Common ML Problems

In basic terms, ML is the process of training a piece of software, called a **model** (/machine-learning/glossary#model), to make useful predictions using a data set. This predictive model can then serve up predictions about previously unseen data. We use these predictions to take action in a product; for example, the system predicts that a user will like a certain video, so the system recommends that video to the user.

Often, people talk about ML as having two paradigms, supervised and unsupervised learning. However, it is more accurate to describe ML problems as falling along a spectrum of supervision between supervised and unsupervised learning. For the sake of simplicity, this course will focus on the two extremes of this spectrum.

finitions of many common ML terms, see the ML Glossary (/machine-learning/glossary)

What is Supervised Learning?

Supervised learning is a type of ML where the model is provided with **labeled** (/machine-learning/glossary#label) training data. But what does that mean?

For example, suppose you are an amateur botanist determined to differentiate between two species of the Lilliputian plant genus (a completely made-up plant). The two species look pretty similar. Fortunately, a botanist has put together a data set of Lilliputian plants she found in the wild along with their species name.

Here's a snippet of that data set:

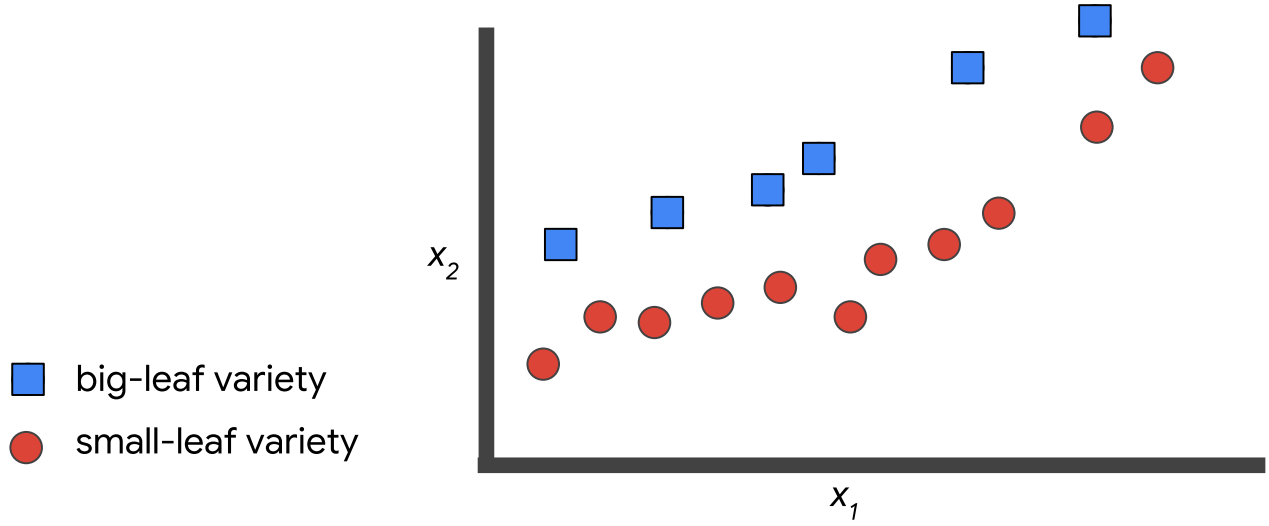
Leaf Width	Leaf Length	Species
2.7	4.9	small-leaf
3.2	5.5	big-leaf
2.9	5.1	small-leaf
3.4	6.8	big-leaf

Leaf width and leaf length are the **features** (/machine-learning/glossary#feature) (which is why the graph below labels both of these dimensions as X), while the species is the label. A real life botanical data set would probably contain far more features (including descriptions of

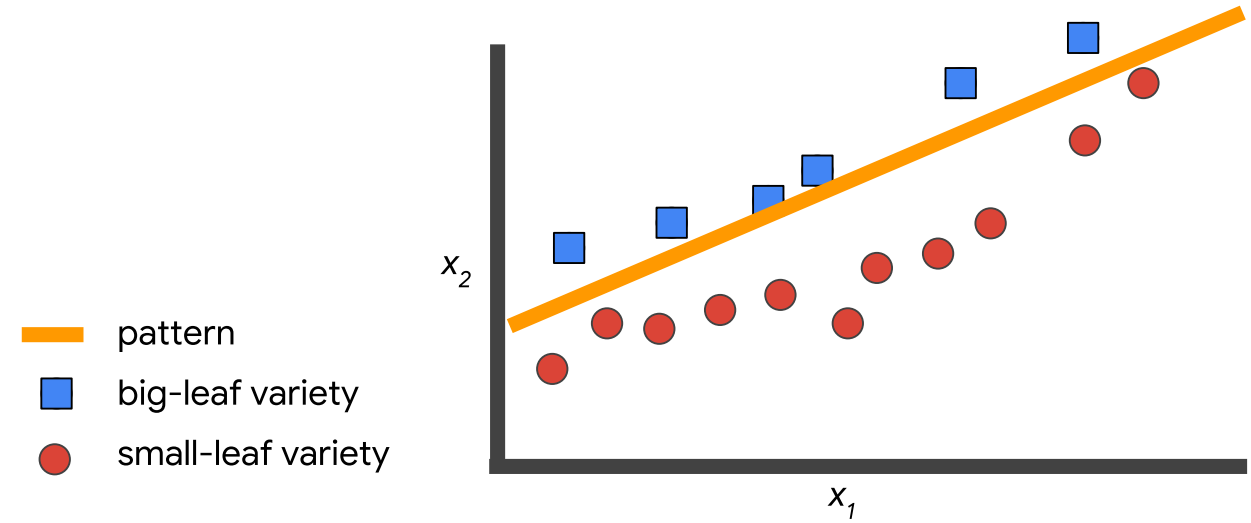
flowers, blooming times, arrangement of leaves) but still have only one label. Features are measurements or descriptions; the label is essentially the "answer." For example, the goal of the data set is to help other botanists answer the question, "Which species is this plant?"

This data set consists of only four **examples** (/machine-learning/glossary#example). A real life data set would likely contain vastly more examples.

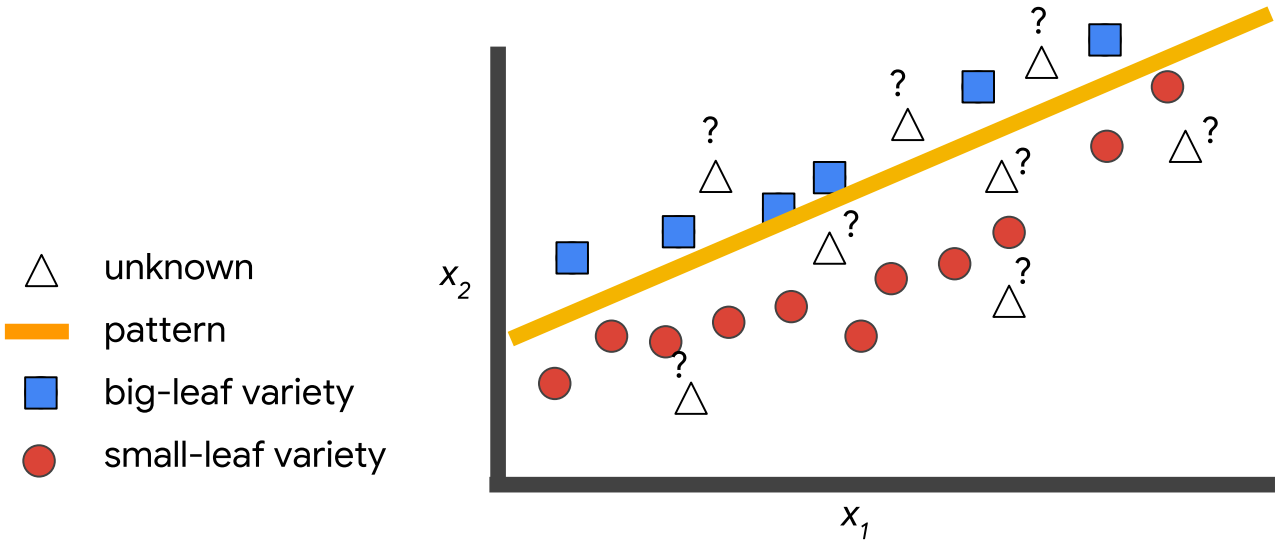
Suppose we graph the leaf width and leaf length and then color-code the species.



In **supervised machine learning** (/machine-learning/glossary#supervised_machine_learning), you feed the features and their corresponding labels into an algorithm in a process called **training** (/machine-learning/glossary#training). During training, the algorithm gradually determines the relationship between features and their corresponding labels. This relationship is called the **model** (/machine-learning/glossary#model). Often times in machine learning, the model is very complex. However, suppose that this model can be represented as a line that separates big-leaf from small-leaf:



Now that a model exists, you can use that model to classify new plants that you find in the jungle. For example:



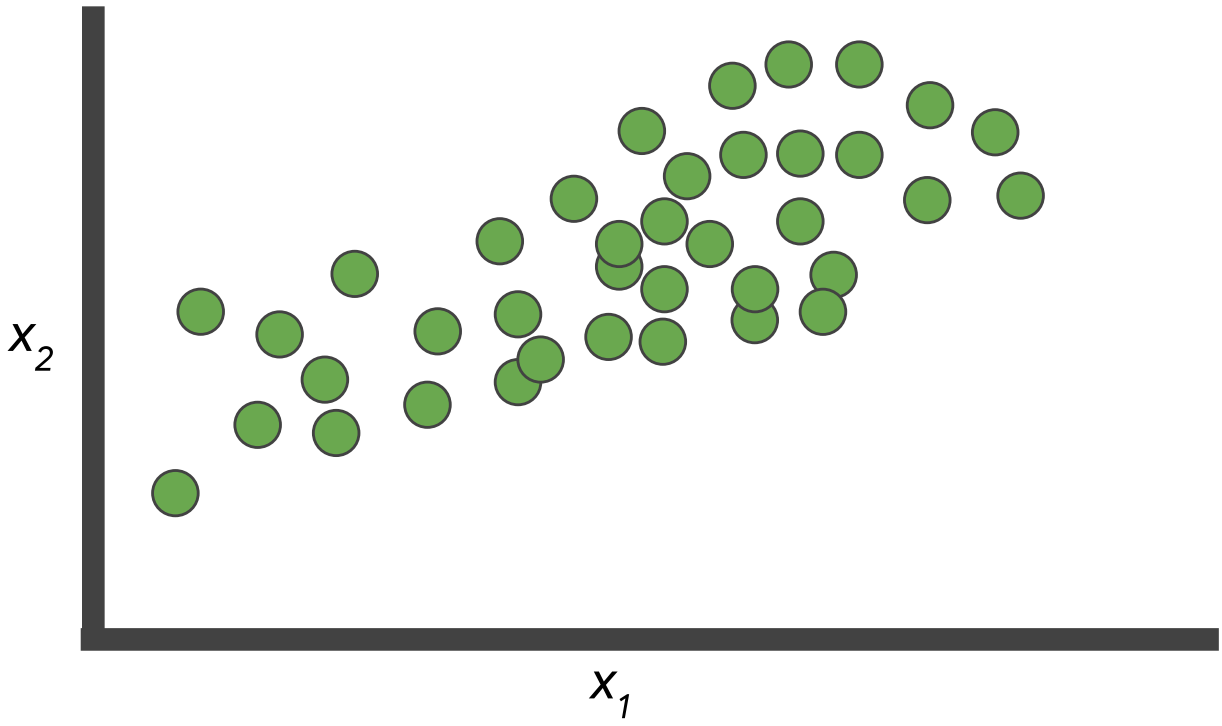
To tie it all together, supervised machine learning finds patterns between data and labels that can be expressed mathematically as functions. Given an input feature, you are telling the system what the expected output label is, thus you are supervising the training. The ML system will learn patterns on this labeled data. In the future, the ML system will use these patterns to make predictions on data that it did not see during training.

An exciting real-world example of supervised learning is a study from Stanford University (https://news.stanford.edu/2017/01/25/artificial-intelligence-used-identify-skin-cancer/) that used a model to detect skin cancer in images. In this case, the training set contained images of skin labeled by dermatologists as having one of several diseases. The ML system found signals that indicate each disease from its training set, and used those signals to make predictions on new, unlabeled images.

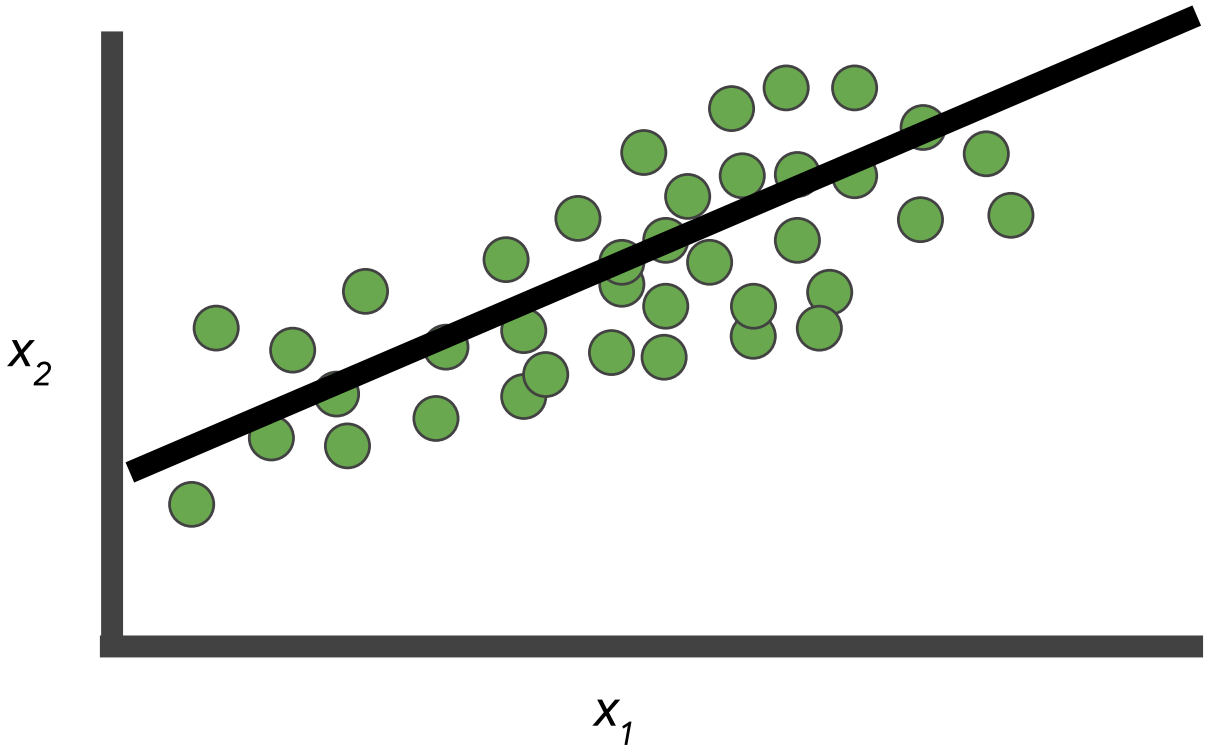
Unsupervised Learning

In unsupervised learning, the goal is to identify meaningful patterns in the data. To accomplish this, the machine must learn from an unlabeled data set. In other words, the model has no hints how to categorize each piece of data and must infer its own rules for doing so.

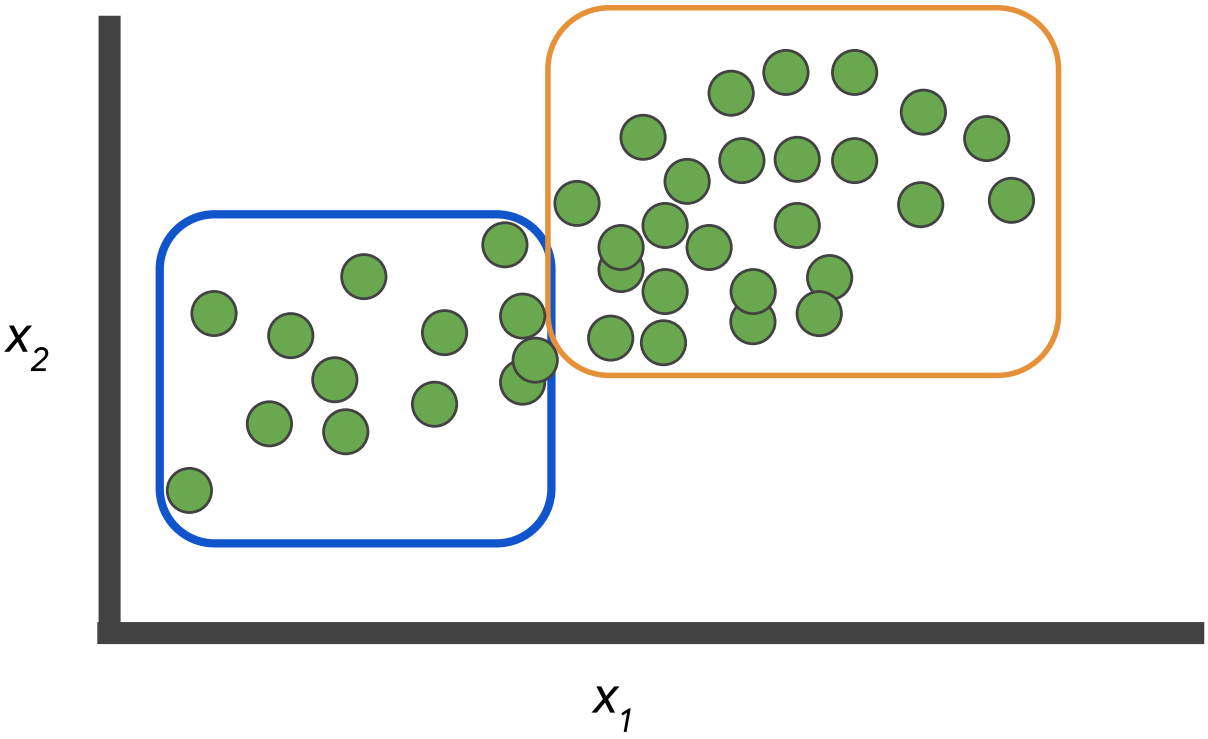
In the following graph, all the examples are the same shape because we don't have labels to differentiate between examples of one type or another here:



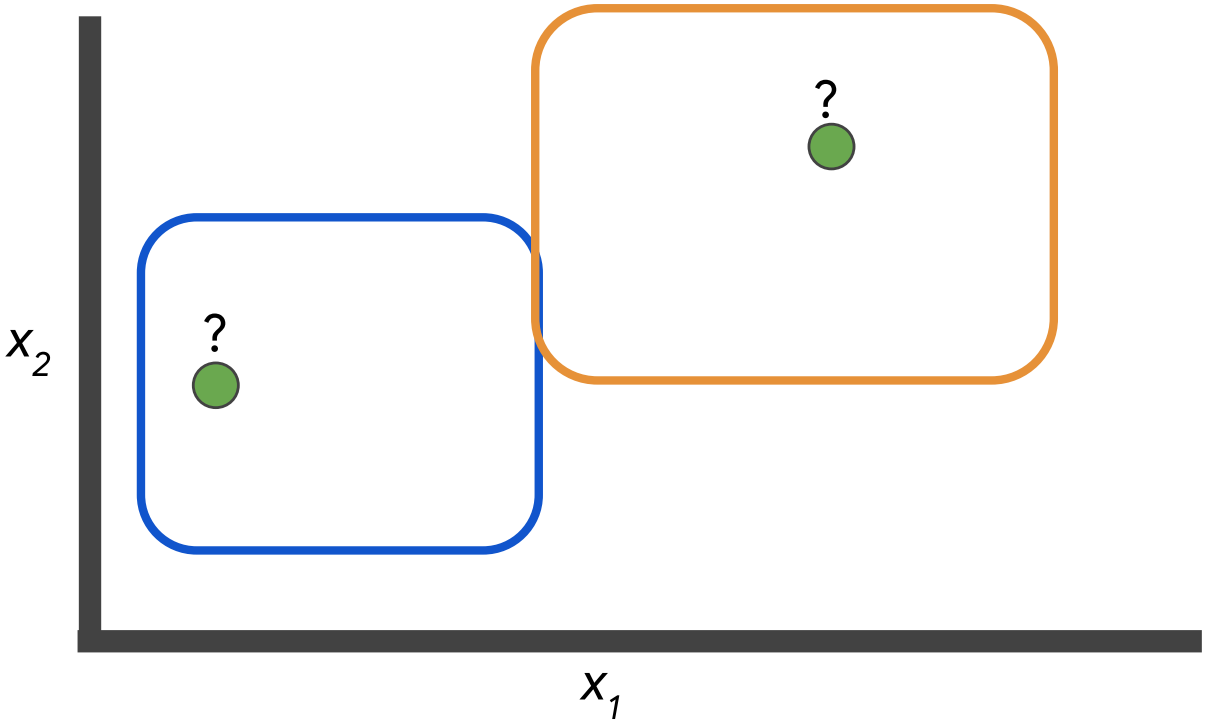
Fitting a line to unlabeled points isn't helpful. We still end up with examples of the same shape on both sides of the line. Clearly we will have to try a different approach.



Here, we have two clusters. (Note that the number of clusters is arbitrary). What do these clusters represent? It can be difficult to say. Sometimes the model finds patterns in the data that you don't want it to learn, such as stereotypes or **bias** (/machine-learning/glossary#bias_ethics).



However, when new data arrives, we can categorize it pretty easily, assuming it fits into a known cluster. But what if your photo clustering model has never seen a pangolin (https://wikipedia.org/wiki/Pangolin) before? Will the system cluster the new photo with armadillos or maybe hedgehogs? This course will talk more about the difficulties of unlabeled data and clustering later on.



While it is very common, clustering is not the only type of unsupervised learning.

Reinforcement Learning

An additional branch of machine learning is **reinforcement learning (RL)**. Reinforcement learning differs from other types of machine learning. In RL you don't collect examples with labels. Imagine you want to teach a machine to play a very basic video game and never lose. You set up the model (often called an **agent** in RL) with the game, and you tell the model not to get a "game over" screen. During training, the agent receives a reward when it performs this task, which is called a reward function. With reinforcement learning, the agent can learn very quickly how to outperform humans.

The lack of a data requirement makes RL a tempting approach. However, designing a good reward function is difficult, and RL models are less stable and predictable than supervised approaches. Additionally, you need to provide a way for the agent to interact with the game to produce data, which means either building a physical agent that can interact with the real world or a virtual agent and a virtual world, either of which is a big challenge. See this blog post (https://www.alexirpan.com/2018/02/14/rl-hard.html) by Alex Irpan for an overview of the types of problems currently faced in RL. Reinforcement learning is an active field of ML research, but in this course we'll focus on supervised solutions because they're a better known problem, more stable, and result in a simpler system.

For comprehensive information on RL, check out *Reinforcement Learning: An Introduction* (https://mitpress.mit.edu/books/reinforcement-learning) by Sutton and Barto.

Types of ML Problems

There are several subclasses of ML problems based on what the prediction task looks like. In the table below, you can see examples of common supervised and unsupervised ML problems.

Type of ML Problem	Description	Example
Classification	Pick one of N labels	Cat, dog, horse, or bear
Regression	Predict numerical values	Click-through rate
Clustering	Group similar examples	Most relevant documents (unsupervised)
Association rule learning	Infer likely association patterns in data	If you buy hamburger buns, you're likely to buy hamburgers (unsupervised)
Structured output	Create complex output	Natural language parse trees, image recognition bounding boxes

Ranking	Identify position on a scale or status	Search result ranking

Check Your Understanding

Which ML problem is an example of unsupervised learning? Click on an answer to expand the section and check your response.

- Structured output
- Clustering
- Regression
- Classification

Contrasting Cases

As you walk through each example, note the types of data used and how that data informed the product design and iterations. Think about how the examples compare to and contrast from each other. Click on each product name button to see more information below.

Smart ReplyYouTube Watch Next (#youtu… Cucumber Sorting (#cucumb…

Suggested short responses to emails.

Smart Reply is an example of ML that utilizes Natural Language Understanding (NLU) and generation, sequence-to-sequence learning, to make replying to a flooded inbox far less painful.

- Computer, respond to this email (https://research.googleblog.com/2015/11/computer-respond-to-this-email.html)
- Smart Reply: Automated Response Suggestion for Email (http://www.kdd.org/kdd2016/papers/files/Paper_1069.pdf) (2016 article)

Thought Questions

Think about the similarities and differences between each of the above cases. Click on the plus icon to expand the section and reveal the answers.

What user problem did these systems solve?

In all three cases there was motivation to build an ML system to address a real problem users were facing.

- Smart Reply: responding to emails can take up too much time
- YouTube: there are too many videos on YouTube for one person to navigate and find videos they like
- Cucumber sorter: the cucumber sorting process is burdensome

What does output from these systems look like?

Each is a bit different.

- Smart Reply: three short suggested responses at the bottom of an email
- YouTube: suggested videos along the right-hand side of the screen
- Cucumber sorter: directions to a robot arm that sorts cucumbers into their correct categories

What data sources were used?

In all three cases the large amounts of historical data had information closely tied to what we wanted to do.

- Smart Reply: conversation data (email messages and responses)
- YouTube: watch time, click-through rate, watch history, search history
- Cucumber sorter: exemplary cucumber data (size, shape, weight, etc.)

Previous
← Introduction to Machine Learning Problem Framing (/machine-learning/problem-framing)

Next
The ML Mindset (/machine-learning/problem-framing/big-questions) →

Last updated 2021-02-05 UTC.