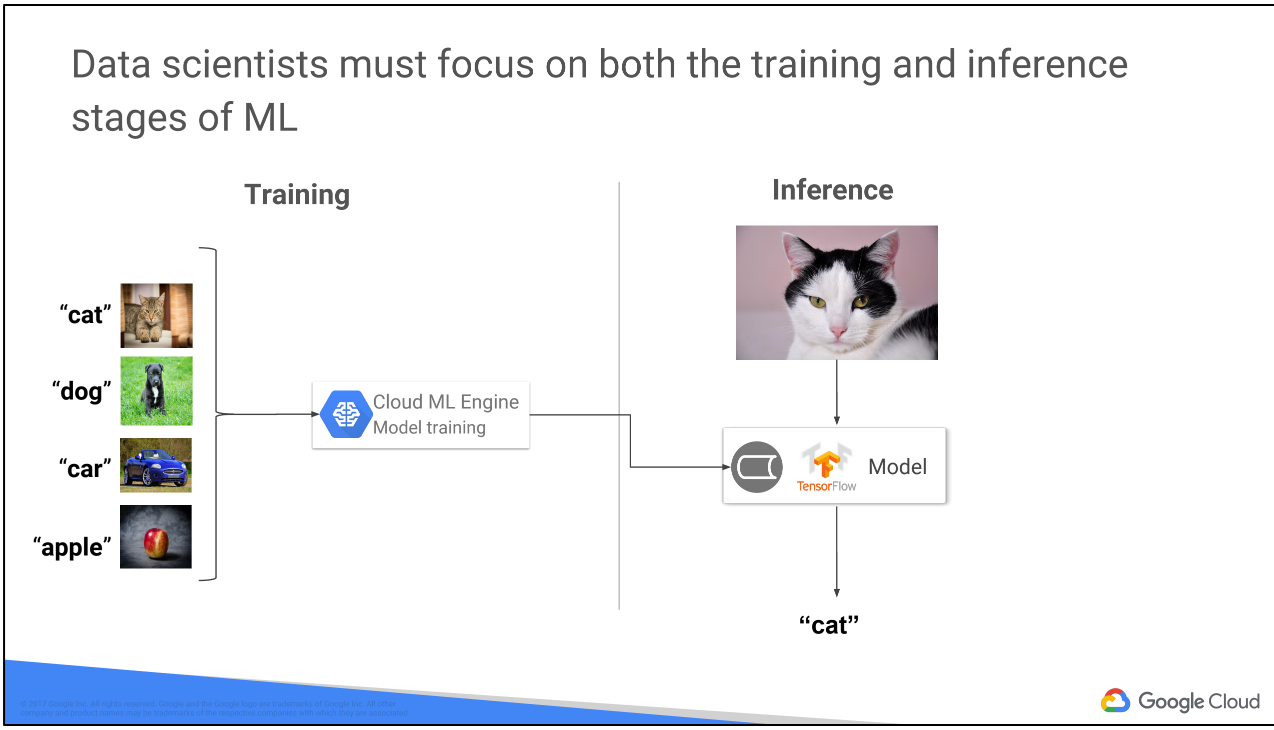
https://pixabay.com/en/domestic-cat-cat-adidas-relaxed-726989/ (cc0)
https://pixabay.com/en/dog-young-dog-puppy-280332/ (cc0)
https://pixabay.com/en/sports-car-vehicle-transportation-1317645/ (cc0)
https://pixabay.com/en/apple-education-school-knowledge-256268/ (cc0)



(Lak)
And after the model is trained, we can use it to "predict" the label of images that it has never seen before.
Here, we are inputting to the trained model this image and because the network has been trained, it is correctly able to output "cat".
Note that the cat image on this slide is different from the one before it -- still works because the ML model has "generalized" from the specific examples of cat images we showed it to a more general idea of what a cat is and what it looks like.

The key to making a ML model generalize is data and lots and lots of it. Having labeled data is a precondition for successful ML.

https://pixabay.com/en/cats-a-normal-cat-pet-796437/ (cc0)



It is important to realize that machine learning has two stages: training and inference.
Sometimes, people refer to “prediction” as “inference” because “prediction” seems to imply a future state and in the case of images like this, we are not really “predicting” that it is a cat, just “inferring” that it is a cat based on the pixel data.

It can be tempting, as a data scientist, to focus all your energy, on the first stage. But this is not enough.
You need to be able to operationalize the model, put the model into production so that you can run inferences.

Look at many books on machine learning, blog posts, university courses ... They tend to ignore this second stage of ML. But in the real-world, what is the use of training a ML model if you can not use it?
In this specialization, we will be careful to show you machine learning, end-to-end. And by end-to-end, we mean putting ML models into production.

https://pixabay.com/en/cat-cute-cat-baby-kitten-pet-1992140/(cc0)
https://pixabay.com/en/mouse-rodent-cute-mammal-nager-1708347/

Course 1: How Google does ML

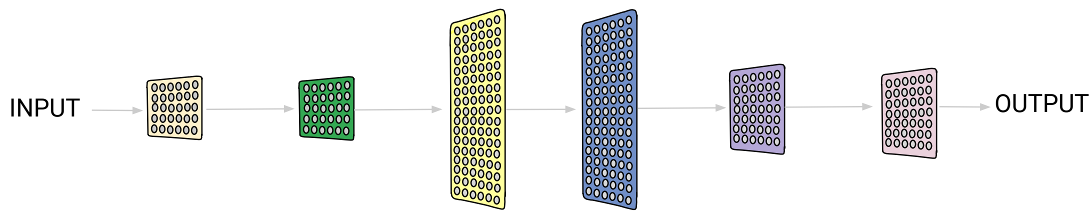
Module 2: What it means to be AI-first

Lesson Title: **ML in Google products**

Format: Screencast

Google Cloud

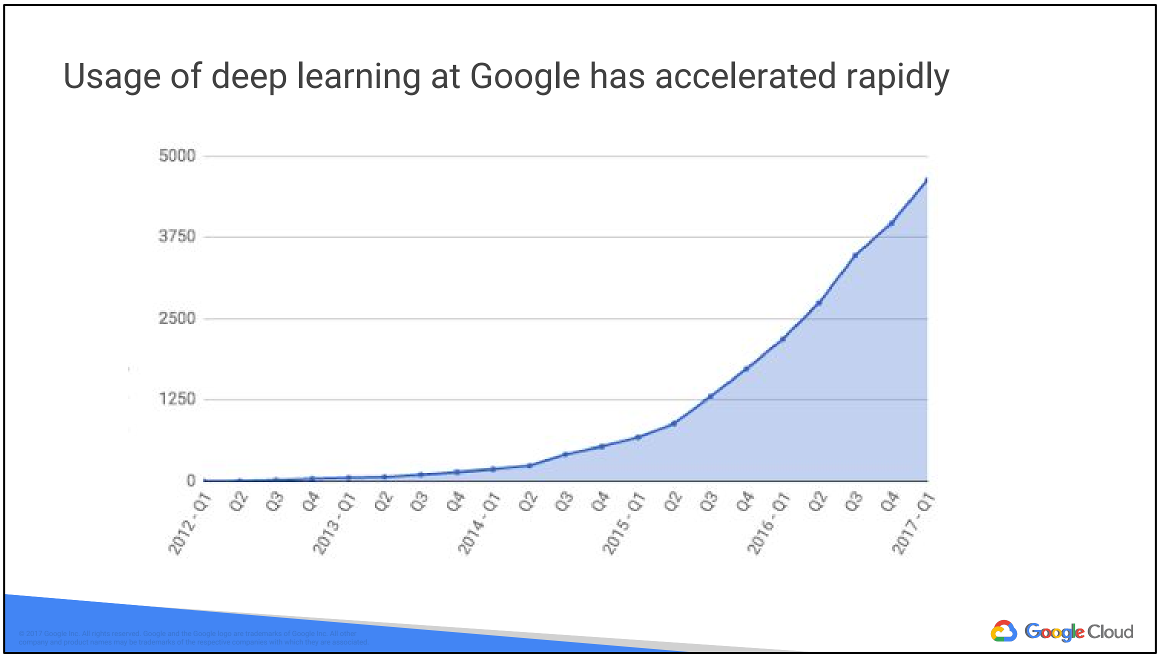
Neural networks is one important technology we use



In the previous slides, I drew the mathematical model in a specific form. The model consists of many layers, arranged one after the other. The input passes through the first layer, then the second, etc. with each of the layers itself being a simple mathematical function. So, the entire model consists of a function of a function of a function of … you get the idea.

The diagram depicts a mathematical model called a neural network. There are other common mathematical models used in machine learning: linear methods, decision trees, radial basis functions, ensembles of trees, radial basis functions followed by linear methods, the list goes on. But we were talking about NNs.

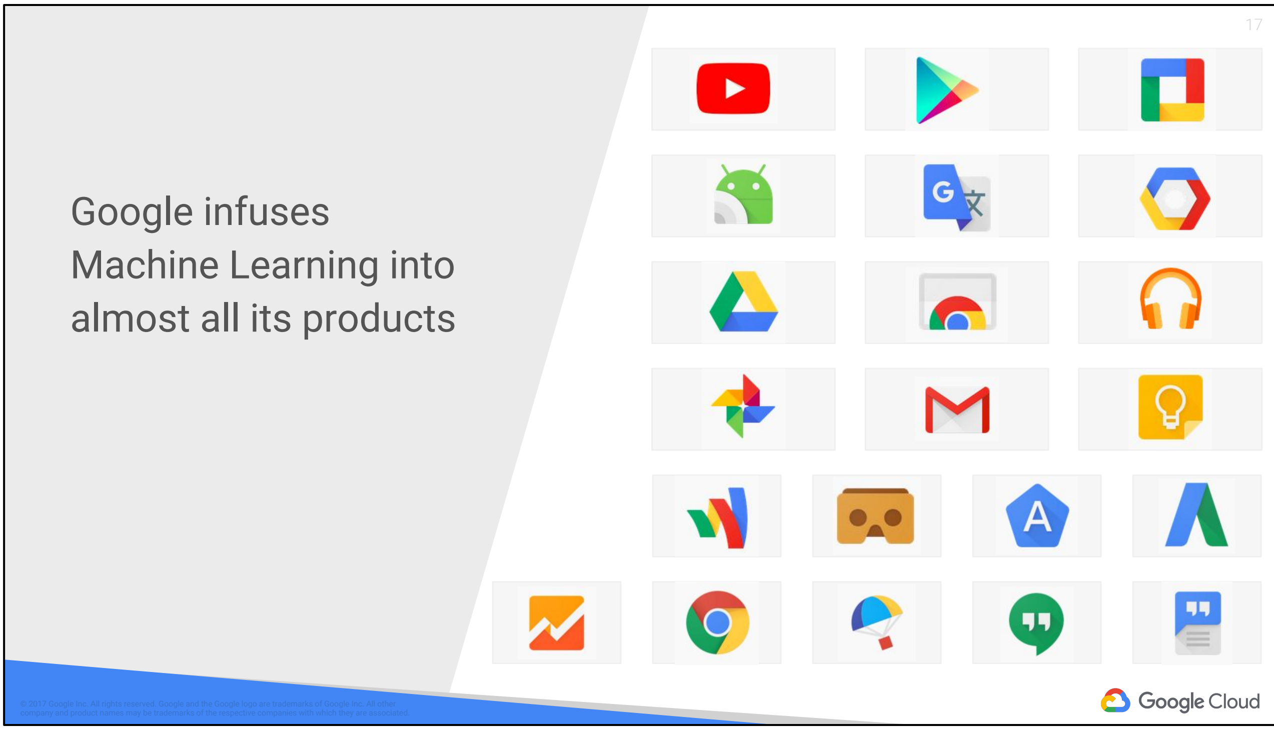
Traditionally, neural network models did not have this many layers. Neural networks date back to the 1970s, but they used to have only one hidden layer. The reason had to do with (1) computational power: training deep neural networks, NNs with lots of layers, takes a lot of computing power; (2) availability of data. As you add more layers, there are more and more weights to adjust and so you need lots and lots more data; and (3) computational tricks. It turns out that if you just add layers, you will run into some issues. The NNs will take a long time to train. Some of the layers will become all zero. Or they will blow up and become all NaN (not a number). So, the research community need to develop a number of tricks and techniques to get deep neural networks to work.



And so in the last few years, neural networks have proven themselves to be the best or near-best in a wide variety of tasks, even tasks that used to be thought to be unsolvable with machine learning. NNs have enabled dramatic improvements in really hard ML problems like language translation, image classification, speech understanding, etc. And they work just as well or better on structured data problems as traditional ML methods such as support vector machines or boosted/bagged decision trees.
You can see this at Google … the use of deep learning at Google has accelerated rapidly. We had pretty much no deep learning models four years ago, and now we have more than 4000 deep learning models within Google.

So, we will use neural networks exclusively in this specialization. We will start off on structured data problems, and once we know how to build an end-to-end pipeline, we'll take that knowledge and show you how to do image problems, and sequence problems, and recommendation systems.

But, look again at this graph. 4000+ models? How can there be so many ML models?



[highlight the icons mentioned as they are mentioned]

Well, ML is part of pretty much every Google product out there. Whether it is YouTube, or Play, or Chrome, or Gmail, or Hangouts … they all use ML. It is not that there is just one ML model at YouTube. There are dozens of ML models per product.

In my experience, this is something that takes getting used to. You might look at a business problem, say, how to forecast whether an item will go out of stock and think of it as a single machine learning model that you have to build. In practice, you will have to build many ML models to solve this problem. You might have to break this problem down into smaller problems based on your knowledge of the business.
For example, your first model might be to predict demand for the product at the store location. Your second model might predict the inventory of this item at your supplier's warehouses and nearby stores. You might need a third model to predict how long it is going to take them to stock your product and use this to predict which supplier you will ask to refill the shelf, and when. Of course, all these models themselves might be more complex. The model to predict the demand for milk is going to be very different from the model to predict the demand for dry noodles. The model for restocking electronics is very different from the model for restocking furniture.

There is not one ML model; there are dozens of ML models per product. This being a teaching course, we will show you how to train, deploy and predict with a single model. In practice, though, you will be building many ML models to solve a use case. Avoid the trap of thinking of building monolithic, one-model-solves-the-whole-problem solutions.

~~But back to the Google experience, we have 4000+ models; you probably have used or heard of some of them.~~

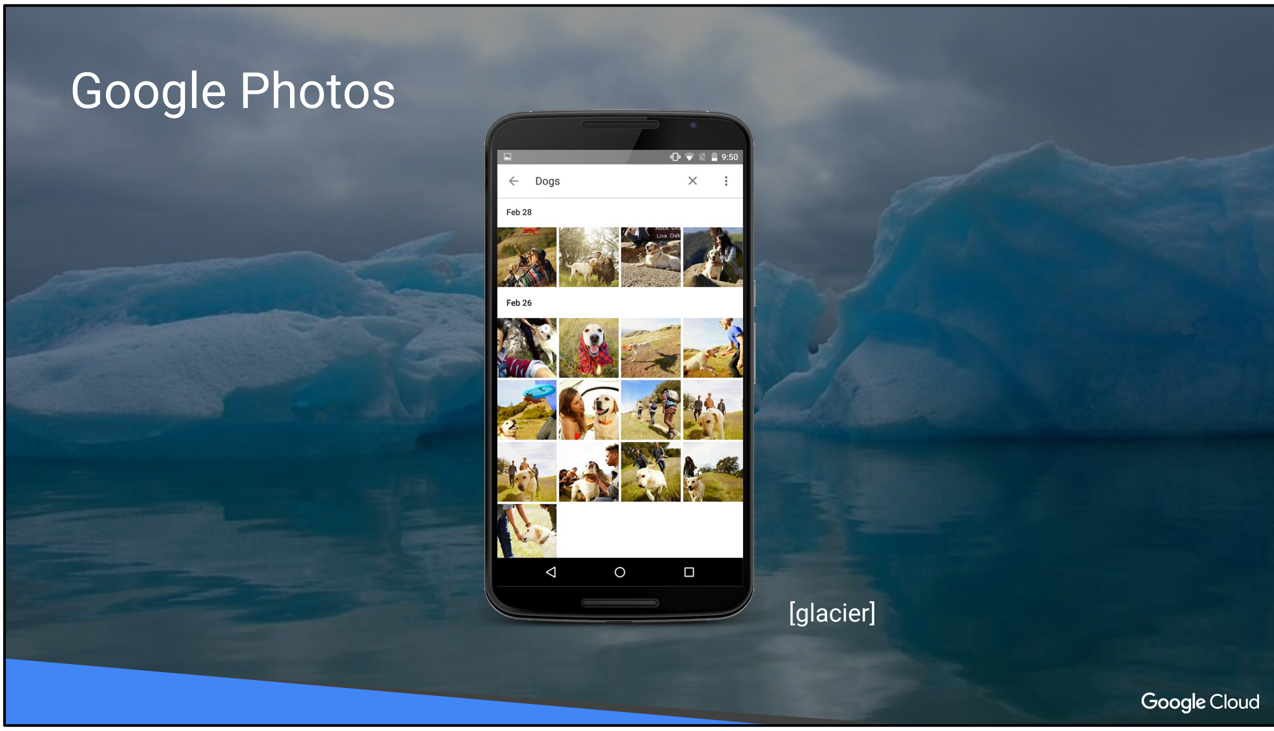
Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **Demo: ML in Google products**

Format: Screencast

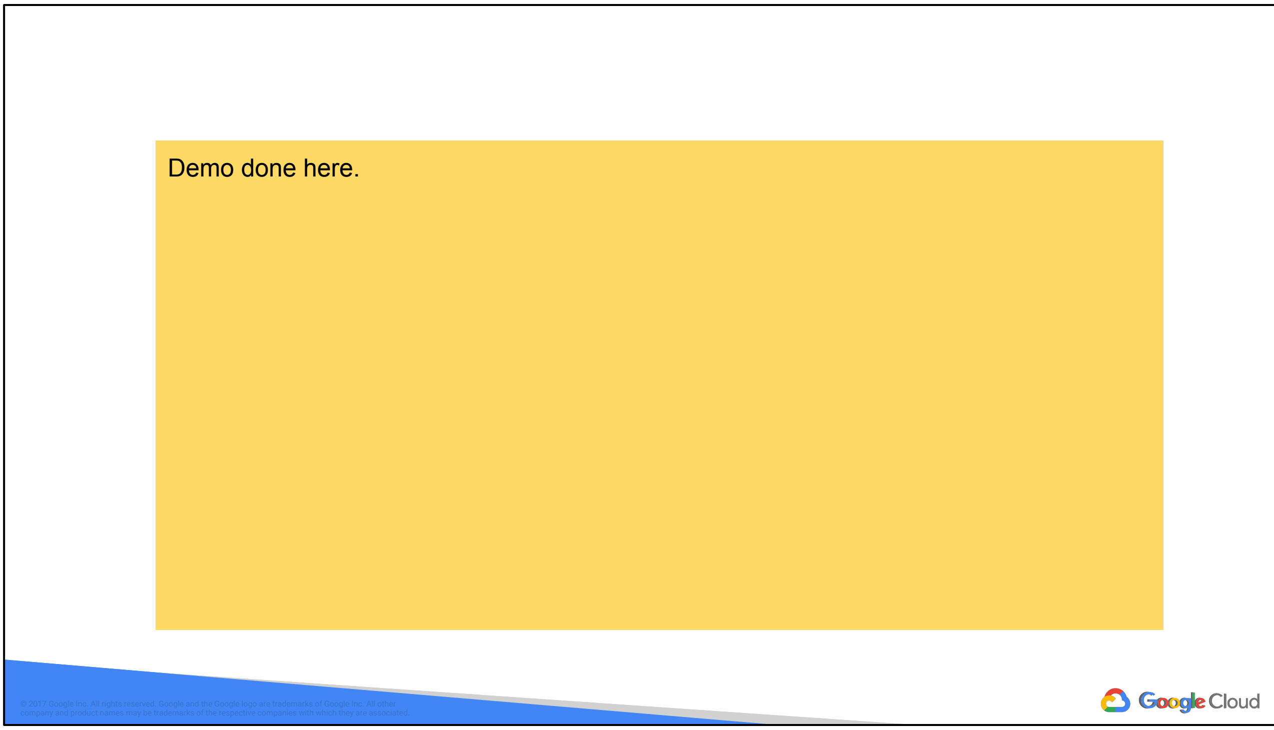
Google Cloud



But back to the Google experience … you have probably used one or more of our 4000 models …

I talked about how deep learning has come a long way in just the past few years, and nothing illustrates that better than Google Photos.
This is the Google product where you can upload photos from your camera to the cloud. You don't need to tag it -- the ML software tags images for you, so that you can then find images …

Let me show you.

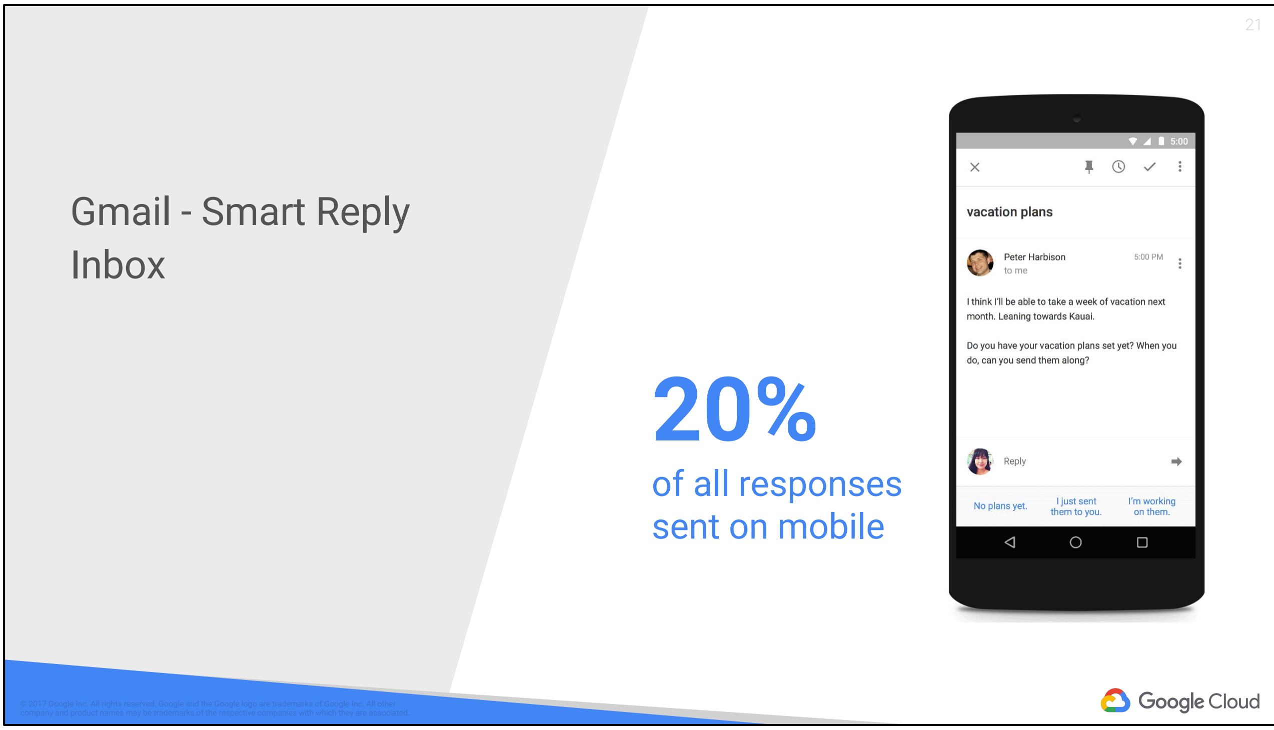


[demo]
I will go to photos.google.com and type in “Ladakh” -- this is one of the most beautiful parts of India. It’s on the Tibetan plateau. And you get pictures of Leh valley and of a monastery that I took in 2011 … and even, somehow, Himalayan peaks that I took from the plane as we were landing ...

The Google Translate app lets you point a phone camera at a street sign and it translates the sign for you.

This is a good example of a combination of several models that is quite intuitive.

One model to find the sign.
Another model to read the sign (do OCR -- optical character recognition -- on it).
A third model to translate the sign. (maybe a third model to detect the language, and a fourth model to translate the sign).
A fifth model to superimpose translated text.
Perhaps even a sixth model to select the font to use …



Smart Reply is the feature of Inbox and Gmail where the email program suggests three possible responses to received emails.
This is, in my view, the most sophisticated ML model in production today. Why do you think that is?
It’s a sequence-to-sequence model (takes the received email as input and generates the response as the output, and text, as we will see later in this course, is usually thought of a sequence of words).
The ML model here needs to understand a small body of text, and predict three dissimilar answers.
We will look at sequence models in the last-but-one course of this specialization. We have lots of ground to cover before we get there!

Source: go/srquality

Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **Replacing heuristics**

Format: Screencast

Google Cloud

(Lak)
We looked at machine learning and said it was a way to derive repeated predictive insights from data.
Then, we talked about the two stages of ML: a training phase where you teach the algorithm using labeled examples.
and a prediction or inference stage where you use the trained model to make inferences on new data.
We then looked a few examples of ML in action -- Photos, Translate, Smart Reply, all from Google products.

So how do you get to the point where your company is innovating like this in ML?

Our execs had a unique answer to the question of "what kinds of problems can machine learning solve"

This is Eric Schmidt, the executive chairman of the board at Google. He’s talking about the new transformation going on at Google, where we are becoming an “AI-first” company.

[read the quote]

Now, this seems strange. When you say ML to most people, they think “predictions from data”, but notice that there is nothing in Eric’s quote about data.

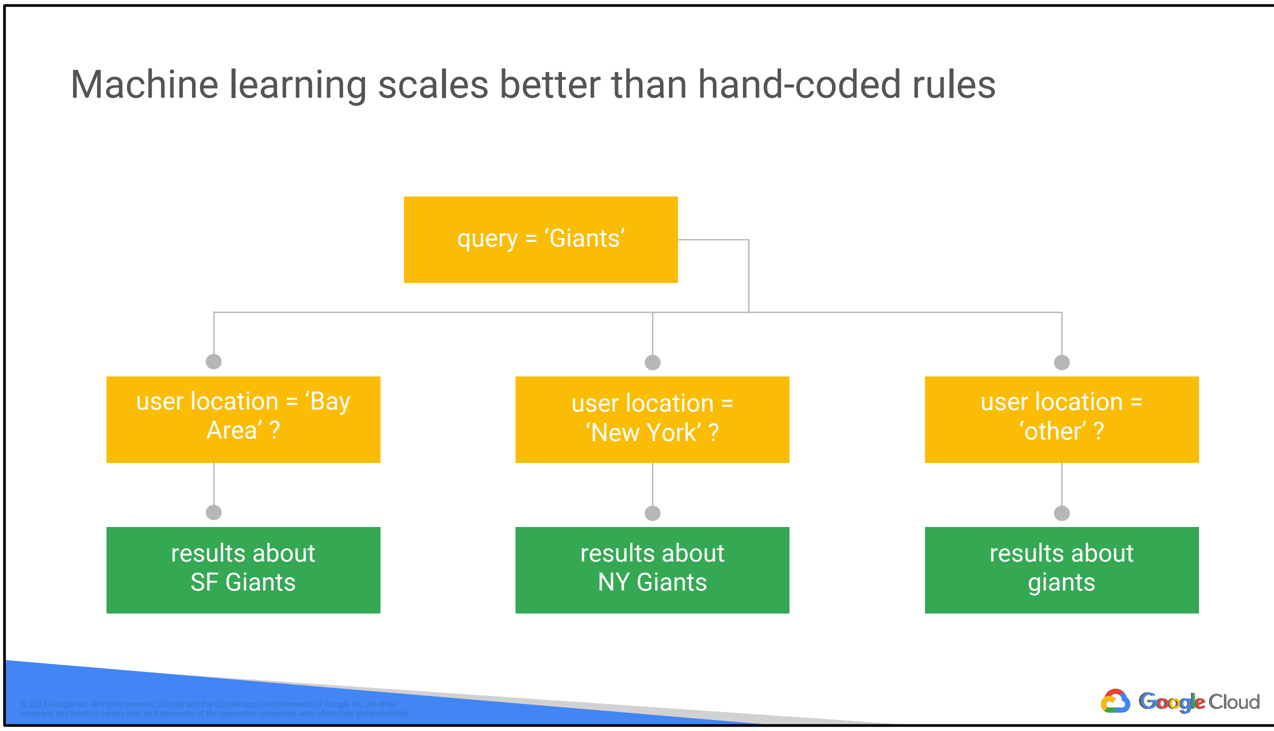
He’s talking about Machine Learning as a way to replace programming. ML, according to Eric, is about logic, not just about data.

What does he mean?

Consider search. This is, of course, our flagship application here at Google.

If you type in "giants", should we show you SF giants or NY giants?
How would you do it?

https://pixabay.com/en/baseball-ball-sport-team-batting-25761/ (cc0)
https://pixabay.com/en/california-usa-americana-2217654/ (cc0)
https://pixabay.com/en/map-new-york-state-geography-43774/ (cc0)

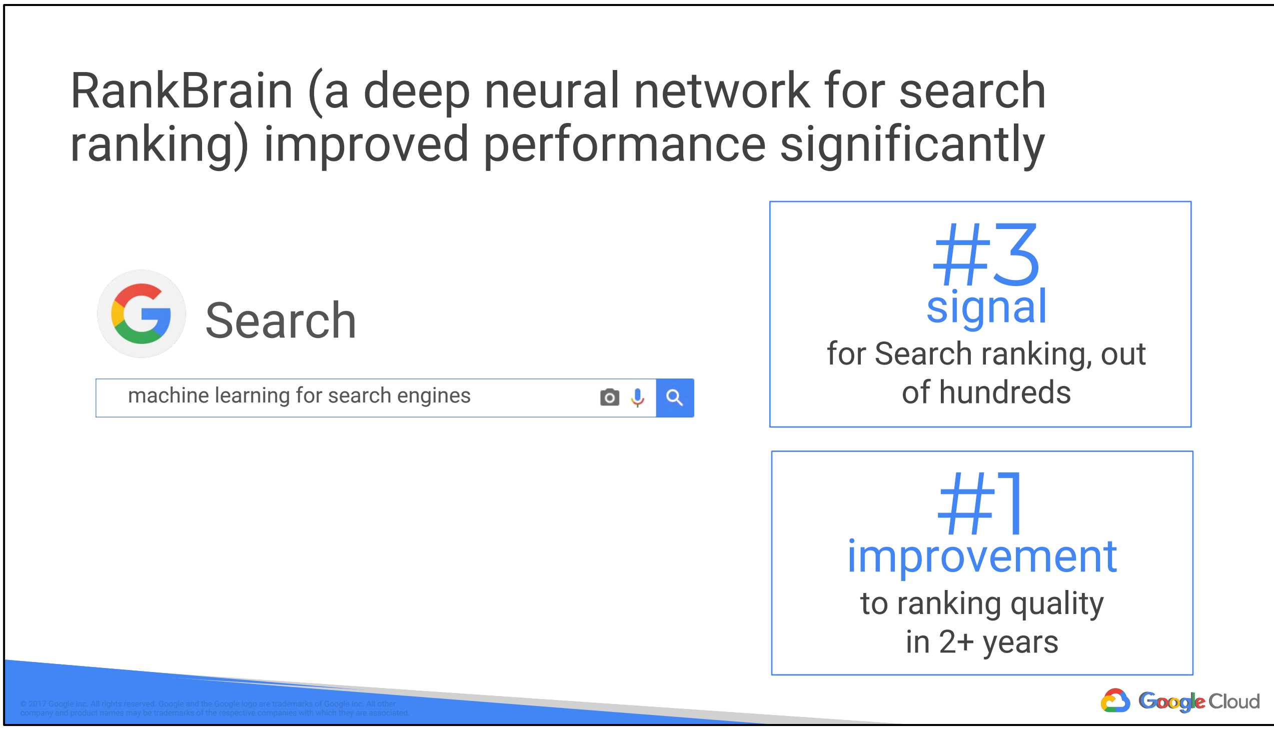


A few years ago, this is how Google search worked.
There were a bunch of rules that were part of the search engine code base to decide which sports team to show a user.
If the query is 'giants' and the user is in the bay area, show them results about san francisco giants.
If the user is in the new york area, show them results about ny giants
If they are anywhere else, show them results about tall people.

This is for just one query.
Multiply this by the large variety of queries that people make, and you can imagine how complex the whole codebase had become.
The codebase was getting unwieldy. Hand-coded rules are hard to maintain.

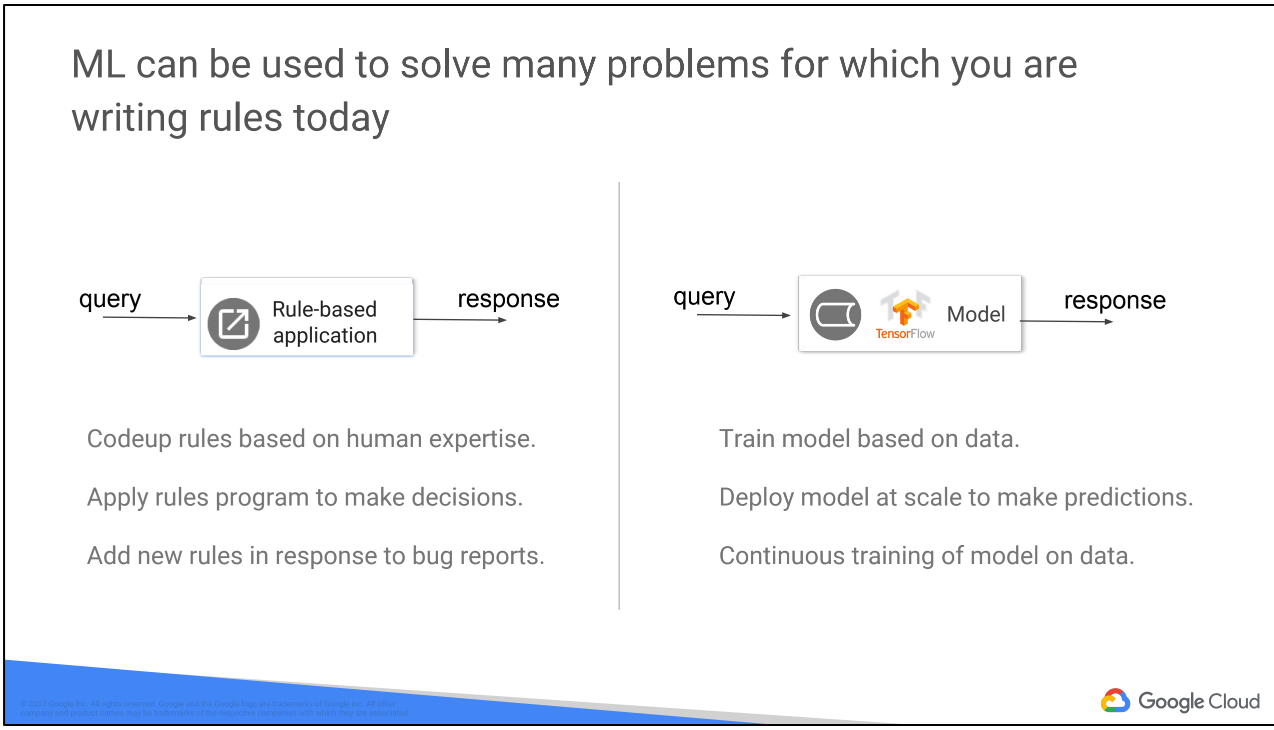
Why not try machine learning? ML scales better because it is all automated …
We knew when we showed people results, which of the links they actually clicked on … how about training a ML model so that it

could do the search ranking?



That was the essential idea behind RankBrain, our deep neural network for search ranking.
It outperformed many human-built signals … we could replace many of the hand-coded rules with ML.
The neural network ended up improving our search quality dramatically.
Plus, the system could continually improve itself based on what users actually preferred.

Replacing heuristic rules by ML. That’s what ML is about.



So, what kinds of problems can you solve with ML? Anything for which you are writing rules today. It is not just about predictive analytics! Google search is not a predictive analytics application, but we use ML for it. Note that is a far more expansive answer to "what kinds of problems can ML solve?" So, that's what we mean when we say Googe is an AI-first company. We think of machine learning as the way to scale, to automate, to personalize. Think about all the heuristic rules you are coding up today; provided you can collect the right data, you may be able to do it using machine learning.

The way you think about problems changes when you do this. You don't think about coding up rules; you think about training models based on data.
You don't think about fixing bug reports by adding new rules; you think in terms of continuously training the model as you get new data
And think of applying rules to inputs, you think in terms of deploying models at scale to make predictions.

Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **It’s all about data**

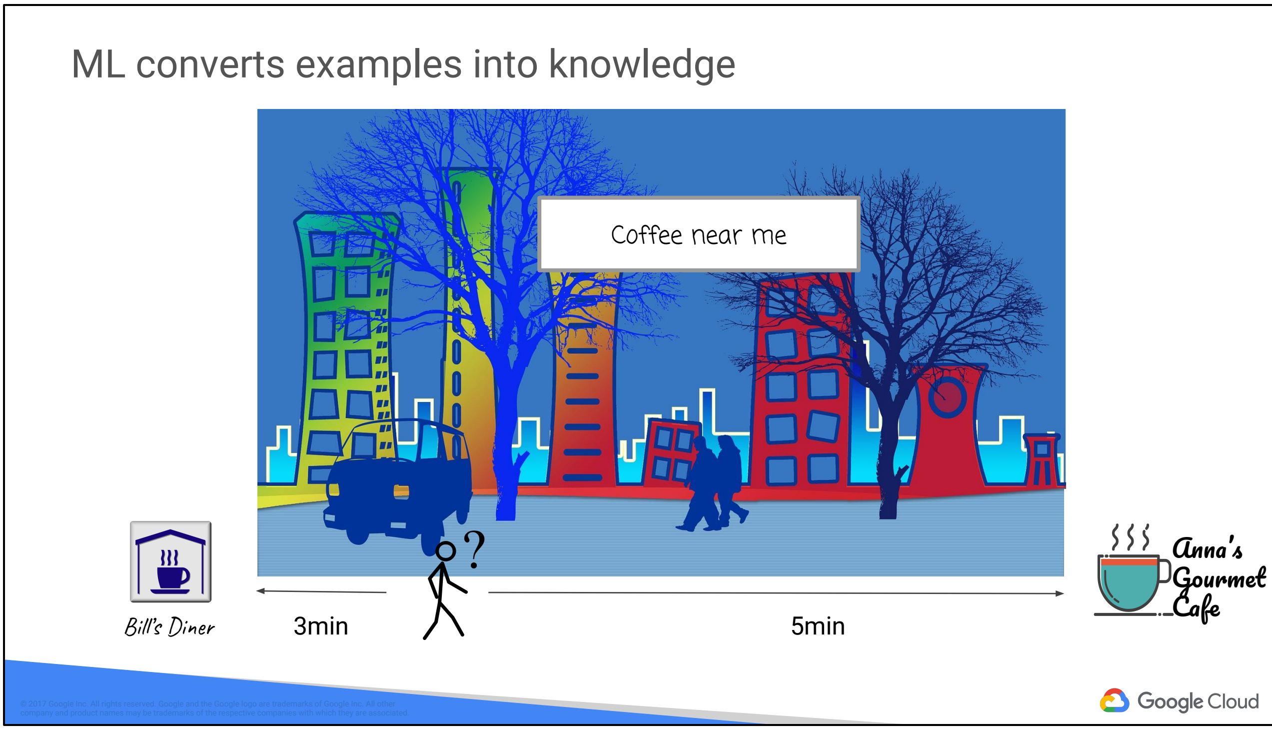
Format: Screencast

So, how does this idea change the way we approach new problems?
A few years ago, we found that certain types of queries were becoming more common.
Japanese toys in San Francisco, live lobster in Kissimmee, vegan donuts near me …

These are hard queries ...local queries … people are not looking for websites but actually businesses on a map

We could write rules for each of these, but it becomes unwieldy rather quickly. Let's see how we approach it from a ML perspective.
We start by thinking about how to collect the data to make it a ML problem.

Image (Lobster) cc0: https://pixabay.com/en/lobster-shediac-canada-2358898/
Image (Donuts) cc0: https://pixabay.com/en/candy-food-donuts-market-2696734/
Image (Beehive) cc0: https://pixabay.com/en/bees-combs-beekeeper-honeycomb-2368228/
Image (Toy) cc0: https://pixabay.com/en/panda-z-panda-toy-children-child-1299683/

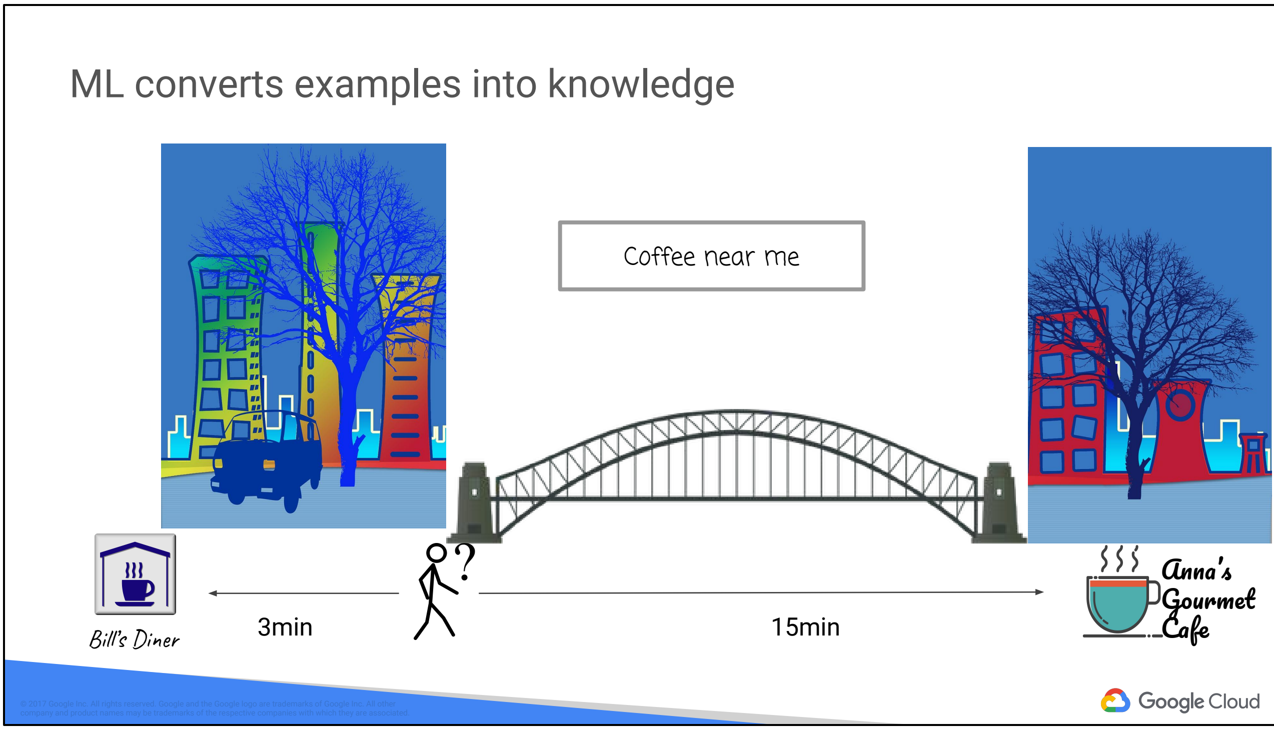


Let's look at an example. The query "coffee near me".

The idea behind ML is to take a bunch of examples and convert that knowledge into future predictions …
When you search for coffee near me, what are the examples? What is the future prediction?

The prediction end is quite straightforward. There are two options: Bill's diner carries coffee and it is only 3 minutes away.
However, there is a gourmet coffee shop just 2 minutes more … and we rather think you'd prefer the gourmet coffee to the sandwich shop.

https://pixabay.com/en/city-trees-city-view-animated-107600/ (cc0)
https://pixabay.com/en/coffee-tea-symbol-cappuccino-hotel-32355/ (cc0)
https://pixabay.com/en/cup-icon-glass-symbol-design-flat-1849083/ (cc0)

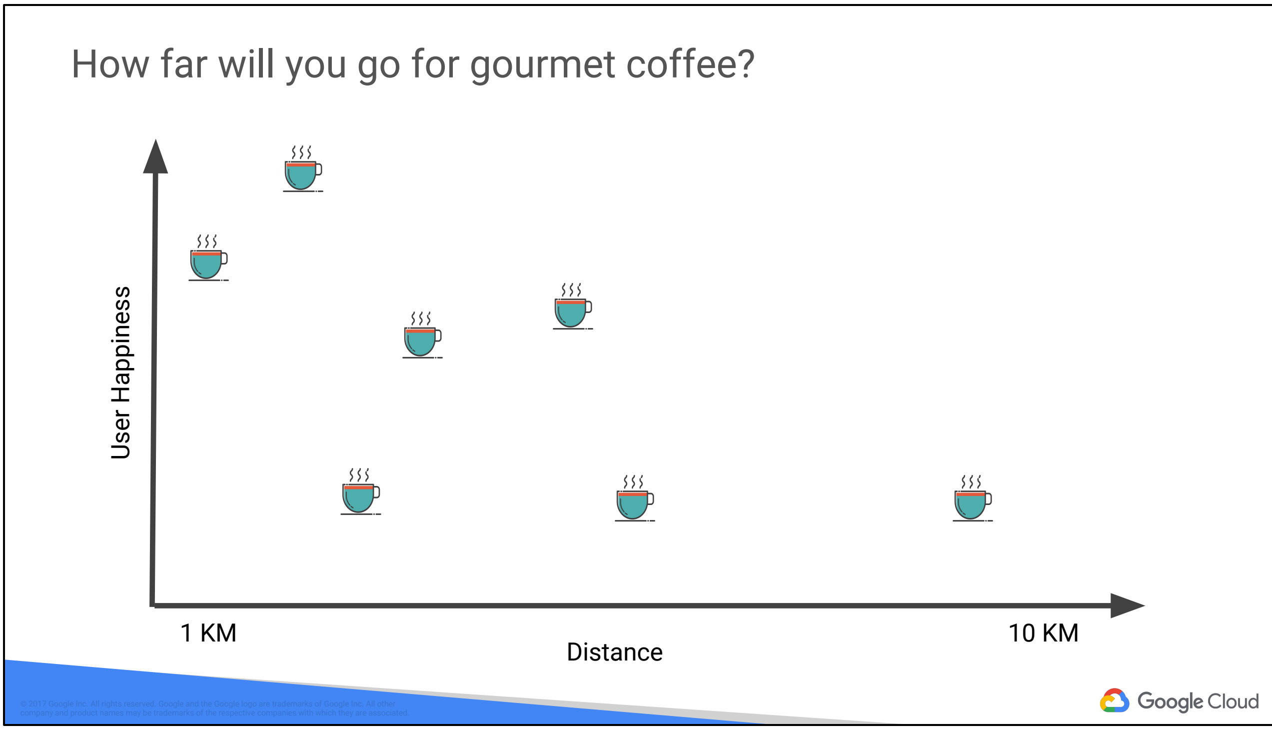


… on the other hand if the gourmet coffee shop is across the bridge, we probably will send you to the diner instead …
Or if the diner typically takes 10 minutes to serve coffee, or does not have take-away coffee (you have to sit down and eat), then perhaps the 15-minute walk is what you would prefer.

How far is too far? How much does the rating of the restaurant, and time it takes to serve you matter?

Rather than guessing and having a whole bunch of rules, we'd rather have users telling us ...

https://pixabay.com/en/architecture-city-home-homes-107595/ (cc0)
https://pixabay.com/en/city-trees-city-view-animated-107600/ (cc0)



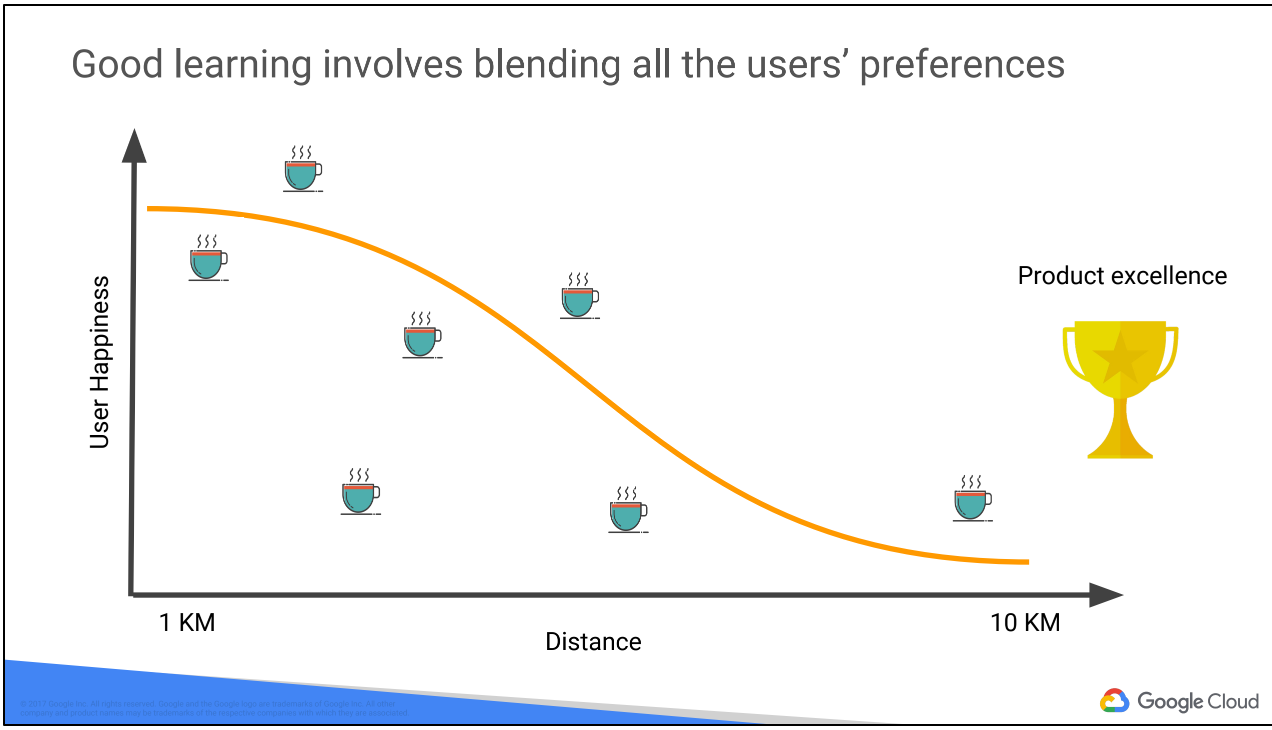
So we look at a bunch of data and do a trade off … distance vs. quality of coffee … for the other factors such as service time also, but let’s consider just the distance .. but where do you get this data?

As an AI-first company, we might start with heuristics, but we do so with the mindset that we are going to throw away the heuristics just as soon as we have enough data about user preferences ...

What we need is examples … remember example = labeled data. Here the input = distance to shop and label = does the user like the result or not?

- So we take an example of a shop 1 km away and user says great …I’ll go 1km
- And we ask another user 3km away and they say “ I don’t even like gourmet coffee.”
- We aggregate a bunch of different examples eventually when we realize when it’s far enough away, nobody wants to go …

And then we try to fit our model



Machine Learning is about collecting the appropriate data and then finding the right balance of good learning, trusting the examples

Image (Trophy) cc0: https://pixabay.com/en/cup-trophy-award-profit-2015198/

Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **Framing an ML problem**

Format: Screencast

Lab: Framing a machine learning problem

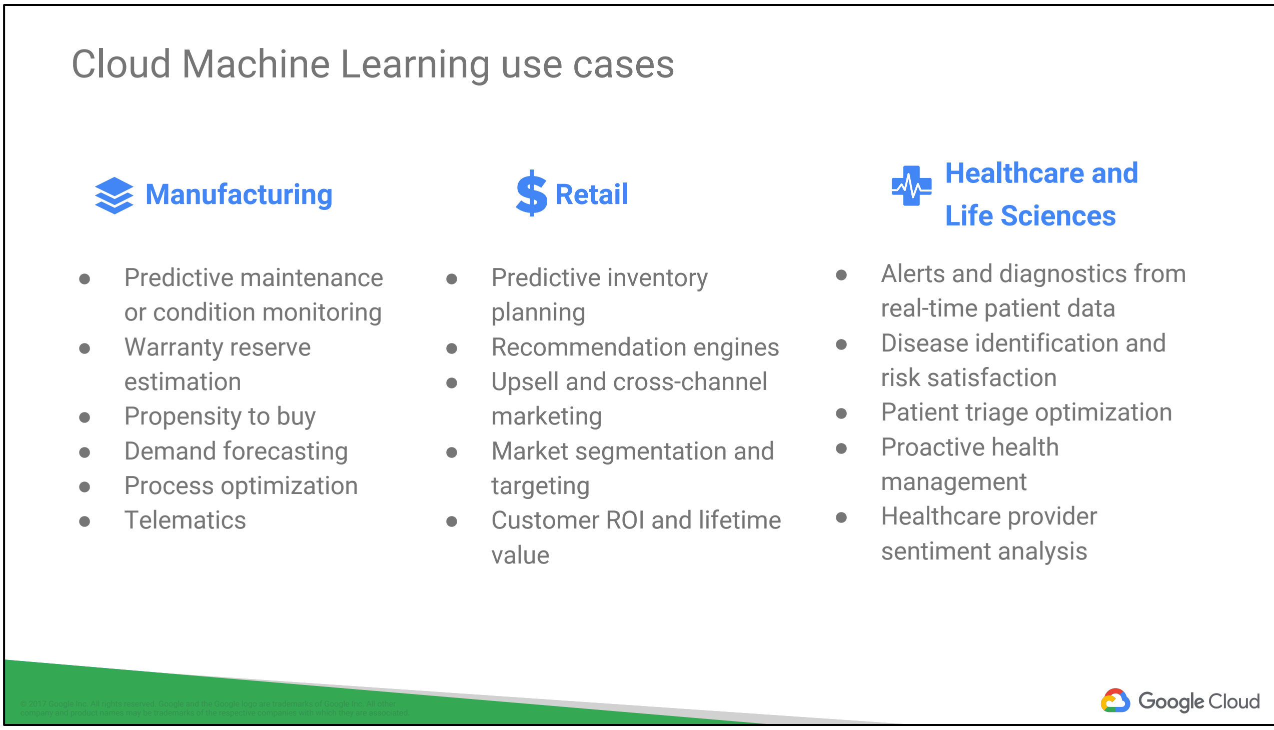
Lab: Framing a machine learning problem

In this lab activity, you will choose 2 from a set of provided use cases and think about the problems from an ML perspective

Google Cloud

Pick two use cases from the following two slides that you are familiar with:
Cast this as Machine Learning problem
What is being predicted?
What data is needed?
Cast the ML problem as a software problem
What is the API for the problem during prediction?
Who will use this service? How are they doing it today?
Now, cast it in the framework of a data problem
What are some key actions to collect, analyze, predict, and react to the data/predictions (different input features might require different actions)

Image (Frames) cc0:
https://pixabay.com/en/frame-wooden-frame-decorative-2480747/



So, here’s the first set of potential use cases. Freeze this frame to pick a problem to work on.

Cloud Machine Learning use cases (continued)

Travel and Hospitality

- Aircraft scheduling
- Dynamic pricing
- Social media – consumer feedback and interaction analysis
- Customer complaint resolution
- Traffic patterns and congestion management

Financial Services

- Risk analytics and regulation
- Customer Segmentation
- Cross-selling and up-selling
- Sales and marketing campaign management
- Credit worthiness evaluation

Energy, Feedstock and Utilities

- Power usage analytics
- Seismic data processing
- Carbon emissions and trading
- Customer-specific pricing
- Smart grid management
- Energy demand and supply optimization

Google Cloud

Here's the second …
Take time to answer the questions in the lab.
Go to the student forums and discuss your answers with other students.
When you are done, check out my answer to the questions.

Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **Framing an ML problem debrief**

Format: Screencast

Google Cloud

Example solution: Demand forecasting in manufacturing

ML problem:

What is being predicted? How many units of widgets X should we manufacture this month?

What data is needed? Historical data on # of units sold, price it was sold at, # of units returned, price of competitor product, # of units of all items that use widget X that were sold (e.g. if widget is a phone display panel, how many smartphones were sold, regardless of which display panel they carried?), economic figures (e.g. customer confidence, interest rate), this-month-last-year

Google Cloud

[debrief]

Read slide and next two ...

Example solution (as software problem)

predictDemand(widgetID, month=CurrentTime.month)

Who will use this service? Product managers, logistics managers

How are they doing it today? They examine trends of e.g. phone sales, overall economy, trade publications and make a decision

Google Cloud

Example solution, as data problem

Data problem:

Collect: economic data, competitor data, industry data, our figures

Analyze: craft features that our experts are looking at today from this data and use as inputs to model

React: automatic?

Google Cloud

Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **ML in Applications**

Format: Screencast

Google Cloud

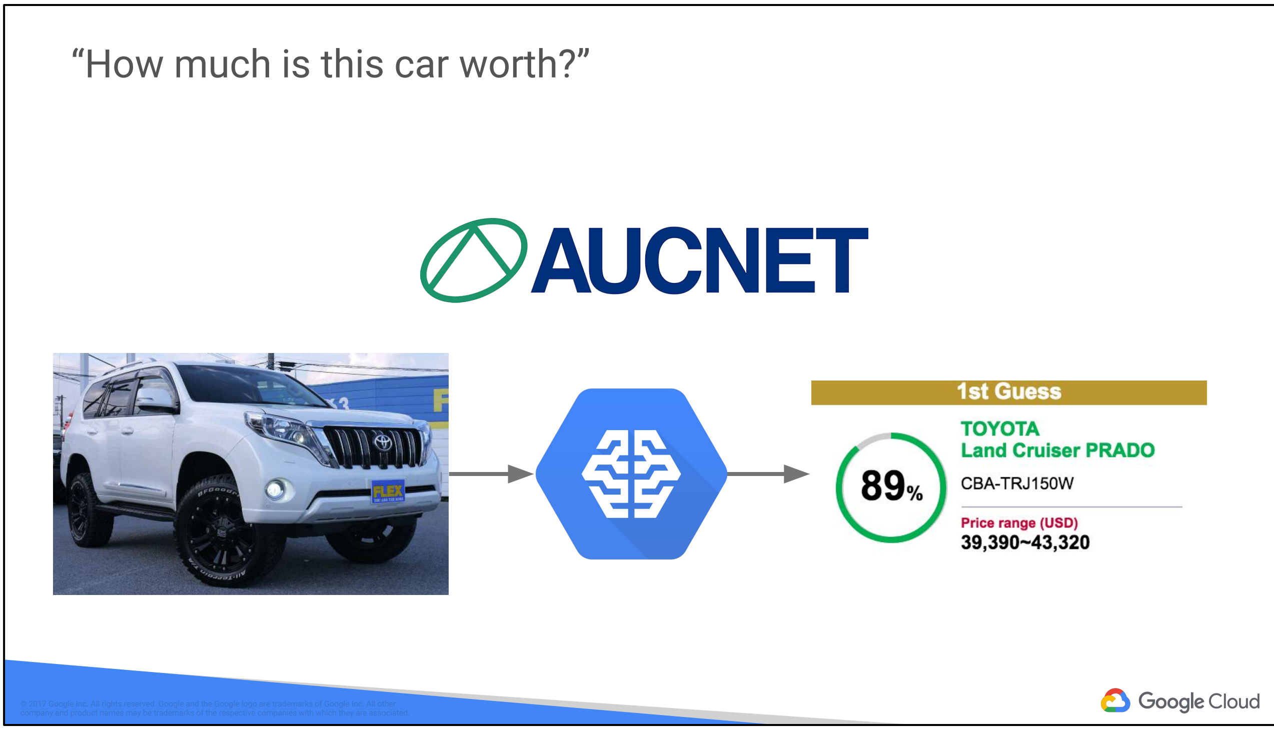
Agenda

- What is ML ?
- What kinds of problems can it solve?
- Infuse your apps with ML
- Build a data strategy around ML
- Use ML creatively, to delight your users

Google Cloud

[Lak]

An easy way to add ML to your apps is to take advantage of pre-trained models.
These are off-the-shelf solutions.
Cases that you don't need to build your own models.



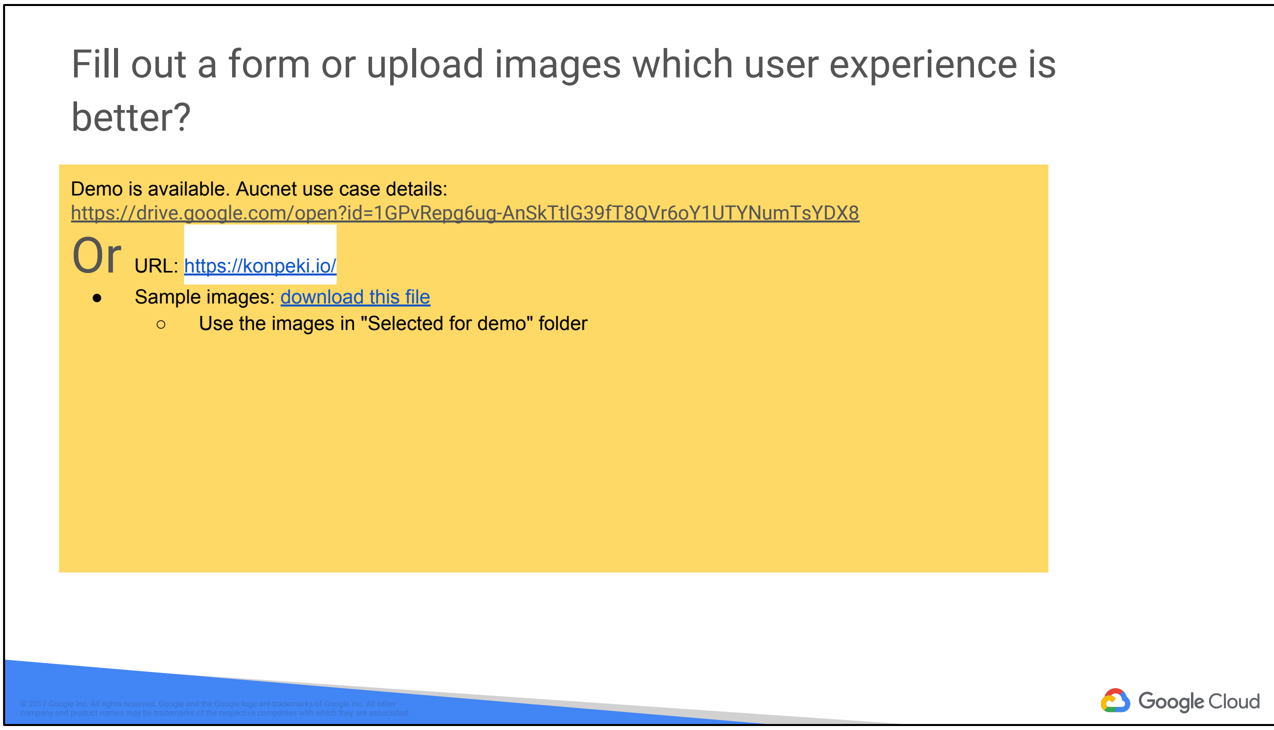
Aucnet is the largest real-time car auction service in Japan, serving over 30K dealers and running auctions worth nearly $4 billion dollars a year.
The way it used to work ... car dealers would take multiple photos of each used car to sell, upload them to the service, and need to specify what model of the car and what part of the car for every photo. It's a time consuming task for the dealers to do across thousands of photos every day.

Now … the new Machine-learning system can detect the model number of the car at high accuracy. It also can show the estimated price range for each model and recognizes what part of the car is being photographed. With this system, the car dealers just drag and drop a bunch of unclassified photos and checks if the model and parts are classified by the system correctly.

//// next slide
Let me show you ...
URL: https://konpeki.io/
Sample images: download this file
Use the images in "Selected for demo" folder

[debrief]
It used to be that you wanted to list a car, you need to fill out a form. Boring!
Aucnet said "upload images of your car" and we'll tell you how much it's worth.
Fill out a form? Upload images from your camera? Which user experience is better?

Demo is available. Aucnet use case details:
https://drive.google.com/open?id=1GPvRepg6ug-AnSkTtIG39fT8QVr6oY1UTYNumTsYDX8



Let me show you ...

URL: https://konpeki.io/

- Sample images: download this file
    - Use the images in "Selected for demo" folder
    -

[debrief]
It used to be that you wanted to list a car, you need to fill out a form. Boring!
Aucnet said “upload images of your car” and we’ll tell you how much it’s worth.
Fill out a form? Upload images from your camera? Which user experience is better?

Demo is available. Aucnet use case details:
https://drive.google.com/open?id=1GPvRepg6ug-AnSkTtlG39fT8QVr6oY1UTYNumTsYDX8

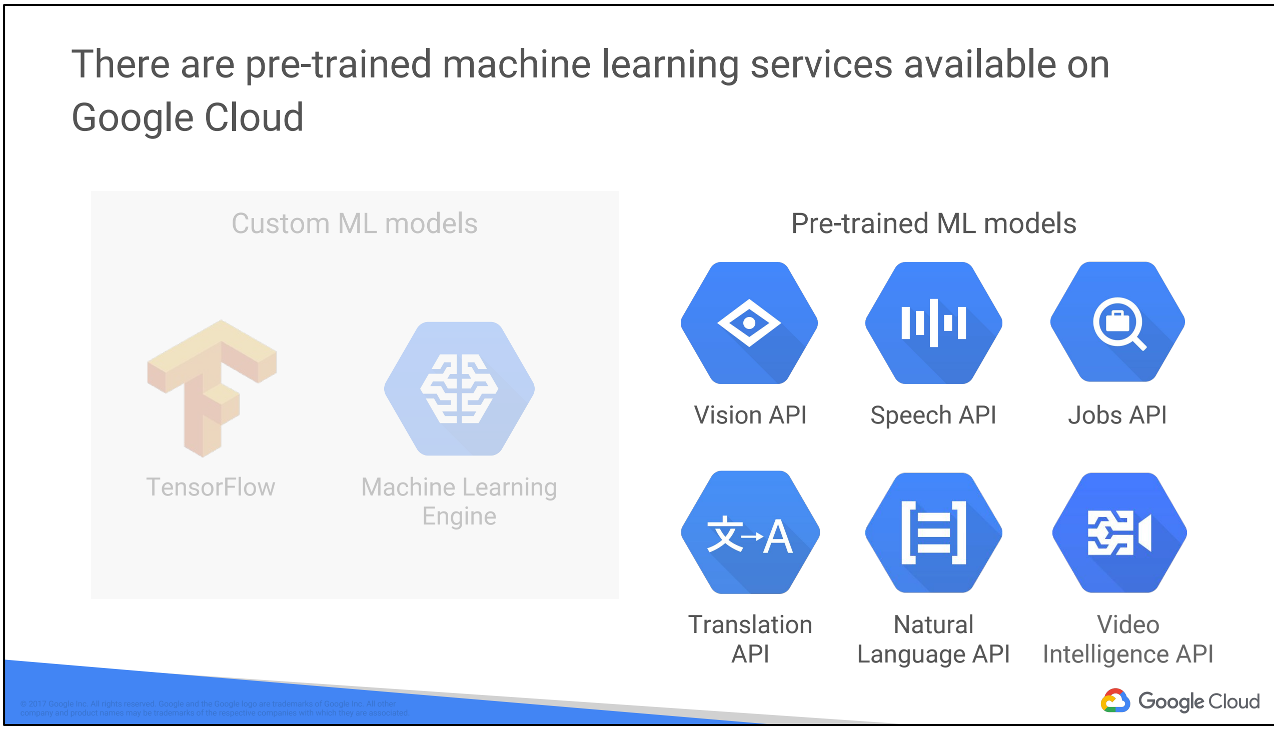
Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **Pretrained Models**

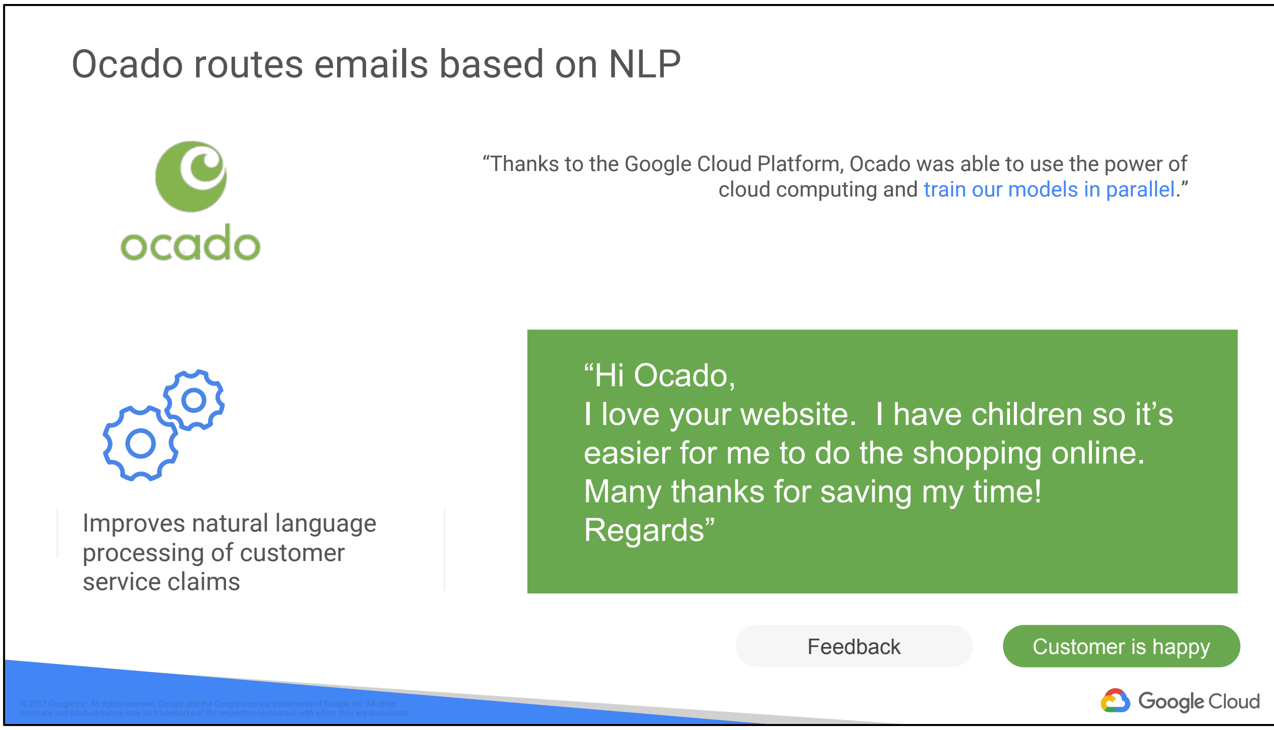
Format: Screencast

Google Cloud

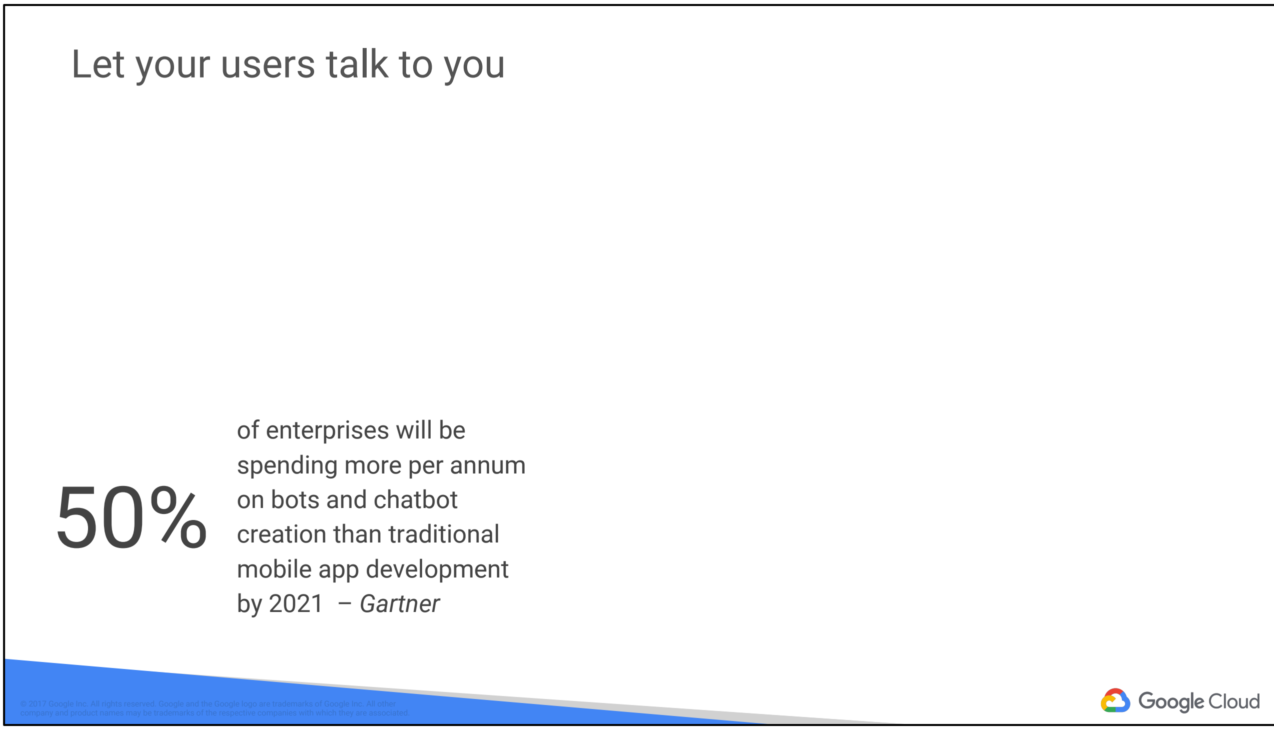


Aucnet built a custom image model on Google Cloud Platform using TensorFlow. They are on the left-hand side of this image.
But increasingly, you don’t have to do that.
There are a variety of domains where Google exposes ML services trained with our own data. For example, if you want to transcribe speech, you could use the Speech API instead of having to collect the data, train it, and predict with it. There are many such pre-trained models.

Such pretrained models are excellent ways to replace user input by ML.



Another example of using a pre-trained model … Ocado is the world's largest online-only grocery, based in the UK. Customers send email, and traditionally each email would get read and routed to the appropriate dept. That doesn't scale. So, Ocado turned to natural language processing. They were able to get sentiment, entities, parsing syntax.
The computational technology helps Ocado parse through the body of emails, tags and routes to help contact center reps determine the priority and context



But increasingly, customers do not want to go to your website and click on a button. They do not want to send you an email.
They want to talk to you. Interactively. To get their questions and concerns answered. Manually answering each call doesn't scale and so Gartner estimates that in a few years, we will be spending more on conversational interfaces than even on mobile apps!

Does this mean using the Speech API, transcribing it, and trying to make sense of it? No … what I'm showing here is a high-level conversational agent tool called Dialogflow.

Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **The ML marketplace is evolving**

Format: Screencast

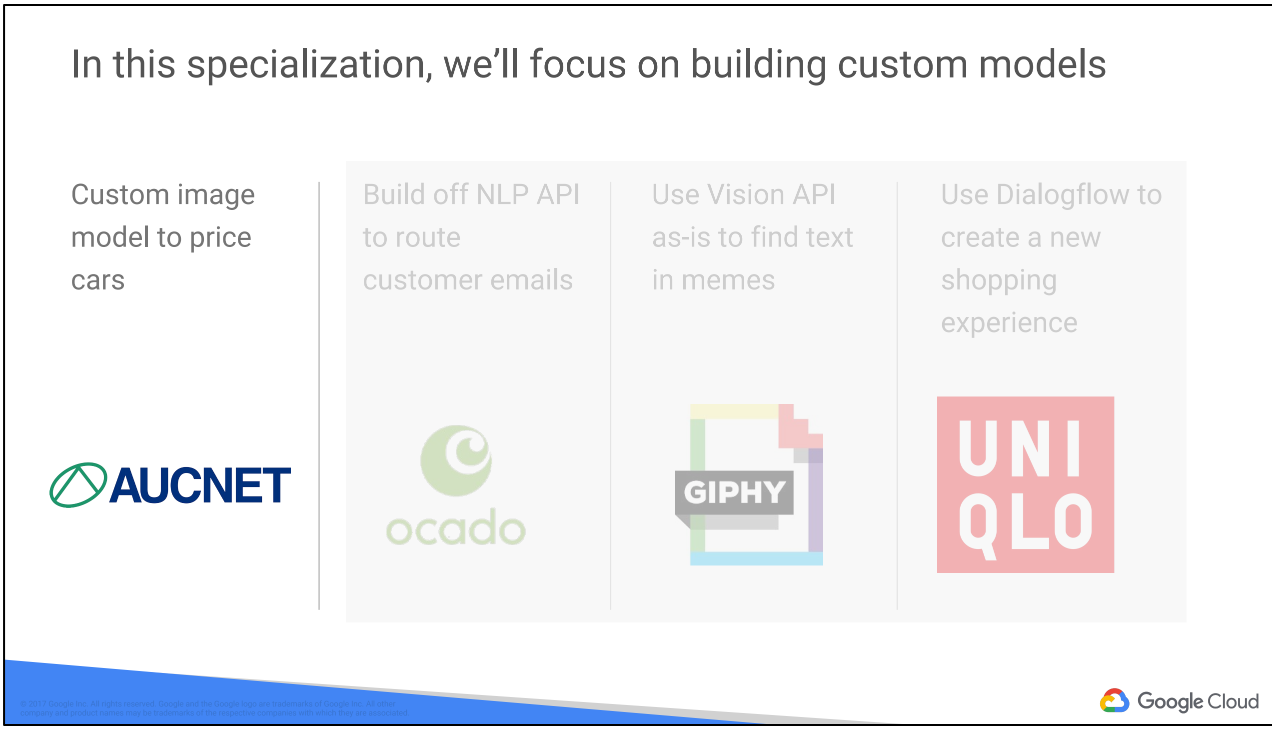
Google Cloud

The ML marketplace is moving towards increasing levels of ML abstraction

Custom image model to price cars	Build off NLP API to route customer emails	Use Vision API as-is to find text in memes	Use Dialogflow to create a new shopping experience
AUCNET	ocado	GIPHY	UNIQLO

Google Cloud

Aucnet built their own custom model to classify car parts and estimate price.
Ocado used parsed results from the NL API to route customer emails
Giphy uses the Vision API to find the text in memes using optical character recognition
The social media company used the vision api to reject inappropriate uploads.
Uniqlo designed a shopping chatbot using dialogflow.



In this specialization, we will teach you how to do machine learning, and the level of abstraction we will work at will be around building custom models. But be aware that, increasingly, you will get to incorporate machine learning into your applications primarily in the form of APIs.

Of course, someone will have to build these APIs for this marketplace, and perhaps you are the builder of such ML APIs.

Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **A data strategy**

Format: Screencast

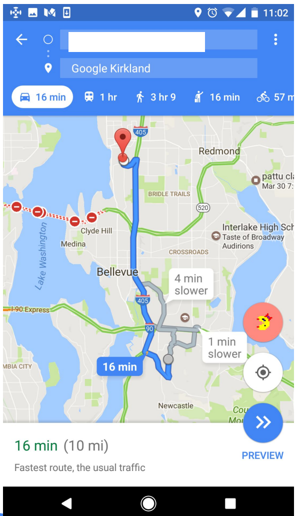
Agenda

- What is ML ?
- What kinds of problems can it solve?
- Infuse your apps with ML
- Build a data strategy around ML
- Use ML creatively, to delight your users

Google Cloud

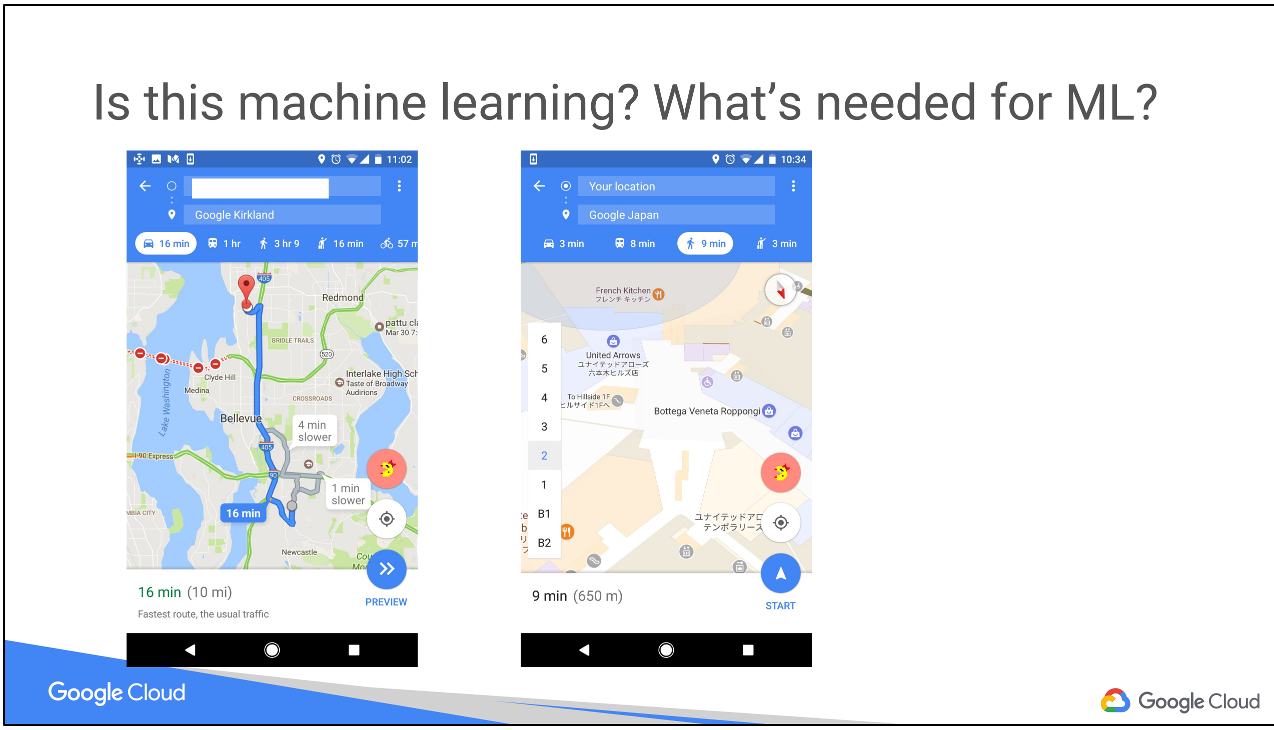
Reimagine what data can help you do

Is this machine learning? What's needed for ML?



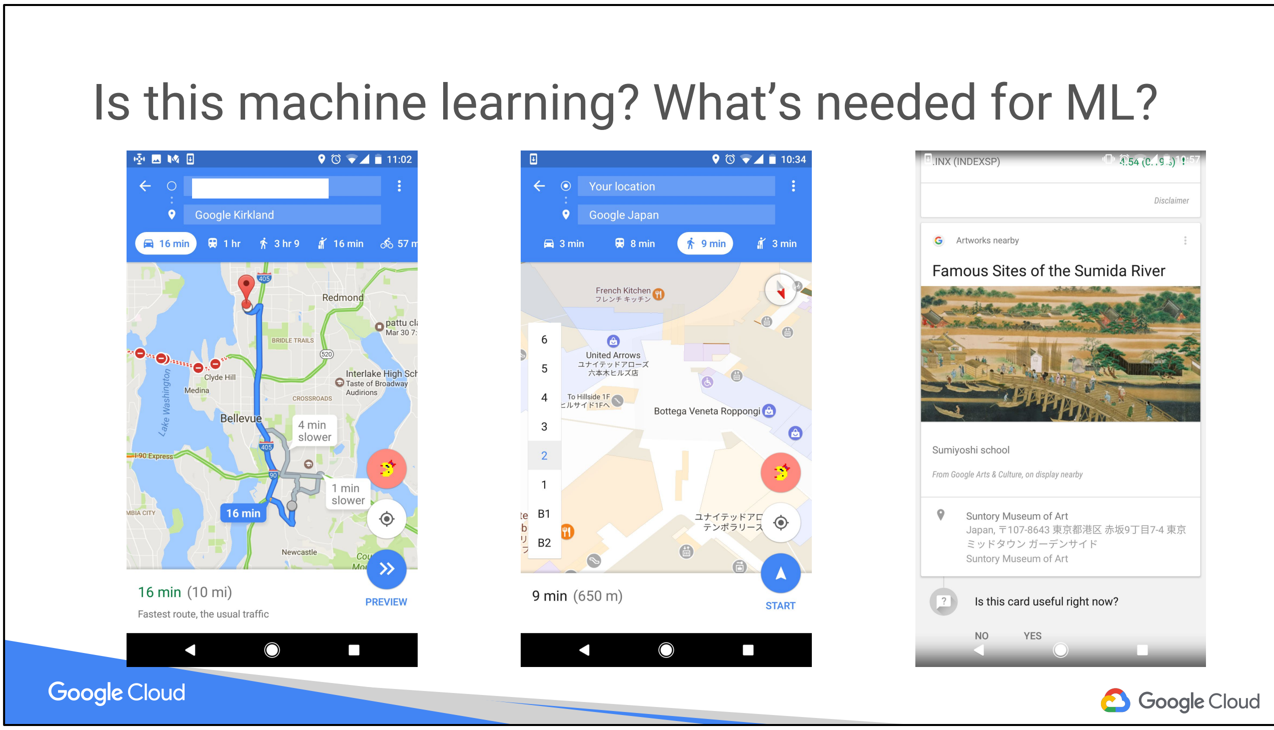
I'm going to be using Google Maps to illustrate several key points.
Take the this map for example. Every morning, I glance at my phone and it tells me the way to get to work. There are three possible routes, and today, the highlighted route is the fastest. Sometimes, I do go to Google Seattle, crossing the floating bridge across Lake Washington and Maps tells me, helpfully, that the bridge is closed. Is this machine learning?
You could think of this as being just a set of rules. Google has to collect a lot of data to make this use case possible -- where the roads are, for one. The traffic on each road. Bridge closures.

The algorithm itself? Routing algorithms between A and B subject to a set of constraints? The A* algorithm is taught in undergraduate computer science classes. So, not that complex once you have the data. This is the kind of thing you can do for countries at a time. Get data on the road network, and provide routing directions. Traffic and bridge closures are a little more difficult in that you have to work with a bunch of smaller government entities, but still not such a huge data problem. The logic, once you have the data, seems quite tractable.



Now, take the case in the middle. Still Maps. I was in Japan, making my way from my hotel to the Google office. I'm in a subway station called Roppongi and Maps tells me that I'm on floor #2.
How does it know? Whatever the data sources it uses -- wifi points, barometric pressure, typical walking speed … it's pretty obvious that this can not be a simple set of rules. Plus, of course, the relevant data to train the model, and the relevant data to keep the model remaining fresh. Once you have the data, you are now going to use ML to sidestep having to write the logic.

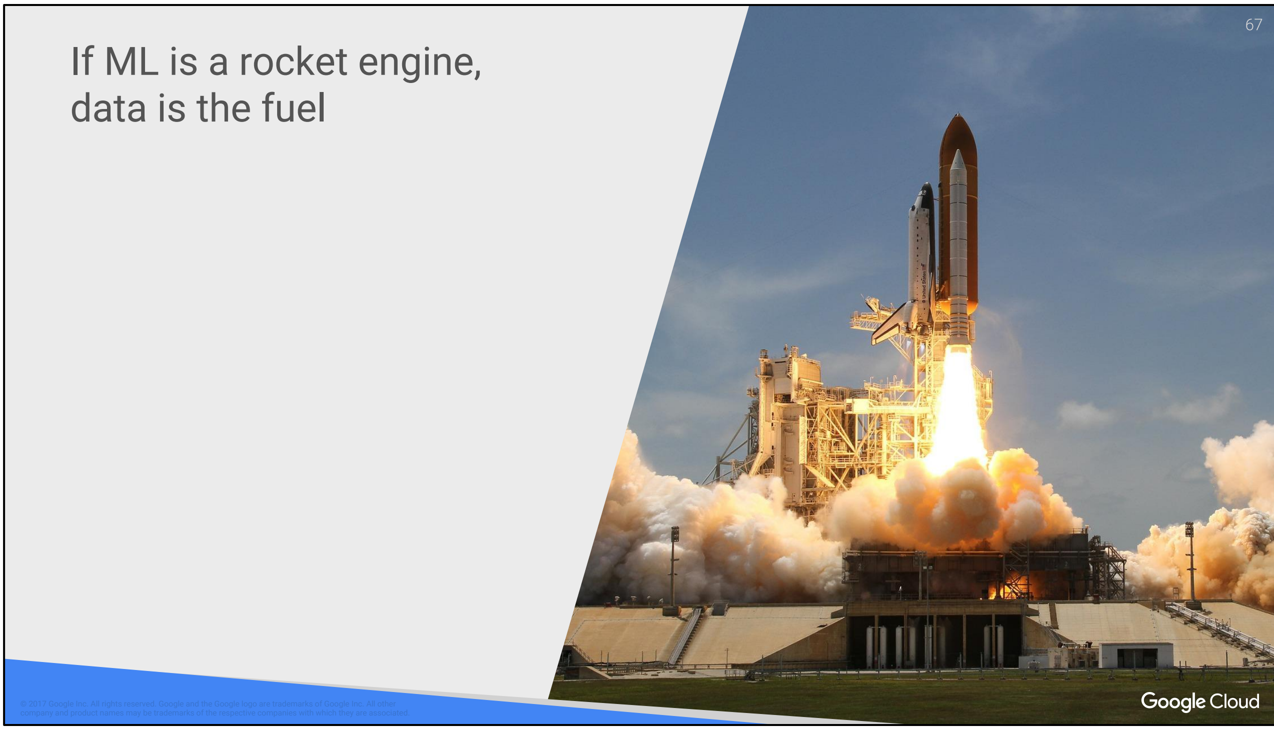
Maps here is anticipating information that you might want to know if you were in a multi-story building … what else can Maps anticipate?



Take the map on the right. Still in Japan, I glanced at my phone in between meetings and noticed that I was getting a recommendation. Maps is now connecting my past history that I like art, that I like museums, and that I am in Japan to now recommend things to me. This is even more of a data problem. The ML is what allows the original limited how-to-get-from-A-B map to now become a virtual assistant.
Personalization of the Maps service is possible only with machine learning.

So, machine learning is about scaling beyond hand-written rules. But then you start being able to do things that you could never achieve if you were writing hand-written rules.
Think back at your business. Your business analysts are essentially looking only at the bulk of your business. That’s akin to the use case on the left. The stuff that everybody in the country needs. One set of rules for everyone. You might be thinking of ML as a way to do the things in the middle -- of being able to take the data you happen to have and training a ML model.
But think of ML as a way to get to kind of things on the right -- of being able to personalize your services for each of your customers. And notice the question at the bottom of the card on the right -- asking for user feedback to keep improving the model.

What’s needed, though, in this transformation from the left (which is quite generic) to the right (which is quite personalized)?



Data, and lots of it. The rules/models are actually quite simple.
If ML is a rocket engine, data is the fuel
As we get into complex models and various ways of tuning a model to get better and better performance, it can be very easy to lose sight of a key point.
Data wins. Every time.

https://pixabay.com/en/rocket-launch-smoke-rocket-take-off-67723/ (cc0)

Simple ML and More Data > Fancy ML and Small Data

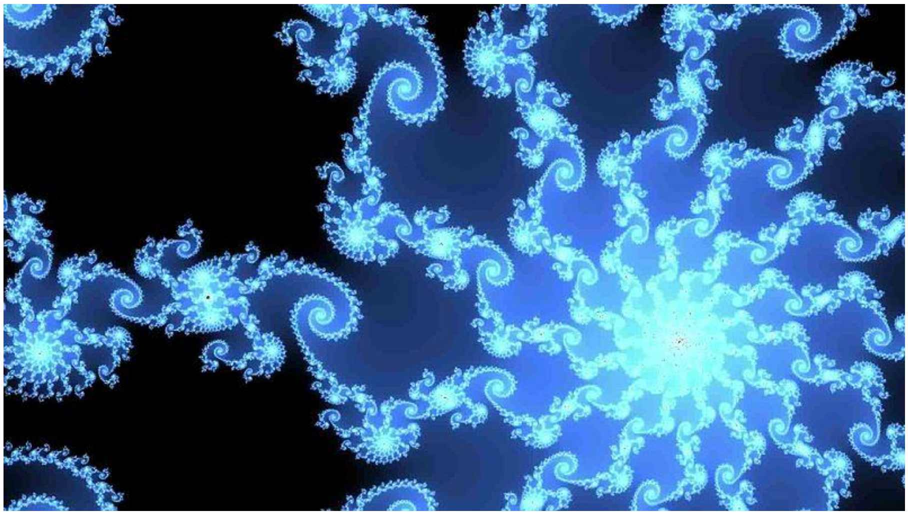


Image Source: https://en.wikipedia.org/wiki/Mandelbrot_se

Google Cloud

Given the choice between more data and more complex models, spend your energy collecting more data.
By that I mean, collecting not just more quantity. Also more variety.

For example, imagine that your data consists of these fractals. If you are zoomed way in, you won't see the patterns. You don't have enough data, so you will end up fitting to very complex rules. As you get more and more data, hopefully you fill out the domain and the overall pattern starts to become much more evident.

A ML strategy is, first and foremost, a data strategy.

From: https://en.wikipedia.org/wiki/Mandelbrot_set

Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **Training-serving skew**

Format: Screencast

So, how do you get started on machine learning? In our experience, we have seen that the typical customer journey, the one most likely to be successful, is to select a use case where you are doing manual data analysis today. This is what Global Fishing Watch, a nonprofit that tries to identify poaching did. They used to manually analyze fishing trips and they scaled up their processing using machine learning to the point that they could analyze 22m data points daily.

There are several reasons why you want to go through manual data analysis to machine learning:

(1) If you are doing manual data analysis, you probably have the data already. Collecting data is often the longest and hardest part of a ML project, and the one most likely to fail. If you have the data already, your chances of success just went up.

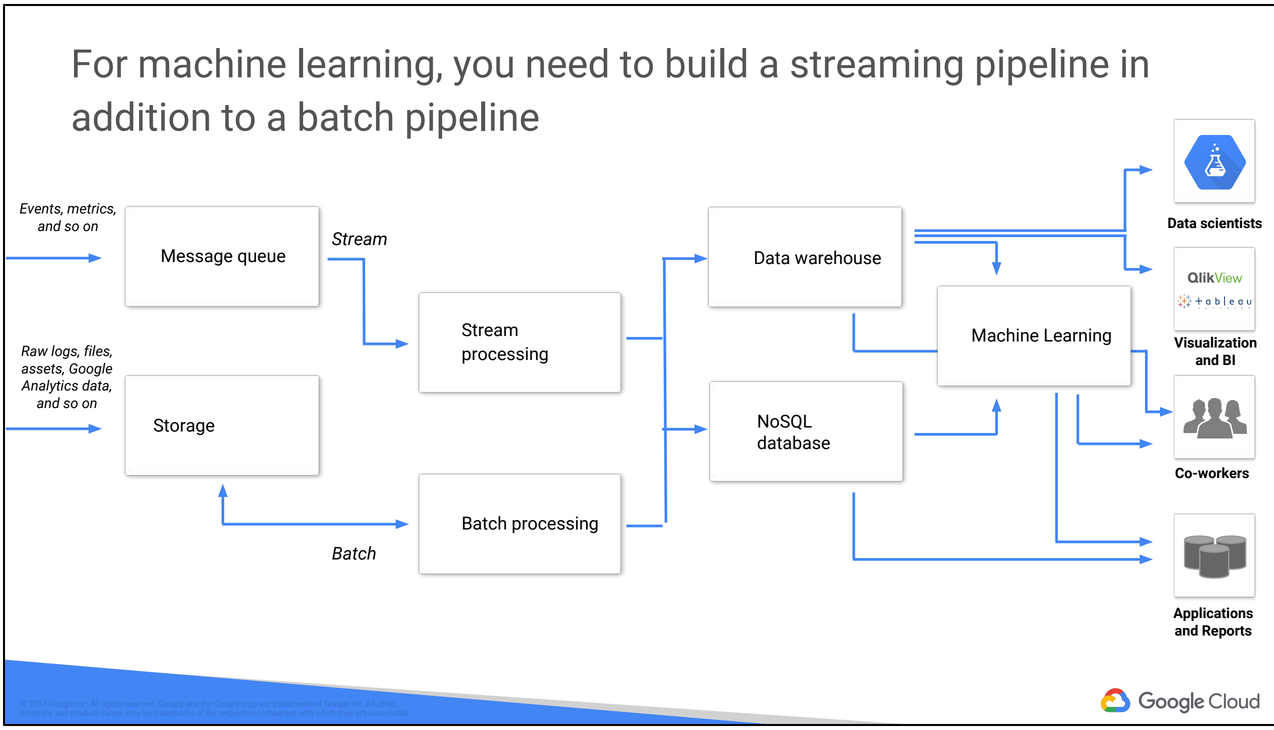
(2) Even if you don't have the data today, and so your ML project involves first collecting and rating data, you want to go through a manual analysis stage. The reason is that if you can't analyze your data to get reasonable inputs towards making decisions, there is no point to doing ML. Manual analysis helps you fail fast and try new ideas in a more agile way. So, don't skip the analysis step.

(3) To build a good ML model, you have to know your data. Since that's the first step, go through the process of doing manual data analysis. Don't jump straight to ML. We'll talk about this more in the next module.

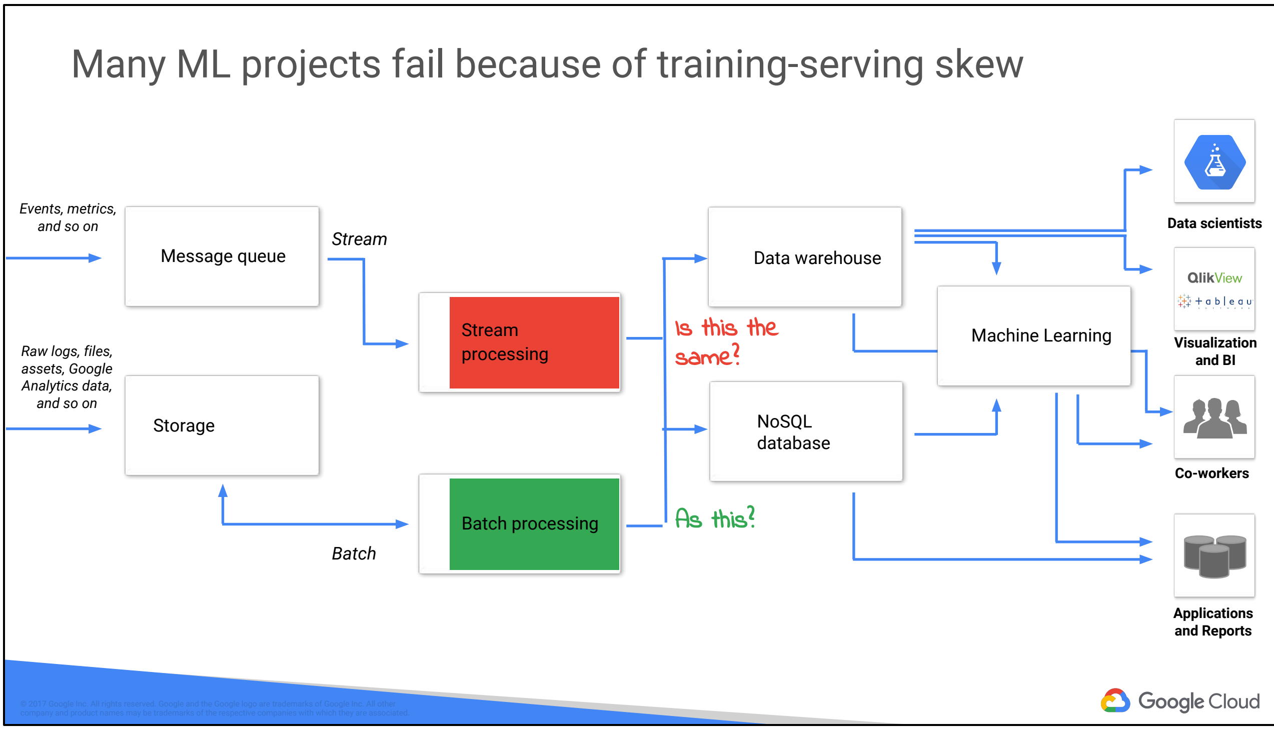
(4) Finally, ML is a journey towards automation and scale. You are automating manual analysis because you want it to scale. Perhaps, like Global fishing

(1) watch, you are manually analyzing a small fraction of fishing trips and you want to automate this so that you can scale up to analyzing a great deal more fishing trips.

More pithily, if you can’t do analytics, you can’t do ML.



When we say ML to engineers, they keep thinking “training”.
But the true utility of ML comes during predictions.
That’s when you are getting value from it.
One key thing, then, is that your models have to work on streaming data. You need to build up your streaming data sophistication.
If you are thinking you could get away with doing things weekly, as batch processing … guess what, your business is only getting faster.

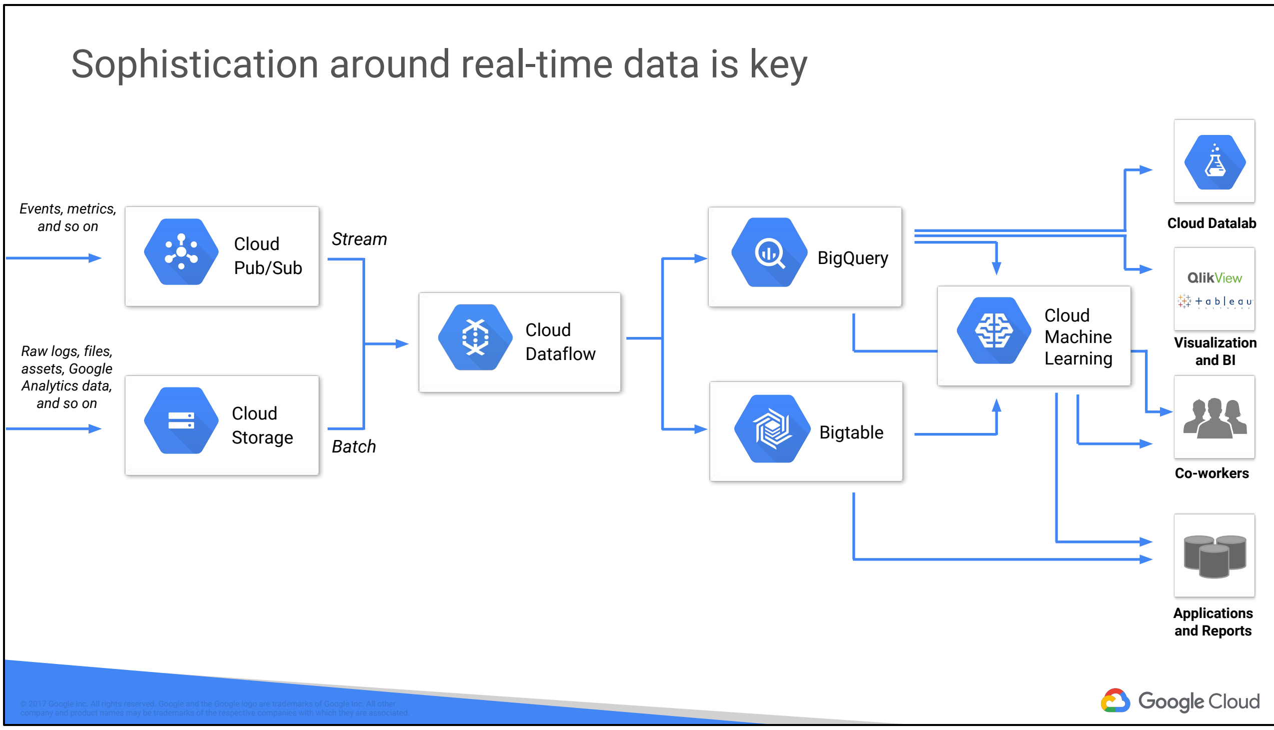


One common reason that ML projects fail is because of something called "training-serving skew". This is where you had a certain system for processing historical data so that you could train it. Perhaps it was a batch processing system written by your data science team.

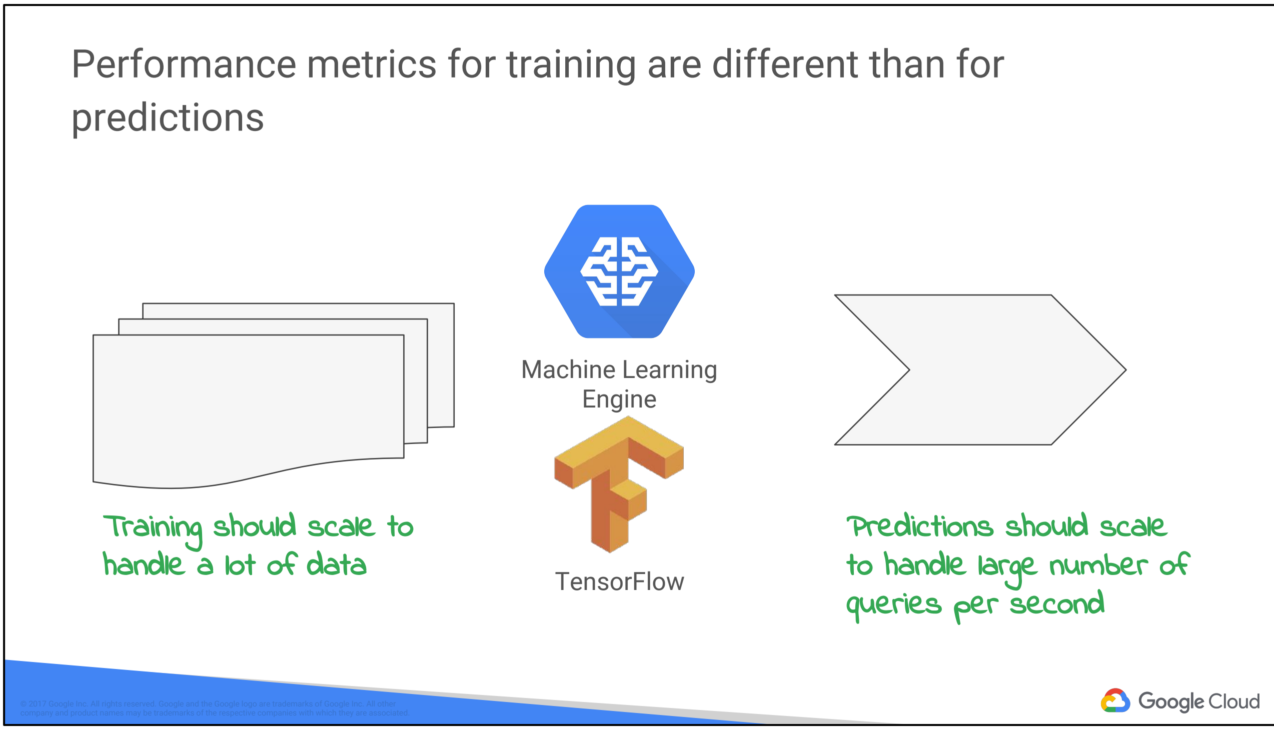
Then, you have a different system that needs to use the ML model during prediction; the system that serves these predictions is probably written in something that your production engineering team writes and maintains. Perhaps it is written in Java using web frameworks.

The problem is that unless the model sees the exact same data in serving as was used to do training, the model predictions are going to be off.

That is the problem that is referred to as "training-serving skew".



One way to reduce the chances of this is to take the same code that was used to process historical data (during training) and reuse it during predictions. But for that to happen, your data pipelines have to process both batch and stream.
This is a key insight behind Cloud Dataflow, a way to author data pipelines in Python, Java or even visually with Cloud Dataprep.
It’s open-sourced as Apache Beam. Where the B stands for batch and -eam stands for stream. A single system to do both batch and stream, because
In machine learning, it is helpful to use the same system in both training and prediction.



The performance metrics you care about change between training and predictions as well.
During training, the key performance aspect you care about is scaling to a lot of data. Distributed training, if you will.
During prediction, though, the key performance aspect is speed of response (high QPS).
This is a key insight behind TensorFlow. Lots of machine learning frameworks exist, for training. Not so many are equally capable of operationalization.

Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **An ML strategy**

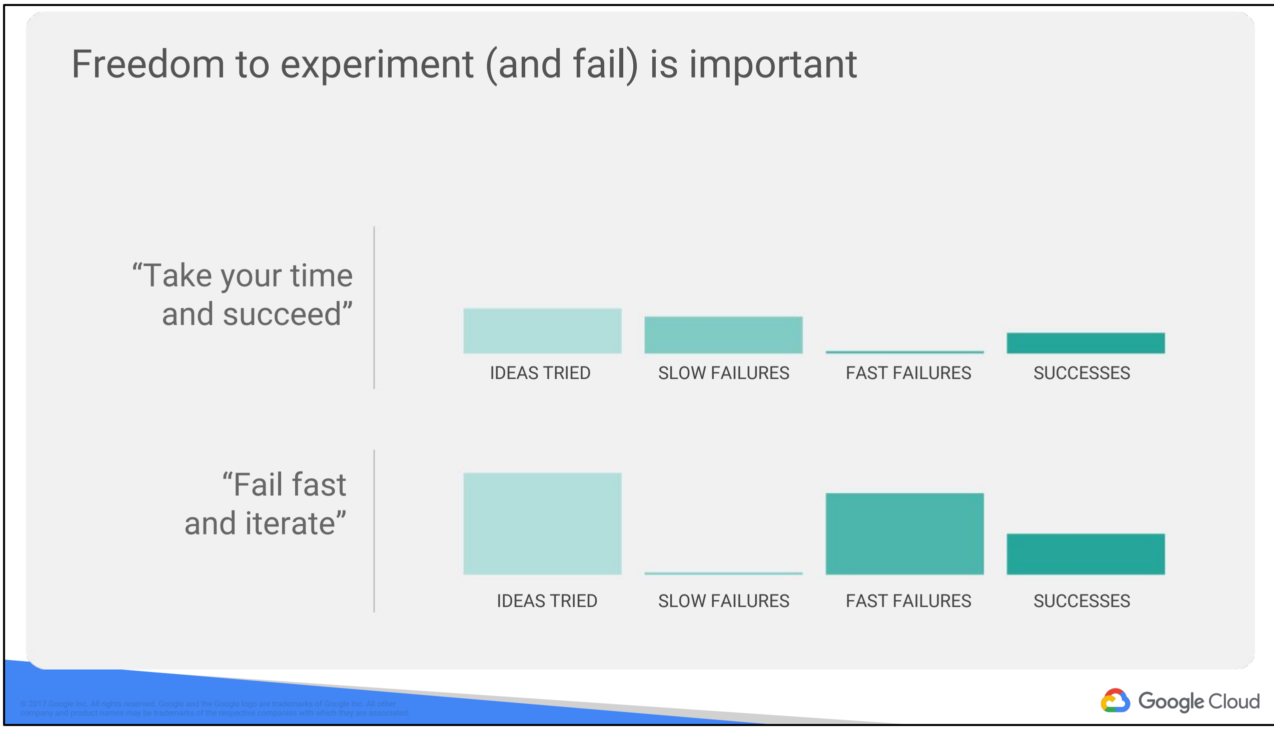
Format: Screencast

Google Cloud

Connect simple ML models into a pipeline

If there is one thing I want you to take away from this module, it is that the magic of ML comes with quantity, not complexity. Many small ML models each of which is simple. Think of my Maps example …

https://pixabay.com/en/blur-blurred-cascade-fall-flow-21653/ (cc0)

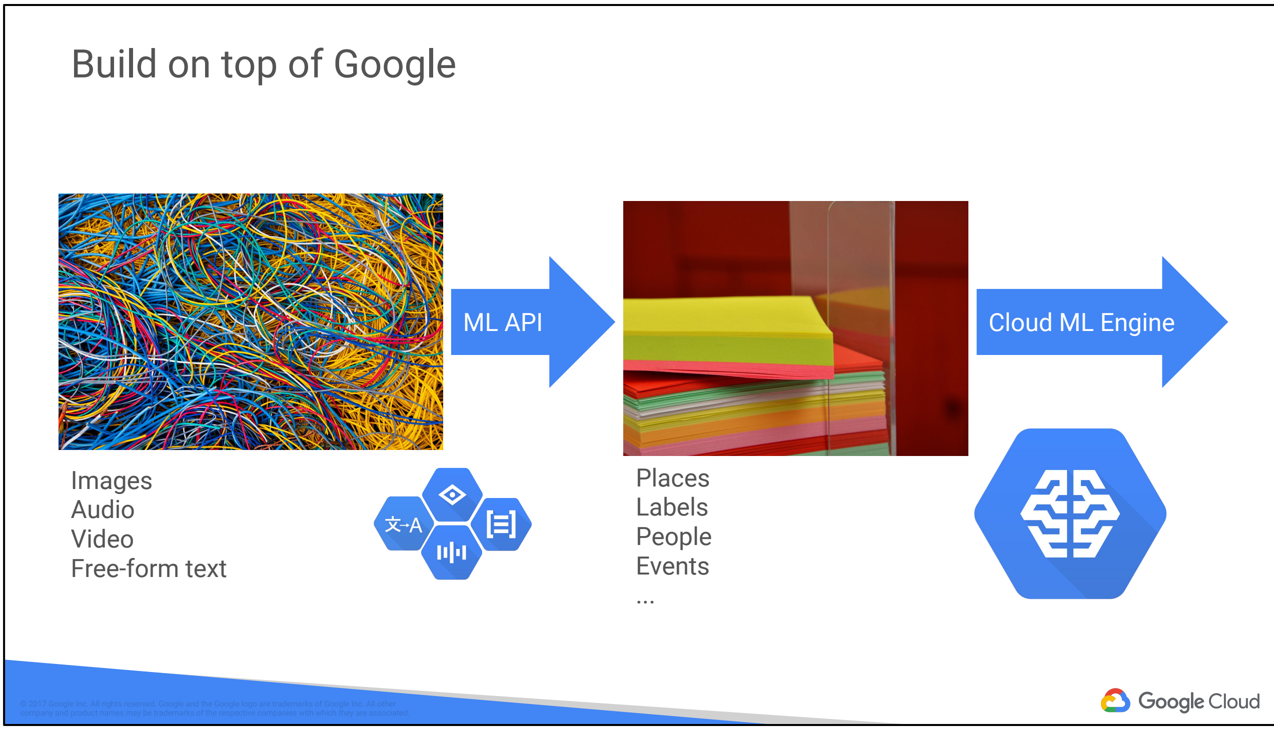


And if you are building many ML models, and planning for many more that you may never build, you want to have an environment where you fail fast.
The idea is that if you are failing fast, you get the ability to iterate.
This ability to experiment is critical in the realm of machine learning.

Unstructured data accounts for 90% of enterprise data*

*Source: IDC

When I say data, most people immediately think “structured data”, in my database.
But 90% of enterprise data is unstructured.
Think emails, video footage, texts, reports, catalogs, fashion shows, events, news, etc.



Dealing with structured data has gotten a lot easier because of the pretrained models we talked about. So, think of a ML pipeline as a way to deal with unstructured data. Take unstructured data, pass them through ML APIs, and then you are left with entities, places, labels, people, things that you can build a simpler ML model out of.

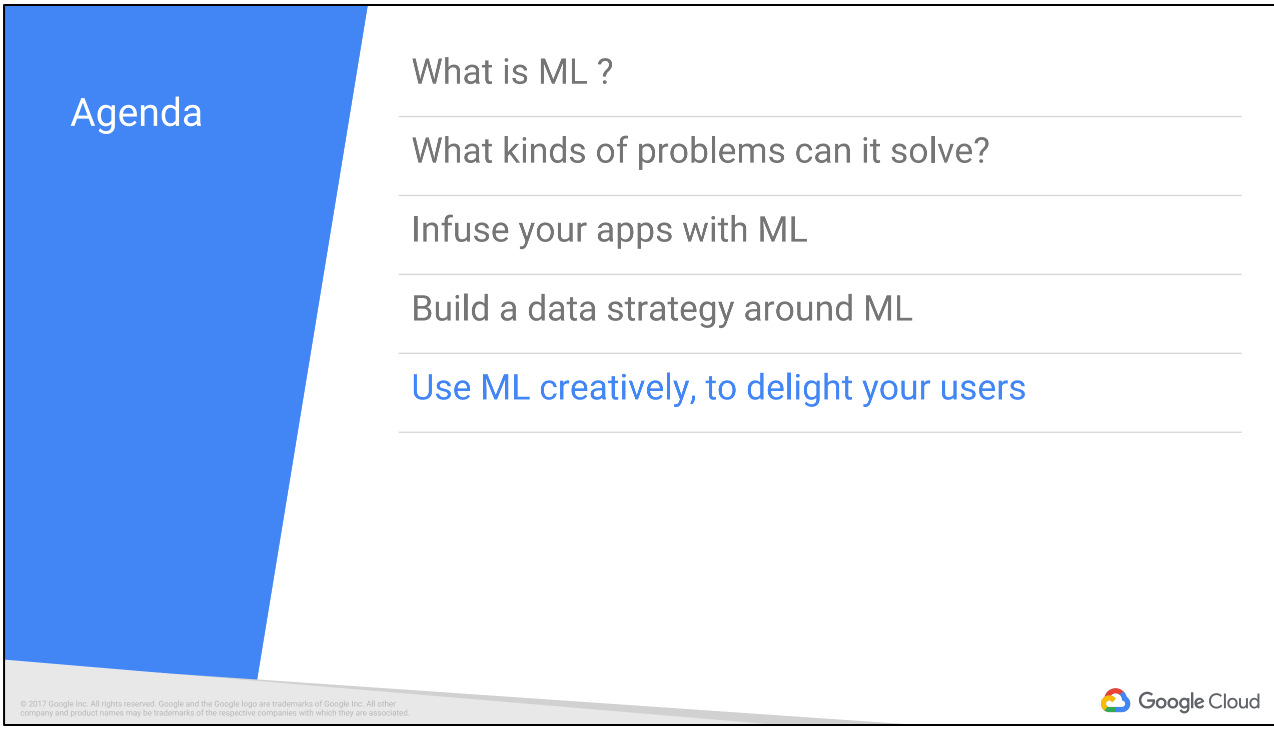
https://pixabay.com/en/list-zettelbox-note-leaves-stack-1925395/ (cc0)

Course 1: How Google does ML

Module 2: What it means to be AI-first

Lesson Title: **Transform your business**

Format: Screencast



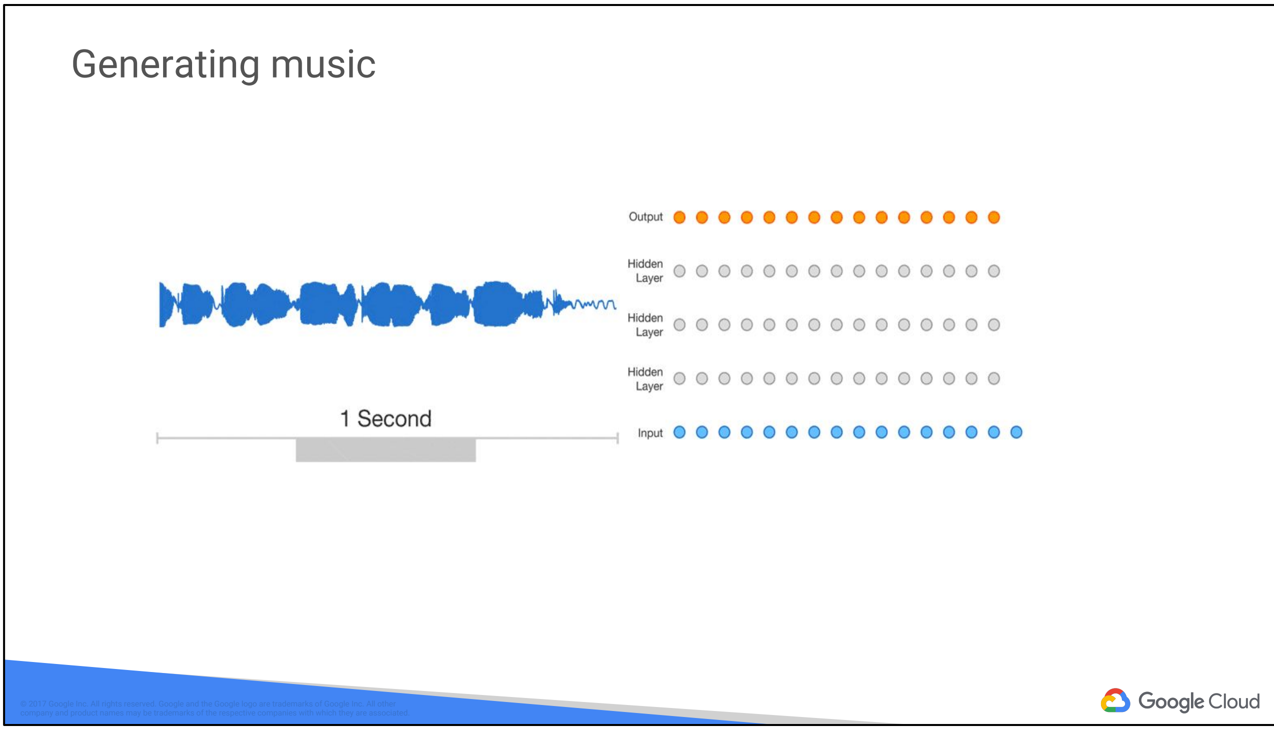
Fulfill user intent, using ML creatively, to delight your users.
The ability of ML to scale, to be personalized, gives you lots of opportunities to do this.

For ML to be magic, you should keep in mind the word “delight”. Delight your users. Anticipate their next need.

Did the card get canceled?
Flight get delayed?
Etc.
Or did they just walk into your store and see an empty shelf?

Does your system automatically find the “right” action to take for this customer? Rebook them? Offer them a meal voucher? Suggest that you are happy they are wearing your brand of sneakers?

https://pixabay.com/en/supermarket-shopping-empty-shelves-665049/ (cc0)



fulfill user intent in interesting ways using the capabilities of ML.

Users want background music in their videos, but using popular songs is a copyright violation. What if you generate music according to their specs??

- Classic piano music generated by WaveNet: https://deepmind.com/blog/wavenet-generative-model-raw-audio/
- Virtual girl images generated by GAN on TF: https://goo.gl/xkzbYI
    - Eg Game companies are trying GAN for generating avatar images for online game players

Your business can benefit from ML

Infuse your apps with ML

Simplify user input
Adapt to user

Fine-tune your business

Streamline your business processes

Create a new business

Delight your users

Anticipate users' needs
Creatively fulfill intent

Google Cloud

(just read slide)

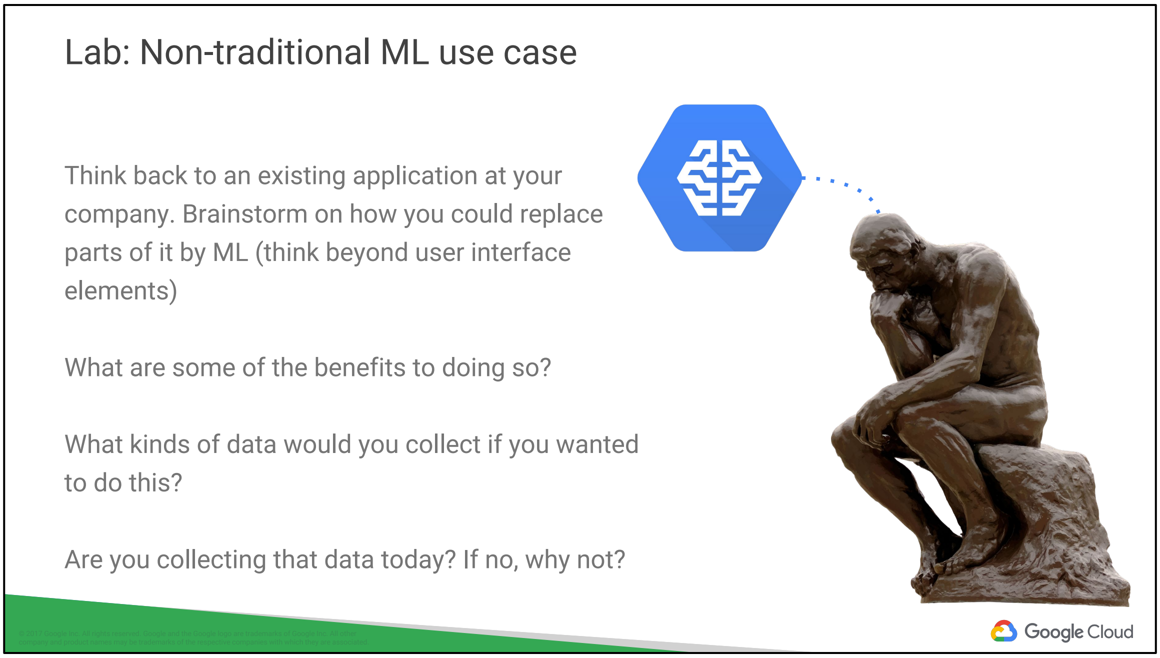
Image (Infuse ML): https://cloud.google.com/products/machine-learning/

Image (Fine-tune business) cc0:
https://pixabay.com/en/electrician-electric-electricity-1080554/

Image (Delight Users) cc0:
https://pixabay.com/en/youth-active-jump-happy-sunrise-570881/

Lab: Non-traditional ML use case

Google Cloud



Think back to an existing application at your company

Brainstorm on how you could replace parts of it by ML (think beyond
user interface elements)

What are some of the benefits to doing so?

What kinds of data would you collect if you wanted to do this?

Are you collecting that data today? If no, why not?

Image (Thinker):
https://pixabay.com/en/art-auguste-rodin-bronze-famous-1301872/


cloud.google.com
Images by Connie Zhou