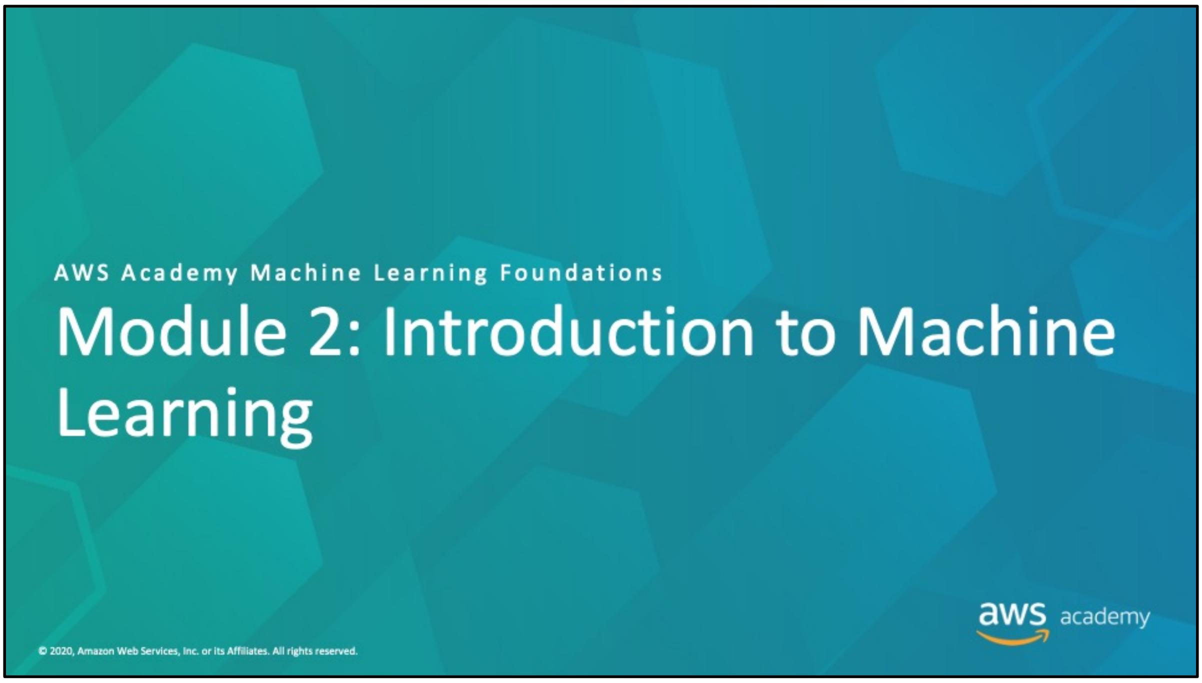


Welcome to Module 2: Introduction to Machine Learning.

Module overview

aws academy

Sections

1. What is machine learning?
2. Business problems solved with machine learning
3. Machine learning process
4. Machine learning tools overview
5. Machine learning challenges

Knowledge check

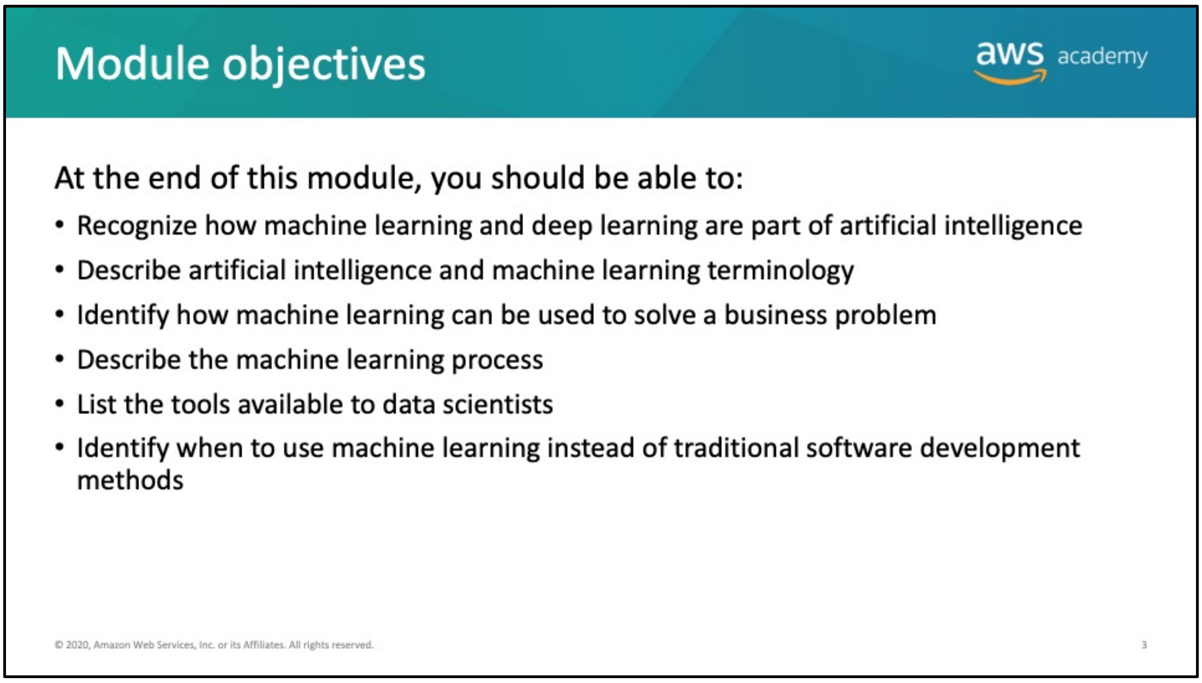
Demonstration

Introducing Amazon SageMaker

This module will address the following topics:

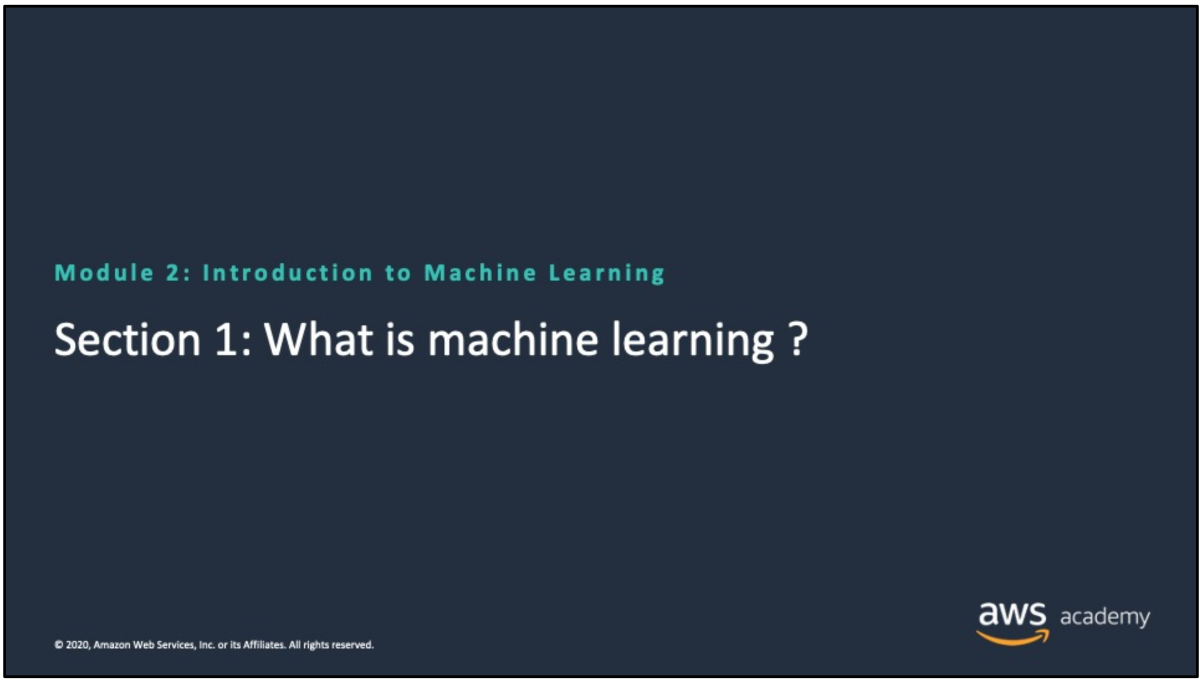
- What is machine learning?
- Business problems solved with machine learning
- Machine learning terminology and process
- Machine learning tools overview
- Machine learning challenges

You will be asked to complete a knowledge check that will test your understanding of key concepts covered in this module.

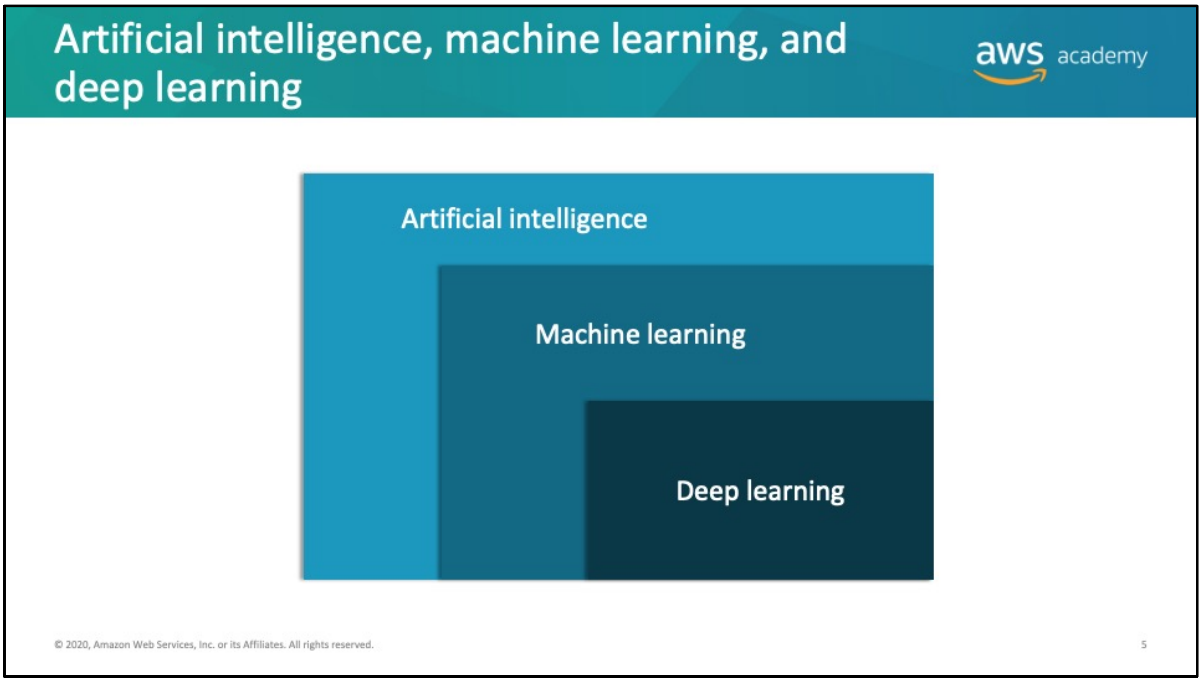


After completing this module, you should be able to:

- Recognize how machine learning and deep learning are part of artificial intelligence
- Describe artificial intelligence and machine learning terminology
- Identify how machine learning can be used to solve a business problem
- Describe the machine learning process
- List the tools available to data scientists
- Identify when to use machine learning instead of traditional software development methods



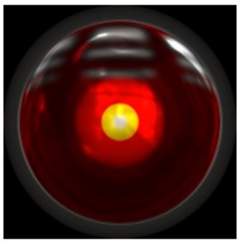
Introducing Section 1: What is machine learning?



This course is an introduction to machine learning (ML), and it shows where machine learning fits in with the larger picture of artificial intelligence (AI).

Machine learning is a subset of AI, which is a broad branch of computer science for building machines that can do human tasks. Deep learning itself a subdomain of machine learning. To understand where these ideas fit together, you will learn about each field.

Artificial intelligence

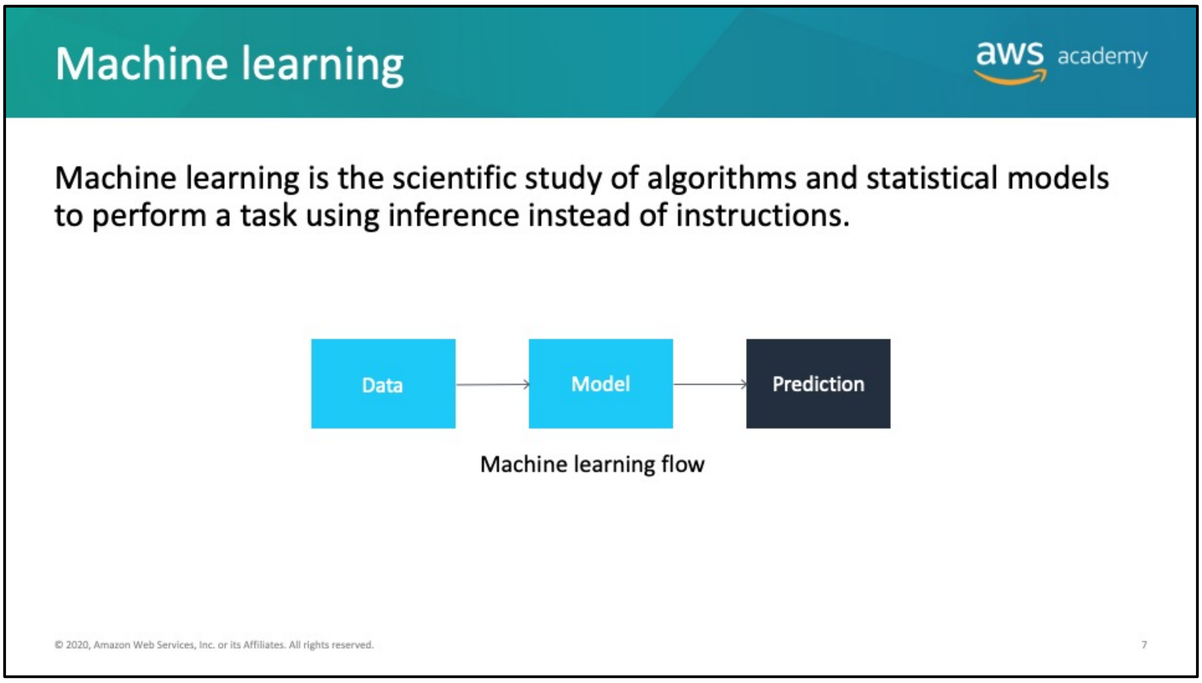

Machine learning is a subset of a broader computer science field that is known as artificial intelligence (AI).

AI is about building machines that can perform tasks that a human would typically perform. In modern culture, AIs appear in movies or works of fiction. You might recall some AIs in science fiction movies or TV shows that control the future world, or act intelligently on their own—sometimes, with negative effects for society or the human beings around them. These AIs started as computer agents that perceived their environments and took actions to achieve a specific goal. However, for some of these fictional AIs, their actions were not the outcome that their creators had originally envisioned.

Other fictional AIs are more benign or positive: they do a better job working with humanity, but they are also more general in purpose. These kinds of general AIs are examples of Artificial General Intelligence (AGI). They have the capability to learn or understand any task that a human being can understand.

AI problems typically span many fields of research: natural language processing, reasoning, knowledge representation, learning, perception, and physical environment interaction. AI isn't yet a reality, unless you are living in a simulation. However, it gets closer in each of these areas each year.

You might have also read or seen commentary on the ethics of creating AI. Not all views are positive—perhaps partly in fear of fictional AIs, which destroyed human beings or used them as power sources. Some might also consider the risk of mass unemployment because an intelligent machine can work continuously without breaks.



You will likely find many definitions of machine learning. No standard definition exists, so you will now learn some definitions of machine learning.

Machine learning is the scientific study of algorithms and statistical models to perform a task by using inference instead of instructions. To help you better understand this idea, consider the following concrete example.

Suppose that you must write an application that determines whether an email message is spam or not. Without machine learning, you write a complex series of decision statements (think *if/else* statements). Perhaps you use words in the subject or body, the number of links, and the length of the email to determine whether an email message is spam. It would be difficult and laborious to compile such a large set of rules to cover every possibility. However, with machine learning, you can use a list of email messages that are marked *spam* or *not spam* to train a machine learning model. The model would learn which patterns of words, lengths, and other indicators are good predictors of spam email messages. When you present the model with an email message that it did not see before, the model would predict whether it was *spam* or *not spam*.

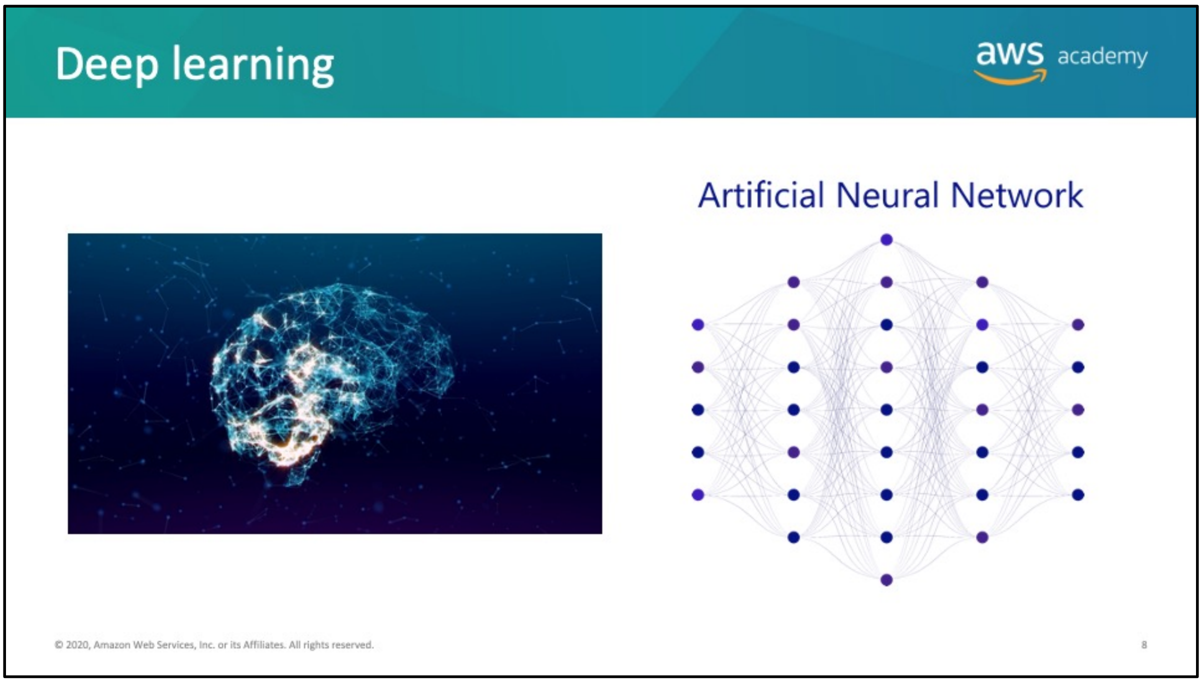
Tom Mitchell, a pioneer of machine learning, wrote this definition: “A computer program is said to learn from experience E with respect to some class of tasks T and performance

measure P, if its performance at tasks in T, as measured by P, improves with experience E."
(Mitchell, Tom. 1997. *Machine Learning*. McGraw Hill. p. 2.)

If you apply this concept to spam, the letters *E*, *T*, and *P* represent:

- E – The email messages that indicate spam or not
- T – The task of identifying spam
- P – The probability that the unseen email message is spam.

Don't be concerned if these ideas are difficult to understand. They are explained more completely later in this module.



Deep learning represents a significant leap forward in the capabilities for AI and ML. The theory behind deep learning was created from how the human brain works. An *artificial neural network (ANN)* is inspired from the biological neurons in the brain, although the implementation is different.

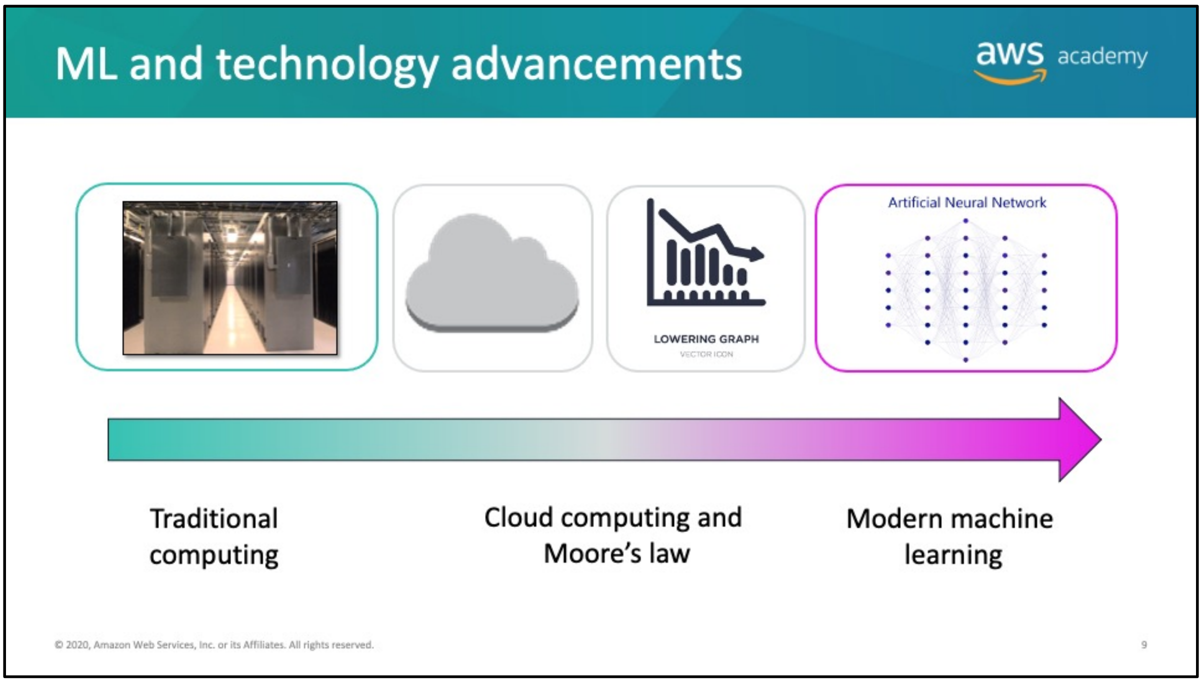
Artificial neurons have one or more inputs and a single output. These neurons fire (or activate their outputs), which are based on a transformation of the inputs.

A neural network is composed of layers of these artificial neurons, with connections between the layers. Typically, a network has input, output, and hidden layers.

The output of a single neuron connects to the inputs of all the neurons in the next layer. The network is then asked to solve a problem. The input layer is populated from the training data. The neurons activate throughout the layers until an answer is presented in the output layer. The accuracy of the output is then measured. If the output hasn't met your threshold, the training is repeated, but with slight changes to the weights of the connections between the neurons. It continues to repeat this process. Each time, it strengthens the connections that lead to success, and diminishes the connections that lead to failure.

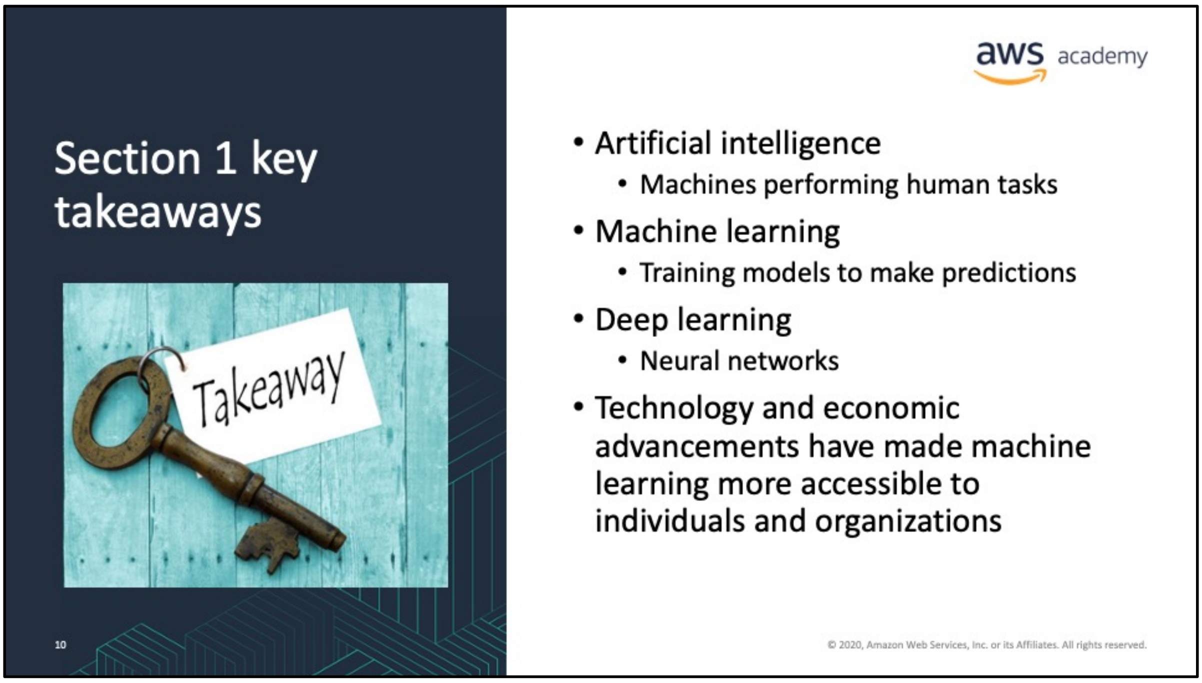
As you will see in this course, machine learning practitioners spend much time optimizing ML models. They select the best data features to train with, and they select the models with the best results. Deep learning practitioners spend almost no time on those tasks. Instead, they spend their time on modeling data with different ANN architectures.

While the theory for deep learning goes back decades, the hardware necessary to run deep learning problems wasn't generally accessible until recently. Now that it's available, you can use deep learning to address problems that are more complex than problems you handled before.



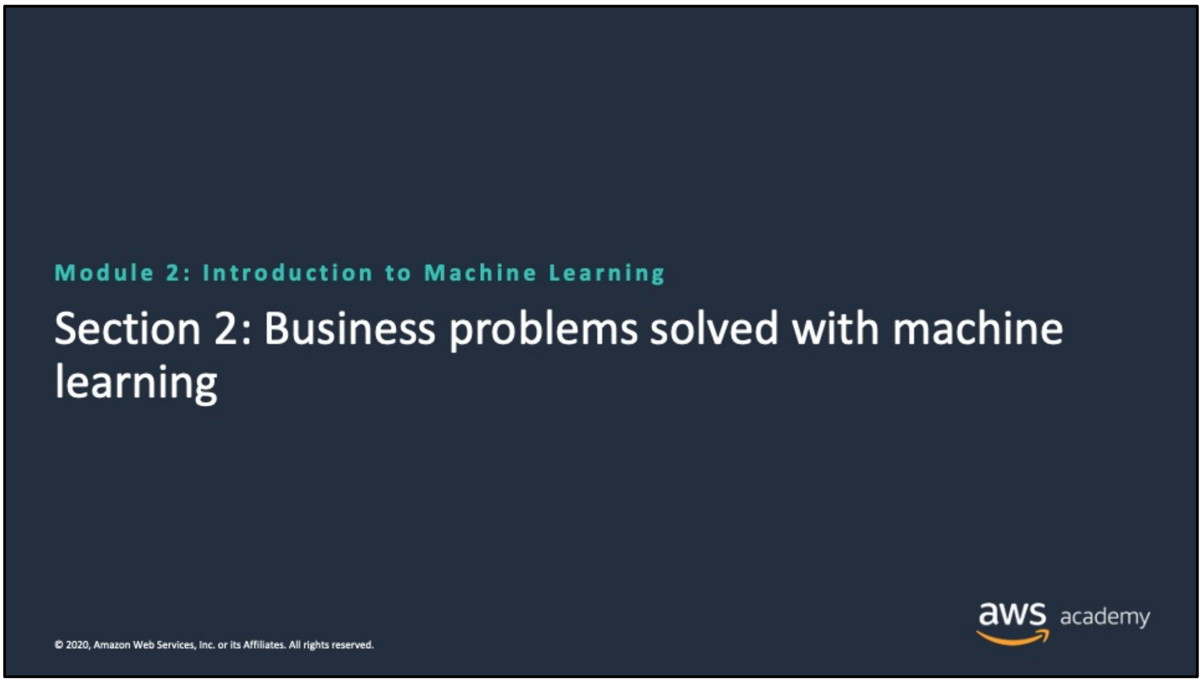
Mainstream machine learning is a fairly recent occurrence. The mid-2000s marked the beginning of rapid advancements in machine learning and deep learning, partly because of Moore's law and the rise of cloud computing. The result is easier access to larger, faster, and cheaper compute and storage capabilities. You can now rent computing power for a few hours for pennies, compared to huge investments that were needed to buy and operate large-scale compute clusters.

In 2012, the use of neural networks began in the ImageNet Large Scale Visual Recognition Challenge, a machine learning competition for image recognition. The accuracy rate jumped to about 82 percent, and has been steadily climbing ever since. It exceeded human performance in 2015.



Some key takeaways from this section of the module include:

- Artificial intelligence is the broad field of building machines to perform human tasks.
- Machine learning is a subset of AI. It focuses on using data to train ML models so the models can make predictions.
- Deep learning is a technique that was inspired from human biology. It uses layers of neurons to build networks that solve problems.
- Advancements in technology, cloud computing, and algorithm development have led to a rise in machine learning capabilities and applications.

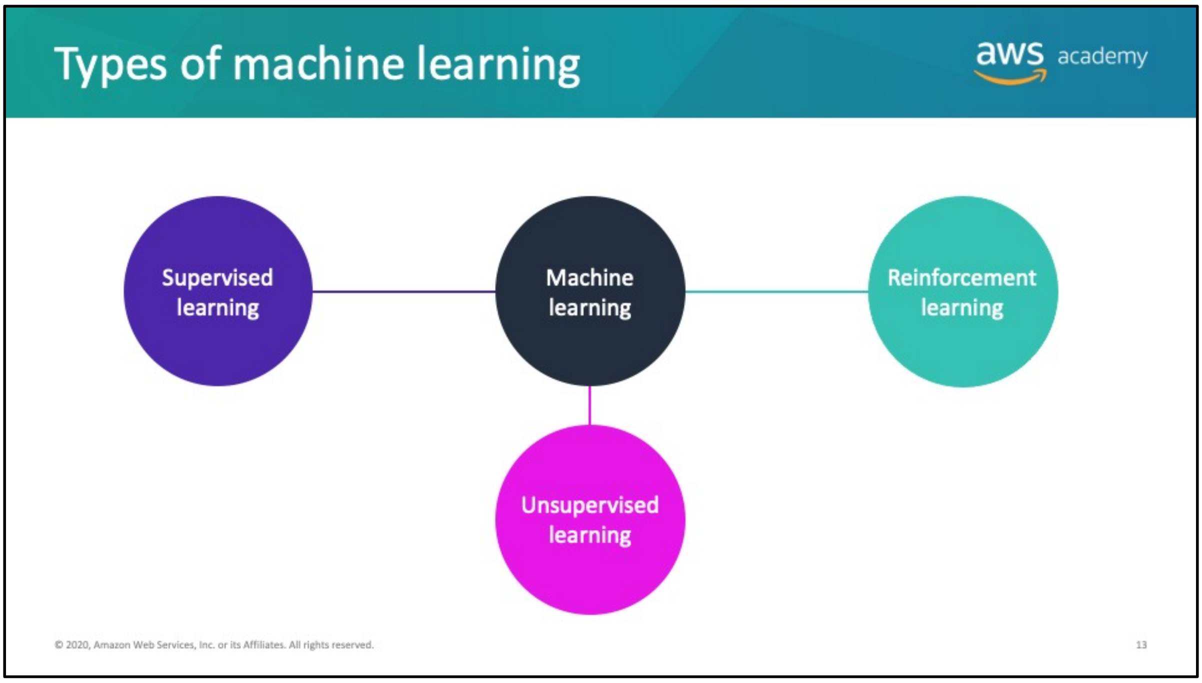


Introducing Section 2: Business problems solved with machine learning.

Machine learning is used throughout a person's digital life. Here are some examples:

- Spam – Your spam filter is the result of an ML program that was trained with examples of spam and regular email messages.
- Recommendations – Based on books that you read or products that you buy, ML programs predict other books or products that you might want. Again, the ML program was trained with data from other readers' habits and purchases.
- Credit card fraud – Similarly, the ML program was trained on examples of transactions that turned out to be fraudulent, along with transactions that were legitimate.

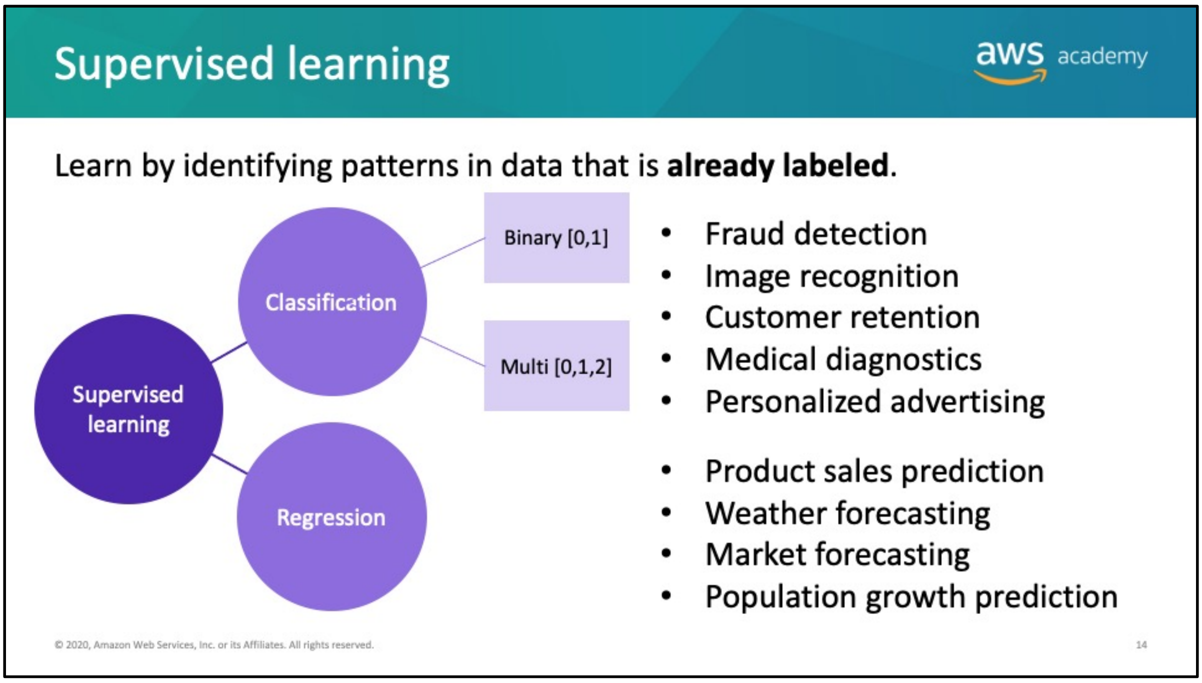
Many more examples exist, including facial detection in social media applications to group your photos, detecting brain tumors in brain scans, or finding anomalies in X-rays.



Machine learning has three main types.

The first type is *supervised learning*, where a model uses known inputs and outputs to generalize future outputs. The second type is *unsupervised learning*, where the model doesn't know inputs or outputs—it finds patterns in the data without help. The third type is *reinforcement learning*, where the model interacts with its environment and learns to take actions that maximize rewards.

It's important to know the different ML types because the type can guide you toward selecting algorithms that make sense for solving your business problem.



Supervised learning is a popular type of ML because it's widely applicable. It's called supervised learning because you need a supervisor—a teacher—who can show the right answers to the model.

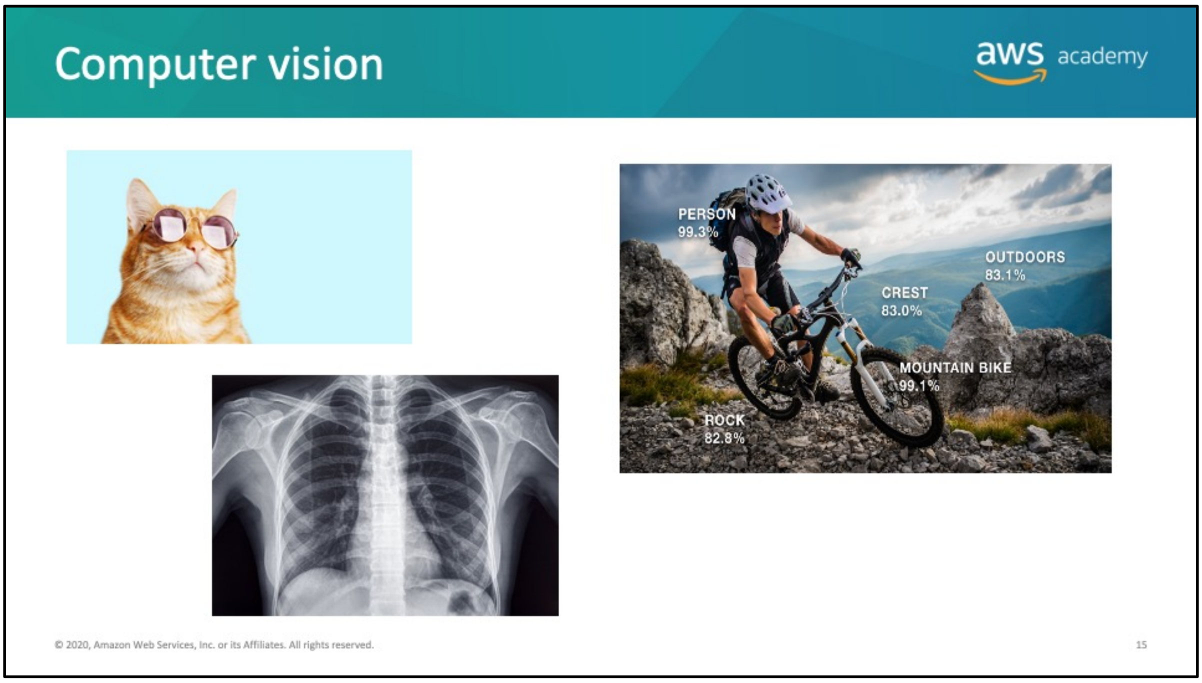
Like any student, a supervised algorithm learns by example. It needs a teacher who uses training data to help it determine the patterns and relationships between the inputs and outputs: "Here in this picture is a car. Here is a car in another picture." The model is trained on this labeled data to accurately identify where a car is in a new picture that it hasn't seen before.

However, you can have different types of problems within supervised learning. These problems can be broadly categorized into two categories: classification and regression.

Classification problems have two types. The first type is considered a *binary classification problem*. Recall the previous example about identifying fraudulent transactions. The target variable in this example is limited to two options: *fraudulent* or *not fraudulent*. This example is a binary classification problem because you are classifying an observation into one of two categories.

Multiclass classification problems also exist. These ML problems classify an observation into one of three or more categories. Suppose that you have an ML model that predicts why a customer is calling your store. Its purpose is to reduce the number of transfers necessary before getting the customer to the correct customer support department. In this case, the different customer support departments represent the various potential target variables—which might be many different departments.

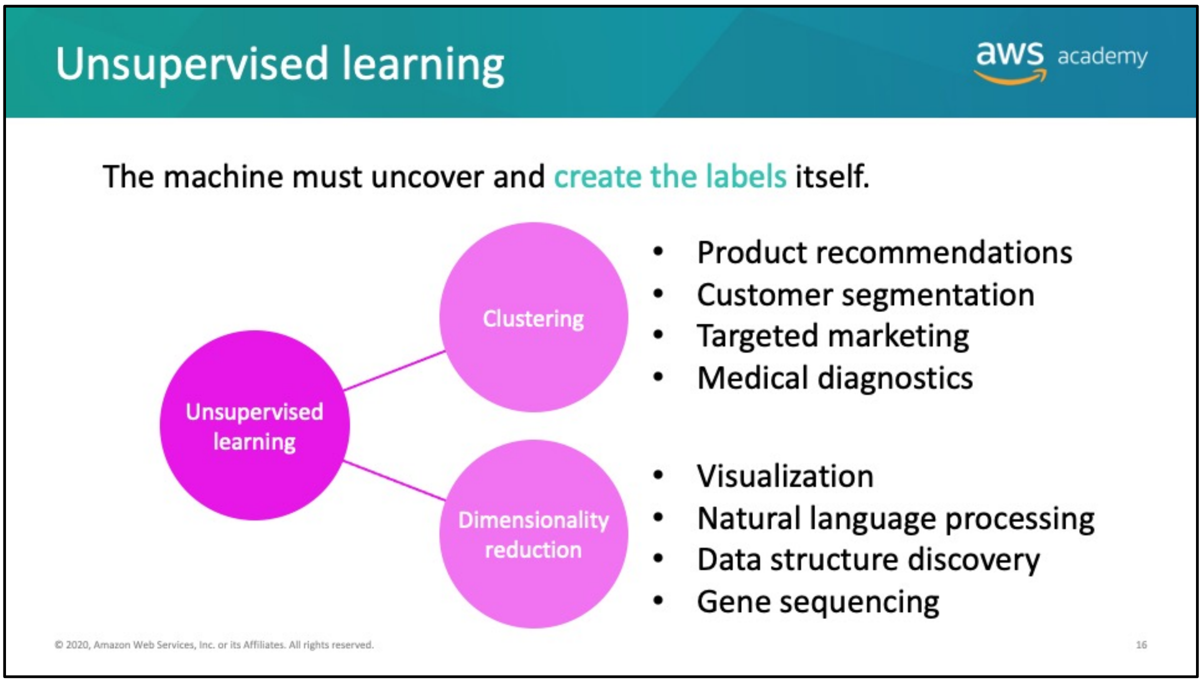
Regression problems also exist. In a regression problem, you are no longer mapping an input to a defined number of categories. Instead, you are mapping inputs to a continuous value, like an integer. One example of an ML regression problem is predicting the price of a company's stock. For instance, a regression-based algorithm would predict that tomorrow, the stock price for a company could go up from $113 per share to $127 per share.



Computer vision (CV) is a large field that consists mostly of classification problems. Is this a cat or a dog? Does this X-ray show a tumor?

Computer vision enables machines to identify people, places, and things in images with accuracy at (or above) human levels, and with greater speed and efficiency. It’s often built with deep learning models. CV automates the extraction, analysis, classification, and understanding of useful information from a single image or a sequence of images. The image data can take many forms, such as single images, video sequences, views from multiple cameras, or three-dimensional data.

You will learn more about computer vision later in this course.

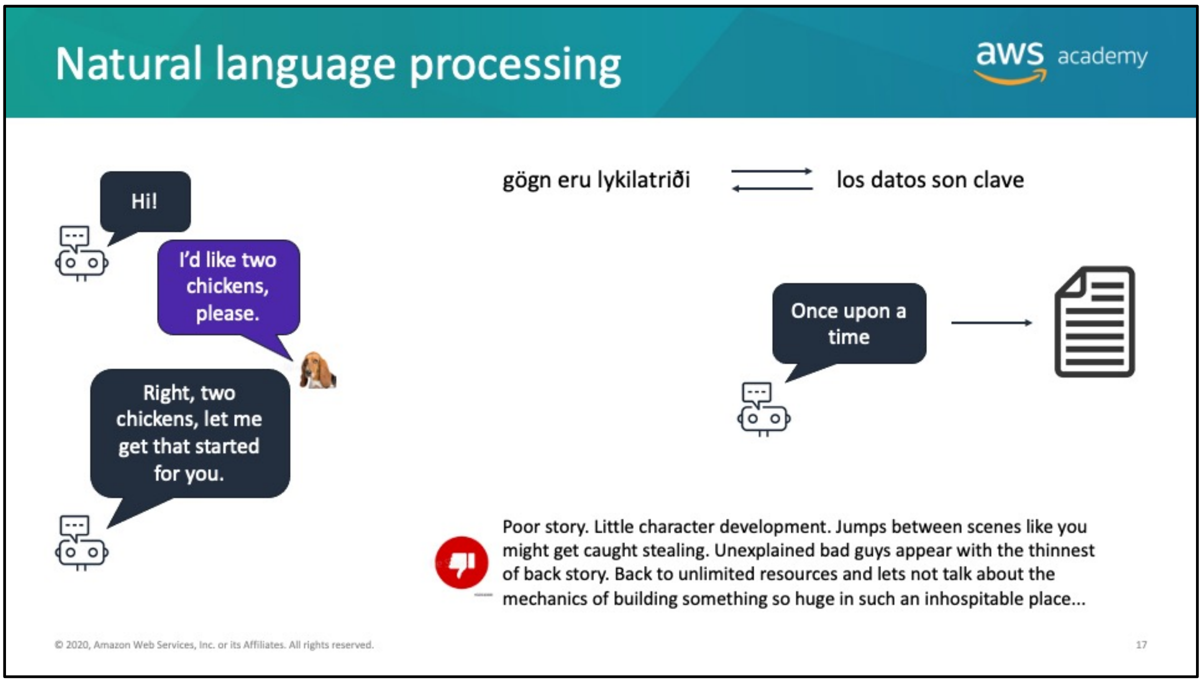


You will now learn about unsupervised machine learning.

Sometimes, the data is all that you have—no supervisor is available. In unsupervised learning, labels are not provided (like they are with supervised learning) because you don't know all the variables and patterns. In these instances, the machine must uncover and create the labels itself. These models use the data that they are presented with to detect emerging properties of the entire dataset, and then construct patterns.

A common subcategory of unsupervised learning is called clustering. This kind of algorithm groups data into different clusters that are based on similar features to better understand the attributes of a specific cluster. For example, suppose you work for a marketing organization that supports companies that provide cloud-computing resources to their customers. By analyzing customer purchasing habits, unsupervised algorithms can identify groups of customers that categorize (or describe) a particular company as being small, medium, or large. The companies can use these insights to make informed strategic marketing decisions based which tier they are in (small, medium, or large) so they can better target their customers' needs for cloud-computing resources. In this situation, clustering might help you realize that you must propose a different marketing strategy for different-sized companies.

The advantage of unsupervised algorithms is that they enable you to see patterns in the data that you were otherwise unaware of. An example might be the existence of two major customer types.

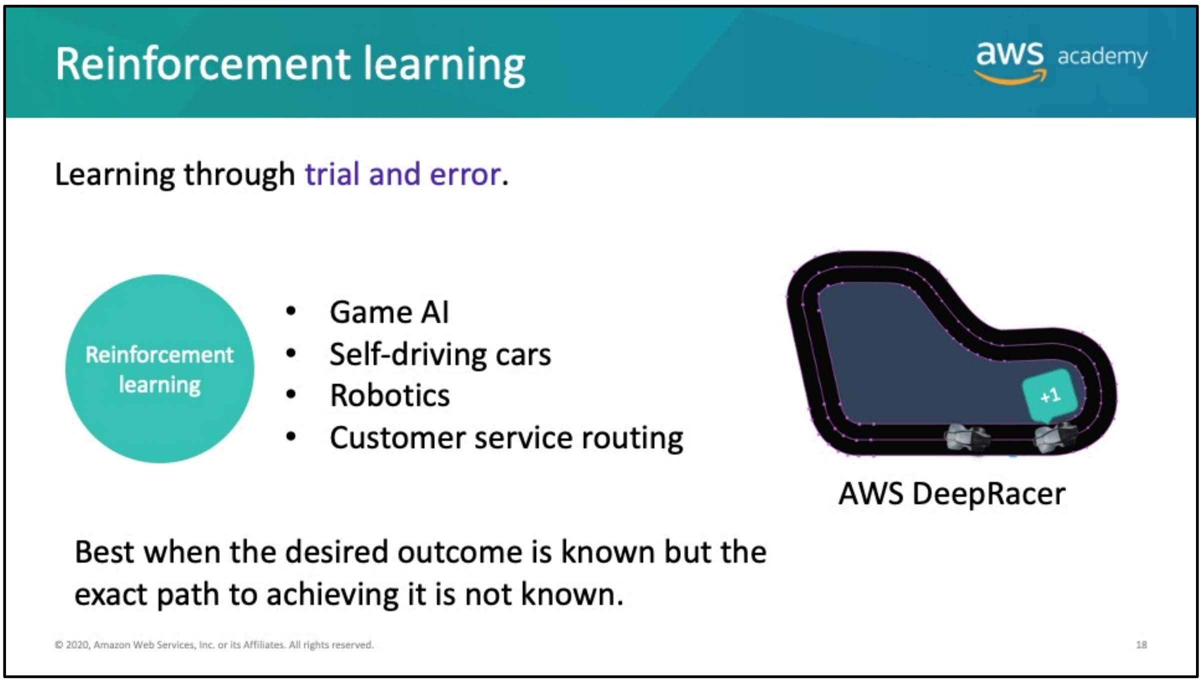


Natural language processing (NLP) is another area of machine learning with increasing use. For example, Amazon Alexa (or any other voice assistant) uses NLP to try to answer your questions. NLP is about speech, and it is also about written text.

NLP is used in many applications such as:

- Chat or call center bots – Automated systems for getting your bank balance or ordering food from a restaurant.
- Translation tools – Converting text between languages, or applications that can translate menus in real time.
- Voice-to-text translations – Converting spoken words into text. Can be used to power automatic subtitles.
- Sentiment analysis – Enables you to analyze the sentiment of comments in reviews of products, music, and movies. These sentiments can be used to give the movie an audience rating.

You will learn about NLP in a later module.



Another kind of machine learning that's been gaining in popularity recently is *reinforcement learning*. Unlike other machine learning, reinforcement learning continuously improves its model by mining feedback from previous iterations. In reinforcement learning, an agent continuously learns, through trial and error, as it interacts in an environment. Reinforcement learning is useful when the reward of an intended outcome is known, but the path to achieving it is not. Discovering that path requires much trial and error.

Consider the example of AWS DeepRacer. In the AWS DeepRacer simulator, the agent is the virtual car, and the environment is a virtual racetrack. The actions are the throttle and steering inputs to the car. The goal is to complete the racetrack as quickly as possible without deviating from the track.

The car must learn the intended driving behavior to reach the goal of completing the track. You use rewards to incentivize your model to learn the intended driving behavior.

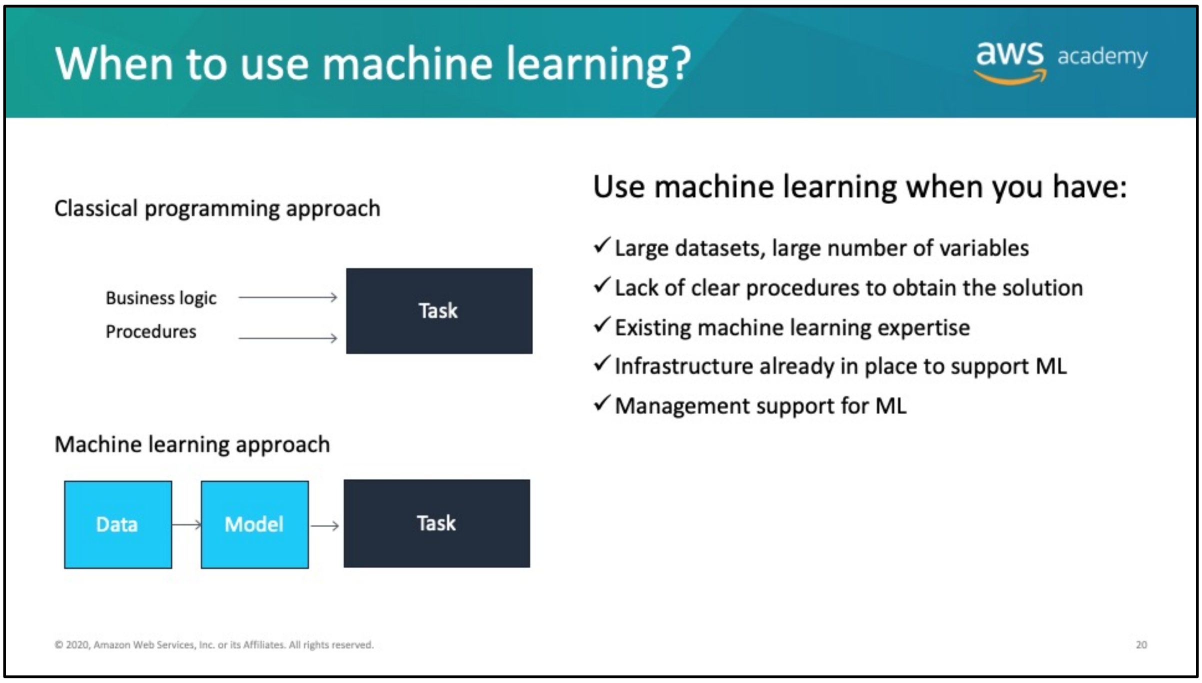
In reinforcement learning, what drives the learning is called the *agent*. In this case, the agent is the AWS DeepRacer car. The *environment* is the place where the agent learns, which—in this example—is the marked race track. When the agent does something in the environment that provokes a response, such as crossing a boundary that it shouldn't cross, it's called an *action*. That response is called a *reward* or *penalty* depending on whether the agent did something that should be reinforced or discouraged in the model. As the agent moves in the environment, its actions should continue to receive more rewards and fewer penalties, until it meets the intended business outcome.

Self-driving vehicles

aws academy

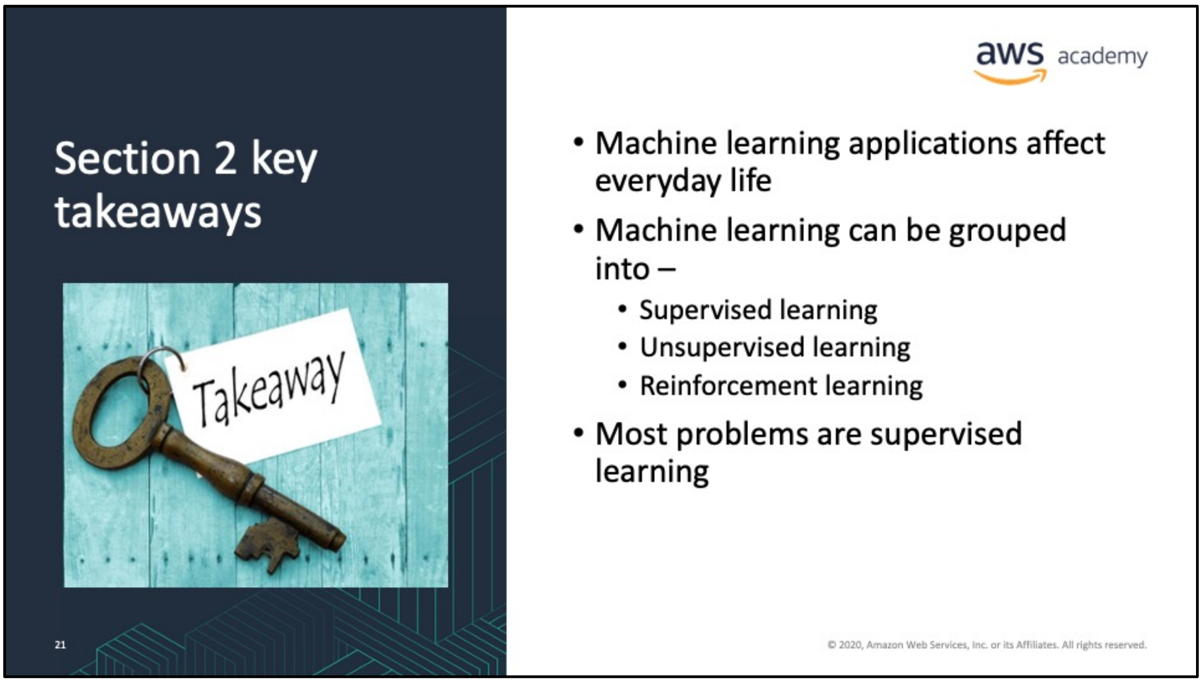
Self-driving vehicles bring together several machine- and deep-learning algorithms and models to solve the problem of driving from A to B. A major task is continuously detecting the environment and forecasting changes. This task involves object detection, which is the localization and prediction of movement of the detected object. The outputs of these findings act as inputs to other systems that make decisions on what to do with the vehicle's various controls.

Some use cases involve self-driving vehicles that require real-time responses to the environment. For example, if a previously hidden pedestrian walks out from behind an obstacle, the vehicle brakes must be applied immediately. Such actions cannot have any latency or room for error.



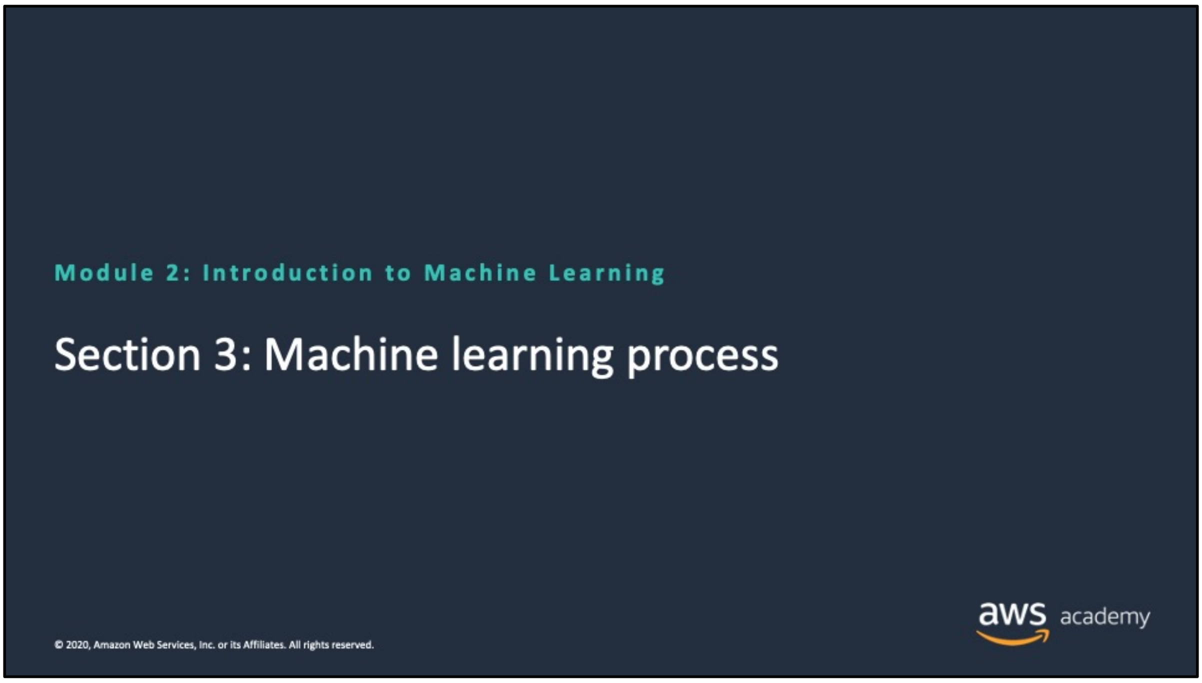
Not every problem should be solved with machine learning, and sometimes basic programming works equally well. When you explore a potential ML solution, look for things such as the existence of large datasets and a large number of variables. You might also be uncertain of the business logic or procedures that are required to obtain an answer or accomplish a task. In such cases, machine learning is often the best choice.

Machine learning systems can be complex. The supporting infrastructure, management support, and expertise must be in place to help the project be successful.



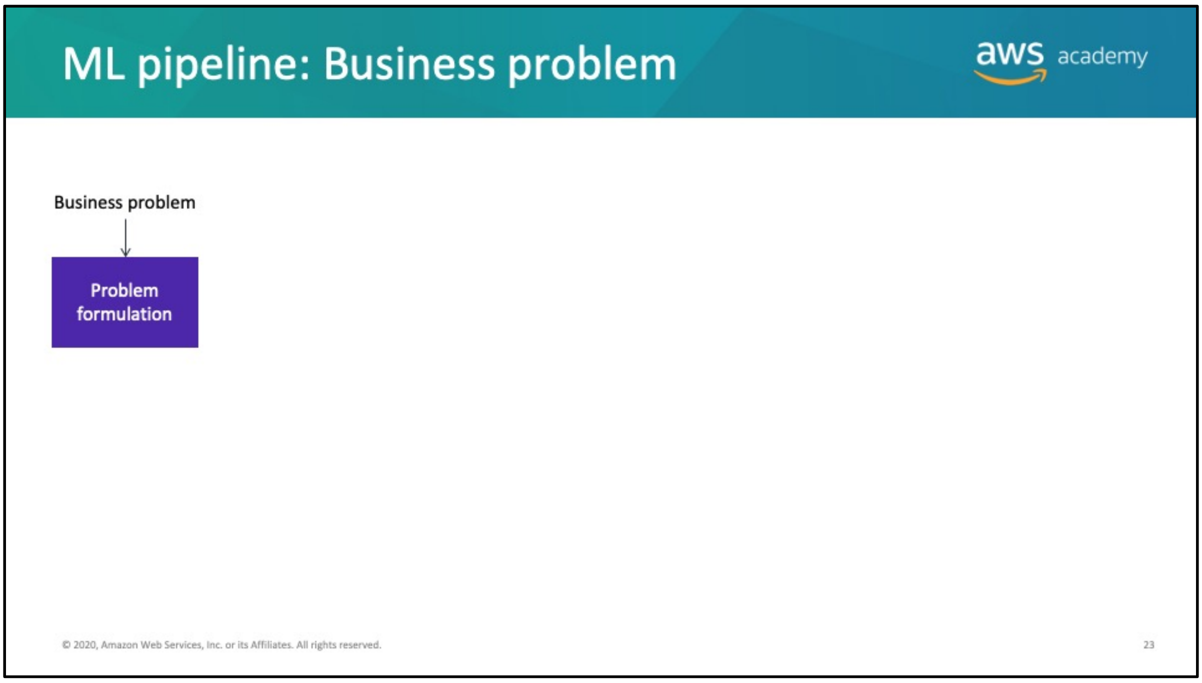
Some key takeaways from this section of the module include:

- Machine learning applications are already part of your everyday life.
- Machine learning problems can be grouped into –
    - Supervised learning: You have training data for which you know the answer.
    - Unsupervised learning: You have data, but you are looking for insights within the data.
    - Reinforcement learning: The model learns in a way that is based on experience and feedback.
- Most business problems are supervised learning.

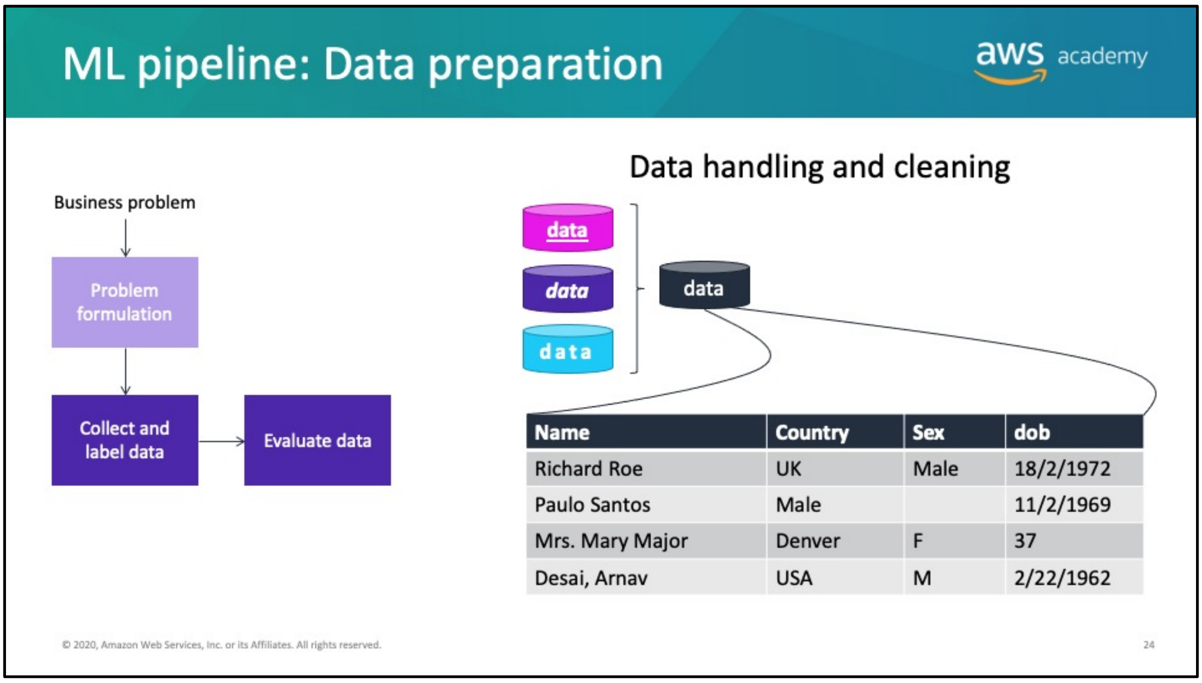


Introducing Section 3: Machine learning process.

In this section, you will get a high-level overview of machine learning terminology and a typical workflow. You will learn more details about this topic later in this course.



To begin, you should always start with the business problem that you or your team believes can benefit from ML. From there, you want to do some problem formulation. This phase entails, in part, articulating your business problem and converting it to an ML problem.

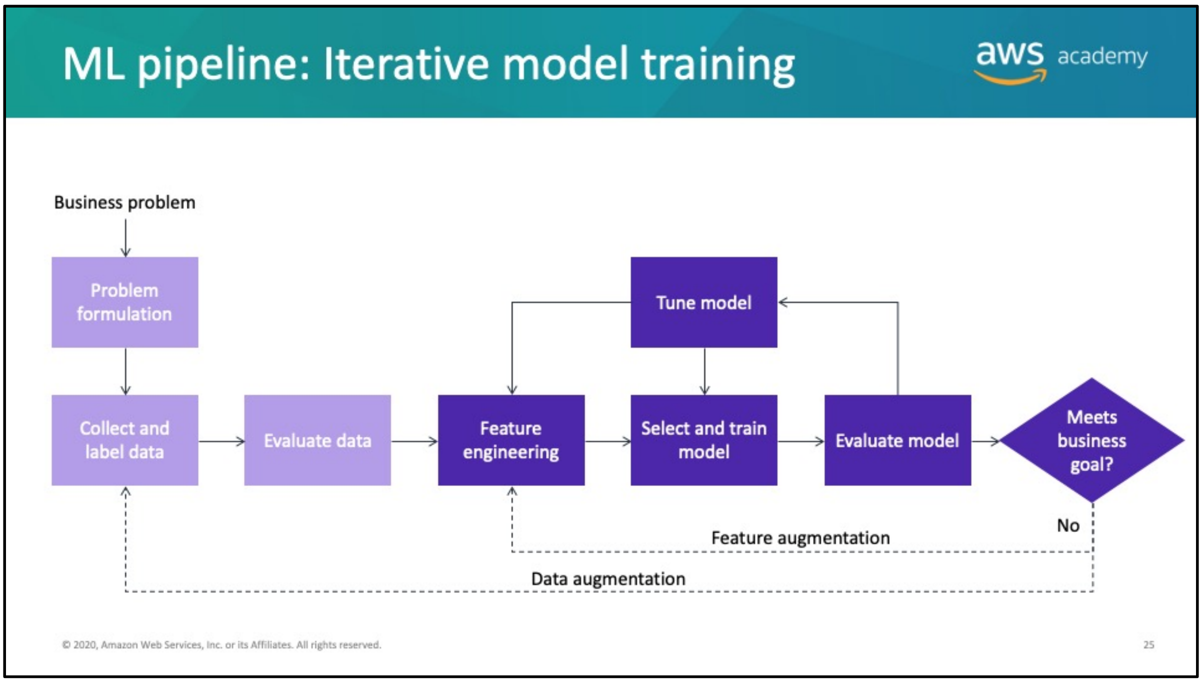


After you formulate the problem, you move to the data preparation and preprocessing phase. In this phase, you will extract data from one or more data sources. These data sources might have differences in data or types that must be reconciled to form a single cohesive view of your data. You must visualize your data and use statistics to determine whether the data is consistent and can be used for machine learning. You will look at some data sources later in the course.

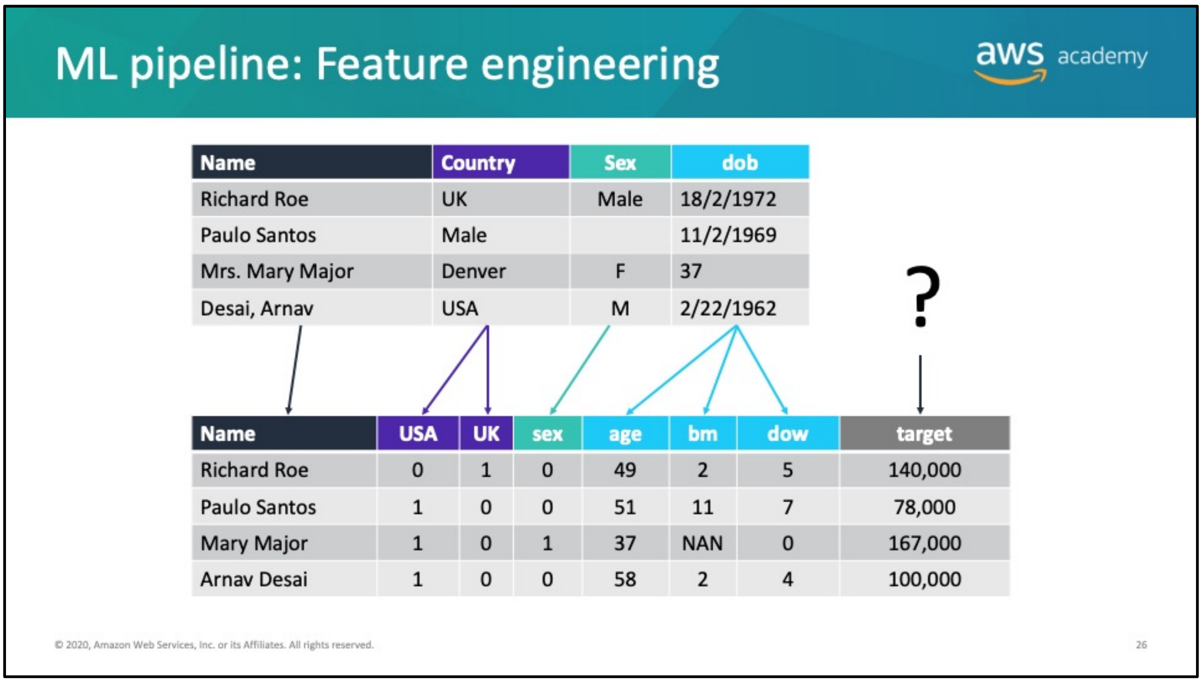
In the example data, you have four columns that contain data that was assembled from three data sources. The sources had slightly different ways of representing data, and the results are shown in the table. In ML problems, columns represent features, and rows represent instances. You can see some issues with the data in some of the instances.

In some cases, you need a subject matter expert or a functional expert to understand the authenticity of the data. For example, *11/2/1969* can be *November 2* or *February 11*. Someone who owns or manages the data pool would be able to clarify what that data means. *Male* can probably be attributed to an import issue where cells shifted position, but it can also mean a location: Male, the capital of Republic of Maldives. At times, this error identification is not as simple and will require an SME. You will learn about the role of an expert later in the course.

You can have a substantial effect on the success of an ML project by having consistent and correct data.



After you prepare your data and ensure its correctness and consistency, it's time to train your model. At this stage, the process becomes iterative and fluid. You are likely to encounter multiple passes of feature engineering, training, evaluating, and tuning before you find a model that can meet your business goals.



Feature engineering is the process of selecting or creating the features that you will use to train your model. *Features* are the columns of data that you have within your dataset. The goal of the model is to try to correctly estimate the target value for new data. The ML algorithm uses the features to predict the *target*. In this example, the target data is the average number of steps taken in a week.

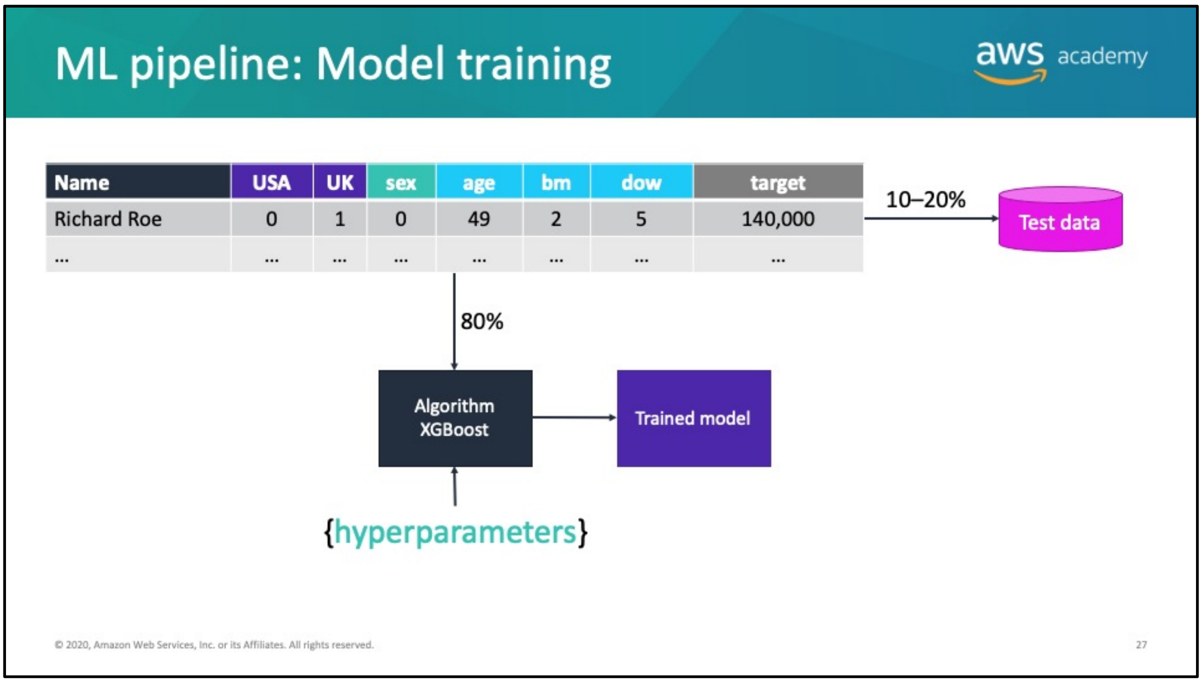
Selecting the correct features might involve adding, removing, or calculating new features. You might encounter these changes during feature engineering:

- **Data cleanup on the Name column –** You might want to make the format consistent, which can be useful later in the model. Or, you could change the format only for cosmetic reasons. Depending on the problem that you are trying to solve with this data, you might not even need the *Name*.
- **Convert the country into a numeric or binary set of columns –** If this database was a traditional database, you might want to move *Country* to a lookup table and then reference it. Most ML algorithms want the data for an instance in a single row. Also, ML algorithms need numerical data to process. You might consider turning the country text into the country's ISO code. However, the model might interpret the numerical value as having meaning, so *UK (44)* would be more significant than, for example, *USA (01)*. In this case, splitting the data into multiple columns is good. This practice is known as *categorical*

encoding, which you will learn about later in this course.

- **Convert the sex data into a binary –** If you convert the text value into a numeric value, with *0* or *1* representing *male* or *female*, the model can use it more easily.
- **Split the date of birth (DOB) into component parts –** Extracting age, birth month (*bm*) and day of week (*dow*) might be appropriate, depending on the problem that you are trying to solve. Does age affect the target variable? What about which day of the week they were born on?

Don't be concerned if this information seems complicated. You learn more details about feature engineering later in this course.

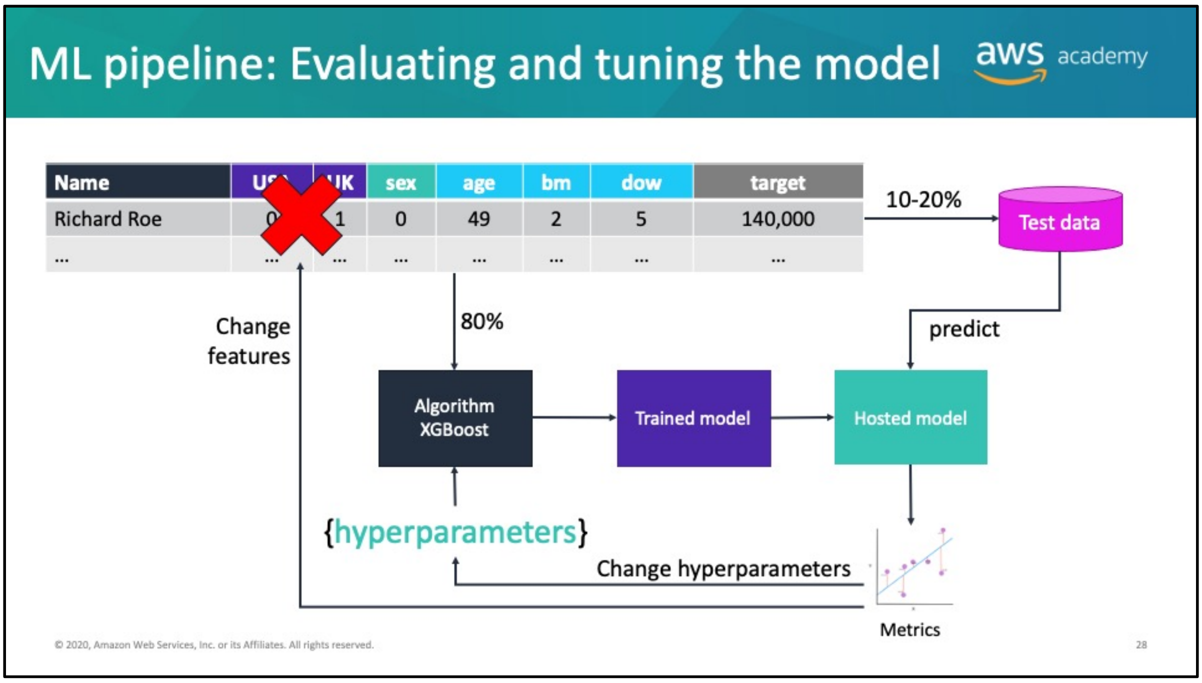


After you clean your data and identify its features, it's time to train a model. You don't use all the data to train your model. Instead, you should hold some data back so that you can have some data to test with. Typically, you use about 80 percent of the data to train with, and save the rest for testing.

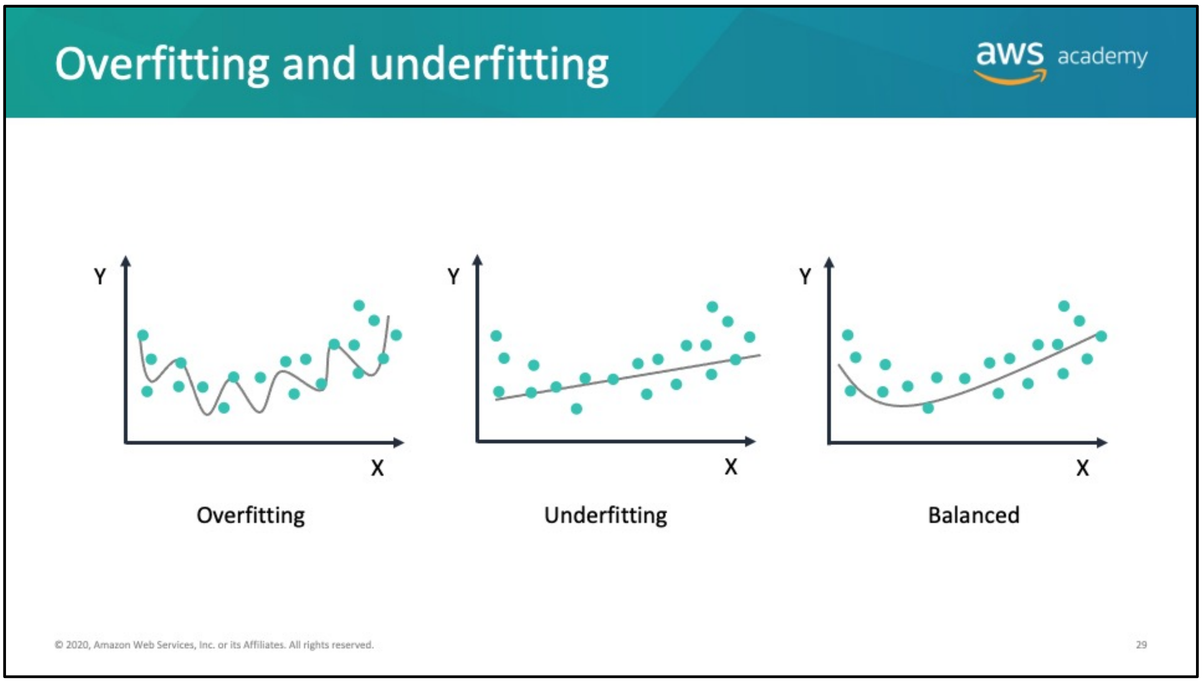
You use the training data to train a model. In the diagram, the model being used is the XGBoost algorithm.

The model itself has some parameters that you can set, and they alter how the algorithm works. These parameters are known as *hyperparameters*.

The output of the training job is a trained model.



With the trained model, you can use some of the test data to see how well the model performs. You can take an instance that the model has not seen, and use it to perform a prediction. Because you already know the target in your test data, you can compare the two values. From these comparisons, you can calculate metrics, which give you data on how well the model performs. You then modify the model's data, features, or hyperparameters until you find the model that yields the best results.



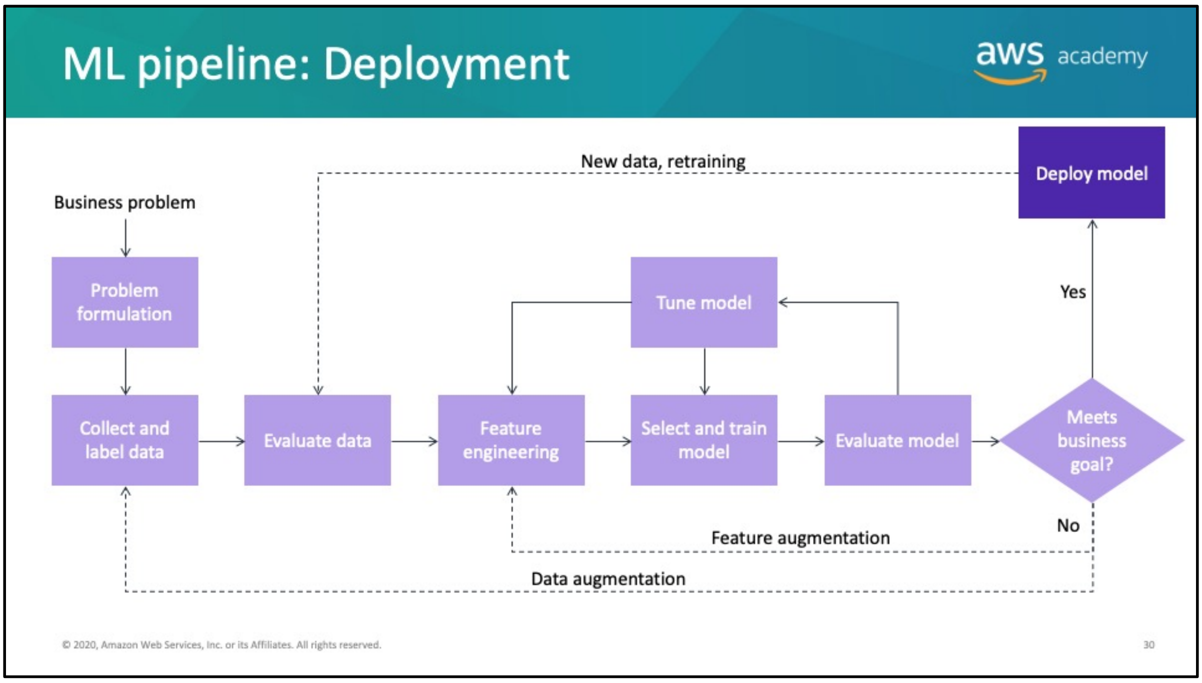
When you train your model, recognize the danger of overfitting or underfitting the model.

Your model is *overfitting* your training data when it performs well on the training data, but it doesn't perform well on the evaluation data. The model memorizes the data that it saw, and it can't generalize to unseen examples—which causes the overfitting.

Your model is *underfitting* the training data when the model performs poorly on the training data. The model can't capture the relationship between the input examples (often called *X*) and the target values (often called *Y*). This inability causes the underfitting.

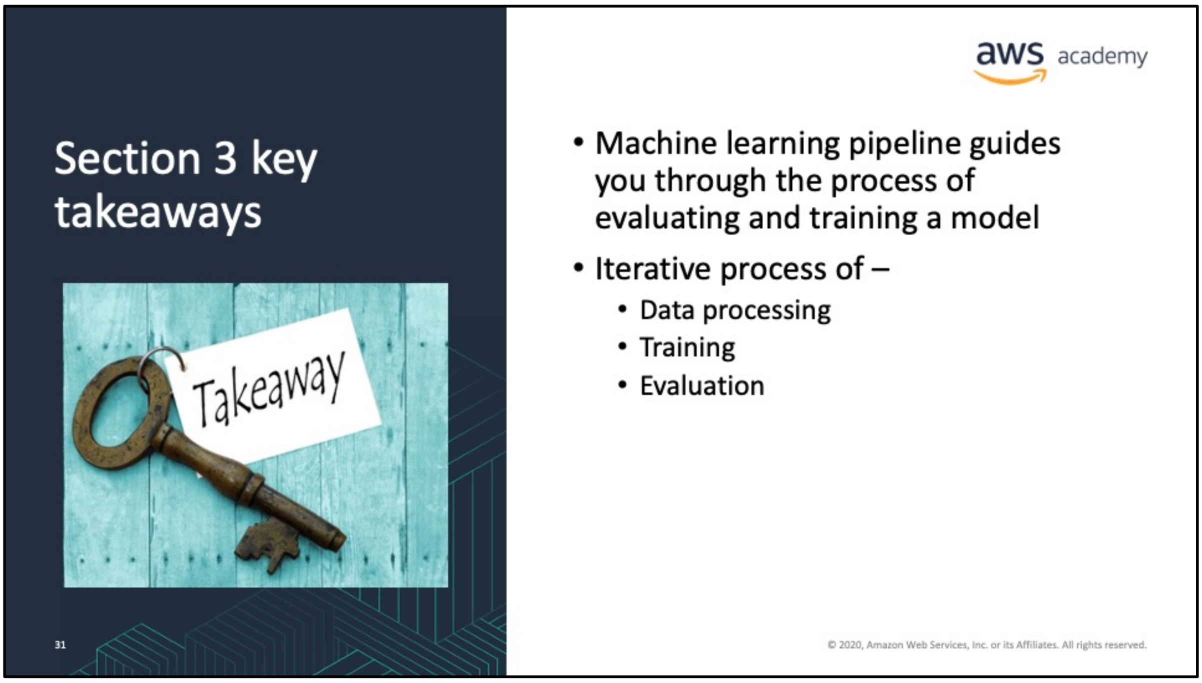
Understanding *model fit* is important for finding the root cause for poor model accuracy. This concept guides you to take corrective steps to determine whether a predictive model is underfitting or overfitting the training data. These steps involve looking at the prediction error on the training data and the evaluation data.

You will learn steps that you can take to avoid this issue later in this course.



After you retrain the model and you are satisfied with the results, your model is deployed to deliver the best possible predictions.

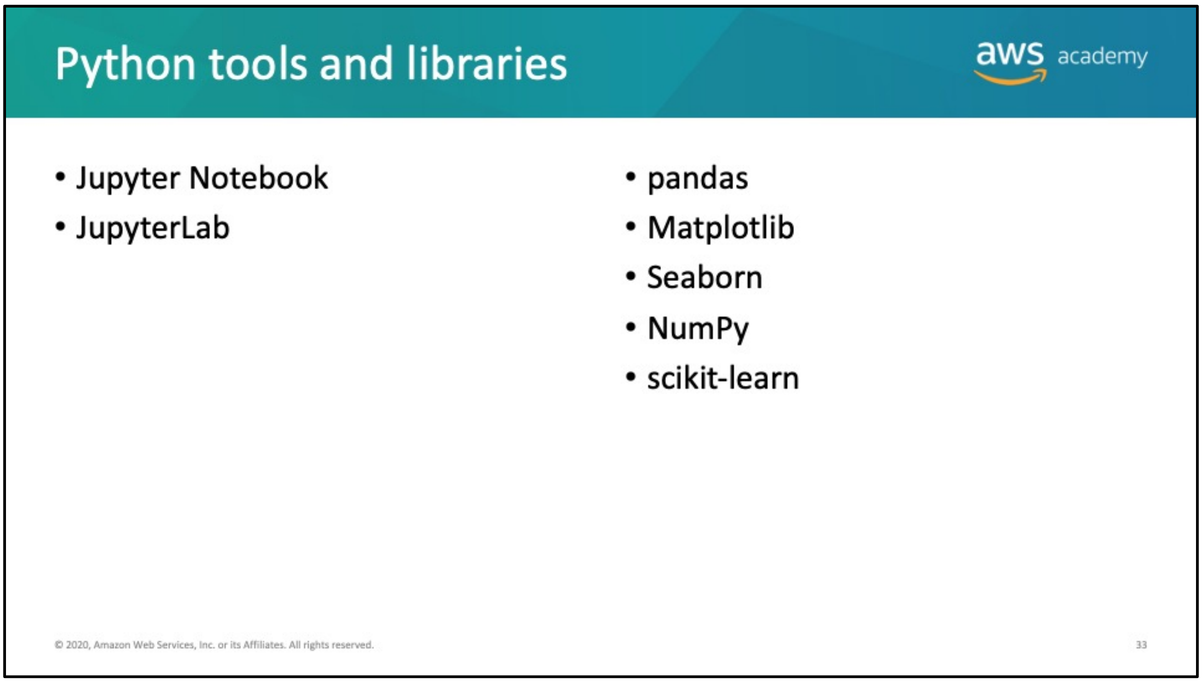
Later in this course, you will walk you through these different phases and gain hands-on experience with each of them. Knowing the process is also useful when you use the managed services that you will explore. However, Amazon ML services do most of the hard work for you.



Some key takeaways from this section of the module include:

- The machine learning pipeline process can guide you through the process of training and evaluating a model.
- The iterative process can be broken into three broad steps –
    - Data processing
    - Model training
    - Model evaluation

Introducing Section 4: Machine learning tools overview.



You will now learn about some of the tools that you will use for machine learning. Note: This list is not a complete list of the ML tools that are available today, and you will use only a few of them in this course.

Jupyter Notebook is an open-source web application that enables you to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible. You can configure and arrange the user interface to support a range of workflows in data science, scientific computing, and machine learning. JupyterLab is also extensible and modular. You can write plugins that add new components and integrate with existing ones.

Later in this course, you will use Amazon SageMaker, which hosts Jupyter notebooks and JupyterLab.

pandas is an open-source Python library. It's used for data handling and analysis. It represents data in a table that is similar to a spreadsheet. This table is known as a *pandas*

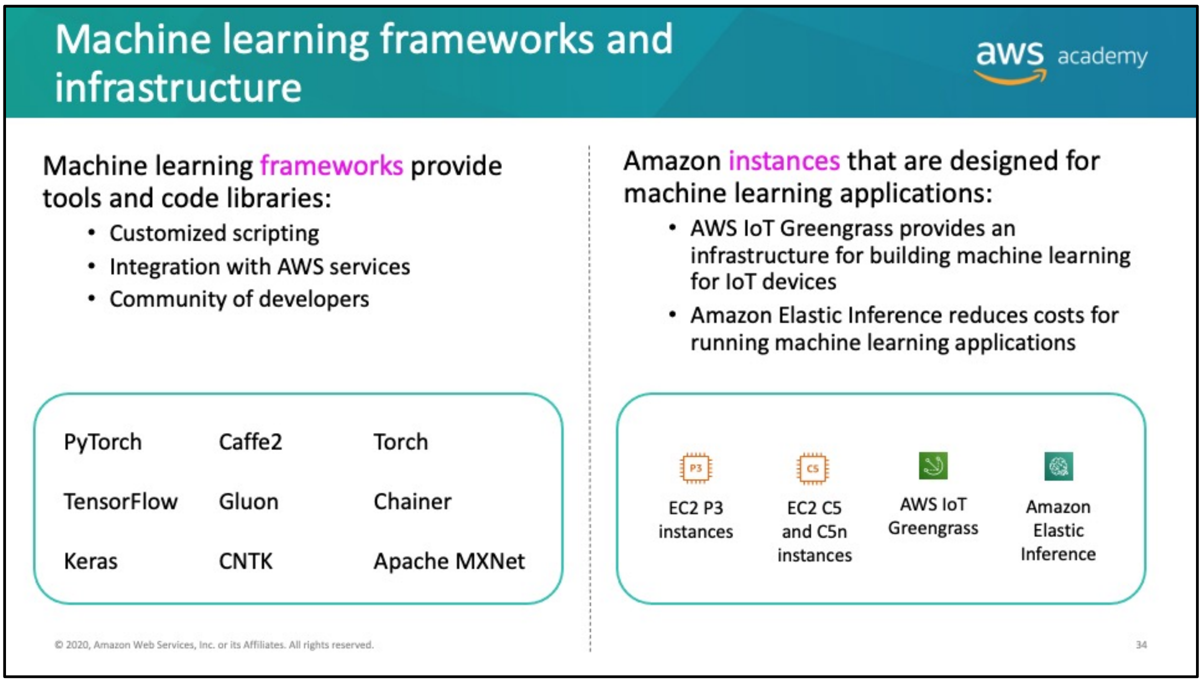
DataFrame.

Matplotlib is a library for creating scientific static, animated, and interactive visualizations in Python. You use it to generate plots of your data later in this course.

Seaborn is another data visualization library for Python. It's built on matplotlib, and it provides a high-level interface for drawing informative statistical graphics.

NumPy is one of the fundamental scientific computing packages in Python. It contains functions for N-dimensional array objects and useful math functions such as linear algebra, Fourier transform, and random number capabilities.

scikit-learn is an open-source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities. scikit-learn is built on NumPy, SciPy, and matplotlib, and it's a good package for exploring machine learning. Although you use it only to borrow a few functions in a later module, you can explore this package in greater detail after you complete this course.

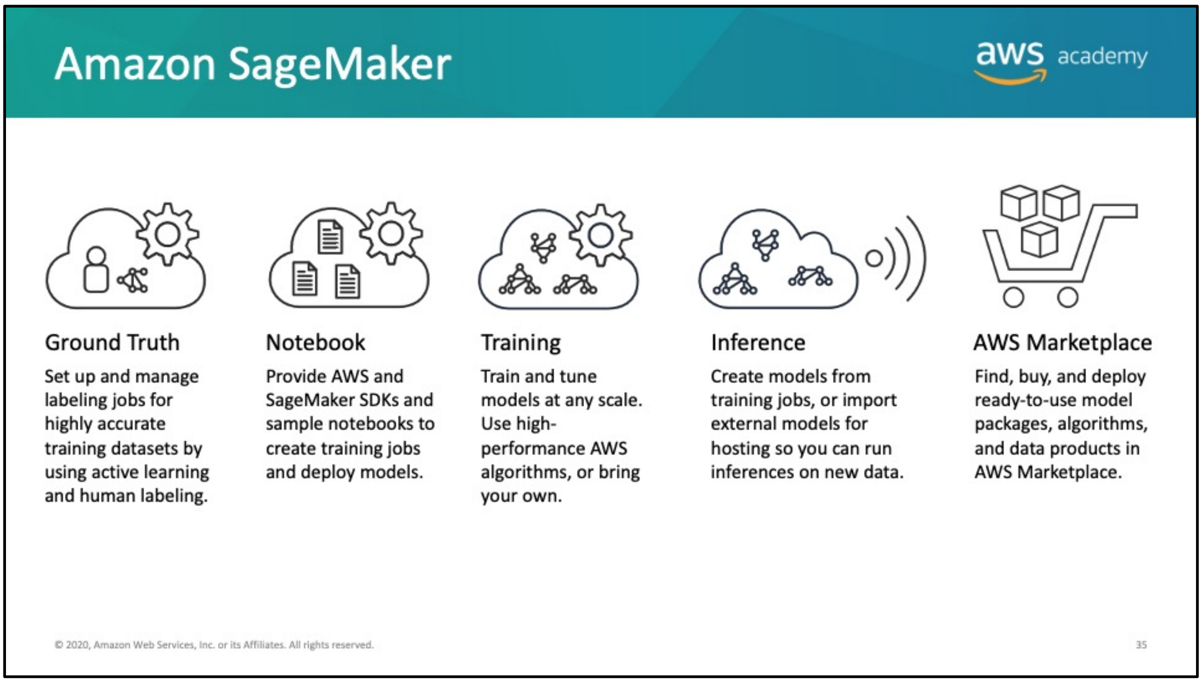


Moving up from individual libraries and packages, you can use tools that contain production-ready frameworks.

You briefly learned about scikit-learn, which is a library that works well for learning. You can also use other libraries, such as TensorFlow for machine learning and Keras for deep learning.

All these frameworks are supported on AWS, and can be used from Amazon SageMaker.

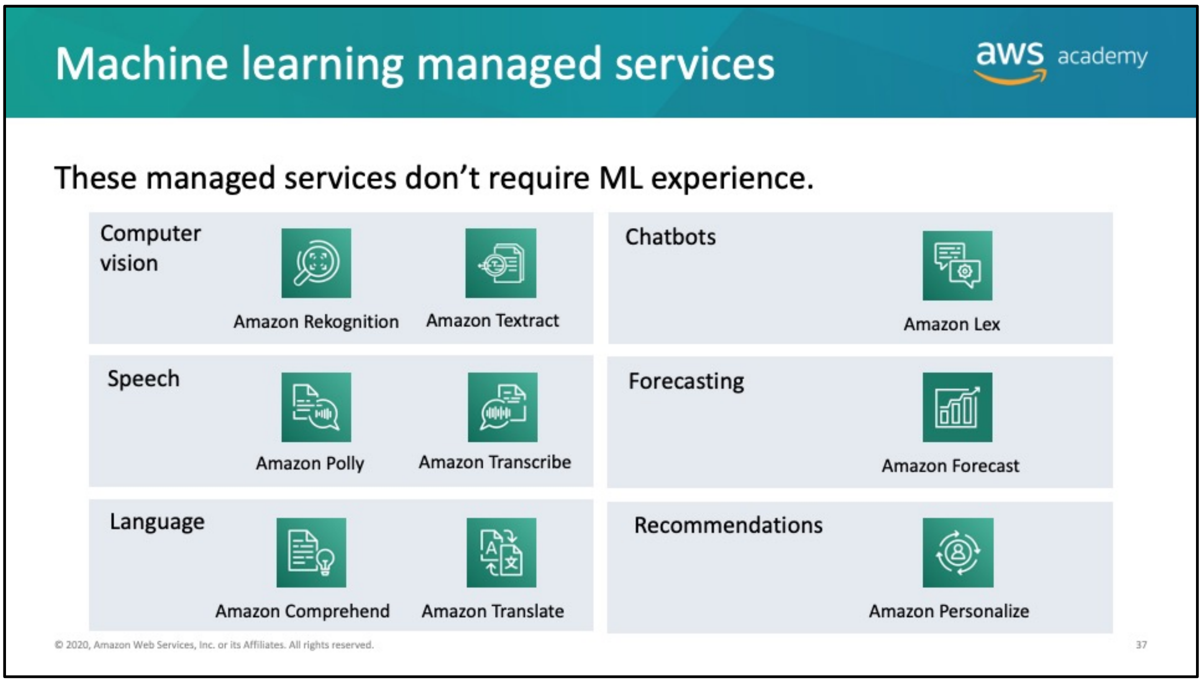
Amazon also provides compute instances that are tuned for machine learning, in both the cloud and at the edge. Compute instances are optimized for learning and inference, and pre-packaged Amazon Machine Images (AMIs) that contain many popular frameworks.



Finally, you can use Amazon SageMaker, which is an AWS service with many capabilities.

Amazon SageMaker can deploy machine learning instances that run Jupyter notebooks and JupyterLab. Amazon SageMaker manages the deployment of these compute resources, so you must connect to the Jupyter environment. Amazon SageMaker also provides tools for labeling data, training models, and hosting trained models. AWS Marketplace also provides a selection of ready-to-use model packages and algorithms from machine learning developers.

Your instructor will now either demonstrate Amazon SageMaker or provide you with access to a recorded demonstration.

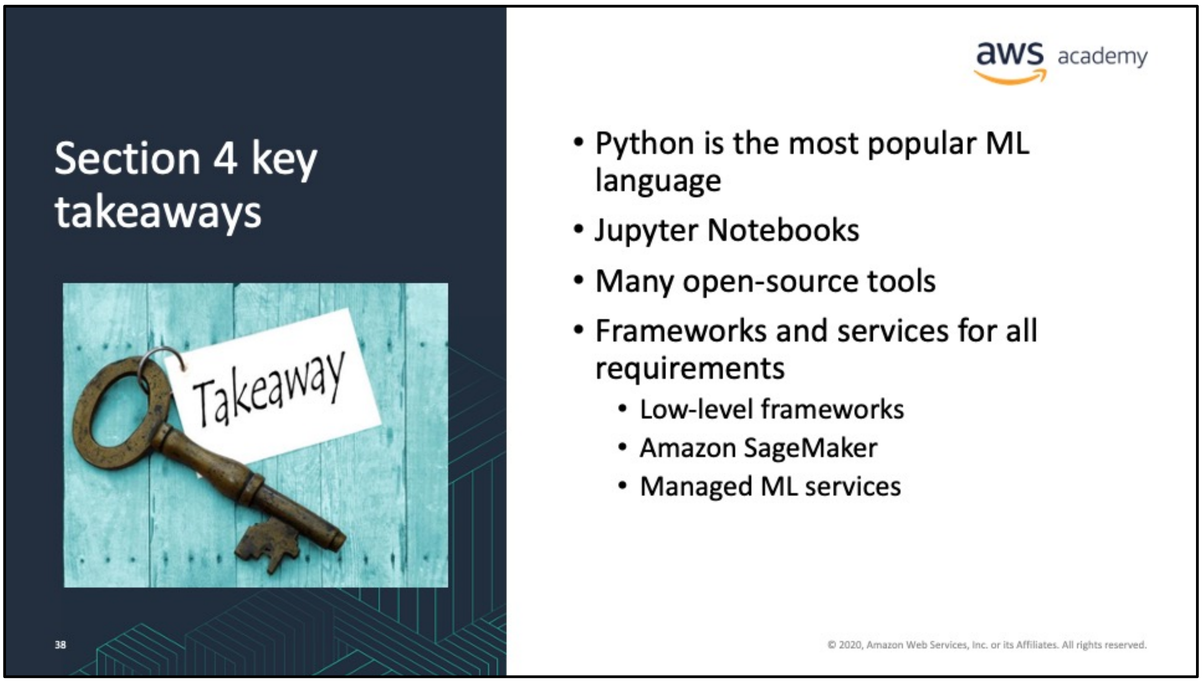


AWS provides a set of managed ML services, which you can integrate into your applications with no ML experience.

- Compute vision – Amazon Rekognition provides object and facial recognition for both image and video. Amazon Textract can extract text from images.
- Speech – Amazon Polly can speak text, while Amazon Transcribe converts spoken audio to text.
- Language – Amazon Comprehend uses NLP to find insights and relationships in text. Amazon Translate can translate text into different languages.
- Chatbots – Amazon Lex is a service that helps you build interactive conversational applications that use voice or text.
- Forecasting – Amazon Forecast uses machine learning to combine time series data with additional variables to build forecasts.
- Recommendations – Amazon Personalize is another machine learning service that can help you create individual personalized recommendations for customers.

These managed services are trained in many aspects of the problem domain, and you provide your specific data to start the process.

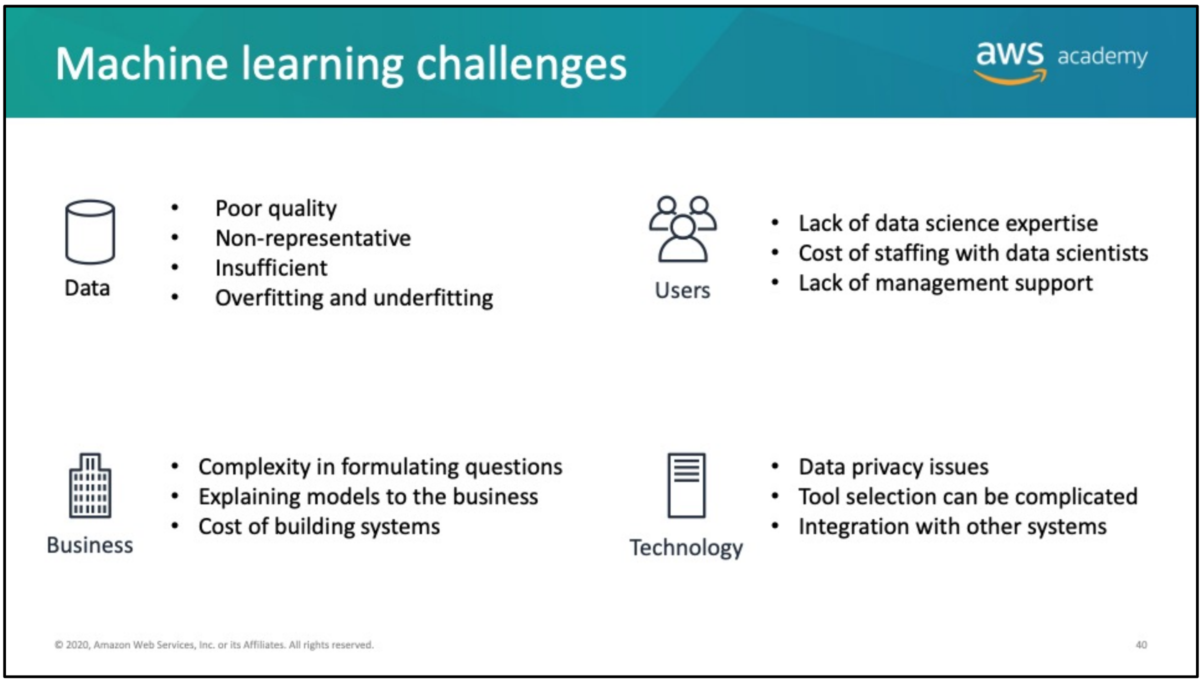
You look at many of these managed services in the second half of this course, after you learn how to do things on your own.



Some key takeaways from this section of the module include:

- Python is the most popular language for performing machine learning tasks.
- Jupyter Notebooks provide you with a web browser hosted development environment for machine learning.
- A large number of open-source tools exist, and only a few key ones are introduced in this course.
- Depending on your requirements, you can start with the low-level frameworks and use your own solution. You might use tools like Amazon SageMaker to help with larger tasks, or adapt one of the managed ML services for your specific problem domain.

Introducing Section 5: Machine learning challenges.



You can expect to encounter many challenges in machine learning, as in the following examples.

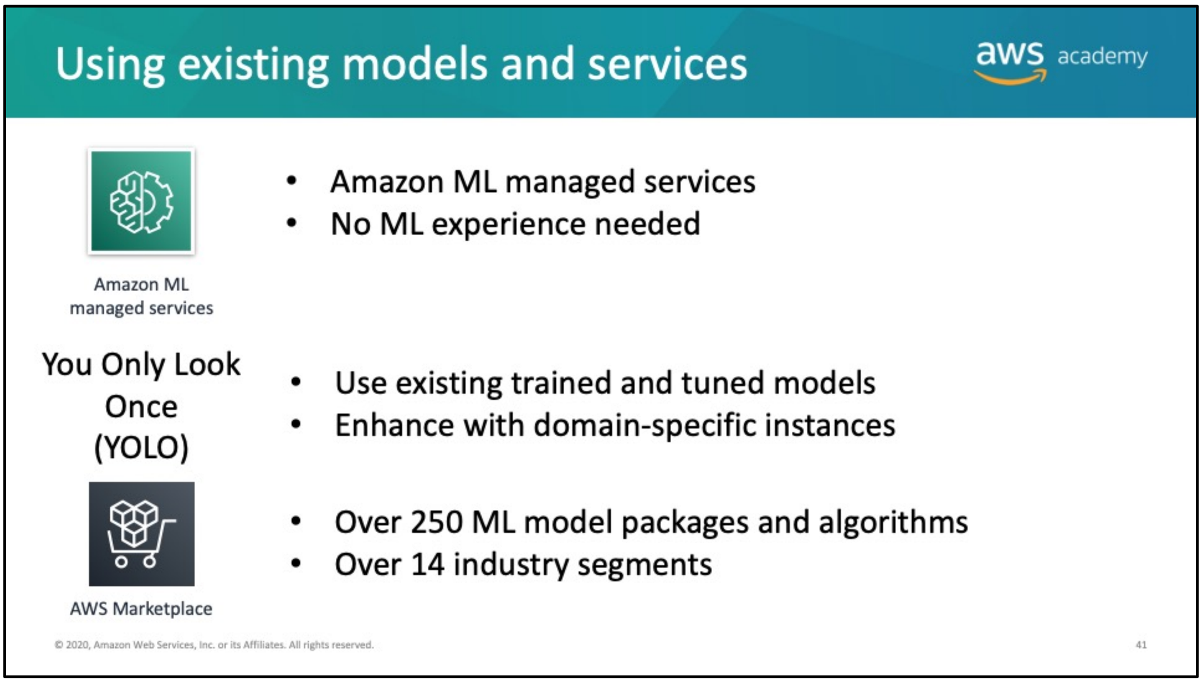
Data:
- There is a large amount of poor-quality and inconsistent data in the world. Much of your job involves getting good data.
- Does the data represent the problem well? If you try to find credit card fraud, do you have examples to train with?
- Do you have enough data? In most cases, the more data you have, the better.
- Are overfitting or underfitting your model?

Users:
- Do you have data-science experience?
- Is staffing a team of data scientists cost-effective?
- Does management support the use of ML?

Business:
- Are the problems too complex to formulate into an ML problem?
- Can the resulting model be explained to the business? If it cannot, it might not get adopted.

- What is the cost of building, updating, and operating an ML solution?

Technology:
- Does the business unit have access to the data that you need? Can the data be secured to meet any regulatory requirements?
- What tools and frameworks do you plan to use?
- How does this solution integrate with other systems?

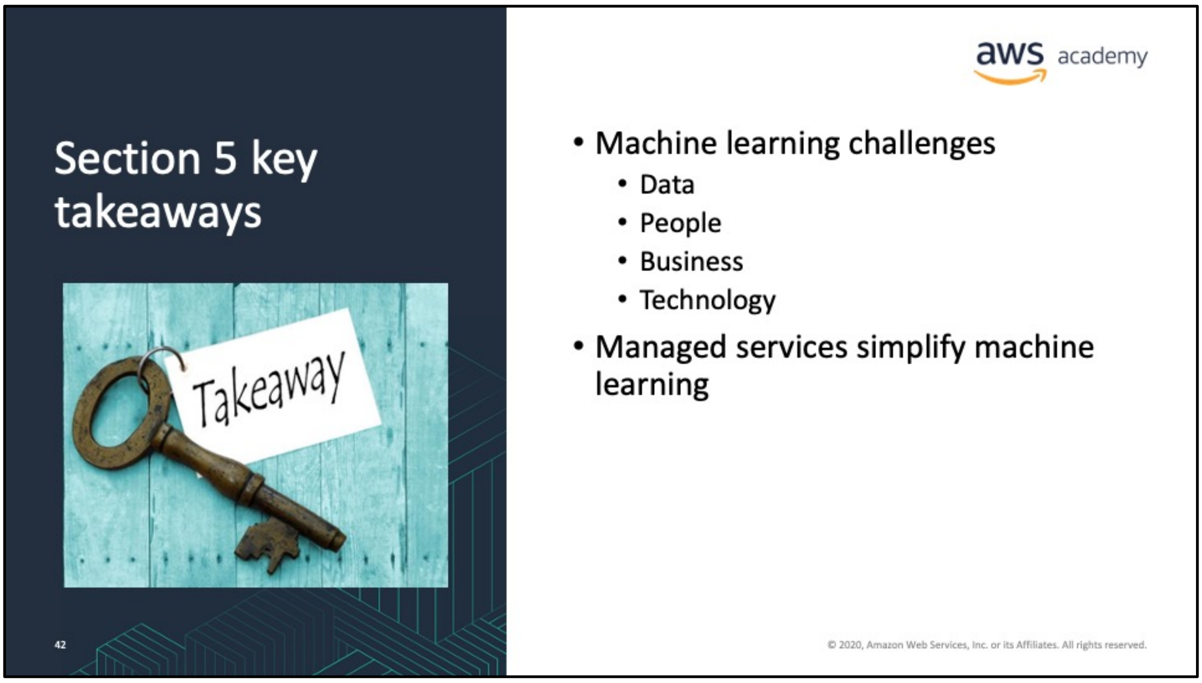


Many ML problems can be solved today—with little ML knowledge—by using existing models.

You have already learned about the AWS managed services for ML. You now need some developer skills in calling APIs, and you can add sophisticated machine learning capabilities to your applications.

You can use other pre-built models, or adapt the popular vision model You Only Look Once (YOLO).

Perhaps you don’t want to create your own algorithm, but instead you want to buy models and services that independent software vendors developed. In addition to the previous scenarios, you can also use AWS Marketplace to find third-party solutions.



Some key takeaways from this section of the module include:

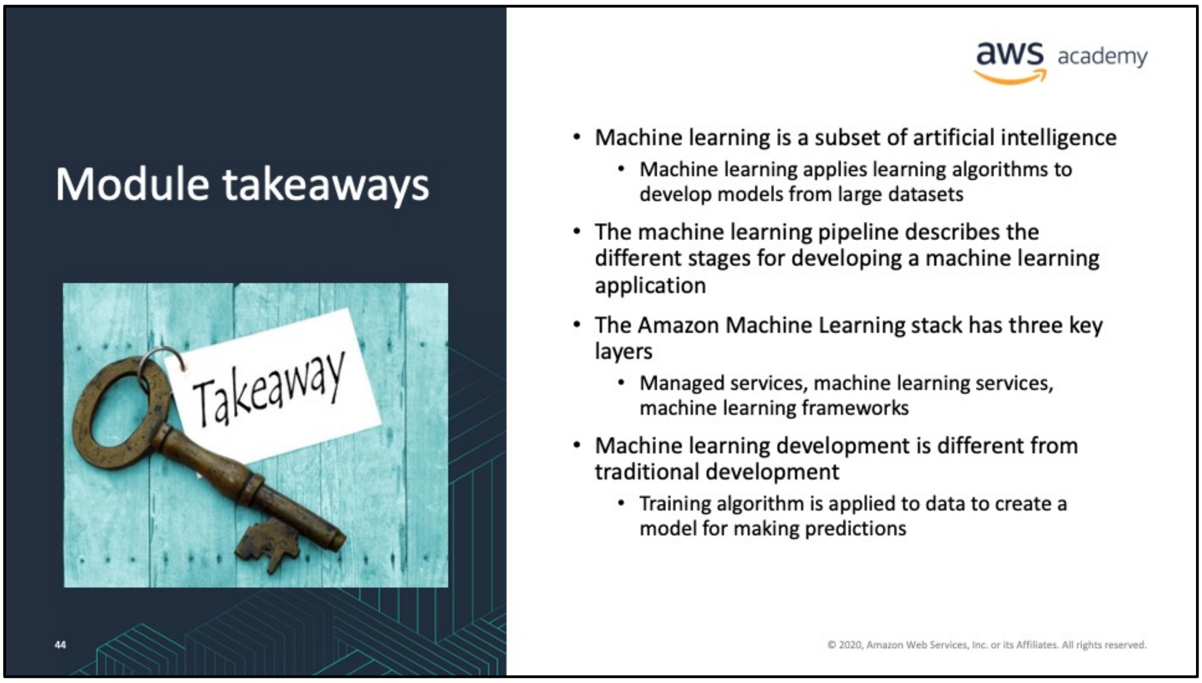
- You will face many machine learning challenges. The biggest problems that you directly influence are related to data, but you will also deal with people, business and technology challenges.
- Consider managed services to help you solve machine learning problems.

Module 2: Introduction to Machine Learning

Module wrap-up

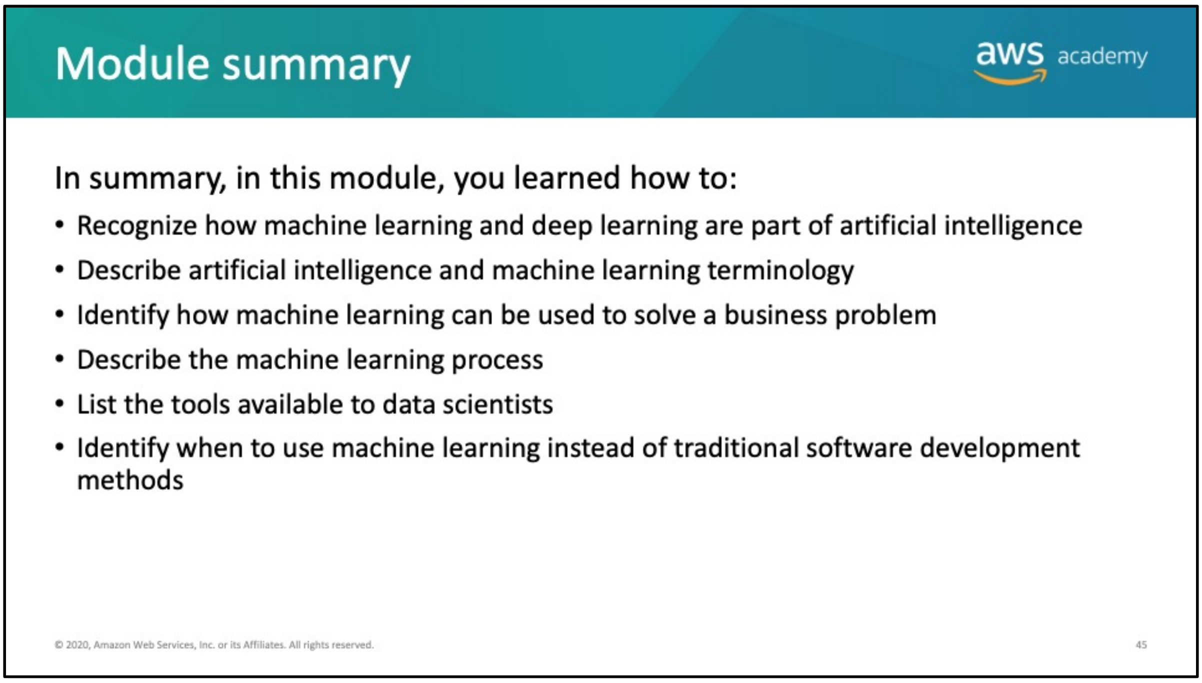
aws academy

It’s now time to review the module and wrap up with a knowledge check.



Some key takeaways from this section of the module include:

- Machine learning is a subset of artificial intelligence.
    - Machine learning applies learning algorithms to develop models from large datasets.
- The machine learning pipeline describes the different stages for developing a machine learning application.
- The machine learning stack has three key layers.
    - API services, machine learning services, and machine learning frameworks
- Machine learning development is different from traditional development.
    - Training algorithms are applied to data to create a model for making predictions.



In summary, in this module you learned how to:

- Recognize how machine learning and deep learning are part of artificial intelligence
- Describe artificial intelligence and machine learning terminology
- Identify how machine learning can be used to solve a business problem
- Describe the machine learning process
- List the tools available to data scientists
- Identify when to use machine learning instead of traditional software development methods

Complete the knowledge check

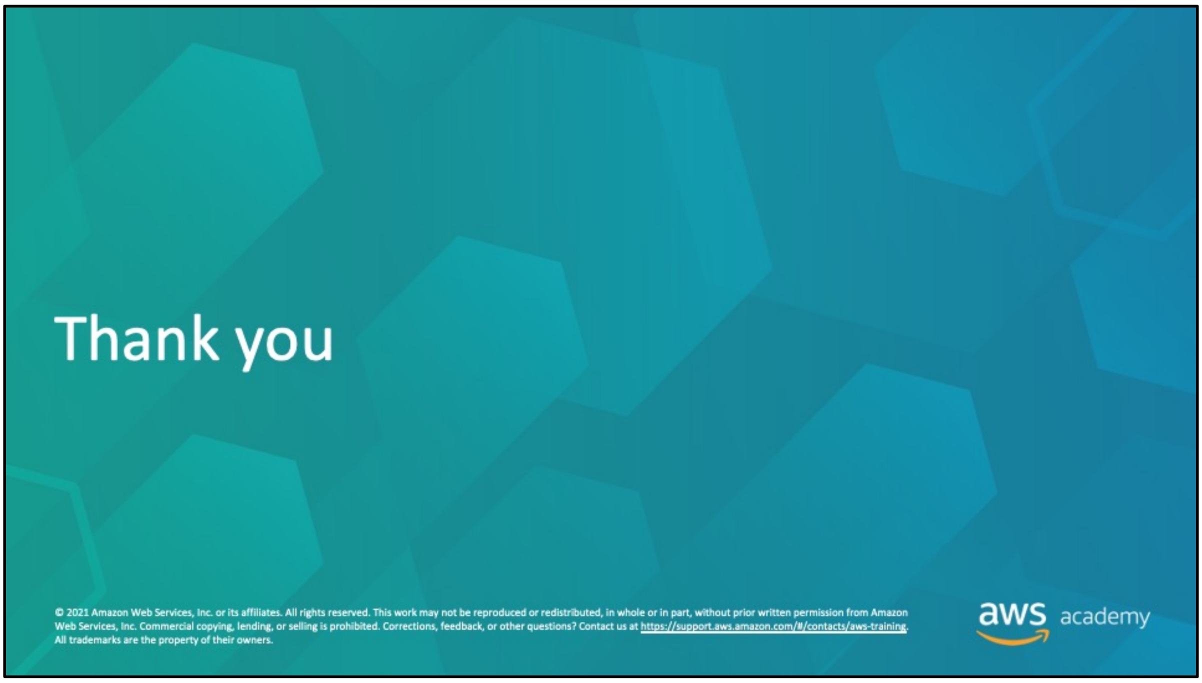
It is now time to complete the knowledge check for this module.

Additional resources

- What is Machine Learning?
- Machine Learning on AWS

If you want to learn more about the topics covered in this module, you might find the following additional resources helpful:

- [What is Machine Learning?]
- [Machine Learning on AWS]



Thank you for completing this module.