Machine Learning

**Types of ML Systems**

ML systems fall into one or more of the following categories based on how they learn to make predictions or generate content:

* Supervised learning
* Unsupervised learning
* Reinforcement learning
* Generative AI
* **supervised machine learning**
* Training a [**model**](https://developers.google.com/machine-learning/glossary#model) from [**features**](https://developers.google.com/machine-learning/glossary#feature) and their corresponding [**labels**](https://developers.google.com/machine-learning/glossary#label). Supervised machine learning is analogous to learning a subject by studying a set of questions and their corresponding answers. After mastering the mapping between questions and answers, a student can then provide answers to new (never-before-seen) questions on the same topic.

Two of the most common use cases for supervised learning are regression and classification.

See the table below for more examples of regression models:

| **Scenario** | **Possible input data** | **Numeric prediction** |
| --- | --- | --- |
| Future house price | Square footage, zip code, number of bedrooms and bathrooms, lot size, mortgage interest rate, property tax rate, construction costs, and number of homes for sale in the area. | The price of the home. |
| Future ride time | Historical traffic conditions (gathered from smartphones, traffic sensors, ride-hailing and other navigation applications), distance from destination, and weather conditions. | The time in minutes and seconds to arrive at a destination. |

**Classification**

[Classification models](https://developers.google.com/machine-learning/glossary#classification-model) predict the likelihood that something belongs to a category. Unlike regression models, whose output is a number, classification models output a value that states whether or not something belongs to a particular category. For example, classification models are used to predict if an email is spam or if a photo contains a cat.

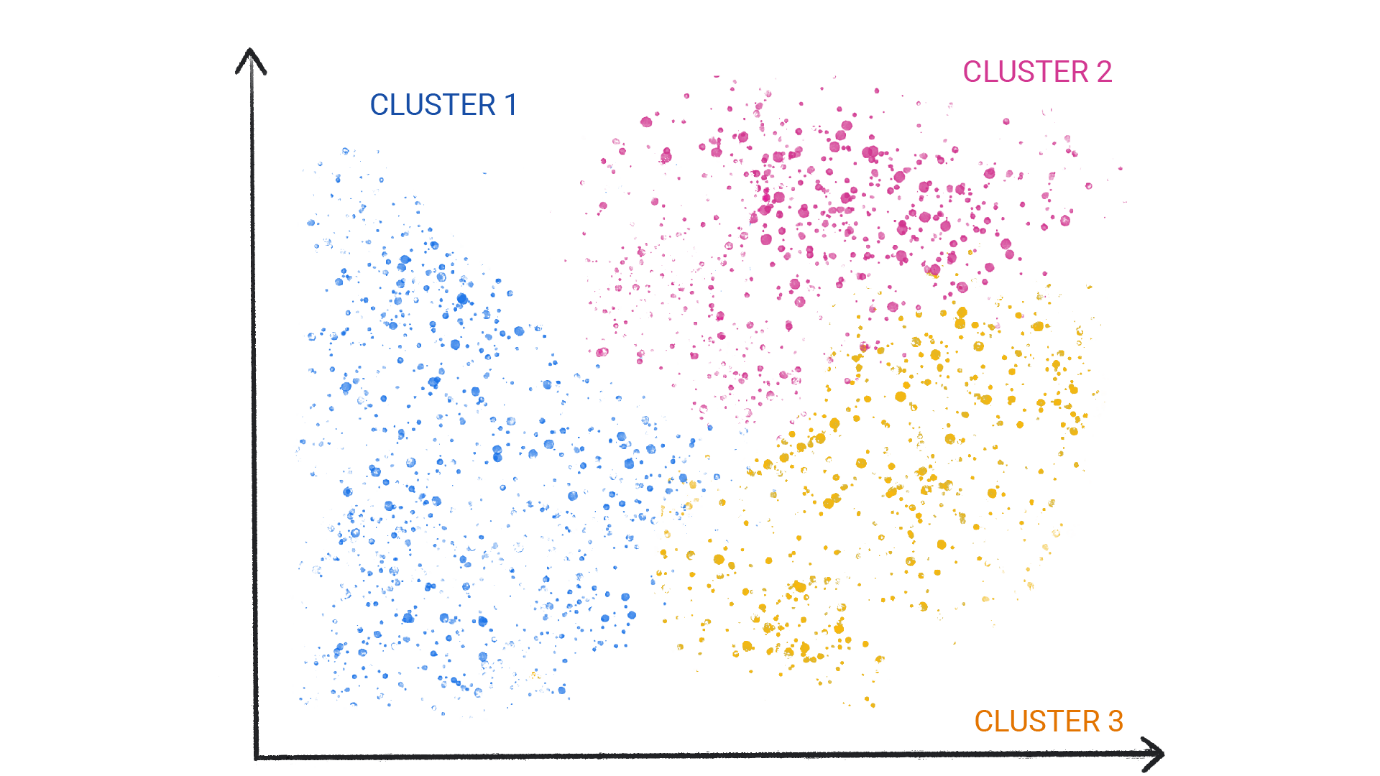
Classification models are divided into two groups: binary classification and multiclass classification. Binary classification models output a value from a class that contains only two values, for example, a model that outputs either rain or no rain. Multiclass classification models output a value from a class that contains more than two values, for example, a model that can output either rain, hail, snow, or sleet.

* **unsupervised machine learning**
* Training a [**model**](https://developers.google.com/machine-learning/glossary#model) to find patterns in a dataset, typically an unlabeled dataset.
* The most common use of unsupervised machine learning is to [**cluster**](https://developers.google.com/machine-learning/glossary#clustering) data into groups of similar examples. For example, an unsupervised machine learning algorithm can cluster songs based on various properties of the music. The resulting clusters can become an input to other machine learning algorithms (for example, to a music recommendation service). Clustering can help when useful labels are scarce or absent. For example, in domains such as anti-abuse and fraud, clusters can help humans better understand the data.

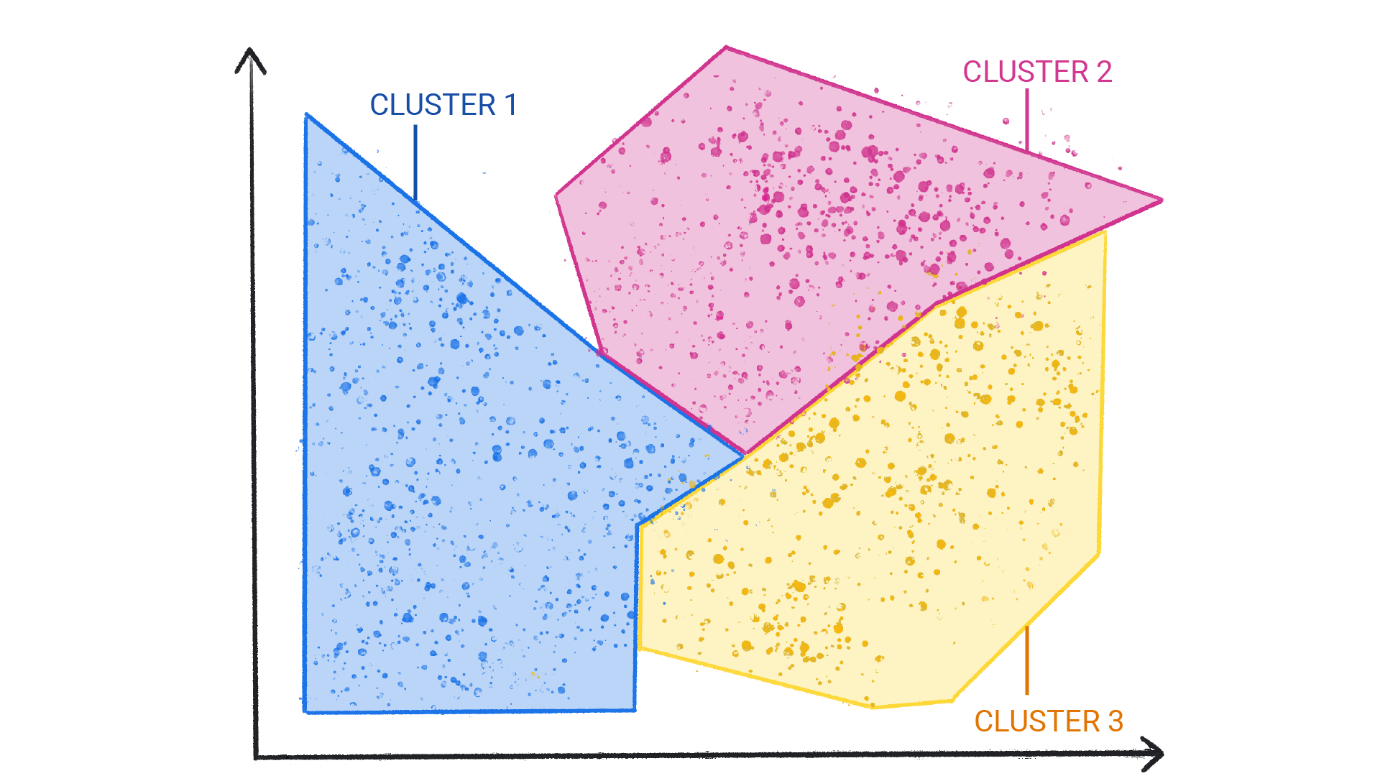
[Unsupervised learning](https://developers.google.com/machine-learning/glossary#unsupervised-machine-learning) models make predictions by being given data that does not contain any correct answers. An unsupervised learning model's goal is to identify meaningful patterns among the data. In other words, the model has no hints on how to categorize each piece of data, but instead it must infer its own rules.

A commonly used unsupervised learning model employs a technique called [clustering](https://developers.google.com/machine-learning/glossary#clustering). The model finds data points that demarcate natural groupings.

**A supervised approach is given data that contains the correct answer. The model's job is to find connections in the data that produce the correct answer. An unsupervised approach is given data without the correct answer. Its job is to find groupings in the data.**



**Figure 1**. An ML model clustering similar data points.



**Reinforcement learning**

[Reinforcement learning](https://developers.google.com/machine-learning/glossary#reinforcement-learning-rl) models make predictions by getting [rewards](https://developers.google.com/machine-learning/glossary#reward) or penalties based on actions performed within an environment. A reinforcement learning system generates a [policy](https://developers.google.com/machine-learning/glossary#policy) that defines the best strategy for getting the most rewards.

Reinforcement learning is used to train robots to perform tasks, like walking around a room, and software programs like [AlphaGo](https://deepmind.com/research/case-studies/alphago-the-story-so-far) to play the game of Go.

**Generative AI**

[Generative AI](https://developers.google.com/machine-learning/glossary#generative-ai) is a class of models that creates content from user input. For example, generative AI can create unique images, music compositions, and jokes; it can summarize articles, explain how to perform a task, or edit a photo.

Generative AI can take a variety of inputs and create a variety of outputs, like text, images, audio, and video. It can also take and create combinations of these. For example, a model can take an image as input and create an image and text as output, or take an image and text as input and create a video as output.

We can discuss generative models by their inputs and outputs, typically written as "type of input"-to-"type of output." For example, the following is a partial list of some inputs and outputs for generative models:

* Text-to-text
* Text-to-image
* Text-to-video
* Text-to-code
* Text-to-speech
* Image and text-to-image

**Supervised Learning**

bookmark\_border

Supervised learning's tasks are well-defined and can be applied to a multitude of scenarios—like identifying spam or predicting precipitation.

**Foundational supervised learning concepts**

Supervised machine learning is based on the following core concepts:

* Data
* Model
* Training
* Evaluating
* Inference

**Data**

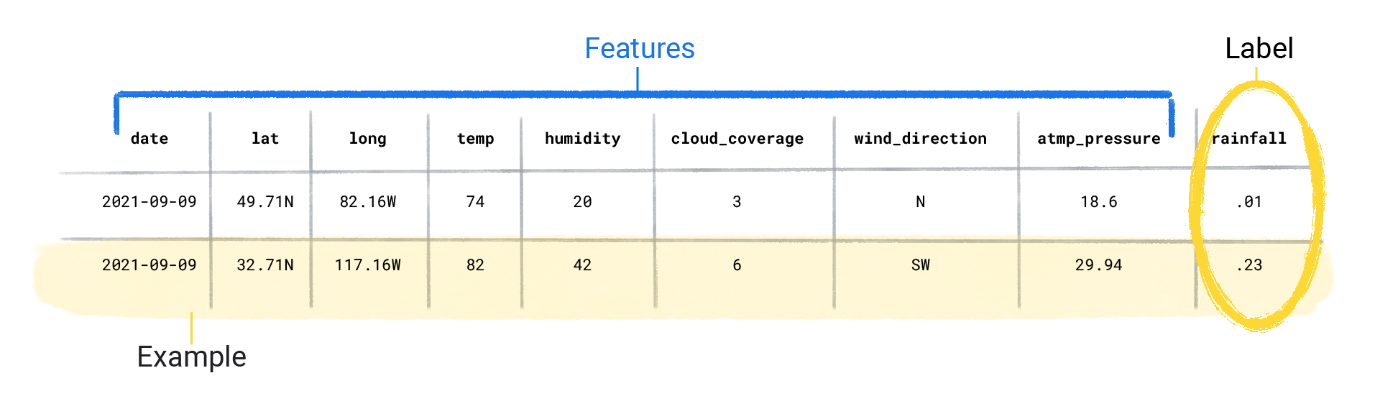
Data is the driving force of ML. Data comes in the form of words and numbers stored in tables, or as the values of pixels and waveforms captured in images and audio files. We store related data in datasets. For example, we might have a dataset of the following:

* Images of cats
* Housing prices
* Weather information

Datasets are made up of individual [examples](https://developers.google.com/machine-learning/glossary#example) that contain [features](https://developers.google.com/machine-learning/glossary#feature) and a [label](https://developers.google.com/machine-learning/glossary#label). You could think of an example as analogous to a single row in a spreadsheet. Features are the values that a supervised model uses to predict the label. The label is the "answer," or the value we want the model to predict. In a weather model that predicts rainfall, the features could be *latitude*, *longitude*, *temperature*, *humidity*, *cloud coverage*, *wind direction*, and *atmospheric pressure*. The label would be *rainfall amount*.

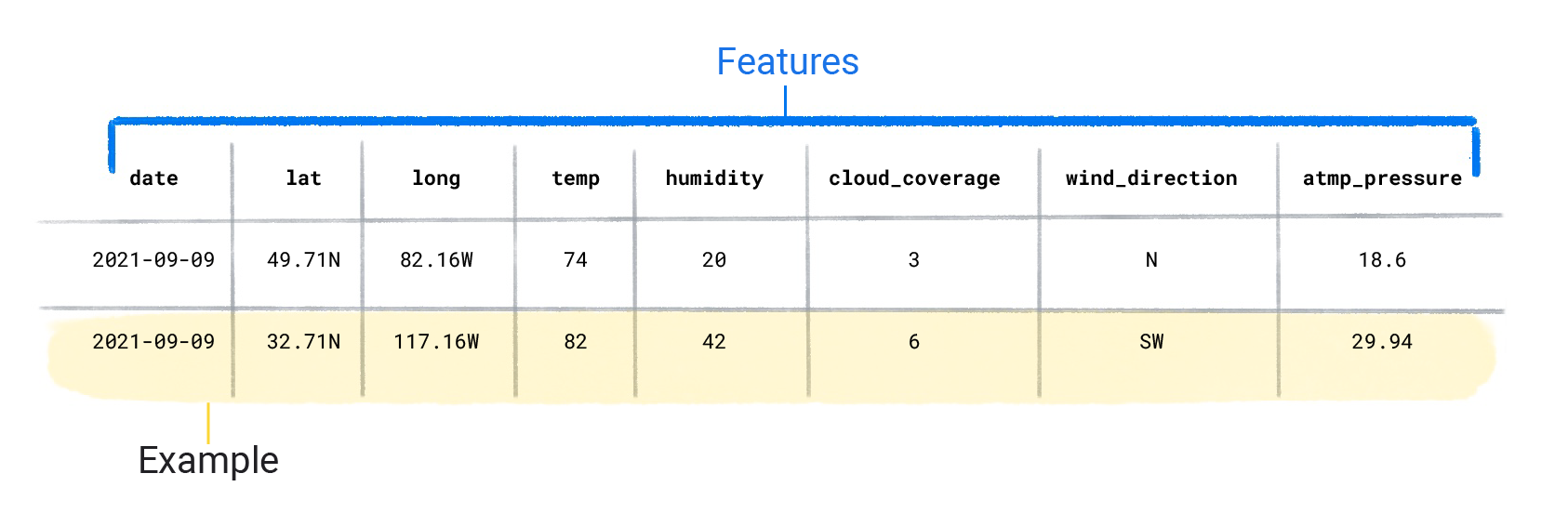
Examples that contain both features and a label are called [labeled examples](https://developers.google.com/machine-learning/glossary" \l "labeled-example).

**Two labeled examples**



In contrast, unlabeled examples contain features, but no label. After you create a model, the model predicts the label from the features.

**Two unlabeled examples**



**Dataset characteristics**

A dataset is characterized by its size and diversity. Size indicates the number of examples. Diversity indicates the range those examples cover. Good datasets are both large and highly diverse.

Some datasets are both large and diverse. However, some datasets are large but have low diversity, and some are small but highly diverse. In other words, a large dataset doesn’t guarantee sufficient diversity, and a dataset that is highly diverse doesn't guarantee sufficient examples.

For instance, a dataset might contain 100 years worth of data, but only for the month of July. Using this dataset to predict rainfall in January would produce poor predictions. Conversely, a dataset might cover only a few years but contain every month. This dataset might produce poor predictions because it doesn't contain enough years to account for variability.

**Check Your Understanding**

What attributes of a dataset would be ideal to use for ML?

Large size / Low diversity

Small size / High diversity

Small size / Low diversity

Large size / High diversity

A dataset can also be characterized by the number of its features. For example, some weather datasets might contain hundreds of features, ranging from satellite imagery to cloud coverage values. Other datasets might contain only three or four features, like humidity, atmospheric pressure, and temperature. Datasets with more features can help a model discover additional patterns and make better predictions. However, datasets with more features don't *always* produce models that make better predictions because some features might have no causal relationship to the label.

**Model**

In supervised learning, a model is the complex collection of numbers that define the mathematical relationship from specific input feature patterns to specific output label values. The model discovers these patterns through training.

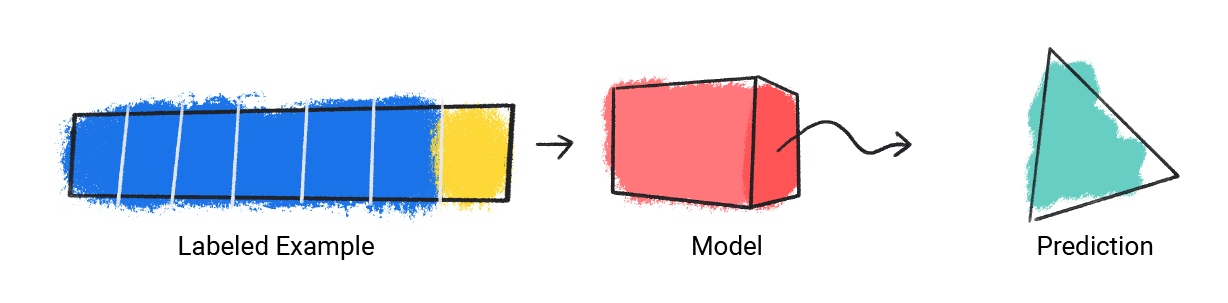
**Training**

Before a supervised model can make predictions, it must be trained. To train a model, we give the model a dataset with labeled examples. The model's goal is to work out the best solution for predicting the labels from the features. The model finds the best solution by comparing its predicted value to the label's actual value. Based on the difference between the predicted and actual values—defined as the [loss](https://developers.google.com/machine-learning/glossary#loss)—the model gradually updates its solution. In other words, the model learns the mathematical relationship between the features and the label so that it can make the best predictions on unseen data.

For example, if the model predicted 1.15 inches of rain, but the actual value was .75 inches, the model modifies its solution so its prediction is closer to .75 inches. After the model has looked at each example in the dataset—in some cases, multiple times—it arrives at a solution that makes the best predictions, on average, for each of the examples.

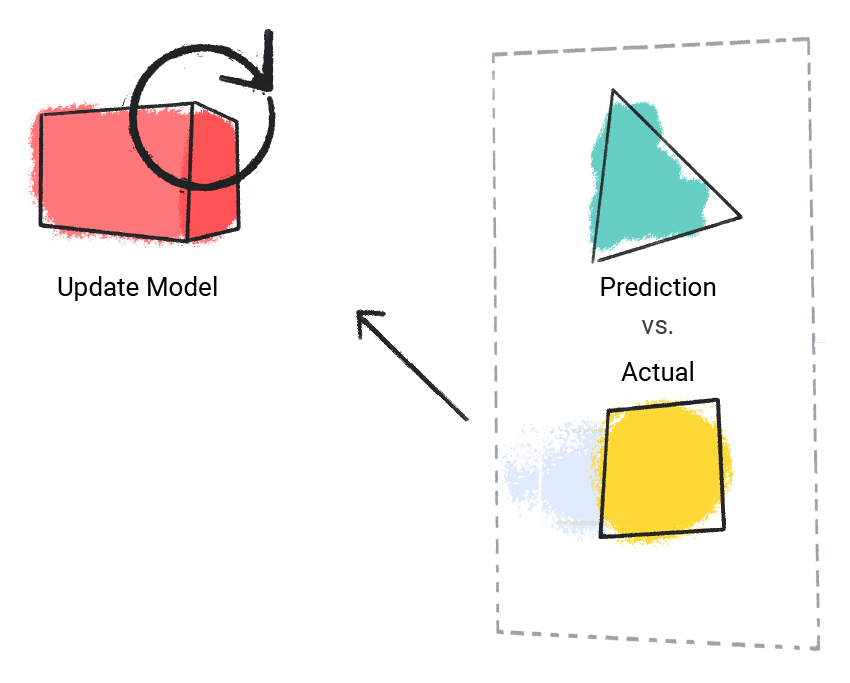
The following demonstrates training a model:

1. The model takes in a single labeled example and provides a prediction.



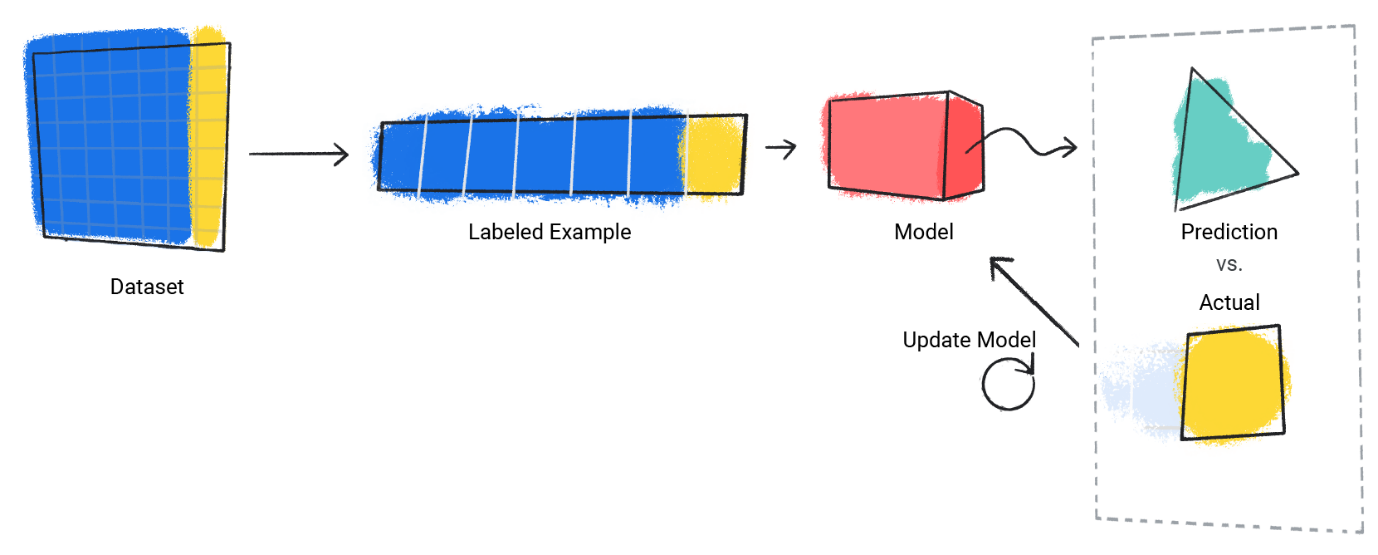
**Figure 1**. An ML model making a prediction from a labeled example.

1. The model compares its predicted value with the actual value and updates its solution.



**Figure 2**. An ML model updating its predicted value.

1. The model repeats this process for each labeled example in the dataset.



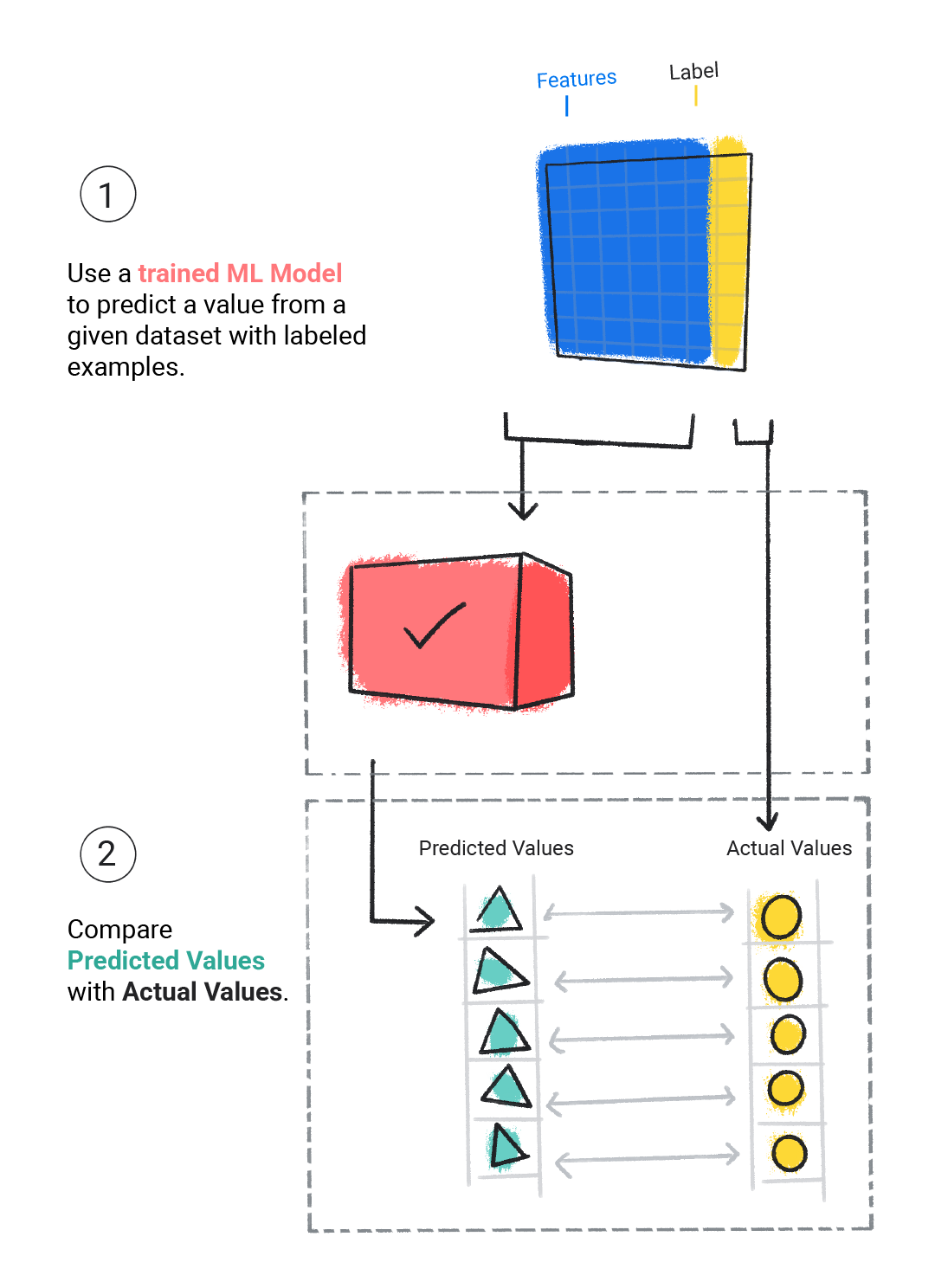
**Figure 3**. An ML model updating its predictions for each labeled example in the training dataset.

In this way, the model gradually learns the correct relationship between the features and the label. This gradual understanding is also why large and diverse datasets produce a better model. The model has seen more data with a wider range of values and has refined its understanding of the relationship between the features and the label.

During training, ML practitioners can make subtle adjustments to the configurations and features the model uses to make predictions. For example, certain features have more predictive power than others. Therefore, ML practitioners can select which features the model uses during training. For example, suppose a weather dataset containstime\_of\_day as a feature. In this case, an ML practitioner can add or remove time\_of\_day during training to see whether the model makes better predictions with or without it.

**Evaluating**

We evaluate a trained model to determine how well it learned. When we evaluate a model, we use a labeled dataset, but we only give the model the dataset's features. We then compare the model's predictions to the label's true values.



**Figure 4**. Evaluating an ML model by comparing its predictions to the actual values.

Depending on the model's predictions, we might do more training and evaluating before deploying the model in a real-world application.